Panasonic®

Programmable Controller

FP0R Control Unit
Command Manual

WUME-FP0RPGR-01

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 Manuals

- There are different types of users manuals for the FP0R series. 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
FP0R Control Unit FP0R Expansion I/O Unit	FP0R User's Manual	ARCT1F475
	FP0R Control Unit Command Manual	WUME-FP0RPGR
FP0R Analog Input Unit	FP0R Analog I/O Unit User's Manual	WUME-FP0RAIO
FP0R Analog Output Unit		
FP0R Analog I/O Unit		
Programming Software FPWIN GR	FPWIN GR Introduction Guidance (for a charge)	ARCT1F332
Programming Software FPWIN GR7	FPWIN GR7 Introduction Guidance	WUME-FPWINGR7

(MEMO)

Table of Contents

(MEMO)

1 List of Instruction Words

1.1 List of Basic Instruction Words

■ Sequence basic instructions

Mnemonic	Name	Steps	Reference page:
ST	Begins a logic operation with a Form A (normally open) contact(Note 1)	1 (2)	"P.2-2"
ST/	Begins a logic operation with a Form B (normally closed) contact(Note 1)	1 (2)	"P.2-2"
OT	Outputs the operation result(Note 1)	1 (2)	"P.2-2"
/	Inverts the operation result	1	"P.2-4"
AN	Connects a Form A (normally open) contact serially(Note 2)	1 (2)	"P.2-5"
AN/	Connects a Form B (normally closed) contact serially(Note 2)	1 (2)	"P.2-5"
OR	Connects a Form A (normally open) contact in parallel(Note 2)	1 (2)	"P.2-7"
OR/	Connects a Form B (normally closed) contact in parallel(Note 2)	1 (2)	"P.2-7"
ST↑	Begins a rise contact logic operation	2	"P.2-9"
ST↓	Begins fall contact logic operation	2	"P.2-9"
AN↑	Connects a contact serially when a rise is detected	2	"P.2-9"
AN↓	Connects a contact serially when a fall is detected	2	"P.2-9"
OR↑	Connects a contact in parallel when a rise is detected	2	"P.2-9"
OR↓	Connects a contact in parallel when a fall is detected	2	"P.2-9"
ALT	Alternate out	3	"P.2-11"
ANS	Connects multiple instruction blocks serially	1	"P.2-13"
ORS	Connects multiple instruction blocks in parallel	1	"P.2-15"
PSHS	Stores the operation result	1	"P.2-17"
RDS	Reads the operation result stored by PSHS	1	"P.2-17"
POPS	Reads and clears the operation result stored by PSHS	1	"P.2-17"
DF	Rise detection	1	"P.2-21"
DF/	Fall detection	1	"P.2-21"
DFI	Rise detection (possible on the first scan)	1	"P.2-26"
SET	Turns ON the output and holds it ON(Note 1)	3	"P.2-28"
RST	Turns OFF the output and holds it OFF(Note 1)	3	"P.2-28"
KP	Outputs with set and reset inputs	1	"P.2-31"
NOP	No operation	1	"P.2-33"

(Note 1) Indicates an instruction for which bit index modification is possible.

(Note 2) Numbers in parentheses in the Steps column indicate the number of steps when index modification is performed or when the device number is large (R1120 or higher, T256 or higher, and C256 or higher).

■ Basic function instructions

Mnemonic	Name	Steps	Reference page:
TML	On-delay timer set in 0.001 s units	3 (4)	"P.3-2"
TMR	On-delay timer set in 0.01 s units	3 (4)	"P.3-2"
TMX	On-delay timer set in 0.1 s units	3 (4)	"P.3-2"
TMY	On-delay timer set in 1 s units	4 (5)	"P.3-2"
F137 STMR	On-delay timer set to 0.01 s	5	"P.3-9"
F183 DSTM	32-bit on-delay timer set to 0.01 s	7	"P.3-12"
CT	Down counter	3 (4)	"P.3-16"
F118 UDC	Up/down counter	5	"P.3-23"
SR	Shift register	1	"P.3-26"
F119 LRSR	Left/right shift register	5	"P.3-29"
F182 FILTR	Time constant processing instruction S1, S2, S3, D	9	"P.3-32"

(Note 1) Numbers in parentheses in the Steps column indicate the number of steps when index modification is performed or when the device number is large (R1120 or higher, T256 or higher, and C256 or higher).

■ Control instructions

Mnemonic	Name	Steps	Reference page:
MC	Master control relay	2	"P.4-2"
MCE	Master control relay end	2	"P.4-2"
JP	Jumps to specified label	2	"P.4-7"
LOOP	Jumps to the specified label the number of times specified by D	4	"P.4-11"
LBL	Labels subject to the processing of instructions such as JP and LOOP	1	"P.4-7" "P.4-11"
ED	Main program area end	1	"P.4-15"
CNDE	Conditional program end	1	"P.4-16"
EJECT	Page break when printing	2	"P.4-18"

■ Step ladder instructions

Mnemonic	Name	Steps	Reference page:
SSTP	Process start	3	"P.5-2"
NSTL	Specified process start-up (every scan execution type)	3	"P.5-2"
NSTP	Specified process start-up (differential execution type)	3	"P.5-2"
CSTP	Clears the specified process	3	"P.5-2"
STPE	Step ladder area end	1	"P.5-2"

Mnemonic	Name	Steps	Referen ce page:
SCLR	Clears multiple processes	5	"P.5-17"

■ Subroutine instructions

Mnemonic	Name	Steps	Referen ce page:
CALL	Calls the specified subroutine	2	"P.6-2"
SUB	Subroutine definition	1 (2)	"P.6-2"
RET	Ends the subroutine program and returns to the main program	1	"P.6-2"

(Note 1) Numbers in parentheses in the Steps column indicate the number of steps when index modification is performed or when the device number is large (R1120 or higher, T256 or higher, and C256 or higher).

■ Interrupt instructions

Mnemonic	Name	Steps	Referen ce page:
INT	Interrupt program definition	1	"P.7-2"
IRET	Ends the interrupt program and returns to the main program	1	"P.7-2"
ICTL	Interrupt control specification	5	"P.7-8"

■ Special setting instructions

Mnemonic	Name	Steps	Referen ce page:
SYS1	Communication conditions setting, password setting, interrupt setting, PLC link setting, MEWTOCOL-COM response control, high-speed counter operation mode change	13	"P.8-2" "P.8-6" "P.8-8" "P.8-10" "P.8-13" "P.8-15"
SYS2	System register change instruction	7	"P.8-17"

■ Compare contact instructions

Mnemonic	Name	Steps	Referen ce page:
ST=	Begins a logical operation to compare 16-bit data	5	"P.9-2"
ST<>	Begins a logical operation to compare 16-bit data	5	"P.9-2"
ST>	Begins a logical operation to compare 16-bit data	5	"P.9-2"
ST>=	Begins a logical operation to compare 16-bit data	5	"P.9-2"

Mnemonic	Name	Steps	Reference page:
ST<	Begins a logical operation to compare 16-bit data	5	"P.9-2"
ST<=	Begins a logical operation to compare 16-bit data	5	"P.9-2"
AN=	16-bit data compare serial connection	5	"P.9-4"
AN<>	16-bit data compare serial connection	5	"P.9-4"
AN>	16-bit data compare serial connection	5	"P.9-4"
AN>=	16-bit data compare serial connection	5	"P.9-4"
AN<	16-bit data compare serial connection	5	"P.9-4"
AN<=	16-bit data compare serial connection	5	"P.9-4"
OR=	16-bit data compare parallel connection	5	"P.9-6"
OR<>	16-bit data compare parallel connection	5	"P.9-6"
OR>	16-bit data compare parallel connection	5	"P.9-6"
OR>=	16-bit data compare parallel connection	5	"P.9-6"
OR<	16-bit data compare parallel connection	5	"P.9-6"
OR<=	16-bit data compare parallel connection	5	"P.9-6"
STD=	Begins a logical operation to compare 32-bit data	9	"P.9-8"
STD<>	Begins a logical operation to compare 32-bit data	9	"P.9-8"
STD>	Begins a logical operation to compare 32-bit data	9	"P.9-8"
STD>=	Begins a logical operation to compare 32-bit data	9	"P.9-8"
STD<	Begins a logical operation to compare 32-bit data	9	"P.9-8"
STD<=	Begins a logical operation to compare 32-bit data	9	"P.9-8"
AND=	32-bit data compare serial connection	9	"P.9-10"
AND<>	32-bit data compare serial connection	9	"P.9-10"
AND>	32-bit data compare serial connection	9	"P.9-10"
AND>=	32-bit data compare serial connection	9	"P.9-10"
AND<	32-bit data compare serial connection	9	"P.9-10"
AND<=	32-bit data compare serial connection	9	"P.9-10"
ORD=	32-bit data compare parallel connection	9	"P.9-12"
ORD<>	32-bit data compare parallel connection	9	"P.9-12"
ORD>	32-bit data compare parallel connection	9	"P.9-12"
ORD>=	32-bit data compare parallel connection	9	"P.9-12"
ORD<	32-bit data compare parallel connection	9	"P.9-12"
ORD<=	32-bit data compare parallel connection	9	"P.9-12"

■ Compare contact instructions

Mnemonic	Name	Steps	Reference page:
STF=	Begins a logical operation to compare single-precision floating point data	10	"P.9-14"
STF<>	Begins a logical operation to compare single-precision floating point data	10	"P.9-14"
STF>	Begins a logical operation to compare single-precision floating point data	10	"P.9-14"
STF>=	Begins a logical operation to compare single-precision floating point data	10	"P.9-14"
STF<	Begins a logical operation to compare single-precision floating point data	10	"P.9-14"
STF<=	Begins a logical operation to compare single-precision floating point data	10	"P.9-14"
ANF=	Single-precision floating point data compare serial connection	10	"P.9-16"
ANF<>	Single-precision floating point data compare serial connection	10	"P.9-16"
ANF>	Single-precision floating point data compare serial connection	10	"P.9-16"
ANF>=	Single-precision floating point data compare serial connection	10	"P.9-16"
ANF<	Single-precision floating point data compare serial connection	10	"P.9-16"
ANF<=	Single-precision floating point data compare serial connection	10	"P.9-16"
ORF=	Single-precision floating point data compare parallel connection	10	"P.9-18"
ORF<>	Single-precision floating point data compare parallel connection	10	"P.9-18"
ORF>	Single-precision floating point data compare parallel connection	10	"P.9-18"
ORF>=	Single-precision floating point data compare parallel connection	10	"P.9-18"
ORF<	Single-precision floating point data compare parallel connection	10	"P.9-18"
ORF<=	Single-precision floating point data compare parallel connection	10	"P.9-18"

1.2 List of High-level Instructions

■ Transfer instructions

Fun no.	Mnemonic	Operands	Name	Steps	Refere nce page:
F0	MV	S, D	16-bit data transfer	5	"P. 10-2"
F1	DMV	S, D	32-bit data transfer	7	"P. 10-4"
F2	MV/	S, D	16-bit data invert and transfer	5	"P. 10-6"
F3	DMV/	S, D	32-bit data invert and transfer	7	"P. 10-8"
F5	BTM	S, n, D	Bit data transfer	7	"P. 10-10"
F6	DGT	S, n, D	Digit data transfer	7	"P. 10-15"
F7	MV2	S1, S2, D	Two 16-bit data transfer to single area	7	"P. 10-19"
F8	DMV2	S1, S2, D	Two 32-bit data transfer to single area	11	"P. 10-21"
F10	BKMV	S1, S2, D	Data block transfer	7	"P. 10-23"
F11	COPY	S, D1, D2	16-bit data block copy	7	"P. 10-26"
F12	ICRD	S1, S2, D	F-ROM read	11	"P. 10-28"
P13	PICWT	S1, S2, D	Write to F-ROM	11	"P. 10-30"
F15	XCH	D1, D2	16-bit data exchange	5	"P. 10-32"
F16	DXCH	D1, D2	32-bit data exchange	5	"P. 10-34"
F17	SWAP	D	High byte and low byte exchange	3	"P. 10-36"
F18	BXCH	D1, D2, D3	Data block exchange	7	"P. 10-38"
F190	MV3	S1, S2, S3, D	Three 16-bit data transfer to single area	10	"P. 10-40"
F191	DMV3	S1, S2, S3, D	Three 32-bit data transfer to single area	16	"P. 10-42"

■ Binary arithmetic instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F20	+	S, D	16-bit data addition [D+S=D]	5	"P. 11-2"
F21	D+	S, D	32-bit data addition [D+S=D]	7	"P. 11-4"
F22	+	S1, S2, D	16-bit data addition [S1+S2=D]	7	"P. 11-6"
F23	D+	S1, S2, D	32-bit data addition [S1+S2=D]	11	"P. 11-8"
F25	-	S, D	16-bit data subtraction [D-S=D]	5	"P. 11-10"
F26	D-	S, D	32-bit data subtraction [D-S=D]	7	"P. 11-13"
F27	-	S1, S2, D	16-bit data subtraction [S1-S2=D]	7	"P. 11-15"
F28	D-	S1, S2, D	32-bit data subtraction [S1-S2=D]	11	"P. 11-18"
F30	*	S1, S2, D	16-bit data multiplication [S1*S2=D+1, D]	7	"P. 11-20"
F31	D*	S1, S2, D	32-bit data multiplication [S1*S2=D+3,D+2,D +1,D]	11	"P. 11-22"
F32	%	S1, S2, D	16-bit data division [S1/S2=D]	7	"P. 11-24"
F33	D%	S1, S2, D	32-bit data division [S1/S2=D+1, D]	11	"P. 11-26"
F34	*W	S1, S2, D	16-bit data multiplication [S1*S2=D]	7	"P. 11-28"
F35	+1	D	16-bit data increment	3	"P. 11-30"
F36	D+1	D	32-bit data increment	3	"P. 11-32"
F37	-1	D	16-bit data decrement	3	"P. 11-34"
F38	D-1	D	32-bit data decrement	3	"P. 11-36"
F39	D*D	S1, S2, D	32-bit data multiplication [S1 × S2 = D+1, D]	11	"P. 11-38"

■ BCD data arithmetic instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F40	B+	S, D	4-digit BCD data addition [D+S=D]	5	"P. 12-2"

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F41	DB+	S, D	8-digit BCD data addition [D+S=D]	7	"P. 12-4"
F42	B+	S1, S2, D	4-digit BCD data addition [S1+S2=D]	7	"P. 12-6"
F43	DB+	S1, S2, D	8-digit BCD data addition [S1+S2=D]	11	"P. 12-8"
F45	B-	S, D	4-digit BCD data subtraction [D-S=D]	5	"P. 12-10"
F46	DB-	S, D	8-digit BCD data subtraction [D-S=D]	7	"P. 12-12"
F47	B-	S1, S2, D	4-digit BCD data subtraction [S1-S2=D]	7	"P. 12-14"
F48	DB-	S1, S2, D	8-digit BCD data subtraction [S1-S2=D]	11	"P. 12-16"
F50	B*	S1, S2, D	4-digit BCD data multiplication [S1*S2=D+1, D]	7	"P. 12-18"
F51	DB*	S1, S2, D	8-digit BCD data multiplication [S1*S2=D+3,D +2,D+1,D]	11	"P. 12-20"
F52	B%	S1, S2, D	4-digit BCD data division [S1/S2=D]	7	"P. 12-22"
F53	DB%	S1, S2, D	8-digit BCD data division [S1/S2=D+1, D]	11	"P. 12-24"
F55	B+1	D	4-digit BCD data increment	3	"P. 12-26"
F56	DB+1	D	8-digit BCD data increment	3	"P. 12-28"
F57	B-1	D	4-digit BCD data decrement	3	"P. 12-30"
F58	DB-1	D	8-digit BCD data decrement	3	"P. 12-32"

■ Data comparison instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F60	CMP	S1, S2	16-bit data comparison	5	"P. 13-2"
F61	DCMP	S1, S2	32-bit data comparison	9	"P. 13-8"
F62	WIN	S1, S2, S3	16-bit data band comparison	7	"P. 13-12"
F63	DWIN	S1, S2, S3	32-bit data band comparison	13	"P. 13-14"
F64	BCMP	S1, S2, S3	Block data comparison	7	"P. 13-16"

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F373	DTR	S, D	16-bit data revision detection	6	"P. 13-19"
F374	DDTR	S, D	32-bit data revision detection	6	"P. 13-21"

■ Boolean instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F65	WAN	S1, S2, D	16-bit data AND	7	"P. 14-2"
F66	WOR	S1, S2, D	16-bit data OR	7	"P. 14-4"
F67	XOR	S1, S2, D	16-bit data exclusive OR	7	"P. 14-6"
F68	XNR	S1, S2, D	16-bit data exclusive NOR	7	"P. 14-8"
F69	WUNI	S1, S2, S3, D	[(S1 AND S3) OR (S2 AND S3) = D] (16-bit)	9	"P. 14-10"
F215	DAND	S1, S2, D	32-bit data AND	12	"P. 14-12"
F216	DOR	S1, S2, D	32-bit data OR	12	"P. 14-14"
F217	DXOR	S1, S2, D	32-bit data exclusive OR	12	"P. 14-16"
F218	DXNR	S1, S2, D	32-bit data exclusive NOR	12	"P. 14-18"
F219	DUNI	S1, S2, S3, D	[(S1 AND S3) OR (S2 AND S3) = D] (32-bit)	16	"P. 14-20"

■ Data conversion instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F70	BCC	S1, S2, S3, D	Block check code (ADD.SUB,XOR,CRC)	9	"P. 15-3"
F71	HEXA	S1, S2, D	Hexadecimal data to ASCII code conversion	7	"P. 15-7"
F72	AHEX	S1, S2, D	ASCII code to hexadecimal data conversion	7	"P. 15-10"
F73	BCDA	S1, S2, D	BCD data to ASCII code conversion	7	"P. 15-14"
F74	ABCD	S1, S2, D	ASCII code to BCD data conversion	7	"P. 15-18"

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F75	BINA	S1, S2, D	16-bit binary data to ASCII code conversion	7	"P. 15-22"
F76	ABIN	S1, S2, D	ASCII code to 16-bit binary data conversion	7	"P. 15-25"
F77	DBIA	S1, S2, D	32-bit binary data to ASCII code conversion	11	"P. 15-29"
F78	DABI	S1, S2, D	ASCII code to 32-bit binary data conversion	11	"P. 15-32"
F80	BCD	S, D	16-bit binary data to BCD data conversion	5	"P. 15-36"
F81	BIN	S, D	BCD data to 16-bit binary data conversion	5	"P. 15-38"
F82	DBCD	S, D	32-bit binary data to BCD data conversion	7	"P. 15-40"
F83	DBIN	S, D	BCD data to 32-bit binary data conversion	7	"P. 15-41"
F84	INV	D	16-bit data inversion	3	"P. 15-42"
F85	NEG	D	16-bit data sign inversion	3	"P. 15-43"
F86	DNEG	D	32-bit data sign inversion	3	"P. 15-44"
F87	ABS	D	16-bit data absolute value	3	"P. 15-46"
F88	DABS	D	32-bit data absolute value	3	"P. 15-47"
F89	EXT	D	Sign extension	3	"P. 15-48"
F90	DECO	S, n, D	Decode	7	"P. 15-50"
F91	SEGT	S, D	7-segment	5	"P. 15-53"
F92	ENCO	S, n, D	Encode	7	"P. 15-55"
F93	UNIT	S, n, D	Digit combine	7	"P. 15-58"
F94	DIST	S, n, D	Digit distribute	7	"P. 15-60"
F96	SRC	S1, S2, S3	16-bit data search	7	"P. 15-62"
F97	DSRC	S1, S2, S3, S4	32-bit data search	9	"P. 15-64"
F230	TMSEC	S, D	Time to second conversion	6	"P. 15-66"

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F231	SECTM	S, D	Second to time conversion	6	"P. 15-69"
F235	GRY	S, D	16-bit data to gray code conversion	6	"P. 15-72"
F236	DGRY	S, D	32-bit data to gray code conversion	8	"P. 15-73"
F237	GBIN	S, D	Gray code to 16-bit data conversion	6	"P. 15-74"
F238	DGBIN	S, D	Gray code to 32-bit data conversion	8	"P. 15-75"
F240	COLM	S1, S2, D	Bit line to bit column conversion	8	"P. 15-77"
F241	LINE	S1, S2, D	Bit column to bit line conversion	8	"P. 15-79"

■ Data shift instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F100	SHR	D, n	16-bit data right shift	5	"P. 16-2"
F101	SHL	D, n	16-bit data left shift	5	"P. 16-4"
F102	DSHR	D, n	32-bit data right shift	5	"P. 16-6"
F103	DSHL	D, n	32-bit data left shift	5	"P. 16-8"
F105	BSR	D	16-bit data 1 digit right shift	3	"P. 16-10"
F106	BSL	D	16-bit data 1 digit left shift	3	"P. 16-12"
F108	BITR	D1, D2, n	Block area bitwise right shift	7	"P. 16-14"
F109	BITL	D1, D2, n	Block area bitwise left shift	7	"P. 16-16"
F110	WSHR	D1, D2	Block area 1 word right shift	5	"P. 16-18"
F111	WSHL	D1, D2	Block area 1 word left shift	5	"P. 16-20"
F112	WBSR	D1, D2	Block area 1 digit right shift	5	"P. 16-22"
F113	WBSL	D1, D2	Block area 1 digit left shift	5	"P. 16-24"

Data rotation instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F120	ROR	D, n	16-bit data right rotation	5	"P. 17-2"
F121	ROL	D, n	16-bit data left rotation	5	"P. 17-4"
F122	RCR	D, n	16-bit data right rotation with carry	5	"P. 17-6"
F123	RCL	D, n	16-bit data left rotation with carry	5	"P. 17-8"
F125	DROR	D, n	32-bit data right rotation	5	"P. 17-10"
F126	DROL	D, n	32-bit data left rotation	5	"P. 17-12"
F127	DRCR	D, n	32-bit data right rotation with carry	5	"P. 17-14"
F128	DRCL	D, n	32-bit data left rotation with carry	5	"P. 17-16"

Data buffer instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F98	CMPR	D1, D2, D3	Compress shift read	7	"P. 18-2"
F99	CMPW	S1, D, S2	Compress shift write	7	"P. 18-6"
F115	FIFT	n, D	FIFO buffer definition	5	"P. 18-11"
F116	FIFR	S, D	FIFO data read	5	"P. 18-14"
F117	FIFW	S, D	FIFO data write	5	"P. 18-18"

Bit manipulation instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F130	BTS	D, n	Specified bit set	5	"P. 19-2"
F131	BTR	D, n	Specified bit reset	5	"P. 19-4"
F132	BTI	D, n	Specified bit inversion	5	"P. 19-6"

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F133	BTT	S, n	Specified bit test	5	"P. 19-8"
F135	BCU	S, D	Count ON bits in 16-bit data	5	"P. 19-10"
F136	DBCU	S, D	Count ON bits in 32-bit data	7	"P. 19-12"

■ Special instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F138	HMSS	S, D	Hour, minute, second data to second data conversion	5	"P. 20-2"
F139	SHMS	S, D	Second data to hour, minute, second data conversion	5	"P. 20-4"
F140	STC		Cy flag set	1	"P. 20-6"
F141	CLC		Cy flag clear	1	"P. 20-7"
F143	IORF	D1, D2	Partial I/O refresh	5	"P. 20-8"
F147	PR	S, D	Print out	5	"P. 20-9"
F148	ERR	n	Self-diagnostic error code set	3	"P. 20-14"
F149	MSG	S	Character send to programming tool	13	"P. 20-16"
F157	CADD	S1, S2, D	Calendar data addition	9	"P. 20-17"
F158	CSUB	S1, S2, D	Calendar data subtraction	9	"P. 20-20"
F160	DSQR	S, D	32-bit data square root	7	"P. 20-25"

■ Serial communication instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page
F145	SEND	S1, S2, D, N	Data send instruction [MEWTOCOL master]	9	"P.21-4"
F146	RECV	S1, S2, N, D	Data receive instruction [MEWTOCOL master]	9	"P.21-7"
F145	SEND	S1, S2, D, N	Data send instruction [MODBUS master: Function code specification]	9	"P.21-23"
F146	RECV	S1, S2, N, D	Data receive instruction [MODBUS master: Function code specification]	9	"P.21-28"

Fun no.	Mnemonic	Operands	Name	Steps	Reference page
F145	SEND	S1, S2, D, N	Data send instruction [MODBUS master: No function code specification]	9	"P.21-10"
F146	RECV	S1, S2, N, D	Data receive instruction [MODBUS master: No function code specification]	9	"P.21-16"
F159	MTRN	S, n, D	General-purpose communication instructions	7	"P.21-33"

■ Sampling trace instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page
F155	SMPL		Sample set data	1	"P.22-3"
F156	STRG		Sampling stop trigger	1	"P.22-4"

■ High-speed counter/PWM output instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page
F0	MV	S, DT90052	High-speed counter control	5	"P.23-2"
F0	MV	S, DT90052	Pulse output control	5	"P.23-4"
F1	DMV	S, DT90300	High-speed counter elapsed value read	7	"P.23-7"
F1	DMV	DT90300, D	High-speed counter elapsed value write	7	"P.23-7"
F165	CAM0	S	Cam control	3	"P.23-10"
F166	HC1S	n, S, D	Target Value Match ON (with Channel Specification)	11	"P.23-19"
F166	HC1S	n, S, D	Target value match ON (High-speed counter control)	11	"P.23-22"
F166	HC1S	n, S, D	Target value match ON (Pulse output control)	11	"P.23-25"
F167	HC1R	n, S, D	Target Value Match OFF (with Channel Specification)	11	"P.23-28"
F167	HC1R	n, S, D	Target value match OFF (High-speed counter control)	11	"P.23-31"
F167	HC1R	n, S, D	Target value match OFF (Pulse output control)	11	"P.23-34"
F173	PWMH	S, n	PWM Output (with Channel Specification)	5	"P.23-37"

■ Character string instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F95	ASC	M,D	Character constant to ASCII code conversion	15	"P. 24-2"
F250	BTOA	S1, S2, S3, D	Multiple binary data to ASCII data string conversion	12	"P. 24-5"
F251	ATOB	S1, S2, S3, D	Multiple ASCII data strings to binary data conversion	12	"P. 24-11"

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F252	ACHK	S1, S2, S3	ASCII code check of multiple ASCII data strings	10	"P. 24-18"
F257	SCMP	S1, S2, D	Character string comparison	10	"P. 24-22"
F258	SADD	S1, S2, D	Character string addition	12	"P. 24-24"
F259	LEN	S, D	Character string length	6	"P. 24-26"
F260	SSRC	S1, S2, D	Character string search	10	"P. 24-28"
F261	RIGHT	S1, S2, D	Right retrieve from character string	8	"P. 24-30"
F262	LEFT	S1, S2, D	Left retrieve from character string	8	"P. 24-32"
F263	MIDR	S1, S2, S3, D	Read from any position in character string	10	"P. 24-34"
F264	MIDW	S1, S2, S3, D	Write to any position in character string	12	"P. 24-36"
F265	SREP	S, D, P, n	Replace character string	12	"P. 24-38"

■ Data manipulation instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F270	MAX	S1, S2, D	Search maximum value from 16-bit data block	8	"P. 25-2"
F271	DMAX	S1, S2, D	Search maximum value from 32-bit data block	8	"P. 25-4"
F272	MIN	S1, S2, D	Search minimum value from 16-bit data block	8	"P. 25-6"
F273	DMIN	S1, S2, D	Search minimum value from 32-bit data block	8	"P. 25-8"
F275	MEAN	S1, S2, D	16-bit data sum and average	8	"P. 25-10"
F276	DMEAN	S1, S2, D	32-bit data sum and average	8	"P. 25-12"
F277	SORT	S1, S2, S3	16-bit data block sort	8	"P. 25-14"
F278	DSORT	S1, S2, S3	32-bit data block sort	8	"P. 25-16"
F282	SCAL	S1, S2, D	16-bit data linearization	8	"P. 25-18"
F283	DSCAL	S1, S2, D	32-bit data linearization	10	"P. 25-21"

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F284	RAMP	S1, S2, S3, D	16-bit data ramp output	10	"P. 25-24"
F285	LIMT	S1, S2, S3, D	16-bit data upper and lower limit control	10	"P. 25-26"
F286	DLIMT	S1, S2, S3, D	32-bit data upper and lower limit control	16	"P. 25-28"
F287	BAND	S1, S2, S3, D	16-bit data deadband control	10	"P. 25-30"
F288	DBAND	S1, S2, S3, D	32-bit data deadband control	16	"P. 25-32"
F289	ZONE	S1, S2, S3, D	16-bit data zone control	10	"P. 25-34"
F290	DZONE	S1, S2, S3, D	32-bit data zone control	16	"P. 25-36"

■ **Floating-point instructions**

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F309	FMV	S, D	Floating point data move	8	"P. 26-3"
F310	F+	S1, S2, D	Floating point data addition	14	"P. 26-5"
F311	F-	S1, S2, D	Floating point data subtraction	14	"P. 26-7"
F312	F*	S1, S2, D	Floating point data multiplication	14	"P. 26-9"
F313	F%	S1, S2, D	Floating point data division	14	"P. 26-11"
F314	SIN	S, D	Floating point data sine operation	10	"P. 26-13"
F315	COS	S, D	Floating point data cosine operation	10	"P. 26-15"
F316	TAN	S, D	Floating point data tangent operation	10	"P. 26-17"
F317	ASIN	S, D	Floating point data arcsine operation	10	"P. 26-19"
F318	ACOS	S, D	Floating point data arccosine operation	10	"P. 26-21"
F319	ATAN	S, D	Floating point data arctangent operation	10	"P. 26-23"
F320	LN	S, D	Floating point data natural logarithmic operation	10	"P. 26-25"
F321	EXP	S, D	Floating point data exponent operation	10	"P. 26-27"

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F322	LOG	S, D	Floating point data logarithm operation	10	"P. 26-29"
F323	PWR	S1, S2, D	Floating point data power operation	14	"P. 26-31"
F324	FSQR	S, D	Floating point data square root operation	10	"P. 26-33"
F325	FLT	S, D	16-bit integer to floating point data conversion	6	"P. 26-35"
F326	DFLT	S, D	32-bit integer to floating point data conversion	8	"P. 26-36"
F327	INT	S, D	Floating point data to 16-bit integer conversion (largest integer not exceeding the floating point data)	8	"P. 26-38"
F328	DINT	S, D	Floating point data to 32-bit integer conversion (largest integer not exceeding the floating-point data)	8	"P. 26-40"
F329	FIX	S, D	Floating point data to 16-bit integer conversion (round-down)	8	"P. 26-42"
F330	DFIX	S, D	Floating point data to 32-bit integer conversion (round-down)	8	"P. 26-44"
F331	ROFF	S, D	Floating point data to 16-bit integer conversion (round-off)	8	"P. 26-46"
F332	DROFF	S, D	Floating point data to 32-bit integer conversion (round-off)	8	"P. 26-48"
F333	FINT	S, D	Floating point data round-down	8	"P. 26-50"
F334	FRINT	S, D	Floating point data round-off	8	"P. 26-52"
F335	F+/-	S, D	Floating point data sign conversion	8	"P. 26-54"
F336	FABS	S, D	Floating point data absolute value conversion	8	"P. 26-56"
F337	RAD	S, D	Degree to radian conversion	8	"P. 26-58"
F338	DEG	S, D	Radian to degree conversion	8	"P. 26-60"

■ Real number data processing instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F345	FCMP	S1, S2	Floating point data comparison	10	"P. 27-2"
F346	FWIN	S1, S2, S3	Floating point data band comparison	14	"P. 27-4"

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F347	FLIMT	S1, S2, S3, D	Floating point data lower limit and upper limit control	18	"P. 27-6"
F348	FBAND	S1, S2, S3, D	Floating point data deadband control	18	"P. 27-8"
F349	FZONE	S1, S2, S3, D	Floating point data zone control	18	"P. 27-10"
F354	FSCAL	S1, S2, D	Real number data scaling	12	"P. 27-12"

■ Process control instructions

Fun no.	Mnemonic	Operands	Name	Steps	Reference page:
F355	PID	S	PID operation	4	"P. 28-2"
F356	EZPID	S1, S2, S3, S4	PID operation: PWM output possible	10	"P. 28-9"

■ Positioning Control Instructions

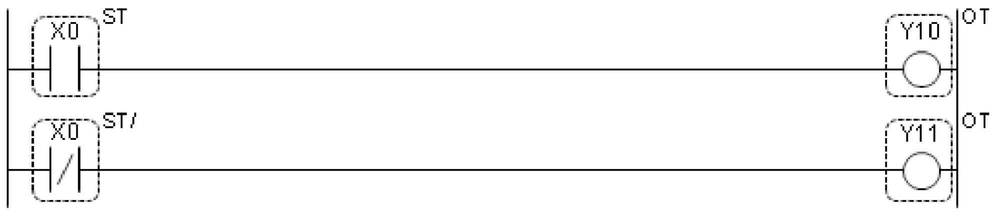
Fun no.	Mnemonic	Operands	Name	Steps	Reference page
F1	DMV	S, DT90348	Pulse output elapsed value read	7	"P.29-2"
F1	DMV	DT90348, D	Pulse output elapsed value write	7	"P.29-2"
F171	SPDH	S, n	Pulse Output (Trapezoidal Control)	5	"P.29-5"
F171	SPDH	S, n	Pulse Output (JOG Positioning Type 0)	5	"P.29-18"
F171	SPDH	S, n	Pulse Output (JOG Positioning Type 1)	5	"P.29-25"
F172	PLSH	S, n	Pulse Output (JOG Operation Type 0/1)	5	"P.29-32"
F174	SP0H	S, n	Pulse output (Selectable data table control operation)	5	"P.29-38"
F175	SPSH	S, n	Pulse output (linear interpolation)	5	"P.29-43"
F177	HOME	S, n	Pulse Output (Home Return)	5	"P.29-49"
F178	PLSM	S1, S2, D	Input pulse measurement	5	"P.29-54"

(MEMO)

2 Sequence Basic Instructions

2.1 ST, ST/ and OT (Start, Start Not and Out)

■ Instruction format



■ Instruction list

Instruction	Description
ST	Input contact starting logical operation as Form A (normally open)
ST/	Input contact starting logical operation as Form B (normally closed)
OT	Coil that outputs logical operation

■ Devices that can be specified (indicated by •)

Operands	X	Y	R	T	C	L	Index modifier
ST	•	•	•	•	•	•	•
ST/	•	•	•	•	•	•	•
OT		•	•			•	•

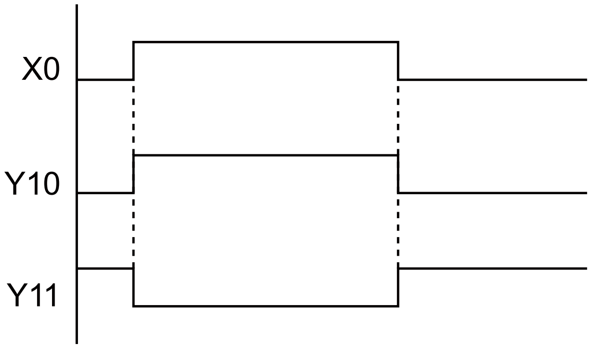
■ Outline of operation

Instruction	Operation
ST	Handles input contact as Form A (normally open) and begins a logical operation.
ST/	Handles input contact as Form B (normally closed) and begins a logical operation.
OT	Outputs operation results to the specified coil.

■ Operation example

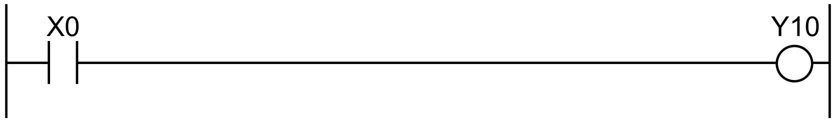
Operation of instruction format description program

- Execution results are output to Y10 when X0 is ON, and to Y11 when X0 is OFF.

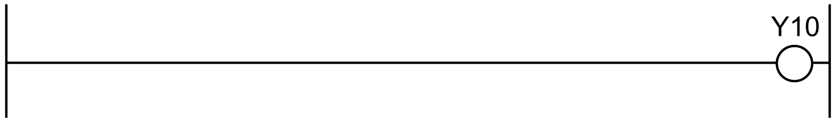


■ **Precautions for programming**

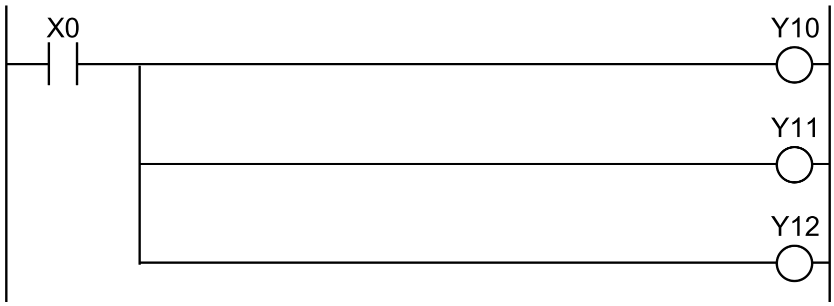
- **ST** instructions begin from the bus bar. (This is the same for **ST/** instructions)



- **OT** instructions cannot begin directly from the bus bar.



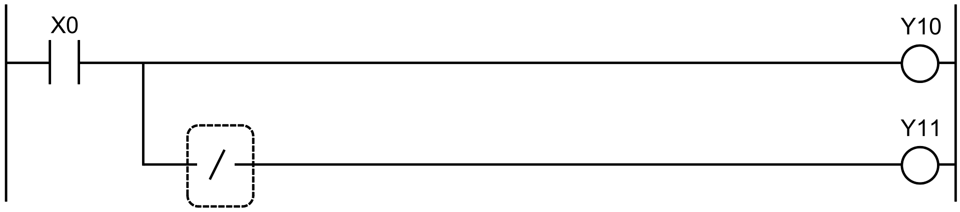
- **OT** instructions can be used consecutively.



- When an external switch is Form B (normally closed), such as an emergency stop switch, take care to use**ST**instructions in programming.

2.2 / (Not)

■ Instruction format



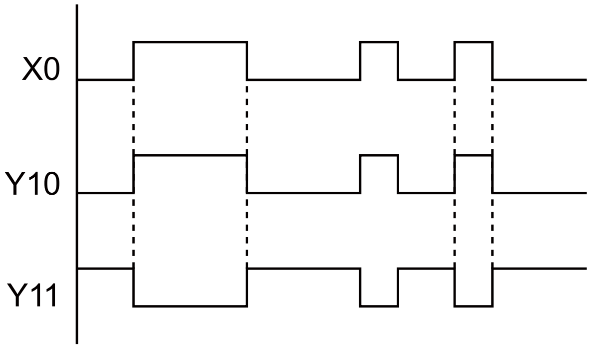
■ Outline of operation

- The **NOT** instruction inverts the operation result up to immediately before this instruction.

■ Operation example

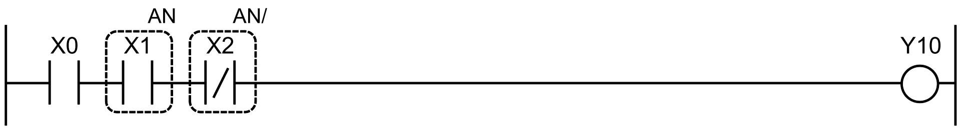
Operation of instruction format description program

- When X0 turns ON, Y10 turns ON and Y11 turns OFF.
- When X0 turns OFF, Y10 turns OFF and Y11 turns ON.



2.3 AN, AN/ (AND, AND Not)

■ Instruction format



■ Instruction list

Instruction	Description
AN	Form A (normally open) contacts connected in series
AN/	Form B (normally closed) contacts connected in series

■ Devices that can be specified (indicated by ●)

Operands	X	Y	R	T	C	L	Index modifier
AN	●	●	●	●	●	●	●
AN/	●	●	●	●	●	●	●

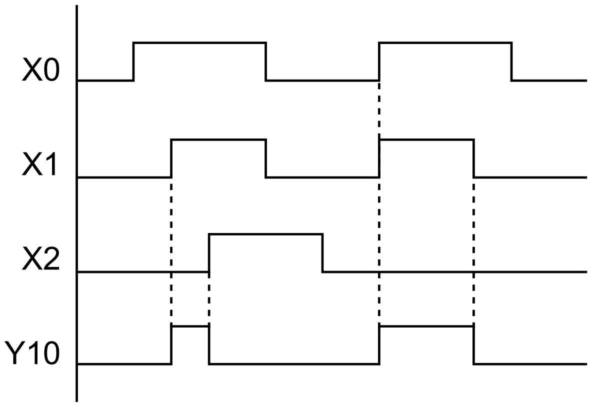
■ Outline of operation

- A logical conjunction is executed with the immediately preceding serially connected operation result.

■ Operation example

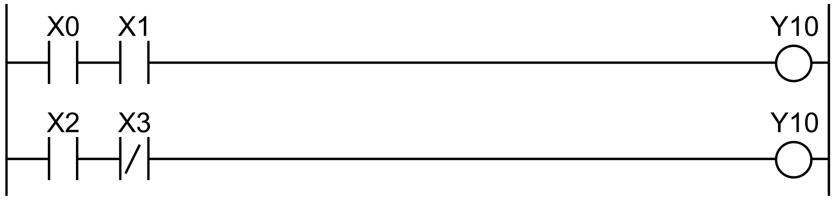
Operation of instruction format description program

- When X0 and X1 turn ON and X2 turns OFF, the result is output to Y10.

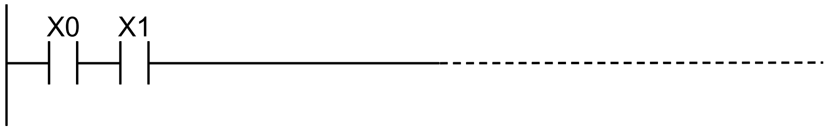


■ Precautions for programming

- Use the **AN** instruction when Form A (normally open) contacts are serially connected.
- Use the **AN/** instruction when Form B (normally closed) contacts are serially connected.



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



2.4 OR, OR/ (OR, OR Not)

■ Instruction format

■ Instruction list

Instruction	Description
OR	Form A (normally open) contact connected in parallel
OR/	Form B (normally closed) contact connected in parallel

■ Devices that can be specified (indicated by ●)

Operands	X	Y	R	T	C	L	Index modifier
OR	●	●	●	●	●	●	●
OR/	●	●	●	●	●	●	●

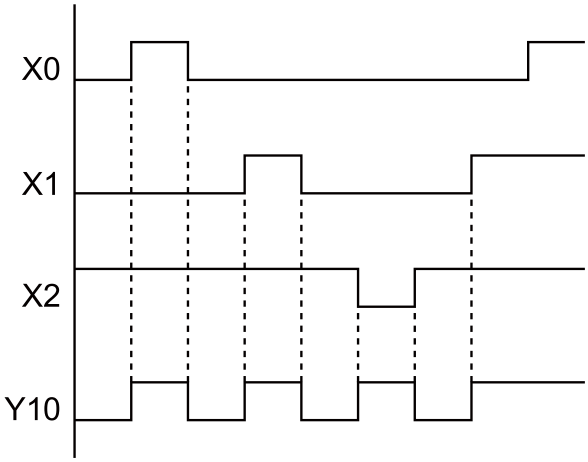
■ Outline of operation

- A logical disjunction is executed with the immediately preceding operation result of the contact connected in parallel.

■ Operation example

Operation of instruction format description program

If any of the conditions of X0 ON, X1 ON, or X2 OFF is satisfied, the result is output to Y10.

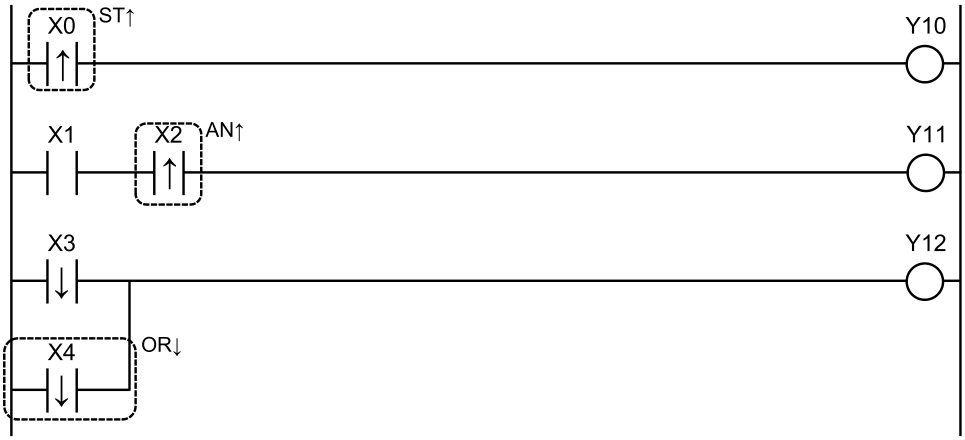


■ **Precautions for programming**

- Use the **OR** instruction when Form A (normally open) contacts are connected in parallel.
- Use the **OR/** instruction when Form B (normally closed) contacts are connected in parallel.
- The **OR** instruction, like the **ST** instruction, starts from the bus bar.
- The **OR** and **OR/** instructions can be used consecutively.

2.5 ST↑, ST↓, AN↑, AN↓, OR↑, OR↓ (Rise Detection, Fall Detection)

■ Instruction format



■ Instruction list

Instruction	Description
ST↑, ST↓	Input contact that starts a logical operation at the rise or fall of a signal
AN↑, AN↓	Contacts connected in series at the rise or fall of a signal
OR↑, OR↓	Contacts connected in parallel at the rise or fall of a signal

■ Devices that can be specified (indicated by ●)

Operands	X	Y	R	T	C	L	Index modifier
ST↑, ST↓	●	●	●	●	●	●	
AN↑, AN↓	●	●	●	●	●	●	
OR↑, OR↓	●	●	●	●	●	●	

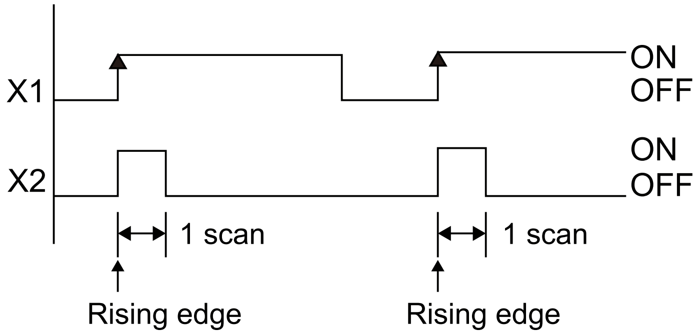
■ Outline of operation

Instruction	Operation
ST↑, AN↑, OR↑	Conduction takes place for 1 scan only following the change of a signal from the OFF to ON state (rise).
ST↓, AN↓, OR↓	Conduction takes place for 1 scan only following the change of a signal from the ON to OFF state (fall).

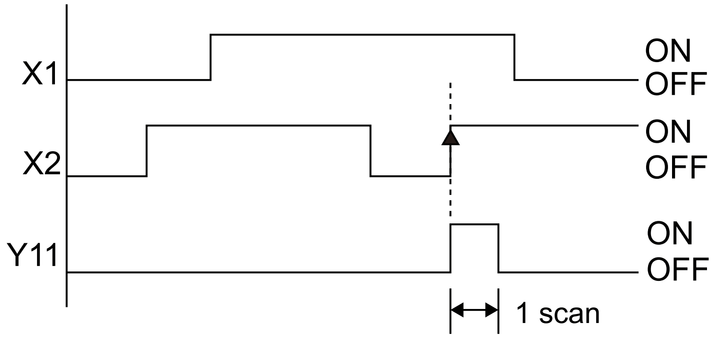
■ Operation example

Operation of instruction format description program

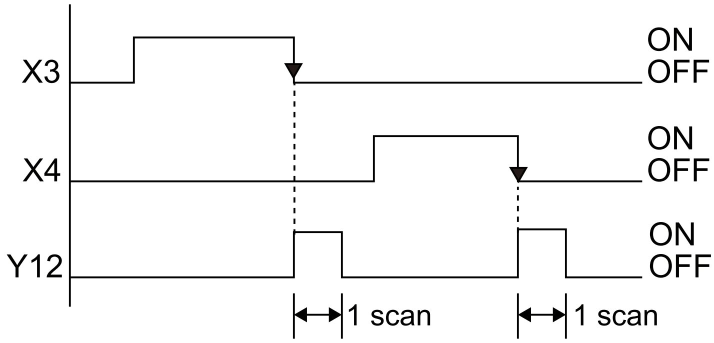
1. When X0 changes from OFF to ON (rise), only 1 scan is output to Y10.



2. Output to Y11 takes place for 1 scan only following the change of X2 from the OFF to ON state (rise) when X1 is ON.

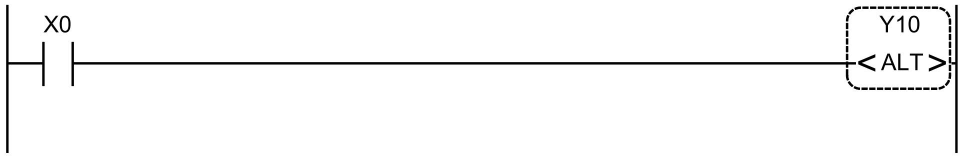


3. Output to Y12 takes place for 1 scan only following the change of X3 or X4 from the ON to OFF state (fall).



2.6 ALT (Alternate Out)

■ Instruction format



■ Instruction list

Instruction	Description
ALT	Coil that controls flip-flops

■ Devices that can be specified (indicated by ●)

Operands	X	Y	R	T	C	L	Index modifier
ALT		●	●			●	

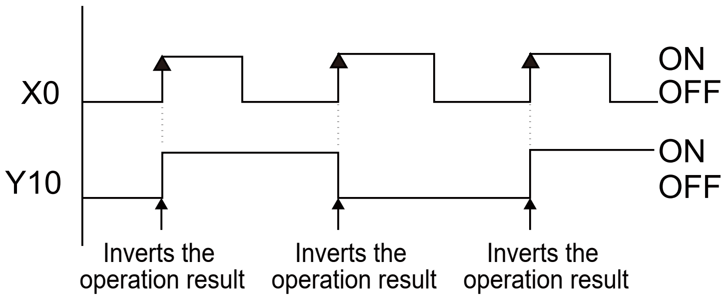
■ Outline of operation

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

■ Operation example

Operation of instruction format description program

Each time X0 changes from OFF to ON (rises), the output Y10 ON/OFF status is inverted.



■ Precautions for programming

The **ALT** instruction detects input OFF to ON rise and inverts the output.

- While the input continues to be ON, it is inverted only during rise. After that it is not inverted.
- When switching to RUN or when powering on in "RUN mode" , if input is ON from the beginning, inversion is not carried out for the first scan.

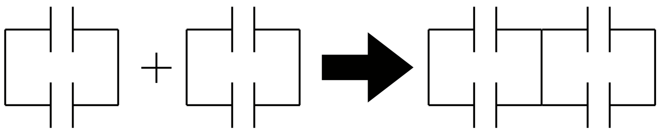
- Be aware that, if used with instructions that change the order of execution such as the MC to MCE instructions or the JP to LBL instructions (see 1 to 6 below), the operation of instructions may change depending on the timing of instruction execution and input.
  1. MC to MCE instructions
  2. JP to LBL instructions
  3. LOOP to LBL instructions
  4. CNDE instruction
  5. Step ladder instructions
  6. Subroutine instructions

2.7 ANS (And Stack)

■ Instruction format

■ Outline of operation

- Blocks that were connected in parallel are connected in series.



Blocks stack in series

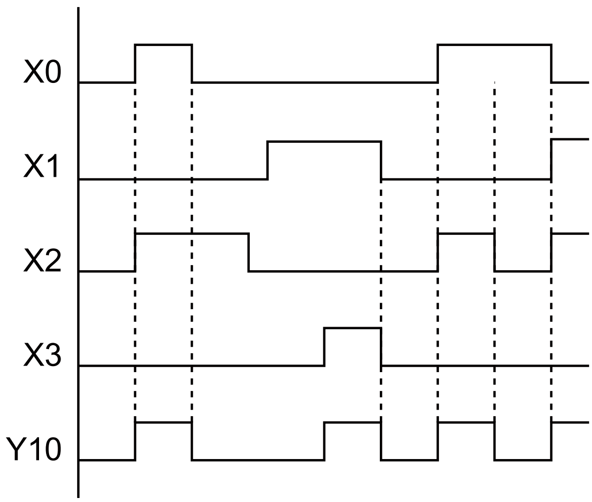
- The start of each block begins with an **ST** instruction.

■ Operation example

Operation of instruction format description program

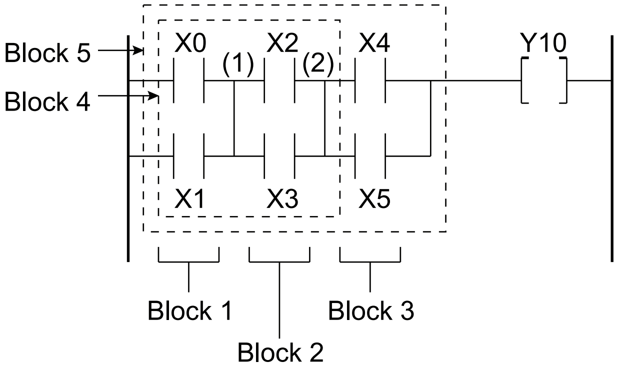
When X0 or X1 are ON, and X2 or X3 are ON, they are output to Y10.

$$\underbrace{\text{(X0 OR X1)}}_{\text{Block 1}}\ \textbf{AND}\ \underbrace{\text{(X2 OR X3)}}_{\text{Block 2}}\rightarrow \text{Y10}$$



■ When blocks are consecutive

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



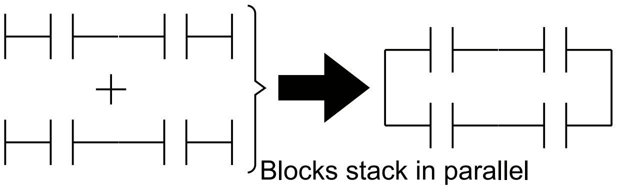
Block 5	Block 4	Block 1	ST X		0
			OR X		1
		Block 2	ST X		2
			OR X		3
			ANS	·············	(1)
	Block 3		ST X		4
			OR X		5
			ANS	·············	(2)
			OT	Y	10

2.8 ORS (OR Stack)

■ Instruction format

■ Outline of operation

- Serially connected blocks are connected in parallel.



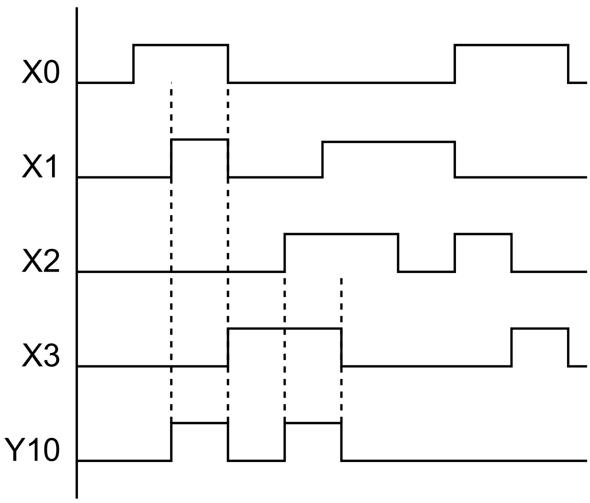
- The start of each block begins with an **ST** instruction.

■ Operation example

Operation of instruction format description program

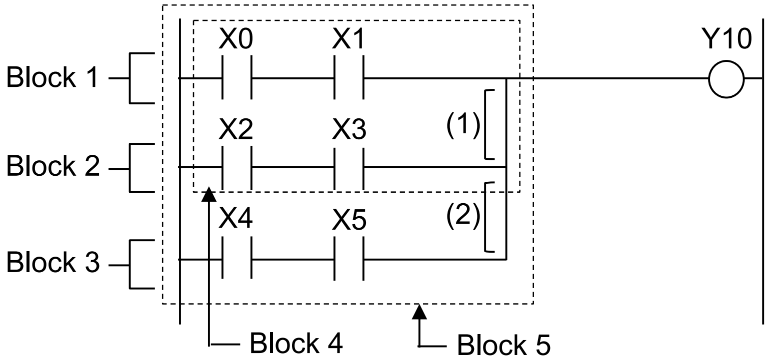
When both X0 and X1 turn ON, or when both X2 and X3 turn ON, the result is output to Y10.

$\underbrace{\text{(X0 AND X1)}}_{\text{Block 1}}$ **OR** $\underbrace{\text{(X2 AND X3)}}_{\text{Block 2}}\rightarrow$Y10



■ When blocks are consecutive

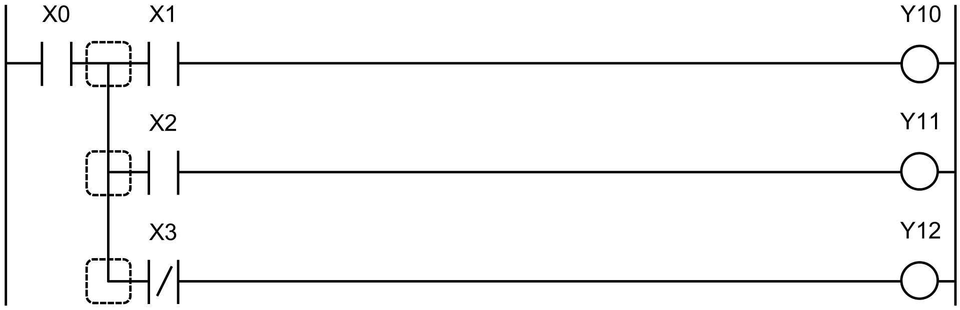
When blocks are consecutive, consider a block division as follows.



Block 5	Block 4	Block 1	ST	X	0	
			AN	X	1	
		Block 2	ST	X	2	
			AN	X	3	
			ORS			------------------- (1)
	Block 3		ST	X	4	
			AN	X	5	
			ORS			------------------- (2)
			OT	Y	10	

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

■ Instruction format



■ Outline of operation

These instructions can be used to store one operation result, read it, and perform multiple processes on it.

Instruction	Operation
PSHS	The operation result immediately before the PSHS instruction is stored and operation continues from the next step.
RDS	The operation result stored by the **PSHS** instruction is read and operation continues from the next step using this result.
POPS	The operation result stored by the **PSHS** instruction is read, operation continues from the next step using this result, and the operation result stored by the **PSHS** instruction is cleared.

This instruction is used when there is branching from a single contact, followed by another contact or contacts.

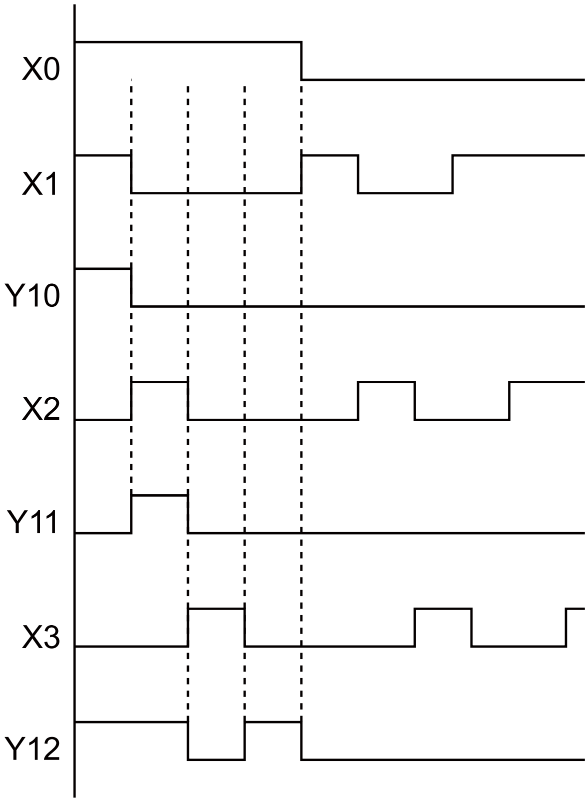
■ Operation example

Operation of instruction format description program

1) When X0 turns ON, the operation result is stored by the**PSHS**instruction, and if X1 is ON, the result is output to Y10.

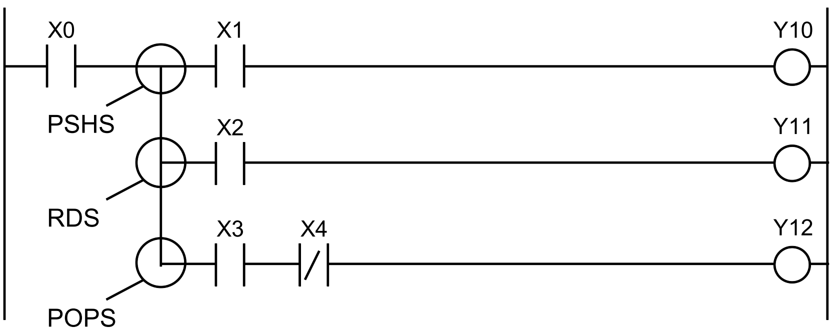
2) The operation result is read by the **RDS** instruction, and if X2 is ON, the result is output to Y11.

3) The operation result is read by the **POPS** instruction, output to Y12 if X3 is OFF, and the operation result stored by the **PSHS** instruction is cleared.

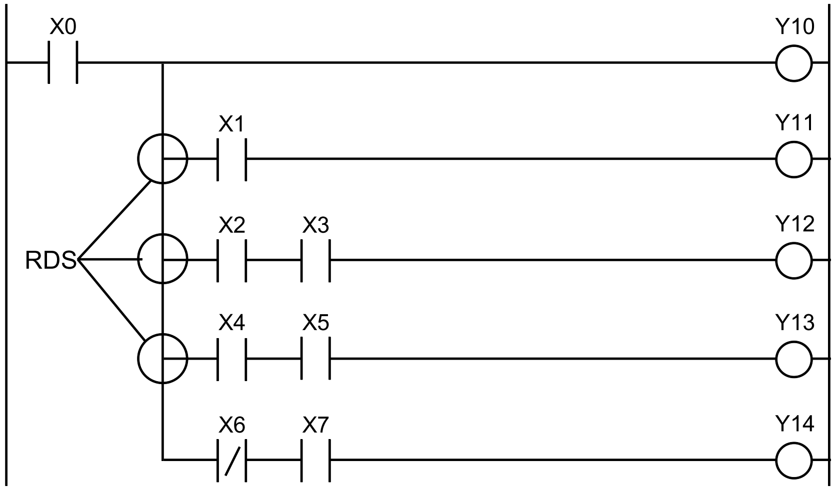


■ **Programming precautions**

- Use the **RDS** instruction when continuing to use the operation result, and use the **POPS** instruction when finishing. (The **POPS** instruction must be included.)

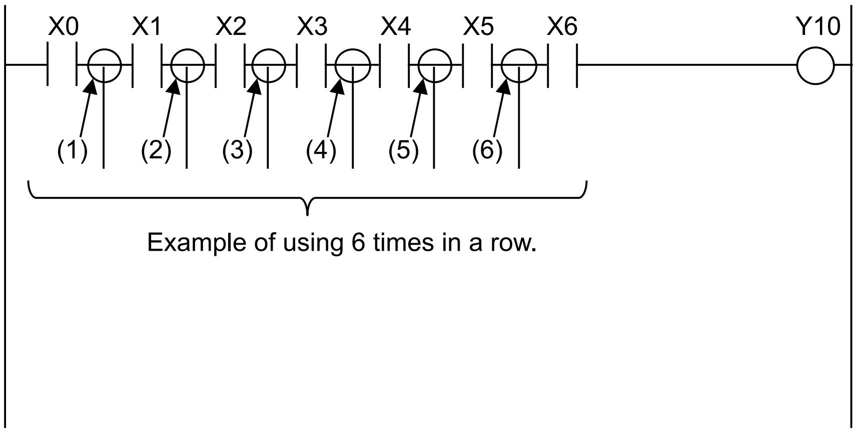


- The **RDS** instruction may be used consecutively as many times as required.



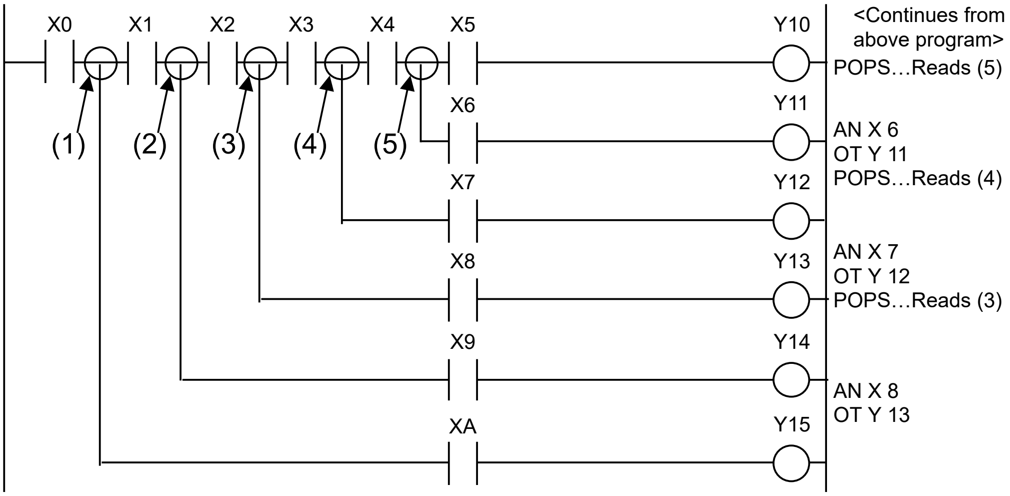
■ Precautions when using thePSHSinstruction consecutively

- The **PSHS** instruction is limited to a maximum of eight consecutive uses.
- Please note that the program will not run correctly if this limit is exceeded.



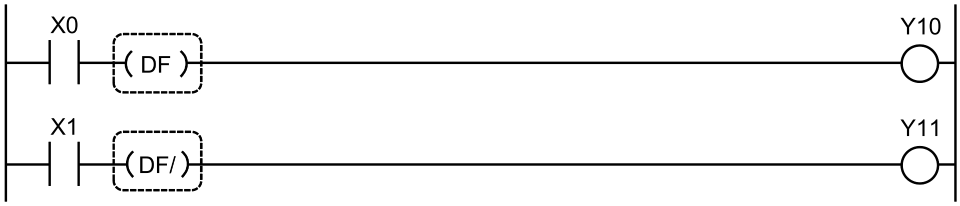

```
ST X 0
PSHS..(1)
AN X 1
PSHS..(2)
AN X 2
PSHS..(3)
AN X 3
PSHS..(4)
AN X 4
PSHS..(5)
AN X 5
PSHS..(6)
AN X 6
OT Y 10
```

- If the **POPS** instruction is used when using the **PSHS** instruction consecutively, reading will take place in order beginning from the last data stored by the **PSHS**instruction.


X0
X1
X2
X3
X4
X5
Y10
<Continues from above program>
POPS…Reads (5)
(1)
(2)
(3)
(4)
(5)
X6
Y11
AN X 6
OT Y 11
POPS…Reads (4)
X7
Y12
X8
Y13
AN X 7
OT Y 12
POPS…Reads (3)
X9
Y14
AN X 8
OT Y 13
XA
Y15

2.10 DF, DF/ (Rise Differential,Fall Differential)

■ Instruction format

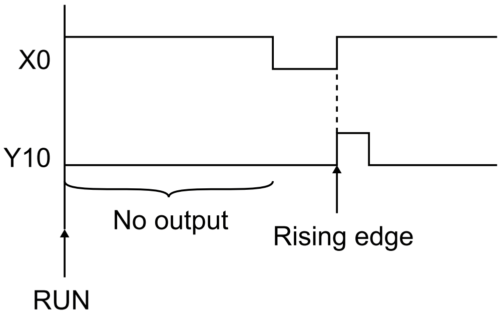


■ Outline of operation

Instruction	Operation
DF	When an execution condition changes from OFF to ON (rise), outputs only that 1 scan (differential output).
DF/	When an execution condition changes from ON to OFF (fall), outputs only that 1 scan (differential output).

- There is no limit to the number of times a differential instruction can be used.
- With a differential instruction, only the changes in the contact’s ON/OFF status are detected, so if execution conditions are met (ON) from the start when switching into "RUN mode" or when powering on in" RUN mode" , there will be no output.

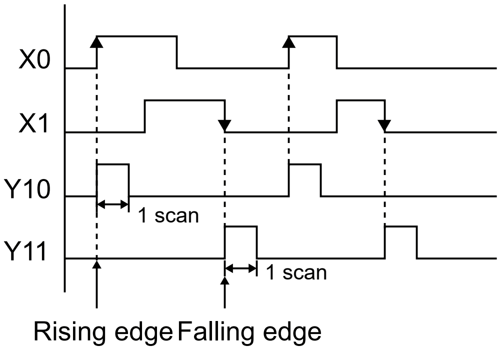
<Example> Rise differential



■ Operation example

Operation of instruction format description program

1. When X0 changes from OFF to ON (rise), only 1 scan is output to Y10.
2. When X1 changes from ON to OFF (fall), only 1 scan is output to Y11.

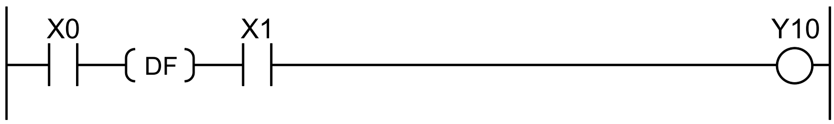


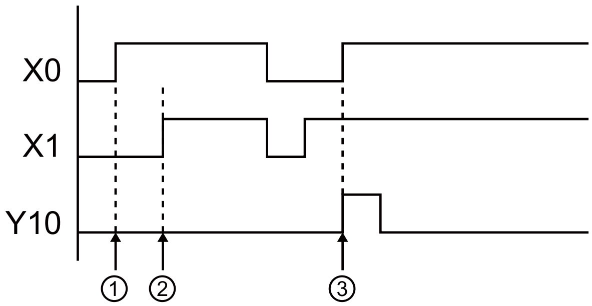
■ **Related instructions**

- The DFI instruction can be used. Only the first 1 scan is executed.

■ **Programming precautions**

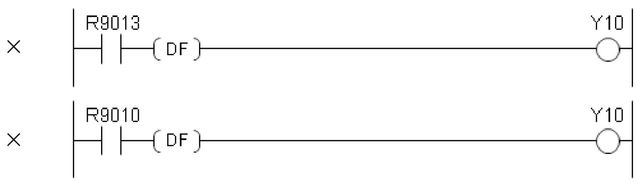
- For the circuit shown below, the operation is as follows.





(1)	When X1 is OFF, Y10 remains OFF even if X0 rises.
(2)	When X0 is ON, Y10 remains OFF even if X1 rises.
(3)	When X1 is ON, if X0 rises, then Y10 turns ON for one scan.

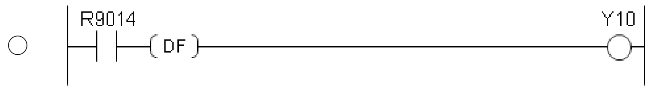
- In the following program, the execution condition is ON from the beginning, so output cannot be obtained.



R9013 only turns ON during the first scan after RUN begins.

R9010 is a normally ON relay.

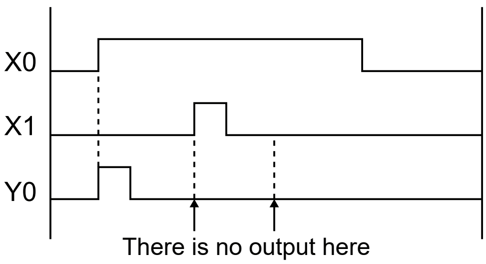
- In the following program, output can be obtained.



R9014 turns ON from the second scan after RUN begins.

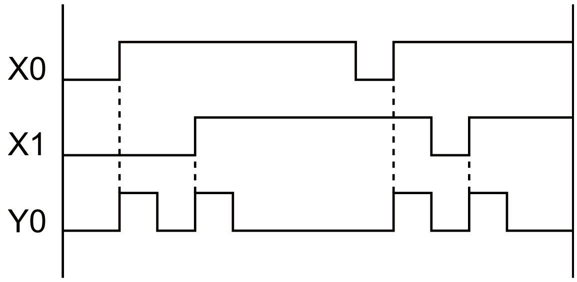
- Caution is required when using differential instructions in combination with instructions that change the order of execution of instructions (1 to 6 below), such as the **MC**/**MCE** instructions or the **JP**/**LBL** instructions.
  1. MC to MCE instructions
  2. JP to LBL instructions
  3. LOOP to LBL instructions
  4. CNDE instruction
  5. Step ladder instructions
  6. Subroutine instructions
- When a differential instruction is combined with an AND stack instruction or a pop stack instruction, take care that the syntax is correct.
- For the circuit shown below, the operation is as follows.

<Time chart>



- To turn Y0 ON at the rise of either X0 or X1, program it as follows.

<Time chart>

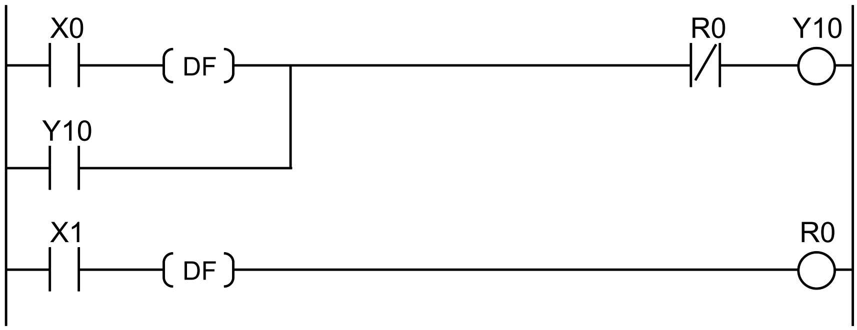


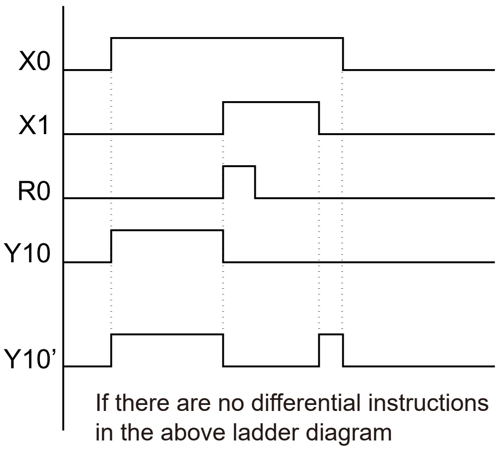
■ Examples of applying differential instructions

- 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.

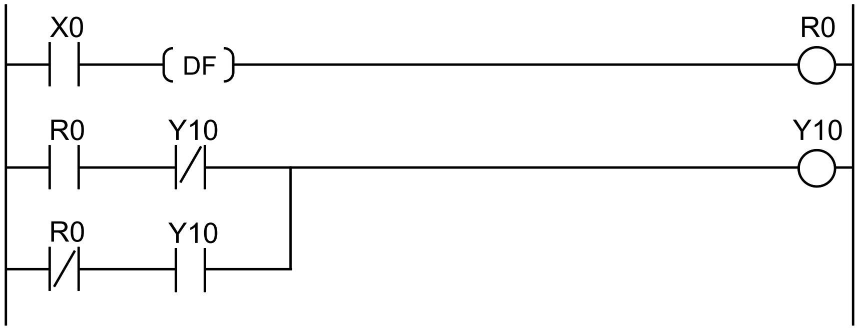




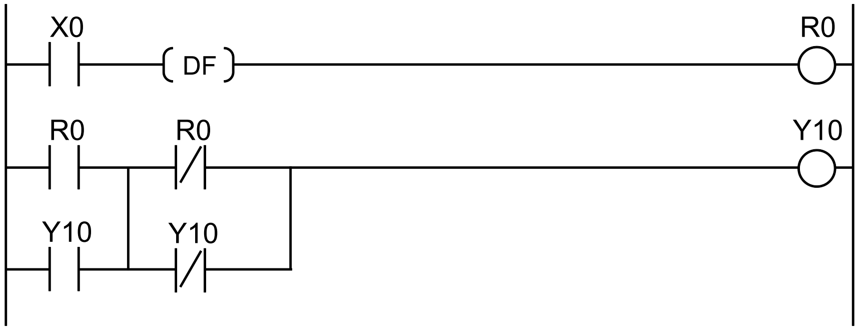
<Example of application to an alternating circuit>

- It can also be applied to alternating circuits that hold and release with a single signal.

<Example 1>

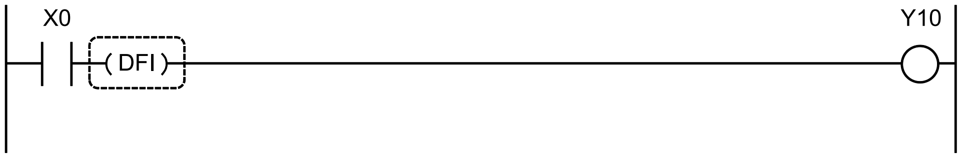


<Example 2>



2.11 DFI [Rise Differential (initial execution type)]

■ Instruction format



■ Outline of operation

Instruction	Operation
DFI	When an execution condition changes from OFF to ON (rise), outputs only that one scan (differential output).

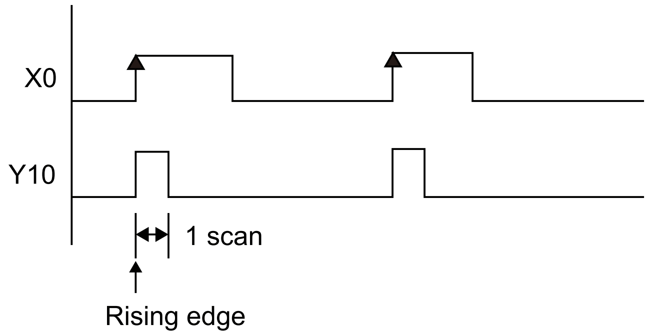
- If the execution condition is met from before RUN starts, output (differential output) is performed at the first scan.
- There is no limit on the number of times the **DFI** instruction can be used.
- If it is possible for execution conditions to be met when switching into "RUN mode" or when powering on in "RUN mode", with the **DF** instruction, output cannot be obtained with the first scan, so using the **DFI** instruction, blocks that were connected in series are connected in parallel.

■ Operation example

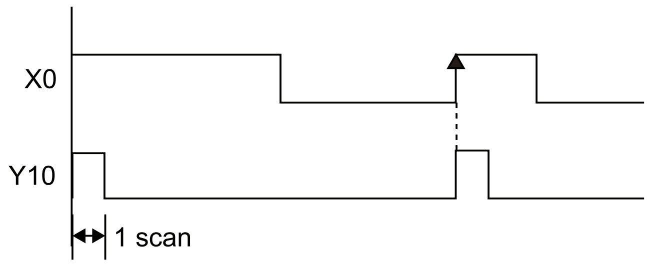
Operation of instruction format description program

When X0 changes from OFF to ON (rise), only 1 scan is output to Y10.

- When execution condition is met after RUN starts



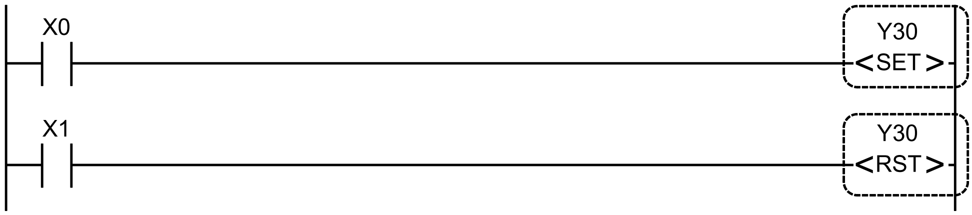
- When execution condition is met from the beginning



- Caution is required when using differential instructions in combination with instructions that change the order of execution of instructions (1 to 6 below), such as the **MC**/**MCE** instructions or the **JP**/**LBL** instructions.
  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
- When a differential instruction is combined with an AND stack instruction or a pop stack instruction, take care that the syntax is correct.

2.12 SET, RST (Set, Reset)

■ Instruction format



■ Instruction list

Instruction	Description
SET	Output coil
RST	Output coil

■ Devices that can be specified (indicated by ●)

Operand	X	Y	R	T	C	L	Index modifier
SET		●	●			●	●
RST		●	●			●	●

■ Outline of operation

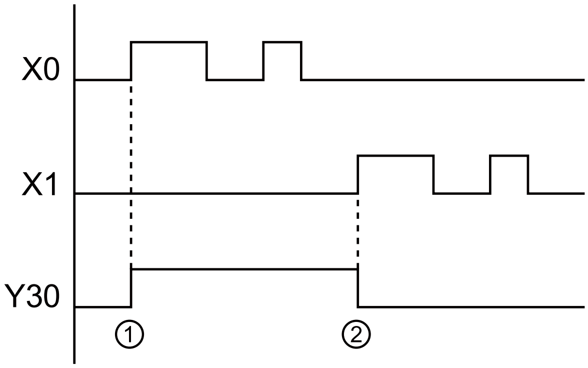
Instruction	Operation
SET	When the execution condition turns ON, the output turns ON and the state is held regardless of a change in the state of the execution condition.
RST	When the execution condition turns ON, the output coil turns OFF and the OFF state is held regardless of a change in the state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

- The same output coil can be specified as many times as desired for the **SET** and **RST** instruction output destinations. (Even if a total check is run, this is not handled as a syntax error.)

■ Operation example

Operation of instruction format description program

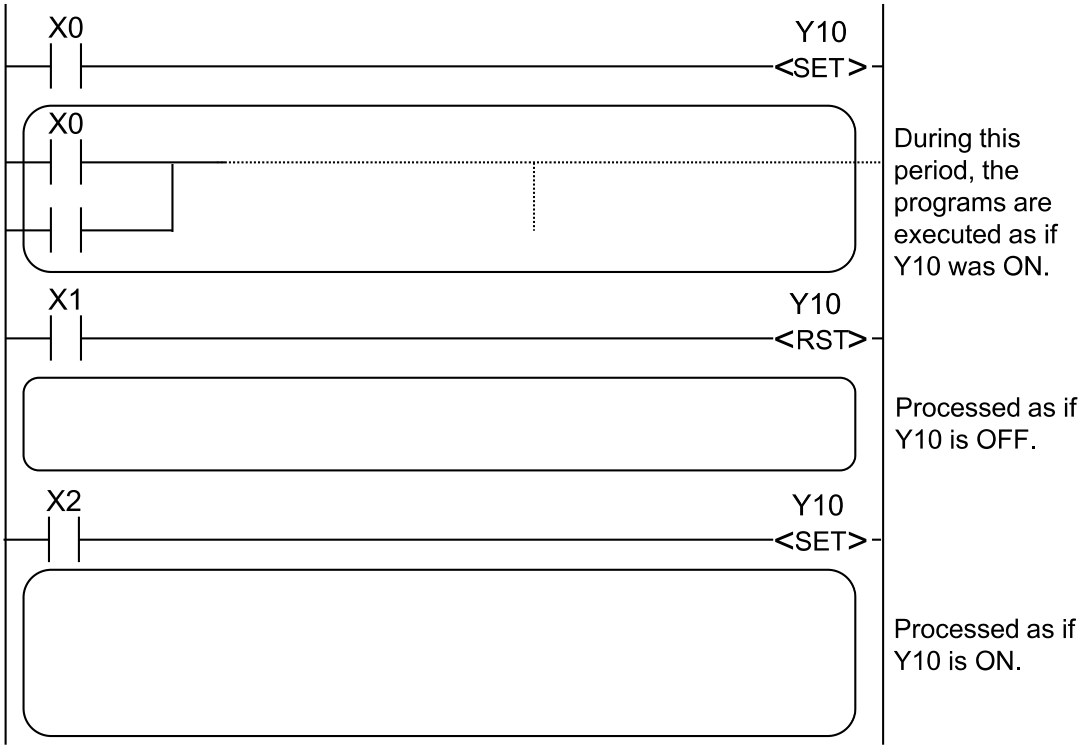
1. When X0 turns ON, Y30 turns ON and is held in that state.
2. When X1 turns ON, Y30 turns OFF and is held in that state.



■ Processing mechanisms when the SET and RST instructions are used

- The output content is overwritten with each step during processing of the operation.

e.g. Processing when X0, X1, and X2 are all turned ON



- I/O refresh is performed when an ED instruction is executed; therefore, the data actually output is determined by the final operation result. In the above example, output occurs with Y10 ON.
- To output a result while the operation is still in progress, use the partial I/O refresh instruction (F143).

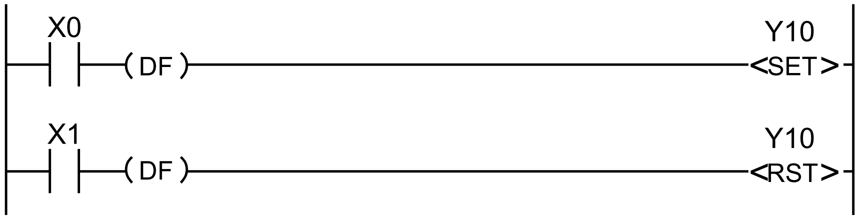
■ Precautions for programming

- The output destination of a SET instruction retains its state even during the operation of an **MC** instruction.

- The output destination of a SET instruction is reset when switching from "RUN" to "PROG. mode" and when the power is turned OFF. (However, if an internal relay set as a hold type is specified as the output destination, reset does not take place.)

■ **SET and RST instructions used as a set with differential instructions**

- Placing a DF differential instruction before the SET and RST instructions makes program development and adjustment easier.
- This is particularly effective when the same output destination is used in several places in the program.

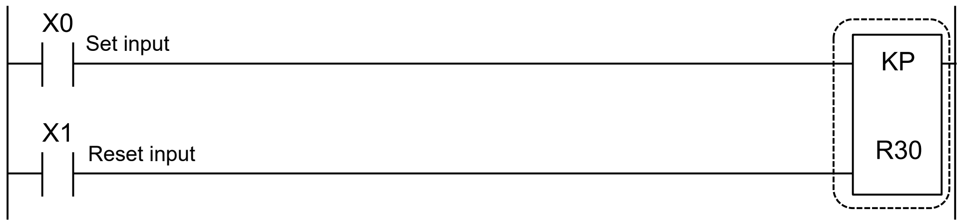


■ **Handling of relays in SET and RST instructions**

- The RST instruction can be used to turn OFF relays.
- Using various relays with a SET or RST instruction does not cause duplicated output.
- Pulse relays cannot be specified as the output destination for the SET and RST instructions.

2.13 KP (Keep)

■ Instruction format



■ Instruction list

Instruction	Description
KP	Output coil

■ Devices that can be specified (indicated by ●)

Operands	X	Y	R	T	C	L	Index modifier
KP		●	●			●	

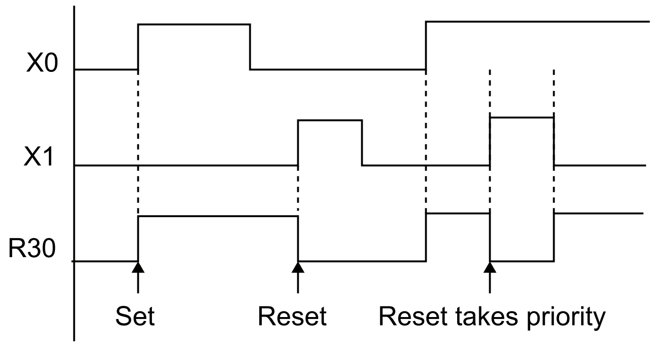
■ Outline of operation

- When the set input turns ON, output of the specified coil turns ON and is held in that state.
- When the reset input turns ON, the hold state is released.
- The output is held in an ON state until the reset input turns ON, regardless of the ON/OFF state of the set input.
- If the set input and reset input turn ON simultaneously, the reset input takes priority. Serially connected blocks are connected in parallel.

■ Operation example

Operation of instruction format description program

1) When X0 turns ON, output of the specified coil turns ON and is held in that state.
2) When X1 turns ON, the hold state is released.

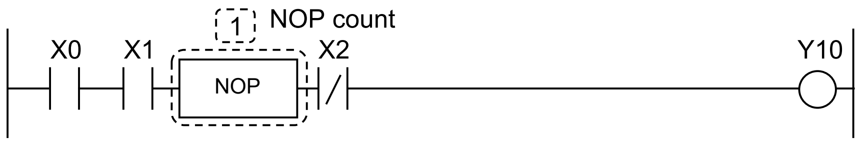


■ Precautions for programming

- The state of the output destination is held even during operation of the **MC** instruction.
- The output is reset when switching from"RUN mode"to"PROG. mode"and when the power is turned OFF. (However, if an internal relay set as a hold type is specified as the output destination, reset does not take place.)

2.14 NOP

■ Instruction format



■ Outline of operation

- This instruction has no effect on the operation results to that point. The same operation is performed even without a **NOP** instruction.
- A **NOP** instruction can be used to make the program easier to read when checking or correcting.
- Write a **NOP** instruction (overwrite the previous instruction) when you want to delete an instruction without changing addresses.
- Insert a **NOP** instruction when you want to move the addresses of one part of a program without changing the program.
- For example, this is a convenient means of breaking a long program into several blocks.

e.g.

To move the starting point of a program block from address 39 to address 40, insert a **NOP** instruction at address 39.

Address			
36	ST	X0	
	OR	X1	
	OT	Y10	
39	ST	X2	→ Want to start from address 40
40	AN	X3	
41	OT	R20	
42	ST	R2	
43	DF		
44	ST	X3	

Address			
36	ST	X0	
	OR	X1	
	OT	Y10	
39	NOP		← Insert NOP
40	ST	X2	
41	AN	X3	
42	OT	R20	
43	ST	R2	
44	DF		
45	ST	X3	

■ DeletingNOPinstructions

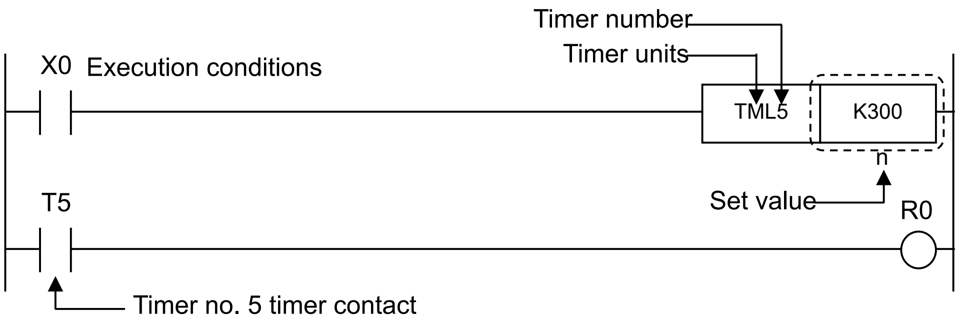
After creating a program, it is possible to delete all **NOP** instructions in a program by using the programming tool.

(MEMO)

3 Basic Function Instructions

3.1 TML/TMR/TMX/TMY (0.001 s, 0.01 s, 0.1 s, 1 s On-delay Timer)

■ Instruction format



■ Instruction list

Instruction	Description
n	Timer set value

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
n	•	•	•	•	•	•	•	•		•	•	•					

■ Outline of operation

- The timer is a non-hold type that is reset when the power is turned off or when switching from "RUN mode" to "PROG. mode". (If the operating state must be held, set system register No. 6. In that case, be sure to use a battery.)
- When the execution condition turns ON, the set time decrements until the elapsed value becomes 0, at which point timer contact Tn (n is the timer contact number) turns ON.
- If the execution condition turns OFF during while the set time is decrementing, the operation is interrupted and the elapsed value is reset (cleared to 0).
- The OT instruction can also be written immediately after a timer coil.

■ Setting the timer period

1. The timer set time is (timer unit) × (timer set value).
2. The timer set value [n] is set as a decimal constant in the range of K1 to K32767.

TML	0.001 to 32.767 seconds in units of 0.001 second
TMR	0.01 to 327.67 seconds in units of 0.01 second
TMX	0.1 to 3276.7 seconds in units of 0.1 second
TMY	1 to 32,767 seconds in units of 1 second

e.g. When K43 is set by TMX, the set time is 0.1 × 43 = 4.3 seconds.
When K500 is set by TMR, the set time is 0.01 × 500 = 5 seconds.

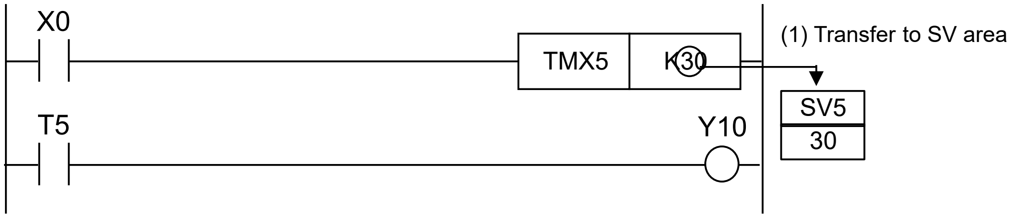
■ Precautions for programming

- As subtraction operations are performed during operation, create the program so that it operates once during one scan. If an operation is performed more than once during one scan or cannot be performed even once due to an interrupt processing program or jump/loop instruction, correct results cannot be obtained.
- When combining a timer instruction with an AND stack instruction or a POP stack instruction, be careful that the programming is correct.

■ Timer operation mechanism

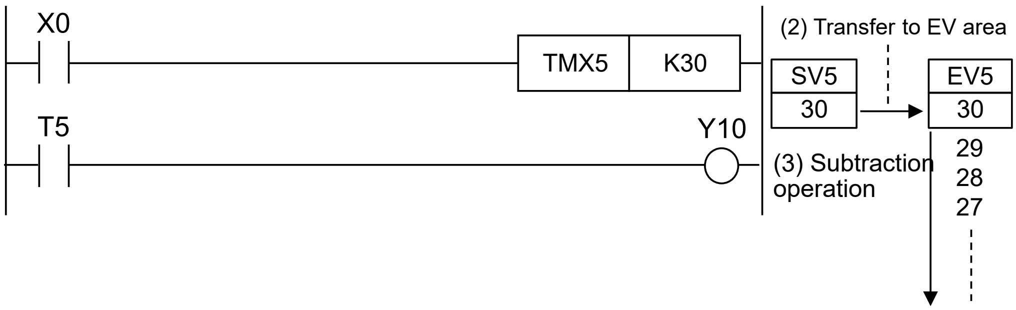
The following are examples of specifying a K constant as the set value. See below for the operation when specifying the set value area number.

1. When the mode is switched to "RUN mode" or when the power is turned ON in "RUN mode" , the timer set value is transferred to the set value area "SV" of the same number.

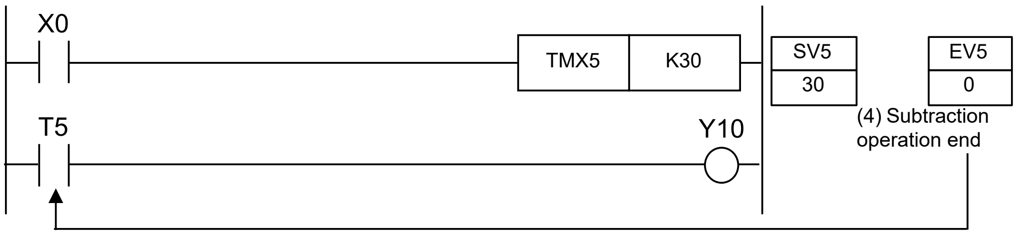


2. When the timer execution condition rises from OFF to ON, the timer set value is transferred from the set value area "SV" to the elapsed value area "EV" of the same number.

   (The same operation is performed when switching to "RUN mode" while the execution condition is ON.)
3. For each scan, if the execution condition is ON, the timer decrements by the value in the elapsed value area "EV" .



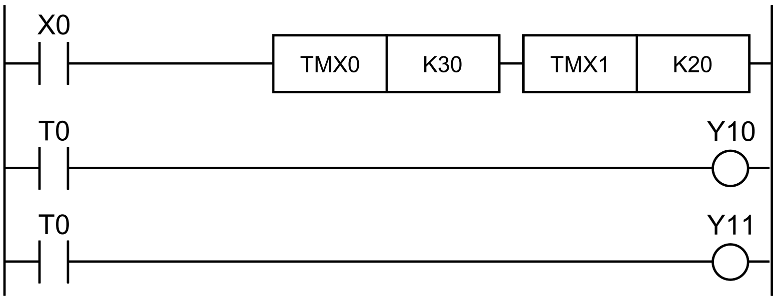
4. When the value of the elapsed value area "EV" becomes 0, the timer contact "T" of the same number turns ON.



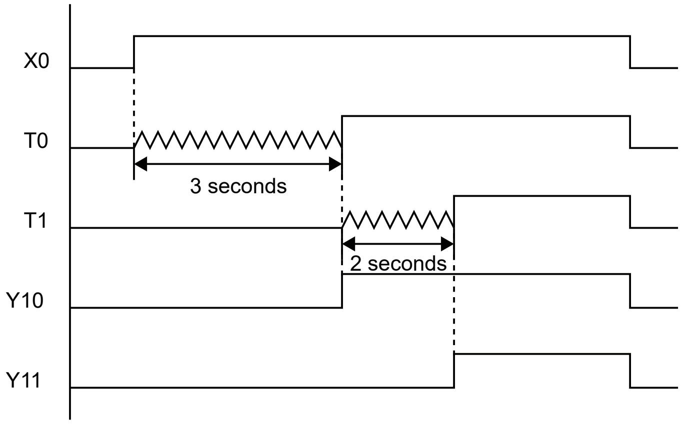
■ Examples of timer instruction application

<Timer series connection>

- **Ladder diagram**

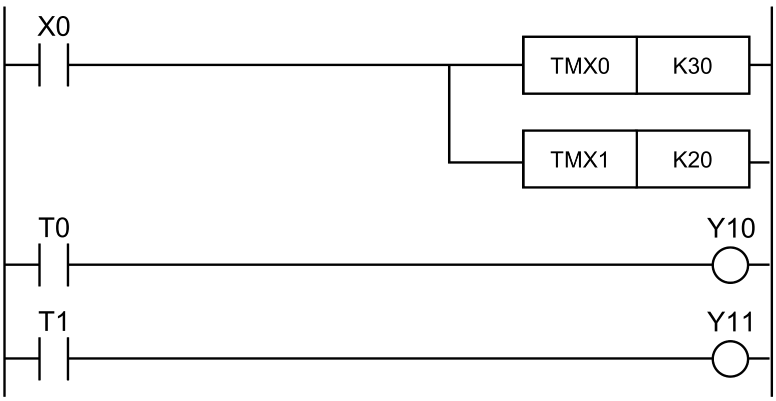


- **Timing chart**

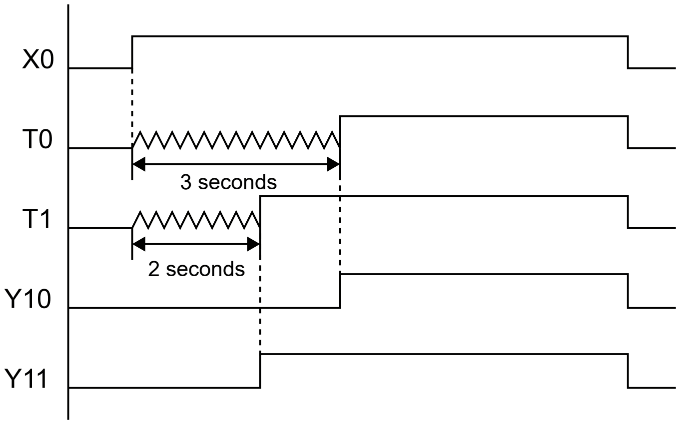


<Timer parallel connection>

- **Ladder diagram**

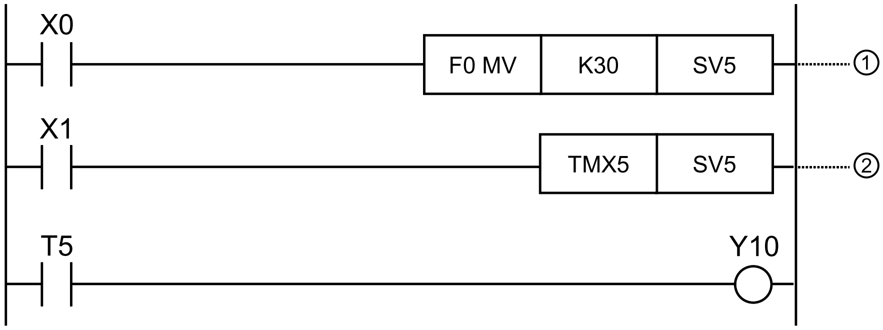


- **Timing chart**



■ How to directly specify the set value area No. for the timer set value

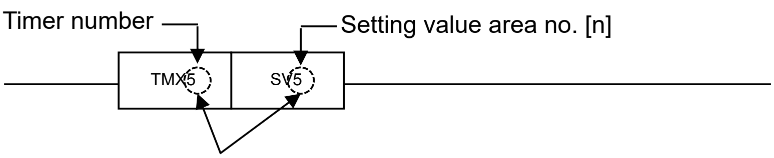
- The set value area number can be specified directly as the set value [n].



The above program in which SV5 is specified as the set value operates as follows.

1. When execution condition X0 turns ON, the data transfer instruction (F0 MV) is executed and SV5 is set to K30.
2. When execution condition X1 turns ON, the set value is set to 30 and the decrement operation starts.

- Set the number of the set value area "SV" specified in [n] to be the same as the timer number.



- Even if the value in the set value area "SV" is changed during the subtraction operation, the subtraction operation will continue from the value before the change.

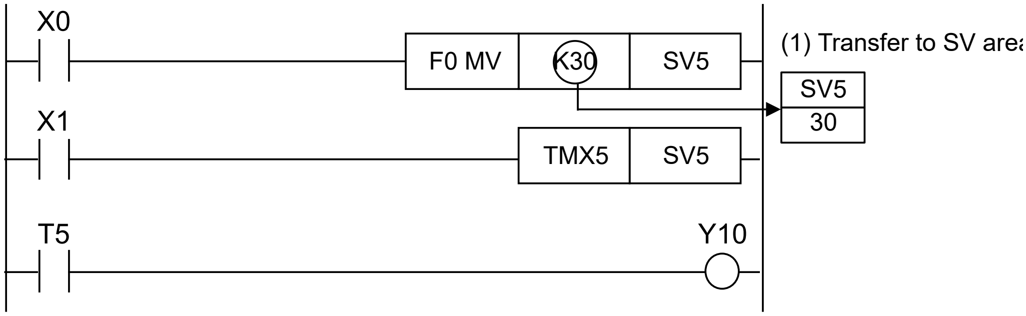
  Timer operation starts with the changed value the next time the execution condition changes from OFF to ON after the decrement operation is completed or interrupted.
- The set value area SV is normally a non-hold type that is reset when the power is turned off or when switching from "RUN mode" to "PROG. mode" .

If the SV value was changed while in RUN mode and that value is to be used as a set value without being reset the next time the power supply is turned on or when switching from "PROG. Mode" to "RUN mode" , set the value to a hold type by using system register no. 6.

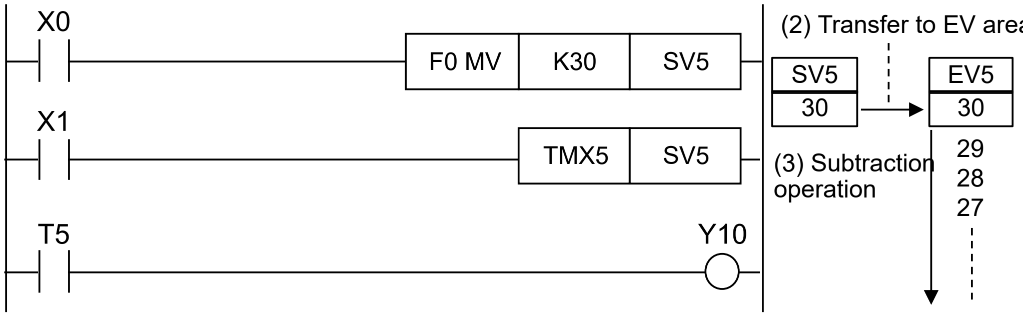
■ Timer operation when the set value area number is directly specified

1. When the execution condition for a high-level instruction is ON, the value is set in the set value area"SV".

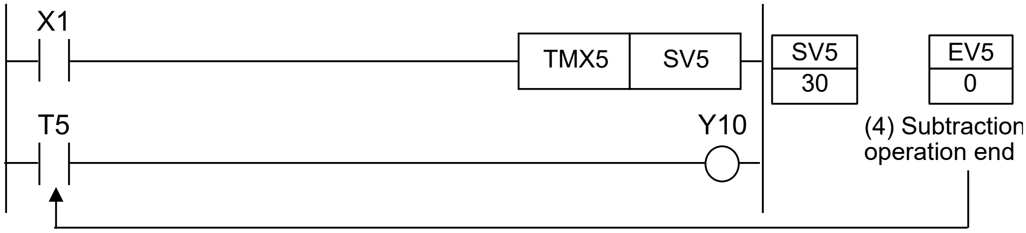
   The following diagram shows an example of using the F0 MV instruction.



2. When the timer execution condition rises from OFF to ON, the timer set value is transferred from the set value area "SV" to the elapsed value area "EV" of the same number. (The same operation is performed when switching to "RUN mode" while the execution condition is ON.)
3. For each scan, if the execution condition is ON, the timer decrements by the value in the elapsed value area "EV ".



4. When the value of the elapsed value area "EV" becomes 0, the timer contact "T" of the same number turns ON.

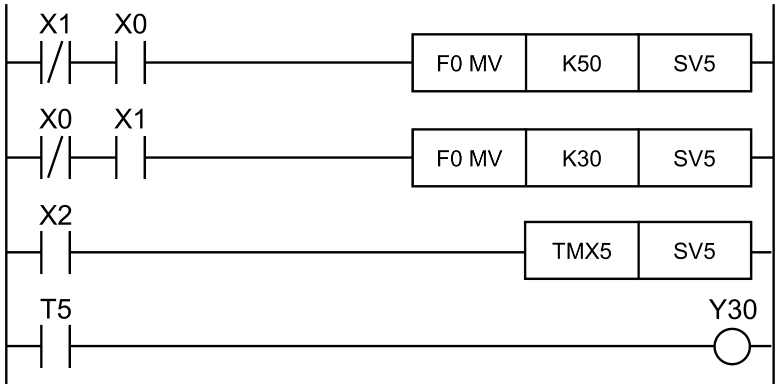


■ Examples of applying direct specification of set value area numbers

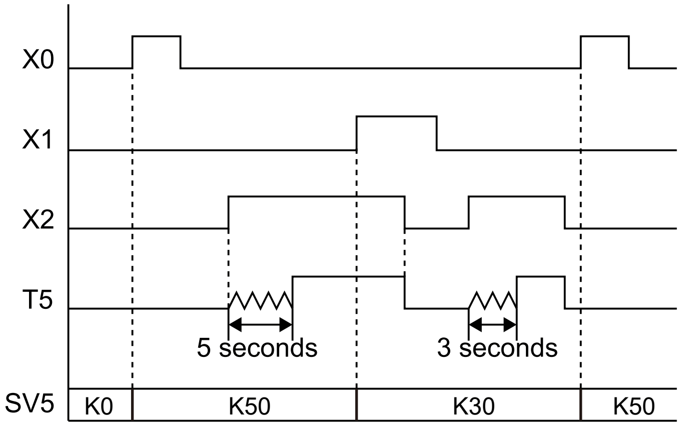
Example 1) Changing set values based on specified conditions

The set value is K50 when X0 is ON and K30 when X1 is ON.

- **Ladder diagram**



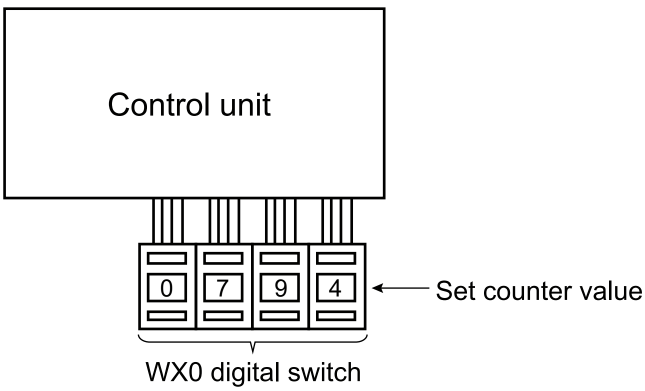
- **Timing chart**



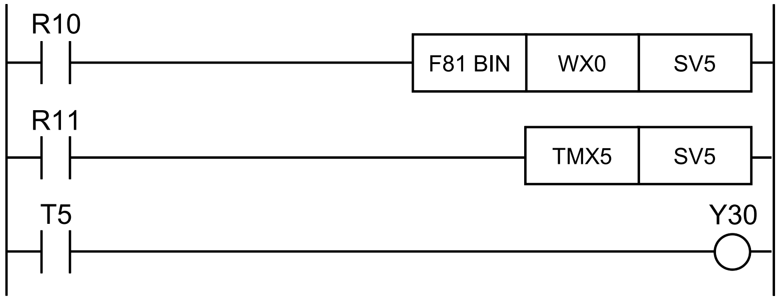
Example 2) Setting a set value from external digital switches

The BCD data of the digital switches connected to X0 through XF is converted and becomes the set value.

- **Connection example**

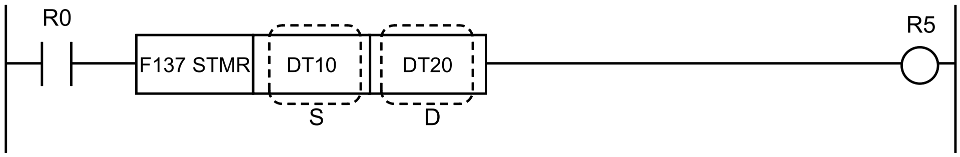


- **Ladder diagram**



3.2 F137 STMR (16-bit, 0.01 s On-delay Timer)

■ Instruction format



■ Instruction list

Instruction	Description
S	Area storing the setting value, or constant data
D	Process value area

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S	●	●	●	●	●	●	●	●	●	●	●	●	●				
D		●	●	●	●	●	●	●	●								

■ Outline of operation

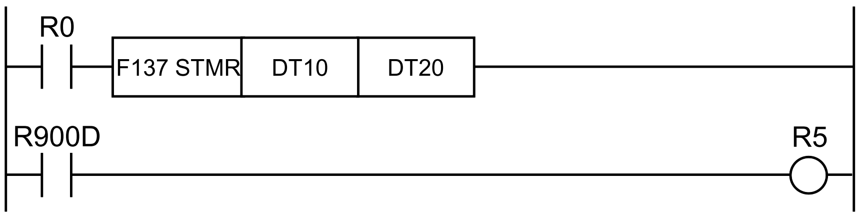
Operates as an ON-delay timer in units of 0.01 seconds. When the internal relay is ON, the setting time is subtracted, and the special internal relay R900D turns ON when the process value [D] becomes 0. (It is OFF when the internal relay is OFF and during subtraction.)

■ Operation example

Operation of instruction format description program

The internal relay is executed, the auxiliary timer is activated, and when a time equal to the value stored in [DT10] × 0.01 seconds has elapsed, R5 turns ON.

- When the internal relay is OFF, the process value area is cleared to 0. The relay in use for the OT instruction turns OFF.
- When the time of the special internal relay R900D is up, it turns ON. It is also possible to use R900D as a timer contact. (It is OFF when the internal relay is OFF and during subtraction.)



Operation is the same as the above example.

■ **Setting the timer period**

1. The timer period is 0.01 × [timer set value].
2. The timer set value is set with a K constant within the range of K1 to K32767.

"STMR" ranges from 0.01 to 327.67 seconds, in units of 0.01 second.

e.g. If the set value is K500, the set time is 0.01 × 500 = 5 seconds.

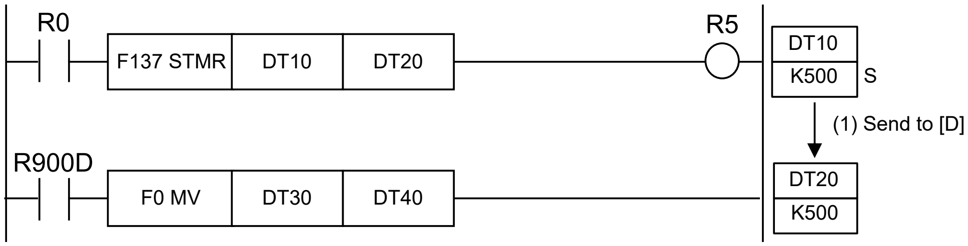
■ **Precautions for programming**

- 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.
- As subtraction operations are performed during operation, create the program so that it operates once during one scan.

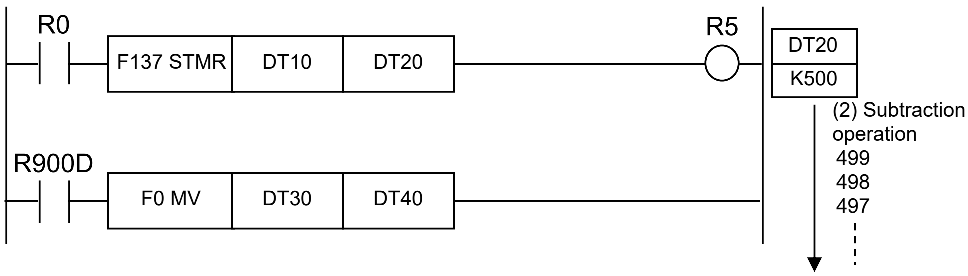
  (During interrupt processing programs or with jump/loop instructions, a correct result cannot be obtained if there are multiple or no operations during one scan.)

■ **How the auxiliary timer works**

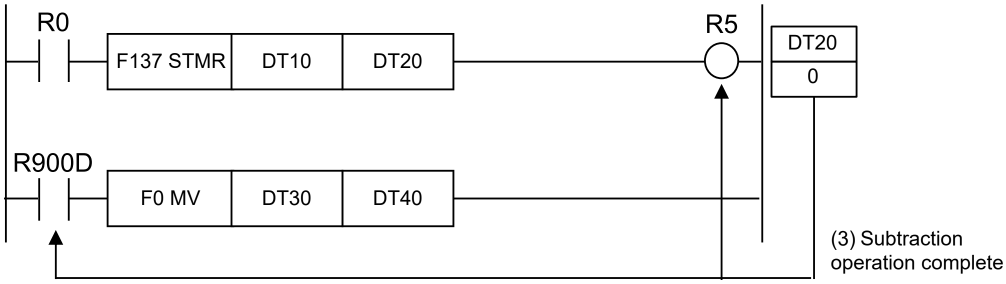
1. When the internal relay turns from OFF to ON, the set value specified by [S] is transferred to the process value area [D].



2. With each scan, if the internal relay is ON, the value of the process value area [D] is subtracted.



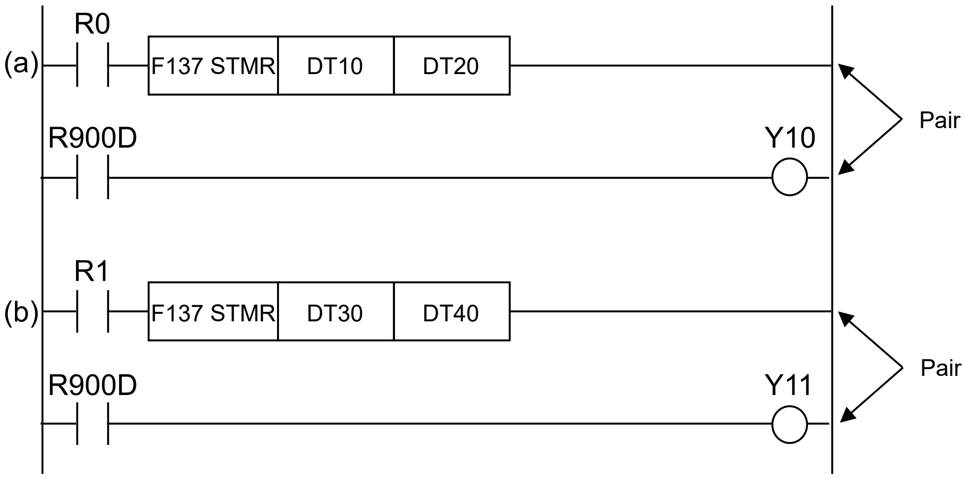
3. If the value of the process value area [D] becomes 0, then the relay in use for the next OT instruction turns ON. The special internal relay R900D also turns ON.



■ Precautions when using R900D

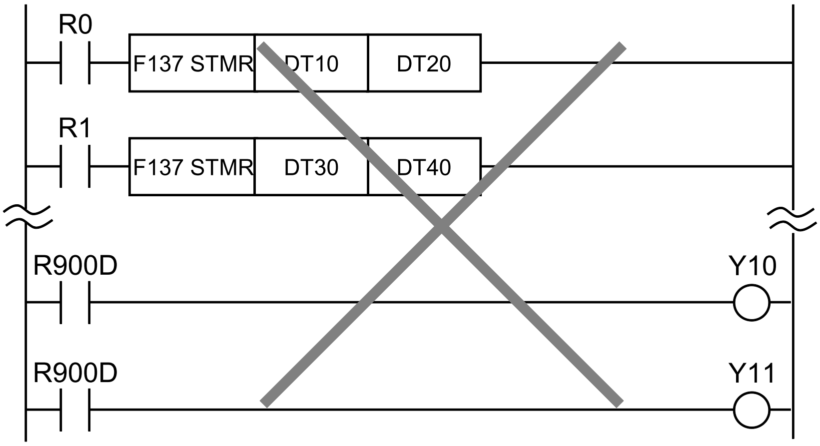
When using multiple auxiliary timers with R900D, ensure that R900D is used on the line after the auxiliary timer instruction.

<Example>



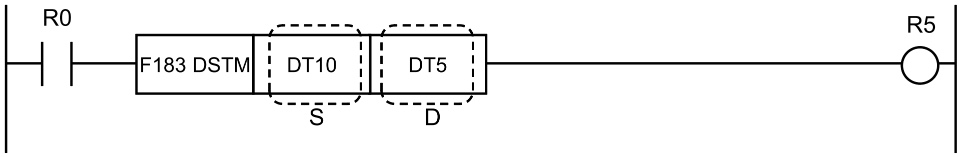
When the time is up for timer (a), activated by R0:ON, Y0 turns ON. When the time is up for timer (b), activated by R1:ON, Y1 turns ON.

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



3.3 F183 DSTM (32-bit, 0.01 s On-delay Timer)

■ Instruction format



■ Instruction list

Instruction	Description
S	Area storing the setting value, or constant data
D	Process value area

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●				
D		●	●	●	●	●	●	●									

■ Outline of operation

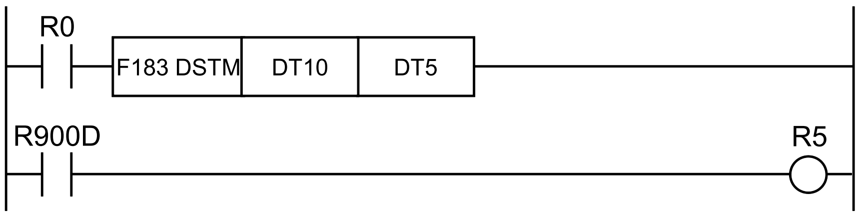
- This instruction operates as a 32-bit addition expression ON-delay timer set in 0.01-second units.
- When the internal relay turns ON, addition of the elapsed time is performed. When the elapsed value [D, D+1] (32 bits) equals or exceeds the set value, the relays used by the OT instruction described next in the program are turned ON.

■ Operation example

Operation of instruction format description program

The internal relay condition is established, the auxiliary timer becomes active, and when the value stored in data registers DT10 and DT11 × 0.01 seconds has elapsed, R5 turns ON.

- When the internal relay is OFF, the process value area is cleared to 0. The relay in use for the OT instruction turns OFF.
- When the time of the special internal relay R900D is up, it turns ON. It is also possible to use R900D as a timer contact. (Turns OFF when the internal relay is OFF and during addition.)



Operation is the same as the above example.

■ Setting the timer period

1. The timer period is 0.01 × [timer set value].
2. The timer set value is set as a K constant in the range of K1 to K2147483647.

0.01 to 21474836.47 seconds in units of 0.01 second.

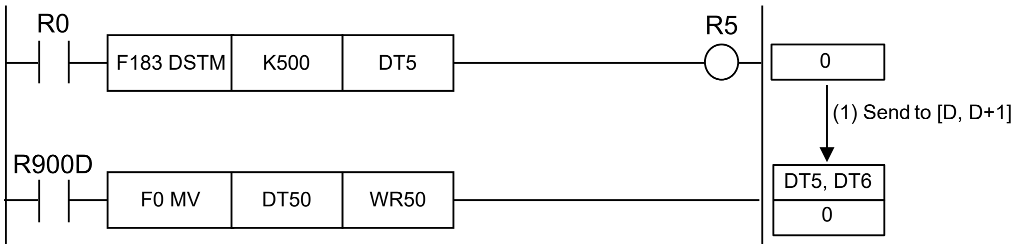
Example) If the set value is K500, the set time is 0.01 × 500 = 5 seconds.

■ Precautions for programming

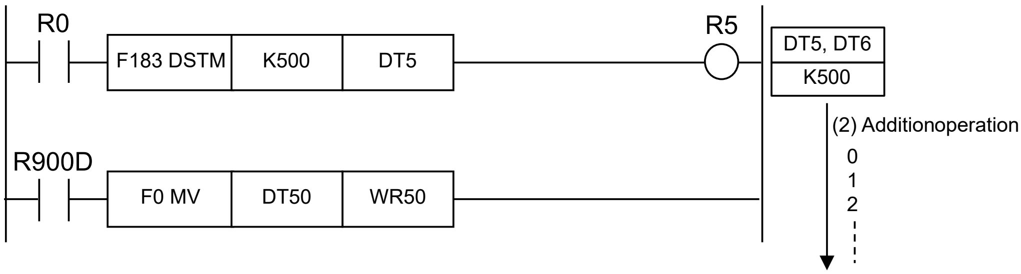
- Ensure that the specifications of the area storing the set value and the process value area do not overlap with other timer/counter instructions or operation memory areas of high-level instructions.
- Addition is performed when the operation is executed, so the program should be created so the an operation is executed once per scan. (If an operation is performed more than once during one scan or cannot be performed even once due to an interrupt processing program or jump/loop instruction, correct results cannot be obtained.)

