MITSUBISHI
ELECTRIC

Mitsubishi Safety Programmable Controller

MELSEC QS series

QSCPU Programming Manual (Common Instructions)

● SAFETY PRECAUTIONS ●

(Always read these cautions before using the product)

Before using this product, please read this manual and the related manuals introduced in this manual, and pay full attention to safety to handle the product correctly.

Please store this manual in a safe place and make it accessible when required. Always forward a copy of the manual to the end user.

● CONDITIONS OF USE FOR THE PRODUCT ●

(1) Although MELCO has obtained the certification for Product's compliance to the international safety standards IEC61508, EN954-1/ISO13849-1 from TUV Rheinland, this fact does not guarantee that Product will be free from any malfunction or failure. The user of this Product shall comply with any and all applicable safety standard, regulation or law and take appropriate safety measures for the system in which the Product is installed or used and shall take the second or third safety measures other than the Product. MELCO is not liable for damages that could have been prevented by compliance with any applicable safety standard, regulation or law.

(2) MELCO prohibits the use of Products with or in any application involving, and MELCO shall not be liable for a default, a liability for defect warranty, a quality assurance, negligence or other tort and a product liability in these applications.
   (a) power plants,
   (b) trains, railway systems, airplanes, airline operations, other transportation systems,
   (c) hospitals, medical care, dialysis and life support facilities or equipment,
   (d) amusement equipments,
   (e) incineration and fuel devices,
   (f) handling of nuclear or hazardous materials or chemicals,
   (g) mining and drilling,
   (h) and other applications where the level of risk to human life, health or property are elevated.

REVISIONS

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

Print Date	*Manual Number	Revision
Sep., 2006	SH(NA)-080628ENG-A	First edition
Sep.,2007	SH(NA)-080628ENG-B	Partial corrections Section 3.2, 5.2.3, 5.3.3, 5.4.1, 6.2.5, 6.3.1, 6.3.2, 6.4.1, INDEX
Apr.,2008	SH(NA)-080628ENG-C	Partial corrections ABOUT MANUALS, Section 1.2, 3.2.2, 3.2.3, Chapter 4, Section 6.1.1, 6.1.2, 6.2.1, 6.2.2, 6.2.3, 6.2.4, 6.2.5, 6.2.6, 6.3.1, 6.3.2, 6.3.3, 6.3.4, 7.1.1, 7.1.2, 7.1.3, 7.1.4, 9.1.1, 9.1.4, 9.1.5, 9.1.6, 9.1.8, Appendix1.1, Appendix 2, Appendix3
May, 2011	SH(NA)-080628ENG-D	Partial corrections ABOUT MANUALS, Section 1.2, 9.1.1, 9.1.4, 9.1.5, Appendix1.1, Appendix 2, Appendix3

Japanese Manual Version SH-080610-D

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2006 MITSUBISHI ELECTRIC CORPORATION

INTRODUCTION

Thank you for choosing the Mitsubishi MELSEC-QS Series of Safety Programmable Logic Controllers.
Before using the equipment, please read this manual carefully to develop full familiarity with the functions and performance of the QS series PLC you have purchased, so as to ensure correct use.
A copy of this manual should be forwarded to the end User.

CONTENTS

ABOUT MANUALS

Introduction Manual

Read the following manual before designing and constructing a safety system.

Manual Name	Manual Number (Model Code)
Safety Application Guide Explains the overview, construction method, laying and wiring examples, and application programs of the safety-related system. (Sold separately)	SH-080613ENG (13JR90)

Related Manuals

The manuals related to this product are listed below.

Please place an order as needed.

Manual Name	Manual Number (Model Code)
QSCPU User’s Manual (Hardware Design, Maintenance and Inspection) Explains the specifications of the QSCPU, safety power supply module, and safety base unit. (Sold separately)	SH-080626ENG (13JR92)
QSCPU User's Manual (Function Explanation, Program Fundamentals) Explains the functions, programming methods, devices and others that are necessary to create programs with the QSCPU. (Sold separately)	SH-080627ENG (13JR93)
CC-Link Safety System Master Module User's Manual Explains the specifications, procedures and settings before operation, parameter settings, and troubleshooting of the QS0J61BT12 CC-Link Safety system master module. (Sold separately)	SH-080600ENG (13JR88)
CC-Link Safety System Remote I/O Module User's Manual Explains the specifications, procedures and settings before operation, parameter settings, and troubleshooting of the CC-Link Safety system remote I/O modules. (Sold separately)	SH-080612ENG (13JR89)
MELSEC-QS CC-Link IE Field Network Master/Local Module User's Manual Explains the system configuration, performance specifications, functions, handling, wiring, and troubleshooting of the CC-Link IE Field Network master/local module (with safety functions). (Sold separately)	SH-080969ENG 13JZ53
CC-Link IE Controller Network Reference Manual Explains the system configuration, specifications, functions, handling, wiring, and troubleshooting of CC-Link IE Controller Network. (Sold separately)	SH-080668ENG (13JV16)
Q corresponding MELSECNET/H Network System Reference Manual (PLC to PLC network) Explains the specifications, procedures and settings before operation, parameter settings, programming, and troubleshooting of a MELSECNET/H network system for PLC to PLC network. (Sold separately)	SH-080049 (13JF92)
Q Corresponding Ethernet Interface Module User’s Manual (Basic) Explains the specifications, procedures for data communication with external devices, line connection (open/close), fixed buffer communication, random access buffer communication, and troubleshooting of the Ethernet module. (Sold separately)	SH-080009 (13JL88)

Manual Name	Manual Number (Model Code)
Q Corresponding Ethernet Interface Module User's Manual (Application) Explains the e-mail function, programmable controller CPU status monitoring function, communication function via CC-Link IE Controller Network, MELSECNET/H or MELSECNET/10, communication function using the data link instructions, file transfer function (FTP server) of the Ethernet module. (Sold separately)	SH-080010 (13JL89)
Q Corresponding MELSEC Communication Protocol Reference Manual Explains the communication methods and control procedures using the MC protocol, which is used by external devices to read and write data of the programmable controller CPU via the serial communication module or Ethernet module. (Sold separately)	SH-080008 (13JF89)
GX Developer Version 8 Operating Manual Explains the online functions of the GX Developer, such as the programming, printout, monitoring, and debugging methods. (Sold separately)	SH-080373E (13JU41)
GX Developer Version 8 Operating Manual (Safety Programmable Controller) Explains the GX Developer functions added and modified for the compatibility with the safety programmable controller. (Sold separately)	SH-080576ENG (13JU53)

Remark

Printed materials are separately available for single item purchase. Order the manual by quoting the manual number on the table above (Model code).

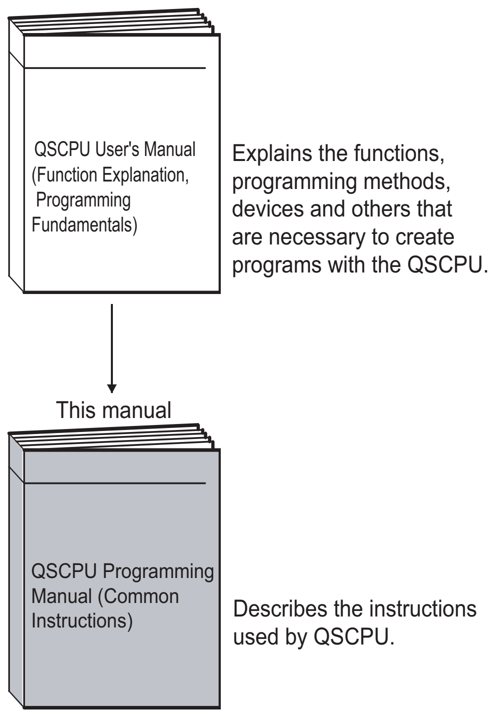
1 GENERAL DESCRIPTION

This manual describes the instructions required to execute programming of the QSCPU.

1.1 Manuals Essential for Programming

Before reading this manual, check the functions, programming methods, devices and others that are necessary to create programs with the QSCPU in the manuals below:

- QSCPU User's Manual (Function Explanation, Program Fundamentals)



1.2 Generic Terms and Abbreviations

This manual describes the QS series CPU module using the following generic terms and abbreviations, unless otherwise specified.

Generic term/Abbreviation	Description
Safety PLC	Generic term for safety CPU module, safety power supply module, safety main base unit, CC-Link safety master module, CC-Link safety remote I/O module, and CC-Link IE Field Network master/local module (with safety functions).
Standard PLC	Generic term of each module for MELSEC-Q series, MELSEC-L series, MELSEC-QnA series, MELSEC-A series and MELSEC-FX series. (Used for distinction from safety PLC.)
QS Series	Abbreviation for Mitsubishi Safety PLC MELSEC-QS Series
QS001CPU	Abbreviation for QS001CPU-type safety CPU module
CPU Module	Other name for QS001CPU
GX Developer	Generic product name for product models SW8D5C-GPPW-E, SW8D5C-GPPW-EA, SW8D5C-GPPW-EV and SW8D5C-GPPW-EVA
QS0J61BT12	Abbreviation for QS0J61BT12-type CC-Link Safety system master module
CC-Link Safety	Abbreviation for the CC-Link Safety system
CC-Link Safety master module	Other name for QS0J61BT12
QS0J65BTS2-8D	Abbreviation for the QS0J65BTS2-8D CC-Link Safety remote I/O module
QS0J65BTS2-4T	Abbreviation for the QS0J65BTS2-4T CC-Link Safety remote I/O module
QS0J65BTB2-12DT	Abbreviation for the QS0J65BTB2-12DT CC-Link Safety remote I/O module
CC-Link Safety remote I/O module	Generic term for the QS0J65BTS2-8D, QS0J65BTS2-4T, and QS0J65BTB2-12DT
CC-Link IE Field Network master/local module (with safety functions)	Abbreviation for MELSEC-QS series CC-Link IE Field Network master/ local module
CC-Link IE Controller Network module	Abbreviation for the QJ71GP21-SX CC-Link IE Controller Network module and QJ71GP21S-SX CC-Link IE Controller Network module (with external power supply function)
MELSECNET/H	Abbreviation for the MELSECNET/H network system
MELSECNET/H module	Abbreviation for the QJ71LP21-25, QJ71LP21S-25, QJ71LP21G, QJ71BR11 MELSECNET/H network module
Ethernet	Abbreviation for the Ethernet network system
Ethernet module	Abbreviation for the QJ71E71-100, QJ71E71-B5, QJ71E71-B2 Ethernet interface module
Intelligent function module	Generic term for the CC-Link Safety master module, CC-Link IE Field Network master/local module (with safety functions), CC-Link IE Controller Network module, MELSECNET/H module, and Ethernet module

MEMO

2 INSTRUCTION TABLES

2.1 Types of Instructions

The major types of safety CPU module instructions are sequence instructions, basic instructions, application instructions, and QSCPU dedicated instructions as shown in Table 2.1

Table 2.1 Types of Instructions

Types of Instructions		Meaning	Reference Chapter
Sequence instruction	Contact instruction	Operation start, series connection, parallel connection	5
	Connection instructions	Ladder block connection, store/read operation results, creation of pulses from operation results	
	Output instruction	Bit device output, output reversal	
	Master control instruction	Master control	
	Termination instruction	Program termination	
	Other instructions	Instructions such as no operation which do not fit in the above categories	
Basic instructions	Comparison operation instruction	Comparisons such as =, >, <	6
	Arithmetic operation instruction	Addition, subtraction, multiplication or division of BIN	
	BCD ↔ BIN conversion instruction	Conversion from BCD to BIN and from BIN to BCD	
	Data transfer instruction	Transmits designated data	
Application instructions	Logical operation instructions	Logical operations such as logical sum, logical product, etc.	7
QSCPU dedicated instruction	QSCPU dedicated instruction	Forced control stop	8

2.2 How to Read Instruction Tables

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

Table 2.2 How to Read Instruction Tables

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
BIN 16-bit addition and subtraction operations	+	+ S D	· (D)+(S) → (D)		3	●	6-6
	+P	+P S D					
	+	+ S1 S2 D	· (S1)+(S2) → (D)		4	●	6-8
	+P	+P S1 S2 D					

1) 2) 3) 4) 5) 6) 7) 8)

Description

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

2) . . . . . Indicates the instruction symbol used in a program.
Instruction code is built around the 16-bit instruction.
The following notations are used to mark 32-bit instructions, instructions executed only at the leading edge of OFF to ON.

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

Example + → D+

16-bit instruction → 32-bit instruction

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

Example + → +P

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

3) . . . . .Shows symbol diagram on the ladder.

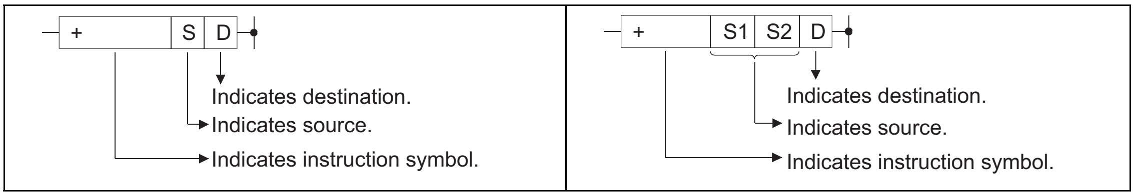


Fig. 2.1 Shows Symbol Diagram on the Ladder

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

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

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

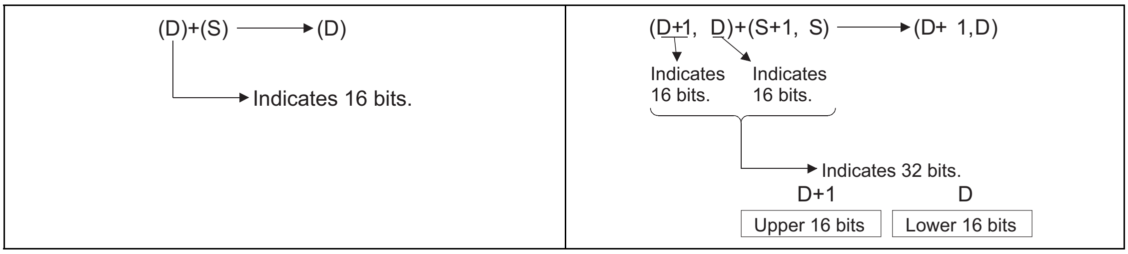
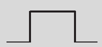
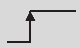
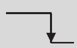


Fig. 2.2 Type of Processing Performed by Individual Instructions

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

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

6) . . . . .Indicates the basic number of steps for individual instructions.

See 3.6 for a description of the number of steps.

7) . . . . .The ● mark indicates instructions for which subset processing is possible.

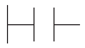
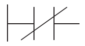
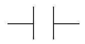
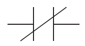
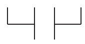
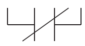
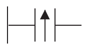
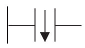
See Section 3.3 for details on subset processing.

8) . . . . .Indicates the page numbers where the individual instructions are explained.

2.3 Sequence Instruction

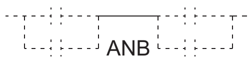
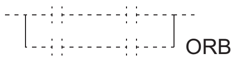
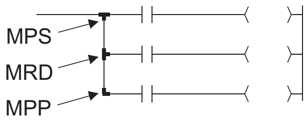
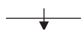
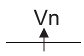
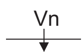
2.3.1 Contact instructions

Table 2.3 Contact Instructions

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

2.3.2 Connection instructions

Table 2.4 Connection Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Connection	ANB		• AND between logical blocks (Series connection between logical blocks)		1	-	5-7
	ORB		• OR between logical blocks (Series connection between logical blocks)				
	MPS		• Memory storage of operation results		1	-	5-9
	MRD		• Read of operation results stored with MPS instruction				
	MPP		• Read and reset of operation results stored with MPS instruction				
	INV		• Inversion of operation result		1	-	5-12
	MEP		• Conversion of operation result to leading edge pulse		1	-	5-14
	MEF		• Conversion of operation result to trailing edge pulse				
	EGP		• Conversion of operation result to leading edge pulse (Stored at Vn)		1	-	5-15
	EGF		• Conversion of operation result to trailing edge pulse (Stored at Vn)		2		

2.3.3 Output instructions

Table 2.5 Output Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Output	OUT	─(D)─	• Device output		*1	-	5-17 5-19 5-23 5-25
	SET	SET D	• Set device	*2	*1	-	5-27 5-31
	RST	RST D	• Reset device	*2	*1	-	5-29 5-31
	PLS	PLS D	• Generates 1 cycle program pulse at leading edge of input signal.		2	-	5-33
	PLF	PLF D	• Generates 1 cycle program pulse at trailing edge of input signal.				
	FF	FF D	• Reversal of device output		2	-	5-36

*1: The number of steps may vary depending on the device being used.
See description pages of individual instructions for number of steps.

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

2.3.4 Master control instructions

Table 2.6 Master Control Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Master control	MC	MC n D	• Starts master control		2	-	5-38
	MCR	MCR n	• Resets master control		1		

2.3.5 Termination instruction

Table 2.7 Termination Instruction

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Termination	END	[END]	• Termination of sequence program		1	-	5-42

2.3.6 Other instructions

Table 2.8 Other Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
No-operation	NOP	———	• Ignored (For program deletion or space)		1	-	5-43
	NOPLF	[NOPLF]	• Ignored (To change pages during printouts)				
	PAGE	[PAGE n]	• Ignored (Subsequent programs will be controlled from step 0 of page n)				

2.4 Basic Instructions

2.4.1 Comparison operation instructions

Table 2.9 Comparison Operation Instructions

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

Table 2.9 Comparison Operation Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
BIN 32-bit data comparisons	LDD =	D= S1 S2	• Conductive status when (S1+1, S1) = (S2+1, S2) • Non-Conductive status when (S1+1, S1) ≠ (S2+1, S2)		3	●	6-4
	ANDD =	D= S1 S2					
	ORD =	D = S1 S2					
	LDD<>	D<> S1 S2	• Conductive status when (S1+1, S1) ≠ (S2+1, S2) • Non-Conductive status when (S1+1, S1) = (S2+1, S2)		3	●	
	ANDD<>	D <> S1 S2					
	ORD<>	D<> S1 S2					
	LDD>	D> S1 S2	• Conductive status when (S1+1, S1) > (S2+1, S2) • Non-Conductive status when (S1+1, S1) ≦ (S2+1, S2)		3	●	
	ANDD>	D> S1 S2					
	ORD>	D> S1 S2					
	LDD<=	D<= S1 S2	• Conductive status when (S1+1, S1) ≦ (S2+1, S2) • Non-Conductive status when (S1+1, S1) > (S2+1, S2)		3	●	
	ANDD<=	D<= S1 S2					
	ORD<=	D<= S1 S2					
	LDD<	D< S1 S2	• Conductive status when (S1+1, S1) < (S2+1, S2) • Non-Conductive status when (S1+1, S1) ≧ (S2+1, S2)		3	●	
	ANDD<	D< S1 S2					
	ORD<	D< S1 S2					
	LDD>=	D>= S1 S2	• Conductive status when (S1+1, S1) ≧ (S2+1, S2) • Non-Conductive status when (S1+1, S1) < (S2+1, S2)		3	●	
	ANDD>=	D>= S1 S2					
	ORD>=	D>= S1 S2					

2.4.2 Arithmetic operation instructions

Table 2.10 Arithmetic Operation Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
BIN 16-bit addition and subtraction operations	+	+ S D	• (D) + (S) → (D)		3	●	6-6
	+P	+P S D					
	+	+ S1 S2 D	• (S1) + (S2) → (D)		4	●	6-8
	+P	+P S1 S2 D					
	–	– S D	• (D) – (S) → (D)		3	●	6-6
	–P	–P S D					
	–	– S1 S2 D	• (S1) – (S2) → (D)		4	●	6-8
	–P	–P S1 S2 D					
BIN 32-bit addition and subtraction operations	D+	D+ S D	• (D+1, D) + (S+1, S) → (D+1, D)		3	●	6-10
	D+P	D+P S D					
	D+	D+ S1 S2 D	• (S1+1, S1) + (S2+1, S2) → (D+1, D)		4	●	6-12
	D+P	D+P S1 S2 D					
	D–	D– S D	• (D+1, D) – (S+1, S) → (D+1, D)		3	●	6-10
	D–P	D–P S D					
	D–	D– S1 S2 D	• (S1+1, S1) – (S2+1, S2) → (D+1, D)		4	●	6-12
	D–P	D–P S1 S2 D					
BIN 16-bit multiplication and division operations	*	* S1 S2 D	• (S1) × (S2) → (D+1,D)		4	●	6-14
	*P	*P S1 S2 D					
	/	/ S1 S2 D	• (S1) / (S2) → Quotient(D), Remainder (D+1)		4	●	
	D/P	/P S1 S2 D					
BIN 32-bit multiplication and division operations	D *	D * S1 S2 D	• (S1+1, S1) × (S2+1, S2) → (D+3, D+2, D+1, D)		4	●	6-16
	D * P	D * P S1 S2 D					
	D/	D/ S1 S2 D	• (S1+1, S1) / (S2+1, S2) → Quotient (D+1, D), Remainder (D+3, D+2)		4	●	
	D/P	D/P S1 S2 D					

Table 2.10 Arithmetic Operation Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
BIN data increment	INC	INC D	• (D) + 1 → (D)		2	●	6-18
	INCP	INCP D					
	DINC	DINC D	• (D+1, D) + 1 → (D+1, D)		2	●	6-20
	DINCP	DINCP D					
	DEC	DEC D	• (D) − 1 → (D)		2	●	6-18
	DECP	DECP D					
	DDEC	DDEC D	• (D+1, D) − 1 → (D+1, D)		2	●	6-20
	DDECP	DDECP D					

2.4.3 Data conversion instructions

Table 2.11 Data Conversion Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
BCD conversions	BCD	BCD S D	• (S) —BCD conversion→ (D) (S): BIN (0 to 9999)		3	●	6-22
	BCDP	BCDP S D					
	DBCD	DBCD S D	• (S+1, S) —BCD conversion→ (D +1,D) (S+1, S): BIN (0 to 99999999)		3	●	
	DBCDP	DBCDP S D					
BIN conversions	BIN	BIN S D	• (S) —BIN conversion→ (D) (S): BCD (0 to 9999)		3	●	6-24
	BINP	BINP S D					
	DBIN	DBIN S D	• (S+1, S) —BIN conversion→ (D +1,D) (S+1, S): BCD (0 to 99999999)		3	●	
	DBINP	DBINP S D					
Complement to 2	NEG	NEG D	• $\overline{(D)}$ → (D) (D): BIN data		2	–	6-27
	NEGP	NEGP D					
	DNEG	DNEG D	• $\overline{(D+1, D)}$ → (D+1,D) (D+1, D): BIN data		2	–	
	DNEGP	DNEGP D					

2.4.4 Data transfer instructions

Table 2.12 Data Transfer Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
16-bit data transfer	MOV	MOV S D	• (S) → (D)		*1	●	6-29
	MOVP	MOVP S D					
32-bit data transfer	DMOV	DMOV S D	• (S+1,S) → (D+1,D)		*2	●	
	DMOVP	DMOVP S D					
16-bit data negation transfer	CML	CML S D	• $\overline{(S)}$ → (D)		*1	●	6-31
	CMLP	CMLP S D					
32-bit data negation transfer	DCML	DCML S D	• $\overline{(S+1,S)}$ → (D+1,D)		*2	●	
	DCMLP	DCMLP S D					
Block transfer	BMOV	BMOV S D n	(S) → (D) n		4	●	6-34
	BMOVP	BMOVP S D n					
Multiple transfers of same data block	FMOV	FMOV S D n	(S) → (D) n		4	●	6-36
	FMOVP	FMOVP S D n					

*1: The number of steps may vary depending on the device being used.

Device		Number of Steps
· Word device:	Internal device	2
· Bit device:	Devices whose device Nos. are multiples of 16 and whose digit designation is K4	
· Constant:	No limitations	
Devices other than above		3

*2: The number of steps may vary depending on the device being used.

Device		Number of Steps
· Word device:	Internal device	3
· Bit device:	Devices whose device Nos. are multiples of 16 and whose digit designation is K8	
· Constant:	No limitations Note 1)	
Devices other than above		3 Note 1)

Note 1) The number of steps may increase due to the conditions described in 3.6.

2.5 Application Instructions

2.5.1 Logical operation instructions

Table 2.13 Logical Operation Instructions

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Logical product	WAND	WAND S D	· (D)∧(S)→(D)		3	●	7-3
	WANDP	WANDP S D					
	WAND	WAND S1 S2 D	· (S1)∧(S2)→(D)		4	●	7-5
	WANDP	WANDP S1 S2 D					
	DAND	DAND S D	· (D+1,D)∧(S+1,S)→(D+1,D)		*1 3	●	7-3
	DANDP	DANDP S D					
	DAND	DAND S1 S2 D	· (S1+1,S1)∧(S2+1,S2)→(D+1,D)		*1 4	●	7-5
	DANDP	DANDP S1 S2 D					
Logical sum	WOR	WOR S D	· (D)∨(S)→(D)		3	●	7-8
	WORP	WORP S D					
	WOR	WOR S1 S2 D	· (S1)∨(S2)→(D)		4	●	7-10
	WORP	WORP S1 S2 D					
	DOR	DOR S D	· (D+1,D)∨(S+1,S)→(D+1,D)		*1 3	●	7-8
	DORP	DORP S D					
	DOR	DOR S1 S2 D	· (S1+1,S1)∨(S2+1,S2)→(D+1,D)		*1 4	●	7-10
	DORP	DORP S1 S2 D					
Exclusive OR	WXOR	WXOR S D	· (D)⊻(S)→(D)		3	●	7-12
	WXORP	WXORP S D					
	WXOR	WXOR S1 S2 D	· (S1)⊻(S2)→(D)		4	●	7-14
	WXORP	WXORP S1 S2 D					
	DXOR	DXOR S D	· (D+1,D)⊻(S+1,S)→(D+1,D)		*1 3	●	7-12
	DXORP	DXORP S D					
	DXOR	DXOR S1 S2 D	· (S1+1,S1)⊻(S2+1,S2)→(D+1,D)		*1 4	●	7-14
	DXORP	DXORP S1 S2 D					

*1: The number of steps may increase due to the conditions described in 3.6.

Table 2.13 Logical Operation Instructions (Continued)

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
NON exclusive logical sum	WXNR	WXNR S D	· $\overline{(D) \veebar (S)} \rightarrow (D)$		3	●	7-16
	WXNRP	WXNRP S D					
	WXNR	WXNR S1 S2 D	· $\overline{(S1) \veebar (S2)} \rightarrow (D)$		4	●	7-18
	WXNRP	WXNRP S1 S2 D					
	DXNR	DXNR S D	· $\overline{(D+1,D) \veebar (S+1,S)} \rightarrow (D+1,D)$		*1 3	●	7-16
	DXNRP	DXNRP S D					
	DXNR	DXNR S1 S2 D	· $\overline{(S1+1,S1) \veebar (S2+1,S2)} \rightarrow (D+1,D)$		*1 4	●	7-18
	DXNRP	DXNRP S1 S2 D					

*1: The number of steps may increase due to the conditions described in 3.6.

2.6 QSCPU Dedicated Instruction

2.6.1 Forced control stop instruction

Table 2.14 Forced Control Stop Instruction

Category	Instruction Symbols	Symbol	Processing Details	Execution Condition	Number of Basic Steps	Subset	See for Description
Forced control stop	S.QSABORT	─┤ S.QSABORT │ S ├─	• Stops program execution. • Places safety CPU module in the stop error state.	⎍	7 *1	–	8-2

*1: 8 steps when a constant is used.

3 CONFIGURATION OF INSTRUCTIONS

3.1 Configuration of Instructions

Most safety CPU module instructions consist of an instruction part and a device part.

Each part is used for the following purpose:

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

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

(1) Source (S)

(a) Source is the data used for operations.

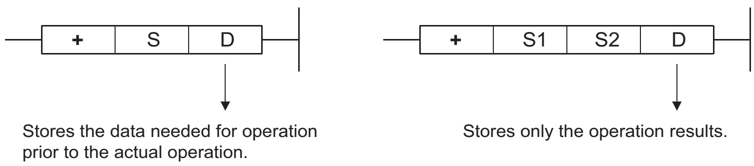
(b) The following source types are available, depending on the designated device:

- Constant .... Designates a numeric value to be used in the operation. This is set when the program is created, and cannot be changed during the execution of the program.
- Bit devices and word devices .... Designates the device that stores the data to be used in the operation. Data must be stored in the designated device until the operation is executed. By changing the data stored in a designated device during program execution, the data to be used in the instruction can be changed.

(2) Destination (D)

(a) The destination stores the data after the operation has been conducted. However, some instructions require storing the data to be used in an operation at the destination prior to the operation execution.

Example An addition instruction involving BIN 16-bit data

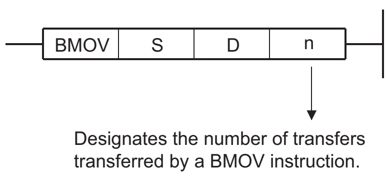


(b) A device for the data storage must always be set to the destination.

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

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

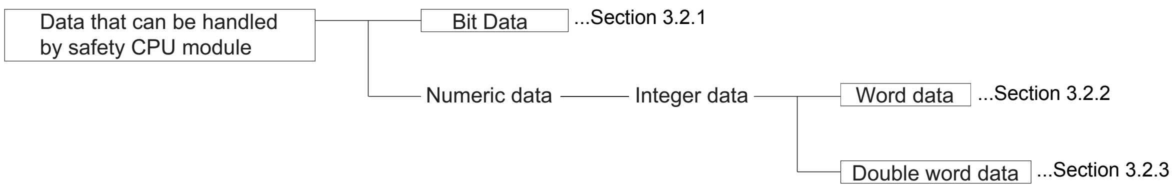
Example Block transfer instruction



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

3.2 Designating Data

The following three types of data can be used with safety CPU module instructions:



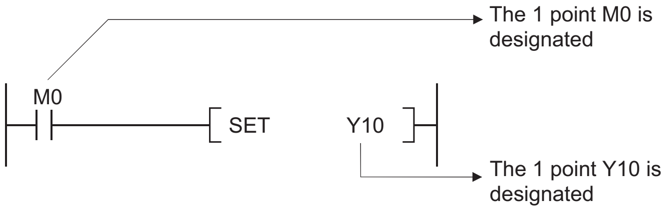
3.2.1 Using bit data

Bit data is data used in one-bit units, such as for contact points or coils.

"Bit devices" and "Bit designated word devices" can be used as bit data.

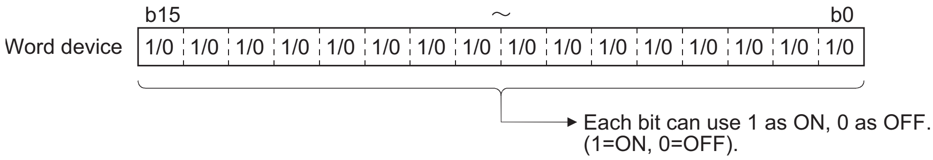
(1) When using bit devices

Bit devices are designated in one-point units.

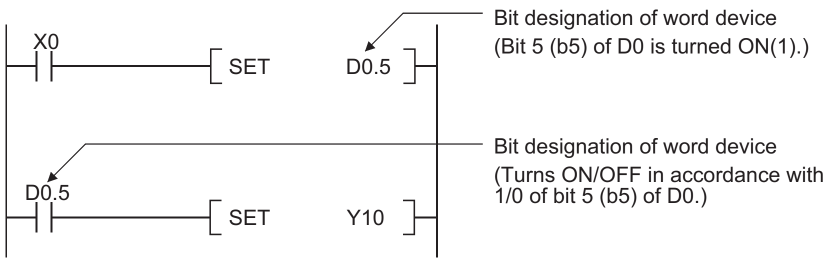


(2) Using word devices

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



(b) Word device bit designation is done by designating " **Word device** . **Bit No.** ".
(Designation of bit numbers is done in hexadecimal.)
For example, bit 5 (b5) of D0 is designated as D0.5, and bit 10 (b10) of D0 is designated as D0.A.
However, there can be no bit designation for timers (T), retentive timers (ST) or counters (C). (Example: C0.0 is not available)



3.2.2 Using word (16 bits) data

Word data is 16-bit numeric data used by basic instructions and application instructions.

The following two types of word data can be used with safety CPU module:

- Decimal constants...................... K-32768 to K32767
- Hexadecimal constants.............. H0000 to HFFFF

Word devices and bit devices designated by digit can be used as word data.

(1) When Using Bit Devices

(a) Bit devices can deal with word data when digits are designated.

Digit designation of bit devices is done by designating "Number of digits Start number of bit device".

Digit designation of bit devices can be done in 4-point (4-bit) units, and designation can be made for K1 to K4.

For example, if X0 is designated for digit designation, the following points would be designated:

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

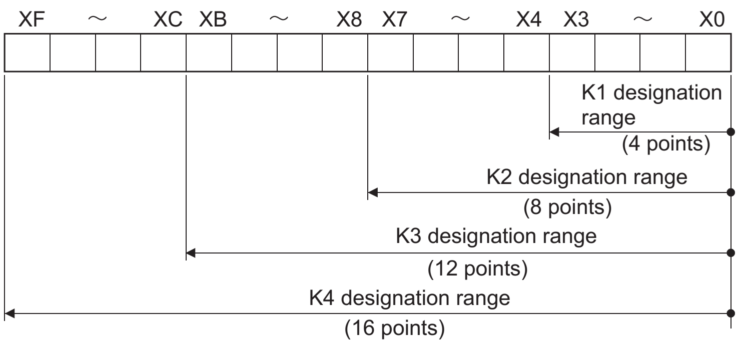


Fig 3.1 Digit Designation Setting Range for 16-Bit Instruction

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

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

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

(c) When destination (D) data is a word device
The word device for the destination becomes 0 following the bit designated by digit designation at the source.

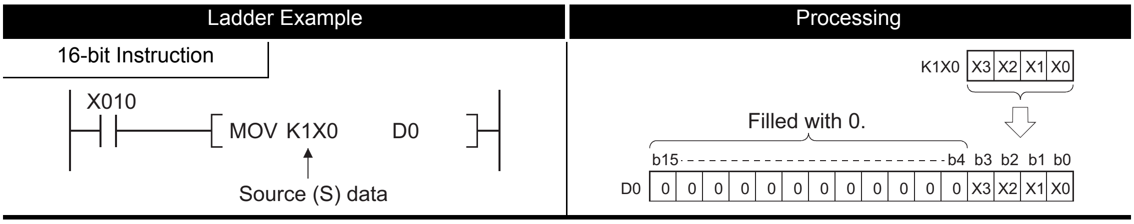


Fig 3.2 Ladder Example and Processing Conducted

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

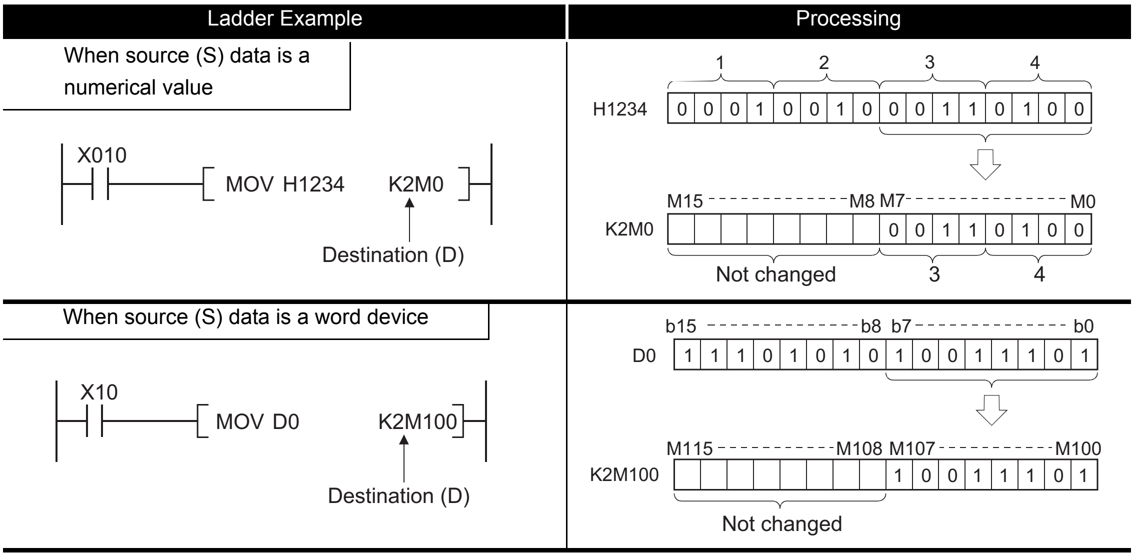
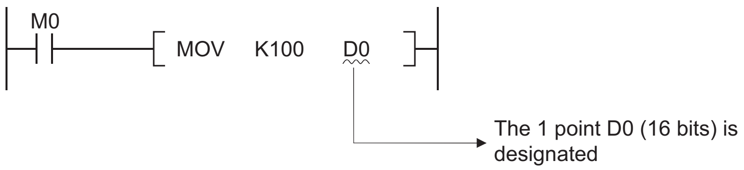


Fig 3.3 Ladder Example and Processing Conducted

(2) Using word devices

Word devices are designated in 1-point (16 bits) units.



POINT

When digit designation processing is conducted, a random value can be used for the bit device head device number.

3.2.3 Using double word (32 bits) data

Double word data is 32-bit numerical data used by basic instructions and application instructions.

The two types of double word data that can be dealt with by CPU module are as follows:

- Decimal constants...................... K–2147483648 to K2147483647
- Hexadecimal constants............. H00000000 to HFFFFFFFF

Word devices and bit devices designated by digit designation can be used as double word data.

(1) When Using Bit Devices

(a) Digit designation can be used to enable a bit device to deal with double word data.

Digit designation of bit devices is done by designating "Number of digits Start number of bit device".

Digit designation of bit devices can be done in 4-point (4-bit) units, and designation can be made for K1 to K8.

For example, if X0 is designated for digit designation, the following points would be designated:

- K1X0.. The 4 points X0 to X3 are designated
- K2X0.. The 8 points X0 to X7 are designated
- K3X0.. The 12 points X0 to XB are designated
- K4X0.. The 16 points X0 to XF are designated
- K5X0... The 20 points X0 to X13 are designated
- K6X0... The 24 points X0 to X17 are designated
- K7X0... The 28 points X0 to X1B are designated
- K8X0... The 32 points X0 to X1F are designated

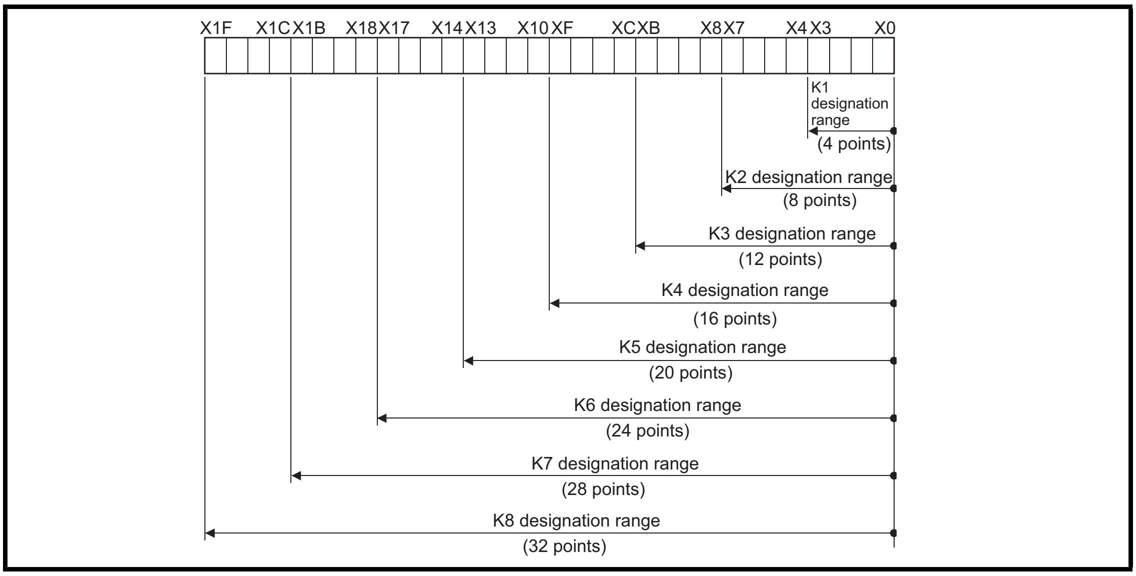


Fig 3.4 Digit Designation Setting Range for 32-Bit Instructions

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

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

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

(c) When destination (D) data is a word device
The word device for the destination becomes 0 following the bit designated by digit designation at the source.

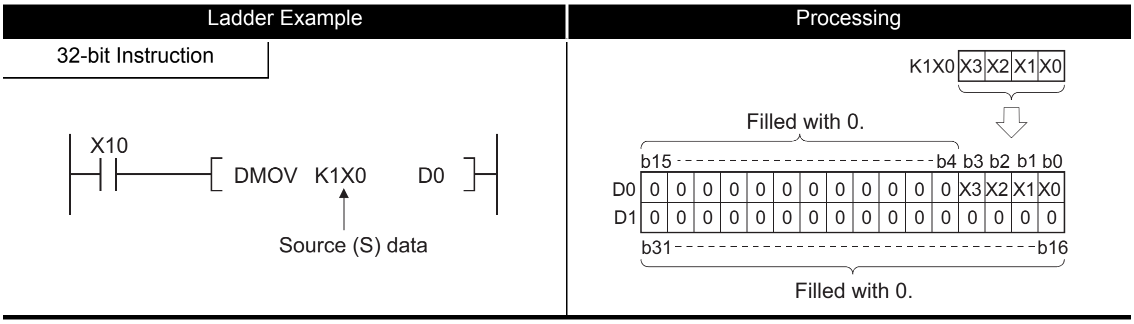


Fig 3.5 Ladder Example and Processing Conducted

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

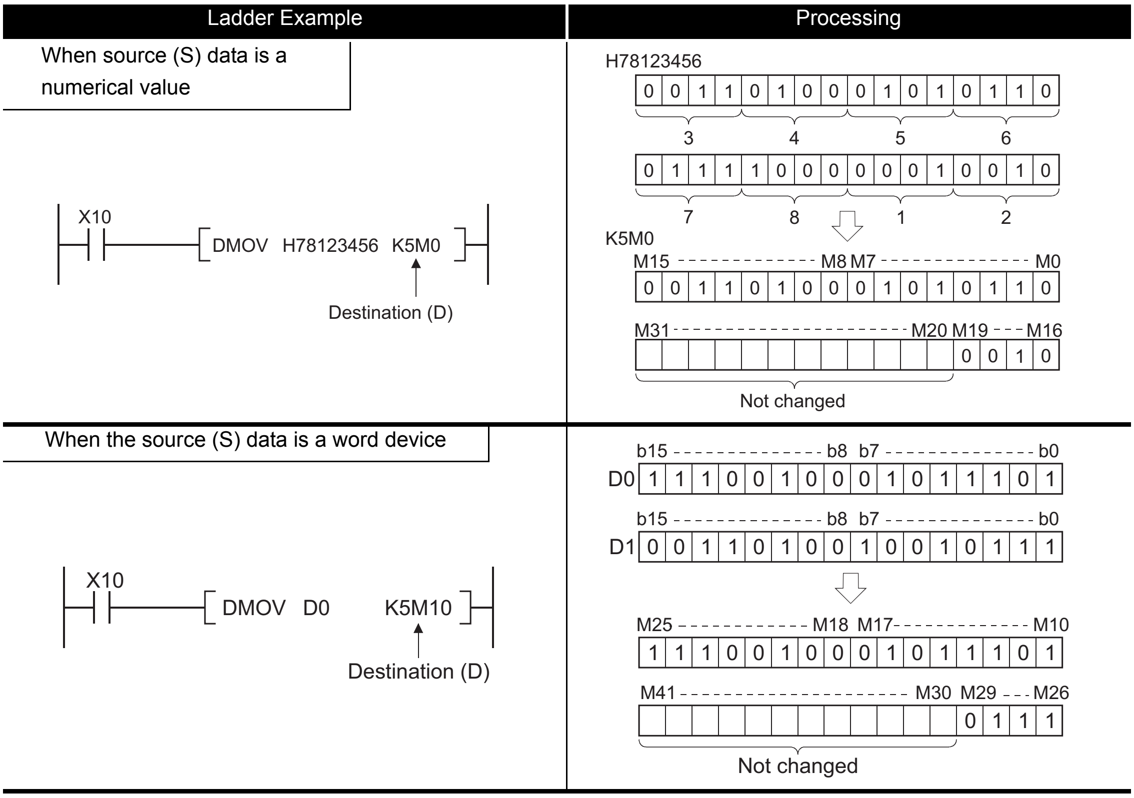


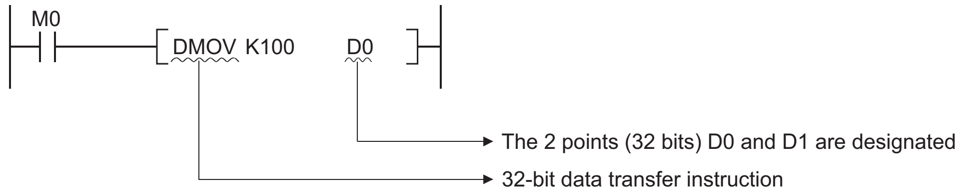
Fig 3.6 Ladder Example and Processing Conducted

POINT

When digit designation processing is conducted, a random value can be used for the bit device head device number.

(2) Using word devices

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



3.3 Subset Processing

Subset processing is used to place limits on bit devices used by basic instructions and application instructions in order to increase processing speed.

However, the instruction symbol does not change.

To shorten scans, run instructions under the conditions indicated below.

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

(a) When using word data

Device	Condition
Bit device	• Designates a bit device number in a factor of 16 • Only K4 can be designated for digit designation.
Word device	• Internal Device
Constants	• No limitations

(b) When using double word data

Device	Condition
Bit device	• Designates a bit device number in a factor of 16 • Only K8 can be designated for digit designation.
Word device	• Internal Device
Constants	• No limitations

(2) Instructions for which subset processing can be used

Types of Instructions	Instruction Symbols
Comparison operation instruction	• =, <>, <, <=, >, >=, D=, D<>, D<, D<=, D>, D>=
Arithmetic operation	• +, −, *, /, INC, DEC, D+, D−, D*, D/, DINC, DDEC
Data conversion instructions	• BCD, BIN, DBCD, DBIN
Data transfer instruction	• MOV, DMOV, CML, DCML • FMOV, BMOV
Logic operations	• WAND, DAND, WOR, DOR, WXOR, DXOR, WXNR, DXNR

3

CONFIGURATION OF INSTRUCTIONS

3.4 Cautions on Programming (Operation Errors)

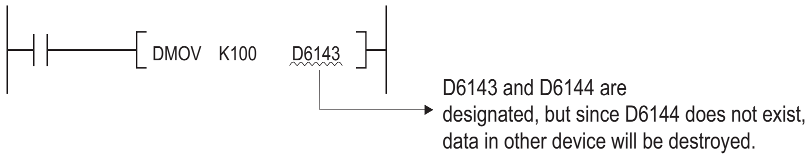
Operation errors are returned in the following cases when executing basic instructions, application instructions and QSCPU dedicated instructions with safety CPU module:

- An error listed on the explanatory page for the individual instruction occurred.

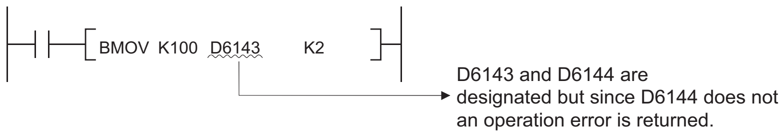
(1) Device range check

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

(a) No device range check is made for instructions dealing with fixed-length devices (MOV, DMOV, etc.).
In cases where the corresponding device range is exceeded, data is written to other devices.*1
For example, in a case where the data register has been allocated 6 k points, there will be no error even if it exceeds D6143.



(b) Device range checks are conducted for instructions dealing with variable-length devices (BMOV, FMOV, and others which designate transfer numbers).
In cases where the corresponding device range has been exceeded, an operation error will be returned.
For example, in a case where the data register has been allocated 6 k points, there will be an error if it exceeds D6143.



*1: See the figure below for the internal user device assignment order.

Head address (fixed)	SM
	X
	Y
	M
	B
	F
	SB
	V
	Contact and coil of T
	Contact and coil of ST
	Contact and coil of C
	Present value of T
	Present value of ST
	Present value of C
	D
	W
	Empty area
	SW
	SD

Remark

Refer to the manual below for how to change the internal user device allocation:
- QSCPU User's Manual (Function Explanation, Program Fundamentals)

(2) Device data check

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

(a) When using BIN data
No error is returned even if the operation results in overflow or underflow.
The carry flag does not go on at such times, either.

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

3.5 Conditions for Execution of Instructions

The following four types of execution conditions exist for the execution of safety CPU module sequence instructions, basic instructions, application instructions and QSCPU dedicated instructions:

- Non-conditional execution..... Instructions executed without regard to the ON/OFF status of the device

  Example LD X0, OUT Y10
- Executed at ON.................... Instructions executed while input condition is ON

  Example MOV instruction, FMOV instruction
- Executed at leading edge...... Instructions executed only at the leading edge of the input condition (when it goes from OFF to ON)

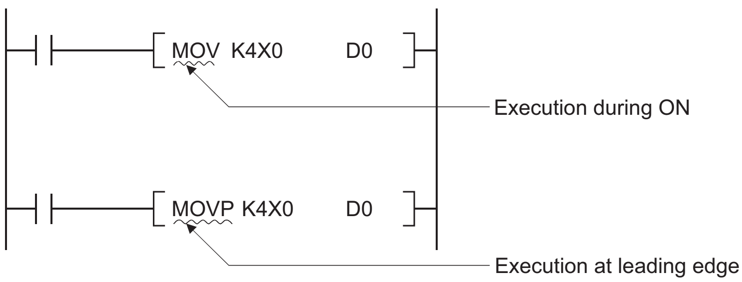
  Example PLS instruction, MOVP instruction
- Executed at trailing edge....... Instructions executed only at the trailing edge of the input condition (when it goes from ON to OFF)

  Example PLF instruction

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

- Instruction to be executed at ON Instruction
- Instruction to be executed at leading edge Instruction + P

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

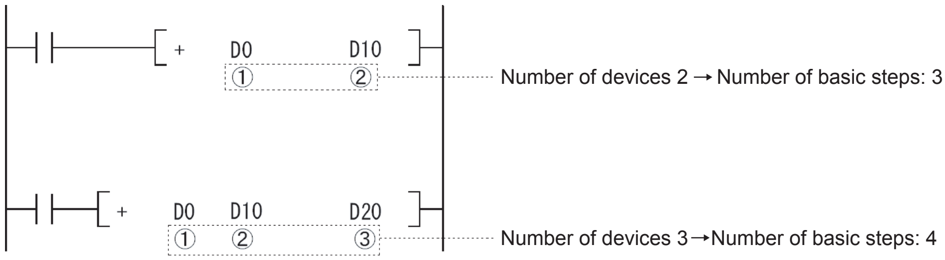


3.6 Counting Step Number

The number of steps in basic instructions and application instructions of the safety CPU module may increase depending on the devices to be used.

