Panasonic®

PROGRAMMABLE CONTROLLER

FP7 CPU Unit

Command Reference Manual

[Applicable Models]
FP7 CPU Unit
- CPS4RE/CPS3RE/CPS3R/CPS2R
- CPS4RES/CPS3RES/CPS3RS

WUME-FP7CPUPGR-15

2024.4 | industry.panasonic.com/

(MEMO)

Introduction

Thank you for purchasing a Panasonic product. Before you use the product, please carefully read through the user’s manual, and understand it in detail to use the product properly.

Types of Manual

- There are different types of user’s manual for the FP7 series, as listed below. Please refer to a relevant manual for the unit and purpose of your use.
- The manuals can be downloaded from the Panasonic website:https://industry.panasonic.com/global/en/downloads/?tab=manual.

Unit name or purpose of use	Manual name	Manual code
FP7 Power Supply Unit FP7 CPU Unit	FP7 CPU Unit User’s Manual (Hardware)	WUME-FP7CPUH
FP7 CPU Unit	FP7 CPU Unit Command Reference Manual	WUME-FP7CPUPGR
	FP7 CPU Unit User’s Manual (Logging Trace Function)	WUME-FP7CPULOG
	FP7 CPU Unit User’s Manual (Security Function)	WUME-FP7CPUSEC
Instructions for Built-in LAN Port	FP7 CPU Unit User’s Manual (LAN Port Communication)	WUME-FP7LAN
	FP7 CPU Unit User’s Manual (Ethernet Expansion Function)	WUME-FP7CPUETEX
	FP7 CPU Unit User’s Manual (EtherNet/IP Communication)	WUME-FP7CPUEIP
	Web Server Function Manual	WUME-FP7WEB
Instructions for Built-in COM Port FP7 Extension Cassette (Communication) (RS-232C / RS485 type)	FP7 Series User’s Manual (SCU Communication)	WUME-FP7COM
FP7 Extension Cassette (Communication) (Ethernet Type)	FP7 Series User’s Manual (Communication Cassette Ethernet Type)	WUME-FP7CCET
FP7 Extension (Function) Cassette Analog Cassette	FP7 Analog Cassette User’s Manual	WUME-FP7FCA
FP7 Digital Input / Output Unit	FP7 Digital Input / Output Unit User’s Manual	WUME-FP7DIO
FP7 Analog Input Unit	FP7 Analog Input Unit User’s Manual	WUME-FP7AIH
FP7 Analog Output Unit	FP7 Analog Output Unit User’s Manual	WUME-FP7AOH
FP7 Thermocouple Multi-analog Input Unit FP7 RTD Input Unit	FP7 Thermocouple Multi-analog Input Unit FP7 RTD Input Unit User’s Manual	WUME-FP7TCRTD
FP7 Multi Input / Output Unit	FP7 Multi Input / Output Unit User’s Manual	WUME-FP7MXY
FP7 High-speed counter unit	FP7 High-speed Counter Unit User’s Manual	WUME-FP7HSC
FP7 Pulse Output Unit	FP7 Pulse Output Unit User’s Manual	WUME-FP7PG

Unit name or purpose of use	Manual name	Manual code
FP7 Positioning Unit	FP7 Positioning Unit User’s Manual	WUME-FP7POSP
FP7 Serial Communication Unit	FP7 Series User’s Manual (SCU Communication)	WUME-FP7COM
FP7 Multi-wire Link Unit	FP7 Multi-wire Link Unit User’s Manual	WUME-FP7MW
FP7 Motion Control Unit	FP7 Motion Control Unit User’s Manual	WUME-FP7MCEC
PHLS System	PHLS System User’s Manual	WUME-PHLS
Programming Software FPWIN GR7	FPWIN GR7 Introduction Guidance	WUME-FPWINGR7

FP7 Connector Compatibility

The connectors of old and new model FP7CPU units and add-on cassettes (hereinafter "cassettes") are shaped differently. Please use old model cassettes with old model units and new model cassettes with new model units as shown in the table below.

■ Old Model

Type	Old Product No.
CPU unit	AFP7CPS41ES, AFP7CPS41E, AFP7CPS31ES, AFP7CPS31E, AFP7CPS31S, AFP7CPS31, AFP7CPS21
Serial Communication Unit	AFP7NSC
Cassette	AFP7CCS1、AFP7CCS2、AFP7CCM1、AFP7CCM2、AFP7CCS1M1、AFP7CCET1、AFP7FCA21、AFP7FCAD2、AFP7FCTC2

■ New Model

Type	New Product No.
CPU unit	AFP7CPS4RES, AFP7CPS4RE, AFP7CPS3RES, AFP7CPS3RE, AFP7CPS3RS, AFP7CPS3R, AFP7CPS2R
Serial Communication Unit	AFP7NSCR
Cassette	AFP7CCRS1、AFP7CCRS2、AFP7CCRM1、AFP7CCRM2、AFP7CCRS1M1、AFP7CCRET1、AFP7FCRA21、AFP7FCRAD2、AFP7FCRTC2

Note

- Each FP7 unit can be connected to the CPU unit of a new or old model.
- Firmware version upgrades for the CPU unit are available for both new and old models.
- When attaching expansion cassettes to the FP7CPU unit, please use only old models, or only new models. Trying to attach a combination of old models and new models may cause damage.

Copyright and Trademarks

Handling Precautions

■ In this manual, the following symbols are used to indicate safety information that must be observed.

	Indicates an action that is prohibited or a matter that requires caution.
	Indicates an action that must be taken.

Info.	Indicates supplemental information.
Note	Indicates details about the subject in question or information useful to remember.
1 2 Procedure	Indicates operation procedures.

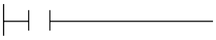
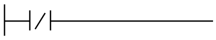
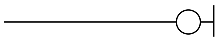
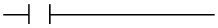
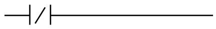
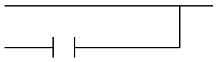
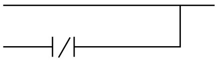
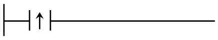
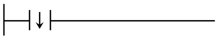
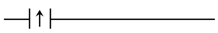
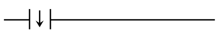
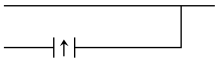
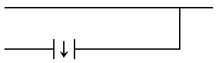
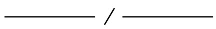
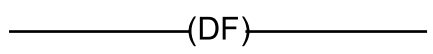
Table of Contents

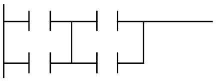
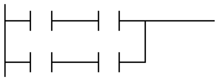
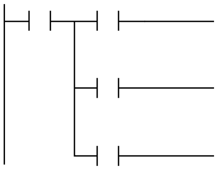
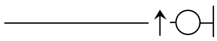
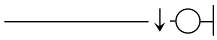
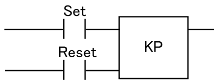
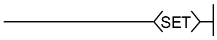
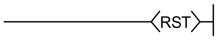
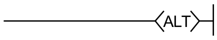
(MEMO)

1 List of Instructions

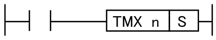
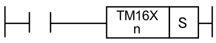
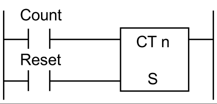
1.1 List of Basic Instructions

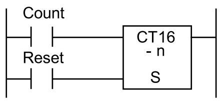
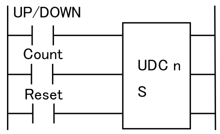
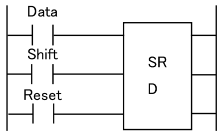
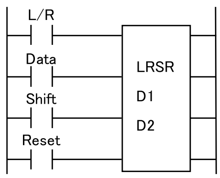
■ Sequence Basic Instructions

Name	Mnemonic	Symbol	Function overview	Page
Start	ST		Begins a logic operation with a normally open contact.	"P.3-3"
Start Not	ST/		Begins a logic operation with a normally closed contact.	"P.3-3"
Out	OT		Outputs the operated result.	"P.3-3"
AND	AN		Connects a normally open contact serially.	"P.3-5"
AND Not	AN/		Connects a normally closed contact serially.	"P.3-5"
OR	OR		Connects a normally open contact in parallel.	"P.3-6"
OR Not	OR/		Connects a normally closed contact in parallel.	"P.3-6"
Rising edge Contact instructions	ST↑		Begins a logic operation only for one scan when the leading edge of the trigger is detected.	"P.3-8"
Falling edge Contact instructions	ST↓		Begins a logic operation only for one scan when the trailing edge of the trigger is detected.	"P.3-8"
Rising edge Contact instructions	AN↑		Connects a normally open contact serially only for one scan when the leading edge of the trigger is detected.	"P.3-9"
Falling edge Contact instructions	AN↓		Connects a normally open contact serially only for one scan when the trailing edge of the trigger is detected.	"P.3-9"
Rising edge Contact instructions	OR↑		Connects a normally open contact in parallel only for one scan when the leading edge of the trigger is detected.	"P.3-10"
Falling edge Contact instructions	OR↓		Connects a normally open contact in parallel only for one scan when the trailing edge of the trigger is detected.	"P.3-10"
Not	/		Inverts the operated result up to this instruction.	"P.3-11"
Leading Edge Differential	DF		Turns ON the contact only for one scan when the leading edge of the trigger is detected.	"P.3-12"
Trailing Edge Differential	DF/		Turns ON the contact only for one scan when the trailing edge of the trigger is detected.	"P.3-12"
Leading Edge Differential (Initial Execution Type)	DFI		Turns ON the contact only for one scan when the leading edge of the trigger is detected. The leading edge detection is possible on the first scan.	"P.3-12"

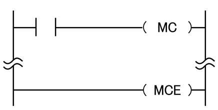
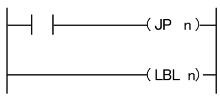
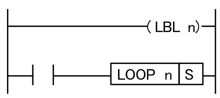
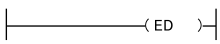
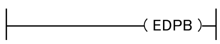
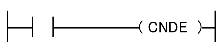
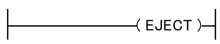
Name	Mnemonic	Symbol	Function overview	Page
AND Stack	ANS		Connects the multiple blocks serially.	"P.3-18"
OR Stack	ORS		Connects the multiple blocks in parallel.	"P.3-20"
Push Stack	PSHS		Stores the operated result up to this instruction.	"P.3-22"
Read Stack	RDS		Reads the operated result stored by the PSHS instruction.	"P.3-22"
Pop Stack	POPS		Reads and clears the operated result stored by the PSHS instruction.	"P.3-22"
Nop	NOP		No operation.	"P.3-26"
Rising edge Detection out	↑OT		Outputs the operated result to the specified output only for one scan when the leading edge of the signal is detected (for pulse relay P).	"P.3-27"
Falling edge Detection out	↓OT		Outputs the operated result to the specified output only for one scan when the trailing edge of the signal is detected (for pulse relay P).	"P.3-27"
Keep	KP		Turns ON at set input trigger and holds until reset trigger turns ON.	"P.3-29"
Set	SET		Output is set to and held at ON.	"P.3-31"
Reset	RST		Output is set to and held at OFF.	"P.3-31"
Alternate out	ALT		Inverts the output condition (ON/OFF) each time the leading edge of the trigger is detected.	"P.3-34"

■ Basic function instructions

Name	Mnemonic	Symbol	Function overview	Page
Timer (32-bit)	TM		On-delay timer. Decrements at the specified time [S]. When the elapsed value is 0, the timer contact turns ON. [S] is specified as 32-bit data (U1 to U4294967295). TMS: 0.01ms units/TML: 1ms units/TMR: 0.01s units/TMX: 0.1s units/TMY: 1s units	"P.3-36"
Timer (16-bit)	TM16		For TM16, [S] is specified as 16-bit data (U1 to U65535).	"P.3-46"
Unsigned 32-Bit Addition Support Timer	SPTM		Functions as an on-delay timer for 0.01 sec units.	"P.3-49"
Down counter (32 bit)	CT		Decrements from the preset value [S]. When the elapsed value is 0, the counter contact turns ON.	"P.3-53"

Name	Mnemonic	Symbol	Function overview	Page
			[S] is specified as 32-bit data (U1 to U4294967295).	
Down counter (16 bit)	CT16		For CT16, [S] is specified as 16-bit data (U1 to U65535).	"P.3-63"
Up/down counter	UDC		Increments or decrements from the preset value [S] based on up/down input. To be used with a comparison instruction described immediately after.	"P.3-67"
Shift register	SR		Shifts the content of the word device specified by [D] 1 bit to the left	"P.3-71"
Left/right shift register	LRSR		Shifts the data range specified by [D1] to [D2] to the left or to the right by 1 bit.	"P.3-75"

■ Control Instructions

Name	Mnemonic	Symbol	Function overview	Page
Master control relay	MC		When execution conditions are OFF, output between MC and MCE is turned OFF.	"P.3-78"
Master control relay end	MCE		Executes the program between MC and MCE instructions when the execution condition is ON.	"P.3-78"
Jump	JP		Jumps to the LBL instruction with the same number when the execution condition is ON and executes the program.	"P.3-83"
Label	LBL			"P.3-83"
Loop	LOOP		Jumps to the LBL instruction with the same number when the execution condition is ON and executes the program. Set the number of times to repeat the operation to [S].	"P.3-89"
Label	LBL			"P.3-89"
End	ED		Ends operation of the program. Indicates the end of the main program.	"P.3-93"
End Program Block	EDPB		Ends PB (program block).	"P.3-94"
Conditional End	CNDE		Ends operation of the program when the execution condition is ON.	"P.3-95"
Eject	EJECT		Makes a page break when printing out.	"P.3-96"

■ Step ladder instructions

Name	Mnemonic	Symbol	Function overview	Page
Start Step	SSTP	SSTP	Header for program "n" which is managed as a process.	"P.3-97"
Next Step	NSTL	NSTL	Starts the specified process "n" and clears other running processes. (Every scan execution type)	"P.3-97"
Clear Step	CSTP	CSTP	Clears running process "n".	"P.3-97"
Step End	STPE	STPE	End of step ladder area.	"P.3-97"
Block Clear	ZRST	ZRST D1 D2	Clears running processes [D1] to [D2].	"P.3-107"

■ Subroutine instructions

Name	Mnemonic	Symbol	Function overview	Page
Subroutine Label	SBL	(SBL n)	When CALL/FCALL/ECALL/EFCALL instruction is executed, it jumps to subroutines with the same label number and executes the subroutines. Use RET instruction to return to the address of the main program and execute the program.	"P.3-111"
Subroutine Return	RET	(RET)		"P.3-111"
Local Subroutine Call	CALL	(CALL n)	When the trigger (execution condition) is ON: subroutine is executed. When the trigger (execution condition) is OFF: subroutine is not executed. Holds the output from within subroutines.	"P.3-113"
Output OFF Type Local Subroutine Call	FCALL	(FCAL n)	When the trigger (execution condition) is ON: subroutine is executed. When the trigger (execution condition) is OFF: subroutine is not executed. However, the output from within subroutines is cleared.	"P.3-113"
Subroutine Call (with PB number Specification)	ECALL	(ECAL n PBm)	For subroutines in the PBn program block: When the trigger (execution condition) is ON: jumps to subroutine. When the trigger (execution condition) is OFF: subroutine is not executed.	"P.3-115"
Output OFF Type Subroutine Call (with PB number specification)	EFCALL	(EFCAL n PBm)	For subroutines in the PBn program block: When the trigger (execution condition) is ON: jumps to subroutine. When the trigger (execution condition) is OFF: subroutine is not executed. However, the output from within subroutines is cleared.	"P.3-117"

■ Interrupt Control Instructions

Name	Mnemonic	Symbol	Function overview	Page
Interrupt Program Definition	INTPG	INTPG	Head of the interrupt program.	"P.3-119"
Interrupt Return	IRET	IRET	End of the interrupt program.	"P.3-119"
CPU Interrupt Disable	DI	DI	Disables the interrupt to the CPU unit.	"P.3-124"
CPU Interrupt Enable	EI	EI	Enables the interrupt to the CPU unit.	"P.3-124"
Unit Interrupt Enable/ Disable	IMASK	IMASK.i S D	Controls to enable or disable the interrupt request from the unit.	"P.3-125"
Unit Interrupt Request Clear	ICLR	ICLR.i S D	Clears the interrupt request from the unit.	"P.3-127"

■ PB Execution Control Instructions

Name	Mnemonic	Symbol	Function overview	Page
PBn Execution Start	STARTPG	STARTPG PBn	Activates the waiting PB when the execution condition turns ON.	"P.3-129"
PBn Execution Stop	STOPPG	STOPPG PBn	Puts the active PB into waiting mode when the execution condition turns ON.	"P.3-129"
Global PB Number Setting	GPB	(GPB n)	Declares the global PB number (n=1000 to 1999) for the PB in which GPB instruction is written.	"P.3-132"

■ Commenting Instructions

Name	Mnemonic	Symbol	Function overview	Page
Comment Out	COMOUT	COMOUT	Comments out the instructions between COMOUT and ENDCOM instructions.	"P.3-134"
Comment Out End	ENDCOM	ENDCOM		"P.3-134"

■ Data comparison instructions

Name	Operation unit	Mnemonic	Symbol	Function overview	Page
Data Comparison (Start)	US, SS, UL, SL, SF, DF	ST =	S1 = S2	Begins a logic operation in the conducting contact if S1 = S2.	"P.3-135"
		ST <>	S1 <> S2	Begins a logic operation in the conducting contact if S1 ≠ S2.	"P.3-135"
		ST >	S1 > S2	Begins a logic operation in the conducting contact if S1 > S2.	"P.3-135"

Name	Operation unit	Mnemonic	Symbol	Function overview	Page
		ST >=	S1 >= S2	Begins a logic operation in the conducting contact if S1 ≥ S2.	"P.3-135"
		ST <	S1 < S2	Begins a logic operation in the conducting contact if S1 < S2.	"P.3-135"
		ST <=	S1 <= S2	Begins a logic operation in the conducting contact if S1 ≤ S2.	"P.3-135"
Data Comparison (AND)	US, SS, UL, SL, SF, DF	AN=	S1 = S2	The conducting contacts are connected in series if S1 = S2.	"P.3-138"
		AN<>	S1 <> S2	The conducting contacts are connected in series if S1 ≠ S2.	"P.3-138"
		AN>	S1 > S2	The conducting contacts are connected in series if S1 > S2.	"P.3-138"
		AN>=	S1 >= S2	The conducting contacts are connected in series if S1 ≥ S2.	"P.3-138"
		AN<	S1 < S2	The conducting contacts are connected in series if S1 < S2.	"P.3-138"
		AN<=	S1 <= S2	The conducting contacts are connected in series if S1 ≤ S2.	"P.3-138"
Data Comparison (OR)	US, SS, UL, SL, SF, DF	OR=	S1 = S2	The conducting contacts are connected in parallel if S1 = S2.	"P.3-141"
		OR<>	S1 <> S2	The conducting contacts are connected in parallel if S1 ≠ S2.	"P.3-141"
		OR>	S1 > S2	The conducting contacts are connected in parallel if S1 > S2.	"P.3-141"
		OR>=	S1 >= S2	The conducting contacts are connected in parallel if S1 ≥ S2.	"P.3-141"
		OR<	S1 < S2	The conducting contacts are connected in parallel if S1 > S2.	"P.3-141"
		OR<=	S1 <= S2	The conducting contacts are connected in parallel if S1 ≤ S2.	"P.3-141"

1.2 List of High-level Instructions

■ Data comparison instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Data compare	US, SS, UL, SL, SF, DF	CMP	(P)	S1, S2	Compares [S1] and [S2] and outputs the results to the system relay (SRA to SRC). (S1) > (S2) → SRA:ON (S1) = (S2) → SRB:ON (S1) < (S2) → SRC:ON	"P.4-2"
Band compare	US, SS, UL, SL, SF, DF	WIN	(P)	S1, S2,S3	Compares the value of [S1] with the lower limit value [S2] and the upper limit value [S3], and outputs the result to the system relay (SRA to SRC). (S1) > (S3) → SRA:ON (S2) ≤ (S1) ≤ (S3) → SRB:ON (S1) < (S2) → SRC:ON	"P.4-6"
Block comparison	–	BCMP	(P)	S1, S2, S3, S4	Compares the comparison block 1 specified by [S3] and the comparison block 2 specified by [S4] for the number of bytes specified by [S2] according to the control data specified by [S1], and outputs the result to the system relay (SRB). When the blocks match, SRB turns ON. When the blocks do not match, SRB turns OFF.	"P.4-9"

■ Data Transfer Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Data Transfer	US, SS, UL, SL, SF, DF	MV	(P)	S, D	(S) → (D)	"P.5-2"
Data Inversion and Transfer	US, SS, UL, SL, SF, DF	MV/	(P)	S, D	(S) → (D)	"P.5-4"
2 Data Transfer	US, SS, UL, SL, SF, DF	MV2	(P)	S1, S2, D	Transfers the two types of data specified by [S1] and [S2] to the area starting with [D].	"P.5-6"
3 Data Transfer	US, SS, UL, SL, SF, DF	MV3	(P)	S1, S2, S3, D	Transfers the three types of data specified by [S1], [S2], and [S3] to the area starting with [D].	"P.5-8"
Block Transfer	US, SS, UL, SL, SF, DF	BKMV	(P)	S1, S2, D	Transfers the data between [S1] and [S2] to the area starting with [D].	"P.5-10"

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Block Copy	US, SS, UL, SL, SF, DF	COPY	(P)	S, D1, D2	Transfers the data for [S1] to all of the areas between [D1] and [D2].	"P.5-13"
Bit Block Transfer	US, SS, UL, SL, SF, DF	BTM	(P)	S1, S2, D	Transfers the data between bit addresses [S1] and [S2] to the area ending with the low bit specified for [D].	"P.5-16"
Digit Data Transfer	–	DGT	(P)	S, S1, n, D, D1	The data for the [n] digit portion starting from the digit specified by [S] and [S1] is saved in the area starting from the digit specified by [D] and [D1].	"P.5-18"
Reset	US, SS, UL, SL, SF, DF	RST	(P)	D	The [D] area is reset to "0".	"P.5-22"
Block Clear	bit	ZRST	(P)	D1, D2	The area specified by bit addresses [D1] to [D2] is reset to "0".	"P.5-24"
16-bit Data Sign-extended Block Transfer	–	BKEXT	(P)	S1, S2, D	Performs sign extension for device values in the area specified by [S1] to [S2], and transfers them to the device address specified by [D] and subsequent addresses.	"P.5-26"
Block Transfer (32-bit Data to 16-bit Data)	–	BKMV16	(P)	S1, S2, D	Transfers only the lower one word of data in the area specified by [S1] to [S2] to an area after the area specified by [D] all at once.	"P.5-28"
Data exchange	US, SS, UL, SL, SF, DF	XCH	(P)	D1, D2	(D1) → (D2), (D2) → (D1)	"P.5-30"
Higher byte Exchange of lower bytes	US, SS	SWAP	(P)	S, D	Exchanges the higher and lower order bytes of [S] data and stores it in [D].	"P.5-32"
n Block Higher byte Exchange of lower bytes	–	BSWAP	(P)	S, n, D	Exchanges the high byte and low byte for [n] words from the device address specified by [S], and transfers it to the area starting with [D].	"P.5-34"
Specified PB Local Device Write	US, SS, UL, SL,	LCWT	(P)	S, n, PBm, D	Writes the data for [n] from the area specified by [S] to the area specified by [PBm]:[D] (local device) and subsequent areas all at once.	"P.5-36"
Specified PB Local Device Read	US, SS, UL, SL,	LCRD	(P)	PBm, S, n, D	Transfers the data for [n] from the area specified by [PBm]:[S] (local device) to the area specified by [D] and subsequent areas.	"P.5-38"

■ Index Register Operation Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Index Register Backup	UL, SL	PUSHIX	(P)	D	(I0) to (IE) → (D) to (D ＋ 29)	"P.5-42"
Index Register Recovery	UL, SL	POPIX	(P)	S	(S) to (S ＋ 29) → (I0) to (IE)	"P.5-44"

■ Arithmetic Operation Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Addition	US, SS, UL, SL, SF, DF	ADD	(P)	S1, S2, D	(S1) + (S2) → (D)	"P.6-2"
Subtraction	US, SS, UL, SL, SF, DF	SUB	(P)	S1, S2, D	(S1) - (S2) → (D)	"P.6-4"
Multiplication	US, SS, UL, SL, SF, DF	MUL	(P)	S1, S2, D	(S1) × (S2) → (D, D+n) "n" changes according to the operation unit.	"P.6-6"
Saturation Multiplication	US, SS, UL, SL,	MLCLIP	(P)	S1, S2, D	(S1) × (S2) → (D)	"P.6-9"
Division	US, SS, UL, SL, SF, DF	DIV	(P)	S1, S2, D	(S1) ÷ (S2) → Quotient (D)	"P.6-11"
Division (With a remainder)	US, SS, UL, SL	DIVMOD	(P)	S1, S2, D	(S1) / (S2) → Quotient (D), Remainder (D +n) "n" changes according to the operation unit.	"P.6-13"
Division (FP2 Compatible)	US, SS, UL, SL	DIVFP2	(P)	S1, S2, D	(S1) / (S2) → Quotient (D), Remainder (SD15, SD16)	"P.6-15"
Increment	US, SS, UL, SL, SF, DF	INC	(P)	D	(D) + 1 → (D)	"P.6-18"
Decrement	US, SS, UL, SL, SF, DF	DEC	(P)	D	(D) - 1 → (D)	"P.6-20"

■ BCD Data Arithmetic Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
BCD data Addition	US, UL	BCDADD	(P)	S1, S2, D	(S1) + (S2) → (D)	"P.6-22"
BCD data Subtraction	US, UL	BCDSUB	(P)	S1, S2, D	(S1) - (S2) → (D)	"P.6-24"
BCD data Multiplication	US, UL	BCDMUL	(P)	S1, S2, D	(S1) × (S2) → (D, D+n) "n" changes according to the operation unit.	"P.6-26"
BCD data Division	US, UL	BCDDIV	(P)	S1, S2, D	(S1) / (S2) → Quotient (D), Remainder (D +n) "n" changes according to the operation unit.	"P.6-28"
BCD data Increment	US, UL	BCDINC	(P)	D	(D) + 1 → (D)	"P.6-30"
BCD data Decrement	US, UL	BCDDEC	(P)	D	(D) + 1 → (D)	"P.6-32"

■ Boolean Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Logical conjunction	US, SS, UL, SL	AND	(P)	S1, S2, D	[S1] ∧ [S2] → (D)	"P.6-34"
Logical disjunction	US, SS, UL, SL	OR	(P)	S1, S2, D	[S1] ∨ [S2] → (D)	"P.6-36"
Exclusive OR	US, SS, UL, SL	XOR	(P)	S1, S2, D	{(S1) ∧ (S2) ∨ {(S1) ∧ (S2)} → (D)	"P.6-38"
Exclusive NOR	US, SS, UL, SL	XNR	(P)	S1, S2, D	{(S1) ∧ (S2) ∨ {(S1) ∧ (S2)} → (D)	"P.6-40"
Combination	US, SS, UL, SL	COMB	(P)	S1, S2, S3, D	{(S1) ∧ (S3)} ∨ {(S2) ∧/(S3)} → (D)	"P.6-42"

■ Data conversion instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Data Inversion	US, SS, UL, SL	INV	(P)	D	Performs logical inversion on each bit of data [D], and stores it in [D].	"P.7-2"
Sign Inversion	SS, SL	NEG	(P)	S, D	Performs sign inversion on data for [S], and stores it in [D].	"P.7-4"
Absolute Value	US, SS, UL, SL	ABS	(P)	S, D	Calculates the absolute value of the data for [S], and stores it in [D].	"P.7-6"

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Sign extension	US, SS	EXT	(P)	S, D	Converts the 16-bit BIN data for [S] to 32-bit BIN data while maintaining the signs, and stores it in (D+1, D).	"P.7-8"
Conversion: BCD Data	US, UL	BCD	(P)	S, D	Converts the BIN data specified with [S] to BCD data, and stores it in the area starting with [D]. Example) K100 → H100	"P.7-10"
Conversion: BCD → BIN	US, UL	BIN	(P)	S, D	Converts the BCD data specified with [S] to BIN data, and stores it in the area starting with [D]. Example) H100 → K100	"P.7-12"
Decode	–	DECO	(P)	S, n, D	Decodes part of the data of [S] and stores it in the area starting with [D]. The target for conversion is specified with "n".	"P.7-14"
7-Segment Decoding	US	SEGT	(P)	S, D	Converts the data of [S] to data for use in a 7-segment display, and stores it in (D+1, D).	"P.7-17"
Encode	–	ENCO	(P)	S, n, D	Encodes part of the data of [S], and stores it in the area starting with [D]. The target for conversion is specified with "n".	"P.7-20"
Digit Unification	US	UNIT	(P)	S, n, D	The least significant digits of the "n" words data of the 16-bit data starting with [S] are combined and stored in [D].	"P.7-23"
Digit Disintegration	US	DIST	(P)	S, n, D	Each digit of the data of [S] is separated and stored in each of the least significant digits of the 16-bit data starting with [D].	"P.7-25"
Byte Data Unification	–	BUNI	(P)	S, n, D	The least significant bytes of the "n" words data of the 16-bit data starting with [S] are combined and stored in the area starting with [D].	"P.7-27"
Byte Data Disintegration	–	BDIS	(P)	S, n, D	The data of the area starting with [S] is separated into byte units, and stored in the least significant bytes of the 16-bit data starting with [D].	"P.7-29"
Conversion: Binary → Gray code	US, UL	GRY	(P)	S, D	Converts the BIN data stored in [S] to gray code data, and stores it in the area starting with [D].	"P.7-31"
Conversion: Gray Code → BIN	US, UL	GBIN	(P)	S, D	Converts the gray code data stored in [S] to BIN data, and stores it in the area starting with [D].	"P.7-34"
Conversion: Bit Line → Bit Column	US	COLM	(P)	S, n, D	Stores the values for bits 0 to 15 of [S] in bits [n] of [D] to [D+15].	"P.7-36"
Conversion: Bit Column → Bit Line	US	LINE	(P)	S, n, D	Stores the values for bits [n] of [S] to [S+15] in bits 0 to 15 of [D].	"P.7-38"

■ Data shift instruction

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
n bits shifted to the right	US, SS, UL, SL	SHR	(P)	D, n	Shifts [n] bits of the data of [D] to the right.	"P.8-2"
n bits shifted to the left	US, SS, UL, SL	SHL	(P)	D, n	Shifts [n] bits of the data of [D] to the left.	"P.8-4"
n-digit Right Shift	US, SS, UL, SL	BSR	(P)	D, n	Shifts [n] digits of the data of [D] to the right.	"P.8-6"
n-digit Left Shift	US, SS, UL, SL	BSL	(P)	D, n	Shifts [n] digits of the data of [D] to the left.	"P.8-8"
Right shift of multiple devices for n bits	bit	BITR	(P)	D1, D2, n	Shifts [n] bits of the area of [D1] to [D2] to the right.	"P.8-10"
n-bit Left Shift of Multiple Devices	bit	BITL	(P)	D1, D2, n	Shifts [n] bits of the area of [D1] to [D2] to the left.	"P.8-12"
Right shift of block area for n words	US, SS	WSHR	(P)	D1, D2, n	Shifts [n] words of the area of [D1] to [D2] to the right.	"P.8-14"
Left shift of block area for n words	US, SS	WSHL	(P)	D1, D2, n	Shifts [n] words of the area of [D1] to [D2] to the left.	"P.8-16"
n-digit Right Shift of Block Area	US, SS	WBSR	(P)	D1, D2, n	Shifts [n] digits of the area of [D1] to [D2] to the right.	"P.8-18"
n-digit Left Shift of Block Area	US, SS	WBSL	(P)	D1, D2, n	Shifts [n] digits of the area of [D1] to [D2] to the left.	"P.8-20"

■ Data rotation instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Right Rotation of Data	US, UL	ROR	(P)	D, n	Rotates [n] bits of the data of [D] to the right.	"P.8-22"
Left Rotation of Data	US, UL	ROL	(P)	D, n	Rotates [n] bits of the data of [D] to the left.	"P.8-24"
Right Rotation of Data with Carry-Flag Data	US, UL	RCR	(P)	D, n	Rotates the area consisting of [D] plus the carry flag (SR9) data to the right by [n] bits.	"P.8-26"
Left Rotation of Data with Carry-Flag Data	US, UL	RCL	(P)	D, n	Rotates the area consisting of [D] plus the carry flag (SR9) data to the left by [n] bits.	"P.8-28"

■ Data buffer instruction

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Compress shift read	US, SS	CMPR	(P)	D1, D2, D3	Transfer [D2] to [D3]. Any parts of the data between [D1] and [D2] that are 0 are compressed, and shifted in order toward [D2].	"P.8-30"
Compress shift write	US, SS	CMPW	(P)	S, D1, D2	Transfer [S] to [D1]. Any parts of the data between [D1] and [D2] that are 0 are compressed, and shifted in order toward [D2].	"P.8-32"
Buffer Definition	US, SS	DEFBUF	(P)	n, D	Defines the area of [n] words starting from [D] as the data buffer area to be used for FIFR/BUFW/LIFR instruction.	"P.8-34"
Data read (First-in-first-out)	US, SS	FIFR	(P)	S, D	Reads data from the area indicated by the read pointer of the FIFO buffer starting from [S], and stores it in [D].	"P.8-37"
Data Write	US, SS	BUFW	(P)	S, D	Stores the value of [S] in the area indicated by the write pointer of the buffer starting with [D].	"P.8-40"
Data read (Last-in-first-out)	US, SS	LIFR	(P)	S, D	Reads data from the area indicated by the LIFO pointer of the LIFO buffer starting with [S], and stores it in [D].	"P.8-42"
Ring Buffer Definition	–	DEFRBUF	(P)	n, D	Defines the area of [n] words starting with [D] as the data buffer area to be used for RBUFW instruction.	"P.8-45"
Write to Ring Buffer, Calculation of Total Value and Moving Average Value	US, SS	RBUFW	(P)	S, D	Reads data from the area indicated by the read pointer of the FIFO buffer starting from [S], and stores it in [D].	"P.8-47"

■ Bit Manipulation Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
16-bit Data Specified Bit Set	US	BTS	(P)	D, n	Turns ON (1) the [n]th bit in the area specified by [D].	"P.9-2"
16-bit Data Specified Bit Reset	US	BTR	(P)	D, n	Turns OFF (0) the [n]th bit in the area specified by [D].	"P.9-4"
Bit Inversion	US	BTI	(P)	D, n	Invert the value of bit position [n] of the data of [D].	"P.9-6"
Bit Test	US	BTT	(P)	D, n	When the value of bit [n] of the data of [D] is 0, the system relay (SRB) turns ON. When the value is 1, the system relay (SRB) turns OFF.	"P.9-8"
Carry-flag set	–	STC	(P)		Turns ON the carry flag (SR9).	"P.9-10"

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Carry-Flag Reset	–	CLC	(P)		Turns OFF the carry flag (SR9).	"P.9-11"

■ Data Processing Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Data Search	US, SS, UL, SL, SF, DF	SRC	(P)	S1, S2, S3, D	Searches for the data of [S1] from the data stored in [S2] to [S3], and stores the matching articles and the relative position of where the data first matched in the area starting with [D].	"P.10-2"
ON Bits Count	US, UL	BCU	(P)	S, D	Counts the number of ON bits in the data of [S], and stores the number in [D].	"P.10-5"
Obtainment of the Max. Value	US, SS, UL, SL, SF, DF	MAX	(P)	S1, S2, D	Searches for the maximum value of the data stored in [S1] to [S2], and stores the maximum value and the relative position of where the maximum value was first detected in the area starting with [D].	"P.10-7"
Obtainment of the Min. value	US, SS, UL, SL, SF, DF	MIN	(P)	S1, S2, D	Searches for the minimum value of the data stored in [S1] to [S2], and stores the minimum value and the relative position of where the minimum value was first detected in the area starting with [D].	"P.10-13"
Obtainment of the Total and the Mean	US, SS, UL, SL, SF, DF	MEAN	(P)	S1, S2, D	Calculates the total and the average of the data stored in [S1] to [S2], and stores it in the area starting with [D].	"P.10-19"
Sort	US, SS, UL, SL, SF, DF	SORT	(P)	S1, S2, S3	Changes the order of the data stored in [S1] to [S2]. The sorting conditions are specified in [S3] (descending order or ascending order).	"P.10-24"
Linearization	US, SS, UL, SL, SF, DF	SCAL	(P)	S1, S2, D	Creates a data table used for scaling (linearization) in the area starting with [S2]. Scales the data of [S1] (input value X), and stores it in [D] (output value Y).	"P.10-27"
Variance and Standard Deviation Acquisition	US, SS	STDDEV	(P)	S, n, D	Stores the variance and standard deviation within the range of the device area specified by [S] and [n] into the device area specified by [D].	"P.10-30"

■ Event Count Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Instruction to Count the No. of Events	UL	EVENTC		S1, S2, n, D	Counts the number of ON times for the [n] bits of the data specified by [S1], and stores it in the area starting with [D].	"P.10-33" (Note 1)

Name	Operation Unit	Mnemonic	Operand	Function overview	Page
Instruction to Count the Time of Events	UL	EVENTT	S1, S2, n, D	Records the ON time in seconds units for the [n] bits of the data specified by [S1], and stores it in the area starting with [D].	"P.10-36" (Note 1)

(Note 1) Since EVENTC instruction and EVENTT instruction hold the operation description, it is necessary to set the execution condition to always ON.

■ PID Instructions

Name	Operation Unit	Mnemonic	Operand	Function overview	Page
PID Operation	−	PID	S	Executes PID operation according to the parameters of [S] to [S+29]. The operation results are stored in [S+3] as manipulated value MV.	"P.10-39" (Note 1)
PID Operation: PWM Output Available	−	EZPID	S1, S2, S3, S4	Executes PID operation according to the parameters of [S1], [S2], [S3] to [S3+3], and [S4] to [S4+29]. The operation results are stored in [S4] as manipulated value MV. An OUT instruction is written immediately after, and PWM output becomes possible.	"P.10-45" (Note 1)

(Note 1) Since PID instruction and EZPID instruction hold the operation description, it is necessary to set the execution condition to always ON.

■ Data Control Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Data Revision Detection	US, SS, UL, SL, SF, DF	DTR	(P)	S, D	Detects revisions in the data of [S], and reflects it to the carry flag (SR9). [D] is used as an area for holding the previous data values.	"P.10-55"
Ramp Output	US, SS, UL, SL, SF, DF	RAMP		S1, S2, S3, D	Executes the linear output for the specified time [S3], from the specified initial value [S1] to the target value [S2].	"P.10-57"
Upper and Lower Limit Control	US, SS, UL, SL, SF, DF	LIMT	(P)	S1, S2, S3, D	Depending on whether the input value [S3] is within the range of the upper limit [S1] and the lower limit [S2], calculates the output and stores it in [D]. When (S1) > (S3), (S1) → (D) When (S2) < (S3), (S2) → (D) When (S1) ≤ (S3) ≤ (S2), (S3) → (D)	"P.10-59"
Deadband Control	SS, SL, SF, DF	BAND	(P)	S1, S2, S3, D	Depending on whether the input value [S3] is within the range of the deadband specified by [S1] and [S2], calculates the output and stores it in [D]. When (S1) > (S3), (S3)-(S1) → (D) When (S2) < (S3), (S3)-(S2) → (D)	"P.10-62"

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
					When (S1) ≤ (S3) ≤ (S2), 0 → (D)	
Zone Control	US, SS, UL, SL, SF, DF	ZONE	(P)	S1, S2, S3, D	Depending on the range of input value [S3], adds up the biases specified by [S1] or [S2], and stores the result in [D]. When (S3) < 0, (S3) + (S1) → (D) When (S3) = 0, 0 → (D) When (S3) > 0, (S3) + (S2) → (D)	"P.10-64"

(Note 1) Since RAMP instruction and FILTER instruction hold the operation description, it is necessary to set the execution condition to always ON.

■ Time constant processing instructions

Name	Operation Unit	Mnemonic	Operand	Function overview	Page
Time Constant Processing	–	FILTR	S1, S2, S3, D	Performs the filtering process on the data specified by [S1] and [S2], and stores it in [D].	"P.10-67"

(Note 1) Since RAMP instruction and FILTER instruction hold the operation description, it is necessary to set the execution condition to always ON.

■ Floating Point Real Number Function Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Sine Operation	SF, DF	SIN	(P)	S, D	SIN (S) → (D)	"P.11-3"
Cosine Operation	SF, DF	COS	(P)	S, D	COS (S) → (D)	"P.11-5"
Tangent Operation	SF, DF	TAN	(P)	S, D	TAN (S) → (D)	"P.11-7"
Arcsine Operation	SF, DF	ASIN	(P)	S, D	SIN^{-1}(S) → (D)	"P.11-9"
Arccosine Operation	SF, DF	ACOS	(P)	S, D	COS^{-1}(S) → (D)	"P.11-11"
Arctangent Operation	SF, DF	ATAN	(P)	S, D	TAN^{-1}(S) → (D)	"P.11-13"
Conversion: Coordinate Data → Angle-Radian	SF, DF	ATAN2	(P)	S1, S2, D	Determines the arctangent from [S1] (X coordinate) and [S2] (Y coordinate), and stores it in [D].	"P.11-15"
Hyperbolic Sine Operation	SF, DF	SINH	(P)	S, D	SINH (S) → (D)	"P.11-17"
Hyperbolic Cosine Operation	SF, DF	COSH	(P)	S, D	COSH (S) → (D)	"P.11-19"

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Hyperbolic Tangent Operation	SF, DF	TANH	(P)	S, D	TANH (S) → (D)	"P.11-21"
Exponential Operation	SF, DF	EXP	(P)	S, D	EXP (S) → (D)	"P.11-23"
Natural Logarithmic Operation	SF, DF	LN	(P)	S, D	LN (S) → (D)	"P.11-25"
Common Logarithmic Operation	SF, DF	LOG	(P)	S, D	LOG (S) → (D)	"P.11-27"
Power Operation	SF, DF	PWR	(P)	S1, S2, D	(S1) ^ (S2) → (D)	"P.11-29"
Square Root Operation	SF, DF	SQR	(P)	S, D	SQR (S) → (D)	"P.11-31"

(Note 1) Since RAMP instruction and FILTER instruction hold the operation description, it is necessary to set the execution condition to always ON.

■ Floating Point Real Number Conversion Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Conversion: Degrees → Radian	SF, DF	RAD	(P)	S, D	Converts the angle data in degrees of [S] to angle data in radians, and stores it in [D].	"P.11-33"
Conversion: Radian → Degrees	SF, DF	DEG	(P)	S, D	Converts the angle data in radians of [S] to angle data in degrees, and stores it in [D].	"P.11-35"
Floating Point Real Number Data - Rounding the First Decimal Point Down	SF, DF	FINT	(P)	S, D	Rounds down the first decimal point of the real numbers stored in [S], and stores them in [D].	"P.11-37"
Floating Point Real Number Data - Rounding the First Decimal Point Off	SF, DF	FRINT	(P)	S, D	Rounds off the first decimal point of the real numbers stored in [S], and stores them in [D].	"P.11-39"
Floating Point Real Number Data - Sign Changes (Negative/ Positive Conversion)	SF, DF	FNEG	(P)	S, D	Changes the signs of the real numbers stored in [S], and stores them in [D].	"P.11-41"

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Floating point Real number data Absolute Value	SF, DF	FABS	(P)	S, D	Determines the absolute values of the real numbers stored in [S], and stores them in [D].	"P.11-43"
Conversion: Single-precision Real Number Data → Double-precision Real Number	SF	STOD	(P)	S, D	Converts the single-precision floating point real number stored in the area starting with [S] to a double-precision floating point real number.	"P.11-45"
Conversion: Double-precision Real Number Data → Single-precision Real Number	DF	DTOS	(P)	S, D	Converts the double-precision floating point real number stored in the area starting with [S] to a single-precision floating point real number.	"P.11-47"
Separation of Mantissa and Exponent of Single-precision or Double-precision Real Number Data	SF, DF	DISF	(P)	S, D1, D2	Separates the floating point real number data stored in the area starting with [S] into mantissa and exponent according to the operation unit [i].	"P.11-49"
Combining of Mantissa and Exponent, and Conversion of Single-precision or Double-precision Real Number	SF, DF	UNIF	(P)	S1, S2, D	Combines the data of mantissa and exponent stored in the areas starting with [S1] and [S2] according to the operation unit [i].	"P.11-52"
Conversion: Integer → Single-Precision Floating Point Real Number Data	US, SS, UL, SL	FLT	(P)	S, D	Converts the integer data stored in [S] to single-precision floating point real number data, and stores it in [D].	"P.11-55"
Conversion: Integer → Double-precision Real Number Data	US, SS, UL, SL	DFLT	(P)	S, D	Converts [S] (signed 32-bit integer data) to real number data, and stores it in [D].	"P.11-58"
Conversion: Single-precision Floating Point Real Number Data → Integer	US, SS, UL, SL	INT	(P)	S, D	Converts the single-precision floating point real numbers stored in [S] to integers (maximum value cannot be exceeded), and stores them in [D].	"P.11-61"

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
(The largest integer not exceeding the floating point type data)						
Conversion: Double-Precision Real Number Data → Integer (The largest integer not exceeding the floating point type data)	US, SS, UL, SL	DINT	(P)	S, D	Converts [S] (real data) to a signed-32-bit integer (maximum value cannot be exceeded), and stores it in [D].	"P.11-64"
Conversion: Single-precision Floating Point Real Number Data → Integer (Rounding the First Decimal Point Down)	US, SS, UL, SL	FIX	(P)	S, D	Converts the single-precision floating point real numbers stored in [S] to integers (rounding the first decimal point down), and stores them in [D].	"P.11-68"
Conversion: Double-Precision Real Number Data → Integer (Rounding the First Decimal Point Down)	US, SS, UL, SL	DFIX	(P)	S, D	Converts [S] (real number data) to a signed-32-bit integer (rounding the first decimal point down), and stores it in [D].	"P.11-71"
Conversion: Single-precision Floating Point Real Number Data → Integer (Rounding to the Nearest Unit)	US, SS, UL, SL	ROFF	(P)	S, D	Converts the single-precision floating point real numbers stored in [S] to integers (rounding the first decimal point off), and stores them in [D].	"P.11-75"
Conversion: Double-Precision Real Number Data → Integer (Rounding to the Nearest Unit)	US, SS, UL, SL	DROFF	(P)	S, D	Converts [S] (real number data) to a signed-32-bit integer (rounding the first decimal point off), and stores it in [D].	"P.11-78"

■ Clock/Calendar Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Conversion: Time Data (Hours, Minutes and Seconds) → Seconds Data	–	HMSS	(P)	S, D	Converts the BIN data stored in [S] to [S+2] which indicates the hour, minute, second into seconds units, and stores it in (D+1, D).	"P.12-2"
Conversion: Seconds Data → Time Data (Hours, Minutes and Seconds)	–	SHMS	(P)	S, D	Converts the seconds data stored in [S+1, S] to hour, minute, second, and stores it in [D] to [D+2] as BIN data.	"P.12-4"
Clock Addition	–	CADD	(P)	S1, S2, D	Adds the time data (hour, minute, second) stored in [S2] to [S2+2] to the clock data (year, month, date, hour, minute, second) stored in [S1] to [S1+5], and stores it in [D] to [D+5].	"P.12-6"
Clock Subtraction	–	CSUB	(P)	S1, S2, D	Subtracts the time data (hour, minute, second) stored in [S2] to [S2+2] from the clock data (year, month, date, hour, minute, second) stored in [S1] to [S1+5], and stores it in [D] to [D+5].	"P.12-8"
Clock Data → Seconds Data from the Base Time	–	TMSEC	(P)	S, D	Converts the clock data specified by [S] to seconds data from the base time (Jan 1, 2001), and stores it in (D+1, D).	"P.12-10"
Seconds Data from the Base Time → Clock Data	–	SECTM	(P)	S, D	Converts the seconds data from the base time (Jan 1, 2001) stored in [S+1, S] to clock data, and stores it in [D] to [D+5].	"P.12-12"
Setting of Clock/ Calendar	–	TIMEWT	(P)	S	Sets the clock data (year, month, date, hour, minute, second) stored in [S] to [S+6] as RTC data in CPU units.	"P.12-14"
Summer Time Acquisition	–	SUMMER	(P)	S1, S2, S3, D	Sets the clock data corrected by [S3] (time difference) in [D] (valid/invalid flag + time data) for the period specified by the period of [S1] to [S2] (month, date, hour, minute).	"P.12-16"

■ Logging Trace Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Logging trace start request	US, SS	LOGST	(P)	n	Requests to start the logging trace function for the number specified by [n].	"P.13-2"
Logging trace stop request	US, SS	LOGED	(P)	n	Requests to stop the logging trace function for the number specified by [n].	"P.13-4"
Sampling Trace	US, SS	SMPL	(P)	n	Performs sampling during tracing.	"P.13-6"

■ Special instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Self-diagnostic error Code Set	US	ERR	(P)	n	Sets the arbitrary error code [n] with the user program. Stores the self-diagnostic error code in the system data register (SD0). Sets the self-diagnosis error occurrence flag (SR0).	"P.13-13"
Watchdog Timer Reset	–	WDTRES	(P)		Resets the watchdog timer.	"P.13-15"
System Area Copy	–	SCOPY	(P)	S1, S2, S3, D	Copies data in the area specified by [S1], [S2], and [S3] to the area specified by [D].	"P.13-16"
Starting Word Number Acquisition of Target Slot	–	GETSTNO	(P)	S, D	The starting word number of a specified slot is obtained.	"P.13-18"

■ Positioning Unit Control Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Setting of Positioning Starting Table	–	POSSET	(P)	S1, S2, S3	Sets the positioning data table to be started, described right before the program which starts positioning. Specify Slot No., Axis No., Table No. for [S1], [S2], [S3].	"P.13-19"
Acquire Axis Status	–	PSTRD	(P)	S1, S2, D	Reads the main flag status as the axis status for Axis No. [S2] of the start positioning unit equipped to Slot No. [S1], and stores it in [D].	"P.13-21"
Acquire Positioning Unit Error/ Warning	–	PERRD	(P)	S1, S2, D	Reads the error codes and warning codes stored in the annunciation buffer 1 for Axis No. [S2] of the start positioning unit equipped to Slot No. [S1], and stores them in [D] and [D1].	"P.13-23"

■ Unit Control Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Error and Warning Clear	–	UCLR	(P)	S	Clears the errors and warnings for the unit equipped to the slot number specified by [S].	"P.13-26"

■ Character String Conversion Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Block Check	US, SS	BCC	(P)	S1, S2,	Calculates the block check code for the data specified by [S2] and [S3], and stores	"P.14-3"

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Code Calculation				S3, D	it in the area starting with [D]. The calculation method is specified by [S1].	
CRC Code Calculation	US, SS	CRC	(P)	S1, S2, S3, D	Calculates the CRC code for the data specified by [S2] and [S3], and stores it in the area starting with [D]. The calculation method is specified by [S1].	"P.14-6"
Conversion: HEX → Hexadecimal ASCII	US, SS, UL, SL	HEXA	(P)	S1, S2, D	Converts the hexadecimal data specified by [S1] and [S2] to ASCII code, and stores it in the area starting with [D]. Example) HABCD → H42414443B A D C	"P.14-10"
Conversion: Hexadecimal ASCII → HEX	US, SS, UL, SL	AHEX	(P)	S1, S2, D	Converts the ASCII code specified by [S1] and [S2] to hexadecimal data, and stores it in the area starting with [D]. Example) H44434241→ HCDABD C B A	"P.14-13"
Conversion: BCD → Decimal ASCII	US, UL	BCDA	(P)	S1, S2, D	Converts the BCD data specified by [S1] and [S2] to ASCII code, and stores it in the area starting with [D]. The conversion method is specified by [S2]. Example 1) H1234 → H32313433 2 1 4 3 Example 2) H1234 → H34333231 4 3 2 1	"P.14-16"
Conversion: Decimal ASCII → BCD	US, UL	ABCD	(P)	S1, S2, D	Converts the ASCII code specified by [S1] and [S2] to BCD data, and stores it in the area starting with [D]. The conversion method is specified by [S2]. Example 1) H34333231→ H 3412 4 3 2 1 Example 2) H34333231→ H 1234 4 3 2 1	"P.14-19"
Conversion: BIN → Decimal ASCII	US, SS, UL, SL	BINA	(P)	S1, S2, D	Converts the BIN data specified by [S1] which indicates a decimal to ASCII code, and stores it in the area starting with [D]. The number of bytes for the conversion result is specified by [S2]. Example) K-100 → H3030312D2020 0 0 1 -	"P.14-23"
Conversion: Decimal ASCII → BIN	US, SS, UL, SL	ABIN	(P)	S1, S2, D	Converts the ASCII code stored in the area starting with [S1] to BIN data which indicates a decimal, and stores it in the area starting with [D]. The number of bytes to convert is specified by [S2]. Example) H3030312[D]2020→ K-100 0 0 1 -	"P.14-26"
Conversion: BIN → ASCII	US, SS, UL, SL, SF, DF	BTOA	(P)	S1, S2, N, D	Converts the BIN data stored in the area starting with [S2] to ASCII data, and stores it in the area starting with [D]. The conversion method is specified by [S1] and [N].	"P.14-30"

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
ASCII data Check	US, SS, UL, SL, SF, DF	ACHK	(P)	S1, S2, N	Checks whether ASCII data converted with ATOB instruction can be converted under normal conditions.	"P.14-44"
Conversion: ASCII → BIN	US, SS, UL, SL, SF, DF	ATOB	(P)	S1, S2, N, D	Converts the ASCII data stored in the area starting with [S2] to BIN data, and stores it in the area starting with [D]. The conversion method is specified by [S1] and [N].	"P.14-46"
Conversion: Character Constant → ASCII Code	–	SSET	(P)	S, D	Converts the character constant specified by [S] to ASCII code, and stores it in the area starting with [D].	"P.14-60"
Text creation	–	PRINT	(P)	S1, S2, D	Inserts the data to be output [S2] to the text creation form [S1], creates the text, and stores it in the area specified by [D].	"P.14-64"
Date and Time Character String Conversion	–	TIMEstr	(P)	S1, S2, D	Converts the date and time information specified by [S1] according to the conversion pattern specified by [S2], and stores the character string in the storage location specified by [D].	"P.14-75"

■ String operation instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
String compare	–	SCMP	(P)	S1, S2	Compares the character string of [S1] with the character string of [S2], and outputs the result to the system relay (SRA to SRC). (S1) > (S2) → SRA:ON (S1) = (S2) → SRB:ON (S1) < (S2) → SRC:ON	"P.14-81"
String Addition	–	SADD	(P)	S1, S2, D	Couples the character string of [S1] with the character string of [S2], and stores it in [D].	"P.14-83"
Obtainment of String Length	–	LEN	(P)	S, D	The number of characters stored in the character string table data starting with [S] is stored in [D].	"P.14-85"
Search String Length (Terminating NULL)		LENGTH	(P)	S1, S2, D	Detects a termination character (null) from a string and acquires the number of characters.	"P.14-87"
String Search	–	SSRC	(P)	S1, S2, D	Searches for data [S1] from the character string table data starting with [S2], and the relative position of the first match is stored in [D].	"P.14-89"
Takeout of the Right Side of a String	–	RIGHT	(P)	S1, S2, D	Retrieves the number of characters [S2] from the right side (the side where the device address is larger) of the character string table starting with [S1], and stores it in [D].	"P.14-92"

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Takeout of the Left Side of a String	–	LEFT	(P)	S1, S2, D	Retrieves the character strings of [S2] number of characters from the left side (the side where the device address is smaller) of the character string table starting with [S1], and stores it in [D].	"P.14-94"
Data Read from a Given Position in the String	–	MIDR	(P)	S1, S2, S3, D	Reads the character strings for [S3] number of character strings from the [S2] byte of the character string table starting with [S1], and stores it in [D].	"P.14-96"
Rewrite from a Given Position in the String	–	MIDW	(P)	S1, S2, D, n	Reads the character string data for [S2] number of characters from the character string table starting with [S1], and stores it in [D] from the position of the [n]th byte.	"P.14-98"
Replacement of a Character String	–	SREP	(P)	S, D, p, n	Replaces the character string table starting with [D] with the character strings of [S1]. The positions and number of characters that can be replaced are specified with [p] and [n].	"P.14-101"

■ Character String Conversion Instruction (With Storage Area Size)

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Conversion: Character Constant → ASCII Code	–	ESSET	(P)	S, D	Converts the character constant specified by [S] to ASCII code, and stores it in the area starting with [D]. The storage area size is specified at the head.	"P.14-104"
Text Creation (With Storage Area Size)	–	EPRINT	(P)	S1, S2, D	Inserts the data to be output [S2] to the text creation form [S1], creates the text, and stores it in the area specified by [D]. The storage area size is specified at the head.	"P.14-107"
Date and Time Character String Conversion (With Storage Area Size)	–	ETIMEstr	(P)	S1, S2, D	Converts the date and time information specified by [S1] according to the conversion pattern specified by [S2], and stores the character string in the storage location specified by [D]. The storage area size is specified at the head.	"P.14-115"

■ String Operation Instruction (With Storage Area Size)

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
String compare	–	ESCMP	(P)	S1, S2	Compares the character string of [S1] with the character string of [S2], and outputs the result to the system relay (SRA to SRC). (S1) > (S2) → SRA:ON (S1) = (S2) → SRB:ON (S1) < (S2) → SRC:ON	"P.14-119"

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
String Addition	–	ESADD	(P)	S1, S2, D	Couples the character string of [S1] with the character string of [S2], and stores it in [D].	"P.14-122"
Obtainment of String Length	–	ELEN	(P)	S, D	The number of characters stored in the character string table data starting with [S] is stored in [D].	"P.14-124"
String Search	–	ESSRC	(P)	S1, S2, D	Searches for data [S1] from the character string table data starting with [S2], and the relative position of the first match is stored in [D].	"P.14-126"
Takeout of the Right Side of a String	–	ERIGHT	(P)	S1, S2, D	Retrieves the number of characters [S2] from the right side (the side where the device address is larger) of the character string table starting with [S1], and stores it in [D].	"P.14-130"
Takeout of the Left Side of a String	–	ELEFT	(P)	S1, S2, D	Retrieves the character strings of [S2] number of characters from the left side (the side where the device address is smaller) of the character string table starting with [S1], and stores it in [D].	"P.14-133"
Data Read from a Given Position in the String	–	EMIDR	(P)	S1, S2, S3, D	Reads the character strings for [S3] number of character strings from the [S2] byte of the character string table starting with [S1], and stores it in [D].	"P.14-136"
Rewrite from a Given Position in the String	–	EMIDW	(P)	S1, S2, D, n	Reads the character string data for [S2] number of characters from the character string table starting with [S1], and stores it in [D] from the position of the [n]th byte.	"P.14-139"
Replacement of a Character String	–	ESREP	(P)	S, D, p, n	Replaces the character string table starting with [D] with the character strings of [S1]. The positions and number of characters that can be replaced are specified with [p] and [n].	"P.14-142"

■ Communication Instructions

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Specify communication unit slot port	–	UNITSEL	(P)	S1, S2	Specifies the target of each instruction to execute, described immediately before the communication instruction. The target slots are specified in [S1], and the target COM port numbers or target connections are specified in [S2].	"P.15-2"
General-Purpose Communication Send Instruction	US, SS	GPTRNS	(P)	S, n, D	Transfers the [n] bytes of data stored in the area starting with [S] to external devices via the COM port (SCU) or the LAN port (ET-LAN) of the unit.	"P.15-4"
General-Purpose Communicatio	US, SS	pGPSEND		S, n, D	Transfers to external devices using the built-in SCU and built-in ET-LAN of CPU, and the add-on communication port of the	"P.15-4"

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
n Send Instruction					SCU unit (SCU: COM port, ET-LAN: connection).	
General-Purpose Communication Send Instruction	US, SS	GPSEND	(P)	S, n, D	Transfers the [n] bytes of data stored in the area starting with [S] to external devices via the COM port (SCU) or the LAN port (ET-LAN) of the unit.	"P.15-4" (Note 1)
General-Purpose Communication Receive Instruction	US, SS	GPRECV	(P)	D1, D2	Copies the data received from external devices via the COM port (SCU) or the LAN port (ET-LAN) of the unit to the range of [D1] to [D2].	"P.15-15"
MEWTOCOL/ MODBUS Master Send Instruction	US, SS	SEND	(P)	S, n, D1, D2, D3	Reads the data from devices within the master unit starting with [S], and stores the data in the partner station number [D1] from the address starting with [D2]. The number of sent data is specified by [n].	"P.15-21"
MEWTOCOL/ MODBUS Master Communication Receive Instruction	US, SS	RECV	(P)	S1, S2, n D1, D2	Reads the data from the partner station number [S1] from the address starting with [S2], and stores the data in the area of the master unit starting with [D1]. The number of received data is specified by [n].	"P.15-29"
MODBUS Master Send Instruction: Function Code Specification	US, SS	SEND	(P)	S, n, D1 D2, D3	Reads the data from devices within the master unit starting with [S], and stores the data in the partner station number [D1] from the address starting with [D2]. The MODBUS function code and station No. are specified by [D1], and the number of sent data is specified by [n].	"P.15-37"
MODBUS Master Receive Instruction: Function Code Specification	US, SS	RECV	(P)	S1, S2, n D1, D2	Reads the data from the partner station number [S1] from the address starting with [S2], and stores the data in the area of the master unit starting with [D1]. The MODBUS function code and station number are specified by [S1], and the number of received data is specified by [n].	"P.15-44"
Change of SCU Parameter	–	PMSET		S, n, D	The data stored in the area of [n] words starting with [S] is set for the communication parameter of the COM port (SCU) of the unit.	"P.15-51" (Note 2)
		pPMSET		S, n, D	For PMSET instruction, the acknowledgment process is performed when the instruction is executed. For pPMSET instruction, the acknowledgment process is performed when the scan is finished.	"P.15-51"
Obtainment of SCU Parameters	–	PMGET	(P)	S, D	Reads the communication parameter set for the COM port (SCU) of the unit, and stores it in the area starting with [D].	"P.15-58"
Configuration Change	–	CONFIG	(P)	-	Changes the settings for MEWTOCOL communication.	"P.15-65"

(Note 1) For GPSEND instruction, it is necessary to set the execution condition to ON from the time when the instruction is started to the time when the sending flag turns OFF.

(Note 2) For PMSET instruction, it is necessary to set the execution condition to ON from the time when the instruction is started to the time when the processing flag turns OFF.

■ Communication Instruction (When Using FP7 Multi-Wire Link Unit)

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Send Instruction (When Using FP7 Multi-Wire Link Unit)	US, SS	SEND	(P)	S, n, D1, D2, D3	Data can be transferred from the master unit to the partner unit between PLCs connected by MEWNET-W or MEWNET-W2. Reads the data from devices within the master unit starting with [S], and stores the data in the partner station number [D1] from the address starting with [D2]. The number of sent data is specified by [n].	"P.16-2"
Receive Instruction (When Using FP7 Multi-Wire Link Unit)	US, SS	RECV	(P)	S1, S2, n, D1, D2	Data can be transferred from the partner unit to the master unit between PLCs connected by MEWNET-W or MEWNET-W2. Reads the data from the partner station number [S1] from the address starting with [S2], and stores it in the device of the master unit starting with [D1]. The number of received data is specified by [n].	"P.16-6"
Acquiring MEWNET-W Parameters	–	PMGET	(P)	S, D	Reads the communication status set for MEWNET-W and the operations of the PLC link, and stores them in the area starting with [D].	"P.16-10"
Obtainment of MEWNET-W2 Parameters	–	PMGET	(P)	S, D	Reads the communication status set for MEWNET-W2 and the operations of the PLC link, and stores them in the area starting with [D].	"P.16-14"
Obtainment of MEWNET-F Parameters	–	PMGET	(P)	S, D	Reads the communication status set for MEWNET-F, and stores it in the area starting with [D].	"P.16-18"
Change of MEWNET-W Parameter	–	PMSET		S, n, D	The data stored in the area of [n] words starting with [S] is set for the communication parameter of MEWNET-W. For PMSET instruction, the acknowledgment process is performed when the instruction is executed. For pPMSET instruction, the acknowledgment process is performed when the scan is finished.	"P.16-20"
		pPMSET		S, n, D		
Change of MEWNET-W2 Parameter	–	PMSET		S, n, D	The data stored in the area of [n] words starting with [S] is set for the communication parameter of MEWNET-W2. For PMSET instruction, the acknowledgment process is performed when the instruction is executed. For pPMSET instruction, the acknowledgment	"P.16-23"
		pPMSET		S, n, D		

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
					process is performed when the scan is finished.	

■ Special Instruction (When Using FP7 Multi-Wire Link Unit)

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Error Clear (When Using FP7 Multi-Wire Link Unit)	US	ERR	(P)	N	Clears internal errors for the FP7 multi-wire link unit. Resets the values for the system relay and system data register.	"P.16-27"

■ Ethernet Communication Instruction (Setting)

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
ET-LAN Status Read	–	RDET	(P)	D	Reads the status for all connections to the LAN port (ET-LAN) of the unit, and stores it in [D].	"P.17-3" (Note 1)
Information Acquisition of Ethernet Unit (IP/MAC Destination)	–	ETSTAT	(P)	S1, S2, D	Stores the unit information (IP/MAC connection destination) specified by [S1] and [S2] in the area starting with [D].	"P.17-5" (Note 1)
Information Acquisition of Ethernet Unit (FTP/HTTP/ SMTP)	–	ETSTAT	(P)	S1, S2, D	Stores the unit information (FTP/HTTP/ SMTP) specified by [S1] and [S2] in the area starting with [D].	"P.17-15" (Note 1)
IP Address Setting	–	IPv4SET	(P)	S	Sets the IP address with the setting parameters stored in the device address starting with [S] or the number of character constants. The subnet mask and default gateway can be omitted.	"P.17-23" (Note 1)
PING request	–	pPINGREQ		S, D	In order to acknowledge the operations of communication relay devices, a PING send request is performed [S] number of times with the devices indicated in [D].	"P.17-28" (Note 1)
User Connection Setting	–	CONSET	(P)	S1, S2, D1, D2	Sets the connection setting parameters specified by [S1] and [S2] to the range of [D1] to [D2].	"P.17-32" (Note 1)
Connection open	–	OPEN	(P)	S	Opens the connection number specified by [S].	"P.17-39" (Note 1)
Connection close	–	CLOSE	(P)	S	Closes the connection number specified by [S].	"P.17-41" (Note 1)
NTP destination server settings	–	NTPcSV	(P)	S1, S2	Sets the SNTP server address and time zone to the CPU unit with a built-in ET-LAN.	"P.17-43" (Note 1)

Name	Operation Unit	Mnemonic	Operand	Function overview	Page
Time adjustment request	–	pNTPcREQ	S1, S2, D	Requests the time adjustment.	"P.17-47" (Note 1)

(Note 1) Can be used with CPU units CPS4RE*/CPS3RE*.

■ Ethernet Communication Instruction (FTP Client)

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
FTP client connected server setting	–	FTPcSV	(P)	S1, S2, S3	Sets the FTP server of the connection destination to the CPU unit according to the parameters specified by [S1], [S2], and [S3].	"P.17-52" (Note 1)
FTP client transfer setting	–	FTPcSET	(P)	S1, S2, S3, S4	Sets and stores the FTP client transfer settings of [S2] to [S4] to the transfer setting area specified by [S1].	"P.17-59" (Note 1)
FTP client logging/trace transfer setting	–	FTPcLOG	(P)	S1, S2, S3	Sets and stores the logging trace transfer settings of [S2] to [S3] to the logging trace transfer settings area specified by [S1].	"P.17-69" (Note 1)
FTP client transfer request	–	FTPcREQ	(P)	S	Requests a transfer of the FTP client transfer No. specified by [S].	"P.17-73" (Note 1)
FTP client transfer control	–	FTPcCTL	(P)	S1, S2	Controls whether to enable, disable, or cancel the transfer of the FTP client, according to the specification of the control description [S2] for control target [S1].	"P.17-77" (Note 1)

(Note 1) Can be used with CPU units CPS4RE*/CPS3RE*.

■ Ethernet Communication Instruction (HTTP Client)

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
HTTP client connected server setting	–	HTTPcSV	(P)	S1, S2, S3	Sets the HTTP server of the connection destination to the CPU unit according to the parameters specified by [S1], [S2], and [S3].	"P.17-81" (Note 1)
HTTP client transfer setting	–	HTTPcSET	(P)	S1, S2, S3, S4	Sets and stores the HTTP client transfer settings of [S2] to [S4] to the transfer setting area specified by [S1].	"P.17-87" (Note 1)
HTTP client transfer request	–	HTTPcREQ	(P)	S	Requests a transfer of the HTTP client transfer No. specified by [S].	"P.17-94" (Note 1)
HTTP client transfer control	–	HTTPcCTL	(P)	S1, S2	Controls whether to enable, disable, or cancel the transfer of the HTTP client, according to the specification of the control description [S2] for control target [S1].	"P.17-98" (Note 1)

(Note 1) Can be used with CPU units CPS4RE*/CPS3RE*.

■ Ethernet Communication (SMTP Mail Sending) Instruction

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Mail Text Setting	–	SMTPcBDY	(P)	S1, S2	Sets the text specified by [S2] into the mail text of the number specified by [S1].	"P.17-102" (Note 1)
Mail Text Read	–	SMTPcBRD	(P)	S, D	Reads the mail text of the number specified by [S] to the area specified by [D].	"P.17-105" (Note 1)
Mail Server Setting	–	SMTPcSV	(P)	S1, S2, S3	Sets the mail transmission server and the sender to the CPU unit according to the parameters specified by [S1], [S2], and [S3].	"P.17-108" (Note 1)
Destination Group Setting	–	SMTPcADD	(P)	S1, S2, S3, S4	Sets the destination group name specified by [S2] and the destination address specified by [S3] and [S4] to the destination group number specified by [S1].	"P.17-115" (Note 1)
Mail Transmission Setting	–	SMTPcSET	(P)	S1, S2, S3, S4	Stores the mail transmission settings of [S1] to [S4] in the mail transmission setting area.	"P.17-120" (Note 1)
Logging/Trace mail setting	–	SMTPcLOG	(P)	S1, S2, S3, S4	Stores the logging/trace mail settings of [S2] to [S4] in the logging/trace transfer setting area specified by [S1].	"P.17-130" (Note 1)
Mail Send Request	–	SMTPcREQ	(P)	S	Requests transmission of the mail with the transmission No. specified by [S].	"P.17-135" (Note 1)
Mail Transmission Control	–	SMTPcCTL	(P)	S1, S2	Controls whether to enable, disable, or cancel the transmission of mail, according to the specification of the control description [S2] for control target [S1].	"P.17-139" (Note 1)

(Note 1) Can be used with CPU units CPS4RE*/CPS3RE*.

■ Ethernet Communication (Ethernet/IP) Instruction

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Information acquisition of EtherNet/IP	–	ETSTAT	(P)	S1, S2, D	Reads the parameter information or status information specified by [S1] and [S2], and stores it in the area starting with [D].	"P.17-142" (Note 1)
EtherNet/IP node status acquisition instruction	–	EIPNDST		S, D1, D2	Stores the status for the node number specified by [S] in the device specified by [D1], and stores the execution result in [D2].	"P.17-148" (Note 1)
Cyclic communication start request	–	EIPSTART		S, n, D	Starts the node on which the start request is made within the maximum node number specified by [n] from the stop request node number table specified by [S].	"P.17-180" (Note 1)
Cyclic communication stop request	–	EIPSTOP		S, n, D	Stops the node on which the stop request is made within the maximum node number specified by [n] from the stop request node number table specified by [S].	"P.17-183" (Note 1)

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
EtherNet/IP input refresh	–	EIP_IN	(P)	S1, S2, D	Refreshes the input for the connections targeted to be refreshed.	"P.17-186" (Note 1)
EtherNet/IP output refresh	–	EIP_OT	(P)	S1, S2, D	Refreshes the output for the connections targeted to be refreshed.	"P.17-191" (Note 1)

(Note 1) Can be used with CPU units CPS4RE*/CPS3RE*.

■ Ethernet Communication (MC Protocol) Instruction

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
MC Protocol Master Transfer Instruction	US, SS	SEND	(P)	S, n, D1 D2, D3	Reads the data from the devices within the master unit starting with [S], and stores the data in the partner station from the address starting with [D1] + [D2]. The device type of the transfer destination device is specified by the high byte of [D1] and the number of sent data is specified by [n].	"P.17-195" (Note 1)
MC Protocol Master Communication Receive Instruction	US, SS	RECV	(P)	S1, S2, n D1, D2	Reads the data of the partner station from the address starting with [S1] + [S2], and stores it in the area of the master unit starting with [D1]. The device type of the transfer source device is specified by the high bytes of [S1] and the number of received data is specified by [n].	"P.17-201" (Note 1)

(Note 1) Can be used with CPU units CPS4RE*/CPS3RE*.

■ SD Memory Card Access Instruction

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Write file to operation memory in BIN format	US, SS	CDTWT	(P)	S, n, D	Writes the binary data for [n] words from the area starting with [S] to the SD memory card as a file.	"P.18-2"
Read from BIN format file to operation memory	US, SS	CDTRD	(P)	S, n, D	Reads [n] pieces of binary data from the file with the file number specified by [S], and stores it in the device address starting with [D].	"P.18-4"
File Data Write Instruction	–	CWT	(P)	S, n, D1, D2	Stores [n] pieces of data stored in the device address starting from [S] to the file specified by [D1] according to the parameter specified by [D2].	"P.18-6"
File data read	–	CRD	(P)	S1, S2, n D	Stores [n] pieces of data to the device address starting from [D] according to the file specified by [S1] and the parameter specified by [S2].	"P.18-19"
Directory creation	–	CMKDIR	(P)	S	Creates a folder in an SD memory card.	"P.18-30"

Name	Operation Unit	Mnemonic		Operand	Function overview	Page
Directory deletion	–	CRMDIR	(P)	S	Deletes a folder in an SD memory card.	"P.18-33"
Directory Delete (valid for directories containing files)	–	CRMDIR FL	(P)	S	Deletes a folder in an SD memory card.	"P.18-33"
File deletion	–	CFDEL	(P)	S	Deletes the file specified by [S] in an SD memory card.	"P.18-36"
ASCII Data Write to File	STR	CPR	(P)	S, D	Writes the character string specified by [S] at the end of the file specified by [D].	"P.18-38"
One line read from file	STR	CRD1	(P)	S, D1, D2	Reads the first row data from the row specified by [D2] in the file specified by [S], and stores it in the device address starting from [D1].	"P.18-42"
File rename	–	CREN	(P)	S1, S2	Changes the file name specified by [S1] to the file name specified by [S2].	"P.18-48"
File copy	–	CCOPY	(P)	S1, S2, S3	Copies the file specified by [S1] to the file specified by [S2] according to the parameter specified by [S3].	"P.18-50"
File move	–	CMV	(P)	S1, S2, S3	Moves the file specified by [S1] to the file specified by [S2] according to the parameter specified by [S3].	"P.18-53"
Acquiring SD Memory Card Free Space	UL	CFREE	(P)	D	Stores the free space amount of the SD memory card in the area specified by [D] in byte units.	"P.18-56"
Acquiring SD Memory Card Free Space	UL	CFREEK	(P)	D	Stores the free space amount of the SD memory card in the area specified by [D] in k (kilo) byte units.	"P.18-58"
File status acquisition	US, SS	CFLS	(P)	S, D	Obtains the status of the file name specified by [S], and stores the result in 10-word area ([D] to [D]+9) from [D].	"P.18-60"
Panasonic SD Memory Card Lifetime Information Read	–	PanaSD	(P)	D1, D2, D3	Reads the lifetime information of a Panasonic SD memory card.	"P.18-63"

(Note 1) SD card control instructions cannot be used for CPU unit CPS2R.

(MEMO)

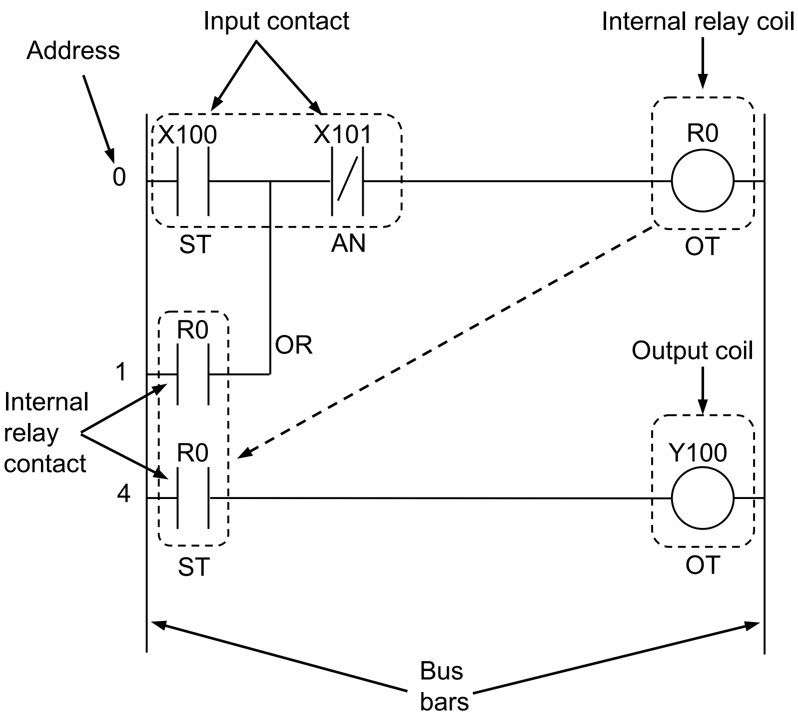
2 Overview of Instructions

2.1 Structure of Instructions

2.1.1 Structural patterns of basic instructions

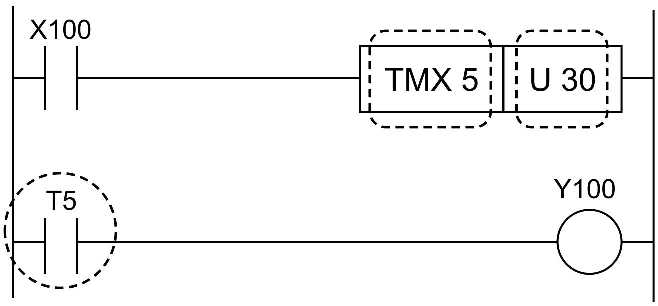
■ Sequence Basic Instructions

- Basic instructions are a group of the most essential instructions based on the relay sequence circuit, used for logic operation by the unit of bits. As indicated below, they are expressed by a combination of relay coils and contacts.
- Available relays vary by instruction. Refer to explanations for respective instructions.



■ Basic function instructions

- There are a group of instructions that execute timer, counter, shift register and other basic functions.



In this example, a 0.1-second timer with the timer number 5 is set to 3.0 seconds. It starts counting time when X100 is ON, and the contact T5 turns ON when the timer reaches 3.0 seconds.

■ Control Instructions

- These are a group of instructions that determine the order and flow for executing a program. Conditions can be set to modify parts of a program to be executed, or to execute necessary parts only.
- Major control instructions include the following.

Mnemonic	Name	Function
MC-MCE	Master control relay	When execution conditions are OFF, output between MC and MCE are turned OFF.
JP/LBL	Jump label	Jumps to the LBL instruction with the same number when the execution condition is ON. When LBL instruction is in an address that comes after JP instruction, the programs between JP and LBL are not executed, so the operation time is shortened.
LOOP/LBL	Loop label	When execution conditions are ON, the part of a program between LBL and LOOP is repeated for specified times.

■ Step ladder instructions

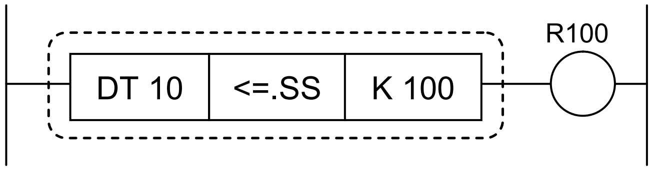
- These instructions handle a section specified by SSTP to STPE as an independent "process", and operate sequential execution or branch execution.

■ Subroutine instructions

- These instructions call and execute subroutines when necessary. Subroutines are programs for repeated execution of operations, etc., as specified by SBL and RET.

■ Data comparison instructions

These are a group of instructions for comparing two sets of data. They operate as a contact that turns ON and OFF in accordance with the comparison result.

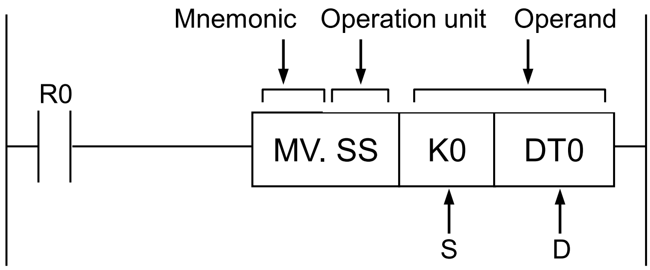


2.1.2 Structural patterns of high-level instructions

■ Structure of high-level instructions

- Mnemonics indicate the specifics of operation (e.g. data transfer, arithmetic operation).
- Operation units indicate the units for operation triggered by instructions (e.g. US, SS, UL, SL, SF, DF).

- Operands indicate the targets of operation and/or methods for operation processes. There are three types of operands: [S], [D], and [n]. The number and types of operands that need to be specified vary by instruction.



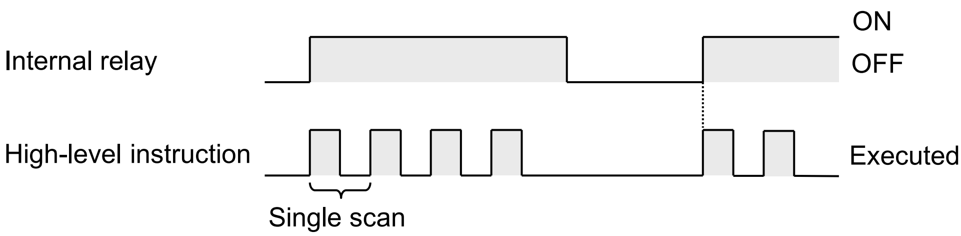
■ Types of operands

Symbol	Abbreviation		Function
S	Source	Source	Specify data targeted by operation and/or methods for processing.
D	Destination	Destination	Specify the storage location for operation result.
n	Number	Number	Numerical data that specify data targeted by operation and/or methods for processing.

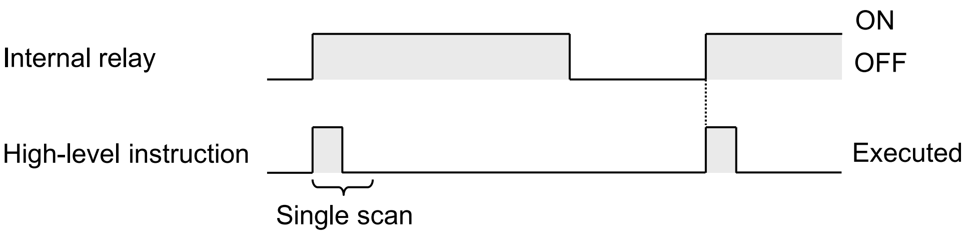
■ Every Scan Execution Type and Differential Execution Type

- For high-level instructions, there are two execution types: every scan execution type and differential execution type.

(1) Every Scan Execution Type
This type of instruction is executed repeatedly for every scan while the internal relay is established.

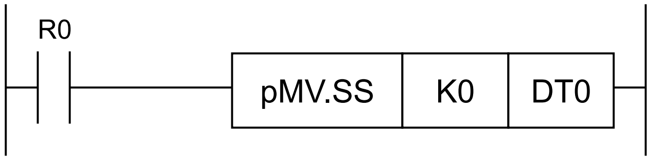


(2) Differential Execution Type
This type of instruction is executed only for a single scan when a leading edge or trailing edge of the internal relay is detected.



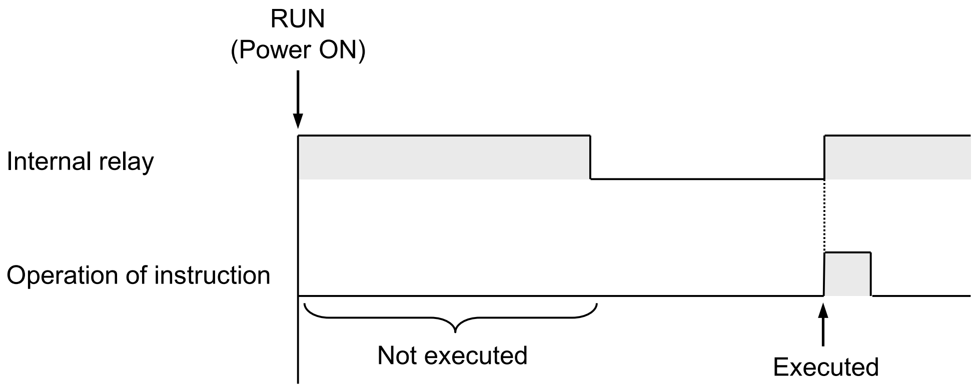
■ **Input of differential execution type instruction**

- To input a differential execution type instruction using the tool software FPWINGR, press the [Shift] key and [F6] key, and then select an instruction from the instruction list dialog box.
- (P) comes at the head of differential execution type instructions.



■ **How differential execution type instructions work**

- While the internal relay that is set as the execution condition for differential execution type instructions is turned ON, instructions are only executed at the leading edge of the signal, and after that instructions are not executed.
- Instructions are not executed for the first scan if the internal relay for differential execution type instruction is established from the beginning when the mode is switched to RUN mode or when the power supply is turned on in RUN mode.



- When differential execution instruction (P instruction) is used with instructions that change the order of execution of instructions such as MC-MCE or JP-LBL ((1) to (6), shown below), the action executed by the instruction may change depending on the timing of the execution of the instruction or the timing of the count input. Take care regarding this point.

  (1) MC to MCE instructions

  (2) JP to LBL instructions

(3) LOOP to LBL instructions
(4) CNDE instruction
(5) Step ladder instructions
(6) Subroutine instructions

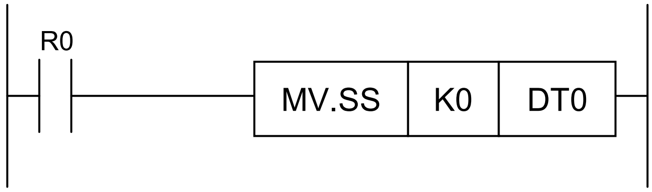
Info.

- For details, refer to "19.7 Rise Detection Method".
- Please take care that the program is not described incorrectly when combining the differential execution type high-level instruction with the AND stack instruction or pop stack instruction. For details, refer to "19.8 Precautions for Programming".

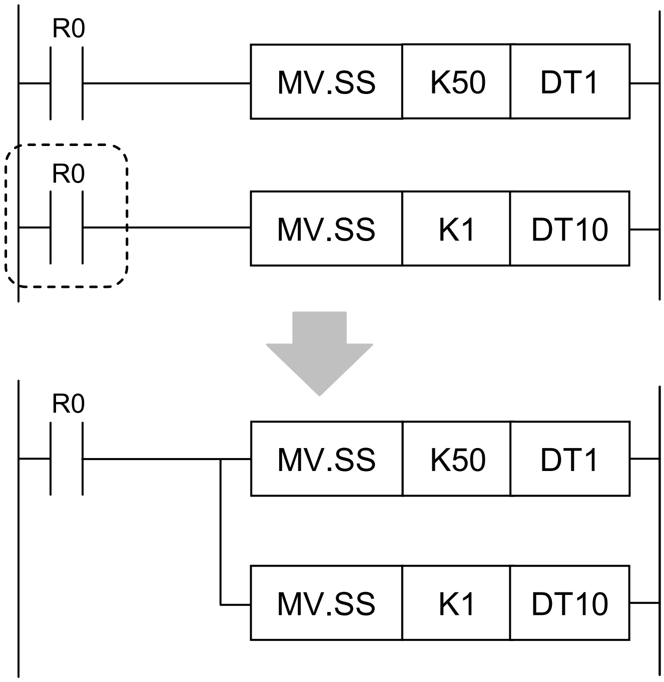
■ High-level instructions and internal relay

- High-level instructions are always used as a pair with internal relay.
- When the operation result of the relay sequence circuit specified for an internal relay is ON, the high-level instruction is executed.

Example) When the internal relay R0 is ON, the MV instruction is executed, and K0 is transferred to DT0.



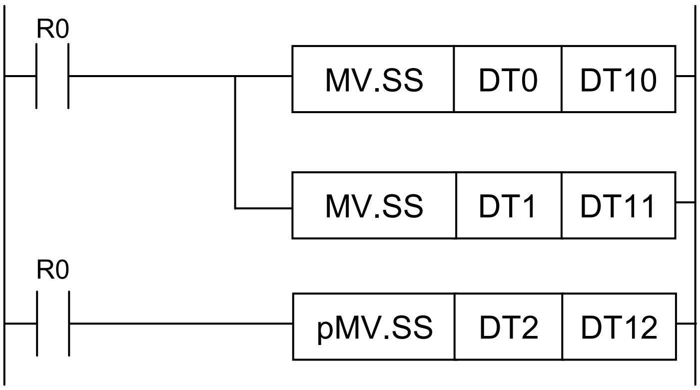
- When using high-level instructions repeatedly and the internal relay is the same, from the second time the internal relay can be omitted.



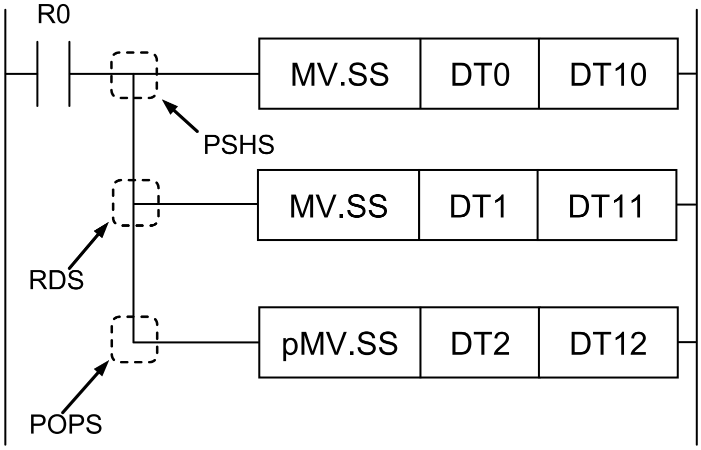
■ Cautions on omitting the internal relay

- If every scan execution type instruction and differential execution type instruction are being mixed for one internal relay, perform the programming in the following manner.

Example 1) Define every scan execution type and differential execution type separately.



Example 2) Use PSHS, RDS, or POPS instruction



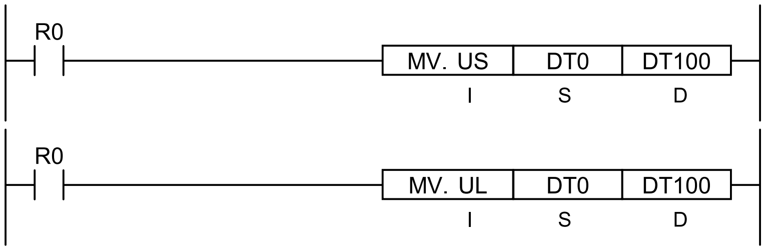
2.2 Operation unit

■ What are operation units?

- They specify basic units for operation triggered by respective instructions.
- Operation result of the same instruction may vary by the specified operation unit. The range of operand targeted by operation and/or the number of words to store the operation result also vary.
- Available operation units vary by instruction. In some instructions, operation units are not specified. Refer to explanations for respective instructions. Operation units are indicated with "i".

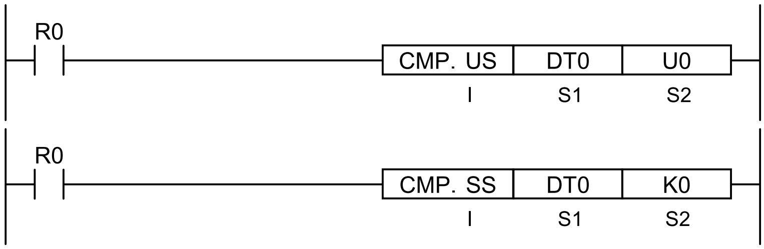
Example 1) When the operation units in data transfer instructions vary

- When the operation unit is US (unsigned 16-bit data), the value of DT0 is transferred to DT100.
- When the operation unit is UL (unsigned 32-bit data), the values of DT0 to DT1 are transferred to DT100 to DT101.



Example 2) When the operation units in comparison instructions vary
When binary data "1111 1111 1111 1111" is included in DT0 (S1)

- If the operation unit is US (unsigned 16-bit data), the value of (S1) is handled as a positive number "65535", resulting in (S1) > (S2), and the comparison flag SRA turns ON.
- If the operation unit is SS (signed 16-bit data), the value of (S1) is handled as a negative number "-1", resulting in (S1) < (S2), and the comparison flag SRC turns ON.



■ Types of operation units

Symbol	Name	Available range
bit	1-bit data	0,1
US	Unsigned 16-bit data	0 to 65535
SS	Signed 16-bit data	-32768 to +32767
UL	Unsigned 32-bit data	0 to 4294967295

Symbol	Name	Available range
SL	Signed 32-bit data	-2147483548 to +2147483547
SF	Single-precision floating point real number data	-1.175494E-38 to -3.402823E+38 0 +1.175494E-38 to +3.402823E+38
DF	Double-precision floating point real number data	-2.2250738585072014E-308 to -1.7976931348623158E+308 0 +2.2250738585072014E-308 to +1.7976931348623158E+308

■ Operation units and available constants (●: Available)

Operation unit			K constant		U constant		H constant		Real number		String
Name	Size	Sign	16bit	32bit	16bit	32bit	16bit	32bit	SF	DF	""
US	16bit	No	-	-	●	-	●	-	-	-	-
SS		Yes	●	-	-	-	●	-	-	-	-
UL	32bit	No	-	-	-	●	-	●	-	-	-
SL		Yes	-	●	-	-	-	●	-	-	-
SF	32bit	Yes	-	-	-	-	-	-	●	-	-
DF	64bit	Yes	-	-	-	-	-	-	-	●	-

2.3 Operation device list

■ Operation device list

No. of bits	Symbol		Name	Points	Range:	Description
	Global	Local				
1	X	_X	External input	8192 points	0 to 511F	Turns ON or OFF based on external input.
	Y	_Y	External output	8192 points	0 to 511F	Externally outputs ON or OFF state.
	R	_R	Internal relay	32768 points	0 to 2047F	Relay that turns ON or OFF only within program
	L	_L	Link relay	16384 points	0 to 1023F	Shared relay used for a PLC link
	T	_T	Timer	4096 points	0 to 4095	Turns ON when the timer reaches the specified time.
	C	_C	Counter	1024 points	0 to 1023	Turns ON when the counter reaches the set value.
	SR	-	System relay	Approx. 1120 points		Relay which turns ON or OFF based on specific conditions and is used as a flag.
	P	-	Pulse relay	4096 points	0 to 255F	Relay which turns ON at the leading edge only for the first scan when the execution condition is on.
	E	-	Error alarm relay	4096 points	0 to 4095	Relay that stores error conditions which are freely allocated by the user in the memory.
	IN	-	Direct input	Maximum 1008 points	0 to 62F	Relay for input/output processing during operation, independent of normal I/O refresh.
	OT	-	Direct output	Maximum 1008 points	0 to 62F	
	DT*.n	_DT*.n	Bit specification of data register	Maximum 1M words (.n: 16 points)	0 to 999423 .n: 0 to F	Device for specifying a specific bit of the 16-bit device DT or LD. This can only be used with basic instruction and communication instruction.
	LD*.n	_LD*.n	Bit specification of link register	16384 words (.n: 16 points)	0 to 16383 .n: 0 to F	
16	WX	_WX	External input	512 words	0 to 511	Code for specifying 16 external input points as one word (16 bits) of data.
	WY	_WY	External output	512 words	0 to 511	Code for specifying 16 external output points as one word (16 bits) of data.
	WR	_WR	Internal relay	2048 words	0 to 2047	Code for specifying 16 internal relay points as one word (16 bits) of data.
	WL	_WL	Link relay	1024 words	0 to 1023	Code for specifying 16 link relay points as one word (16 bits) of data.

No. of bits	Symbol		Name	Points	Range:	Description
	Global	Local				
	DT	_DT	Data register	Maximum 1M words	0 to 999423	Data memory used in program
	LD	_LD	Link register	16384 words	0 to 16383	Shared data memory used for a PLC link
	UM	-	Unit memory	Maximum 512 kw/unit	0 to 524287	Device for accessing the unit memory of an intelligent unit. Its size varies by unit, and is allocated by default.
	SD	-	System data	Approx. 80 words	-	Data memory for storing specific data. Various settings and error codes are stored.
	WI	-	Direct input	Maximum 63 words	0 to 62	Device for input/output processing during operation in word units, independent of normal I/O refresh.
	WO	-	Direct output	Maximum 63 words	0 to 62	
32	TS	_TS	Timer set value	4096 double words	0 to 4095	Data memory for storing the target value of a timer. It corresponds to the timer number.
	TE	_TE	Timer elapsed value	4096 double words	0 to 4095	Data memory for storing the elapsed value of a timer. It corresponds to the timer number.
	CS	_CS	Counter set value	1024 double words	0 to 1023	Data memory for storing the target value of a counter. It corresponds to the counter number
	CE	_CE	Counter elapsed value	1024 double words	0 to 1023	Data memory for storing the elapsed value of a counter. It corresponds to the counter number
	I0 to IE	-	Index modification register	15 double words	I0 to IE	Register used for modifying memory area addresses and/or constants.

(Note 1) Figures in the table indicate the number of devices that can be used in a program. The actual number of input/output points that can be used varies depending on the configuration.

(Note 2) Operation devices are categorized into "hold type", which memorizes the status immediately before power failure or switch to the PROG. mode, and "non-hold type", which resets the status. Non-hold area is cleared to zero when the unit is powered on or the mode is switched between PROG and RUN.

Types of operation devices	Setting of hold or non-hold type
Internal relay (R), Data register (DT), Link relay (L), Link register (LD)	Can be specified as a hold or non-hold type, using the tool software.
Counter (C), Counter set value (CS), Counter set value (CE), Error alarm relay (E)	Hold type
Inputs (X), Outputs (Y), Timers (T), Timer set values (TS), Timer elapsed values (TE), Pulse relays (P), Direct inputs (IN), Direct outputs (OT), Index registers (I), Unit memories (UM), System data registers (SD)	Non-hold type

(Note 3) Direct input (IN), direct output (OT), and unit memory (UM) are used by specifying unit slot numbers and memory addresses to be controlled by instructions.

(Note 4) The number of usable data registers (DT) varies according to the type of CPU unit and memory configuration settings.

However, data registers (DT) that can be used as hold type are a maximum of 262,144 words (DT0 to DT262143) for CPS4R*/CPS3R*. For CPS2R, a maximum of 131,072 words (DT0 to DT131071) can be used.

Unit type	Memory type	Memory selection patterns				
		1	2	3	4	5
CPS4R*	Data register capacity (words)	65536	131072	262144	524288	999424
CPS3R*	Data register capacity (words)	131072	262144	425985	589824	
CPS2R	Data register capacity (words)	131072	262144			

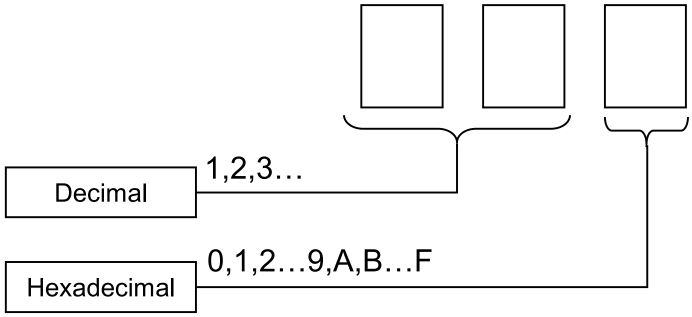
■ List of constants

Symbol	Name	No. of operation bits	Range:
K	Signed decimal constant	16	-32768 to +32767
		32	-2147483548 to +2147483547
U	Unsigned decimal constant	16	0 to 65535
		32	0 to 4294967295
H	Hexadecimal constant	16	0 to FFFF
		32	0 to FFFF FFFF
SF	Single-precision floating point real number constant	32	-1.175494E-38 to -3.402823E+38 0 +1.175494E-38 to +3.402823E+38
DF	Double-precision floating point real number constant	64	-2.2250738585072014E-308 to -1.7976931348623158E+308 0 +2.2250738585072014E-308 to +1.7976931348623158E+308
""	Character constant	8	Up to 256 characters

2.4 Specification of Device Numbers

■ For external input (X), external output (Y), internal relay (R), link relay (L), pulse relay (P), system relay (SR), direct input (IN), direct output (OT)

Since relays may be handled in units of 16 points, their numbers should be expressed as a combination of decimal and hexadecimal numbers.



<Example> External input X

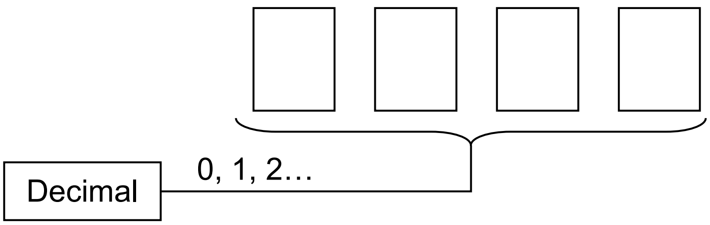
X 0, X 1, ························X F
X10, X11, ························X1F
X20, X21, ························X2F
⋮ ⋮ ⋮

■ Relay number of external input/output

- Only what is actually allocated to the input contact can be used for external input (X).
- Only what is actually allocated to the output contact can be output for external output (Y). External outputs (Y) that are not allocated can be used as internal relays.
- Allocation of numbers is decided according to the combination of units.

■ Timer contact (T), counter contact (C), and error alarm relay (E)

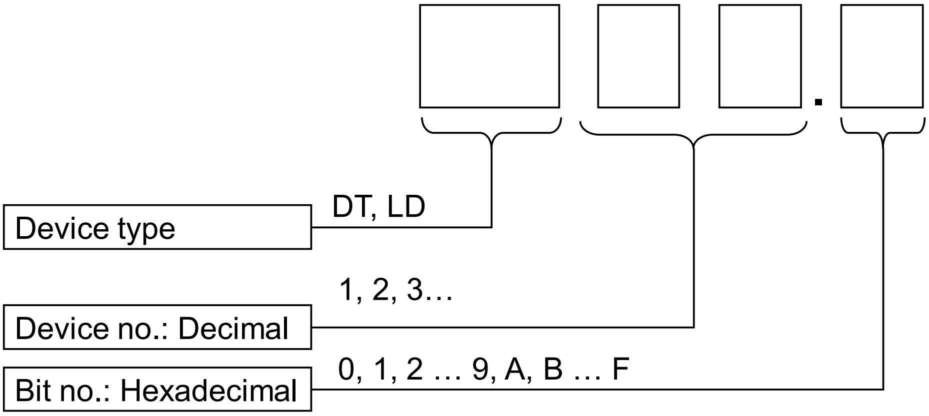
- The numbers of timer/counter contacts correspond to the numbers of timers/counters, and should be specified by decimal numbers.
- The numbers of error alarm relays should be specified by decimal numbers.



T 0,T 1 ························T4095
C 0,C 1 ························C1023
E0, E1 ························ E4095

■ Bit specification of word devices (DT*.n, LD*.n)

- As for data register DT and link register LD, specific bits in 16-bit data can be extracted and used as bit data, by specifying the device type, device number, and bit number.
- The device number and the bit number should be separated by a period (.).
- This can only be used with basic instruction and communication instruction. Refer to "Available devices" of the respective instructions.



<Example> For word devices DT*.n

DT0.0, DT0.1, ------- DT0.F,

DT10.0, DT10.1, ---- DT10.F

DT20.0, DT20.1, ---- DT20.F

⋮

■ How to count memory area numbers

The numbers of respective memory areas (e.g. data register DT) should be specified by decimal numbers. (This excludes index registers.)

<Example> For data register DT

DT0, DT1,-----------DT9

DT10, DT11, -------DT19

⋮

2.5 Explanations about Relays

2.5.1 X External input

■ How external input (X) works

- External input is used to feed signals to the programmable controller from an external input device such as a limit switch or a sensor connected to an input point.

■ Usage restrictions

- External inputs that are not allocated cannot be used.
- The ON/OFF state of external inputs cannot be changed by an operation (program).
- There is no limit to the number of times it can be used in a program.

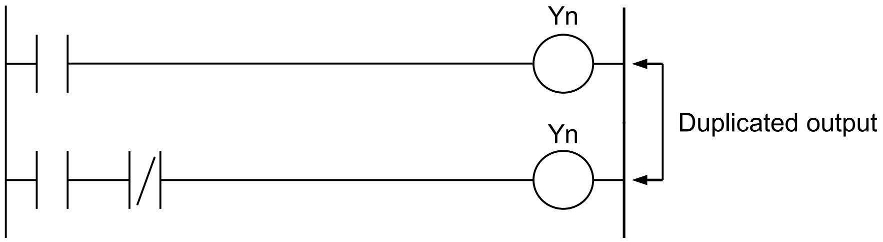
2.5.2 Y External output

■ How external output (Y) works

- External output is used to send the results of the program operations as signals to the loads connected to output points (e.g. relays, solenoids). The ON or OFF state of an external output is sent as a control signal.

■ Usage restrictions

- External outputs that are not allocated can be used as internal relays. (However, they cannot be specified for hold-type devices.)
- When used as contacts, there are no restrictions on the number of times that can be used.
- As a rule, when you specify the relay as the output destination of operation results (using OT and KP instructions), the specification is limited to once in a program. (To prevent duplicate output)
- You can also enable duplicate output using the setting for **Select operation**>**Duplicate output authorization** in the "CPU configuration" dialog box using the tool software FPWIN GR7.
- Use in the SET or RST instruction is not regarded as duplicate output.



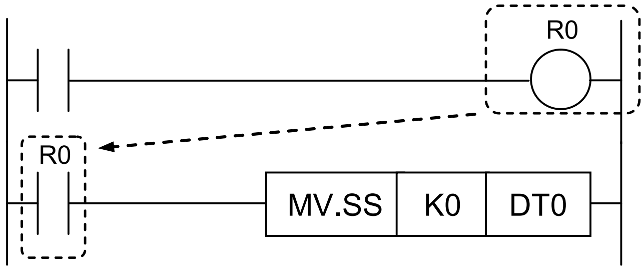
Note

- Even when a project total check is conducted using the tool software FPWIN GR7, the instruction used at the start is not indicated. Instead, the second and later outputs that are regarded as duplicate use are indicated.

2.5.3 R Internal relay

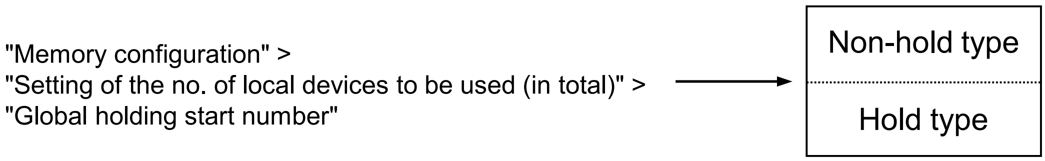
■ How internal relay (R) works

- An internal relay only operates within a program. Its ON or OFF state is not externally output.
- Once the operation result is output and switched ON (Coil: ON), the same relay used as a contact is also switched ON.



■ Hold type relay and non-hold type relay

- Internal relays are categorized into two types, depending on how the relay operates after powering off or switching from RUN to PROG. mode:
  1. Hold type relay, which remembers the ON/OFF status immediately before the relay is stopped and can resume operation using the same status after restarting
  2. Non-hold type relay, which resets the status when the relay is stopped
- You can set the range of hold and non-hold types in **Setting of the no. of local devices to be used (in total)**>**Global hold-type start no.** in in the "Memory Configuration" dialog box using the tool software FPWIN GR7.
- If the beginning of a hold type relay is specified using a word number, relays before that point will be non-hold types, and subsequent relays will be hold types.
- Hold type and non-hold type settings can be set for each global device and local device.



■ Usage restrictions

- When used as contacts, there are no restrictions on the number of times that can be used.
- As a rule, when you specify the relay as the output destination of operation results (using OT and KP instructions), the specification is limited to once in a program. (Double output is prohibited)
- You can also enable duplicate output using the setting for **Duplicate output authorization**>**Select operation** in "CPU configuration". Additionally, use of SET or RST instruction is not regarded as duplicate output.

2.5.4 SR System relay

■ How system relay (SR) works

- A system relay turns ON or OFF under specific conditions.
- Its ON or OFF state is not externally output. It only operates within a program.

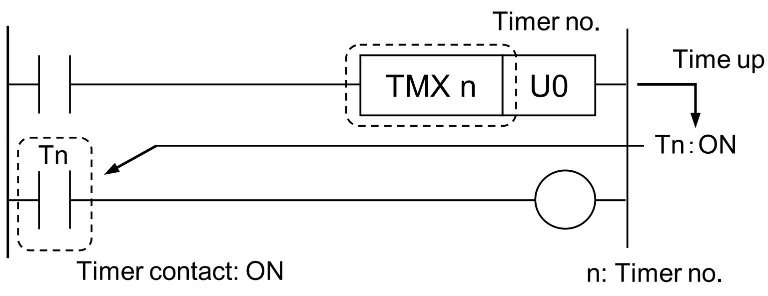
■ Types of system relays (SR)

Classification	Function	Example
Operation status flag	Operation status is indicated by ON or OFF.	Under operation (RUN) (SR20), under forced input/output (SR29), results of comparison instruction (SRA to SRC)
Error flag	Turns ON when an error occurs and indicates abnormalities.	Operation error (SR7, SR8)
Relay that turns ON or OFF under specific conditions	Can be used with the desired conditions in the program.	Normally ON relay (SR10), ON/OFF for each scan (SR12) Initial pulse relay (SR13 to SR14), clock pulse (SR18 to SR1E)

2.5.5 T Timer

■ How timer (T) works

- When a timer is activated and the set time elapses, the timer contact with the same number as the timer turns ON.
- When the timer is in the time-up state and the timer execution condition turns OFF, the timer contact turns OFF. All timers are non-hold type.



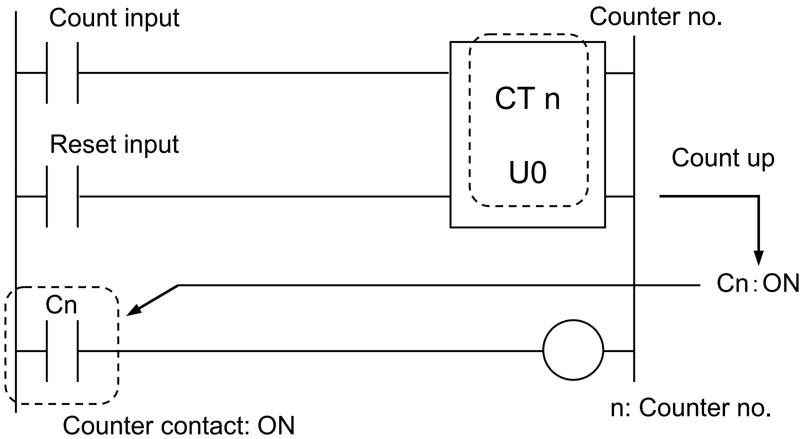
■ Usage restrictions

- When used as contacts, there are no restrictions on the number of times that can be used.

2.5.6 C Counter

■ How counter (C) works

- When the decrement-type preset counter is activated and the counter value reaches zero, the counter contact with the same number as the counter turns ON.
- When the counter's reset input is turned ON, the counter contact turns OFF.
- All counters are hold type.



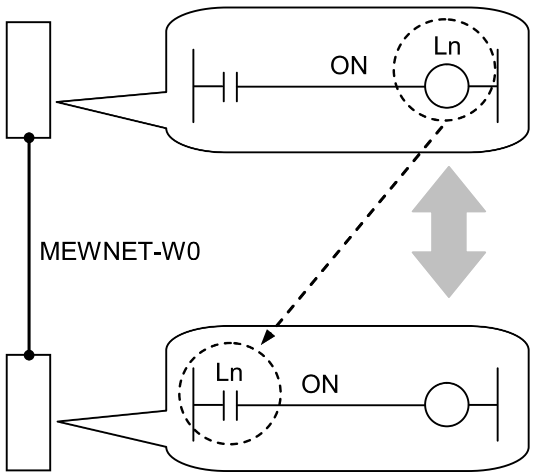
■ Usage restrictions

- When used as contacts, there are no restrictions on the number of times that can be used.

2.5.7 L Link relay

■ How link relay (L) works

- Link relays are relays for when multiple controllers are connected in a network, as a "PLC link" relay for shared use between controllers.
- If the operation result is output to a link relay (coil) on a PLC, it is sent to other PLC connected in the network, and the operation result is reflected in link relays (contacts) with the same number.
- You can send and receive bit information between PLCs in this manner when using link relays.



■ Range used by link relay

- The range of link relays that can be used varies depending on the type of network and unit combinations.
- It is necessary to set the range of use and the number of points for each individual network.

■ Setting of hold type relay and non-hold type relay

- Link relays are categorized into two types, depending on how the relay operates after powering off or switching from RUN to PROG. mode:
  1. Hold type relay, which remembers the ON/OFF status immediately before the relay is stopped, until operation resumes
  2. Non-hold type relay, which resets the status when the relay is stopped
- You can set the range of hold and non-hold types in **Setting of the no. of local devices to be used (in total)**>**Global hold-type start no.** in the "Memory configuration" dialog box using the tool software FPWIN GR7.
- If you specify the beginning of a hold type relay using a word number, relays before that point will be non-hold types, and subsequent relays will be hold types. For example, when the **Setting of the no. of local devices to be used (in total)**>**Global hold-type start no.** in the "Memory configuration" dialog box is set to 10, L0 to L9F will be non-hold type, and L100 to L63F will be hold type.
- Take note that when using a relay as a receiving link relay, even if you set the link relay to hold type in the system configuration, hold operations will not be performed.
- Hold type and non-hold type settings can be set for each global device and local device.

■ Usage restrictions

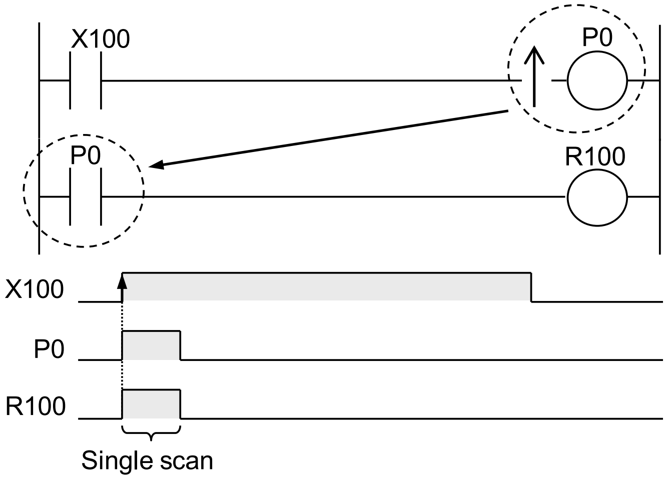
- When used as contacts, there are no restrictions on the number of times that can be used.
- As a rule, when you specify the relay as the output destination of operation results (using OT and KP instructions), the specification is limited to once in a program. (Double output is prohibited)
- You can also enable duplicate output using the setting for **Duplicate output authorization**>**Select operation** in the "CPU configuration" dialog box. Additionally, use of SET or RST instruction is not regarded as duplicate output.

2.5.8 P Pulse relay

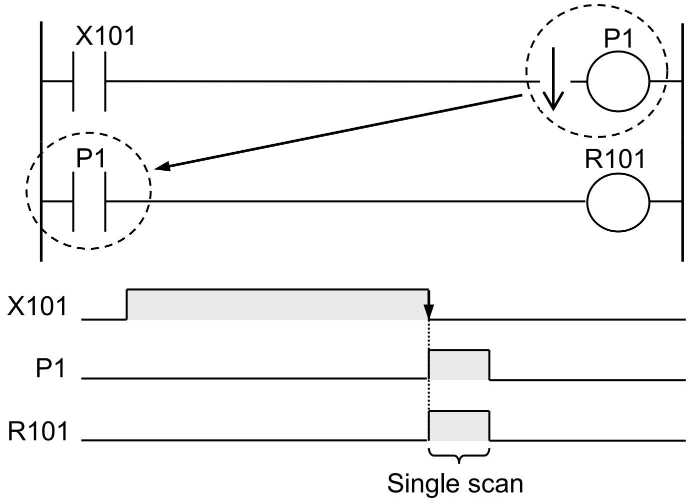
■ How pulse relay (P) works

- Pulse relay is a relay that is only turned ON for one scan. Its ON or OFF state is not externally output. It only operates within a program.
- Turns ON only when Leading Edge Detection Out (OT↑) or Trailing Edge Detection Out (OT↓) instructions are executed.
- When used as the trigger, the pulse relay only operates for one scan when the leading edge or trailing edge is detected.

<Example 1> Differential execution at the leading edge for input X100



<Example 2> Differential execution at the trailing edge for input X101



■ Usage restrictions

- Pulse relays are cleared when the power is turned off.

- As a rule, when you specify the relay as an output destination for an OT↑ or OT↓ instruction, the specification is limited to once in a program. (Double output is prohibited)
- When used as contacts, there are no restrictions on the number of times that can be used.
- A pulse relay cannot be specified as an output destination for an OT, KP, SET, RST or ALT instruction.
- A word unit pulse relay (WP) cannot be specified as a storage location for a high-level instruction.

2.5.9 E Error alarm relay

■ How error alarm relay (E) works

- Error alarm relays are used to feed back error conditions freely assigned by the user to internal relays, and to store them in memory.
- Error alarm relays are turned ON and OFF using the SET and RST instructions in the user program.
- When an error alarm relay goes ON, the number of error alarm relays which are on, the relay numbers, and the data of the calendar timer which went on first are stored in a memory area in the CPU unit.

System data register SD No.	Description	
SD80, SD81	Year/month data	Data for the calendar timer when the relay was initially turned ON
SD82, SD83	Day/hour data	
SD84, SD85	Minute/second data	
SD60	Number of relays that are turned ON	
SD61 to SD79	Number of relays that are turned ON	

- Information for up to 500 error alarm relays can be stored in the memory area. Relays that can be monitored or operated by the user, however, are those in the range from SD61 to SD79 only.

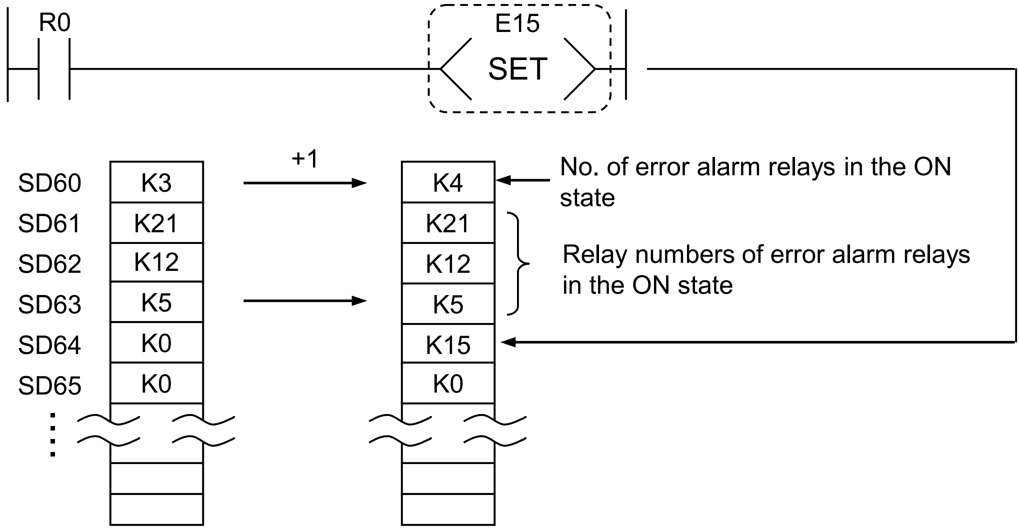
■ Usage restrictions

- An error alarm relay (E) cannot be specified as an output destination for an OT, KP, or ALT instruction.
- An error alarm relay (E) can be turned ON and OFF in multiple locations in the program, using the SET and RST instructions. However, no check is carried out for overlapping use.

■ Program for setting (turning on) an error alarm relays

- The SET instruction should be used to turn ON error alarm relays in the error alarm conditions, as shown below.
- Error alarm relays are held even if the error condition goes OFF.

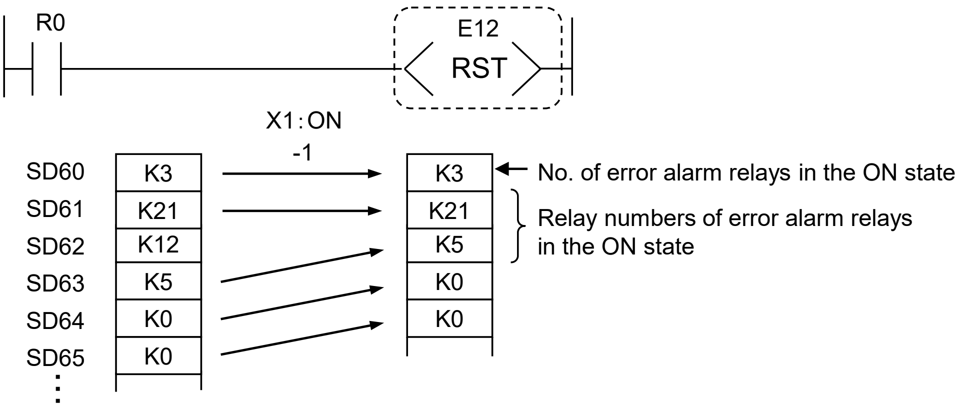
<Example> If R0 goes ON when an error occurs



■ Reset program for a given error alarm relay

- The RST instruction should be used to turn OFF error alarm relays when an error has been corrected.

<Example> If R0 goes ON when an error is corrected



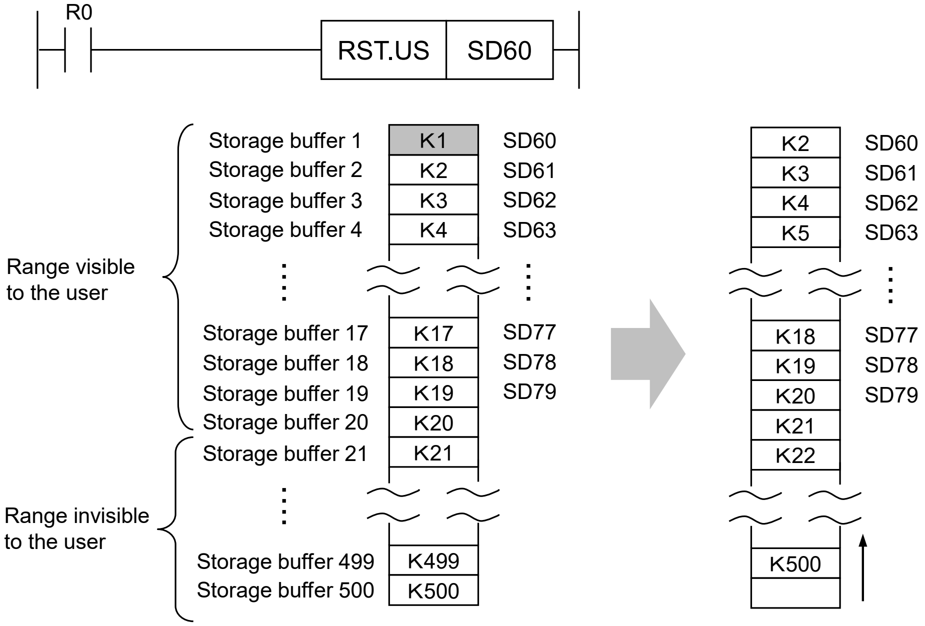
■ Clearing all buffer areas

- To reset all error alarm relays, use RST instruction to specify the system data register SD60, following the method shown below.

■ Clearing buffer areas and initial data

- Of the areas in which relay numbers are stored, only SD60 and SD61 can be cleared by directly specifying the system data register with RST instruction.
- When SD60 is specified, all of the error information in the buffer is cleared, and when SD61 is specified, the relay number at the head of the buffer area is cleared, and the buffer is filled as shown in the example below.

<Example> When the content of SD61 is deleted with RST instruction



2.5.10 IN Direct input

■ How direct input (IN) works

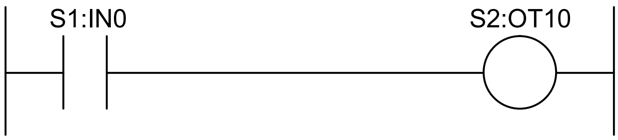
- The reading for external input (X) is updated as a batch when input/output refresh is executed, however if direct input (IN) is used, the reading for external input is updated when the operation is executed.
- This is effective for controls that require a high-speed response.
- This is specified in the program with a combination of a slot number (S1, S2...) and a corresponding address (IN0, IN1...).
- When using direct input (IN), it is necessary to select the setting to exclude this unit from the target for I/O refresh when configuring the settings for the I/O map.

2.5.11 OT Direct output

■ How direct output (OT) works

- External output (Y) is output as a batch when input/output refresh is executed, however if direct output (OT) is used, the operation result up until that time (ON/OFF) is output to the external output when the operation is executed.
- This is effective for controls that require a high-speed response.
- This is specified in the program with a combination of a slot number (S1, S2...) and a corresponding address (OT0, OT1...).

- When using direct output (OT), it is necessary to select the setting to exclude this unit from the target for I/O refresh when configuring the settings for the I/O map.



2.5.12 DT*.n Data register (bit specification)

- Refer to "2.6.2 DT*.n Data register (bit specification)".

2.5.13 LD*.n Link register (bit specification)

- Refer to "2.6.4 LD*.n Link register (bit specification)".

2.6 Description of the memory area

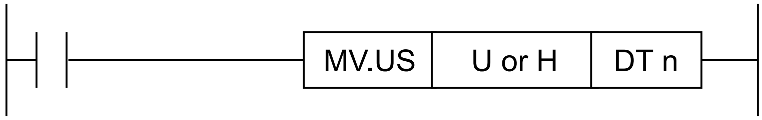
2.6.1 DT Data register

■ How data register (DT) works

- Data registers are memory areas which are handled in word (16-bit) units, and are used to store data such as numerical data that consists of 16 bits.

DTn	0 0 0 1	1 0 1 0	0 1 0 1	1 0 0 0

[Example of program to write numerical values into DTn]



- When 32-bit (double word) data is handled in data registers, use two data registers as a set. In the program, specify the number of the data register for the lower 16 bits.

DT n+1				DT n			
0001	1010	0101	0010	0001	0101	0101	1000

■ Hold data and non-hold data

- Data registers are categorized into two types, depending on how the register operates after powering off or switching from RUN to PROG. mode:
  1. Hold type data register, which remembers the ON/OFF status immediately before the data register is stopped and can resume operation using the same status after restarting
  2. Non-hold type data register, which resets the status when the data register is stopped
- Hold type and non-hold type settings can be set for each global device and local device.
- You can set the range of hold and non-hold types in **Setting of the no. of local devices to be used (in total)**>**Global hold-type start no.** in the "Memory Configuration" dialog box using the tool software FPWIN GR7. If the beginning of a hold type data register is specified using a word number, relays before that point will be non-hold types, and subsequent relays will be hold types.

"Memory configuration" >
"Setting of the no. of local devices to be used (in total)" > →
"Global holding start number"

Non-hold type
Hold type

2.6.2 DT*.n Data register (bit specification)

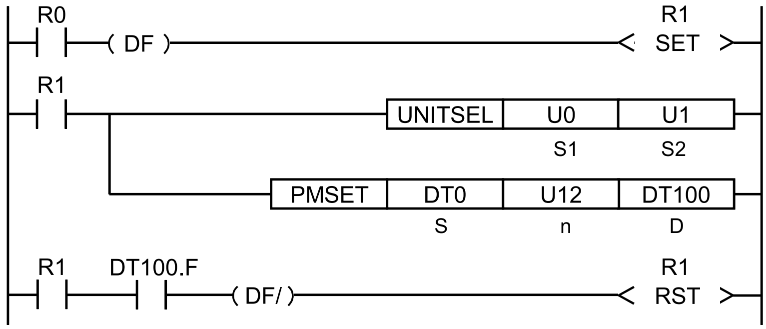
■ How bit specification of data register (DT*.n) works

- For the data register, specific bits of word data (16-bit data) can be extracted and used as bit data by using bit specification.

Examples of usable instructions

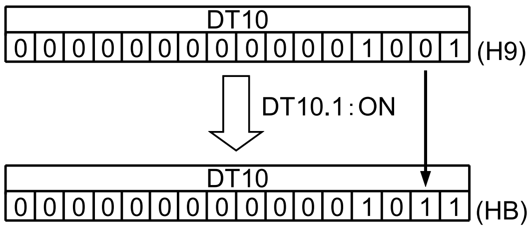
ST, ST/, ST↑, ST↓, AN, AN/, AN↑, AN↓, OR, OR/, OR↑, OR↓, OT, KP, SET, RST, ALT, SEND, RECV, PMSET

Example of program in which the F bit of DT100 is extracted and used as bit data



■ Precautions for use

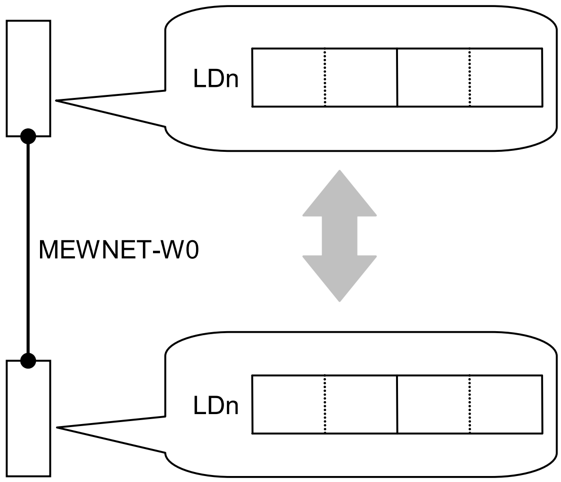
- If the ON/OFF status changes for any of the bit data of the data register (DT*.n), the value of the data register (DT) also changes.



2.6.3 LD Link register

■ How link register (LD) works

- Link register is data memory for when control units are connected in a network, as a "PLC link" data memory for shared use between control units.
- When data is written to the link register of a PLC, the same contents are stored in link registers with the same number for other PLC connected in the network.
- You can send and receive data between PLCs in this manner when using link registers.



■ **Range used by link register**

- The range of link registers that can be used varies depending on the network type and the combination of units.
- It is necessary to set the range of use and the number of points for each individual network.

■ **Setting of hold type register and non-hold type register**

- Link registers are categorized into two types, depending on how the register operates after powering off or switching from RUN to PROG. mode:
  1. Hold type register, which remembers the content of the register immediately before the link register is stopped, until operations resumes
  2. Non-hold type register, which resets the content when the data register is stopped
- You can configure the settings for the range of hold and non-hold types in **Setting of the no. of local devices to be used (in total)**>**Global hold-type start no.** in the "Memory Configuration" dialog box.
- Take note that when using a link register as a receiving link register, even if you set the link register to hold type in the configuration, hold operations will not be performed.
- Hold type and non-hold type settings can be set for each global device and local device.

2.6.4 LD*.n Link register (bit specification)

■ **How bit specification of link register (LD*.n) works**

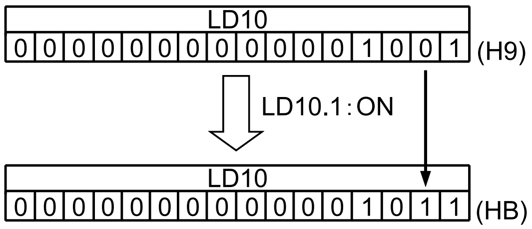
- For the link register, specific bits of word data (16-bit data) can be extracted and used as bit data by using bit specification.

Examples of usable instructions

ST, ST/, ST↑, ST↓, AN, AN/, AN↑, AN↓, OR, OR/, OR↑, OR↓, OT, KP, SET, RST, ALT, SEND, RECV, PMSET

■ Precautions for use

- If the ON/OFF status changes for any of the bit data of the link register (LD*.n), the value of the link register (LD) also changes.



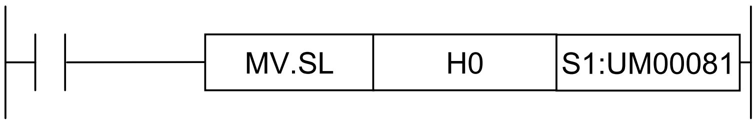
2.6.5 UM Unit memory

■ How unit memory (UM) works

- Unit memory is used to send and receive data between CPU units and intelligent units.
- The amount of data in unit memory and its allocation varies depending on the type of unit.
- The address for the unit memory is specified as a 5-digit hexadecimal. (H 0 to H 7FFFF (Maximum))
- In the program, an address is specified with a corresponding slot number (S1, S2...).

■ Unit memory (UM) usage example

- The following is an example of transferring the constant H0 to the address H 00081 of the unit memory (UM) of slot number 1.



2.6.6 SD System data register

■ How system data register (SD) works

- System data register is a memory area in which designated contents are stored.
- There are three types of areas: an area for read-only, an area for read-and-write, and an area used by the system.
- The types of system data registers are as follows.

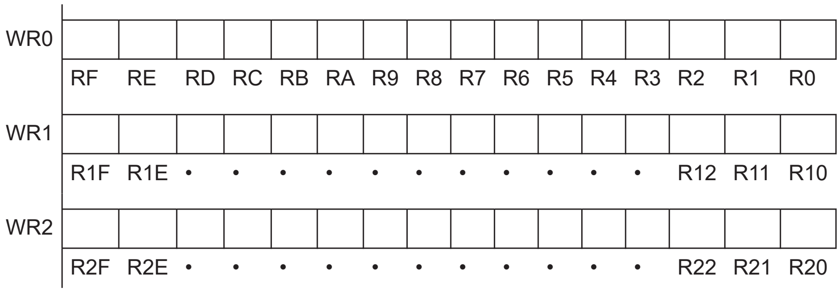
■ Types of system data registers (SD)

Classification	Function
Environment settings, operation status	The operation statuses of the PLC specified with the configuration data and the various types of instructions are stored. Example) Scan time
Description of error	Information such as information of a unit in which an abnormality occurred is stored. Example) Self-diagnostic error code, slot number of the unit in which the abnormality occurred, the address where the operation error occurred
Calendar timer	The year, month, day, hour, minute, second, and day of the week tracked by the calendar timer are stored here.

2.6.7 WX, WY, WR, WL

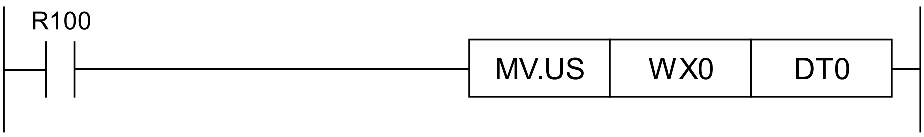
■ How WX, WY, WR, WL works

- Relays (X, Y, R, L) can be handled as blocks of 16 points.
- Pulse relays (P) and error alarm relays (E) cannot be handled in word units.
- These are one-word (16-bit) memory areas, thus they can be treated as data memory.
- The composition of the word-unit memory areas is shown below. Each element has a corresponding number, as shown below.



■ WX, WY, WR, WL usage example

- WX and WY can be used for reading the input from intelligent units and for the external input/output in word units.

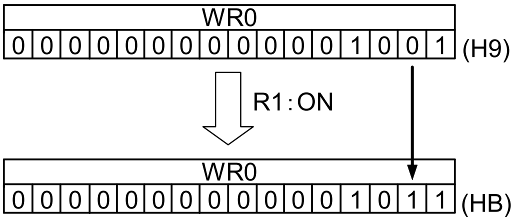


- WR can also be used as a shift register.

- All of the relays can be used to monitor 16-bit words.

■ Precautions for use

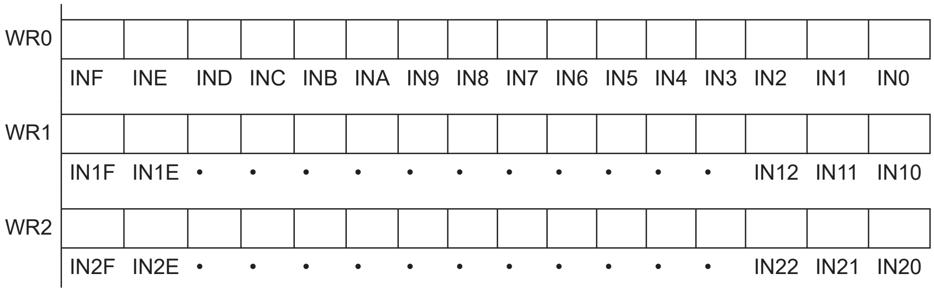
- If an ON or OFF status of one of the relays composing the memory area changes, the memory area value will also change.



2.6.8 WI, WO

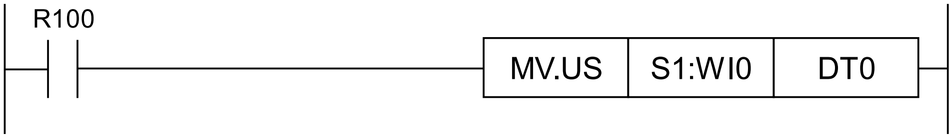
■ How WI, WO works

- Relays (IN, OT) can be handled as blocks of 16 points.
- These are one-word (16-bit) memory areas, thus they can be treated as data memory.
- The composition of the word-unit memory areas is shown below. Each element has a corresponding number, as shown below.



■ WI, WO usage example

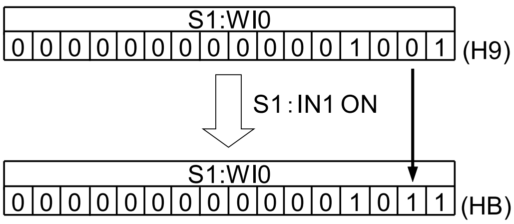
- When WI/WO is specified or when an operation is executed, input and output processing are performed.
- When using WI/WO, it is necessary to select the setting to exclude this unit from the target for I/O refresh when configuring the settings for the I/O map on the programming tool.



- This is specified in the program with a combination of a slot number (S1, S2...) and a corresponding address (WI0, WI1.../WO0, WO1...).
- All of the relays can be used to monitor 16-bit words.

■ Precautions for use

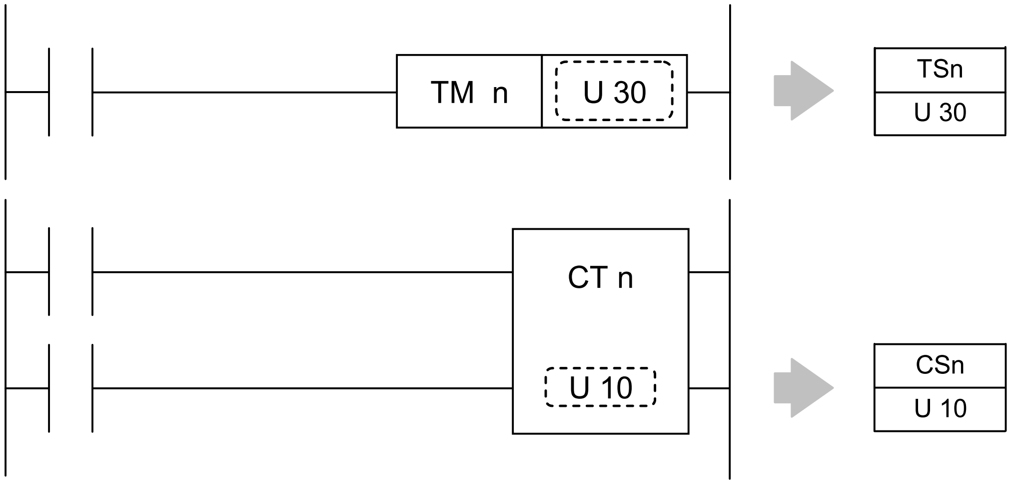
- If an ON or OFF status of one of the relays composing the memory area changes, the memory area value will also change.



2.6.9 TS, CS Timer/ Counter setting value register

■ How the set value area (TS, CS) works

- A set value for a timer (TS) or counter (CS) is stored in the set value area with the same number as the timer or counter.
- To set the value, specify a decimal number or the number of the set value area (TS/CS) when you enter the TM or CT instruction in the program.
- The set value area (TS/CS) is a 32-bit memory area which stores a decimal number from 0 to 4,294,967,295.



■ Using the set value area (TS, CS)

- During RUN mode, a set value for a timer or counter can be changed by rewriting the corresponding set value area.
- The values in the set value area can be overwritten by user programs such as data transfer instructions.
- Reading and writing of values can also be performed with the programming tool.

■ Reference:

- Numbers of TS and TE correspond to timer numbers, and numbers of CS and CE correspond to counter numbers.

Timer and counter No.	Set value area	Process value area
T0 T1 ⋮ T4095	TS0 TS1 ⋮ TS4095	TE0 TE1 ⋮ TE4095
C0 C1 ⋮ C1023	CS0 CS1 ⋮ CS1023	CE0 CE1 ⋮ CE1023

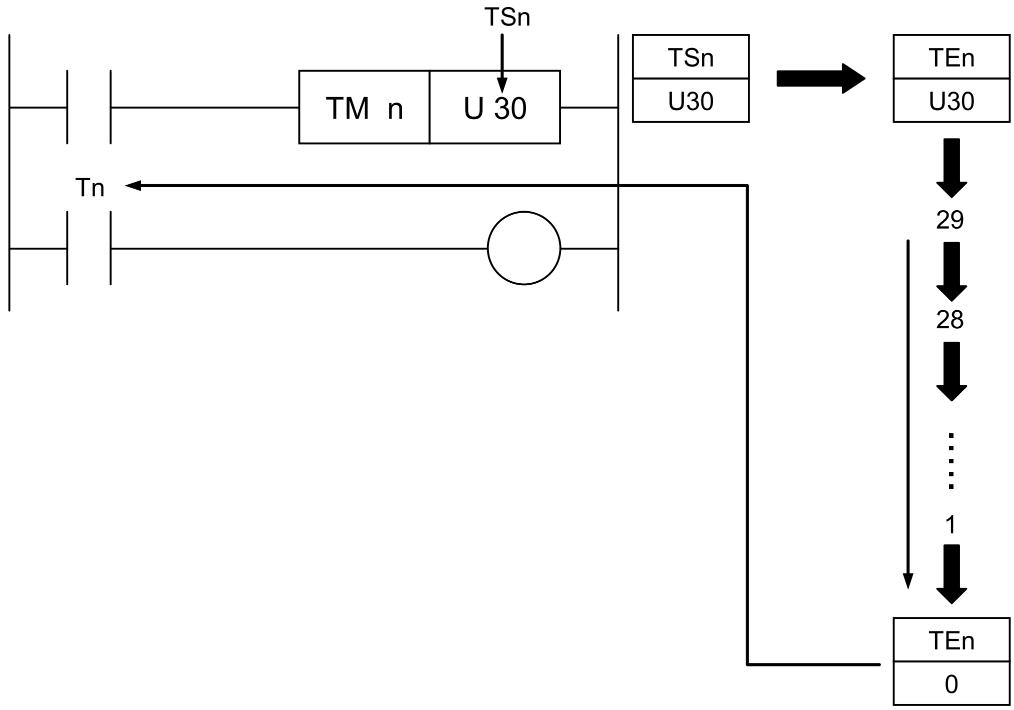
Note

- Timer/counter settings for FP7 series can be specified using unsigned constants (U constants).
- The set value area (TS, CS) occupies a 32-bit area.

2.6.10 TE, CE Timer/counter elapsed value register

■ How the elapsed value area (TE, CE) works

- While a timer or counter is operating, the elapsed value is stored in the elapsed value area (TE/CE) with the same number as the timer or counter.
- When the value of the elapsed value area (TE/CE) reaches zero, the timer or counter contact with the same number turns ON.
- The elapsed value area (TE/CE) is a 32-bit memory area which stores a decimal number from 0 to 4,294,967,295.



■ Using the elapsed value area (TE, CE)

- The elapsed value of a timer or counter in operation can be changed to prolong or shorten the operation.
- The values in the elapsed value area can be overwritten by user programs such as data transfer instructions.
- Reading and writing of values can also be performed with the programming tool.

■ Precautions for programming

- Timer/counter settings for FP7 series can be specified using unsigned constants (U constants).

Note

- The elapsed value area (TE, CE) occupies a 32-bit area.

2.6.11 I0 to IE Index registers

■ How index registers work

- Index registers are used to indirectly specify constants and memory area addresses.
- Depending on the values of the index register, changes to addresses and constants are called "index modification".
- Fifteen 32-bit registers are available for I0 to IE.

■ Notes on using index registers

- When the result of an address modifier exceeds the range for the memory area, an operation error occurs.

  Example) When the result of the modifier is an address with a negative value or value that is too large

■ Values that can be modified with index registers

- Number of the memory area used for a basic instruction
- Slot number or memory area number
- Number of the memory area used for a high-level instruction
- Value of the constant specified for a high-level instruction
- K constant (16-bit, 32-bit)
- U constant (16-bit, 32-bit)
- H constant (16-bit, 32-bit)

■ Value that cannot be modified with index registers

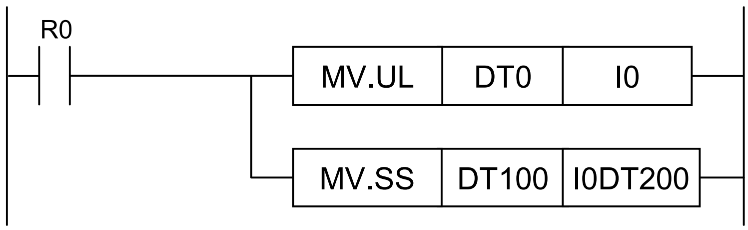
- Floating point data
- Depending on the instruction, index modification cannot be applied to some operands. Refer to the "Available devices" section of the description page for each instruction.

Note

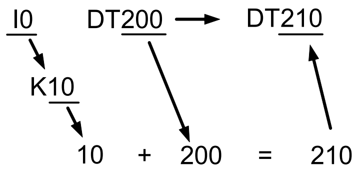
- An index register (I0 to IE) occupies a 32-bit area.

■ How to apply index modification

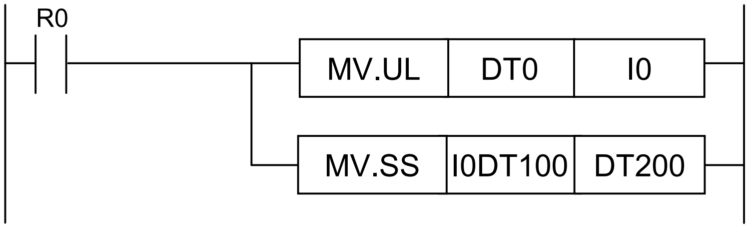
<Example 1> Modifying a transfer destination
The address in the data register (DT) for the transfer destination varies according to the value of DT0.



Example) When the value in DT0 is K10, the value in DT100 is written into DT210.

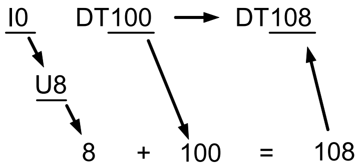


<Example 2> Modifying a transfer destination



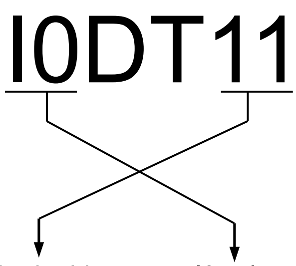
The address in the data register (DT) for the transfer destination varies according to the value of DT0.

Example) When the value of DT0 is U8, the value in DT108 is transferred to DT200.



■ Modifying an address

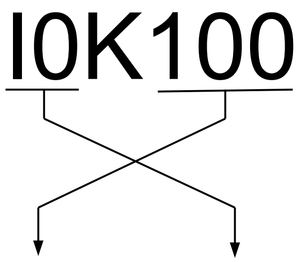
Address = basic address + value of I0 to IE



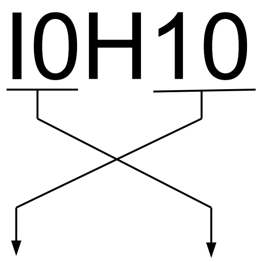
Standard address		I0 value		Targeted address
11	+	K0	=	DT11
11	+	K10	=	DT21
11	+	K-10	=	DT1

■ Modifying a constant

Constant = basic address + value of I0 to IE



Standard value		I0 value		Constant
K100	+	K0	=	K100
K100	+	K10	=	K110
K100	+	K-10	=	K90

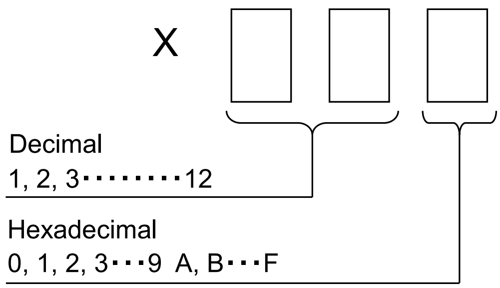


Standard value		I0 value		Constant
H10	+	HA	=	H1A
H10	+	H10	=	H20

■ Notes on index modification of a relay number

- For the external relay (X), external output relay (Y), and internal relay (R), when using index modification on relay numbers, be aware that the last digit of the relay number is hexadecimal and the higher digits are decimal.

<Example> For external input (X)



X 0, X 1, ・・・・・・・・・・・・・・X F
X10, X11, ・・・・・・・・・・・・・・X1F
X20, X21, ・・・・・・・・・・・・・・X2F
⋮ ⋮

■ Examples of index modification

Value of index register		Post-modification relay number
K	H	
0	0	X0
1	1	X1
⋮	⋮	⋮
9	9	X9
10	A	XA
⋮	⋮	⋮
15	F	XF
16	10	X10

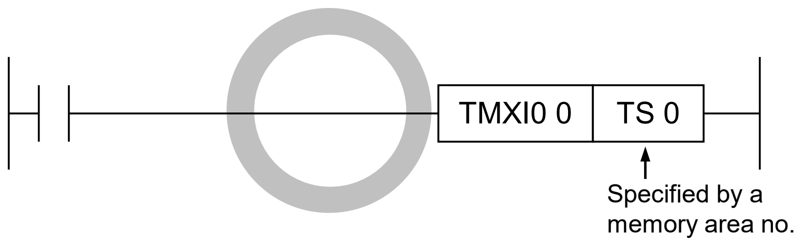
Value of index register		Post-modification relay number
K	H	
⋮	⋮	⋮
31	1F	X1F
⋮	⋮	⋮
159	9F	X9F
160	A0	X100
161	A1	X101
⋮	⋮	⋮
255	FF	X15F
256	100	X160
257	101	X161
⋮	⋮	⋮
266	10A	X16A
267	10B	X16B
⋮	⋮	⋮

■ Modification of No. of basic instruction

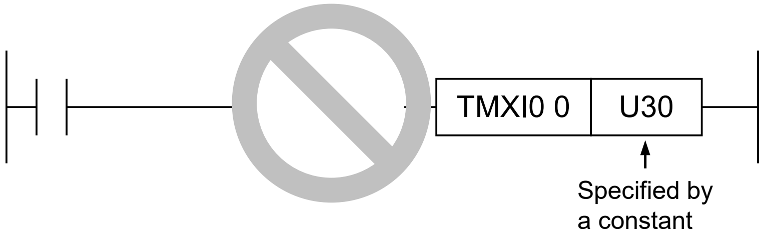
Object	Modification examples	
Timer number	Modify TML20	TML I020
Counter number	Modify CT200	CT I0200
Shift register number	Modify SRWR0	SR I0WR0
Specification of a label number in a jump instruction	Modify JP1	JP I01
Specification of a label number in a loop instruction	Modify LOOP5	LOOP I05
Subroutine program number	Modify CALL10	CALL I010

■ Limitations for modifying the timer/counter number

- The timer number and counter number can only be modified when a memory area has been specified for the setting value.



- When a constant is specified for the setting value, the timer/counter number cannot be modified.



■ Limitations for modifying DT.n (bit specification of data register)

When modifying DT.n (bit specification of data register), the maximum number of DT that can be specified is 32767.

■ Limitations for modifying LD.n (bit specification of link register)

When modifying LD.n (bit specification of link register), the maximum number of LD that can be specified is 16383.

2.7 Explanation of constants

2.7.1 K Signed decimal constant

■ How signed decimal constant (K) works

- The constant is a value which has been converted from binary data into decimal data. When entering the constant, specify "K" before the numerical value.
- The constant is primarily used to specify data sizes and quantities.
- In the PLC, the decimal constant (K) is processed as binary (BIN) data in word units of 16 bits, as shown below.
- The positive/negative sign is determined by the most significant bit (sign bit). A "0" indicates a positive sign (+), and a "1" indicates a negative sign (-).
- Data is normally handled in units of one word (16 bits), however, it is also occasionally handled in units of two words (32 bits). In this case, as well, the most significant bit serves as the sign bit.

■ Format for signed decimal constant (K)

<Example> Decimal "+32" (K32)

```
15        12         9         5         0
0 | 0 0 0 0 | 0 0 0 0 | 0 0 1 0 | 0 0 0 0
[+]
```

<Example> Decimal "-32" (K-32)

```
15        12         9         5         0
1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 0 | 0 0 0 0
[-]
```

■ Range that can be specified using a decimal constant (K)

Operation	Available range
16-bit operation	K-32,768 to K32,767
32-bit operation	K-2,147483648 to K2,147,483,647

2.7.2 U Unsigned decimal constant

■ How unsigned decimal constant (U) works

- The constant is a value which has been converted from binary data into decimal data. When entering and reading the constant, specify "U" before the numerical value.

- The constant is primarily used to specify data sizes and quantities such as the setting values for the timer.
- In the PLC, the decimal constant (U) is processed as binary (BIN) data in units of 16 bits, as shown below.
- For unsigned decimal (U), all 16 bits are used to represent a numerical value, so it is not possible to represent a negative value.
- Data is normally handled in units of one word (16 bits), however, it is also occasionally handled in units of two words (32 bits).

■ Format for unsigned decimal constant (U)

<Example> Decimal "+32" (U32)

15			12				9				5				0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

<Example> Decimal "+65504" (U65504)

15			12				9				5				0
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

■ Range that can be specified using a decimal constant (U)

Operation	Available range
16-bit operation	U0 to U65,535
32-bit operation	U0 to U4,294,967,295

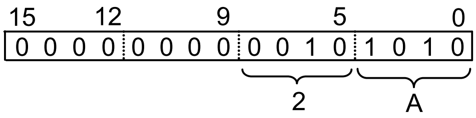
2.7.3 H Hexadecimal constant

■ How hexadecimal constant (H) works

- The constant is a value which has been converted from binary data into hexadecimal data. When entering and reading the constant, specify "H" before the numerical value.
- Hexadecimal constants are primarily used to specify an ordering of 1's and 0's in 16-bit data, such as system data register settings and specification of control data for high-level instructions. Hexadecimal constants are also used to specify BCD data.
- In the PLC, the hexadecimal constant (H) is processed as binary (BIN) data in units of 16 bits, as shown below.
- Data is normally handled in units of one word (16 bits), however, it is also occasionally handled in units of two words (32 bits).

■ Format for hexadecimal constant (H)

<Example> Hexadecimal "2A" (H2A)



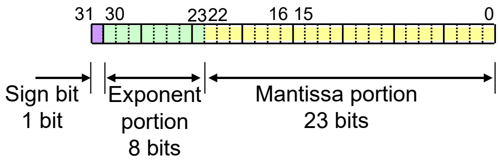
■ Range that can be specified for a hexadecimal constant (H)

Operation	Available range
16-bit operation	H0 to HFFFF
32-bit operation	H0 to HFFFFFFFF

2.7.4 SF Single-precision floating point real number constant

■ Format for single-precision floating point

- Following IEEE754 format, the double precision floating point format consists of 1 sign bit, 8 exponential bits, and 23 mantissa bits.
- Operation instructions for processing real numbers and conversion instructions for real number integers are provided by default, so it is not necessary to worry about data format while programming.



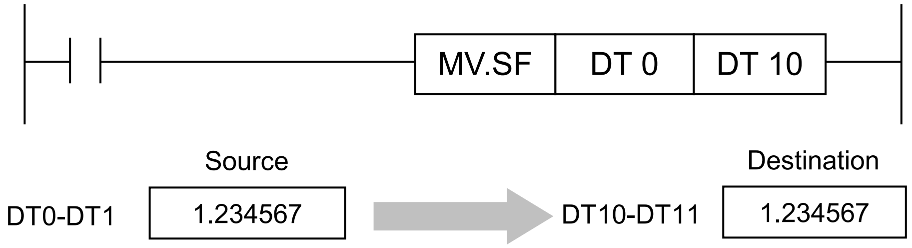
■ Range of single-precision floating point real number constants

Operation	Available range
32-bit operation	Negative range: -1.175494E-38 to -3.402823E+38 0 Positive range: 3.402823E+38 to 1.175494E-38

■ Storage area for single-precision floating point real number constants

- For the storage area for data of real numbers converted using the operation instructions for single-precision floating point real number constants, 1 unit of data uses an area of 2 words (32 bits).
- When using the storage area for real number data in data transfer instructions or other instructions, input SF for the operation unit.

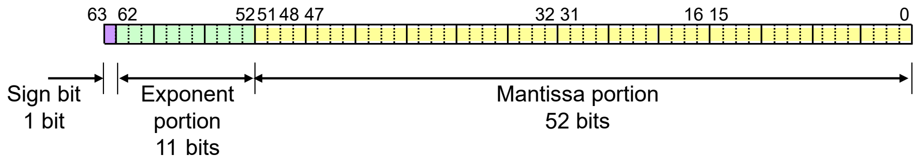
<Example> If SF is set for the operation unit of the instruction code, 2 words of data are included in DT.



2.7.5 DF Double-precision floating point real number constant

■ Format for double-precision floating point

- Following IEEE754 format, the double precision floating point format consists of 1 sign bit, 11 exponential bits, and 52 mantissa bits.
- Operation instructions for processing real numbers and conversion instructions for real number integers are provided by default, so it is not necessary to worry about data format while programming.



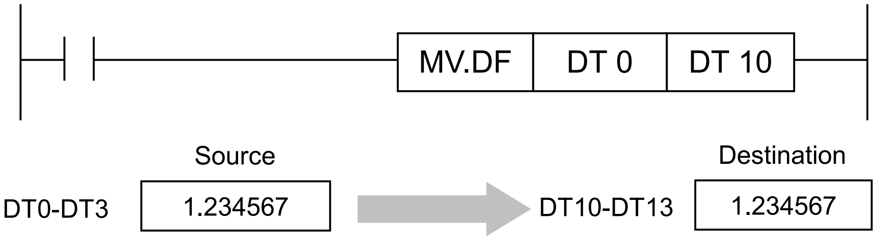
■ Range of double-precision floating point real number constants

Operation	Available range
64-bit operation	Negative range: -2.2250738585072014E-308 to -1.7976931348623158E+308 0 Positive range: +2.2250738585072014E-308 to +1.7976931348623158E+308

■ Storage area for double-precision floating point real number constants

- For the storage area for data of real numbers converted using the operation instructions for double-precision floating point real number constants, 1 unit of data uses an area of 4 words (64 bits).
- When using the storage area for real number data in data transfer instructions or other instructions, input DF for the operation unit.

<Example> If DF is set for the operation unit of the instruction code, 4 words of data are included in DT.



2.7.6 "" Character constant

■ How character constant ("") works

- Binary data is handled as ASCII code. When entering data, enclose characters with "".
- The instruction for which character constants can be set is SSET instruction (conversion of character constant → ASCII code). This can only be input using the tool software.
- Character constants ("") are stored as ASCII code in the specified memory area in the PLC as shown below.

■ Format for character constants ("")

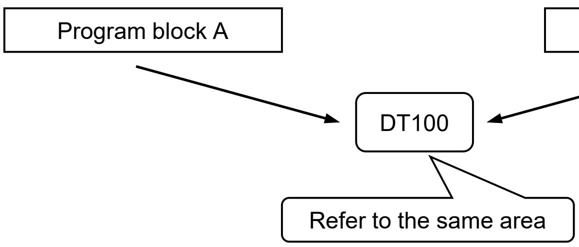
<Example> When character constant "MEWNET" is input

T	E	N	W	E	M	(Characters)
54	45	4E	57	45	4D	(ASCII codes)
1 word		1 word		1 word		

2.8 Global Devices and Local Devices

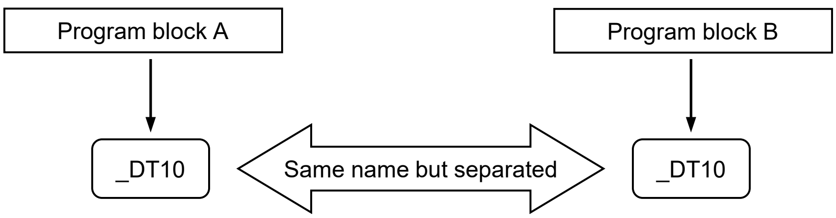
■ Global devices

- While a global device has a unique number throughout the entire program, a local device has a unique number inside each program block.
- For example, "Global device DT100" refers to the same data memory in program blocks A and B. Therefore, data of DT100 is overwritten if the relevant information is revised in either program block A or B.
- Global devices are utilized when the operation memory is required to be shared between multiple program blocks (PB) for external input/output or other purposes.



■ Local devices

- If data of local device dt10 is changed in either program block A or B, since the local device is allocated to different data memory, the data for one program block will not be overwritten by the other program block.



■ List of available memory areas for local devices (●: Available)

Bit device			16-Bit device:		32-Bit device:
X, Y, R, L, T, C	P, E, SR, IN, OT	DT.n, LD.n	WX, WY, WR, WL, DT, LD	SD, WI, WO, UM	TS, CS,TE, CE
●		●	●		●

2.9 Range of data that can be handled in the PLC

■ 16-bit data

Data handled in the PLC (binary 16-bit)	Decimal conversion		Hexa- decimal conversion
	Unsigned	Signed	
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	U32767	K32767	H7FFF
⋮	⋮	⋮	⋮
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	U 1	K 1	H1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	U 0	K 0	H0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	U65535	K -1	HFFFF
⋮	⋮	⋮	⋮
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	U32768	K-32768	H8000

■ 32-bit data

Data handled in the PLC (binary 32-bit)	Decimal conversion		Hexa- decimal conversion
	Unsigned	Signed	
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	U2147483547	K2147483547	H7FFFFFFF
⋮	⋮	⋮	⋮
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	U 1	K 1	H1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	U 0	K 0	H0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	U4294967295	K -1	HFFFFFFFF
⋮	⋮	⋮	⋮
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	U2147438648	K-2147438648	H80000000

■ Range of data that can be handled in the PLC

Type of operation			Operational range
Binary operation	16 bit	Unsigned	U 0 to U 65,535
		Signed	K -32,768 to K 32767
	32 bit	Unsigned	U 0 to U 4,294967295
		Signed	K -2,147,483,548 to K 2,147,483,54
BCD operation	16 bit	Unsigned	H 0 to H 9999
	32 bit	Unsigned	H 0 to H 99999999

(Note 1) If a value goes outside of the range shown above in either direction, it becomes overflow or underflow.

■ Representation of decimals in the PLC

- Decimals are processed as either 16-bit or 32-bit binary data.
- Signed data is processed as follows:

1. If the highest-order bit is a sign bit and a positive number, the value is "0", or if it is negative, the value is "1".
2. For positive numbers, the digits aside from the highest-order digit describe the size of the data in bits.

<Example> Representation of the decimal "1868"

0	0	0	0	0	1	1	1	0	1	0	0	1	1	0	0
	16,384	8,192	4,096	2,048	1,024	512	256	128	64	32	16	8	4	2	1

The other bits indicate the data size.

1,024 + 512 + 256 + 64 + 8 + 4 = 1868

The highest digit bit is '0', representing a positive figure.

3. For negative numbers, 2's complement is used. 2's complement is binary data in which 0 and 1 are inverted in the binary data for the positive number, and then 1 is added to the binary data.

<Example> Representation of the decimal "-4"

0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Decimal BIN represetation of a decimal '4'

Invert 0 and 1

1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1

Add 1

1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0

Decimal BIN represetation of a decimal '-4'

The highest digit bit becomes '1'.

2.10 Overflow and Underflow

■ What are overflow and underflow?

- A value resulting from operation instruction sometimes fails to satisfy the range that can be handled.
- When a value is larger than the maximum value it is called "overflow" and when a value is lower than the minimum value it is called "underflow".

■ Overflow and underflow in binary operations

- If numerical values go beyond the range shown below, those values become overflow or underflow.

<16-bit operation>

(Higher than the maximum value is overflow)

	Unsigned		Signed	
Maximum value	U65535	HFFFF	K32767	H7FFF
	⋮	⋮	⋮	⋮
	U32768	H8000	K 1	H1
	U32767	H7FFF	K 0	H0
	⋮	⋮	K -1	HFFFF
	U 1	H1	⋮	⋮
Minimum value	U 0	H0	K-32768	H8000

(Lower than the minimum value is underflow)

<32-bit operation>

(Higher than the maximum value is overflow)

	Unsigned		Signed	
Maximum value	U4294967295	HFFFFFFFF	K2147483647	H7FFFFFFF
	⋮	⋮	⋮	⋮
	U2147483648	H80000000	K 1	H1
	U2147483647	H7FFFFFFF	K 0	H0
	⋮	⋮	K -1	HFFFFFFFF
	U 1	H1	⋮	⋮
Minimum value	U 0	H0	K-2147483648	H80000000

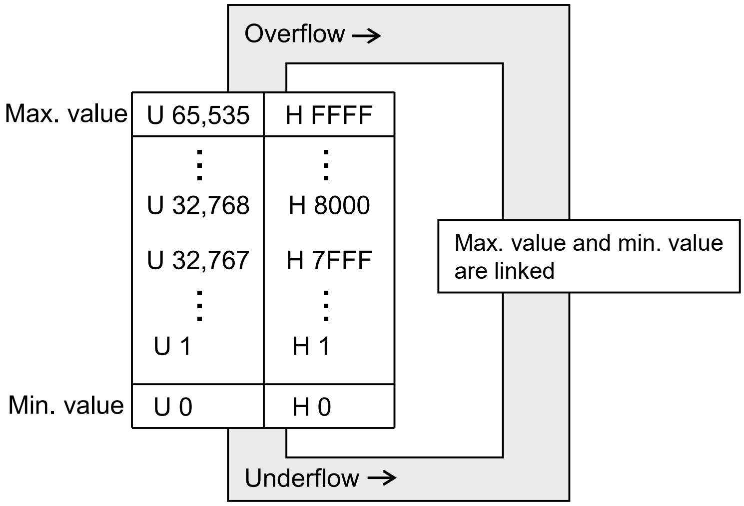
(Lower than the minimum value is underflow)

■ Values for overflow and underflow

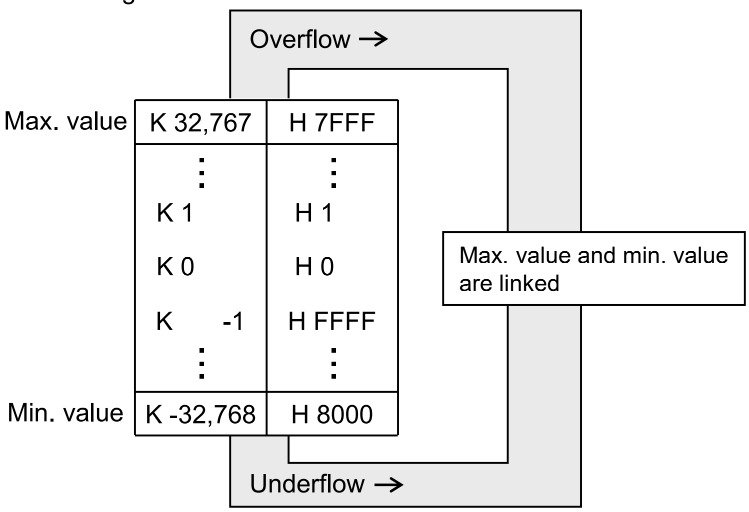
- As shown in the figures, the numerical values that can be handled by an operation are in a loop status, in which the maximum value and minimum value are connected.

Example 1) 16-bit binary operation

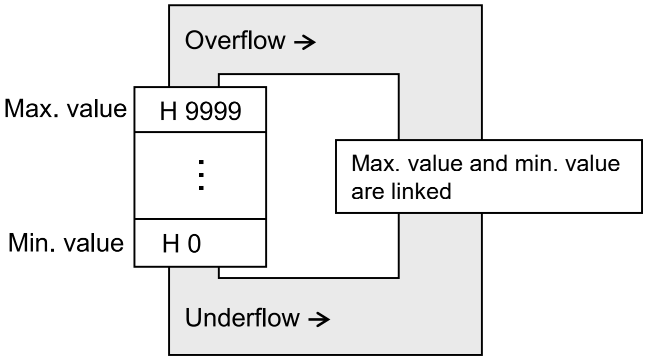
Unsigned



Signed



Example 2) 16-bit BCD operation

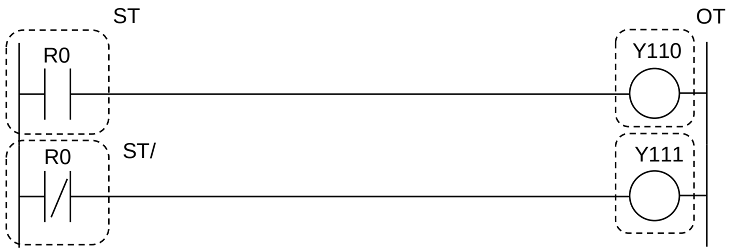


(MEMO)

3 Basic Instructions

3.1 ST, ST/, OT (START, START NOT, OUT)

■ Ladder diagram



■ Devices that can be specified (indicated by •)

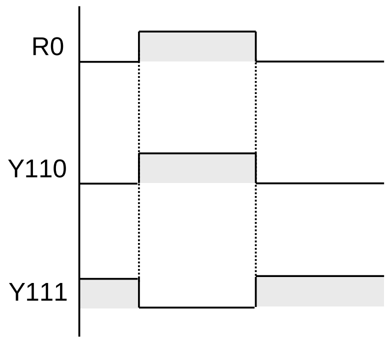
Operand		Bit device											Specification of bit of word device		Index modifier
		X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
ST, ST/	bit	•	•	•	•	•	•	•	•	•	•	•	•	•	•
OT		•	•	•	•							•	•	•	•

■ Outline of operation

Type of instruction	Operation
ST	Begins a logic operation by treating the input contact specified by the ST instruction as a Form A (normally open).
ST/	Begins a logic operation by treating the input contact specified by the ST/ instruction as a Form B (normally closed).
OT/	Outputs the operation result to the specified coil.

■ Operation Example

Program operation of "ST", "ST/", and "OT" in the ladder diagram
Output to Y110 when R0 is ON, and to Y111 when R0 is OFF.

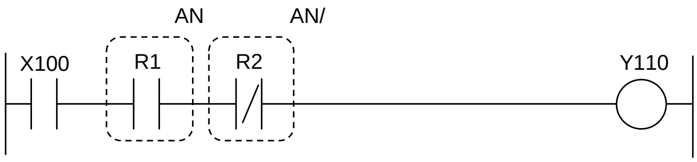


■ Precautions for programming

- The "ST" and "ST/" instructions are initiated from the bus bar.
- When an external switch (e.g., emergency stop switch) is a Form B (normally closed), be sure to use the ST instruction in the program.

3.2 AN, AN/ (AND, AND NOT)

■ Ladder diagram



■ Devices that can be specified (indicated by ●)

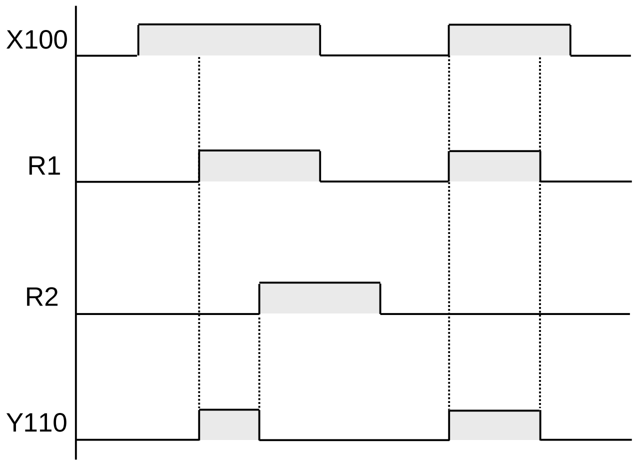
Operand		Bit device											Specification of bit of word device		Index modifier
		X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
AN, AN/	bit	●	●	●	●	●	●	●	●	●	●	●	●	●	●

■ Outline of operation

Type of instruction	Operation
AN	Executes an AND operation with the preceding operation result in serial connection by treating the input contact specified by the AN instruction as a Form A (normally open).
AN/	Executes an AND operation with the preceding operation result in serial connection by treating the input contact specified by the AN/ instruction as a Form B (normally closed).

■ Operation Example

Program operation of "AN" and "AN/" in the ladder diagram
When X100 and R1 turn ON and R2 turns OFF, the result is output to Y110.

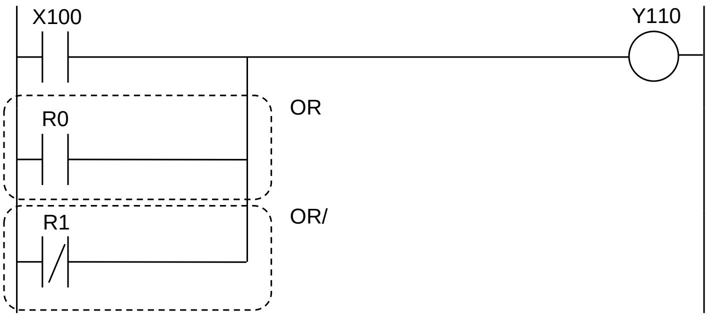


■ Precautions for programming

- The "AN" and "AN/" instructions can be used consecutively.

3.3 OR, OR/ (OR, OR NOT)

■ Ladder diagram



■ Devices that can be specified (indicated by •)

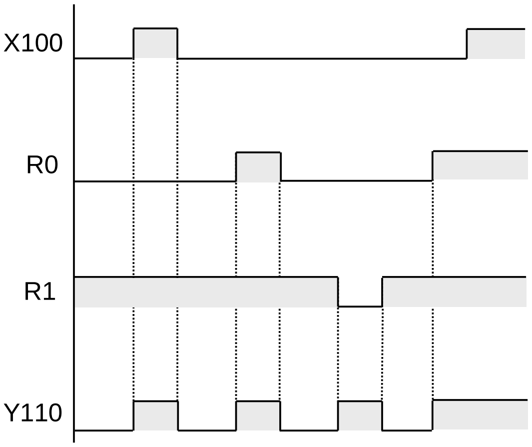
Operand		Bit device											Specification of bit of word device		Index modifier
		X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
OR, OR/	bit	•	•	•	•	•	•	•	•	•	•	•	•	•	•

■ Outline of operation

Type of instruction	Operation
OR	Executes an OR operation with the preceding operation result in parallel connection by treating the input contact specified by the OR instruction as a Form A (normally open).
OR/	Executes an OR operation with the preceding operation result in parallel connection by treating the input contact specified by the OR/ instruction as a Form B (normally closed).

■ Operation Example

Program operation of "OR" and "OR/" in the ladder diagram
If any of the conditions of X100 ON, R1 ON, or R2 OFF is satisfied, the result is output to Y110.

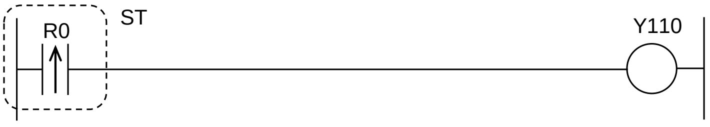


■ Precautions for programming

- The "OR" and "OR/" instructions are initiated from the bus bar.
- The "OR" and "OR/" instructions can be used consecutively.

3.4 ST↑, ST↓ (Leading and Trailing Contact Instructions)

■ Ladder diagram



■ Devices that can be specified (indicated by ●)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
bit	●	●	●	●	●	●	●	●	●	●	●	●	●	

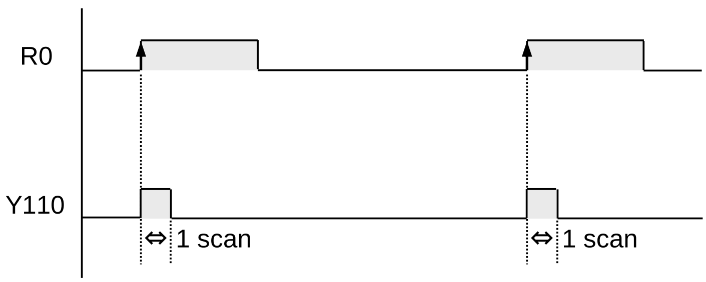
■ Outline of operation

Type of instruction	Operation
ST↑	Conduction takes place for 1 scan only following the change of a signal from the OFF to ON state (rise). Begins a logic operation by treating the input contact as a Form A (normally open) or Form B (normally closed).
ST↓	Conduction takes place for 1 scan only following the change of a signal from the ON to OFF state (fall). Begins a logic operation by treating the input contact as a Form A (normally open) or Form B (normally closed).

■ Operation Example

Program operation of "ST↑" in the ladder diagram

When R0 changes from OFF to ON (rise), only 1 scan is output to Y110.



3.5 AN↑, AN↓ (Leading and Trailing Contact Instructions)

■ Ladder diagram

■ Devices that can be specified (indicated by •)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
bit	•	•	•	•	•	•	•	•	•	•	•	•	•	

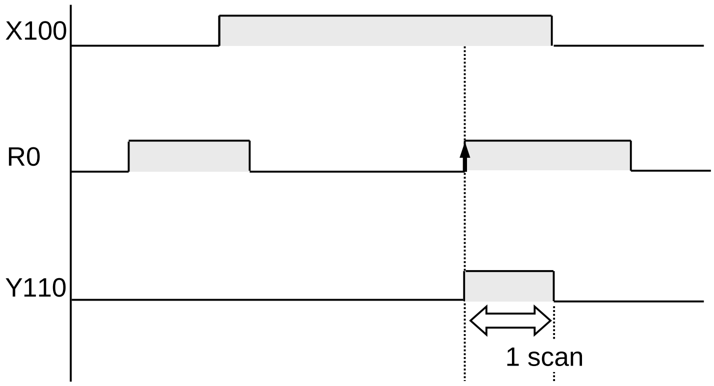
■ Outline of operation

Type of instruction	Operation
AN↑	Conduction takes place for 1 scan only following the change of a signal from the OFF to ON state (rise). Executes an AND operation with the preceding operation result in serial connection by treating the input contact as a Form A (normally open) or Form B (normally closed).
AN↓	Conduction takes place for 1 scan only following the change of a signal from the ON to OFF state (fall). Executes an AND operation with the preceding operation result in serial connection by treating the input contact as a Form A (normally open) or Form B (normally closed).

■ Operation Example

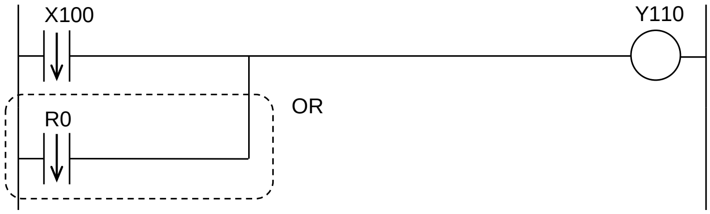
Program operation of "AN↑" in the ladder diagram

Output to Y110 takes place for 1 scan only following the change of R0 from the OFF to ON state (rise) when X100 is ON.



3.6 OR↑, OR↓ (Leading and Trailing Contact Instructions)

■ Ladder diagram



■ Devices that can be specified (indicated by •)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
bit	•	•	•	•	•	•	•	•	•	•	•	•	•	

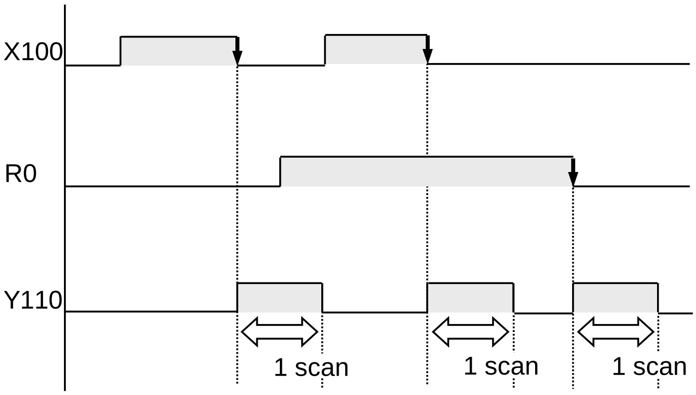
■ Outline of operation

Type of instruction	Operation
OR↑	Conduction takes place for 1 scan only following the change of a signal from the OFF to ON state (rise). Executes an OR operation with the preceding operation result in parallel connection by treating the input contact as a Form A (normally open) or Form B (normally closed).
OR↓	Conduction takes place for 1 scan only following the change of a signal from the ON to OFF state (fall). Executes an OR operation with the preceding operation result in parallel connection by treating the input contact as a Form A (normally open) or Form B (normally closed).

■ Operation Example

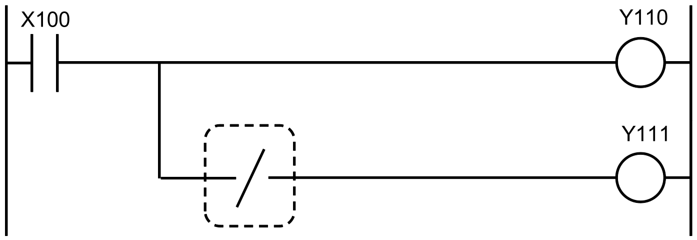
Program operation of "OR↑" in the ladder diagram

Output to Y110 takes place for 1 scan only following the change of X100 or R0 from the ON to OFF state (fall).



3.7 / (NOT)

■ **Ladder diagram**

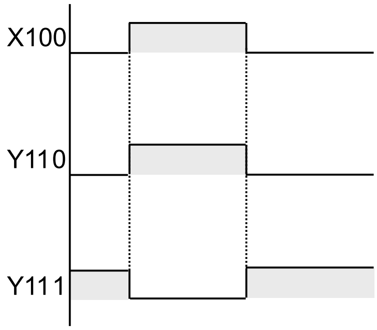


■ **Outline of operation**

- The / instruction inverts the preceding operation result.

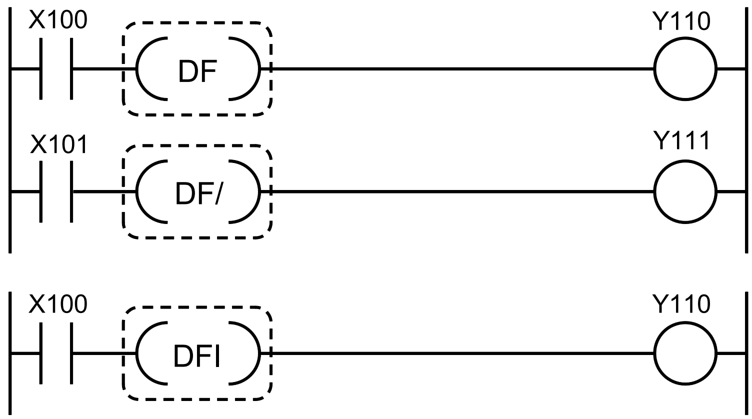
■ **Operation Example**

When X100 is ON, Y110 is ON and Y111 is OFF.



3.8 DF, DF/ (Leading Edge Differential, Trailing Edge Differential), DFI (Leading Edge Differential (Initial Execution Type))

■ Ladder diagram



■ Outline of operation

Type of instruction	Operation
DF	The DF instruction only generates output (differential output) for a single scan where the execution condition changes from the OFF state to the ON state (i.e., rises).
DF/	The DF/ instruction only generates output (differential output) for a single scan where the execution condition changes from the ON state to the OFF state (i.e., falls).
DFI	The DFI instruction only generates output (differential output) for a single scan where the execution condition changes from the OFF state to the ON state (i.e., rises).

■ Precautions for programming

- There are no restrictions on the number of times the differential instruction (DF, DF/, or DFI) can be used.
- The differential instruction only detects changes in the ON/OFF states of a contact.
- The DF and DF/ instructions do not generate output if the execution condition is already satisfied (i.e., ON state) when the operation mode is switched to RUN or the power is turned on in the RUN mode.
- The DFI instruction generates output (differential output) for the first single scan even if the execution condition is already satisfied when the RUN mode is initiated.
- If the execution condition can be already satisfied (i.e., on state) when the operation mode is switched to RUN or the power is turned on in the RUN mode, output will not be obtained for the first single scan by the DF instruction. Use the DFI instruction instead in this case.
- Be careful when using a differential instruction with an instruction (1 to 6 below) which changes the order of instruction execution such as MC - MCE, JP - LBL.

  1) MC - MCE instructions

  2) JP - LBL instructions

  3) LOOP - LBL instructions

  4) CNDE instruction

5) Step ladder instructions

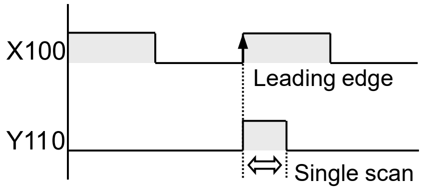
6) Subroutine instructions

- When a differential instruction is combined with an AND stack instruction or a pop stack instruction, take care that the syntax is correct.

■ Operation Example

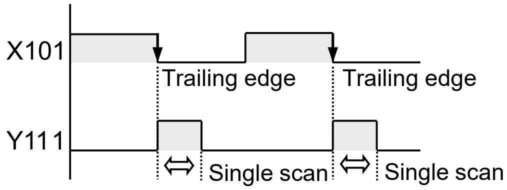
(1) Program operation for "DF" in ladder diagram

Outputs to Y110 only for a single scan where X100 changes from the OFF state to the ON state (i.e., rises).



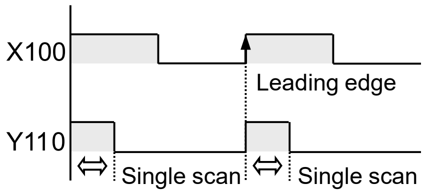
(2) Program operation for "DF/" in ladder diagram

Outputs to Y111 only for a single scan where X101 changes from the ON state to the OFF state (i.e., falls).



(3) Program operation for "DFI" in ladder diagram

Outputs to Y110 only for a single scan where X100 changes from the OFF state to the ON state (i.e., rises).

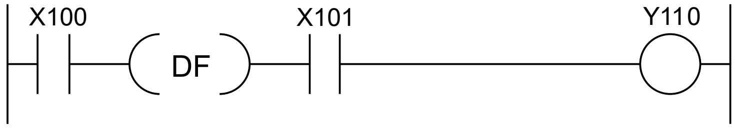


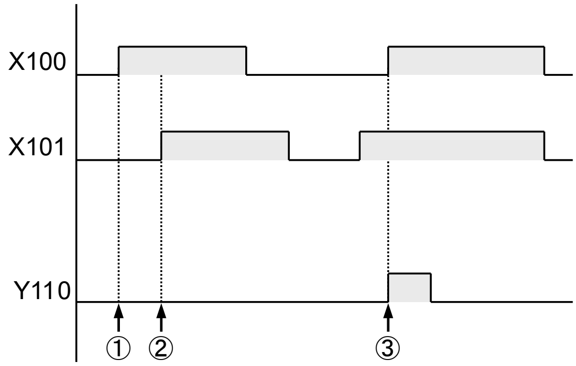
■ Sample program

- For the circuit shown below, the operation is as follows.

(1) Example 1 using DF instruction

→ When the leading edge differential is set between input information (X100, X101)





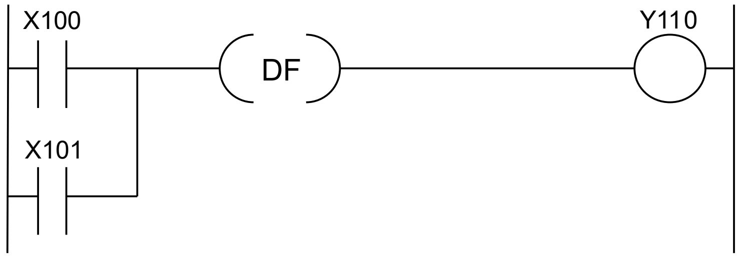
1) While X101 is OFF, Y110 remains OFF even when X100 rises.

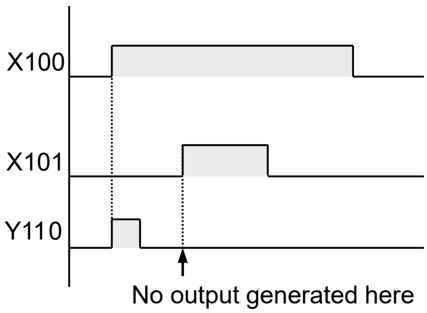
2) While X100 is ON, Y110 remains OFF even when X101 rises.

3) While X101 is ON, Y110 becomes ON for a single scan when X100 rises.

(2) Example 2 using DF instruction

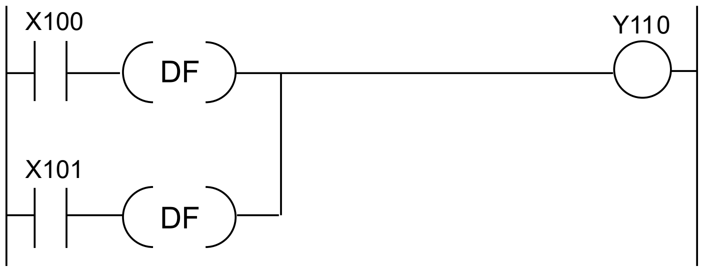
→ When the leading edge differential is set after parallel connection of input information (X100, X101)

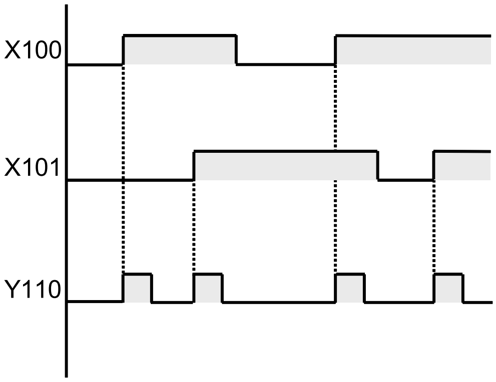




(3) Example 3 using DF instruction

→ When the leading edge differential is set for each input information (X100, X101)



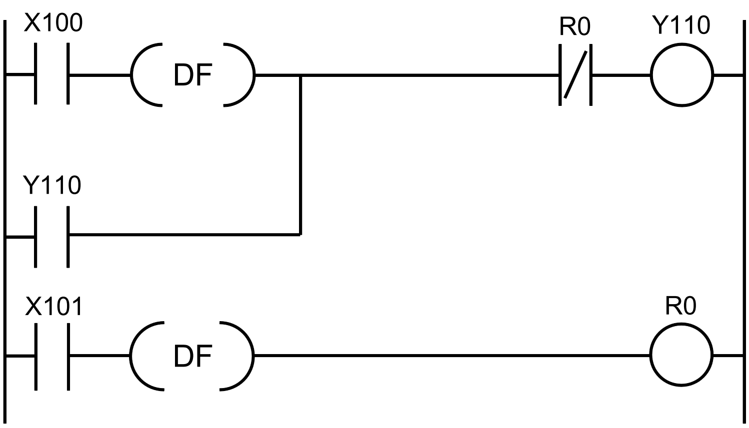


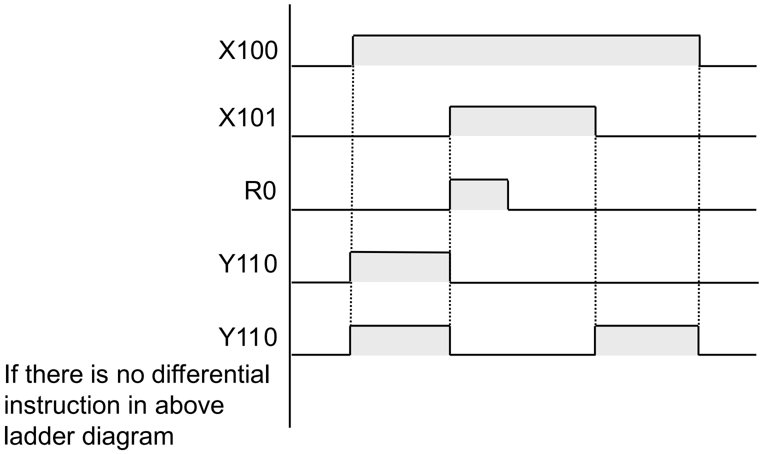
Examples of applying differential instructions

- Using differential instructions makes it easy to create and adjust programs.

<Example of application to a self-holding circuit>

- Using a differential instruction allows longer input signals to be supported.

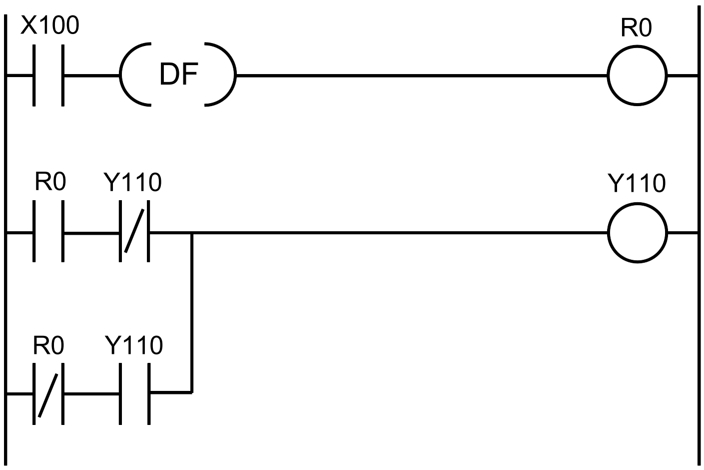




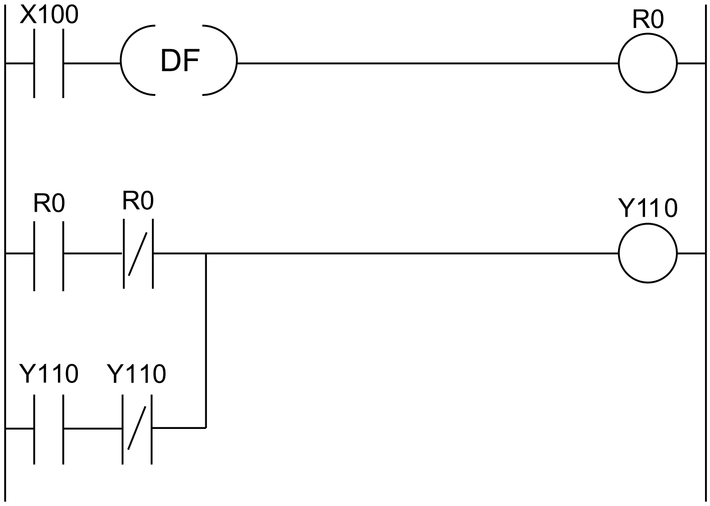
<Application example for alternating circuit>

- Differential instructions can also be applied to an alternating circuit which uses a single signal to hold and release the circuit.

＜例1＞

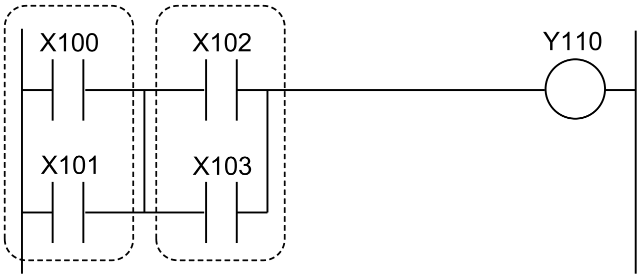


<Example 2>



3.9 ANS (AND stack)

■ Ladder diagram



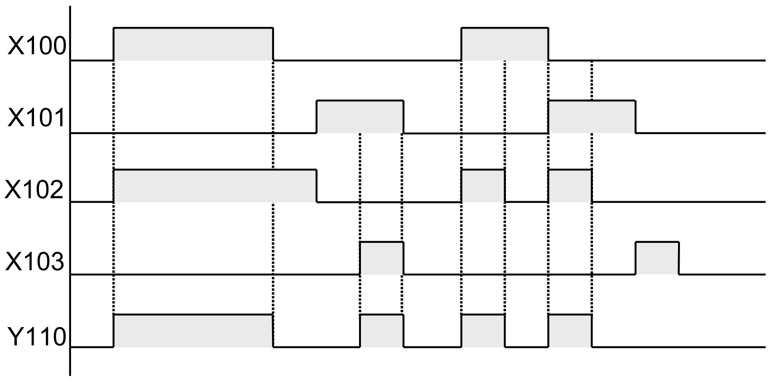
■ Outline of operation

- Blocks that were connected in parallel are connected in series.
- Each block should start with the ST instruction.

■ Operation Example

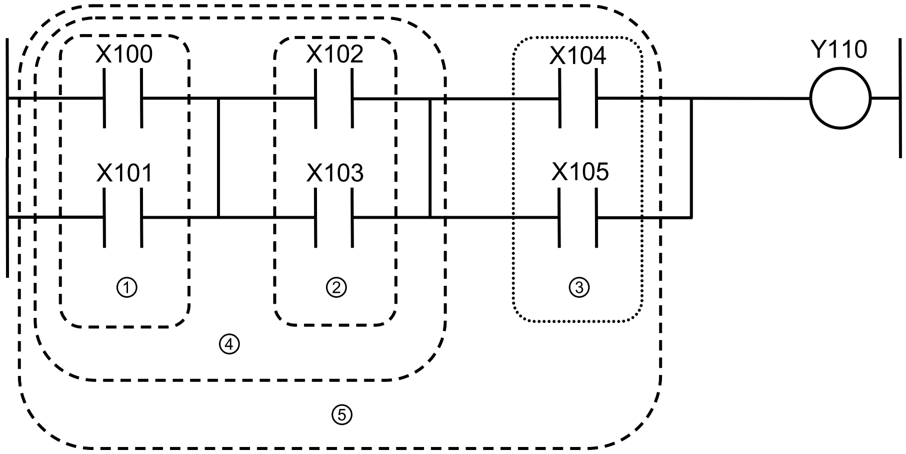
In the above ladder diagram, when X100 or X101 is ON and X102 or X103 is ON, the signal is output to Y110.

(X100 OR X101) AND (X102 OR X103) → Y110



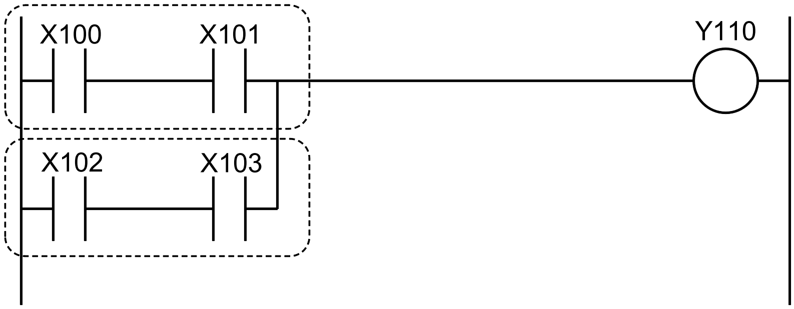
■ When blocks are consecutive

- When blocks are consecutive, consider a block division as follows.


X100
X102
X104
Y110
X101
X103
X105
①
②
③
④
⑤

3.10 ORS (OR Stack)

■ Ladder diagram



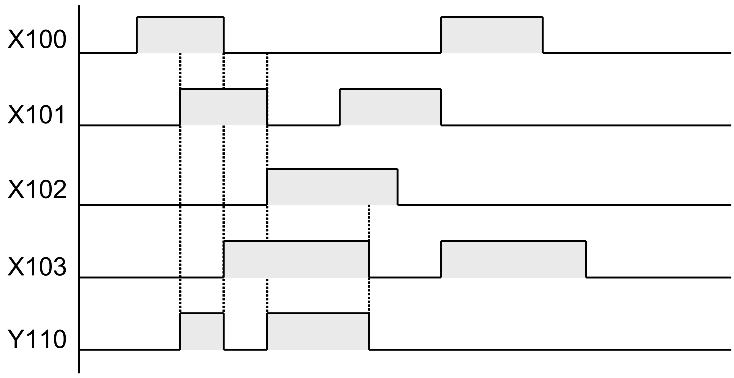
■ Outline of operation

- Serially connected blocks are connected in parallel.
- Each block should start with the ST instruction.

■ Operation Example

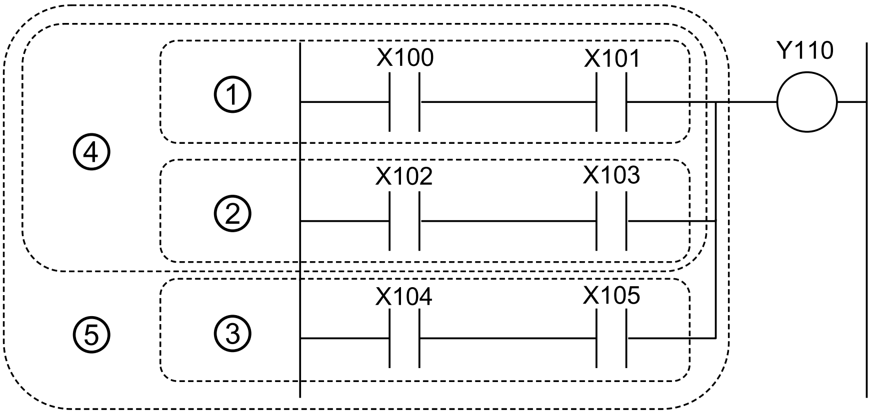
In the above ladder diagram, when X100 and X101 are ON or X102 and X103 are ON, the signal is output to Y110.

(X100 AND X101) OR (X102 AND X103) → Y110



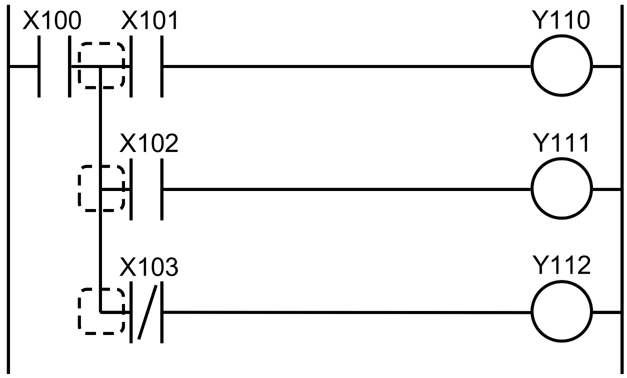
■ When blocks are consecutive

- When blocks are consecutive, consider a block division as follows.


X100
X101
Y110
①
④
X102
X103
②
X104
X105
⑤
③

3.11 PSHS (Push stack), RDS (Read stack), POPS (Pop stack)

■ Ladder diagram



■ Outline of operation

Type of instruction	Operation
PSHS	The operation result immediately before the PSHS instruction is stored and operation continues from the next step.
RDS	Reads the operation result saved by the PSHS instruction, and continues with the operation using it from the next step.
POPS	Reads the operation result saved by the PSHS instruction, continues with the operation using it from the next step, and resets the operation result saved by the PSHS instruction.

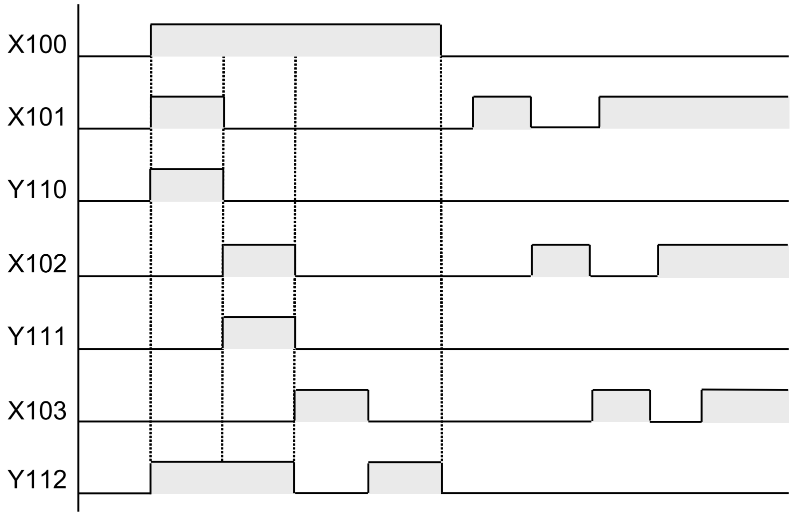
- Saves a single operation result, and reads it and performs multiple operations.
- This instruction is used to branch from a single contact and connect to further contacts.

■ Operation Example

1) When X100 is ON, the PSHS instruction saves the operation result and outputs it to Y110 if X101 is ON.

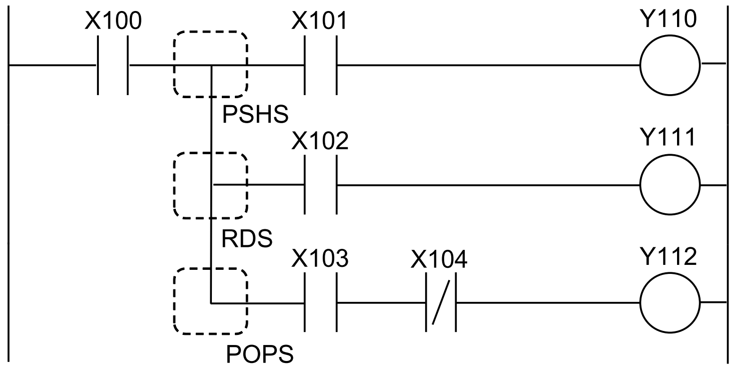
2) The RDS instruction reads the operation result and outputs it to Y111 if X102 is ON.

3) The POPS instruction reads the operation result and outputs it to Y112 if X103 is OFF, and resets the operation result saved by the PSHS instruction.

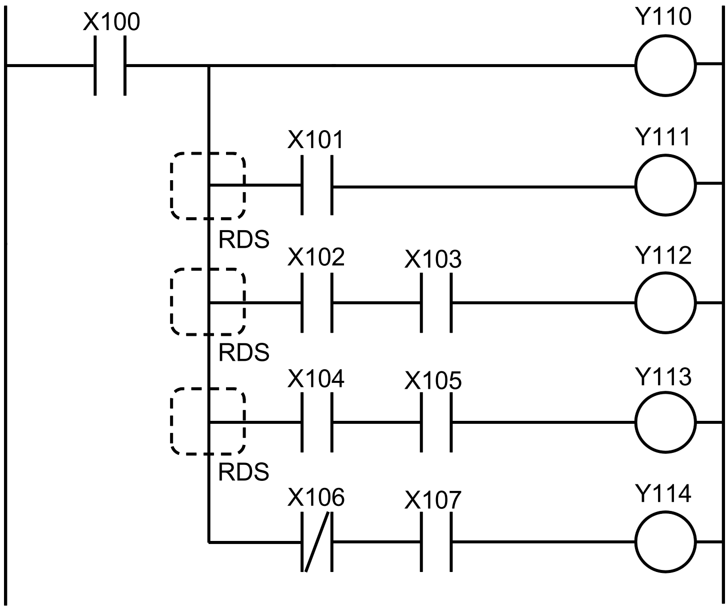


■ Precautions for programming

- Use the RDS instruction when the operation result will be further used and the POPS instruction if it will not be used any more. (Be sure to use the POPS instruction at the end of a series of instructions.)

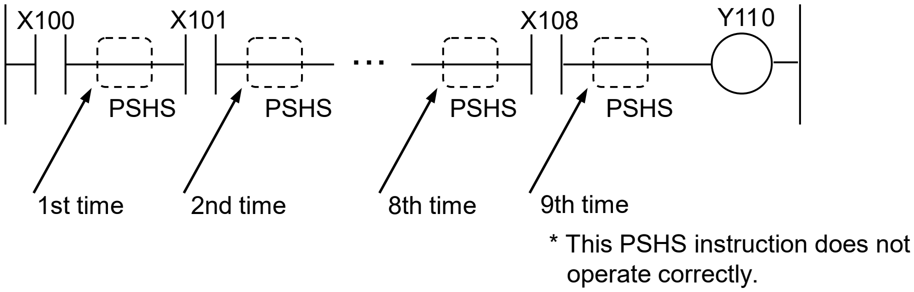


- The RDS instruction can be used repeatedly for an unlimited number of times.

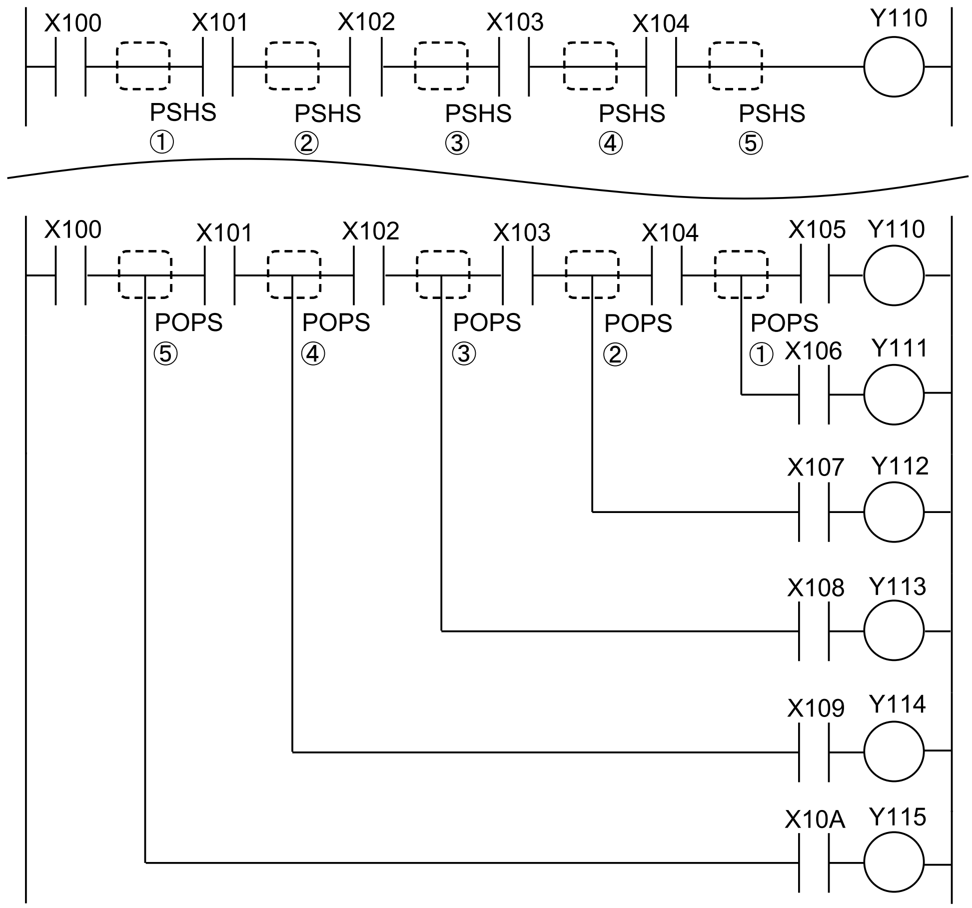


■ Precaution when using PSHS instruction repeatedly

- There is a limit on the number of times the PSHS instruction can be used repeatedly. The maximum number of times it can be used repeatedly before using the next POPS instruction is eight.
- Please note that the program will not run correctly if this limit is exceeded.

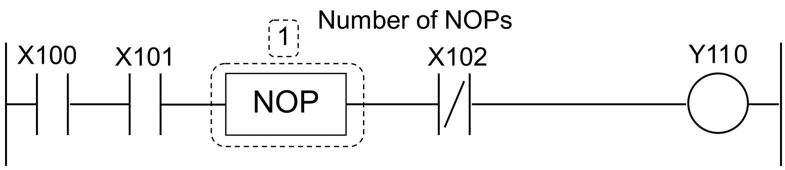


- If the POPS instruction is used while the PSHS instruction is used repeatedly, the operation results will be read with the result saved by the last PSHS instruction first. The numbers in the figure correspond to the instruction results.


X100
X101
X102
X103
X104
Y110
PSHS ①
PSHS ②
PSHS ③
PSHS ④
PSHS ⑤
X100
X101
X102
X103
X104
X105
Y110
POPS ⑤
POPS ④
POPS ③
POPS ②
POPS ①
X106
Y111
X107
Y112
X108
Y113
X109
Y114
X10A
Y115

3.12 NOP

■ Ladder diagram



■ Outline of operation

- This instruction has no effect on the operation results to that point. The program will operate in the same manner whether or not the NOP instruction is used.
- The NOP instruction may be used to make it easy to view the program code when reviewing and/or modifying it.
- To erase an instruction without changing the program address, write the NOP instruction over it.
- To send the address of a program portion without modifying the program, insert the NOP instruction.
- For example, this is a convenient means of breaking a long program into several blocks.

■ Sample program

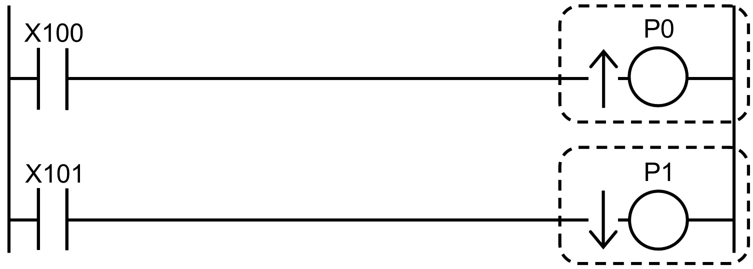
addresses			addresses			
0	ST	X100	0	ST	X100	
1	AN	X101	1	AN	X101	
2	AN/	X102	2	NOP		← NOP instruction is inserted here and the addresses change.
3	OT	Y100	3	AN/	X102	
			4	OT	Y100	

■ Deleting NOP instruction

- After creating a program, all NOP instructions contained in it can be deleted using a programming tool.

3.13 ↑OT, ↓OT (Leading, Trailing Edge Out)

■ Ladder diagram



■ Devices that can be specified (indicated by •)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
bit							•							•

■ Outline of operation

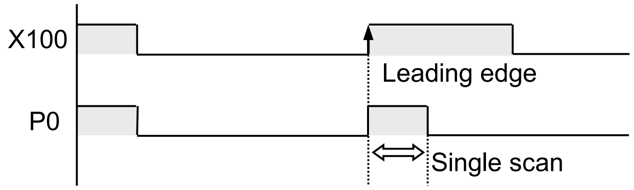
Type of instruction	Operation
↑OT	Generates output only for a single scan where the preceding operation result changes from the OFF state to the ON state (i.e., rises).
↓OT	Generates output only for a single scan where the preceding operation result changes from the ON state to the OFF state (i.e., falls).

■ Operation Example

(1) Program operation for "↑OT" in ladder diagram

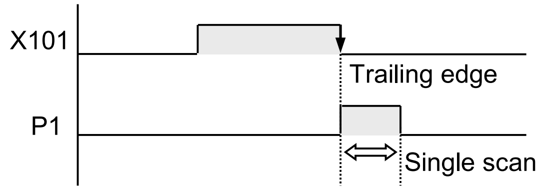
Outputs to the pulse relay P0 only for a single scan where X100 changes from the OFF state to the ON state (i.e., rises).

Also outputs to P0 even if X100 is ON for the first scan.



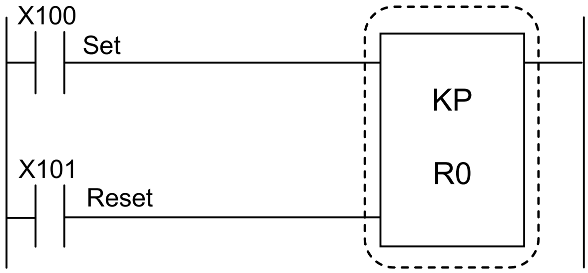
(2) Program operation for "↓OT" in ladder diagram

Outputs to the pulse relay P1 only for a single scan where X101 changes from the ON state to the OFF state (i.e., falls).



3.14 KP (Keep)

■ Ladder diagram



■ Devices that can be specified (indicated by ●)

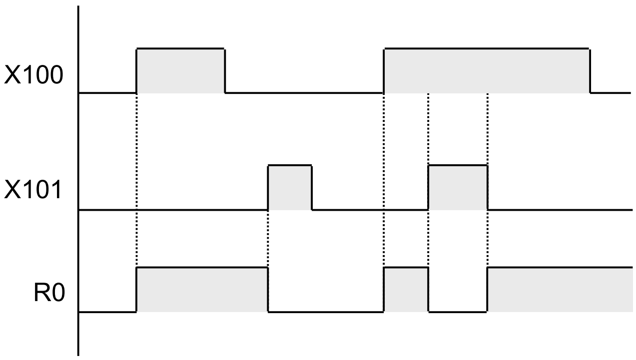
Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
bit	●	●	●	●							●	●	●	●

■ Outline of operation

- This instruction turns ON the specified coil output when the set input (X100) turns ON and holds the ON state. Release the output state when the reset input (X101) turns ON.
- While the state is being held, the output state is held regardless of the ON or OFF state of the set input (X100) until the reset input (X101) is entered.
- If the set input (X100) and the reset input (X101) turn ON simultaneously, the reset input (X101) will take precedence.

■ Operation Example

1) Turns ON the specified coil (R0) output when X100 turns ON and holds the ON state.
2) Release the output state when X101 turns ON.

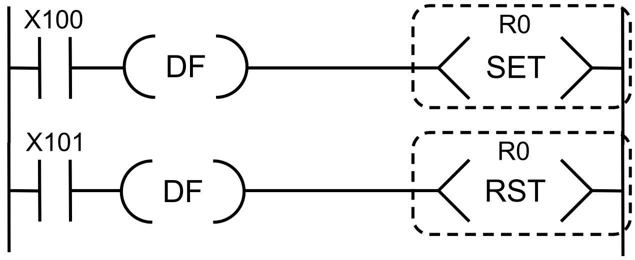


■ Precautions for programming

- The output destination holds the state even while the MC instruction is in operation.
- The state will be reset when the operation mode is switched from RUN to PROG. or the power is turned off. However, this is not the case when an internal relay set to the hold type is specified as the output destination.
- If an instruction is input following the output of the “KP” instruction, the conditions become the same as those for the reset input.

3.15 SET, RST (Set, Reset)

■ Ladder diagram



■ Devices that can be specified (indicated by •)

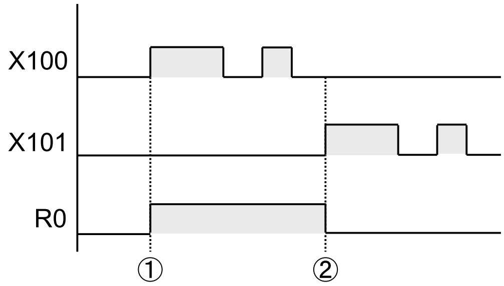
Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
bit	•	•	•	•				•			•	•	•	•

■ Outline of operation

Type of instruction	Operation
SET	When the execution condition turns ON, the output turns ON and the state is held regardless of a change in the state of the execution condition.
RST	When the execution condition turns ON, the output coil turns OFF and the OFF state is held regardless of a change in the state of the execution condition.

■ Operation Example

1) When X100 turns ON, R0 turns ON and its state is held.
2) When X101 turns ON, R1 turns OFF and its state is held.



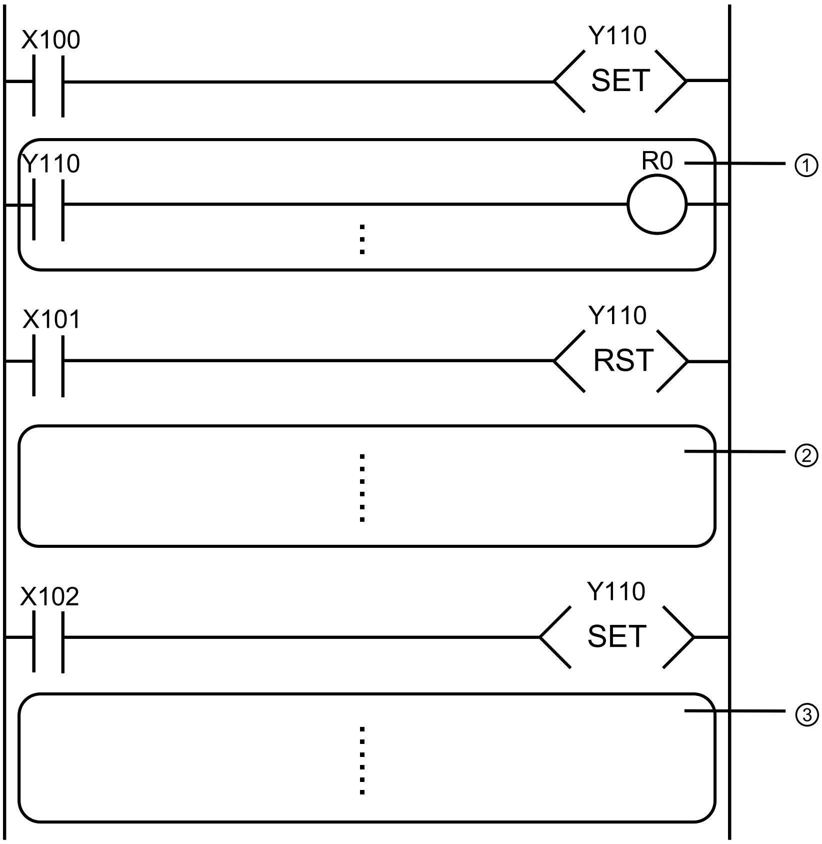
■ Precautions for programming

- The same output coil can be specified as the output destination of the SET and RST instructions for any number of times.

- Using relays with the SET and RST instructions does not result in duplicate output. Even if Total Check is implemented, it is not treated as a syntax error.
- The RST instruction can be used to turn off the relay.
- The output destination of the SET instruction holds the state even while the MC instruction is in operation.
- The output destination of the SET instruction will be reset when the operation mode is switched from RUN to PROG. or the power is turned off. However, this is not the case when an internal relay set to the hold type is specified as the output destination.
- A pulse relay (P) cannot be specified as an output destination of the SET and RST instructions.
- The error alarm buffer can be all cleared by RST SD60.
- The first entry of the error alarm buffer can be cleared by RST SD61.
- When “E” is selected for the operand, index modification is not possible.

■ Processing mechanism of SET and RST instructions

- The output content is overwritten with each step during processing of the operation.
- Since I/O refresh is done when the ED instruction is executed, data actually output depends on the final operation result.
- In order to output an operation result while processing is in progress, use the direct output (OT) instruction.



When X100 to X102 are all ON in the above program

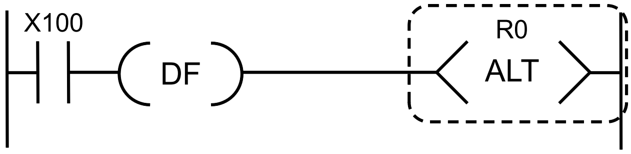
(1) Processing is done with Y110 to be ON.
(2) Processing is done with Y110 to be OFF.
(3) Processing is done with Y110 to be ON.

■ Use SET and RST instructions with differential instructions

- Putting the differential DF instructions before the SET and RST instructions makes it easy to create and adjust the program.
- This is particularly effective when the same output destination is used in several places in the program.

3.16 ALT (Alternate out)

■ Ladder diagram



■ Devices that can be specified (indicated by •)

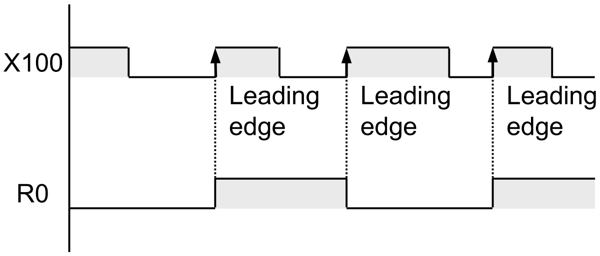
Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
bit	•	•	•	•							•	•	•	

■ Outline of operation

- When the operation result up to immediately before changes (rises) from OFF to ON, the specified coil ON/OFF is inverted.
- The ON/OFF state of the specified coil is held until the ALT instruction specifying the coil is executed next. (Flip-flop control)

■ Operation Example

ON/OFF states of output R0 is inverted whenever X100 changes from the OFF to ON states (rises).



■ Precautions for programming

- The ALT instruction detects rising from OFF to ON of the input and inverts the output.
- While the input continues to be ON, it is inverted only during rise. After that it is not inverted.
- The output will not be inverted for the first scan if the input is already ON when the operation mode is switched to RUN or the power is turned on in the RUN mode.
- Be careful when using this instruction with an instruction that changes the order of instruction execution such as MC - MCE or JP - LBL (1 to 6, shown below), because the operation of the instruction may change depending on the timing of the instruction execution and input.

  1) MC - MCE instructions

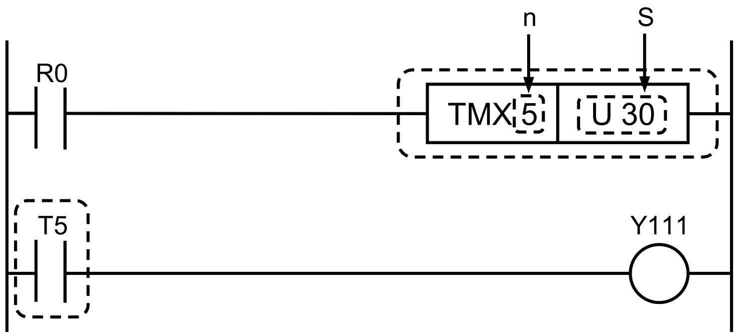
  2) JP - LBL instructions

  3) LOOP - LBL instructions

4) CNDE instruction
5) Step ladder instructions
6) Subroutine instructions

3.17 TM (Timer)

■ Ladder diagram



■ List of operands

Operand	Description
n	Timer number
S	Timer set value

■ Devices that can be specified (indicated by •)

Ope-rand	16-Bit device:											32-Bit device:			Integer			Real number		Str-ing	Index modifier (Note 2) (Note 3)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS (Note 1)	TE CE	IX	K	U	H	SF	DF	" "	
n																•					•
S	•	•	•	•			•	•				•				•					•

(Note 1) Only TS can be specified for TM instruction.

(Note 2) If you specify the index modifier for operand n (timer number), you cannot specify an integer constant for operand S (timer set value).

(Note 3) If specifying the index modifier for operand S (timer set value), you can specify only for 16-bit device or 32-bit device.

■ Outline of operation

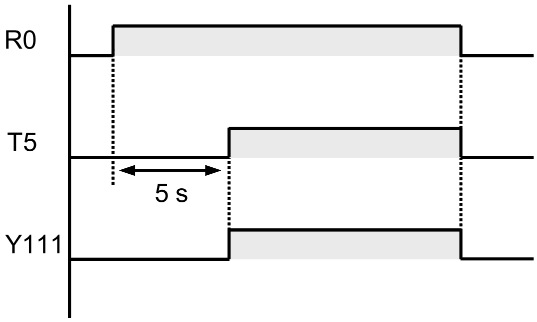
- The timer is the non-hold type that is reset when the power is turned off or the operation mode is switched from RUN to PROG.
- When the execution condition is ON, the timer starts decrementing from the set time [S]. When the elapsed value reaches 0, the timer contact [Tn] (n is the timer contact number) turns ON.
- When the execution condition turns OFF while the timer is decrementing, the timer stops and resets the elapsed value (clears it to zero).
- The OT instruction can also be written immediately after a timer coil.

Regarding the specification of timer time

- The timer set time is (timer unit) × (timer set value).
- The timer set value [S] is specified within the range between U1 to U4294967295, using a decimal constant.

"TMS" is specified within the range from 0.00001 to 42949.67295 seconds, in units of 0.00001 seconds.
"TML" is specified within the range from 0.001 to 4294967.295 seconds, in units of 0.001 seconds.
"TMR" is specified within the range from 0.01 to 42949672.95 seconds, in units of 0.01 seconds.
"TMX" is specified within the range from 0.1 to 429496729.5 seconds, in units of 0.1 seconds.
"TMY" is specified within the range from 1 to 4294967295 seconds, in units of 1 seconds.

Operation Example



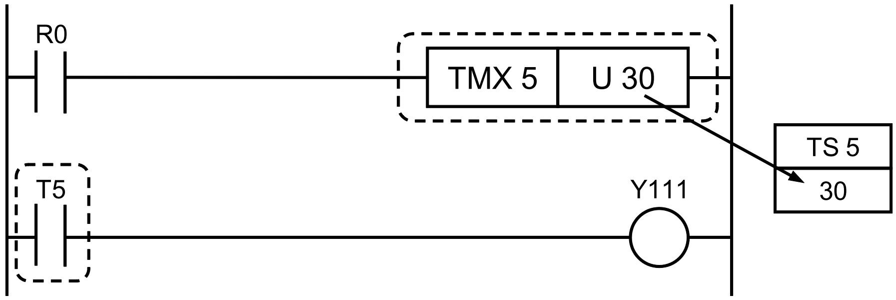
Precautions for programming

- The timer set value area TS and timer elapsed value area TE both occupy 32-bit areas. This is also true when a device such as DT is used in the operand [S]. Be careful not to overwrite the areas with another program.
- Since decrementing occurs during an operation, create the program so that decrementing occurs once during a single scan time. If multiple operations occur during a single scan due to an interruption handler program or jump/loop instruction, or decrementing never occurs, the correct result will not be obtained.
- When using a timer instruction with an AND stack (ANS) instruction or pop stack (POPS) instruction, be sure to write the code correctly.