■ How the auxiliary timer works

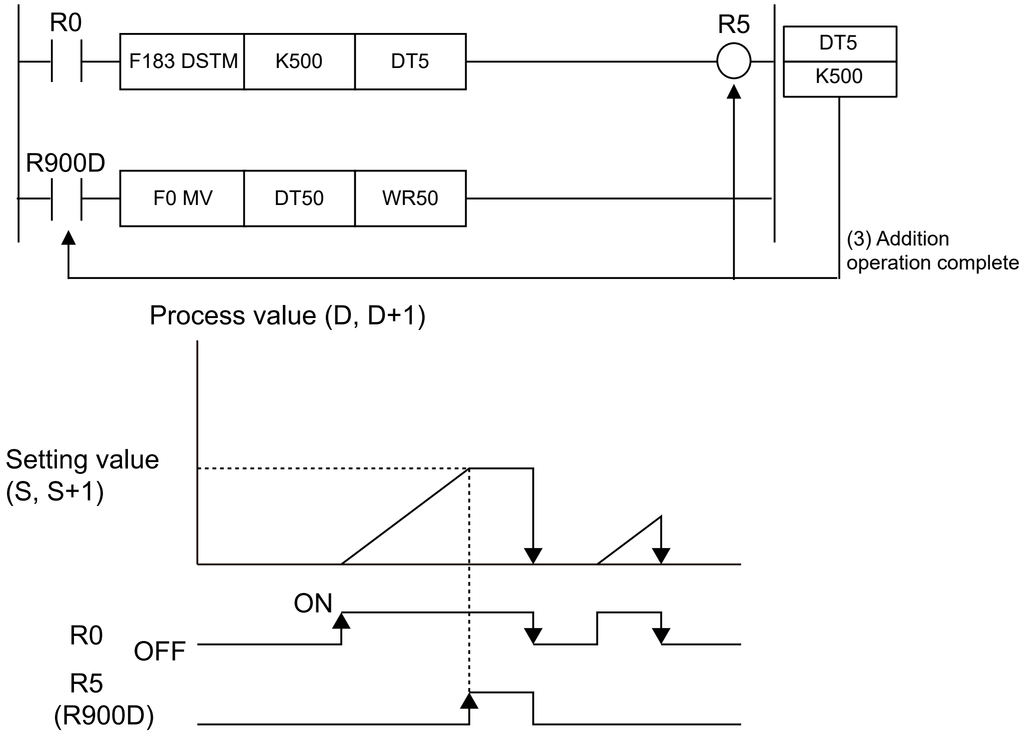
1. When the internal relay changes from OFF to ON, 0s are transferred to the elapsed value area [D, D+1].



2. During each scan, if the internal relay is ON, the values in the elapsed value area of [D, D +1] are added.



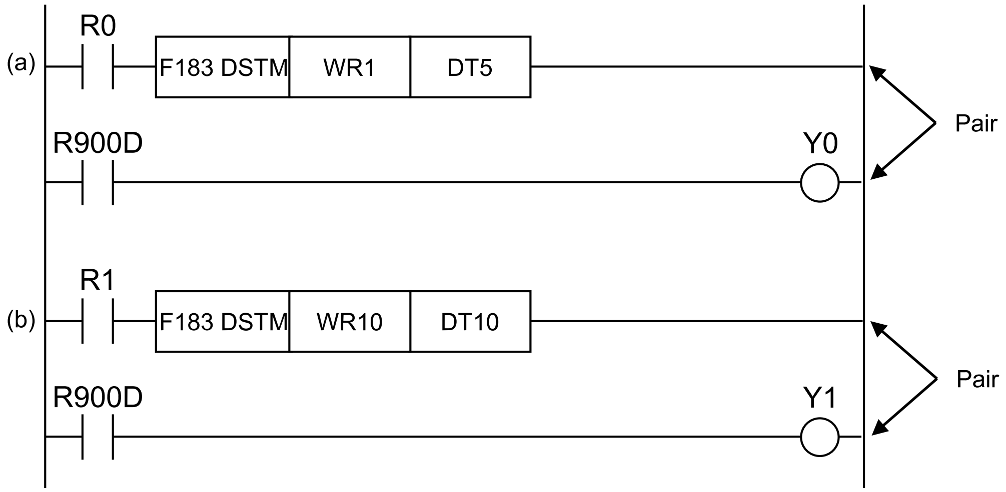
3. When the values in the elapsed value area [D, D+1] equal the values of [S, S+1], the relays used by the OT instruction described next in the program are turned ON. The special internal relay R900D also turns ON.



■ **Precautions when using R900D**

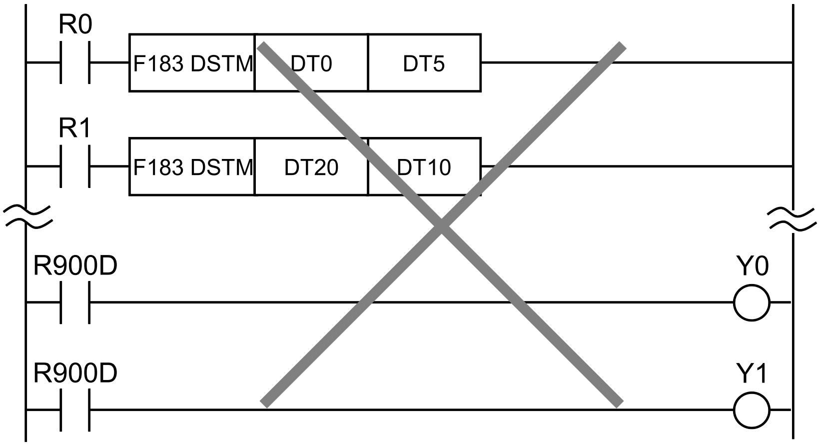
When using multiple auxiliary timers with R900D, ensure that R900D is used on the line after the auxiliary timer instruction.

<Example>



When the time is up for timer (a), activated by R0:ON, Y0 turns ON. When the time is up for timer (b), activated by R1:ON, Y1 turns ON.

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


R0
F183 DSTM
DT0
DT5
R1
F183 DSTM
DT20
DT10
R900D
Y0
R900D
Y1

3.4 CT [Counter (Preset Subtraction Expression)]

■ Instruction format

■ Instruction list

Instruction	Description
n	Counter set value

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
n	●	●	●	●	●	●	●	●		●	●	●					

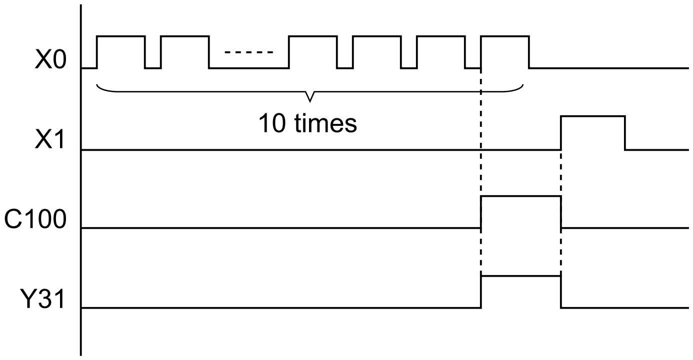
■ Outline of operation

- All counters are subtraction preset counters.
- When the reset input falls from ON to OFF, the value of the set value area SV is preset in the elapsed value area (EV).
- When the reset input is ON, the elapsed value is reset to 0.
- When the count input changes from OFF to ON, the set value is subtracted, and when the elapsed value reaches 0, it is output to the counter contact Cn (n is the counter number).
- 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.

■ Operation example

Operation of instruction format description program

1. If X0 is turned ON 10 times, C100 turns ON, and Y31 turns ON.
2. The elapsed value is reset when X1 turns ON.



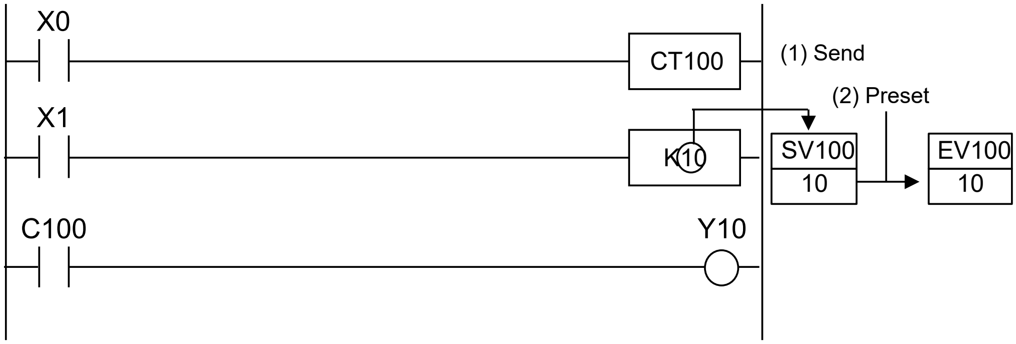
Setting the count value

The count value can be set to a decimal constant (K constant) in a setting range from K0 to K32767.

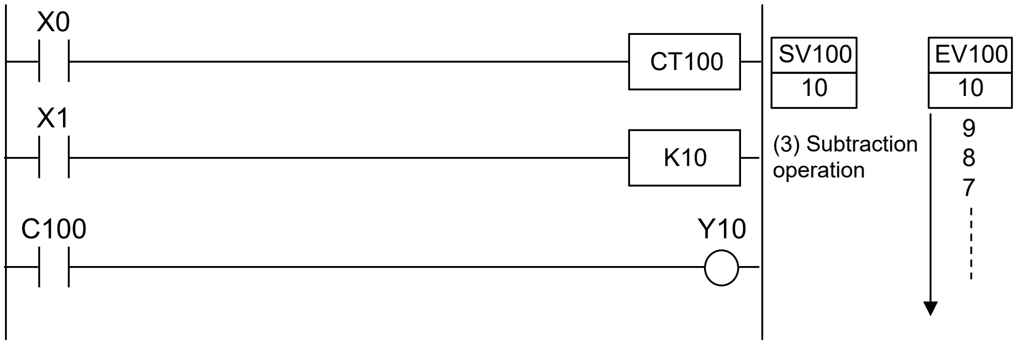
Counter operation

The following are examples of specifying a K constant as the set value. For an explanation of operations when a set value area number is specified, see"P.3-19". (This example shows a case in which “100” is specified for the counter.)

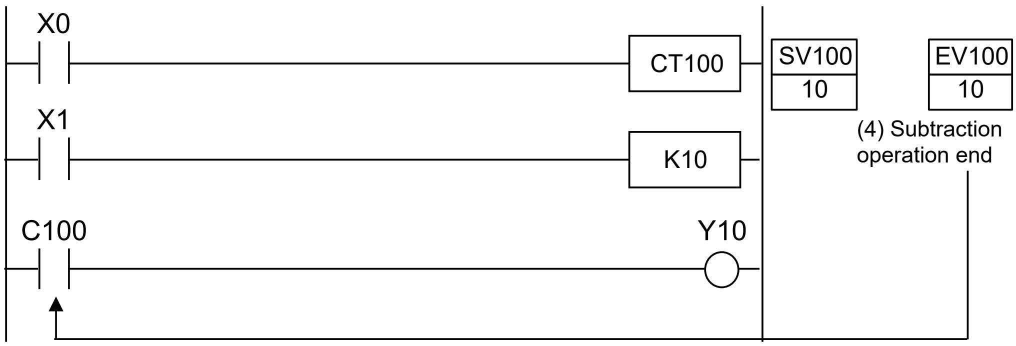
1. When switched to "RUN mode" or when the power is turned ON in "RUN mode" , the counter set value is transferred to the set value area "SV" with the same number.
2. When the reset input falls, the value in the set value area SV is preset in the elapsed value area EV.



3. Each time the count input X0 turns ON, the value in the elapsed value area "EV" is subtracted.



4. When the value in the elapsed value area "EV" reaches zero, the counter contact "C" with the same number turns ON.



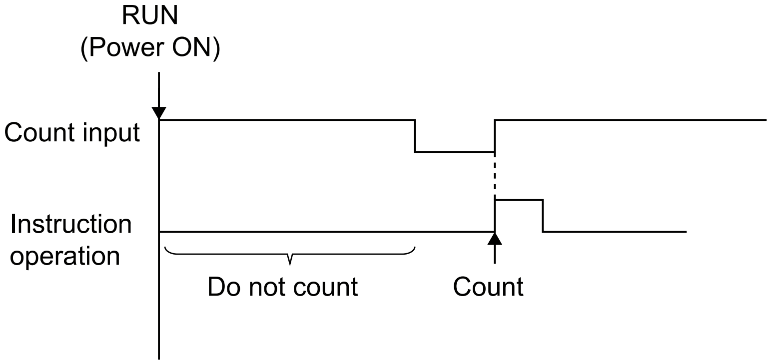
Precautions for programming

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.
- In cases where the count input is initially ON, such as when the mode is switched to RUN or when the power is turned on when in "RUN mode" , subtraction will not take place at the first scan.



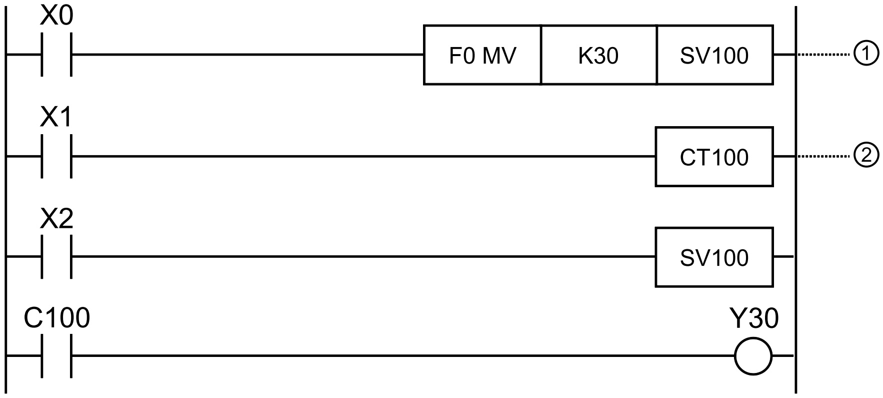
- Be aware that, if used with instructions that change the order of execution such as the MC to MCE instructions or the JP to LBL instructions (see 1 to 6 below), the operation of instructions may change depending on the timing of instruction execution and count input.
  1. MC to MCE instructions
  2. JP to LBL instructions
  3. LOOP to LBL instructions
  4. CNDE instruction
  5. Step ladder instructions
  6. Subroutine instructions

■ Related instructions

- Counter instructions also include an up/down counter instruction (F118 UDC).
- An increment instruction (F35+1) can be used to provide the same type of function.

■ Directly specifying a set value area number as a counter set value

The set value area number can be specified directly as the set value [n].

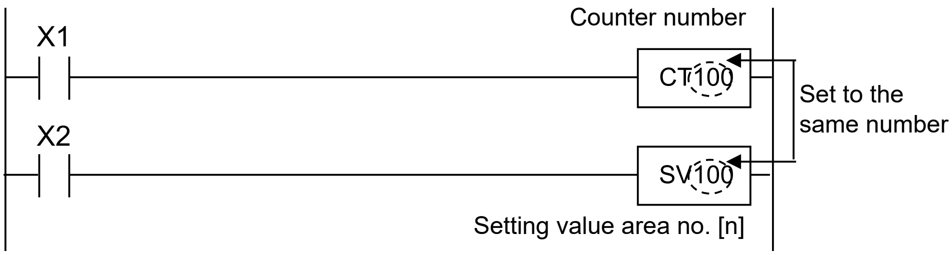


The program described above, which specifies SV100 for the set value, operates as follows.

1. When execution condition X0 is ON, the data transfer instruction (F0 MV) is executed and K30 is set in SV100.
2. When the count input X1 turns ON, the subtraction operation begins from the set value of 30.

- Make the address of the set value area "SV" that specifies [n] the same as the counter number.

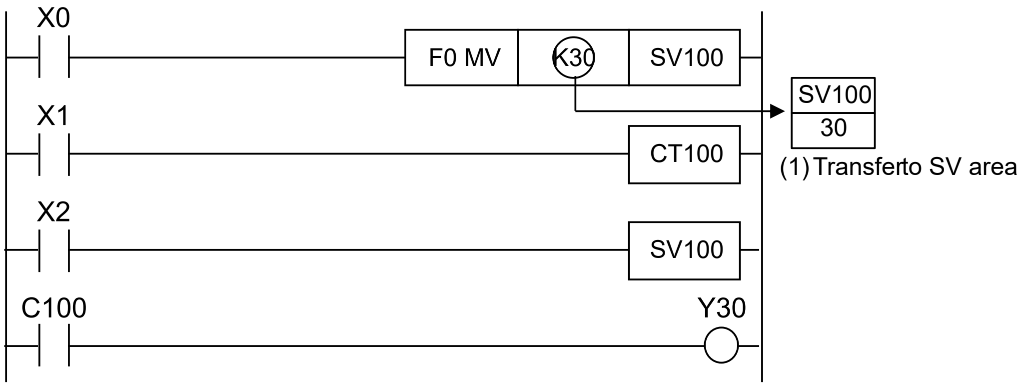
Display:



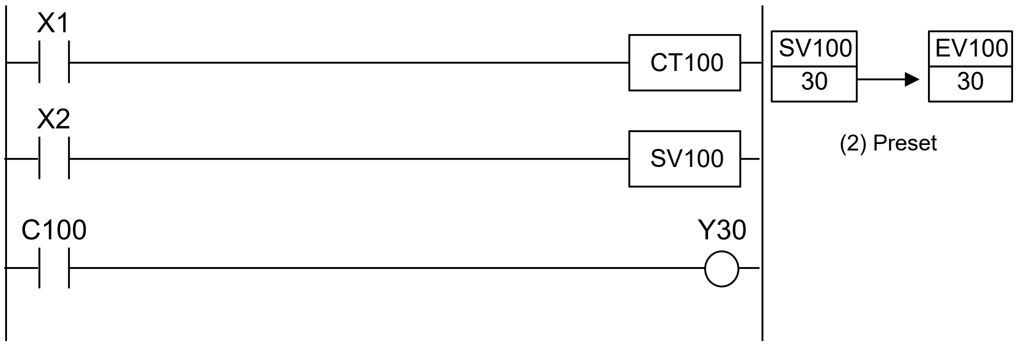
- Even if the value in the set value area "SV" is changed during the subtraction operation, the subtraction operation will continue from the value before the change. Counter operation from the new value will not begin until the counter is reset and the count input subsequently changes from OFF to ON.

■ Counter operation when a set value area number is directly specified

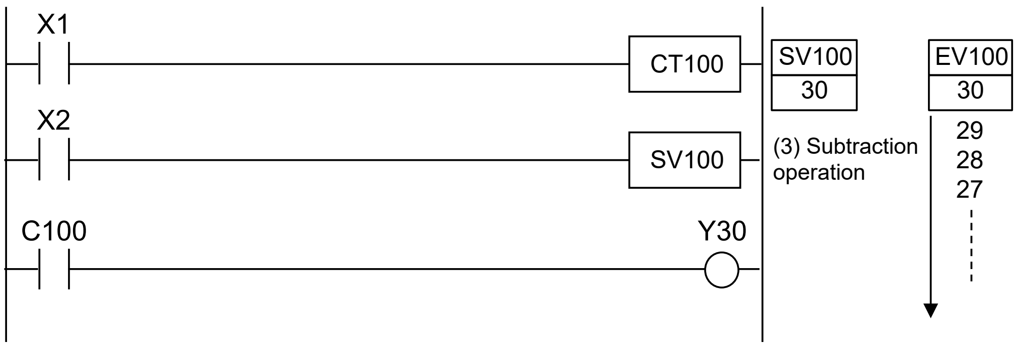
1. When the execution condition for a high-level instruction is ON, the value is set in the set value area "SV" . The following diagram shows an example of using the F0 MV instruction.



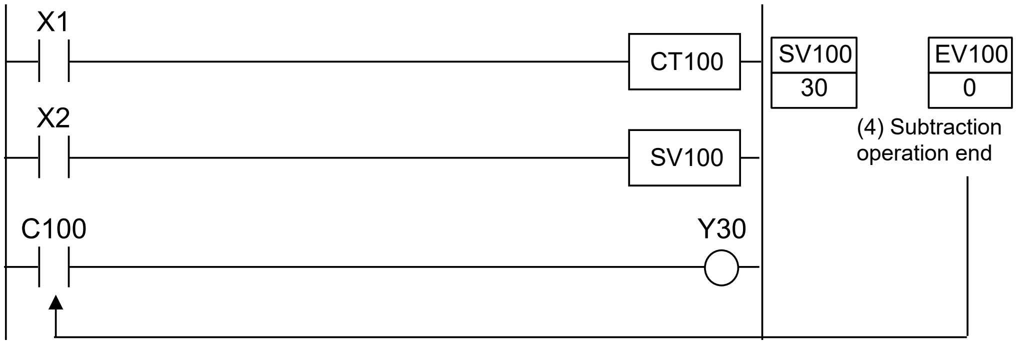
2. When the reset input falls, the value in the set value area "SV" is preset in the elapsed value area "EV" .



3. Each time the count input X1 turns ON, the value in the elapsed value area "EV" is subtracted.



4. When the value in the elapsed value area "EV" reaches zero, the counter contact "C" with the same number turns ON.

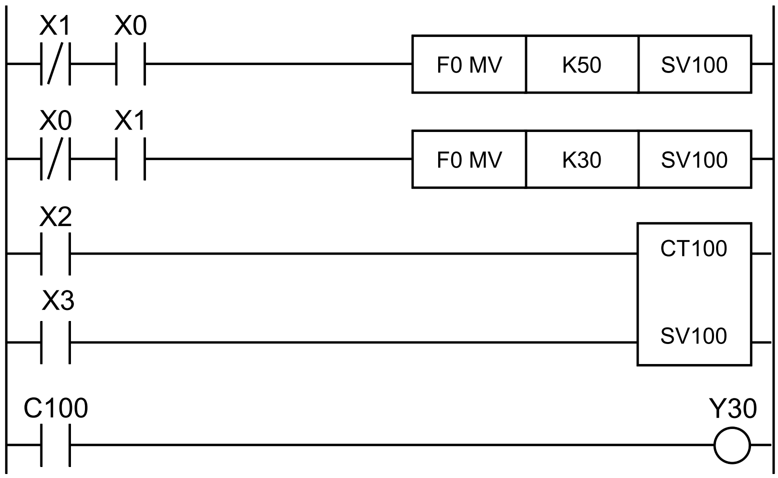


■ **Examples of applying direct specification of set value area numbers**

Example 1) Changing set values based on specified conditions

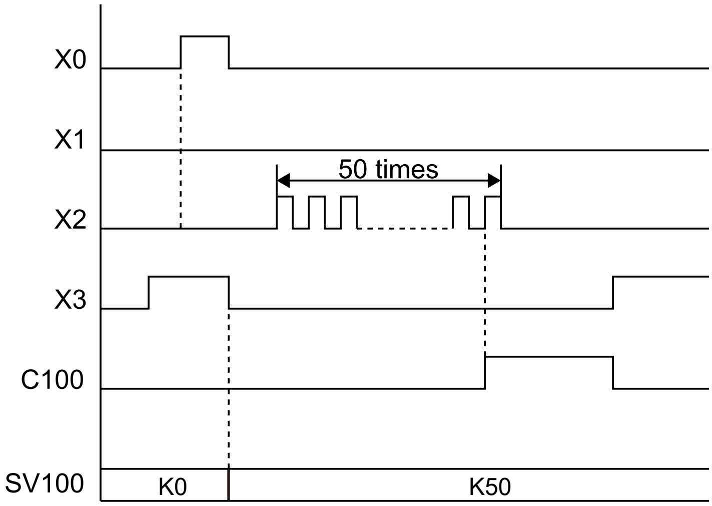
The set value is K50 when X0 is ON and K30 when X1 is ON.

- **Ladder diagram**



- **Timing chart**

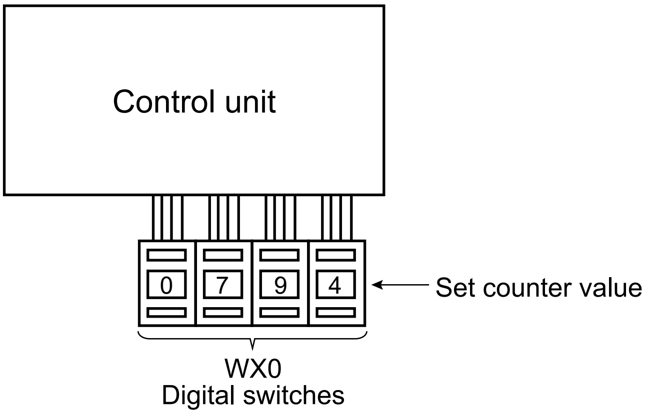
  Example when X0 turns ON.



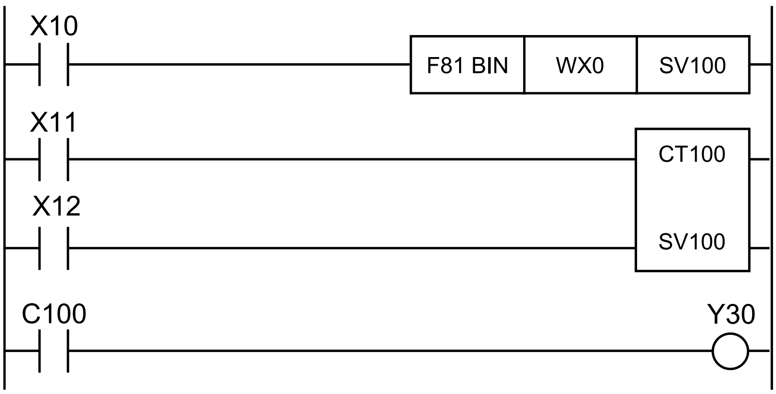
Example 2) Setting a set value from external digital switches

The BCD data of the digital switches connected to X0 through XF is converted and becomes the set value.

- **Connection example**

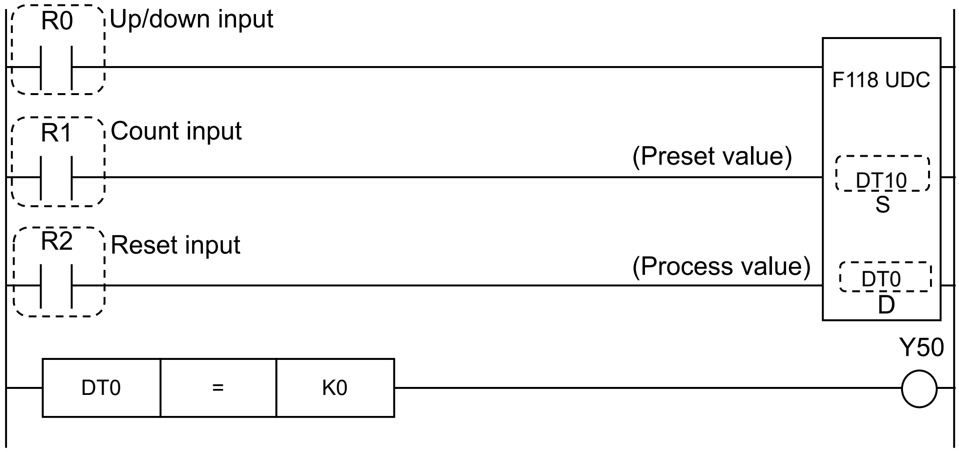


- **Ladder diagram**



3.5 F118 UDC (Up/Down Counter)

■ Instruction format



■ Instruction list

Instruction	Description
S	Area storing preset values, or constant data
D	Up/down counter elapsed value area

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●		●	●	●	●				
D		●	●	●	●	●	●	●									

■ Outline of operation

- This is a counter that switches between incremental counting (addition) and decremental counting (subtraction) depending on whether the relay specified by the up/down input is ON or OFF.
- 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 area specified by [D].
- When the reset input is switched from ON to OFF, the preset value of [S] is transferred to [D]. The count range is K–32,768 (H8000) to K32,767 (H7FFF).
- When the count input is changed from OFF to ON (with reset input in an OFF state), the count operation is performed with the value set in [D] as the default value.
- When the reset input turns ON, the elapsed value area of [D] is cleared.
- The count result can be determined by comparing the elapsed value of [D] with the specified setting value by using the data comparison instruction.
- Execute the data comparison instruction immediately after the F118 UDC instruction.

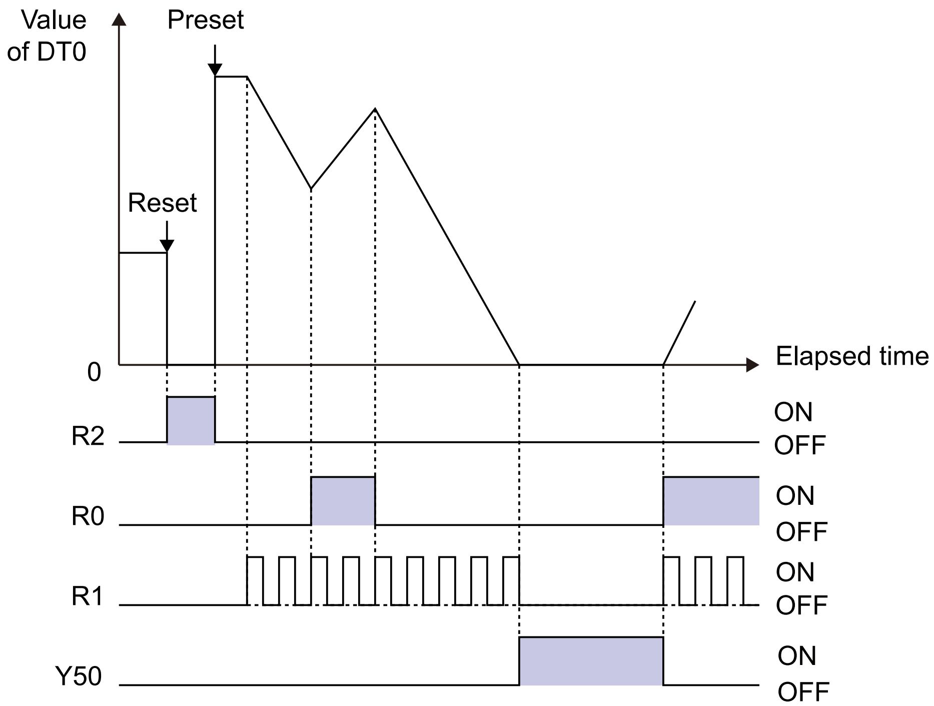
■ Operation example

Operation of instruction format description program

The program on the previous page is an example in which the default value is set, and external output Y50 turns ON when target value is 0.

This can be used, for example, in programs such as those that cause an indicator lamp to light when the work being added or subtracted reaches a certain quantity.

1. When reset input R2 switches from ON to OFF, the DT10 value is written to DT0. This value is the target value.
2. If count input R1 is ON when R0 turns OFF, the DT0 value is decremented by 1 (decremental counting). If count input R1 is ON when R0 turns ON, the DT0 value is incremented by 1 (incremental counting).
3. As a result of work being added or subtracted, the counter elapsed value area DT0 value is compared with K0, and if DT0 is equal to K0, external output Y50 turns ON.



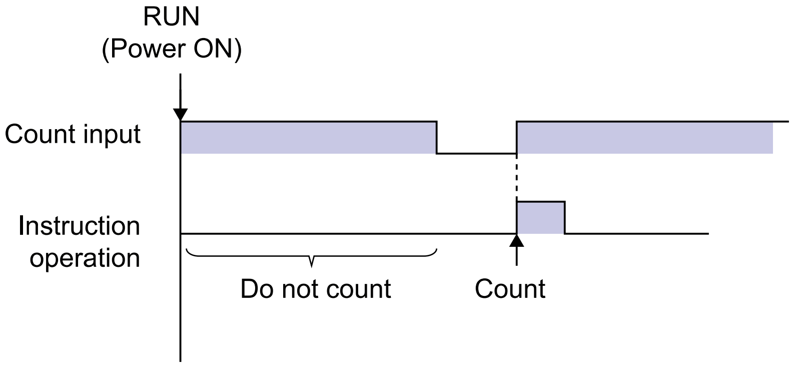
■ Precautions for programming

- 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 combining the F118 UDC instruction with an AND stack instruction or a POP stack instruction, be careful that the programming is correct.

■ Cautions on detecting the count input

With the F118 UDC instruction, the increment or decrement occurs 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.
- When switching to RUN or when powering on in "RUN mode" , if the count input is ON from the beginning, increment/decrement is not carried out for the first scan.

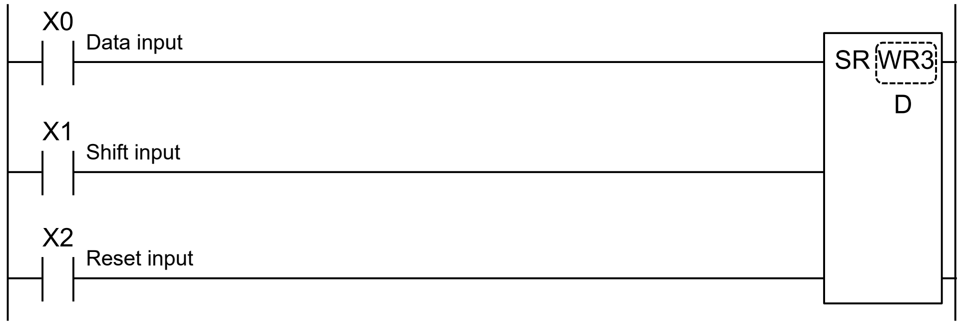


- Be aware that, if used with instructions that change the order of execution such as the MC to MCE instructions or the JP to LBL instructions (see 1 to 6 below), the operation of instructions may change depending on the timing of instruction execution and count input.

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

3.6 SR (Shift Register)

■ Instruction format



■ Instruction list

Instruction	Description
D	Specified register

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D			•														

■ Outline of operation

- An instruction that moves (shifts) the content of the specified register WR (16-bit unit) one bit to the left.

1. When shift input turns ON (rises), the contents of WR is shifted one bit to the left
2. When shifting, the empty bit (least significant bit) is set to 1 if data input is ON or 0 if OFF.
   When shift input turns ON, this instruction operates as shown in the figure below.

3. When reset input is ON, the content of the specified register is cleared.

WR3															
0	0	0	0	1	0	0	0	1	0	0	0	1	1	0	0

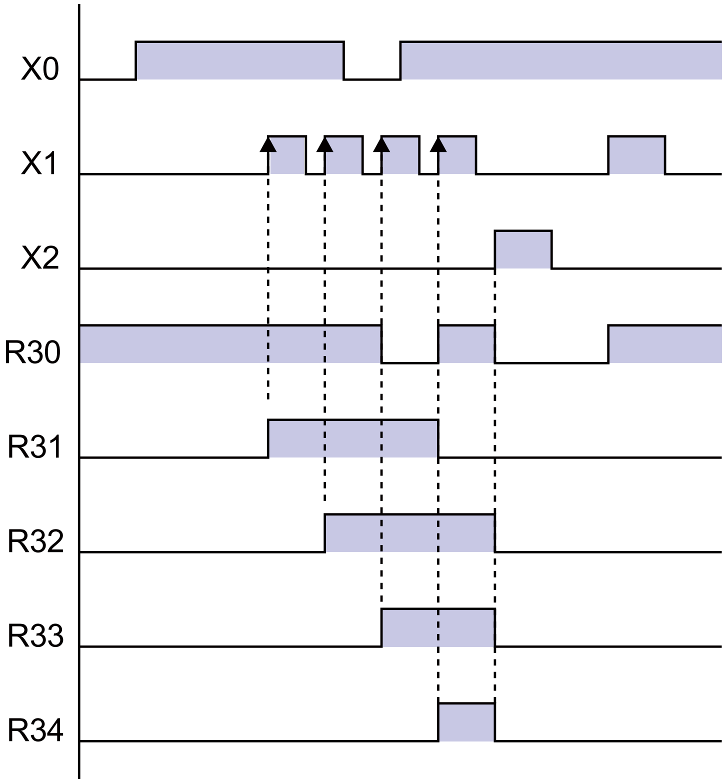
⇩ Clear

WR3															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

■ Operation example

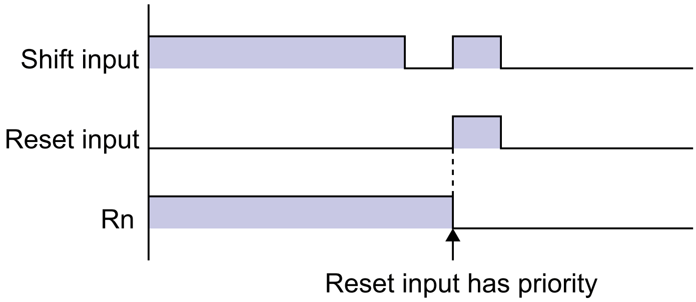
Operation of instruction format description program

1. If X1 turns ON when X2 is in an OFF state, the content of WR3 (internal relays R30 to R3F) is shifted one bit to the left.
2. The bit left empty by the left shift (R30) is set to 1 when X0 is ON and 0 when OFF.
3. When X2 turns ON, the content of WR3 is reset to 0.



■ Precautions for programming

- The **SR** instruction requires data input, shift input, and reset input.
- When reset input and shift input rise simultaneously, reset input is prioritized.

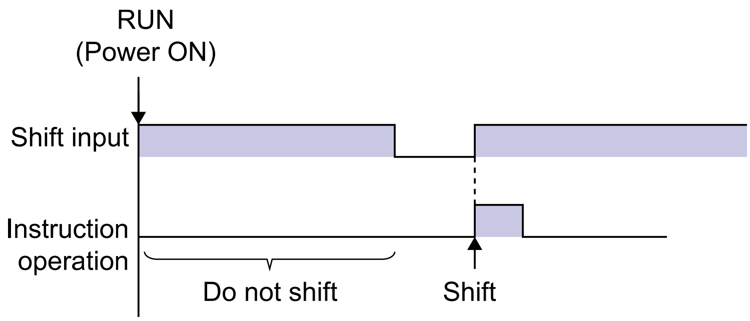


- Note that when a hold type memory area is specified for the shift register, an automatic reset is not performed when the power supply is turned ON.
- When combining a shift register instruction with an AND stack instruction or pop stack instruction, make sure that the syntax is correct.

■ Precautions for shift input detection

The SR instruction performs a shift when an OFF to ON rise is detected.

- If the shift input remains continuously ON, a shift will only take place at the rise. No further shifts will take place.
- In cases where the shift input is initially ON, such as when the mode is switched to RUN or when the power is turned on when in"RUN mode", a shift will not take place at the first scan.



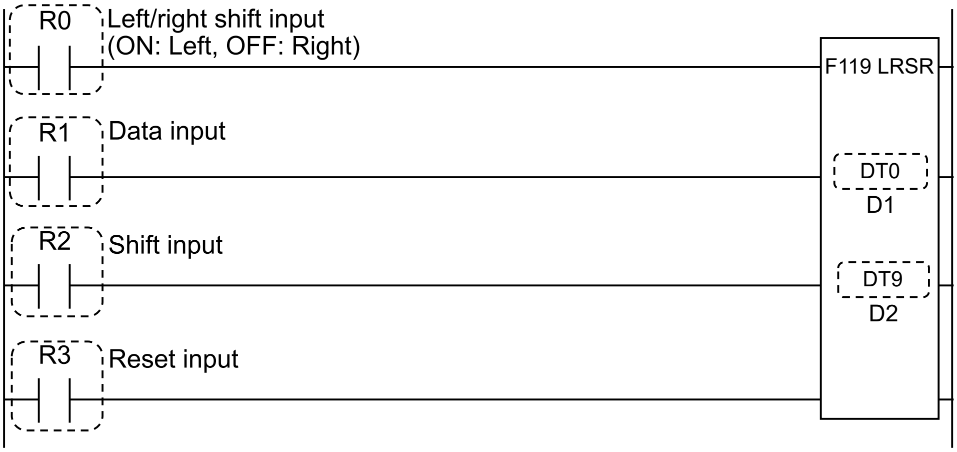
- Be aware that, if used in combination with instructions (see below, 1. to 6.) that change the order of execution of instructions such as the MC to MCE instructions or the JP to LBL instructions, depending on the execution of the instruction and the shift input timing the instruction operation changes.
  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

■ Related instructions

In addition to this instruction, there is also a left/right shift register (F119 LRSR). The same type of operation can be implemented using data shift instructions (F100 SHR to F113 WBSL) or data rotate instructions (F120 ROR to F123 RCL).

3.7 F119 LRSR (Left/Right Shift Register)

■ Instruction format



■ Instruction list

Instruction	Description
D1	Starting number of area to be shifted
D2	End number of area to be shifted

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D1		●	●	●	●	●	●	●									
D2		●	●	●	●	●	●	●									

■ Outline of operation

- This shift register changes direction, either left (direction of most significant bit) or right (direction of least significant bit), in which a shift of one bit is made based on the ON/OFF status of the relay specified by 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 area for both [D1] and [D2]. Additionally, specify values so that [D1] is equal to or less than [D2].
- The following operation is performed.
  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.

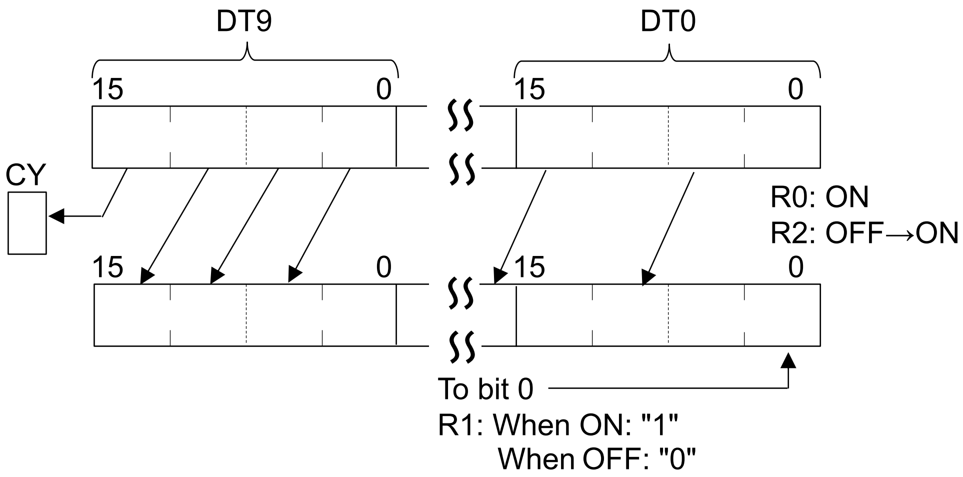
Also, the bit extracted by the shift (the most significant bit for a shift to the left, and the least significant bit for a shift to the right) will be set for the special internal relay R9009 (carry flag).

3. If the reset input is ON, the contents of the specified area are cleared to 0.

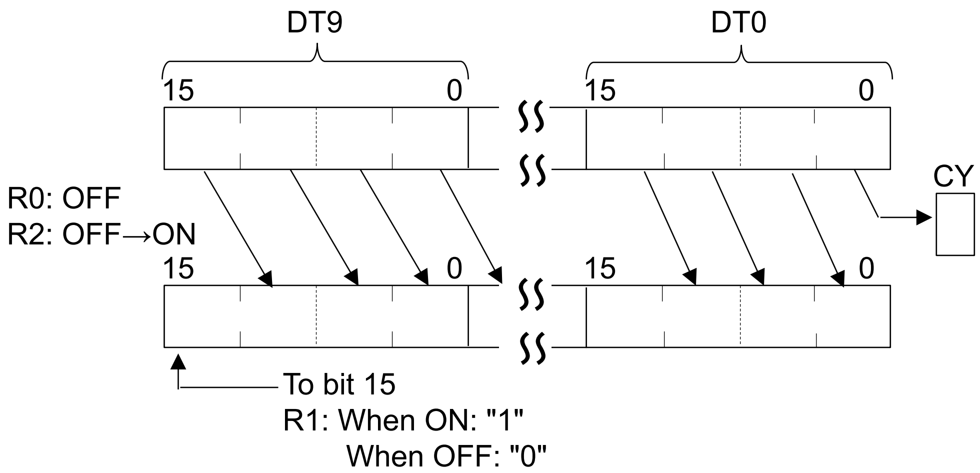
■ Operation example

Operation of instruction format description program

Left shift



Right shift

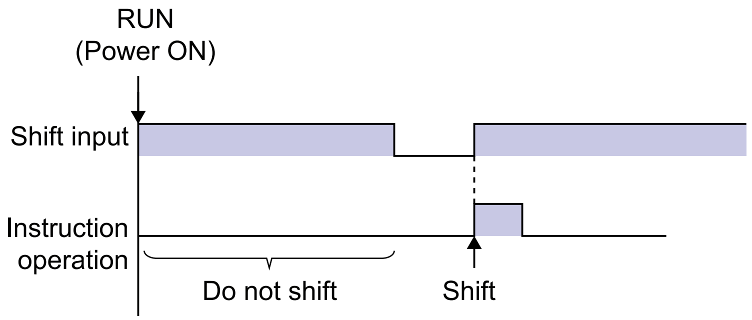


■ Precautions for shift input detection

In the F119 LRSR instruction, shift takes place when the OFF > ON rise of the shift input is detected.

- If the shift input remains continuously ON, a shift will only take place at the rise. No further shifts will take place.

- In cases where the shift input is initially ON, such as when the mode is switched to RUN or when the power is turned on when in "RUN mode" , a shift will not take place at the first scan.



- Be aware that, if used in combination with instructions (see below, 1. to 6.) that change the order of execution of instructions such as the MC to MCE instructions or the JP to LBL instructions, depending on the execution of the instruction and the shift input timing the instruction operation changes.
  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

■ Precautions for programming

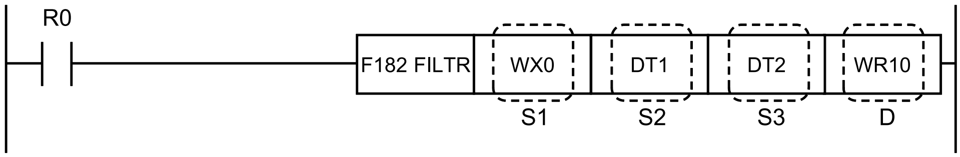
When combining the F119 LRSR instruction with an AND stack instruction or POP stack instruction, be careful that the programming is correct.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the [D1] address > [D2] address
R9009 (CY)	Turns ON when the bit extracted by the shift is "1"

3.8 F182 FILTR (Time Literal Process)

■ Instruction format



■ Instruction list

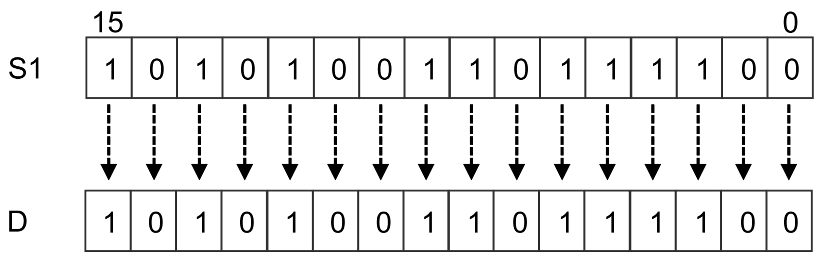
Instruction	Description
S1	Area storing the 16-bit data that is filter processing target
S2	Area storing the filter processing target bits, or constant data
S3	Area storing the filter processing time, or constant data
D	Area storing the filter processing result

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•						
S2	•	•	•	•	•	•	•	•	•	•	•	•	•				
S3	•	•	•	•	•	•	•	•	•	•	•	•	•				
D		•	•	•	•	•	•	•	•								

■ Outline of operation

In the 16-bit data in the area specified by S1, for the bits specified by S2, 0 bits are directly output and 1 bits (filter processing targets) are output after filter processing for the amount of time (0 to 30000, ms units) specified by S3 and the result is output in bit units (the bit positions are the same as for S1) to the area specified by D.



(Note 1) The bit positions of S1 and D correspond.

■ Precautions for programming

- When the execution condition rises, all input bits specified by S1 are directly output unconditionally.
- It is possible that an error of up to one scan may occur in the filter processing time.

■ Example of program execution

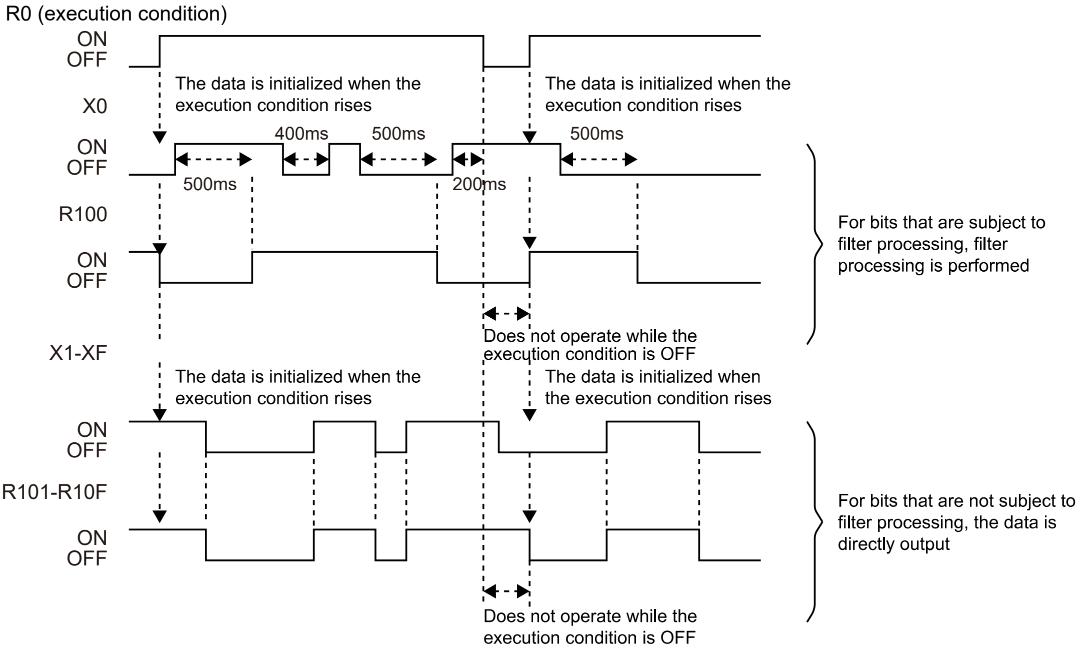
The changes in the execution condition R0 and the values of X0 to XF when the state before execution of this instruction (R0 = 0) is as follows are explained by using a time chart.

WX0 (Filter processing input data) = HA9BC

DT1 (Filter processing target bit) = H0001

DT2 (Filter processing time) = K500

WR10 (Filter processing result) = HFFFF



■ Flag operations

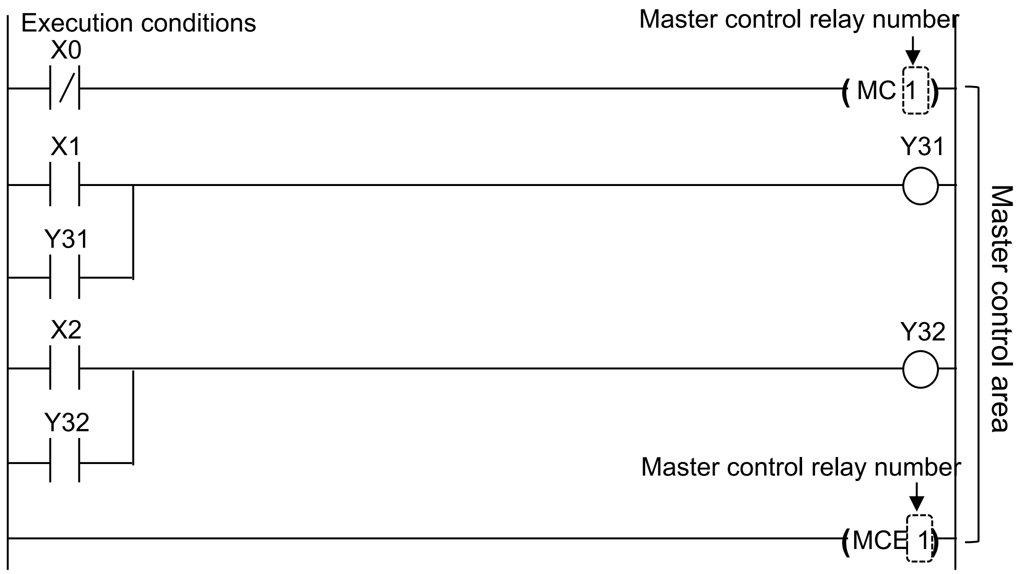
Name	Description
R9007 R9008 (ER)	When the area is exceeded in index modification
	When the filter processing time specified by S3 is outside the range of K0 to K30000

(MEMO)

4 Control Instructions

4.1 MC/MCE (Master Control Relay / Master Control Relay End)

■ Instruction format



■ Outline of operation

- Executes the program between the **MC** and **MCE** instructions when the execution condition turns ON.
- When the execution condition is OFF, the state of each I/O relay is as follows.

OT instruction	All OFF
KP instruction	Holds the state
SET instruction	Holds the state
RST instruction	Holds the state
TM instruction	Reset
CT instruction	Holds the intermediate process
SR instruction	Holds the intermediate process
Differential instruction	Refer to the following
Other instructions	Not executed

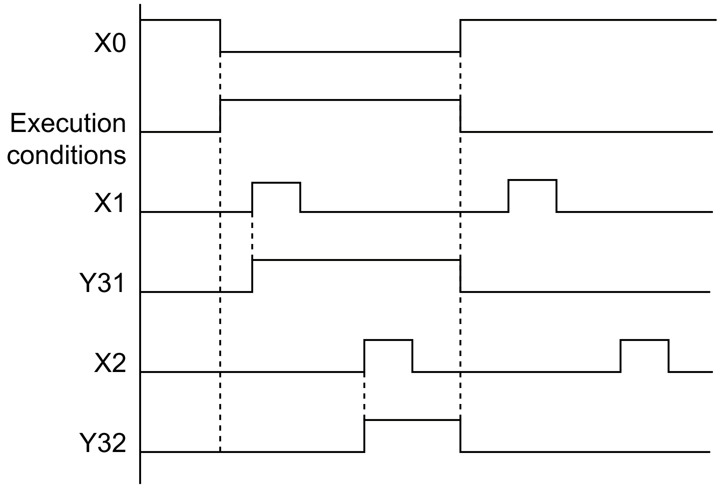
- Caution is required when using an instruction that is executed by detecting the rise of an execution condition, such as a differential instruction (1 to 7 below).

1. DF (rise differential)
2. CT (counter) count input
3. F118 UDC (up-down counter) count input
4. SR (shift register) shift input
5. F119 LRSR (left and right shift register) shift input
6. NSTP (next step)
7. Differential execution type high-level instruction (instruction specified by P and a number)

■ Operation example

Operation of instruction format description program

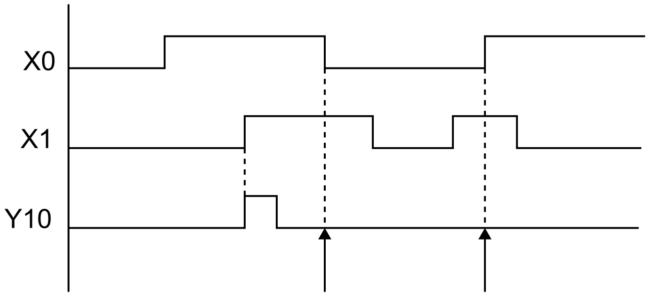
1. Executes the process between the **MC1** and **MCE1** instructions while the execution condition is ON.
2. If the execution condition is OFF, the process between the **MC1** and **MCE1** instructions is not executed and output is turned OFF.



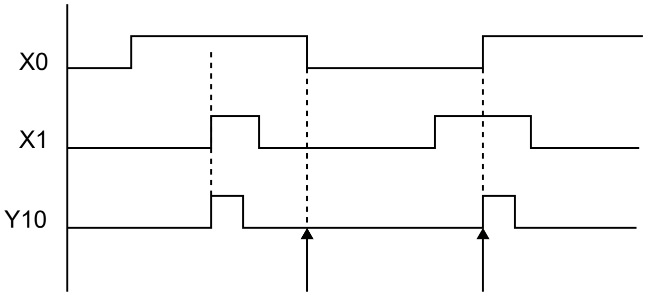
■ Operation of differential instructions between MC and MCE

- Note that if a differential instruction is used between MC and MCE, the output will vary as follows depending on the timing of the MC execution condition and the input of differential instruction.

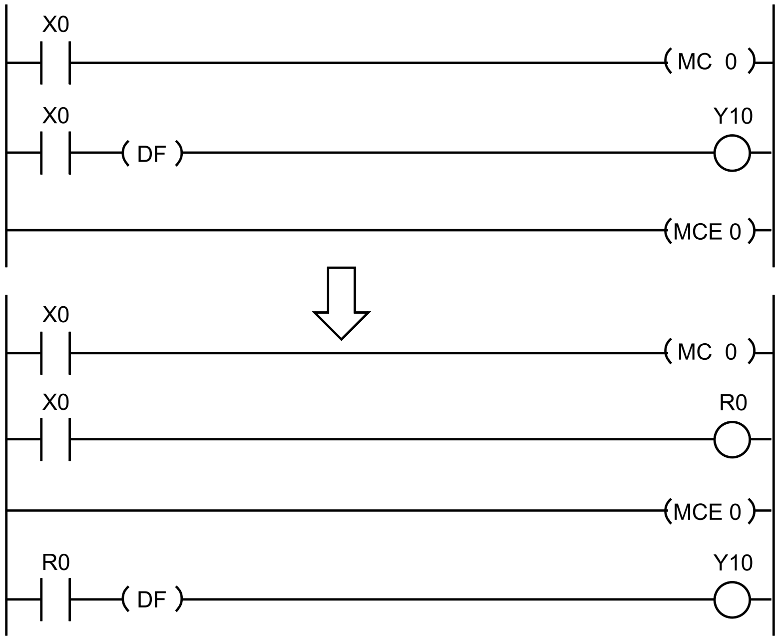
Timing chart 1



Timing chart 2

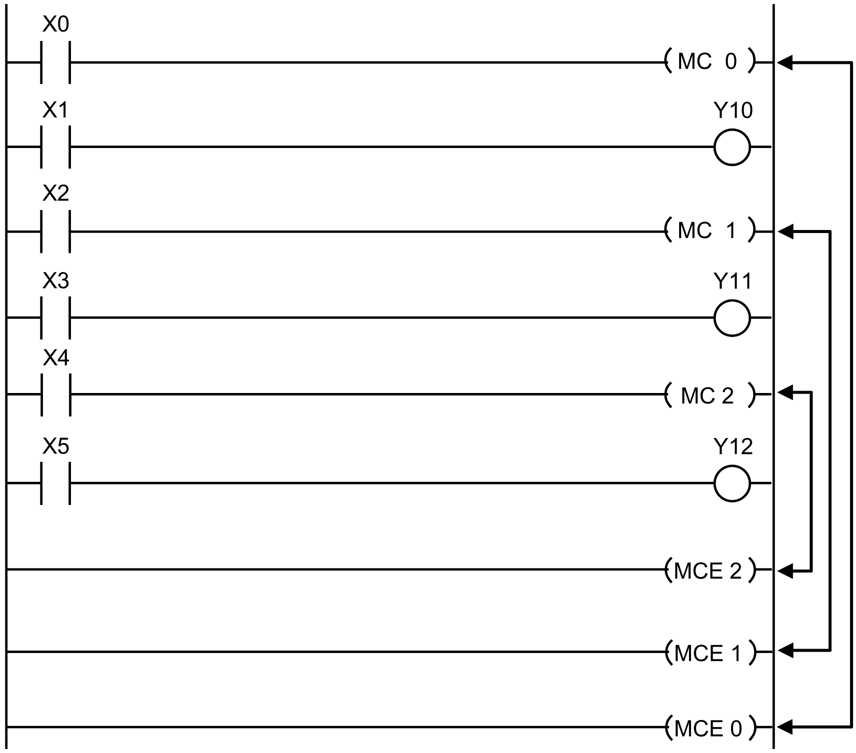


- Output will not be obtained if the same execution condition is specified for an MC instruction and a differential instruction. If output is needed, enter the differential instruction outside of the MC-MCE instruction sequence.

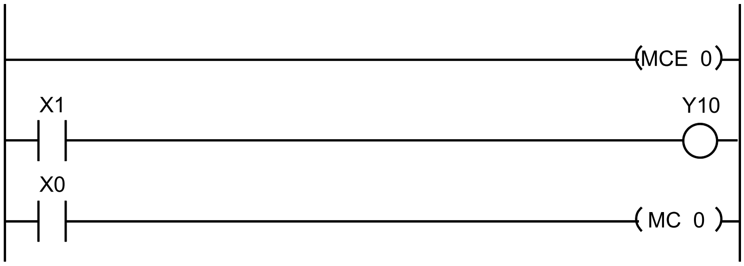


■ Precautions for programming

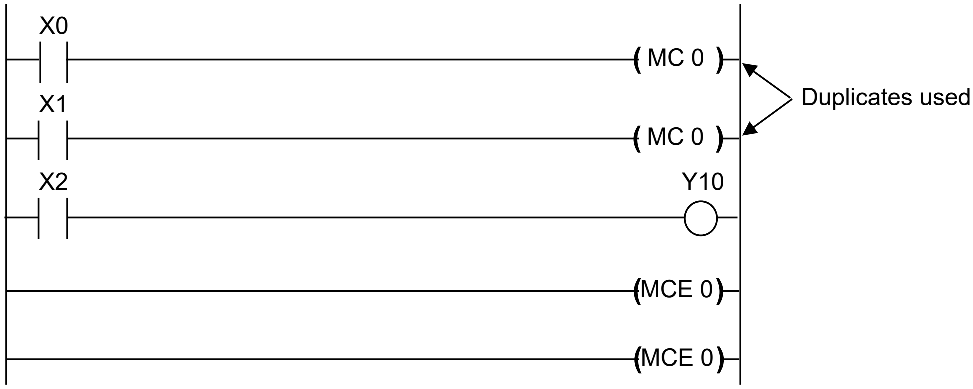
- A second **MC**-**MCE** instruction pair can be entered (nested) between an initial **MC**-**MCE** instruction pair. (There is no limit to the number of nestings.)



- The program cannot be executed in the following cases.
  1. Either **MC** or **MCE** is missing.
  2. The order of **MC** and **MCE** is reversed.

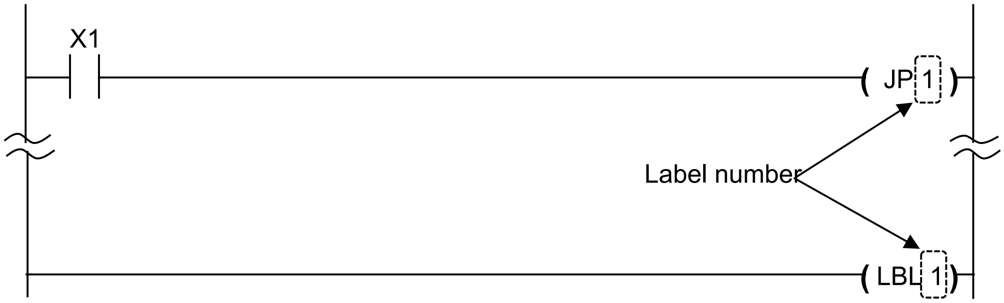


  3. There is duplicated use of the specified number.



4.2 JP/LBL (Jump/Label)

■ Instruction format



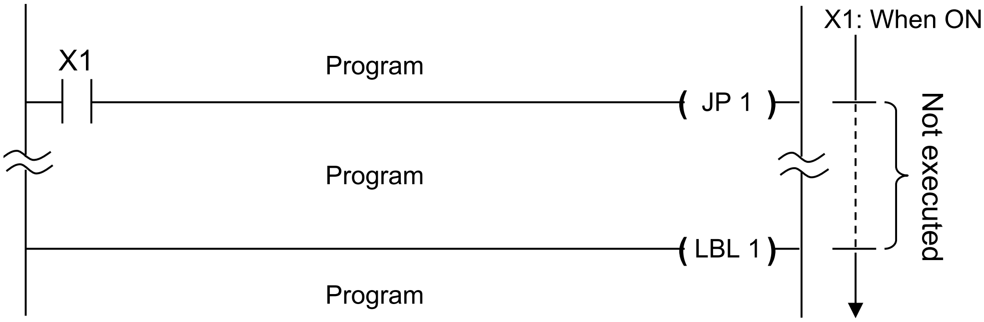
■ Outline of operation

- When the execution condition turns ON, the program jumps to the label (**LBL** instruction) with the same number as the specified number.
- Program execution continues from the next instruction after the jump destination label.

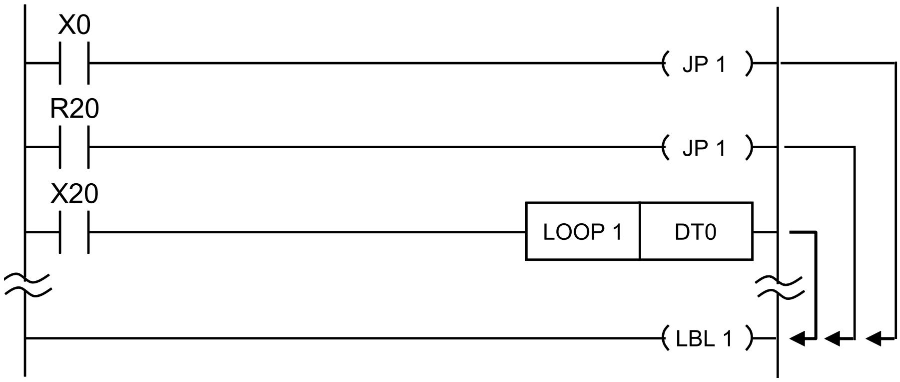
■ Operation example

Operation of instruction format description program

When execution condition X1 turns ON, the program jumps to label 1.



- The same label is used by the **JP** instruction and the **LOOP** instruction. Any instruction can be used as the starting point for the jump destination.
- It is possible to use **JP** instructions with the same label number multiple times.



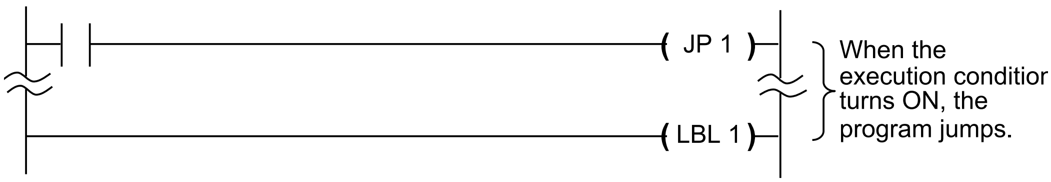
- 2 or more **LBL** instructions with the same number cannot be written in the same program.
- If the jump destination label is not programmed, a syntax error occurs.
- Caution is required when using an instruction that is executed by detecting the rise of an execution condition, such as a differential instruction (1 to 7 below).
  1. DF (rise differential)
  2. CT (counter) count input
  3. F118 UDC (up-down counter) count input
  4. SR (shift register) shift input
  5. F119 LRSR (left and right shift register) shift input
  6. NSTP (next step)
  7. Differential execution type high-level instruction (instruction specified by P and a number)

■ Precautions for programming

- If the label is written to an address before the **JP** instruction, be aware that there is a possibility that the scan cannot be completed, and an operation bottleneck error will occur.
- **JP** and **LBL** instructions cannot be used in a step ladder area (the range from **SSTP** to **STPE**).
- It is not possible to jump from a main program to a subprogram (a subroutine or interrupt program after the **ED** instruction), from a subprogram to a main program, or from a subprogram to another subprogram.

■ Operation of TM, CT, and SR instructions between JP and LBL instructions

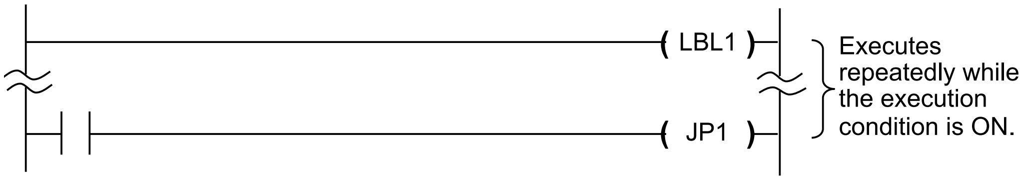
- If the **LBL** instruction is at an address after the **JP** instruction, then processing of each instruction when executing the **JP** instruction will be as follows.



(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.

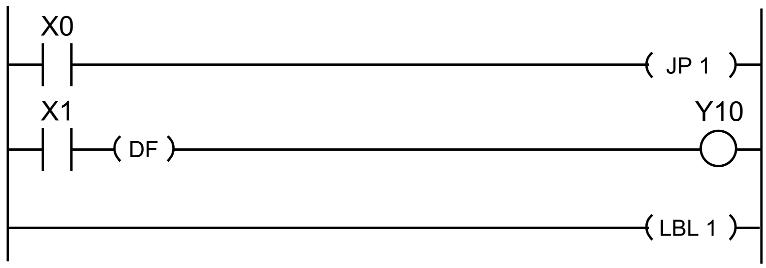
- If the **LBL** instruction is at an address before the **JP** instruction, then processing of each instruction when executing the **JP** instruction will be as follows.



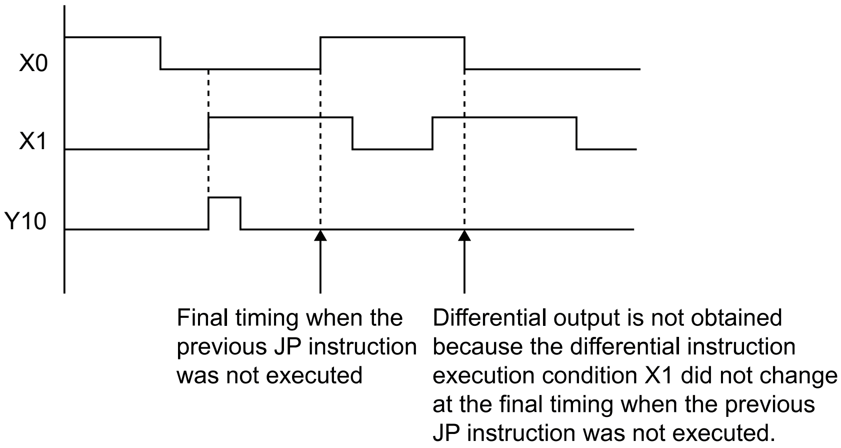
(1)	TM instruction	Multiple timings occur during a single scan, therefore the time cannot be guaranteed.
(2)	CT instruction	If the state of the count input does not change during the scan, it will operate in the usual way.
(3)	SR instruction	If the state of the shift input does not change during the scan, it will operate in the usual way.

■ Operation of a differential instruction between JP and LBL

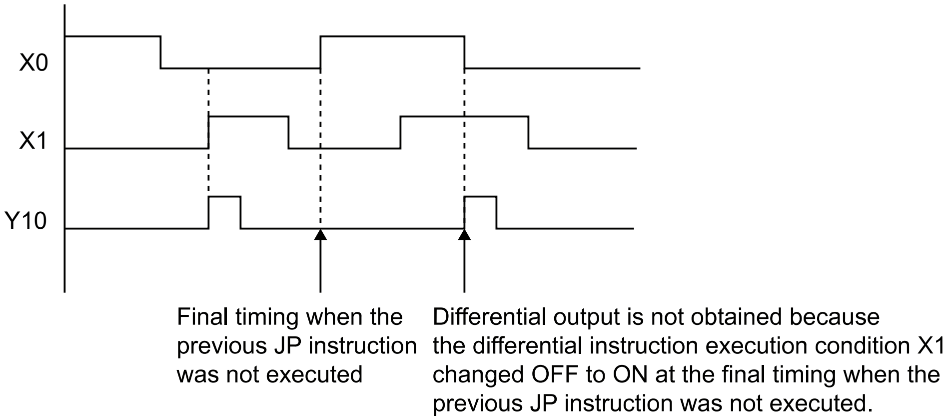
- If a differential instruction is used between a JP and LBL instruction, be aware that the obtained output will differ as shown below depending on the execution condition of the JP and the input timing of the differential instruction.



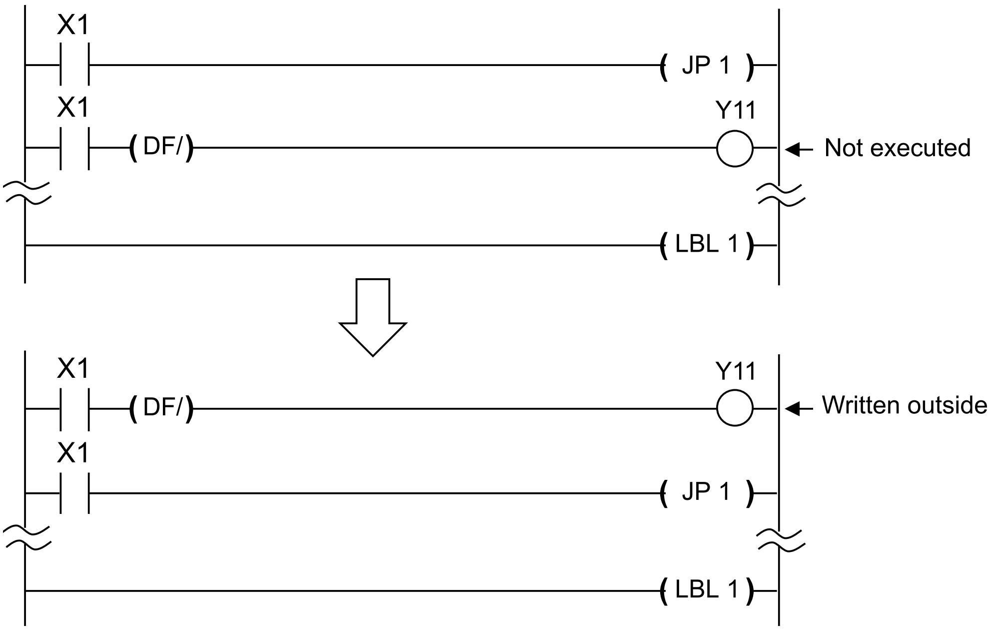
Timing chart 1



Timing chart 2

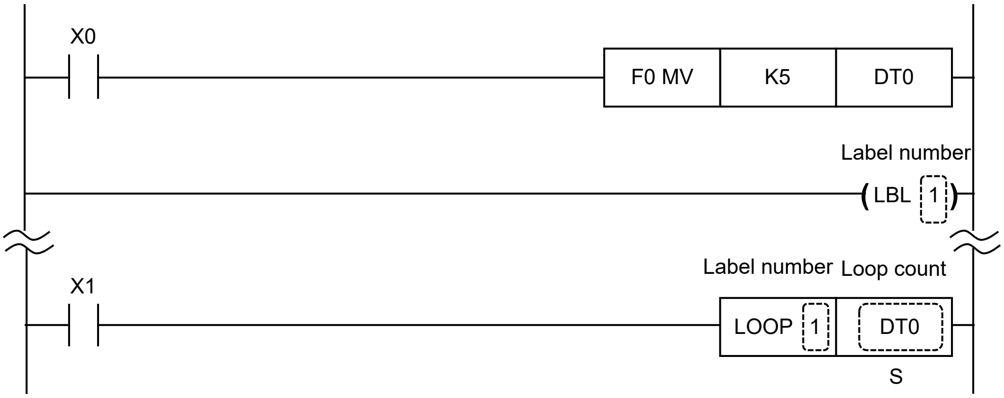


- When the execution conditions for the **JP** instruction are the same as the execution conditions for the differential instruction, the leading edge (or trailing edge) of the execution condition for the differential instruction will not be detected. If differential output is required, write the differential instruction outside of the area between the **JP** and **LBL** instructions.



4.3 LOOP, LBL (Loop, Label)

■ Instruction format



■ Instruction list

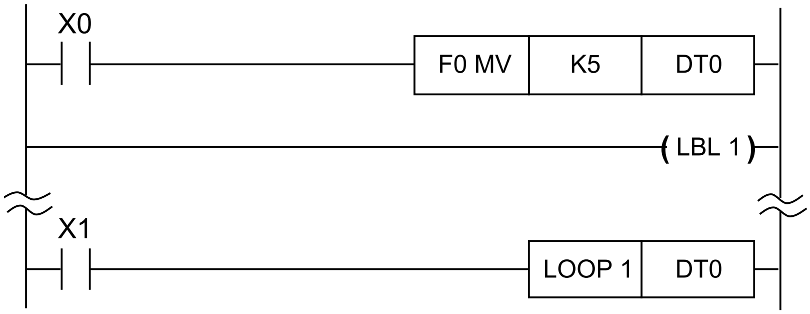
Instruction	Description
S	Area storing number of loop operations

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S		•	•	•	•	•	•	•	•								

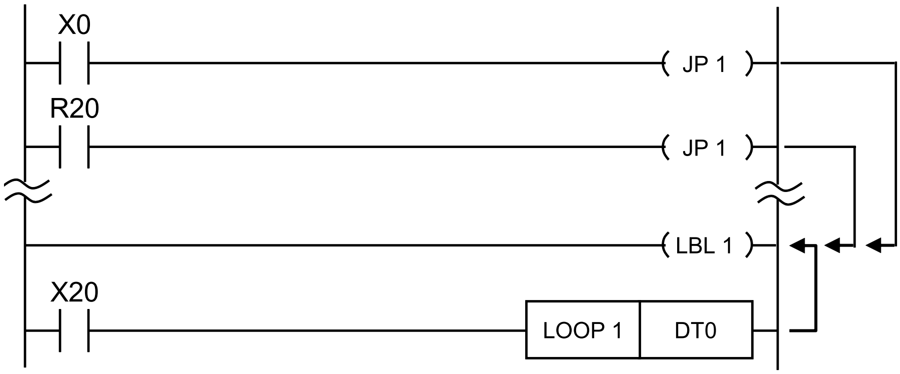
■ Outline of operation

- When the execution condition turns ON, 1 is subtracted from the content of [S] and if the result does not equal 0, the operation jumps to the label (**LBL** instruction) with the same number as the specified number.
- Program execution continues starting from the instruction of the label at the jump destination.
- The **LOOP** instruction is used to set the number of times to execute the program. When the number of times (K constant) specified by [S] is reached, the operation does not jump even if the execution condition is established.



If DT0=K5, then after 5 jumps, there are no more jumps even if X1 is ON.

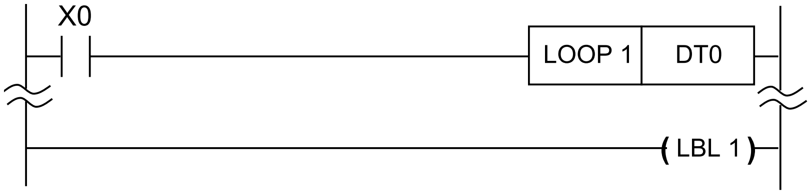
- If the memory area content specified by [S] is 0 from the start, the operation does not jump to a label number, and the next processing is performed.
- The same label is used by the **JP** instruction and the **LOOP** instruction. A label can be used as the jump destination for any instruction, as many times as required.



- Two or more**LBL**instructions with the same number cannot be written in the same program.
- If the jump destination label is not programmed, a syntax error occurs.

■ Operation of TM, CT, and SR instructions between LOOP and LBL instructions

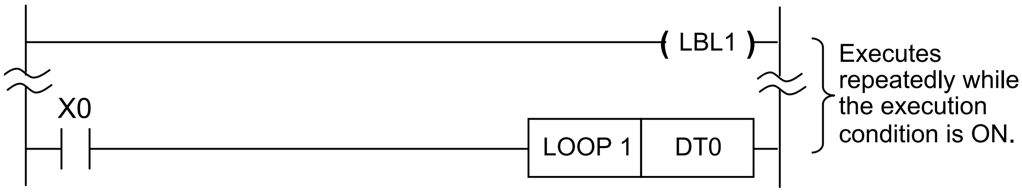
- If the **LBL** instruction address is after that of the **LOOP** instruction, the TM, CT, and SR instructions are processed as follows when the **LOOP** instruction is executed.



When the executio condition turns ON the program jumps

(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 address is before that of the **LOOP** instruction, the TM, CT, and SR instructions are processed as follows when the **LOOP** instruction is executed.

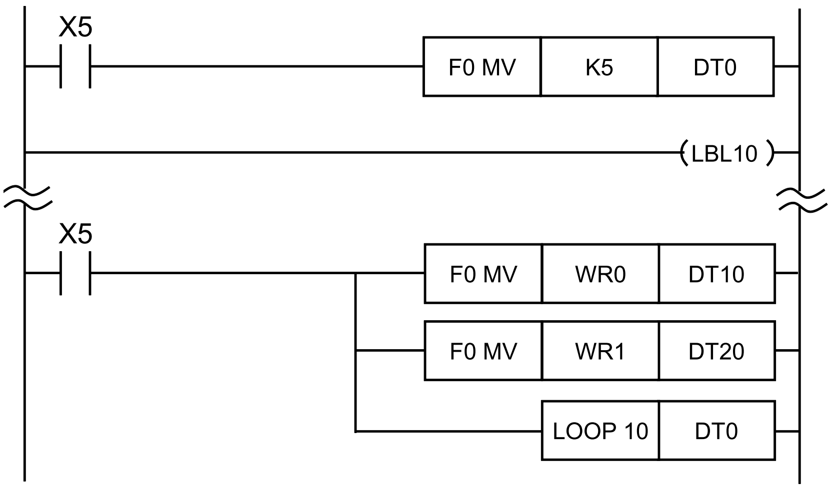


(1)	TM instruction	Multiple timings occur during a single scan, therefore the time cannot be guaranteed.
(2)	CT instruction	If the state of the count input does not change during the scan, it will operate in the usual way.
(3)	SR instruction	If the state of the shift input does not change during the scan, it will operate in the usual way.

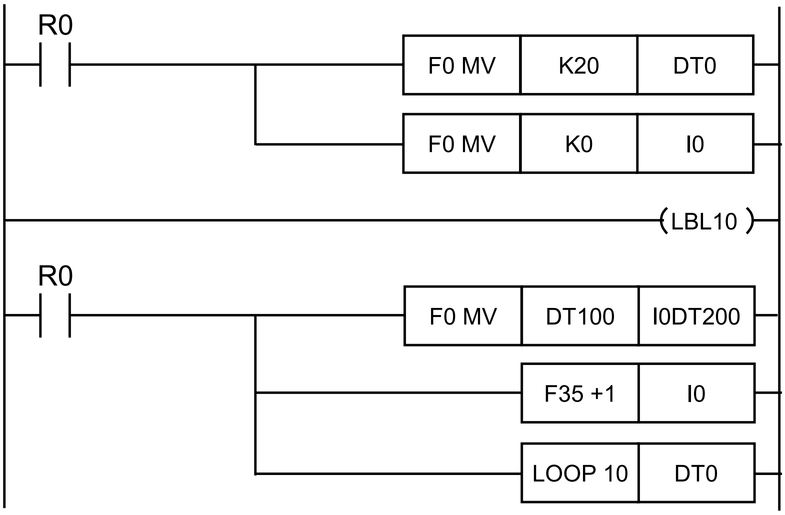
■ Precautions for programming

- If the label is written to an address before the **LOOP** instruction, be aware of the following points.
  1. Ensure that the instruction for setting the loop count is written before **LBL** to **LOOP**. See the "P.4-12" program.
  2. Write each instruction repeatedly executed between **LBL** to **LOOP** so that they are executed under the same conditions as the **LOOP** instruction.
  3. During this repetition, it is possible that a single scan will exceed the operation bottleneck monitoring time and an operation bottleneck error may occur.

Example 1: When X5 turns ON, two F0 MV instructions are repeated five times.



Example 2: The DT100 value is transferred to DT200 to DT219.



- The **LOOP** instruction and **LBL** instruction cannot be used in the step ladder area (**SSTP** to **STPE** range).
- It is not possible to jump from a main program to a subprogram (a subroutine or interrupt program after the **ED** instruction), from a subprogram to a main program, or from a subprogram to another subprogram.
- Caution is required when using an instruction that is executed by detecting the rise of an execution condition, such as a differential instruction (1 to 7 below).
  1. DF (rise differential)
  2. CT (counter) count input
  3. F118 UDC (up-down counter) count input
  4. SR (shift register) shift input
  5. F119 LRSR (left and right shift register) shift input
  6. NSTP (next step)
  7. Differential execution type high-level instruction (instruction specified by P and a number)

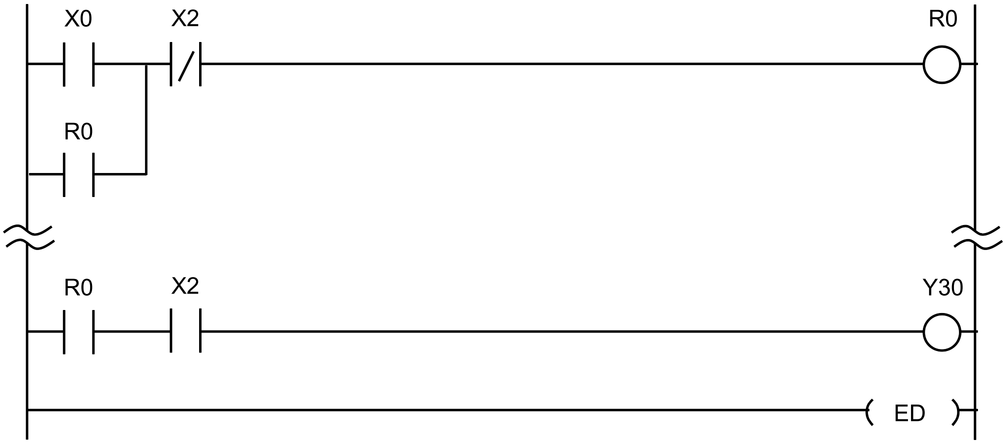
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the content of [S] is a negative value (the most significant bit is 1)

4.4 ED (End)

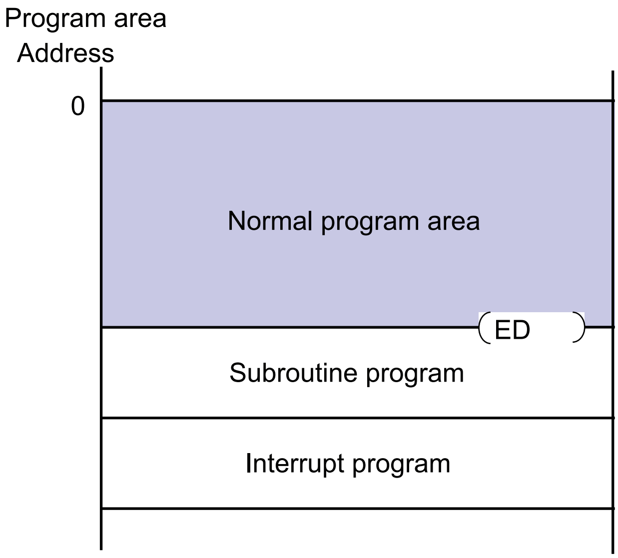
Indicates the end of a regular program area.

■ Instruction format



■ Outline of operation

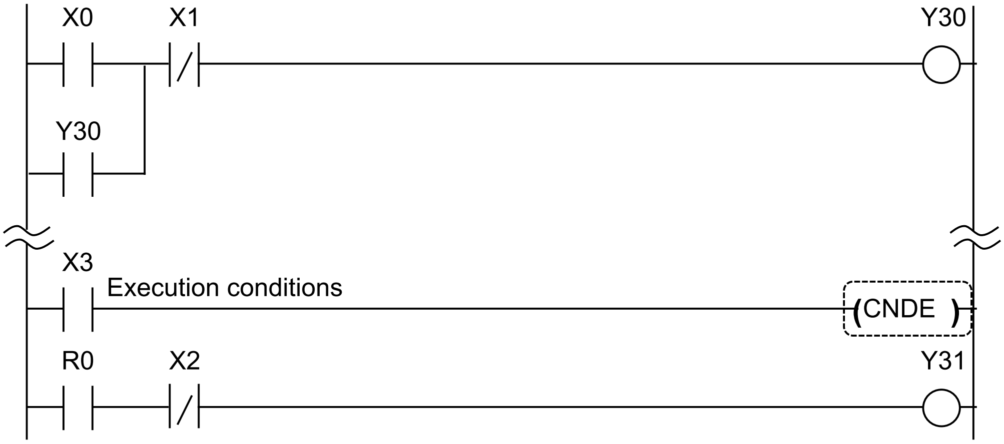
- Write the **ED** instruction at the end of the regular program area.



- Program areas are divided into the regular program area (main program) and "subroutine" and "interrupt program" areas (subprograms) using this instruction.
- Write subroutine programs and interrupt programs after the **ED** instruction.

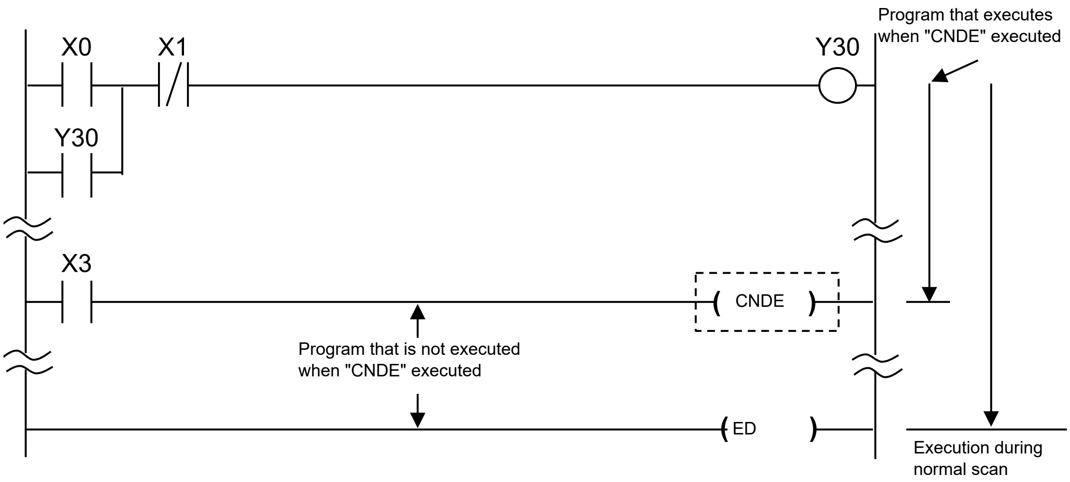
4.5 CNDE (Conditional End)

■ Instruction format



■ Outline of operation

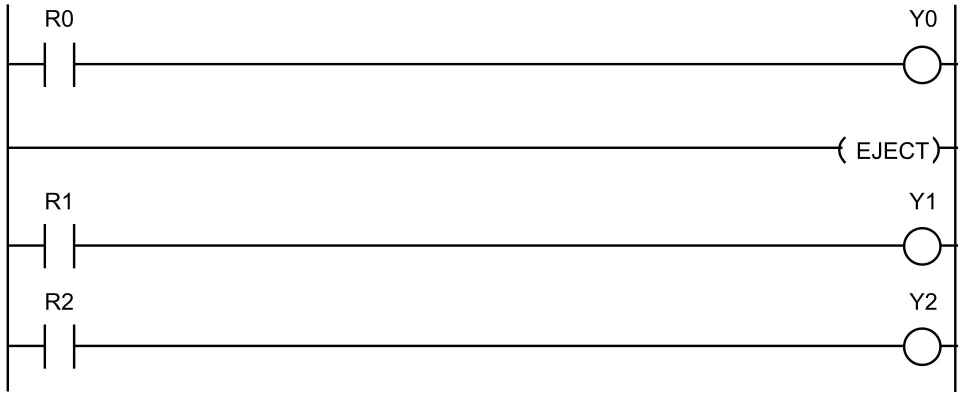
- Ends arithmetic processing of the program at the specified address.
- When the execution condition turns ON, arithmetic processing of the program ends, and processing such as input and output is performed. When processing is complete, the operation returns to the starting address.
- The processing timing can be adjusted by performing the processing only after the required number of program scans are completed.
- The **CNDE** instruction is not available in a subprogram such as a subroutine or interrupt program. Use in the main program area.
- The **CNDE** instruction can be described any number of times in the main program.
- Caution is required when using an instruction that is executed by detecting the rise of an execution condition, such as a differential instruction (1 to 7 below).
  1. DF (rise differential)
  2. CT (counter) count input
  3. F118 UDC (up-down counter) count input
  4. SR (shift register) shift input
  5. F119 LRSR (left and right shift register) shift input
  6. NSTP (next step)
  7. Differential execution type high-level instruction (instruction specified by P and a number)



X0
X1
Y30
Y30
Program that executes when "CNDE" executed
X3
CNDE
Program that is not executed when "CNDE" executed
ED
Execution during normal scan

4.6 EJECT

■ Instruction format



■ Outline of operation

- When printing out a program created using tool software, a page break occurs at the location at which this instruction is inserted.
- As with NOP instructions, no processing is performed in the program.

■ Operation example

Operation of instruction format description program

When printing out a created program, insert an EJECT instruction in the address where you would like a page break.

In the example above, a page break occurs at address 2.

5 Step ladder Instructions

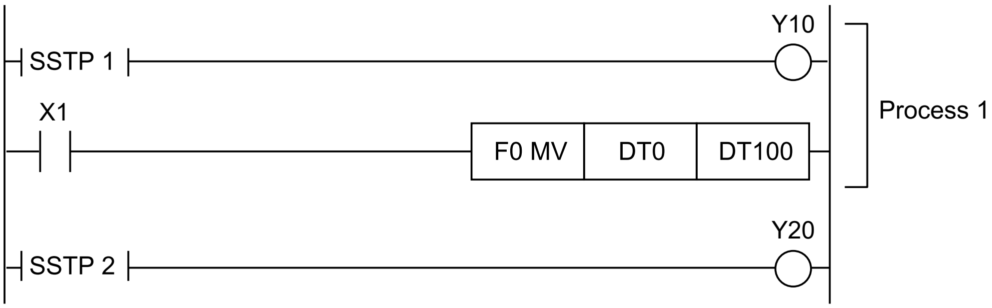
5.1 SSTP, NSTL (NSTP), CSTP, STPE (Start Step, Next Step, Clear Step, Step End)

■ Instruction format

■ Outline of operation

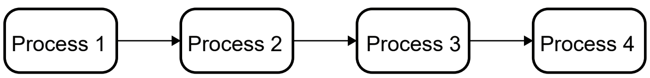
- When the **NSTL** or **NSTP** instruction is executed, the process of the specified number starting from the **SSTP** instruction is started and executed.
- The program from the **SSTP** instruction to the next **SSTP** or **STPE** instruction is considered one process.

<Example>



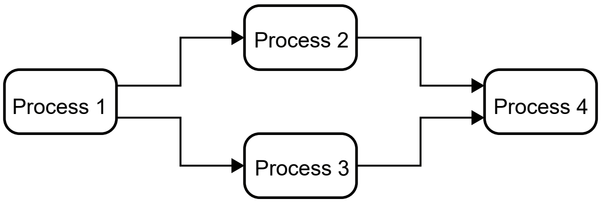
- These instructions make it easy to execute sequence control, selection branch control, parallel branch merge control, and similar operations.

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

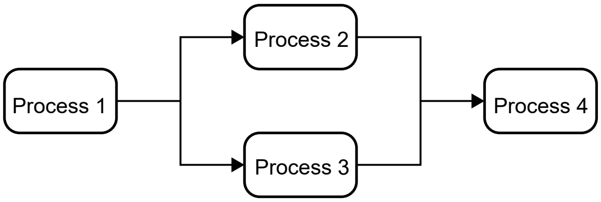


2. Selection branch control

The processes are selected and executed according to conditions.



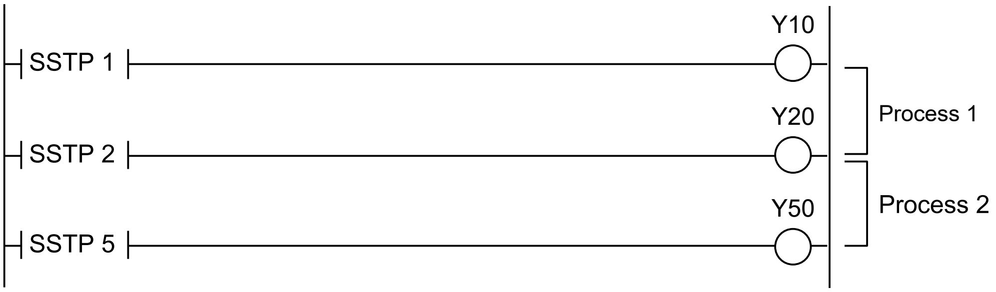
3. Parallel branch merge control
   - Multiple processes are executed simultaneously.
   - After each process is completed, the next process is executed.



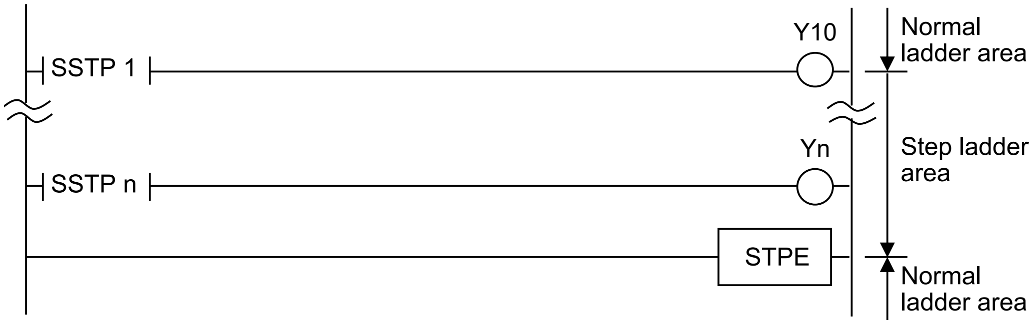
■ Syntax of step ladder instruction

SSTP start step

- This instruction indicates the "start of process n". Be sure to write "SSTP n" at the beginning of the process n program.



- Process n is defined as being from one "SSTP n" instruction to the next **SSTP** or **STPE** instruction.
- 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 used in a subprogram (subroutine or interrupt program).
- The area starting from the first **SSTP** instruction to the **STPE** instruction is referred to as the "step ladder area". The programs in this area are all controlled as processes. Other areas are referred to as "normal ladder areas".



- There is a special internal relay that turns ON for one scan only when a process on the step ladder starts. (R9015: step ladder initial pulse relay.) This relay can be used to process only one scan when starting a process, such as resetting a counter.

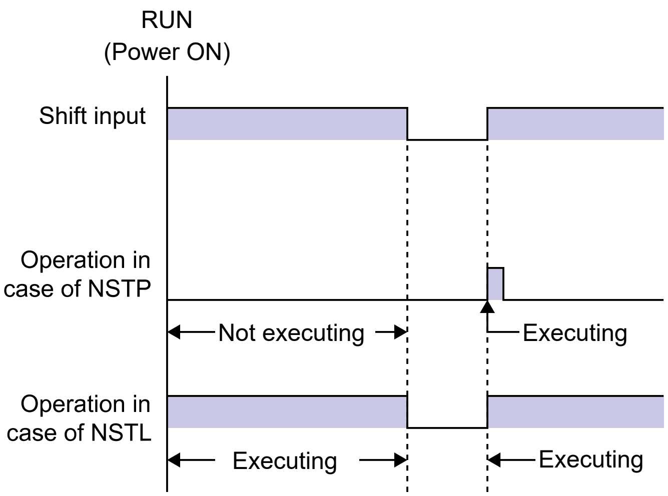
NSTL next step (every scan execution type), NSTP next step (differential execution type)

- When an **NSTL n** or **NSTP n** instruction is executed, process n specified by n is invoked.
- 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.
- A process can be started from the normal ladder area or from a process that is executing.
- 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.

Be aware that the outputs and other processes are actually turned off by the clear operation during the next scan.

- The **NSTP** instruction is a differential execution type instruction, so it is executed for only one time when the execution condition rises. Also, since it only detects if the execution condition has changed between ON and OFF, the instruction is not executed when switching to "RUN mode" or when the power is turned ON while in "RUN mode" and the execution condition is already ON.



- Be aware that, if the **NSTP** instruction is used with instructions that change the order of execution such as the MC to MCE instructions or the JP to LBL instructions (see 1 to 6 below), the operation of instructions may change depending on the instruction execution and execution condition timing.
  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
- When combining the NSTP instruction with an AND stack instruction or a POP stack instruction, be careful that the programming is correct.

CSTP clear step

When a **CSTPn** instruction is executed, process n 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.

<Example>

- A process can be cleared from the normal ladder area or from a process that is already started.

  You can use the **SCLR** (block clear) instruction to clear multiple processes at once by specifying a range.

STPE step end

Indicates the "end of the step ladder area". Be sure to write this instruction at the end of the final process. This makes the final process from **SSTP** to **STPE**.

(Note 1) In this case, process n is the final process.

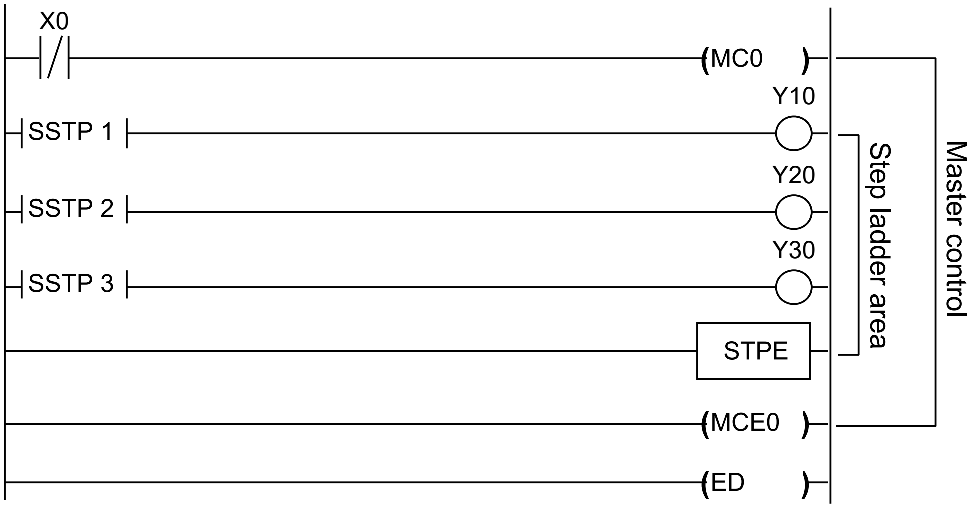
- The **STPE** instruction can only be written once, in the main program. (It cannot be written in subprograms such as subroutine programs and interrupt programs.)

■ 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 (**SUB** and **RET**) (*)
  5. Interrupt instructions (**INT** and **IRET**)
  6. **ED** instruction
  7. **CNDE** instruction

  (Note): The **CALL** instruction can be used within the step ladder area.
- To clear all processes at once, use the master control relay in the program as follows.

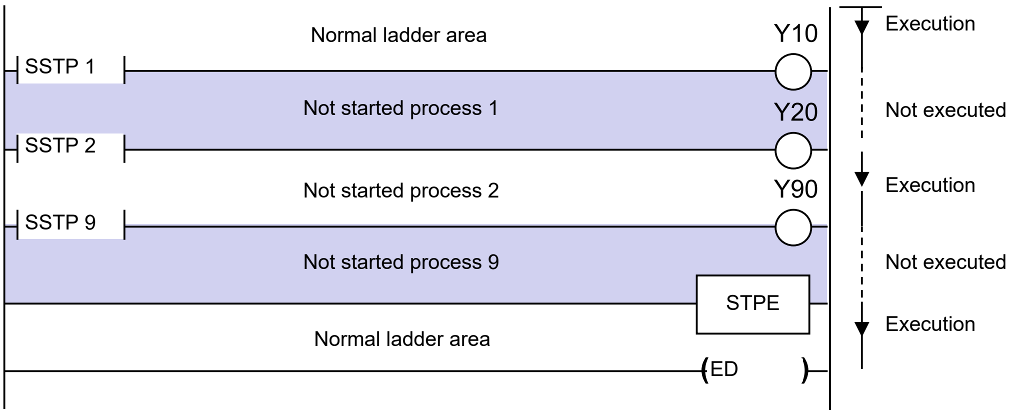
e.g. All processes are cleared when X0 turns ON



- Processes do not need to be started in numerical order. 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

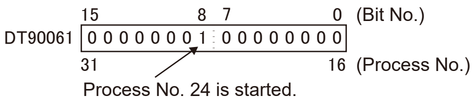
- With step ladder operations, the program in the normal ladder area and the program in the processes invoked by the next step instruction (**NSTL** or **NSTP**) are executed. The program in processes that are not executing is ignored.



When only process 2 is executing as shown in the above figure, the program in the normal ladder area and in process 2 is executed.

- When a process is started and while the first scan is being performed, the step initial pulse relay (R9015) turns ON. It turns OFF for the second and subsequent scans. This relay can be used to reset counters and shift registers.
- The starting/stopping status of processes is stored in special data registers (DT90060 to DT90067).

e.g. Starting status of process Nos. 16 to 31



■ Precautions for clearing a process

- If the next step instruction is executed in an active process, that process is automatically cleared. However, the actual clear operation does not occur until the next scan. For this reason, when a process transitions, two processes may be executing at the same time for one scan. To prevent simultaneous execution of a set of outputs that should not be ON at the same time, write an interlock into the program. (If there is a possibility of processes being simultaneously ON because of hardware response delays, take measures in the hardware processing to allow the response delay to be taken into account, even if the program includes an interlock.)

<Example>

- When a process is cleared, the operation of each instruction used in that process is as follows.

OT instruction	All OFF
KP instruction	Holds the state
SET instruction	Holds the state
RST instruction	Holds the state
TM instruction	Resets the elapsed value and timer contact output
CT instruction	Holds the intermediate process
SR instruction	Holds the intermediate process
Differential instruction	Holds the state of the execution condition (Note 1)
Other instructions	Not executed

(Note 1) This is the same operation as when the execution condition of the **MC** instruction turns OFF. Refer to the explanation of the **MC** and **MCE** instructions.

- Caution is required when using an instruction that is executed by detecting the rise of an execution condition, such as a differential instruction (1 to 7 below).

1. DF (rise differential)
2. CT (counter) count input
3. F118 UDC (up-down counter) count input
4. SR (shift register) shift input
5. F119 LRSR (left and right shift register) shift input
6. NSTP (next step)
7. Differential execution type high-level instruction (instruction specified by P and a number)

■ Examples of step ladder instructions

(1) Sequence control

This is a program that repeats the work in a certain process until it is completed, and then moves to the next process.

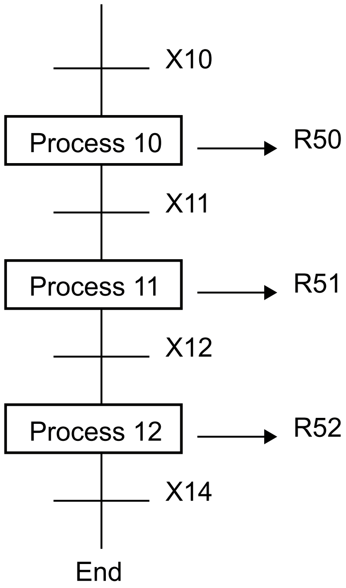
- In the program, write the instruction to start the process to be executed next in each process. When the start instruction is executed, the next process is started, and the process that had been executing is cleared.

- 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.

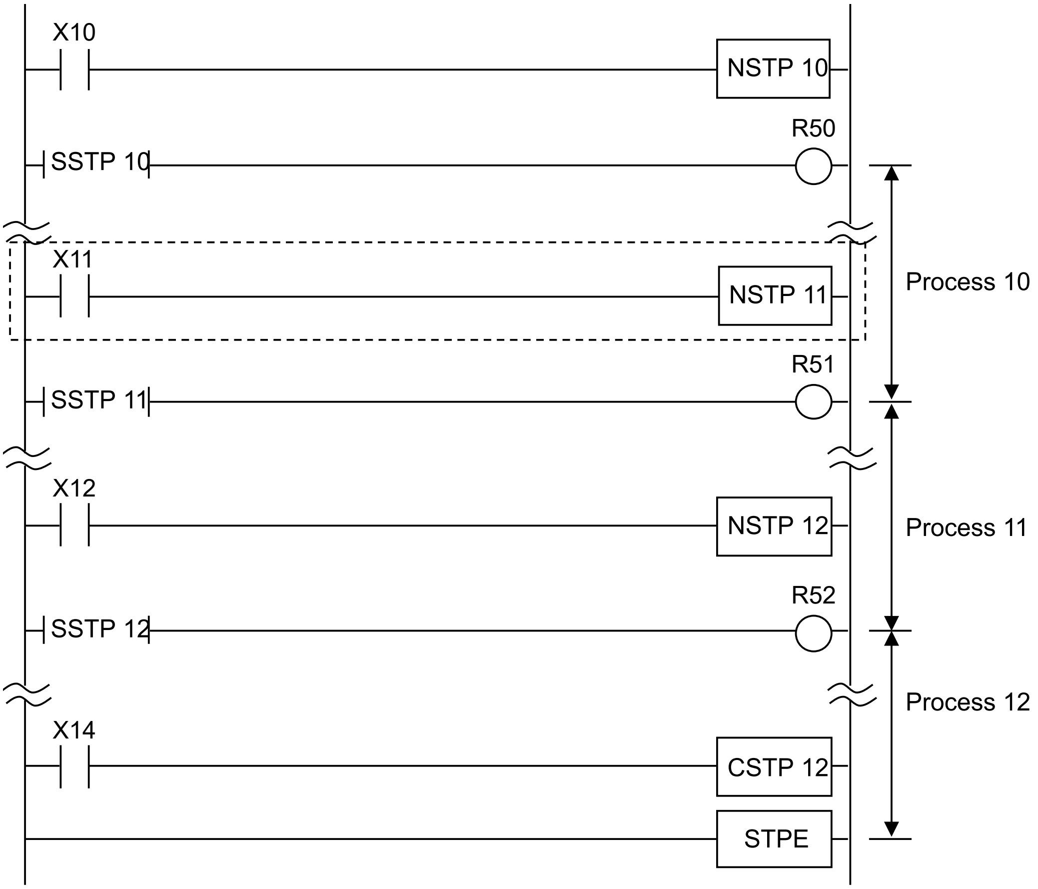
[Program example]

1. When X10 turns ON, process 10 is executed.
2. When X11 turns ON, process 10 is cleared and process 11 is executed.
3. When X12 turns ON, process 11 is cleared and process 12 is executed.
4. When X14 turns ON, process 12 is cleared and step ladder operation finishes.

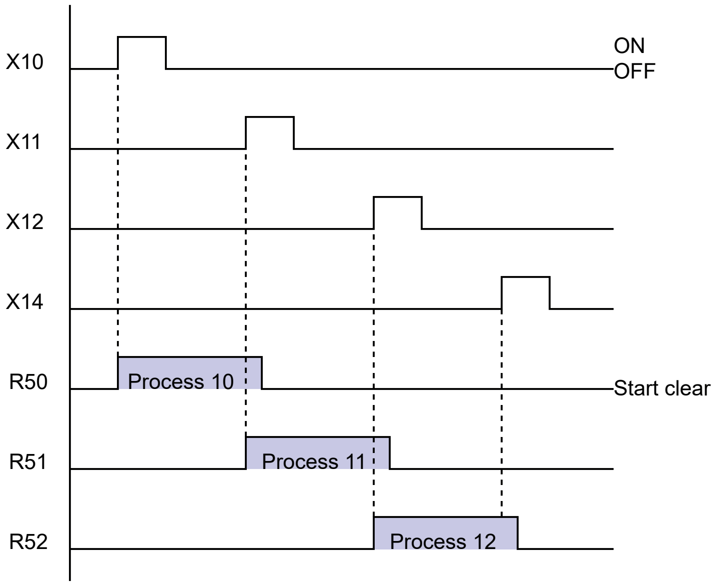
- **Process flowchart**



- **Program**



- **Timing chart**



(2) Selection branch control of a process

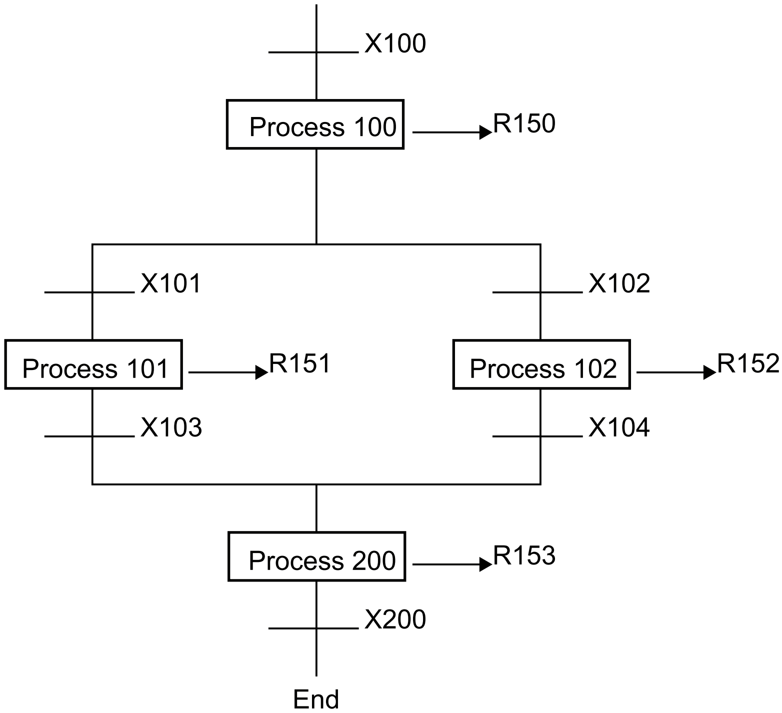
This program selects and switches to the next process according to the actions and results of a particular process. Each process loops until its work is completed.

- 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.

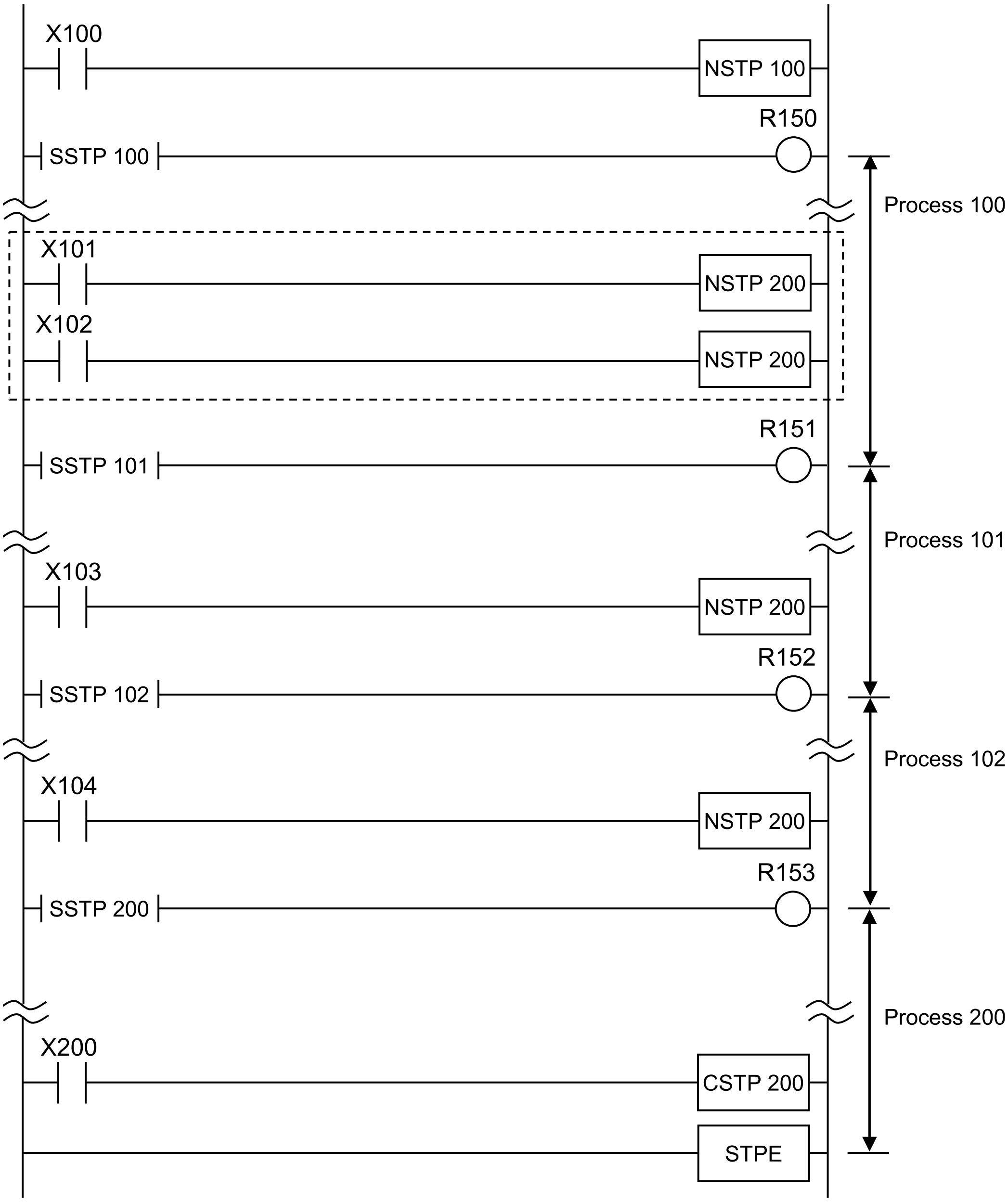
[Program example]

1. When X100 turns ON, process 100 is executed.
2. While process 100 is executing,
   - when X101 turns ON, process 101 is executed.
   - Or when X102 turns ON, process 102 is executed.
3.
   - While process 101 is executing, when X103 turns ON, process 101 is cleared and process 200 is executed.
   - While process 102 is executing, when X104 turns ON, process 102 is cleared and process 200 is executed.
4. When X200 turns ON, process 200 is cleared and step ladder operation finishes.

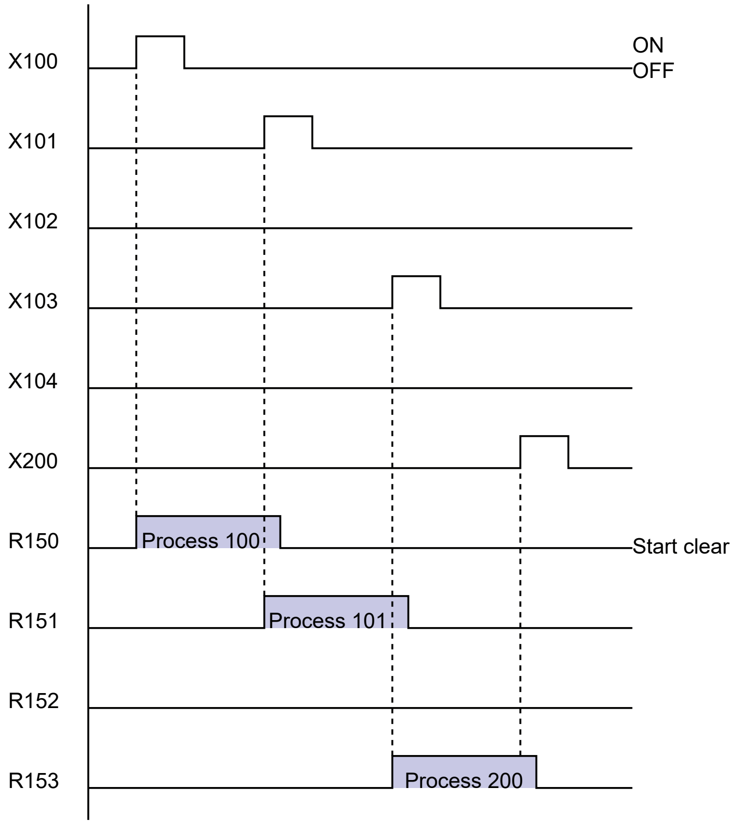
- **Process flowchart**



- **Program**



- **Timing chart**
  This is an example of when X101 turns ON.



(3) Parallel branch merge control of a process

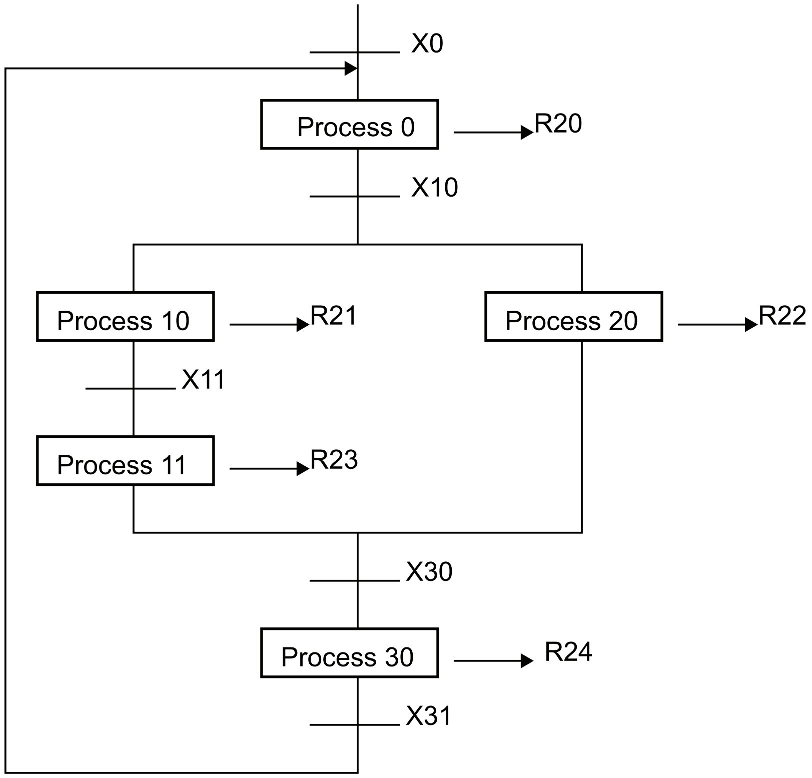
This program starts multiple processes at the same time. When the work is completed in each of the branched processes, they merge again before transferring execution to the next process.

- 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.

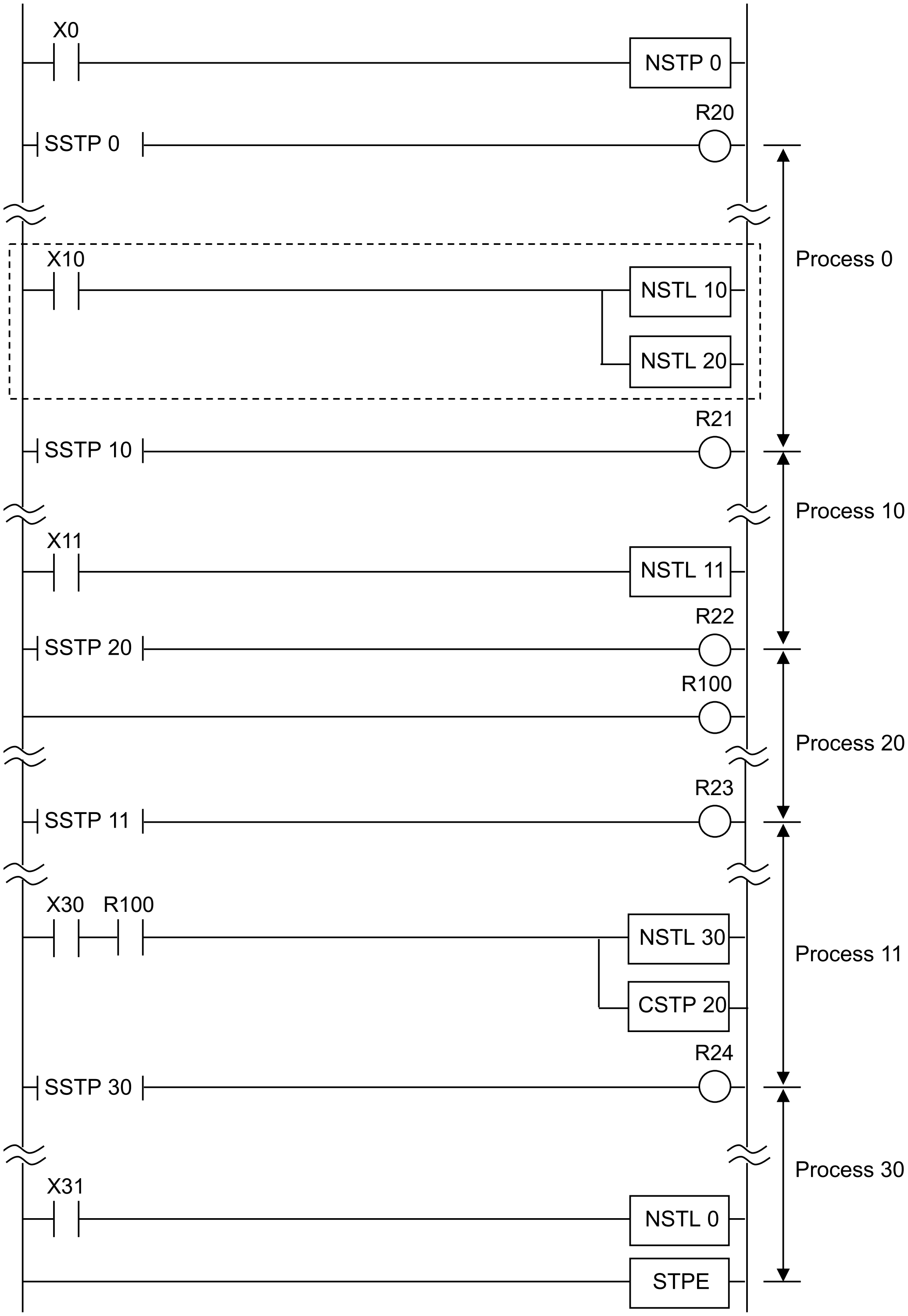
[Program example]

1. When X0 turns ON, process 0 is executed.
2. When X10 turns ON, process 0 is cleared and process 10 and process 20 are executed simultaneously (parallel branch).
3. When X11 turns ON, process 10 transitions to process 11.
4. With processes 11 and 20 executing, when X30 turns ON, execution transfers to process 30 (merge).
   - Process 20 is cleared with the clear instruction.
   - Process 11 is cleared and process 30 is executed.
5. When X31 turns ON, process 30 is cleared and process 0 is executed again.