(1) Counting the number of basic steps

The basic number of steps for basic instructions and application instructions is calculated by adding the device number and 1.
For example, the "+ instruction" would be calculated as follows:

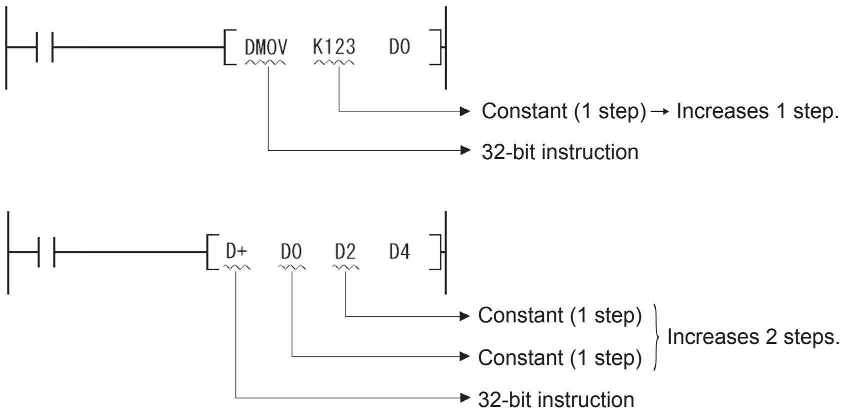


(2) Conditions for increasing the number of steps

In the following case, the number of steps increases over the number of basic steps.

- When a constant is used in device designation with a 32-bit instruction.

Example



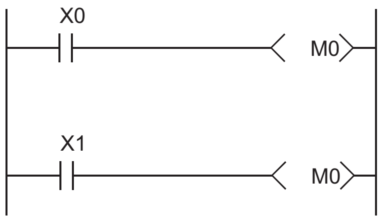
3.7 Operation when OUT, SET/RST, or PLS/PLF Instructions Use the Same Device

The following describes the operation for executing multiple instructions of OUT, SET/RST, or PLS/PLF that use the same device in one scan.

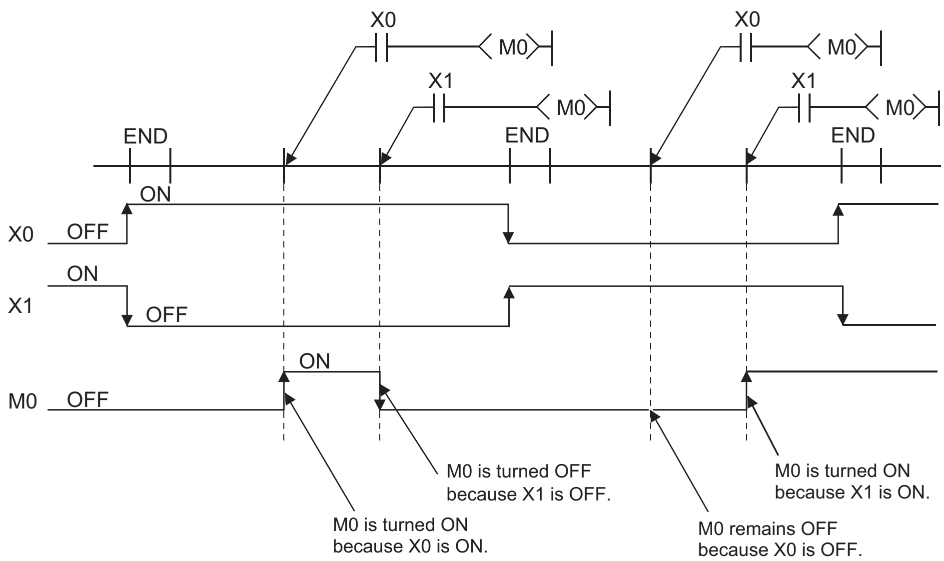
(1) OUT instructions using the same device

Do not program more than one OUT instruction using the same device in one scan.
If the OUT instructions using the same device are programmed in one scan, the specified device will turn ON or OFF every time the OUT instruction is executed, depending on the operation result of the program up to the relevant OUT instruction.
Since turning ON or OFF of the device is determined when each OUT instruction is executed, the device may turn ON and OFF repeatedly during one scan.
The following diagram shows an example of a circuit that turns the same internal relay (M0) with inputs X0 and X1 ON and OFF.

[Circuit]



[Timing Chart]



When specifying output (Y) in OUT instruction, the ON/OFF status of the device at the execution of the last OUT instruction in the scan is returned as the output (Y).

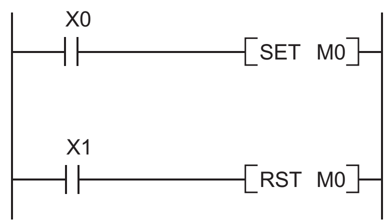
(2) SET/RST instructions using the same device

(a) The SET instruction turns ON the specified device when the execution command is ON and performs nothing when the execution command is OFF.
For this reason, when SET instructions using the same device are executed two or more times in one scan, the specified device will be ON if any one of the execution commands is ON.

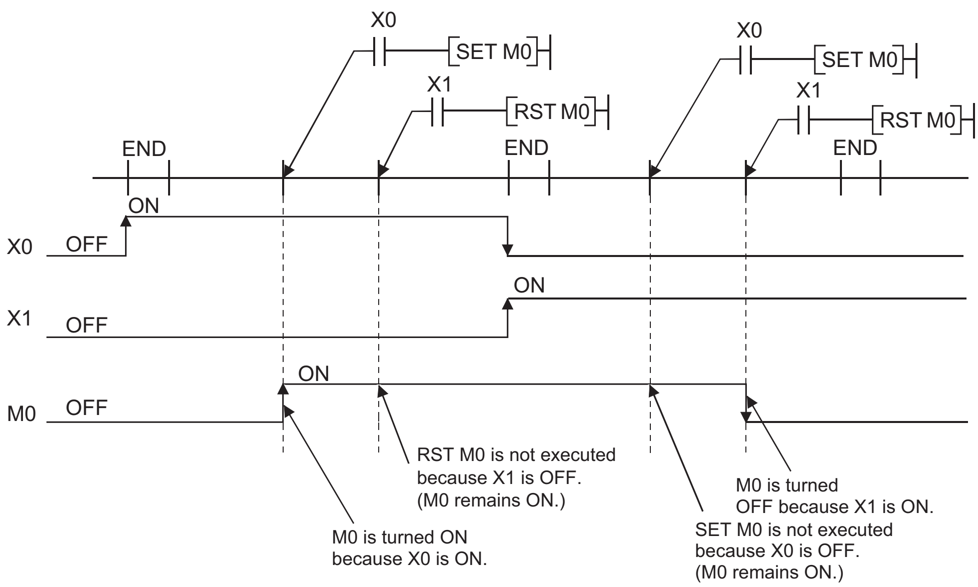
(b) The RST instruction turns OFF the specified device when the execution command is ON and performs nothing when the execution command is OFF.
For this reason, when RST instructions using the same device are executed two or more times in one scan, the specified device will be OFF if any one of the execution commands is ON.

(c) When the SET instruction and RST instruction using the same device are programmed in one scan, the SET instruction turns ON the specified device when the SET execution command is ON and the RST instruction turns OFF the specified device when the RST execution command is ON.
When both the SET and RST execution commands are OFF, the ON/OFF status of the specified device will not be changed.

[Circuit]



[Timing Chart]

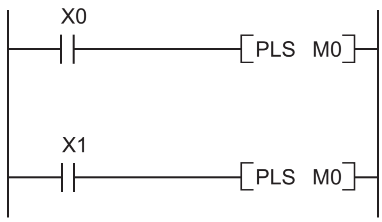


When specifying output (Y) in SET/RST instruction, the ON/OFF status of the device at the execution of the last instruction in the scan is returned as the output (Y).

(3) PLS instructions using the same device

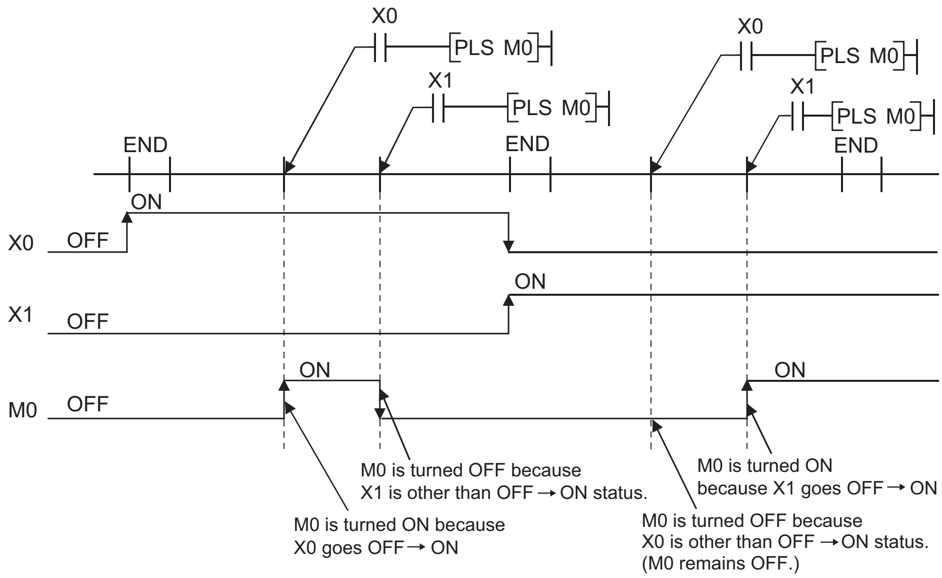
The PLS instruction turns ON the specified device when the execution command is turned ON from OFF.
It turns OFF the device at any other time (OFF to OFF, ON to ON, or ON to OFF).
If two or more PLS instructions using the same device are executed in one scan, each instruction turns ON the device when the corresponding execution command is turned ON from OFF and turns OFF the device in other cases.
For this reason, if multiple PLS instructions using the same device are executed in a single scan, a device that has been turned ON by a PLS instruction may not be turned ON during one scan.

[Circuit]

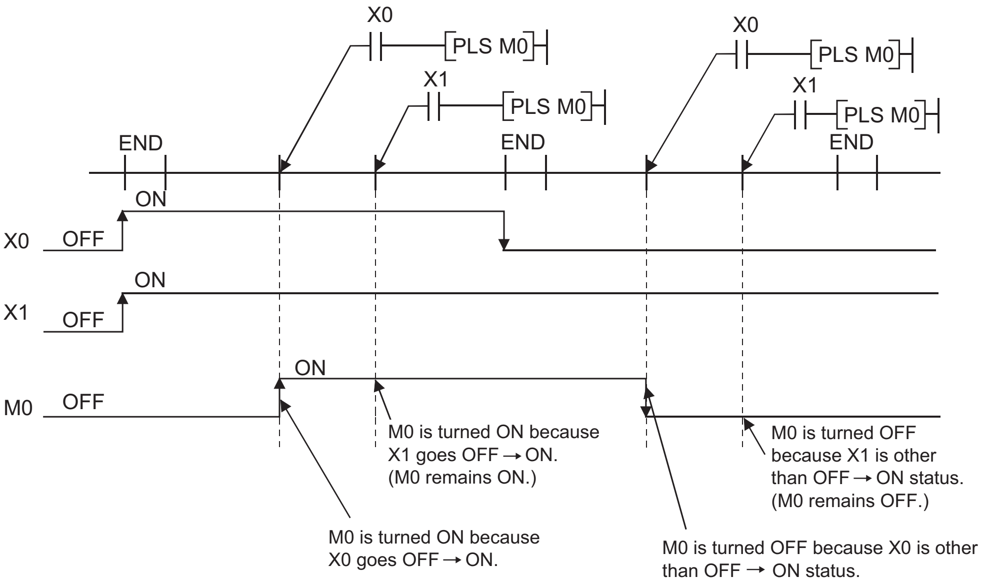


[Timing Chart]

• The ON/OFF timing of the X0 and X1 is different. (The specified device does not turn ON throughout the scan.)



• The X0 and X1 turn ON from OFF at the same time.

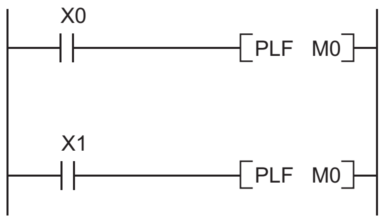


When specifying output (Y) in PLS instructions, the ON/OFF status of the device at the execution of the last PLS instruction in the scan is returned as the output (Y).

(4) PLF instructions using the same device

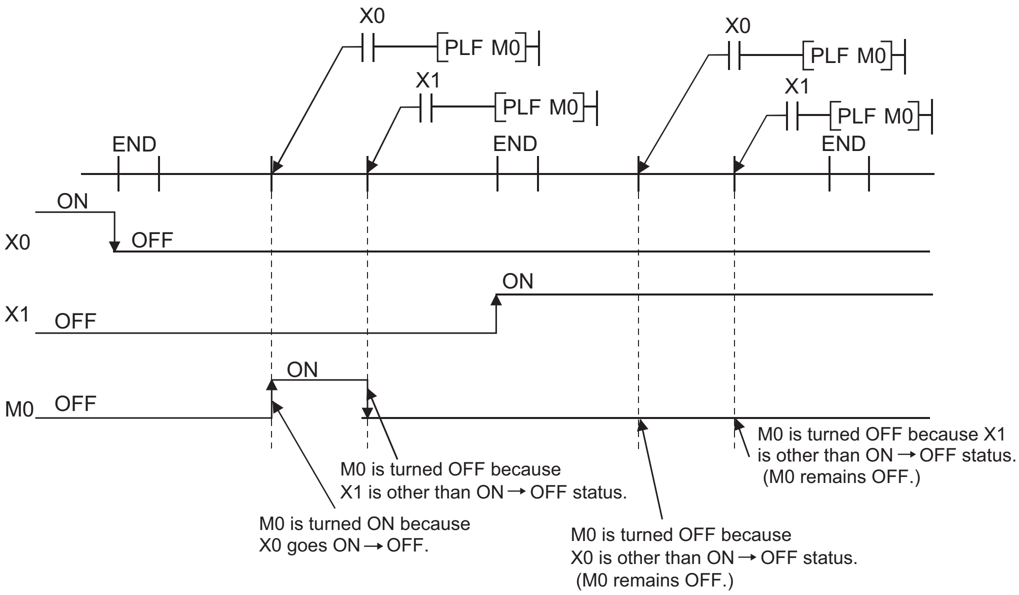
The PLF instruction turns ON the specified device when the execution command is turned OFF from ON.
It turns OFF the device at any other time (OFF to OFF, OFF to ON, or ON to ON).
If two or more PLF instructions using the same device are executed in one scan, each instruction turns ON the device when the corresponding execution command is turned OFF from ON and turns OFF the device in other cases.
For this reason, if multiple PLF instructions using the same device are executed in a single scan, a device that has been turned ON by a PLF instruction may not be turn ON during one scan.

[Circuit]

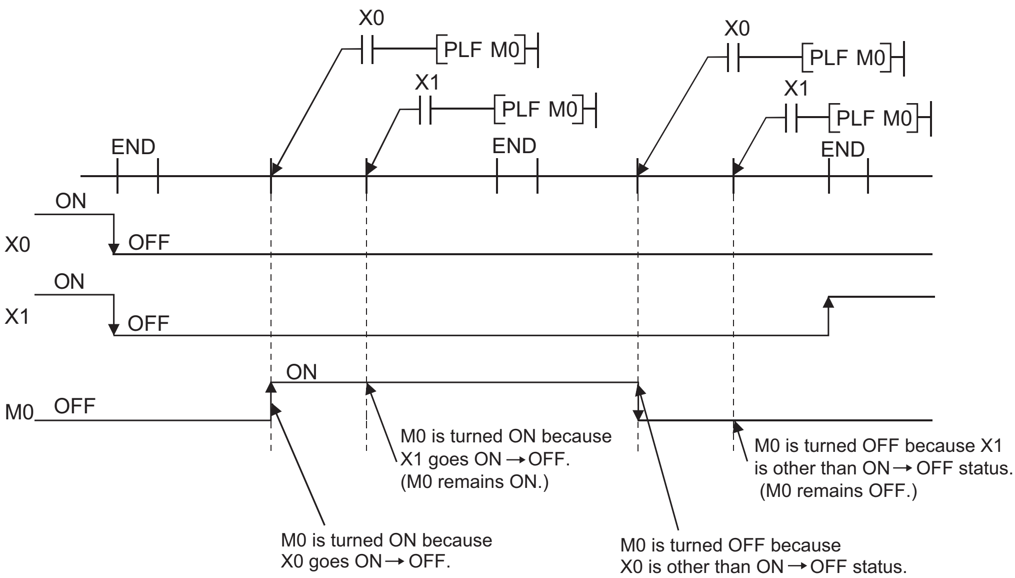


[Timing Chart]

- The ON/OFF timing of the X0 and X1 is different. (The specified device does not turn ON throughout the scan.)



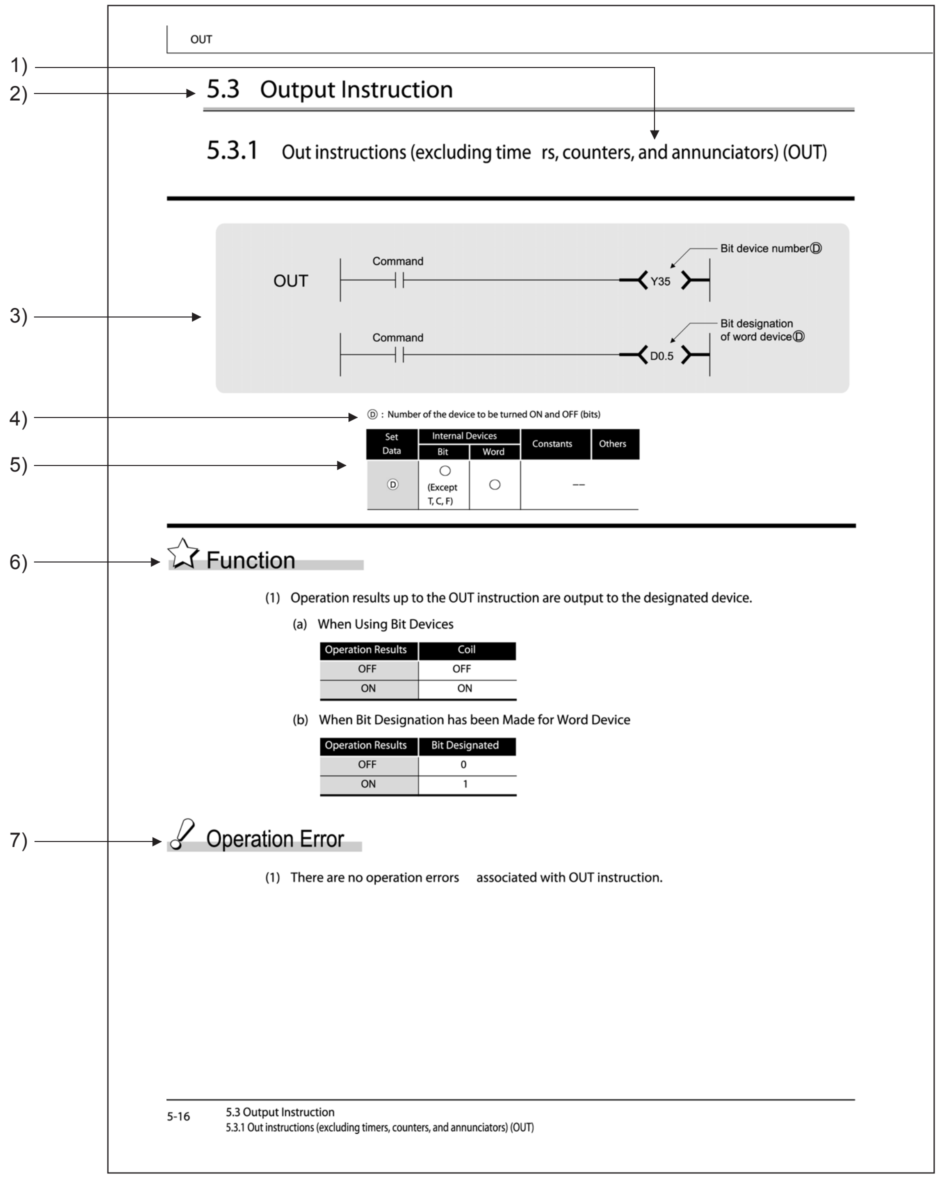
- The X0 and X1 turn OFF from ON at the same time.



When specifying output (Y) in PLF instructions, the ON/OFF status of the device at the execution of the last PLF instruction in the scan is returned as the output (Y).

4 HOW TO READ INSTRUCTIONS

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



1) Code used to write instruction (instruction symbol).

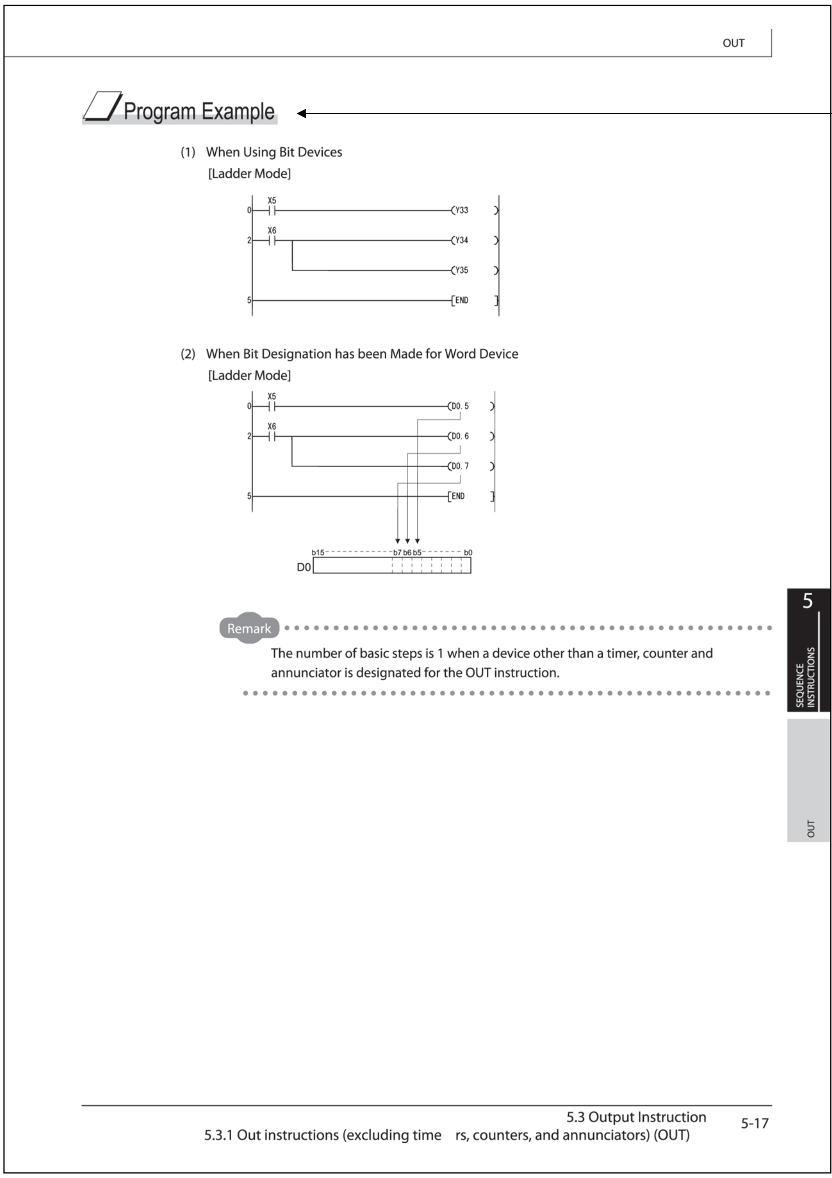
2) Section number and general category of instructions being discussed.

3) Indicates ladder mode expressions and execution conditions for instructions.

Execution Condition	Non-conditional Execution	Executed while ON	Executed One Time at ON	Executed One Time at OFF
Code recorded on description page	No symbol recorded			

4) Discusses the data set for each instruction and the data type.

Data Type	Meaning
Bit	Bit data or start number of bit device
BIN 16 bits	BIN 16-bit data or start number of word device
BIN 32 bits	BIN 32-bit data or start number of double word device
BCD 4-digit	4-digit BCD data
BCD 8-digit	8-digit BCD data


OUT

Program Example

(1) When Using Bit Devices
[Ladder Mode]

(2) When Bit Designation has been Made for Word Device
[Ladder Mode]

Remark

The number of basic steps is 1 when a device other than a timer, counter and annunciator is designated for the OUT instruction.

5 SEQUENCE INSTRUCTIONS

OUT

5.3 Output Instruction
5.3.1 Out instructions (excluding timers, counters, and annunciators) (OUT) 5-17

8)

5) Devices which can be used by the instruction in question are indicated with circle. The types of devices that can be used are as indicated below:

Device Type	Internal Devices (System, User)		Constant *3	Others *3
	Bit	Word		
Applicable devices *1	X, Y, M SM, F, B, SB,	T, ST, C, *2 D, W, SD, SW	K, H	N, V

*1: Refer to the manual below for the description for the individual devices.
- QSCPU User's Manual (Function Explanation, Program Fundamentals)

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

*3: Devices which can be set are described in the "Constant" and the "Others" columns.

6) Indicates the function of the instruction.

7) Indicates conditions under which error is returned, and error number.

8) Indicates simple program examples.
Also indicates the types of individual devices used when the program is executed.

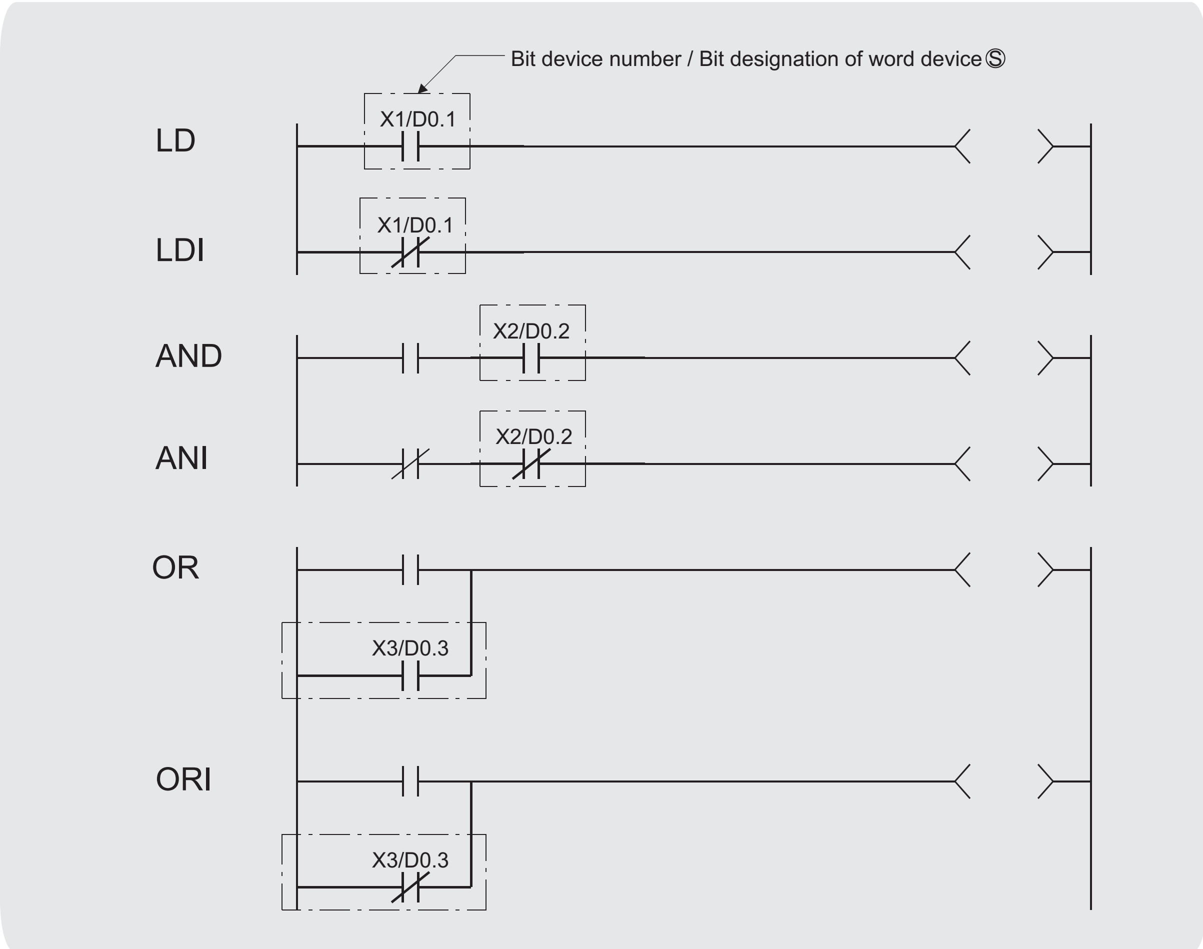
MEMO

5 SEQUENCE INSTRUCTIONS

Category	Processing Details	Reference section
Contact instruction	Operation start, series connection, parallel connection	5.1
Connection instructions	Ladder block connection, creation of pulses from operation results, store/read operation results	5.2
Output instruction	Bit device output, output reversal	5.3
Master control instruction	Master control	5.4
Termination instruction	Program termination	5.5
Other instructions	Instructions such as no operation which do not fit in the above categories	5.6

5.1 Contact Instruction

5.1.1 Operation start, series connection, parallel connection (LD, LDI, AND, ANI, OR, ORI)



Ⓢ : Devices used as contacts (bits)

<table>
<tr><th rowspan="2">Set Data</th><th colspan="2">Internal Devices</th><th rowspan="2">Constants</th><th rowspan="2">Others</th></tr>
<tr><th>Bit</th><th>Word</th></tr>
<tr><td>Ⓢ</td><td colspan="2">○</td><td colspan="2">—</td></tr>
</table>

Function

LD, LDI

(1) LD is the A contact operation start instruction, and LDI is the B contact operation start instruction. They read ON/OFF information from the designated device*1, and use that as an operation result.

*1: When a bit designation is made for a word device, the device turns ON or OFF depending on the 1/0 status of the designated bit.

AND, ANI

(1) AND is the A contact series connection instruction, and ANI is the B contact series connection instruction. They read the ON/OFF data of the designated bit device*2, perform an AND operation on that data and the operation result to that point, and take this value as the operation result.

*2: When a bit designation is made for a word device, the device turns ON or OFF depending on the 1/0 status of the designated bit.

(2) There are no restrictions on the use of AND or ANI, but the following applies to the ladder mode of the GX Developer:

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

OR, ORI

(1) OR is the A contact single parallel connection instruction, and ORI is the B contact single parallel connection instruction. They read ON/OFF information from the designated device*3, and perform an OR operation with the operation results to that point, and use the resulting value as the operation result.

*3: When a bit designation is made for a word device, the device turns ON or OFF depending on the 1/0 status of the designated bit.

(2) There are no restrictions on the use of OR or ORI, but the following applies to the ladder mode of the GX Developer

(a) Write.... OR and ORI can be used to create connections of up to 23 ladders.
(b) Read ... OR and ORI can be used to create connections of up to 23 ladders. The 24th or subsequent ladders cannot be displayed properly.

Remark

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

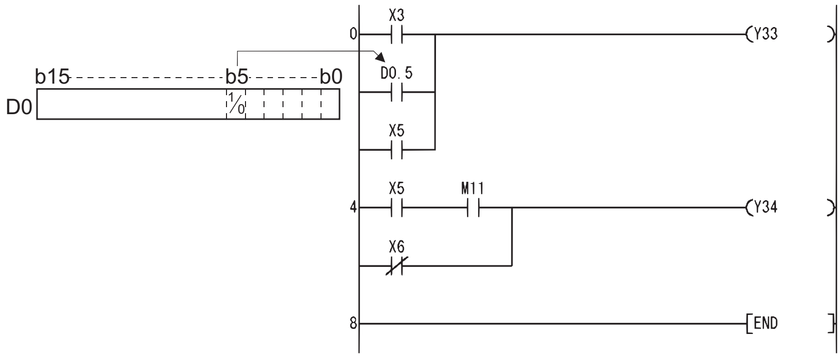
Operation Error

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

Program Example

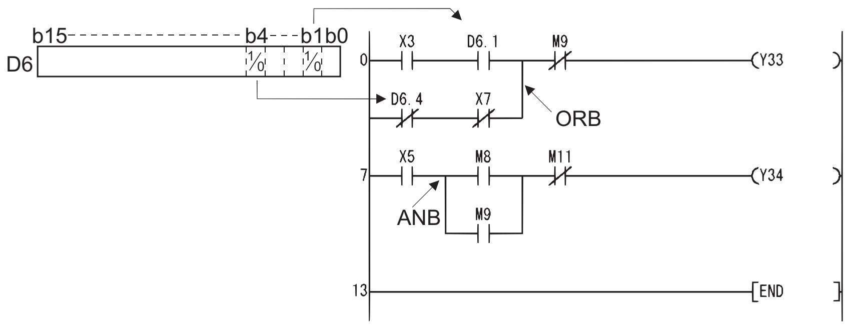
(1) A program using LD, AND, OR, and ORI instructions.

[Ladder Mode]



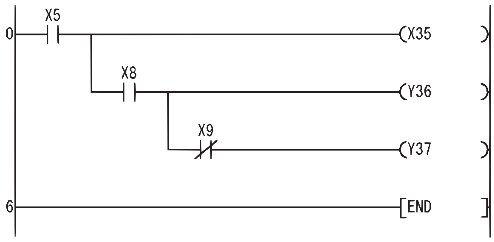
(2) A program linking contact points established through the use of ANB and ORB instructions.

[Ladder Mode]

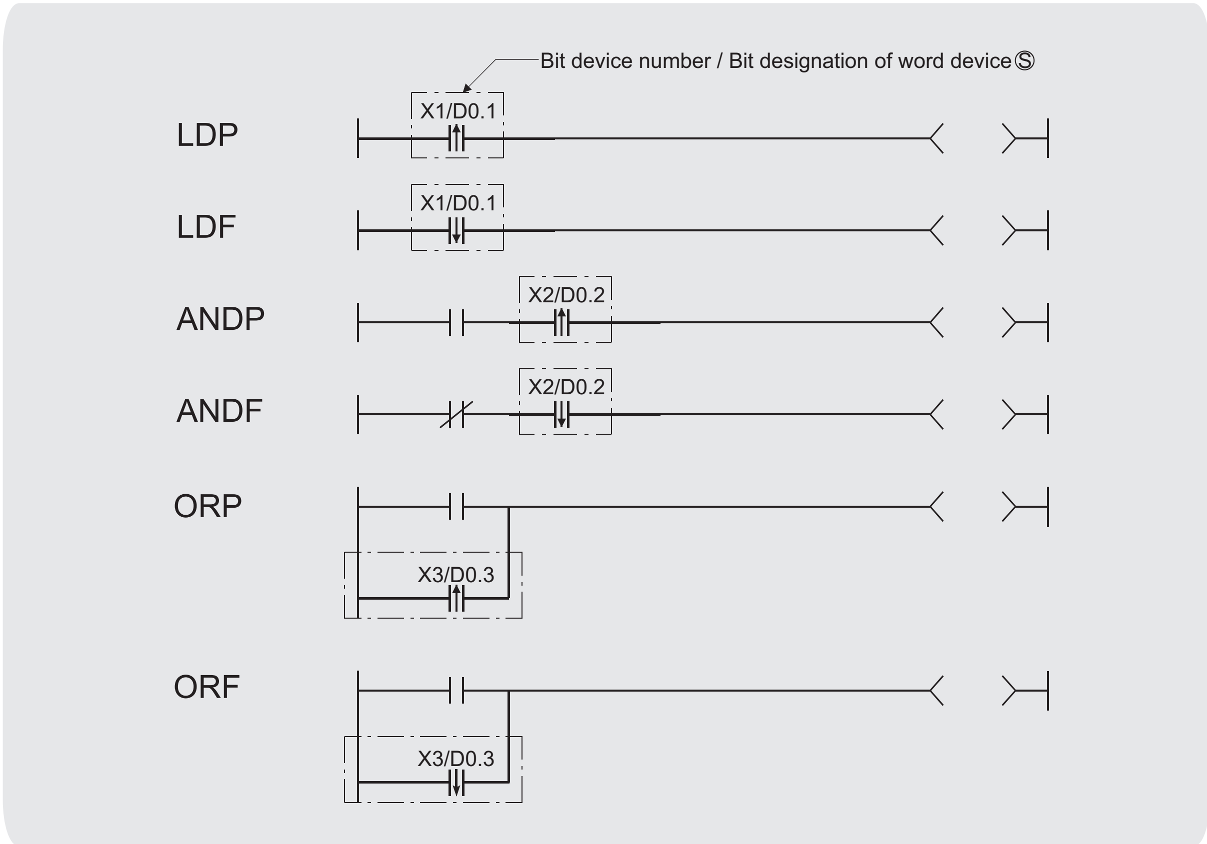


(3) A parallel program with OUT instruction.

[Ladder Mode]



5.1.2 Pulse operation start, pulse series connection, pulse parallel connection (LDP, LDF, ANDP, ANDF, ORP, ORF)



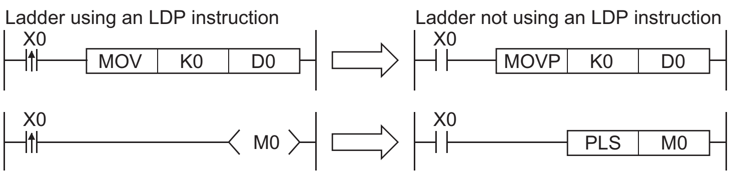
Ⓢ : Devices used as contacts (bits)

Set Data	Internal Devices		Constants	Others
	Bit	Word		
Ⓢ	○		—	

Function

LDP, LDF

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



(2) LDF is the trailing edge pulse operation start instruction, and is ON only at the trailing edge of the designated bit device (when it goes from ON to OFF).

If a word device has been designated, it is ON only when the designated bit changes from 1 to 0.

ANDP, ANDF

(1) ANDP is a leading edge pulse series connection instruction, and ANDF is a trailing edge pulse series connection instruction. They perform an AND operation with the operation result to that point, and take the resulting value as the operation result.

The ON/OFF data used by ANDP and ANDF are indicated in the table below:

Device specified in ANDP or ANDF		ANDP State	ANDF State
Bit device	Bit designated for word device		
OFF to ON	0 to 1	ON	OFF
OFF	0	OFF	
ON	1		
ON to OFF	1 to 0		ON

ORP, ORF

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

The ON/OFF data used by ORP and ORF are indicated in the table below:

Device designated in ORP or ORF		ORP State	ORF State
Bit device	Bit designated for word device		
OFF to ON	0 to 1	ON	OFF
OFF	0	OFF	
ON	1		
ON to OFF	1 to 0		ON

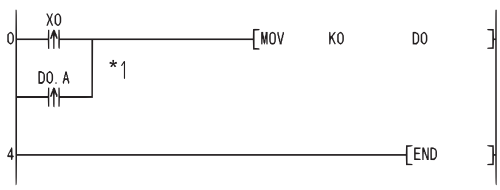
Operation Error

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

Program Example

(1) The following program executes the MOV instruction at input X0, or at the leading edge of b10 (bit 11) of data register D0:

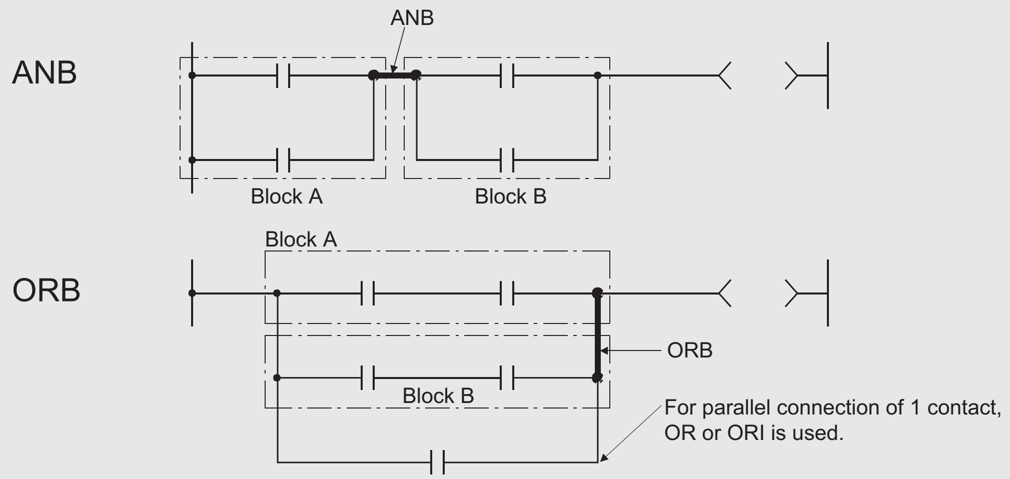
[Ladder Mode]



*1: Word device bit designations are performed in hexadecimal.
Bit b10 of D0 would be D0.A.

5.2 Connection Instructions

5.2.1 Ladder block series connections and parallel connections (ANB, ORB)



Set Data	Internal Devices		Constants	Others
	Bit	Word		
—	—			

☆ Function

ANB

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

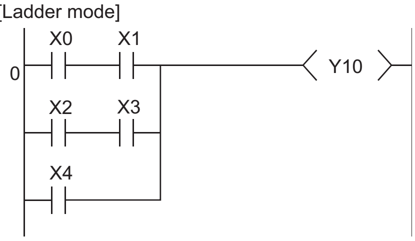
(2) The symbol for ANB is not the contact symbol, but rather is the connection symbol.

ORB

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

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

[Ladder mode]



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

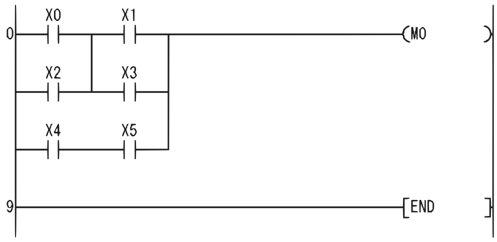
Operation Error

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

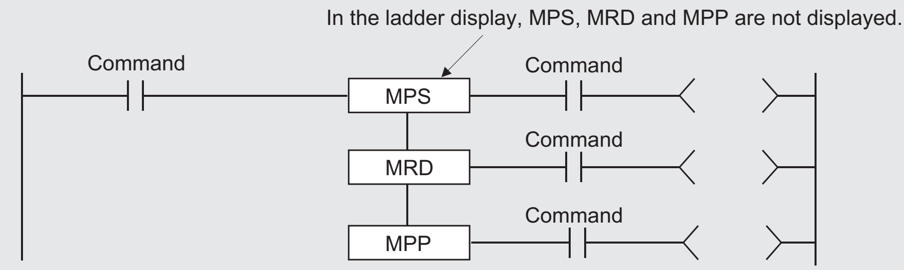
Program Example

(1) A program using ANB and ORB instructions.

[Ladder Mode]



5.2.2 Operation results push, read, pop (MPS, MRD, MPP)



Set Data	Internal Devices		Constants	Others
	Bit	Word		
—	—			

Function

MPS

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

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

MRD

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

MPP

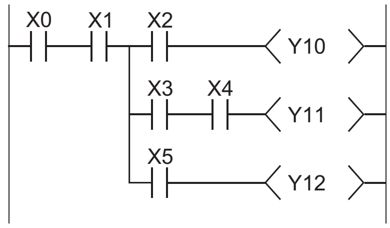
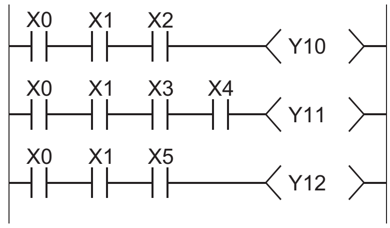
(1) Reads the operation result stored for the MPS instruction, and uses that result to perform the operation in the next step.

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

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

POINT

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

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

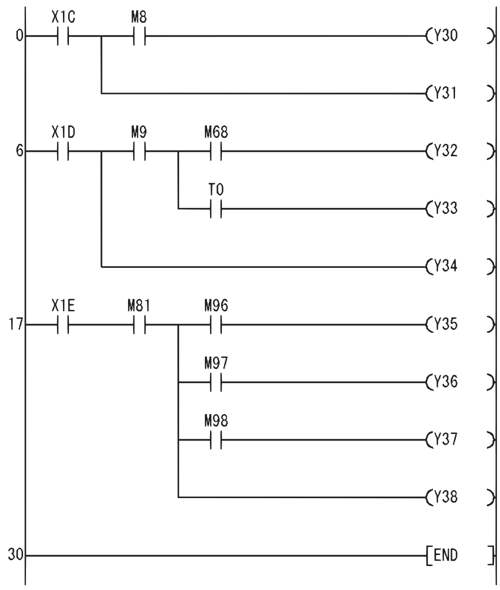
2. The number of times the MPS and MPP instructions are used must be the same.
   If not, correct ladder display is not possible in the ladder mode of the GX Developer.

Operation Error

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

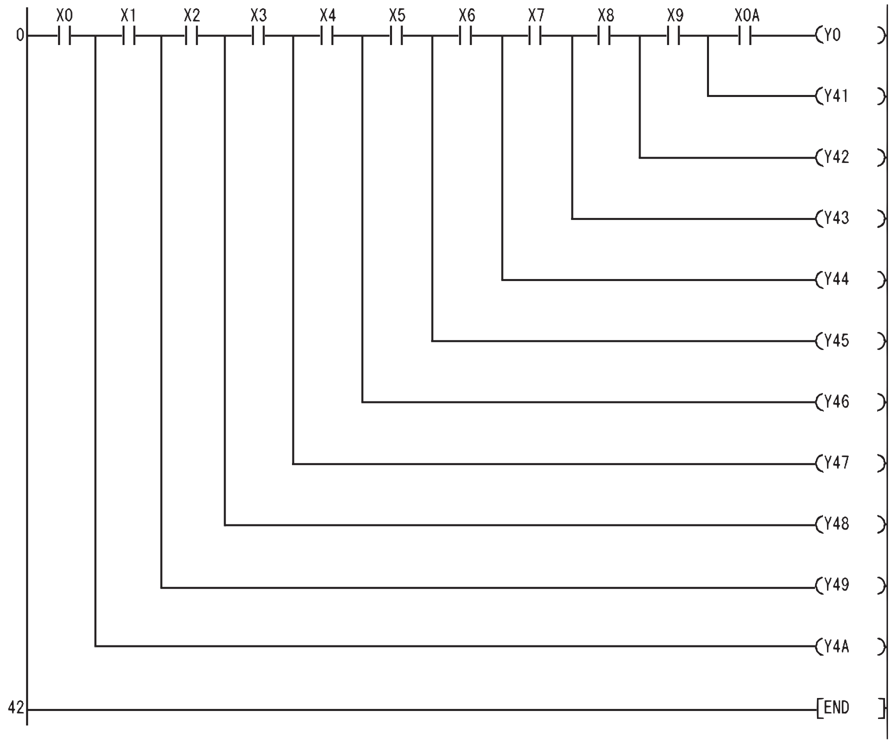
Program Example

(1) A program using the MPS, MRD, and MPP instructions.
[Ladder Mode]

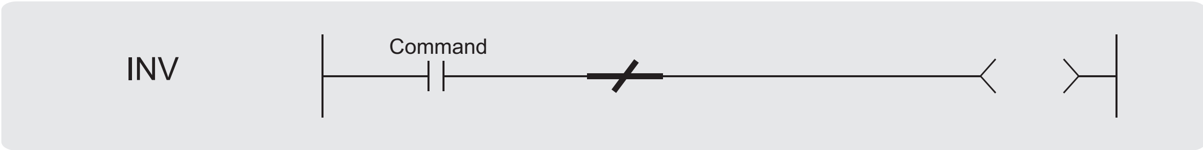


(2) A program using MPS and MPP instructions successively.

[Ladder Mode]



5.2.3 Operation results inversion (INV)



Set Data	Internal Devices		Constants	Others
	Bit	Word		
—	—			

Function

(1) Inverts the operation result immediately prior to the INV instruction.

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

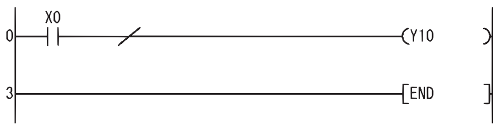
Operation Error

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

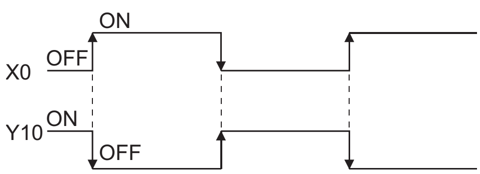
Program Example

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

[Ladder Mode]

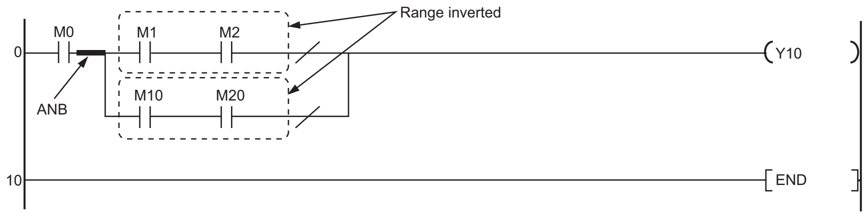


[Timing Chart]



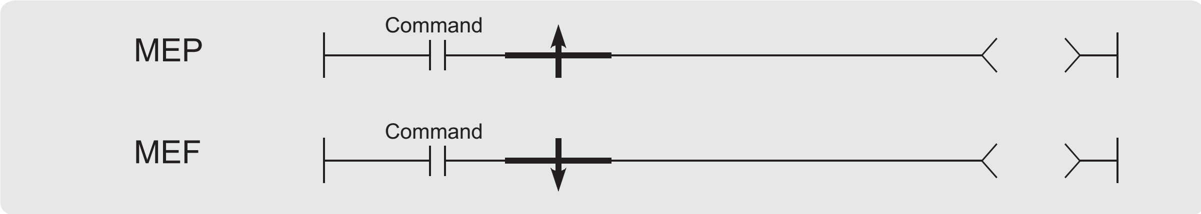
⊠POINT

1. The INV instruction operates based on the results of calculation made until the INV instruction is given. Accordingly, use it in the same position as that of the AND instruction.
   The INV instruction cannot be used at the LD and OR positions.
2. When a ladder block is used, the operation result is inverted within the range of the ladder block. To operate a ladder using the INV instruction in combination with the ANB instruction, pay attention to the range that will be inverted.



For details of the ANB instruction, refer to Section 5.2.1.

5.2.4 Operation result pulse conversion (MEP, MEF)



Set Data	Internal Devices		Constants	Others
	Bit	Word		
—	—			

Function

MEP

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

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

MEF

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

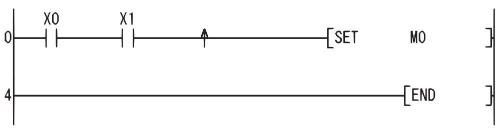
(2) Use of the MEF instruction simplifies pulse conversion processing when multiple contacts are connected in series.

Operation Error

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

Program Example

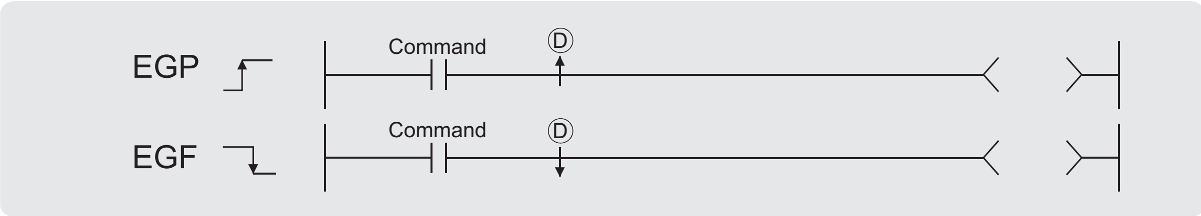
(1) A program which performs pulse conversion on the operation results of X0 and X1:
[Ladder Mode]



POINT

Because the MEP and MEF instructions operate according to the operation result immediately before the execution of these instructions, they must be used at the same position as the AND instruction. The MEP and MEF instructions cannot be used at the position of LD or OR instruction.