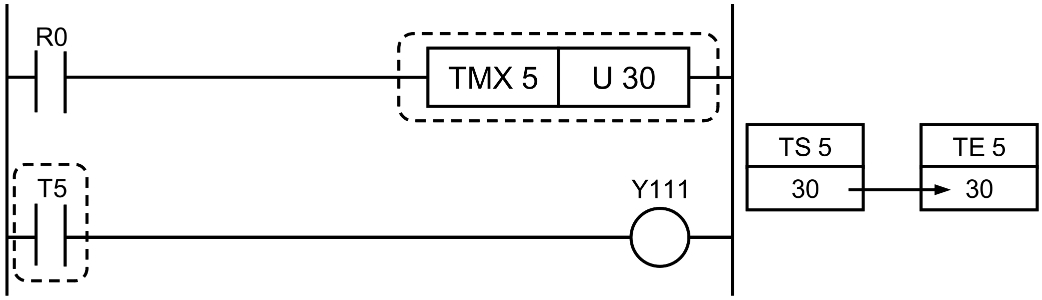
Timer operation mechanism

- This is an example when the U constant is used to specify the set value. See below for the operation when specifying the set value area number.

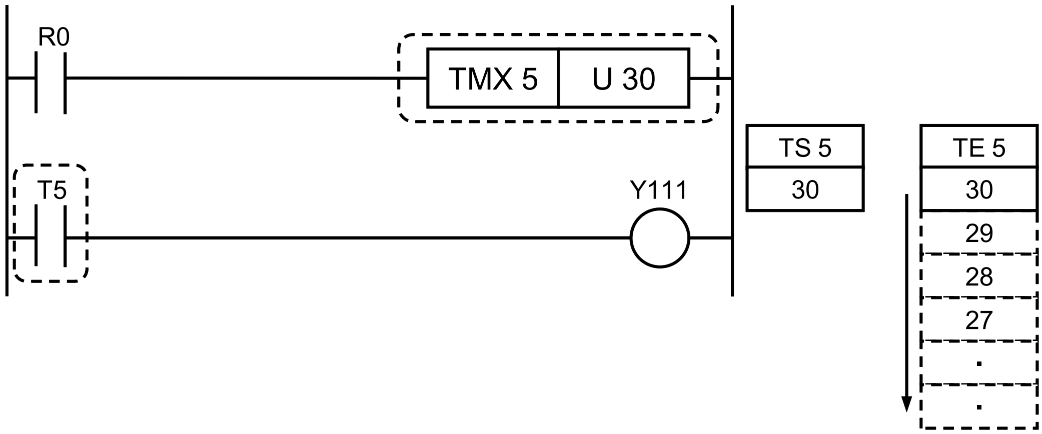
1) When the operation mode is switched to RUN or the power is turned ON in the RUN mode, the timer set value is transferred to the set value area "TS" with the same number.



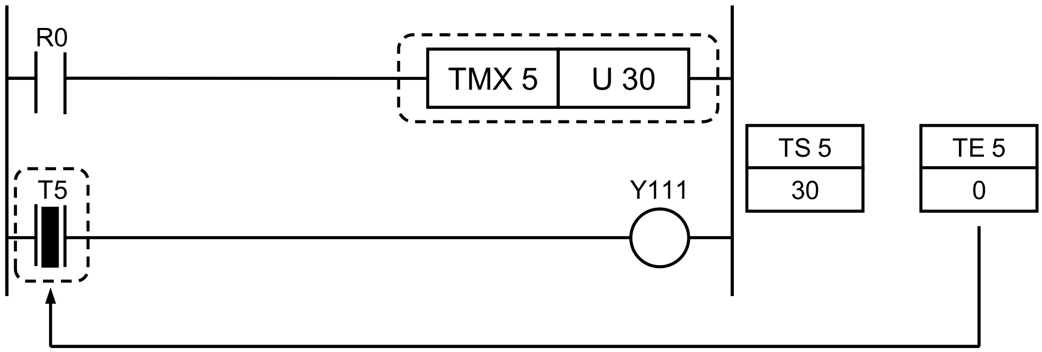
2) When the timer execution condition changes from OFF to ON (i.e., rises), the timer set value is transferred from the set value area "TS" to the elapsed value area "TE" with the same number. (This is also true when the operation mode is switched to RUN while the execution condition is ON.)



3) For each scan, the value in the elapsed value area "TE" decrements if the execution condition is ON.



4) When the value in the elapsed value area "TE" becomes 0, the timer contact "T" with the same number turns ON.

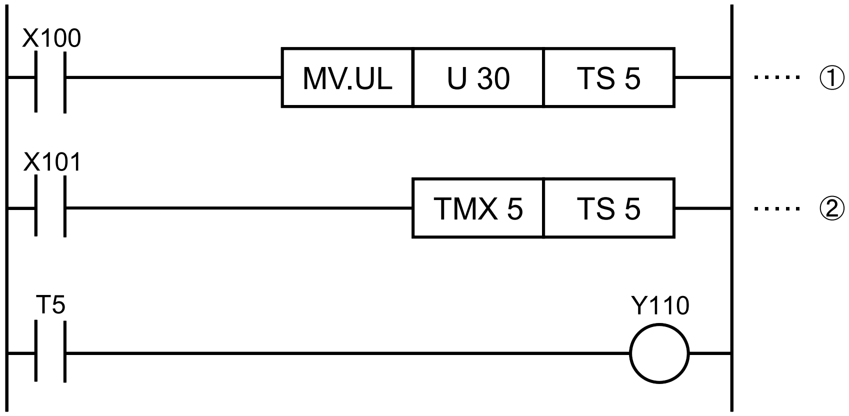


■ Regarding how to directly specify the set value area number to the timer set value

- The following program directly sets the timer set value to the timer set value area.
- Be sure to specify the same number as the timer number in [n] for the setting area "TS."
- The program described above, which specifies TS5 for the set value, operates as follows.

1) When the execution condition X100 is ON, the data transfer instruction MV is executed to start decrementing with U30 set to TS5.

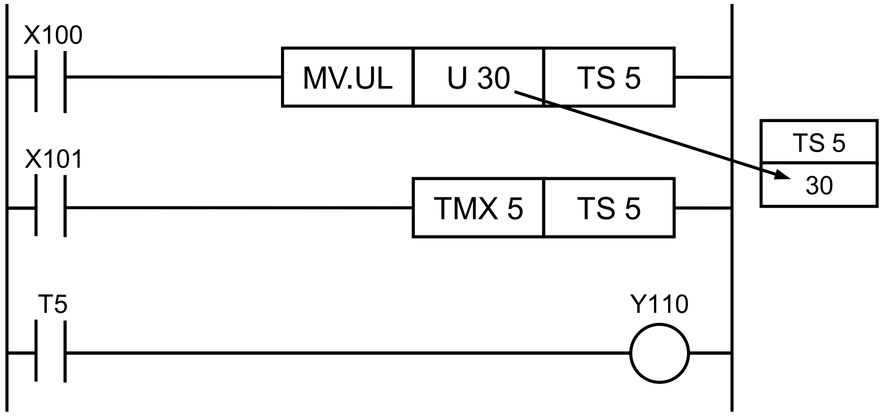
2) When the execution condition X101 turns ON, decrementing starts with 30 as the set value.



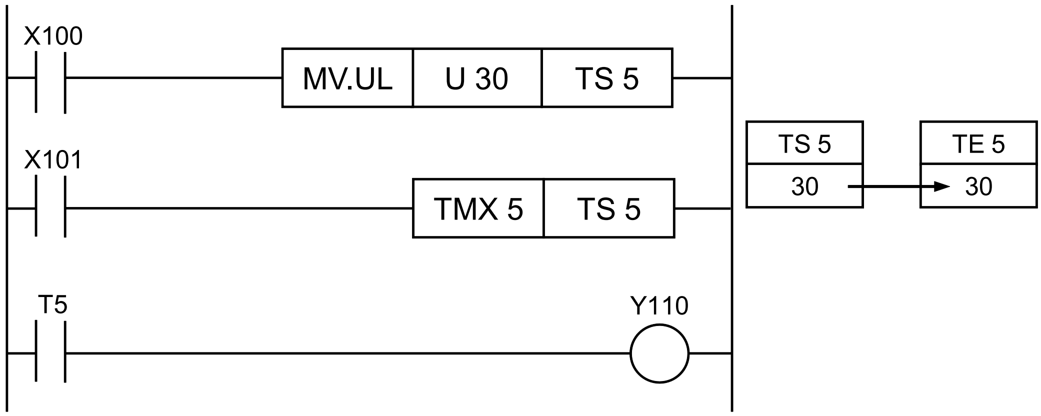
- Even when the value in the set value area "TS" is changed while decrementing, the decrement operation continues with the value before change.
- Timer operation starts with the changed value the next time the execution condition changes from OFF to ON after the decrement operation is completed or interrupted.
- The setting area "TS" is normally the non-hold type that is reset when the power is turned off or the operation mode is switched from RUN to PROG.

■ Mechanism of timer operation (When the set value area number is specified directly)

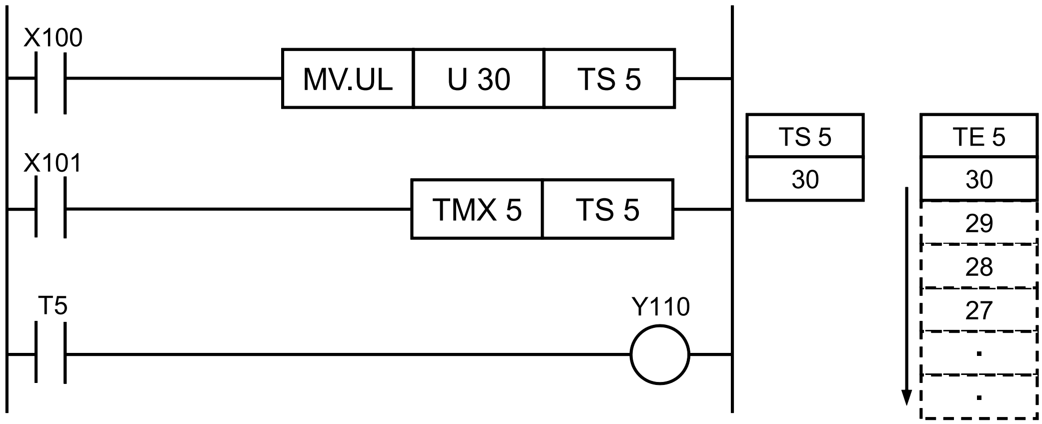
1) When the execution condition of the high-level instruction is ON, the value is set in the set value area "TS." The following shows an example of using the MV instruction.



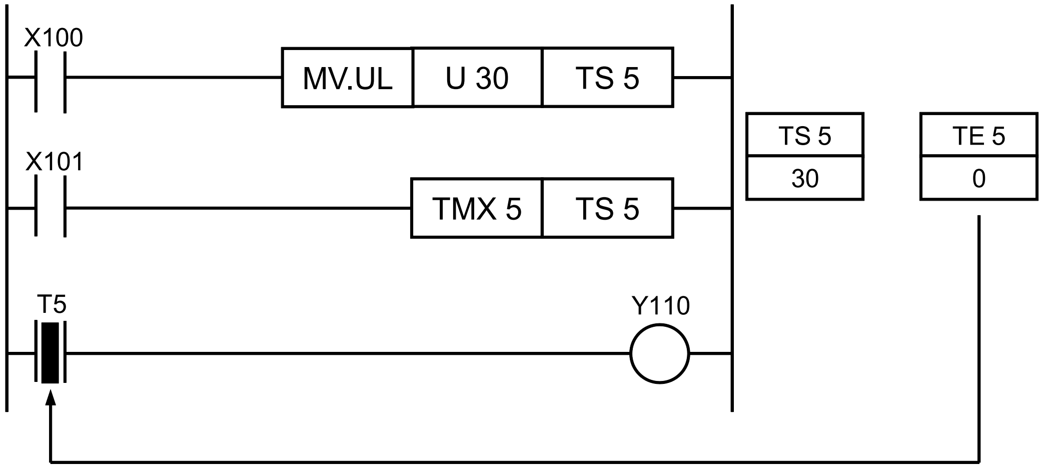
2) When the timer execution condition changes from OFF to ON (i.e., rises), the timer set value is transferred from the set value area "TS" to the elapsed value area "TE" with the same number. (This is also true when the operation mode is switched to RUN while the execution condition is ON.)



3) For each scan, the value in the elapsed value area "TE" decrements if the execution condition is ON.

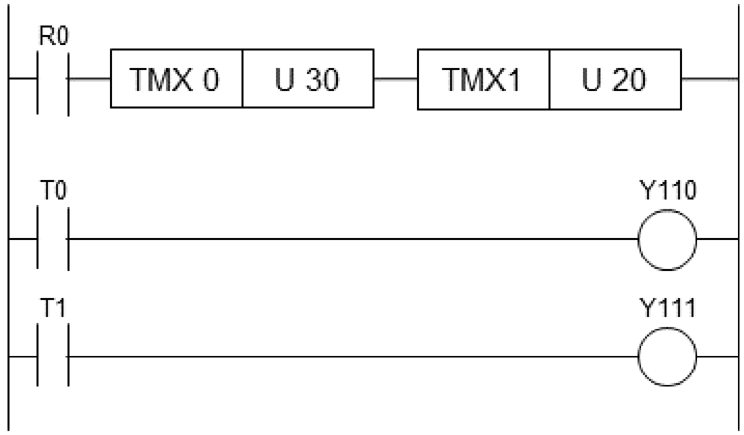


4) When the value in the elapsed value area "TE" becomes 0, the timer contact "T" with the same number turns ON.

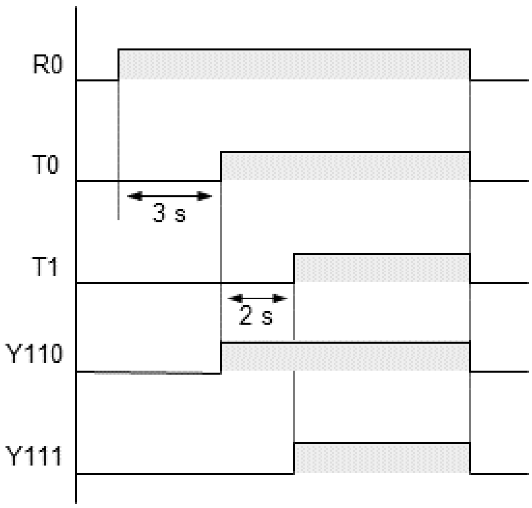


■ Application example of timer instructions (serial connection of timers)

● Ladder diagram

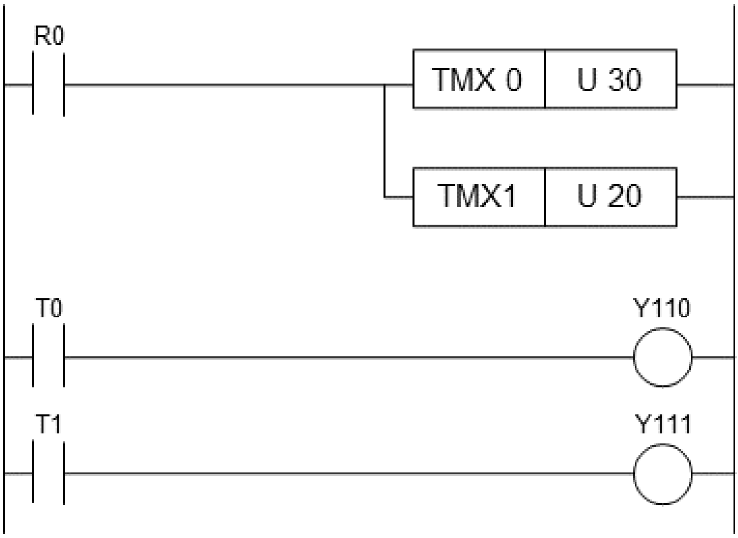


● Time chart

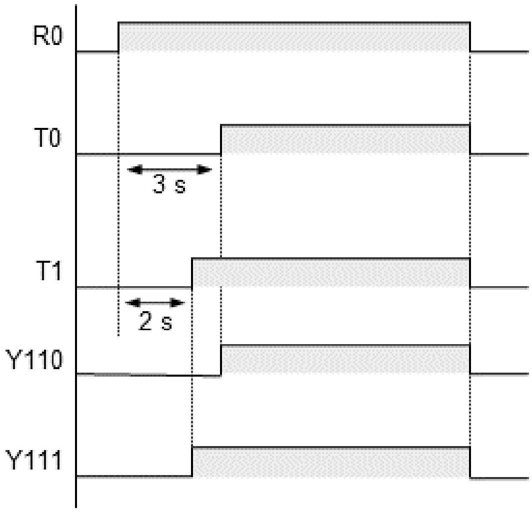


■ Application example of timer instructions (parallel connection of timers)

● Ladder diagram



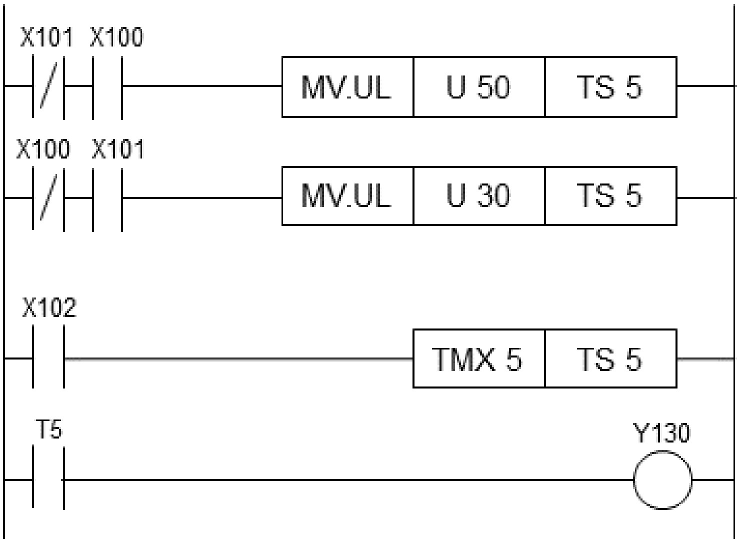
● Time chart



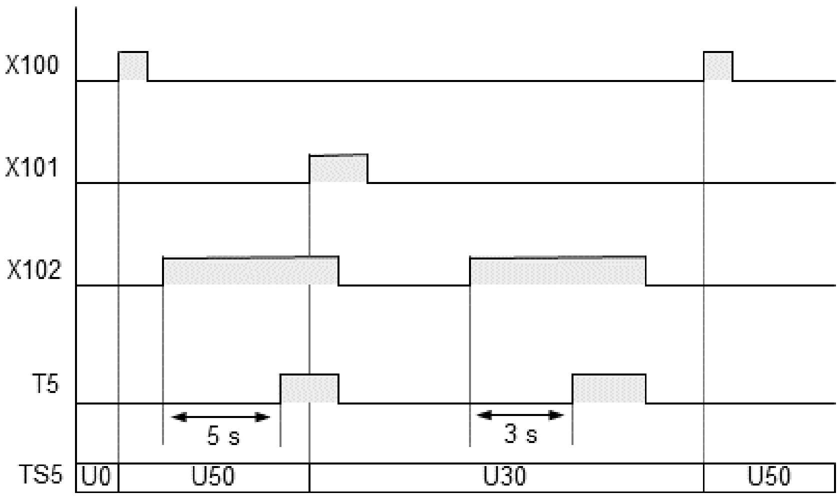
■ **Application example of timer instructions (When the set value area number is specified directly)**

<Example> Switching set values according to the condition

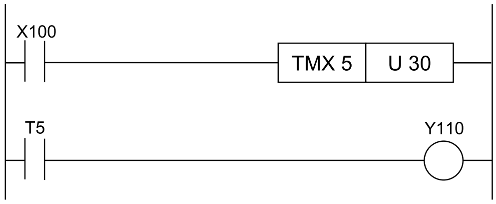
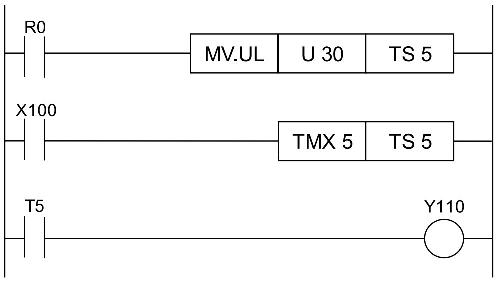
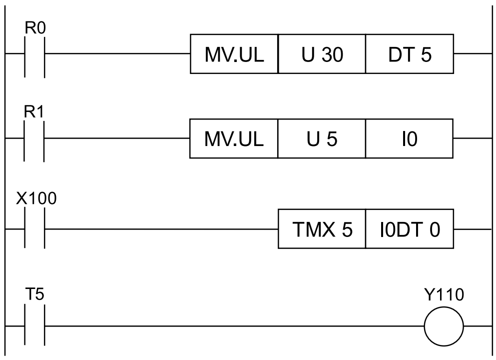
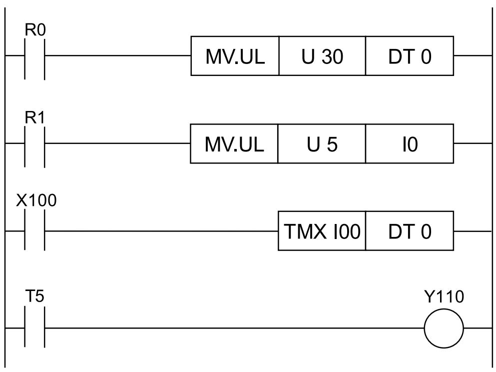
● Ladder diagram



● Time chart



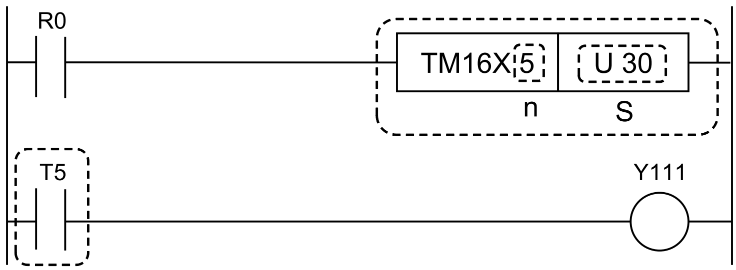
■ Timer number and timer setting value combinations

	Timer number	Timer set value	Ladder diagram	Description
1	Constant	Constant		Setting both the timer number and timer setting value with a constant
2	Constant	Device No.		Setting a constant for the timer number, and a device number for the timer setting value
3	Constant	Device number with index modification		Setting a constant for the timer number, and a device number with index modification for the timer setting value On FPWIN GR7, input the timer instruction in the following order: [TM/CT (F5)] [TMX (F1)][5][ENTER] →[INDEX (F9)][I0 (F1)] [DT (F5)][0][ENTER]
4	Timer number with index modification	Device No.		Setting a constant with index modification for the timer number, and a device number for the timer setting value On FPWIN GR7, input the timer instruction in the following order: [TM/CT (F5)] [TMX (F1)] [INDEX (F9)] [I0 (F1)][0][ENTER] →[DT (F5)][0][ENTER]

(Note 1) Refer to "I0 to IE Index Register" in"2.6 Description of the memory area".

3.18 TM16 (16-bit Timer)

■ Ladder diagram



■ List of operands

Operand	Description
n	Timer number (Available range: 0 to 4095 *For the default memory configuration)
S	Timer setting value (Available range: U0 to U65535)

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	(Note 1) (Note 2)
n																●					●
S	●	●	●	●			●	●								●					●

(Note 1) If you specify the index modifier for operand n (timer number), you cannot specify an integer constant for operand S (timer set value).

(Note 2) If specifying the index modifier for operand S (timer set value), you can specify only for 16-bit device or 32-bit device.

■ Outline of operation

- Unlike the TM instruction, the range available for the setting value [S] for this instruction is U0 to U65535.
- The timer is the non-hold type that is reset when the power is turned off or the operation mode is switched from RUN to PROG.
- When the execution condition is ON, the timer starts decrementing from the set time S. When the elapsed value reaches U0, the timer contact Tn (n is the timer contact number) turns ON.
- When the execution condition turns OFF during the subtraction operation, the elapsed value is reset (cleared to zero).
- The OT instruction can also be written immediately after a timer coil.
- The timer set time is (timer unit) × (timer set value).

"TM16S" is specified within the range from 0.00000 to 0.65535 seconds, in units of 0.00001 seconds.

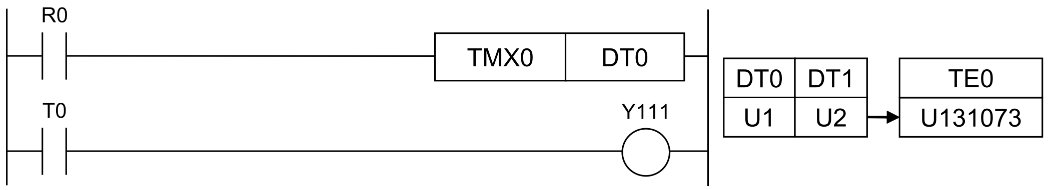
"TM16L" is specified within the range from 0.000 to 65.535 seconds, in units of 0.001 seconds.
"TM16R" is specified within the range from 0.00 to 655.35 seconds, in units of 0.01 seconds.
"TM16X" is specified within the range from 0.0 to 6553.5 seconds, in units of 0.1 seconds.
"TM16Y" is specified within the range from 0 to 65535 seconds, in units of 1 seconds.

■ Precautions for programming

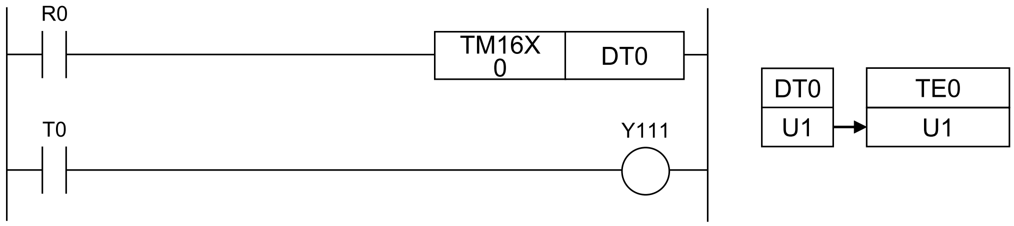
- When specifying a 16-bit device such as DT for the operand S of TM16 instruction, the area used is for 16-bit data.
- Since decrementing occurs during an operation, create the program so that decrementing occurs once during a single scan time. If multiple operations occur during a single scan due to an interruption handler program or jump/loop instruction, or decrementing never occurs, the correct result will not be obtained.
- When U0 is specified for the setting value, the timeout operation is performed when the instruction is executed and the timer contact T turns ON.
- When using a timer instruction with an AND stack (ANS) instruction or pop stack (POPS) instruction, be sure to write the code correctly. (For details, refer to "19.8 Precautions for Programming".)

■ Difference between TM and TM16 instructions

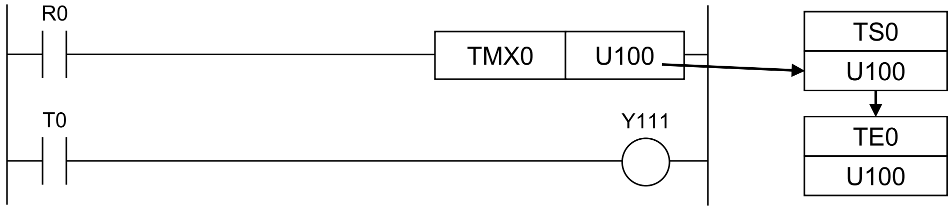
TM instruction (When specifying operand S=16-bit device)
32-bit data U131073 (H20001) written in DT0 to DT1 is treated as the timer setting value. U131073 is set in the elapsed value area TE0 when the input of R0 rises.



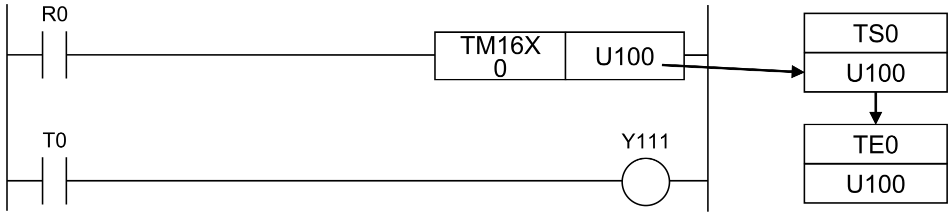
TM16 instruction (When specifying operand S=16-bit device)
16-bit data U1 (H1) written in DT0 is treated as the timer setting value. U1 is set in the elapsed value area TE0 when the input of R0 rises.



TM instruction (When specifying operand S=constant)
Constant U100 is set in the setting value area TS0 when compiling the program. The setting value area TS0 is set in the elapsed value area TE0 when the input of R0 rises. The setting range of a constant is U0 to U4294967295.



TM16 instruction (When specifying operand S=constant)
U100 is set in the setting value area TS0 when compiling the program. The setting value area TS0 is set in the elapsed value area TE0 when the input of R0 rises. The setting range of a constant is U0 to U65535.

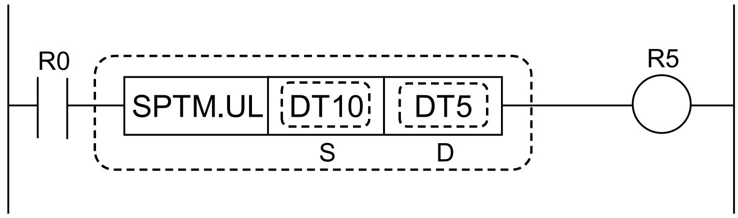


Note

- When a 32-bit value is written into the elapsed value area TE while the timer is being operated using an instruction such as MV instruction, the timer operates with the written 32-bit value. When a 32-bit value is written into the setting value area TS, the timer operates with the written 32-bit value.

3.19 SPTM (Unsigned 32-bit Incremental Auxiliary Timer)

■ Ladder diagram



■ List of operands

Operand	Description
S	Timer set value
D	Timer elapsed value

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier (Note 3)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS (Note 1)	TE CE (Note 2)	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●				●				●	●				●
D	●	●	●	●			●	●					●								●

(Note 1) CS cannot be specified in the first operand [S].

(Note 2) CE cannot be specified in the second operand [D].

(Note 3) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

- This instruction operates as a 32-bit on-delay timer with 0.01 second resolution.
- When the execution condition is ON, the timer increments in the area [D, D+1] where the elapsed time is specified.
- When the elapsed value reaches the timer set value [S], the relay that is connected to the Timer instruction and the system relay SRD are turned ON. (These relays are OFF when the execution condition is OFF or when the timer is incrementing.)
- The system relay SRD can also be used as a timer contact.

■ Setting the timer period

- The timer period is 0.01 × [timer set value].

- The timer set value is specified within the range from U1 (H1) to U4294967295 (HFFFFFFFF), using a U constant.

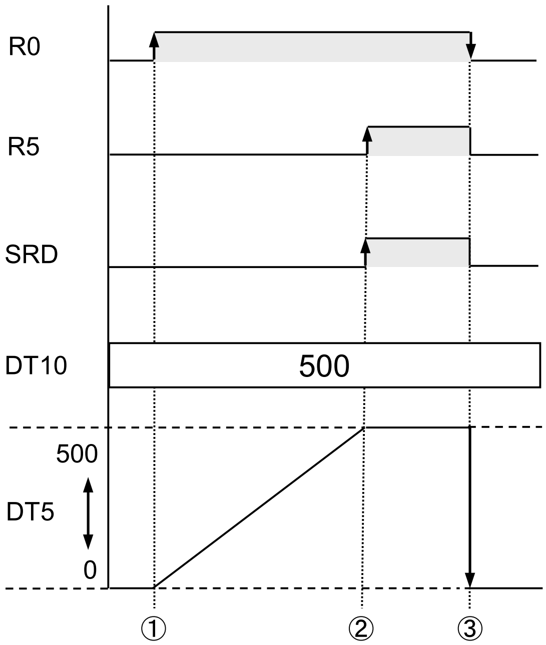
"SPTM" is specified within the range from 0.01 to 42949672.95 seconds, in units of 0.01 seconds.

Example) When the set value is U500, the set time is 0.01 x 500 = 5 seconds.

■ Operation Example

When executed with U500 in the set value [S]: DT10

1) Timer operation starts when R0 turns ON.
   "0" is transferred to the elapsed value area: DT5.
2) When the value of the elapsed value area: DT5 reaches the value in the set value area: DT10 (U500), the system relay SRD and output coil R5 turn ON.
3) When R0 turns OFF, the timer operation stops and "0" is transferred to the elapsed value area: DT5.

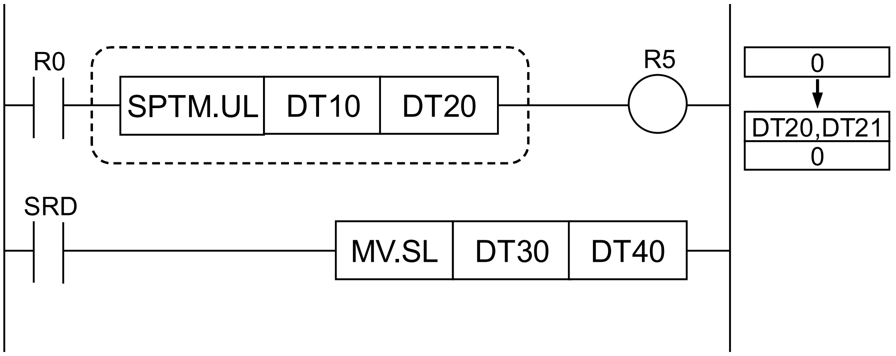


■ Precautions for programming

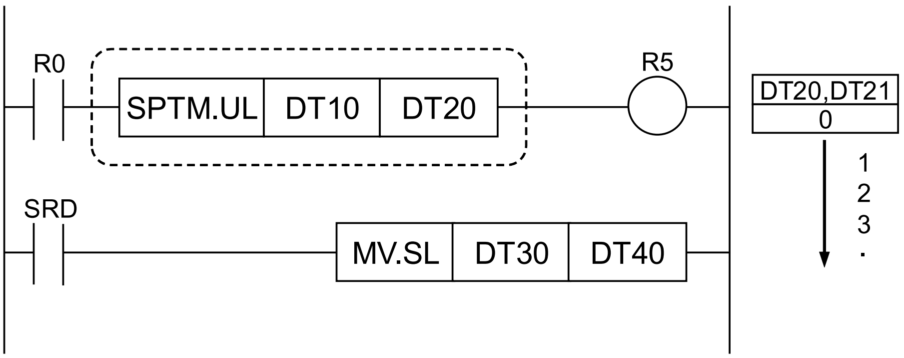
- Ensure that the specifications of the area storing the set value and the process value area do not overlap with other timer/counter instructions or operation memory areas of high-level instructions.
- Since incrementing occurs during an operation, create the program so that incrementing occurs once during a single scan time. If multiple operations occur during a single scan due to an interrupt handler program or jump/loop instruction, or incrementing never occurs, the correct result will not be obtained.

■ **How the auxiliary timer works**

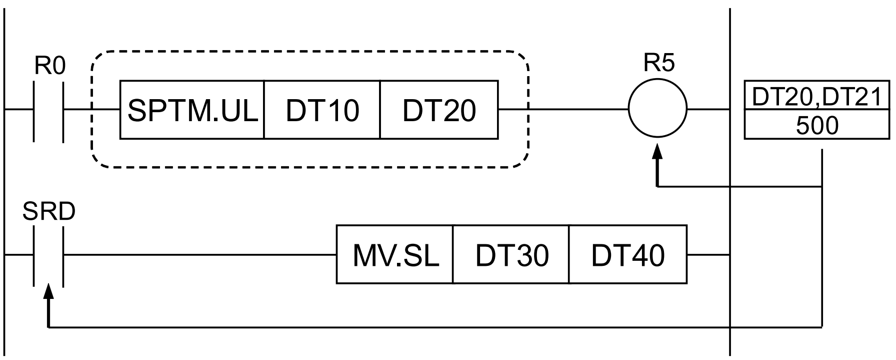
1) When the internal relay changes from OFF to ON, "0" is transferred to the elapsed value area [D,D+1].



2) For each scan, the value in the elapsed value area [D, D+1] increments if the internal relay is ON.



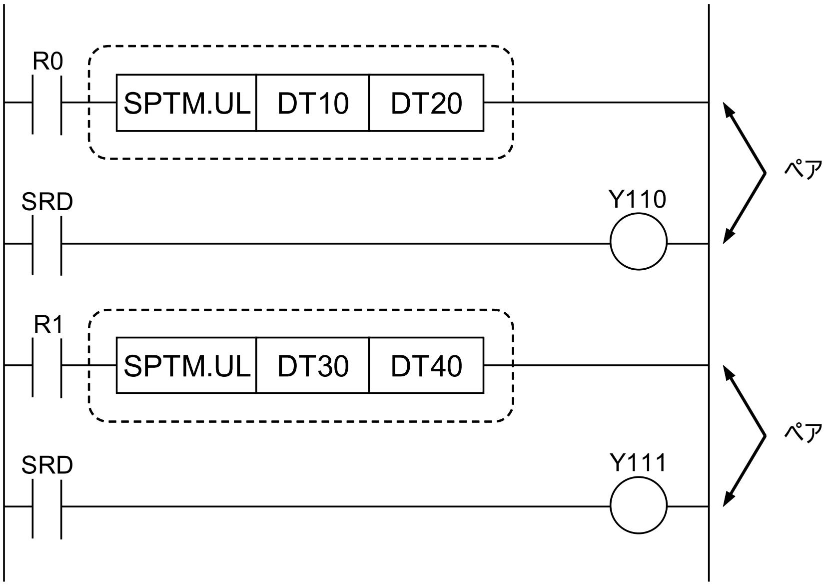
3) If the value for the elapsed value area [D, D+1] is equal to the value for [S,S+1], the relay used for the OT instruction and system relay SRD are ON.



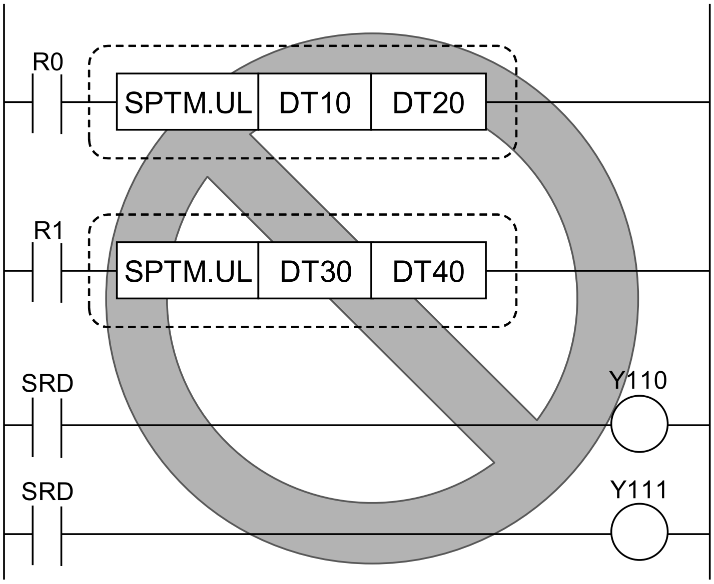
■ **Precaution when using system relay SRD**

- When using multiple auxiliary timers with the SRD, be sure use SRD on the line following the auxiliary timer instruction.

<Example>

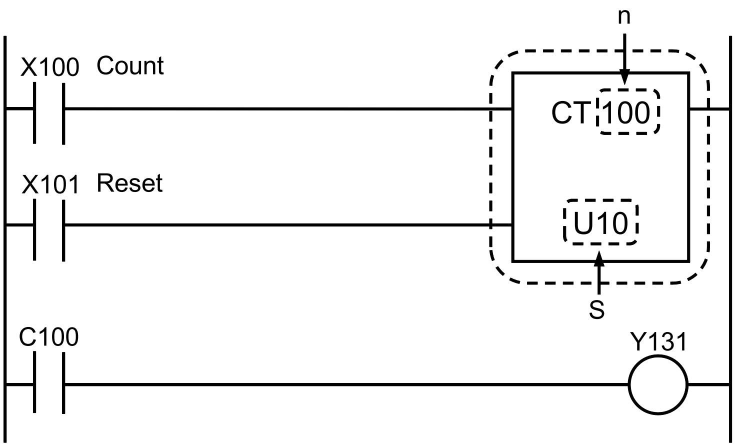


● A correct operation cannot be obtained if specified as shown below.



3.20 CT (Down Counter)

■ Ladder diagram



■ List of operands

Operand	Description
n	Counter number
S	Counter set value

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 2) (Note 3)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS (Note 1)	TE CE	IX	K	U	H	SF	DF	" "	
n																●					●
S	●	●	●	●			●	●				●				●					●

(Note 1) Only CS can be specified for CT instruction.

(Note 2) If you specify the index modifier for operand n (counter number), you cannot specify an integer constant for operand S (counter set value).

(Note 3) If specifying the index modifier for operand S (counter set value), you can specify only for 16-bit device or 32-bit device.

■ Outline of operation

- All counters are decremental preset counters.

- When the reset input rises from OFF to ON, the value in the set value area "CS" is preset to the elapsed value area "CE."
- When the reset input is ON, the elapsed value is reset to 0.
- When the count input changes from OFF to ON, the counter decrements from the set value. When the elapsed value becomes 0, output to the counter contact is generated.
- If the count input and reset input both turn ON at the same time, the reset input is given priority.
- If the count input rises and the reset input falls at the same time, the count input is ignored and preset is executed.
- An OT instruction can be entered immediately after a counter instruction.

■ Setting the count value

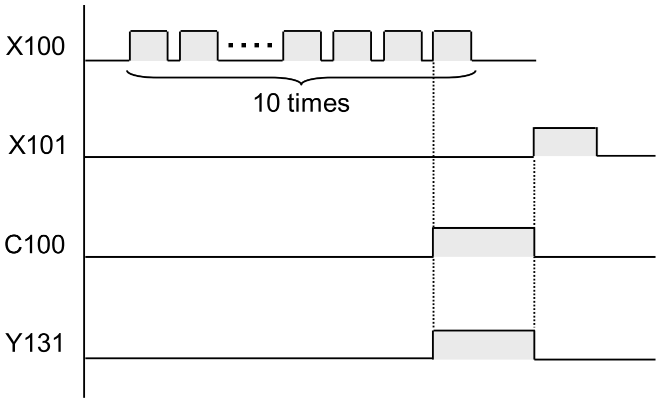
- The count value is specified within the setting range from U1 to 4294967295, using a decimal (U) constant.

■ Operation Example

When counter number [n]=100, set value [S]=10

1) When X100 turns ON for 10 times, C100 turns ON and Y131 turns ON.

2) When X101 turns ON, the elapsed value is reset.



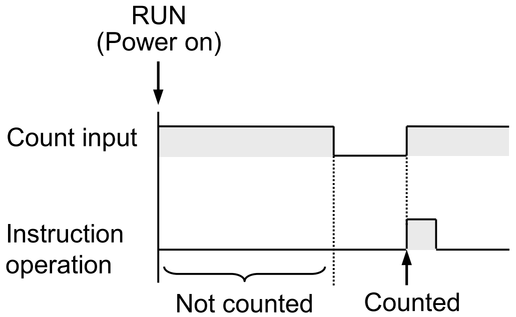
■ Precautions for programming

- The counter set value area CS and counter elapsed value area CE both occupy 32-bit areas. This is also true when a device such as DT is used in the operand [S]. Be careful not to overwrite the areas with another program.
- When combining a counter instruction with an AND stack instruction or POP stack instruction, be careful that the programming is correct.

■ Cautions on detecting the count input

- In a counter instruction, the subtraction takes place when the rise of the count input from OFF to ON is detected.
- Counting is only performed at the rise, so even if the count input remains on, no further counting will occur.

- The set value will not be decremented for the first scan if the count input is already ON when the operation mode is switched to RUN or the power is turned on in the RUN mode.



- Be careful when using this instruction with an instruction that changes the order of instruction execution such as MC - MCE or JP - LBL (1 to 6, shown below), because the operation of the instruction may change depending on the timing of the instruction execution and count input.

  1) MC - MCE instructions

  2) JP - LBL instructions

  3) LOOP - LBL instructions

  4) CNDE instruction

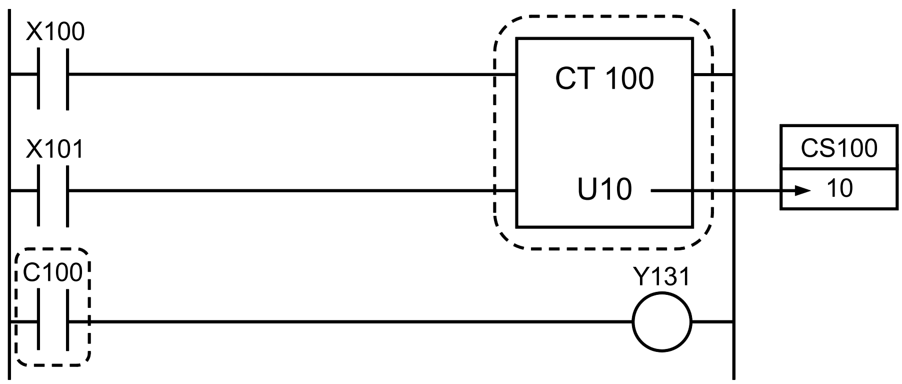
  5) Step ladder instructions

  6) Subroutine instructions

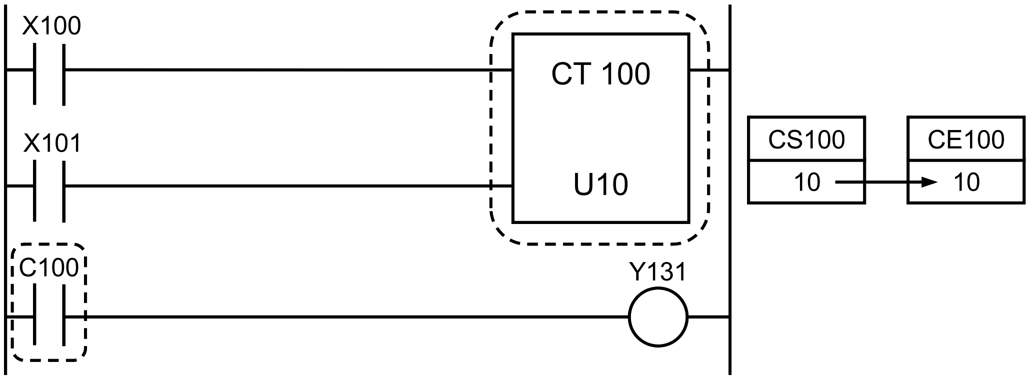
■ Mechanism of down counter operation

- This is an example when the U constant is used to specify the set value. See below for the operation when specifying the set value area number. (This example shows a case in which “100” is specified for the counter.)

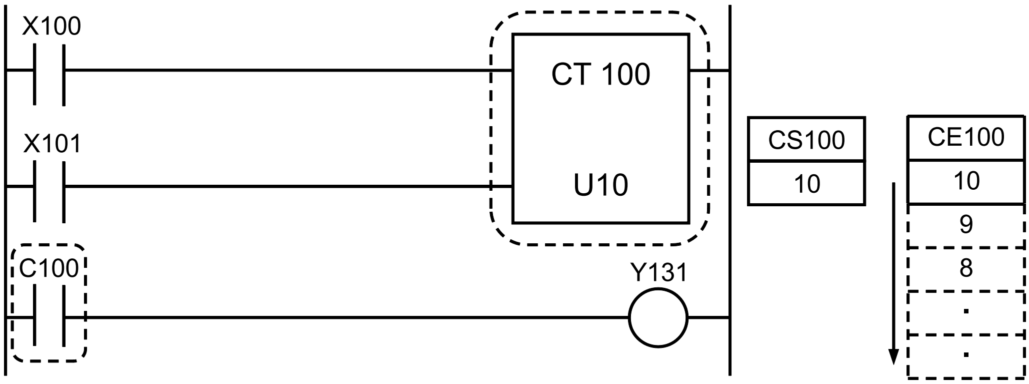
1) When the operation mode is switched to RUN or the power is turned ON in the RUN mode, the counter set value is transferred to the set value area "CS" with the same number.



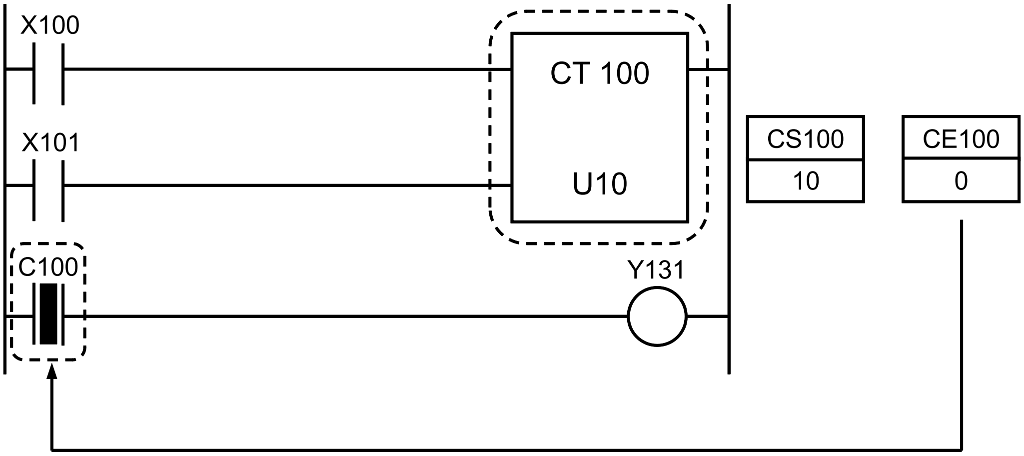
2) At the trailing edge of the reset input, the value in the set value area "CS" is preset to the elapsed value area "CE."



3) Whenever the count input X100 turns ON, the value in the elapsed value area "CE" is decremented.

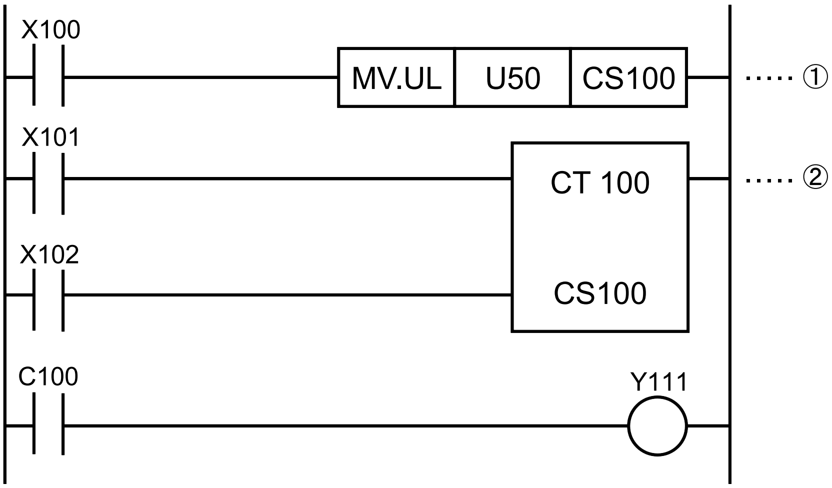


4) When the value in the elapsed value area "CE" becomes 0, the counter contact "C" with the same number turns ON.



■ How to specify the set value area number directly to the down counter setting value

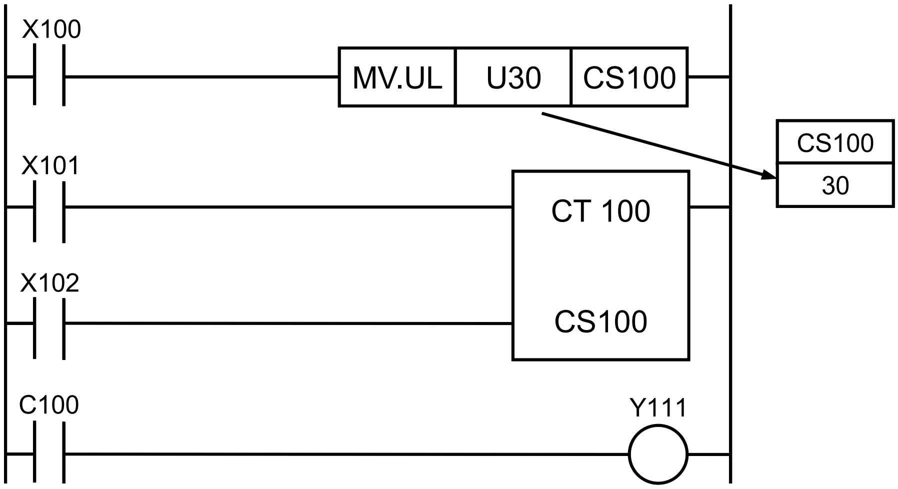
- The program described above, which specifies CS100 for the set value, operates as follows.
  (1) When the execution condition X100 is ON, the data transfer instruction MV is executed to start decrementing with U30 set to CS100.
  (2) When the count input X101 is ON, decrementing starts from the set value 30.
- Be sure to specify the same address as the counter number in [S] for the setting area "CS."



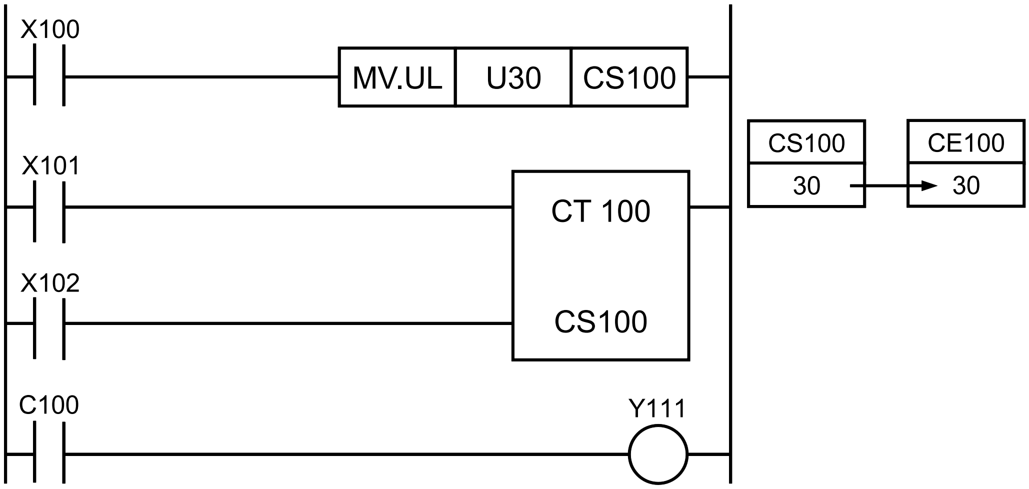
- Even when the value in the set value area "CS" is changed while decrementing, the decrement operation continues with the value before change. The new value will be used for the counter operation after the counter is reset and the counter input next changes from OFF to ON.

■ **Mechanism of down counter operation (When the set value area number is specified directly)**

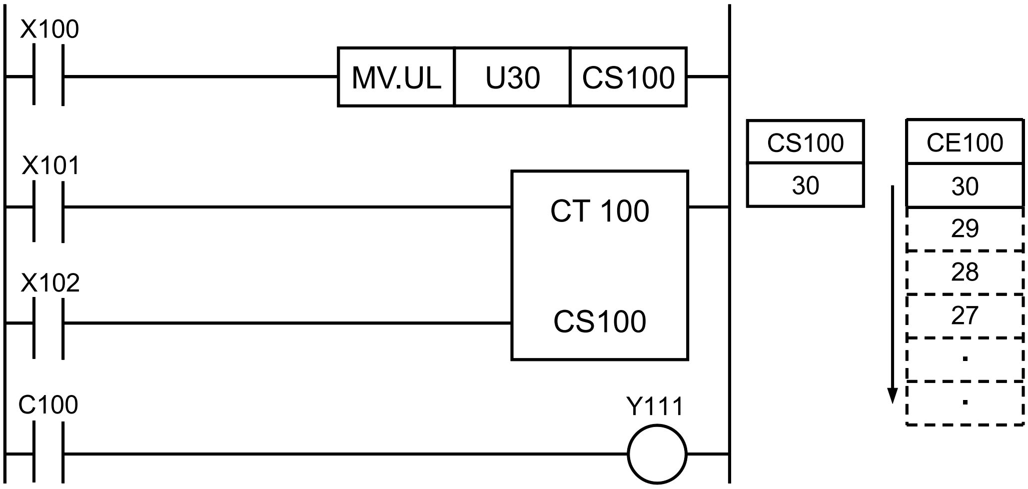
1) When the execution condition of the high-level instruction is ON, the value is set in the set value area "CS." The following shows an example of using the MV instruction.



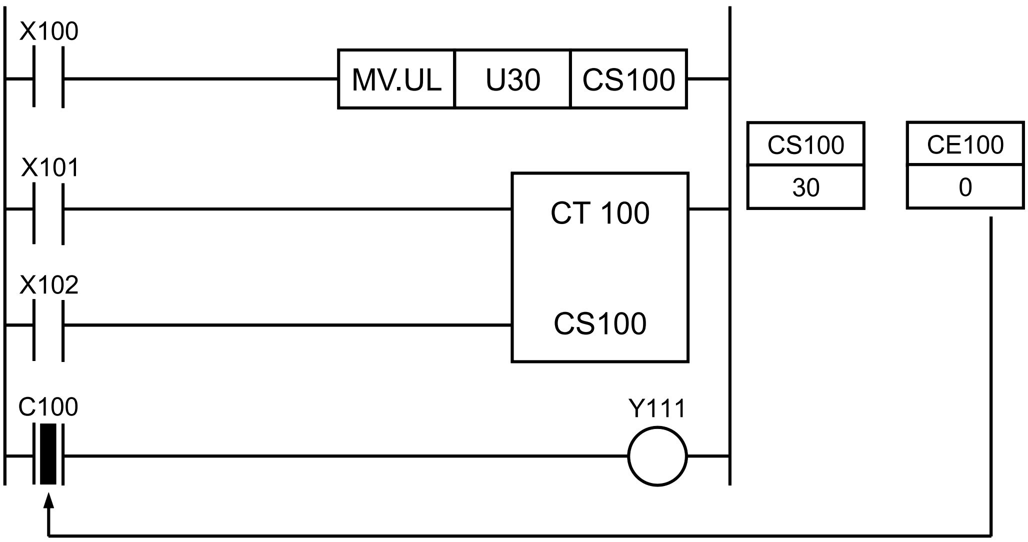
2) At the trailing edge of the reset input, the value in the set value area "CS" is preset to the elapsed value area "CE."



3) Whenever the count input X101 turns ON, the value in the elapsed value area "CE" is decremented.



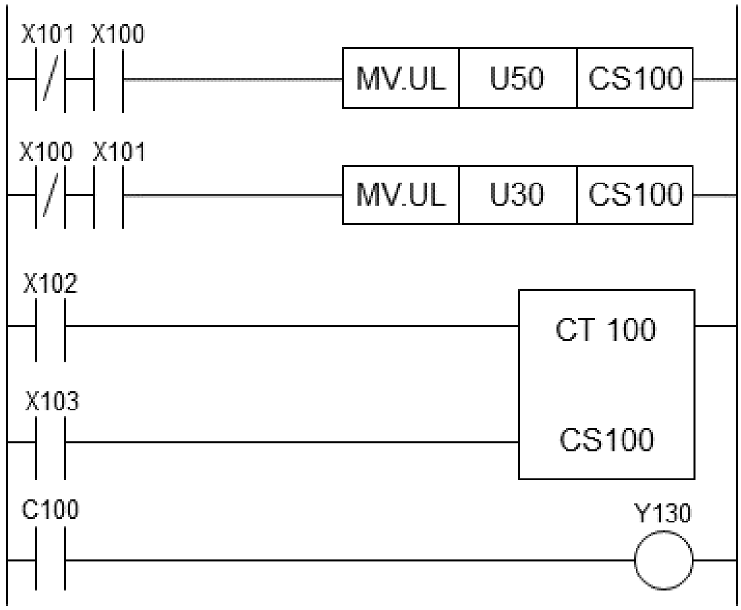
4) When the value in the elapsed value area "CE" becomes 0, the counter contact "C" with the same number turns ON.



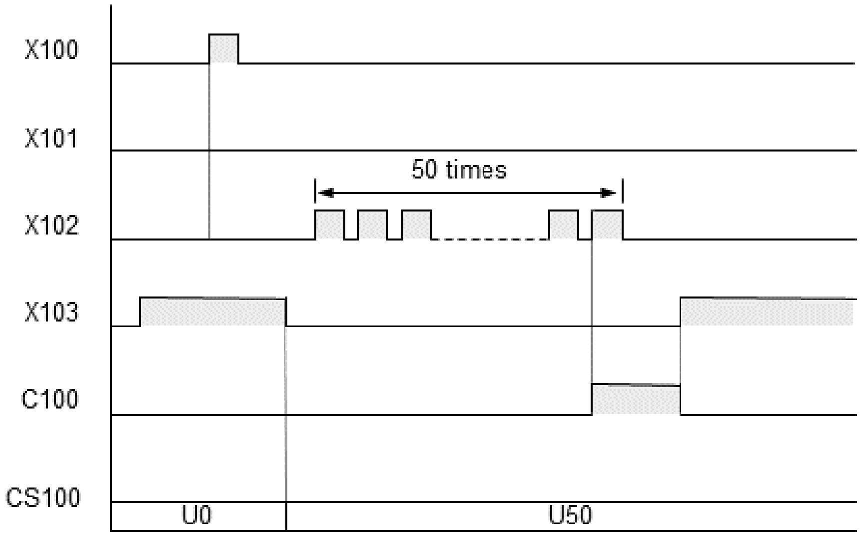
■ **Application example of counter instructions (When the set value area number is specified directly)**

<Example> Switching set values according to the condition

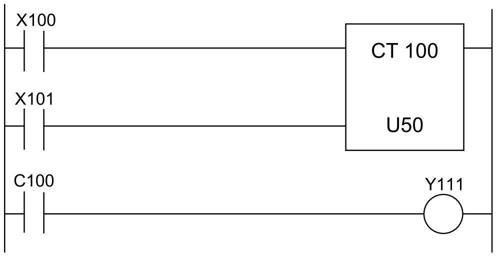
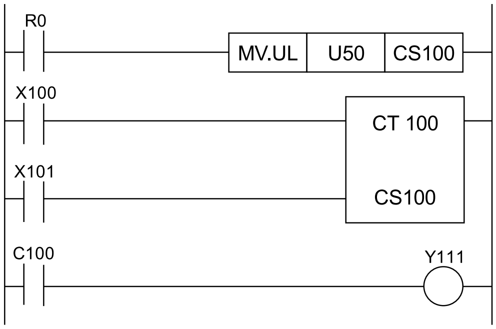
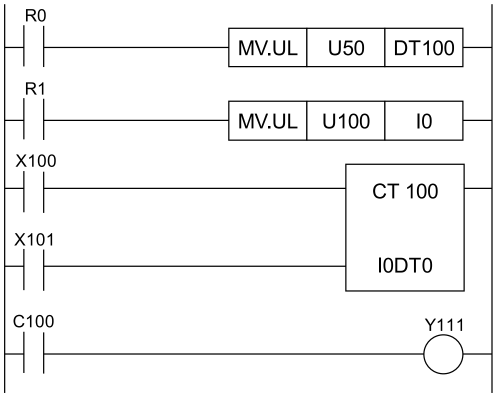
● Ladder diagram

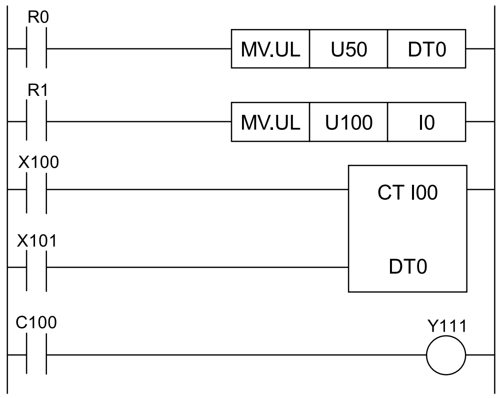


● Time chart



■ Counter number and setting value combinations

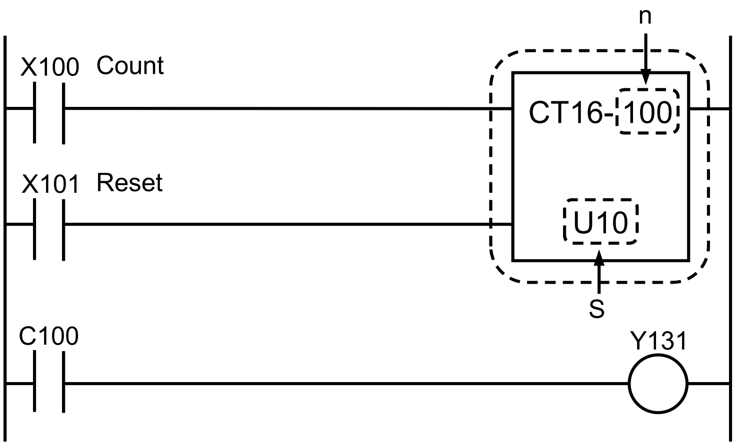
<table>
<tr><th></th><th>Counter number</th><th>Counter set value</th><th>Ladder diagram</th><th>Description</th></tr>
<tr><td>1</td><td>Constant</td><td>Constant</td><td>
</td><td>Setting a constant for both the counter number and the counter setting value</td></tr>
<tr><td>2</td><td>Constant</td><td>Device No.</td><td>
</td><td>Setting a constant for the counter number, and a device number for the counter setting value</td></tr>
<tr><td>3</td><td>Constant</td><td>Device number with Index modification</td><td>
</td><td>Setting a constant for the counter number, and a device number with index modification for the counter setting value
On FPWIN GR7, input the counter instruction in the following order:
[TM/CT (F5)]
[CT (F6)]
[1][0][0][ENTER]→
[INDEX (F9)][I0 (F1)]
[DT (F5)][0][ENTER]</td></tr>
</table>

	Counter number	**Counter set value**	**Ladder diagram**	**Description**
4	Counter number with Index modification	Device No.		Setting a constant with index modification for the counter number, and a device number for the counter setting value On FPWIN GR7, input the counter instruction in the following order: [TM/CT (F5)] [CT (F6)] [INDEX (F9)][I0 (F1)] [0][ENTER]→ [DT (F5)][0][ENTER]

(Note 1) Refer to"2.6.11 I0 to IE Index registers".

3.21 CT16 (16-bit Counter)

■ Ladder diagram



■ List of operands

Operand	Description
n	Counter number (Available range: 4 to 1023 *For the default memory configuration)
S	Counter setting value (Available range: U0 to U65535)E

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	(Note 1) (Note 2)
n																•					•
S	•	•	•	•			•	•								•					•

(Note 1) If you specify the index modifier for operand n (counter number), you cannot specify an integer constant for operand S (counter set value).

(Note 2) If specifying the index modifier for operand S (counter set value), you can specify only for 16-bit device or 32-bit device.

■ Outline of operation

- Unlike the CT instruction, the range available for the setting value [S] for this instruction is U0 to U65535.
- All counters are decremental preset counters.
- When the reset input falls from OFF to ON, the value in the setting value area CS is preset to the elapsed value area CE.

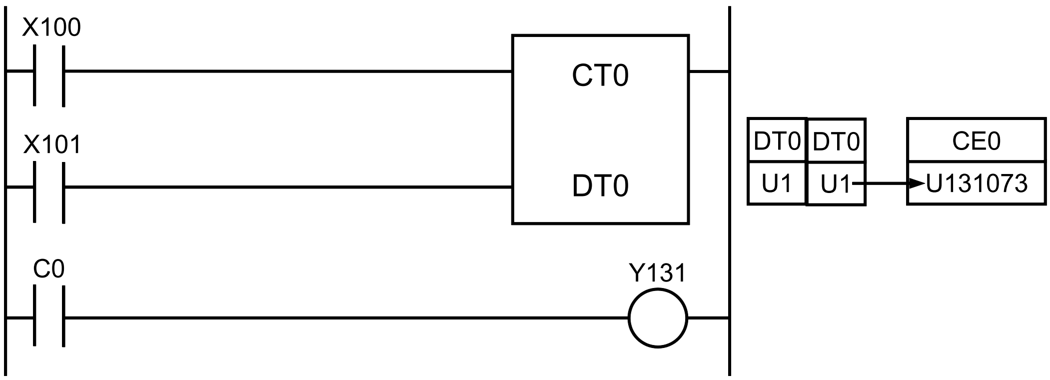
- When the reset input is ON, the elapsed value is reset (cleared to zero). When the count input changes from the OFF state to the ON state, the setting value is subtracted. When the elapsed value becomes U0, it is output to the counter contact.
- If the count input and reset input both turn ON at the same time, the reset input is given priority.
- If the count input rises and the reset input falls at the same time, the count input is ignored and preset is executed.
- An OT instruction can be entered immediately after a counter instruction.

■ Precautions for programming

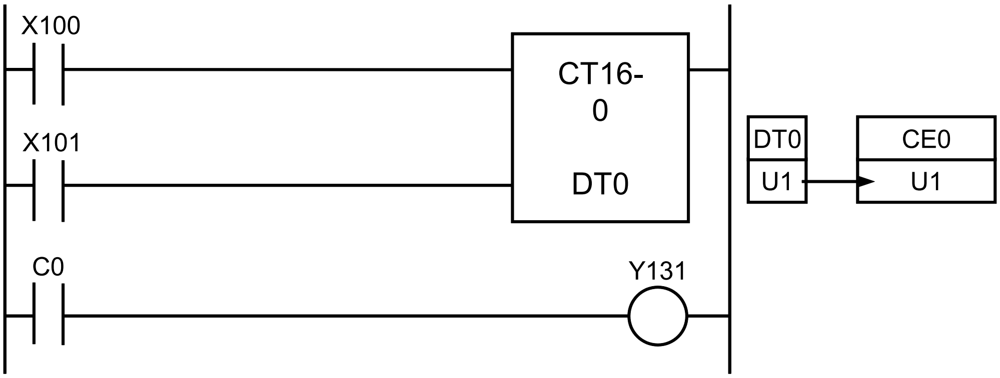
- The counter set value area CS and counter elapsed value area CE both occupy 32-bit areas.
- When specifying a 16-bit device such as DT for the operand S of CT16 instruction, the area used is for 16-bit data.
- When U0 is specified for the setting value, the count-up operation is performed when the instruction is executed and the counter contact C turns ON.

■ Difference between CT and CT16 instructions

CT instruction (When specifying operand S=16-bit device)
32-bit data U131073 (H20001) written in DT0 to DT1 is treated as the counter setting value. U131073 is set in the elapsed value area CE0 when the input of X101 falls.



CT16 instruction (When specifying operand S=16-bit device)
16-bit data U1 (H1) written in DT0 is treated as the counter setting value. U1 is set in the elapsed value area CE0 when the input of X101 falls.

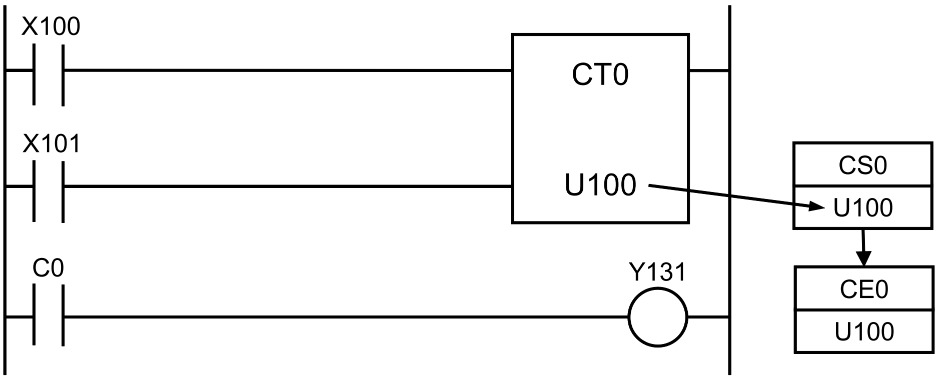


Note

- When a 32-bit value is written into the elapsed value area CE while the counter is being operated using an instruction such as MV instruction, the timer operates with the written 32-bit value.

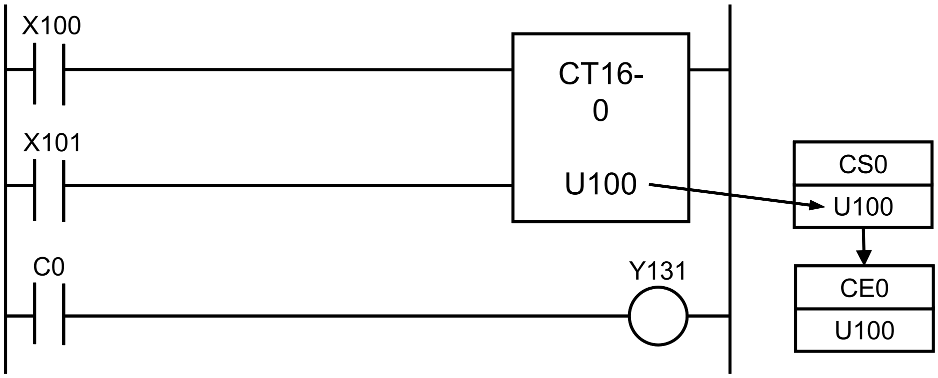
CT instruction (When specifying operand S=constant)

U100 is set in the setting value area CS0 when compiling the program. The setting value area CS0 is set in the elapsed value area CE0 when the input of X101 falls. The setting range of a constant is U0 to U4294967295.



CT16 instruction (When specifying operand S=constant)

U100 is set in the setting value area CS0 when compiling the program. The setting value area CS0 is set in the elapsed value area CE0 when the input of X101 falls. The setting range of a constant is U0 to U65535.

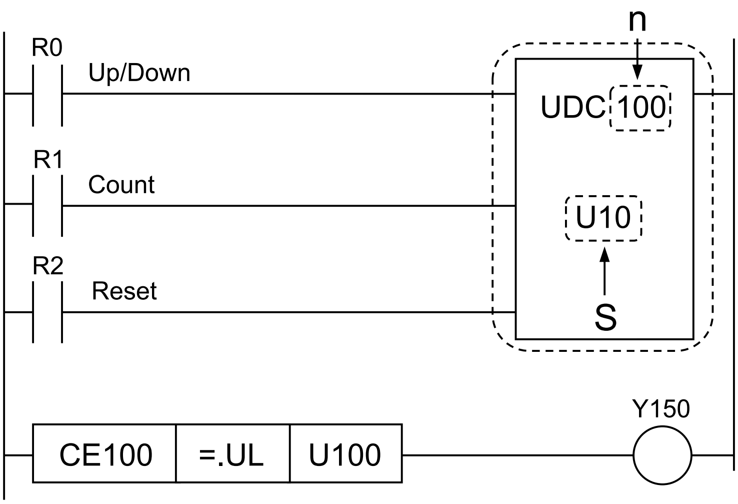


Note

- When a 32-bit value is written into the elapsed value area CE while the counter is being operated using an instruction such as MV instruction, the timer operates with the written 32-bit value. When a 32-bit value is written into the setting value area CS, the timer operates with the written 32-bit value.

3.22 UDC (Up/Down Counter)

■ Ladder diagram



■ List of operands

Operand	Description
n	Counter number
S	Counter set value

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 2) (Note 3)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS (Note 1)	TE CE	IX	K	U	H	SF	DF	" "	
n																●					●
S	●	●	●	●			●	●				●				●					●

(Note 1) Only CS can be specified.

(Note 2) If you specify the index modifier for operand n (counter number), you cannot specify an integer constant for operand S (counter set value).

(Note 3) If specifying the index modifier for operand S (counter set value), you can specify only for 16-bit device or 32-bit device.

■ Outline of operation

- This counter can increment (up count) or decrement (down count) depending on the ON/OFF state of the relay specified by the up/down input.
- The count operation is incremental counting (+1) when the up/down input is ON, and decremental counting (-1) when the up/down input is OFF. The elapsed value is stored in the [CEn] area.
- The preset value in [S] is transferred to [CEn] when the reset input changes from ON to OFF.
- When the count input changes from OFF to ON, the counter starts counting from the value set in [CEn].
- When the reset input turns ON, the elapsed value in [CEn] will be cleared.
- The count result can be judged by comparing the elapsed value in [CEn] with an arbitrary set value by the data comparison instruction.
- Be sure to execute the data comparison instruction immediately after the UDC instruction.

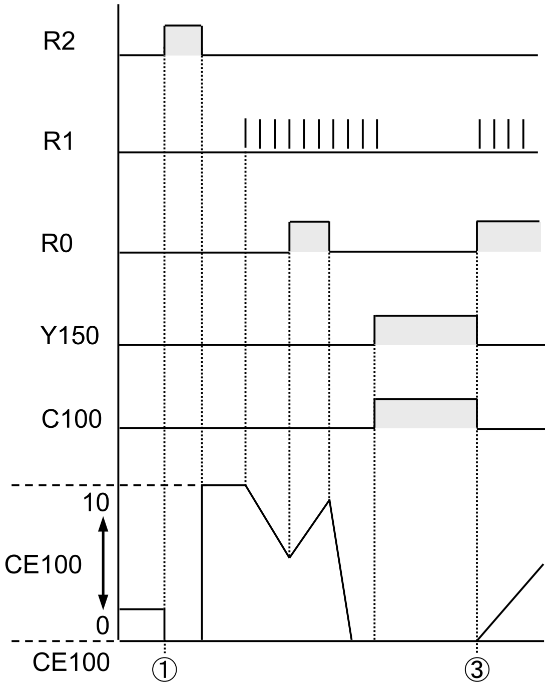
■ Setting the count value

- The count value is specified within the range from U1 to 4294967295, using a decimal (U) constant.

■ Operation Example

1) When the reset input R2 changes from ON to OFF, the set value: U100 is transferred to the elapsed value: CE100. This value is used as the target.
2) When R1 turns ON while R0 is OFF, 1 is subtracted from the elapsed value: CE100 (decremental counting).

   When R1 turns ON while R0 is ON, 1 is added to the elapsed value: CE100 (incremental counting).
3) The counter elapsed value: CE100 and U0 are compared, and if CE100=U0, the external output Y150 turns ON.



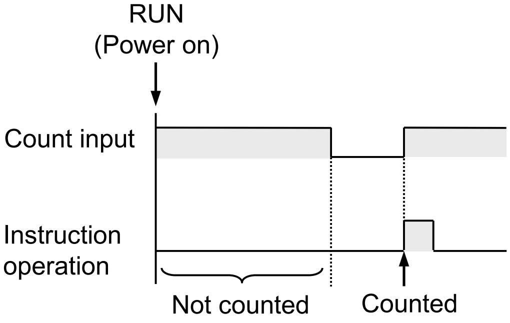
■ Precautions for programming

- If a hold type memory area is specified for the elapsed value area, the elapsed value acts in accordance with the content being held.
- Be aware that the default value when starting operation is not automatically preset to the elapsed value area. When performing preset, switch reset input from ON to OFF.
- When using an UDC instruction with an AND stack (ANS) instruction or pop stack (POPS) instruction, be sure to write the code correctly.

■ Cautions on detecting the count input

- The UDC instruction detects rising from OFF to ON of the count input and increments or decrements the set value.
- Counting is only performed at the rise, so even if the count input remains on, no further counting will occur.

- The set value will not be incremented or decremented for the first scan if the count input is already ON when the operation mode is switched to RUN or the power is turned on in the RUN mode.



- Be careful when using this instruction with an instruction that changes the order of instruction execution such as MC - MCE or JP - LBL (1 to 6, shown below), because the operation of the instruction may change depending on the timing of the instruction execution and count input.

  1) MC - MCE instructions

  2) JP - LBL instructions

  3) LOOP - LBL instructions

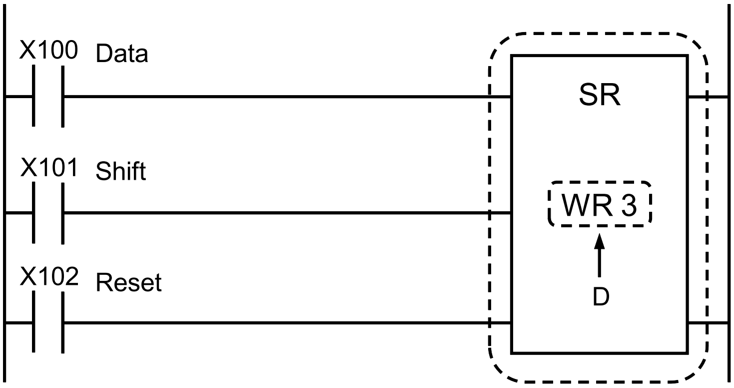
  4) CNDE instruction

  5) Step ladder instructions

  6) Subroutine instructions

3.23 SR (Shift Register)

■ Ladder diagram



■ List of operands

Operand	Description
D	Shifted device

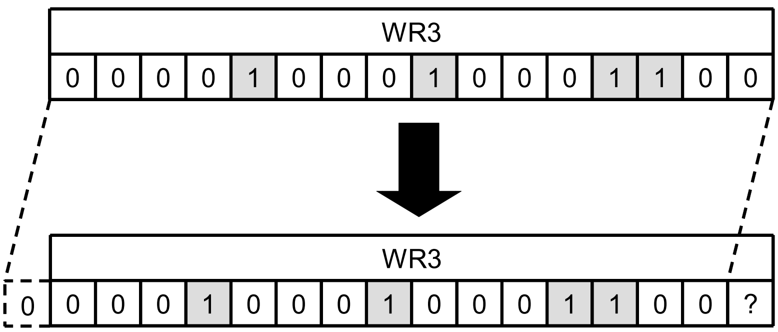
■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D			●																		

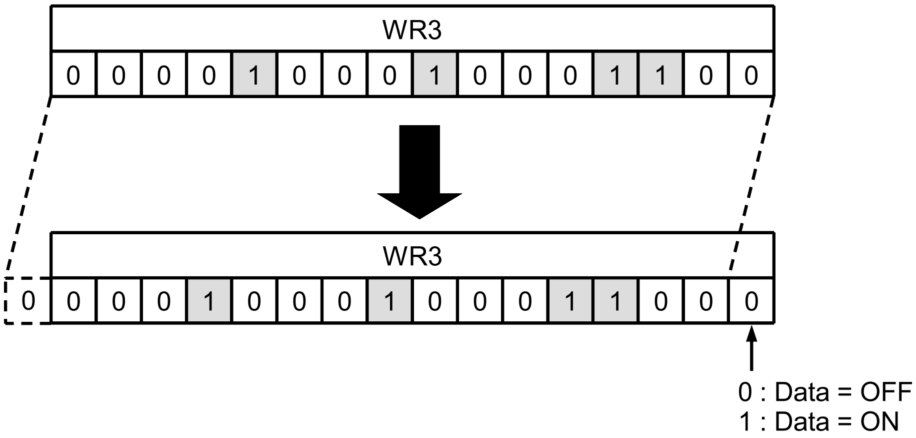
■ Outline of operation

- This instruction moves (i.e., shifts) the contents of the specified register WR (specified operation unit) to the left by one bit.

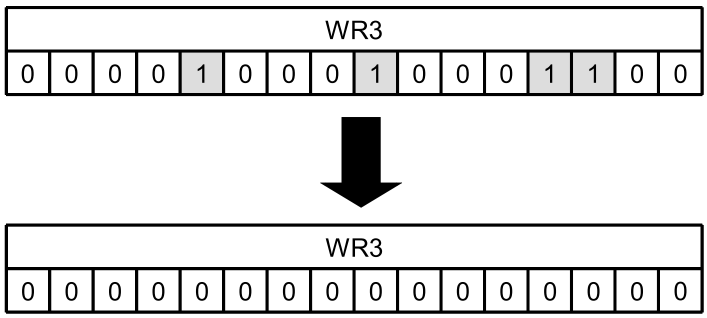
1) When the shift input turns ON (rises), shifts the contents of WR to the left by one bit.



2) Upon shifting, sets 1 or 0 to the blank bit (least significant bit) if the data input is ON or OFF, respectively.

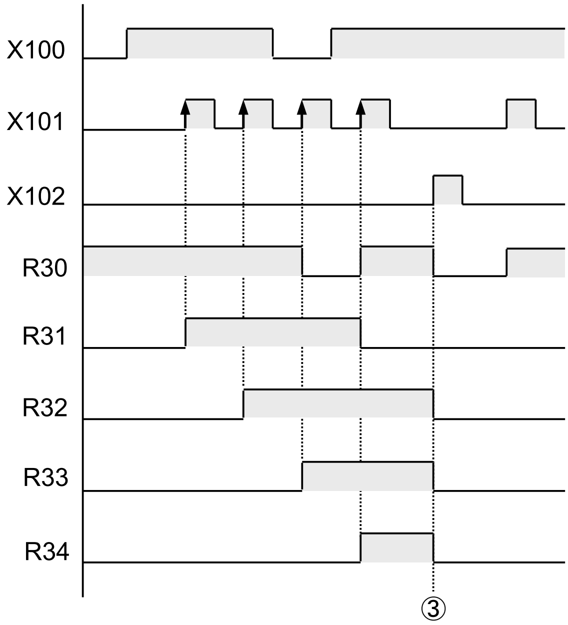


3) When the reset input turns ON, the specified register contents will be cleared.



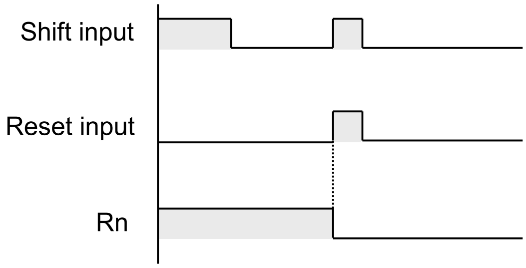
■ Operation Example

1) When X101 turns ON while X102 is OFF, the contents of WR3 (internal relays R30 to R3F) are shifted by 1 bit to the left.
2) In the bit that has become blank due to left shifting (R30), 1 is set when X100 is ON and 0 is set when X100 is OFF.
3) When X102 turns ON, the contents of WR3 are reset to 0.



■ Precautions for programming

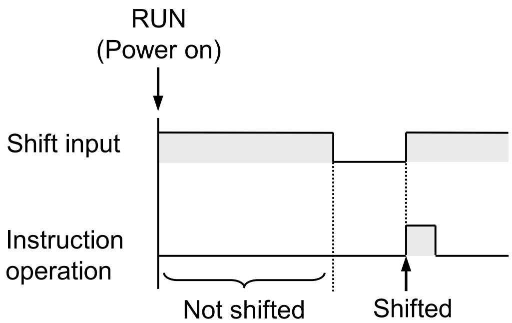
- The data input, shift input, and reset input are required for the SR instruction.
- When reset input and shift input rise simultaneously, reset input is prioritized.



- Note that if a hold-type memory area is specified for the shift register, it will not be automatically reset upon power ON.
- When combining a shift register instruction with an AND stack instruction or pop stack instruction, make sure that the syntax is correct.

■ Precautions for shift input detection

- The SR instruction performs a shift when an OFF to ON rise is detected.
- While the shift input remains ON, shifting is performed at the leading edge only and not performed later.
- The register contents will not be shifted for the first scan if the shift input is already ON when the operation mode is switched to RUN or the power is turned on in the RUN mode.



- Be careful when using this instruction with an instruction that changes the order of instruction execution such as MC - MCE or JP - LBL (1 to 6, shown below), because the operation of the instruction may change depending on the timing of the instruction execution and shift input.

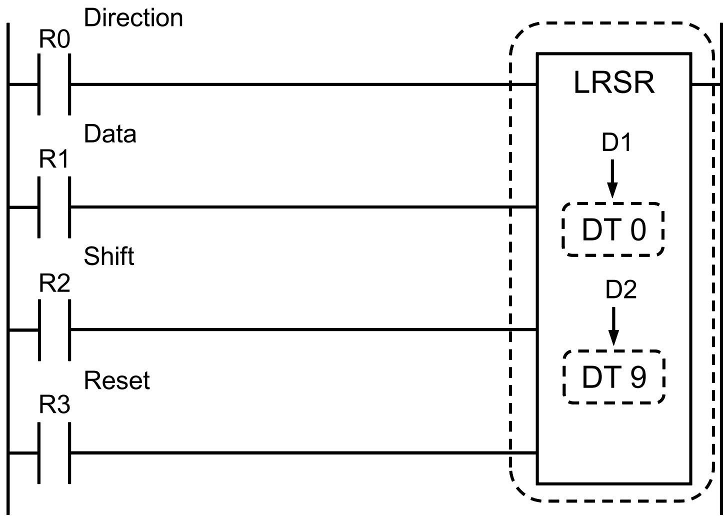
  1) MC - MCE instructions
  2) JP - LBL instructions
  3) LOOP - LBL instructions
  4) CNDE instruction
  5) Step ladder instructions
  6) Subroutine instructions

■ Related instructions

- The left/right shift register (LRSR) instruction is also provided in addition to this instruction for the shift register. You can also use the data shift or data rotate instruction to implement the same operation.

3.24 LRSR (Left/Right Shift Register)

■ Ladder diagram



■ List of operands

Operand	Description
D1	Shift starting position
D2	Shift ending position

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D1	●	●	●	●			●	●	●		●										
D2	●	●	●	●			●	●	●		●										

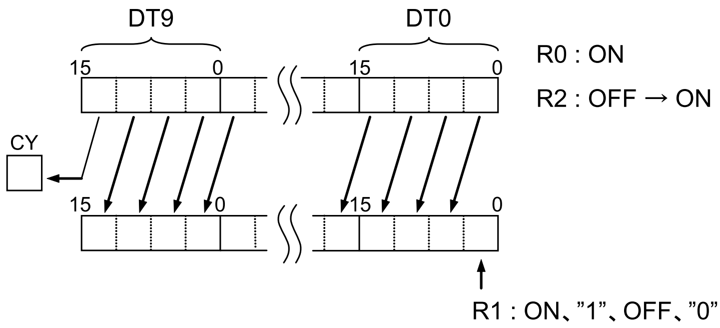
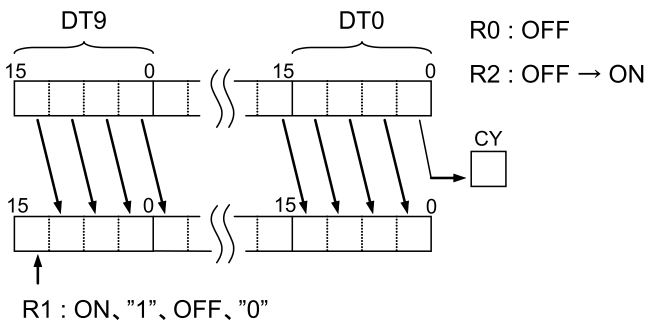
■ Outline of operation

- Left/right shift is a shift register which shifts one bit of the specified data area to the left (to the higher bit position) or to the right (to the lower bit position), depending on the ON/OFF state of the relay specified as the left/right shift input.
- The shift operation is made to the left when the left/right shift input is ON, and to the right when OFF.
- Specify the same type of memory area for [D1] and [D2]. Additionally, specify values so that [D1] is equal to or less than [D2].

■ Operation of LRSR (left/right shift register)

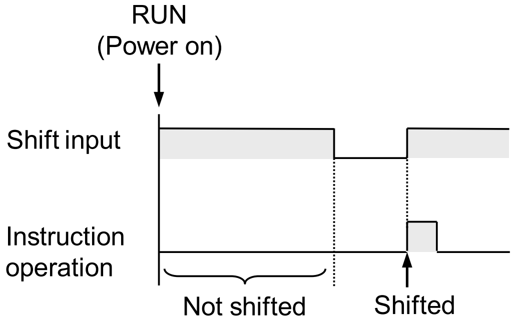
1. When the shift input changes from OFF to ON (the reset input is OFF), the contents of the area specified by [D1] and [D2] are shifted one bit to the left or right.
2. When the data is shifted, 1 will be set in the empty bit left by the shift (the most significant bit or least significant bit) if the data input is ON, and 0 if the data input is OFF. In addition, the bit pushed out due to shifting (the most significant bit for left shift and the least significant bit for right shift) is set in the system relay SR9 (carry flag).
3. If the reset input is ON, the contents of the specified area are cleared to 0.

■ Operation diagram

Left shift	
Right shift	

■ Precautions for programming

- The LRSR instruction detects rising from OFF to ON of the shift input and shifts the register contents. If the shift input remains continuously ON, a shift will only take place at the rise. No further shifts will take place.
- The register contents will not be shifted for the first scan if the shift input is already ON when the operation mode is switched to RUN or the power is turned on in the RUN mode.



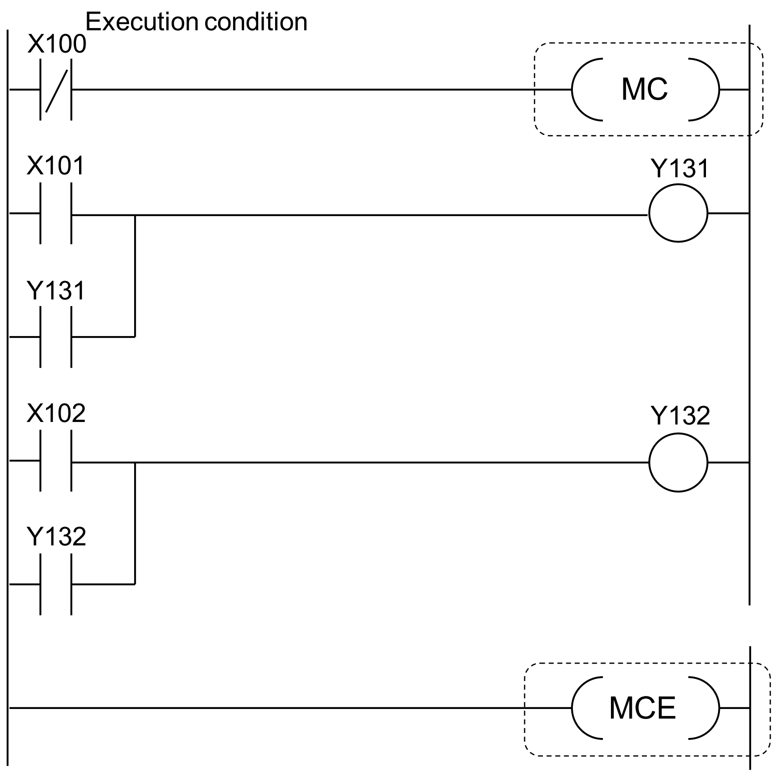
- Be careful when using this instruction with an instruction that changes the order of instruction execution such as MC - MCE or JP - LBL (1 to 6, shown below), because the operation of the instruction may change depending on the timing of the instruction execution and shift input.
  1) MC - MCE instructions
  2) JP - LBL instructions
  3) LOOP - LBL instructions
  4) CNDE instruction
  5) Step ladder instructions
  6) Subroutine instructions
- When using an LRSR instruction with an AND stack (ANS) instruction or pop stack (POPS) instruction, be sure to write the code correctly.

■ Flag operations

Name	Description
SR7, SR8 (ER)	Turns ON when [D1] address > [D2] address.
SR9(CY)	Turns ON when the value pushed out due to shifting is "1."

3.25 MC (Master Control Relay), MCE (Master Control Relay End)

■ Ladder diagram



■ Outline of operation

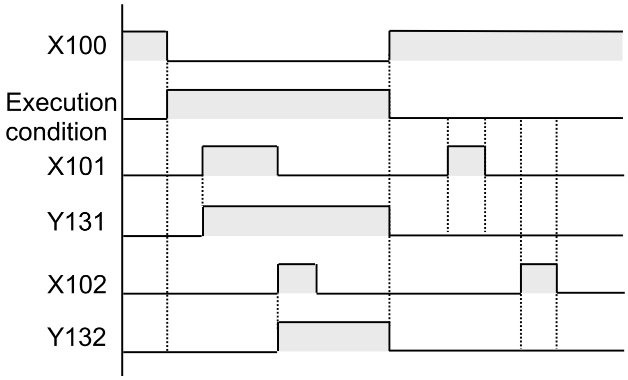
- When the execution condition is ON, this instruction runs the program code between the MC and MCE instructions.
- When the execution condition is OFF, the state of each I/O relay is as follows.

Type of instruction	Operation
OT	All OFF.
KP	Holds the state.
SET	
RST	
TM	Reset.
CT	Holds the current progress.
SR	
Differential instruction	Refer to "Operation of differential instructions between MC and MCE".(Note 1)
Other instructions	Not executed.

(Note 1) The following items are included in differential instructions.

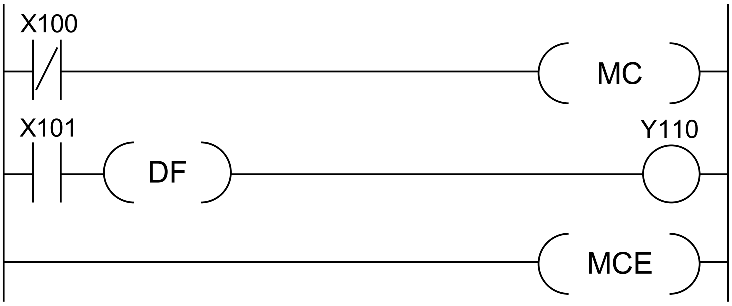
1) DF (leading edge differential) instruction
2) Count input for CT (counter) instruction
3) Count input for UDC (up-down counter) instruction
4) Shift input for SR (shift register) instruction
5) Shift input for LRSR (left and right shift register) instruction
6) Differential execution type high-level instruction (instruction specified by p and instruction name)

■ Operation Example

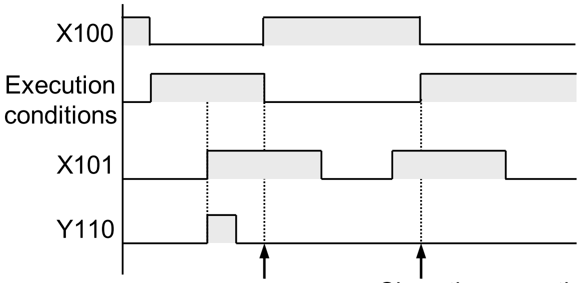


■ Operation of differential instructions between MC and MCE

- If the differential instruction is used between MC and MCE, the output obtained differs depending on the execution condition of MC and the input timing of the differential instruction as shown below.



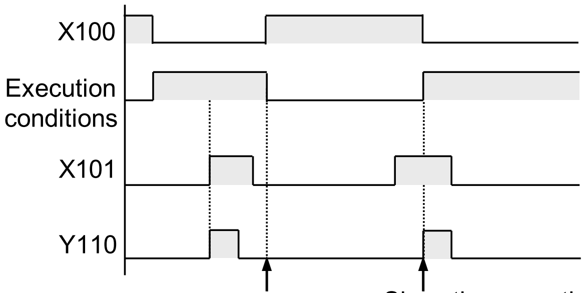
<Timing Chart 1>



Last differential instruction execution

Since the execution condition X101 for the differential instruction has not changed from the last execution, no differential output is obtained.

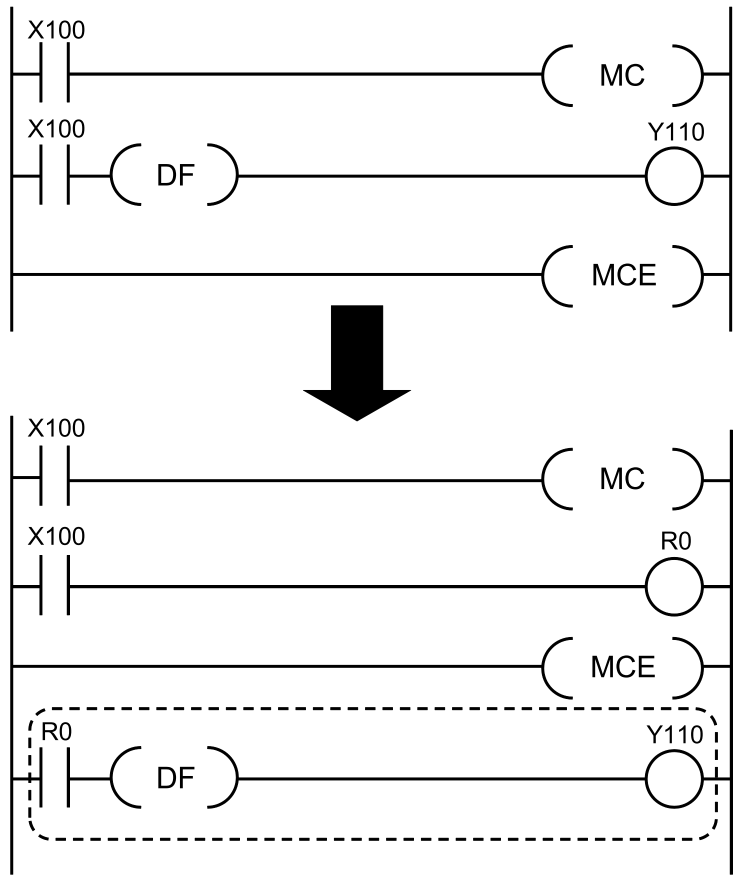
＜ Timing Chart 2＞



Last differential instruction execution

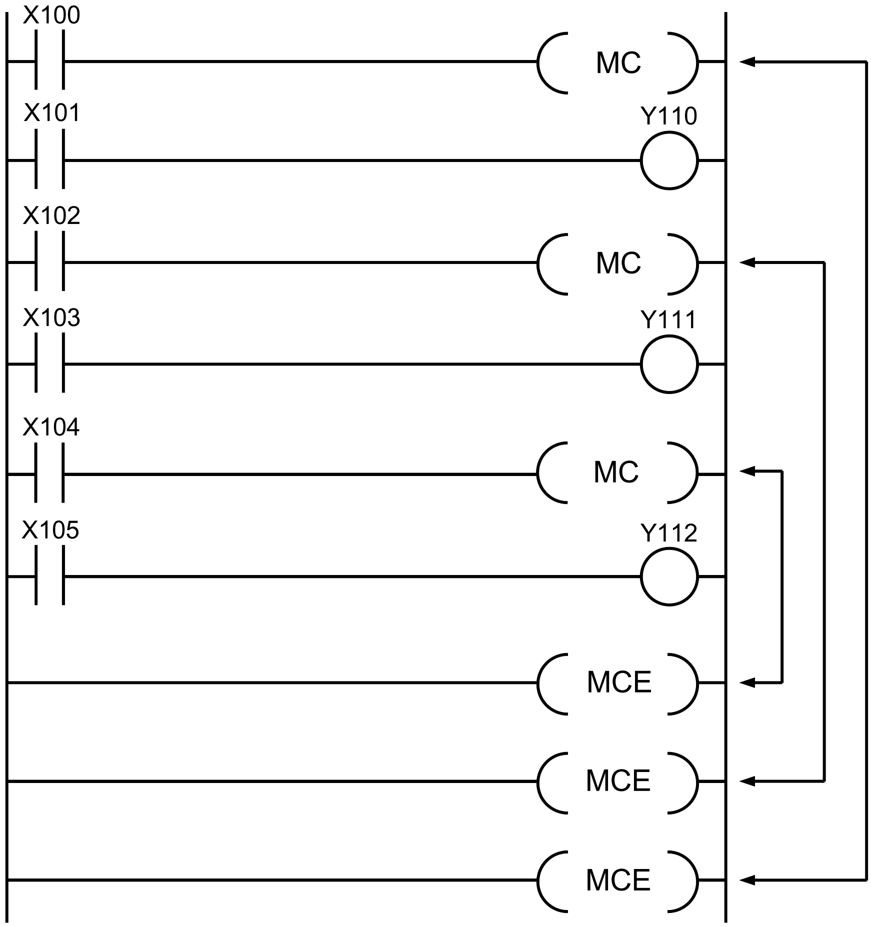
Since the execution condition X101 for the differential instruction has changed from OFF to ON from the last execution, differential output is obtained.

- Output will not be obtained if the same execution condition is specified for an MC instruction and a differential instruction. If output is needed, enter the differential instruction outside of the MC-MCE instruction sequence.

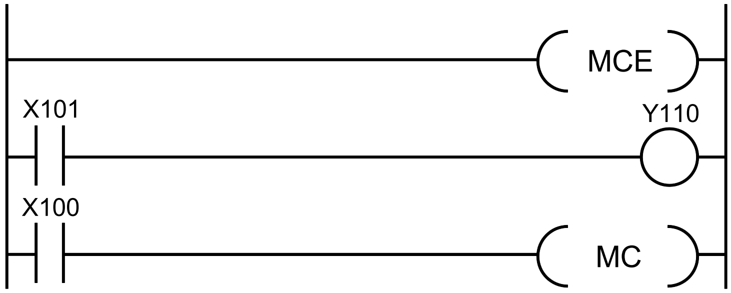


■ Precautions for Programming

- A pair of the MC and MCE instructions can be nested within another pair of the MC and MCE instructions. (Up to 30 levels of nesting are allowed for MC and MCE.)

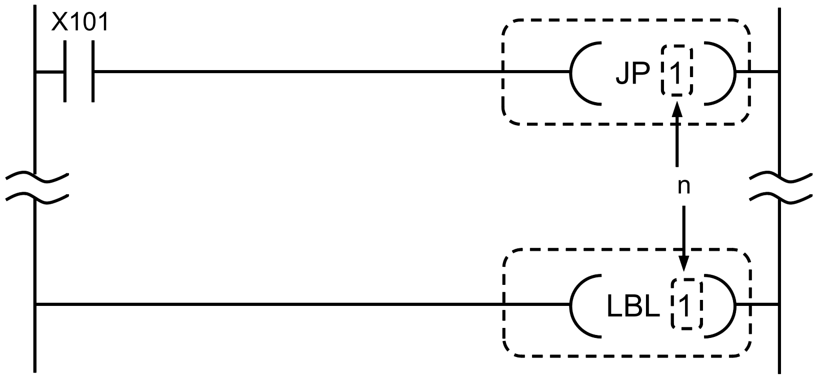


- The program cannot be executed in the following cases.
  (1) "MC" or "MCE" is missing in a pair.
  (2) The order of "MC" and "MCE" is reversed.



3.26 JP/LBL (Jump/Label)

■ Ladder diagram



■ List of operands

Operand	Description
n	Label number (relative jump pointer to LBL)

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											16-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U (Note 1)	H	SF	DF	" "	
n																●					

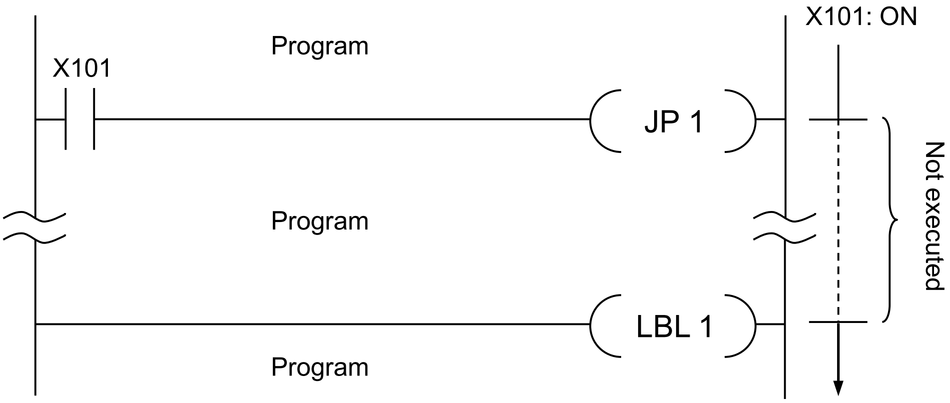
(Note 1) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

- When the execution condition is ON, this instruction jumps to the label with the specified number (LBL instruction).
- The program continues running from the instruction after the target label.

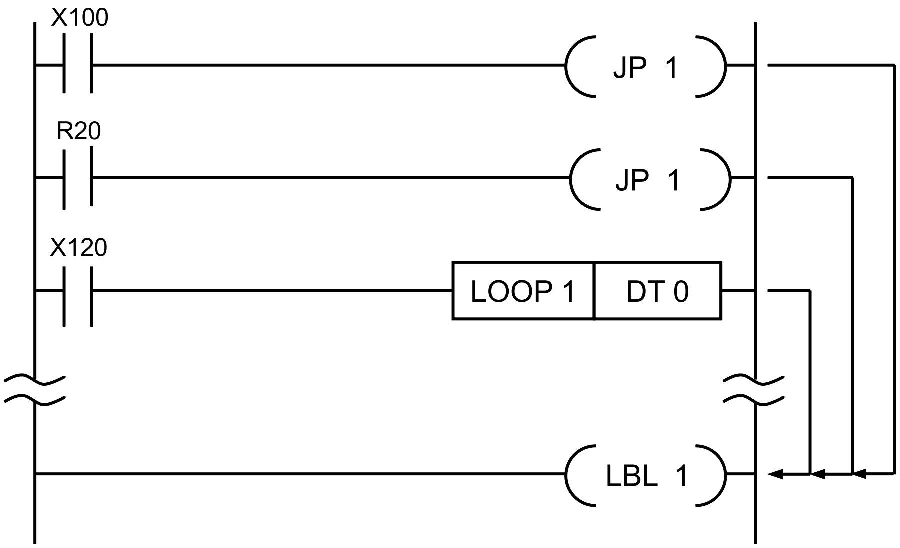
■ Operation Example

- When execution condition X101 turns ON, the program jumps to label 1.



Precautions for programming

- The JP instruction specifying the same label number can be used more than once.



- The LBL instruction specifying the same number can only be written once in a single program.
- If the jump destination label is not programmed, a syntax error occurs.
- Note that if the label is written at an address before the JP instruction, the program cannot end scanning and a WDT error may occur.
- The JP and LBL instructions cannot be used in the step ladder area (within the range from the SSTP to STPE instructions).
- It is not possible to jump from the main program to a subprogram (subroutine or interruption program after the ED instruction), from a subprogram to the main program, or between subprograms.
- Be careful when using an instruction which detects the leading edge of the execution condition and runs (1 - 6 below), including a differential instruction.

  1) DF (leading edge differential) instruction

2) Count input for CT (counter) instruction

3) Count input for UDC (up-down counter) instruction

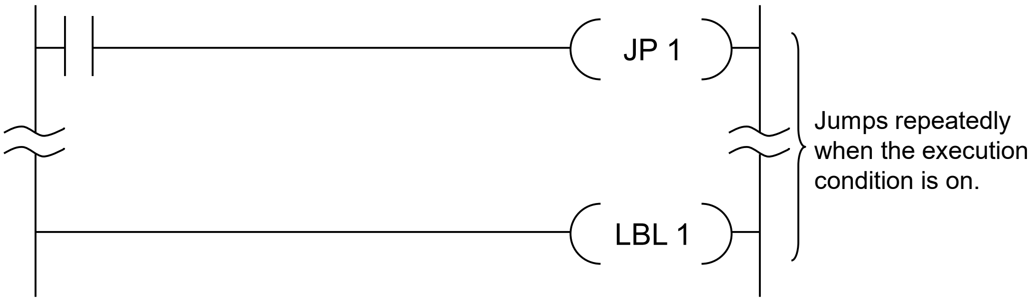
4) Shift input for SR (shift register) instruction

5) Shift input for LRSR (left and right shift register) instruction

6) Differential execution type high-level instruction (instruction specified by p and instruction name)

■ Regarding operation of TM, CT, and SR instructions between JP and LBL instructions

● If the LBL instruction is located at an address after the JP instruction, each instruction is processed as follows when the JP instruction is executed.

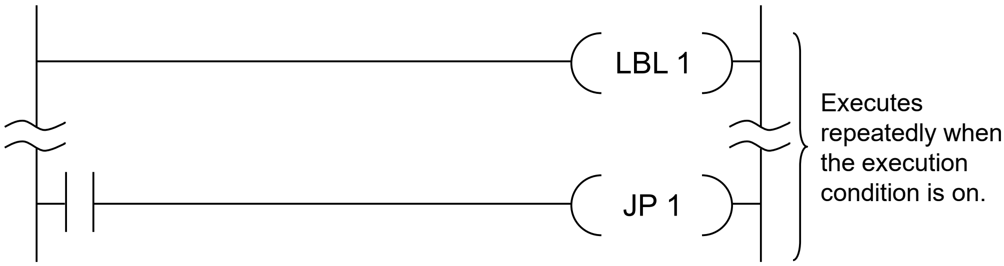


1) TM instruction: Clocking is not performed. If it is not executed once during a single scan, the correct time cannot be guaranteed.

2) CT instruction: Not counted even if the count input is ON. The elapsed value is retained.

3) SR instruction: Even if shift input is ON, no shift is performed. The contents of the specified register are retained.

● If the LBL instruction is located at an address before the JP instruction, each instruction is processed as follows when the JP instruction is executed.



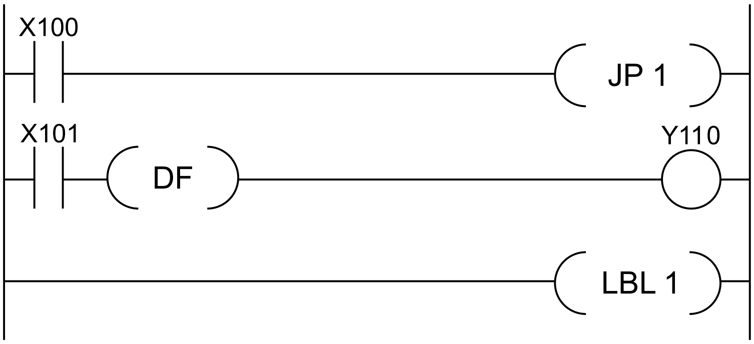
1) TM instruction: Multiple timings occur during a single scan, therefore the time cannot be guaranteed.

2) CT instruction: If the state of the count input does not change during the scan, it will operate in the usual way.

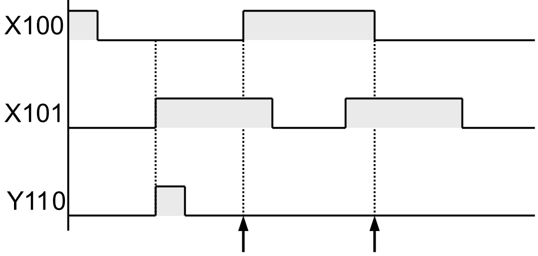
3) SR instruction: If the state of the shift input does not change during the scan, it will operate in the usual way.

■ Operation of differential instruction between JP and LBL instructions

If the differential instruction is used between JP and LBL, the output obtained differs depending on the execution condition of JP and the input timing of the differential instruction as shown below.



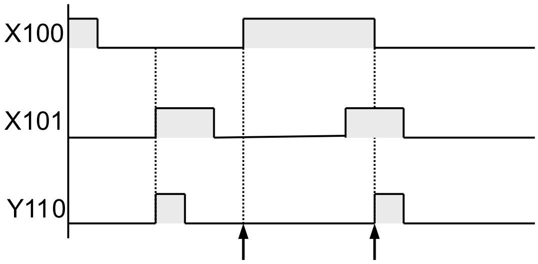
<Timing Chart 1>



Final timing with no execution of the last JP instruction

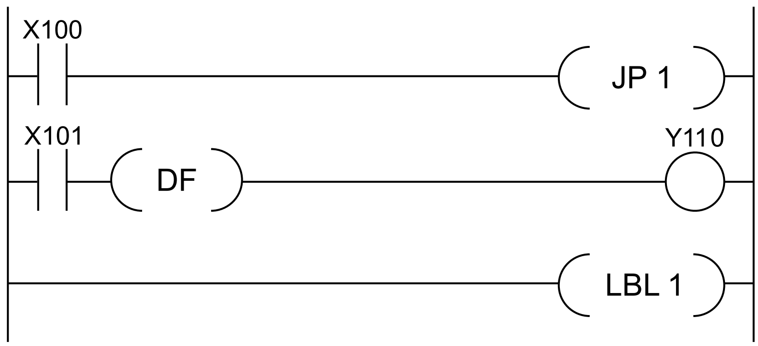
Since the execution condition X101 for the differential instruction has not changed from t he final timing with no execution of the last JP instruction, no differential output is obtained.

<Timing Chart 2>

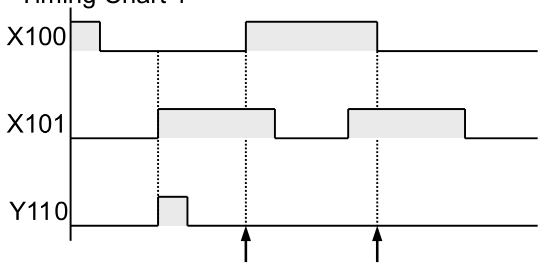


Final timing with no execution of the last JP instruction

Since the execution condition X101 for the differential instruction has changed from OFF to ON from the final timing with no execution of the last JP instruction, differential output is obtained.



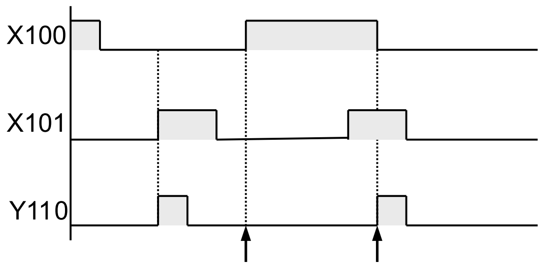
<Timing Chart 1>



Final timing with no execution of the last JP instruction

Since the execution condition X101 for the differential instruction has not changed from t he final timing with no execution of the last JP instruction, no differential output is obtained.

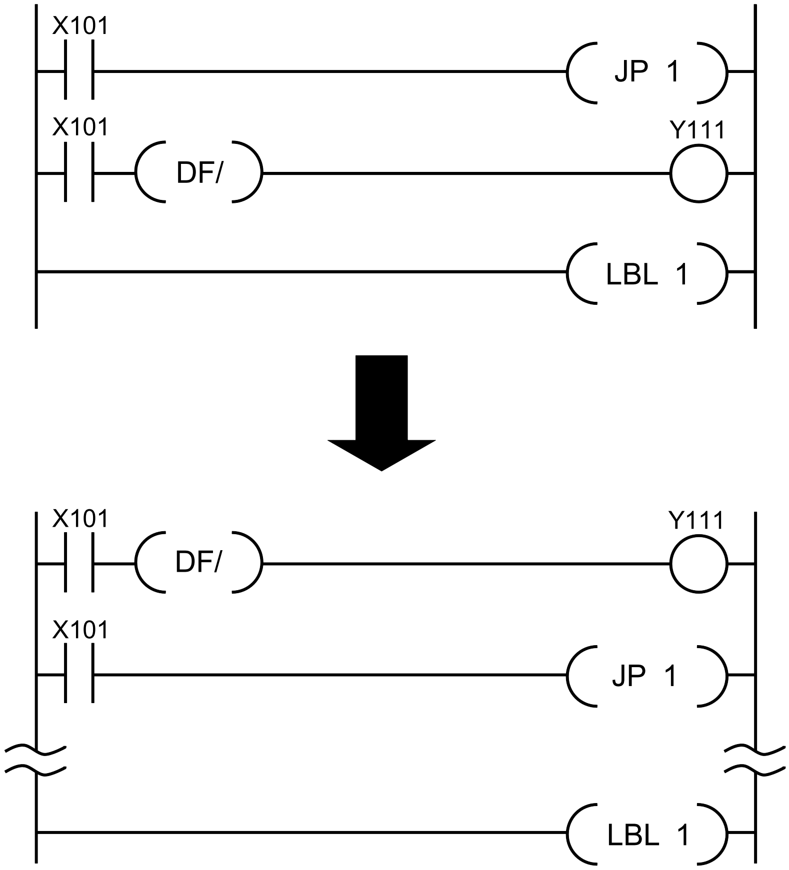
<Timing Chart 2>



Final timing with no execution of the last JP instruction

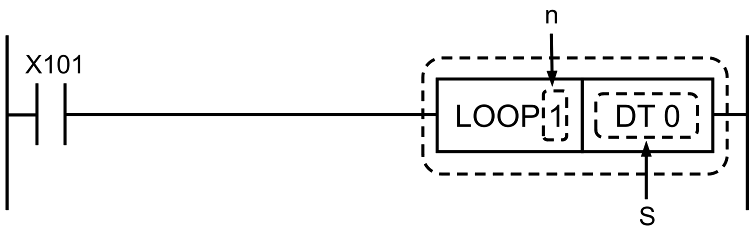
Since the execution condition X101 for the differential instruction has changed from OFF to ON from the final timing with no execution of the last JP instruction, differential output is obtained.

- If the same execution condition is used for the JP and differential instructions, rising (or falling) of the execution condition for the differential instruction will not be detected. If the differential output is necessary, be sure to write the differential instruction outside the range between JP and LBL.


X101
JP 1
X101
Y111
DF/
LBL 1
X101
Y111
DF/
X101
JP 1
LBL 1

3.27 LOOP, LBL (LOOP, Label)

■ Ladder diagram



■ List of operands

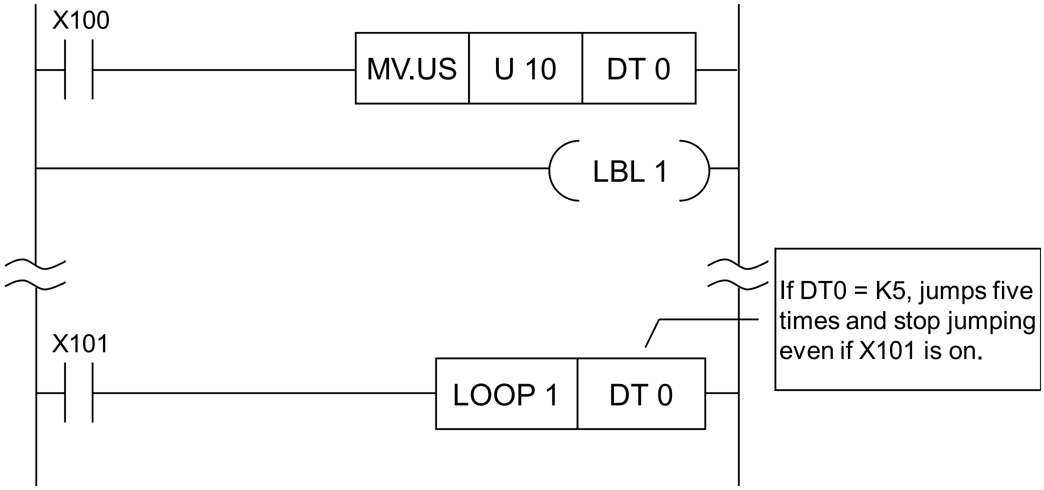
Operand	Description
n	Label number
S	Loop count

■ Devices that can be specified (indicated by ●)

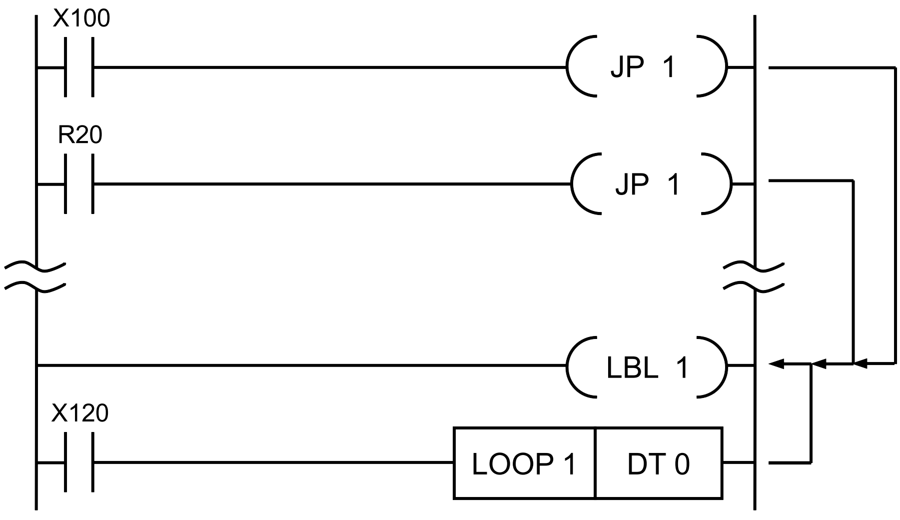
Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
n																●					
S	●	●	●	●			●	●													●

■ Outline of operation

- If the condition is ON, 1 is subtracted from the value in [S]. If the result is not 0, the instruction jumps to the label with the specified number (LBL instruction).
- The program continues running from the instruction after the target label.
- The LOOP instruction specifies the number of times to execute the code. When the code has been executed for the number of times specified in [S], the program will not jump to the specified label even if the execution condition is met.

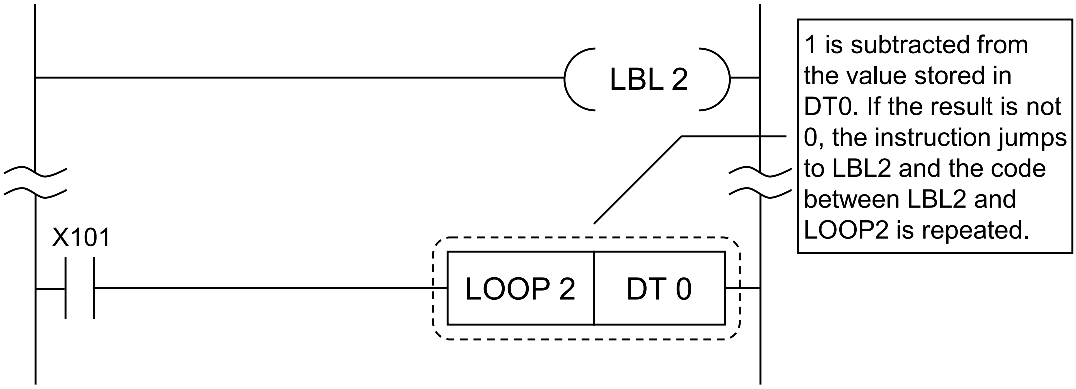


- If the value in the memory area specified in [S] is 0 from the first time, the program does not jump to the specified label but instead performs the next process.
- The label is shared between the JP and LOOP instructions. A label can be used as the jump destination for any instruction, as many times as required.



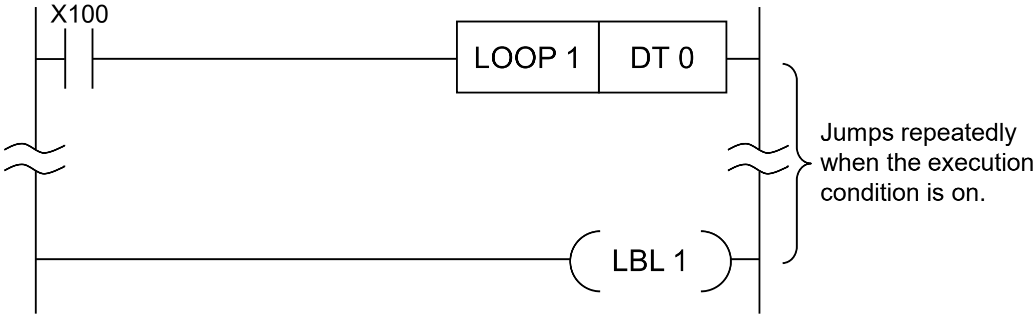
- The LBL instruction specifying the same number can only be written once in a single program.
- If the jump destination label is not programmed, a syntax error occurs.

■ Operation Example



■ Regarding operation of TM, CT, and SR instructions between LOOP and LBL instructions

- If the LBL instruction is located at an address after the LOOP instruction, each instruction is processed as follows when the LOOP instruction is executed.

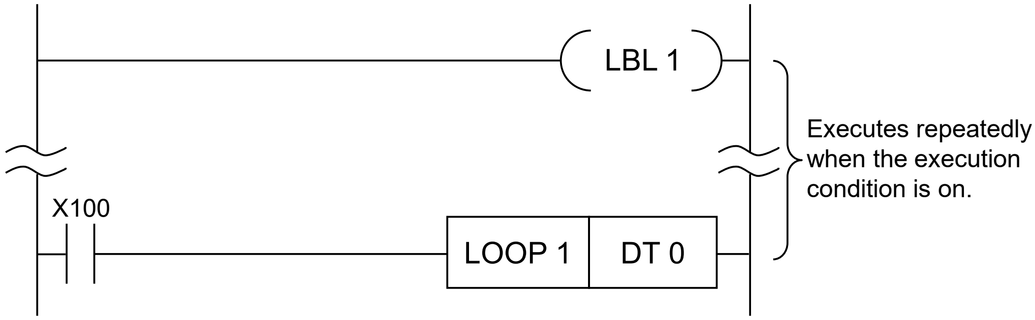


1) TM instruction: Clocking is not performed. If it is not executed once during a single scan, the correct time cannot be guaranteed.

2) CT instruction: Even if count input is ON, counting is not performed. The elapsed value is retained.

3) SR instruction: Even if shift input is ON, no shift is performed. The contents of the specified register are retained.

- If the LBL instruction is located at an address before the LOOP instruction, each instruction is processed as follows when the LOOP instruction is executed.



1) TM instruction: Multiple timings occur during a single scan, therefore the time cannot be guaranteed.

2) CT instruction: If the state of the count input does not change during the scan, it will operate in the usual way.

3) SR instruction: If the state of the shift input does not change during the scan, it will operate in the usual way.

■ Precautions for programming

- When writing a label to an address before the LOOP instruction, pay attention to the following:

  (1) Be sure to write the instruction for setting the loop count before LBL - LOOP instructions.

  (2) Be sure to write the instructions repeated between the LBL and LOOP instructions so that they are executed with the same execution condition as the LOOP instruction.

  (3) While repeating the program code, a single scan may exceed the WDT monitor time limit and a WDT error may occur.
- The LOOP and LBL instructions cannot be used in the step ladder area (within the range from the SSTP to STPE instructions).
- It is not possible to jump from the main program to a subprogram (subroutine or interruption program after the ED instruction), from a subprogram to the main program, or between subprograms.
- Be careful when using an instruction which detects the leading edge of the execution condition and runs (1 - 6 below), including a differential instruction.

  1) DF (leading edge differential) instruction

  2) Count input for CT (counter) instruction

  3) Count input for UDC (up-down counter) instruction

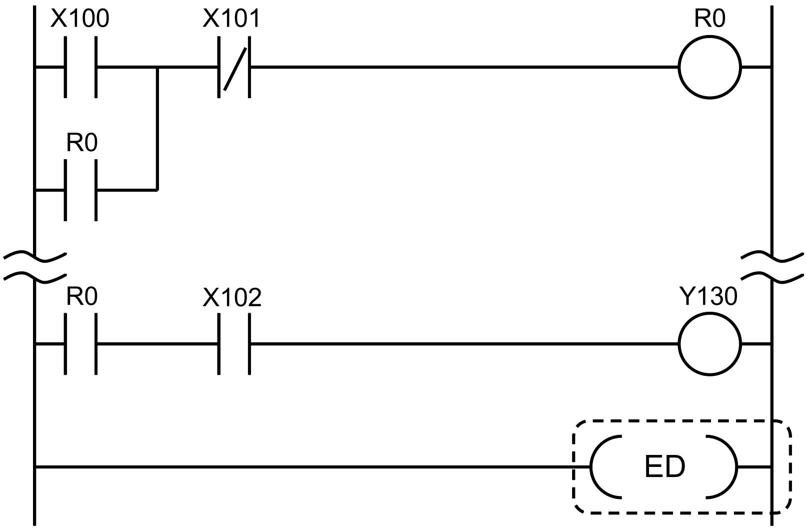
  4) Shift input for SR (shift register) instruction

  5) Shift input for LRSR (left and right shift register) instruction

  6) Differential execution type high-level instruction (instruction specified by p and instruction name)

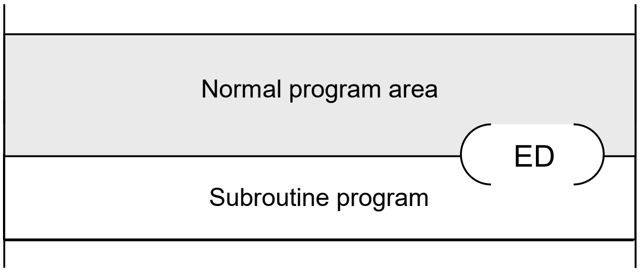
3.28 ED (End)

■ Ladder diagram



■ Outline of operation

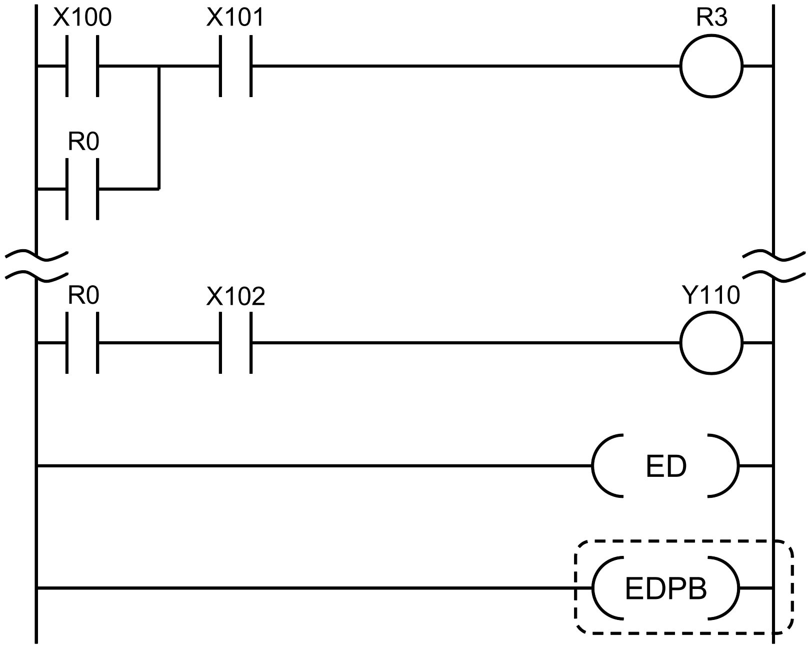
- This instruction writes the ED instruction at the end of the regular program area.



- The program area is divided into the regular program area (main program) and the "subroutine" and "interruption program" areas (subprograms) by this instruction.
- Be sure to write the subroutine and interruption program after the ED instruction.

3.29 EDPB (End Program Block)

■ Ladder diagram

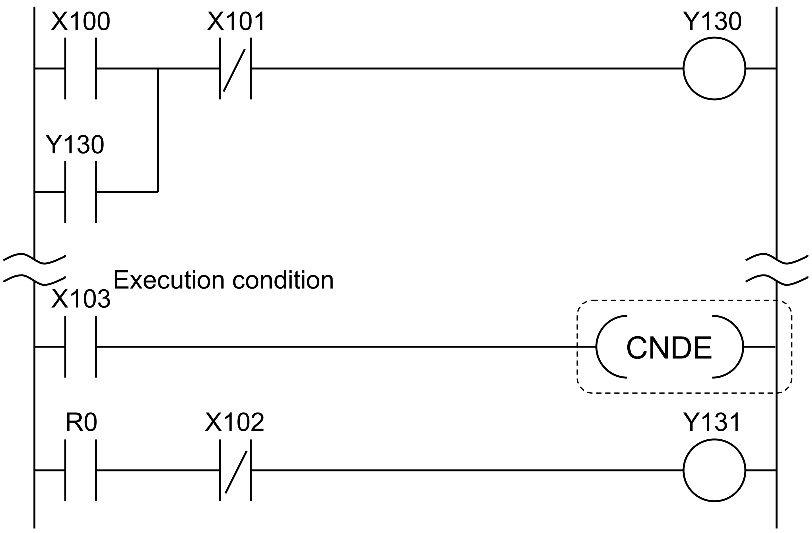


■ Outline of operation

- This instruction indicates the end of PB (program block).

3.30 CNDE (Conditional End)

■ Ladder diagram



■ Outline of operation

- Ends arithmetic processing of the program at the specified address.
- When the execution condition turns ON, the program terminates operation and begins other processing such as I/O. When processing is complete, the operation returns to the starting address.
- The process timing can be adjusted by beginning the process as soon as the necessary program scan finishes.
- The CNDE instruction cannot be used in a subprogram (e.g., subroutine). Use in the main program area.
- The "CNDE" instruction can be used for any number of times in the main program.
- Be careful when using an instruction which detects the leading edge of the execution condition and runs (1 - 6 below), including a differential instruction.

  (1) DF (leading edge differential) instruction

  (2) Count input for CT (counter) instruction

  (3) Count input for UDC (up-down counter) instruction

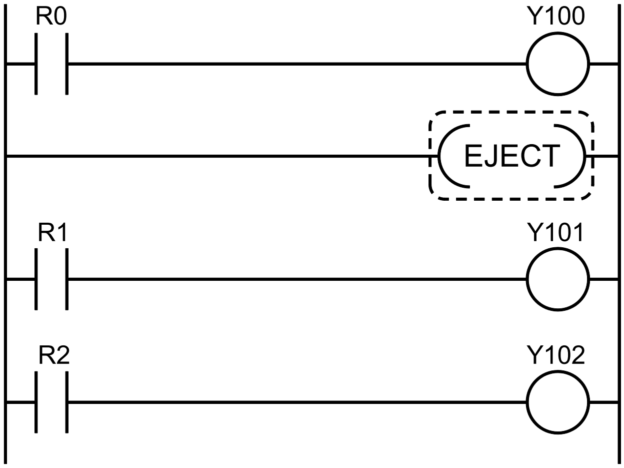
  (4) Shift input for SR (shift register) instruction

  (5) Shift input for LRSR (left and right shift register) instruction

  (6) Differential execution type high-level instruction (instruction specified by p and instruction name)

3.31 EJECT

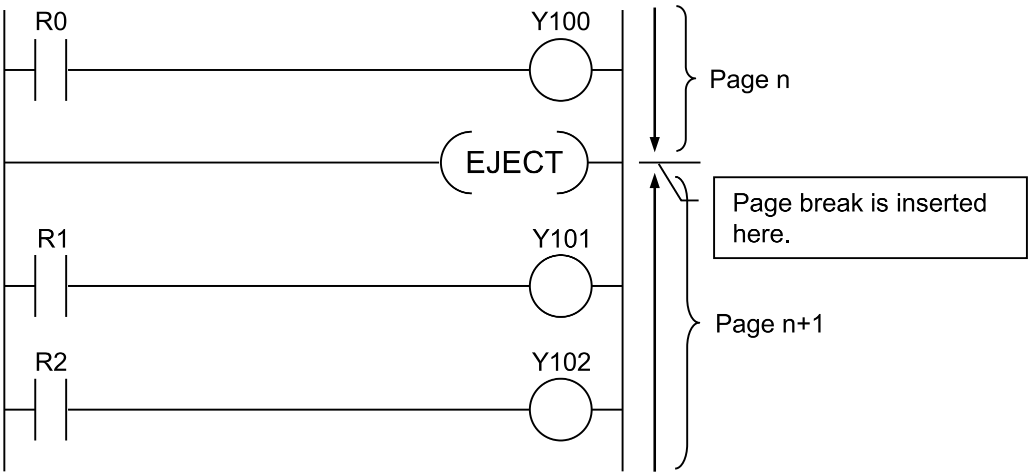
■ Ladder diagram



■ Outline of operation

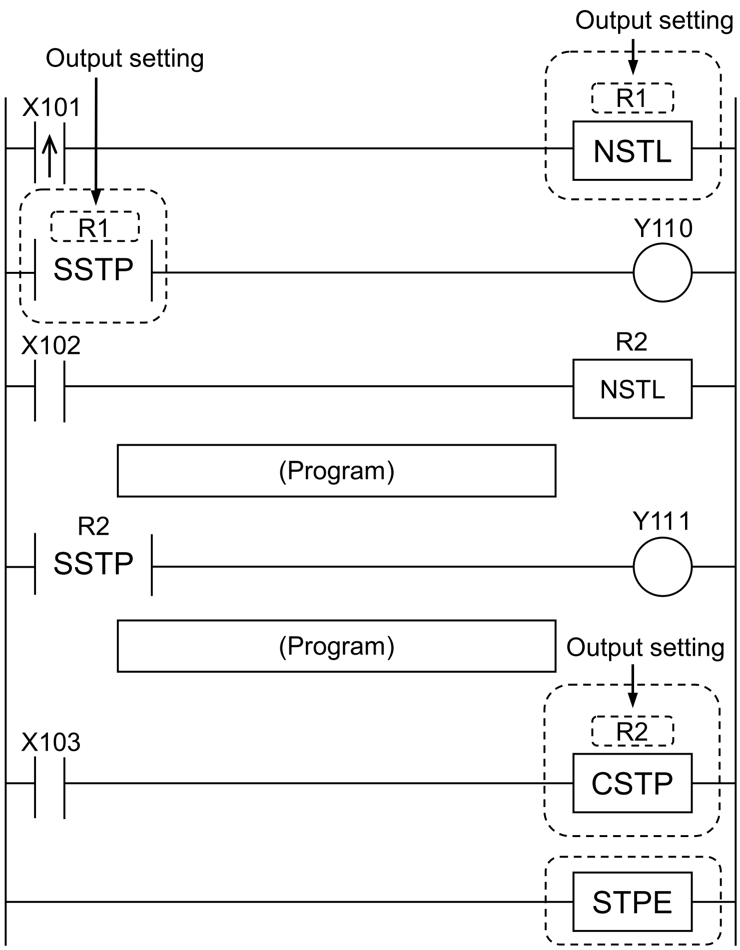
- When creating and printing out program code with the tool software, a page break will be added where this instruction is inserted.
- Similarly to the NOP instruction, no program processing will occur.

■ Operation Example



3.32 SSTP (Start Step) / NSTL (Next Step) / CSTP (Clear Step) / STPE (Step End)

■ Ladder diagram



■ Devices that can be specified (indicated by ●)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
bit	●	●	●	●								●	●	

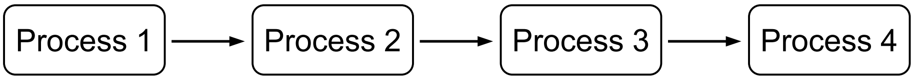
■ Outline of operation

- The NSTL instruction starts and executes a process which begins with the SSTP instruction with the specified number.

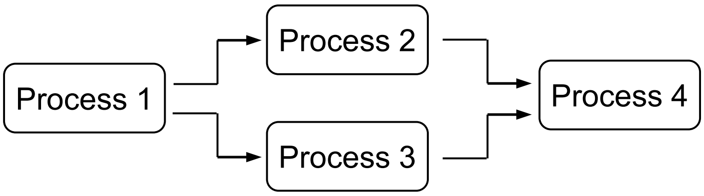
3.32 SSTP (Start Step) / NSTL (Next Step) / CSTP (Clear Step) / STPE (Step End)

- Program code between the SSTP instruction and the next SSTP instruction or the STPE instruction is handled as a single process.
- These instructions make it easy to execute sequence control, selection branch control, parallel branch merge control, and similar operations.

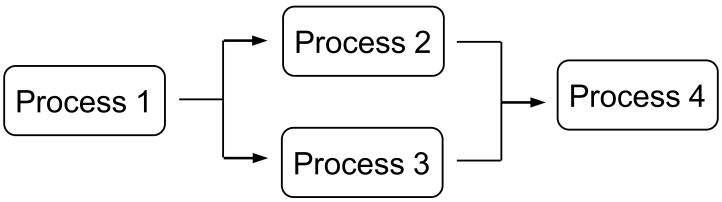
1) Sequence control
Only the necessary processes are switched and executed in order.



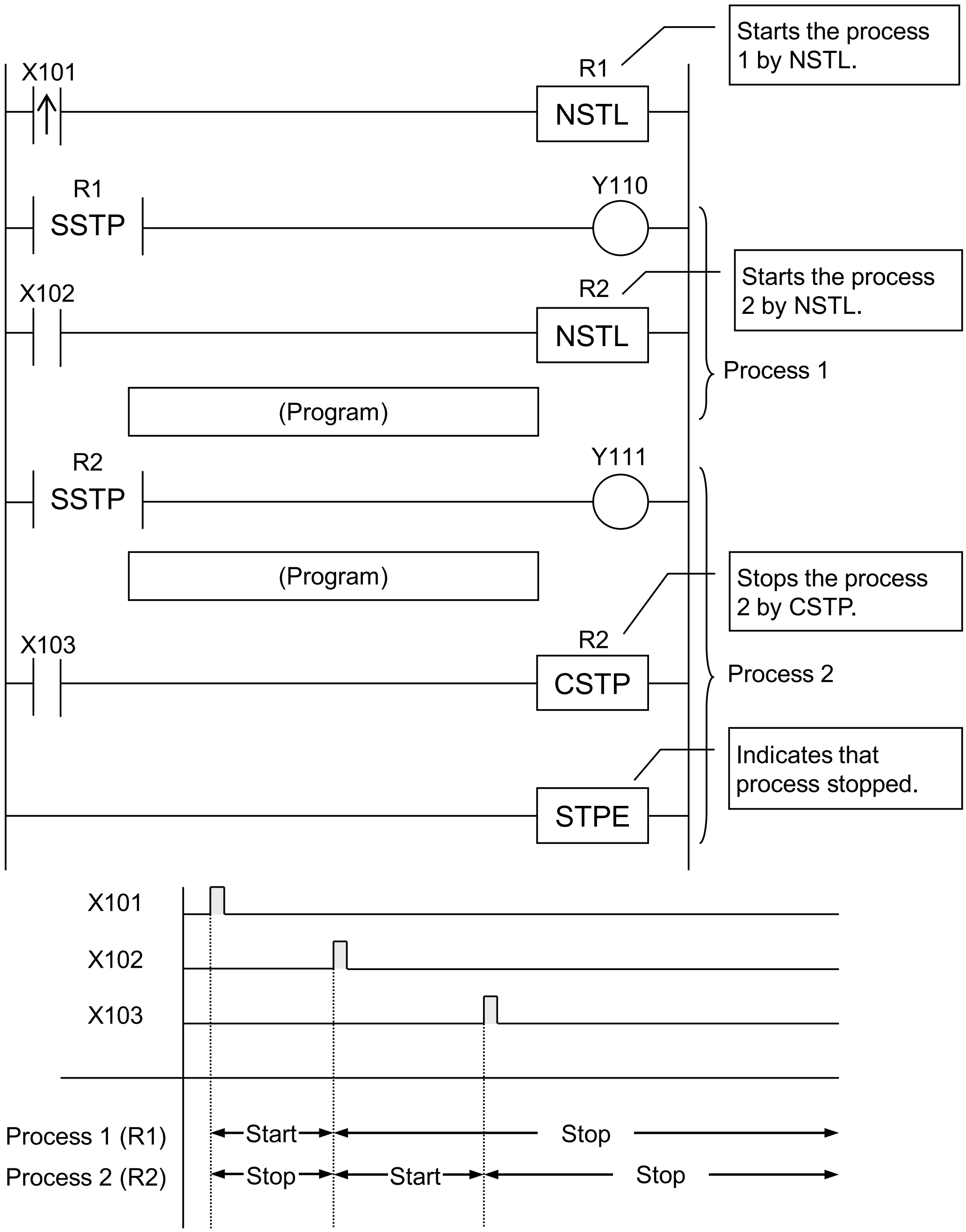
2) Selective branch control
The processes are selected and executed according to conditions.



3) Parallel branch/join control
Multiple processes are executed simultaneously. After each process is completed, the next process is executed.



■ Operation Example



■ SSTP Start Step Instruction

- This instruction indicates "Start of output number [n]." Be sure to write the "SSTP n" at the beginning of the program of the output number [n].
- Program code between "SSTP n" and the next "SSTP" or "STPE" is handled as the area with the output number [n].

- The same process number cannot be defined for more than one process.
- The OUT instruction can be connected directly from the bus bar immediately after the SSTP instruction.
- The SSTP instruction cannot be written in a subprogram (subroutine).
- Program code between the first SSTP instruction and STPE instruction is referred to as a "step ladder area" and is controlled as a process. Other areas are referred to as "normal ladder areas."
- There is a type of system relay which turns ON only for a single scan when a process with a step ladder starts. (SR15: Step ladder initial pulse relay) This relay can be used to perform a process only for a single scan after starting a process (e.g., resetting a counter).

■ NSTL Next Step Instruction (Every Scan Execution Type)

- The NSTL R instruction starts a process specified by the relay number [R].
- The execution condition of the next step instruction becomes the start condition of the process.
- Write the process that starts first in the next step instruction in the normal ladder area.
- The process can be started from the normal ladder area as well as a running process.
- However, when you start a process with a next step instruction from within a process, the process that is executing and contains the next step instruction is automatically cleared and the specified process starts.

■ CSTP Clear Step Instruction

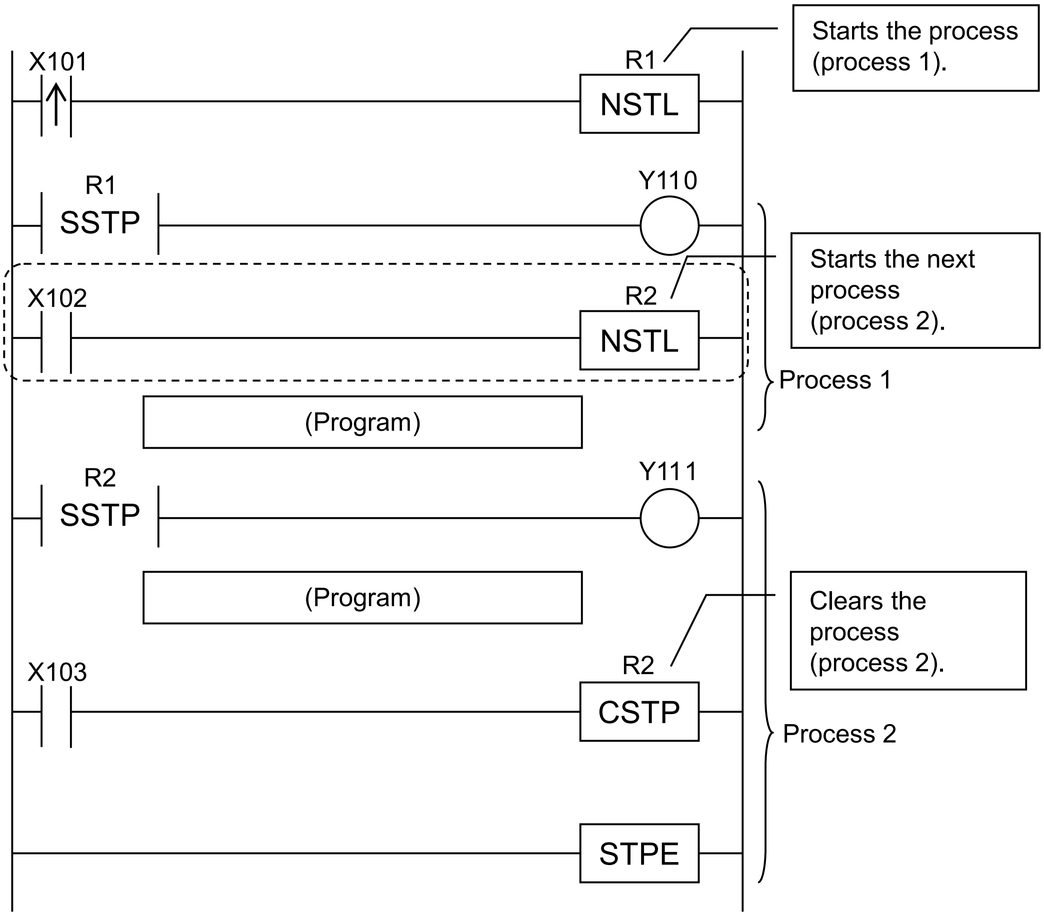
- When the CSTPn instruction is executed, the process specified by "n" is cleared. This instruction can be used to clear the final process or to clear the processes executing in parallel during parallel branch merge control.
- A process can be cleared from the normal ladder area or from a process that is already started.
- When clearing multiple processes, the ZRST instruction can be used.

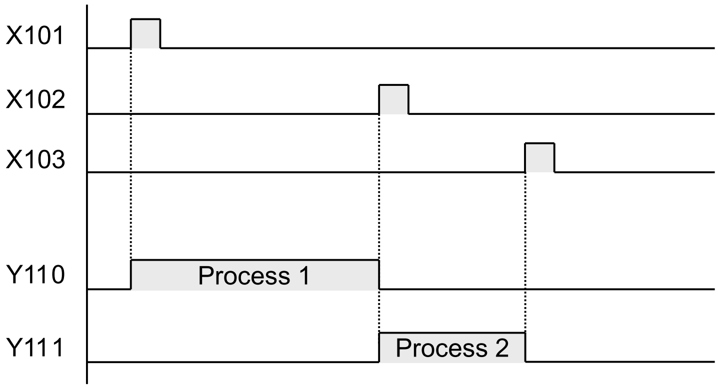
■ STPE Step End Instruction

- This instruction indicates the end of the step ladder area. Be sure to write this instruction at the end of the final process. The final process is defined by the SSTP and STPE instructions.
- Only one STPE instruction can be written in each main program. (It cannot be written in a subprogram such as a subroutine or interruption program.)

■ Example (1) Sequential control of processes

- This is a program that repeats the work in a certain process until it is completed, and then moves to the next process.
- In the program, write the instruction to start the process to be executed next in each process. When the start instruction is executed, the next process is started, and the process that had been executing is cleared.
- Processes do not need to be executed in numerical order. You can also program the start instruction to invoke a previous process according to conditions.

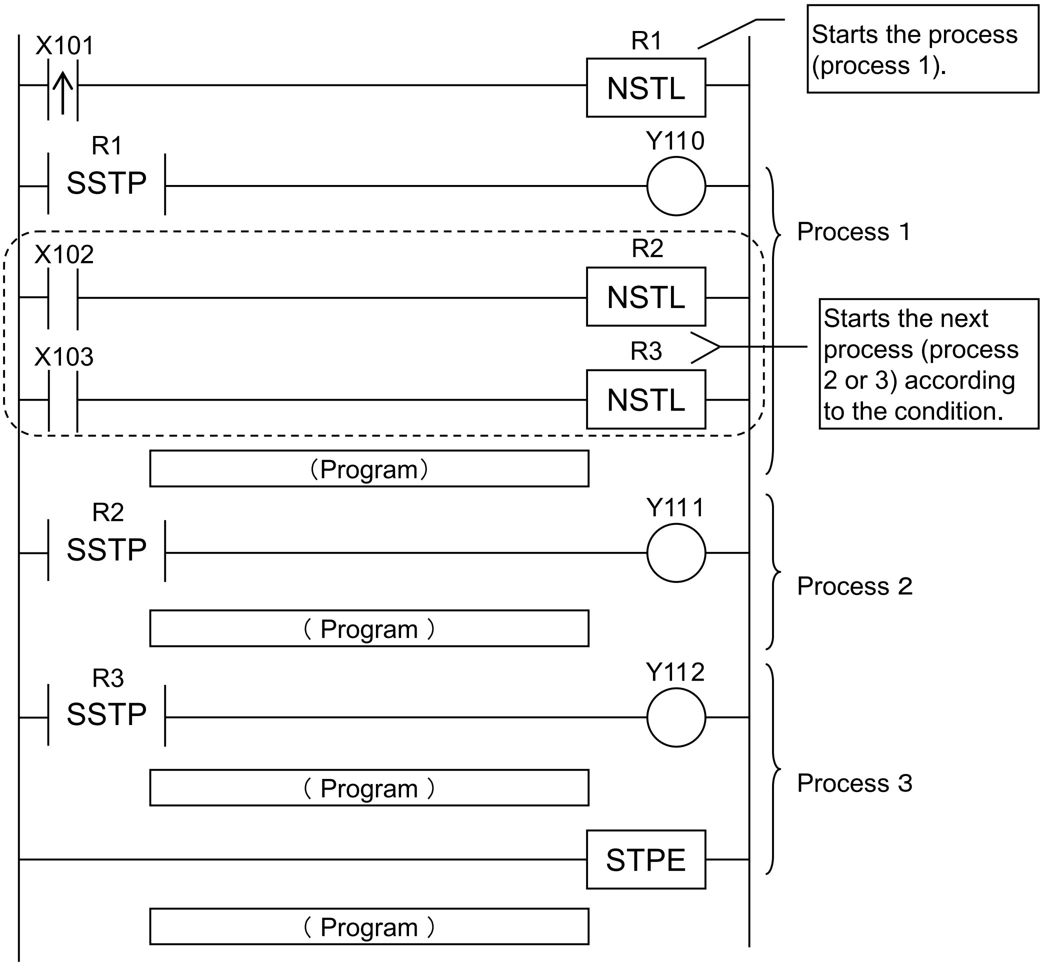


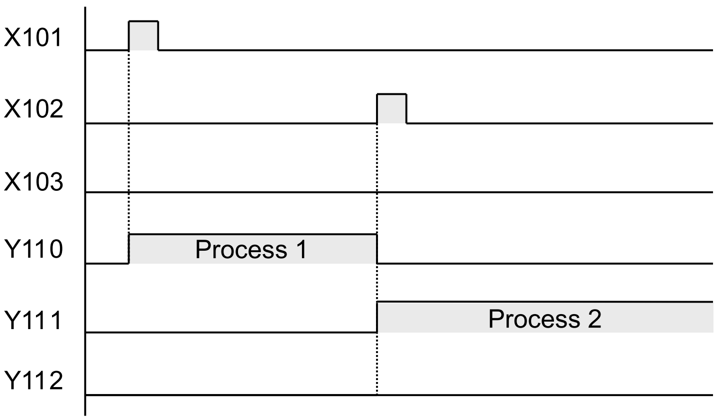


■ Example (2) Selective branch control of processes

- This program selects and switches to the next process according to the actions and results of a particular process. Each process loops until its work is completed.

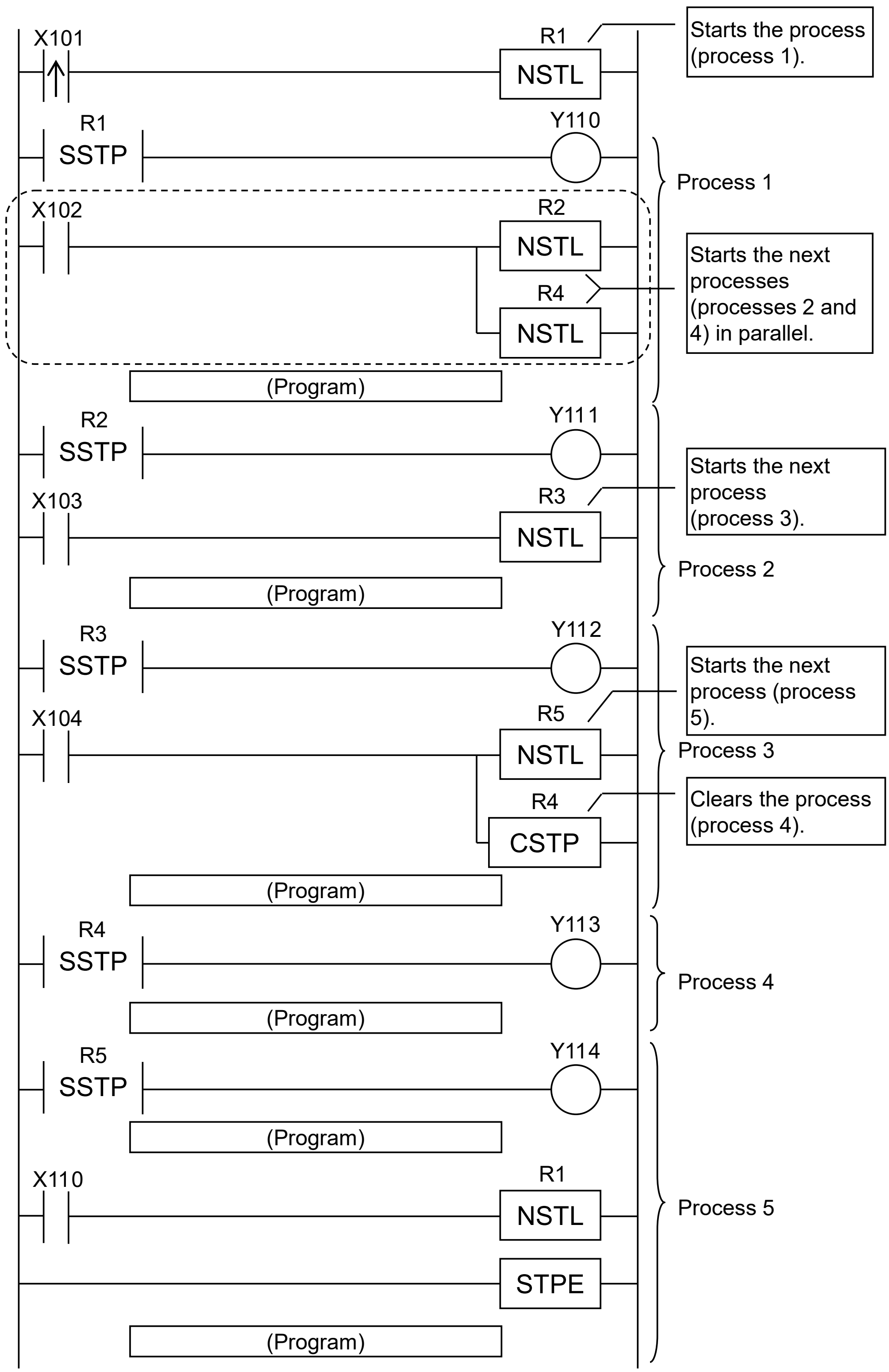
- In the program, write the instruction to start the process to be executed next in each process. The next process is selected and program execution is transferred according to the execution conditions.

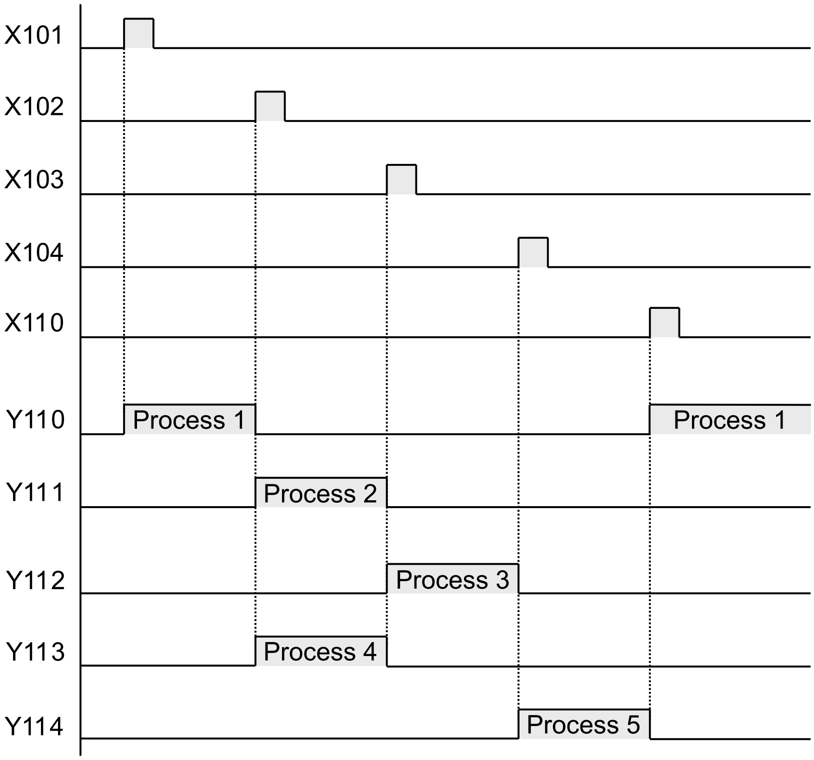




Example (3) Parallel branch/join control of processes

- This program starts multiple processes at the same time. When the work is completed in each of the branched processes, they merge again before transferring execution to the next process.
- In the program, write multiple process transfer instructions for one execution condition in succession in a process.
- To merge processes, include a flag indicating the state of the other processes in the transfer condition for the next process. When they merge and execute the next process, clear all uncleared processes at the same time.


X101
R1
NSTL
Starts the process (process 1).
R1
SSTP
Y110
Process 1
X102
R2
NSTL
R4
NSTL
Starts the next processes (processes 2 and 4) in parallel.
(Program)
R2
SSTP
Y111
X103
R3
NSTL
Starts the next process (process 3).
Process 2
(Program)
R3
SSTP
Y112
X104
R5
NSTL
Starts the next process (process 5).
Process 3
R4
CSTP
Clears the process (process 4).
(Program)
R4
SSTP
Y113
Process 4
(Program)
R5
SSTP
Y114
(Program)
X110
R1
NSTL
STPE
Process 5
(Program)



■ Precautions for programming

- Processes do not need to be written in numerical order.
- In the step ladder area, you cannot use the following instructions:

  1) Jump instructions (JP and LBL)

  2) Loop instructions (LOOP and LBL)

  3) Master control instructions (MC and MCE)

  4) Subroutine instructions (SBL and RET)

  5) ED instruction

  6) CNDE instruction

  Note) The CALL instruction can be used in the step ladder area.
- In order to clear all processes at once, use the master control relay to program the code.
- Processes need not be started in the order of their numbers. You can execute multiple processes simultaneously.
- When the output in a process that has not been started is forcibly turned ON or OFF, even if the forced ON/OFF operation is canceled, the output state will be held until the process starts.

■ Step ladder operations

- The program in the normal ladder area and programs in processes started by the Next Step instruction (NSTL) are executed. The program in processes that are not executing is ignored.

- When a process starts and the first scan is in progress, the step initial pulse relay (SR15) turns ON. It turns OFF for the second and subsequent scans. This relay can be used to reset counters and shift registers.

■ Precautions for clearing a process

- When the Next Step instruction is executed within the program of a running process, the running process will be cleared automatically. The clearing will be done during the next scan. Therefore, two processes may be running simultaneously for a single scan when transiting between them. If there are two outputs which must not be ON at the same time, be sure to provide an interlock to prevent them from tuning ON simultaneously. If these outputs can turn ON simultaneously even when an interlock is provided by the program due to a delay in hardware response, take a counteraction on hardware to consider the delay.
- When a process is cleared, the instructions used in the process will operate as shown in the following table.

Type of instruction	Operation
OT	All OFF.
KP	Holds the state
SET	
RST	
TM	Reset.
CT	Holds the current progress.
SR	
Differential instruction	Refer to "Operation of differential instructions between MC and MCE".(Note 1)
Other instructions	Not executed.

(Note 1) The following items are included in differential instructions.

1) DF (leading edge differential) instruction

2) Count input for CT (counter) instruction

3) Count input for UDC (up-down counter) instruction

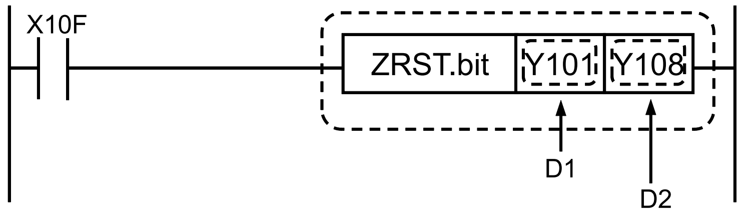
4) Shift input for SR (shift register) instruction

5) Shift input for LRSR (left and right shift register) instruction

6) Differential execution type high-level instruction (instruction specified by p and instruction name)

3.33 ZRST (Clear Multiple Processes)

■ Ladder diagram



■ Available operation units (●: Available)

Operatio n unit	bit	US	SS	UL	SL	SF	DF
i	●						

■ List of operands

Operand	Description
D1	Process clear start number
D2	Process clear end number

■ Devices that can be specified (indicated by ●)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
D1	●	●	●	●							●	●	●	●
D2	●	●	●	●							●	●	●	●

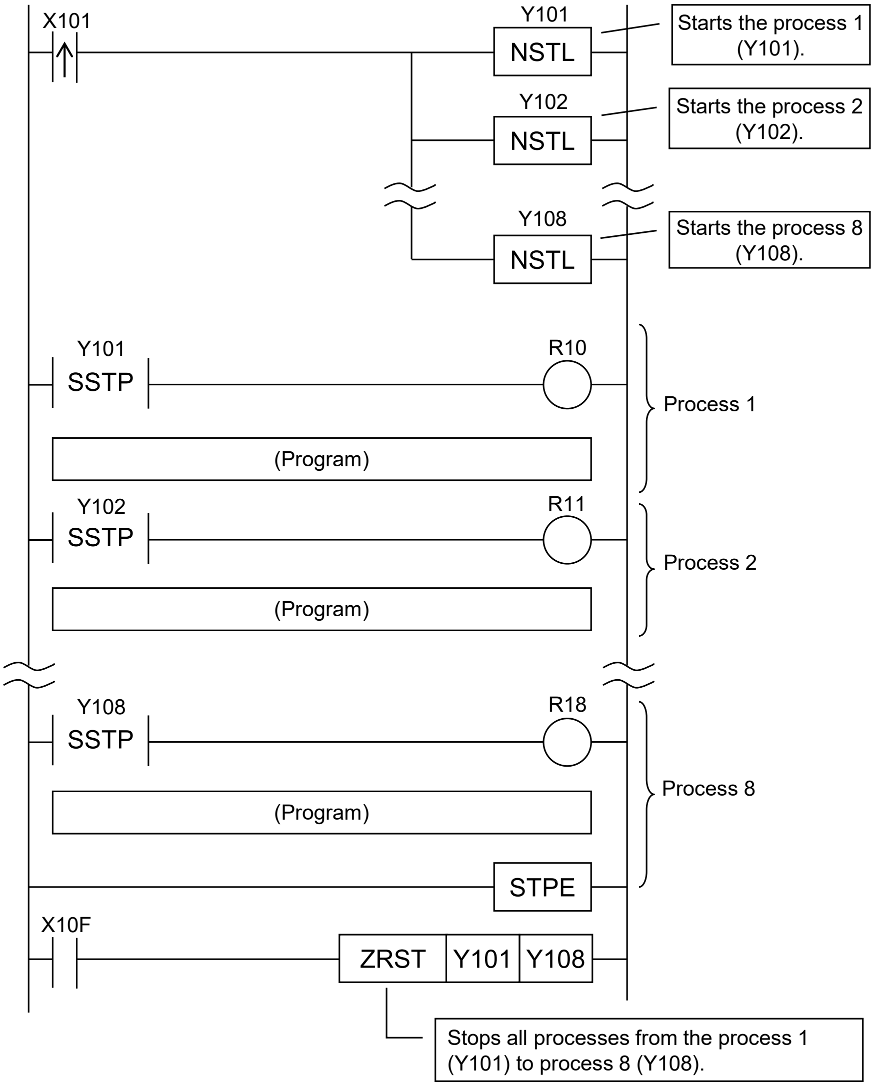
■ Outline of operation

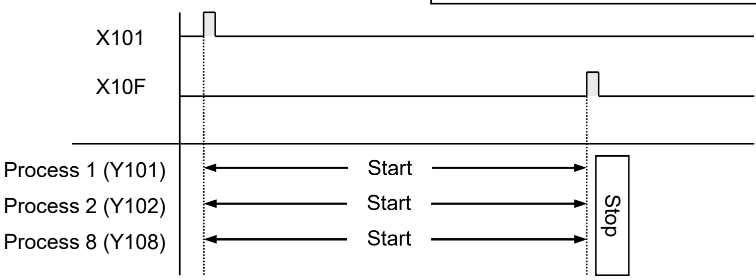
- When the ZRST instruction is executed, all the running processes within the range from the process [D1] and [D2] will be cleared.
- It can also be used to reset (clear to zero) the range from the area (bit address) specified in [D1] to the area (bit address) specified in [D2].

■ Precautions for programming

- Be sure that [D1] is smaller than [D2].
- This instruction can be executed from the normal ladder area as well as a running process.

■ Operation Example

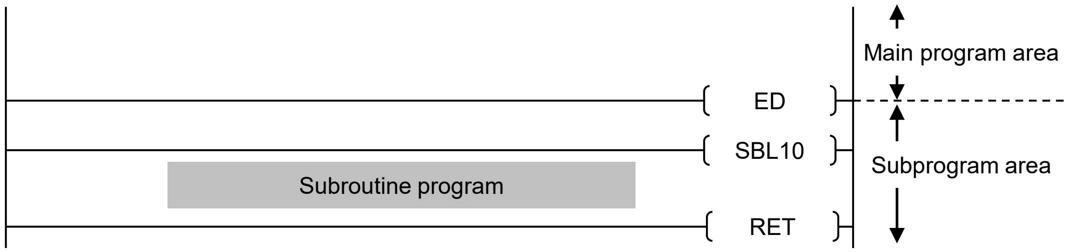




3.34 Common Information for Subroutine Instructions

■ Program configuration

- Subroutine instructions are configured from a subroutine call instruction and a subroutine program.
- Describe subroutine programs in subprogram areas, and use the SBL instruction and RET instruction to indicate the start and end positions of the subroutine program.
- Subroutine call instructions can be described in either the main program area or a subprogram area (another subroutine program).

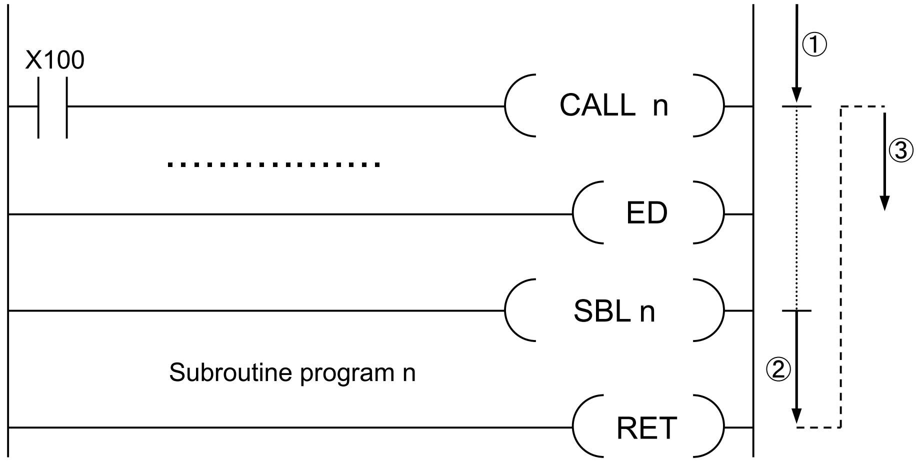


■ Types of call instructions

- There are four types of subroutine call instructions as follows:
  CALL (local subroutine)
  FCALL (output OFF type local subroutine call)
  ECALL (subroutine call (with PB number specification))
  EFCALL (forced output OFF type subroutine call (with PB number specification))
- Local subroutine call instructions (CALL, FCALL) start subroutine programs within the same PB.
- Subroutine call instructions (ECALL, EFCALL) start subroutine programs for a specified PB number.

■ Flow of the program process

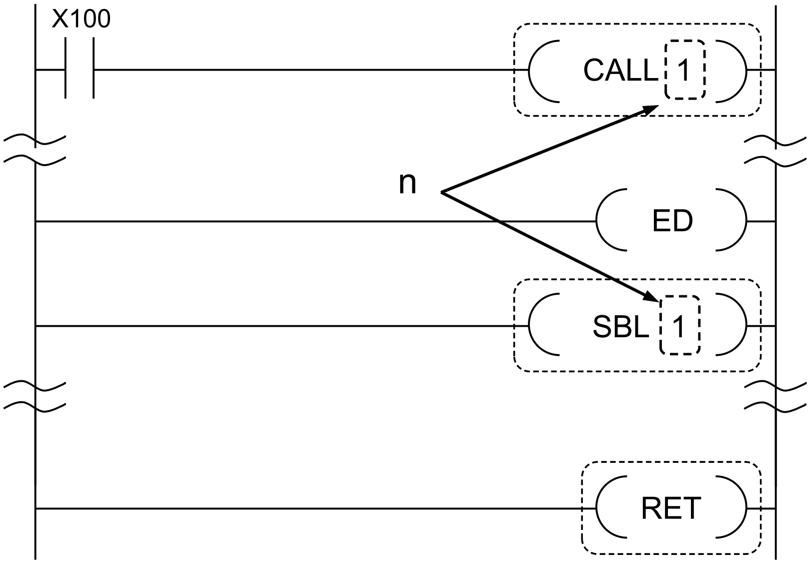
- When the execution condition is ON, the subroutine call instruction is executed to start the subroutine program beginning with the SBL instruction with the specified number. When the execution condition is OFF, nothing occurs.
- When the subroutine program is processed up to the RET instruction, the program will return to the address next to the subroutine call instruction, and continue with processing of the program.



When CALL n is executed, ① to ③ are executed in order

3.35 SBL (Subroutine Label), RET (Subroutine Return)

■ Ladder diagram



■ List of operands

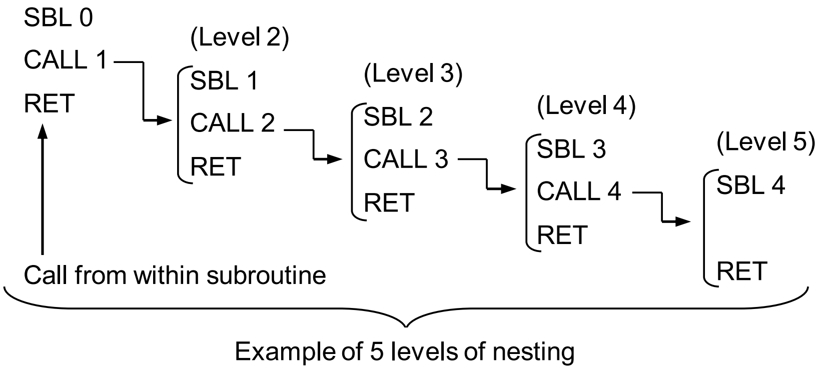
Operand	Description
n	Subroutine program number Available data specification range: 0 to 65535/1 PB (It is recommended to specify sequentially from 0.)

■ Outline of operation

- When the execution condition is ON, the subroutine call instruction is executed to start the subroutine program beginning with the SBL instruction with the specified number.

■ Precautions for programming

- "Subroutine program n" represents the program code between the SBL n instruction and the RET instruction. Be sure to write it to an address which follows the ED instruction.
- In the SBL instruction, specify the values of "n" sequentially from 0.
- Subroutines can be nested up to 16 layers deep.


SBL 0
CALL 1
RET
(Level 2)
SBL 1
CALL 2
RET
(Level 3)
SBL 2
CALL 3
RET
(Level 4)
SBL 3
CALL 4
RET
(Level 5)
SBL 4
RET
Call from within subroutine
Example of 5 levels of nesting

3.36 CALL (Local Subroutine Call), FCALL (Output OFF Type Local Subroutine Call)

■ Ladder diagram

■ List of operands

Operand	Description
n	Local subroutine program number within the same PB Available data specification range: 0 to 65535 (It is recommended to specify sequentially from 0.)

■ Outline of operation

- When the execution condition is ON, the CALL/FCALL instruction is executed to start the local subroutine program beginning with the SBL instruction with the specified number.
- When the subroutine program is processed up to the RET instruction, the program will return to the address next to the CALL/FCALL instruction in the main program, and continue with processing of the main program.

■ Operation when execution condition of CALL/FCALL instruction is OFF

- When the execution condition turns OFF, the operation of the current subroutine stops. (It is also true for calls from the master control or step ladder.) In this case, the operation of each instruction used in the subroutine is as follows.

<table>
<tr><th>Type of instruction</th><th>CALL</th><th>FCALL</th></tr>
<tr><td>OT</td><td>Holds the state</td><td>All OFF. Different operation from the CALL instruction.</td></tr>
<tr><td>KP</td><td rowspan="3">Holds the state.</td><td rowspan="3">Same as on the left.</td></tr>
<tr><td>SET</td></tr>
<tr><td>RST</td></tr>
<tr><td>TM</td><td>Clocking is not performed.
Note that time is not guaranteed if counting does not occur during a single scan.</td><td>Reset. Different operation from the CALL instruction.</td></tr>
<tr><td>CT</td><td>Holds the current progress.</td><td>Same as on the left.</td></tr>
</table>

Type of instruction	CALL	FCALL
SR		
Differential instruction	Operates in the same way as differential instructions used between MC and MCE. Refer to "Operation of differential instructions between MC and MCE".	Same as on the left.
Other instructions	Not executed.	Same as on the left.

(Note 1) The following items are included in differential instructions.

1) DF (leading edge differential) instruction
2) Count input for CT (counter) instruction
3) Count input for UDC (up-down counter) instruction
4) Shift input for SR (shift register) instruction
5) Shift input for LRSR (left and right shift register) instruction
6) Differential execution type high-level instruction (instruction specified by p and instruction name)

■ Precautions for programming

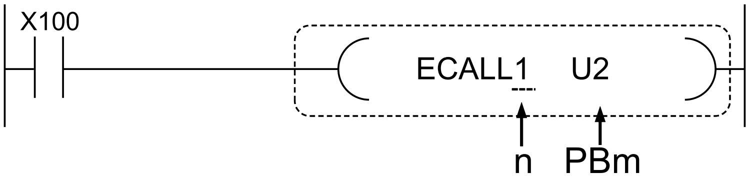
- The CALL/FCALL instruction can be written in another subroutine program or step ladder, in addition to the main program. The CALL/FCALL instruction with the same number can be written repeatedly.
- Note that if a subroutine is executed repeatedly, it will take more time for operations to be processed.

■ Flag operations

Name	Description
SR7 SR8 (ER)	Turns ON when the 16th subroutine executes the CALL/FCALL instruction while subroutines are nested in 16 levels.

3.37 ECALL (Subroutine Call (with PB No. Specification))

■ Ladder diagram



■ List of operands

Operand	Description
n	Subroutine number: 0 to 65535/1 PB
PBm	Target PB number: The number of PB where the subroutine specified by n is stored

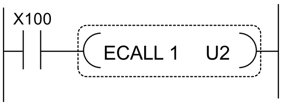
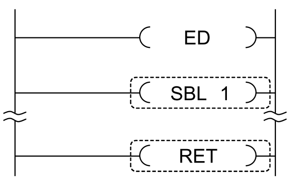
■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
PBm	●	●	●	●			●	●								●					

■ Outline of operation

- When the execution condition is ON, the SBLn subroutine of PBm is called.
- When the subroutine program is processed up to the "RET" instruction, the program will return to the address next to the "ECALL" instruction in the main program, and continue with processing of the main program.
- The local device of the called PBm is used as the local device in the called subroutine.

■ Operation Example

PB	PB1 program	PB2 program	Description
Program example			● Instructions between SBL and RET are called and executed when the ECALL instruction is executed. ● When RET is executed, program returns to the calling ECALL instruction.

■ Operation when the execution condition of the ECALL instruction is OFF

- When the execution condition turns OFF, the operation of the current subroutine stops. (It is also true for calls from the master control or step ladder.) In this case, the operation of each instruction used in the subroutine is as follows.

Type of instruction	ECALL
OT	Holds the state
KP	
SET	
RST	
TM	Clocking is not performed. Note that the time cannot be guaranteed if clocking is not performed once during a scan.
CT	Holds the current progress.
SR	
Differential instruction	Operates in the same way as differential instructions used between MC and MCE. Refer to "Operation of differential instructions between MC and MCE".
Other instructions	Not executed.

(Note 1) The following items are included in differential instructions.

1) DF (leading edge differential) instruction
2) Count input for CT (counter) instruction
3) Count input for UDC (up-down counter) instruction
4) Shift input for SR (shift register) instruction
5) Shift input for LRSR (left and right shift register) instruction
6) Differential execution type high-level instruction (instruction specified by p and instruction name)

■ Precautions for programming

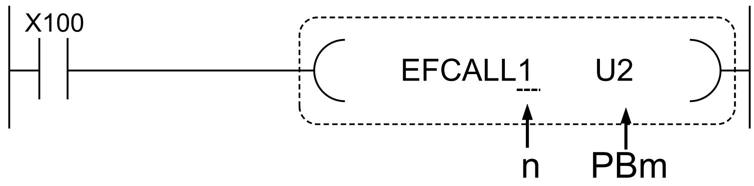
- The "ECALL" instruction can be written in another subroutine program or step ladder, in addition to the main program. The "ECALL" instruction with the same number can be written repeatedly.
- Note that if a subroutine is executed repeatedly, it will take more time for operations to be processed.

■ Flag operations

Name	Description
SR7 SR8 (ER)	Turns ON when the 16th subroutine executes the "ECALL" instruction while subroutines are nested in 16 levels.

3.38 EFCALL (Forced Output OFF Type Subroutine Call (with PB No. Specification))

■ Ladder diagram



■ List of operands

Operand	Description
n	Subroutine number: 0 to 65535/1 PB
PBm	Target PB number: The number of PB where the subroutine specified by n is stored

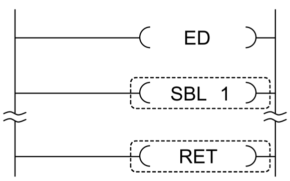
■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
PBm	•	•	•	•			•	•								•					

■ Outline of operation

- When the execution condition is ON, the SBLn subroutine of PBm is called.
- When the subroutine program is processed up to the "RET" instruction, the program will return to the address next to the "EFCALL" instruction in the main program, and continue with processing of the main program.
- The local device of the called PBm is used as the local device in the called subroutine.

■ Operation Example

PB	PB1 program	PB2 program	Description
Program example			• Instructions between SBL and RET are called and executed when the EFCALL instruction is executed. • When RET is executed, program returns to the calling EFCALL instruction.

■ Operation when the execution condition of the EFCALL instruction is OFF

- When the execution condition turns OFF, the operation of the current subroutine stops. (It is also true for calls from the master control or step ladder.) In this case, the operation of each instruction used in the subroutine is as follows.

Type of instruction	EFCALL
OT	All OFF. Different operation from the ECALL instruction.
KP	Holds the state.
SET	
RST	
TM	Reset. Different operation from the ECALL instruction.
CT	Holds the current progress.
SR	
Differential instruction	Operates in the same way as differential instructions used between MC and MCE. Refer to "Operation of differential instructions between MC and MCE".
Other instructions	Not executed.

(Note 1) The following items are included in differential instructions.

1) DF (leading edge differential) instruction
2) Count input for CT (counter) instruction
3) Count input for UDC (up-down counter) instruction
4) Shift input for SR (shift register) instruction
5) Shift input for LRSR (left and right shift register) instruction
6) Differential execution type high-level instruction (instruction specified by p and instruction name)

■ Precautions for programming

- The "EFCALL" instruction can be written in another subroutine program or step ladder, in addition to the main program. The "EFCALL" instruction with the same number can be written repeatedly.
- Note that if a subroutine is executed repeatedly, it will take more time for operations to be processed.

■ Flag operations

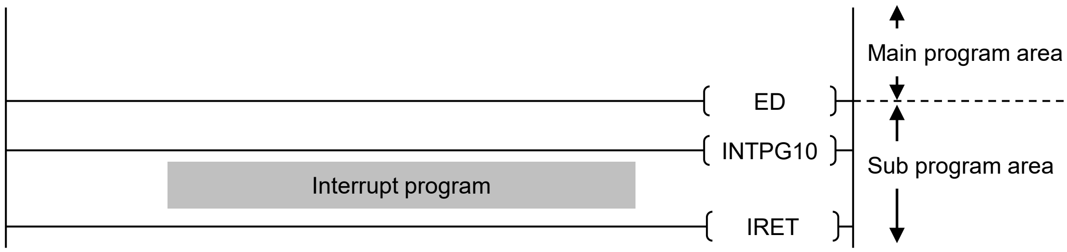
Name	Description
SR7 SR8 (ER)	Turns ON when the 16th subroutine executes the "EFCALL" instruction while subroutines are nested in 16 levels.

3.39 INTPG (Unit Interruption Program Start), IRET (Unit Interruption Program End)

■ Ladder diagram

■ Outline of operation

- These instructions are described in subprogram areas in the same PB to indicate the start and end positions of interruption program.
- Activates the interruption program of a corresponding program number when the unit's interruption condition is met.
- Returns to the main program by executing the IRET instruction.
- To execute an interruption program, it is necessary to interrupt the CPU by the EI instruction and enable a unit to interrupt by the IMASK instruction.
- The interruption activation request signal on the unit side will be held until the corresponding interruption program is executed or the unit interruption clear instruction "ICLR" instruction is executed.



■ Specification of interruption program number [n]

- Interruption program number n is specified in decimal by the combination of a slot number (1 to 16) and a bit number (0 to 7).
- The allocation of the last one digit varies depending on units.
- The interruption program numbers for a high-speed counter unit and multiple input/output unit are as shown below.

Comparison match flag of unit	Corresponding interruption program No.						
	Slot 1	Slot 2	Slot 3	-----	-----	Slot 15	Slot 16
CH0 Comparison match 0 flag	INTPG 10	INTPG 20	INTPG 30	-----	-----	INTPG 150	INTPG 160
CH0 Comparison match 1 flag	INTPG 11	INTPG 21	INTPG 31	-----	-----	INTPG 151	INTPG 161

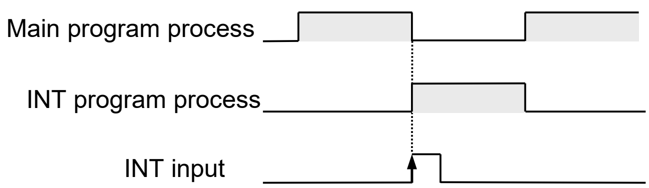
Comparison match flag of unit	Corresponding interruption program No.						
	Slot 1	Slot 2	Slot 3	-----	-----	Slot 15	Slot 16
CH1 Comparison match 0 flag	INTPG 12	INTPG 22	INTPG 32	-----	-----	INTPG 152	INTPG 162
CH1 Comparison match 1 flag	INTPG 13	INTPG 23	INTPG 33	-----	-----	INTPG 153	INTPG 163
CH2 Comparison match 0 flag	INTPG 14	INTPG 24	INTPG 34	-----	-----	INTPG 154	INTPG 164
CH2 Comparison match 1 flag	INTPG 15	INTPG 25	INTPG 35	-----	-----	INTPG 155	INTPG 165
CH3 Comparison match 0 flag	INTPG 16	INTPG 26	INTPG 36	-----	-----	INTPG 156	INTPG 166
CH3 Comparison match 1 flag	INTPG 17	INTPG 27	INTPG 37	-----	-----	INTPG 157	INTPG 167

(Note 1) Interruption program numbers are specified with slot numbers + (0 to 7). The numbers in the above table are for the slot 1.

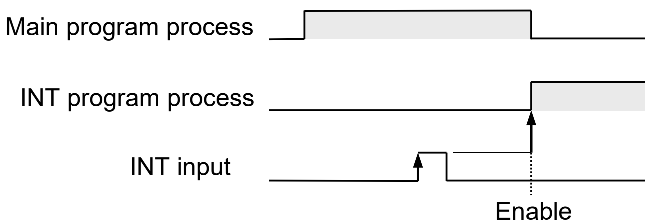
Example) The interruption program number corresponding to the CH1 comparison match 1 flag of the slot number 10 is INTPG103.

■ Interrupt program execution

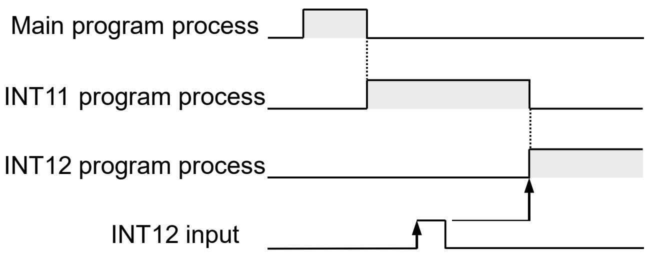
- Executes the interruption program of a corresponding number when interruption occurs.



- When interruption is disabled, the interruption program will be executed by enabling it by the CPU unit interruption enable instruction "EI" and the unit interruption enable/disable instruction "IMASK".

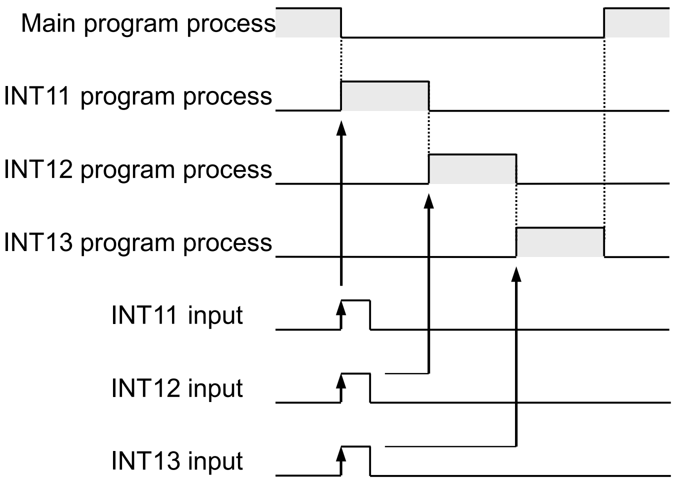


- If another interruption program is being executed, executes it after the completion of the running program.

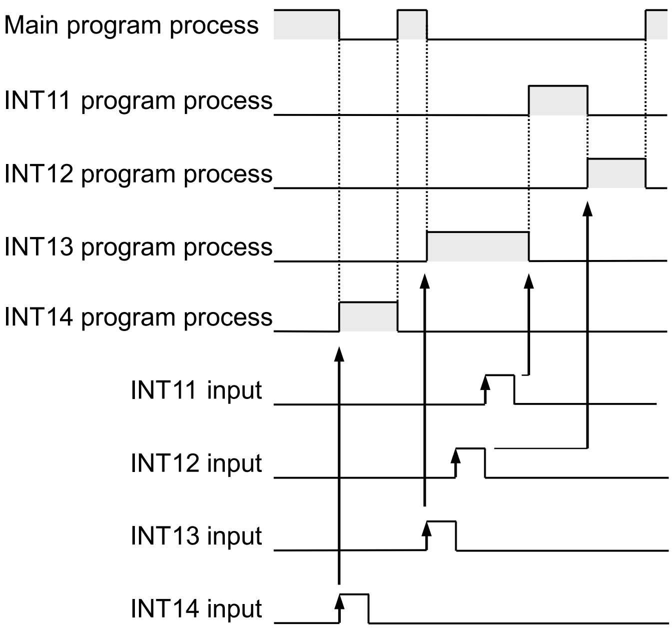


■ Control when multiple interrupts occur simultaneously

- The priority order when multiple interruptions have occurred simultaneously is as follows:
  Unit interruption: INTPG0 > 1 > 2 > ... > 7 > Fixed cycle execution type PB
- If more than one interruption activation request is made from the unit, the process will be carried out from the smallest slot number or the smallest interruption program number.
- If the interruption activation is requested on the completion of the process of interruption program, a higher-priority program will be searched again and the corresponding interruption program will be executed.
- The interruption activation request signal on the unit side will be held until the corresponding interruption program is executed or the unit interruption clear instruction "ICLR" instruction is executed.

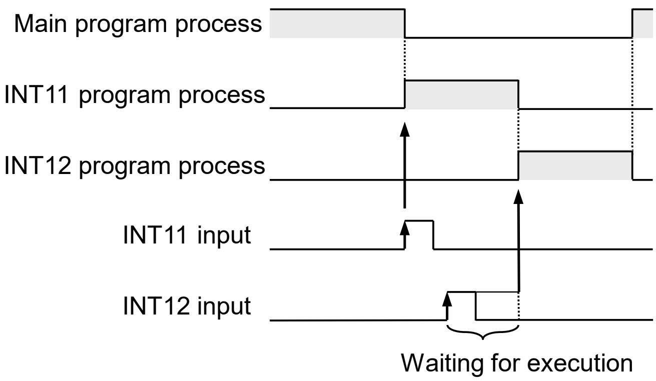


- When multiple interrupts occur during execution of an interrupt program, they will be executed in order from the smallest program number when the program has finished execution.

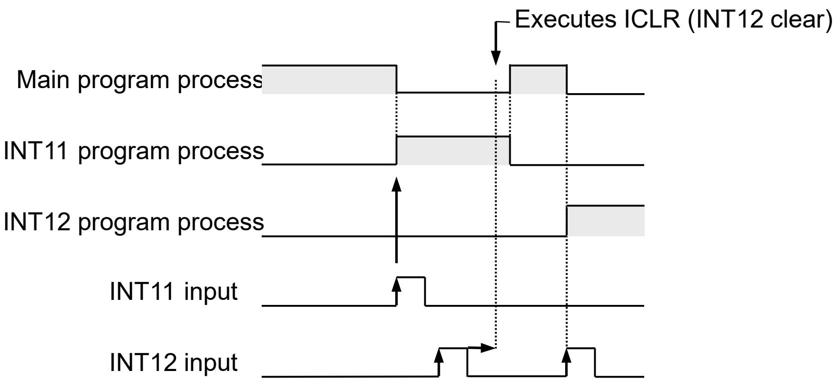


■ Interrupt program execution waiting and clearing

- If multiple interruptions occur simultaneously, or a new interruption occurs during the execution of another interruption program, lower-priority interruptions will be in "waiting" state. They will be executed in order after the completion of other interruption programs.

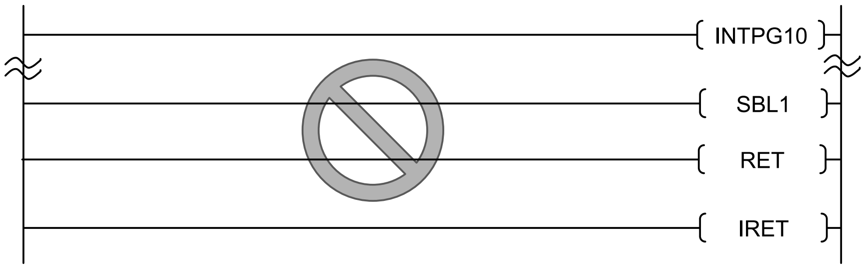


- If placed in execution wait state, there is a time difference between the occurrence of the interrupt and execution of the interrupt program. In such a case, the interruption program in the waiting state can be cleared by the "ICLR" instruction as necessary. Cleared interrupt programs will not be executed.

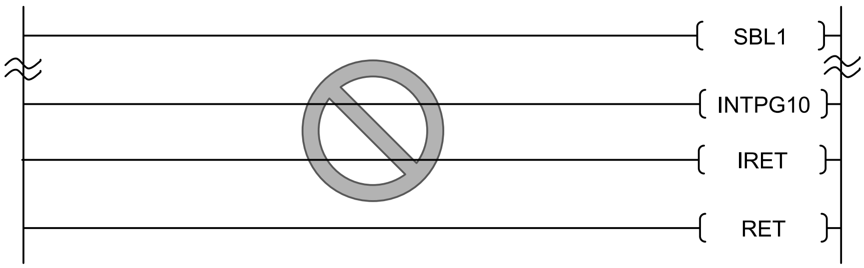


■ Precautions for programming

- Always use the INTPG n and IRET instructions in combination. A syntax error occurs if either "INTPG" instruction or "IRET" instruction is not used.
- More than one INTPG instruction with the same number cannot be specified.
- If a unit in which interruption occurs is not installed in a specified slot number, the no target unit error 10 is displayed and the mode cannot be switched to RUN.
- Branching from the interruption program area (between INTPG and IRET) to other subprogram areas or main program area is not possible.
- A subroutine program cannot be used in an interrupt program.



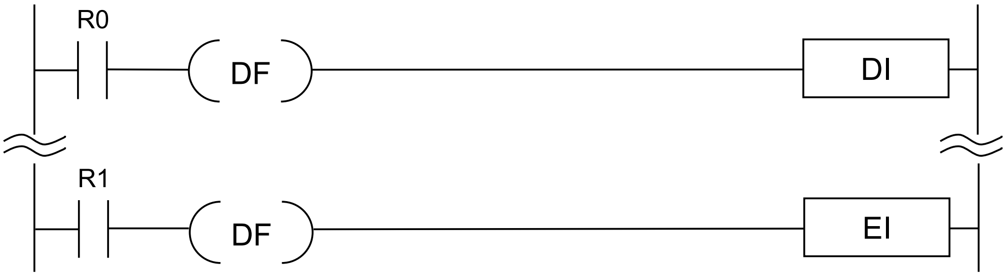
- An interrupt program cannot be used in a subroutine program.



- It is operable even if one IRET instruction is used for multiple interruption programs INTPG.

3.40 DI (CPU Interruption Disable), EI (CPU Interruption Enable)

■ Ladder diagram



■ Outline of operation

DI

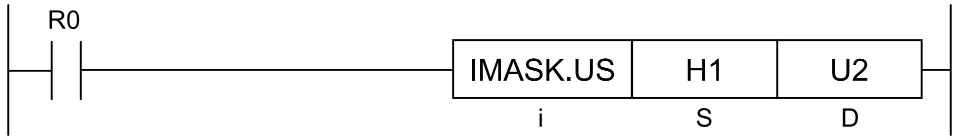
- This instruction disables all interruption programs INTPG and the acceptance of interruption of fixed cycle execution type PB at the same time as the execution of the instruction.
  At this time, the unit in which interruption has occurred suspends the detected interruption.
- Use the ICLR instruction to clear the interruption signals suspended by the unit while the interruption is disabled.
- To disable or enable the interruption for each unit, a unit's interruption detection function can be controlled using the IMASK instruction.

EI

- This instruction enables all permitted interruptions.
- Restarts a fixed cycle execution type PB. Starts the execution of the PB after the elapse of the interval specified after the startup.
- Also accepts the interruption permitted in the unit in which interruption occurs.
- As interruption for the unit in which interruption occurs is disabled after RUN, it is necessary to enable the interruption of the unit by executing the IMASK instruction.
- It is invalid even if the EI instruction is executed while the interruption is enabled. The interruption will stay enabled.

3.41 IMASK (Unit Interruption Disable/Enable Setting)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●					

■ List of operands

Operand	Description
S	Control data specifying INTPG number to disable/enable the unit interruption: H0 to HFF
D	Slot number (U constant) or device number where slot number is stored

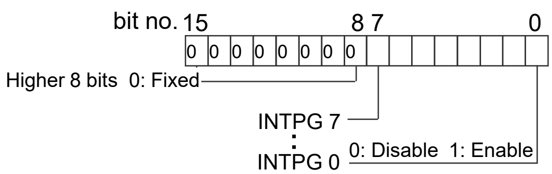
■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●	●	●					●	●				●

■ Outline of operation

This instruction sets to enable or disable the interruption of the unit installed in the slot specified by [D] according to the data specified by [S].

■ Specification of [S]



■ Precautions for programming

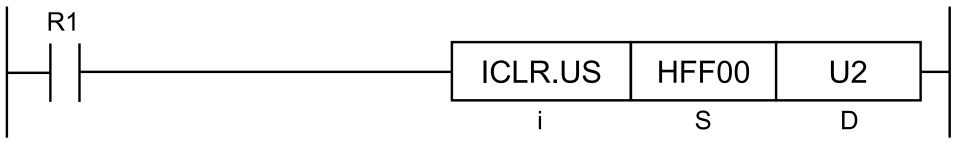
- To enable the interruption of the unit, the interruption to the CPU unit must be enabled using the EI instruction.
- When a unit in which interruption occurs is not installed in the specified slot, an operation error occurs.
- When there is no definition of the interruption program (INTPG) corresponding to an enabled bit, the bit is not enabled. Zero is written.

■ Flag operations

Name	Description
SR7 SR8 (ER)	ON in the case of out-of-range in indirect access (index modification)

3.42 ICLR (Unit Interruption Clear)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●					

■ List of operands

Operand	Description
S	Control data specifying INTPG number to clear interruption: HFF00 to HFFFF
D	Slot number (U constant) or device number where slot number is stored

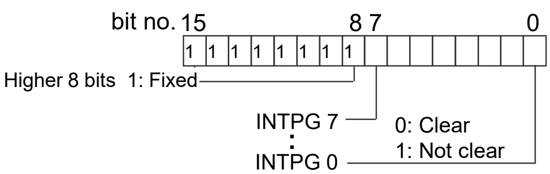
■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●	●	●					●	●				●

■ Outline of operation

- This instruction clears the interruption of the unit installed in the slot specified by [D] according to the data specified by [S].

■ Specification of [S]



■ Precautions for programming

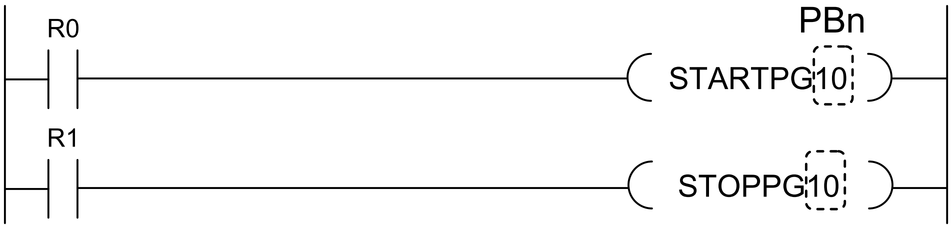
- When a unit in which interruption occurs is not installed in the specified slot, an operation error occurs.
- The suspension of the interruption of a bit on which the interruption is disabled is also cleared.

■ Flag operations

Name	Description
SR7 SR8 (ER)	ON in the case of out-of-range in indirect access (index modification)

3.43 STARTPG (PBn Execution Start), STOPPG (PBn Execution Stop)

■ Ladder diagram



■ List of operands

Operand	Description
PBn	Target PB number (1 to 468)

(Note 1) The maximum value for the PB number varies depending on the type of CPU unit and the settings for the program memory.

■ Outline of operation

- This instruction activates the waiting PBn when the execution condition of STARTPG PBn turns ON.
- This instruction makes the active PBn be in waiting state when the execution condition of STOPPG PBn turns ON.
- When switching PROG to RUN, operates in the default mode (start or wait). The start mode of PB is specified using the menu of tool software FPWIN GR7 for **Creating new program block (PB)** or **Changing PB attribute.** The default is start.
- At the time of rewriting during RUN, holds the PB start mode and does not reset it to the default setting.
- A PB that is in the waiting state after startup clears the output in a single scan with the program block active relay OFF in order to change the state of each input and output relay to the state as shown below.

Type of instruction	Operation
OT	All OFF.
KP	Holds the state
SET	
RST	
TM	Reset.
CT	Holds the current progress.
SR	
Differential instruction	Operates in the same way as differential instructions used between MC and MCE. Refer to "Operation of differential instructions between MC and MCE".
Other instructions	Not executed.

Operation of system relays (SR)

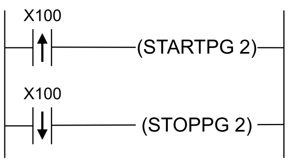
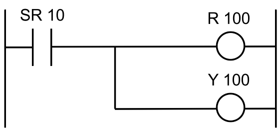
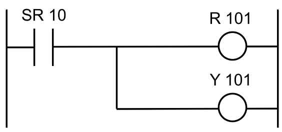
SR state		Waiting	Active 1st scan	Active 2nd and later scans	Standby process 1st scan (Note 1)	Waiting
SR16	ON at the start of PBn execution. OFF from the next scan.	-(Note 2)	ON	OFF	OFF	-(Note 2)
SR17	OFF at the start of PBn execution. ON from the next scan.	-(Note 2)	OFF	ON	OFF	-(Note 2)
SR1000 to SR1499	Program block active relay	OFF	ON	ON	OFF	OFF

(Note 1) Indicates the state during the first scan after the active state is changed to the waiting state.

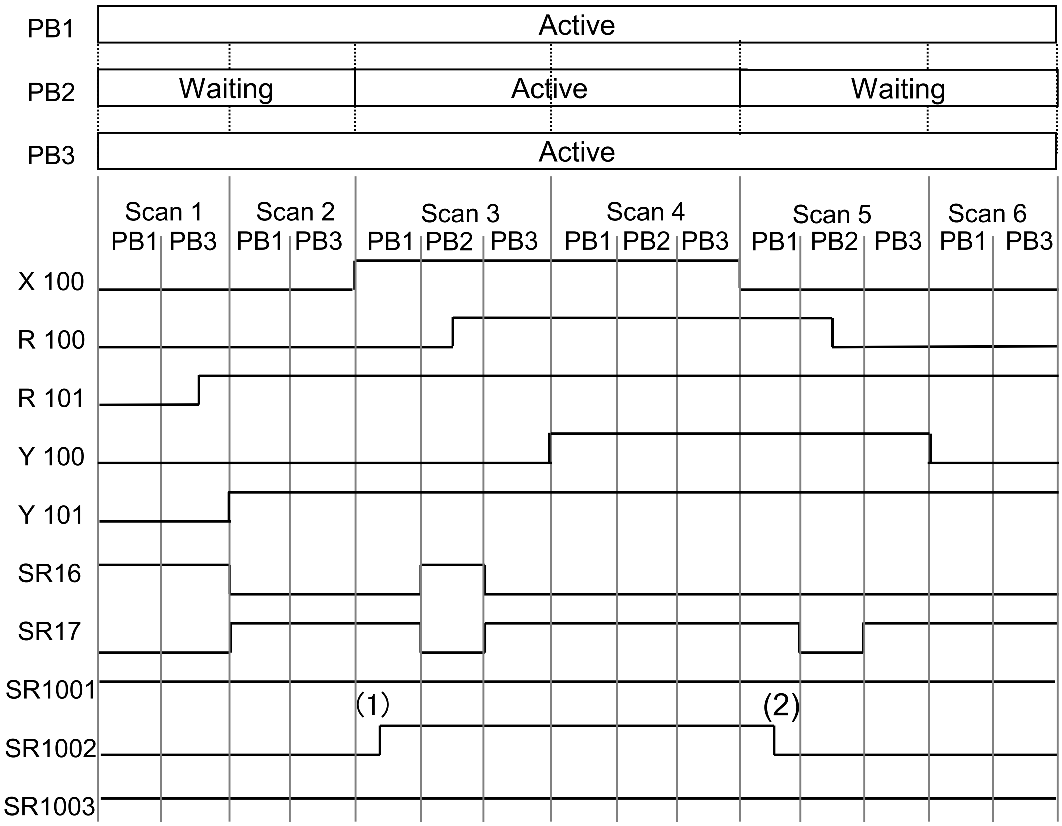
(Note 2) SR16 and SR17 indicates the states of other PBs because the target PBs are not active.

Sample program

Operation when the initial states are as follows: PB1: Active, PB2: Waiting, PB3: Active

PB	PB1 program	PB2 program	PB3 program
Program example			

■ Timing chart



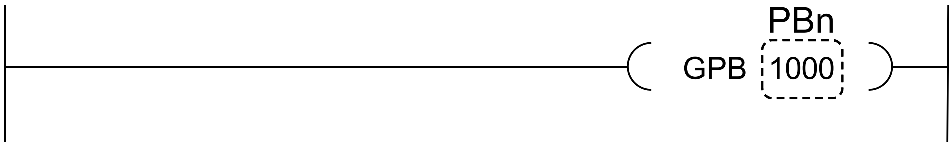
(Note 1) SR1001 to SR1003: Program block active relays of PB1 to PB3

(Note 2) (1) PB1 executes STARTPG instruction, (2) PB1 executes STOPPG instruction.

(Note 3) PB2 clears the output in the 1st scan after the state is changed to waiting. (Program block active relay is OFF.)

3.44 GPB (Global PB Number Setting)

■ Ladder diagram



■ List of operands

Operand	Description
PBn	Global PB number (1000 to 1999)

■ Outline of operation

- A fixed PB number can be used for the global PB number, even if the PB position is changed.
- This instruction should be described at the beginning of a main program area.
- This instruction sets an active PB number to a specified global PB number table in the preprocessing of operation (when switching PROG to RUN, during rewriting during RUN).
- When multiple GPB instructions are described in the same PB, multiple global PB numbers are set for one PB.
- The range of PB numbers that can be specified for instructions that use PB numbers (ECALL, EFCALL, LCWT, LC, STARTPG, and STOPPG) is as follows:

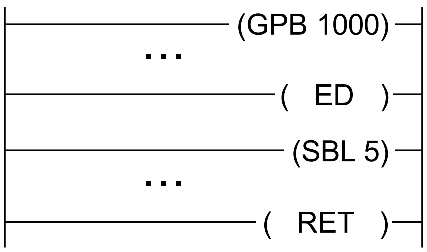
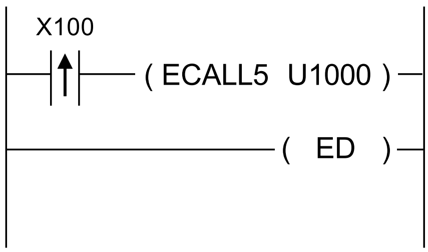
  PB number specification: 1 to 468

  Global PB number specification: 1000 to 1999
- A syntax error occurs when a number (other than 1000 to 1999) that is out of the range of the settable global PB numbers is specified by the GPB instruction.
- An operation error occurs when a global PB number that is not set is specified for an instruction that uses a PB number.

■ Sample program

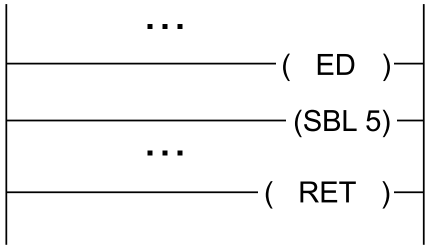
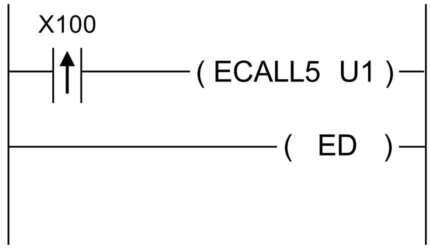
Example 1) Example when using GPB

PB1 declares the global PB number 1000 for own PB, and calls the subroutine for GPB1000 from PB2.

PB	PB1 program	PB2 program
Program example		

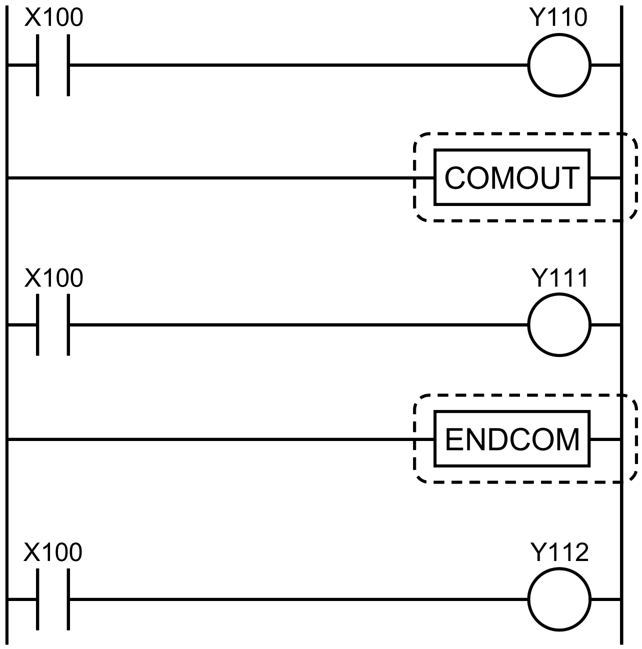
Example 2) Example when not using GPB

Calls the subroutine for PB1 from PB2.

PB	PB1 program	PB2 program
Program example		

3.45 COMOUT (Comment Out) / ENDCOM(Comment Out End)

■ Ladder diagram



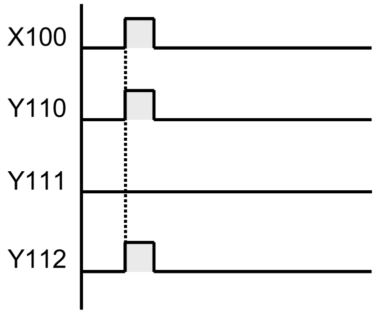
■ Outline of operation

This instruction comments out text between the COMOUT and ENDCOM instructions.

■ Input method

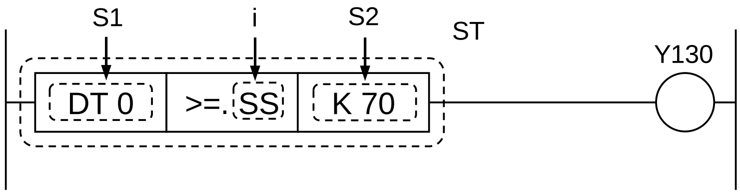
On the program edit screen, select the network you want to comment out, right-click on it and select "Target/Detarget This Network For Conversion" from the menu displayed.
This instruction cannot be directly input.

■ Operation Example



3.46 ST=, ST<>, ST>, ST>=, ST<, ST<= (Data Comparison: Start)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Comparison data 1
S2	Comparison data 2

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

3.46 ST=, ST<>, ST>, ST>=, ST<, ST<= (Data Comparison: Start)

■ Outline of operation

Compares signed data specified in [S1] with signed data specified in [S2].

Begins a logic operation as a contact connected when the comparison result is in the specified state (such as =, <, or >).

■ Comparison result and operation

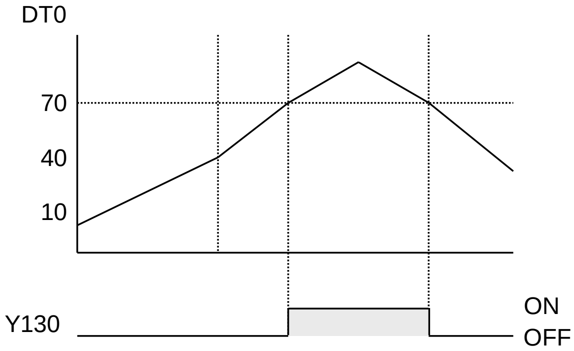
Relationship between [S1] and [S2]		[S1] < [S2]	[S1] = [S2]	[S1] > [S2]
Comparison instruction	ST =	OFF	ON	OFF
	ST <>	ON	OFF	ON
	ST >	OFF	OFF	ON
	ST >=	OFF	ON	ON
	ST <	ON	OFF	OFF
	ST <=	ON	ON	OFF

(Note 1)
- "< >" represents "≠".
- ">=" represents "≥".
- "<=" represents "≤".

■ Operation Example

Program operation of "ST >=" in the ladder diagram

Compares the value of data register DT0 with K70. If DT0 = K70, external output Y130 turns ON.



■ Precautions for use

- The "ST=", "ST<>", "ST>", "ST>=", "ST<", and "ST<=" instructions are initiated from the bus bar.
- Since BCD data is assumed to be a negative value during comparison if the most significant bit is 1, the comparison result may become incorrect. In such a case, use the BIN instruction to convert the data into binary before comparison.

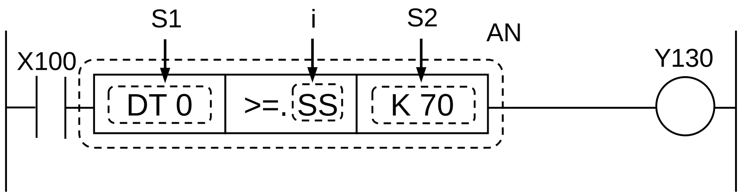
■ Flag operations

Name	Description
SR7	ON if the specified address using the index modification exceeds a limit.

Name	Description
SR8 (ER)	

3.47 AN=, AN<>, AN>, AN>=, AN<, AN<= (Data Comparison: AND)

■ Ladder diagram



■ Available operation units (●: Available)

Operatio n unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Comparison data 1
S2	Comparison data 2

■ Devices that can be specified (indicated by ●)

Operan d	16-Bit device:											32-Bit device: (Note 1)			Integer			Real numbe r		St rin g	Index modifie r (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

Compares signed data specified in [S1] with signed data specified in [S2].

Connects in serial as a contact connected when the comparison result is in the specified state (such as =, <, or >).

■ Comparison result and operation

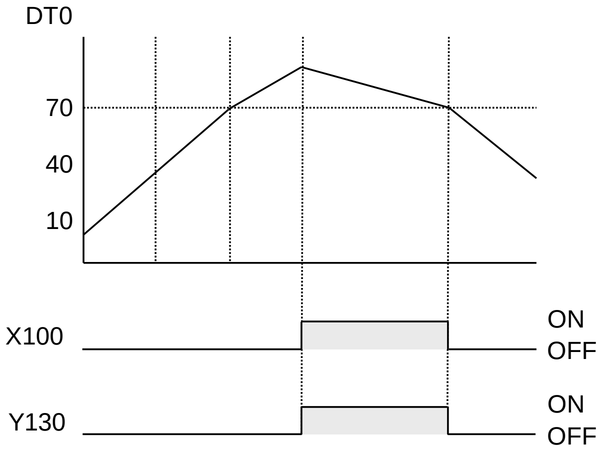
Relationship between [S1] and [S2]		[S1] < [S2]	[S1] = [S2]	[S1] > [S2]
Comparison instruction	AN=	OFF	ON	OFF
	AN<>	ON	OFF	ON
	AN>	OFF	OFF	ON
	AN>=	OFF	ON	ON
	AN<	ON	OFF	OFF
	AN<=	ON	ON	OFF

(Note 1)
- "< >" represents "≠".
- ">=" represents "≥".
- "<=" represents "≤".

■ Operation Example

Program operation of "AN >=" in the ladder diagram

When external output X100 turns ON, the value of DT0 and K70 are compared, and if DT0 is greater than K70, the external output Y130 turns ON.



■ Precautions for use

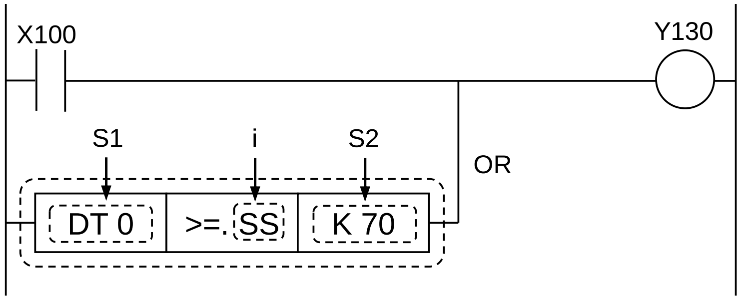
- The "AN=", "AN<>", "AN>", "AN>=", "AN<" and "AN<=" instructions can be used in series.
- Since BCD data is assumed to be a negative value during comparison if the most significant bit is 1, the comparison result may become incorrect. In such a case, use the BIN instruction to convert the data into binary before comparison.

■ Flag operations

Name	Description
SR7 SR8 (ER)	ON if the specified address using the index modification exceeds a limit.

3.48 OR=, OR<>, OR>, OR>=, OR<, OR<= (Data Comparison (OR))

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Comparison data 1
S2	Comparison data 2

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

Compares signed data specified in [S1] with signed data specified in [S2].

Connects in parallel as a contact connected when the comparison result is in the specified state (such as =, <, or >).

■ Comparison result and operation

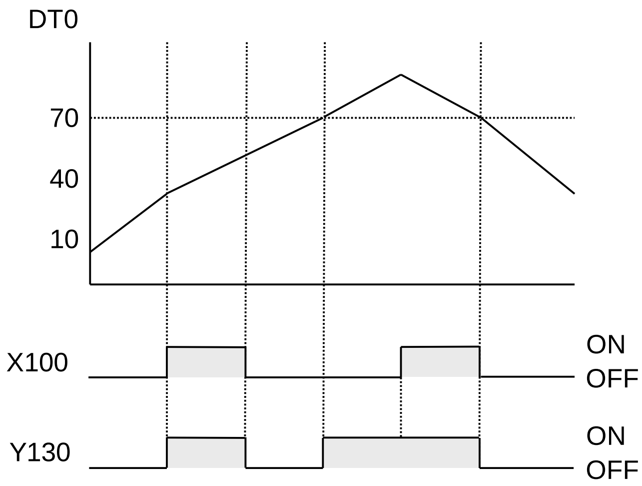
Relationship between [S1] and [S2]		[S1] < [S2]	[S1] = [S2]	[S1] > [S2]
Comparison instruction	OR=	OFF	ON	OFF
	OR<>	ON	OFF	ON
	OR>	OFF	OFF	ON
	OR>=	OFF	ON	ON
	OR<	ON	OFF	OFF
	OR<=	ON	ON	OFF

(Note 1)
- "< >" represents "≠".
- ">=" represents "≥".
- "<=" represents "≤".

■ Operation Example

Program operation of "OR >=" in the ladder diagram

When external output X100 turns ON, or the result of comparison between the value of DT0 and K70 is DT0 > K70, external output Y130 turns ON.



■ Precautions for use

- The "OR=", "OR<>", "OR>", "OR>=", "OR<", and "OR<=" instructions are initiated from the bus bar.
- The "OR=", "OR<>", "OR>", "OR>=", "OR<" and "OR<=" instructions can be used in series.

- Since BCD data is assumed to be a negative value during comparison if the most significant bit is 1, the comparison result may become incorrect. In such a case, use the BIN instruction to convert the data into binary before comparison.

■ Flag operations

Name	Description
SR7 SR8 (ER)	ON if the specified address using the index modification exceeds a limit.

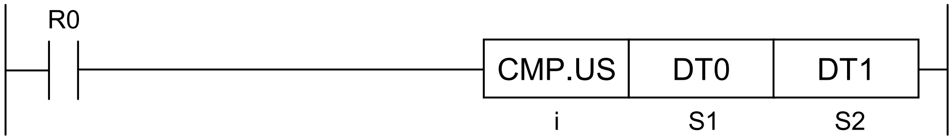
(MEMO)

4 High-level Instructions(Data Comparison)

Applicable Models: All Models

4.1 CMP (Data Compare)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Comparison data 1 (device address or constant)
S2	Comparison data 2 (device address or constant)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

Outline of operation

- According to the operation unit [i], two pieces of data that are stored in two different areas that start from [S1] and [S2] respectively are compared.
- The comparison result is output to the system relays SRA to SRC (assessment flags for comparison flags).

Processing

- Depending on the relationship between [S1] and [S2], the result of the comparison flag (SR (system relay)) is as follows:

Relationship between [S1] and [S2]	Comparison flags (SR (system relays))		
	SRA	SRB	SRC
	>	=	<
[S1] < [S2]	OFF	OFF	ON
[S1] = [S2]	OFF	ON	OFF
[S1] > [S2]	ON	OFF	OFF

Example 1) Operation unit: Unsigned 16 bits (US) (SRC (<) ON)

[i]…US
[S1]…DT0 [S2]…DT1

	Hexadecimal	Unsigned decimal	Signed decimal
DT0	H 6000	**K 24576**	K 24576
DT1	H 8500	**K 34048**	K -31488

Flag operations during execution

	SRA(>)	SRB(=)	SRC(<)
DT0 < DT1	OFF	OFF	ON

Example 2) Operation unit: Signed 16 bits (SS) (SRA (>) ON)

[i]…SS
[S1]…DT0 [S2]…DT1

	Hexadecimal	Unsigned decimal	Signed decimal
DT0	H 6000	K 24576	**K 24576**
DT1	H 8500	K 34048	**K -31488**

Flag operations during execution

	SRA(>)	SRB(=)	SRC(<)
DT0 > DT1	ON	OFF	OFF

Example 3) Operation unit: Unsigned 16 bits (US) (SRB (=) ON)

[i]…US
[S1]…DT0 [S2]…DT1

	Hexadecimal	Unsigned decimal	Signed decimal
DT0	H 1234	**K 4660**	K 4660
DT1	H 1234	**K 4660**	K 4660

Flag operations during execution

	SRA(>)	SRB(=)	SRC(<)
DT0 = DT1	OFF	ON	OFF

Example 4) Operation unit: Signed 32 bits (SL) (SRC (<) ON)

[i]…SL [S1]…I0 [S2]…TS0

	Hexadecimal	Unsigned decimal	Signed decimal
I0	H 85000000	K 2231369728	**K -2063597568**
DT0·DT1	H 60000000	K 1610612736	**K 1610612736**

Flag operation when executed

	SRA(>)	SRB(=)	SRC(<)
I0 < TS0	OFF	OFF	ON

Example 5) Operation unit: Single-precision floating point real number (SF) (SRA (>) ON)

[i]…SF [S1]…DT0 [S2]…LD0

	Decimal real number
DT0･DT1	**SF 1.234E+00**
LD0･LD1	**SF -1.234E+00**

Flag operation when executed

	SRA(>)	SRB(=)	SRC(<)
DT0･DT1 > LD0･LD1	ON	OFF	OFF

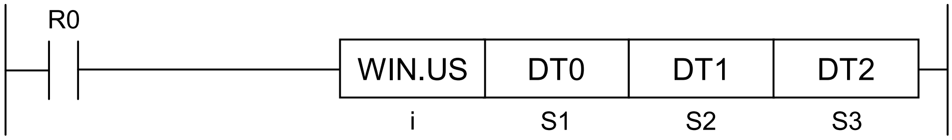
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when a non-real number is specified for [S1] or [S2], and the operation unit is a real number (SF).

Name	Description
SRA (>)	Depending on the comparison result
SRB (=)	
SRC (<)	

4.2 WIN (Band Compare)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Comparison data (device address or constant)
S2	Lower limit (device address or constant)
S3	Upper limit (device address or constant)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)		Integer				Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●		●
S3	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●		●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], [S1] (comparison data) is checked to determine if it is within the range between [S2] (lower limit) and [S3] (upper limit).
- The comparison result is output to the system relays SRA to SRC (assessment flags for comparison flags).

■ Processing

- Results of the flags (SR (system relays)) are as follows, based on relationship between [S1] and [S2] and [S3].

Relationship between [S1] and [S2] and [S3]	Comparison flags (SR (system relays))		
	SRA	SRB	SRC
	>	=	<
[S1] < [S2]	OFF	OFF	ON
[S2] ≦ [S1] ≦ [S3]	OFF	ON	OFF
[S1] > [S3]	ON	OFF	OFF

Example 1) Operation unit: Unsigned 16 bits (US) (SRB(=) ON)

[i]...US
[S1]...DT0 [S2]...DT1 [S3]...DT2

	Hexadecimal	Unsigned decimal	Signed decimal
DT0	H 8000	**K 32768**	K -32767
DT1	H 7000	**K 28672**	K 28672
DT2	H 9000	**K 36864**	K -28671

Flag operations during execution

	SRA(>)	SRB(=)	SRC(<)
DT1 < DT0 < DT2	OFF	ON	OFF

Example 2) Operation unit: Unsigned 16 bits (SRC(<) ON)

[i]...US
[S1]...DT0 [S2]...DT1 [S3]...DT2

	Hexadecimal	Unsigned decimal	Signed decimal
DT0	H 6000	**K 24576**	K 24576
DT1	H 7000	**K 28672**	K 28672
DT2	H 9000	**K 36864**	K -28671

Flag operations during execution

	SRA(>)	SRB(=)	SRC(<)
DT0 < DT1	OFF	OFF	ON

Example 3) Operation unit: Signed 16 bits (SS) (operation error)
[i]…SS
[S1]...DT0 [S2]…DT1 [S3]…DT2

	Hexadecimal	Unsigned decimal	Signed decimal
DT0	H 6000	K 32768	**K -32767**
DT1	H 7000	K 28672	**K 28672**
DT2	H 9000	K 36864	**K -28671**

Flag operations during execution

	SRA(>)	SRB(=)	SRC(<)
DT1 > LD2	OFF	OFF	OFF

* Operation error because of [S2] > [S3] (Set SR7 (latest error) and SR8 (hold error))

Example 4) Operation unit: Single-precision, floating-point real number (SF) (SRA(>) ON)
[i]…SF
[S1]…DT0 [S2]…LD0 [S3]…LD2

	Value (real number decimal)
DT0･DT1	SF 8.000E + 02
LD0･LD1	SF 5.000E + 02
LD2･LD3	SF 7.000E + 02

Flag operations during execution

	SRA(>)	SRB(=)	SRC(<)
DT0･DT1 > LD2･LD3	ON	OFF	OFF

■ Precautions for programming

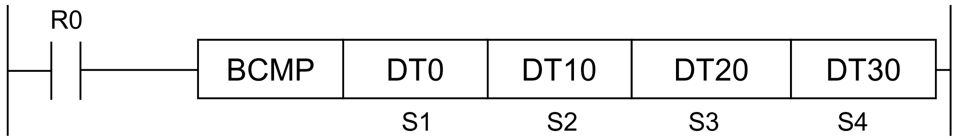
- In the case of a direct address and an index modification address, ensure that [S3] is equal to or larger than [S2].