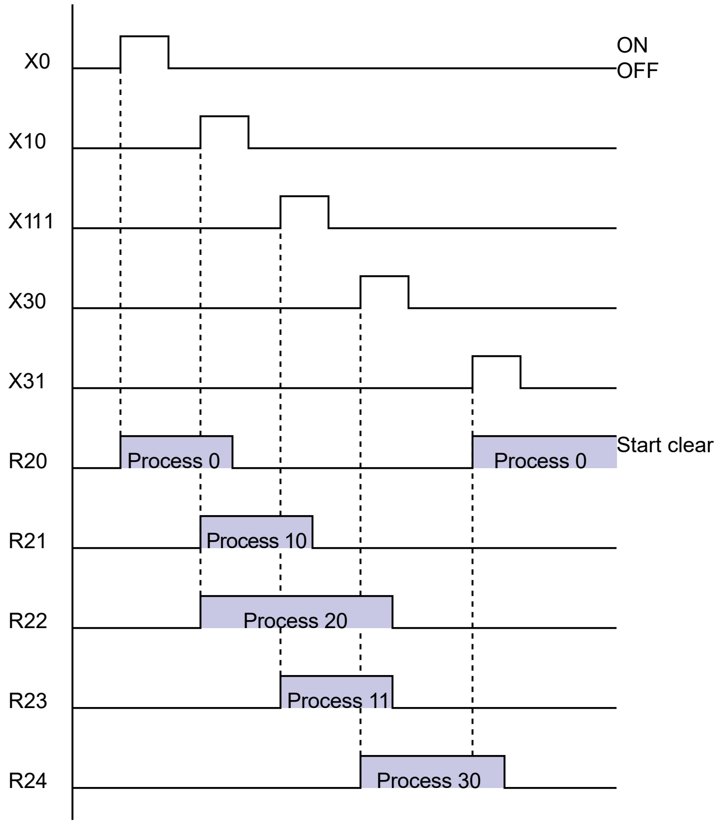
- **Process flowchart**



- **Program**

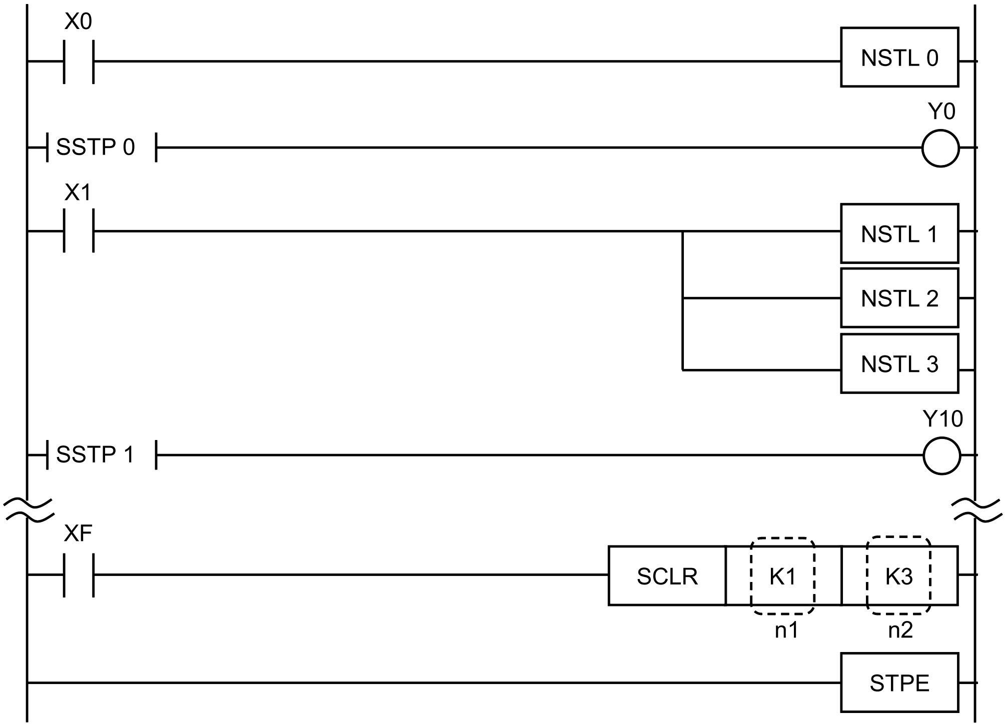

X0
NSTP 0
SSTP 0
R20
X10
NSTL 10
NSTL 20
SSTP 10
R21
X11
NSTL 11
SSTP 20
R22
R100
SSTP 11
R23
X30
R100
NSTL 30
CSTP 20
SSTP 30
R24
X31
NSTL 0
STPE
Process 0
Process 10
Process 20
Process 11
Process 30

- **Timing chart**



5.2 SCLR (Clear Multiple Processes)

■ Instruction format



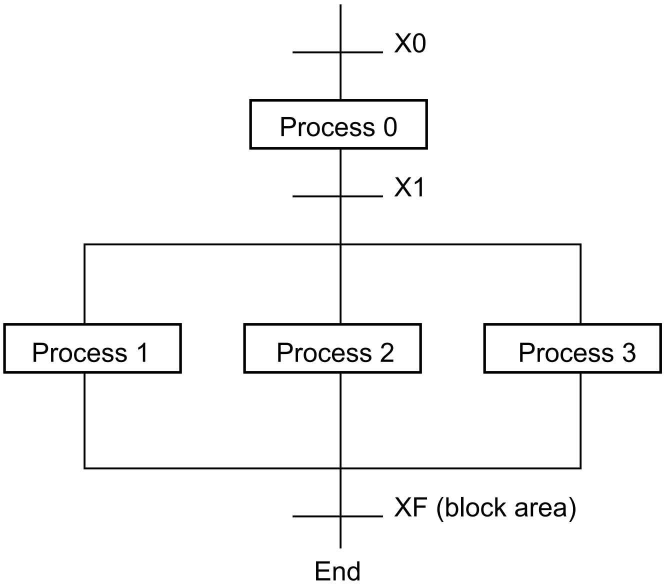
■ Outline of operation

When the **SCLR** instruction is executed, all active processes from process n1 through process n2 are cleared.

■ Operation example

Operation of instruction format description program

When input XF turns ON, active processes from 1 through 3 are cleared.



■ Precautions for programming

- Specify values so that n1 is equal to or smaller than n2.
- The **SCLR** instruction can be executed from both normal ladder areas and active processes.

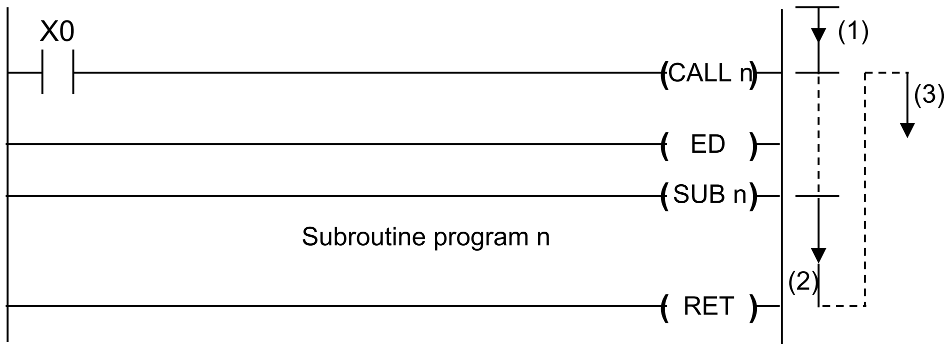
6 Subroutine Instructions

6.1 CALL/SUB/RET (Subroutine Call, Subroutine Entry, Subroutine Return)

■ Instruction format

■ Outline of operation

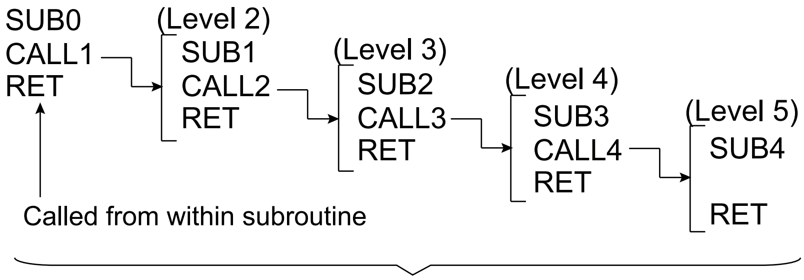
- When the execution condition turns ON, the **CALL** instruction is executed and the subroutine program of the specified number starting from the **SUB** instruction is executed.
- When the **RET** instruction is executed, the program returns to the address following the **CALL** instruction in the main program and execution of the main program continues.



CALL nis executed in the order of (1) to (3).

■ Subroutine program syntax

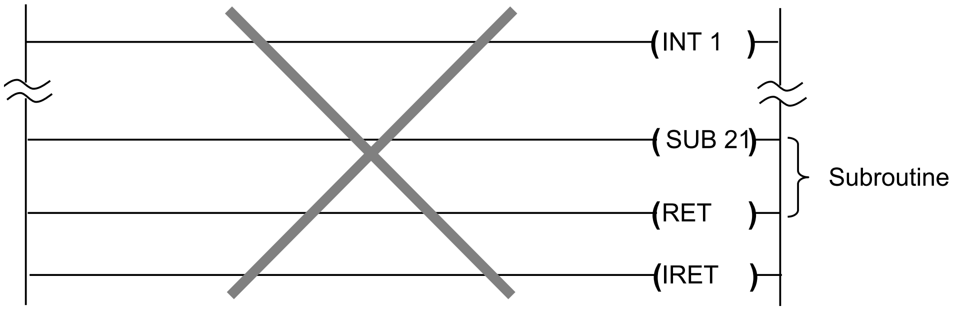
- "Subroutine program n"is the program between the **SUB n** instruction and the **RET** instruction. Always write a subroutine to an address after the **ED** instruction.
- The **CALL n** instruction can be described in the main program and any other subroutine program, interrupt program, or step ladder. Additionally, a **CALL** instruction with the same number can be repeated.
- Subroutines can be nested up to 5 layers deep.



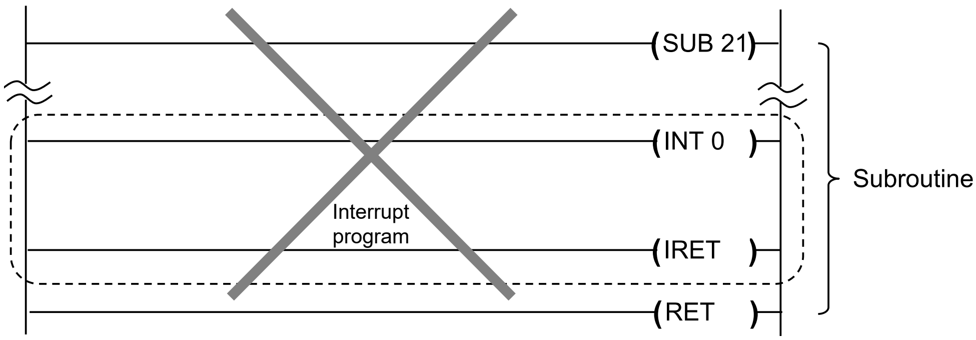
Example of 5 layer nesting

■ Precautions for programming

- A subroutine program cannot be described in an interrupt program.



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



- Caution is required when using an instruction that is executed by detecting the rise of an execution condition, such as a differential instruction (1 to 7 below), in a subroutine.
  1. DF (rise differential)
  2. CT (counter) count input
  3. F118 UDC (up-down counter) count input
  4. SR (shift register) shift input
  5. F119 LRSR (left and right shift register) shift input
  6. NSTP (next step)
  7. Differential execution type high-level instruction (instruction specified by P and a number)

■ Operation when the execution condition of the CALL instruction turns OFF

When the execution condition of the **CALL** instruction turns OFF, the operation of that subroutine is not performed (the same applies to calls in master control and step ladders). In this case, the operation of each instruction used in the subroutine is as follows.

OT instruction	Holds the state.
KP instruction	Holds the state.
SET instruction	Holds the state.
RST instruction	Holds the state.
TM instruction	Clocking is not performed. Note that the time cannot be guaranteed if clocking is not performed once during a scan.
CT instruction	Holds the current progress.
SR instruction	Holds the current progress.
Differential instruction	The same as when a differential instruction is used between MC and MCE.
Other instructions	Not executed.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the**CALL**instruction is executed in the 5th layer of a subroutine when 5-layer nesting is being performed

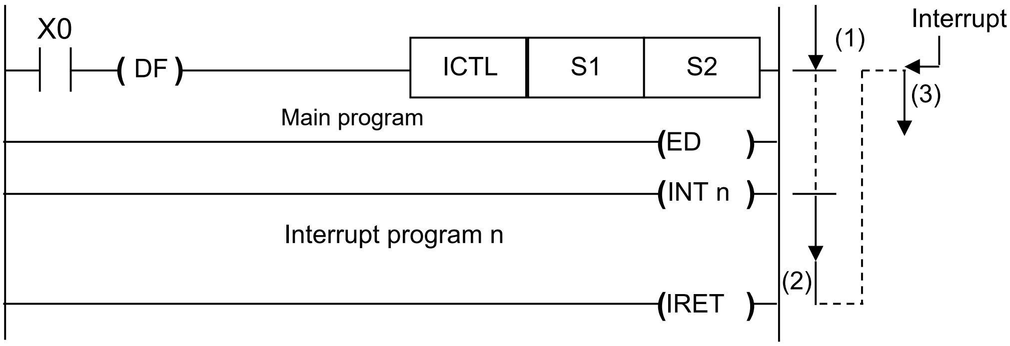
7 Interrupt Instructions

7.1 INT/IRET (Interrupt / Interrupt Return)

■ Instruction format

■ Outline of operation

- When an interrupt is input, the interrupt program of the number specified is executed starting from the **INT** instruction.
- When the interrupt program reaches the **IRET** instruction, the program returns to the address where the interrupt occurred and the main program resumes.



When an interrupt occurs, execution will occur in the order of (1) to (3).

■ Interrupt Program Syntax

- The interrupt program is the program between the **INT n** instruction and the **IRET** instruction. The interrupt program must always be placed in an address after the **ED** instruction.
- The number of the interrupt program is determined by the type of interrupt.

Interrupt program number	Interrupt input	High-speed counter target value match interrupt
INT0	X0	ch0
INT1	X1	ch1
INT2	X2	–
INT3	X3	ch2
INT4	X4	ch3
INT5	X5	–

Interrupt program number	Interrupt input	High-speed counter target value match interrupt
INT6	X6	ch4
INT7	X7	ch5
INT8	–	PLS-ch0
INT9	–	PLS-ch1
INT10	–	PLS-ch2
INT11	–	PLS-ch3
INT12	–	–
INT13	–	–
INT24	Periodic interrupt	

(Note 1) When using a high-speed counter target value match interrupt program, the counting performance of the high-speed counter may decrease upon initiation of the interrupt program..

(Note 2) Only PLS-ch* becomes a pulse output target value match interrupt.

■ Before inputting an interrupt program

1. Specify the contact to be used as the interrupt input.
   Select the input contact to be used as the interrupt input and specify it in system register No. 403.

   Note

   - If the high-speed counter/pulse catch is set, that contact cannot be used as the interrupt input.
   - There is no need to specify the input contact for high-speed counter target value match interrupts and periodic interrupts.
2. "Enable" execution of interrupt programs.
   All interrupt programs are set to "execution disabled" as default. "Enable" interrupt programs to be executed using the **ICTL** instruction.

■ Precautions when rewriting during RUN

If the program is rewritten in "RUN mode", all interrupt programs will be set to "execution disabled", making it necessary to "enable" them after rewriting in RUN.

To automatically re-enable with a ladder program, use R9034 (rewrite during RUN completion flag). R9034 is a special relay that is ON for only 1 scan after completion of a rewrite during RUN.

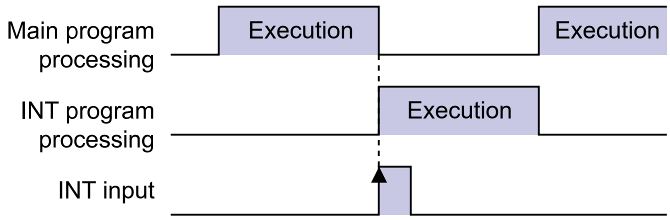
■ Interrupt program execution

There are three types of interrupt.

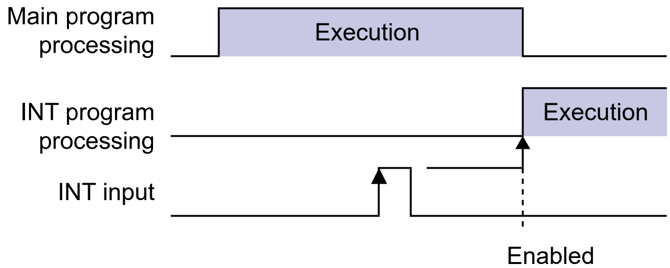
1. Interrupt from the input contact
   An interrupt occurs from the input specified in system register No. 403.
2. High-speed counter target value match interrupt
   When executing a high-speed counter instruction, an interrupt occurs when the high-speed counter elapsed value equals the set target value.
3. Periodic interrupt (INT24)

The interrupt occurs in fixed time intervals. The time interval is set with the **ICTL** instruction.

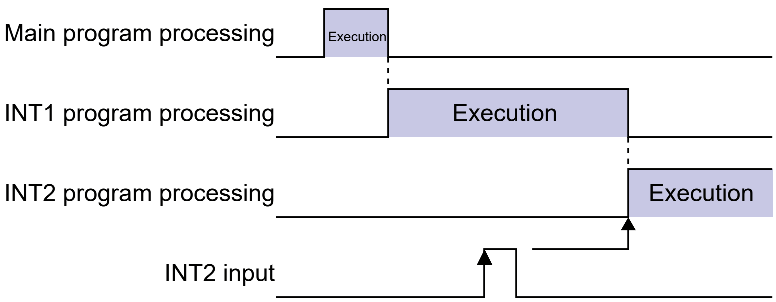
- If the interrupt occurs, the interrupt program with the corresponding number is executed.



- If interrupts are disabled, they will be executed when execution is enabled with the **ICTL** instruction.

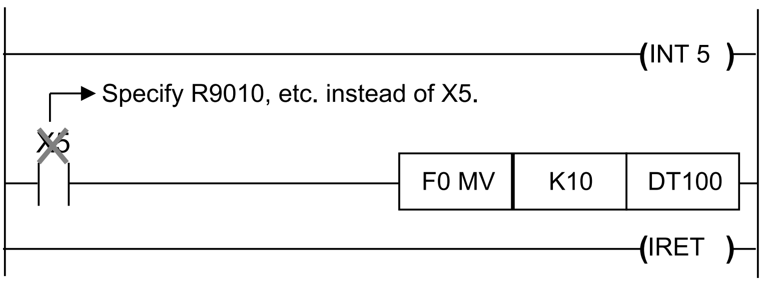


- If an interrupt occurs during execution of another interrupt program, it will be executed after the other program finishes.

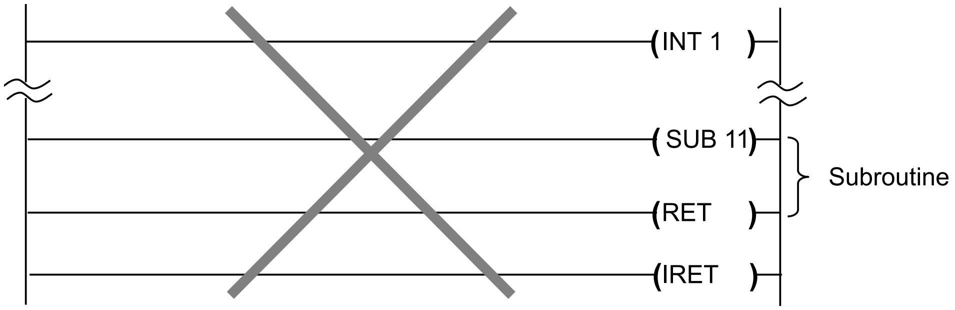


■ Precautions for programming

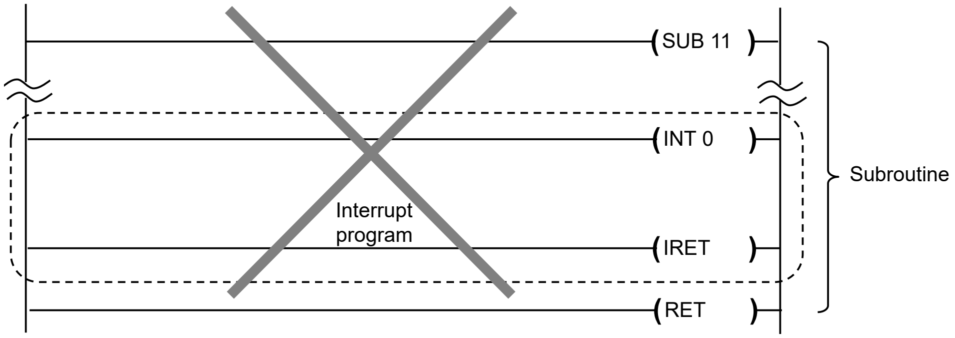
- A syntax error will occur if either the **INT** instruction or **IRET** instruction is missing.
- When an interrupt occurs, the operation memory corresponding to the interrupt input contact is not I/O refreshed. Therefore, contacts other than the interrupt input contact, such as the normally ON relay R9010, should be specified by the input conditions in the interrupt program.



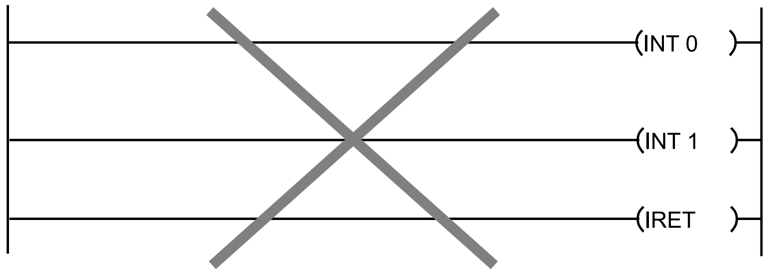
- A subroutine program cannot be used in an interrupt program.



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

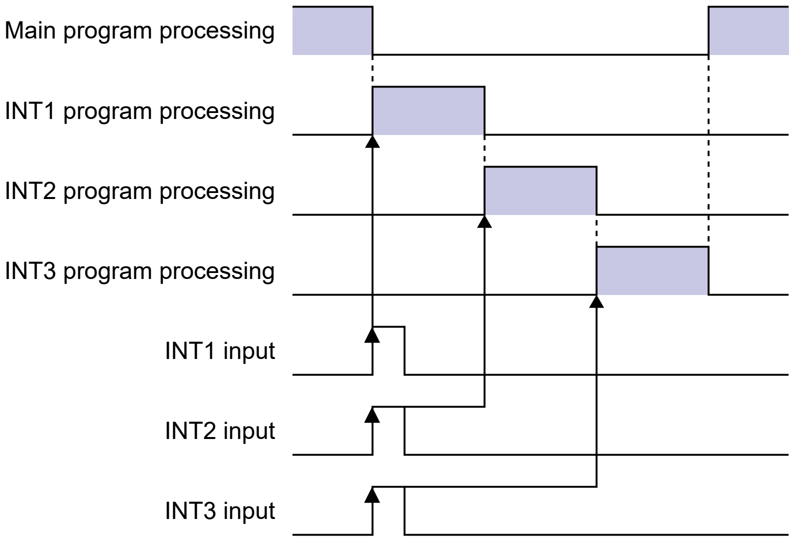


- Another interrupt program cannot be used in an interrupt program.



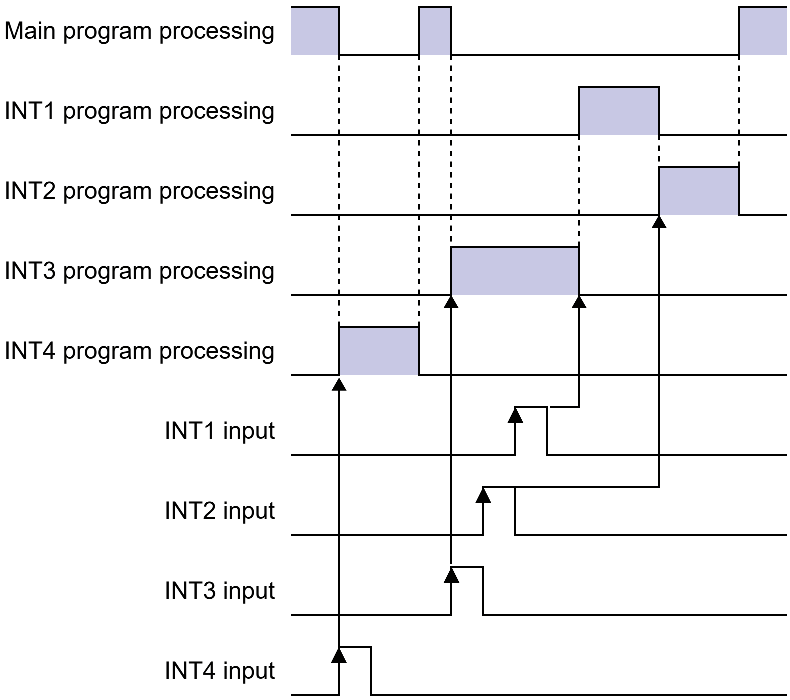
■ Control when multiple interrupts occur simultaneously

- When multiple interrupts occur simultaneously, the interrupt program with the smallest number is executed first. The other interrupt programs are then placed into an execution waiting state., After the first interrupt program is completed, the other programs will be executed in order from the smallest number.



- 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.

e.g.

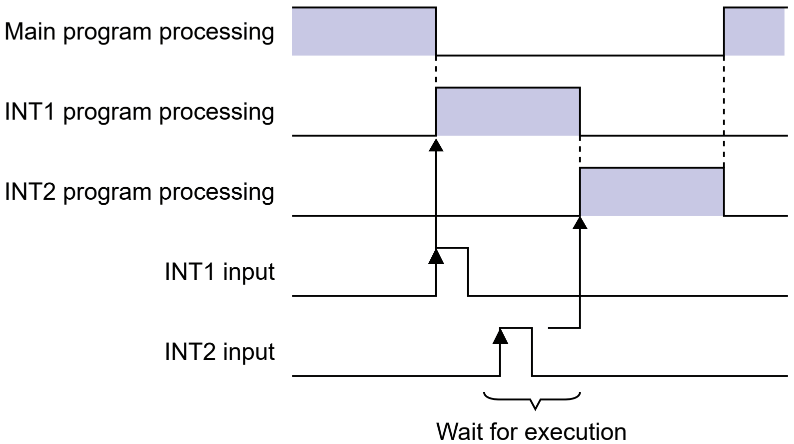


(Note 1) During execution of the INT3 program in the example above, INT1 will be executed before INT2, even if interrupt INT2 occurs before INT1.

■ Interrupt program execution waiting and clearing

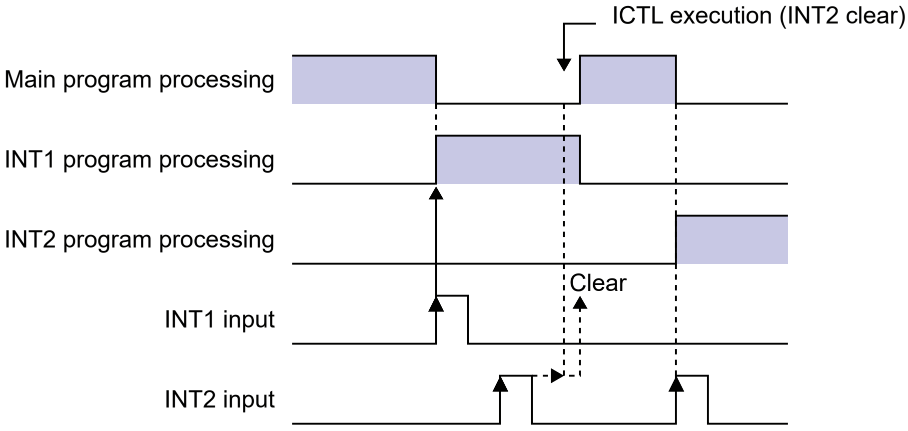
- When multiple interrupts occur simultaneously or when a new interrupt occurs during execution of another interrupt program, the interrupts of lower priority will enter an "execution wait state". They will be executed in order when the other interrupt program finishes execution.

e.g.



- If placed in execution wait state, there is a time difference between the occurrence of the interrupt and execution of the interrupt program. To avoid execution of these execution wait state programs, clear them using the **ICTL** instruction. Cleared interrupt programs will not be executed.

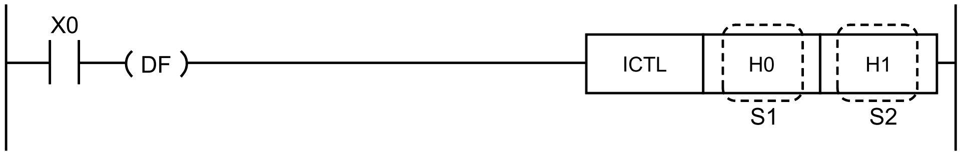
e.g.



- Even when execution of interrupt programs is disabled with the **ICTL** instruction, if an interrupt occurs it will enter an "execution wait state". Waiting interrupt programs will be executed upon enabling execution with the **ICTL** instruction. As noted above, the interrupt programs in an execution wait state can be cleared by using the **ICTL** instruction.

7.2 ICTL (Interrupt Control)

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the control data, or constant data
S2	Area storing the control data, or constant data

■ Devices that can be specified (indicated by •)

Operand s	WX	WY	WR	WL	SV	EV	DT	LD	I	SW R	SD T	Constant K	H	M	f	Index modifier	Integer Device
S1	•	•	•	•	•	•	•	•	•			•	•			•	
S2	•	•	•	•	•	•	•	•	•			•	•			•	

■ Outline of operation

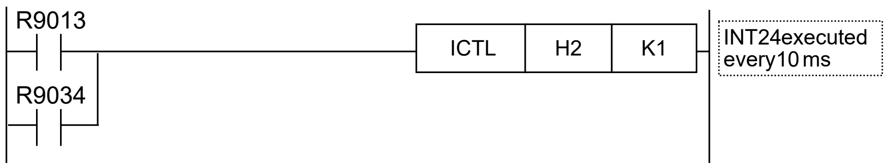
- When the ICTL instruction is executed, based on the content of [S1] and [S2], either (1) enabling or disabling of the interrupt program is specified, or (2) clearing of the interrupt program is specified.
- Perform differential execution using an instruction such as **DF** so that it is only executed once when setting.
- Multiple ICTL instructions can be written consecutively for a single execution condition.

Always execute this instruction before executing an interrupt program to enable interruption.

■ Precautions when rewriting during RUN

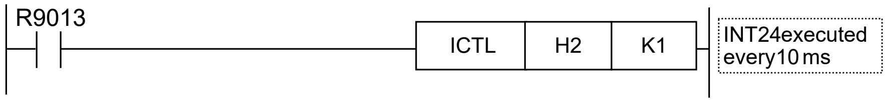
- If a rewrite during RUN is performed while using an interrupt function, the interrupt function will be disabled. It is necessary to re-enable execution of the interrupt program with an ICTL instruction.

e.g. A periodic interrupt every 10 ms is set at the start of operation (re-enables interrupt after rewriting during RUN.)



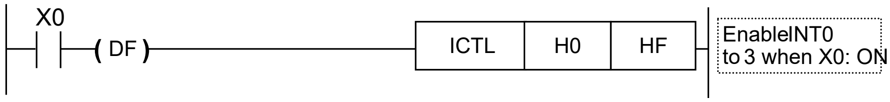
■ Description examples

Example 1) Setting a periodic interrupt every 10 ms at the start of operation

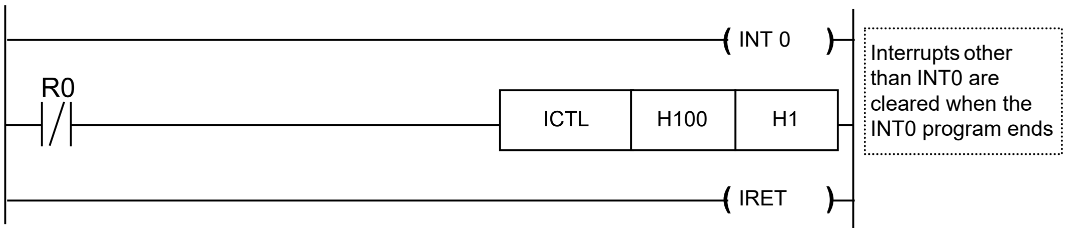


(Note 1) R9013 (initial pulse relay) is a relay that turns ON in only the first scan after execution begins.

Example 2) Enabling INT0 to 3 when X0 rises

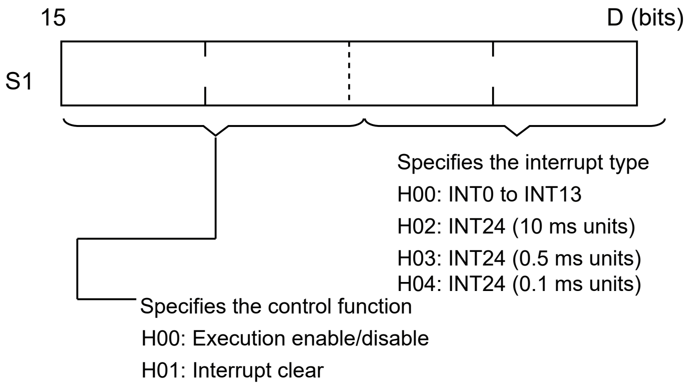


Example 3) Clearing interrupts other than INT0 when the INT0 program ends



■ Specifying control data

[S1]: Specifies the type of interrupt and the function to be controlled



(1)	When specifying enable/disable execution of INT0 to 7	[S1] = H0
(2)	When specifying to clear interrupts for INT0 to 7	[S1] = H100
(3)	Time interval setting for INT24	[S1] = H2 (10 ms units) [S1] = H3 (0.5 ms units) [S1] = H4 (0.1 ms units)

■ Precautions for programming

Inputs that can actually be used as interrupt inputs. (Refer to the table below)

Interrupt program number	Interrupt input
INT0	X0
INT1	X1
INT2	X2
INT3	X3
INT4	X4
INT5	X5
INT6	X6
INT7	X7
INT8	–
INT9	–
INT10	–
INT11	–
INT12	–
INT13	–
INT24	Periodic interrupt

[S2]: Specifies the control content

1. Specifying enable/disable execution of the interrupt program (when S1 = H0 or S1 = H1)
   Set the control data to the bit corresponding to the interrupt program number you wish to control.
   - To enable execution, set the program number bit to "1"
   - To disable execution, set the program number bit to "0"

   e.g. Enabling interrupt program INT1 and INT2, and disabling INT0 and INT3 to INT13

   Bit 15 Bit 0

0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
		13	12	11	10	9	8	7	6	5	4	3	2	1	0

   (INT number)

2. Clear the interrupts (when S1 = H100)
   Set the control data to the bit corresponding to the interrupt program number you wish to control.
   - Set the program number bits to be cleared to "0"
   - Set the program number bits not to be cleared to "1"

e.g. Clearing interrupt program INT0 to INT2, not clearing INT3 to IN13

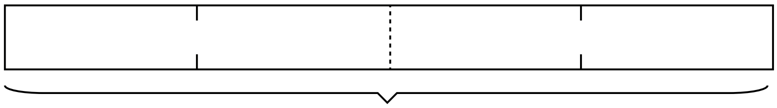
Bit 15 ... Bit 0

0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
		13	12	11	10	9	8	7	6	5	4	3	2	1	0

(INT number)

3. Specifying a periodic interrupt (when S1 = H2)
   Specify the setting value with a decimal.
   Time interval = value of [S2] × 10 (ms)

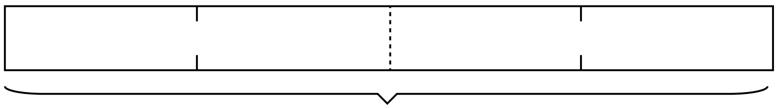
   Bit 15 ... Bit 0

   K0 to K3000

   - Time interval setting is K1 to K3000 (10 ms to 30 s)
   - Disable INT24 is K0
4. Specify a periodic interrupt (when S1 = H3)
   Time interval = value of [S2] × 0.5 (ms)

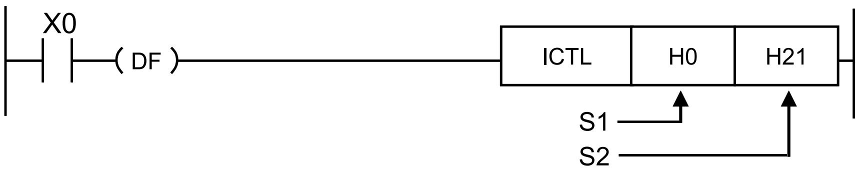
   Bit 15 ... Bit 0

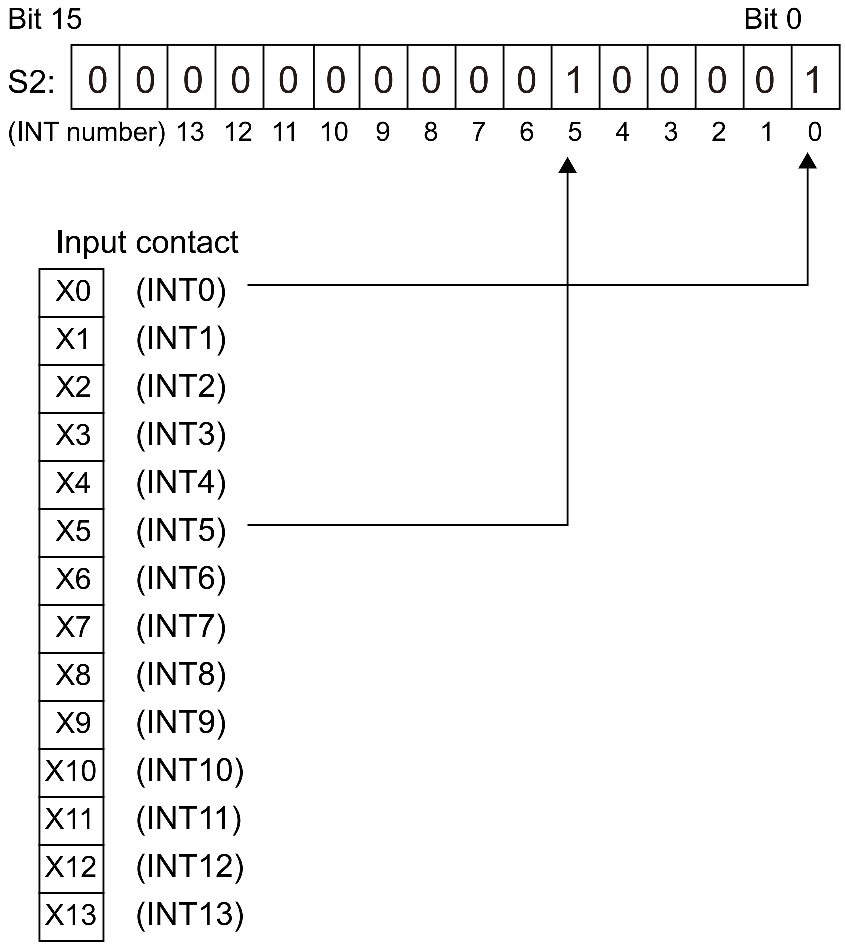
   K0 to K3000

   - Time interval setting is K1 to K3000 (0.5 ms to 1.5 s)
   - Disable INT24 is K0

■ Example setting to enable interrupt program execution



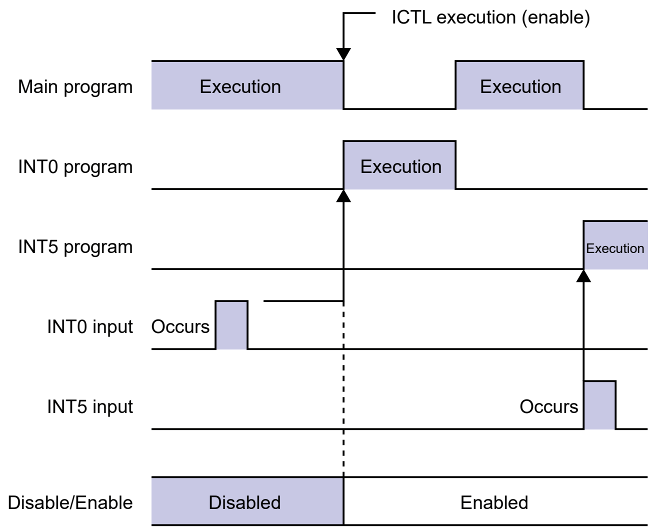
[S1]: H0000	Specifies enable/disable execution of interrupt programs corresponding to interrupts from a specified input contact or interrupts matching the target value
[S2]: H0021	Enable INT0 and INT5 (bits 0 and 5 are"1") and disable others



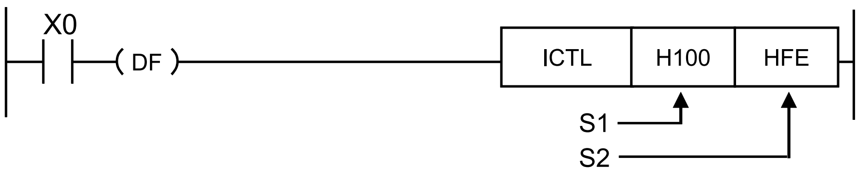
- Set the bits corresponding to the interrupts to be enabled to"1".

Description

If this ICTL instruction is executed, the No. 0 and No. 5 programs will be executed if the corresponding interrupt occurs.



■ Example setting to clear interrupts

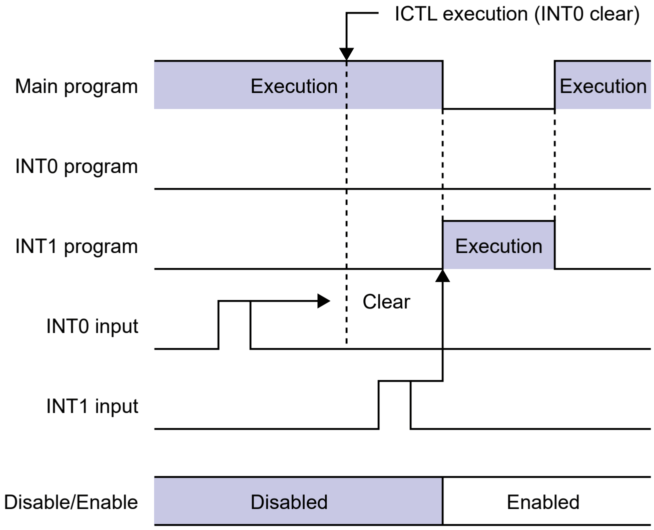


[S1]: H0100	Clears the interrupts from a specified input contact or interrupts matching the target value
[S2]: HFE	Clears INT0 interrupt (bit 0 is"0"), others are not cleared

(Note 1) Refer to the"Enable/Disable"example regarding the correspondence between setting values and interrupt input contacts.

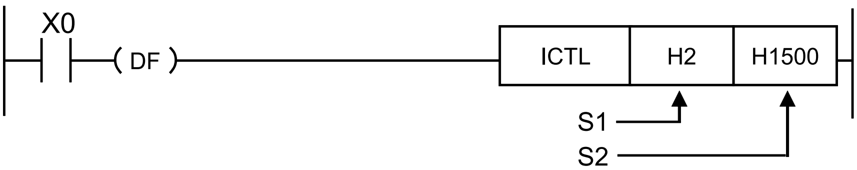
Description

If in a state where an INT0 interrupt is occurring but the corresponding interrupt program is not being executed, executing this ICTL instruction will clear the interrupt.



(Note 1) As INT0 has been cleared, it will not be executed even after being enabled. INT1 has not been cleared, so it will be executed after being enabled.

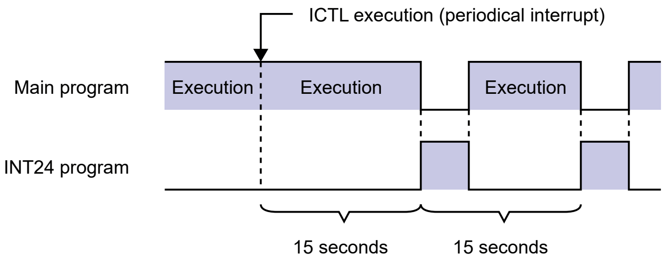
■ Example settings for periodic interrupt



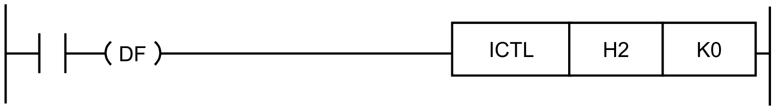
[S1]: H0002	Specifies a periodic interrupt
[S2]: K1500	Specifies the time interval of the periodic interrupt If K1500, the time interval is K1500 × 10 ms = 15000 ms (15 s)

Description

If this ICTL instruction is executed, a periodic interrupt will occur every 15 seconds and the INT24 interrupt program will be executed.



(Note 1) To stop the periodic interrupt, execute the following.



7.2.1 How to start the interrupt program when executing the high-speed counter match ON / match OFF instruction

Procedure

1. Set the counter via the system register. (It is not necessary to set the external interrupt.)
2. Specify the interrupt program in the program.
   The high-speed counters correspond to the interrupt programs as indicated in the table below.

Interrupt program number	High-speed counter target value match interrupt
INT0	ch0
INT1	ch1
INT2	–
INT3	ch2
INT4	ch3
INT5	–
INT6	–
INT7	–
INT8	–
INT9	–
INT10	–
INT11	–
INT12	–
INT13	–

3. Enable the setting via the ICTL instruction. Enable ICTL H0, H9 - - INT0 and INT3.
4. Start the match ON / match OFF instruction.
5. The program is executed when the conditions for match ON / match OFF are met.

Note

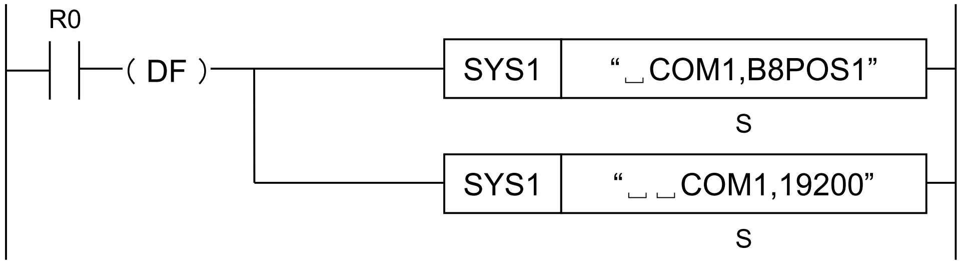
- When using a high-speed counter target value match interrupt program, the counting performance of the high-speed counter may decrease upon initiation of the interrupt program.

(MEMO)

8 Special Setting Instructions

8.1 SYS1 (Communication Condition Setting)

■ Instruction format



(Note 1) In the example shown in the figure above, the transmission format and baud rate of COM1 port are set as below

Character bit length: 8; Parity bit: Odd parity; Stop bit: 1

Baud rate: 19200 bps

■ Operands

Item	Setting
S	Character constant

■ Devices that can be specified (indicated by •)

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant			Index modifier (Note 1)
												K	H	M	
S														•	

(Note 1) A character constant cannot be specified.

■ Outline of operation

- This instruction changes the communication conditions of the port specified as the first keyword to the contents specified as the second keyword.
- The following functions can be changed.
  - Transmission format
  - Baud rate
  - Unit number setting (direct/indirect)
  - Header and terminator
  - RS (Request to Send) control

<Example> In the case of the above program

When R0 turns ON, the transmission format and baud rate of COM port 1 are set as follows.

Character bit length 8, odd parity, stop bit 1, baud rate 19,200 bps

■ Precautions for programming

- Enclose the first and second keywords in double quotation marks (”).

- Separate the first keyword and the second keyword with a comma (,) without inserting spaces between them.
- Put a number of spaces calculated as (12 characters - number of input characters) in front of the first keyword so that the total number of characters for the first and second keywords is 12.

  In FPWIN GR7 Ver. 2.23 and later, if the character constant does not reach 12 characters, spaces are automatically input when the project is converted.

  Example: When entering COM1 as the first keyword and 19200 as the second keyword

Specified contents	"	␣	␣	C	O	M	1	,	1	9	2	0	0	"
Number of characters		1	2	3	4	5	6	7	8	9	10	11	12	

- Executing this instruction does not rewrite the contents of the system ROM of the main unit. As a result, when the power supply is turned OFF and ON again, the contents of the system registers set with FPWIN GR7 are rewritten.
- It is recommended to use a differential instruction for this instruction.
- Because the system register settings are changed, a verification error may occur when verification is performed with FPWIN GR7.
- Specify the COM port as COM1.

■ Specifying the communication conditions (transmission format)

- Specify transmission format (data length, parity check, and stop bit).

First keyword	Second keyword		
Used ports	**Data length**	**Parity check**	**Stop bit**
COM1: COM1 port TOOL: Tool port	B7: 7 bits B8: 8 bits	PN: No parity PO: Odd parity PE: Even parity	S1: 1 S2: 2

Setting examples

Example 1	S	"␣COM1,B8PES2"
Settings		Port: COM1 / Data length: 8 bits / Parity check: Even parity / Stop bit: 2

■ Specifying the communication conditions (baud rate)

- Specify a baud rate.

First keyword	Second keyword	
Used ports	**Baud rate**	
COM1: COM1 port TOOL: Tool port	300: 300 bps 600: 600 bps 1200: 1200 bps 2400: 2400 bps 4800: 4800 bps 9600: 9600 bps	19200: 19200 bps 38400: 38400 bps 57600: 57600 bps 115200: 115200 bps

(Note 1) If the baud rate is changed as below, communications passing through all COM ports will be reset.

Baud rates of all COM ports are 4800 bps or higher ↔ Baud rate of any of the COM ports is 2400 bps or lower

(Note 2) If the baud rate of any of the COM ports is 2400 bps or lower, F-ROM access will slow down.

e.g. F12 (ICRD) instruction, P13 (PICWT) instruction, etc.

Setting example

Example 1	S	"␣␣␣COM1,1200"
Settings		Port: COM1, 1200 bps

■ Specifying the communication conditions (unit number)

● Specify a unit number using directly or indirectly.

First keyword	Second keyword	
Used ports	**Unit number (for direct specification)**	**Unit number (for indirect specification)**
COM1: COM1 port TOOL: Tool port	No1 to No99: Unit numbers 1 to 99	For a DT number that contains a unit number, specify D followed by a 4-digit number, as below. D0000 to D9999: DT0 to DT9999

(Note 1) For direct specification of unit numbers, you can specify unit numbers 1 to 99. For indirect specification of unit numbers, specify a DT number that contains a unit number.

Setting example

Example 1	S	"␣␣␣COM1,No99"
Settings		(For direct specification of unit numbers) Port: COM1 / Unit number: No99

■ Specifying the communication conditions (header/terminator)

● Specify a header or terminator.

First keyword	Second keyword	
Used ports	**Header**	**Terminator**
COM1: COM1 port TOOL: Tool port	STX: With STX NOSTX: Without STX	ETX: ETX CR: CR CRLF: CR + LF NOTERM: No terminator

Setting examples

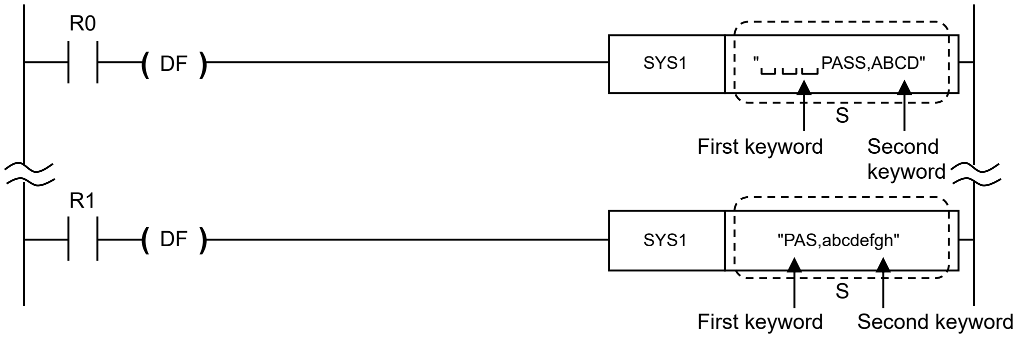
Example 1	S	"␣␣␣␣COM1,ETX"
Settings		Port: COM1 / Terminator: ETX
Example 2	S	"␣␣␣␣␣COM1,CR"
Settings		Port: COM1 / Terminator: CR

■ Flag operations

Name	Description
R9007 R9008 (ER)	Set if non-keyword text or out-of-range values are specified for the first and second keywords.
	Set if there is no comma between the first and second keywords.
	Set if no communication cassette is mounted when COM1 is specified.
	Set if the baud rate or transmission format for COM1 is changed when COM1 is in PLC link mode.
	Set if the baud rate or transmission format is changed while the modem for the COM0, COM1 port is being initialized.
	Set if the communication mode is set to any mode other than general-purpose communication mode when a header or terminator is set.
	Set if any communication cassette other than 1-channel RS-232C type communication cassettes is mounted when RS control is performed.
	Set if a unit number greater than the maximum unit number set in the system register is specified when COM1 is in PLC link mode.
	Set if the communication speed is changed as below while F-ROM is being accessed. Baud rates of all COM ports are 4800 bps or higher ↔ Baud rate of any of the COM ports is 2400 bps or lower

8.2 SYS1 (Password Setting)

■ Instruction format



■ Operands

Item	Setting
S	Character constant

■ Devices that can be specified (indicated by •)

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer device
												K	H	M	f		
S														•			

■ Outline of operation

The password specified for the controller is changed to the contents specified by the second keyword.

■ Operation example

Operation of instruction format description program

When R0 turns ON, the controller password is changed to "ABCD".

■ Specifying keywords

- For a 4-digit password

SYS1," PASS, ABCD"

PASS: Fixed

Password (Example: Set password to "ABCD")

- For an 8-digit password

SYS1," PAS, abcdefgh"

PAS: Fixed

Password (Example: Set password to "abcdefgh")

If there are fewer than eight characters, spaces are automatically added at the end to make eight characters.

■ Precautions for programming

- When this instruction is executed, it takes approximately 100 ms to write to the built-in F-ROM.
- If the specified password is the same as the password that has already been written, the password is not written to the F-ROM.
- It is recommended to use differential execution for this instruction.
- Enter the first and second keywords after "M" with right-aligned 12 characters.

  Separate the first keyword and second keyword with a comma (,) without inserting a space. An operation error occurs.

  Example: When entering "SYS1, M COM1,WAIT2"

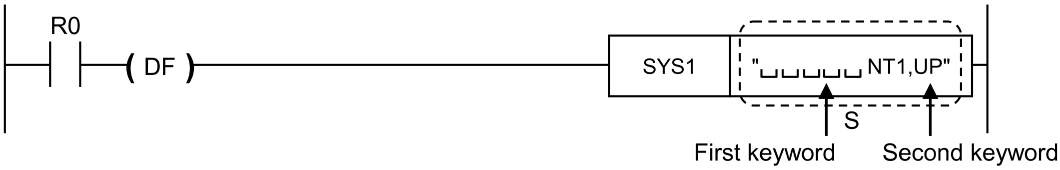
```
Enter"M _ _ C O M 1 , W A I  T  2 ".
       1 2 3 4  5  6 7 8  9 101112
```

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when a character other than a keyword is specified.
	Turns ON when there is no comma between the first and second keywords.
	Turns ON when the keyword is specified in lower-case characters (for a 4-digit password).
	Turns ON when the data specified for the password specifies characters other than 0 to 9 and A to F, or the specified data consists of other than four digits (for a 4-digit password).

8.3 SYS1 (Interrupt Setting)

■ Instruction format



■ Operands

Item	Setting
S	Character constant

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer device
S														•			

■ Outline of operation

The input specified by the first keyword is set as the interrupt input, and the input conditions are changed to the contents specified by the second keyword.

■ Operation example

Operation of instruction format description program

When R0 turns ON, input X1 is set to the interrupt input that becomes valid at the rising edge.

■ Specifying keywords

SYS1," INT2, UP"

Interrupt input
INT0:X0 INT1:X1 INT8:X0 INT9:X1
INT2:X2 INT3:X3 INT10:X2 INT11:X3
INT4:X4 INT5:X5 INT12:X4 INT13:X5
INT6:X6 INT7:X7

Effective edge
UP: Rising edge
DOWN: Falling edge
BOTH: Rising and falling edges

■ Precautions for programming

- Executing this instruction does not rewrite the contents of the system ROM of the main unit. As a result, turning the power supply OFF and then ON again rewrites the contents of the system registers specified by the programming tool software.
- It is recommended to use differential execution for this instruction.
- When UP or DOWN has been specified, the contents of the system registers change in accordance with the specification, meaning a verification error may occur in some cases

when the program is verified. When BOTH has been specified, the contents of the system registers do not change.

- Enter the first and second keywords after "M" with right-aligned 12 characters.

  Separate the first keyword and second keyword with a comma (,) without inserting a space. An operation error occurs.

  Example: When entering "SYS1, M COM1,WAIT2"

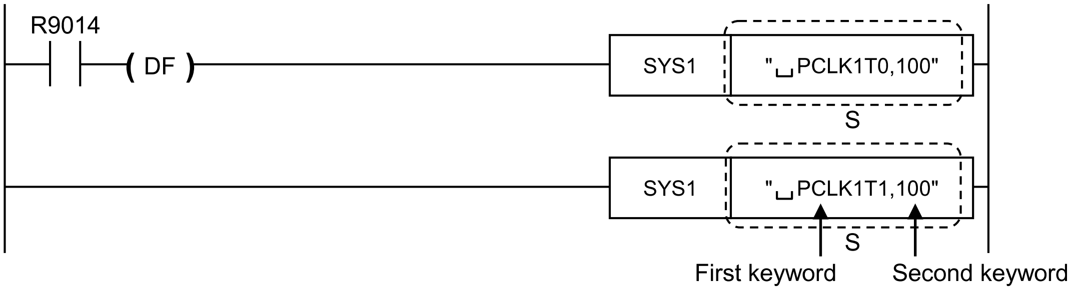
```
Enter"M _ _ C O M 1 , W A I  T  2 ".
      1 2 3 4  5  6 7 8  9 10 11 12
```

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when a character other than a keyword is specified.
	Turns ON when there is no comma between the first and second keywords.
	Turns ON when the keyword is specified in lower-case alphabet characters.

8.4 SYS1 [PC (PLC) Link Time Setting]

■ Instruction format



■ Operands

Item	Setting
S	Character constant

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer device
												K	H	M	f		
S														•			

■ Outline of operation

- Set the condition specified by the first keyword as the time specified by the second keyword.
- The setting for the link entry waiting time is set if the transmission cycle time is shortened when there are stations that have not joined the link (*).

  *Stations that have not joined the link: stations that have not been connected between the No. 1 station and the station with the largest number, or stations for which the power supply has not been turned on
- The error detection time setting for the transmission assurance relay is set if the time between the power supply being turned OFF at one station and the transmission assurance relay from the powered-OFF station being turned OFF at a different station is to be shortened.

■ Operation example

Operation of instruction format description program

During PC (PLC) link, when R9014 turns ON (at leading edge), the link entry waiting time and error detection time for the transmission assurance relay are set as follows.

Link entry waiting time: 100 ms

Transmission assurance relay error detection time: 100 ms

■ Specifying keywords

1. Link entry waiting time

SYS1," PCLK1T0, 100"

PCLK1T0: Fixed

Specified range: 10 to 400 (10 ms to 400 ms)

2. Error detection time for transmission assurance relay

SYS1," PCLK1T1, 100"

PCLK1T1: Fixed

Specified range: 10 to 6400 (10 ms to 6400 ms)

Precautions for programming

- The program should be placed at the beginning of all PLCs being linked, and the same values should be set.
- This instruction should be specified with special internal relay R9014 as the differential execution condition.
- Execution of this instruction does not affect the system register setting contents.
- Enter the first and second keywords after "M" with right-aligned 12 characters.

  Separate the first keyword and second keyword with a comma (,) without inserting a space. An operation error occurs.

  Example: When entering "SYS1, M COMM1,WAIT2"

  ```
  Enter"M _ _ C O M 1 , W A I T 2 ".
        1 2 3 4 5 6 7 8 9 101112
  ```

  Put spaces after M to enter right-aligned 12 characters.

Precautions when setting the link entry waiting time

- This should be set to be at least twice that of the largest scan time of each PLC to be linked.
- If set to a shorter value, there may be some PLCs that are not be able to join the link, even if they are powered on.
- If there are any stations that have not joined the link, the settings should not be changed, especially if there are no problems, even if the link transmission cycle time is longer as a result. (The default value is 400 ms.)

Precautions when setting the error detection time for the transmission assurance relay

- This should be set to be at least twice that of the largest transmission cycle time when all PLCs are linked.
- If set to a shorter value, there is a possibility that the transmission assurance relay will malfunction.
- The settings should not be changed, especially if there are no problems, even if the transmission assurance relay detection time is longer as a result. (The default value is 6400 ms.)

Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when a character other than a keyword is specified.
	Turns ON when there is no comma between the first and second keywords.

Name	Description
	Turns ON when the keyword is specified in lower-case alphabet characters.
	Turns ON when a value outside the specification range is specified.

8.5 SYS1 (MEWTOCOL-COM Response Control)

■ Instruction format

■ Operands

Item	Setting
S	Character constant

■ Devices that can be specified (indicated by ●)

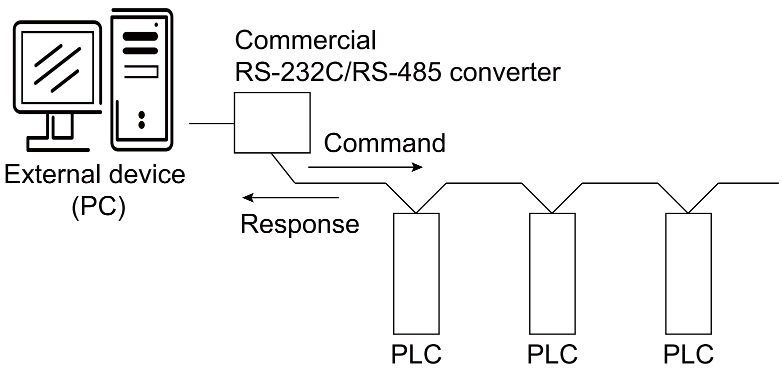
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer device
S														●			

■ Outline of operation

- The MEWTOCOL-COM response time of the port specified by the first keyword is delayed based on the contents specified by the second keyword.
- This instruction is used to delay the response time on the PLC side until a state is reached in which commands can be sent by an external device and responses can be received from the PLC.

Usage example:

When a commercial RS-232C/RS-485 converter is being used to carry out communication between a computer and the PLC, this instruction is used to return the PLC response after switching of the enable signal has been completed on the converter side.



■ Specifying keywords

SYS1," TOOL, WAITn"

Used ports
TOOL: Tool port
COM1: COM port 1

Response time
WAIT0～WAIT999(n:0～999)

- If the communication mode has been set to computer link mode or MODBUS RTU mode
  Set time = scan time x n (n: 0 to 999)
- If the communication mode has been set to PC (PLC) link mode
  Set time = n μs (n: 0 to 999)
- If n = 0, the delay time set by this instruction will be set to "None".

■ Precautions for programming

Because PC (PLC) links may become unstable, do not change settings unless absolutely necessary.

- This instruction is valid only if the setting on the controller side has been set to computer link mode or PC (PLC) link mode.
- Set all the PLCs to be linked to the same value so that execution occurs at the rise of R9014 at the beginning of the program.
- Executing this instruction does not change the settings in the system registers.
- If the settings are changed, set to approximately double or more.
- It is recommended to use differential execution for this instruction.
- When the power supply to the PLC turns OFF, the settings set by this instruction are cleared. (The set value becomes 0.)
  However, the settings will be retained if the mode is switched to "PROG. mode" after this instruction has been executed.
- If a commercial RS-232C/RS-485 converter is being used in PC (PLC) link mode, this instruction should be programmed in all of the connected stations (PLCs).
- Enter the first and second keywords after "M" with right-aligned 12 characters.
  Separate the first keyword and second keyword with a comma (,) without inserting a space. An operation error occurs.
  Example: When entering "SYS1, M COM1,WAIT2"

```
Enter"M _ _ C O M 1 , W A I  T  2 ".
        1 2 3 4  5 6 7 8 9 10 11 12
```

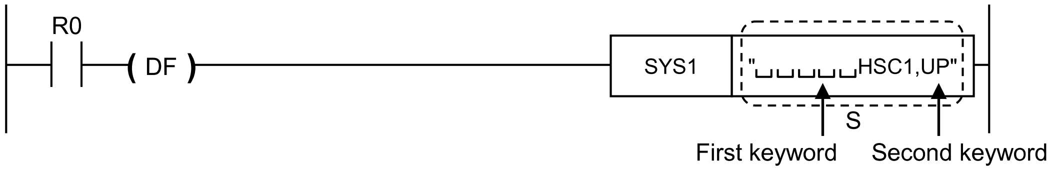
  Put spaces after M to enter right-aligned 12 characters.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when a character other than a keyword is specified.
	Turns ON when there is no comma between the first and second keywords.
	Turns ON when the keyword is specified in lower-case alphabet characters.
	Turns ON when no communication cassette has been installed when COM1 has been set.

8.6 SYS1 (Change High-speed Counter Operation Mode)

■ Instruction format



■ Operands

Item	Setting
S	Character constant

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer device
S														●			

■ Outline of operation

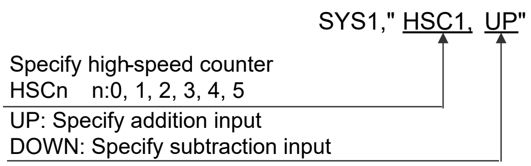
The high-speed counter operation mode specified by the first keyword is changed to the operation mode specified by the second keyword. It is possible to switch between addition input and subtraction input.

■ Operation example

Operation of instruction format description program

When R0 turns ON, the operation mode of high-speed counter CH0 is set to addition mode.

■ Specifying keywords



■ Precautions for programming

- With this instruction, if the high-speed counter system register setting is neither addition input nor subtraction input, an operation error is returned. Specify the system register setting to addition or subtraction in advance. Also, when addition input is specified, even if addition input is specified again, the setting remains addition input. This is the same when subtraction input is specified.
- Executing this instruction does not rewrite the contents of the system ROM of the main unit. As a result, turning the power supply OFF and then ON again rewrites the contents of the system registers specified by the programming tool software.
- It is recommended to use differential execution for this instruction.

- When UP or DOWN has been specified, the contents of the system registers change in accordance with the specification, meaning a verification error may occur in some cases when the program is verified. When BOTH has been specified, the contents of the system registers do not change.
- Enter the first and second keywords after "M" with right-aligned 12 characters.

  Separate the first keyword and second keyword with a comma (,) without inserting a space. An operation error occurs.

  Example: When entering "SYS1, M COM1,WAIT2"

  Enter"M _ _ C O M 1 , W A I T 2 ".
  1 2 3 4 5 6 7 8 9 10 11 12

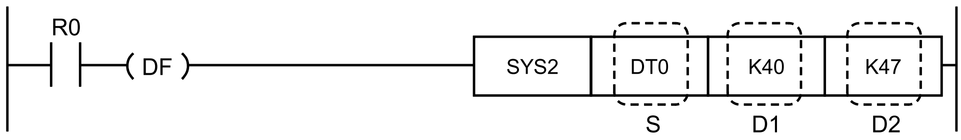
  Put spaces after M to enter right-aligned 12 characters.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when a character other than a keyword is specified.
	Turns ON when there is no comma between the first and second keywords.
	Turns ON when the keyword is specified in lower-case alphabet characters.
	When the system register setting is something other than addition input or subtraction input

8.7 SYS2 (System Register Change Instruction)

■ Instruction format



■ Operands

Item	Setting
S	Starting number of area storing 16-bit data
D1	Starting number of the system register to be specified (K40 to K47, K50 to K57)
D2	Ending number of the system register to be specified (K40 to K47, K50 to K57)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer device
S							●											
D1													●					
D2													●					

■ Outline of operation

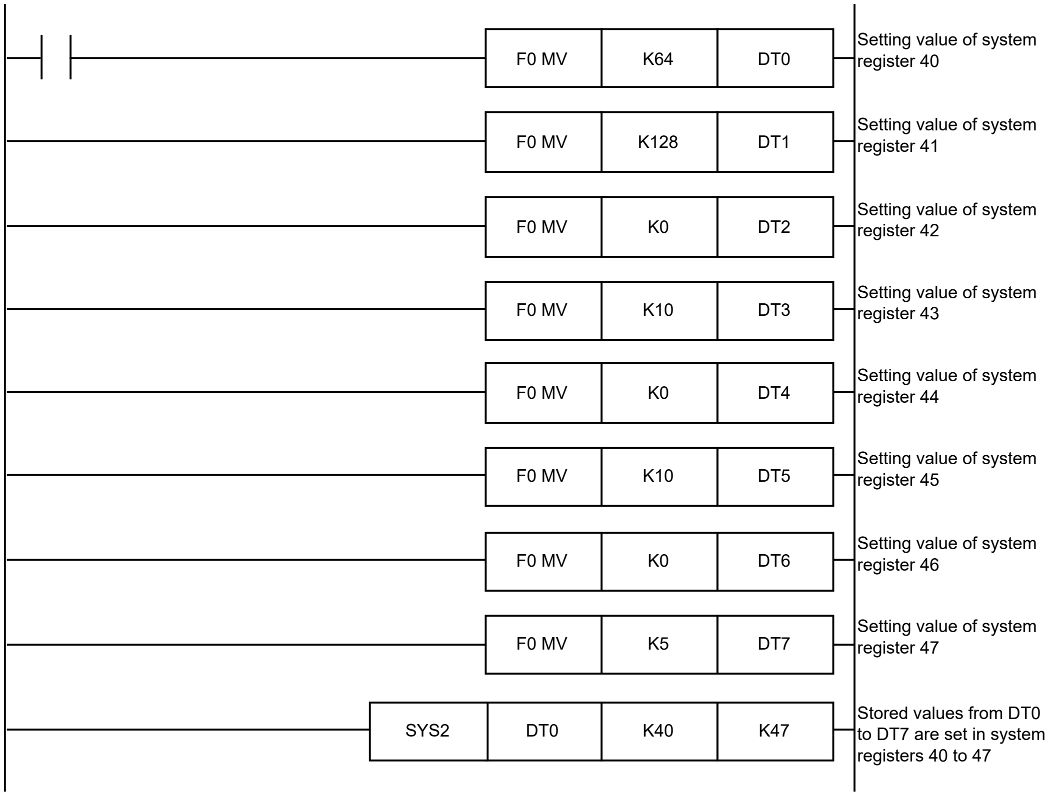
The contents of system registers No. 40 to 47 and No. 50 to 57 are changed to the contents of the data register starting with [S].

■ System registers No. 40 to 47, No. 50 to 57

	No.	Name	Sett values/range
PC (PLC) W0-0 setting	40	Range used by link relay	0 to 64 words
	41	Range used by link register	0 to 128 words
	42	Link relay transmission starting No.	0 to 63
	43	Link relay transmission size	0 to 64 words
	44	Link register transmission starting No.	0 to 127
	45	Link register transmission size	0 to 127 words
	46	PC (PLC) link switch flag	0: Standard, 1: Reverse
	47	MEWNET-W0 PC (PLC) link maximum station number specification	1 to 16
PC (PLC) W0-1 setting	50	Range used by link relay	0 to 64 words
	51	Range used by link register	0 to 128 words

	No.	Name	Sett values/range
	52	Link relay transmission starting No.	64 to 127
	53	Link relay transmission size	0 to 64 words
	54	Link register transmission starting No.	128 to 255
	55	Link register transmission size	0 to 127 words
	57	MEWNET-W0 PC (PLC) link maximum station number specification	1 to 16

■ Program example



■ Precautions for programming

- Executing this instruction does not rewrite the contents of the system ROM of the main unit. As a result, when the power supply is turned OFF and ON again, the contents of the system registers set with the tool software are rewritten.
- Specify a value between K40 and K47 or between K50 and K57 for [D1] or [D2]. Ensure that D1 is less than or equal to D2.
- Since the value of the system register is changed, a verification error may occur during program verification.

■ Flag operations

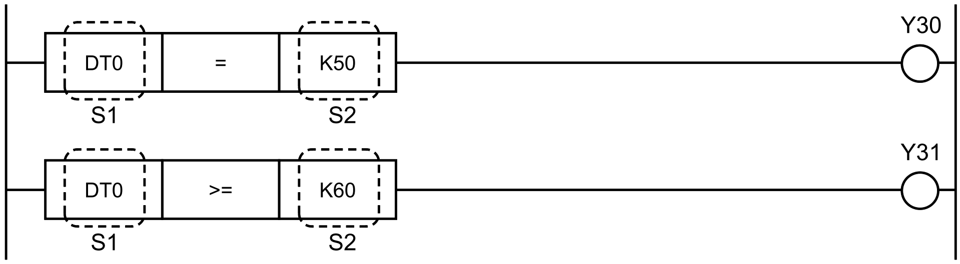
Name	Description
R9007 R9008 (ER)	Turns ON when D1>D2
	Turns ON when a set value is outside the specification range of a system register setting value.

(MEMO)

9 Compare Contact Instructions

9.1 ST=, ST <>, ST>, ST>=, ST<, ST<= [16-bit Data Comparison (Start)]

■ Instruction format



■ Operands

Items	Settings
S1	Comparison data 1: Number of area storing 16-bit data, or constant data
S2	Comparison data 2: Number of area storing 16-bit data, or constant data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

- The signed 16-bit data specified by [S1] is compared with the signed 16-bit data specified by [S2].
- If the comparison results in one of the specified statuses (=, <, >, etc.), a logical operation is initiated with the contacts operating as liaison contacts.
- Comparison results and operations relate as follows.

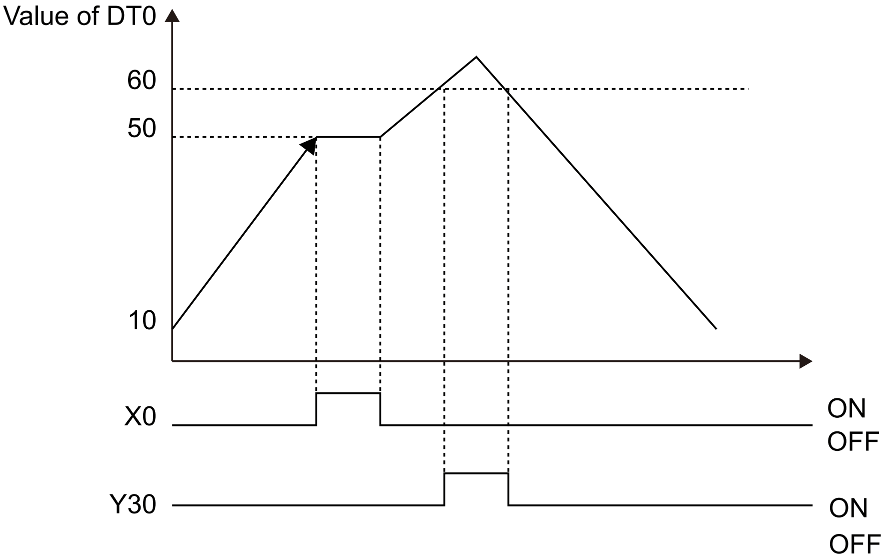
Comparison instruction	Relationship between S1 and S2		
	S1 < S2	S1 = S2	S1 > S2
ST=	OFF	ON	OFF
ST<>	ON	OFF	ON
ST>	OFF	OFF	ON
ST>=	OFF	ON	ON
ST<	ON	OFF	OFF
ST<=	ON	ON	OFF

(Note 1) "<>" is displayed as "≠".
">=" is displayed as "≥".
"<=" is displayed as "≤".

Operation example

Operation of instruction format description program

Compares the value of data register DT0 with K50. If DT0 = K50, external output Y30 turns ON. Compares the value of DT0 with K60. If DT0 ≥ K60, Y31 turns ON.



Precautions for use

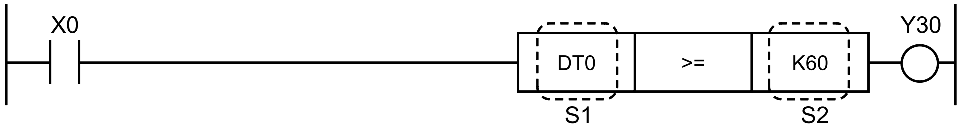
- These instructions start from the bus bar.
- In the case of BCD data, etc., data is compared as a negative value if the most significant bit is 1, so the comparison results may not be accurate. In cases such as this, compare after converting the data to binary data by using an instruction such as F81 BIN.

Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

9.2 AN=, AN<>, AN>, AN>=, AN<, AN<= [16-bit Data Comparison (AND)]

■ Instruction format



■ Operands

Items	Settings
S1	Comparison data 1: Number of area storing 16-bit data, or constant data
S2	Comparison data 2: Number of area storing 16-bit data, or constant data

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	

■ Outline of operation

- The signed 16-bit data specified by [S1] is compared with the signed 16-bit data specified by [S2].
- If the comparison results in one of the specified statuses (=, <, >, etc.), the contacts are connected in series as liaison contacts.
- Comparison results and operations relate as follows.

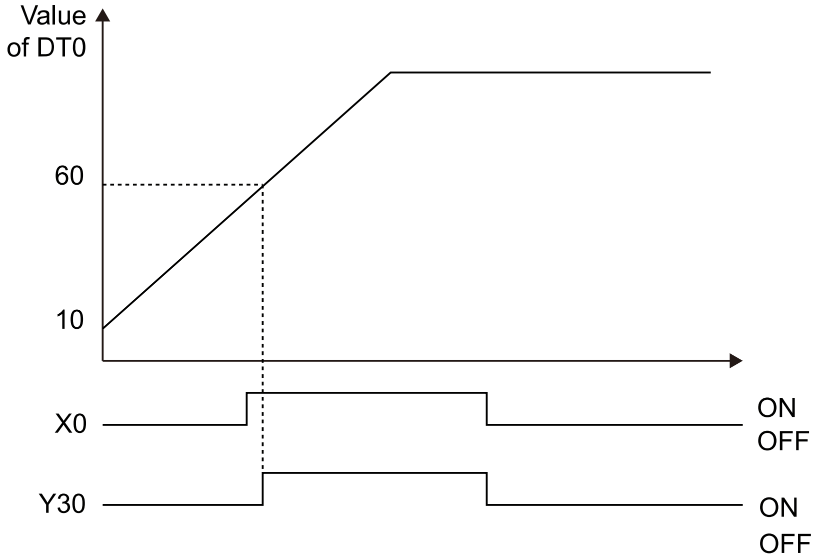
Comparison instruction	Relationship between S1 and S2		
	S1 < S2	S1 = S2	S1 > S2
AN=	OFF	ON	OFF
AN<>	ON	OFF	ON
AN>	OFF	OFF	ON
AN>=	OFF	ON	ON
AN<	ON	OFF	OFF
AN<=	ON	ON	OFF

(Note 1) "<>" is displayed as "≠".
">=" is displayed as "≥".
"<=" is displayed as "≤".

■ Operation example

Operation of instruction format description program

When internal relay X0 turns ON, the value of DT0 and K60 are compared, and if DT0 is equal to or greater than K60, the external output Y30 turns ON. If X0 is OFF or if DT0 is less than K60, Y30 turns OFF.



■ Precautions for use

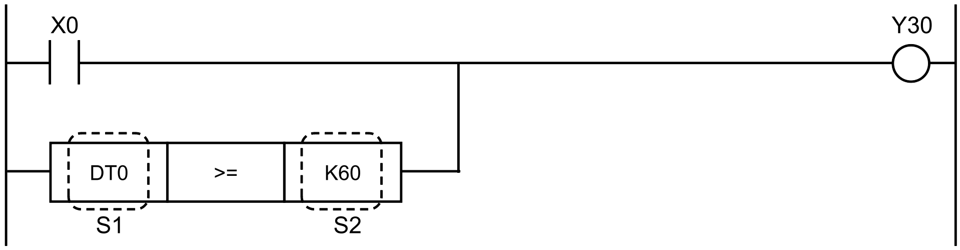
- These instructions can be used consecutively.
- In the case of BCD data, etc., data is compared as a negative value if the most significant bit is 1, so the comparison results may not be accurate. In cases such as this, compare after converting the data to binary data by using an instruction such as F81 BIN.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

9.3 OR= OR <> OR > OR >= OR < OR <= [16-bit Data Comparison (OR)]

■ Instruction format



■ Operands

Items	Settings
S1	Comparison data 1: Number of area storing 16-bit data, or constant data
S2	Comparison data 2: Number of area storing 16-bit data, or constant data

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	

■ Outline of operation

- The signed 16-bit data specified by [S1] is compared with the signed 16-bit data specified by [S2].
- When comparison results are the specified status (=, <, >, etc.), a parallel connection occurs as the conductive contact.
- Comparison results and operations relate as follows.

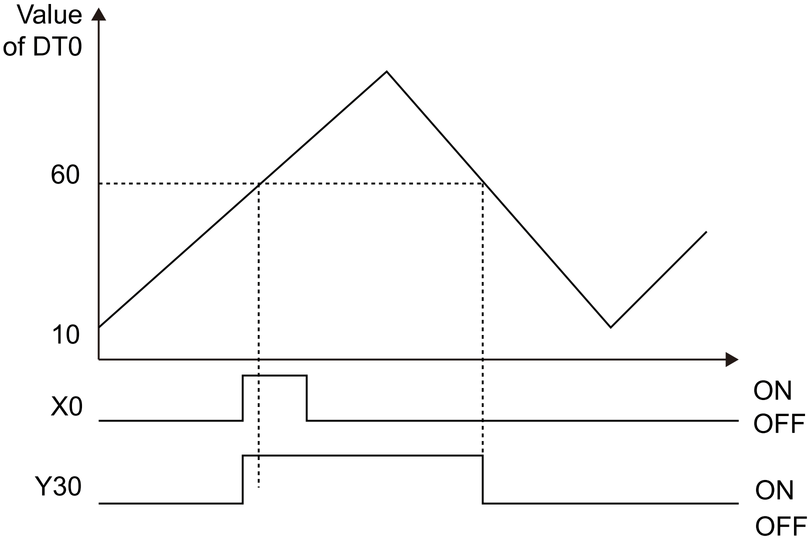
Comparison instruction	Relationship between S1 and S2		
	S1 < S2	S1 = S2	S1 > S2
OR=	OFF	ON	OFF
OR<>	ON	OFF	ON
OR>	OFF	OFF	ON
OR>=	OFF	ON	ON
OR<	ON	OFF	OFF
OR<=	ON	ON	OFF

(Note 1) "<>" is displayed as "≠".
">=" is displayed as "≥".
"<=" is displayed as "≤".

Operation example

Operation of instruction format description program

When external input X0 turns ON, or the result of comparison between the value of DT0 and K60 is DT0 ≥ K60, external output Y30 turns ON. If X0 is OFF and DT0 < K60, Y30 turns OFF.



Precautions for use

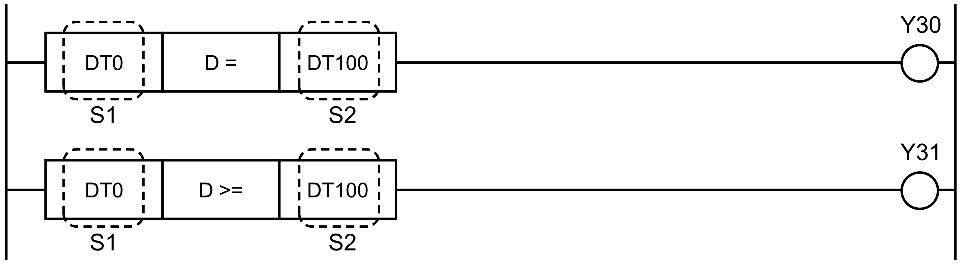
- These instructions start from the bus bar.
- These instructions can be used consecutively.
- In the case of BCD data, etc., data is compared as a negative value if the most significant bit is 1, so the comparison results may not be accurate. In cases such as this, compare after converting the data to binary data by using an instruction such as F81 BIN.

Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

9.4 STD=, STD<>, STD>, STD>=, STD<, STD<= [32-bit Data Comparison(start)]

■ Instruction format



■ Operands

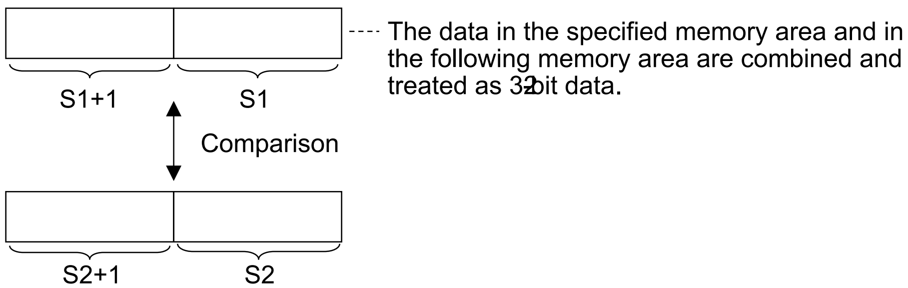
Items	Settings
S1	Comparison data 1: Area number storing the 32-bit data, or constant data
S2	Comparison data 2: Area number storing the 32-bit data, or constant data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

- Compares the signed 32-bit data of the combined area of [S1] and [S1+1] with the signed 32-bit data of the combined area of [S2] and [S2+1].
- If the comparison results in one of the specified statuses (=, <, >, etc.), a logical operation is initiated with the contacts operating as liaison contacts.
- The relationship between comparison results and operation is the same as"9.1 ST=, ST <>, ST>, ST>=, ST<, ST<= [16-bit Data Comparison (Start)]".
- Memory area is specified by the memory area number of the lower order hexadecimal part.



■ Operation example

Operation of instruction format description program

The 32-bit value that is a combination of data registers DT0 and DT1 is compared with the 32-bit value that is a combination of DT100 and DT101, and if (DT0, DT1) = (DT100, DT101), external output Y30 turns ON. If (DT0, DT1) is greater than (DT100, DT101), Y31 turns ON.

■ Precautions for use

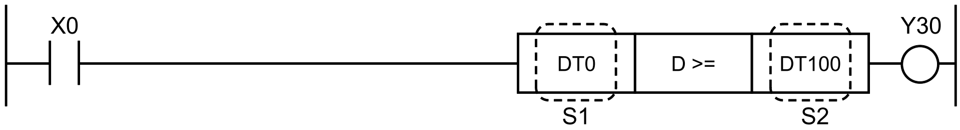
- These instructions start from the bus bar.
- In the case of BCD data, etc., data is compared as a negative value if the most significant bit is 1, so the comparison results may not be accurate. In these instances, use the F83 DBIN instruction or similar to convert to binary data before comparison.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

9.5 AND=, AND<>, AND>, AND>=, AND<, AND<= [32-bit Data Comparison (AND)]

■ Instruction format



■ Operands

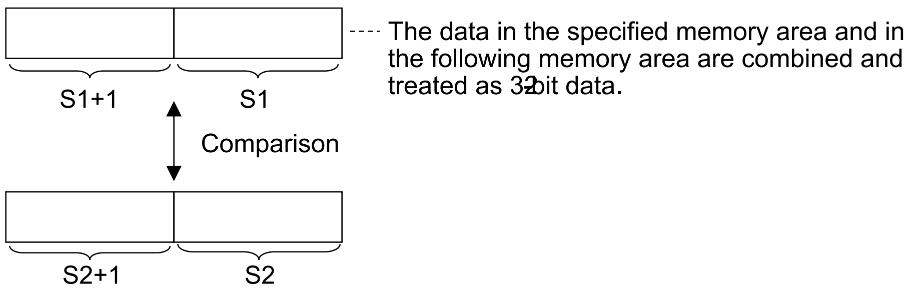
Items	Settings
S1	Comparison data 1: Area number storing the 32-bit data, or constant data
S2	Comparison data 2: Area number storing the 32-bit data, or constant data

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	

■ Outline of operation

- Compares the signed 32-bit data of the combined area of [S1] and [S1+1] with the signed 32-bit data of the combined area of [S2] and [S2+1].
- If the comparison results in one of the specified statuses (=, <, >, etc.), the contacts are connected in series as liaison contacts.
- The relationship between comparison results and operation is the same as"9.2 AN=, AN<>, AN>, AN>=, AN<, AN<= [16-bit Data Comparison (AND)]".
- Memory area is specified by the memory area number of the lower order hexadecimal part.



■ Operation example

Operation of instruction format description program

When the external input X0 is ON, and when the comparison result of the combined 32-bit values of data registers DT0 and DT1 and the combined 32-bit values of DT100 and DT101 is

(DT0, DT1) ≥ (DT100, DT101), the external output Y30 turns ON. If X0 is OFF or if (DT0, DT1) is less than (D100, D101), Y30 turns OFF.

■ Precautions for use

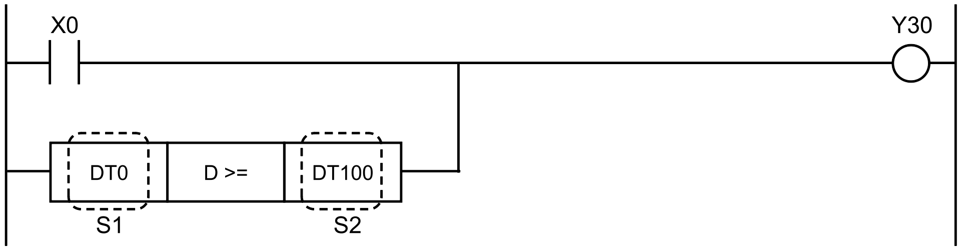
- These instructions can be used consecutively.
- In the case of BCD data, etc., data is compared as a negative value if the most significant bit is 1, so the comparison results may not be accurate. In these instances, use the F83 DBIN instruction or similar to convert to binary data before comparison.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

9.6 ORD=, ORD<>, ORD>, ORD>=, ORD<, ORD<= [32-bit Data Comparison (OR)]

■ Instruction format



■ Operands

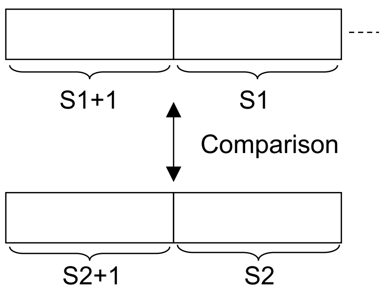
Items	Settings
S1	Comparison data 1: Area number storing the 32-bit data, or constant data
S2	Comparison data 2: Area number storing the 32-bit data, or constant data

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	

■ Outline of operation

- This compares signed 32-bit data for the combined [S1] and [S1+1] area with the signed 32-bit data for the combined [S2] and [S2+1] area.
- When comparison results are the specified status (=, <, >, etc.), a parallel connection occurs as the conductive contact.
- The relationship between comparison results and operation is the same as"9.3 OR= OR <> OR > OR >= OR < OR <= [16-bit Data Comparison (OR)]".
- Memory area is specified by the memory area number of the lower order hexadecimal part.



The data in the specified memory area and in the following memory area are combined and treated as 32-bit data.

■ Operation example

Operation of instruction format description program

When external input X0 turns ON, or when (DT0, DT1) ≥ (DT100, DT101) after a comparison between the 32-bit value from combining data register DT0 and DT1 and the 32-bit value from combining data register DT100 and DT101, then the external output Y30 is ON. If X0 is OFF and (DT0, DT1) < (DT100, DT101), then Y30 turns OFF.

■ Precautions for use

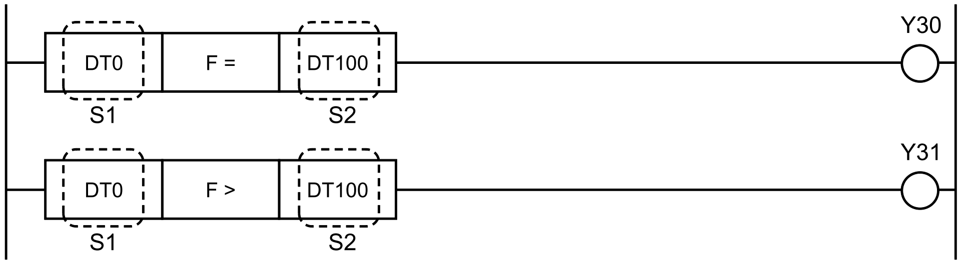
- These instructions start from the bus bar.
- These instructions can be used consecutively.
- In the case of BCD data, etc., data is compared as a negative value if the most significant bit is 1, so the comparison results may not be accurate. In these instances, use the F83 DBIN instruction or similar to convert to binary data before comparison.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

9.7 STF=, STF<>, STF>, STF>=, STF< and STF<= [Floating Point Real Number Data Comparison (start)]

■ Instruction format



■ Operands

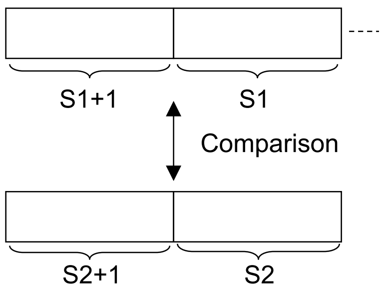
Item	Setting
S1	Area storing real number data, or real number data (comparison data 1) (two words)
S2	Area storing real number data, or real number data (comparison data 2) (two words)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	SWR	SDT	Constant				Index modifier	Integer device
													K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●

■ Outline of operation

- These instructions compare the real number data in the area combining [S1] and [S1+1] with the real number data in the area combining [S2] and [S2+1].
- If the comparison results in one of the specified statuses (=, <, >, etc.), a logical operation is initiated with the contacts operating as liaison contacts.
- The relationship between comparison results and operation is the same as "9.1 ST=, ST <>, ST>, ST>=, ST<, ST<= [16-bit Data Comparison (Start)]".
- Memory area is specified by the memory area number of the lower order hexadecimal part.



The data in the specified memory area and in the following memory area are combined and treated as single precision real number data.

Operation example

Operation of instruction format description program

The real number that is a combination of data registers DT0 and DT1 is compared with the real number that is a combination of data registers DT100 and DT101, and if (DT0, DT1) is equal to (DT100, DT101), external output Y30 turns ON. If (DT0, DT1) is greater than (DT100, DT101), Y31 turns ON.

Precautions for use

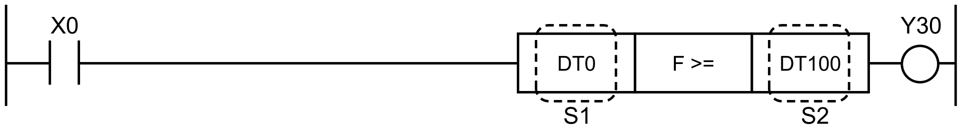
- These instructions start from the bus bar.
- If [S1] and [S2] are specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.
- If a K constant is specified for [S1] or [S2], the same processing is performed as when an integer device is specified.

Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real-number data is specified in [S1, S1+1] or [S2, S2+1].

9.8 ANF=, ANF<>, ANF>, ANF>=, ANF<, ANF<= [Floating Point Real Number Data Comparison (AND)]

■ Instruction format



■ Operands

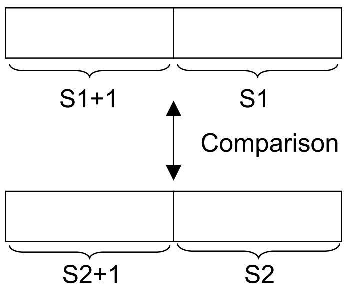
Item	Setting
S1	Area storing real number data, or real number data (comparison data 1) (two words)
S2	Area storing real number data, or real number data (comparison data 2) (two words)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer device
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●

■ Outline of operation

- These instructions compare the real number data in the area combining [S1] and [S1+1] with the real number data in the area combining [S2] and [S2+1].
- If the comparison result is one of the specified statuses (=, >, <, etc.), the contacts are connected in series as liaison contacts.
- The relationship between comparison results and operation is the same as "9.2 AN=, AN<>, AN>, AN>=, AN<, AN<= [16-bit Data Comparison (AND)]".
- Memory area is specified by the memory area number of the lower order hexadecimal part.



The data in the specified memory area and in the following memory area are combined and treated as single precision real number data.

■ Operation example

Operation of instruction format description program

When external input X0 turns ON, the real number that is a combination of data registers DT0 and DT1 is compared with the real number that is a combination of data registers DT100 and

DT101, and if (DT0, DT1) is equal to or greater than (DT100, DT101), external output Y30 turns ON. If X0 is OFF or if (DT0, DT1) is less than (D100, D101), Y30 turns OFF.

■ Precautions for use

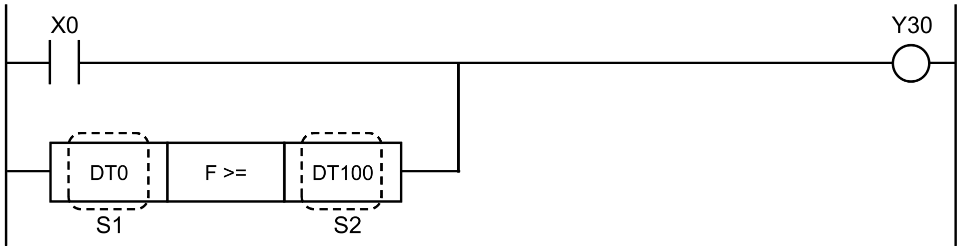
- These instructions can be used consecutively.
- If [S1] and [S2] are specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.
- If a K constant is specified for [S1] or [S2], the same processing is performed as when an integer device is specified.

■ Flag operations

Name	Description
R9007	Turns ON when the area is exceeded in index modification.
R9008 (ER)	Turns ON when non-real-number data is specified in [S1, S1+1] or [S2, S2+1].

9.9 ORF=, ORF<>, ORF>, ORF>=, ORF<, ORF<= [Floating Point Real Number Data Comparison (OR)]

■ Instruction format



■ Operands

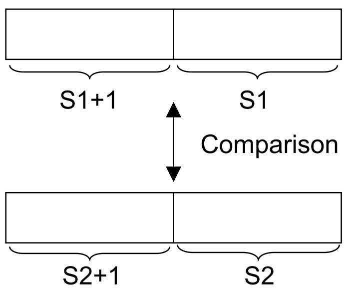
Item	Setting
S1	Area storing real number data, or real number data (comparison data 1) (two words)
S2	Area storing real number data, or real number data (comparison data 2) (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer device
S1	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
S2	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•

■ Outline of operation

- These instructions compare the real number data in the area combining [S1] and [S1+1] with the real number data in the area combining [S2] and [S2+1].
- If the comparison result is in the specified status (=, >, <, …), it is connected in parallel as a conducting contact.
- The relationship between comparison results and operation is the same as "9.3 OR= OR <> OR > OR >= OR < OR <= [16-bit Data Comparison (OR)]".
- Memory area is specified by the memory area number of the lower order hexadecimal part.



The data in the specified memory area and in the following memory area are combined and treated as single precision real number data.

■ Operation example

Operation of instruction format description program

If external input X0 is ON, or if the real number values of combined data registers DT0 and DT1 and the real number values of combined data registers DT100 and DT101 are compared and (DT0, DT1) ≥ (DT100, DT101), then the external output Y30 turns ON. If X0 is OFF and (DT0, DT1) < (DT100, DT101), then Y30 turns OFF.

■ Precautions for use

- These instructions start from the bus bar.
- These instructions can be used consecutively.
- If [S1] and [S2] are specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.
- If a K constant is specified for [S1] or [S2], the same processing is performed as when an integer device is specified.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real-number data is specified in [S1, S1+1] or [S2, S2+1].

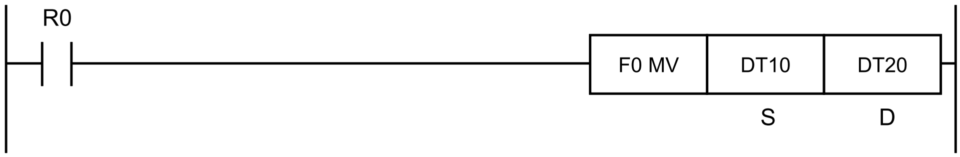
(MEMO)

10 Transfer Instructions

10.1 F0 MV (16-bit Data Transfer)

Transfers the 16-bit data in the specified area number.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the hexadecimal data or constant data
D	Area where data is transferred to

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•	•	•					•	

■ Outline of operation

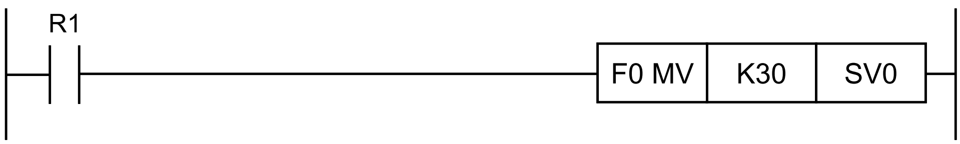
- The 16-bit data in the memory area specified by [S] is transferred to the memory area specified by [D].

■ Operation example

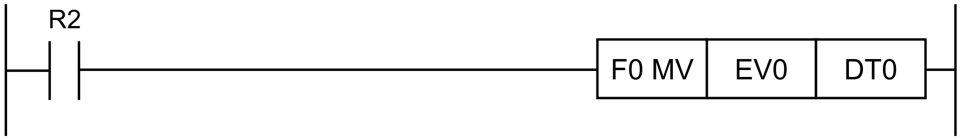
Example 1: Instruction format and described program operation

- When the internal relay R0 turns ON, the content of data register DT10 is transferred to data register DT20.

Example 2: Constant K30 is transferred to the timer 0 setting value area when internal relay R1 turns ON



Example 3: The timer 0 elapsed value is transferred to data register DT0 when R2 turns ON



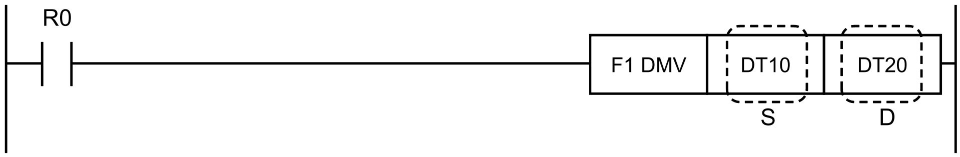
Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

10.2 F1 DMV (32-bit Data Transfer)

Transfers 32-bit data to the specified area number.

■ Instruction format



■ Operands

Items	Settings
S	Area storing 32-bit data, or constant data
D	Area where data is transferred to

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

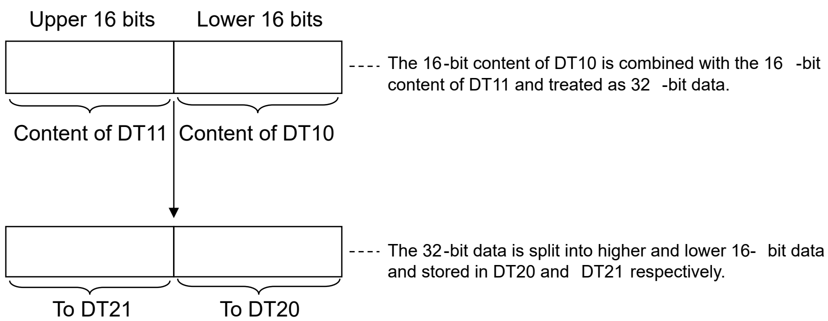
The 32-bit data in the memory area specified by [S] is transferred to the memory area specified by [D].

■ Operation example

Operation of instruction format description program

When the internal relay R0 turns ON, the content of data register DT10 and DT11 is transferred to data register DT20 and DT21.

- Specify a lower 16-bit memory area for the memory area.



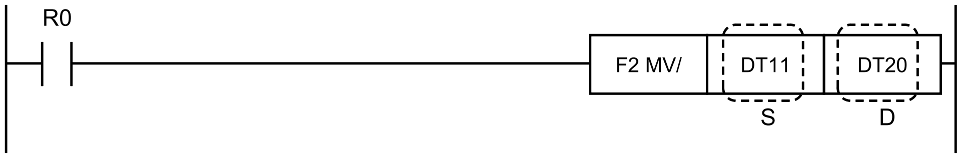
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

10.3 F2 MV/ (16-bit Data Inversion and Transfer)

Inverts and transfers 16-bit data at the specified area number.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the hexadecimal data or constant data
D	Area where data is transferred to

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

The 16-bit data in the area specified by [S] is logically inverted (0⇔1 inversion) and transferred to the area specified by [D].

BIN	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0
HEX	0				4				D				2			

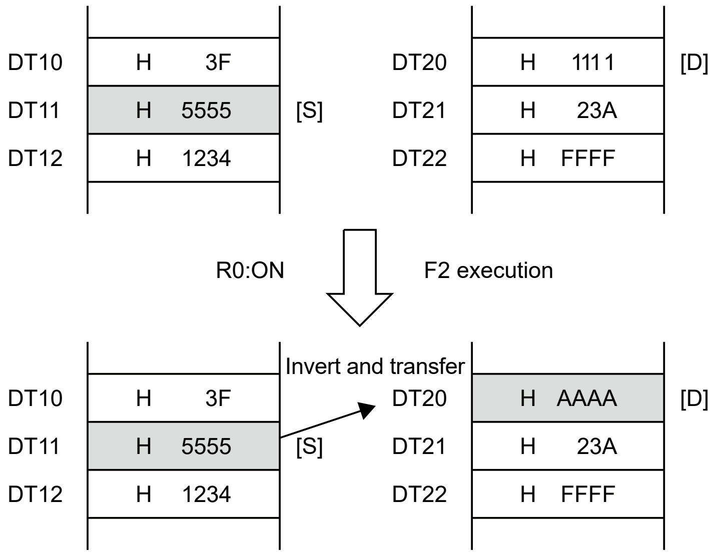
⇩

BIN	1	1	1	1	1	0	1	1	0	0	1	0	1	1	0	1
HEX	F				B				2				D			

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the contents of data register DT11 are logically inverted and transferred to data register DT20.



DT11 = "0101 0101 0101 0101" (H5555)
↓Invert and transfer
DT20 = "1010 1010 1010 1010" (HAAAA)

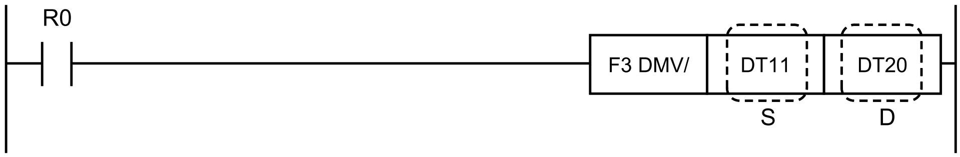
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

10.4 F3 DMV/ (32-bit Data Inversion and Transfer)

Inverts the 32-bit data in the specified area number and transfers it.

■ Instruction format



■ Operands

Items	Settings
S	Area storing 32-bit data, or constant data
D	Area where data is transferred to

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

The 32-bit data in the area specified by [S] is logically inverted (0⇔1 inversion) and transferred to the area specified by [D].

S	1	1	0	0	1	1	0	1	0	0	0	1	0	1	0	1
S+1	0	0	0	0	0	1	1	1	0	1	0	1	1	0	1	1

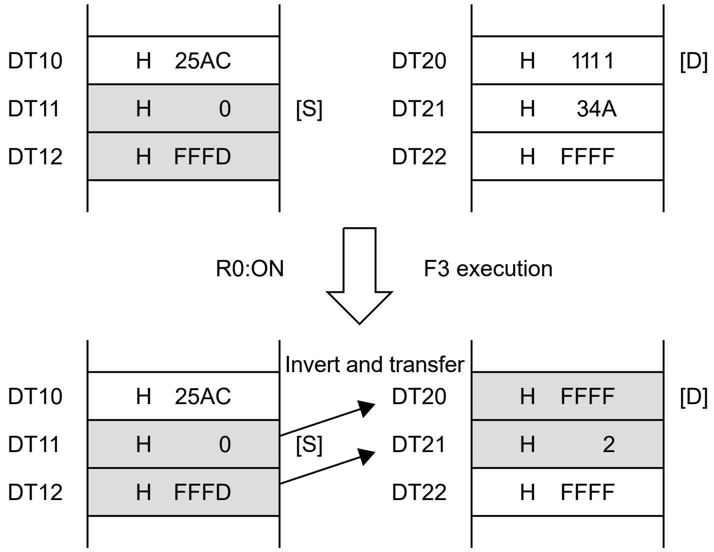
⇩

D	0	0	1	1	0	0	1	0	1	1	1	0	1	0	1	0
D+1	1	1	1	1	1	0	0	0	1	0	1	0	0	1	0	0

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the contents of data registers DT11 and DT12 are logically inverted and transferred to data registers DT20 and DT21.



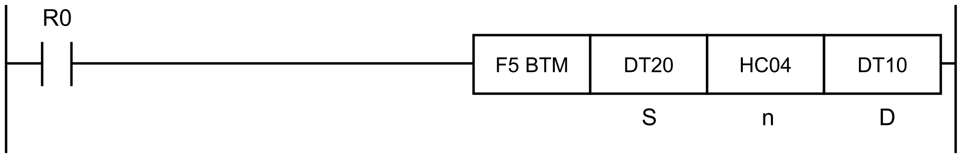
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

10.5 F5 BTM (Bit Data Transfer)

Transfers 1-bit data in the specified 16-bit data to the specified bit.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the hexadecimal data or constant data
n	Area specifying the transfer method
D	Data destination storage area

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
n	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

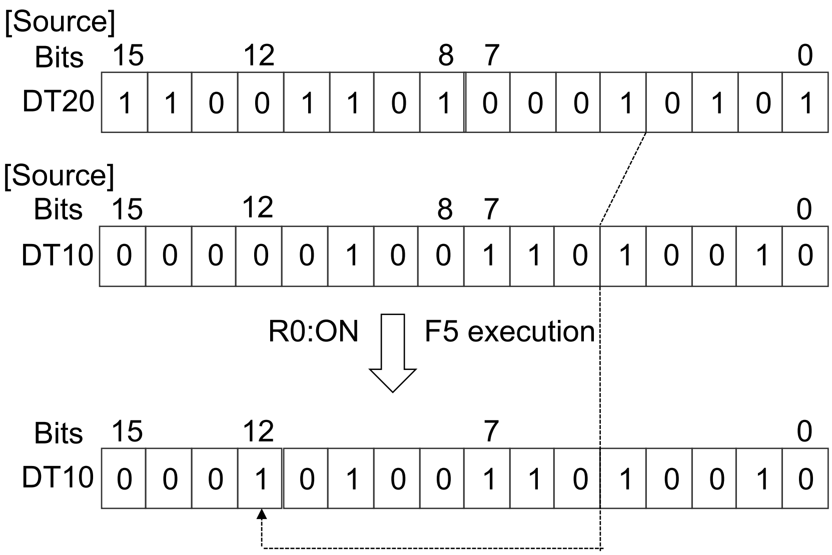
■ Outline of operation

- Transfers the content of one bit ("1"or"0") at any position in the 16-bit data of the area specified by [S] to any bit of the memory area specified by [D]. The bit position is specified by the value of [n].

■ Operation example

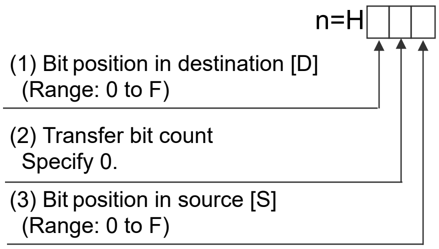
Operation of instruction format description program

- When internal relay R0 turns ON, the content of bit 4 of data register DT20 is transferred to bit 12 of DT10.



■ About transfer method specification [n]

- Specify [n] as an H constant in the following format:



Bit position specification of [S] and [D]

Bits Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Set value (H)	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

For example, specify A to specify bit 10. When transferring bit 4 of [S] to bit 12 of [D], n = HC04.

■ Transferring multiple bits

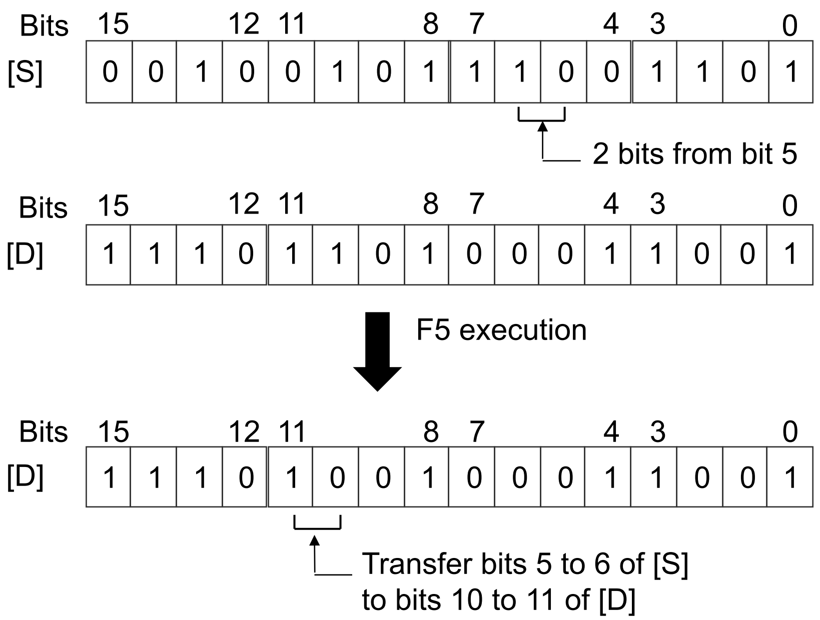
- When the number of transfer bits is specified in n, the specified bits from the position specified by [S] are transferred to the position whose start is specified by [D].
- Up to 16 bits can be transferred. Specify the number of transfer bits as a hexadecimal number. The range is 0 to F (1 bit to 16 bits).

Number of transfer bits	Setting (n)
1 bit	H□0□
2-bit	H□1□

Number of transfer bits	Setting (n)
3 bits	H□2□
4-bit	H□3□
5 bits	H□4□
6 bits	H□5□
7 bits	H□6□
8 bits	H□7□
9 bits	H□8□
10 bits	H□9□
11 bits	H□A□
12 bits	H□B□
13 bits	H□C□
14 bits	H□D□
15 bits	H□E□
16 bits	H□F□

Example 1: When transferring two bits (n = H□1□)

Transfer two bits from [S] bit 5 to [D] bit 10... n = HA15



- When 0 is specified for the number of transfer bits, the single specified bit is transferred.
- If the specified range is outside the area of [S], the contents of the part extending beyond the area are set to 0 and transferred.

Example 2: Transfer four bits from bit 14 of [S] to bit 2 of [D]... n = H23E

Bits			15			12	11			8	7			4	3			0
[S]	0	0	1	1	0	0	1	1	0	1	0	0	0	1	1	0	1	0

→ 4 bits starting from bit 14

Bits	15			12	11			8	7			4	3			0
[D]	0	0	1	0	0	0	0	1	1	0	1	1	1	0	0	1

F5 execution

Bits	15			12	11			8	7			4	3			0
[D]	0	0	1	0	0	0	0	1	1	0	0	0	1	1	0	1

Bits 14 to 15 of [S] are sent to bits 2 to 3 of [D].
0 is stored in bits 4 to 5 of [D].

- If the specified range is outside the area of [D], the part extending beyond the area will not be transferred. Data is not written to the next address.

Example 3: Transfer six bits from bit 6 of [S] to bit 12 of [D]... n = HC56

Bits	15			12	11			8	7			4	3			0
[S]	0	0	0	0	0	1	0	1	1	1	0	1	1	1	1	1

→ 6 bits from bit 6

Bits	15			12	11			8	7			4	3			0
[D]	0	0	0	0	1	0	0	1	1	1	1	1	0	0	1	1

F5 execution

Bits	15			12	11			8	7			4	3			0
[D]	0	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1

Among bits 6 to 11 of [S], bits 6 to 9 are sent to bits 12 to 15 in [D]
(The content of bits 10 to 11 of [S] have no effect)

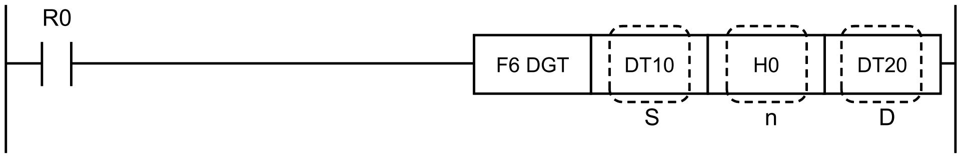
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

10.6 F6 DGT (Digit Data Transfer)

Transfers the specified 16-bit data in 4-bit (digit) units.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the hexadecimal data or constant data
n	Area specifying the transfer method
D	Area where data is transferred to

■ Devices that can be specified (indicated by ●)

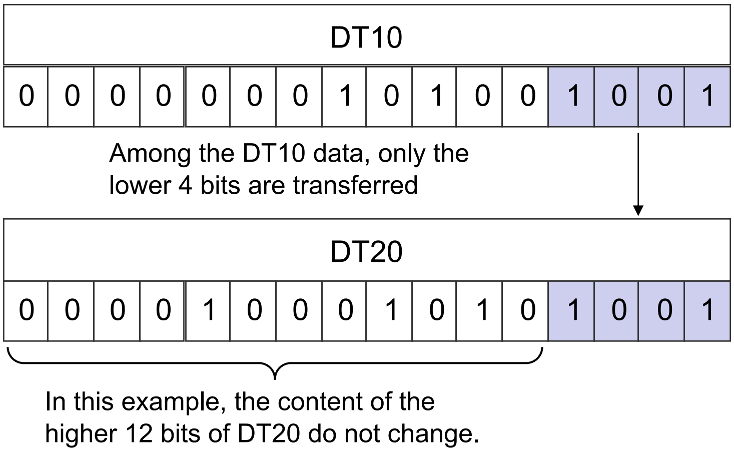
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
n	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

The 16-bit data in the memory area specified by [S] is transferred to the memory area specified by [D], according to the transfer method specified by [n].

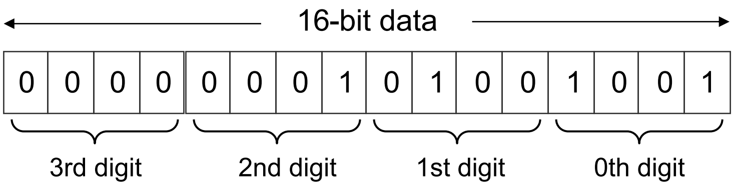
■ Operation example

Operation of instruction format description program



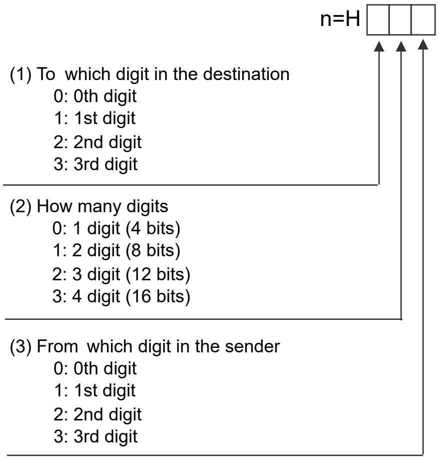
■ What is a digit?

- Digits are units of four bits used when handling data.
- With this instruction, 16-bit data is separated into four digits for convenience. Starting from the lowest four bits, these digits are named digit 0, digit 1, digit 2, and digit 3.



■ About transfer method specification [n]

- For designating
  (1) which digit to transfer to at the transfer destination;
  (2) how many digits to transfer; and
  (3) which digit to transfer from at the transfer source
  with digit transfer.
- Specify [n] as an H constant in the following format:



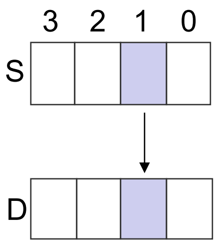
If (1) or (2) is 0, such as"H000"in the program example on the previous page, use the short form"H0".

■ Examples of transfer methods

The following digit transfer patterns are possible based on the specification of [n]:

1. One digit is transferred to a parallel destination

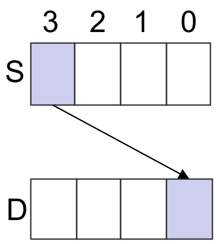
Transferring from digit 1 to digit 1



Set n=H101.

2. One digit is transferred to a non-parallel destination

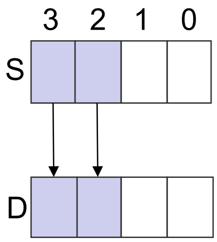
Transferring from digit 3 to digit 0



Set n=H3.
(H003)

3. Multiple digits are transferred to a parallel destination

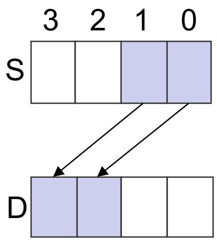
Transferring digits 2 and 3 to digits 2 and 3



Set n=H212.

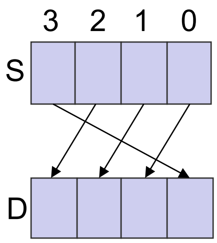
4. Multiple digits are transferred to a non-parallel destination

Transferring digits 0 and 1 to digits 2 and 3



Set n=H210.

5. Four digits are transferred



Set n=H130.

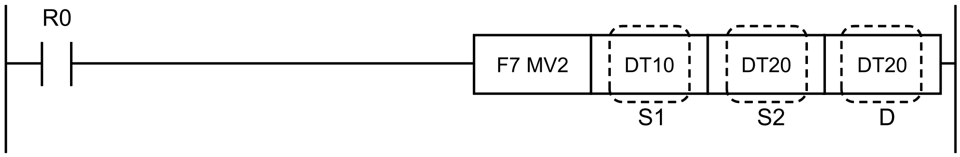
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

10.7 F7 MV2 (Two 16-bit Data Transfer to Single Area)

Two 16-bit data are transferred from the specified area number.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the hexadecimal data or constant data
S2	Area storing the hexadecimal data or constant data
D	Starting address of the data transfer destination (two words)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

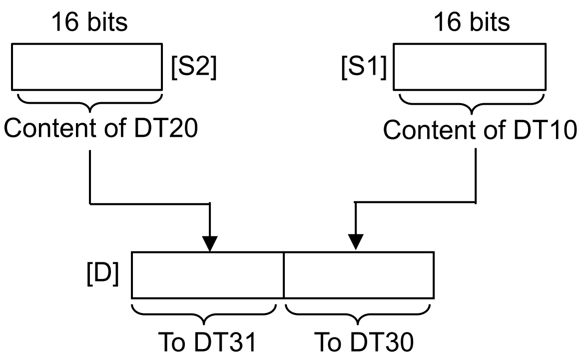
■ Outline of operation

The two 16-bit data (two words) specified by [S1] and [S2] are transferred to the memory area (two words) specified by [D].