5.2.5 Pulse conversion of edge relay operation results (EGP, EGF)



D : Edge relay number where operation results are stored (bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants	Others V
D	—			○

Function

EGP

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

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

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

EGF

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

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

(3) The EGF instruction can be used like an AND instruction.

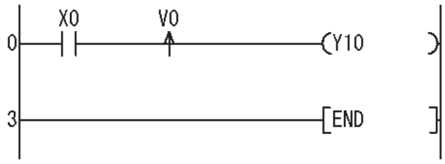
Operation Error

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

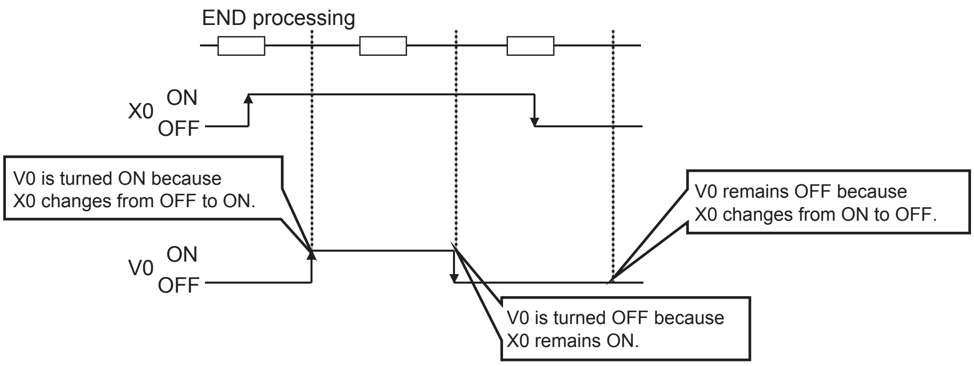
Program Example

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

[Ladder Mode]

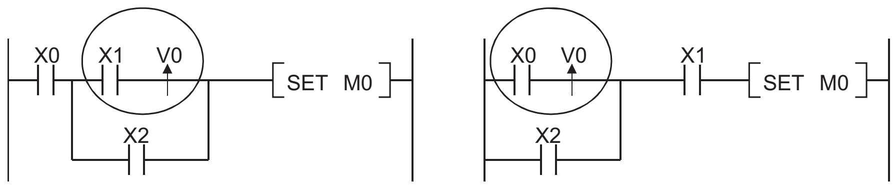


[Operation]



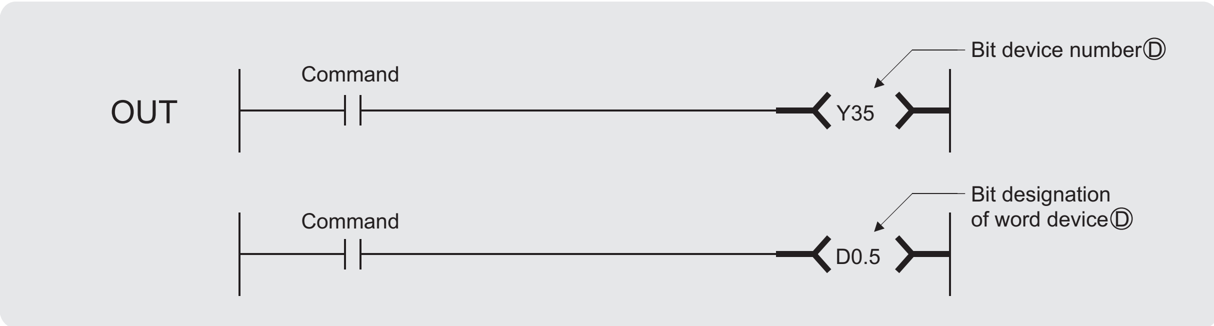
POINT

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



5.3 Output Instruction

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



D : Number of the device to be turned ON and OFF (bits)

Set Data	Internal Devices		Constants	Others
	Bit	Word		
D	○ (Except T, C, F)	○	—	

Function

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

(a) When Using Bit Devices

Operation Results	Coil
OFF	OFF
ON	ON

(b) When Bit Designation has been Made for Word Device

Operation Results	Bit Designated
OFF	0
ON	1

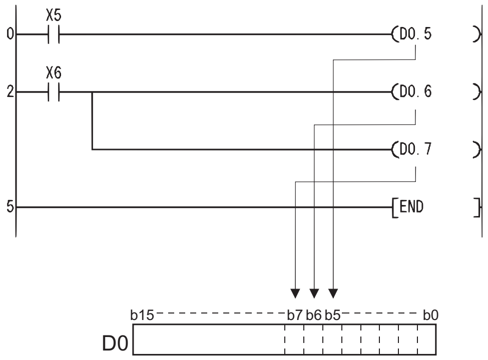
Operation Error

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

Program Example

(1) When Using Bit Devices
[Ladder Mode]

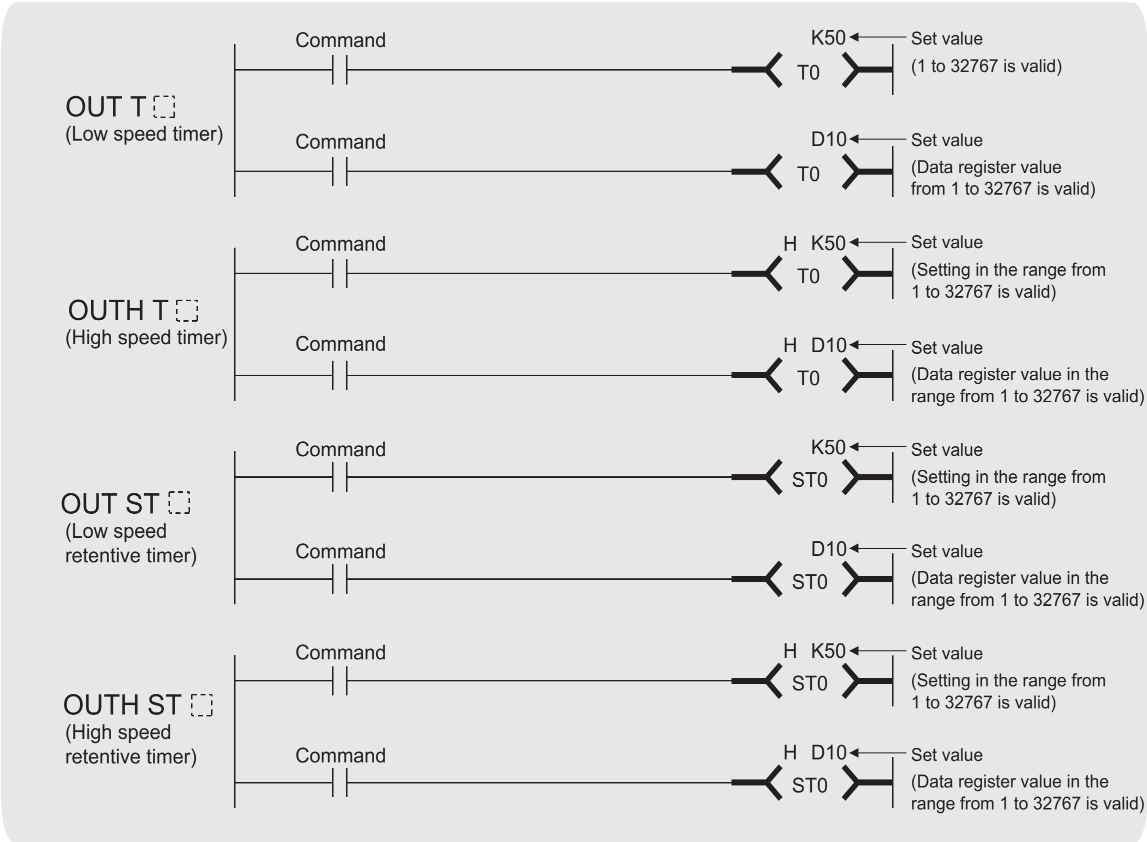
(2) When Bit Designation has been Made for Word Device
[Ladder Mode]



Remark

The number of basic steps is 1 when a device other than a timer, counter and annunciator is designated for the OUT instruction.

5.3.2 Timers (OUT T,OUTH T)



D : Timer number (bit)
Set value : Value set for timer (BIN 16 bits)

Set Data	Internal Devices		Constants K	Others
	Bit	Word		
D	○ (Only T)	—	—	—
Set value	—	○ (Except T, C)	○*1	—

*1: Timer values can be set only in a decimal constant (K). A hexadecimal constant (H) cannot be used for timer settings.

☆ Function

(1) When the operation results up to the OUT instruction are ON, the timer coil goes ON and the timer starts measurement; at the time-up (measured value ≧ set value), the contact is as follows:

A Contact	Continuity
B Contact	Non-continuity

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

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

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

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

(5) The following processing is conducted when the OUT instruction is executed:

- OUT T□ coil turned ON or OFF
- OUT T□ contact turned ON or OFF
- OUT T□ present value updated

If the same OUT T□ instruction is executed twice or more times during the same scan, the present value is updated by the number of times the instruction is executed.

Remark

1. Timer's time limit
   Time limit of the timer is set in the PLC system setting of the PLC parameter dialog box.

Type of Timer	QSCPU	
	Setting Range	Setting unit
Low speed timer Low speed retentive timer	1 ms to 1000 ms (Default: 100 ms)	1 ms
High speed timer High speed retentive timer	0.1 ms to 100 ms (Default: 10 ms)	0.1 ms

2. Refer to the manual below for information on timer counting methods.
   - QSCPU User's Manual (Function Explanation, Program Fundamentals)
3. The number of basic steps of the OUT T□ instruction is 4.

Operation Error

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

Caution

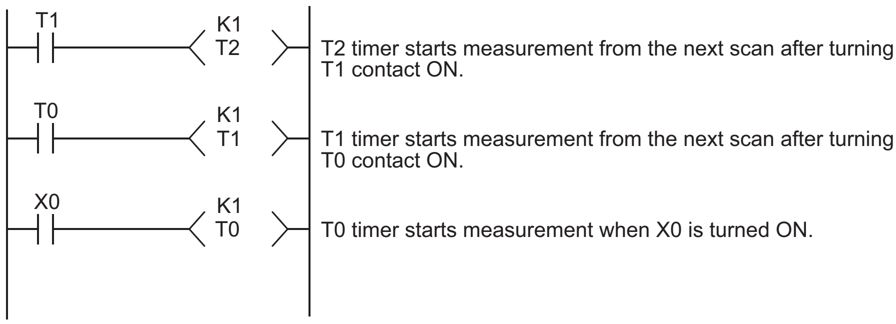
(1) When creating a program in which the operation of the timer contact triggers the operation of other timer, create the program according to the operation order of the timers - create the program for the timer that operates later first.

In the following cases, all timers go ON at the same scan if the program is created in the order the timers operate.

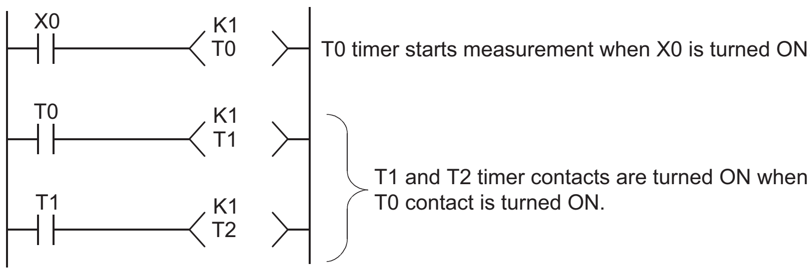
- If the set value is smaller than a scan time.
- If "1" is set.

Example

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



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



Program Example

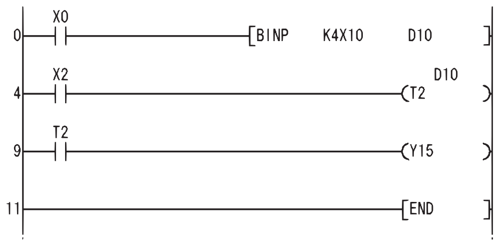
(1) The following program turns Y10 and Y14 ON 10 seconds after X0 has gone ON.

[Ladder Mode]

*2: The set value of the low-speed timer indicates its default time limit (100 ms).

(2) The following program uses the BCD data at X10 to X1F as the timer's set value.

[Ladder Mode]



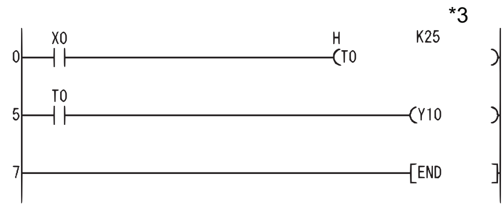
Converts BCD data at X10 to X1F to BIN and stores the converted value in D10.

When X2 is turned ON, starts counting using the data stored in D10 as the set value.

Y15 turns ON at the count up of T2.

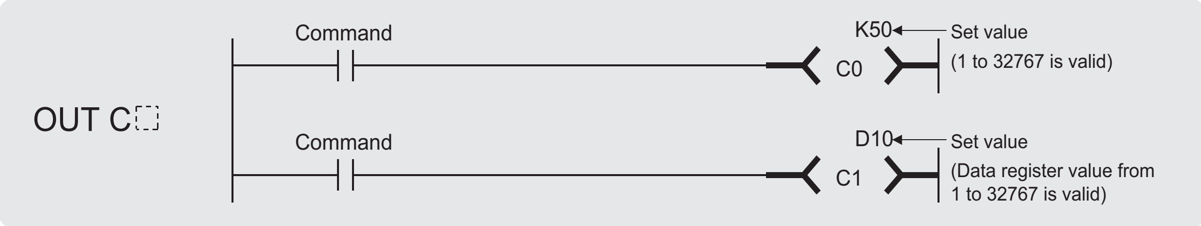
(3) The following program turns Y10 ON 250 m after X0 goes ON.

[Ladder Mode]



*3: The set value of the high speed timer indicates its default time limit (10 ms).

5.3.3 Counters (OUT C)



Ⓓ: Counter number (bits)
Set value: Counter set value (BIN 16 bits)

Set Data	Internal Devices		Constants K	Others
	Bit	Word		
Ⓓ	○ (Only C)	—	—	—
Set value	—	○ (Except T, C)	○*1	—

*1: Counter values can be set only in a decimal constant (K). A hexadecimal constant (H) cannot be used for the counter value setting.

Function

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

A Contact	Continuity
B Contact	Non-continuity

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

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

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

Remark

1. Refer to the manual below for counter counting methods.
   • QSCPU User's Manual (Function Explanation, Program Fundamentals)
2. The number of basic steps of the OUT C□ instruction is 4.

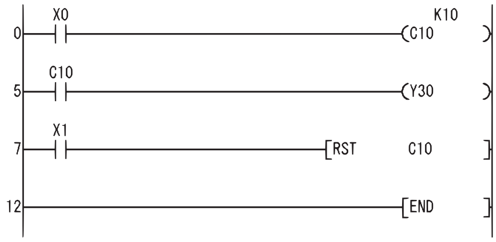
Operation Error

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

Program Example

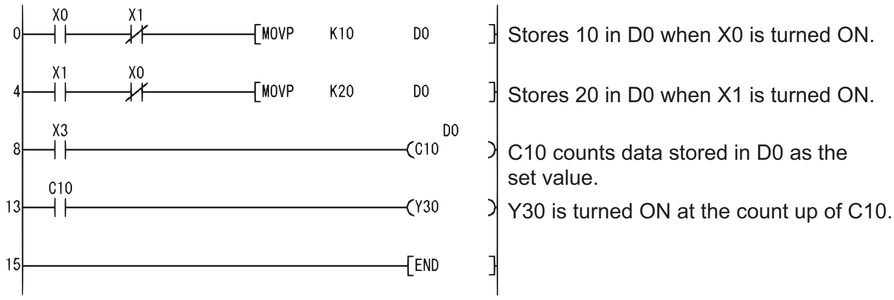
(1) The following program turns Y30 ON after X0 has gone ON 10 times, and resets the counter when X1 goes ON.

[Ladder Mode]



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

[Ladder Mode]



5.3.4 Annunciator output (OUT F)

OUT F□

Ⓓ : Number of the annunciator to be turned ON (bits)

Set Data	Internal Devices		Constants	Others
	Bit	Word		
Ⓓ	○ (Only F)	—		

Function

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

(2) The following responses occur when an annunciator (F) is turned ON.
- The "USER" LED goes ON.
- The annunciator numbers which are ON (F numbers) are stored in special registers (SD64 to SD79).
- The value of SD63 is incremented by 1.

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

(4) When the annunciator is turned OFF by the OUT instruction, although the coil goes OFF, status of the "USER" LED and the contents of SD63 to SD79 are not changed.
To turn OFF the "USER" LED or to delete the annunciator, which was turned OFF by the OUT F□ instruction from SD63 to SD79, use the RST F□ instruction.

Operation Error

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

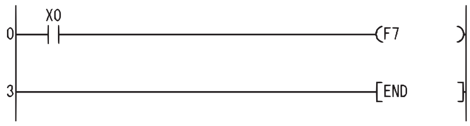
Remark

1. Refer to the manual below for details of annunciators.
   - QSCPU User's Manual (Function Explanation, Program Fundamentals)
2. The number of basic steps for the OUT module F□ instruction is 2.

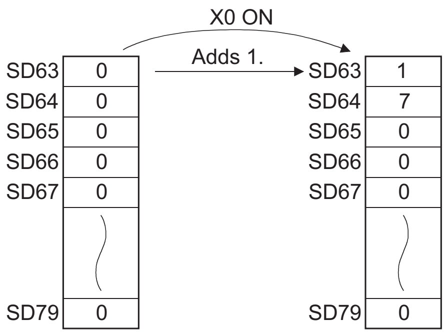
Program Example

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

[Ladder Mode]



[Operation]



5.3.5 Setting devices (except for annunciators) (SET)

D : Bit device number to be set (ON)/Word device bit designation (bits)

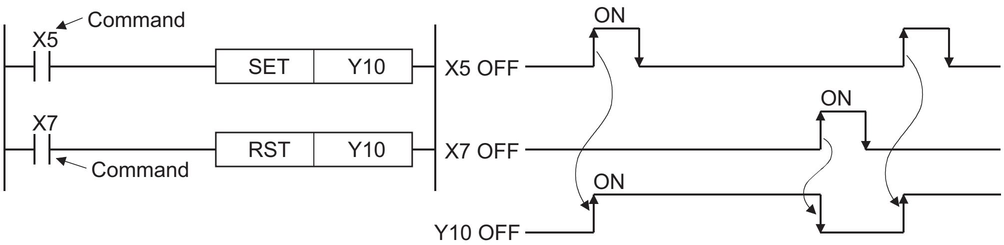
Set Data	Internal Devices		Constants	Others
	Bit	Word		
D	○	○ (Except T, C)	—	○

Function

(1) When the execution command is turned ON, the status of the designated devices becomes as shown below:

Device	Device Status
Bit device	Coils and contacts turned ON
When Bit Designation has been Made for Word Device	Designation bit set at 1

(2) Devices turned ON by the instruction remain ON when the same command is turned OFF. Devices turned ON by the SET instruction can be turned OFF by the RST instruction.



(3) When the execution command is OFF, the status of devices does not change.

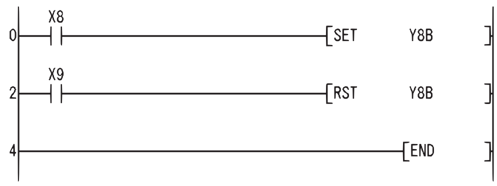
Operation Error

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

Program Example

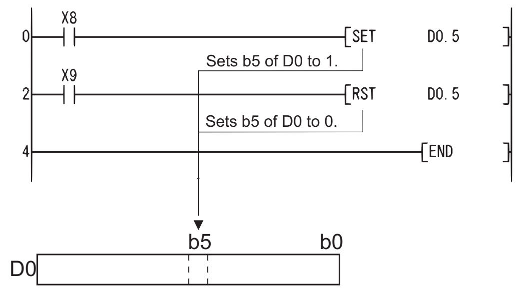
(1) The following program sets Y8B (ON) when X8 goes ON, and resets Y8B (OFF) when X9 goes ON.

[Ladder Mode]



(2) The following program sets the value of D0 bit 5 (b5) to 1 when X8 goes ON, and set the bit value to 0 when X9 goes ON.

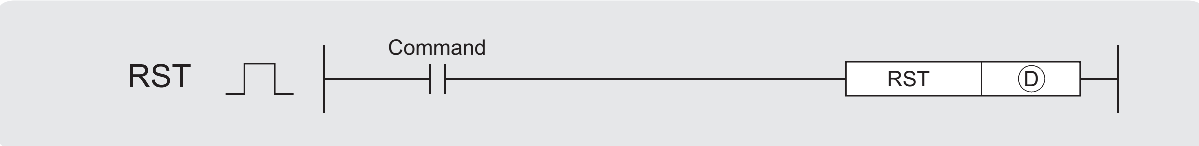
[Ladder Mode]



Remark

1. The number of basic steps is 1 when a device other than an annunciator is designated for the SET instruction.
2. When using X as a device, use the device numbers that are not used for the actual input. If the same number is used for the actual input device and input X, the data of the actual input will be written over the input X specified in the SET instruction.

5.3.6 Resetting devices (except for annunciators) (RST)



Ⓓ : Bit device number to be reset/ Word device bit designation (bits)
Word device number to be reset (BIN 16 bits)

Set Data	Internal Devices		Constants	Others
	Bit	Word		
Ⓓ	○		—	

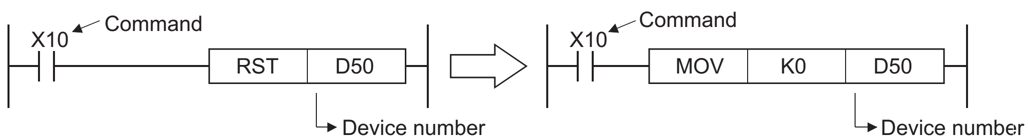
Function

(1) When the execution command is turned ON, the status of the designated devices becomes as shown below:

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

(2) When the execution command is OFF, the status of devices does not change.

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



Operation Error

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

Remark

The basic number of steps of the RST instruction is as follows.

a) For bit processing

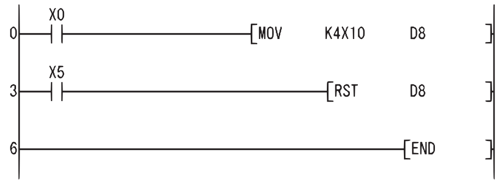
- Internal device (bit to be specified by bit device or word device): 1
- Timer, counter : 4

b) The number of basic steps is 2 for word processing.

Program Example

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

[Ladder Mode]

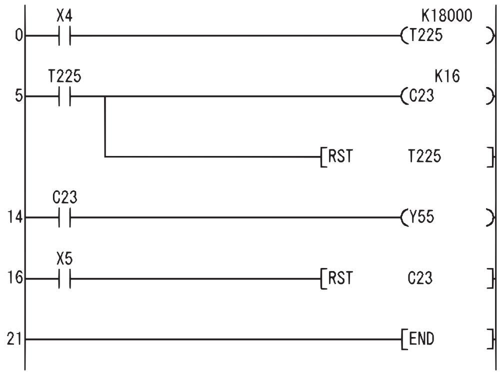


Stores the values of X10 to X1F in D8 when X0 is turned ON.

Sets the value of D8 to 0 when X5 is turned ON.

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

[Ladder Mode]



When T225 is set as retentive timer, it is turned ON when X4 ON time reaches 30 min.

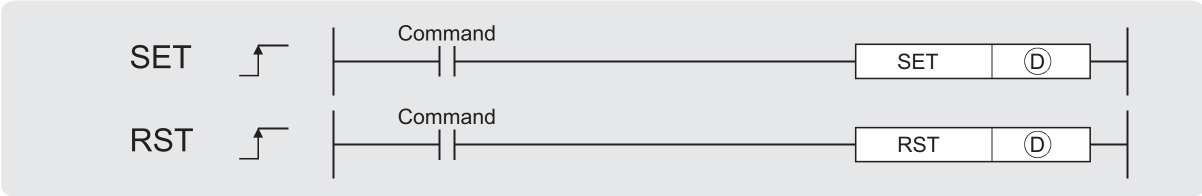
Counts the number of times T225 was turned ON.

Resets the coil, contact and present value of T225 when the contact of T225 is turned ON.

Y55 is turned ON at the count up of C23.

Resets C23 when X5 is turned ON.

5.3.7 Setting and resetting the annunciators (SET F, RST F)



SET Ⓓ : Number of the annunciator to be set (F number) (bits)

RST Ⓓ : Number of the annunciator to be reset (F number) (bits)

Set Data	Internal Devices		Constants	Others
	Bit	Word		
Ⓓ	○ (Only F)	—		

Function

SET

(1) The annunciator designated by Ⓓ is turned ON when the execution command is turned ON.

(2) The following responses occur when an annunciator (F) is turned ON.
- The "USER" LED goes ON.
- The annunciator numbers which are ON (F numbers) are stored in special registers (SD64 to SD79).
- The value of SD63 is incremented by 1.

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

RST

(1) The annunciator designated by Ⓓ is turned OFF when the execution command is turned ON.

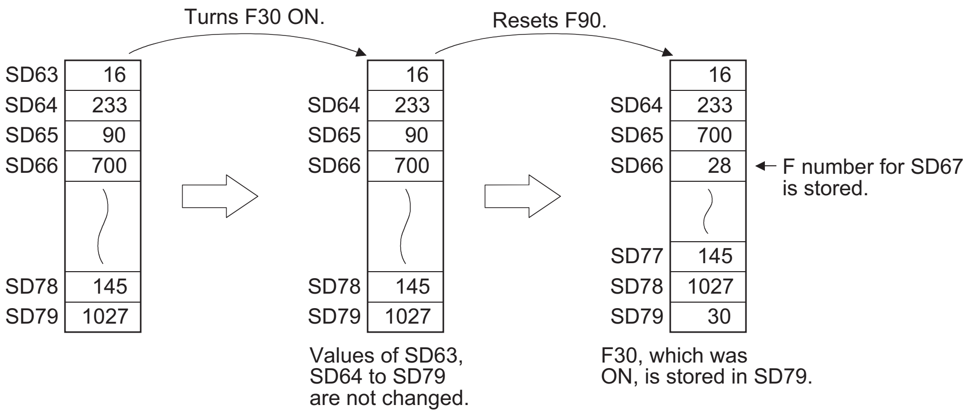
(2) The annunciator numbers (F numbers) of annunciators that have gone OFF are deleted from the special registers (SD64 to SD79), and the value of SD63 is decremented by 1.

Remark

1. Refer to the manual below for details of annunciators.
   - QSCPU User's Manual (Function Explanation, Program Fundamentals)
2. The number of basic steps for the SET F□ and RST F□ instructions is 2.

(3) When the value of SD63 is "16", the annunciator numbers are deleted from SD64 to SD79 by the use of the RST instruction. If the annunciators whose numbers are not registered in SD64 to SD79 are ON, these numbers will be registered.
If all annunciator numbers from SD64 to SD79 are turned OFF, the "USER" LED on the front of the safety CPU module will be turned OFF.

[Operations which take place when SD63 is 16]

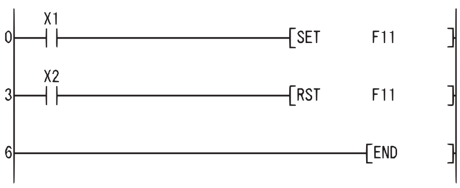


Operation Error

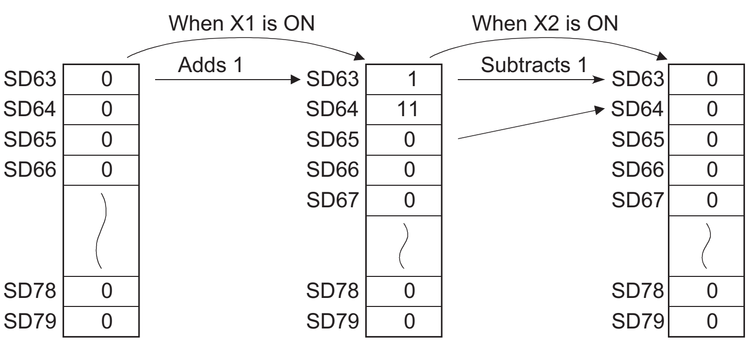
(1) There are no operation errors associated with the SET F□ or RST F□ instructions.

Program Example

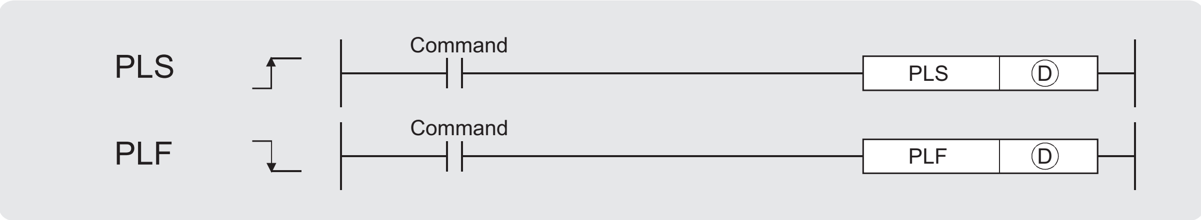
(1) The following program turns annunciator F11 ON when X1 goes ON, and stores the value 11 at the special register (SD64 to SD79). Further, the program resets annunciator F11 if X2 goes ON, and deletes the value 11 from the special registers (SD64 to SD79).
[Ladder Mode]



[Operation]



5.3.8 Leading edge and trailing edge output (PLS, PLF)



Ⓓ : Pulse conversion device (bits)

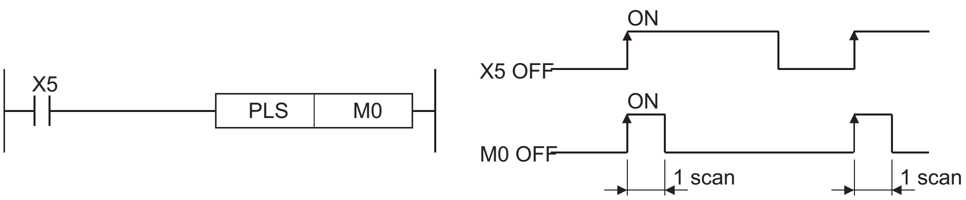
Set Data	Internal Devices Bit	Internal Devices Word	Constants	Others
Ⓓ	○		—	

Function

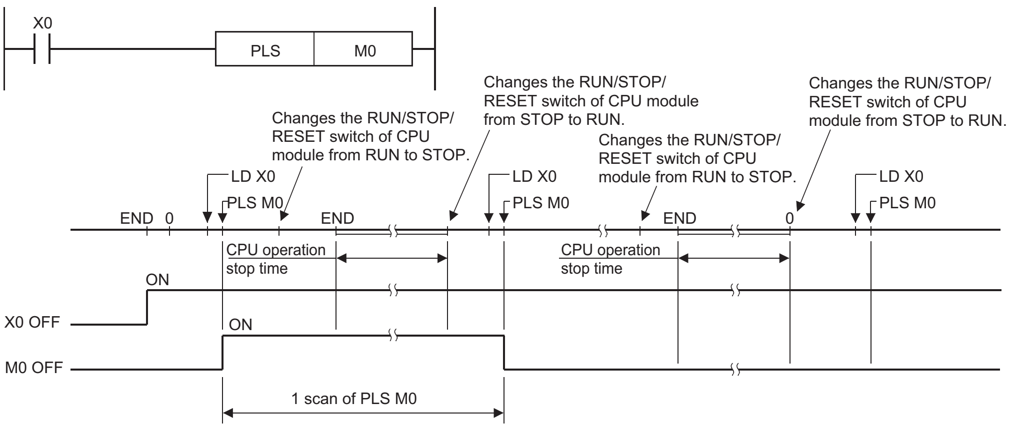
PLS

(1) Turns ON the designated device when the execution command is turned OFF → ON, and turns OFF the device in any other case the execution command is turned OFF → ON (i.e., at ON → ON, ON → OFF or OFF → OFF of the execution command).

When there is one PLS instruction for the device designated by Ⓓ during one scan, the specified device turns ON one scan.
See 3.7 for the operation to be performed when the PLS instruction for the same device is executed more than once during one scan.



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



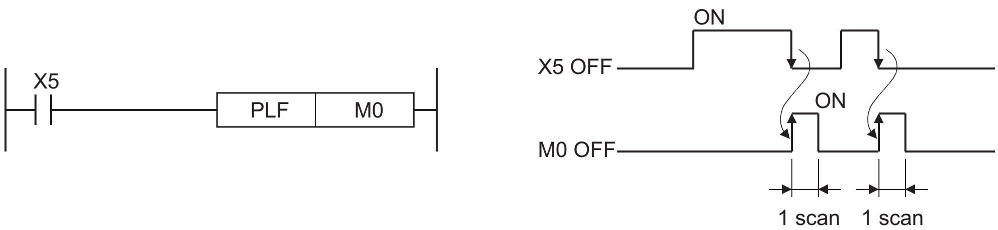
(3) When designating a latch relay (L) for the execution command and turning the power supply OFF to ON with the latch relay ON, the execution command turns OFF to ON at the first scan, executing the PLS instruction and turning ON the designated device.

The device turned ON at the first scan after power-ON turns OFF at the next PLS instruction.

PLF

(1) Turns ON the designated device when the execution command is turned ON → OFF, and turns OFF the device in any other case the execution command is turned ON → OFF (i.e., at OFF → OFF, OFF → ON or ON → ON of the execution command).

When there is one PLF instruction for the device designated by Ⓓ during one scan, the specified device turns ON one scan.
See 3.7 for the operation to be performed when the PLF instruction for the same device is executed more than once during one scan.



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

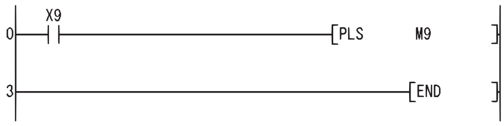
Operation Error

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

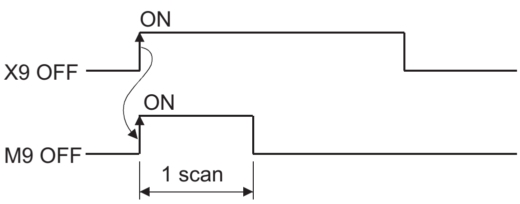
Program Example

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

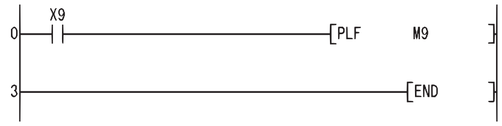
[Ladder Mode]



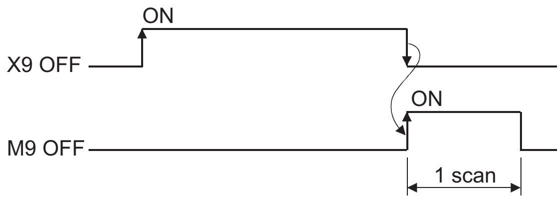
[Timing Chart]



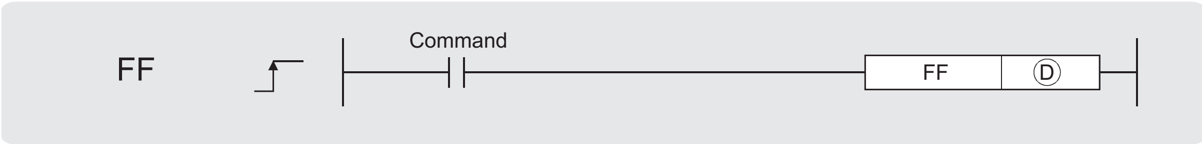
(2) The following program executes the PLF instruction when X9 goes OFF.
[Ladder Mode]



[Timing Chart]



5.3.9 Bit device output reverse (FF)



Ⓓ : Device number of the device to be reversed (bits)

Set Data	Internal Devices		Constants	Others
	Bit	Word		
Ⓓ	○		—	

Function

(1) Reverses the output status of the device designated by Ⓓ when the execution command is turned OFF → ON.

Device	Device Status	
	Prior to FF execution	After FF execution
Bit device	OFF	ON
	ON	OFF
Bit designated for word device	0	1
	1	0

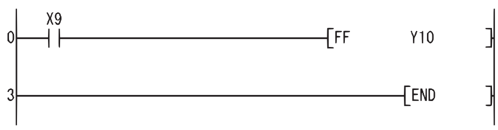
Operation Error

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

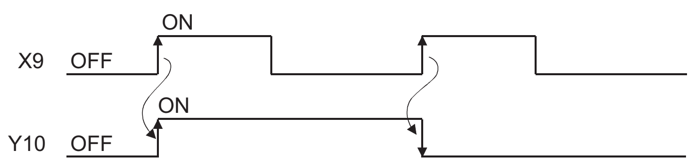
Program Example

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

[Ladder Mode]

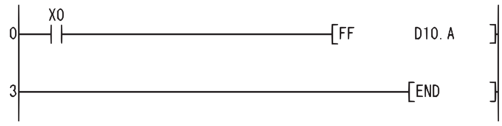


[Timing Chart]

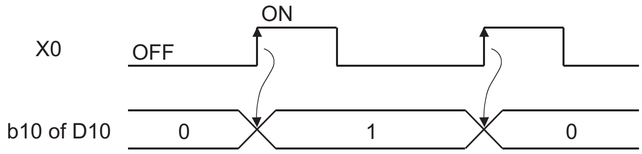


(2) The following program reverses b10 (bit 10) of D10 when X0 goes ON.

[Ladder Mode]

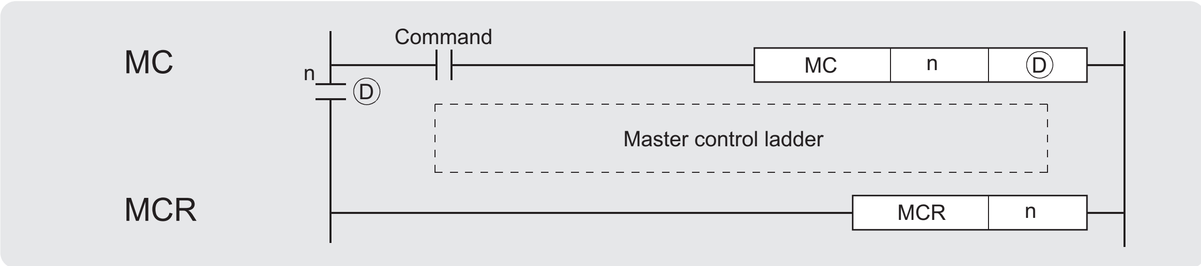


[Timing Chart]



5.4 Master Control Instructions

5.4.1 Setting and resetting the master control (MC, MCR)



n : Nesting (N0 to N14) (Nesting)

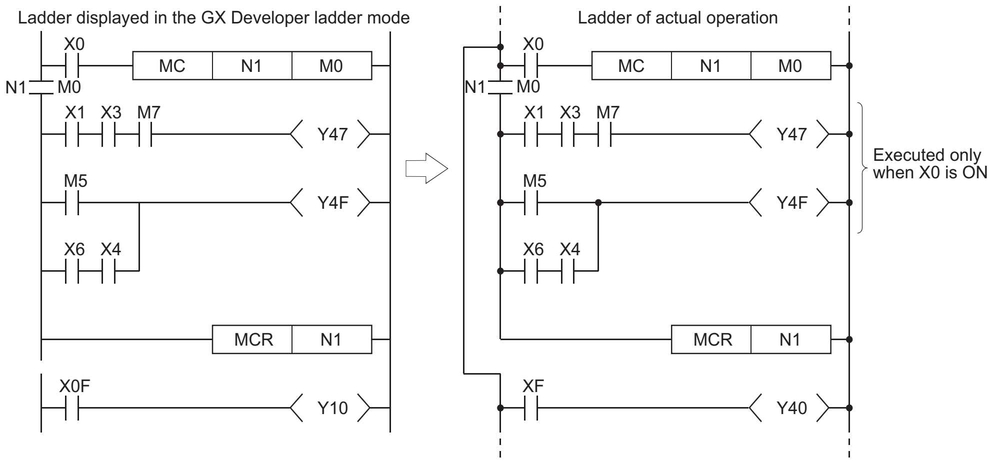
D : Number of the device to turn ON (bits)

Set Data	Internal Devices		Constants	Others
	Bit	Word		N
n	—		—	○
D	○		—	—

Function

(3) The master control instruction is used to enable the creation of highly efficient ladder switching sequence programs, through the opening and closing of a common bus for ladders.

A ladder using the master control would look as shown below:



Remark

Inputting contacts on the vertical bus is not necessary when programming in the write mode of the GX Developer.

These will be automatically displayed when the "conversion" operation is conducted after the creation of the ladder and then "read" mode is set.

MC

(1) If the execution command of the MC instruction is ON when master control is commenced, the result of the operation from the MC instruction to the MCR instruction will be exactly as the instruction (ladder) shows.

If the execution command of the MC instruction is OFF, the result of the operation from the MC instruction to the MCR instruction will be as shown below:

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

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

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

(4) Coils from devices designated by Ⓓ are turned ON when the MC instruction is ON.

Further, using these same devices with the OUT instruction or other instructions will cause them to become double coils, so devices designated by Ⓓ should not be used within other instructions.

MCR

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

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

(3) Use the MC instruction and MCR instruction of the same nesting number as a set.

However, when the MCR instructions are nested in one place, all master controls can be terminated with the lowest nesting (N) number.

(Refer to the "Cautions when Using Nesting Architecture" in the program example.)

Operation Error

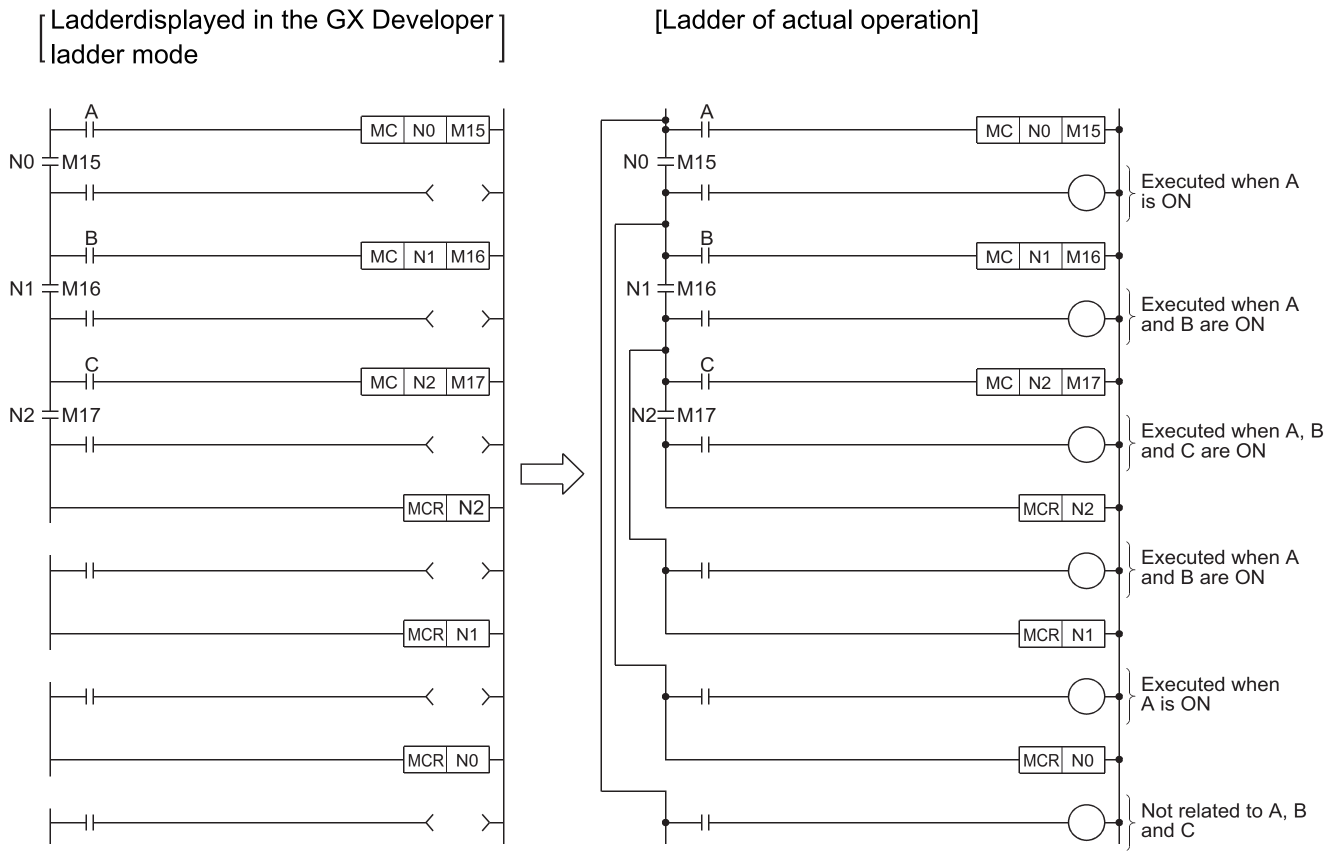
(1) There are no operation errors associated with the MC or MCR instructions.

Program Example

(1) The master control instruction can be used in nesting. The different master control regions are distinguished by nesting (N). Nesting can be performed from N0 to N14.

The use of nesting enables the creation of ladders which successively limit the execution condition of the program.

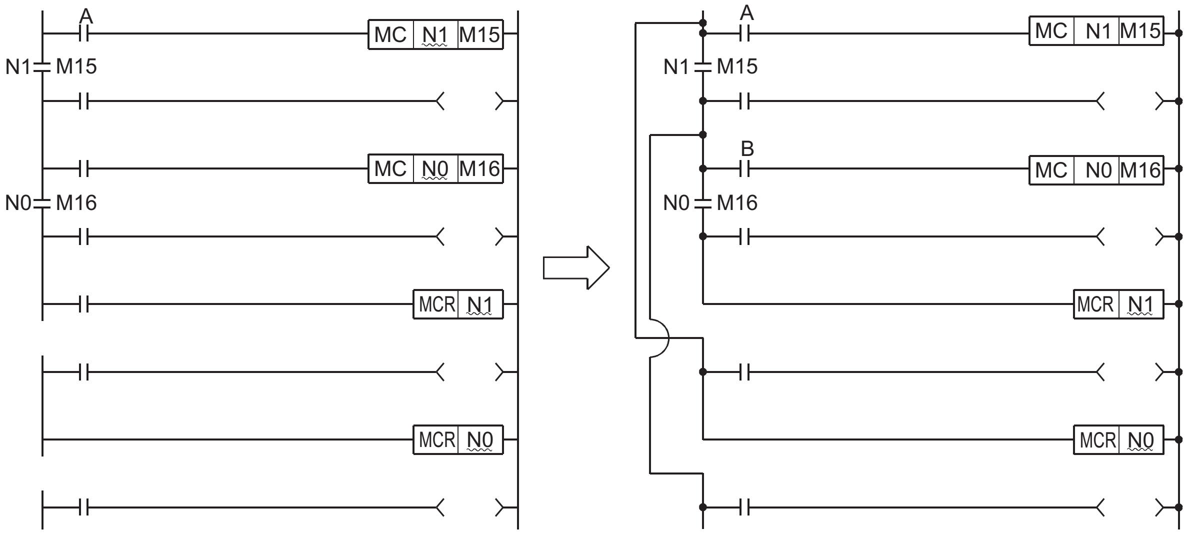
A ladder using nesting would appear as shown below:



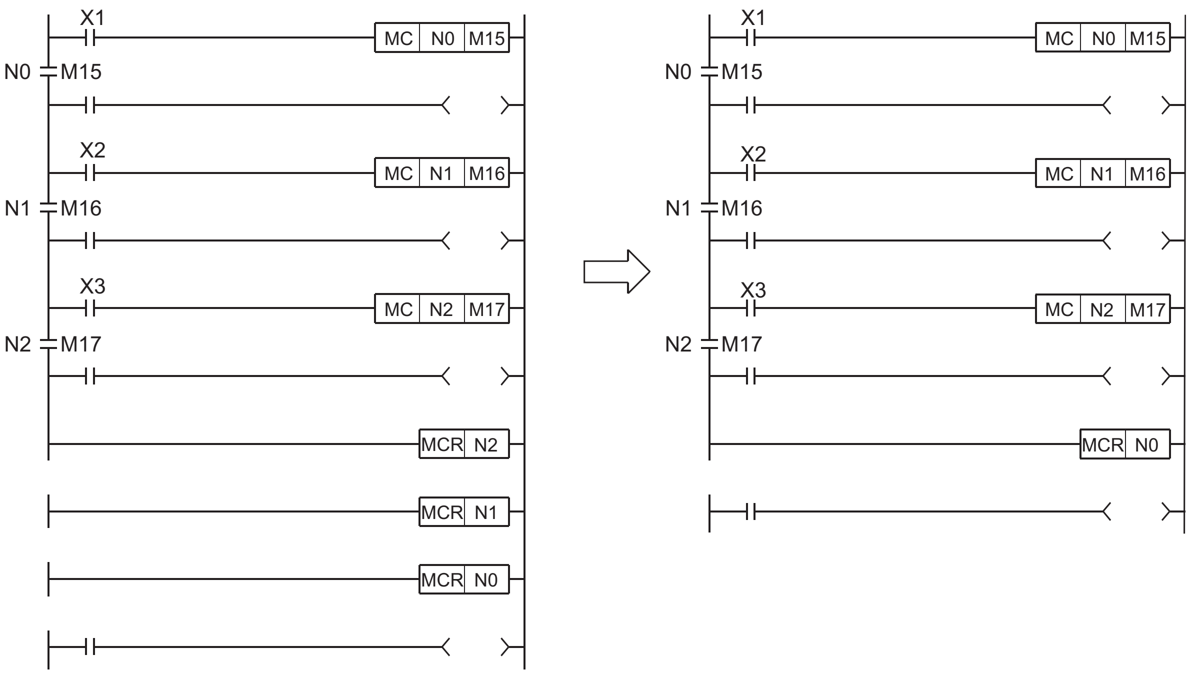
Cautions when Using Nesting Architecture

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

[Ladder displayed in the GX Developer ladder mode] [Ladder of actual operation]

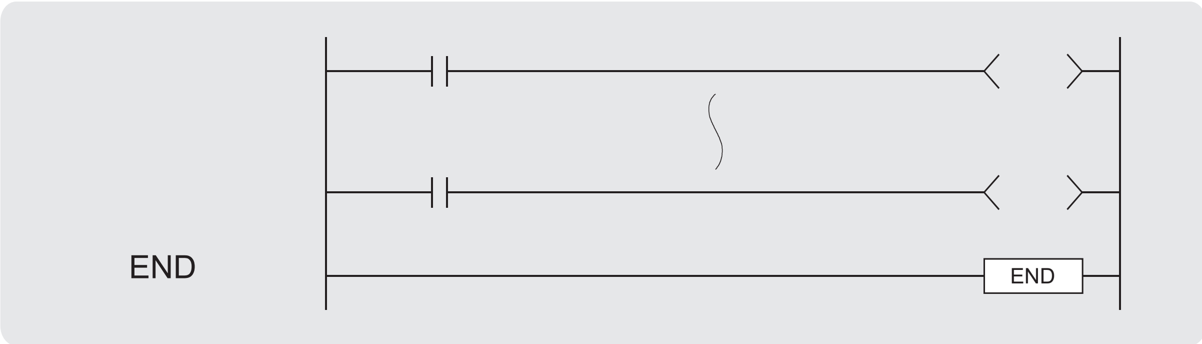


(2) If the nesting architecture results in MCR instructions concentrated in one location, all master controls can be terminated by use of just the lowest nesting number (N).