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [S2] is larger than [S3].
	To be set when a non-real number is specified for [S1], [S2] or [S3], and the operation unit is real numbers (SF).
SRA (>)	Depending on the comparison result
SRB (=)	
SRC (<)	

4.3 BCMP (Block Comparison)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Area that stores control data, or the constant data
S2	Number of data to be compared (device address or constant) (available range: 1 to 4096)
S3	Starting address (device address) of comparison block 1
S4	Starting address (device address) of comparison block 2

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●								●	●				●
S2	●	●	●	●			●	●								●	●				●
S3	●	●	●	●			●	●													●
S4	●	●	●	●			●	●													●

■ Outline of operation

- Data in the area (comparison block 1) that is specified by [S3] is compared with data in the area (comparison block 2) that is specified by [S4].
- If the contents of both blocks match each other, the system relay SRB ("=" flag) turns ON.

■ Specification of control data [S1]

S1	Start of block 1	Start of block 2
0	From low byte	From low byte
1	From high byte	From low byte
2	From low byte	From high byte
3	From high byte	From high byte

■ Processing

Example 1) Comparison between 5 bytes that start from a low byte in block 1 and 5 bytes that start from a low byte in block 2

[S1]...H0 [S2]...U5 [S3]...DT1 [S4]...DT10 → SRB:ON

Block 1

	High	Low
DT0	H 00	H 00
DT1	**H 31**	**H 30**
DT2	**H 33**	**H 32**
DT3	H 35	**H 34**

Block 2

	High	Low
DT10	**H 31**	**H 30**
DT11	**H 33**	**H 32**
DT12	H 35	**H 34**
DT13	H 37	H 36

Example 2) Comparison between 5 bytes that start from a high byte in block 1 and 5 bytes that start from a low byte in block 2

[S1]...H1 [S2]...U5 [S3]...DT1 [S4]...DT10 → SRB:OFF

Block 1

	High	Low
DT0	H 00	H 00
DT1	**H 31**	H 30
DT2	**H 33**	**H 32**
DT3	**H 35**	**H 34**

Block 2

	High	Low
DT10	**H 31**	**H 30**
DT11	**H 33**	**H 33**
DT12	H 35	**H 34**
DT13	H 37	H 36

Example 3) Comparison between 6 bytes that start from a high byte in block 1 and 6 bytes that starts from a low byte in block 2

[S1]...H1 [S2]...U6 [S3]...DT0 [S4]...DT10 → SRB:ON

Block 1

	High	Low
DT0	**H 30**	H 00
DT1	**H 32**	**H 31**
DT2	**H 34**	**H 33**
DT3	H 00	**H 35**

Block 2

	High	Low
DT10	**H 31**	**H 30**
DT11	**H 33**	**H 32**
DT12	**H 35**	**H 34**
DT13	H 37	H 36

Example 4) Comparison between 7 bytes that start from a high byte in block 1 and 7 bytes that starts from a high byte in block 2

[S1]...H3 [S2]...U7 [S3]...DT1 [S4]...DT10 → SRB:ON

Block 1

	High	Low
DT0	H 00	H 00
DT1	**H 31**	H 30
DT2	**H 33**	**H 32**
DT3	**H 35**	**H 34**
DT4	**H 37**	**H 36**
DT5	H 39	H 38

Block 2

	High	Low
DT10	**H 31**	H 30
DT11	**H 33**	**H 32**
DT12	**H 35**	**H 34**
DT13	**H 37**	**H 36**
DT14	H 39	H 38

■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).

Name	Description
SR8 (ER)	To be set when the comparison range is outside the accessible range.
	To be set when the control data is outside the range.
	To be set when the block length is outside the available range.
SRB (=)	To be set when the comparison blocks of [S3] and [S4] match.
	To be reset when the comparison blocks of [S3] and [S4] do not match.

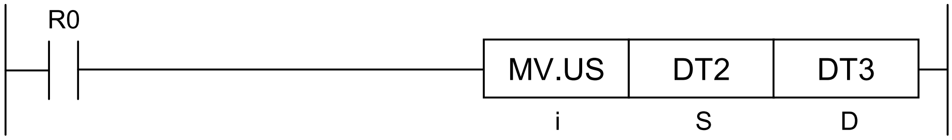
(MEMO)

5 High-level Instructions (Data Transfer)

Applicable Models: All Models

5.1 MV (Data Transfer)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S	The source device address or the constant
D	Destination device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

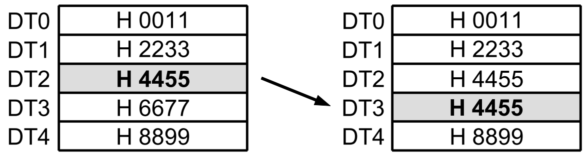
■ Outline of operation

- This instruction transfers the operation unit data specified by [i] from the device address or the constant specified by [S] to the device address specified by [D].
  [S] → [D]

■ Processing

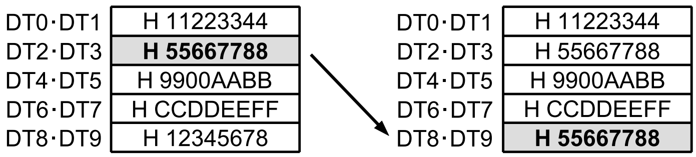
Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S] …DT2 [D] …DT3



Example 2) When the operation unit is 32-bit (UL, SL, SF)

[i]…UL,SL,SF [S] …DT2 [D] …DT8

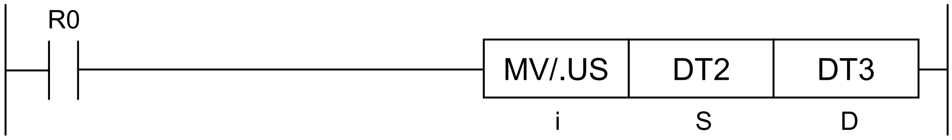


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

5.2 MV/ (Inversion and Transfer)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S	The source device address or the constant
D	Destination device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- This instruction logically inverts and transfers the specified operation unit data [i] from the device address or the constant specified by [S] to the device address specified by [D].

  /[S] → [D]

BIN	0 0 0 0	0 1 0 0	1 1 0 1	0 0 1 0
HEX	0	4	D	2

↓

BIN	1 1 1 1	1 0 1 1	0 0 1 0	1 1 0 1
HEX	F	B	2	D

■ Processing

Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S] …DT2 [D] …DT3

		→		
DT0	H 0011		DT0	H 0011
DT1	H 2233		DT1	H 2233
DT2	**H 4455**		DT2	H 4455
DT3	H 6677		DT3	**H BBAA**
DT4	H 8899		DT4	H 8899

Example 2) When the operation unit is 32-bit (UL, SL, SF)

[i]…UL,SL,SF [S] …DT2 [D] …DT8

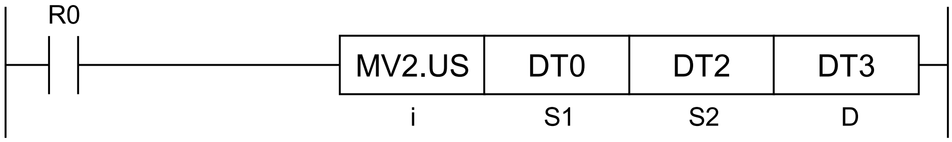
		→		
DT0·DT1	H 11223344		DT0·DT1	H 11223344
DT2·DT3	**H 55667788**		DT2·DT3	H 55667788
DT4·DT5	H 9900AABB		DT4·DT5	H 9900AABB
DT6·DT7	H CCDDEEFF		DT6·DT7	H CCDDEEFF
DT8·DT9	H 12345678		DT8·DT9	**H AA998877**

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

5.3 MV2 (2 Data Transfer)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	The device address of the source 1 or the constant
S2	The device address of the source 2 or the constant
D	Destination device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)		Integer				Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- This instruction transfers two data specified by [S1] and [S2] to the area starting from [D] all at once according to the operation unit specified by [i].

■ Processing

Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S1] …DT0 [S2] …DT2 [D] …DT3

DT0	**H 0011**		DT0	H 0011
DT1	H 2233		DT1	H 2233
DT2	**H 4455**		DT2	H 4455
DT3	H 6677		DT3	**H 0011**
DT4	H 8899		DT4	**H 4455**

Example 2) When the operation unit is 32-bit (UL, SL, SF)

[i]…UL,SL,SF [S1] …DT10 [S2] …DT20 [D] …DT30

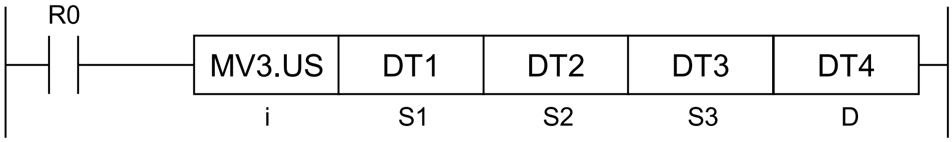
DT11-DT10	H 11223344		DT31-DT30	**H 55667788**
DT13-DT12	**H 55667788**		DT33-DT32	**H CCDDEEFF**
			DT35-DT34	H 00000000
DT21-DT20	H CCDDEEFF		DT37-DT36	H 00000000
DT23-DT22	**H CCDDEEFF**		DT39-DT38	H 00000000

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the transfer range is outside the accessible range.

5.4 MV3 (3 Data Transfer)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	The device address of the source 1 or the constant
S2	The device address of the source 2 or the constant
S3	The device address of the source 3 or the constant
D	Destination device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S3	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

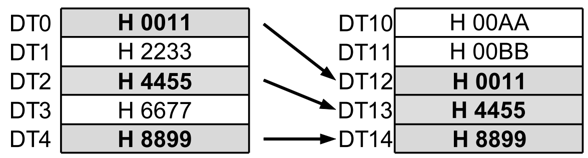
■ Outline of operation

- This instruction transfers three data specified by [S1], [S2] and [S3] to the area starting from [D] all at once according to the operation unit specified by [i].

■ Processing

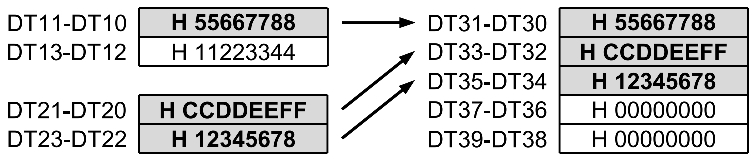
Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S1] …DT0 [S2] …DT2 [S3] …DT4 [D] …DT12



Example 2) When the operation unit is 32-bit (UL, SL, SF)

[i]…UL,SL,SF [S1] …DT10 [S2] …DT20 [S3] …DT22 [D] …DT30



■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the transfer range is outside the accessible range.

5.5 BKMV (Block Transfer)

■ Ladder diagram

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Starting device address of source data
S2	Ending device address of source data
D	Destination starting device address to transfer data

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●							●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●							●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

■ Outline of operation

- This instruction transfers data in the area specified by [S1] to [S2] to the area specified by [D] and subsequent areas all at once.

■ Processing

Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S1]…DT1 [S2]…DT3 [D]…DT12

DT0	H 0011		DT10	H 0011
DT1	**H 2233**	→	DT11	H 2233
DT2	**H 4455**	→	DT12	**H 2233**
DT3	**H 6677**	→	DT13	**H 4455**
DT4	H 8899		DT14	**H 6677**

Example 2) When the operation unit is 32-bit (UL, SL, SF)

[i]…UL,SL,SF [S1]…DT2 [S2]…DT8 [D]…DT10

DT0·DT1	H 11223344		DT10·DT11	**H 55667788**
DT2·DT3	**H 55667788**	→	DT12·DT13	**H 9900AABB**
DT4·DT5	**H 9900AABB**	→	DT14·DT15	**H CCDDEEFF**
DT6·DT7	**H CCDDEEFF**	→	DT16·DT17	**H 12345678**
DT8·DT9	**H 12345678**	→	DT18·DT19	H 24680ACE

■ Precautions for programming

- In the case of a direct address and index modification address, specify the same device for [S1] and [S2]. At the same time, specify [S2] to be greater than or equal to [S1].
- The data are transferred by operation unit, ending with the device containing [S2].

Example 1) Device address of [S2] comes to a low word (operation unit = 32 bits)

[S1]...DT2 [S2]...DT6 [D]...DT12

DT1-DT0	H 11223344		DT11-DT10	H 11111111
DT3-DT2	**H 55667788**	→	DT13-DT12	**H 55667788**
DT5-DT4	**H 9900AABB**	→	DT15-DT14	**H 9900AABB**
DT7-DT6	**H CCDDEEFF**	→	DT17-DT16	**H CCDDEEFF**
DT9-DT8	H 12345678		DT19-DT18	H 55555555

Example 2) Device address of [S2] comes to a high word (operation unit = 32 bits)

[S1]...DT2 [S2]...DT7 [D]...DT12

DT1-DT0	H 11223344		DT11-DT10	H 11111111
DT3-DT2	**H 55667788**	→	DT13-DT12	**H 55667788**
DT5-DT4	**H 9900AABB**	→	DT15-DT14	**H 9900AABB**
DT7-DT6	**H CCDDEEFF**	→	DT17-DT16	**H CCDDEEFF**
DT9-DT8	H 12345678		DT19-DT18	H 55555555

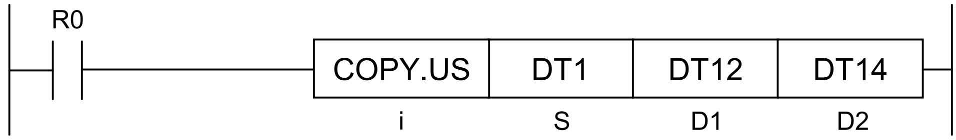
■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).

Name	Description
SR8 (ER)	To be set when [S1] is larger than [S2].
	To be set when the destination range is outside the accessible range.

5.6 COPY (Block Copy)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S	The device address or the constant of the source data
D1	Starting device address of destination area
D2	End device address of the destination area

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)(Note 2)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D1	●	●	●	●			●	●	●		●	●	●	●							●
D2	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

Outline of operation

This instruction copies data specified by [S] to the areas of [D1] to [D2].

Processing

Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S]…DT1 [D1]…DT12 [D2]…DT14

DT0	H 0011	DT10	H 0011
DT1	**H 2233**	DT11	H 2233
DT2	H 4455	DT12	**H 2233**
DT3	H 6677	DT13	**H 2233**
DT4	H 8899	DT14	**H 2233**

Example 2) Operation unit: 32 bits (UL, SL, SF)

[i]…UL,SL,SF [S]…DT6 [D1]…DT10 [D2]…DT16

DT0·DT1	H 11223344	DT10·DT11	**H CCDDEEFF**
DT2·DT3	H 55667788	DT12·DT13	**H CCDDEEFF**
DT4·DT5	H 9900AABB	DT14·DT15	**H CCDDEEFF**
DT6·DT7	**H CCDDEEFF**	DT16·DT17	**H CCDDEEFF**
DT8·DT9	H 12345678	DT18·DT19	H 24680ACE

Precautions for programming

- In the case of a direct address and index modification address, specify the same device for [D1] and [D2]. At the same time, specify [D2] to be greater than or equal to [D1].
- The data are transferred by operation unit, ending with the device containing [D2].

Example 1) Device address of [D2] comes to a high word (operation unit = 32 bits)
[S]...DT6 [D1]...DT10 [D2]...DT17

DT1-DT0	H 11223344	DT11-DT10	**H CCDDEEFF**
DT3-DT2	H 55667788	DT13-DT12	**H CCDDEEFF**
DT5-DT4	H 9900AABB	DT15-DT14	**H CCDDEEFF**
DT7-DT6	**H CCDDEEFF**	DT17-DT16	**H CCDDEEFF**
DT9-DT8	H 12345678	DT19-DT18	H 24680ACE

Example 2) Device address of [D2] comes to a low word (operation unit = 32 bits)
[S]...DT6 [D1]...DT10 [D2]...DT16

DT1-DT0	H 11223344	DT11-DT10	**H CCDDEEFF**
DT3-DT2	H 55667788	DT13-DT12	**H CCDDEEFF**
DT5-DT4	H 9900AABB	DT15-DT14	**H CCDDEEFF**
DT7-DT6	**H CCDDEEFF**	DT17-DT16	**H CCDDEEFF**
DT9-DT8	H 12345678	DT19-DT18	H 24680ACE

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [D1] is larger than [D2].

5.7 BTM (Bit Block Transfer)

■ Ladder diagram

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i	●						

■ List of operands

Operand	Description
S1	Starting bit address of the source data
S2	End bit address of the source data
D	Starting bit address of the data destination

■ Available devices (●: Available)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●
D	●	●	●	●							●	●	●	●

■ Outline of operation

- This instruction performs bit transfer, from the area (bit address) specified by [S1] through the area (bit address) specified by [S2], to the area specified by [D].

Processing

Example 1) Transfer X1 through X8 to Y6 through YD
[S1]…X1 [S2]…X8 [D]…Y6

	WX0															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	0	1	1	0	1	0	0	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	1

	WY0															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	0	0	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	0	0	0	0	0	0

Example 2) Transfer X1 through X8 to YD through Y14
[S1]…X1 [S2]…X8 [D]…YD

	WX0															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	0	1	1	0	1	0	0	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	1

	WY1																WY0															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0

Precautions for programming

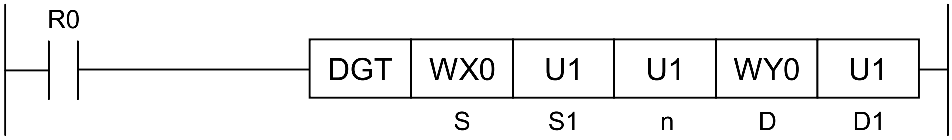
- In the case of a direct address and index modification address, specify the same device for [S1] and [S2]. At the same time, specify [S2] to be greater than or equal to [S1].

Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [S1] is larger than [S2].
	To be set when the destination range is outside the accessible range.

5.8 DGT (Digit Data Transfer)

■ Ladder diagram



■ List of operands

Operand	Description
S	The device address or the constant of the source data
S1	Transfer starting digit in the source (Available data range: 0 to 3)
n	Digits to be transferred (Available data range: 1 to 4)
D	The device address of the destination data
D1	Transfer starting digit in the destination (Available data range: 0 to 3)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●
S1 (Note 1)	●	●	●	●			●	●	●	●	●					●	●				●
n (Note 1)	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●
D1 (Note 1)	●	●	●	●			●	●	●	●	●					●	●				●

(Note 1) To be handled as a 16-bit unsigned integer (US), regardless of operation unit.

■ Outline of operation

- This instruction transfers [n] digits from the [S1]th digit of the area specified by [S], to the [D1] digit of the 16-bit data specified by [D].
- Transfer starts with the 0th digit, 1st digit, 2nd digit, and 3rd digit by every four bits from the lower level.

bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
	Digit 3				Digit 2				Digit 1				Digit 0			

■ Processing

Example 1) Transfer from Digit 1 to Digit 1

[S]…WX0 [S1]…U1(H1)
[n]…U1(H1)
[D]…WY0 [D1]…U1(H1)

	X															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	0	1	1	0	1	0	0	1	**1**	**1**	**0**	**0**	0	0	1	1

	Y															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	0	0	0	0	0	0	0	0	**1**	**1**	**0**	**0**	0	0	0	0

Example 2) Shift by one digit and transfer

[S]…WX0 [S1]…U3(H3)
[n]…U1(H1)
[D]…WY0 [D1]…U0(H0)

	X															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	**0**	**1**	**1**	**0**	1	0	0	1	1	1	0	0	0	0	1	1

	Y															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	0	0	0	0	0	0	0	0	0	0	0	0	**0**	**1**	**1**	**0**

Example 3) Transfer multiple digits in parallel

[S]…WX0 [S1]…U2(H2)
[n]…U2(H2)
[D]…WY0 [D1]…U2(H2)

	X															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	1	1	0	0	0	0	1	1

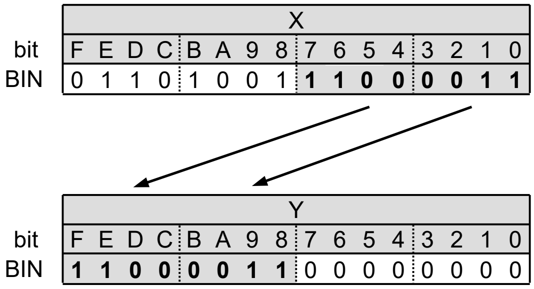
	Y															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	0	0	0	0	0	0	0	0

Example 4) Shift and transfer multiple digits

[S]…WX0 [S1]…U0(H0)

[n]…U2(H2)

[D]…WY0 [D1]…U2(H2)

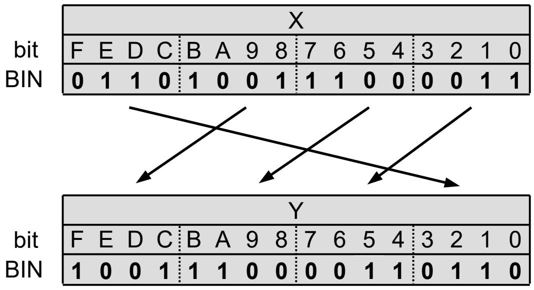


Example 5) Transfer four digits

[S]…WX0 [S1]…U0(H0)

[n]…U4(H4)

[D]…WY0 [D1]…U1(H1)

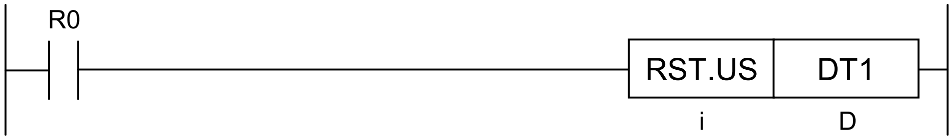


■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).
SR8 (ER)	To be set when the operands [S1], [n], and/or [D1] are out of the specified range.

5.9 RST (Reset)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
D	Target of reset

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
D	●	●	●	●		(Note 3)	●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

(Note 3) Only SD60/SD61 is permitted.

■ Outline of operation

- If the operation unit are 16 bits (US, SS) or 32 bits (UL, SL, SF), the area specified by [D] is reset (cleared to zero).

■ Processing

Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [D]…DT1

	Before			After
DT0	H 0011		DT0	H 0011
DT1	**H 2233**	→	DT1	**H 0000**
DT2	H 4455		DT2	H 4455
DT3	H 6677		DT3	H 6677
DT4	H 8899		DT4	H 8899

Example 2) When the operation unit is 32-bit (UL, SL, SF)

[i]…UL,SL,SF [D]…DT6

	Before			After
DT0·DT1	H 11223344		DT0·DT1	H 11223344
DT2·DT3	H 55667788		DT2·DT3	H 55667788
DT4·DT5	H 9900AABB		DT4·DT5	H 9900AABB
DT6·DT7	**H CCDDEEFF**	→	DT6·DT7	**H 00000000**
DT8·DT9	H 12345678		DT8·DT9	H 12345678

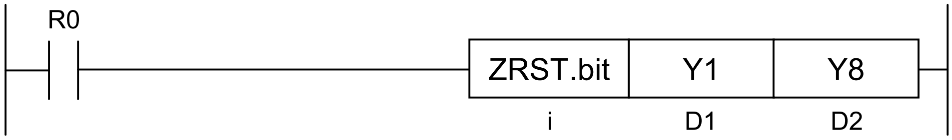
* In the case of SF, all 32 bits are 0 (sign, exponent and mantissa parts are 0), resulting in 0.00e+00.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

5.10 ZRST (Block Clear)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i	●						

■ List of operands

Operand	Description
D1	Starting bit address of the reset data
D2	End bit address of the reset data

■ Available devices (●: Available)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
D1	●	●	●	●							●	●	●	●
D2	●	●	●	●							●	●	●	●

■ Outline of operation

- This instruction clears to zero (resets) from the area (bit address) specified by [D1] through the area (bit address) specified by [D2].
- This can also be used for a package clearance of processes that are starting up from Process [D1] to Process [D2] in the step ladder.

■ Processing

Example 1) Reset Y1 through Y8
[D1]…Y1 [D2]…Y8

	WY0															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	1	1	1	1	1	1	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	1

↓

	WY0															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	1	1	1	1	1	1	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	1

Example 2) Reset YD through Y14
[D1]…YD [D2]…Y14

	WY1																WY0															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	1	1	1	1	1	1	1	1	1	1	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	1	1	1	1	1	1	1	1	1	1	1	1	1

↓

	WY1																WY0															
bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	1	1	1	1	1	1	1	1	1	1	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	1	1	1	1	1	1	1	1	1	1	1	1	1

■ Precautions for programming

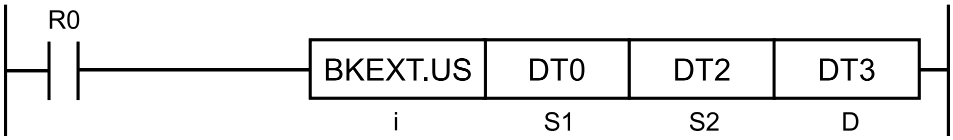
- In the case of a direct address and index modification address, specify the same device for [D1] and [D2]. At the same time, specify [D2] to be greater than or equal to [D1].

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [D1] is larger than [D2].

5.11 BKEXT (16-bit Data Sign-extended Block Transfer)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S1	The starting address of the device storing the data that sign extension is performed.
S2	The ending address of the device storing the data that sign extension is performed.
D	Destination starting device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●													●
S2	●	●	●	●	●	●	●	●													●
D	●	●	●	●			●	●				●	●	●							●

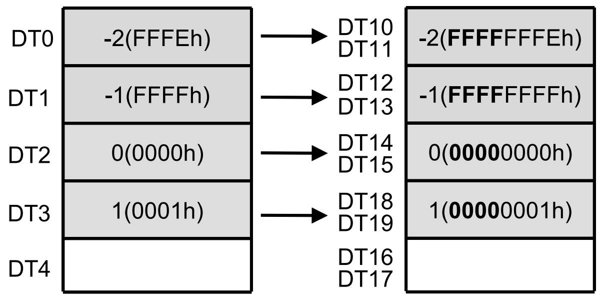
■ Outline of operation

- This instruction performs sign extension for device values in the area specified by [S1] to [S2], and transfers them to the device address specified by [D] and subsequent addresses.

■ Processing

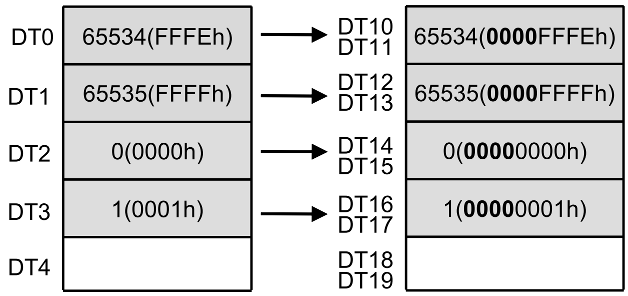
Example 1) Operation unit: signed 16-bit (SS)

[S1]...DT0 [S2]...DT3 [D]...DT10



Example 2) Operation unit: unsigned 16-bit (US)

[S1]...DT0 [S2]...DT3 [D]...TS0



■ Precautions for programming

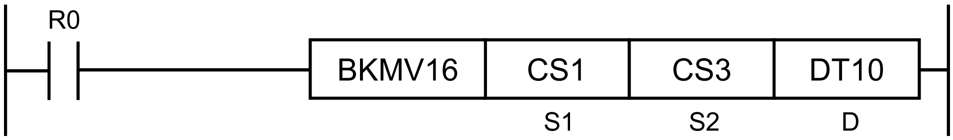
- In the case of a direct address and index modification address, specify the same device for [S1] and [S2]. At the same time, specify [S2] to be greater than or equal to [S1].
- The specified source area and destination area should not overlap each other.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when [S1] is larger than [S2].
	To be set when the destination range is outside the accessible range.
	To be set when the ranges of the source area and the destination area overlap.

5.12 BKMV16 (Block Transfer (32-bit Data to 16-bit Data))

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of source data
S2	Ending device address of source data
D	Destination starting device address to transfer data

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1												●	●	●							●
S2												●	●	●							●
D	●	●	●	●			●	●													●

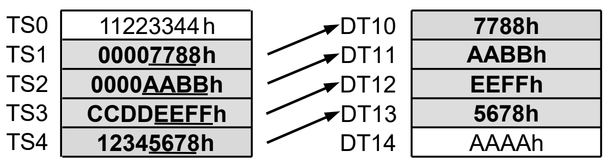
■ Outline of operation

This instruction transfers only one low word of data in the area specified by [S1] to [S2] to the area specified by [D] and subsequent areas all at once.

■ Processing

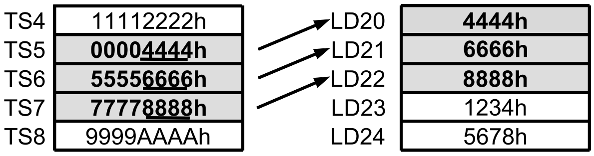
Example 1) When specifying TS for [S1] and [S2], and DT for [D] (transferring only lower one word)

[S1]...TS1 [S2]...TS4 [D]...DT10



Example 2) When specifying TS for [S1] and [S2], and LD for [D] (transferring only lower one word)

[S1]...TS5 [S2]...TS7 [D]...LD20



■ Precautions for programming

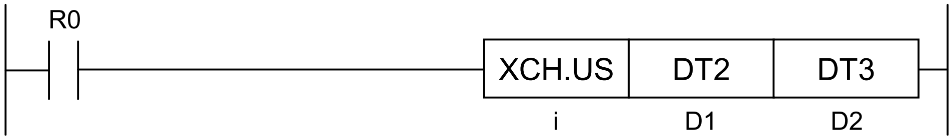
- In the case of a direct address and index modification address, specify the same device for [S1] and [S2]. At the same time, specify [S2] to be greater than or equal to [S1].

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when [S1] is larger than [S2].
	To be set when the destination range is outside the accessible range.

5.13 XCH (Data Exchange)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
D1	Device address of exchanged data 1
D2	Device address of exchanged data 2

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
D1	●	●	●	●			●	●	●		●	●	●	●							●
D2	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

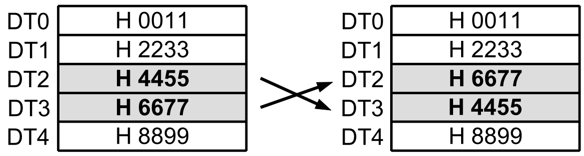
■ Outline of operation

- This instruction exchanges the data of the device address specified by [D1] and the device address specified by [D2] according to the operation unit [i].
  [D1] ⇔ [D2]

■ Processing

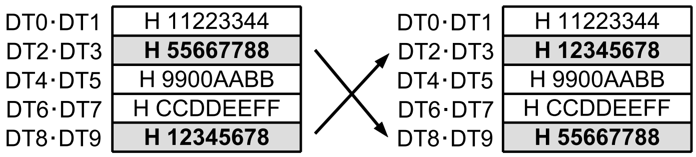
Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [D1]…DT2 [D2]…DT3



Example 2) When the operation unit is 32-bit (UL, SL, SF)

[i]…UL,SL,SF [D1]…DT2 [D2]…DT8



■ Precautions for programming

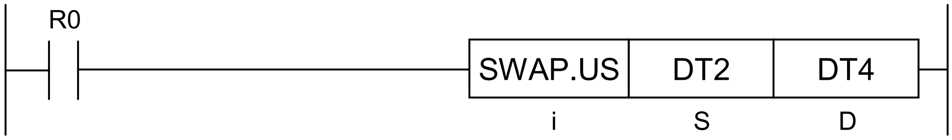
- Ensure that the ranges of the exchanged data do not overlap.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

5.14 SWAP (Exchange of High Bytes and Low Bytes)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S	Device address of the source where high bytes and low bytes should be exchanged
D	Device address of destination of the exchanged data

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●										●
D	●	●	●	●			●	●	●		●										●

■ Outline of operation

- This instruction exchanges high bytes and low bytes of the device address specified by [S], and transfers the resulting data to the device address specified by [D].

■ Processing

Example) Operation unit: 16 bits (US, SS)
[i]…US,SS
[S]…DT2 [D]…DT4

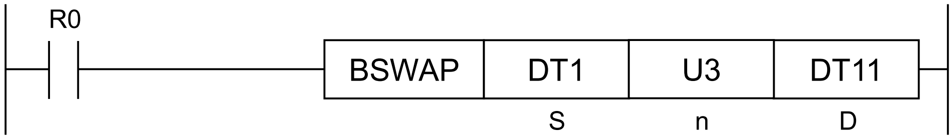
DT0	H 0011		DT0	H 0011
DT1	H 2233		DT1	H 2233
DT2	**H 4455**	→	DT2	H 4455
DT3	H 6677		DT3	H 6677
DT4	H 8899		DT4	**H 5544**

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

5.15 BSWAP (High /Low Byte in n Block Exchange)

■ Ladder diagram



■ List of operands

Operand	Description
S	The starting address of the device to exchange the high and low bytes
n	The number of words to exchange the high and low bytes
D	Device address of destination of the exchanged data

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●													●
n	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

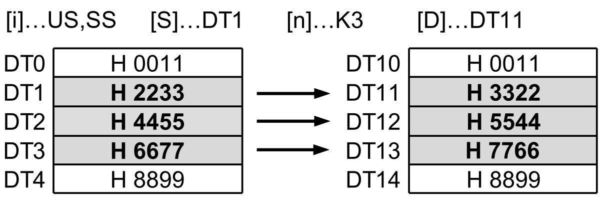
(Note 1) Only 16-bit devices can be modified. (Integer constants cannot be specified.)

■ Outline of operation

- This instruction exchanges the high byte and low byte for [n] words from the device address specified by [S], and transfers it to the area starting from [D].
- The maximum number of exchanged words is 65535.
- When [n] is 0, no operation is performed.

■ Processing

Example) Operation unit: 16 bits (US, SS)

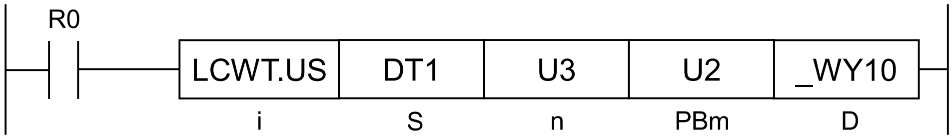


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the transfer range is outside the accessible range.

5.16 LCWT (Specified PB Local Device Write)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S	Starting address of a source device
n	Number of written devices (Available range: 1 to 65535)
PBm	Destination PB number (Available range: 1 to Max. number of PB)
D	The starting address of a destination local device

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●				●	●	●							●
n	●	●	●	●			●	●								●	●				●
PBm	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●				●	●								

(Note 1) Only 16-bit devices and 32-bit devices can be modified. (Integer constants cannot be specified.)

(Note 2) Index register (I0 to IE)

■ Outline of operation

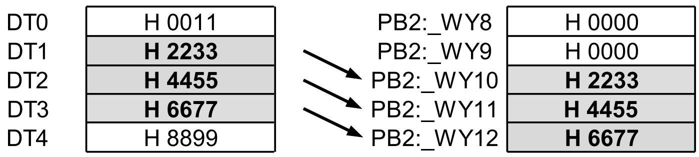
- This instruction writes the data for [n] from the area specified by [S] to the area specified by [PBm]:[D] (local device) and subsequent areas all at once.
- Either a global device or a local device (of the PB in which this instruction is executed) can also be specified for [S].

- For [D], specify a local device that is in the program block of the PB number specified for [PBm].
- Subroutine arguments can be specified by combining this instruction with the ECALL instruction.
  Refer to "Example of processing: Argument, return value operation of ECALL instruction".
- Local devices of multiple PBs can be preset by one PB.
  Refer to "Example of processing: Presetting of specified PB local devices".

■ Processing

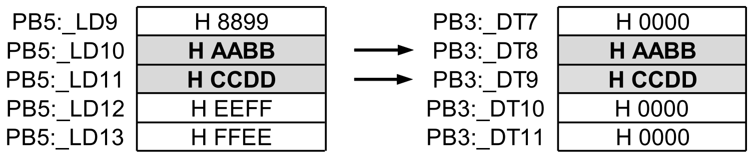
Example 1) When a global device is specified for [S]

[S]…DT1 [n]…3 [PBm]…2 [D]…_WY10



Example 2) When a local device is specified for [S] (Instruction is executed in PB5.)

[S]…_LD10 [n]…2 [PBm]…3 [D]…_DT8

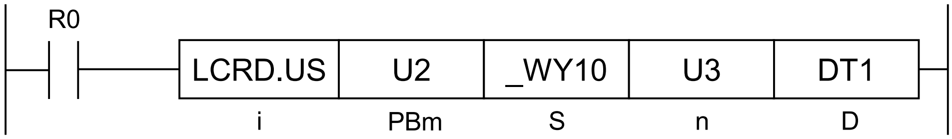


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when an out-of-range value is specified for parameters.
	To be set when the device address specified by [S+n] exceeds the upper limit of the device.
	To be set when [PBm] exceeds the maximum PB number.
	To be set when [D] is specified for a global device.
	To be set when the device address specified by [PBm]:[D+n] exceeds the upper limit of the device.

5.17 LCRD (Specified PB Local Device Read)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
PBm	Source PB number (Available range: 1 to Max. number of PB)
S	Starting address of source local device
n	Number of read devices (Available range: 1 to 65535)
D	Starting address of readout destination device

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
PBm	●	●	●	●			●	●								●	●				●
S	●	●	●	●			●	●				●	●								
n	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●				●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

■ Outline of operation

- This instruction reads the data for [n] from the area specified by [PBm]:[S] (local device) to the area specified by [D] and subsequent areas all at once.
- For [S], specify a local device that is in the program block of the PB number specified for [PBm].

- Either a global device or a local device (of the PB in which this instruction is executed) can be specified for [D].
- Subroutine arguments can be specified by combining this instruction with the ECALL instruction.
  Refer to "Example of processing: Argument, return value operation of ECALL instruction".

■ Processing

Example 1) When a global device is specified for [D]

[PBm]…2 [S]…_WY10 [n]…3 [D]…DT1

PB2:_WY8	H 0000		DT0	H 0011
PB2:_WY9	H 0000		DT1	**H 2233**
PB2:_WY10	**H 2233**	→	DT2	**H 4455**
PB2:_WY11	**H 4455**	→	DT3	**H 6677**
PB2:_WY12	**H 6677**	→	DT4	H 8899

Example 2) When a local device is specified for [D] (Instruction is executed in PB5.)

[PBm]…3 [S]…_DT8 [n]…2 [D]…_LD10

PB3:_DT7	H 0000		PB5:_LD9	H 8899
PB3:_DT8	**H AABB**	→	PB5:_LD10	**H AABB**
PB3:_DT9	**H CCDD**	→	PB5:_LD11	**H CCDD**
PB3:_DT10	H 0000		PB5:_LD12	H EEFF
PB3:_DT11	H 0000		PB5:_LD13	H FFEE

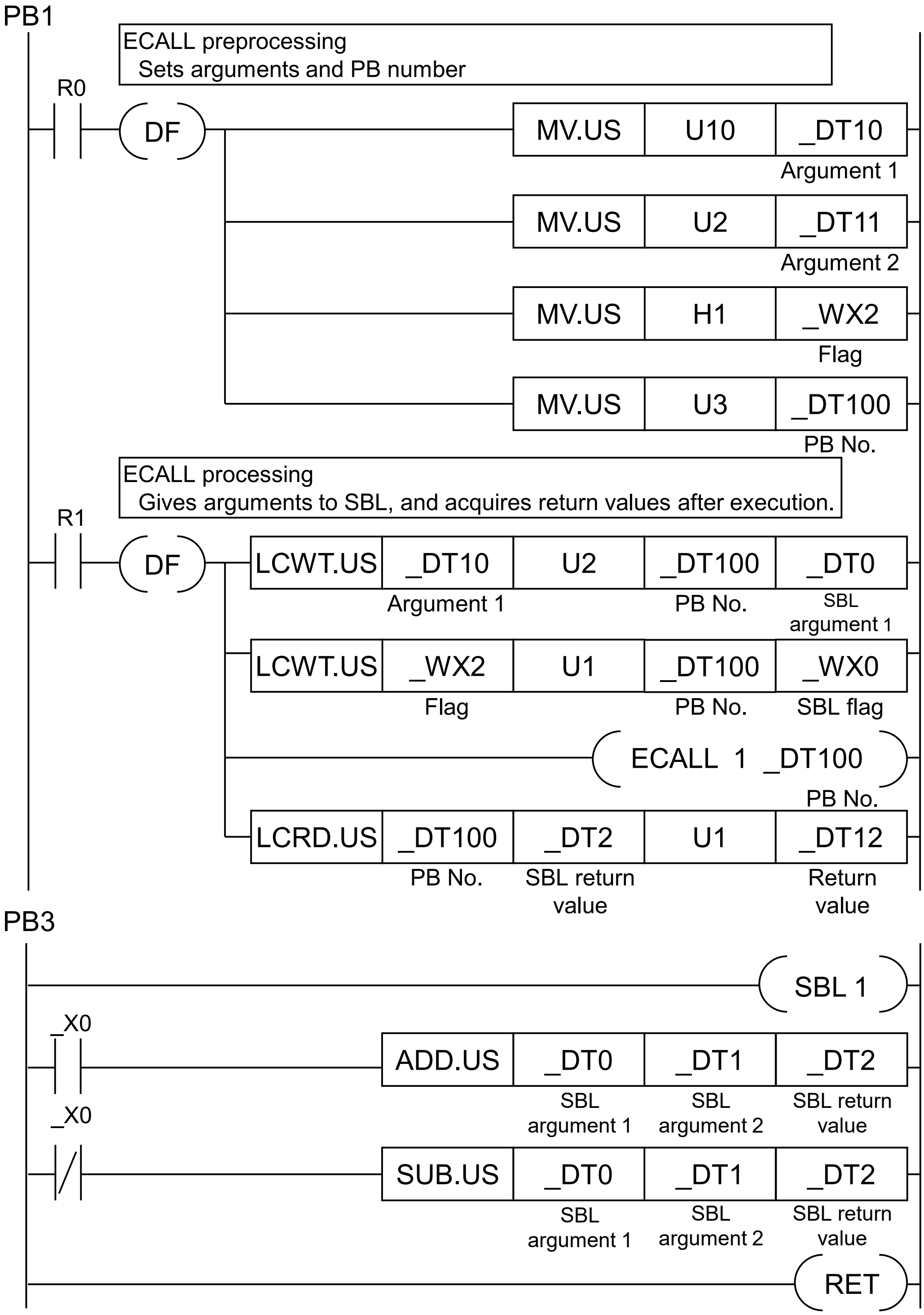
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when an out-of-range value is specified for parameters.
	To be set when [PBm] exceeds the maximum PB number.
	To be set when [S] is specified for a global device.
	To be set when the device address specified by [PBm]:[S+n] exceeds the upper limit of the device.
	To be set when the device address specified by [D+n] exceeds the upper limit of the device.

■ Example of processing: Argument, return value operation of ECALL instruction

The following example shows the operation in which the LCWT instruction is used to pass arguments to a subroutine in another PB and the LCRD instruction is used to receive return values.

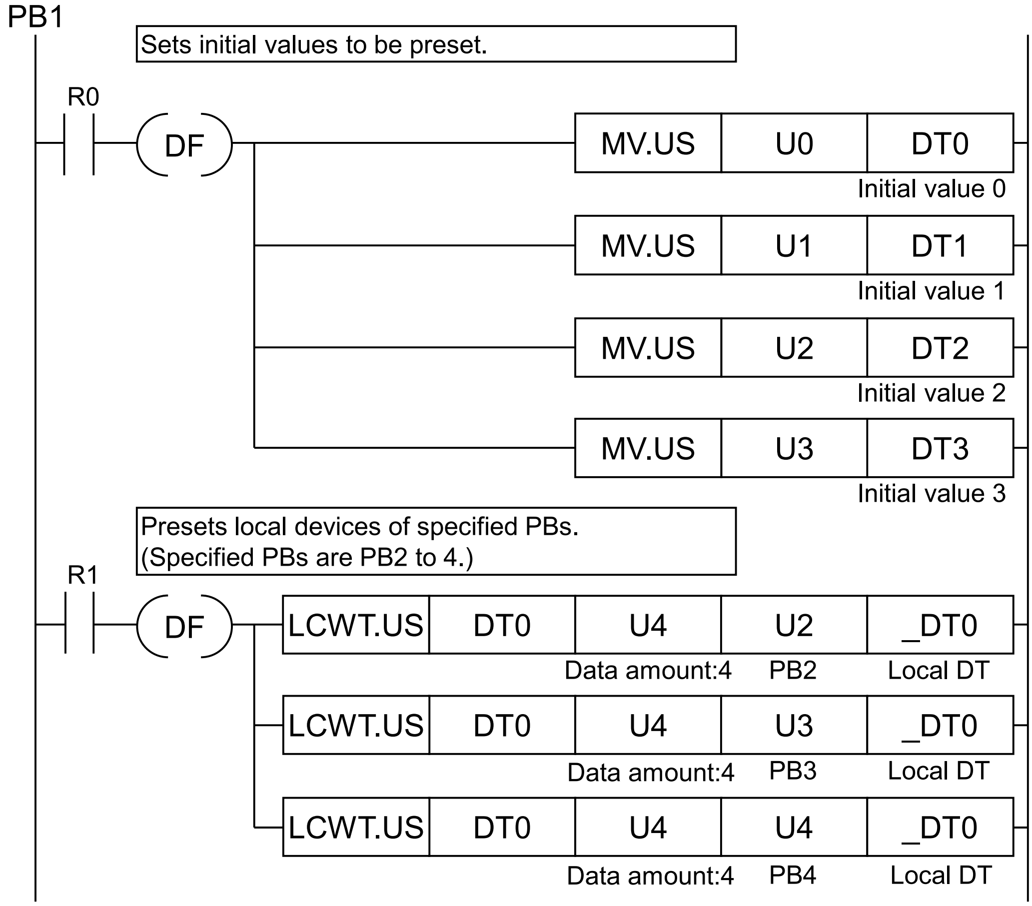
- This operation calls the subroutine SBL1 described in PB3 from PB1 to receive the results.
- SBL1 calculates "Argument 1 + Argument 2 → Return value" or "Argument 1 - Argument 2 → Return value", depending on the condition of _WX0.


PB1
ECALL preprocessing
Sets arguments and PB number
R0
DF
MV.US U10 _DT10
Argument 1
MV.US U2 _DT11
Argument 2
MV.US H1 _WX2
Flag
MV.US U3 _DT100
PB No.
ECALL processing
Gives arguments to SBL, and acquires return values after execution.
R1
DF
LCWT.US _DT10 U2 _DT100 _DT0
Argument 1
PB No.
SBL argument 1
LCWT.US _WX2 U1 _DT100 _WX0
Flag
PB No.
SBL flag
ECALL 1 _DT100
PB No.
LCRD.US _DT100 _DT2 U1 _DT12
PB No.
SBL return value
Return value
PB3
SBL 1
_X0
ADD.US _DT0 _DT1 _DT2
SBL argument 1
SBL argument 2
SBL return value
_X0
SUB.US _DT0 _DT1 _DT2
SBL argument 1
SBL argument 2
SBL return value
RET

■ Example of processing: Presetting of specified PB local devices

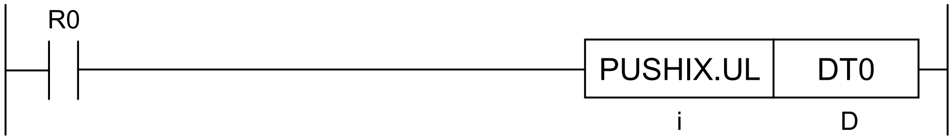
The following example shows how to initialize the local devices of PBs that are specified in one PB.

- Executes the instruction in PB1, and sets to initialize devices to the local devices of PB2 to 4 collectively.



5.18 PUSHIX (Index Register Backup)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i				●	●		

■ List of operands

Operand	Description
D	Start of device address of the backup destination

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●													●

■ Outline of operation

- This instruction backs up 15 data (30 words) of the index register value, starting with [D].
  (I0 to IE) → ([D] to [D] + 29)
- The index register value to be saved does not change.
- The PUSHIX instruction is used to back up the index register content before switching from the main program to the subprogram, in the case where index registers are used in subroutines or other subprograms.
- Please use this in combination with the POPIX (Index Register Recovery) instruction.

■ Processing

Example) Specify DT0 for the 1st operand [D]

I0	**H 00112233**
I1	**H 44556677**
I2	**H 8899AABB**
⋮	⋮
IC	**H CCDDEEFF**
ID	**H 12345678**
IE	**H 90ABCDEF**

Backup →

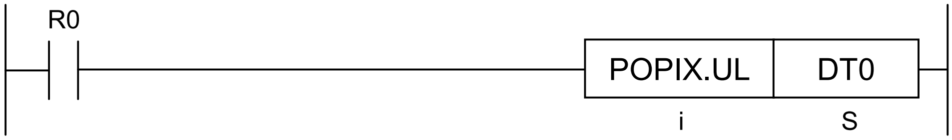
	Before backup	After backup
DT0	H 00000000	**H 00112233**
DT2	H 00000000	**H 44556677**
DT4	H 00000000	**H 8899AABB**
⋮	⋮	⋮
DT24	H 00000000	**H CCDDEEFF**
DT26	H 00000000	**H 12345678**
DT28	H 00000000	**H 90ABCDEF**

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the backup destination range is out of the accessible range.

5.19 POPIX (Index Register Recovery)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i				●	●		

■ List of operands

Operand	Description
S	Start of device address of the recovery source

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●													●

■ Outline of operation

- This instruction recovers 15 data (30 words) for the index register value, starting with [S].
  ([S] to [S] + 29) → (I0 to IE)
- The POPIX instruction is used to recover the content that has been backed up using the PUSHIX (Index Register Backup) instruction, before switching from the subprogram to the main program, in the case where index registers are used in subroutines or other subprograms.
- Please use this in combination with the PUSHIX (Index Register Backup) instruction.

■ Processing

Example) Specify DT0 for the 1st operand [S]

DT0	**H 00112233**
DT2	**H 44556677**
DT4	**H 8899AABB**
⋮	⋮
DT24	**H CCDDEEFF**
DT26	**H 12345678**
DT28	**H 90ABCDEF**

Recovery →

	Before recovery	After recovery
I0	H 00000000	**H 00112233**
I1	H 00000000	**H 44556677**
I2	H 00000000	**H 8899AABB**
⋮	⋮	⋮
IC	H 00000000	**H CCDDEEFF**
ID	H 00000000	**H 12345678**
IE	H 00000000	**H 90ABCDEF**

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the recovery source range is out of the accessible range.

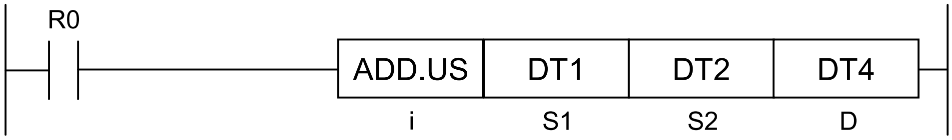
(MEMO)

6 High-level Instructions (Arithmetic/Logic Operations)

Applicable Models: All Models

6.1 ADD (Addition)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant)
S2	Calculation target data 2 (device address or constant)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

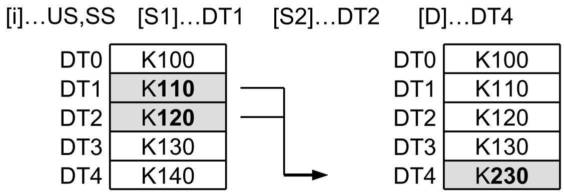
(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

Outline of operation

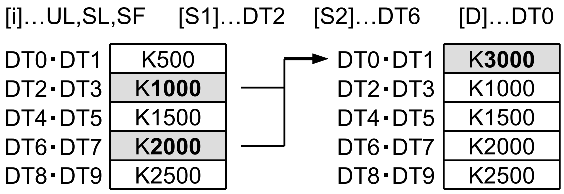
- This instruction adds the values [S1] and [S2] according to the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [S1] + [S2] → [D]

Processing

Example 1) Operation unit: 16 bits (US, SS)



Example 2) Operation unit: 32 bits (UL, SL, SF)

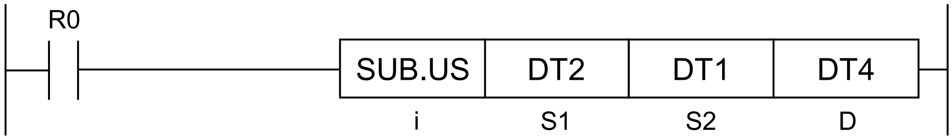


Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S1] or [S2], and the operation unit is a real number (SF).

6.2 SUB (Subtraction)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant)
S2	Calculation target data 2 (device address or constant)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

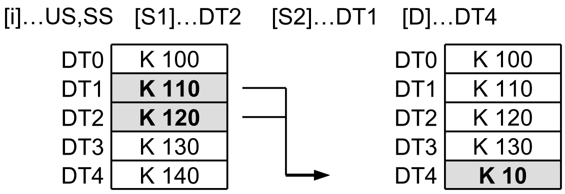
(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

Outline of operation

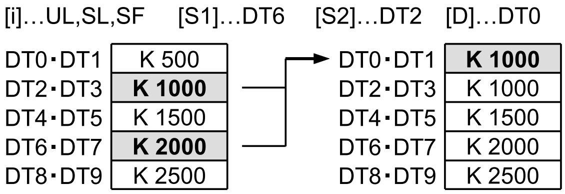
- This instruction subtracts the value [S2] from [S1] according to the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [S1] - [S2] → [D]

Processing

Example 1) Operation unit: 16 bits (US, SS)



Example 2) Operation unit: 32 bits (UL, SL, SF)

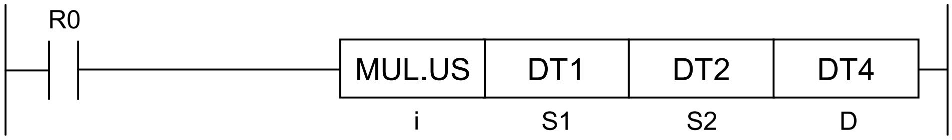


Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S1] or [S2], and the operation unit is a real number (SF).

6.3 MUL (Multiplication)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant)
S2	Calculation target data 2 (device address or constant)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

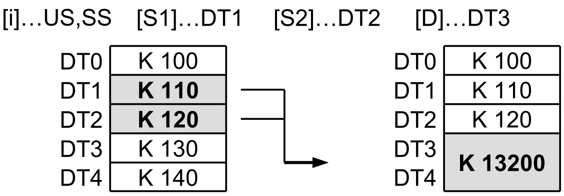
■ Outline of operation

- This instruction multiplies the values [S1] and [S2] according to the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [S1] × [S2] → ([D] to [D]+1)
- The size of the area in which the operation result is stored varies depending on the operation unit.

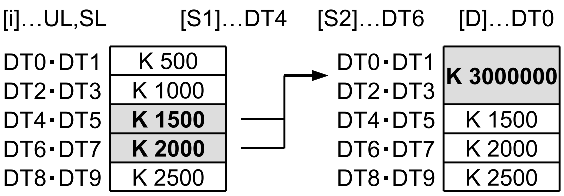
Operation unit	Calculation target data	Calculation result	
US, SS	16-bit × 16-bit	32 bit	Stored in a two-word area that starts with [D]
UL, SL	32-bit × 32-bit	64 bit	Stored in a four-word area that starts with [D]
SF	32-bit × 32-bit	32 bit	Stored in a two-word area that starts with [D]
DF	64-bit × 64-bit	64 bit	Stored in a four-word area that starts with [D]

■ Processing

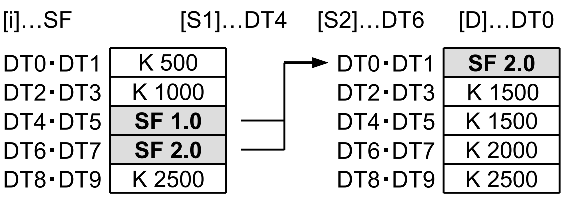
Example 1) Operation unit: 16 bits (US, SS)



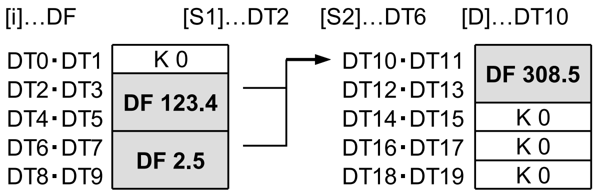
Example 2) Operation unit: 32 bits (UL, SL)



Example 3) Operation unit: 32 bits (SF)



Example 4) Operation unit: 64 bits (DF)



■ Precautions for programming

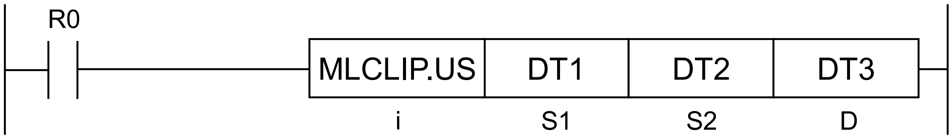
When the operation units are US, SS, UL, or SL, the area where the operation result is stored has twice the size of the operation unit. Allocate the memory area so that other areas will not be overwritten.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when a non-real number is specified for [S1] or [S2], and the operation unit is a real number (SF).

6.4 MLCLIP (Saturation Multiplication)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant)
S2	Calculation target data 2 (device address or constant)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K (Note 3)	U (Note 4)	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 4) Can be specified only when the operation unit is unsigned integer (US, UL).

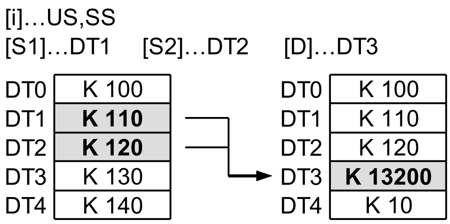
■ Outline of operation

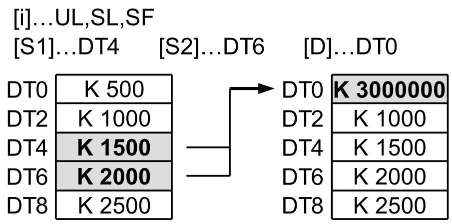
- This instruction multiplies the values [S1] and [S2] according to the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [S1] × [S2] → [D]
- As for the unsigned operation, if the result exceeds the operation unit, it is corrected to the maximum value.

- As for the signed operation, if the result exceeds the operation unit, it is corrected to the maximum or minimum value.

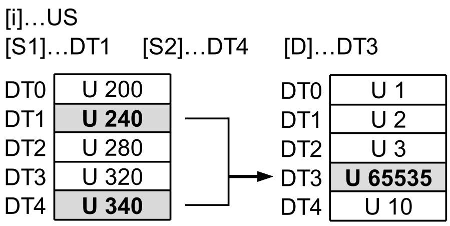
■ Processing

Example 1) Operation unit: 16 bits (US, SS) Example 2) Operation unit: 32 bits (UL, SL, SF)





Example 3) When the operation unit is unsigned 16-bit (US) and exceeds the maximum value

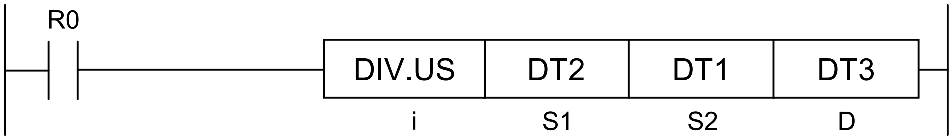


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
SR9 (CY)	To be set when the result is corrected, and cleared when it is not corrected.

6.5 DIV (Division)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant)
S2	Calculation target data 2 (device address or constant)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

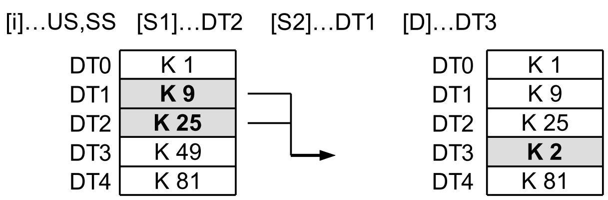
(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

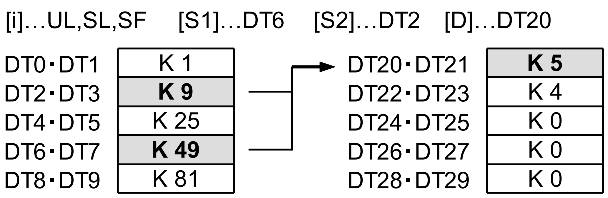
- This instruction divides the value [S1] with [S2] according to the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [S1] / [S2] → Quotient ([D])
- The remainder is not output. If remainder is necessary, use the DIVMOD instruction.

■ Processing

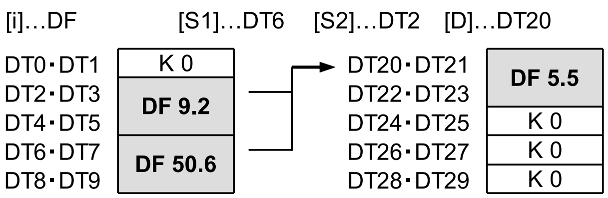
Example 1) Operation unit: 16 bits (US, SS)



Example 2) Operation unit: 32 bits (UL, SL, SF)



Example 3) Operation unit: 64 bits (DF)

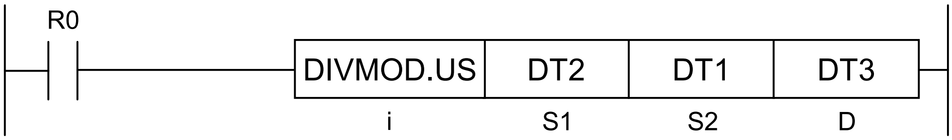


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S1] or [S2], and the operation unit is a real number (SF).
	To be set when '0' is specified for [S2].

6.6 DIVMOD (Division (With a remainder))

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant)
S2	Calculation target data 2 (device address or constant)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K (Note 3)	U (Note 4)	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is signed integer (SS, SL).

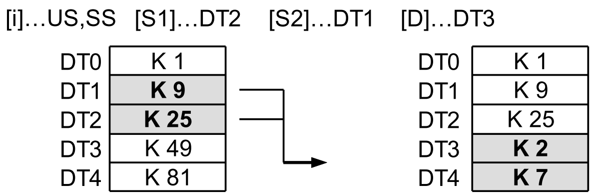
(Note 4) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

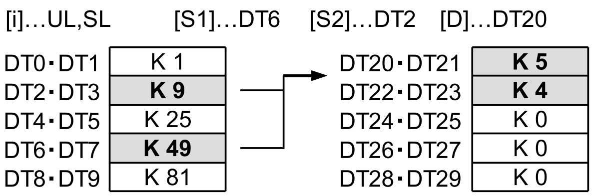
- This instruction divides the value [S1] with [S2] according to the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [S1] / [S2] → Quotient ([D]), Remainder ([D]+1)
- For 16-bit operation, the remainder should be specified as [D+1]. For 32-bit operation, the remainder should be specified as [D+2, D+3].

■ Processing

Example 1) Operation unit: 16 bits (US, SS)



Example 2) Operation unit: 32 bits (UL, SL)

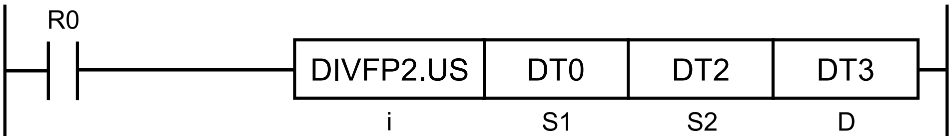


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S1] or [S2], and the operation unit is a real number (SF).
	To be set when '0' is specified for [S2].

6.7 DIVFP2 (Division (FP2 Compatible))

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant)
S2	Calculation target data 2 (device address or constant)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

■ Outline of operation

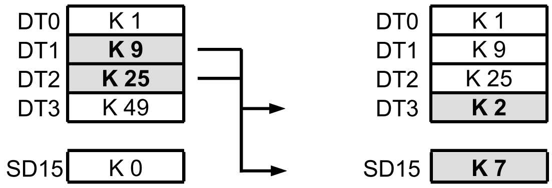
- This instruction divides [S1] by the value of [S2], and stores the quotient in [D] and the remainder in the system data register (SD).

- Calculation results are stored as follows according to the operation unit [i].
  US·SS: (S1) ÷ (S2) → Quotient (D), Remainder (SD15)
  UL·SL: (S1+1, S1) ÷ (S2+1, S2) → Quotient (D+1, D), Remainder (SD16, SD15)
- When an interrupt PB and interrupt program (INTPG) occur, SD15 and SD16 will be automatically removed and then returned.
  Therefore, SD15 and SD16 will not lose their contents being operated in the main program even if the main program is interrupted by other programs containing this command. But remember to complete using SD15 and SD16 by the end of the interrupt program.

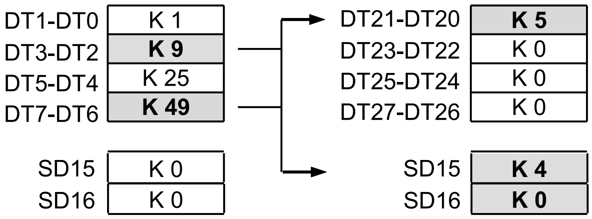
■ Processing

Divide the calculation target data 1 by the target data 2, and set the calculation result data and remainder.

Example 1) Operation unit: 16 bits (US, SS)
[S1]...DT2 [S2]...DT1 [D]...DT3



Example 2) Operation unit: 32 bits (UL, SL)
[S1]...DT6 [S2]...DT2 [D]...DT20



■ Precautions for programming

When a division overflow occurs, the data is output as follows. (In the case of the minimum negative number/-1)

Operation unit SS:	[S1] = -32768(H8000)	[S2] = -1(HFFFF)
	[D] = -32768(H8000)	[SD15] = 0(H0000)
Operation unit SL:	[S1+1, S1] = -2147483648(H80000000)	[S2+1, S2] = -1(HFFFFFFFF)
	[D+1, D] =2147483648(H80000000)	[SD16, SD15] =0(H00000000)

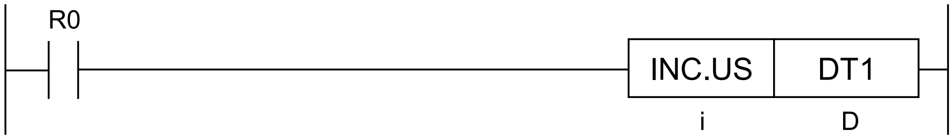
■ Flag operations

Name	Description
SR7	To be set in case of out-of-range values in indirect access (index modification).
SR8	To be set when '0' is specified for [S2].

Name	Description
(ER)	

6.8 INC (Increment)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

■ Outline of operation

- This instruction adds 1 to the value of [D] by the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [D] + 1 → [D]

■ Processing

Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [D]…DT1

DT0	K 100		DT0	K 100
DT1	**K 110**	→	DT1	**K 111**
DT2	K 120		DT2	K 120
DT3	K 130		DT3	K 130

Example 2) When the operation unit is 32-bit (UL, SL, SF)

[i]…UL,SL,SF [D]…DT2

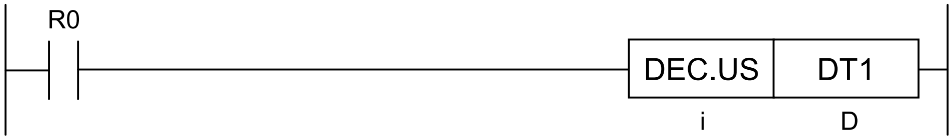
DT0·DT1	K 500		DT0·DT1	K 500
DT2·DT3	**K 1000**	→	DT2·DT3	**K 1001**
DT4·DT5	K 1500		DT4·DT5	K 1500
DT6·DT7	K 2000		DT6·DT7	K 2000

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [D], and the operation unit is a real number (SF).

6.9 DEC (Decrement)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
D	Calculation target data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

■ Outline of operation

- This instruction subtracts 1 from the value of [D] by the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [D] - 1 → [D]

■ Processing

Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [D]…DT1

	Before		After
DT0	K 100	DT0	K 100
DT1	**K 110**	DT1	**K 109**
DT2	K 120	DT2	K 120
DT3	K 130	DT3	K 130

Example 2) When the operation unit is 32-bit (UL, SL, SF)

[i]…UL,SL,SF [D]…DT2

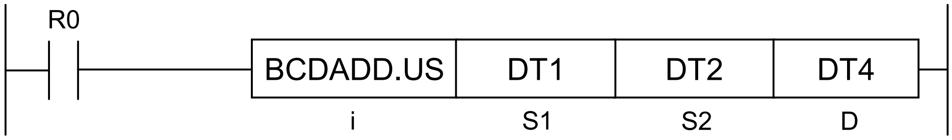
	Before		After
DT0·DT1	K 500	DT0·DT1	K 500
DT2·DT3	**K 1000**	DT2·DT3	**K 999**
DT4·DT5	K 1500	DT4·DT5	K 1500
DT6·DT7	K 2000	DT6·DT7	K 2000

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [D], and the operation unit is a real number (SF).

6.10 BCDADD (BCD Data Addition)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant) (Available data: H0 to H9 for each digit)
S2	Calculation target data 2 (device address or constant) (Available data: H0 to H9 for each digit)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, and 32-bit devices can be modified. (Integer constants, real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

- This instruction adds the BCD data for [S1] and [S2] by the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [S1] + [S2] → [D]

■ Processing

Example 1) Operation unit: 16 bits (US)

[i]…US [S1]…DT1 [S2]…DT2 [D]…DT4

DT0	H 0500	DT0	H 0500
DT1	**H 0510**	DT1	H 0510
DT2	**H 0520**	DT2	H 0520
DT3	H 0530	DT3	H 0530
DT4	H 0540	DT4	**H 1030**

Example 2) Operation unit: 32 bits (UL)

[i]…UL [D1]…DT4 [D2]…DT6 [D]…DT0

DT0·DT1	H 00005000	DT0·DT1	**H 00012500**
DT2·DT3	H 00005500	DT2·DT3	H 00005500
DT4·DT5	**H 00006000**	DT4·DT5	H 00006000
DT6·DT7	**H 00006500**	DT6·DT7	H 00006500
DT8·DT9	H 00007000	DT8·DT9	H 00007000

Example 3) Operation unit: 16 bits (US)

[i]…US [S1]…DT1 [S2]…DT4 [D]…DT3

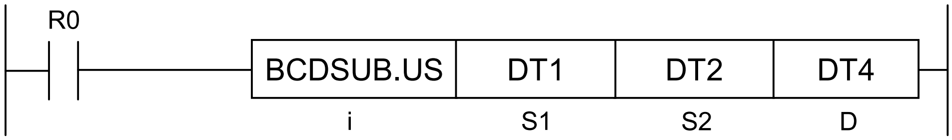
DT0	H 5000	DT0	H 5000
DT1	**H 6000**	DT1	H 6000
DT2	H 7000	DT2	H 7000
DT3	H 8000	DT3	**H 5000**
DT4	**H 9000**	DT4	H 9000

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when data other than BCD are specified for the calculation target data [S1] or [S2].

6.11 BCDSUB (BCD Data Subtraction)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant) (Available data: H0 to H9 for each digit)
S2	Calculation target data 2 (device address or constant) (Available data: H0 to H9 for each digit)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, and 32-bit devices can be modified. (Integer constants, real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

- This instruction subtracts the BCD data for [S1] from [S2] by the operation unit [i].
- The calculation result is stored in the address starting with [D].

  [S1] - [S2] → [D]

■ Processing

Example 1) Operation unit: 16 bits (US)

[i]…US [S1]…DT1 [S2]…DT2 [D]…DT4

	Before		After
DT0	H 0540	DT0	H 0540
DT1	**H 0530**	DT1	H 0530
DT2	**H 0520**	DT2	H 0520
DT3	H 0510	DT3	H 0510
DT4	H 0500	DT4	**H 0010**

Example 2) Operation unit: 32 bits (UL)

[i]…UL [D1]…DT4 [D2]…DT6 [D]…DT0

	Before		After
DT0·DT1	H 00020000	DT0·DT1	**H 00000500**
DT2·DT3	H 00019500	DT2·DT3	H 00019500
DT4·DT5	**H 00019000**	DT4·DT5	H 00019000
DT6·DT7	**H 00018500**	DT6·DT7	H 00018500
DT8·DT9	H 00018000	DT8·DT9	H 00018000

Example 3) Operation unit: 16 bits (US)

[i]…US [S1]…DT1 [S2]…DT4 [D]…DT3

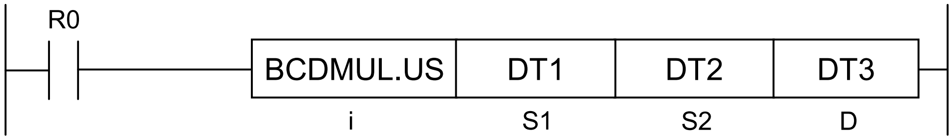
	Before		After
DT0	H 1000	DT0	H 1000
DT1	**H 2000**	DT1	H 2000
DT2	H 3000	DT2	H 3000
DT3	H 4000	DT3	**H 7000**
DT4	**H 5000**	DT4	H 5000

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when data other than BCD are specified for the calculation target data [S1] or [S2].

6.12 BCDMUL (BCD Data Multiplication)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant) (Available data: H0 to H9 for each digit)
S2	Calculation target data 2 (device address or constant) (Available data: H0 to H9 for each digit)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, and 32-bit devices can be modified. (Integer constants, real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

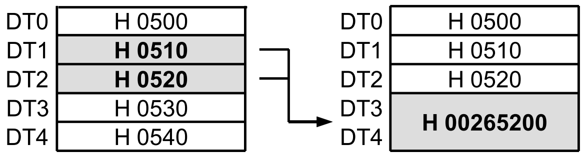
■ Outline of operation

- This instruction multiplies the BCD data for [S1] and [S2] by the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [S1] × [S2] → ([D], [D]+1)
- The unit size of the operation result output in [D] is twice as large as the operation unit.

■ Processing

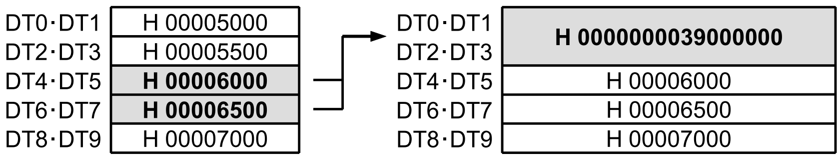
Example 1) Operation unit: 16 bits (US)

[i]…US [S1]…DT1 [S2]…DT2 [D]…DT3



Example 2) Operation unit: 32 bits (UL)

[i]…UL [S1]…DT4 [S2]…DT6 [D]…DT0



■ Precautions for programming

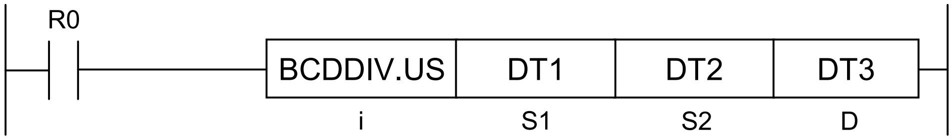
If the ending area of the operation device is specified for [D], the memory for the subsequent device may be overwritten by an area twice the size of the operation unit.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when data other than BCD are specified for the calculation target data [S1] or [S2].

6.13 BCDDIV (BCD Data Division)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant) (Available data: H0 to H9 for each digit)
S2	Calculation target data 2 (device address or constant) (Available data: H0 to H9 for each digit)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, and 32-bit devices can be modified. (Integer constants, real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

- This instruction divides [S1] with the BCD data for [S2] by the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [S1] / [S2] → Quotient ([D]), Remainder ([D]+1)
- The remainder is stored in ([D]+1).

■ Processing

Example 1) Operation unit: 16 bits (US)

[i]…US [S1]…DT1 [S2]…DT2 [D]…DT3

	Before			After
DT0	H 0064		DT0	H 0064
DT1	**H 0049**		DT1	H 0049
DT2	**H 0025**	→	DT2	H 0025
DT3	H 0009		DT3	**H 0001**
DT4	H 0001		DT4	**H 0024**

Example 2) Operation unit: 32 bits (UL)

[i]…UL [S1]…DT4 [S2]…DT6 [D]…DT0

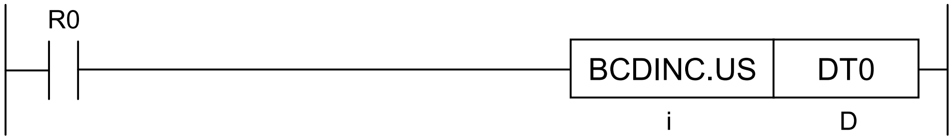
	Before			After
DT0·DT1	H 11111100		DT0·DT1	**H 00033333**
DT2·DT3	H 22222200		DT2·DT3	**H 00000300**
DT4·DT5	**H 33333300**	→	DT4·DT5	H 33333300
DT6·DT7	**H 00001000**		DT6·DT7	H 00001000
DT8·DT9	H 44444400		DT8·DT9	H 44444400

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when '0' is specified for [S2].
	To be set when data other than BCD are specified for the calculation target data [S1] or [S2].

6.14 BCDINC (BCD Data Increment)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
D	Calculation target data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, and 32-bit devices can be modified. (Integer constants, real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

- This instruction adds 1 to the BCD data of [D] by the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [D] + 1 → [D]

■ Processing

Example 1) Operation unit: 16 bits (US)

[i]…US [D]…DT0

	Before			After
DT0	**H 0100**	→	DT0	**H 0101**
DT1	H 0200		DT1	H 0200
DT2	H 0300		DT2	H 0300
DT3	H 0400		DT3	H 0400

Example 2) Operation unit: 32 bits (UL)

[i]…UL [D]…DT2

	Before			After
DT0·DT1	H 01000000		DT0·DT1	H 01000000
DT2·DT3	**H 01999999**	→	DT2·DT3	**H 02000000**
DT4·DT5	H 03000000		DT4·DT5	H 03000000
DT6·DT7	H 04000000		DT6·DT7	H 04000000

Example 3) Operation unit: 16 bits (US)

[i]…US [D]…DT0

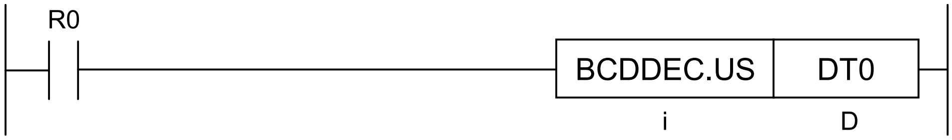
	Before			After
DT0	**H 9999**	→	DT0	**H 0000**
DT1	H 0200		DT1	H 0200
DT2	H 0300		DT2	H 0300
DT3	H 0400		DT3	H 0400

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when data other than BCD are specified for the calculation target data [D].

6.15 BCDDEC (BCD Data Decrement)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
D	Calculation target data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, and 32-bit devices can be modified. (Integer constants, real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

- This instruction subtracts 1 from the BCD data of [D] by the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [D] - 1 → [D]

■ Processing

Example 1) Operation unit: 16 bits (US)

[i]…US [D]…DT0

	Before			After
DT0	**H 0100**	→	DT0	**H 0099**
DT1	H 0200		DT1	H 0200
DT2	H 0300		DT2	H 0300
DT3	H 0400		DT3	H 0400

Example 2) Operation unit: 32 bits (UL)

[i]…UL [D]…DT2

	Before			After
DT0·DT1	H 01000000		DT0·DT1	H 01000000
DT2·DT3	**H 02000000**	→	DT2·DT3	**H 01999999**
DT4·DT5	H 03000000		DT4·DT5	H 03000000
DT6·DT7	H 04000000		DT6·DT7	H 04000000

Example 3) Operation unit: 16 bits (US)

[i]…US [D]…DT0

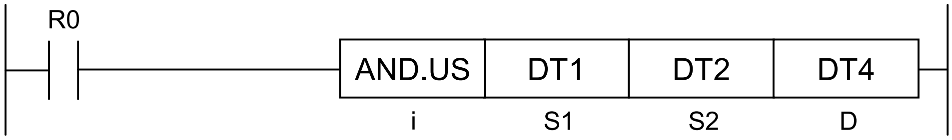
	Before			After
DT0	**H 0000**	→	DT0	**H 9999**
DT1	H 0200		DT1	H 0200
DT2	H 0300		DT2	H 0300
DT3	H 0400		DT3	H 0400

■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).
SR8 (ER)	To be set when data other than BCD are specified for the calculation target data [D].

6.16 AND (Logical Conjunction)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant)
S2	Calculation target data 2 (device address or constant)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K (Note 3)	U (Note 4)	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 4) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

- This instruction calculates the logical conjunction for [S1] and [S2] by the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [S1] ∧ [S2] →[D]

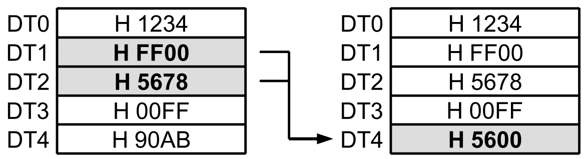
Logical conjunction (AND)

[S1] bit	[S2] bit	Logical conjunction
0	0	0
0	1	0
1	0	0
1	1	1

Processing

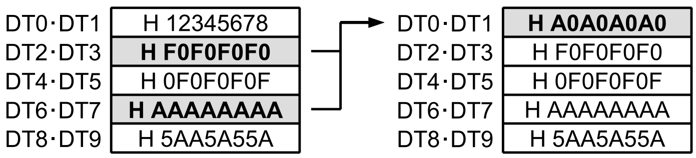
Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S1]…DT1 [S2]…DT2 [D]…DT4



Example 2) Operation unit: 32 bits (UL, SL)

[i]…UL,SL [S1]…DT2 [S2]…DT6 [D]…DT0

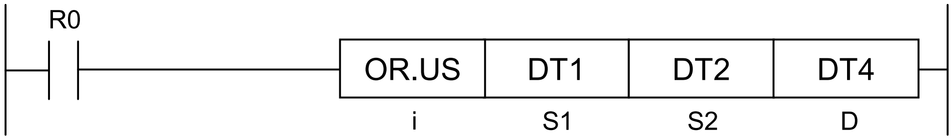


Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

6.17 OR (Logical Disjunction)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant)
S2	Calculation target data 2 (device address or constant)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K (Note 3)	U (Note 4)	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 4) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

- This instruction calculates the logical disjunction for [S1] and [S2] by the operation unit [i].
- The calculation result is stored in the address starting with [D].
  [S1] ∨ [S2] →[D]

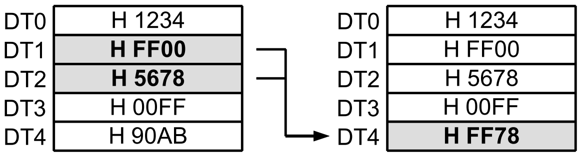
■ Logical disjunction (OR)

[S1] bit	[S2] bit	Logical disjunction
0	0	0
0	1	1
1	0	1
1	1	1

■ Processing

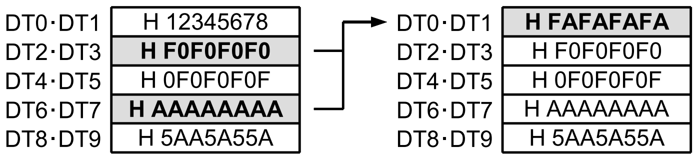
Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S1]…DT1 [S2]…DT2 [D]…DT4



Example 2) Operation unit: 32 bits (UL, SL)

[i]…UL,SL [S1]…DT2 [S2]…DT6 [D]…DT0

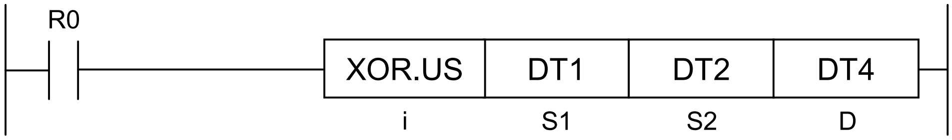


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

6.18 XOR (Exclusive OR)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant)
S2	Calculation target data 2 (device address or constant)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K (Note 3)	U (Note 4)	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 4) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

- This instruction calculates the exclusive OR for [S1] and [S2] by the operation unit [i].
- The calculation result is stored in the address starting with [D].
  { [S1] ∧ /[S2] } ∨ { /[S1] ∧ [S2] } → [D]

■ Exclusive OR (XOR)

[S1] bit	[S2] bit	Exclusive OR
0	0	0
0	1	1
1	0	1
1	1	0

■ Processing

Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S1]…DT1 [S2]…DT2 [D]…DT4

DT0	H 1234	DT0	H 1234
DT1	**H FF00**	DT1	H FF00
DT2	**H 5678**	DT2	H 5678
DT3	H 00FF	DT3	H 00FF
DT4	H 90AB	DT4	**H A978**

Example 2) Operation unit: 32 bits (UL, SL)

[i]…UL,SL [S1]…DT2 [S2]…DT6 [D]…DT0

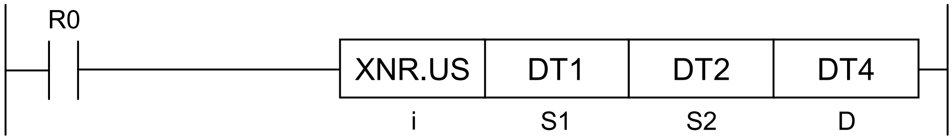
DT0·DT1	H 12345678	DT0·DT1	**H 5A5A5A5A**
DT2·DT3	**H F0F0F0F0**	DT2·DT3	H F0F0F0F0
DT4·DT5	H 0F0F0F0F	DT4·DT5	H 0F0F0F0F
DT6·DT7	**H AAAAAAAA**	DT6·DT7	H AAAAAAAA
DT8·DT9	H 5AA5A55A	DT8·DT9	H 5AA5A55A

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

6.19 XNR (Exclusive NOR)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S1	Calculation target data 1 (device address or constant)
S2	Calculation target data 2 (device address or constant)
D	Calculation result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K (Note 3)	U (Note 4)	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 4) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

- This instruction calculates the exclusive NOR for [S1] and [S2] by the operation unit [i].
- The calculation result is stored in the address starting with [D].
  { [S1] ∧ [S2] } ∨ { /[S1] ∧ /[S2] } → [D]

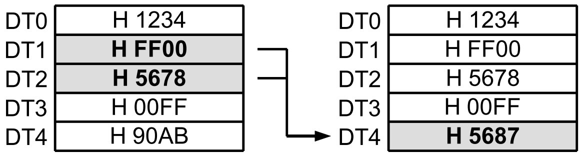
■ Exclusive NOR (XNR)

[S1] bit	[S2] bit	Exclusive NOR
0	0	1
0	1	0
1	0	0
1	1	1

■ Processing

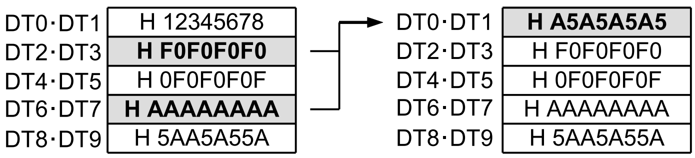
Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S1]…DT1 [S2]…DT2 [D]…DT4



Example 2) Operation unit: 32 bits (UL, SL)

[i]…UL,SL [S1]…DT2 [S2]…DT6 [D]…DT0

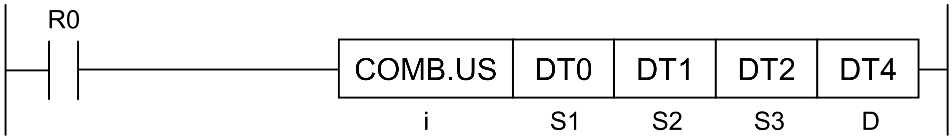


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

6.20 COMB (Combination)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S1	Combined data 1 (device address or constant)
S2	Combined data 2 (device address or constant)
S3	Combination mask data (device address or constant)
D	Combination result data (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K (Note 3)	U (Note 4)	H	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●				●
S3	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 4) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

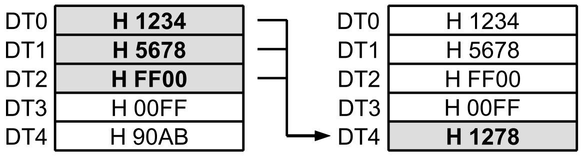
- This instruction combines the data for [S1] and [S2] by the operation unit [i] and the mask data stored in [S3].

- The calculation result is stored in the address starting with [D].
- When the mask data specified for [S3] is ON for the bit, the combination starts from [S1]. When it is OFF, the combination starts from [S2].
  { [S1] ∧ [S3] } ∨ { [S2] ∧ /[S3] } → [D]

■ Processing

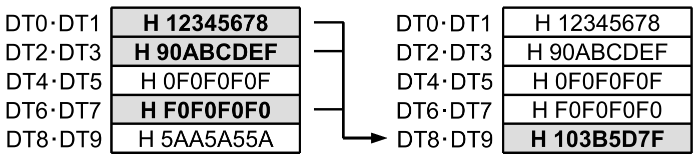
Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S1]…DT0 [S2]…DT1 [S3]…DT2 [D]…DT4



Example 2) Operation unit: 32 bits (UL, SL)

[i]…UL,SL [S1]…DT0 [S2]…DT2 [S3]…DT6 [D]…DT8



■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

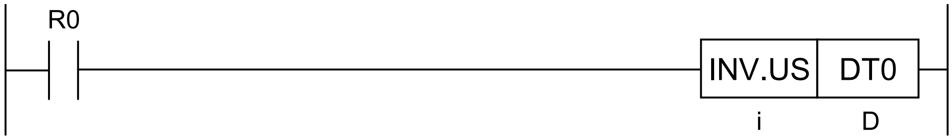
(MEMO)

7 High-level Instructions (Data Conversion)

Applicable Models: All Models

7.1 INV (Data Inversion)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
D	Device address where the data to be inverted are stored

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

- This instruction logically inverts the device address value specified by [D].
  /[D] → [D]

■ Processing

Example 1) Operation unit: 16 bits (US, SS) Example 2) Operation unit: 32 bits (UL, SL)

[i]…US,SS [D]…DT2

	Before		After
DT0	H 0011	DT0	H 0011
DT1	H 2233	DT1	H 2233
DT2	**H 4455**	DT2	**H BBAA**
DT3	H 6677	DT3	H 6677
DT4	H 8899	DT4	H 8899

[i]…UL,SL [D]…DT2

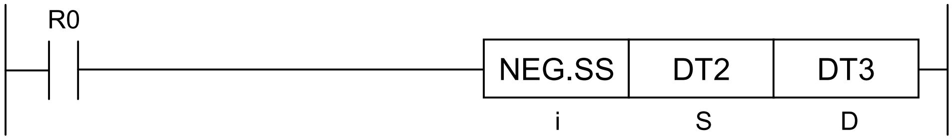
	Before		After
DT0·DT1	H 11223344	DT0·DT1	H 11223344
DT2·DT3	**H 55667788**	DT2·DT3	**H AA998877**
DT4·DT5	H 9900AABB	DT4·DT5	H 9900AABB
DT6·DT7	H CCDDEEFF	DT6·DT7	H CCDDEEFF
DT8·DT9	H 12345678	DT8·DT9	H 12345678

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

7.2 NEG (Sign Inversion)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i			●		●		

■ List of operands

Operand	Description
S	The device address where the data whose sign is to be inverted are stored, or the constant
D	Storage device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

- This instruction calculates two's complement for the device address value or the constant specified by [S], to invert the sign of the data.
- The calculation result is stored in the device address specified by [D].

Processing

Example 1) Operation unit: 16 bits (SS) Example 2) Operation unit: 32 bits (SL)

[i]…SS [S]…DT2 [D]…DT3

	[S]		[D]
DT0	K 100	DT0	K 100
DT1	K 110	DT1	K 110
DT2	**K 120**	DT2	K 120
DT3	K 130	DT3	**K -120**
DT4	K 140	DT4	K 140

[i]…SL [S]…DT2 [D]…DT8

	[S]		[D]
DT0·DT1	K 500	DT0·DT1	K 500
DT2·DT3	**K 1000**	DT2·DT3	K 1000
DT4·DT5	K 1500	DT4·DT5	K 1500
DT6·DT7	K 2000	DT6·DT7	K 2000
DT8·DT9	K 2500	DT8·DT9	**K -1000**

Precautions for programming

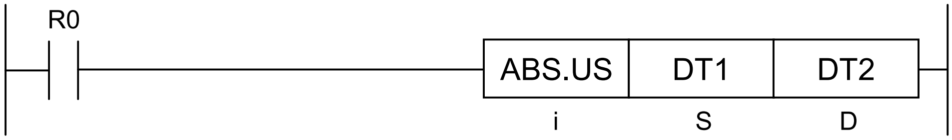
- The result should be the minimum negative value if it has been specified.

Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

7.3 ABS (Absolute Value)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S	The device address where the data for taking an absolute value are stored, or the constant
D	Storage device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

- This instruction takes the absolute value of the device address or the constant specified by [S], and stores it in the device address specified by [D].

■ Processing

Example 1) Operation unit: Signed 16 bits (SS) Example 2) Operation unit: Signed 32 bits (SL)

[i]…SS [S]…DT1 [D]…DT2

	Before		After
DT0	K-100	DT0	K-100
DT1	**K-110**	DT1	K-110
DT2	K-120	DT2	**K110**
DT3	K-130	DT3	K-130
DT4	K-140	DT4	K-140

[i]…SL [S]…DT2 [D]…DT4

	Before		After
DT0·DT1	K-500	DT0·DT1	K-500
DT2·DT3	**K-1000**	DT2·DT3	K-1000
DT4·DT5	K-1500	DT4·DT5	**K1000**
DT6·DT7	K-2000	DT6·DT7	K-2000
DT8·DT9	K-2500	DT8·DT9	K-2500

■ Precautions for programming

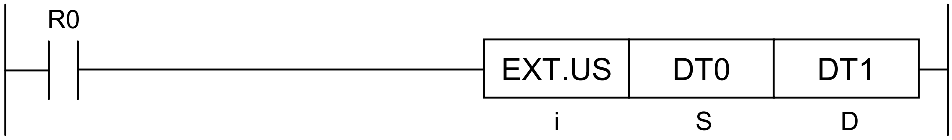
- The same value for unsigned integers (US, UL) is stored.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the minimum negative value is specified for [S].

7.4 EXT (Sign Extension)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S	The device address where the data for sign extension are stored, or the constant
D	Storage device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

- This instruction performs sign extension of a value of the device address or the constant specified by [S], and stores it in the device address specified by [D].

Processing

Example 1) Operation unit: Signed 16 bits (SS)

[i]…SS

[S]…DT0 [D]…DT0

DT0	K -2(H FFFE)
DT1	**K 0(H 0000)**

→ DT0·DT1 K-2(H **FFFF**FFFE)

Example 2) Operation unit: Unsigned 16 bits (US)

[i]…US

[S]…DT0 [D]…DT0

DT0	H FFFE
DT1	**H 1234**

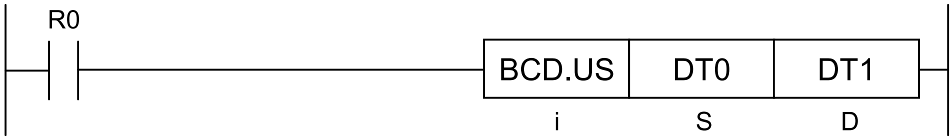
→ DT0·DT1 **H 0000**FFFFE

Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

7.5 BCD (Conversion: BCD Data)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
S	The device address where the binary data to be converted is stored, or the constant
D	Device address to store the conversion result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●								●	●				●
D	●	●	●	●			●	●													●

■ Outline of operation

This instruction converts the device address value or the constant specified by [S] from binary data to BCD data, and stores the result in the device specified by [D].

■ Processing

Example 1) Operation unit: 16 bits (US)

[i]…US

[S]…DT0 [D]…DT10

DT0	H 0010	(K 16) →	DT10	H 0016
DT1			DT11	

Example 2) Operation unit: 32 bits (UL)

[i]…UL

[S]…DT0 [D]…DT10

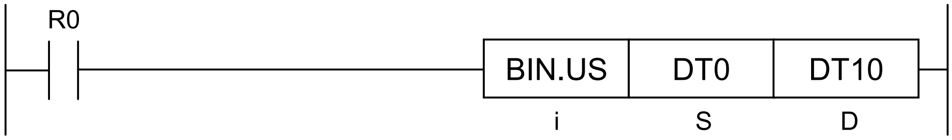
DT0	H 614E	(K 12345678) →	DT10	H 5678
DT1	H 00BC		DT11	H 1234
DT2			DT12	

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the binary data exceed the BCD-convertible range. (Example: When US exceeds K 9999 or UL exceeds K 99999999)

7.6 BIN (Conversion: BCD → BIN)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
S	The device address where the BCD data to be converted is stored, or the constant
D	Device address to store the conversion result

■ Available devices (●: Available)

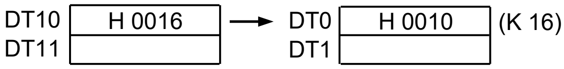
Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●								●	●				●
D	●	●	●	●			●	●													●

■ Outline of operation

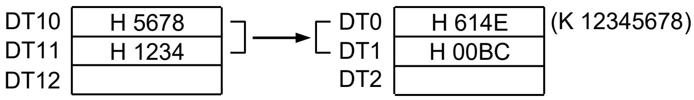
This instruction converts the device address value or the constant specified by [S] from BCD data to binary data, and stores the result in the device address specified by [D].

Processing

Example 1) Operation unit: 16 bits (US)
[i]…US
[S]…DT0 [D]…DT10



Example 2) Operation unit: 32 bits (UL)
[i]…UL
[S]…DT0 [D]…DT10

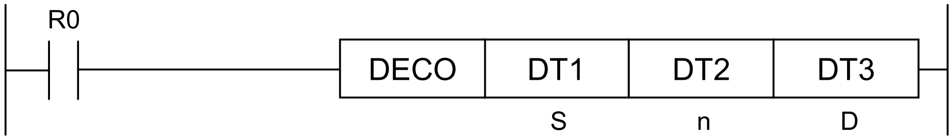


Flag operations

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td rowspan="2">SR7
SR8
(ER)</td><td>To be set in the case of out-of-range in indirect access (index modification).</td></tr>
<tr><td>To be set when [S] is not BCD data.</td></tr>
</table>

7.7 DECO (Decoding)

■ Ladder diagram



■ List of operands

Operand	Description
S	The device address where the data for decoding are stored, or the constant
n	The device address where the control data (specification of the conversion starting bit, specification of the conversion-enabled bit length) are stored, or the constant
D	Storage device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●
n	●	●	●	●	●	●	●	●	●	●	●						●				●
D	●	●	●	●			●	●	●		●										●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

- This instruction decodes a value of the device address or the constant specified by [S], and stores the result in the device address specified by [D].
- The part to be decoded is specified by control data [n].
- The required length of the device address for storing the result depends on the length of the data before decoding.

■ Specification of control data [N]

15 12 11 8 7 4 3 0

n: | ···· | 0000 | ···· | 0000 |

Ignore the bits indicated with '·'.

nH (Specify the conversion start bit)
Bit 0: H0
|
Bit 15: HF

nL (Specify the conversion-enabled bit length)
1 bit: H1
|
8 bits: H8

■ Conversion example

Conversion data	Decoded result (16 bits)	Conversion data	Decoded result (16 bits)
0000	0000 0000 0000 0001	1000	0000 0001 0000 0000
0001	0000 0000 0000 0010	1001	0000 0010 0000 0000
0010	0000 0000 0000 0100	1010	0000 0100 0000 0000
0011	0000 0000 0000 1000	1011	0000 1000 0000 0000
0100	0000 0000 0001 0000	1100	0001 0000 0000 0000
0101	0000 0000 0010 0000	1101	0010 0000 0000 0000
0110	0000 0000 0100 0000	1110	0100 0000 0000 0000
0111	0000 0000 1000 0000	1111	1000 0000 0000 0000

■ Specification of nL and length of operation result

Value of nL Effective bit length for conversion	Occupancy length of the decoded result	Enabled bit length of the decoded result	Value other than the enabled bit length in [D]
1	1 words	2	0
2	1 words	4	0
3	1 words	8	0
4	1 words	16	-
5	2 words	32	-
6	4 words	64	-
7	8 words	128	-
8	16 words	256	-

■ Processing

Example) Decode 4 bits from Bit 4

[S]…DT1 [n]…H 0404 [D]…DT2

* Store the result of decoding the specified portion (“0111 ”=7) into the 16-bit (2^4-bit) device address starting with DT2.

* The 16-bit area starting with DT2 turns ON, while the other bits become '0'.

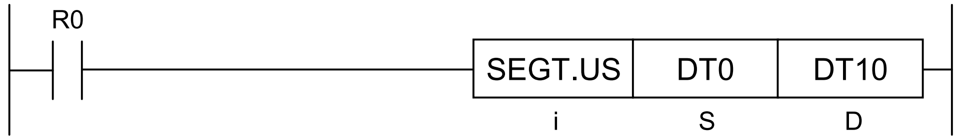
```
      15            4     0              15          7        0
DT1  | 0000 0000 0111 0000 |  ------>  DT2 | 0000 0000 1000 0000 |
                 4 bits                             16 bits
```

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the conversion-enabled bit length "nL" is not in the following range: 1 ≤ nL ≤ 8.
	To be set when the sum of the conversion starting bit number (nH) and the conversion-enabled bit length (nL) is not in the following range: 1 ≤ "nH + nL" ≤ 16.
	To be set if, when the decoded result is stored in the device address specified by [D], it exceeds the area.

7.8 SEGT (7-Segment Decoding)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●					

■ List of operands

Operand	Description
S	The device address where the data for decoding are stored, or the constant
D	Storage device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

● This instruction converts the device address value or a constant specified by [S] to 7-segment data in 4 digits, and stores it to an address of 2 words beginning with [D].

■ Processing

Example) H ABCD is stored in [S]

[S]…DT0 [D]…DT10

DT0 H ABCD →

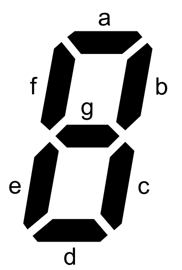
DT10	0011	1001	0101	1110
DT11	0111	0111	0111	1100

■ Precautions for programming

- If an unsigned constant U is specified for [S], it should be converted as HEX data.

■ Notation and corresponding data

Value	Conversion data 1 digit [S]				1-digit data for 7-segment notation [D]								7-segment display
						g	f	e	d	c	b	a	
0	0	0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	0	0	0	0	1	1	0	1
2	0	0	1	0	0	1	0	1	1	0	1	1	2
3	0	0	1	1	0	1	0	0	1	1	1	1	3
4	0	1	0	0	0	1	1	0	0	1	1	0	4
5	0	1	0	1	0	1	1	0	1	1	0	1	5
6	0	1	1	0	0	1	1	1	1	1	0	1	6
7	0	1	1	1	0	0	1	0	0	1	1	1	7
8	1	0	0	0	0	1	1	1	1	1	1	1	8
9	1	0	0	1	0	1	1	0	1	1	1	1	9
A	1	0	1	0	0	1	1	1	0	1	1	1	A
B	1	0	1	1	0	1	1	1	1	1	0	0	b
C	1	1	0	0	0	0	1	1	1	0	0	1	C
D	1	1	0	1	0	1	0	1	1	1	1	0	d
E	1	1	1	0	0	1	1	1	1	0	0	1	E
F	1	1	1	1	0	1	1	1	0	0	0	1	F



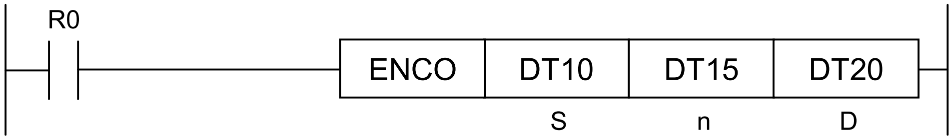
■ Flag operations

Name	Description
SR7 SR8	To be set in the case of out-of-range in indirect access (index modification).

Name	Description
(ER)	To be set if, when the conversion result is stored in the device address specified by [D], it exceeds the area.

7.9 ENCO (Encoding)

■ Ladder diagram



■ List of operands

Operand	Description
S	The device address where the data for encoding are stored, or the constant
n	The device address where control data (specification of the result output start bit, and specification of the conversion-enabled bit length) are stored, or the constant
D	Storage device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●										●
n	●	●	●	●	●	●	●	●	●	●	●						●				●
D	●	●	●	●			●	●	●		●										●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

- This instruction encodes a part of a device address value or a constant specified by [S], and stores the result to the device address specified by [D].
- The target section to be encoded is specified by the control data [n].
- If multiple bits are ON in the target section for encoding, the higher bit is enabled.
- Encode the content for 2nL bits, starting with the device address specified by [S]. The encoded result is stored as a decimal number within 8 bits from the bit specified by "nH".
- In the device address specified by [D], portions other than the one storing the conversion result should be 0 (should be set to 0).

Specification of control data [N]

	15 … 12	11 … 8	7 … 4	3 … 0
n:	····	0000	····	0000

Ignore the bits indicated with '·'.

$\underbrace{0000}_{nH}$ nH (Specify the result output start bit)
Bit 0: H0
|
Bit 15: HF

$\underbrace{0000}_{nL}$ nL (Specify the conversion-enabled bit length)
2^1 bits: H1
|
2^8 bits: H8

Conversion example

Data to be converted (16 bits)	Encoding result	Data to be converted (16 bits)	Encoding result
0000 0000 0000 0001	0000	0000 0001 0000 0000	1000
0000 0000 0000 0010	0001	0000 0010 0000 0000	1001
0000 0000 0000 0100	0010	0000 0100 0000 0000	1010
0000 0000 0000 1000	0011	0000 1000 0000 0000	1011
0000 0000 0001 0000	0100	0001 0000 0000 0000	1100
0000 0000 0010 0000	0101	0010 0000 0000 0000	1101
0000 0000 0100 0000	0110	0100 0000 0000 0000	1110
0000 0000 1000 0000	0111	1000 0000 0000 0000	1111

Specification of nL and length of result

Value of nL	Effective bit length for conversion	Value of nL	Effective bit length for conversion
1	2	5	32(2 word)
2	4	6	64(4 word)
3	8 (1 byte)	7	128(8 word)
4	16(1 word)	8	256(16 word)

Processing

Example) [S]…DT10 [n]…H 0005 [D]…DT20

Conversion-enabled bits are DT10 to DT11 (32 bits from DT10).
The bit numbers that are ON in these two-word area are stored in a decimal form from Bit 0 of DT20.

DT10	0000	0001	0000	0000
DT11	0000	0000	0000	0000

Bit 8, counted from the lowest bit of DT10, is ON.

↓

DT20	0000	0000	0000	1000

Store H8 in DT20.

Precautions for programming

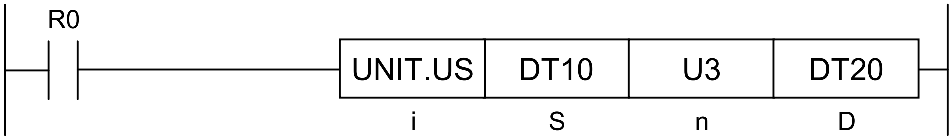
- If an unsigned constant U is specified for [S], it should be converted as Hex data.

Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the conversion-enabled bit length (nL) is not in the following range: 1 ≤ nL ≤ 8.
	To be set (for consistency) when the sum of the result output starting bit number "nH" and the conversion-enabled bit length "nL" is not in the following range: 1 ≤ nH + nL ≤ 16.
	To be set when the data to be encoded is all "0".

7.10 UNIT (Digit Unification)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●					

■ List of operands

Operand	Description
S	The device address where the data to be unified are stored, or the constant (data format: unsigned 16-bit integer)
n	The device address where the number of data to be unified is stored, or the constant (data format: unsigned 16-bit integer)
D	Storage device address (data format: according to the operation unit)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●										●
n	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

(Note 1) Only 16-bit devices, and integer constants can be modified. (32-bit devices, real number constants, and character constants cannot be specified.)

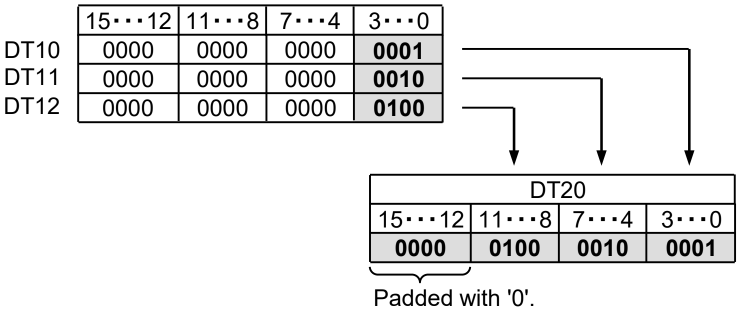
■ Outline of operation

- The lower 4 bits of the 16-bit data for [n] words starting with [S] are combined into 16-bit data.
- The available data amount specified by [n] is 0 to 4. No operation needed if [n]=0.
- The other portions of [D] are padded with "0".

■ Processing

[i]…US

[S]…DT10 [n]…U3 [D]…DT20

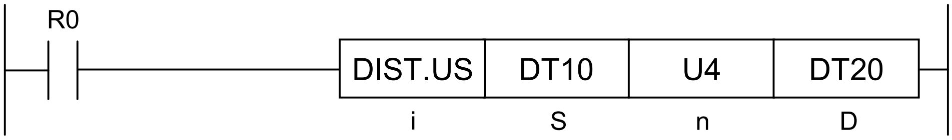


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [n], the number of data to be unified, is out of the specified range.

7.11 DIST (Digit Disintegration)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●					

■ List of operands

Operand	Description
S	The device address where the data to be broken down are stored, or the constant (data format: according to the operation unit)
n	The device address where the number of points into which the data is broken down is stored, or the constant (data format: unsigned 16-bit integer)
D	Storage device address (data format: unsigned 16-bit integer)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●					●	●				●
n	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

(Note 1) Only 16-bit devices, and integer constants can be modified. (32-bit devices, real number constants, and character constants cannot be specified.)

■ Outline of operation

- 16-bit device specified by [S] is broken down into 16-bit data by 4 bits. (The available range for [n], the number into which the data is broken down, is 0 to 4.)
- No operation needed if [n]=0.

■ Processing

Example) Operation unit: 16 bits (US)

[i]…US

[S]…DT10 [n]…U4 [D]…DT20

DT10			
15･･･12	11･･･8	7･･･4	3･･･0
0100	**0011**	**0001**	**0000**

DT20	0000	0000	0000	**0000**
DT21	0000	0000	0000	**0001**
DT22	0000	0000	0000	**0011**
DT23	0000	0000	0000	**0100**

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [n], the number into which the data is broken down, is out of the specified range.
	To be set when, if data equivalent to [n] is transferred from the address specified by [D], it exceeds the device address.

7.12 BUNI (Byte Data Unification)

■ Ladder diagram

■ List of operands

Operand	Description
S	The device address where the data to be unified are stored, or the constant (data format: unsigned 16-bit integer)
n	The device address where the number of data to be unified is stored, or the constant (data format: unsigned 16-bit integer) Available range: 0 to 65535
D	Storage device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●										●
n	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

(Note 1) Only 16-bit devices, and integer constants can be modified. (32-bit devices, real number constants, and character constants cannot be specified.)

(Note 2) Index register (I0 to IE)

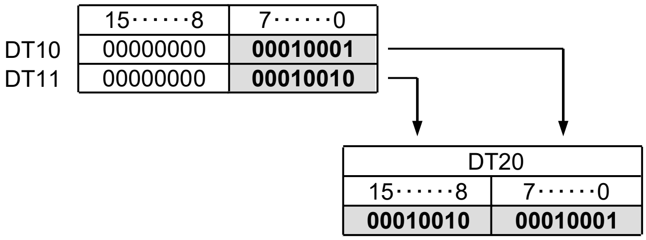
■ Outline of operation

- The target of this instruction is data of [n] words that begins with [S].
- The lower 1 byte of each word data is combined.
- The combined data is stored in a device area of [n] bytes that starts with [D].
- No operation needed if [n]=0.
- When [n] is an odd number, the value of the uppermost/highest byte of the storage device area is 0.

Processing

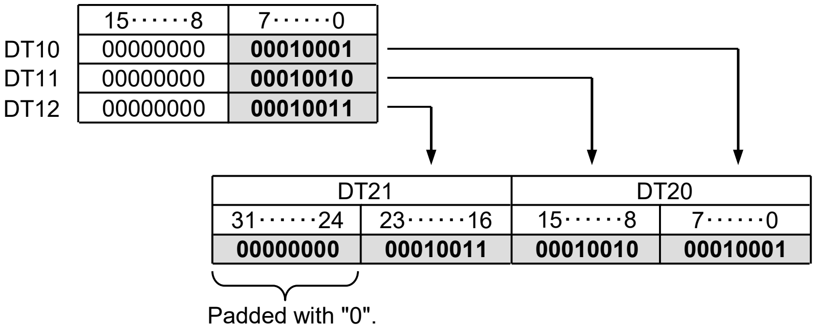
Example 1)

[S]…DT10 [n]…U2 [D]…DT20



Example 2)

[S]…DT10 [n]…U3 [D]…DT20



Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [n], the number of data to be unified, is out of the specified range.

7.13 BDIS (Byte Data Disintegration)

■ Ladder diagram

■ List of operands

Operand	Description
S	The device address where the data to be broken down are stored, or the constant
n	The device address where the number of points into which the data is broken down is stored, or the constant
D	Storage device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●					●	●				●
n	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

(Note 1) Only 16-bit devices, and integer constants can be modified. (32-bit devices, real number constants, and character constants cannot be specified.)

(Note 2) Index register (I0 to IE)

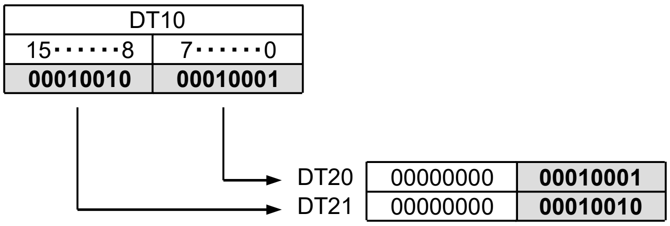
■ Outline of operation

- The target of this instruction is data of [n] words that begins with [S].
- Each word data is broken down into single bytes.
- The broken down data is stored in a 2 × [n]-word device area that starts with [D].
- It stores one byte for each word.
- No operation needed if [n]=0.

■ Processing

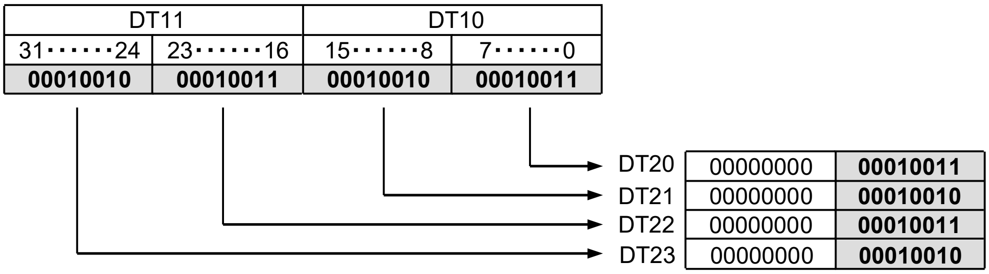
Example 1)

[S]…DT10 [n]…U2 [D]…DT20



Example 2)

[S]…DT10 [n]…U4 [D]…DT20

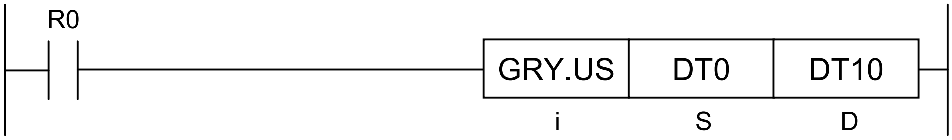


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [n], the number into which the data is broken down, is out of the specified range.
	To be set when, if data equivalent to [n] is transferred from the address specified by [D], it exceeds the device address.

7.14 GRY (Conversion: Binary → Gray Code)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
S	The device address where the data for conversion are stored, or the constant
D	Storage device address

(Note 1) For gray codes, refer to the "Correspondence Table: BIN / Gray Code."

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (US).

(Note 2) Only 16-bit devices, and integer constants can be modified. (32-bit devices, real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

- This instruction converts a value of the device address or the constant specified by [S] to gray code, and stores it in the device address specified by [D].

Processing

Example 1) Operation unit: 16 bits (US)

[i]…US

[S]…DT0 [D]…DT10

DT0 | 0000000000011000 | → DT10 | 0000000000010100 |

Example 2) Operation unit: 32 bits (UL)

[i]…UL

[S]…DT0 [D]…DT10

DT0, DT1 | 00000000000000000000000000011111 |

↓

DT10, DT11 | 00000000000000000000000000010000 |

Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

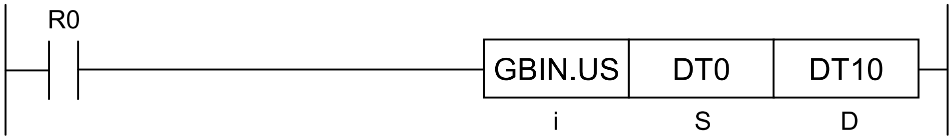
Correspondence Table: DEC / BIN / Gray Code

Decimal Decimal	Binary Binary	Gray code Gray code
0	0000 0000 0000 0000	0000 0000 0000 0000
1	0000 0000 0000 0001	0000 0000 0000 0001
2	0000 0000 0000 0010	0000 0000 0000 0011
3	0000 0000 0000 0011	0000 0000 0000 0010
4	0000 0000 0000 0100	0000 0000 0000 0110
5	0000 0000 0000 0101	0000 0000 0000 0111
6	0000 0000 0000 0110	0000 0000 0000 0101
7	0000 0000 0000 0111	0000 0000 0000 0100
8	0000 0000 0000 1000	0000 0000 0000 1100
9	0000 0000 0000 1001	0000 0000 0000 1101
10	0000 0000 0000 1010	0000 0000 0000 1111
11	0000 0000 0000 1011	0000 0000 0000 1110
12	0000 0000 0000 1100	0000 0000 0000 1010
13	0000 0000 0000 1101	0000 0000 0000 1011
14	0000 0000 0000 1110	0000 0000 0000 1001
15	0000 0000 0000 1111	0000 0000 0000 1000
16	0000 0000 0001 0000	0000 0000 0001 1000
17	0000 0000 0001 0001	0000 0000 0001 1001

Decimal **Decimal**	**Binary** **Binary**	**Gray code** **Gray code**
18	0000 0000 0001 0010	0000 0000 0001 1011
19	0000 0000 0001 0011	0000 0000 0001 1010
20	0000 0000 0001 0100	0000 0000 0001 1110
21	0000 0000 0001 0101	0000 0000 0001 1111
22	0000 0000 0001 0110	0000 0000 0001 1101
23	0000 0000 0001 0111	0000 0000 0001 1100
24	0000 0000 0001 1000	0000 0000 0001 0100
25	0000 0000 0001 1001	0000 0000 0001 0101
26	0000 0000 0001 1010	0000 0000 0001 0111
27	0000 0000 0001 1011	0000 0000 0001 0110
28	0000 0000 0001 1100	0000 0000 0001 0010
29	0000 0000 0001 1101	0000 0000 0001 0011
30	0000 0000 0001 1110	0000 0000 0001 0001
31	0000 0000 0001 1111	0000 0000 0001 0000
32	0000 0000 0010 0000	0000 0000 0011 0000
-	-	-
63	0000 0000 0010 1111	0000 0000 0010 0000
64	0000 0000 0100 1111	0000 0000 0110 0000
-	-	-
255	0000 00001111 1111	0000 0000 1000 0000

7.15 GBIN (Conversion: Gray Code → BIN)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
S	The device address where the data for conversion are stored, or the constant
D	Storage device address

(Note 1) For gray codes, refer to the "Correspondence Table: BIN / Gray Code."

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (US).

(Note 2) Only 16-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

- This instruction converts the gray code of a value of the device address or the constant specified by [S] to binary data, and stores it in the device address specified by [D].

■ Processing

Example 1) Operation unit: 16 bits (US)

[i]…US [S]…DT0 [D]…DT10

DT0 $\boxed{0000000000010100}$ ⟶ DT10 $\boxed{0000000000011000}$

Example 2) Operation unit: 32 bits (UL)

[i]…UL [S]…DT0 [D]…DT10

DT0·DT1 $\boxed{00000000000000000000000000010000}$

↓

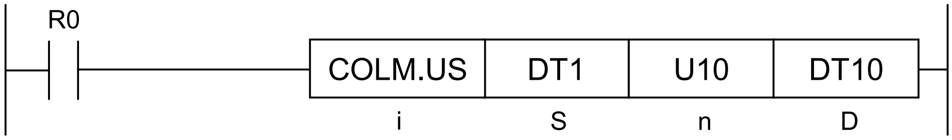
DT10·DT11 $\boxed{00000000000000000000000000011111}$

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

7.16 COLM (Conversion: Bit Line → Bit Column)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●					

■ List of operands

Operand	Description
S	The device address where the data for conversion are stored, or the constant
n	The device address where the specification for the bit position is stored, or the constant (available data range: 0 to 15)
D	Starting address of the device whose bit column is rewritten

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)		Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●				●
n	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●				●
D	●	●	●	●	●			●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

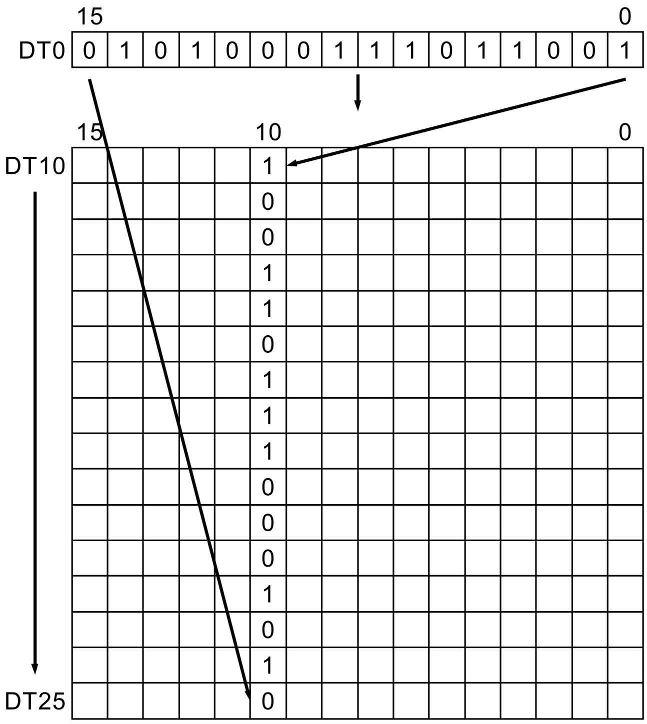
- 16 bit data rows specified by [S] are transferred to [n] bit data columns in the 16 word area specified by [D].
- Portions other than the specified bit column are not changed.

■ Processing

Example) Operation unit: 16 bits (US)

[i]...US

[S]...DT1 [n]...U10 [D]...DT10

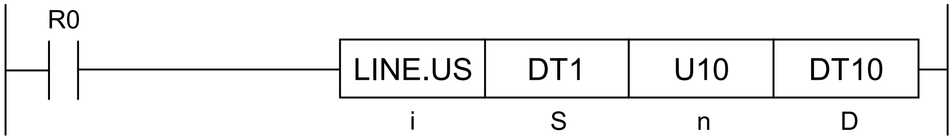


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the specification for the bit position [n] is not in the following range: 0 ≤ n ≤ 15.
	To be set if, when the conversion result is stored in the device address specified by [D], it exceeds the area.

7.17 LINE (Conversion: Bit Column → Bit Line)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●					

■ List of operands

Operand	Description
S	Starting address of the device whose bit column is read.
n	The device address where the specification for the bit position is stored, or the constant (available data range: 0 to 15)
D	Storage device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●										●
n	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

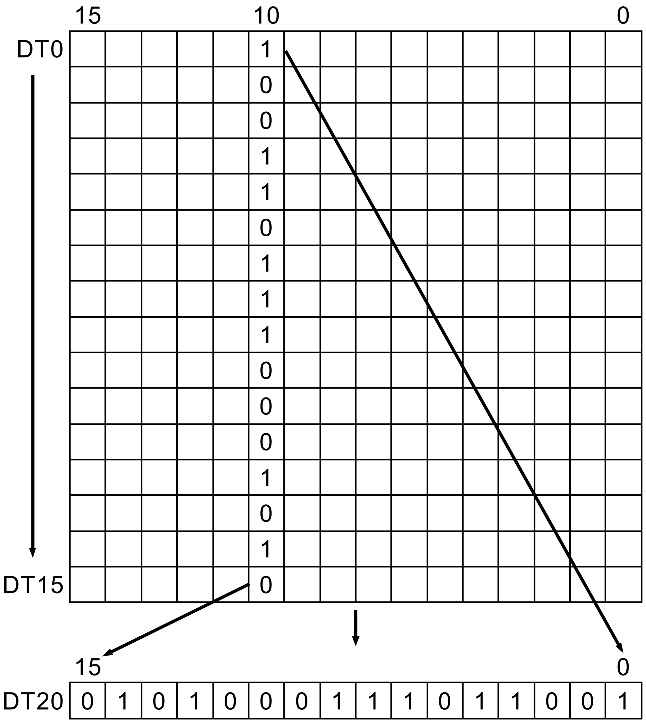
- The [n]-bit column data, in the 16-word device area specified by [S], are transferred to the 16-bit data specified by [D].

■ Processing

Example) Operation unit: 16 bits (US)

[i]...US

[S]...DT1 [n]...U10 [D]...DT20



■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the specification for the bit position [n] is not in the following range: 0 ≤ n ≤ 15.
	To be set when the conversion range specified by [S] exceeds the device address.

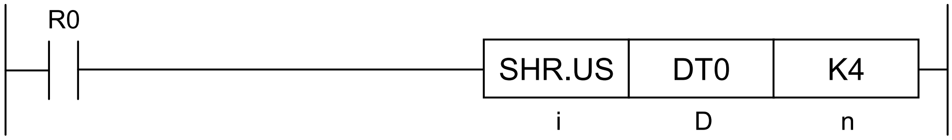
(MEMO)

8 High-level Instructions (Data Shift and Rotation)

Applicable Models: All Models

8.1 SHR (Right Shift for n Bits)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
D	The device address where the data to be shifted is stored
n	The device address where the number of shift bits is stored, or the constant

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)		Integer				Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●	●							●
n	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●				●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

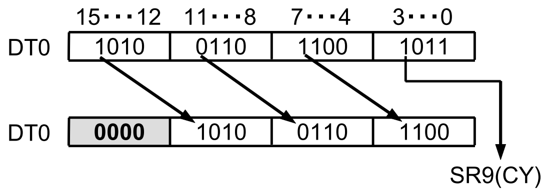
■ Outline of operation

- This instruction shifts the data specified by [D] to the right (to the low bit position), by the data amount specified by [n] (decimal).
- Once the data is shifted, [n] bits are filled with 0 from the uppermost/highest bit. The data from the lowest to the [n]th bit is stored in SR9 (CY).
- Only the lower 8 bits in data are available for [n]. The shift data amount should be specified between 0 and 255 bits.

■ Processing

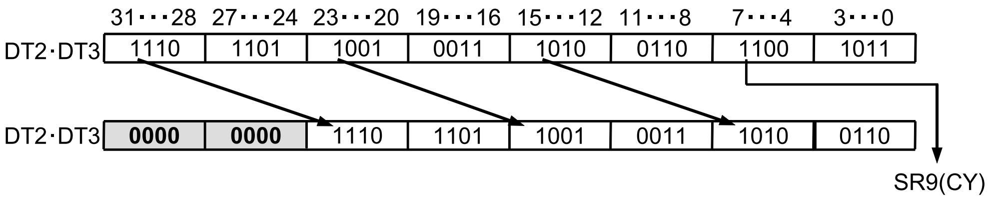
Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [D]…DT0 [n]…K4



Example 2) Operation unit: 32 bits (UL, SL)

[i]…UL,SL [D]…DT2 [n]…K8

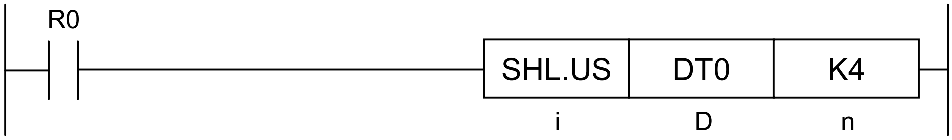


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
SR9 (CY)	To be reset if the [n] (no. of shift bits) is larger than the operation unit. In other cases, data in the [n]th bit from the least significant bit are to be set.

8.2 SHL (Left Shift for n Bits)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
D	The device address where the data to be shifted is stored
n	The device address where the number of shift bits is stored, or the constant

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●	●							●
n	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●				●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is an unsigned integer (US, UL).

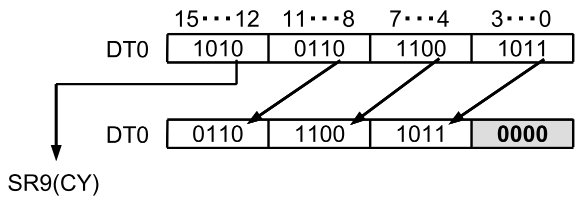
■ Outline of operation

- This instruction shifts the data specified by [D] to the left (to the high bit position), by the data amount specified by [n] (decimal).
- Once the data is shifted, [n] bits are filled with 0 from the least significant bit. The data from the highest to the [n]th bit is stored in SR9 (CY).
- Only the lower 8 bits in data are available for [n]. The shift data amount should be specified between 0 and 255 bits.

■ Processing

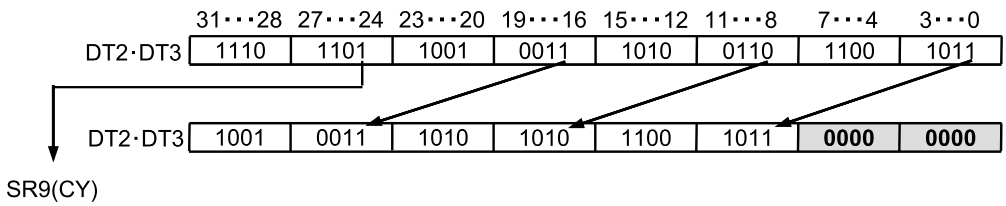
Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [D]…DT0 [n]…K4



Example 2) Operation unit: 32 bits (UL, SL)

[i]…UL,SL [D]…DT2 [n]…K8



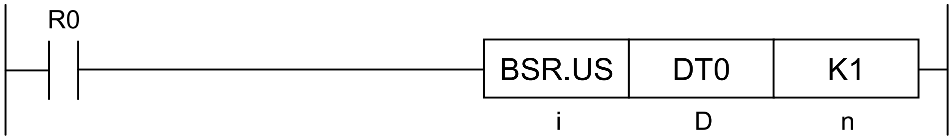
Isn't the shift destination for 11 ... 8 "0110"?

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
SR9 (CY)	To be reset if the [n] (no. of shift bits) is larger than the operation unit. In other cases, data in the [n]th bit from the most significant bit are to be set

8.3 BSR (Right Shift for n Digits)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
D	The device address where the data to be shifted is stored
n	The device address where the number of shift digits is stored, or the constant

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●	●							●
n	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●				●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

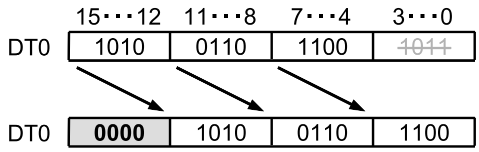
- This instruction shifts the data specified by [D] to the right (to the low bit position) for [n] digits (4 bits) (specified in decimal).
- Once the data is shifted, [n] digits are filled with 0 from the highest bit before shifting the data.
- Only the lower 8 bits in data are available for [n].

- If the operation unit is 16 bits (US, SS), the amount of shift is specified between 1 to 4 digits.
- If the operation unit is 32 bits (UL, SL), the amount of shift is specified between 1 to 8 digits.

■ Processing

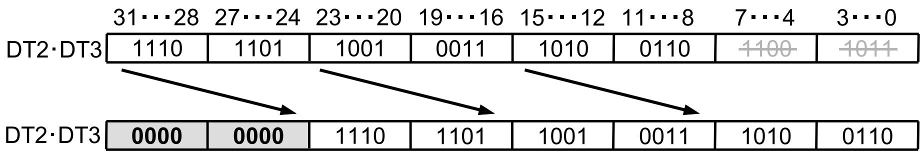
Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [D]…DT0 [n]…K1



Example 2) When the operation unit is 32-bit (UL, SL, SF)

[i]…UL,SL,SF [D]…DT2 [n]…K2



■ Precautions for programming

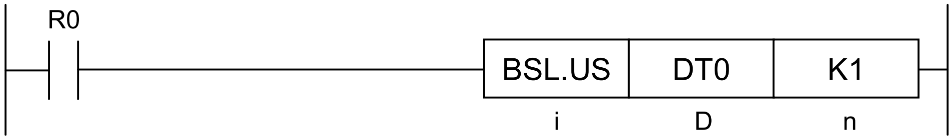
The digit data that have been shifted out are cleared.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

8.4 BSL (Left Shift for n Digits)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
D	The device address where the data to be shifted is stored
n	The device address where the number of shift digits is stored, or the constant

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●	●							●
n	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●				●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

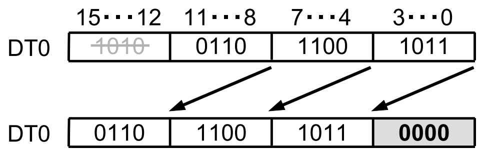
- This instruction shifts the data specified by [D] to the right (to the low bit position) for [n] digits (4 bits) (specified in decimal).
- Once the data is shifted, [n] digits are filled with 0 from the highest bit before shifting the data.
- Only the lower 8 bits in data are available for [n].

- If the operation unit is 16 bits (US, SS), the amount of shift is specified between 1 to 4 digits.
- If the operation unit is 32 bits (UL, SL), the amount of shift is specified between 1 to 8 digits.

■ Processing

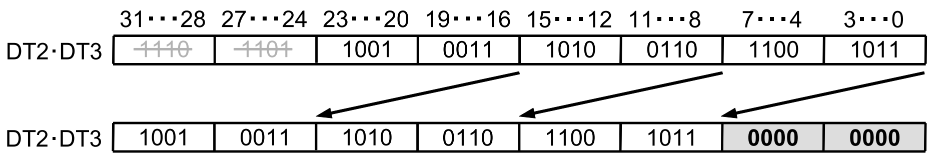
Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [D]…DT0 [n]…K1



Example 2) Operation unit: 32 bits (UL, SL)

[i]…US,SL [D]…DT2 [n]…K2



■ Precautions for programming

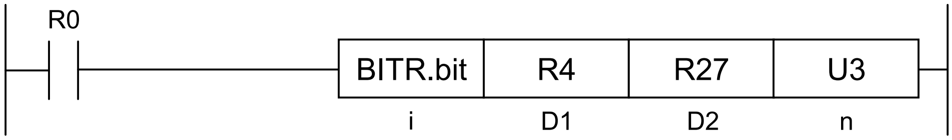
The digit data that have been shifted out are cleared.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

8.5 BITR (Right Shift of Multiple Devices for n Bits)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i	●						

■ List of operands

Operand	Description
D1	Starting address of the devices to be shifted (data format: according to the operation unit)
D2	End address of the devices to be shifted (data format: according to the operation unit)
n	The device address where the number of shift bits is stored, or the constant

■ Available bit devices (●: Available)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
D1	●	●	●	●							●	●	●	●
D2	●	●	●	●							●	●	●	●

■ Available word devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
n	●	●	●	●		●	●	●	●	●	●					●	●				●

(Note 1) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

- This instruction shifts the range from [D1] to [D2] to the right by [n] bits.
- The starting address of the bit is specified by [D1], and the end address by [D2].

- Once the data is shifted, the pre-shift lower [n] bits of [D1] vanish. The post-shift higher [n] bits of [D2] are padded with 0.
- The setting range of [n] is from 0 to 65535 (0 to 15 for CPU units older than Version 4.32). When [n] is 0, no shift takes place.

■ Processing

Example) When R4 to R27 is shifted three bits

[D1]…R4 [D2]…R27 [n]…U3

WR2				WR1				WR0			
15･･･12	11･･･8	7･･･4	3･･･0	15･･･12	11･･･8	7･･･4	3･･･0	15･･･12	11･･･8	7･･･4	3･･･0
0001	0010	0011	0100	0101	0110	0111	1000	1010	1010	1~~011~~	1100

0001	0010	0000	0110	1000	1010	1100	1111	0001	0101	0101	1100

Excluded Excluded 0 inserted for the number of shifted bits Excluded

■ Precautions for programming

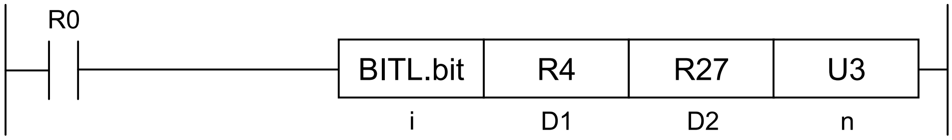
- In the case of a direct address and index modification address, specify the same device for [D1] and [D2]. At the same time, specify [D2] to be greater than or equal to [D1].

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [D1] is larger than [D2].
	To be set when [n] is greater than or equal to 16.

8.6 BITL (Left Shift of Multiple Devices for n Bits)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i	●						

■ List of operands

Operand	Description
D1	Starting address of the devices to be shifted (data format: according to the operation unit)
D2	End address of the devices to be shifted (data format: according to the operation unit)
n	The device address where the number of shift bits is stored, or the constant

■ Available bit devices (●: Available)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
D1	●	●	●	●							●	●	●	●
D2	●	●	●	●							●	●	●	●

■ Available word devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
n	●	●	●	●			●	●	●	●	●					●	●				●

(Note 1) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

- This instruction shifts the range from [D1] to [D2] to the left by [n] bits.
- The starting address of the bit is specified by [D1], and the end address by [D2].

- Once the data is shifted, the pre-shift higher [n] bits of [D1] vanish. The post-shift lower [n] bits of [D2] are padded with 0.
- The setting range of [n] is from 0 to 65535 (0 to 15 for CPU units older than Version 4.32). When [n] is 0, no shift takes place.

■ Processing

Example) When R4 to R27 is shifted three bits

[D1]…R4 [D2]…R27 [n]…U3

WR2				WR1				WR0			
15･･･12	11･･･8	7･･･4	3･･･0	15･･･12	11･･･8	7･･･4	3･･･0	15･･･12	11･･･8	7･･･4	3･･･0
0001	0010	~~001~~1	0100	0101	0110	0111	1000	1010	1010	1011	1100

0001	0010	1010	0010	1011	0011	1100	0101	0101	0101	1000	1100

Excluded Excluded 0 inserted for the number of shifted bits Excluded

■ Precautions for programming

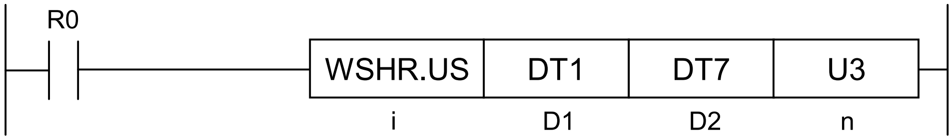
- In the case of a direct address and index modification address, specify the same device for [D1] and [D2]. At the same time, specify [D2] to be greater than or equal to [D1].

■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).
SR8	To be set when [D1] is larger than [D2].
(ER)	To be set when [n] is greater than or equal to 16.

8.7 WSHR (Right Shift of Block Area for n Words)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
D1	Starting address of the shift target
D2	End address of the shift target
n	Number of words to be shifted to the right (Available data range: 0 to 255 words)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K (Note 2)	U (Note 3)	H	SF	DF	" "	
D1	●	●	●	●			●	●	●		●										●
D2	●	●	●	●			●	●	●		●										●
n	●	●	●	●			●	●	●	●	●				●	●	●				●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 2) Can be specified only when the operation unit is signed integer (SS).

(Note 3) Can be specified only when the operation unit is unsigned integer (US).

■ Outline of operation

- This instruction shifts the area from the address specified by [D1] to the address specified by [D2] to the right by [n] words.
- The area from the starting address to the end address of the shift target is shifted to the right by the specified number of shift words.
- The specified number of shift words vanishes from the starting address. The specified number of shift words in the end address is padded with H0.

- If the specified number of shift words is larger than the shift target range, the entire shift target range is padded with H0000.

■ Processing

Example 1) Operation unit: 16 bits (US)

[i]…US

[D1]…DT1 [D2]…DT7 [n]…U3

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	**EEFF**	**CCDD**	**AABB**	**8899**	~~6677~~	~~4455~~	~~2233~~	0011	(HEX)

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	**0000**	**0000**	**0000**	**EEFF**	**CCDD**	**AABB**	**8899**	0011	(HEX)

↑ The shifted bits are padded with H 0000.

Example 2) Operation unit: 16 bits (SS)

[i]…SS

[D1]…DT1 [D2]…DT7 [n]…K2

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	**EEFF**	**CCDD**	**AABB**	**8899**	**6677**	~~4455~~	~~2233~~	0011	(HEX)

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	**0000**	**0000**	**EEFF**	**CCDD**	**AABB**	**8899**	**6677**	0011	(HEX)

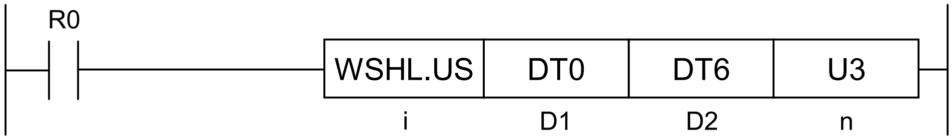
↑ The shifted bits are padded with H 0000.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [D1] is larger than [D2].
	To be set when [n] (specified number of shift words) is out of the available range.

8.8 WSHL (Left Shift of Block Area for n Words)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
D1	Starting address of the shift target
D2	End address of the shift target
n	Number of words to be shifted to the left (Available data range: 0 to 255 words)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K (Note 2)	U (Note 3)	H	SF	DF	" "	
D1	●	●	●	●			●	●	●		●										●
D2	●	●	●	●			●	●	●		●										●
n	●	●	●	●			●	●	●	●	●				●	●	●				●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 2) Can be specified only when the operation unit is signed integer (SS).

(Note 3) Can be specified only when the operation unit is unsigned integer (US).

■ Outline of operation

- This instruction shifts the area from the address specified by [D1] to the address specified by [D2] to the left by [n] words.
- The area from the starting address to the end address of the shift target is shifted to the left by the specified number of shift words.
- The specified number of shift words vanishes from the end address. The specified number of shift words in the starting address is padded with H0.

- If the specified number of shift words is larger than the shift target range, the entire shift target range is padded with H0000.

■ Processing

Example 1) Operation unit: 16 bits (US)

[i]…US

[D1]…DT0 [D2]…DT6 [n]…U3

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	EEFF	~~CCDD~~	~~AABB~~	~~8899~~	**6677**	**4455**	**2233**	**0011**	(HEX)

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	EEFF	**6677**	**4455**	**2233**	**0011**	**0000**	**0000**	**0000**	(HEX)

The shifted bits are padded with H 0000.

Example 2) Operation unit: 16 bits (SS)

[i]…SS

[D1]…DT1 [D2]…DT6 [n]…K2

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	EEFF	~~CCDD~~	~~AABB~~	**8899**	**6677**	**4455**	**2233**	0011	(HEX)

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	**EEFF**	**8899**	**6677**	**4455**	**2233**	**0000**	**0000**	0011	(HEX)

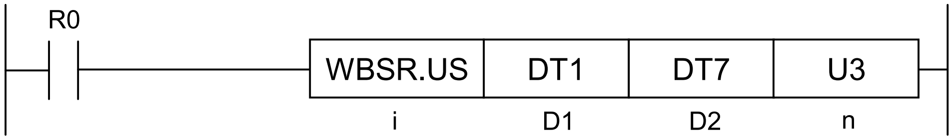
The shifted bits are padded with H 0000.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [D1] is larger than [D2].
	To be set when [n] (specified number of shift words) is out of the available range.

8.9 WBSR (Right Shift of Block Area for n Digits)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
D1	Starting address of the shift target
D2	End address of the shift target
n	Number of digits to be shifted to the right (Available data range: 0 to 255 digits)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K (Note 2)	U (Note 3)	H	SF	DF	" "	
D1	●	●	●	●			●	●	●		●										●
D2	●	●	●	●			●	●	●		●										●
n	●	●	●	●			●	●	●	●	●				●	●	●				●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 2) Can be specified only when the operation unit is signed integer (SS).

(Note 3) Can be specified only when the operation unit is unsigned integer (US).

■ Outline of operation

- This instruction shifts the area from the address specified by [D1] to the address specified by [D2] to the right by [n] digits.
- The area from the starting address to the end address of the shift target is shifted to the right by the specified number of shift digits.
- The specified number of shift digits vanishes from the starting address. The specified number of shift digits in the end address is padded with H0.

- If the specified number of shift digits is larger than the shift target range, the entire shift target range is padded with H0000.

■ Processing

Example 1) Operation unit: 16 bits (US)

[i]…US [D1]…DT1 [D2]…DT7 [n]…U3

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	**EEFF**	**CCDD**	**AABB**	**8899**	**6677**	**4455**	**2**~~233~~	0011	(HEX)

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	**000E**	**EFFC**	**CDDA**	**ABB8**	**8996**	**6774**	**4552**	0011	(HEX)

↑The shifted digits are padded with H0

Example 2) Operation unit: 16 bits (SS)

[i]…SS [D1]…DT1 [D2]…DT7 [n]…K5

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	**EEFF**	**CCDD**	**AABB**	**8899**	**6677**	**445**~~5~~	~~2233~~	0011	(HEX)

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	**0000**	**0EEF**	**FCCD**	**DAAB**	**B889**	**9667**	**7445**	0011	(HEX)

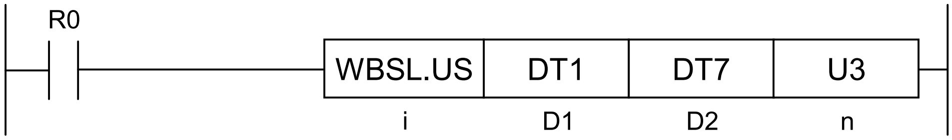
↑The shifted digits are padded with H0

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [D1] is larger than [D2].
	To be set when [n] (specified number of shift digits) is out of the available range.

8.10 WBSL (Left Shift of Block Area for n Digits)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
D1	Starting address of the shift target
D2	End address of the shift target
n	Number of digits to be shifted to the left (Available data range: 0 to 255 digits)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K (Note 2)	U (Note 3)	H	SF	DF	" "	
D1	●	●	●	●			●	●	●		●										●
D2	●	●	●	●			●	●	●		●										●
n	●	●	●	●			●	●	●	●	●				●	●	●				●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 2) Can be specified only when the operation unit is signed integer (SS).

(Note 3) Can be specified only when the operation unit is unsigned integer (US).

■ Outline of operation

- This instruction shifts the area from the address specified by [D1] to the address specified by [D2] to the left by [n] digits.
- The area from the starting address to the end address of the shift target is shifted to the left by the specified number of shift digits.
- The specified number of shift digits vanishes from the end address. The specified number of shift digits in the starting address is padded with H0.

- If the specified number of shift digits is larger than the shift target range, the entire shift target range is padded with H0000.

■ Processing

Example 1) Operation unit: 16 bits (US)

[i]…US [D1]…DT1 [D2]…DT7 [n]…U3

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	~~EEF~~**F**	**CCDD**	**AABB**	**8899**	**6677**	**4455**	**2233**	0011	(HEX)

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	**FCCD**	**DAAB**	**B889**	**9667**	**7445**	**5223**	**3000**	0011	(HEX)

↑The shifted digits are padded with H0

Example 2) Operation unit: 16 bits (SS)

[i]…SS [D1]…DT1 [D2]…DT7 [n]…K5

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	~~EEFF~~	~~C~~**CDD**	**AABB**	**8899**	**6677**	**4455**	**2233**	0011	(HEX)

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	**CDDA**	**ABB8**	**8996**	**6774**	**4552**	**2330**	**0000**	0011	(HEX)

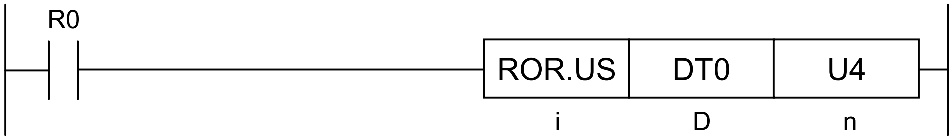
↑The shifted digits are padded with H0

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [D1] is larger than [D2].
	To be set when [n] (specified number of shift digits) is out of the available range.

8.11 ROR (Right Rotation of Data)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
D	The device address where the data to be rotated is stored
n	The device address where the number of rotation bits is stored, or the constant (Available data range: 0 to 255)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●	●							●
n	●	●	●	●			●	●	●	●	●	●	●	●		●	●				●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

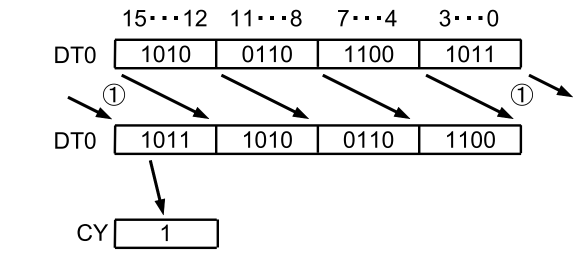
■ Outline of operation

- This instruction rotates the data specified by [D] to the right (to the low bit position), by the number of bits specified by [n] (decimal specification).
- Only the lower 8 bits in data are available for [n]. The rotation amount is specified between 0 and 255 bits.
- (Rotation amount - 1) bits are output to SR9 (CY).
- When the operation unit is 16 bits (US), if [n] is either 0 or a multiple of 16, the rotation amount is regarded as 0, and this instruction is not executed.
- When the operation unit is 32 bits (UL), if [n] is either 0 or a multiple of 32, the rotation amount is regarded as 0, and this instruction is not executed.

■ Processing

Example 1) Operation unit: 16 bits (US)

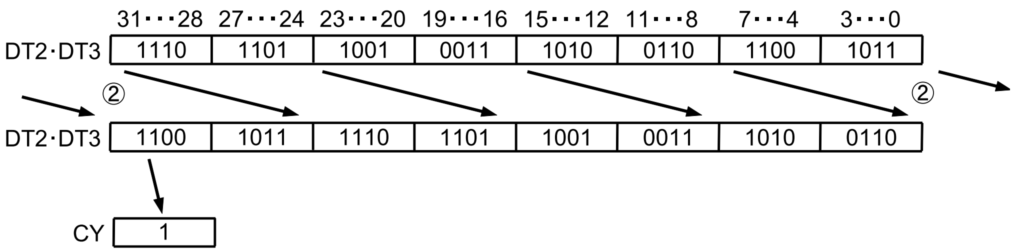
[i]…US [D]…DT0 [n]…U4



Output bit 3 of the pre-operation data to CY. (Output bit 15 of the post-operation data to CY)

Example 2) Operation unit: 32 bits (UL)

[i]…UL [D]…DT2 [n]…U8



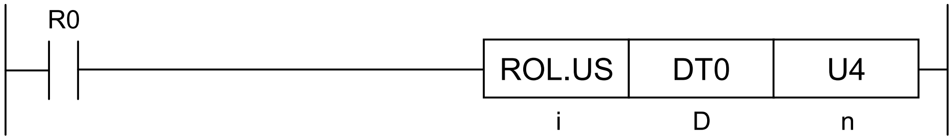
Output bit 7 of the pre-operation data to CY. (Output bit 31 of the post-operation data to CY)

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
SR9 (CY)	(Rotation amount - 1) bits of the pre-operation data are output. When the operation unit is 16 bits (US), if [n] is either 0 or a multiple of 16, the rotation amount is regarded as 0, and no change occurs. When the operation unit is 32 bits (UL), if [n] is either 0 or a multiple of 32, the rotation amount is regarded as 0, and no change occurs.

8.12 ROL (Left Rotation of Data)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
D	The device address where the data to be rotated is stored
n	The device address where the number of rotation bits is stored, or the constant (Available data range: 0 to 255)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●	●							●
n	●	●	●	●			●	●	●	●	●	●	●	●		●	●				●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

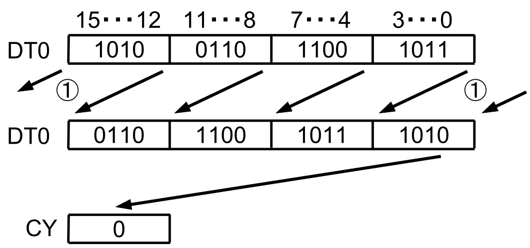
■ Outline of operation

- This instruction rotates the data specified by [D] to the left (to the high bit position), by the number of bits specified by [n] (decimal specification).
- Only the lower 8 bits in data are available for [n]. The rotation amount is specified between 0 and 255 bits.
- (Bit length of the operation unit - rotation amount) bits are output to SR9 (CY).
- When the operation unit is 16 bits (US), if [n] is either 0 or a multiple of 16, the rotation amount is regarded as 0, and this instruction is not executed.
- When the operation unit is 32 bits (UL), if [n] is either 0 or a multiple of 32, the rotation amount is regarded as 0, and this instruction is not executed.

■ Processing

Example 1) Operation unit: 16 bits (US)

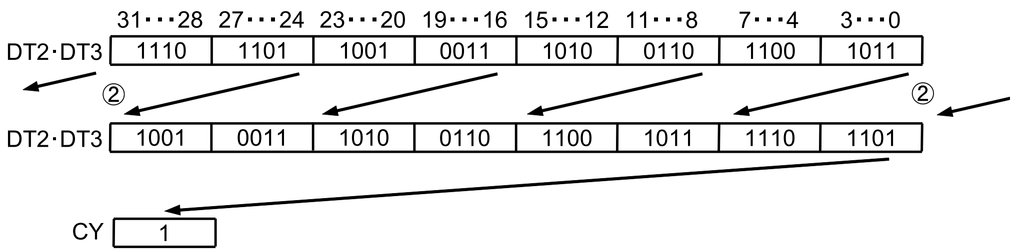
[i]…US [D]…DT0 [n]…U4



Output bit 12 of the pre-operation data to CY. (Output bit 0 of the post-operation data to CY)

Example 2) Operation unit: 32 bits (UL)

[i]…UL [D]…DT2 [n]…U8



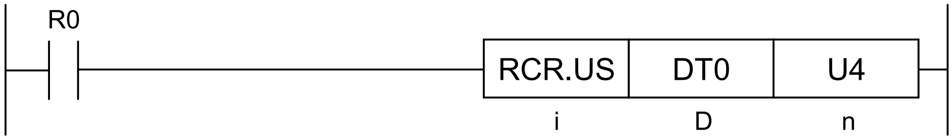
Output bit 24 of the pre-operation data to CY. (Output bit 0 of the post-operation data to CY)

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
SR9 (CY)	(Bit length of the operation unit - rotation amount) bits of the pre-operation data are output. When the operation unit is 16 bits (US), if [n] is either 0 or a multiple of 16, the rotation amount is regarded as 0, and no change occurs. When the operation unit is 32 bits (UL), if [n] is either 0 or a multiple of 32, the rotation amount is regarded as 0, and no change occurs.

8.13 RCR (Right Rotation of Data with Carry-Flag Data)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
D	The device address where the data to be rotated is stored
n	The device address where the number of rotation bits is stored, or the constant (Available data range: 0 to 255)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●	●							●
n	●	●	●	●			●	●	●	●	●	●	●	●		●	●				●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

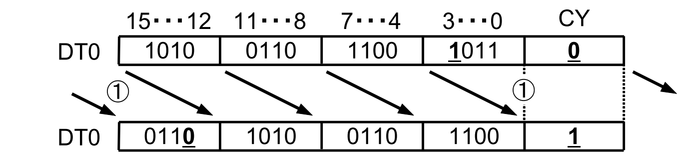
■ Outline of operation

- This instruction rotates the data specified by [D] to the right (to the low bit position), by the number of bits specified by [n] (decimal specification), with SR9 (CY).
- Only the lower 8 bits in data are available for [n]. The rotation amount is specified between 0 and 255 bits.
- (Rotation amount - 1) bits are output to SR9 (CY).
- When the operation unit is 16 bits (US), if [n] is either 0 or a multiple of 17, the rotation amount is regarded as 0, and this instruction is not executed.
- When the operation unit is 32 bits (UL), if [n] is either 0 or a multiple of 33, the rotation amount is regarded as 0, and this instruction is not executed.

■ Processing

Example 1) Operation unit: 16 bits (US)

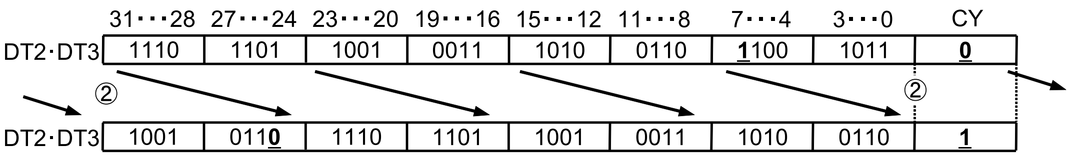
[i]…US [D]…DT0 [n]…U4 CY…0



Output bit 3 of the pre-operation data to CY. Output CY of the pre-operation data to bit 12

Example 2) Operation unit: 32 bits (UL)

[i]…UL [D]…DT2 [n]…U8 CY…0



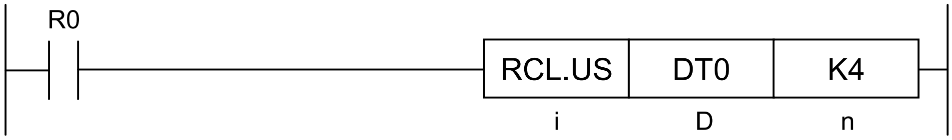
Output bit 7 of the pre-operation data to CY. Output CY of the pre-operation data to bit 24

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
SR9 (CY)	(Rotation amount -1) bits of the pre-operation data are output. When the operation unit is 16 bits (US), if [n] is either 0 or a multiple of 17, the rotation amount is regarded as 0, and no change occurs. When the operation unit is 32 bits (UL), if [n] is either 0 or a multiple of 33, the rotation amount is regarded as 0, and no change occurs.

8.14 RCL (Left Rotation of Data with Carry-Flag Data)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
D	The device address where the data to be rotated is stored
n	The device address where the number of rotation bits is stored, or the constant (Available data range: 0 to 255)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●	●							●
n	●	●	●	●			●	●	●	●	●	●	●	●		●	●				●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

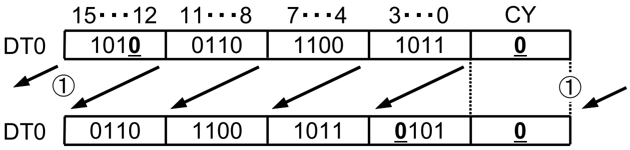
■ Outline of operation

- This instruction rotates the data specified by [D] to the left (to the high bit position), by the number of bits specified by [n] (decimal specification), with SR9 (CY).
- Only the lower 8 bits in data are available for [n]. The rotation amount is specified between 0 and 255 bits.
- (Bit length of the operation unit - rotation amount) bits are output to SR9 (CY).
- When the operation unit is 16 bits (US), if [n] is either 0 or a multiple of 17, the rotation amount is regarded as 0, and this instruction is not executed.
- When the operation unit is 32 bits (UL), if [n] is either 0 or a multiple of 33, the rotation amount is regarded as 0, and this instruction is not executed.

■ Processing

Example 1) Operation unit: 16 bits (US)

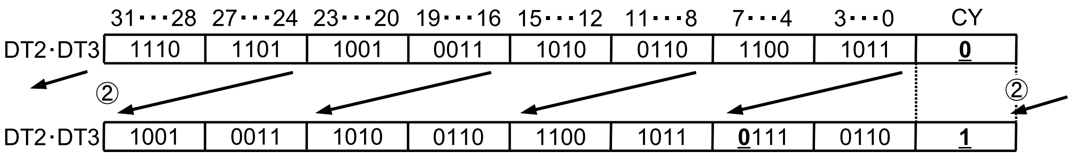
[i]…US [D]…DT0 [n]…K4 CY…0



Output bit 12 of the pre-operation data to CY. Output CY of the pre-operation data to bit 3

Example 2) Operation unit: 32 bits (UL)

[i]…UL [D]…DT2 [n]…K8 CY…0



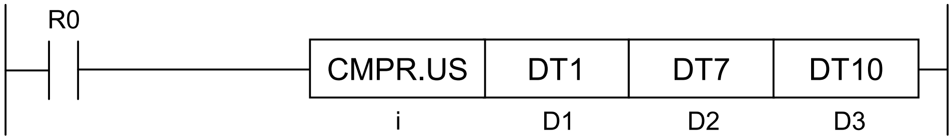
Output bit 24 of the pre-operation data to CY. Output CY of the pre-operation data to bit 7

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
SR9 (CY)	(Bit length of the operation unit - rotation amount) bits of the pre-operation data are output When the operation unit is 16 bits (US), if [n] is either 0 or a multiple of 17, the rotation amount is regarded as 0, and no change occurs. When the operation unit is 32 bits (UL), if [n] is either 0 or a multiple of 33, the rotation amount is regarded as 0, and no change occurs.

8.15 CMPR (Data Table Shift-Out and Compress)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
D1	Starting address of the buffer
D2	End address of the buffer
D3	Device address to store the read data

■ Available devices (●: Available)

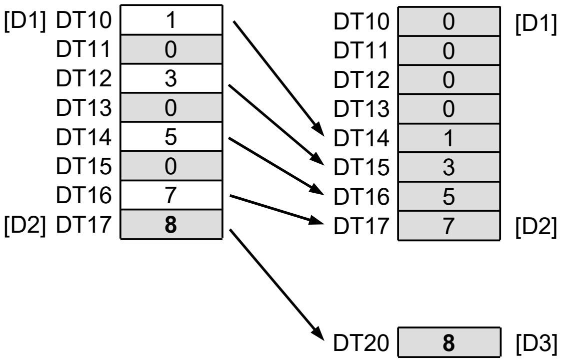
Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 1)	K	U	H	SF	DF	" "	
D1	●	●	●	●			●	●	●		●										●
D2	●	●	●	●			●	●	●		●										●
D3	●	●	●	●			●	●	●		●										●

(Note 1) Index register (I0 to IE)

■ Outline of operation

- According to the operation unit [i], the instruction transfers [D2] to [D3], and compresses the areas specified by [D1] to [D2].
  (Except the data transferred to [D3] at the time of compression)
- The data in the specified area, excluding 0, are allocated in descending order from the higher address of the specified area, and the remaining area is cleared to zero.

Example of data table shift-out and compress when DT10, DT17 and DT20 are respectively specified for [D1], [D2] and [D3].



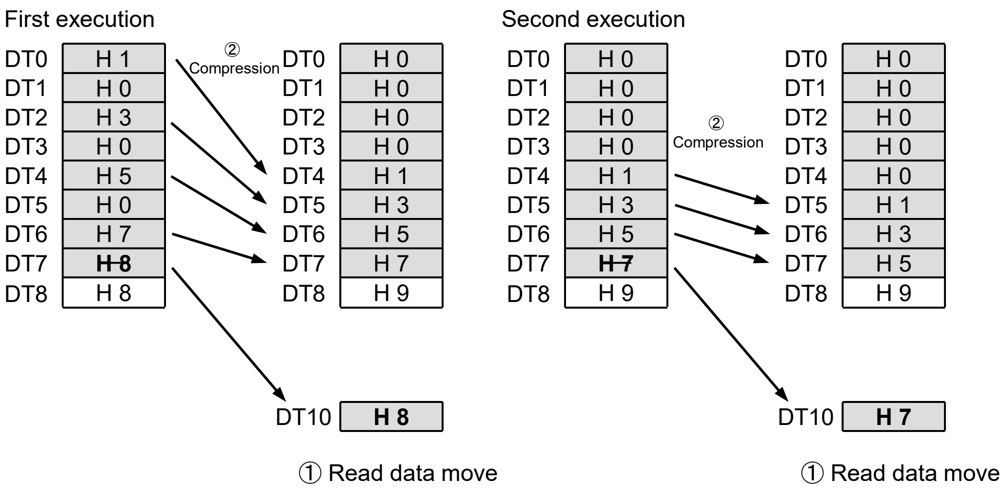
■ Processing

1. The buffer end is transferred to read data.
2. The data are compressed, excluding the data containing buffer end.

Example) Operation unit: 16 bits (US, SS) (executed twice)

[i]…US,SS

[D1]…DT1 [D2]…DT7 [D3]…DT10



■ Precautions during programming

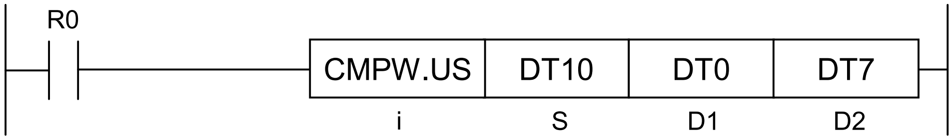
- In the case of a direct address and index modification address, specify the same device for [D1] and [D2]. At the same time, specify [D2] to be greater than or equal to [D1].

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when [D1] is larger than [D2].

8.16 CMPW (Data Table Shift-In and Compress)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S	Write data
D1	Starting address of the buffer
D2	End address of the buffer

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K (Note 3)	U (Note 4)	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●
D1	●	●	●	●			●	●													●
D2	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices and integer constants can be modified.

(Note 2) Index register (I0 to IE)

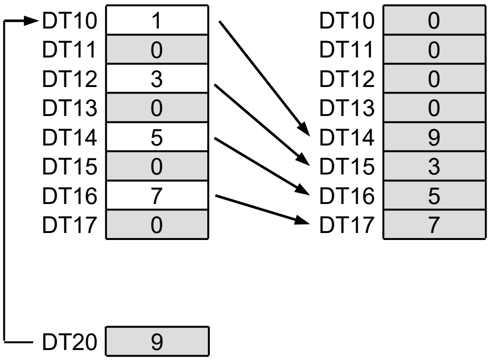
(Note 3) Can be specified only when the operation unit is signed integer (SS).

(Note 4) Can be specified only when the operation unit is an unsigned integer (US).

■ Outline of operation

- According to the operation unit of [n], the instruction transfers [S] to [D1], and compresses the areas specified by [D1] to [D2].
- The data in the specified area, excluding 0, are allocated in descending order from the higher address of the specified area, and the remaining area is cleared to zero.

Example of data table shift-out and compress when DT10, DT17 and DT20 are respectively specified for [D1], [D2] and [D3].



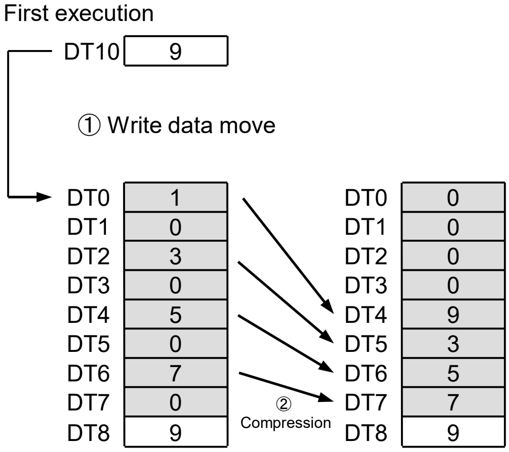
■ Processing

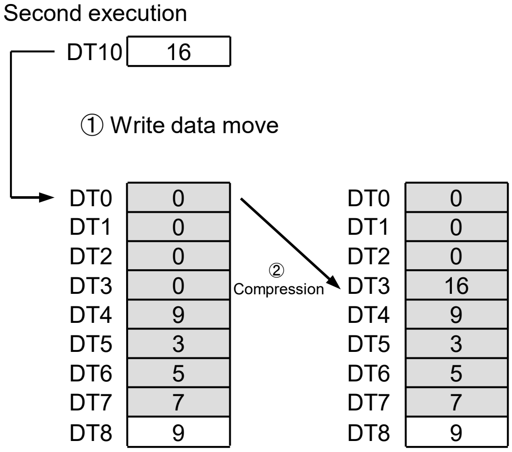
1. The write data are transferred to the buffer start. (The starting data are overwritten.)
2. The data are compressed in the range from buffer start to buffer end.

Example) Operation unit: 16 bits (US, SS) (executed twice)

[i]…US,SS

[S]…DT10 [D1]…DT0 [D2]…DT7





■ Precautions during programming

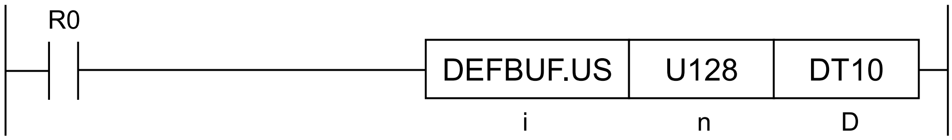
- In the case of a direct address and index modification address, specify the same device for [D1] and [D2]. At the same time, specify [D2] to be greater than or equal to [D1].

■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).
SR8 (ER)	To be set when [D1] is larger than [D2].

8.17 DEFBUF (Buffer Definition)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
n	The device address which specifies the buffer size, or the constant (available data range: 1 to 4096)
D	Starting device address of the data buffer

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K (Note 3)	U (Note 4)	H	SF	DF	" "	
n	●	●	●	●			●	●	●		●				●	●	●				●
D							●	●													●

(Note 1) Only 16-bit devices and integer constants can be modified.

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 4) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

- According to the operation unit [i], the instruction defines the data buffer to be for [n] data starting from the [D] area.
- From ([D]+1) (usable size) to ([D]+3) (write pointer) are initialized (cleared to zero).

■ Format of data buffer (FIFO buffer)

[D]	Buffer size	･･･Size of the data buffer area	Default: [n] (buffer size)
[D]+1	Stored data amount	･･･Stored data amount (by operation unit)	Default: H 0000
[D]+2	Reading pointer	･･･Relative number from [D]+4	Default: H 0000
[D]+3	Writing pointer	･･･Relative number from [D]+4	Default: H 0000
⋮		Data buffer area	

Data buffer area
*** The data buffer area is not cleared.**

■ Format of data buffer (LIFO buffer)

[S]	Buffer size	･･･Size of the data buffer area	Default: [n] (buffer size)
[S]+1	Stored data amount	･･･Stored data amount (by operation unit)	Default: H 0000
[S]+2	Fixed to 0	･･･Fixed to 0	Default: H 0000
[S]+3	LIFO pointer	･･･Relative number from [D]+4	Default: H 0000
⋮			

Data buffer area
*** The data buffer area is not cleared.**

■ Processing

1. [n] (buffer size) is specified in [D] (buffer start).
2. The range from [D]+1 (stored data amount) to [D]+3 (writing pointer) is cleared to zero.

Example) Operation unit: 16 bits (US, SS)

[i]…US,SS

[n]…K 128(U 128)　[D]…DT0

	Before		After	
DT0	**K 0**	→	**K 128**	(Buffer size)
DT1	**K 1**		**K 0**	(Stored data amount)
DT2	**K 2**		**K 0**	(Reading pointer)
DT3	**K 3**		**K 0**	(Writing pointer)
DT4	K 4		K 4	
DT5	K 5		K 5	
DT6	K 6		K 6	
DT7	K 7		K 7	
DT8	K 8		K 8	

■ Related instructions

- FIFR (Read data from the 16- or 32-bit data buffer (First-In-First-Out))
- BUFW (Write data in the 16- or 32-bit data buffer)
- LIFR (Read data from the 16- or 32-bit data buffer (Last-In-First-Out))

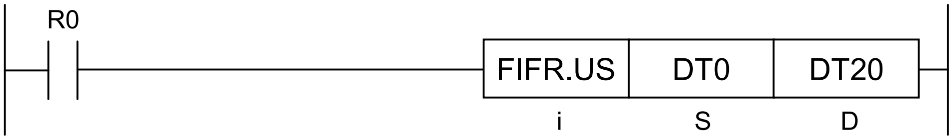
■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).
SR8 (ER)	To be set when [n] (buffer size) is out of the available range.

Name	Description
	To be set when the range [D] (buffer start) + [n] (buffer size) is out of the available range.

8.18 FIFR (Data Read (First-In-First-Out))

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S	Starting device address of the data buffer
D	Device address of the read data

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
S							●	●													●
D	●	●	●	●			●	●	●		●										●

(Note 1) Only 16-bit devices and integer constants can be modified.

(Note 2) Index register (I0 to IE)

■ Outline of operation

- This instruction reads data from FIFO buffer specified by [S], and sets it to [D].
  (In the [S] buffer area, it is necessary to define buffer first using the DEFBUF instruction.)
- Pre-execution buffer consistency check (An operation error occurs in the following cases)
  1. [S] (buffer size) > 4096, or [S] (buffer size) = 0
  2. [S]+1 (stored data amount) = 0
  3. [S]+1 (stored data amount) > [S] (buffer size)
  4. [S]+2 (read pointer) > [S] (buffer size)
  5. Buffer area exceeds the upper limit of the specified device

- According to the operation unit [i], the data of the area specified by "[S]+2" (read pointer) are set to [D].
- "[S]+2" (read pointer) is incremented (+1).
- After incrementing (+1), if "[S]+2" (read pointer) is [S] (buffer size), 0 is set to "[S]+2" (read pointer).
- "[S]+1" (stored data amount) is decremented (-1).

■ Format of data buffer (FIFO buffer)

[S]	Buffer size	･･･Size of the data buffer area
[S]+1	Stored data amount	･･･Stored data amount (by operation unit)
[S]+2	Reading pointer	･･･Relative number from [S]+4
[S]+3	Writing pointer	･･･Relative number from [S]+4
:		Data buffer area

■ Processing

1. Set the area specified by ([S]+2) (read pointer) to [D] (read data).
2. ([S]+2) (read pointer) is incremented (+1).
3. ([S]+1) (stored data amount) is decremented (-1).

Example) Operation unit: 16 bits (US, SS)
[i]…US,SS
[S]…DT0 [D]…DT20

Before:

[S]	DT0	K 5	(Buffer size)
[S]+1	DT1	**K 4**	(Stored data amount)
[S]+2	DT2	**K 0**	(Reading pointer)
[S]+3	DT3	K 4	(Writing pointer)
[S]+4	DT4	H 101	(0th)
[S]+5	DT5	H 102	(1st)
[S]+6	DT6	H 103	(2nd)
[S]+7	DT7	H 104	(3rd)
[S]+8	DT8	H 105	(4th)

DT20 H 20

After:

[S]	DT0	K 5	(Buffer size)	
[S]+1	DT1	**K 3**	(Stored data amount)	③ Decrement (-1)
[S]+2	DT2	**K 1**	(Reading pointer)	② Increment (+1)
[S]+3	DT3	K 4	(Writing pointer)	
[S]+4	DT4	H 101	(0th)	
[S]+5	DT5	H 102	(1st)	
[S]+6	DT6	H 103	(2nd)	
[S]+7	DT7	H 104	(3rd)	
[S]+8	DT8	H 105	(4th)	

DT20 **H 101**

① Because [S]+2 (reading pointer) points at 0, transfer 0th data in the buffer to D.

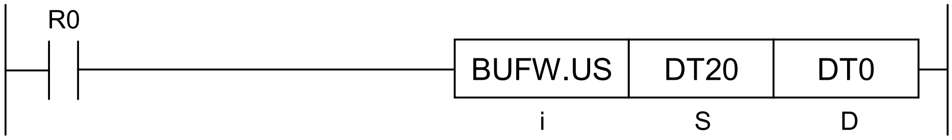
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [S] (buffer size) is larger than 4096, or [S] (buffer size) is 0.
	To be set when "[S]+1" (stored data amount) is 0.

Name	Description
	To be set when "[S]+1" (stored data amount) is larger than [S] (buffer size).
	To be set when "[S] +2" (read pointer) is greater than or equal to [S] (buffer size).
	To be set when the buffer area exceeds the upper limit of a specified device.

8.19 BUFW (Data Write)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S	The device address of the write data, or the constant
D	Starting device address of the data buffer

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K (Note 3)	U (Note 4)	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●
D							●	●													●

(Note 1) Only 16-bit devices and integer constants can be modified.

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 4) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

- This instruction sets the data specified by [S] to the buffer specified by [D].
  (In the [D] buffer area, it is necessary to define buffer first using the DEFBUF instruction.)
- Pre-execution buffer consistency check (An operation error occurs in the following cases)
  1. [D] (buffer size) > 4096, or [D] (buffer size) = 0
  2. [D]+1 (stored data amount) ≥ [D] (buffer size)
  3. [D]+3 (write pointer) ≥ [D] (buffer size)
  4. Buffer area exceeds the upper limit of the specified device

- According to the operation unit [i], [S] is set to the area specified by "[D]+3" (write pointer).
- "[D]+3" (write pointer) is incremented (+1).
- After incrementing, if "[D]+3 (write pointer) is equal to [S] (buffer size)", [D]+3 (write pointer) is set to 0.
- "[S]+1" (stored data amount) is incremented (+1).

■ Format of data buffer (FIFO)

[D]	Buffer size	･･･Size of the data buffer area
[D]+1	Stored data amount	･･･Stored data amount (by operation unit)
[D]+2	Reading pointer	･･･Relative number from [D]+4
[D]+3	Writing pointer	･･･Relative number from [D]+4
⋮		Data buffer area
⋮		
⋮		
⋮		

■ Processing

1. Set [S] (write data) to the area specified by "[D]+3" (write pointer).
2. "[D]+3" (write pointer) is incremented (+1).
3. Increment (+1) "[D]+1" (stored data amount).

Example) 16 bits (US, SS)
[i]…US,SS
[S]…DT20 [D]…DT0

Before:

[D]	DT0	5	(Buffer size)
[D]+1	DT1	**3**	(Stored data amount)
[D]+2	DT2	**0**	(Reading pointer)
[D]+3	DT3	3	(Writing pointer)
[D]+4	DT4	101	(0th)
[D]+5	DT5	102	(1st)
[D]+6	DT6	103	(2nd)
[D]+7	DT7	104	(3rd)
[D]+8	DT8	105	(4th)
	DT20	201	

After:

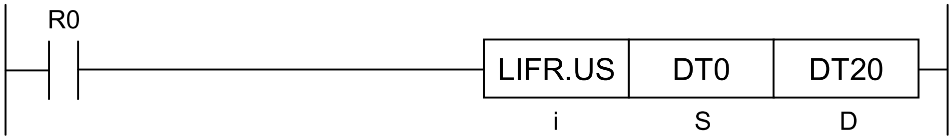
[D]	DT0	5	(Buffer size)	
[D]+1	DT1	**4**	(Stored data amount)	③ Increment (+1)
[D]+2	DT2	0	(Reading pointer)	
[D]+3	DT3	**4**	(Writing pointer)	② Increment (+1)
[D]+4	DT4	101	(0th)	
[D]+5	DT5	102	(1st)	
[D]+6	DT6	103	(2nd)	
[D]+7	DT7	**201**	(3rd)	
[D]+8	DT8	105	(4th)	① Because [D]+3 (writing pointer) points at 3, [S] is transferred to the 3rd area of the buffer.
	DT20	**201**		

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [D] (buffer size) is larger than 4096, or [D] (buffer size) is 0.
	To be set when [D] +1 (stored data amount) is greater than or equal to [D] (buffer size).
	To be set when [D] +3 (write pointer) is greater than or equal to [D] (buffer size).
	To be set when the buffer area exceeds the upper limit of a specified device.

8.20 LIFR (Data Read (Last-In-First-Out))

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S	Starting device address of the data buffer
D	Device address of the read data

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
S							●	●													●
D	●	●	●	●			●	●	●		●										●

(Note 1) Only 16-bit devices and integer constants can be modified.

(Note 2) Index register (I0 to IE)

■ Outline of operation

- This instruction reads data from LIFO buffer specified by [S], and sets it to [D].
  (In the [S] buffer area, it is necessary to define buffer first using the DEFBUF instruction.)
- Pre-execution buffer consistency check (An operation error occurs in the following cases)
  1. [S] (buffer size) > 4096, or [S] (buffer size) = 0
  2. [S]+1 (stored data amount) = 0
  3. [S]+2 ≠ 0
  4. [S]+1 (stored data amount) > [S] (buffer size)
  5. [S]+3 (LIFO pointer) ≥ [S] (buffer size)
  6. Buffer area exceeds the upper limit of the specified device

- If "[S]+3" (LIFO pointer) is 0, set [S] (buffer size) to "[S]+3" (LIFO pointer).
- "[S]+3" (LIFO pointer) is decremented (-1).
- According to the operation unit [i], the data of the area specified by "[S]+3" (LIFO pointer) are set to [D].
- "[S]+1" (stored data amount) is decremented (-1).

■ Format of data buffer (LIFO)

[S]	Buffer size	···Size of the data buffer area
[S]+1	Stored data amount	···Stored data amount (by operation unit)
[S]+2	Fixed to 0	···Fixed to 0
[S]+3	LIFO pointer	···Relative number from [S]+4
⋮		Data buffer area

■ Processing

1. "[S]+3" (LIFO pointer) is decremented (-1).
2. Set the data of the area specified by "[S]+3" (LIFO pointer) to [D] (read data).
3. "[S]+1" (stored data amount) is decremented (-1).

Example) 16 bits (US, SS)
[i]…US,SS
[S]…DT0 [D]…DT20

[S]	DT0	K 5	(Buffer size)
[S]+1	DT1	**K 4**	(Stored data amount)
[S]+2	DT2	K 0	-
[S]+3	DT3	**K 4**	(LIFO pointer)
[S]+4	DT4	H 101	(0th)
[S]+5	DT5	H 102	(1st)
[S]+6	DT6	H 103	(2nd)
[S]+7	DT7	**H 104**	(3rd)
[S]+8	DT8	H 105	(4th)

DT20 H 10

➡

[S]	DT0	K 5	(Buffer size)	
[S]+1	DT1	**K 3**	(Stored data amount)	③ Decrement (-1)
[S]+2	DT2	K 0	-	
[S]+3	DT3	**K 3**	(LIFO pointer)	① Decrement (-1)
[S]+4	DT4	H 101	(0th)	
[S]+5	DT5	H 102	(1st)	
[S]+6	DT6	H 103	(2nd)	
[S]+7	DT7	H 104	(3rd)	
[S]+8	DT8	H 105	(4th)	

DT20 **H 104**

② Because [S]+3 (LIFO pointer) points at 3, transfer 3rd data in the buffer to [D].

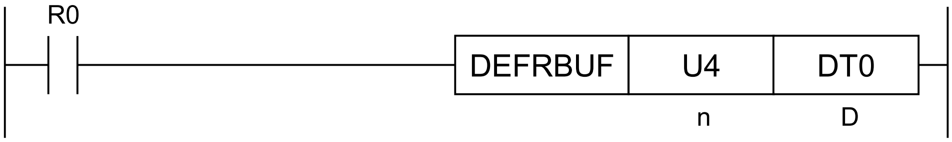
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [S] (buffer size) is larger than 4096, or [S] (buffer size) is 0.
	To be set when "[S]+1" (stored data amount) is 0.
	To be set when "[S]+2" is other than 0.
	To be set when "[S]+1" (stored data amount) is larger than [S] (buffer size).

Name	Description
	To be set when [S] +3 (LIFO pointer) is greater than or equal to [S] (buffer size).
	To be set when the buffer area exceeds the upper limit of a specified device.

8.21 DEFRBUF (Ring Buffer Definition)

■ Ladder diagram



■ List of operands

Operand	Description
n	Device address storing the buffer size or the constant (available range: 1 to 30000)
D	Starting device address of a ring buffer

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
n	●	●	●	●			●	●	●	●	●					●	●				●
D							●	●													●

■ Outline of operation

- This instruction defines a ring buffer that has storage areas for a total value and a moving average value, and creates a ring buffer for [n] data in the area that starts with [D].
- 16-bit integer values can be stored in the buffer.
- Use the RBUFW instruction to write data into a ring buffer that is defined by this instruction.
- When the number of stored data reaches the buffer size, the next data is written from the beginning of the ring buffer and the previous values are overwritten.

■ Structure of ring buffer

	Name	Data type	Description
[D]	Buffer size	Unsigned 16-bit integer	The size of the ring buffer area is stored when the DEFRBUF instruction is executed.
[D+1]	Number of stored data	Unsigned 16-bit integer	The amount of data that is stored in the buffer data area is stored. The value is reset to 0 when the DEFRBUF instruction is executed.
[D+2]	Total value	Signed 32-bit integer	The total value of the stored data is stored. The value is reset to 0 when the DEFRBUF instruction is executed.
[D+3]			
[D+4]	Moving average value	Single-precision floating point real number (32-bit)	The moving average value of the stored data is stored as a single-precision floating point real number. The value is reset to 0 when the DEFRBUF instruction is executed.
[D+5]			

	Name	Data type	Description
[D+6]	Write pointer	Unsigned 16-bit integer	The relative number from [D+7] is stored. The value is reset to 0 when the DEFRBUF instruction is executed. The value is incremented when data is written by the RBUFW instruction. The value returns to 0 when the RBUFW instruction is executed at the end of the data area.
[D+7]	Buffer data area	Unsigned 16-bit integer Signed 16-bit integer	Data is written by the RBUFW instruction.

[D +7+n-1]			

■ Processing

1. [n] (buffer size) is specified in [D] (buffer start).
2. The range from [D+1] (stored data amount) to [D+6] (write pointer) is cleared to zero.

Example) n=U4, D=DT0

Before		→	After		
DT0	**K 0**		DT0	**U 4**	Buffer size
DT1	**K 1**		DT1	**U 0**	Stored data amount
DT2	**K 2**		DT2	**K 0**	Total
DT3	**K 3**		DT3		
DT4	**K 4**		DT4	**SF 0.0**	Average
DT5	**K 5**		DT5		
DT6	**K 6**		DT6	**U 0**	Write pointer
DT7	K 7		DT7	K 5	
DT8	K 8		DT8	K 6	
DT9	K 9		DT9	K 7	
DT10	K 10		DT10	K 8	

■ Precautions for programming

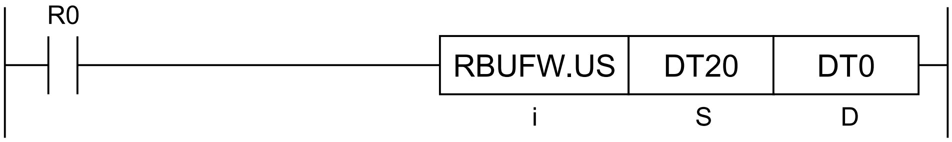
- When this instruction is executed, data in the ring buffer area is not cleared. Use a data transfer instruction or other instructions to reset the area, if necessary.
- Do not use other instructions than the RBUFW instruction to write data into the ring buffer. The calculation of total value and moving average value are not guaranteed.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [n] (buffer size) is out of the available range.
	To be set when the range of [D (the beginning of a buffer) + n (buffer size)] is out of the accessible range.

8.22 RBUFW (Write to Ring Buffer, Calculation of Total Value and Moving Average Value)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S	The device address storing written data, or the constant
D	Starting device address of a ring buffer

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K (Note 1)	U (Note 2)	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●
D							●	●													●

(Note 1) Can be specified only when the operation unit is signed integer (SS).

(Note 2) Can be specified only when the operation unit is unsigned integer (US).

■ Outline of operation

- This instruction writes data to the buffer data area of the ring buffer that is defined by the DEFRBUF instruction, and calculates the total value and the moving average value.
- This instruction writes the data specified by [S] to the buffer data area of the ring buffer that starts from [D].
- This instruction stores the total value of the data in [D+2, D+3] and the moving average value in [D+4, D+5].
- [D+1] (stored data amount) and [D+6] (write pointer) are incremented (+1).
- After incrementing, if "[D+6] (write pointer) is equal to [D] (buffer size)", [D+6] (write pointer) is set to 0. When this instruction is executed the next time, the data is overwritten from the

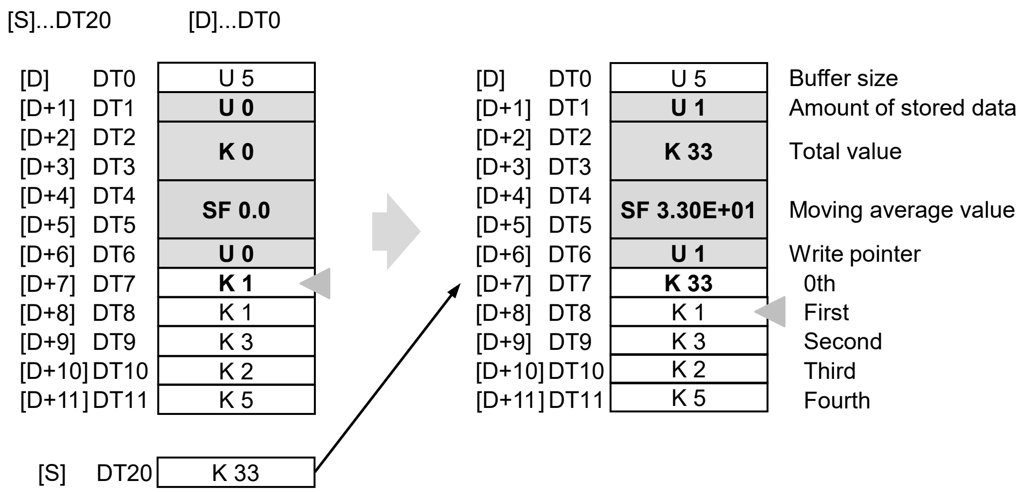
beginning of the ring buffer area. However, the area [D+1] for the stored data amount is not changed.

■ Precautions for programming

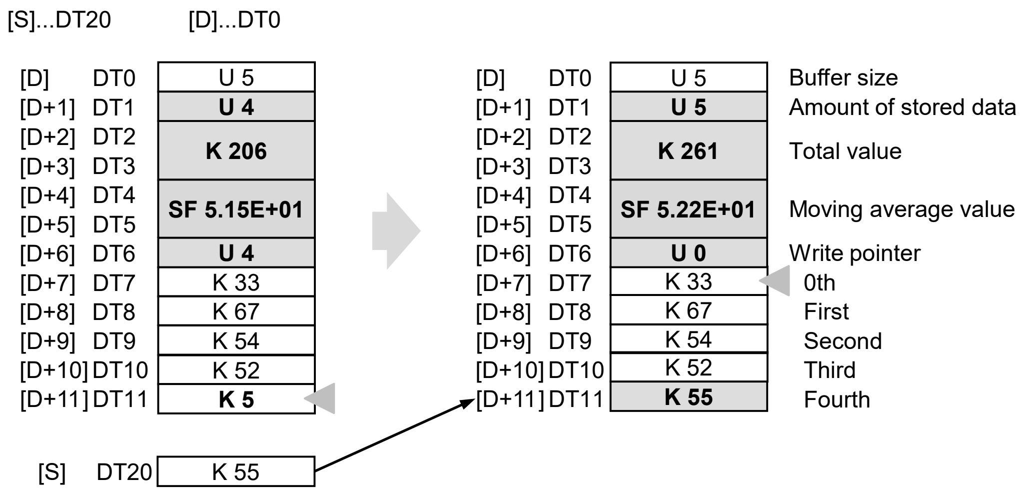
- According to the operation unit [i], set the value of [S] that is written to the buffer data area.

■ Processing

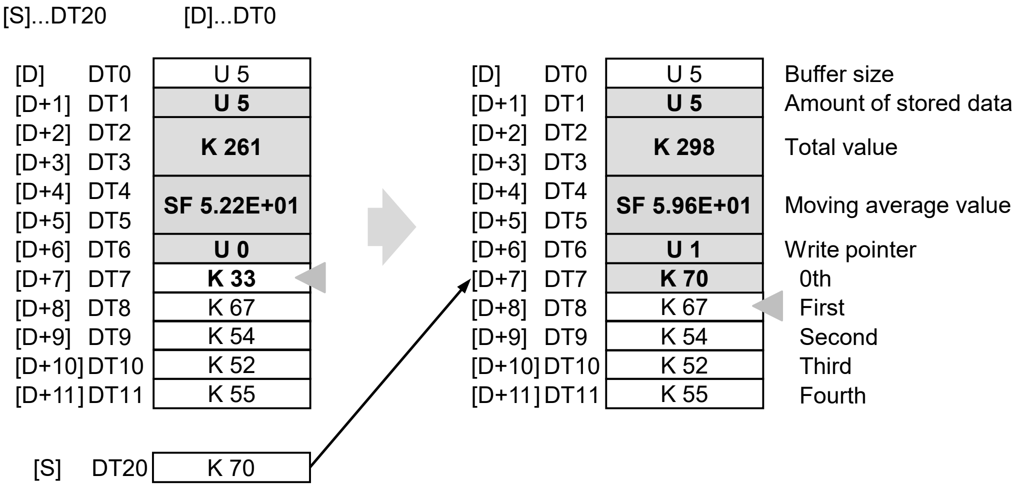
Example 1) When data is written once by the RBUFW instruction with the buffer size of 5



Example 2) When data is written five times by the RBUFW instruction with the buffer size of 5



Example 3) When data is written six times by the RBUFW instruction with the buffer size of 5



■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [D] (buffer size) is larger than 30000, or [D] (buffer size) is 0.
	To be set when [D+1] (stored data amount) is larger than [D] (buffer size).
	To be set when [D+6] (write pointer) is greater than or equal to [D] (buffer size).
	To be set when the buffer area exceeds the upper limit of a specified device.

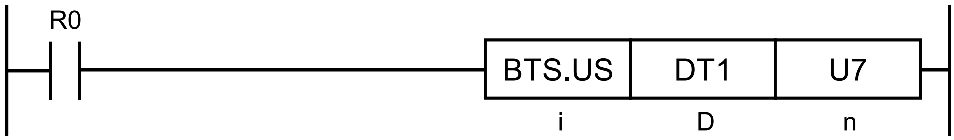
(MEMO)

9 High-level Instructions (Bit Manipulation)

Applicable Models: All Models

9.1 BTS (16-bit Data Specified Bit Set)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●					

■ List of operands

Operand	Description
D	Device address of target data
n	Bit number (device address or constant) (available range: 0 to 15)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●										●
n	●	●	●	●			●	●	●	●	●					●	●				●

(Note 1) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

- This instruction turns ON (1) the [n]th bit in the area specified by [D].
- Other bits except the bit specified by [n] do not change.
- [n] is in the range of U0 to U15.

■ Processing

The [n]th bit of the target data is set.

Example 1) Specifying a constant for the bit number

[D]...DT0 [n]...U7

Bits	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
DT0	0	1	0	1	1	0	0	0	**0**	1	1	1	0	0	0	0

↓

Bits	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
DT0	0	1	0	1	1	0	0	0	**1**	1	1	1	0	0	0	0

Example 2) Specifying a device for the bit number

[D]...DT1 [n]...DT0

bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
DT0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	U8
DT1	0	1	0	1	1	0	0	**0**	0	1	1	1	0	0	0	0	

↓

bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
DT1	0	1	0	1	1	0	0	**1**	0	1	1	1	0	0	0	0

■ Precautions for programming

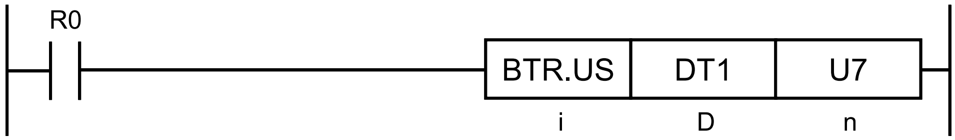
The conventional models (such as FP2 or FP2SH) operate with only the lower four bits as valid even when the specified operand [n] is out of the available range. For FP7, an operation error occurs when the specified value is out of the range.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [n] is out of the range.

9.2 BTR (16-bit Data Specified Bit Reset)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●					

■ List of operands

Operand	Description
D	Device address of target data
n	Bit number (device address or constant) (available range: 0 to 15)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●										●
n	●	●	●	●			●	●	●	●	●					●	●				●

(Note 1) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

- This instruction turns OFF (0) the [n]th bit in the area specified by [D].
- Other bits except the bit specified by [n] do not change.
- [n] is in the range of U0 to U15.

■ Processing

Example 1) Specifying a constant for the bit number

[D]...DT0 [n]...U4

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
DT0	0	1	0	1	1	0	0	0	0	1	1	**1**	0	0	0	0

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
DT0	0	1	0	1	1	0	0	0	0	1	1	**0**	0	0	0	0

Example 2) Specifying a device for the bit number

[D]...DT1 [n]...DT0

Bits	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
DT0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	U5
DT1	0	1	0	1	1	0	0	0	0	1	**1**	1	0	0	0	0	

Bits	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
DT1	0	1	0	1	1	0	0	0	0	1	**0**	1	0	0	0	0

■ Precautions for programming

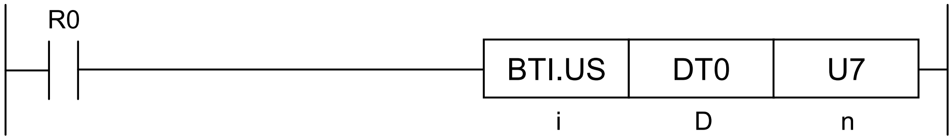
The conventional models (such as FP2 or FP2SH) operate with only the lower four bits as valid even when the specified operand [n] is out of the available range. For FP7, an operation error occurs when the specified value is out of the range.

■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).
SR8 (ER)	To be set when [n] is out of the range.

9.3 BTI (Bit Inversion)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●					

■ List of operands

Operand	Description
D	Inversion target data (device address)
n	Bit number (device address or constant) (available data range: 0 to 15)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
S							●	●													●
D	●	●	●	●			●	●	●		●										●

(Note 1) Only 16-bit devices and integer constants can be modified.

(Note 2) Index register (I0 to IE)

■ Outline of operation

- This instruction inverts the [n]th bit in the area specified by [D] according to the operation unit of [i].

■ Processing

Example) Operation unit: 16 bits (US)
[i]…US
[D]…DT0 [n]…U7 <Invert Bit 7>

	DT0															
Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	0	1	0	1	1	0	0	0	**0**	1	1	1	0	0	0	0

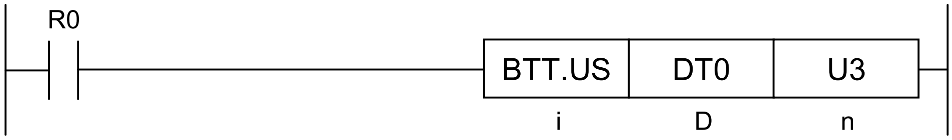
	DT0															
Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	0	1	0	1	1	0	0	0	**1**	1	1	1	0	0	0	0

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when [n] is out of the range.

9.4 BTT (Bit Test)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●					

■ List of operands

Operand	Description
D	Test target data (device address)
n	Bit number (device address or constant) (available data range: 0 to 15)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●										●
n	●	●	●	●			●	●	●	●	●					●	●				●

■ Outline of operation

This instruction tests the [n]th bit in the area specified by [D] (ON/OFF judgment) according to the operation unit [i], and outputs the result to SRB(=).

State of specified bit	SRB (= flag)
ON (1)	OFF (0)
OFF (0)	ON (1)

■ Processing

Example 1）Operation unit: 16 bits (US) (SRB is OFF)

[i]…US [D]…DT0 [n]…U3

	DT0															
Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	0	1	0	1	1	0	0	0	0	1	1	1	**1**	0	0	0

Flag operation during execution

State of the specified bit	SRB(=)
ON(1)	OFF(0)

Example 2) Operation unit: 16 bits (US) (SRB is ON)

[i]…US [D]…DT0 [n]…U3

	DT0															
Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN	0	1	0	1	1	0	0	0	0	1	1	1	**0**	0	0	0

Flag operation during execution

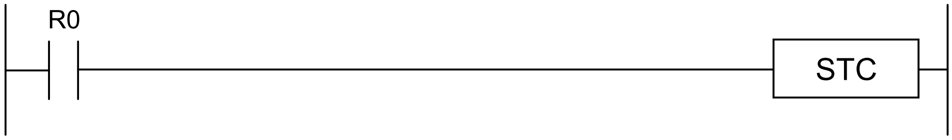
State of the specified bit	SRB(=)
OFF(0)	ON(1)

■ Flag operations

Name	Description
SR7, SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [n] is out of the range.
SRB (=)	To be set when the test bit (Bit [n]) is '0'.
	To be reset when the test bit (Bit [n]) is '1'.

9.5 STC (Carry-Flag Set)

■ Ladder diagram



■ Outline of operation

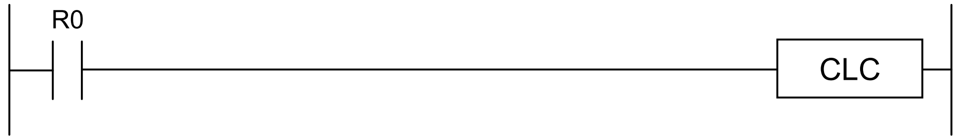
This instruction turns ON SR9 (CY).

■ Flag operations

Name	Description
SR9 (CY)	To be set after this instruction is executed.

9.6 CLC (Carry-Flag Reset)

■ Ladder diagram



■ Outline of operation

This instruction turns OFF SR9 (CY).

■ Flag operations

Name	Description
SR9 (CY)	To be reset after this instruction is executed.

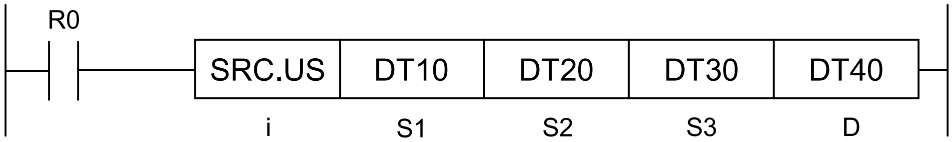
(MEMO)

10 High-Level Instruction (Data Processing Control)

Applicable Models: All Models

10.1 SRC (Data Search)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Device address where the search data is stored, or the constant (data format: according to the operation unit)
S2	Starting position of the search range (data format: according to the operation unit)
S3	End position of the search range (data format: according to the operation unit)
D	Device address to store the search result (data format: unsigned 32-bit integer)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modification (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●			●	●	●	●	●	●	●	●							●
S3	●	●	●	●			●	●	●	●	●	●	●	●							●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- This instruction searches the range specified by [S2] and [S3] for the search data specified by [S1].
- The search result is given in the data format of unsigned 32-bit integer, and stored as follows.

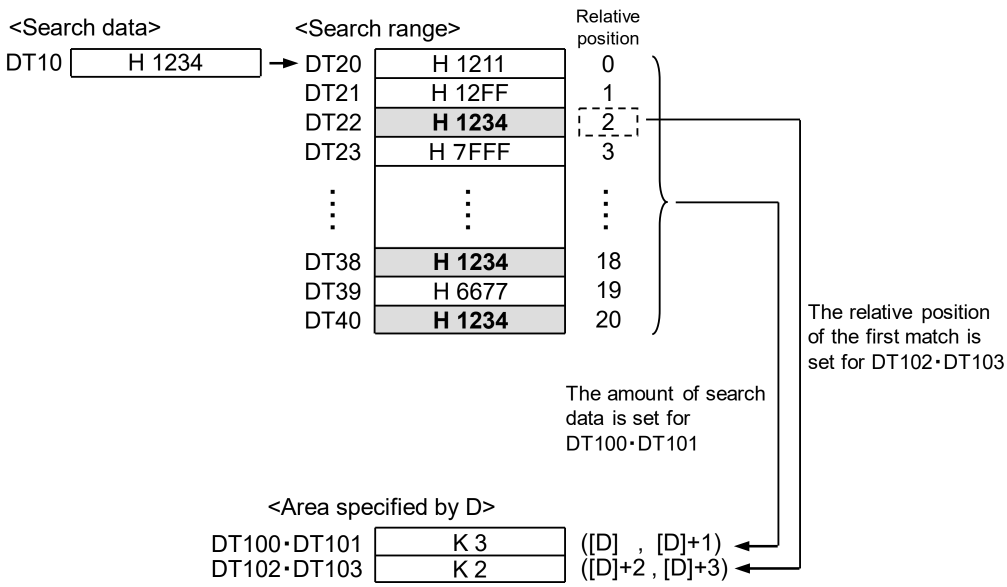
16-bit device	32-bit device	Description of output
[D], [D]+1	[D]	Store the number of data with the same value in a decimal form
[D]+2、[D]+3	[D]+1	Store the position of the first matching data (relative position with the first data as '0')

- The maximum amount of data that can be specified is 30000.
- Search is executed from [S2] to [S3].

■ Processing

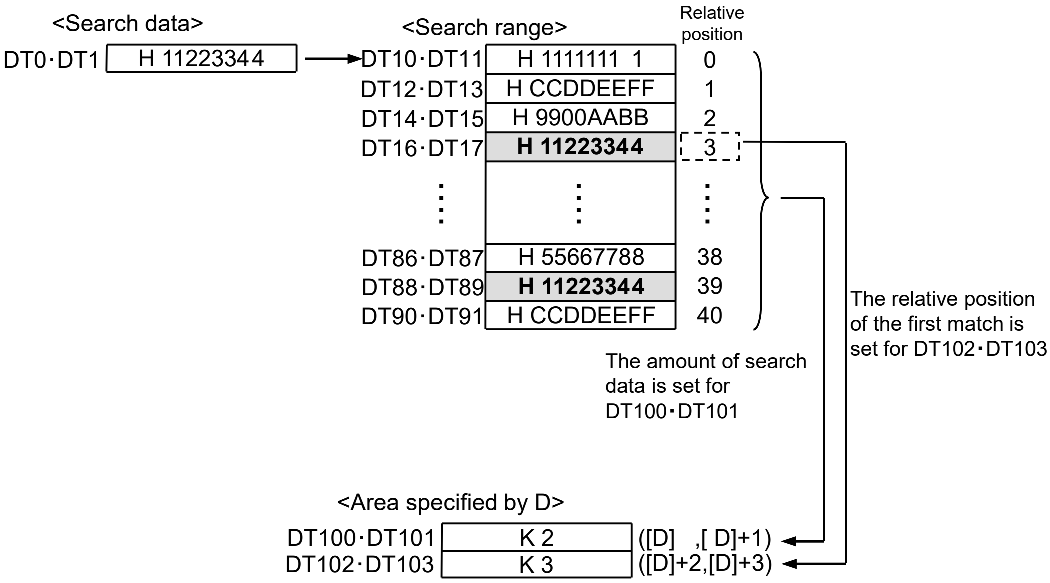
Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S1]…DT10 [S2]…DT20 [S3]…DT40 [D]…DT100



Example 2) When the operation unit is 32-bit (UL, SL, SF)

[i]…UL,SL,SF [S1]…DT0 [S2]…DT10 [S3]…DT90 [D]…DT100



■ Precautions for programming

- The end position of the search range is the device that contains [S3].

Example) When the operation unit is specified as 32 bits, the search range becomes the same whether a higher or lower address is specified for the [S3] device address.

[S2]…DT2 [S3]…DT6

DT0･DT1	H 11223344	
DT2･DT3	**H 55667788**	Search targets
DT4･DT5	**H 9900AABB**	
DT6･DT7	**H CCDDEEFF**	
DT8･DT9	H 12345678	

[S2]…DT2 [S3]…DT6

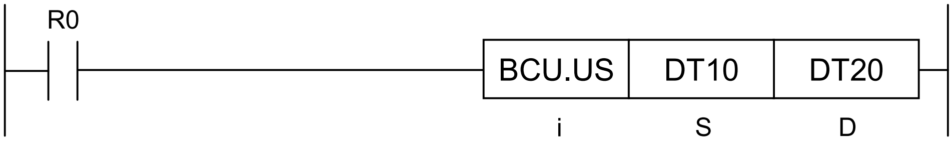
DT0･DT1	H 11223344	
DT2･DT3	**H 55667788**	Search targets
DT4･DT5	**H 9900AABB**	
DT6･**DT7**	**H CCDDEEFF**	
DT8･DT9	H 12345678	

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [S2] is larger than [S3].
	To be set when the [S2] area and the [S3] area differ.

10.2 BCU (ON Bits Count)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
S	Target device address or the constant (data format: according to the operation unit)
D	Device address to store the result (data format: unsigned 16-bit integer)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)		Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●	●	●								●
D	●	●	●	●			●	●	●		●										●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

- This function counts the number of ON bits (bits whose value is 1) in the data specified by [S], and stores the result in the device address specified by [D].
- The result is stored as an unsigned 16-bit integer.

■ Processing

Example 1) Operation units: 16 bits (US)

[i]…US [S1]…DT10 [D]…DT20

	15•••12	11•••8	7•••4	3•••0			
DT10	0000	0001	0011	0101	→	DT20	5

Example 2) Operation units: 32 bits (UL) (specify a 16-bit device for [S])

[i]…UL [S1]…DT10 [D]…DT20

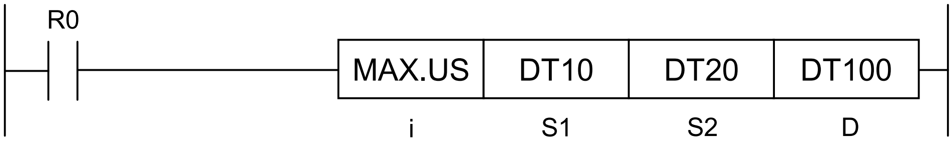
	15•••12	11•••8	7•••4	3•••0			
DT10	0000	0001	0011	0101	→	DT20	12
DT11	1110	0100	0101	0100			

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

10.3 MAX (Acquiring the Maximum Value)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Starting position of the search range for the maximum value (data format: according to the operation unit)
S2	End position of the search range for the maximum value (data format: according to the operation unit)
D	Device address to store the result of the search for the maximum value (data format: according to the operation unit)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●	●	●	●							●
S2	●	●	●	●			●	●	●	●	●	●	●	●							●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

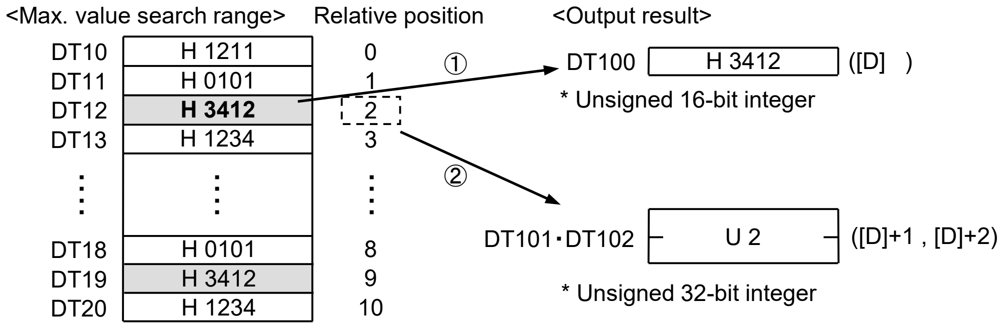
- The range of the device areas specified by [S1] and [S2] is searched for the maximum value.
- The resulting value is stored in the device area specified by [D], and the relative address value from [S1] is stored in [n].

- The relative address storage position ([D]+[n]) varies with the operation unit.
- The maximum amount of data that can be specified is 30000.
- [D] is in the following format according to the operation unit.

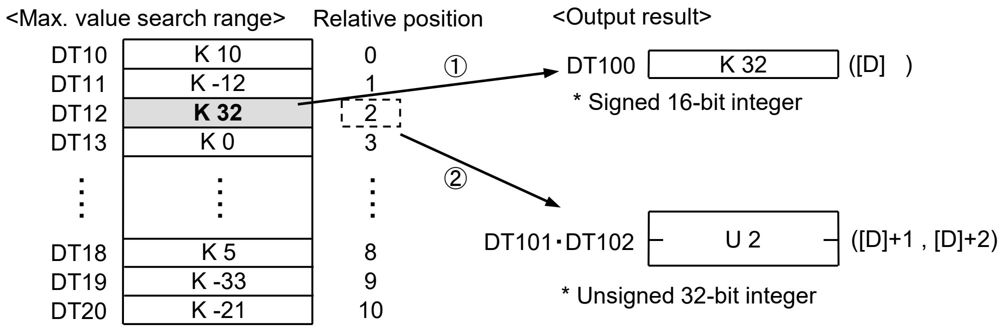
Operation unit	16 bits (US, SS)		32 bits (UL, SL, SF)		64 bits (DF)		Output content
Device size	16 bit	32 bit	16 bit	32 bit	16 bit	32 bit	
Result storage area	[D]	Cannot specify	[D] to [D +1]	[D]	[D] to [D +3]	[D] to [D +1]	Stores the maximum value.
	[D+1] to [D +2]	Cannot specify	[D+2] to [D +3]	[D+1]	[D+4] to [D +5]	[D+2]	Position of the first detected maximum value. Relative position with the start of [S1] as 0.

■ Processing

Example 1) Operation unit: 16 bits (US)
[i]…US
[S1]…DT10 [S2]…DT20 [D]…DT100



Example 2) Operation unit: 16 bits (SS)
[i]…SS
[S1]…DT10 [S2]…DT20 [D]…DT100



Example 3) Operation unit: 32 bits (UL) (specify a 32-bit device)

[i]…UL

[S1]…TS10 [S2]…TS50 [D]…TS100

<Max. value search range>

		Relative position
TS10	H 1111111 1	0
TS11	H CCDDEEFF	1
TS12	H 9900AABB	2
TS13	H 1234	3
⋮	⋮	⋮
TS48	H 55667788	38
TS49	**H FFFFFFFF**	39
TS50	H 9900AABB	40

<Output result>

① TS100	H FFFFFFFF	([D])
	* Unsigned 32-bit integer	
② TS101	U 39	([D]+1)
	* Unsigned 32-bit integer	

Example 4) Operation unit: 32 bits (SL) (specify a 32-bit device)

[i]…SL

[S1]…TS10 [S2]…TS50 [D]…TS100

<Max. value search range>

		Relative position
TS10	K 200	0
TS11	K -100	1
TS12	K 100	2
TS13	K 0	3
⋮	⋮	⋮
TS48	**K 300**	38
TS49	K -300	39
TS50	K -400	40

<Output result>

① TS100	K 300	([D])
	* Signed 16-bit integer	
② TS101	U 38	([D]+1)
	* Unsigned 32-bit integer	

Example 5) Operation unit: 32 bits (UL) (specify a 16-bit device)

[i]…UL

[S1]…DT10 [S2]…DT30 [D]…DT100

<Max. value search range>

		Relative position
DT10･DT11	H 1234FFFF	0
DT12･DT13	H 789A3456	1
DT14･DT15	H 9900AABB	2
DT16･DT17	H 12345678	3
⋮	⋮	⋮
DT26･DT27	H 55667788	8
DT28･DT29	H 9900AABB	9
DT30･DT31	**H AAAA0000**	10

<Output result>

① DT100･DT101	H AAAA0000	([D] , [D]+1)
	* Unsigned 32-bit integer	
② DT102･DT103	U 10	([D]+2 , [D]+3)
	* Unsigned 32-bit integer	

Example 6) Operation unit: 32 bits (SL) (specify a 16-bit device)

[i]…SL

[S1]…DT10 [S2]…DT30 [D]…DT100

<Max. value search range>		Relative position
DT10・DT11	K -10	0
DT12・DT13	K -20	1
DT14・DT15	K 100	2
DT16・DT17	K 0	3
⋮	⋮	⋮
DT26・DT27	**K 200**	8
DT28・DT29	K -300	9
DT30・DT31	K -400	10

<Output result>

① DT100・DT101 | K 200 | ([D] , [D]+1)

* Signed 32-bit integer

② DT102・DT103 | U 8 | ([D]+2 , [D]+3)

* Unsigned 32-bit integer

Example 7) Operation unit: Single-precision, floating-point real number (SF)

[i]…SF

[S1]…DT10 [S2]…DT90 [D]…DT100

<Max. value search range>		Relative position
DT10・DT11	SF 1.11	0
DT12・DT13	SF 3.33	1
DT14・DT15	SF 2.22	2
DT16・DT17	SF 4.44	3
⋮	⋮	⋮
DT86・DT87	**SF 5.55**	38
DT88・DT89	SF 1.11	39
DT90・DT91	SF 3.33	40

<Output result>

① DT100・DT101 | SF 5.55 | ([D] , [D]+1)

* Single precision floating point real number (32 bits)

② DT102・DT103 | U 38 | ([D]+2 , [D]+3)

* Unsigned 32-bit integer

■ Precautions for programming

- The end position of the search range for the maximum value is the device that contains [S2].

Example) When the operation unit is specified as 32 bits, the max. value search range becomes the same whether a higher or lower address is specified for the [S2] device address.

[S1]…DT2 [S2]…DT6

DT0・DT1	H 11223344	
DT2・DT3	**H 55667788**	Max. value search range
DT4・DT5	**H 9900AABB**	
DT6・DT7	**H CCDDEEFF**	
DT8・DT9	H 12345678	

[S1]…DT2 [S2]…DT7

DT0・DT1	H 11223344	
DT2・DT3	**H 55667788**	Max. value search range
DT4・DT5	**H 9900AABB**	
DT6・DT7	**H CCDDEEFF**	
DT8・DT9	H 12345678	

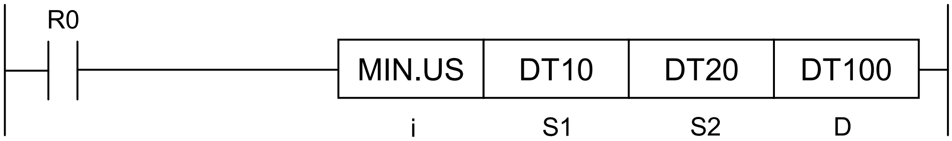
- Data is overwritten if [D] (search result for maximum value) is specified within the search range for the maximum value.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [S1] is larger than [S2].
	To be set when the [S1] device and the [S2] device differ.

10.4 MIN (Acquiring the Minimum Value)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Starting position of the search range for the minimum value (data format: according to the operation unit)
S2	End position of the search range for the minimum value (data format: according to the operation unit)
D	Device address to store the result of the search for the minimum value (data format: according to the operation unit)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●	●	●	●							●
S2	●	●	●	●			●	●	●	●	●	●	●	●							●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

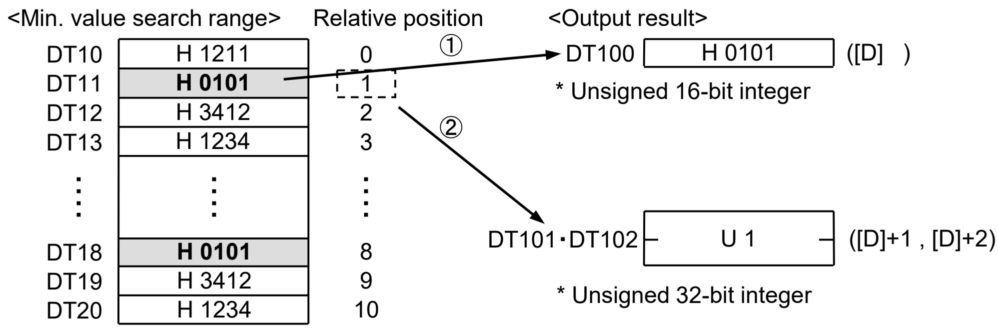
- The range of the device areas specified by [S1] and [S2] is searched for the minimum value. The resulting value is stored in the device area specified by [D], and the relative address value from [S1] is stored in [D]+[n].

- The relative address storage position ([D]+[n]) varies with the operation unit.
- The maximum amount of data that can be specified is 30000.
- [D] is in the following format according to the operation unit.

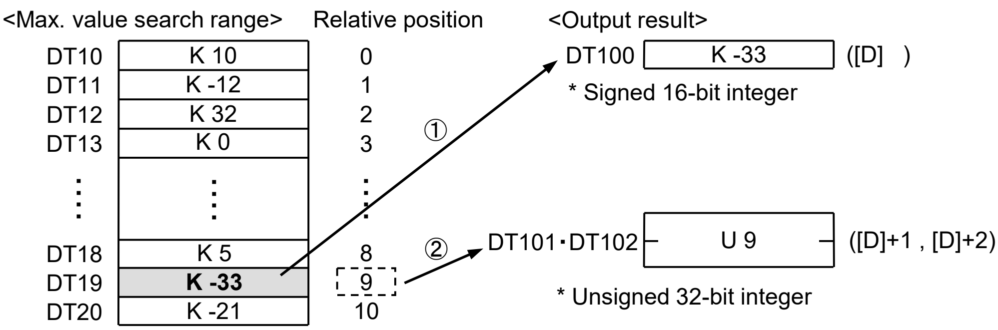
Operation unit	16 bits (US, SS)		32 bits (UL, SL, SF)		64 bits (DF)		Output content
Device size	**16 bit**	**32 bit**	**16 bit**	**32 bit**	**16 bit**	**32 bit**	
Result storage area	[D]	Cannot specify	[D] to [D +1]	[D]	[D] to [D +3]	[D] to [D +1]	Stores the minimum value
	[D+1] to [D +2]	Cannot specify	[D+2] to [D+3]	[D+1]	[D+4] to [D+5]	[D+2]	Position of the first detected maximum value. Relative position with the start of [S1] as 0.

■ Processing

Example 1) Operation unit: 16 bits (US)
[i]…US
[S1]…DT10 [S2]…DT20 [D]…DT100



Example 2) Operation unit: 16 bits (SS)
[i]…SS
[S1]…DT10 [S2]…DT20 [D]…DT100



Example 3) Operation unit: 32 bits (UL) (specify a 32-bit device)

[i]…UL

[S1]…TS10 [S2]…TS50 [D]…TS100

<Min. value search range> Relative position ①

TS10	**H 1111111 1**	0
TS11	H CCDDEEFF	1
TS12	H 9900AABB	2
TS13	H 12345678	3
⋮	⋮	⋮
TS48	H 55667788	38
TS49	H FFFFFFFF	39
TS50	H 9900AABB	40

<Output result>

TS100	H 1111111 1	([D])

* Unsigned 32-bit integer

②

TS101	U 0	([D]+1)

* Unsigned 32-bit integer

Example 4) Operation unit: 32 bits (SL) (specify a 32-bit device)

[i]…SL

[S1]…TS10 [S2]…TS50 [D]…TS100

<Max. value search range> Relative position

TS10	K 200	0
TS11	K -100	1
TS12	K 100	2
TS13	K 0	3
⋮	⋮	⋮
TS48	K 300	38
TS49	K -300	39
TS50	**K -400**	40

<Output result>

①

TS100	K -400	([D])

* Signed 16-bit integer

②

TS101	U 40	([D]+1)

* Unsigned 32-bit integer

Example 5) Operation unit: 32 bits (UL) (specify a 16-bit device)

[i]…UL

[S1]…DT10 [S2]…DT30 [D]…DT100

<Max. value search range> Relative position

DT10･DT11	H 1234FFFF	0
DT12･DT13	H 789A3456	1
DT14･DT15	H 9900AABB	2
DT16･DT17	H 12345678	3
⋮	⋮	⋮
DT26･DT27	**H 121112 FF**	8
DT28･DT29	H 9900AABB	9
DT30･DT31	H AAAA0000	10

<Output result>

①

DT100･DT101	H 121112 FF	([D] , [D]+1)

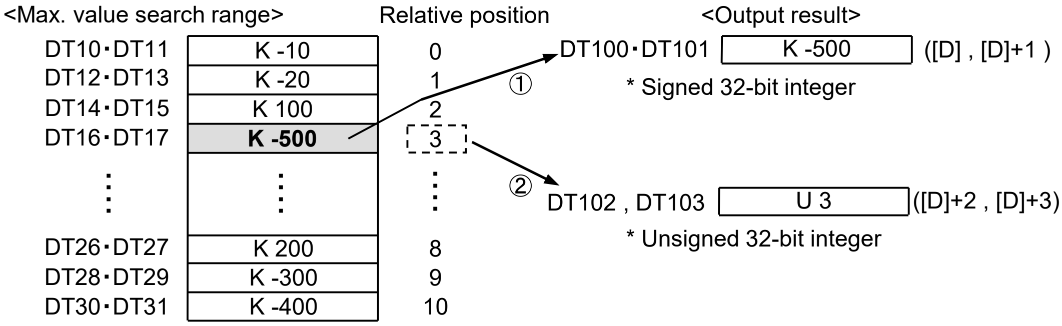
* Unsigned 32-bit integer

②

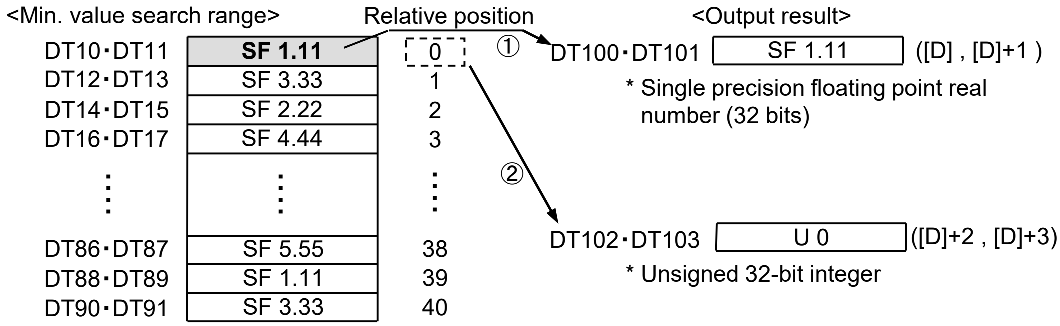
DT102･DT103	U 8	([D]+2 , [D]+3)

* Unsigned 32-bit integer

Example 6) Operation unit: 32 bits (SL) (specify a 16-bit device)
[i]…SL
[S1]…DT10 [S2]…DT30 [D]…DT100



Example 7) Operation unit: Single-precision, floating-point real number (SF)
[i]…SF
[S1]…DT10 [S2]…DT90 [D]…DT100



■ Precautions for programming

- The end position of the search range for the minimum value is the device that contains [S2].

Example) When the operation unit is specified as 32 bits, the min. value search range becomes the same whether a higher or lower address is specified for the [S2] device address.

[S1]…DT2 [S2]…DT6

DT0･DT1	H 11223344	
DT2･DT3	**H 55667788**	Min. value search range
DT4･DT5	**H 9900AABB**	
DT6･DT7	**H CCDDEEFF**	
DT8･DT9	H 12345678	

[S1]…DT2 [S2]…DT7

DT0･DT1	H 11223344	
DT2･DT3	**H 55667788**	Min. value search range
DT4･DT5	**H 9900AABB**	
DT6･DT7	**H CCDDEEFF**	
DT8･DT9	H 12345678	

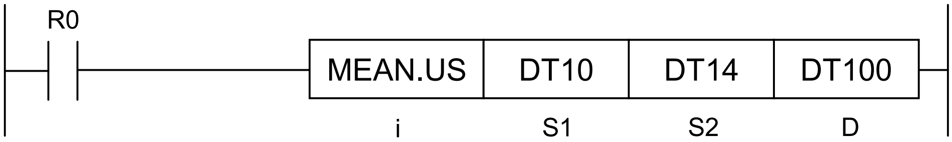
- Data is overwritten if [D] (search result for minimum value) is specified within the search range for the maximum value.

■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).
SR8	To be set when [S1] is larger than [S2].
(ER)	To be set when the [S1] device and the [S2] device differ.

10.5 MEAN (Acquiring the Total and the Mean Value)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Starting position of the target area (data format: according to the operation unit)
S2	End position of the target area (data format: according to the operation unit)
D	Device address to store the result (data format: according to the operation unit)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●	●	●	●							●
S2	●	●	●	●			●	●	●	●	●	●	●	●							●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

- This instruction stores the total and mean values within the device areas specified by [S1] and [S2] in the device area specified by [D].
- The storage position of the total and the mean values, with [D] as the starting address, varies with the operation unit.

- The total value uses twice the area size of the operation unit, and the mean value uses the same area size as the operation unit. Note that, in the case of floating point real number (SF), the total value also uses the same area size as the operation units.
- The maximum amount of data that can be specified is 30,000.
- When the operation unit is US, SS, UL or SL, the mean value is given as an integer rounding the first decimal point down.
- [D] is in the following format according to the operation unit.

Operation unit	16 bits (US, SS)		32 bits (UL, SL)		32 bits (SF)		64 bits (DF)		Output content
Device size	16 bit	32 bit	16 bit	32 bit	16 bit	32 bit	16 bit	32 bit	
Result storage area	[D] to [D +1]	Cannot specify	[D] to [D +3]	[D] to [D +1]		[D]	[D] to [D +3]	[D] to [D +1]	Total value
	[D+2]	Cannot specify	[D+4] to [D+5]	[D+2]	[D+2] to [D+3]	[D+1]	[D+4] to [D+7]	[D+2] to [D+3]	Mean value

■ Processing

Example 1) Operation unit: 16 bits (US)

Total value is given in 32-bit data, and the mean value is given in 16-bit data.

[i]…US

[S1]…DT10 [S2]…DT14 [D]…DT100

<Total/mean calculation range>

	Value
DT10	H 111 1
DT11	H 5555
DT12	H 7777
DT13	H AAAA
DT14	H FFFF

<Output result>

	Value	
DT100・DT101	**H 00028886**	Total value ([D] and [D]+1)

* Unsigned 32-bit integer

	Value	
DT102	**H 81B4**	Mean value ([D]+2)

* Unsigned 16-bit integer

Example 2) Operation unit: 16 bits (SS)

Total value is given in 32-bit data, and the mean value is given in 16-bit data.