■ Operation example

Operation of instruction format description program

When the execution condition R0 turns ON, the contents of data register DT10 is transferred to DT30, and the contents of DT20 is transferred to DT31.



■ Related instructions

Use the F190 MV3 instruction to transfer three types of 16-bit data.

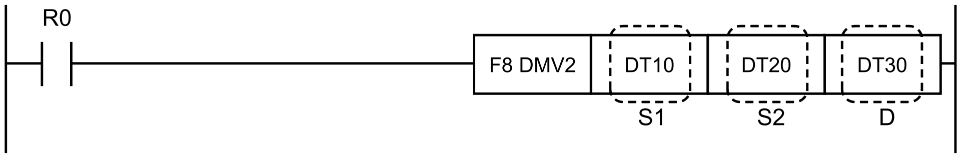
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

10.8 F8 DMV2 (Two 32-bit Data Transfer to Single Area)

Two 32-bit data are transferred from the specified area number.

■ Instruction format



■ Operands

Item	Setting
S1	Area storing 32-bit data, or constant data
S2	Area storing 32-bit data, or constant data
D	Starting address of the data transfer destination area (four words)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	SWR	SDT	Constant				Index modifier	Integer device
													K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●	●							●	

■ Outline of operation

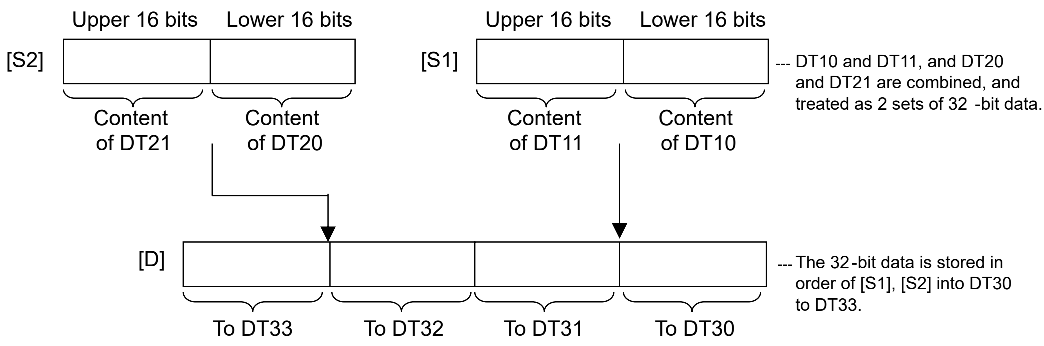
- The two 32-bit data (four words) specified in [S1] and [S2] are transferred to the memory area (four words) specified in [D].
- The specification of [S1] and [S2] specifies the lower 16-bit memory area.
- The specification of [D] specifies the start of the 4 word memory area.

■ Related instructions

Use the F191 DMV3 instruction to transfer three types of 32-bit data.

■ Operation example

Operation of instruction format description program



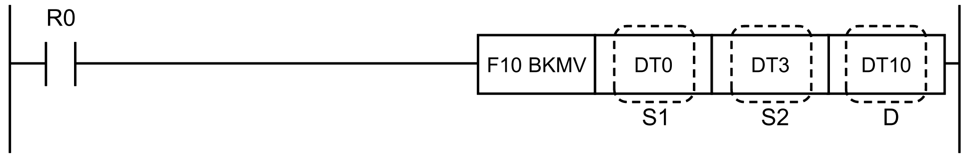
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

10.9 F10 BKMV (Data Block Transfer)

Transfers data at the block unit.

■ Instruction format



■ Operands

Items	Settings
S1	Starting address of the source data
S2	Final address of the source data
D	Data destination storage area

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●		●	●					●	
S2	●	●	●	●	●	●	●	●		●	●					●	
D		●	●	●	●	●	●	●								●	

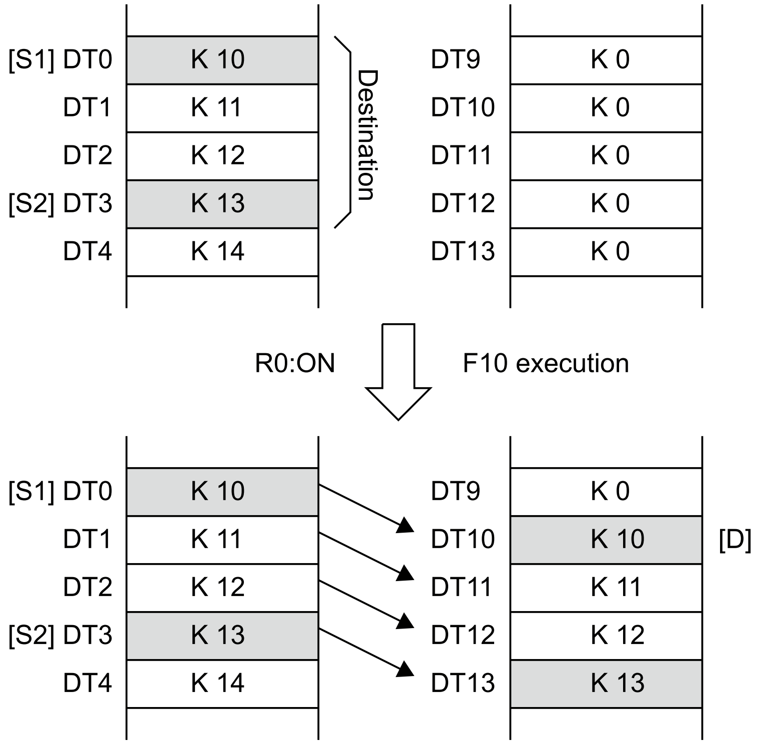
■ Outline of operation

This bulk transfers the data between the area specified by [S1] and the area specified by [S2] to the area specified by [D] and later.

■ Operation example

Operation of instruction format description program

When the internal relay R0 turns ON, the data of data registers DT0 to DT3 is transferred to the data registers DT10 to DT13.



■ Precautions for programming

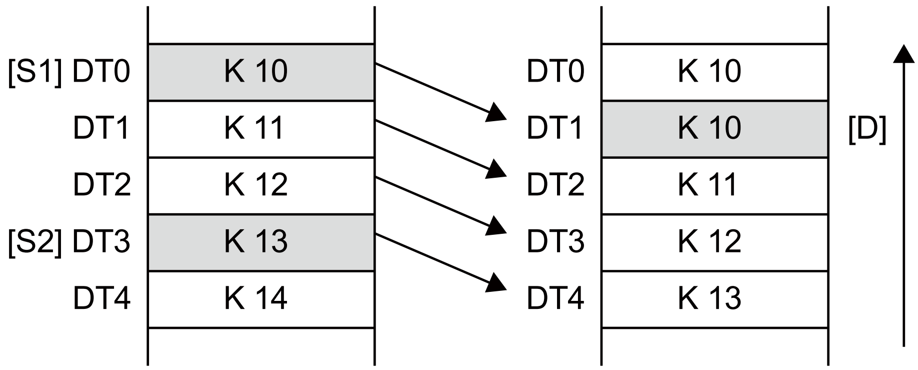
- Specify the same type of memory area for [S1] and [S2].
- Specify the number of the lower address with [S1], and the number of the higher address with [S2].

  If [S1] > [S2] is specified and an instruction executed, an operation error will occur.

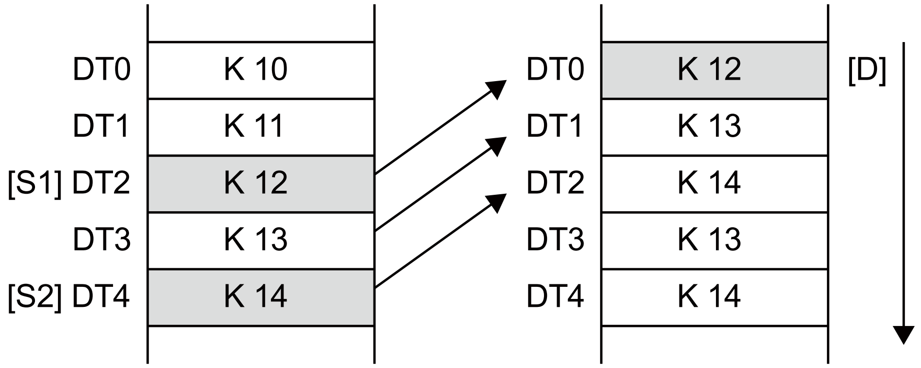
■ Precautions if the same type of memory area is specified for S1, S2, and D

- If [S1] and [D] have the same type and same number of memory area specified, the instruction is not executed.
- If the block being transferred overlaps the destination, transfer results will be overwritten.
- If [S1] < [D], data is transferred starting from the higher address.

  In the following example, the data is stored in the order DT4 > DT3 > DT2 > DT1.



- If [S1] > [D], data is transferred starting from the lower address.
  In the following example, the data is stored in the order DT0 > DT1 > DT2.



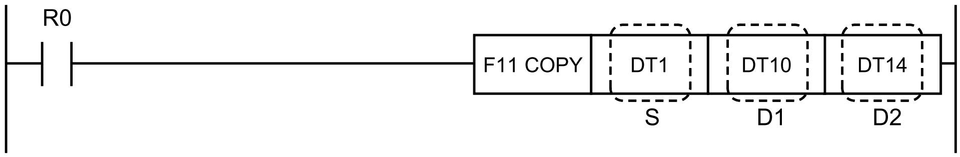
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

10.10 F11 COPY (16-bit Data Block Copy)

Copies the specified data to all areas in the range specified by the block.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the copy source data, or constant data
D1	Starting number of data copy destination area
D2	End number of data copy destination area

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D1		•	•	•	•	•	•	•								•	
D2		•	•	•	•	•	•	•								•	

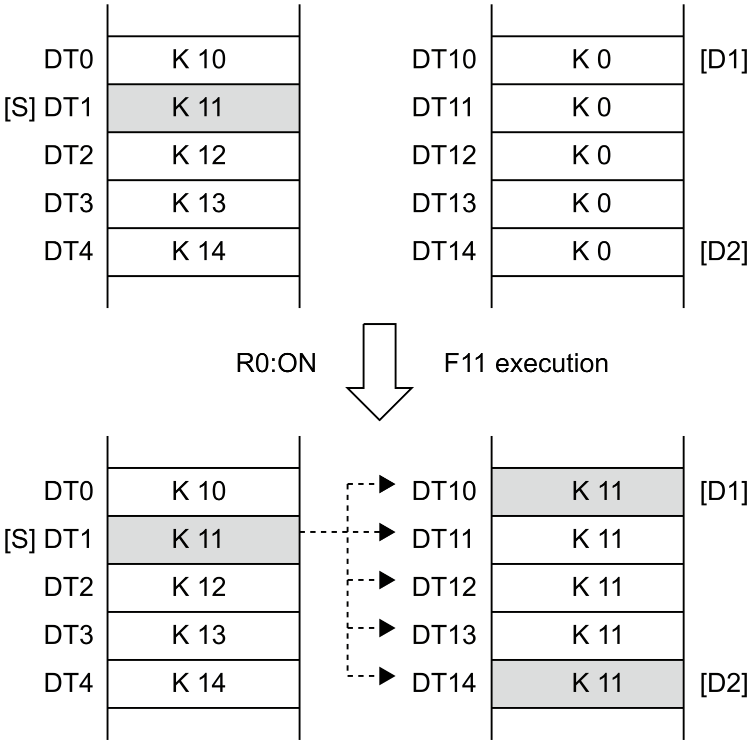
■ Outline of operation

16-bit data in the area specified by [S] is copied to all areas between [D1] and [D2].

■ Operation example

Operation of instruction format description program

The data from data register DT1 is copied to each data register from DT10 to DT14 when internal relay R0 turns ON.



■ Precautions for programming

- Specify the same type of memory area for both [D1] and [D2].
- The area of the lower address for the block being copied should be specified by [D1], and the higher address should be specified by [D2]. If specified as [D1] > [D2], an operation error will occur when the instruction is executed.
- When the same number is specified for [D1] and [D2], the 16-bit data is transferred to that number's area.

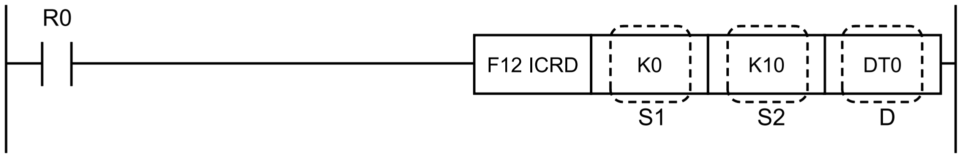
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the D1 address > D2 address

10.11 F12 ICRD (Data Read from F-ROM)

Reads the specified data from the F-ROM area.

■ Instruction format



■ Operands

Item	Setting
S1	Starting block number of the F-ROM area (Settable range: K0 to K15)
S2	Number of blocks to be read (Settable range: (K1 to K16)
D	Starting number of area storing read data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer device
												K	H	M	f		
S1												•					
S2												•					
D							•										

■ Outline of operation

Source

Transfers data starting from the block **S1** in F-ROM for the blocks specified in **S2**.

Destination

Transfers to the memory area starting from the address **D** in the data register.

Transfer units

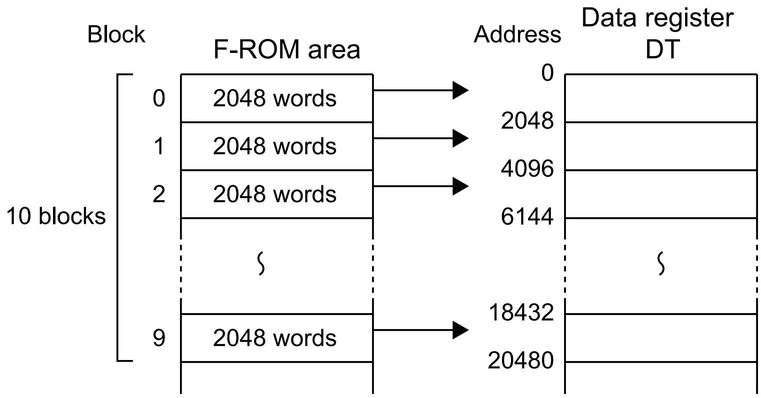
Data is transferred by the following units.

Data to be transferred per block: 2,048 words

■ Operation example

Operation of instruction format description program

When execution condition R0 turns ON, 10 blocks of F-ROM data from block 0 to block 9 are transferred to data registers DT0 to DT20479.



Precautions for programming

- Values that can be specified for [S1], [S2] and [D]

	Memory area
[S1]	K0 to K15
[S2]	K1 to K16
[D]	C32,T32,F32: DT0 to DT30720 C10, C14, C16: DT0 to DT12284

- Volume of data held in F-ROM

  Readable volume

  C32, T32, F32: 32,765 words

  C10, C14, C16: 12,285 words

- Because the initial data in the F-ROM is undefined, caution is required when reading the F-ROM before data has been written to the F-ROM.
- The default values in the F-ROM can be cleared to 0 when the entire program is deleted using the tool software.

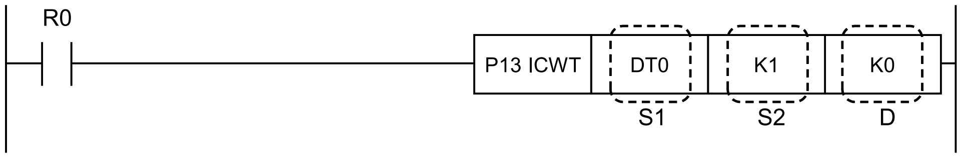
Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the address specified by [S1] does not exist in the F-ROM area.
	Turns ON when the area specified by [S2] exceeds the range of the F-ROM area.
	Turns ON when the area is exceeded when the blocks specified by [D] and subsequent parameters are transferred.

10.12 P13 PICWT (Write to F-ROM)

Transfers the specified data to an area in F-ROM.

■ Instruction format



■ Operands

Item	Setting
S1	Starting number of the area storing the write data
S2	Number of blocks to be written (Settable range: K1)
D	Starting number of the F-ROM area where data is to be written (Settable range: K0 to K15)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer device
												K	H	M	f		
S1							•										
S2												•					
D												•					

■ Outline of operation

Source

Transfers data stored in a data register (data starting from the address **S1**) for the blocks specified in**S2**.

Destination

Transfers to the memory area starting from the block **D** in the F-ROM.

Transfer units

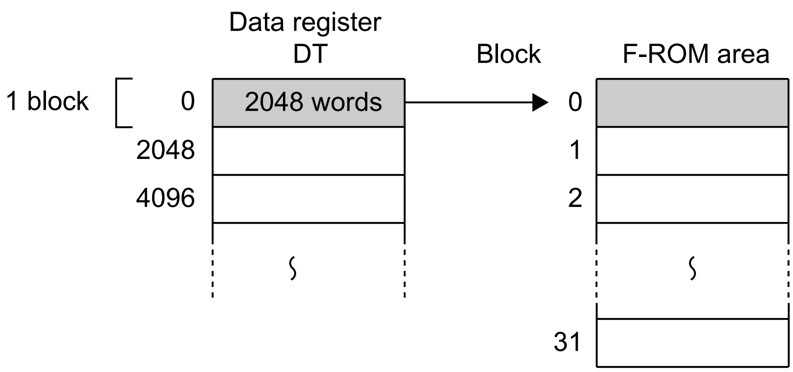
Data is transferred by the following units.

Data to be transferred per block: 2,048 words

■ Operation example

Operation of instruction format description program

When execution condition R0 turns ON, one block (2,048 words) of data from data register DT0 is transferred to block 0 in the F-ROM area.



Precautions for programming

- Values that can be specified for [S1], [S2] and [D]

	Memory area
[S1]	C32, T32, F32: DT0 to DT30720 C10, C14, C16: DT0 to DT12284
[S2]	K1
[D]	K0 to K15

- Volume of data that can be held in F-ROM

  Storable volume

  C32, T32, F32: 32,765 words

  C10, C14, C16: 12285 words

- Only one block can be written at a time.
- It takes up to approximately 100 ms to execute the instruction. When writing multiple blocks, write them across multiple scans.
- The number of times data can be written to F-ROM is 10,000 times or less.
- This instruction is a differential execution type instruction (P13) used to prevent multiple writes to F-ROM caused by programming errors.
- Be careful to avoid repeated writing to F-ROM when creating the program.
- Do not use this instruction in interrupt programs.

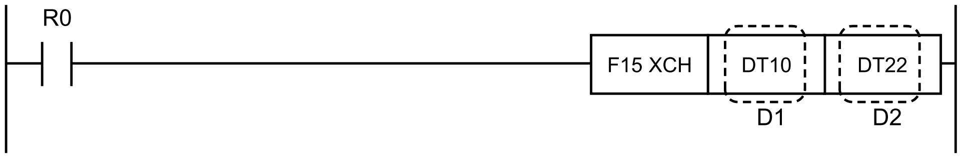
Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded when transferring the blocks to the area specified from [S1].
	Turns ON when one block is not specified by [S2].
	Turns ON when the address specified by [D] is not in the F-ROM area.

10.13 F15 XCH (16-bit Data Exchange)

Exchanges 16-bit data of two areas.

■ Instruction format



■ Operands

Items	Settings
D1	Area that stores the 16-bit data to exchange with D2
D2	Area that stores the 16-bit data to exchange with D1

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D1		•	•	•	•	•	•	•	•							•	
D2		•	•	•	•	•	•	•	•							•	

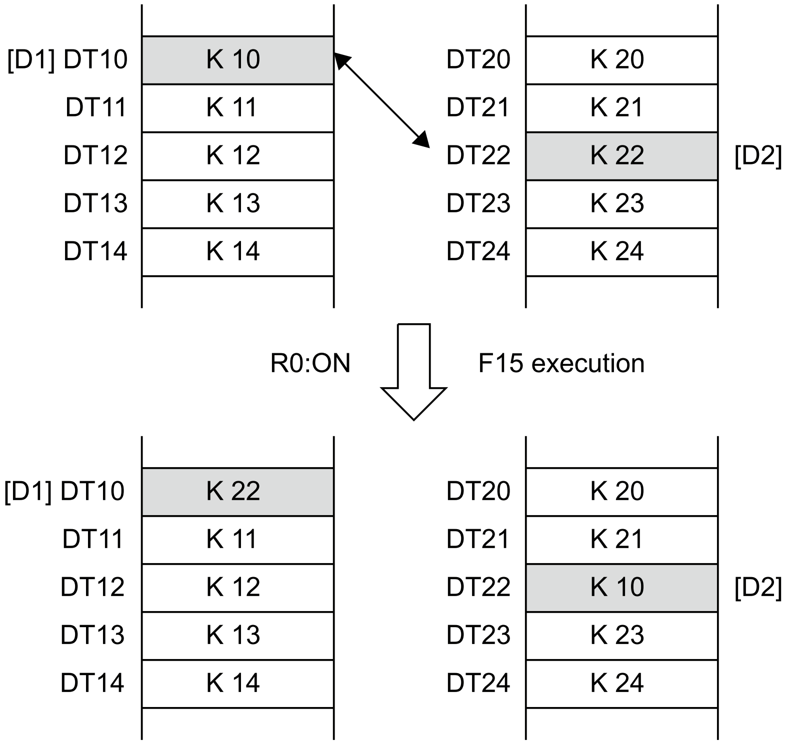
■ Outline of operation

Exchanges the data in the area specified by [D1] with that in the area specified by [D2].

■ Operation example

Operation of instruction format description program

The contents of data register DT10 and data register DT22 are exchanged when internal relay R0 turns ON.



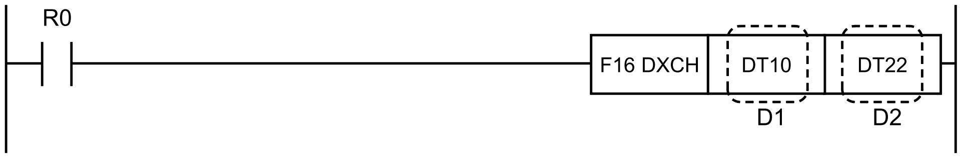
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

10.14 F16 DXCH (32-bit Data Exchange)

Exchanges the 32-bit data of two areas.

■ Instruction format



■ Operands

Items	Settings
D1	Area storing the 32-bit data to be exchanged with D2
D2	Area storing the 32-bit data to be exchanged with D1

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D1		•	•	•	•	•	•	•	•							•	
D2		•	•	•	•	•	•	•	•							•	

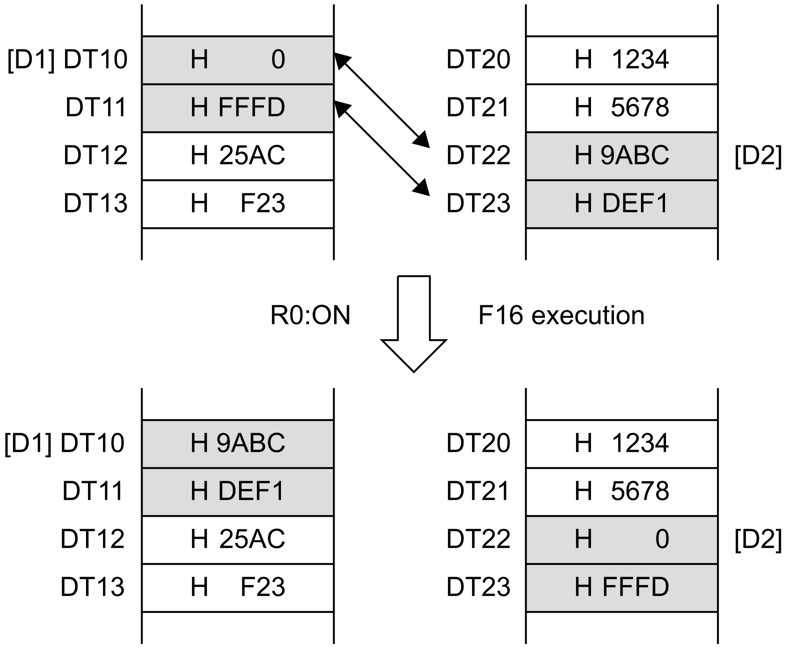
■ Outline of operation

The first two words of the content (32-bit) at the start of the area specified by [D1] are exchanged with the first two words of the content (32-bit) at the start of the area specified by [D2].

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the data in data registers DT10 and DT11 is exchanged with the data in data registers DT22 and DT23.



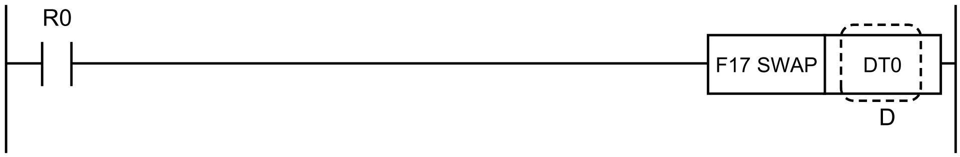
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

10.15 F17 SWAP (Higher/Lower Byte Exchange)

Exchanges higher (8-bit) and lower (8-bit) order bytes in 16-bit data.

■ Instruction format



■ Operands

Items	Settings
D	Area storing 16-bit data for higher 8-bit and lower 8-bit exchange

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	

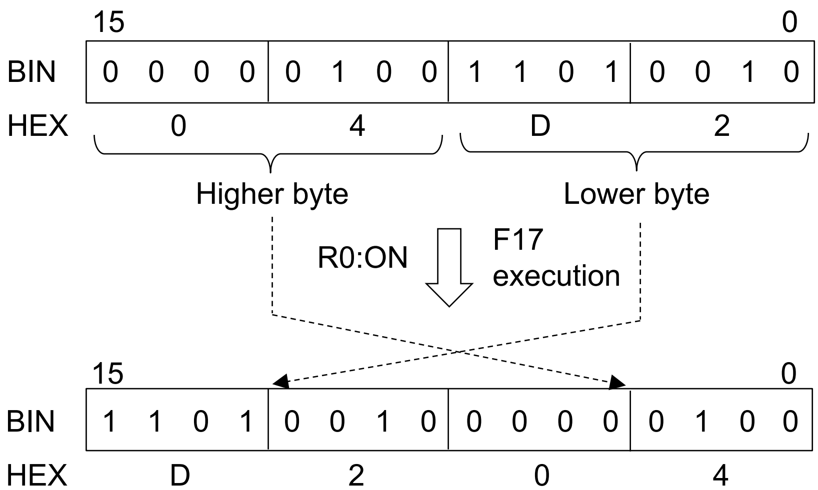
■ Outline of operation

Exchanges the higher and lower order bytes of the 16-bit data stored in the area specified by [D].

■ Operation example

Operation of instruction format description program

The higher and lower bytes stored in data register DT0 are exchanged when internal relay R0 turns ON.



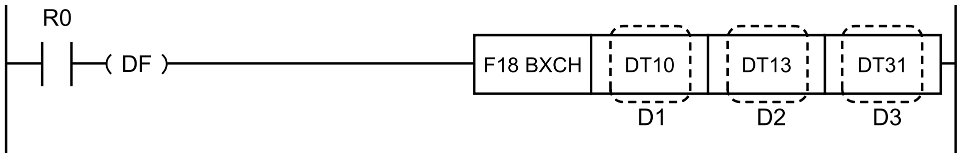
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

10.16 F18 BXCH (Block Exchange)

Exchanges data in blocks.

■ Instruction format



■ Operands

Items	Settings
D1	Starting address for exchange block 1
D2	Ending address for exchange block 1
D3	Starting address for exchange block 2

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D1		•	•	•	•	•	•	•	•							•	
D2		•	•	•	•	•	•	•	•							•	
D3		•	•	•	•	•	•	•	•							•	

■ Outline of operation

Exchanges the data from the area specified in [D1] to the area specified in [D2] with the data in the area starting at [D3].

■ Precautions for programming

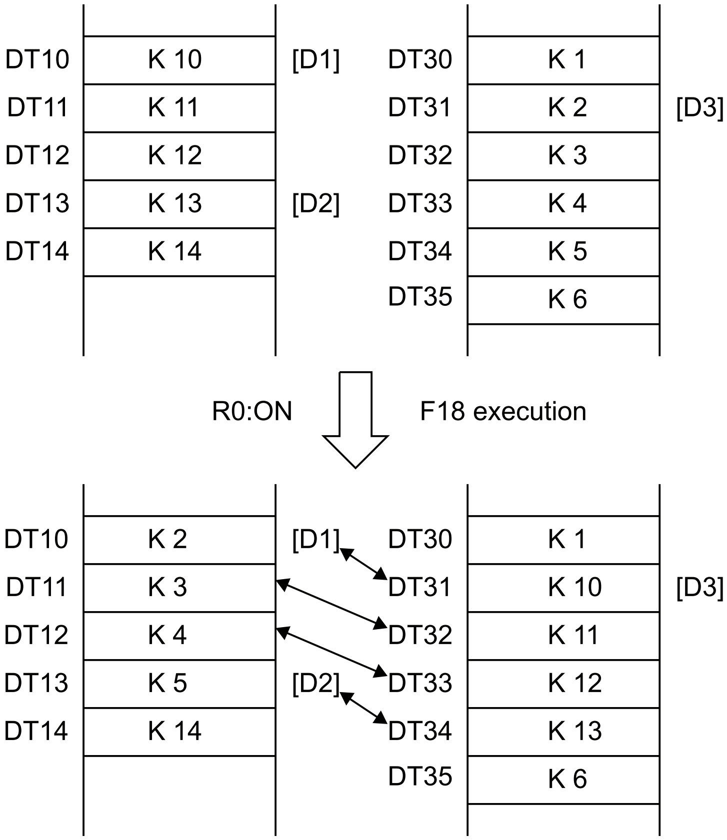
- Specify the same type of memory address for [D1] and [D2].
- Specify the number of the lower address with [D1], and the number of the higher address with [D2].

  If specified as [D1] > [D2], an operation error will occur when the instruction is executed.
- If the blocks to be exchanged overlap, they cannot be exchanged correctly. However, an error will not occur.

■ Operation example

Operation of instruction format description program

When the execution condition R0 is ON, data is exchanged between data registers DT10 to DT13 and DT31 to DT34.



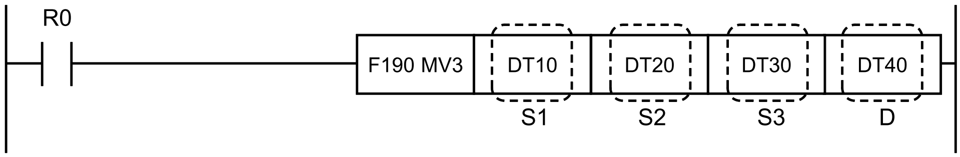
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when [D1] > [D2]
	Turns ON when area is exceeded when exchanging blocks specified in [D3] or higher

10.17 F190 MV3 (Three 16-bit Data Transfer to Single Area)

Three 16-bit data items are batch-transferred from the specified area number.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the hexadecimal data or constant data
S2	Area storing the hexadecimal data or constant data
S3	Area storing the hexadecimal data or constant data
D	Starting address of the data transfer destination area (three words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•			•	•			•	
S2	•	•	•	•	•	•	•	•	•			•	•			•	
S3	•	•	•	•	•	•	•	•	•			•	•			•	
D		•	•	•	•	•	•	•	•							•	

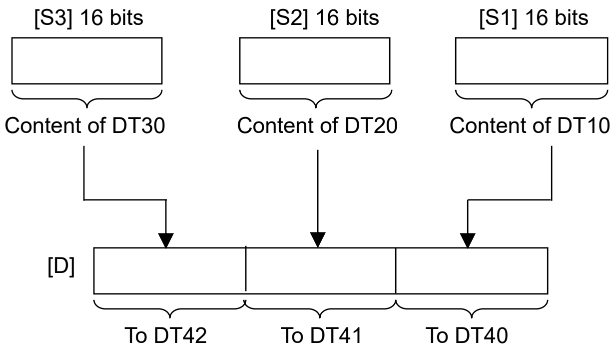
■ Outline of operation

The three types of 16-bit data in the memory areas specified by [S1], [S2], and [S3] are batch-transferred to the memory area (three words) specified by [D].

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of data register DT10 is transferred to DT40, the content of DT20 is transferred to DT41, and the content of DT30 is transferred to DT42, in a batch.



■ Related instructions

Use the F87 MV2 instruction when batch-transferring two types of 16-bit data.

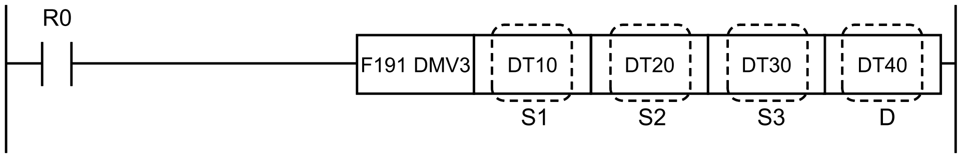
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

10.18 F191 DMV3 (Three 32-bit Data Transfer to Single Area)

Three 32-bit data items are batch-transferred from the specified area number.

■ Instruction format



■ Operands

Item	Setting
S1	Area storing 32-bit data, or constant data
S2	Area storing 32-bit data, or constant data
S3	Area storing 32-bit data, or constant data
D	Starting address of the data transfer destination area (six words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	SWR	SDT	Constant				Index modifier	Integer device
													K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•			•	•			•	
S2	•	•	•	•	•	•	•	•	•	•			•	•			•	
S3	•	•	•	•	•	•	•	•	•	•			•	•			•	
D		•	•	•	•	•	•	•	•	•							•	

■ Outline of operation

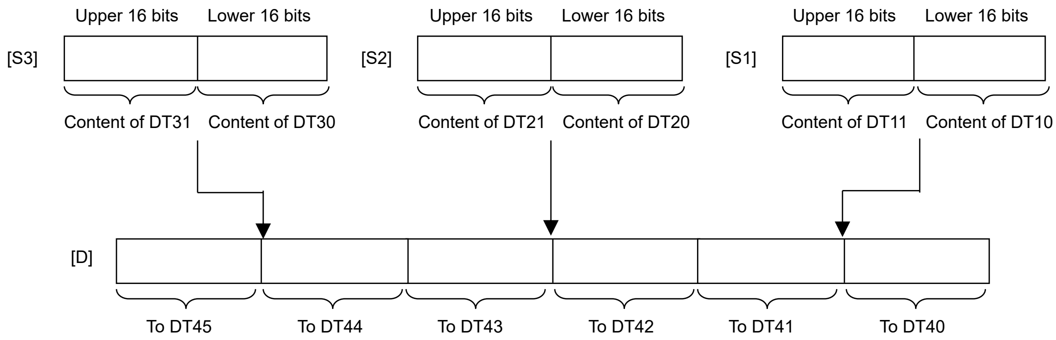
The three types of 32-bit data in the memory areas specified by [S1], [S2], and [S3] are batch-transferred to the memory area (six words) specified by [D].

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the combined 32-bit content of data registers DT10 and DT11, data registers DT20 and DT21, and data registers DT30 and DT31 is batch-transferred to the 6-word area starting from data register DT40.

- The specification of [S1], [S2], and [S3] specifies the lower 16-bit memory area.
- The specification of [D] specifies the start of the 6 word memory area.



■ Related instructions

Use the F8 DMV2 instruction when batch-transferring two types of 32-bit data.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

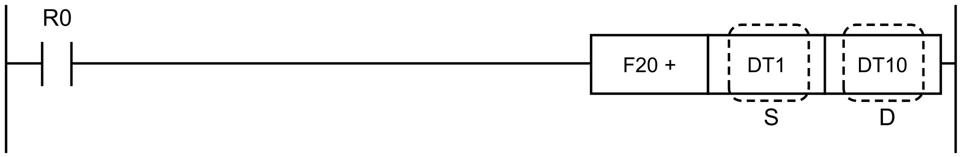
(MEMO)

11 Binary Arithmetic Instructions

11.1 F20 + (16-bit Data Addition [D+S=D])

16-bit data is added.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the 16-bit data to be added, or constant data
D	Area storing the data (16-bit) to be added

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

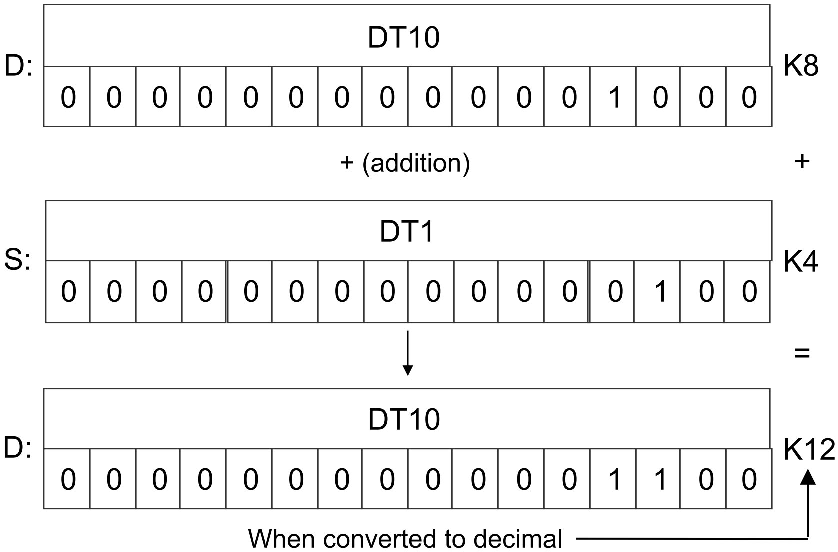
■ Outline of operation

- The 16-bit data specified in [S] is added to the 16-bit data representing the decimal specified in [D].
  (D) + (S) → (D)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of data register DT10 is added to the content of data register DT1. When the decimal number 4 is in DT1, and 8 is in DT10, it will be as follows.



■ Precautions for programming

- With arithmetic operation instructions, in the event that the operation result falls beyond the range of values that can be handled, either an overflow or underflow occurs.
- Under normal circumstances, do not allow an overflow or underflow to occur.
- If an overflow or underflow occurs, use the 32-bit operation instruction.
- Use the F89 EXT sign extension instruction to convert the 16-bit data into 32-bit data.
- If an overflow or underflow occurs, the CY flag (special internal relay R9009) turns ON.

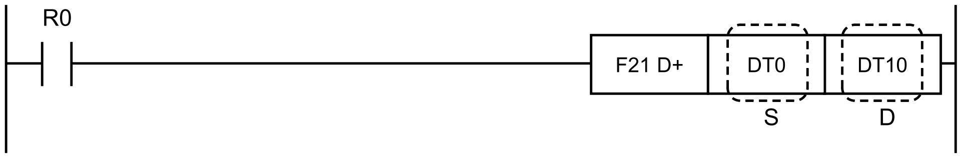
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows/underflows

11.2 F21 D+ (32-bit Data Addition [D+S=D])

32-bit data is added.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the 32-bit data to be added, or constant data
D	Area storing the data (32-bit) to be added

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

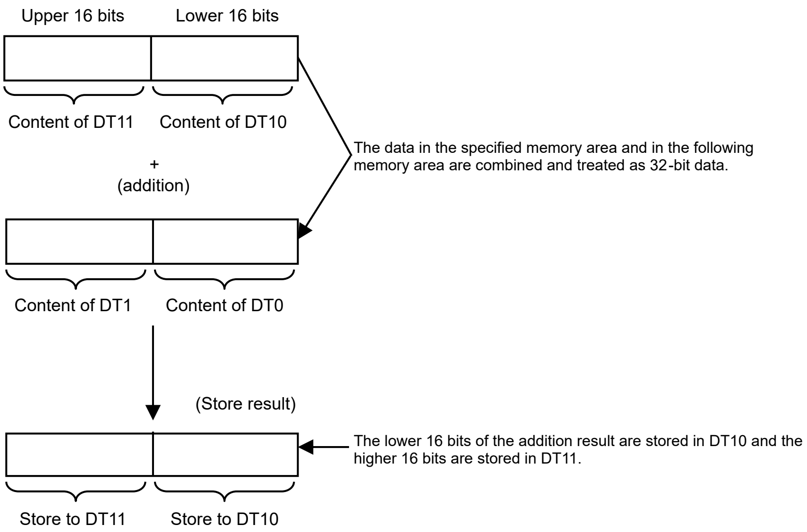
- The 32-bit data specified in [S] is added to the 32-bit data representing the decimal specified in [D].

  (D+1, D) + (S+1, S) → (D+1, D)

■ Operation example

Operation of instruction format description program

When the internal relay R0 is ON, the content (32-bit) of data registers DT10 to DT11 is added to the content (32-bit) of data registers DT0 to DT1.



■ Precautions for programming

- With arithmetic operation instructions, in the event that the operation result falls beyond the range of values that can be handled, either an overflow or underflow occurs.
- Ensure that overflows and underflows do not occur in normal circumstances.
- If an overflow or underflow occurs, the CY flag (special internal relay R9009) turns ON.

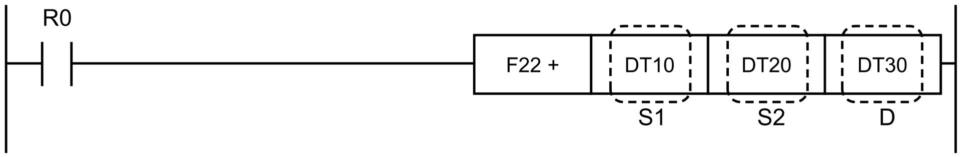
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows/underflows

11.3 F22 + (16-bit Data Addition [S1+S2=D])

This is an instruction that adds 16-bit data.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the 16-bit data to be added, or constant data
S2	Area storing the 16-bit data to be added, or constant data
D	Area storing the addition results

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

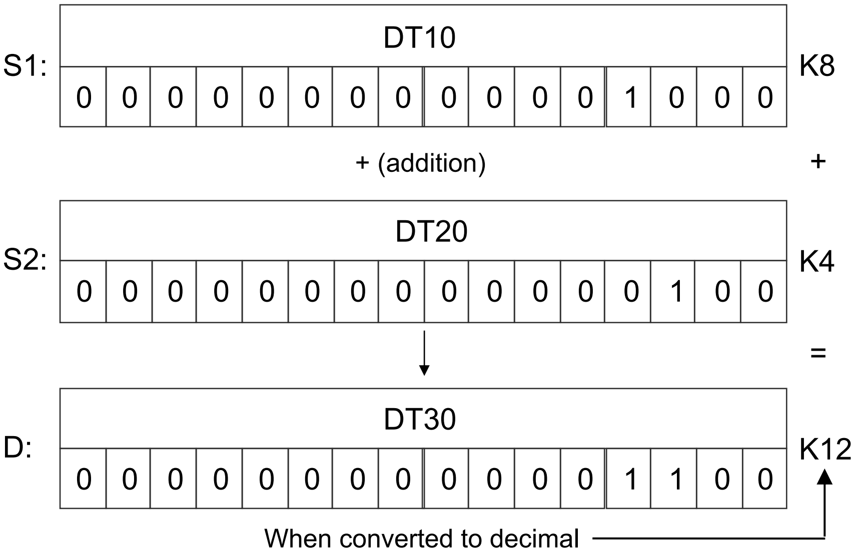
■ Outline of operation

- The 16-bit data expressing a decimal number specified by [S1] and [S2] is added, and the result is stored in [D].
  (S1) + (S2) → (D)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the contents of data register DT10 and data register DT20 are added together, and the result is stored in data register DT30. If DT10 contains decimal 8 and DT20 contains decimal 4, the result is as follows.



■ Precautions for programming

- With arithmetic operation instructions, in the event that the operation result falls beyond the range of values that can be handled, either an overflow or underflow occurs.
- Under normal circumstances, do not allow an overflow or underflow to occur.
- If an overflow or underflow occurs, use the 32-bit operation instruction.
- Use the F89 EXT sign extension instruction to convert the 16-bit data into 32-bit data.
- If an overflow or underflow occurs, the CY flag (special internal relay R9009) turns ON.

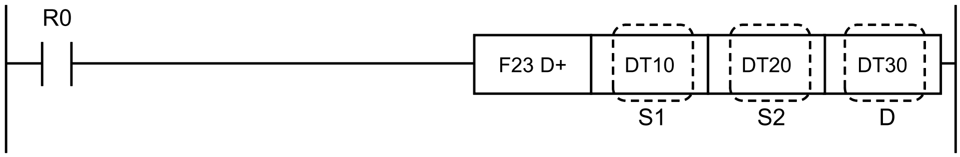
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows/underflows

11.4 F23 D+ (32-bit Data Addition [S1+S2=D])

This is an instruction that adds 32-bit data.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the 32-bit data to be added, or constant data
S2	Area storing the 32-bit data to be added, or constant data
D	Area storing the addition results

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

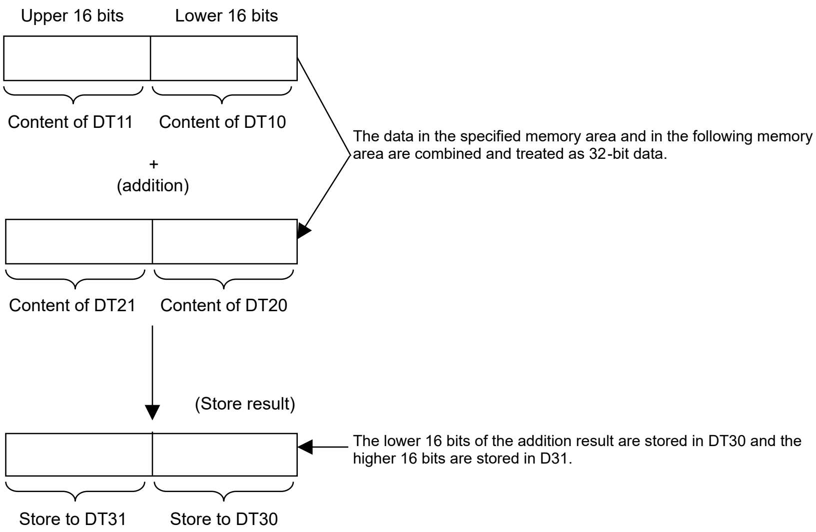
■ Outline of operation

- The 32-bit data expressing a decimal number specified by [S1] and [S2] is added, and the result is stored in [D].
  (S1+1, S1) + (S2+1, S2) → (D+1, D)
- The memory area is specified by the memory area number of the lower 16-bit portion.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the contents of data registers DT10 and DT11 are added to the contents of data registers DT20 and DT21, and the result is stored in data registers DT30 and DT31.



■ Precautions for programming

- With arithmetic operation instructions, in the event that the operation result falls beyond the range of values that can be handled, either an overflow or underflow occurs.
- Ensure that overflows and underflows do not occur in normal circumstances.
- If an overflow or underflow occurs, the CY flag (special internal relay R9009) turns ON.

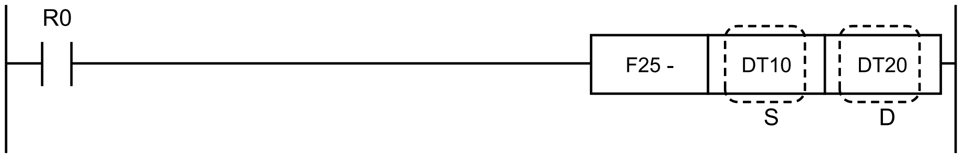
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows/underflows

11.5 F25 - (16-bit Data Subtraction [D-S=D])

16-bit data is subtracted.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the subtrahend (16-bit data), or constant data
D	Area storing the subtrahend from (16-bit data)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The 16-bit data specified by [S] is subtracted from the 16-bit decimal data specified by [D].
  (D) - (S) -> (D)

■ Operation example

Operation of instruction format description program

Subtracts the contents of data register DT10 from the contents of data register DT20 when internal relay R0 turns ON.

Specific Example 1) When the decimal number 16 is in DT20 and the decimal number 4 is in DT10

D:

DT20															
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

K16

- (subtraction) -

S:

DT10															
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

K4

=

D:

DT20															
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

K12

When converted to decimal

Specific Example 2) When the decimal number 3 is in DT20 and the decimal number 5 is in DT10

D:

DT20															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

K3

- (subtraction) -

S:

DT10															
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

K5

=

D:

DT20															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

K-2

When converted to decimal

■ Precautions for programming

- With arithmetic operation instructions, in the event that the operation result falls beyond the range of values that can be handled, either an overflow or underflow occurs.
- Under normal circumstances, do not allow an overflow or underflow to occur.
- If an overflow or underflow occurs, use the 32-bit operation instruction.
- Use the F89 EXT sign extension instruction to convert the 16-bit data into 32-bit data.
- If an overflow or underflow occurs, the CY flag (special internal relay R9009) turns ON.

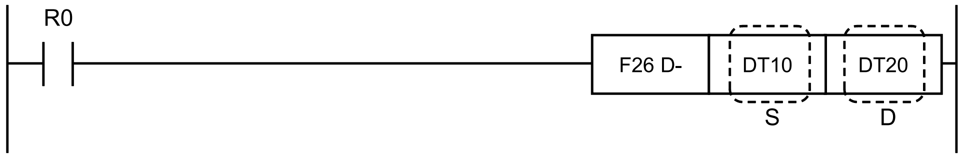
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows/underflows

11.6 F26 D-(32-bit Data Subtraction [D-S=D])

Subtracts 32-bit data.

■ Instruction format



■ Operands

Items	Settings
S	Area that stores subtrahends (32-bit data), or constant data
D	Area storing the number to be subtracted (32-bit data)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

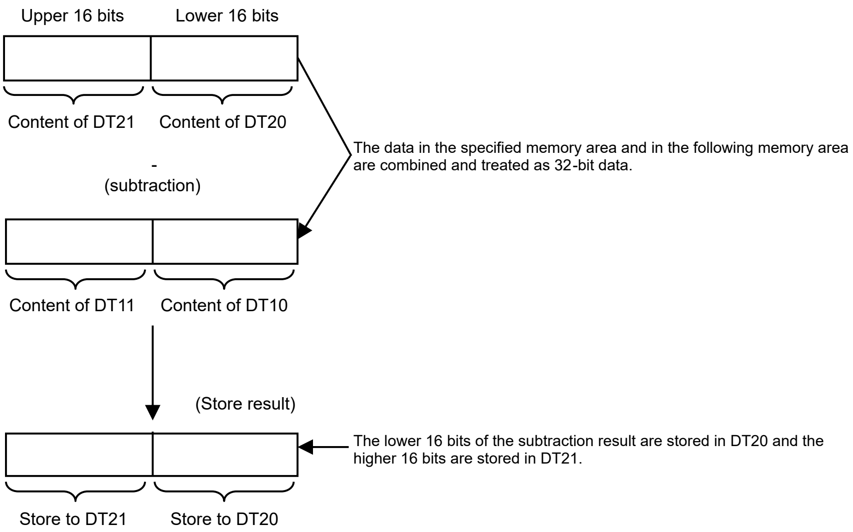
■ Outline of operation

- The 32-bit data specified by [S] is subtracted from the 32-bit data expressing a decimal number specified by [D].
  (D+1, D) – (S+1, S) → (D+1, D)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of data registers DT10 and DT11 (32 bits) is subtracted from the content of data registers DT20 and DT21(32 bits).



■ Precautions for programming

- With arithmetic operation instructions, in the event that the operation result falls beyond the range of values that can be handled, either an overflow or underflow occurs.
- Ensure that overflows and underflows do not occur in normal circumstances.
- If an overflow or underflow occurs, the CY flag (special internal relay R9009) turns ON.

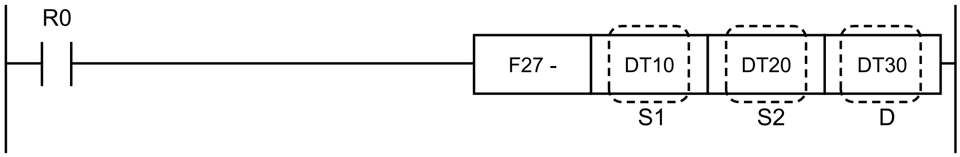
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows/underflows

11.7 F27 - (16-bit Data Subtraction [S1-S2=D])

16-bit data is subtracted.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the number to be subtracted (16-bit data), or constant data
S2	Area storing the subtrahend (16-bit data), or constant data
D	Area that stores operation results

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The 16-bit data specified in [S2] is subtracted from the 16-bit data representing the decimal of the memory area specified in [S1], and the result is stored in [D].
  (S1) - (S2) → (D)

■ Operation example

Operation of instruction format description program

When the internal relay R0 is ON, the content of data register DT20 is subtracted from the content of data register D10, and the operation result is stored in data register DT30.

Example 1) If the decimal 16 is in DT10, and 4 is in DT20

DT10																
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	K16

- (subtraction) -

DT20																
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	K4

↓ =

DT30																
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	K12

When converted to decimal

Example 2) If the decimal 3 is in DT10, and 5 is in DT20

DT10																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	K3

- (subtraction) -

DT20																
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	K5

↓ =

DT30																
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	K-2

When converted to decimal

■ Precautions for programming

- With arithmetic operation instructions, in the event that the operation result falls beyond the range of values that can be handled, either an overflow or underflow occurs.
- Under normal circumstances, do not allow an overflow or underflow to occur.
- If an overflow or underflow occurs, use the 32-bit operation instruction.
- Use the F89 EXT sign extension instruction to convert the 16-bit data into 32-bit data.
- If an overflow or underflow occurs, the CY flag (special internal relay R9009) turns ON.

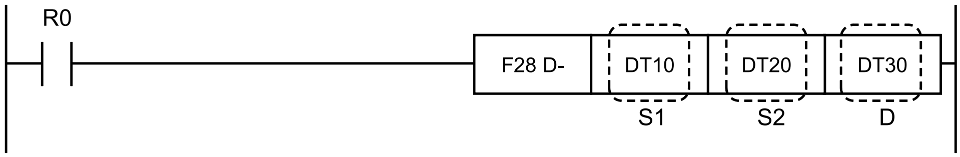
Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows/underflows

11.8 F28 D- (32-bit Data Subtraction [S1-S2=D])

Subtracts 32-bit data.

■ Instruction format



■ Operands

Items	Settings
S1	Area that stores minuends (32-bit data), or constant data
S2	Area that stores subtrahends (32-bit data), or constant data
D	Area that stores operation results

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

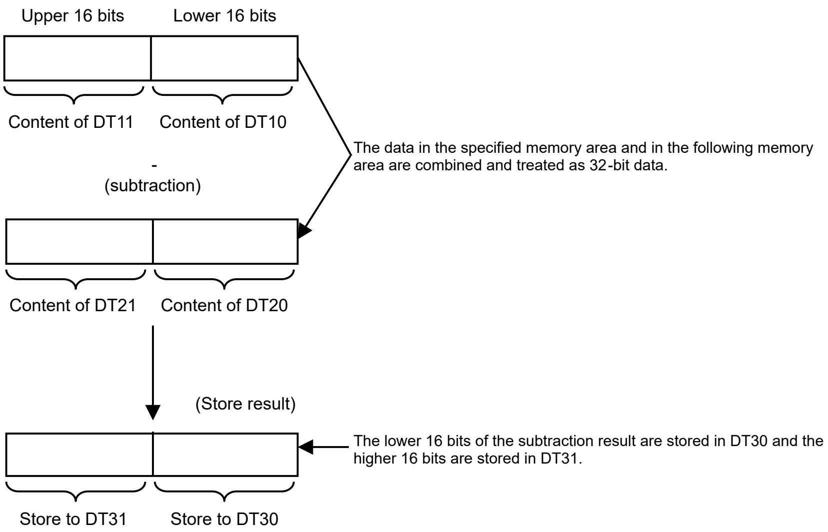
■ Outline of operation

- 32-bit data specified in [S2] is subtracted from the 32-bit data, representing a decimal, of the memory area specified in [S1], and the result is stored in [D].
  (S1+1, S1) - (S2+1, S2) → (D+1, D)
- The memory area is specified by the memory area number of the lower 16-bit portion.

■ Operation example

Operation of instruction format description program

When the internal relay R0 is ON, the content of data registers DT20 to DT21 is subtracted from the content of DT10 to DT11, and the operation result is stored in DT30 to DT31.



■ Precautions for programming

- With arithmetic operation instructions, in the event that the operation result falls beyond the range of values that can be handled, either an overflow or underflow occurs.
- Ensure that overflows and underflows do not occur in normal circumstances.
- If an overflow or underflow occurs, the CY flag (special internal relay R9009) turns ON.

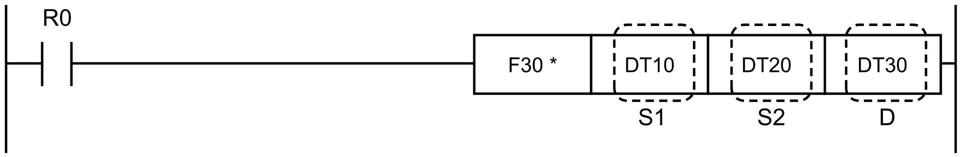
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows/underflows

11.9 F30 * (16-bit Data Multiplication [S1*S2=D+1, D])

Multiplies hexadecimal data.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the hexadecimal data or constant data
S2	Area storing the hexadecimal data or constant data
D	Area storing the multiplication results (32-bit data)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

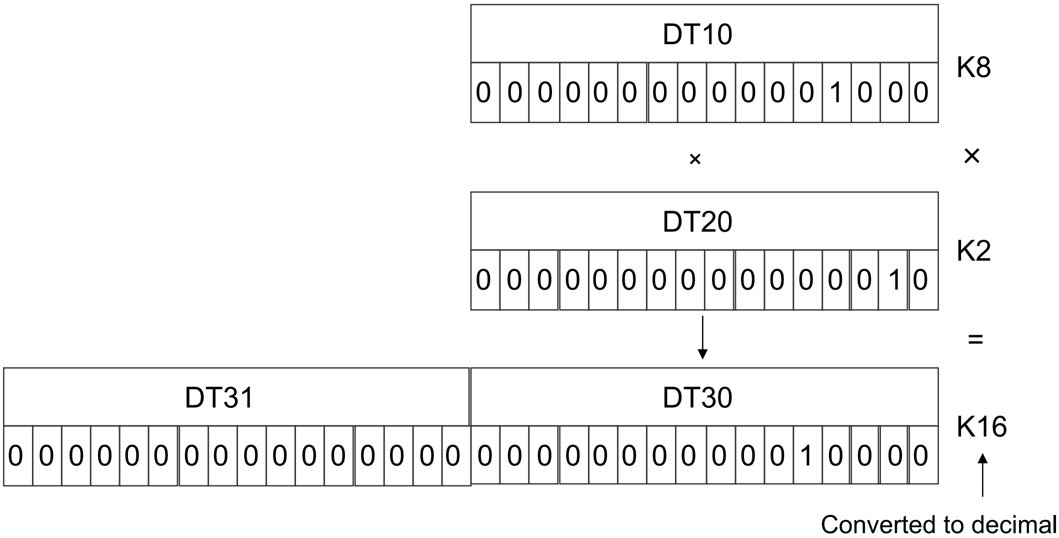
■ Outline of operation

- Multiplies the hexadecimal data expressed in decimal form that is specified by [S1] with the hexadecimal data specified by [S2], and stores the result in the area specified by [D].
  (S1) × (S2) → (D+1, D)
- The calculation result is stored using 32-bit data (K constant).
- Storage destination [D] is specified by the number of the memory area with the lower order 16 bits.

■ Operation example

Operation of instruction format description program

The contents of data registers DT10 and DT20 are multiplied and stored in data registers DT30 and DT31 when internal relay R0 turns ON. When 8 is in the decimal number in DT10 (K constant) and 2 is in the decimal number 4 in DT20.



Of the 32-bit data multiplication results, the lower order 16 bits are stored in the specified memory area (DT30) and the higher order 16 bits is stored in the next area after the specified area (DT31).

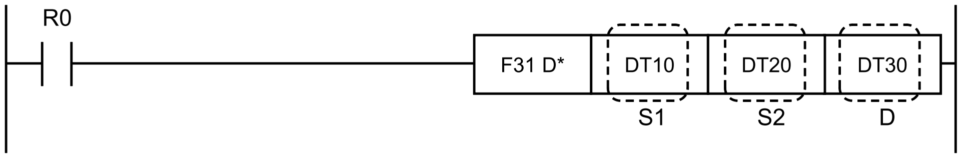
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"

11.10 F31 D* (32-bit Data Multiplication [S1*S2=D+3, D+2, D+1, D]

Multiplies 32-bit data items.

■ Instruction format



■ Operands

Items	Settings
S1	Multiplicand data: Area storing 32-bit data, or constant data
S2	Multiplier data: Area storing 32-bit data, or constant data
D	Storage destination: Area storing multiplication result (64-bit data)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●								●	

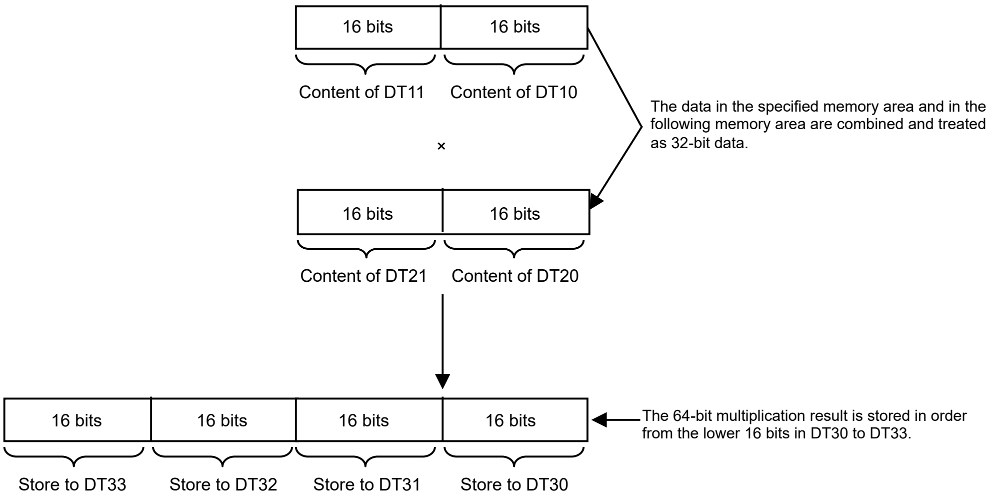
■ Outline of operation

- Multiplies the 32-bit data representing decimal data specified by [S1] and the 32-bit data specified by [S2], and stores the result in the area specified by [D].

  (S1+1, S1) × (S2+1, S2) → (D+3, D+2, D+1, D)
- The calculation result is stored in the 64-bit area.
- The memory area is specified by the number of the lowest 16-bit memory area.

■ Operation example

Operation of instruction format description program



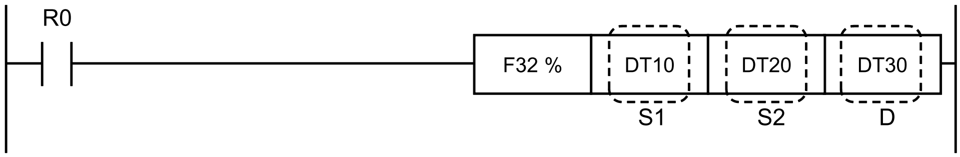
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"

11.11 F32 % (16-bit Data Subtraction [S1/S2=D])

Divides 16-bit data.

■ Instruction format



■ Operands

Items	Settings
S1	Dividend data: Area storing 16-bit data, or constant data
S2	Divisor data: Area storing 16-bit data, or constant data
D	Storage destination: Area storing the division result (quotient) (remainder stored as 16-bit data in DT90015)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The 16-bit data expressing a decimal specified by [S1] is divided by the 16-bit data specified by [S2]. The quotient is stored in [D], and the remainder is stored in special data register DT90015.

  (S1) ÷ (S2) → Quotient (D) Remainder (DT90015)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, data register DT10 is divided by data register DT20, and the quotient is stored in DT30 and the remainder in DT90015. If the content in DT10 is decimal number (K constant) 15 and the content in DT20 is 4, the result is as follows.

DT10															
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

K15

÷ ÷

DT20															
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

K4

=

DT30															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Quotient K3

DT90015															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Remainder K3

When converted to decimal

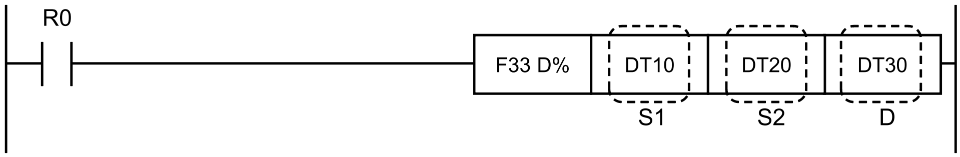
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification. Turns ON when [S2] is"0"
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when the negative maximum value is divided by"–1"

11.12 F33 D% (32-bit Data Subtraction [S1/S2=D+1, D])

Divides 32-bit data.

■ Instruction format



■ Operands

Items	Settings
S1	Dividend data: Area storing 32-bit data, or constant data
S2	Divisor data: Area storing 32-bit data, or constant data
D	Storage destination: Area storing the division result (quotient) (remainder stored as 32-bit data in DT90015 and DT90016)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

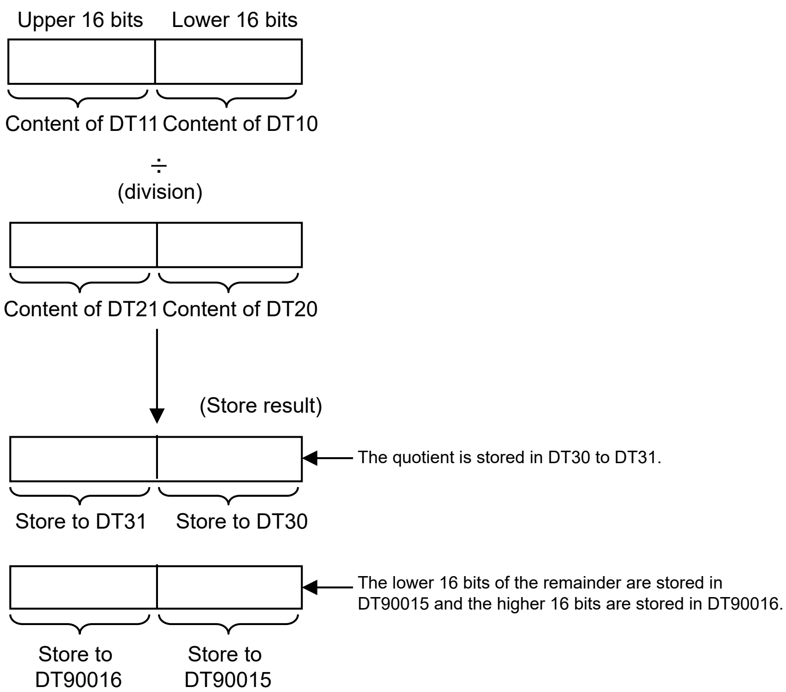
■ Outline of operation

- The 32-bit data expressing a decimal specified by [S1] is divided by the 32-bit data specified by [S2]. The quotient is stored in [D], and the remainder is stored in special data registers DT90015 and DT90016.

  (S1 + 1, S1) ÷ (S2 + 1, S2) → Quotient (D+1, D) Remainder (DT90016, DT90015)
- Memory area is specified by the memory area number of the lower order hexadecimal part.

■ Operation example

Operation of instruction format description program



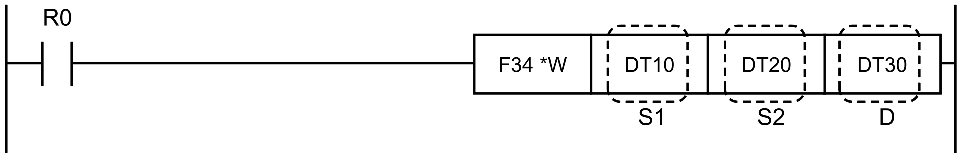
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when [S2] is"0"
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when the negative maximum value is divided by"–1"

11.13 F34 *W (16-bit Data Multiplication [S1*S2=D])

Multiplies 16-bit data and stores the result in a 16-bit, one-word area.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the hexadecimal data or constant data
S2	Area storing the hexadecimal data or constant data
D	Area storing multiplication result (16-bit data)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

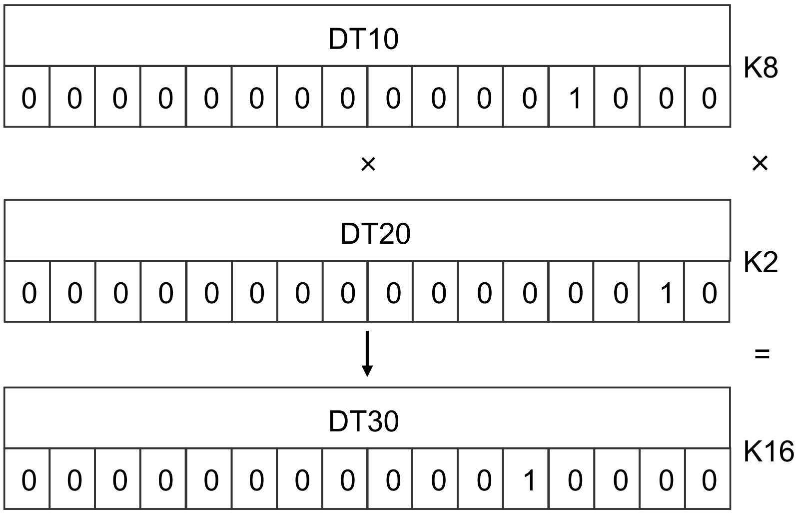
■ Outline of operation

- The 16-bit data specified by [S1] and the 16-bit data specified by [S2] are multiplied, and the result is stored in the area specified by [D].
  (S1) x (S2) → (D)
- The operation result is stored as one word of 16-bit data.

■ Operation example

Operation of instruction format description program

When the DT10 content is decimal 8



■ Precautions for programming

Keep the operation result [D] within the range of K–32768 to K32767.

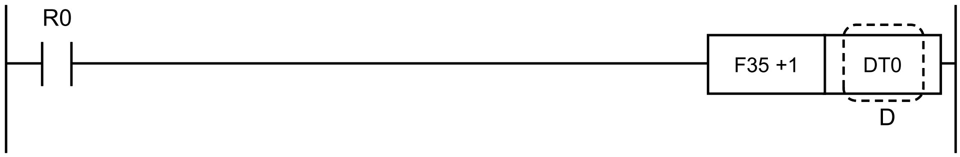
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the operation result exceeds 16 bits
R900B (=)	Turns ON when the calculation result is"0"

11.14 F35 +1 (16-bit Data Increment)

Adds 1 to 16-bit data.

■ Instruction format



■ Operands

Items	Settings
D	Area to which 1 is to be added

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		●	●	●	●	●	●	●	●							●	

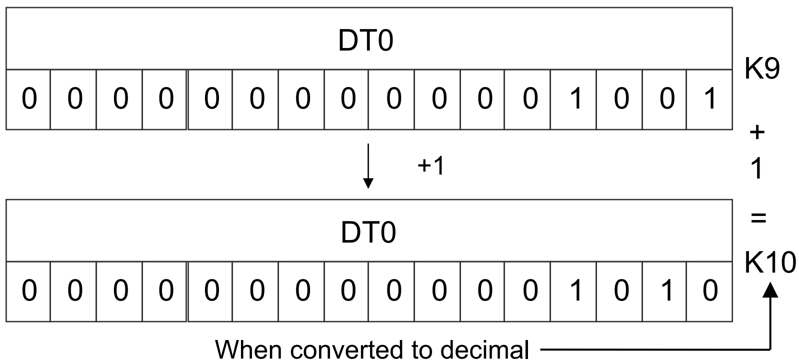
■ Outline of operation

- 1 is added to the 16-bit data that expresses the decimal number specified by [D] and the result is stored in [D].
  (D) + 1 → (D)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, 1 is added to the contents of data register DT0.



■ Precautions for programming

- With arithmetic operation instructions, in the event that the operation result falls beyond the range of values that can be handled, an overflow occurs.

- Ensure that overflows do not occur in normal circumstances.
- If an overflow occurs, use a 32-bit operation instruction.
- If an overflow occurs, the CY flag (special internal relay R9009) turns ON.

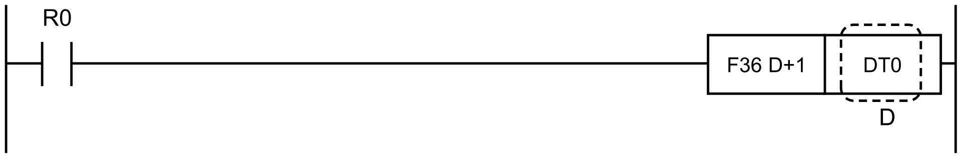
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows

11.15 F36 D+1 (32-bit Data Increment)

Adds 1 to 32-bit data.

■ Instruction format



■ Operands

Items	Settings
D	The area (32-bit) that +1 is added to

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	

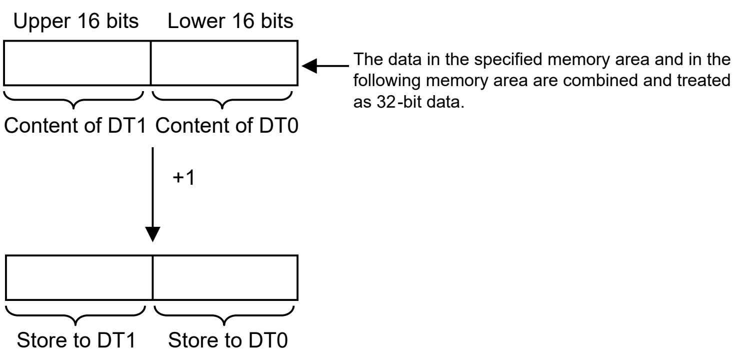
■ Outline of operation

- Adds +1 to the 32-bit data, representing a decimal, specified in [D] and stores it in the 2-word memory area starting at [D].
  (D+1, D) + 1 → (D+1, D)

■ Operation example

Operation of instruction format description program

When the internal relay R0 is ON, adds +1 to the content of the combined 32 bits of data registers DT0 and DT1.



■ Precautions for programming

- With arithmetic operation instructions, in the event that the operation result falls beyond the range of values that can be handled, an overflow occurs.
- Ensure that overflows do not occur in normal circumstances.
- If an overflow occurs, the CY flag (special internal relay R9009) turns ON.

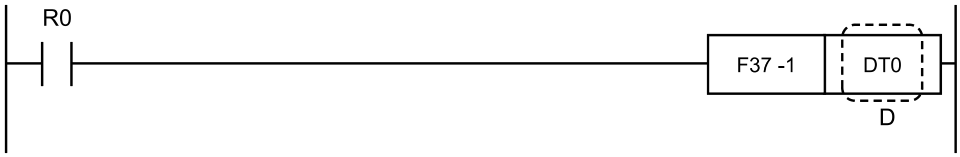
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows

11.16 F37 -1 (16-bit Data Decrement)

Subtracts 1 from 16-bit data.

■ Instruction format



■ Operands

Items	Settings
D	Area to be decreased by 1

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	

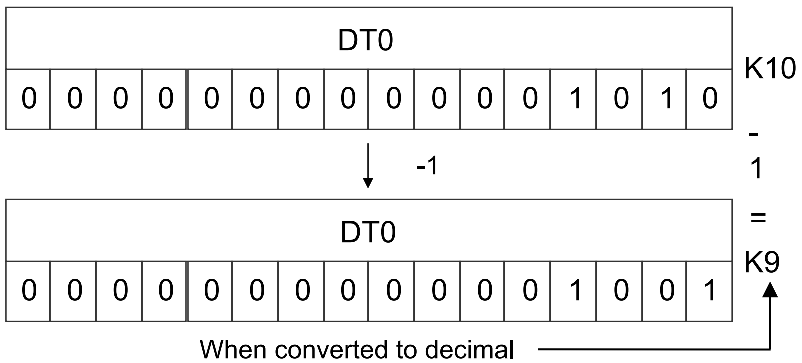
■ Outline of operation

- The 16-bit data specified by [D] and expressed in base 10 is decreased by 1 and stored in [D].
  (D) - 1 → (D)

■ Operation example

Operation of instruction format description program

When internal relay R0 is ON, the content of data register DT0 is decreased by 1.



■ Precautions for programming

- If the result of an arithmetic operation instruction exceeds the numerical range that can be handled, an underflow will result.

- Under normal circumstances, do not allow an underflow to occur.
- If an underflow occurs, use the 32-bit operation instruction.
- If an underflow occurs, the CY flag (special internal relay R9009) will turn ON.

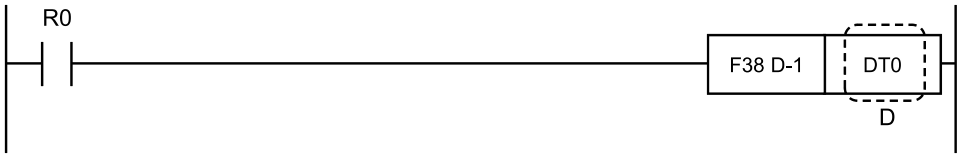
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when the calculation result underflows

11.17 F38 D-1 (32-bit Data Decrement)

Subtracts 1 from 32-bit data.

■ Instruction format



■ Operands

Items	Settings
D	Area (32-bit) from which 1 is subtracted

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		●	●	●	●	●	●	●	●							●	

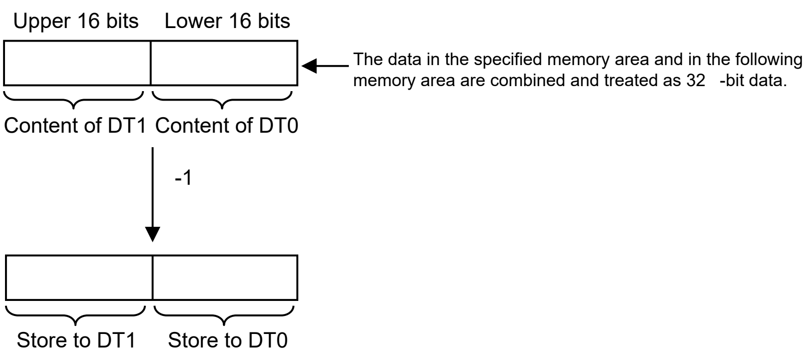
■ Outline of operation

- 1 is subtracted from the 32-bit data that expresses the decimal number specified by [D] and the result is stored in the 2-word memory area starting at [D].
  (D+1, D) – 1 → (D+1, D)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, 1 is subtracted from the contents of the 32-bit data that is a combination of data registers DT0 and DT1.



Precautions for programming

- If the result of an arithmetic operation instruction exceeds the numerical range that can be handled, an underflow will result.
- Under normal circumstances, do not allow an underflow to occur.
- If an underflow occurs, the CY flag (special internal relay R9009) will turn ON.

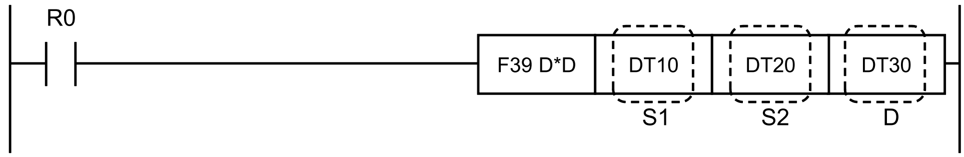
Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when the calculation result underflows

11.18 F39 D*D (32-bit Data Multiplication [S1*S2=D+1, D])

Multiplies 32-bit data items and stores the result in the 32-bit two-word area.

■ Instruction format



■ Operands

Items	Settings
S1	Multiplicand data: Area storing 32-bit data, or constant data
S2	Multiplier data: Area storing 32-bit data, or constant data
D	Storage destination: Area storing multiplication result (32-bit data)

■ Devices that can be specified (indicated by •)

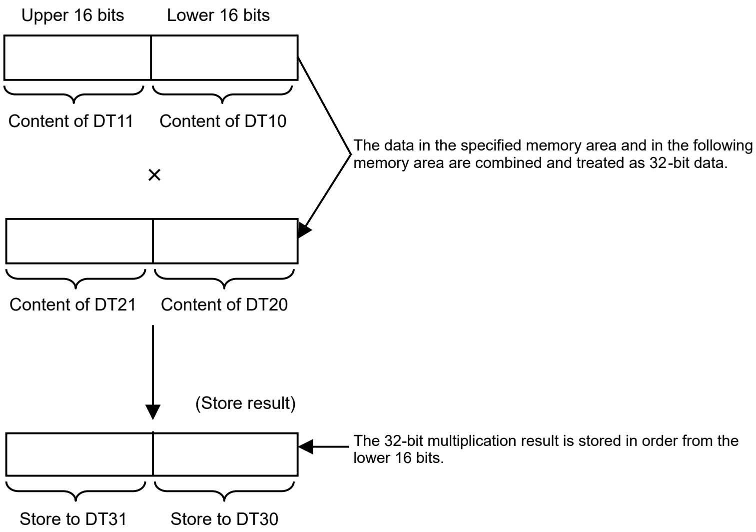
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The 32-bit data specified by [S1] and the 32-bit data specified by [S2] are multiplied, and the result is stored in the area specified by [D].
  (S1+1, S1) × (S2+1, S2) → (D+1, D)
- The operation result is stored as two words of 32-bit data.

■ Operation example

Operation of instruction format description program



■ Precautions for programming

Keep the operation result [D] within the range of K-2147483648 to K2147483647.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification. Turns ON when the operation result exceeds 32 bits
R900B (=)	Turns ON when the calculation result is"0"

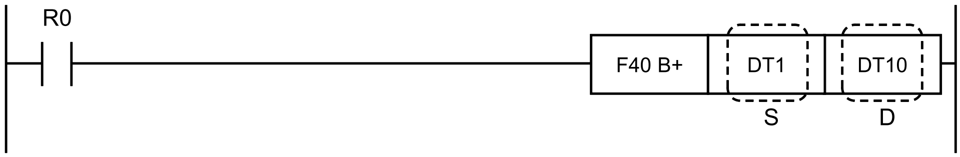
(MEMO)

12 BCD Data Arithmetic Instructions

12.1 F40 B+ (4-digit BCD Data Addition [D+S=D])

Adds 4-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the 4-digit BCD data to be added, or constant data
D	Area storing the 4-digit BCD data to be added to

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The 4-digit BCD data specified by [S] is added to the 4-digit BCD data (H constant) specified by [D].
  (D) + (S) → (D)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of data register DT10 is added to the content of data register DT1. If DT1 contains BCD 4 and DT10 contains 8, the result is as follows.

■ Precautions for programming

- If the result of an arithmetic operation instruction exceeds the maximum value that can be handled, this will result in an overflow.
- Ensure that overflows do not occur in normal circumstances.
- In the case of an overflow, use an 8-digit arithmetic operation instruction.
- If an overflow occurs, the CY flag (special internal relay R9009) turns ON.

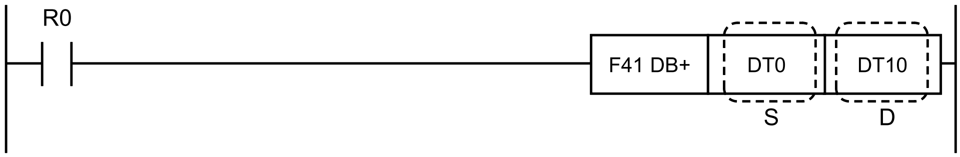
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the specified data is not BCD data
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows

12.2 F41 DB+ (8-digit BCD Data Addition [D+S=D])

Adds 8-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the 8-digit BCD data to be added, or constant data
D	Area storing the 8-digit BCD data to be added to

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

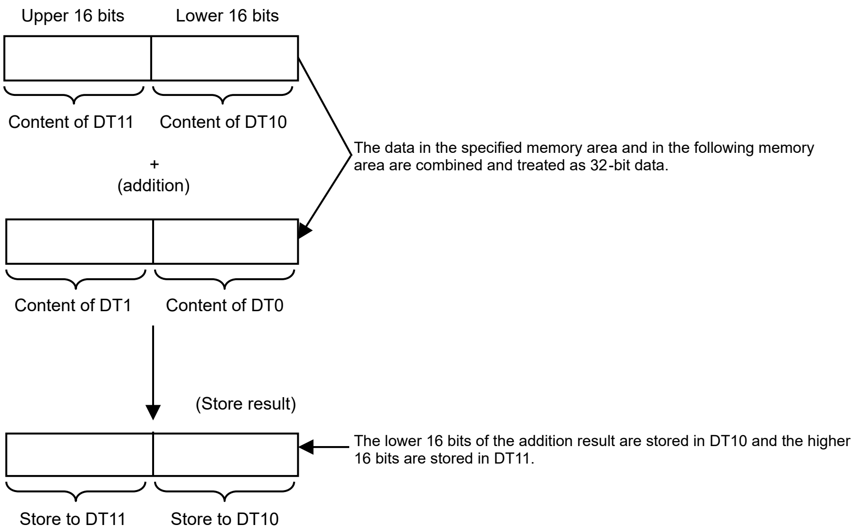
■ Outline of operation

- The 8-digit BCD data specified by [S] is added to the 8-bit BCD data (H constant) specified by [D].
  (D+1, D) + (S+1, S) → (D+1, D)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the contents of data registers DT0 and DT1 are added to the contents of data registers DT10 and DT11.



■ Precautions for programming

- If the result of an arithmetic operation instruction exceeds the maximum value that can be handled, this will result in an overflow.
- Ensure that overflows do not occur in normal circumstances.
- If an overflow occurs, the CY flag (special internal relay R9009) turns ON.

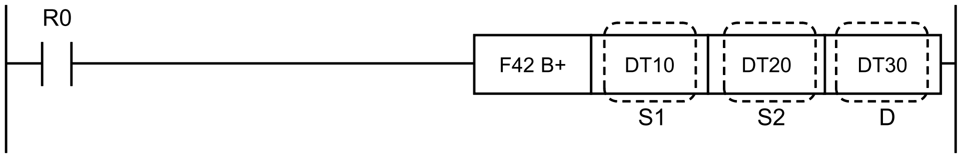
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification. Turns ON when the specified data is not BCD data
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows

12.3 F42 B+ (4-digit BCD Data Addition [S1+S2=D])

Adds 4-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the 4-digit BCD data to be added, or constant data
S2	Area storing the 4-digit BCD data to be added, or constant data
D	Area storing the addition results

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The 4-digit BCD data (H constant) specified by [S1] and [S2] are added together, and the result is stored in [D].
  (S1) + (S2) → (D)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the contents of data register DT10 and data register DT20 are added together, and the result is stored in data register DT30. If DT10 contains BCD 8 and DT20 contains BCD 4, the result is as follows.

	DT10																
S1:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
	0				0				0				8				(BCD)

+ (addition)

	DT20																
S2:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
	0				0				0				4				(BCD)

↓

	DT30																
D:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	
	0				0				1				2				(BCD)

■ Precautions for programming

- If the result of an arithmetic operation instruction exceeds the maximum value that can be handled, this will result in an overflow.
- Ensure that overflows do not occur in normal circumstances.
- In the case of an overflow, use an 8-digit arithmetic operation instruction.
- If an overflow occurs, the CY flag (special internal relay R9009) turns ON.

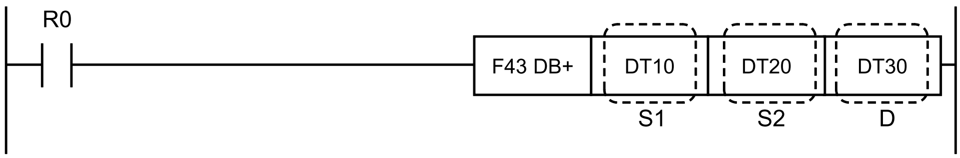
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the specified data is not BCD data
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows

12.4 F43 DB+ (8-digit BCD Data Addition [S1+S2=D])

Adds 8-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the 8-digit BCD data to be added, or constant data
S2	Area storing the 8-digit BCD data to be added, or constant data
D	Area storing the addition results

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

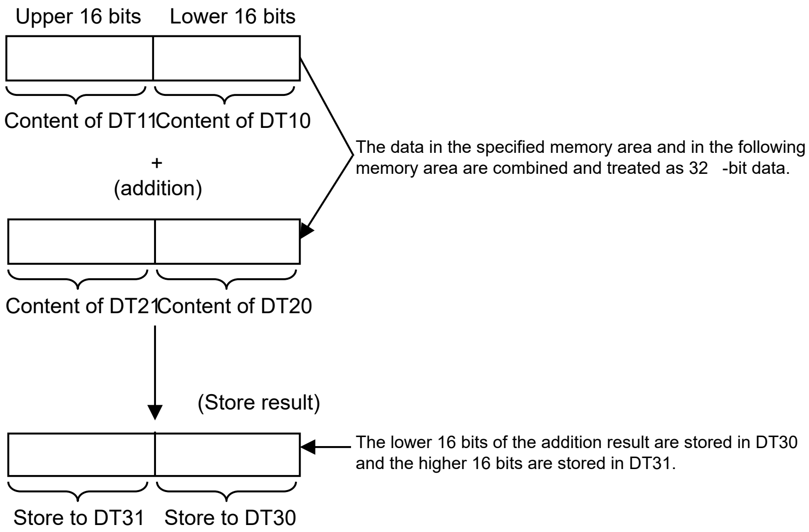
■ Outline of operation

- The 8-digit BCD data (H constant) specified by [S1] and [S2] are added together, and the result is stored in [D].
  (S1+1, S1) + (S2+1, S2) → (D+1, D)
- The memory area is specified by the memory area number of the lower 16-bit portion.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the contents of data registers DT10 and DT11 are added to the contents of data registers DT20 and DT21, and the result is stored in data registers DT30 and DT31.



■ Precautions for programming

- If the result of an arithmetic operation instruction exceeds the maximum value that can be handled, this will result in an overflow.
- Ensure that overflows do not occur in normal circumstances.
- If an overflow occurs, the CY flag (special internal relay R9009) turns ON.

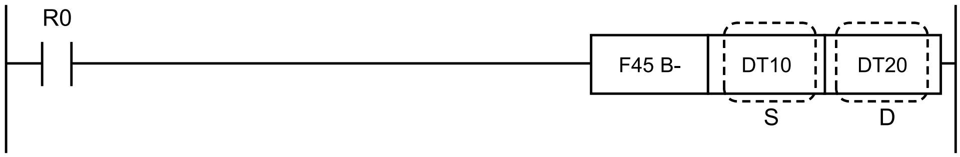
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification. Turns ON when the specified data is not BCD data
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows

12.5 F45 B- (4-digit BCD Data Subtraction [D-S=D])

Subtracts 4-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the subtrahend (4-digit BCD data) or constant data
D	Area storing the subtrahend (4-digit BCD data)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The 4-digit BCD data specified by [S] is subtracted from the 4-digit BCD data (H constant) specified by [D].
  (D) - (S) → (D)

■ Operation example

Operation of instruction format description program

Subtracts the contents of data register DT10 from the contents of data register DT20 when internal relay R0 turns ON. When BCD is 16 in DT20 and 4 in DT10, it is as shown below.

	DT20																
D:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	
	0				0				1				6				(BCD)

- (subtraction)

	DT10																
S:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
	0				0				0				4				(BCD)

↓

	DT20																
D:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	
	0				0				1				2				(BCD)

■ Precautions for programming

- If the result of an arithmetic operation instruction exceeds the maximum value that can be handled, this will result in an underflow.
- Under normal circumstances, do not allow an underflow to occur.
- If an underflow occurs, use an 8-digit arithmetic operation instruction.
- If an underflow occurs, the CY flag (special internal relay R9009) will turn ON.

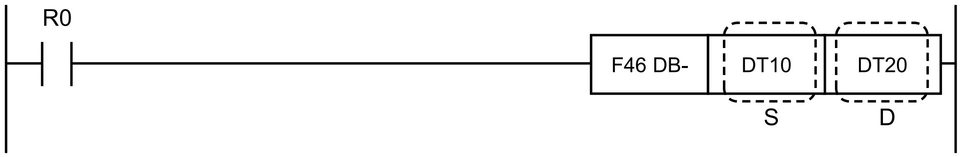
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the specified data is not BCD data
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when the calculation result underflows

12.6 F46 DB- (8-digit BCD Data Subtraction [D-S=D])

Subtracts 8-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S	Area that stores the subtrahend (8-digit BCD data), or constant data
D	Area storing the number to be subtracted (8-digit BCD data)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

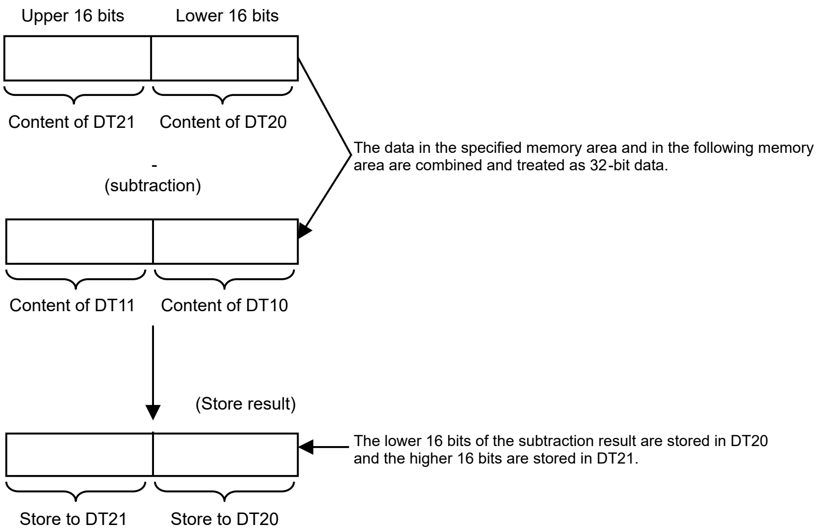
■ Outline of operation

- The 8-digit BCD data specified by [S] is subtracted from the 8-digit BCD data (H constant) specified by [D].
  (D+1, D) - (S+1, S) → (D+1, D)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of data registers DT10 and DT11 is subtracted from the content of data registers DT20 and DT21.



■ Precautions for programming

- If the result of an arithmetic operation instruction exceeds the maximum value that can be handled, this will result in an underflow.
- Under normal circumstances, do not allow an underflow to occur.
- If an underflow occurs, the CY flag (special internal relay R9009) will turn ON.

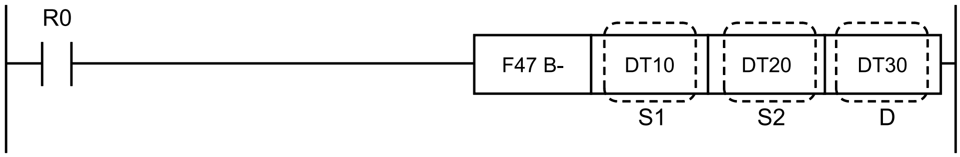
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification. Turns ON when the specified data is not BCD data
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when the calculation result underflows

12.7 F47 B- (4-digit BCD Data Subtraction [S1-S2=D])

Subtracts 4-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the minuend (4-digit BCD data), or constant data
S2	Area storing the subtrahend (4-digit BCD data) or constant data
D	Area that stores the calculation result

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The 4-digit BCD data specified by [S2] is subtracted from the 4-digit BCD data (H constant) specified by [S1], and the result is stored in [D].
  (S1) - (S2) → (D)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of data register DT20 is subtracted from the content of data register DT10, and the result is stored in data register DT30. If DT10 contains BCD 16 and DT20 contains BCD 4, the result is as follows.

	DT10																
S1:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	
	0				0				1				6				(BCD)

- (subtraction)

	DT20																
S2:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
	0				0				0				4				(BCD)

↓

	DT30																
D:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	
	0				0				1				2				(BCD)

■ Precautions for programming

- If the result of an arithmetic operation instruction exceeds the maximum value that can be handled, this will result in an underflow.
- Under normal circumstances, do not allow an underflow to occur.
- If an underflow occurs, use an 8-digit arithmetic operation instruction.
- If an underflow occurs, the CY flag (special internal relay R9009) will turn ON.

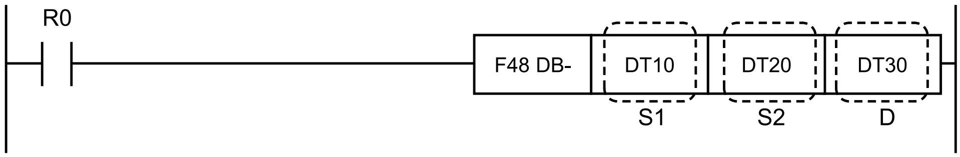
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification. Turns ON when the specified data is not BCD data
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when the calculation result underflows

12.8 F48 DB- (8-digit BCD Data Subtraction [S1-S2=D])

Subtracts 8-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S1	Area that stores the minuend (8-digit BCD data), or constant data
S2	Area that stores the subtrahend (8-digit BCD data), or constant data
D	Area that stores the calculation result

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

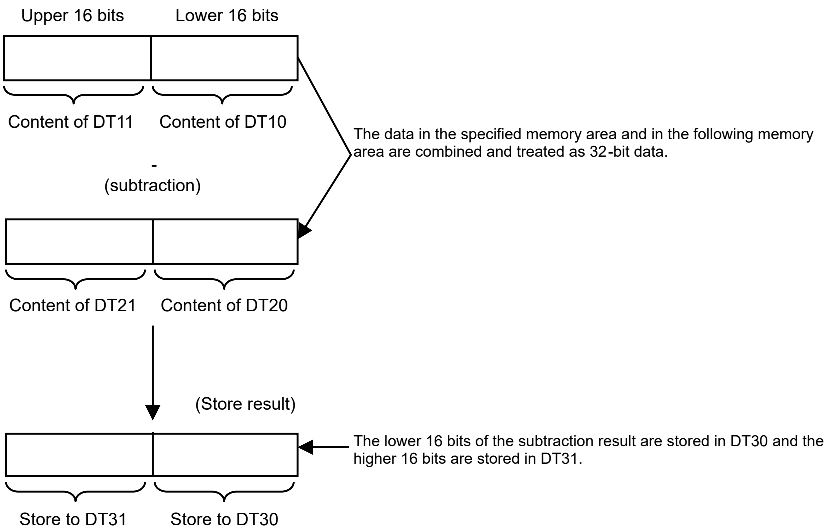
■ Outline of operation

- Subtracts the 8-digit BCD data specified by [S2] from the 8-digit BCD data (H constant) in the area specified by [S1], and stores the result in [D].
  (S1+1, S1) - (S2+1, S2) → (D+1, D)
- Memory area is specified by the memory area number of the lower order hexadecimal part.

■ Operation example

Operation of instruction format description program

Subtracts the contents of data registers DT20 to DT21 from the contents of data registers DT10 to DT11 when internal relay X0 turns ON, and stores the calculation result in data registers DT30 to DT31.



■ Precautions for programming

- If the result of an arithmetic operation instruction falls below the minimum value which can be handled, an underflow will result.
- Under normal circumstances, do not allow an underflow to occur.
- If an underflow occurs, the CY flag (special internal relay R9009) will turn ON.

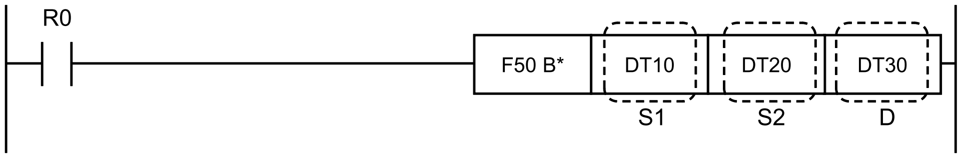
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the specified data is not BCD data
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when the calculation result underflows

12.9 F50 B* (4-digit BCD Data Multiplication [S1*S2=D+1, D])

Multiplies 4-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing 4-digit BCD data, or constant data
S2	Area storing 4-digit BCD data, or constant data
D	Area storing multiplication result (8-digit BCD data)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

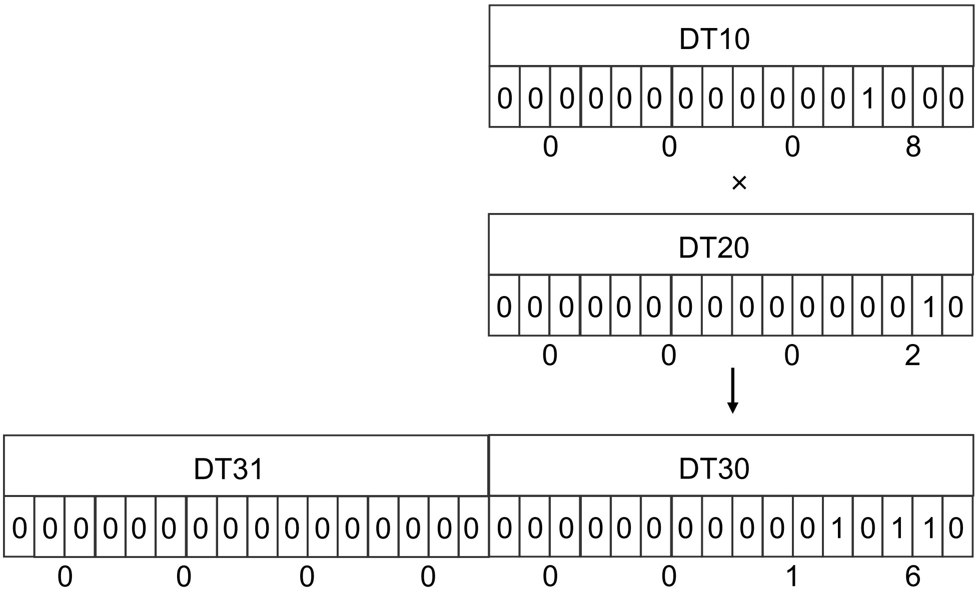
■ Outline of operation

- Multiplies the 4-digit BCD data specified by [S1] (H constant) by the 4-digit BCD data specified by [S2], and the result is stored in the area specified by [D].
  (S1) × (S2) → (D+1, D)
- The operation result is stored as 32-bit data (8-digit BCD).
- Storage destination [D] is specified by the number of the memory area with the lower order 16 bits.

■ Operation example

Operation of instruction format description program

e.g. If DT10 contains BCD 8 and DT20 contains BCD 2



Of the 32-bit data multiplication results, the lower order 16 bits are stored in the specified memory area (DT30) and the higher order 16 bits is stored in the next area after the specified area (DT31).

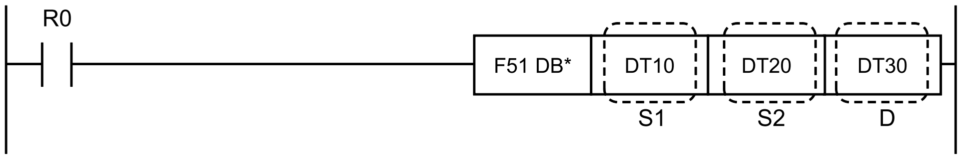
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the specified data is not BCD data
R900B (=)	Turns ON when the calculation result is"0"

12.10 F51 DB* (8-Digit BCD Data Multiplication [S1*S2=D+3, D+2, D+1, D])

Multiplies 8-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S1	Multiplicand data: Area storing 8-digit BCD data, or constant data
S2	Multiplier data: Area storing 8-digit BCD data, or constant data
D	Storage destination: Area storing multiplication result (64-bit data)

■ Devices that can be specified (indicated by ●)

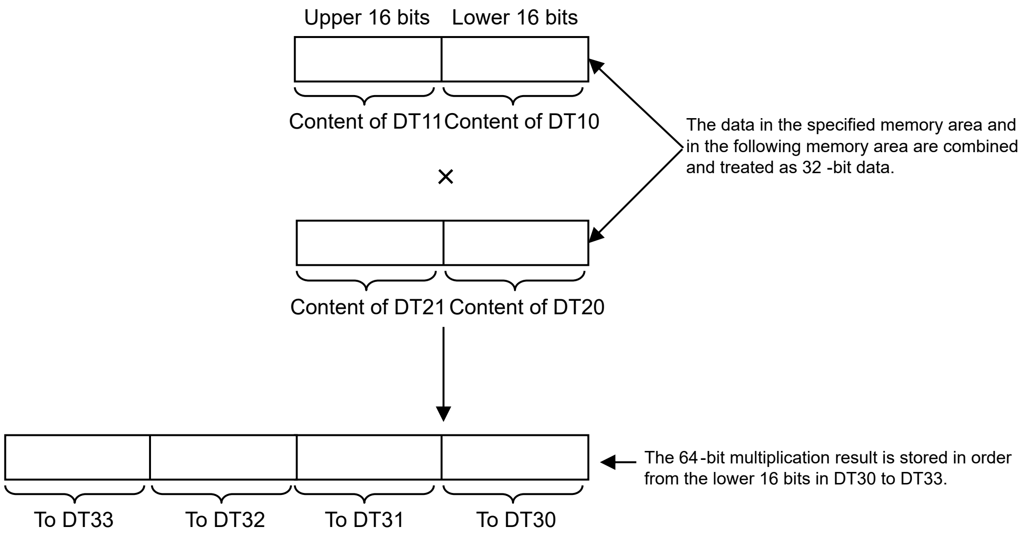
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●								●	

■ Outline of operation

- The 8-digit BCD data (H constant) specified by [S1] is multiplied by the 8-digit BCD data specified by [S2], and the result is stored in the area specified by [D].
  (S1+1, S1) x (S2+1, S2) → (D+3, D+2, D+1, D)
- The operation result is stored as 64-bit data (16-digit BCD).
- The memory area is specified by the number of the lowest 16-bit memory area.

■ Operation example

Operation of instruction format description program



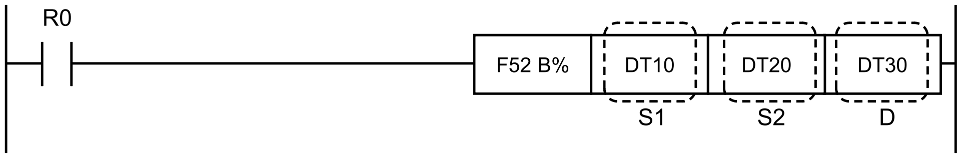
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification. Turns ON when the specified data is not BCD data
R900B (=)	Turns ON when the calculation result is"0"

12.11 F52 B% (4-digit BCD Data Subtraction [S1/S2=D])

Divides 4-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S1	Dividend data: Area storing 4-digit BCD data, or constant data
S2	Divisor data: Area storing 4-digit BCD data, or constant data
D	Storage destination: Area storing the divisor result (quotient) (remainder stored as 16-bit data in DT90015)

■ Devices that can be specified (indicated by •)

Operand s	WX	WY	WR	WL	SV	EV	DT	LD	I	SW R	SD T	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The 4-digit BCD data (H constant) specified by [S1] is divided by the 4-digit BCD data specified by [S2], with the quotient stored in [D] and the remainder stored in a special data register.
  (S1) ÷ (S2) → Quotient (D)
  Remainder (DT90015)

■ Operation example

Operation of instruction format description program

When the internal relay R0 turns ON, the contents of DT10 are divided by the contents of DT20, with the quotient stored in DT30 and the remainder stored as BCD in DT90015.

If DT10 contains BCD 15 and DT20 contains BCD 4, the result is as follows.

DT10															
0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1
0				0				1				5			

÷

DT20															
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0				0				0				4			

↓

DT30																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	Quotient
0				0				0				3				

DT90015																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	Remainder
0				0				0				3				

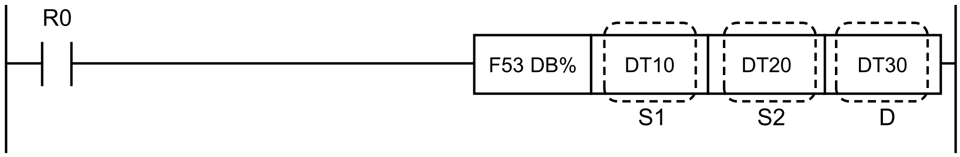
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the specified data is not BCD data
	Turns ON when S2 is"0"(when S1 is divided by"0")
R900B (=)	Turns ON when the operation result (quotient) is"0"

12.12 F53 DB% (8-digit BCD Data Subtraction [S1/S2=D+1, D])

Divides 8-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S1	Dividend data: Area storing 8-digit BCD data, or constant data
S2	Divisor data: Area storing 8-digit BCD data, or constant data
D	Storage destination: Area storing the divisor result (quotient) (remainder stored as 32-bit data in DT90015 and DT90016)

■ Devices that can be specified (indicated by •)

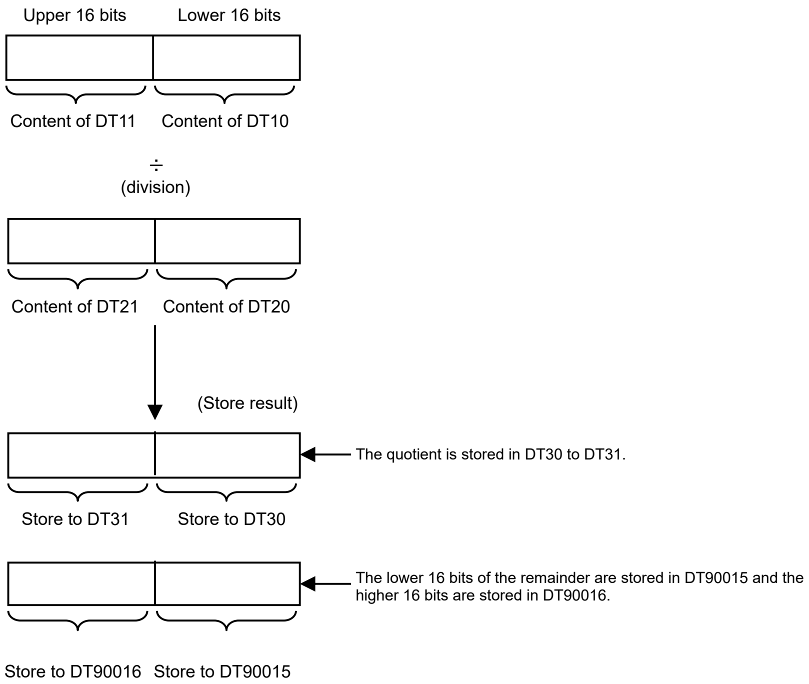
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The 8-digit BCD data (H constant) from the area specified by [S1] is divided by the 8-digit BCD data from the area specified by [S2]. The quotient is stored in the area specified by [D], and the remainder is stored as BCD in special data registers DT90015 and DT90016.
  (S1 + 1, S1) ÷ (S2 + 1, S2) → Quotient (D + 1, D)
  Remainder (DT90016, DT90015)
- Memory area is specified by the memory area number of the lower order hexadecimal part.

■ Operation example

Operation of instruction format description program



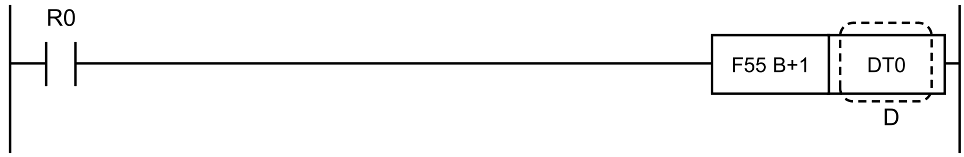
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the specified data is not BCD data
	Turns ON when S2 is"0"(when S1 is divided by"0")
R900B (=)	Turns ON when the operation result (quotient) is"0"

12.13 F55 B+1 (4-digit BCD Data Increment)

Adds 1 to 4-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
D	Area to which 1 is to be added

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	

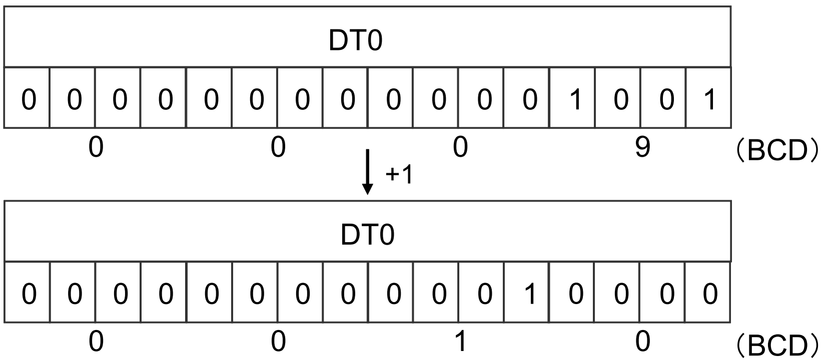
■ Outline of operation

- 1 is added to the 4-digit BCD data (H constant) specified by [D] and the result is stored in [D].
  (D) + 1 → (D)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, 1 is added to the contents of data register DT0.



■ Precautions for programming

- If the result of an arithmetic operation instruction exceeds the maximum value that can be handled, this will result in an overflow.

- Ensure that overflows do not occur in normal circumstances.
- In the case of an overflow, use an 8-digit arithmetic operation instruction.
- If an overflow occurs, the CY flag (special internal relay R9009) turns ON.

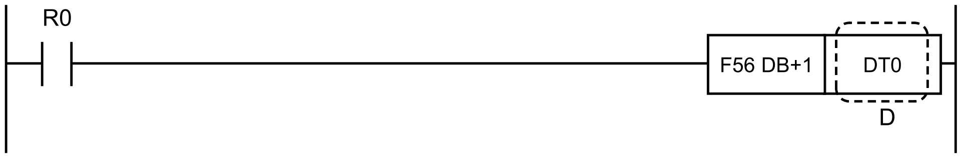
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the content of [D] is not BCD data (BCD error)
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows

12.14 F56 DB+1 (8-digit BCD Data Increment)

Adds 1 to the 8-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
D	The area (32-bit) that +1 is added to

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

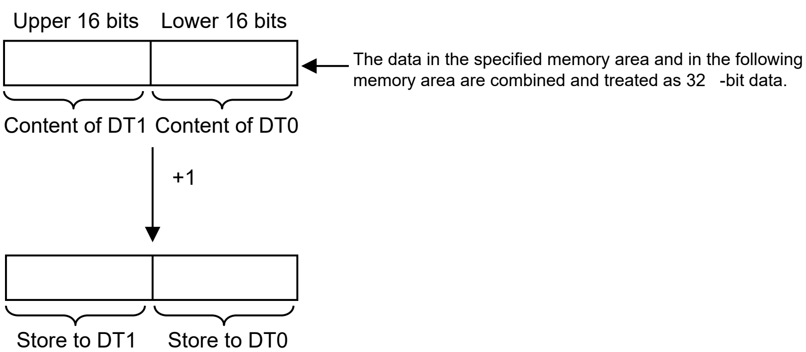
- Adds +1 to the 8-digit BCD data (H constant) specified by [D], then stores the result in the 2-word memory area starting with [D].

  (D+1, D) + 1 -> (D+1, D)

■ Operation example

Operation of instruction format description program

Adds 1 to the contents (8- digit BCD data) of data registers DT1 and DT0 when internal relay R0 turns ON.



Precautions for programming

- If the result of an arithmetic operation instruction exceeds the maximum value that can be handled, this will result in an overflow.
- Ensure that overflows do not occur in normal circumstances.
- If an overflow occurs, the CY flag (special internal relay R9009) turns ON.

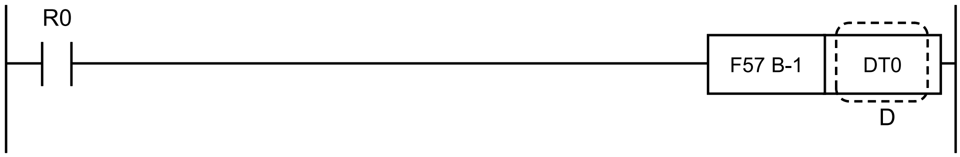
Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification. Turns ON when the content of the area specified by [D] is not BCD data (BCD error)
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when operation result overflows

12.15 F57 B-1 (4-digit BCD Data Decrement)

Subtracts 1 from 4-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
D	Area to be decreased by 1

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		●	●	●	●	●	●	●	●							●	

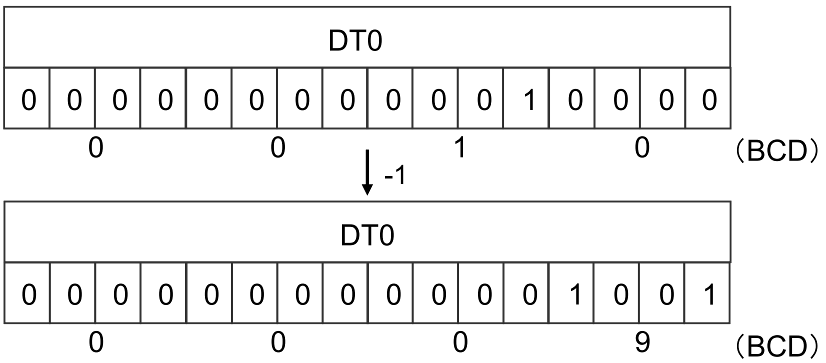
■ Outline of operation

- 1 is subtracted from the 4-digit BCD data (H constant) specified by [D] and the result is stored in [D].
  (D) - 1 → (D)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, 1 is subtracted from the content of data register DT0.



■ Precautions for programming

- If the result of an arithmetic operation instruction exceeds the maximum value that can be handled, this will result in an underflow.
- Under normal circumstances, do not allow an underflow to occur.
- If an underflow occurs, use an 8-digit arithmetic operation instruction.
- If an underflow occurs, the CY flag (special internal relay R9009) will turn ON.

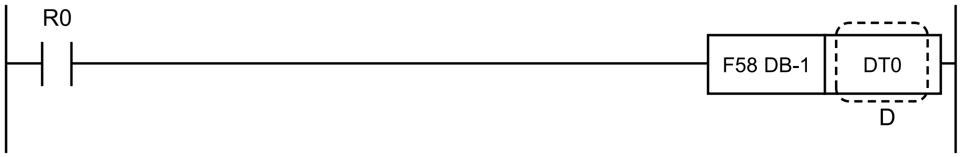
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification. Turns ON when the content of [D] is not BCD data (BCD error)
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when the calculation result underflows

12.16 F58 DB-1 (8-digit BCD Data Decrement)

Subtracts 1 from 8-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
D	Area (32-bit) from which 1 is subtracted

■ Devices that can be specified (indicated by •)

Operand s	WX	WY	WR	WL	SV	EV	DT	LD	I	SW R	SD T	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	

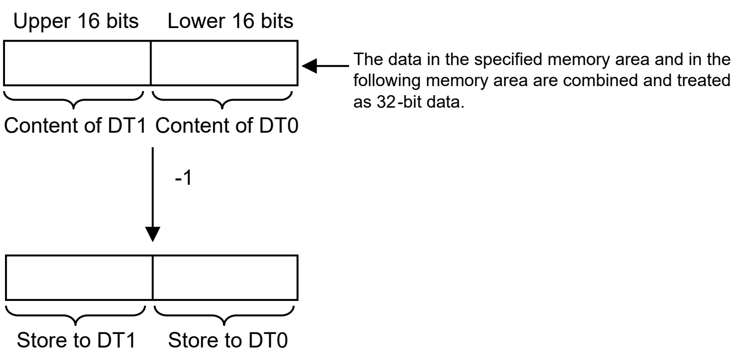
■ Outline of operation

- Subtracts 1 from 8-digit BCD data (H constant) specified by [D] and stores the result in the two-word memory area starting with [D].
  (D+1, D) - 1 -> (D+1, D)

■ Operation example

Operation of instruction format description program

Subtracts 1 from the 8-digit BCD data content of data registers DT0 and DT1 when internal relay R0 turns ON.



■ Precautions for programming

- If the result of an arithmetic operation instruction falls below the minimum value which can be handled, an underflow will result.
- Under normal circumstances, do not allow an underflow to occur.
- If an underflow occurs, the CY flag (special internal relay R9009) will turn ON.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the content of [D] is not BCD data (BCD error)
R900B (=)	Turns ON when the calculation result is"0"
R9009 (CY)	Turns ON when the calculation result underflows

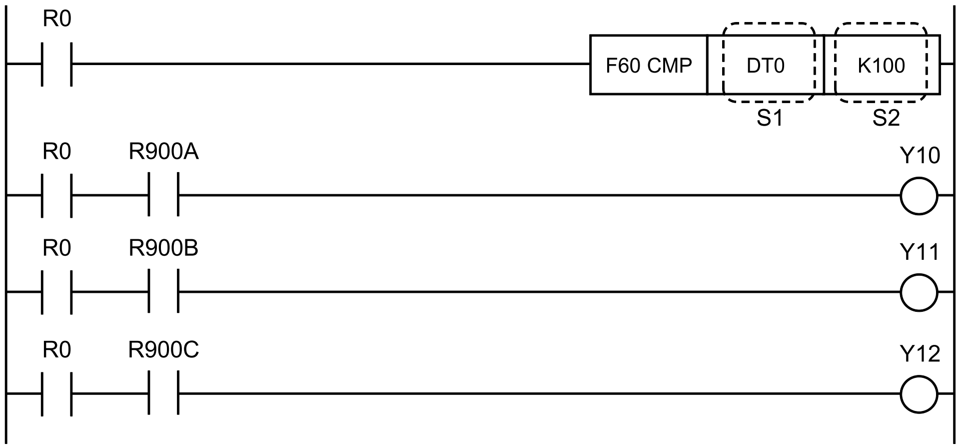
(MEMO)

13 Data Comparison Instructions

13.1 F60 CMP (16-bit Data Comparison)

Compares the two specified 16-bit data and outputs the judgment result to special internal relays.

■ Instruction format



■ Operands

Items	Settings
S1	Comparison data 1: Area storing 16-bit data, or constant data
S2	Comparison data 2: Area storing 16-bit data, or constant data

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	

■ Outline of operation

- The 16-bit data specified by [S1] expressing a decimal number is compared with the 16-bit data specified by [S2], and the judgment result is output to special internal relays R9009 to R900C (comparison instruction judgement flags).
- R9009 to R900C are assigned based on whether [S1] or [S2] is larger or smaller, as shown in the table below.

Relationship between S1 and S2	Flag			
	R900A	R900B	R900C	R9009
	>	=	<	Carry
S1 < S2	OFF	OFF	ON	Indefinite
S1 = S2	OFF	ON	OFF	OFF

Relationship between S1 and S2	Flag			
	R900A	R900B	R900C	R9009
	>	=	<	Carry
S1 > S2	ON	OFF	OFF	Indefinite

(Note 1) The above table shows the comparison results for signed integer.
When comparing unsigned integer or BCD data, refer to "P.13-6".

■ Operation example

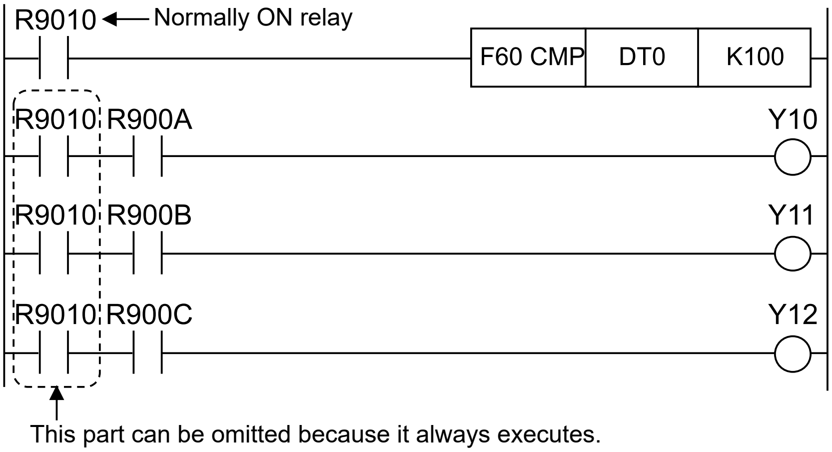
Operation of instruction format description program

When internal relay R0 turns ON and when the data register DT0 value is K100, output relay Y11 turns ON. When the value is smaller than K100, Y12 turns ON, and when the value is larger than K100, Y10 turns ON.