5.5 Termination Instruction

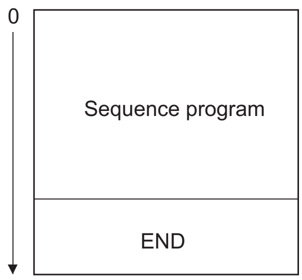
5.5.1 End sequence program (END)



Set Data	Internal Devices		Constants	Others
	Bit	Word		
—	—			

Function

(1) The termination of a sequence program is indicated.
Execution of the END instruction will cause the safety CPU module to terminate the program that was being executed.



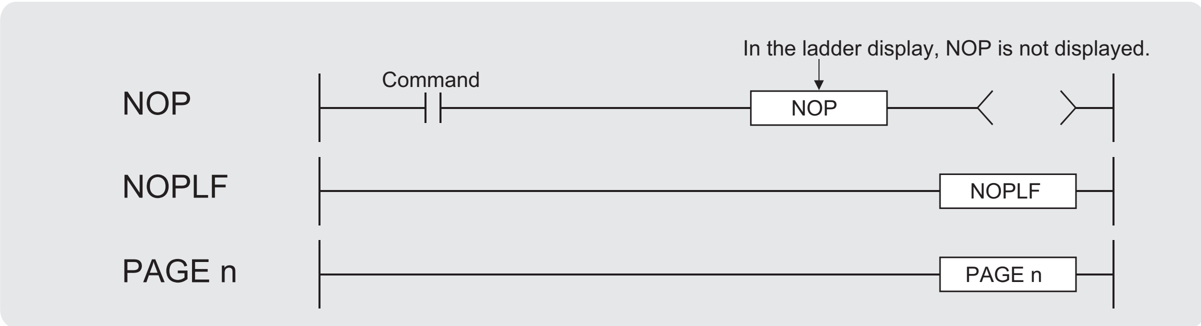
(2) END instruction is automatically set by the GX Developer during programming.

Operation Error

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

5.6 Other Instructions

5.6.1 No-operation (NOP, NOPLF, PAGE n)



Set Data	Internal Devices		Constants K	Others
	Bit	Word		
n	—		○	—

☆ Function

NOP

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

(2) NOP instruction is used to insert space for debugging a sequence program.

NOPLF

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

(2) NOPLF instruction is used to make a page break at a desired position when printing out from the GX Developer.
- 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.

(3) For the print out operation by the GX Developer, refer to the GX Developer Operating Manual.

PAGE n

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

(2) No processing is performed at the GX Developer with this instruction.

Operation Error

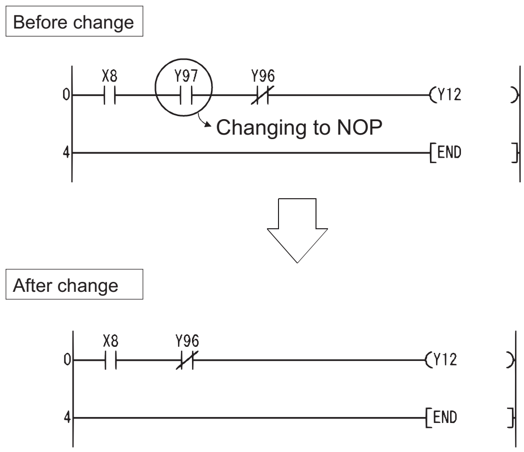
(1) There are no errors associated with the NOP, NOPLF, or PAGE instructions.

Program Example

NOP

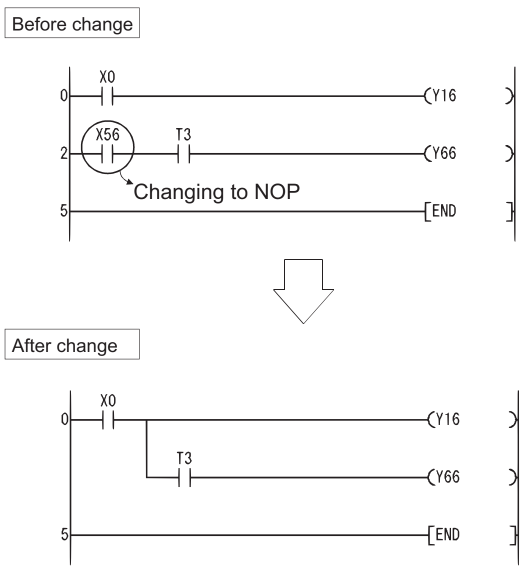
(1) Contact closed......... Deletes AND or ANI instruction.

[Ladder Mode]



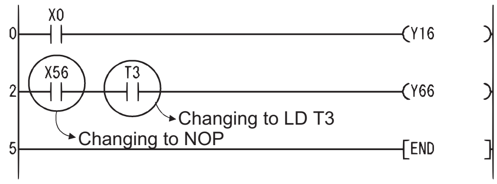
(2) Contact closed......... LD, LDI changed to NOP (Note carefully that changing the LD and LDI instructions to NOP completely changes the nature of the ladder.)

[Ladder Mode]

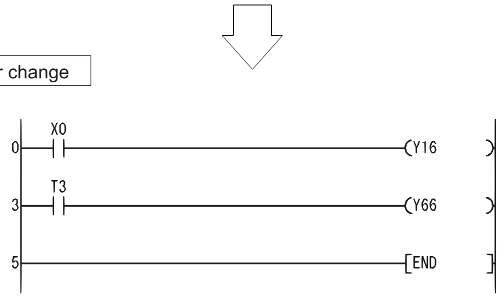


[Ladder Mode]

Before change

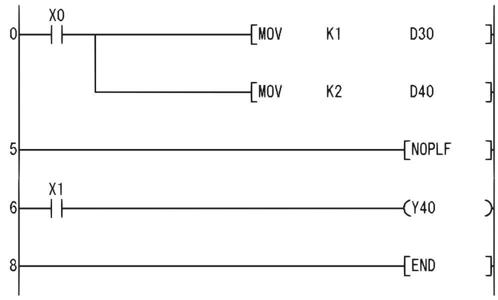


After change

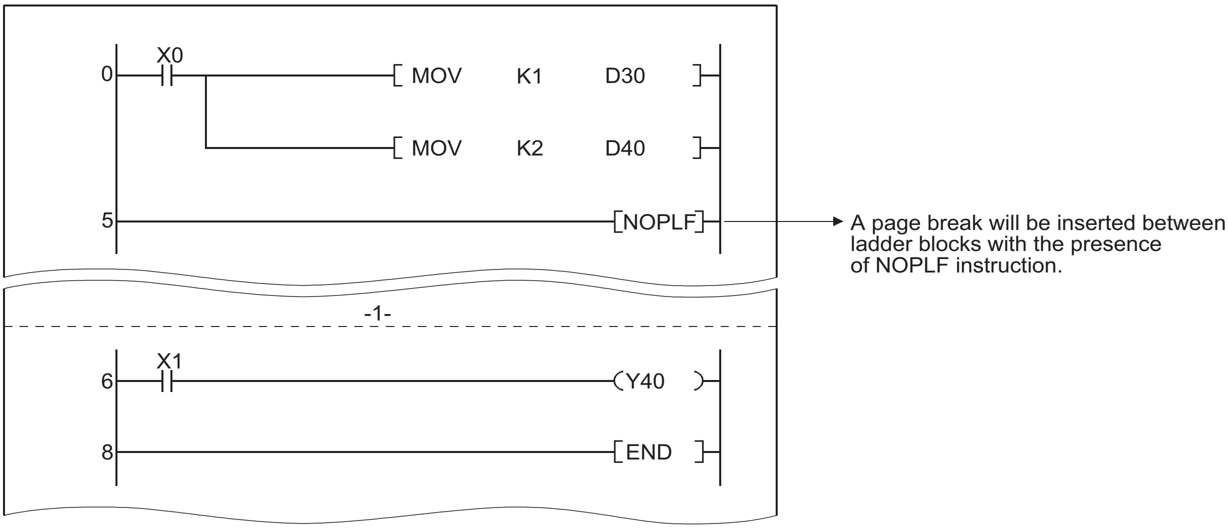


NOPLF

[Ladder Mode]

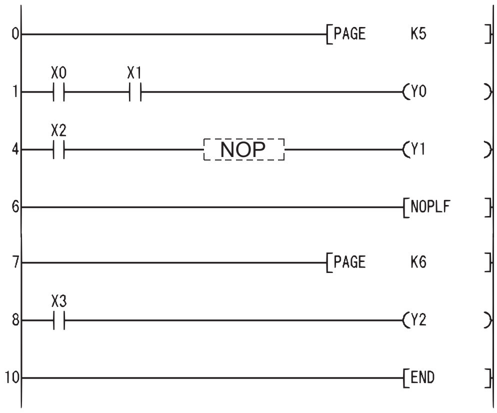


- Printing the ladder will result in the following:



PAGE n

[Ladder Mode]

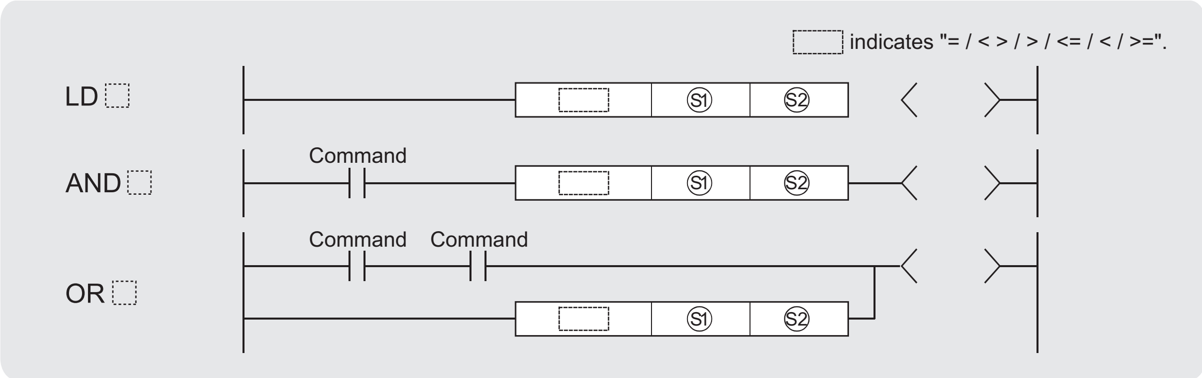


6 BASIC INSTRUCTIONS

Category	Processing Details	Reference section
Comparison operation instruction	Compares data to data	6.1
Arithmetic operation instruction	Adds, subtracts, multiplies, divides, increments, or decrements data with other data	6.2
Data conversion instructions	Converts data types	6.3
Data transfer instruction	Transmits designated data	6.4

6.1 Comparison Operation Instruction

6.1.1 BIN 16-bit data comparisons (= , <> , >, <= , <, >=)



S1, S2 : Data for comparison or start number of the devices where the data for comparison is stored (BIN 16 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
S1	○		○	—
S2	○		○	—

Function

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

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

Instruction Symbol in ☐	Condition	Comparison Operation Result	Instruction Symbol in ☐	Condition	Comparison Operation Result
=	S2 = S1	Continuity	=	S1 ≠ S2	Non-continuity
< >	S1 ≠ S2		< >	S2 = S1	
>	S1 > S2		>	S1 ≦ S2	
<=	S1 ≦ S2		<=	S1 > S2	
<	S1 < S2		<	S1 ≧ S2	
>=	S1 ≧ S2		>=	S1 < S2	

(3) When S1 and S2 are assigned by a hexadecimal constant and the numerical value (8 to F) whose most significant bit (b15) is "1" is designated as a constant, the value is considered as a negative BIN value in comparison operation.

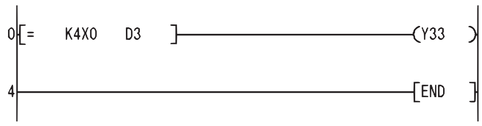
Operation Error

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

Program Example

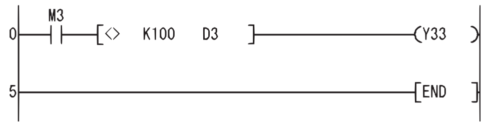
(1) The following program compares the data at X0 to XF with the data at D3, and turns Y33 ON if the data is identical.

[Ladder Mode]



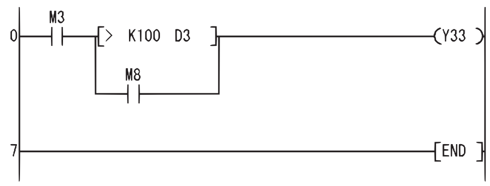
(2) The following program compares BIN value K100 to the data at D3, and establishes continuity if the data in D3 is something other than 100.

[Ladder Mode]



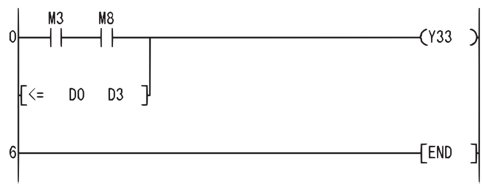
(3) The following program compares the BIN value 100 with the data in X0 to XF, and establishes continuity if the D3 data is less than 100.

[Ladder Mode]

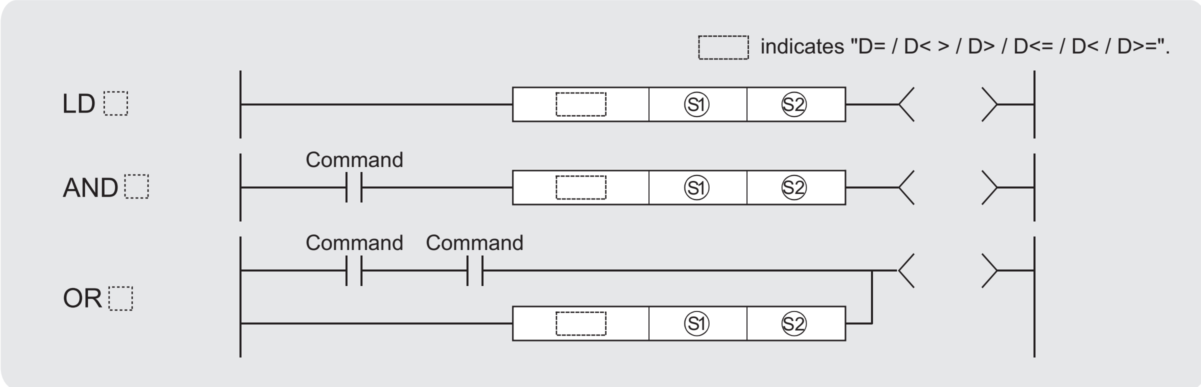


(4) The following program compares the data in D0 and D3, and if the data in D0 is equal to or less than the data in D3, establishes continuity.

[Ladder Mode]



6.1.2 BIN 32-bit data comparisons (D=, D<>, D>, D<=, D<, D>=)



S1, S2 : Data for comparison or start number of the devices where the data for comparison is stored (BIN 32 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
S1	○		○	—
S2	○		○	—

☆ Function

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

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

Instruction Symbol in ▭	Condition	Comparison Operation Result	Instruction Symbol in ▭	Condition	Comparison Operation Result
D =	S2 = S1	Continuity	D =	S1 ≠ S2	Non-continuity
D <>	S1 ≠ S2		D <>	S2 = S1	
D >	S1 > S2		D >	S1 ≦ S2	
D <=	S1 ≦ S2		D <=	S1 > S2	
D <	S1 < S2		D <	S1 ≧ S2	
D >=	S1 ≧ S2		D >=	S1 < S2	

(3) When S1 and S2 are assigned by a hexadecimal constant and the numerical value (8 to F) whose most significant bit (b31) is "1" is designated as a constant, the value is considered as a negative BIN value in comparison operation.

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

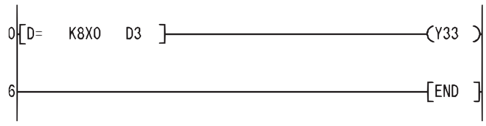
Operation Error

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

Program Example

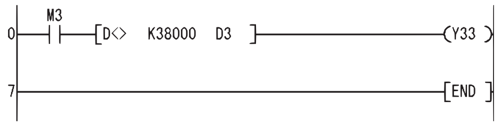
(1) The following program compares the data at X0 to X1F with the data at D3 and D4, and turns Y33 ON, if the data at X0 to X1F and the data at D3 and D4 match.

[Ladder Mode]



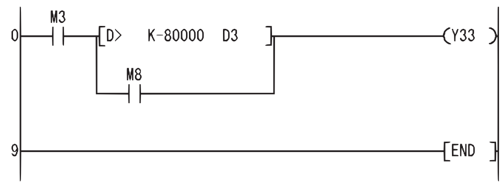
(2) The following program compares BIN value K38000 to the data at D3, and D4, and establishes continuity if the data in D3 and D4 is something other than 38000.

[Ladder Mode]



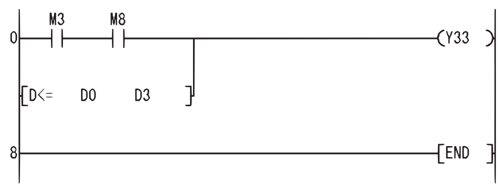
(3) The following program compares BIN value K−80000 to the data at D3 and D4, and establishes continuity if the data in D3 and D4 is less than −80000.

[Ladder Mode]



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

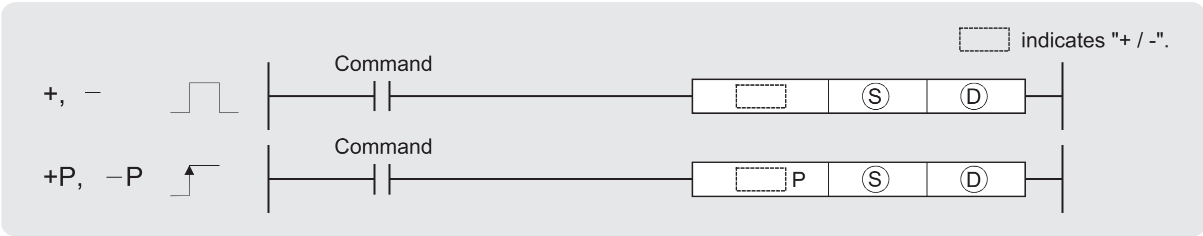
[Ladder Mode]



6.2 Arithmetic Operation Instructions

6.2.1 BIN 16-bit addition and subtraction operations (+(P), –(P))

1 When two data are set (D + S → D, D – S → D)



S : Data for additing/subtracting or start number of the devices where the data for additing/subtracting is stored (BIN 16 bits)

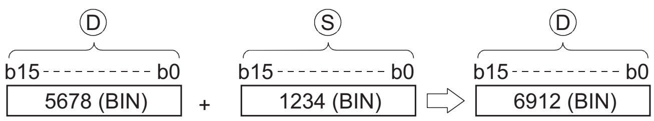
D :Start number of the devices where the data to be added to/subtracted from is stored (BIN 16 bits)

Set Data	Internal Devices		Constants K, H	Others
	Bit	Word		
S	○		○	—
D	○		—	—

Function

+

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



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

(3) The judgment of whether data is positive or negative is made by the most significant bit (b15).
- 0: Positive
- 1: Negative

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

· K32767 (H7FFF) +K2 (H0002) → K−32767 (H8001) ······ Since b15 is "1", the judgment is a negative value.

· K−32768 (H8000) +K−2 (HFFFE) → K32766 (H7FFE) ······ Since b15 is "0", the judgment is a positive value.

–

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

D		S		D
b15 ---------- b0		b15 ---------- b0		b15 ---------- b0
5678 (BIN)	–	1234 (BIN)	⇨	4444 (BIN)

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

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

- 0: Positive
- 1: Negative

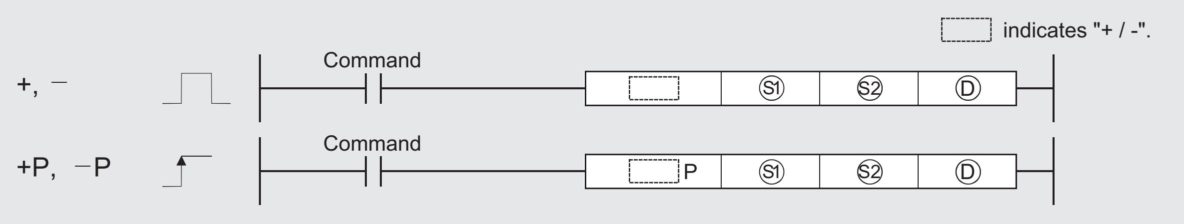
(4) The following will happen when an underflow or overflow is generated in an operation result: The carry flag in this case does not go ON.

- K–32768 – K2 ——→ K32766 ········ Since b15 is "0", the judgment is a positive value.
  (H8000) (H0002) (H7FFE)
- K32767 – K–2 ——→ K–32767 ······ Since b15 is "1", the judgment is a negative value.
  (H7FFF) (HFFFE) (H8001)

Operation Error

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

2 When three data are set (S1 + S2 → D, S1 − S2 → D)



S1 : Data to be added to/subtracted from or start number of the devices where the data to be added to/ subtracted from is stored (BIN 16 bits)

S2 : Data for additing/subtracting or start number of the devices where the data for additing/subtracting is stored (BIN 16 bits)

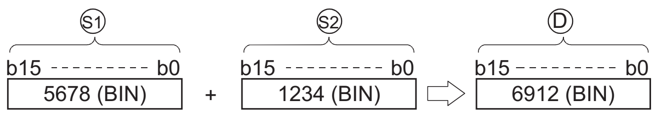
D : Start number of the devices where the addition/subtraction operation result will be stored (BIN 16 bits)

Set Data	Internal Devices		Constants K, H	Others
	Bit	Word		
S1	○		○	—
S2	○		○	—
D	○		—	—

Function

+

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



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

(3) The judgment of whether data is positive or negative is made by the most significant bit (b15).
- 0: Positive
- 1: Negative

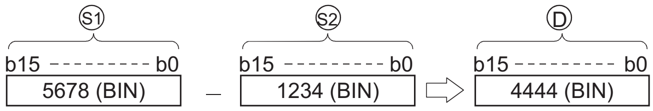
(4) The following will happen when an underflow or overflow is generated in an operation result: The carry flag in this case does not go ON.

· K32767 (H7FFF) +K2 (H0002) ⟶ K−32767 (H8001) ······Since b15 is "1", the judgment is a negative value.

· K−32768 (H8000) +K−2 (HFFFE) ⟶ K32766 (H7FFE) ·········Since b15 is "0", the judgment is a positive value.

–

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



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

(3) The judgment of whether data is positive or negative is made by the most significant bit (b15).
- 0: Positive
- 1: Negative

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

- K−32768 − K2 (H8000) (H0002) → K32766 (H7FFE) ·········Since b15 is "0", the judgment is a positive value.
- K32767 − K−2 (H7FFF) (HFFFE) → K−32767 (H8001) ······Since b15 is "1", the judgment is a negative value.

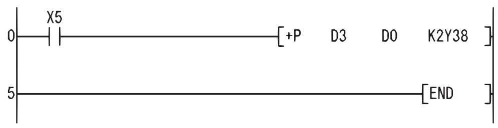
Operation Error

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

Program Example

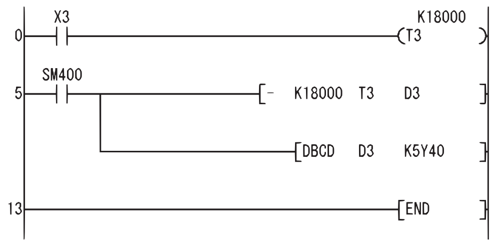
(1) The following program adds, when X5 is turned ON, the data at D3 and D0 and outputs the operation result at Y38 to Y3F.

[Ladder Mode]



(2) The following program outputs the difference between the set value for timer T3 and its present value in BCD to Y40 to Y53.

[Ladder Mode]

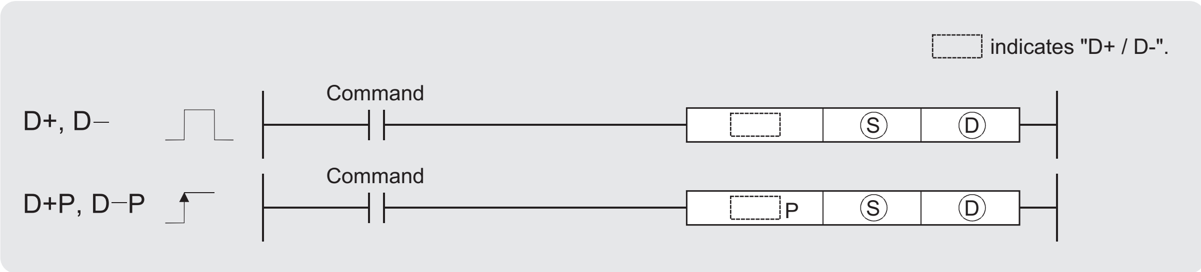


6 BASIC INSTRUCTIONS

+(P), -(P)

6.2.2 BIN 32-bit addition and subtraction operations (D+(P), D–(P))

1 When two data are set ((D + 1, D) + (S + 1, S) → (D + 1, D),
(D +1, D) – (S + 1, S) → (D +1, D))



S : Data for additing/subtracting or start number of the devices where the data for additing/subtracting is stored (BIN 32 bits)

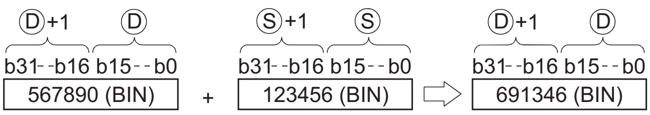
D : Start number of the devices where the data to be added to/subtracted from is stored (BIN 32 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
S	○		○	—
D	○		—	—

Function

D+

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



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

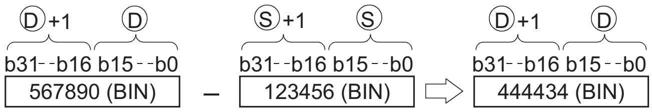
(3) Judgment of whether the data is positive or negative is made on the basis of the most significant bit (b31).
- 0: Positive
- 1: Negative

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

- K2147483647 (H7FFFFFFF) +K2 (H00000002) → K–2147483647 (H80000001) ······Since b31 is "1", the judgment is a negative value.
- K–2147483648 (H80000000) +K–2 (HFFFFFFFE) → K2147483646 (H7FFFFFFE) ·········Since b31 value is "0", the judgment is a positive value.

D –

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



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

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

- 0: Positive
- 1: Negative

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

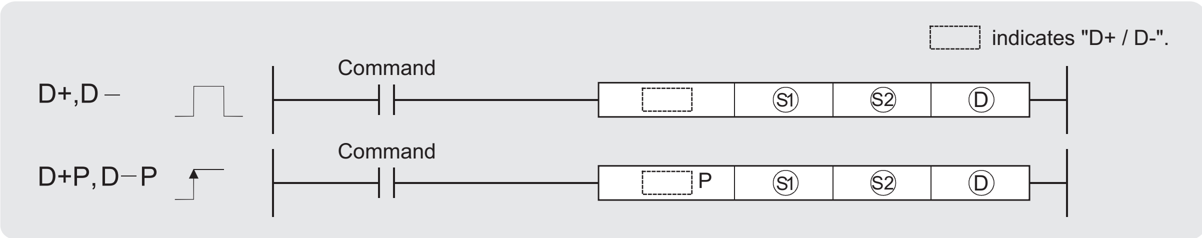
· K−2147483648 − K2 → K2147483646 ········ Since b31 is "0", the judgment is a positive value.
(H80000000) (H00000002) (H7FFFFFFE)

· K2147483647 − K−2 → K−2147483647 ······ Since b31 is "1", the judgment is a negative value.
(H7FFFFFFF) (HFFFFFFFE) (H80000001)

Operation Error

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

2 When three data are set ((S1 + 1, S1) + (S2 + 1, S2) → (D + 1, D), (S1 + 1, S1) – (S2 + 1, S2) → (D +1, D))



S1 : Data to be added to/subtracted from or start number of the devices where the data to be added to/subtracted from is stored (BIN 32 bits)

S2 : Data for additing/subtracting or start number of the devices where the data for additing/subtracting is stored (BIN 32 bits)

D : Start number of the devices where the addition/subtraction operation result will be stored (BIN 32 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
S1	○		○	—
S2	○		○	—
D	○		—	—

Function

D+

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

S1+1 (b31 - - b16)	S1 (b15 - - b0)		S2+1 (b31 - - b16)	S2 (b15 - - b0)		D+1 (b31 - - b16)	D (b15 - - b0)
567890 (BIN)		+	123456 (BIN)		⇨	691346 (BIN)	

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

(3) Judgment of whether the data is positive or negative is made on the basis of the most significant bit (b31).
- 0: Positive
- 1: Negative

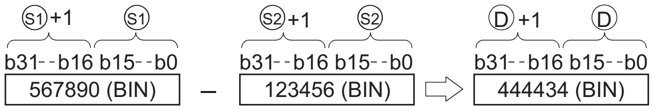
(4) The following will happen when an underflow or overflow is generated in an operation result: The carry flag in this case does not go ON.

· K2147483647 (H7FFFFFFF) +K2 (H00000002) → K−2147483647 (H80000001) ···Since b31 is "1", the judgment is a negative value.

· K−2147483648 (H80000000) +K−2 (HFFFFFFFE) → K2147483646 (H7FFFFFFE) ······Since b31 is "0", the judgment is a positive value.

D –

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



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

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

- 0: Positive
- 1: Negative

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

· K−2147483648 − K2 ⟶ K2147483646 ······ Since b31 is "0", the judgment is a positive value.
(H80000000) (H00000002) (H7FFFFFFE)

· K2147483647 − K−2 ⟶ K−2147483647 ··· Since b31 is "1", the judgment is a negative value.
(H7FFFFFFF) (HFFFFFFFE) (H80000001)

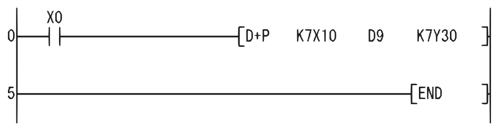
Operation Error

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

Program Example

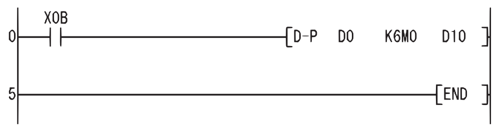
(1) The following program adds 28-bit data from X10 to X2B to the data at D9 and D10 when X0 goes ON, and outputs the result of the operation to Y30 to Y4B.

[Ladder Mode]

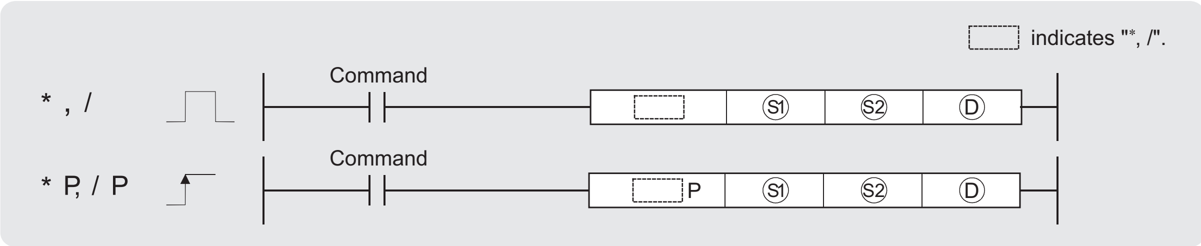


(2) The following program subtracts the data from M0 to M23 from the data at D0 and D1 when XB goes ON, and stores the result at D10 and D11.

[Ladder Mode]



6.2.3 BIN 16-bit multiplication and division operations (*(P), /(P))



S1 : Data to be multiplied/divided or start number of the devices where the data to be multiplied/divided is stored (BIN 16 bits)

S2 : Data for multiplying/dividing or start number of the devices where the data for multiplying/dividing is stored (BIN 16 bits)

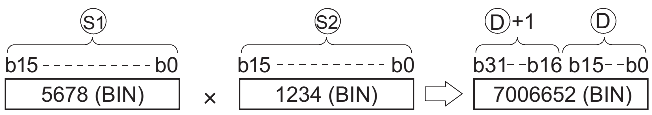
D : Start number of the devices where the multiplication/division operation result will be stored (BIN 32 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
S1	○		○	—
S2	○		○	—
D	○		—	—

Function

*

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



(2) If D is a bit device, designation is made from the lower bits.

Example

K1........ Lower 4 bits (b0 to b3)
K4........ Lower 16 bits (b0 to b15)
K8........ 32 bits (b0 to b31)

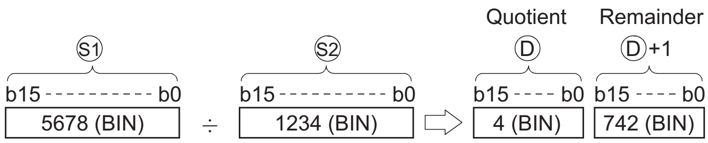
(3) Values for S1 and S2 can be designated between −32768 and 32767 (BIN, 16 bits).

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

- 0: Positive
- 1: Negative

/

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



(2) If a word device has been used, the result of the division operation is stored as 32 bits, and both the quotient and remainder are stored; if a bit device has been used, 16 bits are used and only the quotient is stored.

Quotient: Stored at the lower 16 bits
Remainder: Stored at the upper 16 bits (Stored only when using a word device)

(3) Values for S1 and S2 can be designated between −32768 and 32767 (BIN, 16 bits).

(4) Judgment whether values for S1, S2, D and D + 1 are positive or negative is made on the basis of the most significant bit (b15). (Sign is attached to both the quotient and remainder.)

- 0: Positive
- 1: Negative

Operation Error

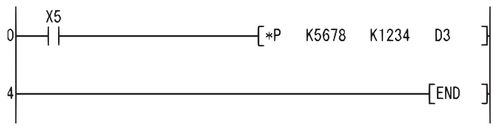
(1) In any of the following cases, an operation error occurs, the error flag (SM0) turns ON, and the corresponding error code is stored into SD0.

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

Program Example

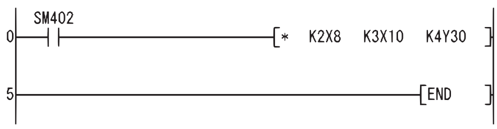
(1) The following program divides "5678" by "1234" when X5 goes ON, and stores the result at D3 and D4.

[Ladder Mode]



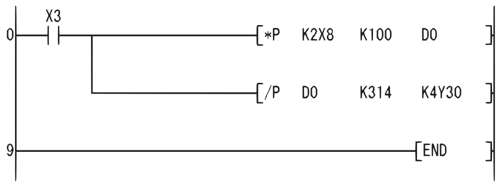
(2) The following program divides BIN data at X8 to XF by BIN data at X10 to X1B, and outputs the result of the division operation to Y30 to Y3F.

[Ladder Mode]



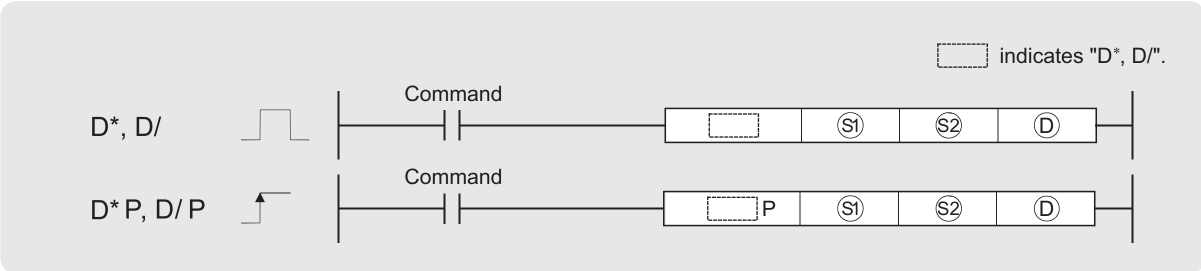
(3) The following program divides, when X3 is turned ON, the data at X8 to XF by 3.14 and outputs the operation result at Y30 to Y3F.

[Ladder Mode]



6 BASIC INSTRUCTIONS
*(P), /(P)

6.2.4 BIN 32-bit multiplication and division operations (D*(P), D/(P))



S1 : Data to be multiplied/divided or start number of the devices where the data to be multiplied/divided is stored (BIN 32 bits)

S2 : Data for multiplying/dividing or start number of the devices where the data for multiplying/dividing is stored (BIN 32 bits)

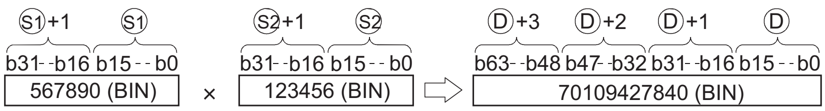
D : Start number of the devices where the multiplication/division operation result will be stored (BIN 64 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
S1	○		○	—
S2	○		○	—
D	○		—	—

Function

D*

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



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

Example

K1........ Lower 4 bits (b0 to b3)
K4........ Lower 16 bits (b0 to b15)
K8........ Lower 32 bits (b0 to b31)

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

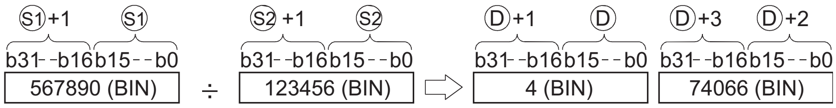
(3) The values for S1 and S2 can be designated at between −2147483648 and 2147483647 (BIN 32 bits).

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

- 0: Positive
- 1: Negative

D/

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



(2) With a word device, the division operation result is stored in 64 bits and both the quotient and remainder are stored. With a bit device, only the quotient is stored as the operation result in 32 bits.

Quotient: Stored at the lower 32 bits
Remainder: Stored at the upper 32 bits (Stored only when using a word device)

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

(4) Judgment whether values for S1, S2, D and D + 2 are positive or negative is made on the basis of the most significant bit (b31).
(Sign is attached to both the quotient and remainder)

- 0: Positive
- 1: Negative

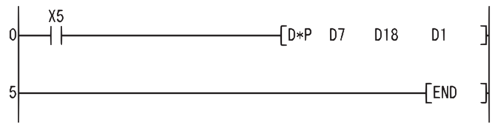
Operation Error

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

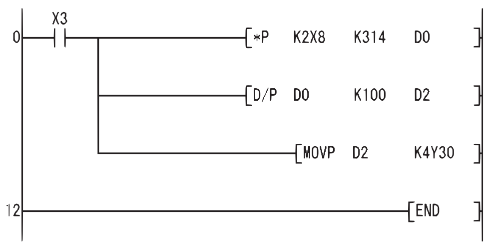
- Attempt to divide S2 by 0. (Error code: 4100)

Program Example

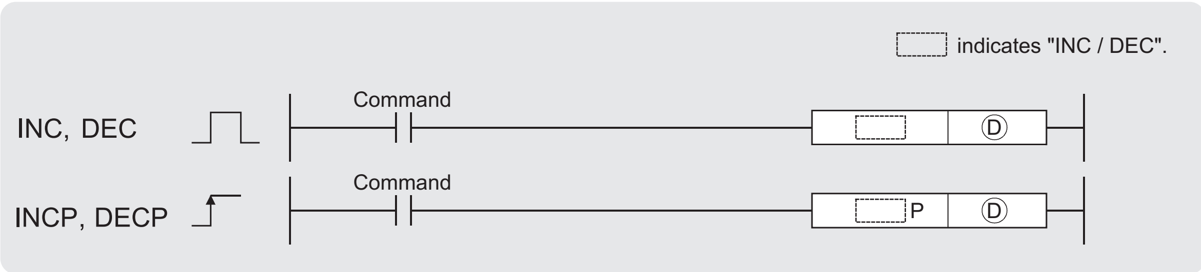
(1) The following program divides the BIN data at D7 and D8 by the BIN data at D18 and D19 when X5 is ON, and stores the result at D1 to D4.
[Ladder Mode]



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



6.2.5 Incrementing and decrementing 16-bit BIN data (INC(P), DEC(P))



Ⓓ : Start number of devices for INC (+1)/DEC (−1) operation (BIN 16 bits)

Set Data	Internal Devices		Constants	Others
	Bit	Word		
Ⓓ	○		—	

Function

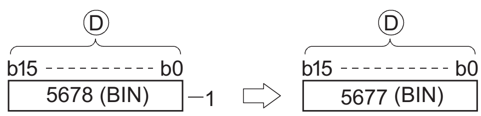
INC

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

(2) When INC/INCP operation is executed for the device designated by Ⓓ, whose content is 32767, the value −32768 is stored at the device designated by Ⓓ.

DEC

(1) Subtracts 1 from the device designated by Ⓓ (16-bit data).



(2) 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 Ⓓ.

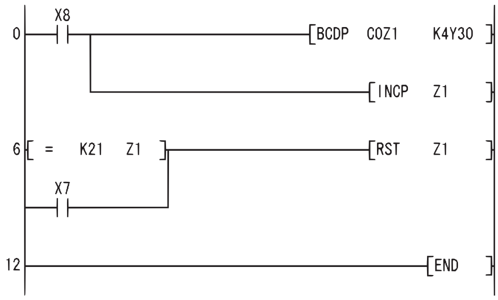
Operation Error

(1) There are no operation errors associated with the INC(P)/DEC(P) instruction.

Program Example

(1) The following program outputs the present value at the counter C0 to C20 to the area Y30 to Y3F in BCD, every time X8 is turned ON. (When present value is less than 9999)

[Ladder Mode]



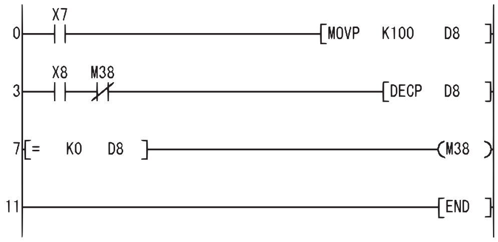
Outputs the present value of C (D+Z1) to Y30 to Y3F.

Executes Z1+1.

Sets 0 to Z1 when Z1=21 or X7 (reset input).

(2) The following is a down counter program.

[Ladder Mode]

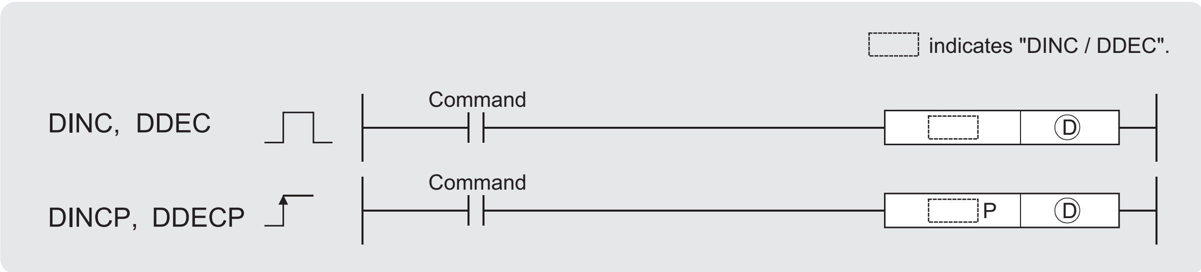


Transfers 100 to D8 when X7 is turned ON.

Turns X8 from OFF to ON
when M38 is OFF, which executes "D8 - 1".

M38 is turned ON when "D8=0".

6.2.6 Incrementing and decrementing 32-bit BIN data (DINC(P), DDEC(P))



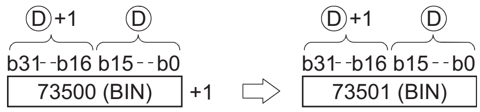
D : Start number of devices for DINC(+1) or DDEC(–1) operation (BIN 32 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants	Others
D	○		—	

Function

DINC

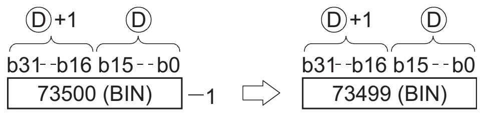
(1) Adds 1 to the device designated by D (32-bit data).



(2) When DINC/DINCP operation is executed for the device designated by D, whose content is 2147483647, the value −2147483648 is stored at the device designated by D.

DDEC

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



(2) 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.

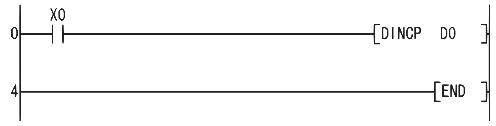
Operation Error

(1) There are no operation errors associated with DINC(P) or DDEC(P).

Program Example

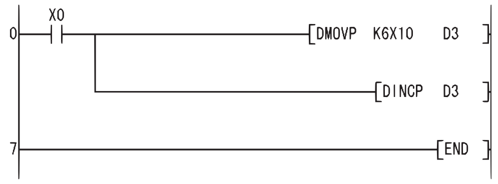
(1) The following program adds 1 to the data at D0 and D1 when X0 is ON.

[Ladder Mode]



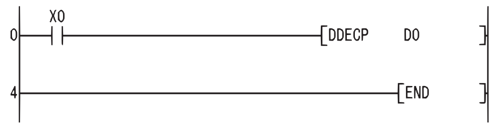
(2) The following program adds 1 to the data set at X10 to X27 when X0 goes ON, and stores the result at D3 and D4.

[Ladder Mode]



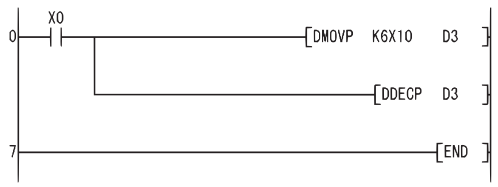
(3) The following program subtracts 1 from the data at D0 and D1 when X0 goes ON.

[Ladder Mode]



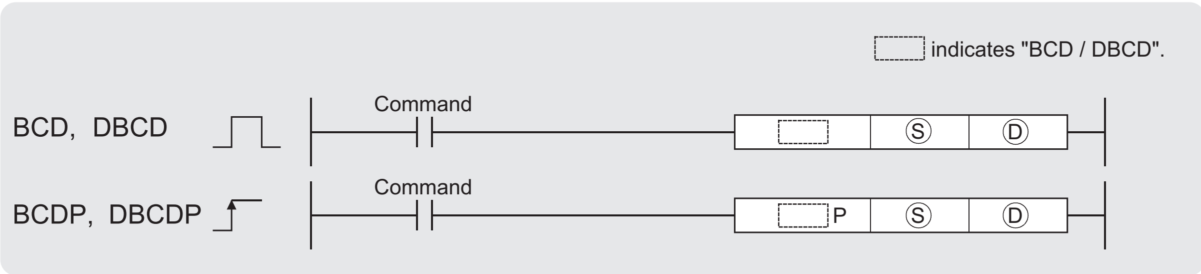
(4) The following program subtracts 1 from the data set at X10 to X27 when X0 goes ON, and stores the result at D3 and D4.

[Ladder Mode]



6.3 Data Conversion Instructions

6.3.1 Conversion from BIN data to 4-digit and 8-digit BCD (BCD(P), DBCD(P))



Ⓢ : BIN data or start number of the devices where the BIN data is stored (BIN 16/32 bits)

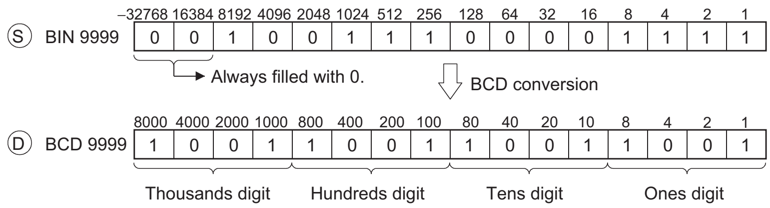
Ⓓ : Start number of the devices where BCD data will be stored (BCD 4/8 digits)

Set Data	Internal Devices		Constants K, H	Others
	Bit	Word		
Ⓢ	○		○	—
Ⓓ	○		—	—

Function

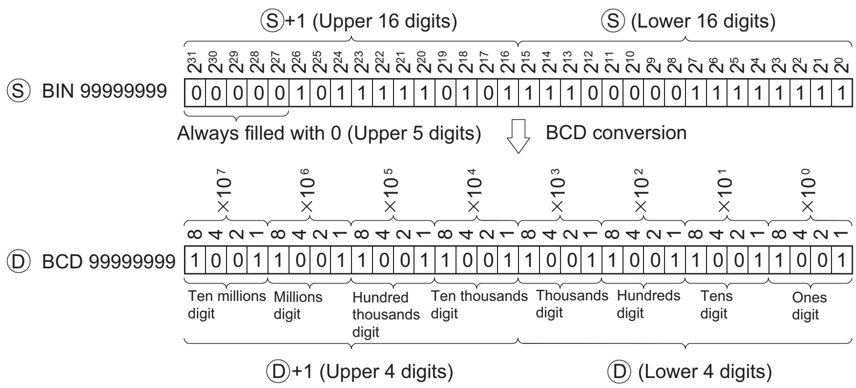
BCD

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



DBCD

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



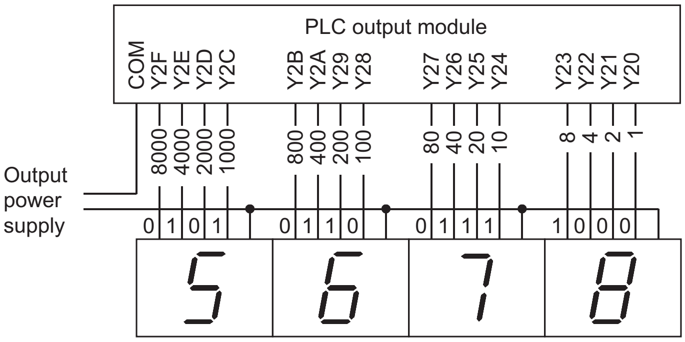
Operation Error

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

- The data of Ⓢ is other than 0 to 9999 at BCD instruction. (Error code: 4100)
- The data of Ⓢ or Ⓢ+1 is other thean 0 to 99999999 at DBCD instruction. (Error code: 4100)

Program Example

(1) The following program outputs the present value of C4 from Y20 to Y2F to the BCD display device.



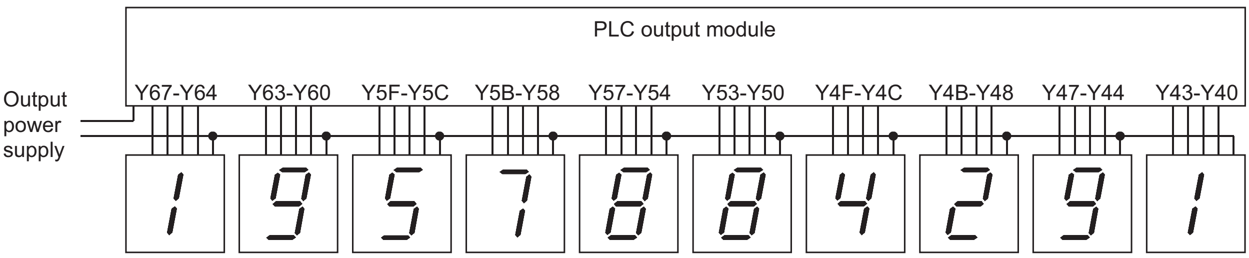
7-segment indicator

[Ladder Mode]

```
    SM400
0 ──┤ ├──────────────────[BCDP  C4   K4Y20 ]

4 ───────────────────────────────────[END  ]
```

(2) The following program outputs 32-bit data from D0 to D1 to Y40 to Y67.