[i]…SS

[S1]…DT10 [S2]…DT14 [D]…DT100

<Total/mean calculation range>

	Value
DT10	K 11
DT11	K -33
DT12	K 44
DT13	K 55
DT14	K -22

<Output result>

	Value	
DT100・DT101	**K -55**	Total value ([D] and [D]+1)

* Signed 32-bit integer

	Value	
DT102	**K 11**	Mean value ([D]+2)

* Signed 16-bit integer

Example 3) Operation unit: 32 bits (UL) (specify a 32-bit device)

Total value is given in 64-bit data, and the mean value is given in 32-bit data.

[i]…UL

[S1]…TS10 [S2]…TS14 [D]…TS100

<Total/mean calculation range>

	Value
TS10	H 00000000
TS11	H 22222222
TS12	H 33333333
TS13	H 44444444
TS14	H 55555555

<Output result>

	Value	
TS100･TS101	**H EEEEEEEE**	Total value ([D] and [D]+1)

* Unsigned 64-bit integer

	Value	
TS102	**H 2FC962FC**	Mean value ([D]+2)

* Unsigned 32-bit integer

Example 4) Operation unit: 32 bits (SL) (specify a 32-bit device)

Total value is given in 64-bit data, and the mean value is given in 32-bit data.

[i]…SL

[S1]…TS10 [S2]…TS14 [D]…TS100

<Total/mean calculation range>

	Value
TS10	K 1000
TS11	K 2000
TS12	K -3000
TS13	K -4000
TS14	K -5000

<Output result>

	Value	
TS100･TS101	**K -9000**	Total value ([D] and [D]+1)

* Signed 64-bit integer

	Value	
TS102	**K -1800**	Mean value ([D]+2)

* Signed 32-bit integer

Example 5) Operation unit: 32 bits (UL) (specify a 16-bit device)

Total value is given in 64-bit data, and the mean value is given in 32-bit data.

[i]…UL

[S1]…DT10 [S2]…DT14 [D]…DT100

<Total/mean calculation range>

	Value
DT10･DT11	H 1111000 0
DT12･DT13	H 33332222
DT14･DT15	H 55554444
DT16･DT17	H 77776666

<Output result>

	Value	
DT100-DT103	**H 11110 CCCC**	Total value ([D] to [D]+3)

* Unsigned 64-bit integer

	Value	
DT104･DT105	**H 44443333**	Mean value ([D]+4, [D]+5)

* Unsigned 32-bit integer

Example 6) Operation unit: 32 bits (SL) (specify a 16-bit device)

Total value is given in 64-bit data, and the mean value is given in 32-bit data.

[i]…SL

[S1]…DT10 [S2]…DT16 [D]…DT100

<Total/mean calculation range>

	Value
DT10･DT11	K -100
DT12･DT13	K 600
DT14･DT15	K 500
DT16･DT17	K -200

<Output result>

	Value	
DT100･DT103	**K 800**	Total value ([D] to [D]+3)

* Signed 64-bit integer

	Value	
DT104･DT105	**K 200**	Mean value ([D]+4, [D]+5)

* Signed 32-bit integer

Example 7) Operation unit: Single-precision, floating-point real number (SF) (specify a 16-bit device)

Total value is given in 32-bit data, and the mean value is given in 32-bit data.

[i]…SF

[S1]…DT10 [S2]…DT24 [D]…DT100

<Total/mean calculation range>

	Value
DT10･DT11	SF 3.33E+00
DT12･DT13	SF 1.11E+00
DT14･DT15	SF 4.44E+00
DT16･DT17	SF 5.55E+00
DT18･DT19	SF 2.22E+00
DT20･DT21	SF 1.11E+00
DT22･DT23	SF 3.33E+00
DT24･DT25	SF 5.55E+00

<Output result>

	Value	
DT100･DT101	**SF 2.66E+01**	Total value ([D] and [D]+1)

* Single precision floating point real number (32 bits)

	Value	
DT102･DT103	**SF 3.33E+00**	Mean value ([D]+2, [D]+3)

* Single precision floating point real number (32 bits)

■ Precautions for programming

- The end position of the total and the mean calculation range is the device that contains [S2].
  Example) When the operation unit is 32 bits, the calculation range is the same whether a device number of higher level or lower level is specified.

[S1]…DT2 [S2]…DT6

DT0･DT1	H 11223344	
DT2･DT3	**H 55667788**	Total/mean calculation range
DT4･DT5	**H 9900AABB**	Total/mean calculation range
DT6･DT7	**H CCDDEEFF**	Total/mean calculation range
DT8･DT9	H 12345678	

[S1]…DT2 [S2]…DT7

DT0･DT1	H 11223344	
DT2･DT3	**H 55667788**	Total/mean calculation range
DT4･DT5	**H 9900AABB**	Total/mean calculation range
DT6･DT7	**H CCDDEEFF**	Total/mean calculation range
DT8･DT9	H 12345678	

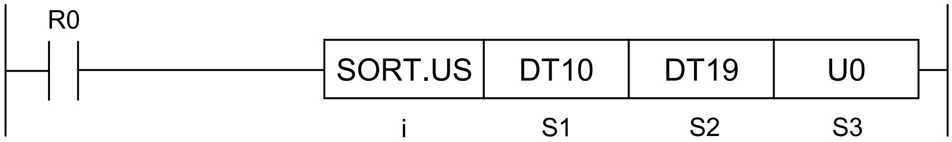
- Data is overwritten if [D] (total and mean calculation results) is specified within the total and the mean calculation range.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [S1] is larger than [S2].
	To be set when the [S1] device and the [S2] device differ.

10.6 SORT (Sort)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Starting position of the target area (data format: according to the operation unit)
S2	End position of the target area (data format: according to the operation unit)
S3	Sort condition (data format: unsigned 16-bit integer)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1							●	●													●
S2							●	●													●
S3	●	●	●	●			●	●								●	●				●

(Note 1) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

- Data in the range from the area specified by [S1] to the area specified by [S2] are sorted in ascending order or in descending order, according to the sort conditions specified by [S3].
- Available sort conditions for [S3] are as follows.
  U0: Ascending order
  U1: Descending order
- The maximum amount of data that can be specified is 30000.

■ Processing

Example 1) Operation unit: 16 bits (US)

[i]…US

[S1]…DT10 [S2]…DT19 [S3]…U0 (Ascending order)

<Before sort>

DT10	H 0123
DT11	H 111 1
DT12	H 3210
DT13	H 2222
DT14	H 3333
DT15	H 0000
DT16	H 3210
DT17	H 4321
DT18	H 3333
DT19	H 5432

<After sort>

DT10	H 0000
DT11	H 0123
DT12	H 111 1
DT13	H 2222
DT14	H 3210
DT15	H 3210
DT16	H 3333
DT17	H 3333
DT18	H 4321
DT19	H 5432

Example 2) Operation unit: 16 bits (SS)

[i]…SS

[S1]…DT10 [S2]…DT19 [S3]…U1 (Descending order)

<Before sort>

DT10	K 300
DT11	K10
DT12	K 3
DT13	K -1
DT14	K 1000
DT15	K -30
DT16	K 100
DT17	K 30
DT18	K 1
DT19	K -3

<After sort>

DT10	K 1000
DT11	K 300
DT12	K 100
DT13	K 30
DT14	K 10
DT15	K 3
DT16	K 1
DT17	K -1
DT18	K -3
DT19	K 30

Example 3) Operation unit: 32 bits (UL)

[i]…UL

[S1]…DT10 [S2]…DT19 [S3]…U0 (Ascending order)

<Before sort>

DT10･DT11	H 22220000
DT12･DT13	H 1111333 3
DT14･DT15	H 55550000
DT16･DT17	H 22222222
DT18･DT19	H 1111444 4

<After sort>

DT10･DT11	H 1111333 3
DT12･DT13	H 1111444 4
DT14･DT15	H 22220000
DT16･DT17	H 22222222
DT18･DT19	H 55550000

Example 4) Operation unit: 32 bits (SL)

[i]…SL

[S1]…DT10 [S2]…DT19 [S3]…U1 (Descending order)

<Before sort>

DT10･DT11	K 11
DT12･DT13	K 33
DT14･DT15	K 55
DT16･DT17	K 22
DT18･DT19	K 44

<After sort>

DT10･DT11	K 55
DT12･DT13	K 44
DT14･DT15	K 33
DT16･DT17	K 22
DT18･DT19	K 11

Example 5) Operation unit: Single-precision, floating-point real number (SF)
[i]…SF
[S1]…DT10 [S2]…DT19 [S3]…U1 (Descending order)

<Before sort>

DT10･DT11	SF 3.33
DT12･DT13	SF 11.11
DT14･DT15	SF 2.22
DT16･DT17	SF 1111 .1
DT18･DT19	SF 4.44

<After sort>

DT10･DT11	SF 1111 .1
DT12･DT13	SF 11.11
DT14･DT15	SF 4.44
DT16･DT17	SF 3.33
DT18･DT19	SF 2.22

■ Precautions for programming

- It must be noted that, since the time for data comparison increases in proportion to the square of the number of data, the sorting process can take a long time when there is a large amount of data to be sorted.
- During sort execution, data in [S1] to [S2] are sorted in sequence according to the sort conditions.
- The end position of the sort range is the device that contains [S2].
  Example) When the operation unit is 32 bits, the sort range is the same whether a device number of higher level or lower level is specified.

[S1]…DT2 [S2]…DT6

DT0･DT1	H 11223344	
DT2･DT3	**H 55667788**	Sort targets
DT4･DT5	**H 9900AABB**	Sort targets
DT6･DT7	**H CCDDEEFF**	Sort targets
DT8･DT9	H 12345678	

[S1]…DT2 [S2]…DT7

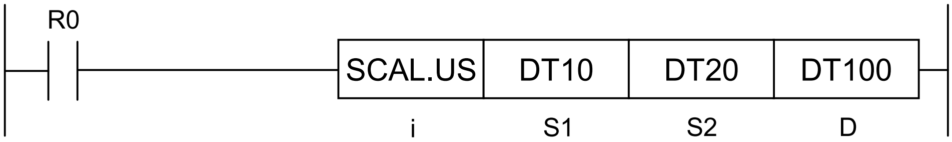
DT0･DT1	H 11223344	
DT2･DT3	**H 55667788**	Sort targets
DT4･DT5	**H 9900AABB**	Sort targets
DT6･**DT7**	**H CCDDEEFF**	Sort targets
DT8･DT9	H 12345678	

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [S1] is larger than [S2].
	To be set when the [S1] device and the [S2] device differ.

10.7 SCAL (Linearization)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Data equivalent to the input value X, or the area to store it (data format: according to the operation unit)
S2	Starting address of the data table used for scaling (linearization) (data format: according to the operation unit)
D	Area to store the output result Y (data format: according to the operation unit)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●										●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

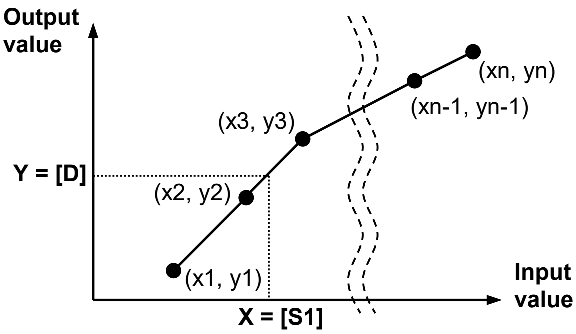
(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

Outline of operation

- The data specified by [S1] are scaled according to the data table specified by [S2]. The result is stored in the device area specified by [D].

Processing

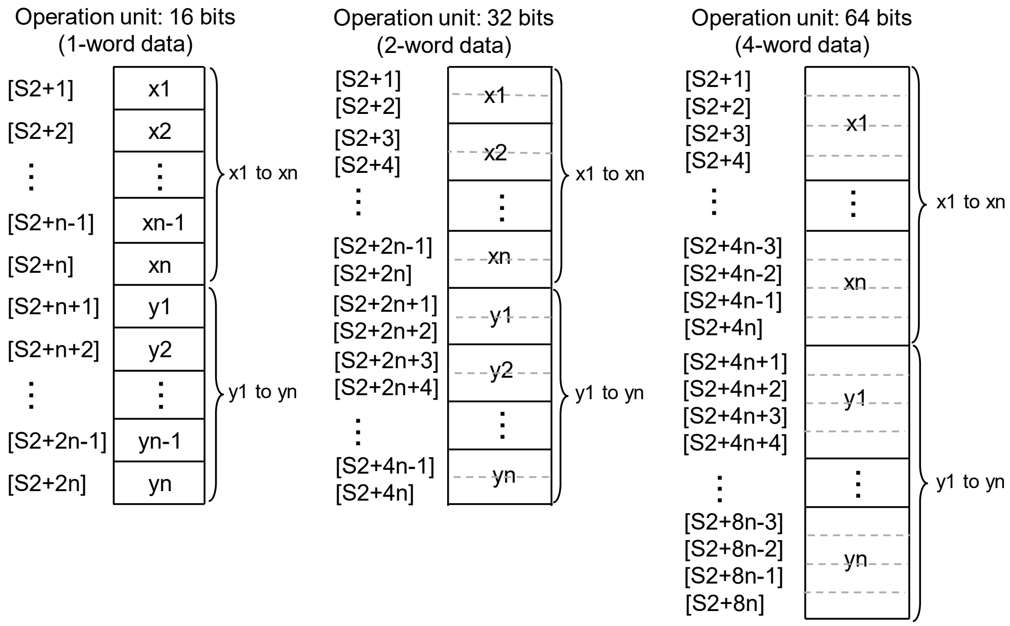
- Regardless of the operation unit, the output value Y for [D], corresponding to the input value X for [S1], is calculated.
- In response to the information of input value X specified by [S1], the information of output value Y is calculated according to the [S2] data table. The result is stored into the [D] area.



Structure of data table [S2]

- The number for data table "n" (setting range: 2 to 256) is determined by the value [n] specified for the starting address of the data table [S2]. n is 16-bit data, regardless of the operation unit.
- Scaling data (x1 to xn, y1 to yn) is stored from [S2]+1. Depending on the operation unit, the data table occupies 16 bits (1 word), 32 bits (2 words), or 64 bits (4 words).
- The structure of data table [S2] that is used for scaling (linearization) is shown below.

[S2] | n | Number for the data table (Available range: 2 to 256)



■ Precautions for programming

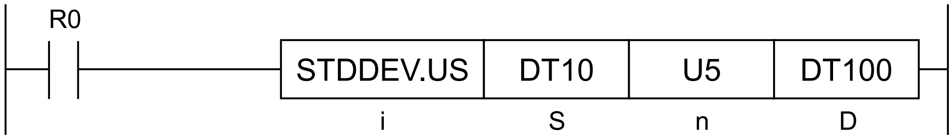
- For data in the data table [S2], ensure that Xn-1 is smaller than Xn.
- When X([S1]) is smaller than x1], Y([D]) equals y1.
- When X([S1]) is larger than xn, Y([D]) equals yn.
- The maximum value for [n] that indicates the number of data in the data table [S2] is 256.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [n] which indicates the number of data in the data table [S2] is smaller than 2 or larger than 256.
	To be set when data in the data table [S2] exceeds the area.
	To be set when Xn is not in ascending order.

10.8 STDDEV (Acquiring the Variance and Standard Deviation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S	Specify the starting position of a target area (data format: according to the operation unit).
n	Specify the number of target data (data format: unsigned 16-bit integer).
D	Specify the device address to store results.

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U (Note 2)	H (Note 3)	SF	DF	" "	
S	●	●	●	●			●	●	●	●	●										●
n	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 2) Can be specified only when the operation unit is unsigned integer (US).

(Note 3) Can be specified only when the operation unit is an integer (US, SS).

■ Outline of operation

- This instruction stores the variance and standard deviation within the range of the device areas specified by [S] and [n] into the device area specified by [D].
- The maximum amount of data that can be specified is 30000.
- The result [D] is output as single-precision real numbers.

16-Bit device:	Output content
[D], [D]+1	Stores variance.
[D]+2, [D]+3	Stores standard deviation.

■ Processing

Method for calculating variance and standard deviation

(Conditions) N data x_1, x_2,...x_n

1. Mean value

$$m = \frac{x_1 + x_2 + \ldots + x_n}{N}$$

2. Variance

$$\sigma^2 = \frac{(x_1 - m)^2 + (x_2 - m)^2 + \ldots + (x_n - m)^2}{N}$$

3. Standard deviation

$$\sigma = \sqrt{\sigma^2}$$

Example 1) Operation unit: 16 bits (US)

[S]...DT10 [n]...U5 [D]...DT100

<Calculation range>

DT10	U 2
DT11	U 4
DT12	U 5
DT13	U 6
DT14	U 8

<Output result>

DT100 DT101	**SF 4.00E+00**	Variance
	* Single-precision real number (32-bit)	
DT102 DT103	**SF 2.00E+00**	Standard deviation
	* Single-precision real number (32-bit)	

The following results are stored.

- Variance of S to S+4 is stored in (D, D+1).
- Standard deviation of S to S+4 is stored in (D+2, D+3).

Example 2) Operation unit: 16 bits (SS)

[S]...DT10 [n]...U5 [D]...DT100

<Calculation range>

DT10	K 16
DT11	K -20
DT12	K 32
DT13	K -35
DT14	K -12

<Output result>

DT100 DT101	**SF 5.9536E+02**	Variance
	* Single-precision real number (32-bit)	
DT102 DT103	**SF 2.4400E+01**	Standard deviation
	* Single-precision real number (32-bit)	

The following results are stored.

- Variance of S to S+4 is stored in (D, D+1).
- Standard deviation of S to S+4 is stored in (D+2, D+3).

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [S+n] exceeds the device address.
	To be set when the result storage area exceeds the device address.
	To be set when the specified ranges of [S1] and [D] overlap.

10.9 EVENTC (Instruction to Count the Number of Events)

■ Ladder diagram

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i				●			

■ List of operands

Operand	Description
S1	Starting address of the counting starting position
S2	Starting address of the working area for counting
n	Number of bits to be counted
D	Starting device address to store the count result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●										●
S2	●	●	●	●			●	●													●
n	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

(Note 1) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

- This instruction counts the number of ONs for [n] bits from the device specified by [S1].
- During the counting operation, a data area that is equal to n/16 words from the device specified in [S2] is used.
- The count result is stored in the area of [n]*2 words starting with [D].
- The working area with the starting address specified by [S2] needs the following number of words:

[(Bits to be counted -1)/16] + 1

Rounded down to the nearest integer

- The number of bits to be counted that is specified by [n] is 1 to 65535. However, this range must not exceed the device area.

Processing

Example 1) Carry out counting for 8 points from WX0

[i]…UL [S1]…WX0 [S2]…DT0 [n]…U 8 [D]…DT10

[S1]

WX0	Targeted bits: 8 bits from X0 to X7

[S2]

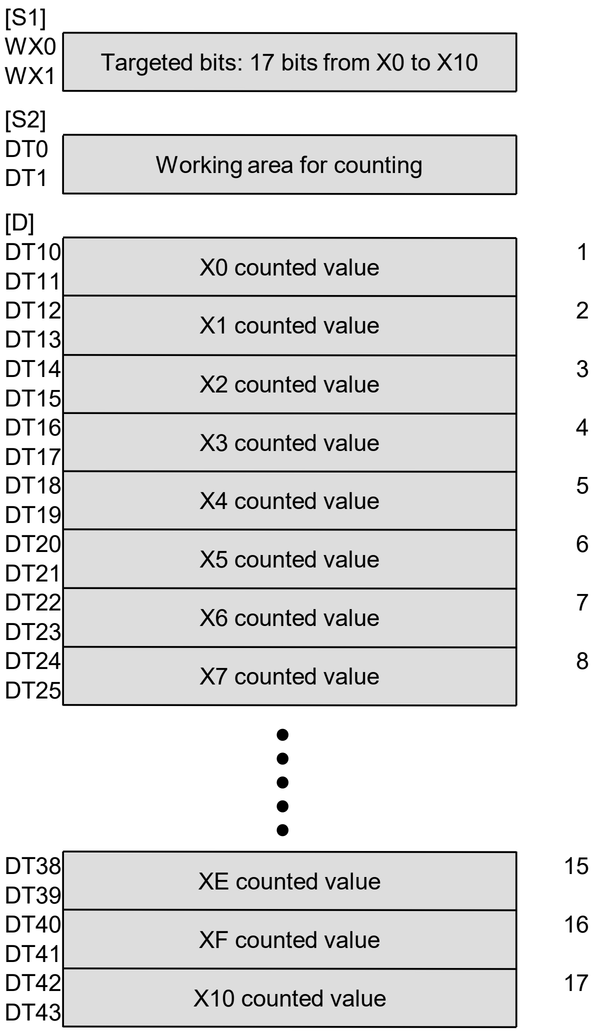
DT0	Working area for counting

[D]

DT10 DT11	X0 counted value	1
DT12 DT13	X1 counted value	2
DT14 DT15	X2 counted value	3
DT16 DT17	X3 counted value	4
DT18 DT19	X4 counted value	5
DT20 DT21	X5 counted value	6
DT22 DT23	X6 counted value	7
DT24 DT25	X7 counted value	8

Example 2) Carry out counting for 17 points from WX0

[i]…UL   [S1]…WX0   [S2]…DT0   [n]…U 17   [D]…DT10



■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	When one of the count target area, the count working area or the count result storage area exceeds the device area.

10.10 EVENTT (Instruction to Count the Time of Events)

■ Ladder diagram

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i				●			

■ List of operands

Operand	Description
S1	Starting address of the counting starting position
S2	Starting address of the working area for counting
n	Number of bits to be counted
D	Starting device address to store the count result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●										●
S2	●	●	●	●			●	●													●
n	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●	●	●										●

(Note 1) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants and character constants cannot be specified.)

■ Outline of operation

- This instruction counts the time of ONs for [n] bits from the device specified by [S1].
- The count result is stored in the area of [n]*2 words starting with [D].
- The working area with the starting address specified by [S2] needs the following number of words:

  [(Bits to be counted -1)/16] + 1

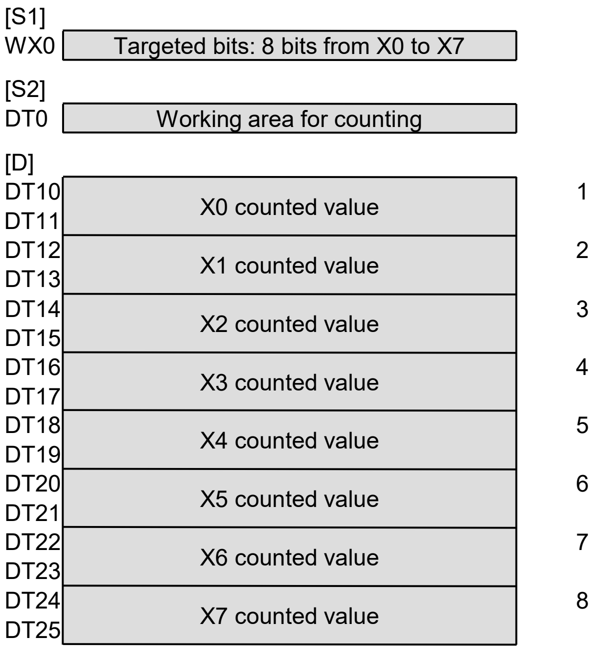
  Fractions are rounded down

- The number of bits to be counted that is specified by [n] is 1 to 65535. However, this range must not exceed the device area.

■ Processing

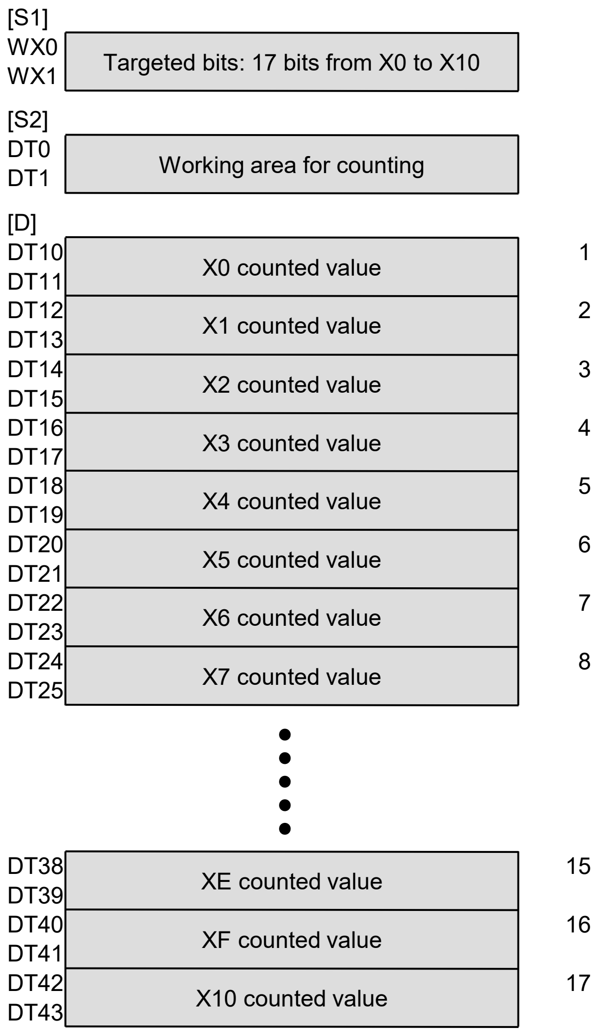
Example 1) Carry out counting for 8 points from WX0

[i]…UL   [S1]…WX0   [S2]…DT0   [n]…U 8   [D]…DT10



Example 2) Carry out counting for 17 points from WX0

[i]…UL [S1]…WX0 [S2]…DT0 [n]…U 17 [D]…DT10

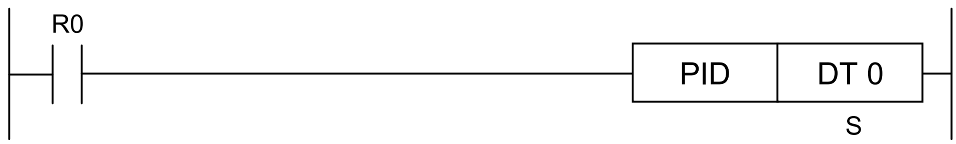


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	When one of the count target area, the count working area or the count result storage area exceeds the device area.

10.11 PID (PID Operation)

■ Ladder diagram



■ List of operands

Operand	Description
S	Starting number of the PID operation parameter area (30 words)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S							●														

■ Outline of operation

- PID operation is carried out to retain the process value PV stored in [S+2], in consistency with the set point value SP specified by [S+1].
- The operation result is stored, as a manipulated value [MV], in the area specified by [S+3].
- Methods for PID operation (derivative-first / proportional-plus-derivative-first, reverse operation / forward operation) and coefficients used for PID operation (proportional gain, integral time, derivative time), as well as the types and interval of operation, are set to the parameter table [S] to [S+29].

■ Types of PID operation

Items	Description
Reverse operation/ forward operation	Select the upward/downward direction of output in the case of change to the process. Specify "reverse operation" if output is increased when the process value decreases (e.g. heating). Specify "forward operation" if output is increased when the process value increases (e.g. cooling).
Derivative-first PID /Proportional-plus-derivative-first PID	Derivative-first PID: Usually, when the set point value is changed, the output variation becomes larger but it converges faster.
	Proportional-plus-derivative-first PID: Usually, when the set point value is changed, the output variation becomes smaller but it converges slower.

Items	Description
Auto-tuning	By measuring process response, the respective optimal values for Kp, Ti and Td as PID parameters are measured. When auto-tuning is executed, the estimated results are reflected in the parameter area after auto-tuning is complete.

■ Parameter table settings

[S]	Control mode
[S+1]	Set point value (SP)
[S+2]	Process value (PV)
[S+3]	Manipulated value (MV)
[S+4]	MV lower limit
[S+5]	MV upper limit
[S+6]	Proportional gain (Kp)
[S+7]	Time integral (Ti)
[S+8]	Time derivative (Td)
[S+9]	Control interval (Ts)
[S+10]	Auto-tuning progress
[S+11] ≈ [S+29]	PID operation work area

■ Setting of parameters: [S] to [S+29]

Operand	Parameter name	Setting range	Setting method
[S]	Control mode	H0 to H3	When not executing auto-tuning H0: Derivative-first, reverse operation H1: Derivative-first, forward operation H2: Proportional-plus-derivative-first, reverse operation H3: Proportional-plus-derivative-first, forward operation
		H8000 to H8003	When executing auto-tuning H8000: Derivative-first, reverse operation H8001: Derivative-first, forward operation H8002: Proportional-plus-derivative-first, reverse operation H8003: Proportional-plus-derivative-first, forward operation
[S+1]	Set point value (SP)	K0 to K10000	Set the target value for the process control within the following range.
[S+2]	Process value (PV)	K0 to K10000	Use an A/D conversion unit, etc., to input the current value of process control.
[S+3]	Output value (MV)	K0 to K10000	PID operation result is stored. Output the value using a D/A converter, etc.
[S+4]	Output lower limit	K0 to K9999	Specify the output value (MV) range. Values for the specified range are output.
[S+5]	Output upper limit	K1 to K10000	
[S+6]	Proportional gain (Kp)	K1 to K9999 (0.1 to 999.9)	Specify the coefficient used for PID operation. The setting value × 0.1 is the actual proportional gain.(Note 1)

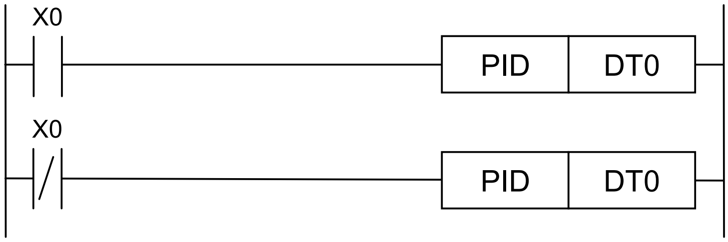
Operand	Parameter name	Setting range	Setting method
[S+7]	Integral time (Ti)	K1 to K30000 (0.1 to 3000 s)	Specify the coefficient used for PID operation. Actual integral time is set point value × 0.1. The integral operation is not executed if 0 is specified.(Note 1)
[S+8]	Control interval (Td)	K0 to K10000 (0 to 1000 s)	Specify the coefficient used for PID operation. Actual derivative time is set point value × 0.1. The integral operation is not executed if 0 is specified.(Note 1)
[S+9]	Control interval (Ts)	K1 to K6000 (0.01 to 60.0 s)	Specify the interval for executing the PID operation. The setting value × 0.01 is the actual control interval.
[S+10]	Auto-tuning progress status	K0 to K5	When auto-tuning is specified in the operation mode, the degree of progress of auto-tuning is displayed. The values of K1 to K5 are stored according to the progress status from the default value [0], and are returned to the default value after auto-tuning is completed.
[S+11] to [S+29]	Work area for PID operation	-	This work area is used by the system program for operation.

(Note 1) If auto-tuning is specified in the control mode, automatic adjustment is carried out, and the set point value is rewritten.

■ Precautions for programming

- Including the work area for operation, a 30 word area is required for the parameter table. Ensure that this area value is not rewritten by other instructions.
- Error is not detected even if the parameter table exceeds the area. When specifying [S], specify a number that is within a minimum of 30 words from the last number.
- Ensure that the area is not exceeded by index modification. Error is not detected even if the area is exceeded.
- As indicated below, the system does not operate correctly if two or more PID instructions specifying the same table are described in the program. Even if execution conditions are not met, PID instructions operate internally using the specified table. In such cases, set the tables to separate addresses.

＜例＞



■ Precautions when executing auto-tuning

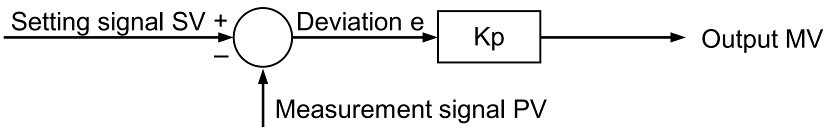
- Auto-tuning may not be executable depending on the process. In that case, the system returns to the original parameter operation.
- After auto-tuning is completed, the control mode [S] area is automatically rewritten from H 8000 through H 8003 to H 0 through H 3. Ensure that this is not rewritten again by your program, etc.

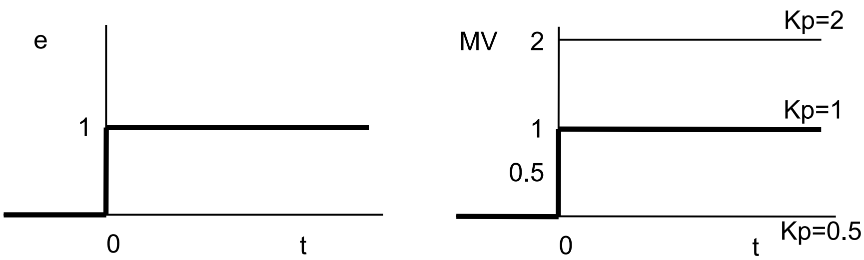
- After auto-tuning is completed, the respective optimal values are stored for proportional gain [Kp], integral time [Ti], and derivative time [Td]. Before execution, it is necessary to specify appropriate values (e.g. lower limits) within the respective setting ranges.
- After auto-tuning is complete, store optimal values for proportional gain [Kp], integral time [Ti], and derivative time [Td]. Be careful that the stored values are not rewritten.
- During auto-tuning, the values for Kp, Ti and Td are calculated by measuring changes to process values (PVs) when manipulated values (MVs) are set to the upper limits, as well as changes to process values (PVs) when manipulated values (MVs) are set to the lower limits, so that process values (PVs) will be increased or decreased in accordance with the respective set point values (SPs).
- Changes to manipulated values (MVs) of auto-tuning are completed in three sessions (upper limit output - lower limit output - upper limit output) in the fewest times possible. If the auto-tuning progress does not change from 0 after multiple sessions, shorten the control interval Ts and retry auto-tuning.

■ Overview of PID control

(1) Proportional operation

Control operation that produces an output proportional to the size of the input

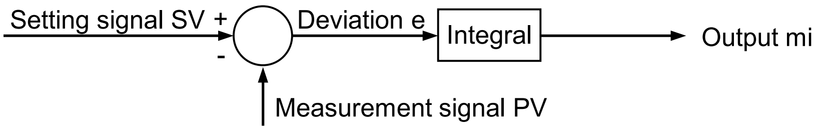




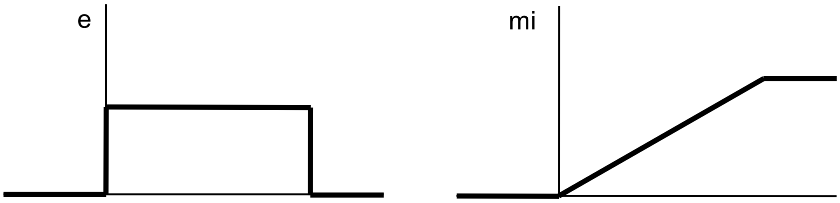
- A constant control quantity is maintained.
- Offset (steady-state deviation) remains.
- The larger the Kp value, the stronger the action of the proportional operation.

(2) Integral operation

Control operation that produces an output proportional to the integral time of the input.



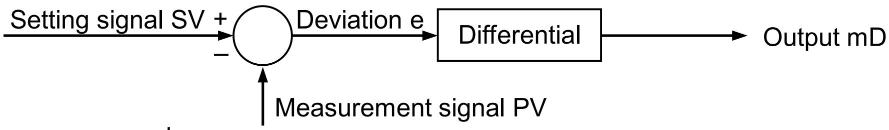
$mi=1/Ti \int edt$



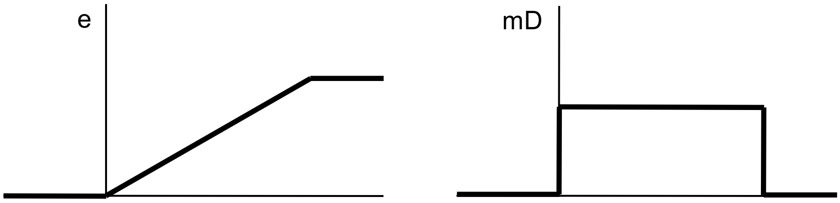
- The resulting offset is removed by combining with proportional operation or proportional-derivative operation.
- The smaller the Ti value, the stronger the action of the integral operation.

(3) Derivative operation

Control operation that produces an output proportional to the time derivative value of the input.



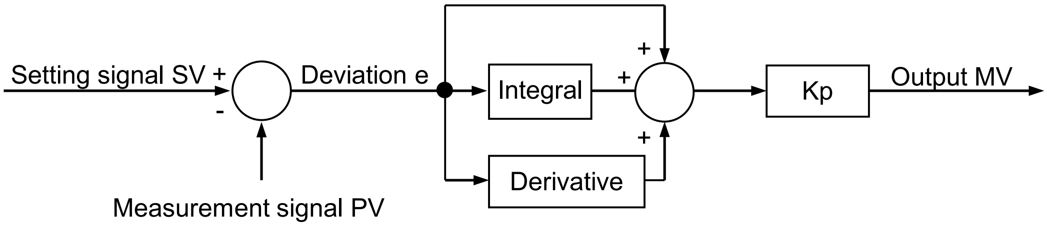
$mD=TD \cdot \frac{de}{dt}$



- The advancing property of derivative operation reduces the negative effects that the delaying property of the process has on control.
- The larger the Td value, the stronger the action of the derivative operation.
- Pure derivative operation becomes temporarily inoperative if noise, etc., is input. This has a negative effect on the controlled process, so incomplete derivative operation is executed.

(4) PID operation

A combination of proportional, integral, and derivative operation is called PID operation.



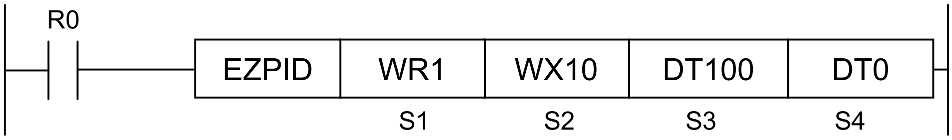
- If the parameters in PID control are set to their optimal values, the control quantity can be quickly matched to the target value and maintained.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when a parameter is out of the setting range.

10.12 EZPID (PID Operation: PWM Output Available)

■ Ladder diagram



■ List of operands

Operand	Description
S1	1-word area for setting control data that determine methods for auto-tuning and output
S2	1-word area for inputting the process value (PV)
S3	4-word area for setting the set point value (SV), proportional gain (Kp), integral time (Ti), and derivative time (Td).
S4	30-word area comprising the following: Storage area for the manipulated value (MV), control interval (Ts), control mode, setting area for parameters related to auto-tuning, and operation work area

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●													
S2	●	●	●	●	●	●	●	●													
S3	●	●	●	●			●	●													
S4	●	●	●	●			●	●													

■ Outline of operation

- PID operation is carried out to retain the process value (PV) stored in [S+2], in consistency with the set point value (SP) specified by [S3].
- The operation result is stored, as a manipulated value (MV), in the area specified by [S4]. If an OUT instruction is described immediately after this instruction, PWM output (ON-OFF output) can be gained, proportionate to the manipulated value.
- Set the parameters used for PID operation (proportional gain, integral time, and derivative time) to [S3+1] to [S3+3]. The auto-tuning function for automatically calculating these values is also available.
- To change the method for PID operation (derivative-first operation/proportional-plus-derivative-first operation, reverse operation/forward operation) or the control interval Ts, etc., set them to the area [S4] to [S4+9].

■ Types of PID operation

Items	Description
Reverse operation/ forward operation	Select the upward/downward direction of output in the case of change to the process. Specify "reverse operation" if output is increased when the process value decreases (e.g. heating). Specify "forward operation" if output is increased when the process value increases (e.g. cooling).
Derivative-first PID /Proportional-plus-derivative-first PID	Derivative-first PID: Usually, when the set point value is changed, the output variation becomes larger but it converges faster.
	Proportional-plus-derivative-first PID: Usually, when the set point value is changed, the output variation becomes smaller but it converges slower.
Auto-tuning	By measuring process response, the respective optimal values for Kp, Ti and Td as PID parameters are measured. When auto-tuning is executed, the estimated results are reflected in the parameter area after auto-tuning is complete.

(Note 1) By default, controls are carried out in reverse operation, derivative-first. To change operation methods, change values in the [S4+5] area before the second execution of EZPID instruction.

(Note 2) To execute auto-tuning, configure the setting in the area of control data [S1].

■ Setting of parameters [S1] to [S3]

Operand	Parameter name	Setting range	Setting method	
[S1]	Control data	Bit 0	Bit 1 = 0 (ON):	Auto-tuning request is issued. After auto-tuning is completed, this is reset when the EZPID instruction is executed. Reset this bit when auto-tuning is canceled.
			Bit 0 = 0 (OFF):	PID operation is performed.
		Bit 1	When auto-tuning is completed correctly, bit 1 is set to 1 (ON). Bit 1 is reset to 0 (OFF) at the start of instruction execution.	
		Bit 2	Specify whether the [S4] manipulated value (MV) should be cleared when OFF is larger than ON in the execution condition for the instruction.	
			Bit 2 = 0 (OFF):	Clear the manipulated value (MV) at the time of execution of the previous instruction.
			Bit 2 = 1 (ON):	Retain the manipulated value (MV) at the time of execution of the previous instruction.
		Bit 3	Select a method for outputting the result of PID operation.	
			Bit 3 = 0 (OFF):	PWM output
			Bit 3 = 1 (ON):	Analog output
		Bit 4	Specify the range (maximum value and minimum value) for internal calculation of the manipulated value (MV).	
			Bit 4 = 0 (OFF):	Use +20% and -20% of difference between the output upper limit and the output lower limit
			Bit 4 = 1 (ON):	Use the output upper limit and the output lower limit
		Bit F to Bit 5	Bits 5 to F are reserved bits. Use them as 0.	

Operand	Parameter name	Setting range	Setting method
[S2]	Process value (PV)	K -30000 to K 30000	Input the current value for process control amount using an analog input unit, etc. It is also possible to directly specify the input data area WXn of the analog input unit.
[S3]	Set point value (SP)	K -30000 to K 30000	Set a target value for process control amount. Set the value using an instruction or an external device (e.g. display).
[S3+1]	Proportional gain (Kp)	U1 to U9999 (0.1 to 999.9)	Specify parameters used for PID operation. Each parameter is obtained by multiplying the set point value by 0.1. When auto-tuning is executed, the relevant data are stored upon its completion.
[S3+2]	Integral time (Ti)	U0 to U30000 (0 to 3000 s)	
[S3+3]	Derivative Time (Td)	U0 to U10000 (0 to 1000 s)	

(Note 1) It is recommended to allocate the [S1] area to a non-hold type operation memory area (WR) where a bit operation is available.

(Note 2) It is recommended to allocate the [S2] area to a non-hold type operation memory area.

(Note 3) It is recommended to allocate the [S3] through [S3+3] areas to a hold type operation memory area.

(Note 4) If the [S3+1] through [S3+3] areas are all "0" when the EZPID instruction is started up, operation is continued with the settings where proportional gain (Kp) is 1, integral time (Ti) is 0, and derivative time (Td) is 0.

■ Setting of parameters [S4] to [S4+29]

Operand	Parameter name	Default	Setting range	Setting method
[S4]	Manipulated value (MV)	K0	K -10000 to K 10000	The result of PID operation is stored. If PWM output is selected in bit 3 of [S1], calculation is carried out in the range 0 to 10000, as the duty ratio (0 to 100%) of PWM output. If analog output is selected in bit 3 of [S1], the value given by the following formula is stored. (Output upper limit - Output lower limit) x Internal calculated value / 10000 + Lower limit Use the user program to convert the stored value into the range of the analog output unit.
[S4+1]	Output lower limit	K0	K-10000 ≤ Lower limit < Upper limit ≤ K 10000	Specify the output value (MV) range. Values for the specified range are output.
[S4+2]	Output upper limit	K10000		
[S4+3]	100% output zone	U0 (0%)	U0 to U80 (0% to 80%)	Specify the level of process value (PV), as percentage to set point value, above which PID control should be initiated. Until the specified process value (PV) is reached, the manipulated value (MV) is retained at 100%. If the process value (PV) is smaller than the set point value (SP), this shortens time until reaching the set point value (SP). When the set point value is U80, the output level is retained at 100% until the process value (PV)

Operand	Parameter name	Default	Setting range	Setting method
				reaches 80% of the set point value (SP). Once the 80% is exceeded, PID control is initiated. If the set point value is K0 (default), PID control is executed from the beginning.
[S4+4]	Control interval (Ts)	U100 (1s)	U1 to U6000 (0.01 to 60.0 s)	Specify the interval to execute PID operation. The setting value × 0.01 is the actual control interval.
[S4+5]	Control mode	U0 (Derivative-first, reverse operation)	U0 to U3	U0: Derivative-first, reverse operation U1: Derivative-first, forward operation U2: Proportional-plus-derivative-first, reverse operation U3: Proportional-plus-derivative-first, forward operation
[S4+6]	Auto-tuning bias value	U0	U0	In order to control overheating during auto-tuning, specify difference between the set point value during auto-tuning and the actual set point value. (Note 3)
[S4+7]	Auto-tuning Proportional gain (Kp) Correction coefficient	U125 (125%)	U50 to U500 (50 to 500%)	By setting coefficients listed on the left and executing auto-tuning, parameter values obtained through auto-tuning are multiplied by the respective coefficients, and stored in the parameter setting areas [S3+1] through [S3+3]. Example) If U200 is set to [S4+7], the value given as 200% of the proportional gain Kp, obtained through auto-tuning, is set to [S3+1].
[S4+8]	Auto-tuning Integral time (Ti) Correction coefficient	U200 (200%)	U50 to U500 (50 to 500%)	
[S4+9]	Auto-tuning Derivative Time (Td) Correction coefficient	U100 (100%)	U50 to U500 (50 to 500%)	
[S4+10]	Auto-tuning progress status	U0	U0 to U5	This indicates the progress of auto-tuning. Starting with the default [0], a value between U1 and U5 is stored in accordance with the progress. After auto-tuning is completed, the value returns to U0.
[S4+11] to [S4+29]	Work area for PID operation	K0	-	This work area is used by the system program for operation. Ensure that this is not overwritten by your program.

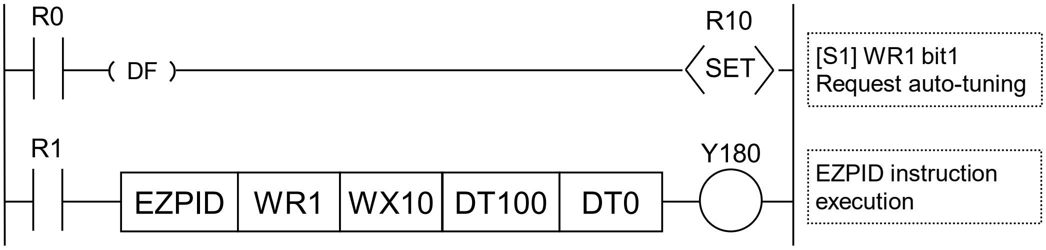
(Note 1) By default, the [S4] manipulated value (MV) is cleared when the EZPID instruction is started up. If it is necessary to retain the value from the previous execution, set 1 to bit 2 of the operand [S1] before execution of the EZPID instruction.

(Note 2) The areas [S4+1] through [S4+29] should be preset to default when the EZPID instruction is started up.

(Note 3) For details of the auto-tuning bias value, see ""Setting of auto-tuning bias value [S4+6]"".

■ Sample program (PWM output)

- In this sample, PWM output by PID operation is carried out after setting proportional gain (Kp), integral time (Ti), and derivative time (Td) through auto-tuning.



- Set the set point value (SP) to the operand [S3] area, using an instruction or a display.
- Using an instruction or a display, set bit 0 (auto-tuning request flag) of [S1]. Subsequently, switch the execution condition for the EZPID instruction to ON to initiate the auto-tuning operation.
- Once auto-tuning is completed correctly, parameters Kp, Ti and Td are set to the operand [S3+1] through [S3+3] areas. At this time, bit 0 (auto-tuning request flag) of the operand [S1] is reset to OFF, and bit 1 (auto-tuning completion flag) of the operand [S1] is switched ON.
- If the execution condition for the EZPID instruction is ON, PID control and PWM output are initiated in the following scan.
- The value (K0 to K10000) of the manipulated value (MV) of the operand [S4] is converted into a duty ratio from 0 to 100%, and PWM output is performed.
- The operand [S4] through [S4+29] areas are preset to default when the execution condition for the EZPID instruction is switched on. If the [S4] manipulated value (MV) should not be cleared, set 1 to bit 2 of [S1] before executing the EZPID instruction.
- If the execution condition of the EZPID instruction is switched OFF during PID control, PWM output Y180 is also turned OFF. At this time, the value of the manipulated value (MV) of operand [S4] is retained.

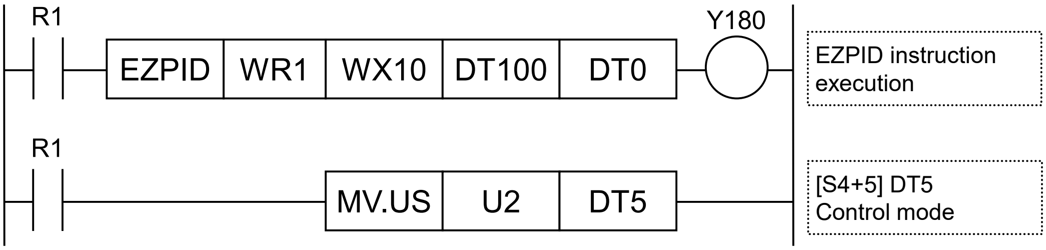
■ Interval and duty ratio of PWM output

- The duty of PWM is determined by the ratio of the manipulated value (MV), stored in [S4], to K0 through K10000.
- If the manipulated value (MV) is K0, PWM output is always OFF. If the MV is K10000, PWM output is always ON.
- The PWM output interval is determined by the control interval value specified by the operand [S4+4]. By default, the interval is set at 1 second.

■ Sample program (change of control conditions)

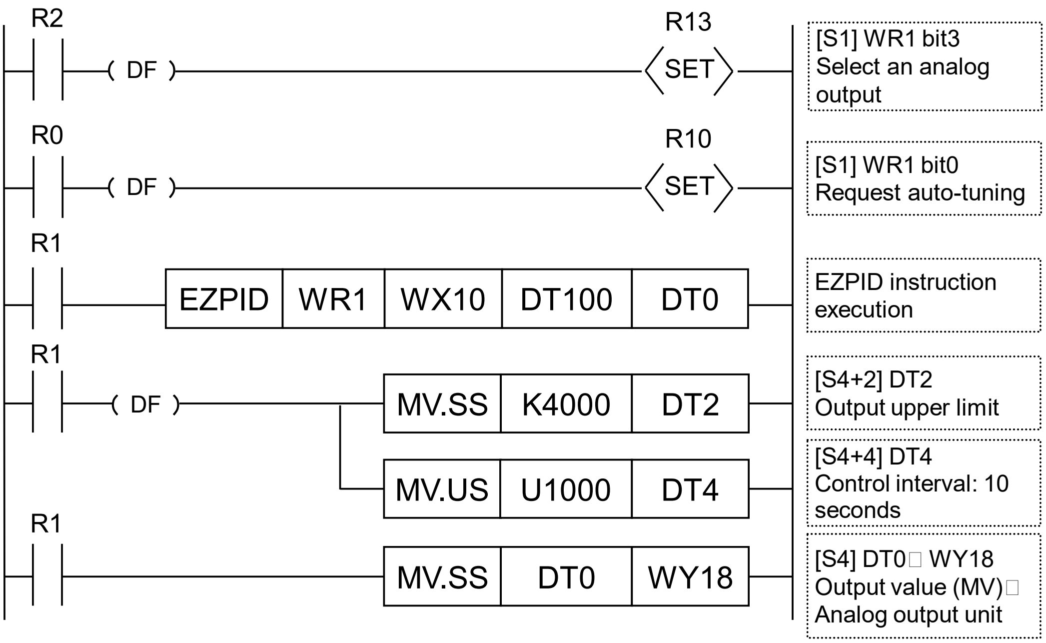
- By default, the following control conditions are used for operation: 1) Operation interval Ts: 1 second; 2) Control method: derivative-first, reverse operation (heating)
- In order to modify the control conditions, change values of the operands [S4+1] through [S4+9] using the MV instruction, etc.
- Carry out modifications between two executions of the EZPID instruction.

Example) Switch to proportional plus derivative control mode and carry out PWM output.



■ **Sample program (analog output)**

Example) In analog output, change the output upper limit [S4+2] to K4000, and the control interval [S4+4] to K1000 (10 seconds).



- In order to use analog output, set 1 to bit 3 of the operand [S1].
- The output lower limit [S4+1] and the output upper limit [S4+2] should be set according to the output range of the analog output unit.
- The value of the control interval (Ts): [S4+4] should be modified according to the input updating interval (normally 0.1 seconds or more) of the analog input unit.

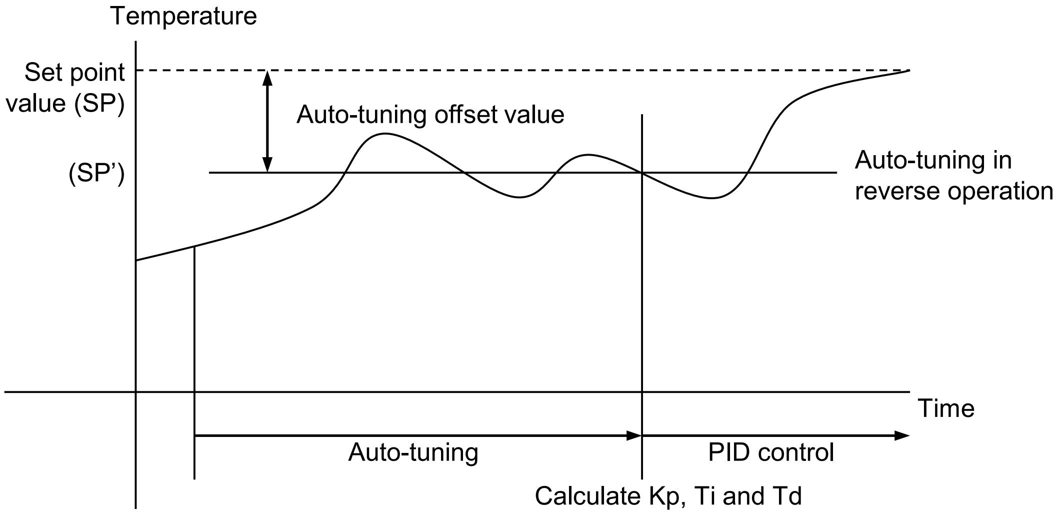
  Example) If the value of [S4+4] is K10, Ts is 100 ms.
- Modify other parameters such as the control mode as necessary.
- The manipulated value (MV) stored in the operand [S4] is transferred to the digital value output area WY that corresponds to the manipulated value of the analog output unit.
- In the manipulated value (MV) of [S4], the output internal calculated value (K0 to K10000) is converted by the following formula, and stored.

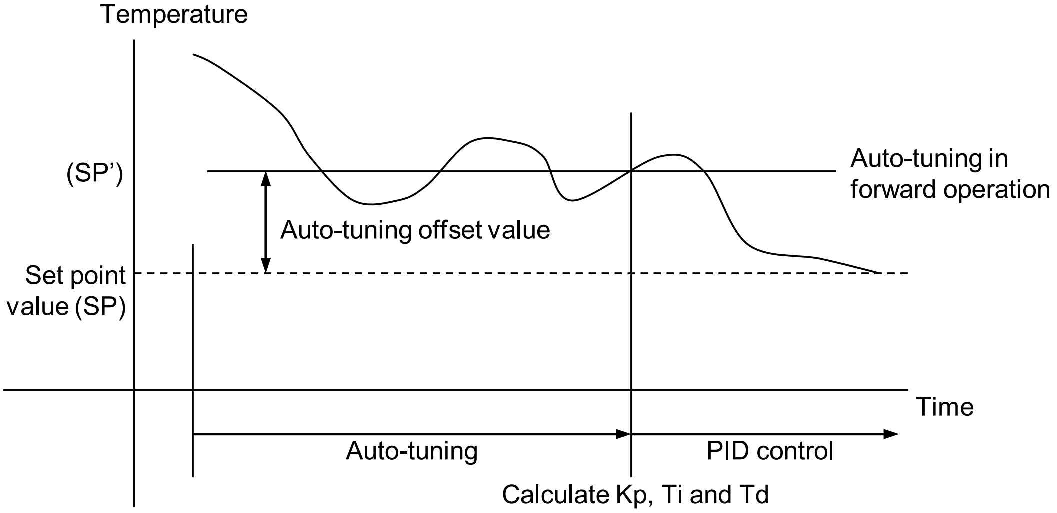
Conversion formula: (Output upper limit - Output lower limit) x (Internal calculated value) / 10000 + (Output lower limit)

- If [S4] is allocated to a hold type area, the manipulated value (MV) is retained even if the execution condition of the EZPID instruction is switched OFF.
- OUT is not necessary after EZPID instruction using analog output. Additionally, PWM is output OFF in this situation.

Setting of auto-tuning bias value [S4+6]

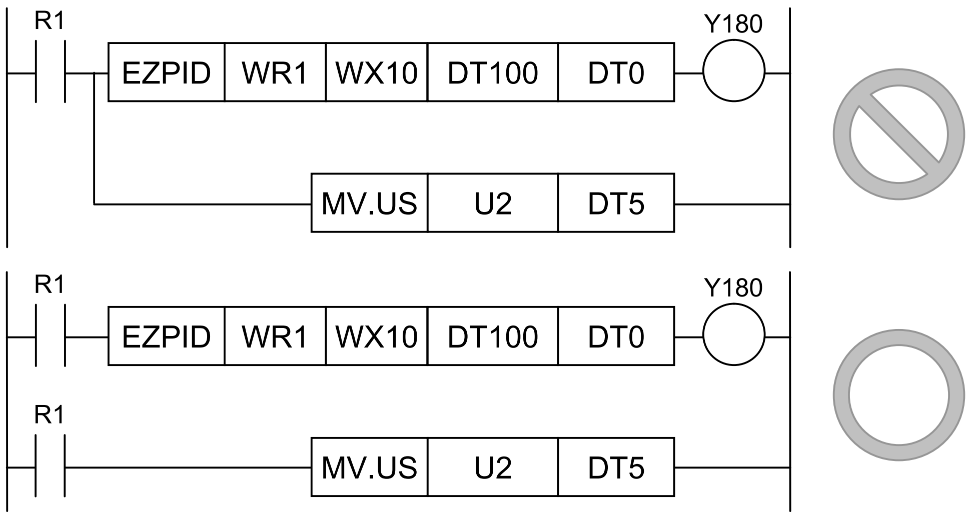
- If it is necessary to control output during auto-tuning (e.g. prevent overheating), set, as a bias value, the difference between the actual set point value (SP) during PID control operation and the set point value (SP') during auto-tuning.
- In the case of reverse operation (heating), (SP) is equal to or larger than (SP') and the difference thereof is given as the bias value.
- In the case of forward operation (cooling), (SP) is equal to or smaller than (SP') and the difference thereof is given as the bias value.
- Even if auto-tuning is started up, with the process value (PV) located near the set point value (SP), auto-tuning is executed using the set point value (SP') adjusted with the bias value.





■ Precautions for programming

- The operand [S4] through [S4+29] areas are initialized when the execution condition is switched on. If values other than default are to be used, set them using the MV instruction, etc.
- PID operation instructions are always internally calculating operation intervals, PWM output timings, etc. Ensure that the operation instructions are carried out only once in every scan. Additionally, do not attempt to execute it during a subroutine or interrupt program. At the same time, it is not possible to describe multiple EZPID instructions specifying the same operand.
- Do not switch off the execution conditions during PID operation. Otherwise, PID processing will be disabled.
- If you do not want to synchronize the PWM output cycle for controlling multiple objects, you can delay the startup timing, for example by adjusting the startup condition rise time.
- The system does not operate correctly if two or more PID instructions specifying the same table are described in the program. Even if execution conditions are not met, PID instructions operate internally using the specified table. In such cases, set the tables to separate addresses.
- Ensure that the work area for PID operation is not rewritten by other instructions.
- Error is not detected even if the parameter table exceeds the area. Ensure to specify a number at least 30 words before the final number of each device (such as the data register) when specifying the [S4] area. Also ensure that the area is not exceeded by index modification. Error is not detected even if the area is exceeded.
- Execution conditions vary immediately after the EZPID instruction, just like PWM output. Therefore, the system does not operate correctly if the subsequent instructions are described.



■ Precautions when executing auto-tuning

- Auto-tuning may not be executable depending on the process. In that case, the system returns to the original parameter operation.
- After auto-tuning is completed, the respective optimal values are stored for proportional gain [Kp], integral time [Ti], and derivative time [Td]. Before execution, it is necessary to specify appropriate values (e.g. lower limits) within the respective setting ranges.
- After auto-tuning is complete, store optimal values for proportional gain [Kp], integral time [Ti], and derivative time [Td]. Be careful that the stored values are not rewritten.
- During auto-tuning, the output values for Kp, Ti and Td are calculated by measuring changes to process values (PVs) when manipulated values (MVs) are set to the upper limits, as well as changes to process values (PVs) when manipulated values (MVs) are set to the lower limits, so that process values (PVs) will be increased or decreased in accordance with the respective set point values (SPs).
- Changes to manipulated values (MVs) of auto-tuning are completed in three sessions (upper limit output - lower limit output - upper limit output) in the fewest times possible. If the auto-tuning progress does not change from 0 after multiple sessions, shorten the control interval Ts and retry auto-tuning.

■ Operation actions

- [S4] through [S4+29] are initialized when the execution condition is switched on.
- If the parameters for proportional gain [Kp], integral time [Ti], and derivative time [Td] are all 0 at the start of PID operation, they are initialized to [Kp]=1, [Ti]=0 and [Td]=0 respectively, and operation is continued.
- When the auto-tuning request signal is switched on, bit 1 of [S1] (auto-tuning completion flag) and [S4+10] (auto-tuning completion code) are cleared.
- Set point values for auto-tuning are operated with the value of the set point value (SP) minus the bias value as the target value.
- After auto-tuning is correctly completed, values are stored for the calculated Kp, Ti and Td multiplied by the correction coefficients specified by [S4+7] through [S4+9].

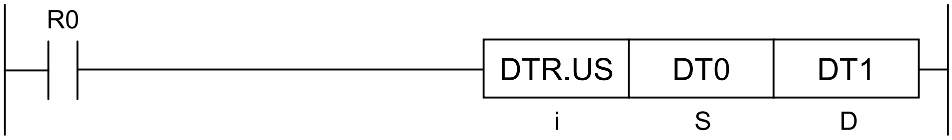
- After auto-tuning is correctly completed, bit 1 of [S1] (auto-tuning completion flag) and the auto-tuning completion code are stored in [S4+10]. If unsuccessful, the parameters for proportional gain [Kp], integral time [Ti], and derivative time [Td] are not updated.
- When bit 3 of [S1] is 0 (PWM output), output is carried out in the range from 0 to 10000, with the duty given by the conversion formula: (Upper limit - Lower limit) x Internal calculated value / 10000.
- When bit 3 of [S1] is 1 (analog output), the internal calculated value is output in the range from 0 to 10000, converted by the following formula, and set to the operand [S4] as the manipulated value (MV). Conversion formula: (Upper limit - Lower limit) x Internal calculated value / 10000 + Lower limit

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the following parameters are out of the setting range: [S2]: process value (PV); [S3]: set point value (SP); [S3]+1: KP; [S3]+2: TI; [S3]+3: TD; and [S4]+4 through [S4]+9.
	To be set when an area specified by [S3] or [S4] exceeds the upper limit of the specified operation device.
	To be set in the case of out-of-range in indirect access (index modification).

10.13 DTR (Data Revision Detection)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S	Device address to detect revision of a data value
D	Device address to store the data value from the previous execution

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●							●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

If data in the device address specified by [S] has been changed from the values in the previous execution, SR9 (CY) is switched to ON.

■ Processing

Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S]…DT0 [D]…DT1

		→		
DT0	H 0011		DT0	H 0011
DT1	**H 2233**		DT1	**H 0011**
CY	**0**		CY	**1**

Example 2) When the operation unit is 32-bit (UL, SL, SF)

[i]…UL,SL,SF [S]…DT0 [D]…DT2

		→		
DT0·DT1	H 11223344		DT0·DT1	H 11223344
DT2·DT3	H 11223344		DT2·DT3	**H 11223344**
CY	**1**		CY	**0**

■ Precautions for programming

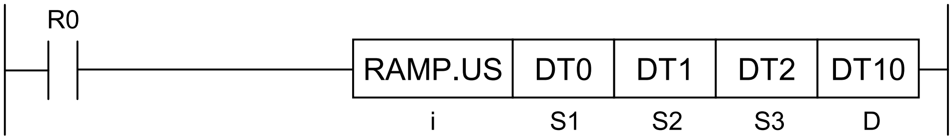
- Even when the operation unit is SF or DF, only data changes are checked. Type check for non-real numbers is not executed.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
SR9(CY)	To be set when the value of [S] differs from the value of [D]. To be reset when the value of [S] equals the value of [D].

10.14 RAMP (Ramp Output)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Device address where the default is stored, or constant
S2	Device address where the target value is stored, or constant
S3	Device address that stores the time width, or the constant (data available range: 1 to 30000)
D	Output storage device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
S2	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
S3	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●				●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

Outline of operation

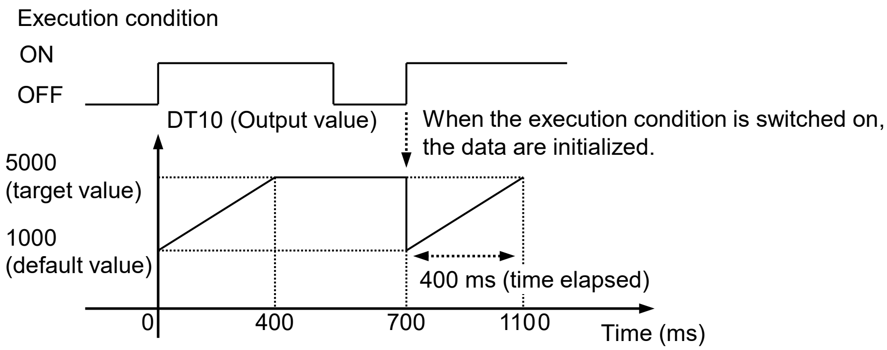
- Scaling is carried out from the output default value, output target value, and output time (in ms) specified by [S1], [S2], and [S3], and linear output is executed in accordance with the time elapsed from the execution start.

Processing

Example 1) Operation unit: 16 bits (US, SS)

[i]…US,SS

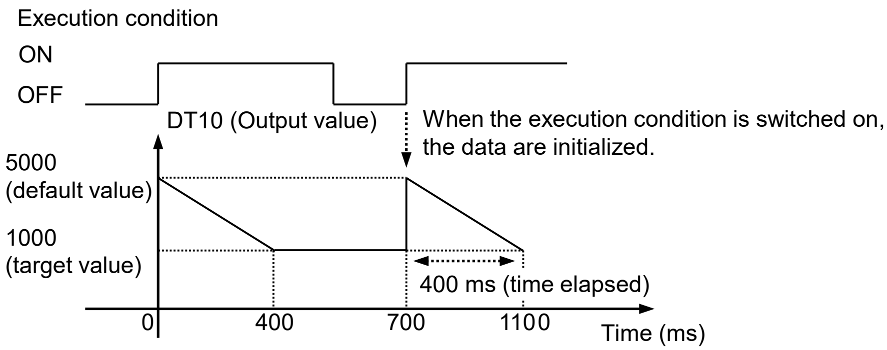
[S1]…DT0 : K1000 [S2]…DT1 : K5000 [S3]…DT2 : K400 [D]…DT10



Example 2) Operation unit: 32 bits (UL, SL)

[i]…UL,SL

[S1]…DT0 : K5000 [S2]…DT1 : K1000 [S3]…DT2 : K400 [D]…DT10

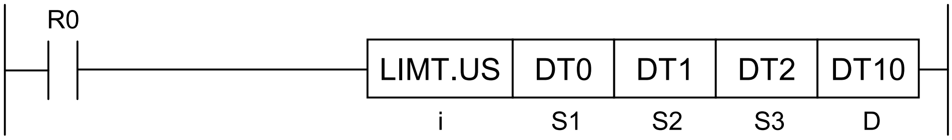


Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the output time width specified by [S3] is out of the accessible range.

10.15 LIMT (Upper and Lower Limit Control)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Device address where the lower limit is stored, or lower limit data
S2	Device address where the upper limit is stored, or upper limit data
S3	Device address where the input value is stored, or the input value data
D	Output storage device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
S2	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
S3	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●				●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- The output value, to be stored in the device address specified by [D], is controlled based on whether or not the input value specified by [S3] falls within the range bounded by the upper and lower limits set in [S1] and [S2].
- Output values are defined as follows:

Lower limit [S1] > Input value [S3]	Lower limit [S1] → Output value [D]
Upper limit [S2] < Input value [S3]	Upper limit [S2] → Output value [D]
Lower limit [S1] ≤ Input value [S3] ≤ Upper limit [S2]	Input value [S3] → Output value [D]

- For control by the upper limit only, the minimum value for the relevant operation unit is specified as the lower limit [S1].
- For control by the lower limit only, the maximum value for the relevant operation unit is specified as the upper limit [S1].

	US	SS	UL	SL	SF
Min. value	0	**-32768**	0	**-2147483648**	**Negative infinite**
Max. value	65535	32767	4294967295	2147483647	Positive infinite

■ Processing

Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S1]…DT0 [S2]…DT1 [S3]…DT2 [D]…DT10

	Before		After
DT0	K 100		K 100
DT1	K 1000		K 1000
DT2	K 500		K 500
DT10	**K 0**	→	**K 500**

Example 2) Operation unit: 32 bits (UL, SL)

[i]…UL,SL [S1]…DT0 [S2]…DT2 [S3]…DT4 [D]…DT10

	Before		After
DT0·DT1	K 100		K 100
DT2·DT3	K 1000		K 1000
DT4·DT5	K 90		K 90
DT10·DT11	**K 0**	→	**K 100**

Example 3) Operation units: Single precision floating point real number (SF)

[i]…SF [S1]…DT0 [S2]…DT4 [S3]…DT8 [D]…DT50

	Before		After
DT0･DT1	SF 1.00E+02	DT0･DT1	SF 1.00E+02
DT4･DT5	SF 1.00E+03	DT4･DT5	SF 1.00E+03
DT8･DT9	SF 2.00E+03	DT8･DT9	SF 2.00E+03
DT50･DT51	**SF 0.00E+00** →	DT50･DT51	**SF 1.00E+03**

■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).
SR8 (ER)	To be set when [S1] is larger than [S2].

10.16 BAND (Deadband Control)

■ Ladder diagram

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i			●		●	●	●

■ List of operands

Operand	Description
S1	Device address where the lower limit is stored, or lower limit data
S2	Device address where the upper limit is stored, or upper limit data
S3	Device address where the input value is stored, or the input value data
D	Output storage device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U	H (Note 5)	SF (Note 6)	DF (Note 7)	" "	
S1	●	●	●	●			●	●				●	●	●	●		●	●	●		●
S2	●	●	●	●			●	●				●	●	●	●		●	●	●		●
S3	●	●	●	●			●	●				●	●	●	●		●	●	●		●
D	●	●	●	●			●	●				●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 6) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 7) Can be specified only when the operation unit is a double-precision floating point real number (DF).

Outline of operation

- The output value, to be stored in the device address specified by [D], is controlled based on whether or not the input value specified by [S3] falls within the range bounded by the upper and lower limits set in [S1] and [S2].
- The output value [D] stores the following data.

Lower limit [S1] > Input value [S3]	Input value [S3] - Lower limit [S1] → Output value [D]
Upper limit [S2] < Input value [S3]	Input value [S3] - Upper limit [S2] → Output value [D]
Lower limit [S1] ≤ Input value [S3] ≤ Upper limit [S2]	0 → Output value [D]

Processing

Example 1) Operation unit: 16 bits (SS)

[i]…SS [S1]…DT0 [S2]…DT1 [S3]…DT2 [D]…DT10

	Before		After
DT0	K 100	DT0	K 100
DT1	K 1000	DT1	K 1000
DT2	K 1100	DT2	K 500
DT10	**K 100** →	DT10	**K 0**

Example 2) Operation unit: 32 bits (SL)

[i]…SL [S1]…DT0 [S2]…DT2 [S3]…DT4 [D]…DT10

	Before		After
DT0･DT1	K 100	DT0･DT1	K 100
DT2･DT3	K 1000	DT2･DT3	K 1000
DT4･DT5	K 1100	DT4･DT5	K 90
DT10･DT11	**K 100** →	DT10･DT11	**K -10**

Example 3) Operation units: Single precision floating point real number (SF)

[i]…SF [S1]…DT0 [S2]…DT2 [S3]…DT4 [D]…DT10

	Before		After
DT0･DT1	SF 1.00E+02	DT0･DT1	SF 1.00E+02
DT2･DT3	SF 1.00E+03	DT2･DT3	SF 1.00E+03
DT4･DT5	SF 1.10E+03	DT4･DT5	SF 3.00E+03
DT10･DT11	**SF 1.00E+02** →	DT10･DT11	**SF 2.00E+03**

Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when [S1] is larger than [S2].

10.17 ZONE (Zone Control)

■ Ladder diagram

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Device address that stores the negative bias value at input, or bias value data
S2	Device address that stores the positive bias value at input, or bias value data
S3	Device address where the input value is stored, or the input value data
D	Output storage device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K (Note 4)	U (Note 5)	H (Note 6)	SF (Note 7)	DF (Note 8)	" "	
S1	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
S2	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
S3	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●				●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

(Note 4) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 5) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 6) Can be specified only when the operation unit is integer (US, SS, UL, SL).

(Note 7) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 8) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- The bias value specified by [S1] or [S2] is added to the input value specified by [S3]. The resulting value is stored in the device address specified by [D].
- Output values are defined as follows:

Input value [S3] < 0	Input value [S3] + Negative bias value [S1] → Output value [D]
Input value [S3] = 0	0 → Output value [D]
Input value [S3] > 0	Input value [S3] + Positive bias value [S2] → Output value [D]

■ Processing

Example 1) When the operation unit is 16-bit (US, SS)

[i]…US,SS [S1]…DT0 [S2]…DT1 [S3]…DT2 [D]…DT10

DT0	K -100		DT0	K -100
DT1	K 1000		DT1	K 1000
DT2	K 0		DT2	K 500
DT10	**K 0**	→	DT10	**K 1500**

Example 2) Operation unit: 32 bits (UL, SL)

[i]…UL,SL [S1]…DT0 [S2]…DT2 [S3]…DT4 [D]…DT10

DT0･DT1	K -100		DT0･DT1	K -100
DT2･DT3	K 1000		DT2･DT3	K 1000
DT4･DT5	K 1		DT4･DT5	K -300
DT10･DT11	**K 1001**	→	DT10･DT11	**K -400**

Example 3) Operation units: Single precision floating point real number (SF)

[i]…SF [S1]…DT0 [S2]…DT4 [S3]…DT8 [D]…DT50

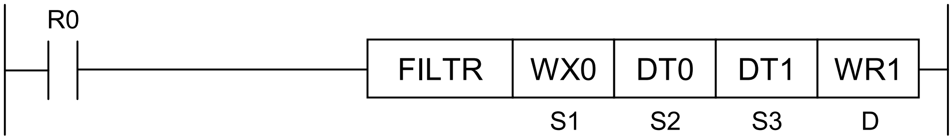
DT0･DT1	SF -1.00E+02		DT0･DT1	SF -1.00E+02
DT4･DT5	SF 1.00E+02		DT4･DT5	SF 1.00E+02
DT8･DT9	SF 1.00E+00		DT8･DT9	SF 0.00E+00
DT50･DT51	**SF 1.01E+02**	→	DT50･DT51	**SF 0.00E+00**

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

10.18 FILTR (Time Constant Processing)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Filtering targeted data (device address)
S2	Filtering targeted bit (device address or constant) (data available range: H0000 to HFFFF)
S3	Filtering time (device address or constant) (data available range: 0 to 30000, in ms)
D	Filtering result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●													●
S2	●	●	●	●			●	●								●	●				●
S3	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

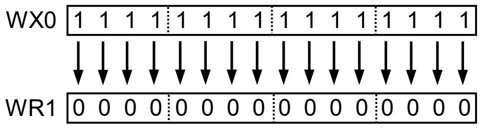
- Among data specified by [S1], bits specified by [S2] with the value 0 are directly output, and those with the value 1 are filtered and output.
- Filtering is carried out for the targeted bits within the time specified by [S3] (in 0 to 30000 ms). The result is output to the area specified by [D].
- When the execution condition is switched on, all input bits specified by [S1] are directly output without conditions.
- It is possible that a delay of up to one scan may be caused in the filtering time.

■ Processing

Among targeted data, specified bits with the value 0 are directly output, and those with the value 1 are filtered and output. Filtering is carried out within the specified time and the result is output.

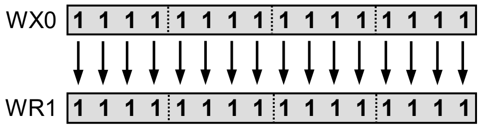
① Default conditions

Operand	Description	Device	Setting value
[S1]	Targeted data	WX0	H FFFF
[S2]	Targeted bit	DT0	H FF00
[S3]	Filtering time	DT1	K500
[D]	Processing result	WR1	H 0000



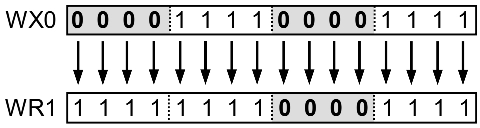
② Execution condition switched on (all input bits are directly output unconditionally)

Operand	Description	Device	Setting value
[S1]	Targeted data	WX0	H **FFFF**
[S2]	Targeted bit	DT0	H FF00
[S3]	Filtering time	DT1	K500
[D]	Processing result	WR1	H **FFFF**



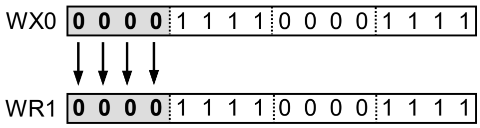
③ Filtering targeted data change (only untargeted bits are output)

Operand	Description	Device	Setting value
[S1]	Targeted data	WX0	H **0**F**0**F
[S2]	Targeted bit	DT0	H FF00
[S3]	Filtering time	DT1	K500
[D]	Processing result	WR1	H FF**0**F



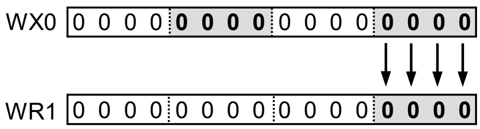
④ Filter processing time elapse (targeted bits are output)

Operand	Description	Device	Setting value
[S1]	Targeted data	WX0	H **0**F0F
[S2]	Targeted bit	DT0	H FF00
[S3]	Filtering time	DT1	K500
[D]	Processing result	WR1	H **0**F0F



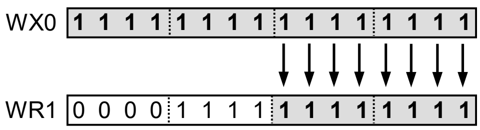
⑤ Filtering targeted data change (only untargeted bits are output)

Operand	Description	Device	Setting value
[S1]	Targeted data	WX0	H 0**0**0**0**
[S2]	Targeted bit	DT0	H FF00
[S3]	Filtering time	DT1	K500
[D]	Processing result	WR1	H 0F0**0**



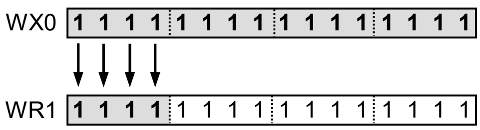
⑥ Filtering targeted data change before filter processing time elapse (only untargeted bits are output)

Operand	Description	Device	Setting value
[S1]	Targeted data	WX0	H **FFFF**
[S2]	Targeted bit	DT0	H FF00
[S3]	Filtering time	DT1	K500
[D]	Processing result	WR1	H 0F**FF**



⑦ Filter processing time elapse (targeted bits are output)

Operand	Description	Device	Setting value
[S1]	Targeted data	WX0	H **FFFF**
[S2]	Targeted bit	DT0	H FF00
[S3]	Filtering time	DT1	K500
[D]	Processing result	WR1	H **F**FFF



■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the filtering time [S3] is out of the range.

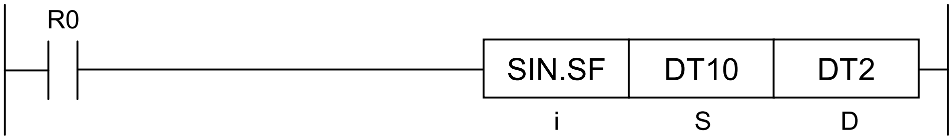
(MEMO)

11 High-level Instructions (Real Number)

Applicable Models: All Models

11.1 SIN (Sine Operation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Angle data (device address or constant) (units: radian)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], SIN for the angle data (units: radian) stored in [S] is calculated.
- The calculation result is stored in the area starting with [D].

  SIN([S]) → [D]

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF) (Designate 30° radian for [S])

[i]…SF
[S]…DT10 [D]…DT2

	Angle	Value (radians)
DT10・DT11	30°	**SF 5.235988E−01**
DT12・DT13	60°	SF 1.047198E+00
DT14・DT15	90°	SF 1.570796E+00

→

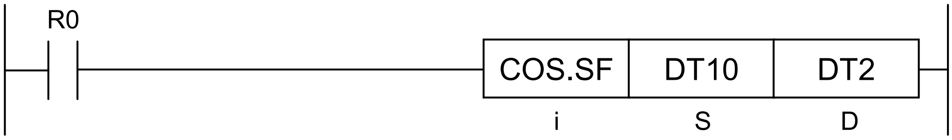
	Value
DT0・DT1	SF 0.000000E+00
DT2・DT3	**SF 5.000000E−01**
DT4・DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when a non-real number is specified for [S] (angle data).

11.2 COS (Cosine Operation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Angle data (device address or constant) (units: radian)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], COS for the angle data (units: radian) stored in [S] is calculated.
- The calculation result is stored in the area starting with [D].
  COS([S]) → (D)

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF) (Designate 30° radian for [S])

[i]…SF
[S]…DT10 [D]…DT2

	Angle	Value (radians)
DT10･DT11	30°	**SF 5.235988E−01**
DT12･DT13	60°	SF 1.047198E+00
DT14･DT15	90°	SF 1.570796E+00

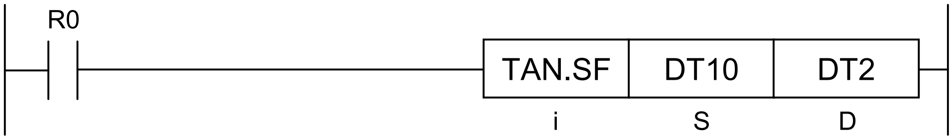
	Value
DT0･DT1	SF 0.000000E+00
DT2･DT3	**SF 8.660254E−01**
DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).
SR8 (ER)	To be set when a non-real number is specified for [S] (angle data).

11.3 TAN (Tangent Operation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Angle data (device address or constant) (units: radian)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], TAN for the angle data (units: radian) stored in [S] is calculated.
- The calculation result is stored in the area starting with [D].
  TAN([S]) → (D)

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF) (Designate 30° radian for [S])

[i]…SF
[S]…DT10 [D]…DT2

	Angle	Value (radians)
DT10･DT11	30°	**SF 5.235988E−01**
DT12･DT13	60°	SF 1.047198E+00
DT14･DT15	90°	SF 1.570796E+00

→

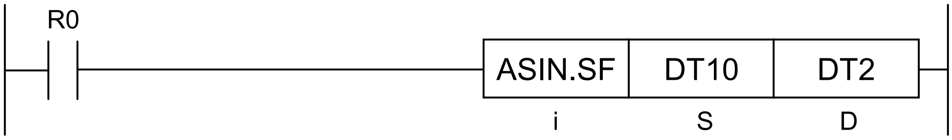
	Value
DT0･DT1	SF 0.000000E+00
DT2･DT3	**SF 5.773503E−01**
DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when a non-real number is specified for [S] (angle data).

11.4 ASIN (Arcsine Operation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Angle data (device address or constant) (SIN value) (data available range: -1.0 to +1.0)
D	Calculation result (device address) (units: radian)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], ASIN (arcsine) for the SIN value stored in [S] is calculated.
- The calculation result is stored in the area starting with [D].
  ASIN([S]) → (D)

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF) (Designate 15□SIN value for [S])

[i]…SF

[S]…DT10 [D]…DT2

	Angle	Value (radians)
DT10･DT11	15□	**SF2.588190E−01**
DT12･DT13	30□	SF1.047198E+00
DT14･DT15	45□	SF1.570796E+00

→

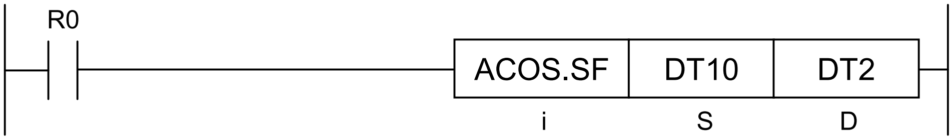
	Value
DT0･DT1	SF0.000000E+00
DT2･DT3	**SF2.617994E−01**
DT4･DT5	SF0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S] (angle data).
	To be set when [S] (angle data) is out of the accessible range.

11.5 ACOS (Arccosine Operation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Angle data (device address or constant) (COS value) (data available range: -1.0 to +1.0)
D	Calculation result (device address) (units: radian)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], ACOS (arccosine) for the COS value stored in [S] is calculated.

  ACOS([S]) → (D)

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF) (Designate 15° COS value for [S])

[i]…SF

[S]…DT10 [D]…DT2

	Angle	Value (radians)
DT10･DT11	15°	**SF9.659258E－01**
DT12･DT13	30°	SF1.047198E＋00
DT14･DT15	45°	SF1.570796E＋00

→

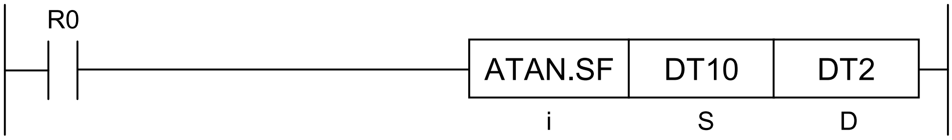
	Value
DT0･DT1	SF0.000000E＋00
DT2･DT3	**SF2.617994E－01**
DT4･DT5	SF0.000000E＋00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S] (angle data).
	To be set when [S] (angle data) is out of the accessible range.

11.6 ATAN (Arctangent Operation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Angle data (device address or constant) (TAN value)
D	Calculation result (device address) (units: radian)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], ATAN (arctangent) for the TAN value stored in [S] is calculated.
- The calculation result is stored in the area starting with [D].
  ATAN([S]) → (D)

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF) (Designate 15° TAN value for [S])

[i]…SF

[S]…DT10 [D]…DT2

	Angle	Value (radians)
DT10･DT11	15°	**SF 2.679392E−01**
DT12･DT13	30°	SF 1.047198E+00
DT14･DT15	45°	SF 1.570796E+00

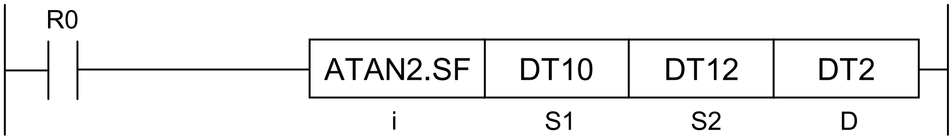
	Value
DT0･DT1	SF 0.000000E+00
DT2･DT3	**SF 2.617994E−01**
DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S] (angle data).

11.7 ATAN2 (Conversion: Coordinate Data → Angle Radian)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S1	Dividend of angle data (device address or constant) (Y coordinate)
S2	Angle data divisor (device address or constant) (X coordinate)
D	Calculation result (device address) (units: radian)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S1	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
S2	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).
(Note 2) Index register (I0 to IE)
(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).
(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], ATAN (units: radian) is calculated from the Y coordinate specified by [S1] and the X coordinate specified by [S2].
- The calculation result is stored in the area starting with [D].
  ATAN2([S1], [S2]) → [D]

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF) (Designate 1.0 for [S1] (Y coordinate) and [S2] (X coordinate))

[i]…SF

[S1]…DT10 [S2]…DT12 [D]…DT2

	Value
DT10･DT11	**SF 1.000000E＋00**
DT12･DT13	**SF 1.000000E＋00**
DT14･DT15	SF 0.000000E＋00

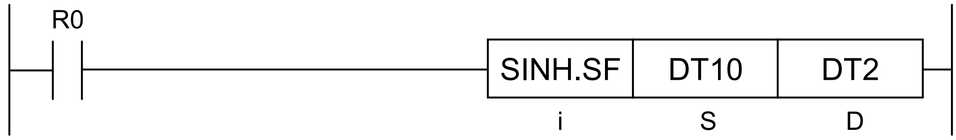
	Value
DT0･DT1	SF 0.000000E＋00
DT2･DT3	**SF 7.853982E－01**
DT4･DT5	SF 0.000000E＋00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S1] (Y coordinate) or [S2] (X coordinate).
	To be set when 0.0 is specified for [S1] (Y coordinate) and 0.0 for [S2] (X coordinate).

11.8 SINH (Hyperbolic Sine Operation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Angle data (device address or constant) (units: radian)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], SINH (hyperbolic sine) for the angle data (units: radian) stored in [S] is calculated.
- The calculation result is stored in the area starting with [D].

  SINH([S]) → [D]

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF) (Designate 30° radian for [S])

[i]…SF
[S]…DT10 [D]…DT2

	Angle	Value (radians)
DT10･DT11	30°	**SF 5.235988E−01**
DT12･DT13	60°	SF 1.047198E+00
DT14･DT15	90°	SF 1.570796E+00

→

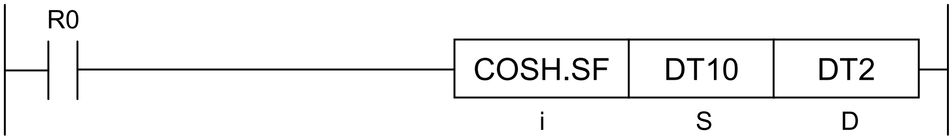
	Value
DT0･DT1	SF 0.000000E+00
DT2･DT3	**SF 5.478535E−01**
DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S] (angle data).

11.9 COSH (Hyperbolic Cosine Operation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Angle data (device address or constant) (units: radian)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], COSH (hyperbolic cosine) for the angle data (units: radian) stored in [S] is calculated.
- The calculation result is stored in the area starting with [D].

  COSH([S]) → [D]

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF) (Designate 30° radian for [S])

[i]...SF

[S]...DT10 [D]…DT2

	Angle	Value (radians)
DT10･DT11	30°	**SF 5.235988E−01**
DT12･DT13	60°	SF 1.047198E+00
DT14･DT15	90°	SF 1.570796E+00

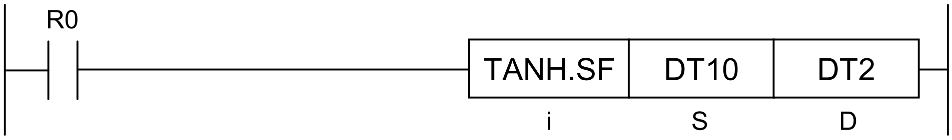
	Value
DT0･DT1	SF 0.000000E+00
DT2･DT3	**SF 1.140238E+00**
DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when a non-real number is specified for [S] (angle data).

11.10 TANH (Hyperbolic Tangent Operation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Angle data (device address or constant) (units: radian)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], TANH (hyperbolic tangent) for the angle data (units: radian) stored in [S] is calculated.
- The calculation result is stored in the area starting with [D].

  TANH([S]) → [D]

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF) (Designate 30° radian for [S])

[i]…SF

[S]…DT10 [D]…DT2

	Angle	Value (radians)
DT10･DT11	30°	**SF 5.235988E−01**
DT12･DT13	60°	SF 1.047198E+00
DT14･DT15	90°	SF 1.570796E+00

→

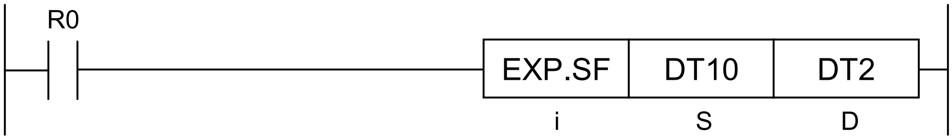
	Value
DT0･DT1	SF 0.000000E+00
DT2･DT3	**SF 4.804728E−01**
DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when a non-real number is specified for [S] (angle data).

11.11 EXP (Exponential Operation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Calculation target data (device address or constant) (real number value)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], the EXP (exponent) is calculated for the real number value stored in the area starting with [S].
- The calculation result is stored in the area starting with [D].
  EXP([S]) → [D]

Processing

Example) Operation unit: Single-precision, floating-point real number (SF)

[i]…SF

[S]…DT10 [D]…DT2

	Value (radians)
DT10･DT11	**SF 3.000000E+00**
DT12･DT13	SF 4.000000E+00
DT14･DT15	SF 5.000000E+00

→

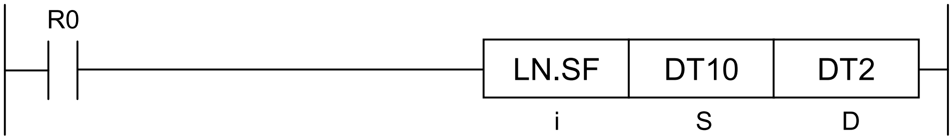
	Value
DT0･DT1	SF 0.000000E+00
DT2･DT3	**SF 2.008554E+01**
DT4･DT5	SF 0.000000E+00

Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S] (calculation target data).

11.12 LN (Natural Logarithmic Operation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Calculation target data (device address or constant) (real number value)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], LN (natural logarithm) for the real number value stored in the area starting with [S] is calculated.
- The calculation result is stored in the area starting with [D].
  LN([S]) → [D]

■ Processing

- LN (calculation target data) is calculated, and set for the calculation result.

Example) Operation unit: Single-precision, floating-point real number (SF)

[i]…SF
[S]…DT10 [D]…DT2

	Value (radians)
DT10･DT11	**SF 3.000000E+00**
DT12･DT13	SF 4.000000E+00
DT14･DT15	SF 5.000000E+00

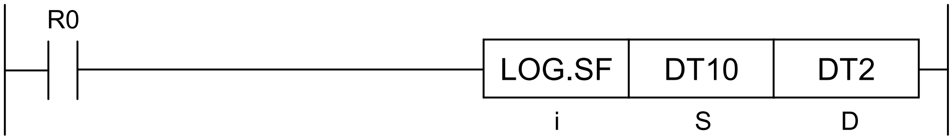
	Value
DT0･DT1	SF 0.000000E+00
DT2･DT3	**SF 1.098612E+00**
DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S] (calculation target data).
	To be set when a value that is 0.0 or less is specified for [S] (calculation target data).

11.13 LOG (Common Logarithmic Operation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Calculation target data (device address or constant) (real number value)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).
(Note 2) Index register (I0 to IE)
(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).
(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], LOG (common logarithm) for the real number value stored in the area starting with [S] is calculated.
- The calculation result is stored in the area starting with [D].
  LOG([S]) → [D]

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF)

[i]…SF

[S]…DT10 [D]…DT2

	Value (radians)
DT10･DT11	**SF 3.000000E+00**
DT12･DT13	SF 4.000000E+00
DT14･DT15	SF 5.000000E+00

→

	Value
DT0･DT1	SF 0.000000E+00
DT2･DT3	**SF 4.771213E−01**
DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S] (calculation target data).
	To be set when a value that is 0.0 or less is specified for [S] (calculation target data).

11.14 PWR (Power Operation)

■ Ladder diagram

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S1	Data to be powered (device address or constant) (real number value)
S2	Powering data (device address or constant) (real number value)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], the real number value stored in the area starting with [S1] is powered by the real number value stored in the area starting with [S2].
- The calculation result is stored in the area starting with [D].

  [S1] ^ [S2] → [D]

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF)

[i]…SF

[S1]…DT10 [S2]…DT12 [D]…DT2

	Value (radians)
DT10･DT11	**SF 3.000000E+00**
DT12･DT13	**SF 4.000000E+00**
DT14･DT15	SF 5.000000E+00

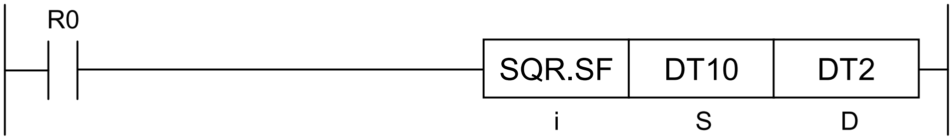
	Value
DT0･DT1	SF 0.000000E+00
DT2･DT3	**SF 8.100000E+01**
DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S1] (data to be powered) or [S2] (powering data).
	To be set when 0.0 is specified for [S1] (data to be powered) and a value that is 0.0 or less is specified for [S2] (powering data).
	To be set when a negative value is specified for [S1] (data to be powered) and a non-integer value is specified for [S2] (powering data).

11.15 SQR (Square Root Operation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Calculation target data (device address or constant) (real number value)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], square root for the real number value stored in the area starting with [S] is calculated.
- The calculation result is stored in the area starting with [D].
  SQR([S]) → [D]

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF)

[i]…SF
[S]…DT10　　[D]…DT2

	Value (radians)
DT10･DT11	**SF 3.000000E+00**
DT12･DT13	SF 4.000000E+00
DT14･DT15	SF 5.000000E+00

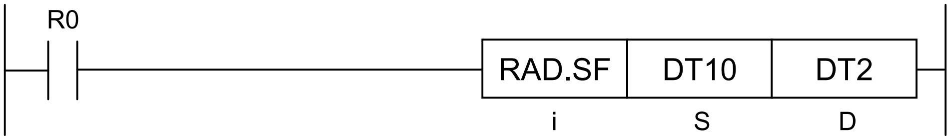
	Value
DT0･DT1	SF 0.000000E+00
DT2･DT3	**SF 1.732051E+00**
DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S] (calculation target data).
	To be set when a negative value is specified for [S] (calculation target data).

11.16 RAD (Conversion: Degrees → Radian)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Angle data (device address or constant) (units: degrees)
D	Angle data (device address) (units: radian)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], angle data (units: degrees), stored in the area starting with [S], are converted into angle data (units: radian).
- The calculation result is stored in the area starting with [D].

  [S] × (π / 180) → [D]

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF) (Designate 30 (degrees) for [S])

[i]…SF

[S]…DT10 [D]…DT2

	Value (degrees)
DT10･DT11	**SF 3.000000E+01**
DT12･DT13	SF 6.000000E+01
DT14･DT15	SF 9.000000E+01

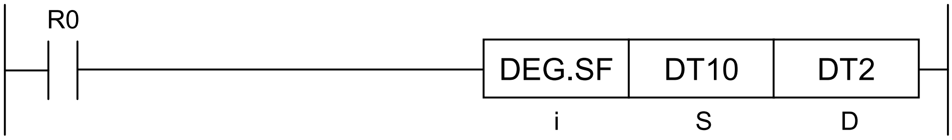
	Value (radians)
DT0･DT1	SF 0.000000E+00
DT2･DT3	**SF 5.235988E−01**
DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S] (angle data).

11.17 DEG (Conversion: Radian → Degrees)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Angle data (device address or constant) (units: radian)
D	Angle data (device address) (units: degrees)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], angle data (units: radian), stored in the area starting with [S], are converted into angle data (units: degrees).
- The calculation result is stored in the area starting with [D].
  [S] × (180 / π) → [D]

■ Processing

Example) Operation unit: Single-precision, floating-point real number (SF) (Designate 30° radian for [S])

[i]…SF

[S]…DT10 [D]…DT2

	Angle	Value (radians)
DT10･DT11	30°	**SF 5.235988E−01**
DT12･DT13	60°	SF 1.047198E+00
DT14･DT15	90°	SF 1.570796E+00

→

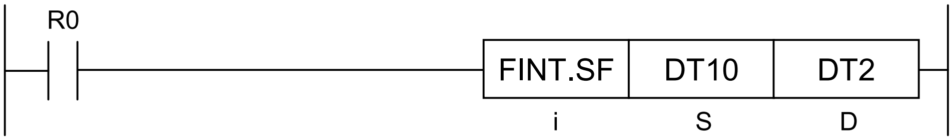
	Value (degrees)
DT0･DT1	SF 0.000000E+00
DT2･DT3	**SF 3.000000E+01**
DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when a non-real number is specified for [S] (angle data).

11.18 FINT (Floating Point Real Number Data - Rounding the First Decimal Point Down)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Calculation target data (device address or constant)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], the real number value stored in the area starting with [S] is processed and rounded the first decimal point down.
- The calculation result is stored in the area starting with [D].

■ Processing

Example 1) Operation unit: Single-precision, floating-point real number (SF) (positive real number)

[i]…SF
[S]…DT10 [D]…DT2

		→		
DT10･DT11	**SF 1.234560E+02**		DT0･DT1	SF 0.000000E+00
DT12･DT13	SF 3.456780E+02		DT2･DT3	**SF 1.230000E+02**
DT14･DT15	SF 5.678900E+02		DT4･DT5	SF 0.000000E+00

Example 2) Operation unit: Single-precision, floating-point real number (SF) (negative value)

[i]…SF
[S]…DT10 [D]…DT2

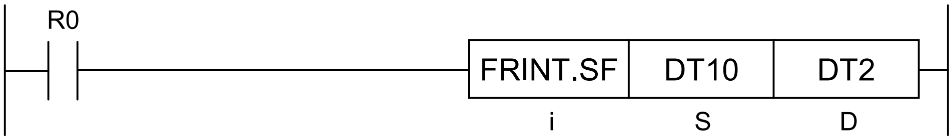
		→		
DT10･DT11	**SF -1.234560E+02**		DT0･DT1	SF 0.000000E+00
DT12･DT13	SF -3.456780E+02		DT2･DT3	**SF -1.240000E+02**
DT14･DT15	SF -5.678900E+02		DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S].

11.19 FRINT (Floating Point Real Number Data - Rounding the First Decimal Point Off)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Calculation target data (device address or constant)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], the real number value stored in the area starting with [S] is processed and rounded the first decimal point off.
- The calculation result is stored in the area starting with [D].

■ Processing

Example 1) Operation unit: Single-precision, floating-point real number (SF) (positive real number)

[i]…SF
[S]…DT10 [D]…DT2

DT10･DT11	**SF 1.234560E+02**	→	DT0･DT1	SF 0.000000E+00
DT12･DT13	SF 3.456780E+02		DT2･DT3	**SF 1.230000E+02**
DT14･DT15	SF 5.678900E+02		DT4･DT5	SF 0.000000E+00

Example 2) Operation unit: Single-precision, floating-point real number (SF) (negative value)

[i]…SF
[S]…DT10 [D]…DT2

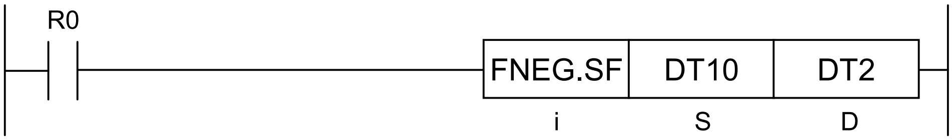
DT10･DT11	**SF -1.234560E+02**	→	DT0･DT1	SF 0.000000E+00
DT12･DT13	SF -3.456780E+02		DT2･DT3	**SF -1.230000E+02**
DT14･DT15	SF -5.678900E+02		DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S].

11.20 FNEG (Floating Point Real Number Data - Sign Changes (Negative/Positive Conversion))

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Calculation target data (device address or constant)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], the sign (negative or positive) of the real number value stored in the area starting with [S] is inverted.
- The calculation result is stored in the area starting with [D].

■ Processing

Example 1) Operation unit: Single-precision, floating-point real number (SF) (positive real number)

[i]…SF

[S]…DT10 [D]…DT2

DT10･DT11	**SF 1.234560E+02**	→	DT0･DT1	SF 0.000000E+00
DT12･DT13	SF 3.456780E+02		DT2･DT3	**SF -1.234560E+02**
DT14･DT15	SF 5.678900E+02		DT4･DT5	SF 0.000000E+00

Example 2) Operation unit: Single-precision, floating-point real number (SF) (negative value)

[i]…SF

[S]…DT10 [D]…DT2

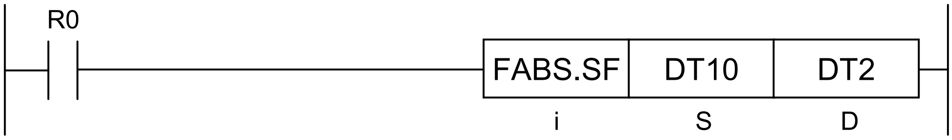
DT10･DT11	**SF -1.234560E+02**	→	DT0･DT1	SF 0.000000E+00
DT12･DT13	SF -3.456780E+02		DT2･DT3	**SF 1.234560E+02**
DT14･DT15	SF -5.678900E+02		DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S].

11.21 FABS (Floating Point Real Number Data - Absolute Value)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Calculation target data (device address or constant)
D	Calculation result (device address)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- According to the operation unit [i], the absolute value of the real number value stored in the area starting with [S] is calculated.
- The calculation result is stored in the area starting with [D].

■ Processing

Example 1) Operation unit: Single-precision, floating-point real number (SF) (positive real number)
[i]…SF
[S]…DT10 [D]…DT2

DT10･DT11	**SF 1.234560E+02**	→	DT0･DT1	SF 0.000000E+00
DT12･DT13	SF 3.456780E+02		DT2･DT3	**SF 1.234560E+02**
DT14･DT15	SF 5.678900E+02		DT4･DT5	SF 0.000000E+00

Example 2) Operation unit: Single-precision, floating-point real number (SF) (negative value)
[i]…SF
[S]…DT10 [D]…DT2

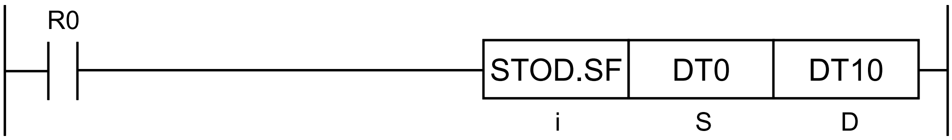
DT10･DT11	**SF -1.234560E+02**	→	DT0･DT1	SF 0.000000E+00
DT12･DT13	SF -3.456780E+02		DT2･DT3	**SF 1.234560E+02**
DT14･DT15	SF -5.678900E+02		DT4･DT5	SF 0.000000E+00

■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).
SR8 (ER)	To be set when a non-real number is specified for [S].

11.22 STOD (Conversion: Single-precision Real Number Data → Double-precision Real Number)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	

■ List of operands

Operand	Description
S	Starting address of the device where the target data to be converted is stored or the constant (data format: single-precision real number data)
D	Starting address of the device where conversion results are stored (data format: double-precision real number data)

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●			●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

■ Outline of operation

- This instruction converts the single-precision floating point real number stored in the area starting with [S] to a double-precision floating point real number.
- The conversion result is stored in the area starting with [D].

■ Example of processing

[S]…DT10 [D]…DT20

DT10･DT11	**-123.456789**
DT12･DT13	345.678000
DT14･DT15	567.890000

→

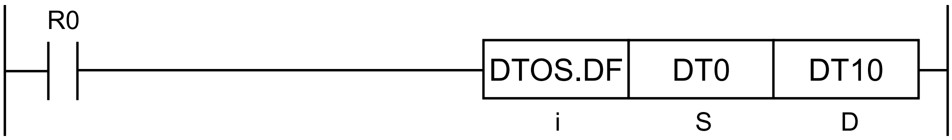
DT20･DT23	**-123.45678900000**
DT24･DT27	0.00000000000
DT28･DT31	0.00000000000

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when a non-real number is specified for [S].

11.23 DTOS (Conversion: Double-precision Real Number Data → Single-precision Real Number)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i							●

■ List of operands

Operand	Description
S	Starting address of the device where the target data to be converted is stored or the constant (data format: double-precision real number data)
D	Starting address of the device where conversion results are stored (data format: single-precision real number data)

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF (Note 3)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●					●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- This instruction converts the double-precision floating point real number stored in the area starting with [S] to a single-precision floating point real number.
- The conversion result is stored in the area starting with [D].

■ Example of processing

[S]…DT20 [D]…DT10

DT20･DT23	**-123.45678901234**	→	DT10･DT11	**-123.4568**
DT24･DT27	0.00000000000		DT12･DT13	345.678000
DT28･DT31	0.00000000000		DT14･DT15	567.890000

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when a non-real number is specified for [S].

11.24 DISF (Separation of Mantissa and Exponent of Single-precision or Double-precision Real Number Data)

■ Ladder diagram

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S	Starting address of the device where the target data to be separated is stored or the constant (data format: according to the operation unit)
D1	Starting address of the device where mantissa is stored (data format: according to the operation unit) Range (common to SF and DF): 0.5 or more, and less than 1.0 as absolute value)
D2	Starting address of the device where exponent is stored (data format: signed 32-bit integer)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF (Note 3)	DF (Note 4)	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D1	●	●	●	●			●	●	●		●	●	●	●							●
D2	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is a single-precision floating point real number (SF).

(Note 4) Can be specified only when the operation unit is a double-precision floating point real number (DF).

■ Outline of operation

- This instruction separates the floating point real number data stored in the area starting with [S] into mantissa and exponent according to the operation unit [i].
- The mantissa is stored in the area starting with [D1] and the exponent in the area starting with [D2].

■ Conversion example

Example 1) Operation units: Single-precision real number (SF) (positive real number)

[i]…SF
[S]…DT10 [D1]…DT2 [D2]…DT4

DT10・DT11	7.000000

* 7.0 = 0.875 x (the cube of 2)

DT0・DT1	0.000000	
DT2・DT3	0.875000	(Mantissa)
DT4・DT5	3	(Exponent)

Example 2) Operation units: Double-precision real number (DF) (positive real number)

[i]…DF
[S]…DT4 [D1]…DT20 [D2]…DT28

DT4・DT7	40.00000000000

* 40.0 = 0.6250 x (the sixth power of 2)

DT20・DT23	0.62500000000	(Mantissa)
DT24・DT27	0.00000000000	
DT28・DT29	6	(Exponent)

Example 3) Operation units: Single-precision real number (SF) (negative real number)

[i]…SF
[S]…DT10 [D1]…DT2 [D2]…DT4

DT10・DT11	-7.000000

* -7.0 = (-0.875) x (the cube of 2)

DT0・DT1	0.000000	
DT2・DT3	-0.875000	(Mantissa)
DT4・DT5	3	(Exponent)

Example 4) Operation units: Double-precision real number (DF) (negative real number)

[i]…DF
[S]…DT4 [D1]…DT20 [D2]…DT28

DT10・DT11	-40.00000000000

* -40.0 = (-0.6250) x (the sixth power of 2)

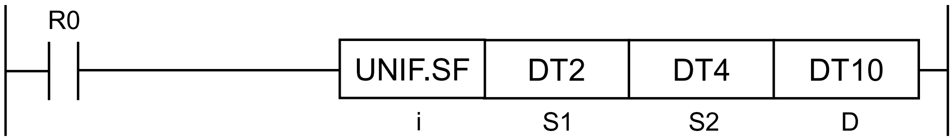
DT20・DT23	-0.62500000000	(Mantissa)
DT24・DT27	0.00000000000	
DT28・DT29	6	(Exponent)

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when a non-real number is specified for [S].

11.25 UNIF (Combining of Mantissa and Exponent, and Conversion of Single-precision or Double-precision Real Number)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i						●	●

■ List of operands

Operand	Description
S1	Starting address of the device where mantissa is stored (data format: according to the operation unit) Range (common to SF and DF): 0.5 or more, and less than 1.0 as absolute value)
S2	Starting address of the device where exponent is stored (data format: signed 32-bit integer) Range (for SF): -126 to 127 Range (for DF): -1022 to 1023
D	Starting address of the device where combined data is stored (data format: according to the operation unit)

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●	●	●	●							●
S2	●	●	●	●			●	●	●	●	●	●	●	●							●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 2) Index register (I0 to IE)

■ Outline of operation

- This instruction combines the data of mantissa and exponent stored in the areas starting with [S1] and [S2] according to the operation unit [i].

- The conversion result is stored in the area starting with [D].

■ Conversion example

Example 1) Operation units: Single-precision real number (SF) (positive real number)

[S1]…DT2 [S2]…DT4 [D]…DT10

DT0･DT1 0.000000
(Mantissa) DT2･DT3 **0.875000**
(Exponent) DT4･DT5 **3**

DT10･DT11 **7.000000**

* 7.0 = 0.875 x (the cube of 2)

Example 2) Operation units: Double-precision real number (DF) (positive real number)

[S1]…DT20 [S2]…DT28 [D]…DT4

(Mantissa) DT20･DT23 **0.62500000000**
DT24･DT27 0.00000000000
(Exponent) DT28･DT29 **6**

DT4･DT7 **40.00000000000**

* 40.0 = 0.6250 x (the sixth power of 2)

Example 3) Operation units: Single-precision real number (SF) (negative real number)

[S1]…DT2 [S2]…DT4 [D]…DT10

DT0･DT1 0.000000
(Mantissa) DT2･DT3 **-0.875000**
(Exponent) DT4･DT5 **3**

DT10･DT11 **-7.000000**

* -7.0 = (-0.875) x (the cube of 2)

Example 4) Operation units: Double-precision real number (DF) (negative real number)

[S1]…DT20 [S2]…DT28 [D]…DT4

(Mantissa) DT20･DT23 **-0.62500000000**
DT24･DT27 0.00000000000
(Exponent) DT28･DT29 **6**

DT4･DT7 **-40.00000000000**

* -40.0 = (-0.6250) x (the sixth power of 2)

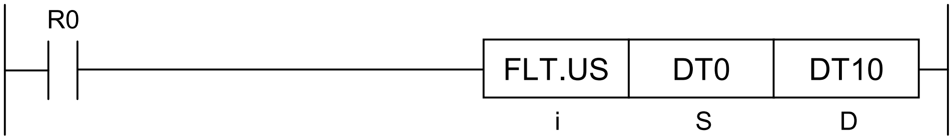
■ Flag operations

Name	Description
SR7	To be set in case of out-of-range values in indirect access (index modification).

Name	Description
SR8 (ER)	To be set when an out-of-range value is specified for [S1] (mantissa).
	To be set when an out-of-range value is specified for [S2] (exponent).

11.26 FLT (Conversion: Integer → Floating Point Real Number Data)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S	Conversion target data (device address or constant (data format: according to the operation unit))
D	Conversion result (device address (data format: single-precision floating point real number data)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K (Note 3)	U (Note 4)	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 4) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

- According to the operation unit [i], the integer value stored in the area starting with [S] is converted into a single-precision floating point real number value.
- The calculation result is stored in the area starting with [D].

■ Processing

Example 1) Unsigned 16 bits (US)

[i]…US
[S]…DT0 [D]…DT10

		→		
DT0	**U 123**		DT10･DT11	**SF 1.230000E+02**
DT1	U 456		DT12･DT13	SF 0.000000E+00
DT2	U 789		DT14･DT15	SF 0.000000E+00

Example 2) Signed 16 bits (SS) (positive value)

[i]…SS
[S]…DT20 [D]…DT10

		→		
DT20	**K 123**		DT10･DT11	**SF 1.230000E+02**
DT21	K 456		DT12･DT13	SF 0.000000E+00
DT22	K 789		DT14･DT15	SF 0.000000E+00

Example 3) Signed 16 bits (SS) (negative value)

[i]…SS
[S]…DT20 [D]…DT10

		→		
DT20	**K -123**		DT10･DT11	**SF -1.230000E+02**
DT21	K -456		DT12･DT13	SF 3.456780E+02
DT22	K -789		DT14･DT15	SF 5.678900E+02

Example 4) Unsigned 32 bits (UL)

[i]…UL
[S]…DT0 [D]…DT10

		→		
DT0･DT1	**U 12345**		DT10･DT11	**SF 1.234500E+04**
DT2･DT3	U 67890		DT12･DT13	SF 0.000000E+00
DT4･DT5	U 13579		DT14･DT15	SF 0.000000E+00

Example 5) Signed 32 bits (SL) (positive value)

[i]…SL
[S]…DT20 [D]…DT10

		→		
DT20･DT21	**K 12345**		DT10･DT11	**SF 1.234500E+04**
DT22･DT23	K 67890		DT12･DT13	SF 0.000000E+00
DT24･DT25	K 13579		DT14･DT15	SF 0.000000E+00

Example 6) Signed 32 bits (SL) (negative value)

[i]…SL
[S]…DT20 [D]…DT10

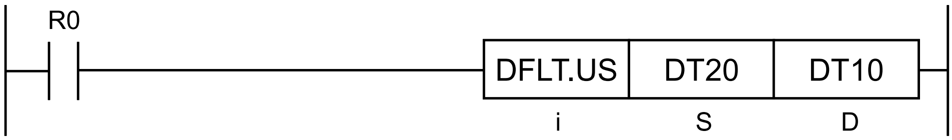
		→		
DT20･DT21	**K -12345**		DT10･DT11	**SF -1.234500E+04**
DT22･DT23	K -67890		DT12･DT13	SF 0.000000E+00
DT24･DT25	K -13579		DT14･DT15	SF 0.000000E+00

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

11.27 DFLT (Conversion: Integer → Double-precision Real Number Data)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S	Starting address of the device where the target data to be converted is stored or the constant (data format: according to the operation unit)
D	Starting address of the device where conversion results are stored (data format: double-precision real number data)

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 2)	K (Note 3)	U (Note 4)	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is a 16-bit integer (SS, US).

(Note 2) Index register (I0 to IE)

(Note 3) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 4) Can be specified only when the operation unit is an unsigned integer (US, UL).

■ Outline of operation

- This instruction converts the integer data stored in the area starting with [S] to a double-precision floating point real number.
- The conversion result is stored in the area starting with [D].

■ Conversion example

Example 1) Unsigned 16 bits (US)

[i]…US
[S] …DT0 [D] …DT10

DT0	**123**	→	DT10･DT13	**123.00000000000**
DT1	456		DT14･DT17	0.000000
DT2	789		DT18･DT21	0.000000

Example 2) Signed 16 bits (SS) (positive value)

[i]…SS
[S] …DT0 [D] …DT10

DT0	**123**	→	DT10･DT13	**123.00000000000**
DT1	456		DT14･DT17	0.000000
DT2	789		DT18･DT21	0.000000

Example 3) Signed 16 bits (SS) (negative value)

[i]…SS
[S] …DT0 [D] …DT10

DT0	**-123**	→	DT10･DT13	**-123.00000000000**
DT1	-456		DT14･DT17	0.000000
DT2	-789		DT18･DT21	0.000000

Example 4) Unsigned 32 bits (UL)

[i]…UL
[S] …DT0 [D] …DT10

DT0･DT1	**12345**	→	DT10･DT13	**12345.00000000000**
DT2･DT3	67890		DT14･DT17	0.000000
DT4･DT5	13579		DT18･DT21	0.000000

Example 5) Signed 32 bits (SL) (positive value)

[i]…SL
[S] …DT0 [D] …DT10

DT0･DT1	**12345**	→	DT10･DT13	**12345.00000000000**
DT2･DT3	67890		DT14･DT17	0.000000
DT4･DT5	13579		DT18･DT21	0.000000

Example 6) Signed 32 bits (SL) (negative value)

[i]…SL
[S] …DT0 [D] …DT10

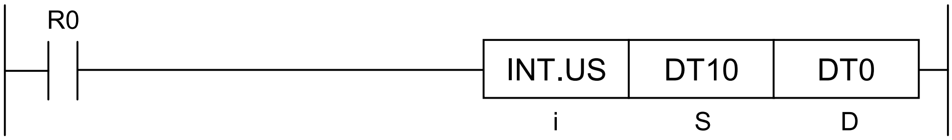
DT0･DT1	**-12345**	→	DT10･DT13	**-12345.00000000000**
DT2･DT3	-67890		DT14･DT17	0.000000
DT4･DT5	-13579		DT18･DT21	0.000000

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).

11.28 INT (Conversion: Floating Point Real Number Data → Integer (Round Down))

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S	Conversion target data (device address or constant (data format: according to the operation unit))
D	Conversion target data (device address or constant (data format: according to the operation unit))

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●			●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 3) Index register (I0 to IE)

■ Outline of operation

- According to the operation unit [i], the single-precision floating point real number value, which is stored in the area starting with [S], is converted into an integer value (the largest possible integer without exceeding the data).
- The calculation result is stored in the area starting with [D].
- The data setting ranges for conversion target data [S] by operation units are indicated below.

Operation unit	Minimum value	Maximum value
US	0.00	65535.99
SS	-32768.00	32767.99
UL	0.00	4294967295.99
SL	-2147483648.00	2147483647.99

■ Processing

Example 1) Unsigned 16 bits (US) (positive value)

[i]…US
[S]…DT10 [D]…DT0

DT10･DT11	**SF 2.345670E+02**	→	DT0	**U 234**
DT12･DT13	SF 3.456780E+02		DT1	U 0
DT14･DT15	SF 4.567890E+02		DT2	U 0

Example 2) Unsigned 16 bits (US) (negative value)

[i]…US
[S]…DT10 [D]…DT0

DT10･DT11	**SF -2.345670E+02**	→	DT0	U 0
DT12･DT13	SF -3.456780E+02		DT1	U 0
DT14･DT15	SF -4.567890E+02		DT2	U 0

* Operation error occurs if an unsigned integer is specified for the operation unit and a negative value is converted.

Example 3) Signed 16 bits (SS) (positive value)

[i]…SS
[S]…DT10 [D]…DT20

DT10･DT11	**SF 2.345670E+02**	→	DT20	**K 234**
DT12･DT13	SF 3.456780E+02		DT21	K 0
DT14･DT15	SF 4.567890E+02		DT22	K 0

Example 4) Signed 16 bits (SS) (negative value)

[i]…SS
[S]…DT10 [D]…DT20

DT10･DT11	**SF -2.345670E+02**	→	DT20	**K -235**
DT12･DT13	SF -3.456780E+02		DT21	K 0
DT14･DT15	SF -4.567890E+02		DT22	K 0

Example 5) Unsigned 32 bits (UL) (positive value)

[i]…UL
[S]…DT10 [D]…DT0

DT10･DT11	**SF 1.234567E+05**	→	DT0･DT1	**U 123456**
DT12･DT13	SF 2.468000E+02		DT2･DT3	K 0
DT14･DT15	SF 1.357000E+02		DT4･DT5	K 0

Example 6) Unsigned 32 bits (UL) (negative value)

[i]…UL
[S]…DT10 [D]…DT0

DT10･DT11	**SF -1.234567E+05**	→	DT0･DT1	U 0
DT12･DT13	SF -2.468000E+02		DT2･DT3	U 0
DT14･DT15	SF -1.357000E+02		DT4･DT5	U 0

* Operation error occurs if an unsigned integer is specified for the operation unit and a negative value is converted.

Example 7) Signed 32 bits (SL) (positive value)

[i]…SL
[S]…DT10 [D]…DT20

DT10･DT11	**SF 1.234567E+05**	→	DT20･DT21	**K 123456**
DT12･DT13	SF 2.468000E+02		DT22･DT23	K 0
DT14･DT15	SF 1.357000E+02		DT24･DT25	K 0

Example 8) Signed 32 bits (SL) (negative value)

[i]…SL
[S]…DT10 [D]…DT20

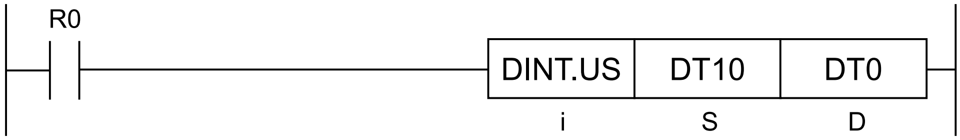
DT10･DT11	**SF -1.234567E+05**	→	DT20･DT21	**K -12346**
DT12･DT13	SF -2.468000E+02		DT22･DT23	K 0
DT14･DT15	SF -1.357000E+02		DT24･DT25	K 0

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S].
	To be set when an out-of-range value is specified for [S] (conversion target data).

11.29 DINT (Conversion: Double-precision Real Number Data → Integer (Round Down))

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S	Starting address of the device where the target data to be converted is stored or the constant (data format: double-precision real number data)
D	Starting address of the device where conversion results are stored (data format: according to the operation unit)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●					●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

- This instruction converts the double-precision floating point real number data stored in the area starting with [S] to an integer (the largest possible integer without exceeding the real number).
- The conversion result is stored in the area starting with [D].
- The data setting ranges for conversion target data [S] by operation units are indicated below.

Operation unit	Minimum value	Maximum value
US	0.00	65535.99
SS	-32768.00	32767.99
UL	0.00	4294967295.99
SL	-2147483648.00	2147483647.99

■ Example of processing

Example 1) Unsigned 16 bits (US) (positive value)

[i]…US
[S] …DT10 [D] …DT0

		→		
DT10･DT13	**234.56700000000**	→	DT0	**234**
DT14･DT17	345.67800000000		DT1	0
DT18･DT21	456.78900000000		DT2	0

Example 2) Unsigned 16 bits (US) (negative value)

Operation error occurs if an unsigned integer is specified for the operation unit and a negative value is converted.

[i]…US
[S] …DT10 [D] …DT0

		→		
DT10･DT13	**-234.56700000000**	→	DT0	0
DT14･DT17	345.67800000000		DT1	0
DT18･DT21	456.78900000000		DT2	0

Example 3) Signed 16 bits (SS) (positive value)

[i]…SS
[S] …DT10 [D] …DT0

		→		
DT10･DT13	**234.56700000000**	→	DT0	**234**
DT14･DT17	345.67800000000		DT1	0
DT18･DT21	456.78900000000		DT2	0

Example 4) Signed 16 bits (SS) (negative value)

[i]…SS
[S] …DT10 [D] …DT0

DT10・DT13	**-234.56700000000**	→	DT0	**-235**
DT14・DT17	345.67800000000		DT1	0
DT18・DT21	456.78900000000		DT2	0

Example 5) Unsigned 32 bits (UL) (positive value)

[i]…UL
[S] …DT10 [D] …DT0

DT10・DT13	**123456.78900000000**	→	DT0・DT1	**123456**
DT14・DT17	246.80000000000		DT2・DT3	0
DT18・DT21	135.70000000000		DT4・DT5	0

Example 6) Unsigned 32 bits (UL) (negative value)

Operation error occurs if an unsigned integer is specified for the operation unit and a negative value is converted.

[i]…UL
[S] …DT10 [D] …DT0

DT10・DT13	**-12345.78900000000**	→	DT0・DT1	0
DT14・DT17	-246.80000000000		DT2・DT3	0
DT18・DT21	-135.70000000000		DT4・DT5	0

Example 7) Signed 32 bits (SL) (positive value)

[i]…SL
[S] …DT0 [D] …DT20

DT0・DT3	**123456.78900000000**	→	DT20・DT21	**123456**
DT4・DT7	246.80000000000		DT22・DT23	0
DT8・DT11	135.70000000000		DT24・DT25	0

Example 8) Signed 32 bits (SL) (negative value)

[i]…SL
[S] …DT0 [D] …DT20

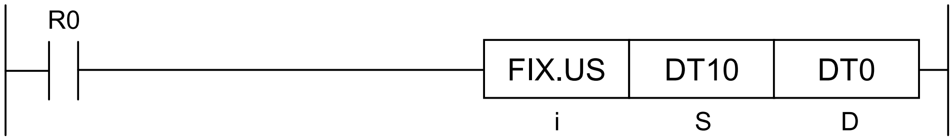
DT0･DT3	**-12345.78900000000**	→	DT20･DT21	**-12346**
DT4･DT7	-246.80000000000		DT22･DT23	0
DT8･DT11	-135.70000000000		DT24･DT25	0

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when a non-real number is specified for [S].
	To be set when an out-of-range value is specified for [S] (conversion target data).

11.30 FIX (Conversion: Floating Point Real Number Data → Integer (Round Decimal Digits))

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S	Conversion target data (device address or constant (data format: according to the operation unit))
D	Conversion target data (device address or constant (data format: according to the operation unit))

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●			●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, and 32-bit devices can be modified (real number constants cannot be specified).

(Note 3) Index register (I0 to IE)

■ Outline of operation

- According to the operation unit [i], the single-precision floating point real number value, which is stored in the area starting with [S], is converted to an integer value (rounding the first decimal point down).
- The calculation result is stored in the area starting with [D].
- The data setting ranges for conversion target data [S] by operation units are indicated below.

Operation unit	Minimum value	Maximum value
US	0.00	65535.99
SS	-32768.00	32767.99
UL	0.00	4294967295.99
SL	-2147483648.00	2147483647.99

■ Processing

Example 1) Unsigned 16 bits (US) (positive value)

[i]…US
[S]…DT10 [D]…DT0

DT10･DT11	**SF 2.345670E+02**	→	DT0	**U 234**
DT12･DT13	SF 3.456780E+02		DT1	U 0
DT14･DT15	SF 4.567890E+02		DT2	U 0

Example 2) Unsigned 16 bits (US) (negative value)

[i]…US
[S]…DT10 [D]…DT0

DT10･DT11	**SF -2.345670E+02**	→	DT0	U 0
DT12･DT13	SF -3.456780E+02		DT1	U 0
DT14･DT15	SF -4.567890E+02		DT2	U 0

* Operation error occurs if an unsigned integer is specified for the operation unit and a negative value is converted.

Example 3) Signed 16 bits (SS) (positive value)

[i]…SS
[S]…DT10 [D]…DT0

DT10･DT11	**SF 2.345670E+02**	→	DT0	**K 234**
DT12･DT13	SF 3.456780E+02		DT1	K0
DT14･DT15	SF 4.567890E+02		DT2	K0

Example 4) Signed 16 bits (SS) (negative value)

[i]…SS
[S]…DT10 [D]…DT0

DT10･DT11	**SF -2.345670E+02**	→	DT0	**K -234**
DT12･DT13	SF -3.456780E+02		DT1	K 0
DT14･DT15	SF -4.567890E+02		DT2	K 0

Example 5) Unsigned 32 bits (UL) (positive value)

[i]…UL
[S]…DT10 [D]…DT0

DT10･DT11	**SF 1.234567E+05**	→	DT0･DT1	**U 123456**
DT12･DT13	SF 2.468000E+02		DT2･DT3	U 0
DT14･DT15	SF 1.357000E+02		DT4･DT5	U 0

Example 6) Unsigned 32 bits (UL) (negative value)

[i]…UL
[S]…DT10 [D]…DT0

DT10·DT11	**SF -1.234567E+05**	→	DT0·DT1	U 0
DT12·DT13	SF -2.468000E+02		DT2·DT3	U 0
DT14·DT15	SF -1.357000E+02		DT4·DT5	U 0

* Operation error occurs if an unsigned integer is specified for the operation unit and a negative value is converted.

Example 7) Signed 32 bits (SL) (positive value)

[i]…SL
[S]…DT10 [D]…DT20

DT10·DT11	**SF 1.234567E+05**	→	DT20·DT21	**K 123456**
DT12·DT13	SF 2.468000E+02		DT22·DT23	K 0
DT14·DT15	SF 1.357000E+02		DT24·DT25	K 0

Example 8) Signed 32 bits (SL) (negative value)

[i]…SL
[S]…DT10 [D]…DT20

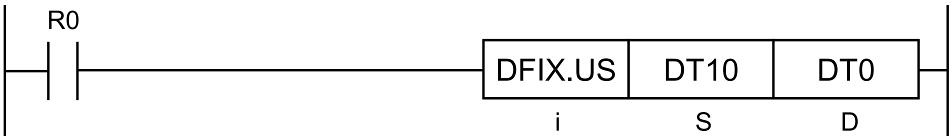
DT10·DT11	**SF -1.234567E+05**	→	DT20·DT21	**K -12345**
DT12·DT13	SF -2.468000E+02		DT22·DT23	K 0
DT14·DT15	SF -1.357000E+02		DT24·DT25	K 0

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S].
	To be set when an out-of-range value is specified for [S] (conversion target data).

11.31 DFIX (Conversion: Double-precision Real Number Data → Integer (Round Decimal Digits))

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S	Starting address of the device where the target data to be converted is stored or the constant (data format: double-precision real number data)
D	Starting address of the device where conversion results are stored (data format: according to the operation unit)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●					●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

- This instruction converts the double-precision floating point real number data stored in the area starting with [S] to an integer (rounding the first decimal point down).
- The conversion result is stored in the area starting with [D].
- The data setting ranges for conversion target data [S] by operation units are indicated below.

Operation unit	Minimum value	Maximum value
US	0.00	65535.99
SS	-32768.00	32767.99
UL	0.00	4294967295.99
SL	-2147483648.00	2147483647.99

■ Conversion example

Example 1) Unsigned 16 bits (US) (positive value)

[i]…US
[S] …DT10 [D] …DT0

	[S]			[D]
DT10･DT13	**234.56700000000**	→	DT0	**234**
DT14･DT17	345.67800000000		DT1	0
DT18･DT21	456.78900000000		DT2	0

Example 2) Unsigned 16 bits (US) (negative value)

Operation error occurs if an unsigned integer is specified for the operation unit and a negative value is converted.

[i]…US
[S] …DT10 [D] …DT0

	[S]			[D]
DT10･DT13	**-234.56700000000**	→	DT0	0
DT14･DT17	345.67800000000		DT1	0
DT18･DT21	456.78900000000		DT2	0

Example 3) Signed 16 bits (SS) (positive value)

[i]…SS
[S] …DT10 [D] …DT0

	[S]			[D]
DT10･DT13	**234.56700000000**	→	DT0	**234**
DT14･DT17	345.67800000000		DT1	0
DT18･DT21	456.78900000000		DT2	0

Example 4) Signed 16 bits (SS) (negative value)

[i]…SS
[S] …DT10 [D] …DT0

	[S]		[D]	
DT10･DT13	**-234.56700000000**	→	DT0	**-234**
DT14･DT17	345.67800000000		DT1	0
DT18･DT21	456.78900000000		DT2	0

Example 5) Unsigned 32 bits (UL) (positive value)

[i]…UL
[S] …DT10 [D] …DT0

	[S]		[D]	
DT10･DT13	**123456.78900000000**	→	DT0･DT1	**123456**
DT14･DT17	246.80000000000		DT2･DT3	0
DT18･DT21	135.70000000000		DT4･DT5	0

Example 6) Unsigned 32 bits (UL) (negative value)

Operation error occurs if an unsigned integer is specified for the operation unit and a negative value is converted.

[i]…UL
[S] …DT10 [D] …DT0

	[S]		[D]	
DT10･DT13	**-12345.78900000000**	→	DT0･DT1	0
DT14･DT17	-246.80000000000		DT2･DT3	0
DT18･DT21	-135.70000000000		DT4･DT5	0

Example 7) Signed 32 bits (SL) (positive value)

[i]…SL
[S] …DT0 [D] …DT20

	[S]		[D]	
DT0･DT3	**123456.78900000000**	→	DT20･DT21	**123456**
DT4･DT7	246.80000000000		DT22･DT23	0
DT8･DT11	135.70000000000		DT24･DT25	0

Example 8) Signed 32 bits (SL) (negative value)

[i]…SL
[S] …DT0 [D] …DT20

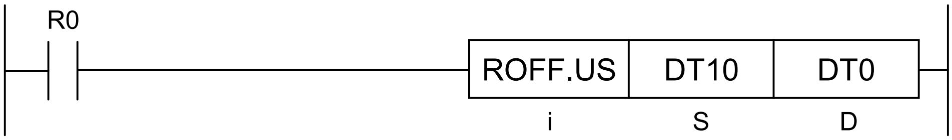
DT0･DT3	**-12345.78900000000**	→	DT20･DT21	**-12345**
DT4･DT7	-246.80000000000		DT22･DT23	0
DT8･DT11	-135.70000000000		DT24･DT25	0

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when a non-real number is specified for [S].
	To be set when an out-of-range value is specified for [S] (conversion target data).

11.32 ROFF (Conversion: Floating Point Real Number Data → Integer (Round to the Nearest Unit))

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S	Conversion target data (device address or constant (data format: according to the operation unit))
D	Conversion target data (device address or constant (data format: according to the operation unit))

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●			●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, and 32-bit devices can be modified (real number constants, and character constants cannot be specified).

(Note 3) Index register (I0 to IE)

■ Outline of operation

- According to the operation unit [i], the single-precision floating point real number value, which is stored in the area starting with [S], is converted to an integer value (rounding the first decimal point off).
- The calculation result is stored in the area starting with [D].
- The data setting ranges for conversion target data [S] by operation units are indicated below.
- The data setting ranges for conversion target data [S] by operation units are indicated below.

Operation unit	Minimum value	Maximum value
US	0.00	65535.49
SS	-32768.49	32767.49
UL	0.00	4294967295.49
SL	-2147483648.49	2147483647.49

■ Processing

Example 1) Unsigned 16 bits (US) (positive value)

[i]…US
[S]…DT10 [D]…DT0

DT10･DT11	**SF 2.345670E+02**	→	DT0	**U 235**
DT12･DT13	SF 3.456780E+02		DT1	U 0
DT14･DT15	SF 4.567890E+02		DT2	U 0

Example 2) Unsigned 16 bits (US) (negative value)

[i]…US
[S]…DT10 [D]…DT0

DT10･DT11	**SF -2.345670E+02**	→	DT0	U 0
DT12･DT13	SF -3.456780E+02		DT1	U 0
DT14･DT15	SF -4.567890E+02		DT2	U 0

* Operation error occurs if an unsigned integer is specified for the operation unit and a negative value is converted.

Example 3) Coded 16 bits (SS) (positive value)

[i]…SS
[S]…DT10 [D]…DT0

DT10･DT11	**SF 2.345670E+02**	→	DT0	**K 235**
DT12･DT13	SF 3.456780E+02		DT1	K 0
DT14･DT15	SF 4.567890E+02		DT2	K 0

Example 4) Coded 16 bits (SS) (negative value)

[i]…SS
[S]…DT10 [D]…DT0

DT10･DT11	**SF -2.345670E+02**	→	DT0	**K -235**
DT12･DT13	SF -3.456780E+02		DT1	K 0
DT14･DT15	SF -4.567890E+02		DT2	K 0

Example 5) Unsigned 32 bits (UL) (positive value)

[i]…UL
[S]…DT10 [D]…DT0

DT10･DT11	**SF 1.234567E+05**	→	DT0･DT1	**U 123457**
DT12･DT13	SF 2.468000E+02		DT2･DT3	U 0
DT14･DT15	SF 1.357000E+02		DT4･DT5	U 0

Example 6) Unsigned 32 bits (UL) (negative value)

[i]…UL
[S]…DT10 [D]…DT0

	[S]	→		[D]
DT10･DT11	**SF -1.234567E+05**	→	DT0･DT1	U 0
DT12･DT13	SF -2.468000E+02		DT2･DT3	U 0
DT14･DT15	SF -1.357000E+02		DT4･DT5	U 0

* Operation error occurs if an unsigned integer is specified for the operation unit and a negative value is converted.

Example 7) Coded 32 bits (SL) (positive value)

[i]…SL
[S]…DT10 [D]…DT20

	[S]	→		[D]
DT10･DT11	**SF 1.234567E+05**	→	DT20･DT21	**K 123457**
DT12･DT13	SF 2.468000E+02		DT22･DT23	K 0
DT14･DT15	SF 1.357000E+02		DT24･DT25	K 0

Example 8) Coded 32 bits (SL) (negative value)

[i]…SL
[S]…DT10 [D]…DT20

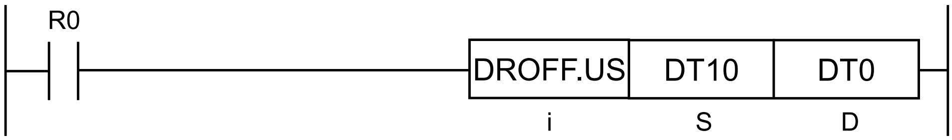
	[S]	→		[D]
DT10･DT11	**SF -1.234567E+05**	→	DT20･DT21	**K -123457**
DT12･DT13	SF -2.468000E+02		DT22･DT23	K 0
DT14･DT15	SF -1.357000E+02		DT24･DT25	K 0

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when a non-real number is specified for [S].
	To be set when an out-of-range value is specified for [S] (conversion target data).

11.33 DROFF (Conversion: Double-precision Real Number → Integer (Round to the Nearest Unit))

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S	Starting address of the device where the target data to be converted is stored or the constant (data format: double-precision real number data)
D	Starting address of the device where the target data to be converted is stored or the constant (data format: according to the operation unit)

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 2)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX (Note 3)	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●					●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

(Note 1) Cannot be specified when the operation unit is 16-bit integer (SS, US).

(Note 2) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 3) Index register (I0 to IE)

■ Outline of operation

- This instruction converts the double-precision floating point real number data stored in the area starting with [S] to an integer (rounding the first decimal point off).
- The conversion result is stored in the area starting with [D].
- The data setting ranges for conversion target data [S] by operation units are indicated below.

Operation unit	Minimum value	Maximum value
US	0.00	65535.49
SS	-32768.49	32767.49
UL	0.00	4294967295.49
SL	-2147483648.49	2147483647.49

■ Example of processing

Example 1) Unsigned 16 bits (US) (positive value)

[i]…US
[S] …DT10 [D] …DT0

DT10･DT13	**123.45600000000**	→	DT0	**123**
DT14･DT17	234.56700000000		DT1	0
DT18･DT21	345.67800000000		DT2	0

Example 2) Unsigned 16 bits (US) (negative value)

Operation error occurs if an unsigned integer is specified for the operation unit and a negative value is converted.

[i]…US
[S] …DT10 [D] …DT0

DT10･DT13	**-123.45600000000**	→	DT0	0
DT14･DT17	-234.56700000000		DT1	0
DT18･DT21	-345.67800000000		DT2	0

Example 3) Signed 16 bits (SS) (positive value)

[i]…SS
[S] …DT10 [D] …DT0

DT10･DT13	**123.45600000000**	→	DT0	**123**
DT14･DT17	234.56700000000		DT1	0
DT18･DT21	345.67800000000		DT2	0

Example 4) Signed 16 bits (SS) (negative value)

[i]…SS
[S] …DT10 [D] …DT0

DT10･DT13	**-123.45600000000**	→	DT0	**-123**
DT14･DT17	-234.56700000000		DT1	0
DT18･DT21	-345.67800000000		DT2	0

Example 5) Unsigned 32 bits (UL) (positive value)

[i]…UL
[S] …DT10 [D] …DT0

DT10･DT13	**123456.45670000000**	→	DT0･DT1	**123456**
DT14･DT17	246.80000000000		DT2･DT3	0
DT18･DT21	135.70000000000		DT4･DT5	0

Example 6) Unsigned 32 bits (UL) (negative value)

Operation error occurs if an unsigned integer is specified for the operation unit and a negative value is converted.

[i]…UL
[S] …DT10 [D] …DT0

DT10･DT13	**-12345.43200000000**	→	DT0･DT1	0
DT14･DT17	-246.80000000000		DT2･DT3	0
DT18･DT21	-135.70000000000		DT4･DT5	0

Example 7) Signed 32 bits (SL) (positive value)

[i]…SL
[S] …DT0 [D] …DT20

DT0･DT3	**123456.456700000000**	→	DT20･DT21	**123456**
DT4･DT7	246.80000000000		DT22･DT23	0
DT8･DT11	135.70000000000		DT24･DT25	0

Example 8) Signed 32 bits (SL) (negative value)

[i]…SL
[S] …DT0 [D] …DT20

DT0･DT3	**-12345.43200000000**	→	DT20･DT21	**-12345**
DT4･DT7	-246.80000000000		DT22･DT23	0
DT8･DT11	-135.70000000000		DT24･DT25	0

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when a non-real number is specified for [S].
	To be set when an out-of-range value is specified for [S] (conversion target data).

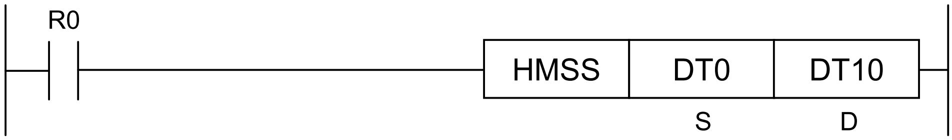
(MEMO)

12 High-level Instructions (Time)

Applicable Models: All Models

12.1 HMSS (Conversion: Time Data (Hours, Minutes and Seconds) → Seconds Data)

■ Ladder diagram



■ List of operands

Operand	Description
S	Starting device address of time data (available data range: 0 to 9999 (hours), 0 to 59 (minutes), 0 to 59 (seconds))
D	Device address of seconds data

(Note 1) For details of time data, refer to “"19.2 Clock and Time Data"”

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●										●
D	●	●	●	●			●	●	●		●										●

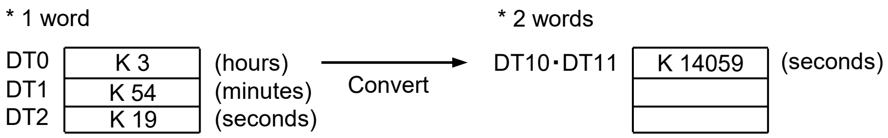
■ Outline of operation

- Time data comprised of 1 word for hours, 1 word for minutes and 1 word for seconds, which is stored in the area starting with [S], is converted into a 2-word integer data representing seconds.
- The calculation result is stored in the area starting with [D].

■ Processing

Example) Convert 3 hours, 54 minutes and 19 seconds

[S]…DT0 [D]…DT10

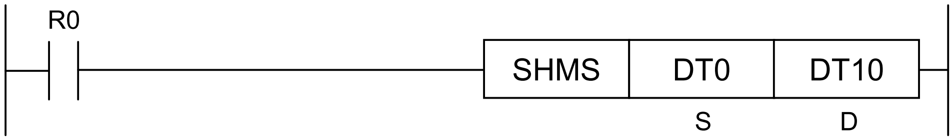


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the time data range is exceeded.

12.2 SHMS (Conversion: Seconds Data → Time Data (Hours, Minutes and Seconds))

■ Ladder diagram



■ List of operands

Operand	Description
S	Device address of seconds data (available data range: 0 to 35,999,999)
D	Starting device address of time data

(Note 1) For details of time data, refer to "19.2 Clock and Time Data"

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●										●
D	●	●	●	●			●	●	●		●										●

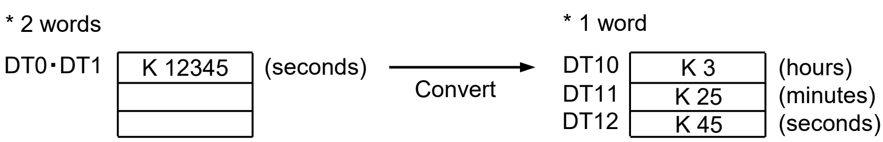
■ Outline of operation

- 2-word integer data representing seconds, which is stored in the area starting with [S], is converted into time data comprised of 1 word for hours, 1 word for minutes and 1 word for seconds.
- The calculation result is stored in the area starting with [D].

■ Processing

Example) Convert 12,345 seconds

[S]…DT0 [D]…DT10

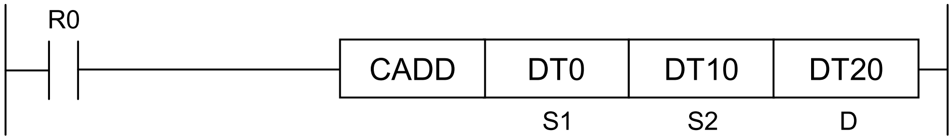


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the seconds data range is exceeded.

12.3 CADD (Clock Addition)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of clock data
S2	Starting device address of time data
D	Starting device address of addition result

(Note 1) For details of clock data and time data, refer to “"19.2 Clock and Time Data"”

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●													●
S2	●	●	●	●			●	●													●
D	●	●	●	●			●	●													●

■ Outline of operation

- Time data (hours, minutes and seconds), stored in the area starting with [S2], are added to the clock data (year, month, day, hours, minutes and seconds), stored in the area starting with [S1].
- The calculation result is stored in the area starting with [D].
- This operation takes leap years into account.

Processing

Example) Add 20 hours, 23 minutes and 45 seconds to 08:54:19, January 1, 2012

[S1]...DT0 [S2]...DT10 [D]...DT20

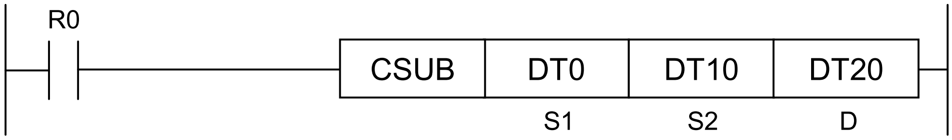
DT0	K 12	(year)						DT20	K 12	(year)
DT1	K 1	(month)						DT21	K 1	(month)
DT2	K 1	(day)	+				→	DT22	K 2	(day)
DT3	K 8	(hours)		DT10	K 20	(hours)		DT23	K 5	(hours)
DT4	K 54	(minutes)		DT11	K 23	(minutes)		DT24	K 18	(minutes)
DT5	K 19	(seconds)		DT12	K 45	(seconds)		DT25	K 4	(seconds)

Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the clock data range and/or the time data range is exceeded.
	To be set when the addition result is out of the accessible range.

12.4 CSUB (Clock Subtraction)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of clock data
S2	Starting device address of time data
D	Starting device address of subtraction result

(Note 1) For details of clock data and time data, refer to "19.2 Clock and Time Data"

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●													●
S2	●	●	●	●			●	●													●
D	●	●	●	●			●	●													●

■ Outline of operation

- Time data (hours, minutes and seconds), stored in the area starting with [S2], are subtracted from the clock data (year, month, day, hours, minutes and seconds), stored in the area starting with [S1].
- The calculation result is stored in the area starting with [D].
- This operation takes leap years into account.

■ Processing

Example) Subtract 20 hours, 23 minutes and 45 seconds from 08:54:19, January 1, 2012
[S1]…DT0 [S2]…DT10 [D]…DT20

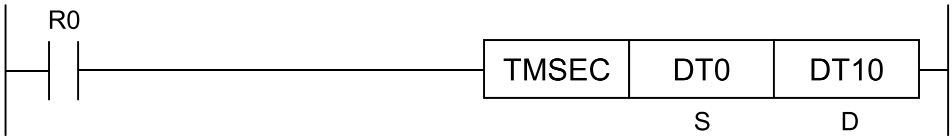
							→			
DT0	K 12	(year)						DT20	K 11	(year)
DT1	K 1	(month)						DT21	K 12	(month)
DT2	K 1	(day)	−					DT22	K 31	(day)
DT3	K 8	(hours)		DT10	K 20	(hours)		DT23	K 12	(hours)
DT4	K 54	(minutes)		DT11	K 23	(minutes)		DT24	K 30	(minutes)
DT5	K 19	(seconds)		DT12	K 45	(seconds)		DT25	K 34	(seconds)

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the clock data range and/or the time data range is exceeded.
	To be set when the subtraction result is out of the accessible range.

12.5 TMSEC (Calculation: Clock Data → Seconds Data from the Base Time)

■ Ladder diagram



■ List of operands

Operand	Description
S	Starting device address of clock data (available data range: 2000/1/1 00:00:00 to 2099/12/31 23:59:59)
D	Device address of seconds data from the base time

(Note 1) For details of clock data, refer to ""19.2 Clock and Time Data"".

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●										●
D	●	●	●	●			●	●	●		●										●

■ Outline of operation

- From the clock data (year, month, day, hours, minutes and seconds) stored in the area starting with [S], time elapsed from the base time is calculated.
- The calculation result is stored in the area starting with [D].
- The base time is 2001/1/1 00:00:00.

■ Processing

Example) Calculate seconds data against the base time, from 08:54:19, January 1, 2012

[S]…DT0   [D]…DT10

* 1 word

DT0	K 12	(year)
DT1	K 1	(month)
DT2	K 1	(day)
DT3	K 8	(hours)
DT4	K 54	(minutes)
DT5	K 19	(seconds)

→

* 2 words

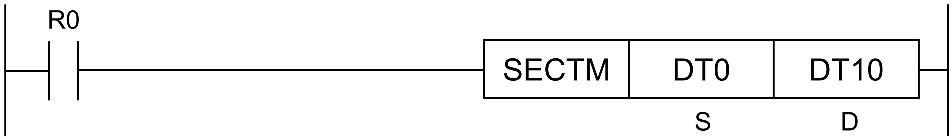
DT10・DT11	K 347100859	(Source seconds data)

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the clock data range is exceeded.

12.6 SECTM (Calculation: Seconds Data from the Base Time → Clock Data)

■ Ladder diagram



■ List of operands

Operand	Description
S	Device address of seconds data from the base time
D	Starting device address of clock data (available data range: 2000/1/1 00:00:00 to 2099/12/31 23:59:59)

(Note 1) For details of clock data, refer to “"19.2 Clock and Time Data"”

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●										●
D	●	●	●	●			●	●	●		●										●

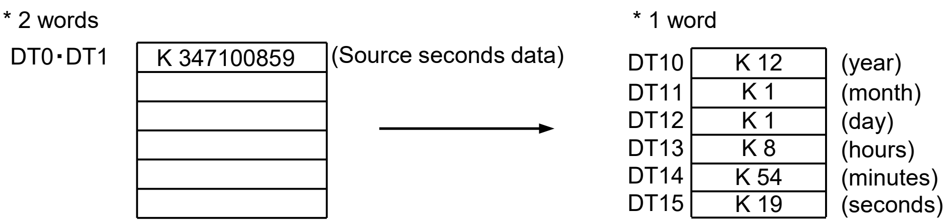
■ Outline of operation

- From the time elapsed from the base time, stored in the area starting with [S], clock data (year, month, day, hours, minutes and seconds) is calculated.
- The calculation result is stored in the area starting with [D].
- The base time is 2001/1/1 00:00:00.

■ Processing

Example) Calculate data from 1,325,408,059 seconds

[S]…DT0 [D]…DT10

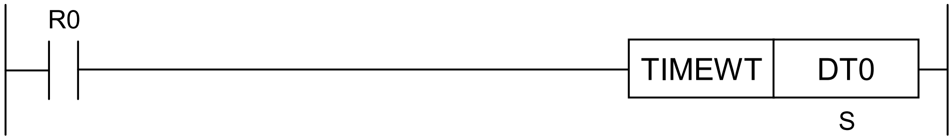


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the clock data range is exceeded.

12.7 TIMEWT (Setting of Clock/Calendar)

■ Ladder diagram



■ List of operands

Operand	Description
S	Starting device address of clock data

(Note 1) Only this instruction comprises 7 words in total, including day of week. For details of clock data, refer to "19.2 Clock and Time Data"

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●												●

■ Outline of operation

- The clock data (year, month, day, hours, minutes and seconds), stored in the area starting with [S], is set as RTC data for the CPU unit.
- The range of clock data that can be set for the calendar timer of the FP7 CPU unit is as follows:

  2000/1/1 00:00:00 to 2099/12/31 23:59:59

■ Processing

Example) Specify 08:54:19, January 1, 2012
[S]…DT0

* 1 word

DT0	K 12	(year)		K 12	(year)
DT1	K 1	(month)		K 1	(month)
DT2	K 1	(day)		K 1	(day)
DT3	K 8	(hours)	→ Update	K 8	(hours)
DT4	K 54	(minutes)		K 54	(minutes)
DT5	K 19	(seconds)		K 19	(seconds)
DT6	K 0	(day of week)		K 0	(day of week)

● Day-of-week data

0	Sun
1	Mon
2	Tue
3	Wed
4	Thu
5	Fri
6	Sat

■ Precautions for programming

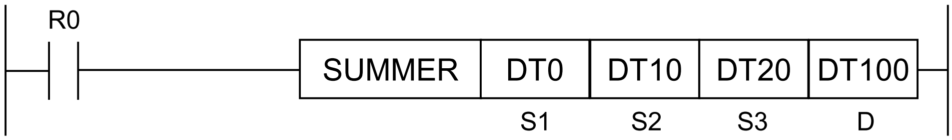
- Consistency of the day of week data with the date is not checked.

■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).
SR8	To be set when the clock data range is exceeded.
(ER)	To be set when the day of week range is exceeded.

12.8 SUMMER (Daylight Saving Time Acquisition)

■ Ladder diagram



■ List of operands

<table>
<tr><th>Operand</th><th>Description</th></tr>
<tr><td>S1</td><td>Starting address of the device that stores clock data when daylight saving time starts (Specify the clock data in the standard time.)
* The formats of the clock data of S1/S2 are the following four words.
<table>
<tr><th></th><th>Word</th></tr>
<tr><td>Sx + 0</td><td>Month</td></tr>
<tr><td>Sx + 1</td><td>Day</td></tr>
<tr><td>Sx + 2</td><td>Hours</td></tr>
<tr><td>Sx + 3</td><td>Minutes</td></tr>
</table></td></tr>
<tr><td>S2</td><td>Starting address of the device that stores clock data when daylight saving time ends (Specify the clock data in the daylight saving time.)
* The formats of the clock data of S1/S2 are the following four words.
<table>
<tr><th></th><th>Word</th></tr>
<tr><td>Sx + 0</td><td>Month</td></tr>
<tr><td>Sx + 1</td><td>Day</td></tr>
<tr><td>Sx + 2</td><td>Hours</td></tr>
<tr><td>Sx + 3</td><td>Minutes</td></tr>
</table></td></tr>
<tr><td>S3</td><td>Time difference within the daylight saving time period (Unit: minute) Settable range: 0 to 180 (minutes)</td></tr>
<tr><td>D</td><td>* The following eight words are the formats for the enable/disable + clock data of D.
<table>
<tr><th></th><th>Word</th></tr>
<tr><td>D + 0</td><td>Enable/Disable</td></tr>
<tr><td>D + 1</td><td>Year</td></tr>
<tr><td>D + 2</td><td>Month</td></tr>
<tr><td>D + 3</td><td>Day</td></tr>
<tr><td>D + 4</td><td>Hours</td></tr>
<tr><td>D + 5</td><td>Minutes</td></tr>
<tr><td>D + 6</td><td>Seconds</td></tr>
<tr><td>D + 7</td><td>Day of the week</td></tr>
</table></td></tr>
</table>

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●													●
S2	●	●	●	●			●	●													●
S3	●	●	●	●			●	●													●
D	●	●	●	●			●	●													●

■ Outline of operation

This instruction is used to acquire the time in daylight saving time.

■ Processing

- During the period specified by [S1] and [S2], clock data is corrected by [S3] (time difference), and the corrected clock data is set in the area that starts with [D].
- If the clock data is within the specified period, "1" is set in [D], the clock data is corrected, and the corrected clock data is set in the area from [D+1] to [D+7].
- If the clock data is outside of the specified period, "0" is set in [D], and the clock data is set in the area from [D+1] to [D+7] without being corrected.

■ Setting example

When the daylight saving time period is from 02:00 on March 8 to 01:59 on November 1, and 60 minutes is specified for the time difference

[S1]...DT0 [S2]...DT10 [S3]...DT20 [D]...DT100

		Description	Value
[S1] (Time to start daylight saving time)	DT0	Month	3
	DT1	Day	8
	DT2	Hours	2
	DT3	Minutes	0
[S2] (Time to end daylight saving time)	DT10	Month	11
	DT11	Day	1
	DT12	Hours	1
	DT13	Minutes	59
[S3] (Time difference: in minutes)	DT20	Time difference	60

Example 1) When the actual time is 01:23:45 on August 31

Because the time is within the specified period, the clock data is corrected and stored.

	Description	Value
SD50	Year	15
SD51	Month	8
SD52	Day	31
SD53	Hours	1
SD54	Minutes	23
SD55	Seconds	45
SD56	Day of the week	1

→

	Description	Value
DT100	flag	1 (Enabled)
DT101	Year	15
DT102	Month	8
DT103	Day	31
DT104	Hours	2
DT105	Minutes	23
DT106	Seconds	45
DT107	Day of the week	1

Example 2) When the actual time is 06:07:08 on November 20

Because the time is outside of the specified period, the data is not corrected.

	Description	Value
SD50	Year	15
SD51	Month	11
SD52	Day	20
SD53	Hours	6
SD54	Minutes	7

→

	Description	Value
DT100	flag	0 (Disabled)
DT101	Year	15
DT102	Month	11
DT103	Day	20

	Description	Value
SD55	Seconds	8
SD56	Day of the week	5

	Description	Value
DT104	Hours	6
DT105	Minutes	7
DT106	Seconds	8
DT107	Day of the week	5

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when [S1] or [S2] (clock data) is out of the range.
	To be set when [S3] (time difference) is out of the range.

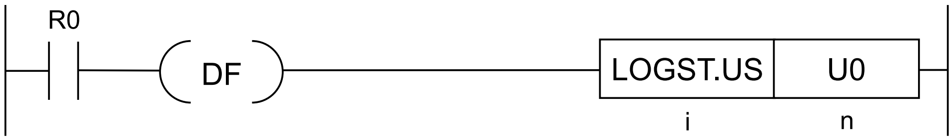
(MEMO)

13 High-level Instructions (Special)

Applicable Models: All Models

13.1 LOGST (Logging Trace Start Request)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
n	Number of logging trace requested to start

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 3)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K (Note 1)	U (Note 2)	H	SF	DF	" "	
n	●	●	●	●			●	●							●	●	●				●

(Note 1) Can be specified only when the operation unit is signed integer (SS).

(Note 2) Can be specified only when the operation unit is an unsigned integer (US).

(Note 3) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

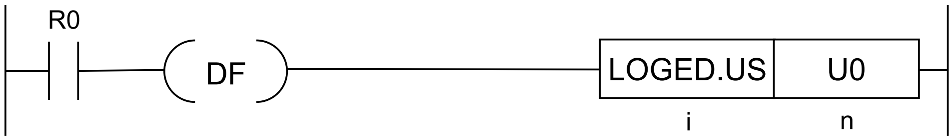
- This instruction requests to start the logging trace function for the number specified by [n].
- There is no problem even if the logging trace start request is made while the logging trace is being started or the logging trace start is being processed.
- The logging trace is set using the configuration menu of the tool software FPWIN GR7.
- For setting the logging trace to be automatically started, or starting it with the tool software FPWIN GR7, there is no need to execute the LOGST instruction.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when a request by a communication command has been accepted (logging trace start/ logging trace stop/logging trace registration).
	To be set when the logging trace stop has been requested.
	To be set in case of out-of-range values in indirect access (index modification).

13.2 LOGED (Logging Trace Stop Request)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
n	Number of logging trace requested to stop

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 3)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K (Note 1)	U (Note 2)	H	SF	DF	" "	
n	●	●	●	●			●	●							●	●	●				●

(Note 1) Can be specified only when the operation unit is signed integer (SS).

(Note 2) Can be specified only when the operation unit is an unsigned integer (US).

(Note 3) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

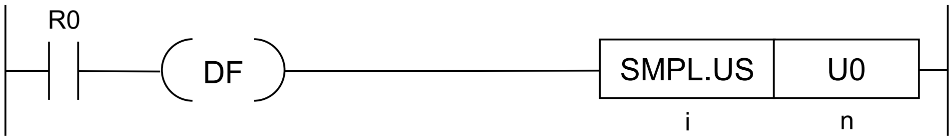
- This instruction requests to stop the logging trace function for the number specified by [n].
- There is no problem even if the logging trace stop request is made while the logging trace is being stopped or the logging trace stop is being processed.
- For stopping the logging trace with the tool software FPWIN GR7, there is no need to execute the LOGED instruction.
- In case of the trace function, when the trace condition set with the tool software FPWIN GR7 (memory buffer full or bit device ON) is not met, the trace operation will not be completed.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when a request by a communication command has been accepted (logging trace start/ logging trace stop/logging trace registration).
	To be set when the logging trace start has been requested.
	To be set in case of out-of-range values in indirect access (index modification).

13.3 SMPL (Sampling Trace)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
n	Logging trace number for which data is stored

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 3)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K (Note 1)	U (Note 2)	H	SF	DF	" "	
n	●	●	●	●			●	●							●	●	●				●

(Note 1) Can be specified only when the operation unit is signed integer (SS).

(Note 2) Can be specified only when the operation unit is an unsigned integer (US).

(Note 3) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

■ Outline of operation

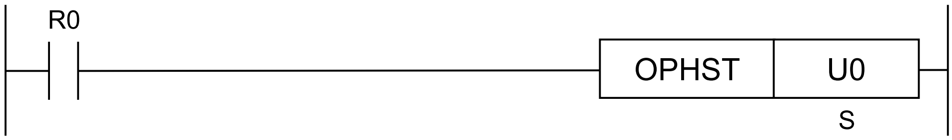
- This instruction performs data logging for the logging trace with the number specified by [n].
- The logging trace is set using the configuration menu of the tool software FPWIN GR7.
- The SMPL instruction is used for setting the trigger condition to start data logging with a user program. There is no need to execute the SMPL instruction when setting the trigger condition for starting logging to bit device, cycle or time with the tool software FPWIN GR7.
- To execute the trace for every scan, set the execution condition so that the trace is always executed.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the data logging condition of logging trace number is set to other than "Instruction".
	To be set in case of out-of-range values in indirect access (index modification).
	To be set when the logging trace stops.

13.4 OPHST (Operation History Start Request)

■ Ladder diagram



■ Available operation units

No operation unit.

■ List of operands

Operand	Description
S	Operation history group number for which startup is requested (0 to 7)

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●								●	●				●

■ Outline of operation

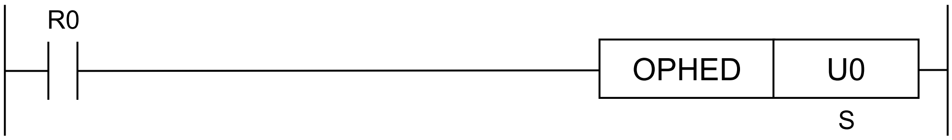
- Startup for operation histories is requested using the group number specified by [S].
- Even if startup is requested while operation histories are being started, no problem will occur. (No processing)

■ Flag operations

Name	Description
SR7	To be set when a startup request is issued to any group whose operation histories have not been registered.
SR8	To be set in case of out-of-range values in indirect access (index modification).
(ER)	To be set when an out-of-range value is specified for [S].

13.5 OPHED (Operation History End Request)

■ Ladder diagram



■ Available operation units

No operation unit.

■ List of operands

Operand	Description
S	Operation history group number for which stoppage is requested (0 to 7)

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●								●	●				●

■ Outline of operation

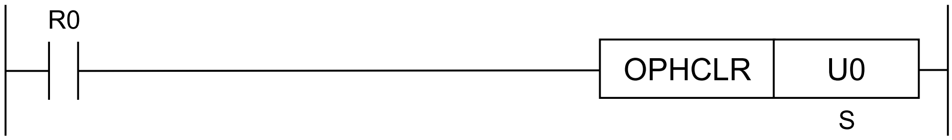
- Stoppage for operation histories is requested using the group number specified by [S].
- Even if stoppage is requested while operation histories are stopped, no problem will occur. (No processing)

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when a startup request is issued to any group whose operation histories have not been registered.
	To be set in case of out-of-range values in indirect access (index modification).
	To be set when an out-of-range value is specified for [S].

13.6 OPHCLR (Operation History Clearing)

■ Ladder diagram



■ Available operation units

No operation unit.

■ List of operands

Operand	Description
S	Operation history group number for which initialization is to be performed (0 to 7)

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●								●	●				●

■ Outline of operation

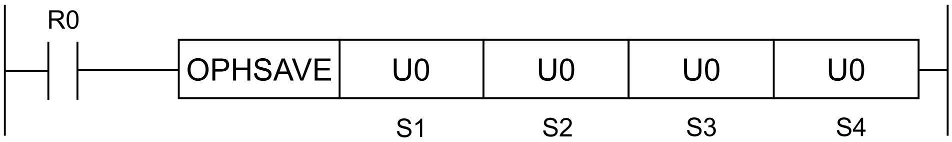
- Operation histories are initialized using the group number specified by [S].
- After initialization is completed, the number of write operations in operation histories becomes zero.
- If the power is turned off while operation histories are being initialized, initialization may not be performed correctly.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when a startup request is issued to any group whose operation histories have not been registered.
	To be set in case of out-of-range values in indirect access (index modification).
	To be set when an out-of-range value is specified for [S].

13.7 OPHSAVE (Operation History Save to SD Card)

■ Ladder diagram



■ Available operation units

No operation unit.

■ List of operands

Operand	Description
S1	Group number whose operation histories are output to SD card (0 to 7)
S2	Order in which operation histories are output to SD card (0: Ascending order of occurrence time, 1: Descending order of occurrence time)
S3	Language number for to SD card output (0 to 3) * You cannot specify any number greater than the number of each language supported in the operation history configuration.
S4	Operation after output to SD card (0: Nothing, 1: Operation histories clearing)

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●								●	●				●
S2	●	●	●	●			●	●								●	●				●
S3	●	●	●	●			●	●								●	●				●
S4	●	●	●	●			●	●								●	●				●

■ Outline of operation

- The operation histories of the group number specified by [S1] are output to the SD card in the order specified by [S2] (ascending or descending order of occurrence time).
- [S3] specifies the language number for which operation histories are to be output to SD card.
- After output, operation histories can be deleted according to the operation specified by [S4].
- The number of files to be output is set for each group as the number of managed generations (up to 128 files) in the operation history configuration. Any files exceeding the maximum limit will be deleted in chronological order (starting from the oldest file).

  Example of output files:

OPH0
├OpeHis0(180619_112345).csv
├OpeHis0(180619_113412).csv
└OpeHis0(180619_114630).csv
・・・
OPH7
├OpeHis7(180620_112345).csv
├・・・
└OpeHis7(180624_114630).csv
Output file directories are named for each group, as below.
\OPHx\ x: Group number
Output files are named as below.
OpeHisx(YYMMDD_HHMMSS).csv x: Group number
YYMMDD: File creation date (year/month/day)
HHMMSS: File creation time (hour/minute/second)

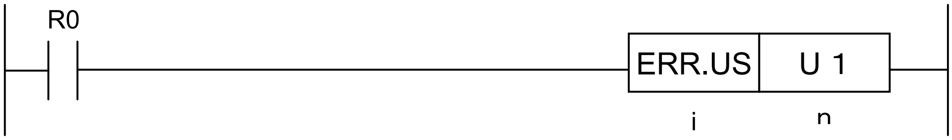
- The OPHSAVE instruction cannot be executed while the SD memory card access instruction is being executed.

■ Flag operations

Name	Description
SD memory card access instruction in progress (SR3A)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SD memory card access instruction execution completed (SR3B)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SD memory card access instruction execution result (SR3C)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7 SR8 (ER)	The flag is set when an out-of-range value is specified for S1.
	The flag is set when an out-of-range value is specified for S2.
	The flag is set when an out-of-range value is specified for S3.
	The flag is set when an out-of-range value is specified for S4.
	The flag is set when a group without registered operation histories is specified.
	To be set in case of out-of-range values in indirect access (index modification).

13.8 ERR (Self-Diagnostic Error Code Set)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●					

■ List of operands

Operand	Description
n	Specify a self-diagnostic error code.

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
n																●	●				

■ Outline of operation

- This instruction sets an error code with the user program.
- A self-diagnostic error code specified by [n] is stored in the self-diagnostic abnormality code register (SD0), and the self-diagnosis error occurrence flag (SR0) is set.
- It is also possible to describe multiple ERR instructions that set the same error code.

■ Setting of a self-diagnostic error code [n]

- n (self-diagnostic error code) can be set within the range from U1000 to U2999.

[n]	Operation when an error occurs
U 1000 to 1999	Operation stops
U 2000 to 2999	Operation continues

- When the ERR instruction is executed with n (self-diagnostic error code) set to U0, self-diagnostic errors that have an error code 80 or higher can be cleared.

[n]	Operation
U0	Self-diagnostic errors are cleared. The system relays (SR), system data (SD), and system monitors (SM) that are shown below are also cleared at the same time.

- When self-diagnostic errors are cleared, the following system relays (SR), system data (SD), and system monitors (SM) are also cleared at the same time.

Device No.	Application
SR0	Operation error detection (Latest)
SR2	Unit error occurrence
SR3	Unit warning occurrence
SR4	Unit verification error occurrence
SR7	Operation error detection (Hold)
SR8	Operation error detection (Latest)
SR22	RTC data error
SR23	Power supply unit service lifetime warning
SR24	RTC backup battery voltage drop hold
SR25	RTC backup battery voltage drop real
SR26	SNTP time update fail

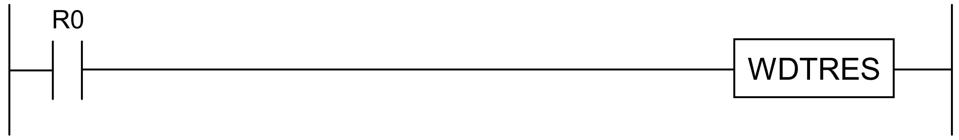
Device No.	Application
SD0	Self-diagnostic error code
SD2	Error occurrence slot number
SD3	Warning occurrence slot number
SD4	Verify error occurrence slot number
SD7	Hold operation error occurrence address (PB number)
SD8	Hold operation error occurrence address (32-bit lower address)
SD9	Hold operation error occurrence address (32-bit upper address)
SD10	Latest operation error occurrence address (PB number)
SD11	Latest operation error occurrence address (32-bit lower address)
SD12	Latest operation error occurrence address (32-bit upper address)

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the error code [n] is out of the range.

13.9 WDTRES (Watchdog Timer Reset)

■ Ladder diagram



■ Outline of operation

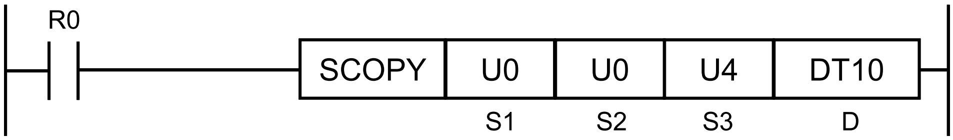
The watchdog timer is reset.

■ Flag operations

No change occurs.

13.10 SCOPY (System Area Copy)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Always zero (Device address storing the system area number to be copied or constant)
S2	The device address storing the starting number of system area or constant
S3	The device address storing the terminating number of system area or constant
D	Starting device address of destination area

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●								●	●				●
S2	●	●	●	●			●	●								●	●				●
S3	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

■ Outline of operation

This instruction copies data in the areas specified by [S1], [S2], and [S3] to the area specified by [D].

■ Processing

Example 1) When [S2] and [S3] are within the system monitor area

[S1]...U0 [S3]...U4 [D]...DT0

SM0	H 0011	→	DT10	H 0011
SM1	H 2233	→	DT11	H 2233
SM2	H 4455	→	DT12	H 4455
SM3	H 6677	→	DT13	H 6677
SM4	H 8899		DT14	H 1234

Example 2) When [S3] exceeds the upper limit of the system monitor area (when the upper limit of the system monitor area is 371)

[S1]...U0 [S2]...U0 [S3]...U400 [D]...DT0

SM0	**H 0011**	→	DT0	**H 0011**
SM1	**H 2233**	→	DT1	**H 2233**
SM2	**H 4455**	→	DT2	**H 4455**
SM3	**H 6677**	→	DT3	**H 6677**
SM4	**H 8899**	→	DT4	**H 8899**
SM5	**H AABB**	→	DT5	**H AABB**
⋮	⋮		⋮	⋮
SM367	**H CCDD**	→	DT367	**H CCDD**
SM368	**H EEFF**	→	DT368	**H EEFF**
SM369	**H FFEE**	→	DT369	**H FFEE**
SM370	**H DDCC**	→	DT370	**H DDCC**
SM371	**H BBAA**	→	DT371	**H BBAA**
			DT372	H 0000
			DT373	H 0000

■ Precautions for programming

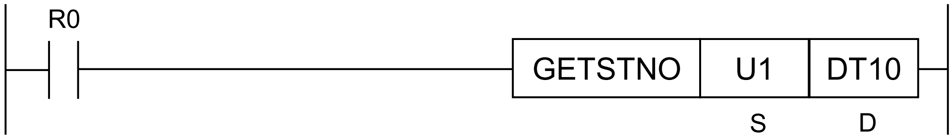
- [S1] is always zero. An operation error occurs when numbers other than zero are specified.
- Specify [S3] to be larger than [S2].
- When [S3] exceeds the upper limit of the system area, an operation error does not occur. The area from [S2] to the upper limit of the system area is copied.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [S1] is other than 0.
	To be set when [S2] is larger than [S3].
	To be set when [S2] is out of the accessible range.
	To be set when the destination range is outside the accessible range.

13.11 GETSTNO (Acquiring Starting Word Number of Specified Slot)

■ Ladder diagram



■ List of operands

Operand	Description
S	Specify the starting address of the device that stores the slot number or a constant.
D	Specify the starting address of a readout destination device.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

■ Outline of operation

- This instruction reads the starting word number of the slot specified by [S] and sets it in [D].

■ Precautions for programming

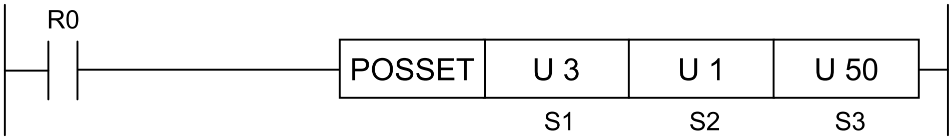
- An operation error occurs when a slot number without unit registration is specified when an I/O map has been registered.
- An operation error occurs when the number of a slot in which no unit is installed is specified when no I/O map has been registered.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when an out-of-range value is specified for [S] (slot number).
	To be set when a slot number (S) without unit registration is specified when an I/O map has been registered.
	To be set when a slot number (S) without an installed unit is specified when no I/O map has been registered.

13.12 POSSET (Setting of Positioning Starting Table)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Slot number where the positioning unit is attached (unsigned 16-bit integer)
S2	Axis number to start up the positioning table (unsigned 16-bit integer); 1 to 4: Axis 1 to 4; 8: Virtual axis
S3	Table number to start up the position control (unsigned 16-bit integer); 1 to 600, 10001 to 10100

(Note 1) If a value out of the available range is set as the starting table number, the positioning unit gives an error notification.

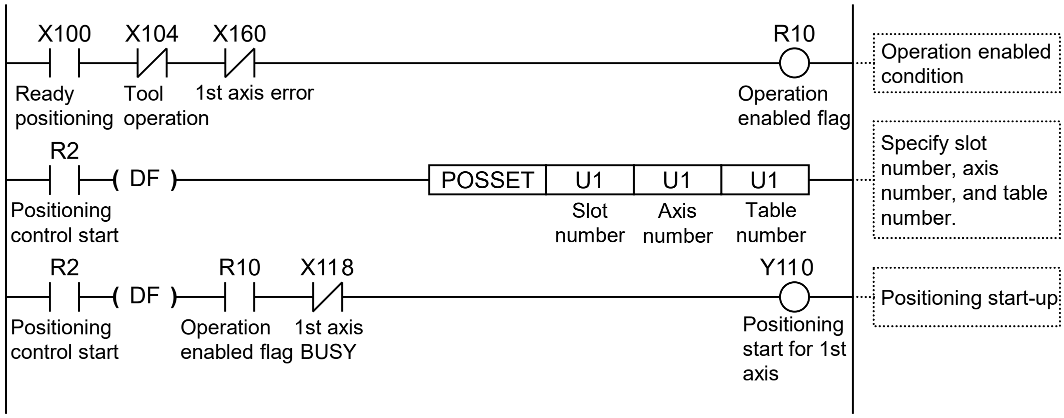
■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●					●	●				●
S2	●	●	●	●			●	●	●	●	●					●	●				●
S3	●	●	●	●			●	●	●	●	●					●	●				●

■ Outline of operation

- It is described right before the program which starts positioning and the positioning data table to be started is set.
- For the axis number of the positioning units specified by [S1] and [S2], the data table specified by [S3] is set.
- Data for the positioning data table should be set using Configurator PM7 in the tool software FPWIN GR7, or a user program.
- Positioning parameters should be set in the configuration menu of the tool software FPWIN GR7.

■ Program example

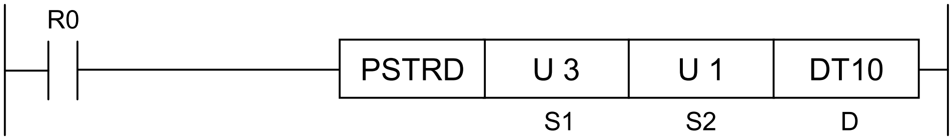


■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).
SR8 (ER)	To be set when the slot number and/or the axis number is out of the available range.

13.13 PSTRD (Acquiring Axis Status)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Slot number where the positioning unit is attached (unsigned 16-bit integer)
S2	Axis number to read the axis status information (unsigned 16-bit integer); 1 to 4: Axis 1 to 4; 8: Virtual axis
D	Device address to store the axis status information (unsigned 16-bit integer)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●					●	●				●
S2	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

■ Outline of operation

- For the axis number of the positioning units specified by [S1] and [S2], the main flag status is acquired as the axis status.
- The acquired information is converted into integer values in hexadecimal format, in accordance with the relevant allocation, and stored in the area specified by [D].

■ Type of axis status information

Status information	Description
Tool operation	Turned ON in the case of tool operation using Configurator PM, regardless of the specified axis.
Error annunciation	Turned ON when an error occurs in the specified axis.
Warning annunciation	Turned ON when a warning occurs in the specified axis.
BUSY	Turned ON when the specified axis is operating.
Operation done	Turned ON when the specified axis has completed operation.
Home return done	Turned ON when the specified axis has completed home return.

■ Allocation of axis status information to be stored in [D]

bit	Status information	Axis 1	Axis 2	Axis 3	Axis 4	Virtual axis
0	Tool operation	X4	X4	X4	X4	X4
1	Error annunciation	X60	X61	X62	X63	X67
2	Warning annunciation	X68	X69	X6A	X6B	X6F
3	BUSY	X18	X19	X1A	X1B	X1F
4	Operation done	X20	X21	X22	X23	X27
5	Home return done	X28	X29	X2A	X2B	X2F

(Note 1) The I/O numbers in the above table show relative addresses based on the base word number. The I/O numbers actually used vary according to the slot number where the unit is installed and the starting word number. Example) The tool operation flag is X104 for slot number 1 if the starting word is number 10.

■ Example of processing

Axis status information for the first axis of the positioning unit attached to slot number 3 is read.

bit	Status information	Axis 1	Value
0	Tool operation	IN4	0
1	Error annunciation	IN60	0
2	Warning annunciation	IN68	0
3	BUSY	IN18	1
4	Operation done	IN20	0
5	Home return done	IN28	0

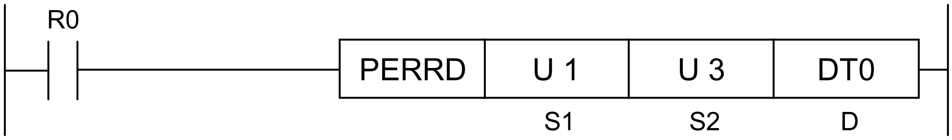
	Value
DT10	H 0008

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the slot number and/or the axis number is out of the available range.

13.14 PERRD (Acquiring Error/Warning in the Positioning Unit)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Slot number where the positioning unit is attached (unsigned 16-bit integer)
S2	Axis number to read the error/warning information (unsigned 16-bit integer); 1 to 4: Axis 1 to 4; 8: Virtual axis
D	Device address to store the error/warning code (unsigned 16-bit integer)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●					●	●				●
S2	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

■ Outline of operation

- For the axis number of the positioning units specified by [S1] and [S2], the error codes and warning codes stored in each annunciation buffer 1 are acquired.
- The error codes are stored in the area specified by [D], and warning codes in the area specified by [D+1].

■ Example of processing

- Error codes and warning codes for the third axis of the positioning unit attached to the slot number 1 are read.

Classification	Name	UM (Hex)	Value
Error information	Three-axis error code Alarm buffer 1	UM 0015A	H 4022
Warning information	Three-axis warning code Alarm buffer 1	UM 001E2	H B010

・Storage location
[D]……DT0 H 4022
[D+1]…DT1 H B010

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the slot number and/or the axis number is out of the available range.
	To be set when the destination range is outside the accessible range.

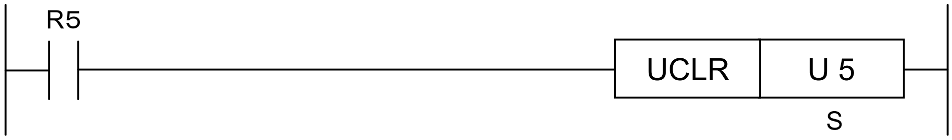
■ Error code/warning code annunciation buffers

Name	Unit memory number (HEX)				
	Axis 1	Axis 2	Axis 3	Axis 4	Virtual axis
Number of occurrences of errors	UM00129	UM00139	UM00149	UM00159	UM00199
Error code annunciation buffer 1	UM0012A UM0012B	UM0013A UM0013B	UM0014A UM0014B	UM0015A UM0015B	UM0019A UM0019B
Error code annunciation buffer 2	UM0012C UM0012D	UM0013C UM0013D	UM0014C UM0014D	UM0015C UM0015D	UM0019C UM0019D
Error code annunciation buffer 3	UM0012E UM0012F	UM0013E UM0013F	UM0014E UM0014F	UM0015E UM0015F	UM0019E UM0019F
Error code annunciation buffer 4	UM00130 UM00131	UM00140 UM00141	UM00150 UM00151	UM00160 UM00161	UM001A0 UM001A1
Error code annunciation buffer 5	UM00132 UM00133	UM00142 UM00143	UM00152 UM00153	UM00162 UM00163	UM001A2 UM001A3
Error code annunciation buffer 6	UM00134 UM00135	UM00144 UM00145	UM00154 UM00155	UM00164 UM00165	UM001A4 UM001A5
Error code annunciation buffer 7	UM00136 UM00137	UM00146 UM00147	UM00156 UM00157	UM00166 UM00167	UM001A6 UM001A7
Number of occurrences of warnings	UM001C1	UM001D1	UM001E1	UM001F1	UM00231
Warning code annunciation buffer 1	UM001C2 UM001C3	UM001D2 UM001D3	UM001E2 UM001E3	UM001F2 UM001F3	UM00232 UM00233
Warning code annunciation buffer 2	UM001C4 UM001C5	UM001D4 UM001D5	UM001E4 UM001E5	UM001F4 UM001F5	UM00234 UM00235
Warning code annunciation buffer 3	UM001C6 UM001C7	UM001D6 UM001D7	UM001E6 UM001E7	UM001F6 UM001F7	UM00236 UM00237

Name	Unit memory number (HEX)				
	Axis 1	Axis 2	Axis 3	Axis 4	Virtual axis
Warning code annunciation buffer 4	UM001C8 UM001C9	UM001D8 UM001D9	UM001E8 UM001E9	UM001F8 UM001F9	UM00238 UM00239
Warning code annunciation buffer 5	UM001CA UM001CB	UM001DA UM001DB	UM001EA UM001EB	UM001FA UM001FB	UM0023A UM0023B
Warning code annunciation buffer 6	UM001CC UM001CD	UM001DC UM001DD	UM001EC UM001ED	UM001FC UM001FD	UM0023C UM0023D
Warning code annunciation buffer 7	UM001CE UM001CF	UM001DE UM001DF	UM001EE UM001EF	UM001FE UM001FF	UM0023E UM0023F

13.15 UCLR (Error/Warning Clear)

■ Ladder diagram



■ List of operands

Operand	Description
S	Specify the slot number (unsigned 16-bit integer)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●		●						●	●				●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

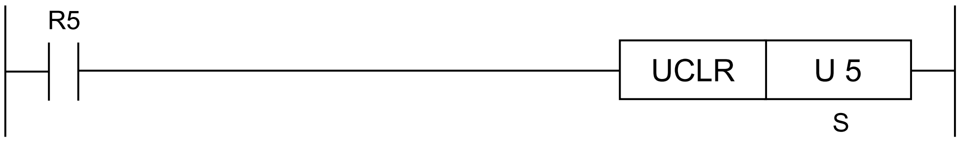
■ Outline of operation

- This instruction clears the errors/warnings for the unit installed in the slot number specified by [S].
- The UCLR instruction can clear the errors/warnings for the following units:

  High-speed counter unit, positioning unit, pulse output unit, motion control unit, serial communication unit

■ Program example

Example) Clear an error/warning in the unit attached to Slot No.5
[S]…U5



■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the slot number and/or the axis number is out of the available range.

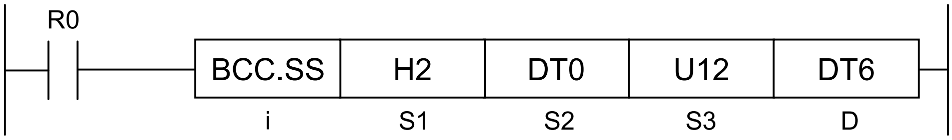
(MEMO)

14 High-level Instructions (Strings)

Applicable Models: All Models

14.1 BCC (Block Check Code Calculation)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S1	Area storing data specifying the calculation method, or constant data
S2	Starting address of the area storing target data
S3	The area that stores the length (number of bytes) of the targeted data, or the constant
D	Area that stores operation results

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TCS	TCE	IX	K (Note 1)	U (Note 2)	H (Note 3)	SF	DF	" "	
S1	●	●	●	●			●	●							●	●	●				●
S2	●	●	●	●			●	●													●
S3 (Note 4)	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

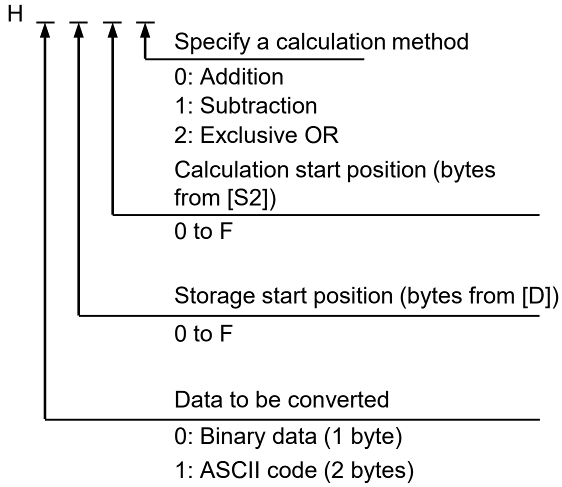
(Note 1) Can be specified only when the operation unit is signed integer (SS).
(Note 2) Can be specified only when the operation unit is unsigned integer (US).
(Note 3) Can be specified only when the operation unit is an integer (US, SS).
(Note 4) 16-bit unsigned integers, regardless of operation unit.

■ Outline of operation

- Calculates block check code (BCC).

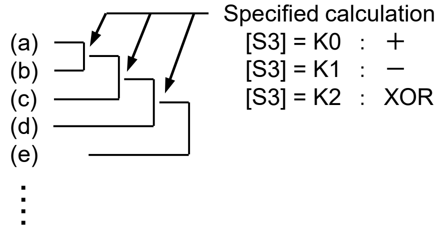
- The block check code (BCC) for the targeted data, which is the number of bytes specified by [S3] starting from the calculation start position specified by [S2], is calculated with the calculation method specified by [S1].
- The calculation result is stored according to the conversion method specified by [S1], starting from the storage position specified by [D] and [S1].

■ Specification of control data [S1]



As indicated below, calculation should be carried out as specified, for every eight bits.

% →	H 25 →	00100101	(a)
0 →	H 30 →	00110000	(b)
1 →	H 31 →	00110001	(c)
# →	H 23 →	00100011	(d)
R →	H 52 →	01010010	(e)
⋮			



■ Conversion example

- Calculate the block check code (BCC) for a message to be sent "%01#RCSX0000", and append the result to the message.

Example 1) Operation unit: 16 bits (SS) / Calculation method: Exclusive OR; Data to be converted: ASCII code

[BCC.SS H1002 , DT0 , U12 , DT6]

[S1] Exclusive OR

[S2] Start of the targeted data

[S3] Length of the targeted data

[D] Calculation result

		(characters)
DT0	H 3025	0%
DT1	H 2331	#1
DT2	H 4352	CR
DT3	H 5853	XS
DT4	H 3030	00
DT5	H 3030	00
DT6	H 0000	

BCC ⟶

		(characters)
DT0	H 3025	0%
DT1	H 2331	#1
DT2	H 4352	CR
DT3	H 5853	XS
DT4	H 3030	00
DT5	H 3030	00
DT6	H **4431**	**D1**

Example 2) Operation unit: 16 bits (US) / Calculation method: Addition; Data to be converted: ASCII code

[BCC.US H1000 , DT0 , HC , DT6]

[S1] Addition

[S2] Start of the targeted data

[S3] Length of the targeted data

[D] Calculation result

		(characters)
DT0	H 3025	0%
DT1	H 2331	#1
DT2	H 4352	CR
DT3	H 5853	XS
DT4	H 3030	00
DT5	H 3030	00
DT6	H 0000	

BCC ⟶

		(characters)
DT0	H 3025	0%
DT1	H 2331	#1
DT2	H 4352	CR
DT3	H 5853	XS
DT4	H 3030	00
DT5	H 3030	00
DT6	H 3941	**9A**

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).

14.2 CRC (CRC Code Calculation)

■ Ladder diagram

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S1	Area storing data specifying the calculation method, or constant data
S2	Starting address of the area storing target data
S3	The area that stores the length (number of bytes) of the targeted data, or the constant
D	Area that stores operation results

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K (Note 1)	U (Note 2)	H (Note 3)	SF	DF	" "	
S1	●	●	●	●			●	●							●	●	●				●
S2	●	●	●	●			●	●													●
S3(Note 4)	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

(Note 1) Can be specified only when the operation unit is signed integer (SS).

(Note 2) Can be specified only when the operation unit is unsigned integer (US).

(Note 3) Can be specified only when the operation unit is an integer (US, SS).

(Note 4) 16-bit unsigned integers, regardless of operation unit.

■ Outline of operation

- This instruction calculates block check code (BCC) using the CRC (16-bit CRC code) calculation method.

- The block check code (BCC) for the targeted data, which is the number of bytes specified by [S3] starting from the calculation start position specified by [S2], is calculated with the CRC calculation method which is specified by [S1].
- The calculation result is stored according to the conversion method specified by [S1], starting from the storage position specified by [D] and [S1].

■ CRC calculation

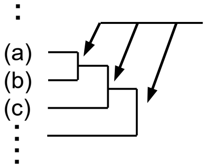
- Apply a specified calculation method to every 16 bits as follows.
- The calculation should be carried out using the following generating polynomial. (This is the same calculation method as MODBUS-RTU.)

・Generating polynomial: (CRC-16) $X^{16} + X^{15} + X^{2} + 1$

・Generating polynomial: (CRCITT) $X^{16} + X^{12} + X^{5} + 1$

%0	→	H 2530	→	0010 0101 0011 0000	(a)
1#	→	H 3123	→	0011 0001 0010 0011	(b)
RC	→	H 5243	→	0101 0010 0100 0011	(c)

⋮



Specified calculation
[S1]= K0 : CRC-16
[S1]= K1 : CCITT

■ Specification of control data [S1]

H _ _ _ _

0: Fixed

Storage starting position (Number of bytes from [D])
0 to F

Calculation starting position (Number of bytes from [S2])
0 to F

Specification of the calculation method

Setting value	CRC type	Default	Shift method	XOR of operation result	
0	CRC-16	FFFFH	Right shift	0000H	
1	CCITT	FFFFH	Right shift	0000H	
2	CRC-16	0000H	Right shift	0000H	(Note)
3	CCITT	0000H	Right shift	0000H	(Note)
4	CRC-16	FFFFH	Left shift	0000H	(Note)
5	CCITT	FFFFH	Left shift	0000H	(Note)
6	CRC-16	0000H	Left shift	0000H	(Note)
7	CCITT	0000H	Left shift	0000H	(Note)
8	CRC-16	FFFFH	Right shift	FFFFH	(Note)
9	CCITT	FFFFH	Right shift	FFFFH	(Note)
A	CRC-16	0000H	Right shift	FFFFH	(Note)
B	CCITT	0000H	Right shift	FFFFH	(Note)
C	CRC-16	FFFFH	Left shift	FFFFH	(Note)
D	CCITT	FFFFH	Left shift	FFFFH	(Note)
E	CRC-16	0000H	Left shift	FFFFH	(Note)
F	CCITT	0000H	Left shift	FFFFH	(Note)

(Note): Can be used with CPU units CPS4*/CPS3* Ver.4.32 or later, CPS2* Ver.1.32 or later

■ Conversion example

Calculate the block check code for a message to be sent "%01#RCSX0000", and append the result to the message.

Example 1) Operation unit: 16 bits (SS) / Calculation method: CRC-16; Initial value = FFFFH; Right shift; XOR = 0000H

```
[CRC.SS    H 0 , DT0 , U12 , DT6 ]
```

[S1] CRC-16
[S2] Header of target data
[S3] Length of target data (12 bytes)
[D] Calculation result

		(Characters)				(Characters)
DT0	H 3025	0%		DT0	H 3025	0%
DT1	H 2331	#1		DT1	H 2331	#1
DT2	H 4352	CR		DT2	H 4352	CR
DT3	H 5853	XS		DT3	H 5853	XS
DT4	H 3030	00		DT4	H 3030	00
DT5	H 3030	00		DT5	H 3030	00
DT6	H 0000		CRC →	DT6	H 2E0A	

Example 2) Operation unit: 16 bits (US) / Calculation method: CRC-16; Initial value = FFFFH; Right shift; XOR = 0000H

```
[CRC.US    H 1 , DT0 , U12 , DT6 ]
```

[S1] CRCITT
[S2] Header of target data
[S3] Length of target data (12 bytes)
[D] Calculation result

		(Characters)				(Characters)
DT0	H 3025	0%		DT0	H 3025	0%
DT1	H 2331	#1		DT1	H 2331	#1
DT2	H 4352	CR		DT2	H 4352	CR
DT3	H 5853	XS		DT3	H 5853	XS
DT4	H 3030	00		DT4	H 3030	00
DT5	H 3030	00		DT5	H 3030	00
DT6	H 0000		CRCITT →	DT6	H A02E	

Example 3) Operation unit: 16 bits (SS) / Calculation method: CRC-16; Initial value = 0000H; Right shift; XOR = 0000H

[CRC.SS H 2, DT0, U12, DT6]

[S1] CRC16
[S2] Header of target data
[S3] Length of target data (12 bytes)
[D] Calculation result

		(Characters)
DT0	H 3025	0%
DT1	H 2331	#1
DT2	H 4352	CR
DT3	H 5853	XS
DT4	H 3030	00
DT5	H 3030	00
DT6	H 0000	

CRC →

		(Characters)
DT0	H 3025	0%
DT1	H 2331	#1
DT2	H 4352	CR
DT3	H 5853	XS
DT4	H 3030	00
DT5	H 3030	00
DT6	H 2C6E	

Example 4) Operation unit: 16 bits (US) / Calculation method: CRCITT; Initial value = FFFFH; Left shift; XOR = 0000H

[CRC.US H 5, DT0, U12, DT6]

[S1] CRCITT
[S2] Header of target data
[S3] Length of target data (12 bytes)
[D] Calculation result

		(Characters)
DT0	H 3025	0%
DT1	H 2331	#1
DT2	H 4352	CR
DT3	H 5853	XS
DT4	H 3030	00
DT5	H 3030	00
DT6	H 0000	

CRCITT →

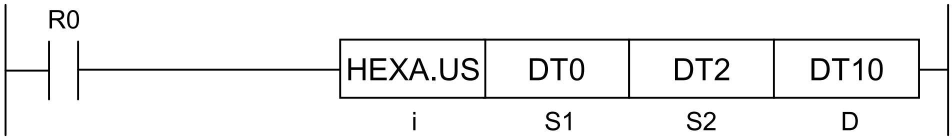
		(Characters)
DT0	H 3025	0%
DT1	H 2331	#1
DT2	H 4352	CR
DT3	H 5853	XS
DT4	H 3030	00
DT5	H 3030	00
DT6	H 6194	

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification). To be set in the case that the length of the target data for [S3] is 0. To be set in the case that the control data specified by [S1] is out of the specified range. To be set when [S2 + calculation starting position + S3] exceeds the [S2] area. To be set when the address specified by [D + storage starting position] exceeds the [D] area.

14.3 HEXA (Conversion: HEX → Hexadecimal ASCII)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S1	Starting number for the area storing the hexadecimal numeric values
S2	The area that stores the length (number of bytes) to be converted, or the constant
D	Starting number of the area that stores the ASCII code as the conversion result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K (Note 2)	U (Note 3)	H (Note 4)	SF	DF	" "	
S1	●	●	●	●	●	●	●	●													●
S2	●	●	●	●			●	●							●	●	●				●
D	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 2) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 3) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 4) Can be specified only when the operation unit is integer (US, SS, UL, SL).

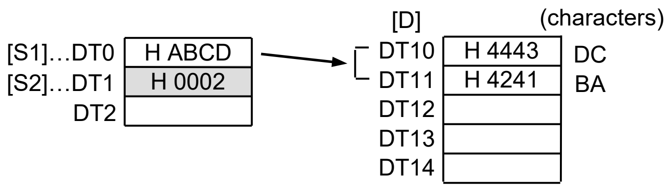
■ Outline of operation

- This instruction converts a hexadecimal figure into an ASCII code.
- The hexadecimal numerical data stored in the area specified by [S1] is converted into an ASCII code and stored in the area specified by [D].
- [S2] specifies the number of data bytes to be converted.

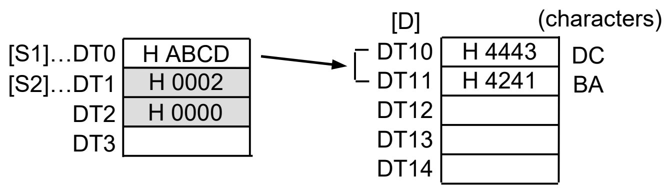
- The result (ASCII code) should have twice the size of the source data.

■ Conversion example

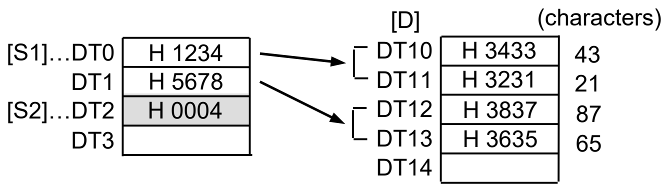
Example 1) Operation unit: 16 bits (US, SS)
[i]…US,SS
[S1]…DT0 [S2]…DT1
[D]…DT10



Example 2) Operation unit: 32 bits (UL, SL)
[i]…UL,SL
[S1]…DT0 [S2]…DT1
[D]…DT10

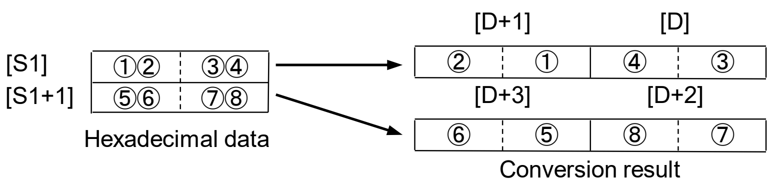


Example 3) Operation unit: 16 bits (US, SS)
[i]…US,SS
[S1]…DT0 [S2]…DT2
[D]…DT10



■ Precautions for programming

- The two characters that make up one byte are interchanged when stored.
- Converts two bytes as one section.

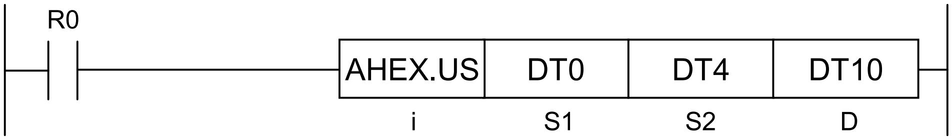


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the number of bytes specified by [S2] exceeds the area of the conversion range.
	To be set when the conversion result exceeds the area.
	To be set when [S2] is specified as '0' or a negative value.

14.4 AHEX (Conversion: Hexadecimal ASCII → HEX)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S1	Starting number of the area that stores an ASCII code
S2	The area that stores the number of ASCII codes (number of characters) to be converted, or the constant
D	Starting number of the area that stores the hexadecimal figure as the conversion result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K (Note 1)	U (Note 2)	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●													●
S2	●	●	●	●			●	●							●	●	●				●
D	●	●	●	●			●	●													●

(Note 1) Can be specified only when the operation unit is signed integer (SS, SL).

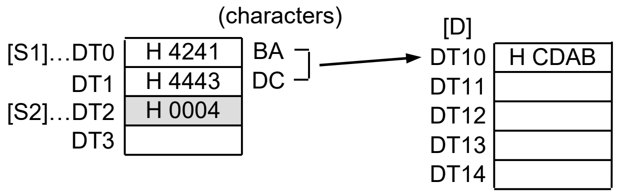
(Note 2) Can be specified only when the operation unit is unsigned integer (US, UL).

■ Outline of operation

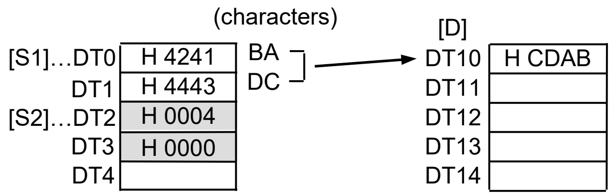
- This instruction converts the ASCII code string into a hexadecimal figure.
- The ASCII code stored in the area specified by [S1] is converted into a hexadecimal figure and stored in the area specified by [D].
- The number of ASCII codes (number of characters) to be converted is specified by [S2].
- The conversion results are stored in byte units.
- If the number of characters to be converted is an odd number, Bits 0 to 3 of the final data (bytes) of the conversion result are padded with '0'.

■ Conversion example

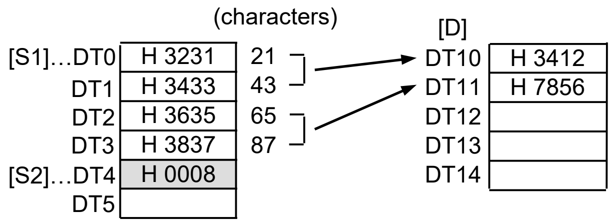
Example 1) Operation unit: 16 bits (US, SS)
[i]…US,SS
[S1]…DT0 [S2]…DT2
[D]…DT10



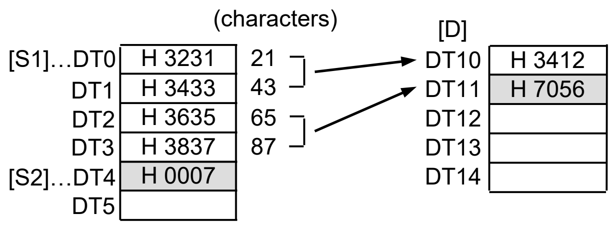
Example 2) Operation unit: 32 bits (UL, SL)
[i]…UL,SL
[S1]…DT0 [S2]…DT2
[D]…DT10



Example 3) Operation unit: 16 bits (US, SS)
[i]…US,SS
[S1]…DT0 [S2]…DT4
[D]…DT10

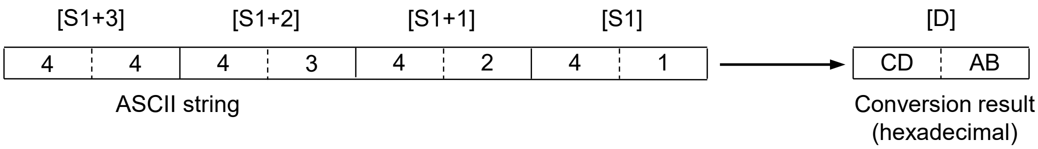


Example 4) Operation unit: 16 bits (US, SS); No. of characters to be converted is an odd number
[i]…US,SS
[S1]…DT0 [S2]…DT4
[D]…DT10



■ Precautions for programming

- Two characters of ASCII code are converted into two 1-byte numerical digits. At this time, the upper and lower characters are interchanged.
- Four characters are converted as one segment of data.

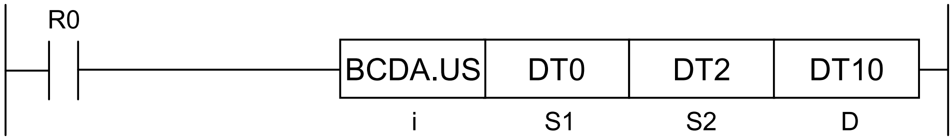


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the number of characters specified by [S2] exceeds the area of the conversion range.
	To be set when the conversion result exceeds the area.
	To be set when [S2] is specified as '0' or a negative value.
	To be set when the ASCII codes specified by [S1] contains character codes other than 0 to F.

14.5 BCDA (Conversion: BCD → Decimal ASCII)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
S1	Starting number of the area storing the BCD numerical value
S2	The area that stores the data that express data size to be converted or conversion direction, or the constant
D	Starting number of the area that stores the ASCII code as the conversion result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●													●
S2	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

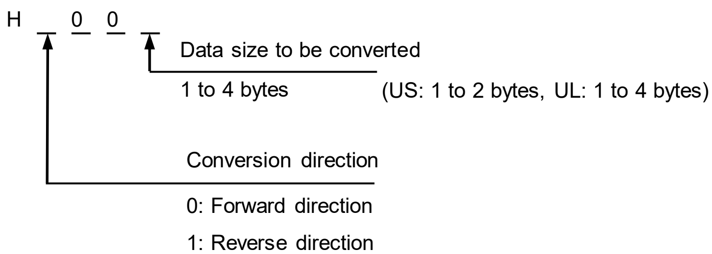
■ Outline of operation

- This instruction converts up to 8-digit BCD data into an ASCII code string.
- The BCD figure stored in the area specified by [S1] is converted into ASCII codes.
- The conversion result is stored in the area starting with [D].
- The BCD data size (number of bytes) to be converted, and the conversion direction, are specified by [S2].
- The conversion result (ASCII code) should have twice the size of the source data.
- The maximum value of the data size to be converted varies with the operation unit. (US: 2 bytes, UL: 4 bytes)

- Since the amount of data to be converted is specified in bytes, it is also possible to convert only the low byte of one word data.

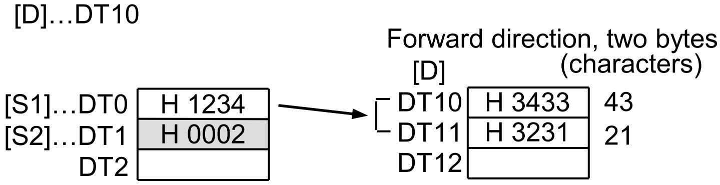
■ Setting the conversion data amount and conversion direction [S2]

Specify the value with a four-digit H constant.

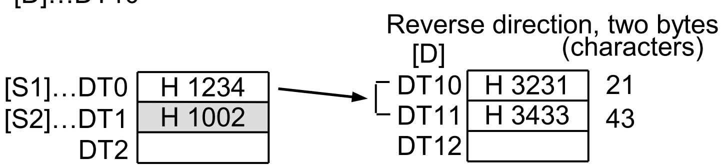


■ Conversion example

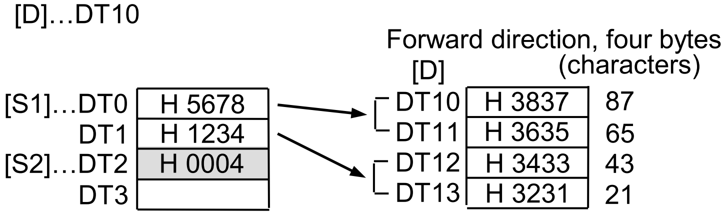
Example 1) Operation unit: 16 bits (US)
[i]…US
[S1]…DT0 [S2]…DT1
[D]…DT10



Example 2) Operation unit: 16 bits (US)
[i]…US
[S1]…DT0 [S2]…DT1
[D]…DT10



Example 3) Operation unit: 32 bits (UL)
[i]…UL
[S1]…DT0 [S2]…DT2
[D]…DT10

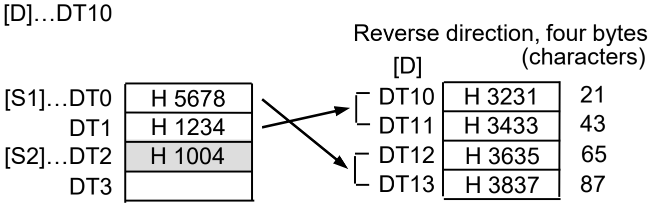


Example 4) Operation unit: 32 bits (UL)

[i]…UL

[S1]…DT0 [S2]…DT2

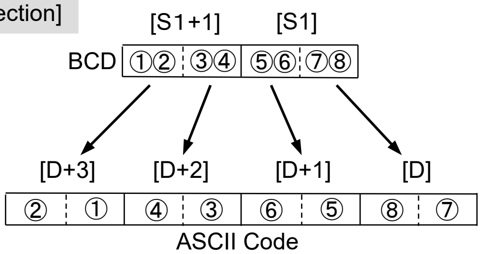
[D]…DT10



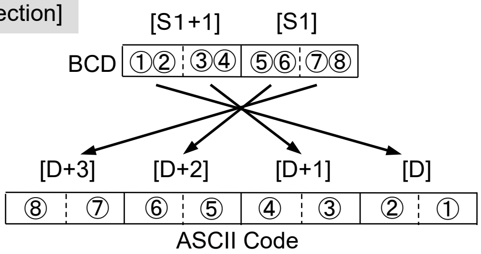
■ Precautions for programming

- The two characters that comprise 1 byte are stored in the opposite order following conversion.
- Converts two bytes as one section.

[Forward direction]



[Reverse direction]

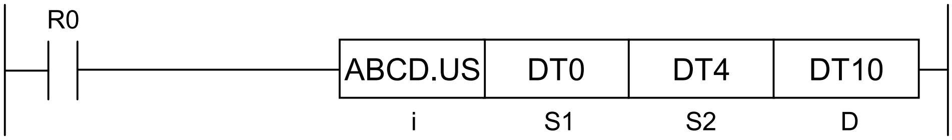


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the data specified by [S1] contain non-BCD data.
	To be set when the number of bytes specified by [S2] exceeds the [S1] area.
	To be set when the conversion result exceeds the area.
	To be set when [S2] is specified as '0'.
	To be set when the conversion direction of [S2] is out of the range.
	To be set when the conversion data size of [S2] is out of the range.

14.6 ABCD (Conversion: Decimal ASCII → BCD)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●		●			

■ List of operands

Operand	Description
S1	Starting number of the area that stores an ASCII code
S2	The area that stores the data that express the number of ASCII codes to be converted and conversion direction, or the constant
D	Number of the start of the area storing the BCD value that is the result of conversion

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●	●	●	●	●													●
S2	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Character constants cannot be specified.)

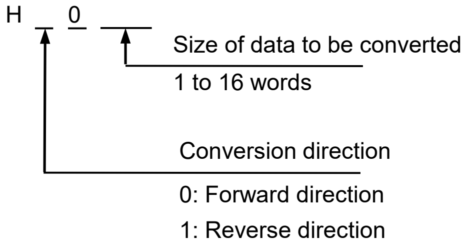
■ Outline of operation

- This instruction converts up to 8-character ASCII code string into BCD data.
- The ASCII code stored in the area specified by [S1] is converted into BCD data and stored in the area specified by [D].
- The number of ASCII codes (number of characters) to be converted and conversion direction are specified by [S2].
- The resulting BCD data should have half the size of the source ASCII code string.
- The maximum value of the data size to be converted varies with the operation unit. (US: 4 characters, UL: 8 characters)

- Since the amount of data to be converted is specified in bytes, it is also possible to convert only the low byte of one word data.
- If the data size to be converted is an odd number, bits of the final data of the conversion result are padded with '0'.
  1. Bits 0 to 3 are padded with '0'. (In the forward direction)
  2. Bits 0 to 7 are padded with '0'. (In the reverse direction)

■ Setting the conversion data amount and conversion direction [S2]

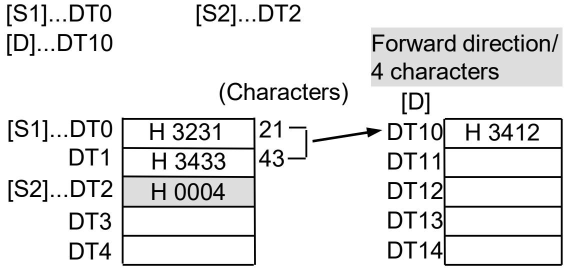
Specify a 4-digit BCD (H constant) according to the following format.



■ Conversion example

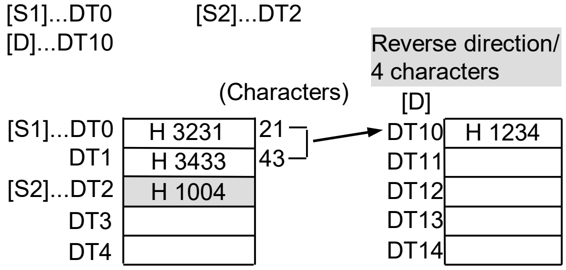
Example 1) Operation unit: 16 bits (US)

[i]...US
[S1]...DT0 [S2]...DT2
[D]...DT10



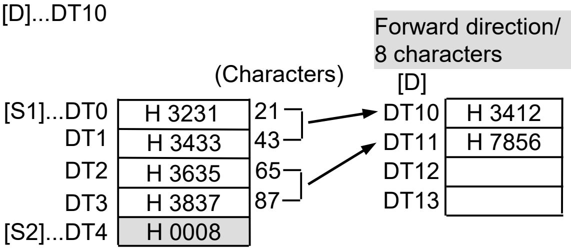
Example 2) Operation unit: 16 bits (US)

[i]...US
[S1]...DT0 [S2]...DT2
[D]...DT10



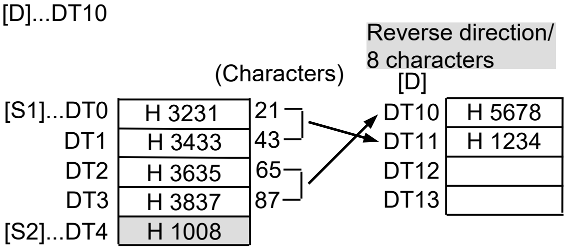
Example 3) Operation unit: 32 bits (UL)

[i]...UL
[S1]...DT0 [S2]...DT4
[D]...DT10



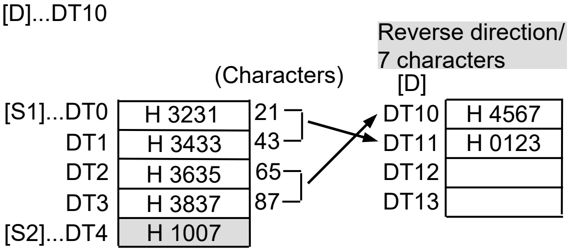
Example 4) Operation unit: 32 bits (UL)

[i]...UL
[S1]...DT0 [S2]...DT4
[D]...DT10



Example 5) Operation unit: 32 bits (UL)

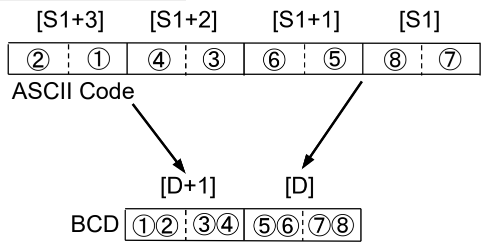
[i]...UL
[S1]...DT0 [S2]...DT4
[D]...DT10



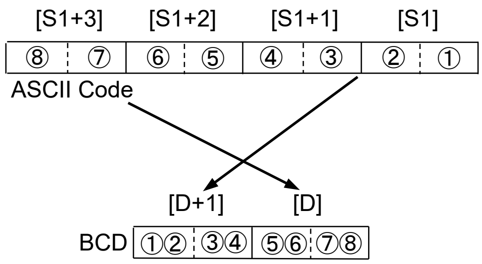
■ Precautions for programming

- The two characters that make up one byte are interchanged when stored.
- Converts two bytes as one section.

[Forward direction]



[Reverse direction]

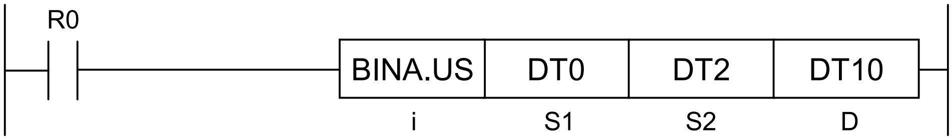


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the ASCII codes specified by [S1] contain data other than 0 to 9.
	To be set when the number of bytes specified by [S2] exceeds the [S1] area.
	To be set when the conversion result exceeds the area.
	To be set when [S2] is specified as '0'.
	To be set when the conversion direction of [S2] is out of the range.
	To be set when the conversion data size of [S2] is out of the range.

14.7 BINA (Conversion: BIN → Decimal ASCII)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S1	The area that stores the BIN data that express a decimal figure, or the constant
S2	The area that stores the number of bytes of the area to store the conversion result, or the constant
D	Starting number of the area that stores the ASCII code as the conversion result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K (Note 2)	U (Note 3)	H (Note 4)	SF	DF	" "	
S1	●	●	●	●	●	●	●	●							●	●	●				●
S2	●	●	●	●			●	●							●	●	●				●
D	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 2) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 3) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 4) Can be specified only when the operation unit is an integer (US, SS, UL, SL).

■ Outline of operation

- This instruction converts BIN data that expresses a decimal figure into an ASCII code string.
- The BIN data expressed as a decimal figure specified by [S1] is converted into an ASCII code and stored in the area specified by [D].
- The start of the storage area is specified by [D] and its size is specified by [S2].

- If the number of bytes of the resulting ASCII code (including the negative sign) is larger than the number of bytes specified by [S2], an operation error occurs.
- For [S2], specify the number of digits for the conversion target including the sign.

■ Conversion example

Example 1) Operation unit: 16 bits (US)

[i]…US
[S1]…DT0   [S2]…DT1
[D]…DT10

				[D]		(characters)
[S1]…DT0	H FF9C	(U 65436)	→	DT10	H 3620	6
[S2]…DT1	H 0006			DT11	H 3435	4 5
				DT12	H 3633	6 3

DT0: H FF9C (U 65436) → DT10 to DT12: ” 65436”

Example 2) Operation unit: 16 bits (SS)

[i]…SS
[S1]…DT0   [S2]…DT1
[D]…DT10

				[D]		(characters)
[S1]…DT0	H FF9C	(K -100)	→	DT10	H 2020	
[S2]…DT1	H 0006			DT11	H 312D	1 -
				DT12	H 3030	0 0

DT0: H FF9C (K -100) → DT10 to DT12: ” -100”

Example 3) Operation unit: 32 bits (UL)

[i]…UL
[S1]…DT0   [S2]…DT2
[D]…DT10

			[D]		(characters)
[S1]…DT0	H FF9C	→	DT10	H 3234	2 4
DT1	H FFFF		DT11	H 3439	4 9
[S2]…DT2	H 000A		DT12	H 3639	6 9
			DT13	H 3137	1 7
			DT14	H 3639	6 9

DT0: H FFFF FF9C (U 4294967196) → DT10 to DT14: ” 4294967196”

Example 4) Operation unit: 32 bits (SL)

[i]…SL
[S1]…DT0   [S2]…DT2
[D]…DT10

			[D]		(characters)
[S1]…DT0	H FF9C	→	DT10	H 2020	
DT1	H FFFF		DT11	H 2020	
[S2]…DT2	H 000A		DT12	H 2020	
			DT13	H 312D	1 -
			DT14	H 3030	0 0

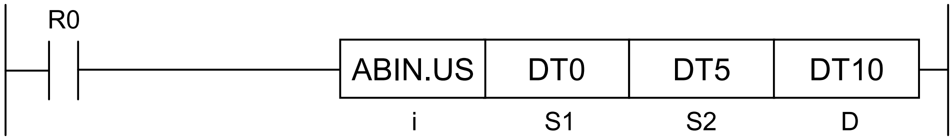
DT0: H FFFF FF9C (K -100) → DT10 to DT14: ” -100”

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the number of bytes specified by [S2] exceeds the [D] area.
	To be set when the conversion result exceeds the area.
	To be set when the resulting number of bytes exceeds the number of bytes specified by [S2].

14.8 ABIN (Conversion: Decimal ASCII → BIN)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

■ List of operands

Operand	Description
S1	Starting number of the area that stores the ASCII code to be converted
S2	The area that stores the number of bytes to be converted, or the constant
D	Area to store the conversion result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K (Note 2)	U (Note 3)	H (Note 4)	SF	DF	" "	
S1	●	●	●	●	●	●	●	●													●
S2 (Note 5)	●	●	●	●			●	●							●	●	●				●
D	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices, 32-bit devices, and integer constants can be modified. (Real number constants, and character constants cannot be specified.)

(Note 2) Can be specified only when the operation unit is signed integer (SS, SL).

(Note 3) Can be specified only when the operation unit is unsigned integer (US, UL).

(Note 4) Can be specified only when the operation unit is an integer (US, SS, UL, SL).

(Note 5) To be handled as a 16-bit integer (US, SS), regardless of operation unit.

■ Outline of operation

- This instruction converts the ASCII code string into BIN data that express a decimal figure.

- The ASCII code that express a decimal figure equivalent to the number of bytes (i.e. number of characters) specified by [S2], starting from the area specified by [S1], is converted into a decimal figure and stored in the area specified by [D].
- If a negative sign (-) is contained in the ASCII code specified by [S1], specify SS or SL for the operation unit. An operation error occurs if operation units US or UL are used.

■ Conversion example

Example 1) Operation unit: 16 bits (US)

[i]…US

[S1]…DT0 [S2]…DT3 [D]…DT10

		(characters)
[S1]…DT0	H 3620	6
DT1	H 3435	4 5
DT2	H 3633	6 3
[S2]…DT3	H 0006	

[D]		
DT10	H FF9C	(U 65436)
DT11		

DT0 to DT2: "65436" → DT10: H FF9C (U 65436)

Example 2) Operation unit: 16 bits (SS)

[i]…SS

[S1]…DT0 [S2]…DT3 [D]…DT10

		(characters)
[S1]…DT0	H 2020	
DT1	H 312D	1 -
DT2	H 3030	0 0
[S2]…DT3	H 0006	

[D]		
DT10	H FF9C	(K -100)
DT11		

DT0 to DT2: "-100" → DT10: H FF9C (K -100)

Example 3) Operation unit: 32 bits (UL)

[i]…UL

[S1]…DT0 [S2]…DT5 [D]…DT10

		(characters)
[S1]…DT0	H 3234	2 4
DT1	H 3439	4 9
DT2	H 3639	6 9
DT3	H 3137	1 7
DT4	H 3639	6 9
[S2]…DT5	H 000A	

[D]		
DT10	H FF9C	(U 429467196)
DT11	H FFFF	
DT12		

DT0 to DT4: "4294967196" → DT10 to DT11: H FFFF FF9C (U 4294967196)

Example 4) Operation unit: 32 bits (SL)

[i]…SL

[S1]…DT0 [S2]…DT5 [D]…DT10

		(characters)
[S1]…DT0	H 3234	
DT1	H 3439	
DT2	H 3639	
DT3	H 3137	1 -
DT4	H 3639	0 0
[S2]…DT5	H 0006	

[D]		
DT10	H FF9C	(K -100)
DT11	H FFFF	
DT12		

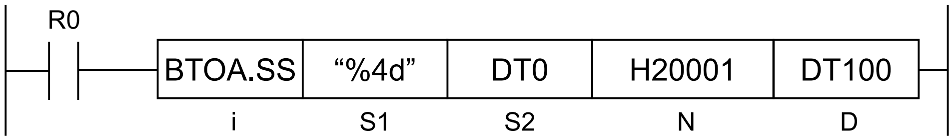
DT0 to DT4: "-100" → DT10 to DT11: H FFFF FF9C (K -100)

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the number of bytes specified by [S2] exceeds the [D] area.
	To be set when the conversion result exceeds the area.
	To be set when [S1] contains character ASCII codes other than 0 to 9, sign code, or space.

14.9 BTOA (Conversion: BIN → ASCII)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Control string (2 to 16 characters)
S2	Starting address of the device that stores binary data
N	Conversion method
D	Starting address of the device that stores the ASCII code as the conversion result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●	●	●	●	●													●
N(Note 1)	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

(Note 1) To be handled as a 32-bit integer (UL), regardless of operation unit.

■ Outline of operation

- This instruction converts the binary data stored in the area starting with [S2] into ASCII codes.
- For [S1], specify the type, number of digits, and precision of the data to be converted.
- For [N], specify the number of data to be converted, the storage starting position, and the conversion direction.
- The conversion result is stored in the area starting with [D].

- The maximum number of characters after conversion for a single datum is 32. An operation error occurs when it exceeds 32 characters.
- To directly specify a character constant for the operand [S1], enter the constant in the "Instruction list" dialog box in the FPWIN GR7 programming tool.

■ Setting the control string [S1]

Specifies the conversion data type, number of characters, precision, etc. using the string data of the following formats. A variety of options (such as inserting a sign or spaces) can also be selected depending on the type of data to be converted. For details, see below.

[S1] = " % + 12.5 d , "

Option setting (1)

0 : Zero padding
+ : A sign is added (plus sign)
␣: A space is inserted
- : Left align (default is right align)
: Characters are added according to the conversion data

Option setting (2)

, : A comma is added
BCD : Postfix characters are added
H: Postfix characters are added

Number of characters after conversion and the precision

Specify the total number of characters (n) and the number of characters of precision (m) with [n.m], [n], or [.m]. The number of characters of precision (m) changes according to the type of conversion data

Type of data to be converted	Number of characters of precision (m)
d , i , u , x , X , b	represents the number of characters in numerical strings.
f	represents the number of characters after the decimal point.
g	represents the number of significant figures.

Type of data to be converted

d: Signed integer → Decimal ASCII
u: Unsigned integer → Decimal ASCII
x: Unsigned integer → Hexadecimal ASCII
b: BDC integer → Hexadecimal ASCII
f: Floating point real number → Floating point ASCII
e: Floating point real number → Exponential notation ASCII
g: Floating point real number → Floating point ASCII or exponential notation ASCII

■ Setting the conversion method [N]

Conversion method [N] should be specified in a Hex format in the 32-bit area.

[N] = H 2 00 01

Amount of data to be converted

H1 to H FFFF (1 to 65535)

Convert the data to hexadecimal and specify it here

Storage starting position

H0 to H FF (Conversion)

Specify the header where the converted character string is stored from the relative position of the low byte of [D]. The round numbers in the following diagram indicate the relative position. The character string is stored from relative position ① for H00 and from relative position ② for H01.