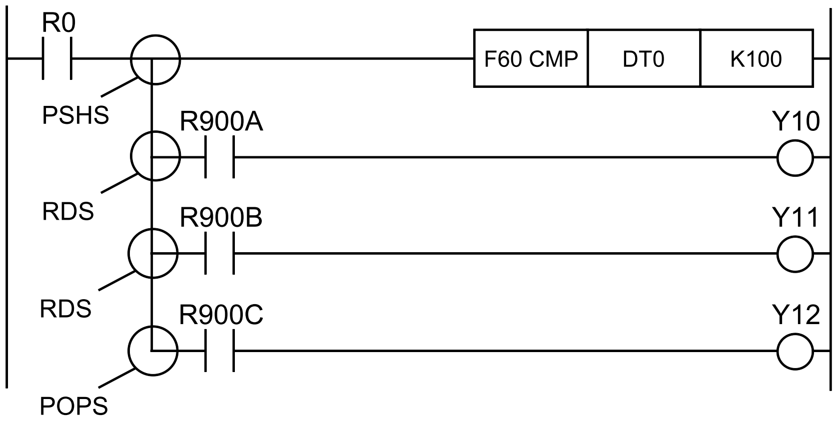
■ About internal relays

- In the program example on the previous page, comparison is only performed when R0 turns ON.
- If ongoing comparison is necessary, use relay R9010, which is always ON, as the internal relay.

e.g.



- The following programming is possible using instructions PSHS, RDS, and POPS.



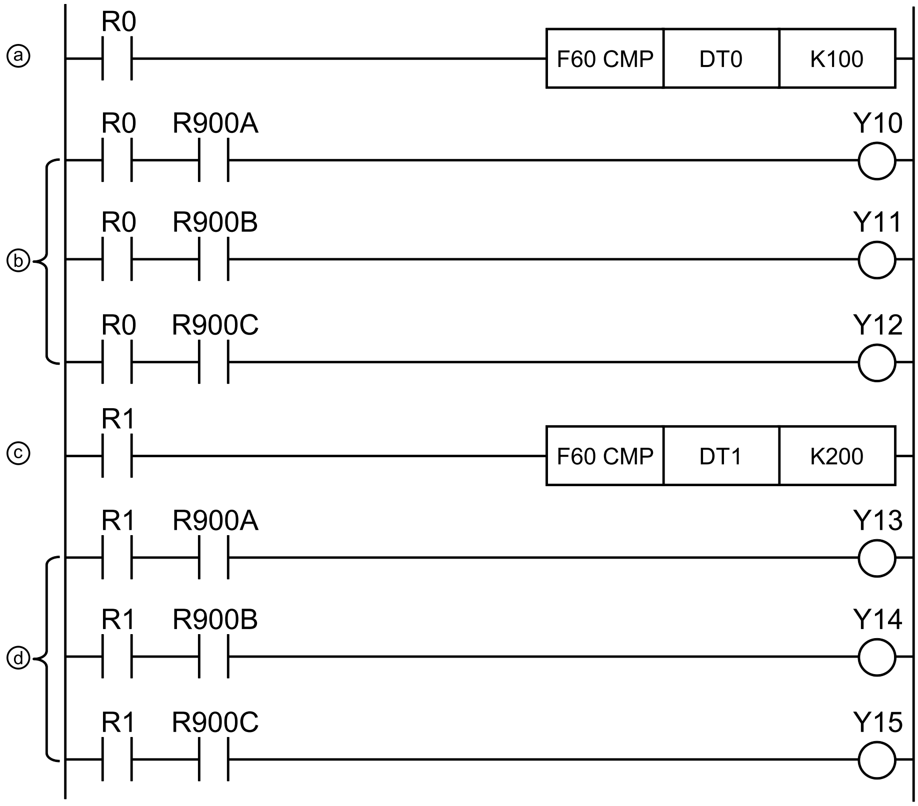
This program has the same operation as the program example.

■ Precautions when using two or more comparison instructions

- The comparison instruction judgment flags R900A to R900C are updated each time comparison instructions are executed.
- Therefore, when using two or more comparison instructions:

1. Insert programs using judgment flags immediately after the comparison instruction.
2. Output to the output relay or internal relay for each comparison instruction.

e.g. Example of comparison of DT0 and K100, and DT1 and K200

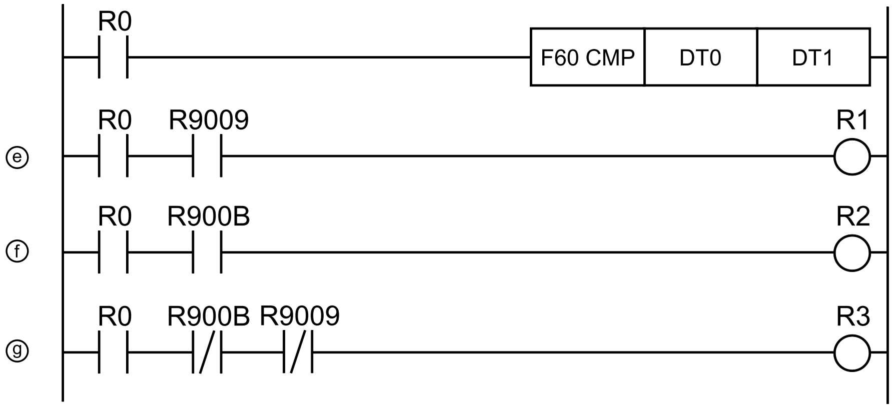


The comparison result for (a) is reflected in the contents of output relays Y10 to Y12 of program (b), and the comparison result for (c) is reflected in the contents of output relays Y13 to Y15 of program (d).

■ Precautions when comparing BCD data or external data

- When comparing BCD data or unsigned 16-bit data (0 to FFFF), construct a judgment program such as the one shown below using R900B and R9009 instead of R900A and R900C.

e.g. Comparing the BCD data in DT0 and DT1



(e)	When DT0 is less than DT1, R1 turns ON
(f)	When DT0 is equal to DT1, R2 turns ON
(g)	When DT0 is greater than DT1, R3 turns ON

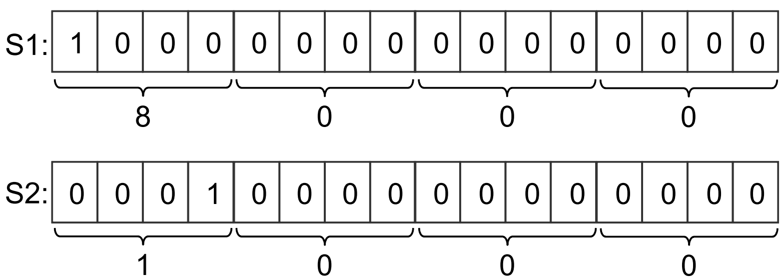
- Flag operation when comparing BCD data or unsigned 16-bit data (0 to FFFF)

Relationship between S1 and S2	Flag			
	R900A	R900B	R900C	R9009
	>	=	<	Carry
S1 < S2	Indefinite	OFF	Indefinite	ON
S1 = S2	OFF	ON	OFF	OFF
S1 > S2	Indefinite	OFF	Indefinite	OFF

(Note 1) The above table shows the comparison results for unsigned integer or BCD data.
When comparing signed data, refer to "P.13-2".

<Remarks>

For example, because R900A turns OFF and R900C turns ON when S1 = H8000 and S2 = H1000, accurate comparison results cannot be obtained with a judgment program that uses R900A and R900C.



■ Flag operations

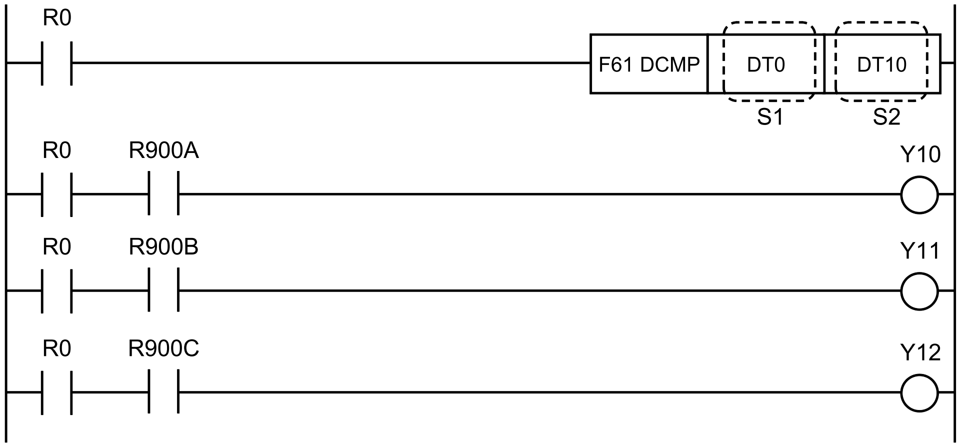
Name	Description
R9007	Turns ON when the area is exceeded in index modification.

Name	Description
R9008 (ER)	

13.2 F61 DCMP (32-bit Data Comparison)

Compares two specified 32-bit data, and outputs the result to special internal relays.

■ Instruction format



■ Operands

Items	Settings
S1	Comparison data 1: Area storing 32-bit data, or constant data
S2	Comparison data 2: Area storing 32-bit data, or constant data

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	

■ Outline of operation

- Compare the 32-bit data specified by [S1] and expressed as a decimal with 32-bit data in the area specified by [S2], and outputs the result to special internal relay flags (R9009 to R900C).
- R9009 to R900C are assigned based on whether [S1] or [S2] is larger or smaller, as shown in the table below.

Relationship between S1 and S2	Flag			
	R900A	R900B	R900C	R9009
	>	=	<	Carry
(S1+1, S1)<(S2+1, S2)	OFF	OFF	ON	Indefinite
(S1+1, S1)=(S2+1, S2)	OFF	ON	OFF	OFF
(S1+1, S1)>(S2+1, S2)	ON	OFF	OFF	Indefinite

(Note 1) The above table shows the comparison results for signed integer.
When comparing unsigned integer or BCD data, refer to "P.13-11".

- Memory area is specified by the memory area number of the lower order hexadecimal part.

■ Operation example

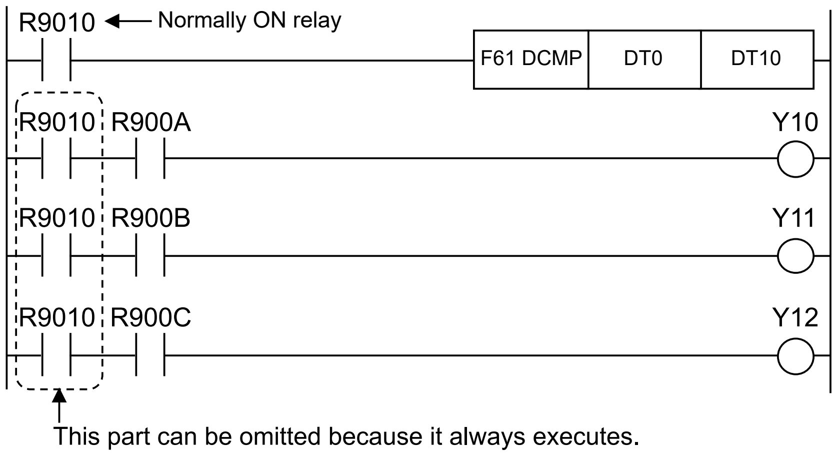
Operation of instruction format description program

When internal relay R0 is ON, the 32-bit data that is a combination of data registers DT0 and DT1 is compared with the 32-bit data that is a combination of data registers DT10 and DT11, and if the values of the two data are the same, the output relay Y11 turns ON. If the data in DT0 to DT1 is smaller than the data in DT10 to DT11, Y12 turns ON, and if it is larger Y10 turns ON.

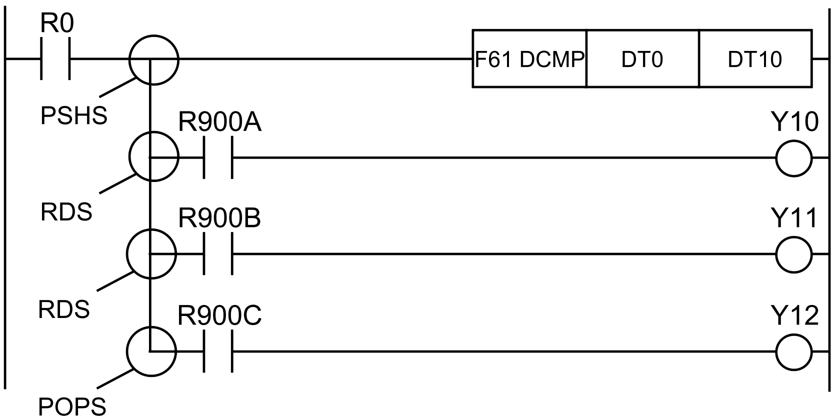
■ About internal relays

- In the above program example, the comparison is performed only when R0 is ON.
- If ongoing comparison is necessary, use relay R9010, which is always ON, as the internal relay.

e.g.



- The following programming is possible using instructions PSHS, RDS, and POPS.

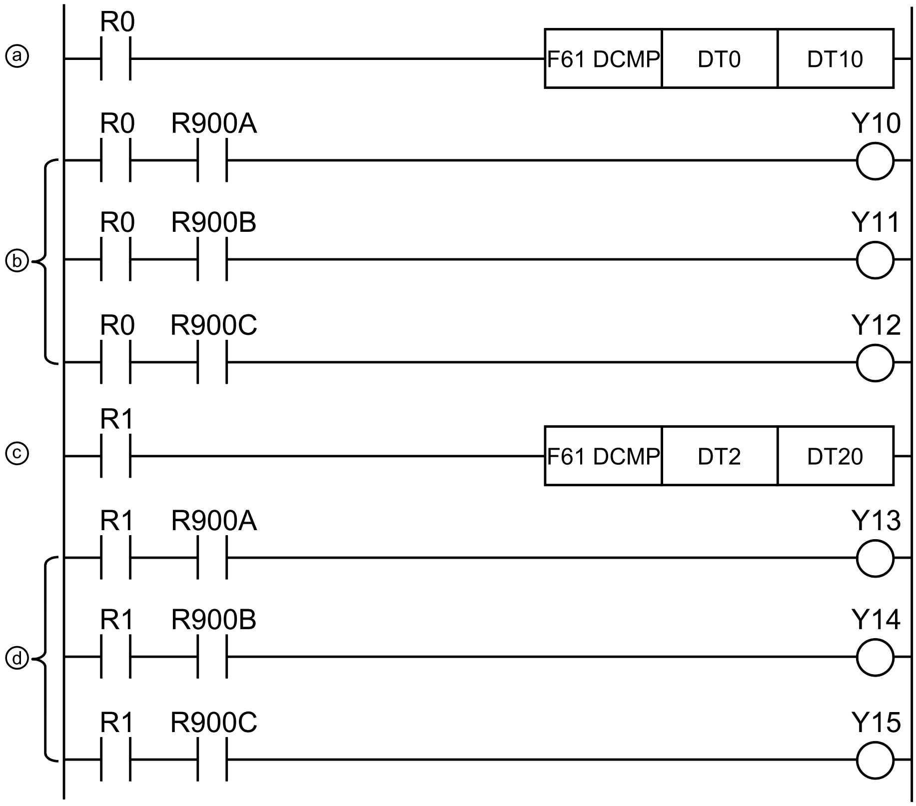


This program has the same operation as the program example.

■ Precautions when using two or more comparison instructions

- The comparison instruction judgment flags R900A to R900C are updated each time comparison instructions are executed.
- Therefore, when using two or more comparison instructions:
  1. Insert programs using judgment flags immediately after the comparison instruction.
  2. Output to the output relay or internal relay for each comparison instruction.

e.g. Comparison of DT0 to DT1 with DT10 to DT11, and DT2 to DT3 with DT20 to DT21

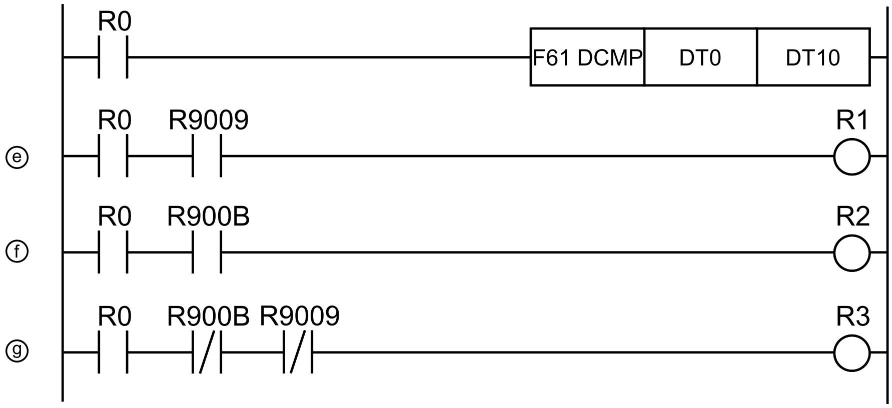


The comparison result for (a) is reflected in the contents of output relays Y10 to Y12 of program (b), and the comparison result for (c) is reflected in the contents of output relays Y13 to Y15 of program (d).

■ Precautions when comparing BCD data or external data

- When comparing BCD data or unsigned 16-bit data (0 to FFFFFFFF), do not use R900A and R900C. Use R900B and R9009, and create a judgment program such as the one shown below.

e.g. Comparing BCD data in DT0 to DT1 with BCD Data in DT10 to DT11



(e)	R1 turns ON when (DT1, DT0) < (DT11, DT10)
(f)	R2 turns ON when (DT1, DT0) = (DT11, DT10)
(g)	R3 turns ON when (DT1, DT0) > (DT11, DT10)

- Flag operations when comparing BCD data or unsigned 32-bit data (0 to FFFFFFFF)

Relationship between (S1+1, S1) and (S2+1, S2)	Flag			
	R900A	R900B	R900C	R9009
	>	=	<	Carry
(S1+1, S1)<(S2+1, S2)	Indefinite	OFF	Indefinite	ON
(S1+1, S1)=(S2+1, S2)	OFF	ON	OFF	OFF
(S1+1, S1)>(S2+1, S2)	Indefinite	OFF	Indefinite	OFF

(Note 1) The above table shows the comparison results for unsigned integer or BCD data.
When comparing signed data, refer to "P.13-8".

<Remarks>

For example, when S1 = H80000000 (K - 2,147,483,648) and S2 = H10000001 (K + 268,435,457), and when the F61 DCMP instruction is executed, the judgment is S1 < S2, R900A turns OFF, and R900C turns ON. Correct comparison results cannot be obtained with judgment programs that use R900A and R900C.

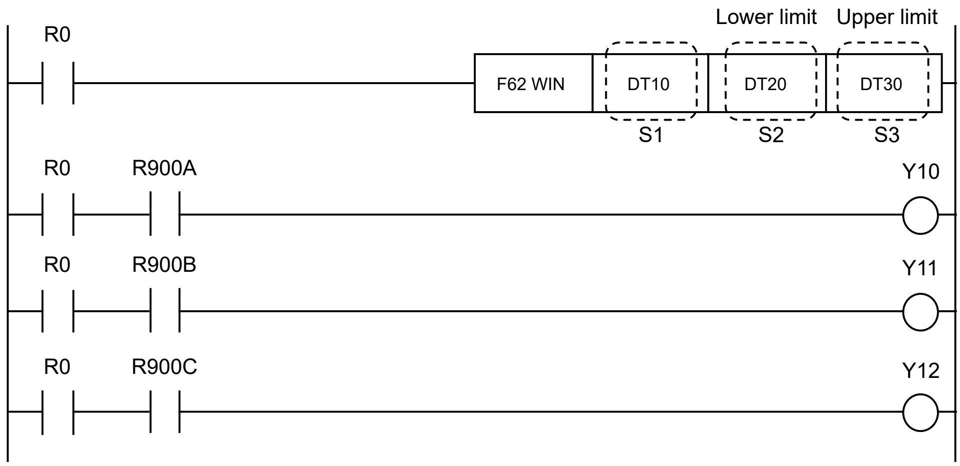
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

13.3 F62 WIN (16-bit Data Band Comparison)

Performs a band comparison of signed 16-bit data and outputs the comparison result to special internal relays.

■ Instruction format



■ Operands

Items	Settings
S1	Comparison data: Area storing 16-bit data, or constant data
S2	Lower limit data: Area storing 16-bit data, or constant data
S3	Upper limit data: Area storing 16-bit data, or constant data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S3	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

- A band comparison is performed on signed 162-bit data expressing a decimal number. The signed 16-bit data specified by [S1] is compared with the range specified by [S2] (lower limit value) and [S3] (upper limit value) to determine whether it falls in that range, and the comparison result is output to the special internal relays R9009 to R900C (comparison instruction judgment flag).
- The relationship between [S1], [S2], and [S3] affects R9009 to R900C as follows.

Relationship between S1, S2, and S3	Flag			
	R900A	**R900B**	**R900C**	**R9009**
	>	**=**	**<**	**Carry**
S1 < S2	OFF	OFF	ON	×
S2 ≤ S1 ≤ S3	OFF	ON	OFF	×
S3 < S1	ON	OFF	OFF	×

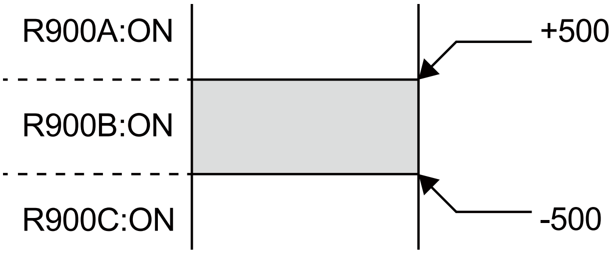
(Note 1) ×: Does not change.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the value of DT10 is compared with the range bounded by the lower limit value of DT20 and the upper limit value of DT30 to determine if it falls within that range.

e.g. When DT20 contains K-500 and DT30 contains K500



When DT10 = K-680	R900C: ON, Y12: ON
When DT10 = K-500	R900B: ON, Y11: ON
When DT10 = K256	R900B: ON, Y11: ON
When DT10 = K680	R900A: ON, Y10: ON

■ Precautions for programming

Set so that the lower limit value is equal to or less than the upper limit value (S2 ≤ S3).

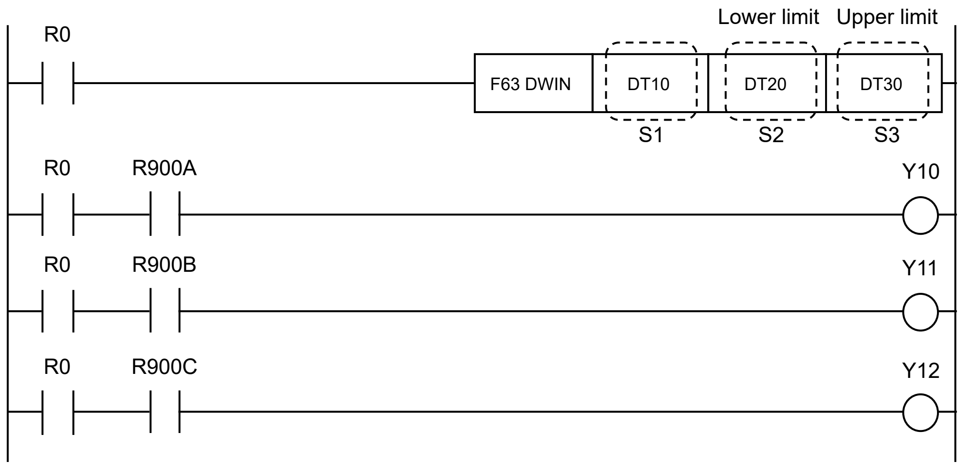
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	ON when S2 > S3

13.4 F63 DWIN (32-bit Data Band Comparison)

Performs a band comparison of signed 32-bit data and outputs the comparison result to special internal relays.

■ Instruction format



■ Operands

Items	Settings
S1	Comparison data: Area storing 32-bit data, or constant data
S2	Lower limit data: Area storing 32-bit data, or constant data
S3	Upper limit data: Area storing 32-bit data, or constant data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S3	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

- A band comparison is performed on signed 32-bit data expressing a decimal number. The signed 32-bit data specified by [S1] is compared with the range specified by [S2] (lower limit value) and [S3] (upper limit value) to determine whether it falls in that range, and the comparison result is output to the special internal relays R9009 to R900C (comparison instruction judgement flag).
- The 32-bit data specified by each operand is read from the next area.
  - [S1] = (S1+1, S1)
  - [S2] = (S2+1, S2)
  - [S3] = (S3+1, S3)

- The relationship between [S1], [S2], and [S3] affects R9009 to R900C as follows.

Relationship between S1, S2, and S3	Flag			
	R900A	R900B	R900C	R9009
	>	=	<	Carry
S1 < S2	OFF	OFF	ON	×
S2 ≤ S1 ≤ S3	OFF	ON	OFF	×
S3 < S1	ON	OFF	OFF	×

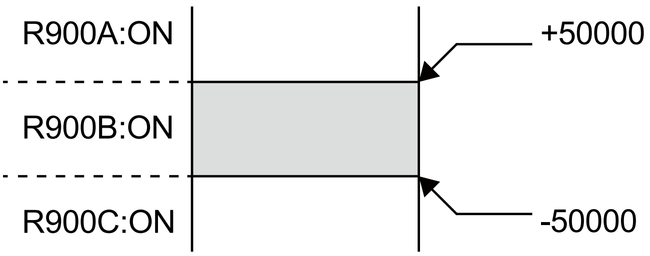
(Note 1) ×: Does not change.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the value of (DT11, DT10) is compared with the range bounded by the lower limit value of (DT21, DT20) and the upper limit value of (DT31, DT30) to determine if it falls within that range.

e.g. When DT20 and DT21 contain K–50000, and DT30 and DT31 contain K50000



When (DT11, DT10) = K–68000	R900C: ON, Y12: ON
When (DT11, DT10) = K–50000	R900B: ON, Y11: ON
When (DT11, DT10) = K25600	R900B: ON, Y11: ON
When (DT11, DT10) = K68000	R900A: ON, Y10: ON

■ Precautions for programming

Set so that the lower limit value (S2+1, S2) is equal to or less than the upper limit value (S3+1, S3).

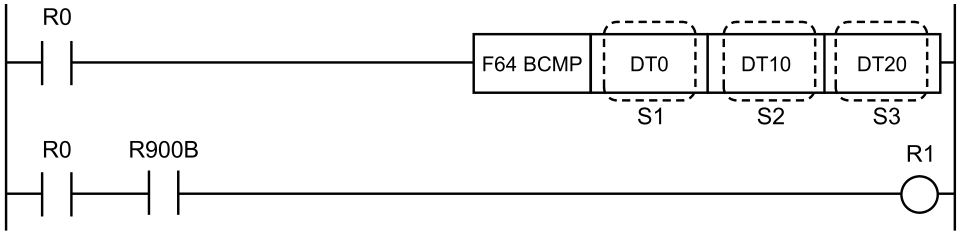
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when (S2+1, S2) is greater than (S3+1, S3)

13.5 F64 BCMP (Block Data Comparison)

Detects matches in two block-specified areas in byte units.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the control data (4-digit BCD data), or constant data
S2	Starting address of comparison block 1
S3	Starting address of comparison block 2

■ Devices that can be specified (indicated by •)

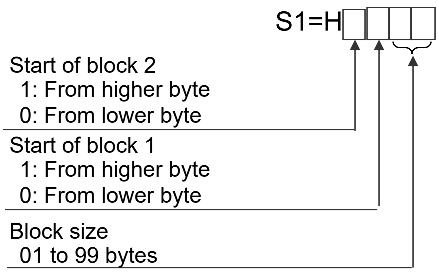
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•		•	•					•	
S3	•	•	•	•	•	•	•	•		•	•					•	

■ Outline of operation

- The contents of the area specified by [S2] (comparison block 1) are compared with the contents of the area specified by [S3] (comparison block 2).
- When the comparison result shows that the contents of the blocks match, special internal relay R900B ("="flag) turns ON.
- [S1] is the control data that determines factors such as the size of the comparison.

■ How to specify control data [S1]

Specify a 4-digit BCD (H constant) according to the following format.



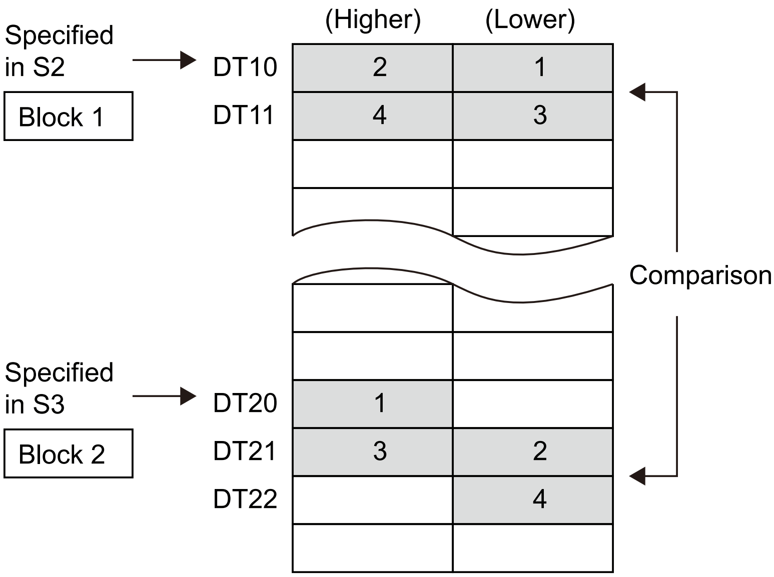
<Setting example>

When specifying the 4 bytes from the low byte of the area specified by [S2] as block 1 and the 4 bytes from the high byte of the area specified by [S3] as block 2, set [S1] to H1004.

■ Operation example

Operation of instruction format description program

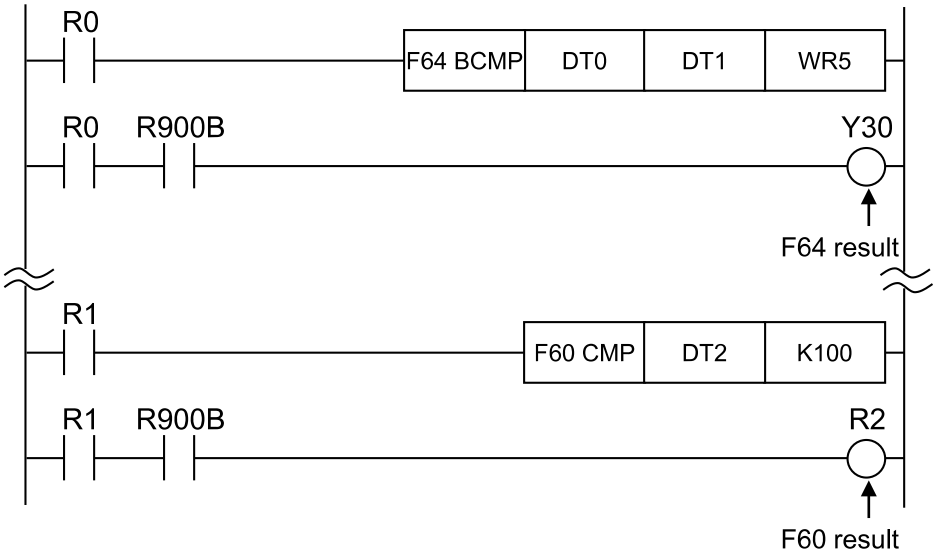
When internal relay R0 turns ON, the block starting at data register DT10 is compared with the block starting at data register DT20. When the values of the two blocks are the same, R1 turns ON. If H1004 is entered in DT0, the two blocks are as follows.



■ Precautions for programming

The flag R900B used for comparison instruction judgment is refreshed each time a comparison instruction, etc., is executed. Accordingly:

1. The program that uses R900B should be inserted immediately after the BCMP instruction.
2. Output the flag value to an output relay or internal relay and save the result.



(Note 1) As shown in the program example above, make sure to place the comparison internal relay before the flag relay. This is not necessary for normal execution.

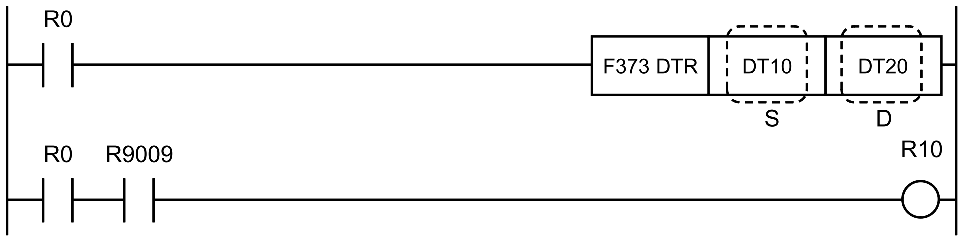
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the content specified by [S1] is not comprised of BCD data
	Turns ON when the specified block range exceeds the area

13.6 F373 DTR (16-bit Data Change Detection)

Detects changes in word data numerical values.

■ Instruction format



■ Operands

Items	Settings
S	Area that detects data changes
D	Area that stores data status during the previous execution

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•					•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- If the data in the area specified by [S] has changed since the previous time it was executed, the internal relay R9009 ("CY"flag) turns ON.
  [D] is used as an area for memorizing the preceding values, and the current values are stored when the instruction is completed.

■ Operation example

Operation of instruction format description program

When execution condition R0 is ON, if there are changes compared to when data register DT10 was previously executed, R9009 turns ON, and R10 also turns ON following this.

■ Precautions for programming

Flag R9009, which is used for detecting data changes, is updated each time a calculation instruction, etc. is executed. Therefore,

- a program using R9009 should be inserted immediately after the F373 DTR instruction.
- Output to an output relay or internal relay to hold the results.

Note

- Always insert execution conditions before the flag relay (R9009), as shown in the above program example. This is not necessary for normal execution.

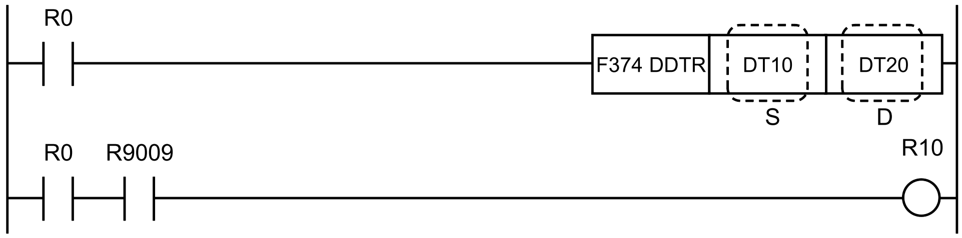
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Turns ON if there are changes to the specified data area

13.7 F374 DDTR (32-bit Data Change Detection)

Detects changes in double-word data (32-bit data) values.

■ Instruction format



■ Operands

Items	Settings
S	Area that detects data changes
D	Area that stores data status during the previous execution

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●					●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- Internal relay R9009 (the"CY"flag) turns ON if the data in the area specified by [S, S+1] has changed from the data values of the previous execution.
  [D, D+1] is used as an area for recording previous values, with the current values being stored when instruction execution is complete.

■ Operation example

Operation of instruction format description program

When execution condition R0 is ON, if there are changes compared to when data register DT10 was previously executed, R9009 turns ON, and R10 also turns ON following this.

■ Precautions for programming

Flag R9009, which is used for detecting data changes, is updated each time a calculation instruction, etc., is executed. Therefore:

- The program that uses R9009 should be inserted immediately after the F374 DDTR instruction.
- Output to an output relay or internal relay to hold the results.

Note

- Always insert execution conditions before the flag relay (R9009), as shown in the above program example. This is not necessary for normal execution.

■ Flag operations

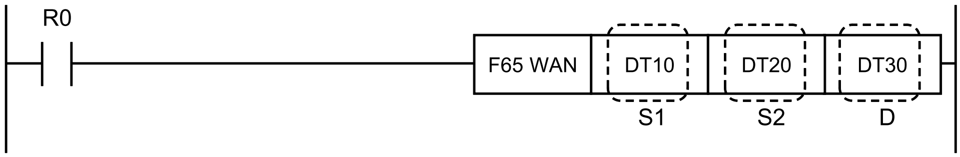
Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Turns ON if there are changes to the specified data area

14 Boolean Instructions

14.1 F65 WAN (16-bit Data AND)

Calculates the logical conjunction of 16-bit data.

■ Instruction format



■ Operands

Items	Settings
S1	Data 1: Area storing data on which to perform the logical operation, or constant data
S2	Data 2: Area storing data on which to perform the logical operation, or constant data
D	Storage location: Area storing the operation result

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- A bitwise logical conjunction is performed on each bit of the contents of the area specified by [S1] and the contents of the area specified by [S2], and the results are stored in [D].
  (S1) ^ (S2) → (D)
- This instruction can be used for operations such as forcibly turning OFF (bit masking) specific parts of data.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, a bitwise logical conjunction is performed on each bit of the contents of data register DT10 and the contents of data register DT20, and the results are stored in data register DT30.

	15	·	·	12	11	·	·	8	7	·	·	4	3	·	·	0 (bit)
DT10:	0	1	0	0	1	1	0	1	1	0	1	1	1	0	0	1
DT20:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

⇩ R0:ON

DT30:	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	1

■ Logical conjunction (AND)

S1 bit	S2 bit	Logical conjunction
0	0	0
0	1	0
1	0	0
1	1	1

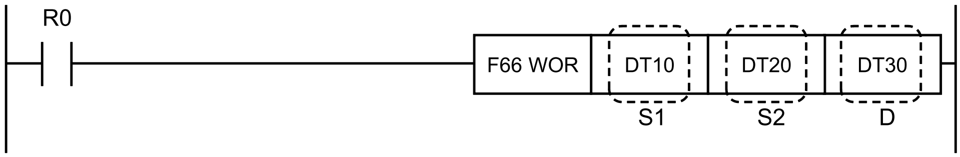
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"

14.2 F66 WOR (16-bit Data OR)

Calculates the OR of 16-bit data.

■ Instruction format



■ Operands

Items	Settings
S1	Data 1: Area storing data on which to perform the logical operation, or constant data
S2	Data 2: Area storing data on which to perform the logical operation, or constant data
D	Storage location: Area storing the operation result

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- A bitwise OR is performed on each bit of the contents of the area specified by [S1] and the contents of the area specified by [S2], and the results are stored in the area specified by [D].
  (S1) v (S2) → (D)
- This instruction can be used to forcibly turn ON specific parts of data.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, a bitwise OR is performed on each bit of the contents of data register DT10 and the contents of data register DT20, and the result is stored in data register DT30.

	15	·	·	12	11	·	·	8	7	·	·	4	3	·	·	0 (bit)
DT10:	0	1	0	0	1	1	0	1	1	0	1	1	1	0	0	1
DT20:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

R0:ON

DT30:	0	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1

■ Logical disjunction (OR)

S1 bit	S2 bit	Logical disjunction
0	0	0
0	1	1
1	0	1
1	1	1

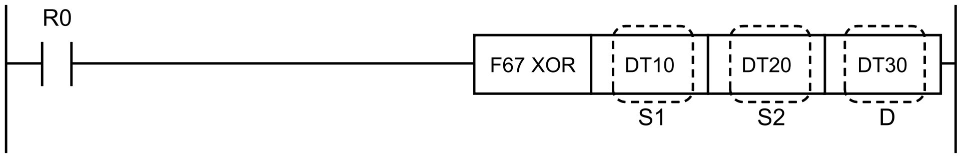
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"

14.3 F67 XOR (16-bit Data Exclusive OR)

Calculates the exclusive OR of 16-bit data.

■ Instruction format



■ Operands

Items	Settings
S1	Data 1: Area storing data on which to perform the logical operation, or constant data
S2	Data 2: Area storing data on which to perform the logical operation, or constant data
D	Storage location: Area storing the operation result

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- An exclusive OR is performed on each bit of the contents of the area specified by [S1] and the contents of the area specified by [S2], and the results are stored in the area specified by [D].
  {(S1) ^ (S2)} v {(S1) ^ (S2)} → (D)
- This can be used to detect bits whose ON/OFF status does not match.
- When the values of [S1] and [S2] are the same, all the bits in the data specified by [D] become 0.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, an exclusive OR is performed on each bit of the contents of data register DT10 and the contents of data register DT20, and the result is stored in data register DT30.

	15	·	·	12	11	·	·	8	7	·	·	4	3	·	·	0 (bit)
DT10:	0	1	0	0	1	1	0	1	1	0	1	1	1	0	0	1
DT20:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

R0:ON

DT30:	0	1	0	0	1	1	0	1	0	1	0	0	0	1	1	0

■ Exclusive OR (XOR)

S1 bit	S2 bit	Exclusive OR
0	0	0
0	1	1
1	0	1
1	1	0

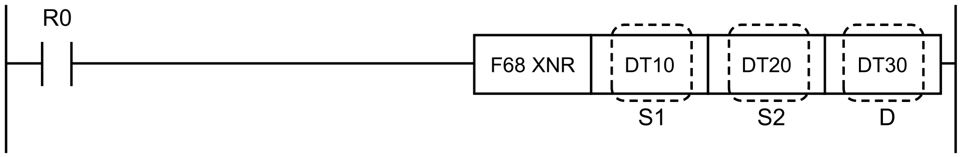
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"

14.4 F68 XNR (16-bit Data Exclusive NOR)

Calculates the exclusive NOR of 16-bit data.

■ Instruction format



■ Operands

Items	Settings
S1	Data 1: Area storing data on which to perform the logical operation, or constant data
S2	Data 2: Area storing data on which to perform the logical operation, or constant data
D	Storage location: Area storing the operation result

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- An exclusive NOR is performed on each bit of the contents of the area specified by [S1] and the contents of the area specified by [S2], and the results are stored in the area specified by [D].
  {(S1) ^ (S2)} v {(S1) ^ (S2)} → (D)
- This can be used to detect bits with matching ON/OFF status.
- When the values of [S1] and [S2] are the same, all the bits in the data specified by [D] become 1.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, if the values of the bits in the same positions in data registers DT10 and DT20 are equal, the bits in the same positions in data register DT30 turn ON (1). If they are not equal, they turn OFF (0).

	15	·	·	12	11	·	·	8	7	·	·	4	3	·	·	0 (bit)
DT10:	0	1	0	0	1	1	0	1	1	0	1	1	1	0	0	1
DT20:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

⇓ R0:ON

DT30:	1	0	1	1	0	0	1	0	1	0	1	1	1	0	0	1

■ Exclusive NOR (XNR)

S1 bit	S2 bit	Exclusive NOR
0	0	1
0	1	0
1	0	0
1	1	1

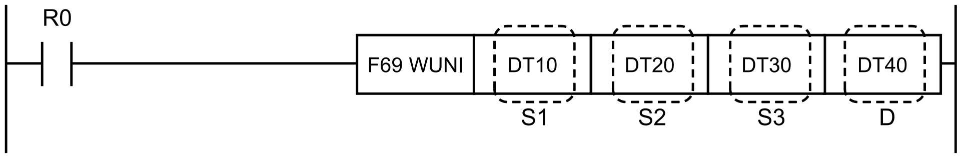
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"

14.5 F69 WUNI [(S1 AND S3) OR (S2 AND S3) = D] (16-bit)

Combines two sets of word data.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing data to be combined, or constant data
S2	Area storing data to be combined, or constant data
S3	Area storing mask data for combining, or constant data
D	Area that stores operation results

■ Devices that can be specified (indicated by •)

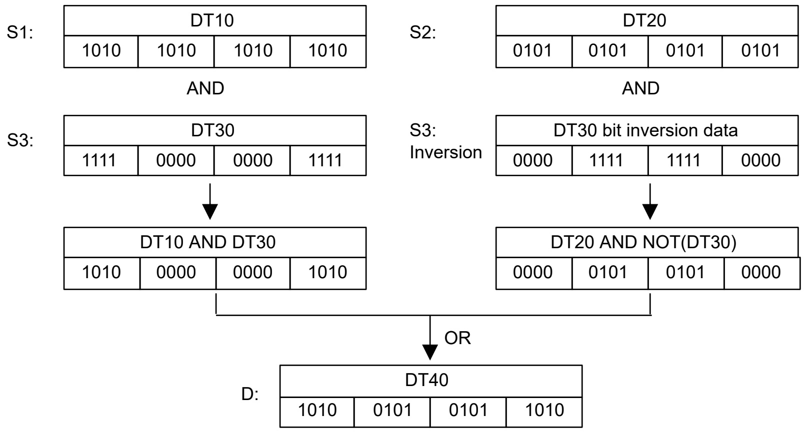
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S3	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- Using the mask data specified by [S3], the two sets of word data specified by [S1] and [S2] are combined in bit units, and stored in the area specified by [D].
  ([S1] AND [S3]) OR ([S2] AND [S3]) → [D]
  When [S3] is H0, [S2] is stored in [D]
  When [S3] is HFFFF, [S1] is stored in [D].

■ Operation example

Operation of instruction format description program



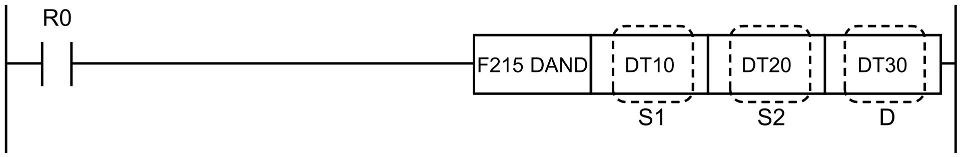
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"

14.6 F215 DAND (32-bit Data AND)

Calculates logical conjunction of double word data.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the data on which OR operations will be performed, or constant data (two words)
S2	Area storing the data on which OR operations will be performed, or constant data (two words)
D	Storage destination: Area that stores calculation results (two words)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- Takes the logical conjunction for each bit of the double word data specified by [S1, S1+1] and the double word data specified by [S2, S2+2], and stores the results in [D, D+1].

■ Operation example

Operation of instruction format description program

	15															0
DT10	1	0	0	1	0	1	1	0	0	0	1	0	0	0	0	1
DT11	1	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1

Logical conjunction

	15															0
DT20	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
DT21	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

	15															0
DT30	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
DT31	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1

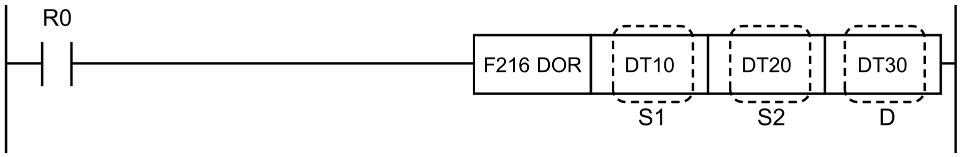
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"

14.7 F216 DOR (32-bit Data OR)

Performs OR operations double word data.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the data on which OR operations will be performed, or constant data (two words)
S2	Area storing the data on which OR operations will be performed, or constant data (two words)
D	Storage destination: Area that stores calculation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- Performs OR operation on each bit of the double word data specified by [S1, S1+1] and [S2, S2+1], and stores the results in [D, D+1].

■ Operation example

Operation of instruction format description program

	15															0
DT10	1	0	0	1	0	1	1	0	0	0	1	0	0	0	0	1
DT11	1	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1

Logical disjunction

	15															0
DT20	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
DT21	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

↓

	15															0
DT30	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	1
DT31	1	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1

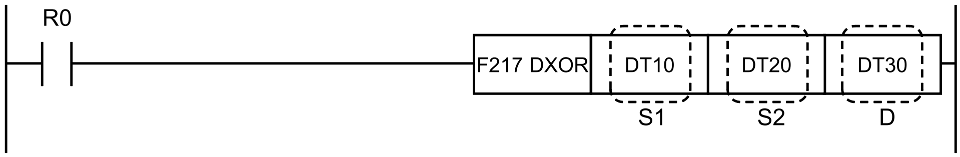
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"

14.8 F217 DXOR (32-bit Data Exclusive OR)

Calculates the exclusive OR of double-word data.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the data on which OR operations will be performed, or constant data (two words)
S2	Area storing the data on which OR operations will be performed, or constant data (two words)
D	Storage destination: Area that stores calculation results (two words)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- An exclusive OR is performed on each bit of the double-word data specified by [S1, S1+1] and the double-word data specified by [S2, S2+1], and the results are stored in the area specified by [D, D+1].
- This can be used to detect which bits are not the same.

  Matching bit = 0

  Non-matching bit = 1

■ Operation example

Operation of instruction format description program

	15															0
DT10	1	0	0	1	0	1	1	0	0	0	1	0	0	0	0	1
DT11	1	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1

Exclusive OR

	15															0
DT20	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
DT21	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

↓

	15															0
DT30	0	1	1	0	1	0	0	1	0	0	1	0	0	0	0	1
DT31	1	1	0	0	0	1	1	0	0	1	0	1	0	1	1	0

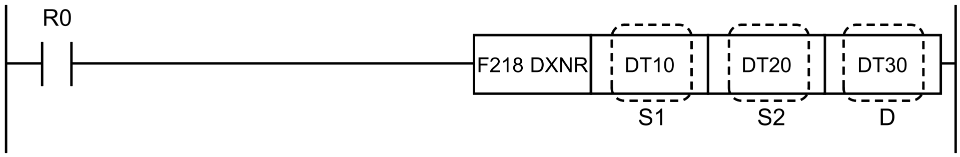
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"

14.9 F218 DXNR (32-bit Data Exclusive NOR)

Calculates the exclusive NOR of double word data.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the data on which OR operations will be performed, or constant data (two words)
S2	Area storing the data on which OR operations will be performed, or constant data (two words)
D	Storage destination: Area that stores calculation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- Retrieves the exclusive NOR of each bit for the double word data specified by [S1, S1+1] and the double word data specified by [S2, S2+1] before storing the result in [D, D+1].
- This can be used to determine whether each bit matches.

  Matching bit = 1

  Non-matching bit = 0

■ Operation example

Operation of instruction format description program

	15															0
DT10	1	0	0	1	0	1	1	0	0	0	1	0	0	0	0	1
DT11	1	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1

Exclusive NOR

	15															0
DT20	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
DT21	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

↓

	15															0
DT30	1	0	0	1	0	1	1	0	1	1	0	1	1	1	1	0
DT31	0	0	1	1	1	0	0	1	1	0	1	0	1	0	0	1

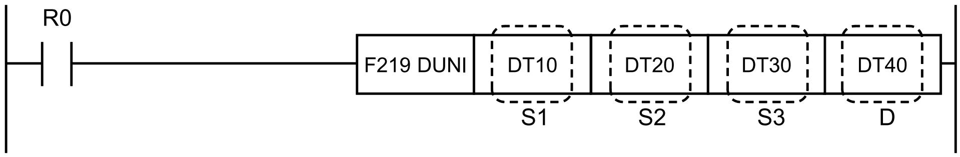
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"

14.10 F219 DUNI [(S1 AND S3) OR (S2 AND S3) = D] (32-bit)

Combines two double words.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the data to be combined, or constant data (two words)
S2	Area storing the data to be combined, or constant data (two words)
S3	Area storing mask data for combination, or constant data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

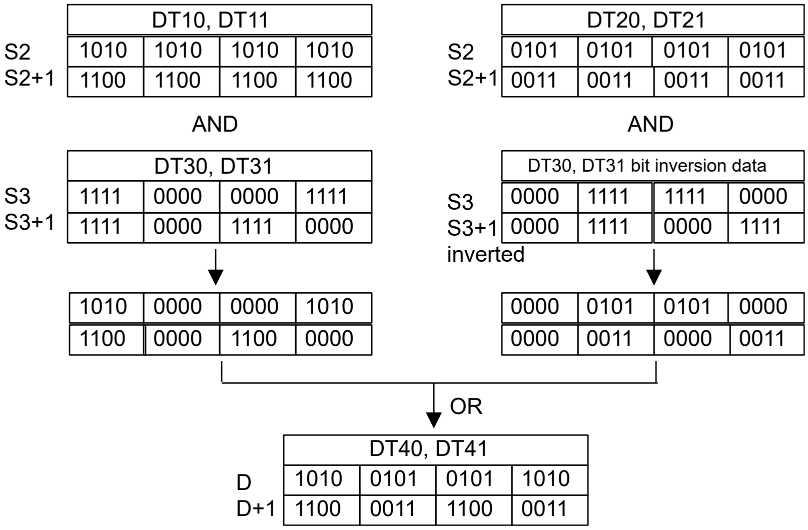
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S3	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- Using the mask data specified by [S3, S3+1], the two double word data specified by [S1, S1+1] and [S2, S2+1] are combined in bit units and stored in the area specified by [D, D+1].
  ([S1, S1+1] AND [S3, S3+1]) OR ([S2, S2+1] AND [S3, S3+1]) → [D, D+1]
- If [S3, S3+1] is H0, then [S2, S2+1] → [D, D+1]
- If [S3, S3+1] is HFFFFFFFF, then [S1, S1+1] → [D, D+1]

■ Operation example

Operation of instruction format description program



■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"

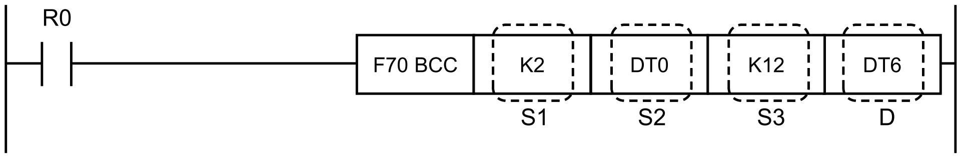
(MEMO)

15 Data Conversion Instructions

15.1 F70 BCC [Block Check Code (ADD, SUB, XOR, CRC)]

Calculates block check code (BCC).

■ Instruction format



■ Operands

Items	Settings
S1	Area storing data specifying the calculation method, or constant data
S2	Starting address of the area storing target data
S3	Area storing the length (number of bytes) of the target data, or constant data
D	Area that stores operation results

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●		●	●					●	
S3	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●								●	

■ Outline of operation

- Creates block check code (BCC) from the starting position for the calculation specified by S1 and S2 using the calculation method specified by S1, and stores the result at the storage position specified by D and S1 according to the conversion method specified by S1.

Specification of control data [S1]

H □□□□

Conversion data
- 0: Binary data
  (CRC: 2 bytes, Not CRC: 1 byte)
- 1: ASCII code (2 bytes)

Storage starting position (number of bytes from D)
- 0 to F

Storage starting position (number of bytes from S2)
- 0 to F

Specify the calculation method
- 0: Addition
- 1: Subtraction
- 2: Exclusive OR
- A: CRC-16

(Note 1) If CRC−16 is specified as the calculation method, ASCII code cannot be specified for the conversion data.

■ Calculation method

If the calculation method specified by [S1] is CRC, the calculation is carried out using the following generator polynomial. (Same calculation method as MODBUS−RTU.)

Generator polynomial: $X^{16}+X^{15}+X^{2}+1$

■ Operation example

Operation of instruction format description program

When the internal relay R0 turns ON, this calculates the BCC for the 12 bytes of data stored starting from data register DT0, via an exclusive OR operation. The result is stored in the lower byte of DT6.

■ Usage example 1

In this example, the block check code of the message being sent"%01#RCSX0000"is calculated and is added after the message.

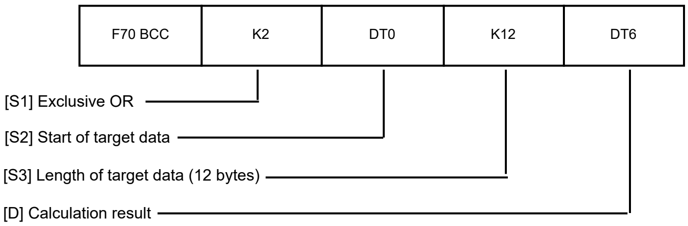
- Transmission is performed using ASCII codes.
- BCC is calculated via an exclusive OR.

1. The message should be stored in the memory area as shown below.

DT6	DT5	DT4	DT3	DT2	DT1	DT0
	H3030	H3030	H5853	H4352	H2331	H3025
↑ BCC	00	00	XS	CR	#1	0%

$\underbrace{\text{DT5 to DT0}}_{\text{12 bytes}}$

2. The BCC instruction is as shown below.

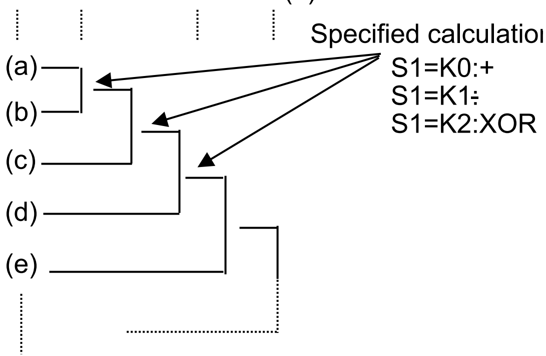


> When this is executed, BCC (H 1D) is stored in the lower byte of DT6 of [D].

- Calculation method

  Calculation is performed as shown below. (Explained in Usage example 2.)

% → H25 → 00100101 (a)
0 → H30 → 00110000 (b)
1 → H31 → 00110001 (c)
→ H23 → 00100011 (d)
R → H52 → 01010010 (e)



Calculation is performed in the order of carrying out the specified calculation in 8-bit units, and carrying out calculations on that result with the next 8 bits.

■ Usage example 2

In this example the block check code of the message being sent"%01#RCSX0000"is calculated and is added at the end of the message

- Calculation method: addition, conversion data: binary data

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
H0000	H0000	H0000	H0000	H3030	H3030	H5853	H4352	H2331	H3025
				00	00	XS	CR	#1	0%

Specification of control data [S1]
DT10 = H0C00

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
H0000	H0000	H0000	H00A9	H3030	H3030	H5853	H4352	H2331	H3025
				00	00	XS	CR	#1	0%

- Calculation method: addition, conversion data: ASCII code

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
H0000	H0000	H0000	H0000	H3030	H3030	H5853	H4352	H2331	H3025
				00	00	XS	CR	#1	0%

Specification of control data [S1]
DT10 = H1C00

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
H0000	H0000	H0000	H3941	H3030	H3030	H5853	H4352	H2331	H3025
			9A	00	00	XS	CR	#1	0%

- Calculation method: addition, conversion data: ASCII code

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
H0000	H0000	H0030	H3030	H3058	H5343	H5223	H3130	H2500	H0000
		0	00	0X	SC	R#	10	%	

Specification of control data [S1]
DT10 = H1F30

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
H0000	H0039	H4130	H3030	H3058	H5343	H5223	H3130	H2500	H0000
	9	A0	00	0X	SC	R#	10	%	

- Calculation method: CRC, conversion data: binary data

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
H0000	H0000	H0000	H0000	H3030	H3030	H5853	H4352	H2331	H3025
				00	00	XS	CR	#1	0%

Specification of control data [S1]
DT10 = H0C0A

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
H0000	H0000	H0000	H2E0A	H3030	H3030	H5853	H4352	H2331	H3025
				00	00	XS	CR	#1	0%

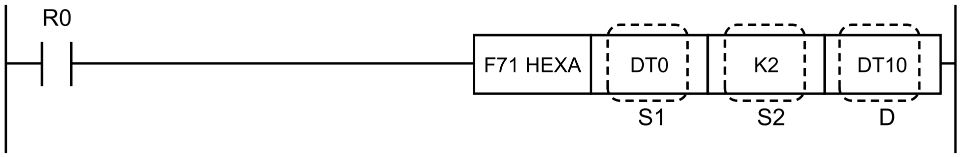
■ Flag operations

Name	Description
R9007 R9008 (ER)	When the calculation method specified by S1 is outside the specified range
	When the conversion data specified by S1 is outside the specified range

15.2 F71 HEXA (Hexadecimal Data to ASCII Code Conversion)

Converts hexadecimal numeric values to ASCII code.

■ Instruction format



■ Operands

Items	Settings
S1	Starting number for the area storing the hexadecimal numeric values
S2	Area storing the length of the numeric value (number of bytes) to be converted, or constant data
D	Starting number of the area storing the ASCII code of conversion result

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●		●	●					●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●								●	

■ Outline of operation

- The hexadecimal numeric data stored in the area specified by [S1] is converted to ASCII codes and stored in the area specified by [D].
- [S2] specifies the number of data bytes to be converted.
- The amount of the result (ASCII code) is twice the converted data.

■ Operation example

Operation of instruction format description program

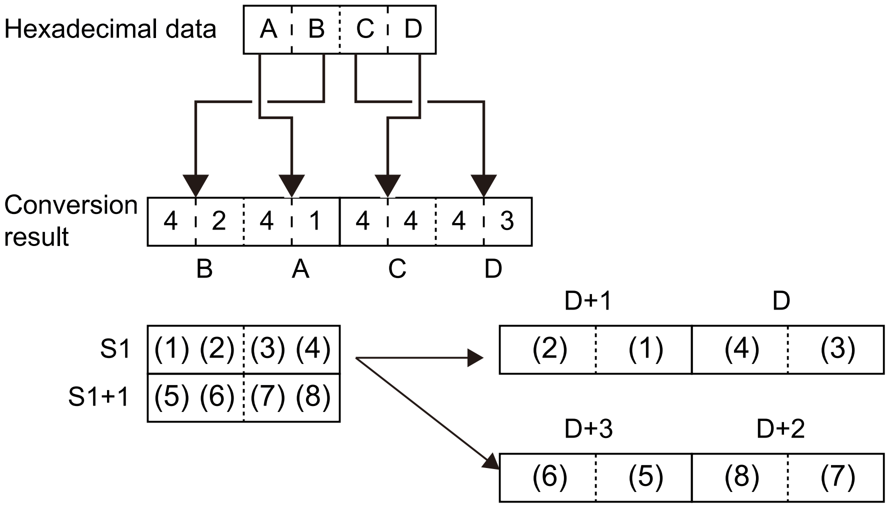
When internal relay R0 turns ON, the hexadecimal numeric data stored in data register DT0 (two bytes) is converted to ASCII codes and stored in DT10 and DT11.

Hexadecimal number (DT0)
H ABCD

ASCII code (DT11, DT10)
H 42 41 44 43
B A D C
DT11 DT10

■ Precautions for programming

1. The two characters that make up one byte are interchanged when stored.
2. Converts two bytes as one section.

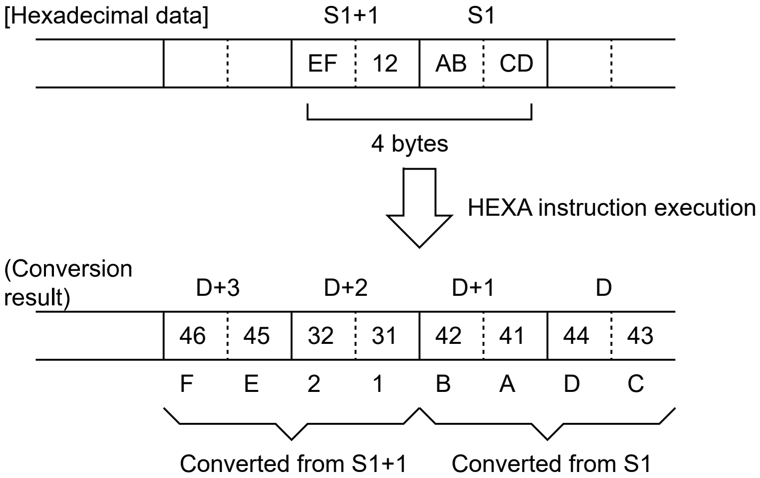


Hexadecimal data Conversion result

■ Conversion example

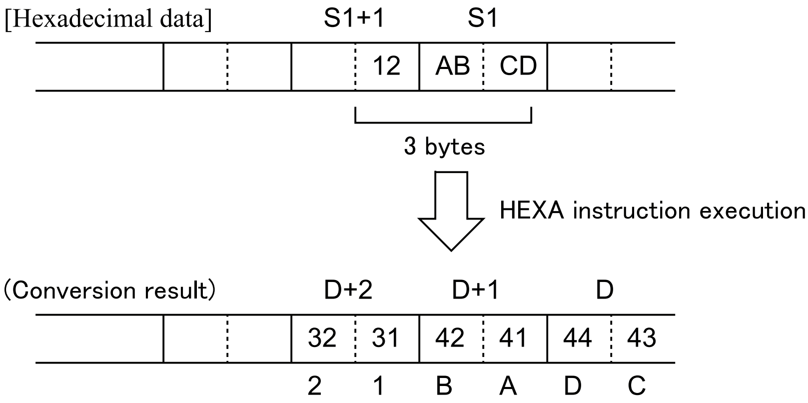
The following shows the conversion of hexadecimal number data to ASCII code.

Conversion of 4 bytes (S2 = K4)



Conversion of 3 bytes (S2 = K3)

Since the data to be converted is specified in byte units, it is also possible to convert only the low byte of one-word data.



■ Reference: ASCII code

		Higher	
		3	**4**
Lower	0	0	@
	1	1	A
	2	2	B
	3	3	C
	4	4	D
	5	5	E
	6	6	F
	7	7	G
	8	8	H
	9	9	I

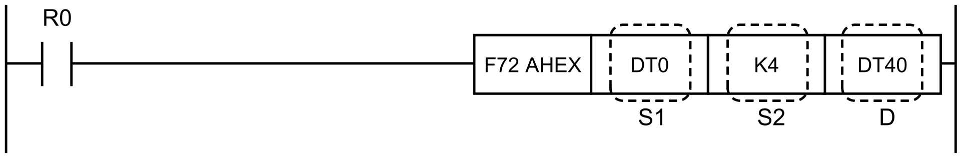
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the conversion range of the number of bytes specified by [S2] exceeds the area
	Turns ON when the conversion result exceeds the area
	Turns ON when the [S2] specification is"0"

15.3 F72 AHEX (ASCII Code to Hexadecimal Data Conversion)

Converts character strings in ASCII code to hexadecimal numbers.

■ Instruction format



■ Operands

Items	Settings
S1	Starting number of the area storing the ASCII code
S2	Area storing the number of ASCII codes (number of characters) to be converted, or constant data
D	Number of the start of the area storing the hexadecimal number that is the result of conversion

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•		•	•					•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•								•	

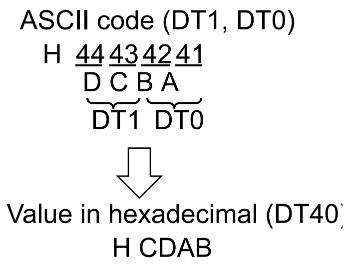
■ Outline of operation

- The ASCII codes stored in the area specified by [S1] are converted into hexadecimal numeric data and stored in the area specified by [D].
- The number of ASCII codes (number of characters) to be converted is specified by [S2].
- The volume of the result (hexadecimal numeric data) is half that of the converted ASCII codes.

■ Operation example

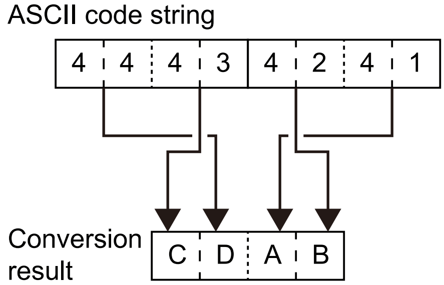
Operation of instruction format description program

When the internal relay R0 turns ON, the ASCII codes stored in data registers DT0 and DT1 (four characters) are converted into hexadecimal numeric data and stored in DT40.



Precautions for programming

1. Two ASCII code characters are converted into two 1-byte numeric digits. At this time, the upper and lower characters are interchanged.
2. Four characters are converted as one segment of data.

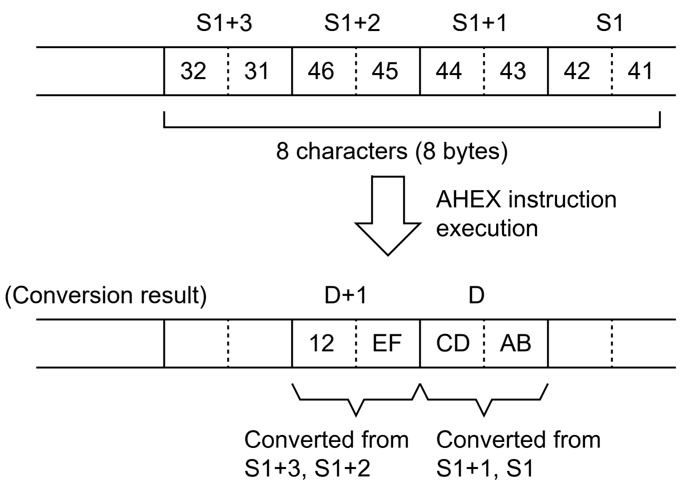


Conversion example

- ASCII codes are converted into hexadecimal data as shown below.

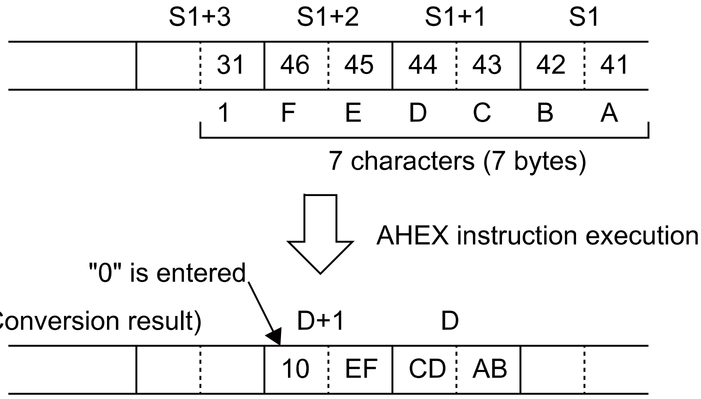
Conversion of eight characters (S2 = K8)

[ASCII code]



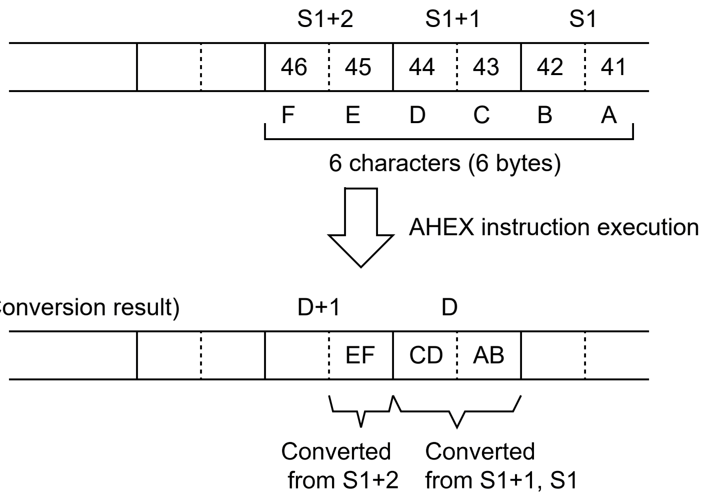
Conversion of seven characters (S2 = K7)

[ASCII code]



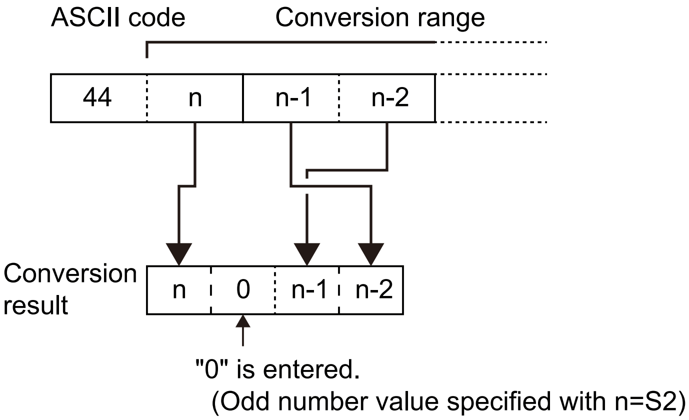
Conversion of six characters (S2 = K6)

[ASCII code]



(Note 1) In the conversion results, only the data for the low byte is stored in the D+1 word. The data for the high byte is left as it is and does not change.

- The conversion results are stored in byte units. If an odd number of characters is being converted, bits 0 to 3 of the final data (byte) of the conversion results will be filled with"0".



■ Reference: ASCII code

		Higher	
		3	**4**
Lower	0	0	@
	1	1	A
	2	2	B
	3	3	C
	4	4	D
	5	5	E
	6	6	F
	7	7	G
	8	8	H
	9	9	I

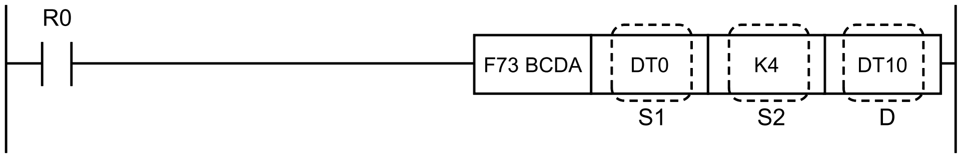
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the conversion range of the number of bytes specified by [S2] exceeds the area
	Turns ON when the conversion result exceeds the area
	Turns ON when the [S2] specification is"0"
	Turns ON when there is a character code other than 0 to F in the ASCII codes specified by [S1]

15.4 F73 BCDA (BCD Data to ASCII Code Conversion)

Converts up to eight digits of BCD data to ASCII code character strings.

■ Instruction format



■ Operands

Items	Settings
S1	Starting number of the area storing the BCD numerical value
S2	Area storing data indicating the amount and direction of data to be converted, or constant data
D	Starting number of the area storing the ASCII code of conversion result

■ Devices that can be specified (indicated by •)

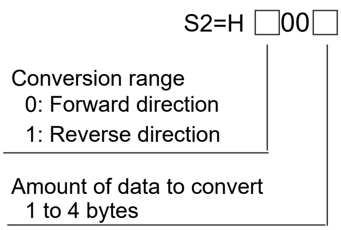
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•		•	•					•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•								•	

■ Outline of operation

- The BCD data stored in the area specified by [S1] is converted to ASCII code and stored in the area specified by [D]. Up to four bytes (8 digits) can be converted.
- The amount (number of bytes) of BCD data to be converted and the conversion direction is specified by [S2].
- The amount of the conversion result (ASCII code) is twice the converted data.

■ Setting the conversion data amount and conversion direction [S2]

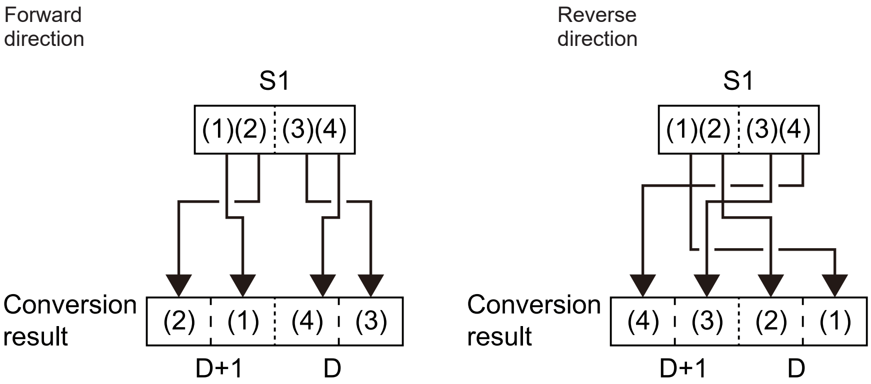
Specify a 4-digit BCD (H constant) according to the following format.



- 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.
- Refer to the example for a description of the conversion direction.

■ Precautions for programming

- The two characters that make up one byte are interchanged when stored.
- Converts two bytes as one section.



<Example>

When internal relay R0 turns ON, the BCD data stored in data register DT0 is converted to ASCII code and stored in DT10.

1. When S2 = H2 (forward direction, 2-byte conversion)

BCD data (DT0)
H 1234
⇩
ASCII code (DT11, DT10)
H $\underbrace{\underline{32}\,\underline{31}}_{\text{DT11}}\ \underbrace{\underline{34}\,\underline{33}}_{\text{DT10}}$
2 1 4 3

2. When S2 = H1002 (reverse direction, 2-byte conversion)

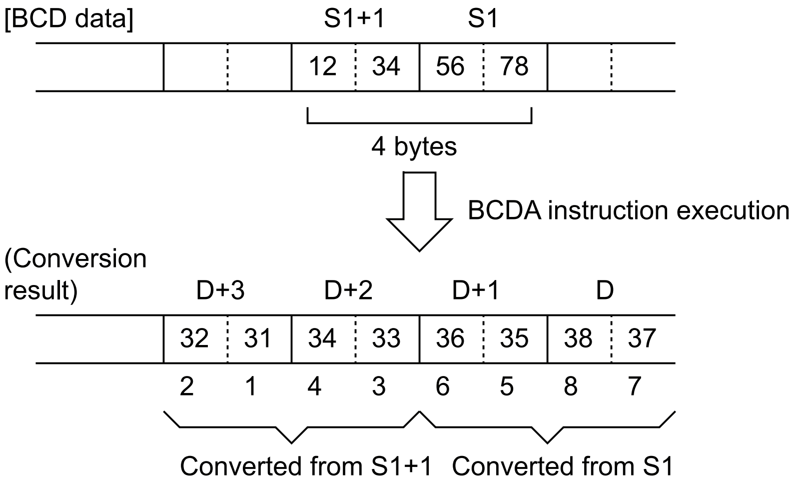
BCD data (DT0)
H 1234
⇩
ASCII code (DT11, DT10)
H $\underbrace{\underline{34}\,\underline{33}}_{\text{DT11}}\ \underbrace{\underline{32}\,\underline{31}}_{\text{DT10}}$
4 3 2 1

■ Conversion example

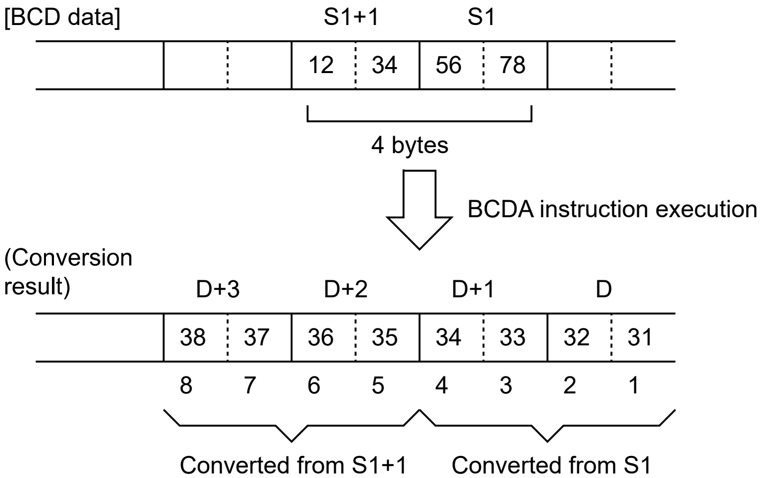
For the above program

The conversion from BCD data to ASCII code is performed as shown below.

Forward conversion of four bytes (S2 = H0004)



Reverse conversion of four bytes (S2 = H1004)



■ Reference: ASCII code

		Higher
		3
Lower	0	0
	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7

		Higher
		3
	8	8
	9	9

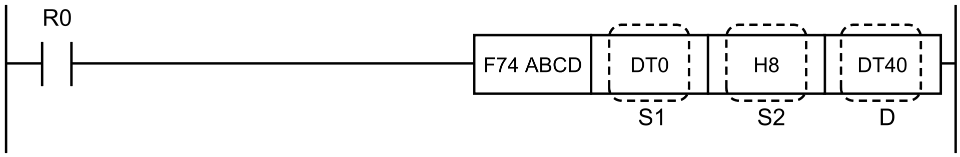
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when there is data other than BCD in the data starting with [S1]
	Turns ON when the number of bytes specified by [S2] exceeds the area of [S1]
	Turns ON when the conversion result exceeds the area
	Turns ON when the number of bytes specified by [S2] is"0"
	Turns ON when the number of bytes specified by [S2] is greater than four

15.5 F74 ABCD (ASCII Code to BCD Data Conversion)

Converts an ASCII character string to 4-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S1	Starting number of the area storing the ASCII code
S2	Area storing data indicating the number of ASCII codes and direction of data to be converted, or constant data
D	Number of the start of the area storing the BCD value that is the result of conversion

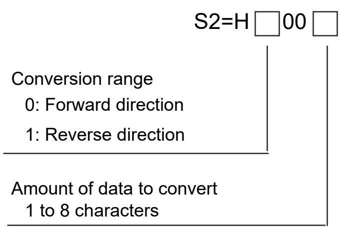
■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•		•	•					•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•								•	

■ Outline of operation

- The ASCII codes that are stored in the area starting from the number specified by [S1] are converted into BCD data and stored in the area starting from the number specified by [D]. A maximum of eight characters can be converted.
- The number of ASCII codes (number of characters) to be converted and the conversion direction are specified by [S2].
- The conversion result (BCD data) is half the volume of the converted ASCII code strings.

■ Specification of number of characters to be converted and conversion direction [S2]

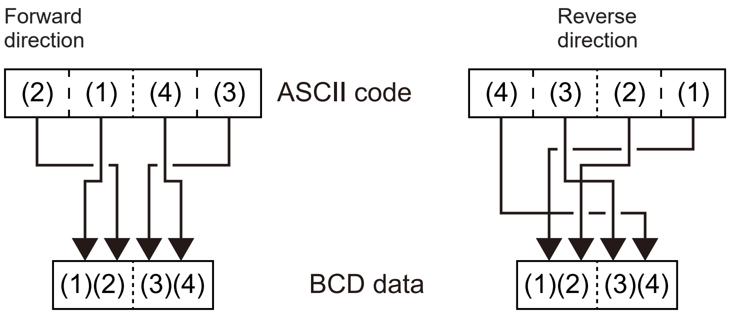


■ Precautions for programming

- Two ASCII code characters are converted into 1-byte numeric values (two digits). At this time, the upper and lower characters are interchanged.
- Four characters are stored as one segment of data.
- The conversion results are stored in byte units. If an odd number of characters is being converted, the conversion result is as follows.

  i) Bits 0 to 3 of the final data are filled with"0". (In the forward direction)

  ii) Bits 4 to 7 of the final data are filled with"0". (In the reverse direction)



<Example>

When internal relay R0 turns ON, the ASCII codes stored in data registers starting from DT0 are converted to BCD numeric data and stored in DT40.

1. When S2 = H4 (forward direction, 4-byte conversion)

   ASCII code (DT1, DT0)

   H $\underbrace{\underset{4}{\underline{34}}\ \underset{3}{\underline{33}}}_{DT1}\ \underbrace{\underset{2}{\underline{32}}\ \underset{1}{\underline{31}}}_{DT0}$

   ⇩

   BCD data (DT40)

   H 3412

2. When S2 = H1004 (reverse direction, 4-byte conversion)

   ASCII code (DT1, DT0)

   H $\underbrace{\underset{4}{\underline{34}}\ \underset{3}{\underline{33}}}_{DT1}\ \underbrace{\underset{2}{\underline{32}}\ \underset{1}{\underline{31}}}_{DT0}$

   ⇩

   BCD data (DT40)

   H 1234

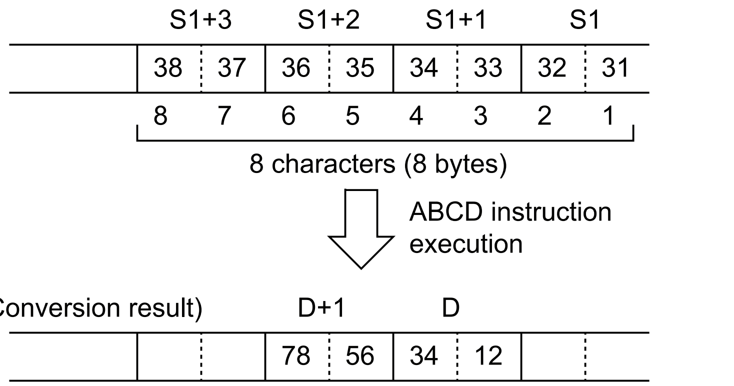
■ Conversion example

For the above program

ASCII codes are converted into BCD data as shown below.

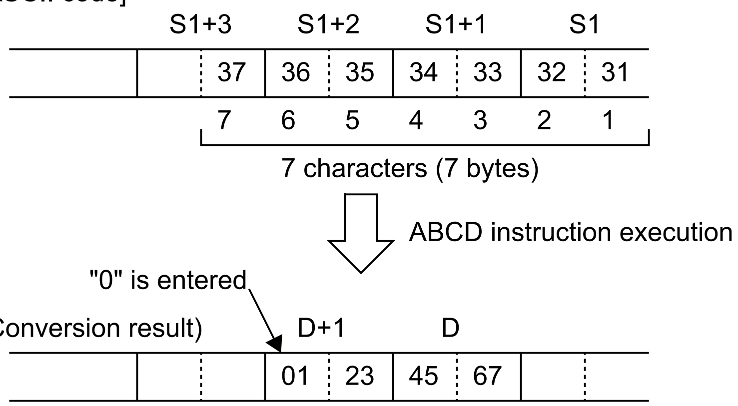
Conversion of eight characters (S2 = H0008)

[ASCII code]



Conversion of seven characters (S2 = H1007)

[ASCII code]



■ Reference: ASCII code

		Higher
		3
Lower	0	0
	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9

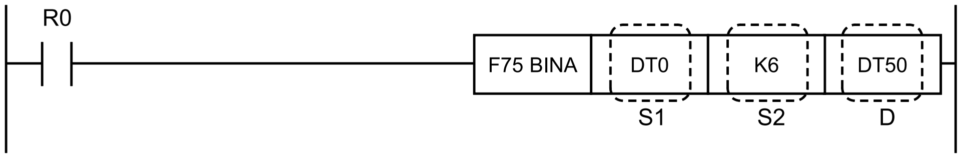
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when there is a character code other than 0 to 9 in the ASCII codes specified by [S1]
	Turns ON when the number of characters specified by [S2] exceeds the area of [S1]
	Turns ON when the conversion result exceeds the area
	Turns ON when the number of characters specified by [S2] is"0"
	Turns ON when the number of characters specified by [S2] is greater than 8

15.6 F75 BINA (16-bit Binary Data to ASCII Code Conversion)

Converts 16-bit BIN data expressing a decimal number to an ASCII code character string.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the hexadecimal data or constant data
S2	Area storing the number of bytes of the area storing the conversion results, or constant data
D	Starting number of the area storing the ASCII code of conversion result

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•								•	

■ Outline of operation

- The 16-bit data expressing a decimal number specified by [S1] is converted to ASCII code. The ASCII code is stored in the area specified by [D]. The start of the storage area is specified by [D] and its size is specified by [S2].
- Specify the number of bytes in [S2] as a decimal number. (This specification cannot be made with BCD data.)

■ Operation example

Operation of instruction format description program

When internal relay R0 is ON, the 16-bit data (expressing a decimal number) stored in data register DT0 is converted to ASCII code and stored in DT50 to DT52 (six bytes).

DT0: K-100（H FF9C）

⇩

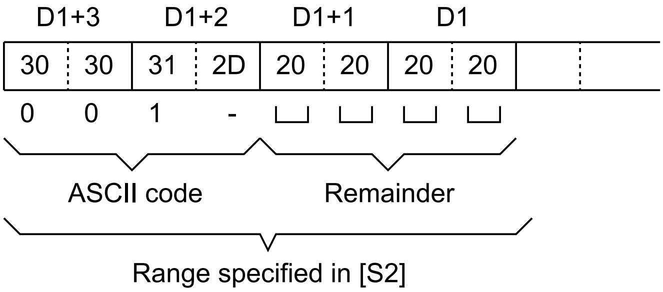
DT52～DT50: H 30 30 31 2D 20 20

0 0 1 -

DT52 DT51 DT50

■ Precautions for programming

- If the conversion target is a positive number, a sign code (+) is not added in front of the numeric data.
- If the conversion target is a negative number, a sign code (–: H2D) is added in front of the numeric data.
- Any remaining storage area is filled with spaces (H20).
- The position of the ASCII code may change depending on the size of the storage area as data is filled in the direction of the final address.

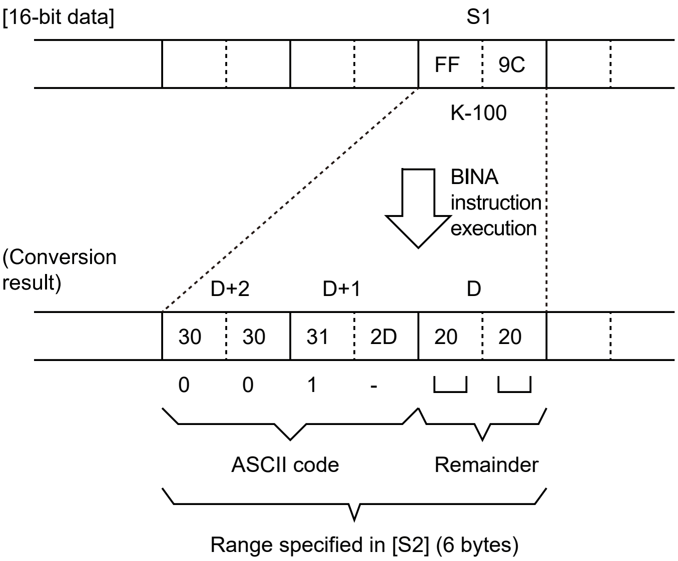


- An operation error occurs if the number of bytes of ASCII codes following conversion (including the minus sign) is larger than the number of bytes specified by S2. When specifying S2, make sure the number of digits to be converted including the sign is taken into consideration.

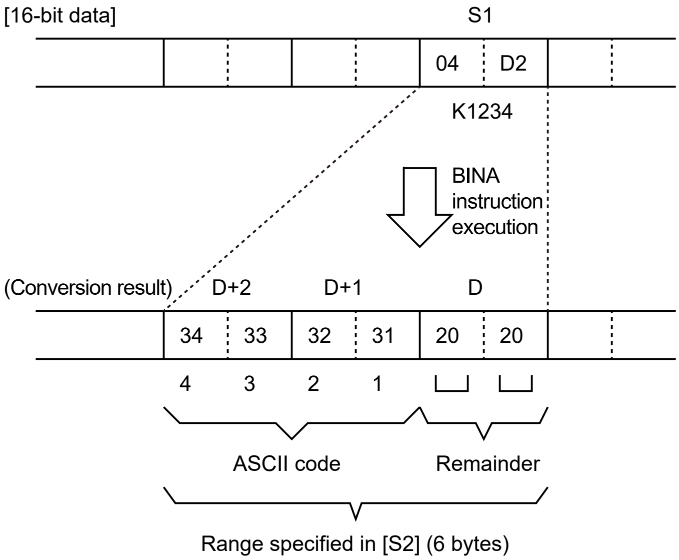
■ Conversion example

The conversion from a 16-bit decimal number to ASCII code is performed as follows.

When converting a negative number



When converting a positive number



■ Reference: ASCII code

		Higher
		3
Lower	0	0
	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9

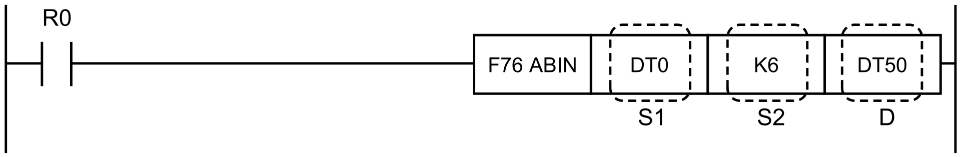
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the number of bytes specified by [S2] exceeds the area specified by [D]
	Turns ON when the number of bytes specified by [S2] is"0"
	Turns ON when the conversion result exceeds the area
	Turns ON when the number of bytes of the conversion result exceeds the number of bytes specified by [S2]

15.7 F76 ABIN (ASCII Code to 16-bit Binary Data Conversion)

Converts an ASCII code character string expressing a decimal number to 16-bit BIN data.

■ Instruction format



■ Operands

Items	Settings
S1	Starting number of the area storing the ASCII code to be converted
S2	Area storing the number of bytes of data to be converted, or constant data
D	Area to store the conversion result

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●		●	●					●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●								●	

■ Outline of operation

- The ASCII code expressing a decimal value of the number of bytes (number of characters) specified by [S2] starting from the area specified by [S1] is converted to a decimal value (16-bit K constant). The decimal value is stored in the area specified by [D].
- Specify the number of bytes in [S2] as a decimal number. (This specification cannot be made with BCD data.)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the ASCII code stored in data registers DT0 to DT2 (6 bytes) is converted to a decimal number (16-bit data), and stored in DT50.

ASCII code (DT2 to DT0)
H 30 30 31 2D 30 30
0 0 1 -
DT2 DT1 DT0

↓

16-bit data (DT50)
K-100

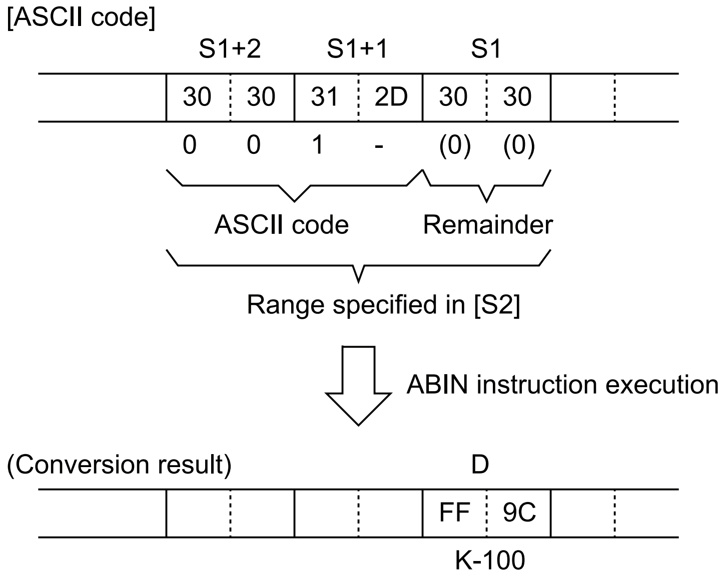
■ Precautions for programming

- Store the ASCII code for conversion in the direction of the final address of the specified area.
- Fill the remaining bytes with"0"(H30) or spaces (H20).
- Signed ASCII codes (+: H2B, -: H2D) are also converted. The + sign can be omitted.

■ Conversion example

Conversion of ASCII code to a 16-bit decimal number is performed as shown below.

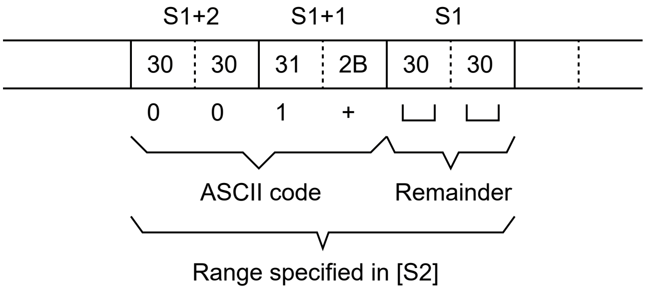
Example of conversion of an ASCII code expressing a negative number



Example of conversion of an ASCII code expressing a positive number

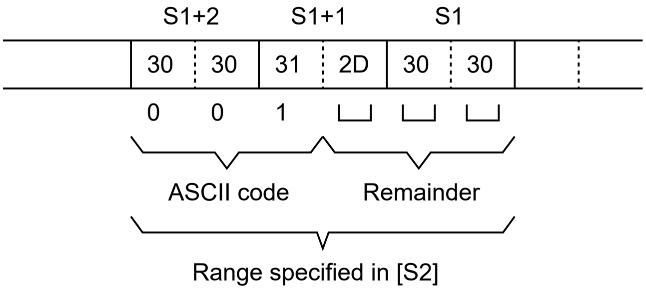
Example (1)

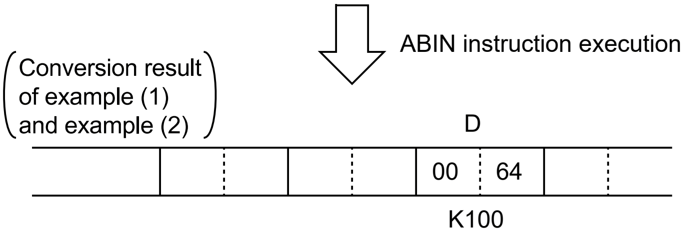
[ASCII code]



Example (2)

[ASCII code]





■ Reference: ASCII code

		Higher
		3
Lower	0	0
	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8

		Higher
		3
	9	9

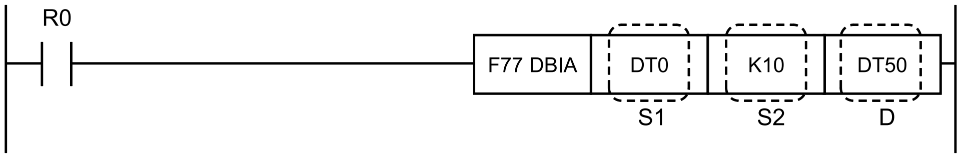
■ Flag operations

Name	**Description**
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the number of bytes specified by [S2] exceeds the area of [S1]
	Turns ON when the number of bytes specified by [S2] is"0"
	Turns ON when the conversion result exceeds the area
	Turns ON when the conversion result exceeds 16 bits of data
	Turns ON when an ASCII code containing characters other than the numbers 0 to 9, signed code, or spaces is specified for [S1]

15.8 F77 DBIA (32-bit Binary Data to ASCII Code Conversion)

Converts 32-bit BIN data expressing a decimal number to an ASCII code character string.

■ Instruction format



■ Operands

Items	Settings
S1	Starting number of the area storing 32-bit data, or constant data
S2	Area storing the number of bytes of the area storing the conversion results, or constant data
D	Starting number of the area storing the ASCII code of conversion result

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●								●	

■ Outline of operation

- The 32-bit data expressing a decimal number specified by [S1] is converted to ASCII code. The ASCII code is stored in the area starting with the area specified by [D]. The start of the storage area is specified by [D] and the number of bytes is specified by [S2].
- Specify the number of bytes in [S2] as a decimal number (K constant).

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the 32-bit data stored in data registers DT0 and DT1 is converted to ASCII code expressing a decimal number and stored in DT50 to DT54 (10 bytes).

32-bit data (DT0, DT1)
K12345678

⇩

ASCII codes DT50 to DT54:
H 38 37 36 35 34 33 32 31 20 20
8 7 6 5 4 3 2 1
DT54 DT53 DT52 DT51 DT50

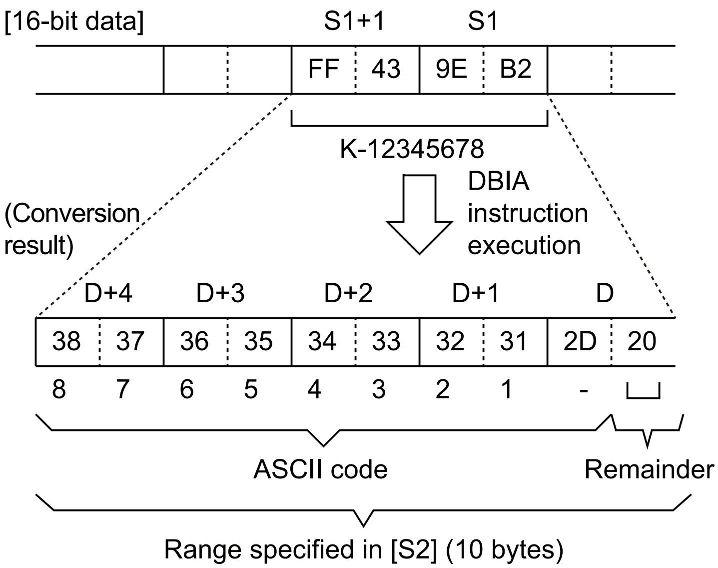
■ Precautions for programming

- If the conversion target is a positive number, a sign code (+) is not added in front of the numeric data.
- If the conversion target is a negative number, a sign code (–: H2D) is added in front of the numeric data.
- Any remaining storage area is filled with spaces (H20).
- The position of the ASCII code may change depending on the size of the storage area as data is filled in the direction of the final address.
- An operation error occurs if the number of bytes of ASCII codes following conversion (including the minus sign) is larger than the number of bytes specified by S2. When specifying S2, make sure the number of digits to be converted including the sign is taken into consideration.

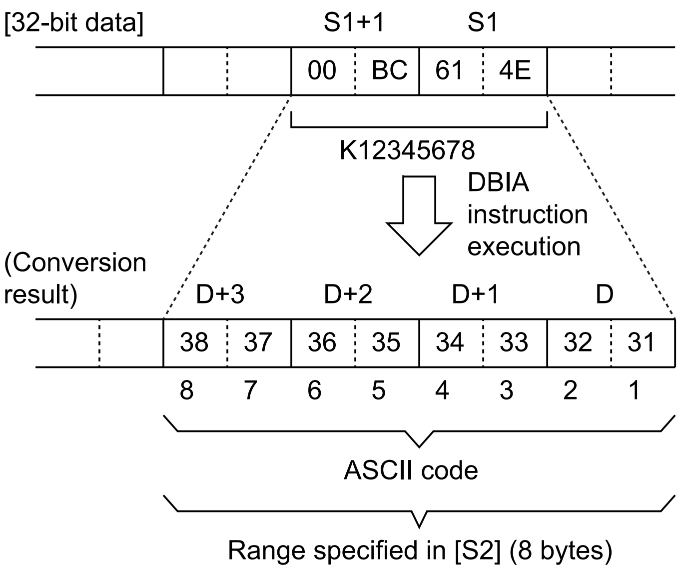
■ Conversion example

The following shows conversion of a 32-bit decimal number to ASCII codes.

When converting a negative number



When converting a positive number



■ Reference: ASCII code

		Higher
		3
Lower	0	0
	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9

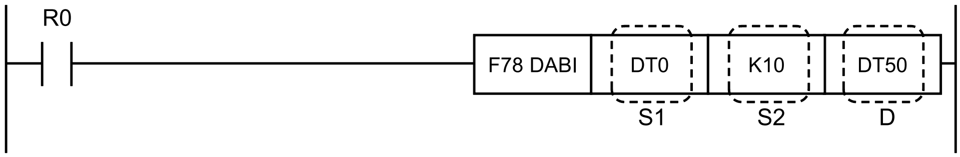
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the number of bytes specified by [S2] exceeds the area specified by [D]
	Turns ON when the number of bytes specified by [S2] is"0"
	Turns ON when the conversion result exceeds the area
	Turns ON when the number of bytes of the conversion result exceeds the number of bytes specified by [S2]

15.9 F78 DABI (ASCII Code to 32-bit Binary Data Conversion)

Converts an ASCII code character string expressing a decimal number to 32-bit BIN data.

■ Instruction format



■ Operands

Items	Settings
S1	Starting number of the area storing the ASCII code to be converted
S2	Area storing the numerical values (number of bytes = number of characters) representing the range to be converted, or constant data
D	Number of the start of the area storing the conversion result

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•		•	•					•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•								•	

■ Outline of operation

- The ASCII code string expressing a decimal value of the number of bytes (number of characters) specified by [S2] starting from the area specified by [S1] is converted to a decimal value (32-bit K constant). The decimal value is stored in two words starting from the area specified by [D].
- Specify the number of bytes in [S2] as a decimal number.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the ASCII codes stored in data registers DT0 to DT4 (10 bytes) are converted to decimal numbers, and stored in DT50 and DT51.

ASCII code (DT0 to DT4)

H 38 37 36 35 34 33 32 31 20 20

8 7 6 5 4 3 2 1

DT4 DT3 DT2 DT1 DT0

⇩

32-bit data (DT50, DT51)

K 12345678

■ Precautions for programming

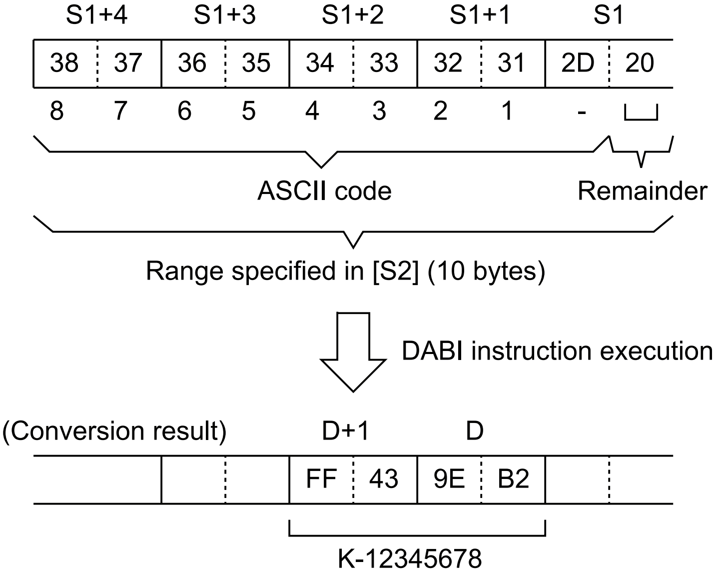
- Store the ASCII code for conversion in the direction of the final address of the specified area.
- Fill the remaining bytes with"0"(H30) or spaces (H20).
- Signed ASCII codes (+: H2B, -: H2D) are also converted. The + sign can be omitted.

■ Conversion example

Conversion of ASCII code to a 32-bit decimal number is performed as shown below.

Example of conversion of an ASCII code string expressing a negative number

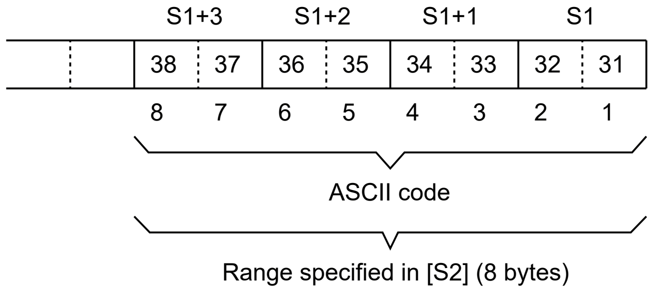
[ASCII code]



Example of conversion of an ASCII code expressing a positive number

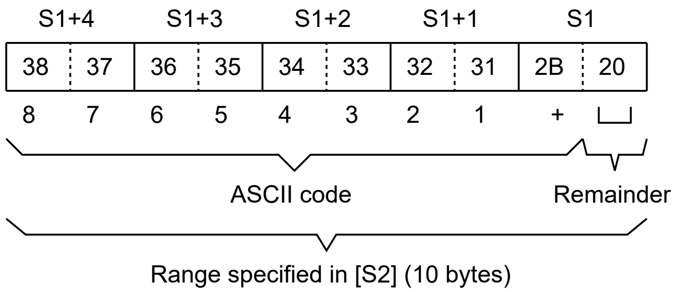
Example (1)

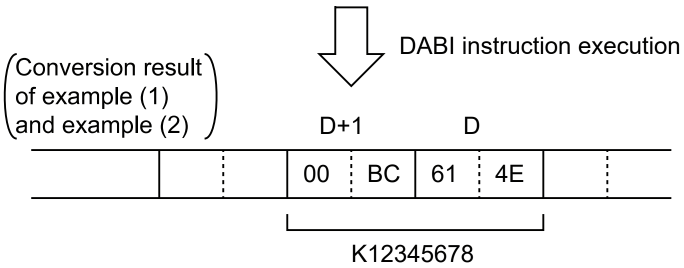
[ASCII code]



Example (2)

[ASCII code]





■ Reference: ASCII code

		Higher
		3
Lower	0	0
	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7

		Higher
		3
	8	8
	9	9

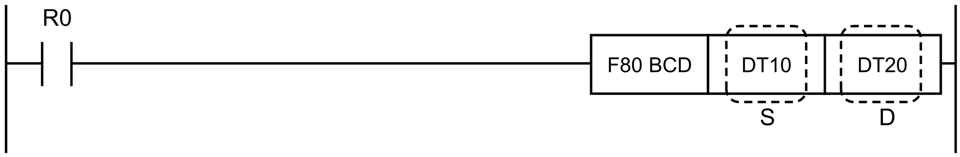
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the number of bytes specified by [S2] exceeds the area of [S1]
	Turns ON when the number of bytes specified by [S2] is"0"
	Turns ON when the conversion result exceeds the area
	Turns ON when the conversion result exceeds 32 bits of data
	Turns ON when an ASCII code containing characters other than the numbers 0 to 9, signed code, or spaces is specified for [S1]

15.10 F80 BCD (16-bit Binary Data to BCD Data Conversion)

Converts 16-bit binary data to 4-digit BCD.

■ Instruction format



■ Operands

Items	Settings
S	Target data: Area storing 16-bit data, or constant data
D	Storage destination: Area storing 4-digit BCD data following conversion

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

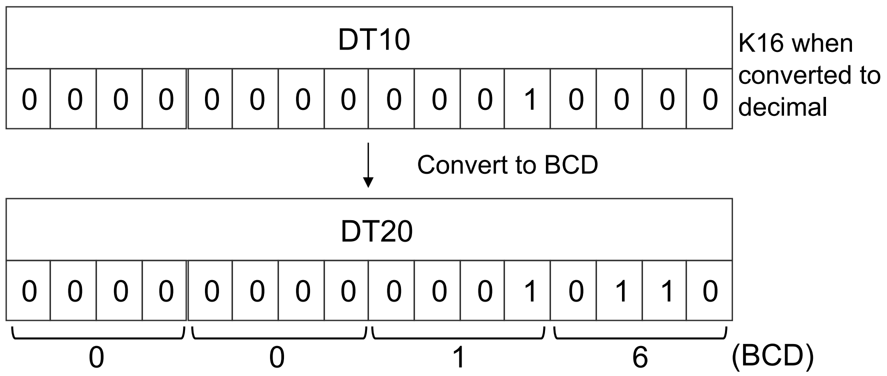
- The 16-bit data expressing a decimal number specified by [S] is converted to 4-digit BCD data and stored in the area specified by [D].

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the contents of data register DT10 are converted to 4-digit BCD data and stored in data register DT20.

If DT10 is converted decimal number 16, the following will be stored in DT20.



Precautions for programming

- The maximum value of 16-bit data that can be converted is K9999 (H270F).

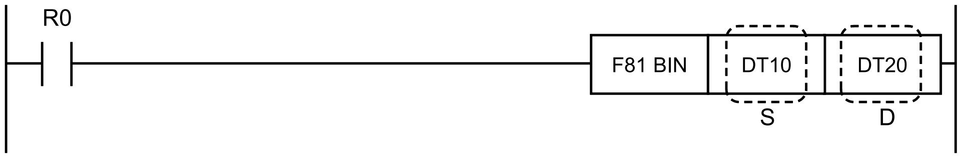
Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the binary data exceeds the range that can be converted to BCD (when negative, or over K9999)

15.11 F81 BIN (BCD Data to 16-bit Binary Data Conversion)

Converts 4-digit BCD data to 16-bit binary data.

■ Instruction format



■ Operands

Items	Settings
S	Target data: Area storing 4-digit BCD data, or constant data
D	Storage destination: Area storing converted binary data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

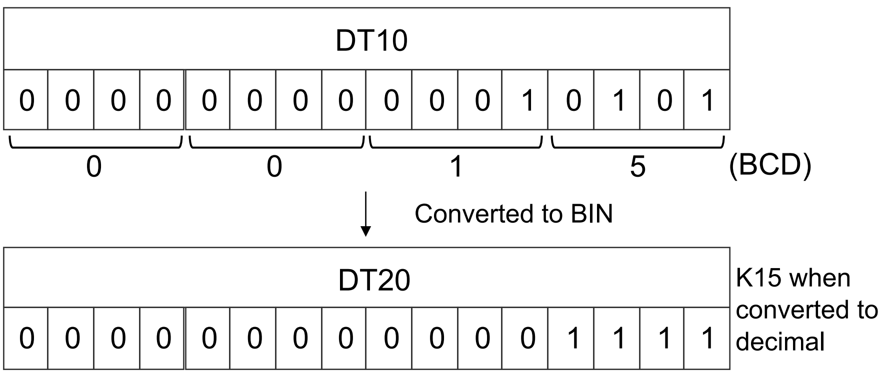
■ Outline of operation

The 4-digit BCD data specified by [S] is converted to 16-bit data expressing a decimal number and stored in the area specified by [D].

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of data register DT10 is converted to 16-bit data expressing a decimal number and stored in data register DT20. If DT10 is BCD data consisting of H15, the following will be stored in DT20.



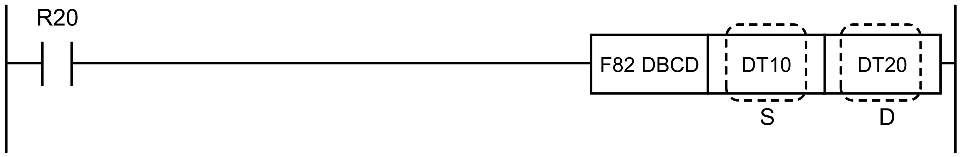
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON if [S] is not BCD data

15.12 F82 DBCD (32-bit Binary Data to BCD Data Conversion)

Converts 32-bit binary data to 8-digit BCD data.

■ Instruction format



■ Operands

Items	Settings
S	Target data: Area storing 32-bit data, or constant data
D	Storage destination: Area storing 8-digit BCD data following conversion

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

The 32-bit data specified by [S] expressing a decimal number is converted to 8-digit BCD data and stored in the area specified by [D].

■ Operation example

Operation of instruction format description program

When internal relay R20 turns ON, the content of data registers DT10 and DT11 is converted to 8-digit BCD data, and stored in DT21 and DT22.

■ Precautions for programming

The maximum value of binary data that can be converted is K99999999 (H5F5E0FF).

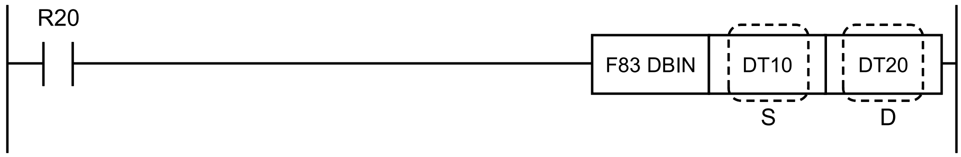
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	When the binary data exceeds the range that can be converted to BCD data (when the value is negative or exceeds K99999999)

15.13 F83 DBIN (BCD Data to 32-bit Binary Data Conversion)

Converts 8-digit BCD data to 32-bit binary data.

■ Instruction format



■ Operands

Items	Settings
S	Target data: Area storing 8-digit BCD data, or constant data
D	Storage destination: Area storing converted binary data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

The 8-digit BCD data specified by [S] is converted to 32-bit data expressing a decimal number and stored in the area specified by [D].

■ Operation example

Operation of instruction format description program

When internal relay R20 turns ON, the value expressing the 8-digit BCD data in data registers DT10 and DT11 is converted to 32-bit data (K constant) and stored in DT20 and DT21.

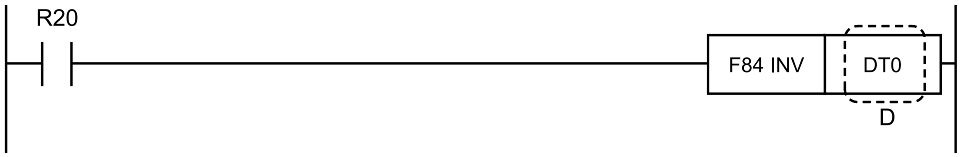
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON if [S] is not BCD data

15.14 F84 INV (16-bit Data Invert)

Inverts 16-bit data.

■ Instruction format



■ Operands

Items	Settings
D	Area that stores the data to invert

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- Inverts 1 (ON) and 0 (OFF) of each bit of the 16-bit data specified by [D].
- This instruction can be used to output to 7-segment display that uses negative logic operation.

■ Operation example

Operation of instruction format description program

Inverts the contents of data register DT0 when internal relay R20 turns ON.

	15	·	·	12	11	·	·	8	7	·	·	4	3	·	·	0 (bit)
DT0:	0	1	0	1	1	1	1	0	1	0	1	1	1	1	0	1

⇩ R20: ON (inversion)

DT0:	1	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0

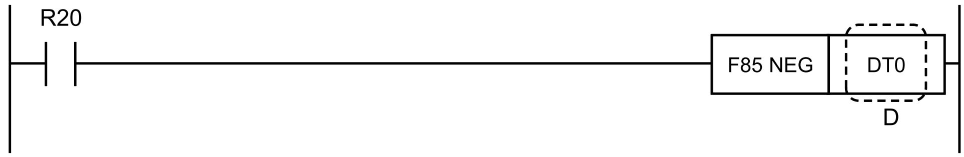
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

15.15 F85 NEG (16-bit Data Sign Inversion)

Takes complement of 2 in hexadecimal data.

■ Instruction format



■ Operands

Items	Settings
D	Area for storing original data and its complement of 2

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		●	●	●	●	●	●	●	●							●	

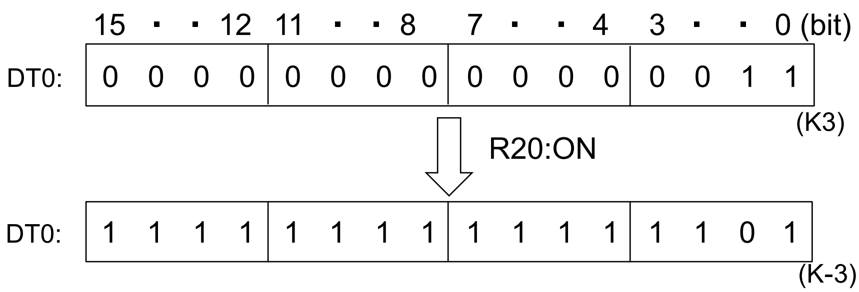
■ Outline of operation

- Inverts the content of hexadecimal data specified by [D] and adds +1 (takes complement of 2).
- Useful for inverting the signs of 16-bit data.

■ Operation example

Operation of instruction format description program

Inverts the content of data register DT0 and adds +1 when internal relay R20 turns ON.



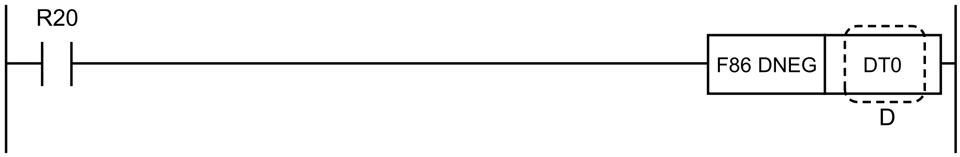
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

15.16 F86 DNEG (32-bit Data Sign Inversion)

Takes complement of 2 in 32-bit data.

■ Instruction format



■ Operands

Items	Settings
D	Starting number of area for storing original data and its complement of 2

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- Inverts the content of 32-bit data specified by [D] and [D+1] and adds +1.
- Useful for inverting the signs of 32-bit data.

■ Operation example

Operation of instruction format description program

Inverts the 32-bit content of DT0 and DT1 and adds +1 when internal relay R20 turns ON.

	15	·	·	12	11	·	·	8	7	·	·	4	3	·	·	0 (bit)
DT0:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
DT1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(K-3)

⇩ R20:ON

DT0:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
DT1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(K3)

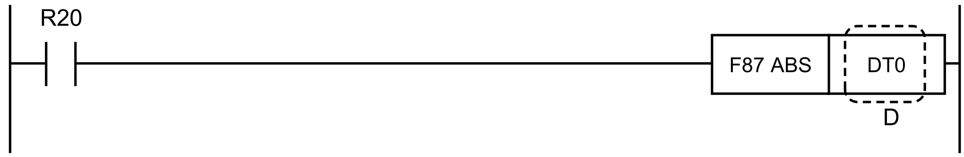
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

15.17 F87 ABS (Absolute Value of 16-bit Data)

Calculates the absolute value of signed 16-bit data.

■ Instruction format



■ Operands

Items	Settings
D	Area storing the data for which the absolute value will be calculated

■ Devices that can be specified (indicated by •)

Operand s	WX	WY	WR	WL	SV	EV	DT	LD	I	SW R	SD T	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The absolute value of the signed 16-bit data specified by [D] is calculated and stores in [D].
- This is effective for processing data in which the polarity (+ or -) changes.

■ Operation example

Operation of instruction format description program

When internal relay R20 turns ON, the absolute value of the value of data register DT0 is calculated. For instance, regardless of whether the value of DT0 is K1 or K-1, it will be K1 when this instruction is executed.

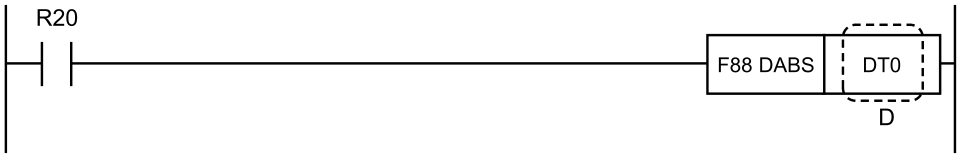
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the minimum value is negative (H8000)
R9009 (CY)	Turns ON when the value is negative (other than the minimum)

15.18 F88 DABS (Absolute Value of 32-bit Data)

Calculates the absolute value of signed 32-bit data.

■ Instruction format



■ Operands

Items	Settings
D	Starting number of the area storing the data for which the absolute value will be calculated

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The absolute value of the signed 32-bit data stored in [D] and [D+1] is calculated and stored in [D] and [D+1].
- This is effective for processing data in which the polarity (+ or -) changes.

■ Operation example

Operation of instruction format description program

When internal relay R20 turns ON, the absolute value of the signed 32-bit data in DT0 and DT1 is calculated and stored in DT0 and DT1.

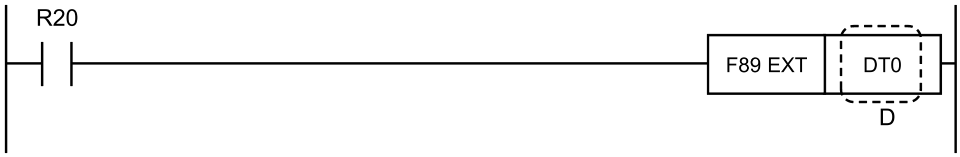
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification. Turns ON when the minimum value is negative (H80000000)
R9009 (CY)	Turns ON when the value is negative (other than the minimum)

15.19 F89 EXT (Sign Extension)

Extends 16-bit data to 32-bit data without changing signs or values.

■ Instruction format



■ Operands

Items	Settings
D	Area where data for sign extension is stored

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		●	●	●	●	●	●	●	●							●	

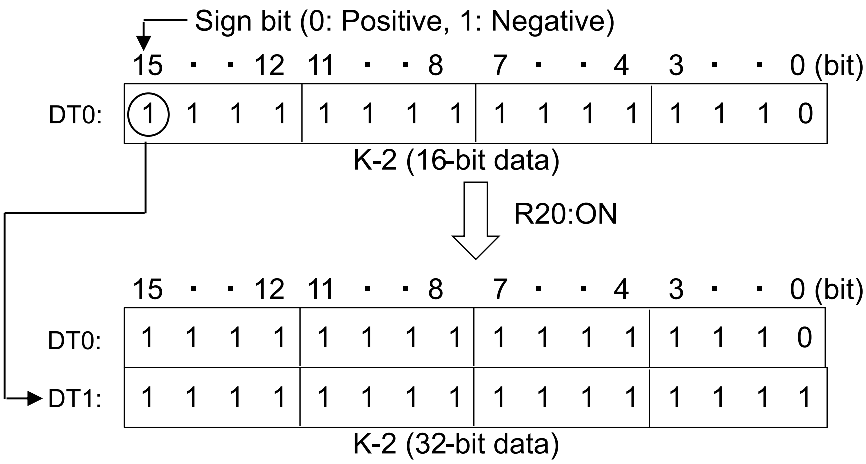
■ Outline of operation

- Converts 16-bit data to 32-bit data without changing its signs or values.
- If the sign bit (bit 15) of the 16-bit data specified in [D] is 0, all 16 bits in the area following [D] become 0. If the sign bit is 1, all 16 bits become 1. Thus, 16-bit data is converted to 32-bit data without its signs or values being changed.
- After execution of the F89 EXT instruction, double word data starting at [D] can be used as an operand for a 32-bit operation instruction.