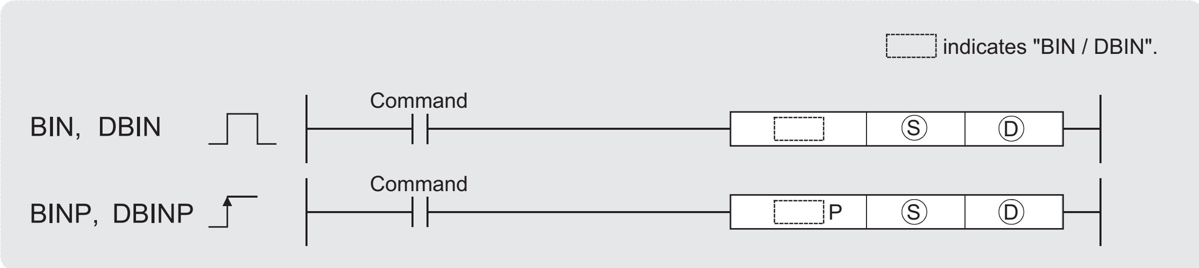
7-segment indicator

[Ladder Mode]

```
    SM400
0 ──┤ ├──┬──────────[D/    D0    K10000   D2   ]
         │
         ├──────────[DBCDP D2             K6Y50]
         │
         └──────────[BCD   D4             K4Y40]

12 ───────────────────────────────[END         ]
```

6.3.2 Conversion from BCD 4-digit and 8-digit data to BIN data (BIN(P), DBIN(P))



Ⓢ : BCD data or start number of the devices where the BCD data is stored (BCD 4/8 digits)

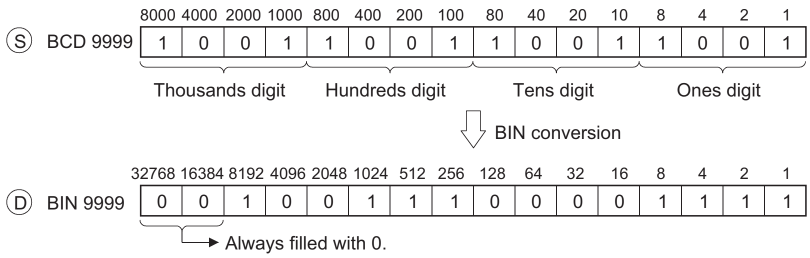
Ⓓ : Start number of the devices where BIN data will be stored (BIN 16/32 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
Ⓢ	○		○	—
Ⓓ	○		—	—

Function

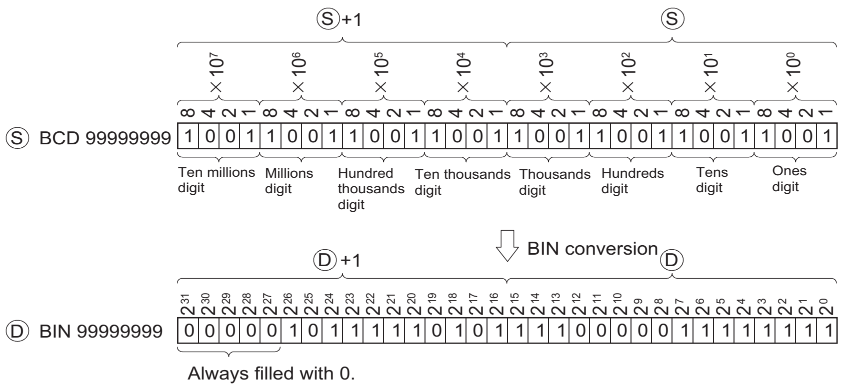
BIN

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



DBIN

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



Operation Error

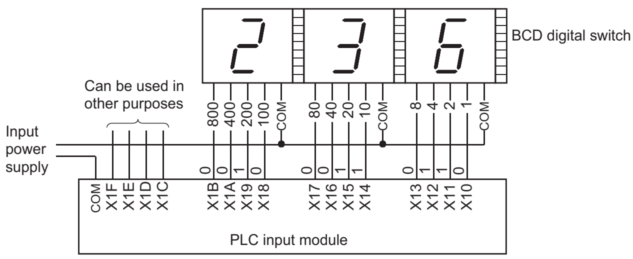
(1) In the following cases, an operation error occurs, the error flag (SM0) turns ON, an error code is stored in SD0, and the instruction is not executed.

- When values other than 0 to 9 are designated to any digits of Ⓢ.(Error code: 4100)

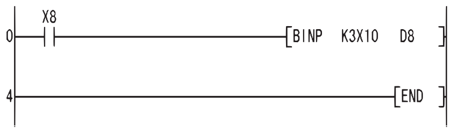
In this regard, however, the error above can be suppressed by turning SM722 ON.
However, the instruction is not executed regardless of whether SM722 is turned ON or OFF if the designated value is out of the available range.
For the BINP/DBINP instruction, the next operation will not be performed until the command (execution condition) is turned from OFF to ON regardless of the presence/absence of an error.

Program Example

(1) The following program converts the BCD data at X10 to X1B to BIN when X8 is ON, and stores it at D8.

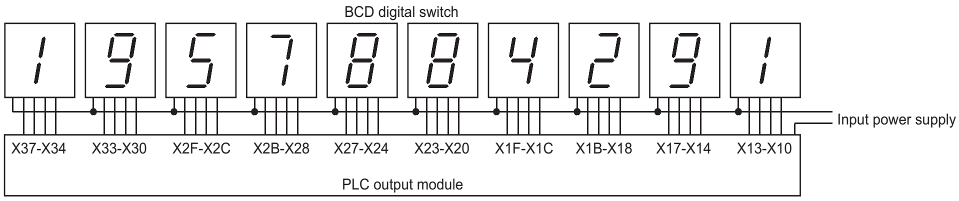


[Ladder Mode]

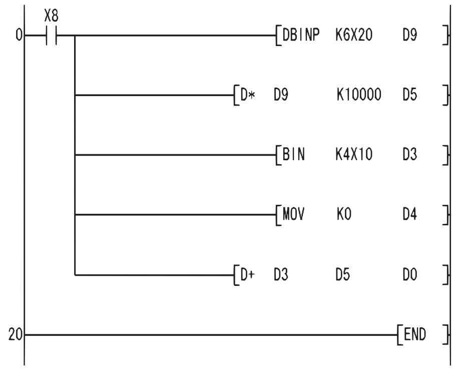


(2) The following program converts the BCD data at X10 to X37 to BIN when X8 is ON, and stores it at D0 and D1.

(Addition of the BIN data converted from BCD at X20 to X37 and the BIN data converted from BCD at X10 to X1F)

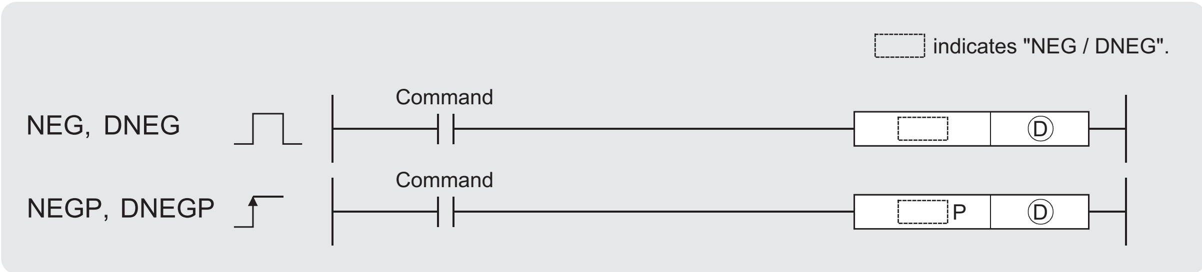


[Ladder Mode]



If the data set at X10 to X37 is a BCD value which exceeds 2147483647, the value at D0 and D1 will be a negative value, because it exceeds the range of numerical values that can be handled by a 32-bit device.

6.3.3 Complement of 2 of BIN 16- and 32-bit data (sign reversal) (NEG(P), DNEG(P))



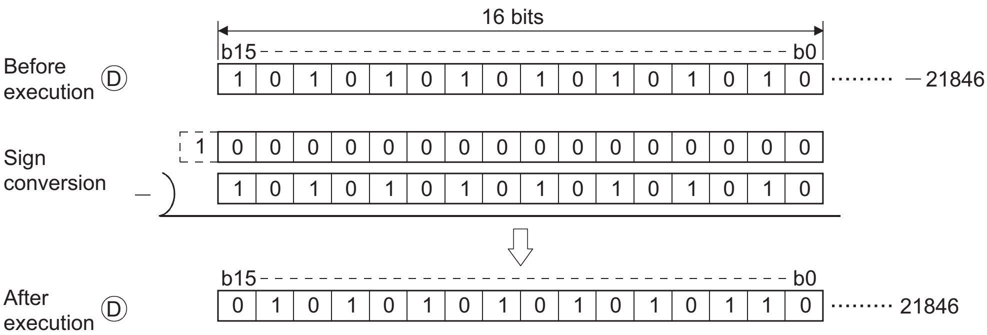
Ⓓ : Start number of the devices where the data for which complement of 2 is performed is stored (BIN 16/32 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants	Others
Ⓓ	○		—	

Function

NEG

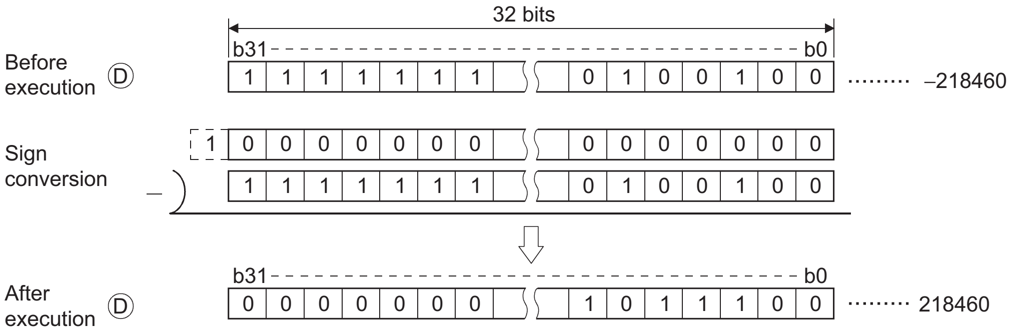
(1) Reverses the sign of the 16-bit device designated by Ⓓ and stores at the device designated by Ⓓ.



(2) Used when reversing positive and negative signs.

DNEG

(1) Reverses the sign of the 32-bit device designated by Ⓓ and stores at the device designated by Ⓓ.



(2) Used when reversing positive and negative signs.

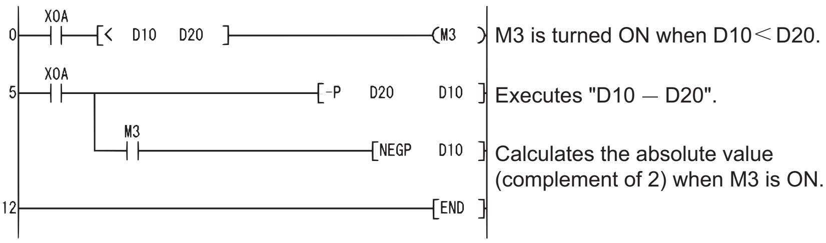
Operation Error

(1) There are no operation errors associated with the NEG(P) or DNEG(P) instructions.

Program Example

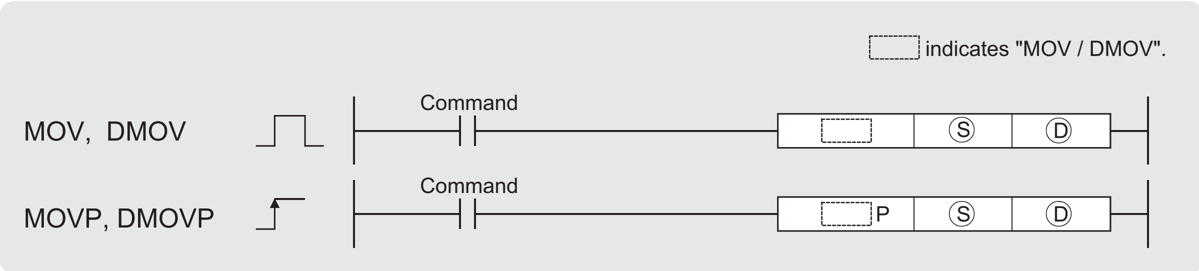
(1) The following program calculates a total for the data at D10 through D20 when XA goes ON, and seeks an absolute value if the result is negative.

[Ladder Mode]



6.4 Data Transfer Instructions

6.4.1 16-bit and 32-bit data transfers (MOV(P), DMOV(P))



Ⓢ : Data to be transferred or the number of the device where the data to be transferred is stored (BIN 16/32 bits)

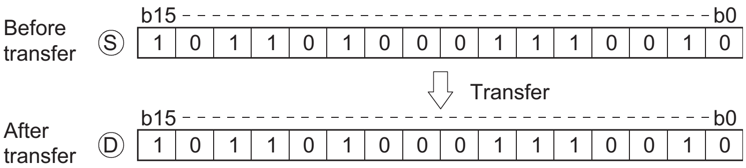
Ⓓ : Number of the device where the data will be transferred (BIN 16/32 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
Ⓢ	○		○	—
Ⓓ	○		—	—

Function

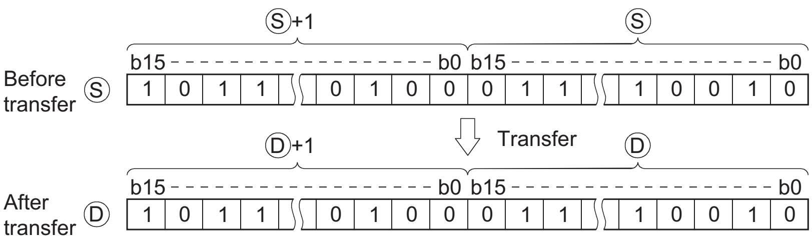
MOV

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



DMOV

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



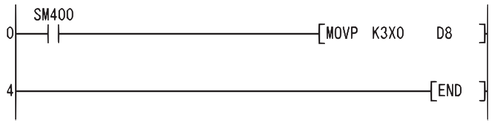
Operation Error

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

Program Example

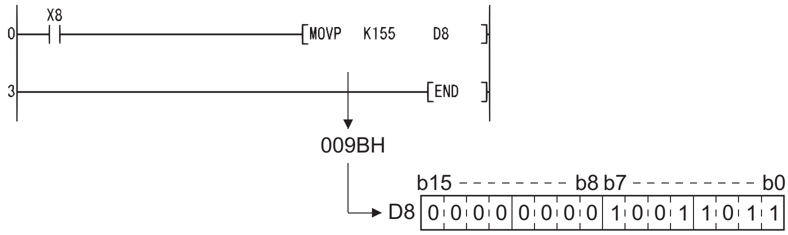
(1) The following program stores input data from X0 to XB at D8.

[Ladder Mode]



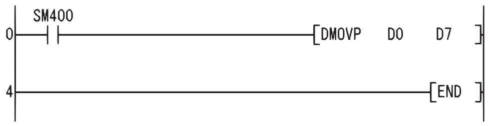
(2) The following program stores the constant K155 at D8 when X8 goes ON.

[Ladder Mode]



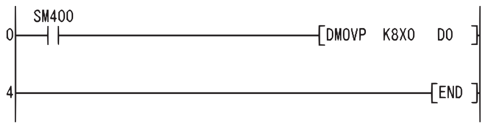
(3) The following program stores the data from D0 and D1 at D7 and D8.

[Ladder Mode]

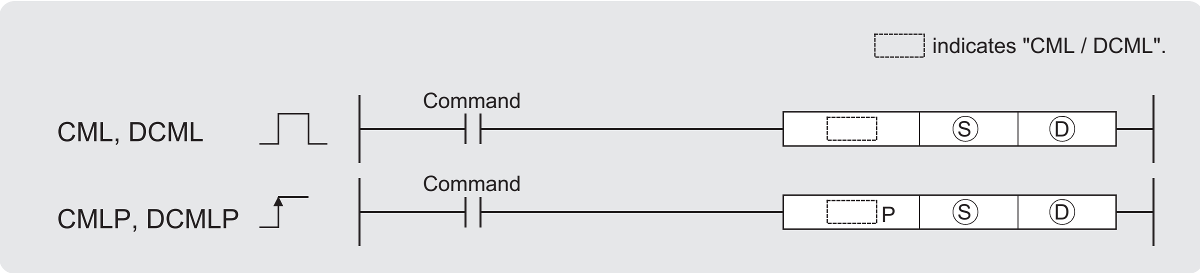


(4) The following program stores the data from X0 to X1F at D0 and D1.

[Ladder Mode]



6.4.2 16-bit and 32-bit negation transfers (CML(P), DCML(P))



Ⓢ : Data to be reversed or the number of the device where data to be reversed is stored (BIN 16/32 bits)

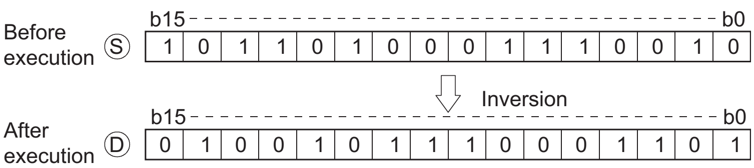
Ⓓ : Number of the device where the reversing result will be stored (BIN 16/32 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
Ⓢ	○		○	—
Ⓓ	○		—	—

Function

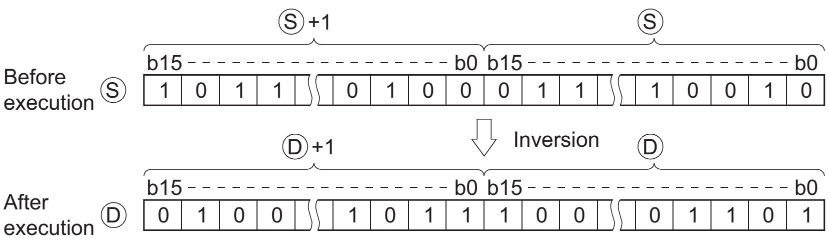
CML

(1) Inverts 16-bit data designated by Ⓢ bit by bit, and transfers the result to the device designated by Ⓓ.



DCML

(1) Inverts 32-bit data designated by Ⓢ bit by bit, and transfers the result to the device designated by Ⓓ.



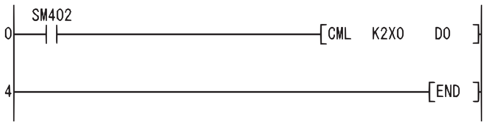
Operation Error

(1) There are no operation errors associated with the CML(P) or DCML(P) instructions.

Program Example

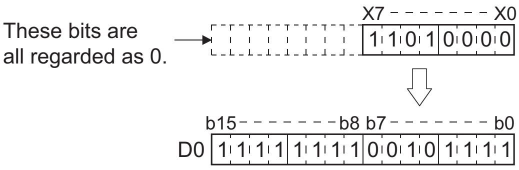
(1) The following program inverts the data from X0 to X7, and transfers result to D0.

[Ladder Mode]



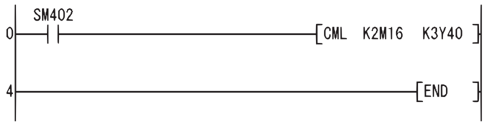
[Operation]

When "Number of bits of (S) < Number of bits of (D)"



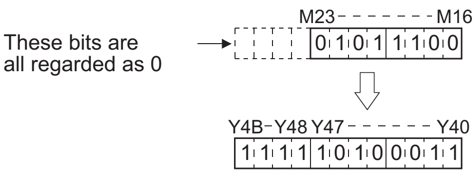
(2) The following program inverts the data at M16 to M23, and transfers the result to Y40 to Y47.

[Ladder Mode]



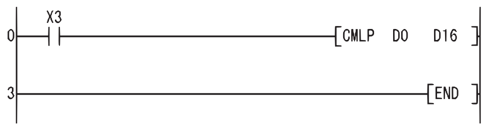
[Operation]

When "Number of bits of (S) < Number of bits of (D)"

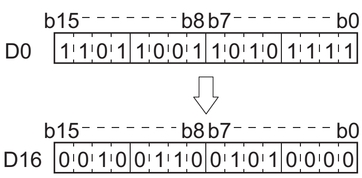


(3) The following program inverts the data at D0 when X3 is ON, and stores the result at D16.

[Ladder Mode]

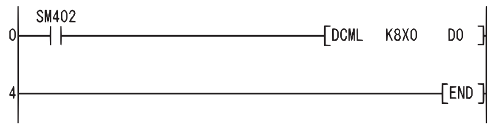


[Operation]



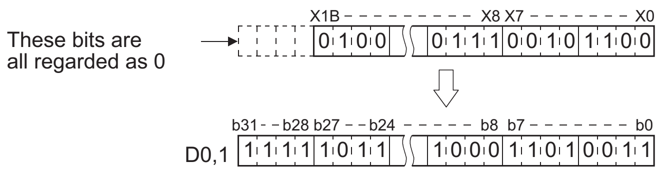
(4) The following program inverts the data at X0 to X1F, and transfers results to D0 and D1.

[Ladder Mode]



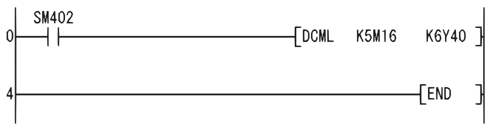
[Operation]

When "Number of bits of (S) < Number of bits of (D)"



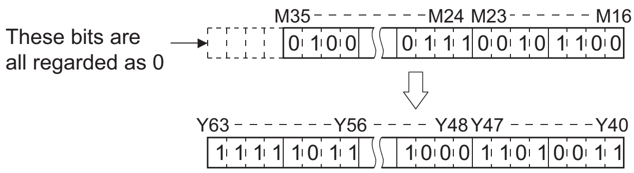
(5) The following program inverts the data at M16 to M35, and transfers it to Y40 to Y63.

[Ladder Mode]



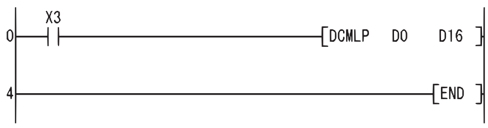
[Operation]

When "Number of bits of (S) < Number of bits of (D)"

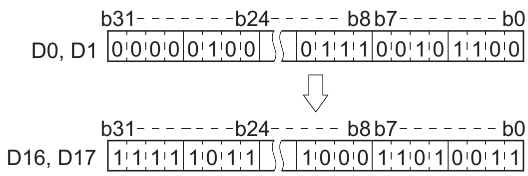


(6) Inverts the data at D0 and D1 when X3 is ON, and stores the result at D16 and D17.

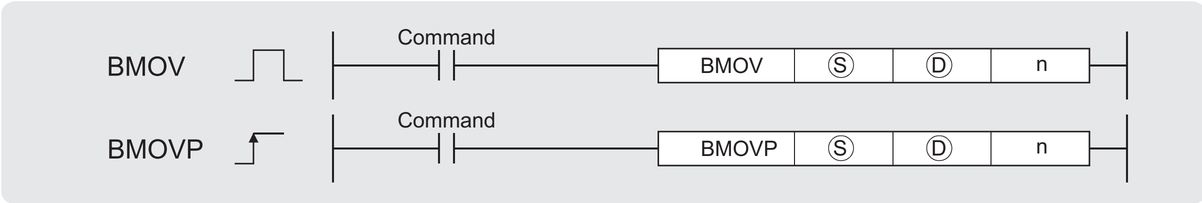
[Ladder Mode]



[Operation]



6.4.3 Block 16-bit data transfers (BMOV(P))



S : Start number of the devices where the data to be transferred is stored (BIN 16 bits)

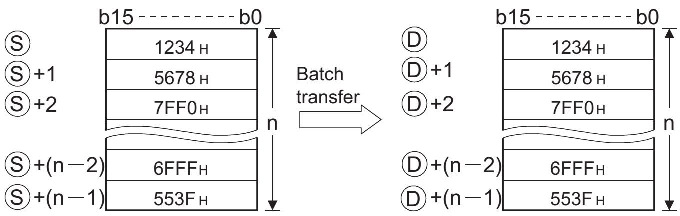
D : Start number of the devices of transfer destination (BIN 16 bits)

n : Number of data to be transferred (BIN 16 bits)

Set Data	Internal Devices		Constants K, H	Others
	Bit	Word		
S	○		—	—
D	○		—	—
n	○		○	—

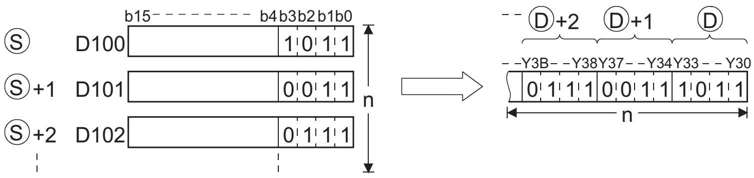
Function

(1) Transfers in batch 16-bit data n-points from the device designated by S to location n-points from the device designated by D.



(2) Transfers can be accomplished even in cases where there is an overlap between the source and destination device.
In the case of transmission to the smaller device number, transmission is from S; for transmission to the larger device number, transmission is from S + (n − 1).

(3) When S is a word device and D is a bit device, the target for the word device is the number of bits designated by the bit device digit specification.
If D is designated for K1Y30, the lower four bits of the word device designated by S are the target.



(4) If bit device has been designated for S and D, then S and D should always have the same number of digits.

Operation Error

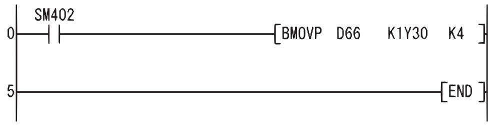
(1) In any of the following cases, an operation error occurs, the error flag (SM0) turns ON, and the corresponding error code is stored into SD0.

- The device range of n-points from Ⓢ or Ⓓ exceeds the corresponding device range.
(Error code: 4101)

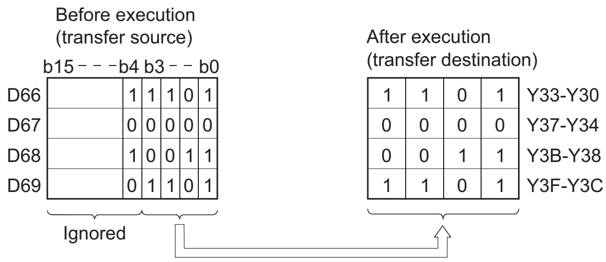
Program Example

(1) The following program outputs the lower 4 bits of data at D66 to D69 to Y30 to Y3F in 4-point units.

[Ladder Mode]

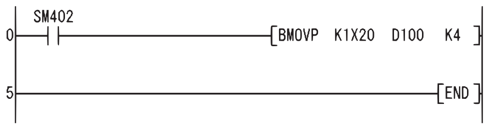


[Operation]

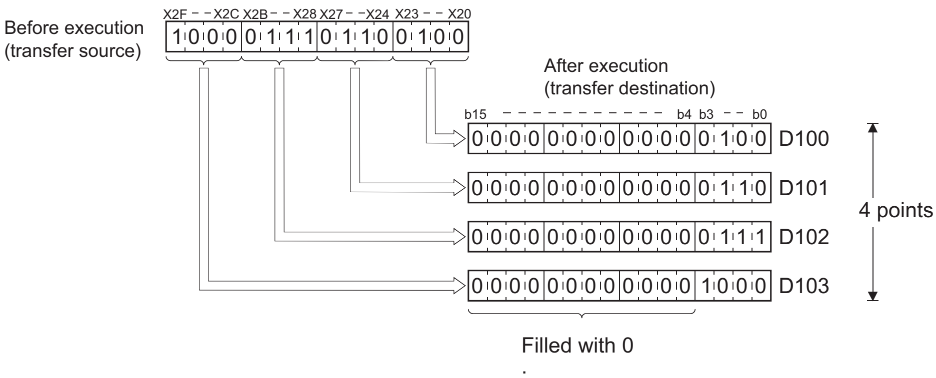


(2) The following program outputs the data at X20 to X2F to D100 to D103 in 4-point units.

[Ladder Mode]



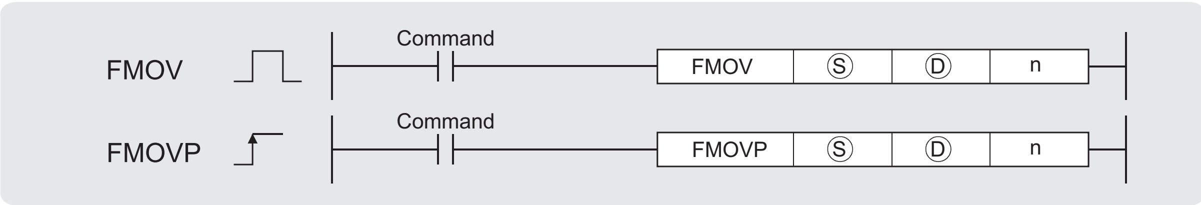
[Operation]



6 BASIC INSTRUCTIONS

BMOV(P)

6.4.4 Identical 16-bit data block transfers (FMOV(P))



Ⓢ : Data to be transferred or the start number of the devices where the data to be transferred is stored (BIN 16 bits)

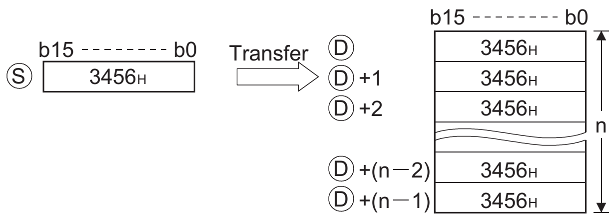
Ⓓ : Start number of the devices of transfer destination (BIN 16 bits)

n : Number of data to be transferred (BIN 16 bits)

Set Data	Internal Devices		Constants K, H	Others
	Bit	Word		
Ⓢ	○		○	—
Ⓓ	○		—	—
n	○		○	—

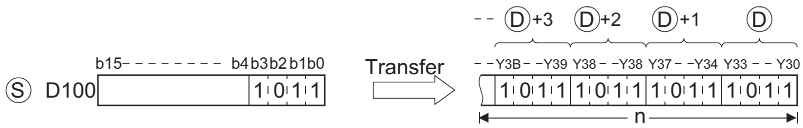
Function

(1) Transfers 16-bit data at the device designated by Ⓢ to n points of devices starting from the one designated by Ⓓ.



(2) When Ⓢ is a word device and Ⓓ is a bit device, the target for the word device Ⓢ is the number of bits designated by the bit device digit specification.

If Ⓓ is designated for K1Y30, the lower 4 bits of the word device designated by Ⓢ are the target.



(3) If bit device has been designated for Ⓢ and Ⓓ, then Ⓢ and Ⓓ should always have the same number of digits.

Operation Error

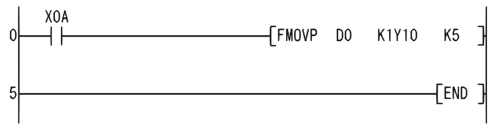
(1) In any of the following cases, an operation error occurs, the error flag (SM0) turns ON, and the corresponding error code is stored into SD0.

- The device range of n-points from Ⓓ or exceeds the corresponding device range.
(Error code: 4101)

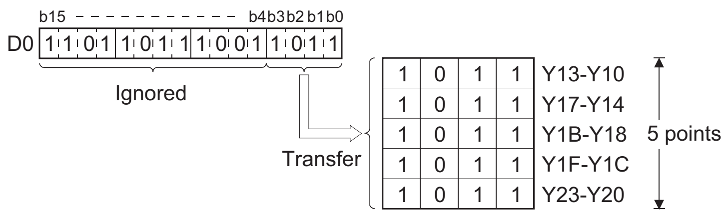
Program Example

(1) The following program outputs the lower 4 bits of D0 when XA goes ON to Y10 to Y23 in 4-bit units.

[Ladder Mode]

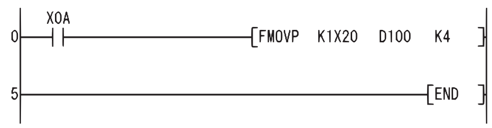


[Operation]

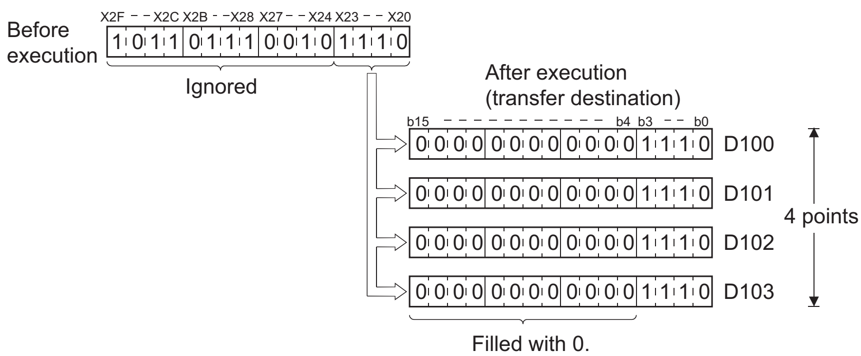


(2) The following program outputs the data at X20 through X23 to D100 through D103 when XA goes ON.

[Ladder Mode]



[Operation]



MEMO

7 APPLICATION INSTRUCTIONS

Category	Processing Details	Reference section
Logical operation instructions	Logical operations such as logical sum, logical product, etc.	7.1

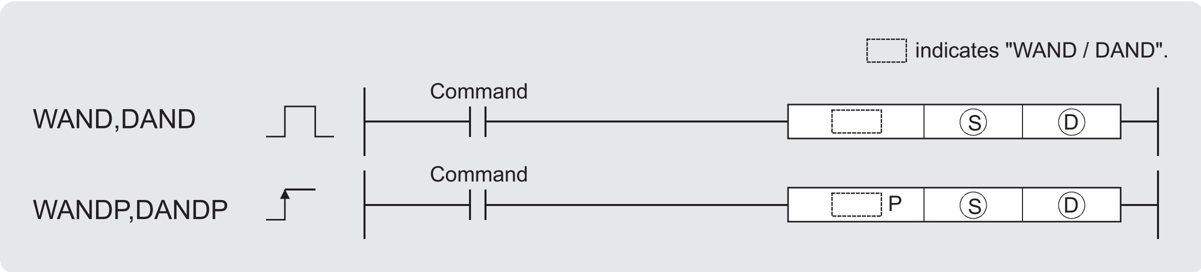
7.1 Logical Operation Instructions

(1) The logical operation instructions perform logical sum, logical product or other logical operations in 1-bit units.

Category	Processing Details	Formula for Operation	Example		
			A	B	Y
Logical product (AND)	Becomes 1 only when both input A and input B are 1; otherwise, is 0	$Y = A * B$	0	0	0
			0	1	0
			1	0	0
			1	1	1
Logical sum (OR)	Becomes 0 only when both input A and input B are 0; otherwise, is 1	$Y = A + B$	0	0	0
			0	1	1
			1	0	1
			1	1	1
Exclusive OR (XOR)	Becomes 0 if input A and input B are equal; otherwise, is 1	$Y = \overline{A} * B + A * \overline{B}$	0	0	0
			0	1	1
			1	0	1
			1	1	0
NON exclusive logical sum (XNR)	Becomes 1 if input A and input B are equal; otherwise, is 0	$Y = (\overline{A} + B)(A + \overline{B})$	0	0	1
			0	1	0
			1	0	0
			1	1	1

7.1.1 Logical products with 16-bit and 32-bit data (WAND(P), DAND(P))

1 When two data are set (D ∧ S → D, (D + 1, D) ∧ (S + 1, S) → (D + 1, D))



S : Data for a logical product operation or the start number of the devices where the data is stored (BIN 16/32 bits)

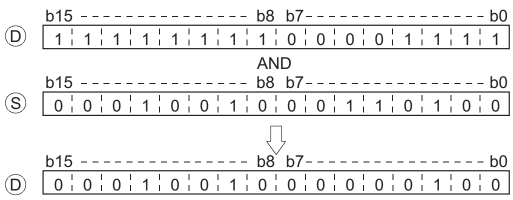
D : Start number of the devices where the logical product operation result will be stored (BIN 16/32 bits)

Set Data	Internal Devices		Constants K, H	Others
	Bit	Word		
S	○		○	—
D	○		—	—

Function

WAND

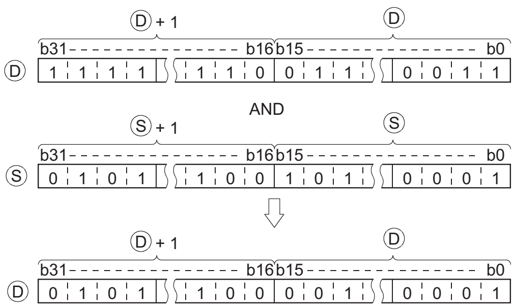
(1) A logical product operation is conducted for each bit of the 16-bit data of the device designated at D and the 16-bit data of the device designated at S, and the results are stored in the device designated at D.



(2) When bit devices are designated, the bit devices below the points designated as digits are regarded as "0" in the operation. (See Program Example (2))

DAND

(1) Conducts a logical product operation on each bit of the 32-bit data for the device designated by D and the 32-bit data for the device designated by S, and stores the results at the device designated by D.



(2) When bit devices are designated, the bit devices below the points designated as digits are regarded as "0" in the operation. (See Program Example (2))

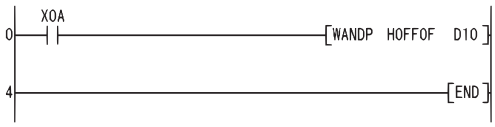
Operation Error

(1) There are no operation errors associated with the WAND(P) or DAND(P) instruction.

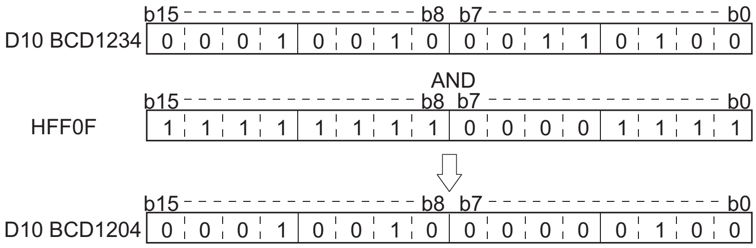
Program Example

(1) The following program masks the digit in the 10s place of the 4-digit BCD value at D10 (second digit from the end) to 0 when XA is turned ON.

[Ladder Mode]

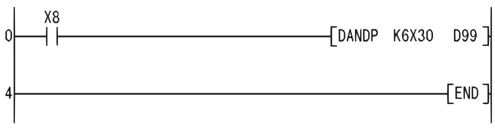


[Operation]



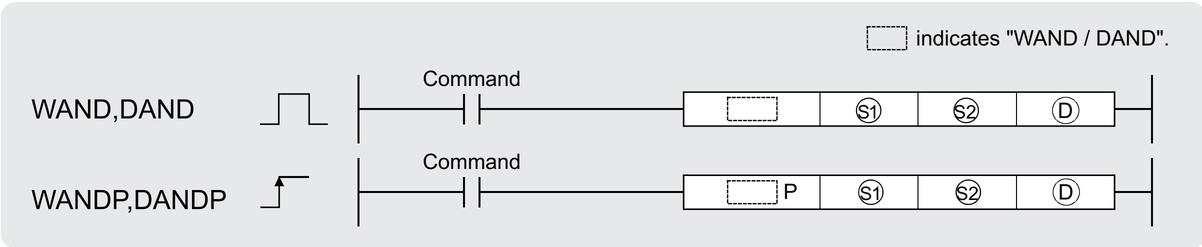
(2) The following program performs a logical product operation on the data at D99 and D100, and the 24-bit data between X30 and X47 when X8 is ON, and stores the results at D99 and D100.

[Ladder Mode]



[Operation]

2 When three data are set (S1 ∧ S2 → D, (S1 + 1, S1) ∧ (S2+1, S2) → (D+1, D))



S1, S2 : Data for a logical product operation or the start number of the devices where the data is stored (BIN 16/32 bits)

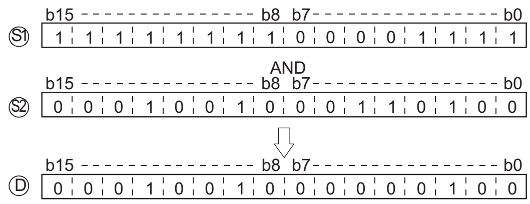
D : Start number of the devices where the logical product operation result will be stored (BIN 16/32 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
S1	○		○	—
S2	○		○	—
D	○		—	—

Function

WAND

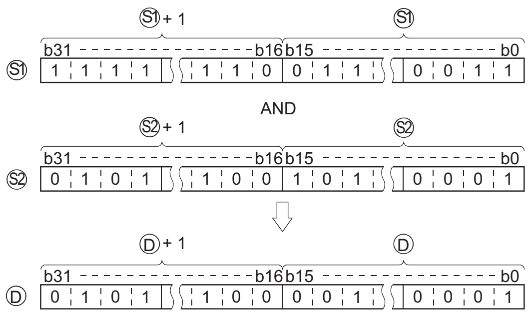
(1) A logical product operation is conducted for each bit of the 16-bit data of the device designated at S1 and the 16-bit data of the device designated at S2, and the results are stored in the device designated at D.



(2) For bit devices, the bit devices below the points designated by digit specification are regarded as "0" in the operation. (See Program Examples (1) and (2))

DAND

(1) Conducts a logical product operation on each bit of the 32-bit data for the device designated by S1 and the 32-bit data for the device designated by S2, and stores the results at the device designated by D.



(2) For bit devices, the bit devices below the points designated by digit specification are regarded as "0" in the operation. (See Program Example (3))

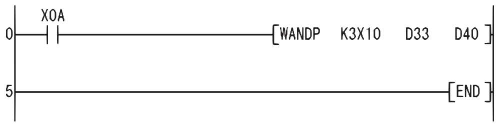
Operation Error

(1) There are no operation errors associated with the WAND(P) or DAND(P) instruction.

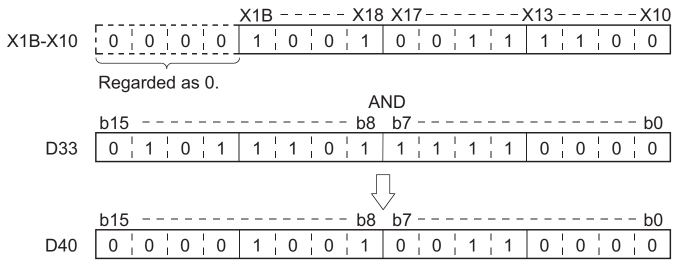
Program Example

(1) The following program performs a logical product operation on the data from X10 to X1B and the data at D33 when XA is ON, and stores the results at D40.

[Ladder Mode]

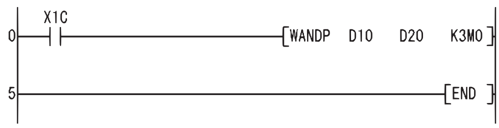


[Operation]

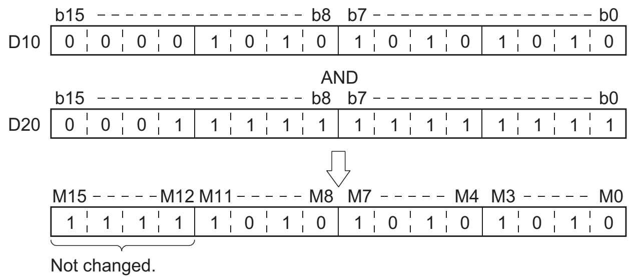


(2) The following program performs a logical product operation on the data at D10 and at D20 when X1C is ON, and stores the results from M0 to M11.

[Ladder Mode]

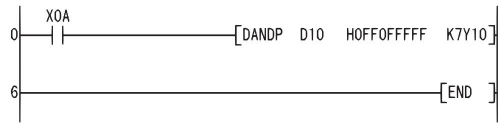


[Operation]

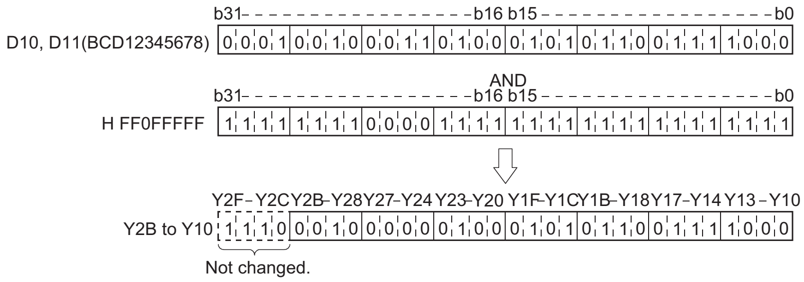


(3) The following program masks the digit in the hundred-thousands place of the 8-digit BCD value at D10 and D11 (sixth digit from the end) to 0 when XA is ON, and outputs the results to from Y10 to Y2B.

[Ladder Mode]

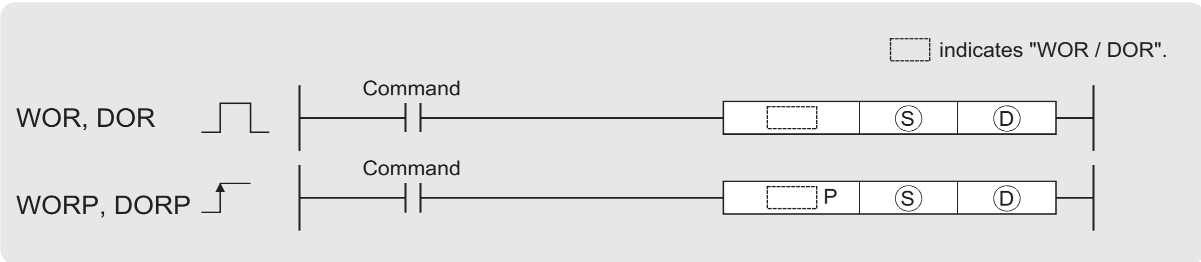


[Operation]



7.1.2 Logical sums of 16-bit and 32-bit data (WOR(P), DOR(P))

1 When two data are set (D ⩡ S → D, (D+1, D) ⩡ (S+1, S) → (D+1, D))



S : Data for a logical sum operation or start number of the devices where the data is stored (BIN 16/32 bits)

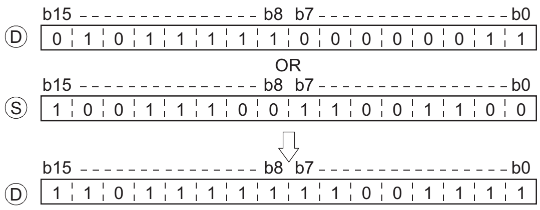
D : Start number of the devices where the logical sum operation result will be stored (BIN 16/32 bits)

Set Data	Internal Devices		Constants K, H	Others
	Bit	Word		
S	○		○	—
D	○		—	—

Function

WOR

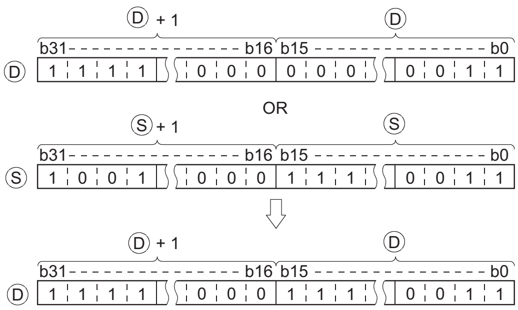
(1) Conducts a logical sum operation on each bit of the 16-bit data of the device designated by D and the 16-bit data of the device designated by S, and stores the results at the device designated by D.



(2) For bit devices, the bit devices below the points designated by digit specification are regarded as "0" in the operation.

DOR

(1) Conducts a logical sum operation on each bit of the 32-bit data of the device designated by D and the 32-bit data of the device designated by S, and stores the results at the device designated by D.



(2) For bit devices, the bit devices below the points designated by digit specification are regarded as "0" in the operation.

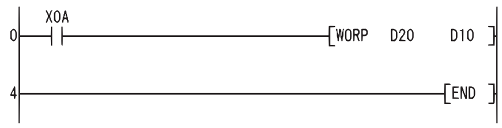
Operation Error

(1) There are no operation errors associated with the WOR(P) or DOR(P) instructions.

Program Example

(1) The following program performs a logical sum operation on the data at D10 and D20 when XA goes ON, and stores the results at D10.

[Ladder Mode]



[Operation]

	b15							b8	b7							b0
D10	1	1	0	0	1	1	0	0	1	1	1	1	0	0	0	0

OR

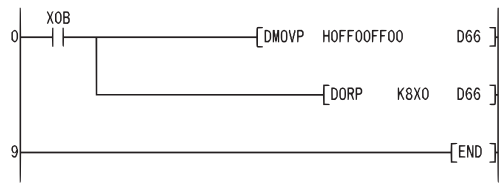
	b15							b8	b7							b0
D20	0	0	0	0	1	0	1	1	1	0	0	1	0	0	0	1

⇩

	b15							b8	b7							b0
D10	1	1	0	0	1	1	1	1	1	1	1	1	0	0	0	1

(2) The following program performs a logical sum operation on the 32-bit data from X0 to X1F, and on the hexadecimal value FF00FF00$_H$ when XB goes ON, and stores the results at D66 and D67.

[Ladder Mode]



[Operation]

	S + 1								S					
	X1F		X1C				X10	XF			X3			X0
X1F～X0	1	0	1	0	0	0	0	0	0	0	1	0	1	0

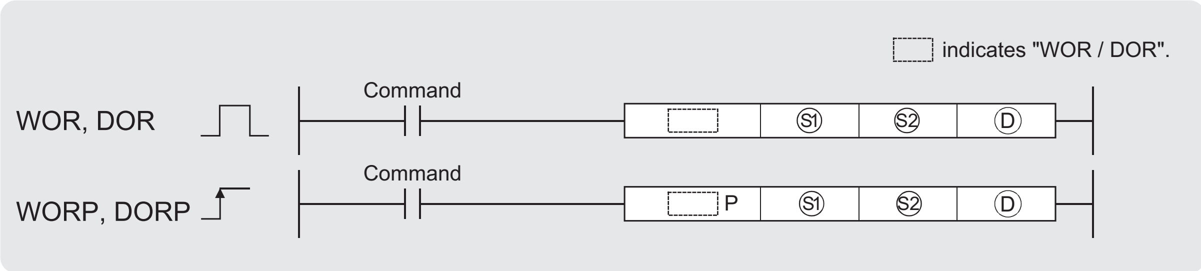
OR

	D + 1								D					
	b31						b16	b15						b0
FF00FF00H	1	1	1	1	0	0	0	1	1	1	0	0	0	0

⇩

	D + 1								D					
	b31						b16	b15						b0
D67,D66	1	1	1	1	0	0	0	1	1	1	1	0	1	0

2 When three data are set (S1 ∨ S2 → D, (S1+1, S1) ∨ (S2+1, S2) → (D+1, D))



S1, S2 : Data for a logical sum operation or start number of the devices where the data is stored (BIN 16/32 bits)

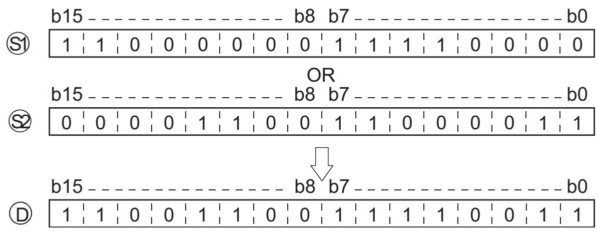
D : Start number of the devices where the logical sum operation result will be stored (BIN 16/32 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
S1	○		○	—
S2	○		○	—
D	○		—	—

Function

WOR

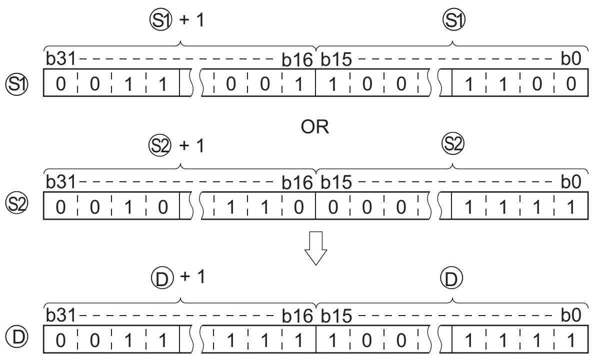
(1) Conducts a logical sum operation on each bit of the 16-bit data of the device designated by S1 and the 16-bit data of the device designated by S2, and stores the results at the device designated by D.



(2) For bit devices, the bit devices below the points designated by digit specification are regarded as "0" in the operation. (See Program Example (1))

DOR

(1) Conducts a logical sum operation on each bit of the 32-bit data of the device designated by S1 and the 32-bit data of the device designated by S2, and stores the results at the device designated by D.