	H	L
[D]	②	①
[D+1]	④	③
[D+2]	⑥	⑤
[D+3]	⑧	⑦

Conversion direction

H00: Forward direction
H01: Reverse direction

Specify a conversion direction. Forward direction can only be specified for hexadecimal data (control string %x) or BCD data (control string %b). For all other data formats, reverse direction is specified (fixed).

[S2] H 1234

Forward direction

			(Characters)
[D]	H31	H32	1 2
[D+1]	H33	H34	3 4

Reverse direction

			(Characters)
[D]	H32	H31	2 1
[D+1]	H34	H33	4 3

■ Setting conversion data for the control string [S1]

Control character string	Data format: BIN data before conversion	Data format: ASCII data after conversion	Available operation units	Usage example
"%d" or "%i"	16-bit data (signed integer)	Decimal ASCII data	SS	"%d" , "%5d" , "%+5d" , "%-5d" , %05d" , %10.5d" , "%d," , "% d"
	32-bit data (signed integer)	Decimal ASCII data	SL	
"%u"	16-bit data (unsigned integer)	Decimal ASCII data	US	"%u" , "%5u" , "%+5u" , "%-5u","%05u" , %10.5u" , "%u,"
	32-bit data (unsigned integer)	Decimal ASCII data	UL	
"%x"	16-bit data	Hexadecimal ASCII data (Forward/reverse direction)	US	"%x" , "%5x" , "%-5x" , "%05x", "%10.5x" , "%x," , "%#x" , "%X"
	32-bit data	Hexadecimal ASCII data (Forward/reverse direction)	UL	
"%b"	16-bit BCD data	Decimal ASCII data (Forward/reverse direction)	US	"%b" , "%5b" , "%-5b" , "%05b", "%10.5b" , "%b,"
	32-bit BCD data	Decimal ASCII data (Forward/reverse direction)	UL	
"%f"	32-bit single-precision real number data	Floating point number ASCII data	SF	"%f" , "%5.2f" , "%+5.2f" "%-5.2f" , "%05.2f" , "%f," "%#f" , "% f"
	64-bit double-precision real number data	Floating point number ASCII data	DF	
"%e"	32-bit single-precision real number data	Exponential notation ASCII data	SF	"%e" , "% 5.2e" , "%+5.2e" , "%-5.2e" , "%05.2e" ,"%# 5.2e" , "% e" , "%E"
	64-bit double-precision real number data	Convert to exponential notation ASCII data	DF	
"%g"	32-bit single-precision real number data	Exponential notation ASCII data or floating-point ASCII data (whichever is shorter in the relevant notation)	SF	"%g" , "%5.2g" , "%+5.2g", "%-5.2g" , "%05.2g" , "%g,", "%#5.2g" , "%G"
	64-bit double-precision real number data	Exponential notation ASCII data or floating-point ASCII data (whichever is shorter in the relevant notation)	DF	

(Note 1) The number of converted digits for a control string is up to 16.

■ Application example of control string [S1]

The following is a conversion example according to the operation unit and conversion direction.

Operation unit	Conversion direction	Control character string	Binary data	ASCII data	Description
US	Reverse	%+12.5u	K 1234	"␣␣␣␣␣␣␣01234"	When %u is specified, the result is not signed.

Operation unit	Conversion direction	Control character string	Binary data	ASCII data	Description
US	Reverse	%+10.7b	H 0123	"␣␣␣0000123"	When %b is specified, the result is not signed.
US	Forward	%8.5X,	H 0123	"2301␣0␣␣"	If an 8-character string is converted in the forward direction, it appears as shown below.
US	Forward	%05b	H 01233456	"2301056340"	Conversion of an odd number of characters in the forward direction is as shown below.
SF	Reverse	%+10.3e	SF123.4567	"1.235e+02␣"	The exponent is output with at least two digits.

■ Options for the control string [S1]

Items	Control character string	Binary data	ASCII data	Description
Specification of upper / lower case characters	%x	H ABCD	"abcd"	Specify alphabetical upper or lower case characters used in hexadecimal and exponential notation ASCII data.
	%X	H ABCD	"ABCD"	
	%e	SF1234.567	"1.234567e+3"	
	%E	SF1234.567	"1.234567E+3"	
Specification of the number of display digits	%d	K 100	"100"	The display digit is specified with "Total number of characters" and "Number of characters of precision." It is specified with “n.m”, “n”, or “.m”, etc. n: Total number of characters, m: Number of characters of precision <Number of characters of precision> [d , i , u , x , X , b] represent the number of characters in numerical strings. [f , e , E] represent the number of characters after the decimal point. [g , G] represent the number of significant figures. If the number of characters is not specified, the number of digits and the storage area size of the converted data will vary according to the data before conversion.
	%5d	K 100	"␣␣100"	
	%10.5d	K 100	"␣␣␣␣␣00100"	
	%x	H 12A	"12a"	
	%5x	H 12A	"␣␣12a"	
	%10.5x	H 12A	"␣␣␣␣␣0012a"	
	%b	H 123	"123"	
	%5b	H 123	"␣␣123"	
	%f	SF 123.4567	"123.4567"	
	%8.3f	SF 123.4567	"␣123.457"	
	%e	SF 1234.567	"1.234567e+03"	
	%10.3e	SF 1234.567	"␣1.235e+03"	
	%g	SF 1234.567	"1234.567"	
	%8.6g	SF 1234.567	"␣1234.57"	
Specification of zero padding	%05d	K 100	"00100"	When the setting for the display digit is available, zero padding can be specified. Put zero (0) before the display digit.
	%05x	H 12A	"0012a"	
	%05b	H 123	"00123"	
	%08.3f	SF 123.4567	"0123.457"	
	%010.3e	SF 1234.567	"01.235e+03"	

<table>
<tr><th>Items</th><th>Control character string</th><th>Binary data</th><th>ASCII data</th><th colspan="2">Description</th></tr>
<tr><td rowspan="5">Specification of right align and left align</td><td>%-5d</td><td>K 100</td><td>"100␣␣"</td><td rowspan="5" colspan="2">Default is right align. To set to left align, add minus (-) before the specification of digit number.</td></tr>
<tr><td>%-5x</td><td>H 12A</td><td>"12a␣␣"</td></tr>
<tr><td>%-5b</td><td>H 123</td><td>"123␣␣"</td></tr>
<tr><td>%-8.3f</td><td>SF 123.4567</td><td>"123.457␣"</td></tr>
<tr><td>%-010.3e</td><td>SF 1234.567</td><td>"1.235e+03␣"</td></tr>
<tr><td rowspan="5">Specification of sign</td><td>%+d</td><td>K 100</td><td>"+100"</td><td rowspan="5" colspan="2">A plus sign (+) is not added by default.
To add a plus sign (+), add (+). Not available when %u, %b, or %x is specified.</td></tr>
<tr><td>%+d</td><td>K -100</td><td>"-100"</td></tr>
<tr><td>%+5d</td><td>K 100</td><td>"␣+100"</td></tr>
<tr><td>%+8.3f</td><td>123.4567</td><td>"+123.457"</td></tr>
<tr><td>%+10.3e</td><td>1234.567</td><td>"+1.234e+03"</td></tr>
<tr><td rowspan="6">Specification of numerical position</td><td>%␣d</td><td>K 100</td><td>"␣100"</td><td rowspan="6" colspan="2">In the case of a positive number, a space is added to align the position in the case of a negative number. Add a space to align the position.</td></tr>
<tr><td>%␣d</td><td>K -100</td><td>"-100"</td></tr>
<tr><td>%␣8.3f</td><td>SF 123.4567</td><td>"␣123.457"</td></tr>
<tr><td>%␣8.3f</td><td>SF -123.4567</td><td>"-123.457"</td></tr>
<tr><td>%␣10.3e</td><td>SF 1234.567</td><td>"␣1.235e+03"</td></tr>
<tr><td>%␣10.3e</td><td>SF -1234.567</td><td>"-1.235e+03"</td></tr>
<tr><td rowspan="7">Specification of another output format for numerical data type</td><td>%#x</td><td>H 12A</td><td>"0x12a"</td><td>"0x" is added.</td><td rowspan="7">Another output type is automatically given by adding "#".</td></tr>
<tr><td>%#X</td><td>H 12A</td><td>"0X12A"</td><td>"0X" is added.</td></tr>
<tr><td>%#8.0f</td><td>SF 123.4567</td><td>"␣␣␣␣123."</td><td rowspan="3">"." is always added.</td></tr>
<tr><td>%#10.0e</td><td>SF 1234.567</td><td>"␣␣␣␣1.e+03"</td></tr>
<tr><td>%#10.3E</td><td>SF 1234.567</td><td>"␣␣␣␣1.E+03"</td></tr>
<tr><td>%#9.0g</td><td>SF 1234</td><td>"␣␣␣1234.0"</td><td rowspan="2">"." is always added, and "0" after the decimal point is not omitted.</td></tr>
<tr><td>%#.9G</td><td>SF 1234</td><td>"1234.0000"</td></tr>
<tr><td rowspan="4">Specification of additional characters after numerical data</td><td>%d,</td><td>K 100</td><td>"100,"</td><td rowspan="4" colspan="2">The characters that follow the characters specified for conversion (d , x , b , f , e , g) are added after the numerical ASCII data.</td></tr>
<tr><td>%x,</td><td>H 100</td><td>"100,"</td></tr>
<tr><td>%xH</td><td>H 100</td><td>"100H"</td></tr>
<tr><td>%bBCD</td><td>H 100</td><td>"100BCD"</td></tr>
</table>

(Note 1) "␣" in the table represents a space.

(Note 2) For exponential notation, it consists of a code (e or E), a sign, and a 2-digit number.

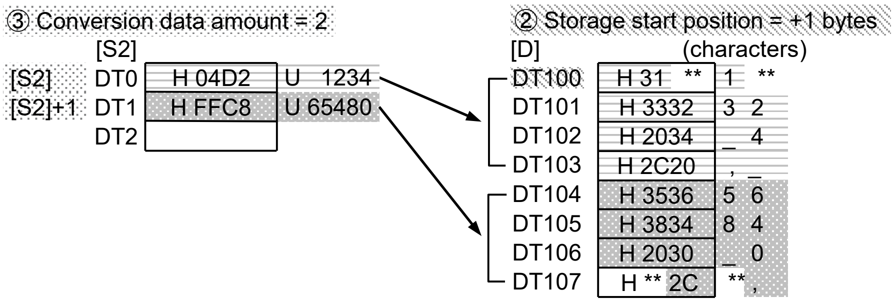
(Note 3) If the conversion results in having fewer enabled digits than before conversion, the result is rounded off.

■ Conversion example

Example 1) Converting unsigned 16-bit binary data (2 data) to decimal ASCII data (6 digits + comma) x 2

The high byte of DT100 is set as the beginning of the storage area. It is left-aligned (low word side), and a comma is added before the data is stored. If it is fewer than 6 digits, spaces are inserted.

[i]…US
[S1]…"%-6u," : Convert 16-bit data into decimal ASCII data (6 digits)
[S2]…DT0
[N]…H 0002 01 01 : ① ASCII data (reverse direction)
[D]…DT100 ② Storage start position (1) → [D]+1 bytes
③ Conversion data amount (2) → Convert [S2] and [S2]+1

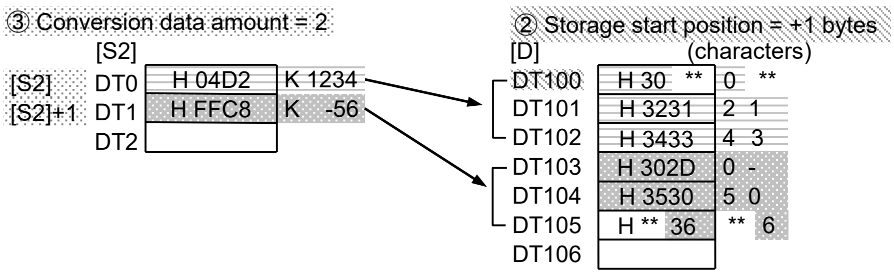


DT0 : U 1234 (H 04D2) → DT100 to DT103 : "1234 ,"	
DT1 : U 65480 (H FFC8) → DT104 to DT107 : "65480 ,"	

Example 2) Converting signed 16-bit binary data (2 data) to decimal ASCII data (5 digits x 2)

The high byte of DT100 is set as the beginning of the storage area. Zero padding is used to store the data.

[i]…SS
[S1]…"%05d" : Convert 16-bit data into decimal ASCII data (5 digits) (0 padding)
[S2]…DT0
[N]…H 0002 01 01 : ① ASCII data (reverse direction)
[D]…DT100 ② Storage start position (1) → [D]+1 bytes
③ Conversion data amount (2) → Convert [S2] and [S2]+1

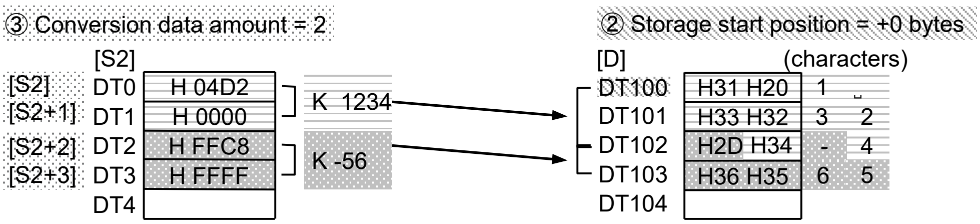


DT0: K 1234 (H 04D2) → DT100 to DT102: "01234"	
DT1: K -56 (H FFC8) → DT103 to DT105: "-0056"	

Example 3) Converting signed 32-bit binary data (2 data) to decimal ASCII data (number of digits not specified)

The low byte of DT100 is set as the beginning of the storage area. For a positive number, a space is inserted. Since the number of digits is not specified, the number of characters after conversion and the size of the area where the data is stored will vary according to the value of the binary data that is converted.

[i]…SL
[S1]…"%␣d" : Convert 32-bit data into decimal ASCII data (For a positive number, a space is inserted)
[S2]…DT0
[N]…H 0002 00 01 : ① ASCII data (reverse direction)
[D]…DT100
② Storage start position (0) → [D]+0 bytes
③ Conversion data amount (2) → Convert [S2] and [S2]+2

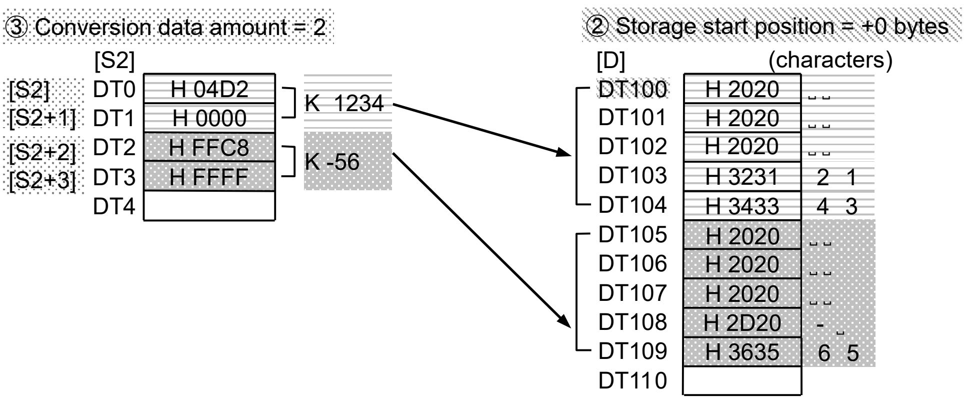


DT0 to DT1: U 1234 (H 0000 04D2) → DT100 to DT102: "␣1234 "	
DT2 to DT3: U 4294967240 (H FFFF FFC8) → DT102 to DT103: "-56"	

Example 4) Converting signed 32-bit binary data (2 data) to decimal ASCII data (10 digits x 2)

The low byte of DT100 is set as the beginning of the storage area. It is right-aligned (high word side), and a space is inserted before the data is stored.

[i]…SL
[S1]…"%10d" : Convert 32-bit data into decimal ASCII data (10 digits)
[S2]…DT0
[N]…H 0002 00 01 : ① ASCII data (reverse direction)
[D]…DT100
② Storage start position (0) → [D]+0 bytes
③ Conversion data amount (2) → Convert [S2] and [S2]+2

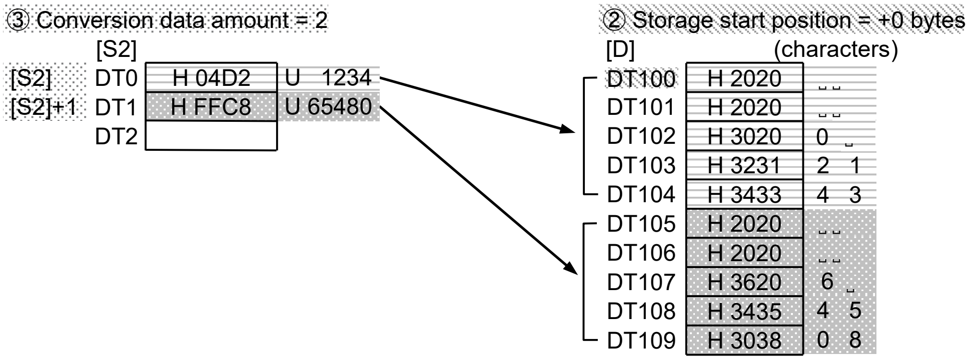


DT0 to DT1: U 1234 (H 0000 04D2) → DT100 to DT104: "␣␣␣␣␣␣ 1234"	
DT2 to DT3: U 4294967240 (H FFFF FFC8) → DT105 to DT109: "␣␣␣␣␣␣␣ -56"	

Example 5) Converting unsigned 16-bit binary data (2 data) to decimal ASCII data (10 digits x 2)

The low byte of DT100 is set as the beginning of the storage area. 5 digits are stored as significant figures. If the control string is "u", the sign "+" is not output.

[i]…US
[S1]…"%+10.5u" : Convert 16-bit data into decimal ASCII data (10 digits)
[S2]…DT0
[N]…H 0002 00 01 : ①ASCII data (reverse direction)
[D]…DT100 ②Storage start position (0) → [D]+0 bytes
③Conversion data amount (2) → Convert [S2] and [S2]+1

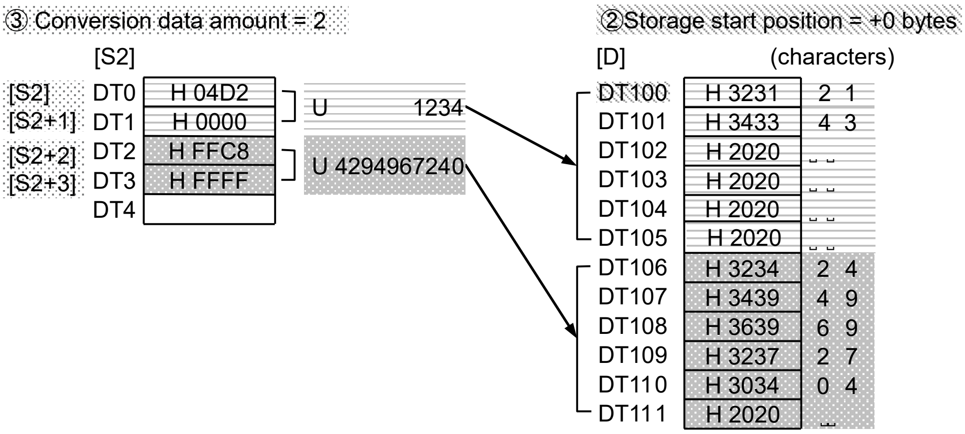


DT0: U 1234 (H 04D2) →	DT100 to DT104: "␣␣␣␣ 01234"
DT1: U 65480 (H FFC8) →	DT105 to DT109: "␣␣␣␣ 65480"

Example 6) Converting unsigned 32-bit binary data (2 data) to decimal ASCII data (12 digits x 2)

The low byte of DT100 is set as the beginning of the storage area. It is stored left-aligned (low word side). Spaces are inserted.

[i]…UL
[S1]…"%-12u" : Convert 32-bit data into decimal ASCII data (12 digits) (left align)
[S2]…DT0
[N]…H 0002 00 01 : ①ASCII data (reverse direction)
[D]…DT100 ②Storage start position (0) → [D]+0 bytes
③Conversion data amount (2) → Convert [S2] and [S2]+2

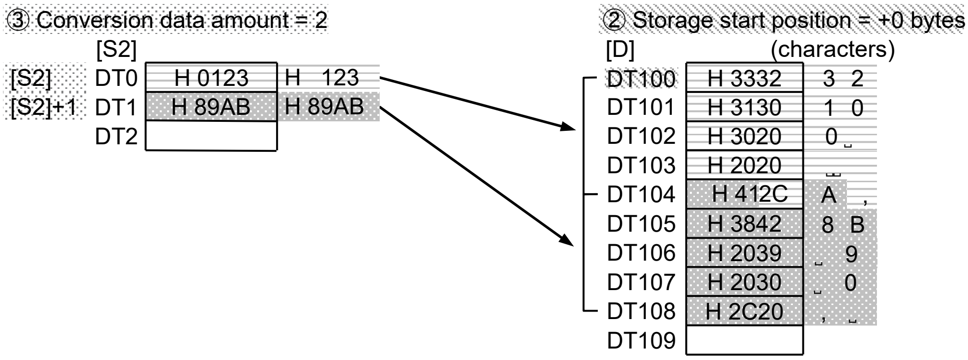


DT0 to DT1: U 1234 (H 0000 04D2)	→	DT100 to DT105: "1234␣␣␣␣␣␣␣␣"	
DT2 to DT3: U 4294967240 (H FFFF FFC8)	→	DT106 to DT110: "4294967240␣␣"	

Example 7) Converting unsigned 16-bit binary data (2 data) to hexadecimal ASCII data (8 digits + comma) x 2

The low byte of DT100 is set as the beginning of the storage area. It is stored left-aligned (low word side) for 5 significant figures. Spaces and a comma are inserted. Hex data is in upper case.

[i]…US
[S1]…"%8.5X, " : Convert 16-bit data into hexadecimal ASCII data (8 digits)
[S2]…DT0
[N]…H 0002 00 00 : ① ASCII data (forward direction)
[D]…DT100
② Storage start position (0) → [D]+0 bytes
③ Conversion data amount (2) → Convert [S2] and [S2]+1

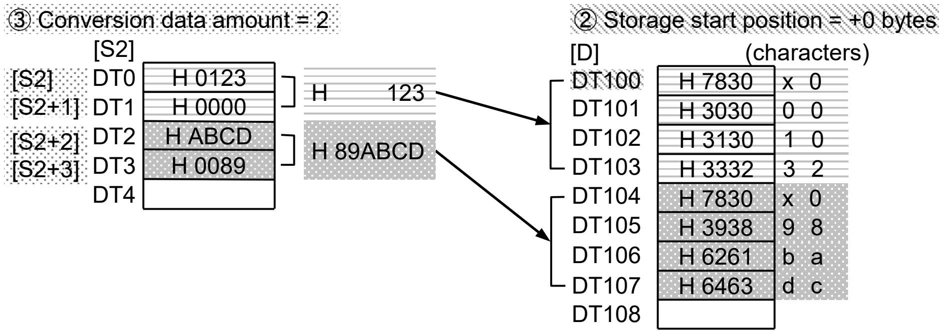


DT0: H 0123 → DT100 to DT104: "2301␣ 0␣␣ ,"
DT1: H 89AB → DT104 to DT108: "AB89␣ 0␣␣ ,"

Example 8) Converting unsigned 32-bit binary data (2 data) to hexadecimal ASCII data (8 digits x 2)

The low byte of DT100 is set as the beginning of the storage area. It is stored right-aligned (high word side) for 6 significant figures. The characters "0x" that represent Hex data are added.

[i]…UL
[S1]…"%#8.6x" : Convert 32-bit data into hexadecimal ASCII data (8 digits) (add "0x")
[S2]…DT0
[N]…H 0002 00 01 : ① ASCII data (reverse direction)
[D]…DT100
② Storage start position (0) → [D]+0 bytes
③ Conversion data amount (2) → Convert [S2] and [S2]+2

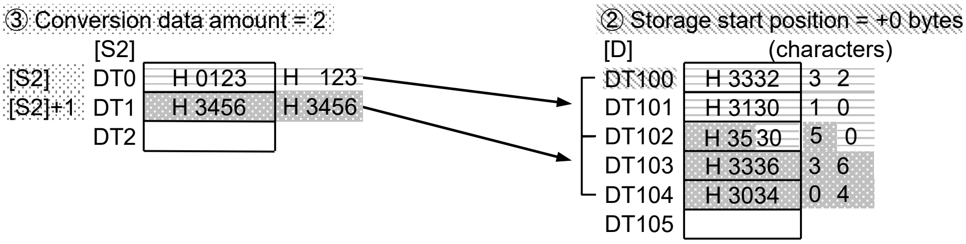


DT0 to DT1: H 0123 → DT100 to DT103: "0x000123"
DT2 to DT3: H 89AB → DT104 to DT107: "0x89abcd"

Example 9) Converting unsigned 16-bit BCD data (2 data) to decimal ASCII data (5 digits x 2)

The low byte of DT100 is set as the beginning of the storage area. It is stored left-aligned (low word side) for 5 significant figures. Zero padding is used.

[i]…US
[S1]…"%05b" : Convert 16-bit BCD data into decimal ASCII data (5 digits) (zero padding)
[S2]…DT0
[N]…H 0002 00 00 : ① ASCII data (forward direction)
[D]…DT100 ② Storage start position (0) → [D]+0 bytes
③ Conversion data amount (2) → Convert [S2] and [S2]+1

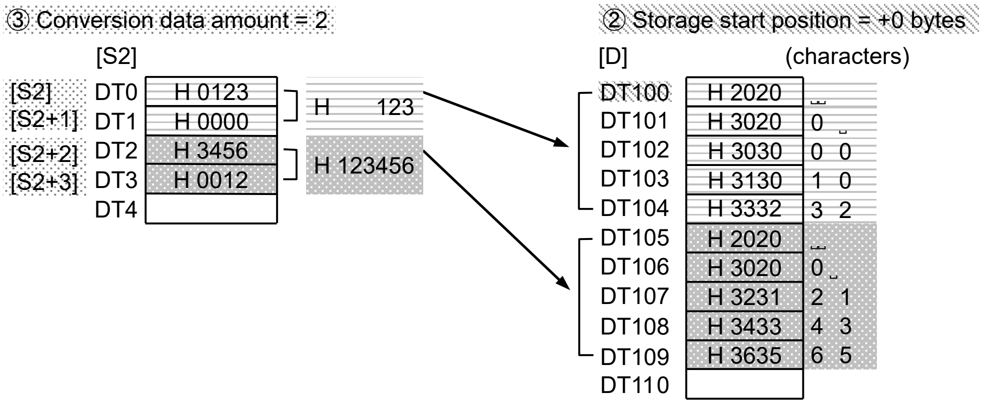


DT0: H 0123 →	DT100 to DT102: "23010" is the data that is converted from H0123.
DT1: H 3456 →	DT102 to DT104: "56340" is the data that is converted from H3456.

Example 10) Converting unsigned 32-bit BCD data (2 data) to decimal ASCII data (10 digits x 2)

The low byte of DT100 is set as the beginning of the storage area. It is stored right-aligned (high word side). Zero padding is used within the 7 significant figures. Spaces are inserted for the remaining digits.

[i]…UL
[S1]…"%10.7b" : Convert 32-bit BCD data into decimal ASCII data (10 digits)
[S2]…DT0
[N]…H 0002 00 01 : ① ASCII data (reverse direction)
[D]…DT100 ② Storage start position (0) → [D]+0 bytes
③ Conversion data amount (2) → Convert [S2] and [S2]+2

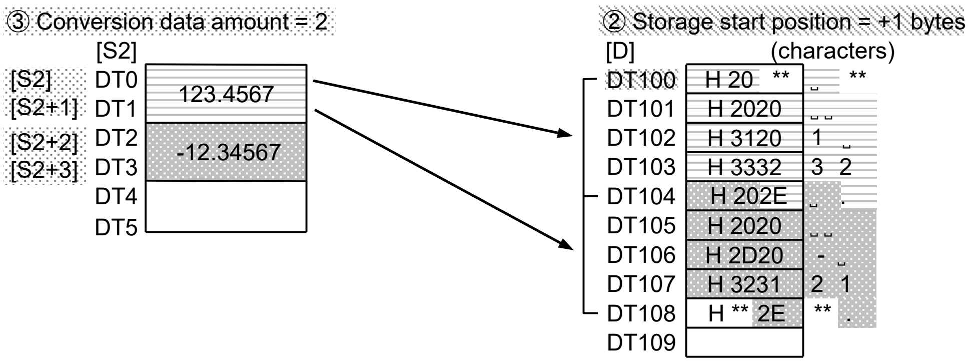


DT0 to DT1: H 0000 0123	→ DT100 to DT104: "␣␣ 0000123"
DT2 to DT3: H 0012 3456	→ DT105 to DT109: "␣␣ 0123456"

Example 11) Converting 32-bit single-precision floating point real number data (2 data) to floating point ASCII data (8 digits x 2)

The high byte of DT100 is set as the beginning of the storage area. It is stored right-aligned (high word side). A decimal point is added followed by no digit.

[i]…SF
[S1]…"%#8.0f" : Convert 32-bit single precision real number data into floating point ASCII data (8 digits)
[S2]…DT0
[N]…H 0002 01 01 : ① ASCII data (reverse direction)
[D]…DT100 ② Storage start position (1) → [D]+1 bytes
③ Conversion data amount (2) → Convert [S2] and [S2]+2

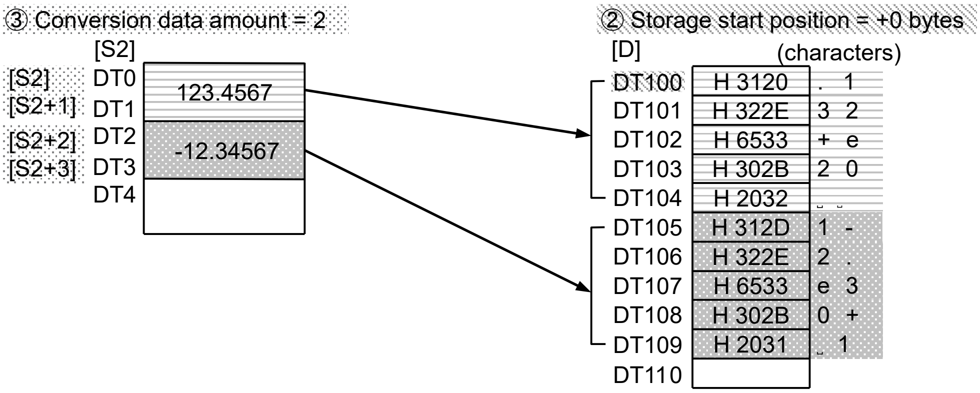


DT0 to DT3: 123.4567	→ DT100 to DT104: "␣␣␣ 123."
DT4 to DT7: -12.34567	→ DT104 to DT108: "␣␣␣ -12."

Example 12) Converting 32-bit single-precision floating point real number data (2 data) to exponential notation ASCII data (10 digits x 2)

The low byte of DT100 is set as the beginning of the storage area. 2 digits after the decimal point. Exponential notation is used. For a positive number, a space is inserted.

[i]…SF
[S1]…"%-10.2e" : Convert 32-bit single precision real number data into exponential ASCII data (10 digits)
[S2]…DT0
[N]…H 0002 00 01 : ① ASCII data (reverse direction)
[D]…DT100 ② Storage start position (0) → [D]+0 bytes
③ Conversion data amount (2) → Convert [S2] and [S2]+2



DT0 to DT3: 123.4567	→	DT100 to DT104: "1.23e+02␣␣"
DT4 to DT7: -12.34567	→	DT105 to DT109: "-1.23e+01␣"

Example 13) Converting 32-bit single-precision floating point real number data (2 data) to floating point ASCII data or exponential notation ASCII data (9 digits x 2)

The high byte of DT100 is set as the beginning of the storage area. It is stored left-aligned (low word side). The data is converted to 7 significant figure floating point ASCII data before storing.

Note

- The conversion is either to floating point ASCII data or to exponential notation ASCII data, whichever is shorter.

[i]…SF

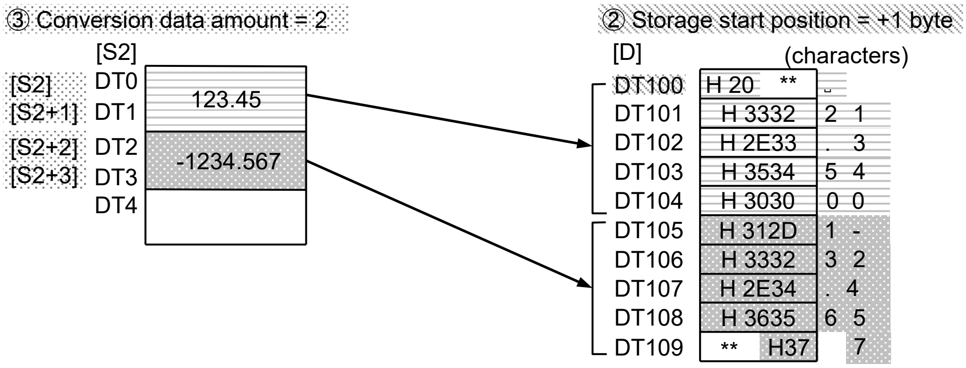
[S1]…"%#9.7g" : Convert 32-bit single precision real number data into exponential ASCII data (9 digits) or floating point ASCII data, whichever is shorter in the relevant notation

[S2]…DT0

[N]…H 0002 01 01 : ① ASCII data (reverse direction)

[D]…DT100 ② Storage start position (0) → [D]+1 byte

③ Conversion data amount (2) → Convert [S2] and [S2]+2



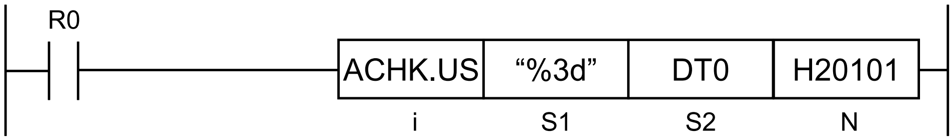
DT0 to DT1: 123.4567	→ DT100 to DT104: " ˽ 123.4500"
DT2 to DT3: -12.34567	→ DT104 to DT108: " -1234.567"

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the conversion format specified by [S1] is not an available operation unit.
	To be set when the conversion format specified by [S1] is not a control character.
	To be set when the number of ASCII code digits specified by [N] exceeds the maximum number of digits for the control characters specified by [S1].
	To be set when the storage start position specified by [N] is out of the range.
	To be set when the conversion data amount specified by [N] exceeds the [S2] area.
	To be set when the conversion data amount specified by [N] is out of the range.
	To be set when the conversion result exceeds the ASCII code storage area specified by [N].
	To be set when the conversion result exceeds the area specified by [D].

14.10 ACHK (ASCII Data Check)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Control string (2 to 16 characters)
S2	Starting number of the area that stores an ASCII code
N	Conversion method

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●	●	●	●	●													●
N (Note 1)	●	●	●	●			●	●								●	●				●

(Note 1) To be handled as a 32-bit integer (UL), regardless of operation unit.

■ Outline of operation

- This instruction checks whether the specified ASCII code string can be converted using the ATOB instruction.
- According to the control characters specified by [S1], this instruction checks whether the ASCII code stored in the area specified by [S2] can be normally converted using the conversion method specified by [N].
- If the check result is normal, the system relay SRB turns ON (1), and if abnormal, SRB turns OFF (0).
- Specify the same value as the ATOB instruction for [S1] the control string, [S2] the beginning of the source data, and [N] the conversion method.

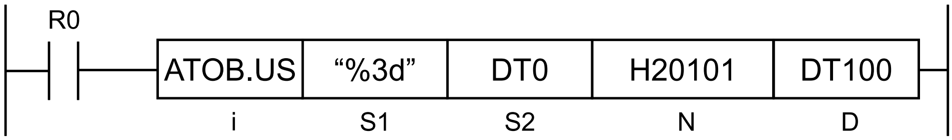
- The operations for the maximum length of ASCII string and for the valid range of conversion data values are the same as the ATOB instruction.
- To directly specify a character constant for the operand [S1], enter the constant in the "Instruction list" dialog box in the FPWIN GR7 programming tool.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the conversion format specified by [S1] is not an available operation unit.
	To be set when the conversion format specified by [S1] is not a control character.
	To be set when the number of ASCII code digits specified by [N] exceeds the maximum number of digits for the control characters specified by [S1].
	To be set when the storage start position specified by [N] is out of the range.
	To be set when the conversion data amount specified by [N] exceeds the [S2] area.
	To be set when the conversion data amount specified by [N] is out of the range.

14.11 ATOB (Conversion: ASCII → BIN)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ List of operands

Operand	Description
S1	Control string (2 to 16 characters)
S2	Starting number of the area that stores the ASCII data targeted for conversion
N	Conversion method
D	Starting number of the area for storing the binary data of the converted result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●	●	●	●	●													●
N(Note 1)	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

(Note 1) To be handled as a 32-bit unsigned integer, regardless of operation unit.

■ Outline of operation

- This instruction converts the ASCII code string data stored in the area starting with [S2] into binary data.
- Specify the conversion method for [S1] and [N].
- The conversion result is stored in the area starting with [D].

■ Format of ASCII data

- Data is processed for the number of data specified for [N], or considering a NULL character as the end of the ASCII string data to be converted.
- If a comma (",") is included in the string data, it is processed as a separator for the data.
- The maximum number of digits for a single datum is 28.

Format	ASCII data before conversion	Binary data after conversion
Decimal integer	" ␣␣␣123456"	U 123456
	"123456␣␣ "	U 123456
Hexadecimal integer	"12AB"	H 12AB
	" ␣␣␣12AB "	H 12AB
	"00000012AB "	H 12AB
	"0x12AB",	H 12AB
Floating point real number	"1234.56789",	f 1234.56
	" 1234"	f 1234.00
Exponential notation Real number	"1.234E+2"	f 123.4
	"1.23E-2 ",	f 0.01234
	"␣␣␣1.234e+2",	f 123.4

■ Setting of control string [S1]

Control character string	Format of data for conversion		Available operation units	Data range
	ASCII data before conversion	BIN data after conversion		
"%nd"	Unsigned decimal ASCII data	Unsigned 16-bit data	US	U0 to U65,535
	Signed decimal ASCII data	Signed 16-bit data	SS	K−32,768 to K32,767
	Unsigned decimal ASCII data	Unsigned 32-bit data	UL	U0 to U4,294,967,295
	Signed decimal ASCII data	Signed 32-bit data	SL	K-2,147483,648 to K2,147,483,647
"%nx"	Hexadecimal ASCII data	16-bit data (Forward/ reverse direction)	US	H0 to HFFFF
	Hexadecimal ASCII data	32-bit data (Forward/ reverse direction)	UL	H0 to HFFFFFFFF
"%nb"	Decimal ASCII data	16-bit BCD data (Forward/Reverse direction)	US	H0 to H9999
	Decimal ASCII data	32-bit BCD data (Forward/Reverse direction)	UL	H0 to H99999999
"%nf" "%ne"	Floating point real number (including exponential notation) ASCII data	32-bit single-precision floating point real number data	SF	3.402823E+38 to 1.175494E-38
				-1.175494E-38 -3.402823E+38

Control character string	Format of data for conversion		Available operation units	Data range
	ASCII data before conversion	BIN data after conversion		
	Floating point real number (including exponential notation) ASCII data	64-bit double-precision floating point real number data	DF	1.797693134862231E+308 to 2.2250738585072014E-308 -2.2250738585072014E-308 to -1.797693134862231E+308

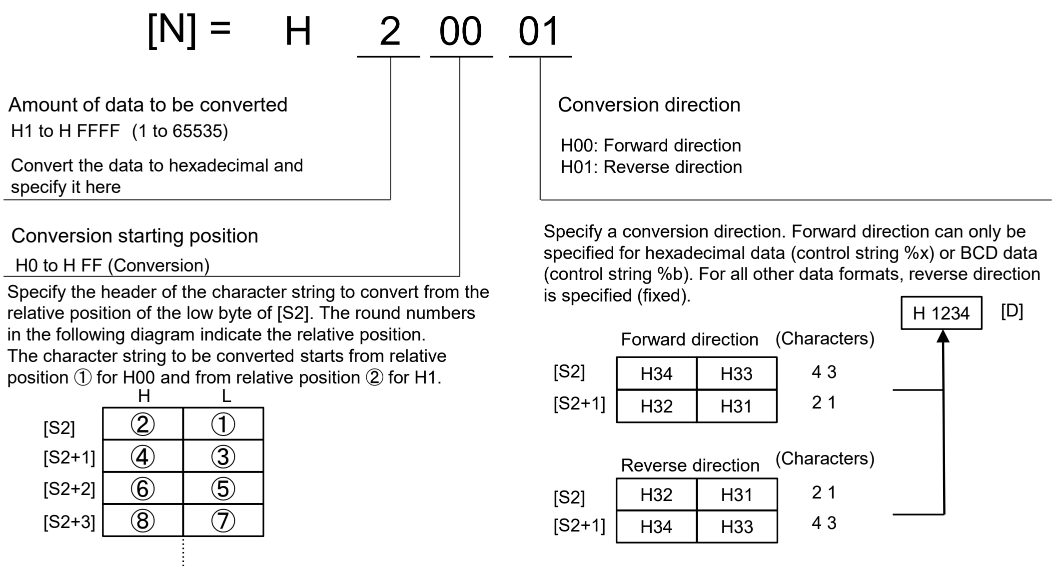
(Note 1) For "n", specify the number of digits. Conversion is performed for every "n" digits. However, data separation occurs wherever a comma (",") appears.

(Note 2) When omitting "n", insert a comma (",") to separate the data.

(Note 3) When there is a comma followed by another comma, it is converted to the value "0". Example: When the number of the data is three, ",," is processed as the data "0, 0, and 0".

■ Specification of conversion method [N]

Conversion method [N] should be specified in a Hex format (8 digits) in the 32-bit area.



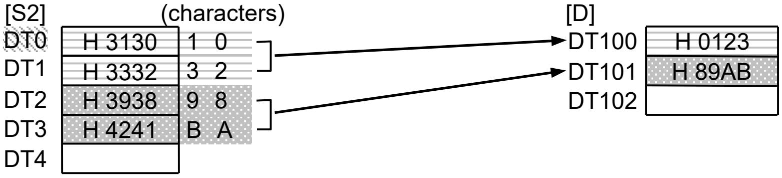
■ Conversion example

Example 1) Converting two hexadecimal ASCII data (4 digits) to two 16-bit binary data (hexadecimal)

The conversion starts from the low byte of DT0. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data).

[i]…US
[S1]…"%4x" : Converts hexadecimal ASCII data (4-digit) to 16-bit data
[S2]…DT0
[N]…H 0002 00 01 : ① ASCII data (reverse direction)
[D]…DT100 ② Conversion starting position (0) → [S2]+0 byte
③ Amount of conversion data (2) → Converts two 4-digit data

② Conversion starting position = +0 byte
③ Amount of data to be converted = 2



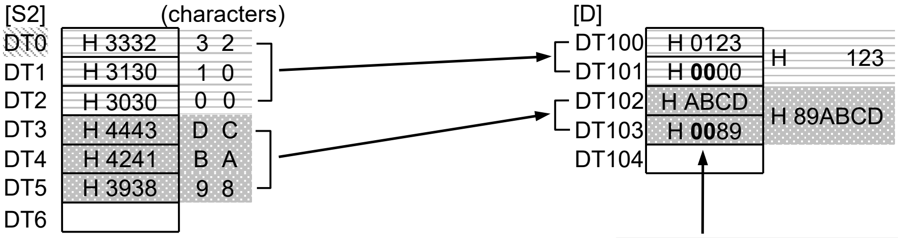
DT0 to DT1: "0123"	→ DT100: H 0123
DT2 to DT3: "89AB"	→ DT101: H 89AB

Example 2) Converting two hexadecimal ASCII data (6 digits) to two 32-bit binary data (hexadecimal)

The conversion starts from the low byte of DT0. It is converted in forward direction (the high word side of [S2] is considered as high-order numerical data). For empty digits of the storage area, zeros (0) are inserted.

[i]…UL
[S1]…"%6x" : Converts hexadecimal ASCII data (6-digit) to 32-bit data
[S2]…DT0
[N]…H 0002 00 00 : ① ASCII data (forward direction)
[D]…DT100 ② Conversion starting position (0) → [S2]+0 byte
③ Amount of conversion data (2) → Converts two 6-digit data

② Conversion starting position = + 0 byte
③ Amount of data to be converted = 2



* When the number of characters is smaller than the number of converted bits, the portions that are lacking are padded with “0”.

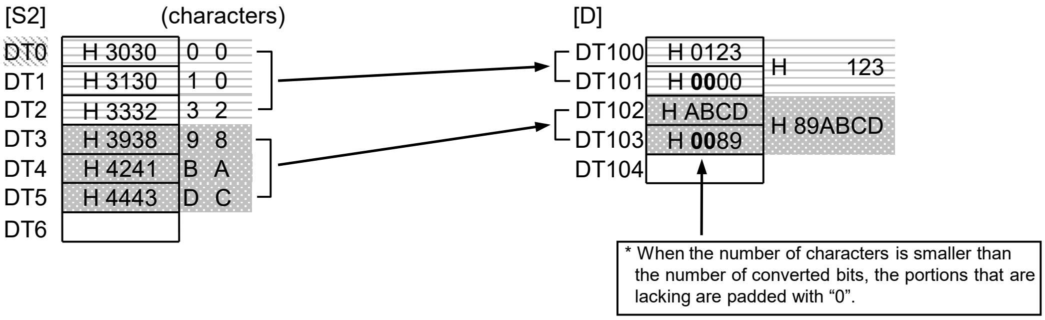
DT0 to DT2: "000123"	→ DT100: H 0000 0123
DT3 to DT5: "89ABCD"	→ DT101: H 0089 ABCD

Example 3) Converting two hexadecimal ASCII data (6 digits) to two 32-bit binary data (hexadecimal)

The conversion starts from the low byte of DT0. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data). For empty digits of the storage area, zeros (0) are inserted.

[i]…UL
[S1]…"%6x" : Converts hexadecimal ASCII data (6-digit) to 32-bit data
[S2]…DT0
[N]…H 0002 00 01 : ① ASCII data (reverse direction)
[D]…DT100 ② Conversion starting position (0) → [S2]+0 byte
③ Amount of conversion data (2) → Converts two 6-digit data

② Conversion starting position = + 0 byte
③ Amount of data to be converted = 2



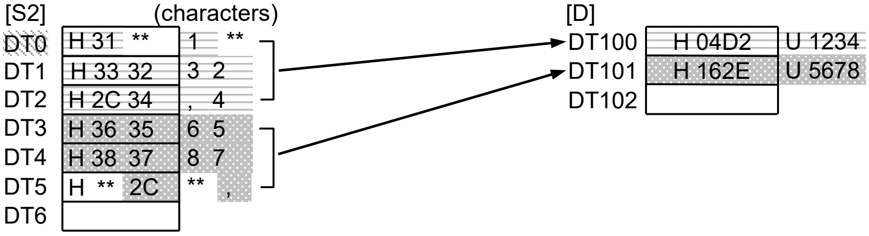
DT0 to DT2: "000123" → DT100: H 0000 0123
DT3 to DT5: "89ABCD" → DT101: H 0089 ABCD

Example 4) Converting two hexadecimal ASCII data (separated by commas) to two 10-bit binary data (decimal)

The conversion starts from the high byte of DT0. If there is no specification for the number of digits, it is processed considering commas as the data delimiter. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data).

[i]…US
[S1]…"%d" : Converts decimal ASCII data to 16-bit data (Data end: ',')
[S2]…DT0
[N]…H 0002 01 01 : ① ASCII data (reverse direction)
[D]…DT100
② Conversion starting position (1) → [S2]+1 bytes
③ Amount of conversion data (2) → 2 data separated by a comma

② Conversion starting position = +1 byte
③ Amount of data to be converted = 2



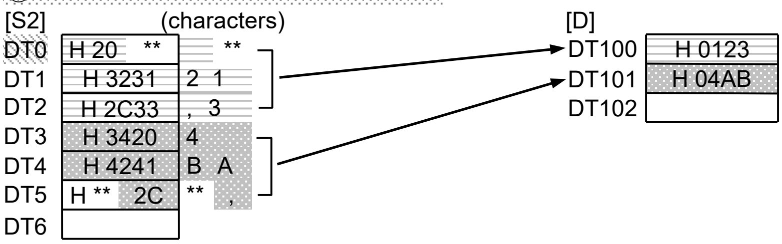
DT0 to DT2: "1234," → DT100: U 1234 (H 04D2)
DT3 to DT5: "5678," → DT101: U 5678 (H 162E)

Example 5) Converting two hexadecimal ASCII data (separated by commas) to two 16-bit binary data (hexadecimal)

The conversion starts from the high byte of DT0. If there is no specification for the number of digits, it is processed considering commas as the data delimiter. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data).

[i]…US
[S1]… "%x" : Converts hexadecimal ASCII data to 16-bit data (Data end: ',')
[S2]…DT0
[N]…H 0002 01 01 : ① ASCII data (reverse direction)
[D]…DT100
② Conversion starting position (1) → [S2]+1 byte
③ Amount of conversion data (2) → 2 data separated by a comma

② Conversion starting position = + 1 byte
③ Amount of data to be converted = 2



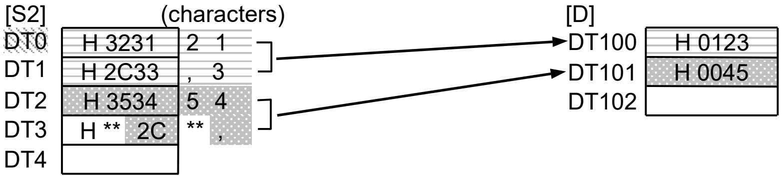
DT0 to DT2: "123," → DT100: H 0123
DT3 to DT5: "4AB," → DT101: H 04AB

Example 6) Converting two decimal ASCII data (4 digits) to two 16-bit BCD data (if there is a comma at the end of the data)

The conversion starts from the low byte of DT0. If the data includes a comma, it is processed as a data delimiter even if the number of digits is less than specified. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data).

[i]…US
[S1]… "%4b" : Converts decimal ASCII data to 16-bit BCD data (Data end: ',')
[S2]…DT0
[N]…H 0002 00 01 : ① ASCII data (reverse direction)
[D]…DT100
② Conversion starting position (0) → [S2]+0 byte
③ Amount of conversion data (2) → 2 data separated by a comma

② Conversion starting position = +0 byte
③ Amount of data to be converted = 2



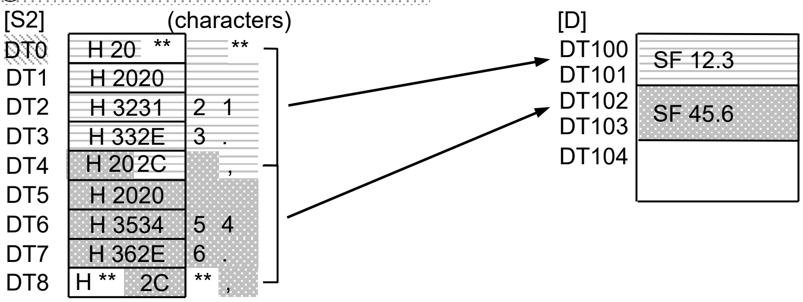
DT0 to DT1: "123," → DT100: U 0291 (H 0123)	
DT2 to DT3: "45," → DT101: U 0069 (H 0045)	

Example 7) Converting two floating point ASCII data (with comma delimiters) to two 32-bit single-precision real numbers

The conversion starts from the high byte of DT0. If there is no specification for the number of digits, it is processed considering commas as the data delimiter. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data).

[i]…SF
[S1]… "%f" : Converts floating point ASCII data to 32-bit real number data (Data end: ',')
[S2]…DT0
[N]…H 0002 01 01 : ① ASCII data (reverse direction)
[D]…DT100
② Conversion starting position (1) → [S2]+1 byte
③ Amount of data to be converted (2) → 2 data separated by a comma

② Conversion starting position = +1 byte
③ Amount of data to be converted = 2

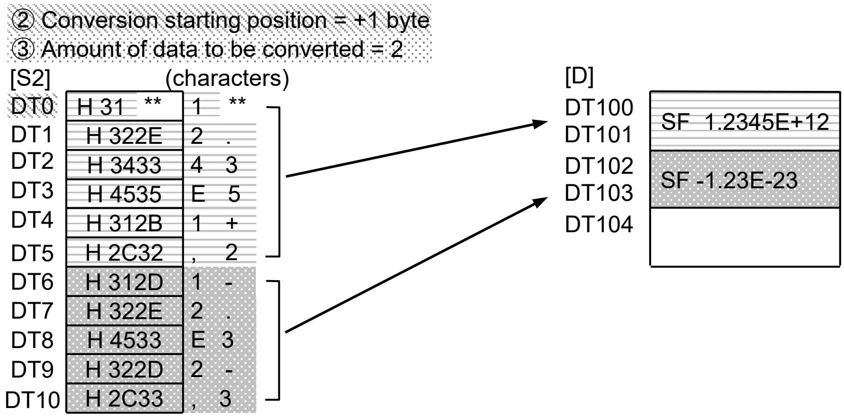


DT0 to DT4: " 12.3," → DT100 to DT101: SF 12.3	
DT4 to DT8: " 45.6," → DT102 to DT103: SF 45.6	

Example 8) Converting two exponential notation ASCII data (with comma delimiters) to two 32-bit single-precision real numbers

The conversion starts from the high byte of DT0. If there is no specification for the number of digits, it is processed considering commas as the data delimiter. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data).

[i]…SF
[S1]…"%e" : Converts exponential notation ASCII data to 32-bit single-precision real number data (Data end: ',')
[S2]…DT0
[N]…H 00020101 : ① ASCII data (reverse direction)
[D]…DT100 ② Conversion starting position (1) → [S2]+1 byte
③ Amount of conversion data (2) → 2 data separated by a comma



DT0 to DT5: "1.2345E+12,"	→ DT100 to DT101: SF 1.2345E+12
DT6 to DT10: "-1.23E-23,"	→ DT102 to DT103: SF -1.23E-23

Example 9) Converting two decimal ASCII data (5 digits) to two 16-bit unsigned binary data (decimal)

The conversion starts from the high byte of DT0. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data).

[i]…US
[S1]… "%5d" : Converts decimal ASCII data (5-digits) to 16-bit data (unsigned)
[S2]…DT0
[N]…H 0002 01 01 : ① ASCII data (reverse direction)
[D]…DT100
② Conversion starting position (1) → [S2]+1 byte
③ Amount of conversion data (2) → two 5-digit data

② Conversion starting position = +1 byte
③ Amount of data to be converted = 2

[S2]		(characters)
DT0	H 20 **	**
DT1	H 3231	2 1
DT2	H 3433	4 3
DT3	H 3536	5 6
DT4	H 3834	8 4
DT5	H ** 30	** 0
DT6		

[D]		
DT100	H 04D2	U 1234
DT101	H FFC8	U 65480
DT102		

DT0 to DT2: " 1234" → DT100: U 1234 (H 04D2)
DT3 to DT5: "65480" → DT101: U 65480 (H FFC8)

Example 10) Converting two decimal ASCII data (5 digits) to two 16-bit signed binary data (decimal)

The conversion starts from the high byte of DT0. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data).

[i]…SS
[S1]…"%5d" : Converts decimal ASCII data (5-digit) to 16-bit data
[S2]…DT0
[N]…H 0002 01 01 : ① ASCII data (reverse direction)
[D]…DT100
② Conversion starting position (1) → [S2]+1 byte
③ Amount of conversion data (2) → two 5-digit data

② Conversion starting position = +1 byte
③ Amount of data to be converted = 2

[S2]		(characters)
DT0	H 20 **	**
DT1	H 3231	2 1
DT2	H 3433	4 3
DT3	H 2020	
DT4	H 352D	5 -
DT5	H ** 36	** 6
DT6		

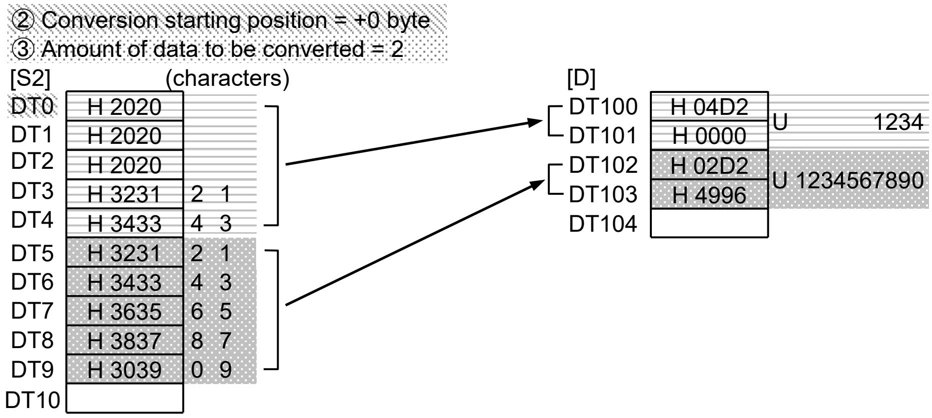
[D]		
DT100	H 04D2	K 1234
DT101	H FFC8	K -56
DT102		

DT0 to DT2: " 1234" → DT100: K 1234 (H 04D2)
DT3 to DT5: " -56" → DT101: K -56 (H FFC8)

Example 11) Converting two decimal ASCII data (10 digits) to two 32-bit unsigned binary data (decimal)

The conversion starts from the low byte of DT0. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data).

[i]…UL
[S1]…"%10d" : Converts decimal ASCII data (10-digit) to 32-bit data
[S2]…DT0
[N]…H 0002 00 01 : ① ASCII data (reverse direction)
[D]…DT100
② Conversion starting position (0) → [S2]+0 byte
③ Amount of conversion data (2) → two 10-digit data

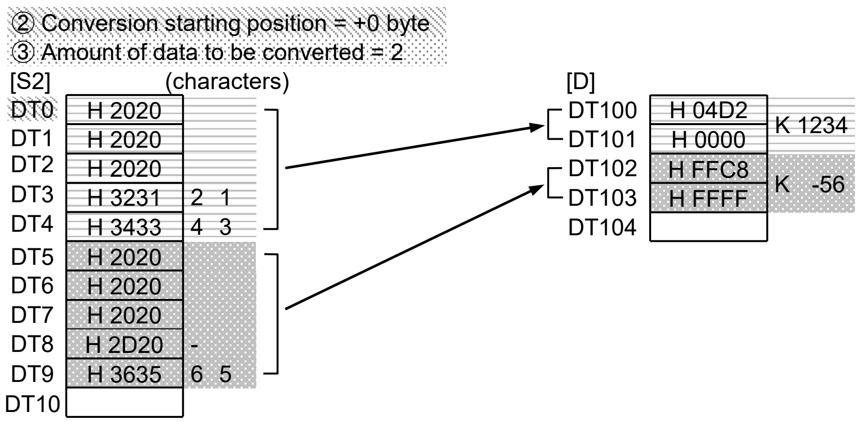


DT0 to DT4: " 1234" → DT100 to DT101: U	1234 (H 0000 04D2)
DT5 to DT9: "1234567890" → DT102 to DT103: U	1234567890 (H 4996 02D2)

Example 12) Converting two decimal ASCII data (10 digits) to two 32-bit signed binary data (decimal)

The conversion starts from the low byte of DT0. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data).

[i]…SL
[S1]…"%10d" : Converts decimal ASCII data (10-digit) to 32-bit data
[S2]…DT0
[N]…H 0002 00 01 : ① ASCII data (reverse direction)
[D]…DT100
② Conversion starting position (0) → [S2]+0 byte
③ Amount of conversion data (2) → two 10-digit data



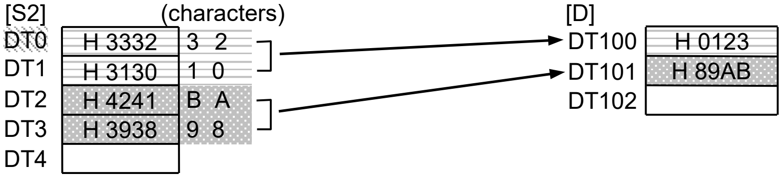
DT0 to DT4: " 1234" → DT100 to DT101:	K 1234
DT5 to DT9: " -56" → DT102 to DT103:	K -56

Example 13) Converting two hexadecimal ASCII data (4 digits) to two 16-bit binary data (hexadecimal)

The conversion starts from the low byte of DT0. It is converted in forward direction (the high word side of [S2] is considered as high-order numerical data).

[i]…US
[S1]…"%4x" : Converts hexadecimal ASCII data (4-digit) to 16-bit data
[S2]…DT0
[N]…H 0002 00 00 : ① ASCII data (forward direction)
[D]…DT100 ② Conversion starting position (0) → [S2]+0 byte
③ Amount of conversion data (2) → two 4-digit data

② Conversion starting position = +0 byte
③ Amount of data to be converted = 2



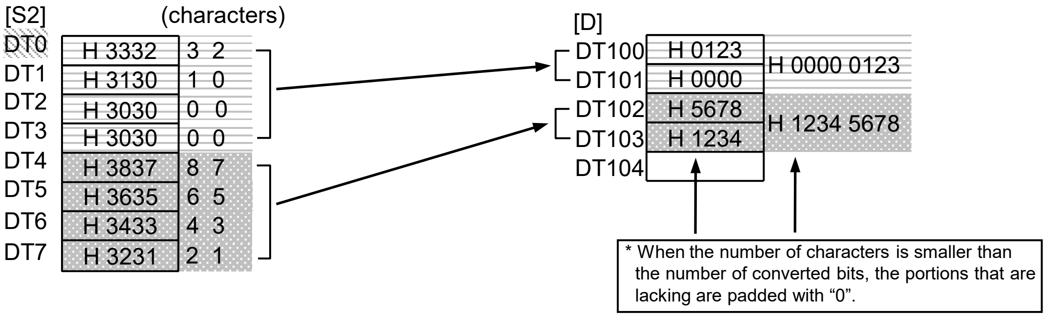
DT0 to DT1: "0123" → DT100: H 0123	
DT2 to DT3: "89AB" → DT101: H 89AB	

Example 14) Converting two decimal ASCII data (8 digits) to two 32-bit BCD data

The conversion starts from the low byte of DT0. It is converted in forward direction (the high word side of [S2] is considered as high-order numerical data). For empty digits of the storage area, zeros (0) are inserted.

[i]…UL
[S1]…"%8b" : Converts decimal ASCII data (8-digit) to 32-bit BCD data
[S2]…DT0
[N]…H 0002 00 00 : ① ASCII data (forward direction)
[D]…DT100 ② Conversion starting position (0) → [S2]+0 byte
③ Amount of conversion data (2) → two 8-digit data

② Conversion starting position = +0 byte
③ Amount of data to be converted = 2

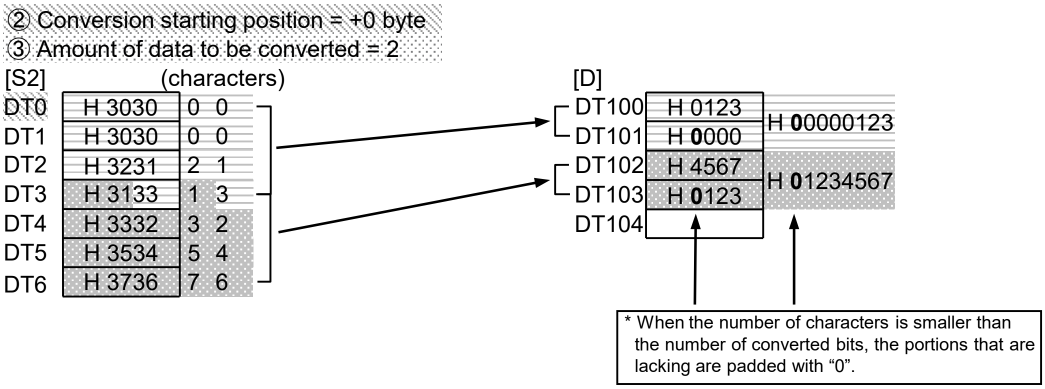


DT0 to DT3 : "00000123" → DT100 to DT101 : H 0000 0123	
DT4 to DT7 : "12345678" → DT102 to DT103 : H 1234 5678	

Example 15) Converting two decimal ASCII data (7 digits) to two 32-bit BCD data

The conversion starts from the low byte of DT0. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data). For empty digits of the storage area, zeros (0) are inserted.

[i]…UL
[S1]…"%7b" : Converts decimal ASCII data (7-digit) to 32-bit BCD data
[S2]…DT0
[N]…H 0002 00 01 : ① ASCII data (reverse direction)
[D]…DT100 ② Conversion starting position (0) → [S2]+0 byte
③ Amount of conversion data (2) → two 7-digit data



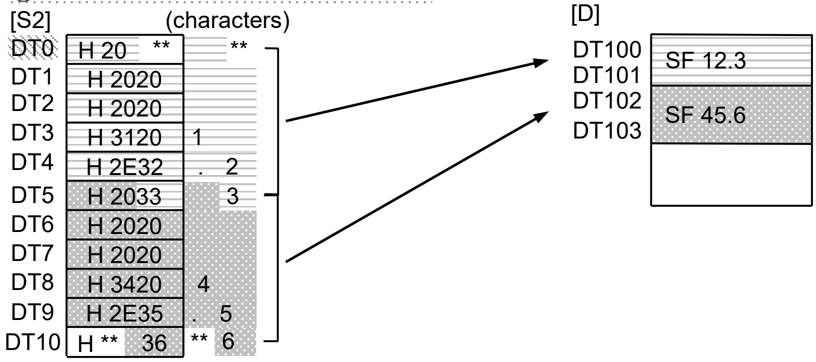
DT0 to DT3: "0000123" →	DT100 to DT101: H **0**000 0123
DT3 to DT6: "1234567" →	DT102 to DT103: H **0**123 4567

Example 16) Converting two floating point ASCII data (10 digits) to two 32-bit single-precision real number data

The conversion starts from the high byte of DT0. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data).

[i]…SF
[S1]… "%10f" : Converts floating point ASCII data (10 characters) to 32-bit real number data
[S2]…DT0
[N]…H 0002 01 01 : ① ASCII data (reverse direction)
[D]…DT100 ② Conversion starting position (1) → [S2]+1 byte
③ Amount of conversion data (2) → two 10-digit data

② Conversion starting position = +1 byte
③ Amount of data to be converted = 2



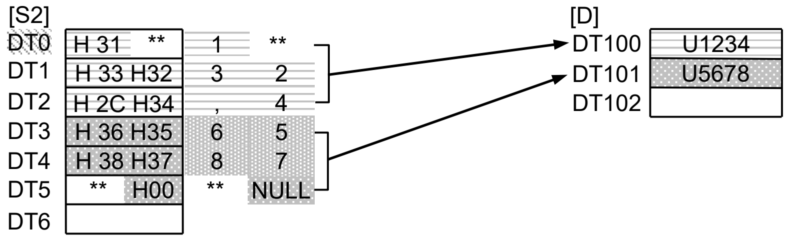
DT0 to DT5: " 12.3" → DT100 to DT101: SF 12.3	
DT5 to DT10: " 45.6" → DT102 to DT103: SF 45.6	

Example 17) Converting decimal ASCII data (separated by commas, ending with NULL) to two 16-bit binary data (decimal)

The conversion starts from the high byte of DT0. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data). It is processed considering commas as the data delimiter, and NULL as end of the data.

[i]...US
[S1]..."%d" : Converts decimal ASCII data to 16-bit data
[S2]...DT0
[N]...H 0002 01 01 : ① ASCII data (reverse direction)
[D]...DT100 ② Conversion starting position (1) → [S2] + 1 byte
③ Amount of conversion data (2) → 2 data separated by a comma and ending with NULL

② Conversion starting position = + 1 byte
③ Amount of data to be converted = 2

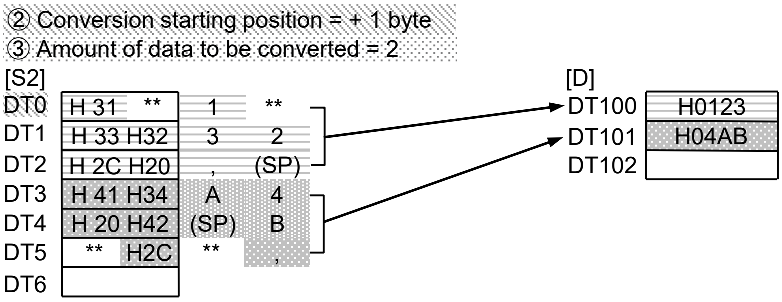


DT0 to DT2: "1234" → DT100: U1234	
DT3 to DT5: "5678" → DT101: U5678	

Example 18) Converting two hexadecimal ASCII data (separated by commas, with spaces) to two 16-bit binary data (hexadecimal)

The conversion starts from the high byte of DT0. It is converted in reverse direction (the low word side of [S2] is considered as high-order numerical data). It is processed considering commas as the data delimiters. If a space is inserted after numerical data, the space is ignored and the numerical data is converted.

[i]...US
[S1]..."%x" : Converts hexadecimal ASCII data to 16-bit data
[S2]...DT0
[N]...H 00020101 : ① ASCII data (reverse direction)
[D]...DT100
② Conversion starting position (1) → [S2] + 1 byte
③ Amount of conversion data (2) → 2 data separated by a comma



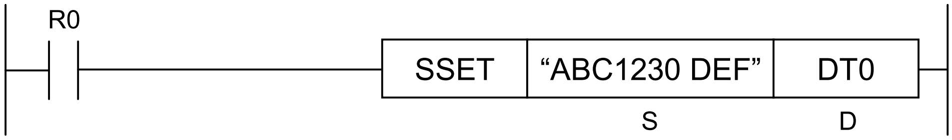
DT0 to DT2: "123" → DT100: H0123	
DT3 to DT5: "AB" → DT101: H04AB	

■ Flag operations

Name	Description
SR7 SR8 (ER)	Set when the range is exceeded during indirect access (index modification).
	To be set when the conversion format specified by [S1] is not an available operation unit.
	To be set when the conversion format specified by [S1] is not a control string.
	To be set when the number of digits for the conversion format specified by [S1] is 28 or more.
	To be set when the number of digits for the conversion format specified by [S1] is omitted, and the string for [S2] is specified with 28 or more digits without delimiters.
	To be set when the ASCII data specified by [S2] is a string that cannot be converted. Example 1) When the operation unit is US and the value exceeds the maximum number for conversion "65535" Example 2) When the strings for conversion contain characters other than values that can be converted (0 to 9, A to F)
	To be set when the beginning of conversion specified by [N] exceeds the [S2] area.
	To be set when the conversion data amount specified by [N] exceeds the [S2] area.
	To be set when the conversion data amount specified by [N] is out of the range.
	To be set when the conversion result exceeds the area specified by [D].

14.12 SSET (Conversion: Character Constant → ASCII Code)

■ Ladder diagram



■ List of operands

Operand	Description
S	Source string
D	Destination starting device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S																				●	
D	●	●	●	●			●	●	●												●

■ Outline of operation

- This instruction converts the character constant specified by [S] to an ASCII code. The result is stored in the area starting with [D].
- Character constants should be put between "" (double quotations) for specification.
- From 0 to 256 characters can be specified for a character constant.
- The number of characters is stored in 1 word at the beginning of the storage area specified by [D]. In the subsequent areas, the character data converted to ASCII is stored in order starting from the low byte.
- When setting, NULL (00) is not added to the end of characters.

■ Processing

Example 1) Setting the 11 characters of the string "ABC1230 DEF" in DT0

“ABC1230 DEF” →

Byte address	Higher bytes	Lower bytes
DT0	11 (Number of characters)	
DT1	H 42 (B)	H 41 (A)
DT2	H 31 (1)	H 43 (C)
DT3	H 33 (3)	H 32 (2)
DT4	H 20 ( )	H 30 (0)
DT5	H 45 (E)	H 44 (D)
DT6		H 46 (F)

Example 2) Setting 256 characters to DT0, repeating a set of the 16 characters from A to P

“ABCDE…LMNOP” →

Byte address	Higher bytes	Lower bytes
DT0	256 (Number of characters)	
DT1	H 42 (B)	H 41 (A)
DT2	H 44 (D)	H 43 (C)
DT3	H 46 (F)	H 45 (E)
⋮	⋮	
DT126	H 4C (L)	H 4B (K)
DT127	H 4E (N)	H 4D (M)
DT128	H 50 (P)	H 4F (O)

■ Precautions for programming

- The character data of the [D] area from before performing the operation is overwritten.

■ Flag operations

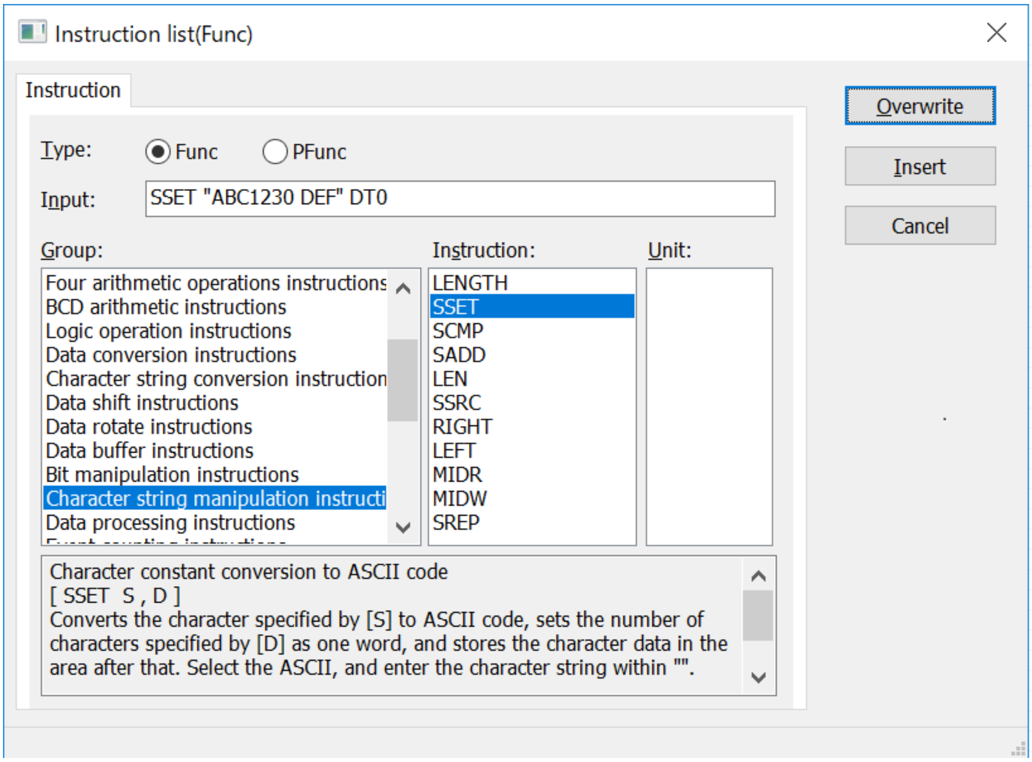
Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the destination range is outside the accessible range.

■ Reference: Using FPWIN GR7 to enter instructions

Use the following procedure to enter instructions.

1. Press <Func (F6)>.
   The "Instruction list (Func)" dialog box is displayed.
2. Enter "SSET".
3. Enter a space.
4. Either select the **ASCII** shown in the pull-down menu, or enter a double quotation mark.
5. Enter a user-defined string to be specified by operand [S].
6. Enter a double quotation mark.
7. Enter a space.
8. Enter the device number for the desired device.

9. Press either the [Overwrite] or [Insert] button.



■ Reference Table: ASCII Codes

								b7								
								b6	0	0	0	0	1	1	1	1
								b5	0	0	1	1	0	0	1	1
								b4	0	1	0	1	0	1	0	1
b7	b6	b5	b4	b3	b2	b1	b0	R \ C	0	1	2	3	4	5	6	7
				0	0	0	0	0	NUL	DEL	SPACE	0	@	P	`	p
				0	0	0	1	1	SOH	DC_1	!	1	A	Q	a	q
				0	0	1	0	2	STX	DC_2	"	2	B	R	b	r
				0	0	1	1	3	ETX	DC_3	#	3	C	S	c	s
				0	1	0	0	4	EOT	DC_4	$	4	D	T	d	t
				0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
				0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
				0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
				1	0	0	0	8	BS	CAN	(	8	H	X	h	x
				1	0	0	1	9	HT	EM	)	9	I	Y	i	y
				1	0	1	0	A	LF	SUB	*	:	J	Z	j	z
				1	0	1	1	B	VT	ESC	+	;	K	[	k	{
				1	1	0	0	C	FF	FS	,	<	L	¥	l	|
				1	1	0	1	D	CR	GS	-	=	M	]	m	}
				1	1	1	0	E	SO	RS	.	>	N	^	n	~
				1	1	1	1	F	SI	US	/	?	O	_	o	DEL

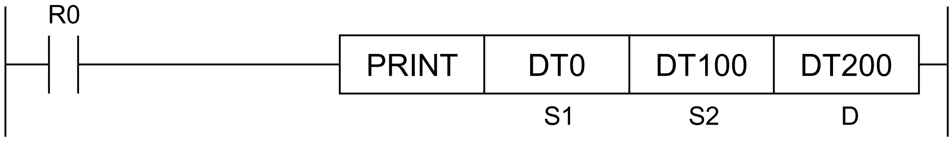
■ Reference Table: JIS8 Codes

								b7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
								b6	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
								b5	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
								b4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b7	b6	b5	b4	b3	b2	b1	b0	R \ C	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
				0	0	0	0	0	NUL	TC_7 (DEL)	(SP)	0	@	P	`	p	Undefined	Undefined	Undefined	ー	タ	ミ	Undefined	Undefined
				0	0	0	1	1	TC_1 (SOH)	DC_1	!	1	A	Q	a	q			。	ア	チ	ム		
				0	0	1	0	2	TC_2 (STX)	DC_2	"	2	B	R	b	r			「	イ	ツ	メ		
				0	0	1	1	3	TC_3 (ETX)	DC_3	#	3	C	S	c	s			」	ウ	テ	モ		
				0	1	0	0	4	TC_4 (EOT)	DC_4	$	4	D	T	d	t			、	エ	ト	ヤ		
				0	1	0	1	5	TC_5 (ENQ)	TC_8 (NAK)	%	5	E	U	e	u			・	オ	ナ	ユ		
				0	1	1	0	6	TC_6 (ACK)	TC_9 (SYN)	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
				0	1	1	1	7	BEL	ETB	'	7	G	W	g	w			ァ	キ	ヌ	ラ		
				1	0	0	0	8	EE_0 (BS)	CAN	(	8	H	X	h	x			ィ	ク	ネ	リ		
				1	0	0	1	9	EE_1 (HT)	EM	)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
				1	0	1	0	A	EE_2 (LF)	SUB	*	:	J	Z	j	z			ェ	コ	ハ	レ		
				1	0	1	1	B	EE_3 (VT)	ESC	+	;	K	[	k	|			ォ	サ	ヒ	ロ		
				1	1	0	0	C	EE_4 (FF)	IS_4 (FS)	,	<	L	¥	l	|			ャ	シ	フ	ワ		
				1	1	0	1	D	EE_5 (CR)	IS_3 (GS)	-	=	M	]	m	|			ュ	ス	ヘ	ン		
				1	1	1	0	E	SO	IS_2 (RS)	.	>	N	^	n	¯			ョ	セ	ホ	”		
				1	1	1	1	F	SI	IS_1 (US)	/	?	O	_	o	DEL			ッ	ソ	マ	’		

Do not use the undefined sections in the JIS8 code table.

14.13 PRINT (Text Creation)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting address of the device storing the string data which indicates the create text form or a character constant
S2	Starting address of the device storing the data to be output in text format
D	Starting address of the device storing the text

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●			●	●													●
D	●	●	●	●			●	●													●

■ Outline of operation

- This instruction is used for creating texts of mails, etc.

■ Processing

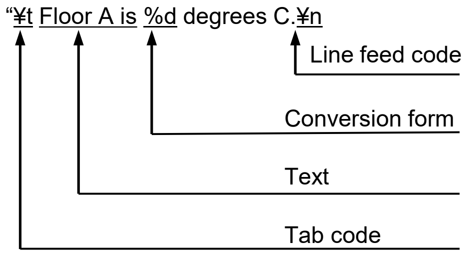
- ASCII code texts are created according to the specified text creation form.
- The text creation form can be specified using the operand [S1], SSET instruction or the mail setting screen of FPWIN GR7. Created texts can be connected using the SADD instruction.
- The maximum size of a mail text is 4096 bytes for sending an event mail, and 256 bytes for sending a logging/trace mail.

■ Operand [S1] setting

- Specify the device address storing the text creation form or character constant (max. 256 characters).
- The text creation form is composed of a main text, conversion form (such as %d, %e), linefeed code (\n) and horizontal tab code (\t).

Example of the create text form

This example includes a tab code, a body (a conversion form for 1 datum is inserted), and a linefeed code.



- Tab code (\t) is converted to ASCII code HT (09h).
- The body is converted to the supported ASCII code or Shift JIS code.
- In the part where a conversion form is inserted, the output data specified by [S2] is stored as ASCII code, according to the conversion form. For more information about the conversion form, refer to"14.14 PRINT/EPRINT Instruction Shared Conversion Form Table"
- Linefeed code (\n) is converted to ASCII code CR+LF (0A0Dh).

Restrictions

- Up to 4096 characters can be specified for the text creation form. An operation error occurs when it exceeds 4096 characters.
- Up to 16 digits can be specified for one conversion form. An operation error occurs when it exceeds 16 digits.
- The maximum number of characters after conversion for a single datum excluding %s and %S is 32. An operation error occurs when it exceeds 32 characters.
- The maximum number of characters for %s or %S after conversion is 4096.
- All strings that are not recognized as conversion forms are treated as main texts.

  Example: Conversion forms that do not allow upper case characters (%D, etc.)

  Character strings that contain characters not recognized as discriminant characters of conversion forms (%A, %Z, etc.)
- To enter "%" in the body, specify "%%" (% x 2).

■ Operand [S2] setting

- Specify the starting address of the device storing the data to be output in the create text form.
- Arrange conversion data in the order specified in the conversion form.
- As for character data for %s, the data storing the number of (1-byte) characters is specified at the beginning. It can be set using the SSET instruction.

Example:

SSET "Floor" DT112
S1 = "%d %u %x %b %f %e %Lg %s"
S2 = DT100
Result: -1 65535 ffff 1000 123.4567 123.4567 123.456789 Floor

DT100	H FFFF		Data for %d	
DT101	H FFFF		Data for %u	
DT102	H FFFF		Data for %x	
DT103	H 1000		Data for %b	
DT104 DT105	SF 123.4567		Data for %f	
DT106 DT107	SF 123.4567		Data for %e	
DT108 DT109 DT110 DT111	DF 123.456789		Data for %Lg	
DT112	K 5		Data for %s	No. of byte
DT113	H 6c (l)	H 46 (F)		Data part
DT114	H 6f (o)	H 6f (o)		Data part
DT115	**	H 72 (r)		Data part

■ Setting example

Example 1) When inserting into the text two conversion forms (%d) that represent 16-bit signed integers and a linefeed code (\n)

In the place of the conversion form (%d), the ASCII code that is equivalent to the integer data specified by [S2] is inserted. If a number of digits is not specified for the conversion form, the number of converted data and the size of the storage area will vary according to the value [S2].

- Image of mail text

Floor A: 25 degrees C. Floor B: 28 degrees C.

- Setting values

S1="Floor A: %d degrees C.¥nFloor B: %d degrees C."
S2=DT100
D=DT200

DT100	K 25	Data for %d
DT101	K 28	Data for %d
DT102		

DT200	H002D		The number of bytes is stored.
DT201	H 6C (l)	H 46 (F)	
DT202	H 6F (o)	H 6F (o)	
DT203	H 20 (␣)	H 72 (r)	
DT204	H 3A (:)	H 41 (A)	
DT205	H 32 (2)	H 20 (␣)	The converted data for %d is inserted.
DT206	H 20 (␣)	H 35 (5)	
DT207	H 65 (e)	H 64 (d)	
DT208	H 72 (r)	H 67 (g)	
DT209	H 65 (e)	H 65 (e)	
DT210	H 20 (␣)	H 73 (s)	
DT211	H 2E (.)	H 43 (C)	The ¥ line feed code (CR) is inserted.
DT212	H 46 (F)	H 0D (CR)	
DT213	H 6F (o)	H 6C (l)	
DT214	H 72 (r)	H 6F (o)	
DT215	H 42 (B)	H 20 (␣)	
DT216	H 20 (␣)	H 3A (:)	The converted data for %d is inserted.
DT217	H 38 (8)	H 32 (2)	
DT218	H 64 (d)	H 20 (␣)	
DT219	H 67 (g)	H 65 (e)	
DT220	H 65 (e)	H 72 (r)	
DT221	H 73 (s)	H 65 (e)	
DT222	H 43 (C)	H 20 (␣)	
DT223	H 00	H 2E (.)	

Example 2) When inserting into the text a conversion form (%d) that represents a 16-bit signed integer

In the place of the conversion form (%d), the ASCII code that is equivalent to the integer data specified by [S2] is inserted. If a number of digits is not specified for the conversion form, the number of converted data and the size of the storage area will vary according to the value [S2].

Example 2)

- Image of mail text

Production volume: 5

- Setting values

S1="Production volume: %d"
S2=DT1
D=DT50

DT1	K 5
DT2	
DT3	

DT50	U 14		No. of bytes
DT51	H 72 (r)	H 50 (P)	Data part
DT52	H64 (d)	H 6f (o)	
DT53	H 63 (c)	H 75 (u)	
DT54	H 69 (i)	H 74 (t)	
DT55	H 6e (n)	H 6f (o)	
DT56	H 76 (v)	H 20 (SPACE)	
DT57	H 6c (l)	H 6f (o)	
DT58	H 6d (m)	H 75 (u)	
DT59	H 3a (:)	H 65 (e)	
DT60	H 35 (5)	H 20 (SPACE)	

Example 3) When inserting into the text a horizontal tab code (\t: H09)

In the place of the conversion form (\t), the ASCII code that is equivalent to the horizontal tab code is inserted. If a conversion form is not included in [S1], the data for [S2] will have no effect on the conversion results.

- Image of mail text

(Tab)Normal operation

- Setting values

S1="\tNormal operation"
S2=DT1
D=DT50

DT1	
DT2	
DT3	

DT50	U 11		No. of bytes
DT51	H 4e (N)	H 09 (HT)	Data part
DT52	H 72 (r)	H 6f (o)	
DT53	H 61 (a)	H 6d (m)	
DT54	H 20 (SPACE)	H 6c (l)	
DT55	H 70 (p)	H 6f (o)	
DT56	H 72 (r)	H 65 (e)	
DT57	H 74 (t)	H 61 (a)	
DT58	H 6f (o)	H 69 (i)	
DT59	**	H 6e (n)	

Example 4) When inserting into the text two conversion forms (%s) that represent strings

In the place of the conversion form (%s), the ASCII code that is equivalent to the string data specified by [S2] is inserted. If a number of digits is not specified for the conversion form, the number of converted data and the size of the storage area will vary according to the value [S2].

Example 4)

- Image of mail text

Location: Nagoya, Aichi

- Setting values

S1="Location: %s, %s"
S2=DT1
D=DT50

DT1	U 6	
DT2	H 61 (a)	H 4E (N)
DT3	H 6F (o)	H 67 (g)
DT4	H 61 (a)	H 79 (y)
DT5	U 5	
DT6	H 69 (i)	H 41 (A)
DT7	H 68 (h)	H 63 (c)
DT8	**	H 69 (i)

DT50	U 17		No. of bytes
DT51	H 6F (o)	H 4C (L)	Data part
DT52	H 61 (a)	H 63 (c)	
DT53	H 69 (i)	H 74 (t)	
DT54	H 6E (n)	H 6F (o)	
DT55	H 20 (SPACE)	H 3A (:)	
DT56	H 61 (a)	H 4E (N)	
DT57	H 6F (o)	H 67 (g)	
DT58	H 61 (a)	H 79 (y)	
DT59	H 20 (SPACE)	H 2C (,)	
DT60	H 69 (i)	H 41 (A)	
DT61	H 68 (h)	H 63 (c)	
DT62	**	H 69 (i)	

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when an out-of-range value is specified for parameters.
	To be set when the text creation form exceeds 4096 characters.
	To be set when texts exceed 4096 bytes.
	To be set when the conversion form is specified by a real number and conversion data is a non-real number.
	To be set when the size specified by the conversion form exceeds 32 characters. (excluding the conversion form %s)

14.14 PRINT/EPRINT Instruction Shared Conversion Form Table

This table indicates the format for the "Conversion Form" that can be inserted in the "Text Creation Form" to be specified for operand [S1] from the PRINT instruction or the EPRINT instruction.

■ Setting the control string [S1]

Specifies the conversion data type, number of characters, precision, etc. using the string data of the following formats. A variety of options (such as inserting a sign or spaces) can also be selected depending on the type of data to be converted. For details, see below.

[S1] = " % + 12.5 L d , "

Option setting (1)

0 : Zero padding
+ : A sign is added (plus sign)
␣: A space is inserted
- : Left align (default is right align)
: Characters are added according to the conversion data

No. of characters after conversion and the precision

Specify the total number of characters (n) and the number of characters of precision (m) with [n.m], [n], or [.m]. The number of characters of precision (m) changes according to the type of conversion data

Type of data to be converted	No. of characters of precision (m)
d , Ld, i , Li, u ,Lu, x , Lx , b, Lb	represents the number of characters in numerical strings.
f, Lf, e, Le, E, LE	represents the number of characters after the decimal point.
g, Lg, G, LG	represents the number of significant figures.

Option setting (2)

, : A comma is added
BCD : Postfix characters are added
H : Postfix characters are added

Type of data to be converted (1)

d: Signed integer → Decimal ASCII
u: Unsigned integer → Decimal ASCII
x: Unsigned integer → Hexadecimal ASCII
b: BDC integer → Hexadecimal ASCII
f: Floating point real number → Floating point ASCII
e: Floating point real number → Exponential notation ASCII
g: Floating point real number → Floating point ASCII or exponential notation ASCII
s: String data → ASCII

Type of data to be converted (2)

L: Specify for 32-bit integer data or 64-bit real number data

■ Conversion form table

Conversion form	Data format		Usage example
	Data before conversion	ASCII data after conversion	
"%d" or "%i"	16-bit data (signed integer)	Decimal ASCII data	"%d" , "%5d" , "%+5d" , "%-5d" , "%05d" , "%10.5d" , "% d"
"%Ld" or "%Li"	32-bit data (signed integer)	Decimal ASCII data	"%Ld" , "%5Ld" , "%+5Ld" , "%-5Ld" , "%05Ld" , "%10.5Ld" , "% Ld"
"%u"	16-bit data (unsigned integer)	Decimal ASCII data	"%u" , "%5u" , "%-5u" , "%05u" , "%10.5u"
"%Lu"	32-bit data (unsigned integer)	Decimal ASCII data	"%Lu" , "%5Lu" , "%-5Lu" , "%05Lu" , "%10.5Lu"
"%x"	16-bit data	Hexadecimal ASCII data	"%x" , "%5x" , "%-5x" , "%05x" , "%10.5x" , "%#x" , "%X"
"%Lx"	32-bit data	Hexadecimal ASCII data	"%Lx" , "%5Lx" , "%-5Lx" , "%05Lx" , "%10.5Lx" , "%#Lx" , "%LX"

Conversion form	Data format		Usage example
	Data before conversion	ASCII data after conversion	
"%b"	16-bit BCD data	Decimal ASCII data	"%b" , "%5b" , "%-5b" , "%05b" , "%10.5b"
"%Lb"	32-bit BCD data	Decimal ASCII data	"%Lb" , "%5Lb" , "%-5Lb" , "%05Lb" , "%10.5Lb"
"%f"	32-bit single-precision real number data	Floating point number ASCII data	"%f" , "%5.2f" , "%+5.2f" , "%-5.2f" , "%05.2f" , "%#f" , "% f"
"%Lf"	64-bit double-precision real number data	Floating point number ASCII data	"%Lf" , "%5.2Lf" , "%+5.2Lf" , "%-5.2Lf" , "%05.2Lf" , "%#Lf" , "% Lf"
"%e"	32-bit single-precision real number data	Exponential notation ASCII data	"%e" , "%5.2e" , "%+5.2e" , "%-5.2e" , "%05.2e" , "%#5.2e" , "% e" , "%E"
"%Le"	64-bit double-precision real number data	Exponential notation ASCII data	"%Le" , "%5.2Le" , "%+5.2Le" , "%-5.2Le" , "%05.2Le" , "%#5.2Le" , "% Le" , "%LE"
"%g"	32-bit single-precision real number data	Exponential notation ASCII data or floating-point ASCII data (whichever is shorter in the relevant notation)	"%g" , "%5.2g" , "%+5.2g" , "%-5.2g" , "%05.2g" , "%#5.2g" , "%G"
"%Lg"	64-bit double-precision real number data	Exponential notation ASCII data or floating-point ASCII data (whichever is shorter in the relevant notation)	"%Lg" , "%5.2Lg" , "%+5.2Lg" , "%-5.2Lg" , "%05.2Lg" , "%#5.2Lg" , "%LG"
"%s"	String data	String data (for the specified number of characters)	"%s" , "%5s" , %-5s" ,"%-05s"
"%S"	String data	String data (conversion for the specified number of characters, or up to H0)	"%S" , "%5S" , %-5S" ,"%-05S"

(Note 1) The number of converted digits for the conversion form is up to 16 digits.

(Note 2) Conversion modifier 'L' can also specified in a lower case character.

(Note 3) "%S" (upper-case letter) is supported for CPU unit Ver. 4.10 and later, or Ver. 3.40 to Ver. 3.**.

■ Options for the conversion form [S1] (BIN data → ASCII data)

Items	Conversion form	BIN data before conversion	ASCII data after conversion	Description
Specification of upper / lower case characters	%x	H ABCD	"abcd"	Specifies upper or lower case alphabets used for hexadecimal / exponential notation ASCII data. For %d, %u, %b, and %f, upper-case letters are handled as body data.
	%X	H ABCD	"ABCD"	
	%e	SF1234.567	"1.234567e+3"	
	%E	SF1234.567	"1.234567E+3"	
Specification of the number of display digits	%d	K 100	"100"	The display digit is specified with "Total number of characters" and "Number of characters of precision." It is specified with “n.m”, “n”, or “.m”, etc.
	%5d	K 100	"␣␣100"	
	%10.5d	K 100	"␣␣␣␣␣00100"	

<table>
<tr><th>Items</th><th>Conversion form</th><th>BIN data before conversion</th><th>ASCII data after conversion</th><th colspan="2">Description</th></tr>
<tr><td rowspan="11"></td><td>%x</td><td>H 12A</td><td>"12a"</td><td colspan="2" rowspan="11">n: Total number of characters, m: Number of characters of precision
<Number of characters of precision>
[d , ld , i , Li , u , Lu , x , Lx , X , LX , b , Lb] represents the number of characters of numerical strings.
[f , Lf , e , Le , E , LE] represents the number of characters after the decimal point. Be sure to also specify the number of places after the decimal point.
[g , Lg , G , LG] represent the number of significant figures.
If there is no specification for the number of characters, the number of digits for the data after conversion and the storage area size will vary according to the data before conversion.</td></tr>
<tr><td>%5x</td><td>H 12A</td><td>"␣␣12a"</td></tr>
<tr><td>%10.5x</td><td>H 12A</td><td>"␣␣␣␣␣0012a"</td></tr>
<tr><td>%b</td><td>H 123</td><td>"123"</td></tr>
<tr><td>%5b</td><td>H 123</td><td>"␣␣123"</td></tr>
<tr><td>%f</td><td>SF 123.4567</td><td>"123.4567"</td></tr>
<tr><td>%8.3f</td><td>SF 123.4567</td><td>"␣123.457"</td></tr>
<tr><td>%e</td><td>SF 1234.567</td><td>"1.234567e+03"</td></tr>
<tr><td>%10.3e</td><td>SF 1234.567</td><td>"␣1.235e+03"</td></tr>
<tr><td>%g</td><td>SF 1234.567</td><td>"1234.567"</td></tr>
<tr><td>%8.6g</td><td>SF 1234.567</td><td>"␣1234.57"</td></tr>
<tr><td rowspan="5">Specification of zero padding</td><td>%05d</td><td>K 100</td><td>"00100"</td><td colspan="2" rowspan="5">When the setting for the display digit is available, zero padding can be specified. Put zero (0) before the display digit.</td></tr>
<tr><td>%05x</td><td>H 12A</td><td>"0012a"</td></tr>
<tr><td>%05b</td><td>H 123</td><td>"00123"</td></tr>
<tr><td>%08.3f</td><td>SF 123.4567</td><td>"0123.457"</td></tr>
<tr><td>%010.3e</td><td>SF 1234.567</td><td>"01.235e+03"</td></tr>
<tr><td rowspan="5">Specification of right align and left align</td><td>%-5d</td><td>K 100</td><td>"100␣␣"</td><td colspan="2" rowspan="5">Default is right align. To set to left align, add minus (-) before the specification of digit number.</td></tr>
<tr><td>%-5x</td><td>H 12A</td><td>"12a␣␣"</td></tr>
<tr><td>%-5b</td><td>H 123</td><td>"123␣␣"</td></tr>
<tr><td>%-8.3f</td><td>SF 123.4567</td><td>"123.457␣"</td></tr>
<tr><td>%-010.3e</td><td>SF 1234.567</td><td>"1.235e+03␣"</td></tr>
<tr><td rowspan="5">Specification of sign</td><td>%+d</td><td>K 100</td><td>"+100"</td><td colspan="2" rowspan="5">This option is specified to add a plus sign (+). A plus sign (+) is not added by default.</td></tr>
<tr><td>%+d</td><td>K -100</td><td>"-100"</td></tr>
<tr><td>%+5d</td><td>K 100</td><td>"␣+100"</td></tr>
<tr><td>%+8.3f</td><td>123.4567</td><td>"+123.457"</td></tr>
<tr><td>%+10.3e</td><td>1234.567</td><td>"+1.234e+03"</td></tr>
<tr><td rowspan="6">Specification of numerical position</td><td>%␣d</td><td>K 100</td><td>"␣100"</td><td colspan="2" rowspan="6">In the case of a positive number, a space is added to align the positive number with negative numbers.
When specifying %u, %x, or %b, existence of "␣" does not affect the results.</td></tr>
<tr><td>%␣d</td><td>K -100</td><td>"-100"</td></tr>
<tr><td>%␣8.3f</td><td>SF 123.4567</td><td>"␣123.457"</td></tr>
<tr><td>%␣8.3f</td><td>SF -123.4567</td><td>"-123.457"</td></tr>
<tr><td>%␣10.3e</td><td>SF 1234.567</td><td>"␣1.235e+03"</td></tr>
<tr><td>%␣10.3e</td><td>SF -1234.567</td><td>"-1.235e+03"</td></tr>
<tr><td rowspan="3">Specification of another output format for numerical data type</td><td>%#x</td><td>H 12A</td><td>"0x12a"</td><td>"0x" is added.</td><td rowspan="3">Another output type is automatically given by adding "#".</td></tr>
<tr><td>%#X</td><td>H 12A</td><td>"0X12A"</td><td>"0X" is added.</td></tr>
<tr><td>%#8.0f</td><td>SF 123.4567</td><td>"␣␣␣␣123."</td><td>"." is always added.</td></tr>
</table>

<table>
<tr><th>Items</th><th>Conversion form</th><th>BIN data before conversion</th><th>ASCII data after conversion</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4"></td><td>%#10.0e</td><td>SF 1234.567</td><td>"␣␣␣␣1.e+03"</td><td rowspan="2"></td><td rowspan="4">When specifying %u, %x, or %b, existence of "#" does not affect the results.</td></tr>
<tr><td>%#10.3E</td><td>SF 1234.567</td><td>"␣␣␣␣1.E+03"</td></tr>
<tr><td>%#9.0g</td><td>SF 1234</td><td>"␣␣␣1234.0"</td><td rowspan="2">"." is always added, and "0" after the decimal point is not omitted.</td></tr>
<tr><td>%#.9G</td><td>SF 1234</td><td>"1234.0000"</td></tr>
</table>

(Note 1) "␣" in the table represents a space.

(Note 2) For exponential notation, it consists of a code (e or E), a sign, and a 2-digit number.

(Note 3) If the conversion results in having fewer enabled digits than before conversion, the result is rounded off.

(Note 4) If a plus sign (+) and a space (␣) are used together to specify the sign and the digit position respectively and the space (␣) comes first, neither the "sign indication" nor the "specification of digit position" will be valid. When (+) comes first, "sign indication" will be valid.

Example 1) %␣+d K100 → The output data is "100", and neither a space nor the sign is added.

Example 2) %+␣d K100 → Output data is "+100" with a (+) sign.

■ Processing when conversion forms are combined (BIN data to ASCII data)

<table>
<tr><th>Conversion form</th><th>Binary data before conversion</th><th>ASCII data after conversion</th><th>Remarks</th></tr>
<tr><td>%-10.3e</td><td>SF123.4567</td><td>"1.235e+02␣"</td><td>Exponent is output in at least 2 digits.</td></tr>
<tr><td>%+ u</td><td>U1234</td><td>"1234"</td><td>For %u, %x, or %b, the existence of a plus sign (+) in the conversion form does not affect the result.</td></tr>
<tr><td>% ␣ u</td><td>U1234</td><td>"1234"</td><td>For %u, %x, or %b, the existence of a space (␣) in the conversion form does not affect the result.</td></tr>
<tr><td>%#u</td><td>U 1234</td><td>"1234"</td><td>For %u, %x, or %b, the existence of a number sign (#) in the conversion form does not affect the result.</td></tr>
<tr><td>%␣+d</td><td>K1234</td><td>"1234"</td><td rowspan="2">If a plus sign (+) and a space (␣) are used together to specify the sign and the digit position respectively and the space (␣) comes first, neither the "sign indication" nor the "specification of digit position" will be valid. When plus sign (+) comes first, "sign indication" will be valid.
Example 1) %␣+d K100 → The output data is "100", and neither a space nor the sign is added.
Example 2) %+␣d K100 → Output data is "+100" with a (+) sign.</td></tr>
<tr><td>%+␣d</td><td>K1234</td><td>"+1234"</td></tr>
</table>

■ Options for the conversion form [S1] (String data → ASCII data)

<table>
<tr><th>Items</th><th>Conversion form</th><th>String data before conversion</th><th>ASCII data after conversion</th><th>Description</th></tr>
<tr><td rowspan="2">Specification of the number</td><td>%s</td><td>"abcdef"</td><td>"abcdef"</td><td rowspan="2">In the case of "%s", it is left-aligned by default.</td></tr>
<tr><td>%10s</td><td>"abcdef"</td><td>"abcdef␣␣␣␣"</td></tr>
</table>

Items	Conversion form	String data before conversion	ASCII data after conversion	Description
of display digits	%10.5s	"abcdef"	"abcdef␣␣␣␣"	Specify the number of digits per byte (equivalent to 1-byte character). For 2-byte characters, the number of digits is 2. When the digit number is not enough, an operation error occurs. When the decimal part is specified with %s, the settings after (.) will be invalid.
Specification of zero padding	%-010s	"abcdef"	"00000abcdef"	When the setting for the display digit is available, zero padding can be specified. Put zero (0) before the display digit.
	%010s	"abcdef"	"abcdef␣␣␣␣"	
Specification of right align and left align	%-10s	"abcdef"	"␣␣␣␣abcdef "	Default is left align. To set to right align, add a minus sign (-) before the specification of the number of digits.

(Note 1) "␣" in the table represents a space.

14.15 TIMEstr (Date and Time Character String Conversion)

■ Ladder diagram

■ List of operands

Operand	Description
S1	Starting address of the device that stores date and time information (7 words)
S2	Starting address of the device that stores conversion patterns (hex data, 1 word)
D	Starting address of the device that stores the string data as the conversion result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●		●	●	●													●
S2	●	●	●	●			●	●									●				●
D	●	●	●	●			●	●													●

■ Outline of operation

- This instruction converts data and time information to character strings.
- The date and time information to be output is year, month, day, day of the week, hour, minute, and second.
- This instruction can be used when date and time information is required for creating mail texts. This instruction is used in combination with the PRINT instruction.

■ Processing

- Converts the date and time information specified by [S1] to ASCII code, and stores it as string data in the area that starts with [D].
- The conversion pattern is specified for [S2].
- The string data of the conversion result for [D] is output as the ASCII code (single byte) or Shift JIS code (double byte).

■ [S1]: Settings of date and time information

- Specify the starting address of the device that stores the date and time information.
- Specify SD50 if you want to output the current time and date of PLC.

- Checking the combination of year, month, day, or day of the week is not performed.
  Example) A setting of February 31 is not treated as an error. When SD50 is specified, the combination of year, month, day, or day of the week is correct.
- Always store the data in the order mentioned in the table below regardless of the conversion pattern of [S2].

Operand	Description	Specified range	Remarks
[S1]	Year	U0 to U99	The character string after conversion is 2000 to 2099.
[S1+1]	Month	U1 to U12	
[S1+2]	Day	U1 to U31	
[S1+3]	Hours	U0 to U23	
[S1+4]	Minutes	U0 to U59	
[S1+5]	Seconds	U0 to U59	
[S1+6]	Day of the week	U0 to U6	

■ [S2]: Specification of conversion pattern

The conversion pattern is specified by 4-digit hex data.

H	(First digit)	(Second digit)	(Third digit)	(Fourth digit)

<table>
<tr><th>Items</th><th>Description</th></tr>
<tr><td>(First digit)
Output pattern</td><td>Specify the output pattern for the string after conversion. Refer to the table below for ways to display the month and day of the week.
<table>
<tr><th>Value</th><th>Constitution (Order)</th><th>Format</th><th>Month display</th><th>Day of the week display</th></tr>
<tr><td>0</td><td>Year/Month/Day/(Day of the week)</td><td>Single-byte</td><td>Number</td><td>Japanese</td></tr>
<tr><td>1</td><td>Year/Month/Day/(Day of the week)</td><td>Double-byte</td><td>Number</td><td>Japanese</td></tr>
<tr><td>2</td><td>Year/Month/Day/(Day of the week)</td><td>Single-byte</td><td>Number</td><td>Chinese</td></tr>
<tr><td>3</td><td>Year/Month/Day/(Day of the week)</td><td>Double-byte</td><td>Number</td><td>Chinese</td></tr>
<tr><td>4</td><td>Day of the week/Day/Month/Year</td><td>Single-byte</td><td>Number</td><td>English</td></tr>
<tr><td>5 (Note 1)</td><td>Day of the week/Day/Month/Year</td><td>Single-byte</td><td>Alphabet</td><td>English</td></tr>
<tr><td>6</td><td>Day of the week/Day/Month/Year</td><td>Single-byte</td><td>Number</td><td>English</td></tr>
<tr><td>7 (Note 2)</td><td>Day of the week/Day/Month/Year</td><td>Single-byte</td><td>Alphabet</td><td>English</td></tr>
</table></td></tr>
<tr><td>(Second digit)
Date and time</td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Date and time</td></tr>
<tr><td>1</td><td>Date only</td></tr>
<tr><td>2</td><td>Time only</td></tr>
</table></td></tr>
<tr><td>(Third digit)
Addition of day of the week</td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>No</td></tr>
<tr><td>1 (Note 3)(Note 4)</td><td>Yes</td></tr>
</table></td></tr>
<tr><td>(Fourth digit)
Delimiter</td><td>Specify a delimiter for data.
<table>
<tr><th>Value</th><th>Date</th><th>Between date and time</th><th>Time</th></tr>
<tr><td>0</td><td>/ (slash)</td><td>␣ (space)</td><td>: (colon)</td></tr>
<tr><td>1</td><td>␣ (space)</td><td>␣ (space)</td><td>␣ (space)</td></tr>
<tr><td>2</td><td>- (hyphen)</td><td>␣ (space)</td><td>: (colon)</td></tr>
<tr><td>3</td><td>. (period)</td><td>␣ (space)</td><td>. (period)</td></tr>
<tr><td>4 (Note 5)</td><td>Chinese character</td><td>␣ (space)</td><td>Chinese character</td></tr>
<tr><td>5</td><td>No</td><td>␣ (space)</td><td>No</td></tr>
<tr><td>6</td><td>No</td><td>_ (underscore)</td><td>No</td></tr>
</table></td></tr>
</table>

(Note 1) When specifying 5 or 6 for the first digit, which is the delimiter, output pattern 5 is automatically corrected to pattern 4 for processing.

(Note 2) When specifying 5 or 6 for the first digit, which is the delimiter, output pattern 7 is automatically corrected to pattern 6 for processing.

(Note 3) When specifying 5 or 6 for the first digit, which is the delimiter, the specification of the addition of the day of the week is processed as "0" (No).

(Note 4) When specifying 2 for the third digit, which is for the specification of date and time, the specification of addition of the day of the week is processed as "0" (No).

(Note 5) When specifying a value other than 0 to 3 for the fourth digit of the output pattern, it is processed as the value 5.

■ Example of specification

Conversion pattern	Output content	Output image
H0000	yyyy/mm/dd␣hh:mm:ss	2014/09/05␣05:06:32
H0001	yyyy␣mm␣dd␣hh␣mm␣ss	2014␣09␣05␣05␣06␣32
H0004	yyyy/mm/dd␣hh:mm:ss	2014/09/05 05:06:32
H0005	yyyymmdd␣hhmmss	20140905␣050632
H0006	yyyymmdd_hhmmss	20140905_050632
H0014	yyyy/mm/dd(ddd)␣hh:mm:ss	2014/09/05(Fri) 05:06:32
H0102	yyyy-mm-dd	2014-09-05
H0203	hh.mm.ss	05.06.32
H1000	yyyy/mm/dd␣hh:mm:ss	2014/09/05 05:06:32
H1010	yyyy/mm/dd(d)␣hh:mm:ss	2014/09/05(Fri) 05:06:32
H2014	yyyy/mm/dd(d)␣hh:mm:ss	2014/09/05(5) 05:06:32
H3010	yyyy/mm/dd(d)␣hh:mm:ss	2014/09/05(5) 05:06:32
H4000	dd/mm/yyyy␣hh:mm:ss	05/09/2014␣05:06:32
H4006	ddmmyyyy_hhmmss	05092014_050632
H4012	d␣dd-mm-yyyy␣hh:mm:ss	Fri␣05-09-2014␣05:06:32
H4112	d␣dd-mm-yyyy	Fri␣05-09-2014
H5000	dd/mm/yyyy␣hh:mm:ss	05/Sep/2014␣05:06:32
H5012	d␣dd-mm-yyyy␣hh:mm:ss	Fri␣05-Sep-2014␣05:06:32
H6000	mm/dd/yyyy␣hh:mm:ss	09/05/2014␣05:06:32
H6006	mmddyyyy_hhmmss	09052014_050632
H6012	d␣mm-dd-yyyy	Fri␣09-05-2014
H7000	mm/dd/yyyy␣hh:mm:ss	Sep/05/2014␣05:06:32
H7012	d␣mm-dd-yyyy␣hh:mm:ss	Fri␣Sep-05-2014␣05:06:32
H7112	d␣mm-dd-yyyy	Fri␣Sep-05-2014

■ Example of special specification (when automatically corrected)

Conversion pattern	Output content	Output image
H5005(Note 1)	ddmmyyyy␣hhmmss	05092014␣050632
H7006(Note 2)	mmddyyyy_hhmmss	09052014_050632

Conversion pattern	Output content	Output image
H7016(Note 3)	mmddyyyy_hhmmss	09052014_050632
H5216(Note 4)	hhmmss	050632
H4014(Note 5)	ddmmyyyy_hhmmss	05092014_050632

(Note 1) When specifying 5 or 6 for the first digit, which is the delimiter, output pattern 5 is automatically corrected to pattern 4 for processing.

(Note 2) When specifying 5 or 6 for the first digit, which is the delimiter, output pattern 7 is automatically corrected to pattern 6 for processing.

(Note 3) When specifying 5 or 6 for the first digit, which is the delimiter, the specification of the addition of the day of the week is processed as "0" (No).

(Note 4) When specifying 2 for the third digit, which is for the specification of date and time, the specification of addition of the day of the week is processed as "0" (No).

(Note 5) When specifying a value other than 0 to 3 for the fourth digit of the output pattern, it is processed as the value 5.

■ Example of processing

Example 1)

[S1]…SD50 [S2]…DT0 [D]…DT10

・Output example: Thu␣09-25-2014␣12:54:31

SD50	U 14	Year
SD51	U 9	Month
SD52	U 25	Day
SD53	U 12	Hour
SD54	U 54	Minute
SD55	U 31	Second
SD56	U 4	Day of the week

DT0	H 0014

DT10	U23		No. of bytes
DT11	H 68 (h)	H 54 (T)	
DT12	H 20 (␣)	H 75 (u)	
DT13	H 39 (9)	H 30 (0)	
DT14	H 32 (2)	H 2D (-)	
DT15	H 2D (-)	H 35 (5)	
DT16	H 30 (0)	H 32 (2)	
DT17	H 34 (4)	H 31 (1)	
DT18	H 31 (1)	H 20 (␣)	
DT19	H 3A (:)	H 32 (2)	
DT20	H 34 (4)	H 35 (5)	
DT21	H 33 (3)	H 3A (:)	
DT22		H 31 (1)	

Example 2)

[S1]...DT100 [S2]...DT0 [D]...DT150

・Output example `Wed␣08-06-2020␣23:20:05`

DT100	U 20	Year
DT101	U 6	Month
DT102	U8	Day
DT103	U 23	Hour
DT104	U 20	Minute
DT105	U5	Second
DT106	U3	Day of the week

DT0	H 4012

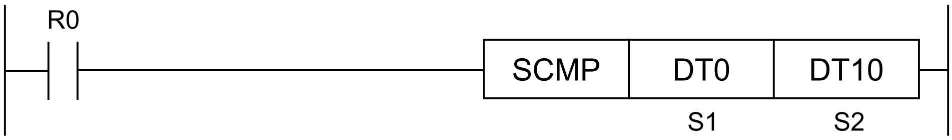
DT150	U 23		No. of bytes
DT151	H 65 (e)	H 57(W)	
DT152	H 20 (␣)	H 64 (d)	
DT153	H 38 (8)	H 30 (0)	
DT154	H 30 (0)	H 2D (-)	
DT155	H 2D (-)	H 36 (6)	
DT156	H 30 (0)	H 32 (2)	
DT157	H 30 (0)	H 32 (2)	
DT158	H 32 (2)	H 20 (␣)	
DT159	H 3A (:)	H 33 (3)	
DT160	H 30 (0)	H 32 (2)	
DT161	H 30 (0)	H 3A (:)	
DT162		H 35 (5)	

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the parameter of [S1] is out of the setting range.
	To be set when the parameter of [S2] is out of the setting range.
	To be set when the range between [S1] to [S1+6] is out of the accessible range.
	To be set when the destination range is outside the accessible range.

14.16 SCMP (String Compare)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Character string 1 for comparison
Example 1	Character string 2 for comparison

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●												●
Example 1	●	●	●	●			●	●	●												●

■ Outline of operation

- This instruction compares the string specified by [S1] and the string specified by [S2]. The comparison result is output to the system relays SRA to SRC (assessment flags for the comparison instruction).
- Comparison flags (system relays SRA to SRC) are processed as follows.

	SRA	SRB	SRC
	>	=	<
[S1] < [S2]	OFF	OFF	ON
[S1] = [S2]	OFF	ON	OFF
[S1] > [S2]	ON	OFF	OFF

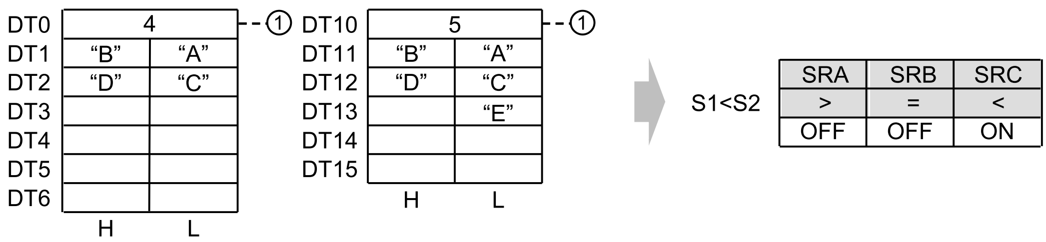
- If the numbers of characters to be compared are different, they are processed as follows.

[S1]		[S2]
"ABCDE"	=	"ABCDE"
“ABCD”	<	"ABCDE"
"B"	>	"ABCDE"

■ Processing

Example) Comparing strings "ABCD" and "ABCDE", which are stored in the data register

[S1]...DT0 [S2]...DT10 SRA...OFF SRB...OFF SRC...ON



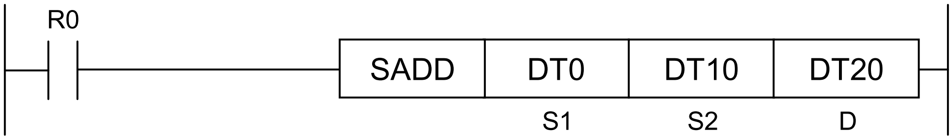
(1)	Character count

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when the string range specified by [S1] or [S2] is out of the accessible range.
SRA (>)	Depending on the comparison result
SRB (=)	
SRC (<)	

14.17 SADD (String Addition)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of String 1 to be connected
S2	Starting device address of String 2 to be connected
D	Starting device address to store the connected string

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●												●
S2	●	●	●	●			●	●	●												●
D	●	●	●	●			●	●	●												●

■ Outline of operation

- This instruction combines the string specified by [S1] with the string specified by [S2], and sets the combined string to the device address specified by [D].
- The maximum number of characters for the result is 4096 characters.

■ Processing

Example) Combine the strings of DT0 and DT10, and set the result to DT20.

Byte address	Higher bytes	Lower bytes
DT0	5 (No. of characters)	
DT1	"B"	"A"
DT2	"D"	"C"
DT3		"E"
DT4		
DT5		
DT6		

+

Byte address	Higher bytes	Lower bytes
DT10	3 (No. of characters)	
DT11	"2"	"1"
DT12		"3"
DT13		
DT14		
DT15		
DT16		

=

Byte address	Higher bytes	Lower bytes
DT20	8 (No. of characters)	
DT21	"B"	"A"
DT22	"D"	"C"
DT23	"1"	"E"
DT24	"3"	"2"
DT25		
DT26		

■ Precautions for programming

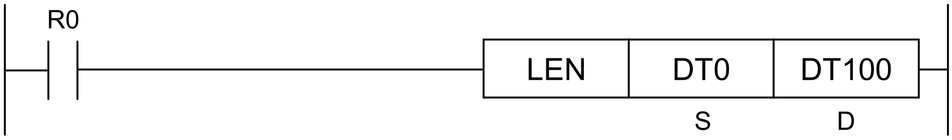
- The character data of the [D] area from before performing the operation is overwritten.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the string range specified by [S1] or [S2] is out of the accessible range.
	To be set when the destination range is outside the accessible range.
	To be set when the connected string exceeds the maximum number of characters.

14.18 LEN (Obtainment of String Length)

■ Ladder diagram



■ List of operands

Operand	Description
S	Starting device address of the string
D	Starting device address to store the string length

■ Available devices (●: Available)

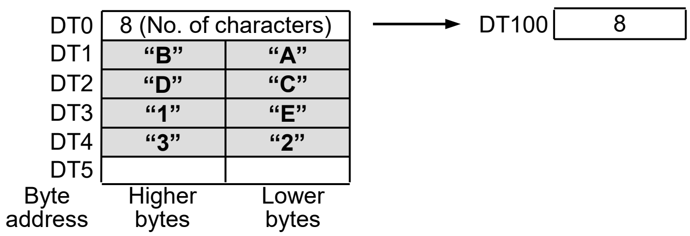
Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●	●												●
D	●	●	●	●			●	●	●		●										●

■ Outline of operation

- This instruction sets the number of characters stored in the beginning of the character string specified by [S] to the device address specified by [D].

■ Processing

Example) Set the number of characters of DT0 in DT100



■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when the obtained number of characters exceeds 4096.

Name	Description
	To be set when the string range specified by [S] is out of the accessible range.

14.19 LENGTH (Search String Length (Terminating NULL))

■ Instruction format

■ List of operands

Operand	Description
S1	The starting address of the string to be searched is specified.
S2	The starting device address for storing the maximum searched string length or a constant is specified. (Available range: 1 to 4096)
D	The starting address storing string length is specified.

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●													●
S2	●	●	●	●	●	●	●	●								●	●				●
D	●	●	●	●			●	●													●

■ Outline of operation

This instruction detects a termination character (null) from a string and acquires the number of characters. Null characters are not included in the number of characters.

■ Processing

- Searches the length of the string specified by [S1].
- Searches NULL characters for the number of characters specified by [S2] (maximum string length) from [S1], and stores the string length excluding NULL characters in [D] (result) when NULL characters exist.
- When there is no NULL character within the maximum string length [S2], the CY (SR9) flag is set, and the maximum string length is stored in the result [D].
- When a NULL character is found, the CY (SR9) flag is not cleared.

■ Example of processing

[S1] … DT0, [S2] … U10, [D] … DT100

Example 1) With NULL

[S1] device content "0123456" + NULL

Processing result String length … 7

	Value
DT0	3130h
DT1	3332h
DT2	3534h
DT3	0036h
DT4	****h

→

	Value
DT100	0007h

SR9 … No change

Example 2) Without NULL

[S1] device content "0123456789"

Processing result String length … 10

	Value
DT0	3130h
DT1	3332h
DT2	3534h
DT3	3736h
DT4	3938h

→

	Value
DT100	000Ah

SR9 … ON

■ Precautions for programming

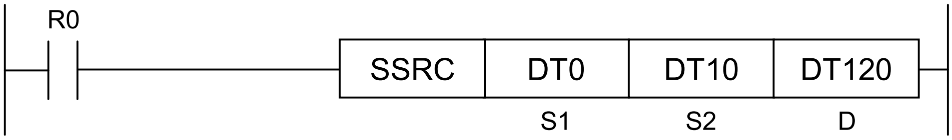
- Error flags are not cleared even when normal operation is performed.
- Use ERR instruction for clearing error flags.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when the maximum string length specified by [S2] is out of the range.
	To be set when the maximum string length specified by [S2] exceeds the [S1] area.
CY(SR9)	To be set when a NULL character is not included in the string of [S1].

14.20 SSRC (String Search)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of the string data to be searched for
S2	Starting device address of the string to be searched
D	Starting device address to store the search result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●												●
S2	●	●	●	●			●	●	●												●
D	●	●	●	●			●	●	●		●										●

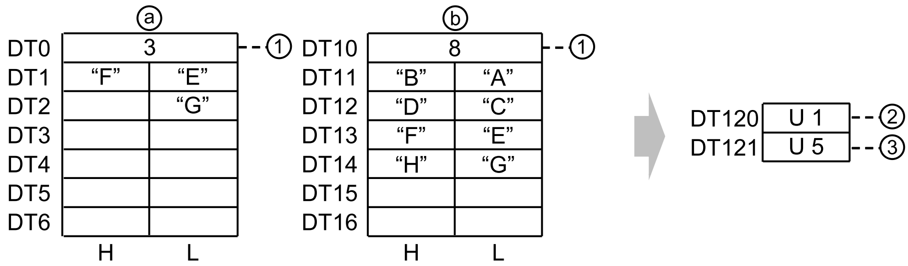
■ Outline of operation

- This instruction searches the character data specified by [S1] from the character string specified by [S2].
- As for the search result, the number of the same character data is stored in the device address specified by [D], and the first matched relative position (byte units) is stored in [D]+1.

■ Processing

Example 1) Searching for string "EFG" stored in DT0, in the string table from DT11

[S1]...DT0 [S2]...DT10 [S2]...DT120

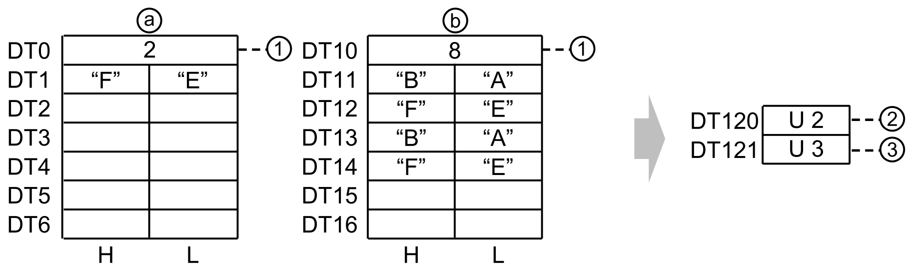


(a)	Area in which the string to be searched for is stored	(b)	String table that is searched
(1)	Character count	(2)	Number of matching strings
		(3)	Relative position of matching string

(Note 1) Using the low byte of DT11 at the beginning of the string table as a reference, the relative position of the low byte of DT13 is calculated as a value 5, based on where it matches with string "EFG".

Example 2) When the string "EF" being searched for is found in two locations on the string table after DT11

[S1]...DT0 [S2]...DT10 [S2]...DT120



(a)	Area in which the string to be searched for is stored	(b)	String table that is searched
(1)	Character count	(2)	Number of matching strings
		(3)	Relative position of matching string (Note 1)

(Note 1) Using the low byte of DT11 at the beginning of the string table as a reference, the relative position for the low byte of DT12 is calculated as a value 3, based on where it matches with string "EF" first.

■ Precautions for programming

For [S1], the number of characters to search for in a string, specify the number of characters to be searched for.

In the following figure, 1 is specified for the number of characters, and the string "A" is searched for. When 2 is specified for the number of characters, the string "AB" is searched for.

(1)	Character count

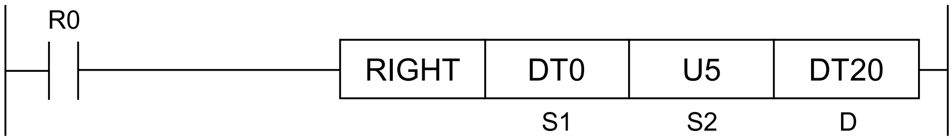
DT0	2		1
DT1	"B"	"A"	
DT2	"D"	"C"	

■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).
SR8	To be set when the number of characters [S1] is larger than [S2].
(ER)	To be set when the string range specified by [S1] or [S2] is out of the accessible range.

14.21 RIGHT (Takeout of the Right Side of a String)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of the source data
S2	Number of characters to be taken out (available data range: 1 to 4096)
D	Starting device address to store the result that is taken out

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●												●
S2	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●												●

■ Outline of operation

- This instruction takes out the characters for the number of characters specified by [S2] from the right side (end of the character data) of the character string specified by [S1], and stores it in the device address specified by [D].

■ Processing

Example 1) Taking out the last five characters from the DT0 string to transfer them to DT20

[S1]...DT0 [S2]...U5 [D]...DT20

Characters to search

DT0	8 (No. of characters)	
DT1	"B"	"A"
DT2	"D"	"C"
DT3	"1"	"E"
DT4	"3"	"2"
DT5		
DT6		

→

String table that is searched

DT20	5 (No. of characters)	
DT21	**"E"**	**"D"**
DT22	**"2"**	**"1"**
DT23		**"3"**
DT24		
DT25		
DT26		

Byte address High Low Byte address High Low

Example 2) The number of characters of [S2] is larger than the number of characters in the string of [S1]

[S1]...DT0 [S2]...U7 [D]...DT20

Characters to search

DT0	3 (No. of characters)	
DT1	"B"	"A"
DT2		"C"
DT3		
DT4		
DT5		

→

String table that is searched

DT20	3 (No. of characters)	
DT21	**"B"**	**"A"**
DT22		**"C"**
DT23		
DT24		
DT25		

Byte address High Low Byte address High Low

■ Precautions for programming

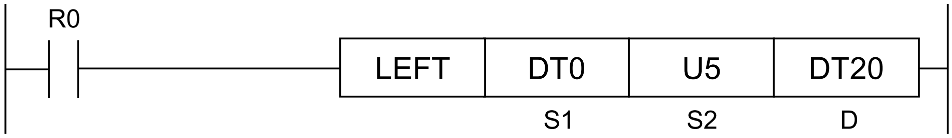
- The character data of the [D] area from before performing the operation is overwritten.
- When the number of characters of [S2] is greater than the number of characters in the string of [S1], the transferal is performed for the number of characters of [S1].

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the destination range is outside the accessible range.
	To be set when [S2] (number of characters) is out of the range.

14.22 LEFT (Takeout of the Left Side of a String)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of the source data
S2	Number of characters to be taken out (available data range: 1 to 4096)
D	Starting device address to store the result that is taken out

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●												●
S2	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●												●

■ Outline of operation

- This instruction takes out the characters for the number of characters specified by [S2] from the left side (beginning of the character data) of the character string specified by [S1], and stores it in the device address specified by [D].

■ Processing

Example 1) Taking out the first five characters from the DT0 string to transfer them to DT20

[S1]...DT0 [S2]...U5 [D]...DT20

Characters to search

DT0	8 (No. of characters)	
DT1	"B"	"A"
DT2	"D"	"C"
DT3	"1"	"E"
DT4	"3"	"2"
DT5		
DT6		

→

String table that is searched

DT20	5 (No. of characters)	
DT21	**"B"**	**"A"**
DT22	**"D"**	**"C"**
DT23		**"E"**
DT24		
DT25		
DT26		

Byte address High Low Byte address High Low

Example 2) The number of characters of [S2] is larger than the number of characters in the string of [S1]

[S1]...DT0 [S2]...U7 [D]...DT20

Characters to search

DT0	3 (No. of characters)	
DT1	"B"	"A"
DT2		"C"
DT3		
DT4		
DT5		
DT6		

→

String table that is searched

DT20	3 (No. of characters)	
DT21	**"B"**	**"A"**
DT22		**"C"**
DT23		
DT24		
DT25		
DT26		

Byte address High Low Byte address High Low

■ Precautions for programming

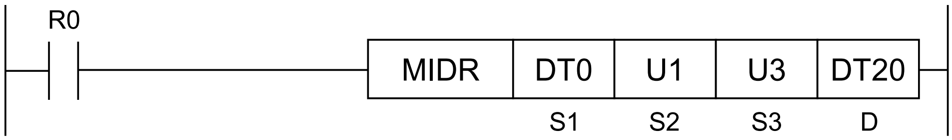
- The character data of the [D] area from before performing the operation is overwritten.
- When the number of characters of [S2] is greater than the number of characters in the string of [S1], the transferal is performed for the number of characters of [S1].

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the destination range is outside the accessible range.
	To be set when [S2] (number of characters) is out of the range.

14.23 MIDR (Data Read from a Given Position in the String)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of the source data
S2	Starting position (available data range: 0 to 4095)
S3	Number of characters to be taken out (available data range: 1 to 4096)
D	Starting device address to store the result that is taken out

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●												●
S2	●	●	●	●	●	●	●	●	●	●	●					●	●				●
S3	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●												●

■ Outline of operation

- This instruction takes out data for the number of characters specified by [S3] from the position specified by [S2] of the character string specified by [S1], and stores it in the device address specified by [D].

Processing

Example 1) Taking out three characters from the first byte (second character) of the DT0 string to transfer them to DT20

[S1]...DT0 [S2]...U1 [S3]...U3 [D]...DT20

Characters to search

	High	Low
DT0	8 (No. of characters)	
DT1	**"B"**	"A"
DT2	**"D"**	**"C"**
DT3	"1"	"E"
DT4	"3"	"2"
DT5		
DT6		

String table that is searched

	High	Low
DT20	3 (No. of characters)	
DT21	**"C"**	**"B"**
DT22		**"D"**
DT23		
DT24		
DT25		
DT26		

Byte address High Low Byte address High Low

Example 2) When the [S2] position + the number of characters of [S3] is larger than the number of characters in the string of [S1]

[S1]…DT0 [S2]…U5 [S3]…U5 [D]…DT20

Characters to search

	High	Low
DT0	8 (No. of characters)	
DT1	"B"	"A"
DT2	"D"	"C"
DT3	**"1"**	"E"
DT4	**"3"**	**"2"**
DT5		
DT6		

String table that is searched

	High	Low
DT20	3 (No. of characters)	
DT21	**"2"**	**"1"**
DT22		**"3"**
DT23		
DT24		
DT25		
DT26		

Byte address High Low Byte address High Low

Precautions for programming

- The character data of the [D] area from before performing the operation is overwritten.
- When the number of characters of [S3] is larger than the number of characters of the [S1] string starting from the [S2] position, the transferal is performed for the number of characters of [S1].
- The [S2] position should be specified by a number counted from low order in sequence, putting the least significant byte as K0 (Byte 0): 0, 1, 2 and so on.

Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the destination range is outside the accessible range.
	To be set when [S3] (number of characters) is out of the range.
	To be set when the number of characters for [S1] is smaller than [S2].

14.24 MIDW (Rewrite from a Given Position in the String)

■ Ladder diagram

■ List of operands

Operand	Description
S1	Starting device address of the source data
S2	Number of characters (available data range: 1 to 4096)
D	Destination starting device address
n	Starting position of the destination string (available data range: 0 to 4095)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●												●
S2	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●												●
n	●	●	●	●	●	●	●	●	●	●	●					●	●				●

■ Outline of operation

- This instruction takes out data for the number of characters specified by [S2] from the character string specified by [S1], and transfers it to the position [n] of the character string specified by [D].

■ Processing

Example 1) Taking out three characters from the DT0 string to transfer them to the first byte (second character) of the DT20 string

[S1]...DT0 [S2]...U3 [D]...DT20 [n]...U1

Character string

Byte address	High	Low
DT0	8 (No. of characters)	
DT1	**"B"**	**"A"**
DT2	"D"	**"C"**
DT3	"F"	"E"
DT4	"H"	"G"
DT5		
DT6		

Transfer destination

Byte address	High	Low
DT20	8 (No. of characters)	
DT21	"2"	"1"
DT22	"4"	"3"
DT23	"6"	"5"
DT24	"8"	"7"
DT25		
DT26		

Transfer destination after execution

Byte address	High	Low
DT20	8 (No. of characters)	
DT21	**"A"**	"1"
DT22	**"C"**	**"B"**
DT23	"6"	"5"
DT24	"8"	"7"
DT25		
DT26		

Example 2) The number of characters of [S2] is larger than the number of characters in the string of [S1]

[S1]...DT0 [S2]...U5 [D]...DT20 [n]...U1

Character string

Byte address	High	Low
DT0	4 (No. of characters)	
DT1	**"B"**	**"A"**
DT2	**"D"**	**"C"**
DT3		
DT4		
DT5		
DT6		

Transfer destination

Byte address	High	Low
DT20	8 (No. of characters)	
DT21	"2"	"1"
DT22	"4"	"3"
DT23	"6"	"5"
DT24	"8"	"7"
DT25		
DT26		

Transfer destination after execution

Byte address	High	Low
DT20	8 (No. of characters)	
DT21	**"A"**	"1"
DT22	**"C"**	**"B"**
DT23	"6"	**"D"**
DT24	"8"	"7"
DT25		
DT26		

Precautions for programming

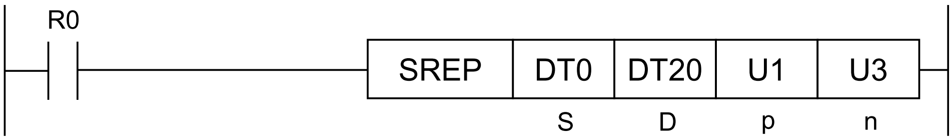
- The character data of the [D] area from before performing the operation is overwritten.
- If the number of characters in [S2] is greater than the number of characters in the [S1] character string, the number of characters of the [S1] character string is sent.
- The [n] position should be specified by a number counted from low order in sequence, putting the least significant byte as K0 (Byte 0): 0, 1, 2 and so on.

Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [S2] (number of characters) is out of the range.
	To be set when the number of characters of [D] is smaller than [n].

14.25 SREP (Replacement of a String)

■ Ladder diagram



■ List of operands

Operand	Description
S	Starting device address of the source string
D	Starting device address of the destination string
p	Replacement start position of the destination string (available data range: 0 to 4095)
n	Number of characters to be replaced (available data range: 1 to 4096)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●	●												●
D	●	●	●	●			●	●	●												●
p	●	●	●	●	●	●	●	●	●	●	●					●	●				●
n	●	●	●	●	●	●	●	●	●	●	●					●	●				●

■ Outline of operation

- This instruction replaces the string specified by [S] with the characters specified by [n] from the position [p] in the string specified by [D].

■ Processing

Example 1) Replacing the DT0 string with three characters from the 1st byte (2nd character) of DT20

[S1]...DT0 [D]...DT20 [p]...U1 [n]...U3

Character string

DT0	5 (No. of characters)	
DT1	**"B"**	**"A"**
DT2	**"D"**	**"C"**
DT3		**"E"**
DT4		
DT5		
DT6		
Byte address	High	Low

Transfer destination

DT20	8 (No. of characters)	
DT21	**"2"**	"1"
DT22	**"4"**	**"3"**
DT23	"6"	"5"
DT24	"8"	"7"
DT25		
DT26		
Byte address	High	Low

Transfer destination after execution

DT20	**10 (No. of characters)**	
DT21	**"A"**	"1"
DT22	**"C"**	**"B"**
DT23	**"E"**	**"D"**
DT24	"6"	"5"
DT25	"8"	"7"
DT26		
Byte address	High	Low

*In this case, the original data of the first byte to the fourth byte of DT20 that is the destination is deleted.
As a result, the number of characters increases from 8 characters to 10 characters.

Example 2) The number of characters [n] is larger than the number of characters in the string [S1] as from the position specified by [p].

[S1]...DT0 [D]...DT20 [p]...U4 [n]...U8

Character string

DT0	5 (No. of characters)	
DT1	**"B"**	**"A"**
DT2	**"D"**	**"C"**
DT3		**"E"**
DT4		
DT5		
DT6		
Byte address	High	Low

Transfer destination

DT20	10 (No. of characters)	
DT21	"2"	"1"
DT22	"4"	"3"
DT23	**"6"**	**"5"**
DT24	**"8"**	**"7"**
DT25	**"0"**	**"9"**
DT26		
Byte address	High	Low

Transfer destination after execution

DT20	9 (No. of characters)	
DT21	"2"	"1"
DT22	"4"	"3"
DT23	**"B"**	**"A"**
DT24	**"D"**	**"C"**
DT25		**"E"**
DT26		
Byte address	High	Low

Precautions for programming

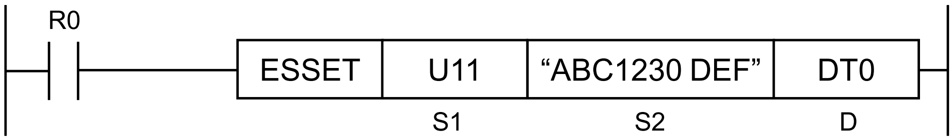
- Character data of the [D] area from before performing the operation is not cleared. (It is overwritten.)
- If the number of characters [n] is larger than the number of characters in the string of [S1] as from the position specified by [p], then the replacement is performed for the number of characters of the string [S1] as from the position specified by [p].
- The [p] position should be specified by a number counted from low order in sequence, putting the least significant byte as K0 (Byte 0): 0, 1, 2 and so on.

Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the number of characters > string size.
	To be set when the number of characters of [D] is smaller than [n].
	To be set when [S2] (number of characters) is out of the range.

14.26 ESSET (Conversion: Character Constant → ASCII Code: With Storage Area Size)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Storage area size (available range: U1 to U65534)
S2	Character constant to be converted (available range: 0 to 256 characters)
D	Destination starting device address

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●							●	●				●
S2																				●	
D	●	●	●	●			●	●													●

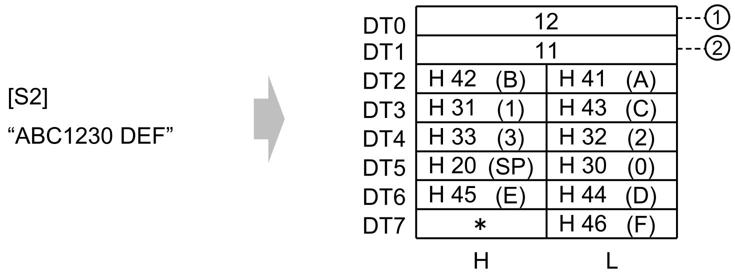
■ Outline of operation

- This instruction stores the storage area size specified by [S1] in [D].
- The character constant specified by [S2] is converted to ASCII code, the number of characters (1 word) is stored in [D+1], and the character data converted to ASCII is stored in subsequent areas in sequence from the low byte.
- Character constants should be put between "" (double quotations) for specification.
- From 0 to 256 characters can be specified for a character constant.
- When there are double quotation marks enclosing no character, it is recognized as a NULL character
- When setting, NULL (00) is not added to the end of characters.

■ Processing

Example 1) Converting the string "ABC1230 DEF" (11 characters including a space)

[S1]...U12 [S2]...”ABC1230 DEF” [D]...DT0

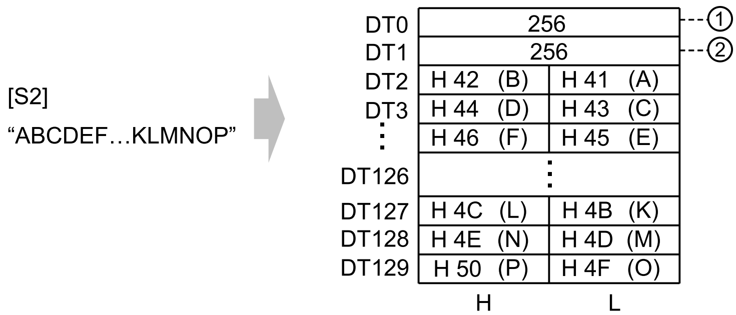


(Note 1) The data (*) out of the destination range, the high byte of DT7, does not change.

(1)	Storage area size	(2)	Character count

Example 2) Setting 256 characters to DT0, repeating a set of the 16 characters from A to P

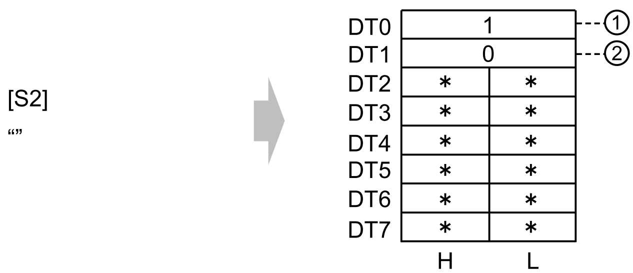
[S1]...U256 [S2]... "ABCDEF…KLMNOP" [D]...DT0



(1)	Storage area size	(2)	Character count

Example 3) Converting 0 characters of the string "" (repeated double quotation marks)

[S1]...U1 [S2]... "" [D]...DT0



(Note 1) The data (*) that is out of the destination range, DT2 to DT7 , does not change.

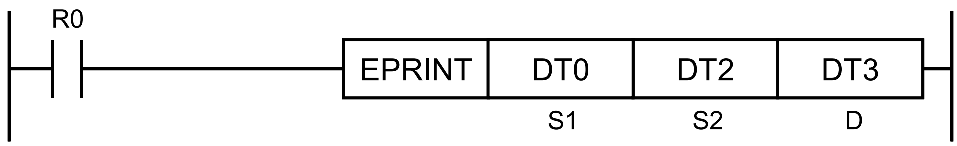
(1)	Storage area size	(2)	Character count

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when securing a storage area large enough to store the area starting with D causes the size to be out of the accessible range.
	To be set when 0 or 65535 is set for S1.
	To be set when the number of characters is larger than the storage area size.

14.27 EPRINT (Text Creation: With Storage Area Size)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting address of the device storing the string data which indicates the create text form or a character constant
S2	Starting address of the device storing the data to be output in text format
D	Starting address of the device storing the text

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●			●	●													●
D	●	●	●	●			●	●													●

■ Outline of operation

- This instruction is used for creating texts of mails, etc.

■ Processing

- ASCII code texts are created according to the specified text creation form.
- The text creation form can be specified using the operand [S1], ESSET instruction or the mail setting screen of FPWIN GR7. Created texts can be connected using the ESADD instruction.
- The maximum size of a mail text is 4096 bytes for sending an event mail, and 256 bytes for sending a logging/trace mail.

■ Operand [S1] setting

- Specify the device address storing the text creation form or character constant (max. 256 characters).
- The text creation form is composed of a main text, conversion form (such as %d, %e), linefeed code (\n) and tab code (\t).

Example of the create text form

This example includes a tab code, a body (a conversion form for 1 datum is inserted), and a linefeed code.

"¥t Floor A is %d degrees C.¥n

Line feed code

Conversion form

Text

Tab code

- Tab code (\t) is converted to ASCII code HT (09h).
- The body is converted to the supported ASCII code or Shift JIS code.
- In the part where a conversion form is inserted, the output data specified by [S2] is stored as ASCII code, according to the conversion form. For more information about the conversion form, refer to"14.14 PRINT/EPRINT Instruction Shared Conversion Form Table".
- Linefeed code (\n) is converted to ASCII code CR+LF (0A0Dh).

Restrictions

- Up to 4096 characters can be specified for the text creation form. An operation error occurs when it exceeds 4096 characters.
- Up to 16 digits can be specified for one conversion form. An operation error occurs when it exceeds 16 digits.
- The maximum number of characters after conversion for a single datum excluding %s and %S is 32. An operation error occurs when it exceeds 32 characters.
- The maximum number of characters for %s or %S after conversion is 4096.
- All strings that are not recognized as conversion forms are treated as main texts.

  Example: Conversion forms that do not allow upper case characters (%D, etc.)

  Character strings that contain characters not recognized as discriminant characters of conversion forms (%A, %Z, etc.)
- To enter "%" in the body, specify "%%" (% x 2).

■ Operand [S2] setting

- Specify the starting address of the device storing the data to be output in the create text form.
- Arrange conversion data in the order of the conversion form specified for [S1].
- For the character data for %s and %S, specify data storing the number of (1-byte) characters in the starting word. String data can be set with the ESSET instruction.

ESSET U2 "Floor" DT112
S1 = "%d %u %x %b %f %e %Lg %s"
S2 = DT100
Result: -1 65535 ffff 1000 123.4567 123.4567 123.456789 Floor

	value		
DT100	FFFFh	Data for %d	
DT101	FFFFh	Data for %u	
DT102	FFFFh	Data for %x	
DT103	1000h	Data for %b	
DT104	SF 123.4567	Data for %f	
DT105			
DT106	SF 123.4567	Data for %e	
DT107			
DT108	DF 123.456789	Data for %Lg	
DT109			
DT110			
DT111			
DT112	2	Data for %s	Storage area size
DT113	2		No. of bytes
DT114	4b8a(Floor)		Data part

■ Setting example

Example 1) When inserting into the text two conversion forms (%d) that represent 16-bit signed integers and a linefeed code (\n)

In the place of the conversion form (%d), the ASCII code that is equivalent to the integer data specified by [S2] is inserted. If a number of digits is not specified for the conversion form, the number of converted data and the size of the storage area will vary according to the value [S2].

- Image of mail text

Floor A: 25°C
Floor B: 28°C

- Setting values

S1="Floor A: %d°C¥nFloor B: %d°C"
S2=DT100
D=DT200

DT100	K 25
DT101	K 28
DT102	

→

DT200	U 32		Storage area size (*1)
DT201	U 30		No. of bytes
DT202	H 6C (l)	H 46 (F)	Data part
DT203	H 6F (o)	H 6F (o)	
DT204	H 20 (SPACE)	H 72 (r)	
DT205	H 3A (:)	H 41 (A)	
DT206	H 32 (2)	H 20 (SPACE)	
DT207	H C2 (°)	H 35 (5)	
DT208	H 43 (C)	H B0 (°)	
DT209	H 0A (LF)	H 0D (CR)	
DT210	H 6C (l)	H 46 (F)	
DT211	H 6F (o)	H 6F (o)	
DT212	H 20 (SPACE)	H 72 (r)	
DT213	H 3A (:)	H 42 (B)	
DT214	H 32 (2)	H 20 (SPACE)	
DT215	H C2 (°)	H 38 (8)	
DT216	H 43 (C)	H B0 (°)	
DT217	(*2)		

(Note 1) The start area (storage area size) for [D] is set before executing this instruction.

(Note 2) The data out of the destination range does not change.

Example 2) When inserting into the text a conversion form (%d) that represents a 16-bit signed integer

In the place of the conversion form (%d), the ASCII code that is equivalent to the integer data specified by [S2] is inserted. If a number of digits is not specified for the conversion form, the number of converted data and the size of the storage area will vary according to the value [S2].

- Image of mail text

Production volume: 5

- Setting values
  S1="Production volume: %u"
  S2=DT1
  D=DT50

DT1	U 5
DT2	
DT3	

DT50	U 22		Storage area size (*1)
DT51	U 20		No. of bytes
DT52	H 72 (r)	H 50 (P)	Data part
DT53	H64 (d)	H 6f (o)	
DT54	H 63 (c)	H 75 (u)	
DT55	H 69 (i)	H 74 (t)	
DT56	H 6e (n)	H 6f (o)	
DT57	H 76 (v)	H 20 (SPACE)	
DT58	H 6c (l)	H 6f (o)	
DT59	H 6d (m)	H 75 (u)	
DT60	H 3a (:)	H 65 (e)	
DT61	H 35 (5)	H 20 (SPACE)	
DT62	(*2)		

(Note 1) The start area (storage area size) for [D] is set before executing this instruction.

(Note 2) The data out of the destination range does not change.

Example 3) When inserting into the text a horizontal tab code (\t: H09)

In the place of the conversion form (\t), the ASCII code that is equivalent to the horizontal tab code is inserted. If a conversion form is not included in [S1], the data for [S2] will have no effect on the conversion results.

- Image of mail text

Normal operation

- Setting values

S1="¥tNormal operation"
S2=DT1
D=DT50

DT1	(*3)
DT2	(*3)
DT3	(*3)

→

DT50	U 18		Storage area size (*1)
DT51	U 17		No. of bytes
DT52	H 4e (N)	H 09 (HT)	Data part
DT53	H 72 (r)	H 6f (o)	
DT54	H 61 (a)	H 6d (m)	
DT55	H 20 (SPACE)	H 6c (l)	
DT56	H 70 (p)	H 6f (o)	
DT57	H 72 (r)	H 65 (e)	
DT58	H 74 (t)	H 61 (a)	
DT59	H 6f (o)	H 69 (i)	
DT60	(*2)	H 6e (n)	

(Note 1) The start area (storage area size) for [D] is set before executing this instruction.

(Note 2) The data out of the destination range does not change.

(Note 3) The [S2] data has no effect on the conversion results.

Example 4) When inserting into the text two conversion forms (%s) that represent strings

In the place of the conversion form (%s), the ASCII code that is equivalent to the string data specified by [S2] is inserted. If a number of digits is not specified for the conversion form, the number of converted data and the size of the storage area will vary according to the value [S2]. Use the ESSET instruction to reset the [S2] string data.

- Image of mail text

Location: Nagoya, Aichi

- Setting values

S1="Location: %s, %s"
S2=DT1
D=DT50

DT1	U 8	
DT2	U 6	
DT3	H 61 (a)	H 4E (N)
DT4	H 6F (o)	H 67 (g)
DT5	H 61 (a)	H 79 (y)
DT6	**	
DT7	U 8	
DT8	U 5	
DT9	H 69 (i)	H 41 (A)
DT10	H 68 (h)	H 63 (c)
DT11	**	H 69 (i)
DT12	**	

→

DT50	U 24		Storage area size (*1)
DT51	U 23		No. of bytes
DT52	H 6F (o)	H 4C (L)	Data part
DT53	H 61 (a)	H 63 (c)	
DT54	H 69 (i)	H 74 (t)	
DT55	H 6E (n)	H 6F (o)	
DT56	H 20 (SPACE)	H 3A (:)	
DT57	H 61 (a)	H 4E (N)	
DT58	H 6F (o)	H 67 (g)	
DT59	H 61 (a)	H 79 (y)	
DT60	H 20 (SPACE)	H 2C (,)	
DT61	H 69 (i)	H 41 (A)	
DT62	H 68 (h)	H 63 (c)	
DT63	(*2)	H 69 (i)	

(Note 1) The start area (storage area size) for [D] is set before executing this instruction.

(Note 2) The data out of the destination range does not change.

Example 5) When inserting into the text two conversion forms (%s) that represent strings

In the place of the conversion form (%s), the ASCII code that is equivalent to the string data specified by [S2] is inserted. The number of digits of the data to insert, right align, and left align are specified. Use the ESSET instruction to reset the [S2] string data.

- Image of mail text

Location: [␣␣Nagoya], [Aichi␣␣␣] (*3)(*4)

Right-aligned Left-aligned

- Setting values

S1="Location: %-8s, %8s"
S2=DT1
D=DT50

DT1	U 10	
DT2	U 6	
DT3	H 61 (a)	H 4E (N)
DT4	H 6F (o)	H 67 (g)
DT5	H 61 (a)	H 79 (y)
DT6	**	
DT7	U 10	
DT8	U 8	
DT9	H 69 (i)	H 41 (A)
DT10	H 68 (h)	H 63 (c)
DT11	**	H 69 (i)
DT12	**	

→

DT50	U 32		Storage area size (*1)
DT51	U 32		No. of bytes
DT52	H 6F (o)	H 4C (L)	Data part
DT53	H 61 (a)	H 63 (c)	
DT54	H 69 (i)	H 74 (t)	
DT55	H 6E (n)	H 6F (o)	
DT56	H 20 (SPACE)	H 3A (:)	
DT57	H 20 (SPACE)	H 5B ([)	
DT58	H 4E (N)	H 20 (SPACE)	
DT59	H 67 (g)	H 61 (a)	
DT60	H 79 (y)	H 6F (o)	
DT61	H 5D (])	H 61 (a)	
DT62	H 20 (SPACE)	H 2C (,)	
DT63	H 41 (A)	H 5B ([)	
DT64	H 63 (c)	H 69 (i)	
DT65	H 69 (i)	H 68 (h)	
DT66	H 20 (SPACE)	H 20 (SPACE)	
DT67	H 5D (])	H 20 (SPACE)	

(Note 1) The start area (storage area size) for [D] is set before executing this instruction.

(Note 2) The data out of the destination range does not change.

(Note 3) If %s (which represents string) is specified, it is left-aligned by default. If a minus sign is added to the value for [S1], it is right-aligned.

(Note 4) The ␣ symbol in figures represents a space.

Example 6) Combining the conversion forms (%s), (%d), and (%s) which represent strings, and inserting a minus sign (-)

In place of the conversion forms (%s), (%d), and (%s), ASCII code that is equivalent to the data specified by [S2] is inserted. Insert a minus sign. Use the ESSET instruction to reset the [S2] string data area.

- Image of mail text

Category: A-1-a

- Setting values

S1="Category: %s-%d-%s"
S2=DT1
D=DT50

DT1	U 2	
DT2	U 1	
DT3	**	H 41 (A)
DT4	H0001	
DT5	U 2	
DT6	U 1	
DT7	**	H 61 (a)

→

DT50	U 16		Storage area size (*1)
DT51	U 15		No. of bytes
DT52	H 61 (a)	H 43 (C)	Data part
DT53	H 65 (e)	H 74 (t)	
DT54	H 6F (o)	H 67 (g)	
DT55	H 79 (y)	H 72 (r)	
DT56	H 20 (SPACE)	H 3A (:)	
DT57	H 2D (-)	H 41 (A)	
DT58	H 2D (-)	H 31 (1)	
DT59	(*2)	H 61 (a)	

(Note 1) The start area (storage area size) for [D] is set before executing this instruction.

(Note 2) The data out of the destination range does not change.

Example 7) Combining the conversion forms (%s) and (%d) which represent strings to insert them

In the places of conversion forms (%s) and (%d), the ASCII code that is equivalent to the data array specified by [S2] is inserted. Use the ESSET instruction to reset the [S2] string data area.

- Image of mail text

Nagoya city: 25°C

- Setting values

S1="%s: %d°C"
S2=DT1
D=DT50

CY	ON

DT1	U 8	
DT2	U 6	
DT3	H 61 (a)	H 4E (N)
DT4	H 6F (o)	H 67 (g)
DT5	H 61 (a)	H 79 (y)
DT6	**	
DT7	K 25	

→

DT50	U 18		Storage area size (*1)
DT51	U 18		No. of bytes
DT52	H 61 (a)	H 4E (N)	Data part
DT53	H 6F (o)	H 67 (g)	
DT54	H 61 (a)	H 79 (y)	
DT55	H 63 (c)	H 20 (SPACE)	
DT56	H 74 (t)	H 69 (i)	
DT57	H 3A (:)	H 79 (y)	
DT58	H 32 (2)	H 20 (SPACE)	
DT59	H C2 (°)	H 35 (5)	
DT60	H 43 (C)	H B0 (°)	

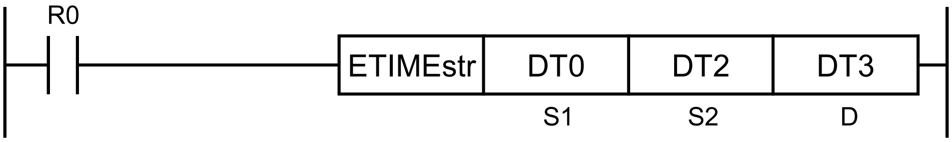
(Note 1) The start area (storage area size) for [D] is set before executing this instruction.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	Set when a value outside the range is specified for the parameter.
	To be set when the text creation form exceeds 4096 characters.
	To be set when texts exceed 4096 bytes.
	To be set when the conversion form is specified by a real number and conversion data is a non-real number.
	To be set when the size specified by the conversion form exceeds 32 characters. (Excluding the conversion forms %s, %S)
	To be set when the number of characters is larger than the storage area size for [S1] and [S2].
CY (SR9)	To be set when created texts are larger than the storage area size of [D].

14.28 ETIMEstr (Date and Time Character String Conversion: With Storage Area Size)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting address of the device that stores date and time information (7 words)
S2	Starting address of the device that stores conversion patterns (hex data, 1 word)
D	Starting address of the device that stores the string data as the conversion result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●		●	●	●													●
S2	●	●	●	●			●	●									●				●
D	●	●	●	●			●	●													●

■ Outline of operation

- This instruction converts data and time information to character strings.
- The date and time information to be output is year, month, day, day of the week, hour, minute, and second.
- This instruction can be used when date and time information is required for creating mail texts. This instruction is used in combination with the EPRINT instruction.

■ Processing

- Converts the date and time information specified by [S1] to ASCII code, and stores it as string data in the area that starts with [D].
- The conversion pattern is specified for [S2].
- The string data of the conversion result for [D] is output as the ASCII code (single byte) or Shift JIS code (double byte).

■ [S1]: Settings of date and time information

- Specify the starting address of the device that stores the date and time information.

- Specify SD50 if you want to output the current time and date of PLC.
- Checking the combination of year, month, day, or day of the week is not performed.
  Example) A setting of February 31 is not treated as an error. When SD50 is specified, the combination of year, month, day, or day of the week is correct.
- Always store the data in the order mentioned in the table below regardless of the conversion pattern of [S2].

Operand	Description	Specified range	Remarks
[S1]	Year	U0 to U99	The character string after conversion is 2000 to 2099.
[S1+1]	Month	U1 to U12	
[S1+2]	Day	U1 to U31	
[S1+3]	Hours	U0 to U23	
[S1+4]	Minutes	U0 to U59	
[S1+5]	Seconds	U0 to U59	
[S1+6]	Day of the week	U0 to U6	

■ [S2]: Conversion pattern settings

The conversion pattern is specified by 4-digit hex data.Refer to the page for the"14.15 TIMEstr (Date and Time Character String Conversion)"instruction.

■ Example of processing

Example 1)

[S1]…SD50 [S2]…DT0 [D]…DT10

Output image ... 2014/09/25(Thu)␣12:54:31

[S1]…SD50 [S2]…DT0 [D]…DT10

・Output example Thu␣ 09-25-2014␣ 12:54:31

SD50	U 14	Year
SD51	U 9	Month
SD52	U 25	Day
SD53	U 12	Hour
SD54	U 54	Minute
SD55	U 31	Second
SD56	U 4	Day of the week

DT0	H 0014	

DT10	U 24		Storage area size (*1)
DT11	U 23		Number of bytes
DT12	H 68 (h)	H 54 (T)	
DT13	H 20 (␣)	H 75 (u)	
DT14	H 39 (9)	H 30 (0)	
DT15	H 32 (2)	H 2D (-)	
DT16	H 2D (-)	H 35 (5)	
DT17	H 30 (0)	H 32 (2)	
DT18	H 34 (4)	H 31 (1)	
DT19	H 31 (1)	H 20 (␣)	
DT20	H 3A (:)	H 32 (2)	
DT21	H 34 (4)	H 35 (5)	
DT22	H 33 (3)	H 3A (:)	
DT23		H 31 (1)	

(Note 1) The start area (storage area size) for [D] is set before executing this instruction.

Example 2)

[S1]…DT100 [S2]…DT0 [D]…DT150

Output image ... Wed␣08-06-2020␣23:20:05

[S1]...DT100 [S2]...DT0 [D]...DT150

・Output example Wed␣ 08-06-2020␣ 23:20:05

DT100	U 20	Year
DT101	U 6	Month
DT102	U 8	Day
DT103	U 23	Hour
DT104	U 20	Minute
DT105	U 5	Second
DT106	U 3	Day of the week
DT0	H 4012	

DT150	U 24		Storage area size (*1)
DT151	U 23		Number of bytes
DT152	H 65 (e)	H 57 (W)	
DT153	H 20 (␣)	H 64 (d)	
DT154	H 38 (8)	H 30 (0)	
DT155	H 30 (0)	H 2D (-)	
DT156	H 2D (-)	H 36 (6)	
DT157	H 30 (0)	H 32 (2)	
DT158	H 30 (0)	H 32 (2)	
DT159	H 32 (2)	H 20 (␣)	
DT160	H 3A (:)	H 33 (3)	
DT161	H 30 (0)	H 32 (2)	
DT162	H 30 (0)	H 3A (:)	
DT163		H 35 (5)	

(Note 1) The start area (storage area size) for [D] is set before executing this instruction.

Example 3) When created character strings exceed the storage area size and CY is set

[S1]…SD50 [S2]…DT0 [D]…DT10

Output image ... 2014/09/25(Thu)␣12:54:31

[S1]...SD50 [S2]...DT0 [D]...DT10

・Output example 2014年09月25日(木)␣ 12時54分31秒

SD50	U 14	Year
SD51	U 9	Month
SD52	U 25	Day
SD53	U 12	Hour
SD54	U 54	Minute
SD55	U 31	Second
SD56	U 4	Day of the week
DT0	H 0014	

DT10	U 16		Storage area size (*1)
DT11	U 16		Number of bytes
DT12	H 30 (0)	H 32 (2)	
DT13	H 34 (4)	H 31 (1)	
DT14	H 4E94 (年)		
DT15	H 39 (9)	H 30 (0)	
DT16	H 8E8C (月)		
DT17	H 35 (5)	H 32 (2)	
DT18	H FA93 (日)		
DT19	H 96 (木)	H 28 (()	

CY	ON

(Note 1) The start area (storage area size) for [D] is set before executing this instruction.

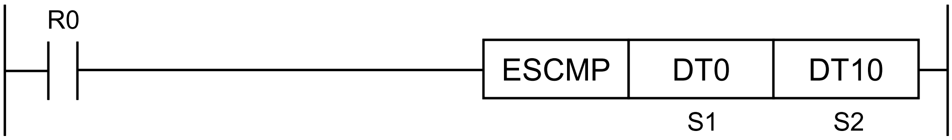
■ Flag operations

Name	Description
SR7	To be set in case of out-of-range values in indirect access (index modification).

Name	Description
SR8 (ER)	To be set when the parameter of [S1] is out of the setting range.
	To be set when the parameter of [S2] is out of the setting range.
	Set when the [S1] to [S1+6] range exceeds the accessible range.
	To be set when the destination range is outside the accessible range.
CY (SR9)	To be set when created character strings are larger than the storage area size for [D].

14.29 ESCMP (String Compare: With Storage Area Size)

■ Ladder diagram



■ List of operands

Operand	Description
S1	String 1 to be compared (available range: 0 to 65534; for character constant: 0 to 256 characters)
S2	String 2 to be compared (available range: 0 to 65534; for character constant: 0 to 256 characters)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	●
S2	●	●	●	●			●	●												●	●

(Note 1) Only 16-bit devices can be modified. (Character constants cannot be specified.)

■ Outline of operation

- This instruction compares the string specified by [S1] and the string specified by [S2]. The comparison result is output to the system relays SRA to SRC (assessment flags for the comparison instruction).
- The size of the storage area is not included in the judgment conditions.
- Comparison flags (system relays SRA to SRC) are operated as follows.

	SRA	SRB	SRC
	>	=	<
[S1] < [S2]	OFF	OFF	ON
[S1] = [S2]	OFF	ON	OFF
[S1] > [S2]	ON	OFF	OFF

- If the numbers of characters to be compared are different, they are processed as follows.

[S1]		[S2]
"ABCDE"	=	"ABCDE"
“ABCD”	<	"ABCDE"
"B"	>	"ABCDE"

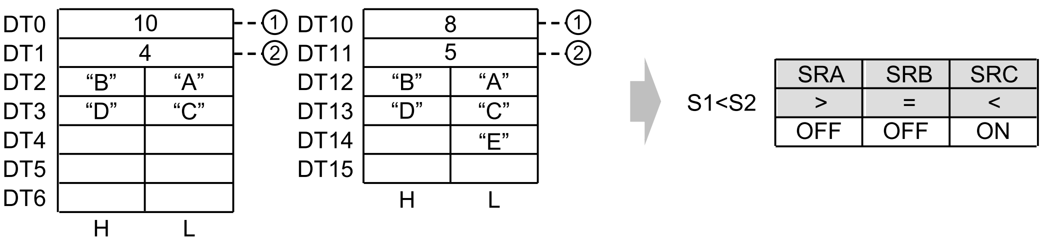
- If "NULL" is included in the comparison, it is processed as follows.

[S1]		[S2]
NULL	=	NULL
NULL	<	"ABCDE"
"B"	>	NULL

■ Processing

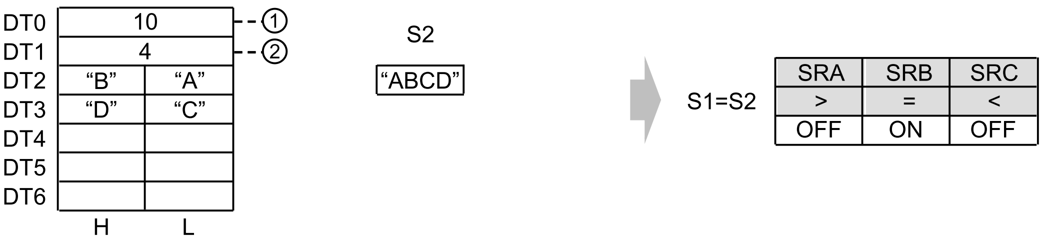
Example 1) Comparing strings "ABCD" and "ABCDE", which are stored in the data register

[S1]...DT0 [S2]...DT10 SRA...OFF SRB...OFF SRC...ON



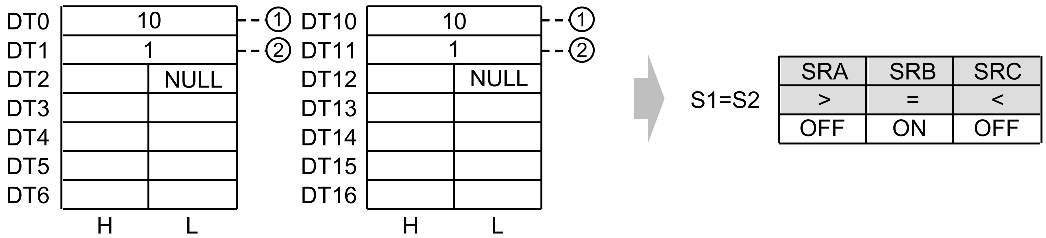
Example 2) Comparing the string "ABCD" that is stored in the data register with the character constants "ABCDE" that are specified for the operand

[S1]...DT0 [S2]..."ABCD" SRA...OFF SRB...ON SRC...OFF



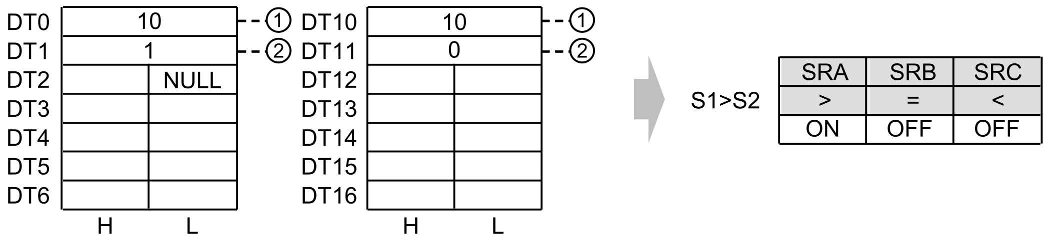
Example 3) Comparing the NULL characters that are stored in the data register

[S1]...DT0 [S2]...DT10 SRA...OFF SRB...ON SRC...OFF



Example 4) Comparing the NULL character and the empty string that are stored in the data register

[S1]...DT0 [S2]...DT10 SRA...ON SRB...OFF SRC...OFF



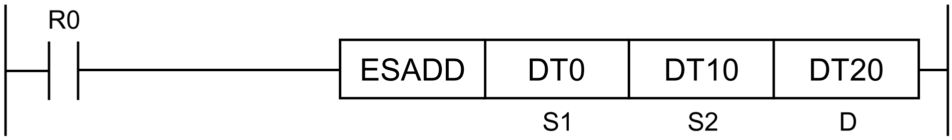
(1)	Storage area size	(2)	Character count

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the storage areas specified by S1 and S2 are out of the accessible range.
	To be set when the number of characters is larger than the storage area size for each operand.
	To be set when a data table with a storage area size of 0 or 65535 is specified.
SRA (>), SRB (=) SRC (<)	Depending on the comparison result

14.30 ESADD (String Addition: With Storage Area Size)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of String 1 to be connected (available range: 0 to 65534; for character constant: 0 to 256 characters)
S2	Starting device address of String 2 to be connected (available range: 0 to 65534; for character constant: 0 to 256 characters)
D	Starting device address to store the connected string

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	●
S2	●	●	●	●			●	●												●	●
D	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices can be modified. (Character constants cannot be specified.)

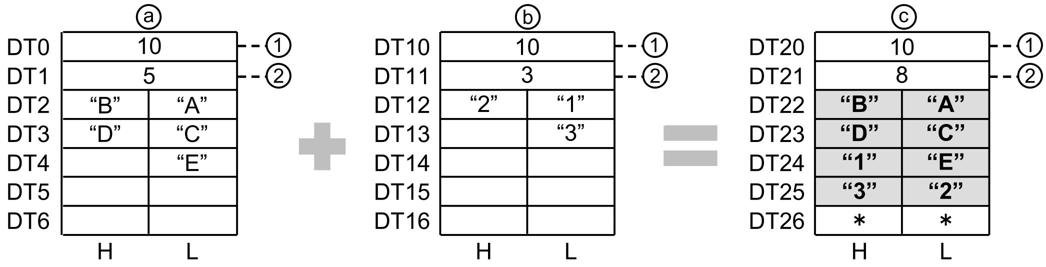
■ Outline of operation

- This instruction combines the string specified by [S1] with the string specified by [S2], and stores the combined string to the device address specified by [D].
- The maximum number of characters for the result is 65534 characters.
- For [S1], [S2], and [D], the size of the starting storage area must be set by using the user program before an instruction is executed.

■ Processing

Example 1) Connecting strings "ABCDE" and "123", which are stored in the data register

[S1]...DT0 [S2]...DT10 [D]...DT20



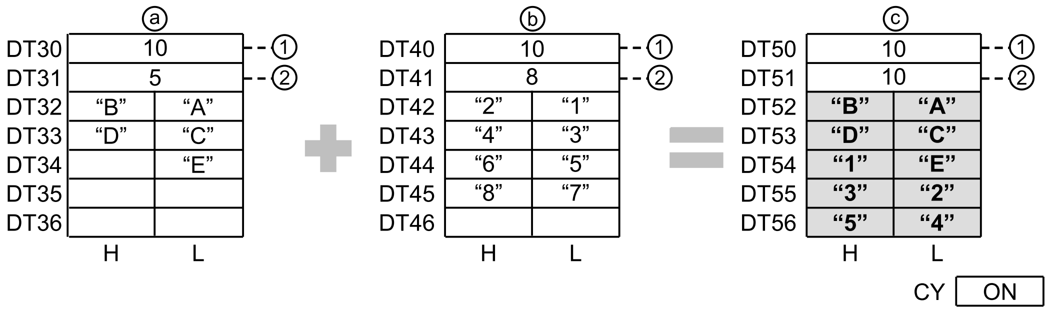
(Note 1) The content of the data (*) DT26 does not change when it is out of the range of the destination.

(1)	Storage area size	(2)	Character count

Example 2) When the size of the connected strings exceeds the storage area size for [D]

The strings are stored up to the range allowed by the storage area size, and carry flag SR9 (CY) is set.

[S1]...DT30 [S2]...DT40 [S3]...DT50 CY...ON



■ Precautions for programming

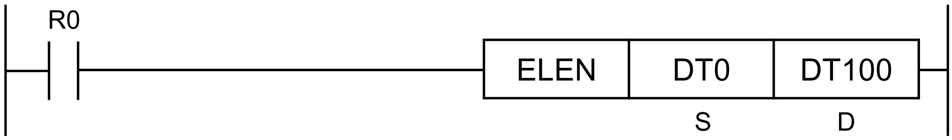
- If the connection result is larger than the storage area size of [D], the string is stored only up to the storage area size for [D].
- When specifying with a character constant, the maximum is 256 characters.
- A NULL character is processed as one character.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the string range specified by [S1] or [S2] is out of the accessible range.
	To be set when the destination range is outside the accessible range.
	To be set when the number of characters is larger than the storage area size for S1 and S2.
	To be set when a data table with a storage area size of 0 or 65535 is specified.
SR9 (CY)	To be set when the connection results is larger than the storage area size of [D].

14.31 ELEN (Obtainment of String Length: With Storage Area Size)

■ Ladder diagram



■ List of operands

Operand	Description
S	Starting device address of the string (available range: 0 to 65534)
D	Starting device address to store the string length

■ Available devices (●: Available)

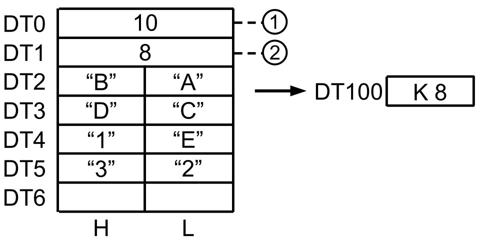
Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●													●
D	●	●	●	●			●	●	●		●										●

■ Outline of operation

- This instruction stores the number of characters stored in the beginning of the character string specified by [S] to the device address specified by [D].
- A NULL character is processed as one character.

■ Processing

Stores the number of characters in a character string that are stored after data register DT0.
[S]...DT0 [D]...DT100



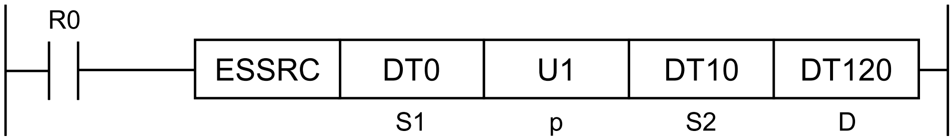
(1)	Storage area size	(2)	Character count

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the string range specified by [S] is out of the accessible range.
	To be set when the number of characters is larger than the storage area size for [S].
	To be set when a data table with a storage area size of 0 or 65535 is specified.

14.32 ESSRC (String Search: With Storage Area Size)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of the string data to be searched for (available range: 0 to 65534; for character constant: 0 to 256 characters)
p	Starting search position of the string to be searched for (available range: 1 to 65534)
S2	Starting device address of the string to be searched (available range: 1 to 65534; for character constant: 0 to 256 characters)
D	Starting device address to store the search result

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	(Note 1)
S1	●	●	●	●			●	●												●	●
p	●	●	●	●			●	●	●							●	●				●
S2	●	●	●	●			●	●												●	●
D	●	●	●	●			●	●	●		●										●

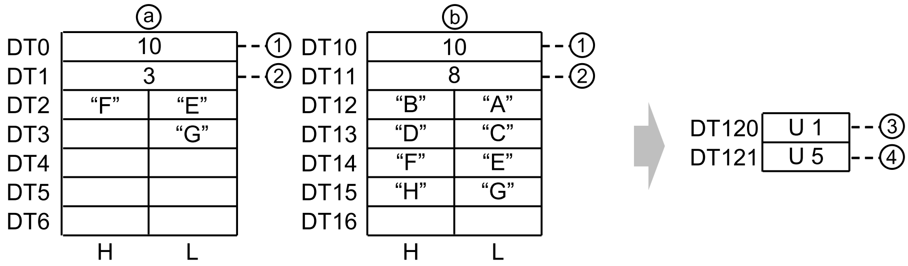
(Note 1) Only 16-bit devices, and integer constants can be modified. (Character constants cannot be specified.)

■ Outline of operation

- This instruction searches for the string data specified by [S1] in the string table starting from [S2].
- For [p], specify the relative position (by byte) in the string table where the search starts.
- For the search result, the number of same character data is stored in the device whose address is specified by [D], and the relative position (by byte) of the first match is stored in [D +1].

■ Processing

Example 1) Searching for string "EFG" stored in DT0, in the string table from DT10
[S1]...DT0 [p]...U1 [S2]...DT10 [S2]...DT120

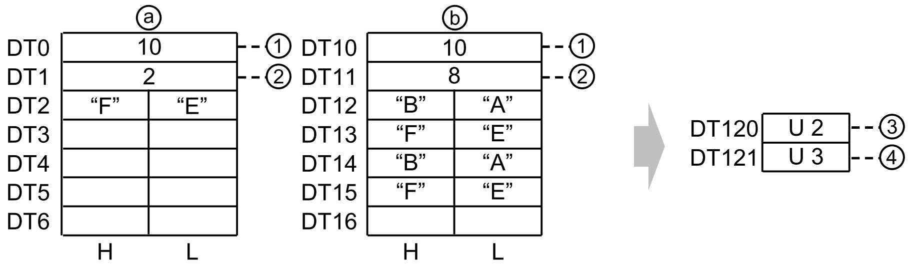


(a)	Area in which the string to be searched for is stored	(b)	String table that is searched
(1)	Storage area size	(3)	Number of matching strings
(2)	Character count	(4)	Relative position of matching string

(Note 1) Using the low byte of DT12 at the beginning of the string table as a reference, the relative position of the low byte of DT14 is calculated as a value 5, based on where it matches with string "EFG".

Example 2) When the string "EF" being searched for is found in two locations on the string table after DT10

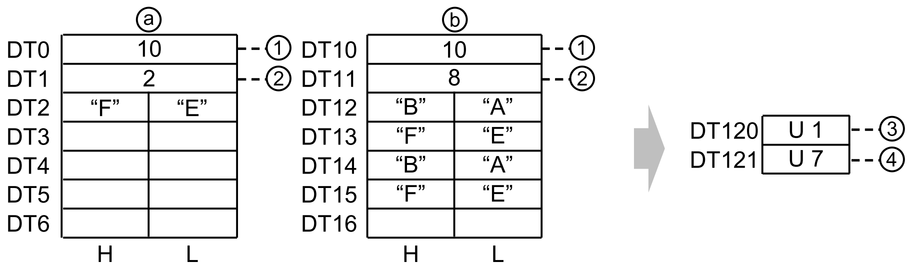
[S1]...DT0 [p]...U1 [S2]...DT10 [S2]...DT120



(Note 1) Using the low byte of DT12 at the beginning of the string table as a reference, the relative position for the low byte of DT13 is calculated as a value 3, based on where it matches with string "EF" first.

Example 3) Searching for string "EF" midway through a string data table (p = 5: 5th byte)

[S1]...DT0 [p]...U1 [S2]...DT10 [S2]...DT120

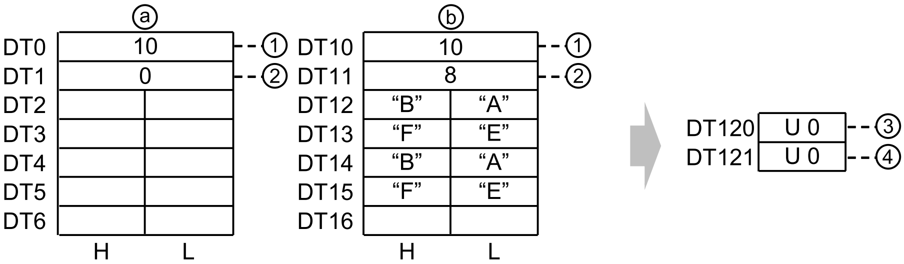


(Note 1) Using the low byte of DT12 at the beginning of the string table as a reference, the relative position for the low byte of DT15 is calculated as a value 7, based on where it matches with string "EF" first.

Example 4) When 0 is specified for the number of characters for the string data to be searched for

For both the number and the relative position, 0 is stored.

[S1]...DT0 [p]...U1 [S2]...DT10 [S2]...DT120



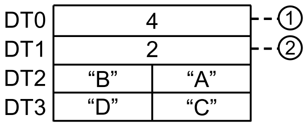
(a)	Area in which the string to be searched for is stored	(b)	String table that is searched
(1)	Storage area size	(3)	Number of matching strings
(2)	Character count	(4)	Relative position of matching string

■ Precautions for programming

- For [S1], the number of characters to search for in a string, specify the number of characters to be searched for.

  In the following figure, 1 is specified for the number of characters, and the string "A" is searched for. When 2 is specified for the number of characters, the string "AB" is searched for.

(1)	Storage area size	(2)	Character count



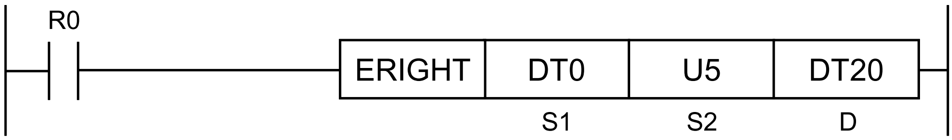
- The number of data for search results is the number of matches that occurred from the starting search position to the end of the data table.
- The relative position where the search results match is the number of bytes from the start of the data table to where the search result first matches after the starting search position.
- To search for the 2nd and subsequent relative positions, specify a value equal to the previous relative position + 1 for the operand [p] representing the starting search position, and then execute the instruction again.
- When specifying with a character constant, the maximum is 256 characters.
- If an empty string is specified for the search string data [S1], 0 is output for search results [D] and [D+1].
- The number of output bytes for the search results of the ESSRC instruction is counted starting from 1.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the number of characters [S1] is larger than [S2].
	To be set when [p] is larger than the number of characters of [S2].
	To be set when the string range specified by [S1] or [S2] is out of the accessible range.
	To be set when the number of characters is larger than the storage area size for [S1] and [S2].
	To be set when a data table with a storage area size of 0 or 65535 is specified.

14.33 ERIGHT (Takeout of the Right Side of a String: With Storage Area Size)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of the source data (available range: 0 to 65534; for character constant: 0 to 256 characters)
S2	Number of characters to be taken out (available range: 0 to 65534)
D	Starting device address to store the result that is taken out

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	●
S2	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Character constants cannot be specified.)

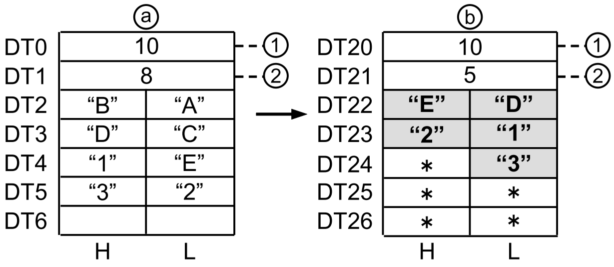
■ Outline of operation

- This instruction takes out the characters for the number specified for [S2] from the right side (end of the character data) of the string specified by [S1], and stores them as string data in the device whose address is specified by [D].
- The number of bytes of string data that are taken out is stored in [D+1].
- For both [S1] and [D], the size of the starting storage area must be set by using the user program before an instruction is executed.

■ Processing

Example 1) Taking out the last five characters "DE123" from the string "ABCDE123" to store them as string data with storage area in DT20 and later

[S1]...DT0 [S2]...U5 [S2]...DT20



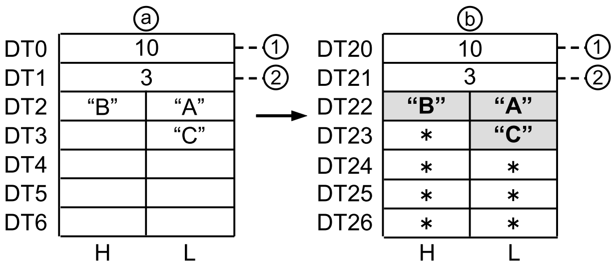
(Note 1) The data (*) that is out of the destination range, starting from the high byte of DT24 and ending with DT26, does not change.

(a)	Area in which the string to be taken out is stored	(b)	Area in which the string that is taken out is stored
(1)	Storage area size	(2)	Character count

Example 2) When the number of characters specified for [S2] is larger than the number of characters for the string that is stored in [S1]

The number of characters that is stored in [S1] is taken out and stored in [D].

[S1]...DT0 [S2]...U7 [S2]...DT20

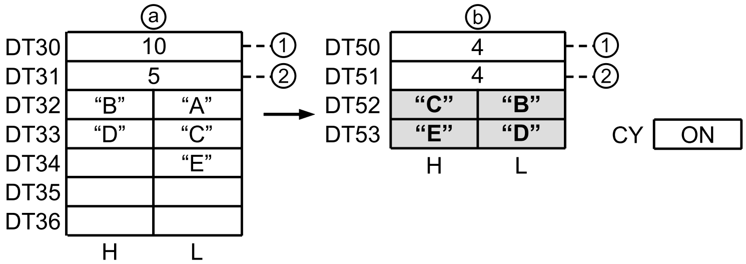


(Note 1) The data (*) that is out of the destination range, starting from the high byte of DT23 and ending with DT26, does not change.

Example 3) When the number of characters that were taken out exceeds the storage area size for [D]

Only the amount of characters that can be stored in [D] is transferred, and the carry flag SR9 (CY) is set to ON.

[S1]...DT30 [S2]...U7 [S2]...DT50



(a)	Area in which the string to be taken out is stored	(b)	Area in which the string that is taken out is stored
(1)	Storage area size	(2)	Character count

■ Precautions for programming

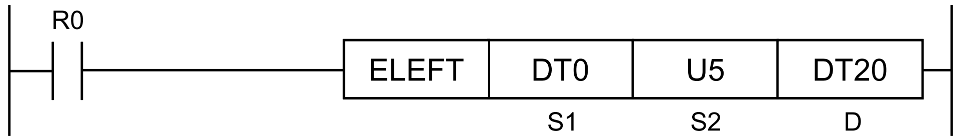
- When the number of characters of [S2] is greater than the number of characters in the string of [S1], the transferal is performed for the number of characters of [S1].
- When the number of characters in [S2] is larger than the storage area size of [D], the transferal is performed for the storage area size of [D].
- When specifying with a character constant, the maximum is 256 characters.
- When specifying an empty string for [S1], the result is always 0.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the destination range is outside the accessible range.
	To be set when [S2] (number of characters) is out of the range.
	To be set when the number of characters is larger than the storage area size for [S1].
	To be set when a data table with a storage area size of 0 or 65535 is specified.
SR9 (CY)	To be set when the number of characters taken out is larger than the storage area size of [D].

14.34 ELEFT (Takeout of the Left Side of a String: With Storage Area Size)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of the source data (available range: 0 to 65534; for character constant: 0 to 256 characters)
S2	Number of characters to be taken out (available range: 0 to 65534)
D	Starting device address to store the result that is taken out

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	(Note 1)
S1	●	●	●	●			●	●												●	●
S2	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Character constants cannot be specified.)

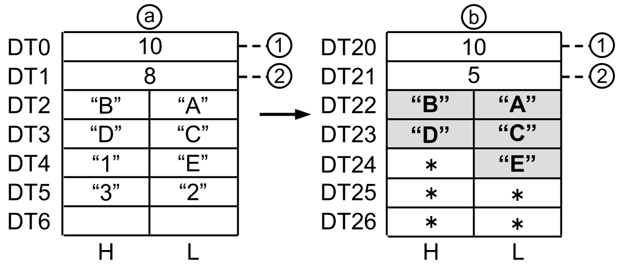
■ Outline of operation

- This instruction takes out characters as specified by [S2] from the left side (start of the character data) of the string specified by [S1], and stores them as string data with storage area in the device address specified by [D].
- The number of bytes of string data that are taken out is stored in [D+1].
- For both [S1] and [D], the size of the starting storage area must be set by using the user program before an instruction is executed.

■ Processing

Example 1) Taking out the first five characters "ABCDE" from the string "ABCDE123" to store them as string data with storage area in DT20 and later

[S1]...DT0 [S2]...U5 [S2]...DT20



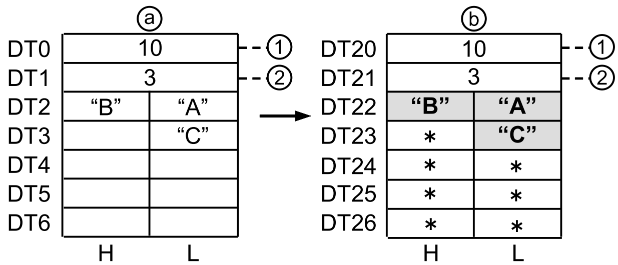
(Note 1) The data (*) that is out of the destination range, starting from the high byte of DT24 and ending with DT26, does not change.

(a)	Area in which the string to be taken out is stored	(b)	Area in which the string that is taken out is stored
(1)	Storage area size	(2)	Character count

Example 2) When the number of characters specified for [S2] is larger than the number of characters for the string that is stored in [S1]

The number of characters that is stored in [S1] is taken out and stored in [D].

[S1]...DT0 [S2]...U7 [S2]...DT20

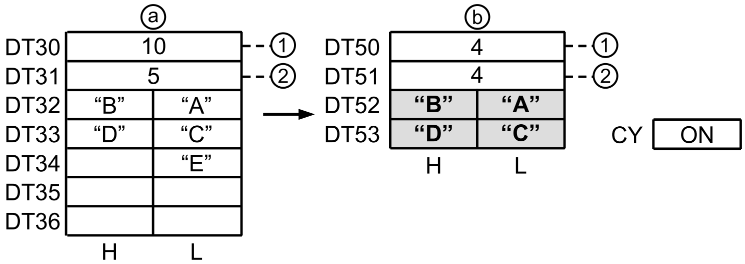


(Note 1) The data (*) that is out of the destination range, starting from the high byte of DT23 and ending with DT26, does not change.

Example 3) When the number of characters that were taken out exceeds the storage area size for [D]

Only the amount of characters that can be stored in [D] is transferred, and the carry flag SR9 (CY) is set to ON.

[S1]...DT30 [S2]...U7 [S2]...DT50



(a)	Area in which the string to be taken out is stored	(b)	Area in which the string that is taken out is stored
(1)	Storage area size	(2)	Character count

■ Precautions for programming

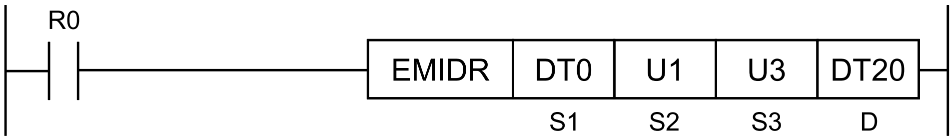
- When the number of characters of [S2] is greater than the number of characters in the string of [S1], the transferal is performed for the number of characters of [S1].
- When the number of characters in [S2] is larger than the storage area size of [D], the transferal is performed for the storage area size of [D].
- When specifying with a character constant, the maximum is 256 characters.
- When specifying an empty string for [S1], the result is always 0.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the destination range is outside the accessible range.
	To be set when [S2] (number of characters) is out of the range.
	To be set when the number of characters is larger than the storage area size for [S1].
	To be set when a data table with a storage area size of 0 or 65535 is specified.
SR9 (CY)	To be set when the number of characters taken out is larger than the storage area size of [D].

14.35 EMIDR (Data Read from a Given Position in the String: With Storage Area Size)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of the source data (available range: 0 to 65534; for character constant: 0 to 256 characters)
S2	Starting position (available range: 0 to 65533)
S3	Number of characters to be taken out (available range: 0 to 65534)
D	Starting device address to store the result that is taken out

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	●
S2	●	●	●	●			●	●	●	●	●					●	●				●
S3	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Character constants cannot be specified.)

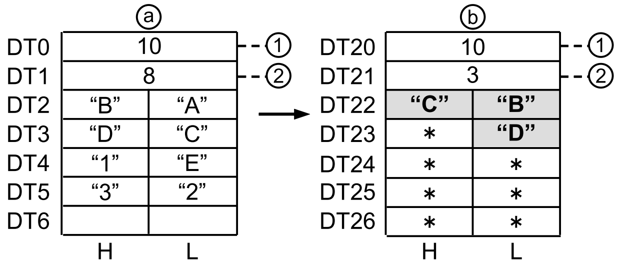
■ Outline of operation

- This instruction takes out data for the number of characters specified by [S3] from the string data specified by [S1], and stores them as string data with a storage area in the device whose address is specified by [D].
- For [S2], specify the position for taking out string data.
- For both [S1] and [D], the size of the starting storage area must be set by using the user program before an instruction is executed.

■ Processing

Example 1) Taking out the three characters "BCD" from the 1st byte (2nd character) of the string "ABCDE123" to store them as string data with storage area in DT20 and later

[S1]...DT0 [S2]...U1 [S3]...U3 [D]...DT20



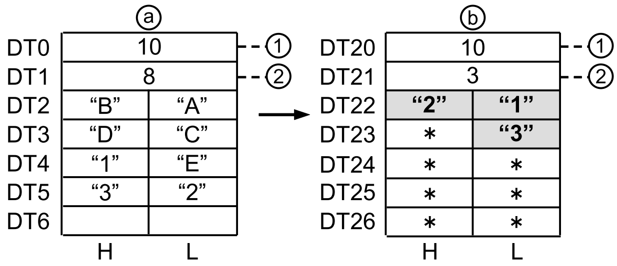
(Note 1) The data (*) that is out of the destination range, starting from the high byte of DT23 and ending with DT26, does not change.

(a)	Area in which the string to be taken out is stored	(b)	Area in which the string that is taken out is stored
(1)	Storage area size	(2)	Character count

Example 2) When the number of characters specified for [S3] is larger than the number of characters for the string that is stored in [S1] after [S2]

Take out the three characters "123" which come after the 5th character of [S2], and store them in [D].

[S1]...DT0 [S2]...U5 [S3]...U5 [D]...DT20

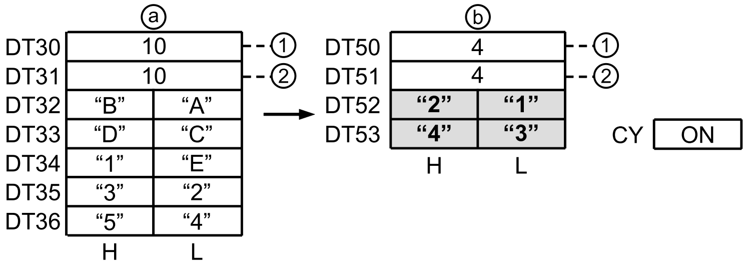


(Note 1) The data (*) that is out of the destination range, starting from the high byte of DT23 and ending with DT26, does not change.

Example 3) When the number of characters that were taken out exceeds the storage area size for [D]

Only the amount of characters that can be stored in [D] is transferred, and the carry flag SR9 (CY) is set to ON.

[S1]...DT0 [S2]...U5 [S3]...U5 [D]...DT20



(a)	Area in which the string to be taken out is stored	(b)	Area in which the string that is taken out is stored
(1)	Storage area size	(2)	Character count

■ Precautions for programming

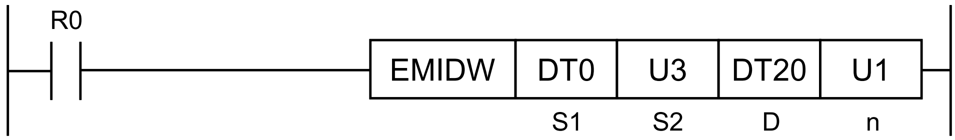
- When the number of characters of [S3] is larger than the number of characters of the [S1] string starting from the [S2] position, the transferal is performed for the number of characters of [S1].
- The [S2] position should be specified by a number counted from low order in sequence, putting the least significant byte as K0 (Byte 0): 0, 1, 2 and so on.
- When the number of characters in the operation result is larger than the storage area size of [D], the transferal is performed for the storage area size of [D].
- When specifying with a character constant, the maximum is 256 characters.
- When specifying an empty string for [S1], the result is always 0.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the destination range is outside the accessible range.
	To be set when the number of characters for [S1] is smaller than or equal to [S2].
	To be set when the number of characters is larger than the storage area size for [S1].
	To be set when a data table with a storage area size of 0 or 65535 is specified.
SR9 (CY)	To be set when the number of characters taken out is larger than the storage area size of [D].

14.36 EMIDW (Rewrite from a Given Position in the String: With Storage Area Size)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting device address of the source data (available range: 0 to 65534; for character constant: 0 to 256 characters)
S2	Number of characters (available range: 0 to 65534)
D	Destination starting device address
n	Starting position of the destination string (available range: 0 to 65533)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	●
S2	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●													●
n	●	●	●	●			●	●	●	●	●					●	●				●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Character constants cannot be specified.)

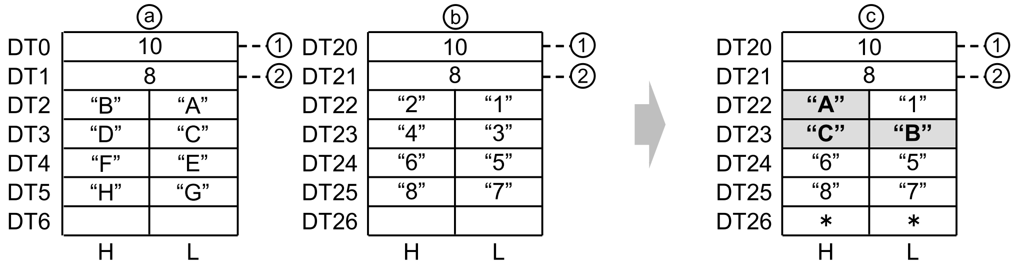
■ Outline of operation

- This instruction takes out data for the number of characters specified by [S2] from the character string specified by [S1], and transfers it to the position [n] of the character string specified by [D].

■ Processing

Example 1) Taking out the three characters "ABC" from the DT0 string to transfer them to the 1st byte (2nd character) of the DT20 string

[S1]...DT0 [S2]...U3 [D]...DT20 [n]...U1



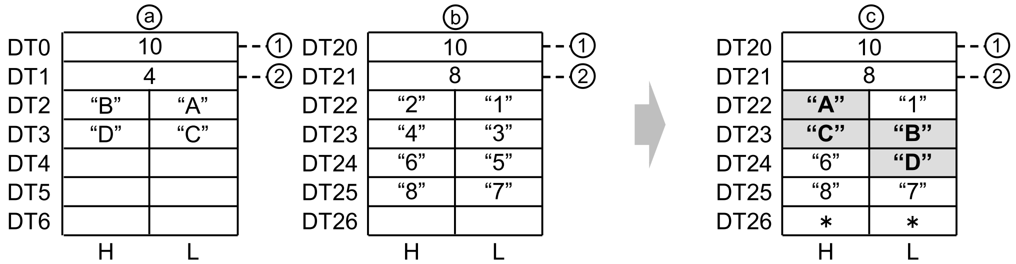
(Note 1) The data (*) for DT26 that is out of the destination range does not change.

(a)	Area in which the string to be taken out is stored	(b)	Destination area (before transferring)	(c)	Destination area (after transferring)
(1)	Storage area size	(2)	Character count	-	

Example 2) Taking out the four characters "ABCD" from the DT0 string to transfer them to the 1st byte (2nd character) of the DT20 string

This indicates that the number of characters specified for [S2] is larger than the number of characters for the string that is stored in [S1].

[S1]...DT0 [S2]...U5 [D]...DT20 [n]...U1

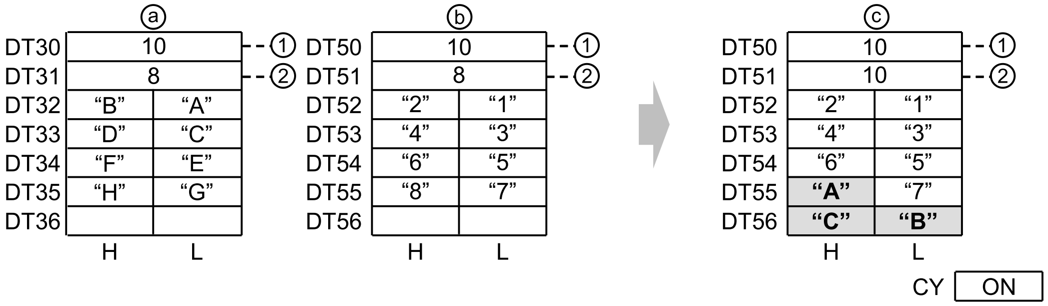


(Note 1) The data (*) for DT26 that is out of the destination range does not change.

Example 3) When the number of characters that were taken out exceeds the storage area size for [D]

Only the amount of characters that can be stored in [D] starting from the 7th is transferred, and carry flag SR9 (CY) is set to ON.

[S1]...DT30 [S2]...U5 [D]...DT50 [n]...U7



(a)	Area in which the string to be taken out is stored	(b)	Destination area (before transferring)	(c)	Destination area (after transferring)
(1)	Storage area size	(2)	Character count	-	

■ Precautions for programming

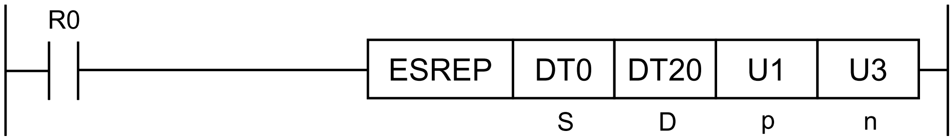
- When the number of characters of [S2] is larger than the number of characters in the string of [S1], the transferal is performed for the number of characters in the string of [S1].
- The [n] position should be specified counting by a number counted from low order in sequence, putting the least significant byte as K0 (Byte 0): 0, 1, 2 and so on.
- When the number of characters in the operation result is larger than the storage area size of [D], the transferal is performed for the storage area size of [D].
- When specifying with a character constant, the maximum is 256 characters.
- When specifying an empty string for [S1], the result is always 0.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [S2] (number of characters) is out of the range.
	To be set when the number of characters of [D] is smaller than [n].
	To be set when the number of characters is larger than the storage area size for [S1] and [D].
	To be set when a data table with a storage area size of 0 or 65535 is specified.
SR9 (CY)	To be set when the number of characters taken out is larger than the storage area size of [D].

14.37 ESREP (Replacement of a String: With Storage Area Size)

■ Ladder diagram



■ List of operands

Operand	Description
S	Starting device address of the source string (available range: 0 to 65534; for character constant: 0 to 256 characters)
D	Starting device address of the destination string (available range: 1 to 65534)
p	Replacement start position of the destination string (available range: 0 to 65533)
n	Number of characters to be replaced (available range: 0 to 65534)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●												●	●
D	●	●	●	●			●	●													●
p	●	●	●	●			●	●	●	●	●					●	●				●
n	●	●	●	●			●	●	●	●	●					●	●				●

(Note 1) Only 16-bit devices, and integer constants can be modified. (Character constants cannot be specified.)

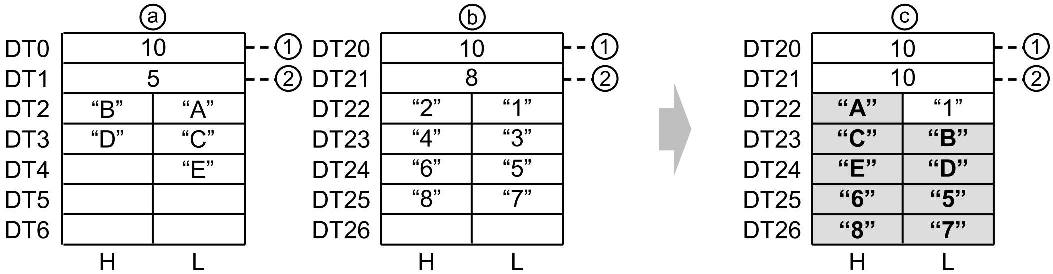
■ Outline of operation

- This instruction replaces the string specified by [D] with the string specified by [S].
- For [p], specify the relative position in [D] where the string is replaced, and for [n] specify the number of characters to be replaced.

■ Processing

Example 1) Taking out the five characters "ABCDE" from the DT0 string to transfer them to the 1st byte (2nd character) of the DT20 string

[S]...DT0 [D]...DT20 [p]...U1 [n]...U3



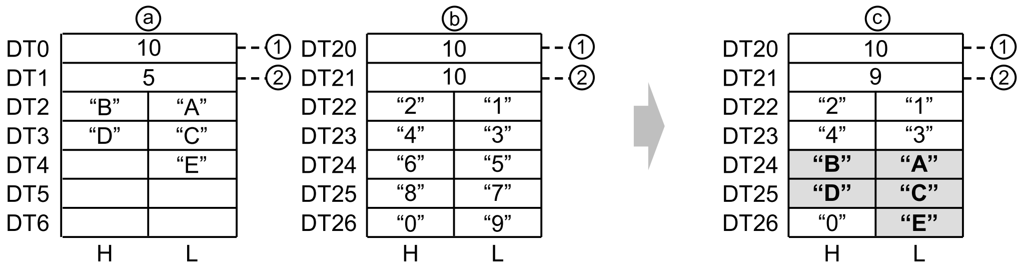
(Note 1) The data "234", from the 2nd to the 4th character in the destination area (before transferring), is deleted, and the data "5678", from the 5th to the 8th character, is shifted.

(a)	Area in which the string to be taken out is stored	(b)	Destination area (before transferring)	(c)	Destination area (after transferring)
(1)	Storage area size	(2)	Character count	-	

Example 2) Taking out the five characters "ABCDE" from the DT0 string to transfer them to the 4th byte (5th character) of the DT20 string

This indicates that the number of characters specified for [n] is larger than the number of characters for the string that is stored in [S1].

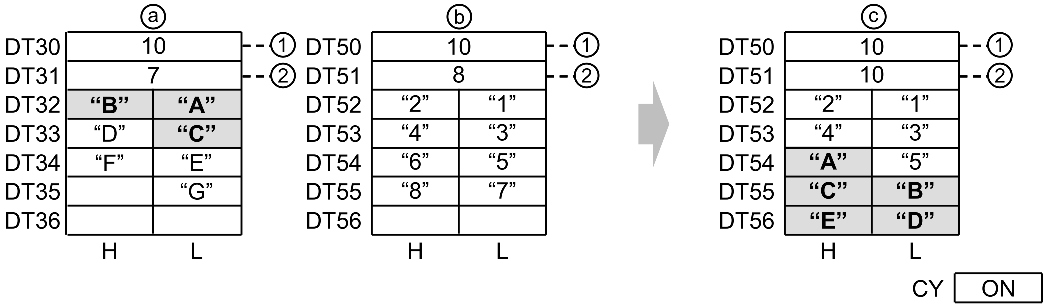
[S]...DT0 [D]...DT20 [p]...U4 [n]...U8



Example 3) When the number of characters that were taken out exceeds the storage area size for [D]

Only the amount of characters that can be stored in [D] starting from the 5th byte (6th character) is transferred, and carry flag SR9 (CY) is set to ON.

[S]...DT30 [D]...DT50 [p]...U5 [n]...U3



(a)	Area in which the string to be taken out is stored	(b)	Destination area (before transferring)	(c)	Destination area (after transferring)
(1)	Storage area size	(2)	Character count	-	

■ Precautions for programming

- If the number of characters [n] is larger than the number of characters in the string [S1] as from the position specified by [p], then the replacement is performed for the number of characters of the string [S] as from the position specified by [p].
- The [p] position should be specified by a number counted from low order in sequence, putting the least significant byte as K0 (Byte 0): 0, 1, 2 and so on.
- When specifying with a character constant, the maximum is 256 characters.
- When specifying an empty string for [S1], the result is always 0.

■ Flag operations

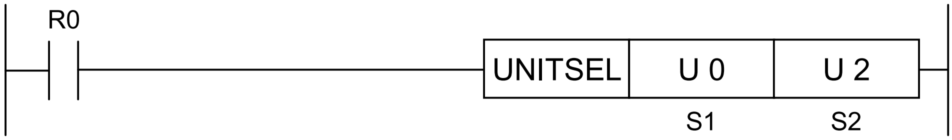
Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when [S] (number of characters) is out of the range.
	To be set when the number of characters of [D] is smaller than [n].
	To be set when the position of [p] is larger than the number of characters of [D].
	To be set when the number of characters is larger than the storage area size for [S] and [D].
	To be set when a data table with a storage area size of 0 or 65535 is specified.
	To be set when [n] (number of characters to be replaced) is out of the range.
SR9 (CY)	To be set when the number of characters taken out is larger than the storage area size of [D].

15 High-level Instructions (Communication)

Applicable Models: All Models

15.1 UNITSEL (Specification of a Communication Unit Slot Port)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Slot number of the unit
S2	COM port number or user connection number

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●								●					●
S2	●	●	●	●			●	●								●					●

■ Outline of operation

- This instruction should be described immediately before the following communication instructions, to specify the targets of execution.

  GPTRNS, pGPSEND, GPSEND, GPRECV, SEND, RECV, PMSET, pPMSET, PMGET, RDET, ETSTAT, IPv4SET, PINGREQ, CONSET, OPEN, CLOSE, NTPcREQ, NTPcSV, FTPcSV, FTPcSET, FTPcLOG, FTPcREQ, FTPcCTL, HTTPcSV, HTTPcSET, HTTPcREQ, HTTPcCTL, SMTPcBDY, SMTPcBRD, SMTPcSV, SMTPcADD, SMTPcSET, SMTPcREQ, SMTPcCTL, EIPNDST, EIPSTART, EIPSTOP, EIP_IN, EIP_OT
- In the case of a CPU with built-in SCU, specify a slot number (U0) for [S1] and a COM port number for [S2].
- In the case of a CPU with built-in ET-LAN, specify a slot number (U100) for [S1] and a connection number for [S2].
- In the case of a serial communication unit, specify a slot number (U1 to 16) for [S1] and a COM port number for [S2].
- This instruction obtains the type of slot specified by [S1], and checks that the communication port number specified by [S2] falls within the available range. If the number is out of the range, an error will occur.
- In the case of SCU, check that the specified communication port (COM port number) is equipped with a communication cassette. If the specified COM port is not equipped with a communication cassette, an error will occur.

- If no error occurs, the values of [S1] and [S2] should be set to system data register (SD40, SD41) of the CPU unit.

■ Specification of [S1] and [S2]

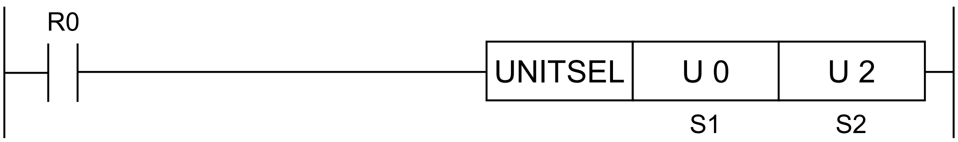
- Specify a slot number of the unit in [S1]. The set value for [S1] should be stored in system data register SD40.
- Specify a communication port in [S2]. (In the case of SCU: COM port number. In the case of CPU with built-in ET-LAN: connection number.) The set value for [S2] should be stored in system data register SD41.

Unit type	[S1] Slot No.	[S2] COM port No. User connection No.
CPU with built-in SCU	U0	U0 to U2
CPU with built-in ET-LAN	U100	U1 to U16 U17 to U216 (Note 1)
Serial Communication Unit (SCU)	U1 to U64	U1 to U4

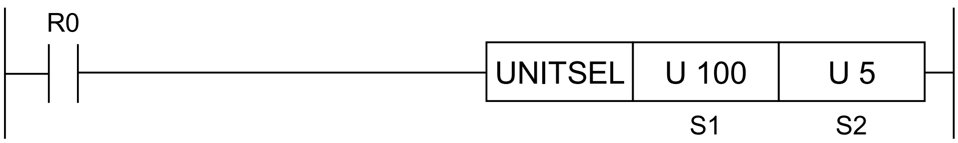
(Note 1) CPU unit Ver. 3 and later is required to use connection numbers U17 to U216 for a CPU unit with built-in ET-LAN. In addition, the number of user connections must also be changed in tool software "FPWIN GR7 Configuration" **Built-in ET-LAN**>**Add-on**.

■ Program example

Example 1) Specify COM2 for SCU with built-in CPU in Slot 0



Example 2) Specify User Connection 5 for ET-LAN with built-in CPU in Slot 100

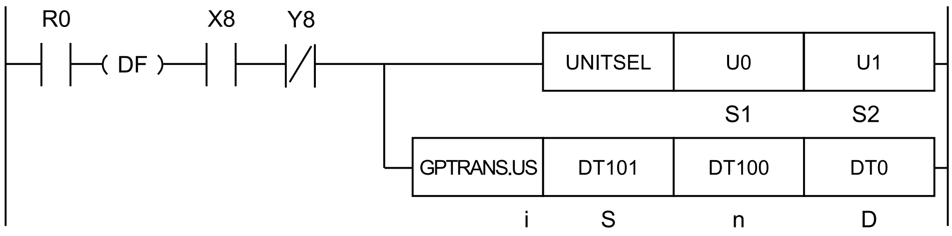


■ Flag operations

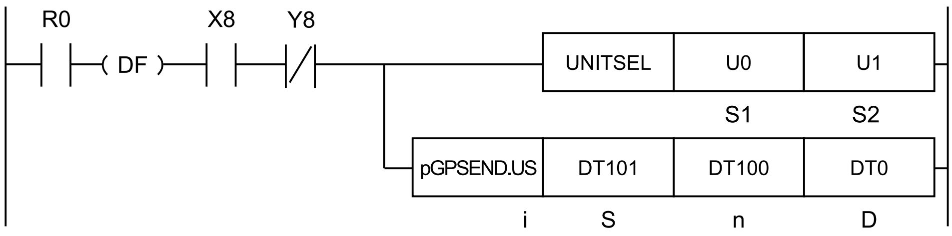
Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	The COM port specified by [S2] does not exist (no cassette, not a communication cassette).
	The connection specified by [S2] does not exist (out of the connection number range).

15.2 GPTRNS / pGPSEND /GPSEND (General-Purpose Communication Send Instruction)

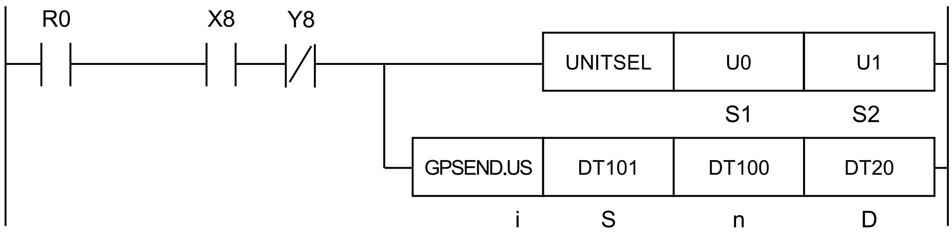
■ Ladder diagram (GPTRNS)



■ Ladder diagram (pGPSEND)



■ Ladder diagram (GPSEND)



(Note 1) The above figure shows the case that S1=U0 (CPU unit with built-in SCU) and S2=U1 (port number 1) are specified by the UNITSEL instruction.

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

(Note 1) When a negative value is specified for operand [n], it is necessary to specify an SS operation unit.

■ List of operands

Operand	Description
S	Starting number for the device for storing the sent data
n	Number of bytes of the sent data, or starting number of the device where the amount of sent data is stored
D	Starting number of the device that stores the processing result (1 word)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●													●
n(Note 1)	●	●	●	●			●	●							●	●	●				●
D (Note 1)	●	●	●	●			●	●													●

(Note 1) Always 16-bit data/device, regardless of the specification of operation units [i].

■ Outline of operation

- Data are sent from the communication port to external devices.
- Data of [n] bytes are sent from the unit / communication port set by the UNITSEL instruction, starting with the starting address (word address) of the sent data area specified by [S].
- Data to be sent are set by the user program, in the area starting with [S].
- The processing result is stored in the area specified by [D].

■ Processing

- The slot numbers and communication port numbers specified with UNITSEL instruction are obtained from the system data register (SD).
- This instruction confirms that the general-purpose communication clear to send flag of a specified communication port is ON and the general-purpose communication sending flag is OFF.
- When sending is enabled, sent data is transferred to the send buffer of a communication port and a request to send is executed.

Info.

- The case of SCU shows the case that it is used in the following combination.
  - COM.0 port equipped in the CPU unit
  - Communication cassettes attached to the CPU unit (COM.1 to COM.2 ports)
  - Communication cassettes attached to the serial communication unit (COM.1 to COM.4 ports)
- The case of ET-LAN shows the case that it is used in the following combination.
  - LAN port equipped in the CPU unit (Applicable models: CPU unit CPS4RE* and CPS3RE* only)
- As the communication cassette (Ethernet type) has an Ethernet-serial conversion function, the internal interface operates with similar programs as the case of SCU. The setting method and programming method are different from those for the CPU with built-in ET-LAN.

■ Comparison of GPTRNS / pGPSEND / GPSEND instructions

<table>
<tr><th>Instruction</th><th colspan="2">Characteristics</th></tr>
<tr><td rowspan="3">GPTRNS
pGPSEND
(Note 1)</td><td colspan="2">The data send confirmation process is performed on the completion of scan. (Note 2)</td></tr>
<tr><td>Advantage</td><td>Data is sent only once by turning on execution conditions at the time of data send.</td></tr>
<tr><td>Disadvantage</td><td>Up to 16 send operations can be performed simultaneously to different COM ports and connections. (The total of simultaneous usage of SEND, RECV, GPTRNS, pGPSEND, and pPMSET instructions)</td></tr>
<tr><td rowspan="3">GPSEND</td><td colspan="2">The data send confirmation process is performed in the operation processing for the GPSEND instruction. (Note 2)</td></tr>
<tr><td>Advantage</td><td>Data can be sent to different COM ports and connections simultaneously without limit.</td></tr>
<tr><td>Disadvantage</td><td>It is necessary to turn ON the execution condition of the GPSEND instruction until the end of data sending, and turn OFF the execution condition at a scan in which the end of data transmission is confirmed.</td></tr>
</table>

(Note 1) As the result of executing an operation, the conditions under which an error is set for operand [D] will vary.

Condition in which errors occur	GPTRNS	pGPSEND	GPSEND
General-purpose communication clear to send flag is OFF.	(Note 2)	●	●
16 or more SEND, RECV, GPTRNS, pGPSEND, and pPMSET instructions are used simultaneously.	●	●	-
Communication error	●	●	●

(Note 2) Even when the same port is specified and the GPTRNS instruction is executed during the execution of the general-purpose communication transmission, an error does not occur and the result is not updated.

(Note 3) The data sending confirmation process indicates if the sending flag is cleared, if the send results flag is set, and if the processing results are stored in [D].

■ Creation of send data table [S]

- Data to be sent is stored from the low byte of a given area specified by [S].

- The figure below shows the case where the string "ABCDE" is converted with the SSET instruction.

DT100	U5		①
DT101	H42(B)	H41(A)	②
DT102	H44(D)	H43(C)	
DT103	-----	H45(E)	
DT104	-----	-----	

(1)	Once the SSET instruction is executed, the number of characters is set in the starting word. Then, the characters that are converted are stored in the following area.
(2)	DT101 is set for operand [S] of the GPTRNS / pGPSEND / GPSEND instruction, and when the instruction is executed the data is sent in ascending order from the low byte.

■ Setting the number of bytes in sent data [n]

Unit type	Set value	Description
SCU (Note 1)(Note 2)	1 to 4096	When the value is positive, an end code is automatically added according to the "Terminator setting" of COM settings.
	-1 to -4096	When the value is negative, an end code is not automatically added regardless of the "Terminator setting" of COM settings.
ET-LAN	1 to 16384	• When "Add no special header" is on (Default connection setting) (Note 3) • Send data and the end code are not distinguished. It is not automatically added.
	1 to 16372	• When specifying "Append a special header" (Note 3) • Send data and the end code are not distinguished. It is not automatically added.

(Note 1) For SCU, up to 4096 bytes can be sent, including a start code and an end code. The start code and end code are set from the configuration menu of the tool software or with the PMSET instruction.

When "Start code STX" is set to "Enabled", the maximum amount of sent data that can be specified is decremented by one.

When "Terminator setting" is set to "ETX" or "CR", the maximum amount of sent data is decremented by one.

When "Terminator setting" is set to "CR+LF", the maximum amount of sent data is decremented by two.

When "Terminator setting" is set to "Time", the maximum amount of sent data is not decremented.

(Note 2) For specifying a negative value for [n] (signed integer K), specify SS for the operation units.

(Note 3) A "special header" is added when communicating with a conventional FP2 ET-LAN unit, and when communication is performed with MEWTOCOL. Normally, select "Add no special header" in the user connection setting.

■ Operand [D] settings

- Specify the device area of the master unit storing the processing result (1 word).
- The following values are stored depending on the state.

Status	Value that is set
When starting the transmission request	H0
When transmission is completed	Number of transmitted bytes
When an error occurs	HFFFF

■ Precautions for programming

- To perform communication, setup is required in the configuration menu of the tool software.
- For a CPU unit with a built-in SCU, select **General-purpose communication** in **FP7 Configuration**>**Built-in SCU**>**Communication mode**. For a Serial Communication Unit, also select "General-purpose communication" in "Communication mode".
- For a CPU unit with a built-in ET-LAN, use the FPWIN GR7 tool software to select "General-purpose communication" in **FP7 Configuration**>**Built-in ET-LAN**>**User connection information setting**>**Operation mode setting**.
- Before executing the GPTRNS instruction, pGPSEND instruction, or GPSEND instruction, describe the UNITSEL instruction and specify the target unit and communication port or connection.
- The GPTRNS, pGPSEND, and GPSEND instructions should be executed after confirming that the general-purpose communication clear to send flag for the target COM port and connection is ON and that the general-purpose communication sending flag is OFF.
- For the GPSEND instruction, be sure to keep the execution condition ON until the completion of the transmission that sets the general-purpose communication sending flag to OFF.
- The general-purpose communication sent flag in the WX area is used for confirming the completion of the transmission using the general-purpose communication.
- When data is sent to a communication port that is undergoing transmission, it results in no operation. No error occurs.
- Sending zero-byte data results in an error.
- For a GPTRNS instruction and a pGPSEND instruction, up to 16 instructions can be executed simultaneously for different COM ports and connections. (The total of simultaneous usage of SEND, RECV, GPTRNS, pGPSEND, and pPMSET instructions)
- The GPTRNS, pGPSEND, and GPSEND instructions are not available in interrupt programs.

■ Precautions during programming (in the case of SCU)

- If a positive number is specified for [n], the start code and end code that are specified in the configuration menu are automatically added to the data to be sent. Do not include a start code or an end code in the sent data.
- If an end code will not be added, specify a negative number for the amount of sent data [n]. In addition, select "SS" as the operation units.
- The maximum volume of data that can be sent with GPTRNS, pGPSEND, and GPSEND instructions is 4,096 bytes, including a start code and an end code.

■ Precautions during programming (in the case of CPU with built-in ET-LAN)

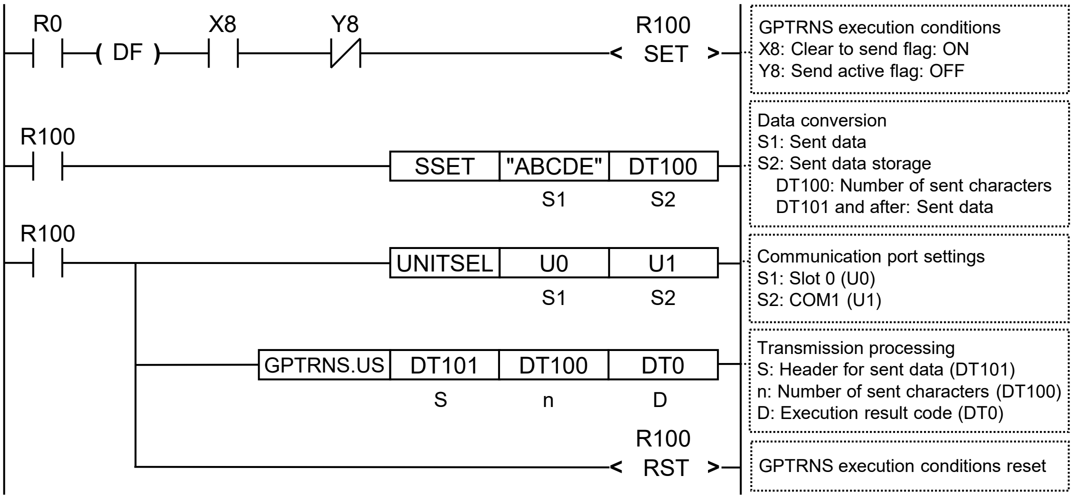
- General-purpose communication can use user connections 1 to 16. It cannot be used for the system connection and the expansion user connections 17 to 216.
- User connections should be in the "connected" state. We recommend using the FPWIN GR7 tool software to set "Open automatically" in **Built-in ET-LAN**>**User connection information setting**>**Open type**. The connections can also be connected with the OPEN instruction.
- No header or terminator is added to data to be sent. Store the start code and end code as part of the send data if they need to be sent to match the external device protocol.

- The maximum volume of data that can be sent in a single instance with GPTRNS, pGPSEND, and GPSEND instructions is 16,384 bytes.

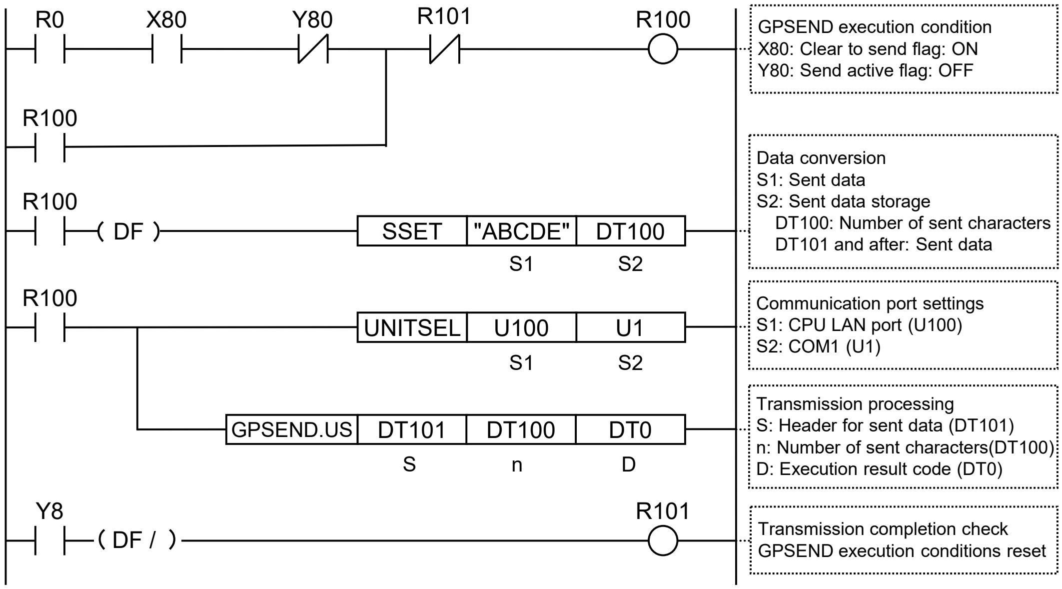
■ Sample program (in the case of SCU)

- This program checks that the general-purpose communication mode is on (X8:ON) and that general-purpose sending is not in progress in the same port (Y8:OFF), and then starts up the sending program.
- Using the SSET instruction, convert any given message into an ASCII string. Set the number of sent characters to the data register DT100, and the sent message to the data register DT101.
- Using the UNITSEL instruction, specify the slot number (U0) and the COM. port number (U1).
- In the GPTRNS / pGPSEND / GPSEND instruction, specify and execute the start of the table that stores the message to be sent (DT101) and the number of characters in the data (DT100).

GPTRNS instruction



GPSEND instruction

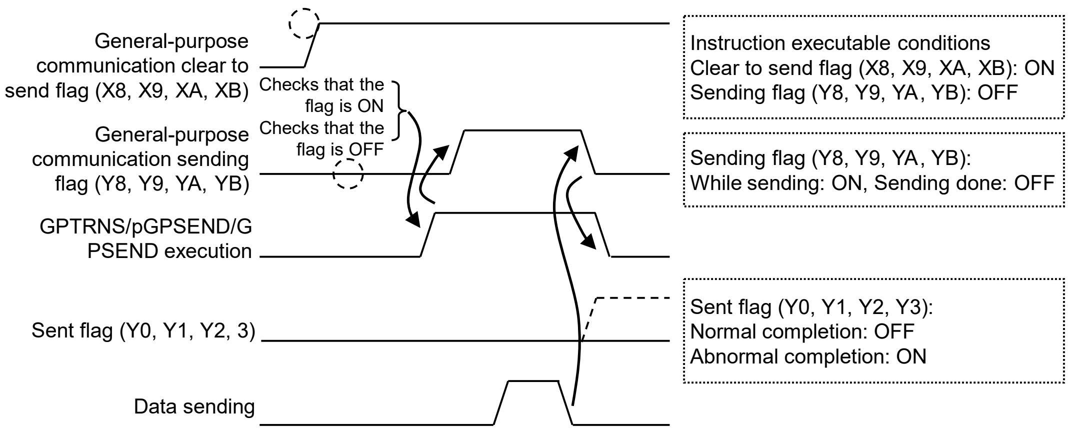


Info.

- For the GPSEND instruction, it is necessary to turn ON the execution condition until the end of data sending, and turn OFF the execution condition at a scan in which the end of data sending is confirmed.