■ Operation example

Operation of instruction format description program

When the internal relay R20 is ON, all 16 bits of DT1 are filled with the content of bit 15 of the data in DT0. If K-2 is stored in DT0, the data will be as follows.



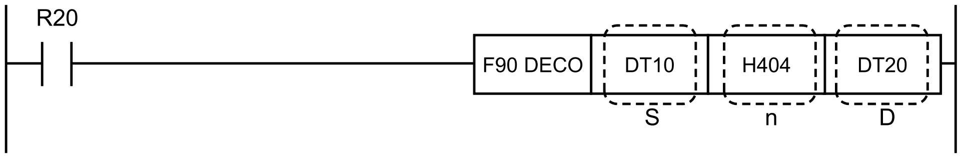
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

15.20 F90 DECO (Decode)

Decodes the specified data.

■ Instruction format



■ Operands

Items	Settings
S	Area storing conversion data, or constant data
n	Area storing the control data, or constant data
D	Starting address of the area storing the conversion result

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
n	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

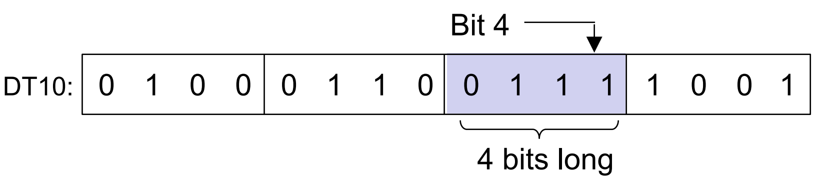
- The part of the data specified by [S] is decoded and the decoded result is stored in the area specified by [D].
- The part to be decoded is specified by control data [n].
- The length of the area required to store the decoded result depends on the length of the data to be decoded.

■ Operation example

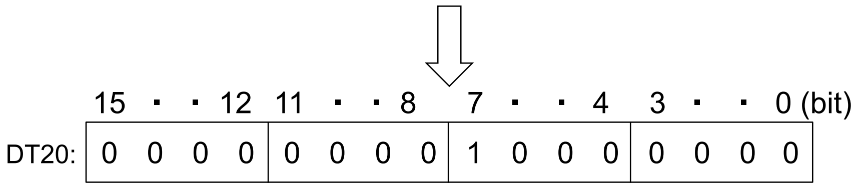
Operation of instruction format description program

When internal relay R20 turns ON, the part of data register DT10 specified by [n] = H404 (H constant) is decoded and the result is stored in data register DT20.

e.g. When the value (control data) of [n] is H404



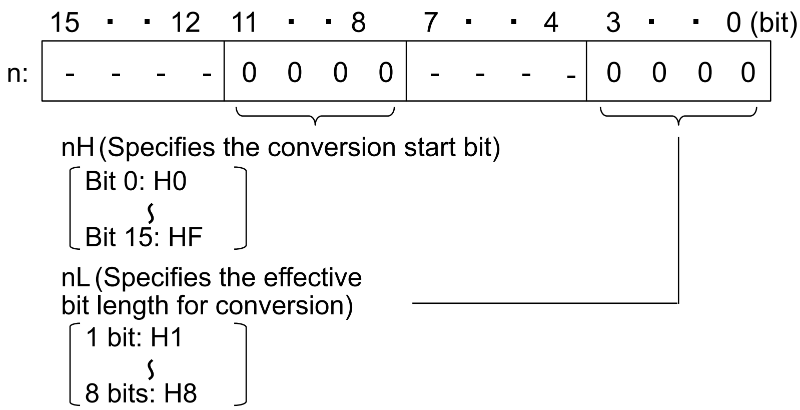
The decoded result for the specified part ("0111"= 7) is stored in the 2^4bit area starting from DT20.



Bit 7 of the 2^4bit area starting from DT20 is turned ON, and the other bits are set to 0.

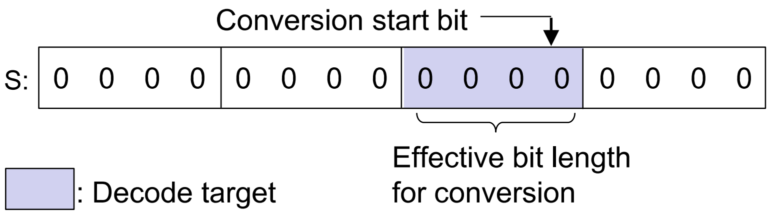
■ Specifying the data to be decoded (control data [n])

Specify the conversion start bit and conversion effective bit length.



The effective bit length of the decoded result is 2^{nL}bits.
See the table below for the effective bit length and occupied length of the result.

e.g. When control data [n] is H0404 and the data to be decoded is the 4 bits from bit 4 in the area specified by [S].



Specification of nL and length of result

Effective bit length for <nL value> conversion	Occupation length of decoded result	Effective bit length of decoded result	Value other than effective bit length in D
1	1 word	2-bit	0
2	1 word	4-bit	0
3	1 word	8 bits	0
4	1 word	16 bits	-

Effective bit length for <nL value> conversion	Occupation length of decoded result	Effective bit length of decoded result	Value other than effective bit length in D
5	2 words	32 bits	-
6	4 words	64 bits	-
7	8 words	128 bits	-
8	16 words	256 bits	-

Conversion example

When decoding 4-bit data (nL = 4), the contents of the conversion data and the decoded result are as follows.

Conversion data	Decoded result
0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 1 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 1 1	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 1 0 0	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 1 0 1	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 1 1 0	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 1 1 1	0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
1 0 0 0	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 0 0 1	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
1 0 1 0	0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
1 0 1 1	0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
1 1 0 0	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 1	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

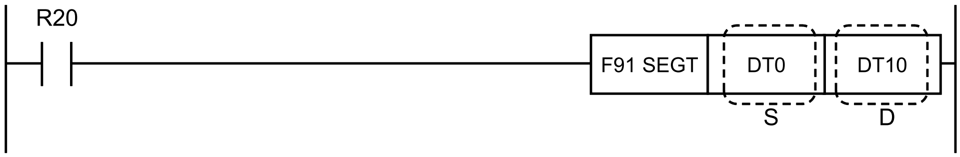
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the effective bit length for conversion (nL) is not 1 ≤ nL ≤ 8
	Turns ON (integrity) when the conversion start bit No. (nH) and conversion effective bit length (nL) are not 1 ≤ (nH + nL) ≤ 16
	Turns ON when the decoded result exceeds the area specified by [D] when stored

15.21 F91 SEGT (7-segment)

Converts specified 16-bit data to 4-digit data for 7-segment display.

■ Instruction format



■ Operands

Items	Settings
S	Area storing conversion data, or constant data
D	Starting address of the area storing the conversion result

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

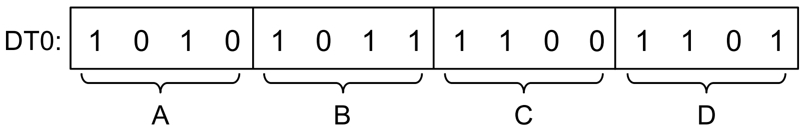
- Converts 16-bit data specified by [S] to four-digit data for 7-segment display, and stores this in the area starting from the two-word area specified by [D].
- Refer to the table below for the relationship between the displayed contents, the contents specified for [S], and the 7-segment display data.

■ Operation example

Operation of instruction format description program

Converts the contents of data register DT0 to 7-segment display data when internal relay R20 turns ON. The converted results are stored in data registers DT10 and DT11. For example, to display"ABCD", the following would be entered.

1. DT0 is set to H ABCD.



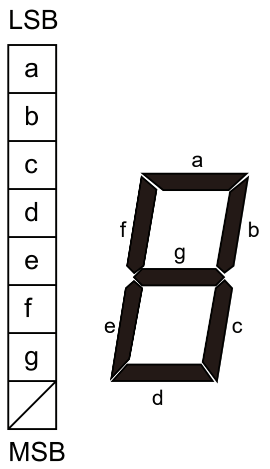
2. When the content of DT0 is converted to 7-segment display data, it is as follows.

	C								D							
DT10:	0	0	1	1	1	0	0	1	0	1	0	1	1	1	1	0

	A								B							
DT11:	0	1	1	1	0	1	1	1	0	1	1	1	1	1	0	0

■ Relationship between display content and data

Value	Conversion data 1 digit [S]				Data for 7-segment display 1 digit [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



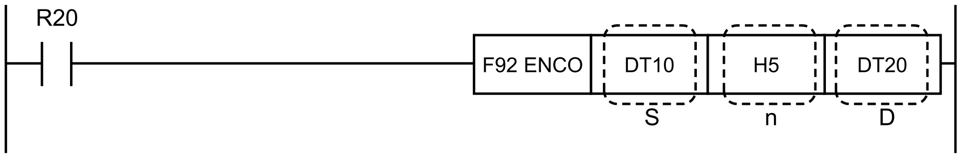
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the area is exceeded when conversion results are stored in the area specified by [D]

15.22 F92 ENCO (Encode)

Encodes the specified data.

■ Instruction format



■ Operands

Items	Settings
S	Starting address of the area storing conversion data
n	Area storing the control data, or constant data
D	Area to store the conversion result

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●		●	●					●	
n	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- Encodes a section of the data specified in [S], and stores the encoded result in the area specified in [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.
- The content of the 2^{nL}bits starting from the area specified in [S] are encoded. The encoded result is stored as a decimal, within the 8 bits starting from the bit specified in nH.
- Sections of the area specified in [D] that are not storing the conversion result will be 0.

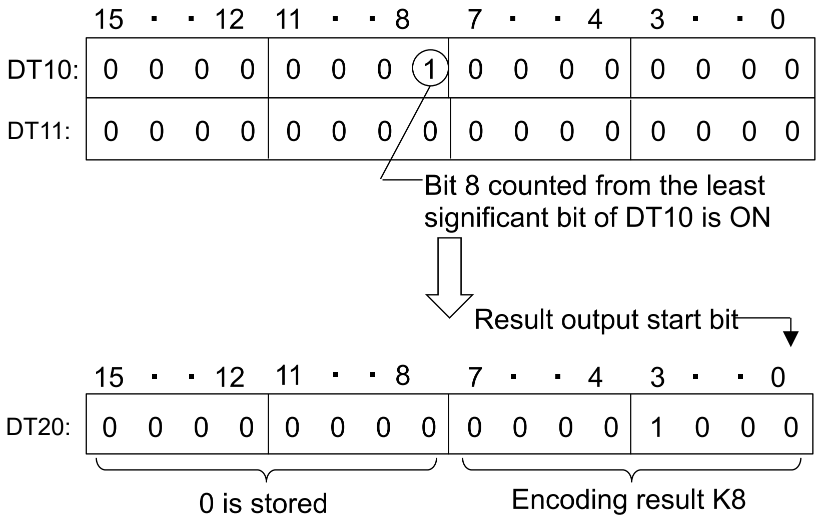
■ Operation example

Operation of instruction format description program

When the internal relay R20 is ON, the bit area (data register starting at DT10) specified in [n] = H5 (H constant) is encoded, and the result is stored in DT20.

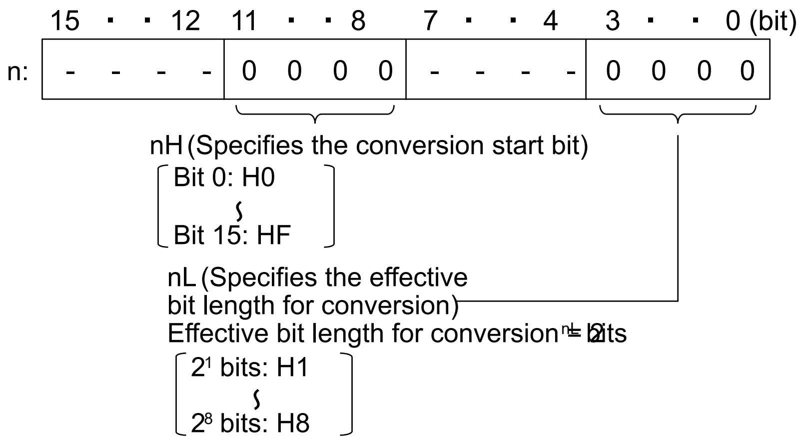
When the value of [n] (control data) is H5

The effective bits for conversion are the 32-bit section from DT10 (DT10 to DT11). The bit numbers that are ON in this two-word area are stored as decimals from bit 0 of DT20.



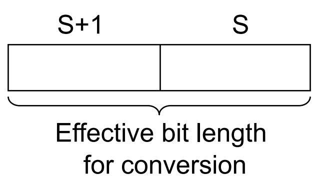
■ **Specifying the target to be encoded (control data [n])**

Specifies the effective bit length for conversion and the starting bit for output of the result.



e.g. When the control data [n] is H0005

The target to be encoded is the 2^5bits (32-bit = two words) starting from the area specified by [S].



The result is stored from bit 0 in the area specified by [D].

Specification of nL and length of result

Value of nL	Effective bit length for conversion
1	2-bit
2	4-bit
3	8-bit (one byte)
4	16-bit (one word)
5	32-bit (two words)
6	64-bit (four words)
7	128-bit (eight words)
8	256-bit (16 words)

Conversion example

When encoding 16-bit data (nL = 4), the content of the conversion data and the encoding result will be as follows.

Conversion data (16-bit)				Encoding result
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0 0 0 1
0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0	0 0 1 0
0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0	0 0 1 1
0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0	0 1 0 0
0 0 0 0	0 0 0 0	0 0 1 0	0 0 0 0	0 1 0 1
0 0 0 0	0 0 0 0	0 1 0 0	0 0 0 0	0 1 1 0
0 0 0 0	0 0 0 0	1 0 0 0	0 0 0 0	0 1 1 1
0 0 0 0	0 0 0 1	0 0 0 0	0 0 0 0	1 0 0 0
0 0 0 0	0 0 1 0	0 0 0 0	0 0 0 0	1 0 0 1
0 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0	1 0 1 0
0 0 0 0	1 0 0 0	0 0 0 0	0 0 0 0	1 0 1 1
0 0 0 1	0 0 0 0	0 0 0 0	0 0 0 0	1 1 0 0
0 0 1 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 0 1
0 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 0
1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1

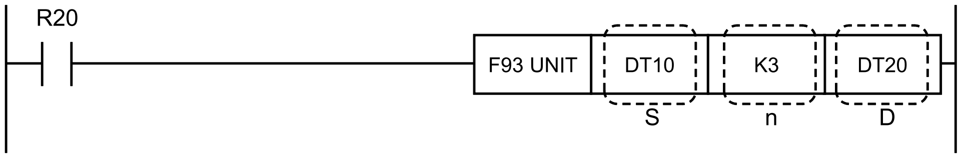
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the effective bit length for conversion (nL) is not 1 ≤ nL ≤ 8
	Turns ON when the result output start bit no. (nH) and the effective bit length for conversion (nL) is not 1 ≤ (nH + nL) ≤ 16 (consistency)
	Turns ON when all the data to be encoded is"0"

15.23 F93 UNIT (Digit Combine)

Combines the lower order 4 bits (bits 0 to 3) of 16-bit data.

■ Instruction format



■ Operands

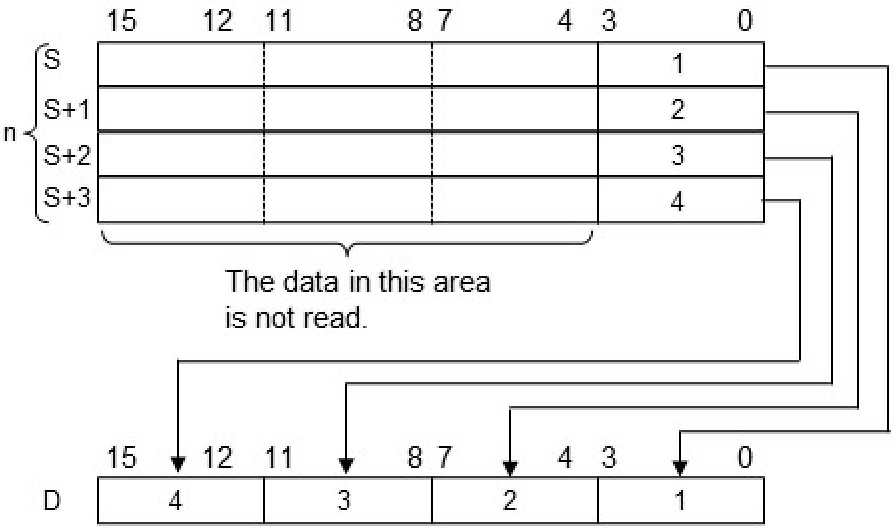
Items	Settings
S	The starting address of the area that stores the data to be combined
n	Area storing the number of data to be combined, or constant data
D	Area that stores the combined data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•		•	•					•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

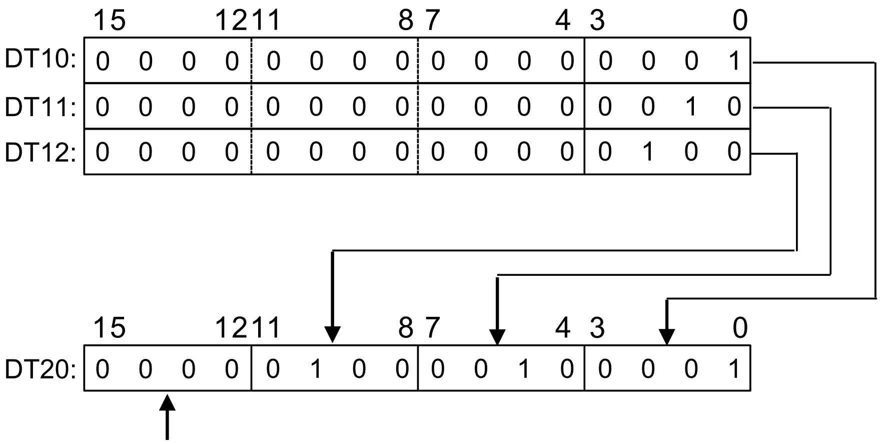
- The lower 4 bits of n points of data from the area specified by [S] are stored in order 4 bits at a time from the lower order of the area specified by [D].
- The number of data areas to be combined [n] can be specified within the range 0 to 4.
- When n = 0, no operation takes place.
- If n < 4, the remainder of [D] is filled with"0".



■ Operation example

Operation of instruction format description program

When internal relay R20 is ON, the lower 4 bits from data register 10, the lower 4 bits from DT11, and the lower 4 bits from DT12 are each stored from the lower order of DT20 4 bits at a time.



If [n] is less than 4, the 4 bits corresponding to the output destination are filled with"0".

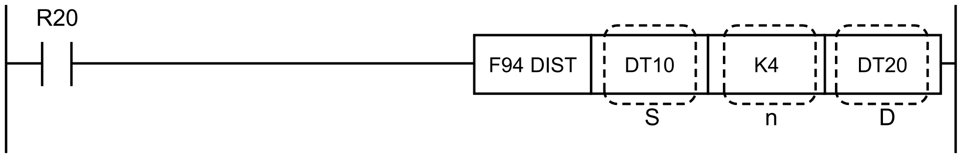
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON if the number of data areas to be combined [n] is n ≥ 5

15.24 F94 DIST (Digit Distribute)

Divides 16-bit data into four 4-bit units and distributes it.

■ Instruction format



■ Operands

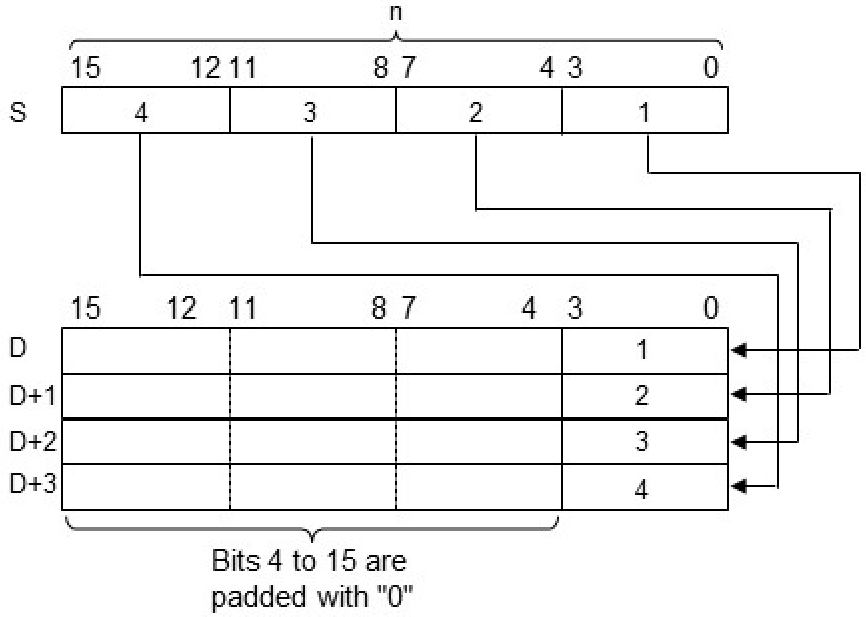
Items	Settings
S	Area storing the 16-bit data to be divided, or constant data
n	Area storing the number of data items to be divided, or constant data
D	Starting address of the area storing each divided digit

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

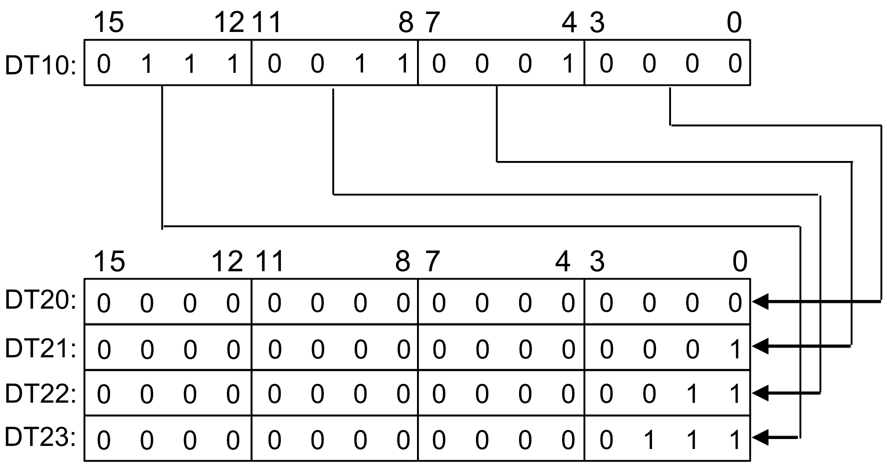
- The 16-bit data specified by [S] is divided into 4-bit (1-digit) units, and the digits specified by [n] are each stored in the lower 4 bits (bit positions 0 to 3) of n areas in order starting from the area specified by [D].
- The range of the number of data divisions that can be specified [n] is 0 to 4.
- When n = 0, no operation takes place.



■ Operation example

Operation of instruction format description program

When internal relay R20 turns ON, the data of data register DT10 is divided into 4 bits from the low bit, and 1 digit each is stored in order in the lower 4 bits of data registers DT20 to DT23.



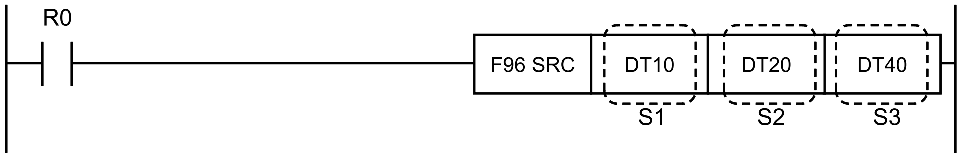
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the number of divided data items [n] is equal to or greater than 5
	Turns ON when the area is exceeded when distributing n data items to the address specified by [D]

15.25 F96 SRC (16-bit Data Search)

Searches for the specified 16-bit data from the area in the specified range (table).

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the data to be searched, or constant data
S2	Search table starting address
S3	Search table ending address

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2		●	●	●	●	●	●	●								●	
S3		●	●	●	●	●	●	●								●	

■ Outline of operation

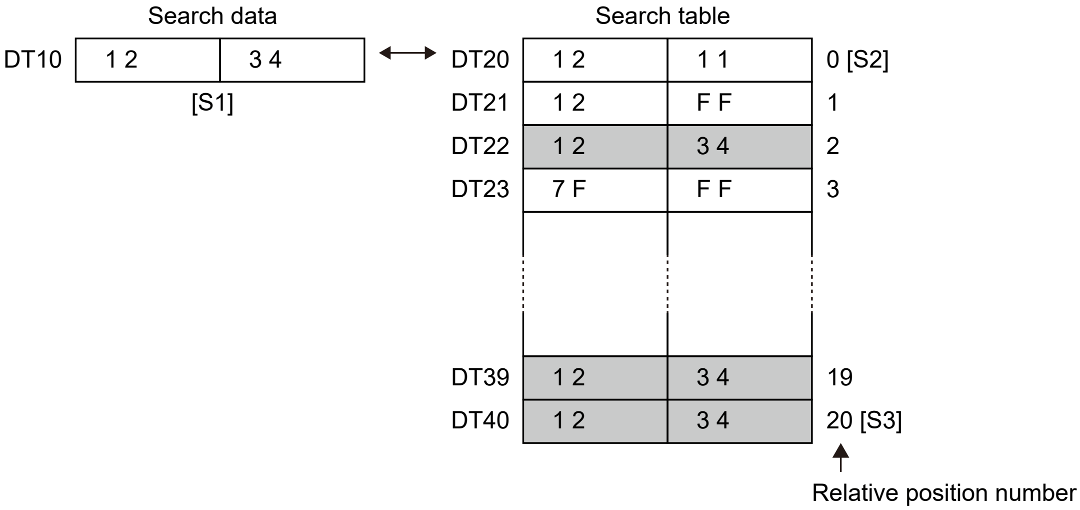
- The search data comprised of the 16-bit data specified by [S1] is searched for in the area (table) in the range specified by [S2] and [S3].
  The search results are stored as follows.
  1. The number of registers that have the same value is stored as a decimal number in special data register DT90037.
  2. The position of the first matching register is stored in special data register DT90038 at a relative position to [S2].
- [S2] specifies the starting address, and [S3] the ending address for the table.
- Specify the same type of memory area for [S2] and [S3]. Additionally, specify values so that [S2] is equal to or less than [S3].
- Data is searched in the direction from [S2] to [S3].

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, data that is the same content as the data in data register DT10 is searched in the range of data registers DT20 to DT40.

For example, to search the area of the value H1234, H1234 is written to DT10.



If DT22, DT39, and DT40 match the searched data, the following occurs.

1. If the number of registers matching the searched data equals 3
   "K3"is stored in DT90037.
2. If the position of the first matching data (the relative position number) equals 2
   "K2"is stored in DT90038.

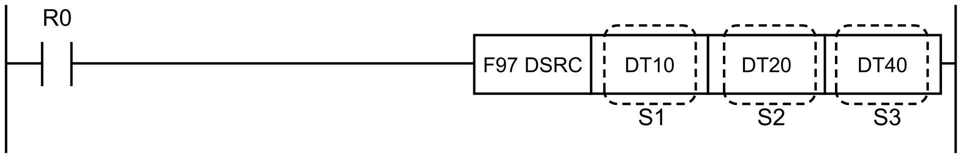
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	ON when [S2] > [S3]

15.26 F97 DSRC (32-bit Data Search)

Searches for specified 32-bit data in any area range (table).

■ Instruction format



■ Operands

Item	Setting
S1	Area storing the data to search for, or constant data (32-bit)
S2	Address of the search table starting area (32-bit)
S3	Address of the search table ending area (32-bit)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2		•	•	•	•	•	•	•	•							•	
S3		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- This instruction searches the area range (table) specified by [S2] and [S3] for the 32-bit search data specified by [S1].

  The search results are stored as follows.

  1. The number of registers with the same value is stored in special data register DT90037.
  2. The position of the first matching register is stored in special data register DT90038 at a relative position to [S2].
- [S2] specifies the starting address, and [S3] the ending address for the table.
- Specify the same type of memory area for [S2] and [S3]. Additionally, specify values so that [S2] is equal to or less than [S3].
- Data is searched in the direction from [S2] to [S3].

■ Precautions for programming

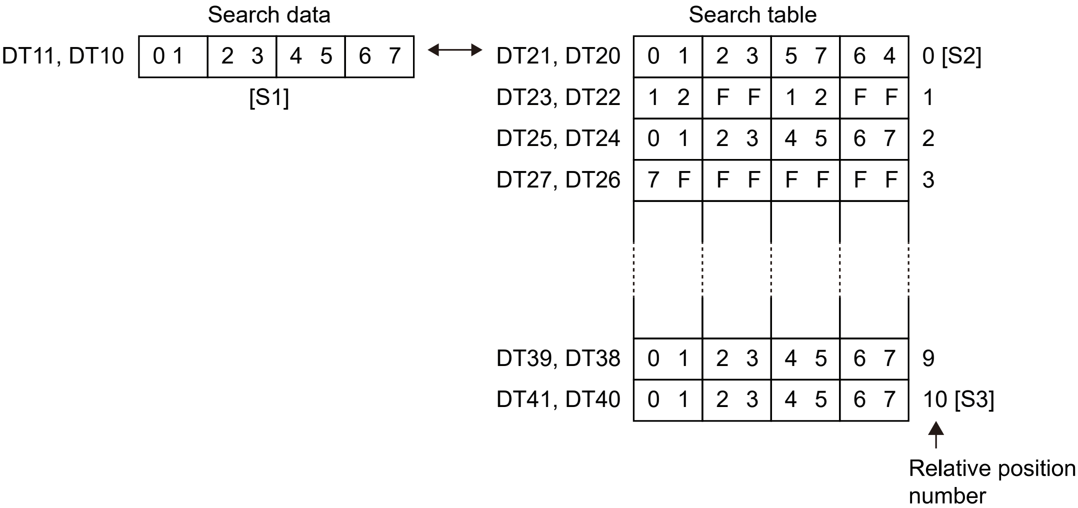
- When a memory area of 0 or even number is specified for [S2], also specify a memory area of even number for [S3].
- When a memory area of odd number is specified for [S2], also specify a memory area of odd number for [S3].

■ Operation example

Operation of instruction format description program

The program searches data registers DT20 through DT40 for the same data as that in data registers DT10 and DT11 when execution condition R0 turns on.

For example, to search the area for the value "H01234567", it writes "H01234567" to DT10 and DT11.



If "DT24, DT25", "DT38, DT39", and "DT40, DT41" match the searched data, the following occurs.

1. If the number of registers matching the searched data equals 3
   "K3" is stored in DT90037.
2. If the position of the first matching data (the relative position number) equals 2
   "K2" is stored in DT90038.

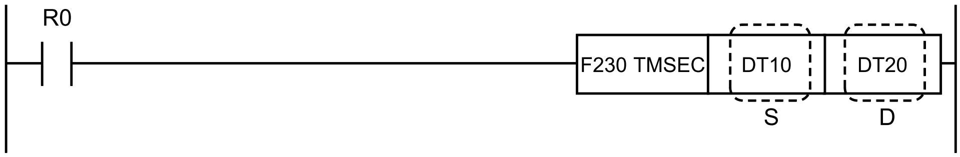
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when [S2] > [S3].

15.27 F230 TMSEC (Time to Seconds Conversion)

Converts the specified time of day data (year, month, day, hour, minute, second) into number of seconds.

■ Instruction format



■ Operands

Item	Setting
S	Area storing the data to be converted, or constant data
D	Area to store the conversion result

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	SWR	SDT	Constant				Index modifier	Integer device
													K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●					●	
D		●	●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The input time data [S to S+2] is converted from standard time (*1) to number of seconds and the conversion result is stored in [D, D+1] as a 32-bit integer value.

  (*1): Standard time is 00:00'00" on January 1, '01. The conversion result is output as a binary value.
- Time data conversion outputs time that takes into account leap years.

1 minute	60 seconds conversion
1 hour	60 minutes conversion
1 day	24 hours conversion
1 year (leap year)	366 days conversion
1 year (regular year)	365 days conversion
Leap year	2/29 (every 4 years)

- The time data (S) must be specified as BCD data and a value within the range must be registered.

Time data (S) (BCD)

	(Higher)	(Lower)
S:	Minutes (H00 to H59)	Seconds (H00 to H59)
S+1:	Day (H01 to H31)	Hour (H01 to H23)
S+2:	Year (H00 to H99)	Month (H00 to H12)

⇩

Seconds data (D) (32-bit integer)

D:	Seconds data	(Higher)
D+1:	(H00000000 to HBC19137F)	(Lower)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the time data of data registers DT10 to DT12 is converted from standard time to number of seconds and the result is stored in DT20 and DT21.

Example 1)

	(Higher)	(Lower)
DT10	H37	H26
DT11	H03	H12
DT12	H02	H12

(02/12/03 – 12:37:26) (2002)

⇨

H039CDD06	DT20
(Binary value)	DT21

(60, 611, 846 seconds)

Example 2)

	(Higher)	(Lower)
DT10	H59	H59
DT11	H31	H23
DT12	H00	H12

(00/12/31 - 23:59:59) (2100)

⇨

HBC19137F	DT20
(Binary value)	DT21

(3,155,759,999 seconds)

Correspondence between time of day data and second data

	Time data (S)	Second data (D)
2001	'01/01/01 00:00:00	H00000000
:	'01/01/01 00:00:01	H00000001
:	:	:
:	'01/01/01 00:01:00	H0000003C
:	:	:

	Time data (S)	Second data (D)
:	'01/01/01 01:00:00	H00000E10
:	:	:
:	'01/02/01 00:00:00	H00015180
:	:	:
2099	'12/31/99 23:59:59	HBA368E7F
2100	'00/01/01 00:00:00	HBA368E80
:	:	:
2100	'12/31/00 23:59:59	HBC19137F

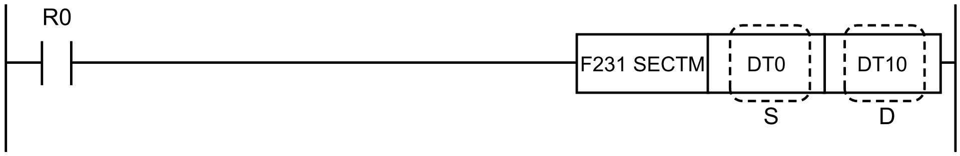
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when a value other than BCD is specified for [S].
	Turns ON when a value that exceeds the range is specified for any one of month, day, hour, minute, or second in the time data of [S].
	Turns ON when the data of [S] exceeds the area.

15.28 F231 SECTM (Seconds to Time Conversion)

The specified number of seconds is changed into time data (year/month/day/hour/minute/second).

■ Instruction format



■ Operands

Item	Setting
S	Area storing the number of seconds (32 bits)
D	Starting area storing the time data

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	FL	I	SWR	SDT	Constant				Index modifier	Integer device
													K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●					●	
D		●	●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The input number of seconds [S to S+2] is converted to the time data based on the standard time (*1), and stored in [D, D+1].
  (*1): Standard time is 00:00'00" on January 1, '01.
- Time data conversion outputs time that takes into account leap years.

1 minute	60 seconds conversion
1 hour	60 minutes conversion
1 day	24 hours conversion
1 year (leap year)	366 days conversion
1 year (regular year)	365 days conversion
Leap year	2/29 (every 4 years)

- The number of seconds (S) must be within a range of values that can be expressed in time data, equaling up to 100 years.

H 0 to H BC19137F	Normal conversion
H BC191380 to H FFFFFFFF	Conversion error

Seconds data (S) (32-bit integer)

S:	Seconds data	(Lower)
S+1:	(H00000000 to HBC19137F)	(Higher)

Time of day data (D) (BCD)

	(Higher)	(Lower)
D:	Minutes (H00 to H59)	Seconds (H00 to H59)
D+1:	Day (H01 to H31)	Hour (H01 to H23)
D+2:	Year (H00 to H99)	Month (H00 to H12)

■ Operation example

Operation of instruction format description program

When the internal relay R0 is ON, the number of seconds for the data registers DT0 and DT1 is converted to the time data based on the standard time, and stored in DT10 to DT12.

Example 1)

DT0	H039D0A6A
DT1	(Binary value)

(60, 623, 466 seconds)

(Higher)	(Lower)	
H51	H06	DT10
H03	H15	DT11
H02	H12	DT12

(02/12/03 – 15:51:06) (2002)

Example 2)

DT0	HBC19137F
DT1	(Binary value)

(3,155,759,999 seconds)

(Higher)	(Lower)	
H59	H59	DT10
H31	H23	DT11
H00	H12	DT12

(00/12/31 - 23:59:59) (2100)

Second conversion

Second data (D)	Time data (S)	
H00000000	'01/01/01 00:00:00	2001
H00000001	'01/01/01 00:00:01	:
:	:	:
H0000003C	'01/01/01 00:01:00	:
:	:	:

Second data (D)	Time data (S)	
H00000E10	'01/01/01 01:00:00	:
:	:	:
H00015180	'01/02/01 00:00:00	:
:	:	:
HBA368E7F	'12/31/99 23:59:59	2099
HBA368E80	‘00/01/01 00:00:00	2100
:	:	:
HBC19137F	'12/31/00 23:59:59	2100

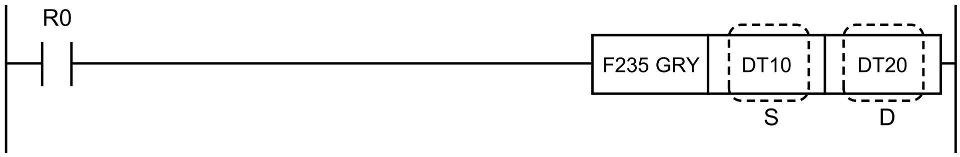
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the number of seconds [S] is [S] ≥ HBC191380 (number of seconds in 100 years).
	Turns ON when the data memory of [D] exceeds the area.

15.29 F235 GRY (16-bit Data to Gray Code Conversion)

Converts the specified 16-bit data to gray code.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the data to be converted, or constant data
D	Area to store the conversion result

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The 16-bit data in the area specified by [S] is converted to gray code and stored in the area specified by [D].

Info.

- For the gray code, refer the correspondence table in"P.15-75".

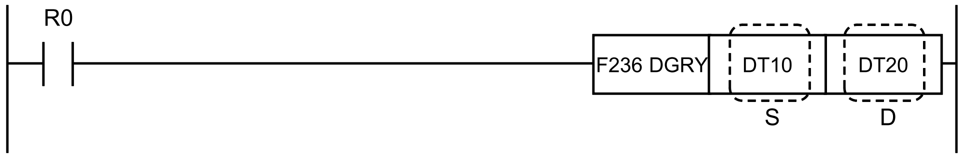
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

15.30 F236 DGRY (32-bit Data to Gray Code Conversion)

Converts specified 32-bit data to gray code.

■ Instruction format



■ Operands

Items	Settings
S	Area (two word) storing the data to be converted, or constant data
D	Area (two word) to store the conversion result

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- Converts the 32-bit data specified by [S] to gray code, and stores the converted data in the area specified by [D].

Info.

- For the gray code, refer the correspondence table in"P.15-75".

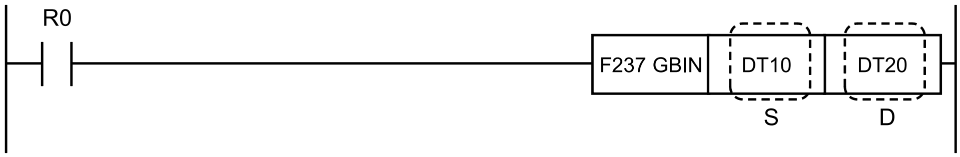
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

15.31 F237 GBIN (Gray Code to 16-bit Data Conversion)

Converts the gray code in the specified area to 16-bit binary data.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the data to be converted, or constant data
D	Area to store the conversion result

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The gray code of the area specified by [S] is converted to 16-bit binary data and stored in the area specified by [D].

Info.

- For the gray code, refer the correspondence table in"P.15-75".

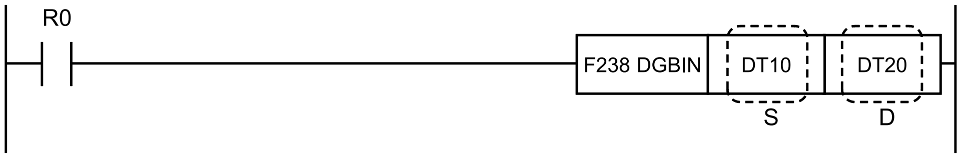
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

15.32 F238 DGBIN (Gray Code to 32-bit Data Conversion)

The gray code in the specified area is converted to 32-bit binary data.

■ Instruction format



■ Operands

Items	Settings
S	Area (two word) storing the data to be converted, or constant data
D	Area (two words) to store the conversion result

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The gray code in the area specified in [S] is converted to 32-bit binary data and stored in the area specified in [D].

Info.

- For the gray code, refer the correspondence table in"P.15-75".

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

BIN/Gray Code Correspondence Table

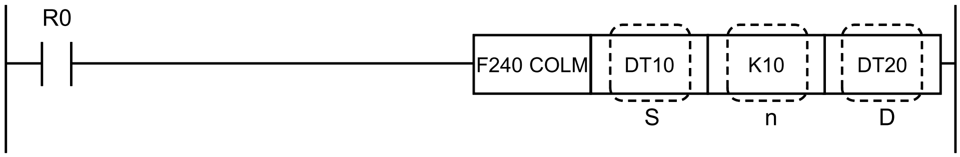
Decimal (Decimal)	Binary (Binary)	Gray code (Gray code)
0	0000 0000 0000 0000	0000 0000 0000 0000
1	0000 0000 0000 0001	0000 0000 0000 0001
2	0000 0000 0000 0010	0000 0000 0000 0011

Decimal (Decimal)	Binary (Binary)	Gray code (Gray code)
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
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 1101
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 0011 1111	0000 0000 0010 0000
64	0000 0000 0100 0000	0000 0000 0110 0000
:	:	:
255	0000 0000 1111 1111	0000 0000 1000 0000

15.33 F240 COLM (Bit Line to Bit Column Conversion)

Converts a bit line to a bit column.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the hexadecimal data or constant data
n	Area storing the bit position specification, or constant data
D	Starting address of the area that will be overwritten by the bit column

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
n	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The bit data at the position specified by [n] in the 16-word data area starting from [D] is rewritten by the 16-bit data of the area specified by [S].
- The contents of the bits of the 16-word data area starting from [D] that are not specified do not change.
- [n] can be specified in the range of 0 to 15.

e.g. When the specified bit position n = 10 (K10)

	15															0
S	0	1	0	1	0	0	0	1	1	1	0	1	1	0	0	1

↓

	15					10										0
D						1										
D+1						0										
D+2						0										
D+3						1										
D+4						1										
D+5						0										
D+6						1										
D+7						1										
D+8						1										
D+9						0										
D+10						0										
D+11						0										
D+12						1										
D+13						0										
D+14						1										
D+15						0										

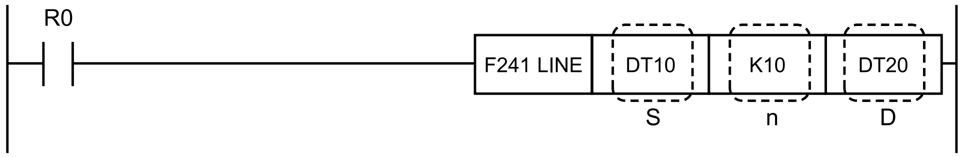
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON if 0 ≤ [n] ≤ 15 is not true
	Turns ON when the conversion result exceeds the area specified by [D] when stored

15.34 F241 LINE (Bit Column to Bit Line Conversion)

Converts a bit column to a bit line.

■ Instruction format



■ Operands

Items	Settings
S	Starting address of area where bit column will be read
n	Area storing the bit position specification, or constant data
D	Area to store the conversion result

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●					●	
n	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- Reads the bit data at the position specified by [n] from the area specified by [S] and stores it in the area specified by [D].
- [n] can be specified in the range of 0 to 15.

e.g. When the specified bit position n = 10 (K10)

	15					10										0
S						1										
S+1						0										
S+2						0										
S+3						1										
S+4						1										
S+5						0										
S+6						1										
S+7						1										
S+8						1										
S+9						0										
S+10						0										
S+11						0										
S+12						1										
S+13						0										
S+14						1										
S+15						0										

↓

	15															0
D	0	1	0	1	0	0	0	1	1	1	0	1	1	0	0	1

■ Flag operations

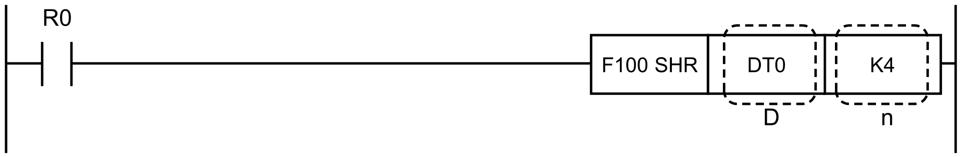
Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON if 0 ≤ [n] ≤ 15 is not true
	Turns ON when the conversion range specified by [S] exceeds the area

16 Data Shift Instruction

16.1 F100 SHR (16-bit Data Right Shift)

Shifts 16-bit data to the right by a specified number of bits.

■ Instruction format



■ Operands

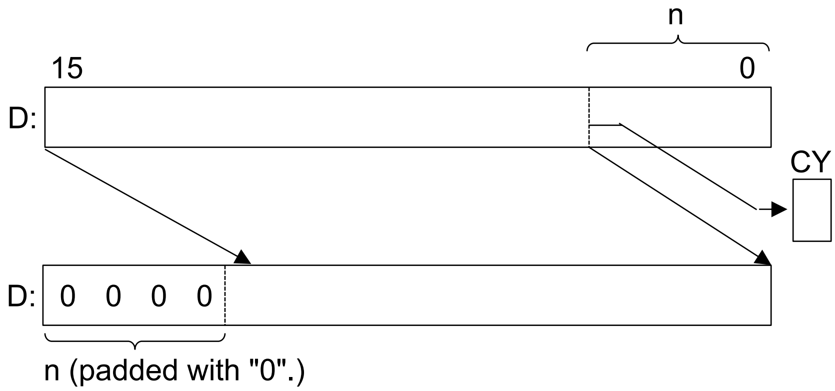
Items	Settings
D	Area storing the 16-bit data to be shifted
n	Area storing the number of bits to be shifted, or constant data

■ Devices that can be specified (indicated by •)

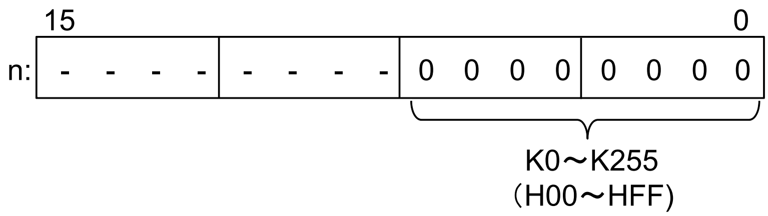
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

- Shifts the 16-bit data specified by [D] by the number of bits (specified in decimal form) specified by [n] to the right (the lower bit direction).



- When the data is shifted to the right,
  1. the n bits from the most significant bit are filled with 0.
  2. The content from the least significant bit to the nth bit is stored in the CY (carry) flag (R9009).
- For [n], only the lower 8 bits of the 16-bit data are valid. The shift amount can be selected from 1 bit to 255 bits.

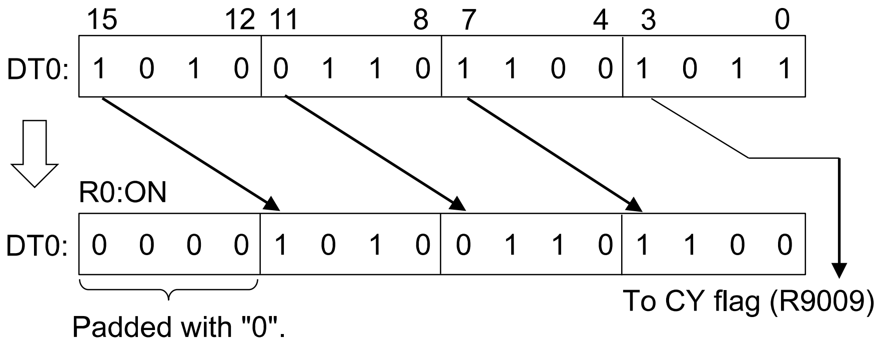


(Note 1) The bits marked with - are invalid.

■ Operation example

Operation of instruction format description program

Shifts the content of DT0 four bits to the right when internal relay R0 turns ON.
The content of bit 3 before the shift is stored in the CY (carry) flag.



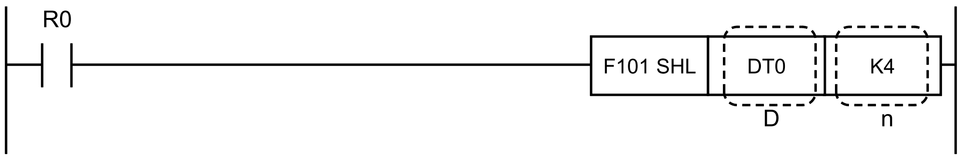
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Turns ON when the content of the least significant bit to the n bit is"1"

16.2 F101 SHL (16-bit Data Left Shift)

Shifts 16-bit data to the left by the specified number of bits.

■ Instruction format



■ Operands

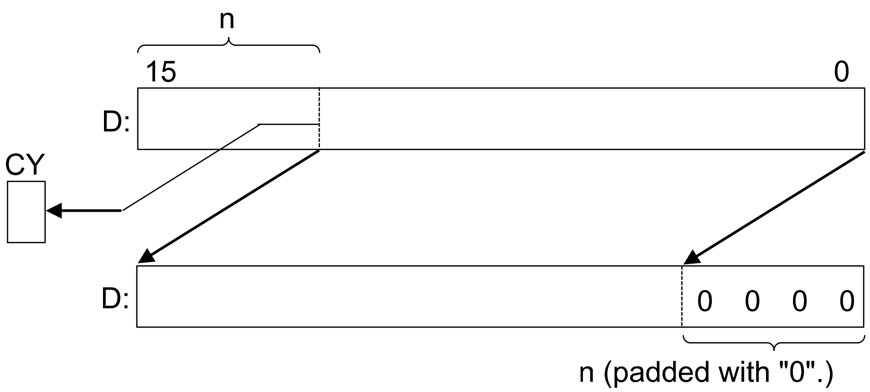
Items	Settings
D	Area storing the 16-bit data to be shifted
n	Area storing the number of bits to be shifted, or constant data

■ Devices that can be specified (indicated by •)

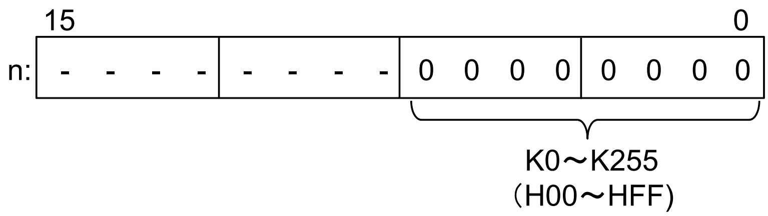
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

- The 16-bit data specified by [D] is shifted to the left (in the high bit direction) by the number of bits specified by [n] (specified as a decimal number).



- When the data is shifted to the left,
  1. the n bits from the least significant bit are filled with 0.
  2. The content from the most significant bit to the nth bit is stored in the CY (carry) flag (R9009).
- For [n], only the lower 8 bits of the 16-bit data are valid. The shift amount can be selected from 1 bit to 255 bits.

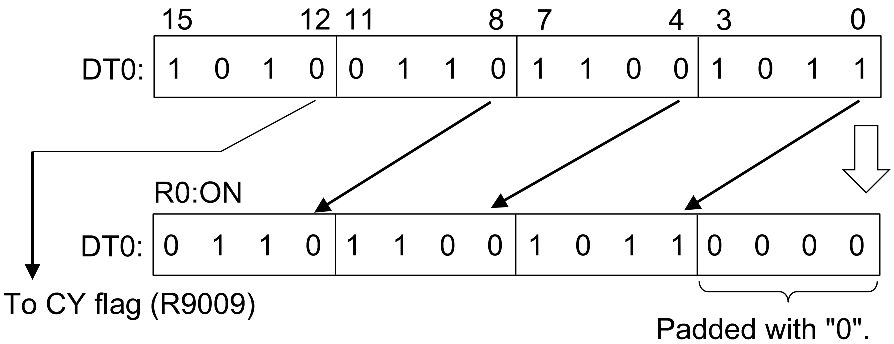


(Note 1) The bits marked with - are invalid.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of DT0 shifts four bits to the left.
The content of bit 12 before the shift is stored in the CY (carry) flag.



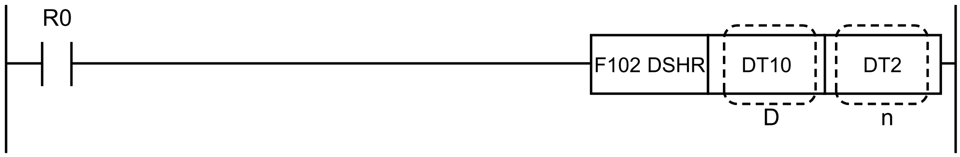
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Turns ON when the content of the nth bit from the most significant bit is"1"

16.3 F102 DSHR (32-bit Data Right Shift)

Shifts 32-bit data (double-word data) n bits to the right.

■ Instruction format



■ Operands

Items	Settings
D	Area storing the double-word data to be shifted (two words)
n	Area storing the number of bits to be shifted, or constant data

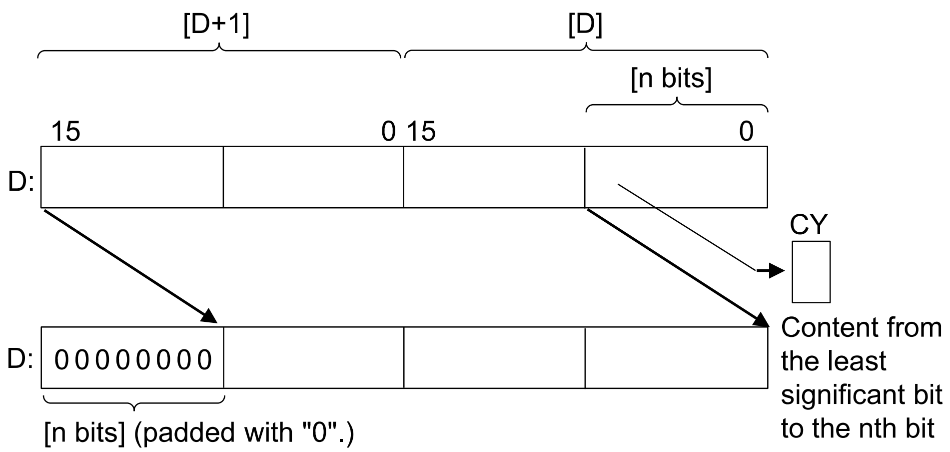
■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

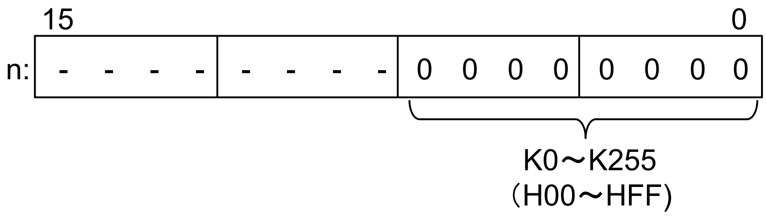
- The double-word data specified by [D, D+1] is shifted to the right (in the low bit direction) by the number of bits specified by [n] (16-bit K constant).

n bits shifted to the right



- When the data is shifted to the right,
  1. the n bits from the most significant bit are filled with 0.
  2. The content from the least significant bit to the nth bit is stored in the CY (carry) flag (R9009).

- For [n], only the lower 8 bits of the 16-bit data are valid. The shift amount can be selected from 1 bit to 255 bits.



(Note 1) The bits marked with - are invalid.

- When [n] = K0, the content of [D, D+1] and the CY flag do not change.
- When [n] is specified as K32 or higher, the content of [D, D+1] changes to 0.

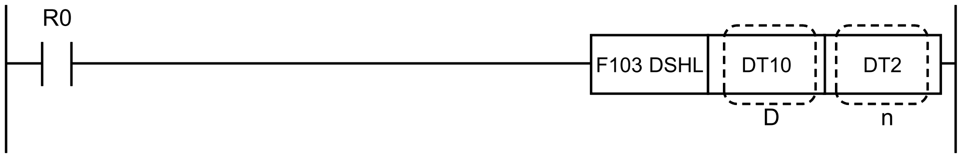
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Reflects the content of the nth bit from the least significant bit immediately before the instruction is executed.

16.4 F103 DSHL (32-bit Data Left Shift)

Shifts 32-bit data (double-word data) n bits to the left.

■ Instruction format



■ Operands

Items	Settings
D	Area storing the double-word data to be shifted (two words)
n	Area storing the number of bits to be shifted, or constant data

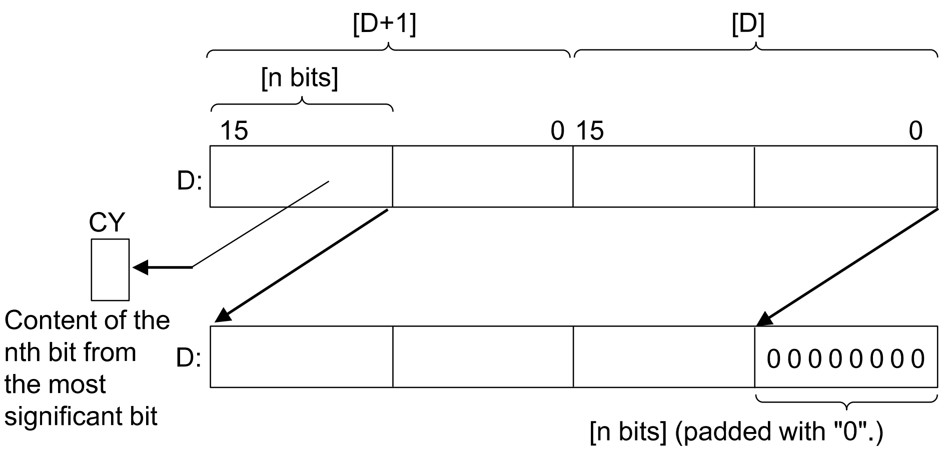
■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		●	●	●	●	●	●	●	●							●	
n	●	●	●	●	●	●	●	●	●	●	●	●	●			●	

■ Outline of operation

- The double-word data specified by [D, D+1] is shifted to the left (in the high bit direction) by the number of bits specified by [n] (16-bit K constant).

n bits shifted to the left



- When the data is shifted to the left,
  1. the n bits from the least significant bit are filled with 0.
  2. The content from the most significant bit to the nth bit is stored in the CY (carry) flag (R9009).

- For [n], only the lower 8 bits of the 16-bit data are valid. The shift amount can be selected from 1 bit to 255 bits.

	15			0
n:	- - - -	- - - -	0 0 0 0	0 0 0 0

$\underbrace{0\ 0\ 0\ 0\ \ 0\ 0\ 0\ 0}_{\text{K0～K255 (H00～HFF)}}$

(Note 1) The bits marked with - are invalid.

- When [n] = K0, the content of [D, D+1] and the CY flag do not change.
- When [n] is specified as K32 or higher, the content of [D, D+1] changes to 0.

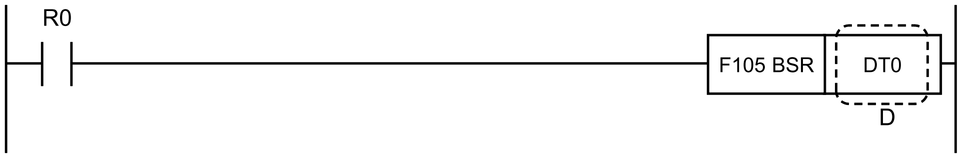
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Reflects the content of the nth bit from the most significant bit immediately before the instruction is executed.

16.5 F105 BSR (16-bit Data 1-Digit Right Shift)

Shifts 16-bit data one digit (four bits) to the right.

■ Instruction format



■ Operands

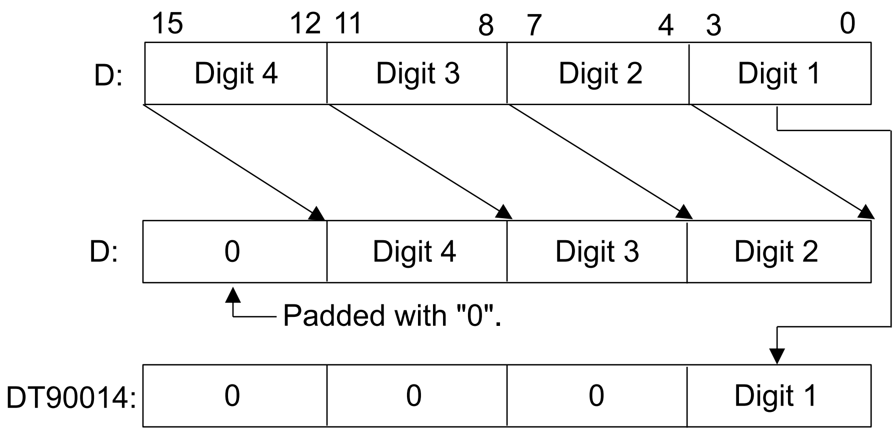
Items	Settings
D	Area storing the 16-bit data to be shifted

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The 16-bit data (four digits) specified by [D] is shifted one digit (four bits) to the right (downward direction).

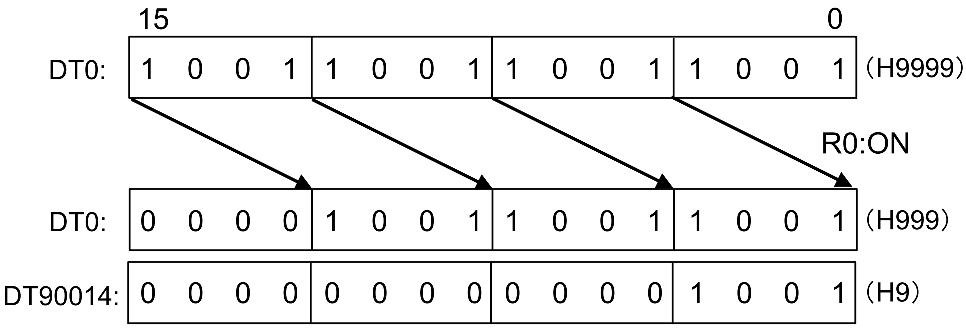


- When the data is shifted to the right,
  1. bits 0 to 3 (Digit 1) before the shift are stored in bits 0 to 3 of special data register DT90014.
  2. After the shift, bits 12 to 15 are filled with 0.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of DT0 shifts one digit to the right. The content of bits 0 to 3 before the shift are stored in bits 0 to 3 of DT90014.



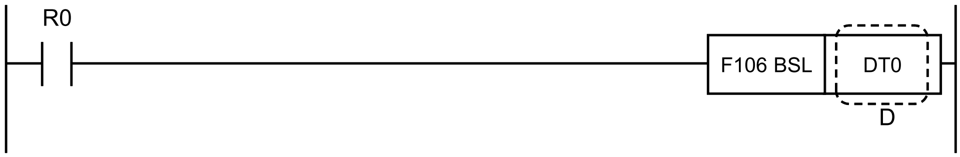
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

16.6 F106 BSL (16-bit Data 1-Digit Left Shift)

Shifts 16-bit data one digit (four bits) to the left.

■ Instruction format



■ Operands

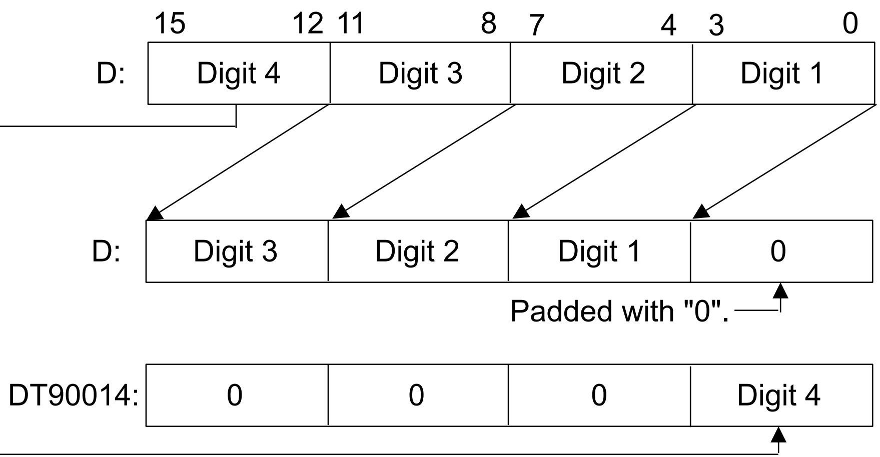
Items	Settings
D	Area storing the 16-bit data to be shifted

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The 16-bit data (four digits) specified by [D] is shifted one digit (four bits) to the left (upward direction).

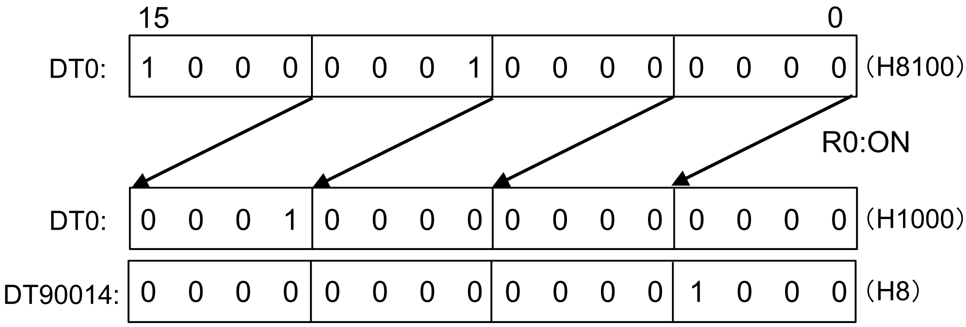


- When the data is shifted to the left,
  1. bits 12 to 15 before the shift are stored in bits 0 to 3 of special data register DT90014.
  2. After the shift, bits 0 to 3 are filled with 0.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of DT0 shifts one digit to the left. The contents of bits 12 to 15 before the shift are stored in bits 0 to 3 of DT90014.



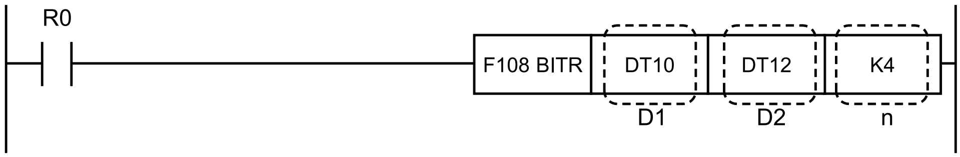
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

16.7 F108 BITR (Block Area Bitwise Right Shift)

Shifts a block area to the right in bit units.

■ Instruction format



■ Operands

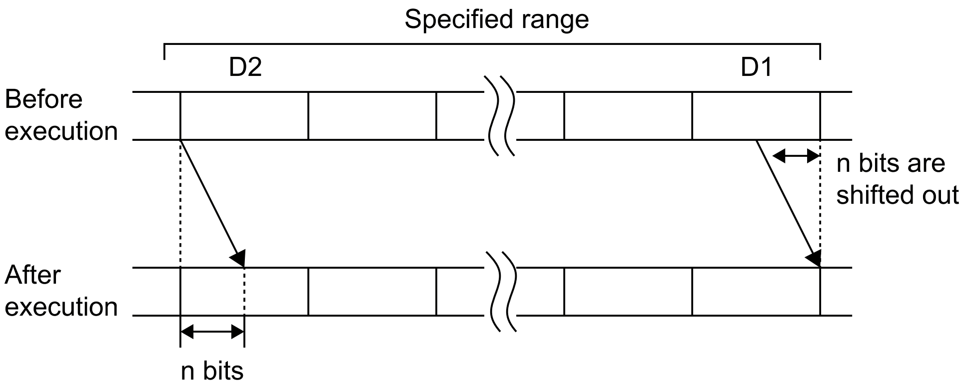
Items	Settings
D1	Starting address of the area to be shifted
D2	Ending address of the area to be shifted
n	Area storing the number of bits to be shifted, or constant data

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D1		●	●	●	●	●	●	●	●							●	
D2		●	●	●	●	●	●	●	●							●	
n	●	●	●	●	●	●	●	●	●	●	●	●	●			●	

■ Outline of operation

- The area in the range specified by [D1] and [D2] is shifted to the right by the number of bits specified by [n].



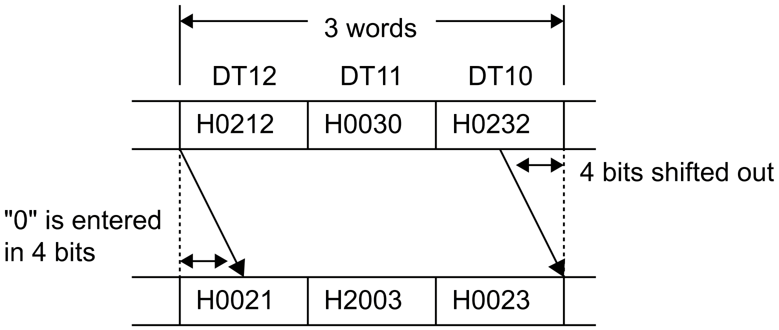
- The starting address of the area to be shifted is specified by [D1] and the ending address is specified by [D2].
- Specify the same type of area for both [D1] and [D2]. Also, specify values so that D1 is equal to or smaller than D2.

- When the data is shifted to the right,
  1. the lower n bits of [D1] before the shift are shifted out.
  2. After the shift, the upper n bits of [D2] are filled with 0.
- No operation takes place if [n] = 0.
- If [n] is set to a number of bits that exceeds the area in the range specified by [D1] and [D2], the value of the area from [D1] to [D2] is 0.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the three-word data in DT10 to DT12 is shifted four bits to the right



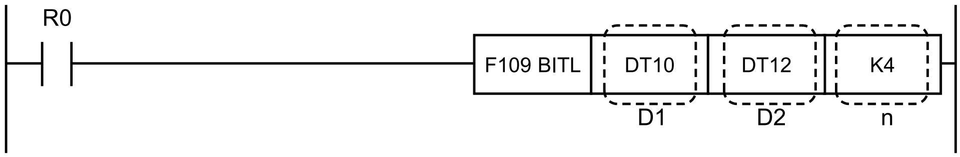
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when [D1] > [D2]

16.8 F109 BITL (Block Area Bitwise Left Shift)

Shifts a block area left in bit units.

■ Instruction format



■ Operands

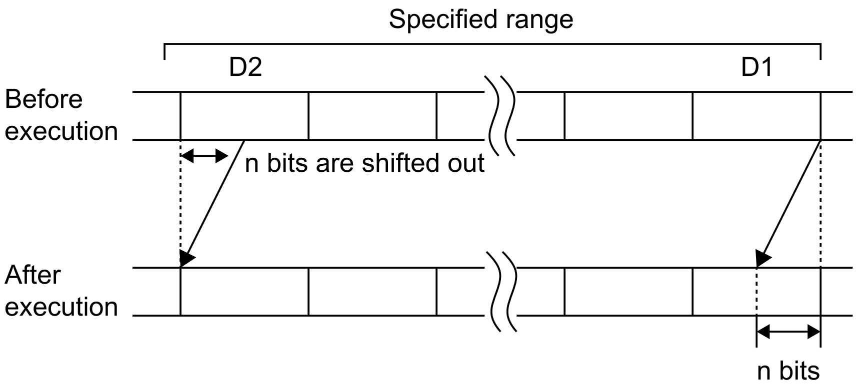
Items	Settings
D1	Starting address of the area to be shifted
D2	Ending address of the area to be shifted
n	Area storing the number of bits to be shifted, or constant data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D1		•	•	•	•	•	•	•	•							•	
D2		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

- The area in the range specified by [D1] and [D2] is shifted left by the number of bits specified by [n].



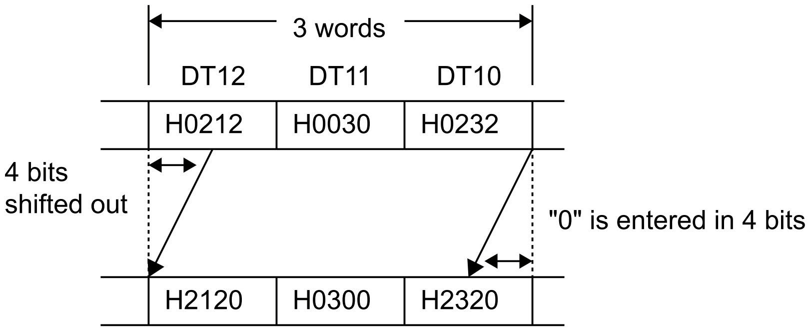
- The starting address of the area to be shifted is specified by [D1] and the ending address is specified by [D2].
- Specify the same type of area for both [D1] and [D2]. Also, specify values so that D1 is equal to or smaller than D2.

- When the data is shifted to the left,
  1. the upper n bits of [D2] before the shift are shifted out.
  2. After the shift, the lower n bits of [D1] are filled with 0.
- No operation takes place if [n] = 0.
- If [n] is set to a number of bits that exceeds the area in the range specified by [D1] and [D2], the value of the area from [D1] to [D2] is 0.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the three-word data in DT10 to DT12 is shifted four bits to the left.



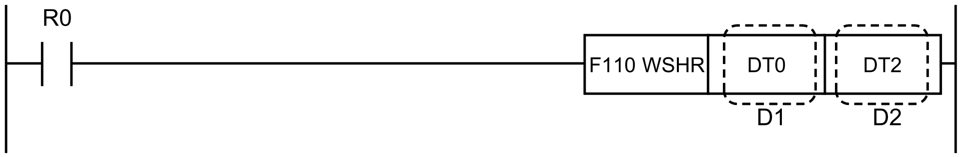
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when [D1] > [D2]

16.9 F110 WSHR (Block Area 1 Word Right Shift)

Shifts the specified data range one word to the right.

■ Instruction format



■ Operands

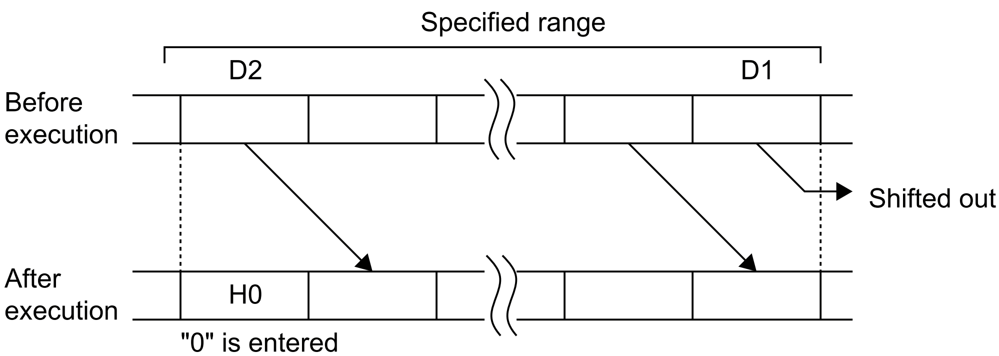
Items	Settings
D1	Starting address of the area to be shifted
D2	Ending address of the area to be shifted

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D1		•	•	•	•	•	•	•								•	
D2		•	•	•	•	•	•	•								•	

■ Outline of operation

- The area of the range specified by [D1] and [D2] is shifted one word to the right (downward direction).

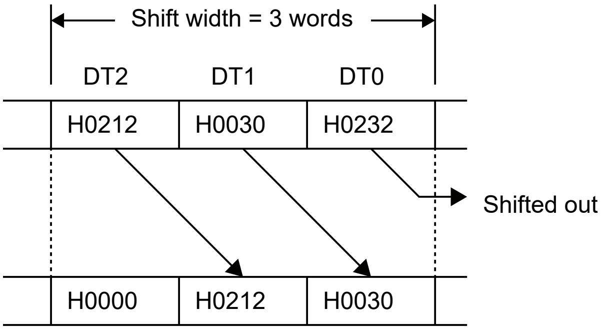


- The starting address of the area to be shifted is specified by [D1] and the ending address is specified by [D2].
- Specify the same type of area for both [D1] and [D2]. Also, make sure that [D1] address ≤ [D2] address.
- When the data is shifted to the right,
  1. the content of [D1] before the shift is lost.
  2. After the shift, [D2] is filled with H0.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the three-word data in DT0 to DT2 is shifted one word to the right.



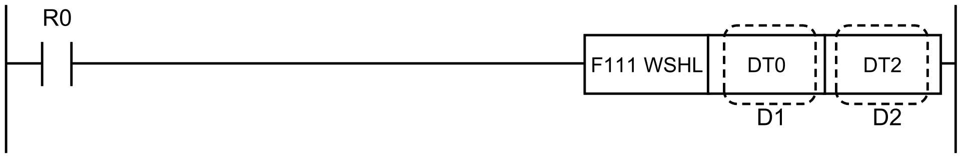
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the [D1] address > [D2] address

16.10 F111 WSHL (Block Area 1 Word Left Shift)

Data in the specified range is shifted one word to the left.

■ Instruction format



■ Operands

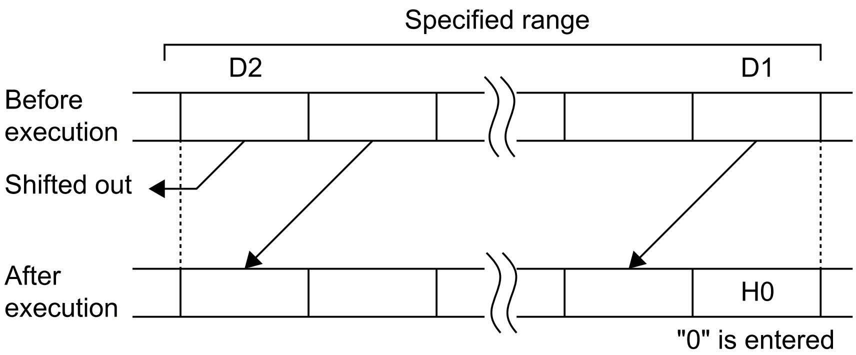
Items	Settings
D1	Starting address of the area to be shifted
D2	Ending address of the area to be shifted

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
D1		•	•	•	•	•	•	•								•	
D2		•	•	•	•	•	•	•								•	

■ Outline of operation

- The range area specified by [D1] and [D2] is shifted to the left (upper direction) by one word.

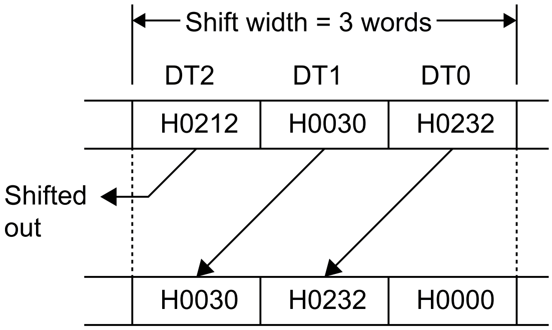


- The starting address of the area to be shifted is specified by [D1] and the ending address is specified by [D2].
- Specify the same type of area for both [D1] and [D2]. Also, make sure that [D1] address ≤ [D2] address.
- When the data is shifted to the left,
  1. the content of [D2] before the shift is lost.
  2. After the shift, [D1] is filled with H0.

■ Operation example

Operation of instruction format description program

Three-word data from DT0 to DT2 is shifted one word to the left when internal relay R0 turns ON.



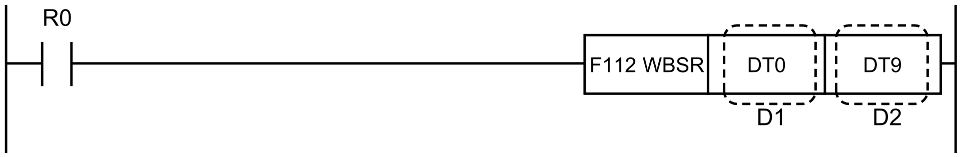
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the [D1] address > [D2] address

16.11 F112 WBSR (Block Area 1 Digit Right Shift)

Data in the specified range is shifted 1 digit to the right.

■ Instruction format



■ Operands

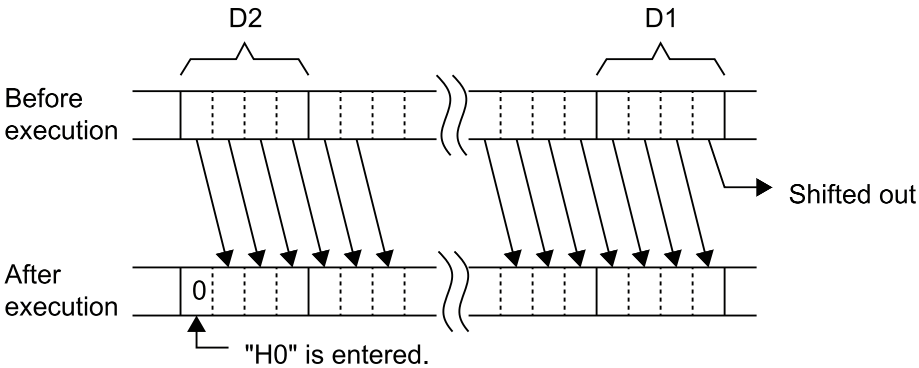
Items	Settings
D1	Starting address of the area to be shifted
D2	Ending address of the area to be shifted

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D1		•	•	•	•	•	•	•								•	
D2		•	•	•	•	•	•	•								•	

■ Outline of operation

- The area of the range specified in [D1] and [D2] is shifted to the right (lower direction) by 1 digit (4 bits).

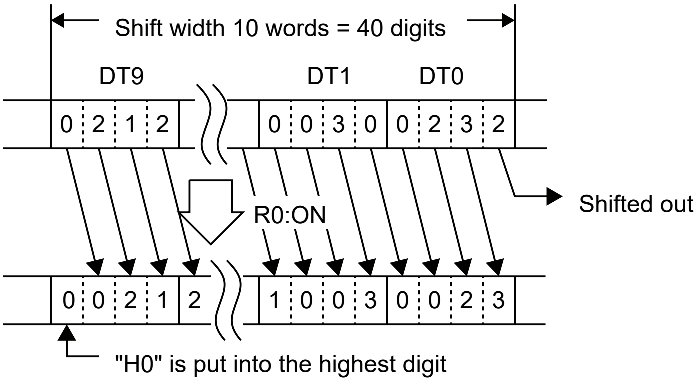


- The starting address of the area to be shifted is specified by [D1] and the ending address is specified by [D2].
- Specify the same type of area for both [D1] and [D2]. Also, make sure that [D1] address ≤ [D2] address.
- When the data is shifted to the right,
  1. the content of bits 0 to 3 (Digit 1) of [D1] before the shift is lost.
  2. After the shift, bits 12 to 15 of [D2] (Digit 4) are filled with "0".

■ Operation example

Operation of instruction format description program

When internal relay R0 is ON, 10 word data of DT0 to DT9 is shifted 1 digit to the right.



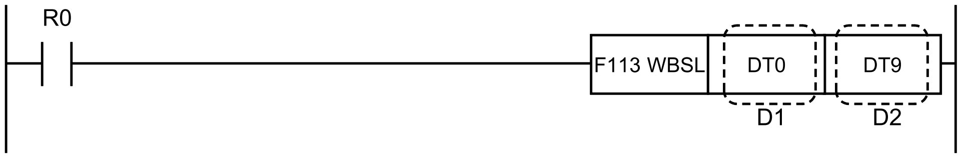
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the [D1] address > [D2] address

16.12 F113 WBSL (Block Area 1 Digit Left Shift)

Shifts data in a specified range one digit to the left.

■ Instruction format



■ Operands

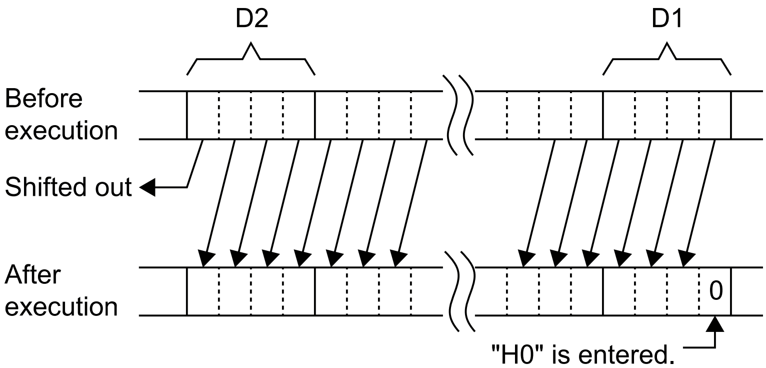
Items	Settings
D1	Starting address of the area to be shifted
D2	Ending address of the area to be shifted

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
D1		●	●	●	●	●	●	●								●	
D2		●	●	●	●	●	●	●								●	

■ Outline of operation

- Shifts an area of a range specified in [D1] and [D2] one digit (4 bits) to the left (toward the higher digit).

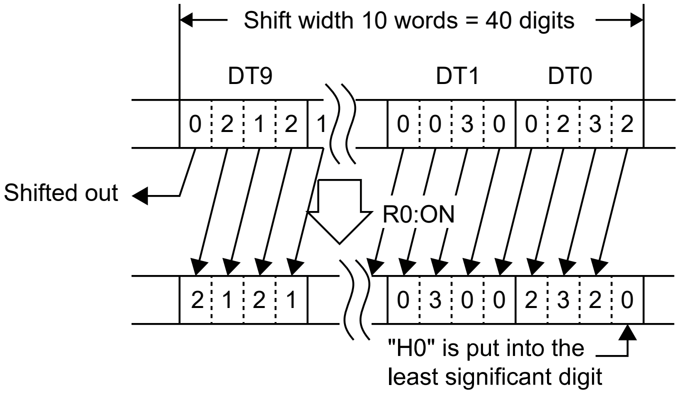


- The starting address of the area to be shifted is specified by [D1] and the ending address is specified by [D2].
- Specify the same type of area for both [D1] and [D2]. Also, make sure that [D1] address ≤ [D2] address.
- When the data is shifted to the left,
  1. the content of bits 12 to 15 (Digit 4) of [D2] before the shift is lost.
  2. After the shift, bits 0 to 3 of [D1] (Digit 1) are filled with "0".

■ Operation example

Operation of instruction format description program

When the internal relay R0 is ON, the data of 10 words from DT0 to DT9 is shifted to the left by one digit.



■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the [D1] address > [D2] address

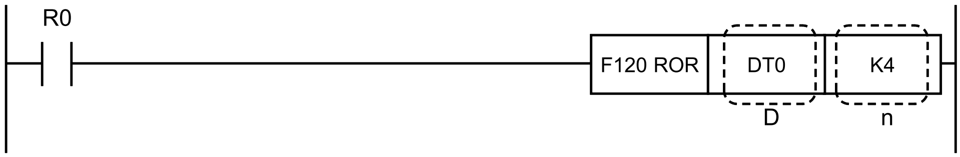
(MEMO)

17 Data Rotation Instructions

17.1 F120 ROR (16-Bit Data Rotation to the Right)

Rotates the specified 16-bit data to the right.

■ Instruction format



■ Operands

Items	Settings
D	Area targeted for rotation
n	Area storing the number of bits specified to be rotated, or constant data

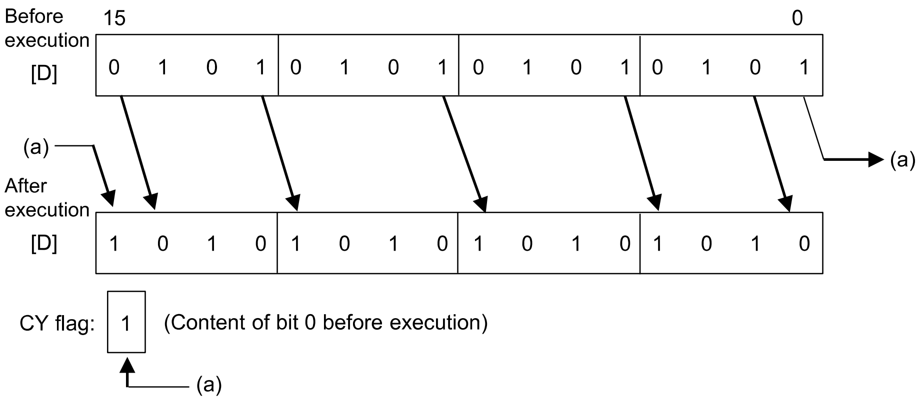
■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

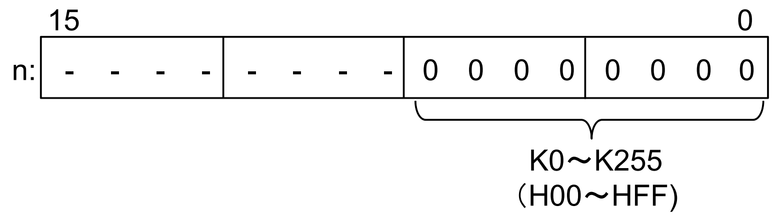
■ Outline of operation

- The 16-bit data specified by [D] is rotated to the right (in the low bit direction) by the number of bits specified by [n].

Example of rotation 1 bit to the right



- When rotated to the right, the content of the bit that is 1 bit below the bit that moves to the least significant bit when rotated is stored in the CY flag (R9009). This bit is moved to the most significant bit as a result of rotation.
- For [n], only the lower 8 bits of the 16-bit data are valid.

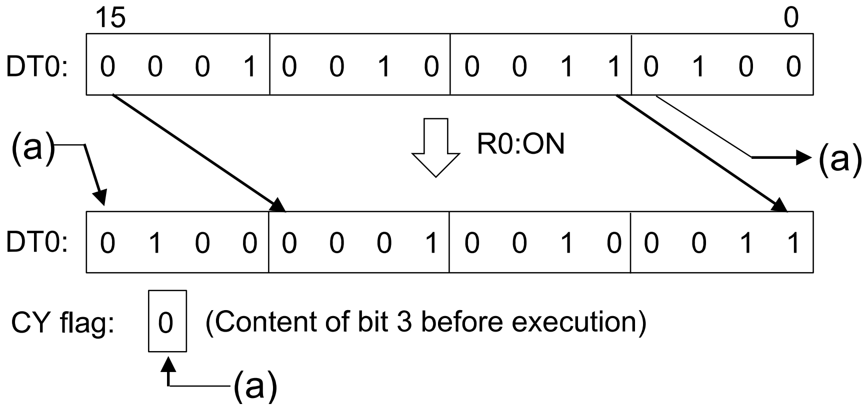


(Note 1) The bits marked with - are invalid.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of data register DT0 is rotated 4 bits to the right.



■ Precautions for programming

For the value of n, the operation is the same for every multiple of 16.

e.g.

When n = 16, the operation is the same as when n = 0 (the CY flag does not change either)

When n = 17, the operation is the same as when n = 1

When n = 32, the operation is the same as when n = 0 (the CY flag does not change either)

When n = 33, the operation is the same as when n = 1

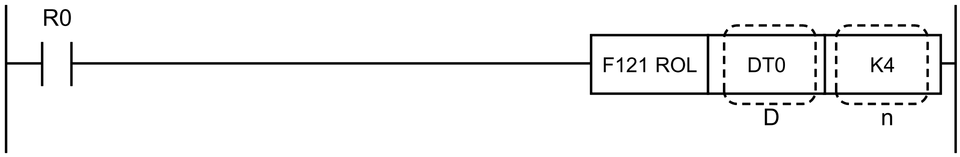
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Turns ON when the instruction is executed when the [n]th bit from the least significant bit is"1"before execution

17.2 F121 ROL (16-Bit Data Rotation to the Left)

Rotates the specified 16-bit data to the left.

■ Instruction format



■ Operands

Items	Settings
D	Area targeted for rotation
n	Area storing the number of bits specified to be rotated, or constant data

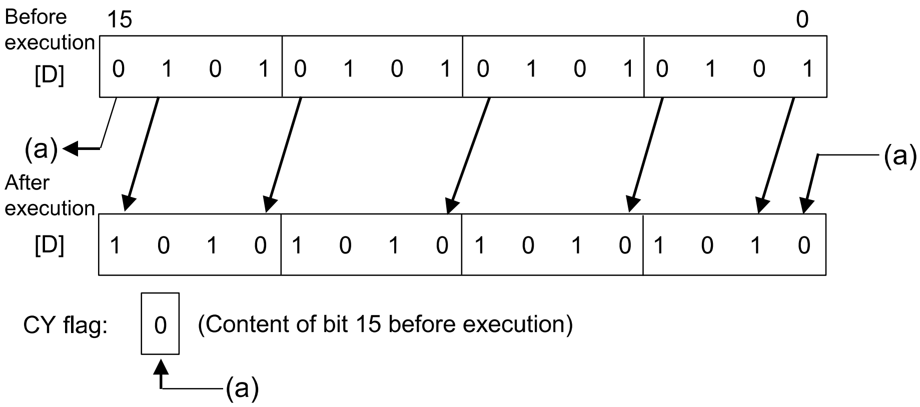
■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

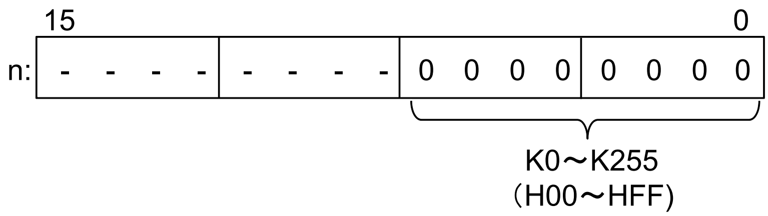
■ Outline of operation

- The 16-bit data specified by [D] is rotated to the left (in the high bit direction) by the number of bits specified by [n].

Example of rotation 1 bit to the left



- When rotated to the left, the content of the bit that is 1 bit above the bit that moves to the most significant bit when rotated is stored in the CY flag (R9009). This bit is moved to the least significant bit as a result of rotation.
- For [n], only the lower 8 bits of the 16-bit data are valid.

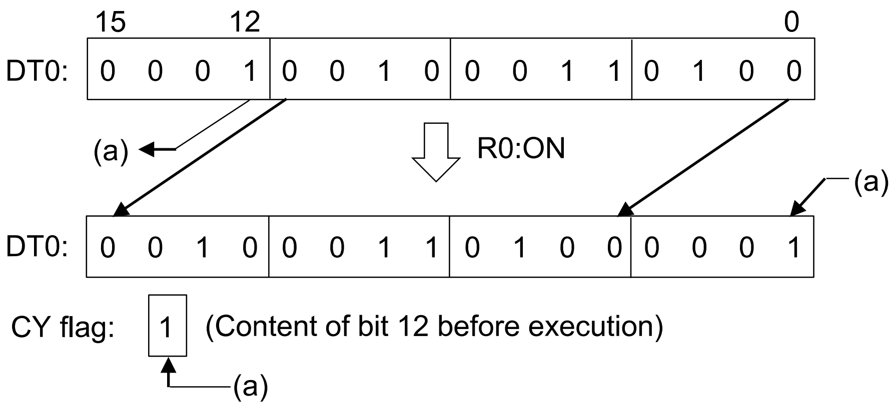


(Note 1) The bits marked with - are invalid.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of data register DT0 is rotated 4 bits to the left.



■ Precautions for programming

For the value of n, the operation is the same for every multiple of 16.

e.g.

When n = 16, the operation is the same as when n = 0 (the CY flag does not change either)

When n = 17, the operation is the same as when n = 1

When n = 32, the operation is the same as when n = 0 (the CY flag does not change either)

When n = 33, the operation is the same as when n = 1

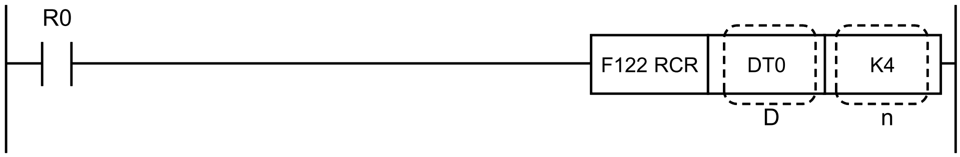
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Turns ON when the instruction is executed when the [n]th bit from the most significant bit is"1"before execution

17.3 F122 RCR (16-bit Data Right Rotation with Carry)

Rotate 17 bits of data made up of the specified 16-bit data and the carry flag to the right.

■ Instruction format



■ Operands

Items	Settings
D	Area targeted for rotation
n	Area storing the number of bits specified to be rotated, or constant data

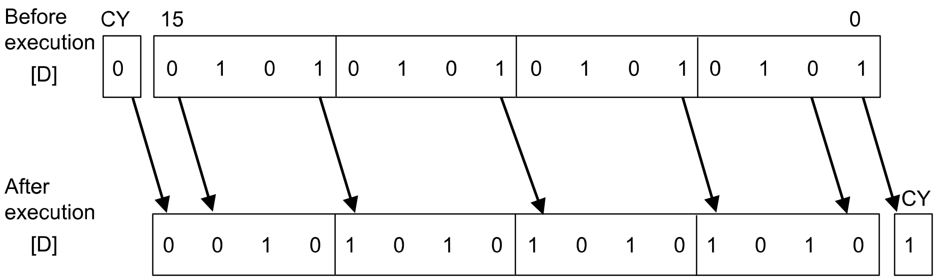
■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

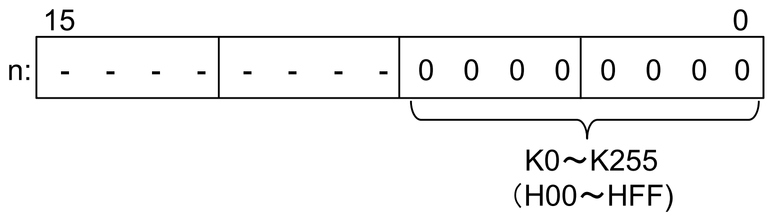
■ Outline of operation

- The 16-bit data specified by [D] is rotated to the right (in the low bit direction) by the number of bits specified by [n], including the CY (carry) flag (R9009).

Example of rotation 1 bit to the right



- When the data is rotated to the right,
  1. the content of the bit that is 1 bit lower than the bit that moves to the least significant bit when rotated is stored in the CY flag (R9009).
  2. The content of the CY flag (R9009) before the rotation is stored in the [n]th bit from the most significant bit.
- For [n], only the lower 8 bits of the 16-bit data are valid.

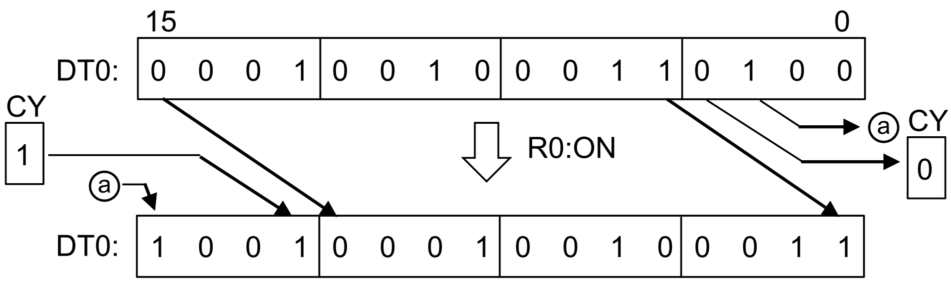


(Note 1) The bits marked with - are invalid.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of data register DT0 is rotated 4 bits to the right.
(The CY value immediately before execution is assumed to be 1.)



■ Precautions for programming

For the value of n, the operation is the same for every multiple of 17.
e.g.
When n = 17, the operation is the same as when n = 0
When n = 18, the operation is the same as when n = 1
When n = 34, the operation is the same as when n = 0
When n = 35, the operation is the same as when n = 1

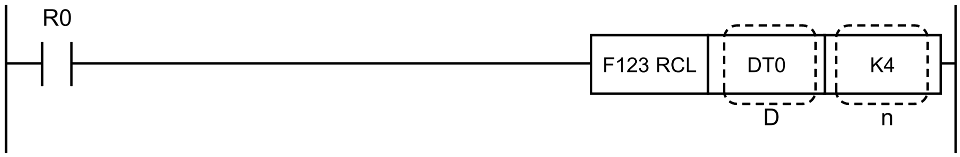
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Turns ON when the instruction is executed when the [n]th bit from the least significant bit is"1"before execution

17.4 F123 RCL (16-bit Data Left Rotation with Carry)

Rotates 17-bit data, consisting of specified 16-bit data with carry flag data added, to the left.

■ Instruction format



■ Operands

Items	Settings
D	Area targeted for rotation
n	Area storing the number of bits specified to be rotated, or constant data

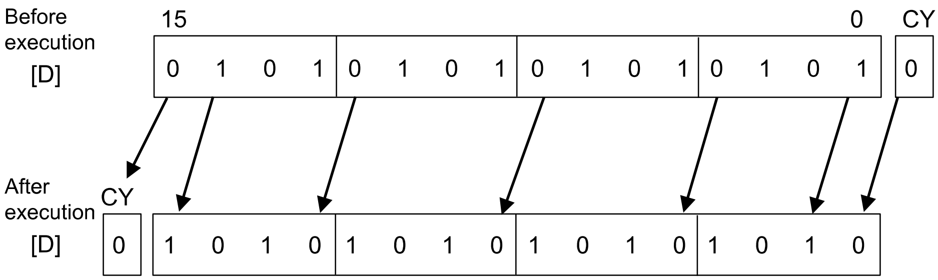
■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

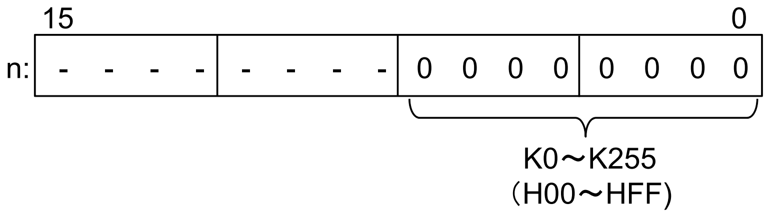
■ Outline of operation

- Rotates 16-bit data specified by [D], including CY (carry) flag (R9009) data, to the left (toward higher bits) by the number of bits specified by [n].

Example of rotation 1 bit to the left



- When the data is rotated to the left,
  1. the content of the bit that is 1 bit higher than the bit that moves to the most significant bit when rotated is stored in the CY flag (R9009).
  2. The content of the CY flag (R9009) before the rotation is stored in the [n]th bit from the least significant bit.
- For [n], only the lower 8 bits of the 16-bit data are valid.

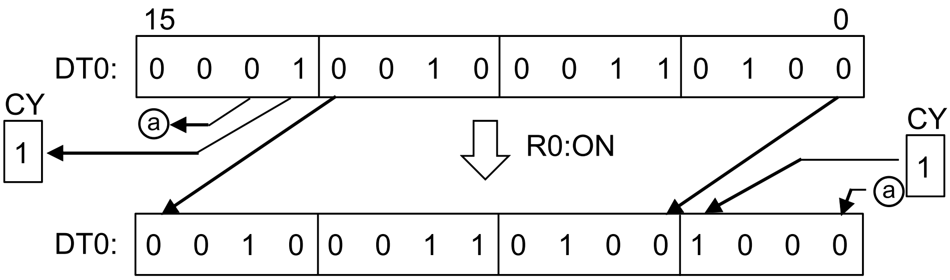


(Note 1) The bits marked with - are invalid.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of data register DT0 is rotated 4 bits to the left.
(The CY value immediately before execution is assumed to be 1.)



■ Precautions for programming

For the value of n, the operation is the same for every multiple of 17.

e.g.

When n = 17, the operation is the same as when n = 0

When n = 18, the operation is the same as when n = 1

When n = 34, the operation is the same as when n = 0

When n = 35, the operation is the same as when n = 1

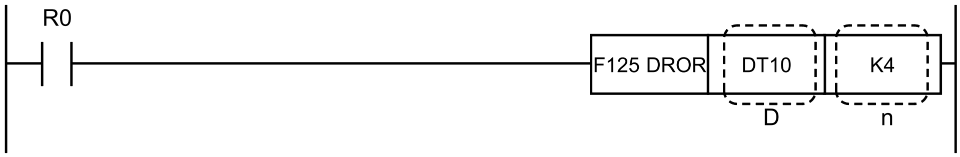
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Turns ON when the instruction is executed when the [n]th bit from the most significant bit is"1"before execution

17.5 F125 DROR [32-Bit Data Right Rotation]

Rotates "n" bits of 32-bit data (double word data) to the right.

■ Instruction format



■ Operands

Items	Settings
D	Area to be rotated (two words)
n	Area storing the number of bits specified to be rotated, or constant data

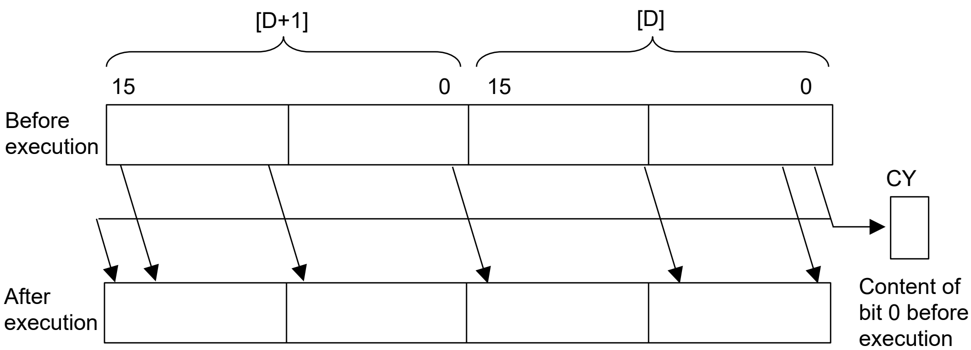
■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		●	●	●	●	●	●	●	●							●	
n	●	●	●	●	●	●	●	●	●	●	●	●	●			●	

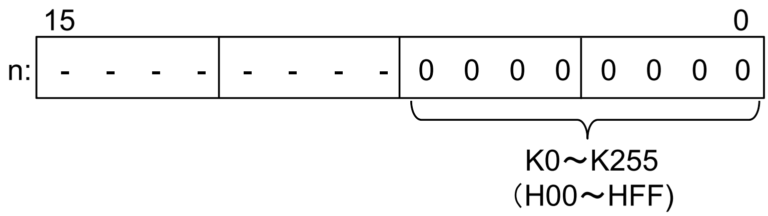
■ Outline of operation

- Rotates a number of bits specified by [n] of double word data specified by [D, D+1], to the right (toward lower bits).

Example of rotation 1 bit to the right



- When data is rotated to the right, the data which moves to 1 bit above the least significant bit position when rotation occurs is stored in the CY flag (R9009). This bit is moved to the most significant bit position as a result of the rotation.
- For [n], only the lower 8 bits of the 16-bit data are valid.



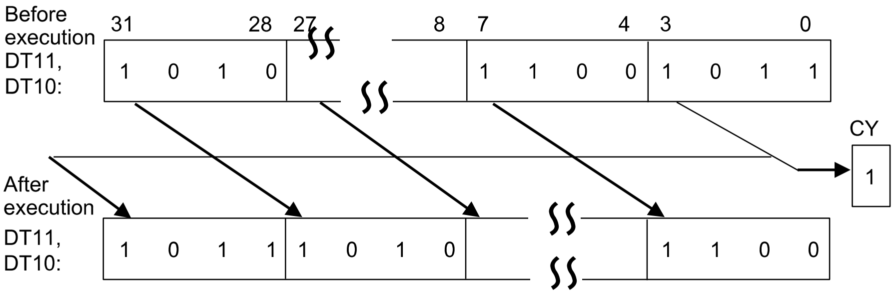
(Note 1) The bits marked with - are invalid.

- When [n] = K0, the contents of [D, D+1] and the CY flag do not change.

■ Operation example

Operation of instruction format description program

When the internal relay R0 turns ON, the contents of DT11 and DT10 are rotated 4 bits to the right. The content of bit 3 before execution is stored in the CY flag.



■ Precautions for programming

If n is a multiple of 32, this will result in the same operation as n = 0.

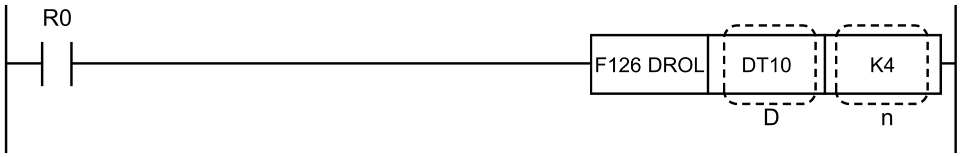
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Reflects the content of the nth bit from the least significant bit immediately before the instruction is executed.

17.6 F126 DROL (32-bit data left rotation)

Rotates 32-bit data (double word data) n bits to the left.

■ Instruction format



■ Operands

Items	Settings
D	Area to be rotated (two words)
n	Area storing the number of bits specified to be rotated, or constant data

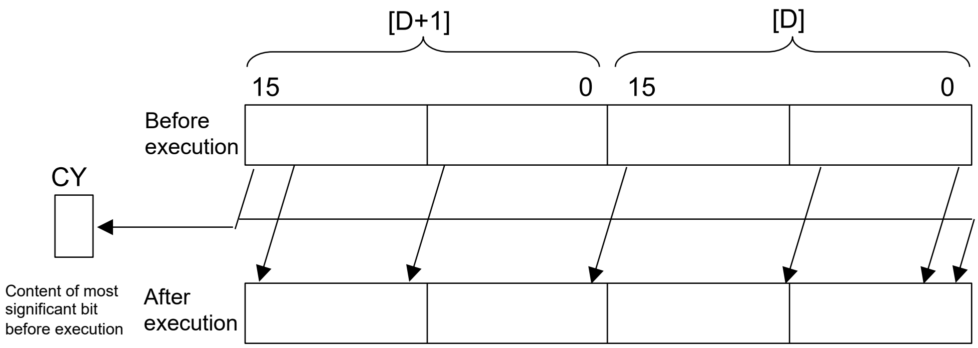
■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

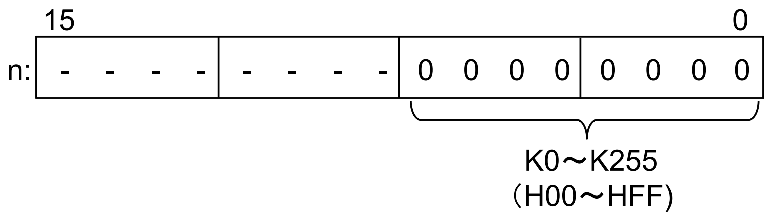
■ Outline of operation

- Rotates double word data specified by [D, D+1] a number of bits specified by [n] to the left (toward higher bits).

Example of rotation 1 bit to the left



- When rotated to the left, the content of the bit that is 1 bit above the bit that moves to the most significant bit when rotated is stored in the CY flag (R9009). After rotation, this bit moves to the least significant bit.
- For [n], only the lower 8 bits of the 16-bit data are valid.



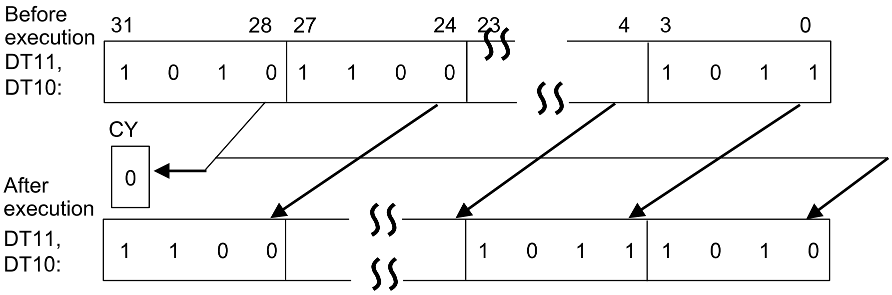
(Note 1) The bits marked with - are invalid.

- When [n]=K0, the contents of [D, D+1] and the CY flag do not change.

Operation example

Operation of instruction format description program

When the internal relay R0 turns ON, the contents of DT11 and DT10 are rotated 4 bits to the left. The CY flag stores the contents of bit 28 from before execution.



Precautions for programming

If n is a multiple of 32, this will result in the same operation as n=0.

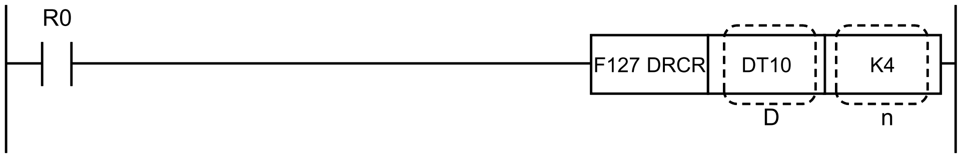
Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Reflects the content of the nth bit from the most significant bit immediately before the instruction is executed.

17.7 F127 DRCR (32-bit Data Right Rotation with Carry)

Rotates 32-bit data (double-word data) n bits to the right together with carry data.

■ Instruction format



■ Operands

Items	Settings
D	Area to be rotated (two words)
n	Area storing the number of bits specified to be rotated, or constant data

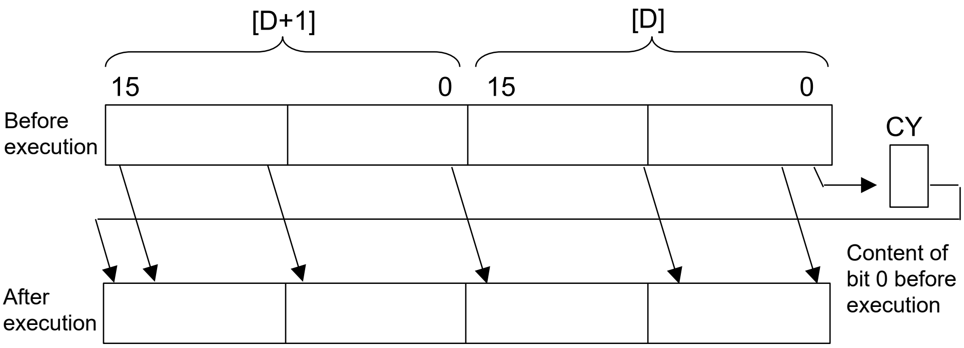
■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		●	●	●	●	●	●	●	●							●	
n	●	●	●	●	●	●	●	●	●	●	●	●	●			●	

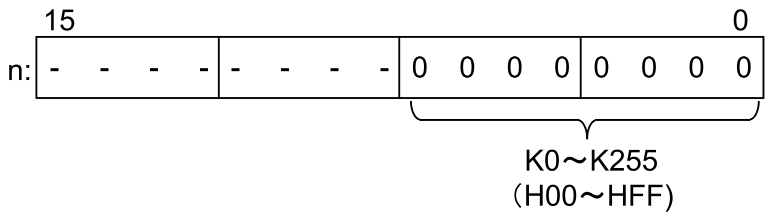
■ Outline of operation

- The double-word data specified by [D, D+1] is rotated to the right (in the low bit direction) by the number of bits specified by [n], including the CY (carry) flag (R9009).

Example of 1-bit right rotation (with carry)



- When the data is rotated to the right,
  1. the content of the bit that is 1 bit lower than the bit that moves to the least significant bit when rotated is stored in the CY flag (R9009).
  2. The content of the CY flag (R9009) before the rotation is stored in the [n]th bit from the most significant bit.
- For [n], only the lower 8 bits of the 16-bit data are valid.



(Note 1) The bits marked with - are invalid.

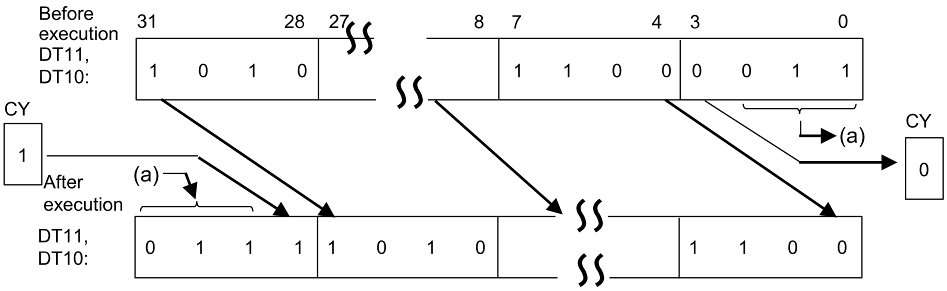
- When [n] = K0, the contents of [D, D+1] and the CY flag do not change.

■ Operation example

Operation of instruction format description program

When internal relay R10 turns ON, the contents of DT11 and DT10 are rotated 4 bits to the right.

The content of bit 3 before execution is stored in the CY flag. The content of the CY flag before execution is stored in bit 28.



■ Precautions for programming

When n = (a multiple of 33), the operation is the same as when n = 0.

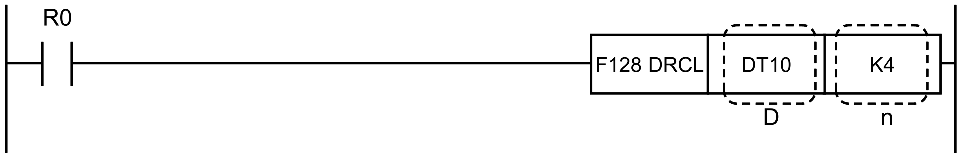
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Reflects the content of the nth bit from the least significant bit immediately before the instruction is executed.

17.8 F128 DRCL (32-bit Data Left Rotation with Carry)

Rotates 32-bit data (double-word data) n bits to the left with carry data.

■ Instruction format



■ Operands

Items	Settings
D	Area to be rotated (two words)
n	Area storing the number of bits specified to be rotated, or constant data

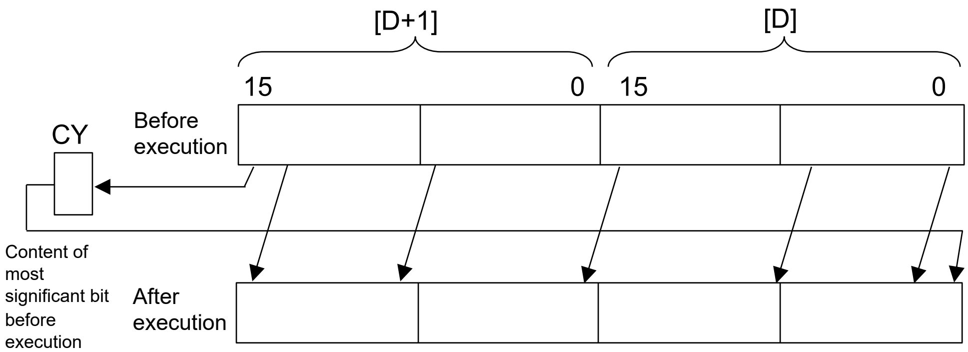
■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

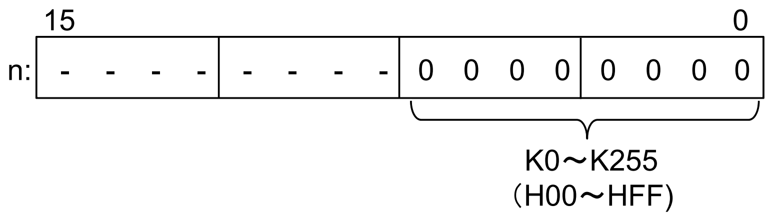
■ Outline of operation

- The double-word data specified by [D, D+1] is rotated to the left (in the high bit direction) by the number of bits specified by [n], including the CY (carry) flag (R9009).

Example of rotation 1 bit to the left (with carry data)



- When the data is rotated to the left,
  1. the content of the bit that is 1 bit higher than the bit that moves to the most significant bit when rotated is stored in the CY flag (R9009).
  2. The content of the CY flag (R9009) before the rotation is stored in the [n]th bit from the least significant bit.
- For [n], only the lower 8 bits of the 16-bit data are valid.



(Note 1) The bits marked with - are invalid.

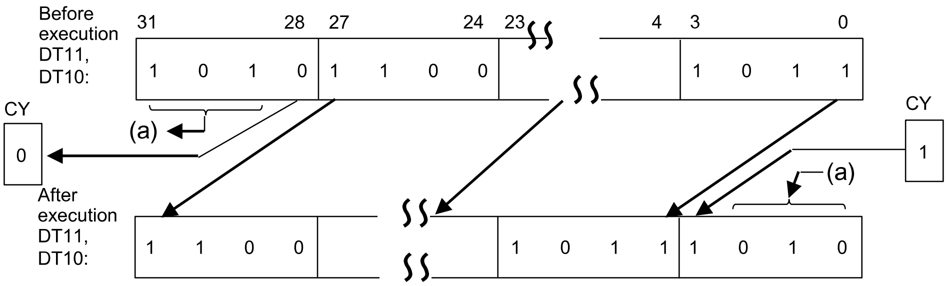
- When [n] = K0, the contents of [D, D+1] and the CY flag do not change.

■ Operation example

Operation of instruction format description program

When the internal relay R0 turns ON, the contents of DT11 and DT10 are rotated 4 bits to the left.

The CY flag stores the contents of bit 28 from before execution. The content of the CY flag before execution is stored in bit 3.



■ Precautions for programming

When n = (a multiple of 33), the same operation is the same as when n = 0.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	Reflects the content of the nth bit from the most significant bit immediately before the instruction is executed.

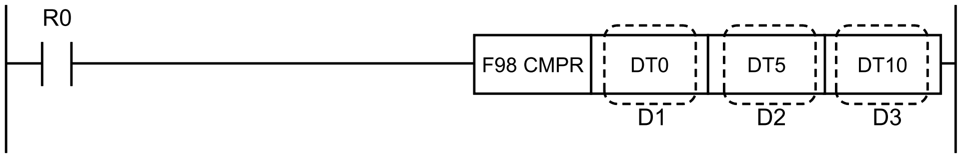
(MEMO)

18 Data Buffer Instruction

18.1 F98 CMPR (Compress Shift Read)

Reads the data at the highest address in the specified range and compresses the data upward.

■ Instruction format



■ Operands

Items	Settings
D1	Starting address of specified range
D2	Final address of specified range
D3	Area storing read data

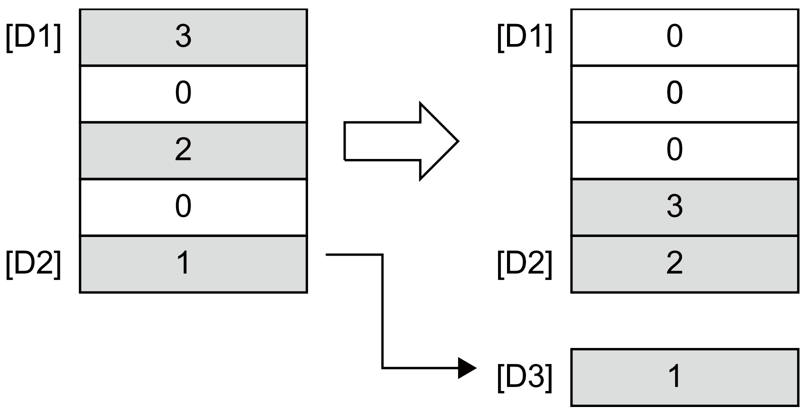
■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
D1		•	•	•	•	•	•	•								•	
D2		•	•	•	•	•	•	•								•	
D3		•	•	•	•	•	•	•								•	

■ Outline of operation

- In the area of the range specified by [D1] and [D2], the content of [D2] (highest address in the specified range) is transferred to the area specified by [D3].

  Non-zero data is shifted (compressed) in sequential order in the direction of the higher addresses in the specified range.



- The starting address of the area is specified by [D1] and the final address is specified by [D2].