(2) When bit devices are designated, the bit devices below the points designated as digits are regarded as "0" in the operation. (See Program Example (2))

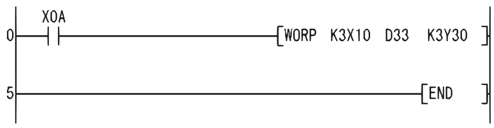
Operation Error

(1) There are no operation errors associated with the WOR(P) or DOR(P) instructions.

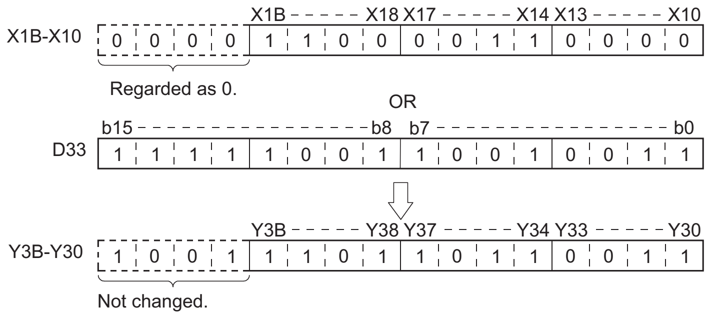
Program Example

(1) The following program performs a logical sum operation on the data from X10 to X1B, and the data at D33, and stores the result at Y30 to Y3B when XA is ON.

[Ladder Mode]

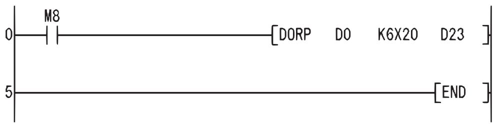


[Operation]

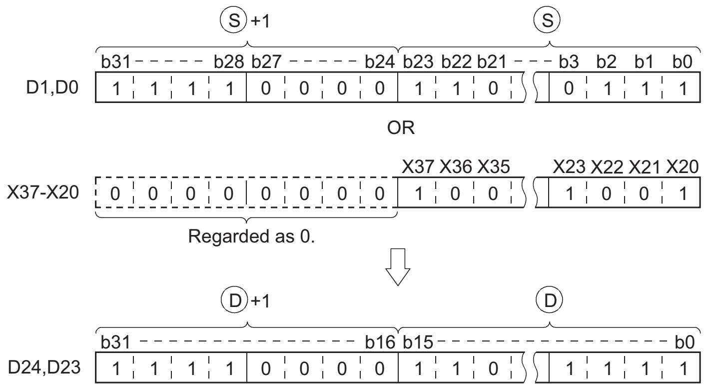


(2) The following program performs a logical sum operation on the 32-bit data at D0 and D1, and the 24-bit data from X20 to X37, and stores the results at D23 and D24 when M8 is ON.

[Ladder Mode]

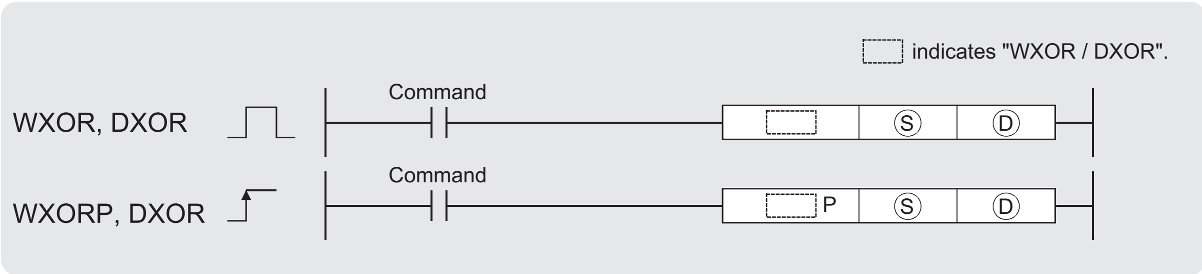


[Operation]



7.1.3 16-bit and 32-bit exclusive OR operations (WXOR(P), DXOR(P))

1 When two data are set (D ⊻ S → D, (D+1, D) ⊻ (S+1, S) → (D+1, D))



S : Data for an exclusive OR operation or start number of the devices where the data is stored (BIN 16/32 bits)

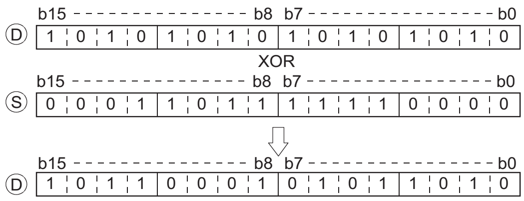
D : Start number of the devices where the exclusive OR operation result will be stored (BIN 16/32 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
S	○		○	—
D	○		—	—

Function

WXOR

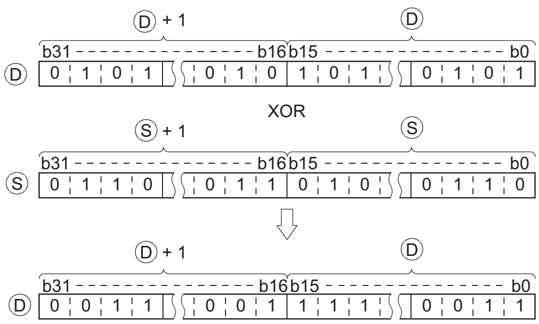
(1) Conducts an exclusive OR operation on each bit of the 16-bit data of the device designated by D and the 16-bit data of the device designated by S, and stores the results at the device designated by D.



(2) For bit devices, the bit devices below the points designated by digit specification are regarded as "0" in the operation.

DXOR

(1) Conducts an exclusive OR operation on each bit of the 32-bit data of the device designated by D and the 32-bit data of the device designated by S, and stores the results at the device designated by D.



(2) For bit devices, the bit devices below the points designated by digit specification are regarded as "0" in the operation.

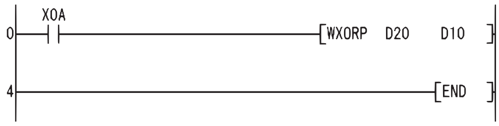
Operation Error

(1) There are no operation errors associated with the WXOR(P) or DXOR(P) instructions.

Program Example

(1) The following program performs an exclusive OR operation on the data at D10 and D20 when XA is ON, and stores the result at D10.

[Ladder Mode]

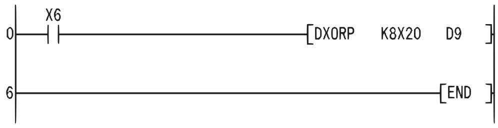


[Operation]

	b15 – b12	b11 – b8	b7 – b4	b3 – b0
D10	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1
XOR				
D20	0 0 1 1	1 0 0 1	0 0 1 1	1 0 0 1
⇩				
D10	0 1 1 0	1 1 0 0	0 1 1 0	1 1 0 0

(2) The following program compares the bit pattern of the 32-bit data from X20 to X3F with the bit pattern of the data at D9 and D10 when X6 is ON.

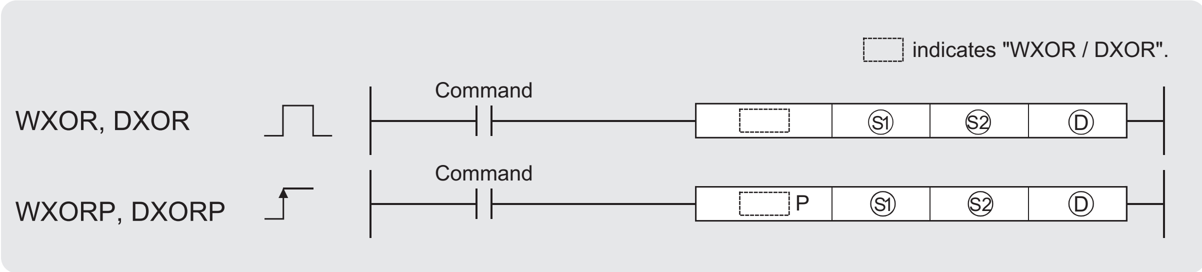
[Ladder Mode]



[Operation]

	S + 1				S			
	X3F – X3C	X3B – X38	X37 – X34	X33 – X30	X2F – X2C	X2B – X28	X27 – X24	X23 – X20
X3F～X20	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1
XOR	D + 1 (b31 – b16)				D (b15 – b0)			
D10,D9	1 0 1 1	1 0 0 1	1 1 0 0	0 0 0 1	1 0 1 1	1 1 0 1	1 0 1 0	1 1 0 1
⇩	D + 1 (b31 – b16)				D (b15 – b0)			
D10,D9	1 1 1 0	1 1 0 0	1 0 0 1	0 1 0 0	1 1 1 0	1 0 0 0	1 1 1 1	1 0 0 0

2 When three data are set (S1 ⩡ S2 → D (S1+1, S1) ⩡ (S2+1, S2) → (D+1, D))



S1, S2: Data for an exclusive OR operation or start number of the devices where the data is stored (BIN 16/32 bits)

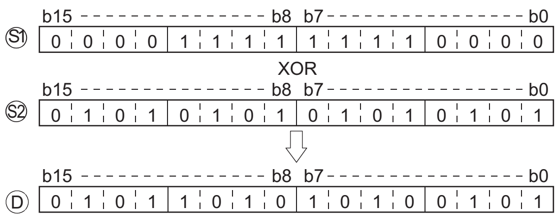
D: Start number of the devices where the exclusive OR operation result will be stored (BIN 16/32 bits)

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
S1	○		○	—
S2	○		○	—
D	○		—	—

Function

WXOR

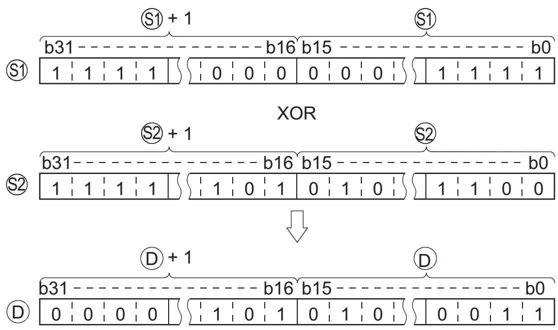
(1) Conducts an exclusive OR operation on each bit of the 16-bit data of the device designated by S1 and the 16-bit data of the device designated by S2, and stores the results at the device designated by D.



(2) For bit devices, the bit devices below the points designated by digit specification are regarded as "0" in the operation. (See Program Example (1))

DXOR

(1) Conducts an exclusive OR operation on each bit of the 32-bit data of the device designated by S1 and the 32-bit data of the device designated by S2, and stores the results at the device designated by D.



(2) For bit devices, the bit devices below the points designated by digit specification are regarded as "0" in the operation.

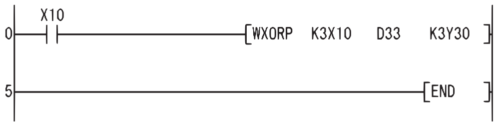
Operation Error

(1) There are no operation errors associated with the WXOR(P) or DXOR(P) instructions.

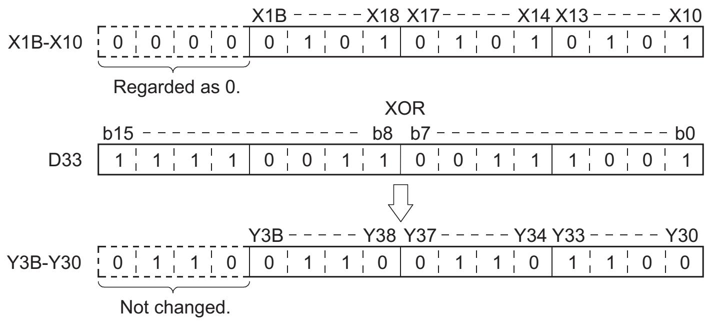
Program Example

(1) The following program conducts an exclusive OR operation on the data from X10 to X1B and the data at D33 when X10 is ON, and outputs the result to from Y30 to Y3B.

[Ladder Mode]

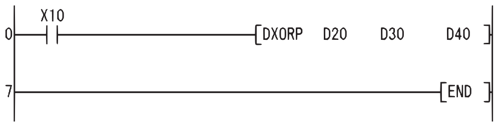


[Operation]

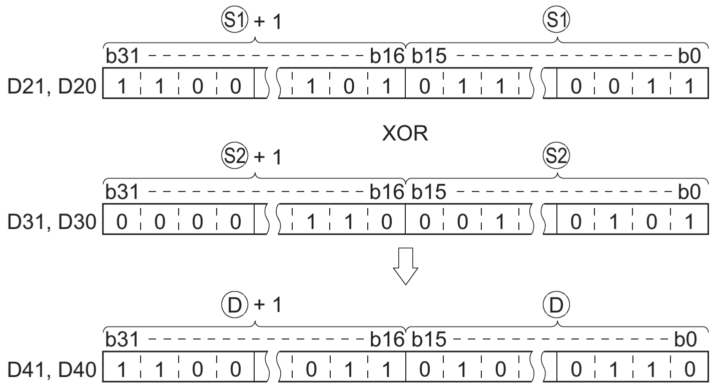


(2) The following program conducts an exclusive OR operation on the data at D20 and D21, and the data at D30 and D31 when X10 goes ON, and stores the results at D40 and D41.

[Ladder Mode]

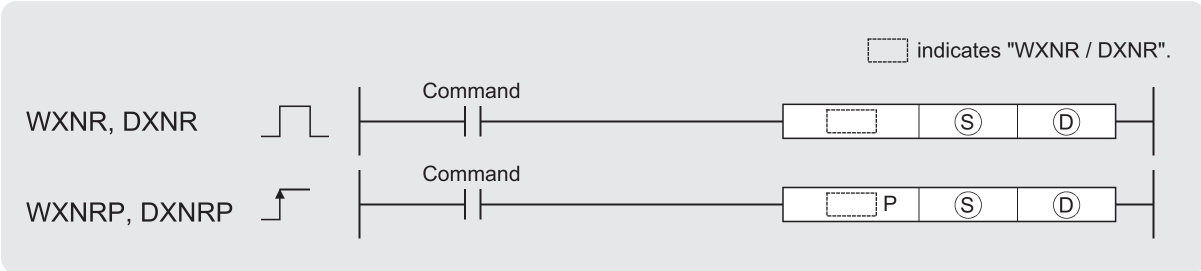


[Operation]



7.1.4 16-bit and 32-bit data exclusive NOR operations (WXNR(P), DXNR(P))

1 When two data are set $(\overline{D \veebar S} \rightarrow D,\ \overline{(D+1, D) \veebar (S+1, S)} \rightarrow (D+1, D))$



S : Data for an exclusive NOR operation or start number of the devices where the data is stored (BIN 16/32 bits)

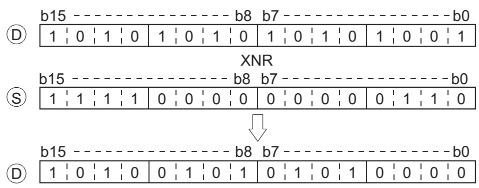
D : Start number of the devices where the exclusive NOR operation result will be stored (BIN 16/32 bits)

Set Data	Internal Devices		Constants K, H	Others
	Bit	Word		
S	○		○	—
D	○		—	—

Function

WXNR

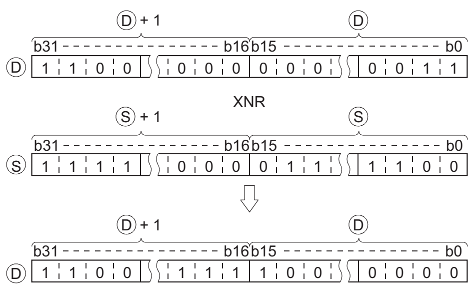
(1) Conducts an exclusive NOR operation on the 16-bit data of the device designated by D and the 16-bit data of the device designated by S, and stores the results at the device designated by D.



(2) For bit devices, the bit devices below the points designated by digit specification are regarded as "0" in the operation.

DXNR

(1) Conducts an exclusive NOR operation on the 32-bit data of the device designated by D and the 32-bit data of the device designated by S, and stores the results at the device designated by D.



(2) For bit devices, the bit devices below the points designated by digit specification are regarded as "0" in the operation.

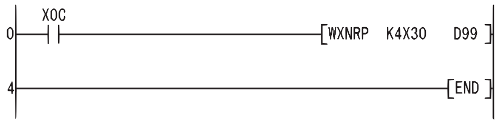
Operation Error

(1) There are no operation errors associated with the WXNR(P) or DXNR(P) instructions.

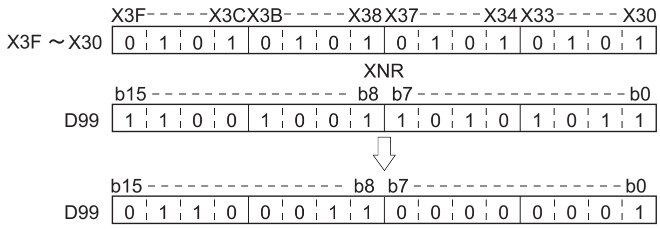
Program Example

(1) The following program compares the bit pattern of the 16-bit data from X30 to X3F with the bit pattern of the 16-bit data at D99 when X6 is ON

[Ladder Mode]

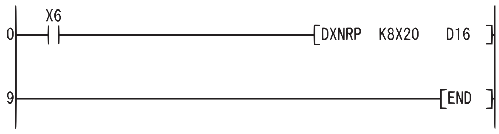


[Operation]

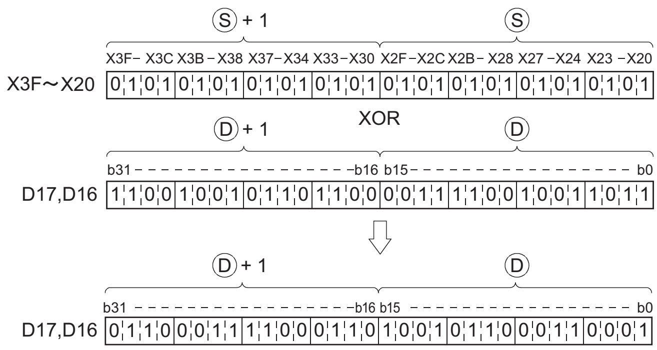


(2) The following program compares the bit pattern of the 32-bit data from X20 to X3F with the bit pattern of the data at D16 and D17 when X6 is ON.

[Ladder Mode]



[Operation]

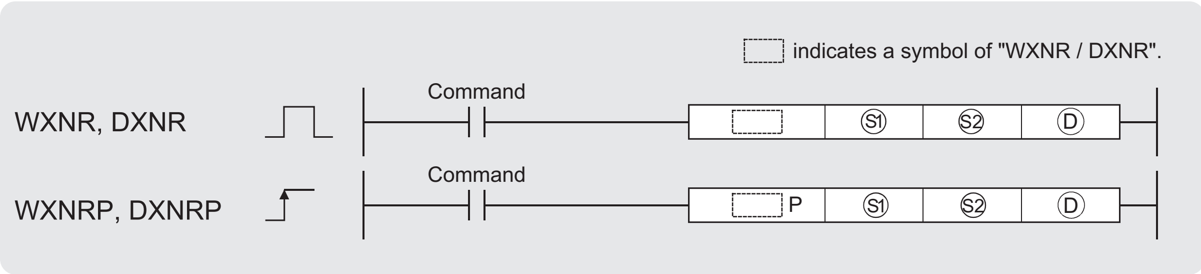


7

APPLICATION INSTRUCTIONS

WXNR(P), DXNR(P)

2 When three data are set $(\overline{(S1) \veebar (S2)} \rightarrow (D), \overline{((S1)+1, (S1)) \veebar ((S2)+1, (S2))} \rightarrow ((D)+1, (D)))$



S1, S2: Data for an exclusive NOR operation or start number of the devices where the data is stored (BIN 16/32 bits)

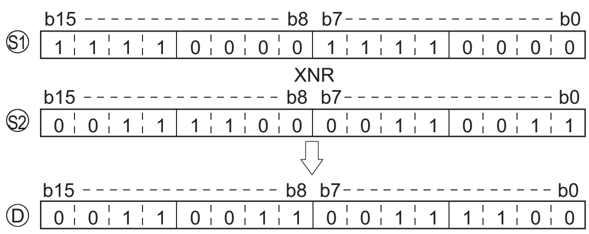
D: Start number of the devices where the exclusive NOR operation result will be stored (BIN 16/32 bits)

Set Data	Internal Devices		Constants K, H	Others
	Bit	Word		
S1	○		○	—
S2	○		○	—
D	○		—	—

☆ Function

WXNR

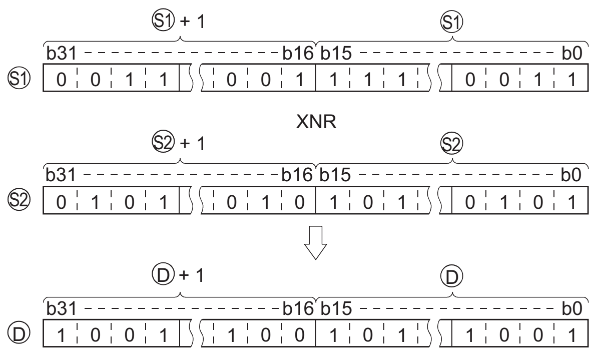
(1) Conducts an exclusive NOR operation on the 16-bit data of the device designated by S1 and the 16-bit data of the device designated by S2, and stores the results at the device designated by D.



(2) For bit devices, the bit devices below the points designated by digit specification are regarded as "0" in the operation.

DXNR

(1) Conducts an exclusive NOR operation on the 32-bit data of the device designated by S1 and the 32-bit data of the device designated by S2, and stores the results at the device designated by D.



(2) For bit devices, the bit devices below the points designated by digit specification are regarded as "0" in the operation.

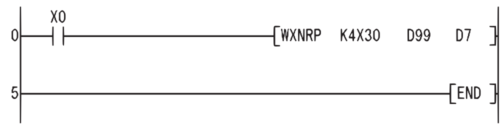
Operation Error

(1) There are no operation errors associated with the WXNR(P) or DXNR(P) instructions.

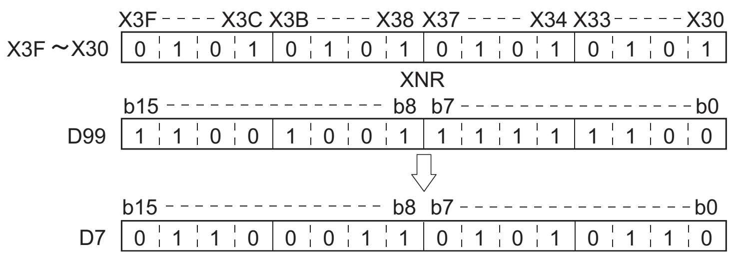
Program Example

(1) The following program performs an exclusive NOR operation on the 16-bit data from X30 to X3F and the data at D99 when X0 is turned ON, and stores the results to D7.

[Ladder Mode]

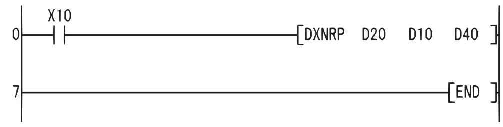


[Operation]

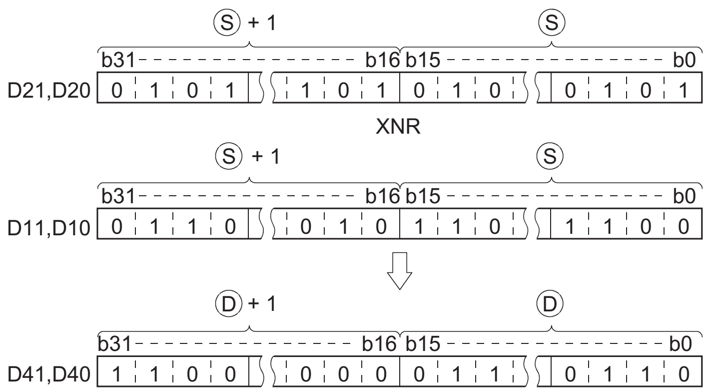


(2) The following program performs an exclusive NOR operation on the 32-bit data at D20 and D21 and the data at D10 and D11 when X10 is turned ON, and stores the result to D40 and D41.

[Ladder Mode]



[Operation]

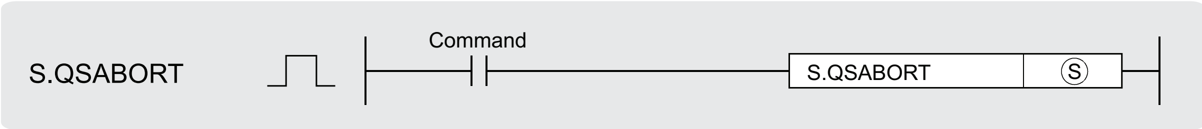


MEMO

8 QSCPU DEDICATED INSTRUCTIONS

Category	Processing Details	Reference section
Forced control stop instruction	Forced control stop	8.1

8.1 Forced Control Stop Instruction (S.QSABORT)



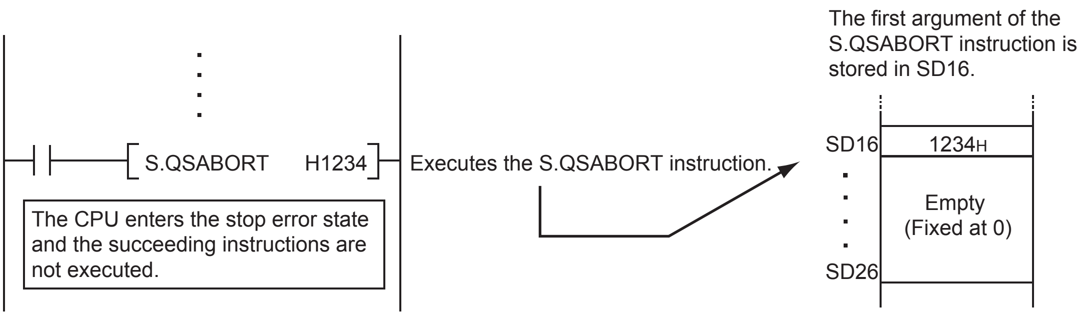
Ⓢ : Data to be stored in SD16 as the abort code, or the number of the device where the data is stored (BIN 16 bits).

Set Data	Internal Devices Bit	Internal Devices Word	Constants K, H	Others
Ⓢ	○		○	—

☆ Function

(1) Execution of the S.QSABORT instruction stops program execution and brings the safety CPU module into the stop error status (error code: 4700).*1
SM0 (diagnostics error) turns ON to store the error information in SD0 to SD26. In this case, however, SM1 (self-diagnostics error) does not turn ON.

*1: For the CPU operation at a stop error, refer to the following manual
• QSCPU User's Manual (Function Explanation, Program Fundamentals)



(2) Details of the program error are stored in the common information (SD5 to SD15).

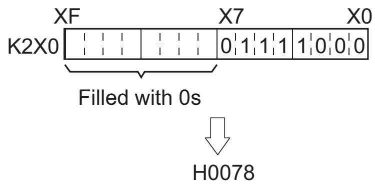
Device	Meaning	
SD5	File name (ASCII code: 8 characters)	
SD6		
SD7		
SD8		
SD9	Extension (ASCII code: 3 characters)	2EH(.)
SD10		
SD11	Fixed to 0	
SD12	0 (Block No.)	
SD13	0 (Step No. / Shift conditions)	
SD14	Sequence step No. (L) where the S.QSABORT instruction was executed	
SD15	Sequence step No. (H) where the S.QSABORT instruction was executed	

(3) Program abort information is stored in the individual information (SD16 to SD26).

Device	Meaning
SD16	Abort code (The first argument of the S.QSABORT instruction is stored.)
SD17	Empty (Fixed at 0)
SD18	
SD19	
SD20	
SD21	
SD22	
SD23	
SD24	
SD25	
SD26	

(4) If the abort code is designated using digit designation of a bit device, the data of the designated digit in the designated bit device is obtained from the device memory as abort code. If the designated number of bits is less than 16 bits, empty bits are filled with 0.

Example



Operation Error

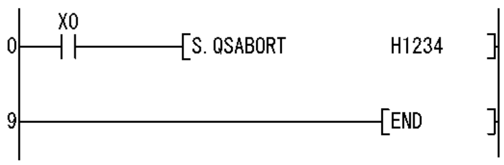
(1) In any of the following cases, an operation error occurs, the error flag (SM0) turns ON, and the corresponding error code is stored into SD0.

- The designated instruction name is incorrect. (Error code: 4002)
- The number of the arguments used in the instruction is incorrect. (Error code: 4003)
- A device that cannot be used in an argument is designated. (Error code: 4004)

Program Example

(1) The program below shows that, when X0 turns ON, the CPU module enters the stop error state and the abort code is stored in SD16.

[Ladder Mode]



MEMO

9 ERROR CODE

9.1 Error Code List

If an error occurs under one of the following conditions, the QS series CPU module uses the self-diagnostic function to display error information (LED indication) and stores it in the special relay (SM) and special register (SD):

• the programmable controller is powered on

• the programmable controller is reset

• the programmable controller is switched from STOP to RUN

• the programmable controller is running

If an error occurs when a communication request is issued from GX Developer, intelligent function module or network system to the CPU module, the CPU module returns the error code (4000_H to $4FFF_H$) to the request source.

The following describes the description of errors which occur in the CPU module and the corrective actions for the errors.

(1) How to read the error code list
The following describes how to read Section 9.1.3 Error code list (1000 to 1999) to Section 9.1.8 Error code list (8000 to 9000).

(a) Error code, common information and individual information
Alphanumeric characters in the parentheses of the titles indicate the special register numbers where each information is stored.

(b) Compatible CPU
QS: Compatible with the QSCPU.

9.1.1 Error codes

Errors are detected by the self diagnostic function of the CPU module or detected during communication with the CPU module.

The relation between the error detection pattern, error detection location and error code is shown in the following table.

Table9.1 The relation between the error detection pattern, error detection location and error code

Error detection pattern	Error detection location	Error code	Reference
Detection by the self diagnostics function of CPU module	CPU module	1000 to 9000*1	Section 9.1.3 to 9.1.8
Detection at communication with CPU module	CPU module	4000_H to $4FFF_H$	QSCPU User's Manual (Hardware Design, Maintenance and Inspection)
	CC-Link Safety master module	$B000_H$ to $BFFF_H$	CC-Link Safety System Master Module User's Manual
	Ethernet module	$C000_H$ to $CFFF_H$	Ethernet Interface Module UserAfs Manual
	CC-Link IE Field Network master/local module (with safety functions)	$D000_H$ to $DFFF_H$	MELSEC-QS CC-Link IE Field Network Master/Local Module User's Manual
	CC-Link IE Controller Network module	$E000_H$ to $EFFF_H$	CC-Link IE Controller Network Reference Manual
	MELSECNET/H module	$F000_H$ to $FFFF_H$	MELSECNET/H Network System Reference Manual

*1: CPU module error codes are classified into minor, moderate, major errors as shown below.

•Minor error:Errors that may allow the CPU module to continue the operation, e.g., battery error. (Error code: 1300 to 9000)

•Moderate error:Errors that may cause the CPU module to stop the operation, e.g., WDT error. (Error code: 1300 to 9000)

•Major error:Errors that may cause the CPU module to stop the operation, e.g., RAM error. (Error code: 1000 to 1299)

"Errors that may allow the CPU module to continue the operation" and "Errors that may cause the CPU module to stop the operation" can be distinguished using "Operating Statuses of CPU" of Section 9.1.3 to 9.1.8 Error code list.

9.1.2 Reading an error code

If an error occurs, information such as the error code and error message for troubleshooting can be read using GX Developer.
To read error codes, follow the instructions described below.

1) Start GX Developer.
2) Connect the CPU module to the personal computer that started GX Developer.
3) On GX Developer, choose the [Online] → [Read from PLC] menu and read the project from the CPU module.
4) Choose the [Diagnostic] → [PLC diagnostic] menu.
5) Click the "Current error" button in the PLC diagnostic dialog box to display the error code and error message.
6) Choose the [Help] → [CPU error] menu and check details of the corresponding error code.

For details, refer to the following.
GX Developer Operating Manual

9.1.3 Error code list (1000 to 1999)

The following shows the error messages from the error code 1000 to 1999, the contents and causes of the errors, and the corrective actions for the errors.

Table9.2 Error code

<table>
<tr><th rowspan="2">Error Code (SD0)</th><th rowspan="2">Error Message</th><th rowspan="2">Common Information (SD5 to 15)</th><th rowspan="2">Individual Information (SD16 to 26)</th><th colspan="2">LED Status</th><th rowspan="2">CPU Operation Status</th><th rowspan="2">Diagnostic Timing</th><th rowspan="2"></th></tr>
<tr><th>RUN</th><th>ERROR</th></tr>
<tr><td>1000</td><td rowspan="7">MAIN CPU DOWN</td><td rowspan="7">–</td><td rowspan="7">–</td><td rowspan="7">Off</td><td>On/ Flash</td><td rowspan="7">Stop</td><td rowspan="7">Always</td><td rowspan="7"></td></tr>
<tr><td>1001</td><td>On</td></tr>
<tr><td>1002</td><td rowspan="5">Flash</td></tr>
<tr><td>1003</td></tr>
<tr><td>1004</td></tr>
<tr><td>1006</td></tr>
<tr><td>1009</td></tr>
<tr><td>1010</td><td>END NOT EXECUTE</td><td>–</td><td>–</td><td>Off</td><td>Flash</td><td>Stop</td><td>When an END instruction executed.</td><td></td></tr>
<tr><td>1030</td><td rowspan="2">MAIN CPU DOWN</td><td rowspan="2">–</td><td rowspan="2">Error information</td><td rowspan="2">Off</td><td rowspan="2">Flash</td><td rowspan="2">Stop</td><td rowspan="2">Always</td><td rowspan="2"></td></tr>
<tr><td>1031</td></tr>
</table>

*1 BAT.ALM LED is displayed at BATTERY ERROR.

Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
1000	Runaway or failure of the main CPU • Malfunctioning due to noise or other reason • Hardware fault	• Take noise reduction measures. • Reset the CPU module and RUN it again.If the same error is displayed again, this suggests a CPU module hardware fault.(Contact your local Mitsubishi representative.)	QS
1001			
1002			
1003			
1004			
1006			
1009	• The power supply module detected an incorrect power waveform. • A failure is detected on the power supply module, CPU module, or base unit.	• Correct the power waveform applied to the power supply module. • Reset the CPU module and RUN it again.If the same error is detected again, it is considered that the power supply module, CPU module, or base unit is failure. (Contact your local Mitsubishi representative.)	
1010	Entire program was executed without the execution of an END instruction. • When the END instruction is executed it is read as another instruction code, e.g. due to noise. • The END instruction has been changed to another instruction code somehow.	• Take noise reduction measures. • Reset the CPU module and RUN it again. If the same error is displayed again, this suggests a CPU module hardware fault. (Contact your local Mitsubishi representative.)	QS
1030	Run mode suspended or failure of main CPU • Malfunctioning due to noise or other reason • Hardware fault	• Take noise reduction measures. • Reset the CPU module and RUN it again. If the same error is displayed again, this suggests a CPU module hardware fault. (Contact your local Mitsubishi representative.)	QS
1031			

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status RUN	LED Status ERROR	CPU Operation Status	Diagnostic Timing
1131	RAM ERROR	–	Error information	Off	Flash	Stop	At power-ON/ At reset
1132							
1133							
1136							
1137							
1141							Always
1142							
1143							
1146							
1210	OPERATION CIRCUIT ERROR	–	Error information	Off	Flash	Stop	At power-ON/ At reset/When an END instruction executed
1311	I/O INTERRUPT ERROR	–	–	Off	Flash	Stop	During interrupt
1401	INTELLIGENT FUNCTION MODULE DOWN	Module No.	–	Off	Flash	Stop	At power ON/ At reset/When intelligent function module is accessed.
1403	INTELLIGENT FUNCTION MODULE DOWN	Module No.	–	Off	Flash	Stop	When an END instruction executed.
1404	INTELLIGENT FUNCTION MODULE DOWN	Module No.	–	Off	Flash	Stop	When an END instruction executed.

*1 BAT.ALM LED is displayed at BATTERY ERROR.

Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
1131 1132 1133 1136 1137 1141 1142 1143 1146	A fault was detected in the internal memory of the CPU module.	Hardware error of the CPU module. (Please consult your local Mitsubishi Service or representative.)	QS
1210	The operation circuit for sequence processing in the CPU module does not operate normally.	Hardware error of the CPU module. (Please consult your local Mitsubishi Service or representative.)	QS
1311	An interrupt request from the module where interrupt pointer setting has not been made in the PLC parameter dialog box was detected.	Hardware error of either of the CPU module or base unit. (Please consult your local Mitsubishi Service or representative.)	QS
1401	• There was no response from the intelligent function module in the initial processing. • The size of the buffer memory of the intelligent function module is invalid.	Hardware error of the intelligent function module, CPU module or base unit is expecting a hardware fault. (Please consult your local Mitsubishi Service or representative.)	QS
1403	• The hardware test of the module installed in the slot indicated by module number has completed. • There was no response from the intelligent function module when the END instruction is executed. • An error is detected at the intelligent function module. • The intelligent function module being accessed is broken down.	• Confirm if the setting of hardware test of the module installed in the slot indicated by the module number has been set or not. • Hardware error of the access target intelligent function module. (Please consult your local Mitsubishi Service or representative.)	QS
1404	An intelligent function module response data error was detected.	Reset the CPU module and RUN it again. If the same error is displayed again, the intelligent function module, CPU module or base unit is faulty. (Contact your local Mitsubishi representative.)	QS

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing	
				RUN	ERROR			
1411	CONTROL-BUS ERROR	Module No.	–	Off	Flash	Stop	At power ON/ At reset	
1413	CONTROL-BUS ERROR	–	–	Off	Flash	Stop	Always	
1414	CONTROL-BUS ERROR	–	–	Off	Flash	Stop	When an END instruction executed.	
1415	CONTROL-BUS ERROR	Base No.	–	Off	Flash	Stop	When an END instruction executed.	
1500	AC/DC DOWN	–	–	On	Off	Continue	Always	
1600	BATTERY ERROR*1	Drive Name	–	On	Off	Continue	Always	
1610	EXCEED MAX FLASH ROM REWRIT. ERR.	–	–	On	On	Continue	When an END instruction executed.	

*1 BAT.ALM LED is displayed at BATTERY ERROR.

Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
1411	When performing a parameter I/O allocation the intelligent function module could not be accessed during initial communications. (On error occurring, the head I/O number of the corresponding intelligent function module is stored in the common information.)	Reset the CPU module and RUN it again. If the same error is displayed again, the intelligent function module, CPU module or base unit is faulty. (Contact your local Mitsubishi representative.)	QS
1413	An error was detected on the system bus.	The intelligent function module, CPU module or base unit is faulty. (Contact your local Mitsubishi representative.)	QS
1414	An error was detected on the system bus.	The intelligent function module, CPU module or base unit is faulty. (Contact your local Mitsubishi representative.)	QS
1415	Fault of the base unit was detected.	The intelligent function module, CPU module or base unit is faulty. (Contact your local Mitsubishi representative.)	QS
1500	• A momentary power supply interruption has occurred. • The power supply went off.	Check the power supply.	QS
1600	• The battery voltage in the CPU module has dropped below stipulated level. • The lead connector of the CPU module battery is not connected. • The lead connector of the CPU module battery is not securely engaged.	• Change the battery. • Install a lead connector of the battery. • Check the lead connector of the CPU module for looseness. Firmly engage the connector if it is loose.	QS
1610	The number of writing to the standard ROM exceeded one hundred thousand times. (Number of writing>100,000 times)	Replace the CPU modules.	QS

9.1.4 Error code list (2000 to 2999)

The following shows the error messages from the error code 2000 to 2999, the contents and causes of the errors, and the corrective actions for the errors.

Table9.3 Error code

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing	
				RUN	ERROR			
2000	MODULE VERIFY ERROR	Module No.	–	Off	Flash	Stop	When an END instruction executed.	
2100	MODULE LAYOUT ERROR	Module No.	–	Off	Flash	Stop	At power ON/ At reset	
2106	MODULE LAYOUT ERROR	Module No.	–	Off	Flash	Stop	At power ON/ At reset	
2107	MODULE LAYOUT ERROR	Module No.	–	Off	Flash	Stop	At power ON/ At reset	

Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
2000	• Intelligent function module information at power ON are changed. • During operation, Intelligent function module are not installed properly or installed on the base unit.	Read the common information of the error using the GX Developer, and check and/or change the module that corresponds to the numerical values (module number) there. Alternatively, monitor the special registers SD 150 to SD 153 at a GX Developer, and change the fuse at the output module whose bit has a value of "1".	QS
2100	• In the parameter I/O allocation settings, an Inteli (intelligent function module) was allocated to a location reserved for an I/O module. • In the I/O assignment setting of the PLC parameter dialog box, the number of points assigned to the intelligent function module is less than the number of points of the mounted module.	Reset the parameter I/O allocation setting to conform to the actual status of the intelligent function module.	QS
2106	• More than three CC-Link Safety master modules are mounted. • More than two CC-Link IE Controller Network modules are mounted. • More than two MELSECNET/H modules are mounted. • More than two Ethernet modules are mounted. • A module that the Safety CPU module cannot recognize has been mounted. • More than two CC-Link IE Field Network master/local modules (with safety functions) are mounted.	• Mount two or less CC-Link Safety master modules. • Mount either only one CC-Link IE Controller Network module or MESECNET/H module. • Mount only one Ethernet module. • Mount a module supported in the Safety CPU module. • Mount only one CC-Link IE Field Network master/local module (with safety functions).	QS
2107	The start X/Y set in the PLC parameter's I/O assignment settings is overlapped with the one for another module.	Make the PLC parameter's I/O assignment setting again so it is consistent with the actual status of the intelligent function module.	QS

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing	
				RUN	ERROR			
2112	INTELLIGENT FUNCTION MODULE ERR.	Module No.	Program error location	Off	Flash	Stop	At an execution of instruction	
2124	MODULE LAYOUT ERROR	Module No.	–	Off	Flash	Stop	At power ON/ At reset	
2125	MODULE LAYOUT ERROR	Module No.	–	Off	Flash	Stop	At power ON/ At reset	
2200	MISSING PARAMETER	Drive No.	–	Off	Flash	Stop	At power ON/ At reset	
2210	BOOT ERROR	Drive No.	–	Off	Flash	Stop	At power ON/ At reset	

	Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
	2112	• The module other than intelligent function module is specified by the intelligent function module dedicated instruction. Or, it is not the corresponding intelligent function module. • There is no network No. specified by the network dedicated instruction. Or the relay target network does not exit.	Read the individual information of the error using GX Developer, check and correct the intelligent function module dedicated instruction corresponding to its value (program error location).	QS
	2124	• A module is installed to the actual I/O points or greater. • A module is installed to the slot whose assigned I/O range includes the limit of actual I/O points.	• Remove the module installed to the actual I/O points or greater. • Reset the I/O assignment setting of the parameter so as not to exceed the actual I/O points.	QS
	2125	• A module that the safety CPU module cannot recognize has been mounted. • There was no response from the intelligent function module.	• Mount a module supported in the safety CPU module. • The intelligent function module is experiencing a hardware fault. (Contact your local Mitsubishi representative.)	QS
	2200	There is no parameter file at the program memory.	Set the parameter file to the program memory.	QS
	2210	The contents of the boot file are incorrect.	Check the boot setting.	QS

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing
				RUN	ERROR		
2500	CAN'T EXECUTE PROGRAM	File name/ Drive No.	–	Off	Flash	Stop	At power ON/ At reset/ STOP→RUN
2501							
2502							
2503							

Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
2500	• There is a program file that uses a device that is out of the range set in the PLC parameter device setting.	Read the common information of the error using the GX Developer, check to be sure that the parameter device allocation setting and the program file device allocation correspond to the numerical values there (file name), and correct if necessary.	QS
2501	• More than two program files exist for one drive. • The program name differs from the program contents.	• Delete unnecessary program files. • Match the program name with the program contents.	QS
2502	The program file is incorrect. Alternatively, the file contents are not those of a sequence program.	Check whether the program version is＊＊＊.QPG, and check the file contents to be sure they are for a sequence program.	QS
2503	There are no program files at all. (A drive No. is only displayed on the common information.)	• Check program configuration. • Check parameters and program configuration.	QS

9.1.5 Error code list (3000 to 3999)

The following shows the error messages from the error code 3000 to 3999, the contents and causes of the errors, and the corrective actions for the errors.

Table9.4 Error code

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing	
				RUN	ERROR			
3000	PARAMETER ERROR	File name/ Drive No.	Parameter number	Off	Flash	Stop	At power ON/ At reset	
3001								
3003	PARAMETER ERROR	File name/ Drive No.	Parameter number	Off	Flash	Stop	At power ON/ At reset	
3004	PARAMETER ERROR	File name/ Drive No.	Parameter number	Off	Flash	Stop	At power ON/ At reset	
3008	PARAMETER ERROR	File name/ Drive No.	Parameter number	Off	Flash	Stop	When CC-Link Safety remote station return	
3100	NETWORK PARAMETER ERROR	File name/ Drive No.	Parameter number	Off	Flash	Stop	At power ON/ At reset	

Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
3000	The PLC parameter settings for timer time limit setting, the RUN-PAUSE contact, and number of vacant slots is outside the range that can be used by the CPU module.	Read the detailed information of the error using the GX Developer, check the parameter items corresponding to those numerical values (parameter numbers), and correct when necessary.	QS
3001	The parameter settings are corrupted.		
3003	The number of devices set at the PLC parameter device settings exceeds the possible CPU module range.	Read the detailed information of the error using the GX Developer, check the parameter items corresponding to those numerical values (parameter numbers), and correct when necessary.	QS
3004	The parameter file is incorrect. Alternatively, the contents of the file are not parameters.	Check whether the parameter file version is ＊＊＊.QPA, and check the file contents to be sure they are parameters.	QS
3008	The system power is not restarted or the CPU module is not reset after writing the parameter to the CPU module. (When the remote I/O station returns while the system power is restarted or the CPU module is reset after writing the PLC parameter into CPU module, this error occurs.)	Restart the power or reset the CPU module.	QS
3100	• The number of modules actually mounted is different from that is set in Network parameter for CC-Link IE Field Network. • The start I/O number of the module actually mounted is different from that is set in Network parameter for CC-Link IE Field Network. • Some data in the parameters are not supported.	• Check the setting in Network parameter and actual mounting status, and if they differ, correct either the setting or mounting status so that they become the same. If the parameter setting is corrected, write Network parameter to the CPU module. • If an error occurs even after correction, a hardware failure is considered. (Please consult your local Mitsubishi representative.)	QS*3

*1 This applies when the serial number (first five digits) of the MELSECNET/H module is "08102" or later.
*2 This applies when the serial number (first five digits) of the CPU module is "10031" or earlier.
*3 This applies when the serial number (first five digits) of the CPU module is "13042" or later.

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing	
				RUN	ERROR			
3100	NETWORK PARAMETER ERROR	File name/ Drive No.	Parameter number	Off	Flash	Stop	At power ON/ At reset	

Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
3100	• The number of modules actually mounted is different from that is set in Network parameter for CC-Link IE Controller Network. • The start I/O number of the module actually mounted is different from that is set in Network parameter for CCLink IE Controller Network. • Some data in the parameters are not supported. • The station type for a CC-Link IE Controller Network has been changed while the power is ON. (RESET → RUN is required to change the station type.)	• Check the setting in Network parameter and actual mounting status, and if they differ, correct either the setting or mounting status so that they become the same. If the parameter setting is corrected, write Network parameter to the CPU module. • If an error occurs even after correction, a hardware failure is considered. (Please consult your local Mitsubishi representative.)	QS
	• The number of modules actually mounted is different from that is set in Network parameter for MELSECNET/H. • The start I/O number of the module . actually mounted is different from thatis set in Network parameter for MELSECNET/H. • Some data in the parameters are not supported. • The station type for a MELSECNET/H has been changed while the power is ON. (RESET → RUN is required to change the station type.) • The mode switches of MELSECNET/H module*1 are out of the setting range.	• Check the setting in Network parameter and actual mounting status, and if they differ, correct either the setting or mounting status so that they become the same. If the parameter setting is corrected, write Network parameter to the CPU module. • If an error occurs even after correction, a hardware failure is considered. (Please consult your local Mitsubishi representative.) • Set the mode switches of MELSECNET/H module*1 within the setting range.	QS

*1 This applies when the serial number (first five digits) of the MELSECNET/H module is "08102" or later.
*2 This applies when the serial number (first five digits) of the CPU module is "10031" or earlier.
*3 This applies when the serial number (first five digits) of the CPU module is "13042" or later.

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status RUN	LED Status ERROR	CPU Operation Status	Diagnostic Timing	
3101	NETWORK PARAMETER ERROR	File name/ Drive No.	Parameter number	Off	Flash	Stop	At power ON/ At reset	
3102	NETWORK PARAMETER ERROR	File name/ Drive No.	Parameter number	Off	Flash	Stop	At power ON/ At reset	
3103	NETWORK PARAMETER ERROR	File name/ Drive No.	Parameter number	Off	Flash	Stop	At power ON/ At reset	

Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
3101	The refresh parameter for CC-Link IE Controller Network is out of the setting range. • The refresh parameter for CC-Link IE Field Network is out of the setting range. • The refresh parameter for CC-Link IE Field Network is overlapped with the refresh parameter or the I/O assignment setting for other networks. • The start I/O number of the module actually mounted is different from that is set in Network parameter. • The refresh parameter for MELSECNET/H is out of the setting range.	Check the setting in Network parameter and actual mounting status, and if they differ, correct either the setting or mounting status so that they become the same. If the parameter setting is corrected, write Network parameter to the CPU module.	QS
3102	• The network module detected an error in Network parameter. • The MELSECNET/H inherent parameter setting is incorrect.	• Correct the parameter setting and write Network parameter to the CPU. • If an error occurs even after correction, a hardware failure is considered. (Please consult your local Mitsubishi representative.)	QS
3103	• The number of modules actually mounted is different from that is set in Network parameter for Ethernet. • The start I/O number of the module actually mounted is different from that is set in Network parameter for Ethernet.	• Check the setting in Network parameter and actual mounting status, and if they differ, correct either the setting or mounting status so that they become the same. If the parameter setting is corrected, write Network parameter to the CPU module. • If an error occurs even after correction, a hardware failure is considered. (Please consult your local Mitsubishi representative.)	QS

*1 This applies when the serial number (first five digits) of the MELSECNET/H module is "08102" or later.
*2 This applies when the serial number (first five digits) of the CPU module is "10031" or earlier.
*3 This applies when the serial number (first five digits) of the CPU module is "13042" or later.

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing	
				RUN	ERROR			
3104	NETWORK PARAMETER ERROR	File name/ Drive No.	Parameter number	Off	Flash	Stop	At power ON/ At reset	
3105	CC-LINK PARAMETER ERROR	File name/ Drive No.	Parameter number	Off	Flash	Stop	At power ON/ At reset	
3106	CC-LINK PARAMETER ERROR	File name/ Drive No.	Parameter number	Off	Flash	Stop	At power ON/ At reset	

	Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
	3104	• The same network number is used for Ethernet, CC-Link IE Controller Network, and MELSECNET/H. • The network number, station number, and/or group number set in Network parameter are out of the setting range. • The specified I/O number is outside the range of the used CPU module. • The Ethernet inherent parameter setting is incorrect.	• Correct the parameter setting and write Network parameter to the CPU module. • If an error occurs even after correction, a hardware failure is considered. (Please consult your local Mitsubishi representative.)	QS
	3105	• Though the number of CC-Link modules set in the network parameters is one or more, the number of actually mounted modules is zero. • The start I/O number in the common parameters is different from that of the actually mounted module. • The station type of the CC-Link module count setting parameters is different from that of the actually mounted station.	• Correct and write the network parameters. • If the error occurs after correction, it suggests a hardware fault. (Contact your local Mitsubishi representative.)	QS
	3106	• The network refresh parameter for CC-Link is out of range. • Although the safety remote station set in the network parameter does not support the safety data monitoring time, it is set for the station. • The refresh parameter for CC-Link IE Field Network is overlapped with the refresh parameter or the I/O assignment setting for other networks.	• Check the [Model name] and [Module technical version] of the safety remote station settings, or delete the setting of the safety data monitoring time. • Correct and write the network parameters. • If the error occurs after correction, it suggests a hardware fault. (Contact your local Mitsubishi representative.)	QS
		The safety data monitoring time has been set.	Delete the setting of the safety data monitoring time.	QS*2

*1 This applies when the serial number (first five digits) of the MELSECNET/H module is "08102" or later.
*2 This applies when the serial number (first five digits) of the CPU module is "10031" or earlier.
*3 This applies when the serial number (first five digits) of the CPU module is "13042" or later.