■ Time chart (in the case of SCU)

- Data are sent in ascending order from low bytes of [S+1] in the table specified by the GPTRNS / pGPSEND / GPSEND instruction.
- During sending, the general-purpose communication sending flags (Y8, Y9, YA, YB) are turned ON. They turn OFF when sending is completed.
- The sending result (0: normal completion; 1: abnormal completion) is stored in the general-purpose communication sent flags (Y0, Y1, Y2, Y3).



I/O allocation (in the case of CPU with built-in SCU)

COM Port No.			Name	Description
1	2	0		
X8	X9	XA	General-purpose communication clear to send flag	Turns ON when the unit is set to the general-purpose communication mode.
Y8	Y9	YA	General-purpose communication sending flag	Turns ON when sending with general-purpose communication mode. Turns OFF when the sending process is completed.
Y0	Y1	Y2	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

(Note 2) In the case of the GPTRNS and pGPSEND instructions, the sending flag and send results flag are updated after the scan is complete.

(Note 3) In the case of the GPSEND instruction, the sending flag and send results flag are updated after sending is completed when the next GPSEND instruction is executed. If sending time is shorter than scan time, the general-purpose communication sending flags (Y8, Y9, YA, YB) are turned OFF when the GPSEND instruction is executed in the subsequent scan following completion of data sending. The flags remain ON for at least one scan time.

I/O allocation (in the case of Serial Communication Unit)

COM Port No.				Name	Description
1	2	3	4		
X8	X9	XA	XB	General-purpose communication clear to send flag	Turns ON when the unit is set to the general-purpose communication mode.
Y8	Y9	YA	YB	General-purpose communication sending flag	Turns ON when sending with general-purpose communication mode. Turns OFF when the sending process is completed.
Y0	Y1	Y2	Y3	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

(Note 2) In the case of the GPTRNS and pGPSEND instructions, the sending flag and send results flag are updated after the scan is complete.

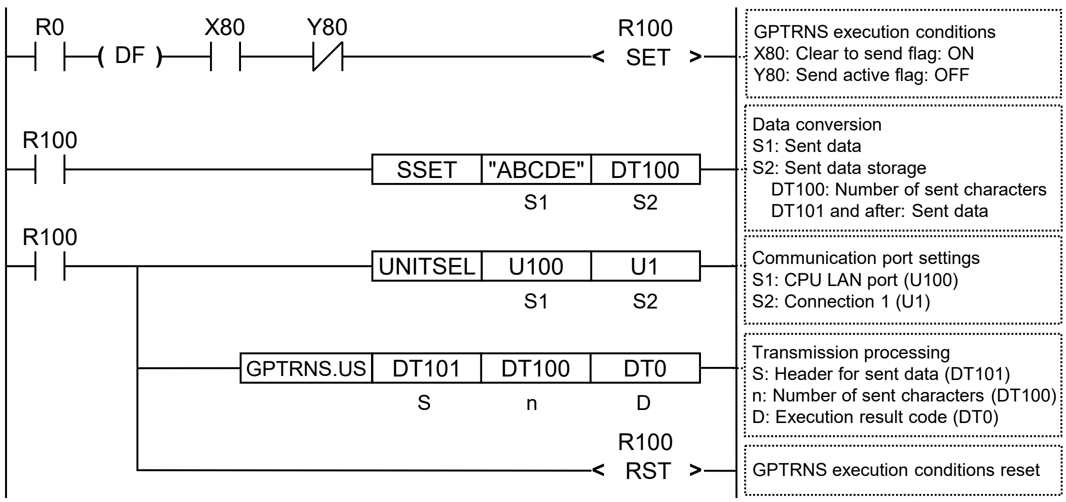
(Note 3) In the case of the GPSEND instruction, the sending flag and send results flag are updated after sending is completed when the next GPSEND instruction is executed. If sending time is shorter than scan time, the general-purpose communication sending flags (Y8, Y9, YA, YB) are turned OFF when the GPSEND instruction is executed in the subsequent scan following completion of data sending. The flags remain ON for at least one scan time.

Sample program (in the case of CPU with built-in ET-LAN)

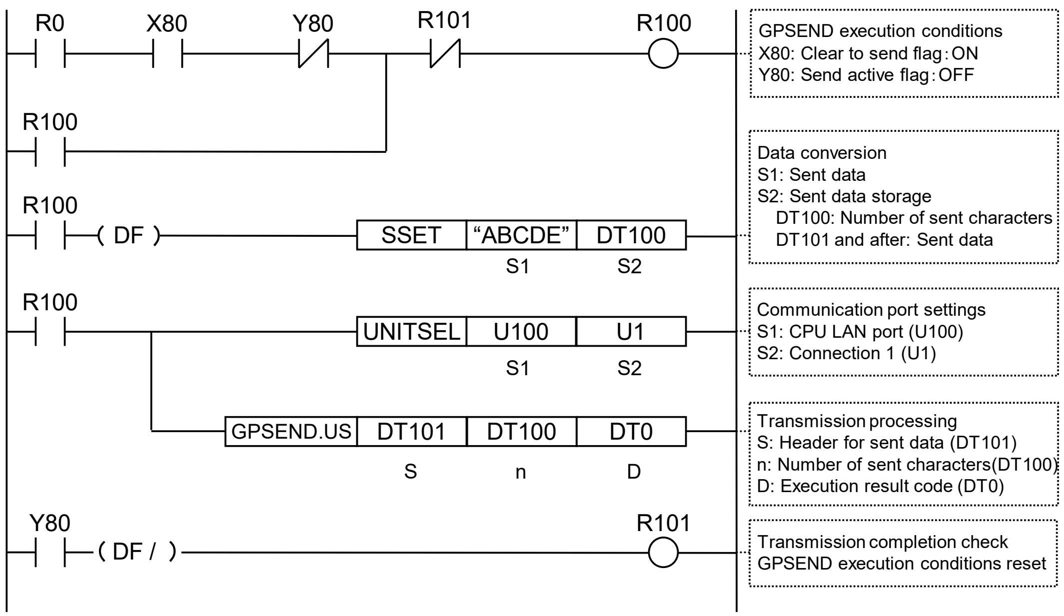
- This program checks that Connection 1 is established in the general-purpose communication mode (X80:ON) and that general-purpose sending is not in progress in the same port (Y80:OFF), and then starts up the sending program.

- Using the SSET instruction, convert any given message into an ASCII string. Set the number of sent characters to the data register DT100, and the sent message to the data register DT101.
- The UNITSEL instruction is used to specify a slot number (LAN port: U100) and the connection number (U1).
- In the GPTRNS / pGPSEND / GPSEND instruction, specify and execute the start of the table that stores the message to be sent (DT101) and the number of characters in the data (DT100).

GPTRNS instruction



GPSEND instruction

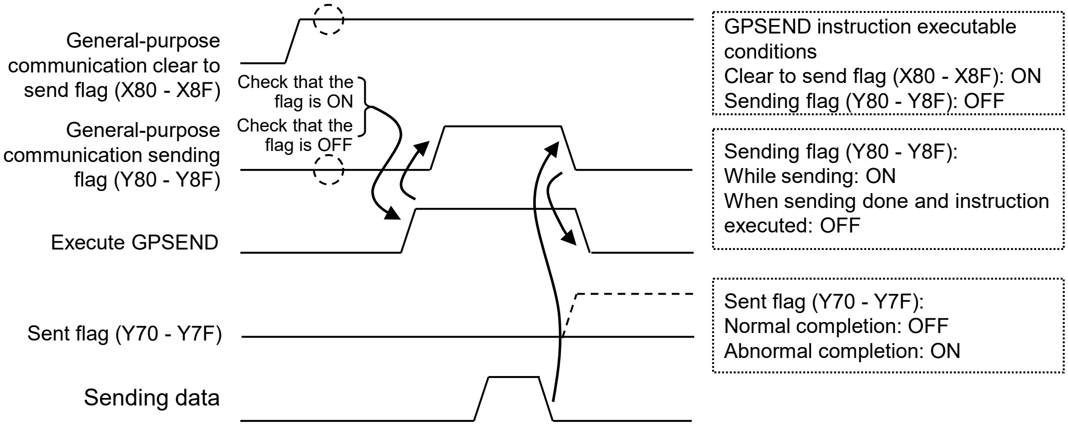


Info.

- For the GPSEND instruction, it is necessary to turn ON the execution condition until the end of data sending, and turn OFF the execution condition at a scan in which the end of data sending is confirmed.

■ Time chart (in the case of CPU with built-in ET-LAN)

- Data are sent in ascending order from low bytes of [S+1] in the table specified by the GPTRNS / pGPSEND / GPSEND instruction.
- During sending, the general-purpose communication sending flags that correspond to the connection (Y80 to Y8F) are turned ON. They turn OFF when sending is completed.
- The sending result (0: normal completion; 1: abnormal completion) is stored in the general-purpose communication sent flags (Y70 to Y7F).



■ I/O allocation (in the case of CPU with built-in ET-LAN)

I/O number	Name	Description
X80 to X8F	General-purpose communication clear to send flag	Turns ON when general-purpose communication is in a connected status.
Y80 to Y8F	General-purpose communication sending flag	Turns ON when sending with general-purpose communication.
Y70 to Y7F	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

(Note 2) In the case of the GPTRNS and pGPSEND instructions, the sending flag and send results flag are updated after the scan is complete.

(Note 3) In the case of the GPSEND instruction, the sending flag and send results flag are updated after sending is completed when the next GPSEND instruction is executed.

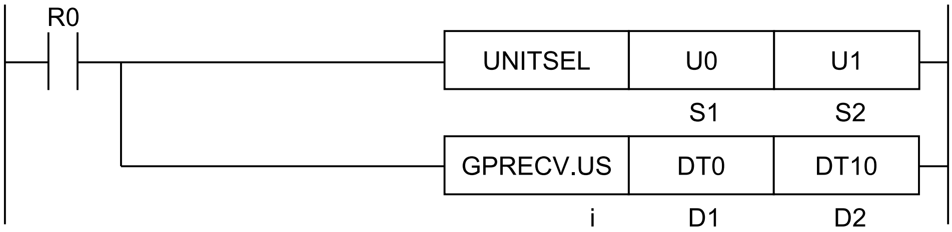
■ Flag operations

Name	Description
SR7	To be set in the case of out-of-range in indirect access (index modification).

Name	Description
SR8 (ER)	The connection specified with UNITSEL is closed (other than "Connect").
	The communication mode of the communication port specified with UNITSEL is not "General-purpose communication."
	When the data device specified by [S] exceeds the area
	When the number of sent data specified by [n] is 0. The volume including a start code and an end code exceeds the specified maximum value.
	When the number of sent data specified by [n] exceeds the data area
	Either 0 or a negative value is set for [N] in the settings of sending to ET-LAN.
	Set when executed in an interrupt program.

15.3 GPRECV (General-Purpose Communication Receive Instruction)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U0 (CPU unit with built-in SCU) and S2=U1 (port number 1) are specified by the UNITSEL instruction.

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
D1	Starting address of the received data storage data area
D2	Ending address of the received data storage data area

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D1 (Note 1)	●	●	●	●			●	●													●
D2 (Note 1)	●	●	●	●			●	●													●

(Note 1) Always 16-bit data/device, regardless of the specification of operation units [i].

■ Outline of operation

- This instruction reads data that is sent from an external device and received by the communication port of the unit.
- This instruction reads received data from the communication unit and the communication port set by the UNITSEL instruction, and stores the number of received bytes in the area specified by [D1] and the received data in the areas [D1+1] to [D2].

- In the case of SCU, data received from the partner are stored in 8 receive buffers for each COM port. By executing the GPRECV instruction, data in the receive buffer can be copied to a given operation memory.
- In the case of CPU with built-in ET-LAN, data received from the partner are stored in 1 receive buffer for each connection. By executing the GPRECV instruction, data in the receive buffer can be copied to a given operation memory.

■ Amount of received data and end code

Items	SCU	In the case of CPU with built-in ET-LAN
Number of received data	0 to 4096(Note 1)	0 to 16384
End code identification	Yes (according to the SCU communication settings (end settings))	No identification

(Note 1) For SCU, up to 4096 bytes can be sent, including a start code and an end code. The start code and end code are set from the configuration menu of the tool software or with the PMSET instruction.

When "Start code STX" is set to "Enabled", the maximum amount of sent data that can be specified is decremented by one.

When "Terminator setting" is set to "ETX" or "CR", the maximum amount of sent data is decremented by one.

When "Terminator setting" is set to "CR+LF", the maximum amount of sent data is decremented by two.

When "Terminator setting" is set to "Time", the maximum amount of sent data is not decremented.

■ Storage method for received data

DT200	U5	
DT201	H32(2)	H31(1)
DT202	H34(4)	H33(3)
DT203	-----	H35(5)
DT204	-----	-----
-----	-----	-----

Received number of bytes are stored.

Received data are stored in ascending order from lower bytes.

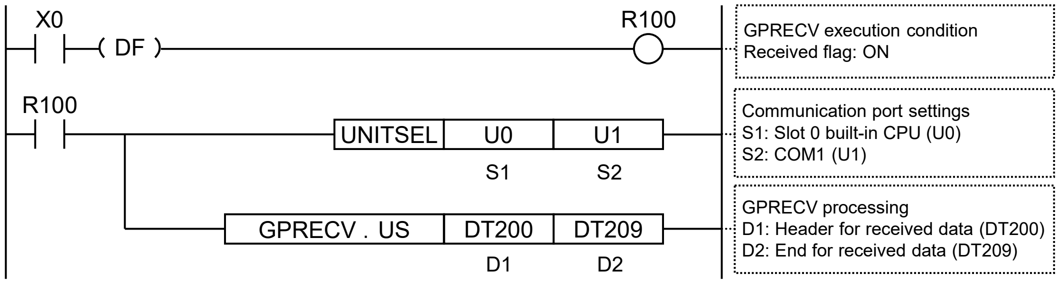
If the number of received bytes is smaller than the area specified by the GPRECV instruction, the value is not overwritten.

Info.

- The case of SCU shows the case that it is used in the following combination.
  - COM.0 port equipped in the CPU unit
  - Communication cassettes attached to the CPU unit (COM.1 to COM.2 ports)
  - Communication cassettes attached to the serial communication unit (COM.1 to COM.4 ports)
- As the communication cassette (Ethernet type) has an Ethernet-serial conversion function, the internal interface operates with similar programs as the case of SCU. The setting method and programming method are different from those for the CPU with built-in ET-LAN.

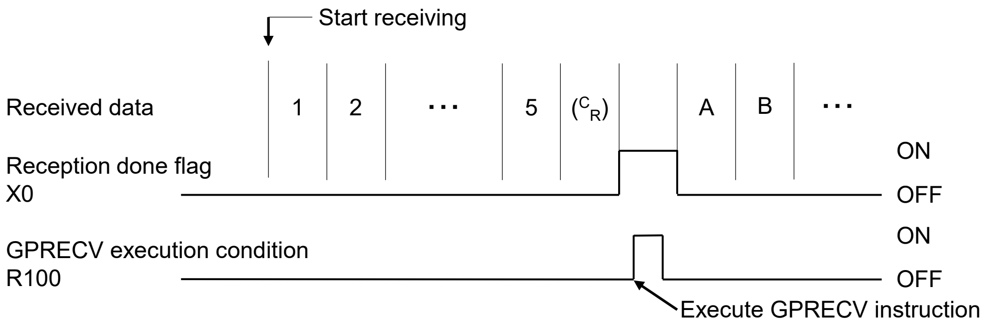
Sample program (in the case of SCU)

- When the received flag (X0) turns ON, the reception program is started up by the GPRECV instruction.
- Using the UNITSEL instruction, specify the slot number (U0) and the COM. port number (U1).
- In the GPRECV instruction, specify and execute the start of the data table that stores the received message (DT200) and the final address (DT209).

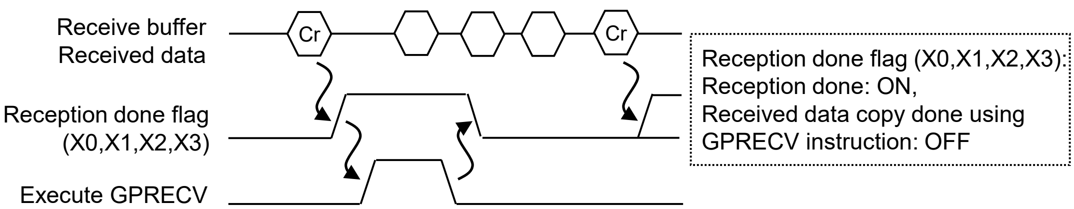


Time chart (in the case of SCU)

- Data received from an external device are stored in the receive buffer.
- When the end code is received, the received flag (X0, X1, X2, X3) turns ON. Subsequently, the following data are stored in the buffers upon reception. Data for 8 buffers can be received consecutively.



- When the GPRECV instruction is executed, data are copied to the specified area, and the received flags (X0, X1, X2, X3) are turned OFF. The received flags (X0, X1, X2, X3) are turned OFF when I/O refresh is executed at the start of the following scans.



■ I/O allocation (in the case of CPU with built-in SCU)

COM Port No.			Name	Description
1	2	0		
X0	X1	X2	General-purpose communication received flag	Turns ON when the receiving process is completed in the general-purpose communication mode.
X4	X5	X6	General-purpose communication received data copied flag	Turns ON when the GPRECV instruction is executed and the received data have been copied into the specified operation memory. Turns OFF when there are no applicable data.

■ I/O allocation (in the case of Serial Communication Unit)

COM Port No.				Name	Description
1	2	3	4		
X0	X1	X2	X3	General-purpose communication received flag	Turns ON when the receiving process is completed in the general-purpose communication mode.
X4	X5	X6	X7	General-purpose communication received data copied flag	Turns ON when the GPRECV instruction is executed and the received data have been copied into the specified operation memory. Turns OFF when there are no applicable data.

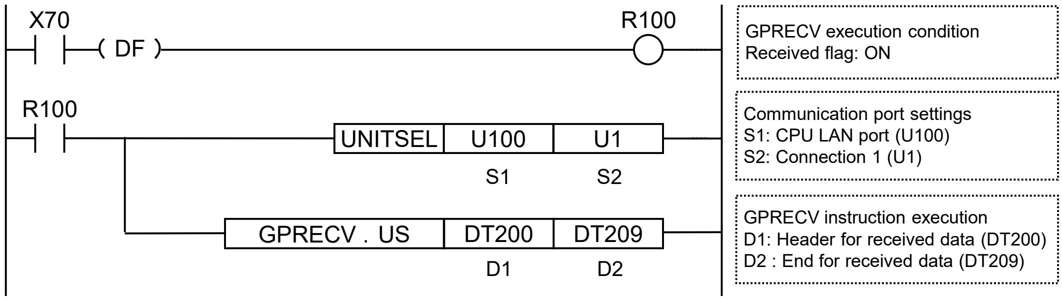
■ Precautions during programming (in the case of SCU)

- Use the UNITSEL instruction immediately before the GPRECV instruction to specify a target port for communication.
- When the general-purpose communication received flag is ON for the targeted COM port, execute GPRECV.
- When multiplex reception is in progress, the received flag remains ON after the received data have been copied using the GPRECV instruction. The received data cannot be copied at the leading edge of the reception done signal.
- The received data copied by the GPRECV instruction do not include a start code or an end code.
- It is also possible to receive binary data using the GPRECV instruction. In this case, "time" should be used for the end setting.
- The received data or the received data amount do not include the end code. (It is stripped off.)
- In the case of SCU which has eight 4096-byte buffers inside, data equivalent to the eight buffers can be received consecutively.
- If the reception of the ninth datum of data is performed by SCU before the GPRECV instruction is executed to take out data from SCU's receive buffer, a buffer FULL error occurs in SCU, and the ninth datum is discarded.
- If the GPRECV instruction is executed when the receive buffer FULL error is on, the oldest received datum is taken out, and the receive buffer FULL error is canceled.
- When no data have been received, the general-purpose communication control flag (received copy flag) turns OFF.
- After data have been received, and copy to the operation memory of the CPU unit has been completed, the general-purpose communication control flag (received data copied flag) turns ON.

- In the case of a direct address and an index modification address, specify the same device for D1 and D2. At the same time, specify the addresses so that D1 is less than D2.

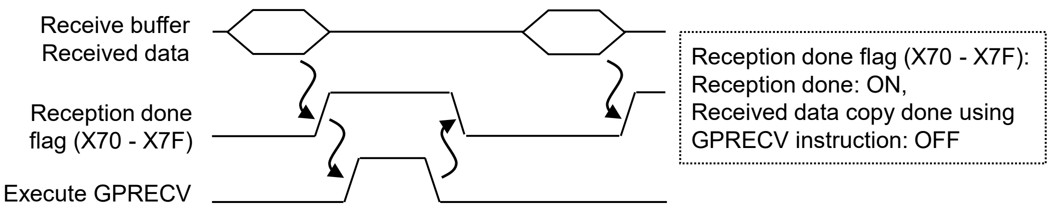
■ Sample program (in the case of CPU with built-in ET-LAN)

- When the received flag (X70) of Connection 1 turns ON, the reception program is started up by the GPRECV instruction.
- The UNITSEL instruction is used to specify a slot number (LAN port: U100) and the connection number (U1).
- In the GPRECV instruction, specify and execute the start of the data table that stores the received message (DT200) and the final address (DT209).



■ Time chart (in the case of CPU with built-in ET-LAN)

- Data received from an external device are stored in the receive buffer for each connection.
- When data are received, the received flag (X70 to X7F) turns ON.
- When the GPRECV instruction is executed, data are copied to the specified area, and the received flags (X70 to X7F) are turned OFF. The received flags (X70 to X7F) are turned OFF when I/O refresh is executed at the start of the following scans.



■ I/O allocation (in the case of CPU with built-in ET-LAN)

I/O allocations	Name	Description
X70 to X7F	General-purpose communication received flag	Turns ON when receiving is completed in the general-purpose communication mode.

■ Precautions during programming (in the case of CPU with built-in ET-LAN)

- Use the UNITSEL instruction immediately before the GPRECV instruction to specify a target connection for communication.
- When the general-purpose communication received flag is ON for the targeted connection, execute GPRECV.
- The maximum volume of data that can be received in one session using the GPRECV instruction, from the LAN port of the FP7 CPU unit, is 16,384 bytes.

- Depending on the communication format of an external device, if a header and a terminator are contained, they are stored in the operation memory as part of receive data. When necessary, insert a program to extract data content.
- In the case of a direct address and an index modification address, specify the same device for D1 and D2. At the same time, specify the addresses so that D1 is less than D2.

Info.

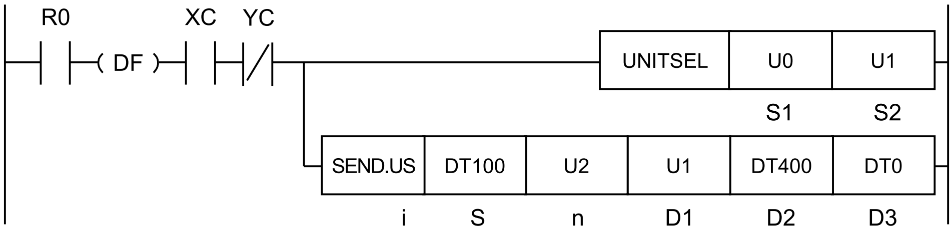
- As the communication cassette (Ethernet type) has an Ethernet-serial conversion function, the internal interface operates with similar programs as the case of SCU. The setting method and programming method are different from those for the CPU with built-in ET-LAN. Refer to the section describing the case of SCU.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the destination range is outside the accessible range.
	An SCU or ET-LAN unit does not exist in the slot specified by UNITSEL.
	The communication mode of the communication port specified with UNITSEL is not "General-purpose communication".
	The COM port specified by UNITSEL does not exist.
	Connection specified by UNITSEL is in a "reception done OFF" status, but not in a "connected" status.
	Data device specified by [D1] and/or [D2] exceeds the area.
	The specified [D1] is greater than or equal to [D2].
	The devices specified for [D1] and [D2] differ.

15.4 SEND (MEWTOCOL Master / MODBUS Master)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U0 (CPU unit with built-in SCU) and S2=U1 (port number 1) are specified by the UNITSEL instruction.

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S	Starting address of the sender data area
n	Amount of sent data
D1	Partner station number
D2	Starting address of the device in the receiver data area of the partner unit
D3	Starting address of the device area of the master unit that stores the execution result code (1 word)

■ Available word devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●													●
n	●	●	●	●			●	●								●	●				●
D1	●	●	●	●			●	●								●	●				●
D2 (Note 1)	(Note 2)	●	●	(Note 2)			●	(Note 2)													●
D3	●	●	●	●			●	●													●

(Note 1) When the destination unit is FP7, only global devices can be specified. (Local devices cannot be specified.)

(Note 2) In the MODBUS mode, this cannot be specified as the receiver.

■ Available bit devices (●: Available)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
S	●	●	●	●								●	●	●
n														
D1														
D2 (Note 1)	(Note 2)	●	●	(Note 2)										●
D3														

(Note 1) When the receiver is FP7, only global devices can be specified. (A local device cannot be specified.)

(Note 2) In the case of MODBUS and MEWTOCOL-COM, a bit device cannot be specified.

■ Outline of operation

- Commands are sent from the communication port of the unit to perform the data transmission with external devices. Message in accordance with the protocol is automatically formulated by PLC. The user program only has to specify the station number and the memory address, and execute the SEND/RECV instruction, to carry out reading and writing.
- Select a**communication mode**in the configuration menu of the tool software FPWIN GR7.
- When the SEND instruction is executed, data are read from the device in the master unit, starting with [S], and stored in the address starting with [D2] of the partner unit.
- The transfer method (register transfer/bit transfer) varies according to the device types that are specified by [S] and [D2].
- The amount of sent data [n] is specified in words for the register transfer, and in the number of bits for the bit transfer.
- The execution result code is stored in the one-word area of the master unit that is specified by [D3].

■ Setting the amount of sent data [n]

Transfer method	Communication Mode	Amount of sent data n	Note
Register transfer	MEWTOCOL-COM	1 to 507 words	
	MEWTOCOL-DAT	1 to 1020 words	Connection setting: MEWTOCOL Communication type setting: Connect with FP2 ET-LAN
		1 to 2038 words	Connection setting: MEWTOCOL Communication type setting: Do not connect with FP2 ET-LAN
	MODBUS	1 to 127 words	Use MODBUS command 15 (to write to WY or WR) and command 16 (to write multiple words to DT).
Bit transfer	MEWTOCOL-COM	Fixed at 1 bit	During MEWTOCOL-COM, WCS command is used.

Transfer method	Communication Mode	Amount of sent data n	Note
	MEWTOCOL-DAT	Fixed at 1 bit	During MEWTOCOL-DAT, contact information write 52H is used.
	MODBUS	1 to 2040	Use the force multiple coils command 15.

(Note 1) The transfer method varies according to the device type specified for operands [S] and [D2]. The register transfer is used for 16-bit devices and the bit transfer is used for 1-bit devices.

(Note 2) The amount of sent data is specified in words for the register transfer and in bits for the bit transfer.

■ Specification of partner unit station number [D1]

Communication Mode	When SCU is used	When ET-LAN is used
MEWTOCOL-COM	1 to 99, 238 (decimal) = EE (hexadecimal)	1 to 64, 238 (decimal) = EE (hexadecimal)
MEWTOCOL-DAT	Non-SCU-compliant	
MODBUS	0 to 255	0 to 255

(Note 1) In the case of SCU, when "0" is specified for the partner station number, global transfer is selected. In this case, there is no response message from the destination.

(Note 2) For connection between FP7 and FP7, specify "1". Destination is determined by the IP address.

■ Specification of destination address [D2]

Transfer method	Communication Mode	Address range
Register transfer	MEWTOCOL-COM	0 to 99999
	MEWTOCOL-DAT	0 to 65535F
	MODBUS	0 to 65535 (H FFFF)
Bit transfer	MEWTOCOL-COM	0 to 999F
	MEWTOCOL-DAT	0 to 65535 (H FFFF)
	MODBUS	0 to 65535 (H FFFF)

(Note 1) When the receiver is the file register FL, specify a constant. Example) For FL100, specify U100. For the file register, only bank 0 can be specified.

■ Execution result code [D3]

Code	Description	Code	Description
H0	Normal end	H7	I/O allocation shortage error(Note 2)
H1	The communication port is being used in the master communication.	H8	The send buffer is being used.(Note 3)
H2	The communication port is being used in the slave communication.	H41	Format error
H3	The number of master communication instructions simultaneously used is exceeded.	H60	Parameter error
H4	Transmission timeout	H61	Data error
H5	Response reception timeout	H91	Missing expansion slave unit error

Code	Description	Code	Description
H6	Reception error(Note 1)		

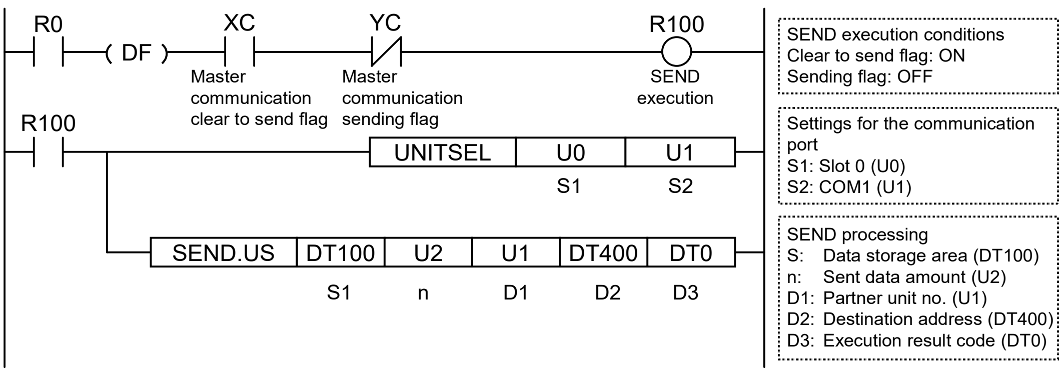
(Note 1) It occurs when an abnormal telegram is received. When there is a format error in the header of an individual protocol, the communication discards the received data and a response reception timeout occurs.

(Note 2) It occurs when the communication control I/O relays corresponding to the communication port (master communication clear to send flag, master communication send active flag, master communication send done result relay)are not allocated as I/O words of the CPU unit in the I/O map. It occurs only when the number of user connections of ET-LAN is expanded and this instruction is executed specifying the expanded connections.

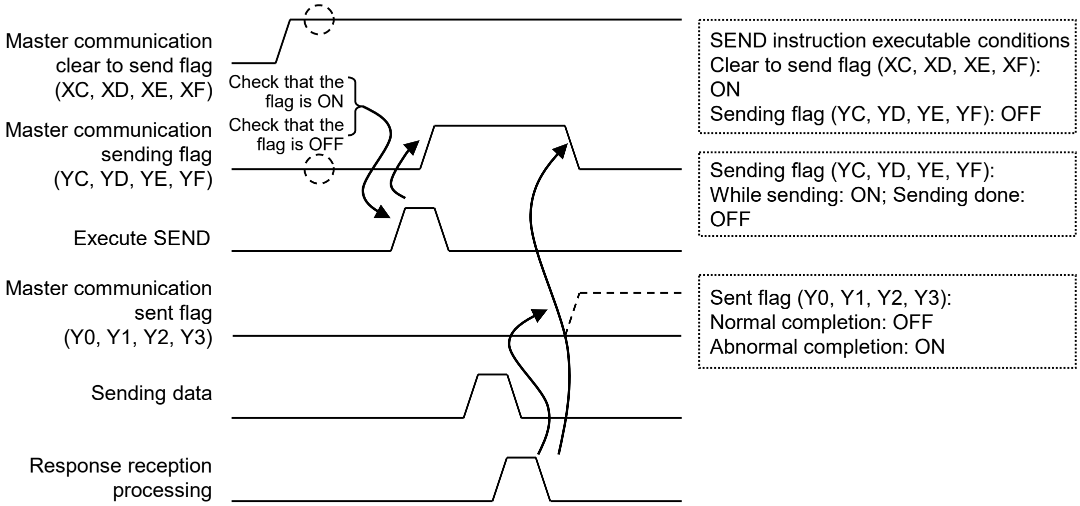
(Note 3) Effective when the version is 4.57 or later.

■ Sample program (in the case of SCU)

- This program sends the command from the COM1 port of the CPU unit to write the content of PLC's data registers DT100 to DT101 into the data areas DT400 to DT401 of the external device (station number 1).
- This program checks that the master mode is on (XC) and that sending is not in progress in the same port (YC), and then starts up the SEND instruction.
- Using the UNITSEL instruction, specify the slot number (U0) and the COM. port number (U1).
- In the SEND instruction, specify and execute the source's starting address (DT100) and data amount (U2), the destination's station number (U1) and starting address (DT400).



■ Time chart (in the case of SCU)



Info.

- The case of SCU shows the case that it is used in the following combination.
  - COM.0 port equipped in the CPU unit
  - Communication cassettes attached to the CPU unit (COM.1 to COM.2 ports)
  - Communication cassettes attached to the serial communication unit (COM.1 to COM.4 ports)
- As the communication cassette (Ethernet type) has an Ethernet-serial conversion function, the internal interface operates with similar programs as the case of the CPU with built-in SCU. The setting method and programming method are different from those for the CPU with built-in ET-LAN. The communication cassette (Ethernet type) does not support MODBUS.

■ I/O allocation (in the case of CPU with built-in SCU)

COM Port No.			Name	Description
1	2	0		
XC	XD	XE	Master communication clear to send flag	Turns ON when MEWTOCOL-COM, MEWTOCOL7, or MODBUS-RTU is set for the communication mode, and the unit is in the RUN mode.
YC	YD	YE	Master communication sending flag	Turns ON during sending data based on SEND/RECV instruction. Turns OFF when the sending process is completed.
Y0	Y1	Y2	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

I/O allocation (in the case of Serial Communication Unit)

COM Port No.				Name	Description
1	2	3	4		
XC	XD	XE	XF	Master communication clear to send flag	Turns ON when MEWTOCOL-COM, MEWTOCOL7, or MODBUS-RTU is set for the communication mode, and the unit is in the RUN mode.
YC	YD	YE	YF	Master communication sending flag	Turns ON during sending data based on SEND/RECV instruction. Turns OFF when the sending process is completed.
Y0	Y1	Y2	Y3	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

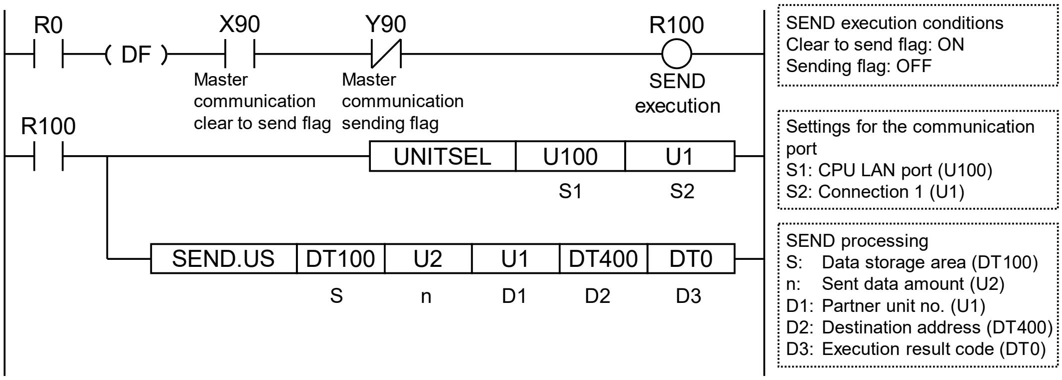
(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

Precautions during programming (in the case of SCU)

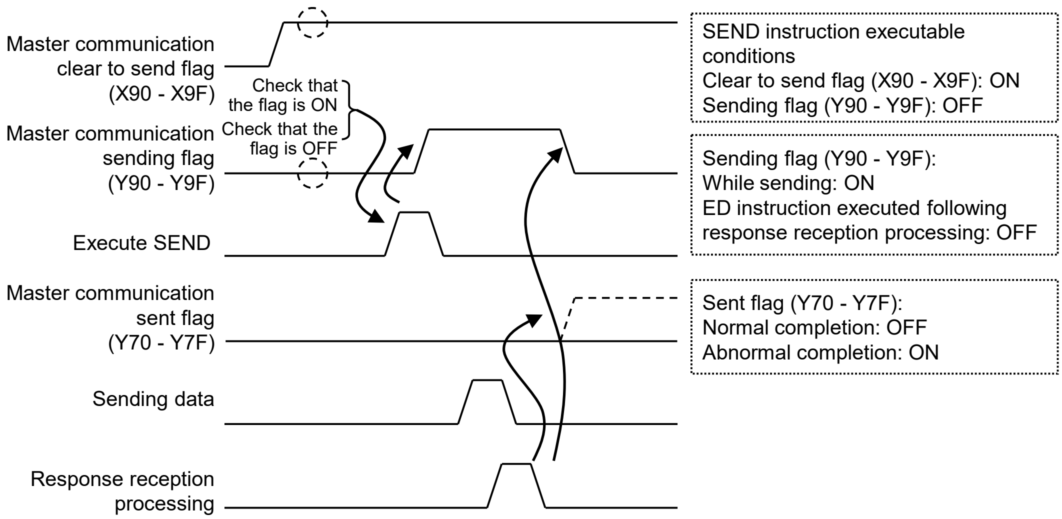
- Use the UNITSEL instruction immediately before the SEND/RECV instruction to specify a target port for communication.
- Master communication is only enabled when MEWTOCOL or MODBUS is selected. Confirm that the "master communication clear to send flags" (XC to XF) are ON for the corresponding channel, and execute the SEND/RECV instruction.
- Another SEND/RECV instruction cannot be executed for a communication port where master communication is in progress. Confirm that the "master communication sending flags" (YC to YF) are OFF, and execute the instruction.
- A SEND/RECV instruction cannot be executed for a port where slave communication is in progress.
- If there is no response, the "master communication sending flags" (YC to YF) remain ON during the time-out period set in the CPU configuration.
- Up to 16 SEND/RECV instructions can be executed simultaneously for differing COM ports.

Sample program (in the case of CPU with built-in ET-LAN)

- This program sends the command from the LAN port of the CPU unit to write the content of PLC's data registers DT100 to DT101 into the data areas DT400 to DT401 of the external device.
- After it is confirmed that connection 1 is established in master mode (X90) and no transmissions are currently being executed for the same port (Y90), the SEND instruction is started.
- The UNITSEL instruction is used to specify a slot number (LAN port: U100) and the connection number (U1).
- In the SEND instruction, specify and execute the source's starting address (DT100) and data amount (U2), the destination's station number (U1) and starting address (DT400).



■ Time chart (in the case of CPU with built-in ET-LAN)



■ I/O allocation (in the case of CPU with built-in ET-LAN)

I/O number	Name	Description
X90 to X9F	Master communication clear to send flag	Turns ON when a connection is established in the master communication.
Y90 to Y9F	Master communication sending flag	Turns ON during sending data based on the SEND/RECV instruction. Turns OFF when the ED instruction is executed after the completion of the response receive processing.
Y70 to Y7F	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

■ Precautions during programming (in the case of CPU with built-in ET-LAN)

- Use the UNITSEL instruction immediately before the SEND/RECV instruction to specify a target connection number for communication.

- Master communication is only enabled when MEWTOCOL or MODBUS is selected. Confirm that the "master communication clear to send flags" (X90 to X9F) are ON for the corresponding connection, and execute the SEND/RECV instruction.
- The SEND or RECV instruction cannot be executed for the connection for which the master communication is in progress. Confirm that the "master communication sending flags" (Y90 to Y9F) are OFF, and execute the instruction.
- The SEND or RECV instruction cannot be executed for the connection for which the slave communication is in progress.
- Up to 16 SEND/RECV instructions can be executed simultaneously for differing connections.
- For communication between LAN ports of FP7, specify "U1" for the partner station number. The receiver is determined by the IP address.

Info.

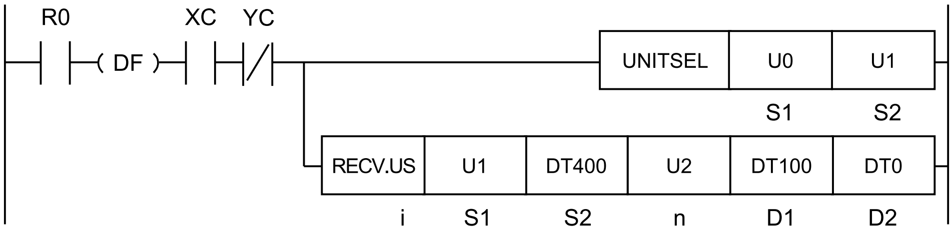
- As the communication cassette (Ethernet type) has an Ethernet-serial conversion function, the internal interface operates with similar programs as the case of the CPU with built-in SCU. The setting method and programming method are different from those for the CPU with built-in ET-LAN. Refer to the section describing the case of SCU.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the source range is outside the accessible range.
	COM port or connection specified by UNITSEL does not exist, or communication is not possible in the specified connection.
	Data device specified by [S] is invalid, or exceeds the area.
	Sent data amount specified by [n] is invalid.
	Station number specified by [D1] is out of the range.
	Data device specified by [D2] is invalid, or exceeds the area.
	Result storage device specified by [D3] is invalid.
	Integer specification for [D2] is only available for the MODBUS direct address specification type. It is invalid for other types.
	Specified bit devices for [S] and [D2], and/or specified 16-bit device, differ.

15.5 RECV (MEWTOCOL Master / MODBUS Master)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U0 (CPU unit with built-in SCU) and S2=U1 (port number 1) are specified by the UNITSEL instruction.

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S1	Partner station number
S2	Starting address of the device in the sender data area of the partner unit
n	Number of received data
D1	Starting address of the device in the receiver data area of the master unit
D2	Starting address of the device area of the master unit that stores the execution result code (1 word)

■ Available word devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●								●	●				●
S2(Note 1)	●	●	●	●			●	●													●
n	●	●	●	●			●	●								●	●				●
D1	●	●	●	●			●	●													●
D2	●	●	●	●			●	●													●

(Note 1) When the source unit is FP7, only global devices can be specified. (Local devices cannot be specified.)

■ Available bit devices (●: Available)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
S1														
S2(Note 1)	●	●	●	(Note 2)								(Note 3)	(Note 3)	●
n														
D1	●	●	●	●								●	●	●
D2														

(Note 1) When the sender is FP7, only global devices can be specified. (A local device cannot be specified.)

(Note 2) In the case of MODBUS mode, a bit device cannot be specified.

(Note 3) In the case of MEWTOCOL-COM mode or MODBUS mode, a bit device cannot be specified.

■ Outline of operation

- Commands are sent from the communication port of the unit to perform the data transmission with external devices.
- Data can be read and written by specifying station numbers and memory addresses and executing the SEND and RECV instructions in a user program, because PLC automatically creates messages according to the protocol.
- Select a**communication mode**in the**configuration**menu of the tool software FPWIN GR7.
- When the RECV instruction is executed, the data is read from the address that starts with [S2] in the partner station number [S1] and the data is stored in the area that starts with [D1] in the master unit.
- Depending on the type of device specified by [S2] and [D1], the transfer method (register transfer / bit transfer) varies.
- The amount of received data [n] is specified in words for the register transfer, and in the number of bits for the bit transfer.
- The execution result code is stored in the one-word area of the master unit that is specified by [D2].

■ Specification of partner unit station number [S1]

Communication Mode	When SCU is used	When ET-LAN is used
MEWTOCOL-COM	1 to 99, 238 (decimal) = EE (hexadecimal)	1 to 64, 238 (decimal) = EE (hexadecimal) (Note 1)
MEWTOCOL-DAT(Note 1)	Non-SCU-compliant	
MODBUS	1 to 255	1 to 255

(Note 1) For connection between FP7 and FP7, specify "1". Destination is determined by the IP address.

■ Specify the starting address [S2] of the sender data area

Transfer method	Communication Mode	Address range
Register transfer	MEWTOCOL-COM	0 to 99999
	MEWTOCOL-DAT	0 to 65535F

Transfer method	Communication Mode	Address range
	MODBUS	0 to 65535 (H FFFF)
Bit transfer	MEWTOCOL-COM	0 to 999F
	MEWTOCOL-DAT	0 to 65535 (H FFFF)
	MODBUS	0 to 65535 (H FFFF)

(Note 1) When the receiver is the file register FL, specify a constant. Example) For FL100, specify U100. For the file register, only bank 0 can be specified.

■ Specification of the amount of received data [n]

Transfer method	Communication Mode	Types of communication port	Setting range
Register transfer (Note 1)	MEWTOCOL-COM	1 to 509 words	RCC command and RD command are used.
	MEWTOCOL-DAT	1 to 1020 words	Connection setting: Setting of the MEWTOCOL Communication type setting: Connect with FP2 ET-LAN
		1 to 2038 words	Connection setting: Setting of the MEWTOCOL Communication type setting: Do not connect with FP2 ET-LAN
	MODBUS	1 to 127 words	Command 1 is used for reading WY and WR. Command 2 is used for WX. Command 3 is used for reading DT Command 4 is used for reading WL and LD
Bit transfer (Note 2)	MEWTOCOL-COM	Fixed to 1 bit	During MEWTOCOL-COM, RCS command is used.
	MEWTOCOL-DAT	Fixed to 1 bit	During MEWTOCOL-DAT, read contact information 53H is used.
	MODBUS	1 to 2040 bits	Command 1 is used for reading Y and R. Command 2 is used for X.

(Note 1) When 16-bit devices are specified for sender [S] and receiver [D2]

(Note 2) When bit devices are specified for sender [S] and receiver [D2]

■ Execution result code [D2]

Code	Description	Code	Description
H0	Normal end	H7	I/O allocation shortage error(Note 2)
H1	The communication port is being used in the master communication.	H8	The send buffer is being used.(Note 3)
H2	The communication port is being used in the slave communication.	H41	Format error
H3	The number of master communication instructions simultaneously used is exceeded.	H60	Parameter error
H4	Transmission timeout	H61	Data error
H5	Response reception timeout	H91	Missing expansion slave unit error
H6	Reception error(Note 1)		

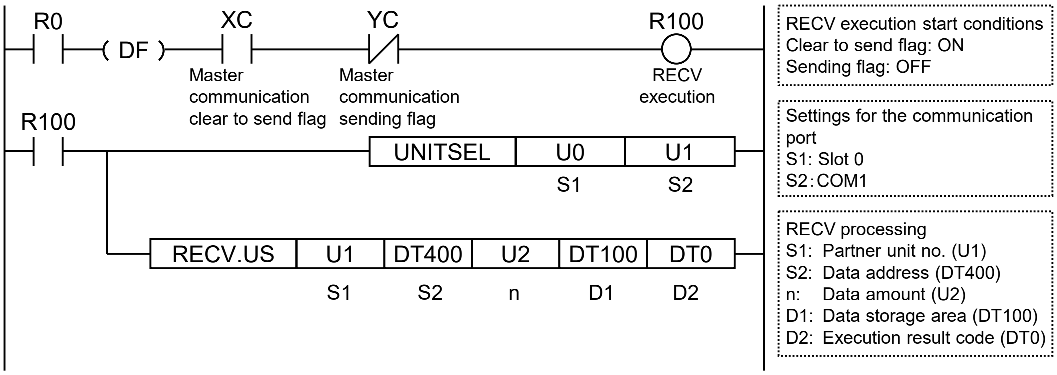
(Note 1) It occurs when an abnormal telegram is received. When there is a format error in the header of an individual protocol, the communication discards the received data and a response reception timeout occurs.

(Note 2) It occurs when the communication control I/O relays corresponding to the communication port (master communication clear to send flag, master communication send active flag, master communication send done result relay)are not allocated as I/O words of the CPU unit in the I/O map. It occurs only when the number of user connections of ET-LAN is expanded and this instruction is executed specifying the expanded connections.

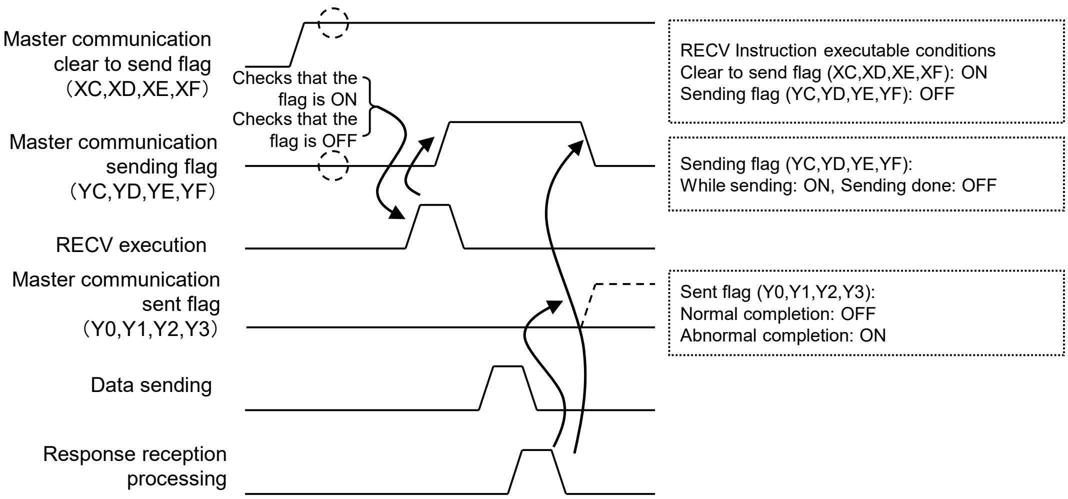
(Note 3) Effective when the version is 4.57 or later.

■ Sample program (in the case of SCU)

- This program sends the command from the COM1 port of the CPU unit, reads the data from the data areas DT400 to DT401 of the external device (station number 1), and writes the data into the data registers DT100 to DT101 of the PLC.
- This program checks that the master mode is on (XC) and that sending is not in progress in the same port (YC), and then starts up the RECV instruction.
- Using the UNITSEL instruction, specify the slot number (U0) and the COM. port number (U1).
- In the RECV instruction, specify and execute the partner station number (U1), starting address (DT400), data amount (U2), and PLC's starting address to store the data (DT100).



■ Time chart (in the case of SCU)



Info.

- The case of SCU shows the case that it is used in the following combination.
  - COM.0 port equipped in the CPU unit
  - Communication cassettes attached to the CPU unit (COM.1 to COM.2 ports)
  - Communication cassettes attached to the serial communication unit (COM.1 to COM.4 ports)
- As the communication cassette (Ethernet type) has an Ethernet-serial conversion function, the internal interface operates with similar programs as the case of the CPU with built-in SCU. The setting method and programming method are different from those for the CPU with built-in ET-LAN. The communication cassette (Ethernet type) does not support MODBUS.

■ I/O allocation (in the case of CPU with built-in SCU)

COM Port No.			Name	Description
1	2	0		
XC	XD	XE	Master communication clear to send flag	Turns ON when MEWTOCOL-COM, MEWTOCOL7, or MODBUS-RTU is set for the communication mode, and the unit is in the RUN mode.
YC	YD	YE	Master communication sending flag	Turns ON during sending data based on SEND/RECV instruction. Turns OFF when the sending process is completed.
Y0	Y1	Y2	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

■ I/O allocation (in the case of Serial Communication Unit)

COM Port No.				Name	Description
1	2	3	4		
XC	XD	XE	XF	Master communication clear to send flag	Turns ON when MEWTOCOL-COM, MEWTOCOL7, or MODBUS-RTU is set for the communication mode, and the unit is in the RUN mode.
YC	YD	YE	YF	Master communication sending flag	Turns ON during sending data based on SEND/RECV instruction. Turns OFF when the sending process is completed.
Y0	Y1	Y2	Y3	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

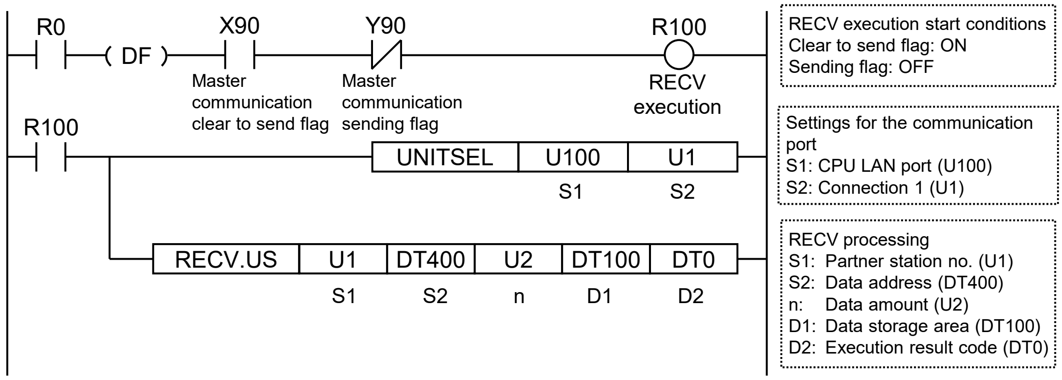
■ Precautions during programming (in the case of SCU)

- Use the UNITSEL instruction immediately before the SEND/RECV instruction to specify a target port for communication.
- Master communication is only enabled when MEWTOCOL or MODBUS is selected. Confirm that the "master communication clear to send flags" (XC to XF) are ON for the corresponding channel, and execute the SEND/RECV instruction.

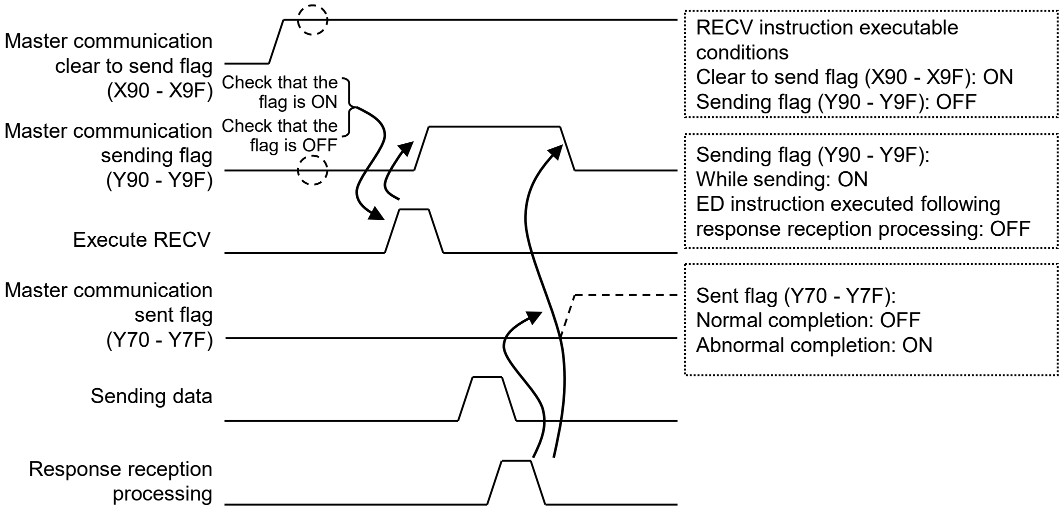
- Another SEND/RECV instruction cannot be executed for a communication port where master communication is in progress. Confirm that the "master communication sending flags" (YC to YF) are OFF, and execute the instruction.
- A SEND/RECV instruction cannot be executed for a port where slave communication is in progress.
- If there is no response, the "master communication sending flags" (YC to YF) remain ON during the time-out period set in the CPU configuration.
- Up to 16 SEND/RECV instructions can be executed simultaneously for differing COM ports.

■ Sample program (in the case of CPU with built-in ET-LAN)

- This program sends the command from the LAN port of the CPU unit, reads the data from the data areas DT400 to DT401 of the external device, and writes the data into the data registers DT100 to DT101 of the PLC.
- After it is confirmed that connection 1 is established in master mode (X90) and no transmissions are currently being executed for the same port (Y90), the RECV instruction is started.
- The UNITSEL instruction is used to specify a slot number (LAN port: U100) and the connection number (U1).
- In the RECV instruction, specify and execute the partner station number (U1), starting address (DT400), data amount (U2), and PLC's starting address to store the data (DT100).



■ Time chart (in the case of CPU with built-in ET-LAN)



■ I/O allocation (in the case of CPU with built-in ET-LAN)

I/O number	Name	Description
X90 to X9F	Master communication clear to send flag	Turns ON when a connection is established in the master communication.
Y90 to Y9F	Master communication sending flag	Turns ON during sending data based on the SEND/RECV instruction. Turns OFF when the ED instruction is executed after the completion of the response receive processing.
Y70 to Y7F	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

■ Precautions during programming (in the case of CPU with built-in ET-LAN)

- Use the UNITSEL instruction immediately before the SEND/RECV instruction to specify a target connection number for communication.
- Master communication is only enabled when MEWTOCOL or MODBUS is selected. Confirm that the "master communication clear to send flags" (X90 to X9F) are ON for the corresponding connection, and execute the SEND/RECV instruction.
- The SEND or RECV instruction cannot be executed for the connection for which the master communication is in progress. Confirm that the "master communication sending flags" (Y90 to Y9F) are OFF, and execute the instruction.
- The SEND or RECV instruction cannot be executed for the connection for which the slave communication is in progress.
- Up to 16 SEND/RECV instructions can be executed simultaneously for differing connections.
- For communication between LAN ports of FP7, specify "U1" for the partner station number. The receiver is determined by the IP address.

Info.

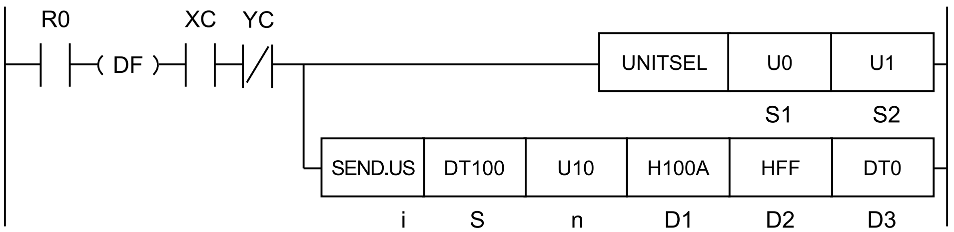
- As the communication cassette (Ethernet type) has an Ethernet-serial conversion function, the internal interface operates with similar programs as the case of the CPU with built-in SCU. The setting method and programming method are different from those for the CPU with built-in ET-LAN. Refer to the section describing the case of SCU.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the destination range is outside the accessible range.
	COM port or connection specified by UNITSEL does not exist, or communication is not possible in the specified connection.
	Partner station number specified by [S1] is out of the range.
	Partner unit sender data device specified by [S2] is invalid.
	Sent data amount specified by [n] is invalid.
	Data device of the receiver data area in the master unit specified by [D1] is invalid, or exceeds the area.
	Result storage device specified by [D2] is invalid.
	Specified bit devices for [S2] and [D1], and/or specified 16-bit device, differ.
	Integer specification for [S2] is only available for the MODBUS address direct specification type. It is invalid for other types.

15.6 SEND (MODBUS Master: Function Code Specification)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U0 (CPU unit with built-in SCU) and S2=U1 (port number 1) are specified by the UNITSEL instruction.

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Items	Settings		Setting range
S	Starting address of the sender data area		-
n	Amount of sent data		1 to 127 words 1 to 2040 bits
D1	MODBUS command to be used, and the partner station number		
	Higher byte	MODBUS function code (two hexadecimal digits)	H5, H6, HF, H10
	Lower byte	Partner station number (two hexadecimal digits)	H0 to HFF (0 to 255)
D2	MODBUS starting address of the receiver data area of the partner unit		H0 to HFFFF (0 to 65535)
D3	Starting address of the device area of the master unit that stores the execution result code (1 word)		-

■ Available word devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●													●
n	●	●	●	●			●	●								●	●				●
D1	●	●	●	●			●	●								●	●				●

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D2	●	●	●	●			●	●								●	(Note 1)				●
D3	●	●	●	●			●	●													●

(Note 1) When the destination unit is FP7, only global devices can be specified. (Local devices cannot be specified.)

■ Available bit devices (●: Available)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
S	●	●	●	●								●	●	●
n														
D1														
D2														
D3														

■ Outline of operation

- The MODBUS command is sent from the communication port of the unit to perform the data transmission with external devices.
- Data can be read and written by specifying station numbers and memory addresses and executing the SEND and RECV instructions in a user program, because PLC automatically creates messages according to the protocol.
- Select a communication mode in the configuration menu of the tool software FPWIN GR7.
- Specify the MODBUS command to be used, and the partner MODBUS station number, in a Hex format in [D1].
- When the SEND instruction is executed, data are read from the device in the master unit, starting with [S], and stored in the address starting with [D2] of the partner unit.
- The transfer method (register transfer/bit transfer) and the type of MODBUS command that can be used vary, depending on the type of device specified by [S] and data amount specified by [n].
- The amount of sent data [n] is specified in words for the register transfer, and in the number of bits for the bit transfer.
- The execution result code is stored in the one-word area of the master unit that is specified by [D3].

■ Specification of [S], [n] and [D1]

- The transfer method and the MODBUS function code that can be used vary, depending on the type of device specified by the operand [S] and the sent data amount specified by [n].

Types of device to be specified for [S]	Transfer method	Number of sent data [n]	Value that can be specified for high bytes of [D1]
16-Bit device: WX, WY, WR, WL, DT, LD	Register transfer	1	H6: Preset single register (06) HF: Force multiple coils (15) H10: Preset multiple registers (16)
		2 to 127	HF: Force multiple coils (15) H10: Preset multiple registers (16)
1-Bit device: X, Y, R, L, DT.n, LD.n	Bit transfer	1	H5: Force single coil (05) HF: Force multiple coils (15)
		2 to 2040	HF: Force multiple coils (15)

- The amount of sent data [n] is specified in words for the register transfer, and in bits for the bit transfer.
- Operand [D1] is specified as a combination of a two-digit hexadecimal MODBUS function code and a two-digit hexadecimal partner station number.

  Example: Specify "H100A" in the case of MODBUS function code 16 (preset multiple registers) and station number 10.
- In the case of SCU, when "0" is specified for the partner station number, global transfer is selected. In this case, there is no response message from the destination.

■ Execution result code [D3]

Code	Description	Code	Description
H0	Normal end	H6	Reception error(Note 1)
H1	The communication port is being used in the master communication.	H7	I/O allocation shortage error(Note 2)
H2	The communication port is being used in the slave communication.	H8	The send buffer is being used.(Note 3)
H3	The number of master communication instructions simultaneously used is exceeded.	H8001	Function code error
H4	Transmission timeout	H8002	Device quantity error (out of range)
H5	Response reception timeout	H8003	Device quantity error (out of range)

(Note 1) It occurs when an abnormal telegram is received. When there is a format error in the header of an individual protocol, the communication discards the received data and a response reception timeout occurs.

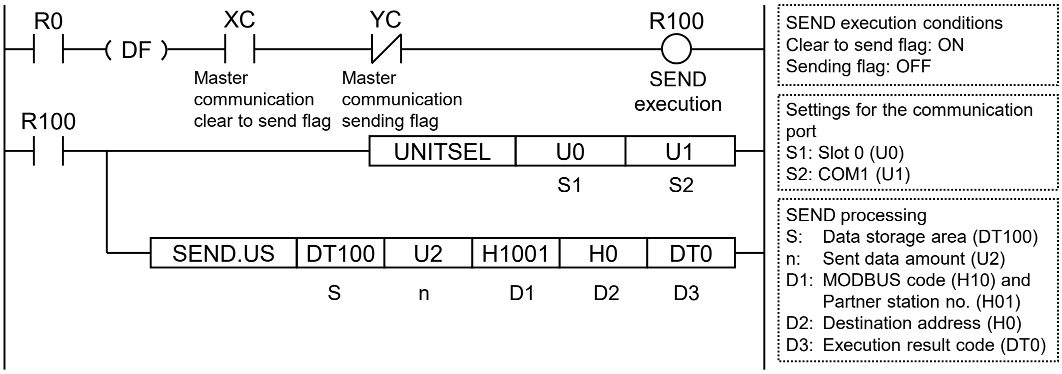
(Note 2) It occurs when the communication control I/O relays corresponding to the communication port (master communication clear to send flag, master communication send active flag, master communication send done result relay)are not allocated as I/O words of the CPU unit in the I/O map. It occurs only when the number of user connections of ET-LAN is expanded and this instruction is executed specifying the expanded connections.

(Note 3) Effective when the version is 4.57 or later.

■ Sample program (in the case of SCU)

- This program sends the command from the COM1 port of the CPU unit, and then writes the content of PLC's data registers DT100 to DT101 into the data areas 400001 to 400002 of the external device (station number 1).

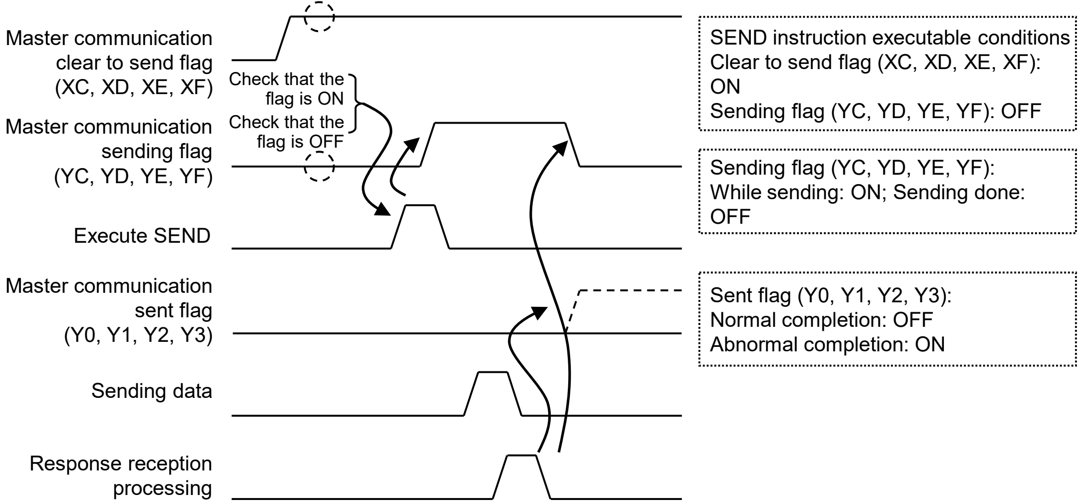
- This program checks that the master mode is on (XC) and that sending is not in progress in the same port (YC), and then starts up the SEND instruction.
- Using the UNITSEL instruction, specify the slot number (U0) and the COM. port number (U1).
- In the SEND instruction, specify and execute PLC's starting address (DT100) and data amount (U2), MODBUS function code to be used (16: H10), and partner station number (H01) and starting address (H0). Check addresses of connected devices in the instruction manuals of devices.



(Note 1) Operand [D1] of SEND instruction is specified by combining two hexadecimal digits of MODBUS function code with two hexadecimal digits of partner device station number. When the MODBUS function code is 16, [D1] H10 should be specified.

(Note 2) When the partner device is FP series PLC, Operand [D2] of SEND instruction can be specified using the device number.

■ Time chart (in the case of SCU)



Info.

- The case of SCU shows the case that it is used in the following combination.
  - COM.0 port equipped in the CPU unit
  - Communication cassettes attached to the CPU unit (COM.1 to COM.2 ports)
  - Communication cassettes attached to the serial communication unit (COM.1 to COM.4 ports)
- The communication cassette (Ethernet type) does not support MODBUS.

■ I/O allocation (in the case of CPU with built-in SCU)

COM Port No.			Name	Description
1	2	0		
XC	XD	XE	Master communication clear to send flag	Turns ON when MEWTOCOL-COM, MEWTOCOL7, or MODBUS-RTU is set for the communication mode, and the unit is in the RUN mode.
YC	YD	YE	Master communication sending flag	Turns ON during sending data based on SEND/RECV instruction. Turns OFF when the sending process is completed.
Y0	Y1	Y2	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

■ I/O allocation (in the case of Serial Communication Unit)

COM Port No.				Name	Description
1	2	3	4		
XC	XD	XE	XF	Master communication clear to send flag	Turns ON when MEWTOCOL-COM, MEWTOCOL7, or MODBUS-RTU is set for the communication mode, and the unit is in the RUN mode.
YC	YD	YE	YF	Master communication sending flag	Turns ON during sending data based on SEND/RECV instruction. Turns OFF when the sending process is completed.
Y0	Y1	Y2	Y3	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

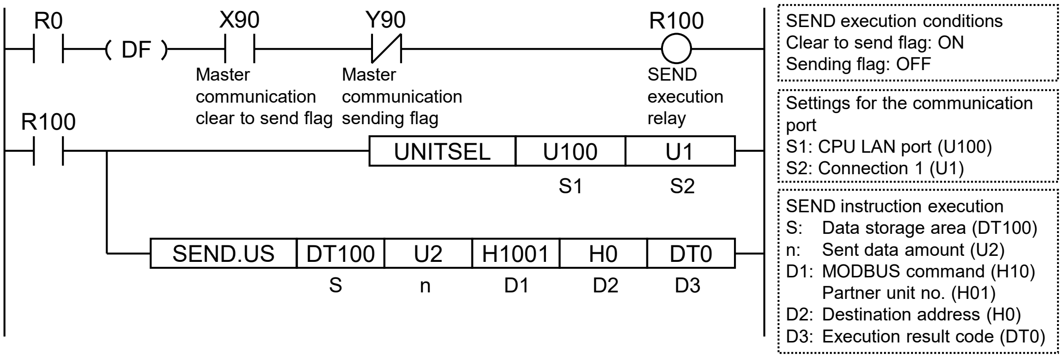
■ Precautions during programming (in the case of SCU)

- Use the UNITSEL instruction immediately before the SEND/RECV instruction to specify a target port for communication.
- Master communication is only enabled when MEWTOCOL or MODBUS is selected. Confirm that the "master communication clear to send flags" (XC to XF) are ON for the corresponding channel, and execute the SEND/RECV instruction.
- Another SEND/RECV instruction cannot be executed for a communication port where master communication is in progress. Confirm that the "master communication sending flags" (YC to YF) are OFF, and execute the instruction.

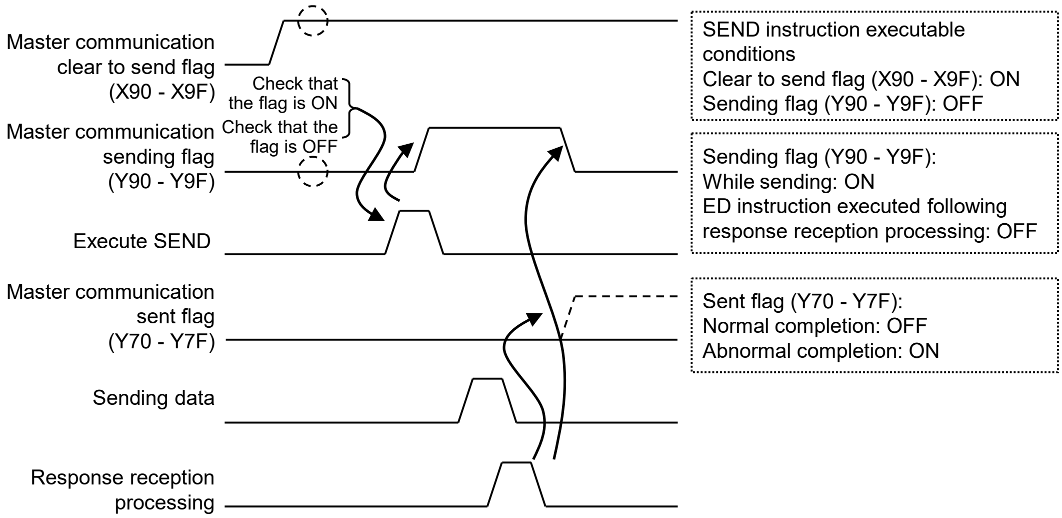
- A SEND/RECV instruction cannot be executed for a port where slave communication is in progress.
- If there is no response, the "master communication sending flags" (YC to YF) remain ON during the time-out period set in the CPU configuration.
- Up to 16 SEND/RECV instructions can be executed simultaneously for differing COM. ports.

■ Sample program (in the case of CPU with built-in ET-LAN)

- This program sends the MODBUS command (16) from the LAN port of the CPU unit, and then writes the content of PLC's data registers DT100 to DT101 into the data areas 400001 to 400002 of the external device (MODBUS addresses 0000H to 0001H).
- After it is confirmed that connection 1 is established in master mode (X90) and no transmissions are currently being executed for the same port (Y90), the SEND instruction is started.
- The UNITSEL instruction is used to specify a slot number (LAN port: U100) and the connection number (U1).
- In the SEND instruction, specify and execute PLC's starting address (DT100) and data amount (U2), MODBUS command (16 = H10), and partner station number (H01) and starting address (H0). Check addresses of connected devices in the instruction manuals of devices.



■ Timing chart



■ I/O allocations

I/O number	Name	Description
X90 to X9F	Master communication clear to send flag	Turns ON when a connection is established in the master communication.
Y90 to Y9F	Master communication sending flag	Turns ON during sending data based on the SEND/RECV instruction. Turns OFF when the ED instruction is executed after the completion of the response receive processing.
Y70 to Y7F	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

■ Precautions during programming (in the case of CPU with built-in ET-LAN)

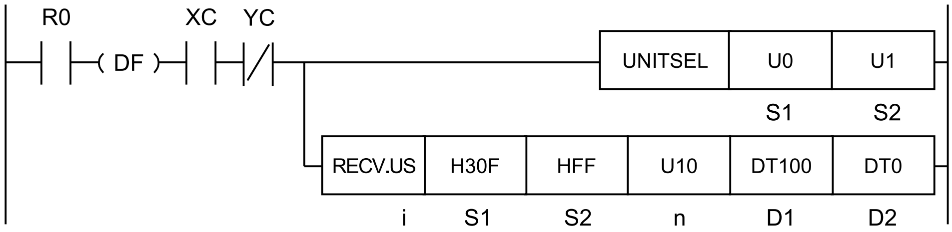
- Use the UNITSEL instruction immediately before the SEND/RECV instruction to specify a target connection number for communication.
- Master communication is only enabled when MEWTOCOL or MODBUS is selected. Confirm that the "master communication clear to send flags" (X90 to X9F) are ON for the corresponding connection, and execute the SEND/RECV instruction.
- The SEND or RECV instruction cannot be executed for the connection for which the master communication is in progress. Confirm that the "master communication sending flags" (Y90 to Y9F) are OFF, and execute the instruction.
- The SEND or RECV instruction cannot be executed for the connection for which the slave communication is in progress.
- Up to 16 SEND/RECV instructions can be executed simultaneously for differing connections.
- In the MODBUS-TCP mode, specify the partner station number as operand for the SEND/ RECV instruction.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the source range is outside the accessible range.
	COM port or connection specified by UNITSEL does not exist, or communication is not possible in the specified connection.
	Data device specified by [S] is invalid, or exceeds the area.
	Sent data amount specified by [n] is invalid.
	MODBUS command and/or station number specified by [D1] is invalid.
	Data device specified by [D2] is invalid, or exceeds the area.
	Result storage device specified by [D3] is invalid.
	Integer specification for [D2] is only available for the MODBUS address direct specification type. It is invalid for other types.
	Result storage device specified by [D3] is invalid.

15.7 RECV (MODBUS Master: Function Code Specification)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U0 (CPU unit with built-in SCU) and S2=U1 (port number 1) are specified by the UNITSEL instruction.

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Items	Settings			Setting range
S1	MODBUS function code to be used, and the partner station number			
		Higher byte	MODBUS function code (two hexadecimal digits)	H1 to H4 (1 to 4)
		Lower byte	Partner station number (two hexadecimal digits)	H1 to HFF (1 to 255)
S2	MODBUS starting address of the sender in the partner unit			H0 to HFFFF (0 to 65535)
n	Number of received data			1 to 127 words 1 to 2040 bits
D1	Device starting address of the receiver data area in the master unit			-
D2	Starting address of the device area of the master unit that stores the execution result code (1 word)			

■ Available word devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●								●	●				●
S2	●	●	●	●			●	●								●	(Note 1)				●
n	●	●	●	●			●	●								●	●				●

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D1	●	●	●	●			●	●													●
D2	●	●	●	●			●	●													●

(Note 1) Only in the case of "direct address specification" (main instruction) in the MODBUS mode, an integer can be specified for the sender address.

■ Available bit devices (●: Available)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
S1														
S2	●	●	●	●								●	●	●
n														
D1	●	●	●	●								●	●	●
D2														

■ Outline of operation

- The MODBUS command is sent from the communication port of the unit to send/receive data to/from external devices.
- Data can be read and written by specifying station numbers and memory addresses and executing the SEND and RECV instructions in a user program, because PLC automatically creates messages according to the protocol.
- Specify the MODBUS command and the partner MODBUS address in a Hex format in [S1].
- When the RECV instruction is executed, data are read from the address starting with [S2] in the partner unit, and stored in the area starting with [D1] in the master unit.
- The transfer method (register transfer/bit transfer) and the MODBUS function code that can be used vary, depending on the type of device specified by [D1].
- The amount of received data [n] is specified in words for the register transfer, and in the number of bits for the bit transfer.
- The execution result code is stored in the one-word area of the master unit that is specified by [D2].

■ Specification of [S1] and [n]

- Operand [S1] is specified as a combination of a two-digit hexadecimal MODBUS function code and a two-digit hexadecimal partner station number.

  Example: Specify "H030F" in the case of MODBUS function code 03 (read holding registers) and station number 15.
- The transfer method and the MODBUS function code that can be used vary, depending on the type of device specified by the operand [D1].

Device to be specified for [D1]	Transfer method	Value that can be specified for high bytes of [S1]
16-Bit device: WX, WY, WR, WL, DT, LD	Register transfer	H1: Read coil state (01) H2: Read input state (02) H3: Read hold register (03) H4: Read input register (04)
1-Bit device: X, Y, R, L, DT.n, LD.n	Bit transfer	H1: Read coil state (01) H2: Read input state (02)

- The amount of received data is specified in words for the register transfer, and in bits for the bit transfer.

■ Execution result code [D2]

Code	Description	Code	Description
H0	Normal end	H6	Reception error(Note 1)
H1	The communication port is being used in the master communication.	H7	I/O allocation shortage error(Note 2)
H2	The communication port is being used in the slave communication.	H8	The send buffer is being used.(Note 3)
H3	The number of master communication instructions simultaneously used is exceeded.	H8001	Function code error
H4	Transmission timeout	H8002	Device number error (out of range)
H5	Response reception timeout	H8003	Device quantity error (out of range)

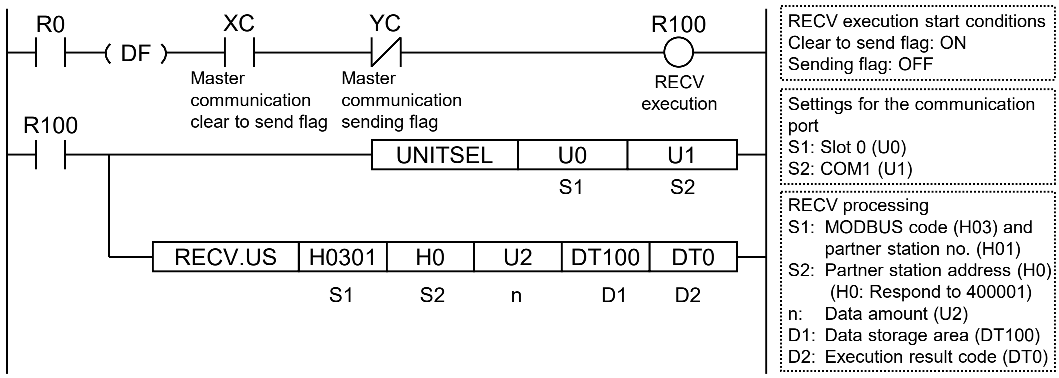
(Note 1) It occurs when an abnormal telegram is received. When there is a format error in the header of an individual protocol, the communication discards the received data and a response reception timeout occurs.

(Note 2) It occurs when the communication control I/O relays corresponding to the communication port (master communication clear to send flag, master communication send active flag, master communication send done result relay)are not allocated as I/O words of the CPU unit in the I/O map. It occurs only when the number of user connections of ET-LAN is expanded and this instruction is executed specifying the expanded connections.

(Note 3) Effective when the version is 4.57 or later.

■ Sample program (in the case of SCU)

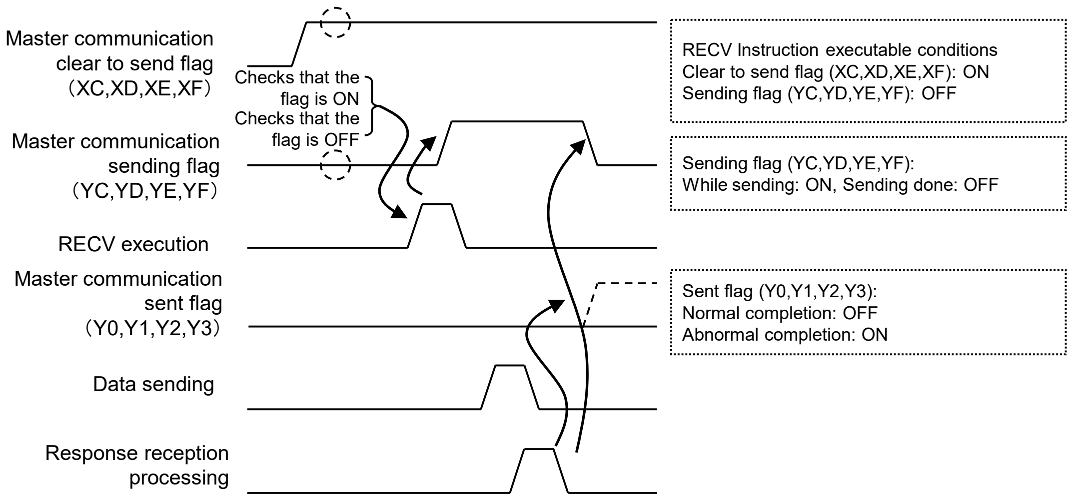
- This program sends the command from the COM1 port of the CPU unit, reads the data from the data areas 400001 to 400002 of the external device (station number 1), and writes the data into the data registers DT100 to DT101 of the PLC.
- This program checks that the master mode is on (XC) and that sending is not in progress in the same port (YC), and then starts up the RECV instruction.
- Using the UNITSEL instruction, specify the slot number (U0) and the COM. port number (U1).
- In the RECV instruction, specify and execute the partner station number (U1), MODBUS command to be used and partner station number (H0301), starting address (400001), data amount (U2), and PLC's starting address to store the data (DT100). Check addresses of connected devices in the instruction manuals of devices.



(Note 1) Operand [S1] of RECV instruction is specified by combining two hexadecimal digits of MODBUS function code with two hexadecimal digits of partner device station number.

(Note 2) When the partner device is FP series PLC, Operand [S2] of RECV instruction can be specified using the device number.

■ Time chart (in the case of SCU)



Info.

- The case of SCU shows the case that it is used in the following combination.
  - COM.0 port equipped in the CPU unit
  - Communication cassettes attached to the CPU unit (COM.1 to COM.2 ports)
  - Communication cassettes attached to the serial communication unit (COM.1 to COM.4 ports)
- The communication cassette (Ethernet type) does not support MODBUS.

■ I/O allocation (in the case of CPU with built-in SCU)

COM Port No.			Name	Description
1	2	0		
XC	XD	XE	Master communication clear to send flag	Turns ON when MEWTOCOL-COM, MEWTOCOL7, or MODBUS-RTU is set for the communication mode, and the unit is in the RUN mode.
YC	YD	YE	Master communication sending flag	Turns ON during sending data based on SEND/RECV instruction. Turns OFF when the sending process is completed.
Y0	Y1	Y2	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

■ I/O allocation (in the case of Serial Communication Unit)

COM Port No.				Name	Description
1	2	3	4		
XC	XD	XE	XF	Master communication clear to send flag	Turns ON when MEWTOCOL-COM, MEWTOCOL7, or MODBUS-RTU is set for the communication mode, and the unit is in the RUN mode.
YC	YD	YE	YF	Master communication sending flag	Turns ON during sending data based on SEND/RECV instruction. Turns OFF when the sending process is completed.
Y0	Y1	Y2	Y3	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

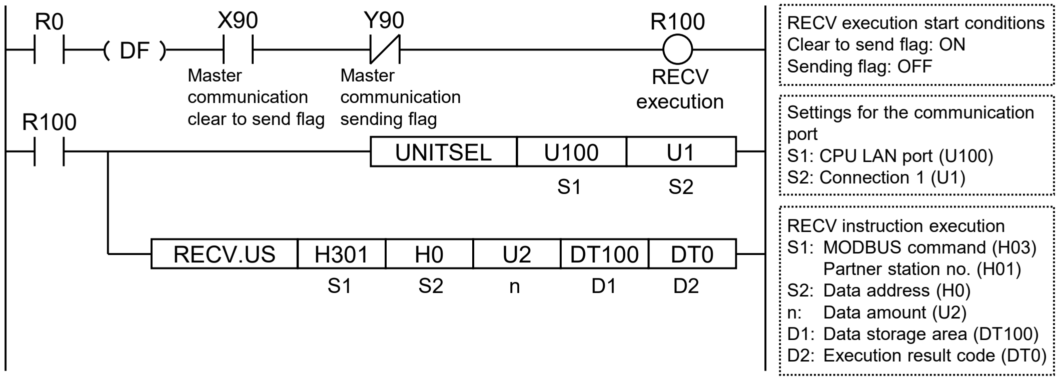
■ Precautions during programming (in the case of SCU)

- Use the UNITSEL instruction immediately before the SEND/RECV instruction to specify a target port for communication.
- Master communication is only enabled when MEWTOCOL or MODBUS is selected. Confirm that the "master communication clear to send flags" (XC to XF) are ON for the corresponding channel, and execute the SEND/RECV instruction.
- Another SEND/RECV instruction cannot be executed for a communication port where master communication is in progress. Confirm that the "master communication sending flags" (YC to YF) are OFF, and execute the instruction.
- A SEND/RECV instruction cannot be executed for a port where slave communication is in progress. If there is no response, the "master communication sending flags" (YC to YF) remain ON during the time-out period set in the CPU configuration.
- Up to 16 SEND/RECV instructions can be executed simultaneously for differing COM ports.

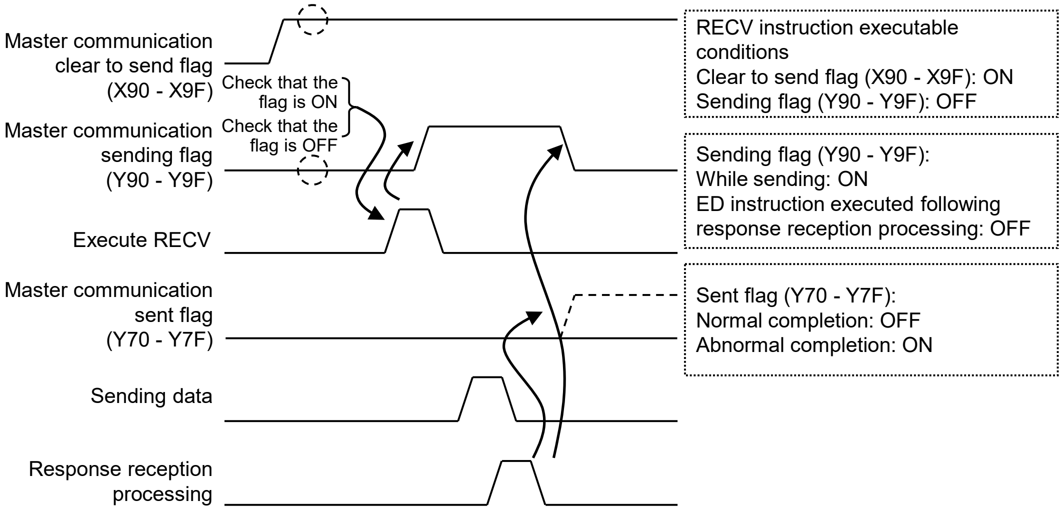
■ Sample program (in the case of CPU with built-in ET-LAN)

- This program sends MODBUS commands (03) from the LAN port of the CPU unit, reads the data from the data area of an external device 400001 to 400002 (MODBUS address 0000H to 0001H), and writes the content into PLC's data register DT100 to DT101.

- After it is confirmed that connection 1 is established in master mode (X90) and no transmissions are currently being executed for the same port (Y90), the RECV instruction is started.
- The UNITSEL instruction is used to specify a slot number (LAN port: U100) and the connection number (U1).
- In the RECV instruction, specify and execute the type of MODBUS command and partner station number (H0301), starting address (H0), data amount (U2), and PLC's starting address to store the data (DT100). Check addresses of connected devices in the instruction manuals of devices.



■ Time chart (in the case of CPU with built-in ET-LAN)



■ I/O allocation (in the case of CPU with built-in ET-LAN)

I/O number	Name	Description
X90 to X9F	Master communication clear to send flag	Turns ON when a connection is established in the master communication.
Y90 to Y9F	Master communication sending flag	Turns ON during sending data based on the SEND/RECV instruction. Turns OFF when the ED instruction is executed after the completion of the response receive processing.

I/O number	Name	Description
Y70 to Y7F	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

■ Precautions during programming (in the case of CPU with built-in ET-LAN)

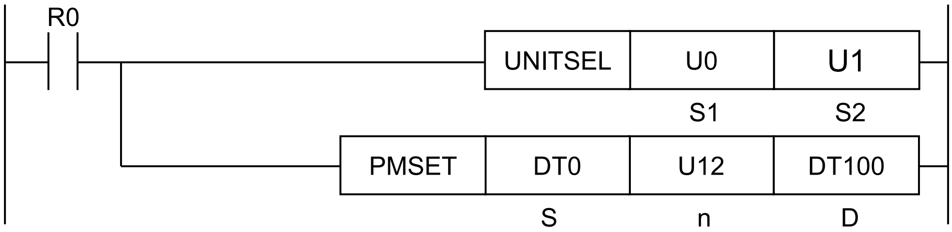
- Use the UNITSEL instruction immediately before the SEND/RECV instruction to specify a target connection number for communication.
- Master communication is only enabled when MEWTOCOL or MODBUS is selected. Confirm that the "master communication clear to send flags" (X90 to X9F) are ON for the corresponding connection, and execute the SEND/RECV instruction.
- The SEND or RECV instruction cannot be executed for the connection for which the master communication is in progress. Confirm that the "master communication sending flags" (Y90 to Y9F) are OFF, and execute the instruction.
- The SEND or RECV instruction cannot be executed for the connection for which the slave communication is in progress.
- Up to 16 SEND/RECV instructions can be executed simultaneously for differing connections.
- In the MODBUS-TCP mode, specify the partner station number as operand for the SEND/RECV instruction.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the destination range is outside the accessible range.
	COM port or connection specified by UNITSEL does not exist, or communication is not possible in the specified connection.
	Partner station number specified by [S1] is out of the range.
	Partner unit sender data device specified by [S2] is invalid.
	Sent data amount specified by [n] is invalid.
	Data device of the receiver data area in the master unit specified by [D1] is invalid, or exceeds the area.
	Result storage device specified by [D2] is invalid.
	Integer specification for [S2] is only available for the MODBUS address direct specification type. It is invalid for other types.

15.8 PMSET / pPMSET (Change of SCU Parameters)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U0 (CPU unit with built-in SCU) and S2=U1 (port number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Start of the area that stores data to be set as communication parameters
n	Specified number of words Other than PLC link mode (setting range: 1 to 12). PLC link mode (setting range: 1 to 26)
D	Starting address of the device area in the master unit that stores the processing result (1 word)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●													●
n	●	●	●	●			●	●								●					●
D	●	●	●	●			●	●													●

■ Outline of operation

- Communication parameters of the COM port of the unit is changed with a user program.
- Set communication parameters to be changed within [n] words from the area starting with [S], and execute the PMSET/pPMSET instruction, to issue the setting change request to the unit.
- While the requested change is being processed, bit 15 of the processing result storage area [D] turns ON. When the process is completed, it turns OFF.
- The processing result is stored in the area specified by [D]. If there is any abnormality, bit 14 of [D] turns ON. The error code is stored in low bytes of [D].
- By reading setting parameters using the PMGET instruction, and setting parameters to be changed using the PMSET/pPMSET instruction, the settings can be simplified.

Precautions for programming

- Describe UNITSEL instruction immediately before the PMSET/pPMSET instruction, and specify the slot and COM port numbers of the unit the parameters of which are changed.
- Checking of the processing result should be carried out when bit 15 (process in-progress flag) of the area specified by [D] is switched from 1 to 0.
- If parameter change is carried out for a COM port where sending/receiving is in progress, the sending/receiving process is canceled and parameters are changed. At this time, received data are lost. The sending process is suspended.
- The content set by PMSET/pPMSET instruction is not held in case of power outage. Turn on the power supply again and switch to RUN mode to return to the configuration information set in the tool software.
- It is not possible to change the mode from other than the PLC link mode to the PLC link mode.
- When changing the station number, make sure that the new number does not duplicate the number for any other device.

Comparison between the PMSET instruction and the pPMSET instruction

Instruction	Characteristics	
PMSET	The processing result is checked during the operation processing of the PMSET instruction.(Note 1)	
	Advantage	The PMSET instruction can be executed without any restrictions for different COM ports.
	Disadvantage	It is necessary to turn ON the execution condition of the PMSET instruction until the end of processing, and turn OFF the execution condition at a scan in which the end of data transmission is confirmed.
pPMSET (Note 1)	The processing result is checked at the end of a scan.(Note 1)	
	Advantage	The parameter change processing is performed only once by turning on the execution condition at the time of data send.
	Disadvantage	Up to 16 send operations can be performed simultaneously to different COM ports. (The total of simultaneous usage of SEND, RECV, pGPSEND, GPTRNS, and pPMSET instructions.)

(Note 1) Checking of the processing result means that the result of the parameter change processing is stored in [D].

Parameter settings

Operand	Parameter	Range	Settings
[S]	Communication Mode	U0 U1 U2 U8 U9	U0 : MEWTOCOL-COM U1: MEWTOCOL7-COM U2: MODBUS-RTU U8: General-purpose Communication U9: PLC link (only station number can be changed)
[S+1]	Station number setting	U1 to U999	Station number U1 to U999 MEWTOCOL-COM: U1 to U99 MEWTOCOL7-COM: U1 to U999 MODBUS-RTU: U1 to U247

Operand	Parameter	Range	Settings
			PLC link: U1 to U16 (Default: 0)
[S+2]	Baud rate setting	U0 to U10	U0: 300, U1: 600, U2: 1200, U3: 2400, U4: 4800, U5: 9600, U6: 19200, U7: 38400, U8: 57600, U9: 115200, U10: 230400 bps
[S+3]	Data length setting	U0, U1	U0: 7-bit length, U1: 8-bit length
[S+4]	Parity setting	U0 to U2	U0: No parity, U1: Odd parity, U2: Even parity, U3: Parity fixed to 0 (Note 3)
[S+5]	Stop bit length setting	U0, U1	U0: 1 bit, U1: 2 bits
[S+6]	RS/CS enabled or disabled (Note 1)	U0, U1	U0: Disable, U1: Enable
[S+7]	Send waiting time	U0 to U10000	U0: Immediate Effective time = Un x 0.01 ms (0 to 100 ms)
[S+8]	Start code STX	U0, U1	U0: Disable, U1: Enable
[S+9]	Terminator setting	U0 to U3	U0: cR; U1: cR+Lf; U2: Time; U3: ETX
[S+10]	Terminator judgement time	U0 to U10000	U0: For 32 bits Effective time = Un × 0.01 ms (Only enabled when the end setting is "Time".)
[S+11]	Modem initialization	U0 to U2	U0: Not initialize U1: Execute the first initialization only U2: Re-execute initialization at the time of setting. (Note 2)
[S+12]	Reserved area	U0	U0
[S+13]	Communication option	U0, U1	bit 0: Used to select whether to continue or stop the PLC link. 0 = The PLC link stops when a communication error occurs. 1 = The PLC link continues even when a communication error occurs. bit 1 to 15: Reserved

(Note 1) RS/CS can be selected only when a 1-channel, 5-wire communication cassette (product number AFP7CCS2) is used.

(Note 2) The modem is initialized when the power is on, PMSET/pPMSET instruction is executed, and the RUN mode is turned ON. Initialization is executed only in the first session. (Excluding when the power supply is turned off and then on again.)

(Note 3) U3 can be specified only for a built-in SCU. In addition, parity bit is fixed to 0 when data is sent, and the parity bit is not checked when data is received.

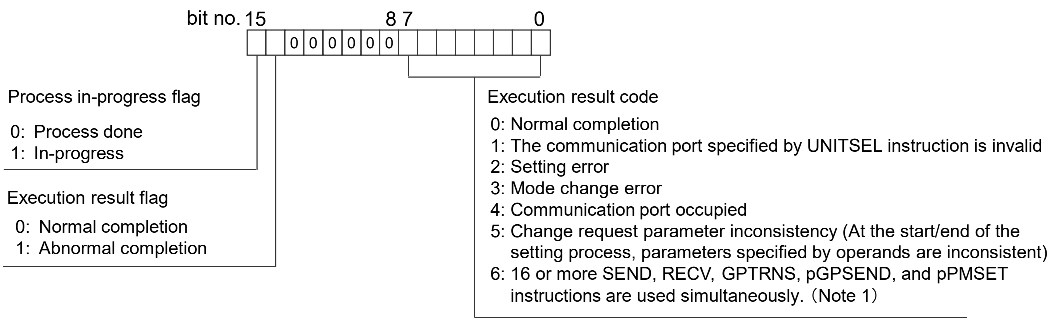
PLC link W0 setting (Setting is enabled when the communication mode is set to the PLC link mode; only COM1 is enabled)

Operand	Parameter	Range	Settings
[S+14]	Link area block number	U0, U1	Block number of link relay/link register area
[S+15]	PLC link W0 maximum station number	U2 to U16	PLC link W0 maximum station number
[S+16]	Range of link relays	U0 to U64	Specify the range of link relays used for communication (specification by number of words, relative values within the specified block)

Operand	Parameter	Range	Settings
[S+17]	Range of link registers	U0 to U128	Specify the range of link registers used for communication (relative values within the specified block)
[S+18]	Link relay transmission start number	U0 to U63	Link relay sending start number (specification by number of words, relative value within the specified block)
[S+19]	Link relay transmission size	U0 to U64	Size of link relay send area (specification by number of words)
[S+20]	Link register transmission start number	U0 to U127	Link register sending start number (specification by number of words, relative value within the specified block)
[S+21]	Link register transmission size	U0 to U127	Size of link register send area (specification by number of words)
[S+22]	Reserved area	U0	-
[S+23]	Reserved area	U0	-
[S+24]	Reserved area	U0	-
[S+25]	Reserved area	U0	-

■ Content of the processing result [D]

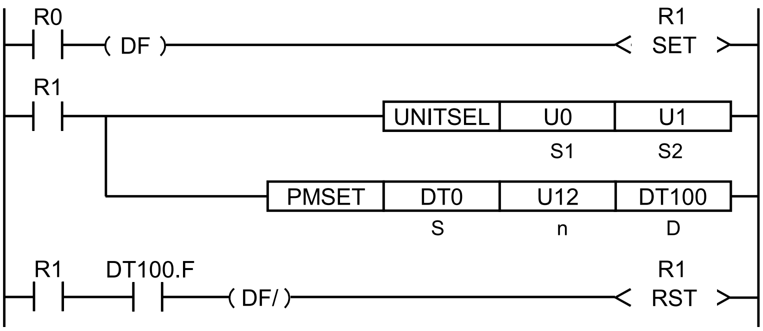
- Execution results are stored in the area of one word.
- The execution result code in the low byte is valid when the process in-progress flag of bit 15 is zero.



(Note 1) Execution result code "6" is enabled for the pPMSET instruction.

■ Program example 1 (PMSET instruction)

- This program sets 12 words of communication parameters for COM1 port of CPU with built-in SCU, which are stored in the area starting with Data Register DT0. The processing result is stored in DT100.
- For the PMSET instruction, it is necessary to turn ON the execution condition of the PMSET instruction until the end of processing, and turn OFF the execution condition at a scan in which the end of data transmission is confirmed.

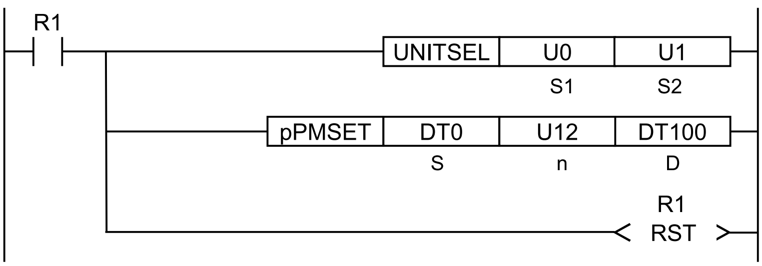


DT0	U 0	Communication mode
DT1	U 1	Unit no. setting
DT2	U 5	Baud rate setting
DT3	U 1	Data length setting
DT4	U 1	Parity setting
DT5	U 0	Stop bit length setting
DT6	U 0	RS/CS valid or invalid
DT7	U 0	Send waiting time
DT8	U 0	Header STX
DT9	U 0	Terminator setting
DT10	U 0	Terminator judgment time
DT11	U 0	Modem initialization

	Higher byte	Lower byte	
DT100	H 0	H 0	Processing result ('0' for normal completion)

■ Program example 2 (pPMSET instruction)

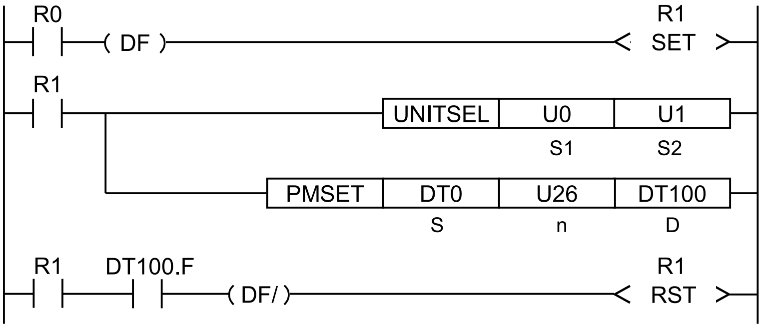
- This program sets 12 words of communication parameters for COM1 port of CPU with built-in SCU, which are stored in the area starting with Data Register DT0. The processing result is stored in DT100.
- For the pPMSET instruction, when the execution condition arises, the parameter change processing is performed only once.



■ Program example 3 (PMSET instruction, specification by PLC link parameter)

- This program sets 26 words of communication parameters for COM1 port of CPU with built-in SCU, which are stored in the area starting with Data Register DT0. The processing result is stored in DT100.

- For the PMSET instruction, it is necessary to turn ON the execution condition of the PMSET instruction until the end of processing, and turn OFF the execution condition at a scan in which the end of data transmission is confirmed.



DT0	U 9	Communication mode
DT1	U 1	Station number setting
DT2	U 5	Baud rate setting
DT3	U 1	Data length setting
DT4	U 1	Parity setting
DT5	U 0	Stop bit length setting
DT6	U 0	RS/CS enabled or disabled
DT7	U 0	Send waiting time
DT8	U 0	Start code STX
DT9	U 0	Terminator setting
DT10	U 0	Terminator judgement time
DT11	U 0	Modem initialization
DT12	U 0	Reserved area
DT13	U 0	Communication option
DT14	U 0	Link area block number
DT15	U 16	PLC link W0 maximum station number
DT16	U 32	Range of link relays
DT17	U 64	Range of link registers
DT18	U 0	Link relay sending start number
DT19	U 19	Size of link relay send area
DT20	U 0	Link register sending start number
DT21	U 32	Size of link register send area
DT22	U 0	Reserved area
DT23	U 0	Reserved area
DT24	U 0	Reserved area
DT25	U 0	Reserved area

	High byte	Low byte	
DT100	H 0	H 0	Processing result ('0' for normal completion)

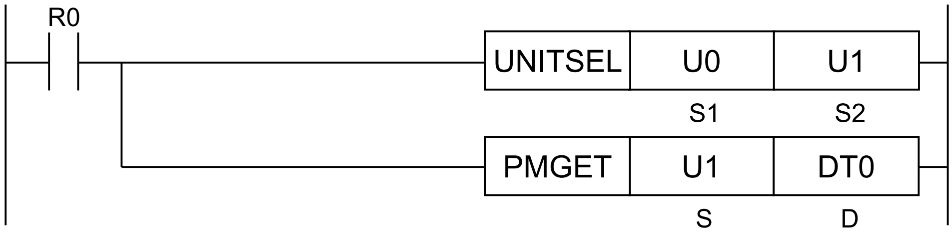
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the COM port specified by UNITSEL does not exist.

Name	Description
	To be set when the device specified for [S] to set the parameters is invalid.
	To be set when the number of words specified by [n] is out of the available range.
	To be set when the device specified for [D] to store the processing result is invalid.

15.9 PMGET (Acquiring SCU Parameters)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U0 (CPU unit with built-in SCU) and S2=U1 (port number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Type of acquired data 0: Communication parameters; 1: Communication monitoring area information; 2: PLC link status flag information; 3: PLC link error frequency information; 4: PLC link time interval information; 5: PLC link settings parameter monitoring information
D	Starting address of the area that stores the acquired communication parameter (monitor information)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●								●					●
D	●	●	●	●			●	●													●

■ Outline of operation

- This instruction reads the parameters of the COM port of the unit, and stores them in the area starting with [D].
- Specify the type of acquired data in [S].

■ Precautions for programming

- Describe the UNITSEL instruction immediately before the PMGET instruction, and specify the slot and COM port numbers of the unit for which the parameters are acquired.
- The size of the area storing data varies in the range of 7 to 192 words according to the data type specified in [S].

[S] value	Type of acquired data	Number of stored data
0	Communication parameter	26 word
1	Communication monitoring area information	7 word
2	PLC link status flag information	8 word
3	PLC link error frequency information	9 word
4	PLC link time interval information	8 word
5	PLC link settings parameter monitoring information	192 word

■ Acquired data (communication parameters): When [S] = 0

Storage location	Items	Range:	Description
[D]	Communication Mode	H0 U1 U2 U8 U9	U0: MEWTOCOL-COM U1: MEWTOCOL7-COM U2: MODBUS-RTU U8: General-purpose Communication U9: PLC link
[D+1]	Station number setting	U1 to U999	Station number U1 to U999 MEWTOCOL-COM: U1 to U99 MEWTOCOL7-COM: U1 to U999 MODBUS-RTU: U1 to U247 PLC link: U1 to U16 (Default: 0)
[D+2]	Baud rate setting	U0 to U10	U0: 300, U1: 600, U2: 1200, U3: 2400, U4: 4800, U5: 9600, U6: 19200, U7: 38400, U8: 57600, U9: 115200, U10: 230400 bps
[D+3]	Data length setting	U0, U1	U0: 7-bit length, U1: 8-bit length
[D+4]	Parity setting	U0 to U2	U0: No parity, U1: Odd parity, U2: Even parity, U3: Parity fixed to 0
[D+5]	Stop bit length setting	U0, U1	U0: 1 bit, U1: 2 bits
[D+6]	RS/CS enabled or disabled (Note 1)	U0, U1	U0: Disable, U1: Enable
[D+7]	Send waiting time	U0 to 10000	U0: Immediate Effective time = Un x 0.01 ms (0 to 100 ms)
[D+8]	Start code STX	U0, U1	U0: Disable, U1: Enable
[D+9]	Terminator setting	U0 to U3	U0: CR, U1: CR+LF, U2: Time, U3: ETX
[D+10]	Terminator judgement time	U0 to U10000	U0: 32 bits Effective time = Un × 0.01 ms (Only enabled when the end setting is "Time".)
[D+11]	Modem initialization	U0 to U2	U0: Not initialize U1: Execute the first initialization only (Note 2) U2: Re-execute initialization at the time of setting.
[D+12] to [D+13]	Reserved for system	H0	Reserved area for system
[D+14]	Link area block number	U0, U1	Block number of link relay/link register area

Storage location	Items	Range:	Description
[D+15]	PLC link W0 maximum station number	U2 to U16	Values out of the range are handled as "16".
[D+16]	Range of link relays	U0 to U64	Specify the range of link relays used for communication (specification by number of words, relative values within the specified block)
[D+17]	Range of link registers	U0 to U128	Specify the range of link registers used for communication (relative values within the specified block)
[D+18]	Link relay sending start number	U0 to U63	Link relay sending start number (specification by number of words, relative value within the specified block)
[D+19]	Size of link relay send area	U0 to U64	Size of link relay send area (specification by number of words)
[D+20]	Link register sending start number	U0 to U127	Link register sending start number (specification by number of words, relative value within the specified block)
[D+21]	Size of link register send area	U0 to U127	Size of link register send area (specification by number of words)
[D+22] to [D+25]	Reserved for system	H0	Reserved area for system

(Note 1) RS/CS can be selected only when an RS-232C (5-wire) x 1-channel communication cassette (product number AFP7CCS2) is used.

(Note 2) The modem is initialized at the time of setting (when the power is on, PMGET instruction is executed, or switching to the RUN mode). Initialization is executed only in the first session. (Excluding when the power supply is turned off and then on again.)

(Note 3) Settings of [D+14] to [D+21] are only enabled when the communication mode of the COM1 port is PLC link.

■ Acquired data (communication monitoring area information): When [S] = 1

Storage location	Items	Range:	Description
[D]	Operation mode	U 0 U 1 U 2 U 8 U 9 HFFFF	U0: MEWTOCOL-COM U1: MEWTOCOL7-COM U2: MODBUS-RTU U8: General-purpose Communication U9: PLC link HFFFF: Modem initialization
[D+1]	Communication cassette detection	H0 U232 U422 U485	U0: No communication cassette U 232: RS-232C U 422: RS-422 U 485: RS-485
[D+2]	Reception error code	-	The bits corresponding to when an error occurs turn ON. (0: Normal, 1: Error) Bit 0: Receive buffer overrun Bit 1: Stop bit undetected (frame error) Bit 2: Parity inconsistency Bit 7 to 3: 0 (fixed) Bit 8: Receive buffer overflow Bit 9: Receive buffer FULL

Storage location	Items	Range:	Description
			Bit 15 to 10: 0 (fixed)
[D+3]	Number of occurrences of reception error	U0 to U65535	The number of times reception errors were detected is stored.
[D+4]	Setting error code	-	The bits corresponding to when an error occurs turn ON. (0: Normal, 1: Error) Bit 0: Mode setting/change abnormality (A mode number that cannot be set or changed is specified) Bit 7 to 1: 0 (fixed) Bit 8: Communication parameters setting error Bit 9: Sent data amount abnormality Bit 15 to 10: 0 (fixed)
[D+5]	Error parameter number	U1 to U12	Parameter number which data outside the range is specified (Effective only when the communication parameter setting error occurs.)
[D+6]	Modem initialization status	H0000 H0100 H0200 H02FF	No operation During initialization Initialization succeeded (When modem initialization is completed, the operation mode automatically returns to its original state.) Initialization failed (When modem initialization is completed, the operation mode automatically returns to its original state.)

■ Acquired data (PLC link status flag information): When [S] = 2

Storage location	Items	Range:	Description
[D]	Master unit number	U1 to U999	U1 to U999 MEWTOCOL-COM: U1 to U99 MEWTOCOL7-COM: U1 to U999 MODBUS-RTU: U1 to U247 PLC link: U1 to U16 (Default: 0)
[D+1]	Error flag 1	H0000 to H00DF	Bit 0: Transmission error (0 = Normal, 1 = Transmission error) Bit 1: Station number overlap (0 = Normal, 1 = Error) Bit 2: Area overlap (0 = Normal, 1 = Error) Bit 3: Link error (0 = Normal, 1 = Error) Bit 4: Maximum station number inconsistency (0 = Normal, 1 = Error) Bit 5: Fixed to 0 Bit 6: Link stop (0 = Operating, 1 = Stop status) Bit 7: Initializing (0 = Regular, 1 = Initializing) Bit 15 to 8: Not used
[D+2]	Error flag 2	H0000 to H00FF	Bit 0: Lost token (0 = Normal, 1 = Error) Bit 1: Duplicate token (0 = Normal, 1 = Error) Bit 2: Lost signal (0 = Normal, 1 = Error) Bit 3: Reception of undefined command Bit 4: BCC error (0 = Normal, 1 = Error) Bit 5: Received data format error (0 = Normal, 1 = Error)

Storage location	Items	Range:	Description
			Bit 6: Transmission error (0 = Normal, 1 = Error) Bit 7: Procedural error (0 = Normal, 1 = Error) Bit 15 to 8: Not used
[D+3]	Transmission error flag	H0000 to H0003	Bit 0: Link 0 (0 = Normal, 1 = Error) Bit 1: Link 1 (0 = Normal, 1 = Error) Bit 15 to 2: Not used
[D+4]	Transmission assurance flag by station	H0000 to H00FF	Bit 0: Station number 1 (0: Stop, Error state, 1: Normal transmission state) : Bit 15: Station number 16 (0: Stop, Error state, 1: Normal transmission state)
[D+5]	Station PLCs Operation mode flag	H0000 to H00FF	Bit 0: Station number 1 (0: PROG mode, 1: RUN mode) : Bit 15: Station number 16 (0: PROG mode, 1: RUN mode)
[D+6]	Station PLCs Operation state flag	H0000 to H00FF	Bit 0: Station number 1 (0: Normal, 1: Error) : Bit 15: Station number 16 (0: Normal, 1: Error)
[D+7]	Area overlap flag	H0000 to H00FF	Bit 0: Station number 1 (0: Normal, 1: Area overlap occurs (Position of partner unit is ON.) : Bit 15: Station number 16 (0: Normal, 1: Area overlap occurs (Position of partner unit is ON.)

■ Acquired data (PLC link error frequency information): When [S] = 3

Storage location	Items	Range:	Description
[D]	Number of lost tokens	U0 to U65535	The number of occurrences of each error is stored.
[D+1]	Number of duplicate tokens		
[D+2]	Number of no signal states		
[D+3]	Number of times of receptions of undefined commands		
[D+4]	Number of sum check errors for reception		
[D+5]	Number of received data format errors		
[D+6]	Number of transmission errors		
[D+7]	Number of procedural errors		
[D+8]	Number of duplicate parent units		

■ Acquired data (PLC link time interval information): When [S] = 4

Storage location	Items	Range:	Description
[D]	RING counter for the number of receptions	U0 to U65535	The number of receptions is stored. When the value exceeds 65535, it returns to 0.
[D+1]	Current receive interval (x 1 ms)	-	The reception interval is stored.
[D+2]	Minimum receive interval (x 1 ms)	-	
[D+3]	Maximum receive interval (x 1 ms)	-	
[D+4]	RING counter for the number of transmissions	U0 to U65535	The number of transmissions is stored. When the value exceeds 65535, it returns to 0.
[D+5]	Current send interval (x 1 ms)	-	The send interval is stored.
[D+6]	Minimum send interval (x 1 ms)	-	
[D+7]	Maximum send interval (x 1 ms)	-	

■ Acquired data (PLC link setting parameter monitoring information): When [S] = 5

Storage location	Items	Range:	Description
[D]	Link area block number	U0, U1 HFFFF	The block number of the link relay/link register area is stored. U0, U1: Block number of the master unit HFFFF: Other units
[D+1]	PLC link W0 maximum station number	U2 to U16	Values out of the range are handled as "16".
[D+2]	Range of link relays	U0 to U64	Specify the range of link relays used for communication (specification by number of words, relative values within the specified block)
[D+3]	Range of link registers	U0 to U128	Specify the range of link registers used for communication (relative values within the specified block)
[D+4]	Link relay sending start number	U0 to U63	Link relay sending start number (specification by number of words, relative value within the specified block)
[D+5]	Size of link relay send area	U0 to U64	Size of link relay send area (specification by number of words)
[D+6]	Link register sending start number	U0 to U127	Link register sending start number (specification by number of words, relative value within the specified block)
[D+7]	Size of link register send area	U0 to U127	Size of link register send area (specification by number of words)
[D+8] to [D+11]	Reserved for system	-	Reserved area

(Note 1) The storage destinations shown above are for station number 1. 12 words are assigned to each station and are stored in a 192-word area in order starting from the information for station number 1. In addition, they occupy the 192-word area.

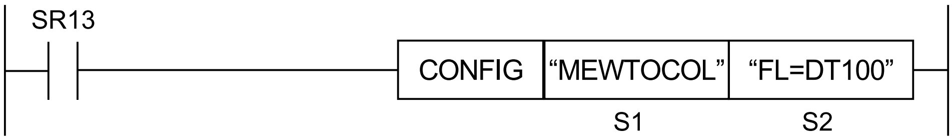
Items	PLC link station No. and storage destination				
	Station No. 1	Station No. 2	-----	Station No. 15	Station No. 16
Link area block number	[D]	[D+12]	-----	[D+168]	[D+180]
PLC link W0 maximum station number	[D+1]	[D+13]	-----	[D+169]	[D+181]
Range of link relays	[D+2]	[D+14]	-----	[D+170]	[D+182]
Range of link registers	[D+3]	[D+15]	-----	[D+171]	[D+183]
Link relay sending start number	[D+4]	[D+16]	-----	[D+172]	[D+184]
Size of link relay send area	[D+5]	[D+17]	-----	[D+173]	[D+185]
Link register sending start number	[D+6]	[D+18]	-----	[D+174]	[D+186]
Size of link register send area	[D+7]	[D+19]	-----	[D+175]	[D+187]
Reserved for system	[D+8] to [D +11]	[D+20] to [D +23]	-----	[D+176] to [D +179]	[D+188] to [D +191]

■ Flag operations

Name	Description
SR7 SR8 (ER)	Out-of-range in indirect access (index modification)
	Destination range is out of the accessible range.
	SCU unit does not exist in the slot specified by UNITSEL.
	COM port specified by UNITSEL does not exist.
	The parameter storage device specified by [D] is invalid.

15.10 CONFIG (Change Configuration)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Specify the character constant or the starting address of the device storing the string data that indicates the configuration type.
S2	Specify the character constant or the starting address of the device storing the string data that indicates the contents of change of the configuration.

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●			●	●												●	

■ Outline of operation

- When “MEWTOCOL” is specified for [S1]

  This instruction configures the settings so that reception is possible if the MEWTOCOL-COM(RD,WD) or MEWTOCOL-DAT(50H,51H) command is used to specify a device FL from another FP series PLC. This function is compatible with FP2SH.

  This instruction can be used for an FP7 CPU unit with built-in SCU, built-in ET-LAN, or Serial Communication Unit.

  Specify whether to assign data to FL0 or to the device number (DT/LD) of the FP7 CPU unit, if FL is specified by the partner device for MEWTOCOL-COM or MEWTOCOL-DAT communication.

- When “TCP-NODELAY” is specified for [S1]

  Possible to change the enable/disable setting for the TCP-NODELAY option between all system connections and user connection.

  While communication is performed with a device that is set with TCP delay ACK, high-speed communication becomes possible if the TCP-NODELAY option is enabled.

  The state of the enable/disable setting for the TCP-NODELAY option can be confirmed using the TCP-NODELAY option flag (X63).

  When the power is turned ON, the TCP-NODELAY option is disabled. After this instruction is executed, the TCP-NODELAY option is enabled.

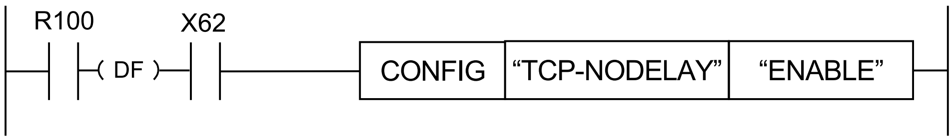
■ Flag operations

S1	S2	Settings
MEWTOCOL	FL=DTx	Assign FL0 to the address DTx.
	FL=LDx	Assign FL0 to the address LDx.
TCP-NODELAY	ENABLE	The TCP-NODELAY option is enabled.
	DISABLE	The TCP-NODELAY option is disabled.

■ Precautions for programming

- Set this instruction to be executed only one time after switching to RUN.
- When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- Both upper and lower case letters can be used for operands.
- Not possible to execute while the built-in Ethernet is initialized. Before executing this instruction, make sure that the IP address establishment flag (X62) is turned OFF.

■ Example of program (for enabling TCP-NODELAY option)



■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the device specified by [S1] exceeds the area.
	To be set when the device specified by [S2] exceeds the area.
	To be set when a keyword that is not expected is specified.
	To be set when the device corresponding to FL0 specified by [S2] exceeds the area. (Note 1)
	Set when the number of characters for operand specifying character constant exceeds 256.
	To be set for a device that does not have the Ethernet function. (Note 2)
SR9	Set when the instruction is the initialization of Ethernet. The detail code set for SD29 is “11: Ethernet initialization active”. (Note 2)

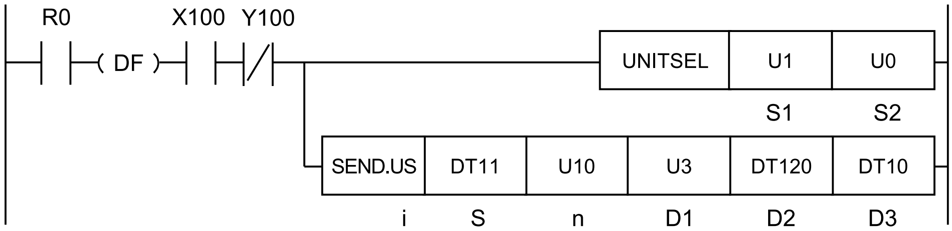
(Note 1) Only when “MEWTOCOL” is specified for [S1].

(Note 2) Only when “TCP-NODELAY” is specified for [S1].

16 High-level instructions (Multi-wire Link Communication)

16.1 SEND (When FP7 Multi-wire Link Unit Is Used)

■ Ladder diagram



(Note 1) The above figure shows the case that the FP7 multi-wire link unit for S1=U1 (slot number 1) is specified by the UNITSEL instruction.

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S	Starting address of the sender data area
n	Amount of sent data
D1	Partner station number
D2	Starting address of the device in the receiver data area of the partner unit
D3	Starting address of the device area of the master unit that stores the execution result code (1 word)

■ Available word devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●													●
n	●	●	●	●			●	●								●	●				●
D1	●	●	●	●			●	●								●	●				●
D2(Note 1)	●	●	●	●			●	●													●
D3	●	●	●	●			●	●													●

(Note 1) When the destination unit is FP7, only global devices can be specified. (Local devices cannot be specified.)

■ Available bit devices (●: Available)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
S	●	●	●	●								●	●	●
n														
D1														
D2 (Note 1)	●	●	●	●										●
D3														

(Note 1) When the receiver is FP7, only global devices can be specified. (A local device cannot be specified.)

■ Outline of operation

- Data can be transferred from the master unit to the partner unit between PLCs connected by MEWNET-W or MEWNET-W2.
- When the SEND instruction is executed, the data is read from the device that starts with [S] in the master unit and the data is stored in the address that starts with [D2] in the partner unit.

■ Setting the amount of sent data [n]

Transfer method	Communication Mode	Amount of sent data n
Register transfer	W mode	1 to 55 words
	W2 mode	1 to 1020 words
Bit transfer	W mode	Fixed at 1 bit
	W2 mode	Fixed at 1 bit

(Note 1) The transfer method varies according to the device type specified for operands [S] and [D2]. The register transfer is used for 16-bit devices and the bit transfer is used for 1-bit devices.

Devices specified for [S] and [D2]	Transfer method
16-bit device: WX, WY, WR, WL, DT, LD	Register transmission
1-bit device: X, Y, R, L, DT.n, LD.n	Bit transmission

(Note 2) The amount of sent data is specified in words for the register transfer and in bits for the bit transfer.

■ Specification of partner unit station number [D1]

Communication Mode	Setting range
W mode	1 to 32
W2 mode	1 to 64

■ Specification of destination address [D2]

Transfer method	Communication Mode	Address range
Register transfer	W mode	0 to 65535

Transfer method	Communication Mode	Address range
	W2 mode	
Bit transfer	W mode	0 to 65535F
	W2 mode	

(Note 1) When the receiver is the file register FL, specify a constant. Example) For FL100, specify U100. For the file register, only bank 0 can be specified.

■ Execution result code [D3]

Code	Description	Code	Description
H0	Normal end	H24	Transmission format error
H1	The communication port is being used in the master communication.	H25	FP7 multi-wire link unit hardware error
H2	The communication port is being used in the slave communication.	H26	The unit number setting error occurs.
H3	The number of master communication instructions simultaneously used is exceeded.	H27	NOT support
H4	Transmission timeout	H28	No response
H5	Response reception timeout	H29	FP7 multi-wire link unit hardware error
H6	Reception error(Note 1)	H30	Transmission time-out error
H7	I/O allocation shortage error(Note 2)	H31 to H39	FP7 multi-wire link unit hardware error
H8	The send buffer is being used.		
H9	Master unit station number unset error	H41	Format error
H21	NACK	H60	Parameter error
H22	WACK	H61	Data error
H23	The unit number duplicate error occurs.	H91	Missing expansion slave unit error

(Note 1) It occurs when an abnormal telegram is received. When there is a format error in the header of an individual protocol, the communication discards the received data and a response reception timeout occurs.

(Note 2) It occurs when the communication control I/O relays corresponding to the communication port (master communication clear to send flag, master communication send active flag, master communication send done result relay)are not allocated as I/O words of the CPU unit in the I/O map. It occurs only when the number of user connections of ET-LAN is expanded and this instruction is executed specifying the expanded connections.

■ Precautions for programming

- Describe the UNITSEL instruction immediately before the SEND instruction and specify the slot number of the FP7 multi-wire link unit in [S] and "0" in [S2].
- For FP7 multi-wire link unit, the SEND instruction executes the transmission and reception by MEWTOCOL-DAT (fixed).
- Up to 16 send instructions can be performed to different COM ports and connections simultaneously. (The total of simultaneous usage of SEND, RECV, pPSEND, GPTRNS, and pPMSET instructions)

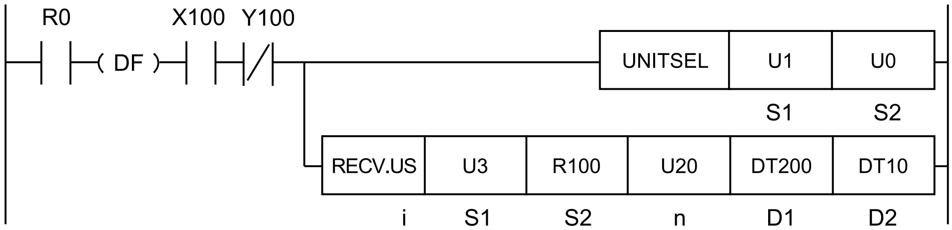
- For FP7 multi-wire link unit, only one of those instructions can be executed for one unit at a time.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the source range is outside the accessible range.
	To be set when the FP7 multi-wire link unit that is specified by UNITSEL does not exist.
	To be set when the data device specified by [S] is incorrect or exceeds the area.
	To be set when the amount of sent data specified by [n] is incorrect.
	To be set when the station number that is specified by [D1] is out of the range.
	To be set when the data device specified by [D2] is incorrect or exceeds the area.
	To be set when the result storing device that is specified by [D3] is incorrect.
	To be set when the type (1-bit or 16-bit) of the specified device is different between [S] and [D2].

16.2 RECV (When FP7 Multi-wire Link Unit Is Used)

■ Ladder diagram



(Note 1) The above figure shows the case that the FP7 multi-wire link unit for S1=U1 (slot number 1) is specified by the UNITSEL instruction.

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S1	Partner station number
S2	Starting address of the device in the sender data area of the partner unit
n	Number of received data
D1	Starting address of the device in the receiver data area of the master unit
D2	Starting address of the device area of the master unit that stores the execution result code (1 word)

■ Available word devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●								●	●				●
S2(Note 1)	●	●	●	●			●	●													●
n	●	●	●	●			●	●								●	●				●
D1	●	●	●	●			●	●													●
D2	●	●	●	●			●	●													●

(Note 1) When the source unit is FP7, only global devices can be specified. (Local devices cannot be specified.)

■ Available bit devices (●: Available)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
S1														
S2(Note 1)	●	●	●	●								●	●	●
n														
D1	●	●	●	●								●	●	●
D2														

(Note 1) When the sender is FP7, only global devices can be specified. (A local device cannot be specified.)

■ Outline of operation

- Data can be transferred from the partner unit to the master unit between PLCs connected by MEWNET-W or MEWNET-W2.
- When the RECV instruction is executed, the data is read from the address that starts with [S2] in the partner station number [S1] and the data is stored in the area that starts with [D1] in the master unit.

■ Specification of partner unit station number [S1]

Communication Mode	Setting range
W mode	1 to 32
W2 mode	1 to 64

■ Specify the starting address [S2] of the sender data area

Transfer method	Communication Mode	Address range
Register transfer	W mode	0 to 65535
	W2 mode	
Bit transfer	W mode	0 to 65535F
	W2 mode	

(Note 1) The transfer method varies according to the device type specified for operands [S2] and [D1]. The register transfer is used for 16-bit devices and the bit transfer is used for 1-bit devices.

Device specified in [S2] and [D1]	Transfer method
16-bit device: WX, WY, WR, WL, DT, LD	Register transmission
1-bit device: X, Y, R, L, DT.n, LD.n	Bit transmission

(Note 2) The bit devices DT, n, LD and n cannot be specified for the starting address of the source data of partner unit.

(Note 3) When the receiver is the file register FL, specify a constant. Example) For FL100, specify U100. For the file register, only bank 0 can be specified.

■ Setting the amount of received data [n]

Transfer method	Communication Mode	Amount of sent data n
Register transfer	W mode	1 to 56 words
	W2 mode	1 to 1020 words
Bit transfer	W mode	Fixed at 1 bit
	W2 mode	

(Note 1) The amount of sent data is specified in words for the register transfer and in bits for the bit transfer.

■ Execution result code [D2]

Code	Description	Code	Description
H0	Normal end	H24	Transmission format error
H1	The communication port is being used in the master communication.	H25	FP7 multi-wire link unit hardware error
H2	The communication port is being used in the slave communication.	H26	The unit number setting error occurs.
H3	The number of master communication instructions simultaneously used is exceeded.	H27	NOT support
H4	Transmission timeout	H28	No response
H5	Response reception timeout	H29	FP7 multi-wire link unit hardware error
H6	Reception error(Note 1)	H30	Transmission time-out error
H7	I/O allocation shortage error(Note 2)	H31 to H39	FP7 multi-wire link unit hardware error
H8	The send buffer is being used.		
H9	Master unit station number unset error	H41	Format error
H21	NACK	H60	Parameter error
H22	WACK	H61	Data error
H23	The unit number duplicate error occurs.	H91	Missing expansion slave unit error

(Note 1) It occurs when an abnormal telegram is received. When there is a format error in the header of an individual protocol, the communication discards the received data and a response reception timeout occurs.

(Note 2) It occurs when the communication control I/O relays corresponding to the communication port (master communication clear to send flag, master communication send active flag, master communication send done result relay)are not allocated as I/O words of the CPU unit in the I/O map. It occurs only when the number of user connections of ET-LAN is expanded and this instruction is executed specifying the expanded connections.

■ Precautions for programming

- Describe the UNITSEL instruction immediately before the RECV instruction and specify the slot number of the FP7 multi-wire link unit in [S1] and "0" in [S2].
- For FP7 multi-wire link unit, the RECV instruction executes the transmission and reception by MEWTOCOL-DAT (fixed).

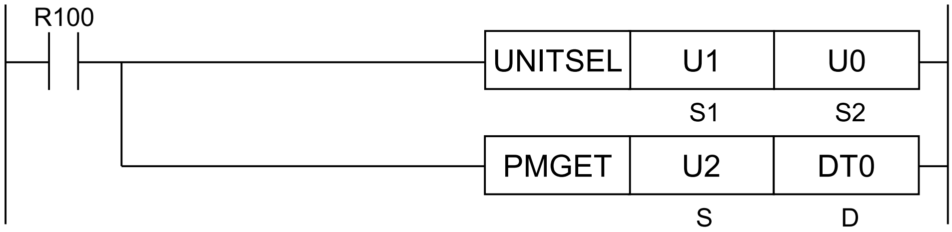
- Up to 16 send instructions can be performed to different COM ports and connections simultaneously. (The total of simultaneous usage of SEND, RECV, pPSEND, GPTRNS, and pPMSET instructions)
- For FP7 multi-wire link unit, only one of those instructions can be executed for one unit at a time.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the destination range is outside the accessible range.
	To be set when the FP7 multi-wire link unit that is specified by UNITSEL does not exist.
	To be set when the partner unit station number that is specified by [S1] is out of the range.
	To be set when the source data device of the partner unit specified by [S2] is incorrect.
	To be set when the amount of sent data specified by [n] is incorrect.
	To be set when the data device in the receiver data area in the master unit specified by [D1] is incorrect or exceeds the area.
	To be set when the result storing device that is specified by [D2] is incorrect.
	To be set when the type (1-bit or 16-bit) of the specified device is different between [S2] and [D1].

16.3 PMGET (Acquiring MEWNET-W Parameters)

■ Ladder diagram



(Note 1) The above figure shows the case that the FP7 multi-wire link unit for S1=U1 (slot number 1) is specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Type of acquired data 0: PLC link communication state, 1: Network participation state, 2: W link communication error information, 3: PLC link refresh operation monitoring information
D	Starting address of the area that stores the acquired communication parameter (monitor information)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●								●					●
D	●	●	●	●			●	●													●

■ Outline of operation

- Monitor information showing the communication state and PLC link operation can be acquired.
- The MEWNET-W communication parameters of the FP7 multi-wire link unit are read and stored in the area that starts with [D].
- Specify the type of acquired data in [S].

■ Precautions for programming

- Describe the UNITSEL instruction immediately before the PMGET instruction. Specify the slot number of the unit from which the parameters are acquired for [S1] and specify 0 for [S2].
- The size of the area storing data varies in the range of 3 to 15 words according to the data type specified in [S].

[S] value	Type of acquired data	Number of stored data	Storage location
0	PLC link communication state	3 word	[D] to [D+2]
1	Network participation state	3 word	[D] to [D+2]
2	W link communication error information	15 word	[D] to [D+14]
3	PLC link refresh operation monitoring information	8 word	[D] to [D+7]

■ Acquired data (PLC link communication state): When [S] = 0

Storage location	Items	Range:	Description
[D]	PLC link address duplicate station number	H0 to HF	Bit 0 to bit 15: Unit number 1 to unit number 16 OFF: Normal ON: Area duplication occurs (The position of the partner unit is set to ON)
[D+1]	PLC link transmission assurance relays		Bit 0 to bit 15: Unit number 1 to unit number 16 OFF: When stopped or in abnormal state ON: PLC link communicating normally
[D+2]	PLC link operation mode relay		Bit 0 to bit 15: Unit number 1 to unit number 16 OFF: PROG mode ON: RUN mode

■ Acquired data (network participation state): When [S] = 1

Storage location	Items	Range:	Description
[D]	Number of units added to the link	U0 to U32	The number of units added to the link is stored. The value is 0 when the existing unit in the network is only one or a station number is being changed.
[D+1]	Link participation unit flag Unit nos. 1 to 16	H0 to HF	Bit 0 to bit 15: Unit number 1 to unit number 16 OFF: Not exist ON: Participating
[D+2]	Link participation unit flag Unit nos. 17 to 32		Bit 0 to bit 15: Unit number 17 to unit number 32 OFF: Not exist ON: Participating

■ Acquired data (W link communication error information): When [S] = 2

Storage location	Items	Range:	Description
[D]	Number of occurrences of non-token state	U0 to U255	The number of occurrences of each error is stored.
[D+1]	Number of occurrences of duplicate tokens		
[D+2]	Number of occurrences of non-signal state		

Storage location	Items	Range:	Description
[D+3]	Number of occurrences of synchronous abnormality		
[D+4]	Number of occurrences of transmission answer NACK		
[D+5]	Number of occurrences of three consecutive transmission answers NACK		
[D+6]	Number of occurrences of transmission answer WACK		
[D+7]	Number of occurrences of three consecutive transmission answers WACK		
[D+8]	Number of occurrences of non-response		
[D+9]	Number of occurrences of three consecutive non-response		
[D+10]	Number of occurrences of receive command code error		
[D+11]	Number of occurrences of received data CRC error		
[D+12]	Number of absences of received data end code		
[D+13]	Number of occurrences of received data format error		
[D+14]	Number of occurrences of received data NOT support error		

■ Acquired data (PLC link refresh operation monitoring information): When [S] = 3

Storage location	Items	Range:	Description
[D]	RING counter for the number of receptions	U0 to U65535	The number of receptions is stored. When the value exceeds 65535, it returns to 0.
[D+1]	Current receive interval (x 1 ms)	-	The reception interval is stored.
[D+2]	Minimum receive interval (x 1 ms)	-	
[D+3]	Maximum receive interval (x 1 ms)	-	
[D+4]	RING counter for the number of transmissions	U0 to U65535	The number of transmissions is stored. When the value exceeds 65535, it returns to 0.
[D+5]	Current send interval (x 1 ms)	-	The send interval is stored.

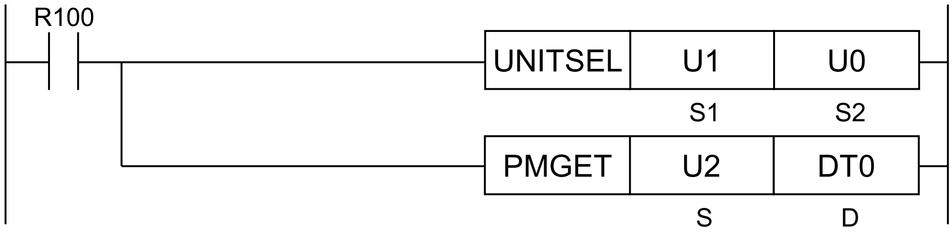
Storage location	Items	Range:	Description
[D+6]	Minimum send interval (x 1 ms)	-	
[D+7]	Maximum send interval (x 1 ms)	-	

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification, pointer access).
	To be set when the destination range is outside the accessible range.
	To be set when the FP7 multi-wire link unit does not exist in the slot that is specified by UNITSEL.
	To be set when the COM port specified by UNITSEL does not exist.
	To be set when the parameter storing device that is specified by [D] is invalid.

16.4 PMGET (Acquiring MEWNET-W2 Parameters)

■ Ladder diagram



(Note 1) The above figure shows the case that the FP7 multi-wire link unit for S1=U1 (slot number 1) is specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Type of acquired data 0: PLC link communication state, 1: Network participation state, 2: W2 link error system counter type error area, 3: W2 link error system error register area
D	Starting address of the area that stores the acquired communication parameter (monitor information)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●								●					●
D	●	●	●	●			●	●													●

■ Outline of operation

- Monitor information showing the communication state and PLC link operation can be acquired.
- The MEWNET-W2 communication parameters of the FP7 multi-wire link unit are read and stored in the area that starts with [D].
- Specify the type of acquired data in [S].

■ Precautions for programming

- Describe the UNITSEL instruction immediately before the PMGET instruction. Specify the slot number of the unit from which the parameters are acquired for [S1] and specify 0 for [S2].
- The size of the area storing data varies in the range of 5 to 18 words according to the data type specified in [S].

[S] value	Type of acquired data	Number of stored data	Storage location
0	PLC link communication state	6 word	[D] to [D+5]
1	Network participation state	5 word	[D] to [D+4]
2	W2 link error system counter type error area	18 word	[D] to [D+17]
3	W2 link error system error register area	10 word	[D] to [D+9]

■ Acquired data (PLC link communication state): When [S] = 0

Storage location	Items	Range:	Description
[D] to [D+1]	PLC link state monitor flag station number 1 to 32	H0 to HFFFFFFFF	Bit 0 to bit 31: Unit number 1 to unit number 32 OFF: Stopped ON: PLC link communicating normally
[D+2] to [D +3]	PLC link operation mode flag station number 1 to 32		Bit 0 to bit 31: Unit number 1 to unit number 32 OFF: PROG mode ON: RUN mode
[D+4] to [D +5]	PLC link operation state flag station number 1 to 32		Bit 0 to bit 31: Unit number 1 to unit number 32 OFF: No error ON: Error occurs in PLC for which transmission assurance is ON.

(Note 1) Even when (3) has been set for the PLC link operation state flag in the configuration "W2 link unit setting" of the tool software, the data of 6 words is read by the PMGET instruction.

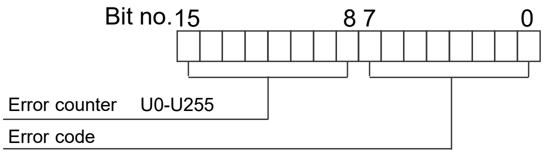
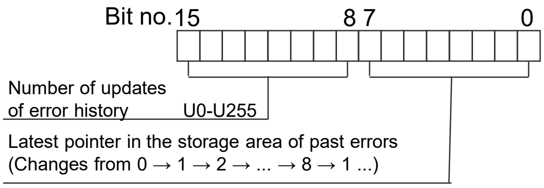
■ Acquired data (network participation state): When [S] = 1

Storage location	Items	Range:	Description
[D]	Number of units added to the link	U0 to U64	The number of units added to the link is stored. The value is 0 when the existing unit in the network is only one or a station number is being changed.
[D+1]	Link participation unit flag Unit nos. 1 to 16	H0 to HFFFF	Bit 0 to bit 15: Unit number 1 to unit number 16 OFF: Not exist ON: Participating
[D+2]	Link participation unit flag Unit nos. 17 to 32		Bit 0 to bit 15: Unit number 17 to unit number 32 OFF: Not exist ON: Participating
[D+3]	Link participation unit flag Unit nos. 33 to 48		Bit 0 to bit 15: Unit number 33 to unit number 48 OFF: Not exist ON: Participating
[D+4]	Link participation unit flag Unit nos. 49 to 64		Bit 0 to bit 15: Unit number 49 to unit number 64 OFF: Not exist ON: Participating

■ Acquired data (W2 link error system counter type error area): When [S] = 2

Storage location	Items	Range:	Description
[D]	Number of lost tokens	U0 to U255	The number of occurrences of each error is stored.
[D+1]	Number of duplicate tokens		
[D+2]	Number of occurrences of non-signal state		
[D+3]	Number of occurrences of synchronous abnormality		
[D+4]	Number of occurrences of send NACK error (When an error occurs)		
[D+5]	Number of occurrences of send NACK error (At the time of third retry)		
[D+6]	Number of occurrences of send WACK error (When an error occurs)		
[D+7]	Number of occurrences of send WACK error (When occurred 16 times continuously)		
[D+8]	Number of occurrences of non-response (When an error occurs)		
[D+9]	Number of occurrences of non-response (At the time of third retry)		
[D+10]	Number of receptions of undefined commands		
[D+11]	Number of occurrences of receive parity check error		
[D+12]	Number of occurrences of END CODE reception error		
[D+13]	Number of occurrences of received data format error		
[D+14]	Number of occurrences of received data NOT support error		
[D+15]	Number of token retransmissions		
[D+16]	Number of detection of unit OFF		
[D+17]	Number of occurrences of link error state		

■ Acquired data (W2 link error system error register area): When [S] = 3

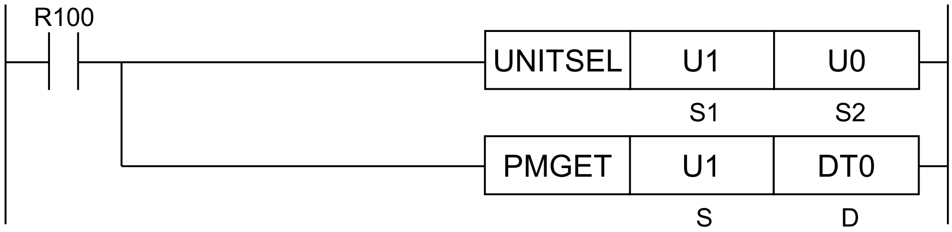
Storage location	Items	Description
[D]	State of error currently occurs	 When the same error occurs, the error counter of high byte will be updated. When the error content changes and the error is cleared, the information will be stored in the error occurrence state history area ([D+2] to [D+9]).
[D+1]	Error occurrence state history management	 (Changes from 0 → 1 → 2 → ... → 8 → 1...) When the latest pointer in the storage area of past errors is 0, it indicates that there is no error. When the number of error updates is 0, it indicates that there is no update.
[D+2]	Error occurrence state history area 1	History of error occurrence state (parameter of [D]) 1
[D+3]	Error occurrence state history area 2	History of error occurrence state (parameter of [D]) 2
[D+4]	Error occurrence state history area 3	History of error occurrence state (parameter of [D]) 3
[D+5]	Error occurrence state history area 4	History of error occurrence state (parameter of [D]) 4
[D+6]	Error occurrence state history area 5	History of error occurrence state (parameter of [D]) 5
[D+7]	Error occurrence state history area 6	History of error occurrence state (parameter of [D]) 6
[D+8]	Error occurrence state history area 7	History of error occurrence state (parameter of [D]) 7
[D+9]	Error occurrence state history area 8	History of error occurrence state (parameter of [D]) 8

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification, pointer access).
	To be set when the destination range is outside the accessible range.
	To be set when the FP7 multi-wire link unit does not exist in the slot that is specified by UNITSEL.
	To be set when the COM port specified by UNITSEL does not exist.
	To be set when the parameter storing device that is specified by [D] is invalid.

16.5 PMGET (Acquiring MEWNET-F Parameters)

■ Ladder diagram



(Note 1) The above figure shows the case that the FP7 multi-wire link unit for S1=U1 (slot number 1) is specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Type of acquired data 0: Number of F link services, 1: F link operation state monitor
D	Starting address of the area that stores the acquired communication parameter (monitor information)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●								●					●
D	●	●	●	●			●	●													●

■ Outline of operation

- Monitor information showing the communication state can be acquired.
- MEWNET-F communication parameters of the FP7 multi-wire link unit are read and stored in the area that starts with [D].
- Specify the type of acquired data in [S].

■ Precautions for programming

- Describe the UNITSEL instruction immediately before the PMGET instruction. Specify the slot number of the unit from which the parameters are acquired for [S1] and specify 0 for [S2].
- The size of the area storing data varies in the range of 1 to 10 words according to the data type specified in [S].

[S] value	Type of acquired data	Number of stored data	Storage location
0	Number of F link services	1 word	[D]
1	F link operation state monitor	10 word	[D] to [D+9]

■ Acquired data (number of F link services): When [S] = 0

Storage location	Items	Range:	Description
[D]	F link service counter	U0 to U65535	Service ring counter of master unit

■ Acquired data (F link operation state monitor): When [S] = 1

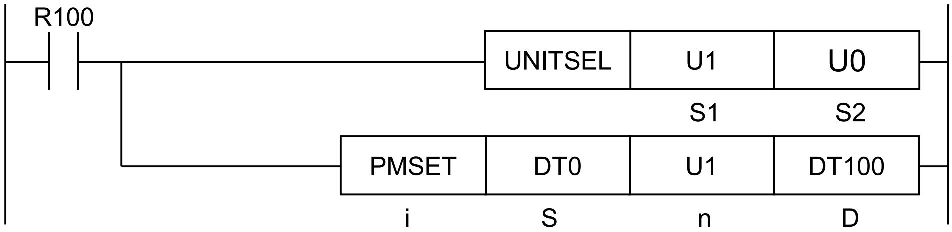
Storage location	Items	Range:	Description
[D] to [D+1]	Connected unit	H0 to HFFFFFFFF	Bit 0 to bit 31: Unit number 1 to unit number 32 OFF: Disconnected unit ON: Connected unit
[D+2] to [D +3]	Abnormal unit current value		Bit 0 to bit 31: Unit number 1 to unit number 32 OFF: Normal unit ON: Abnormal unit
[D+4] to [D +5]	Abnormal unit cumulative value		
[D+6] to [D +7]	Setting of slave unit where I/O verification error occurred		
[D+8] to [D +9]	Slave unit where instantaneous power failure occurred		

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification, pointer access).
	To be set when the destination range is outside the accessible range.
	To be set when the FP7 multi-wire link unit does not exist in the slot that is specified by UNITSEL.
	To be set when the COM port specified by UNITSEL does not exist.
	To be set when the parameter storing device that is specified by [D] is invalid.

16.6 PMSET / pPMSET (Change of MEWNET-W Parameters)

■ Ladder diagram



(Note 1) The above figure shows the case that the FP7 multi-wire link unit for S1=U1 (slot number 1) is specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Start of the area that stores data to be set as communication parameters
n	Specified number of words (Setting range: 10 or 1) (Note 1)
D	Starting address of the device area in the master unit that stores the processing result (1 word)

(Note 1) When the station number is 17 or more, specify 1.

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●													●
n	●	●	●	●			●	●								●					●
D	●	●	●	●			●	●													●

■ Outline of operation

- The MEWNET-W communication parameters of the FP7 multi-wire link unit are changed with a user program.
- Set communication parameters to be changed within [n] words from the area starting with [S], and execute the PMSET/pPMSET instruction, to issue the setting change request to the unit.
- By reading setting parameters using the PMGET instruction, and setting parameters to be changed using the PMSET/pPMSET instruction, the settings can be simplified.

■ Precautions for programming

- The station number of the FP7 multi-wire link unit can be set by the PMSET/pPMSET instruction only when the station number selector on the unit is set to 0.
- Describe the UNITSEL instruction immediately before the PMSET/pPMSET instruction. Specify the slot number of the unit from which the parameters are acquired for [S1] and specify 0 for [S2].
- Checking of the processing result should be carried out when bit 15 (process in-progress flag) of the area specified by [D] is switched from 1 to 0.
- The content set by the PMSET/pPMSET instruction is not held in the case of power outage. Turn on the power supply again and switch to RUN mode to return to the configuration information set in the tool software.
- When setting it for the FP7 multi-wire link unit, it cannot be used in an interrupt program.

■ Comparison between the PMSET instruction and the pPMSET instruction

Instruction	Characteristics	
PMSET	The processing result is checked during the operation processing of the PMSET instruction.(Note 1)	
	Advantage	The PMSET instruction can be executed without any restrictions for different COM ports.
	Disadvantage	It is necessary to turn ON the execution condition of the PMSET instruction until the end of processing, and turn OFF the execution condition at a scan in which the end of data transmission is confirmed.
pPMSET (Note 1)	The processing result is checked at the end of a scan.(Note 1)	
	Advantage	The parameter change processing is performed only once by turning on the execution condition at the time of data send.
	Disadvantage	Up to 16 send operations can be performed simultaneously to different COM ports. (The total of simultaneous usage of SEND, RECV, pGPSEND, GPTRNS, and pPMSET instructions.)

(Note 1) Checking of the processing result means that the result of the parameter change processing is stored in [D].

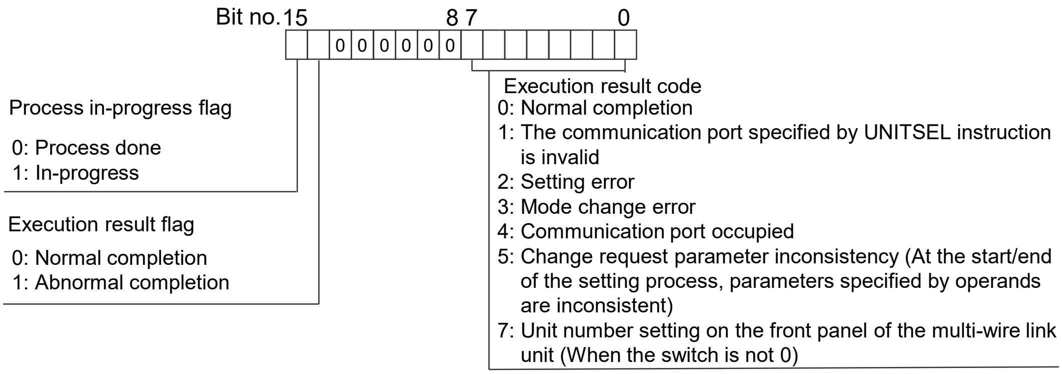
■ Communication parameter settings in W mode

Operand	Parameter	Range	Settings
[S]	Station number	U1 to U32	It can be changed only when the rotary switch on the front panel of the unit is set to 0. When using the PLC link, set it in the range of 1 to 16.
[S+1]	Link relay holding start number	U0 to U128	Specify the hold area of link relays by word numbers.
[S+2]	Link register holding start number	U0 to U128	Specify the hold area of link relays by word numbers.
[S+3]	Memory block number	U0 to U7	Memory block number of PLC link area
[S+4]	Range of link relays	U0 to U64	Link relay usable range in the above memory block
[S+5]	Range of link registers	U0 to U128	Link register usable range in the above memory block

Operand	Parameter	Range	Settings
[S+6]	Link relay transmission start number	U0 to U63	Link relay transmission start number
[S+7]	Link relay transmission size	U0 to U64	Link relay transmission size
[S+8]	Link register transmission start number	U0 to U127	Link register transmission start number
[S+9]	Link register transmission size	U1 - U127	As for the link register transmission size, up to 127 words can be sent.

■ Content of the processing result [D]

- Execution results are stored in the area of one word.
- The execution result code in the low byte is valid when the process in-progress flag of bit 15 is zero.
- If an error occurs, the execution result flag (bit 14) is turned ON. The description of the error is stored in the execution result code (bits 0 to 7).



■ Flag operations

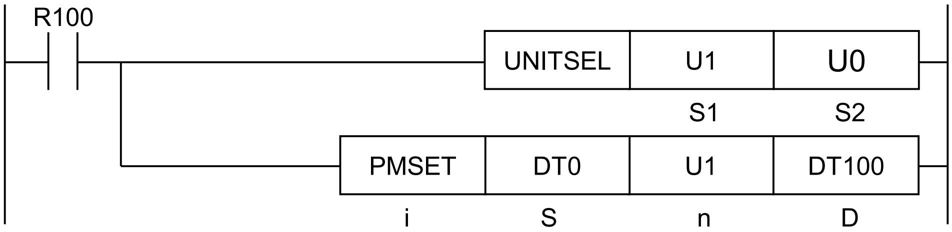
Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification, pointer access).
	To be set when the FP7 multi-wire link unit does not exist in the slot that is specified by UNITSEL.
	To be set when the device specified by [S] is out of the range.
	To be set when the number of words specified by [n] is out of the available range.
	To be set when the instruction is executed in an interrupt program and the FP7 multi-wire link unit is the target.

Info.

- For details of sample programs, refer to "16.7 PMSET / pPMSET (Change of MEWNET-W2 Parameters)".

16.7 PMSET / pPMSET (Change of MEWNET-W2 Parameters)

■ Ladder diagram



(Note 1) The above figure shows the case that the FP7 multi-wire link unit for S1=U1 (slot number 1) is specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Start of the area that stores data to be set as communication parameters
n	Specified number of words (Setting range: U1) (Note 1)
D	Starting address of the device area in the master unit that stores the processing result (1 word)

(Note 1) For MEWNET-W2, specify 1.

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●													●
n	●	●	●	●			●	●								●					●
D	●	●	●	●			●	●													●

■ Outline of operation

- The MEWNET-W2 communication parameters of the FP7 multi-wire link unit are changed with a user program.
- Set communication parameters to be changed within [n] words from the area starting with [S], and execute the PMSET/pPMSET instruction, to issue the setting change request to the unit.
- By reading setting parameters using the PMGET instruction, and setting parameters to be changed using the PMSET/pPMSET instruction, the settings can be simplified.

■ Precautions for programming

- The station number of the FP7 multi-wire link unit can be set by the PMSET/pPMSET instruction only when the rotary switch on the front panel of the unit is set to 0.
- Describe UNITSEL instruction immediately before PMSET/pPMSET instruction. Specify the slot number of the unit from which the parameters are acquired for [S1] and specify 0 for [S2].
- Checking of the processing result should be carried out when bit 15 (process in-progress flag) of the area specified by [D] is switched from 1 to 0.
- The content set by PMSET/pPMSET instruction is not held in case of power outage. Turn on the power supply again and switch to RUN mode to return to the configuration information set in the tool software.
- When setting it for the FP7 multi-wire link unit, it cannot be used in an interrupt program.

■ Comparison between the PMSET instruction and the pPMSET instruction

<table>
<tr><th>Instruction</th><th colspan="2">Characteristics</th></tr>
<tr><td rowspan="3">PMSET</td><td colspan="2">The processing result is checked during the operation processing of the PMSET instruction.(Note 1)</td></tr>
<tr><td>Advantage</td><td>The PMSET instruction can be executed without any restrictions for different COM ports.</td></tr>
<tr><td>Disadvantage</td><td>It is necessary to turn ON the execution condition of the PMSET instruction until the end of processing, and turn OFF the execution condition at a scan in which the end of data transmission is confirmed.</td></tr>
<tr><td rowspan="3">pPMSET (Note 1)</td><td colspan="2">The processing result is checked at the end of a scan.(Note 1)</td></tr>
<tr><td>Advantage</td><td>The parameter change processing is performed only once by turning on the execution condition at the time of data send.</td></tr>
<tr><td>Disadvantage</td><td>Up to 16 send operations can be performed simultaneously to different COM ports. (The total of simultaneous usage of SEND, RECV, pGPSEND, GPTRNS, and pPMSET instructions.)</td></tr>
</table>

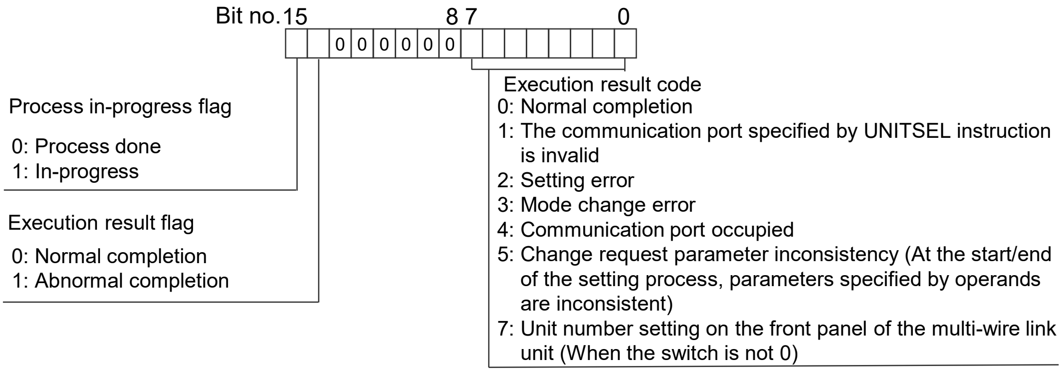
(Note 1) Checking of the processing result means that the result of the parameter change processing is stored in [D].

■ Communication parameter settings in W2 mode

Operand	Parameter	Range	Settings
[S]	Station number	U1 to U64	It can be changed only when the rotary switch on the front panel of the unit is set to 0.

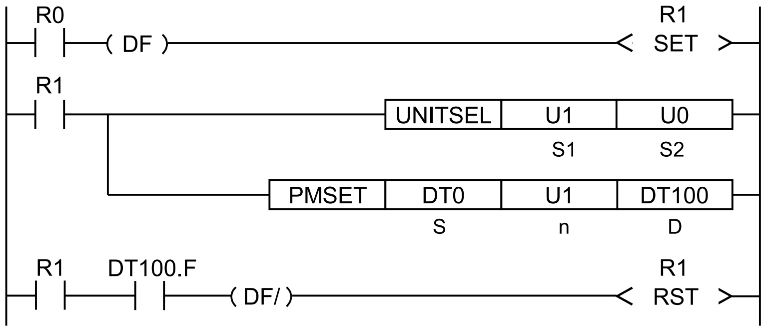
■ Content of the processing result [D]

- Execution results are stored in the area of one word.
- The execution result code in the low byte is valid when the process in-progress flag of bit 15 is zero.
- If an error occurs, the execution result flag (bit 14) is turned ON. The description of the error is stored in the execution result code (bits 0 to 7).



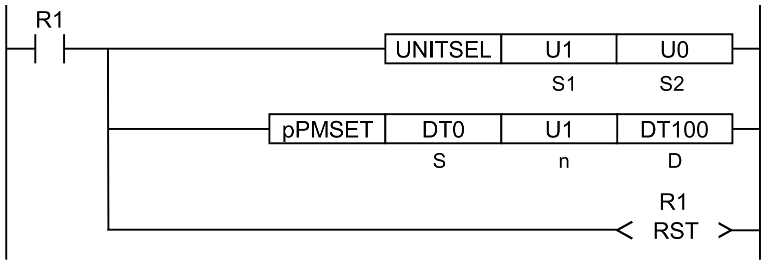
■ Program example (PMSET instruction)

- The example below shows the case when the communication parameter of one word that is stored in the area that starts with the data register DT0 is set in the FP7 multi-wire link unit in slot number 1. The processing result is stored in DT100.
- For the PMSET instruction, it is necessary to turn ON the execution condition of the PMSET instruction until the end of processing, and turn OFF the execution condition at a scan in which the end of data transmission is confirmed.



■ Program example (pPMSET instruction)

- The example below shows the case when the communication parameter of one word that is stored in the area that starts with the data register DT0 is set in the FP7 multi-wire link unit in slot number 1. The processing result is stored in DT100.
- For the pPMSET instruction, when the execution condition arises, the parameter change processing is performed only once.

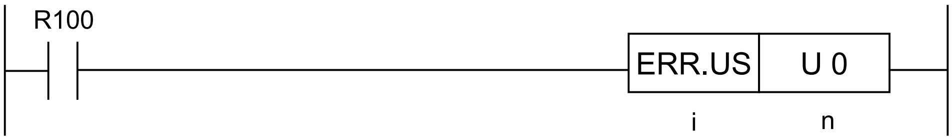


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification, pointer access).
	To be set when the FP7 multi-wire link unit does not exist in the slot that is specified by UNITSEL.
	To be set when the device specified by [S] is out of the range.
	To be set when the number of words specified by [n] is out of the available range.
	To be set when the instruction is executed in an interrupt program and the FP7 multi-wire link unit is the target.

16.8 ERR (When FP7 Multi-wire Link Unit Is Used)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●					

■ List of operands

Operand	Description
n	Specify a self-diagnostic error code. (0: Clear the self-diagnostic error.)

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
n																●	●				

■ Outline of operation

When the ERR instruction is executed with [n] set to U0, the instruction operates as follows:

- Clears errors in the FP7 multi-wire link unit.
- Resets the values of the system relays and the system data registers in the table shown below.

Device No.	Application
SR50	FP7 multi-wire link unit 1 error
SR51	FP7 multi-wire link unit 2 error
SR52	FP7 multi-wire link unit 3 error
SR53	FP7 multi-wire link unit 4 error
SR54	FP7 multi-wire link unit 5 error
SR55	FP7 multi-wire link unit 6 error

Device No.	Application
SD90	FP7 multi-wire link unit 1 error (High-order 8 bits = Error code, Low-order 8 bits = Unit number)
SD91	FP7 multi-wire link unit 2 error (High-order 8 bits = Error code, Low-order 8 bits = Unit number)
SD92	FP7 multi-wire link unit 3 error (High-order 8 bits = Error code, Low-order 8 bits = Unit number)
SD93	FP7 multi-wire link unit 4 error (High-order 8 bits = Error code, Low-order 8 bits = Unit number)
SD94	FP7 multi-wire link unit 5 error (High-order 8 bits = Error code, Low-order 8 bits = Unit number)
SD95	FP7 multi-wire link unit 6 error (High-order 8 bits = Error code, Low-order 8 bits = Unit number)

(Note 1) Error information can be read by PMGET instruction.

■ Flag operations

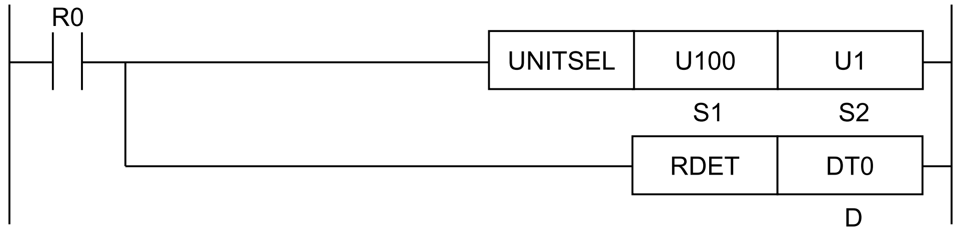
Name	Description
SR7 SR8 (ER)	To be set when the error code n is out of the range.

17 High-level Instructions (Ethernet Communication)

Applicable Models: CPS4RE*/CPS3RE*

17.1 RDET (ET-LAN Status Read)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
D	The status is stored in the first 7-word area [D] to [D+6] of the status storage area.

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●													●

■ Outline of operation

- This instruction acquires a status summary that indicates the statuses of all connections of ET-LAN.
- It describes the UNITSEL instruction immediately before the RDET instruction, and specifies the targeted ET-LAN port. For connection numbers, set any value within the range of 1 to 16.
- The acquired information is converted into integer values in the Hex format, in accordance with the relevant allocation, and stored in the 7-word area starting with [D].

■ Precautions for programming

- It is necessary to set the slot number and connection number of ET-LAN targeted in communication, using the UNITSEL instruction.

■ ET-LAN status information

- Connection status of all connections
- OPEN status
- OPEN abnormality status
- Number of connections in-progress in the FTP server

■ ET-LAN status information

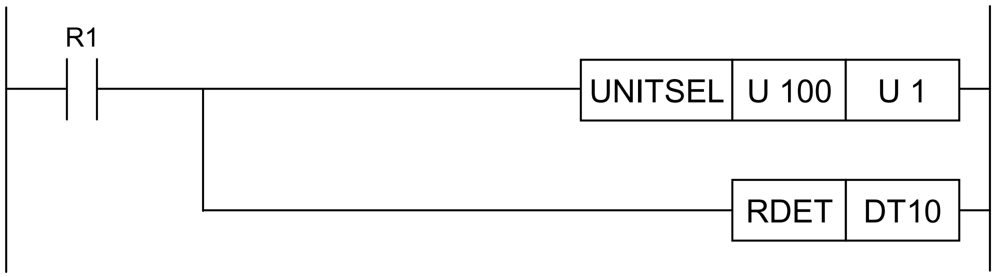
Operand	Data name		Data to be stored	
[D]	Connection status summary	Lower word	0: Other than connected 1: Connected	They are stored in the corresponding bits as allocated in the following table. S1 to S4: System connection U1 to U16: User connection FTP-S: FTP server
[D+1]		Higher word		
[D+2]	OPEN status summary	Lower word	0: Close 1: Open	
[D+3]		Higher word		
[D+4]	OPEN abnormality status summary	Lower word	0: No error 1: Error	
[D+5]		Higher word		
[D+6]	Number of connections in-progress in the FTP server			

(Note 1) Correspondence between connections and bits for the connection status summary, OPEN status summary, and OPEN abnormality status summary

Higher word							Lower word															
b15-b9	b8	b7-b4	b3	b2	b1	b0	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	FTP-S	-	U16	U15	U14	U13	U12	U11	U10	U9	U8	U7	U6	U5	U4	U3	U2	U1	S4	S3	S2	S1

■ Program example

All the connection summaries for the built-in ET-LAN in the CPU unit are acquired and stored in the 7-word area starting with DT10.

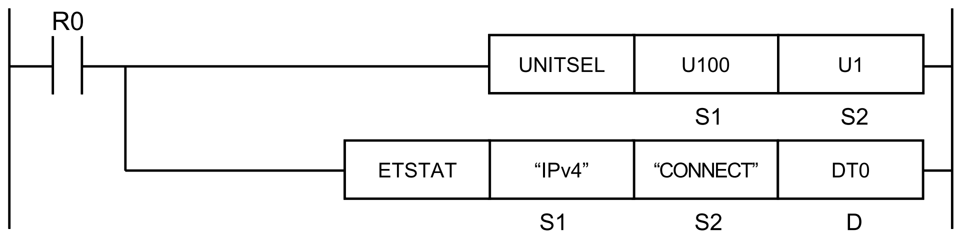


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the destination range is outside the accessible range.
	The connection specified by UNITSEL does not exist, or the value is out of the range.
	The parameter storage device specified by [D] is invalid.

17.2 ETSTAT (Acquiring Ethernet Unit Information: IP/MAC/Destination)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

ETSTAT "IPv4" "CONNECT" DT0

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates a read type, or a character constant.
S2	Starting address of the device area that stores the string data that indicates a target to be read, or a character constant.
D	Starting address of a readout destination device

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TCS	TCE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●			●	●												●	
D	●	●	●	●			●	●													

■ Outline of operation

- This instruction reads the information of the Ethernet unit.

■ Processing

- The parameter information or status information specified by [S1] and [S2] is read and stored in the area starting with [D].
- The number of words in the storage area varies according to the type of read data and the target.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] and [S2], specify the starting address of the device area that stores the string data that indicates the information to be read, or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different. Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous.
- This instruction is not available in interrupt programs.

■ Setting of the operands [S1] and [S2]

Setting item	Settings		
S1	Read type	When specifying reading IPv4 address	Specify "IPv4".
		When specifying reading IPv6 address	Specify "IPv6".
S2	Read target	■ Refer to "Data to be read and the number of words".	
D	Read destination	Specify the destination device address to which the state is read out.	

■ Data to be read and the number of words

Data to be read and the number of words vary depending on the combination of [S1] and [S2].

[S1][S2]	Storage location	Name	Number of words	Format	Description
[S1]: "IPv4" [S2]: "MAC"	[D] - [D+3]	Master unit IP address (IPv4)	4	Decimal	Master unit IPv4 address
	[D+4]- [D+7]	Subnet mask (IPv4)	4	Decimal	Subnet mask
	[D+8]- [D+11]	Default gateway (IPv4)	4	Decimal	Default gateway
	[D+12]- [D+14]	Master unit MAC address	3	Hexadecimal	Master unit MAC address
	Total number of words		15		-
[S1]: "IPv4" [S2]: "CONNECT"	[D] - [D+3]	Master unit IP address (IPv4)	4	Decimal	Master unit IPv4 address
	[D+4]- [D+7]	Subnet mask (IPv4)	4	Decimal	Subnet mask
	[D+8]- [D+11]	Default gateway (IPv4)	4	Decimal	Default gateway
	[D+12]	Destination IP address type	1	Decimal	0:IPv4
	[D+13]- [D+16]	Destination IP address	4	Decimal	Destination IP address (for IPv4) (Note 2)
	Total number of words		17		-
[S1]: "IPv4" [S2]: "CONNECT 1"	[D] - [D+3]	Master unit IP address (IPv4)	4	Decimal	Master unit IPv4 address

[S1][S2]	Storage location	Name	Number of words	Format	Description
	[D+4]- [D+7]	Subnet mask (IPv4)	4	Decimal	Subnet mask
	[D+8]- [D+11]	Default gateway (IPv4)	4	Decimal	Default gateway
or	[D+12]	Master unit port number	1	Decimal	The master unit port number in use
"CONNECT 2"	[D+13]	Destination IP address type	1	Decimal	0:IPv4
	[D+14]- [D+17]	Destination IP address	4	Decimal	Destination IP address (for IPv4) (Note 2)
	[D+18]	Destination port number	1	Decimal	Partner unit port number (Note 2)
	Total number of words		19		-

(Note 1) For IPv4, a decimal number is stored in each area of the storage.

Example) When the master unit IP address is 192.168.1.5, the IP address is stored as follows:

[D]=U192, [D+1]=U168, [D+2]=U1, [D+3]=U5

(Note 2) The destination varies depending on the setting of the communication method (TCP/UDP) as shown in the following table.

Communication method	Setting of [S2]	
	CONNECT, CONNECT1	CONNECT2
TCP client	Partner unit (server) that is set as the destination	● Partner unit (server) to which the connection is established ● No connection is established: 0
TCP server (specific partner unit)	Partner unit (client) that is allowed to be connected	● Partner unit (client) whose connection is established ● No connection is established: 0
TCP server (any partner unit)	Partner unit (client) that is connected most recently A partner unit (client) has never been connected: 0	
UDP	Partner unit that is set as the destination (for master communication)	

[S1][S2]	Storage location	Name	Number of words	Format	Description
	[D] - [D+7]	Master unit IP address 1 (IPv6)	8	Hexadecimal	Master unit IPv6 address (Manual setting)
	[D+8]- [D+15]	Master unit IP address 2 (IPv6)	8	Hexadecimal	Home IPv6 address (Link local) (Note 2)
[S1]: "IPv6" [S2]: "MAC"	[D+16] - [D+23]	Master unit IP address 3 (IPv6)	8	Hexadecimal	Home IPv6 address (Router) (Note 3)
	[D+24] - [D+31]	Master unit IP address 4 (IPv6)	8	Hexadecimal	Home IPv6 address (DHCP) (Note 3)
	[D+32]	Subnet prefix length	1	Decimal	Subnet prefix length

[S1][S2]	Storage location	Name	Number of words	Format	Description
	[D+33] - [D +40]	Default gateway (IPv6)	8	Hexadecimal	Default gateway
	[D+41] - [D +43]	Master unit MAC address	3	Hexadecimal	Master unit MAC address
	Total number of words		44		-
[S1]: "IPv6" [S2]: "CONNECT"	[D] - [D+7]	Master unit IP address 1 (IPv6)	8	Hexadecimal	Master unit IPv6 address (Manual setting)
	[D+8]- [D +15]	Master unit IP address 2 (IPv6)	8	Hexadecimal	Home IPv6 address (Link local) (Note 2)
	[D+16] - [D +23]	Master unit IP address 3 (IPv6)	8	Hexadecimal	Home IPv6 address (Router) (Note 3)
	[D+24] - [D +31]	Master unit IP address 4 (IPv6)	8	Hexadecimal	Home IPv6 address (DHCP) (Note 3)
	[D+32]	Subnet prefix length	1	Decimal	Subnet prefix length
	[D+33] - [D +40]	Default gateway (IPv6)	8	Hexadecimal	Default gateway
	[D+41]	Destination IP address type	1	Decimal	1:IPv6
	[D+42] - [D +49]	Destination IP address	8	Hexadecimal	Destination IP address (for IPv6) (Note 4)
	Total number of words		50		
[S1]: "Ipv6" [S2]: "CONNECT 1" or "CONNECT 2"	[D] - [D+7]	Master unit IP address 1 (IPv6)	8	Hexadecimal	Master unit IPv6 address (Manual setting)
	[D+8]- [D +15]	Master unit IP address 2 (IPv6)	8	Hexadecimal	Home IPv6 address (Link local) (Note 2)
	[D+16]- [D +23]	Master unit IP address 3 (IPv6)	8	Hexadecimal	Home IPv6 address (Router) (Note 3)
	[D+24]- [D +31]	Master unit IP address 4 (IPv6)	8	Hexadecimal	Home IPv6 address (DHCP) (Note 3)
	[D+32]	Subnet prefix length	1	Decimal	Subnet prefix length
	[D+33]- [D +40]	Default gateway (IPv6)	8	Hexadecimal	Default gateway
	[D+41]	Master unit port number	1	Decimal	The master unit port number in use
	[D+42]	Destination IP address type	1	Decimal	1:IPv6
	[D+43]- [D +50]	Destination IP address	8	Hexadecimal	Destination IP address (for IPv6) (Note 4)
	[D+51]	Destination port number	1	Decimal	Partner unit port number (Note 4)
	Total number of words		52		-

(Note 1) For IPv6, a hexadecimal number is stored in each area of the storage.

Example) When the master unit IP address is fe80::1234:5678:1234:5678, the IP address is stored as follows:

[D]=HFE80, [D+1]=H0, [D+2]=H0, [D+3]=H0, [D+4]=H1234, [D+5]=H5678, [D+6]=H1234, [D+7]=H5678

(Note 2) The value that is set by the FP7 CPU unit is stored in the area for the master unit IPv6 address (link local).

(Note 3) The master unit IPv6 address (router) is stored when automatic acquisition from the router is selected. The master unit IPv6 address (DHCP) is stored when automatic acquisition from the DHCP server is selected. If there is no response, "0" is stored.

(Note 4) The destination varies depending on the setting of the communication method (TCP/UDP) as shown in the following table.

<table>
<tr><th rowspan="2">Communication method</th><th colspan="2">Setting of [S2]</th></tr>
<tr><th>CONNECT, CONNECT1</th><th>CONNECT2</th></tr>
<tr><td>TCP client</td><td>Partner unit (server) that is set as the destination</td><td>- Partner unit (server) to which the connection is established
- No connection is established: 0</td></tr>
<tr><td>TCP server (specific partner unit)</td><td>Partner unit (client) that is allowed to be connected</td><td rowspan="2">- Partner unit (client) whose connection is established
- No connection is established: 0</td></tr>
<tr><td>TCP server (any partner unit)</td><td>Partner unit (client) that is connected most recently
A partner unit (client) has never been connected: 0</td></tr>
<tr><td>UDP</td><td colspan="2">Partner unit that is set as the destination (for master communication)</td></tr>
</table>

■ Execution example

Example 1) When specifying IPv4 address and MAC address

The results are stored in the 15-word area that starts with [D].

[S1]... "IPv4" [S2]... "MAC" [D]...DT0

<table>
<tr><th></th><th>Value</th><th>Description</th></tr>
<tr><td>DT0</td><td>H00C0 (U192)</td><td rowspan="4">The master unit IPv4 address is stored.
Example) 192.168.5.30</td></tr>
<tr><td>DT1</td><td>H00A8 (U168)</td></tr>
<tr><td>DT2</td><td>H0005 (U5)</td></tr>
<tr><td>DT3</td><td>H001E (U30)</td></tr>
<tr><td>DT4</td><td>H00FF (U255)</td><td rowspan="4">The subnet mask is stored.
Example) 255.255.255.0</td></tr>
<tr><td>DT5</td><td>H00FF (U255)</td></tr>
<tr><td>DT6</td><td>H00FF (U255)</td></tr>
<tr><td>DT7</td><td>H0000 (U0)</td></tr>
<tr><td>DT8</td><td>H00C0 (U192)</td><td rowspan="4">Default gateway
Example) 192.168.5.1</td></tr>
<tr><td>DT9</td><td>H00A8 (U168)</td></tr>
<tr><td>DT10</td><td>H0005 (U5)</td></tr>
<tr><td>DT11</td><td>H0001 (U1)</td></tr>
<tr><td>DT12</td><td>HAABB</td><td>The MAC address of the master unit is stored.</td></tr>
</table>

	Value	Description
DT13	HCCDD	Example) AA-BB-CC-DD-EE-FF
DT14	HEEFF	

Example 2) When specifying IPv4 address and the destination IP address of a specified connection

The results are stored in the 17-word area that starts with [D].

[S1]... "IPv4" [S2]... "CONNECT" [D]...DT0

	Value	Description
DT0	H00C0 (U192)	The master unit IPv4 address is stored. Example) 192.168.5.30
DT1	H00A8 (U168)	
DT2	H0005 (U5)	
DT3	H001E (U30)	
DT4	H00FF (U255)	The subnet mask is stored. Example) 255.255.255.0
DT5	H00FF (U255)	
DT6	H00FF (U255)	
DT7	H0000 (U0)	
DT8	H00C0 (U192)	Default gateway Example) 192.168.5.1
DT9	H00A8 (U168)	
DT10	H0005 (U5)	
DT11	H0001 (U1)	
DT12	H0000	This value indicates the type of the IP address. For IPv4, the value is "0".
DT13	H00C0 (U192)	The destination IPv4 address is stored. Example) 192.168.5.11
DT14	H00A8 (U168)	
DT15	H0005 (U5)	
DT16	H000B (U11)	

Example 3) When specifying IPv6 address and the destination IP address of a specified connection

The results are stored in the 50-word area that starts with [D].

[S1]... "IPv6" [S2]... "CONNECT" [D]...DT0

	Value	Description
DT0	HFE80	The master unit IPv6 address (manual setting) is stored. Example) fe80:0011:2233:4455:6677:8899:aabb:ccdd
DT1	H0011	
DT2	H2233	
DT3	H4455	
DT4	H6677	
DT5	H8899	
DT6	HAABB	
DT7	HCCDD	

	Value	Description
DT8 -DT15	-	The master unit IPv6 address (link local) is stored.
DT16 -DT23	-	When [Automatically acquire IPv6 address] - [Acquire from router] is selected, the master unit IPv6 address is stored. For manual setting, "0" is stored.
DT24 -DT31	-	When [Automatically acquire IPv6 address] - [Acquire from DHCP] is selected, the master unit IPv6 address is stored. For manual setting, "0" is stored.
DT32	H0100 (U64)	The subnet prefix length is stored. Example) 64
DT33	HFE80	The IPv6 address of the default gateway is stored. Example) fe80::1
DT34	H0000	
DT35	H0000	
DT36	H0000	
DT37	H0000	
DT38	H0000	
DT39	H0000	
DT40	H0001	
DT41	H0001	The type of the IP address is stored. For IPv6, the value is "1".
DT42	HFE80	The destination IPv6 address is stored. Example) fe80:0011:2233:4455:6677:8899:aabb:ccdd
DT43	H0011	
DT44	H2233	
DT45	H4455	
DT46	H6677	
DT47	H8899	
DT48	HAABB	
DT49	HCCDD	

Example 4) When specifying IPv6 address and MAC address

The results are stored in the 44-word area that starts with [D].
[S1]... "IPv6" [S2]... "MAC" [D]...DT0

	Value	Description
DT0	HFE80	The master unit IPv6 address (manual setting) is stored. Example) fe80:0011:2233:4455:6677:8899:99aa:aabb
DT1	H0011	
DT2	H2233	
DT3	H4455	
DT4	H6677	
DT5	H8899	
DT6	H99AA	
DT7	HAABB	
DT8	-	The master unit IPv6 address (link local) is stored.

	Value	Description
-DT15		
DT16 -DT23	-	When [Automatically acquire IPv6 address] - [Acquire from router] is selected, the master unit IPv6 address is stored. For manual setting, "0" is stored.
DT24 -DT31	-	When [Automatically acquire IPv6 address] - [Acquire from DHCP] is selected, the master unit IPv6 address is stored. For manual setting, "0" is stored.
DT32	H0100 (U64)	The subnet prefix length is stored. Example) 64
DT33	HFE80	The Ipv6 address of the default gateway is stored. Example) fe80::1
DT34	H0000	
DT35	H0000	
DT36	H0000	
DT37	H0000	
DT38	H0000	
DT39	H0000	
DT40	H0001	
DT41	HAABB	The MAC address of the destination unit is stored. Example) AA-BB-CC-DD-EE-FF
DT42	HCCDD	
DT43	HEEFF	

Example 5) When specifying the destination port number for a specified IPv4 connection

The results are stored in the 19-word area that starts with [D].

[S1]... "IPv4" [S2]... "CONNECT1" [D]...DT20

	Value	Description
DT20	H00C0 (U192)	The master unit IPv4 address is stored. Example) 192.168.5.30
DT21	H00A8 (U168)	
DT22	H0005 (U5)	
DT23	H001E (U30)	
DT24	H00FF (U255)	The subnet mask is stored. Example) 255.255.255.0
DT25	H00FF (U255)	
DT26	H00FF (U255)	
DT27	H0000 (U0)	
DT28	H00C0 (U192)	Default gateway Example) 192.168.5.1
DT29	H00A8 (U168)	
DT30	H0005 (U5)	
DT31	H0001 (U1)	
DT32	H9001	The master unit port number is stored.
DT33	H0000	This value indicates the type of the IP address. For IPv4, the value is "0".
DT34	H00C0 (U192)	The destination IPv4 address is stored.

	Value	Description
DT35	H00A8 (U168)	Example) 192.168.5.1
DT36	H0005 (U5)	
DT37	H000B (U11)	
DT38	H8001 (U32769)	The destination port number is stored. Example) 32769

Example 6) When specifying the destination port number for a specified IPv6 connection

The results are stored in the 52-word area that starts with [D].

[S1]... "IPv6" [S2]... "CONNECT1" [D]...DT0

	Value	Description
DT0	HFE80	The master unit IPv6 address (manual setting) is stored. Example) fe80:0011:2233:4455:6677:8899:aabb:ccdd
DT1	H0011	
DT2	H2233	
DT3	H4455	
DT4	H6677	
DT5	H8899	
DT6	HAABB	
DT7	HCCDD	
DT8 -DT15	-	The master unit IPv6 address (link local) is stored.
DT16 -DT23	-	When [Automatically acquire IPv6 address] - [Acquire from router] is selected, the master unit IPv6 address is stored. For manual setting, "0" is stored.
DT24 -DT31	-	When [Automatically acquire IPv6 address] - [Acquire from DHCP] is selected, the master unit IPv6 address is stored. For manual setting, "0" is stored.
DT32	H0100 (U64)	The subnet prefix length is stored. Example) 64
DT33	HFE80	The IPv6 address of the default gateway is stored. Example) fe80::1
DT34	H0000	
DT35	H0000	
DT36	H0000	
DT37	H0000	
DT38	H0000	
DT39	H0000	
DT40	H0001	
DT41	H9001	The master unit port number is stored.
DT42	H0001	The type of the IP address is stored. For IPv6, the value is "1".
DT43	HFE80	The destination IPv6 address is stored. Example) fe80:0011:2233:4455:6677:8899:99AA:ccdd
DT44	H0011	

	Value	Description
DT45	H2233	
DT46	H4455	
DT47	H6677	
DT48	H8899	
DT49	H99AA	
DT50	HCCDD	
DT51	H8001 (U32769)	The destination port number is stored. Example) 32769

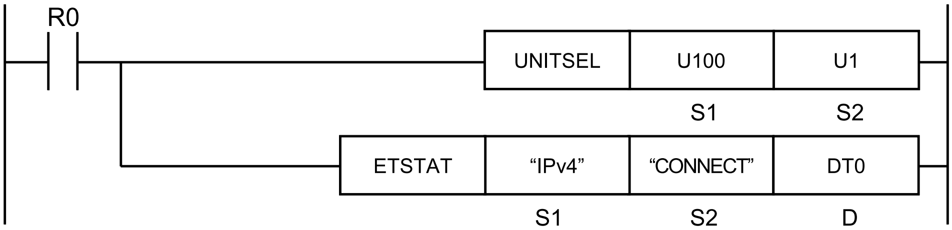
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the read area is out of the range. To be set when the read type (S1) is set to an item other than "IPv4", "IPv6", "FTPc", "HTTPc", or "SMTPc". To be set when the target to be read (S2) is set to an item other than "MAC", "CONNECT", "IDx", "LOGx", "IDALL", or "LOGALL". To be set when an unset transfer setting is specified. To be set when an unset logging/trace transfer setting is specified. To be set when the unit specified by UNITSEL is not the built-in ET-LAN in the CPU unit. Set when executed in an interrupt program.

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.3 ETSTAT (Acquiring Ethernet Unit Information: FTP/HTTP/SMTP)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

ETSTAT "FTPc" "IDALL" DT0

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates a read type, or a character constant.
S2	Starting address of the device area that stores the string data that indicates a target to be read, or a character constant.
D	Starting address of a readout destination device

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●			●	●												●	
D	●	●	●	●			●	●													

■ Outline of operation

This instruction reads the information of the Ethernet unit.

■ Processing

- The parameter information or status information specified by [S1] and [S2] is read and stored in the area starting with [D].
- The number of words in the storage area varies according to the type of read data and the target.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] and [S2], specify the starting address of the device area that stores the string data that indicates the set parameters, or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different. Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous.
- This instruction is not available in interrupt programs.

■ Setting of the operands [S1] and [S2]

Setting item	Settings		
S1	Read type	When specifying FTP client	Specify "FTPc".
		When specifying HTTP client	Specify "HTTPc" (Note 1)
		When specifying mail transmission	Specify "SMTPc".
S2	Read target	When specifying transfer numbers individually	Specify "IDx" with x being a value from 0 to 15.
		When specifying logging individually	Specify 0 to 15 for x with "LOGx" (Note 1)
		When specifying all transfer numbers	Specify "IDALL".
		When specifying all loggings	Specify "LOGALL" (Note 1)
D	Read destination	Specify the destination device address to which the state is read out.	

(Note 1) When "HTTPc" is specified for [S1], neither "LOGx" nor "LOGALL" can be specified for [S2]. If one of them is specified, an operation error occurs.

■ Data to be read and the number of words

Data to be read and the number of words vary depending on the setting of [S2].

	[S2]	Storage location	Name	Number of words	Description
1	"IDALL" "LOGALL" (Note 1) (Note 2)	[D]	Transferring ID number	1	0 to 15 Transfer setting ID or log setting ID (for FTP/HTTP) Trigger setting ID or log setting ID (for SMTP)
		[D+1]	Transferring data type	1	0: File transfer or event mail 1: Logging/trace transfer or logging/trace mail
		[D+2]	Transfer status	1	Higher byte H0: Retry not in progress, H1: During retry
					Lower byte H00: No request, H01: Waiting for transfer, H02: During login, H03: During sending, H04: During receiving, H05: Transfer complete
		[D+3]	Transfer result	1	0: Transfer succeeded, 1: Login error, 2: Transfer error, 3: Transfer canceled

	[S2]	Storage location	Name	Number of words	Description
		[D+4]- [D+9]	Latest transfer success time	6	Year, month, day, hour, minute and second when the last transfer succeeded
		[D+10]- [D +15]	Latest transfer failure time	6	Year, month, day, hour, minute and second when the last transfer failed
		[D+16]- [D +17]	Number of transfer successes (Whole)	2	Number of times that transfer succeeded
		[D+18]- [D +19]	Number of transfer failures (Whole)	2	Number of times that transfer failed
		Total number of words		20	-
2	"IDx" "IDALL" (Note 1)	[D]	Control relay(Note 3)	1	FTPc control relay, HTTPc control relay, Mail transmission control relay
		[D+1]	Execution done code(Note 4)	1	0: Normal completion. An error code is stored at abnormal completion.
		[D+2]	Transfer done code	1	FTP/HTTP/SMTP response code(Note 5)
		[D+3]- [D+4]	Number of successful executions (individual)	2	Number of times that transfer succeeded
		[D+5]- [D+6]	Number of failed executions (individual)	2	Number of times that transfer failed
		Total number of words		7	-
3	"LOGx" "LOGALL" (Note 2)	[D]	Control relay(Note 3)	1	FTPc logging control relay, HTTPc logging control relay, Mail transmission logging control relay
		[D+1]	Execution done code(Note 4)	1	0: Normal completion. An error code is stored at abnormal completion.
		[D+2]	Transfer done code	1	FTP/HTTP/SMTP response code(Note 5)
		[D+3]- [D+4]	Number of successful executions (individual)	2	Number of times that transfer succeeded
		[D+5]- [D+6]	Number of failed executions (individual)	2	Number of times that transfer failed
		Total number of words		7	-

(Note 1) When "IDALL" is specified, the entire status (20 words) and the status (7 words) for each registered ID are read.

(Note 2) When "LOGALL" is specified, the entire status (20 words) and the status (7 words) for each registered LOG are read.

(Note 3) The control relay reads the states of relays for each ID or LOG setting. Refer to "P.17-19".

(Note 4) For details of execution done codes at abnormal completion, refer to "P.17-19".

(Note 5) For details of FTP/HTTP/SMTP response codes, refer to "P.17-20" to "P.17-21".

■ Execution example

Example 1) When specifying a transfer number

The 7-word status for the transfer number that is specified by [S2] is read.
[S1]... "FTPc" [S2]... "ID5" [D]...DT0

DT0	Control relay
DT1	Execution done code
DT2	Transfer done code
DT3-DT4	Number of successful transfers (individual)
DT5-DT6	Number of failed transfers (individual)

Example 2) When "IDALL" (all ID numbers) is specified

The entire status for all transfer IDs and the status for each ID that is set are read.
[S1]... "FTPc" [S2]... "IDALL" [D]...DT0

<table>
<tr><td>DT0</td><td>Transferring ID number</td><td rowspan="8"></td></tr>
<tr><td>DT1</td><td>Transferring data type</td></tr>
<tr><td>DT2</td><td>Transfer status</td></tr>
<tr><td>DT3</td><td>Transfer result</td></tr>
<tr><td>DT4-DT9</td><td>Latest transfer success time</td></tr>
<tr><td>DT10-DT15</td><td>Latest transfer failure time</td></tr>
<tr><td>DT16-DT17</td><td>Number of transfer successes (Whole)</td></tr>
<tr><td>DT18-DT19</td><td>Number of transfer failures (Whole)</td></tr>
<tr><td>DT20</td><td>ID transfer setting</td><td>Only the bit for each ID number that is set is turned ON.</td></tr>
<tr><td>DT21-DT27</td><td>Status of ID0</td><td rowspan="5">The status data (7 words) for each of the 16 IDs is read.
Control relay: 1 word
Execution done code: 1 word
Transfer done code: 1 word
Number of successful executions (individual): 2 words
Number of failed executions (individual): 2 words</td></tr>
<tr><td>DT28-DT34</td><td>Status of ID1</td></tr>
<tr><td>DT35-DT41</td><td>Status of ID2</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>DT(21+7x)
-DT(27+7x)</td><td>Status of IDx</td></tr>
</table>

Example 3) When "LOGALL" (all LOG numbers) is specified

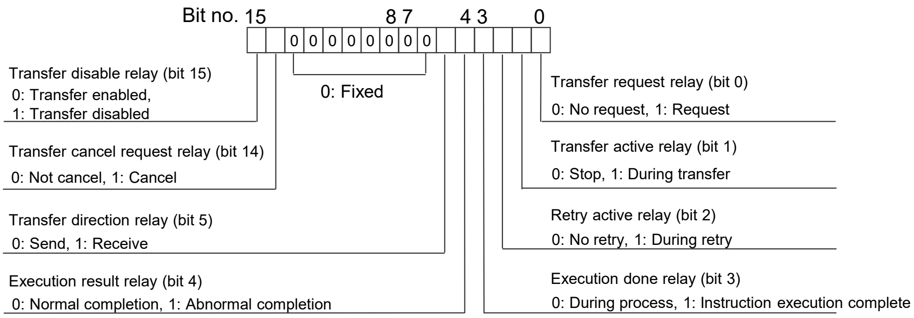
The entire status of the logging trace and the status of each ID that is set for the logging trace are read.
[S1]... "FTPc" [S2]... "LOGALL" [D]...DT0

<table>
<tr><td>DT0</td><td>Transferring ID number</td><td rowspan="3"></td></tr>
<tr><td>DT1</td><td>Transferring data type</td></tr>
<tr><td>DT2</td><td>Transfer status</td></tr>
</table>

DT3	Transfer result	
DT4-DT9	Latest transfer success time	
DT10-DT15	Latest transfer failure time	
DT16-DT17	Number of transfer successes (Whole)	
DT18-DT19	Number of transfer failures (Whole)	
DT20	LOG transfer setting	Only the bit for each ID number that is set is turned ON.
DT21-DT27	Status of LOG0	The status data (7 words) for each of the 16 LOG numbers is read. Control relay: 1 word Execution done code: 1 word Transfer done code: 1 word Number of successful executions (individual): 2 words Number of failed executions (individual): 2 words
DT28-DT34	Status of LOG 1	
DT35-DT41	Status of LOG 2	
-	-	
DT(21+7x) -DT(27+7x)	Status of LOG x	

■ Control relay

Each of the following bits is allocated for the control relay (1 word).



(Note 1) The transfer direction relay (bit 5) is "0" for logging or an HTTP client.

(Note 2) The transfer cancel request relay (bit 14) is "0" for logging or an HTTP client.

■ List of execution done codes

Code	Name	Description
0	Normal end	To be set when the processing of a transfer request instruction is completed successfully.
1	Transfer server unset error	To be set when the setting of the server that is accessed during the execution of a transfer request instruction is not completed.
2	Transfer setting unset error	To be set when the transfer setting for the transfer number that is specified during the execution of a transfer request instruction is not completed.
3	Destination group unset error	To be set when the destination group setting for the transfer number that is specified during the execution of a transfer request instruction is not completed.
4	Client registration error	To be set when a process request to a client fails to register.

Code	Name	Description
5	Transfer disabled error	To be set when the transfer disable relay is "1=Transfer disabled" for the transfer number that is specified during the execution of a transfer request instruction.
6	Transfer canceled error	To be set when the transfer cancel request relay is changed from "0" to "1" (the leading edge OFF to ON) which means a request to cancel.
7	Transfer failed error	To be set when the transfer done relay is "1=Transfer done" and the transfer failure relay is "1=Transfer failed".
8	Data decompression error (write)	To be set when an error occurs during decompression of data for registration to a client.
9	Data decompression error (read)	To be set when an error occurs during acquisition of data from a client.
10	File delete error	To be set when file deletion after transfer is specified but the file cannot be deleted.

■ List of transfer done codes (FTP error codes)

Error code	Description
226	Normal end
421	It is not possible to provide services. Ends control connection. At the time of the shutdown of server.
425	It is not possible to open data connection.
426	Connection was closed and data transfer was canceled for some reason.
450	It is not possible to execute the request for any reason of access authority or file system.
451	Processing was canceled due to a local error.
452	It is not possible to execute due to any problem in disk capacity.
500	Syntax error of commands
501	Syntax error of arguments or parameters
502	Command is not implemented.
503	The order of using commands is wrong.
504	Arguments or parameters are not implemented.
530	User could not log in.
532	Charging information must be confirmed with ACCT command for file transmission.
550	It is not possible to execute the request for any reason of access authority or file system.
551	It is not possible to execute because of a problem in the type of page structure.
552	It is not possible to execute due to any problem in disk capacity.
553	it is not possible to execute due to an incorrect file name.
1XXX	An error occurred during file deletion after transfer (not to be retried).
9XX	Client service error

■ List of transfer done codes (HTTP error codes)

Error code	Description
2XX	Normal end

Error code	Description
300	Multiple pages can be used.
301	This address was moved to another address.
302	This address is temporarily placed in another address.
303	Refer to another page.
304	Although the access was permitted, the target document has not been updated.
305	Only the access via the proxy of Location field can be permitted.
307	This address temporarily belongs to another address.
400	An error occurs in the request such as a typing mistake.
401	Failed in authentication. (This error occurs in cases such as the entry of a wrong password.)
403	You do not have access rights.
404	The page of the appropriate address does not exist, or the server is down.
405	A request of an unpermitted method type was received.
406	As a result drawn from the Accept header, unacceptable content was included.
407	Proxy authentication is required first.
408	No response was made to the request within the waiting time.
409	The request could not be completed because it conflicts with the resource of the current state.
410	The request cannot be used in the server and the destination address is unknown.
411	The request without the defined Content-Length was rejected.
412	The condition given in more than one request header field was judged incorrect in the test on the server.
413	The request was rejected because its size is larger than the processible size.
414	The request was rejected because its URI is too long.

■ List of transfer done codes (HTTP error codes)

Error code	Description
415	The requested service was rejected by the server because the requested resource is an unsupported format for the requested method.
416	The request contains the Range header field, but no If-Range request header field.
417	The expansion of the Expect request header field was not accepted.
500	An error occurs in CGI script, etc.
501	The function required for executing the request is not supported.
502	An incorrect response was received when the server acting as a gateway or proxy attempted to execute a request.
503	It is not possible to access the address for some reason.
504	A response necessary for completing the request could not be received from a server such as DNS.
505	An unsupported HTTP protocol version was received.
9XX	Client service error

■ List of transfer done codes (SMTP error codes)

Error code	Description
0	Normal end
421	Not available.
450	Failed because mailbox is not available (temporarily).
451	Server error
452	Memory shortage
500	Unknown command
501	Command argument error
502	Command is not implemented.
503	Command sequence is incorrect.
504	Command parameter is not implemented.
550	Failed because mailbox is not available (permanently).
551	User is not a local user.
552	Command was cancelled because client memory area assignment is exceeded.
553	Mailbox name is invalid.
554	Transaction failed.
9XX	Client service error

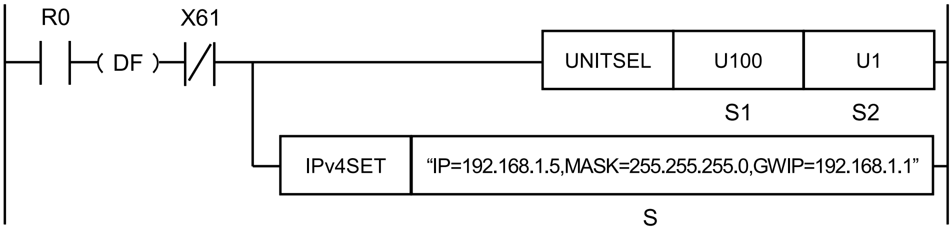
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the read area is out of the range. To be set when the read type (S1) is set to an item other than "IPv4", "IPv6", "FTPc", "HTTPc", or "SMTPc". To be set when the target to be read (S2) is set to an item other than "MAC", "CONNECT", "IDx", "LOGx", "IDALL", or "LOGALL". To be set when a combination other than the combinations listed in the restrictions on combination is specified for the type (S1) and target (S2) to be read. To be set when an unset transfer setting is specified. To be set when an unset logging/trace transfer setting is specified. To be set when the unit specified by UNITSEL is not the built-in ET-LAN in the CPU unit. Set when executed in an interrupt program.

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.4 IPv4SET (IP Address Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

IPv4SET "IP=192.168.1.5, MASK=255.255.255.0, GWIP=192.168.1.1"

■ List of operands

Operand	Description
S	The starting address of a device that stores string data representing the parameter to be set, or a character constant

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●												●	

■ Outline of operation

This instruction configures the IP (IPv4) address setting.

■ Processing

- The IPv4 setting parameter for [S] is stored in the operation work area, and the IP address (required), the subnet mask (optional), and the gateway (optional) of the Ethernet unit are initialized. Values specified by tool software are applied to items that are not modified by instructions.
- Communication is not available while Ethernet initialization is in progress.
- Statuses such as the establishment of IPv4 address or cable disconnection can be checked in the input relay area WX6 (X60 to X69).
- For details of the input relay area WX6, refer to "19.10 Ethernet Function: IP Addresses".

- If this instruction is executed with an IP address that is out of the available range, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed. Refer to the range of available IP addresses.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.
- The content set by this instruction is not held in the case of power outage. When the unit is switched back from PROG. mode to RUN mode, the configuration information set by the tool software will be preset.
- After this instruction is executed, the PLC operates as shown in the following table.

Conditions		Operation
The power supply for the PLC is switched from OFF to ON.		Setting using the configuration
The PLC is changed to PROG mode.	Changes to RUN mode without rewriting the configuration.	Setting using instructions
	Changes to RUN mode after rewriting the configuration.	Setting using the configuration

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- If the IP address setting is changed during communication, the process in progress will fail.
- Execute this instruction only once at the startup of PLC. Do not execute it repeatedly.
- It takes three seconds or longer to complete initialization following setting. Communication is disconnected until the completion of the initialization. All connections using the Ethernet function are disconnected during execution.
- This instruction is not available in interrupt programs.

■ Operand [S] setting

- Specify the starting address of a device that stores string data representing the parameter to be set, or a character constant.
- Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous.
- When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- A part of parameters can be omitted. The settings are not changed when parameters are omitted partially.
- When omitting the part before a specified keyword, omit only "keyword" without omitting ",".
- When omitting the part after a specified keyword, omit both "," and "keyword".
- It is prohibited to specify the same keyword redundantly. An error is caused in the case of redundant specification.

Setting item	Settings	
S	IPv4 address	Specify IP address (IPv4). Specify the keyword “IP=” at the beginning. IP=111.122.133.144

Setting item	Settings	
	Subnet mask	Specify a subnet mask. Specify the keyword “MASK=” at the beginning. MASK=255.255.255.0
	Default gateway	Specify an IP address for default gateway. Specify the keyword “GWIP=” at the beginning. GWIP=111.122.133.4 Specify "0" when default gateway is not to be used.

(Note 1) Setting parameters should be entered with each setting parameter separated by a comma “,”.

(Note 2) Both upper and lower cases can be used for specifying keywords.

(Note 3) Specify keywords in the order indicated in the table above.

Setting example

Example 1	S	"IP=192.168.1.5,MASK=255.255.255.0,GWIP=192.168.1.1"
Settings		IP address = 192.168.1.5; Subnet mask = 255.255.255.0; Default gateway = 192.168.1.1
Example 2	S	"IP=192.168.1.5,MASK=255.255.255.0,GWIP=0"
Settings		IP address = 192.168.1.5; Subnet mask = 255.255.255.0; Default gateway = Not used

- When an address is specified that is unusable for the parameters, the system relay SR9 (carry flag CY) is set to ON, one of the error codes 1 (IP address error) to 4 (default gateway error) is set for the system data register SD29 (Ethernet communication error code), and the process is terminated.
- For details of the available range of the address, refer to "19.10 Ethernet Function: IP Addresses".

■ Setting status when parameters are omitted

- IPv4 address is mandatory. It must be indicated.
- "Subnet mask" and "Default gateway" can be omitted. Omitted parameters are not changed.

Parameter			How to specify	Result reflected in parameters		
IP	MASK	GWIP		IP address	Subnet mask	Default gateway
Mandatory	Omit	Set	“IP=○○○○, GWIP=○○○○”	Change	Not change	Change
Mandatory	Set	Omit	“IP=○○○○, MASK=○○○○”	Change	Change	Not change
Mandatory	Omit	Omit	“IP=○○○○”	Change	Not change	Not change

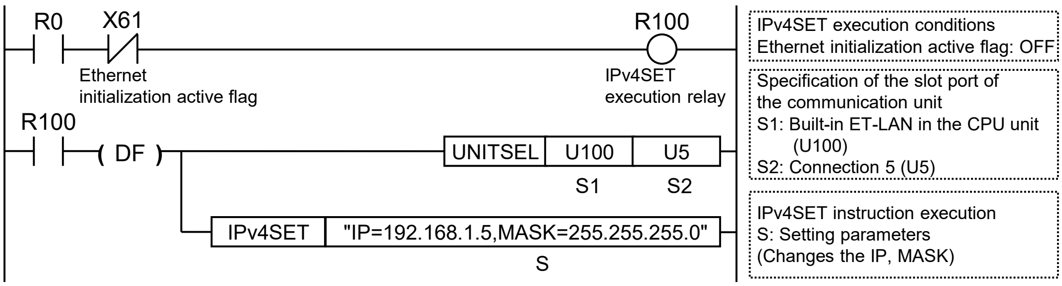
Setting example

Example 1	S	"IP=192.168.1.5,,GWIP=192.168.1.1"
Settings		IP address = 192.168.1.5; Subnet mask = Not change; Default gateway = 192.168.1.1
Example 2	S	“IP=192.168.1.5, MASK=255.255.255.0”
Settings		IP address: 192.168.1.5; Subnet mask: 255.255.255.0; Default gateway: Not change

Example 3	S	“IP=192.168.1.5”
Settings		IP address: 192.168.1.5; Subnet mask: Not change; Default gateway: Not change

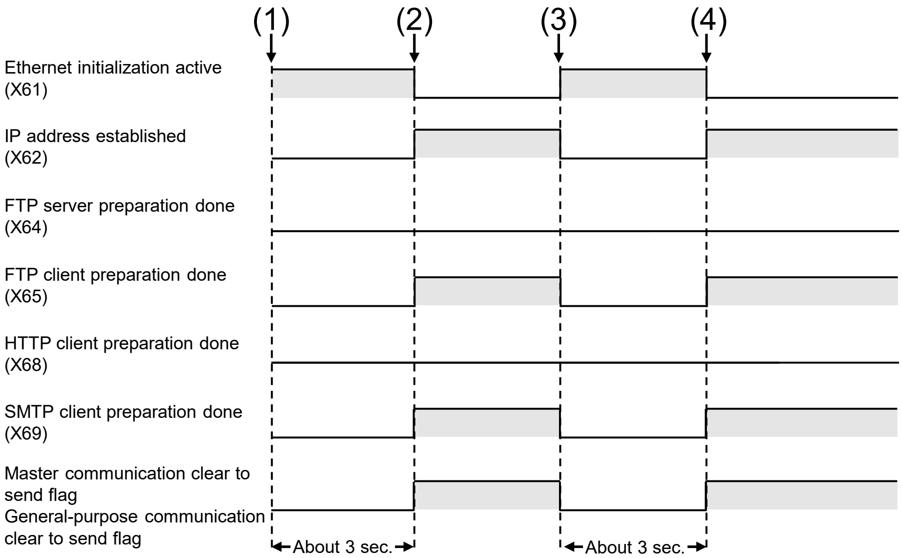
■ Program example

- After confirming that the Ethernet initialization active flag (X61) is OFF, the instruction is executed.
- The UNITSEL instruction is used to specify a slot number (LAN port: U100) and a connection number (U1 to U16).
- Once the instruction is executed, the IPv4 setting parameter will be written into the system work area, and initialization will be requested to the unit.
- Once initialization is requested, the unit will close all connections and disconnect communication.
- The unit turns OFF the IP address established flag (X62) and initializes the Ethernet unit with the value specified in the system work area.
- The unit starts auto negotiation at the time of initialization.
- The IP address established flag (X62) turns ON upon the completion of initialization. It takes about three seconds to complete initialization.
- Each communication task of FTPc, HTTPc and SMTPc starts according to the settings. It is possible to confirm those states with the ready flag for each operation.
- Each connection which automatic connection has been set is made, and the clear to send flag turns ON when the connections are complete.



■ Timing chart

The following figure shows the case for executing IPv4SET instruction using the FTP client function and mail send function (SMTP client).



(1)	PROG > RUN (Power ON)	(3)	IPv4 address setting (Executes IPv4SET instruction)
(2)	Ethernet initialization completed FTP client/SMTP client preparation completed Connection established	(4)	Ethernet initialization completed FTP client/SMTP client preparation completed Connection established

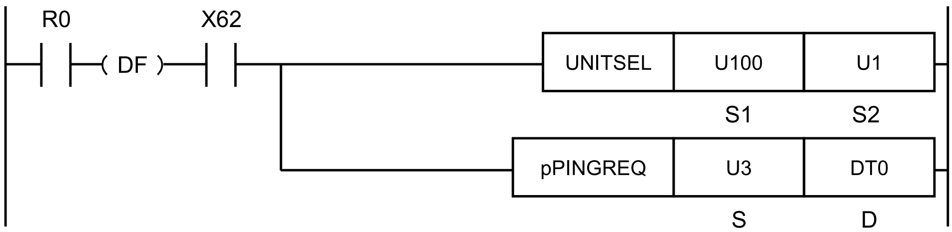
■ Flag operations

Name	Description
SR7 SR8 (ER)	Set when a value outside the range is specified for the parameter. Set when the same keyword is specified redundantly. To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). To be set when the setting is other than IPv4. To be set when executed in an interrupt program. Set when the number of characters for operand specifying character constant exceeds 256.
CY (SR9)	Set when the instruction is executed while the specified IP address is incorrect. The detail code set in SD29 is "1: Specification of incorrect IP address". Set when the instruction is executed while the specified subnet mask is incorrect. The detail code set in SD29 is "2: Specification of incorrect subnet mask". Set when the instruction is executed while the specified default gateway is incorrect. The detail code set in SD29 is "3: Specification of incorrect default gateway". To be set when executed in combination with incorrect IP addresses. The detail code set in SD29 is "4: Combination of incorrect IP addresses". Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.5 pPINGREQ (PING Request)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Number of requests to send PING (Available range: 1 to 10 times)
D	Starting address of the device area that stores the results of the PING requests

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	•	•	•	•			•	•								•	•				•
D	•	•	•	•			•	•													

■ Outline of operation

- This instruction performs a PING send request to the partner unit IP address of a specified connection for the number of times that is specified by [S].
- This instruction is used for checking the operation status of a communication relay device.
- This instruction is dedicated to ET-LAN.

■ Processing

- The PING request results are stored in the area that starts with [D].
- The timeout period for one PING response is one second (fixed).
- When the Ethernet task is initialized during a PING request, zero is set to all the areas in which the results are stored ([D]).
- The size of sent/received data is 56 bytes (fixed).
- If the number of responses to the PING is less than the number of requests, the PING may be sent one time more than the specified number of requests to send.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- If the partner unit IP address is not set, an error occurs.
- Use the ETSTAT instruction to check the target IP address for the PING request.

■ Operand [S] setting

The instruction requests the sending of PING for the number of times that is specified for [S].

Setting item	Settings		Setting range
S	Number of times for sending PING	Specify the number of times.	1 to 10 times

■ Execution result storage area [D] to [D+5]

Operand	Execution result	Description
[D]	Execution result code	HFFFF: In progress, 0: Normal end, 1x: Request error, 2x: Response error
[D+1]	No. of transmissions	
[D+2]	No. of responses	
[D+3]	Response time (maximum)	U0 to U1000 (ms) Response time is in 10 ms unit. When it is less than 10 ms, 0 is stored.
[D+4]	Response time (minimum)	
[D+5]	Response time (Average)	

■ Execution result code [D]

- In the case of abnormal request (10 to 13), it is set when the instruction is executed and the PING request is not performed.
- The response error (20) occurs when no response is returned from the Ethernet task.

Code	Execution result	
0	Normal end	
10	Double startup error	PING request instruction is being executed.
11	Number of requests to send error	The number of requests to send is not in the available range (1 to 10).
12	Ethernet unit unselected error	The unit selected with UNITSEL is not Ethernet unit.
13	Connection unused error	The specified connection is set to "Not use".
14	Disconnection error	Ethernet is disconnected.
15	Ethernet initialization in progress error	Ethernet is being initialized.
20	Ethernet task response timeout	It occurs when no response is returned from the Ethernet task.

■ **Example of processing**

Example 1) Once, when PING request, send and response has been completed successfully (when the response time is 10 ms)

[S]…U1 [D]…DT10

	When instruction is executed	PING is being executed.	PING response is completed.
DT10	H FFFF	H FFFF	H 0
DT11	K 0	K 0	K 1
DT12	K 0	K 0	K 1
DT13	K 0	K 0	K 10
DT14	K 0	K 0	K 10
DT15	K 0	K 0	K 10

Example 2) Three times, when the PING request, transmission, and response have been completed successfully (when the response time is 10, 13, or 22 ms)

[S]…U3 [D]…DT10

	When instruction is executed	PING is being executed.	PING response is completed.
DT10	H FFFF	H FFFF	H 0
DT11	K 0	K 0	K 3
DT12	K 0	K 0	K 3
DT13	K 0	K 0	K 22
DT14	K 0	K 0	K 10
DT15	K 0	K 0	K 15

Example 3) Three times, when PING request was made, and the operation timed out once (when the response time is 10 or 14 ms)

[S]…U3 [D]…DT10

	When instruction is executed	PING is being executed.	PING response is completed.
DT10	H FFFF	H FFFF	K 0
DT11	K 0	K 0	K 3
DT12	K 0	K 0	K 2
DT13	K 0	K 0	K 14
DT14	K 0	K 0	K 10
DT15	K 0	K 0	K 12
DT16			

Example 4) When PING request abended (Disconnection detection)

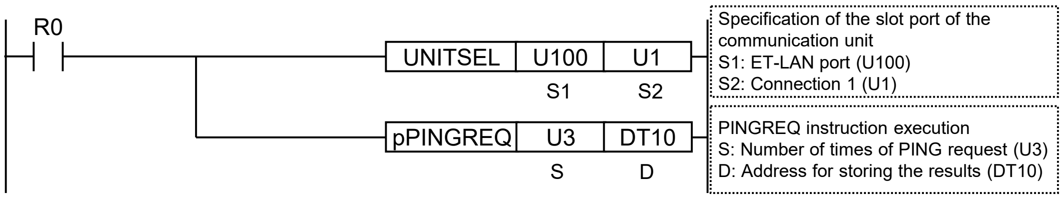
[S]…U1 [D]…DT10

DT10	K 13
DT11	K 0
DT12	K 0
DT13	K 0
DT14	K 0
DT15	K 0

When instruction is executed	PING is being executed.	PING response is completed.

■ Program example

- The UNITSEL instruction is used to specify a slot number (LAN port: U100) and a connection number (U1 to U16 for general-purpose communication).
- The PINGREQ instruction checks the operation status of the specified unit.



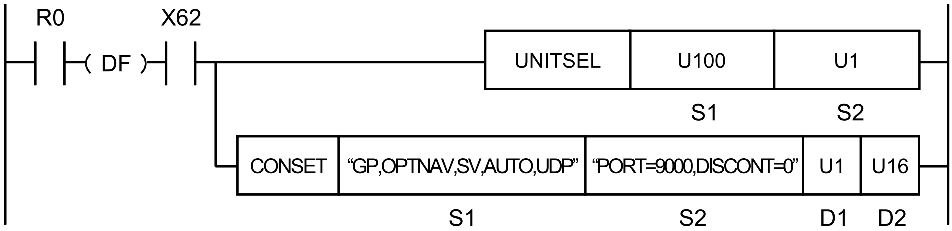
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the device address specified by [D] to [D+5] exceeds the upper limit of the device. To be set in the case of out-of-range in indirect access (index modification). To be set when executed in an interrupt program.

(Note 1) For details of the error codes stored in the system data SD29, refer to"20.2 List of System Data Registers".

17.6 CONSET (User Connection Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

CONSET "GP,OPTNAV,SV,AUTO,UDP" "PORT=9000,DISCONT=0" U1 U16

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates the parameters for operation setting, or a character constant.
S2	Starting address of the device area that stores the string data that indicates the parameters for port setting, or a character constant.
D1	Device address where the setting start connection number is stored, or a constant
D2	Device address where the setting end connection number is stored, or a constant

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	•	•	•	•			•	•												•	
S2	•	•	•	•			•	•												•	
D1	•	•	•	•			•	•								•	•				•
D2	•	•	•	•			•	•								•	•				•

■ Outline of operation

- This instruction sets the connection setting parameters that are specified by [S1] and [S2], for the connections which are in the range specified by [D1] and [D2].

■ Processing

- This instruction sets the connection setting parameters that are specified by [S1] and [S2], for the connections which are in the range specified by [D1] and [D2].
- If an incorrect IP address is specified, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- This instruction can be executed when the IP address establishment flag (X62) is ON. As an execution condition of the instruction, insert a program that checks the status of the flag (X62). If this instruction is executed when the flag (X62) is OFF, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.
- After this instruction is executed, the PLC operates as shown in the following table.

Condition		Operation
The power supply for the PLC is switched from OFF to ON.		Setting using the configuration
The PLC is changed to PROG mode.	Changes to RUN mode without rewriting the configuration.	Settings with Instructions
	Changes to RUN mode after rewriting the configuration.	Setting using the configuration

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] and [S2], specify the starting address of the device area that stores the string data that indicates the set parameters, or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different. Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous.
- Set to make [D1] be equal to or smaller than [D2].
- The maximum number that can be set for connection numbers for [D1] and [D2] is "Number of user connection information settings" in the Ethernet unit configuration data.
- When the open method is by client connection, the partner unit IP address is set incrementally one by one for each connection from the setting start connection to the setting end connection.
- When the open method is by server connection, the master unit port number is set incrementally one by one for each connection from the setting start connection to the setting end connection.
- Specify [D1] and [D2] so that the IP address of the partner unit or the master unit port does not exceed the available range.
- An operation error occurs when a connection is open or a connection with the automatic open setting exists at the time of the execution. However, when multiple connections are set, the settings for the connections before the connection in which an operation error occurs will change. The settings for the connections after the connection in which an operation error occurs will not change.
- An operation error occurs if a set of connections in the range that is specified by [D1] and [D2] contains connections for a multi connection server.
- This instruction is not available in interrupt programs.

■ Operand [S1] setting

- Specify the starting address of the device area that stores the string data that indicates the parameters for operation setting, or a character constant.
- When "INITIAL" is specified instead of parameters, the instruction operates according to the table of special keywords.

<table>
<tr><th>Setting item</th><th colspan="2">Settings</th></tr>
<tr><td rowspan="4">S1</td><td>Operation mode setting
(Essential)</td><td>Specify operation mode.
<table>
<tr><td>MEWCOM</td><td>Specifies MEWTOCOL-COM</td></tr>
<tr><td>MEW7COM</td><td>Specifies MEWTOCOL-7</td></tr>
<tr><td>MODBUS</td><td>Specifies MODBUS-TCP</td></tr>
<tr><td>MEWDAT</td><td>Specifies MEWTOCOL-DAT</td></tr>
<tr><td>MC3EBIN</td><td>Specifies the MC protocol (3E BINARY)</td></tr>
<tr><td>GP</td><td>Specifies general-purpose communication</td></tr>
<tr><td>GP_LARGE</td><td>Specifies general-purpose communication (with large capacity reception)</td></tr>
</table>
* An operation error occurs when GT is specified for the 17th or later user connections.
* An operation error occurs when GP_LARGE is specified for the second or later user connections.</td></tr>
<tr><td>Option settings
(Essential)</td><td>Specify protocol options. Available options differ according to operation modes.
OPTAV: Option available; OPTNAV: Option not available
<table>
<tr><th>Operation mode select</th><th>OPTAV</th><th>OPTNAV</th></tr>
<tr><td>MEWTOCOL-COM</td><td>Connect with FP2 ET-LAN</td><td>Not connect</td></tr>
<tr><td>MEWTOCOL7-COM</td><td>Not available</td><td>-</td></tr>
<tr><td>MODBUS-TCP</td><td>Not available</td><td>-</td></tr>
<tr><td>MEWTOCOL-DAT</td><td>Connect with FP2 ET-LAN</td><td>Not connect</td></tr>
<tr><td>MC protocol (3E BINARY)</td><td>Not available</td><td></td></tr>
<tr><td>General-purpose communication</td><td>Not append a special header</td><td>Append a special header</td></tr>
<tr><td>General-purpose communication (with large capacity reception)</td><td>Not append a special header</td><td>Cannot specify</td></tr>
</table>
* An operation error occurs if OPTNAV is specified when GP_LARGE is specified for the operation mode setting.</td></tr>
<tr><td>Open method setting
Server/Client
(Essential)</td><td>Specify open method (Server/Client).
CL: Client connection, SV: Server connection (any partner)</td></tr>
<tr><td>Open method setting
Auto/Manual</td><td>Specify open method (Auto/Manual).
AUTO: Open automatically
MANU: Not open automatically (Open with open instruction)</td></tr>
</table>

Setting item	Settings	
	(Essential)	
	Communication method setting (Essential)	Specify communication method (TCP/UDP). TCP: TCP/IP setting, UDP: UDP/IP setting * An operation error occurs if UDP is specified when GP_LARGE is specified for the operation mode setting.

(Note 1) For operation settings, input each setting parameter separated by a comma ",".

(Note 2) Both upper and lower cases can be used for specifying keywords.

(Note 3) The operation setting parameters cannot be omitted.

(Note 4) There is the following difference between high-level instructions and configuration data when UDP is specified for the communication method. Although the open method (server/client) setting is not available for configuration data, it must be specified either server or client for high-level instructions. Specify SV for using it as slave connection, and specify CL for using it as master connection.

(Note 5) General-purpose communication (with large capacity reception) is available for the CPU units CPS4R* / CPS3R*.

Setting example

Example 1	S1	"MEWCOM,OPTAV,CL,AUTO,TCP"
Settings		Operating mode setting: MEWCOM, Option setting: Option available Open type (Server/Client): Client, Open type (Automatic/Manual): Open automatically, Communication type: TCP/IP
Example 2	S1	"MODBUS,OPTNAV,SV,MANU,UDP"
Settings		Operation mode setting: MODBUS, Option setting: Option not available, Open method (Server/Client): Server (any partner), Open method (Automatic/Manual): Not open automatically, Communication method: UDP/IP
Example 3	S1	"GP,OPTNAV,SV,AUTO,UDP"
Settings		Operation mode setting: GP, Option setting: Option not available Open method (Server/Client): Server (any partner), Open method (Automatic/Manual): Open automatically, Communication method: UDP/IP
Example 4	S1	"GP_LARGE,OPTAV,SV,MANU,TCP"
Settings		Operation mode setting: General-purpose communication (large capacity general-purpose reception), Option setting: Add no special header Open type (server/client): Server connection (any partner) Open method (Automatic/Manual): Not open automatically, Communication method: TCP/IP

■ Special keyword of operand [S1] setting

Special keyword	Description
INITIAL	Set an initial value.

Setting example

Example	S1	“INITIAL”

Settings	Operating mode setting: MEWTOCOL-COM, Option setting: Option not available, Open type (Server/Client): Client, Open type (Automatic/Manual): Open automatically, Communication type: TCP/IP

■ Operand [S2] setting

- Specify the starting address of the device area that stores the string data that indicates the parameters for port setting, or a character constant. Setting items differ between specifying Client and specifying Server. It is prohibited to specify the same setting parameter redundantly. An error is caused in the case of redundant specification.

<When specifying Client (when connecting from FP7)>

- The partner unit IP address is set by being incremented by one for each connection from the setting start connection number to the setting end connection number. The increment range is the lower one block only.
- Partner unit port numbers and unused connection disconnect time are not incremented.
- An error occurs if the value of IPv4 address exceeds 255 or the value of IPv6 address exceeds FFFFh when they are incremented.

Setting item	Settings	
S2	Partner unit IP address (Essential)	Specify the destination unit IP address of the setting start connection. Specify the keyword "IPv4=" or "IPv6=" at the beginning. ● For an IPv4 address IPv4=111.122.133.144 ● For an IPv6 address IPv6=1111:1222::1555:0:0:1888
		* When specifying IPv4, 000.000.000.000(0.0.0.0) cannot be specified. * When specified, CY flag (SR9) turns ON and 1 (IP address error) is set to SD29, and the process is terminated. * An operation error does not occur. The setting is not made.
	Partner unit Port No. (Essential)	Specify the port number (1 to 65535) of partner unit. Specify the keyword “PORT=” at the beginning. PORT=xxxx
	Unused connection disconnect time (Essential)	Specify unused connection disconnect time (0 to 4294967295: 10 ms unit). However, when 0 is specified, connection is not automatically disconnected. Specify the keyword "DISCOUNT=" at the beginning. DISCONT=xxxx

(Note 1) Both upper and lower cases can be used for specifying keywords.

(Note 2) All the items cannot be omitted. Specify them in the order of the above table.

Setting example

Example 1	S2	"IPv4=192.255.2.10,PORT=9000,DISCONT=0"
Settings		Partner unit IP address: 192.255.2.10, Partner unit port number: 9000, Unused connection disconnect time: 0
Example 2	S2	"IPv6=1111:1222::1555:0:0:1999,PORT=10000,DISCONT=30000"

Settings		Partner unit IP address: 1111:1222::1555:0:0:1999, Partner unit port number: 10000, Unused connection disconnect time: 30000
Example 3	S2	"IPv4=192.255.100.11,PORT=2500,DISCONT=50"
Settings		Partner unit IP address: 192.255.100.11, Partner unit port number: 2500, Unused connection disconnect time: 50

<When specifying Server (when connecting to FP7)>

- The master unit port number is set by being incremented by one for each connection from the setting start connection number to the setting end connection number. The unused connection disconnect time is not incremented.
- An error occurs if the port number exceeds 65535 when it is incremented.

Setting item	Settings	
S2	Master unit port number (Essential)	Specify the master unit port number (1 to 65535) of setting start connection. Specify the keyword "PORT=" at the beginning. PORT=xxxx
	Unused connection disconnect time (Essential)	Specify unused connection disconnect time (0 to 4294967295: 10 ms unit). However, when 0 is specified, connection is not automatically disconnected. Specify the keyword "DISCOUNT=" at the beginning. DISCONT=xxxx

Setting example

Example 1	S2	"PORT=9000,DISCONT=0"
Settings		Master unit port number: 9000, Unused connection disconnect time: 0
Example 2	S2	"PORT=10000,DISCONT=30000"
Settings		Master unit port number: 10000, Unused connection disconnect time: 30000
Example 3	S2	"PORT=10000,DISCONT=70"
Settings		Master unit port number: 10000, Unused connection disconnect time: 70

■ Operand [D1] setting

- Specify the device address storing a setting start connection number or a constant.

Setting item	Settings		Setting range
D1	Setting start connection number	Specify setting start connection number.	1 to 216 (maximum)

■ Operand [D2] setting

- Specify the device address storing a setting end connection number or a constant.

Setting item	Settings		Setting range
D2	Setting end connection number	Specify setting end connection number.	1 to 216 (maximum)

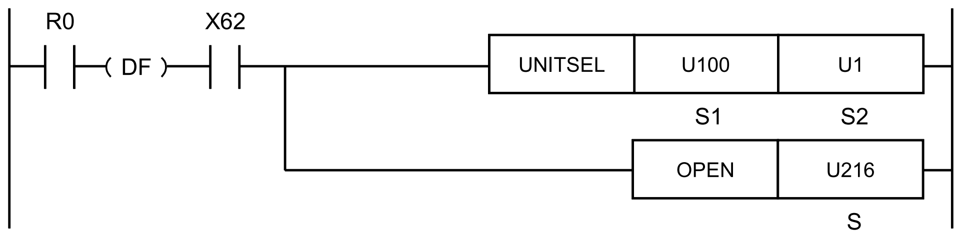
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when [D1] is larger than [D2]. To be set when [D1] and [D2] exceed the number of user connection information settings. Set when a value outside the range is specified for the parameter. Set when the same keyword is specified redundantly. To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). To be set when the lower one block of IP address exceeds the available range when incremented. To be set when the master unit port number exceeds the settable range when incremented. To be set when executed in an interrupt program. Set when the number of characters for operand specifying character constant exceeds 256. To be set when there is an open connection. To be set when there is a connection with the automatic open setting. An operation error occurs if a set of connections in the range that is specified by [D1] and [D2] contains connections for a multi connection server. To be set when "GP_LARGE" is specified for the operation mode setting in [S1] and "OPTNAV" is specified for the option setting in [S1]. To be set when "GP_LARGE" is specified for the operation mode setting in [S1] and "UDP" is specified for the communication method in [S1]. To be set when "GP_LARGE" is specified for the operation mode setting in [S1] and [D1] or [D2] is not 1.
CY (SR9)	Set when the instruction is executed while the specified IP address is incorrect. The detail code set in SD29 is "1: Specification of incorrect IP address". To be set when the instruction is executed while the IP address is not established. The detail code set in SD29 is "12: IP address not established".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.7 OPEN (Connection Open)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Device address storing the connection number to be opened or a constant.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●								●	●				●

■ Outline of operation

- This instruction opens a specified connection.

■ Processing

- The communication circuit of the connection specified by [S] is opened.
- When the connection is already open, this instruction is not executed.
- The completion of the open operation can be confirmed by the status (ON) of the clear to send flag for the master communication or general-purpose communication.
- The open method setting (automatic/manual) is not changed.
- This instruction can be executed when the IP address establishment flag (X62) is ON. As an execution condition of the instruction, insert a program that checks the status of the flag (X62). If this instruction is executed when the flag (X62) is OFF, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.
- During the processing of connection, the system relay SR9 (carry flag CY) is set and this instruction is not executed.
- When the connection is occupied, this instruction is not executed.