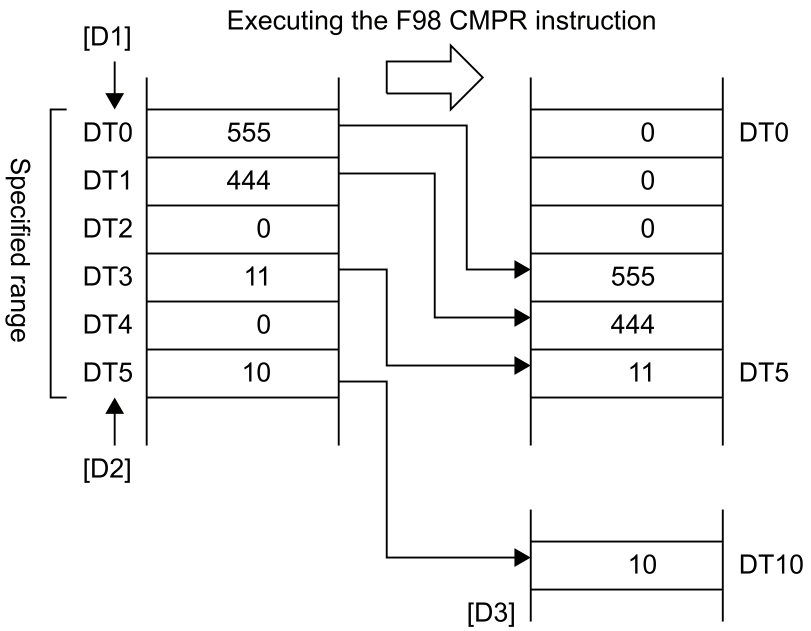
- Specify the same type of area for both [D1] and [D2]. Additionally, specify values so that [D1] is equal to or less than [D2].
- If all of the content in the range specified by [D1] and [D2] is 0, 0 is stored in [D3].

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of data register DT5 is transferred to data register DT10.

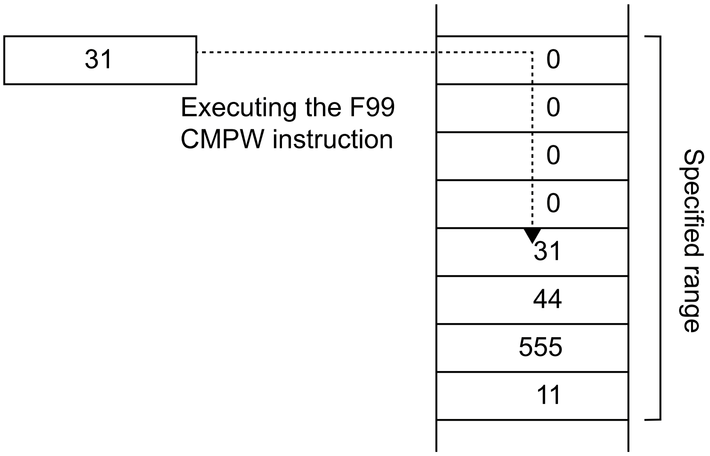
Additionally, the non-zero content in the range of DT0 to DT5 is stored in order from DT5. Any remaining content becomes"0".



■ Application example

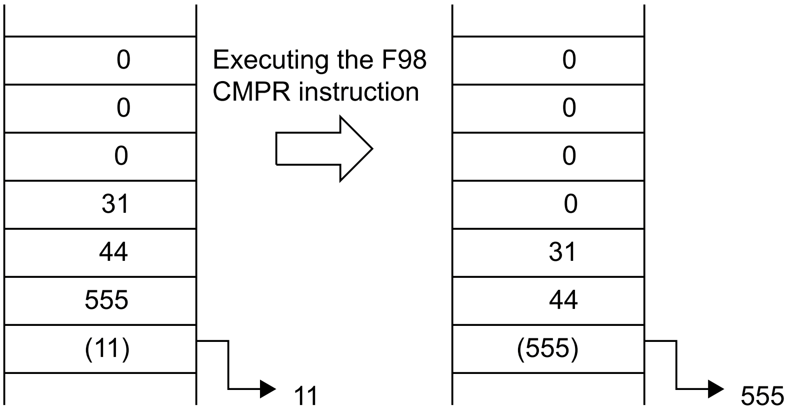
- This instruction can be combined with the "Compress shift Write" (F99 CMPW) instruction to use a memory area of any range as a buffer.
  1. Executing the F99 CMPW instruction

     When data is written to the starting address of the buffer (the area of the specified range), it accumulates in the buffer in sequential order. The oldest data will be at the final address of the buffer.

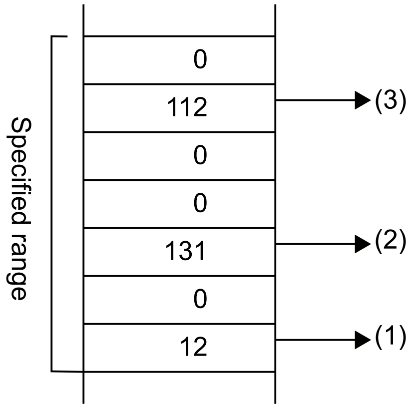


2. Executing the F98 CMPR instruction

   When data at the final address of the buffer (the area of the specified range) is read, data can be extracted in sequential order, starting from the oldest data. Any remaining data in the buffer is shifted in the direction of the higher addresses, so the oldest data at any point will always be stored at the final address.



- This can be used to extract valid non-zero data from data written in random order.



Each time the F98 CMPR instruction is executed, data is extracted in sequential order from (1) to (3).

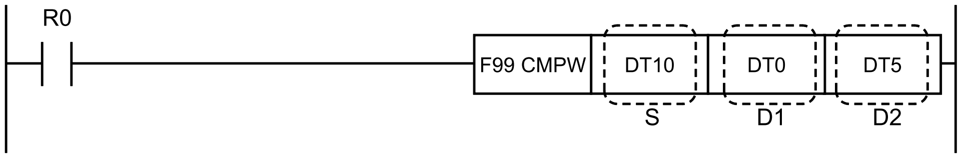
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when [D1] > [D2]
	Turns ON when [D1] and [D2] are not the same type of area

18.2 F99 CMPW (Compress Shift Write)

Writes data to the starting address in the specified range, and compresses the data upward.

■ Instruction format



■ Operands

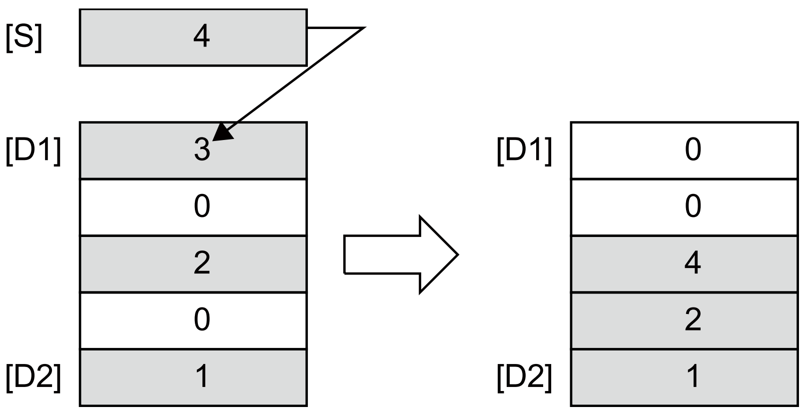
Items	Settings
S	Area storing the hexadecimal data or constant data
D1	Starting address of specified range
D2	Final address of specified range

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D1		•	•	•	•	•	•	•								•	
D2		•	•	•	•	•	•	•								•	

■ Outline of operation

- In the area of the range specified by [D1] and [D2], the content of the area specified by [S] is transferred to [D1] (starting address in the specified range).
  Non-zero data is shifted (compressed) in sequential order in the direction of the higher addresses in the specified range.



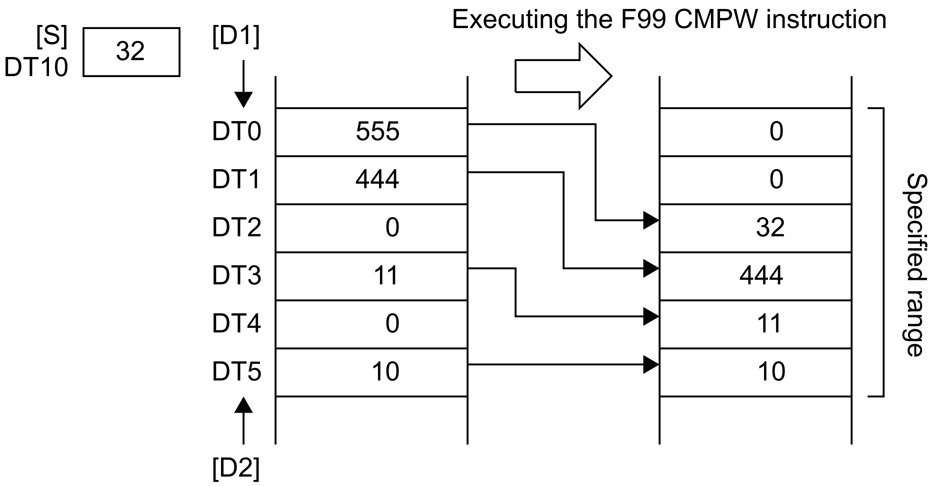
- The starting address of the area is specified by [D1] and the final address is specified by [D2].

- Specify the same type of area for both [D1] and [D2]. Additionally, specify values so that [D1] is equal to or less than [D2].
- If the content of [S] is 0, only a compressed shift is carried out.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the content of data register DT10 is transferred to data register DT0. Additionally, the non-zero content in the range of DT0 to DT5 is stored in order from DT5. Any remaining content becomes"0".

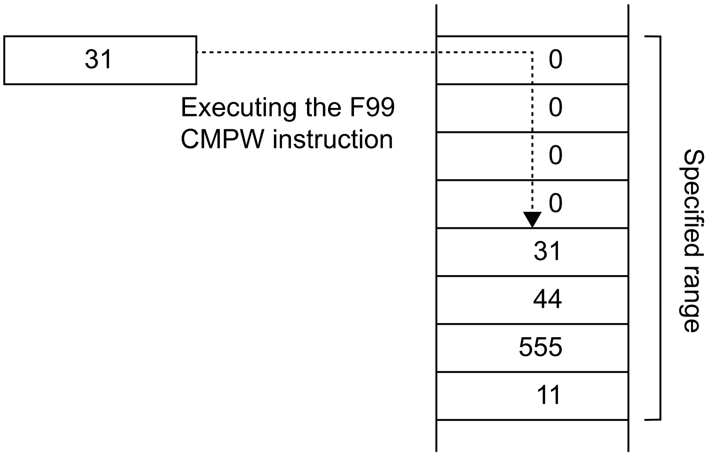


(Note 1) Because the content of [S] is written to DT0 first, the original content of DT0 (555 for example) is overwritten.

■ Application example

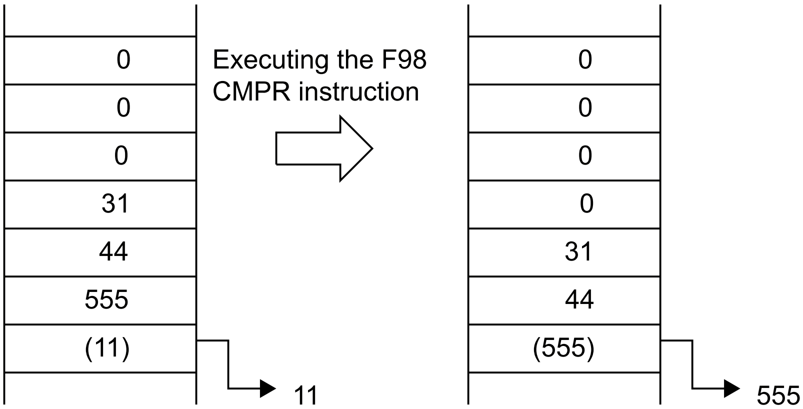
- This instruction can be combined with the "Compress shift read" (F98 CMPR) instruction to use a memory area of the specified range as a buffer.
  1. Executing the F99 CMPW instruction

     When data is written to the starting address of the buffer (the area of the specified range), it accumulates in the buffer in sequential order. The oldest data will be at the final address of the buffer.

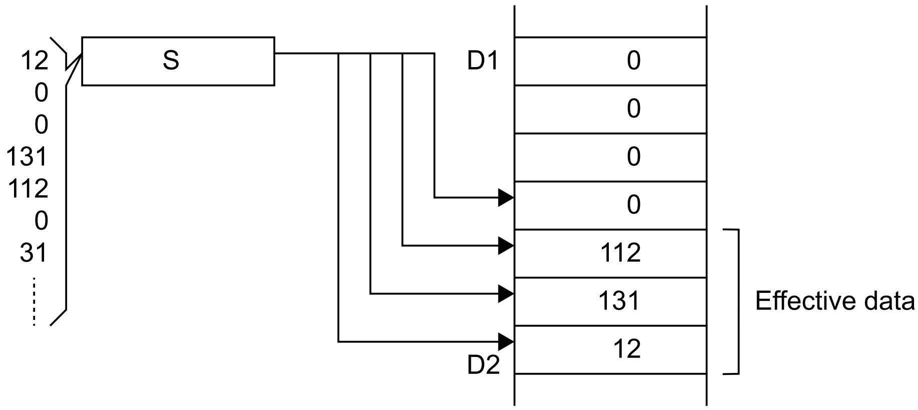


2. Executing the F98 CMPR instruction

   When data at the final address of the buffer (the area of the specified range) is read, data can be extracted in sequential order, starting from the oldest data. Any remaining data in the buffer is shifted in the direction of the higher addresses, so the oldest data at any point will always be stored at the final address.



- This can be used to extract valid non-zero data from data written in random order.



Executing the F99 CMPW instruction causes only the valid data to be stored.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when [D1] > [D2]
	Turns ON when [D1] and [D2] are not the same type of area

18.3 How to Use the FIFO (First-in First-out) Buffer

The FIFO buffer is a buffer area that stores data in the order it is written, and starts reading in order from the first data stored. It is convenient to use the FIFO buffer as a record of the order of objects on a conveyor line or buffer line.

1 2 Procedure

1. The F115 FIFT instruction defines the area to be used as the FIFO buffer. (Use it just once before read/write.)

2. Use the F117 FIFW instruction for data write, and the F116 FIFR instruction for read.

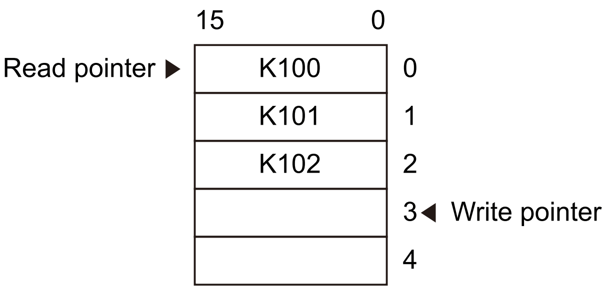
Data write

- After data is written, it is stored in the data storage area in order starting from the first written data. The write pointer indicates the next write area.
- When the data storage area becomes full, it is no longer possible to write.

Data read

- When read is executed, data is transferred in order from the first data that was stored. The read pointer indicates the area that will be read.
- If read is executed when there is no data written to the data storage area, an error is returned.

<Example of data storage area>

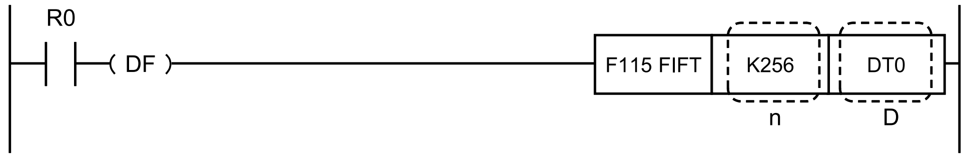


As shown in the figure above, when data is written, it is stored in area"3". The write pointer moves to"4". (Data will next be written to"4".) When a read is executed, data is read from the"0"area. The read pointer moves to"1". (Data will next be read from"1").

18.4 F115 FIFT (FIFO Buffer Definition)

Defines the start and size of the FIFO buffer area.

■ Instruction format



■ Operands

Items	Settings
n	Area storing the size (number of words) of the FIFO buffer, or constant data
D	Starting address for the FIFO buffer area

■ Devices that can be specified (indicated by •)

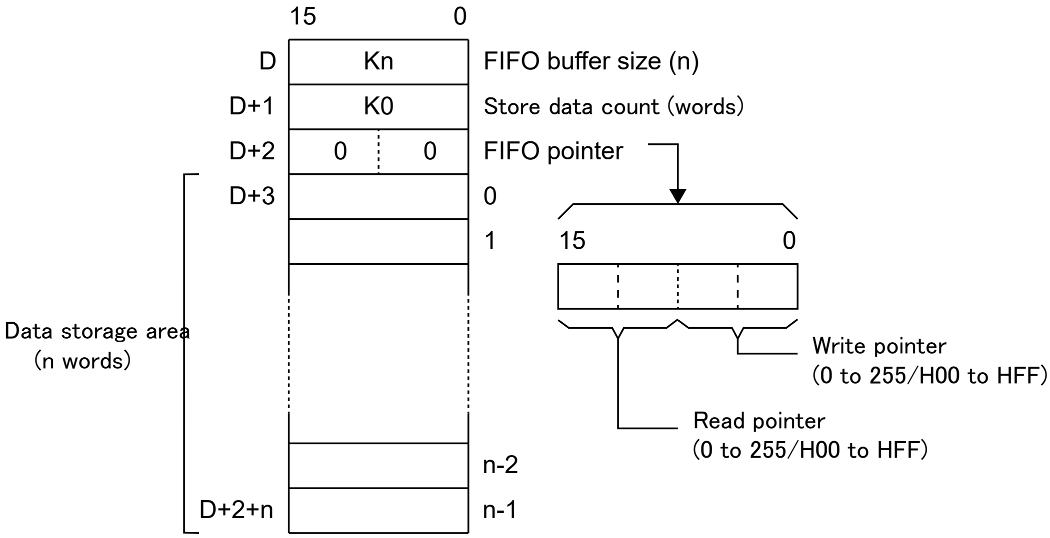
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•								•	

■ Outline of operation

- The area used as the FIFO buffer is defined. A data storage area of n words (n = K1 to K256) is defined for the area specified by [D].

Definition of the area using the F115 FIFT instruction should be executed only once, before writing to or reading from the FIFO buffer. Normally, reading and writing are disabled while this instruction is being executed.

- When the F115 FIFT instruction is executed, the FIFO buffer area is defined as follows.

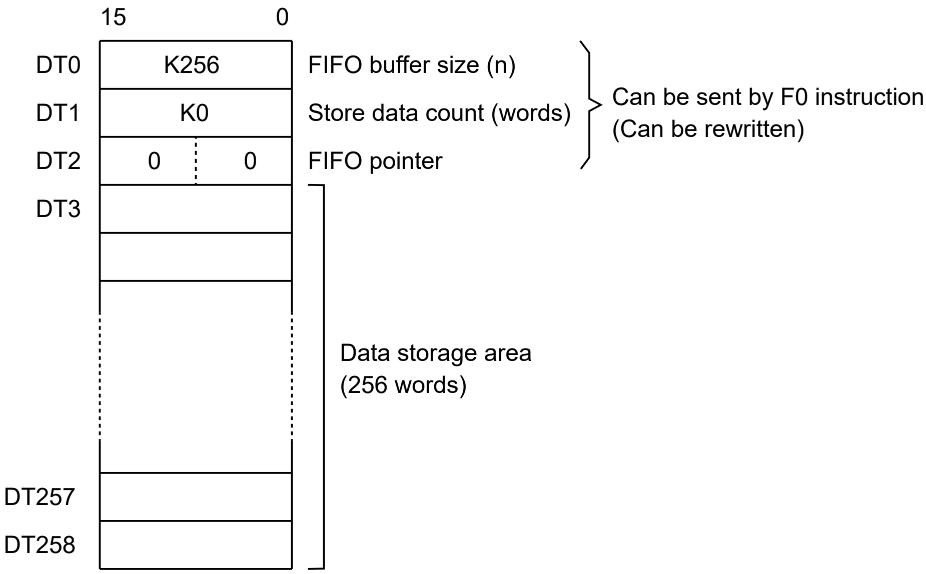


- When the F115 FIFT instruction is executed, the following are stored as default values: [D] = n (the value specified by the F115 FIFT instruction), [D+1] = K0 and [D+2] = H0000.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the area starting from DT0 is defined as the FIFO buffer area."FIFO buffer size"(K256) is stored in DT0,"number of data items"is stored in DT1 (with a default value of K0), and"FIFO pointer"(with a default value of H0000) is stored in DT2. When n = K256, the 256 words from DT3 to DT258 are defined as the data storage area.



■ Flag operations

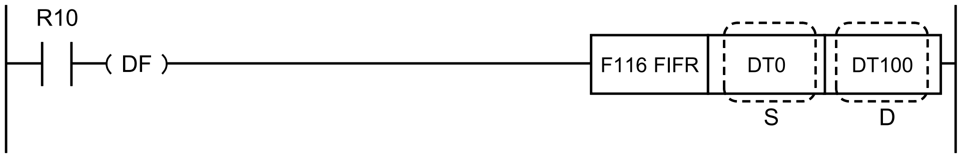
Name	Description
R9007	Turns ON when the area is exceeded in index modification.

Name	Description
R9008 (ER)	Turns ON when n = 0
	Turns ON when n > 256
	Turns ON when the final address of the FIFO set according to the FIFO size exceeds the area

18.5 F116 FIFR (FIFO Data Read)

Reads the data from the specified FIFO buffer.

■ Instruction format



■ Operands

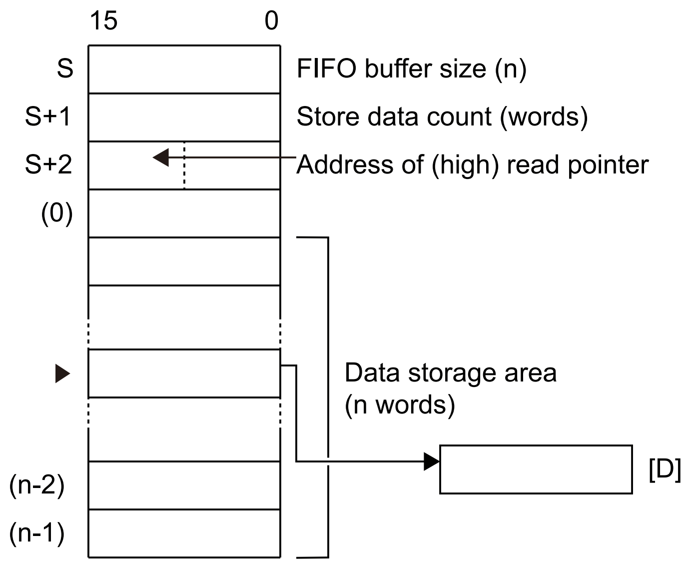
Items	Settings
S	Starting address for the FIFO buffer area
D	Area storing the data read from the FIFO buffer

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S		●	●	●	●	●	●	●								●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The data is read from the FIFO buffer at the start of the area specified by [S], and is stored in the area specified by [D]. For [S], specify the start of the FIFO buffer defined by the FIFT instruction.
- Data is read from the address specified by the read pointer when the instruction is executed.



(Note 1) (0) to (n–1) are addresses assigned to the data storage areas.

(Note 2) n is the value specified by the F115 FIFT instruction.

(Note 3) ► is the read pointer.

- The read pointer is stored in the upper eight bits of the third word of the FIFO buffer area. It is indicated by an address in the data storage area. The actual address is the starting address of the FIFO buffer area specified by [S], plus 3, plus the read pointer value (in which only the upper byte is a decimal value).
- When a read is executed, 1 is subtracted from the number of stored data and the read pointer is incremented by 1.

Note

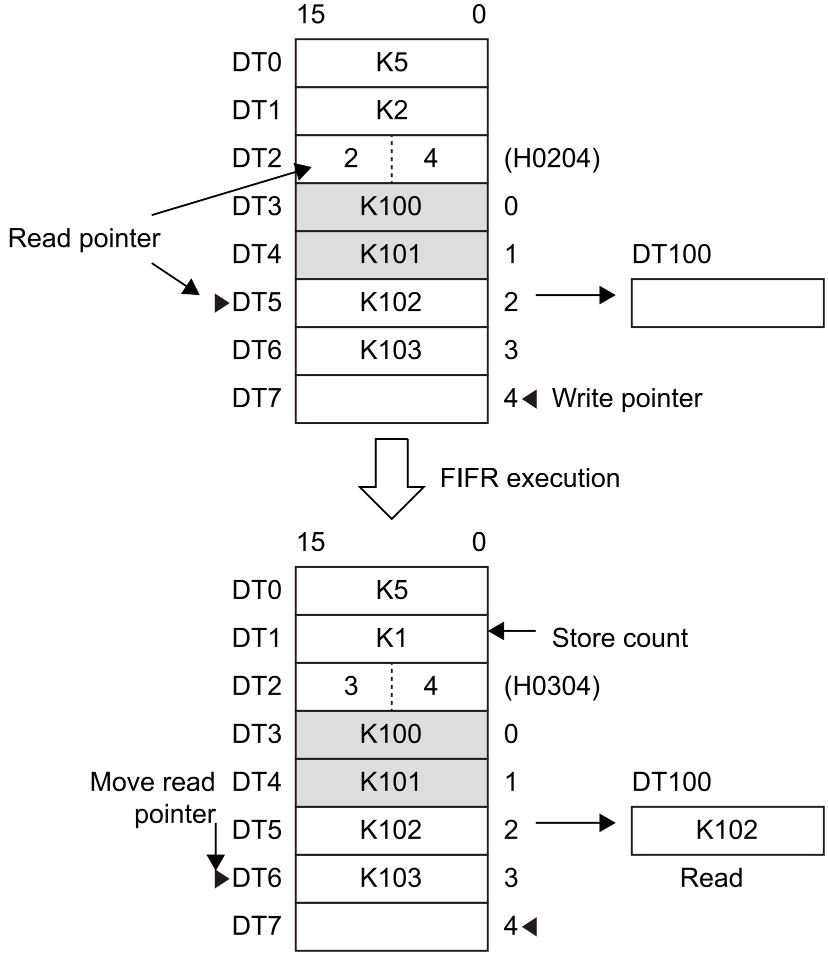
- An error occurs if the instruction is executed when the number of stored data is 0. No data set for [D].
- A read is only performed when the read pointer is not equal to the write pointer.
- If this instruction is executed while the read pointer is pointing to the ending address of the FIFO buffer (n defined by the F115 FIFT instruction minus 1), the read pointer becomes 0.

■ Operation example

Operation of instruction format description program

When internal relay R10 turns ON, data is read from the FIFO buffer area at the start of DT0 and stored in DT100.

[When the read pointer is 2]



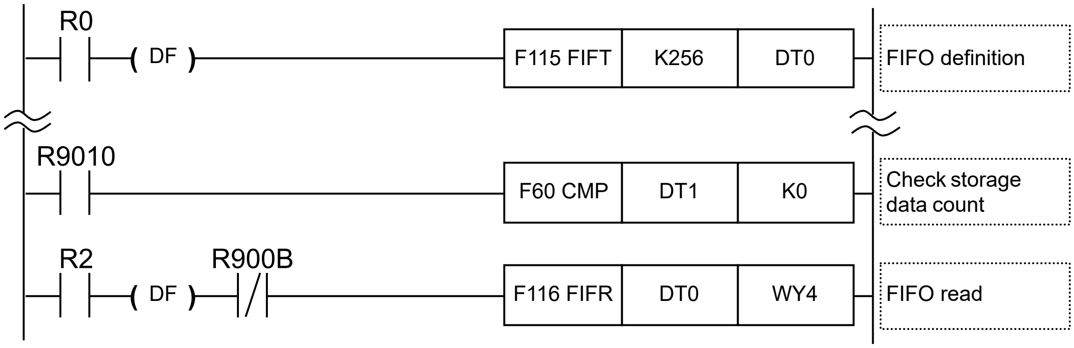
1. The content of DT5 indicated by read pointer 2 is transferred to DT100.
2. After reading, 1 is subtracted from the content of DT1 (number of stored data), and the read pointer moves to 3. (The next time a read is executed, the content of DT6 indicated by 3 is transferred to DT100.)

■ Precautions for programming

An error occurs if the F116 FIFR instruction is executed when the number of stored data ([S+1]) is 0.

[Reference]

In the program below, the F116 FIFR instruction is not executed when the data storage number is 0.



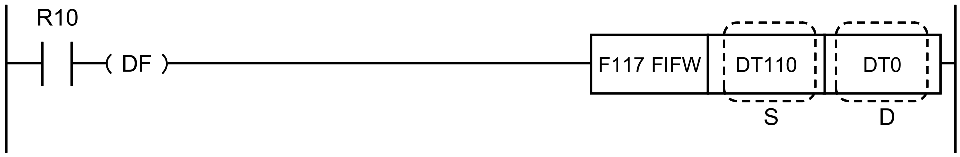
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the size of the FIFO specified by [S] (n) is n = 0 or n > 256
	Turns ON when the number of data stored in the FIFO is 0
	Turns ON when the number of stored data items of the FIFO is larger than the FIFO size (n)
	Turns ON when the final address of the FIFO based on the FIFO size (n) exceeds the area
	Turns ON when the FIFO read pointer is larger than the size of the FIFO (n)
	Turns ON when, after reading data, the FIFO read pointer is K256 (H100) or higher

18.6 F117 FIFW (FIFO Data Write)

Writes data to the specified FIFO buffer.

■ Instruction format



■ Operands

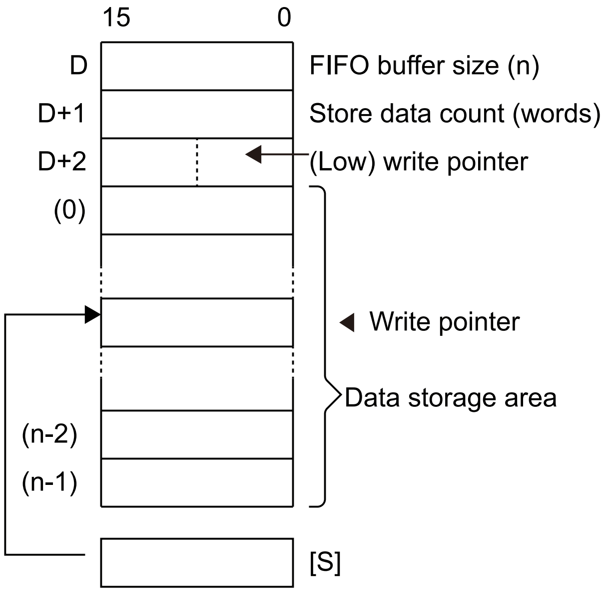
Items	Settings
S	Area storing the 16-bit data to write to the FIFO buffer, or constant data
D	Starting address for the FIFO buffer area

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●								●	

■ Outline of operation

- The 16-bit data specified by [S] is stored in the FIFO buffer starting at the area specified by [D]. Specify the start of the FIFO buffer defined by the FIFT instruction for [D].
- The specified data is written to the address indicated by the write pointer when the instruction is executed.



(Note 1) (0) to (n–1) are addresses assigned to the data storage areas.

(Note 2) n is the value specified by the F115 FIFT instruction.

- The write pointer is stored in the lower eight bits of the third word of the FIFO buffer area. It is indicated by a relative position in the data storage area. The actual address is the starting address of the FIFO buffer area specified by [D], plus 3, plus the write pointer value (in which only the lower byte is a decimal value).
- When a write is executed, 1 is added to the number of stored data items, and the write pointer is incremented 1.

Note

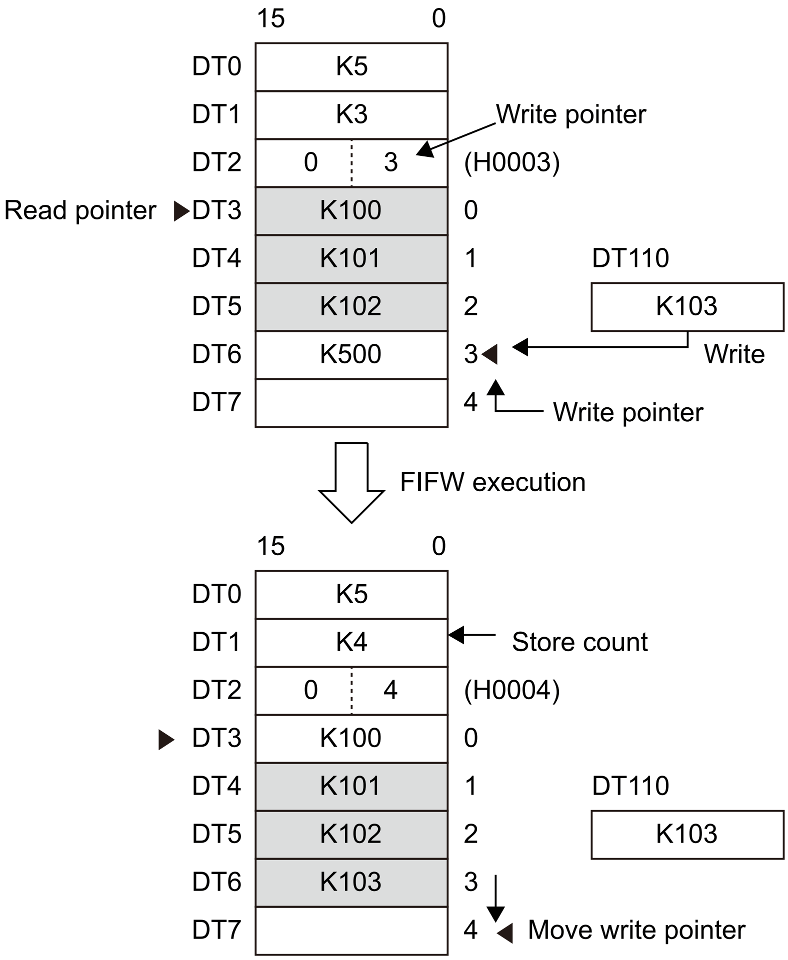
- An error occurs if this instruction is executed when the FIFO buffer is full (the number of stored data items = size n of the FIFO defined by the F115 FIFT instruction). In this case, the write is not performed.
- If this instruction is executed when the write pointer is indicating the final address in the FIFO buffer (the n value defined by the F115 FIFT instruction), the write pointer will be set to 0.

■ Operation example

Operation of instruction format description program

When internal relay R10 turns ON, the contents of DT110 are written to the FIFO buffer area that starts from by DT0.

When the write pointer is 3

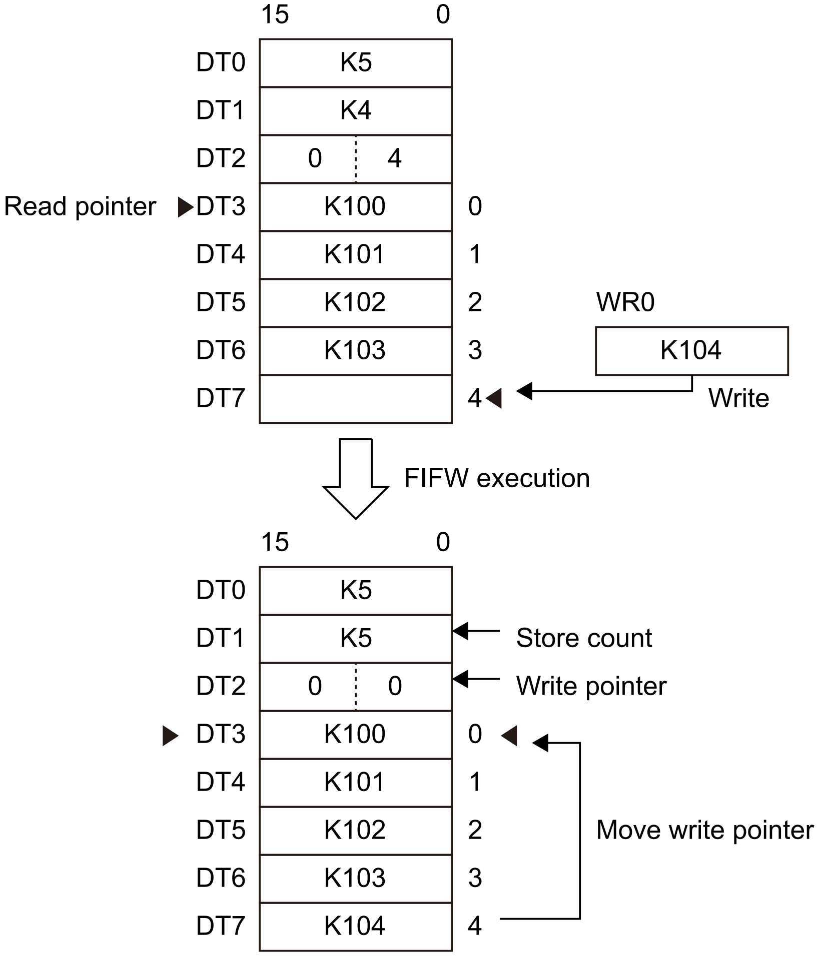


1. The contents"103"of DT110 are sent to DT6, which is indicated by pointer 3.
2. After the data has been written, 1 is added to the contents of DT1 (the number of stored data items), and the write pointer moves to 4. (The next time that writing is executed, the contents of DT110 are written to DT7, which is indicated by 4.)

■ Precautions when using this instruction

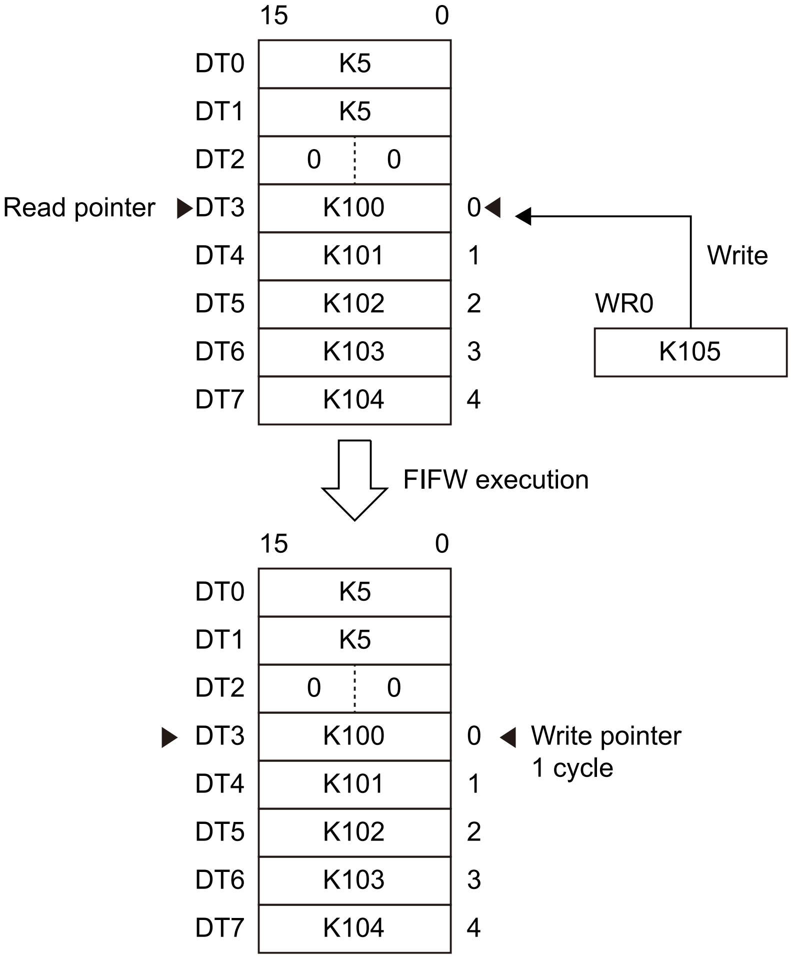
If data is received that exceeds the capacity of the buffer, an operation error occurs.

Example: If the write pointer is at the end of the FIFO buffer



When the F117 FIFW instruction is executed, after data is written to the final address (4) in the buffer, the write pointer becomes the starting address (0).

Example: When the write pointer has made one complete cycle

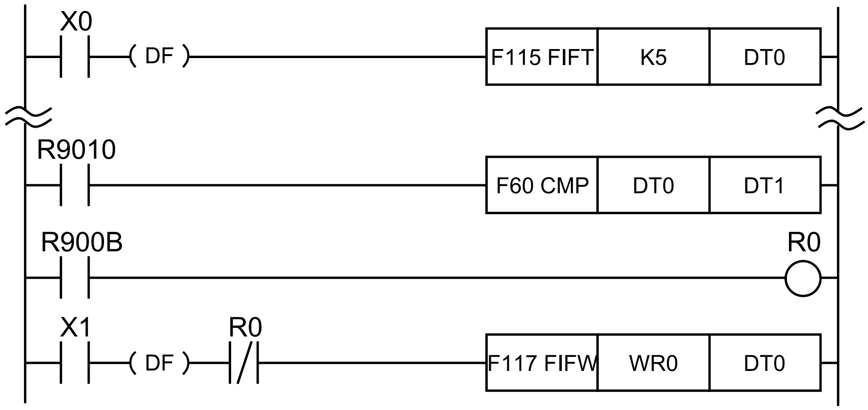


An error occurs and processing is not carried out.

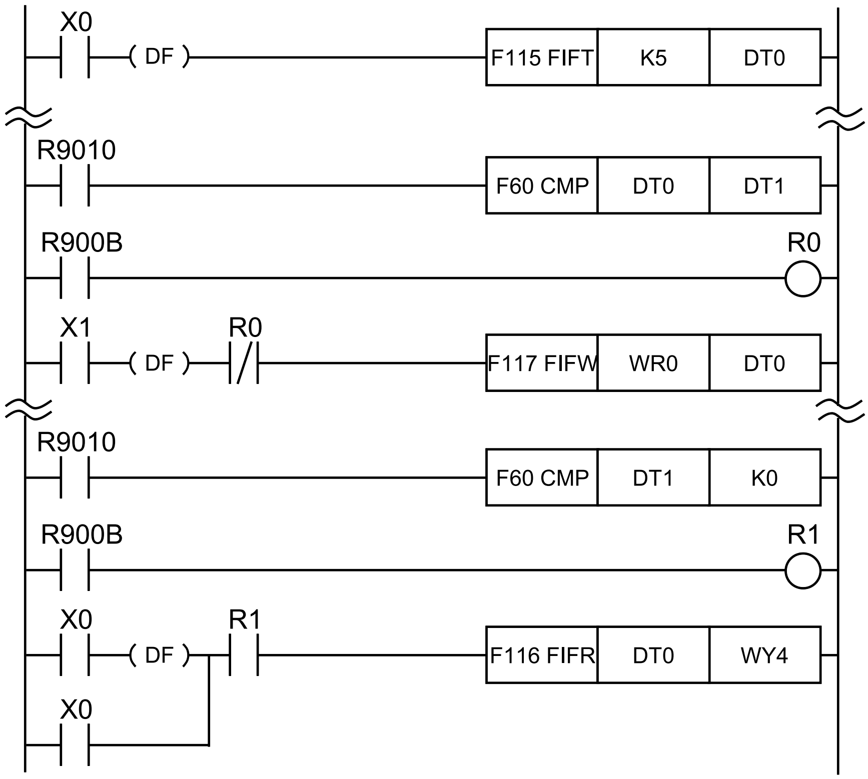
Because the number of data items stored in the FIFO buffer (DT1 = 5) exceeds the size of the FIFO buffer (DT0 = 5), the operation is not executed, and an operation error occurs.

■ Measures to avoid operation errors

1. Do not execute the F117 FIFW instruction using the comparison instruction. Avoid executing the F117 FIFW instruction when the size of the FIFO buffer (DT0) is equal to the number of data items stored in the buffer (DT1).



2. Execute the F117 FIFW instruction after executing the F116 FIFR instruction.



■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the size (n) of the FIFO specified by [D] is n = 0, or when n > 256
	Turns ON when the number of stored data items of the FIFO is larger than the FIFO size (n)
	Turns ON when the final address of the FIFO based on the FIFO size (n) exceeds the area

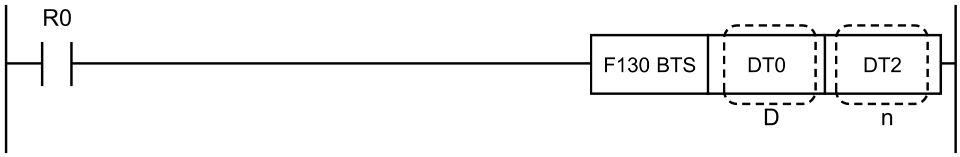
Name	Description
	Turns ON when the write pointer of the FIFO is larger than the FIFO size (n)
	Turns ON when the FIFO write pointer is K256 (H100) or higher after the data is written

19 Bit Manipulation Instructions

19.1 F130 BTS (Specified Bit Set)

Turns a bit of the specified 16-bit data ON.

■ Instruction format



■ Operands

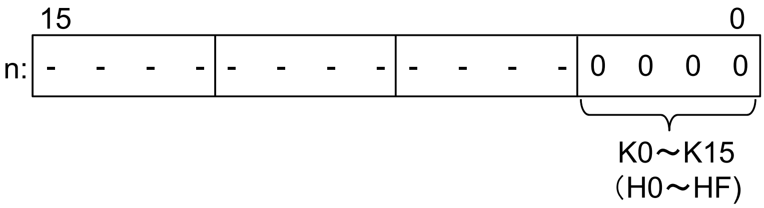
Items	Settings
D	Area in which bit is to be set
n	Area storing position of bit to be set, or constant data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

- The bit with the number specified by [n] in the 16-bit data specified by [D] is turned ON. Bits other than the specified bit do not change.
- Set [n] in the range from K0 to K15. Only the lower 4 bits of the 16-bit data are valid.

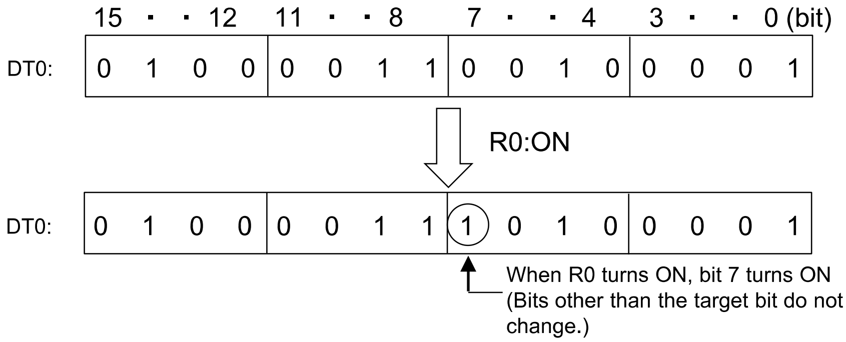


(Note 1) The bits marked with - are invalid.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the bit specified by DT2 in the data stored in DT0 is turned ON. When DT2 = K7, the operation is as shown below.



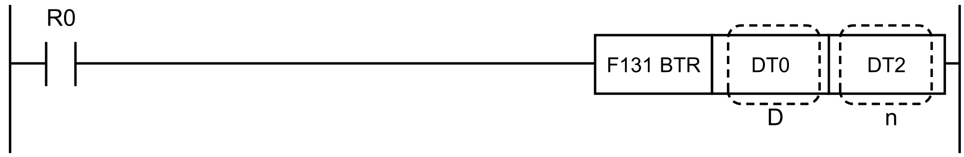
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

19.2 F131 BTR (Specified Bit Reset)

Turns OFF a specified bit of 16-bit data.

■ Instruction format



■ Operands

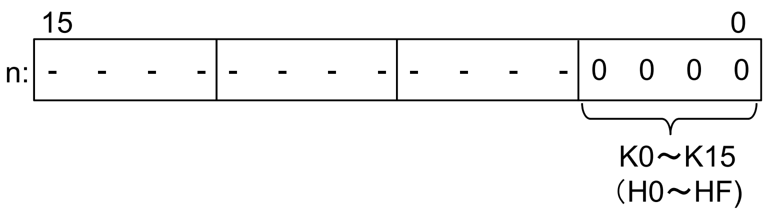
Items	Settings
D	Area where the bit will be reset
n	Area storing the specification of the bit position to be reset, or constant data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

- Turns OFF a bit specified by the number [n] in the 16-bit data specified by [D]. Bits other than the specified bit do not change.
- Set [n] in the range from K0 to K15. Only the lower 4 bits of the 16-bit data are valid.

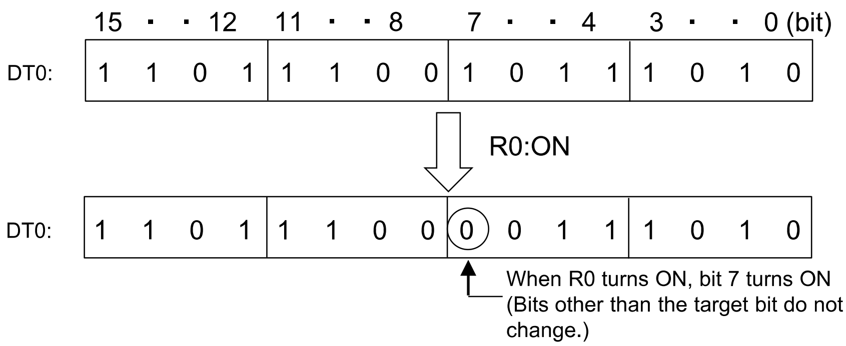


(Note 1) The bits marked with - are invalid.

■ Operation example

Operation of instruction format description program

Turns OFF the bit specified by DT2 in the data stored in DT0 when internal relay R0 turns ON. When DT2 = K7, the operation is as shown below.



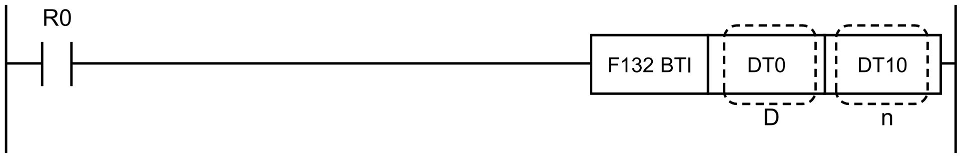
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

19.3 F132 BTI (Specified Bit Inversion)

Inverts a specific bit in 16-bit data.

■ Instruction format



■ Operands

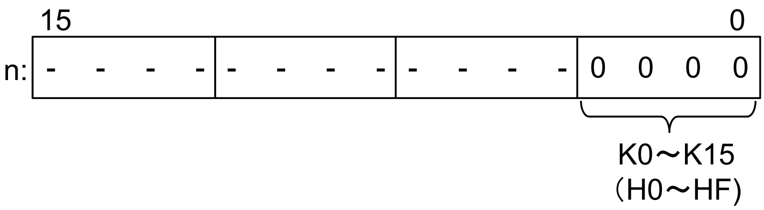
Items	Settings
D	Target area for bit inversion
n	Area storing the number of the bit to be inverted, or constant data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

- Inverts (OFF -> ON or ON -> OFF) the bit number specified by [n] in the 16-bit data specified by [D]. Bits other than the specified bit do not change.
- [n] is in the range of K0 to K15. Only the lower 4 bits of the 16-bit data are valid.

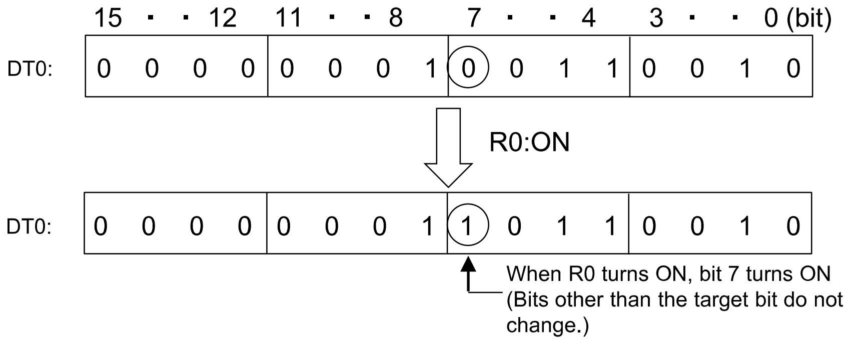


(Note 1) The bits marked with - are invalid.

■ Operation example

Operation of instruction format description program

Inverts the bit specified by DT10 in data stored in DT0 when internal relay R0 turns ON. When DT10 = K7, the operation is as shown below.



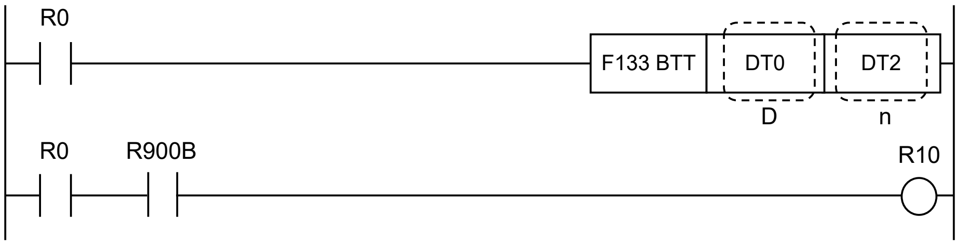
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

19.4 F133 BTT (Specified Bit Test)

Tests the specified bit in the specified 16-bit data (to determine whether it is ON or OFF).

■ Instruction format



■ Operands

Items	Settings
D	Target area for bit test
n	Area storing the numbers of the bits to be tested, or constant data

■ Devices that can be specified (indicated by •)

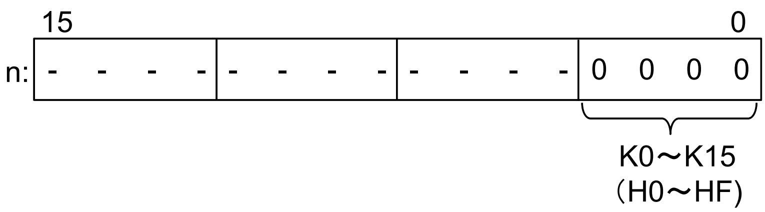
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

- The bit with the number specified by [n] in the 16-bit data specified by [S] is judged to either be ON or OFF, and the judgment result is output to special internal relay R900B ["=(ZERO)"flag].
- The judgment result is as follows.

State of specified bit	"=(ZERO)" flag (R900B)
ON (1)	OFF (0)
OFF (0)	ON (1)

- [n] can be specified in the range of K0 to K15. Only the lower 4 bits of the 16-bit data are valid.



(Note 1) The bits marked with - are invalid.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the bit specified by DT2 in the data stored in data register DT0 is determined to either be ON or OFF. If the specified bit is OFF, internal relay R10 turns ON. If DT2 = K7, then the following happens.

	15	·	·	12	11	·	·	8	7	·	·	4	3	·	·	0 (bit)
DT0:	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0

Test bit 7.

As bit 7 is OFF (0), R900B: ON (test result), so R10: ON

■ Precautions when using the judgment flag (R900B) twice or more

- The judgment flag R900B is updated each time an operation instruction or comparison instruction is executed.
- Accordingly, when using the judgment flag twice or more,
  1. the program using the judgment flag should be inserted immediately after the instruction that executes the judgment; and
  2. the flag should be output to an output relay or internal relay for each instruction.

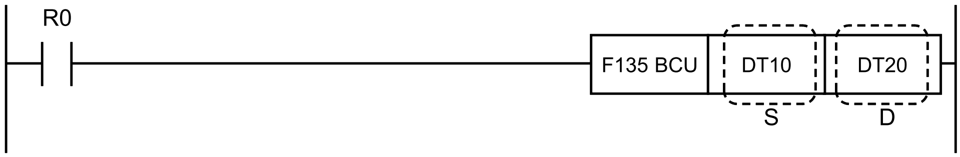
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the test bit (bit n) is"0"

19.5 F135 BCU (Count ON Bits in 16-bit Data)

Counts the number of ON bits in the specified 16-bit data.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the 16-bit data subject to the bit count, or constant data
D	Area storing the number of ON bits

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

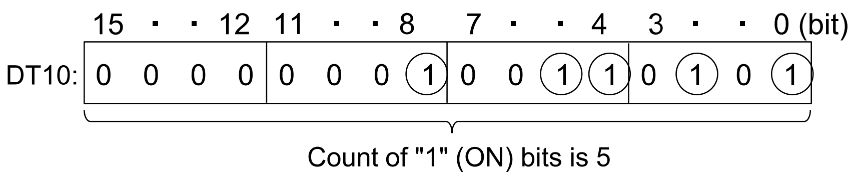
■ Outline of operation

- The number of ON bits (bits with a value of 1) in the 16-bit data specified by [S] is counted, and the result is stored in the area specified by [D].
- The result is stored as a decimal number.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the number of ON bits in the data stored in DT10 is stored in DT20.



When R0 turns ON, K5 is stored in DT20.

■ Flag operations

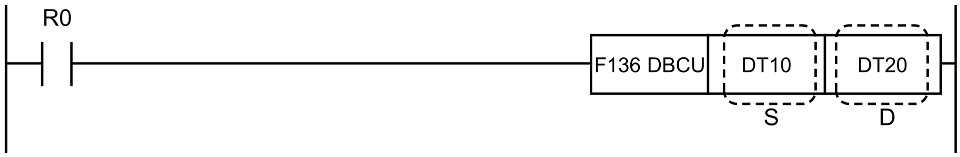
Name	Description
R9007 R9008	Turns ON when the area is exceeded in index modification.

Name	Description
(ER)	

19.6 F136 DBCU (Count ON Bits in 32-bit Data)

Counts the number of ON bits in the specified 32-bit data.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the 32-bit data subject to the bit count, or constant data
D	Area storing the number of ON bits

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

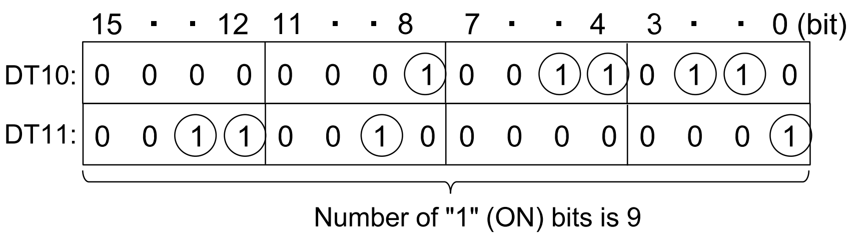
■ Outline of operation

- The number of ON bits (bits with a value of 1) in the 32-bit data specified by [S] and [S+1] is counted, and the result is stored in the area specified by [D].
- The result is stored as a decimal number.

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the number of ON bits in the data stored in DT10 and DT11 is stored in DT20.



When R0 turns ON, K9 is stored in DT20.

■ Flag operations

Name	Description
R9007	Turns ON when the area is exceeded in index modification.

Name	Description
R9008 (ER)	

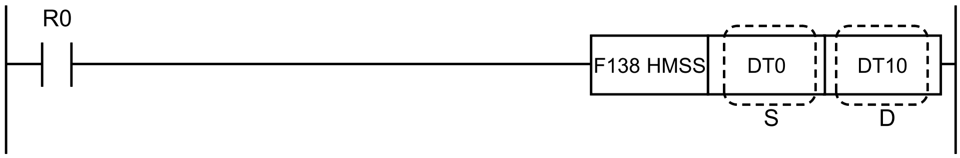
(MEMO)

20 Special Instructions

20.1 F138 HMSS (Hour, Minute, Second Data to Second Data Conversion)

Converts data representing hours, minutes, and seconds into data representing seconds.

■ Instruction format



■ Operands

Items	Settings
S	Starting address of the area storing the two-word data representing hours/minutes/seconds
D	Starting address of the area storing the conversion result (second data)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•					•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- Converts the 2-word time data (hours/minutes/seconds) starting at the address specified by [S], converts it to seconds, and then stores the result in the 2-word area starting from the address specified by [D].

■ Data Structure

- Time data [S] representing hours, minutes, and seconds
  - is composed of 2-word BCD (H constant) data.
  - Specify it as shown below: hours (4-digit), minutes (2-digit), and seconds (2-digit).
    (Can be specified with a maximum of 9999 hours, 59 minutes, and 59 seconds.)

	(Higher)	(Lower)
S	Minutes (H00 to H59)	Seconds (H00 to H59)
S+1	Time (H0000 to H9999)	

e.g. 3 hours, 45 minutes, and 19 seconds
S = H4519
S+1 = H0003

- Time data [D] representing seconds

- is composed of 2-word BCD (H constant, maximum 8-digit) data.
- It is stored as shown below.

D+1	D
Seconds (H0000000	0 to H99999999)

e.g. 35,999,999 seconds

D = H9999

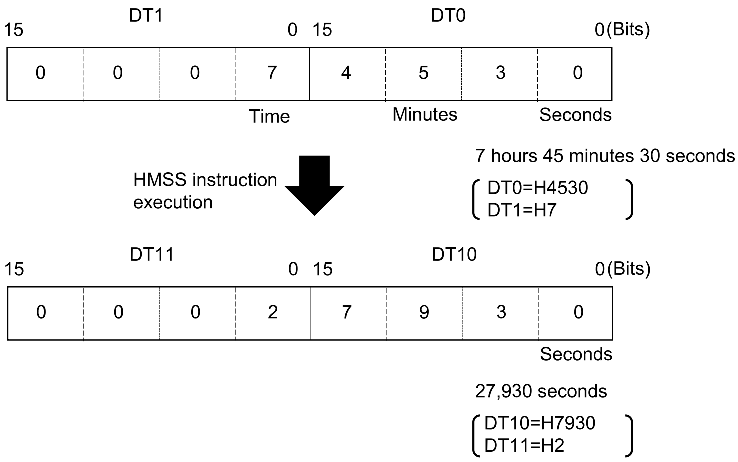
D+1 = H3599

Note: As the maximum time data that can be specified is 9999 hours, 59 minutes, and 59 seconds, the actual maximum value for the seconds that will be stored in [D] is 35,999,999 seconds.

■ Operation example

Operation of instruction format description program

The time data representing hours, minutes, and seconds that is stored in data registers DT0 and DT1 is converted to seconds and then stored in DT10 and DT11 when internal relay R0 turns ON.



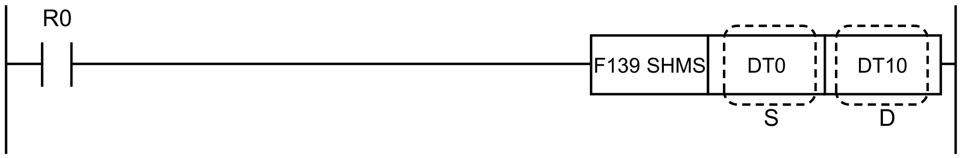
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the data specified by [S] is not BCD data
	Turns ON when the portion of [S] representing minutes and seconds is exceeds the range of 00 to 59

20.2 F139 SHMS (Second Data to Hour, Minute, Second Data Conversion)

Converts data representing seconds (up to 8 digits) to data representing hours, minutes, and seconds.

■ Instruction format



■ Operands

Items	Settings
S	Starting address of the area storing the 2-word data representing seconds
D	Starting address of the area that stores the conversion result (hours, minutes, and seconds data)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•					•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- Converts the 2-word time data (in seconds) starting from the address specified by [S] to time data expressed in hours, minutes, and seconds (H constant), and stores the result in the 2-word area whose starting address is specified by [D].

■ Data structure

- Time data representing seconds [S]
  - is composed of 2-word BCD (H constant, maximum 8-digit) data.
  - Specify it in seconds as shown below.

S+1	S
Seconds (H0000	0000 to H35999999)

e.g. 35,999,999 seconds

S = H9999

S+1 = H3599

Note: The maximum value that can be stored in [D] is 9,999 hours, 59 minutes and 59 seconds, so the maximum value that can be specified for the time data for the seconds unit is 35,999,999 seconds.

- Time data representing hours, minutes, and seconds [D]
  - is composed of 2-word BCD (H constant) data.
  - The time data represents hours (4 digits), minutes (2 digits), and seconds (2 digits) as shown below.

	(Higher)	(Lower)
D	Minutes (H00 to H59)	Seconds (H00 to H59)
D+1	Time (H0000 to H9999)	

e.g. 3 hours, 45 minutes, and 19 seconds

D = H4519

D+1 = H0003

■ Operation example

Operation of instruction format description program

Converts the seconds data stored in data registers DT0 to DT1 to hour, minute, and second data when internal relay R0 turns ON. The converted hour, minute, and second data is stored in data registers DT10 to DT11.

DT1 15			0	DT0 15			0 (Bits)
0	0	0	0	4	0	0	0
							Seconds

SHMS instruction execution

4000 seconds (DT0=H4000, DT1=H0)

DT11 15			0	DT10 15			0 (Bits)
0	0	0	1	0	6	4	0
			Hours		Minutes		Seconds

1 hour 6 minutes 40 seconds (DT10=H640, DT11=H01)

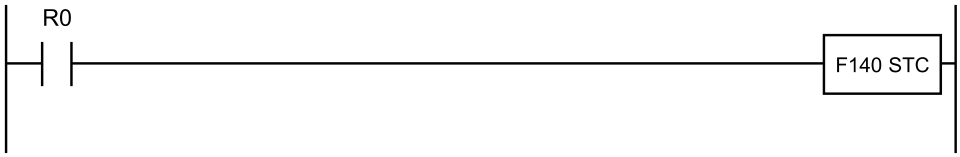
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the data specified by [S] is not BCD data
	Turns ON when the content of [S] exceeds 35,999,999

20.3 F140 STC (Cy Flag Set)

Turns the CY flag ON.

■ Instruction format



■ Outline of operation

The CY (carry) flag (R9009) is turned ON.

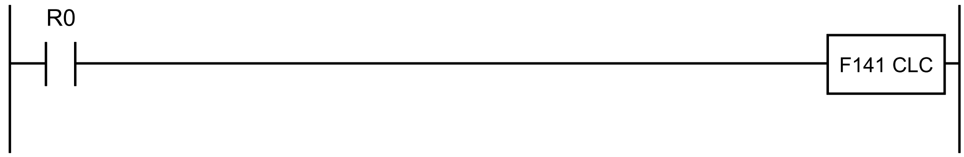
■ Flag operations

Name	Description
R9009 (CY)	Turns ON after this instruction is executed

20.4 F141 CLC (Cy Flag Clear)

Turns the CY flag OFF.

■ Instruction format



■ Outline of operation

The CY (carry) flag (R9009) is turned OFF.

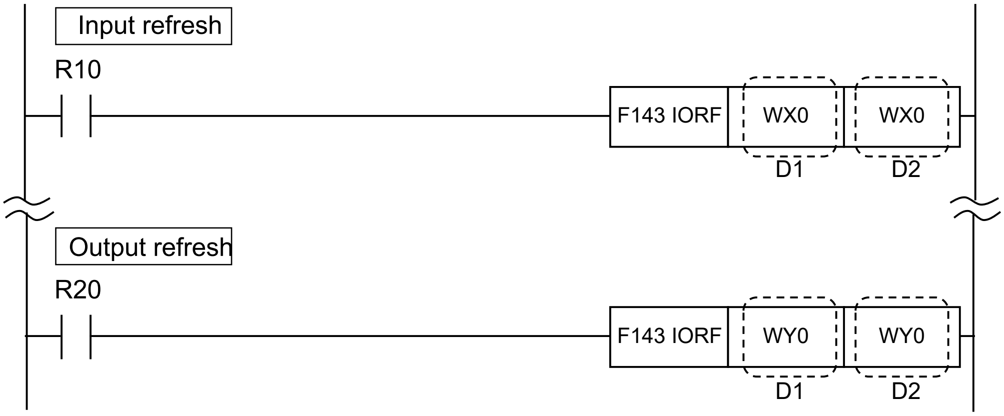
■ Flag operations

Name	Description
R9009 (CY)	Turns OFF after this instruction is executed

20.5 F143 IORF (Partial I/O Refresh)

The input or output of a specified range is refreshed.

■ Instruction format



■ Operands

Item	Setting
D1	The starting word No. of the I/O to be refreshed.
D2	The ending word No. of the I/O to be refreshed.

■ Devices that can be specified (indicated by •)

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer device
												K	H	M	f		
D1	•	•														•	
D2	•	•														•	

■ Outline of operation

- An I/O refresh (input/output processing) of the external input X or external output Y is executed for a range from the number specified in [D1] to the number specified in [D2].
- When refreshing input, specify WX** to [D1] and [D2].
- When refreshing output, specify WY** to [D1] and [D2].

Main unit	FP0R expansion
Available	Available

(Note 1) Partial refresh of FP0 expansion is possible, but will take approx. 1 ms per unit.

■ Operation example

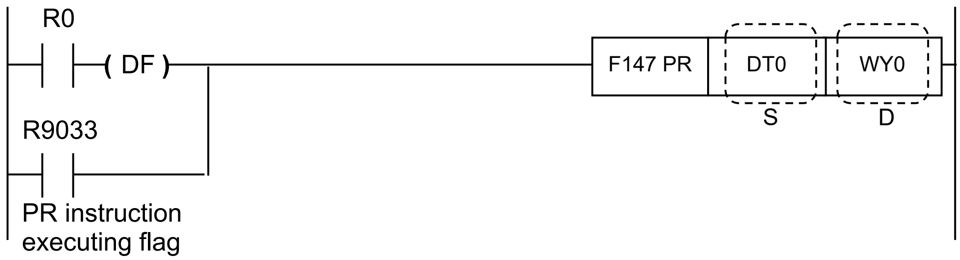
Operation of instruction format description program

When internal relay R10 is ON, an I/O refresh of input relay WX0 (X0 to XF) is executed.
When internal relay R20 is ON, an I/O refresh of output relay WY0 (Y0 to YF) is executed.

20.6 F147 PR (Printout)

Outputs text data (ASCII codes) to the printer.

■ Instruction format



■ Operands

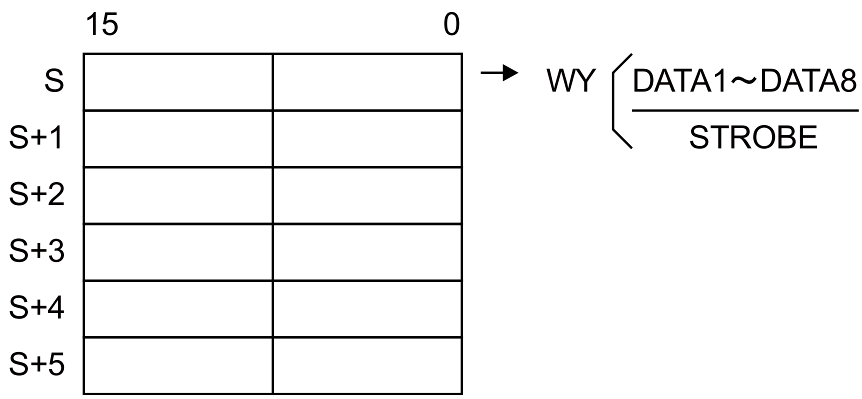
Items	Settings
S	Starting address of the area storing printout data (ASCII codes)
D	Area for output of printout data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S	•	•	•	•	•	•	•	•	•	•	•						
D		•															

■ Outline of operation

- Outputs the ASCII codes (for 12 characters) stored in the six-word area starting with the address specified by [S] to the area WY specified by [D].



- In the WY area, Y□0 to Y□7 are data signals DATA1 to DATA8, and Y□8 is the strobe signal. Y□9 to Y□F are not used. When the printout instruction is executed, the printout data is output from Y0 to Y7 (ASCII code), and the strobe signal is output from Y8.
- ASCII code is output in order from the starting address.

- Be sure to set the printer control code (LF, CR) as data within the 6-word (12 characters) area above.
- After the start of execution of a printout instruction, 37 scans are required until 12 characters complete output. (See the"P.20-12"Time Chart for more details.)

■ Precautions for programming

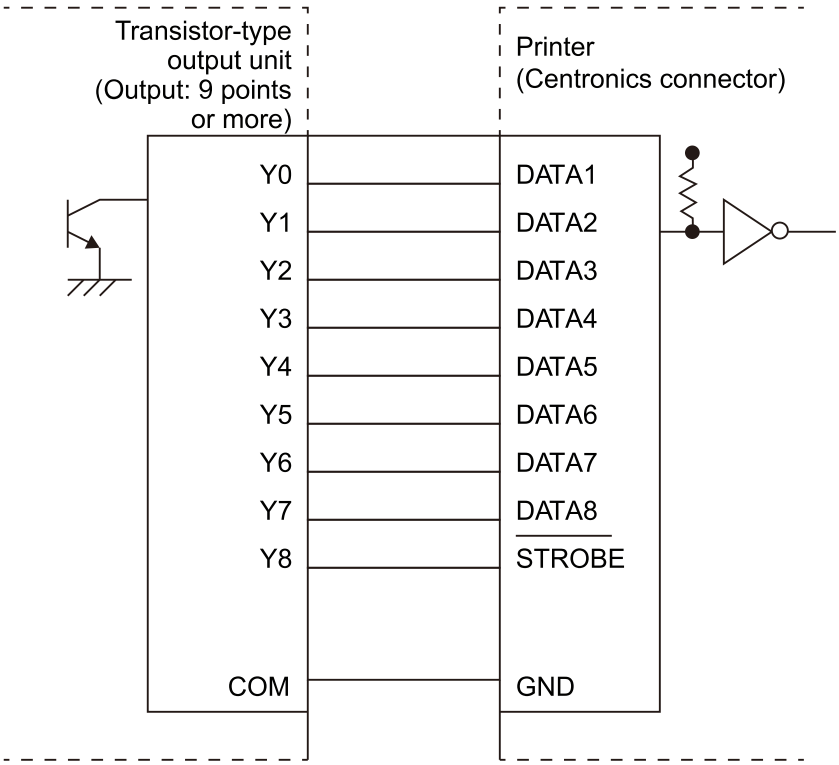
- Multiple F147 PR instructions cannot be executed at the same time. The program should be set up so that the printout flag (R9033) is used during execution of a F147 PR instruction to inhibit simultaneous execution.
- The ASCII code conversion instruction (F95 ASC) can be used to convert character constants (M) to ASCII codes.
- Character constants can be input only with programming tool software.
- A transistor-type output unit (output board) is necessary.
- When this instruction is executed, zero <OFF> is set for Y□9 to Y□F in the WY area specified by [D].

■ Operation example

Operation of instruction format description program

The ASCII codes stored in data registers DT0 to DT5 are output to WY0 when internal relay R10 turns ON.

■ Connection method



■ Data setting

Set the data to be printed out in order from the lower byte of the first word.

<Example> Outputting 10 characters"ABCDEFGHIJ"to a printer

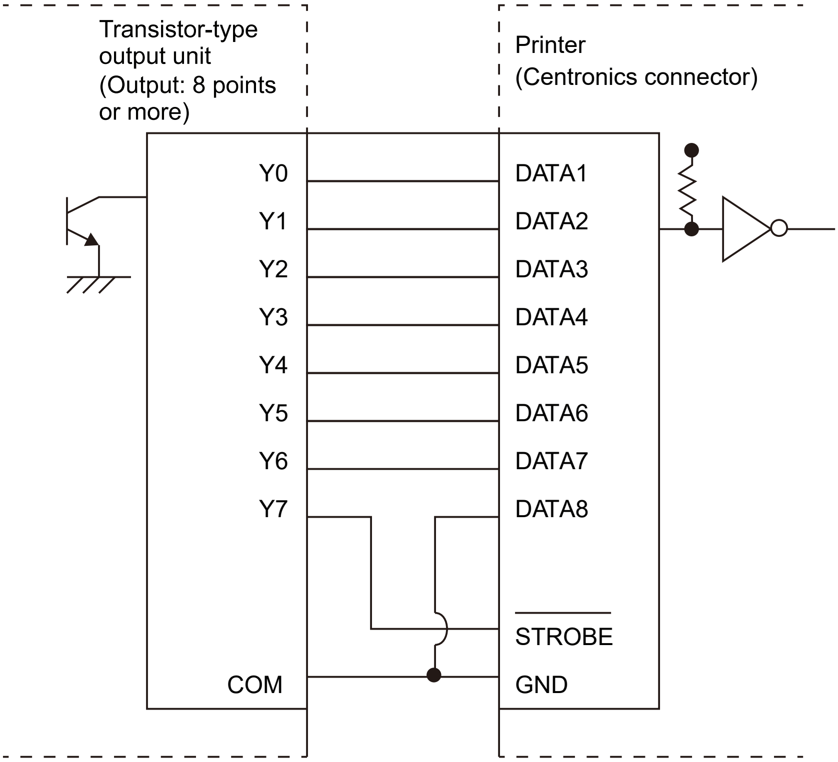
	15	0
DT0	42	41
DT1	44	43
DT2	46	45
DT3	48	47
DT4	4A	49
DT5	0A	0D

LF CR

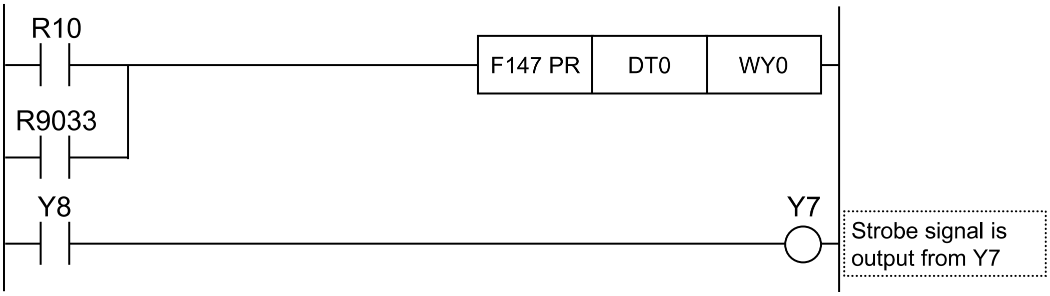
■ Printer output using eight-point output

- When only eight output points are being used, connections should be made as shown below, and the program should be set up so that the strobe signal is output from Y7.
  However, in this case, only alphanumeric characters can be output.

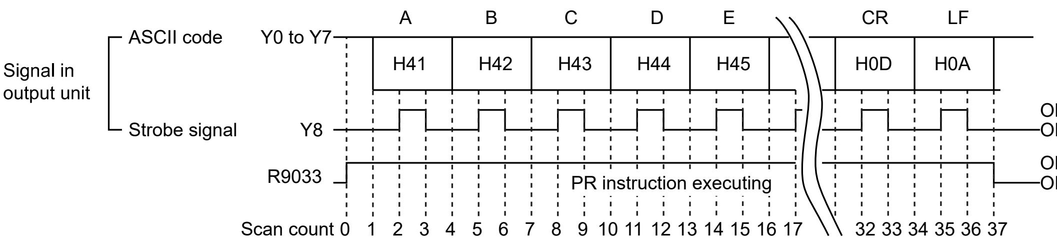
Connection example



Program example



■ Timing chart



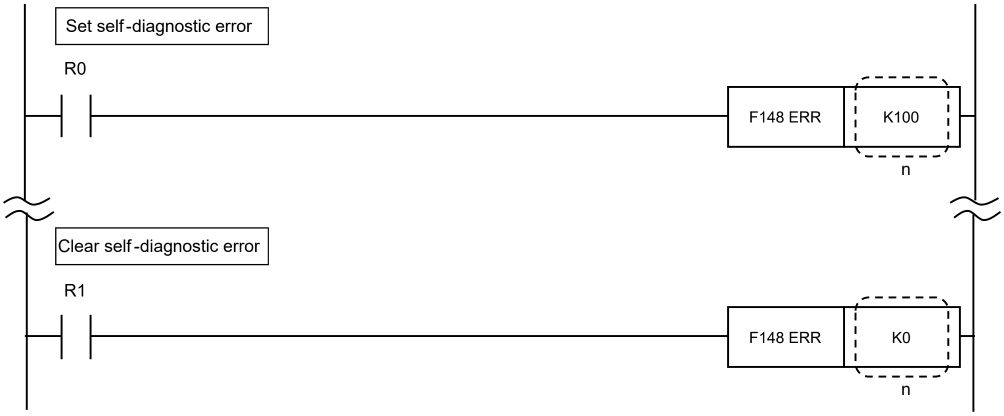
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the six words starting with [S] exceed the range of the area
	Turns ON when another F147 (PR) instruction attempts execution while one F147 (PR) instruction is being executed

20.7 F148 ERR (Self-diagnostic Error Code Set)

Detects self-diagnostic errors using a set detection condition.

■ Instruction format



■ Operands

Item	Setting
n	Self-diagnostic error code (0, 100 to 299)

■ Devices that can be specified (indicated by ●)

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer device
												K	H	M	f		
n												●	●				

■ Outline of operation

- The self-diagnostic error code specified by [n] is stored in the special data register DT90000 and the self-diagnostic error flag (R9000) is turned ON. Additionally, the ERR/ALM LED flashes.
- [n] (self-diagnostic error code) can be set in the range from K100 to K299. The set value determines whether to stop or continue operation upon execution.

[n] setting	Operation when an error occurs
K100 to K199	Operation stops
K200 to K299	Operation continues

- When K200 to K299 is set for [n], if multiple F148 ERR instructions are processed at the same time, the code with the lower number will be accepted with priority.
- When the F148 ERR instruction is executed with [n] set to 0, self-diagnostic errors with error code 43 or higher are cleared.

Item		Operation when self-diagnostic error is cleared
ERR/ALM LED	-	Light switch off

Item		Operation when self-diagnostic error is cleared
R9000	Self-diagnostic error flag	OFF
R9005	Backup battery abnormality flag (current type)	
R9006	Backup battery abnormality flag (hold type)	
R9007	Operation error flag (hold type) (ER flag)	
R9008	Operation error flag (new type) (ER flag)	
DT90000	Self-diagnostic error code	0 clear
DT90017	Address with operation error (hold type)	
DT90018	Address with operation error (new type)	

- F148 ERR instructions with the same error code can be notated in duplicate.

■ Operation example

Operation of instruction format description program

- If internal relay R0 is ON, self-diagnostic error 100 is set. The ERR/ALM LED flashes and operation stops. (Ensure that internal relay R0 is turned ON in situations where the self-diagnostic error 100 is to be set.)
- When internal relay R1 is ON, self-diagnostic errors with error code 43 or higher are cleared.

■ Confirming self-diagnostic errors

- The confirmation method is the same as for normal self-diagnostic errors.
  Special data register number: DT90000, DT90017, DT90018

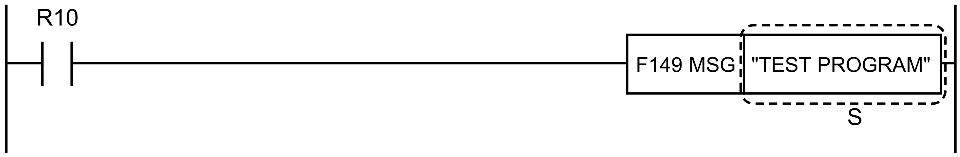
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when [n] is outside of the specified range.

20.8 F149 MSG (Character Send to Programming Tool)

Displays a message on the programming tool.

■ Instruction format



■ Operands

Items	Settings
S	Message (character constant)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S														●			

■ Outline of operation

- The characters specified by [S] are displayed on the programming tool connected to the controller.
- The message can also be read from"Message display"on the tool software menu.
- The character constant M can only be input by programming tool software.
- The message flag (R9026) turns ON, and the content of [S] is set to special data registers DT90030 to DT90035.
- If a message is already being displayed, the displayed content does not change even if this instruction is executed. To clear the message displayed, click the "Cancel" button on the "Display PLC Message" screen using the programming tool software.

■ Operation example

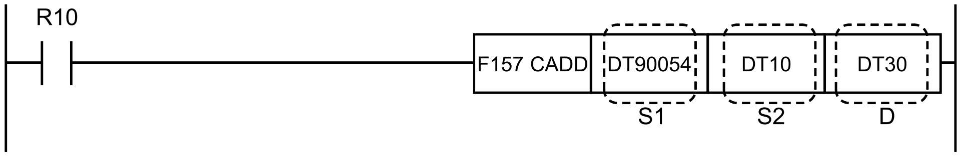
Operation of instruction format description program

When internal relay R10 turns ON, the message"TEST PROGRAM"is displayed on the programming tool.

20.9 F157 CADD (Calendar Data Addition)

Calculates the date and time after a specified amount of time (hours, minutes, and seconds) has elapsed since a certain date and time (year, month, day, hour, minute, second).

■ Instruction format



■ Operands

Items	Settings
S1	Starting address of area storing date and time data (three words)
S2	Starting address of area storing date and time data (two words), or constant data
D	Starting address of area storing addition result date and time data (three words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•		•	•					•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•								•	

■ Outline of operation

- The three-word date and time data (year, month, day, hour, minute, second) that starts at the address specified by [S1] and the time data (hours, minutes, and seconds) specified by [S2] are added together. The result (time of elapsed value) is stored in the three-word area that starts at the address specified by [D].

<Time data>

	(Higher)	(Lower)
[S1]	Minutes (H00 to H59)	Seconds (H00 to H59)
[S1+1]	Day (H01 to H31)	Hour (H01 to H23)
[S1]+2	Year (H00 to H99)	Month (H00 to H12)

+ (addition)

<Time data>

	(Higher)	(Lower)
[S1]	Minutes (H00 to H59)	Seconds (H00 to H59)
[S1+1]	Time (H0000 to H9999)	

↓

<Time data>

	(Higher)	(Lower)
[D]	Minutes (H00 to H59)	Seconds (H00 to H59)
[D1+1]	Day (H01 to H31)	Hour (H01 to H23)
[D1]+2	Year (H00 to H99)	Month (H00 to H12)

- Specify the values for date and time data [S1] and time data [S2] using BCD data (H constant).

  [Example of date and time data]

  14 hours, 23 minutes, and 31 seconds on August 1, 1992

  S1 = H2331 (23 hours, 31 minutes)

  S1+1 = H0114 (1st of the month, 14th hour)

  S1+2 = H9208 (1992, August)

  [Example of time data]

  32 hours, 50 minutes, and 45 seconds

  S2 = H5045 (50 minutes, 45 seconds)

  S2+1 = H0032 (32 hours)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the built-in calendar timer reads date and time data and adds the time data stored in data registers DT10 and DT11. The date and time resulting from the addition is stored in DT30 to DT32.

DT90054	H30	H24	1992, June,17th 10 hours 30 minutes 24 seconds
DT90055	H17	H10	
DT90056	H92	H06	

+ (addition)

DT10	H45	H00	20 hours 45 minutes later
DT11	H00	H20	

↓

DT30	H15	H24	1992, June,18th 7 hours 15 minutes 24 seconds
DT31	H18	H07	
DT32	H92	H06	

■ Data configuration of built-in calendar timer

	(Higher)	(Lower)
DT90054	Minutes	Seconds
DT90055	Day	Hours
DT90056	Year	Month

■ Precautions for programming

Special data registers DT90054 to DT90056, in which the values of the built-in calendar timer are stored, cannot be specified directly for [D]. To change the values of the built-in calendar timer, store the addition results in a separate memory area, and then use the F0 MV instruction to transfer the values to DT90054 to DT90056.

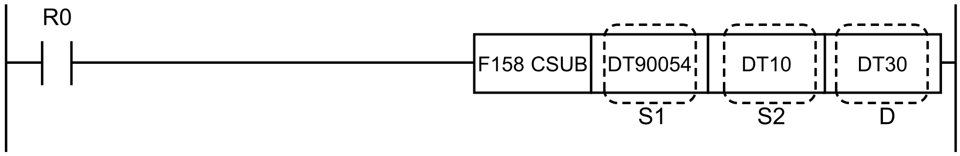
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the data specified by [S1] and [S2] is not BCD data
	Turns ON when the data specified by [S1] is not date and time data
	Turns ON when the data specified by [S2] is not time data
	Turns ON when the specified data exceeds the area

20.10 F158 CSUB (Calendar Data Subtraction)

Calculates the date and time a specified amount of time (hours, minutes, and seconds) before a certain date and time (year, month, day, hour, minute, second).

■ Instruction format



■ Operands

Items	Settings
S1	Starting address of area storing date and time data (three words)
S2	Starting address of area storing date and time data (two words), or constant data
D	Starting address of area storing subtraction result date and time data (three words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•		•	•					•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•								•	

■ Outline of operation

- The time data (hours, minutes, and seconds) specified by [S2] is subtracted from the three-word date and time data (year, month, day, hour, minute, second) that starts at the address specified by [S1]. The result is stored in the three-word area that starts at the address specified by [D].

<Time data>

	(Higher)	(Lower)
[S1]	Minutes (H00 to H59)	Seconds (H00 to H59)
[S1+1]	Day (H01 to H31)	Hour (H01 to H23)
[S1+2]	Year (H00 to H99)	Month (H00 to H12)

- (subtraction)

<Time data>

	(Higher)	(Lower)
[S1]	Minutes (H00 to H59)	Seconds (H00 to H59)
[S2+1]	Time (H0000 to H9999)	

↓

<Time data>

	(Higher)	(Lower)
[D]	Minutes (H00 to H59)	Seconds (H00 to H59)
[D1+1]	Day (H01 to H31)	Hour (H01 to H23)
[D1+2]	Year (H00 to H99)	Month (H00 to H12)

- Specify the values for date and time data [S1] and time data [S2] using BCD data (H constant).

  [Example of date and time data]

  14 hours, 23 minutes, and 31 seconds on December 1, 1994

  S1 = H2331 (23 hours, 31 minutes)

  S1+1 = H0114 (1st of the month, 14th hour)

  S1+2 = H9412 (1994, December)

  [Example of time data]

  32 hours, 50 minutes, and 45 seconds

  S2 = H5045 (50 minutes, 45 seconds)

  S2+1 = H0032 (32 hours)

■ Operation example

Operation of instruction format description program

When internal relay R0 turns ON, the built-in calendar timer reads date and time data and subtracts the time data stored in data registers DT10 and DT11. The date and time resulting from the subtraction is stored in DT30 to DT32.

DT90054	H30	H24
DT90055	H17	H10
DT90056	H92	H06

1992, June,17th
10 hours
30 minutes
24 seconds

- (subtraction)

DT10	H30	H30
DT11	H00	H03

3 hours
30 minutes
30 second earli

↓

DT30	H59	H54
DT31	H17	H06
DT32	H92	H06

1992, June,17th
6 hours
59 minutes
54 seconds

■ Precautions for programming

Special data registers DT90054 to DT90056, in which the values of the built-in calendar timer are stored, cannot be specified directly for [D]. To change the values of the built-in calendar timer, store the addition results in a separate memory area, and then use the F0 MV instruction to transfer the values to DT90054 to DT90056.

■ Usage example: Calculating elapsed time

The F158 CSUB instruction can be used to calculate elapsed time. Using the calendar timer, the starting date and time and the ending date and time are stored in the data memory and the time that has elapsed between them is calculated.

This is explained using the example of calculating the stopped time for an operation that stopped at 08 hours, 02 minutes, and 15 seconds and restarted at 10 hours, 30 minutes, and 25 seconds.

This can be thought of as"subtracting 8 hours, 2 minutes, and 15 seconds from 10 hours, 30 minutes, and 25 seconds".

Start time [S2]

02	15	2 minutes, 15 seconds
23	08	23rd day, 8 hours
94	12	1994, December

Start time [S2]

30	25	30 minutes, 25 seconds
23	10	23rd day, 10 hours
94	12	1994, December

The data to be subtracted is taken from the starting date and time data as is shown below.

02	15	2 minutes, 15 seconds
00	08	8 hours

The part representing "day" is set to 0

CSUB instruction executio

The result will be as follows.

[D]

02	10	2 minutes,10 seconds
23	02	23rd day, 2 hours
94	12	1994, December

■ Data configuration of built-in calendar timer

	(Higher)	(Lower)
DT90054	Minutes	Seconds
DT90055	Day	Hours
DT90056	Year	Month

■ Flag operations

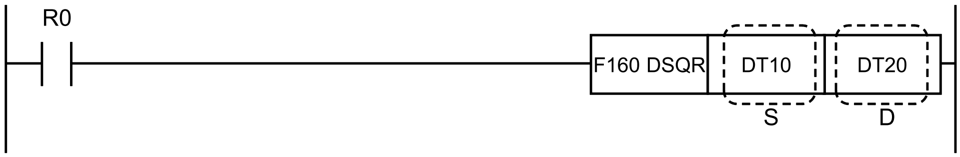
Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the data specified by [S1] and [S2] is not BCD data
	Turns ON when the data specified by [S1] is not date and time data
	Turns ON when the data specified by [S2] is not time data

Name	Description
	Turns ON when the specified data exceeds the area

20.11 F160 DSQR (32-bit Data Square Root)

Calculates the square root of the specified 32-bit data.

■ Instruction format



■ Operands

Items	Settings
S	Area storing the data for square root calculation, or constant data
D	Area storing the calculated square root

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S	•	•	•	•	•	•	•	•	•			•	•			•	
D		•	•	•	•	•	•	•	•							•	

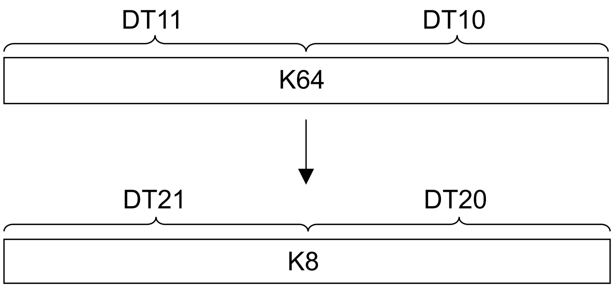
■ Outline of operation

- Calculates the square root of the 32-bit data (K constant) stored in [S] and [S+1], then stores the result (K constant) in [D] and [D+1]. Fractions are rounded down.
  √[S] -> [D]

■ Operation example

Operation of instruction format description program

The square root (√) of the 32-bit data stored in DT10 and DT11 is calculated and the result stored in DT20 and DT21 when internal relay R0 turns ON. When K64 is stored in DT10 to DT11, it will be as follows.



Finds the square root of 64, which is 8.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	ON when the data specified by [S] is a negative value

21 Serial Communication Instructions

21.1 F145 Data Send/F146 Data Receive Instruction Common Items (Serial Communication)

■ System register settings

Use tool software to set the communication mode of the COM port used.

Mode	System register No. 412
MEWTOCOL master	Computer link
MODBUS master	MODBUS RTU

■ Conditions for execution of the instruction

- Multiple SEND instructions or RECV instructions cannot be executed simultaneously to the same communication port. Set up the program so that SEND/RECV instructions are executed when the SEND/RECV execution enabled flag is ON (1).

Info.

- In global transfer (send implemented by specifying H00 as the unit number), the SEND/RECV execution enabled flag does not turn OFF (0). Set up the program so that after a send is completed, the next send is performed after waiting for at least the maximum scan time.

■ Confirmation of execution of the instruction

- The operation processing time for the SEND/RECV instruction is only for the request to send; the actual send is performed when the ED instruction is executed. Check the SEND/RECV instruction completion flag to confirm the completion of sending.
- If the instruction ends abnormally, the SEND/RECV completion flag turns ON. Also, an error code is stored in the SEND/RECV done code. For details of error codes, refer to the following table.

Special data register	Error code	Description
DT90124	H0	Normal end
	H73	Timeout error occurred while waiting for a response
	Error code of each protocol	For details of error codes, refer to the error codes of each protocol. ● "31.6.3 List of MEWTOCOL-COM/DAT Communication Error Codes"

Name	Operation	COM1
SEND/RECV Execution enabled flag	0: Execution disabled 1: Execution enabled	R9044
SEND/RECV Completion flag	0: Normal end 1: Abnormal end	R9045
SEND/RECV Done codes	In case of abnormal end, the error code is stored.	DT90124

Timeout period setting

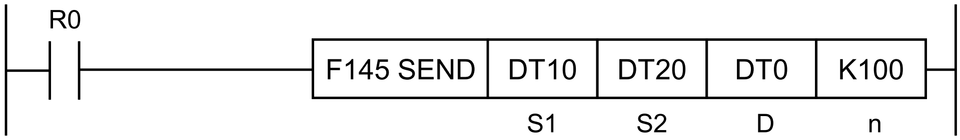
- If the error code is H73, it means that a timeout has occurred while waiting for a response.
- The timeout time can be changed from 10.0 ms to 81.9 s (in 2.5-ms units) by using system register No. 32. By default, the value is set to 10 s.

Other restrictions

These instructions cannot be executed on special internal relays (R9000 and up) or special data registers (DT90000).

21.2 F145 SEND [Data Send Instruction (MEWTOCOL Master)]

■ Instruction format



■ Operands

<table>
<tr><th>Operand</th><th colspan="2">Setting</th></tr>
<tr><td rowspan="3">S1</td><td colspan="2">Specify the starting No. of the area (two words) that stores control data.</td></tr>
<tr><td>S1</td><td>Specify a transfer method.
Word transfer: Specify the number of transferred words.
Bit transfer: Specify master unit bit No. and partner unit bit No.</td></tr>
<tr><td>S1+1</td><td>Specify the COM port number of the master unit and the unit number of the partner unit.</td></tr>
<tr><td>S2</td><td>Specification of master unit</td><td>Specify the area of a master unit that stores send data.</td></tr>
<tr><td>D</td><td>Specification of partner unit</td><td>Specify the type of area of the partner unit that stores send data. The number is specified at 0.</td></tr>
<tr><td>n</td><td>Specification of partner unit</td><td>Specify the starting address of the partner unit that stores send data. (Specification range: H0 to HFFFF)</td></tr>
</table>

■ Devices that can be specified (indicated by •)

Operand	WX	WY	WR	WL	SV	EV	DT	LD	In	SWR	SDT	Constant		Index modifier (Note 1)
												K	H	
S1	•	•	•	•	•	•	•	•		•	•			•
S2	•	•	•	•	•	•	•	•		•	•			•
D		•	•	•	•	•	•	•						
n		•	•	•	•	•	•	•				•	•	•

(Note 1) A character constant cannot be specified.

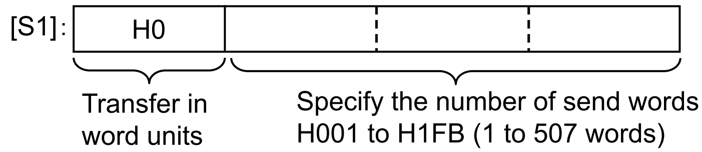
■ Outline of operation

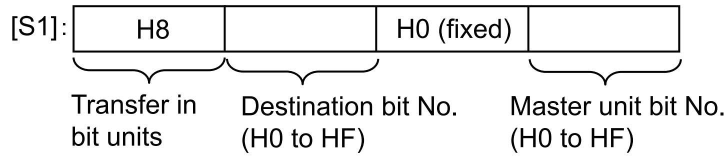
- Use this instruction to send commands in computer link mode by connecting a unit that can receive MEWTOCOL-COM commands to a serial port (COM1) of the specified unit.Before using it, set the operation mode of the COM port to be used to Computer Link (in the system register settings).
- The data specified by [S2] in the master unit is written from the area specified by [D] and [n] in the partner unit in accordance with the specification of 2-word data stored in the control data starting at the area specified by [S1].

■ Specifying control data

Specifying [S1]

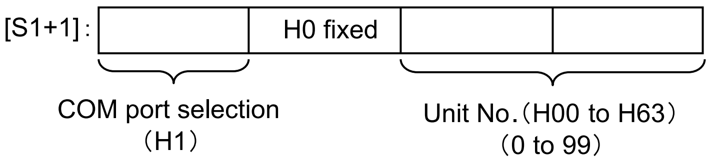
For [S1], specify the transfer method. The specification method differs depending on word transfer and bit transfer.





Specifying [S1+1]

For [S1+1], specify the COM port number of the master unit and the unit number of the partner unit.



COM port selection

COM ports can be selected as shown in the following table.

COM port selection (upper 4 bits)	H1
Port number	COM1

■ Specifying the transfer unit and method [S1]

Specify the quantity of send data in word units, or the location of target bits of send data in bit units.

■ Specifying the partner unit [S1+1]

Specify the unit number of a partner unit. When H00 is specified, global transfer is applied (noresponse). Specify the source port from which data is sent to the partner unit as COM1.

■ Specifying the master unit area that stores send data [S2]

Specify the memory area of a master unit that stores send data.

■ Specifying the partner unit storage area [D], [n]

Specify "0" for [D] as the device No.

Specify the memory area of a partner unit that stores sent data, combining type [D] and address [n].

Example 1: [D]: DT0, [n]: K100

↓

DT100

Example 2: [D]: DT0, [n]: HFFF0

↓

DT65520

When DT0 or LD0 is specified for [D] and an H constant is specified for [n], data can be sent without checking.

(Example) When DT0 and HFFFFF are specified, DT65535 is accessible. This is useful for accessing data registers of the Eco-Power Meter KW8M.

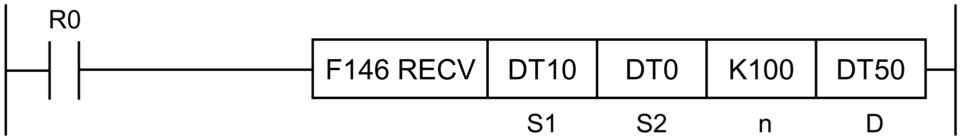
- A MEWTOCOL-COM command is created according to the operands specified by [S1], [S1+1], [S2], [D], and [n].

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the control data value in [S1] and [S1+1] is outside the specified range.
	Turns ON when acquiring the number of words specified in [S1] exceeds the [S2] or [D] area during transfer in word units.
	Turns ON when [D]+[n] exceeds the [D] area.
	Turns ON when the operation mode of the target COM port is other than computer link.
	Word unit ● Turns ON if [D] is [DT/LD], when [n] is not from 0 to 99999. ● Turns ON if [D] is WY/WR/WL/SV/EV, when [n] is not from 0 to 9999. Bit unit ● Turns ON when [D] is not WY/WR/WL. ● Turns ON when [n] is not 0 to 999.
	Turns ON when the [D] device No. is not 0.

21.3 F146 RECV [Data Receive Instruction (MEWTOCOL Master)]

■ Instruction format



■ Operands

Operand	Setting		
S1	Specify the starting No. of the area (two words) that stores control data.		
	S1	Specify a transfer method. Word transfer: Specify the number of transferred words. Bit transfer: Specify master unit bit No. and partner unit bit No.	
	S1+1	Specify the COM port number of the master unit and the unit number of the partner unit.	
S2	Specification of partner unit	Specify the source data area of a partner unit. (Device No. is fixed to “0”)	
n	Specification of partner unit	Specify the starting address of the device in the source data area of a partner unit. (Setting range: H0 to HFFFF)	
D	Specification of master unit	Specify the device starting address of the receive data storage area in the master unit.	

■ Devices that can be specified (indicated by ●)

Operand	WX	WY	WR	WL	SV	EV	DT	LD	In	SWR	SDT	Constant		Index modifier (Note 1)
												K	H	
S1	●	●	●	●	●	●	●	●		●	●			●
S2	●	●	●	●	●	●	●	●						
n		●	●	●	●	●	●	●				●	●	●
D		●	●	●	●	●	●	●						●

(Note 1) A character constant cannot be specified.

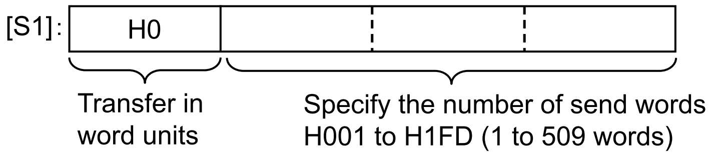
■ Outline of operation

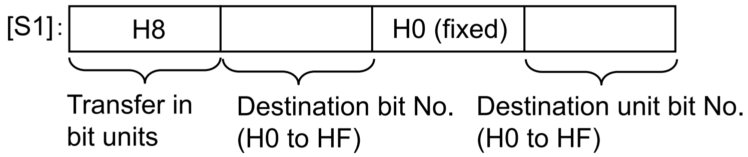
- Use this instruction to send commands in computer link mode by connecting a unit that can receive MEWTOCOL-COM commands to a serial port (COM1) of the specified unit.
- In accordance with the specification of 2-word data stored in the control data starting with an area specified in [S1], reading is performed from the area specified by [S2] and [n] in a partner unit, starting with the [D] area of the master unit.

■ Specifying control data [S1], [S1+1]

Specifying [S1]

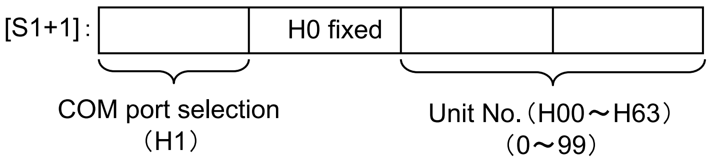
For [S1], specify the transfer method. The specification method differs depending on word transfer and bit transfer.





Specifying [S1+1]

For [S1+1], specify the COM port number of the master unit and the unit number of the partner unit.



COM port selection

COM ports can be selected as shown in the following table.

COM port selection (upper 4 bits)	H1
Port number	COM1

■ Specifying the transfer unit and method [S1]

Specify the quantity of send data in word units, or the location of target bits of send data in bit units.

■ Specifying the partner unit [S1+1]

Specify the unit number of a partner unit.

Specify the destination port of the partner unit as COM1.

■ Specifying the starting address of the source data area [S2]

Specify "0" for [S2] as the device No. Specify the memory area of a partner unit that stores sent data, combining type [S2] and address [n].

Example 1) [S2]: DT0, [n]: K100

↓

DT100

Example 2) [S2]: DT0, [n]: HFFF0

↓

DT65520

When DT0 or LD0 is specified for [D] and an H constant is specified for [n], data can be sent without checking.

(Example) When DT0 and HFFFFF are specified, DT65535 is accessible. This is useful for accessing data registers of the Eco-Power Meter KW8M.

■ Specifying the master unit area that stores received data [D]

Specify the memory area of the master unit that stores received data.

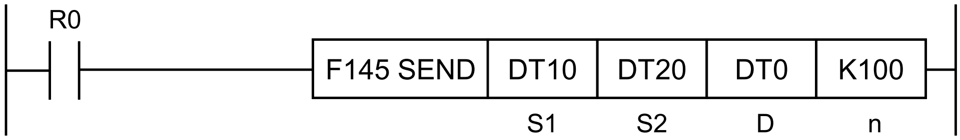
- A MEWTOCOL-COM command is created according to the operands specified by [S1], [S1+1], [S2], [D], and [n].

■ Flag operations

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td rowspan="6">R9007
R9008
(ER)</td><td>Turns ON when the control data value in [S1] and [S1+1] is outside the specified range.</td></tr>
<tr><td>Turns ON when acquiring the number of words specified in [S1] exceeds the [S2] or [D] area during transfer in word units.</td></tr>
<tr><td>Turns ON when [S2]+[n] exceeds the [S2] area.</td></tr>
<tr><td>Turns ON when the operation mode of the target COM port is other than computer link.</td></tr>
<tr><td>Word unit
● Turns ON when [n] is not between 0 and 99999 when [S2] is DT/LD.
● Turns ON when [n] is not between 0 and 9999 when [S2] is WX/WY/WR/WL/SV/EV.
Bit unit
● Turns ON when [S2] is not WX/WY/WR/WL
● Turns ON when [n] is not 0 to 999.</td></tr>
<tr><td>Turns ON when the [S2] device number is not 0.</td></tr>
</table>

21.4 F145 SEND［Data Send Instruction（MODBUS Master Mode）］

■ Instruction format



■ Operands

Operands	Setting	
S1	Specify the starting No. of the area (two words) that stores control data.	
	S1	Specify a transfer method. Word transfer: Specify the number of transferred words. Bit transfer: Specify master unit bit No. and partner unit bit No.
	S1+1	Specify the COM port number of the master unit and the unit number of the partner unit.
S2	Specification of master unit	Specify the area of a master unit that stores send data.
D	Specification of partner unit	Specify the type of area of the partner unit that stores send data. The number is specified at 0.
n	Specification of partner unit	Specify the starting address of the partner unit that stores send data.

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	In	SWR	SDT	Constant		Index modifier (Note 1)
												K	H	
S1	•	•	•	•	•	•	•	•		•	•			•
S2	•	•	•	•	•	•	•	•		•	•			•
D		•	•				•							
n		•	•	•	•	•	•	•				•	•	•

(Note 1) A character constant cannot be specified.

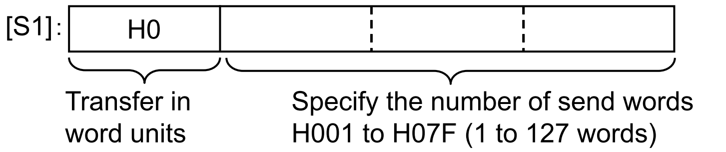
■ Outline of operation

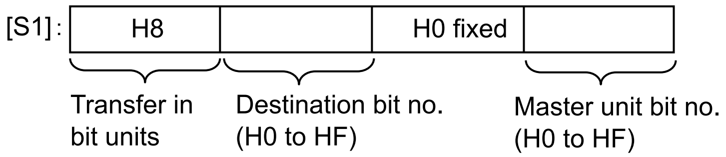
- It is used to send commands to the serial port (COM1) of the specified unit in the MODBUS mode connecting the unit that enables to receive the MODBUS command. (MODBUS command 05,06,15,16)
- The data of the local area specified by [S2] is written in the area of the remote unit specified by [D] and [N], according to the specification for the 2−word data stored in the control data with starting area specified by [S1].

■ Specifying control data [S1], [S1+1]

Specifying [S1]

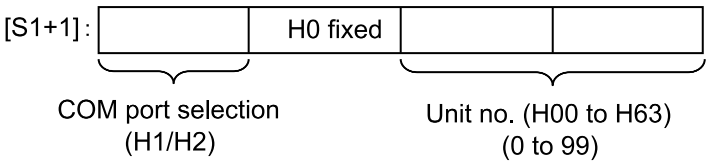
Specify the following for [S1]. The specification method differs depending on word transfer and bit transfer.





Specifying [S1+1]

Specify the following for [S1+1].



■ Specifying the transfer unit and method [S1]

Specify the quantity of send data in word units, or the location of target bits of send data in bit units.

*In word units, the maximum number of words is 127 (7Fh) because the transmission range of data is up to 254 bytes.

■ Specifying the partner unit [S1+1]

Specify the unit number of a partner unit. When H00 is specified, global transfer is applied (noresponse). Specify the source port from which data is sent to the partner unit as COM1.

■ Specifying the master unit area that stores send data [S2]

Specify the memory area of a master unit that stores send data.

■ Specifying the partner unit storage area [D,] [n]

Specify "0" for [D] as the device No. Specify the memory area of a partner unit that stores sent data, combining type [D] and address [n].

Example 1) [S2]: DT0, [n]: K100

↓

DT100

- A MODBUS command is created according to the operands specified by [S1], [S1+1], [S2], [D], and [n].

  During transfer in word units

  Command 06 (DT1 word write), command 15 (multiple point Y/R write), and command 16(DT multiple word write) can be sent.

  During transfer in bit units

  Command 05 (single Y/R write) can be sent.
- The created MODBUS command is sent with 2 bytes of CRC added to the end.

■ Examples of specifying MODBUS commands

Sending command 05 (single Y/R write)

- Example 1) To transfer the value of bit 0 of WR3 to the 1st bit of WY1 of partner unit number 7 from COM1

[F145 (SEND), DT10, WR3, WY0, K1]

[S1]:

H8	H1	H0 fixed	H0	DT10
Transfer in bit units	Destination bit no. (H0 to HF)		Master unit bit no. (H0 to HF)	

[S1+1]:

H1	H0 fixed	H0	H7	DT11
COM port selection		Unit no. (H00 to H63) (0 to 99)		

[S1]: DT10 (DT10=8100H, DT11=1007H)
[S2]: WR3 (WR3=0007H)
[D]: WY0
[n]: K1

Command conversion

MODBUS command

1	Slave address	07
2	Command (05H)	05
3	Coil number (H)	00
4	Coil number (L)	11
5	Set state (H)	FF
6	Set state (L)	00
7	CRC16(H)	DC
8	CRC16(L)	59

(Note 1) To send command 05, specify bit units (H8) for the transfer method in [S1].

(Note 2) The value of WR3 bit 0 is read and, depending on whether it is ON or OFF, the set state of [S1+1] is set.
Set FF00 if ON, or 0000 if OFF.

(Note 3) Y11 is specified for the coil number of the write destination (partner unit).

Sending command 06 (DT1 word write)

- Example 2) To transfer the 1-word data of WR3 to the DT1000 of partner unit number 7 from COM1

[F145 (SEND), DT10, WR3, DT0, K1000]

[S1]:

H0	H0	H0	H1	DT10
Transfer in word units	Specifies the send word count (H1 to H7F)			

[S1+1]:

H1	H0 fixed	H0	H7	DT11
COM port selection		Unit no. (H00 to H63) (0 to 99)		

[S1]: DT10 (DT10=0001H, DT11=1007H)
[S2]: WR3 (WR3=1234H)
[D]: DT0
[n]: K1000

Command conversion

MODBUS command

1	Slave address	07
2	Command (06H)	06
3	Write start number (H)	03
4	Write start number (L)	E8
5	Write data (H)	12
6	Write data (L)	34
7	CRC16(H)	04
8	CRC16(L)	AB

(Note 1) When sending command 06, specify word units (H0) for the transfer method in [S1], and (H1) for the send word count.

(Note 2) The word data is read from WR3 and set in the write data.

Sending command 15 (multiple point Y/R write)

- Example 3) To transfer from COM1 the 64-bit data starting from bit 0 of WR3 to bit F of WR6 to Y0 to Y3F of partner unit number 7

[F145 (SEND), DT10, WR3, WY0, K0]

[S1]:

H0	H0	H0	H4	DT10
Transfer in word units	Specifies the send word count (H1 to H7F)			

[S1+1]:

H1	H0 fixed	H0	H7

DT11

COM port selection — Unit no. (H00 to H63) (0 to 99)

[S1]: DT10 (DT10=0004H, DT11=1007H)
[S2]: WR3 (WR3=3210H
WR4=7654H
WR5=BA98H
WR6=FEDCH)
[D]: WY0
[n]: K0

Command conversion

MODBUS command

1	Slave address	07
2	Command (0FH)	0F
3	State change startnumber (H)	00
4	State change start number (L)	00
5	Change coil count (H)	00
6	Change coil count (L)	40
7	Data count (byte count)	08
8	Set data 1	10
9	Set data 2	32
10	Set data 3	54
11	Set data 4	76
12	Set data 5	98
13	Set data 6	BA
14	Set data 7	DC
15	Set data 8	FE
16	CRC16(H)	6C
17	CRC16(L)	B3

(Note 1) When sending command 15, specify word units (H0) for the transfer method in [S1].

(Note 2) For the state change start number, the coil number of the write destination is specified (partner unit).

(Note 3) For the change coil count, the write bit number is changed to HEX.

(Note 4) The maximum change coil count is 2032 (07F0H) due to the limitations of the MODBUS protocol.
The data count (byte count) is calculated as 8 coils being 1 data (1 byte). [Maximum 254 (FEH) bytes]

Command 16 (DT multiple words write) send

- Example 4) To transfer from COM1 the 3-word data starting from WR3 to WR5 to DT500 to DT502 of partner unit number 7

[F145 (SEND), DT10, WR3, DT0, K500]

[S1]:

H0	H0	H0	H3

DT10

Transfer in word units — Specifies the send word count (H1 to H7F)

[S1]:

H1	H0 fixed	H0	H7	DT11

COM port selection (H1)

Unit no. (H00 to H63) (0 to 99) (H0 H7)

[S1]: DT10 (DT10=0003H, DT11=1007H)
[S2]: WR3 (WR3=0011H
WR4=2233H
WR5=4455H)
[D]: DT0
[n]: K500

Command conversion

MODBUS command

1	Slave address	07
2	Command (10H)	10
3	Write start number (H)	01
4	Write start number (L)	F4
5	Write register count (H)	00
6	Write register count (L)	03
7	Data count (byte count)	06
8	Write data 1 (H)	00
9	Write data 1 (L)	11
10	Write data 2 (H)	22
11	Write data 2 (L)	33
12	Write data 3 (H)	44
13	Write data 3 (L)	55
14	CRC16(H)	5A
15	CRC16(L)	E7

(Note 1) When sending command 16, specify word units (H0) for the transfer method in [S1].

(Note 2) The maximum write register count is 127 (7FH) due to the limitations of the MODBUS protocol.

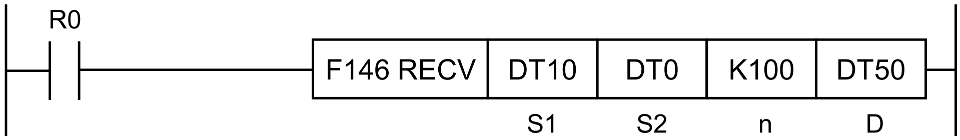
(Note 3) The data count (byte count) is calculated as the write register count being 2 bytes. [Maximum 254 (FEH) bytes]

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the control data value in [S1] and [S1+1] is outside the specified range.
	Turns ON when acquiring the number of words specified in [S1] exceeds the [S2] or [D] area during transfer in word units.
	Turns ON when [D]+[n] exceeds the [D] area.
	Turns ON when the control data COM port setting specified in [S1+1] is not MODBUS mode.
	Turns ON when the [D] area is DT during transfer in bit units.
	Turns ON when the [D] device No. is not 0.

21.5 F146 RECV［Data Receive Instruction（MODBUS Master Mode）］

■ Instruction format



■ Operands

<table>
<tr><th>Operand</th><th colspan="3">Setting</th></tr>
<tr><td rowspan="3">S1</td><td colspan="3">Specify the starting No. of the area (two words) that stores control data.</td></tr>
<tr><td>S1</td><td colspan="2">Specify a transfer method.
Word transfer: Specify the number of transferred words.
Bit transfer: Specify master unit bit No. and partner unit bit No.</td></tr>
<tr><td>S1+1</td><td colspan="2">Specify the COM port number of the master unit and the unit number of the partner unit.</td></tr>
<tr><td>S2</td><td colspan="2">Specification of partner unit</td><td>Specify the source data area of a partner unit. (Device No. is fixed to “0”)</td></tr>
<tr><td>n</td><td colspan="2">Specification of partner unit</td><td>Specify the starting address of the device in the source data area of a partner unit.</td></tr>
<tr><td>D</td><td colspan="2">Specification of master unit</td><td>Specify the device starting address of the receive data storage area in the master unit.</td></tr>
</table>

■ Devices that can be specified (indicated by •)

Operand	WX	WY	WR	WL	SV	EV	DT	LD	In	SWR	SDT	Constant		Index modifier (Note 1)
												K	H	
S1	•	•	•	•	•	•	•	•		•	•			•
S2	•	•	•	•			•	•						
n		•	•	•	•	•	•	•				•	•	•
D		•	•	•	•	•	•	•						•

(Note 1) A character constant cannot be specified.

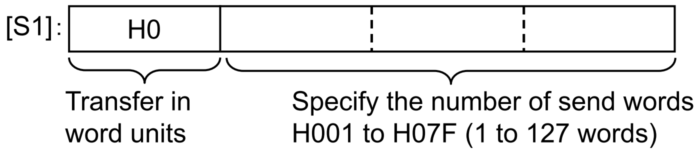
■ Outline of operation

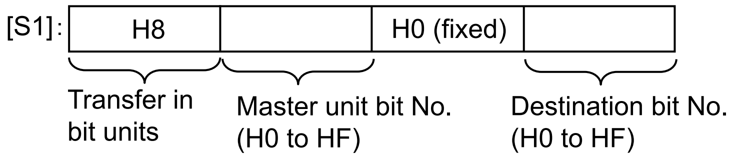
- It is used to send commands to the serial port (COM1) of the specified unit in the MODBUS mode connecting the unit that enables to receive the MODBUS command. (MODBUS command 01,02, 03 and 04).
- The data is sent from the area of the remote unit specified by [S2] and [n], and is stored in the area of the local unit that starts with [D], according to the specification for the 2−word data stored in the control data that starts with the area specified by [S1].

■ Specifying control data [S1]

Specifying [S1]

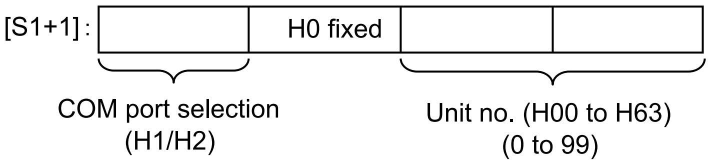
Specify the following for [S1]. The specification method differs depending on word transfer and bit transfer.





Specifying [S1+1]

Specify the following for [S1+1].



■ Specifying the transfer unit and method [S1]

Specify the quantity of send data in word units, or the location of target bits of send data in bit units.

*In word units, the maximum number of words is 127 (7Fh) because the transmission range of data is up to 254 bytes.

■ Specifying the partner unit [S1+1]

Specify the unit number of a partner unit.

Specify the destination port of the partner unit as COM1.

■ Specifying the partner unit area that receives data [S2], [n]

Specify "0" for [S2] as the device No. Specify the memory area of a partner unit that stores sent data, combining type [S2] and address [n].

Example 1) [S2]: DT0, [n]: K100

↓

DT100

■ Specifying the master unit area that stores received data [D]

Specify the memory area of the master unit that stores received data.

- A MODBUS command is created according to the operands specified by [S1], [S1+1], [S2], [D], and [n].

  During transfer in word units

  Command 01 (Y/R coil read), command 02 (X contact reading), command 03 (DT read),and command 04 (WL/LD read) can be sent.

  During transfer in bit units

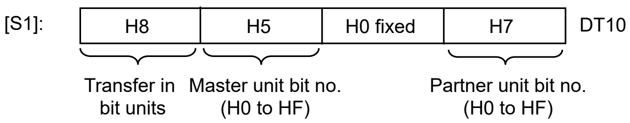
  Command 01 (Y/R coil read) and command 02 (X contact reading) can be sent.
- The created MODBUS command is sent with 2 bytes of CRC added to the end.

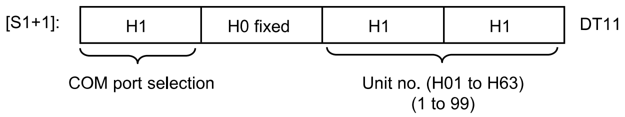
■ Examples of specifying MODBUS commands

Sending command 01 (Y/R coil read)

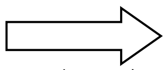
- Example 1) Sending the command from COM1 to read 1 bit of Y17 from partner unit number 17 and transfer the read bit data to the 5th bit of DT100 in the master unit

[F146(RECV), DT10, WY0, K1, DT100]





[S1]: DT10 (DT10=8507H, DT11=1011H)
[S2]: WY0
[n]: K1
[D]: DT100



Command conversion

MODBUS command

1	Slave address	11
2	Command (01H)	01
3	Read start number (H)	00
4	Read start number (L)	17
5	Read count (H)	00
6	Read count (L)	01
7	CRC16(H)	DC
8	CRC16(L)	59

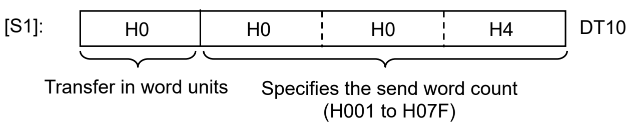
(Note 1) To use command 01 to read 1 bit only, specify bit units (H8) for the transfer method in [S1].

(Note 2) For the read start number, specify the read destination coil number (partner unit: Y17).

(Note 3) The read count is 1.

- Example 2) Sending the command from COM1 to read 64 bits (4 words) from Y10 to Y4F from the partner unit number 17 and transfer the read data to the master unit area starting from DT100

[F146(RECV), DT10, WY0, K1, DT100]



[S1+1]:	H1	H0 fixed	H1	H1	DT11
	COM port selection		Unit no. (H01 to H63) (1 to 99)		

[S1]: DT10 (DT10=0004H, DT11=1011H)
[S2]: WY0
[n]: K1
[D]: DT100

Command conversion

MODBUS command

1	Slave address	11
2	Command (01H)	01
3	Read start number (H)	00
4	Read start number (L)	10
5	Read count (H)	00
6	Read count (L)	40
7	CRC16(H)	3E
8	CRC16(L)	AF

(Note 1) To use command 01 to read word units, specify bit units (H0) for the transfer method in [S1].

(Note 2) For the read start number, specify the read destination coil number (partner unit: Y10).

(Note 3) The read count is the specified number of words × 16 (64-bit reading).