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing	
				RUN	ERROR			
3107	CC-LINK PARAMETER ERROR	File name/ Drive No.	Parameter number	Off	Flash	Stop	At power ON/ At reset	
3150	DUPLICAT.NET.NO.	File name/ Drive No.	Parameter number (A block No. of the switch setting is stored.)	Off	Flash	Stop	At power ON/ At reset	
3400	REMOTE PASSWORD ERROR	---	---	Off	Flash	Stop	At power ON/ At reset	
3401	REMOTE PASSWORD ERROR	---	---	Off	Flash	Stop	At power ON/ At reset	

Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
3107	The CC-Link parameter setting is incorrect.	• Correct and write the network parameters. • If the error occurs after correction, it suggests a hardware fault. (Contact your local Mitsubishi representative.)	QS
3150	The network No. set in Network parameter for CC-Link IE Field Network is overlapped with the one set for another network module.	Check the parameter setting.	QS*3
3400	The start I/O number of the target module in Remote password is set to other than 0H to 3E0H.	Change the start I/O number of the target module to the number within 0H to 3E0H.	QS
3401	Ethernet module of function version B or later is not mounted on the slot specified for the start I/O number of Remote password.	Mount the Ethernet module of function version B or later on the slot specified for the start I/O number of Remote password.	QS

*1 This applies when the serial number (first five digits) of the MELSECNET/H module is "08102" or later.
*2 This applies when the serial number (first five digits) of the CPU module is "10031" or earlier.
*3 This applies when the serial number (first five digits) of the CPU module is "13042" or later.

9.1.6 Error code list (4000 to 4999)

The following shows the error messages from the error code 4000 to 4999, the contents and causes of the errors, and the corrective actions for the errors.

Table9.5 Error code

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing
				RUN	ERROR		
4000	INSTRUCTION CODE ERROR	Program error location	–	Off	Flash	Stop	At power ON/ At reset/ STOP→RUN
4002	INSTRUCTION CODE ERROR	Program error location	–	Off	Flash	Stop	At power ON/ At reset/ STOP→RUN
4003	INSTRUCTION CODE ERROR	Program error location	–	Off	Flash	Stop	At power ON/ At reset/ STOP→RUN
4004	INSTRUCTION CODE ERROR	Program error location	–	Off	Flash	Stop	At power ON/ At reset/ STOP→RUN
4010	MISSING END INSTRUCTION	Program error location	–	Off	Flash	Stop	At power ON/ At reset/ STOP→RUN
4100	OPERATION ERROR	Program error location	–	Off/ On	Flash	Stop	When instruction executed.
4101	OPERATION ERROR	Program error location	–	Off/ On	Flash	Stop	When instruction executed.

Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
4000	• The program contains an instruction code that cannot be decoded. • An unusable instruction is included in the program. • An unusable safety FB is included in the program.	Read the common information of the error using a GX Developer, check error step corresponding to its numerical value (program error location), and correct the problem.	QS
4002	• The name of dedicated instruction specified in the program is incorrect. • The dedicated instruction specified in the program cannot be executed by the specified module. • An unsupported instruction exists.		QS
4003	The number of devices for the dedicated instruction specified in the program is incorrect.		
4004	The device, which cannot be used by the dedicated instruction specified in the program, is specified.		
4010	There is no END instruction in the program.		QS
4100	The instruction cannot process the contained data.	Read the common information of the error using the GX Developer, check error step corresponding to its numerical value (program error location), and correct the problem.	QS
4101	• The designated device number for data processed by the instruction exceeds the usable range. • Alternatively, the stored data or constants for the devices designated by the instruction exceeds the usable range.	Read the common information of the error using the GX Developer, check error step corresponding to its numerical value (program error location), and correct the problem.	QS

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing	
				RUN	ERROR			
4102	OPERATION ERROR	Program error location	–	Off	Flash	Stop	At an execution of instruction	
4700	PROGRAM ABORT EXECUTED	Program error location	Aborted program information	Off	Flash	Stop	When executing the S.QSABORT instructions.	

	Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
	4102	• The network number and/or station number specified by the dedicated instruction are incorrect. • The module number, network number, and/or number of character strings specified by the dedicated instruction exceed the allowable range.	Read the common information of the error using the GX Developer, check error step corresponding to its numerical value (program error location), and correct the problem.	QS
	4700	The S.QSABORT instruction was executed, and the program was forcefully stopped.	Remove the cause before executing the S.QSABORT instruction.	QS

9.1.7 Error code list (5000 to 5999)

The following shows the error messages from the error code 5000 to 5999, the contents and causes of the errors, and the corrective actions for the errors.

Table9.6 Error code

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing	
				RUN	ERROR			
5001	WDT ERROR	Time (value set)	Time (value actually measured)	Off	Flash	Stop	Always	
5010	PROGRAM SCAN TIME OVER	Time (value set)	Time (value actually measured)	On	On	Continue	Always	

	Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
	5001	The program scan time exceeded the WDT value specified in the PLC RAS setting of the PLC parameter dialog box.	Read the individual information of the error with the GX Developer, check its value (time), and shorten the scan time.	QS
	5010	The program scan time exceeded the constant scan time specified in the PLC RAS setting of the PLC parameter dialog box.	Review the constant scan time in the PLC parameter so that the margin time of constant scan may be fully reserved.	QS

9.1.8 Error code list (8000 to 9000)

The following shows the error messages from the error code 8000 to 9000, the contents and causes of the errors, and the corrective actions for the errors.

Table9.7 Error code

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing
				RUN	ERROR		
8000	INTERNAL REGISTER ERROR	–	Error information	Off	Flash	Stop	At power ON/ At reset/When an END instruction executed.
8010	INTERNAL BUS ERROR	–	Error information	Off	Flash	Stop	At power ON/ At reset/When an END instruction executed.
8020	CPU A & B CAN'T BE SYNCHRO-NIZED	–	Error information	Off	Flash	Stop	Always
8021							When an END instruction executed.
8031	INCORRECT FILE	–	Diagnostics file information	Off	Flash	Stop	At power ON/ At reset
8032	INCORRECT FILE	–	Diagnostics file information	Off	Flash	Stop	When an END instruction executed.
8050	SAFETY OUTPUT VERIFY ERROR	Module No./ Station No.	–	Off	Flash	Stop	When an END instruction executed.

*1 The operating status of a CPU module in case of an error can be set in the "Operation settings during remote station error" of "Parameter". The default is set to "Stop" (The LED indication changes according to the status).
*2 At occurrence of "F****", the “USER” LED turns on.

Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
8000	Error is detected by the inside register diagnostics built in the CPU module.	This suggests a CPU module hardware fault. (Contact your local Mitsubishi representative.)	QS
8010	Error is detected inside the bus of the CPU module.	This suggests a CPU module hardware fault. (Contact your local Mitsubishi representative.)	QS
8020	Mismatch has occurred in the execution status of CPU A and CPU B.	• Take measure against noise. • Reset it and run it again. If the same error is displayed again, this suggests a CPU module hardware fault. (Contact your local Mitsubishi representative.)	QS
8021	Mismatch of program execution times is detected between CPU A and CPU B.		
8031	Error of a file stored in the program memory or the standard ROM is detected.	The file indicated by the individual information SD17~SD22 is written into the individual information SD16, and turn the CPU power is turned OFF→ON or reset→reset canceling.If the same error is displayed again, this suggests a CPU module hardware fault. (Contact your local Mitsubishi representative.)	QS
8032			
8050	The verification of safety outputs between the CPU A and CPU B in a CPU module resulted in a mismatch.	• Check if the program for outputing safety outputs is correct. • Take measure against noise. • Reset it and run it again. If the same error is displayed again, this suggests a CPU module hardware fault. (Contact your local Mitsubishi representative.)	QS

*3 This applies when the serial number (first five digits) of the CPU module is "13042" or later.

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status RUN	LED Status ERROR	CPU Operation Status	Diagnostic Timing	
8060	INCORRECT FIRMWARE	–	Error information	Off	Flash	Stop	At power ON/ At reset/When an END instruction executed.	
8070	INTERNAL CPU COMMUNI-CATION ERROR	–	Error information	Off	Flash	Stop	At power ON/ At reset	
8071								
8072								
8073							When an END instruction executed.	
8074								
8080	POWER SUPPLY ERROR	–	Error information	Off	Off/On	Stop	Always	
8090	VOLTAGE DIAGNOSIS ERROR	–	Error information	Off	Flash	Stop	When an END instruction executed.	
8100	TEST MODE TIME EXCEEDED	–	–	On	On	Continues	When an END instruction executed.	
8120	WDT CLOCK CHECK ERROR	–	–	Off	Flash	Stop	Always	
8300	CC-LINK REMOTE DETECTION ERROR	CC-Link Safety information	CC-Link Safety information	Off/On*1	Flash/ On*1	Stop/ Continues *1	Always	

*1 The operating status of a CPU module in case of an error can be set in the "Operation settings during remote station error" of "Parameter". The default is set to "Stop" (The LED indication changes according to the status).
*2 At occurrence of "F****", the “USER” LED turns on.

	Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
	8060	Error of system programs is detected.	• Take measure against noise. • Reset it and run it again. If the same error is displayed again, this suggests a CPU module hardware fault. (Contact your local Mitsubishi representative.)	QS
	8070	The initial communication between CPU A and CPU B was unsuccessful.	• Take measure against noise. • Reset it and run it again. If the same error is displayed again, this suggests a CPU module hardware fault. (Contact your local Mitsubishi representative.)	QS
	8071	CPU A and CPU B cannot send data to each other.		
	8072	CPU A and CPU B cannot receive data from each other.		
	8073	CPU A and CPU B cannot send data to each other.		
	8074	CPU A and CPU B cannot receive data from each other.		
	8080	Power supply voltage error has been detected in a CPU module.	• Take measure against noise. • Reset it and run it again. If the same error is displayed again, this suggests a CPU module hardware fault. (Contact your local Mitsubishi representative.)	QS
	8090	The error of line voltage monitoring circuit is detected.	• Take measure against noise. • Reset it and run it again. If the same error is displayed again, this suggests a CPU module hardware fault. (Contact your local Mitsubishi representative.)	QS
	8100	The continuous operation time on TEST MODE exceeds the TEST MODE continuous operation time set by the parameter.	Confirm that the safety CPU operation mode can be switched to the SAFETY MODE, and start operation after switching the TEST MODE to the SAFETY MODE.	QS
	8120	Clock stop of the WDT is detected.	• Take measure against noise. • Reset it and run it again. If the same error is displayed again, this suggests a CPU module hardware fault. (Contact your local Mitsubishi representative.)	QS
	8300	Error information is received from CC-Link Safety remote station.	Confirm the error code of the relevant CC-Link Safety remote station. (Refer to the manual of the CC-Link Safety remote module for the confirmation.)	QS

*3 This applies when the serial number (first five digits) of the CPU module is "13042" or later.

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing	
				RUN	ERROR			
8310	CC-LINK PRODUCT INFO. MISMATCH	CC-Link Safety information	CC-Link Safety information	Off/On*1	Flash/ On*1	Stop/ Continues *1	Always	
8320	CC-LINK DATA RECEPTION TIMEOUT	CC-Link Safety information	CC-Link Safety information	Off/On*1	Flash/ On*1	Stop/ Continues *1	While initializing remote station	
8321							Always	
8322							When receiving remote station's error information	

*1 The operating status of a CPU module in case of an error can be set in the "Operation settings during remote station error" of "Parameter". The default is set to "Stop" (The LED indication changes according to the status).
*2 At occurrence of "F****", the “USER” LED turns on.

Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
8310	The installed product is different from the specified one by network parameter.	Check that [Model name], [Module technical version] or [Production information] of the CC-Link Safety remote station set in the network parameter matches the product information of the relevant CC-Link Safety remote station.(Refer to the manual of the CC-Link Safety remote module for the confirmation.)	QS
8320	The response data cannot be received during the initial processing of CC-Link Safety remote station.	• Check that the following operations are not executed. (1)Switching the operation mode (2)Writing the program memory to ROM (3)Registration/change of the CPU access password (4)Initialization of PLC memory (If executed, this error may occur due to the increase of the interval between data communications of CC-Link Safety.) • When instantaneous power failure occurs to the supply power, change to the asynchronous mode or slow down the speed. • Execute the link test to check the soundness of transmission path. • Check the setting of transmission speed. • Check if the setting value of the Safety refresh monitoring time is appropriate. • Check if the setting value of the Safety data monitoring timer is appropriate.	QS
8321	The response data cannot be received during the normal communication with CC-Link Safety remote station.		QS
8322	The response data cannot be received during processing error information from CC-Link Safety remote station.		QS

*3 This applies when the serial number (first five digits) of the CPU module is "13042" or later.

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing	
				RUN	ERROR			
8330	CC-LINK RECEIVED DATA ERROR	CC-Link Safety information	CC-Link Safety information	Off/On*1	Flash/ On*1	Stop/ Continues *1	Always	
8331								
8332								
8333								
8334								
8400	CC IE ANOTHER MODULE DETECT ERR.	Error classification, error item, network No., station No.	CC-Link IE Field Network information	On	On	Continue	Always	
8410	CC IE PRODUCT INFO. MISMATCH	Network No., station No.	CC-Link IE Field Network information	On	On	Continue	When connection with safety stations established	

*1 The operating status of a CPU module in case of an error can be set in the "Operation settings during remote station error" of "Parameter". The default is set to "Stop" (The LED indication changes according to the status).
*2 At occurrence of "F****", the “USER” LED turns on.

Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
8330	The received command differs from the expected value.	• Check the cable status visually or by a line test. • Hardware error of the CC-Link Safety master module or the relevant CC-Link Safety remote module (Contact your local Mitsubishi representative.)	QS
8331	Lapse in separated receiving data has occurred.		QS
8332	The link ID in receiving data is different from the expected value.	• Check if the link ID setting of the relevant remote station and the link ID that has been set in the network parameter are identical. • Hardware error of the CC-Link Safety master module or the relevant CC-Link Safety remote module (Contact your local Mitsubishi representative.)	QS
8333	The running No. in receiving data is different from the expected value.	• Check if the setting value of the Safety refresh monitoring time is appropriate. • Hardware error of the CC-Link Safety master module or the relevant CC-Link Safety remote module (Contact your local Mitsubishi representative.)	QS
8334	The CC-Link Safety master station cannot recognize the received data.	• Check the cable status visually or by a line test. • Hardware error of the CC-Link Safety master module or the relevant CC-Link Safety remote module (Contact your local Mitsubishi representative.)	QS
8400	Safety communication target station detection error (Error information is received from a safety communication target station on CC-Link IE Field Network.)	Check the error code of the communication target station on CC-Link IE Field Network.	QS*3
8410	Safety station product information mismatch ("Model name" or "Production information" of a safety station is incorrect.)	It may be malfunction due to noise. Check the soundness of transmission path by performing the CC-Link IE Field Network diagnostics. Check the distance between power lines and communication cables, and the installation status of each device as well.	QS*3

*3 This applies when the serial number (first five digits) of the CPU module is "13042" or later.

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing	
				RUN	ERROR			
8420	CC IE DATA RECEPTION TIMEOUT	Network No., station No.	CC-Link IE Field Network information	On	On	Continue	When connection established with safety stations	
8421							Always	
8422							When an error received from a safety station	
8423							Always	
8424								

*1 The operating status of a CPU module in case of an error can be set in the "Operation settings during remote station error" of "Parameter". The default is set to "Stop" (The LED indication changes according to the status).
*2 At occurrence of "F****", the “USER” LED turns on.

	Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
	8420	Safety monitoring timeout error (A timeout error occurred during safety communication.)	• Check that the following operations are not performed. 1) Switching the operation mode 2) Writing the program memory to ROM 3) Registration/change of the CPU access password 4) Initialization of PLC memory 5) Monitoring, diagnostics (If executed, this error may occur due to the increase of the interval between data communications of CC-Link IE Field.) • It may be malfunction due to noise. Check the soundness of transmission path by performing the CC-Link IE Field Network diagnostics. Check the distance between power lines and communication cables, and the installation status of each device as well.	QS*3
	8421			
	8422			
	8423			
	8424			

*3 This applies when the serial number (first five digits) of the CPU module is "13042" or later.

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status		CPU Operation Status	Diagnostic Timing
				RUN	ERROR		
8425	CC IE DATA RECEPTION TIMEOUT	Network No., station No.	CC-Link IE Field Network information	On	On	Continue	Always

*1 The operating status of a CPU module in case of an error can be set in the "Operation settings during remote station error" of "Parameter". The default is set to "Stop" (The LED indication changes according to the status).
*2 At occurrence of "F****", the "USER" LED turns on.

	Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
	8425	Safety monitoring timeout error (A timeout error occurred during safety communication.)	• Check that the following operations are not performed. 1) Switching the operation mode 2) Writing the program memory to ROM 3) Registration/change of the CPU access password 4) Initialization of PLC memory 5) Monitoring, diagnostics (If executed, this error may occur due to the increase of the interval between data communications of CC-Link IE Field.) • Check that the transmission interval monitoring time for the target station is correctly set. • It may be malfunction due to noise. Check the soundness of transmission path by performing the CC-Link IE Field Network diagnostics. Check the distance between power lines and communication cables, and the installation status of each device as well.	QS*3

*3 This applies when the serial number (first five digits) of the CPU module is "13042" or later.

Error Code (SD0)	Error Message	Common Information (SD5 to 15)	Individual Information (SD16 to 26)	LED Status RUN	LED Status ERROR	CPU Operation Status	Diagnostic Timing	
8426	CC IE DATA RECEPTION TIMEOUT	Network No., station No.	CC-Link IE Field Network information	On	On	Continue	Always	
8430	CC IE RECEIVED DATA ERROR	Network No., station No.	CC-Link IE Field Network information	On	On	Continue	Always	
8431								
8432								
8440	CC IE CONNECTION SETTING ERROR	Network No., station No.	CC-Link IE Field Network information	On	On	Continue	Always	
9000	F**** *2	Program error location	Annunciator number	On	Off	Continue	When instruction executed.	

*1 The operating status of a CPU module in case of an error can be set in the "Operation settings during remote station error" of "Parameter". The default is set to "Stop" (The LED indication changes according to the status).
*2 At occurrence of "F****", the “USER” LED turns on.

	Error Code (SD0)	Error Contents and Cause	Corrective Action	Corresponding CPU
	8426	Safety monitoring timeout error (A timeout error occurred during safety communication.)	• Check that the following operations are not performed. 1) Switching the operation mode 2) Writing the program memory to ROM 3) Registration/change of the CPU access password 4) Initialization of PLC memory 5) Monitoring, diagnostics (If executed, this error may occur due to the increase of the interval between data communications of CC-Link IE Field.) • Set the larger value for the safety refresh monitoring time. Or set the smaller value for the transmission interval monitoring time. • It may be malfunction due to noise. Check the soundness of transmission path by performing the CC-Link IE Field Network diagnostics. Check the distance between power lines and communication cables, and the installation status of each device as well.	QS*3
	8430 8431 8432	Safety station reception data error (Incorrect data is received.)	It may be malfunction due to noise. Check the soundness of transmission path by performing the CC-Link IE Field Network diagnostics. Check the distance between power lines and communication cables, and the installation status of each device as well.	QS*3
	8440	Safety connection setting error (Incorrect CID is received.)	Check that safety connection is correctly set for own station.	QS*3
	9000	Annunciator (F) was set ON (**** in the error message indicates the detected annunciator number.)	Read the individual information of the error using the GX Developer, and check the program corresponding to the numerical value (annunciator number).	QS

*3 This applies when the serial number (first five digits) of the CPU module is "13042" or later.

Memo

A APPENDICIES

Appendix 1 OPERATION PROCESSING TIME

Appendix 1.1 Definition

(1) Processing time taken by the QSCPU is the total of the following processing times.
- Total of each instruction processing time
- END processing time
- I/O refresh time
- Service processing time

(2) Instruction processing time
This is the total of processing time of each instruction shown in Appendix 1.2.

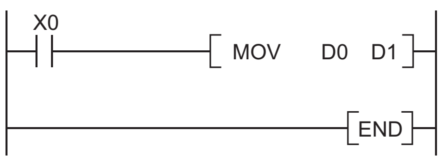
(3) END processing time
END instruction processing time is the total of the following processing times.
- END instruction processing time shown in Appendix 1.2
- CC-Link IE Field Network, CC-Link IE Controller Network, and MELSECNET/H refresh time
- Auto refresh time of CC-Link Safety[*1]
- MELSECNET/H refresh time[*2]
- Communication processing time with GX Developer [*3]

*1: For details CC-Link IE Field Network, CC-Link IE Controller Network and MELSECNET/H refresh time, refer to the following manual.
• QSCPU User's Manual (Function Explanation, Program Fundamentals)
*2: For details the auto refresh time of CC-Link Safety., refer to the following manual.
• CC-Link Safety System Master Module User's Manual
*3: Also refer to the following manual for the communication processing time with the GX Developer.
• QSCPU User's Manual (Function Explanation, Program Fundamentals)

(4) I/O refresh time can be calculated using the equation below.
I/O refresh time (μs) = Number of I/O points × 0.224 + 310 (μs)

(5) For details of service processing time, refer to the following manual.
- QSCPU User's Manual (Function Explanation, Program Fundamentals)

(6) Calculation of processing time
The total of the processing time when executed by the QSCPU is calculated as follows:

Ladder Example	Processing Time
	[Instruction processing time] LD X0 . . . . 0.10μs MOV D0 D1 . . . . 0.35μs → 0.10(μs) + 0.35(μs) = 0.45(μs) [END processing time] END instruction . . . . 8200μs [I/O refresh time] I/O refresh time . . . . 310μs TOTAL 0.45(μs) + 8200(μs) + 310(μs) = 8510.45(μs)

Appendix 1.2 Operation Processing Time

The processing times for the individual instructions are shown in the table on the following pages.

Operation processing times can vary substantially depending on the nature of the sources and destinations of the instructions, and the values contained in the following tables should therefore be taken as a set of general guidelines to processing times rather than as being strictly accurate.

(1) Sequence instruction

<table>
<tr><th>Instruction</th><th colspan="2">Conditions (Device)</th><th>Processing Time (μs)</th></tr>
<tr><td rowspan="2">LD
LDI
AND
ANI
OR
ORI</td><td colspan="2">X0</td><td>0.10</td></tr>
<tr><td colspan="2">D0.0</td><td>0.15</td></tr>
<tr><td rowspan="2">LDP
LDF
ANDP
ANDF
ORP
ORF</td><td colspan="2">X0</td><td rowspan="2">0.15</td></tr>
<tr><td colspan="2">D0.0</td></tr>
<tr><td>ANB
ORB
MPS
MRD
MPP</td><td colspan="2">—</td><td>0.10</td></tr>
<tr><td rowspan="2">INV</td><td colspan="2">When not executed</td><td rowspan="2">0.10</td></tr>
<tr><td colspan="2">When executed</td></tr>
<tr><td rowspan="2">MEP
MEF</td><td colspan="2">When not executed</td><td rowspan="2">0.15</td></tr>
<tr><td colspan="2">When executed</td></tr>
<tr><td rowspan="2">EGP</td><td>When not executed</td><td>(OFF→OFF)
(ON→ON)</td><td rowspan="2">0.10</td></tr>
<tr><td>When executed</td><td>(OFF→ON)
(ON→OFF)</td></tr>
<tr><td rowspan="4">EGF</td><td rowspan="2">When not executed</td><td>(OFF→OFF)</td><td>11</td></tr>
<tr><td>(ON→ON)</td><td>14</td></tr>
<tr><td rowspan="2">When executed</td><td>(OFF→ON)</td><td>14</td></tr>
<tr><td>(ON→OFF)</td><td>16</td></tr>
</table>

APPENDICES

Instruction		Conditions (Device)			Processing Time (μs)
OUT	Y	When not changed	(OFF→OFF) (ON→ON)		0.10
		When changed	(OFF→ON) (ON→OFF)		
	D0.0	When not changed	(OFF→OFF) (ON→ON)		0.20
		When changed	(OFF→ON) (ON→OFF)		
	F	When OFF			18
		When ON	When displayed		370
			Display completed		240
	T	When not executed			0.55
		When executed	After Time Up		0.55
			When added	K	0.55
				D	0.60
	C	When not executed			0.55
		When executed	After Time Up		0.55
			When added	K	0.55
				D	0.60
OUTH	T	When not executed			0.55
		When executed	After Time Up		0.55
			When added	K	0.55
				D	0.60
SET	Y	When not executed			0.10
		When executed	When not changed (ON→ON)		0.10
			When changed (OFF→ON)		0.10
	D0.0	When not executed			0.20
		When executed	When not changed (ON→ON)		0.20
			When changed (OFF→ON)		0.20
	F	When not executed			0.25
		When executed	When displayed		365
			Display completed		235

Instruction		Conditions (Device)		Processing Time (μs)
RST	Y	When not executed		0.10
		When executed	When not changed (OFF→OFF)	0.10
			When changed (ON→OFF)	0.10
	D0.0	When not executed		0.20
		When executed	When not changed (ON→ON)	0.20
			When changed (OFF→ON)	0.20
	SM	When not executed		0.10
		When executed		0.10
	F	When not executed		0.25
		When executed	When displayed	115
			Display completed	87
	T, C	When not executed		0.40
		When executed		0.50
	D	When not executed		0.20
		When executed		0.30
PLS		—		7.1
PLF		—		7.1
FF	Y	When not executed		0.25
		When executed		4.9
MC		M0		0.20
		D0.0		0.30
MCR		—		0.10
END		Performs error check		8200
NOP		—		0.10
NOPLF PAGE		—		0.10

(2) Basic instructions

The processing time when the instruction is not executed is calculated as follows:
0.10 $\times$ (Number of steps of each instruction +1) μs

Instruction	Conditions (Device)		Processing Time (μs)
LD =	When continuity established		0.40
	When no continuity		0.40
AND =	When not executed		0.35
	When executed	When continuity established	0.40
		When no continuity	0.40
OR =	When not executed		0.35
	When executed	When continuity established	0.40
		When no continuity	0.40
LD < >	When continuity established		0.40
	When no continuity		0.40
AND < >	When not executed		0.35
	When executed	When continuity established	0.40
		When no continuity	0.40
OR < >	When not executed		0.35
	When executed	When continuity established	0.40
		When no continuity	0.40
LD >	When continuity established		0.40
	When no continuity		0.40
AND >	When not executed		0.35
	When executed	When continuity established	0.40
		When no continuity	0.40
OR >	When not executed		0.35
	When executed	When continuity established	0.40
		When no continuity	0.40
LD < =	When continuity established		0.40
	When no continuity		0.40

Instruction	Conditions (Device)		Processing Time (μs)
AND < =	When not executed		0.35
	When executed	When continuity established	0.40
		When no continuity	0.40
OR < =	When not executed		0.35
	When executed	When continuity established	0.40
		When no continuity	0.40
LD <	When continuity established		0.40
	When no continuity		0.40
AND <	When not executed		0.35
	When executed	When continuity established	0.40
		When no continuity	0.40
OR <	When not executed		0.35
	When executed	When continuity established	0.40
		When no continuity	0.40
LD > =	When continuity established		0.40
	When no continuity		0.40
AND > =	When not executed		0.35
	When executed	When continuity established	0.40
		When no continuity	0.40
OR > =	When not executed		0.35
	When executed	When continuity established	0.40
		When no continuity	0.40
LDD =	When continuity established		0.50
	When no continuity		0.50
ANDD =	When not executed		0.40
	When executed	When continuity established	0.50
		When no continuity	0.50
ORD =	When not executed		0.40
	When executed	When continuity established	0.50
		When no continuity	0.50
LDD < >	When continuity established		0.50
	When no continuity		0.50
ANDD < >	When not executed		0.40
	When executed	When continuity established	0.50
		When no continuity	0.50
ORD < >	When not executed		0.40
	When executed	When continuity established	0.50
		When no continuity	0.50
LDD >	When continuity established		0.50
	When no continuity		0.50
ANDD >	When not executed		0.40
	When executed	When continuity established	0.50
		When no continuity	0.50
ORD >	When not executed		0.40
	When executed	When continuity established	0.50
		When no continuity	0.50
LDD < =	When continuity established		0.50
	When no continuity		0.50

Instruction	Conditions (Device)		Processing Time (μs)
ANDD < =	When not executed		0.40
	When executed	When continuity established	0.50
		When no continuity	0.50
ORD < =	When not executed		0.40
	When executed	When continuity established	0.50
		When no continuity	0.50
LDD <	When continuity established		0.50
	When no continuity		0.50
ANDD <	When not executed		0.40
	When executed	When continuity established	0.50
		When no continuity	0.50
ORD <	When not executed		0.40
	When executed	When continuity established	0.50
		When no continuity	0.50
LDD > =	When continuity established		0.50
	When no continuity		0.50
ANDD > =	When not executed		0.40
	When executed	When continuity established	0.50
		When no continuity	0.50
ORD > =	When not executed		0.40
	When executed	When continuity established	0.50
		When no continuity	0.50
+ S D +P S D	When executed		0.50
+ S1 S2 D +P S1 S2 D	When executed		0.60
+ S D - P S D	When executed		0.50
+ S1 S2 D - P S1 S2 D	When executed		0.60
D+ S D D+P S D	When executed		0.65
D+ S1 S2 D D+P S1 S2 D	When executed		0.75
D – S D D – P S D	When executed		0.65
D – S1 S2 D D – P S1 S2 D	When executed		0.75
* S1 S2 D * P S1 S2 D	When executed		0.55
/ S1 S2 D /P S1 S2 D	—		14
D * S1 S2 D D * P S1 S2 D	—		42
D/ S1 S2 D D/P S1 S2 D	—		25

Instruction	Conditions (Device)	Processing Time (μs)
INC INCP	—	0.35
DINC DINCP	—	0.45
DEC DECP	—	0.35
DDEC DDECP	—	0.45
BCD BCDP	—	16
DBCD DBCDP	—	23
BIN BINP	—	15
DBIN DBINP	—	18
NEG NEGP	—	14
DNEG DNEGP	—	15
MOV MOVP	S = D0, D = D1	0.35
DMOV DMOVP	S = D0, D = D1	0.45
CML CMLP	—	0.35
DCML DCMLP	—	0.45
BMOV S D n BMOVP S D n	n = 1	35
	n = 96	67
FMOV S D n FMOVP S D n	n = 1	30
	n = 96	48

APPENDICES

(3) Application instructions

The processing time when the instruction is not executed is calculated as follows:
0.10 × (Number of steps of each instruction +1) μs

Instruction	Conditions (Device)	Processing Time (μs)
WAND Ⓢ Ⓓ WANDP Ⓢ Ⓓ	When executed	0.50
WAND Ⓢ1 Ⓢ2 Ⓓ WANDP Ⓢ1 Ⓢ2 Ⓓ	When executed	0.60
DAND Ⓢ Ⓓ DANDP Ⓢ Ⓓ	When executed	0.65
DAND Ⓢ1 Ⓢ2 Ⓓ DANDP Ⓢ1 Ⓢ2 Ⓓ	When executed	0.75
WOR Ⓢ Ⓓ WORP Ⓢ Ⓓ	When executed	0.50
WOR Ⓢ1 Ⓢ2 Ⓓ WORP Ⓢ1 Ⓢ2 Ⓓ	When executed	0.60
DOR Ⓢ Ⓓ DORP Ⓢ Ⓓ	When executed	0.65
DOR Ⓢ1 Ⓢ2 Ⓓ DORP Ⓢ1 Ⓢ2 Ⓓ	When executed	0.75
WXOR Ⓢ Ⓓ WXORP Ⓢ Ⓓ	When executed	0.50
WXOR Ⓢ1 Ⓢ2 Ⓓ WXORP Ⓢ1 Ⓢ2 Ⓓ	When executed	0.60
DXOR Ⓢ Ⓓ DXORP Ⓢ Ⓓ	When executed	0.65
DXOR Ⓢ1 Ⓢ2 Ⓓ DXORP Ⓢ1 Ⓢ2 Ⓓ	When executed	0.75
WXNR Ⓢ Ⓓ WXNRP Ⓢ Ⓓ	When executed	0.50
WXNR Ⓢ1 Ⓢ2 Ⓓ WXNRP Ⓢ1 Ⓢ2 Ⓓ	When executed	0.60
DXNR Ⓢ Ⓓ DXNRP Ⓢ Ⓓ	When executed	0.65
DXNR Ⓢ1 Ⓢ2 Ⓓ DXNRP Ⓢ1 Ⓢ2 Ⓓ	When executed	0.75

(4) QSCPU dedicated instruction

Instruction	Conditions (Device)	Processing Time (μs)
S.QSABORT Ⓢ	When executed (K1234)	344
	When not executed	34

Appendix 2 SPECIAL RELAY LIST

Special relays, SM, are internal relays whose applications are fixed in the Programmable Controller.
For this reason, they cannot be used by sequence programs in the same way as the normal internal relays.
However, they can be turned ON or OFF as needed in order to control the CPU module and remote I/O modules.
The heading descriptions in the following special relay lists are shown in TableApp.2.1.

TableApp.2.1 Descriptions of the special relay lists headings

Item	Function of Item
Number	• Indicates special register number
Name	• Indicates name of special register
Meaning	• Indicates contents of special register
Explanation	• Discusses contents of special register in more detail
Set by (When set)	• Indicates whether the relay is set by the system or user, and, if it is set by the system, when setting is performed. <Set by> S : Set by system U : Set by user (sequence programs or test operations from GX Developer) S/U : Set by both system and user <When set> Indicated only for registers set by system Every END : Set during every END processing Initial : Set only during initial processing (when power supply is turned ON, or when going from STOP to RUN) Status change : Set only when there is a change in status Error : Set when error occurs Instruction execution : Set when instruction is executed Request : Set only when there is a user request (through SM, etc.)

For details on the following items, refer to the following manuals:
• Networks → Manuals of each network module

POINT

In the program that achieves the safety function, only SM1000 to SM1299 can be used.
Special relay other than SM1000 to SM1299 cannot be used in the program that achieves the safety function.

(1) Diagnostic Information

TableApp.2.2 Descriptions of the special relay headings

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding CPU
SM0	Diagnostic errors	OFF : No error ON : Error	• Turns ON when an error is detected by diagnostics (Includes when an annunciator is ON) • Remains ON if the condition is restored to normal thereafter.	S (Error)	QS
SM1	Self-diagnosis error	OFF : No self-diagnosis errors ON : Self-diagnosis	• Turns ON when an error is detected by self-diagnostics (Does not include when an annunciator is ON) • Remains ON if the condition is restored to normal thereafter.	S (Error)	
SM5	Error common information	OFF : No error common information ON : Error common information	• When SM0 is ON, ON if there is error common information	S (Error)	
SM16	Error individual information	OFF : No error individual information ON : Error individual information	• When SM0 is ON, ON if there is error individual information	S (Error)	
SM50	Error reset	OFF→ON: Error reset	• Conducts error reset operation	U	
SM51	Battery low latch	OFF : Normal ON : Battery low	• ON if battery voltage at CPU module or memory card drops below rated value. • Remains ON if the battery voltage returns to normal thereafter. • Synchronous with BAT. LED	S (Error)	
SM52	Battery low	OFF : Normal ON : Battery low	• Same as SM51, but goes OFF subsequently when battery voltage returns to normal.	S (Error)	
SM53	AC DOWN detection	OFF : AC DOWN not detected ON : AC DOWN detected	• Turns ON if an instantaneous power failure of within 20ms occurs during use of the AC power supply module. Reset when the power supply is switched OFF, then ON.	S (Error)	
SM56	Operation error	OFF : Normal ON : Operation error	• ON when operation error is generated • Remains ON if the condition is restored to normal thereafter.	S (Error)	
SM61	I/O module verify error	OFF : Normal ON : Error	• Turns ON if the I/O module differs from the status registered at power on. • Remains ON if the condition is restored to normal thereafter.	S (Error)	
SM62	Annunciator detection	OFF : Not detected ON : Detected	• Goes ON if even one annunciator F goes ON.	S (Instruction execution)	

(2) System information

TableApp.2.3 Special relay

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding CPU
SM203	STOP contact	STOP status	• Turns ON when the CPU is in STOP status.	S (Status change)	QS
SM210	Clock data set request	OFF : Ignored ON : Set request	• Writes clock data stored in SD210 to SD213 to the CPU module after the END instruction of the scan where the relay changes OFF to ON has been executed.	U	
SM211	Clock data error	OFF : No error ON : Error	• Turns ON when an error is detected in the clock data (SD210 to SD213) and turns OFF if no error is detected.	S (Request)	
SM213	Clock data read request	OFF : Ignored ON : Read request	• Reads clock data to SD210 to SD213 in BCD value when the relay is ON.	U	
SM232	Number of writes to ROM	OFF : Within the number of writes ON : Over the number of writes	• Turns ON when the number of writes to ROM exceeds 100,000.	S (Error)	

(3) System clocks/counters

TableApp.2.4 Special relay

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding CPU
SM400	Always ON	ON OFF	• Normally is ON	S (Every END)	QS
SM401	Always OFF	ON OFF	• Normally is OFF	S (Every END)	
SM402	After RUN, ON for 1 scan only	ON OFF 1 scan	• After RUN, ON for 1 scan only.	S (Every END)	
SM403	After RUN, OFF for 1 scan only	ON OFF 1 scan	• After RUN, OFF for 1 scan only.	S (Every END)	
SM410	0.1 second clock	0.05s 0.05s	• Repeatedly changes between ON and OFF at each designated time interval. • When programmable controller power supply is turned ON or a CPU module reset is performed, goes from OFF to start.	S (Status change)	
SM411	0.2 second clock	0.1s 0.1s			
SM412	1 second clock	0.5s 0.5s			
SM413	2 second clock	1s 1s			
SM414	2n second clock	ns ns	• This relay alternates between ON and OFF at intervals of the time (unit: s) specified in SD414. • When programmable controller power supply is turned ON or a CPU module reset is performed, goes from OFF to start.	S (Status change)	

(4) Safety CPU

TableApp.2.5 Special relay

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding CPU
SM560	TEST MODE flag	OFF : Other than TEST MODE ON : TEST MODE	• Turns ON when operating on the TEST MODE. • Turns OFF when operating on the other mode (SAFETY MODE, SAFETY MODE (wait-for-restart)).	S (Status change)	QS
SM561	Continuous RUN of tolerance time setting for the TEST MODE	OFF : Within the setting time ON : Over the setting time	• Turns ON when the continuous RUN of tolerance time set for the TEST MODE in the parameter is exceeded.	S (Error)	

(5) Boot operation

TableApp.2.6 Special relay

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding CPU
SM660	Boot operation	OFF : Program memory execution ON : During boot operation	(On the TEST MODE) • Turns ON during the boot operation from standard ROM. • Turns OFF when the boot operation from standard ROM is not run. (On the SAFETY MODE) • Always ON	S (Initial)	QS

(6) Instruction-Related Special Relays

TableApp.2.7 Special relay

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding CPU
SM722	BIN/DBIN instruction error disabling flag	OFF : Error detection performed ON : Error detection not performed	• Turned ON when "OPERATION ERROR" is suppressed for BIN or DBIN instruction.	U	QS

(7) CC-Link Safety

TableApp.2.8 Special relay

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding CPU
SM1004	Safety refresh communication status of each safety remote station (Safety master module 1)	OFF : Normal ON : Communication error	The safety station refresh communication status is stored. (The status of each station are stored in SD1004 to SD1007.)	S (Status change)	QS
SM1204	Safety refresh communication status of each safety remote station (Safety master module 2)	OFF : Normal ON : Communication error	The safety station refresh communication status is stored. (The status of each station are stored in SD1204 to SD1207.)	S (Status change)	

(8) CC-Link IE Field Network

TableApp.2.9 Special relay

Number	Name	Meaning	Explanation	Set by (When Set)	Corresponding CPU
SM1400	Setting status of safety communication with master station	OFF : Not set ON : Set	The setting status of safety communication with the master station is stored.	S (Initial)	QS*1
SM1420	Safety refresh communication status of each safety station	OFF : Normal (excluding safety refresh communication with the master station) ON : Communication error	The safety refresh communication status with each safety station (excluding the master station) is stored. (The status of each safety station is stored in SD1420 to SD1427.)	S (Status change)	
SM1421	Safety refresh communication status of safety master station	OFF : Normal or safety communication with master station on CC-Link IE Field Network not set ON : Safety refresh communication error	The safety refresh communication status with the master station is stored. When safety communication with the master station is not set in parameter or own station is the master station, this bit turns OFF.		
SM1700	Safety master station interlock status	OFF : Not interlocked ON : Interlocked	When an error is detected and an interlock is activated, this bit turns ON.		
SM1720	Safety master station interlock release request	OFF : I/O interlock of safety station on CC-Link IE Field Network not released ON : I/O interlock of safety station on CC-Link IE Field Network released	The safety station interlock status is released by changing this bit from OFF to ON.	U	

* 1: This applies when the serial number (first five digits) of the CPU module is "13042" or later.

APPENDICES

Appendix 3 SPECIAL REGISTER LIST

Special relays, SM, are internal relays whose applications are fixed in the Programmable Controller.
For this reason, it is not possible to use these registers in sequence programs in the same way that normal registers are used.
However, data can be written as needed in order to control the CPU modules and remote I/O modules.
Data stored in the special registers are stored as BIN values if no special designation has been made to the contrary.

The heading descriptions in the following special register lists are shown in TableApp.3.1.

TableApp.3.1 Descriptions of the special register list headings

Item	Function of Item
Number	• Indicates special register number
Name	• Indicates name of special register
Meaning	• Indicates contents of special register
Explanation	• Discusses contents of special register in more detail
Set by (When set)	• Indicates whether the relay is set by the system or user, and, if it is set by the system, when setting is performed. <Set by> S : Set by system U : Set by user (sequence programs or test operations from GX Developer) S/U : Set by both system and user <When set> Indicated only for registers set by system Every END : Set during every END processing Initial : Set only during initial processing (when power supply is turned ON, or when going from STOP to RUN) Status change : Set only when there is a change in status Error : Set when error occurs Instruction execution : Set when instruction is executed Request : Set only when there is a user request (through SM, etc.) Writing to ROM : Set when writing to ROM

For details on the following items, refer to the following manuals:
- Networks → Manuals of each network module

POINT

In the program that achieves the safety function, only SD1000 to SD1299 can be used.
Special register other than SD1000 to SD1299 cannot be used in the program that achieves the safety function.

(9) Diagnostic Information

TableApp.3.2 Special register

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU
SD0	Diagnostic errors	Diagnosis error code	• Error codes for errors detected by diagnostics are stored as BIN data. • Contents identical to latest fault history information.	S (Error)	QS
SD1	Clock time for diagnosis error occurrence	Clock time for diagnosis error occurrence	• Stores the year (last two digits) and month when SD0 data was updated as BCD 2-digit code. b15 to b8 b7 to b0 Year (0 to 99) | Month (1 to 12) (Example) September, 2006 H0609	S (Error)	
SD2			• Stores the day and hour when SD0 data was updated as BCD 2-digit code. b15 to b8 b7 to b0 Day (1 to 31) | Hour (0 to 23) (Example) 10 a.m. on 25th H2510		
SD3			• Stores the minute and second when SD0 data was updated as BCD 2-digit code. b15 to b8 b7 to b0 Minutes (0 to 59) | Seconds (0 to 59) (Example) 35 min. 48 sec. H3548		
SD4	Error information categories	Error information category code	Category codes to identify what type of error information is stored in the common information (SD5 to SD15) or in the individual information (SD16 to SD26). b15 to b8 b7 to b0 Individual information category codes | Common information category codes • The common information category codes store the following codes: 0 : No error 1: Module No./Base No. 2: File name/Drive name 3: Time (value set) 4: Program error location 9: CC-Link Safety information 10: Module No./Station No. 11: CC-Link IE Field Network information • The individual information category codes store the following codes: 0: No error 2: File name/Drive name 3: Time (value actually measured) 4: Program error location 5: Parameter number 6: Annunciator (F) number 9: Error information 10: CC-Link Safety information 11: Program abort information 12: File diagnostics information 13: CC-Link IE Field Network information	S (Error)	

TableApp.3.2 Special register

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When set)</th><th>Corresponding CPU</th></tr>
<tr><td>SD5</td><td rowspan="11">Error common information</td><td rowspan="11">Error common information</td><td rowspan="11">• Common information corresponding to the error codes (SD0) is stored here.
• The following seven types of information are stored here:
1) Module No./Base No.
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD5</td><td>Slot No./Base No. *1</td></tr>
<tr><td>SD6</td><td>I/O No. *2</td></tr>
<tr><td>SD7</td><td rowspan="9">(Empty)</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td></tr>
<tr><td>SD10</td></tr>
<tr><td>SD11</td></tr>
<tr><td>SD12</td></tr>
<tr><td>SD13</td></tr>
<tr><td>SD14</td></tr>
<tr><td>SD15</td></tr>
</table>
*1: The storing value "255" in SD5 (Slot No.) indicates that the slot number for a module specified by an instruction cannot be identified.
When storing the base number to SD5, store 0 (main base unit).
*2: The storing value "FFFFH" in SD6 (I/O No.) indicates that the I/O number cannot be identified on the I/O assignment setting tab of PLC parameter due to overlapping of I/O numbers or that the I/O number cannot be identified from the network number specified by an instruction. In this case, the error location can be identified in SD5.
2) File name/Drive name
<table>
<tr><th>Number</th><th colspan="2">Meaning</th></tr>
<tr><td>SD5</td><td colspan="2">Drive</td></tr>
<tr><td>SD6</td><td colspan="2" rowspan="4">File name
(ASCII code: 8 characters)</td></tr>
<tr><td>SD7</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td></tr>
<tr><td>SD10</td><td>Extension *3</td><td>2EH(.)</td></tr>
<tr><td>SD11</td><td colspan="2">(ASCII code: 3 characters)</td></tr>
<tr><td>SD12</td><td colspan="2" rowspan="4">(Empty)</td></tr>
<tr><td>SD13</td></tr>
<tr><td>SD14</td></tr>
<tr><td>SD15</td></tr>
</table>
(Example) File name = MAIN.QPG
<table>
<tr><th>b15 to b8</th><th>b7 to b0</th></tr>
<tr><td>41H(A)</td><td>40H(M)</td></tr>
<tr><td>43H(N)</td><td>49H(I)</td></tr>
<tr><td>20H(SP)</td><td>20H(SP)</td></tr>
<tr><td>20H(SP)</td><td>20H(SP)</td></tr>
<tr><td>51H(Q)</td><td>2EH(.)</td></tr>
<tr><td>47H(G)</td><td>50H(P)</td></tr>
</table>
</td><td rowspan="11">S (Error)</td><td rowspan="11">QS</td></tr>
<tr><td>SD6</td></tr>
<tr><td>SD7</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td></tr>
<tr><td>SD10</td></tr>
<tr><td>SD11</td></tr>
<tr><td>SD12</td></tr>
<tr><td>SD13</td></tr>
<tr><td>SD14</td></tr>
<tr><td>SD15</td></tr>
</table>

Remark

*3 : Extensions are shown in TableApp.3.3.

TableApp.3.3Extension name

SDn	SDn+1		Extension name	File type
Higher 8 bits	Lower 8 bits	Higher 8 bits		
51H	50H	41H	QPA	Parameters
51H	50H	47H	QPG	Sequence program
51H	43H	44H	QCD	Device comment

TableApp.3.2 Special register

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When set)</th><th>Corresponding CPU</th></tr>
<tr><td>SD5</td><td rowspan="11">Error common information</td><td rowspan="11">Error common information</td><td rowspan="11">
3) Time (value set)
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD5</td><td>Time : 1μs units (0 to 999μs)</td></tr>
<tr><td>SD6</td><td>Time : 1ms units (0 to 65535ms)</td></tr>
<tr><td>SD7</td><td rowspan="9">(Empty)</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td></tr>
<tr><td>SD10</td></tr>
<tr><td>SD11</td></tr>
<tr><td>SD12</td></tr>
<tr><td>SD13</td></tr>
<tr><td>SD14</td></tr>
<tr><td>SD15</td></tr>
</table>
4) Program error location
<table>
<tr><th>Number</th><th colspan="2">Meaning</th></tr>
<tr><td>SD5</td><td colspan="2" rowspan="4">File name
(ASCII code: 8 characters)</td></tr>
<tr><td>SD6</td></tr>
<tr><td>SD7</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td><td>Extension *3</td><td>2EH(.)</td></tr>
<tr><td>SD10</td><td colspan="2">(ASCII code: 3 characters)</td></tr>
<tr><td>SD11</td><td colspan="2">(Empty)</td></tr>
<tr><td>SD12</td><td colspan="2">Block No.*4</td></tr>
<tr><td>SD13</td><td colspan="2">Step No. *4</td></tr>
<tr><td>SD14</td><td colspan="2">Sequence step No. (L)</td></tr>
<tr><td>SD15</td><td colspan="2">Sequence step No. (H)</td></tr>
</table>
*4: "0" is stored to the block number and the step number.
9) CC-Link Safety information
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD5</td><td>Error classification*5</td></tr>
<tr><td>SD6</td><td>Error item*5</td></tr>
<tr><td>SD7</td><td>Link ID</td></tr>
<tr><td>SD8</td><td>Station No.</td></tr>
<tr><td>SD9</td><td>System area 1</td></tr>
<tr><td>SD10</td><td>System area 2</td></tr>
<tr><td>SD11</td><td>System area 3</td></tr>
<tr><td>SD12</td><td>System area 4</td></tr>
<tr><td>SD13</td><td>System area 5</td></tr>
<tr><td>SD14</td><td>System area 6</td></tr>
<tr><td>SD15</td><td>System area 7</td></tr>
</table>
*5: For details on the error classification and error item, refer to the manuals for the connected devices.
10) Module No./Station No.
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD5</td><td>Slot No.</td></tr>
<tr><td>SD6</td><td>I/O No.</td></tr>
<tr><td>SD7</td><td>Station No.</td></tr>
<tr><td>SD8</td><td rowspan="8">(Empty)</td></tr>
<tr><td>SD9</td></tr>
<tr><td>SD10</td></tr>
<tr><td>SD11</td></tr>
<tr><td>SD12</td></tr>
<tr><td>SD13</td></tr>
<tr><td>SD14</td></tr>
<tr><td>SD15</td></tr>
</table>
11) CC-Link IE Field Network information
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD5</td><td>Error classification*6</td></tr>
<tr><td>SD6</td><td>Error item*6</td></tr>
<tr><td>SD7</td><td>Network No.</td></tr>
<tr><td>SD8</td><td>Station No.</td></tr>
<tr><td>SD9</td><td>System area 1</td></tr>
<tr><td>SD10</td><td>System area 2</td></tr>
<tr><td>SD11</td><td>System area 3</td></tr>
<tr><td>SD12</td><td>System area 4</td></tr>
<tr><td>SD13</td><td>System area 5</td></tr>
<tr><td>SD14</td><td>System area 6</td></tr>
<tr><td>SD15</td><td>System area 7</td></tr>
</table>
*6: For details on the error classification and error item, refer to the manuals for the connected devices.
If the QSCPU is connected, "350" is stored as the error classification and the error code of the transfer target is stored as the error item.
</td><td rowspan="11">S (Error)</td><td rowspan="11">QS</td></tr>
<tr><td>SD6</td></tr>
<tr><td>SD7</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td></tr>
<tr><td>SD10</td></tr>
<tr><td>SD11</td></tr>
<tr><td>SD12</td></tr>
<tr><td>SD13</td></tr>
<tr><td>SD14</td></tr>
<tr><td>SD15</td></tr>
</table>

TableApp.3.2 Special register

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU
SD16					
SD17					
SD18					
SD19					
SD20					
SD21	Error individual information	Error individual information		S (Error)	QS
SD22					
SD23					
SD24					
SD25					
SD26					

- Individual information corresponding to error codes (SD0) is stored here.
- There are the following ten different types of information are stored.