- To open connections for a multi connection server, specify the first connection. If this instruction is executed for a connection other than the first connection, an operation error occurs.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- When the open type is set to open automatically, it is not necessary to execute this instruction.
- This instruction is not available in interrupt programs.

■ Operand [S] setting

Specify the device address storing the connection number to be opened or a constant.

Setting item	Settings		Setting range
S	Connection number	Specify a connection number.	1 to 216

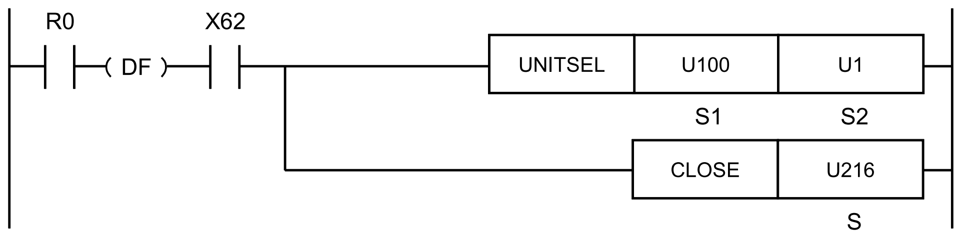
■ Flag operations

Name	Description
SR7 SR8 (ER)	Set when a value outside the range is specified for the parameter. To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). Set when executed in an interrupt program. To be set when this instruction is executed for a connection other than the first connection in a multi connection server.
CY (SR9)	To be set when the instruction is executed while the IP address is not established. The detail code set in SD29 is "12: IP address not established". To be set when executed during the processing of connection. The detail code set in SD29 is "14: Connection being processed".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.8 CLOSE (Connection Close)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Device address storing the connection number to be closed or a constant.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●								●	●				●

■ Outline of operation

- This instruction closes a specified connection.

■ Processing

- The communication circuit of the connection specified by [S] is closed.
- When the communication circuit is already closed, this instruction is not executed.
- The completion of the close operation can be confirmed by the status (OFF) of the clear to send flag for the master communication or general-purpose communication.
- This instruction can be executed when the IP address establishment flag (X62) is ON. As an execution condition of the instruction, insert a program that checks the status of the flag (X62). If this instruction is executed when the flag (X62) is OFF, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the connection is occupied, the system relay SR9 (carry flag CY) is set and this instruction is not executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

- To close connections for a multi connection server, specify the first connection. If this instruction is executed for a connection other than the first connection, an operation error occurs.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- When the open type is set to open automatically, the connection is closed once, but it will be automatically connected again.
- This instruction is not available in interrupt programs.

■ Operand [S] setting

Specify the device address storing the connection number to be closed or a constant.

Setting item	Settings		Setting range
S	Connection number	Specify a connection number.	1 to 216 (maximum)

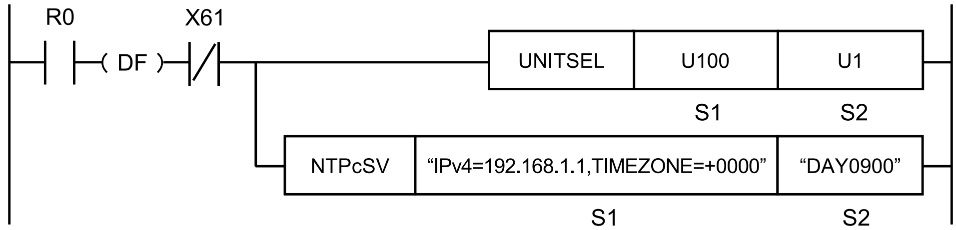
■ Flag operations

Name	Description
SR7 SR8 (ER)	Set when a value outside the range is specified for the parameter. To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). Set when executed in an interrupt program. To be set when this instruction is executed for a connection other than the first connection in a multi connection server.
CY (SR9)	To be set when the instruction is executed while the IP address is not established. The detail code set in SD29 is "12: IP address not established". To be set when the instruction is executed while the connection is occupied. The detail code set in SD29 is "15: Connection being occupied".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.9 NTPcSV (NTP Destination Server Setting Instruction)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

NTPcSV "IPv4=192.168.1.1,TIMEZONE=+0000" "DAY0900"

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates the parameters for specifying a server, or a character constant.
S2	Starting address of the device area that stores the string data that indicates the parameters for specifying time acquisition timing, or a character constant.

■ Available devices (●: Available)

Operand	16-Bit device:											32-Bit device: (Note 1)			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●			●	●												●	

(Note 1) Only 16-bit devices can be modified. (Character constants cannot be specified.)

■ Processing

- This instruction sets a destination NTP server and time zone to send a time adjustment request.
- The NTP server address and the time zone are set in the CPU unit (built-in ET-LAN) according to [S1].
- The timing of the time acquisition request is set according to [S2].
- The data that is already set in the CPU configuration is invalid and the NTP time acquisition request is executed at the timing that is specified by this instruction.

- The settings remain valid until the power is turned OFF. Also, the setting is valid until PROG. mode changes to RUN mode after a project is copied using an SD card and a communication command (download of project, backup/restoration of project, writing of configuration fixed area, forced cancel of security, initialization of system (factory default setting)) is executed. However, the time acquisition timing (CPU configuration) setting follows the project setting simultaneously when the project is changed, regardless of the mode change from PROG. to RUN.
- The settings will not be lost even when the IPv4SET instruction is executed.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Details of operand [S1]

Starting address of the device area that stores the string data that indicates the parameters for specifying a server, or a character constant.

Setting item	Settings	
S1	NTP server IP address or host name (can be omitted)	Specify IP address or host name. For an IP address, specify the keyword "IPv4=" or "IPv6=" at the beginning. For a host name, specify "HOST=". * The number of characters should be within 68 characters including a keyword ("IPv4=", "IPv6=", or "HOST="). ● For an IPv4 address "IPv4=111.122.133.144" ● For an IPv6 address "IPv6=1111:1222::1555:0:0:1888" ● For a host name: "HOST=ntp.pidsx.com"
	Time zone Set (can be omitted)	Specify the time zone setting in "dHHMM" format (d: "+" or "-", HH: hour "00" to "24", MM: minute "00" to "59"). Specify the keyword "TIMEZONE=" at the beginning. ● For GMT+09:00 (Osaka, Sapporo, Tokyo) "TIMEZONE=+0900" ● For GMT-10:00 (Hawaii) "TIMEZONE=-1000" ● For GMT 00:00 (Greenwich Mean Time) "TIMEZONE=+0000"

- Input "IP address or host name of NTP server" and "Time zone setting" separated by a comma ",".
- The keywords should be specified in the order shown in the above table. Upper and lower case characters can be used for specifying keywords.
- A part of parameters can be omitted. Omitted parameters are not changed.
- When omitting the part before a specified keyword, omit only "keyword" without omitting ",".
- When omitting the part after a specified keyword, omit both "," and "keyword".
- It is prohibited to specify the same keyword redundantly. An error is caused in the case of redundant specification.

Setting example

Example 1	S1	"IPv4=111.122.133.144,TIMEZONE=+0900"
	Settings	NTP server (IPv4): 111.122.133.144, Time zone: GMT+09:00

Setting example when omitting a part of a keyword

Example 2	S1	"HOST=ntp.pidsx.com"
	Settings	NTP server (Host name): ntp.pidsx.com, Time zone: Unchanged
Example 3	S1	",TIMEZONE=+0900"
	Settings	NTP server (IPv4): Unchanged, Time zone: GMT+09:00
Example 4	S1	""
	Settings	NTP server (IPv4): Unchanged, Time zone: Unchanged

■ Details of operand [S2]

Specify the starting address of the device area that stores the string data that indicates the parameters for specifying time acquisition timing, or a character constant.

Setting item	Settings	
S2	Once daily Appointed time (can be omitted)	Time data acquisition timing: Once a day at a specified time. DAY=DISABLE: Not set HHMM: Set, HH: hour "00" to "23", MM: minute "00" to "59"
	Once weekly Specified day of the week and time (can be omitted)	Time data acquisition timing: Once a week at a specified day of the week and time. WEEK=DISABLE: Not set WHHMM: Set, W: 0 (Sunday) to 6 (Saturday), HH: hour "00" to "23", MM: minute "00" to "59"
	Once monthly Specified date and time (can be omitted)	Time data acquisition timing: Once a month at a specified date and time. MONTH=DISABLE: Not set DDHHMM: Set, DD: "01" to "28", HH: hour "00" to "23", MM: minute "00" to "59"

- Input "Once daily/specified time", "Once weekly/specified day of the week and time" or "Once monthly/specified date and time" separated by a comma ",".
- The keywords should be specified in the order shown in the above table. Both upper and lower cases can be used for specifying keywords.
- A part of parameters can be omitted. Omitted parameters are not changed.
- When omitting the part before a specified keyword, omit only "keyword" without omitting ",".
- When omitting the part after a specified keyword, omit both "," and "keyword".
- It is prohibited to specify the same keyword redundantly. An error is caused in the case of redundant specification.

Setting example

Example 1	S2	"DAY=1234,WEEK=62345,MONTH=010010"
	Settings	Once daily/specified time: Once daily at 12:34, Once weekly/specified day of the week and time: Every Saturday at 23:45, Once monthly/specified date and time: First day of every month at 00:10

Setting example when omitting a part of a keyword

Example 2	S2	“DAY=1234”
	Settings	Once daily/specified time: Once daily at 12:34, Once weekly/specified day of the week and time: Unchanged, Once monthly/specified date and time: Unchanged
Example 3	S2	",WEEK=01234"
	Settings	Once daily/specified time: Unchanged, Once weekly/specified day of the week and time: Every Sunday at 12:34, Once monthly/specified date and time: Unchanged
Example 4	S2	",,MONTH=112233"
	Settings	Once daily/specified time: Unchanged, Once weekly/specified day of the week and time: Unchanged, Once monthly/specified date and time: 11th day of every month at 22:33
Example 5	S2	"DAY=DISABLE,WEEK=DISABLE,MONTH=282356"
	Settings	Once daily/specified time: Not set (setting disabled), Once weekly/specified day of the week and time: Not set (setting disabled), Once monthly/specified date and time: 28th day of every month at 23:56
Example 6	S2	""
	Settings	Once daily/specified time: Unchanged, Once weekly/specified day of the week and time: Unchanged, Once monthly/specified date and time: Unchanged

■ Precautions for programming

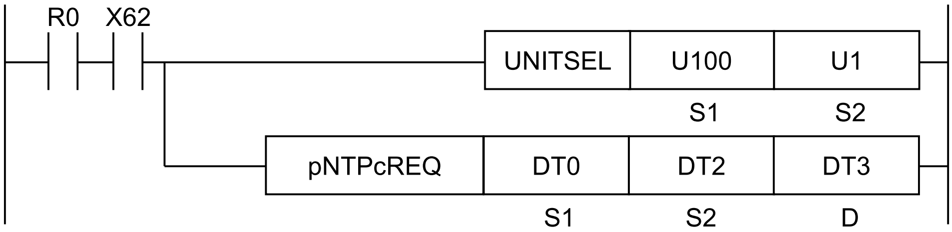
- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] and [S2], specify the starting address of the device area that stores the string data that indicates the set parameters, or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- "When power supply is ON" cannot be set for the acquisition timing.
- After this instruction is executed, if one of the settings "Specified time once a day", "Specified day of the week once a week", and "Specified date and time once a month" is set to "Set", set to "Use automatic retrieval from the SNTP server as a method to acquire time data". If one of the settings "Specified time once a day", "Specified day of the week once a week", and "Specified date and time once a month" is set to "Set", set to "Do not use automatic retrieval from the SNTP server as a method to acquire time data".
- This instruction is not available in interrupt programs.
- This instruction cannot be used while the NTP time is being acquired. It is recommended to use this instruction only once.

■ Flag operations

Name	Description
SR7 SR8 (ER)	Set when a value outside the range is specified for the parameter.
	To be set when the unit specified by UNITSEL is not the built-in ET-LAN in the CPU unit.
	Set when executed in an interrupt program.
	To be set while acquiring the time of NTP.
CY (SR9)	Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

17.10 pNTPcREQ (Time Adjustment Request Instruction)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S1	Specify the number of times of request processing. (Settable range: 0 to 20 times)
S2	Specify the interval of request processing. (Settable range: 16 to 600 seconds)
D	Specify the starting address of the device area that stores the execution result of time adjustment. HFFFF: In progress, 0: Normal end, 1x: Request error, 2x: Communication error, 3x: Response error

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●								●	●				●
S2	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													

■ Outline of operation

- This instruction requests to adjust the time.

■ Processing

- Set the number of processing times for the time adjustment request by S1.
- If the time adjustment timeout is predicted, add the number of times of retransmission.
- For canceling the time adjustment retransmission, set the number of processing times to 0. For the retransmission canceling process, the execution result is not stored in [D].
- Set the time adjustment processing interval by [S2].
- The execution result of time adjustment is stored in the area that starts with [D].

- The timeout period for one time adjustment attempt is fixed to three seconds. When multiple time adjustment attempts are specified, a new request starts after the timeout period elapses (3 seconds) plus [S2] seconds (the processing interval).
- The total timeout period (seconds) for time adjustment is obtained by the following formula: S1x3 + (S2x(S1-1)). (Here, S1 is larger than 0.)

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- It is necessary to set the SNTP server address in [Built-in ET-LAN configuration]. The **CPU configuration**>**Time data acquisition method**>**Acquire automatically from SNTP server** setting does not affect the instruction.
- When the Ethernet task is initialized during a time adjustment request, zero is set to all the areas in which the results are stored ([D]).
- Do not execute time adjustment continuously as there is a possibility that the access to the server will be prohibited.
- When this instruction is executed during the execution of an SNTP request using the time specified in **CPU configuration**>**Time data acquisition method**>**Acquisition timing**, the request starts as a new request taking the execution of this instruction as a starting point.
- When the time specified in "CPU configuration" comes during the execution of this instruction, the execution of this instruction takes priority.
- The execution of the SNTP request by the [CPU configuration] setting is performed as follows; Timeout period = 3 seconds, Number of processing times = 20 times, Processing interval = 16 seconds.
- Even when the number of processing times [S1] is set to zero, specify a value within the normal range for the processing interval [S2].
- The NTP time adjustment processing by this instruction continues even after the PLC mode switches from RUN to PROG.

■ Execution result code [D]

- In the case of abnormal request (10 to 15), the time adjustment request that is set when the instruction is executed is not performed.
- The communication error (20) occurs if no response is returned from the server after time adjustment is requested. (No response means that any response is not returned even when the processing is performed for the specified number of times.)
- The response error (30) occurs if no response is returned from the Ethernet task.

Code	Execution result	
0	Normal end	
10	Double startup error	The time adjustment request instruction is being executed (Note 1)
11	SNTP server address setting error	ET-LAN setting, SNTP server address setting = "0.0.0.0"
12	Disconnection error	Ethernet is disconnected.
13	Ethernet initialization in progress error	Own IP address is not established. (X62 OFF)
14	Number of processing times setting error	The specified number of processing times is out of the range.

Code	Execution result	
15	Processing interval setting error	The specified processing interval is out of the range.
20	Response timeout error	The time adjustment response exceeds the specified time (Note 2)
30	Ethernet task response timeout	It occurs when no response is returned from the Ethernet task.

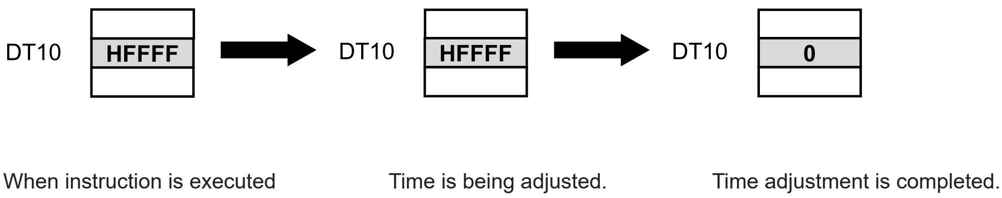
(Note 1) The instruction with the number of processing times set to 0 to cancel the time adjustment request instruction does not cause a double startup error.

(Note 2) This error also occurs if the NTP IP address is unresolved.

■ Example of processing

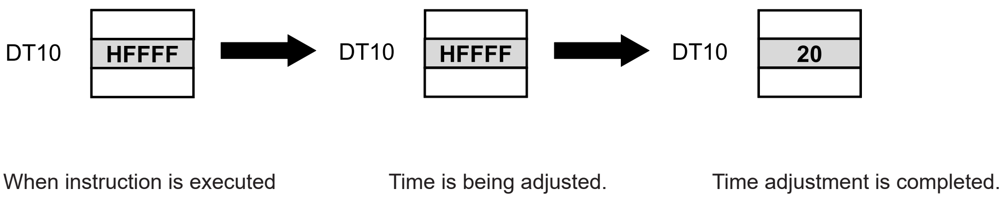
Example 1) Time adjustment request → Time is being adjusted. → Time adjustment retrieval is normally ended.

[S1]...U1 [S2]...U16 [D]...DT10



Example 2) When a response timeout has occurred in the time adjustment

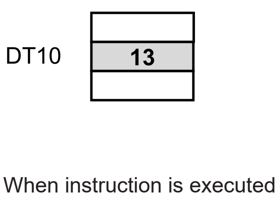
[S1]...U3 [S2]...U16 [D]...DT10



Timeout occurs after 41 seconds: S1x3 + (S2x(S1-1)). (Total processing time before timeout is 3 seconds multiplied by 3. Total processing interval is 16 seconds multiplied by (3-1).)

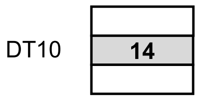
Example 3) When the time adjustment request ends abnormally (Ethernet initialization in progress error)

[S1]...U2 [S2]...U16 [D]...DT10



Example 4) When the time adjustment request ends abnormally (Number of processing times setting error)

[S1]...U2 [S2]...U21 [D]...DT10

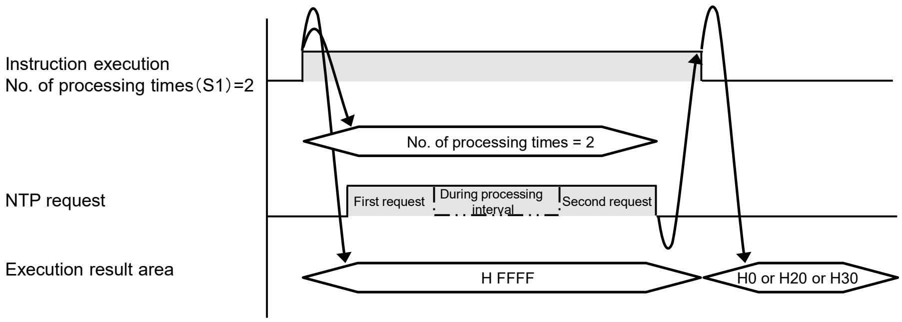


When instruction is executed

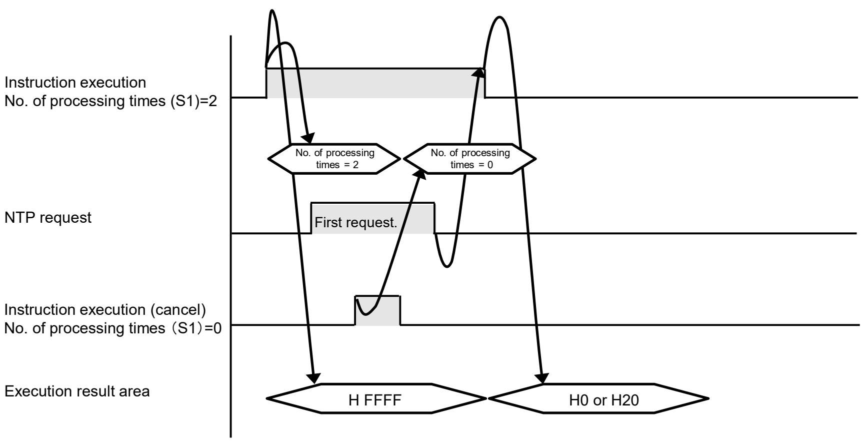
■ Cancelation of the time adjustment request

- When the number of processing times is set to zero and the pNTPcREQ instruction is executed, a time adjustment request that is being executed is canceled.
- Only canceling the request process is performed. The response waiting state for an NTP request (whose timeout period is fixed to 3 seconds) cannot be canceled.
- The following figures show some cases when the pNTPcREQ instruction is executed with the number of times (S1) being 2.

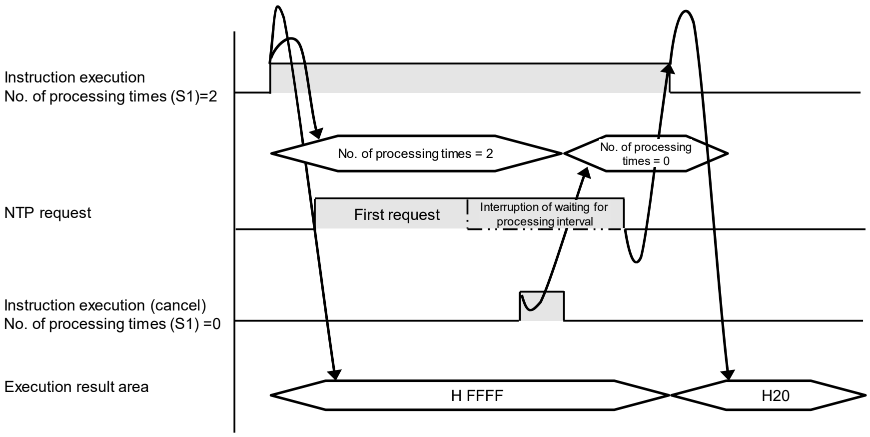
1) Normal execution



2) Cancelation of the processing during an NTP request



3) Cancelation of the processing during the processing interval of an NTP request

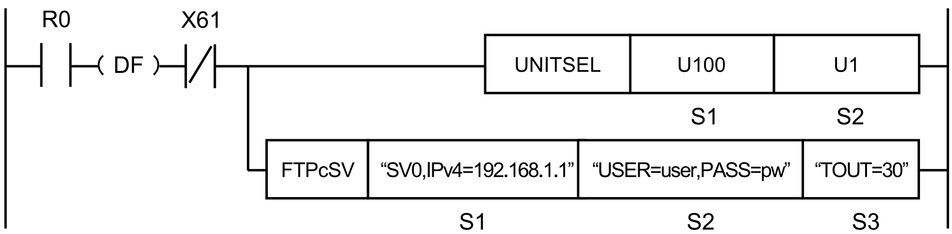


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when the unit specified by UNITSEL is not the built-in ET-LAN.
	Set when executed in an interrupt program.

17.11 FTPcSV (FTP Client Connected Server Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

FTPcSV "SV0,IPv4=192.168.1.1" "USER=user,PASS=pw" "TOUT=30"

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates the parameters for specifying a server, or a character constant.
S2	Starting address of the device area that stores the string data that indicates the login setting parameters, or a character constant.
S3	Starting address of the device area that stores the string data that indicates the detailed setting parameters, or a character constant.

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	•	•	•	•			•	•												•	
S2	•	•	•	•			•	•												•	
S3	•	•	•	•			•	•												•	

■ Outline of operation

- This instruction sets the server to which the FTP client is connected.

■ Processing

- The settings for the server to which the FTP client is connected are specified in the CPU unit according to the specified parameters.

- The instruction can be executed when the transfer request relays of the FTPc control relay and the FTPc logging/trace control relay are OFF (0: No request). As an execution condition of the instruction, insert a program that checks the state of the transfer request relay. The states of the transfer request relay and the logging transfer request relay can be read with the ETSTAT instruction. Store the state that is read in a device such as an internal relay. An operation error occurs if the instruction is executed when one of the transfer request relays is ON.
- After this instruction is executed, the PLC operates as shown in the following table.

Conditions		Operation
The power supply for the PLC is switched from OFF to ON.		Setting using the configuration
The PLC is changed to PROG mode.	Changes to RUN mode without rewriting the configuration.	Setting using instructions
	Changes to RUN mode after rewriting the configuration.	Setting using the configuration

- If an incorrect IP address is specified, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] to [S3], specify the starting address of the device area that stores the string data that indicates the set parameters, or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different. The number of characters should not exceed 256.
- Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous. "Abcd", "ABCD" and "abcd" are all synonymous.
- This instruction is not available in interrupt programs.

■ Operand [S1] setting

- Specify the starting address storing the server specification parameter or a character constant.
- A part of parameters can be omitted. The settings are not changed when parameters are omitted partially.
- When omitting the part before a specified keyword, omit only "keyword" without omitting ",".
- When omitting the part after a specified keyword, omit both "," and "keyword".
- Specify the FTP server setting from SV0 in order. When the right order is skipped, an error occurs. It is possible to specify when the setting has been already registered.
- Only one server can be specified at the same time.
- Specify an FTP server number, the IP address or host name of an FTP server, a port number, an open method, and the SSL3/TLS1 authentication setting within 256 one-byte characters in total.

- It is prohibited to specify the same keyword redundantly. An error is caused in the case of redundant specification.

Setting item	Settings	
S1	FTP server number (Essential)	Specify FTP servers. Specify the following keywords. SV0: Server 0, SV1: Server 1, SV2: Server 2, SV3: Server 3
	IP address or host name of FTP server (Essential)	Specify IP address or host name. For an IP address, specify the keyword "IPv4=" or "IPv6=" at the beginning. For a host name, specify "HOST=". ● For IPv4: IPv4 = 111.122.133.144 ● For IPv6: IPv6=1111:122:2:1555:0:0:1888 * For details of the range of IPv4 addresses that can be specified, refer to "19.10 Ethernet Function: IP Addresses""IP address setting specifications". ● For a host name: HOST=FTP.pidsx.com
	Port No. (Can be omitted)	Specify port number. Port No. Range: 1 to 65535 PORT=: Port number (Default: 21)
	Open method (Can be omitted)	Specify open method. Active=act/Passive=pasv OPEN=: Open method (Default = act)
	SSL3/TLS1 authentication (can be omitted)	Specify whether or not to use SSL3/TLS1 authentication. SSL: Use SSL3/TLS1 NON: Not use

(Note 1) Input an FTP server number, the IP address or host name of an FTP server, a port number, an open method, and the SSL3/TLS1 authentication setting separated by a comma ",".

(Note 2) Both upper and lower cases can be used for specifying keywords.

(Note 3) Specify the parameters for specifying servers in the order of the above table.

Setting example

Example 1	S1	"SV0,IPv4=192.255.2.10,PORT=21,OPEN=act,SSL"
Settings		FTP server number: 0, IP address: 192.255.2.10, Port number: 21, Open method: Active, SSL3/TLS1 authentication: Use
Example 2	S1	"SV1,IPv6=1111:1222:1555:0:0:1888,SSL"
Settings		FTP server number: 1, IP address: 1111:1222:1555:0:0:1888, Port number: Omitted (Default: 21), Open method: Omitted (Default: Active), SSL3/TLS1 authentication: Use
Example 3	S1	"SV2,HOST=FTP.pidsx.com,PORT=28,OPEN=pasv,NON"
Settings		FTP server number: 2, Host name: FTP.pidsx.com, Port number: 28, Open method: Passive, SSL3/TLS1 authentication: Not use

■ Operand [S2] setting

- Specify the starting address of the device area that stores the string data that indicates parameters, or a character constant.
- A part of parameters can be omitted. The settings are not changed when parameters are omitted partially.

- When omitting the part before a specified keyword, omit only "keyword" without omitting ",".
- When omitting the part after a specified keyword, omit both "," and "keyword".
- When "INITIAL" or "KEEP" is specified instead of parameters, the instruction operates according to the table of special keywords.
- It is prohibited to specify the same keyword redundantly. An error is caused in the case of redundant specification.

Setting item	Settings		Setting range
S2	User name (Can be omitted)	Specify a user name. Specify the keyword “USER=” at the beginning. USER=XXX (Default: root)	Maximum 32 one-byte characters
	Password (Can be omitted)	Specify a password. Specify the keyword “PASS=” at the beginning. PASS=XXX (Default: root)	Maximum 32 one-byte characters

(Note 1) Input a user name and password separated by a comma ",".

(Note 2) Both upper and lower cases can be used for specifying keywords.

(Note 3) Specify the login setting parameters in the order of the above table.

Setting example

Example 1	S2	"USER=root,PASS=pidsx"
Settings		User name: root, Password: pidsx
Example 2	S2	"USER=PANASONIC,PASS=SUNX"
Settings		User name: PANASONIC, Password: SUNX

■ Operand [S2]: user name and password setting

Patterns	How to specify
Specify user name: Delete password	"USER=xxx,PASS="
Delete user name: Specify password	"USER=,PASS=xxx"
Delete user name: Delete password	"USER=,PASS="
Specify user name: Not change password	"USER=xxx"
Not change user name: Specify password	",PASS=xxx"

Setting example

Example 1	S2	"USER=root,PASS="
Settings		User name: root, Password: Delete
Example 2	S2	"USER=,PASS=SUNX"
Settings		User name: Delete, Password: SUNX
Example 3	S2	"USER=,PASS="

Settings		User name: Delete, Password: Delete
Example 4	S2	"USER=root"
Settings		User name: root, Password: Not change
Example 5	S2	",PASS=SUNX"
Settings		User name: Not change, Password: SUNX

■ Special keyword of operand [S2] setting

Special keyword	Description
INITIAL	Set an initial value.
KEEP	The current setting is not changed.

Setting example

Example 1	S2	"INITIAL"
Settings		User name: root, Password: root
Example 2	S2	"KEEP"
Settings		User name: Not change, Password: Not change

■ Operand [S3] setting

- Specify the starting address of the device area that stores the string data that indicates parameters, or a character constant.
- A part of parameters can be omitted. The settings are not changed when parameters are omitted partially.
- When omitting the part before a specified keyword, omit only "keyword" without omitting ",".
- When omitting the part after a specified keyword, omit both "," and "keyword".
- When "INITIAL" or "KEEP" is specified instead of parameters, the instruction operates according to the table of special keywords.
- It is prohibited to specify the same keyword redundantly. An error is caused in the case of redundant specification.

Setting item	Settings		Setting range
S3	Timeout period (Can be omitted)	Specify a timeout period. TOUT=: Time setting (Default: 60 seconds)	30 to 300 seconds
	No. of retries (Can be omitted)	Specify the number of retries. RTRY=: Number of retries (Default: 3 times)	0 to 3
	Retry interval (Can be omitted)	Specify the retry interval. RTTM=: Retry interval (Default: 600 seconds) (Note 4)	10 to 86400 seconds

(Note 1) Input a timeout period, number of retries and retry interval separated by a comma ",".

(Note 2) Both upper and lower cases can be used for specifying keywords.

(Note 3) Specify the detailed setting parameters in the order of the above table.

(Note 4) The retry interval can be specified in 10-second units. It is rounded down to the nearest 10. (Example: When specifying 38 seconds, it becomes 30 seconds.)

Setting example

Exam ple 1	S3	“TOUT=30,RTRY=2,RTTM=500”
Settings		Timeout period: 30 seconds, No. of retries: 2, Retry interval: 500 seconds
Exam ple 2	S3	“TOUT=270,RTRY=0,RTTM=4900”
Settings		Timeout period: 270 seconds, No. of retries: 0 (Not retry), Retry interval: 4900 seconds
Exam ple 3	S3	“TOUT=30,RTRY=25”
Settings		Timeout period: 30 seconds, No. of retries: 25, Retry interval: Not change
Exam ple 4	S3	“,RTRY=25,RTTM=3000”
Settings		Timeout period: Not change, No. of retries: 25, Retry interval: 3000 seconds

■ Special keyword of operand [S3] setting

Special keyword	Description
INITIAL	Set an initial value.
KEEP	The current setting is not changed.

Setting example

Exam ple 1	S3	“INITIAL”
Settings		Timeout period: 60 seconds, Number of retries: 3, Retry interval: 600 seconds
Exam ple 2	S3	“KEEP”
Settings		Timeout period: Not change, Number of retries: Not change, Retry interval: Not change

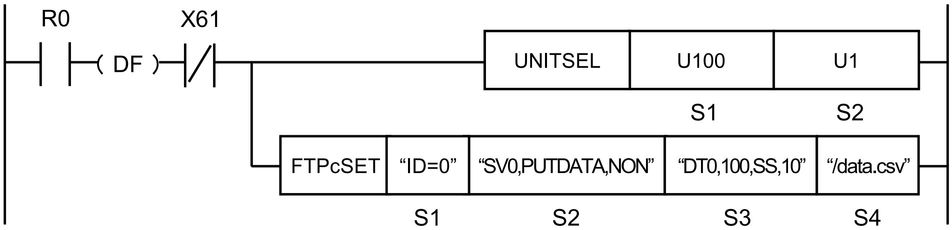
■ Flag operations

Name	Description
SR7 SR8 (ER)	Set when a value outside the range is specified for the parameter. Set when the same keyword is specified redundantly. To be set when even one request active relay of FTPc control relay or FTPc logging/trace control relay is 1: Requesting. To be set when "Add-on" is set to "Not use" in Built-in ET-LAN setting. To be set when server numbers are not specified in the right order. To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). Set when executed in an interrupt program. Set when the number of characters for operand specifying character constant exceeds 256.
CY (SR9)	Set when the instruction is executed while the specified IP address is incorrect. The detail code set in SD29 is "1: Specification of incorrect IP address". To be set when executed during the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.12 FTPcSET (FTP Client Transfer Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

FTPcSET "ID=0" "SV0,PUTDATA,NON" "DT0,100,SS,10" "/data.csv"

■ List of operands

Operand	Description	
S1	Starting address of the device area that stores the string data that indicates a transfer setting number, or a character constant.	
S2	Starting address of the device area that stores the string data that indicates the parameters for operation setting, or a character constant.	
S3	File transfer:	Starting address of the device area that stores the string data that indicates a source file name, or a character constant.
	PUT operation for a device:	Starting address of the device area that stores the string data that indicates source device settings, or a character constant.
	GET operation for a device:	Starting address of the device area that stores the string data that indicates destination device settings, or a character constant.
S4	File transfer:	Starting address of the device area that stores the string data that indicates a destination folder name, or a character constant.
	PUT operation for a device:	Starting address of the device area that stores the string data that indicates destination file settings, or a character constant.
	GET operation for a device:	Starting address of the device area that stores the string data that indicates a source file name, or a character constant.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●			●	●												●	

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S3	●	●	●	●			●	●												●	
S4	●	●	●	●			●	●												●	

Outline of operation

- This instruction configures the FTP client transfer settings (0 to 15).
- Before executing this instruction, use the "17.11 FTPcSV (FTP Client Connected Server Setting)" instruction or the programming tool software "FPWIN GR7" to configure the settings of the destination server.

Processing

- The FTP client transfer settings of [S2] to [S4] are stored in the transfer setting area that is specified by [S1].
- The instruction can be executed when the transfer request relays of the FTPc control relay and the FTPc logging/trace control relay are OFF (0: No request). As an execution condition of the instruction, insert a program that checks the state of the transfer request relay. The states of the transfer request relay and the logging transfer request relay can be read with the ETSTAT instruction. Store the state that is read in a device such as an internal relay. An operation error occurs if the instruction is executed when one of the transfer request relays is ON.
- After this instruction is executed, the PLC operates as shown in the following table.

Conditions		Operation
The power supply for the PLC is switched from OFF to ON.		Setting using the configuration
The PLC is changed to PROG mode.	Changes to RUN mode without rewriting the configuration.	Setting using instructions
	Changes to RUN mode after rewriting the configuration.	Setting using the configuration

- Data is actually sent to files or acquired from files when the FTP client transfer request (FTPcREQ) instruction is executed after the completion of the FTP client transfer settings.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] to [S4], specify the starting address of the device area that stores the string data that indicates the set parameters, or a character constant. When specifying a device area for

an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.

- The number of characters should not exceed 256.
- Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous. However, the folder name and the file name that are included in a path name are case-sensitive.
- This instruction is not available in interrupt programs.

■ Operand [S1] setting

- Specify the starting address of the device area that stores the string data that indicates a transfer setting number, or a character constant.

Setting item	Settings		Setting range
S1	Transfer setting number	Specify a transfer setting number. ID=: Transfer setting number	0 to 15

(Note 1) Transfer setting numbers should be specified from number 0 in ascending order. An error occurs when transfer setting numbers are not specified in ascending order. If transfer settings have been already registered, this rule is not applied.

Setting example

Example 1	S1	“ID=1”
Settings		Transfer setting number: 1
Example 2	S1	“ID=8”
Settings		Transfer setting number: 8

■ Operand [S2] setting

- Specify the starting address of the device area that stores the operation setting parameter, or a character constant.

Setting item	Settings			
S2	Specification of FTP server	Specify FTP servers. (3 digits fixed) SV0: Server 0, SV1: Server 1, SV2: Server 2, SV3: Server 3		
	Target and operation of transfer	Set the target for the transfer and operation.		
		Parameter string	Target	Transfer operation
		PUTFILE	File	Send to servers (Overwrite method)
		PUTFILE-OVW	File	Send to servers (Overwrite method)
		PUTFILE-REN	File	Send to servers (Rename method)
		GETFILE	File	Obtain from servers
		PUTDATA	Device	Send to servers (Overwrite method)
		PUTDATA-OVW	Device	Send to servers (Overwrite method)
		PUTDATA-REN	Device	Send to servers (Rename method)
		GETDATA	Device	Obtain from servers

Setting item	Settings	
	File after transfer	Setting for deleting source files after transfer. (3 digits fixed) DEL: Delete, NON: Not deleted

(Note 1) Input each operation setting parameter separated by a comma ",".

(Note 2) The operation setting parameters cannot be omitted.

(Note 3) For details of the transfer operations (overwrite method and rename method), refer to "Overwrite method and rename method" (p."P.17-62").

Setting example

Example 1	S2	"SV3,PUTFILE,NON"
Settings		FTP server: 3, Target: File, Operation: Send (PUT) Overwrite method, File after transfer: Not deleted
Example 2	S2	"SV1,PUTFILE-OVW,DEL"
Settings		FTP server: 1, Target: File, Operation: Send (PUT) Overwrite method, File after transfer: Delete
Example 3	S2	"SV0,PUTFILE-REN,DEL"
Settings		FTP server: 0, Target: File, Operation: Send (PUT) Rename method, File after transfer: Delete
Example 4	S2	"SV2,GETFILE,DEL"
Settings		FTP server: 2, Target: File, Operation: Get (GET), File after transfer: Delete
Example 5	S2	"SV1,GETFILE,NON"
Settings		FTP server: 1, Target: File, Operation: Get (GET), File after transfer: Not deleted

■ Overwrite method and rename method

The overwrite method (default) or rename method can be selected for file transfer (PUTFILE or PUTDATA).

Items	Description
Operation of overwrite method	● Files are written with specified file names. ● When writing is interrupted for some reason (such as trouble with the network or servers), the partially written file remains. ● It is not possible to judge on the server side whether files have been transferred successfully or not without checking the file size or the contents.
Operation of rename method	● Specified data or files are transferred with tentative file names, and they are renamed to specified file names after the successful completion of transfer. ● The successful completion of file transfer can be confirmed by checking the specified file names on the server side. ● The processing time is longer than that of the overwrite method.
Tentative file name	● FP7_MAC address (Hexadecimal 12 characters).tmp (Extension tmp) ● If a file already exists when renaming files, that file is deleted before renaming. ● When retrying the transfer of multiple files, this situation may occur.

Info.

- For transferring files to FTP servers, the overwrite method or rename method is selectable. As tentative file names are renamed after the completion of the transfer in the rename method, it is possible to confirm that the files have reached to FTP servers successfully.

■ Operand [S3] setting (for file transfer)

- Specify the starting address of the device area that stores the string data that indicates a source file name, or a character constant.

<table>
<tr><th>Setting item</th><th colspan="2">Settings</th></tr>
<tr><td rowspan="2">S3</td><td rowspan="2">Source File Name</td><td>For PUT
Specify a file name in an SD memory card with an absolute path.</td></tr>
<tr><td>For GET
Specify a file name from the home directory of a user which logs in FTP servers with a relative path.</td></tr>
</table>

(Note 1) Wild cards "*" and "?" are usable for file names.

(Note 2) An error occurs when the number of files which match wild cards 101 or more.

■ Operand [S3] setting (PUT operation for a device)

- Specify the starting address of the device area that stores the string data that indicates source device settings, or a character constant.

<table>
<tr><th>Setting item</th><th colspan="2">Settings</th><th>Setting range</th></tr>
<tr><td rowspan="4">S3</td><td rowspan="2">Source device</td><td>● Global devices
Specify device code + device number.
● Local devices
"PB" + PB number + "_" (underscore) + device code + device number</td><td rowspan="2"></td></tr>
<tr><td>Devices that can be specified
<table>
<tr><th>Global devices</th><th>Local devices</th></tr>
<tr><td>WX</td><td>WX</td></tr>
<tr><td>WY</td><td>WY</td></tr>
<tr><td>WR</td><td>WR</td></tr>
<tr><td>WL</td><td>WL</td></tr>
<tr><td>DT</td><td>DT</td></tr>
<tr><td>LD</td><td>LD</td></tr>
</table></td></tr>
<tr><td>Number of transferred data
(data amount)</td><td>Specify the number of transferred data (number of data).
* The number of data that can be transferred simultaneously is 1MB for all 16 IDs. However, they are calculated with data after conversion.</td><td>1 to 524228
(512k data)</td></tr>
<tr><td>Conversion method</td><td>Specify a conversion method.</td><td></td></tr>
</table>

Setting item	Settings		Setting range

Parameter		Extension (Saving format)
BIN1w	Unconverted 16-bit binary	.BIN (binary data)
US	16-bit unsigned decimal	.CSV (comma-separated text)
SS	16-bit signed decimal	
UL	32-bit unsigned decimal	
SL	32-bit signed decimal	
SF	32-bit single-precision floating point	
DF	64-bit double-precision floating point	
HEX1w	16bitHEX	
HEX2w	32bitHEX	
HEX4w	64bitHEX	
ASCII	ASCII character (Output enclosed with "")	

Setting item	Settings		Setting range
	Line feed position	Specify line feed position. 0: Output the end of file only n: Output by n data	0 to 255

(Note 1) Input each source device setting parameter separated by a comma ",".

(Note 2) Specify the operation setting parameters in the order of the above table.

(Note 3) When omitting "conversion method" and subsequent items, the conversion method is set to 16-bit binary and the line feed position is set to 0 (output the end of file only).

(Note 4) When omitting "Line feed position", it is set to 0: Output the end of file only.

Setting example

Example 1	S3	"WX16,32,BIN1w,0"
Settings		Device setting, Device division: Global, Device code: WX, Device No.: 16 Number of transferred data: 32 (32 words), Conversion method: Unconverted 16-bit binary, Line feed position: Output the end of file only
Example 2	S3	"DT123456,250,SS,10"
Settings		Device setting, Device division: Global, Device code: DT, Device No.: 123456 No. of transferred data: 250 points (250 words), Conversion method: 16-bit signed decimal, Line feed position: Output by 10 data
Example 3	S3	"WR0,16,DF"
Settings		Device setting, Device division: Global, Device code: WR, Device No.: 0 No. of transferred data: 16 points (64 words), Conversion method: 64-bit double-precision floating point, Line feed position: Output the end of file only
Example 4	S3	"WL10,128"

Settings		Device setting, Device division: Global, Device code: WL, Device No.: 10 Number of transferred data: 128 (128 words), Conversion method: Unconverted 16-bit binary, Line feed position: Output the end of file only
Example 5	S3	"PB100_WR1000,50,US,0"
Settings		Device setting, Device division: Local, PB number: 100, Device code: WR, Device number: 1000 No. of transferred data: 50 points (50 words), Conversion method: 16-bit unsigned decimal, Line feed position: Output the end of file only
Example 6	S3	"PB15_LD16,40,HEX4w,2"
Settings		Device setting, Device division: Local, PB number: 15, Device code: LD, Device number: 16 No. of transferred data: 40 points (160 words), Conversion method: 64-bit HEX, Line feed position: Output by 2 data
Example 7	S3	"PB10_WL10,32,UL"
Settings		Device setting, Device division: Local, PB number: 10, Device code: WL, Device number: 10 No. of transferred data: 32 points (64 words), Conversion method: 32-bit unsigned decimal, Line feed position: Output the end of file only
Example 8	S3	"PB1_WY128,5"
Settings		Device setting, Device division: Local, PB number: 1, Device code: WY, Device number: 128 Number of transferred data: 5 (5 words), Conversion method: Unconverted 16-bit binary, Line feed position: Output the end of file only

■ Operand [S3] setting (when getting device)

- Specify the starting address of the device area that stores the string data that indicates destination device settings, or a character constant.

<table>
<tr><th>Setting item</th><th colspan="2">Settings</th><th>Setting range</th></tr>
<tr><td rowspan="2">S3</td><td>Destination device</td><td>• Global devices
Specify device code + device number.
• Local devices
"PB" + PB number + "_" (underscore) + device code + device number

Devices that can be specified
<table><tr><th>Global devices</th><th>Local devices</th></tr><tr><td>WX</td><td>WX</td></tr><tr><td>WY</td><td>WY</td></tr><tr><td>WR</td><td>WR</td></tr><tr><td>WL</td><td>WL</td></tr><tr><td>DT</td><td>DT</td></tr><tr><td>LD</td><td>LD</td></tr></table></td><td></td></tr>
<tr><td>Number of transferred data (data amount)</td><td>Specify the number of transferred data (number of data).
* The number of data that can be transferred simultaneously is 1MB for all 16 IDs. They are calculated with file size.</td><td>1 to 524228
(512k data)</td></tr>
</table>

<table>
<tr><th>Setting item</th><th colspan="2">Settings</th><th>Setting range</th></tr>
<tr><td></td><td>Conversion method</td><td>Specify a conversion method.
<table>
<tr><th colspan="2">Parameter</th><th>Extension (Saving format)</th></tr>
<tr><td>BIN1w</td><td>Unconverted 16-bit binary</td><td>.BIN (binary data)</td></tr>
<tr><td>US</td><td>16-bit unsigned decimal</td><td rowspan="10">.CSV
(comma-separated text)</td></tr>
<tr><td>SS</td><td>16-bit signed decimal</td></tr>
<tr><td>UL</td><td>32-bit unsigned decimal</td></tr>
<tr><td>SL</td><td>32-bit signed decimal</td></tr>
<tr><td>SF</td><td>32-bit single-precision floating point</td></tr>
<tr><td>DF</td><td>64-bit double-precision floating point</td></tr>
<tr><td>HEX1w</td><td>16bitHEX</td></tr>
<tr><td>HEX2w</td><td>32bitHEX</td></tr>
<tr><td>HEX4w</td><td>64bitHEX</td></tr>
<tr><td>ASCII</td><td>ASCII character (Output enclosed with "")</td></tr>
</table></td><td></td></tr>
</table>

(Note 1) Input each source device setting parameter separated by a comma ",".

(Note 2) Specify the operation setting parameters in the order of the above table.

(Note 3) When omitting "Conversion method", it is set to unconverted 16-bit binary.

(Note 4) The number of data that can be transferred simultaneously is 1MB for all 16 IDs. They are calculated with file size.

Setting example

<table>
<tr><td>Example 1</td><td>S3</td><td>"WX16,32,BIN1w"</td></tr>
<tr><td colspan="2">Settings</td><td>Device setting, Device division: Global, Device code: WX, Device number: 16, Number of transferred data: 32 (32 words), Conversion method: Unconverted 16-bit binary</td></tr>
<tr><td>Example 2</td><td>S3</td><td>"DT123456,250,SS"</td></tr>
<tr><td colspan="2">Settings</td><td>Device setting, Device division: Global, Device code: DT, Device number: 123456, Number of transferred data: 250 (250 words), Conversion method: 16-bit signed decimal</td></tr>
<tr><td>Example 3</td><td>S3</td><td>"WR0,16,DF"</td></tr>
<tr><td colspan="2">Settings</td><td>Device setting, Device division: Global, Device code: WR, Device number: 0, Number of transferred data: 16 (64 words), Conversion method: 64-bit double-precision floating point</td></tr>
<tr><td>Example 4</td><td>S3</td><td>"WL10,128"</td></tr>
<tr><td colspan="2">Settings</td><td>Device setting, Device division: Global, Device code: WL, Device number: 10 Number of transferred data: 128 (128 words), Conversion method: Unconverted 16-bit binary</td></tr>
</table>

■ Operand [S4] setting (for file transfer)

● Specify the starting address of the device area that stores the string data that indicates a destination folder name, or a character constant.

Setting item	Settings		
S4	Destination file name	For PUT	Specify a folder name from the home directory of a user which logs in FTP servers with a relative path. For specifying the home directory, specify "/" or "\" only. Note) When an English keyboard is used, use "\" instead of "¥".
		For GET	Specify a storage folder name in an SD memory card with an absolute path.

(Note 1) When the specified destination folder does not exist, the folder is automatically created with up to eight hierarchies.

■ Operand [S4] setting (PUT operation for a device)

● Specify the starting address of the device area that stores the string data that indicates destination file settings, or a character constant.

Setting item	Settings	
S4	Destination file name	Specify a destination file name. Specify a folder name and file name with its relative path from the home directory of the user who logs in to the FTP server. * The string after the last "." (period) is applied as the extension of the file name.
	File name Automatic addition position	Specify the position of the automatic additional data added to a file name. TOP: Automatic additional data is added before a file name. END: Automatic additional data is added after a file name. * Automatic additional data is year, month, day, hour, minute and second "(yymmdd_hhmmss)".

(Note 1) Specify a destination file name within 240 characters.

(Note 2) When the specified destination folder does not exist, the folder is automatically created with up to eight hierarchies.

(Note 3) Specify the operation setting parameters in the order of the above table.

(Note 4) When omitting "File name automatic addition position", automatic additional data is not added to the file name.

Setting example

Example 1	S4	"\FTP\PutData1.bin,TOP"
Settings		Destination file name: \FTP\PutData1.bin Automatic additional data: year, month, day, hour, minute and second "(yymmdd_hhmmss)" Automatic addition position: Automatic additional data is added before the file name.
Example 2	S4	"\FTP\PutData2.bin,END"
Settings		Destination file name: \FTP\PutData2.bin Automatic additional data: year, month, day, hour, minute and second "(yymmdd_hhmmss)" Automatic addition position: Automatic additional data is added after the file name.

Example 3	S4	"\FTP\PutData3.bin"
Settings		Destination file name: \FTP\PutData3.bin Automatic addition position: Automatic additional data is not added to the file name.

■ Operand [S4] setting (when getting device)

Setting item	Settings	
S4	Source File Name	Specify the starting address of the device area that stores the string data that indicates a source file name, or a character constant.

(Note 1) Specify a folder name and file name with its relative path from the home directory of the user who logs in to the FTP server.

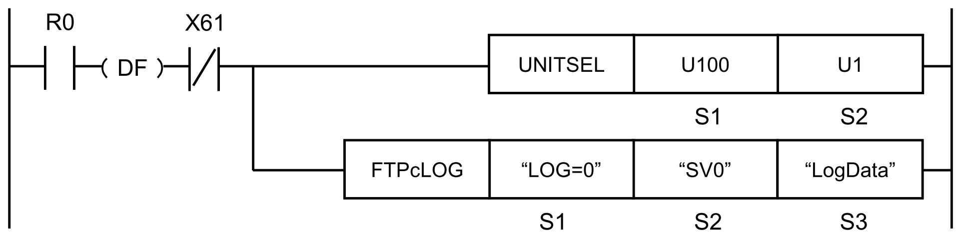
■ Flag operations

Name	Description
SR7 SR8 (ER)	Set when a value outside the range is specified for the parameter. To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). To be set when transfer setting numbers are not specified in ascending order. Set when executed in an interrupt program. Set when the number of characters for operand specifying character constant exceeds 256. To be set when an FTP server that has not been specified with the destination server setting instruction or the tool software is specified.
CY (SR9)	To be set when executed during the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.13 FTPcLOG (FTP Client Logging/Trace Transfer Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

FTPcLOG "LOG=0" "SV0" "LogData"

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates a logging/trace number, or a character constant.
S2	Starting address of the device area that stores the string data that indicates the parameters for operation setting, or a character constant.
S3	Starting address of the device area that stores the string data that indicates a destination folder name, or a character constant.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●			●	●												●	
S3	●	●	●	●			●	●												●	

■ Outline of operation

- This instruction configures the logging/trace transfer setting.

■ Processing

- The logging/trace transfer settings of [S2] to [S3] are stored in the logging/trace transfer setting area that is specified by [S1].

- The instruction can be executed when the transfer request relay of the FTPc logging/trace control relay is OFF (0: No request). As an execution condition of the instruction, insert a program that checks the state of the transfer request relay. The state of the FTPc logging transfer request relay can be read with the ETSTAT instruction. Store the state that is read in a device such as an internal relay. An operation error occurs if this instruction is executed when the transfer request relay is ON.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] to [S3], specify the starting address of the device area that stores the string data that indicates the set parameters, or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous. However, the destination folder name is case-sensitive.
- This instruction is not available in interrupt programs.

■ Operand [S1] setting

Specify the starting address of the device area that stores the string data that indicates a logging/trace number, or a character constant.

Setting item	Settings
S1	Specify a LOG number (0 to 15) as string data. Example) "LOG=0"

■ Operand [S2] setting

- Specify the starting address of the device area that stores the string data that indicates the parameters for operation setting, or a character constant.
- Only one server can be specified at the same time. Specify an FTP server number with one-byte three characters.

Setting item	Settings		
S2	Specification of an FTP server (essential)		Specify an FTP server (server 0 to 3) as string data. Example) "SV0
	Specification of transfer operation (Can be omitted)		Select the operation for transferring logging/trace files. Specify the operation after the keyword "MODE=". When either method is not specified, "Overwrite method" is applied. MODE=OVW or MODE=REN
		OVW Overwrite method (Default)	Performs transfer files with file names specified by the logging/trace setting. When the transfer is interrupted due to any trouble with the network or the server, the file transferred partway remains in the server. Confirm if the transfer has succeeded with an instruction such as ETSTAT instruction.

Setting item	Settings		
		REN Rename method	Performs transfer files with temporary file names, and renames them to specified file names after the success of the transfer. The successful completion of file transfer can be confirmed by checking the file names specified by the logging/trace setting. The processing time is longer than that of the overwrite method.

(Note 1) Input each operation setting parameter separated by a comma ",".

(Note 2) Specify the operation setting parameters in the order of the above table. The order of keywords cannot be changed.

(Note 3) Upper and lower case characters can be used for specifying keywords.

Setting example

Example 1	S2	"SV0,MODE=OVW"
Settings		FTP server: 0, Transfer operation: Overwrite method
Example 2	S2	"SV3,MODE=REN"
Settings		FTP server: 2, Transfer operation: Rename method
Example 3	S2	"SV3"
Settings		FTP server: 3, Transfer operation: (Omitted)

■ Operand [S3] setting

- Specify the starting address of the device area that stores the string data that indicates a destination folder name, or a character constant.
- A destination folder name should be specified within 256 one-byte characters.

Setting item	Settings		Setting range
S3	Destination folder name	Specify the starting address of the device area that stores the string data that indicates a destination folder name, or a character constant.	Maximum 256 one-byte characters

(Note 1) When the specified destination folder does not exist, the folder is automatically created with up to eight hierarchies.

(Note 2) Specify a folder name from the home directory of a user which logs in FTP servers with a relative path.

■ Flag operations

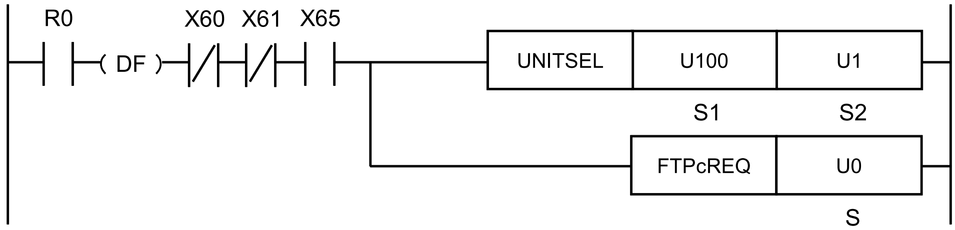
Name	Description
SR7 SR8 (ER)	To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). To be set when the request active relay of the FTPc logging/trace control relay for a specified number is 1: Requesting. To be set when the logging/trace condition of a specified LOGn number is not registered. To be set when an out-of-range value is specified for parameters. To be set when executed in an interrupt program. Set when the number of characters for operand specifying character constant exceeds 256. To be set when an unset FTP server is specified.

Name	Description
CY (SR9)	Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.14 FTPcREQ (FTP Client Transfer Request)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Device address where the transfer number (0 to 15) is stored, or a constant.

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	•	•	•	•			•	•								•	•				•

■ Outline of operation

- This instruction requests the transfer of the FTP client.
- Before executing this instruction, use the "17.12 FTPcSET (FTP Client Transfer Setting)" instruction or the programming tool software "FPWIN GR7" to configure HTTP transfer settings.

■ Operand [S] setting

Setting item	Settings		Setting range
S	Transfer number	Specify the device address storing a transfer number or a constant.	0 to 15

■ Processing

- The transfer request relay of the transfer number that is specified by [S] is turned ON.
- This instruction can be executed when the FTP client preparation done flag (X65) is ON. As an execution condition of the instruction, insert a program that checks the status of the flag (X65). An operation error occurs if this instruction is executed when the flag (X65) is OFF.

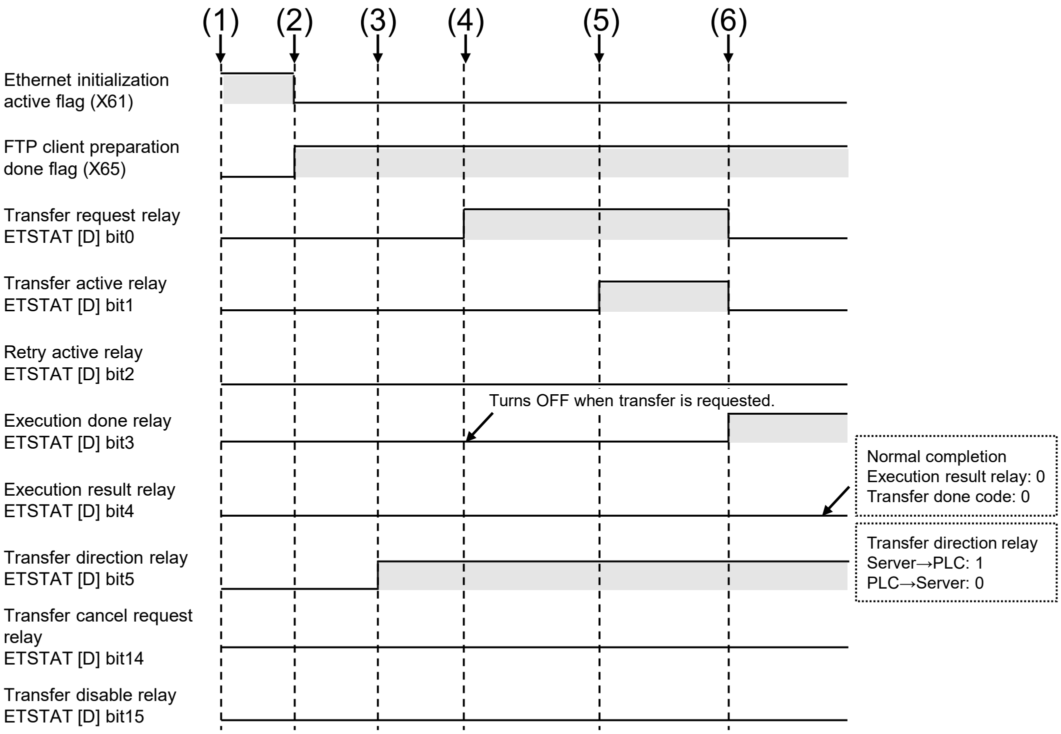
- This instruction can be executed when the cable disconnection detection flag (X60) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X60). If this instruction is executed when the flag (X60) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- This instruction is not available in interrupt programs.

■ Timing chart

- The following diagram shows the process in which a transfer request is executed and data transfer from a server to FP7 is completed successfully.
- The control relays (bit0 to bit15) can be monitored by using the ETSTAT instruction to read and store their state in arbitrary operation devices.



(1)	RUN (Power on)	(4)	Transfer request (Executes FTPcREQ instruction)
(2)	FTP client preparation done	(5)	FTP client login succeeded (Starts transfer)

(3)	Transfer setting (Executes FTPcSET instruction)	(6)	Transfer process done (Completes the execution of FTPcREQ instruction)

■ Control relay

Name	Bit No.	Description
Transfer request relay	0	0: No request, 1: Request
Transfer active relay	1	0: Stop, 1: During transfer
Retry active relay	2	0: No retry, 1: During retry
Execution done relay	3	0: During process, 1: Instruction execution complete
Execution result relay	4	0: Normal 1: Failed
Transfer direction relay	5	0: Send, 1: Receive
Reserved for system	6 to 13	-
Transfer cancel request relay	14	0: Not cancel, 1: Cancel
Transfer disable relay	15	0: Transfer enabled, 1: Transfer disabled

(Note 1) The state of control relays can be read with ETSTAT instruction.

■ Done codes

Name	Number of words	Description
Execution done code	1	Execution done code
Transfer done code	1	Response code of FTP client

(Note 1) The state of completion codes can be read with ETSTAT instruction.

When the instruction is executed under one of the following conditions, a transfer error occurs and the corresponding error code is set in the execution done code.

Status	Code	Status	Code
Destination server is not set.	1	Transfer prohibition setting	5
Transfer setting is not set.	2	Data decompression failed. (When accessing data with PUT)	8
Registering a process request failed.	4	Data decompression failed. (When accessing data with GET)	9

■ FTP client preparation done (WX6 bit 5)

Name	Bit No.	Description
FTP client preparation done (X65)	5	0: FTP client preparation incomplete, 1: FTP client preparation complete

(Note 1) For details of Ethernet-related flags, refer to "19.10 Ethernet Function: IP Addresses".

■ Flag operations

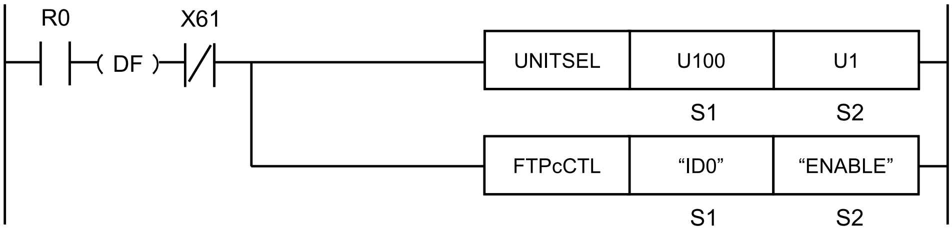
Name	Description
SR7	To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN).
SR8	To be set in the case of out-of-range in indirect access (index modification).

Name	Description
(ER)	To be set when the FTP client preparation done (X65) is OFF at the time of the execution of instruction. Set when a value outside the range is specified for the parameter. To be set when the transfer request relay of a specified ID is "Request". Set when executed in an interrupt program. To be set when a file transfer that has not been specified with the transfer setting instruction or the tool software is specified.
CY (SR9)	To be set when executed while the Ethernet cable is disconnected. The detail code set in SD29 is "10: Ethernet cable disconnected". To be set when executed during the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.15 FTPcCTL (FTP Client Transfer Control)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

FTPcCTL "ID0" "ENABLE"

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates a control target, or a character constant.
S2	Starting address of the device area that stores the string data that indicates the control content (transfer enabled/disabled/canceled), or a character constant.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●			●	●												●	

■ Outline of operation

- This instruction configures the settings for enabling, disabling, or canceling transfers for an FTP client.
- Before executing this instruction, use the "FTPcSET (FTP Client Transfer Setting)" instruction or the programming tool software "FPWIN GR7" to configure transfer settings. (when control targets are specified with send numbers)
- Before executing this instruction, use the "17.13 FTPcLOG (FTP Client Logging/Trace Transfer Setting)" instruction or the programming tool software "FPWIN GR7" to configure transfer settings. (when control targets are specified with LOG numbers)
- It takes some time to accept the processing of the transfer cancel request. After executing the instruction, check the transfer status to see if the transfer stops. For details on checking

the transfer status, refer to the "17.3 ETSTAT (Acquiring Ethernet Unit Information: FTP/HTTP/SMTP)" instruction.

■ Processing

- The instruction controls whether to enable, disable, or cancel transfer for the target [S1] according to the specification of the control content [S2].
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] and [S2], specify the starting address of the device area that stores the string data that indicates the set parameters, or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous.
- This instruction is not available in interrupt programs.

■ Setting of the operands [S1] and [S2]

Setting item	Settings		
S1	Control target	1) When specifying an individual transfer number	Specify "IDx" with x being a value from 0 to 15.
		2) When specifying an individual LOG number	Specify "LOGx" with x being a value from 0 to 15.
		3) When specifying all transfer numbers and all LOG numbers	Specify "ALL".
S2	Control content	1) When enabling transfer	Specify "ENABLE".
		2) When disabling transfer	Specify "DISABLE".
		3) When canceling transfer	Specify "CANCEL".

Setting example

<table>
<tr><th></th><th>Settings</th><th>S1</th><th>S2</th></tr>
<tr><td>Example 1</td><td>When enabling the sending of send number 5</td><td>"ID5"</td><td>"ENABLE"</td></tr>
<tr><td>Example 2</td><td>When disabling all sending</td><td>"ALL"</td><td>"DISABLE"</td></tr>
<tr><td>Example 3</td><td>When canceling the transfer of LOG7</td><td>"LOG7"</td><td>"CANCEL"</td></tr>
<tr><td>Example 4</td><td>When enabling the sending of send number 10 (Note 1)</td><td>DT0<table>
<tr><th></th><th colspan="2">Value</th></tr>
<tr><td>DT0</td><td colspan="2">4 (No. of characters)</td></tr>
<tr><td>DT1</td><td>H44(D)</td><td>H49(I)</td></tr>
<tr><td>DT2</td><td>H30(0)</td><td>H31(1)</td></tr>
<tr><td>DT3</td><td></td><td></td></tr>
</table></td><td>DT10<table>
<tr><th></th><th colspan="2">Value</th></tr>
<tr><td>DT10</td><td colspan="2">6 (No. of characters)</td></tr>
<tr><td>DT11</td><td>H4E(N)</td><td>H45(E)</td></tr>
<tr><td>DT12</td><td>H42(B)</td><td>H41(A)</td></tr>
<tr><td>DT13</td><td>H45(E)</td><td>H4C(L)</td></tr>
<tr><td>DT14</td><td></td><td></td></tr>
</table></td></tr>
</table>

(Note 1) For specifying a device for an operand which can specify character constants, store string data with SSET instruction excluding a double quotation mark.

■ Operation of FTPc control relay

	Name	Transfer enabled	Transfer disabled	Transfer canceled
ETSTAT [D] bit0	Transfer request	Not change	Not change	Not change
ETSTAT [D] bit1	Transfer active	Not change	Not change	Not change
ETSTAT [D] bit2	Transfer retry active	Not change	Not change	Not change
ETSTAT [D] bit3	Transfer done	Not change	Not change	Not change
ETSTAT [D] bit4	Transfer failed	Not change	Not change	Not change
ETSTAT [D] bit5	Transfer direction	Not change	Not change	Not change
-----				
ETSTAT [D] bit14	Transfer cancel relay	Not change	Not change	ON

	Name	Transfer enabled	Transfer disabled	Transfer canceled
ETSTAT [D] bit15	Transfer disable relay	OFF	ON	Not change

(Note 1) The states of control relays can be checked by using the ETSTAT instruction to read and store the state in any operation memory.

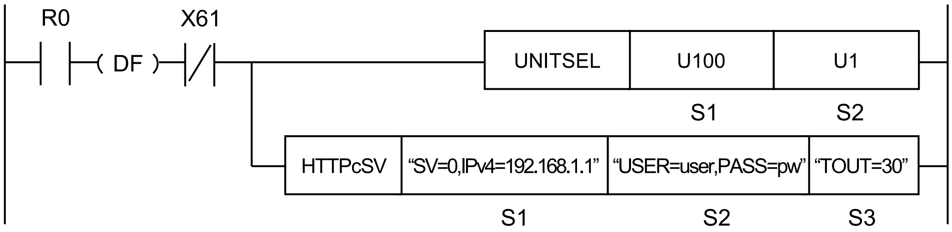
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when an item other than "IDx" or "LOGx" or "ALL" is specified for the control target (S1). (x: 0 to 15) To be set when an unset transfer setting is specified. To be set when an unset logging/trace transfer setting is specified. To be set when an item other than "ENABLE", "DISABLE" or "CANCEL" is specified for the control content (S2). To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). Set when executed in an interrupt program. Set when the number of characters for operand specifying character constant exceeds 256. To be set when a file transfer that has not been specified with the transfer setting instruction or the tool software is specified. To be set when a logging/trace transfer setting that has not been specified with the logging/trace transfer setting instruction or the tool software is specified.
CY (SR9)	To be set when executed during the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.16 HTTPcSV (HTTP Client Connected Server Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.
HTTPcSV "SV=0,IPv4=192.168.1.1" "USER=user,PASS=pw" "TOUT=30"

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates the parameters for specifying a server, or a character constant.
S2	Starting address of the device area that stores the string data that indicates the login setting parameters, or a character constant.
S3	Starting address of the device area that stores the string data that indicates the detailed setting parameters, or a character constant.

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	•	•	•	•			•	•												•	
S2	•	•	•	•			•	•												•	
S3	•	•	•	•			•	•												•	

■ Outline of operation

This instruction configures the settings for a server to which the FP7 CPU unit is connected using the HTTP client function.

■ Processing

- The settings for the server to which the CPU unit is connected using the HTTP client function are specified in the CPU unit according to specified parameters.

- The instruction can be executed when the transfer request relays of the HTTPc control relay and the HTTPc logging/trace control relay are OFF (0: No request). As an execution condition of the instruction, insert a program that checks the state of the transfer request relay. The states of the transfer request relay and the logging transfer request relay can be read with the ETSTAT instruction. Store the state that is read in a device such as an internal relay. An operation error occurs if the instruction is executed when one of the transfer request relays is ON.
- After this instruction is executed, the PLC operates as shown in the following table.

Conditions		Operation
The power supply for the PLC is switched from OFF to ON.		Setting using the configuration
The PLC is changed to PROG mode.	Changes to RUN mode without rewriting the configuration.	Setting using instructions
	Changes to RUN mode after rewriting the configuration.	Setting using the configuration

- If an incorrect IP address is specified, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] to [S3], specify the starting address of the device area that stores the string data that indicates the set parameters, or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- The number of characters should not exceed 256.
- Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous.
- This instruction is not available in interrupt programs.

■ Operand [S1] setting

- Specify the starting address storing the server specification parameter or a character constant.
- A part of parameters can be omitted. The settings are not changed when parameters are omitted partially.
- When omitting the part before a specified keyword, omit only "keyword" without omitting ",".
- When omitting the part after a specified keyword, omit both "," and "keyword".
- Specify the HTTP server setting from SV0 in order. When the right order is skipped, an error occurs. It is possible to specify by overwriting when the setting has been already registered.
- Only one server can be specified at the same time.
- Specify an HTTP server number, the IP address or host name of an HTTP server, a port number, an open method, and the SSL3/TLS1 authentication setting within 256 one-byte characters in total.

- It is prohibited to specify the same keyword redundantly. An error is caused in the case of redundant specification.

Setting item	Settings	
S1	HTTP server number (Essential)	Specify HTTP servers. Specify the following keywords. SV0: Server 0, SV1: Server 1, SV2: Server 2, SV3: Server 3
	IP address or host name of HTTP server (Essential)	Specify IP address or host name. For an IP address, specify the keyword "IPv4=" or "IPv6=" at the beginning. For a host name, specify "HOST=". ● For IPv4 IPv4=111.122.133.144 ● For IPv6 IPv6=1111:122:2:1555:0:0:1888 * For details of the range of IPv4 addresses that can be specified, refer to "19.10 Ethernet Function: IP Addresses" "IP address setting specifications". ● For a host name: HOST=HTTP.pidsx.com
	Port No. (Can be omitted)	Specify port number. Port No. Range: 1 to 65535 PORT=: Port number (Default: 80)
	SSL3/TLS1 Authentication (Can be omitted)	Specify whether or not to use SSL3/TLS1 authentication. SSL: Use SSL3/TLS1 NON: Not use (Default: Not use)

(Note 1) Input an HTTP server number, the IP address or host name of an HTTP server, a port number, and the SSL3/TLS1 authentication setting separated by a comma ",".

(Note 2) Both upper and lower cases can be used for specifying keywords.

(Note 3) Specify the parameters for specifying servers in the order of the above table.

Setting example

Example 1	S1	"SV0,IPv4=192.255.2.10,PORT=80,SSL"
Settings		HTTP server number: 0, IP address: 192.255.2.10, Port number: 80, SSL3/TLS1 authentication: Use
Example 2	S1	"SV1,IPv6=1111:1222::1555:0:0:1888,PORT=8080,SSL"
Settings		HTTP server number: 1, IP address: 1111:1222::1555:0:0:1888, Port number: 8080, SSL3/TLS1 authentication: Use
Example 3	S1	"SV2,HOST=HTTP.pidsx.com,PORT=80,NON"
Settings		HTTP server number: 2, Hos name: HTTP.pidsx.com, Port number: 80, SSL3/TLS1 authentication: Not use

■ Operand [S2] setting

- Specify the starting address of the device area that stores the login setting parameter, or a character constant.

- A part of parameters can be omitted. The settings are not changed when parameters are omitted partially.
- When omitting the part before a specified keyword, omit only "keyword" without omitting ",".
- When omitting the part after a specified keyword, omit both "," and "keyword".
- When "INITIAL" or "KEEP" is specified instead of parameters, the instruction operates according to the table of special keywords.
- It is prohibited to specify the same keyword redundantly. An error is caused in the case of redundant specification.

Setting item	Settings		Setting range
S2	User name (Can be omitted)	Specify a user name. Specify the keyword “USER=” at the beginning. USER=XXX (Default: root)	Maximum 32 one-byte characters
	Password (Can be omitted)	Specify a password. Specify the keyword “PASS=” at the beginning. PASS=XXX (Default: root)	Maximum 32 one-byte characters

(Note 1) Input a user name and password separated by a comma ",".

(Note 2) Both upper and lower cases can be used for specifying keywords.

(Note 3) Specify the login setting parameters in the order of the above table.

Setting example

Example 1	S2	"USER=root,PASS=pidsx"
Settings		User name: root, Password: pidsx
Example 2	S2	"USER=PANASONIC,PASS=SUNX"
Settings		User name: PANASONIC, Password: SUNX

■ Operand [S2]: user name and password setting

Patterns	How to specify
Specify user name: Delete password	"USER=xxx,PASS="
Delete user name: Specify password	"USER=,PASS=xxx"
Delete user name: Delete password	"USER=,PASS="
Specify user name: Not change password	"USER=xxx"
Not change user name: Specify password	",PASS=xxx"

Setting example

Example 1	S2	"USER=root,PASS="
Settings		User name: root, Password: Delete
Example 2	S2	"USER=,PASS=SUNX"
Settings		User name: Delete, Password: SUNX

Example 3	S2	"USER=,PASS="
Settings		User name: Delete, Password: Delete
Example 4	S2	"USER=root"
Settings		User name: root, Password: Not change
Example 5	S2	",PASS=SUNX"
Settings		User name: Not change, Password: SUNX

■ Special keyword of operand [S2] setting

Special keyword	Description
INITIAL	Set an initial value.
KEEP	The current setting is not changed.

Setting example

Example 1	S2	“INITIAL”
Settings		User name: root, Password: root
Example 2	S2	“KEEP”
Settings		User name: Not change, Password: Not change

■ Operand [S3] setting

Setting item	Settings		Setting range
S3	Timeout period (Can be omitted)	Specify a timeout period. TOUT=: Time setting (Default: 60 seconds)	30 to 300 seconds
	No. of retries (Can be omitted)	Specify the number of retries. RTRY=: Number of retries (Default: 3 times)	0 to 3
	Retry interval (Can be omitted)	Specify the retry interval. RTTM=: Retry interval (Default: 600 seconds)	10 to 86400 seconds

(Note 1) Input a timeout period, number of retries and retry interval separated by a comma ",".

(Note 2) Both upper and lower cases can be used for specifying keywords.

(Note 3) Specify the detailed setting parameters in the order of the above table.

(Note 4) The retry interval can be specified in 10-second units. It is rounded down to the nearest 10. (Example: When specifying 38 seconds, it becomes 30 seconds.)

Setting example

Example 1	S3	“TOUT=30,RTRY=2,RTTM=500”
Settings		Timeout period: 30 seconds, No. of retries: 2, Retry interval: 500 seconds
Example 2	S3	“TOUT=270,RTRY=0,RTTM=4900”

Settings		Timeout period: 270 seconds, No. of retries: 0 (Not retry), Retry interval: 4900 seconds
Example 3	S3	“TOUT=120,RTRY=3”
Settings		Timeout period: 120 seconds, No. of retries: 3, Retry interval: Not change

■ Special keyword of operand [S3] setting

Special keyword	Description
INITIAL	Set an initial value.
KEEP	The existing state is held and the setting is not changed.

Setting example

Example 1	S3	“INITIAL”
Settings		Timeout period: 60 seconds, Number of retries: 3, Retry interval: 600 seconds
Example 2	S3	“KEEP”
Settings		Timeout period: Not change, Number of retries: Not change, Retry interval: Not change

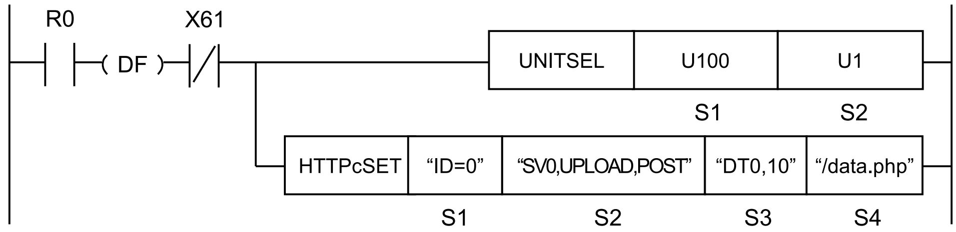
■ Flag operations

Name	Description
SR7 SR8 (ER)	Set when a value outside the range is specified for the parameter. Set when the same keyword is specified redundantly. To be set when the transfer request relay of HTTPc transfer control relay is 1: Requesting. To be set when "Add-on" is set to "Not use" in Built-in ET-LAN setting. To be set when server numbers are not specified in the right order. To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). To be set when executed in an interrupt program. Set when the number of characters for operand specifying character constant exceeds 256.
CY (SR9)	Set when the instruction is executed while the specified IP address is incorrect. The detail code set in SD29 is "1: Specification of incorrect IP address". Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.17 HTTPcSET (HTTP Client Transfer Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

HTTPcSET "ID=0" "SV0,UPLOAD,POST" "DT0,10" "/data.csv"

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates a transfer setting number, or a character constant.
S2	Starting address of the device area that stores the string data that indicates the parameters for operation setting, or a character constant.
S3	Starting address of the device area that stores the string data that indicates source device settings, or a character constant.
S4	Starting address of the device area that stores the string data that indicates a destination URL, or a character constant.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●			●	●												●	
S3	●	●	●	●			●	●												●	
S4	●	●	●	●			●	●												●	

■ Outline of operation

- This instruction configures the HTTP client transfer settings (0 to 15).

- Before executing this instruction, use the "17.16 HTTPcSV (HTTP Client Connected Server Setting)" instruction or the programming tool software "FPWIN GR7" to configure the settings of the destination server.

■ Processing

- The HTTP client transfer settings of [S2] to [S4] are stored in the transfer setting area that is specified by [S1].
- The instruction can be executed when the transfer request relays of the HTTPc control relay and the HTTPc logging/trace control relay are OFF (0: No request). As an execution condition of the instruction, insert a program that checks the state of the transfer request relay. The states of the transfer request relay and the logging transfer request relay can be read with the ETSTAT instruction. Store the state that is read in a device such as an internal relay. An operation error occurs if the instruction is executed when one of the transfer request relays is ON.
- After this instruction is executed, the PLC operates as shown in the following table.

Conditions		Operation
The power supply for the PLC is switched from OFF to ON.		Setting using the configuration
The PLC is changed to PROG mode.	Changes to RUN mode without rewriting the configuration.	Setting using instructions
	Changes to RUN mode after rewriting the configuration.	Setting using the configuration

- Data is actually sent or acquired when the HTTP client transfer request (HTTPcREQ) instruction is executed after the completion of the HTTP client transfer settings.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] to [S4], specify the starting address of the device area that stores the string data that indicates the set parameters, or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- The number of characters should not exceed 256.
- Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous. However, the folder name and the file name that are included in a path name are case-sensitive.
- This instruction is not available in interrupt programs.

■ Operand [S1] setting

- Specify the starting address of the device area that stores the string data that indicates a transfer setting number, or a character constant.

Setting item	Settings		Setting range
S1	Transfer setting number	Specify a transfer setting number. ID=: Transfer setting number	0 to 15

(Note 1) Transfer setting numbers should be specified from number 0 in ascending order. An error occurs when transfer setting numbers are not specified in ascending order. If transfer settings have been already registered, this rule is not applied.

Setting example

Example 1	S1	“ID=1”
Settings		Transfer setting number: 1
Example 2	S1	“ID=8”
Settings		Transfer setting number: 8

■ Operand [S2] setting

- Specify the starting address of the device area that stores the string data that indicates the parameters for operation setting, or a character constant.

<table>
<tr><th>Setting item</th><th colspan="3">Settings</th></tr>
<tr><td rowspan="6">S2</td><td>Specification of HTTP server</td><td colspan="2">Specify HTTP servers. (3 digits fixed)
SV0: Server 0
SV1: Server 1
SV2: Server 2
SV3: Server 3</td></tr>
<tr><td rowspan="4">Target and operation of transfer</td><td colspan="2">Specify the target (device) and operation (Send or Get) of transfer.</td></tr>
<tr><td>UPLOAD</td><td>The target is Device, and the operation is Send.</td></tr>
<tr><td>DOWNLOAD</td><td>The target is Device, and the operation is Get.</td></tr>
<tr><td>UPDOWN</td><td>The target is Device, and the operation is Send and Get.</td></tr>
<tr><td>Command used</td><td colspan="2">Specify a command to be used for transfer.
POST: POST command is used
GET: GET command is used
* Only POST can be specified for Upload or Upload and Download.</td></tr>
</table>

(Note 1) Input each operation setting parameter separated by a comma ",".

(Note 2) The operation setting parameters cannot be omitted.

Setting example

Example 1	S2	"SV3,UPLOAD,POST"
Settings		HTTP server: 3, Target: Device, Operation: Send (UPLOAD), Command used: POST (Fixed)
Example 2	S2	"SV0,UPLOAD,POST"
Settings		HTTP server: 0, Target: Device, Operation: Send (UPLOAD), Command used: POST (Fixed)

■ Operand [S3] setting (UPLOAD operation for a device)

- Specify the starting address of the device area that stores the string data that indicates source device settings, or a character constant.

<table>
<tr><th>Setting item</th><th colspan="2">Settings</th><th>Setting range</th></tr>
<tr><td rowspan="3">S3</td><td rowspan="2">Source device setting</td><td>Specify the source device setting.
● Global devices
Specify device code + device number.
● Local devices
"PB" + PB number + "_" (underscore) + device code + device number</td><td rowspan="2"></td></tr>
<tr><td>Devices that can be specified<table><tr><th>Global devices</th><th>Local devices</th></tr><tr><td>WX</td><td>WX</td></tr><tr><td>WY</td><td>WY</td></tr><tr><td>WR</td><td>WR</td></tr><tr><td>WL</td><td>WL</td></tr><tr><td>DT</td><td>DT</td></tr><tr><td>LD</td><td>LD</td></tr></table></td></tr>
<tr><td>Number of transferred data (number of bytes)</td><td>Specify the number of transferred data (number of bytes). (1 to 7 digits)
* The number of bytes that can be simultaneously transferred is 1 MB for all 16 IDs.</td><td>1 to 1048576 (1MB)</td></tr>
</table>

(Note 1) Input each source device setting parameter separated by a comma ",".

(Note 2) Specify the operation setting parameters in the order of the above table.

Setting example

Example 1	S3	"WX16,32"
Settings		Device setting, Device division: Global, Device code: WX, Device number: 16, Number of bytes: 32 bytes
Example 2	S3	"DT123456,250"
Settings		Device setting, Device division: Global, Device code: DT, Device number: 123456, Number of bytes: 250 bytes
Example 3	S3	"WR0,64"
Settings		Device setting, Device division: Global, Device code: WR, Device number: 0, Number of bytes: 64 bytes
Example 4	S3	"WL10,128"
Settings		Device setting, Device division: Global, Device code: WL, Device number: 10, Number of bytes: 128 bytes

■ Operand [S3] setting (DOWNLOAD operation for a device)

- Specify the starting address of the device area that stores the string data that indicates destination device settings, or a character constant.

<table>
<tr><th>Setting item</th><th colspan="2">Settings</th><th>Setting range</th></tr>
<tr><td rowspan="3">S3</td><td rowspan="2">Destination device setting</td><td>Specify the destination device setting.
● Global devices
Specify device code + device number.
● Local devices
"PB" + PB number + "_" (underscore) + device code + device number</td><td rowspan="2"></td></tr>
<tr><td>Devices that can be specified<table><tr><th>Global devices</th><th>Local devices</th></tr><tr><td>WX</td><td>WX</td></tr><tr><td>WY</td><td>WY</td></tr><tr><td>WR</td><td>WR</td></tr><tr><td>WL</td><td>WL</td></tr><tr><td>DT</td><td>DT</td></tr><tr><td>LD</td><td>LD</td></tr></table></td></tr>
<tr><td>Number of transferred data
(number of bytes)</td><td>Specify the number of transferred data (number of bytes). (1 to 7 digits)
* The number of bytes that can be simultaneously transferred is 1 MB for all 16 IDs.</td><td>1 to 1048576
(1MB)</td></tr>
</table>

(Note 1) Input each setting parameter for the destination device setting separated by a comma ",".

(Note 2) Specify the operation setting parameters in the order of the above table.

Setting example

<table>
<tr><td>Example 1</td><td>S3</td><td>"WX16,32"</td></tr>
<tr><td colspan="2">Settings</td><td>Device setting, Device division: Global, Device code: WX, Device number: 16, Number of bytes: 32 bytes</td></tr>
<tr><td>Example 2</td><td>S3</td><td>"DT123456,250"</td></tr>
<tr><td colspan="2">Settings</td><td>Device setting, Device division: Global, Device code: DT, Device number: 123456, Number of bytes: 250 bytes</td></tr>
<tr><td>Example 3</td><td>S3</td><td>"WR0,64"</td></tr>
<tr><td colspan="2">Settings</td><td>Device setting, Device division: Global, Device code: WR, Device number: 0, Number of bytes: 64 bytes</td></tr>
<tr><td>Example 4</td><td>S3</td><td>"WL10,128"</td></tr>
<tr><td colspan="2">Settings</td><td>Device setting, Device division: Global, Device code: WL, Device number: 10, Number of bytes: 128 bytes</td></tr>
</table>

■ Operand [S3] setting (UPDOWN operation for a device)

- Specify the starting address of the device area that stores the string data that indicates source device settings, or a character constant.
- Downloaded data is stored immediately after uploaded data. The number of acquisitions (the number of bytes) is stored in the first two words.

<table>
<tr><th>Setting item</th><th colspan="2">Settings</th><th>Setting range</th></tr>
<tr><td rowspan="4">S3</td><td rowspan="2">Source device setting</td><td>Specify the source device setting.
● Global devices
Specify device code + device number.
● Local devices
"PB" + PB number + "_" (underscore) + device code + device number</td><td rowspan="2"></td></tr>
<tr><td><Devices that can be specified>
<table>
<tr><th>Global devices</th><th>Local devices</th></tr>
<tr><td>WX</td><td>WX</td></tr>
<tr><td>WY</td><td>WY</td></tr>
<tr><td>WR</td><td>WR</td></tr>
<tr><td>WL</td><td>WL</td></tr>
<tr><td>DT</td><td>DT</td></tr>
<tr><td>LD</td><td>LD</td></tr>
</table></td></tr>
<tr><td>Number of transferred data
(number of bytes)</td><td>Specify the number of transferred data (number of bytes). (1 to 7 digits)
* The number of bytes that can be simultaneously transferred is 1 MB for all 16 IDs.</td><td>1 to 1048576
(1MB)</td></tr>
<tr><td>Maximum number of acquisitions
(number of bytes)</td><td>Specify the maximum number of acquisitions (number of bytes). (1 to 7 digits)
* Data can be obtained up to the maximum number of acquisitions.
* The number of bytes that can be simultaneously acquired is 1 MB for all 16 IDs.</td><td>1 to 1048576
(1MB)</td></tr>
</table>

(Note 1) Input each source device setting parameter separated by a comma ",".

(Note 2) Specify the operation setting parameters in the order of the above table.

Setting example

<table>
<tr><td>Example 1</td><td>S3</td><td>"WX16,32.32"</td></tr>
<tr><td colspan="2">Settings</td><td>Device setting, Device division: Global, Device code: WX, Device number: 16, Number of bytes: 32 bytes,
Number of acquisitions: 32 bytes</td></tr>
<tr><td>Example 2</td><td>S3</td><td>"DT123456,250,250"</td></tr>
<tr><td colspan="2">Settings</td><td>Device setting, Device division: Global, Device code: DT, Device number: 123456, Number of bytes: 250 bytes,
Number of acquisitions: 250 bytes</td></tr>
</table>

Example 3	S3	"WR0,64.64"
Settings		Device setting, Device division: Global, Device code: WR, Device number: 0, Number of bytes: 64 bytes, Number of acquisitions: 64 bytes
Example 4	S3	"WL10,128,128"
Settings		Device setting, Device division: Global, Device code: WL, Device number: 10, Number of bytes: 128 bytes, Number of acquisitions: 128 bytes

■ Operand [S4] setting (UPLOAD operation for a device)

- Specify the starting address of the device area that stores a destination URL, or a character constant.
- Specify a folder name and file name with its relative path from the home directory of the user who logs in to the HTTP server.

■ Operand [S4] setting (DOWNLOAD operation for a device)

- Specify the starting address of the device area that stores the string data that indicates a source URL, or a character constant.
- Specify a folder name and file name with its relative path from the home directory of the user who logs in to the HTTP server.

■ Operand [S4] setting (UPDOWN operation for a device)

- Specify the starting address of the device area that stores the string data that indicates a destination URL, or a character constant.
- Specify a folder name and file name with its relative path from the home directory of the user who logs in to the HTTP server.

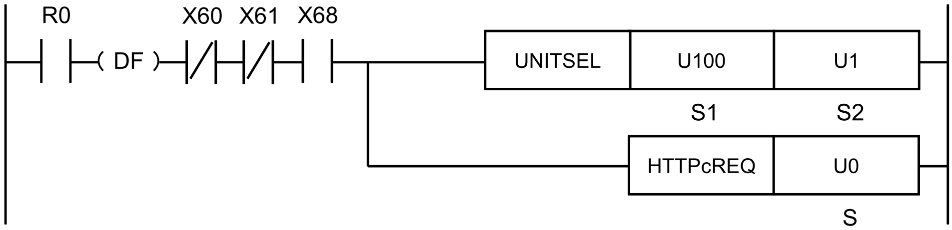
■ Flag operations

Name	Description
SR7 SR8 (ER)	Set when a value outside the range is specified for the parameter. To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). To be set when transfer setting numbers are not specified in ascending order. To be set when executed in an interrupt program. Set when the number of characters for operand specifying character constant exceeds 256. To be set when an HTTP server that has not been specified with the destination server setting instruction or the tool software is specified.
CY (SR9)	Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.18 HTTPcREQ (HTTP Client Transfer Request)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Device address where the transfer number (0 to 15) is stored, or a constant.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●								●	●				●

■ Outline of operation

- This instruction requests the transfer of the HTTP client.
- Before executing this instruction, use the "17.17 HTTPcSET (HTTP Client Transfer Setting)" instruction or the programming tool software "FPWIN GR7" to configure HTTP transfer settings.

■ Processing

- The transfer request relay of the transfer number that is specified by [S] is turned ON.
- This instruction can be executed when the HTTP client preparation done flag (X68) is ON. As an execution condition of the instruction, insert a program that checks the status of the flag (X68). An operation error occurs if this instruction is executed when the flag (X68) is OFF.
- This instruction can be executed when the cable disconnection detection flag (X60) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X60). If this instruction is executed when the flag (X60) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag

(X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.

- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Precautions for programming

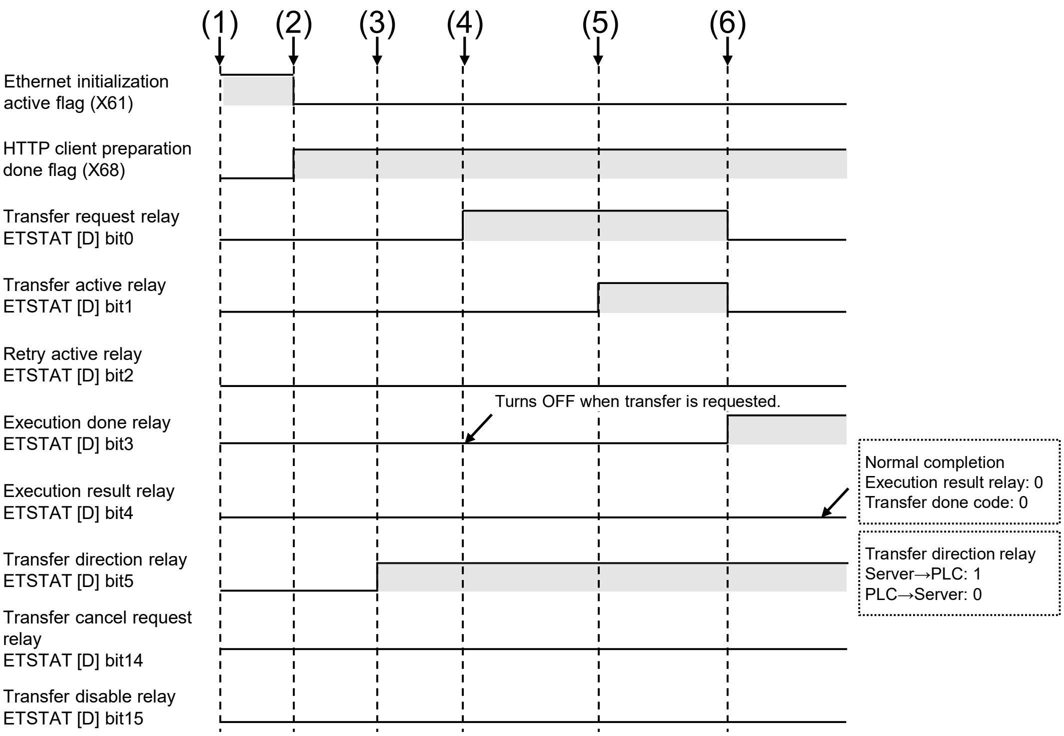
- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- This instruction is not available in interrupt programs.

■ Operand [S] setting

Setting item	Settings		Setting range
S	Transfer number	Specify the device address storing a transfer number or a constant.	0 to 15

■ Timing chart

- The following diagram shows the process in which a transfer request is executed and data transfer from a server to FP7 is completed successfully.
- The control relays (bit0 to bit15) can be monitored by using the ETSTAT instruction to read and store their state in arbitrary operation devices.



(1)	RUN (Power on)	(4)	Transfer request (Executes HTTPcREQ instruction)
(2)	HTTP client preparation done	(5)	HTTP client login succeeded (Starts transfer)

(3)	Transfer setting (Executes HTTPcSET instruction)	(6)	Transfer process done (Completes the execution of HTTPcREQ instruction)

■ Control relay

Name	Bit No.	Description
Transfer request relay	0	0: No request, 1: Request
Transfer active relay	1	0: Stop, 1: During transfer
Transfer retry active relay	2	0: No retry, 1: During retry
Execution done relay	3	0: During process, 1: Instruction execution complete
Execution result relay	4	0: Normal 1: Failed
Transfer direction relay (Note 1)	5	0: Send, 1: Receive
Reserved for system	6 to 13	-
Transfer cancel request relay (Note 2)	14	0: Not cancel, 1: Cancel
Transfer disable relay	15	0: Transfer enabled, 1: Transfer disabled

(Note 1) It is 0 (fixed) for logging and sending mails.

(Note 2) It is 0 (fixed) for logging and HTTPc.

(Note 3) The state of control relays can be read with ETSTAT instruction.

■ Done codes

Name	Number of words	Description
Execution done code	1	Execution done code
Transfer done code	1	Response code of HTTP client

(Note 1) The state of completion codes can be read with ETSTAT instruction.

When the instruction is executed under one of the following conditions, a transfer error occurs and the corresponding error code is set in the execution done code.

Status	Code	Status	Code
Destination server is not set.	1	Transfer prohibition setting	5
Transfer setting is not set.	2	Data decompression failed. (When accessing data with PUT)	8
Registering a process request failed.	4	Data decompression failed. (When accessing data with GET)	9

■ HTTP client preparation done (WX6 bit 8)

Name	Bit No.	Description
HTTP client preparation done (X68)	8	0: HTTP client preparation incomplete, 1: HTTP client preparation complete

(Note 1) For details of Ethernet-related flags, refer to "19.10 Ethernet Function: IP Addresses".

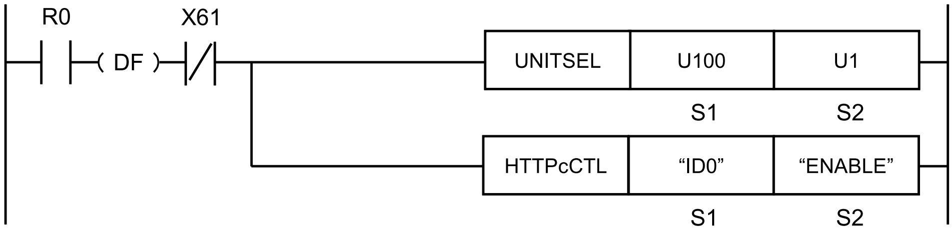
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). To be set in the case of out-of-range in indirect access (index modification). To be set when the HTTP client preparation done (X68) is OFF at the time of the execution of instruction. Set when a value outside the range is specified for the parameter. To be set when the transfer request relay of a specified ID is "Request". Set when executed in an interrupt program. To be set when a transfer setting that has not been specified with the transfer setting instruction or the tool software is specified.
CY (SR9)	To be set when the instruction is executed while the Ethernet cable is disconnected. The detail code set in SD29 is "10: Ethernet cable disconnected". Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.19 HTTPcCTL (HTTP Client Transfer Control)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

HTTPcCTL "ID0" "ENABLE"

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates a control target, or a character constant.
S2	Starting address of the device area that stores the string data that indicates the control content (transfer enabled/disabled/canceled), or a character constant.

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	•	•	•	•			•	•												•	
S2	•	•	•	•			•	•												•	

■ Outline of operation

- This instruction configures the settings for enabling or disabling transfers for an HTTP client.
- Before executing this instruction, use "17.17 HTTPcSET (HTTP Client Transfer Setting)" or the programming tool software "FPWIN GR7" to configure transfer settings.
- It takes some time to accept the processing of the transfer cancel request. After executing the instruction, check the transfer status to see if the transfer stops. For details on checking the transfer status, refer to "17.3 ETSTAT (Acquiring Ethernet Unit Information: FTP/HTTP/SMTP)".

Processing

- The instruction controls whether to enable, disable, or cancel transfer for the target [S1] according to the specification of the control content [S2].
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] and [S2], specify the starting address of the device area that stores the string data that indicates the set parameters, or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous.
- This instruction is not available in interrupt programs.

■ Setting of the operands [S1] and [S2]

Setting item	Settings		
S1	Control target	1) When specifying an individual transfer number	Specify "IDx" with x being a value from 0 to 15.
		2) When specifying all transfer numbers	Specify "ALL".
S2	Control content	1) When enabling transfer	Specify "ENABLE".
		2) When disabling transfer	Specify "DISABLE".
		3) When canceling transfer	Specify "CANCEL".

Setting example

<table>
<tr><th></th><th>Settings</th><th>S1</th><th>S2</th></tr>
<tr><td>Example 1</td><td>When enabling the sending of send number 5</td><td>"ID5"</td><td>"ENABLE"</td></tr>
<tr><td>Example 2</td><td>When disabling all sending</td><td>"ALL"</td><td>"DISABLE"</td></tr>
<tr><td>Example 3</td><td>When canceling the transfer of ID7</td><td>"ID7"</td><td>"CANCEL"</td></tr>
<tr><td>Example 4</td><td>When enabling the sending of send number 10 (Note 1)</td><td>DT0
<table>
<tr><th></th><th colspan="2">Value</th></tr>
<tr><td>DT0</td><td colspan="2">4 (No. of characters)</td></tr>
<tr><td>DT1</td><td>H44(D)</td><td>H49(I)</td></tr>
<tr><td>DT2</td><td>H30(0)</td><td>H31(1)</td></tr>
<tr><td>DT3</td><td></td><td></td></tr>
</table></td><td>DT10
<table>
<tr><th></th><th colspan="2">Value</th></tr>
<tr><td>DT10</td><td colspan="2">6 (No. of characters)</td></tr>
<tr><td>DT11</td><td>H4E(N)</td><td>H45(E)</td></tr>
<tr><td>DT12</td><td>H42(B)</td><td>H41(A)</td></tr>
<tr><td>DT13</td><td>H45(E)</td><td>H4C(L)</td></tr>
<tr><td>DT14</td><td></td><td></td></tr>
</table></td></tr>
</table>

(Note 1) For specifying a device for an operand which can specify character constants, store string data with SSET instruction excluding a double quotation mark.

■ Operation of HTTPc control relay

Name	Transfer enabled	Transfer disabled	Transfer canceled
Transfer cancel relay	Not change	Not change	ON
Transfer disable relay	OFF	ON	Not change
Transfer request	Not change	Not change	Not change
Transfer active	Not change	Not change	Not change
Transfer retry active	Not change	Not change	Not change
Transfer done	Not change	Not change	Not change
Transfer failed	Not change	Not change	Not change
Transfer direction	Not change	Not change	Not change

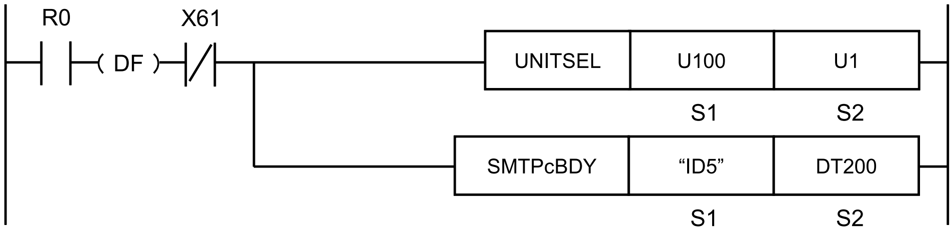
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when an item other than "IDx" or "ALL" is specified for the control target [S1]. (x: 0 to 15) To be set when a transfer setting that has not been specified with the transfer setting instruction or the tool software is specified. To be set when an item other than "ENABLE", "DISABLE" or "CANCEL" is specified for the control content [S2]. To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). Set when executed in an interrupt program. Set when the number of characters for operand specifying character constant exceeds 256.
CY (SR9)	Set when the instruction is the initialization of Ethernet. The detail code that is set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.20 SMTPcBDY (Mail Text Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction. Set a desired value for [S2].

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

SMTPcBDY "ID5" DT200

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates a setting number, or a character constant.
S2	Device address that stores mail text

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●			●	●													●

■ Outline of operation

This instruction sets the specified text as mail text.

■ Processing

- The text that is specified by [S2] is set in the mail text for the setting number that is specified by [S1].
- The instruction can be executed when the mail send request relay for the specified setting number is OFF (0: No request). As an execution condition of the instruction, insert a program that checks the state of the mail send request relay. The state of the send request relay can be read with the ETSTAT instruction. Store the state that is read in a device such as an internal relay. An operation error occurs if this instruction is executed when the send request relay is ON.

- After this instruction is executed, the PLC operates as shown in the following table.

<table>
<tr><th colspan="2">Conditions</th><th>Operation</th></tr>
<tr><td colspan="2">The power supply for the PLC is switched from OFF to ON.</td><td>Setting using the configuration</td></tr>
<tr><td rowspan="2">The PLC is changed to PROG mode.</td><td>Changes to RUN mode without rewriting the configuration.</td><td>Setting using instructions</td></tr>
<tr><td>Changes to RUN mode after rewriting the configuration.</td><td>Setting using the configuration</td></tr>
</table>

- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- Make the event mail setting before executing the instruction.
- This instruction is not available in interrupt programs.

■ Operand [S1] setting

- Specify the starting address of the device area that stores the string data that indicates a setting number (string) and a send trigger, or a character constant.

Setting item	Settings		Setting range
S1	Setting number	Specify a setting number. Event mail send setting number: Idx	0 to 15

■ Operand [S2] setting

- Specify the device address where stores the text is stored.
- When "IDx" is specified for [S1], the maximum size of the text is 4096 bytes. An operation error occurs when it exceeds 4096 bytes.

Setting example

Example)

・Mail text example

Floor A: 25 degrees C. Floor B: 28 degrees C.

S1="ID5" S2=DT200

DT200	H002D		No. of bytes
DT201	H 6C(l)	H 46(F)	Data part
DT202	H 6F(o)	H 6F(o)	Data part
DT203	H 20(˽)	H 72(r)	Data part
DT204	H 3A(:)	H 41(A)	Data part
DT205	H 32(2)	H 20(˽)	Data part
DT206	H 20(˽)	H 35(5)	Data part
DT207	H 65(e)	H 64(d)	Data part
DT208	H 72(r)	H 67(g)	Data part
DT209	H 65(e)	H 65(e)	Data part
DT210	H 20(˽)	H 73(s)	Data part
DT211	H 2E(.)	H 43(C)	Data part
DT212	H 46(F)	H 0D(CR)	Data part
DT213	H 6F(o)	H 6C(l)	Data part
DT214	H 72(r)	H 6F(o)	Data part
DT215	H 42(B)	H 20(˽)	Data part
DT216	H 20(˽)	H 3A(:)	Data part
DT217	H 38(8)	H 32(2)	Data part
DT218	H 64(d)	H 20(˽)	Data part
DT219	H 67(g)	H 65(e)	Data part
DT220	H 65(e)	H 72(r)	Data part
DT221	H 73(s)	H 65(e)	Data part
DT222	H 43(C)	H 20(˽)	Data part
DT223	H 00	H 2E(.)	Data part

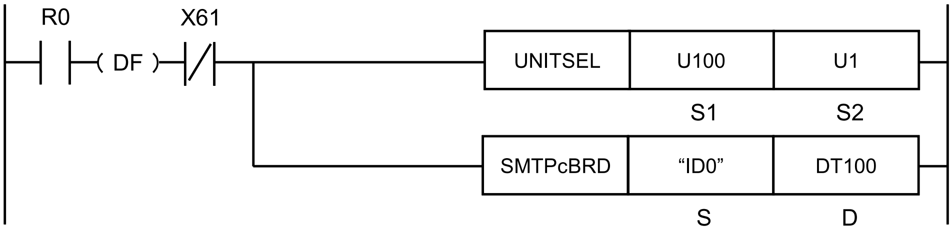
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification). To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). To be set when an out-of-range value is specified for parameters. Set when executed in an interrupt program. To be set when the send request of the mail transmission control relay of a target ID number is 1: Requesting. To be set when the mail transmission setting for a target ID number is not set with the mail transmission setting instruction or the tool software.
CY (SR9)	Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.21 SMTPcBRD (Mail Text Read)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction. Set a desired value for [S2].

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

SMTPcBRD "ID0" DT0

■ List of operands

Operand	Description
S	Starting address of the device area that stores the string data that indicates a setting number, or a character constant.
D	Starting address of the device area that stores mail text

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●												●	
D	●	●	●	●			●	●													●

■ Outline of operation

- This instruction reads the contents of mail texts.

■ Processing

- The instruction is used to read the text creation form that is set for mail text in the mail setting screen of the setting tool. When a mail text is not set, it cannot be read. Zero is stored in the number of bytes of the starting address.
- The mail text for the number that is specified by [S] is read and stored in the device address that is specified by [D].
- The instruction can be executed when the mail send request relay for the specified setting number is OFF (0: No request). As an execution condition of the instruction, insert a program that checks the state of the mail send request relay. The state of the mail send request relay

can be read with the ETSTAT instruction. Store the state that is read in a device such as an internal relay. An operation error occurs if this instruction is executed when the send request relay is ON.

- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- Make the event mail setting before executing the instruction.
- This instruction is not available in interrupt programs.

■ Operand [S] setting

- Specify the starting address of the device area that stores the string data that indicates a setting number (string) and a send trigger, or a character constant.

Setting item	Settings		Setting range
S	Setting number	Specify a setting number. Event mail send setting number: Idx	0 to 15

■ Operand [D] setting

- Specify the starting address of the device area that stores mail text.

Setting example

Example)

・Mail text example

2014/%d/%d Temperature is %d degrees C.

S1="ID0" S2=DT100

DT100	H0027		No. of bytes
DT101	H 30(0)	H 32(2)	
DT102	H 34(4)	H 31(1)	
DT103	H 25(%)	H 2F(/)	
DT104	H 2F(/)	H 64(d)	
DT105	H 64(d)	H 25(%)	
DT106	H 54(T)	H 0D(CR)	
DT107	H 6D(m)	H 65(e)	
DT108	H 65(e)	H 70(p)	
DT109	H 61(a)	H 72(r)	
DT110	H 75(u)	H 74(t)	
DT111	H 65(e)	H 72(r)	
DT112	H 69(i)	H 20(␣)	
DT113	H 20(␣)	H 73(s)	
DT114	H 64(d)	H 25(%)	
DT115	H 64(d)	H 20(␣)	
DT116	H 67(g)	H 65(e)	
DT117	H 65(e)	H 72(r)	
DT118	H 73(s)	H 65(e)	
DT119	H 43(C)	H 20(␣)	
DT120	H 00	H 2E(.)	

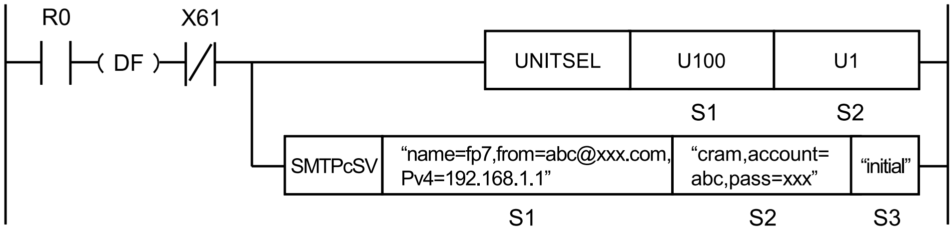
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification). To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). To be set when an out-of-range value is specified for parameters. Set when executed in an interrupt program. To be set when the send request of the mail transmission control relay of a target ID number is 1: Requesting. To be set when the mail transmission setting for a target ID number is not set with the mail transmission setting instruction or the tool software.
CY (SR9)	Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.22 SMTPcSV (Mail Server Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction. Set a desired value for [S2].

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

SMTPcSV "name=fp7,from=abc@xxx.com,IPv4=192.168.1.1" "cram,account=abc,pass=xxx" "initial"

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates sender information and mail sending server information, or a character constant.
S2	Starting address of the device area that stores the string data that indicates the authentication setting parameters, or a character constant.
S3	Starting address of the device area that stores the string data that indicates the detailed setting parameters, or a character constant.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●			●	●												●	
S3	●	●	●	●			●	●												●	

■ Outline of operation

- This instruction sets the information of the connected mail sending server and the sender.

■ Processing

- The mail sending server setting and the sender setting are configured in the CPU unit according to specified parameters.

- The instruction can be executed when the mail send request relays of the mail transmission control relay and the mail send logging/trace control relay are OFF (0: No request). As an execution condition of the instruction, insert a program that checks the state of the mail send request relay. The state of the mail send request relay can be read with the ETSTAT instruction. Store the state that is read in a device such as an internal relay. An operation error occurs if this instruction is executed when the send request relay is ON.
- The initial value is set with the instruction when the server setting is not specified.
- After this instruction is executed, the PLC operates as shown in the following table.

<table>
<tr><th colspan="2">Conditions</th><th>Operation</th></tr>
<tr><td colspan="2">The power supply for the PLC is switched from OFF to ON.</td><td>Setting using the configuration</td></tr>
<tr><td rowspan="2">The PLC is changed to PROG mode.</td><td>Changes to RUN mode without rewriting the configuration.</td><td>Setting using instructions</td></tr>
<tr><td>Changes to RUN mode after rewriting the configuration.</td><td>Setting using the configuration</td></tr>
</table>

- If an incorrect IP address is specified, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- Both upper and lower case characters can be used for operands. "Abcd", "ABCD" and "abcd" are all synonymous. However, the source name, the mail address, the host name, the user name, and the password are case-sensitive.
- This instruction is not available in interrupt programs.

■ Operand [S1] setting

- Specify the starting address of the device area that stores the string data that indicates sender information and mail sending server information, or a character constant.
- A part of parameters can be omitted. The settings are not changed when parameters are omitted partially.
- When omitting the part before a specified keyword, omit only "keyword" without omitting ",".
- When omitting the part after a specified keyword, omit both "," and "keyword".
- Specify a source name, a source e-mail address, the IP address or the host name of a mail server, a port number, and the SSL3/TLS1 authentication setting within 256 one-byte characters in total.
- It is prohibited to specify the same keyword redundantly. An error is caused in the case of redundant specification.

Setting item	Settings	
S1	Source name (can be omitted)	Specify a source name. Specify the keyword "NAME=" at the beginning.
	Source e-mail address (essential)	Specify a source e-mail address. Specify the keyword "FROM=" at the beginning.
	IP address or host name of mail server (essential)	Specify IP address or host name. For an IP address, specify the keyword "IPv4=" or "IPv6=" at the beginning. For a host name, specify "HOST=". ● For IPv4: IPv4 = 111.122.133.144 ● For Ipv6: IPv6 = 1111:122:2:1555:0:0:1888 * For details of the range of IPv4 addresses that can be specified, refer to "19.10 Ethernet Function: IP Addresses""IP address setting specifications". ● For a host name: HOST=smtp.pidsx.com
	Port number (can be omitted)	Specify port number. (Default = 25) Setting range: 1 to 65535
	SSL3/TLS1 Authentication (Can be omitted)	Specify whether or not to use SSL3/TLS1 authentication. SSL= Use SSL3/TLS1 NON=Not use

(Note 1) Input a source name, a source e-mail address, the IP address or the host name of a mail server, a port number, and the SSL3/TLS1 authentication setting separated by a comma ",".

(Note 2) Both upper and lower cases can be used for specifying keywords.

(Note 3) Specify the sender information in the order of the above table.

Setting example

Example 1	S1	"NAME=FP7_001,FROM=pana@pana.com,IPv4=192.255.2.10,PORT=25,SSL"
Settings		Source name: FP7_001, source e-mail address: pana@pana.com IP address: 192.255.2.10, Port number: 25, SSL3/TLS1 authentication: Use
Example 2	S1	",FROM=sunx@sunx.com,IPv6=1111:1222::a8dd:0:0:6666,PORT=100,SSL"
Settings		Source name: Not change, Source e-mail address: sunx@sunx.com IP address: 1111:1222::a8dd:0:0:6666, Port number: 100, SSL3/TLS1 authentication: Use
Example 3	S1	"NAME=FP7_002,FROM=pewsunx@pewsunx.com,HOST=SMTPmailserver.com, PORT=1000,NON"
Settings		Source name: FP7_002, Source e-mail address: pewsunx@pewsunx.com Host name: SMTPmailserver.com, Port number: 1000, SSL3/TLS1 authentication: Not use
Example 4	S1	"NAME=FP7_002,FROM=pewsunx@pewsunx.com,HOST=SMTPmailserver.com"
Settings		Source name: FP7_002, Source e-mail address: pewsunx@pewsunx.com Host name: SMTPmailserver.com, Port number: Not change, SSL3/TLS1 authentication: Not change

■ Operand [S2] setting

- Specify the starting address of the device area that stores the string data that indicates the authentication setting parameters, or a character constant.

- A part of parameters can be omitted. The settings are not changed when parameters are omitted partially.
- When omitting the part before a specified keyword, omit only "keyword" without omitting ",".
- When omitting the part after a specified keyword, omit both "," and "keyword".
- When "NOUSE" or "KEEP" is specified instead of parameters, the instruction operates according to the table of special keywords.
- It is prohibited to specify the same keyword redundantly. An error is caused in the case of redundant specification.

Setting item	Settings		Setting range
S2	SMTP authentication method (Essential)	Specify SMTP authentication method. CRAM: CRAM-MD5 is used. PLAIN1: PLAIN1 (ID/PASS) is used. PLAIN2: PLAIN2 (ID/PASS) is used. LOGIN: LOGIN is used.	
	Account (Can be omitted)	Specify an account. ACCOUNT=XXX (Default: root)	Maximum 32 one-byte characters
	Password (Can be omitted)	Specify a password. Specify the keyword “PASS=” at the beginning. PASS=XXX (Default: root)	Maximum 32 one-byte characters

(Note 1) Input an SMTP authentication method, an account, and a password separated by a comma ",".

(Note 2) Both upper and lower cases can be used for specifying keywords.

(Note 3) Specify the authentication setting parameters in the order of the above table.

Setting example

Example 1	S2	"CRAM,ACCOUNT=sunx,PASS=control"
Settings		SMTP authentication method: CRAM-MD5, Account: sunx, Password: control
Example 2	S2	"PLAIN2,ACCOUNT=FP0R,PASS=small"
Settings		SMTP authentication method: PLAIN2, Account: FP0R, Password: small
Example 3	S2	"LOGIN,ACCOUNT=FP2SH,PASS=middle"
Settings		SMTP authentication method: LOGIN, Account: FP2SH, Password: middle

■ Operand [S2]: account name and password setting

Patterns	How to specify
Account is specified. : Password is deleted.	"CRAM,ACCOUNT=xxx,PASS="
Account is deleted. : Password is specified.	"PLAIN1,ACCOUNT=,PASS=xxx"
Account is deleted. : Password is deleted.	"PLAIN2,ACCOUNT=,PASS="
Account is specified. : Password is not changed.	"LOGIN,ACCOUNT=xxx"
Account is not changed. : Password is specified.	"CRAM,,PASS=xxx"

Setting example

Example 1	S2	"CRAM,ACCOUNT=root,PASS="
Settings		SMTP authentication method: CRAM-MD5, Account: root, Password: Delete
Example 2	S2	"PLAIN1,ACCOUNT=,PASS=SUNX"
Settings		SMTP authentication method: PLAIN1, Account: Delete, Password: SUNX
Example 3	S2	"PLAIN2,ACCOUNT=,PASS="
Settings		SMTP authentication method: PLAIN2, Account: Delete, Password: Delete
Example 4	S2	"LOGIN,ACCOUNT=root"
Settings		SMTP authentication method: LOGIN, Account: root, Password: Not change
Example 5	S2	"CRAM,,PASS=SUNX"
Settings		SMTP authentication method: CRAM, Account: Not change, Password: SUNX

■ Special keyword of operand [S2] setting

Special keyword	Description
NOUSE	The SMTP authentication setting is not used.
KEEP	The current setting is not changed.

Setting example

Example 1	S2	"NOUSE"
Settings		SMTP authentication method: Not use, Account: Not use, Password: Not change
Example 2	S2	“KEEP”
Settings		SMTP authentication method: Not change, Account: Not change, Password: Not change

■ Operand [S3] setting

- Specify the starting address of the device area that stores the string data that indicates the detailed setting parameters, or a character constant.
- A part of parameters can be omitted. The settings are not changed when parameters are omitted partially.
- When omitting the part before a specified keyword, omit only "keyword" without omitting ",".
- When omitting the part after a specified keyword, omit both "," and "keyword".
- When "INITIAL" or "KEEP" is specified instead of parameters, the instruction operates according to the table of special keywords.
- It is prohibited to specify the same keyword redundantly. An error is caused in the case of redundant specification.

Setting item	Settings		Setting range
S3	Maximum sent mail size	Specify the maximum size of a sent mail.	1 to 10240KB

Setting item	Settings		Setting range
	(Can be omitted)	MAIL SIZE=: Sent mail size (Default: 100)	
	Timeout period (Can be omitted)	Specify a timeout period. TOUT=: Time setting (Default: 60 seconds)	30 to 300 seconds
	No. of retries (Can be omitted)	Specify the number of retries. RTRY=: Number of retries (Default: 3 times)	0 to 3
	Retry interval (Can be omitted)	Specify the retry interval. RTTM=: Retry interval (Default: 600 seconds) (Note 1)	10 to 86400 seconds
	Language (Can be omitted)	Specify a language to be used for Subject and Text. JPN= Japanese (Default) ENG= English	

(Note 1) Input the maximum sent mail size, timeout period, number of retries, retry interval and language separated by a comma ",".

(Note 2) Both upper and lower cases can be used for specifying keywords.

(Note 3) The retry interval can be specified in 10-second units. It is rounded down to the nearest 10. (Example: When specifying 38 seconds, 30 seconds are set.)

(Note 4) Specify the authentication setting parameters in the order of the above table.

Setting example

Example 1	S3	"MAILSIZE=1000,TOUT=30,RTRY=2,RTTM=500,JPN"
Settings		Maximum size: 1000, Timeout period: 30 seconds, Number of retries: 2, Retry interval: 500 seconds, Language: Japanese
Example 2	S3	"MAILSIZE=10000,TOUT=270,RTRY=0,RTTM=4900,ENG"
Settings		Maximum size: 10000, Timeout period: 270 seconds, Number of retries: 0 (Not retry), Retry interval: 4900 seconds, Language: English
Example 3	S3	"MAILSIZE=500,TOUT=30,RTRY=3,RTTM=200"
Settings		Maximum size: 500, Timeout period: 30 seconds, Number of retries: 3, Retry interval: 200 seconds, Language: Not change
Example 4	S3	"MAILSIZE=5000,,RTRY=5,RTTM=3000,ENG"
Settings		Maximum size: 5000, Timeout period: Not change, Number of retries: 55, Retry interval: 3000 seconds, Language: English

■ Special keyword of operand [S3] setting

Special keyword	Description
INITIAL	Set an initial value.
KEEP	The current setting is not changed.

Setting example

Example 1	S3	“INITIAL”
Settings		Maximum size: 100, Timeout period: 60 seconds, Number of retries: 3, Retry interval: 600 seconds, Language: Japanese
Example 2	S3	“KEEP”
Settings		Maximum size: Not change, Timeout period: Not change, Number of retries: Not change, Retry interval: Not change, Language: Not change

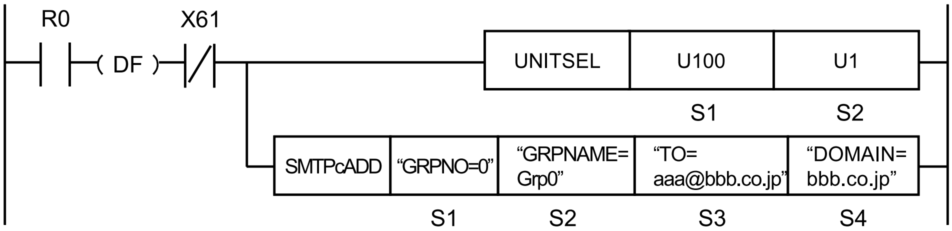
■ Flag operations

Name	Description
SR7 SR8 (ER)	Set when a value outside the range is specified for the parameter. Set when the same keyword is specified redundantly. To be set when even one request active relay of mail transmission control relay or mail transmission logging/trace control relay is 1: Requesting. To be set when "Add-on" is set to "Not use" in Built-in ET-LAN setting. To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). Set when executed in an interrupt program. Set when the number of characters for operand specifying character constant exceeds 256.
CY (SR9)	To be set when the instruction is executed with an incorrect IP address. The detail code set in SD29 is "1: Specification of incorrect IP address". Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.23 SMTPcADD (Destination Group Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction. Set a desired value for [S2].

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

SMTPcADD "GRPNO=0" "GRPNAME=Grp0" "TO=aaa@bbb.co.jp" "DOMAIN=bbb.co.jp"

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates a destination group number, or a character constant.
S2	Starting address of the device area that stores the string data that indicates a destination group name, or a character constant.
S3	Starting address of the device area that stores the string data that indicates a destination address (host name), or a character constant.
S4	Starting address of the device area that stores the string data that indicates a destination address (domain name), or a character constant.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	
S2	●	●	●	●			●	●												●	
S3	●	●	●	●			●	●												●	
S4	●	●	●	●			●	●												●	

■ Outline of operation

- This instruction configures the destination group setting.
- Before executing this instruction, use "17.22 SMTPcSV (Mail Server Setting)" or the programming tool software "FPWIN GR7" to configure the settings of the destination server.

Processing

- This instruction specifies the destination group name specified by [S2] and the destination address specified by [S3] and [S4], for the destination group number specified by [S1].
- The instruction can be executed when the mail send request relays of the mail transmission control relay and the mail send logging/trace control relay are OFF (0: No request). As an execution condition of the instruction, insert a program that checks the state of the mail send request relay. The state of the mail send request relay can be read with the ETSTAT instruction. Store the state that is read in a device such as an internal relay. An operation error occurs if this instruction is executed when the send request relay is ON.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- When specifying a device area for [S1] to [S4], set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- The number of characters should not exceed 256.
- Upper and lower case characters can be used for operands which character constant can be specified. "Abcd", "ABCD" and "abcd" are all synonymous. However, the destination group name, the destination address, the host name, and the domain name are case-sensitive.
- This instruction is not available in interrupt programs.

Operand [S1] setting

- Specify the starting address of the device area that stores the string data that indicates a destination group number (string), or a character constant.

Setting item	Settings		Setting range
S1	Destination group number	Specify a destination group number. Specify the keyword “GRPNO=” at the beginning. GRPNO=Destination group number	0 to 7

(Note 1) Upper and lower case characters can be used for specifying keywords.

Setting example

Example 1	S1	“GRPNO=0”
Settings		Destination group number: 0
Example 2	S1	"GrpNo=7"
Settings		Destination group number: 7

■ Operand [S2] setting

- Specify the starting address of the device area that stores the string data that indicates a destination group name, or a character constant.

Setting item	Settings		Setting range
S2	Destination group name	Specify a destination group name. Specify the keyword "GRPNAME=" at the beginning. GRPNAME=Destination group name	Maximum 64 one-byte characters

(Note 1) Upper and lower case characters can be used for specifying keywords.

Setting example

Example 1	S2	"GRPNAME=Grp0"
Settings		Group name: Grp0
Example 2	S2	"GrpName=Grp1"
Settings		Group name: Grp1

■ Operand [S3] setting

- Specify the starting address of the device area that stores the string data that indicates a destination address, or a character constant.

Setting item	Settings	
S3	Destination address (Host name)	Specify a destination address (host name). Specify the keyword “TO=” at the beginning. TO=Destination address

(Note 1) The destination address of S3 can be specified with a host name only or host name and domain name.

(Note 2) When a domain name is omitted, the destination address is created by the addition of the domain name of S4.

(Note 3) Multiple addresses can be specified by separating each address with ",".

(Note 4) Upper and lower case characters can be used for specifying keywords.

Setting example

Example 1	S3	"TO=suzuki@sunx.co.jp"
Settings		Destination address: suzuki@sunx.co.jp, Domain name: Specified for [S3]
Example 2	S3	"TO=sato"
Settings		Destination address: sato@sunx.co.jp, Domain name: Omitted for [S3], specified as "DOMAIN=sunx.co.jp" for [S4].
Example 3	S3	"TO=suzuki@sunx.co.jp,yamamoto@pana.co.jp"
Settings		Destination address: Multiple addresses (suzuki@sunx.co.jp and yamamoto@pana.co.jp) are specified,

		Domain name: Specified for [S3]
Example 4	S3	"TO=yamamoto,ito"
Settings		Destination address: Multiple addresses (yamamoto@pana.co.jp and ito@pana.co.jp) are specified. Domain name: Omitted for [S3], specified as "DOMAIN=pana.co.jp" for [S4].
Example 5	S3	"TO=suzuki@sunx.co.jp,yamamoto,ito"
Settings		Destination address: Multiple addresses are specified. Domain name: Mix of specified/omitted for [S3], specified "DOMAIN= pana.co.jp" for [S4].

■ Operand [S4] setting

Specify the starting address of the device area that stores the string data that indicates a destination address (domain name), or a character constant.

Setting item	Settings		Setting range
S4	Destination address (Domain name)	Specify a destination address (domain name). Specify the keyword “DOMAIN=” at the beginning. DOMAIN=Domain name	Maximum 32 one-byte characters

(Note 1) When a domain name is omitted for the specification of the destination address of S3, a specified domain name is added.

(Note 2) When all domain names are specified for the specification of the destination addresses of S3, the specification of the domain name of S4 can be omitted.

(Note 3) Only one domain name can be specified.

(Note 4) Upper and lower case characters can be used for specifying keywords.

Setting example

Example 1	S4	"DOMAIN=sunx.co.jp"
Settings		Domain name: sunx.co.jp
Example 2	S4	"Domain=sunx.co.jp"
Settings		Domain name: sunx.co.jp
Example 3	S4	"DOMAIN="
Settings		Domain name: Omitted

■ Flag operations

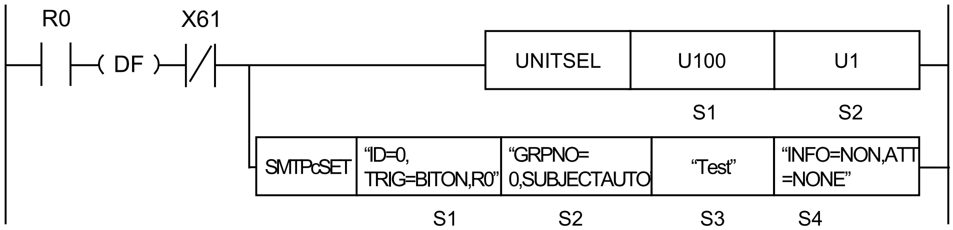
Name	Description
SR7 SR8 (ER)	Set when a value outside the range is specified for the parameter. To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). To be set when even one request active relay of mail transmission control relay or mail transmission logging/trace control relay is 1: Requesting. To be set when the domain name for [S4] is omitted while the destination address [S3] is also specified with the domain name omitted. To be set when executed in an interrupt program.

Name	Description
	Set when the number of characters for operand specifying character constant exceeds 256. To be set when a mail sending server that has not been specified with the destination server setting instruction or the tool software is specified.
CY (SR9)	Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.24 SMTPcSET (Mail Transmission Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction. Set a desired value for [S2].

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

SMTPcSET "ID=0,TRIG=BITON,R0" "GRPNO=0,SUBJECTAUTO" "Test" "INFO=NON,ATT=NONE"

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates a setting number and a send trigger, or a character constant
S2	Starting address of the device area that stores the string data that indicates the destination group number and the subject of the mail to be sent, or a character constant.
S3	Starting address of the device area that stores the string data that indicates the text of the mail to be sent, or a character constant.
S4	Starting address of the device area that stores the string data that indicates the attached data specification of the mail to be sent, or a character constant.

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	•	•	•	•			•	•												•	
S2	•	•	•	•			•	•												•	
S3	•	•	•	•			•	•												•	
S4	•	•	•	•			•	•												•	

■ Outline of operation

- This instruction configures the mail transmission settings.
- Before executing this instruction, use the "17.23 SMTPcADD (Destination Group Setting)" instruction or the programming tool software "FPWIN GR7" to configure event mail settings.

■ Processing

- The mail transmission settings of [S1] to [S4] are stored in the mail transmission setting area.
- The instruction can be executed when the mail send request relay is OFF (0: No request). As an execution condition of the instruction, insert a program that checks the state of the mail send request relay. The state of the mail send request relay can be read with the ETSTAT instruction. Store the state that is read in a device such as an internal relay. An operation error occurs if this instruction is executed when the send request relay is ON.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] to [S4], specify the starting address of the device area that stores the string data that indicates the set parameters, or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous. However, the subject, the mail text, and the attachment file name are case-sensitive.
- This instruction is not available in interrupt programs.

■ Operand [S1] setting

- Specify the starting address of the device area that stores the string data that indicates a setting number (string) and a send trigger, or a character constant.

<table>
<tr><th>Setting item</th><th colspan="2">Settings</th><th>Setting range</th></tr>
<tr><td rowspan="2">S1</td><td>Setting number</td><td>Specify a transfer setting number. Specify the keyword “ID=” at the beginning.
ID=: Transfer setting number</td><td>0 to 15</td></tr>
<tr><td>Send trigger</td><td colspan="2">Specify a send trigger. Specify the keyword “TRIG=” at the beginning.
TRIG=xxxxx
* For information on send triggers, refer to "Operand [S1] send trigger setting".</td></tr>
</table>

(Note 1) Only one setting number and send trigger can be specified simultaneously.

(Note 2) It is prohibited to specify the same keyword redundantly. An error is caused in the case of redundant specification.

(Note 3) Input each setting parameter separated by a comma ",".

(Note 4) Each parameter cannot be omitted. Specify them in the order of the above table. The order of keywords cannot be changed.

(Note 5) Upper and lower case characters can be used for specifying keywords.

(Note 6) Setting numbers should be specified from number 0 in ascending order. An error occurs when transfer setting numbers are not specified in ascending order. If transfer settings have been already registered, this rule is not applied.

■ Operand [S1] send trigger setting

<table>
<tr><th>Set</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">Bit device</td><td colspan="2">Specify the detection of bit device OFF to ON as a trigger. Specify "BITON" for the keyword "TRIG=", and set the bit device.
TRIG=BITON,xxxxx</td></tr>
<tr><td>For global device</td><td>Specify device code + device number.
Example) "X10", "R1024", "DT12345.6"
Addressable devices: X, Y, R, L, T, C, S, P, E, DT.n, LD.n
For system relays, specify "S".</td></tr>
<tr><td>For a local device</td><td>"PB" + PB number + "_" (underscore) + device code + device number
Example) "PB1_X50", "PB80_R512", "PB200_DT102.4"
Addressable devices: X, Y, R, L, T, C, P, E, DT.n, LD.n</td></tr>
<tr><td rowspan="2">Time</td><td colspan="2">Specify sent time. Specify "TIME" for the keyword "TRIG=", and set the time.
TRIG=TIME,xxxx,yyyy</td></tr>
<tr><td colspan="2">● Setting format of cycle
<table>
<tr><th>Cycle</th><th>Set value of xxxx</th><th>Set format of yyyy</th></tr>
<tr><td>Every minute</td><td>/min</td><td>ss</td></tr>
<tr><td>Every hour</td><td>/hour</td><td>mm:ss</td></tr>
<tr><td>Every day</td><td>/day</td><td>hh:mm:ss</td></tr>
<tr><td>Every month</td><td>/mon</td><td>DD:hh:mm:ss</td></tr>
<tr><td>Every year</td><td>/year</td><td>MM:DD:hh:mm:ss</td></tr>
<tr><td>Every week</td><td>/week</td><td>hh:mm:ss-w</td></tr>
</table>
* Specify the format of yyyy as follows; ss = seconds (0 to 59), mm = minutes (0 to 59), hh = hours (0 to 23), DD (1 to 31) = days, MM = months (1 to 12), w = weeks (0 to 6)
* Specify w for every week as follows; 0 = Sunday, 1 = Monday, 2 = Tuesday, 3 = Wednesday, 4 = Thursday, 5 = Friday, 6 = Saturday</td></tr>
<tr><td rowspan="2">Cycle</td><td colspan="2">Specify "CYCLIC" and the following strings in combination for the keyword "TRIG=".
TRIG=CYCLIC,xxxx</td></tr>
<tr><td colspan="2">● Setting value of cycle time
<table>
<tr><th>Cycle unit</th><th>Set value</th></tr>
<tr><td>Seconds</td><td>30SEC</td></tr>
<tr><td>Minutes</td><td>1MIN, 2MIN, 3MIN, 4MIN, 5MIN, 6MIN, 10MIN, 15MIN, 30MIN</td></tr>
<tr><td>Hours</td><td>1HOUR, 2HOUR, 3HOUR, 4HOUR, 6HOUR, 12HOUR, 24HOUR</td></tr>
</table>
* The shortest cycle is 30 seconds.
* Only one cycle time can be set. Setting values such as "1MIN30SEC" cannot be set.</td></tr>
<tr><td>Instruction</td><td colspan="2">Specify SMTPcREQ instruction as a trigger. Specify "PROGRAM" for the keyword "TRIG=".
TRIG=PROGRAM</td></tr>
<tr><td rowspan="2">PLC status change</td><td colspan="2">Specify "STATUS" and the following strings in combination for the keyword "TRIG=".
TRIG=STATUS,xxxx</td></tr>
<tr><td colspan="2">Multiple items can be specified. Separate each item with a comma (,).</td></tr>
</table>

Set	Description	
	Set value	**Meaning**
	PROG>RUN	When switching the swtich PROG to RUN
	RUN>PROG	When switching the swtich RUN to PROG
	ERR>STOP	Operation stop self-diagnostic error detected.
	ERR>RUN	Operation continue self-diagnostic error detected.
	ERRCLR	When error is cleared

Setting example

Example 1	S1	"ID=0,TRIG=BITON,DT100.1"
Settings		Setting number: 0, Send trigger: Bit device (Global device: DT100 Bit 1)
Example 2	S1	"ID=1,TRIG=TIME,/day,13:30:00"
Settings		Setting number: 1, Send trigger: Time (Every day at 13:30)
Example 3	S1	"ID=2,TRIG=TIME,/year,4:1:9:0:0"
Settings		Setting number: 2 Send trigger: Time (Every year at 9:00 on April 1)
Example 4	S1	"ID=3,TRIG=TIME,/week,23:50:00-5"
Settings		Setting number: 3, Send trigger: Time (Every week at 23:50 on Friday)
Example 5	S1	"ID=4,TRIG=CYCLIC,30SEC"
Settings		Setting number: 4, Send trigger: Cycle (30-second cycle)
Example 6	S1	"ID=5,TRIG=CYCLIC,10MIN"
Settings		Setting number: 5, Send trigger: Cycle (10-minute cycle)
Example 7	S1	"ID=6,TRIG=CYCLIC,12HOUR"
Settings		Setting number: 6, Send trigger: Cycle (12-hour cycle)
Example 8	S1	"ID=7,TRIG=PROGRAM"
Settings		Setting number: 7, Send trigger: Instructions
Example 9	S1	"ID=8,TRIG=STATUS,PROG>RUN"
Settings		Setting number: 8, Send trigger: PLC status change (When the switch changes PROG to RUN)
Example 10	S1	"ID=9,TRIG=STATUS,RUN>PROG"
Settings		Setting number: 9, Send trigger: PLC status change (When the switch changes RUN to PROG)
Example 11	S1	"ID=10,TRIG=STATUS,ERR>STOP"
Settings		Setting number: 10, Send trigger: PLC status change (When operation stop self-diagnostic error is detected.)

Example 12	S1	"ID=11,TRIG=STATUS,ERR>RUN"
Settings		Setting number: 11, Send trigger: PLC status change (When operation continue self-diagnostic error is detected.)
Example 13	S1	"ID=12,TRIG=STATUS,ERRCLR"
Settings		Setting number: 12, Send trigger: PLC status change (When error is cleared.)
Example 14	S1	"ID=13,TRIG=STATUS,ERR>STOP,ERR>RUN,ERRCLR"
Settings		Setting number: 13, Send trigger: PLC status change (When operation stop self-diagnostic error is detected.), PLC status change (When operation continue self-diagnostic error is detected.), PLC status change (When error is cleared.)

■ Operand [S2] setting

- Specify the starting address of the device area that stores the string data that indicates the destination group number (string) and the subject of the mail to be sent, or a character constant.

Setting item	Settings		Setting range
S2	Destination group number	Specify a destination group. Specify a destination group number for the keyword "GRPNO=". GRPNO=n1+n2 ... +n8 * Up to eight different group numbers connected with pluses (+) can be selected at the same time.	0 to 7
	Subject	Specify a mail subject. User-specified subject: SUBJECT=xxxxx Automatically-generated subject: SUBJECTAUTO * For details of subjects generated automatically, refer to ""Subjects automatically generated"".	

(Note 1) Input each setting parameter separated by a comma ",".

(Note 2) Each parameter cannot be omitted. Specify them in the order of the above table. The order of keywords cannot be changed.

(Note 3) Upper and lower case characters can be used for specifying keywords.

(Note 4) Up to 100 destinations can be set.

Subjects automatically generated

Subject automatically generated (Japanese)	Subject automatically generated (English)
bit on detect (R100)	bit on detect (R100)
Interval mail (1minute)	Interval mail (1minute)
Interval mail (24hour)	Interval mail (24hour)
Specified Time (Every Minute 0s)	Specified Time (Every Minute 0s)
Specified Time (Every Hour 0m0s)	Specified Time (Every Hour 0m0s)
Specified Time (Every Day 17h30m0s)	Specified Time (Every Day 17h30m0s)
Specified Time (Every Friday 17h30m00s)	Specified Time (Every Friday 17h30m00s)
PLC status change (Power On)	PLC status change (Power On)

Subject automatically generated (Japanese)	Subject automatically generated (English)
PLC status change (Prog > Run)	PLC status change (Prog > Run)
PLC status change (Run > Prog)	PLC status change (Run > Prog)
PLC status change (Operation stop error)	PLC status change (Operation stop error)
PLC status change (Operation continuous error)	PLC status change (Operation continuous error)
PLC status change (Error release)	PLC status change (Error release)
SMTPcREQ command	SMTPcREQ command

(Note 1) When multiple "PLC status change" settings have been specified as send triggers, the subject automatically generated is the PLC status change that is actually detected.

(Note 2) The language used for subjects automatically generated is specified in the mail server setting.

Setting example

Example 1	S2	"GRPNO=0,SUBJECT=Time Notify Mail"
Settings		Destination group number: 0, Subject: User-specified subject "Time Notify Mail"
Example 2	S2	"GRPNO=1+3+4+7,SUBJECT= Cyclic Notify Mail"
Settings		Destination group numbers: 1, 3, 4, 7, Subject: User-specified subject "Cyclic Notify Mail"
Example 3	S2	"GRPNO=0+1+2+3+4+5+6+7,SUBJECTAUTO"
Settings		Destination group numbers: 0 to 7, Subject: Automatically generated

■ Operand [S3] setting

Specify the starting address of the device area that stores the string data that indicates the setting of the text of the mail to be sent, or a character constant.

Setting item	Settings	
S3	Mail text	Specify the starting address of the device area that stores the setting of the text of the mail to be sent, or a character constant.
	Character count (counted as one-byte characters)	Maximum 4096 characters for CPU units Ver.4.1 or later, and Ver.3.4 to Ver.3.x. Maximum 256 characters for CPU units that are other than the above.

■ Operand [S4] setting

- Specify the starting address of the device area that stores the string data that indicates the text auto addition setting and the attached data specification of the mail to be sent, or a character constant.

Setting item	Settings	
S4	Event mail setting	Specify whether to add event transfer information after a mail text specified by user or not. INFO=NON: Not add automatically. INFO=ADD: Add automatically.

Setting item	Settings	
	Specification of automatic addition	* For details of information added automatically, refer to "Automatic additional information".
	Specification of attached data	Specify data to be attached to a mail. Specify the keyword “ATT=” at the beginning.
		● No attached data Specify "NON" for the keyword "ATT". ATT=NONE
		● Specify device (attached to mail text) Specify "DATA" for the keyword "ATT" and specify the device to be added to the mail text. ATT=DATA,xxxxxxxxxxx * For information on how to specify devices, refer to the section "How to specify devices". * For details of information added to mail text, refer to "Device information added to mail text".
		● Specify device (with attached file) Specify "DATA" for the keyword "ATT" and specify the device to be added and attached files. ATT=DATA,xxxxxxxxxxx,FILE=yyyyyyyyyyy * For information on how to specify devices, refer to the section "How to specify devices". * For information on how to specify attached files, refer to "".
		● Specify attached file Specify a file to be attached with full path after specifying "FILE" for the keyword "ATT". ATT=FILE,FileName * LOG folder names ("\LOG0" to "\LOG15") cannot be specified.

Automatic additional information

Character strings added to mails (Japanese)	Character strings added to mails (English)
Basic information	Basic information
Source	From:
CPU Part Number: (Example: CPS4RE, etc.)	CPU Part Number:
IPv4 address	IPv4 address:
IPv6 address	IPv6 address:
Detailed information	Detailed information
bit on detect (R100)	bit on detect (R100)
Interval mail (1minute)	Interval mail (1minute)
Interval mail (24hour)	Interval mail (24hour)
Specified Time (Every Minute 0s)	Specified Time (Every Minute 0s)
Specified Time (Every Hour 0m0s)	Specified Time (Every Hour 0m0s)
Specified Time (Every Day 17h30m0s)	Specified Time (Every Day 17h30m0s)
Specified Time (Every Friday 17h30m00s)	Specified Time (Every Friday 17h30m00s)

Character strings added to mails (Japanese)	Character strings added to mails (English)
PLC status change (Power On)	PLC status change (Power On)
PLC status change (Prog > Run)	PLC status change (Prog > Run)
PLC status change (Run > Prog)	PLC status change (Run > Prog)
PLC status change (Operation stop error)	PLC status change (Operation stop error)
PLC status change (Operation continuous error)	PLC status change (Operation continuous error)
PLC status change (Error release)	PLC status change (Error release)
SMTPcREQ command (PB##, xxxx)	SMTPcREQ command (PB10, 100)

(Note 1) IPv4 address is output only when using IPv4 address, and IPv6 address is output only when using IPv6 address.

(Note 2) The language to be output to mails is specified in the mail server setting.

Device information added to mail text

Character strings added to mails (Japanese)	Character strings added to mails (English)
Device get information	Device get information
Device number: DT100	Device number: DT100
Getting number: 4 devices	Getting number: 4 devices
Conversion method	Exchange method:
1234, 5558, 764, 18270	1234, 5558, 764, 18270

(Note 1) The language to be output to mails is specified in the mail server setting.

■ Operand [S4] Device setting

<table>
<tr><th>Set</th><th>Description</th></tr>
<tr><td rowspan="2">Source device setting</td><td>Specify the source device setting.
● Global device
Specify device code + device number.
Example) such as "WX10", "WR1024", and "DT123456"
● Local device
"PB" + PB number + "_" (underbar) + Device code + Device number
Example) such as "PB1_WX50", "PB80_WR512", and "PB200_DT1024"</td></tr>
<tr><td>Devices that can be specified
<table>
<tr><th>Global devices</th><th>Local devices</th></tr>
<tr><td>WX</td><td>WX</td></tr>
<tr><td>WY</td><td>WY</td></tr>
<tr><td>WR</td><td>WR</td></tr>
<tr><td>WL</td><td>WL</td></tr>
<tr><td>DT</td><td>DT</td></tr>
<tr><td>LD</td><td>LD</td></tr>
<tr><td>SD</td><td></td></tr>
</table></td></tr>
</table>

Set	Description
Number of transferred data (data amount)	Specify the number of transferred data (number of data). (1 to 1000)
Conversion method	Specify a conversion method. **Parameter** BIN1w : Unconverted 16-bit binary US : 16-bit unsigned decimal SS : 16-bit signed decimal UL : 32-bit unsigned decimal SL : 32-bit signed decimal SF : 32-bit single-precision floating point DF : 64-bit double-precision floating point HEX1w : 16-bit HEX HEX2w : 32-bit HEX HEX4w : 64-bit HEX ASCII : ASCII character (Output enclosed with "") * BIN1w cannot be specified for adding to mail texts. For specifying BIN1w, select "Method for adding attached files".
Line feed position	Specify line feed position. ● The setting range is 0 to 255. 0: Output the end of file only n: Output by n data

■ Operand [S4] Attached file setting

Setting item	Description
Attached File Name	Output a device value, and specify the name of the file to be attached to the mail after the keyword "FILE=". FILE=xxxxxxxxx
File name Automatic addition position	Specify the position of the automatic additional data added to a file name. TOP: Automatic additional data is added before a file name. END: Automatic additional data is added after a file name. * Automatic additional data is year, month, day, hour, minute and second "(yymmdd_hhmmss)".

(Note 1) When omitting "File name automatic addition position", automatic additional data is not added to the file name.

(Note 2) Specify the operation setting parameters in the order of the above table.

Setting example

Example 1	S4	"INFO=NON,ATT=NONE"

Settings		Automatic additional information: Not add automatically, Specification of attached data: No attached file
Example 2	S4	"INFO=ADD,ATT=NONE"
Settings		Automatic additional information: Add automatically, Specification of attached data: No attached file
Example 3	S4	"INFO=NON,ATT=DATA,DT100,10,HEX1w"
Settings		Automatic additional information: Not add automatically, Specification of attached data: Specify device (attached to mail text) Device setting, Device division: Global, Device code: DT, Device No.: 100 No. of transferred data: 10 points (10 words), Conversion method: 16-bit HEX
Example 4	S4	"INFO=ADD,ATT=DATA,PB100_WR1000,50,US,FILE=PB100_WR1000_50.csv,TOP"
Settings		Automatic additional information: Add automatically, Specification of attached data: Specify device Device setting, Device division: Local, PB number: 100, Device code: WR, Device number: 1000 Number of transferred data: 50 points (50 words), Conversion method: 16-bit unsigned decimal, Addition of attached file: FIL=PB100_WR1000_50.csv, Automatic addition position: Automatic additional data is added before the file name.
Example 5	S4	"INFO=NON,ATT=FILE,\Folder\FileName.bin"
Settings		Automatic additional information: Not add automatically, Specification of attached data: specify file (\Folder\FileName.bin)

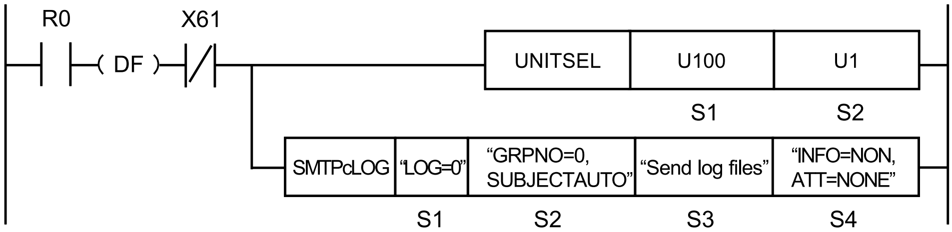
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when an out-of-range value is specified for parameters.
	To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN).
	To be set when setting numbers are not specified in ascending order.
	To be set when the same destination group number is specified redundantly.
	To be set when executed in an interrupt program.
	To be set when the send request of the mail transmission control relay of a specified setting number is 1: Requesting.
	To be set when the number of characters for an operand that allows specifying a character constant exceeds its upper limit. The upper limit is 4096 characters for CPU units Ver.4.1 or later, and Ver.3.4 to Ver.3.x, and 256 characters at maximum for other CPU units.
	To be set when a mail sending server that has not been specified with the destination server setting instruction or the tool software is specified.
	To be set when a destination group number that has not been specified with the destination group setting instruction or the tool software is specified.
CY (SR9)	Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.25 SMTPcLOG (Logging/Trace Mail Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction. Set a desired value for [S2].

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

SMTPcLOG "LOG=0" "GRPNO=0,SUBJECTAUTO" "Send log files" "INFO=NONE,ATT=NONE"

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates a target LOG number, or a character constant.
S2	Starting address of the device area that stores the string data that indicates a subject and a destination group number, or a character constant.
S3	Starting address of the device area that stores the string data that indicates mail text, or a character constant.
S4	Starting address of the device area that stores the string data that indicates the settings for text auto generation and file attachments, or a character constant.

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	•	•	•	•			•	•												•	
S2	•	•	•	•			•	•												•	
S3	•	•	•	•			•	•												•	
S4	•	•	•	•			•	•												•	

■ Outline of operation

- This instruction configures the mail transmission settings for when determining a file in logging/trace.

- Before executing this instruction, use the "SMTPcADD (Destination Group Setting)" instruction or the programming tool software "FPWIN GR7" to configure logging/trace mail settings.

Processing

- The logging/trace mail settings of [S2] to [S4] are stored in the logging/trace setting area that is specified by [S1].
- The instruction can be executed when the mail send request relay is OFF (0: No request). As an execution condition of the instruction, insert a program that checks the state of the mail send request relay. The state of the mail send request relay can be read with the ETSTAT instruction. Store the state that is read in a device such as an internal relay. An operation error occurs if this instruction is executed when the send request relay is ON.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] to [S4], specify the starting address of the device area that stores the string data that indicates the set parameters, or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- The number of characters should not exceed 256.
- Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous. However, the subject, the mail text, and the attachment file name are case-sensitive.
- This instruction is not available in interrupt programs.

Operand [S1] setting

- Specify the starting address of the device area that stores the string data that indicates a logging/trace number (string), or a character constant.

Setting item	Settings		Setting range
S1	Target LOG number	Specify a target LOG number (0 to 15). Specify the keyword “LOG=” at the beginning. LOG=x	0 to 15

(Note 1) Upper and lower case characters can be used for specifying keywords.

Operand [S2] setting

- Specify the starting address of the device area that stores the string data that indicates a subject and a destination group number (string), or a character constant.
- More than one destination group number can be specified simultaneously (maximum 8). Numbers are specified with bits.

Setting item	Settings		Setting range
S2	Destination group number	Specify a destination group. (0 to 7) Specify the keyword "GRPNO=" at the beginning. * Up to eight different group numbers connected with pluses (+) can be selected at the same time.	0 to 7
	Subject	Specify a mail subject. ● User-specified subject Specify a mail subject. Specify a subject for the keyword "SUBJECT=". SUBJECT=xxxx ● Automatically-generated subject A mail subject is automatically generated. Specify the keyword "SUBJECTAUTO". SUBJECTAUTO * For details of subjects generated automatically, refer to "Subjects automatically generated".	User-specified time Maximum 64 one-byte characters

(Note 1) Input each setting parameter for a subject and destination group numbers separated by a comma ",".

(Note 2) A subject and destination group numbers cannot be omitted. Specify them in the order of the above table. The order of keywords cannot be changed.

(Note 3) Upper and lower case characters can be used for specifying keywords.

Setting example

Example 1	S2	"GRPNO=0,SUBJECT=LogFileSend"
Settings		Subject: LogFileSend, Destination group number: 0
Example 2	S2	"GrpNo=0+1+2+3+4+5,Subject=TestSend"
Settings		Subject: TestSend, Destination group numbers: 0,1,2,3,4,5
Example 3	S2	"GrpNo=0+1+2+3+4+5,SubjectAUTO"
Settings		Subject: Automatic, Destination group numbers: 0,1,2,3,4,5

Subjects automatically generated

Subject automatically generated (Japanese)	Subject automatically generated (English)
Logging/Trace (LOG0)	Logging/Trace (LOG0)
Logging/Trace (LOG1)	Logging/Trace (LOG1)
⋮	⋮
Logging/Trace (LOG14)	Logging/Trace (LOG14)
Logging/Trace (LOG15)	Logging/Trace (LOG15)

■ Operand [S3] setting

- Specify the starting address of the device area that stores the string data that indicates mail text, or a character constant.
- Enter a mail text within one-byte 256 characters.

Setting item	Settings		Setting range
S3	Body	Specify the starting address of the device area that stores the string data that indicates mail text, or a character constant.	Maximum 256 one-byte characters

■ Operand [S4] setting

Specify the starting address of the device area that stores the string data that indicates the settings for text auto generation and file attachments, or a character constant.

Setting item	Settings	
S4	Add or not add character strings automatically generated by the unit	Specify whether to generate a message automatically or not. Specify the keyword “INFO=” at the beginning. Not generate a message automatically: INFO=NONE Generate a message automatically: INFO=AUTO
	Attach or not attach files	Specify whether to attach files or not. Specify the keyword “ATT=” at the beginning. Not attach files: ATT=NONE Attach files: ATT=FILE

(Note 1) Input each parameter for setting whether or not to generate a message automatically and whether or not to attach files separated by a comma ",".

(Note 2) The parameters for the automatic generation and file attachment cannot be omitted. Specify them in the order of the above table. The order of keywords cannot be changed.

(Note 3) Upper and lower case characters can be used for specifying keywords.

Setting example

Example 1	S4	"INFO=NONE,ATT=NONE"
Settings		Generate a message automatically: No, Attach files: No
Example 2	S4	"Info=AUTO,Att=FILE"
Settings		Generate a message automatically: Yes, Attach files: Yes

Automatic additional information

Character strings added to mails (Japanese)		Character strings added to mails (English)	
Basic information		Basic information	
	Source		From:
	CPU Part Number: (Example: CPS4RE, etc.)		CPU Part Number:
	IPv4 address		IPv4 address:
	IPv6 address		IPv6 address:
	Logging Trace ID:		Logging Trace ID:
	File fixed Time:		File fixed Time:

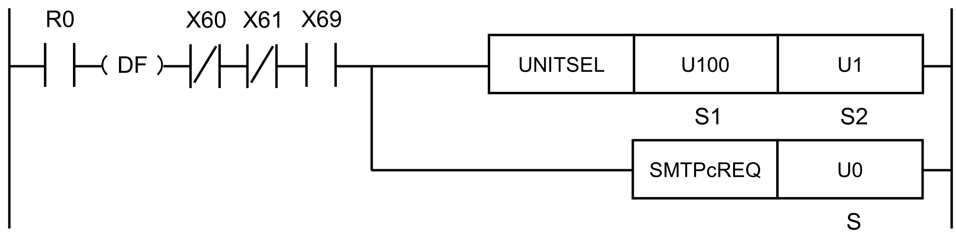
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). To be set when the send request relay of the mail send logging/trace control relay for a target LOG number is "1: Request". To be set when the LOGn send setting for a target LOG number is not registered. To be set when an out-of-range number is specified for a destination group number. To be set when an out-of-range value is specified for parameters. To be set when executed in an interrupt program. Set when the number of characters for operand specifying character constant exceeds 256. To be set when an unset destination group number is specified. To be set when a mail sending server is not specified. To be set when a mail sending server that has not been specified with the destination server setting instruction or the tool software is specified. To be set when a destination group number that has not been specified with the destination group setting instruction or the tool software is specified.
CY (SR9)	Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.26 SMTPcREQ (Mail Send Request)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction. Set a desired value for [S2].

■ List of operands

Operand	Description
S	Specify the device address where the transfer number (0 to 15) is stored, or a constant.

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	•	•	•	•			•	•								•	•				•

■ Outline of operation

- This instruction requests to send a mail.
- Before executing this instruction, use the "SMTPcSET" instruction or the programming tool software "FPWIN GR7" to configure event mail settings.

■ Processing

- The send request relay for the send number that is specified by [S] is turned ON.
- This instruction can be executed when the cable disconnection detection flag (X60) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X60). If this instruction is executed when the flag (X60) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- This instruction can be executed when the SMTP client preparation done flag (X69) is ON. As an execution condition of the instruction, insert a program that checks the status of the

flag (X69). An operation error occurs if this instruction is executed when the flag (X69) is OFF.

- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Precautions for programming

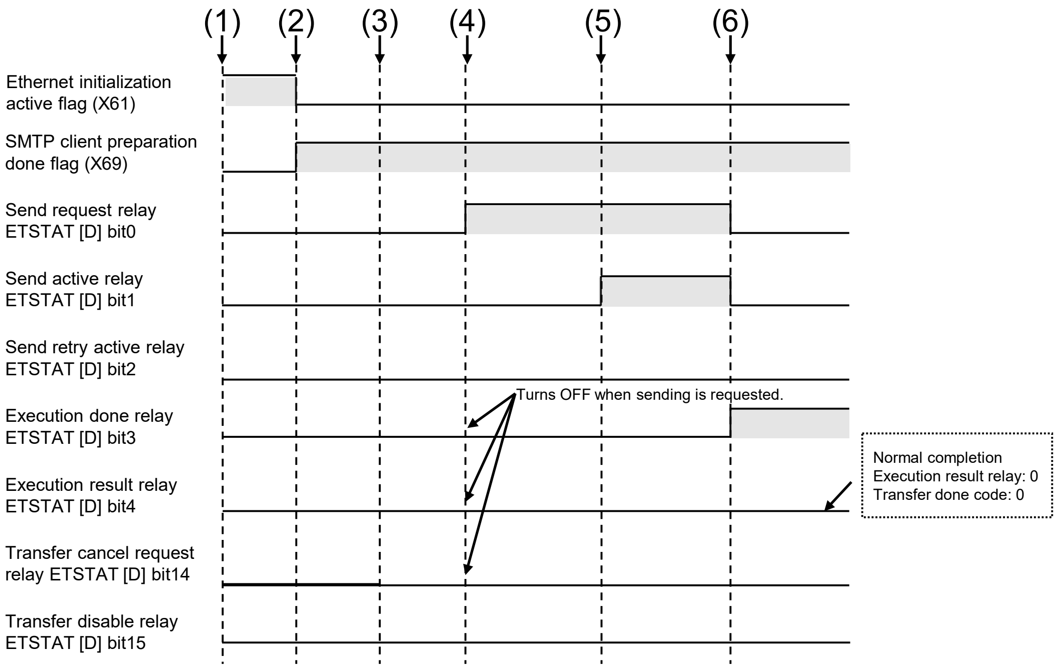
- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- This instruction is not available in interrupt programs.

■ Operand [S] setting

Setting item	Settings		Setting range
S	Send number	Specify the device address storing a send number or a constant.	0 to 15

■ Timing chart

- The following diagram shows the process in which a transfer request is executed and data transfer from a server to FP7 is completed successfully.
- The control relays (bit0 to bit15) can be monitored by using the ETSTAT instruction to read and store their state in arbitrary operation devices.



(1)	RUN (Power on)	(4)	Transfer request (Executes SMTPcREQ instruction)
(2)	SMTP client preparation done	(5)	SMTP client login succeeded (Starts transfer)
(3)	Transfer setting (Executes SMTPcSET instruction)	(6)	Transfer process done (Completes the execution of SMTPcREQ instruction)

■ Control relay

Name	Bit No.	Description
Send request relay	0	0: No request, 1: Request
Send active relay	1	0: Stop, 1: During transfer
Send retry active relay	2	0: No retry, 1: During retry
Execution done relay	3	0: During process, 1: Instruction execution complete
Execution result relay	4	0: Normal 1: Failed
Transfer direction relay	5	0: Send, 1: Receive
Reserved for system	6 to 13	-
Send cancel request relay	14	0: Not cancel, 1: Cancel
Send disable relay	15	0: Transfer enabled, 1: Transfer disabled

(Note 1) The state of control relays can be read with ETSTAT instruction.

■ Done codes

Name	Number of words	Description
Execution done code	1	Execution done code
Send done code	1	Response code of SMTP client

(Note 1) The state of completion codes can be read with ETSTAT instruction.

When the instruction is executed under one of the following conditions, a transfer error occurs and the corresponding error code is set in the execution done code.

Status	Code	Status	Code
Destination server is not set.	1	Transfer prohibition setting	5
Transfer setting is not set.	2	Data decompression failed. (When accessing data with PUT)	8
Destination group is not set.	3	Data decompression failed. (When accessing data with GET)	9
Registering a process request failed.	4		

■ SMTP client preparation done (WX6 bit 9)

Name	Bit No.	Description
SMTP client preparation done (X69)	9	0: SMTP client preparation incomplete, 1: SMTP client preparation complete

(Note 1) For details of Ethernet-related flags, refer to "19.10 Ethernet Function: IP Addresses".

■ Flag operations

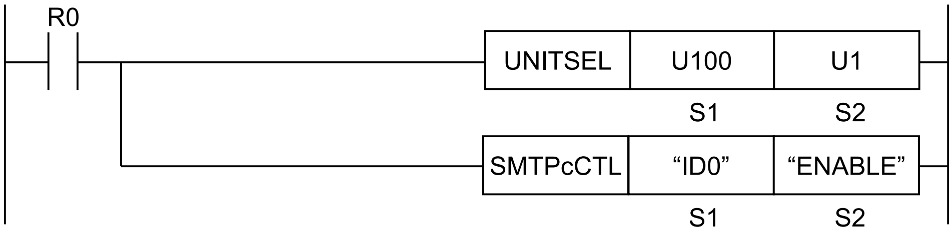
Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification). To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). To be set when the SMTP client preparation done (X69) is OFF at the time of the execution of instruction.

Name	Description
	Set when a value outside the range is specified for the parameter. To be set when the send disable relay is "Send disabled". To be set when the send request relay of a specified ID is "Request". Set when executed in an interrupt program. To be set when a mail transmission setting that has not been specified with the mail transmission setting instruction or the tool software is specified.
SR9 (CY)	To be set when the instruction is executed while the Ethernet cable is disconnected. The detail code set in SD29 is "10: Ethernet cable disconnected". Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.27 SMTPcCTL (Mail Transmission Control)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction. Set a desired value for [S2].

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

SMTPcCTL "ID0" "ENABLE"

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates a control target, or a character constant.
S2	Starting address of the device area that stores the string data that indicates the control content (send enabled/disabled/canceled), or a character constant.

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	•	•	•	•			•	•												•	
S2	•	•	•	•			•	•												•	

■ Outline of operation

- This instruction configures the settings for enabling, disabling, or canceling the sending of mail.
- Before executing this instruction, use the "17.24 SMTPcSET (Mail Transmission Setting)" instruction or the programming tool software "FPWIN GR7" to configure event mail settings. (when control targets are specified with send numbers)
- Before executing this instruction, use the "17.25 SMTPcLOG (Logging/Trace Mail Setting)" instruction or the programming tool software "FPWIN GR7" to configure logging/trace mail settings. (when control targets are specified with LOG numbers)
- It takes some time to accept the processing of the transfer cancel request. After executing the instruction, check the transfer status to see if the transfer stops. For details on checking

the transfer status, refer to the "17.3 ETSTAT (Acquiring Ethernet Unit Information: FTP/HTTP/SMTP)" instruction.

■ Processing

- The instruction controls whether to enable, disable, or cancel mail sending for the target [S1] according to the specification of the control content [S2].
- This instruction can be executed when the Ethernet initialization active flag (X61) is OFF. As an execution condition of the instruction, insert a program that checks the status of the flag (X61). If this instruction is executed when the flag (X61) is ON, the system relay SR9 (carry flag CY) is set and the instruction is terminated without being executed.
- When the instruction is completed successfully, the system relay SR9 (carry flag CY) and the system data register SD29 (Ethernet communication error code) are cleared.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] and [S2], specify the starting address of the device storing the string data which indicates the set parameters or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous.

■ Operand [S1] setting

Setting item	Settings		
S1	Control target	1) When specifying an individual send number	Specify "IDx" with x being a value from 0 to 15.
		2) When specifying an individual LOG number	Specify "LOGx" with x being a value from 0 to 15.
		3) When specifying all send numbers and all LOG numbers	Specify "ALL".

■ Operand [S2] setting

Setting item	Settings		
S2	Control content	1) When enabling sending	Specify "ENABLE".
		2) When disabling sending	Specify "DISABLE".
		3) When canceling sending	Specify "CANCEL".

Setting example

	Settings	S1	S2
Example 1	When enabling the sending of send number 5	"ID5"	"ENABLE"
Example 2	When disabling all sending	"ALL"	"DISABLE"

<table>
<tr><th></th><th>Settings</th><th>S1</th><th>S2</th></tr>
<tr><td>Example 3</td><td>When canceling the transfer of LOG7</td><td>"LOG7"</td><td>"CANCEL"</td></tr>
<tr><td>Example 4</td><td>When enabling the sending of send number 10(Note 1)</td><td>DT0
<table>
<tr><th></th><th colspan="2">Value</th></tr>
<tr><td>DT0</td><td colspan="2">4 (No. of characters)</td></tr>
<tr><td>DT1</td><td>H44(D)</td><td>H49(I)</td></tr>
<tr><td>DT2</td><td>H30(0)</td><td>H31(1)</td></tr>
<tr><td>DT3</td><td></td><td></td></tr>
</table></td><td>DT10
<table>
<tr><th></th><th colspan="2">Value</th></tr>
<tr><td>DT10</td><td colspan="2">6 (No. of characters)</td></tr>
<tr><td>DT11</td><td>H4E(N)</td><td>H45(E)</td></tr>
<tr><td>DT12</td><td>H42(B)</td><td>H41(A)</td></tr>
<tr><td>DT13</td><td>H45(E)</td><td>H4C(L)</td></tr>
<tr><td>DT14</td><td></td><td></td></tr>
</table></td></tr>
</table>

(Note 1) For specifying a device for an operand which can specify character constants, store string data with SSET instruction excluding a double quotation mark.

■ Mail transmission control relay flag operation

Name	Transfer enabled	Transfer disabled	Transfer canceled
Send cancel relay	Not change	Not change	ON
Send disable relay	OFF	ON	Not change
Send request	Not change	Not change	Not change
Send active	Not change	Not change	Not change
Send retry active	Not change	Not change	Not change
Send done	Not change	Not change	Not change
Send failed	Not change	Not change	Not change
Send direction	Not change	Not change	Not change

(Note 1) The send cancel relay turns OFF when the SMTPc transfer request instruction is executed.

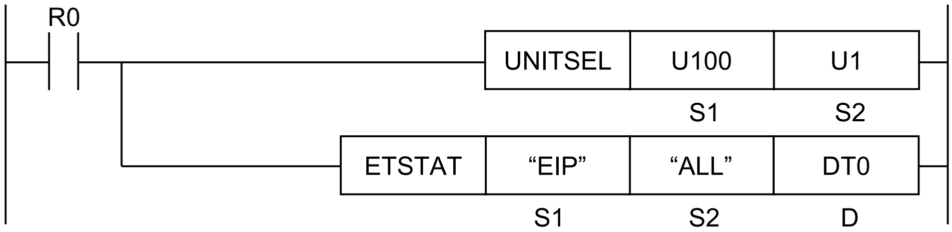
■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when an item other than "IDx" or "LOGx" or "ALL" is specified for the control target (S1). (x: 0 to 15) To be set when an item other than "ENABLE", "DISABLE" or "CANCEL" is specified for the control content (S2). To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN). Set when the number of characters for operand specifying character constant exceeds 256. To be set when a mail transmission setting that has not been specified with the mail transmission setting instruction or the tool software is specified. To be set when a logging/trace mail setting that has not been specified with the logging/trace mail setting instruction or the tool software is specified.
CY (SR9)	Set when the instruction is the initialization of Ethernet. The detail code set in SD29 is "11: Ethernet initialization active".

(Note 1) For details of the error codes stored in the system data SD29, refer to "20.2 List of System Data Registers".

17.28 ETSTAT (Acquiring EtherNet/IP Information)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

(Note 2) By copying and pasting the following text in the instruction list box of FPWIN GR7, the operand part of the above program can be input.

ETSTAT "EIP" "ALL" DT0

■ List of operands

Operand	Description
S1	Starting address of the device area that stores the string data that indicates a read type, or a character constant.
S2	Starting address of the device area that stores the string data that indicates a target to be read, or a character constant.
D	Specify the starting address of the device area that stores the read information.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	●
S2	●	●	●	●			●	●												●	●
D	●	●	●	●			●	●													●

■ Processing

- Reads the parameter information or status information specified by [S1] and [S2], and stores it in the area starting with [D].
- The number of words in the storage area starting with [D] varies according to the type of read data and the target.

■ Precautions for programming

- Insert the UNITSEL instruction immediately before this instruction and specify the unit (built-in ET-LAN in the CPU unit) and the connection number.
- For [S1] and [S2], specify the starting address of the device storing the string data which indicates the set parameters or a character constant. When specifying a device area for an operand, set string data using the SSET instruction in advance. However, the ESSET instruction cannot be used because the format is different.
- Both upper and lower case characters can be used. "Abcd", "ABCD" and "abcd" are all synonymous.
- This instruction is not available in interrupt programs.

■ Setting of [S1] and [S2]

Setting item	Settings		
S1	Read type	For specifying the read of the EtherNet/IP communication state	Specify "EIP".
S2	Read target	For specifying the communication state of EtherNet/IP	Specify "ALL" or "ALL + Number".
		For specifying the cyclic communication registration node table	Specify "NODE".
		For specifying the cyclic communication normal node table	Specify "NORMAL".
		For specifying the cyclic communication stop node table	Specify "STOP".
		For specifying the cyclic communication abnormal node table	Specify "ERR".
		For specifying the RUN/IDLE bit monitor (PLC standby flag)	Specify "PLC".

(Note 1) The RUN/IDLE bit monitor is available for the CPU unit Ver.4.11 or later.

■ Setting of [S2] and targets to be read

- The read contents vary according to the character string set in [S2].
- The number of read words varies according to the maximum registered node number.

Name	Number of words (Note 1)	Character string set in [S2] and read object (●: Read, Blank: Not read)						
		ALL	ALL + Number (1 to 16) (Note 2)	NODE	NORMAL	STOP	ERR	PLC
Registered maximum node number	1	●	●					
Cyclic communication registration node table (Note 3)	0 to 16	●	●	●				

Name	Number of words (Note 1)	Character string set in [S2] and read object (●: Read, Blank: Not read)						
		ALL	ALL + Number (1 to 16) (Note 2)	NODE	NORMAL	STOP	ERR	PLC
Cyclic communication normal node table (Note 3)	0 to 16	●	●		●			
Cyclic communication stop node table (Note 3)	0 to 16	●	●			●		
Cyclic communication abnormal node table (Note 3)	0 to 16	●	●				●	
RUN/IDLE bit monitor (PLC standby flag) (Note 3)	0 to 16	●	●					●
Read word count (Note 1)		1 to 81	1 to 81	1 to 17	1 to 17	1 to 17	1 to 17	1 to 17

(Note 1) The number of read words varies according to the registered maximum node number.

Maximum node number	Number of valid words
0	0
1 to 16	1
17 to 32	2
33 to 48	3
49 to 64	4
:	:
225 to 239	15
241 to 256	16

(Note 2) When specifying "ALL + Number (1 to 16)" for [S2], the information for the number of effective words that is specified by the "Number" is read.

(Note 3) The bits in the following table are allocated to the node table numbers and RUN/IDLE bit monitor.

	Bit No.															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Node number	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
	:															
	256	255	254	253	252	251	250	249	248	247	246	245	244	243	242	241

■ Example of processing

Example 1) When specifying the reading of EtherNet/IP communication state

[S1]... "EIP" [S2]... "ALL" [D]...DT20

	Value	
DT20	U15	Maximum registration node number
DT21	0111 1111 1111 1111	Cyclic communication registration node table (Node nos. 1 to 16)
DT22	0111 1000 1011 1111	Cyclic communication normal node table (Node nos. 1 to 16)
DT23	0000 0111 1010 0000	Cyclic communication stop node table (Node nos. 1 to 16)
DT24	0000 0000 0100 0000	Cyclic communication abnormal node table (Node nos. 1 to 16)
DT25	0000 0000 0000 1111	RUN/IDLE bit monitor (PLC standby flag) (Node nos. 1 to 16)

Example 2) When specifying the reading of EtherNet/IP communication state

When the maximum registered node number is "0", only the value of [D] is updated and the values after [D+1] are not updated.

[S1]... "EIP" [S2]... "ALL" [D]...DT20

	Value	
DT20	0	Maximum registration node number

Example 3) When specifying the reading of cyclic communication registration node table

When setting "ALL+2" for [S2], the information for 32 (=2x16) nodes (node numbers 1 to 32) is read.

[S1]... "EIP" [S2]... "ALL+2" [D]...DT20

	Value	
DT20	15	Maximum registration node number
DT21	1st word	Cyclic communication registration node table (Node nos. 1 to 16)
DT22	2nd word	Cyclic communication registration node table (Node nos.17 to 32)
DT23	1st word	Cyclic communication normal node table (Node nos. 1 to 16)
DT24	2nd word	Cyclic communication normal node table (Node nos. 17 to 32)
DT25	1st word	Cyclic communication stop node table (Node nos. 1 to 16)
DT26	2nd word	Cyclic communication stop node table (Node nos. 17 to 32)
DT27	1st word	Cyclic communication abnormal node table (Node nos. 1 to 16)
DT28	2nd word	Cyclic communication abnormal node table (Node nos. 17 to 32)
DT29	1st word	RUN/IDLE bit monitor (PLC standby flag) (Node nos. 1 to 16)
DT30	2nd word	RUN/IDLE bit monitor (PLC standby flag) (Node nos. 1 to 32)

Example 4) When fixing the number of valid words (The communication states of node numbers 1 to 16 are displayed.)

[S1]... "EIP" [S2]... "ALL+1" [D]...DT20

When setting "ALL+1" for [S2], the information for only one word (node numbers 1 to 16) is read regardless of the maximum registered node number.

	Value	
DT20	100	Maximum registration node number
DT21	1st word	Cyclic communication registration node table (Node nos. 1 to 16)
DT22	1st word	Cyclic communication normal node table (Node nos. 1 to 16)
DT23	1st word	Cyclic communication stop node table (Node nos. 1 to 16)
DT24	1st word	Cyclic communication abnormal node table (Node nos. 1 to 16)
DT25	1st word	RUN/IDLE bit monitor (PLC standby flag) (Node nos. 1 to 16)

Example 5) When specifying the reading of cyclic communication registration node table

[S1]... "EIP" [S2]... "NODE" [D]...WX100

	Value	
WX100	40	Maximum registration node number
WX101	1111 1111 1111 1111	Cyclic communication registration node table (Node nos. 1 to 16)
WX102	1111 1111 1111 1111	Cyclic communication registration node table (Node nos.17 to 32)
WX103	0000 0000 1111 1111	Cyclic communication registration node table (Node nos. 33 to 48)

Example 6) When specifying the reading of cyclic communication normal node table

[S1]... "EIP" [S2]... "NORMAL" [D]...WY100

	Value	
WY100	7	Maximum registration node number
WY101	0000 0000 0111 1111	Cyclic communication normal node table (Node nos. 1 to 16)

Example 7) When specifying the reading of cyclic communication stop node table

[S1]... "EIP" [S2]... "STOP" [D]...WR100

	Value	
WR100	8	Maximum registration node number
WR101	0000 0000 1111 1111	Cyclic communication stop node table (Node nos. 1 to 16)

Example 8) When specifying the reading of cyclic communication abnormal node table

[S1]... "EIP" [S2]... "ERR" [D]...WR100

	Value	
WR100	5	Maximum registration node number
WR101	0000 0000 0000 1000	Cyclic communication abnormal node table (Node nos. 1 to 16)

Example 9) When specifying the reading of RUN/IDLE bit monitor (PLC standby flag)

[S1]... "EIP" [S2]... "PLC" [D]...WR2000

	Value	
WR2000	50	Maximum registration node number
WR2001	1111 1111 1111 1111	RUN/IDLE bit monitor (Node nos. 1 to 16)

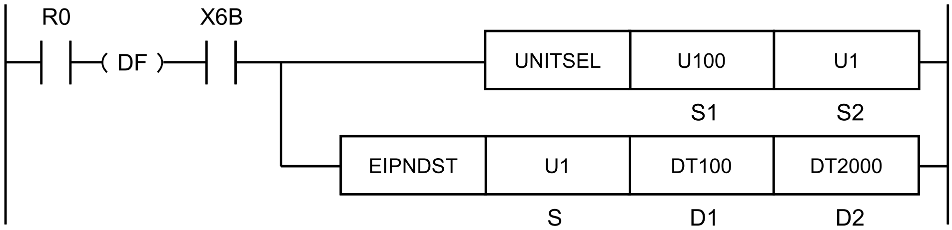
	Value	
WR2002	1111 1111 1111 1111	RUN/IDLE bit monitor (Node nos. 17 to 32)
WR2003	1111 1111 1111 1111	RUN/IDLE bit monitor (Node nos. 33 to 48)
WR2004	0000 0000 0000 0011	RUN/IDLE bit monitor (Node nos. 49 to 64)

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the read area is out of the range.
	To be set when the read type [S1] is set to an item other than "IPv4", "IPv6", "FTPc", "HTTPc", "SMTPc" or "EIP".
	To be set when the target to be read [S2] is set to an item other than "MAC", "CONNECT", "IDx", "LOGx", "IDALL", "LOGALL", "ALL", "NODE", "NORMAL", "STOP", "ERR" or "PLC".
	To be set when a combination other than the combinations listed in the restrictions on combination is specified for the type [S1] and target [S2] to be read.
	To be set when the unit specified by UNITSEL is not the built-in ET-LAN.
	Set when executed in an interrupt program.

17.29 EIPNDST (EtherNet/IP Node Status Acquisition Instruction)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Device area that stores the node number (1 to 256) of the EtherNet/IP device whose status is acquired, or a constant.
D1	Device address for storing the acquired status
D2	Device address for storing the execution result of the instruction

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●								●	●				●
S2	●	●	●	●			●	●													●
D	●	●	●	●			●	●													●

■ Processing

- The status of the node for the node number that is specified by [S] is stored in the device that is specified by [D1], and the execution result of the instruction is stored in [D2].
- The node status is acquired when the trigger (execution condition) turns ON.

■ Precautions for programming

- Execute this instruction after X6B (EtherNet/IP preparation done) turns ON. If the instruction is executed before X6B turns ON, the EtherNet/IP communication preparation incomplete error is returned as an execution result in [D2].
- Multiple EIPNDST instructions cannot be executed simultaneously. A multiple execution error occurs. Be sure to execute this instruction after confirming the completion of the previous execution.

■ Operand [S] setting

Specify node numbers in the range of 1 to 256.

■ Operand [D1] setting

The results of read node statuses are set as follows.

Bits	Name	Definition
0	Owned	Turns ON when FP7 is a target and connected from an originator.
1	Reserved	It is always 0.
2	Configured	Turns ON when the settings of the EtherNet/IP device are different from the factory default settings.
3	Reserved	It is always 0.
4 to 7	Extended Device Status	Shows the detailed status of EtherNet/IP device. It is a vendor-specific status or a status according to CIP.(Note 1)
8	Minor Recoverable Fault	Stores the error information of the EtherNet/IP device. Error contents vary depending on vendors. Recoverable Fault: In a recoverable state Unrecoverable Fault: In an unrecoverable state
9	Minor Unrecoverable Fault	
10	Major Recoverable Fault	
11	Major Unrecoverable Fault	
12 to 15	Reserved	It is always 0.

(Note 1) For bits 4 to 7, the following field definition contents for "Extended Device Status" are stored. FP7 does not return the codes that are indicated as "Not supported" in the following table.

Bits 4 to 7	Name	FP7
0000	During self-testing operation or unknown	Not supported
0001	During the update of firmware	Not supported
0010	More than one I/O connection is in a fault state	Not supported
0011	No I/O connection has been established	
0100	Setting error of non-volatile memory	Not supported
0101	Major Fault. The bit 10 or 11 is ON.	Not supported
0110	More than one I/O connection is established and there is more than one connection that receives RUN mode.	
0111	More than one I/O connection is established and all received connections are in the Idle mode.	

Bits 4 to 7	Name	FP7
1000 to 1001	Reserved	Not suppor ted
1010 to 1111	Peculiar to vendors. Or peculiar to products	Not suppor ted

■ Operand [D2] setting

Specify the area that stores the execution result. One of the following execution codes is stored.

	Name	Value	Description
[D]	Normal end	0	The acquisition of a specified node status is complete.
	In progress	1	The acquisition of a specified node is in progress.
	Timeout	2	Communication timeout (10 seconds)
	Multiple executions	3	Multiple execution of the EIPNDST instruction
	Communication error	4	In the case of communication errors
	CIP error	5	In the case of a CIP error
	EtherNet/IP communication preparation incomplete	6	When the preparation of EtherNet/IP communication is incomplete
[D2+1]	CIP general status	1 to 255	If the value of [D] is "5", CIP general status and CIP extended status are stored. If the value of [D] is not "5", "0" is stored in [D2+1] and [D2+2].
[D2+2]	CIP extended status	0 to 65535	

■ Usage example

Example 1) Acquires the node status of node number 1.

- EtherNet/IP configuration setting

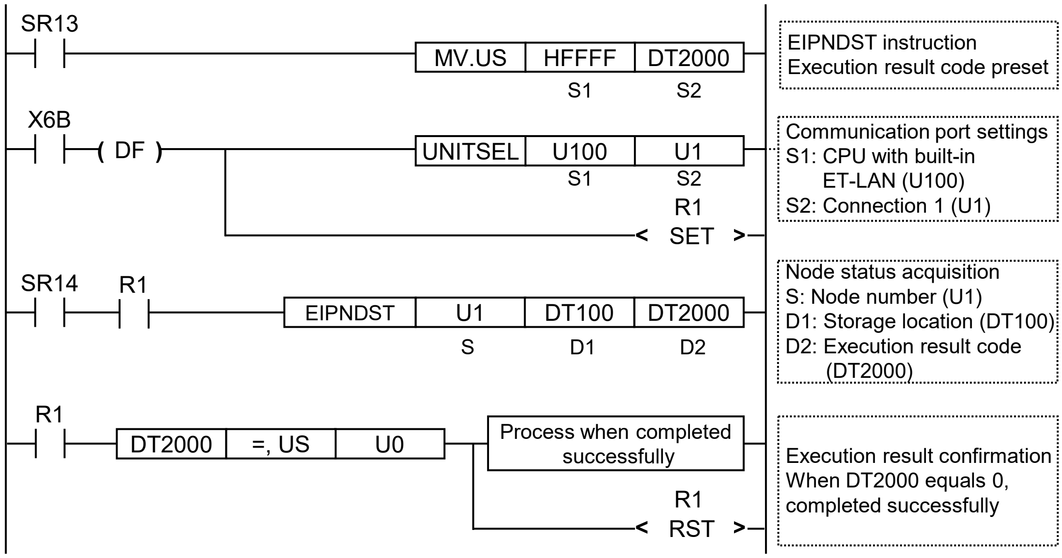
  The EtherNet/IP devices that the node status is acquired should be registered in the scan list.

Node	IP address	Valid/Invalid flag
1	192.168.1.6	Invalid
2	192.168.1.7	Enabled

There is no problem even if the valid/invalid flag is invalid when acquiring the node status. Select valid or invalid to determine whether to perform the cyclic communication or not.

■ Sample program

- The UNITSEL instruction is used to specify the connection number of the built-in ET-LAN in the CPU unit.
- The acquisition result of the node status is stored in DT100 and the execution result is in DT2000. When the operation is complete successfully, 0 is stored in DT2000, and the node status is stored in DT100 and subsequent DTs.



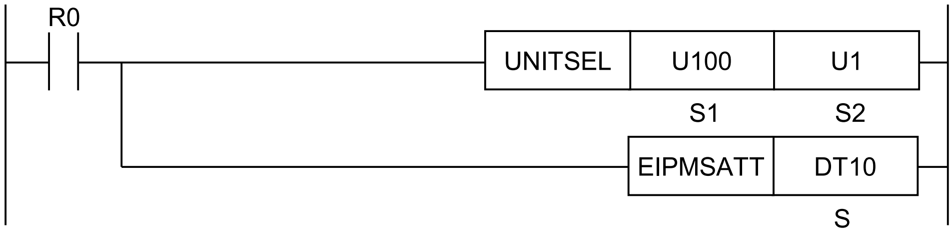
The initial preset is required to acquire the execution result of the EIPNDST instruction.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the unit specified by UNITSEL is not the built-in ET-LAN.
	To be set when the EtherNet/IP function is set to Not Use in the Ethernet unit configuration.
	To be set in case of out-of-range values in indirect access (index modification).
	Set when executed in an interrupt program.
	To be set when the node specified by [S] does not exist.
	To be set when the 3-word device area that starts from the device that is specified by [D2] cannot be allocated.

17.30 EIPMSATT (EIP Message Send Destination Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ Available operation units

No operation unit.

■ List of operands

Operand	Description
S	Specify the starting device number that stores the message communication targets.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●													●

■ Outline of operation

- This instruction specifies the Ethernet unit to be targeted by the UNITSEL instruction.
- It sets the destination data of EIPMSEND instruction in the send buffer.
- The EIPMBODY instruction is used in combination with the EIPMSATT and EIPMSEND instruction.
- When this instruction is called while message communication is being performed, no operation is performed.

■ Processing

- Sets the destination data specified by [S] in the send buffer.

Destination data

S	1st byte of IP address
S+1	2nd byte of IP address

S+2	3rd byte of IP address
S+3	4th byte of IP address
S+4	Service code
S+5	Class ID (Note 1)(Note 2)
S+6	Instance ID (Note 1)(Note 2)
S+7	Attribute ID (Note 1)(Note 2)

(Note 1) The setting range is 0000 to FFFEH. Omitted if set to FFFFH.

(Note 2) For corresponding service codes, class IDs, instance IDs, attribute IDs, refer to relevant manuals for each EtherNet/IP device.

■ Example of processing

Example 1) When executing the Get_Attribute_Single service for an EtherNet/IP device (IP address: 192.168.1.10) to read the product code of the Identity object.

[S]... DT10

	Value
DT9	
DT10	U192
DT11	U168
DT12	U1
DT13	U10
DT14	000EH
DT15	0001H
DT16	0001H
DT17	0003H
DT18	

Setting item	Set value
Destination IP address	192.168.1.10
Service code	000EH
Class ID	0001H
Instance ID	0001H
Attribute ID	0003H

Example 2) When executing the Continuous Data Read service for an EtherNet/IP device (IP address: 192.168.2.1) to continuously read the device data of the PLC object.

[S]... DT100

	Value
DT99	
DT100	U192

DT101	U168
DT102	U2
DT103	U1
DT104	004BH
DT105	0065H
DT106	0001H
DT107	FFFFH (Note 1)
DT108	

(Note 1) FFFFH is specified when this is omitted.

Setting item	Set value
Destination IP address	192.168.2.1
Service code	004BH
Class ID	0065H
Instance ID	0001H
Attribute ID	- (Note 1)

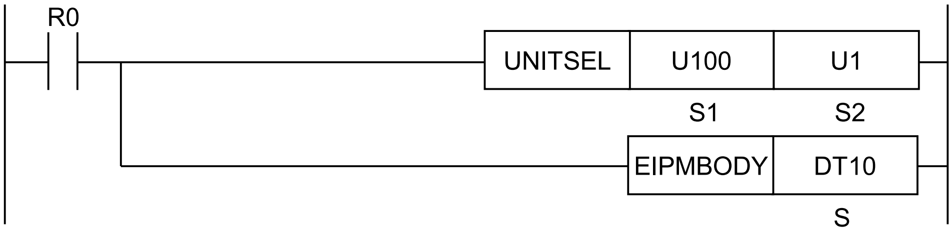
(Note 1) FFFFH is specified when this is omitted.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the unit specified by UNITSEL is not an Ethernet unit.
	To be set when the EtherNet/IP function is set to Not Use in the Ethernet unit configuration.
	To be set in case of out-of-range in indirect access (index modification).
	To be set when the device address of [S+7] is outside the device range.

17.31 EIPMBODY (EIP Message Body Setting)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ Available operation units

No operation unit.

■ List of operands

Operand	Description
S	Specify the starting device number that stores the message body.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●													●

■ Outline of operation

- This instruction specifies the Ethernet unit to be targeted by the UNITSEL instruction.
- It sets the message body data of EIPMSEND instruction in the send buffer.
- The EIPMBODY instruction is used in combination with the EIPMSATT and EIPMSEND instruction.
- When this instruction is called while message communication is being performed, no operation is performed.

■ Processing

- Sets the send buffer in the message body data specified by [S]. The data created by the CIPMSET instruction can be used as the contents of the message body data.

Message body data

	Higher byte	Lower byte
1 word	Message body size (0 to 502 bytes)	
2 words :	Message body data	

(Note 1) For details about the commands and responses, refer to relevant manuals for each EtherNet/IP device.

■ Examples of the maximum service data size

---: Omitted

	Service code	Size	Segment	Class ID	Segment	Instance ID	Segment	Attirubte ID	Service data
1	1byte	0x00	-	-	-	-	-	-	Max. 502 (bytes)
2	1byte	0x01	0x20	1byte	-	-	-	-	Max. 500 (bytes)
3	1byte	0x02	0x0021	2byte	-	-	-	-	Max. 498 (bytes)
4	1byte	0x02	0x20	1byte	0x24	1byte	-	-	Max. 498 (bytes)
5	1byte	0x03	0x20	1byte	0x0025	2byte	-	-	Max. 496 (bytes)
6	1byte	0x03	0x0021	2byte	0x24	1byte	-	-	Max. 496 (bytes)
7	1byte	0x04	0x0021	2byte	0x0025	2byte	-	-	Max. 494 (bytes)
8	1byte	0x03	0x20	1byte	0x24	1byte	0x30	1byte	Max. 496 (bytes)
9	1byte	0x04	0x20	1byte	0x24	1byte	0x0031	2byte	Max. 494 (bytes)
10	1byte	0x04	0x20	1byte	0x0025	2byte	0x30	1byte	Max. 494 (bytes)
11	1byte	0x05	0x20	1byte	0x0025	2byte	0x0031	2byte	Max. 492 (bytes)
12	1byte	0x04	0x0021	2byte	0x24	1byte	0x30	1byte	Max. 494 (bytes)
13	1byte	0x05	0x0021	2byte	0x24	1byte	0x0031	2byte	Max. 492 (bytes)
14	1byte	0x05	0x0021	2byte	0x0025	2byte	0x30	1byte	Max. 492 (bytes)

	Service code	Size	Segment	Class ID	Segment	Instance ID	Segment	Attirubte ID	Service data
15	1byte	0x06	0x0021	2byte	0x0025	2byte	0x0031	2byte	Max. 490 (bytes)

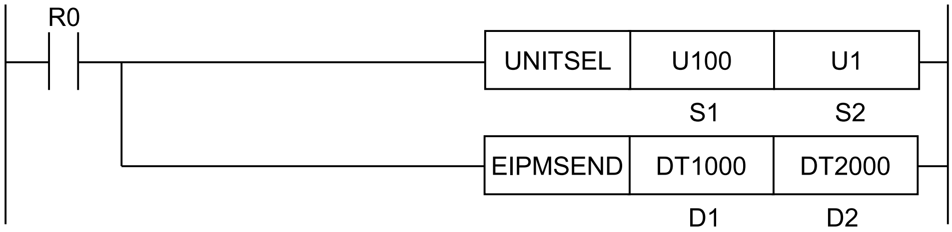
(Note 1) The maximum data size per connection is 504 bytes.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the unit specified by UNITSEL is not an Ethernet unit.
	To be set when the EtherNet/IP function is set to Not Use in the Ethernet unit configuration.
	To be set in case of out-of-range in indirect access (index modification).
	Set when a value outside the range is specified for the parameter.
	To be set when the end of the message body data specified by [S] exceeds the device limit.

17.32 EIPMSEND (EIP Message Send)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction. Set a desired value for [S2].

■ Available operation units

No operation unit.

■ List of operands

Operand	Description
D1	Specify the device address storing received data.
D2	Specify the device address for setting execution results of instructions.

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D1	•	•	•	•			•	•													•
D2	•	•	•	•			•	•													•

■ Outline of operation

- This instruction sends an EIP message when the execution condition turns ON.
- This instruction specifies the Ethernet unit to be targeted by the UNITSEL instruction.
- A UCMM message set by the EIPMSATT and EIPMBODY instructions is sent.
- The response is stored.
- Call this instruction after X6B (EIP preparation done) turns ON. If it is called before X6B turns ON, the EIP communication preparation incomplete error is returned.
- The instruction cannot be used in interrupt programs.
- Multiple EIPMSEND instructions cannot be executed simultaneously. A multiple execution error occurs. The next execution must be executed after confirming the completion of an instruction.

■ Processing

- A UCMM message is sent, received data is stored in [D1] and execution results are stored in [D2]. The destination and the content to be sent are set by the EIPMSATT and EIPMBODY instructions.

  D1: Received data size (byte)

  D1+1: Received data

D1	Received data size (1 to 504 bytes)
D1+1	Received data
D1+2	
D1+x	

(Note 1) When a timeout, multiple execution, or communication error occurs, values are not stored in the received data size and received data.

D2: Execution results

Name	Value	Description
Normal end	0	Message communication is complete.
In progress	1	Message communication is being performed.
Timeout	2	Communication timeout (10 seconds)
Multiple executions	3	Multiple executions of EIPMSEND instruction
Communication error	4	In the case of communication errors
CIP error	5	In the case of CIP errors
EIP communication preparation incomplete	6	When the preparation of EIP communication is incomplete.
Send message size error	7	When the send message size exceeds 504 bytes.

D2+1: CIP general status

D2+2: CIP extended status

	Value	Description
D2+1	1 to 255	CIP general status (Note 1)
D2+2	0 to 65535	CIP extended status (Note 1)

(Note 1) When the execution result is other values than "5", "0" is stored in D2+1 and D2+2.

■ Example of processing

Example) Performing message communication using the connection 1 of the built-in ET-LAN in the CPU unit

- During the configuration setting, it is necessary to set the built-in ET-LAN connection and the EIP scan list.
- The slot number for the built-in ET-LAN needs to be specified to be “100”.

1. First, using the UNITSEL instruction, specify “100” as the slop number for the built-in ET-LAN (S1 = U100), and user connection 1 (S2 = U1).

UNITSEL	U100	U1
	S1	S2

2. Set the destination data using the EIPMSATT instruction.

EIPMSATT	DT100
	S

3. Create a value to be set in the message body data using the CIPMSET instruction.

CIPMSET	DT200	U5	DT500
	S1	Example 1	D

4. Set the message body data using the EIPMBODY instruction.

EIPMBODY	DT500
	S

5. Perform message communication using the EIPMSEND instruction. Received data is stored in D1 and execution results are stored in D2.

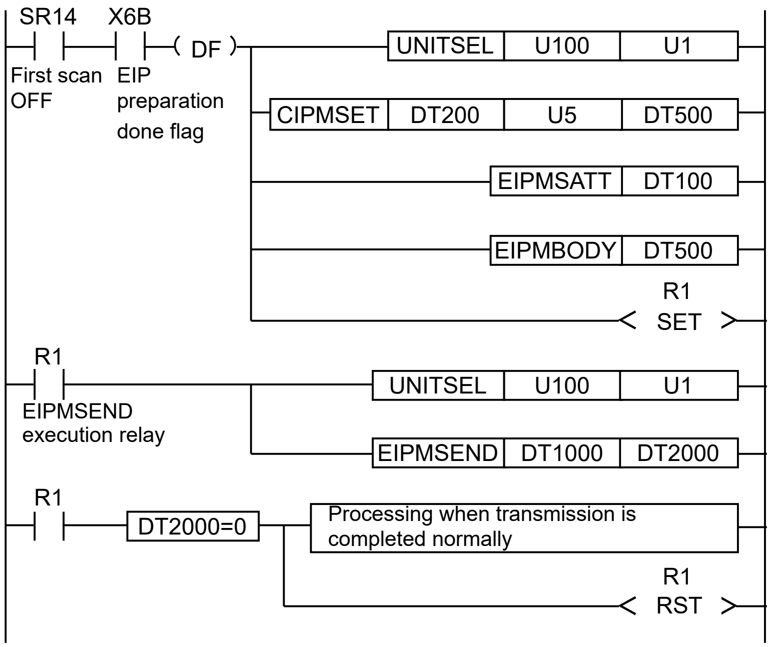
EIPMSEND	DT1000	DT2000
	D1	D2

6. **Results produced when message communication is completed normally**

DT1000	U6 (No. of bytes)		Received data size
DT1001	H00	H8E	Received data
DT1002	H00	H00	
DT1003	H00	HE	

DT2000	H0	Execution result

■ Program example

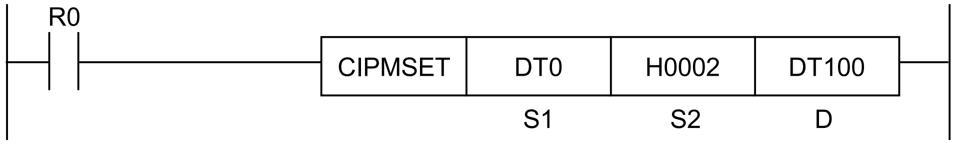


■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the unit specified by UNITSEL is not an Ethernet unit.
	To be set when the EtherNet/IP function is set to Not Use in the Ethernet unit configuration.
	To be set in case of out-of-range in indirect access (index modification).
	Set when executed in an interrupt program.
	To be set when 253-word device cannot be assured from the device address specified by [D1].
	To be set when 3-word device cannot be assured from the device of D2.

17.33 CIPMSET [CIP Message Data Setting (Merging)]

■ Ladder diagram



■ Available operation units

No operation unit.

■ List of operands

Operand	Description
S1	Specify the starting device of send data to be added.
S2	Specify the data format of added send data or the device storing it.
D	Specify the starting device of send data to be created.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32 bit v			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●													●
S2	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

■ Outline of operation

- This instruction is used to create data to be sent in the message communication of CIP.
- If there already exists CIP message data in the storage destination, the data is added to the existing CIP message data.

■ Processing

- The data specified to be added by [S1] is added (merged) to the CIP message specified by [D] according to the format specified by [S2].

S1: Specify the starting device of the data to be added.
When writing character string data, create data using the SSET instruction.
For character string data, specify data that contains character string length.

S2: Specify the format and size of the data to be added.

Specified range: 0 to 502 (000H to 1F6H)

Set value	Description	
0	Character string	Specify when the data to be added is character strings. Add data equivalent to “Starting device value of S1 + 2”.
1 to 502	Other than character string	Specify when the data to be added is other than character strings. Add data equivalent to “set value”.

D: Specify the starting device of the data to be added.
The number of bytes of the currently stored data is set in the starting device.
For character string data, specify data that contains character string length.
If the starting device is not 0, it is recognized that message data already exists and the new data is added next to the position shifted from the starting data by the number of bytes of the existing data.
When writing is completed, the added data size length is added to the CIP data length.

CIP message send data format

	Value	
D	CIP data length	
D+1 onward	CIP data	Complex data consisting of short type, double type, and string data type

Example) [D]: DT100 (shaded) Data write starting position

- When there is no data

	Value	
DT100	0000H	
DT101	41H (A)	42H (B)
DT102	43H (C)	44H (D)

- When there is data

	Value	
DT100	0002H	
DT101	41H (A)	42H (B)
DT102	43H (C)	44H (D)

■ Precautions for programming

- Even if the add source (S1) range overlaps with the add destination (D) range, data is added without causing any error.

■ Example of processing

Example 1) Creating a new CIP message. (Data other than character string data is written in 2 bytes)

[S1]... DT10 [S2]... H0002 [D]...DT100

S1: Data to be added

	Value	
DT0	00H	05H
DT1	42H (B)	41H (A)
DT2	44H (D)	43H (C)
DT3	00H	45H (E)

S2: Format of the data to be added

	Value
S2	0002H

D: CIP message storage destination

	Value		
DT100	0000H		Data length
DT101	34H	12H	

Operation result

S1: Data to be added

	Value	
DT0	00H	05H
DT1	42H (B)	41H (A)
DT2	44H (D)	43H (C)
DT3	00H	45H (E)

Move data equivalent to 2 bytes →

D: CIP message storage destination

	Value	
DT100	0002H	Data length
DT101	0005H	

Example 2) Creating a new CIP message. (Writing character string data “while the data size is set to 0”)

[S1]... DT0 [S2]... H0000 [D]...DT100

S1: Data to be added

	Value	
DT0	00H	05H
DT1	42H (B)	41H (A)
DT2	44H (D)	43H (C)
DT3	00H	45H (E)

S2: Format of the data to be added

	Value
S2	0000H

↑
Writing character string data

D: CIP message storage destination

	Value		
DT100	0000H		Data length
DT101	34H	12H	
DT102	78H	56H	
DT103	12H	90H	
DT104	56H	34H	

Operation result

S1: Data to be added

	Value	
DT0	00H	05H
DT1	42H (B)	41H (A)
DT2	44H (D)	43H (C)
DT3	00H	45H (E)

Move data of string length + 2 bytes →

D: CIP message storage destination

	Value		
DT100	0007H		Data length
DT101	0005H		String length
DT102	'B'	'A'	
DT103	'D'	'C'	
DT104	56H	'E'	

Example 3) Adding data to the existing CIP message. (Data other than character string data is written in 4 bytes)

[S1]... DT1 [S2]... H0004 [D]...DT100

S1: Data to be added

	Value	
DT1	00H	03H
DT2	32H (2)	31H (1)
DT3	00H	33H (3)

S2: Format of the data to be added

	Value
S2	0004H

D: CIP message storage destination

	Value		
DT100	0003H		Data length
DT101	0001H		Written data
DT102	12H	'A'	
DT103	56H	34H	
DT104	90H	78H	

Operation result

S1: Data to be added

	Value	
DT1	00H	03H
DT2	32H (2)	31H (1)
DT3	00H	33H (3)

Move data equivalent to 4 bytes →

D: CIP message storage destination

	Value		
DT100	0003H→0007H		Data length
DT101	0001H		
DT102	03H	'A'	
DT103	31H	00H	
DT104	90H	32H	

Example 4) Adding data to the existing CIP message. (Writing character string data “while the data size is set to 0”)

[S1]... DT1 [S2]... H0000 [D]...DT100

S1: Data to be added

	Value	
DT1	00H	03H
DT2	32H (2)	31H (1)
DT3	00H	33H (3)

S2: Format of the data to be added

	Value
S2	0000H

↑
Writing character string data

D: CIP message storage destination

	Value		
DT100	0003H		Data length
DT101	0001H		Written data
DT102	12H	'A'	
DT103	56H	34H	
DT104	90H	78H	

Operation result

S1: Data to be added

	Value	
DT1	00H	03H
DT2	32H (2)	31H (1)
DT3	00H	33H (3)

Move data of string length + 2 bytes →

D: CIP message storage destination

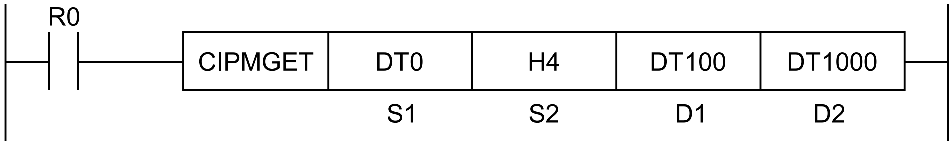
	Value		
DT100	0003H		Data length
DT101	0001H		
DT102	03H	'A'	
DT103	'1'	00H	String length
DT104	'3'	'2'	

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the EtherNet/IP function is set to Not Use in the Ethernet unit configuration.
	To be set in case of out-of-range in indirect access (index modification).
	Set when a value outside the range is specified for the parameter.
	To be set when the data size after the addition exceeds 502.

17.34 CIPMGET (CIP Message Data Getting)

■ Ladder diagram



■ Available operation units

No operation unit.

■ List of operands

Operand	Description
S1	Specify the starting device of received data (CIP data type).
S2	Specify the data format of acquired data or the device storing it.
D1	Specify the device storing the byte offset position from the beginning of the received data which specifies the acquisition position.
D2	Specify the starting device of the device storing acquired data.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●													●
S2	●	●	●	●			●	●								●	●				●
D1	●	●	●	●			●	●													●
D2	●	●	●	●			●	●													●

■ Outline of operation

- This instruction acquires string data and numerical data from the data received in the message communication of CIP.
- Data other than string data is read from lower bytes.

■ Processing

- Data is separated and transferred to the memory specified by [D2] according to the number of data specified by [S2] from the position shifted by the offset of [D1] from the CIP message receive data specified by [S1].

S1: Specify the starting device of CIP message receive data.

Example) S1 = DT0

CIP message send data format

	Value	
D	CIP data length	
D+1	Service code	} CIP receive header
D+2	General Status	
D+3 onward	CIP data	Complex data consisting of short type, double type, and string data type

Stores the following three data items as CIP message data:

- 1
- AB
- 1234H

	Value		
DT0	0011H		Data length
DT1	CIP receive header		
DT2			
DT3	0001H		1st data: '1'
DT4	02H	31H (1)	(Note): The starting one word of the character string data is for the character string length.
DT5	41H (A)	00H	2nd data: 'AB'
DT6	34H	42H (B)	(Note): The starting one word of the character string data is for the character string length.
DT7	ffH	12H	3rd data: 1234H

S2: Specify the data format and data size of the data to be acquired.
Specified range: 0 to 504 (000H to 1F8H)

Set value	Description	
0	Character string	Specify when acquired data is character strings. Acquire data equivalent to “Starting device value of S1 + 2”.
1 to 504	Other than character string	Specify when acquired data is other than character strings. Acquire data equivalent to “set value”.

D1: Specify the device that stores the data acquisition starting position.
Update the data equivalent to the number of data acquired after the instruction is completed.
Starting data length size is not included in the starting position.

Example) When acquiring the second data | Data acquisition starting position

[S1]... DT0 [D1]... DT10

- Before the instruction is issued

	Value
DT10	000BH

- After the instruction is issued

	Value
DT10	000FH

CIP message receive data example

	Value		
DT0	000DH		Total data length
DT1	CIP receive header		
DT2			
DT3	0001H		1st data
DT4	02H	31H (1)	
DT5	41H (A)	00H	2nd data: Acquisition data
DT6	34H	42H (B)	
DT7	ffH	12H	3rd data

Offset position

	Value		
DT0			The data length is not included in the offset position.
DT1	1	0	The CIP header is also extracted.
DT2	3	2	
DT3	5	4	
DT4	7	6	
DT5	9	8	
DT6	B	A	
DT7	D	C	

D2: Specify the storage destination device for the acquired data.

■ Precautions for programming

- With this instruction, delimitation of the CIP message data cannot be checked. Therefore, operation continues without detecting an error even an illegal offset position is specified. Fully grasp the content of a received CIP message, and then set the offset position and data size.
- Even if the acquisition source (S1) range overlaps with the storage location (D2) range, data is acquired without causing any error.

■ Example of processing

Example 1) Acquiring data sequentially from the start of the CIP message.

CIP message receive data example

	Value			Acquire data from the start of CIP message data. Data is acquired to the following device.
DT0	000DH		Total data length	
DT1	00CBH		CIP receive header	→ (1)DT1000
DT2	0000H			
DT3	0001H		1st data	→ (2)DT2000
DT4	02H	31H (1)		
DT5	41H (A)	00H	2nd data	→ (3)DT3000
DT6	34H	42H (B)		
DT7	ffH	12H	3rd data	→ (4)DT4000

(1) Acquiring CIP receive header information from its start

[S1]... DT0 [S2]... H4 [D1]... DT100 [D2]... DT1000

S2: Acquired data format

	Value
S2	0004H

D1: Offset position

	Value
DT100	0000H

D2: Acquired data storage destination

	Value
DT1000	0000H
DT1001	ffffH

Operation result

S1: CIP message receive data

	Value	
DT0	000DH	
DT1	00CBH	
DT2	0000H	
DT3	0001H	
DT4	02H	31H (1)
DT5	41H (A)	42H (B)
DT6	34H	
DT7	ffH	12H

Acquiring data equivalent to 4 bytes →

D1: Offset position

	Value
DT100	0000H→0004H

D2: Acquired data

	Value
DT1000	00CBH
DT1001	0000H

Offset position after updating

	Value		
DT0			
DT1	1	0	CIP receive data
DT2	3	2	
DT3	5	4	1st data
DT4	7	6	
DT5	9	8	2nd data
DT6	B	A	
DT7	D	C	3rd data

(2) Acquiring character string data from the offset position

[S1]... DT0 [S2]... H0 [D1]... DT100 [D2]... DT2000

S2: Acquired data format

	Value
S2	0000H

↑
Acquisition of the character string data

D1: Offset position

	Value
DT100	0004H(Note 1)

D2: Acquired data storage destination

	Value
DT2000	0000H
DT2001	ffffH

(Note 1) The D1 offset position is updated to the start position of the 1st data item when the CIPMGET instruction is issued (1).

Operation result

S1: CIP message receive data

	Value	
DT0	000DH	
DT1	00CBH	
DT2	0000H	
DT3	0001H	
DT4	02H	31H (1)
DT5	41H (A)	00H
DT6	34H	42H (B)
DT7	ffH	12H

Acquiring data equivalent to the character string length + 24 bytes

→

D1: Offset position

	Value
DT100	0004H→0007H

D2: Acquired data

	Value	
DT2000	0001H	
DT2001	ffH	'1'

Offset position after updating

	Value		
DT0			
DT1	1	0	CIP receive data
DT2	3	2	
DT3	5	4	1st data
DT4	7	6	
DT5	9	8	2nd data
DT6	B	A	
DT7	D	C	3rd data

(3) Acquiring character string data from the offset position

[S1]... DT0 [S2]... H0 [D1]... DT100 [D2]... DT3000

S2: Acquired data format

	Value
S2	0000H

↑

Acquisition of the character string data

D1: Offset position

	Value
DT100	0007H(Note 1)

D2: Acquired data storage destination

	Value
DT3000	0000H
DT3001	0000H

(Note 1) The D1 offset position is updated to the start position of the 2nd data item when the CIPMGET instruction is issued (2).

Operation result

S1: CIP message receive data

	Value	
DT0	000DH	
DT1	00CBH	
DT2	0000H	
DT3	0001H	
DT4	02H	31H (1)
DT5	41H (A)	00H
DT6	34H	42H (B)
DT7	ffH	12H

Acquiring data equivalent to the character string length + 24 bytes

→

D1: Offset position

	Value
DT100	0007H→000BH

D2: Acquired data

	Value	
DT3000	0002H	
DT3001	‘B‘	‘A‘

Offset position after updating

	Value		
DT0			
DT1	1	0	CIP receive data
DT2	3	2	
DT3	5	4	1st data
DT4	7	6	
DT5	9	8	2nd data
DT6	B	A	
DT7	D	C	3rd data

(4) Acquiring data other than character string data from the offset position

[S1]... DT0 [S2]... H2 [D1]... DT100 [D2]... DT4000

S2: Acquired data format

	Value
Example 1	0002H

D1: Offset position

	Value
DT100	000BH(Note 1)

D2: Acquired data storage destination

	Value
DT4000	0000H
DT4001	0000H

(Note 1) The D1 offset position is updated to the start position of the 3rd data item when the CIPMGET instruction is issued (3).

Operation result

S1: CIP message receive data

	Value	
DT0	000DH	
DT1	00CBH	
DT2	0000H	
DT3	0001H	
DT4	02H	31H (1)
DT5	41H (A)	00H
DT6	34H	42H (B)
DT7	ffH	12H

Acquiring data equivalent to 2 bytes

D1: Offset position

	Value
DT100	000BH→000DH

D2: Acquired data

	Value
DT4000	1234H
DT4001	0000H

Offset position after updating

	Value		
DT0			
DT1	1	0	CIP receive data
DT2	3	2	
DT3	5	4	1st data
DT4	7	6	
DT5	9	8	2nd data
DT6	B	A	
DT7	D	C	3rd data

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the EtherNet/IP function is set to Not Use in the Ethernet unit configuration.
	To be set in case of out-of-range in indirect access (index modification).
	Set when a value outside the range is specified for the parameter.

Name	Description
	To be set when [D1] (offset position) exceeds the value of the 1st word (total number of data) of [S1] (CIP message) before processing.
	To be set when [D1] (offset position) exceeds the value of the 1st word (total number of data) of [S1] (CIP message) after processing.

■ CIP status codes

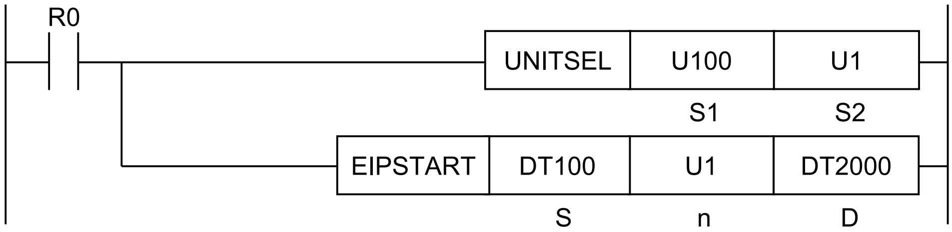
Status code	Status name	Description
0x00	Success	Execution of the service by the specified object was successful.
0x01	Communications Related Problem	A connection-related service was unsuccessful along the connection path.
0x02	Resource unavailable	The resources required for the object to perform the requested service were not available.
0x03	Invalid parameter value	To select the correct value for this condition, refer to Status Code 20 (hexadecimal number).
0x04	Path segment error	The path segment identifier or segment syntax was not interpreted by the processing node. Path processing is stopped if an error occurs in the path segment.
0x05	Path destination unknown	The path references an object class, instance, or structural element that is not identified or contained in the processing node. Path processing is stopped if a path destination unknown error occurs.
0x06	Partial transfer	Only part of the expected data was transferred.
0x07	Connection lost	The messaging connection was interrupted.
0x08	Service not supported	The requested service was not implemented. Or, it was not defined for this object class/instance.
0x09	Invalid attribute value	Invalid attribute data was detected.
0x0A	Attribute list error	An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status.
0x0B	Already in requested mode/state	The object is already in the mode/state being requested by the service.
0x0C	Object state conflict	The object cannot perform the requested service in the current mode/state.
0x0D	Object already exists	The requested instance of the object to be created already exists.
0x0E	Attribute not settable	A request to modify a non-modifiable attribute was received.
0x0F	Privilege violation	A permission/privilege verification was unsuccessful.
0x10	Device state conflict	The device cannot perform the requested service in the current mode/state.
0x11	Reply data too large	The data transmitted in the response buffer is larger than the allocated response buffer.
0x12	Fragmentation of a primitive value	The service specified an operation that is going to fragment a primitive data value, i.e. half a REAL data type.
0x13	Not enough data	The service did not supply enough data to perform the specified operation.
0x14	Attribute not supported	The attribute specified in the request is not supported
0x15	Too much data	The service supplied more data than was expected.

Status code	Status name	Description
0x16	Object instance does not exist	The specified object does not exist in the device.
0x17	Service fragmentation out of sequence	The fragmentation sequence for this service is not active for this data.
0x18	No stored attribute data	The attribute data of this object was not stored before the requested service.
0x19	Store operation failure	The attribute data of this object was not stored due to a detected error during the attempt.
0x1A	Routing failure, request packet too large	The service request packet was too large for transmission on a network in the path to the destination. The routing device was forced to stop the service.
0x1B	Routing failure, response packet too large	The service response packet was too large for transmission on a network in the path from the destination. The routing device was forced to stop the service.
0x1C	Missing attribute list entry data	The service did not provide an attribute from the attribute list required by the service to perform the requested behavior.
0x1D	Invalid attribute value list	The service returns the list of attributes that contains status information about invalid attributes.
0x1E	Embedded service error	An embedded service resulted in an error.
0x1F	Vendor specific error	A vendor-specific error was detected. The additional code field of the error response specifies the detected error. This general error code must only be used if none of the error codes displayed in this table or in an object class definition accurately represents the detected error.
0x20	Invalid parameter	A parameter associated with the request was invalid. This code is used if a parameter does not comply with the requirements of this specification and/or the requirements defined in an application object specification.
0x21	Write-once value or medium already written	An attempt was made to write to a write-once medium (for example, WORM drive, PROM) that has already been written. Or, an attempt was made to modify a value that cannot be modified once established.
0x22	Invalid Reply Received	An invalid response is received (for example, reply service code does not correspond to the request service code, or the response message is shorter than the minimum expected response size). This status code can be used for other purposes of invalid responses.
0x23	Buffer Overflow	The message received is larger than the receiving buffer can handle. The entire message was discarded.
0x24	Message Format Error	The format of the received message is not supported by the server.
0x25	Key Failure in path	The key segment that was included as the first segment in the path does not correspond to the target module. The object-specific status must specify which part of the key check was unsuccessful.
0x26	Path Size Invalid	The size of the path sent with the service request is either not large enough to allow the request to be forwarded to an object, or too much routing data has been included.
0x27	Unexpected attribute in list	The attribute cannot be set at this time.
0x28	Invalid Member ID	The member ID specified in the request does not exist in the specified class/instance/attribute.

Status code	Status name	Description
0x29	Member not settable	A request to modify a non-modifiable member was received.
0x2A	Group 2 only server general failure	This error code is reported by DeviceNet Group 2 only. It is used only as substitute for those with a code space of 4K or less, for the service not supported, for the attribute not supported, and for the attribute not settable.
0x2B	Unknown Modbus Error	A CIP to Modbus translator has received an undefined Modbus exception code.
0x2C	Attribute not gettable	A request to read a non-readable attribute was received.
0x2D	Instance Not Deletable	The requested object instance cannot be deleted.
0x2E	Service Not Supported for Specified Path 1	The object supports the service, but not for the designated application path (for example, attribute). Note: This cannot be used in cases where more specific general status codes are applied. Example: 0x0E (attributes are not settable) or 0x29 (members are not settable).
0x2F to 0xCF		Reserved by CIP for future extensions.
0xD0 to 0xFF	Reserved for Object Class specific errors	This range of error codes is to be used to indicate errors specific to the object class. Use of this range should only be performed when none of the error codes presented in this table accurately reflect the error that was encountered.

17.35 EIPSTART (Cyclic Communication Start Request)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Specify the starting address of the device area that stores the start request node number table.
n	Specify the device address storing the maximum node number (1 to 256) or a constant.
D	Specify the device address storing execution results.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●													●
S2	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

■ Processing

- The instruction requests the starting of the EtherNet/IP cyclic communication according to the start request node number table that is stored in the area that starts from [S].
- For [n], specify the maximum node number among the nodes to which the start of the EtherNet/IP cyclic communication is requested.
- The execution result is stored in [D].

■ Operand [S] setting

- Specify the starting address of the device area that stores the start request node number table.
- Use a user program to create the start request node number table. Turn ON the bits (that is, set the bits to 1) that correspond to the node numbers to which the start request is made.

(Example) When [S] is set to WR100 and the start request is made to nodes number 1 and 2

Set bit 0 (R1000) and bit 1 (R1001) in WR100 to "1" and execute the EIPSTART instruction.

	Bit No.															
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Node number	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
	:															
	256	255	254	253	252	251	250	249	248	247	246	245	244	243	242	241

■ Operand [n] setting

- Specify the device address storing the maximum node number or a constant.
- The number of valid words for the start request node number table varies (from 1 to 16 words) according to the maximum node number that is specified by [n].

Maximum node number	Number of valid words
0	0
1 to 16	1
17 to 32	2
33 to 48	3
49 to 64	4
:	:
225 to 239	15
241 to 256	16

■ Operand [D] setting

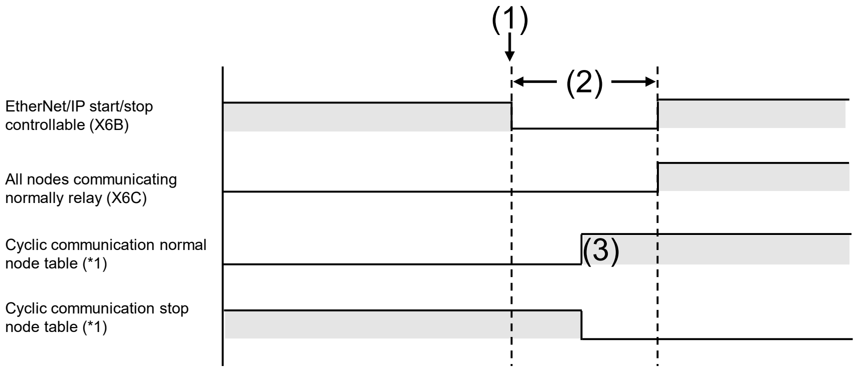
Specify the device address storing execution results.

Code	Status	Description
0	Normal end	The specified node start is complete.
1	In progress	The specified node start processing is in progress.
2	Start failed	The specified node start failed.
3	Multiple executions	Multiple execution of the EIPSTART instruction or the EIPSTOP instruction

■ Relay operation

When the cyclic communication start request instruction is executed and the cyclic communication of the specified node starts normally, the cyclic communication normal node table for the node is turned ON and the cyclic communication stop node table for the node is turned OFF.

Example) Relay operation when the cyclic communication start request is made on a stopped node



(Note 1) The state can be checked by the ETSTAT instruction.

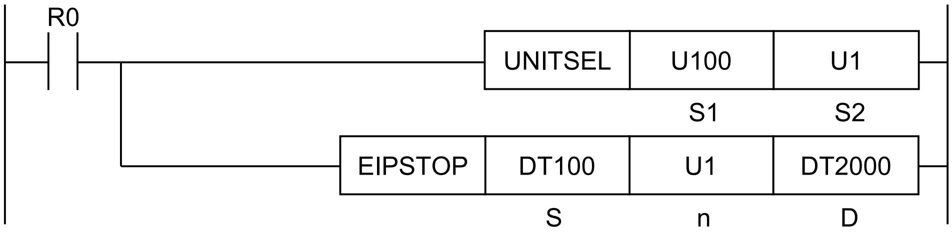
(1)	Cyclic Communication Start Request (EIPSTART)	(2)	Instruction reception impossible period	(3)	The specified node start is complete.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the unit specified by UNITSEL is not the built-in ET-LAN.
	To be set when the EtherNet/IP function is set to Not Use in the Ethernet unit configuration.
	Set when executed in an interrupt program.
	To be set when the value of [n] exceeds 256.
	To be set when the address that is specified by [S] + [Number of valid words for [n]] is out of the device range.
	To be set in the case of out-of-range in indirect access (index modification).

17.36 EIPSTOP (Cyclic Communication Stop Request)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S	Specify the starting address of the device area that stores the stop request node number table.
n	Specify the device address storing the maximum node number (1 to 256) or a constant.
D	Specify the device address storing execution results.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●													●
S2	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

■ Processing

- The instruction requests the stopping of the EtherNet/IP cyclic communication according to the stop request node number table that is stored in the area that starts from [S].
- For [n], specify the maximum node number among the nodes to which the stop of the EtherNet/IP cyclic communication is requested.
- The execution result is stored in [D].

■ Operand [S] setting

- Specify the starting address of the device area that stores the stop request node number table.
- Use a user program to create the stop request node number table. Turn ON the bits (that is, set the bits to 1) that correspond to the node numbers to which the stop request is made.

Example) When [S] is set to WR100 and the stop request is made to nodes number 1 and 2

Set bit 0 (R1000) and bit 1 (R1001) in WR100 to "1" and execute the EIPSTOP instruction.

	Bit No.															
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Node number	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
	:															
	256	255	254	253	252	251	250	249	248	247	246	245	244	243	242	241

■ Operand [n] setting

- Specify the device address storing the maximum node number or a constant.
- The number of valid words for the stop request node number table varies (from 1 to 16 words) according to the maximum node number that is specified by [n].

Maximum node number	Number of valid words
0	0
1 to 16	1
17 to 32	2
33 to 48	3
49 to 64	4
:	:
225 to 239	15
241 to 256	16

■ Operand [D] setting

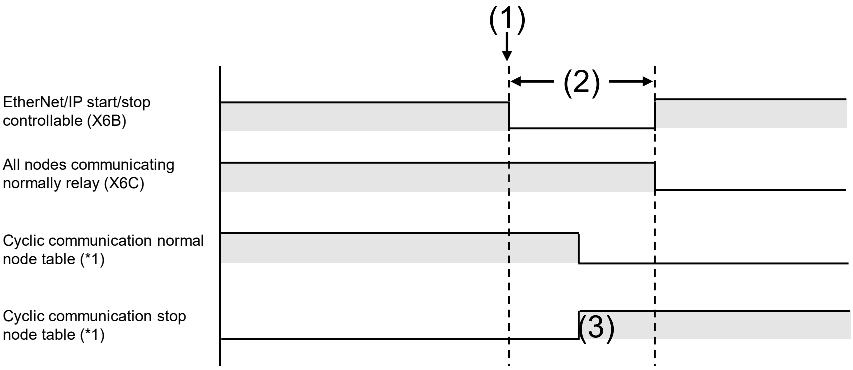
Specify the device address storing execution results.

Code	Status	Description
0	Normal end	The specified node stop is complete
1	In progress	The specified node stop processing is in progress.
2	Start failed	The specified node stop failed.
3	Multiple executions	Multiple execution of the EIPSTART instruction or the EIPSTOP instruction

■ Relay operation

When the cyclic communication stop request instruction is executed and the cyclic communication of the specified node stops normally, the cyclic communication stop node table for the node is turned ON and the cyclic communication normal node table for the node is turned OFF.

Example) Relay operation when the cyclic communication stop request is made on a started node



(Note 1) The state can be checked by the ETSTAT instruction.

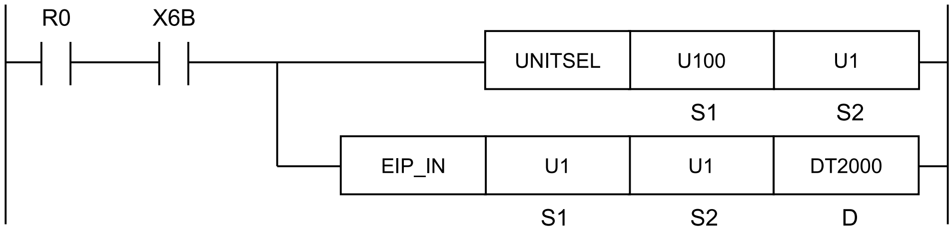
(1)	Cyclic Communication Stop Request (EIPSTOP)	(2)	Instruction reception impossible period	(3)	The specified node stop is complete

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the unit specified by UNITSEL is not the built-in ET-LAN.
	To be set when the EtherNet/IP function is set to Not Use in the Ethernet unit configuration.
	Set when executed in an interrupt program.
	To be set when the value of [n] exceeds 256.
	To be set when the address that is specified by [S] + [Number of valid words for [n]] is out of the device range.
	To be set in the case of out-of-range in indirect access (index modification).

17.37 EIP_IN (EtherNet/IP Input Refresh)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S1	Specify the target node number of the input refresh.
S2	Specify the target connection number of the input refresh.
D	Specify the device address storing refresh results.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●								●	●			●	●
S2	●	●	●	●			●	●								●	●			●	●
D	●	●	●	●			●	●													●

■ Processing

- Only when the connection that is to be refreshed receives new data, this instruction refreshes data for the connection. "Input refresh" means that the data is copied from the receive buffers to the allocated devices.

■ Precautions for programming

- Execute this instruction after the EtherNet/IP preparation done flag (X6B) turns ON. If the instruction is executed before the flag turns ON, the EtherNet/IP communication preparation incomplete error occurs.
- This instruction causes a processing load. Do not execute the instruction successively in one scan.
- Before executing this instruction, use the cyclic communication normal node table to confirm that the communication of the specified connection is performed normally. The cyclic

communication normal node table can be checked by using the "ETSTAT (Acquiring EtherNet/IP Information)" instruction.

- Use this instruction only for the connections in which the refresh method of the "EtherNet/IP setting" is set to "Instruction" by the tool software. An operation error occurs if the batch refresh method or the division refresh method is specified.

■ Operand [S1] setting

Specify a node number to be refreshed. An error occurs when a value over the maximum value specified by the scan list is specified.

An error also occurs when a reserved node is specified.

	Set value
Scan List	1 to 256

■ Operand [S2] setting

Specify a connection number to be refreshed. Specify a relative number within nodes for the connection number.

An error occurs when a value over the maximum value specified by the scan list is specified.

	Set value
Connection number	1 to 256

■ Operand [D] setting

- Specify the device address storing refresh results.
- When there is no new received data, the refresh operation is not performed.

Execution result	Description
0	Refresh operation is complete successfully.
1	No data is received. Refresh is not performed.
2	EtherNet/IP communication preparation incomplete

■ Usage example

Example 1) Refreshing data from the receive buffer of connection number 1 of node number 1 (when the refresh is completed normally)

[S1]... U1 [S2]... U1 [D]... DT20

- EtherNet/IP configuration setting

Setting item	Settings
Node number	1
Connection	1
Input information (T>0) device allocation	LD20 to LD23
	LD51 to LD52

	Receive buffer	Value
*	0	0011h
*	1	2233h
*	2	4455h
*	3	6677h
	4	
	5	
	6	
*	7	EEFFh
*	8	FFEEh
	9	

	Value
LD19	
LD20	0011h
LD21	2233h
LD22	4455h
LD23	6677h
LD24	

LD50	
LD51	EEFFh
LD52	FFEEh
LD53	

	Value
DT20	0

(Note 1) Receive buffers to which devices are allocated

Example 2) Refreshing data from the receive buffer of the connection 2 of the node number 5 (when there is no new data)

[S1]... U5 [S2]... U2 [D]... DT100

● EtherNet/IP configuration setting

Setting item	Settings
Node number	5
Connection	2
Input information (T>0) device allocation	LD111 to LD112
	LD115 to LD116
	LD118 to LD119

	Receive buffer	Value
*	0	CCDDh
*	1	EEFFh
	2	
	3	
*	4	3344h
*	5	5566h
	6	
*	7	99AAh
*	8	BBCCh
	9	

		Value
	LD110	
x	LD111	CCDDh
x	LD112	EEFFh
	LD113	
	LD114	
x	LD115	3344h
x	LD116	5566h
	LD117	
x	LD118	99AAh
x	LD119	BBCCh
	LD120	

	Value
DT100	1

(Note 1) Receive buffers to which devices are allocated

Example 3) When refreshing data by the periodical interrupt processing when the scan time is long and RPI is short. (When acquiring for each received data)

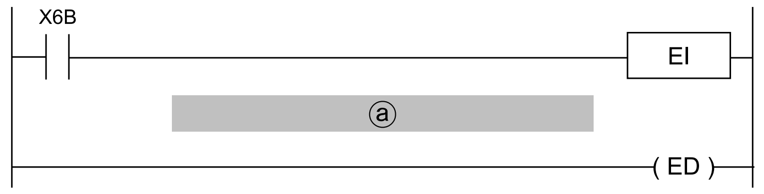
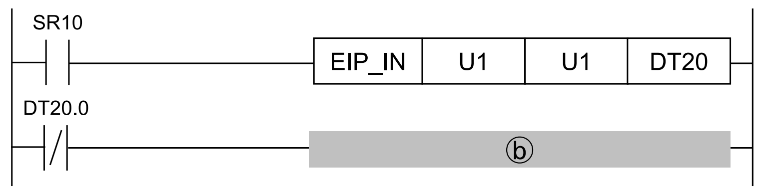
Scan time: 10 ms,

RPI (transmission interval for the EtherNet/IP cyclic communication): 500 μs

- When the scan time is longer than the setting time of RPI, the refresh cannot be executed during the processing. In this case, describe the EIP_IN instruction in a fixed cycle execution type PB and use interrupt processing to execute the refresh.
- If the interrupt cycle is set to the same value as that of RPI, the refresh instruction may be executed while the receive buffer is being written, and the operation may fail. Perform the processing after checking the refresh result.

EtherNet/IP configuration setting

Setting item	Settings
Node number	1
Connection	1
Input information (T>0) device allocation	LD0 to LD1

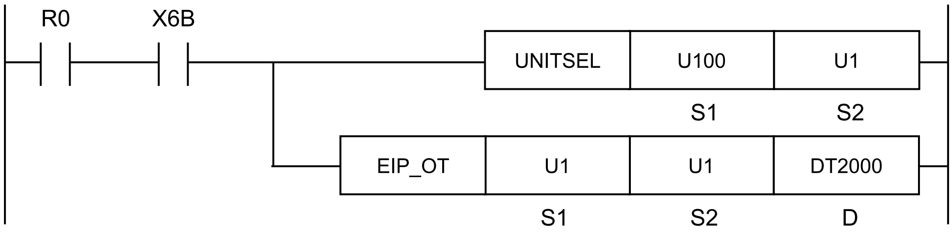
<table>
<tr>
<td rowspan="2">Every scan execution
Program block</td>
<td colspan="2">
</td>
</tr>
<tr>
<td>(a)</td>
<td>Processing that takes 10 ms or more</td>
</tr>
<tr>
<td rowspan="2">Constant period execution time
(0.1 ms units)
Program block
0.5-ms cyclic program</td>
<td colspan="2">
</td>
</tr>
<tr>
<td>(b)</td>
<td>Only when refresh is completed successfully, data can be adopted (LD0 and LD1 in this case).</td>
</tr>
</table>

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the unit specified by UNITSEL is not the built-in ET-LAN.
	To be set when the EtherNet/IP function is set to Not Use in the Ethernet unit configuration.
	To be set in case of out-of-range values in indirect access (index modification).
	Set when a value outside the range is specified for the parameter.
	To be set when the node that is specified by [S1] or the connection that is specified by [S2] does not exist.
	Use this instruction only for the connections in which the refresh method of the EtherNet/IP setting is set to Instruction. An operation error occurs when the connection that other refresh method other than that has been specified is specified.
	To be set when the connection for which the number of input data is 0 is specified.
	To be set when the connection for which the number of refreshed data is 0 is specified.

17.38 EIP_OT (EtherNet/IP Output Refresh)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ List of operands

Operand	Description
S1	Specify a target node number of output fresh.
S2	Specify a target connection number of output refresh.
D	Specify the device address storing refresh results.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●								●	●				●
S2	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

■ Processing

- This instruction executes the output refresh for connections to be refreshed. "Output refresh" means that the data is copied from the allocated devices to the send buffers.

■ Precautions for programming

- Execute this instruction after the EtherNet/IP preparation done flag (X6B) turns ON. If the instruction is executed before the flag turns ON, the EtherNet/IP communication preparation incomplete error occurs.
- This instruction causes a processing load. Do not execute the instruction successively in one scan.
- Before executing this instruction, use the cyclic communication normal node table to confirm that the communication of the specified connection is performed normally. The cyclic

communication normal node table can be checked by using the "ETSTAT (Acquiring EtherNet/IP Information)" instruction.

- Use this instruction only for the connections in which the refresh method of the "EtherNet/IP setting" is set to "Instruction" by the tool software. An operation error occurs if the batch refresh method or the division refresh method is specified.

■ Operand [S1] setting

- Specify the node number that data is set to the send buffer.
- The I/O map is used for sending data to a destination scanner device (PLC).

	Set value
I/O map	0
Scan List	1 to 256

■ Operand [S2] setting

Specify a connection number to be refreshed. Specify a relative number within nodes for the connection number.

	Set value
I/O map number or connection number	1 to 256

■ Operand [D] setting

- Specify the device address storing refresh results.
- If this instruction is executed in a cycle faster than RPI, the output refresh may not be performed.

Execution result	Description
0	Refresh operation is complete successfully.
1	Refresh is not performed.
2	EtherNet/IP communication preparation incomplete

■ Usage example

Example 1) When performing the output refresh for the send buffer of the I/O map number 1 (Normal end)

[S1]... U0 [S2]... U1 [D]... DT10

- EtherNet/IP configuration setting

Setting item	Settings
I/O map number	1
Device Allocation	LD10 to LD11
	LD14 to LD16

	Value
LD10	2233h
LD11	4455h
LD12	
LD13	
LD14	AABBh
LD15	CCDDh
LD16	EEFFh
LD17	

Send buffer	Value	
0		
1	2233h	*
2	4455h	*
3		
4		
5	AABBh	*
6	CCDDh	*
7	EEFFh	*
8		

	Value
DT10	0

(Note 1) Send buffers to which devices are allocated

Example 2) When performing the output refresh for the send buffer of the connection number 5 of the node number 2 (Abnormal end)

[S1]... U2 [S2]... U5 [D]... DT100

- EtherNet/IP configuration setting

Setting item	Settings
Node number	2
Connection	5
Output Information (O>T) Device Allocation	LD101 to LD102
	LD104 to LD107

	Value
LD100	
LD101	2233h
LD102	4455h
LD103	
LD104	8899h
LD105	AABBh
LD106	CCDDh
LD107	EEFFh
LD108	

	Send buffer	Value	
	0		
	1		
x	2	AABBh	*
x	3	CCDDh	*
	4		
x	5	AABBh	*
x	6	CCDDh	*
x	7	EEFFh	*
x	8	1122h	*

	Value
DT10	1

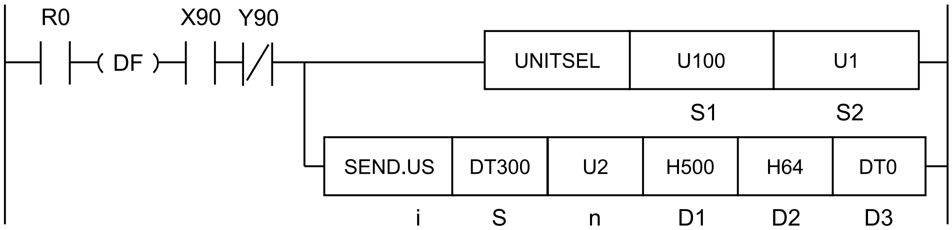
(Note 1) Send buffers to which devices are allocated

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set when the unit specified by UNITSEL is not the built-in ET-LAN.
	To be set when the EtherNet/IP function is set to Not Use in the Ethernet unit configuration.
	To be set in case of out-of-range values in indirect access (index modification).
	Set when a value outside the range is specified for the parameter.
	To be set when the I/O map, node, or connection that is specified by [S1] or [S2] does not exist.
	Use this instruction only for the connections in which the refresh method of the EtherNet/IP setting is set to Instruction. An operation error occurs when the connection that other refresh method other than that has been specified is specified.
	To be set when the connection for which the amount of output data is 0 is specified.
	To be set when the connection for which the number of refreshed data is 0 is specified.

17.39 SEND (MC Protocol Master)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Items	Settings		Setting range
S	Starting address of the sender data area		-
n	Amount of sent data		1 to 960 words 1 to 7168 bits
D1	Type and upper address of the destination device of the partner unit		
	Higher byte	Device type (Hexadecimal 2-digit)	H0 to H8
	Lower byte	Upper address of the device (Hexadecimal 2-digit)	H00 to HFF
D2	Lower address of the device for the partner unit. (Hexadecimal 4-digit)		H0 to HFFFF (0 to 65535)
D3	Starting address of the device area of the master unit that stores the execution result code (1 word)		

■ Available word devices (●: Available)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●													●
n	●	●	●	●			●	●								●	●				●
D1	●	●	●	●			●	●								●	●				●

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D2	●	●	●	●			●	●								● (Note 1)	● (Note 1)				●
D3	●	●	●	●			●	●													●

(Note 1) Only when "direct addressing" in the MC protocol mode is set, integers can be specified for a destination address.

■ Available bit devices (●: Available)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
S	●	●	●	●								●	●	●
n														
D1														
D2														
D3														

(Note 1) Bit devices cannot be specified for the operands n, D1, D2, and D3.

■ Outline of operation

- This instruction sends commands from the communication port of the unit to send/receive data to/from devices that support "MC protocol".
- Data can be read and written by specifying station numbers and memory addresses and executing the SEND and RECV instructions in a user program, because PLC automatically creates messages according to the protocol.
- Select a communication mode in the configuration menu of the tool software FPWIN GR7.
- When the SEND instruction is executed, the data is read from the device that starts from [S] in the master unit and the data is stored in the address that starts from [D1] (upper address) + [D2] of the partner unit.
- The transfer method (register transfer/bit transfer) varies according to the device types that are specified by [S] and [D1].
- The execution result code is stored in the one-word area of the master unit that is specified by [D3].

■ Specification of [S], [n], [D1], and [D2]

- The transfer method and the amount of sent data [n] vary according to the type of the device on FP7 that is specified by the operand [S].

Type of FP7 device specified for [S]	Transfer method	No. of sent data [n]	Remark
16-bit device. WX, WY, WR, WL, DT, LD	Register transfer	1 to 960	
1-bit device. X, Y, R, L, DT.n, LD.n	Bit transfer	1 to 7168	When the number of sent data is odd, 4-bit dammy code H0 is added.

- The amount of sent data [n] is specified in words for the register transfer and in bits for the bit transfer.
- For the operand [D1], specify hexadecimal data that consists of the destination device code and the device address of a partner unit.

  Example) When the device code is 3 (internal relay) and the upper hexadecimal 2 digit of the device address is H00, specify H300.
- For the "high byte of [D1]" that is the device code of the partner unit, specify one of the following values.

Unit	Device type			High byte of [D1]
Bit	Input	X	Hexadecimal	H0
	Output	Y	Hexadecimal	H 1
	Link relay	B	Hexadecimal	H 2
	Internal relay	M	Decimal	H 3
	Latch relay	L	Decimal	H 4
Word	Data register	D	Decimal	H 5
	File register	R	Decimal	H 6
		ZR	Hexadecimal	H 7
	Link register	W	Hexadecimal	H8

- For the device address of the partner unit, specify six-digit hexadecimal (three-byte) data that consists of the low byte of [D1] and the value of [D2]. When the device address is in the range of H0 to H00FFFF, specify "H00" for the low byte of [D1].

■ Execution result code [D3]

Code	Description	Code	Description
H0	Normal end	H5	Response reception timeout
H1	The communication port is being used in the master communication.	H6	Reception error(Note 1)
H2	The communication port is being used in the slave communication.	H7	I/O allocation shortage error(Note 2)
H3	The number of master communication instructions simultaneously used is exceeded.	H8	The send buffer is being used.(Note 3)
H4	Transmission timeout		

(Note 1) It occurs when an abnormal telegram is received. When there is a format error in the header of an individual protocol, the communication discards the received data and a response reception timeout occurs.

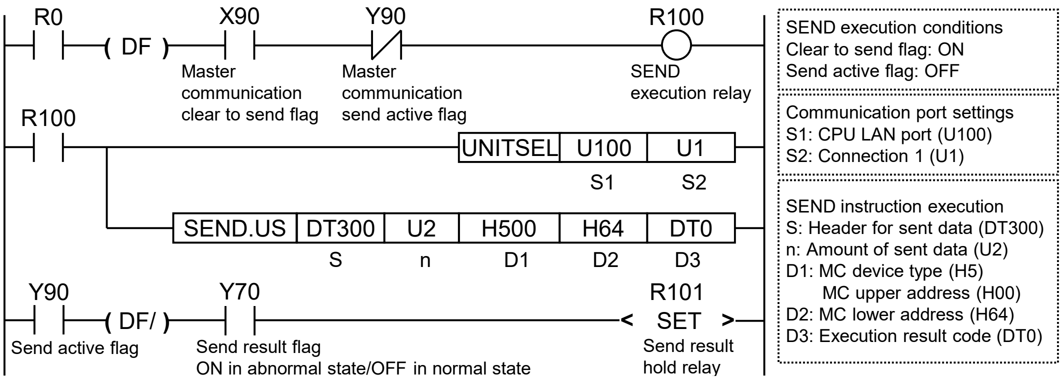
(Note 2) It occurs when the communication control I/O relays corresponding to the communication port (master communication clear to send flag, master communication send active flag, master communication send done result relay)are not allocated as I/O words of the CPU unit in the I/O map. It occurs only when the number of user connections of ET-LAN is expanded and this instruction is executed specifying the expanded connections.

(Note 3) Effective when the version is 4.57 or later.

(Note 4) For details of the execution result codes that may be set if a communication error occurs, refer to "Exit codes when communication error occurs""P.17-200".

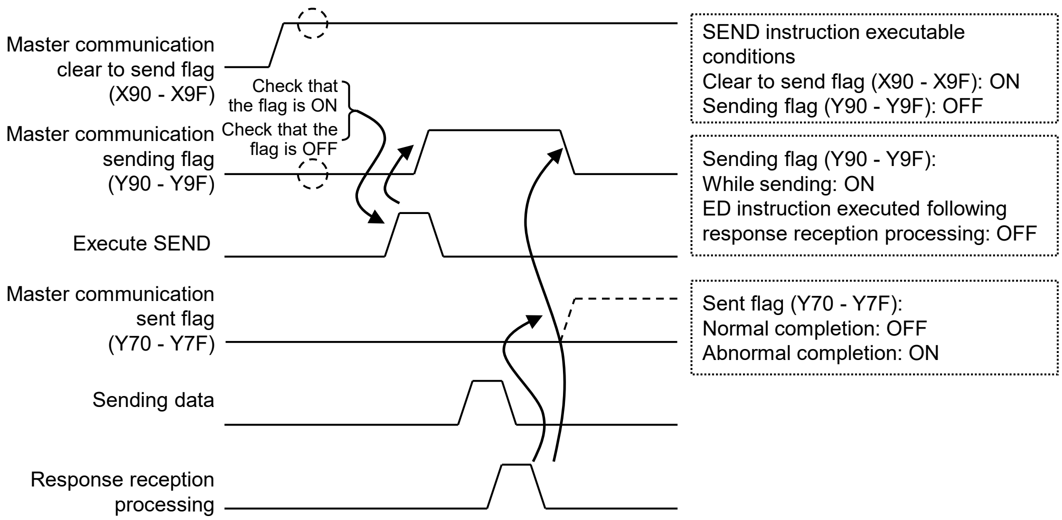
■ Sample program

- An MC protocol command (bulk write) is sent from the LAN port of the CPU unit, and the contents of the data registers DT300 to DT301 of FP7 are written to the addresses D000100 to D000101 of an external device.
- After it is confirmed that connection 1 is established in master mode (X90) and no transmissions are currently being executed for the same port (Y90), the SEND instruction is started.
- The UNITSEL instruction is used to specify a slot number (LAN port: U100) and the connection number (U1).
- The SEND instruction is executed with the address (DT300) and the number of data (U2) of FP7, the code (H50) that indicates the device type and the upper address of the partner unite, and the lower address (H64=100) of the partner unit.
- It is possible to check if a send error occurs by the sent flag (Y70) when the sending flag (Y90) turns OFF.



(Note 1) The above program example holds the send result hold relay (R101). Insert a program that resets the relay after the relay is checked.

■ Timing chart



■ I/O allocations

I/O number	Name	Description
X90 to X9F	Master communication clear to send flag	Turns ON when a connection is established in the master communication.
Y90 to Y9F	Master communication sending flag	Turns ON during sending data based on the SEND/RECV instruction. Turns OFF when the ED instruction is executed after the completion of the response receive processing.
Y70 to Y7F	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

(Note 2) The above I/O numbers are those for the connections 1 to 16.

■ Precautions for programming

- Use the UNITSEL instruction immediately before the SEND/RECV instruction to specify a target connection number for communication.
- Confirm the "master communication clear to send flag" of a corresponding connection is ON, and execute the SEND/RECV instruction. The master communication clear to send flag will not turn ON until the connection to the partner is established. It is recommended to specify to enable the auto open function in the connection setting of ET-LAN. Also, the connections can be connected with OPEN instruction.
- The SEND or RECV instruction cannot be executed for the connection for which the master communication is in progress. Confirm that the "master communication sending flag" is OFF, and execute the instructions.
- The SEND or RECV instruction cannot be executed for the connection for which the slave communication is in progress. (such as performing a data request from a host computer).
- Up to 16 instructions can be executed simultaneously for different COM ports and connections. (The total of simultaneous usage of SEND, RECV, pGPSEND, GPTRNS, and pPMSET instructions.)

■ Exit codes when communication error occurs

When a wrong command is sent or an error occurs in the CPU unit, a different exit code is returned. Exit codes in an error state are as follows.

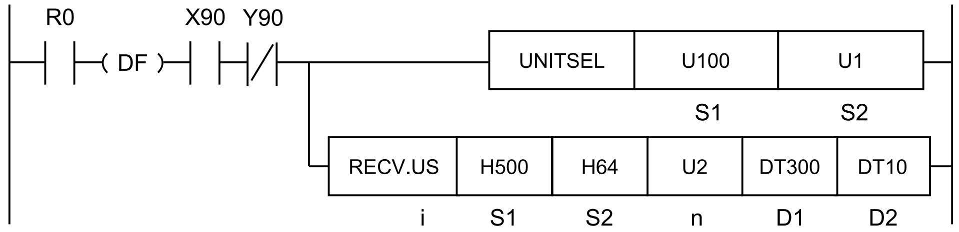
Code	Timing of occurrence
4031	Address is too long (Starting device + Number of written points)
C051	The number of devices is outside the specified range.
C056	The starting device is outside the specified range.
C059	Command search: There is no command that matches the receive data command in the MC protocol command table.
C059	The subcommand is outside the specified range.
C05B	The device code is outside the specified range.
C05C	Subcommand is in bit unit (0001) and device code is word device.
C05F	Receive header check: "Network number" check
C05F	Receive header check: "PC number" check
C05F	Receive header check: "Destination unit I/O number" check
C05F	Error in the number of received and written data
C060	Error in written contact data (except 0/1)
C061	Receive header content check: the number of receive data is smaller than the minimum received bytes that support header content check.
C061	The number of receive data is smaller than the minimum number of receive bytes.
**50 (Note) No reference destination	Receive header check: When a value other than 0x5000 is specified for the sub header, the value consisting of the first byte of the subheader plus 0x80 is inserted in the high byte "**" of the error code.

■ Flag operations

Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the source range is outside the accessible range.
	To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN).
	When there is not a connection that is specified with the UNITSEL instruction, the connection is closed.
	To be set when the data device specified by S is incorrect or exceeds the area.
	The amount of sent data specified by n is incorrect.
	The device code and the destination upper address specified by D1 is out of the range.
	The destination lower address specified by D2 is out of the range.
	The device in which results are stored specified by D3 is incorrect.

17.40 RECV (MC Protocol Master)

■ Ladder diagram



(Note 1) The above figure shows the case that S1=U100 (built-in ET-LAN in the CPU unit) and S2=U1 (connection number 1) are specified by the UNITSEL instruction.

■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Items	Settings		Setting range
S1	Type and upper address of the source device of the partner unit		
	Higher byte	Device type (Hexadecimal 2-digit)	H0 to H8
	Lower byte	Upper address of the device (Hexadecimal 2-digit)	H00 to HFF
S2	Lower address of the device for the partner unit. (Hexadecimal 4-digit)		H0 to HFFFF (0 to 65535)
n	Number of received data		1 to 960 words 1 to 7168 bits
D1	Starting address of the device area in the master unit that stores the received data		-
D2	Starting address of the device area of the master unit that stores the execution result code (1 word)		

■ Available word devices (●: Available)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TSCS	TECE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●								●	●				●
S2(Note 1)	●	●	●	●			●	●								●	●				●

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
																(Note 1)	(Note 1)				
n	●	●	●	●			●	●								●	●				●
D1	●	●	●	●			●	●													●
D2	●	●	●	●			●	●													●

(Note 1) Only when "direct addressing" in the MC protocol mode is set, integers can be specified for a destination address.

■ Available bit devices (●: Available)

Operand	Bit device											Specification of bit of word device		Index modifier
	X	Y	R	L	T	C	P	E	SR	IN	OT	DT.n	LD.n	
S1														
S2														
n														
D1	●	●	●	●								●	●	●
D2														

(Note 1) Bit devices cannot be specified for the operands S1, S2, n, and D2.

■ Outline of operation

- This instruction sends commands from the communication port of the unit to send/receive data to/from devices that support "MC protocol".
- Data can be read and written by specifying station numbers and memory addresses and executing the SEND and RECV instructions in a user program, because PLC automatically creates messages according to the protocol.
- Select a communication mode in the configuration menu of the tool software FPWIN GR7.
- When the RECV instruction is executed, the data is read from the address that starts from [S1] (upper address) + [S2] in the partner unit and the data is stored in the area that starts from [D1] in the master unit.
- The transfer method (register transfer/bit transfer) varies according to the device types that are specified by [S1], [S2], and [D1].
- The execution result code is stored in the one-word area of the master unit that is specified by [D2].

■ Precautions for programming

- Use the UNITSEL instruction immediately before the SEND/RECV instruction to specify a target connection number for communication.

- Confirm that the master communication clear to send flag is ON for the corresponding connection, and execute the SEND/RECV instruction. The master communication clear to send flag will not turn ON until the connection to the partner is established. It is recommended to specify to"enable the auto open function"in the connection setting of ET-LAN. Also, the connections can be connected with OPEN instruction.
- The SEND or RECV instruction cannot be executed for the connection for which the master communication is in progress. Confirm that the master communication sending flag is OFF, and execute the instructions.
- The SEND or RECV instruction cannot be executed for the connection for which the slave communication is in progress. (such as performing a data request from a host computer).
- Up to 16 instructions can be executed simultaneously for different COM ports and connections. (The total of simultaneous usage of SEND, RECV, pGPSEND, GPTRNS, and pPMSET instructions.)

Specification of [S1], [S2], [n], and [D1]

- For the operand [S1], specify hexadecimal data that consists of the source device code and the upper device address of the partner unit.

  Example: When the device code is 3 (internal relay) and the two-digit hexadecimal of the upper device address is H00, specify H300.
- Specify one of the following values as the "high byte of [S1]" that indicates the device code of the partner unit.

Unit	Device type			High byte of [S1]
Bit	Input	X	Hexadecimal	H0
	Output	Y	Hexadecimal	H 1
	Link relay	B	Hexadecimal	H 2
	Internal relay	M	Decimal	H 3
	Latch relay	L	Decimal	H 4
Word	Data register	D	Decimal	H 5
	File register	R	Decimal	H 6
		ZR	Hexadecimal	H 7
	Link register	W	Hexadecimal	H8

- For the device address of the partner unit, specify six-digit hexadecimal (three-byte) data that consists of the low byte of [S1] and the value of [S2]. When the device address is in the range of H0 to H00FFFF, specify "H00" for the low byte of [S1].
- The number of received data [n] is specified in words for the register transfer and in bits for the bit transfer.
- The transfer method and the number of received data [n] vary according to the device code on FP7 that is specified by the operand [D1].

Type of FP7 device specified for [D1]	Transfer method	Number of received data[n]	Remark
16-bit device. WX, WY, WR, WL, DT, LD	Register transfer	1 to 960	
1-bit device. X, Y, R, L, DT.n, LD.n	Bit transfer	1 to 7168	When the number of receive data is an odd number, a four-bit dummy code H0 is added.

■ Execution result code [D2]

Code	Description	Code	Description
H0	Normal end	H5	Response reception timeout
H1	The communication port is being used in the master communication.	H6	Reception error(Note 1)
H2	The communication port is being used in the slave communication.	H7	I/O allocation shortage error(Note 2)
H3	The number of master communication instructions simultaneously used is exceeded.	H8	The send buffer is being used.(Note 3)
H4	Transmission timeout		

(Note 1) It occurs when an abnormal telegram is received. When there is a format error in the header of an individual protocol, the communication discards the received data and a response reception timeout occurs.

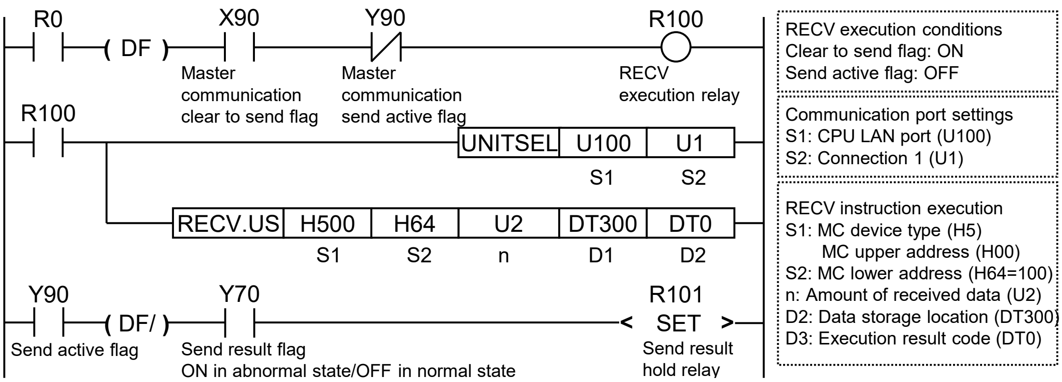
(Note 2) It occurs when the communication control I/O relays corresponding to the communication port (master communication clear to send flag, master communication send active flag, master communication send done result relay)are not allocated as I/O words of the CPU unit in the I/O map. It occurs only when the number of user connections of ET-LAN is expanded and this instruction is executed specifying the expanded connections.

(Note 3) Effective when the version is 4.57 or later.

(Note 4) For details of the execution result codes that may be set if a communication error occurs, refer to "Exit codes when communication error occurs""P.17-206".

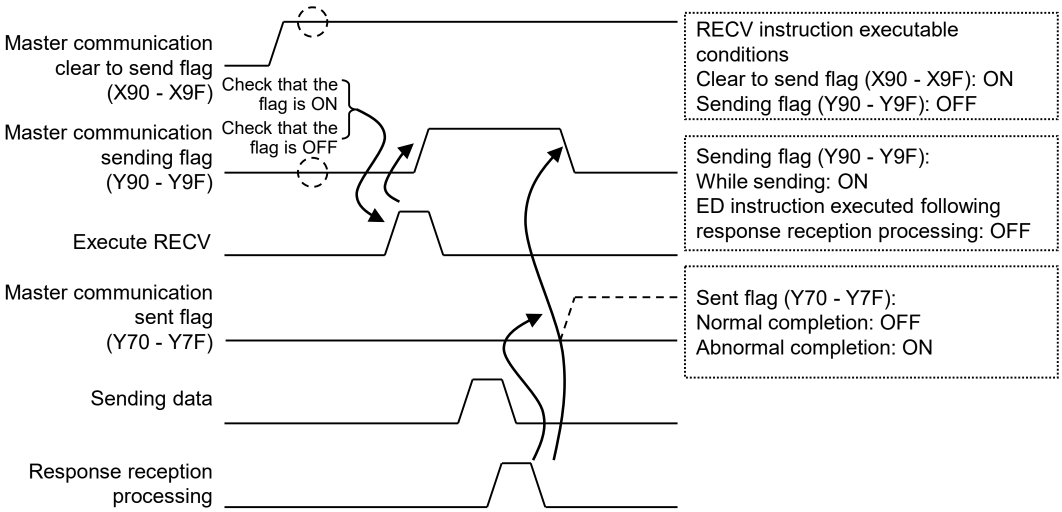
■ Sample program

- An MC protocol command (bulk read) is sent from the LAN port of the CPU unit, and data in the addresses D000100 to D000101 of an external device is read and stored in the data registers DT300 to DT301 of FP7.
- After it is confirmed that connection 1 is established in master mode (X90) and no transmissions are currently being executed for the same port (Y90), the RECV instruction is started.
- The UNITSEL instruction is used to specify a slot number (LAN port: U100) and the connection number (U1).
- The RECV instruction is executed with the code (H500) that indicates the device type and the upper address of the partner unit, the lower address (H64=100), the number of data (U2), and the storage address (DT300) of FP7.
- It is possible to check if a send error occurs by the sent flag (Y70) when the sending flag (Y90) turns OFF.



(Note 1) The above program example holds the send result hold relay (R101). Insert a program that resets the relay after the relay is checked.

■ Timing chart



■ I/O allocations

I/O number	Name	Description
X90 to X9F	Master communication clear to send flag	Turns ON when a connection is established in the master communication.
Y90 to Y9F	Master communication sending flag	Turns ON during sending data based on the SEND/RECV instruction. Turns OFF when the ED instruction is executed after the completion of the response receive processing.
Y70 to Y7F	Sent flag	Reports completion result of sending data in genera-purpose communication or master communication. (Normal completion: 0, Abnormal completion: 1)

(Note 1) Each contact is used for reading the operation state. Do not write over it with a user program.

(Note 2) The above I/O numbers are those for the connections 1 to 16.

Exit codes when communication error occurs

When a wrong command is sent or an error occurs in the CPU unit, a different exit code is returned. Exit codes in an error state are as follows.

Code	Timing of occurrence
4031	Address is too long (Starting device + Number of written points)
C051	The number of devices is outside the specified range.
C056	The starting device is outside the specified range.
C059	Command search: There is no command that matches the receive data command in the MC protocol command table.
C059	The subcommand is outside the specified range.
C05B	The device code is outside the specified range.
C05C	Subcommand is in bit unit (0001) and device code is word device.
C05F	Receive header check: "Network number" check
C05F	Receive header check: "PC number" check
C05F	Receive header check: "Destination unit I/O number" check
C05F	Error in the number of received and written data
C060	Error in written contact data (except 0/1)
C061	Receive header content check: the number of receive data is smaller than the minimum received bytes that support header content check.
C061	The number of receive data is smaller than the minimum number of receive bytes.
**50	Receive header check: When a value other than 0x5000 is specified for the sub header, the value consisting of the first byte of the subheader plus 0x80 is inserted in the high byte "**" of the error code.

Flag operations

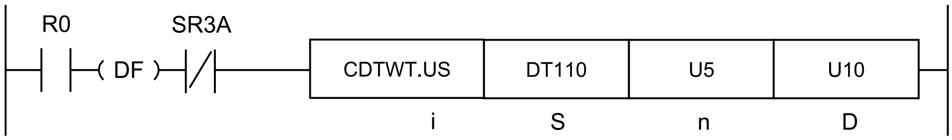
Name	Description
SR7 SR8 (ER)	To be set in the case of out-of-range in indirect access (index modification).
	To be set when the destination range is outside the accessible range.
	To be set when the slot number [S1] specified with UNITSEL is not 100 (built-in ET-LAN).
	When there is not a connection that is specified with the UNITSEL instruction, the connection is closed.
	The device code and the source upper address specified by S1 is out of the range.
	The source lower address specified by S2 is out of the range.
	The amount of sent data specified by n is incorrect.
	The data device in the receiver data area in master unit specified by D1 is incorrect or exceeds the area.
	The device in which results are stored specified by D2 is incorrect.

18 High-level Instructions (SD Memory Card)

Applicable Models: CPU units except CPS2R

18.1 CDTWT (Operation Memory File Write in BIN Format)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S	Starting address of the memory device for operation in which data to be written is stored
n	Number of data items to be written. Range: 0 to 65535
D	File number (3 digits) that is added to the file name of the file to be created or overwritten. Range: 0 to 999

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●													●
n	●	●	●	●			●	●							●	●	●				●
D	●	●	●	●			●	●							●	●	●				●

(Note 1) Only 16-bit devices can be modified. (Integer constants cannot be specified.)

■ Outline of operation

- Binary data of [n] words is read from the area starting with [S], and the data is written on an SD memory card as a binary format file.
- The folder name is \data, the file name is dtxxx.bin. The number specified by the operand [D] is used for "xxx" of the file name.
- When there is no specified folder, create a folder. When the file already exists, overwrite it.

■ Example of processing

- Five words from the device DT110 specified by [S] are read, and are written in the folder \data in an SD memory card as a binary format file (bin.).

- The file name is "dt010.bin". The file number 10 specified by [D] is added.

[i]…US
[S] …DT110 [n] …5 [D] …10

DT108	H 0108
DT109	H 0109
DT110	**H 0110**
DT111	**H 0111**
DT112	**H 0112**
DT113	**H 0113**
DT114	**H 0114**
DT115	H 0115
DT116	H 0116
DT117	H 0117
DT118	H 0118

File dt**010**.bin (16-bit binary format)

1001110112 0113011401

Precautions for programming

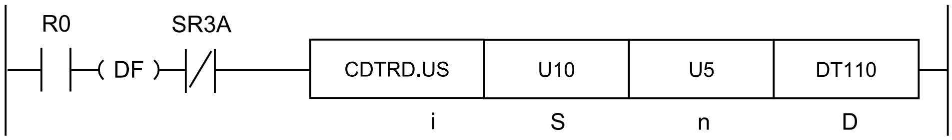
- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".
- When the set file attribute is read only, any data cannot be written.
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.
- Confirm that the SD memory card access instruction execution done flag (SR3B) is turned OFF, and turn OFF the execution condition.

Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when an out-of-range value is specified for [n].
	To be set when an out-of-range value is specified for [D].

18.2 CDTRD (Data Read from BIN Format File to Operation Memory)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S	File number (3 digits) of the file in an SD memory card in which the data to be read is stored. Range: 0 to 999
n	Number of data items to be read. Range: 0 to 65535
D	Starting address of the device for arithmetic operations that stores the data to be read

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●							●	●	●				●
n	●	●	●	●			●	●							●	●	●				●
D	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices can be modified. (Integer constants cannot be specified.)

■ Outline of operation

- This instruction reads [n] pieces of data from the binary format file in the SD memory card, and stores it in the device of the address starting with [D].
- The folder name is \data, and the file name is dtxxx.bin. xxx in the file name is the file number specified by the operand [S].

■ Example of processing

- A binary format file is read from the folder \data in the SD memory card, and is stored in the device for operation starting with [D].

- The file name of the binary format file is "dt010.bin". The file number 10 specified by [D] is added.

[i]…US
[S] …10 [n] …5 [D] …DT110

DT108	H 0108
DT109	H 0109
DT110	**H 0110**
DT111	**H 0111**
DT112	**H 0112**
DT113	**H 0113**
DT114	**H 0114**
DT115	H 0115
DT116	H 0116
DT117	H 0117
DT118	H 0118

File dt**010**.bin (16-bit binary format)

25022602270228022902

→

DT108	H 0108
DT109	H 0109
DT110	**H 0225**
DT111	**H 0226**
DT112	**H 0227**
DT113	**H 0228**
DT114	**H 0229**
DT115	H 0115
DT116	H 0116
DT117	H 0117
DT118	H 0118

■ Precautions for programming

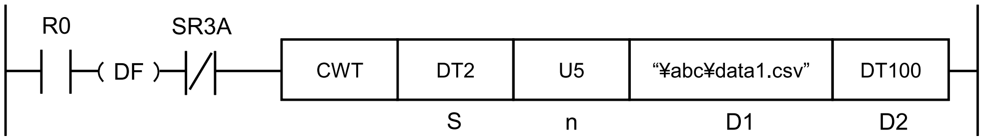
- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.
- During the execution of the CDTRD instruction, data values read from the SD memory card are written from the beginning of a specified data device area in order. Therefore, do not read the data in the range of data device processed by the CDTRD instruction after the start of the reading process until the completion.
- When the number of data of the stored file is less than the specified number of data to be read, data is read up to the number of data of the file.
- An error occurs when there is no folder, or no file with a specified file number in the folder.

■ Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when an out-of-range value is specified for [n].
	To be set when an out-of-range value is specified for [S].

18.3 CWT (File Data Write Instruction)

■ Ladder diagram



■ List of operands

Operand	Description
S	Starting address of the device where data to be written is stored (data format: unsigned 16-bit integer)
n	Number of write data items (data format: Unsigned 16-bit integer)
D1	Starting address of the device area that stores the path name (folder name + file name: maximum 256 characters) of the file in which data is written and the number of characters of the path name (data format: character data)
D2	Starting address of the device area that stores parameters related to information such as saving format (data format: unsigned 16-bit integer)

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	•	•	•	•			•	•													•
n	•	•	•	•			•	•								•	•				•
D1	•	•	•	•			•	•												•	•
D2	•	•	•	•			•	•													•

(Note 1) Only 16-bit devices can be modified. (Integer constants and character constants cannot be specified.)

■ Outline of operation

- [n] items of data stored at the device address starting from [S] are read and written to the SD memory card file specified by [D1] according to the parameters specified by [D2].

■ [n]: Specification of number of write data items

Saving format	Set value of [D2]	Setting range of [n]
16-bit data	U1, U2, U7, U11	0 to 65535
32-bit data	U3, U4, U5, U8	0 to 32767
64-bit data	U6, U9	0 to 16383

Saving format	Set value of [D2]	Setting range of [n]
ASCII	U10	0 to 1999

(Note 1) When “0 is specified for [n], the result is as follows.

1: When creating a new file, a 0-byte file is created.

2: When rewriting a file, a 0-byte file is created.

3: When appending to a file, only the file date is changed.

■ [D1]: Specification of folder and file name

Setting device	Description
D1	Specifies the number of characters used in the folder name and file name of the file to be written. (Specify the full path.)
From D1+1	Specify the folder name and the file name to be written. Specify the full path, up to 256 characters including the folder name and filename

(Note 1) In the tool software FPWIN GR7, a path name (a folder name and a file name) can be entered directly as a character constant.

(Note 2) To specify a memory area such as a data register DT, use the SSET instruction to store a path name (a folder name and a file name) as character data.

■ [D2] to [D2+6]: Specification of writing format

<table>
<tr><th>Operand</th><th>Items</th><th colspan="5">Description</th></tr>
<tr><td rowspan="12">D2</td><td rowspan="12">Write format</td><td>Set value of D2</td><td colspan="2">Write contents</td><td>Fixed number of digits</td><td>Extension</td></tr>
<tr><td>H0</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>U1</td><td rowspan="4">DEC</td><td>Unsigned 16-bit integer</td><td>5</td><td rowspan="10">.CSV
(comma-separated text)</td></tr>
<tr><td>U2</td><td>Signed 16-bit integer</td><td>6</td></tr>
<tr><td>U3</td><td>Unsigned 32-bit integer</td><td>10</td></tr>
<tr><td>U4</td><td>Signed 32-bit integer</td><td>11</td></tr>
<tr><td>U5</td><td rowspan="2">Floating point real number</td><td>32bit</td><td>13</td></tr>
<tr><td>U6</td><td>64bit</td><td>23</td></tr>
<tr><td>U7</td><td rowspan="3">HEX</td><td>1 words</td><td>4</td></tr>
<tr><td>U8</td><td>2 words</td><td>8</td></tr>
<tr><td>U9</td><td>4 words</td><td>16</td></tr>
<tr><td>U10</td><td>ASCII</td><td>Character string</td><td>-</td></tr>
<tr><td></td><td></td><td>U11</td><td>BIN</td><td>16bit</td><td>-</td><td>.BIN (BIN data)</td></tr>
<tr><td>D2+1</td><td>Write mode
(Note 1)</td><td colspan="2">0: New file mode</td><td colspan="3">Deletes the file contents, and then writes data. When no file exists, a new one is created.</td></tr>
</table>

Operand	Items	Description	
		1: Add mode	Writes additional data from the end of the file. When no file exists, a new one is created.
		2: Write position specification mode 1	Writes data from the position after the number of bytes stored in [S2+3] and [S2+4] from the head of the file.
		3: Write position specification mode 2	Writes data from the position after the number of bytes stored in [S2+3] and [S2+4] from the end of the file.
D2+2	Options (Note 2)	-	
D2+3 D2+4	Writing position (file pointer) Number of bytes from the beginning or end of the file (Note 3)	● This is effective when the write position specification mode is selected for [D2+1]. ● The setting of the writing position (file pointer) indicates the position in bytes from the head of the stored file. ● In the case of writing position specification mode 1, when completing writing to an SD memory card, the end position of the newly-saved data counting from the head of the file is stored in the areas [D2+3] and [D2+4]. ● In the case of writing position specification mode 2, when completing writing to an SD memory card, the end position of the newly-saved data counting from the end of the file is stored in the areas [D2+3] and [D2+4].	
D2+5 D2+6	Number of written data	Stores the number of data items that could be written after writing to the file. Example 1: If the number of data items to be written is 40 and the file has 100 free spaces, 40 is stored for the number of written data. Example 2: If the number of data items to be written is 40 and the file has 30 free spaces, 30 is stored for the number of written data. Example 3: If the number of characters to be written is 40 and the file has free space for 100 characters, 40 is stored for the number of characters written. Example 4: If the number of characters to be written is 40 and the file has free space for 30 characters, 30 is stored for the number of characters written.	

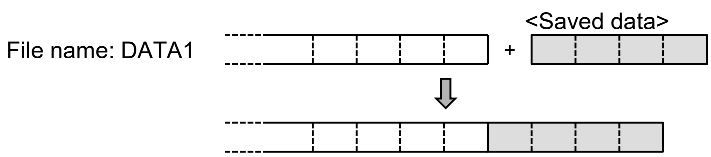
(Note 1) Refer to "[D2+1]: Specification of writing mode".

(Note 2) Refer to "[D2+2]: Specification of options".

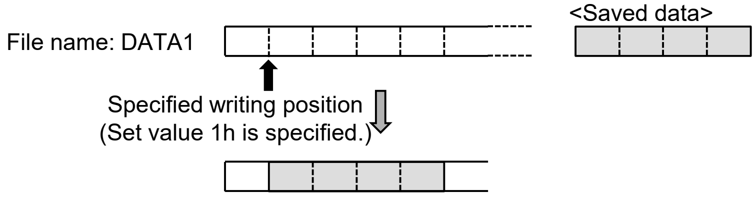
(Note 3) Refer to "[D2+3] and [D2+4]: Specification of writing position".

■ [D2+1]: Specification of write mode

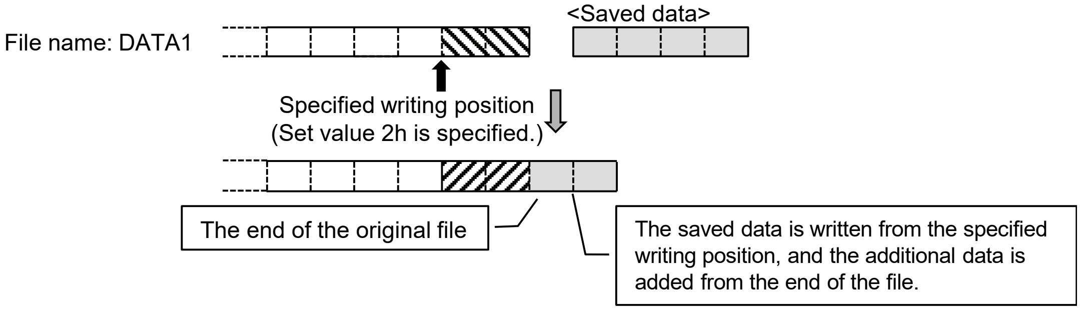
Example 1: When the addition a file is specified



Example 2) When specifying a writing position from the head of a file



Example 3) When specifying a writing position from the end of a file



■ [D2+2]: Specification of options

Specified bits	Description	
bit0 to 7	Line break	Line break setting for lines other than the end of the file when outputting to a CSV file. 0: Line break is not used except at the end of the file. 1 to 255: Insert a line break at each of the specified number of comma-separated data pieces. When U10 (ASCII) or U11 (BIN) is specified for [D2] saving method, the line break setting is invalid.
bit 8	Postfix	Specifies the data to be added to the end of the write data when outputting to a CSV file. 0: Insert a line break (0Dh+0Ah). 1: Insert a comma (2Ch). However, lines with line breaks do not have a trailing comma.
bit 9	Zero suppression	When outputting to a CSV file, specify whether to perform zero suppression or not. 0: Do not perform zero suppression. 1: Perform zero suppression. (Delete unnecessary zeros, and output right-aligned data.)
Bit10 to 15		Reserved for system (Zero is set.)

Example of option settings

- The following table shows how data is written depending on the value of bits 0 to 7 of [D2+2] for the writing format [D2] = 7 (HEX 16 bits), bit 9 of [D2+2] = 0 (Do not perform zero suppression), and written data "1 2 3 4 5".

[D2+2] bit 0 to 7	Written data									
0	0001	,	0002	,	0003	,	0004	,	0005	(Postfix, specified data)
1	0001	0Dh0Ah	0002	0Dh0Ah	0003	0Dh0Ah	0004	0Dh0Ah	0005	(Postfix, specified data)
2	0001	,	0002	0Dh0Ah	0003	,	0004	0Dh0Ah	0005	(Postfix, specified data)
3	0001	,	0002	,	0003	0Dh0Ah	0004	,	0005	(Postfix, specified data)
4	0001	,	0002	,	0003	,	0004	0Dh0Ah	0005	(Postfix, specified data)
5	0001	,	0002	,	0003	,	0004	,	0005	(Postfix, specified data)
6	0001	,	0002	,	0003	,	0004	,	0005	(Postfix, specified data)

Examples of conversion when zero suppression is ON and OFF

[D2] Specification of writing format		Number of digits	Zero suppression: ON	Zero suppression: OFF
1	Unsigned 16-bit integer	5	␣␣␣␣ 0	00000
2	Signed 16-bit integer	6	␣␣␣␣␣ 0	␣00000
			␣␣␣␣ - 1	-00001
3	Unsigned 32-bit integer	10	␣␣␣␣␣␣␣␣␣ 0	0000000000
4	Signed 32-bit integer	11	␣␣␣␣␣␣␣␣␣␣ 0	␣0000000000
			␣␣␣␣␣␣␣␣␣- 1	-0000000001
5	Floating point real number 32bit	13	␣␣␣␣␣␣␣␣␣␣␣␣ 0	␣000000000000
			␣␣␣␣␣␣␣␣␣␣␣ - 1	-000000000001
			␣␣␣␣␣␣␣␣1E-10	␣00000001E-10
			␣␣␣␣␣1.234567	␣00001.234567
			-3.402823E+38	-3.402823E+38
6	Floating point real number 64bit	23	␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣ 0	00000000000000000000000
			␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣ -1	-0000000000000000000001
			␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣1E-10	␣000000000000000001E-10
			␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣1.234567	␣000000000000001.234567
			␣1.797693134862315E+308	␣1.797693134862315E+308

[D2] Specification of writing format		Number of digits	Zero suppression: ON	Zero suppression: OFF
7	HEX 1-word	4	␣␣␣0	0000
8	HEX 2-word	8	␣␣␣␣␣␣␣0	00000000
9	HEX 4-word	16	␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣0	0000000000000000

(Note 1) "␣" in the table represents a space (H20).

■ [D2+3], [D2+4]: Specification of write position (file pointer)

16-bit integer bin. format	01	00	17	00	59	01	D7	11	D5	DD	01	00	17	00	59	01	FF	FF	
16-bit integer csv. format	(20H)	(20H)	(20H)	(20H)	1 (31H)	, (2CH)	(20H)	(20H)	(20H)	2 (32H)	3 (33H)	, (2CH)	(20H)	(20H)	3 (33H)	4 (34H)	5 (35H)	, (2CH)	
ASCII csv. format	“ (22H)	A (41H)	B (42H)	C (43H)	D (44H)	E (45H)	“ (22H)	, (2CH)	“ (22H)	a (61H)	b (62H)	c (63H)	d (64H)	“ (22H)	, (2CH)	“ (22H)	1 (31H)	2 (32H)	
Writing position (File pointer)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12

- The writing position (file pointer) is specified in hexadecimal format in bytes.
- The file pointer setting is available only when the writing position specification mode1 or mode 2 is specified.
- After the writing process into the file, the position of the end of written data is stored at the writing position (file pointer). If the writing operation is performed again in this state, the next data will be written.
- In case of the writing position specification mode 1, data is written in the file from the writing position (file pointer) counted from the head of the file.
- In case of the writing position specification mode 2, data is written in the file from the writing position (file pointer) counted from the end of the file.
- In the new file mode, data is always written from the head of a file. The writing position (file pointer) after the writing process is not stored.
- In the add mode, data is always written from the end of a file. The writing position (file pointer) after the writing process is not stored.
- The writing position is specified by one byte.

■ Example of processing (csv format file)

Example 1)

- Five pieces (five words) of unsigned 16-bit integer data are read from the area starting with device DT2, and the data pieces are written in CSV format into the file "\abc\data1.csv" in the SD memory card in new file mode.
- A blank line is inserted in the data through zero suppression, and a comma (2Ch) is inserted at the end.

[S]...DT2   [n]...5   [D1]..."¥abc¥data1.csv"   [D2]...DT100

DT0	U1	
DT1	U12	
DT2	**U123**	...Head of device to be read
DT3	**U1234**	
DT4	**U12345**	
DT5	**U5**	
DT6	**U56**	
DT7	U567	
DT8	U5678	
DT9	U56789	

DT100	**U1**	...Writing format
DT101	**H 0000**	...Writing mode
DT102	**H 0300**	...Option (*)
DT103 DT104	**H 00000000**	...Writing position
DT105 DT106	**U0**	...Amount of writing data

Writing file "data1.csv"

`␣␣123,␣1234,12345,␣␣␣␣5,␣␣␣56,`

"␣" indicates a space (20h)

DT105 DT106	**U5**	...Amount of writing data

(*) Option

- bit0-7(0): Line break settings - Do not insert line breaks except at the end of the file
- bit8(1): Postfix - Insert a comma (2Ch)
- bit9(1): Zero suppression - ON

Example 2)

- Five pieces (10 words) of unsigned 32-bit integer data are read from the area starting with device DT4, and the data pieces are written in CSV format into the file "\abc\data1.csv" in the SD memory card in new file mode.
- A blank line is inserted in the data through zero suppression, and a comma (2Ch) is inserted at the end.

[S]...DT4   [n]...5   [D1]..."¥abc¥data1.csv"   [D2]...DT100

DT0	U1234	
DT2	U12345	
DT4	**U123456**	...Head of device to be read
DT6	**U1234567**	
DT8	**U12345678**	
DT10	**U123456789**	
DT12	**U1234**	
DT14	U12345	
DT16	U123456	
DT18	U1234567	

Writing file "data1.csv"

`␣␣␣␣123456,␣␣␣1234567,␣␣12345678,␣123456789,␣␣␣␣␣␣1234 (0Dh)(0Ah)`

DT100	**U3**	...Writing format
DT101	**H 0000**	...Writing mode
DT102	**H 0200**	
DT103 DT104	**H 00000000**	...Writing position
DT105 DT106	**U0**	...Amount of writing data

DT105 DT106	**U5**	...Amount of writing data

(*) Option

- bit0-7(0): Line break settings - Do not insert line breaks except at the end of the file
- bit8(0): Postfix - Insert a line break (0Dh+0Ah)
- bit9(1): Zero suppression - ON

Example 3)

- Five pieces (five words) of unsigned 16-bit integer data are read from the area starting with device DT2, and the data pieces are written in CSV format into the file "\abc\data1.csv" in the SD memory card in new file mode.
- A line break (0Dh+0Ah) is inserted at the third data separation and at the end of the file.

[S]...DT2 [n]...5 [D1]..."¥abc¥data1.csv" [D2]...DT100

DT0	U1	
DT1	U12	
DT2	**U123**	...Head of device to be read
DT3	**U1234**	
DT4	**U12345**	
DT5	**U5**	
DT6	**U56**	
DT7	U567	
DT8	U5678	
DT9	U56789	

DT100	**U1**	...Writing format
DT101	**H 0000**	...Writing mode
DT102	**H 0003**	...Option (*)
DT103 DT104	**H 00000000**	...Writing position
DT105 DT106	**U0**	...Amount of writing data

Writing file "data1.csv"

00123,01234,12345 (0Dh)(0Ah)
00005,00056 (0Dh)(0Ah)

DT105 DT106	**U5**	...Amount of writing data

(*) Option

- ･bit0-7(3): Line break settings - Insert a delimiter and a break before the third data
- ･bit8(0): Postfix - Insert a line break (0Dh+0Ah)
- ･bit9(0): Zero suppression - OFF

Example 4)

- Five pieces (five characters) of ASCII data in the area starting with device DT2 are read from low bytes in order, and the data pieces are written in CSV format into the file "\abc\data1.csv" in the SD memory card in new file mode.
- A comma (2Ch) is inserted at the end of the file.

[S]...DT2 [n]...5 [D1]..."¥abc¥data1.csv" [D2]...DT100

DT0	"8"	"7"	
DT1	"0"	"9"	
DT2	**"b"**	**"a"**	...Head of device to be read
DT3	**"d"**	**"c"**	
DT4	"f"	**"e"**	
DT5	"h"	"g"	
DT6	"j"	"i"	
DT7	"l"	"k"	
DT8	"n"	"m"	
DT9	H 0000		

DT100	**U10**	...Writing format
DT101	**H 0000**	...Writing mode
DT102	**H 0100**	...Option (*)
DT103 DT104	**H 00000000**	...Writing position
DT105 DT106	**U0**	...Amount of writing data

Writing file "data1.csv"

"abcde",

DT105 DT106	**U5**	...Amount of writing data

(*) Option
- bit0-7(0): Line break settings - Invalid
- bit8(1): Postfix - Insert a comma (2Ch)
- bit9(0): Zero suppression - Invalid

Example 5)

- Six pieces (six characters) of ASCII data are read from the area starting with device DT2, and the data pieces are written in CSV format from the file pointer position of the existing file "\abc\data1.csv" in an SD memory card.
- A comma (2Ch) is inserted at the end of the file.

[S]...DT2 [n]...6 [D1]..."¥abc¥data1.csv" [D2]...DT100

Byte address	High	Low	
DT0	"8"	"7"	
DT1	"0"	"9"	
DT2	**"b"**	**"a"**	...Head of device to be read
DT3	**"d"**	**"c"**	
DT4	**"f"**	**"e"**	
DT5	"h"	"g"	
DT6	"j"	"i"	
DT7	"l"	"k"	
DT8	"n"	"m"	
DT9	H 0000		

DT100	**U10**	...Writing format
DT101	**H 0002**	...Writing mode
DT102	**H 0100**	...Option (*)
DT103, DT104	**H 00000007**	...Writing position
DT105, DT106	**U0**	...Amount of writing data

(*) Option

- ・bit0-7(0): Line break settings - Invalid
- ・bit8(1): Postfix - Insert a comma (2Ch)
- ・bit9(0): Zero suppression - Invalid

Writing file "data1.csv"

"abcdef**ghijklmno**pqrstuvwxyz",

File pointer

→

Writing file "data1.csv"

"abcdef**"abcdef",**pqrstuvwxyz",

File pointer

DT103, DT104	**H 00000010**	...Writing position
DT105, DT106	**U6**	...Amount of writing data

Example 6)

- 10000 pieces (10000 words) of signed 16-bit integer data are read from the area starting with device DT10000, and the data pieces are written in CSV format into "\FP7\DT.CSV" in the SD memory card in new file mode.
- An empty line is inserted in the data by zero suppression. Line breaks (0Dh+0Ah) are inserted as delimiters for the 10th data and the end of the data.

[S]...DT10000 [n]...10000 [D1]..."¥FP7¥DT.CSV" [D2]...DT50

DT10000	**K10000**	...Head of device to be read
DT10001	**K10001**	
DT10002	**K10002**	
.		
.		
.		
DT19997	**K19997**	
DT19998	**K19998**	
DT19999	**K19999**	
DT20000	U0	

DT50	**U2**	...Writing format
DT51	**H 0000**	...Writing mode
DT52	**H 020A**	...Option (*)
DT53 DT54	**H 00000000**	...Writing position
DT55 DT56	**U0**	

(*) Option

- bit0-7(0Ah): Line break settings ・Insert line break after 10th data
- bit8(0): Postfix ・Insert line break (0Dh+0Ah)
- bit9(1): Zero suppression ・On

Writing file "¥FP7¥DT.CSV"

```
˽10000,˽10001,˽10002,˽10003,˽10004,˽10005,˽10006,˽10007,˽10008,˽10009 0Dh0Ah
˽10010,˽10011,˽10012,˽10013,˽10014,˽10015,˽10016,˽10017,˽10018,˽10019 0Dh0Ah
                                   .
                                   .
                                   .
˽19990,˽19991,˽19992,˽19993˽19994,˽19995,˽19996,˽19997,˽19998,˽19999 0Dh0Ah
```

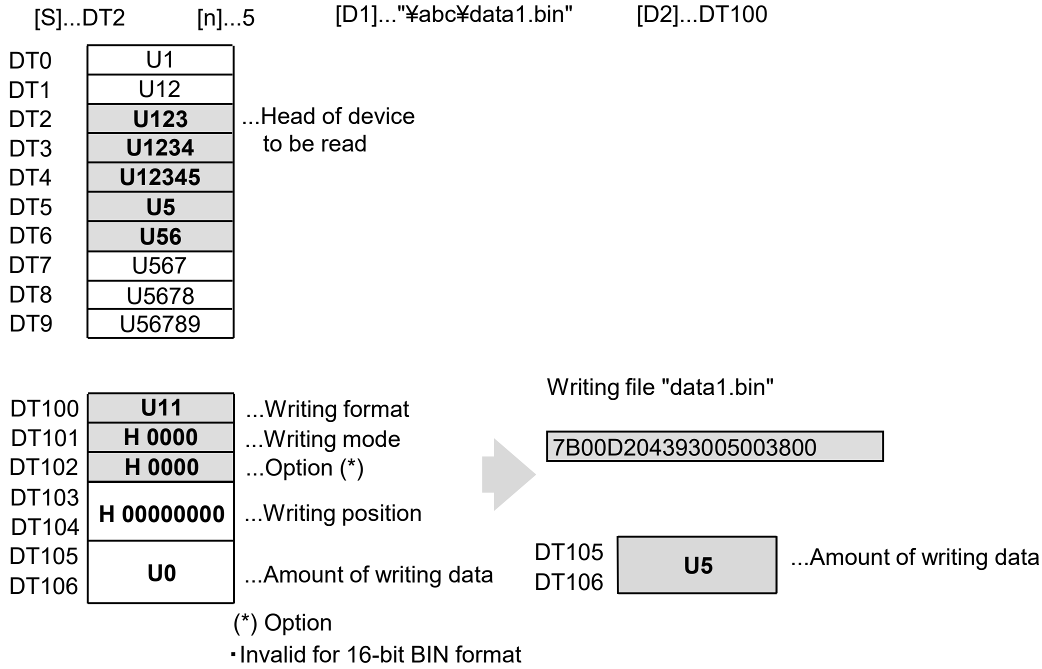
"˽ " indicates a space (20h)

DT55 DT56	**U10000**	...Amount of writing data

■ Processing examples (.bin format file)

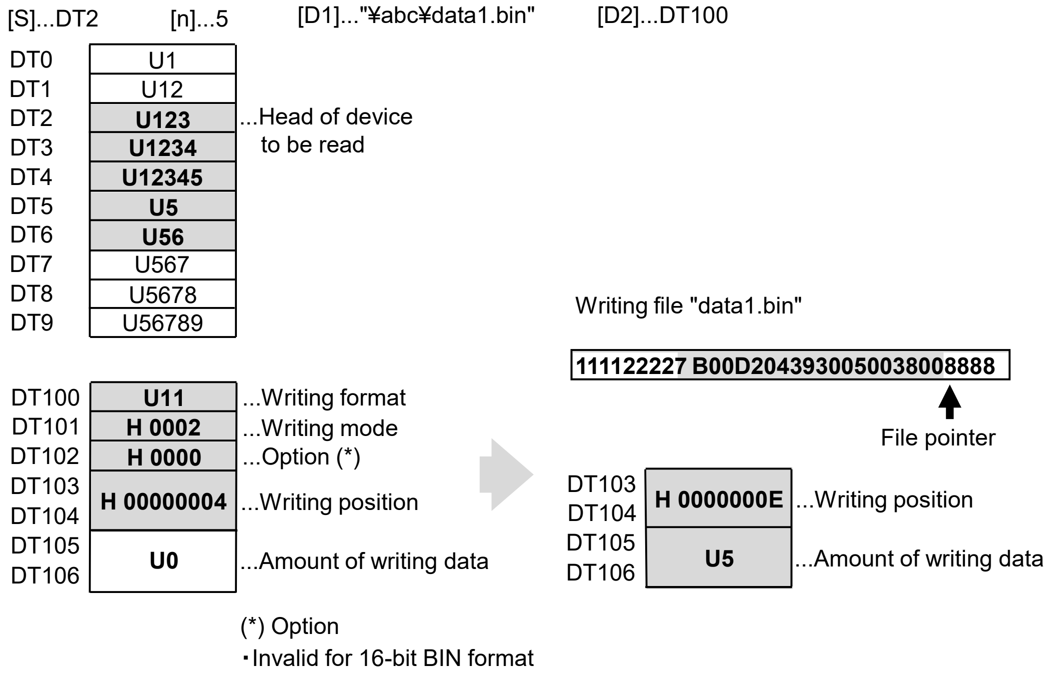
Example 1)

- Five pieces (five words) of 16-bit data are read from the area starting with device DT2, and the data pieces are written in the BIN format into the file "\abc\data1.bin" in the SD memory card in new file mode.



Example 2)

- Five pieces (five words) of 16-bit data are read from the area starting with device DT2, and the data pieces are written in the BIN format from the file pointer position of the existing file "\abc\data1.bin" in an SD memory card.



■ Precautions for programming

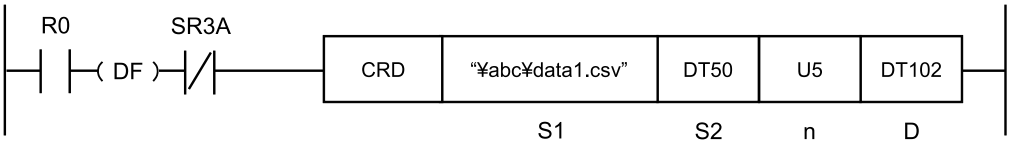
- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.
- In the case of the saving format 10 (ASCII string), character strings written from D2 are enclosed in double quotation marks and output.
- A double quotation mark (") in character strings is converted to two double quotation marks ("").
- When the set file attribute is read only, any data cannot be written.

■ Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	Set when a value outside the range is specified for the parameter.
	To be set when an out-of-range value is specified in the reserved area for system.

18.4 CRD (File Data Read Instruction)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Path name (folder name + file name: 1 to 256 characters) of the file from which data is read and the number of characters or the starting address of the stored device (data format: character data)
S2	Starting address of the device that stores the parameters related to the data to be read (data format: unsigned 16-bit integer)
n	Number of read data (data format: unsigned 16-bit integer)
D	Starting address of the device where the data to be read is stored (data format: unsigned 16-bit integer)

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	●
S2	●	●	●	●			●	●													●
n	●	●	●	●			●	●								●	●				●
D	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices can be modified. (Integer constants and character constants cannot be specified.)

■ Outline of operation

- This instruction reads [n] pieces of data from the file in an SD memory card specified by [S1] according to the parameter specified by [S2], and stores it in the device starting with [D].
- A comma and a line break code (LF or CR+LF) are recognized as delimiters between data pieces.
- If you specify ASCII data and read an odd number of bytes, only the lowest byte of the last word is stored.
- If you specify binary data and read an odd number of bytes, H00 is stored in the highest byte of the last word.

■ [S1], [S1+1]: Specification of folder name and file name

Setting device	Description
S1	Specifies the number of characters in the filename of the read file. (Specify the full path.)
S1+1 to S1+128	Specify the file to be read. ● Specify the full path, up to 256 characters including the folder name and filename

(Note 1) In the tool software FPWIN GR7, a path name (a folder name and a file name) can be entered directly as a character constant.

(Note 2) To specify a memory area such as a data register DT, use the SSET instruction to store a path name (a folder name and a file name) as character data.

■ [n]: Specification of number of data items to be read

Saving format	Set value of [S2]	Setting range of n
16-bit data	U1, U2, U7, U11	0 to 65535
32-bit data	U3, U4, U5, U8	0 to 32767
64-bit data	U6, U9	0 to 16383
ASCII	U10, U13	0 to 1999

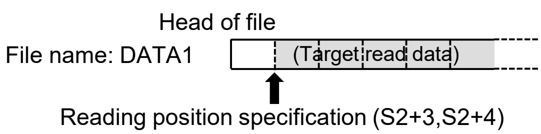
■ [S2] to [S2+6]: Specification of the data format for the data to be read

<table>
<tr><th>Operand</th><th>Description</th><th colspan="4">Description</th></tr>
<tr><td rowspan="15">S2</td><td rowspan="15">Reading format</td><td>Set value of S2</td><td colspan="2">Read contents</td><td>Extension</td></tr>
<tr><td>U0</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>U1</td><td rowspan="4">DEC</td><td>Unsigned 16-bit integer</td><td rowspan="10">.CSV
(comma-separated text)</td></tr>
<tr><td>U2</td><td>Signed 16-bit integer</td></tr>
<tr><td>U3</td><td>Unsigned 32-bit integer</td></tr>
<tr><td>U4</td><td>Signed 32-bit integer</td></tr>
<tr><td>U5</td><td rowspan="2">Floating point real number</td><td>32bit</td></tr>
<tr><td>U6</td><td>64bit</td></tr>
<tr><td>U7</td><td rowspan="3">HEX</td><td>1 words</td></tr>
<tr><td>U8</td><td>2 words</td></tr>
<tr><td>U9</td><td>4 words</td></tr>
<tr><td>U10</td><td>ASCII</td><td>Character string (Use NULL as a delimiter.)</td></tr>
<tr><td>U11</td><td>BIN</td><td>16bit</td><td>.BIN</td></tr>
<tr><td>U12</td><td colspan="2">Reserved for system</td><td>Reserved for system</td></tr>
<tr><td>U13</td><td>ASCII</td><td>Character string (Do not use NULL as a delimiter.)</td><td>.CSV
(comma-separated text)</td></tr>
</table>

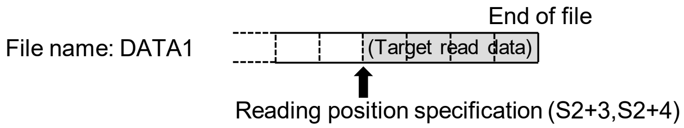
<table>
<tr><th>Operand</th><th>Description</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">S2+1</td><td rowspan="4">Reading mode</td><td>0: Normal mode:</td><td>Always reads data from the beginning of the file.</td></tr>
<tr><td>1: Normal mode:</td><td>Always reads data from the beginning of the file.
*The same operation as mode 0.</td></tr>
<tr><td>2: Reading position specification mode 1:</td><td>Reads data from the position after the number of bytes stored in [S2+3] and [S2+4] from the head of the file.</td></tr>
<tr><td>3: Reading position specification mode 2:</td><td>Reads data from the position after the number of bytes stored in [S2+3] and [S2+4] from the end of the file.</td></tr>
<tr><td>S2+2</td><td>Reserved for system</td><td colspan="2">(Zero is set.)</td></tr>
<tr><td>S2+3</td><td rowspan="2">Read position
(File pointer)</td><td colspan="2" rowspan="2">● Available only when the reading position specification mode 1 or mode 2 is set in [S2+1].
● The reading position (file pointer) is specified in bytes.
● When the file reading process is completed, the reading position (file pointer) moves to the end of read data. If the reading operation is performed again in this state, the next data will be read.
● In the case of reading position specification mode 1, data in the file is read from the reading position (file pointer) counted from the start of the file.
● In the case of the reading position specification mode 2, data in the file is read from the reading position (file pointer) counted from the end of the file.
● When the reading mode is set to the normal mode, the file reading position is disabled. Therefore, data is always read from the head of a file. In this case, the storage to the reading position (file pointer) is not performed after the reading process.</td></tr>
<tr><td>S2+4</td></tr>
<tr><td>S2+5</td><td rowspan="2">Number of read data</td><td colspan="2" rowspan="2">Stores the number of data items that could be read as a result of reading from the file.
Example 1) When the number of data to be read is 40 and 100 pieces of data exist in the file, and the data is read starting from the head of the file, "40" (the number of read data) is stored.
Example 2) When the number of data to be read is 40 and 30 pieces of data exist in the file, and the data is read starting from the head of the file, "30" (the number of read data) is stored.
Example 3) When the number of data to be read is 40 and 100 character data exist in the file, and the data is read starting from the head of the file, "40" (the number of read characters) is stored.
Example 4) When the number of data to be read is 40 and 30 character data exist in the file, and the data is read starting from the head of the file, "30" (the number of read characters) is stored.</td></tr>
<tr><td>S2+6</td></tr>
</table>

■ [S2+1]: Specification of reading mode

Example 1) When set to reading position specification mode 1



Example 2) When set to reading position specification mode 2



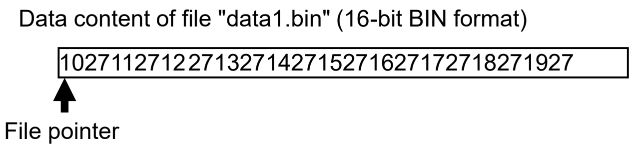
■ Example of processing

Example 1)

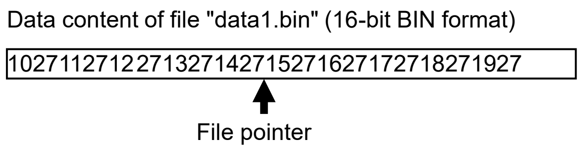
- Five pieces (five words) of 16-bit BIN data are read from the file "\abc\data1.bin" in an SD memory card, and the data pieces are stored in the area starting with DT102.
- As the "Reading position specification mode 1" is selected, the file pointer moves after reading.

[S1]..."¥abc¥data1.bin" [S2]...DT50 [n]...5 [D]...DT102

DT50	**H 000B**	...Reading format
DT51	**H 0002**	...Reading mode
DT52	**H 0000**	...Reserved for system
DT53 DT54	**H 00000000**	...Reading position
DT55 DT56	**U0**	...No. of read data



DT100	H 0000	DT100	H 0000
DT101	H 0000	DT101	H 0000
DT102	H 0000	DT102	**H 1027**
DT103	H 0000	DT103	**H 1127**
DT104	H 0000	DT104	**H 1227**
DT105	H 0000	DT105	**H 1327**
DT106	H 0000	DT106	**H 1427**
DT107	H 0000	DT107	H 0000
DT108	H 0000	DT108	H 0000
DT109	H 0000	DT109	H 0000



Example 2)

- Five pieces of data are read from the file "\abc\data1.csv" in an SD memory card, and the data pieces are stored in the area starting with DT102 as five pieces (five words) of 16-bit unsigned integer data.
- As the “Reading position specification mode 1” is selected, the file pointer moves after reading.

[S1]..."¥abc¥data1.csv" [S2]...DT50 [n]...5 [D]...DT102

DT50	**U1**	...Reading format
DT51	**H 0002**	...Reading mode
DT52	**H 0000**	...Reserved for system
DT53 DT54	**H 00000000**	...Reading position
DT55 DT56	**U0**	...No. of read data

Data content of file "data1.csv" (16-bit DEC format)

12,123,1234,12345,5,56,567,5678,56789,1,

File pointer

	Before	After
DT100	U0	U0
DT101	U0	U0
DT102	U0	**U12**
DT103	U0	**U123**
DT104	U0	**U1234**
DT105	U0	**U12345**
DT106	U0	**U5**
DT107	U0	U0
DT108	U0	U0
DT109	U0	U0

Data content of file "data1.csv" (16-bit DEC format)

12,123,1234,12345,5,56,567,5678,56789,1,

File pointer

Example 3)

- Five pieces of data are read from the file "\abc\data1.csv" in an SD memory card, and the data pieces are stored in the area starting with DT102 as five pieces (ten words) of 32-bit unsigned integer data.
- As the “Reading position specification mode 1” is selected, the file pointer moves after reading.

[S1]..."¥abc¥data1.csv" [S2]...DT50 [n]...5 [D]...DT102

DT50	U3	...Reading format
DT51	H 0002	...Reading mode
DT52	H 0000	...Reserved for system
DT53 DT54	H 00000000	...Reading position
DT55 DT56	U0	... No. of read data

Data content of file "data1.csv" (32-bit DEC format)

Byte address

123456,1234567,12345678,123456789,567890,5678901,..

File pointer

DT100	U0
DT102	U0
DT104	U0
DT106	U0
DT108	U0
DT110	U0
DT112	U0
DT114	U0
DT116	U0
DT118	U0

DT100	U0
DT102	**U123456**
DT104	**U1234567**
DT106	**U12345678**
DT108	**U123456789**
DT110	**U567890**
DT112	U0
DT114	U0
DT116	U0
DT118	U0

Data content of file "data1.csv" (32-bit DEC format)

123456,1234567,12345678,123456789,567890,5678901,..

File pointer

Example 4)

- Reads ten data (ten characters) of ASCII (000AH) data from the file "\abc\data1.csv" in an SD memory card. (When using NULL as a delimiter)
- When there are two consecutive double quotation marks (""), each one (") is read as one character.
- The data is stored in the area starting with DT102 as character data.
- As the “Reading position specification mode 1” is selected, the file pointer moves after reading.

[S1]..."¥abc¥data1.csv" [S2]...DT50 [n]...10 [D]...DT102

DT50	**U10**	...Reading format
DT51	**H 0002**	...Reading mode
DT52	**H 0000**	...Reserved for system
DT53 DT54	**H 00000000**	...Reading position
DT55 DT56	**U0**	...No. of read data

Data content of file "data1.csv" (ASCII string format)

"ABC""DE""FGH","IJ","KLMNOP",

File pointer

DT100	H 0000	
DT101	H 0000	
DT102	H 00	H 00
DT103	H 00	H 00
DT104	H 00	H 00
DT105	H 00	H 00
DT106	H 00	H 00
DT107	H 0000	
DT108	H 0000	

DT100	H 0000	
DT101	H 0000	
DT102	**"B"**	**"A"**
DT103	**""""**	**"C"**
DT104	**"E"**	**"D"**
DT105	**"F"**	**""""**
DT106	**"H"**	**"G"**
DT107	H 0000	
DT108	H 0000	

Data content of file "data1.csv" (ASCII string format)

"ABC""DE""FGH","IJ","KLMNOP",

File pointer

Example 5)

- Reads four data (four characters) of ASCII (000AH) data from the file "\abc\data1.csv" in an SD memory card. (When using NULL as a delimiter)
- The data is stored in the area starting with DT102 as character data.
- As the “Reading position specification mode 1” is selected, the file pointer moves after reading.

[S1]..."¥abc¥data1.csv" [S2]...DT50 [n]...4 [D]...DT102

DT50	**H 000A**	...Reading format
DT51	**H 0002**	...Reading mode
DT52	**H 0000**	...Reserved for system
DT53 DT54	**H 00000000**	...Reading position
DT55 DT56	**U0**	...No. of read data

Data content of file "data1.csv" (ASCII string format)

"ABC“,"DEF“,"GHI","JKL",

File pointer

DT100	H 0000	
DT101	H 0000	
DT102	H 0000	
DT103	H 00	H 00
DT104	H 00	H 00
DT105	H 00	H 00
DT106	H 00	H 00
DT107	H 00	H 00
DT108	H 0000	
DT109	H 0000	

DT100	H 0000	
DT101	H 0000	
DT102	**H 0004**	
DT103	**“B”**	**“A”**
DT104	**“D”**	**“C”**
DT105	H 00	H 00
DT106	H 00	H 00
DT107	H 00	H 00
DT108	H 0000	
DT109	H 0000	

Data content of file "data1.csv" (ASCII string format)

"ABC“,"DEF“,"GHI","JKL",

File pointer

Example 6)

- Reads three data (three characters) of ASCII (000DH) data from the file "\abc\data1.csv" in an SD memory card. (When not using NULL as a delimiter)
- The data is stored in the area starting with DT102 as character data.
- As the “Reading position specification mode 1” is selected, the file pointer moves after reading.

[S1]..."¥abc¥data1.csv" [S2]...DT50 [n]...3 [D]...DT102

DT50	**H 000D**	...Reading format
DT51	**H 0002**	...Reading mode
DT52	**H 0000**	...Reserved for system
DT53 DT54	**H 00000000**	...Reading position
DT55 DT56	**U0**	...No. of read data

Data content of file "data1.csv" (ASCII string format)

"A[NULL]B“,"DEF“,"GHI","JKL",

File pointer

DT100	H 0000	
DT101	H 0000	
DT102	H 0000	
DT103	H 00	H 00
DT104	H 00	H 00
DT105	H 00	H 00
DT106	H 00	H 00
DT107	H 00	H 00
DT108	H 0000	
DT109	H 0000	

DT100	H 0000	
DT101	H 0000	
DT102	**H 0003**	
DT103	[NULL]	**“A”**
DT104	H 00	**“B”**
DT105	H 00	H 00
DT106	H 00	H 00
DT107	H 00	H 00
DT108	H 0000	
DT109	H 0000	

Data content of file "data1.csv" (ASCII string format)

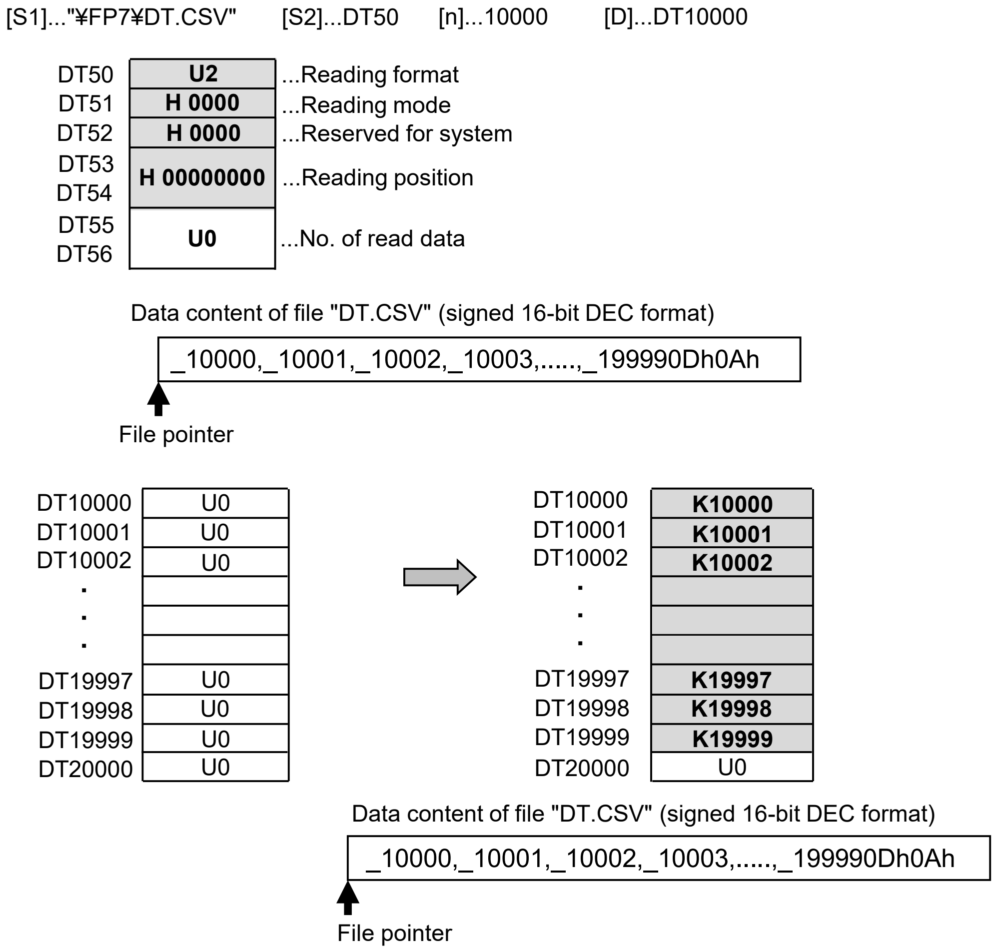
"A[NULL]B“,"DEF“,"GHI","JKL",

File pointer

Example 7)

- 10000 pieces of data are read from the file "\FP7\DT.CSV" in an SD memory card, and the data pieces are stored in the area starting with DT10000 as 10000 pieces (10000 words) of 16-bit signed integer data.

  Because "normal mode 0" is selected, data is always read from the head of a file.

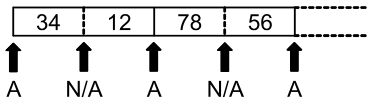


■ Precautions for programming

- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.
- When there are two consecutive double quotation marks ("") in character strings, each one (") is read as one character. Double quotes alone are ignored.
- If a space, comma or line break is inserted at the position of file pointer after reading a CSV file, the file pointer output to the result data is at the data position next to the comma or line break. A space, comma or line break at the end of data is skipped.
- When reading a CSV file, null fields (such as parts with successive commas) are skipped, and the data is not stored in devices. The next data to be read is stored in the next device (that is not a skipped null field). In this case, the skipped data is counted as part of the data count.
- During the execution of the CRD instruction, data values read from an SD memory card are written from the beginning of a specified device area in order. Until the completion of the CRD instruction, do not read the data of the device specified by the CRD instruction.

- When reading as ASCII data, data containing delimiters (commas or line break codes) may not be processed correctly.
- Specify the points at which each data is separated for [S2+3] and [S2+4], reading positions (file pointers). "A" in the figure below shows the positions where data can be read properly.

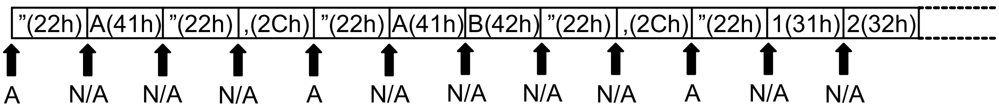
Example 1: 16-bit integer data (bin. format file)



Example 2: ASCII data (csv. format file)

A: Position where data can be properly read

N/A: Position where data cannot be properly read

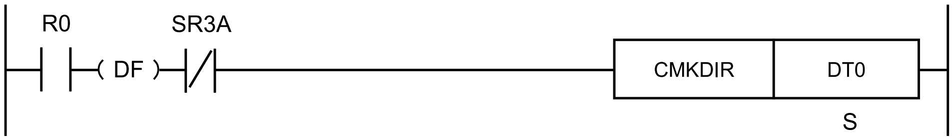


■ Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	Set when a value outside the range is specified for the parameter.
	To be set when an out-of-range value is specified in the reserved area for system.

18.5 CMKDIR (Directory Creation)

■ Ladder diagram



■ List of operands

Operand	Description
S	Starting address of the device area that stores the path name (folder name) of the folder to be created and the number of characters of the path name (data format: character data) Specify the number of characters for [S], and path name (folder name) for [S+1] and successive operand.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●												●	●

(Note 1) Only 16-bit devices can be modified. (Character constants cannot be specified.)

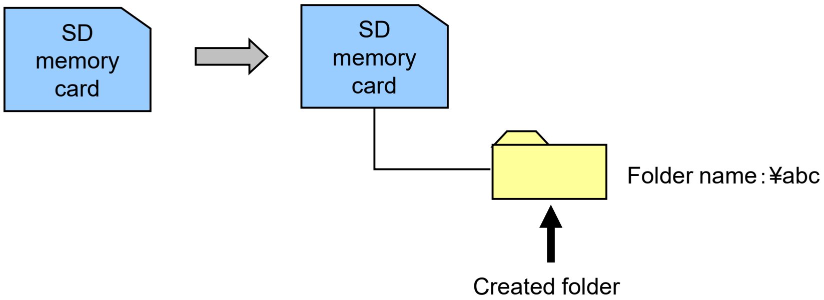
■ Outline of operation

- Creates a folder within an SD memory card.
- Specify by storing the number of characters in the folder name in [S], and the characters indicating the folder name as ASCII data in [S+1] onwards.

■ Example of processing

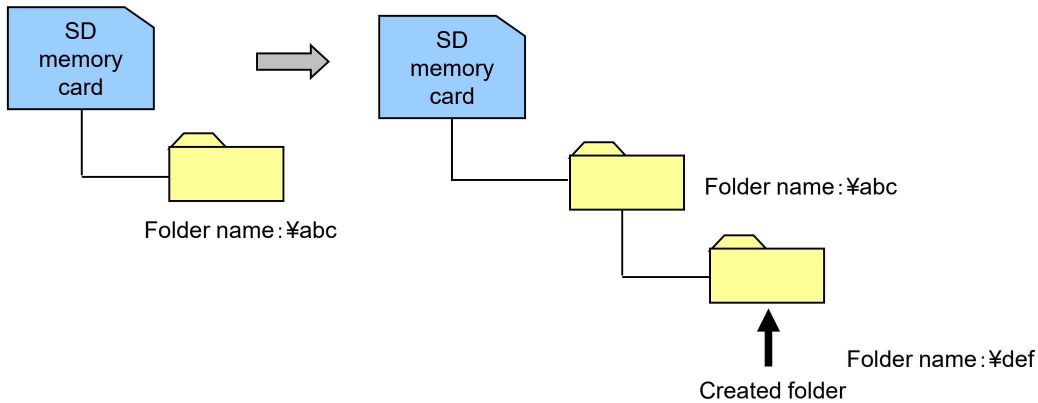
Example 1) When creating a folder "\abc" in an SD memory card

DT0	U4		・・・ No. of characters of folder name
DT1	"a"	"¥"	・・・ Folder name
DT2	"c"	"b"	
DT3	H 0000		
DT4	H 0000		
DT5	H 0000		
Byte address	High	Low	



Example 2) When creating a folder "\abc\def" in an SD memory card

DT0	U8		・・・No. of characters of folder name
DT1	"a"	"¥"	・・・Folder name
DT2	"c"	"b"	
DT3	"d"	"¥"	
DT4	"f"	"e"	
DT5	H 0000		
Byte address	High	Low	



■ Precautions for programming

- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".

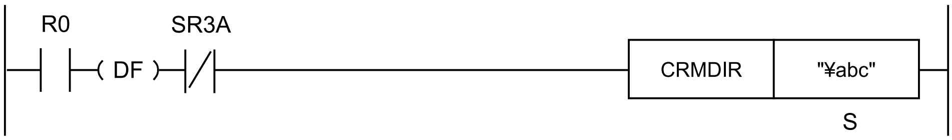
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.
- To create a folder in a lower hierarchy like the folder "\abc\def", create the folder of the upper hierarchy in advance. Folders cannot be created at the same time.
- If a non-existent folder is specified to a higher level, an error will result.
- If the folder to be created already exists, no operation is performed and it ends normally.

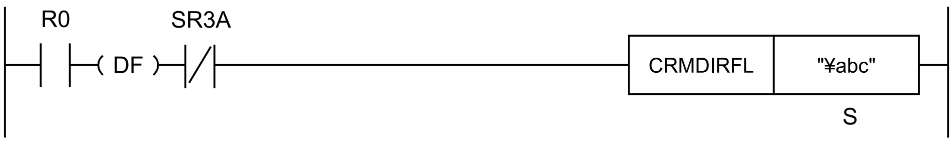
■ Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).

18.6 CRMDIR/CRMDIRFL (Directory Deletion)

■ Ladder diagram





■ List of operands

Operand	Description
S	Starting address of the device area that stores the path name (folder name) of the folder to be deleted and the number of characters of the path name (data format: character data) Specify the number of characters for [S], and the path name (folder name) for [S+1] and subsequent operands.

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:		Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●												●	●

(Note 1) Only 16-bit devices can be modified. (Character constants cannot be specified.)

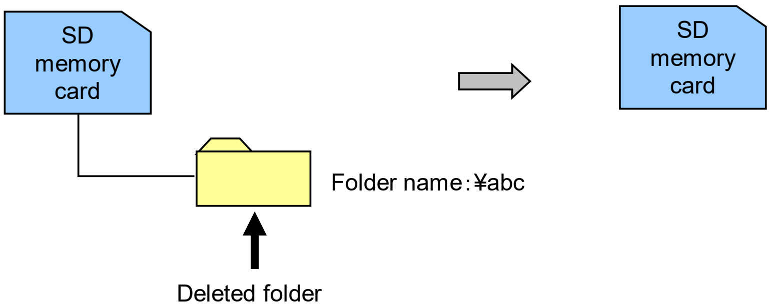
■ Outline of operation

- The directory in the SD memory card specified by [S] is deleted.
- To specify the folder name, store the number of characters in [S], and store the characters that indicate the folder name as ASCII data in [S+1] and subsequent operands.

■ Example of processing

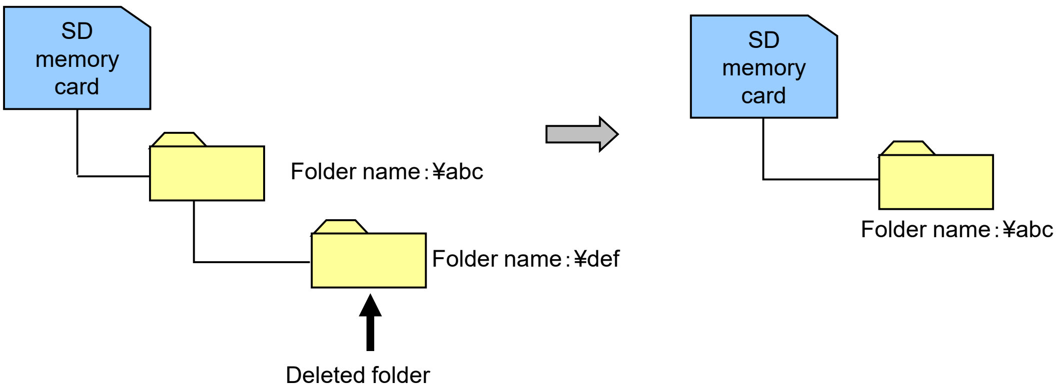
Example 1) When deleting a folder "\abc" in an SD memory card

Byte address	High	Low	
DT0	U4		･･･No. of characters of folder name
DT1	"a"	"¥"	･･･Folder name
DT2	"c"	"b"	
DT3	H 0000		
DT4	H 0000		
DT5	H 0000		



Example 2) When deleting a folder "\abc\def" in an SD memory card

Byte address	High	Low	
DT0	U8		･･･No. of characters of folder name
DT1	"a"	"¥"	･･･ Folder name
DT2	"c"	"b"	
DT3	"d"	"¥"	
DT4	"f"	"e"	
DT5	H 0000		



■ Precautions for programming

- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".

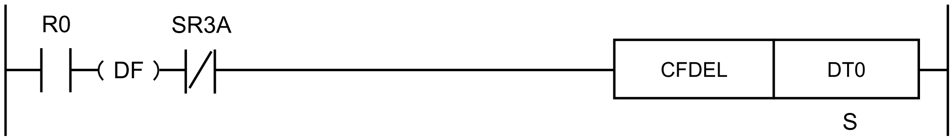
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.
- When the folder to be deleted does not exist an error occurs.
- In the case of the CRMDIR instruction, an error occurs when there are files or folders in the specified folder. Check inside the folder.
- In the case of the CRMDIRFL instruction, the specified folder can be deleted even when there are files in the folder. If there is a folder inside the folder an error occurs.
- The CRMDIRFL instruction can be used for CPU units CPS4R*/CPS3R*.

■ Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).

18.7 CFDEL (File Delete)

■ Ladder diagram



■ List of operands

Operand	Description
S	Specify the starting address of the device area that stores the path name (folder name + file name: maximum 256 characters) of the file to be deleted, the number of characters of the path name (data format: character data), the number of characters for [S], and the path name (folder name + file name) for [S+1] and successive operand

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●												●	●

(Note 1) Only 16-bit devices can be modified. (Character constants cannot be specified.)

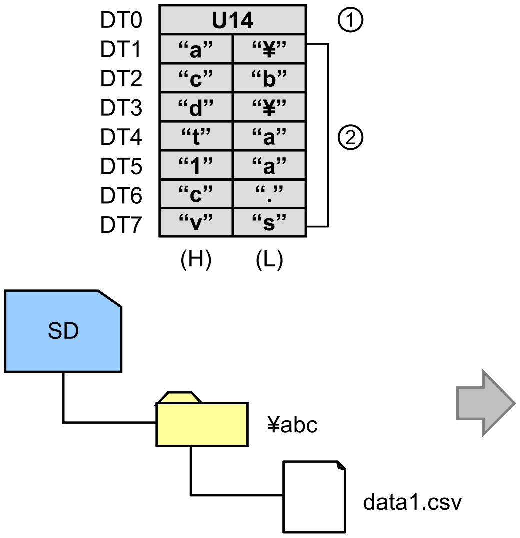
■ Outline of operation

- Files in the SD memory card specified by [S] are deleted.

■ Example of processing

Example) When deleting a file "\abc\data1" in an SD memory card

(1)	Character count	(2)	Path name (folder name + file name)



■ Precautions for programming

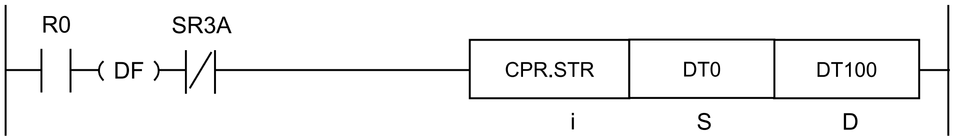
- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.
- If the file to be deleted does not exist, an error will result.

■ Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).

18.8 CPR (ASCII Data Write to File)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF	STR
i								●

■ List of operands

Operand	Description
S	Character string data to be written or the starting address of the device area where the character string is stored
D	Starting address of the device area that stores the path name (folder name + file name: maximum 256 characters) of the file in which string data is written and the number of characters of the path name (data format: character data)

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●												●	●
D	●	●	●	●			●	●												●	●

(Note 1) Only 16-bit devices can be modified. (Character constants cannot be specified.)

■ Outline of operation

- Adds the character string specified by [S] to the end of the file named with the character string specified by [D].
- When the file specified by [D] does not exist, create a new file.

■ [S] to [S+1]: Specification of character string data

- Parameters related to the character string data written into a file name in a SD memory card

Setting device	Description
S	Number of character strings to be written

Setting device	Description
	Up to 4096 characters
S+1	Write character string data

■ [D] to [D+1]: Specification of folder name and file name

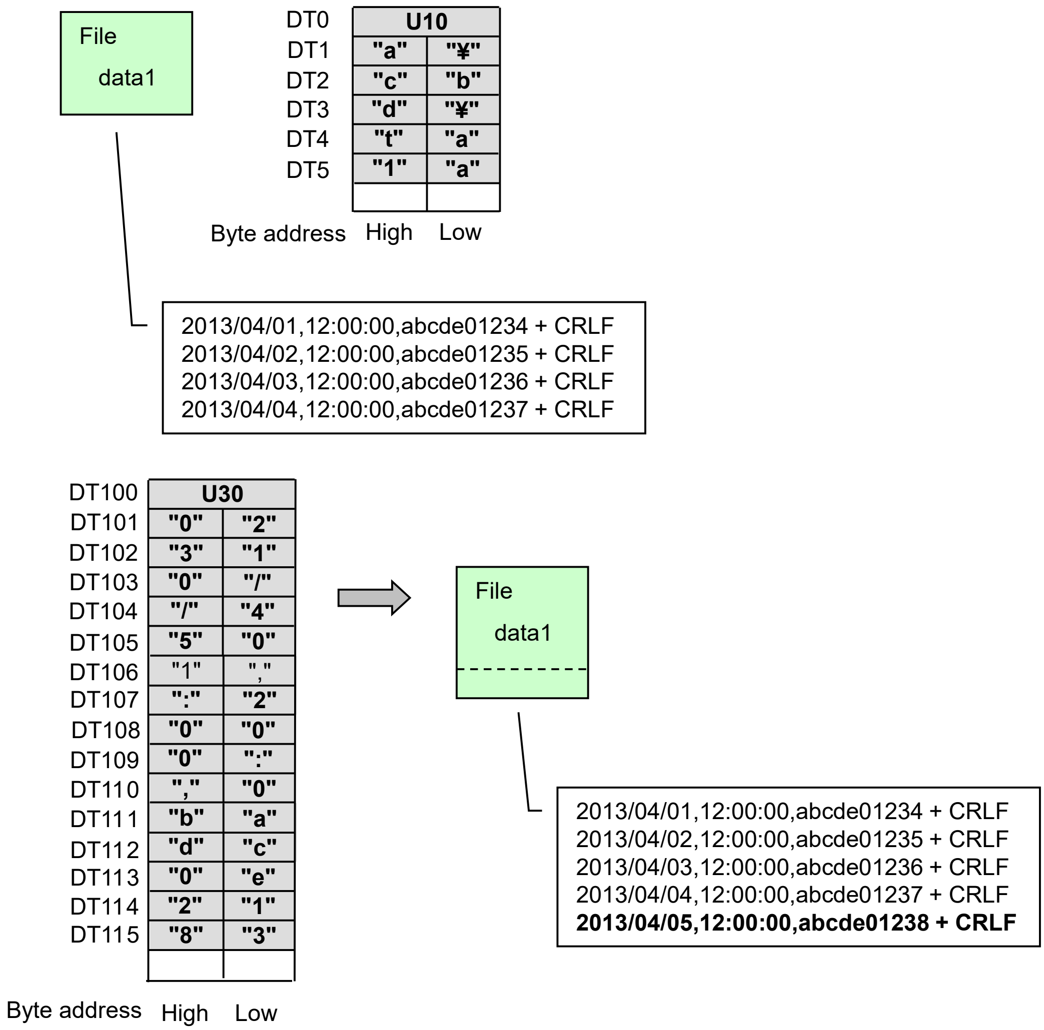
- Starting device that stores the file name (folder name + file name: 1 to 256 characters) of the file to be written into an SD memory card and the number of characters of the file name

Setting device	Description
D	Specify the number of characters of the file name to be written. (Specify the full path.)
D+1	Specify the file to be written. ● The full path configuration is up to a maximum of 256 characters including the folder name + filename

■ Example of processing

When writing a character string "2013/04/05,12:00:00,abcde01238" in a file "\abc\data1":

[i] …STR [D] …DT0 [S] …DT100



■ Precautions for programming

- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.

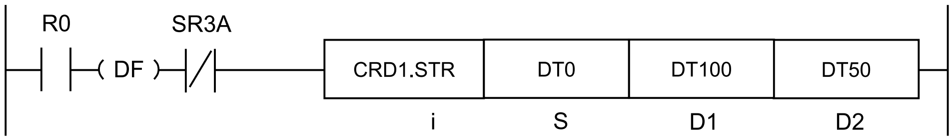
■ Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.

Name	Description
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when the character string specified by [S] exceeds 4096 characters.

18.9 CRD1 (One Line Read from File)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF	STR
i								●

■ List of operands

Operand	Description
S	Starting address of the device area that stores the path name (folder name + file name: maximum 256 characters) of the file from which data is read and the number of characters of the path name (data format: character data)
D1	Starting address of the device storing the character string data that was read
D2	Starting address of the device storing the parameter data relating to the position and upper limit of the read data

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●												●	●
D1	●	●	●	●			●	●													●
D2	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices can be modified. (Character constants cannot be specified.)

■ Outline of operation

- This instruction reads the file specified by [S] from the reading position specified by [D2] up to a specified maximum number of bytes until the detection of CR+LF, and stores it in the device address starting with [D1].

■ [S] to [S+1]: Specification of folder name and filename

- Starting device that stores the folder name (folder name + file name: 1 to 256 characters) of the folder stored in an SD memory card and the number of characters of the folder name

Setting device	Description
S	Specifies the number of characters in the filename of the read file. (Specify the full path.)
S+1	Specify the file to be read. ● The full path configuration is up to a maximum of 256 characters including the folder name + filename

■ [D2] to [D2+3]: Specification of read position and upper limit

Setting device	Settings	Details
D2	Read position (File pointer)	Specifies the byte position from the start of the file. Each line feed character (CR(0DH) or LF(0AH) is counted as one character. After the instruction is executed, [D2, D2+1] is updated with a read pointer value that has had the number of bytes that were read added to it. If the reading operation is performed again in this state, the next data will be read. The reading position can be specified by eight bits (by one byte).
D2+1		
D2+2	Maximum number of bytes to be read (Setting range: 1 to 4,096)	Sets the maximum number of bytes in the read data. The setting range becomes 1 to 4,096. When 0 is set, operation is performed as 4096. When a line feed character (CR(0DH), LF(0HA) or CR+LF) exists before reaching the specified maximum number of bytes, the reading operation ends at that point.
D2+3	Reserved for system	Set to 0.

■ Example of processing

Example 1) When reading data from a file "\data1.txt" specifying the head of the file for the reading position

[i] …STR
[S1] …DT0　[D1] …DT100　[D2] …DT50

Byte address	High	Low
DT0	U10	
DT1	"d"	"¥"
DT2	"t"	"a"
DT3	"1"	"a"
DT4	"t"	"."
DT5	"t"	"x"
DT6	H 0000	
DT7	H 0000	

DT100	H 0000
DT101	H 0000
DT102	H 0000
DT103	H 0000
DT104	H 0000
DT105	H 0000
DT106	H 0000
DT107	H 0000
DT108	H 0000
DT109	H 0000

DT50 DT51	H 00000000	...From the head of file
DT52	U100	... Max. size: 100 bytes
DT53	H 0000	...Reserved for system
DT54	H 0000	

Contents of fie "data1.txt"

12345678 + CRLF
a,ab,abc,abcd,abcde + CRLF

: Position of file pointer

Byte address	High	Low
DT100	U8	
DT101	"2"	"1"
DT102	"4"	"3"
DT103	"6"	"5"
DT104	"8"	"7"
DT105	H 0000	
DT106	H 0000	
DT107	H 0000	
DT108	H 0000	
DT109	H 0000	

DT50 DT51	H 0000000A	...Read data Reading position at the end
DT52	U100	
DT53	H 0000	
DT54	H 0000	

Contents of fie "data1.txt"

12345678 + CRLF
a,ab,abc,abcd,abcde + CRLF

Example 2) When reading data from a file "\data1.txt" specifying the 4th byte from the head of the file for the reading position

[i] …STR

[S1] …DT0 [D1] …DT100 [D2] …DT50

DT0	U10	
DT1	"d"	" ¥ "
DT2	"t"	"a"
DT3	"1"	"a"
DT4	"t"	"."
DT5	"t"	"x"
DT6	H 0000	
DT7	H 0000	
Byte address	High	Loe

DT100	H 0000
DT101	H 0000
DT102	H 0000
DT103	H 0000
DT104	H 0000
DT105	H 0000
DT106	H 0000
DT107	H 0000
DT108	H 0000
DT109	H 0000

DT50	H 00000004	...Position of 4th byte from the head of file
DT51		
DT52	U100	...Max. size: 100 bytes
DT53	H 0000	...Reserved for system
DT54	H 0000	

Contents of file "data1.txt"

12345678 + CRLF

a,ab,abc,abcd,abcde + CRLF

:Position of file pointer

DT100	U4	
DT101	"6"	"5"
DT102	"8"	"7"
DT103	H 0000	
DT104	H 0000	
DT105	H 0000	
DT106	H 0000	
DT107	H 0000	
DT108	H 0000	
DT109	H 0000	
Byte address	High	Low

DT50	H 0000000A	...Read data Reading position at the end
DT51		
DT52	U100	
DT53	H 0000	
DT54	H 0000	

Contents of file "data1.txt"

12345678 + CRLF

a,ab,abc,abcd,abcde + CRLF

Example 3) When reading data from a file "\abc\data1.txt" specifying the 10th byte from the head of the file for the reading position

[i] …STR

[S1] …DT0 [D1] …DT100 [D2] …DT50

	High	Low
DT0	U14	
DT1	"a"	"¥"
DT2	"c"	"b"
DT3	"d"	"¥"
DT4	"t"	"a"
DT5	"1"	"a"
DT6	"t"	"."
DT7	"t"	"x"
DT8	H 0000	

Byte address High Low

DT100	H 0000
DT101	H 0000
DT102	H 0000
DT103	H 0000
DT104	H 0000
DT105	H 0000
DT106	H 0000
DT107	H 0000
DT108	H 0000
DT109	H 0000

DT50, DT51	H 0000000A	··· Position of 9th byte from the head of file
DT52	U10	··· Max. size: 10 bytes
DT53	H 0000	··· Reserved for system
DT54	H 0000	

Contents of file "data1.txt"

12345678 + CRLF
a,ab,abc,abcd,abcde + CRLF

: Position of file pointer

	High	Low
DT100	U10	
DT101	","	"a"
DT102	"b"	"a"
DT103	"a"	","
DT104	"c"	"b"
DT105	"a"	","
DT106	H 0000	
DT107	H 0000	
DT108	H 0000	
DT109	H 0000	

Byte address High Low

DT50, DT51	H 00000014	··· Read data Reading position at the end
DT52	U10	
DT53	H 0000	
DT54	H 0000	

Contents of file "data1.txt"

12345678 + CRLF
a,ab,abc,abcd,abcde + CRLF

■ Precautions for programming

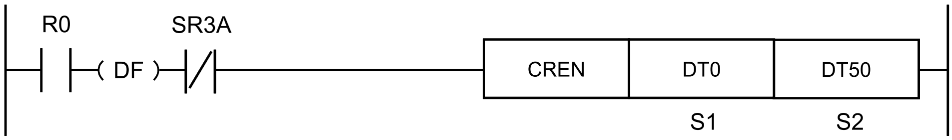
- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.
- During the execution of the CRD1 instruction, data read from the SD memory card is written from the beginning of the specified data device area in order. Therefore, until this instruction is complete, do not read data within the range of the data device specified to store the data.

Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).
	To be set when the value of [D2+2] exceeds 4096 characters.
	To be set when [D1]+[D2+2] is outside the device range.

18.10 CREN (File Rename)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting address of the device area that stores the path name (folder name + file name: maximum 256 characters) of the file to be renamed and the number of characters of the path name (data format: character data) Specify the number of characters for [S1], and path name (folder name + file name) for [S1+1] and successive operand.
S2	Starting address of the device area that stores the path name (folder name + file name: maximum 256 characters) of the file after being renamed and the number of characters of the path name (data format: character data) Specify the number of characters for [S2], and path name (folder name + file name) for [S2+1] and successive operand. The folder name can be omitted

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	•	•	•	•			•	•												•	•
S2	•	•	•	•			•	•												•	•

(Note 1) Only 16-bit devices can be modified. (Character constants cannot be specified.)

■ Outline of operation

- The file name specified by [S1] is changed to the file name specified by [S2].

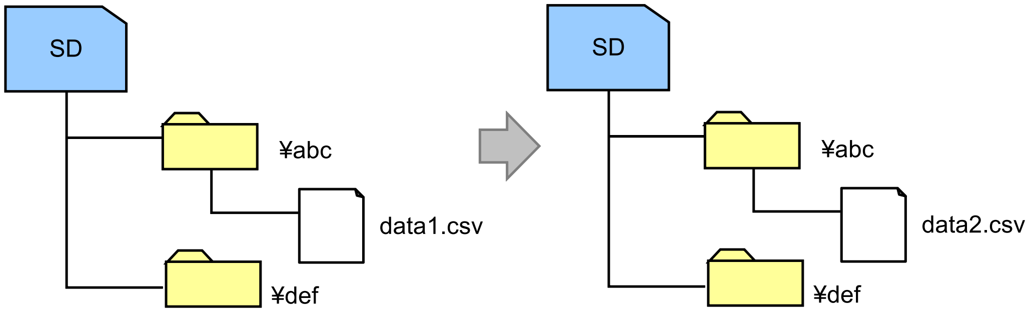
■ Example of processing

Example) When renaming a file "\abc\data1" to "\abc\data2"

(1)	Character count	(2)	Path name (folder name + file name)

	(H)	(L)	
DT0	U14		①
DT1	“a”	“¥”	
DT2	“c”	“b”	
DT3	“d”	“¥”	
DT4	“t”	“a”	②
DT5	“1”	“a”	
DT6	“c”	“.”	
DT7	“v”	“s”	

	(H)	(L)	
DT50	U9		①
DT51	“a”	“d”	
DT52	“a”	“t”	
DT53	“.”	“2”	②
DT54	“s”	“c”	
DT55	**	“v”	
DT56	**	**	



■ Precautions for programming

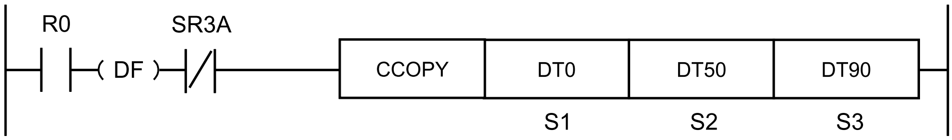
- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.
- The CREN instruction cannot be executed when the SD memory card access instruction is being executed.
- When the full path is specified for the file name and folder name after renaming, abnormality is detected and operation abends if it does not match the folder name before renaming (all path names).

■ Flag operations

Name	Description
SD memory card access instruction active (SR3A)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SD memory card access instruction execution done (SR3B)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SD memory card access instruction execution result (SR3C)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).

18.11 CCOPY (File Copy)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting address of the device area that stores the path name (folder name + file name: maximum 256 characters) of the file to be copied and the number of characters of the path name (data format: character data) Specify the number of characters for [S1], and path name (folder name + file name) for [S1+1] and successive operand.
S2	Starting address of the device area that stores the path name (folder name + file name: maximum 256 characters) of the copy destination file and the number of characters of the path name (data format: character data) Specify the number of characters for [S2], and path name (folder name + file name) for [S2+2] and successive operand.
S3	Setting of copy format

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	•	•	•	•			•	•												•	•
S2	•	•	•	•			•	•												•	•
S3	•	•	•	•			•	•							•	•	•				•

(Note 1) Only 16-bit devices can be modified. (Integer constants and character constants cannot be specified.)

■ Outline of operation

- Copies the file specified by [S1] to the file specified by [S2] in accordance with the parameters specified by [S3].
- If the folder is specified in [S1], all of the files directly under the folder in [S1] are copied directly under the folder specified by [S2].
- Does not copy folders in lower levels further than the [S1] folder.
- Files with the read-only attribute are not overwritten.
- If [S1] and [S2] are exactly the same, an error is returned regardless of the value of [S3].
- When specifying a file for [S1] and a folder for [S2], copies the file specified by [S1] into the folder specified by [S2].

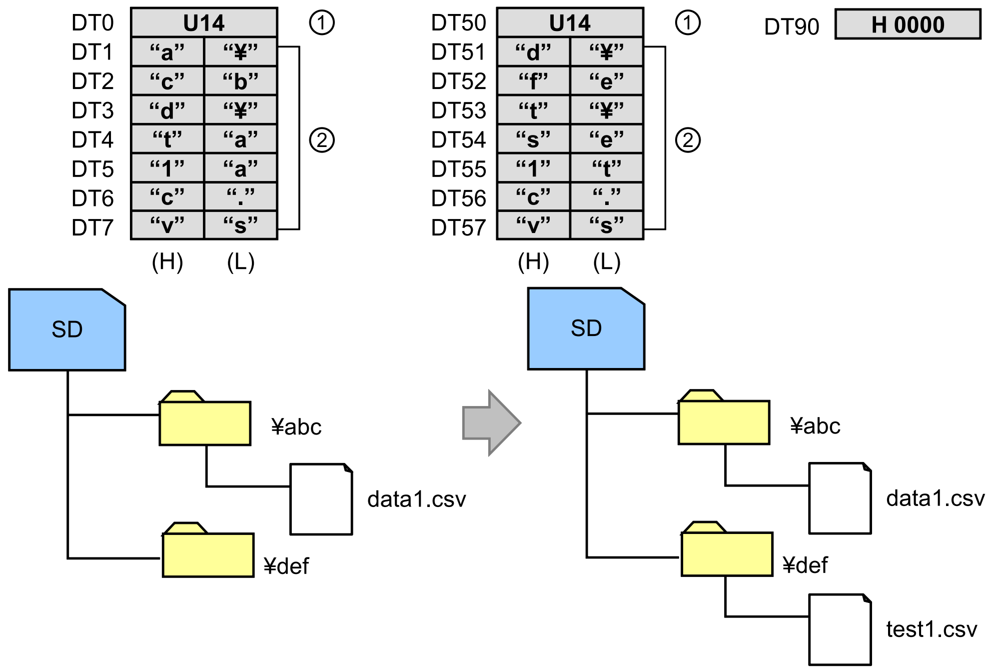
■ [S3]: Specification of copy format

Operand	Specified bits	Description
S3	bit 0	0: Overwrites if there is a file with the same name in the copy destination. 1: Terminates with an error without overwriting if there is a file with the same name in the copy destination.
	Bit1 to 15	(Reserved for system)

■ Example of processing

Example) When copying a file "abc\data1" to "\def\test1"

(1)	Character count	(2)	Path name (folder name + file name)



■ Precautions for programming

- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.
- When a folder is specified for [S1] and a file for [S2], "file name error" occurs.

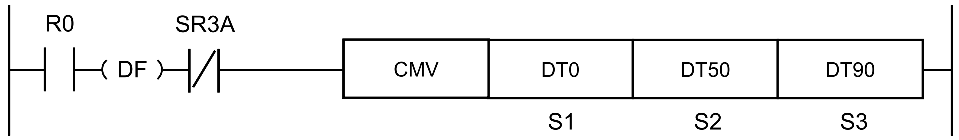
■ Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.

Name	Description
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).

18.12 CMV (File Move)

■ Ladder diagram



■ List of operands

Operand	Description
S1	Starting address of the device area that stores the path name (folder name + file name: maximum 256 characters) of the file to be moved and the number of characters of the path name (data format: character data) Specify the number of characters for [S1], and path name (folder name + file name) for [S1+1] and successive operand.
S2	Starting address of the device area that stores the path name (folder name + file name: maximum 256 characters) of the move destination file and the number of characters of the path name (data format: character data) Specify the number of characters for [S2], and path name (folder name + file name) for [S2+2] and successive operand.
S3	Setting of transfer format

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●												●	●
S2	●	●	●	●			●	●												●	●
S3	●	●	●	●			●	●							●	●	●				●

(Note 1) Only 16-bit devices can be modified. (Integer constants and character constants cannot be specified.)

■ Outline of operation

- This instruction moves the file specified by the character string data starting with [S1] as the file specified by the character string data starting with [S2].
- When a folder is specified in [S1], all files in the [S1] folder are transferred to the folder specified in [S2].
- Any folders within the folder specified in [S1] are not transferred.
- Read-only files remain read-only after transfer.
- A free space larger than the file size is necessary in a card for internal operation.
- When a file is specified in [S1] and a folder is specified in [S2], the file specified in [S1] will be transferred to the folder specified in [S2].

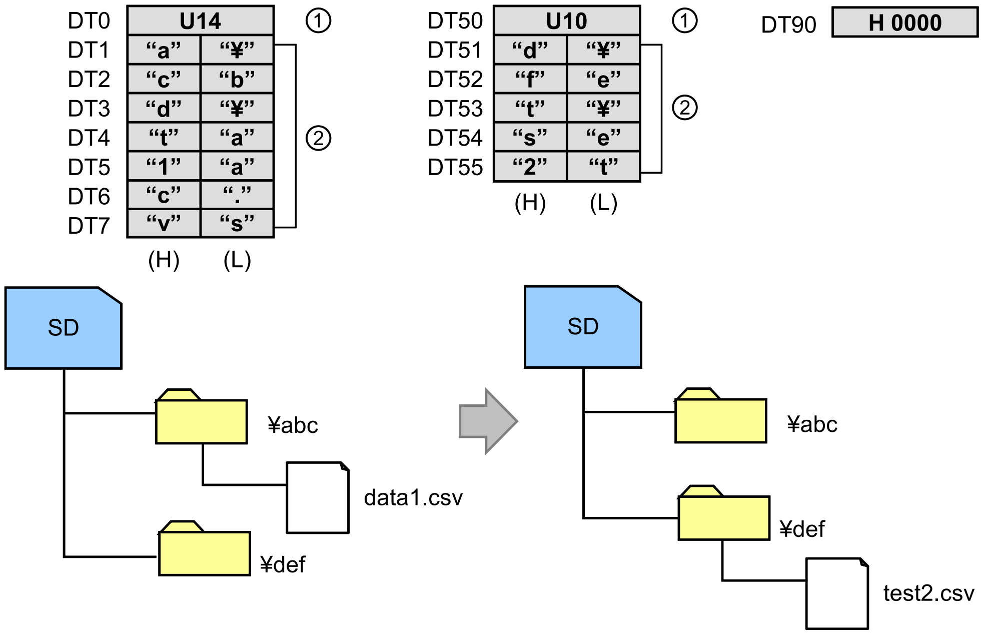
■ [S3]: Specification of movement type

Operand	Specified bits	Description
S3	bit 0	0: Overwrites if there is a file with the same name in the copy destination. 1: Terminates with an error without overwriting if there is a file with the same name in the copy destination.
	Bit1 to 15	(Reserved for system)

■ Example of processing

Example) When moving a file "abc\data1" to "\def\test2"

(1)	Character count	(2)	Path name (folder name + file name)



■ Precautions for programming

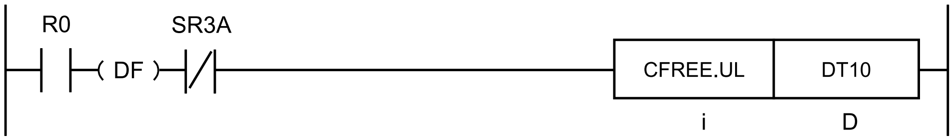
- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.
- When the folder specified by [S2] does not exist, "No file/folder" error occurs.
- When there is not enough free space, "memory card capacity shortage" error occurs.
- When a folder is specified for [S1] and a file for [S2], "file name error" occurs.

■ Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).

18.13 CFREE (Acquiring SD Memory Card Free Space)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i				●			

■ List of operands

Operand	Description
D	Starting address of the device storing the acquired free space in byte units

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●													●

■ Outline of operation

- The SD memory card free space is stored in byte units in the area specified by [D].

e.g. When the SD memory card has 4 GB of free space

[i]…UL [D] …DT10

DT10	**H 0000**	H 1 0000 0000 (4,294,967,296 byte)
DT11	**H 0000**	
DT12	**H 0001**	
DT13	**H 0000**	
DT14	H 0000	
DT15	H 0000	

■ Precautions for programming

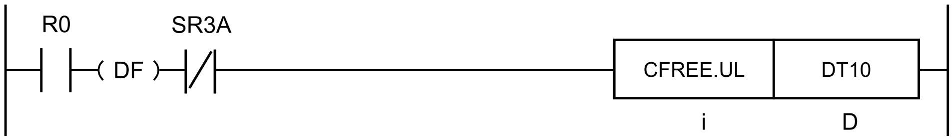
- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.

■ Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).

18.14 CFREEK (Acquiring SD Memory Card Free Space)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i				●			

■ List of operands

Operand	Description
D	Starting address of the device storing the acquired free space in kilobyte units

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:		Integer				Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D	●	●	●	●			●	●													●

■ Outline of operation

- This instruction stores the amount of free space on an SD memory card in the area specified by [D] in k(kilo) bytes.

e.g. When the SD memory card has 4 GB of free space

[i]…UL   [D] …DT10

DT10 DT11	**H 00400000**
DT12	**H 0000**

■ Precautions for programming

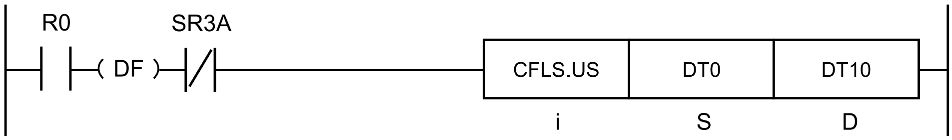
- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.

■ Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).

18.15 CFLS (Acquiring File Status)

■ Ladder diagram



■ Available operation units (●: Available)

Operation unit	bit	US	SS	UL	SL	SF	DF
i		●	●				

■ List of operands

Operand	Description
S	Starting address of the device area that stores the path name (folder name + file name: maximum 256 characters) of the file whose status is acquired and the number of characters of the path name (data format: character data)
D	Starting address of the device storing the acquired file status

■ Devices that can be specified (indicated by ●)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier (Note 1)
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●												●	●
D	●	●	●	●			●	●													●

(Note 1) Only 16-bit devices can be modified. (Character constants cannot be specified.)

■ Outline of operation

- The file name status specified by [S] is acquired and the result is stored in the 10-word area from [D] (D to D+9).

■ Details of stored content

Device storing acquired result	Acquired contents		
D	File attributes	Bit position	Description
		0	Turns ON in the case of a read-only file
		1	Turns ON in the case of a hidden file

Device storing acquired result	Acquired contents		
		2	Turns ON in the case of a system file
		3	Turns ON in the case of a volume label
		4	Turns ON in the case of a directory
		5	Turns ON in the case of an archive
		6 to 15	(Reserved: 0 (fixed))
D+1	(Reserved)		
D+2	File size: Stored as a decimal number.		
D+3			
D+4	Last modified: Stored as a decimal number.		Year (0 to 99)
D+5			Month (1 to 12)
D+6			Day (1 to 31)
D+7			Hour (0 to 23)
D+8			Minute (0 to 59)
D+9			Second (0 to 59)

■ Example of processing

Example) When acquiring the status of the file "\abc\data1"

- File attribute: Read-only
- File size: 123456 bytes
- Last modified: January 23, 2012 at 12:34:56

	(H)	(L)	
DT0	**U14**		①
DT1	**“a”**	**“¥”**	②
DT2	**“c”**	**“b”**	
DT3	**“d”**	**“¥”**	
DT4	**“t”**	**“a”**	
DT5	**“1”**	**“a”**	
DT6	**“c”**	**“.”**	
DT7	**“v”**	**“s”**	

		→		
DT10	**H 0000**		DT10	**H 0001**
DT11	**H 0000**		DT11	**H 0000**
DT12	**H 0000**		DT12	**U123456**
DT13	**H 0000**		DT13	
DT14	**H 0000**		DT14	**U12**
DT15	**H 0000**		DT15	**U1**
DT16	**H 0000**		DT16	**U23**
DT17	**H 0000**		DT17	**U12**
DT18	**H 0000**		DT18	**U34**
DT19	**H 0000**		DT19	**U56**
DT20	H 0000		DT20	H 0000

Precautions for programming

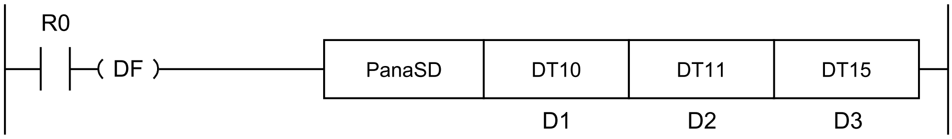
- Also refer to section "19.9 Common Precautions for SD Memory Card Access Instructions".
- The SD memory card access instruction active flag (SR3A) turns ON after the execution condition has turned ON until the completion of the execution of the instruction. During this time other SD memory card access instructions cannot be executed.

Flag operations

Name	Description
SR3A (SD memory card access instruction execution in progress)	Turns ON when instruction execution starts. Turns OFF when instruction execution is complete.
SR3B (SD memory card access instruction execution complete)	Turns OFF when instruction execution starts. Turns ON when instruction execution is complete.
SR3C (SD memory card access instruction execution result)	Indicates the execution results when instruction execution is complete. Normal end = 0, abnormal end = 1
SR7, SR8 (ER)	To be set in case of out-of-range values in indirect access (index modification).

18.16 PanaSD (Panasonic SD Memory Card Lifetime Information Read)

■ Ladder diagram



■ List of operands

Operand	Description
D1	Execution result code storage device address
D2	Starting address of device storing SD memory card lifetime information acquisition time
D3	Number of rewrites information storage device address

■ Devices that can be specified (indicated by •)

Operand	16-Bit device:											32-Bit device:			Integer			Real number		String	Index modifier
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D1	•	•	•	•			•	•												•	
D2	•	•	•	•			•	•													
D3	•	•	•	•			•	•													

■ Outline of operation

- This instruction is used to read Panasonic SD memory card lifetime information.
- This instruction and the SD card access instruction can be used at the same time.
- This instruction stores the execution result of this instruction in the area starting with [D1], [D2] and [D3].
- This instruction cannot be executed in duplicate.
- Do not use this instruction frequently. Executing this event by a differential instruction is recommended.
- This is a dedicated instruction for use with SD memory cards manufactured by Panasonic. It cannot be used with any other SD memory cards.
- The SD memory cards that this instruction supports are as follows.

■ Compatible SD memory cards

Type	Series	Conventional product
SLC	EX	FX
MLC	GD	JD, PC

(Note 1) The EX and GD series can be used for CPU units Ver.4.0 or later.

■ Execution result storage area

Operand	Items		Description
[D1]	Execution result code(Note 1)		Stores the execution result code. HFFFF: Executing H0: Normal end H1: Dual-boot error H2: SD memory card cover open error H3: SD memory card not installed error H4: Incompatible SD memory card error
[D2]	Acquisition time	Year and month	The SD memory card lifetime information acquisition time is stored as three-word BCD data. e.g. When the acquisition time is 15 hours 55 minutes and 30 seconds on October 3, 2014, it is stored as follows. [D2] H1410 [D2+1] H0315 [D2+2] H5530
[D2+1]		Day and hour	
[D2+2]		Minute and second	
[D3]	Number of rewrite information		The ratio (%) of the average number of rewrites of management blocks to the maximum number of possible rewrites is expressed by the following formula: [Number of rewrites (average of all management blocks)] / [Maximum number of possible rewrites] x 100

(Note 1) The most significant bit of the execution result code can be used as a flag during execution.

■ Example of processing

Example 1) When the execution result of the PanaSD instruction is Normal

[D1]...DT10 [D2]...DT11 to DT13 [D3]...DT15

When instruction is executed

DT10	H FFFF
DT11	
DT12	
DT13	
DT14	
DT15	

When SD card information is being read

DT10	H FFFF
DT11	
DT12	
DT13	
DT14	
DT15	

Reading SD card is complete (normal completion)

DT10	H 0	Execution result code	:0
DT11	H 1410	Year and month	:1410
DT12	H 0318	Day and hour	:318
DT13	H 5530	Minute and second	:5530
DT14			
DT15	H 0052	No. of rewrites information (%)	:82

Example 2) When the execution result of the PanaSD instruction is Error

[D1]...DT10 [D2]...DT11 to DT13 [D3]...DT15

When instruction is executed

DT10	H FFFF
DT11	
DT12	
DT13	
DT14	
DT15	

When SD card information is being read

DT10	H FFFF
DT11	
DT12	
DT13	
DT14	
DT15	

Reading SD card is complete (unsupported SD card error)

DT10	H 3	Execution result code	:3
DT11	H 0	Year and month	:0
DT12	H 0	Day and hour	:0
DT13	H 0	Minute and second	:0
DT14			
DT15	H 0	No. of rewrites information (%)	:0

■ Flag operations

Name	Description
SR7 SR8 (ER)	Set when the [D2] to [D2+2] range exceeds the accessible range.
	Set when executed in an interrupt program.

(MEMO)

19 Precautions for programming

19.1 Common precautions

The FP7 series differs from the existing FP series in the following points:

■ Operations of the carry flag and comparison flags (●: Change, -: Do not change)

Type of instruction	Mnemonic	Name of instruction	System relay number			
			Carry flag	Comparison flags		
			SR9(CY)	SRA (>)	SRB (=)	SRC(<)
Data comparison instructions	CMP	Data compare	-	●	●	●
	WIN	Band compare	-	●	●	●
	BCMP	Block comparison	-	-	●	-
Bit comparison instructions	BTT	16-bit data specified bit test	-	-	●	-
	STC	Carry-flag set	●	-	-	-
	CLC	Carry-flag reset	●	-	-	-
String operation instructions	ACHK	ASCII data check	-	-	●	-
	SCMP	String compare	-	●	●	●

- When operation is executed, the carry flag SR9 (CY) and comparison flags SRA (>), SRB (=), and SRC (<) are respectively activated by the instructions listed above.
- The flags do not change even if the operation result is "0" in the case of overflow or underflow with instructions other than listed above.

■ Operation at the time of overflow and underflow

- The carry flag SR9 (CY) does not change even in the case of overflow or underflow.
- Note that overflow or underflow may result if incorrect operation unit is specified.

■ Error flag

- Operation error flags are not cleared even in the case of normal operation.
- Use ERR instruction for clearing error flags.

■ Operation device areas

- When the operation unit is 32 bits (UL, SL, SF) or 64 bits (DF), if an area near the end of an operation device is specified, the device memory with a consecutive address within PLC is overwritten. Specify an operand so that areas are not incorrectly read or written.

Example) When a transfer instruction is used, and the operation unit is specified as 32 bits, if WY511 is selected for the transfer instruction [D], the starting area of WL0 is overwritten.

DT0	H 0011		WY509	H 1111
DT1	H 2233	↘	WY510	H 2222
DT2	H 4455	↘	WY511	H 2233
DT3	H 6677		WL0	H 4455
DT4	H 8899		WL1	H 5555

19.2 Clock and Time Data

■ Built-in calendar timer of the CPU unit

- The calendar timer should be adjusted in the **Set PLC Date and Time** menu of the tool software FPWIN GR7, or using the TIMEWT instruction.
- The set values are stored in the system data registers SD50 through 56, as listed below.

Device No.	Name of instruction	Data range	Remarks
SD50	Calendar timer (year)	00 to 99	The two lower digits of year in A.D. can be adjusted (up to 2099). Leap years can be set in this operation. Take leap years into account during setting. A year that can be divided by 4 is a leap year, and the days of February can be set to 29.
SD51	Calendar timer (month)	01 to 12	
SD52	Calendar timer (day)	01 to 31	
SD53	Calendar timer (hours)	0 to 23	
SD54	Calendar timer (minutes)	0 to 59	
SD55	Calendar timer (seconds)	0 to 59	
SD56	Calendar timer (day-of-the-week)	0 to 6	0: Sun., 1: Mon., 2: Tue., 3: Wed., 4: Thu., 5: Fri., 6: Sat.

■ Instructions that handle clock or time data

Mnemonic	Operand	Function
HMSS	S, D	(Time data) → (Seconds data)
SHMS	S, D	(Seconds data) → (Time data)
CADD	S1,S2,D	(Clock data) + (Time data) → (Clock data)
CSUB	S1,S2,D	(Clock data) - (Time data) → (Clock data)
TMSEC	S, D	(Clock data) → (Seconds data from the base time)
SECTM	S, D	(Seconds data from the base time) → (Clock data)
TIMEWT	S	(Clock data) → (SD50 to SD56)
SUMMER	S1,S2, S3,D	(Clock data) → (Clock data)

■ Format of clock data

One-word 16-bit binary data are allocated to each unit (year, month, day, hours, minutes and seconds), and the overall clock data are handled in the unit of 6 words.

Example) DT0 is specified for operand

	Word	Range:
DT0	Year	00 to 99
DT1	Month	01 to 12
DT2	Day	01 to 31

	Word	Range:
DT3	Hours	0 to 23
DT4	Minutes	0 to 59
DT5	Seconds	0 to 59

■ Format of time data

- One-word 16-bit binary data are allocated to each unit (hours, minutes and seconds), and the overall time data are handled in the unit of 3 words.
- Hours data can be specified in the range from 0 to 9999.

Example) DT0 is specified for operand

	Word	Range:
DT0	Hours	0 to 9999
DT1	Minutes	0 to 59
DT2	Seconds	0 to 59

19.3 Data Table for String Instructions

■ Data table structure

- Data handled by a string instruction store the number of characters in the initial word, and character data in the subsequent words.
- The maximum available size for string data is 4,096 bytes.

	Higher bytes	Lower bytes	
[S]	No. of characters		← No. of characters in a character string
[S+1]	Character 2 (Byte 1)	Character 1 (Byte 0)	Character data
[S+2]	Character 4 (Byte 3)	Character 3 (Byte 2)	
[S+3]	Character 6 (Byte 5)	Character 5 (Byte 4)	
[S+4]	Character 8 (Byte 7)	Character 7 (Byte 6)	
...	...		

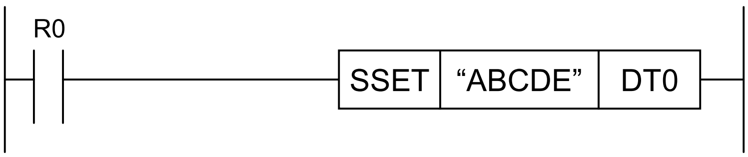
Example) A string data table is specified with the number of characters: 5, and character data: "ABCDE"

	Higher bytes	Lower bytes	
DT0	5 (no. of characters)		←No. of characters in a character string
DT1	H42 (B)	H41 (A)	Character data
DT2	H44 (D)	H43 (C)	
DT3		H45 (E)	
DT4			
...	...		

■ Conversion of string data using the SSET instruction

- Using the SSET instruction, a given string can be easily converted into a string data table.
- The maximum size of string data that can be specified for the SSET instruction is 256 bytes.

Example) Convert string data "ABCDE"
The number of characters is stored in DT0, and ASCII-converted character data are stored in DT1 and subsequent data registers.



19.4 Floating Point Real Number Operation

For floating point real number operations, the outcome of the operation with an infinitesimal or infinite operation result or with an infinitesimal or infinite input value is as follows.

■ Infinitesimal operation result

Instruction	SIN	COS	TAN	ASIN	ACOS	ATAN	ATAN2	SINH	COSH	TANH	EXP	LN	LOG	PWR	SQR
Input value	-	-	-π/2	●	●	-	●	-	-	-	-	●	●	●	●

(Note 1) Instructions indicated with "-" do not result in an infinitesimal value.

(Note 2) Instructions indicated with ● result in an operation error.

(Note 3) An operation error is output if the input value is out of the available range.

■ Infinite operation result

Instruction	SIN	COS	TAN	ASIN	ACOS	ATAN	ATAN2	SINH	COSH	TANH	EXP	LN	LOG	PWR	SQR
Input value	-	-	π/2	●	●	-	●	-	-	-	-	●	●	●	●

(Note 1) Instructions indicated with "-" do not result in an infinite value.

(Note 2) Instructions indicated with ● result in an operation error.

(Note 3) An operation error is output if the input value is out of the available range.

■ Infinitesimal (-∞) input value

Instruction	SIN	COS	TAN	ASIN	ACOS	ATAN	ATAN2	SINH	COSH	TANH	EXP	LN	LOG	PWR	SQR
Input value	nan	nan	nan	●	●	-π/2	●	-∞	+∞	-1.0	+0.0	●	●	●	●

(Note 1) Instructions indicated with "nan" result in an indefinite value.

(Note 2) Instructions indicated with ● result in an operation error.

(Note 3) An operation error is output if the input value is out of the available range.

■ Infinite (+∞) input value

Instruction	SIN	COS	TAN	ASIN	ACOS	ATAN	ATAN2	SINH	COSH	TANH	EXP	LN	LOG	PWR	SQR
Input value	nan	nan	nan	●	●	π/2	●	+∞	+∞	+1.0	+∞	●	●	●	●

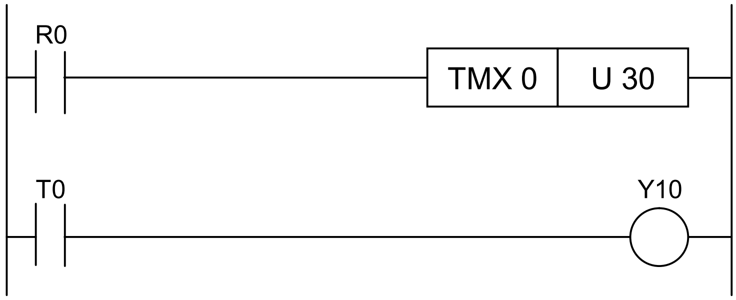
(Note 1) Instructions indicated with "nan" result in an indefinite value.

(Note 2) Instructions indicated with ● result in an operation error.

(Note 3) An operation error is output if the input value is out of the available range.

19.5 Changing Timer/Counter Set Value in the RUN Mode

19.5.1 Method to rewrite a constant in a program



■ Set value (constant) in a program

A constant in a program can be rewritten under the following conditions.

- Operation mode: RAM/ROM operation only
- Rewriting method: (1) Use the tool software

Here is an example of changing the set value of timer 0 from U30 to U50.

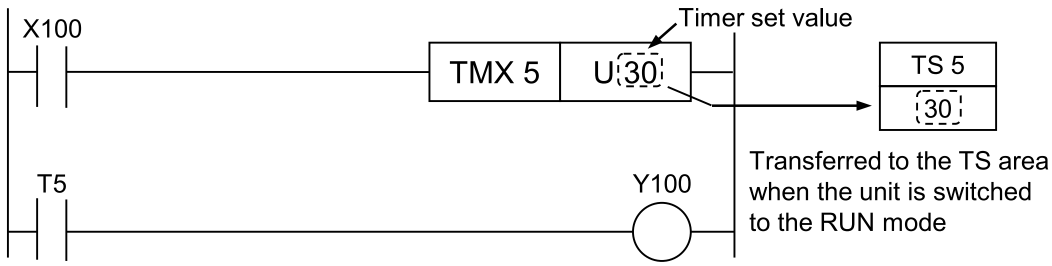
1 2 Procedure

1. Place the cursor over the set value for Timer 0 ("U30").
2. Enter a new constant "U50" and press the <return> key.
3. Press <Ctrl> + <F1> keys to execute **Convert PBs**.
4. Press [Yes (Y)] after a confirmation message is displayed.

Operation and cautions after the change

- Timer/counter in operation continues the pre-change operation. The program will start operating with the changed settings after the next execution condition changes from OFF to ON.
- When a constant in a program is rewritten, the program itself is overwritten. Therefore, when the unit is switched to another mode and then returned to RUN, or when it is powered off and then on again, the program is preset with the changed set value.

19.5.2 Method to rewrite a value in the set value area



■ Changing a value in the set value area TS/CS

The set value area TS/CS can be rewritten under the following conditions.

- Operation mode: RAM/ROM operation
- Rewriting method: (1) Use the tool software; (2) Use a programmed high-level instruction

■ Operation and cautions after the change

- Timer/counter in operation continues the pre-change operation. The program will start operating with the changed settings after the next execution condition changes from OFF to ON.
- The program itself is not overwritten even if a value in the set value area TS/CS is changed. Therefore, operations as described below follow when the unit is switched to another mode and then returned to RUN, or when it is powered off and then on again.
  1. When a set value is specified by a U constant

     The constant is preset in the set value area TS/CS. After the change, the value will no longer be valid.
  2. When the set value is specified by a set value area number

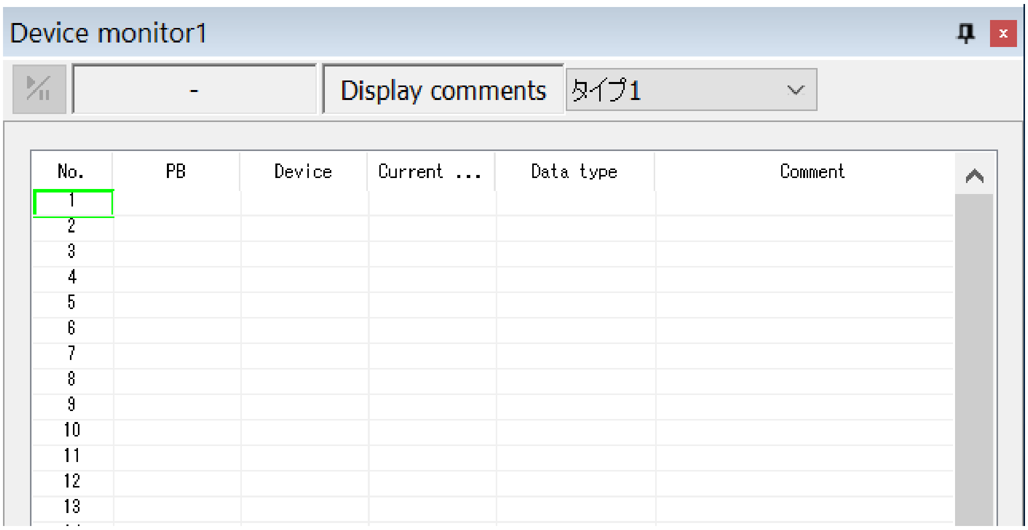
     In the case of the timer, "0" is preset in the set value area TS.

     In the case of the counter, a value changed in a method as described below is preset in the set value area CS.

19.5.3 Method 1: Use the tool software FPWIN GR7

1 2 Procedure

1. From the menu bar, select **Online**>**Device monitor**.
The "Device monitor" dialog box is displayed.



2. Double-click on the **Device** column.

The "Register monitor device" dialog box is displayed.

3. Under the device type, select "TS (timer set value)" or "CS (counter set value)" and enter a given device number.

The entered number is displayed in the "Device monitor" dialog box.

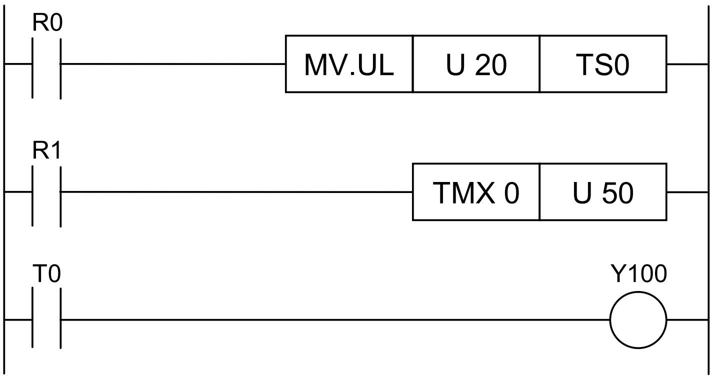
4. Enter a given value under current value and press the <return> key.

The set value area is updated.

19.5.4 Method 2: Using a program (high-level instruction)

- When a set value for the timer/counter is to be changed due to input conditions or for other reasons, rewrite the value of the set value area TS of the timer/counter to be changed, using a high-level instruction as described below.

Example: Changing the set value to U20 when input R0 turns ON
When R0 is ON, the timer set value is changed from 5 to 2 seconds.



- The set value can be changed by specifying the data register DT as the set value area, and modifying the value to be transferred using the MV instruction, etc. It is also possible to specify the number of the set value area (TS/CS) as an operand for the set value area.

19.6 Use of Duplicate Output

■ Duplicate Output

- Duplicate output refers to repeatedly specifying the same output in a single sequence program.
- If the same output is specified for the "OT" and "KP" instructions, it is considered to be duplicate outputs.
- Even if the same output is used for other multiple instructions, such as SET, RST, or high-level instructions (for example, data transfer), it is not regarded as duplicate outputs.
- Usually, an error occurs if the unit is switched to the RUN mode with duplicate outputs. The ERROR LED goes ON and the self-diagnosis flag SR0 turns ON.

■ How to check for duplicate output

- You can check for duplicate outputs in a program using the tool software FPWIN GR7 in the following method.
  1. From the menu bar, select **Debug**>**Project total check**.
     The "Project total check" dialog box is displayed.
  2. Press the "Execute (E)" button.
     If duplicate outputs are identified, the PB name, address, and error details (duplicate outputs definition error) are displayed.
  3. Select a given line and press the [Jump] button.
     The cursor moves to the instruction where duplicate outputs are involved.

■ Enabling duplicate output

- If you need to use output repeatedly due to the content of the program, duplicate output can be enabled.
- In such cases, switch **CPU configuration**>**Select operation**>**Duplicate output authorization** to "ON".
- In this case, an error will not result even if the program is executed.

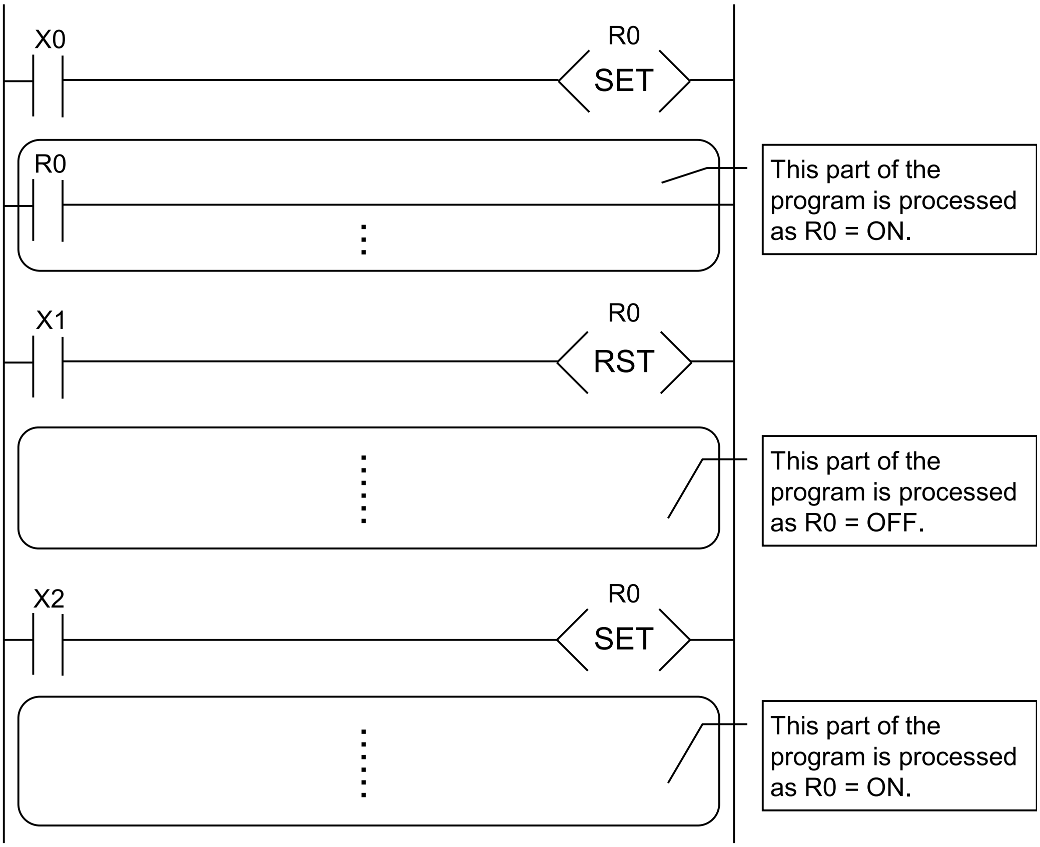
Note

- Even when a project total check is conducted using the tool software FPWIN GR7, the instruction used at the start is not indicated. Instead, the second and later outputs that are regarded as duplicate use are indicated.

■ Processing of duplicate uses of the same output

- If the same coil is specified in instructions output to internal relays or output relays (for example, the OT, KP, SET, RST and/or transfer instructions), contents thereof are rewritten at each step during operation.

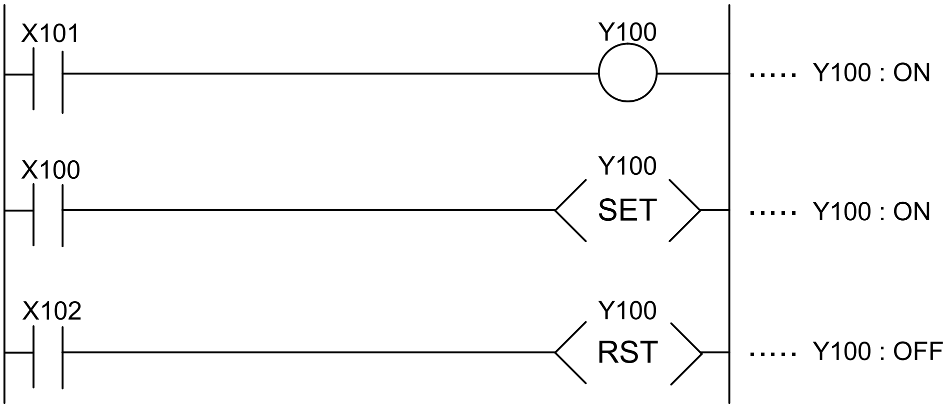
Example) Processing when the SET, RST and OT instructions are used (with X100 to X102 all ON)



■ Determination of operation result

- If the same output is used in duplicate by several instructions such as the OT instruction, KP instruction, SET instruction, RST instruction, or a transfer instruction, the output obtained when I/O refresh is performed is determined by the final operation results.

Example) The same output relay Y100 is used in the OT, SET and RST instructions



19.7 Rise Detection Method

■ Instructions using the leading edge detection operation

(1) DF (leading edge differential)

(2) Count input for CT (counter)

(3) Count input for UDC (up-down counter)

(4) Shift input for SR (shift register)

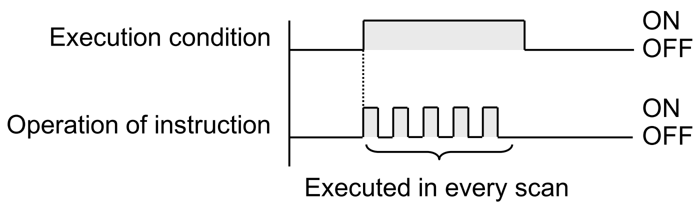
(5) Shift input for LRSR (left and right shift register)

(6) High-level instructions executed only at the leading edge (instructions specified with P and a number)

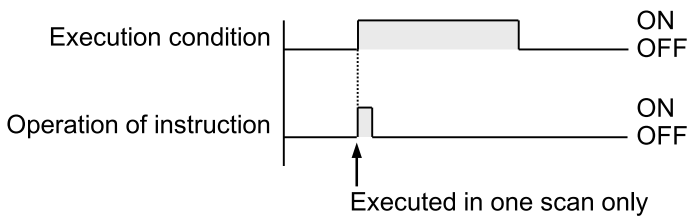
■ What is rise detection?

- Instructions for which rise detection is performed are only executed in the scan when the execution condition changes from OFF to ON.

(1) Normal input detection



(2) Rise detection



■ Rise detection method

- The previous execution condition is compared with the current execution condition, and the instruction is executed only when the previous condition was OFF and the current condition is ON.
- The instruction will not be executed otherwise.

■ Precautions when using an instruction which performs leading edge detection

- When RUN is started, such as when the power is turned on, instructions are not executed because the change of the execution condition from OFF to ON is not detected.
- Be aware that, if used with instructions that change the order of execution, such as the instructions in (1) to (6) below, the operation of instructions may change depending on the input timing.

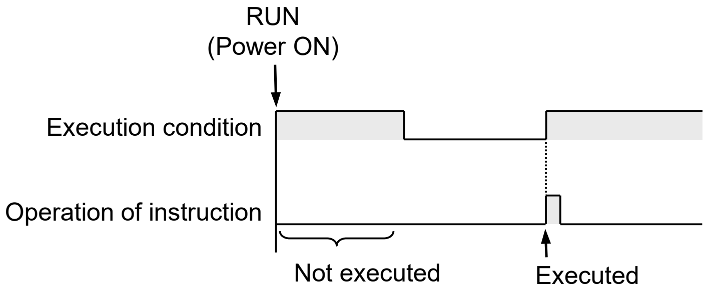
  [Be careful when leading edge detection type instructions are used in combination with:]

  (1) MC to MCE instructions

(2) JP to LBL instructions
(3) LOOP to LBL instructions
(4) CNDE instruction
(5) Step ladder instructions
(6) Subroutine instructions

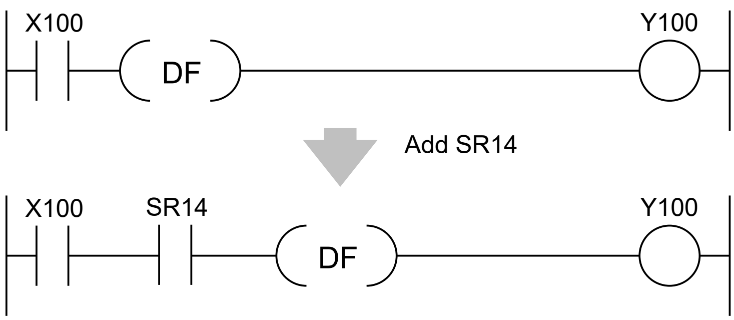
■ Operation and Precautions at Run Start Time

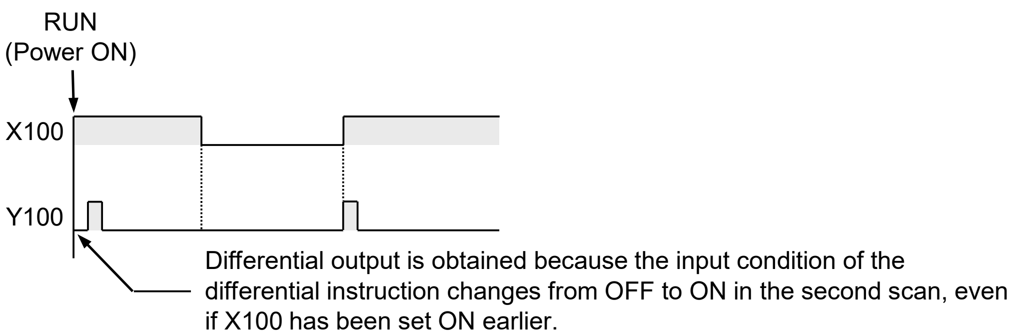
- The leading edge detection instruction is not executed when the mode has been switched to the RUN mode, or when the power supply is booted in the RUN mode, and if the trigger (execution condition) is already ON.



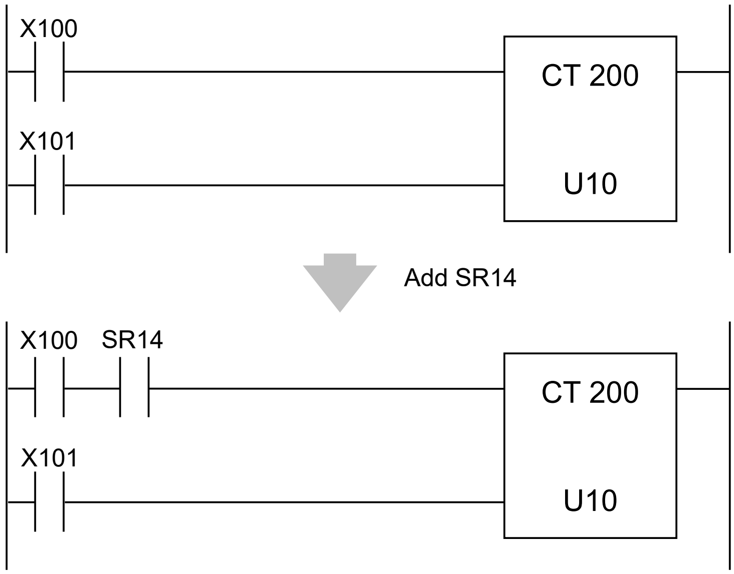
- If you need to execute an instruction when the trigger (execution condition) is ON prior to switching to the RUN mode, make a program as below using SR14 (special internal relay). (SR14 is a special internal relay which is OFF during the first scan and turns ON at the second scan onwards.)

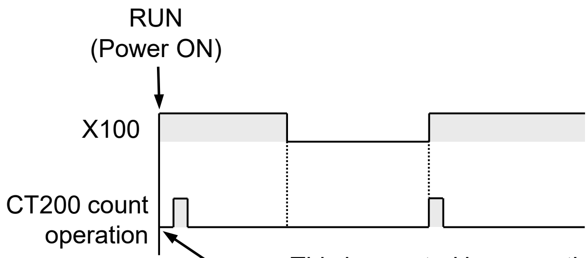
Example 1) DF (leading edge differential) instruction





Example 2) CT (counter) instruction



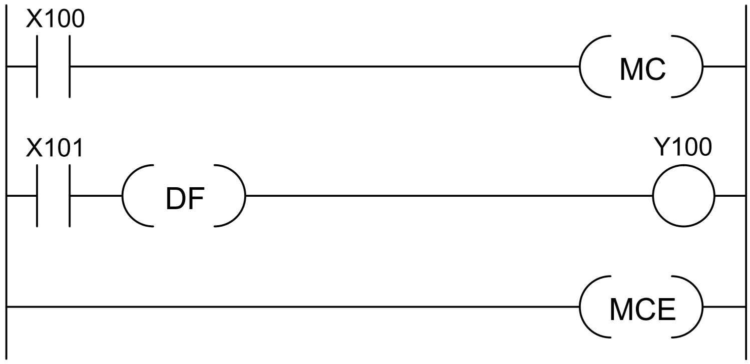


This is counted because the counter input condition changes from OFF to ON in the second scan, even if X100 has been set ON earlier.

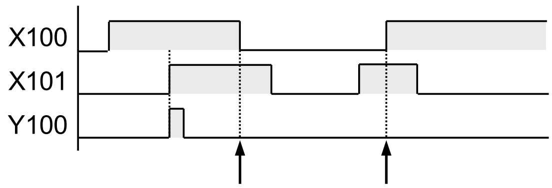
■ Precautions When Using Control Instructions

- The condition of the previous execution and the condition of the current execution are compared, and the leading edge detection instruction is executed only if the previous condition was OFF and the current condition is ON. They are not executed in any other circumstance.
- Therefore, take care that, when a leading edge detection instruction is used in combination with an instruction which changes the order of instruction execution, such as MC - MCE or JP - LBL, operations may change as follows depending on input timing.

Example 1) When using the differential instruction DF between MC and MCE



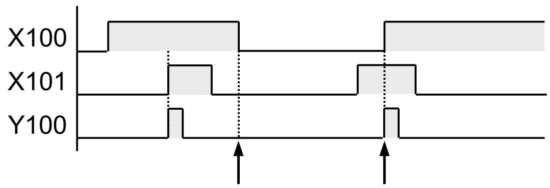
Timing chart 1



Previous execution of differential instruction

Differential output is not obtained because X101 as the input condition for the differential instruction has not been changed from OFF to ON after the previous execution.

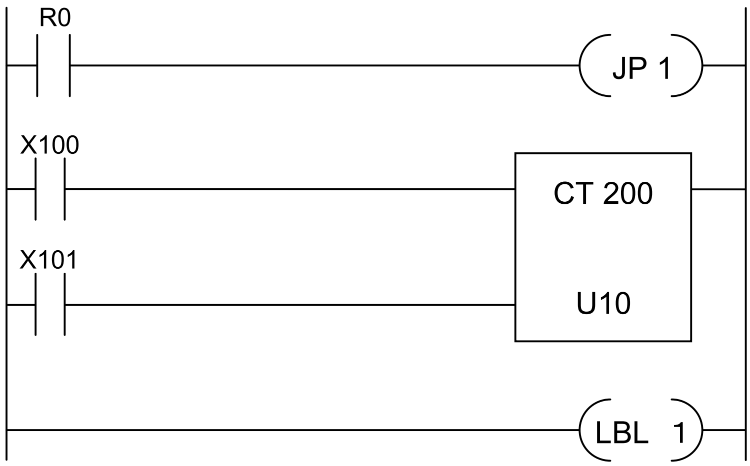
Timing chart 2



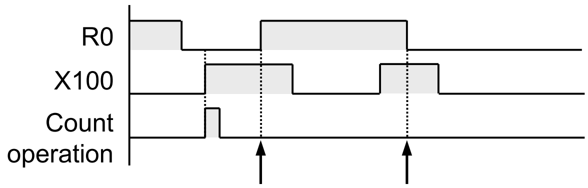
Previous execution of differential instruction

Differential output is obtained because X101 as the input condition for the differential instruction has been changed from OFF to ON after the previous execution.

Example 2: When using the counter instruction between JP and LBL



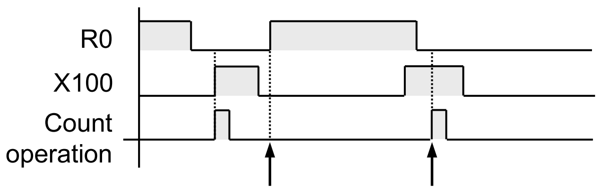
Timing chart 1



Final timing of the previous non-execution of JP instruction

This is not counted because X0 as the trigger (execution condition) for counter input has not been changed after the final timing of the previous non-execution of JP instruction.

Timing chart 2



Final timing of the previous non-execution of JP instruction

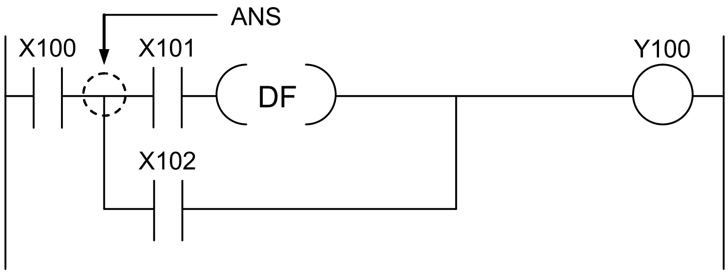
This is counted because counter input has been changed from OFF to ON after the final timing of the previous non-execution of JP instruction

19.8 Precautions for Programming

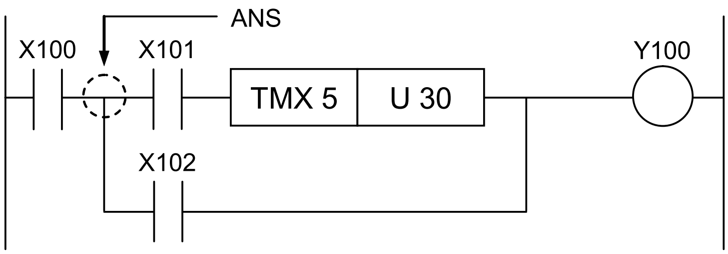
■ Programs that do not execute correctly

- Do not write the following programs as they will not be executed correctly.

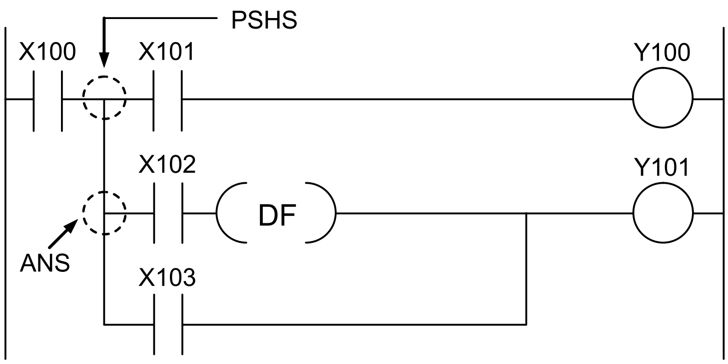
Example 1) If X101 is set ON earlier, Y100 does not go ON even when X100 is turned ON.



Example 2) Regardless of the ON or OFF state of X100, TMX5 is activated if X101 is turned ON.



Example 3) If X102 is set ON earlier, Y101 does not go ON even when X100 is turned ON.

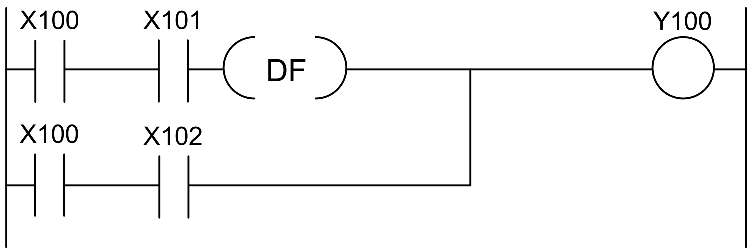


- When a combination of contacts are set as the trigger (execution condition) of a differential instruction or timer instruction, do not use the ANS instruction, RDS instruction, or POPS instruction.

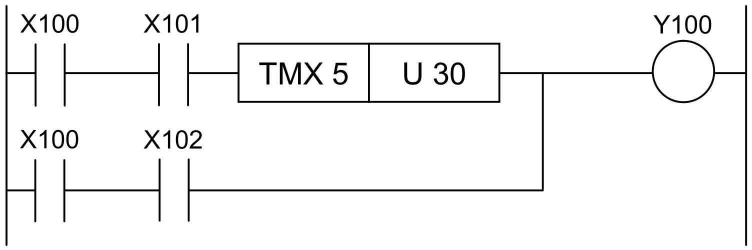
■ Examples for correcting invalid programs

- The invalid programs above can be respectively corrected in the following manner.

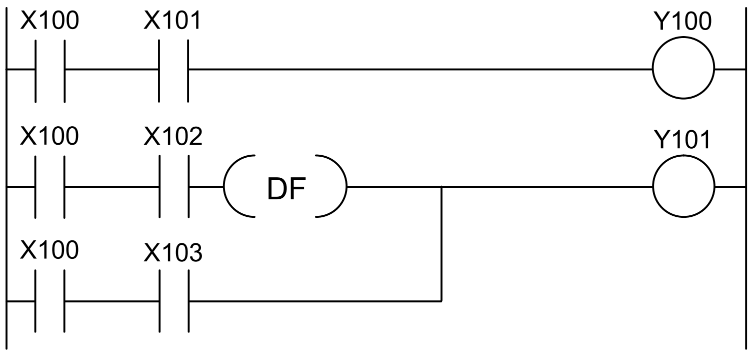
Program for correcting example 1



Program for correcting example 2



Program for correcting example 3

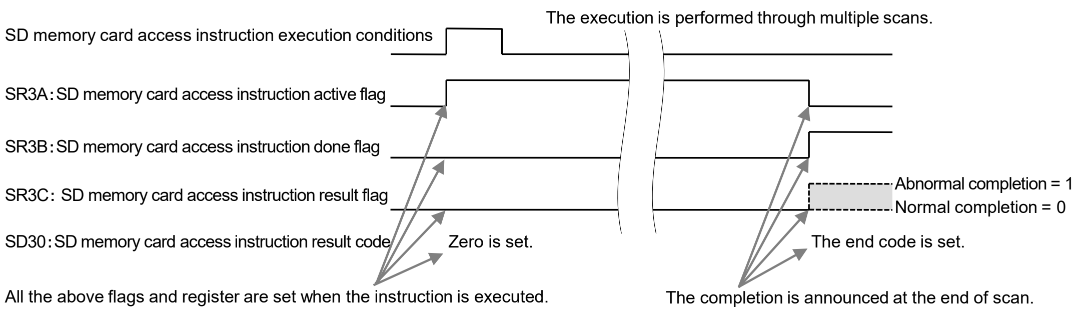


19.9 Common Precautions for SD Memory Card Access Instructions

■ Instruction operation

- At the start of instruction execution, checks are conducted, whether an SD memory card is inserted or not, if the cover is closed, and whether the card is write-protected or not.
- During execution, the SD memory card access instruction active flag (SR3A) is ON, and the execution completion flag (SR3B) is OFF.
- During the execution, the SD memory card access instruction active flag (SR3A) is OFF, and the execution done flag (SR3B) is ON.
- The execution is performed through multiple scans.
- On completion of execution, the SD memory card access instruction execution result flag (SR3C) turns ON or OFF according to the result, and the execution end code is stored in system data register SD30.
- Use the execution result flag to judge whether the SD memory card access instruction is completed normally or abnormally when the execution done flag turns ON. The contents of errors are stored in system data register SD30.
- Only one SD memory card access instruction can be executed at a time. To execute more than one instruction, perform exclusive control using flags such as the SD memory card access instruction active flag.
- If another SD memory card access instruction is being executed when starting an instruction, the new instruction cannot be executed.

■ Flag operation



(Note 1) However, when errors such as no SD memory card, SD memory card write protected, or improper SD memory card file name length are detected, the completion is announced at the start of instruction execution without turning ON the active flag.

■ Error Codes Table

Error code	Description	Cause	Types of detected instructions and the timing
0	Normal end		
1	No SD memory card	No SD memory card is inserted, or the cover is open.	All SD memory card access instructions, at the start of instruction execution
2	SD memory card write-protected	The SD memory card is write-protected.	Write, delete, move, copy, and rename instructions

Error code	Description	Cause	Types of detected instructions and the timing
3	Specified file name error	Code that cannot be specified for a file name is used. There are too many hierarchies for the specified folder.	Folder access and file access instructions
4	No specified file	The specified file does not exist.	Folder access and file access instructions
5	File already exists	The specified file already exists.	Move, copy, and rename instructions
6	File read error		At the time of reading
7	File write error	A write-protect attribute is set for the specified file.	Write, delete, move, copy, and rename instructions
8	File access position error	The reading position or writing position is incorrect.	CWT/CRD, CRD1. At the time of execution.
9	SD memory card capacity shortage	Cannot be executed because there is not enough free space on the SD memory card.	Write, delete, move, copy, and rename instructions
10	Read format error	Error in the conversion format when reading a file.	CRD. At the time of execution.
11	File access contention	A file that is being logged is specified. A file that is being accessed via FTP is specified.	Write, delete, move, copy, and rename instructions
12	The specified directory is not empty.	A directory or a file exists under the directory to be deleted.	CRMDIR. At the time of execution.
-1 to -99	Others		All instructions

■ How to specify folder and file names in SD memory card access instructions

- Specify the full path (up to 256 characters). Do not specify the drive name.
  Example) When specifying abc.txt directly under the root directory: When specifying def.txt under the folder A:
  \abc.txt \a\def.txt
- When using the SSET instruction;
  SSET "\abc.txt" DT0
  Specify as above, and specify DT0 in the file name specification of the SD memory card access instruction.
- When specifying a non-existent folder with CWT, CMKDIR or CPR instruction, only a sub folder directly under the parent folder can be automatically created. Two or more new folders cannot be created by one instruction. If specifying more than two files, no specified file error occurs.

■ SD memory card control specifications

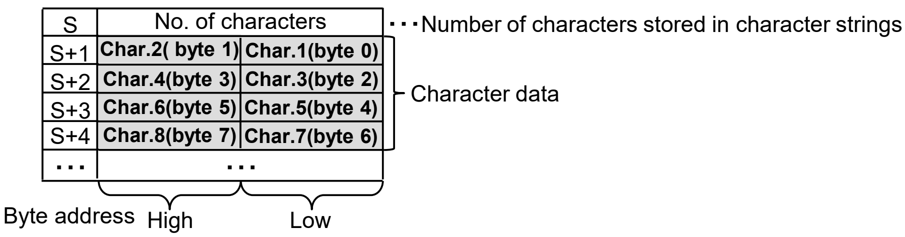
	SD	SDHC
File system	FAT16	FAT32
Max. length of file name	Supports long file names (VFAT)	
Max. volume size	2GB	32GB

	SD	SDHC
Max. file size	2GB	4GB
Max. number of files (8.3 format): Root directory	512	65535
Max. number of files (8.3 format): Sub directory	65534	65534
Max. number of files (long format): Root directory	170	21845
Max. number of files (long format): Sub directory	56634	65534

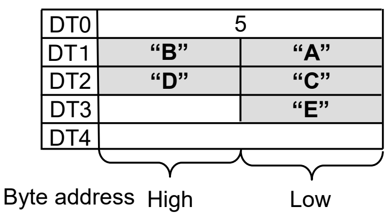
Items	Description
Long file name	256 bytes for full path (when specifying route with \. It is 255 bytes when omitting route\.)
File name/Directory name	ASCII characters (H20 to H7E) / One-byte katakana (HA1 to HDF)
	Japanese (S-JIS code) (H'8140~H'9FFC,H'E040~H'EAA4)

■ Common precautions

- Error flags are not cleared even when normal arithmetic operations are performed. Use ERR instruction for clearing error flags.
- An SD memory card access instruction cannot be executed when another SD memory card access instruction is already being executed. Do not execute the SD memory card access instruction until the active instruction is complete.
- It may take several scans for the processing to complete.
- Use in an interrupt program is not possible.
- Character string data is set in the order of number of characters then character data.



<Example> When specifying 5 for the number of characters, and "ABCDE" for character data



- Include the extension in the file name specification.

19.10 Ethernet Function: IP Addresses

■ IP address setting specifications

Address range	Remarks
000.000.000.001 to 000.255.255.255	Although this range can be set, try not to use it wherever possible.
001.000.000.000 to 126.255.255.255	
128.000.000.000 to 223.255.255.255	

■ List of conditional IP addresses

○: Available, ×: Not available, △: Self IP address is not available, default gateway is available

Address range	Setting using instructions		
	IPv4SET (Self IP address setting)	CONSET (Destination address setting)	FTPcSV HTTPcSV SMTPcSV (Server address setting)
000,000,000,000	△	×	×
127.000.000.000 to 127.255.255.255	×	○	○
224.000.000.000 to 224.255.255.255	×	○	○
:	×	○	○
239.000.000.001 to 239.255.255.255	×	○	○
240.000.000.001 to 240.255.255.255	×	○	○
:	×	○	○
247.000.000.001 to 247.255.255.255	×	○	○
248.000.000.001 to 248.255.255.255	×	○	○
:	×	○	○
255.000.000.001 to 255.255.255.254	×	○	○
255,255,255,255	×	○	×

(Note 1) When an IP address that cannot be set is specified with an instruction, an operation error will not occur and the error flags of CY (SR9) and SD29 will be set.

■ Net mask setting

Masked bits should be left-justified for net mask setting. The following specifications are invalid.

Input notation	Binary notation
255.255.253.0	11111111. 11111111. 11111101. 00000000

■ Default gateway setting

- Setting may not be possible depending on the combination of IP address and default gateway.
- Specify "000.000.000.000" when default gateway is not to be used.
- Setting is not possible in the following case.
  (IP address AND netmask) ≠ (Default gateway address AND netmask)

■ Judgment based on the combination of IP address and net mask

- The following combination is not possible.
- IP address AND (Inverse all bits of net mask: 1's complement) = 0
- IP address OR (net mask) = 255.255.255.255
- Only when the router IP address is other than 000.000.000.000, the above combination judgment is performed for the routing setting.

*The combination above may occur when masks are set to omission using IPv4SET instruction.

Example) When the net mask is 255.255.0.0, set the IP address = 0.0.255.255 using IPv4SET.

The set values for IP address, net mask and default gateway are initialized when communication process is performed using the combination above. Default values are as follows.

IP address =192.168.1.5, Net mask = 255.255.255.0, Default gateway = 192.168.1.1

19.11 Ethernet Function: I/O Allocation

■ I/O relays that are related to Ethernet functions

For using each function of the built-in ET-LAN, the following I/O areas are occupied.

Unit type		Application	Number of occupied words (Number of occupied points)	
			Input	Output
CPU unit	CPU unit Built-in ET-LAN	Common occupied area	1 word (16 points) WX6	-
		User connections 1 to 16	3 word (48 points) WX7-WX9	3 word (48 points) WY7 to WY9
		User connections 17 to 216	Max. 26 words (416 points) WX11 to WX36	Max. 26 words (416 points) WX11 to WX36

(Note 1) Input/output contacts of the CPU unit are allocated for using the functions of each cassette. Regardless of use of such functions, input occupies 10 words (160 points, WX0 to WX9) and output occupies 10 words (160 points, WY0 to WY9).

(Note 2) Occupied area in the area of user connections 17 to 216 varies according to the number of used connections.

(Note 3) The starting numbers of I/O contacts of each unit including the CPU unit can be changed by the setting of tool software.

■ Common occupied areas that are used for Ethernet functions

When using the Ethernet-related functions, flags for confirming the initialization, connection of network and the completion of preparation are allocated.

Address	Application		
X60	Disconnection detection relay	1 = Disconnect	0 = Connect
X61	Ethernet initialization active	1 = During initialization	0 = Completed
X62	IP address established	1 = Establish	0 = Not establish
X63	TCP-NODELAY option	1 = Enabled	0 = Disable
X64	FTP server preparation done	1 = Preparation done	0 = Unusable
X65	FTP client preparation done	1 = Preparation done	0 = Unusable
X66	Reserved for system		
X67	Reserved for system		
X68	HTTP client preparation done	1 = Preparation done	0 = Unusable
X69	Mail send (SMTP client) preparation done	1 = Preparation done	0 = Unusable
X6A	Reserved for system		
X6B	EtherNet/IP preparation done	1 = Preparation done	0 = Other than preparation done

Address	Application		
X6C	EtherNet/IP cyclic communication/all nodes	1 = Communicating (normal)	0 = Other than the above
X6D	EtherNet/IP cyclic communication/all nodes	1 = Stop	0 = Other than the above
X6E	EtherNet/IP abnormal node	1 = Exists	0 = Does not exist
X6F	EtherNet/IP start/stop	1 = Controllable	0 = Not controllable

(MEMO)

20 Appendix

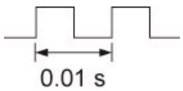
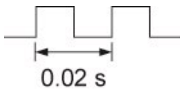
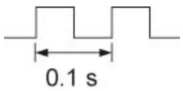
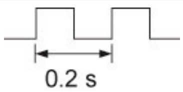
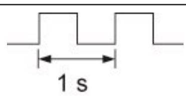
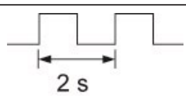
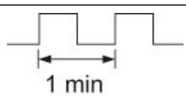
20.1 List of System Relays

WS0

Device No.	Name	Description
SR0	Self-diagnostic error flag	Turns on when a self-diagnosis error occurs. Self-diagnosis error codes are saved in the system data register SD0.
SR1	Unit alarm occurrence	Turns on when a unit alarm is detected. The slot number of the unit where an alarm has occurred is saved in the system data register SD1.
SR2	Unit error occurrence	Turns on when a unit error is detected. The slot number of the unit where an error has occurred is saved in the system data register SD2.
SR3	Unit warning occurrence	Turns on when a unit warning is detected. The slot number of the unit where a warning has occurred is saved in the system data register SD3.
SR4	Unit verification error occurrence	Turns on when an I/O verification error is detected. The slot number of the unit where an I/O verification error has occurred is saved in the system data register SD4.
SR5	Unit installation error detection	Turns on when a unit installation error is detected. The slot number of the unit where an I/O verification error has occurred is saved in the system data register SD5.
SR6	Memory configuration inconsistency detection relay	Turns ON when an inconsistency is detected in memory configuration. The setting where an error has occurred is saved in the system data register SD6.
SR7	Operation error flag (hold type)	Turns on when an operation error occurs after the unit has started operating, and remains on while the unit operation continues. The PB number where an error has occurred is saved in the system data SD7, and the address is saved in system data registers SD8 to SD9. It indicates the first operation error that has occurred.
SR8	Operation error flag (latest type)	Turns on every time an operation error occurs. The PB number where an operation error has occurred is saved in the system data register SD10, and the address is saved in system data registers SD11 to SD12. Every time a new error occurs, the data are updated. It does not turn off even if the instruction is normally completed after the occurrence of the error(s). In order to check if any error has occurred in a specific instruction, either see address data saved in the SD, or clear error flags using ERR instruction immediately before the specific instruction, and check flags immediately after executing the instruction.
SR9	Carry flag (CY flag)	Used in shift instruction and rotate instruction with a carry flag. The flag can also be operated in carry set instruction and carry reset instruction. It is not set in overflow or underflow of operation results. Turns on when an error occurs during the execution of Ethernet communication instruction. The error code is stored in the system data register SD29.
SRA	> flag	Executes comparison instruction, and turns on if the result is larger.
SRB	= flag	Executes comparison instruction, and turns on if the result is equal. Executes operation instruction, and turns on if the result is '0'.
SRC	< flag	Executes comparison instruction, and turns on if the result is smaller.
SRD	Support timer instruction flag	Turns on after support timer instruction (SPTM) is executed and subsequently specified time has passed. Turns off when execution conditions go off.
SRE	All error alarms relay	Turns on when any of the error alarm relays E0 to E4095 turns on. Turns off once all of the error alarm relays go off.

Device No.	Name	Description
SRF	Constant scan error flag	Turns on if scan time exceeds the setting during constant scan. It also turns on if '0' is set in FP7 configuration.

WS1

Device No.	Name	Description	
SR10	Normally ON relay	Is normally on.	
SR11	Normally-off relay	Is normally off.	
SR12	Scan relay	Turns on or off in each scan.	
SR13	Initial pulse relay (ON)	Goes on for only the first scan after operation (RUN) has been started, and goes off for the second and subsequent scans.	
SR14	Initial pulse relay (OFF)	Goes off for only the first scan after operation (RUN) has been started, and goes on for the second and subsequent scans.	
SR15	Step ladder Initial pulse relay (ON)	Turns on in the first scan only, following startup of any single process, during stepladder control.	
SR16	PB initial relay (ON)	Turns on at the start of execution of a program block. Turns off in the next scan.	
SR17	PB initial relay (OFF)	Turns off at the start of execution of a program block. Turns on in the next scan.	
SR18	0.01-second clock pulse relay	Clock pulse with a 0.01-second cycle	
SR19	0.02-second clock pulse relay	Clock pulse with a 0.02-second cycle	
SR1A	0.1-second clock pulse relay	Clock pulse with a 0.1-second cycle	
SR1B	0.2-second clock pulse relay	Clock pulse with a 0.2-second cycle	
SR1C	1-second clock pulse relay	Clock pulse with a 1-second cycle	
SR1D	2-second clock pulse relay	Clock pulse with a 2-second cycle	
SR1E	1-minute clock pulse relay	Clock pulse with a 1-minute cycle	
SR1F	Not used		

WS2

Device No.	Name	Description
SR20	CPU operation modes	ON: RUN mode

Device No.	Name	Description
		OFF: PROG mode
SR21	Operation program memory	ON: SD memory card OFF: ROM
SR22	RTC data error	Turns on if an error is detected in calendar timer data when the unit is powered on.
SR23	Power supply unit service lifetime warning	Turns on when it is detected that a power supply unit is close to its lifetime.
SR24	RTC backup battery Error flag 1 (Hold type)	Turns on when an RTC backup battery error is detected. The flag turns on if the battery is out, even if battery error alarm is disabled in the configuration menu. Once a battery error has been detected, this is held even after recovery has been made. The flag is turned off when power supply is cut off.
SR25	RTC backup battery Error flag 1 (Current type)	Turns on when an RTC backup battery error is detected. It is OFF in the normal state. The flag turns on if the battery is out, even if battery error alarm is disabled in the configuration system register.
SR26	SNTP time update fail	Turns on if acquisition of time data has failed during time synch via LAN port. Is off in the normal status.
SR27	SNTP time update completed	Turns off when time is being updated with SNTP, and turns on when the update is completed. (Note 1)
SR28	(Not used)	
SR29	Forcing flag	Turns on while forced input/output operations are in progress.
SR2A	Interrupt enable	Turns on when interrupt is enabled.
SR2B	Interrupt error flag	Turns on when an interrupt error occurs.
SR2C	Interrupting flag	Turns on when an interrupt program is being executed. Only valid within a PB for execution at a specified interval or within an INT program.
SR2D	PB for execution at a specified interval in progress	Turns on when a PB (program block) for execution at a specified interval is being executed.
SR2E	(Not used)	
SR2F	Rewriting during RUN completed	Turns on in the first scan only following completion of rewriting during RUN.

(Note 1) Available from the CPU unit Ver.3.03 or later.

WS3

Device No.	Name	Description
SR30	SD slot cover status flag	ON: Cover open OFF: Cover closed
SR31	SD memory card attachment flag	ON: There is an SD memory card OFF: No SD memory card
SR32	SD memory card recognition completed flag	ON: Completed recognition of an SD memory card OFF: Other than the above
SR33	SD memory card recognition result flag	ON: Error OFF: Normal

Device No.	Name	Description
SR34	SD memory card write protection flag	ON: Protected OFF: Not protected
SR35	SD memory card type	ON: SD OFF: SDHC
SR36	SD memory card file system	ON: FAT16 OFF: FAT32
SR37	Logging into FTP server	Turns on while logging in.
SR38	Logging trace execution	ON: Executing OFF: Not in operation
SR39	Logging trace start	ON: Active OFF: Not in operation
SR3A	SD memory card access instruction active	This relay is used to check whether other SD memory card access instructions are executed or not. ON: Executing OFF: Not in operation
SR3B	SD memory card access instruction execution complete	This relay is used to check the completion of SD memory card access instruction with the change of this flag (ON to OFF), and used to turn off the trigger of the instruction. ON: Execution complete OFF: Executing
SR3C	SD memory card access instruction execution result	The execution result of SD memory card access instruction is stored. Error codes are stored in system data register SD30. ON: Error OFF: Normal
SR3D	SD card logging graph aggregation in process	ON: Aggregating OFF: Aggregation complete
SR3E	(Not used)	
SR3F	SD memory card is being accessed. Power OFF	Turns on if the CPU unit is powered off while accessing an SD memory card.

WS5

Device No.	Name	Description
SR50	MW unit error alarm relay (1st unit)	Turns on when an error occurs in the FP7 MW Unit. The error code and unit number is stored in the system registers SD90 to SD95.
SR51	MW unit error alarm relay (2nd unit)	
SR52	MW unit error alarm relay (3rd unit)	
SR53	MW unit error alarm relay (4th unit)	
SR54	MW unit error alarm relay (5th unit)	
SR55	MW unit error alarm relay	

Device No.	Name	Description
	(6th unit)	

WS10 (Logging trace control relays: For LOG0)

Device No.	Name	Description
SR100	Logging trace execution	Turns on when the logging trace is performed. Other relays in LOGn turns off while this relay turns on. Storing data in the buffer memory is executed while this relay turns on.
SR101	SD memory card logging active	Turns on when writing files to a SD card becomes enabled after the logging trace execution relay turned on (buffer logging was enabled).
SR102	Logging trace completed	Turns on after the completion of file writing when stopping the logging trace is requested or it is automatically stopped.
SR103	Logging excessive speed relay	● Turns on when the buffer logging speed exceeds the writing speed to an SD memory card in logging operation. Turns on when the number of data previously stored and the number of data stored this time increase. ● Turns on at the timing of buffer logging, and turns off at the timing of buffer logging or the end of scan.
SR104	Buffer overflow	● Turns on when the buffer memory has been exhausted. At that time, new data cannot be stored. ● The value of the buffer overflow counter SD120 is incremented (+1). In that case, writing to the SD memory card does not stop. ● Turns off at the end of scan when buffer vacancy occurs while writing to a SD memory card is performed. The buffer overflow counter SD120 is cleared to 0. ● After buffer vacancy occurred, data logging is executed at the timing of logging to the buffer.
SR105	Logging trace error	Turns on when an error is detected during the logging trace and stops the logging trace.
SR106	No SD memory card free space	Turns on when an error is detected during the logging trace and stops the logging trace.
SR107	Device and trigger setting error	Turns on when an error is detected in set values during the startup operation. The error relay SR105 also turns on. At that time, the execution relay SR100 does not turn on as the logging trace function cannot be started.
SR108	Tracing stop trigger monitor	Monitors a registered trace stop trigger when executing tracing. Turns on when conditions are met.
SR109	Trace data acquisition complete	Turns on after logging data for a specified number of times after detecting the tracing stop trigger during the execution of trace.
SR10A - SR10F	(Not used)	

WS11 - WS25 (Logging trace control relays: For LOG1 - LOG15)

Device No.	Name	Description
SR110 -SR119	Logging trace control relay For LOG1	For the details of each control relay, refer to the previous page.
SR120 -SR129	Logging trace control relay For LOG2	
SR130	Logging trace control relay	

Device No.	Name	Description
-SR139	For LOG3	
SR140 -SR149	Logging trace control relay For LOG4	
SR150 -SR159	Logging trace control relay For LOG5	
SR160 -SR169	Logging trace control relay For LOG6	
SR170 -SR179	Logging trace control relay For LOG7	
SR180 -SR189	Logging trace control relay For LOG8	
SR190 -SR199	Logging trace control relay For LOG9	
SR200 -SR209	Logging trace control relay For LOG10	
SR210 -SR219	Logging trace control relay For LOG11	
SR220 -SR229	Logging trace control relay For LOG12	
SR230 -SR239	Logging trace control relay For LOG13	
SR240 -SR249	Logging trace control relay For LOG14	
SR250 -SR259	Logging trace control relay For LOG15	

WS90

Device No.	Name	Description
SR900	Operation history No.0 active relay	ON: Active OFF: Not in operation
SR901	Operation history No.1 active relay	
SR902	Operation history No.2 active relay	
SR903	Operation history No.3 active relay	
SR904	Operation history No.4 active relay	
SR905	Operation history No.5 active relay	
SR906	Operation history No.6 active relay	
SR907	Operation history No.7 active relay	

Device No.	Name	Description
SR908 - SR90F	(Not used)	

WS91

Device No.	Name	Description
SR910	Operation history No.0 device settings error relay	ON: Error OFF: Normal
SR911	Operation history No.1 device settings error relay	
SR912	Operation history No.2 device settings error relay	
SR913	Operation history No.3 device settings error relay	
SR914	Operation history No.4 device settings error relay	
SR915	Operation history No.5 device settings error relay	
SR916	Operation history No.6 device settings error relay	
SR917	Operation history No.7 device settings error relay	
SR918 - SR91F	(Not used)	

WS100 - WS149

<table>
<tr><th>Device No.</th><th>Name</th><th>Description</th></tr>
<tr><td>SR1000
to SR1499</td><td>Program block
PB active relay</td><td>Can monitor program blocks that are being started up. SR1000 to SR1499 are allocated to 500 PBs.
<table>
<tr><th>Device No.</th><th>PB No.</th></tr>
<tr><td>SR1000</td><td>PB 000</td></tr>
<tr><td>SR1001</td><td>PB 001</td></tr>
<tr><td>SR1002</td><td>PB 002</td></tr>
<tr><td>-----</td><td>-----</td></tr>
<tr><td>SR1009</td><td>PB 009</td></tr>
<tr><td>SR1010</td><td>PB 010</td></tr>
<tr><td>SR1011</td><td>PB 011</td></tr>
<tr><td>-----</td><td>-----</td></tr>
<tr><td>-----</td><td>-----</td></tr>
<tr><td>SR1498</td><td>PB498</td></tr>
<tr><td>SR1499</td><td>PB499</td></tr>
</table>
</td></tr>
</table>

20.2 List of System Data Registers

SD0-SD28

<table>
<tr><th>Device No.</th><th>Name</th><th>Description</th></tr>
<tr><td>SD0</td><td>Self-diagnostic error code</td><td>Stores the error code when a self-diagnosis error occurred.</td></tr>
<tr><td>SD1</td><td>Alarm Occurrence Unit Slot No.</td><td>Saves the slot number of the unit where an alarm has occurred.
Slot No.1 to 64</td></tr>
<tr><td>SD2</td><td>Error Occurrence Unit Slot No.</td><td>Stores the slot number of the unit in which the error occurred or the slot number of the CPU unit with built-in ET-LAN.
Slot No.1 to 64, 100 (CPU unit with built-in ET-LAN)</td></tr>
<tr><td>SD3</td><td>Warning Occurrence Unit Slot No.</td><td>Stores the slot number of the unit in which the warning occurred or the slot number of the CPU unit with built-in ET-LAN.
Slot No.1 to 64, 100 (CPU unit with built-in ET-LAN)
When the number is 0, it indicates the warning for COM0, COM1, or COM2 (communication cassette or SCU port warning).</td></tr>
<tr><td>SD4</td><td>Verification Error Occurrence Unit Slot No.</td><td>Saves the slot number of the unit where a verification error has occurred.
Slot No.1 to 64</td></tr>
<tr><td>SD5</td><td>Installation error detection slot No.</td><td>Saves the slot number of the unit where an installation error was detected.
Slot No.1 to 64</td></tr>
<tr><td>SD6</td><td>Memory configuration inconsistency details</td><td>When a memory configuration inconsistency is detected, information about the setting where the error has occurred is saved
<table>
<tr><th>Bit No.</th><th>Description</th></tr>
<tr><td>0</td><td>Logging/Trace settings</td></tr>
<tr><td>1</td><td>FTPc settings</td></tr>
<tr><td>2</td><td>HTTPc settings</td></tr>
<tr><td>3</td><td>Mailc settings</td></tr>
<tr><td>4</td><td>EIP settings</td></tr>
<tr><td>5</td><td>W-PLC link settings</td></tr>
<tr><td>6</td><td>W2-PLC link settings</td></tr>
</table></td></tr>
<tr><td>SD7</td><td>Operation error occurrence PB number (hold type)</td><td>Saves the PB number where the first operation error occurred after the unit has started operating.</td></tr>
<tr><td>SD8</td><td>Address with operation error (hold type)
(32-bit lower address)</td><td rowspan="2">Saves the address where the first operation error occurred after the unit has started operating. Perform monitoring using 32-bit data.</td></tr>
<tr><td>SD9</td><td>Address with operation error (hold type)
(32-bit higher address)</td></tr>
<tr><td>SD10</td><td>Operation error occurrence PB number (latest type)</td><td>Saves the PB number where an operation error occurred. Every time a new error occurs, the data are updated. The value '0' is recorded at the start of the scan.</td></tr>
<tr><td>SD11</td><td>Address with operation error (new type)</td><td>Saves the address where an operation error occurred. Every time a new error occurs, the data are updated. The value '0' is recorded at the start of the scan. Perform monitoring using 32-bit data.</td></tr>
</table>

Device No.	Name	Description	
	(32-bit lower address)		
SD12	Address with operation error (new type) (32-bit higher address)		
SD13 -SD14	(Not used)		
SD15 -SD16	Remainder of DIVFP2 instruction execution results	The division remainder of the DIVFP2 instruction is stored.	
SD17 -SD18	(Not used)		
SD19	RING counter 2.5 ms	The saved value is increased by one every time the respective time unit has passed. (H0 to HFFFF) Current values of SD19 to SD21 can only be read when SD19 to SD21 are directly specified and read by MV instruction. The scan start value is read by other instructions.	
SD20	RING counter 10 μs		
SD21	RING counter 100 μs		
SD22	Scan time (current value)	Saves the current value.	[Saved value (decimal)] x 10 μs scan time indication: Indicates operation cycle time in the RUN mode only. Max. and Min. values are cleared at switching between the RUN mode and the PROG. mode.
SD23	Scan time (minimum value)	Stores the minimum value.	
SD24	Scan time (maximum value)	Saves the maximum value.	
SD25 -SD26	(Not used)		
SD27	Interval for PB for execution at a specified interval	Saves interval for PB for execution at a specified interval.	
SD28	(Not used)		

(Note 1) SD0 to SD5 are available only when the corresponding system relays SR0 to SR5 are on.

SD29

Device No.	Name	Description
SD29	Ethernet communication error code	Saves the error code when the Ethernet communication instruction is executed.

SR9	SD29
0: Normal	0: Normal
1: Error	1: Incorrect IP address specification
	2: Incorrect subnet mask specification
	3: Incorrect default gateway specification
	4: Incorrect IP addresses are combined.
	10: Ethernet cable disconnected
	11: Ethernet initialization in progress
	12: IP address unestablished

Device No.	Name	Description
		SR9 / SD29: 13: Client not started 14: Connection processing in progress 15: Connection occupied

SR9	SD29
	13: Client not started
	14: Connection processing in progress
	15: Connection occupied

SD30-SD39

Device No.	Name	Description
SD30	SD Memory Card Access Instruction Execution result	Error codes while the SD memory card access instruction is executed are stored.
SD31 -SD39	(Not used)	

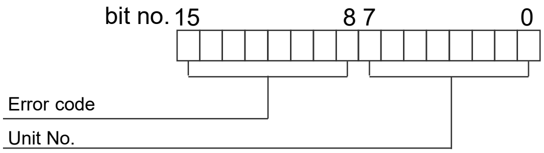
Value	Name	Detailed information
0	Normal end	
1	No SD memory card	No SD memory card is inserted, or the cover is open.
2	SD memory card write-protected	The SD memory card is write-protected.
3	Specified file name error	Code that cannot be specified for a file name is used. There are too many hierarchies for the specified folder.
4	No specified file	The specified file does not exist.
5	File already exists	The specified file already exists.
6	File read error	
7	File write error	A write-protect attribute is set for the specified file.
8	File access position error	The reading position or writing position is incorrect.
9	SD memory card capacity shortage	Cannot be executed because there is not enough free space on the SD memory card.
10	Read format error	Error in the conversion format when reading a file.
11	File access contention	A file that is being logged is specified. A file that is being accessed via FTP is specified.
12	The specified directory is not empty.	A directory or a file exists under the directory to be deleted.
-1 to -99	Others	

SD40-SD49

Device No.	Name	Description
SD40	Slot No.	The slot number specified by the UNITSEL instruction operand [S1] is stored. CPU with built-in SCU U0: U0 to U2 CPU with built-in ET-LAN: U100 Serial Communication Unit (SCU): U1 to U64
SD41	COM port No. or User connection No.	The COM port number or user connection number specified by the UNITSEL instruction operand [S2] is stored. CPU with built-in SCU: U0 to U2 CPU with built-in ET-LAN: U1 to U216 Serial Communication Unit (SCU): U1 to U4
SD42 -SD49	(Not used)	

SD50-SD99

<table>
<tr><th>Device No.</th><th>Name</th><th>Description</th></tr>
<tr><td>SD50</td><td>Calendar timer (year)</td><td rowspan="7">● Saves year, month, day, hour, minute, second and day-of-the-week data of the calendar timer as 16-bit binary data. The built-in calendar timer will operate correctly through the year 2099 and support leap years.
● The calendar timer can be set (time synch) by writing desired values using the programming tool or a program based on calendar setting instruction (TIMEWT).</td></tr>
<tr><td>SD51</td><td>Calendar timer (month)</td></tr>
<tr><td>SD52</td><td>Calendar timer (day)</td></tr>
<tr><td>SD53</td><td>Calendar timer (hours)</td></tr>
<tr><td>SD54</td><td>Calendar timer (minutes)</td></tr>
<tr><td>SD55</td><td>Calendar timer (seconds)</td></tr>
<tr><td>SD56</td><td>Calendar timer (day-of-the-week)</td></tr>
<tr><td>SD60</td><td>Total ON number of error alarm relays</td><td>Saves the total number of error alarm relays that are on. (Max. 4096 relays). By specifying SD60 in RST instruction, all data in the error alarm buffer can be cleared.</td></tr>
<tr><td>SD61</td><td>No.1 error alarm relay that turned on</td><td>Saves the number of the error alarm relay that turned on in the first place (No.1). By specifying SD61 in RST instruction, all data in the error alarm buffer can be cleared.</td></tr>
<tr><td>SD62
-SD79</td><td>No.2 to No.19 error alarm relays that turned on</td><td>Saves the numbers of the error alarm relays that turned on. By specifying the device number in RST instruction, all data of the relevant relay(s) in the error alarm buffer can be cleared. Device numbers of system data registers SDs and error alarm relays correspond as follows.
<table>
<tr><th>Device No.</th><th>Error alarm relay</th></tr>
<tr><td>SD62</td><td>No.2</td></tr>
<tr><td>SD63</td><td>No.3</td></tr>
<tr><td>SD64</td><td>No.4</td></tr>
<tr><td>SD65</td><td>No.5</td></tr>
<tr><td>SD66</td><td>No.6</td></tr>
<tr><td>SD67</td><td>No.7</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Device No.</th><th colspan="2">Name</th><th>Description</th></tr>
<tr><td></td><td colspan="2"></td><td><table><tr><th>Device No.</th><th>Error alarm relay</th></tr><tr><td>SD68</td><td>No.8</td></tr><tr><td>SD69</td><td>No.9</td></tr><tr><td>SD70</td><td>No.10</td></tr><tr><td>SD71</td><td>No.11</td></tr><tr><td>SD72</td><td>No.12</td></tr><tr><td>SD73</td><td>No.13</td></tr><tr><td>SD74</td><td>No.14</td></tr><tr><td>SD75</td><td>No.15</td></tr><tr><td>SD76</td><td>No.16</td></tr><tr><td>SD77</td><td>No.17</td></tr><tr><td>SD78</td><td>No.18</td></tr><tr><td>SD79</td><td>No.19</td></tr></table></td></tr>
<tr><td>SD80</td><td rowspan="6">For error alarm relay</td><td>Calendar timer (year)</td><td rowspan="6">Saves time when the error alarm relay saved in SD61 turned on.</td></tr>
<tr><td>SD81</td><td>Calendar timer (month)</td></tr>
<tr><td>SD82</td><td>Calendar timer (day)</td></tr>
<tr><td>SD83</td><td>Calendar timer (hours)</td></tr>
<tr><td>SD84</td><td>Calendar timer (minutes)</td></tr>
<tr><td>SD85</td><td>Calendar timer (seconds)</td></tr>
<tr><td>SD86
-SD89</td><td colspan="2">(Not used)</td><td></td></tr>
<tr><td>SD90</td><td colspan="2">FP7 MW Unit error annunciation register 1st unit</td><td rowspan="5">When an error occurs in the FP7 MW Unit, the error code is stored in the high byte and the unit number is stored in the low byte.
</td></tr>
<tr><td>SD91</td><td colspan="2">FP7 MW Unit error annunciation register 2nd unit</td></tr>
<tr><td>SD92</td><td colspan="2">FP7 MW Unit error annunciation register 3rd unit</td></tr>
<tr><td>SD93</td><td colspan="2">FP7 MW Unit error annunciation register 4th unit</td></tr>
<tr><td>SD94</td><td colspan="2">FP7 MW Unit error annunciation register 5th unit</td></tr>
</table>

Device No.	Name	Description
SD95	FP7 MW Unit error annunciation register 6th unit	
SD96 -SD99	(Not used)	

SD100 - SD115 and SD120 - SD135 (For logging trace control)

Device No.	Name	Description
SD100	Buffer free space for LOG0	Stores free space of buffer memory during logging. Unit: kB
SD101	Buffer free space for LOG1	
SD102	Buffer free space for LOG2	
SD103	Buffer free space for LOG3	
SD104	Buffer free space for LOG4	
SD105	Buffer free space for LOG5	
SD106	Buffer free space for LOG6	
SD107	Buffer free space for LOG7	
SD108	Buffer free space for LOG8	
SD109	Buffer free space for LOG9	
SD110	Buffer free space for LOG10	
SD111	Buffer free space for LOG11	
SD112	Buffer free space for LOG12	
SD113	Buffer free space for LOG13	
SD114	Buffer free space for LOG14	
SD115	Buffer free space for LOG15	
SD120	Buffer overflow counter for LOG0	● Saves the number of times buffer overflow flags (e.g. SR104 for LOG0) turn on. ● For checking the number of times logging data is lost during the buffer overflow, register the buffer overflow counter as logging data.
SD121	Buffer overflow counter for LOG1	
SD122	Buffer overflow counter for LOG2	
SD123	Buffer overflow counter for LOG3	
SD124	Buffer overflow counter for LOG4	
SD125	Buffer overflow counter for LOG5	
SD126	Buffer overflow counter for LOG6	
SD127	Buffer overflow counter for LOG7	
SD128	Buffer overflow counter for LOG8	
SD129	Buffer overflow counter for LOG9	
SD130	Buffer overflow counter for LOG10	
SD131	Buffer overflow counter for LOG11	
SD132	Buffer overflow counter for LOG12	
SD133	Buffer overflow counter for LOG13	

Device No.	Name	Description
SD134	Buffer overflow counter for LOG14	
SD135	Buffer overflow counter for LOG15	

SD220-SD254

Device No.	Name	Description
SD220	Operation history No.0 Number of records	Stores the number of accumulated operation history records.
SD221	Operation history No.1 Number of records	
SD222	Operation history No.2 Number of records	
SD223	Operation history No.3 Number of records	
SD224	Operation history No.4 Number of records	
SD225	Operation history No.5 Number of records	
SD226	Operation history No.6 Number of records	
SD227	Operation history No.7 Number of records	
SD228 -SD229	(Not used)	
SD230	Operation history No.0 Number of free buffers	Stores the number of free buffers for operation history.
SD231	Operation history No.1 Number of free buffers	
SD232	Operating history No.2 Number of free buffers	
SD233	Operation history No.3 Number of free buffers	
SD234	Operation history No.4 Number of free buffers	
SD235	Operating history No.5 Number of free buffers	
SD236	Operating history No.6 Number of free buffers	
SD237	Operating history No.7 Number of free buffers	
SD238 -SD239	(Not used)	
SD240	Operation history No.0 Update counter	Stores a value +1 for each operation history record accumulated.

Device No.	Name	Description
SD241	Operation history No.1 Update counter	
SD242	Operation history No.2 Update counter	
SD243	Operation history No.3 Update counter	
SD244	Operation history No.4 Update counter	
SD245	Operation history No.5 Update counter	
SD246	Operation history No.6 Update counter	
SD247	Operation history No.7 Update counter	
SD248 -SD254	(Not used)	

20.3 Error Codes Table

■ Self-diagnostic error code

Error codes 1 to 6

Code	Name	Operating	Error contents and steps to take
1	CPU hardware error 1	Stop	There may be a hardware problem. Please contact your dealer.
2	CPU hardware error 2	Stop	
3	I/O bus power supply error (including "no end cover")	Stop	Error in the I/O bus part is probable, such that the end unit has not been attached. Double-check the attachment status of units. This error also occurs when an expansion cable is removed during operation.
4	Unit attachment limit exceeded	Stop	is probable that the unit attachment limit has been exceeded. Double-check the configuration.
5	Project data error	Stop	Turns on when there is an error in project data.
6	Expansion unit power supply synchronization error	Stop	Turns on when there is an error in the expansion block side such that the power to the expansion unit is not on or the expansion cable is not connected correctly when the power turns on. The wait time until the power turns on can be set in the I/O map configuration dialog box.

Error codes 20 to 27

Code	Name	Operating	Error contents and steps to take
20	Syntax error	Stop Auto clear	A program with a syntax error has been written. Switch to the PROG. mode and correct the error.
21	Duplicated use	Stop Auto clear	The same relay is used multiple times in OT instruction, etc. Switch to the PROG. mode and correct the error. Or, set the duplicated output to “enable” in the CPU configuration. Applicable devices and instructions are as follows. ● Operation device (X, Y, R, L), timer/counter instruction, SSTP instruction
22	Not paired	Stop Auto clear	For instructions which must be used in a pair, one instruction is either missing or in an incorrect position. Switch to the PROG. mode and enter the two instructions which must be used in a pair in the correct positions. Applicable instructions and cases are as follows. ● MC and MCE are not paired ● LBLs corresponding to LOOP and JP are not located in the same area (normal program area / same sub-routine area / same interrupt program area) ● There are no sub-routines corresponding to CALL and FCALL ● There is no STPE corresponding to SSTP
24	Program area error	Stop Auto clear	An instruction that can only executed in a specified area is written in another location. Switch to the PROG. mode and correct the error. Applicable instructions and cases are as follows.

Code	Name	Operating	Error contents and steps to take
			● LBL, LOOP, JP, MC and MCE are in the stepladder area ● MC is nested in more than 16 layers ● CNDE is outside the normal program area ● EDPB is outside the free area ● ED is inside the sub-routine area or the interrupt area ● SBL is outside the free area or the sub-routine area ● An interrupt program is outside the free area or the interrupt program area ● RET is outside the sub-routine area ● IRET is outside the interrupt program area ● STPE is outside the stepladder area
25	High-level instruction execution combination error	Stop Auto clear	In the program, high-level instructions, which execute in every scan and at the leading edge of the trigger, are programmed to be triggered by one contact. Correct the program so that the high-level instructions executed in every scan and only at the leading edge are triggered separately.
27	Compile memory full error	Stop Auto clear	The program is too large to compile in the program memory. Switch to the PROG. mode and reduce the total number of steps for the program.

(Note 1) For errors where "Auto clear" is indicated in the 'Operation' column, error clearance is executed when power supply is cut off, or when the unit is set to the RUN mode again after the status has been corrected.

Error codes 40 to 79

Code	Name	Operating	Error contents and steps to take
40	Copy failure Cover open	Stop Auto clear	The card cover is open and the copy process cannot be executed. Close the cover.
41	Copy failure No SD card	Stop Auto clear	Copying cannot be executed because there is no SD memory card. Insert an SD memory card.
42	Copy failure SD card reading error (FAT / file error)	Stop Auto clear	Copying cannot be executed because the SD memory card is damaged. Insert a normal SD memory card.
43	Copy failure No file	Stop Auto clear	Copying cannot be executed because there is no file in the SD memory card. Check if a project file is saved in the card.
44	Copy failure Password inconsistency (Limited distribution function)	Stop Auto clear	Copying cannot be executed because the password for the project file saved in the SD memory card is not consistent with the password for the execution project saved in the built-in ROM. Check the password settings.
45	Copy failure Invalid project data	Stop Auto clear	Copying cannot be executed because an error has been identified in project data saved in the SD memory card. Check the project data.
50	SD not operable Cover open	Stop	SD memory card operation cannot be executed because the card cover is open. Close the cover.

Code	Name	Operatin g	Error contents and steps to take
		Auto clear	
51	SD not operable No SD card	Stop Auto clear	SD memory card operation cannot be executed because there is no SD memory card. Insert an SD memory card.
52	SD not operable SD card reading error (FAT / file error)	Stop Auto clear	SD memory card operation cannot be executed because the SD memory card is damaged. Insert a normal SD memory card.
53	SD not operable No file	Stop Auto clear	SD memory card operation cannot be executed because there is no file in the SD memory card. Check if a project file is saved in the card.
54	SD not operable Password inconsistency (Limited distribution function)	Stop Auto clear	SD memory card operation cannot be executed because the password for the project file saved in the SD memory card is not consistent with the password for the execution project saved in the built-in ROM. Check the password settings.
55	SD not operable Invalid project data	Stop Auto clear	SD memory card operation cannot be executed because an error has been identified in project data saved in the SD memory card. Check the project data.
60	Duplicated or excessive collected I/O maps	Stop Auto clear	There is an error with I/O maps that have been collected in the CPU unit. Verify the registered data.
61	Duplicated or excessive registered I/O maps	Stop Auto clear	There is an error with I/O maps that have been registered in the CPU unit. Verify the registered data.
62	Interrupt error 1	Stop Auto clear	There may be a hardware problem. Please contact your dealer.
63	Interrupt error 2	Stop Auto clear	The interrupt program definition by INTPG instruction may be disappeared by rewriting during RUN. Check the program.
79	Operation congestion error	Stop	Occurs in the simulation function of FPWIN GR7 or FPWIN Pro. It is generated when the number of instruction executions exceeds 16,777,215 per one cycle. Reduce the number of instructions to be executed in one scan. Reduce the number of loops or shorten the program.

(Note 1) For errors where "Auto clear" is indicated in the 'Operation' column, error clearance is executed when power supply is cut off, or when the the same operation is executed again after the status has been corrected.

Error codes 80 to 119

Code	Name	Operating	Error contents and steps to take
80	Unit alarm occurrence	Select (Default stop)	An alarm has occurred in an attached unit. Check the status of the unit in the slot number saved in the system data register SD1.
81	Unit error occurrence	Select (Default stop)	An error has occurred in an attached unit. Check the status of the unit in the slot number saved in the system data register SD2. Verify the configuration settings.

Code	Name	Operating	Error contents and steps to take
82	Unit verification error detection	Select (Default stop)	Unit wiring condition has changed compared to that at the time of power-up. Check the status of the unit in the slot number saved in the system data register SD4.
83	Registered unit number inconsistency	Select (Default stop)	The number of units differs from that registered in the I/O map. Check the I/O map and the attachment status.
84	Unit initial completed Waiting timeout	Select (Default stop)	An error has occurred during the unit initial operation. Check the unit status.
85	Unit configuration data inconsistency with applicable unit	Select (Default stop)	The unit's configuration data is not consistent with the applicable unit. Check the I/O map and the configuration data.
86	Operation error	Select (Default stop)	An operation error has occurred. Reasons for an operation error vary by instruction. Refer to the Instruction Manual, etc. and correct the appropriate reasons. PB and address where an operation error has occurred are saved in the system data registers SD7 to SD12.
87	Communication error occurrence	Select (Default stop)	Communication is not working with the slave unit. Check if the slave unit is in an operable state and also check if the slave unit is correctly connected.
88	Error occurrence in the input/output unit installed on the remote slave	Select (Default stop)	Communication is not working with the slave unit. Check if the slave unit is in an operable state and also check if the slave unit is correctly connected.
89	Slave unit connection wait timeout	Select (Default stop)	Communication is not working with the slave unit. Check if the slave unit is in an operable state and also check if the slave unit is correctly connected.
90	Slave unit refresh synchronization timeout	Select (Default stop)	Communication is not working with the slave unit. Check if the slave unit is in an operable state and also check if the slave unit is correctly connected.
100	Bus current error	Select (Default stop)	A bus error is probable. Please contact your dealer.
104	Service power supply current error	Select (Default stop)	An error has been detected in the GT power supply terminal part. Check if it is correctly connected.
105	CPU temperature error 1	Select (Default stop)	A temperature rise has been detected in hardware. In general, select "Continue".
106	CPU temperature error 2	Select (Default continue)	
118	Built-in LAN unit error	Continue	Check the details in the system history data.
119	Ethernet communication system error	Continue	It is possible that an illegal packet is received. Check if an illegal packet is sent to FP7. Restart the power supply if this error occurs.

(Note 1) For errors where "Select" is indicated in the 'Operation' column, either "Stop" or "Continue" can be selected in the configuration menu.

Error codes 120 to 129, 132 to 135, 1000 to 2999

Code	Name	Operating	Error contents and steps to take
120	RTC data error	Continue	An error has been detected in clock data of the calendar timer.
121	Power supply unit service lifetime warning	Continue	It is alarmed that the power supply unit is close to its lifetime. Replace the power supply unit.
122	Battery voltage decline	Continue	Voltage of the optional battery has declined. Replace the battery. If no battery is used, disable battery error alarm in the CPU configuration.
123	Gold capacitor voltage decline	Continue	It is alarmed that voltage of the built-in gold capacitor of the CPU unit has declined. Charge the CPU unit.
124	SNTP time acquisition failure	Continue	Acquisition of time data has failed during time synch via LAN port.
125	Logging settings mismatch	Continue	An error has been detected in logging data settings.
126	Logging data error	Continue	An error has been detected in logging data.
127	Comment data error	Continue	An error has been detected in comment data.
128	PLC link area error duplication error	Continue	One of the following areas is duplicated between network devices in the PLC link. ● PLC link area ● Automatic transfer PLC link operation state ● Error Information
129	Memory configuration inconsistency	Continue	An inconsistency has been detected in the memory configuration.
132	Operating history buffer error	Continue	There is an error in the operating history buffer.
133	Operating history configuration data error	Continue	There is an error in the operating history configuration data. Check the operating history data settings.
134	Operating history data error (when loading settings from SD card)	Continue	There is an error in the operating history data saved on the SD card.
135	Power supply unit lifetime data read error	Continue	There is a memory error in the power supply unit. Please contact your dealer.
1000 -1999	Error by ERR instruction	Stop	An error as specified by ERR instruction in the user program has occurred. Handle the error in accordance with the specified detection conditions.
2000 -2999	Error by ERR instruction	Continue	

(Note 1) If an RTC data error is detected, the date is set to "April 1, 2012".

■ MEWTOCOL-COM Communication Error Codes

Code	Name	Description of error
!26	Abnormal unit number setting	A command was received that cannot be used globally (station number FF).
!40	BCC error	A transmission error has occurred in the received data.

Code	Name	Description of error
!41	Format error	A command was received that does not match the format.
!42	No support error	An unsupported command was received.
!43	Multiple frames procedure error	Another command was received while multiple frame processing was in progress.
!60	Parameter error	The specified parameter content does not exist or cannot be used.
!61	Data error	There are errors in contact, data area, specification of data No., size specification, range, or format specification.
!62	Registration over error	The number of registrations has been exceeded or the unit has been operated in an unregistered state.
!63	PC mode error	A command that cannot be processed was executed in RUN mode or while SD memory card copy was in progress.
!64	Defective external memory error	The hardware is defective. It is likely an abnormality in the built-in ROM (F-ROM). The specified content exceeded the capacity during ROM transfer. A reading/writing error has occurred.
!65	Protect error	A program or system register write operation was executed while the device was in the protected state (password set).
!66	Address error	There is an error in the address data code format, or if the address data was excessive or insufficient, there was an error in the range setting.
!67	No program error and No data error	Message read and sampling trace start/read was executed on unregistered data.
!68	Cannot write during RUN error	While RUN was in progress, an attempt was made to edit an instruction that cannot be rewritten (ED, SUB, RET, INT, IRET, SSTP, STPE). Nothing can be written to the control unit.
!71	Exclusive access control error	A command was executed that cannot be processed at the same time as the command in progress.
!78	No SD card error	No SD card is installed.
!80	Abnormal guarantee data error	The guarantee data (CRC code) is abnormal.
!81	No valid data error	No valid data exists.
!90	Error during logging trace	A command was executed that cannot be processed during logging trace.
!92	Unsupported SD card error	The SD card is not an industrial SD card made by Panasonic.

■ MEWTOCOL7-COM Communication Error Codes

Code	Name	Description of error
41	Format error	A command was received that does not match the format.
42	No support error	An unsupported command was received.
60	Parameter error	The specified parameter content does not exist or cannot be used.
61	Data error	There are errors in contact, data area, specification of data No., size specification, range, or format specification.
62	Registration over error	The number of registrations has been exceeded or the unit has been operated in an unregistered state.

Code	Name	Description of error
63	PC mode error	A command that cannot be processed was executed in RUN mode or PROG mode.
71	Exclusive access control error	A command was executed that cannot be processed at the same time as the command in progress.
80	Abnormal guarantee data error	The guarantee data (CRC code) is abnormal.
91	Missing expansion slave unit error	A slot number has been specified in a range where no expansion slave unit is mounted.

20.4 ASCII Code Table, JIS8 Code Table

■ Reference Table: ASCII Codes

								b7								
								b6	0	0	0	0	1	1	1	1
								b5	0	0	1	1	0	0	1	1
								b4	0	1	0	1	0	1	0	1
b7	b6	b5	b4	b3	b2	b1	b0	R \ C	0	1	2	3	4	5	6	7
				0	0	0	0	0	NUL	DEL	SPACE	0	@	P	`	p
				0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
				0	0	1	0	2	STX	DC2	"	2	B	R	b	r
				0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
				0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
				0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
				0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
				0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
				1	0	0	0	8	BS	CAN	(	8	H	X	h	x
				1	0	0	1	9	HT	EM	)	9	I	Y	i	y
				1	0	1	0	A	LF	SUB	*	:	J	Z	j	z
				1	0	1	1	B	VT	ESC	+	;	K	[	k	{
				1	1	0	0	C	FF	FS	,	<	L	¥	l	|
				1	1	0	1	D	CR	GS	-	=	M	]	m	}
				1	1	1	0	E	SO	RS	.	>	N	^	n	~
				1	1	1	1	F	SI	US	/	?	O	_	o	DEL

■ Reference Table: JIS8 Codes

<table>
<tr><td colspan="5">b7</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</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td colspan="5">b6</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td colspan="5">b5</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td colspan="5">b4</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td>C / R</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>NUL</td><td>TC$_7$ (DEL)</td><td>(SP)</td><td>0</td><td>@</td><td>P</td><td>`</td><td>p</td><td rowspan="16">Undefined</td><td rowspan="16">Undefined</td><td>Undefined</td><td>ー</td><td>タ</td><td>ミ</td><td rowspan="16">Undefined</td><td rowspan="16">Undefined</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>TC$_1$ (SOH)</td><td>DC$_1$</td><td>!</td><td>1</td><td>A</td><td>Q</td><td>a</td><td>q</td><td>。</td><td>ア</td><td>チ</td><td>ム</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td><td>TC$_2$ (STX)</td><td>DC$_2$</td><td>"</td><td>2</td><td>B</td><td>R</td><td>b</td><td>r</td><td>「</td><td>イ</td><td>ツ</td><td>メ</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td><td>TC$_3$ (ETX)</td><td>DC$_3$</td><td>#</td><td>3</td><td>C</td><td>S</td><td>c</td><td>s</td><td>」</td><td>ウ</td><td>テ</td><td>モ</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4</td><td>TC$_4$ (EOT)</td><td>DC$_4$</td><td>$</td><td>4</td><td>D</td><td>T</td><td>d</td><td>t</td><td>、</td><td>エ</td><td>ト</td><td>ヤ</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td><td>TC$_5$ (ENQ)</td><td>TC$_8$ (NAK)</td><td>%</td><td>5</td><td>E</td><td>U</td><td>e</td><td>u</td><td>・</td><td>オ</td><td>ナ</td><td>ユ</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6</td><td>TC$_6$ (ACK)</td><td>TC$_9$ (SYN)</td><td>&</td><td>6</td><td>F</td><td>V</td><td>f</td><td>v</td><td>ヲ</td><td>カ</td><td>ニ</td><td>ヨ</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7</td><td>BEL</td><td>ETB</td><td>'</td><td>7</td><td>G</td><td>W</td><td>g</td><td>w</td><td>ァ</td><td>キ</td><td>ヌ</td><td>ラ</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8</td><td>EE$_0$ (BS)</td><td>CAN</td><td>(</td><td>8</td><td>H</td><td>X</td><td>h</td><td>x</td><td>ィ</td><td>ク</td><td>ネ</td><td>リ</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>9</td><td>EE$_1$ (HT)</td><td>EM</td><td>)</td><td>9</td><td>I</td><td>Y</td><td>i</td><td>y</td><td>ゥ</td><td>ケ</td><td>ノ</td><td>ル</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>A</td><td>EE$_2$ (LF)</td><td>SUB</td><td>*</td><td>:</td><td>J</td><td>Z</td><td>j</td><td>z</td><td>ェ</td><td>コ</td><td>ハ</td><td>レ</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>B</td><td>EE$_3$ (VT)</td><td>ESC</td><td>+</td><td>;</td><td>K</td><td>[</td><td>k</td><td>|</td><td>ォ</td><td>サ</td><td>ヒ</td><td>ロ</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>C</td><td>EE$_4$ (FF)</td><td>IS$_4$ (FS)</td><td>,</td><td><</td><td>L</td><td>¥</td><td>l</td><td>|</td><td>ャ</td><td>シ</td><td>フ</td><td>ワ</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>D</td><td>EE$_5$ (CR)</td><td>IS$_3$ (GS)</td><td>-</td><td>=</td><td>M</td><td>]</td><td>m</td><td>|</td><td>ュ</td><td>ス</td><td>ヘ</td><td>ン</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>E</td><td>SO</td><td>IS$_2$ (RS)</td><td>.</td><td>></td><td>N</td><td>^</td><td>n</td><td>‾</td><td>ョ</td><td>セ</td><td>ホ</td><td>”</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>F</td><td>SI</td><td>IS$_1$ (US)</td><td>/</td><td>?</td><td>O</td><td>_</td><td>o</td><td>DEL</td><td>ッ</td><td>ソ</td><td>マ</td><td>’</td></tr>
</table>

Do not use the undefined sections in the JIS8 code table.

(MEMO)

Index

L

M

N

O

P

R

S

T

U

W

X

Y

Z

Record of changes

The manual No. is written at the bottom of the cover page.

Date	Manual No.	Record of Changes
Dec, 2012	WUME-FP7CPUPGR-01	1st Edition
Jun. 2013	WUME-FP7CPUPGR-02	2nd Edition
Sep. 2013	-	- Revised description of communication instructions in accordance with new serial communication unit and communication cassette models UNITSEL, GPSEND, GPRECV, SEND, RECV, PMSET, PMGET
Apr. 2014	-	-
Jan. 2017	-	- ● Consolidated the instruction descriptions in the following manuals: 1. FP7 CPU Unit Command Reference Manual Supplement (Dec. 2013) INTPG, IRET, DI, EI, IMASK, ICLR, STARTPG, STOPPG, STOD, DTOS, DISF, UNIF, DFLT, DINT, DFIX, DROFF, LOGST, LOGED, SMPL, CDTWT, CDTRD, CWT, CRD, CMKDIR, CRMDIR, CFDEL, CPR, CRD1, CREN, CCOPY, CMV, CFREE, CFREEK, CFLS 2. FP7 CPU Unit Users Manual, Ethernet Expansion Function (Apr. 2014) BCMP, GETSTNO, PGPSEND, IPv4SET, CONSET, ETSTAT, OPEN, CLOSE, PINGREQ, FTPcSV, FTPcSET, FTPcLOG, FTPcREQ, FTPcCTL, HTTPcSV, HTTPcSET, HTTPcREQ, HTTPcCTL, SMTPcSV, SMTPcADD, SMTPcSET, SMTPcREQ, SMTPcLOG, SMTPcCTL 3. FP7 CPU Unit Ver.3.20 Additional Function Manual (Dec. 2014) GPB, MV2, MV3, BSWAP, LCWT, LCRD, MLCLIP, TIMEstr, DEFRBUF, RBUFW, STDDEV, SCOPY, PanaSD 4. FP7 CPU Unit Users Manual, Ethernet Expansion Function (Feb. 2015 Revision) PRINT, PINGREQ, SEND/RECV (MC protocol), SMTPcBDY, SMTPcBRD 5. FP7 CPU Unit Ver.4.10 Additional Function Manual (Aug. 2015) SUMMER, PRINT (added Format%S), EPRINT, ETIMEstr, GPTRNS, NTPcREQ, SMTPcSET (expanded to 4096 characters)

Date	Manual No.	Record of Changes
		6. FP7 CPU Unit Users Manual, EtherNet/IP Communication Function (Sep. 2015) ETSTAT, EIPNDST, EIPSTART, EIPSTOP, EIP_IN, EIP_OT 7. FP7 CPU Unit Ver.4.20 Additional Function Manual (Dec. 2015) TM16, CT16, BKEXT, BKMV16, DIVFP2, BTS, BTR 8. FP7 CPU Unit Ver.4.30 Additional Function Manual (Aug. 2016) LENGTH, NTPcSV, CRMDIRFL ● Changed the configuration of manual to improve searching ● Errors corrected
Feb. 2019	WUME-FP7CPUPGR-03	3rd Edition ● Consolidated the instruction description section of the FP7 Multi-Wire Link Unit User's Manual (Jun. 2017) SEND, RECV, PMGET, PMSET/pPMSET, ERR ● Added SD card access instruction execution result error ● Added list of error codes ● Errors corrected
Sep. 2019	WUME-FP7CPUPGR-04	4th Edition ● Added instructions related to operation history OPHST, OPHED, OPHCLR, OPHSAVE ● Added items to list of system relays ● Added items to list of system data registers ● Added items to list of error codes ● Errors corrected
Mar. 2020	WUME-FP7CPUPGR-05	5th Edition ● Added Ethernet communication instructions EIPMSATT, EIPMBODY, EIPMSEND, CIPMSET, CIPMGET ● Errors corrected
Apr. 2020	WUME-FP7CPUPGR-06	6th Edition ● Added PMSET (pPMSET) instruction specifications ● Added parameter for PLC link noise measures ● Added CONFIG instruction specifications
Oct. 2020	WUME-FP7CPUPGR-07	7th Edition ● Errors corrected
Apr. 2021	WUME-FP7CPUPGR-08	8th Edition ● Added CRD (File Data Read Instruction) specifications

Date	Manual No.	Record of Changes
		● Errors corrected
Nov. 2022	WUME-FP7CPUPGR-12	12th Edition ● Changed product type following FP7 update ● Changed the manual format. ● Corrected errors.
Apr. 2023	WUME-FP7CPUPGR-13	13th Edition ● Added an outline of operation and precautions. TM16, SORT, KP, pPINGREQ, PRINT, EPRINT, TIMEstr, ETIMEstr ● Added error codes. ● Corrected errors.
Dec. 2023	WUME-FP7CPUPGR-14	14th Edition ● Added restrictions for when applying index modification to timer or counter operands. ● Added H8 (The send buffer is being used.) to the execution result code for SEND/RECV instructions. ● Added error codes.
Apr. 2024	WUME-FP7CPUPGR-15	15th Edition ● Change in Corporate name

(MEMO)

(MEMO)

Panasonic Industry Co., Ltd.
1006, Oaza Kadoma, Kadoma-shi, Osaka 571-8506, Japan
https://industry.panasonic.com/

Please visit our website for inquiries and about our sales network.

April, 2024

WUME-FP7CPUPGR-15