2) File name/Drive name

Number	Meaning	
SD16	Drive	
SD17	File name (ASCII code: 8 characters)	
SD18		
SD19		
SD20		
SD21	Extension ∗3	2EH(.)
SD22	(ASCII code: 3 characters)	
SD23	(Empty)	
SD24		
SD25		
SD26		

(Example) File name = MAIN.QPG

b15 to b8	b7 to b0
41H(A)	40H(M)
43H(N)	49H(I)
20H(SP)	20H(SP)
20H(SP)	20H(SP)
51H(Q)	2EH(.)
47H(G)	50H(P)

3) Time (value Actually measured)

Number	Meaning
SD16	Time : 1 μs units (0 to 999 μs)
SD17	Time : 1ms units (0 to 65535ms)
SD18	(Empty)
SD19	
SD20	
SD21	
SD22	
SD23	
SD24	
SD25	
SD26	

4) Program error location

Number	Meaning	
SD16	File name (ASCII code: 8 characters)	
SD17		
SD18		
SD19		
SD20	Extension ∗3	2EH(.)
SD21	(ASCII code: 3 characters)	
SD22	(Empty)	
SD23	Block No. ∗6	
SD24	Step No. ∗6	
SD25	Sequence step No. (L)	
SD26	Sequence step No. (H)	

*6: "0" is stored to the block number and the step number.

5) Parameter No.

Number	Meaning
SD16	Parameter No.
SD17	(Empty)
SD18	
SD19	
SD20	
SD21	
SD22	
SD23	
SD24	
SD25	
SD26	

6) Annunciator number

Number	Meaning
SD16	No.
SD17	(Empty)
SD18	
SD19	
SD20	
SD21	
SD22	
SD23	
SD24	
SD25	
SD26	

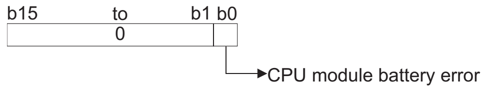
9) Error information

Number	Meaning
SD16	Error information 1
SD17	Error information 2
SD18	Error information 3
SD19	Error information 4
SD20	Error information 5
SD21	Error information 6
SD22	Error information 7
SD23	Error information 8
SD24	Error information 9
SD25	Error information 10
SD26	Error information 11

TableApp.3.2 Special register

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When set)</th><th>Corresponding CPU</th></tr>
<tr><td>SD16</td><td rowspan="11">Error individual information</td><td rowspan="11">Error individual information</td><td rowspan="11">
10) CC-Link Safety information
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD16</td><td>Number of items for individual information</td></tr>
<tr><td>SD17</td><td>individual information 1</td></tr>
<tr><td>SD18</td><td>individual information 2</td></tr>
<tr><td>SD19</td><td>individual information 3</td></tr>
<tr><td>SD20</td><td>individual information 4</td></tr>
<tr><td>SD21</td><td>individual information 5</td></tr>
<tr><td>SD22</td><td>individual information 6</td></tr>
<tr><td>SD23</td><td>individual information 7</td></tr>
<tr><td>SD24</td><td>individual information 8</td></tr>
<tr><td>SD25</td><td>individual information 9</td></tr>
<tr><td>SD26</td><td>individual information 10</td></tr>
</table>
11) Program abort information
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD16</td><td>Abort code ∗5</td></tr>
<tr><td>SD17</td><td rowspan="10">(Empty)</td></tr>
<tr><td>SD18</td></tr>
<tr><td>SD19</td></tr>
<tr><td>SD20</td></tr>
<tr><td>SD21</td></tr>
<tr><td>SD22</td></tr>
<tr><td>SD23</td></tr>
<tr><td>SD24</td></tr>
<tr><td>SD25</td></tr>
<tr><td>SD26</td></tr>
</table>
*5 : The specified abort code is stored by the S.QSABORT instruction.
12) File diagnostics information
<table>
<tr><th>Number</th><th colspan="2">Meaning</th></tr>
<tr><td>SD16</td><td>Error information</td><td>Drive No.</td></tr>
<tr><td>SD17</td><td colspan="2" rowspan="4">File name
(ASCII code: 8 characters)</td></tr>
<tr><td>SD18</td></tr>
<tr><td>SD19</td></tr>
<tr><td>SD20</td></tr>
<tr><td>SD21</td><td>Extension ∗3</td><td>2EH(.)</td></tr>
<tr><td>SD22</td><td colspan="2">(ASCII code: 3 characters)</td></tr>
<tr><td>SD23</td><td colspan="2" rowspan="2">Error information 2</td></tr>
<tr><td>SD24</td></tr>
<tr><td>SD25</td><td colspan="2" rowspan="2">Error information 3</td></tr>
<tr><td>SD26</td></tr>
</table>
(Example) File name = MAIN.QPG
<table>
<tr><th>b15 to b8</th><th>b7 to b0</th></tr>
<tr><td>41H(A)</td><td>4DH(M)</td></tr>
<tr><td>4EH(N)</td><td>43H(I)</td></tr>
<tr><td>20H(SP)</td><td>20H(SP)</td></tr>
<tr><td>20H(SP)</td><td>20H(SP)</td></tr>
<tr><td>51H(Q)</td><td>2EH(.)</td></tr>
<tr><td>47H(G)</td><td>50H(P)</td></tr>
</table>
13) CC-Link IE Field Network information
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD16</td><td>Number of items for individual information</td></tr>
<tr><td>SD17</td><td>individual information 1</td></tr>
<tr><td>SD18</td><td>individual information 2</td></tr>
<tr><td>SD19</td><td>individual information 3</td></tr>
<tr><td>SD20</td><td>individual information 4</td></tr>
<tr><td>SD21</td><td>individual information 5</td></tr>
<tr><td>SD22</td><td>individual information 6</td></tr>
<tr><td>SD23</td><td>individual information 7</td></tr>
<tr><td>SD24</td><td>individual information 8</td></tr>
<tr><td>SD25</td><td>individual information 9</td></tr>
<tr><td>SD26</td><td>individual information 10</td></tr>
</table>
</td><td rowspan="11">S (Error)</td><td rowspan="11">QS</td></tr>
<tr><td>SD17</td></tr>
<tr><td>SD18</td></tr>
<tr><td>SD19</td></tr>
<tr><td>SD20</td></tr>
<tr><td>SD21</td></tr>
<tr><td>SD22</td></tr>
<tr><td>SD23</td></tr>
<tr><td>SD24</td></tr>
<tr><td>SD25</td></tr>
<tr><td>SD26</td></tr>
</table>

TableApp.3.2 Special register

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU
SD27	Diagnostics error CPU identifier	CPU identifier (CPU A/CPU B)	• The identifier of the CPU module that issued the diagnostics error (SD0 to SD26) is stored. 0001H : CPU A 0002H : CPU B	S (Error)	
SD50	Error reset	Error code to be cleared	• An error code to be cleared is stored.	U	
SD51	Battery low latch	Bit pattern indicating where battery voltage drop occurred	• All corresponding bits go 1(ON) when battery voltage drops. • Subsequently, these remain 1(ON) even after battery voltage has been returned to normal.  	S (Error)	
SD52	Battery low	Bit pattern indicating where battery voltage drop occurred	• Same configuration as SD51 above • Turns to 0 (OFF) when the battery voltage returns to normal thereafter.	S (Error)	
SD53	AC DOWN detection	Number of times for AC DOWN detection	• Every time the input voltage falls to or below 85% (AC power) of the rating during calculation of the CPU module, the value is incremented by 1 and stored in BIN code.	S (Error)	
SD61	I/O module verify error number	I/O module verify error module number	• The lowest I/O number of the module where the I/O module verification error took place.	S (Error)	
SD62	Annunciator number	Annunciator number	• The first annunciator number (F number) to be detected is stored here.	S (Instruction execution)	
SD63	Number of annunciators	Number of annunciators	• Stores the number of annunciators searched.	S (Instruction execution)	QS
SD64 SD65 SD66 SD67 SD68 SD69 SD70 SD71 SD72 SD73 SD74 SD75 SD76 SD77 SD78 SD79	Table of detected annunciator numbers	Annunciator detection number	• When F goes ON due to OUT F or SET F, the F numbers which go progressively ON from SD64 through SD79 are registered. • The F numbers turned OFF by RST F are deleted from SD64 - SD79, and the F numbers stored after the deleted F numbers are shifted to the preceding registers. After 16 annunciators have been detected, detection of the 17th will not be stored from SD64 through SD79.	S (Instruction execution)	

	SET F50	SET F25	SET F99	RST F25	SET F15	SET F70	SET F65	SET F38	SET F110	SET F151	SET F210	RST F50	
SD62	0	50	50	50	50	50	50	50	50	50	50	50	99
SD63	0	1	2	3	2	3	4	5	6	7	8	9	8

SD62: (Number detected)
SD63: (Number of annunciators detected)

SD64	0	50	50	50	50	50	50	50	50	50	50	50	99
SD65	0	0	25	25	99	99	99	99	99	99	99	99	15
SD66	0	0	0	99	0	15	15	15	15	15	15	15	70
SD67	0	0	0	0	0	0	70	70	70	70	70	70	65
SD68	0	0	0	0	0	0	0	65	65	65	65	65	38
SD69	0	0	0	0	0	0	0	0	38	38	38	38	110
SD70	0	0	0	0	0	0	0	0	0	110	110	110	151
SD71	0	0	0	0	0	0	0	0	0	0	151	151	210
SD72	0	0	0	0	0	0	0	0	0	0	0	210	0
SD73	0	0	0	0	0	0	0	0	0	0	0	0	0
SD74	0	0	0	0	0	0	0	0	0	0	0	0	0
SD75	0	0	0	0	0	0	0	0	0	0	0	0	0
SD76	0	0	0	0	0	0	0	0	0	0	0	0	0
SD77	0	0	0	0	0	0	0	0	0	0	0	0	0
SD78	0	0	0	0	0	0	0	0	0	0	0	0	0
SD79	0	0	0	0	0	0	0	0	0	0	0	0	0

(Number detected)

TableApp.3.2 Special register

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When set)</th><th>Corresponding CPU</th></tr>
<tr><td>SD81</td><td>Cause of error</td><td>Cause of error</td><td>• When a continuation error occurs, the corresponding bits are all set to ON.
• Canceling the error, starting up the safety PLC power or canceling the safety CPU module reset after eliminating the cause of the error makes the bits go OFF.
<table>
<tr><th>Bit No.</th><th>Name of the cause</th></tr>
<tr><td>0</td><td>Instantaneous power failure</td></tr>
<tr><td>1</td><td>Battery low</td></tr>
<tr><td>2</td><td>Standard ROM write count excess</td></tr>
<tr><td>3</td><td>TEST MODE continuous RUN tolerance timeout</td></tr>
<tr><td>4</td><td>Scan timeout</td></tr>
<tr><td>5</td><td>Annunciator ON</td></tr>
<tr><td>6</td><td>Safety remote station detection error
Safety communication target station detection error</td></tr>
<tr><td>7</td><td>Safety remote station product information mismatch
Safety station product information mismatch</td></tr>
<tr><td>8</td><td>Initial monitoring timeout error
Safety monitoring timeout error
Error monitoring timeout error</td></tr>
<tr><td>9</td><td>Safety remote station data split error
Safety remote command error
Safety remote station link ID error
Safety remote station running number error
Safety remote station reception data error
Safety station reception data error
Safety connection setting error</td></tr>
<tr><td>10 to 15</td><td>Empty (fixed to 0)</td></tr>
</table></td><td>S (Error)</td><td>QS</td></tr>
<tr><td>SD150
SD151
SD152
SD153</td><td>I/O module verify error</td><td>Bit pattern, in units of 16 points, indicating the modules with verify errors.
0: No I/O verify errors
1: I/O verify error present</td><td>• When an I/O module whose information is different from the one registered at power-on is detected, the corresponding I/O module number (in units of 16 points) is stored. (The I/O module number set in parameter is stored if it is set.)
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SD150</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (XY0)</td></tr>
<tr><td>SD151</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (XY190)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SD153</td><td>0</td><td>1 (XY3E0)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
Indicates an I/O module verify error
If a module has more than 16 I/O points, the I/O module number (in units of 16 points) corresponds to the start 16 I/O points turns on.
(Example) When a module with 64 I/O points is mounted in slot 0, only b0 turns on when an error is detected.
• Not cleared even if the blown fuse is replaced with a new one. This flag is cleared by error resetting operation.</td><td>S (Error)</td><td>QS</td></tr>
</table>

APPENDICES

(10) System information

TableApp.3.4 Special register

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU
SD200	Status of switch	Status of CPU switch	• The CPU switch status is stored in the following format. b15 to b4 b3 to b0 Empty 1) 1): CPU switch status 0: RUN 1: STOP 2: RESET	S (Every END)	QS
SD201	LED status	Status of CPU-LED	• The following bit patterns are used to store the statuses of the LEDs on the CPU module: • 0 is off, 1 is on, and 2 is flicker. b15 to b12 b11 to b8 b7 to b4 b3 to b0 8) 7) 6) 5) 4) 3) 2) 1) 1): RUN 5): Empty 2): ERR. 6): Empty 3): USER 7): TEST 4): BAT. 8): Empty	S (Status change)	
SD203	Operating status of CPU	Operating status of CPU	• The CPU operating status is stored as indicated in the following figure: b15 to b12 b11 to b8 b7 to b4 b3 to b0 2) 1) 1): Operating status of CPU 0: RUN 2: STOP 2): STOP cause 0: Instruction in remote operation program from RUN/STOP/RESET switch 1: Remote contact 2: Remote operation from GX Developer 4: Error 5: SAFETY MODE (wait-for-restart) 6: "Write to PLC" was executed. Note stores the above-mentioned factors from the smallest number in priority to the largest one. However, "4:error" is treated as the highest priority.	S (Every END)	

TableApp.3.4 Special register

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU
SD210	Clock data	Clock data (year, month)	• The year (last two digits) and month are stored as BCD code at SD210 as shown below: b15 to b12 b11 to b8 b7 to b4 b3 to b0 Year / Month Example: September, 2006 0609H	S (Request)/U	QS
SD211	Clock data	Clock data (day, hour)	• The day and hour are stored as BCD code at SD211 as shown below: b15 to b12 b11 to b8 b7 to b4 b3 to b0 Day / Hour Example: 25st, 10 a.m. 2510H	S (Request)/U	QS
SD212	Clock data	Clock data (minute, second)	• The minutes and seconds (after the hour) are stored as BCD code at SD212 as shown below: b15 to b12 b11 to b8 b7 to b4 b3 to b0 Minute / Second Example: 35 min., 48 sec. 3548H	S (Request)/U	QS
SD213	Clock data	Clock data (later digits of year, day of week)	• Stores the year (two digits) and the day of the week in SD213 in the BCD code format as shown below. b15 to b12 b11 to b8 b7 to b4 b3 to b0 Example: 2006, Monday 2001H Higher digits of year (19 or 20) Day of the week: 0 Sunday, 1 Monday, 2 Tuesday, 3 Wednesday, 4 Thursday, 5 Friday, 6 Saturday	S (Request)/U	QS
SD232 SD233	ROM write count	ROM write count up to now	• Store the ROM write count up to now.	S (Writing to ROM)	QS
SD240	Base mode	0: Automatic mode	• Stores the base mode.(0 fixed)	S (Initial)	QS
SD241	Extension stage number	0: Main base only	• Stores the maximum number of the extension bases being installed. (0 fixed)	S (Initial)	QS

APPENDICES

TableApp.3.4 Special register

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU
SD242	Installed Q base presence/ absence	Base type differentiation 0: Base not installed 1: QS**B is installed	b15 to b1 b0 Empty | → Main base unit	S (Initial)	QS
SD243	No. of base slots (Operation status)	No. of base slots	b15 to b4 b3 to b0 SD243: Empty | Main SD244: Empty • As shown above, each area stores the number of slots being installed. (Number of set slots when parameter setting has been made)	S (Initial)	
SD244					
SD245	No. of base slots (Mounting status)	No. of base slots	b15 to b4 b3 to b0 SD245: Empty | Main SD246: Empty • As shown above, each area stores the numbers of module-mounted slots of the base unit (actual number of slots of the installed base unit).	S (Initial)	
SD246					
SD250	Loaded maximum I/O	Loaded maximum I/O No.	• The upper 2 digits of the final I/O number plus 1 of the modules loaded are stored as BIN values.	S (Initial)	
SD254	CC-Link IE Controller Network, MELSECNET /H information	Number of modules mounted	• Indicates the number of mounted CC-Link IE Controller Network or MELSECNET/H modules.	S (Initial)	
SD255		I/O number	• Indicates the I/O number of mounted CC-Link IE Controller Network or MELSECNET/H module.		
SD256		Network number	• Indicates the network number of mounted CC-Link IE Controller Network or MELSECNET/H module.		
SD257		Group number	• Indicates the group number of mounted CC-Link IE Controller Network or MELSECNET/H module.		
SD258		Station number	• Indicates the station number of mounted CC-Link IE Controller Network or MELSECNET/H module.		
SD290	Device assignment (Same as the parameter setting)	Number of points assigned for X	• Stores the number of points currently set for X devices	S (Initial)	
SD291		Number of points assigned for Y	• Stores the number of points currently set for Y devices		
SD292		Number of points assigned for M	• Stores the number of points currently set for M devices		
SD294		Number of points assigned for B	• Stores the number of points currently set for B devices		
SD295		Number of points assigned for F	• Stores the number of points currently set for F devices		
SD296		Number of points assigned for SB	• Stores the number of points currently set for SB devices		
SD297		Number of points assigned for V	• Stores the number of points currently set for V devices		
SD299		Number of points assigned for T	• Stores the number of points currently set for T devices		

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU
SD300	Device assignment (Same as the parameter setting)	Number of points assigned for ST	• Stores the number of points currently set for ST devices	S (Initial)	QS
SD301		Number of points assigned for C	• Stores the number of points currently set for C devices		
SD302		Number of points assigned for D	• Stores the number of points currently set for D devices		
SD303		Number of points assigned for W	• Stores the number of points currently set for W devices		
SD304		Number of points assigned for SW	• Stores the number of points currently set for SW devices		
SD340	Ethernet information	Number of modules mounted	• Indicates the number of mounted Ethernet modules.	S (Initial)	
SD341		I/O number	• Indicates the I/O number of mounted Ethernet module.		
SD342		Network number	• Indicates the network number of mounted Ethernet module.		
SD343		Group number	• Indicates the group number of mounted Ethernet module.		
SD344		Station number	• Indicates the station number of mounted Ethernet module.		

(11) System clocks/counters

TableApp.3.5 Special register

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU
SD412	1 second counter	Number of counts in 1-second units	• Following programmable controller CPU module RUN, 1 is added each second • Count repeats from 0 to 32767 to -32768 to 0	S (Status change)	QS
SD414	2n second clock setting	2n second clock units	• Stores value n of 2n second clock (Default is 30) • Setting can be made between 1 to 32767	U	
SD420	Scan counter	Number of counts in each scan	• Incremented by 1 for each scan execution after the CPU module is set to RUN. • Count repeats from 0 to 32767 to -32768 to 0	S (Every END)	

(12) Scan information

TableApp.3.6 Special register

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU
SD520	Current scan time	Current scan time (in 1 ms units)	• The current scan time is stored into SD520 and SD521. (Measurement is made in 100 μs units.) SD520: Stores the value of ms. (Storage range: 0 to 6553) SD521: Stores the value of μs. (Storage range: 0 to 900) (Example) When the current scan time is 23.6ms, the following values are stored. SD520 = 23 SD521 = 600 • The accuracy of processing time of scantime is ±0.1ms.	S (Every END)	QS
SD521		Current scan time (in 100 μs units)			
SD524	Minimum scan time	Minimum scan time (in 1 ms units)	• Stores the minimum value of the scan time into SD524 and SD525. (Measurement is made in 100 μs units.) SD524: Stores the ms place. (Storage range: 0 to 6553) SD525: Stores the μs place. (Storage range: 0 to 900) • The accuracy of processing time of scantime is μ 0.1ms.	S (Every END)	
SD525		Minimum scan time (in 100 μs units)			
SD526	Maximum scan time	Maximum scan time (in 1 ms units)	• Stores the maximum value of the scan time into SD526 and SD527. (Measurement is made in 100 μs units.) SD526: Stores the ms place. (Storage range: 0 to 6553) SD527: Stores the μs place. (Storage range: 0 to 900) • The accuracy of processing time of scantime is ±0.1ms.	S (Every END)	
SD527		Maximum scan time (in 100 μs units)			
SD540	END processing time	END processing time (in 1 ms units)	• Stores the time from when the scan program ends until the next scan starts into SD540 and SD541. (Measurement is made in 100 μs units.) SD540: Stores the ms place. (Storage range: 0 to 6553) SD541: Stores the μs place. (Storage range: 0 to 900) • The accuracy of END processing time is ±0.1ms.	S (Every END)	
SD541		END processing time (in 100 μs units)			
SD542	Constant scan wait time	Constant scan wait time (in 1 ms units)	• Stores the wait time for constant scan setting into SD542 and SD543. (Measurement is made in 100 μs units.) SD542: Stores the ms place. (Storage range: 0 to 6553) SD543: Stores the μs place. (Storage range: 0 to 900) • The accuracy of constant scan wait time is ±0.1ms.	S (Every END)	
SD543		Constant scan wait time (in 100 μs units)			
SD548	Scan program execution time	Scan program execution time (in 1 ms units)	• Stores the execution time of a scan program during one scan into SD548 and SD549. (Measurement is made in 100 μs units.) SD548: Stores the ms place. (Storage range: 0 to 6553) SD549: Stores the μs place. (Storage range: 0 to 900) • Stored every scan. • The accuracy of scan program execution time is ±0.1ms.	S (Every END)	
SD549		Scan program execution time (in 100 μs units)			

(13) Safety CPU

TableApp.3.7 Special register

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU
SD560	Safety CPU operation mode	Safety CPU operation mode	• Stores the safety CPU operation mode. b15 to b2 b1 b0 Empty 00 : SAFETY MODE 01 : TEST MODE 10 : SAFETY MODE (Wait-for-restart)	S (Status change)	QS
SD561 SD562	TEST MODE continuous RUN time	TEST MODE continuous RUN time (seconds)	• Stores the TEST MODE continuous RUN time. (Measured in seconds) (RUN time in TEST MODE. Start measurement when STOP & RUN (Time when operation is STOP is not included.) • Stores the measurement valve with the range of 1 to 2147483647.	S (Every END)	

(14) Memory card

TableApp.3.8 Special register

Number	Name	Meaning	Explanation	Set by (When set)	Corresponding CPU
SD620	Memory type	Memory type	• Indicates the type of built-in memory. b15 to b8 b7 to b4 b3 to b0 0 0 Drive 4 (Standard ROM) Fixed at "3 (FLASH ROM)"	S (Initial)	QS
SD623	Drive 4 (ROM) capacity	Drive 4 capacity	• Drive 4 capacity is stored in 1 kbyte units.	S (Initial)	

(15) CC-Link Safety

TableApp.3.9 Special register

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When set)</th><th>Corresponding CPU</th></tr>
<tr><td>SD1000
to
SD1003</td><td>Safety remote station specification (CC-Link Safety master module 1)</td><td>0: No safety remote station specification
1: Safety remote station specification</td><td>• The specified status of safety remote station is stored.
• "0" is stored for the standard remote station.
<table>
<tr><th></th><th>b15</th><th>b14</th><th>to</th><th>b1</th><th>b0</th></tr>
<tr><td>SD1000</td><td>16</td><td>15</td><td>to</td><td>2</td><td>1</td></tr>
<tr><td>SD1001</td><td>32</td><td>31</td><td>to</td><td>18</td><td>17</td></tr>
<tr><td>SD1002</td><td>48</td><td>47</td><td>to</td><td>34</td><td>33</td></tr>
<tr><td>SD1003</td><td>64</td><td>63</td><td>to</td><td>50</td><td>49</td></tr>
</table>
1 to 64 in the table indicate station numbers.</td><td>S (Initial)</td><td rowspan="4">QS</td></tr>
<tr><td>SD1004
to
SD1007</td><td>Safety refresh communication status of each safety remote station (CC-Link Safety master module 1)</td><td>0: Normal communication, reserved station, unused station, standard remote station on CC-Link Safety
1: Safety station communication error</td><td>• The refresh communication status of safety remote station is stored.
• "0" is stored for the standard remote station.
<table>
<tr><th></th><th>b15</th><th>b14</th><th>to</th><th>b1</th><th>b0</th></tr>
<tr><td>SD1004</td><td>16</td><td>15</td><td>to</td><td>2</td><td>1</td></tr>
<tr><td>SD1005</td><td>32</td><td>31</td><td>to</td><td>18</td><td>17</td></tr>
<tr><td>SD1006</td><td>48</td><td>47</td><td>to</td><td>34</td><td>33</td></tr>
<tr><td>SD1007</td><td>64</td><td>63</td><td>to</td><td>50</td><td>49</td></tr>
</table>
1 to 64 in the table indicate station numbers.</td><td>S (Status change)</td></tr>
<tr><td>SD1008
to
SD1071</td><td>Safety station communication status (CC-Link Safety master module 1)</td><td>The status of communication with safety remote station is stored.</td><td>• The status of communication with each safety remote station is stored.
• SD1008: Station number 1 to SD1071: Station number 64 (0 fixed in the case of standard remote station, reserved station specified, or without connection)
0: At normal communication
10: At initial
20: During internal information access
30: Link error
8300: Link error (Safety remote station detection error)
8310: Link error (Product information mismatch)
8320: Link error (Initial monitor timeout)
8321: Link error (Safety monitor timeout)
8322: Link error (Error monitor timeout)
8330: Link error (Command error)
8331: Link error (Data split number error)
8332: Link error (Link ID error)
8333: Link error (Running number error)
8334: Link error (Received data error)</td><td>S (Status change)</td></tr>
<tr><td>SD1072
to
SD1075</td><td>Safety station interlock status (CC-Link Safety master module 1)</td><td>0: Interlock is not executed
1: During interlock</td><td>Bit corresponding to the station number turns 1 when the master station goes to the interlock status after the error was detected at the master station.
<table>
<tr><th></th><th>b15</th><th>b14</th><th>to</th><th>b1</th><th>b0</th></tr>
<tr><td>SD1072</td><td>16</td><td>15</td><td>to</td><td>2</td><td>1</td></tr>
<tr><td>SD1073</td><td>32</td><td>31</td><td>to</td><td>18</td><td>17</td></tr>
<tr><td>SD1074</td><td>48</td><td>47</td><td>to</td><td>34</td><td>33</td></tr>
<tr><td>SD1075</td><td>64</td><td>63</td><td>to</td><td>50</td><td>49</td></tr>
</table>
1 to 64 in the table indicate station numbers.</td><td>S (Status change)</td></tr>
</table>

TableApp.3.9 Special register

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When set)</th><th>Corresponding CPU</th></tr>
<tr><td>SD1076 to SD1079</td><td>Safety station interlock cancel request (CC-Link Safety master module 1)</td><td>0: I/O interlock of safety station on CC-Link Safety not released
1: I/O interlock of safety station on CC-Link Safety released</td><td>Cancel the I/O interlock of safety station by changing the bit of register from 0 to 1.
<table><tr><td></td><td>b15</td><td>b14</td><td>to</td><td>b1</td><td>b0</td></tr><tr><td>SD1076</td><td>16</td><td>15</td><td>to</td><td>2</td><td>1</td></tr><tr><td>SD1077</td><td>32</td><td>31</td><td>to</td><td>18</td><td>17</td></tr><tr><td>SD1078</td><td>48</td><td>47</td><td>to</td><td>34</td><td>33</td></tr><tr><td>SD1079</td><td>64</td><td>63</td><td>to</td><td>50</td><td>49</td></tr></table>1 to 64 in the table indicate station numbers.</td><td>U (Request)</td><td rowspan="4">QS</td></tr>
<tr><td>SD1200 to SD1203</td><td>Safety remote station specification (CC-Link Safety master module 2)</td><td>0: No safety remote station specification
1: Safety remote station specification</td><td>• The specified status of safety remote station is stored.
• "0" is stored for the standard remote station.
<table><tr><td></td><td>b15</td><td>b14</td><td>to</td><td>b1</td><td>b0</td></tr><tr><td>SD1200</td><td>16</td><td>15</td><td>to</td><td>2</td><td>1</td></tr><tr><td>SD1201</td><td>32</td><td>31</td><td>to</td><td>18</td><td>17</td></tr><tr><td>SD1202</td><td>48</td><td>47</td><td>to</td><td>34</td><td>33</td></tr><tr><td>SD1203</td><td>64</td><td>63</td><td>to</td><td>50</td><td>49</td></tr></table>1 to 64 in the table indicate station numbers.</td><td>S (Initial)</td></tr>
<tr><td>SD1204 to SD1207</td><td>Safety refresh communication status of each safety remote station (CC-Link Safety master module 2)</td><td>0: Normal communication, reserved station, unused station, standard remote station on CC-Link Safety
1: Safety station communication error</td><td>• The refresh communication status of safety remote station is stored.
• "0" is stored for the standard remote station.
<table><tr><td></td><td>b15</td><td>b14</td><td>to</td><td>b1</td><td>b0</td></tr><tr><td>SD1204</td><td>16</td><td>15</td><td>to</td><td>2</td><td>1</td></tr><tr><td>SD1205</td><td>32</td><td>31</td><td>to</td><td>18</td><td>17</td></tr><tr><td>SD1206</td><td>48</td><td>47</td><td>to</td><td>34</td><td>33</td></tr><tr><td>SD1207</td><td>64</td><td>63</td><td>to</td><td>50</td><td>49</td></tr></table>1 to 64 in the table indicate station numbers.</td><td>S (Status change)</td></tr>
<tr><td>SD1208 to SD1271</td><td>Safety station communication status (CC-Link Safety master module 2)</td><td>The status of communication with safety station is stored.</td><td>• The status of communication with each safety remote station is stored.
• SD1208: Station number 1 to SD1271: Station number 64 (0 fixed in the case of standard remote station, reserved station specified, or without connection)
0: At normal communication
10: At initial
20: During internal information access
30: Link error
8300: Link error (Safety remote station detection error)
8310: Link error (Product information mismatch)
8320: Link error (Initial monitor timeout)
8321: Link error (Safety monitor timeout)
8322: Link error (Error monitor timeout)
8330: Link error (Command error)
8331: Link error (Data split number error)
8332: Link error (Link ID error)
8333: Link error (Running number error)
8334: Link error (Received data error)</td><td>S (Status change)</td></tr>
</table>

APPENDICES

TableApp.3.9 Special register

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When set)</th><th>Corresponding CPU</th></tr>
<tr><td>SD1272
to
SD1275</td><td>Safety station interlock status (CC-Link Safety master module 2)</td><td>0: Interlock is not executed
1: During interlock</td><td>Bit corresponding to the station number turns 1 when the master station goes to the interlock status after the error was detected at the master station.
<table>
<tr><th></th><th>b15</th><th>b14</th><th>to</th><th>b1</th><th>b0</th></tr>
<tr><td>SD1272</td><td>16</td><td>15</td><td>to</td><td>2</td><td>1</td></tr>
<tr><td>SD1273</td><td>32</td><td>31</td><td>to</td><td>18</td><td>17</td></tr>
<tr><td>SD1274</td><td>48</td><td>47</td><td>to</td><td>34</td><td>33</td></tr>
<tr><td>SD1275</td><td>64</td><td>63</td><td>to</td><td>50</td><td>49</td></tr>
</table>
1 to 64 in the table indicate station numbers.</td><td>S (Status change)</td><td rowspan="2">QS</td></tr>
<tr><td>SD1276
to
SD1279</td><td>Safety station interlock cancel request (CC-Link Safety master module 2)</td><td>0: I/O interlock of safety station on CC-Link Safety not released
1: I/O interlock of safety station on CC-Link Safety released</td><td>Cancel the I/O interlock of safety station by changing the bit of register from 0 to 1.
<table>
<tr><th></th><th>b15</th><th>b14</th><th>to</th><th>b1</th><th>b0</th></tr>
<tr><td>SD1276</td><td>16</td><td>15</td><td>to</td><td>2</td><td>1</td></tr>
<tr><td>SD1277</td><td>32</td><td>31</td><td>to</td><td>18</td><td>17</td></tr>
<tr><td>SD1278</td><td>48</td><td>47</td><td>to</td><td>34</td><td>33</td></tr>
<tr><td>SD1279</td><td>64</td><td>63</td><td>to</td><td>50</td><td>49</td></tr>
</table>
1 to 64 in the table indicate station numbers.</td><td>U (Request)</td></tr>
</table>

(16) CC-Link IE Field Network

TableApp.3.9 Special register

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When set)</th><th>Corresponding CPU</th></tr>
<tr><td>SD1400
to
SD1407</td><td>Setting status of safety communication with each station</td><td>0: Not set
1: Set</td><td>The safety communication setting status for each station is stored.
For standard stations, "0" is stored.
<table>
<tr><th></th><th>b15</th><th>b14</th><th>to</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SD1400</td><td>16</td><td>15</td><td>to</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SD1401</td><td>32</td><td>31</td><td>to</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SD1402</td><td>48</td><td>47</td><td>to</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SD1403</td><td>64</td><td>63</td><td>to</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
<tr><td>SD1404</td><td>80</td><td>79</td><td>to</td><td>74</td><td>73</td><td>72</td><td>71</td><td>70</td><td>69</td><td>68</td><td>67</td><td>66</td><td>65</td></tr>
<tr><td>SD1405</td><td>96</td><td>95</td><td>to</td><td>90</td><td>89</td><td>88</td><td>87</td><td>86</td><td>85</td><td>84</td><td>83</td><td>82</td><td>81</td></tr>
<tr><td>SD1406</td><td>112</td><td>111</td><td>to</td><td>106</td><td>105</td><td>104</td><td>103</td><td>102</td><td>101</td><td>100</td><td>99</td><td>98</td><td>97</td></tr>
<tr><td>SD1407</td><td>-</td><td>-</td><td>to</td><td>-</td><td>-</td><td>120</td><td>119</td><td>118</td><td>117</td><td>116</td><td>115</td><td>114</td><td>113</td></tr>
</table>
1 to 120 in the table indicate station numbers.
-: Fixed to "0"</td><td>S (Initial)</td><td>QS*1</td></tr>
<tr><td>SD1420
to
SD1427</td><td>Safety refresh communication status of each safety station</td><td>0: Normal, Reserved station specified, unused station, standard station on CC-Link IE Field Network, or own station
1: Communication error of safety station on CC-Link IE Field Network</td><td>The communication status with each safety station is stored.
For standard stations, "0" is stored.
<table>
<tr><th></th><th>b15</th><th>b14</th><th>to</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SD1420</td><td>16</td><td>15</td><td>to</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SD1421</td><td>32</td><td>31</td><td>to</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SD1422</td><td>48</td><td>47</td><td>to</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SD1423</td><td>64</td><td>63</td><td>to</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
<tr><td>SD1424</td><td>80</td><td>79</td><td>to</td><td>74</td><td>73</td><td>72</td><td>71</td><td>70</td><td>69</td><td>68</td><td>67</td><td>66</td><td>65</td></tr>
<tr><td>SD1425</td><td>96</td><td>95</td><td>to</td><td>90</td><td>89</td><td>88</td><td>87</td><td>86</td><td>85</td><td>84</td><td>83</td><td>82</td><td>81</td></tr>
<tr><td>SD1426</td><td>112</td><td>111</td><td>to</td><td>106</td><td>105</td><td>104</td><td>103</td><td>102</td><td>101</td><td>100</td><td>99</td><td>98</td><td>97</td></tr>
<tr><td>SD1427</td><td>-</td><td>-</td><td>to</td><td>-</td><td>-</td><td>120</td><td>119</td><td>118</td><td>117</td><td>116</td><td>115</td><td>114</td><td>113</td></tr>
</table>
1 to 120 in the table indicate station numbers.
-: Fixed to "0"</td><td>S (Status change)</td><td>QS*1</td></tr>
<tr><td>SD1440
to
SD1560</td><td>Safety communication status of each safety station</td><td>Stores the communication status with each safety station.</td><td>• The communication status with each safety station is stored.
• SD1440: Station number 0 to SD1560: Station number 120
• (For reserved stations, unused stations, standard stations, or own station, "0" is stored.)
0: Safety refresh communication
10 to 15: Safety initial communication
30: Communication error or interlocked
8400: Safety communication target station detection error
8420, 8421, 8422, 8423, 8424, 8425: Safety monitoring timeout error
8430, 8431, 8432: Safety station reception data error
8440: Safety connection setting error</td><td>S (Status change)</td><td>QS*1</td></tr>
</table>

TableApp.3.9 Special register

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by (When set)</th><th>Corresponding CPU</th></tr>
<tr><td>SD1700 to SD1707</td><td>Safety station interlock status</td><td>0: Not interlocked
1: Interlocked</td><td>• When an error is detected and an interlock is activated, the bit corresponding to the error station turns to "1".
<table>
<tr><th></th><th>b15</th><th>b14</th><th>to</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SD1700</td><td>16</td><td>15</td><td>to</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SD1701</td><td>32</td><td>31</td><td>to</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SD1702</td><td>48</td><td>47</td><td>to</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SD1703</td><td>64</td><td>63</td><td>to</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
<tr><td>SD1704</td><td>80</td><td>79</td><td>to</td><td>74</td><td>73</td><td>72</td><td>71</td><td>70</td><td>69</td><td>68</td><td>67</td><td>66</td><td>65</td></tr>
<tr><td>SD1705</td><td>96</td><td>95</td><td>to</td><td>90</td><td>89</td><td>88</td><td>87</td><td>86</td><td>85</td><td>84</td><td>83</td><td>82</td><td>81</td></tr>
<tr><td>SD1706</td><td>112</td><td>111</td><td>to</td><td>106</td><td>105</td><td>104</td><td>103</td><td>102</td><td>101</td><td>100</td><td>99</td><td>98</td><td>97</td></tr>
<tr><td>SD1707</td><td>-</td><td>-</td><td>to</td><td>-</td><td>-</td><td>120</td><td>119</td><td>118</td><td>117</td><td>116</td><td>115</td><td>114</td><td>113</td></tr>
</table>
1 to 120 in the table indicate station numbers.
-: Fixed to "0"</td><td>S (Status change)</td><td>QS*1</td></tr>
<tr><td>SD1720 to SD1727</td><td>Safety station interlock release request</td><td>0: I/O interlock of safety station on CC-Link IE Field Network not released
1: I/O interlock of safety station on CC-Link IE Field Network released</td><td>• The interlock status of a safety station is released when the corresponding bit in this register is changed from "0" to "1".
<table>
<tr><th></th><th>b15</th><th>b14</th><th>to</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SD1720</td><td>16</td><td>15</td><td>to</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>SD1721</td><td>32</td><td>31</td><td>to</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td>SD1722</td><td>48</td><td>47</td><td>to</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr>
<tr><td>SD1723</td><td>64</td><td>63</td><td>to</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr>
<tr><td>SD1724</td><td>80</td><td>79</td><td>to</td><td>74</td><td>73</td><td>72</td><td>71</td><td>70</td><td>69</td><td>68</td><td>67</td><td>66</td><td>65</td></tr>
<tr><td>SD1725</td><td>96</td><td>95</td><td>to</td><td>90</td><td>89</td><td>88</td><td>87</td><td>86</td><td>85</td><td>84</td><td>83</td><td>82</td><td>81</td></tr>
<tr><td>SD1726</td><td>112</td><td>111</td><td>to</td><td>106</td><td>105</td><td>104</td><td>103</td><td>102</td><td>101</td><td>100</td><td>99</td><td>98</td><td>97</td></tr>
<tr><td>SD1727</td><td>-</td><td>-</td><td>to</td><td>-</td><td>-</td><td>120</td><td>119</td><td>118</td><td>117</td><td>116</td><td>115</td><td>114</td><td>113</td></tr>
</table>
1 to 120 in the table indicate station numbers.
-: Fixed to "0"</td><td>U (Request)</td><td>QS*1</td></tr>
</table>

*1: This applies when the serial number (first five digits) of the CPU module is "13042" or later.

I INDEX

MEMO

WARRANTY

Please confirm the following product warranty details before using this product.

1. Limited Warranty and Product Support.

a. Mitsubishi Electric Company ("MELCO") warrants that for a period of eighteen (18) months after date of delivery from the point of manufacture or one year from date of Customer's purchase, whichever is less, Mitsubishi MELSEC Safety programmable logic controllers (the "Products") will be free from defects in material and workmanship.

b. At MELCO's option, for those Products MELCO determines are not as warranted, MELCO shall either repair or replace them or issue a credit or return the purchase price paid for them.

c. For this warranty to apply:

(1) Customer shall give MELCO (i) notice of a warranty claim to MELCO and the authorized dealer or distributor from whom the Products were purchased, (ii) the notice shall describe in reasonable details the warranty problem, (iii) the notice shall be provided promptly and in no event later than thirty (30) days after the Customer knows or has reason to believe that Products are not as warranted, and (iv) in any event, the notice must given within the warranty period;

(2) Customer shall cooperate with MELCO and MELCO's representatives in MELCO's investigation of the warranty claim, including preserving evidence of the claim and its causes, meaningfully responding to MELCO's questions and investigation of the problem, grant MELCO access to witnesses, personnel, documents, physical evidence and records concerning the warranty problem, and allow MELCO to examine and test the Products in question offsite or at the premises where they are installed or used; and

(3) If MELCO requests, Customer shall remove Products it claims are defective and ship them to MELCO or MELCO's authorized representative for examination and, if found defective, for repair or replacement. The costs of removal, shipment to and from MELCO's designated examination point, and reinstallation of repaired or replaced Products shall be at Customer's expense.

(4) If Customer requests and MELCO agrees to effect repairs onsite at any domestic or overseas location, the Customer will pay for the costs of sending repair personnel and shipping parts. MELCO is not responsible for any re-commissioning, maintenance, or testing on-site that involves repairs or replacing of the Products.

d. Repairs of Products located outside of Japan are accepted by MELCO's local authorized service facility centers ("FA Centers"). Terms and conditions on which each FA Center offers repair services for Products that are out of warranty or not covered by MELCO's limited warranty may vary.

e. Subject to availability of spare parts, MELCO will offer Product repair services for (7) years after each Product model or line is discontinued, at MELCO's or its FA Centers' rates and charges and standard terms in effect at the time of repair. MELCO usually produces and retains sufficient spare parts for repairs of its Products for a period of seven (7) years after production is discontinued.

f. MELCO generally announces discontinuation of Products through MELCO's Technical Bulletins. Products discontinued and repair parts for them may not be available after their production is discontinued.

2. Limits of Warranties.

a. MELCO does not warrant or guarantee the design, specify, manufacture, construction or installation of the materials, construction criteria, functionality, use, properties or other characteristics of the equipment, systems, or production lines into which the Products may be incorporated, including any safety, fail-safe and shut down systems using the Products.

b. MELCO is not responsible for determining the suitability of the Products for their intended purpose and use, including determining if the Products provide appropriate safety margins and redundancies for the applications, equipment or systems into which they are incorporated.

c. Customer acknowledges that qualified and experienced personnel are required to determine the suitability, application, design, construction and proper installation and integration of the Products. MELCO does not supply such personnel.

d. MELCO is not responsible for designing and conducting tests to determine that the Product functions appropriately and meets application standards and requirements as installed or incorporated into the end-user's equipment, production lines or systems.

e. MELCO does not warrant any Product:

(1) repaired or altered by persons other than MELCO or its authorized engineers or FA Centers;

(2) subjected to negligence, carelessness, accident, misuse, or damage;

(3) improperly stored, handled, installed or maintained;

(4) integrated or used in connection with improperly designed, incompatible or defective hardware or software;

(5) that fails because consumable parts such as batteries, backlights, or fuses were not tested, serviced or replaced;

(6) operated or used with equipment, production lines or systems that do not meet applicable and commensurate legal, safety and industry-accepted standards;

(7) operated or used in abnormal applications;

(8) installed, operated or used in contravention of instructions, precautions or warnings contained in MELCO's user, instruction and/or safety manuals, technical bulletins and guidelines for the Products;

(9) used with obsolete technologies or technologies not fully tested and widely accepted and in use at the time of the Product's manufacture;

(10)subjected to excessive heat or moisture, abnormal voltages, shock, excessive vibration, physical damage or other improper environment; or

(11)damaged or malfunctioning due to Acts of God, fires, acts of vandals, criminals or terrorists, communication or power failures, or any other cause or failure that results from circumstances beyond MELCO's control.

f. All Product information and specifications contained on MELCO's website and in catalogs, manuals, or technical information materials provided by MELCO are subject to change without prior notice.

g. The Product information and statements contained on MELCO's website and in catalogs, manuals, technical bulletins or other materials provided by MELCO are provided as a guide for Customer's use. They do not constitute warranties and are not incorporated in the contract of sale for the Products.
h. These terms and conditions constitute the entire agreement between Customer and MELCO with respect to warranties, remedies and damages and supersede any other understandings, whether written or oral, between the parties. Customer expressly acknowledges that any representations or statements made by MELCO or others concerning the Products outside these terms are not part of the basis of the bargain between the parties and are not factored into the pricing of the Products.
i. THE WARRANTIES AND REMEDIES SET FORTH IN THESE TERMS ARE THE EXCLUSIVE AND ONLY WARRANTIES AND REMEDIES THAT APPLY TO THE PRODUCTS.
j. MELCO DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

3. Limits on Damages.

a. MELCO'S MAXIMUM CUMULATIVE LIABILITY BASED ON ANY CLAIMS FOR BREACH OF WARRANTY OR CONTRACT, NEGLIGENCE, STRICT TORT LIABILITY OR OTHER THEORIES OF RECOVERY REGARDING THE SALE, REPAIR, REPLACEMENT, DELIVERY, PERFORMANCE, CONDITION, SUITABILITY, COMPLIANCE, OR OTHER ASPECTS OF THE PRODUCTS OR THEIR SALE, INSTALLATION OR USE SHALL BE LIMITED TO THE PRICE PAID FOR PRODUCTS NOT AS WARRANTED.
b. Although MELCO has obtained the certification for Product's compliance to the international safety standards IEC61508 and EN954-1/ISO13849-1 from TUV Rheinland, this fact does not guarantee that Product will be free from any malfunction or failure. The user of this Product shall comply with any and all applicable safety standard, regulation or law and take appropriate safety measures for the system in which the Product is installed or used and shall take the second or third safety measures other than the Product. MELCO is not liable for damages that could have been prevented by compliance with any applicable safety standard, regulation or law.
c. MELCO prohibits the use of Products with or in any application involving power plants, trains, railway systems, airplanes, airline operations, other transportation systems, amusement equipments, hospitals, medical care, dialysis and life support facilities or equipment, incineration and fuel devices, handling of nuclear or hazardous materials or chemicals, mining and drilling, and other applications where the level of risk to human life, health or property are elevated.
d. MELCO SHALL NOT BE LIABLE FOR SPECIAL, INCIDENTAL, CONSEQUENTIAL, INDIRECT OR PUNITIVE DAMAGES, FOR LOSS OF PROFITS, SALES, OR REVENUE, FOR INCREASED LABOR OR OVERHEAD COSTS, FOR DOWNTIME OR LOSS OF PRODUCTION, FOR COST OVERRUNS, OR FOR ENVIRONMENTAL OR POLLUTION DAMAGES OR CLEAN-UP COSTS, WHETHER THE LOSS IS BASED ON CLAIMS FOR BREACH OF CONTRACT OR WARRANTY, VIOLATION OF STATUTE, NEGLIGENCE OR OTHER TORT, STRICT LIABILITY OR OTHERWISE.
e. In the event that any damages which are asserted against MELCO arising out of or relating to the Products or defects in them, consist of personal injury, wrongful death and/or physical property damages as well as damages of a pecuniary nature, the disclaimers and limitations contained in these terms shall apply to all three types of damages to the fullest extent permitted by law. If, however, the personal injury, wrongful death and/or physical property damages cannot be disclaimed or limited by law or public policy to the extent provided by these terms, then in any such event the disclaimer of and limitations on pecuniary or economic consequential and incidental damages shall nevertheless be enforceable to the fullest extent allowed by law.
f. In no event shall any cause of action arising out of breach of warranty or otherwise concerning the Products be brought by Customer more than one year after the cause of action accrues.
g. Each of the limitations on remedies and damages set forth in these terms is separate and independently enforceable, notwithstanding the unenforceability or failure of essential purpose of any warranty, undertaking, damage limitation, other provision of these terms or other terms comprising the contract of sale between Customer and MELCO.

4. Delivery/Force Majeure.

a. Any delivery date for the Products acknowledged by MELCO is an estimated and not a promised date. MELCO will make all reasonable efforts to meet the delivery schedule set forth in Customer's order or the purchase contract but shall not be liable for failure to do so.
b. Products stored at the request of Customer or because Customer refuses or delays shipment shall be at the risk and expense of Customer.
c. MELCO shall not be liable for any damage to or loss of the Products or any delay in or failure to deliver, service, repair or replace the Products arising from shortage of raw materials, failure of suppliers to make timely delivery, labor difficulties of any kind, earthquake, fire, windstorm, flood, theft, criminal or terrorist acts, war, embargoes, governmental acts or rulings, loss or damage or delays in carriage, acts of God, vandals or any other circumstances reasonably beyond MELCO's control.

5. Choice of Law/Jurisdiction.

These terms and any agreement or contract between Customer and MELCO shall be governed by the laws of the State of New York without regard to conflicts of laws. To the extent any action or dispute is not arbitrated, the parties consent to the exclusive jurisdiction and venue of the federal and state courts located in the Southern District of the State of New York. Any judgment there obtained may be enforced in any court of competent jurisdiction.

6. Arbitration.

Any controversy or claim arising out of, or relating to or in connection with the Products, their sale or use or these terms, shall be settled by arbitration conducted in accordance with the Center for Public Resources (CPR) Rules for Non-Administered Arbitration of International Disputes, by a sole arbitrator chosen from the CPR's panels of distinguished neutrals. Judgment upon the award rendered by the Arbitrator shall be final and binding and may be entered by any court having jurisdiction thereof. The place of the arbitration shall be New York City, New York. The language of the arbitration shall be English. The neutral organization designated to perform the functions specified in Rule 6 and Rules 7.7(b), 7.8 and 7.9 shall be the CPR.

Microsoft, Windows, Windows NT are registered trademarks of Microsoft Corporation in the United States and other countries.
Pentium and Celeron are trademarks of Intel Corporation in the United States and other countries.
Ethernet is a trademark of Xerox Co., Ltd. in the United States.
CompactFlash is a trademark of SanDisk Corporation.
VxWorks, Tornado, WindPower, WindSh and WindView are registered trademarks of Wind River Systems, Inc.
Other company names and product names used in this document are trademarks or registered trademarks of respective owners.

SH(NA)-080628ENG-D

SH(NA)-080628ENG-D(1105)MEE
MODEL: QSCPU-P-KY-E
MODEL CODE: 13JW01

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.