Sending command 02 (X contact reading)

- Example 3) Sending the command from COM1 to read 1 bit of X17 from partner unit number 17 and transfer the read bit data to the 5th bit of DT100 in the master unit

[F146(RECV), DT10, WX0, K1, DT100]

[S1]:	H8	H5	H0 fixed	H7	DT10
	Transfer in bit units	Master unit bit no. (H0 to HF)		Partner unit bit no. (H0 to HF)	

[S1+1]:	H1	H0 fixed	H1	H1	DT11
	COM port selection		Unit no. (H01 to H63) (1 to 99)		

[S1]: DT10 (DT10=8507H, DT11=1011H)
[S2]: WX0
[n]: K1
[D]: DT100

Command conversion

MODBUS command

1	Slave address	11
2	Command (02H)	02
3	Read start number (H)	00
4	Read start number (L)	17
5	Read count (H)	00
6	Read count (L)	01
7	CRC16(H)	0B
8	CRC16(L)	5E

(Note 1) To use command 02 to read 1 bit only, specify bit units (H8) for the transfer method in [S1].

(Note 2) For the read start number, specify the read destination coil number (partner unit: X17).

(Note 3) The read count is 1.

- Example 4) Sending the command from COM1 to read 64 bits (4 words) from X10 to X4F from the partner unit number 17 and transfer the read data to the master unit area starting from DT100

[F146(RECV), DT10, WX0, K1, DT100]

[S1]:	H0	H0	H0	H4	DT10
	Transfer in word units	Specifies the send word count (H001 to H07F)			

[S1+1]:	H1	H0 fixed	H1	H1	DT11
	COM port selection		Unit no. (H01 to H63) (1 to 99)		

[S1]: DT10 (DT10=0004H, DT11=1011H)
[S2]: WX0
[n]: K1
[D]: DT100

Command conversion

MODBUS command

1	Slave address	11
2	Command (02H)	02
3	Read start number (H)	00
4	Read start number (L)	10
5	Read count (H)	00
6	Read count (L)	40
7	CRC16(H)	7A
8	CRC16(L)	A0

(Note 1) To use command 02 to read word units, specify bit units (H0) for the transfer method in [S1].

(Note 2) For the read start number, specify the read destination coil number (partner unit: X10).

(Note 3) The read count is the specified number of words × 16 (64-bit reading).

Send Command 03 (Read DT)

- Example 5) Sending the command from COM1 to read 6 words (DT500 to DT505) from partner unit number 17 and transfer the read data to the area starting from DT100 in the master unit

[F146 (RECV), DT10, DT0, K500, DT100]

[S1]:	H0	H0	H0	H6	DT10
	Transfer in word units	Specifies the send word count (H001 to H07F)			

[S1+1]:	H1	H0 fixed	H1	H1	DT11
	COM port selection		Unit no. (H01 to H63) (1 to 99)		

[S1]: DT10 (DT10=0006H, DT11=1011H)
[S2]: DT0
[n]: K500
[D]: DT100

Command conversion

MODBUS command

1	Slave address	11
2	Command (03H)	03
3	Read start number (H)	01
4	Read start number (L)	F4
5	Read count (H)	00
6	Read count (L)	06
7	CRC16(H)	87
8	CRC16(L)	56

(Note 1) To use command 03 to read word units, specify word units (H0) for the transfer method in [S1].

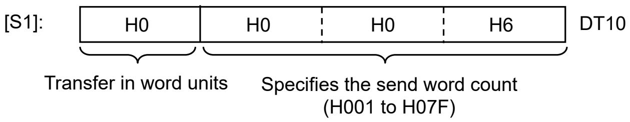
(Note 2) For the read start number, specify the read destination data number (partner unit: DT500).

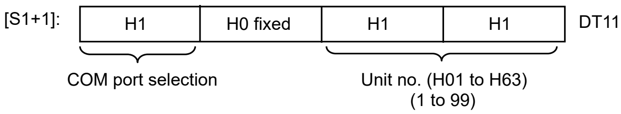
(Note 3) The read count is the specified number of words (6-word reading).

Sending command 04 (WL/LD read)

- Example 6) Sending the command from COM1 to read 6 words (WL20 to WL25) from partner unit number 17 and transfer the read data to the area starting from DT100 in the master unit

[F146(RECV), DT10, WL0, K20, DT100]





[S1]: DT10 (DT10=0006H, DT11=1011H)
[S2]: WL0
[n]: K20
[D]: DT100

Command conversion

MODBUS command

1	Slave address	11
2	Command (04H)	04
3	Read start number (H)	00
4	Read start number (L)	14
5	Read count (H)	00
6	Read count (L)	06
7	CRC16(H)	32
8	CRC16(L)	9C

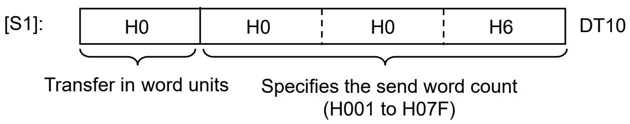
(Note 1) To use command 04 to read word units, specify word units (H0) for the transfer method in [S1].

(Note 2) For the read start number, specify the read destination data number (partner unit: WL20).

(Note 3) The read count is the specified number of words (6-word reading).

- Example 7) Sending the command from COM1 to read 6 words (LD100 to LD105) from partner unit number 17 and transfer the read data to the area starting from DT100 in the master unit

[F146(RECV), DT10, LD0, K100, DT100]



[S1+1]:	H1	H0 fixed	H1	H1	DT11
	COM port selection		Unit no. (H01 to H63) (1 to 99)		

[S1]: DT10 (DT10=0006H, DT11=1011H)
[S2]: LD0
[n]: K100
[D]: DT100

Command conversion

MODBUS command

1	Slave address	11
2	Command (04H)	04
3	Read start number (H)	08
4	Read start number (L)	34
5	Read count (H)	00
6	Read count (L)	06
7	CRC16(H)	31
8	CRC16(L)	36

(Note 1) To use command 04 to read word units, specify word units (H0) for the transfer method in [S1].

(Note 2) For the read start number, specify the read destination data number (partner unit: LD100).

(Note 3) The read count is the specified number of words (6-word reading).

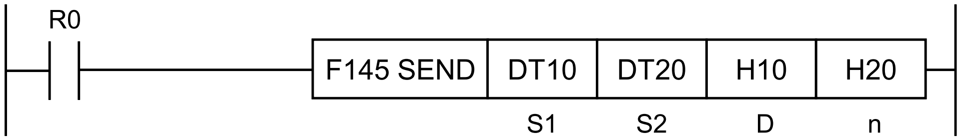
(Note 4) When LD is specified, it is from 07D0H (LD0).

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the control data value in [S1] and [S1+1] is outside the specified range.
	Turns ON when acquiring the number of words specified in [S1] exceeds the [S2] or [D] area during transfer in word units.
	Turns ON when [S2]+[n] exceeds the [S2] area.
	Turns ON when the control data COM port setting specified in [S1+1] is not MODBUS mode.
	Turns ON when the [S2] area is DT/WL/LD during transfer in bit units.
	Turns ON when the [S2] device number is not 0.

21.6 F145 SEND［Data Send Instruction（MODBUS Master II: Type Directly Specifying MODBUS Address)］

■ Instruction format



■ Operands

Item	Setting	
S1	Specify the COM port number of a master unit, the send MODBUS command, and the partner unit No.	
S2	Specification of master unit	Operation memory area storing data to be sent
D	Specification of partner unit	MODBUS address specification (Specification range: H0 to HFFFF)
n	Specification of partner unit	Specify the number of send data. Specification range: 1 to 127 words (word specification), 1 to 2040 bits (bit specification)

■ Devices that can be specified (indicated by ●)

Operand	WX	WY	WR	WL	SV	EV	DT	LD	In	SWR	SDT	Constant		Index modifier (Note 1)
												K	H	
S1	●	●	●	●	●	●	●	●		●	●	●	●	●
S2	●	●	●	●	●	●	●	●		●	●			●
D		●	●	●	●	●	●	●				●	●	
n		●	●	●	●	●	●	●				●	●	●

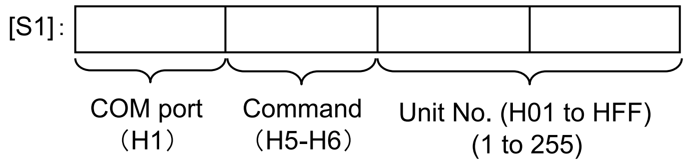
(Note 1) A character constant cannot be specified.

■ Outline of operation

- This instruction sends MODBUS commands to the MODBUS address specified by D, with the sending port, sending command (5 or 6), and partner unit number specified by [S1], and the send data specified by [S2]. (MODBUS commands: 05, 06, 15, 16).
- This instruction has a feature that data can be sent with only one instruction.

■ Specifying [S1]

- S1]: Specifying port number, transmission command and destination unit number



COM port selection (upper 4 bits)	H1
Port number	COM1

■ Specifying a COM port

Specify H1 for COM1 port .

■ Specifying the sending command

If multiple points are specified by n, where H5 (HD) is bit data and H6 (HE) is word data, the data is automatically converted into command 15 or 16 for sending multiple bits or multiple words.

Specifying HD or HE allows this instruction to issue multi-point writing commands (0F or 10) even for one-bit or one-word transfer.

■ Specifying the partner unit number

- Specify the area that stores send data in [S2].
  Specify the starting number of the master unit operation memory that stores send data.
  This instruction can be executed even if the send command specified in S1 and the device type of the source are not the same.
  In other words, it is possible to specify bit data to send the content of DT or word data to send the content of WR.
  When bit data is specified in S1, the data is always sent starting from bit 0.
- Specify the MODBUS address of a partner unit that receives the send data in [D]. Addresses that can be specified: H0 to HFFFF
- Specify the number of send data in [n]. Number of data that can be specified:
  In the case of bit data, the max. number is 2040 (07F8H).
  In the case of word data, the max. number is 127 (7FH).

■ Specifying a partner unit

When “0” is specified for the partner unit number, global transfer is applied.

In this case, there is no response message from the partner unit.

■ Examples of specifying MODBUS commands

Command 05 (Force single coil)

- Example 1) To transfer from COM1 the value of bit 0 of WR3 to the bit address H7788 of unit number 7

[F145 (SEND), H1507, WR3, H7788, K1]

[S1]:

H1	H5	H0	H7

COM port selection (H1) — Unit no. (H00 to HFF) (H0 H7)

[S1]: H1507
[S2]: WR3 (WR3=0007H)
[D]: H7788
[n]: K1

Command conversion

MODBUS command

1	Slave address	07
2	Command (05H)	05
3	Coil number (H)	77
4	Coil number (L)	88
5	Set state (H)	FF
6	Set state (L)	00
7	CRC16(H)	17
8	CRC16(L)	C2

(Note 1) The value of WR3 bit 0 is read and, depending on whether it is ON or OFF, the set state is set. Set FF00 if ON, or 0000 if OFF.

Command 06 (Preset single register)

- Example 2) To transfer from COM1 the 1-word data from WR3 to the address H7788 of unit number 7

[F145 (SEND), H1607, WR3, H7788, K1]

[S1]:

H1	H6	H0	H7

COM port selection (H1) — Unit no. (H00 to HFF) (H0 H7)

[S1]: H1607
[S2]: WR3 (WR3=1234H)
[D]: H7788
[n]: K1

Command conversion

MODBUS command

1	Slave address	07
2	Command (06H)	06
3	Write start number (H)	77
4	Write start number (L)	88
5	Write data (H)	12
6	Write data (L)	34
7	CRC16(H)	53
8	CRC16(L)	C2

(Note 1) The word data is read from WR3 and set in the write data.

Command 15 (Force multiple coils)

- Example 3) To transfer from COM1 the 64-bit data from bit 0 of WR3 to bit F of WR6 to the bit address H7788 of partner unit number 7

[F145 (SEND), H1507, WR3, H7788, K64]

[S1]:

H1	H5	H0	H7

$\underbrace{\text{H1}}_{\text{COM port selection}}$ $\underbrace{\text{H0 H7}}_{\text{Unit no. (H00 to HFF)}}$

[S1]: H1507
[S2]: WR3 (WR3=3210H
WR4=7654H
WR5=BA98H
WR6=FEDCH)
[D]: H7788
[n]: K64

Command conversion

MODBUS command

1	Slave address	07
2	Command (0FH)	0F
3	State change start number (H)	77
4	State change start number (L)	88
5	Change coil count (H)	00
6	Change coil count (L)	40
7	Data count (byte count)	08
8	Set data 1	10
9	Set data 2	32
10	Set data 3	54
11	Set data 4	76
12	Set data 5	98
13	Set data 6	BA
14	Set data 7	DC
15	Set data 8	FE
16	CRC16(H)	3B
17	CRC16(L)	65

(Note 1) When multiple points are specified with n, the command is automatically corrected.

The state change start number at H5 bit single write => H15 bit multiple write is H7788 (partner unit)

For the change coil count, the write bit count is changed to HEX. The maximum change coil count is 2040 (07F8H) (due to the limitations of the MODBUS protocol)

The data count (byte count) is calculated as 8 coils being 1 data (1 byte). [Maximum 255 (FFH) bytes]

Command 16 (Preset multiple registers)

- Example 4) To transfer from COM1 the 3-word data from DT3 to DT5 to the address H7788 of partner unit number 7

[F145 (SEND), H1607, DT3, H7788, K3]

[S1]:

H1	H6	H0	H7

COM port selection (H1, H6)　Unit no. (H00 to HFF) (H0, H7)

[S1]: H1607
[S2]: DT3 (DT3=0011H
DT4=2233H
DT5=4455H)
[D]: H7788
[n]: K3

Command conversion

MODBUS command

1	Slave address	07
2	Command (10H)	10
3	Write start number (H)	77
4	Write start number (L)	88
5	Write register count (H)	00
6	Write register count (L)	03
7	Data count (byte count)	06
8	Write data 1 (H)	00
9	Write data 1 (L)	11
10	Write data 2 (H)	22
11	Write data 2 (L)	33
12	Write data 3 (H)	44
13	Write data 3 (L)	55
14	CRC16(H)	2C
15	CRC16(L)	BA

(Note 1) When multiple points are specified with n, the command is automatically corrected.

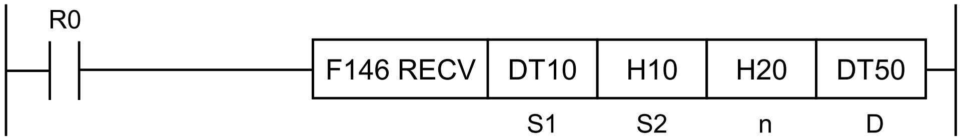
(Note 2) The maximum write register count is 127 (7FH) (due to the limitations of the MODBUS protocol).
The data count (byte count) is calculated as the write register count being 2 bytes. [Maximum 254 (FEH) bytes]

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the [S1] control data value is outside the specification range.
	Turns ON when the COM port specification for control data specified by [S1] is not MODBUS mode.
	Turns ON when the number of send data [n] is 0.
	Turns ON when the number of send data is negative.
	Turns ON when the number of send data [n] exceeds the area of operation memory specified by [S2].
	Turns ON when the number of send data [n] exceeds limitations in the MODBUS specifications.

21.7 F146 RECV［Data Receive Instruction（MODBUS Master Mode II: Type Directly Specifying MODBUS Address）］

■ Instruction format



■ Operands

Item	Setting	
S1	Specify the COM port number of a master unit, the send MODBUS command, and the partner unit No.	
S2	Specification of partner unit	MODBUS address specification (Specification range: H0 to HFFFF)
n	Specification of partner unit	Specify the number of received data. Specify the number of words for resistor transfer. (Specification range: 1 to 127 words) Specify the number of bits for bit transfer. (Specification range: 1 to 2040 bits)
D	Specification of master unit	Operation memory area storing received data

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	In	SWR	SDT	Constant		Index modifier (Note 1)
												K	H	
S1	●	●	●	●	●	●	●	●		●	●	●	●	●
S2	●	●	●	●			●	●				●	●	
n		●	●	●	●	●	●	●				●	●	●
D		●	●	●	●	●	●	●						●

(Note 1) A character constant cannot be specified.

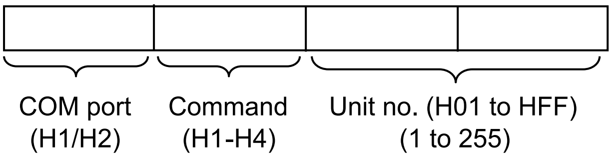
■ Outline of operation

- This instruction sends a volume of data specified by n from the MODBUS address specified by S2 and stores it in the operation memory specified by D, with the sending port, sending command (1 or 2), and the partner unit number specified by S1. It sends MODBUS commands. (MODBUS commands: 01, 02, 03, and 04)
- This instruction has a feature that data can be sent with only one instruction.

■ Specifying [S1]

- S1]: Specifying port number, transmission command and destination unit number

[S1]:



COM port selection (upper 4 bits)	H1
Port number	COM1

■ Specifying a COM port

Specify H1 for COM1 port .

■ Specifying the sending command

H1, H2, H3 or H4 can be specified.

■ Specifying the partner unit number

H1 to HFF can be specified.

- Specify the MODBUS address of a partner unit that receives the send data in [S2].
  Addresses that can be specified: H0 to HFFFF
- Specify the number of received data in [n].
  In the case of bit data, the max. number of data that can be specified is 2040 (07F8H).
  In the case of word data, the max. number is 127 (7FH).
- Specify the area that stores received data.
  Specify the starting number of the master unit operation memory that stores received data. This instruction can be executed even if the send command specified in S1 and the device type of the source are not the same. In other words, it is possible to specify bit data to send the content of DT or word data to send the content of WR. When command 1 or 2 is specified, data is always stored from bit 0 of D.

■ Examples of specifying MODBUS commands

Command 01 (Read coil state)

- Example 1) To read one bit from bit address H7788 of unit number 17 connected to COM1 and write it to bit 0 of DT100 of the master unit

[F146(RECV), H1111, H7788, K1, DT100]

[S1]:

H1	H1	H1	H1
COM port selection		Unit no. (H01 to HFF)	

[S1]: H1111
[S2]: H7788
[n]: K1
[D]: DT100

Command conversion

MODBUS command

1	Slave address	11
2	Command (01H)	01
3	Read start number (H)	77
4	Read start number (L)	88
5	Read count (H)	00
6	Read count (L)	01
7	CRC16(H)	64
8	CRC16(L)	C4

- Example 2) To read 64 bits (four words) from bit address H7788 of unit number 17 connected to COM1 and write it to the master unit area starting from bit 0 of DT100

[F146 (RECV), H1111, H7788, K64, DT100]

[S1]:

H1	H1	H1	H1
COM port selection		Unit no. (H01 to HFF)	

[S1]: H1111
[S2]: H7788
[n]: K64
[D]: DT100

Command conversion

MODBUS command

1	Slave address	11
2	Command (01H)	01
3	Read start number (H)	77
4	Read start number (L)	88
5	Read count (H)	00
6	Read count (L)	40
7	CRC16(H)	A4
8	CRC16(L)	F4

Command 02 (Read input state)

- Example 3) To read one bit from bit address H7788 of unit number 17 connected to COM1 and write it to bit 0 of DT100 of the master unit

[F146(RECV), H1211, H7788, K1, DT100]

[S1]:

H1	H2	H1	H1
COM port selection		Unit no. (H01 to HFF)	

[S1]: H1211
[S2]: H7788
[n]: K1
[D]: DT100

Command conversion

MODBUS command

1	Slave address	11
2	Command (02H)	02
3	Read start number (H)	77
4	Read start number (L)	88
5	Read count (H)	00
6	Read count (L)	01
7	CRC16(H)	20
8	CRC16(L)	C4

- Example 4) To read 64 bits (four words) from bit address H7788 of unit number 17 connected to COM1 and write it to the master unit area starting from bit 0 of DT100

[F146 (RECV), H1211, H7788, K64, DT100]

[S1]:

H1	H2	H1	H1

COM port selection (H1 H2) Unit no. (H01 to HFF) (H1 H1)

[S1]: H1211
[S2]: H7788
[n]: K64
[D]: DT100

Command conversion

MODBUS command

1	Slave address	11
2	Command (02H)	02
3	Read start number (H)	77
4	Read start number (L)	88
5	Read count (H)	00
6	Read count (L)	40
7	CRC16(H)	E0
8	CRC16(L)	F4

Command 03 (Read hold register)

- Example 5) Read six words from address H7788 of unit number 17 connected to COM1 and write it to the master unit area starting from DT100

[F146(RECV), H1311,H7788, K6, DT100]

[S1]:

H1	H3	H1	H1

COM port selection (H1 H3) Unit no. (H01 to HFF) (H1 H1)

[S1]: H1311
[S2]: H7788
[n]: K6
[D]: DT100

Command conversion

MODBUS command

1	Slave address	11
2	Command (03H)	03
3	Read start number (H)	77
4	Read start number (L)	88
5	Read count (H)	00
6	Read count (L)	06
7	CRC16(H)	5C
8	CRC16(L)	C6

Command 04 (Read input register)

- Example 6) Read six words from address H7788 of unit number 17 connected to COM1 and write it to the master unit area starting from DT100

[F146(RECV), H1411,H7788, K6, DT100]

[S1]: H1411
[S2]: H7788
[n]: K6
[D]: DT100

Command conversion

MODBUS command

1	Slave address	11
2	Command (04H)	04
3	Read start number (H)	77
4	Read start number (L)	88
5	Read count (H)	00
6	Read count (L)	06
7	CRC16(H)	E9
8	CRC16(L)	06

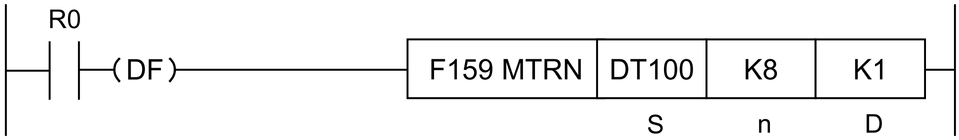
■ Flag operations

Name	Description
R9007 R9008	Turns ON when the [S1] control data value is outside the specification range.

Name	Description
(ER)	Turns ON when the COM port specification for control data specified by [S1] is not MODBUS mode.
	Turns ON when number of received data [n] is 0.
	Turns ON when the number of received data is negative.
	Turns ON when number of received data [n] exceeds the MODBUS specifications.
	Turns ON when the area of operation memory specified in [D] is exceeded when number of received data [n] is received.

21.8 F159 MTRN (General-purpose Communication Instruction)

■ Instruction format



■ Operands

Item	Setting
S	Starting area (data register) of the data table
n	Area storing the number of bytes of data to be sent, or constant data ● When the value is positive, a transmission end code is added ● When the value is negative, a transmission end code is not added ● When the value is H8000, the usage purpose of the COM port is changed
D	COM port number for sending data （K0/K1） K0: Tool port / K1: COM port

■ Devices that can be specified (indicated by ●)

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier (Note 1)
										K	H	
S							●					●
n	●	●	●	●	●	●	●	●	●	●	●	●
D										●		●

(Note 1) A character constant cannot be specified.

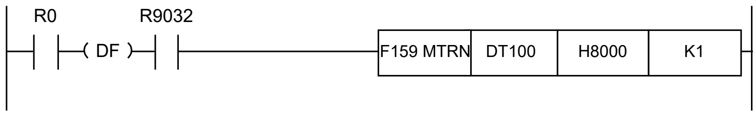
■ Outline of operation

This instruction is used to send and receive data between external devices connected to the COM ports. The F159 MTRN instruction has the following three functions.

Item	Description
Send	[n] bytes of data stored in the data table starting from the area specified by [S] are sent to an external device from the communication port specified by [D]. Data can be sent with the header and the terminator automatically added. The maximum number of transmission bytes is 2048.
Receive	Reception is controlled by the ON/OFF state of the received flag (R9038/R9048). If the received flag is OFF, data can be always received, and data sent to the RS-232C port is stored in data registers specified by system register numbers 416 to 419. The F159 (MTRN) instruction is used to turn OFF the received flag (R9038/R9048) to enable reception. The maximum number of received bytes is 4094.
RS-232C port usage purpose switching	The F159 instruction can be executed to switch the COM port between [general-purpose communication mode] and [computer link mode]. To execute the instruction, specify "H8000" in n (Number of send data bytes).

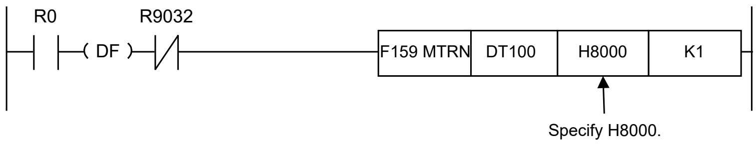
<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td></td><td>

● Switching from general-purpose port to computer link



R9032 or R9042: Turns ON when the RS-232C port selection flag "general-purpose port" is selected.

● Switching from computer link to general-purpose port



Note

● The COM port is set to the usage purpose selected in system register No. 412 when the power is turned ON. The tool port is always set to computer link mode in PROG mode.

</td></tr>
</table>

■ System register settings

- The COM port must be set to "general-purpose communication mode" by using a system register.
- The baud rate and transmission format must be set to match those of the connected external device by using a system register.
- To secure an area for storing received data in the data register (DT), it is required to specify “Receive buffer starting number in general-purpose communication” and “Receive buffer capacity in general-purpose communication” using the system register.

■ Related flags and system register numbers

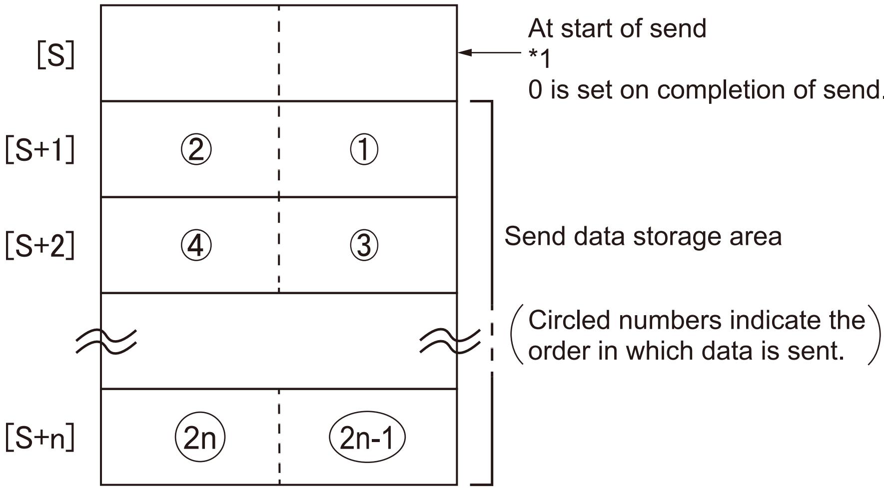
	Operation	COM1	TOOL
Communication error flag	0: Normal 1: Error	R9032	R9040
Received flag	0: Reception possible 1: Reception not possible (reception completed)	R9038	R903E
Sent flag	0: Sending 1: Sending completed (clear to send)	R9039	R903F
Receive buffer start	The range of DT used as the receive buffer is specified by a system register.	No.416	No.420
Receive buffer capacity		No.417	No.421

■ Programming and operation to send data

To send data, write the send data to the data table, specify the table in the F159 (MTRN) instruction, and execute the instruction.

Send data table

Use data that starts with the data register specified by [S] as the send data table.



- Write the send data into the send data storage area specified by [S] using the F0 (MV) instruction, F95 (ASC) instruction, etc.

1. Do not include an end code in the send data as it will be added automatically.
2. When the header (start code) is set to "STX" in system register No. 413 or 414, do not add the header to send data. The header is added automatically.
3. The maximum number of send data bytes [n] is 2048.

<Example> To send eight characters of A, B, C, D, E, F, G, and H (8-byte data)

The following example is a data table of DT100 to DT104.

DT100	K8		At start of send *1 0 is set at start of send.
DT101	H42(B)	H41(A)	Data is sent in order from the lower byte
DT102	H44(D)	H43(C)	
DT103	H46(F)	H45(E)	
DT104	H48(H)	H47(G)	

Data table before send

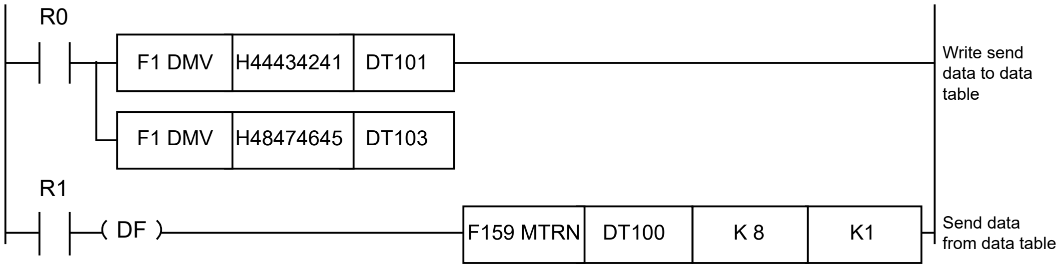
(Note 1) The number of send data is set.

Note

- When using a 1-channel RS-232C type communication cassette, data is sent only when CS (Permission to Send) is ON. If you are not connecting to a destination device, connect to RS (Request to Send).

Programming

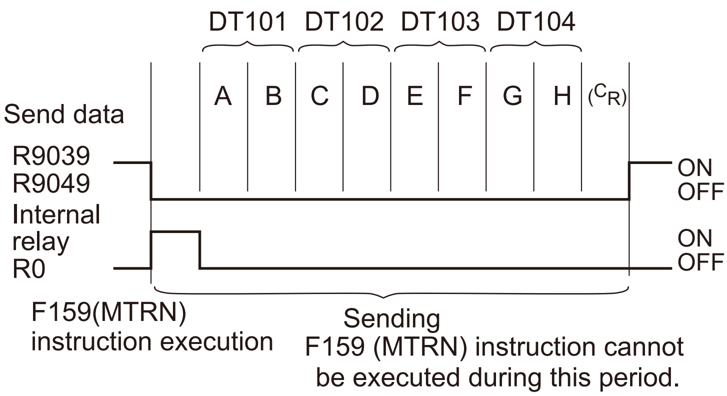
Specify the starting address of the send data table in [S] and the number of send data bytes in [n].



Operation

When the sent flag (R9039/R9049) is ON and the execution condition of the F159 (MTRN) instruction is ON, the operation is as follows.

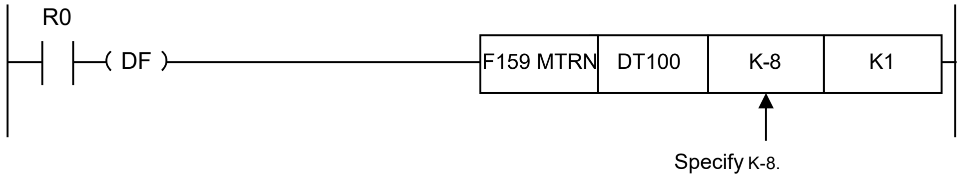
1. [n] is preset in [S]. In addition, the received flag R9038/R9048 is turned OFF and the number of received data is cleared to 0.
2. The set data is sent in order from the lower byte of [S+1] in the table.
   - While the data is being sent, the sent flag (R9039/R9049) is OFF.
   - If the header is set to "STX exists" in system register No. 413, the header is automatically added to the beginning of the data.
   - The terminator specified in system register No. 413 is automatically added to the end of the data.



3. When the specified volume of data has all been sent, the value of [S] changes to 0 and the sent flag (R9039/R9049) turns ON.

- If you do not want to add a terminator when sending data, specify so in either of the following ways.
  - Specify the number of send bytes as a negative value.
  - If you do not want to add a terminator for both sending and receiving, set the terminator to "None" in system register No. 413 or 414.

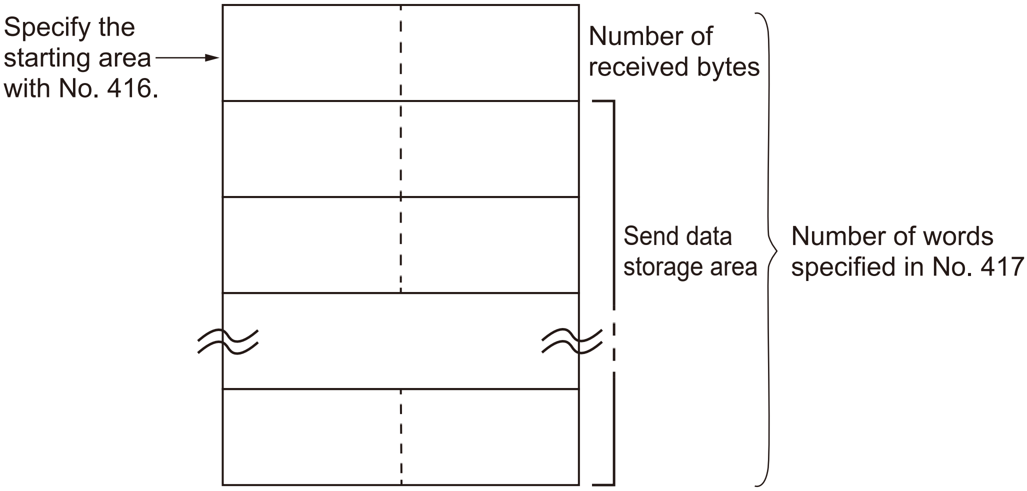
Programming for sending 8-byte data without a terminator



■ Preparation for receiving data

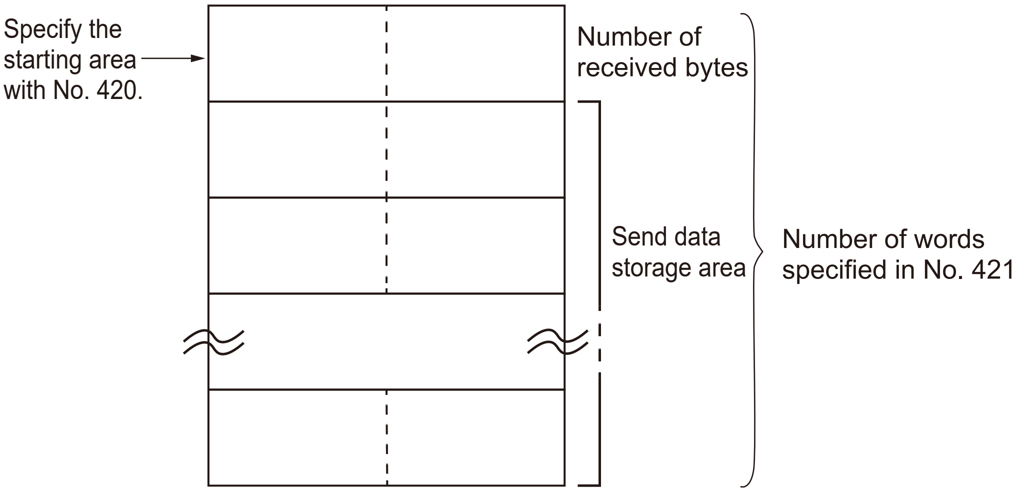
Receive buffer setting for COM port (No. 416, No. 417)

In the default setting, the data register area from DT0 to DT2047 is the receive buffer. The maximum allowable number of received bytes is 4094.



Receive buffer setting for tool port (Nos. 420, No. 421)

In the default setting, the data register area from DT4096 to DT6143 is the receive buffer. The maximum allowable number of received bytes is 4094.

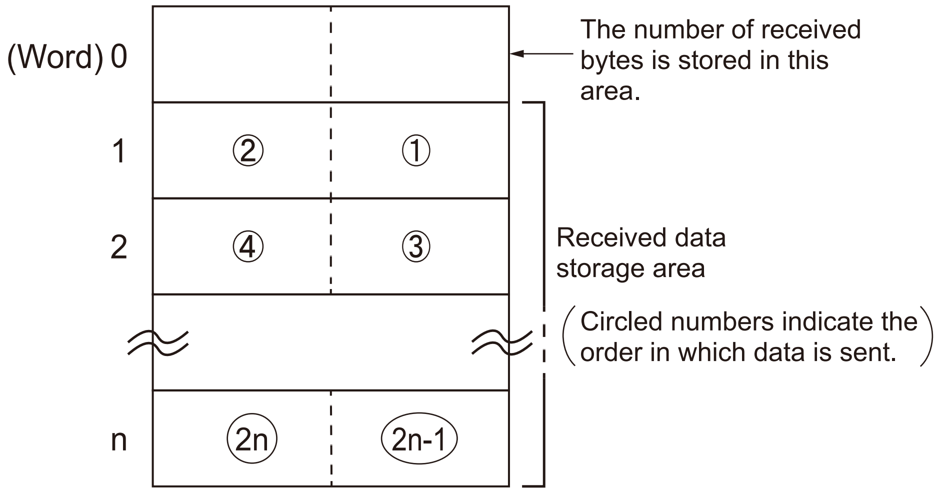


■ Programming and operation to receive data

Data sent from an external device connected to the RS-232C port is stored in data registers that are set as the receive buffer.

- The receive buffer uses the data registers. Specify the settings in system register No. 416 to 419.
- In the starting address of the receive buffer, the number of received data bytes is stored. The default value is 0.
- Received data is stored in the receive data storage area in order from the lower byte.

[Receive buffer] Uses the data registers

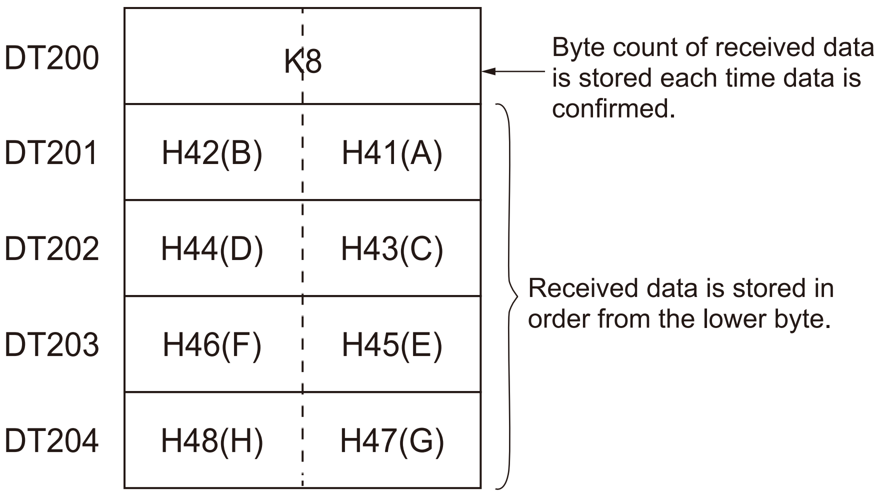


<Example> Receiving an 8-byte data of A, B, C, D, E, F, G, and H from an external device on COM1 port

Assume DT200 to DT204 as the receive buffer. The system register settings are as follows.

No.416: K200

No.417: K5

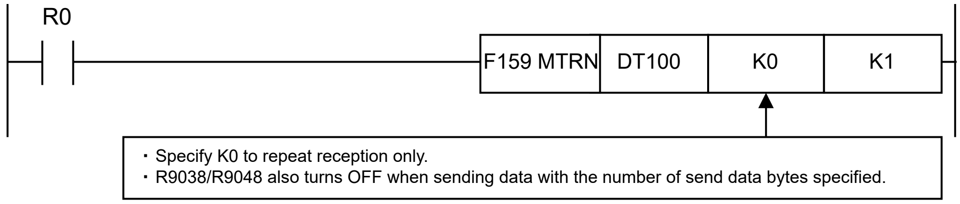


Receive buffer when receive complete

Programming

When reception of data from the external device is completed, the received flag (R9038/R9048) turns ON to disable reception of subsequent data.

To receive the next data, it is necessary to execute the F159 (MTRN) instruction to turn OFF the received flag (R9038/R9042) and clear the byte count to 0.

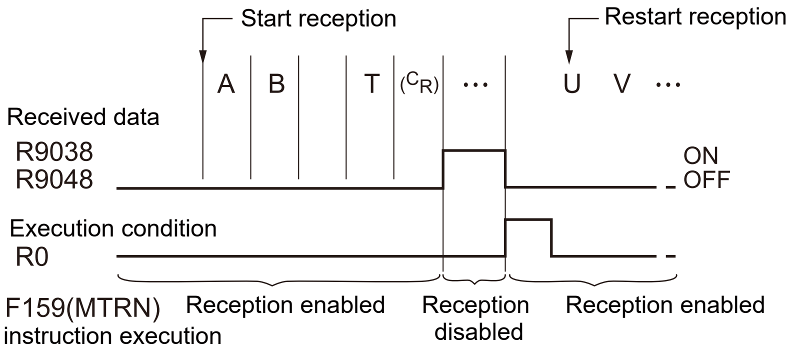


Operation

When the received flag (R9038/R9048) is OFF, the following operations are performed when data is received from an external device. (R9038/R9048 is OFF during the first scan after RUN. In addition, 0 is set in the starting area of the receive buffer specified by the system registers).

1. Received data is stored in order from the lower byte of the 2nd word area of the receive buffer.

   The header and terminator are not stored.



2. The received flag (R9038/R9048) turns ON when the terminator is received. Reception of the next data is then disabled.
3. When the F159 (MTRN) instruction is executed, the received flag (R9038/R9048) turns OFF, the number of received bytes is cleared, and the next data is stored in order from the lower byte.

Note

- To receive data repeatedly, refer to the following procedure.
  1. Receive data
  2. Reception completed (R9038/R9048: ON, reception disabled)
  3. Process received data
  4. Execute F159 (MTRN) instruction (R9038/R9048: OFF, reception enabled)
  5. Receive next data

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the data table of the number of bytes specified by [n] exceeds the area.

22 Sampling Trace Instructions

22.1 Sampling Trace

This is a function used to sample the ON/OFF status of registered contacts and the data stored in the registers, either periodically or when the appropriate conditions have been established, and store the results in memory. This function can be used to confirm changes in the data.

- 16 contacts and up to three words for registers can be registered.

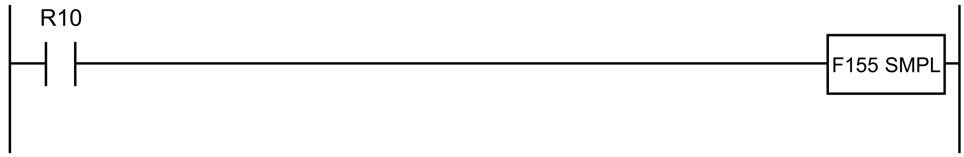
1 2 Procedure

1. Specify registration of the data to be sampled and the sampling method (such as the number of times or the time interval).
2. Instruct the sampling trace to begin.
3. Execute sampling.
   Sampling can be executed as periodic sampling or according to the F155 (SMPL) instruction.
4. Stop the sampling trace.
   Apply a stop command trigger by using a programming tool software online operation or by executing the F156 (STRG) instruction. When the trigger is applied, the sampling trace is stopped after sampling of the specified delay count is performed. The programming tool software can also be used to initiate a forced stop.)
5. The programming tool software can be used to read the sampling results from the control unit, and to monitor and confirm them.

22.2 F155 SMPL (Sample Set Data)

Performs sampling when a sampling trace is executed.

■ Instruction format



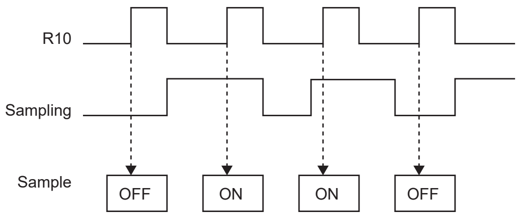
■ Outline of operation

- During a sampling trace, sampling is performed on the specified data (contacts and registers), and the executed data content is stored in the sampling trace memory.
- If the sampling trace settings and startup have not been specified by using the programming tool software, processing is not performed even if the internal relay condition is established.

■ Operation example

Operation of instruction format description program

When the internal relay R10 is ON, sampling is performed on previously registered contacts or registers.

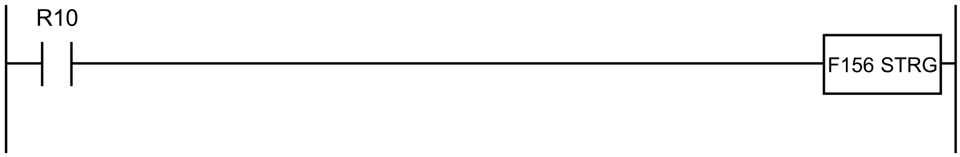


Registration of the data to be sampled, specification of the sampling method (such as the cable and the time interval), and specification of the command to start a sampling trace can only be performed by using the programming tool software.

22.3 F156 STRG (Sampling Stop Trigger)

Applies a stop command trigger during sampling trace execution.

■ Instruction format



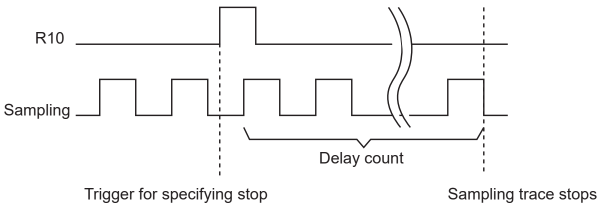
■ Outline of operation

- This instruction applies a sampling trace stop command trigger. When the trigger is applied, the sampling trace is stopped after sampling of the specified delay count is performed.
- If the sampling trace settings and startup have not been specified by using the programming tool software, processing is not performed even if the internal relay condition is established.

■ Operation example

Operation of instruction format description program

When internal relay R10 turns ON, a sampling trace stop command trigger is applied.



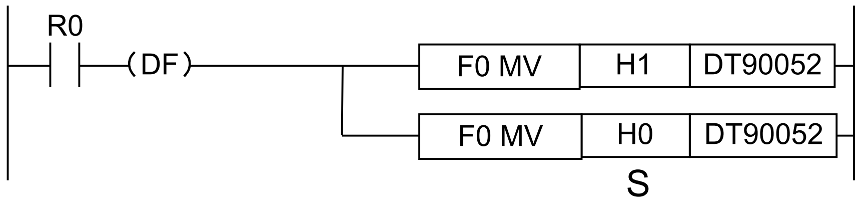
Registration of the data to be sampled, specification of the sampling method (such as the cable and the time interval), and specification of the command to start a sampling trace can only be performed by using the programming tool software.

23 High-speed Counter / PWM Output Instructions

23.1 F0 MV (High-speed Counter Control)

Performs the controls such as the software reset, disabling the count, and clearing the high-speed counter instruction.

■ Instruction format



■ Operands

Operand	Setting
S	Area storing the control code of the high-speed counter or constant data

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S	•	•	•	•	•	•	•	•	•	•	•	•

■ Outline of operation

- This instruction performs the high-speed counter control according to the control code specified by [S].
- This instruction is used when performing the following operations with the high-speed counter.
  1. When performing the software reset
  2. When disabling the count
  3. When disabling the reset input by an external input temporarily
  4. When canceling the control executed by the high-speed counter instruction F165 (CAM0)/F166 (HC1S)/F167 (HC1R) or when clearing the target value match interrupt
- The control codes once written are held until the next writing.
- The control code written by the F0 (MV) instruction is written to the special data register DT90052. At the same time, it is written to the control code monitor area. The written data is the data for lower 8 bits only.

■ Precautions for programming

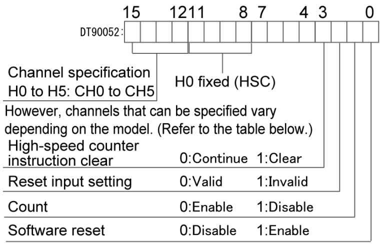
- The setting of disabling the rest input is valid only when allocating the reset input in the system register.
- In the external reset input setting, the reset input (X2 or X5) allocated to the control unit input is switched between valid and invalid.
- Disabling the hardware reset is valid only when the reset input is used.

■ High-speed counter/pulse output control flag area

- The figure below shows the allocation of the area DT90052 in which the relevant channel and control code will be written.
- The control code written by the F0 (MV) instruction is written to the special data register DT90052. At the same time, it is written to the control code monitor area.

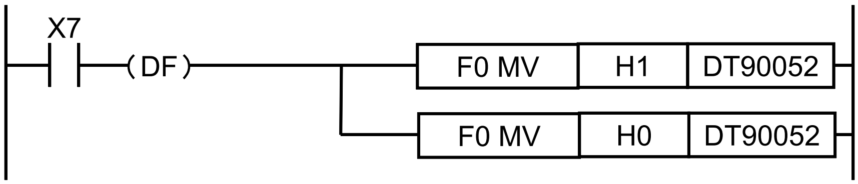
(Note) Only the lower 8 bits of data are written.

High-speed counter/pulse output control flag area



■ Program example

The following example shows the program for performing the software reset of the high-speed counter CH0 using the input X7.



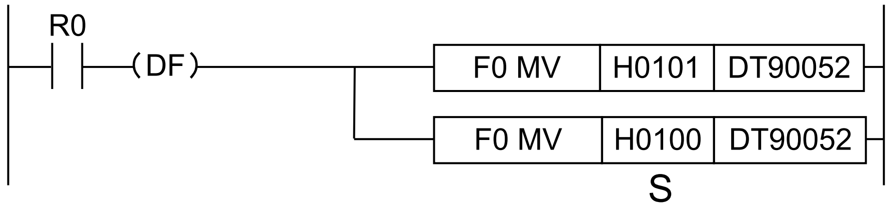
■ Table of correspondence between channel No. and elapsed value area

Channel No.	Control code monitor area
ch0	DT90370
ch1	DT90371
ch2	DT90372
ch3	DT90373
ch4	DT90374
ch5	DT90375

23.2 F0 MV (Pulse Output Control)

Performs the controls such as software reset, disabling the count, and stopping the pulse output.

■ Instruction format



■ Operands

Operand	Setting
S	Area storing the control code of the pulse output or constant data

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S	•	•	•	•	•	•	•	•	•	•	•	•

■ Outline of operation

- This instruction performs pulse output control according to the control code specified by [S].
- This instruction is used when performing the following operations with the pulse output.
  1. When performing the software reset
  2. When disabling the count
  3. When forcibly stopping the positioning/pulse output
  4. When canceling the control executed by pulse output related instructions F171 (SPDH)/ F172 (PLSH)/F173 (PWMH)/F174 (SP0H)/F175 (SPSH)
  5. When shifting to deceleration by setting the near home input during home return operation
- The control codes once written are held until the next writing.

■ Precautions for programming

- Disabling the count or performing the software reset during home return operation disables the near home processing.
- Although the near home bit is held, to perform near home processing during home return operation, it is necessary to write 1 to the target bit every time.

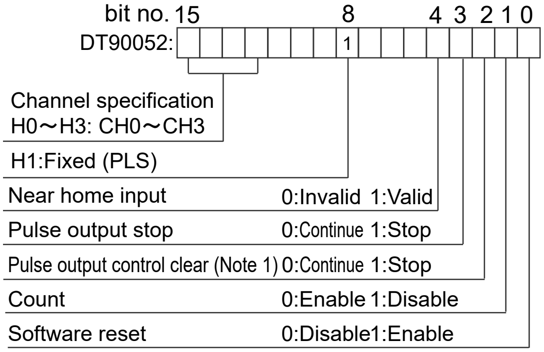
■ High-speed counter/pulse output control flag area

- The figure below shows the allocation of the area DT90052 in which the relevant channel and control code will be written.

- The control code written by the F0 (MV) instruction is written to the special data register DT90052. At the same time, it is written to the control code monitor area.
  1. Only the lower 8 bits of data are written.

Channel No.	Control code monitor area
ch0	DT90380
ch1	DT90381
ch2	DT90382
ch3	DT90383

High-speed counter/pulse output control flag area

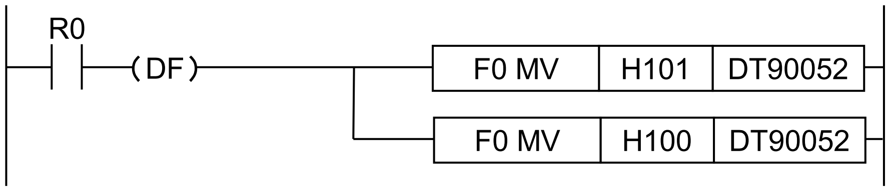


(Note 1) Pulse output control is valid when the F166 (HC1S) or F167 (HC1R) instruction is used to control a pulse output channel.

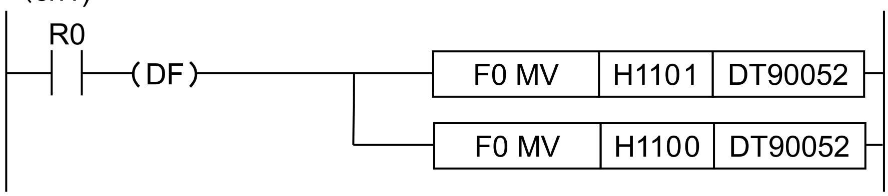
■ Program example

<Example 1> When performing the software reset of the pulse output

(ch0)

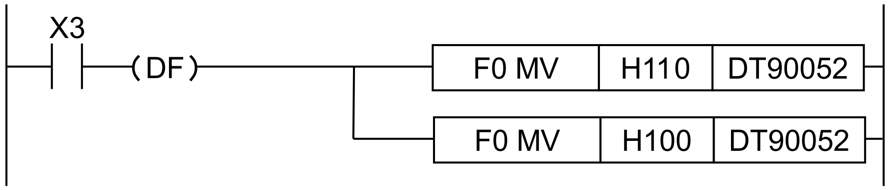


(ch1)

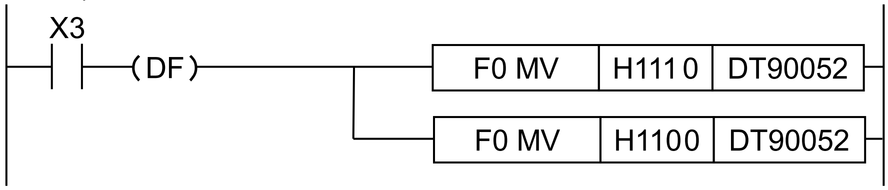


<Example 2> When setting the near home input during home return operation to shift to deceleration

(ch0)



(ch1)



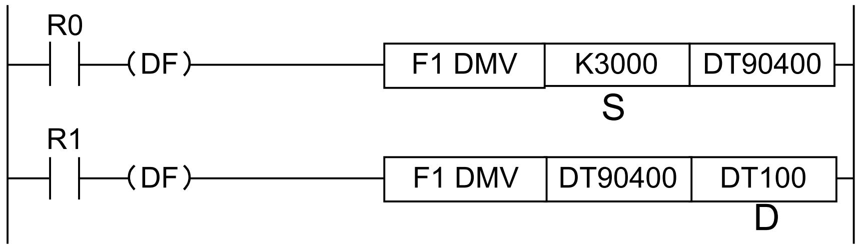
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when [S] is outside the specified range.

23.3 F1 DMV (High-speed Counter Elapsed Value Write/Read Instruction)

Writes and reads the elapsed value of the high-speed counter.

■ Instruction format



■ Operands

Operand	Setting
S	When setting: Area storing the elapsed value (32-bit) set in the high-speed counter or constant data K-2,147,483,648 to K2,147,483,647
D	When reading: Area reading the elapsed value of the high-speed counter

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S	●	●	●	●	●	●	●	●	●	●	●	●
D	-	●	●	●	●	●	●	●	●	-	-	●

■ Outline of operation (Reading elapsed value)

- Reads the content of the special data register storing the elapsed value of the high-speed counter and writes to the area specified by [D].

■ Outline of operation (Setting elapsed value)

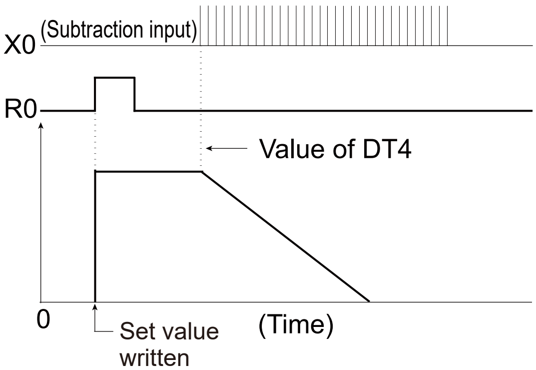
- At the same time as writing the value to the elapsed value area of the high-speed counter which uses 32-bit data specified by [S], sets it in the elapsed value area of the high-speed counter used within the system.

■ Precautions for programming

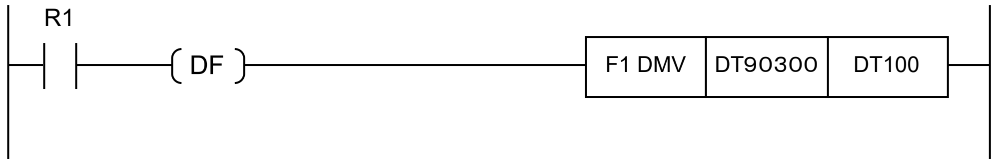
- Only F1 (DMV) instruction can perform the writing. The writing cannot be performed by other high-level instructions such as transfer instruction F0 (MV) and arithmetic instructions.
- Specify the memory area of [S] or [D] with the memory area number for the lower 16 bits.

■ Program example

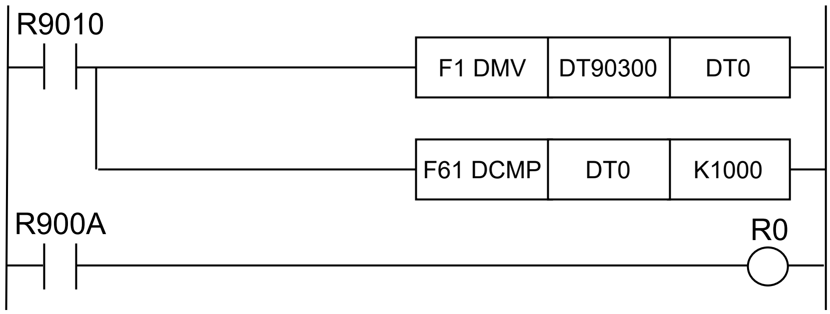
<Example 1> Set the value of the data register DT4 as the set value in the elapsed value area of CH0 with the input of R0.



<Example 2> Store the elapsed value of CH0 in the data register DT100 with the input of R1.



<Example 2> Store the elapsed value of CH0 in the data register DT100 with the input of R1.



■ Table of correspondence between channel No. and elapsed value area

High-speed counter channel No.	Pulse output channel No.	Elapsed value area
ch0	–	DT90300 to DT90301

High-speed counter channel No.	Pulse output channel No.	Elapsed value area
ch1	–	DT90304 to DT90305
ch2	–	DT90308 to DT90309
ch3	–	DT90312 to DT90313
ch4	–	DT90316 to DT90317
ch5	–	DT90320 to DT90323
–	ch0	DT90400 to DT90401
–	ch1	DT90410 to DT90411
–	ch2	DT90420 to DT90421
–	ch3	DT90430 to DT90431

23.4 F165 CAM0 (Cam Control)

Enables high-speed counter control based on the target values for up to 31 points.

■ Instruction format

■ Operands

Operand	Setting
S	Starting number of data table

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S	-	-	-	-	-	-	•	-	-	-	-	•

■ Outline of operation

This instruction notifies by internal relay that the target values for up to 31 points have been reached, using the pattern set in a data table that starts from the address specified by [S]. The internal relay corresponding to the target position turns ON.

It is also possible to start the interrupt program INTn at the target position.

* The execution of the interrupt program must be enabled by the ICTL instruction.

1. The instruction inspects whether the target values in the control table are not the same and are in ascending order.
2. It judges at which position the current value of the high-speed counter is located, sets 1 in the corresponding bit of the target position notification area of the internal relay, and clears all other bits to 0.
3. After that, for addition, it switches the internal relay for target position notification every time the target value is matched.

   However, for subtraction, "target position minus 1" is used as the target position data.

[When the upper limit control is disabled]

When 0 is specified for the upper limit and the reset input is disabled,

refer to the following explanation for upper limit control.

1. If the maximum target position m is matched during addition, the next target position is the negative minimum value.
2. If the minimum "target position 1 minus 1" is matched during subtraction, the target position is the positive maximum value.

In addition to the above control operation, this instruction provides an upper limit value control. The upper limit can be specified at the end of the table. When the elapsed value matches the upper limit, the elapsed value is cleared to 0 and the beginning of the internal relay in the position notification area turns ON.

For upper limit control, all target position data must be specified as positive integers.

[When the upper limit control is enabled]

- It is possible to return to the beginning of the table at the upper limit of the table, or by a hardware or soft reset signal. (V1.06 or later)
- For addition, when the upper limit is reached (when the reset signal is detected), the elapsed value is cleared to 0 and the 1st bit of the position notification relay turns ON.
- For subtraction, when -1 is reached, the upper limit is set to the elapsed value, and the bit corresponding to the target position m turns ON for the position output.

Info.

- Hardware reset: X2 for CH0 and CH1 and X5 for CH2 and CH3

Operational description of hardware reset signals

	When the upper limit control is disabled	When the upper limit control is disabled
V1.06 or later	The elapsed value is cleared to 0 with the table pointer returned to the beginning.	Only the elapsed value is cleared to 0.
V1.05 or earlier	Only the elapsed value is cleared to 0.	Only the elapsed value is cleared to 0.

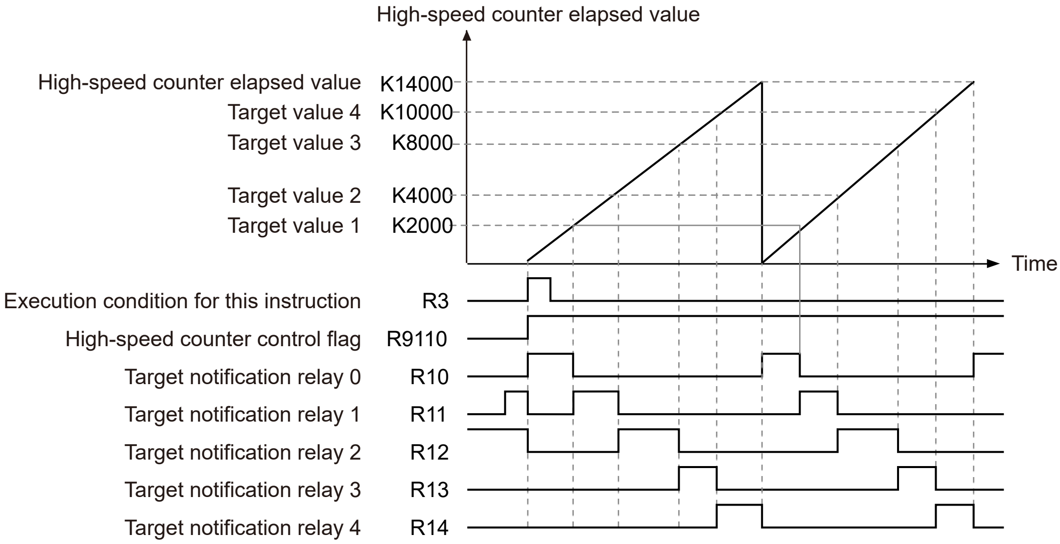
To use a hardware reset signal to return to the beginning of the table when the upper limit control is enabled, specify a large upper limit value that cannot be reached.

Operation image

When performing the upper limit control for the high-speed counter channel CH0

Condition: Example when this instruction is executed at the leading edge of execution condition R3 and the elapsed value at the start of the execution is less than the target position 1

For the target position notification area, relays starting from R10 are specified.



■ Precautions for programming

- This instruction cannot be used when the high-speed counter function is not used.

- The high-speed counter control active flag corresponding to the specified channel (R9110 to R9115) is ON until the cam control is canceled after the execution condition of the F165 (CAM0) instruction has turned ON.
- When the high-speed counter control active flag (R9110 to R9115) is on, the high-speed counter control instruction F166 (HC1S)/F167 (HC1R)/F178 (PLSM) for which the same channel is specified cannot be executed.
- To stop the control of this instruction, execute "Clear execution of high-speed counter instruction".
- If the elapsed value of the control target of this instruction is rewritten by an instruction, an unexpected operation may result.
- When controlling the control target using the main program, set each target value so that "minimum moving time between each target value" is larger than "1 scan time".
- When controlling the control target using an interrupt program, set each target value so that "minimum moving time between each target value" is larger than "maximum execution time of interrupt program".
- Set the target values so that adjacent target values are not matched within 1 ms.
- This instruction can be executed for only two channels at the same time.
- When the upper limit control and the hardware/software reset is used at the same time, do not operate them intensively in a short time.
- When hardware/soft reset is used, set the 1st target value to an integer value that is 1 or more.

■ High-speed counter channels and areas used

High-speed counter channel No.	Control active flag	Elapsed value area	Target value area	Interrupt program
ch0	R9110	DT90300 to DT90301	DT90302 to DT90303	INT0
ch1	R9111	DT90304 to DT90305	DT90306 to DT90307	INT1
ch2	R9112	DT90308 to DT90309	DT90310 to DT90311	INT3
ch3	R9113	DT90312 to DT90313	DT90314 to DT90315	INT4
ch4	R9114	DT90316 to DT90317	DT90318 to DT90319	INT6
ch5	R9115	DT90320 to DT90321	DT90322 to DT90323	INT7

■ Data table specification

S S+1	High-speed counter channel	(Note 1)
S+2 S+3	Word No. of internal relay for position notification	(Note 2)
S+4 S+5	No. of target positions m	(Note 3) Up to 31 target positions can be specified.
S+6 S+7	Target position 1	(Note 4)
S+8 S+9	Target position 2	(Note 4)
S+10 S+11	Target position 3	(Note 4)
	Target position m-1	(Note 4)
	Target position m	(Note 4)
	Upper limit	(Note 5)

(Note 1) High-speed counter channel

Specify the channel of the high-speed counter to be used as an H constant in the starting area (2 words) of the data table.

H 0 0 0 $\boxed{0}$ 0 0 0 $\boxed{1}$

<High-speed counter channel specification>
Settable range: 0 to 5

Upper limit control specification
0: Upper limit control disable
1: Upper limit control enable

<High-speed counter specification>
0: High-speed counter specification

(Note 2) Word No. of the internal relay that outputs the target position

(Note) Specify this so that the maximum area of the internal relay is not exceeded.

In the area of the 3rd and 4th words, specify the word number of the internal relay that outputs the target position.

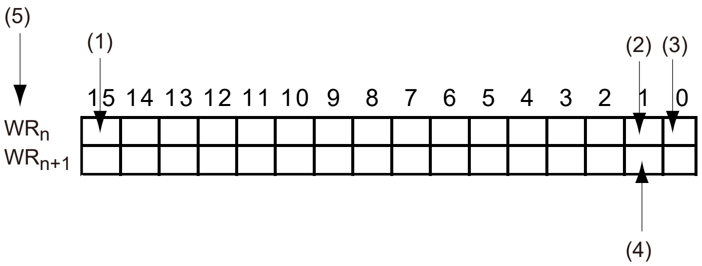
(Note 3) No. of target positions

(Note) Specify this so that the maximum area of the internal relay is not exceeded.

Specify the number of target positions.

Settable range: K1 to K31

<Target position notification method>



(1)	Turns ON when the 15th target value is matched.
(2)	Turns ON when the 1st target value is matched.
(3)	Turns ON when this instruction is executed. However, the current value must be less than target position 1.
(4)	Turns ON when the 17th target value is matched.
(5)	Value specified in S+2 and S+3

WRn:

If the number of target positions is 1 to 16, 1 word is used.

WRn+1:

If the number of target positions is 17 to 31, 2 words are used.

(Note 4) Target positions: Set the target positions in the 5th and subsequent words.

Settable range: K-2147483648 to K2147483647 (H80000000 to H7FFFFFF)

(Note 5) Upper limit: Specify the upper limit at the address following the target value of the final target position.

Settable range: K-2147483648 to K2147483647 (H80000000 to H7FFFFFF)

Note

- Specify the target values for target position 1 to target position m in ascending order. The same values cannot be specified.
- The interrupt program INTn corresponding to the specified high-speed counter channel can be executed at the target positions. Create an interrupt program corresponding to the channel to be controlled. After starting this instruction, execute the ICTL instruction to enable interrupts.

■ Setting example 1

[Conditions]

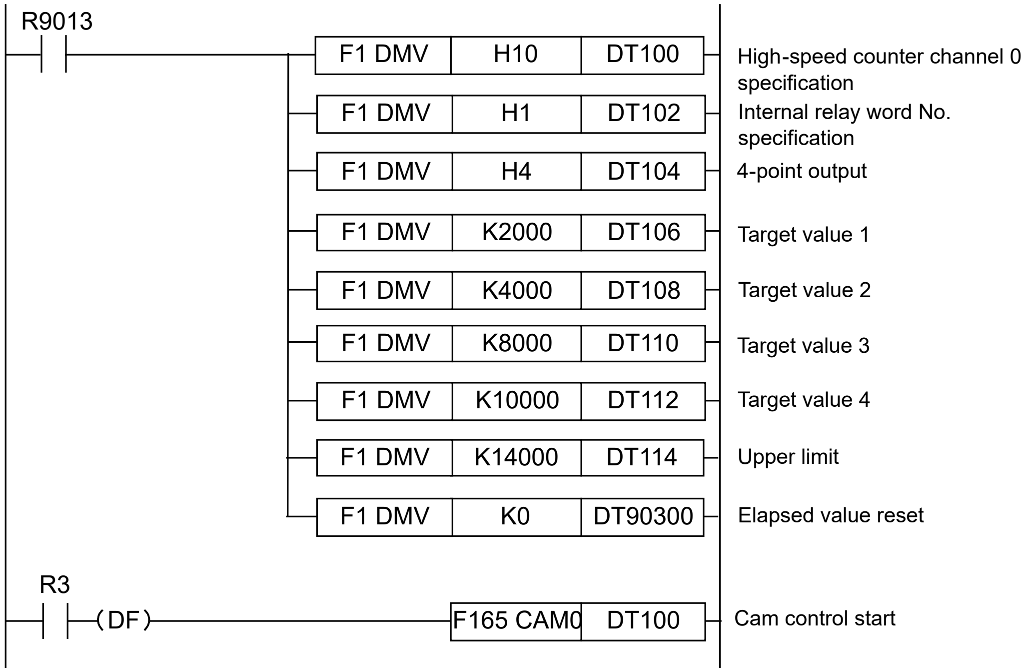
1. Target values are specified for 4 points. R10 generates position output.
2. Each target value is shown in the following table.

Position output	Target value
1(R11)	2000
2(R12)	4000
3(R13)	8000

Position output	Target value
4(R14)	10000

3. The upper limit is 14000 pulses.
4. Before position output starts, clear the elapsed value of the high-speed counter to 0.

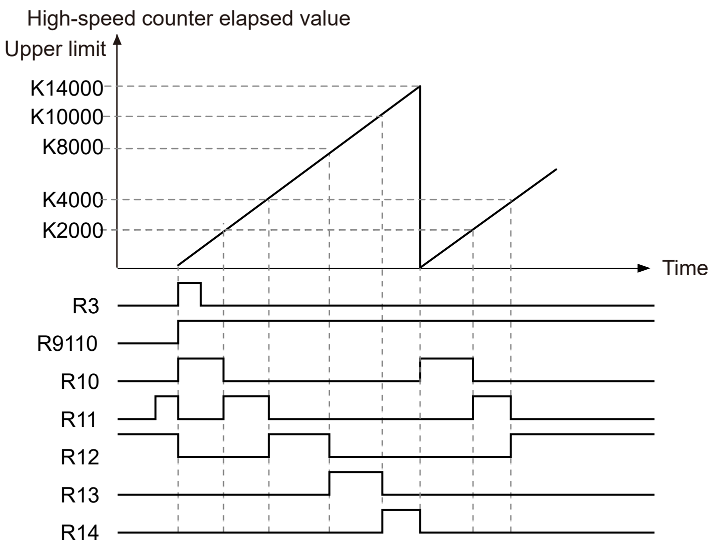
[Program]



[Description of program operation]

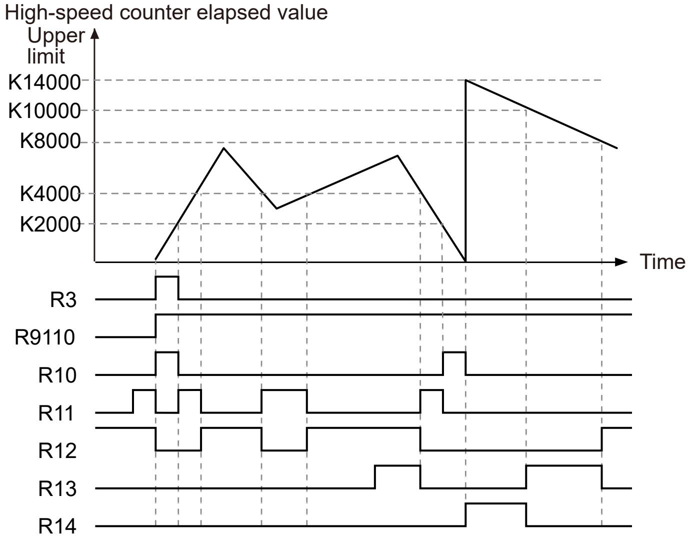
- For addition of the elapsed value with the upper limit

  When internal relay R3 is ON, the operation is as follows.



- For addition and subtraction of the elapsed value with the upper limit

When internal relay R3 is ON, the operation is as follows.



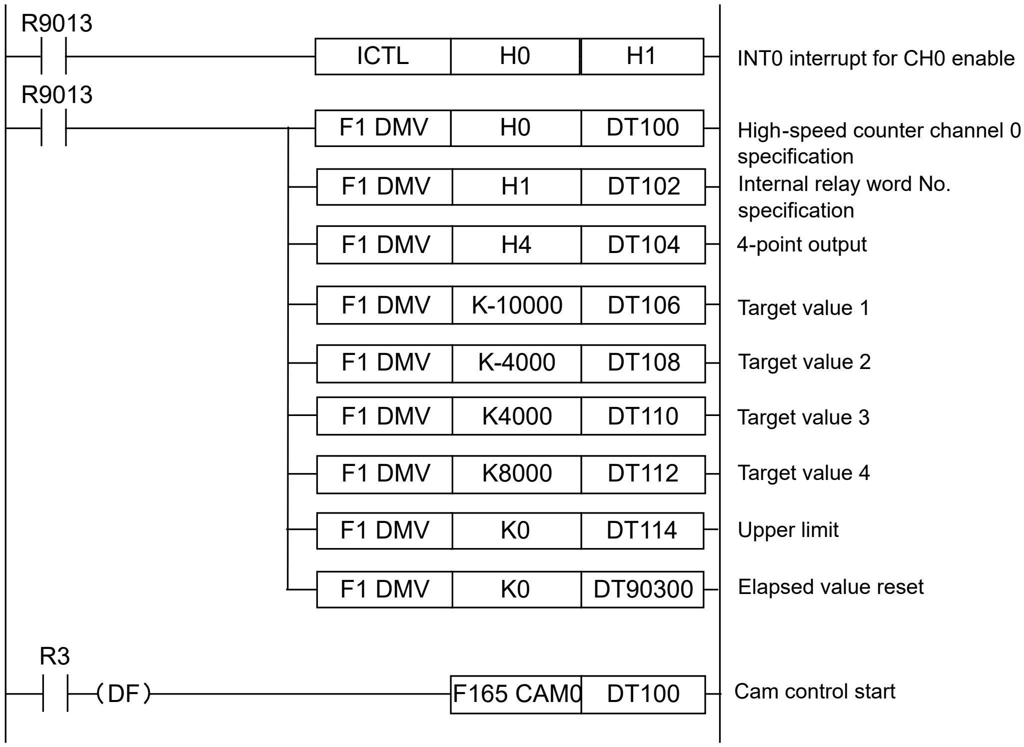
■ Setting example 2

[Conditions]

1. Target values for cam output is specified for 4 points. R10 to R13 generates output.
2. The target value for each cam is shown in the following table.

Position output	Target value
1(R11)	-10000
2(R12)	-4000
3(R13)	4000
4(R14)	8000

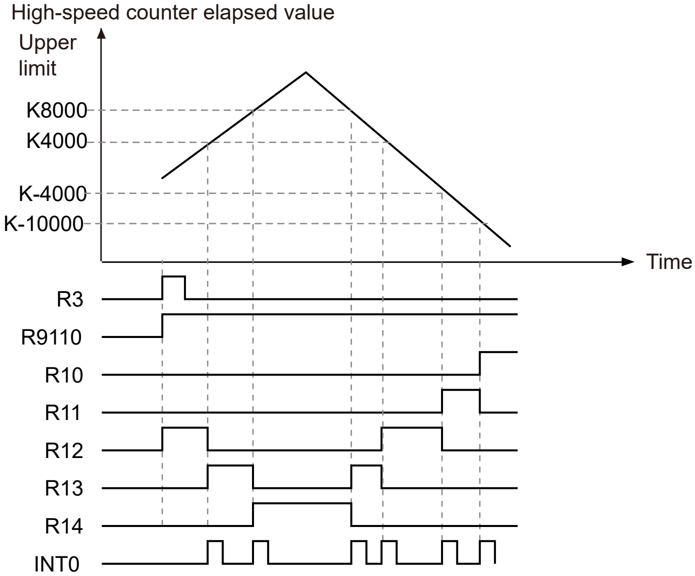
[Program]



[Description of program operation]

For addition and subtraction of the elapsed value without the upper limit and with interrupt control, if the elapsed value at the start of this instruction is greater than k-4000 and less than k4000, the operation when the internal relay R3 is ON is as follows.

* An interrupt program can be executed when it is enabled by the ICTL instruction.



(Note 1) The operation time in the interrupt program must be less than the moving time between control positions.

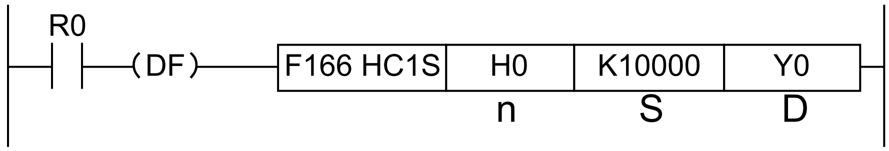
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the specified channel is outside the range.
	Turns ON when other than the high-speed counter is specified.
	Turns ON when the word number of the internal relay that outputs the target position is outside the range.
	Turns ON when the specified number of target values exceeds the limit (up to 31).
	Turns ON when the target value is greater than the upper limit.
	Turns ON when the target value is 0.
	Turns ON when the target values are not arranged in ascending order.
	Turns ON when the data table exceeds the area.
	Turns ON when the high-speed counter is specified, but the high-speed counter setting for the specified channel is not made in the system register.

23.5 F166 HC1S [Target Value Match ON (with Channel Specification)]

Turns ON the specified output when the elapsed value of a specified high-speed counter channel matches the target value.

■ Instruction format



■ Operands

Operand	Setting
n	Target channel number of the high-speed counter for match output (H0 to H5)
S	Target value data of the high-speed counter or the starting number of the area storing data
D	Output coil which turns ON when the values match (Yn)

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
n	-	-	-	-	-	-	-	-	-	●	●	-
S	●	●	●	●	●	●	●	●	●	●	●	●
D	-	-	-	-	-	-	-	-	-	-	-	-

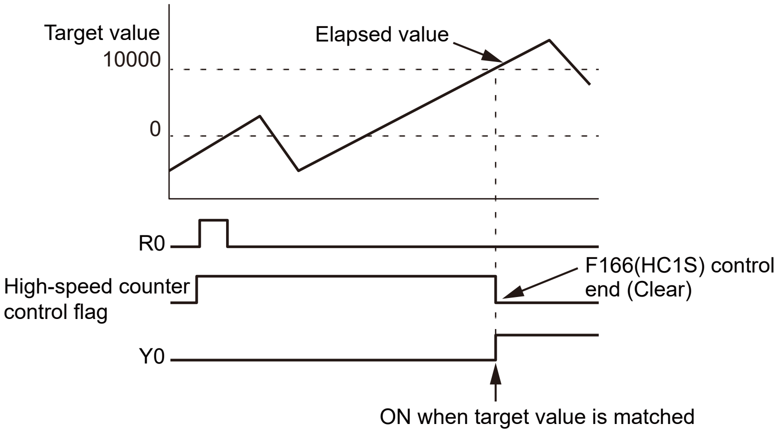
■ Outline of operation

- This instruction sets the value specified by [S] as the target value of the high-speed counter, and controls the specified output [Yn] when the elapsed value matches the target value. This operation is executed as interrupt processing.
- The setting of the target value and the control of the target value match output are canceled when the elapsed value matches the target value.
- The value of 32-bit data specified as the target value [S] should be within the range as shown below.

  K-2,147,483,648 to K2,147,483,647
- The value of [S] is stored in the target value area when the instruction is executed.
- Range of [Yn] that can be specified: Y0 to Y1F (devices that can be specified for match ON/OFF output)

  However, for devices that are not implemented, it turns ON/OFF output to memory only.

Program example



High-speed counter control active flag: R9110

The channel number of the high-speed counter control active flag changes depending on the channel being used. Refer to the table below for the channel number and the control active flag for each model.

Channel No.	Control active flag	Elapsed value area	Target value area
ch0	R9110	DT90300 to DT90301	DT90302 to DT90303
ch1	R9111	DT90304 to DT90305	DT90306 to DT90307
ch2	R9112	DT90308 to DT90309	DT90310 to DT90311
ch3	R9113	DT90312 to DT90313	DT90314 to DT90315
ch4	R9114	DT90316 to DT90317	DT90318 to DT90319
ch5	R9115	DT90320 to DT90321	DT90322 to DT90323

Precautions for programming

- Before using this instruction, make the high-speed counter setting in the system register.
- The high-speed counter control active flag is ON until the target value match output turns ON after the execution condition of the F166 (HC1S) instruction has turned ON. At this time, some instructions (F166 to F175) cannot be executed for the high-speed counter of the same channel.
- Even if the hardware reset is performed before the elapsed value matches the target value, the settings of the target value and the target value match output will not be cleared. (The elapsed value is cleared to 0.)
- For the output Y specified for the target value match output, it is not checked whether the output is overlapped with the OT, KP and other high-level instructions.
- To turn OFF the target value match output turned ON by this instruction, use the RST instruction or the F0 (MV) instruction to reset the output, or use it in pair with the F167 (HC1R) instruction.
- When describing the same channel in both the normal program and the interrupt program, be sure to program not to execute them simultaneously.
- An interrupt program can be executed when the elapsed value matches the target value.

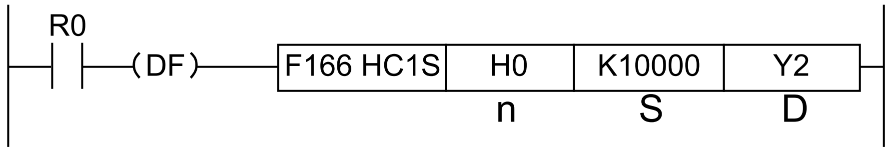
■ Flag operations

Name	Description
Hold error (R9007) Latest error (R9008)	Turns ON when the area is exceeded in index modification.
	Turns ON when [n] is outside the specified range.
	Turns ON when [S] is outside the specified range.
	Turns ON when [D] is outside the specified range.
	Turns ON when the high-speed counter setting for the specified channel is not made in the system register.

23.6 F166 HC1S [Target Value Match ON (High-speed Counter Control)]

Turns ON the specified output when the elapsed value of a specified high-speed counter (HSC) channel matches the target value.

■ Instruction format



■ Operands

Operand	Setting
n	Target channel number of the high-speed counter for the match output
S	Target value data of the high-speed counter or the starting number of the area storing data
D	Output coil which turns ON when the values match (Yn)

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
n	-	-	-	-	-	-	-	-	-	●	●	-
S	●	●	●	●	●	●	●	●	●	●	●	●
D	-	-	-	-	-	-	-	-	-	-	-	-

■ Outline of operation

- This instruction sets the value specified by [S] as the target value of the high-speed counter, and turns ON the specified output [Yn] when the elapsed value matches the target value. This operation is executed as interrupt processing.
- Specify the channel No. of the high-speed counter in [n].

  H 0 $\boxed{0}$ 0 $\boxed{0}$

  <High-speed counter/pulse output channel specification>
  Settable range for high-speed counter specification: 0 to 5

  <High-speed counter/pulse output specification>
  0: High-speed counter specification

- The setting of the target value and the control of this instruction are canceled when the elapsed value matches the target value.
- The value of 32-bit data specified as the target value [S] should be within the range as shown below.

  K-2,147,483,648 to K2,147,483,647

- The value of [S] is stored in the target value area when the instruction is executed.
- Range of [Yn] that can be specified: Y0 to Y1F (devices that can be specified for match ON/OFF output)
  However, for devices that are not implemented, it turns ON/OFF output to memory only.

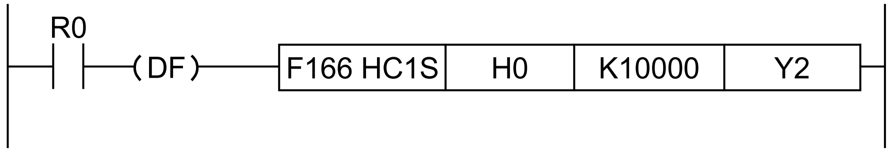
■ Setting example for target value match ON

<Specifying high-speed counter>

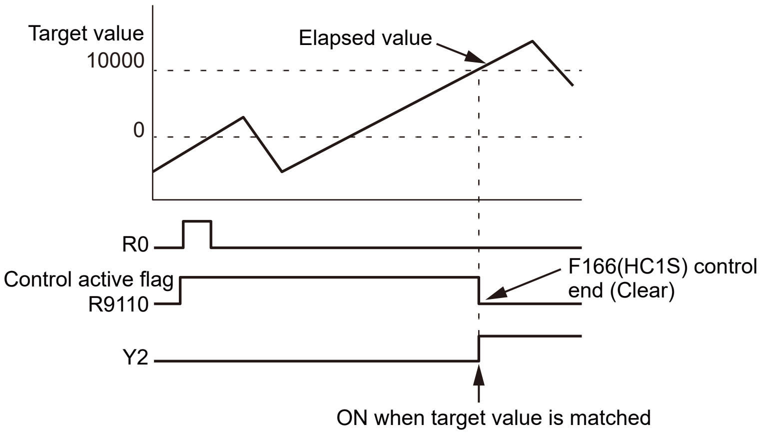
[Conditions]

1. Channel No. 0 is specified for the high-speed counter channel.
2. The target value is 10000.
3. Output coil Y2 turns ON when the values match.

[Program]



[Executing the program]



The channel number of the high-speed counter control active flag changes depending on the channel being used. Refer to the table below for the channel number and the control active flag for each model.

Channel No.	Control active flag	Elapsed value area	Target value area	Interrupt program
ch0	R9110	DT90300 to DT90301	DT90302 to DT90303	INT0
ch1	R9111	DT90304 to DT90305	DT90306 to DT90307	INT1
ch2	R9112	DT90308 to DT90309	DT90310 to DT90311	INT3
ch3	R9113	DT90312 to DT90313	DT90314 to DT90315	INT4
ch4	R9114	DT90316 to DT90317	DT90318 to DT90319	INT6
ch5	R9115	DT90320 to DT90321	DT90322 to DT90323	INT7

Precautions for programming

- This instruction cannot be executed when the specified high-speed counter channel is not set in the system register. An operation error occurs.
- The high-speed counter control active flag (R9110 to R9115) is ON until the elapsed value matches the target value after the execution condition of the instruction has turned ON. At this time, the high-speed counter control instructions (F165 (CAM0), F166 (HC1S), F167 (HC1R), and F178 (PLSM)) cannot be executed for the high-speed counter of the same channel.
- Even if the hardware reset is performed before the elapsed value matches the target value, the settings of the target value and the target value match output will not be cleared. (The elapsed value is cleared to 0.)
- For the output Y specified for the target value match output, it is not checked whether the output is overlapped with the OT, KP and other high-level instructions.
- If the control is canceled by the F0 (MV) S, DT90052 instruction, the control of this instruction is canceled and the high-speed counter control active flag turns OFF.
- When describing the same channel in both the normal program and the interrupt program, be sure to program not to execute them simultaneously.
- An interrupt program can be executed when the elapsed value matches the target value. This requires writing the interrupt program and enabling it with the ICTL instruction.

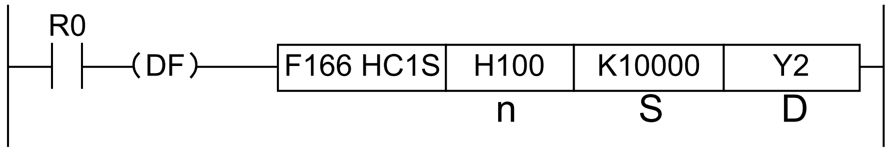
Flag operations

Name	Description
Hold error (R9007) Latest error (R9008)	Turns ON when the area is exceeded in index modification.
	Turns ON when [n] is outside the specified range.
	Turns ON when [S] is outside the specified range.
	Turns ON when [D] is outside the specified range.
	Turns ON when the high-speed counter setting for the specified channel is not made in the system register.

23.7 F166 HC1S [Target Value Match ON (Pulse Output Control)]

Turns ON the specified output when the elapsed value of a specified pulse output channel matches the target value.

■ Instruction format



■ Operands

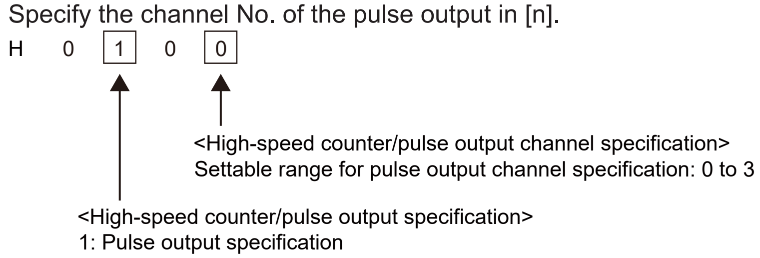
Operand	Setting
n	Target channel number of the pulse output for match output
S	Target value data of the pulse output or the starting number of the area storing data
D	Output coil which turns ON when the values match (Yn)

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
n	-	-	-	-	-	-	-	-	-	●	●	-
S	●	●	●	●	●	●	●	●	●	●	●	●
D	-	-	-	-	-	-	-	-	-	-	-	-

■ Outline of operation

- This instruction sets the value specified by [S] as the target value of the pulse output for match ON, and turns ON the specified output [Yn] when the elapsed value matches the target value. This operation is executed as interrupt processing.
- Specify the channel No. of the pulse output in [n].



- The setting of the target value and the control of this instruction are canceled when the elapsed value matches the target value. The control active flag also turns OFF.
- The value of 32-bit data specified as the target value [S] should be within the range as shown below.

  K-2,147,483,648 to K2,147,483,647

- The value of [S] is stored in the target value area when the instruction is executed.
- Range of [Yn] that can be specified: Y0 to Y1F (devices that can be specified for match ON/OFF output)
  However, for devices that are not implemented, it turns ON/OFF output to memory only.

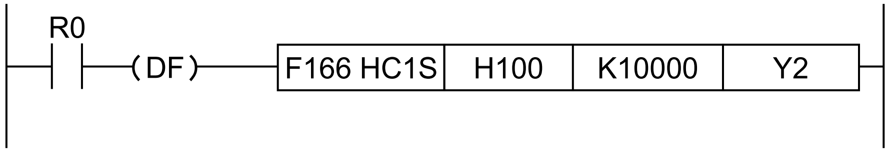
■ Setting example for target value match ON

<Specifying pulse output>

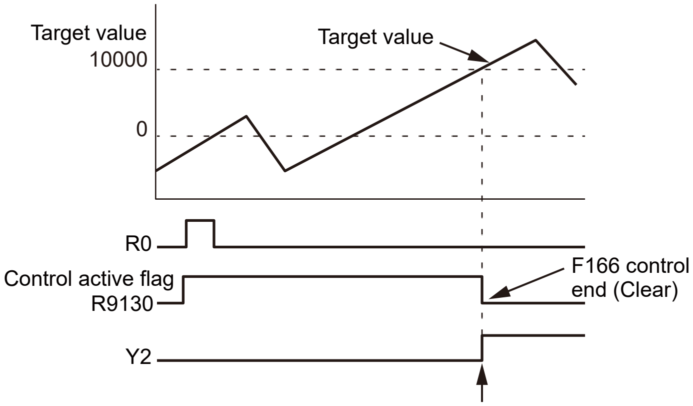
[Conditions]

1. Specify channel No. 0 for the pulse output channel.
2. The target value is 10000.
3. Output coil Y2 turns ON when the values match.

[Program]



■ Program example



The channel number of the high-speed counter control active flag changes depending on the channel being used. Refer to the table below for the channel number and the control active flag for each model.

Channel No.	For pulse output		
	Control active flag	Elapsed value area	Target value area
ch0	R9120	DT90400 to DT90401	DT90402 to DT90403
ch1	R9121	DT90410 to DT90411	DT90412 to DT90413
ch2	R9122	DT90420 to DT90421	DT90422 to DT90423
ch3	R9123	DT90430 to DT90431	DT90432 to DT90433

Channel No.	For pulse output control		
	Control active flag	Target value area	Interrupt program
ch0	R9130	DT90404 to DT90405	INT8
ch1	R9131	DT90414 to DT90415	INT9
ch2	R9132	DT90424 to DT90425	INT10
ch3	R9133	DT90434 to DT90435	INT11

■ Precautions for programming

- This instruction cannot be executed when the specified pulse output channel is not set in the system register. An operation error occurs.
- The pulse control active flag (R9130 to R9133) is ON until the elapsed value matches the target value after the execution condition of the instruction has turned ON. At this time, the pulse output control instructions [(F165 (CAM0), F166 (HC1S), and F167 (HC1R) cannot be executed for the pulse output of the same channel.
- This instruction is valid for all pulse output instructions except the F173 (PWMH) instruction. This instruction can be executed both before and after execution of the above pulse output instructions.
- For the output Y specified for the target value match output, it is not checked whether the output is overlapped with the OT, KP and other high-level instructions.
- If the control is canceled by the F0 (MV) S, DT90052 instruction, the control of this instruction is canceled and also the active flag turns OFF, but the pulse output continues.
- When describing the same channel in both the normal program and the interrupt program, be sure to program not to execute them simultaneously.
- An interrupt program can be executed when the elapsed value matches the target value. This requires writing the interrupt program and enabling it with the ICTL instruction.

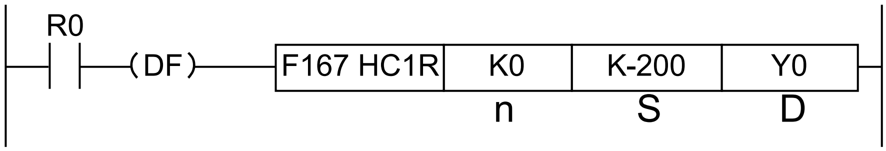
■ Flag operations

名称	Description
R9007 R9008 （ER）	Turns ON when the area is exceeded in index modification.
	Turns ON when [n] is outside the specified range.
	Turns ON when [S] is outside the specified range.
	Turns ON when [D] is outside the specified range.
	Turns ON when the pulse output is specified, but the pulse output for the specified channel is not set in the system register.

23.8 F167 HC1R [Target Value Match OFF (with Channel Specification)]

Turns OFF the specified output when the elapsed value of a specified high-speed counter channel matches the target value.

■ Instruction format



■ Operands

Operand	Setting
n	Target channel number of the high-speed counter for the match output
S	Target value data of the high-speed counter or the starting number of the area storing data
D	Output coil which turns OFF when the values match (Yn)

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
n	-	-	-	-	-	-	-	-	-	●	●	-
S	●	●	●	●	●	●	●	●	●	●	●	●
D	-	-	-	-	-	-	-	-	-	-	-	-

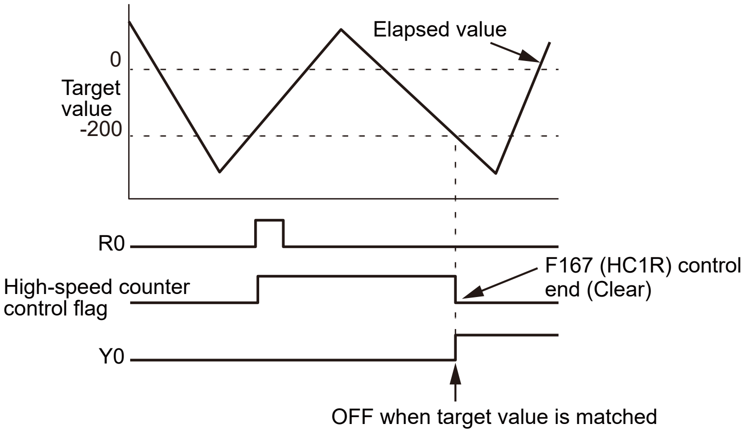
■ Outline of operation

- This instruction sets the value specified by [S] as the target value of the high-speed counter, and turns OFF the specified output [Yn] when the elapsed value matches the target value. This operation is executed as interrupt processing.
- The setting of the target value and the control of the target value match output OFF are canceled when the elapsed value matches the target value.
- The value of 32-bit data specified as the target value [S] should be within the range as shown below.

  K-2,147,483,648 to K2,147,483,647
- The value of [S] is stored in the target value area when the instruction is executed.
- Range of [Yn] that can be specified: Y0 to Y1F (devices that can be specified for match ON/OFF output)

  However, for devices that are not implemented, it turns ON/OFF output to memory only.

Program example



High-speed counter control active flag: R9110

The channel number of the high-speed counter control active flag changes depending on the channel being used. Refer to the table below for the channel number and the control active flag for each model.

Channel No.	Control active flag	Elapsed value area	Target value area
ch0	R9110	DT90300 to DT90301	DT90302 to DT90303
ch1	R9111	DT90304 to DT90305	DT90306 to DT90307
ch2	R9112	DT90308 to DT90309	DT90310 to DT90311
ch3	R9113	DT90312 to DT90313	DT90314 to DT90315
ch4	R9114	DT90316 to DT90317	DT90318 to DT90319
ch5	R9115	DT90320 to DT90321	DT90322 to DT90323

Precautions for programming

- Before using this instruction, make the high-speed counter setting in the system register.
- The high-speed counter control active flag is ON until the target value match output turns OFF after the execution condition of the F167 (HC1R) instruction has turned ON. At this time, some instructions (F166 to F173) cannot be executed for the high-speed counter of the same channel.
- Even if the hardware reset is performed before the elapsed value matches the target value, the settings of the target value and the target value match output will not be cleared. (The elapsed value is cleared to 0.)
- For the output Y specified for the target value match output, it is not checked whether the output is overlapped with the OT, KP and other high-level instructions.
- To turn ON the target value match output turned OFF by this instruction, use the SET instruction or the F0 (MV) instruction to reset the output, or use it in pair with the F166 (HC1S) instruction.
- When describing the same channel in both the normal program and the interrupt program, be sure to program not to execute them simultaneously.
- An interrupt program can be executed when the elapsed value matches the target value.

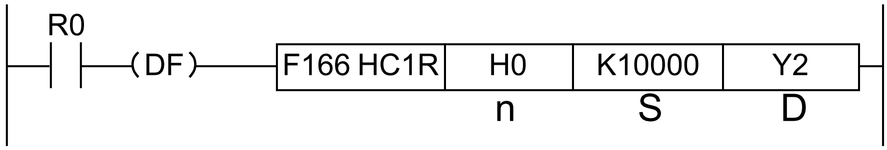
■ Flag operations

Name	Description
Hold error (R9007) Latest error (R9008)	Turns ON when the area is exceeded in index modification.
	Turns ON when [n] is outside the specified range.
	Turns ON when [S] is outside the specified range.
	Turns ON when [D] is outside the specified range.
	Turns ON when the high-speed counter setting for the specified channel is not made in the system register.

23.9 F167 HC1R [Target Value Match OFF (High-speed Counter Control)]

Turns OFF the specified output when the elapsed value of a specified high-speed counter (HSC) channel matches the target value.

■ Instruction format



■ Operands

Operand	Setting
n	Target channel number of the high-speed counter for the match output
S	Target value data of the high-speed counter or the starting number of the area storing data
D	Output coil which turns ON when the values match (Yn)

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
n	-	-	-	-	-	-	-	-	-	●	●	-
S	●	●	●	●	●	●	●	●	●	●	●	●
D	-	-	-	-	-	-	-	-	-	-	-	-

■ Outline of operation

- This instruction sets the value specified by [S] as the target value of the high-speed counter, and turns ON the specified output [Yn] when the elapsed value matches the target value. This operation is executed as interrupt processing.
- Specify the channel No. of the high-speed counter in [n].

H 0 $\boxed{0}$ 0 $\boxed{0}$

<High-speed counter/pulse output channel specification>
Settable range for high-speed counter specification: 0 to 5

<High-speed counter/pulse output specification>
0: High-speed counter specification

- The setting of the target value and the control of this instruction are canceled when the elapsed value matches the target value.
- The value of 32-bit data specified as the target value [S] should be within the range as shown below.

  K-2,147,483,648 to K2,147,483,647

- The value of [S] is stored in the target value area when the instruction is executed.
- Range of [Yn] that can be specified: Y0 to Y1F (devices that can be specified for match ON/OFF output)
  However, for devices that are not implemented, it turns ON/OFF output to memory only.

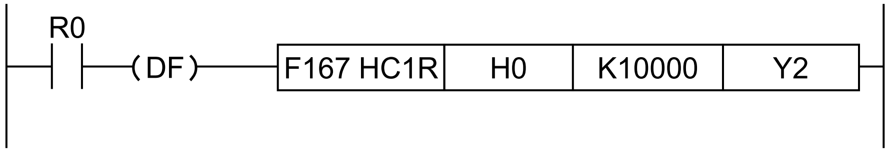
■ Setting example for target value match OFF

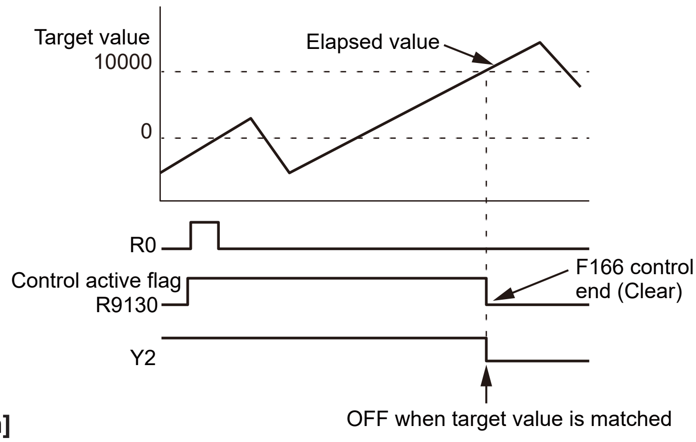
<Specifying high-speed counter>

[Conditions]

1. Channel No. 0 is specified for the high-speed counter channel.
2. The target value is 10000.
3. Output coil Y2 turns OFF when the values match.

[Program]





[Executing the program]

The channel number of the high-speed counter control active flag changes depending on the channel being used. Refer to the table below for the channel number and the control active flag for each model.

Channel No.	Control active flag	Elapsed value area	Target value area	Interrupt program
ch0	R9110	DT90300 to DT90301	DT90302 to DT90303	INT0
ch1	R9111	DT90304 to DT90305	DT90306 to DT90307	INT1
ch2	R9112	DT90308 to DT90309	DT90310 to DT90311	INT3
ch3	R9113	DT90312 to DT90313	DT90314 to DT90315	INT4
ch4	R9114	DT90316 to DT90317	DT90318 to DT90319	INT6
ch5	R9115	DT90320 to DT90321	DT90322 to DT90323	INT7

Precautions for programming

- This instruction cannot be executed when the specified high-speed counter channel is not set in the system register. An operation error occurs.
- The high-speed counter control active flag (R9110 to R9115) is ON until the elapsed value matches the target value after the execution condition of the instruction has turned ON. At this time, the high-speed counter control instructions [F165 (CAM0), F166 (HC1S), F167 (HC1R), and F178 (PLSM)] cannot be executed for the high-speed counter of the same channel.
- Even if the hardware reset is performed before the elapsed value matches the target value, the target value and the control of this instruction will not be canceled. (The elapsed value is cleared to 0.)
- For the output Y specified for the target value match output, it is not checked whether the output is overlapped with the OT, KP and other high-level instructions.
- If the control is canceled by the F0 (MV) S, DT90052 instruction, the control of this instruction is canceled and the high-speed counter control active flag turns OFF.
- When describing the same channel in both the normal program and the interrupt program, be sure to program not to execute them simultaneously.
- An interrupt program can be executed when the elapsed value matches the target value. This requires writing the interrupt program and enabling it with the ICTL instruction.

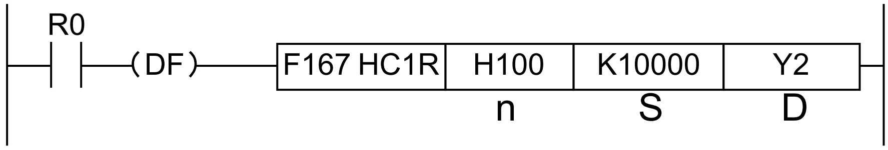
Flag operations

Name	Description
Hold error (R9007) Latest error (R9008)	Turns ON when the area is exceeded in index modification.
	Turns ON when [n] is outside the specified range.
	Turns ON when [S] is outside the specified range.
	Turns ON when [D] is outside the specified range.
	Turns ON when the high-speed counter setting for the specified channel is not made in the system register.

23.10 F167 HC1R [Target Value Match OFF (Pulse Output Control)]

Turns OFF the specified output when the elapsed value of a specified pulse output channel matches the target value.

■ Instruction format



■ Operands

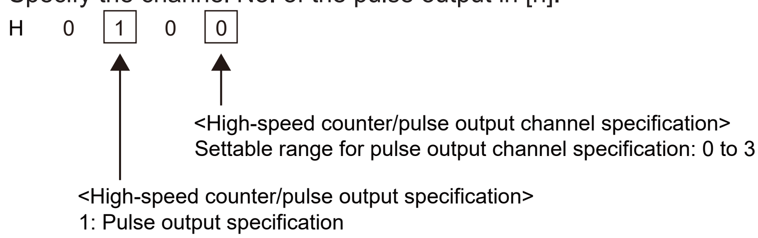
Operand	Setting
n	Target channel number of the pulse output for match output
S	Target value data of the pulse output or the starting number of the area storing data
D	Output coil which turns ON when the values match (Yn)

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
n	-	-	-	-	-	-	-	-	-	●	●	-
S	●	●	●	●	●	●	●	●	●	●	●	●
D	-	-	-	-	-	-	-	-	-	-	-	-

■ Outline of operation

- This instruction sets the value specified by [S] as the target value of the pulse output for match ON, and turns ON the specified output [Yn] when the elapsed value matches the target value. This operation is executed as interrupt processing.
- Specify the channel No. of the pulse output in [n].

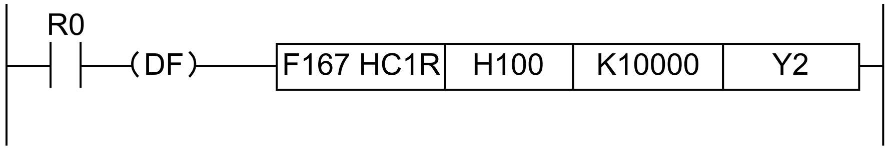


- The setting of the target value and the control of this instruction are canceled when the elapsed value matches the target value. The control active flag also turns OFF.
- The value of 32-bit data specified as the target value [S] should be within the range as shown below.

  K-2,147,483,648 to K2,147,483,647
- The value of [S] is stored in the target value area when the instruction is executed.

- Range of [Yn] that can be specified: Y0 to Y1F (devices that can be specified for match ON/OFF output)
  However, for devices that are not implemented, it turns ON/OFF output to memory only.

■ Setting example for target value match OFF

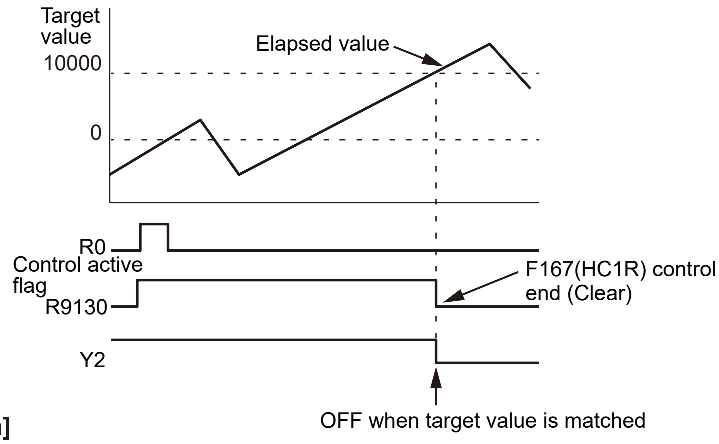


<Specifying pulse output>

[Conditions]

1. Specify channel No. 0 for the pulse output channel.
2. The target value is 10000.
3. Output coil Y2 turns OFF when the values match.

[Program]



[Executing the program]

The channel number of the high-speed counter control active flag changes depending on the channel being used. Refer to the table below for the channel number and the control active flag for each model.

Channel No.	For pulse output		
	Control active flag	**Elapsed value area**	**Target value area**
ch0	R9120	DT90400 to DT90401	DT90402 to DT90403
ch1	R9121	DT90410 to DT90411	DT90412 to DT90413
ch2	R9122	DT90420 to DT90421	DT90422 to DT90423
ch3	R9123	DT90430 to DT90431	DT90432 to DT90433

Channel No.	For pulse output control		
	Control active flag	Elapsed value area	Interrupt program
ch0	R9130	DT90404 to DT90405	INT8
ch1	R9131	DT90414 to DT90415	INT9
ch2	R9132	DT90424 to DT90425	INT10
ch3	R9133	DT90434 to DT90435	INT11

■ Precautions for programming

- This instruction cannot be executed when the specified pulse output channel is not set in the system register. An operation error occurs.
- The pulse control active flag (R9110 to R9115) is ON until the elapsed value matches the target value after the execution condition of the instruction has turned ON. At this time, the pulse output control instructions [(F165 (CAM0), F166 (HC1S), and F167 (HC1R) cannot be executed for the pulse output of the same channel.
- This instruction is valid for all pulse output instructions except the F173 (PWMH) instruction. This instruction can be executed both before and after execution of the above pulse output instructions.
- For the output Y specified for the target value match output, it is not checked whether the output is overlapped with the OT, KP and other high-level instructions.
- If the control is canceled by the F0 (MV) S, DT90052 instruction, the control of this instruction is canceled and also the active flag turns OFF, but the pulse output continues.
- When describing the same channel in both the normal program and the interrupt program, be sure to program not to execute them simultaneously.
- An interrupt program can be executed when the elapsed value matches the target value. This requires writing the interrupt program and enabling it with the ICTL instruction.

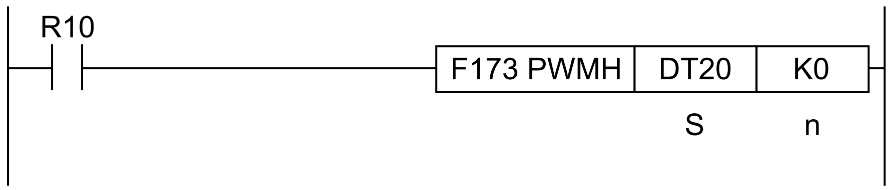
■ Flag operations

Name	Description
Hold error (R9007) Latest error (R9008)	Turns ON when the area is exceeded in index modification.
	Turns ON when [n] is outside the specified range.
	Turns ON when [S] is outside the specified range.
	Turns ON when [D] is outside the specified range.
	Turns ON when the pulse output is specified, but the pulse output for the specified channel is not set in the system register.

23.11 F173 PWMH [PWM Output (with Channel Specification)]

Outputs PWM pulses from a specified PWM output channel according to specified parameters.

■ Instruction format



■ Operands

Operand	Setting
S	Starting number of the area in which the data table is registered
n	Target output channel for PWM output

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S	-	-	-	-	-	-	●	-	-	-	-	●
n	-	-	-	-	-	-	-	-	-	●	●	-

■ Outline of operation

- This instruction performs PWM output from the specified PWM output channel when the corresponding control active flag is OFF and the execution condition is ON. The output continues while the execution condition is ON.
- Frequency and duty are specified in the data table below, which is created by the user program
- The duty may be different from the set ratio depending on the load voltage and load current especially in the vicinity of minimum and maximum values.
- The duty can be changed for each scan. However, the control code cannot be changed during the execution of the instruction.

■ List of used areas

Channel No.	Output	Pulse input/output instruction executing flag
ch0	Y0	R9120
ch1	Y2	R9121
ch2	Y4	R9122
ch3	Y6	R9123

■ Precautions for programming

- This instruction cannot be executed when the control active flag corresponding to each channel is ON or when the circular interpolation control active flag (R904E) is ON.
- When describing the same channel in both the normal program and the interrupt program, be sure to program not to execute them simultaneously.
- This requires setting the PWM output in the system register.
- When rewriting during RUN is performed during the operation, the pulse output stops while a program is being rewritten.
- If the control code is changed after this instruction is started, the change affects only the duty resolution with no effect on the frequency.
- If the control code is changed to a value outside of the settable range after startup, the instruction operates with a duty resolution of 100 without causing an operation error.
- If the duty setting is changed to 100% or more after startup, this instruction operates at the maximum value of the specified resolution without causing an operation error.

■ Data table settings

S	Control code	(Note 1)
S+1	Duty	(Note 2)

(Note 1) Specify the control code with a k constant.

(Note 2) Specify the duty with a K constant.

When the control code is set to K0 to K19, the duty is K0 to K999 (0.0% to 99.9%).

When the control code is set to K20 to K24, the duty is K0 to K99 (0% to 99%). The value is set in units of 1% (K10) (rounded down to the nearest whole number).

Setting	Frequency (Hz)	Cycle (ms)
K3	6	166.67
K4	7.5	133.33
K5	12.5	80.00
K6	25	40.00
K7	50	20.00
K8	100	10.00
K9	200	5.00
K10	400	2.50
K11	600	1.67
K12	800	1.25
K13	1000	1.00
K14	1200	0.83
K15	1600	0.63
K16	2000	0.50

Setting	Frequency (Hz)	Cycle (ms)
K17	3000	0.33
K18	4800	0.21
Other than above	Cannot be specified	

■ Flag operations

Name	Description
Hold error (R9007) Latest error (R9008)	Turns ON when the area is exceeded in index modification.
	Turns ON when [n] is other than 0 or 2.
	Turns ON when the control code is outside the settable range (at startup of the instruction).
	Turns ON when the duty is 100% or more (at startup of the instruction).
	Turns ON when the PWM output is not set in the system register.

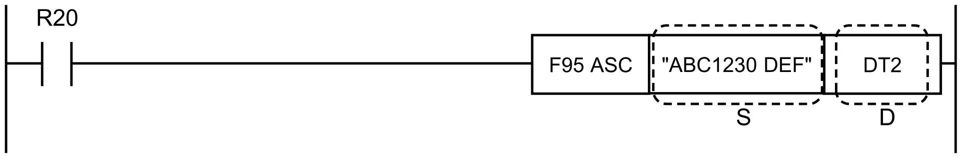
(MEMO)

24 Character String Instructions

24.1 F95 ASC (Character Constant to ASCII Code Conversion)

Converts the specified character constants into ASCII codes.

■ Instruction format



■ Operands

Items	Settings
S	Character constants (12 characters)
D	Number at the start of the area storing the ACSII codes

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S														•			
D		•	•	•	•	•	•	•	•								

■ Outline of operation

The character constants specified by [S] (12 characters) are converted into ASCII codes and stored in the 6-word area starting from [D].

■ Operation example

Operation of instruction format description program

When internal relay R20 turns ON, the specified character constants (ABC1230 DEF) are converted into ASCII codes and stored in DT2 to DT7.

[S]
"ABC1230 DEF"

R20:ON

	H	L
DT2	H 42 (B)	H 41 (A)
DT3	H 31 (1)	H 43 (C)
DT4	H 33 (3)	H 32 (2)
DT5	H 20 (SP)	H 30 (0)
DT6	H 45 (E)	H 44 (D)
DT7	H 20 (SP)	H 46 (F)

If the number of character constants specified by [S] is less than 12, the blanks in the destination storage area are filled with spaces (H20).

■ Precautions for programming

The character constant M can only be input by programming tool software.

Conversion example

When converting one letter (A), there are three possible input methods.

1. At the start of the specified character constants (1st character)
2. At the end of the specified character constants (12th character)
3. In the middle of the specified character constants (2nd to 11th character)

(1) At the start (1st character)

[S]

"A   "

11 spaces (whitespace)

F95 ASC	"A"	DT2

R20:ON

	H	L
DT2	H 20 (SP)	H 41 (A)
DT3	H 20 (SP)	H 20 (SP)
DT4	H 20 (SP)	H 20 (SP)
DT5	H 20 (SP)	H 20 (SP)
DT6	H 20 (SP)	H 20 (SP)
DT7	H 20 (SP)	H 20 (SP)

The letter is input as above. A is only input to the low byte of DT2. The blanks are all filled with spaces (H20) in the destination storage area.

(2) At the end (12th character)

[S]

"   A"

11 spaces (whitespace)

F95 ASC	" A"	DT2

R20:ON

	H	L
DT2	H 20 (SP)	H 20 (SP)
DT3	H 20 (SP)	H 20 (SP)
DT4	H 20 (SP)	H 20 (SP)
DT5	H 20 (SP)	H 20 (SP)
DT6	H 20 (SP)	H 20 (SP)
DT7	H 41 (A)	H 20 (SP)

The letter is input as above. A is only input to the high byte of DT7. DT2 to DT6 and the low byte of D27 are all filled with spaces (H20) in the destination storage area.

(3) In the middle (7th character)

[S]

"   A   "

String of A surrounded by 6 and 5 spaces (whitespace)

F95 ASC	" A"	DT2

R20:ON

	H	L
DT2	H 20 (SP)	H 20 (SP)
DT3	H 20 (SP)	H 20 (SP)
DT4	H 20 (SP)	H 20 (SP)
DT5	H 20 (SP)	H 41 (A)
DT6	H 20 (SP)	H 20 (SP)
DT7	H 20 (SP)	H 20 (SP)

The letter is input as above. A is only input to the low byte of DT5. The rest of the destination storage area is filled with spaces (H20).

■ Reference: JIS8 code table

				b_7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
				b_6	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				b_5	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				b_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b_3	b_2	b_1	b_0	Column / Row	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	ー	タ	ミ		
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	Undefined	Undefined	・	オ	ナ	ユ	Undefined	Undefined
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	TC$_{10}$(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	゜		

(Note 1) Only the character constants in the range indicated by ☐ in the table above can be input by programming tool software.

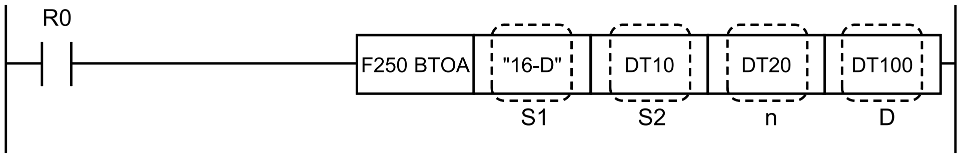
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area exceeds the 6-word area starting from [D]

24.2 F250 BTOA (Multiple Binary Data to ASCII Data String Conversion)

Converts 16-bit/32-bit binary data to an ASCII code character string.

■ Instruction format



■ Operands

Items	Settings
S1	Control character string
S2	Starting number of area storing binary data
n	Conversion method
D	Starting number of the area storing the ASCII code of conversion result

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier (Note 1)	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•			•		•	
S2	•	•	•	•	•	•	•	•	•	•	•					•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

(Note 1) A character constant cannot be specified.

■ Outline of operation

Converts the binary data stored in the area specified by S2 to ASCII data using the conversion method of n according to the 4 control characters specified by S1. The converted result is stored in the area specified by D.

■ Specifying each item

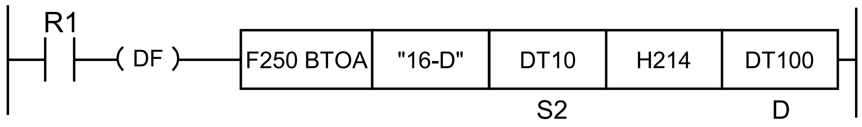
Specifying control character strings and their meanings [S1]

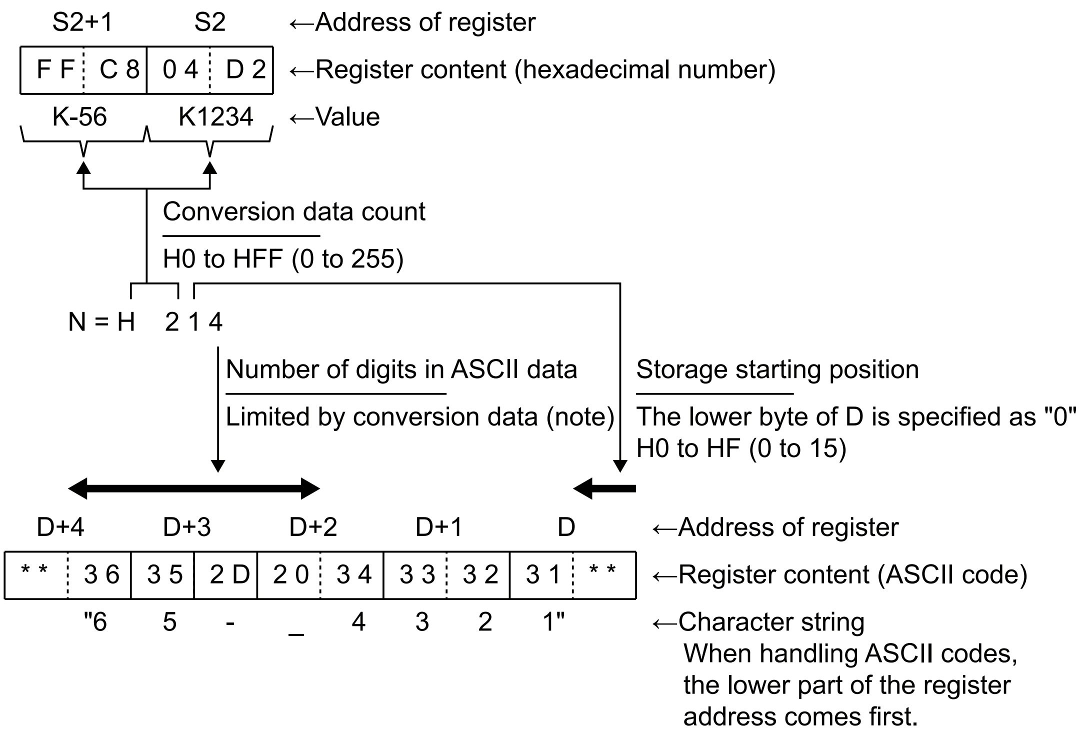
"16-D"	Converts 16-bit data to decimal ASCII data
"32−D"	Converts 32-bit data to decimal ASCII data
"16+H"	Converts 16-bit data to hexadecimal ASCII data (normal direction)
"32+H"	Converts 32-bit data to hexadecimal ASCII data (normal direction)
"16-H"	Converts 16-bit data to hexadecimal ASCII data (reverse direction)
"32−H"	Converts 32-bit data to hexadecimal ASCII data (reverse direction)

(Note 1) Details of normal and reverse directions are described later

Specifying the conversion method [n]

Example of converting 16-bit data (K1234 and K56) to decimal ASCII codes





Note

- Number of digits in ASCII data

  When the number of digits of the ASCII data is larger than the converted result, a "_" (space) is stored before the data.

  - When converting 16-bit data to hexadecimal ASCII data

    Specified range: H1 to H4

    When less than H4, the specified number of digits is stored from the lower bytes. If the digit number of the original data is larger with a specification less than H4, this is an error

  - When converting 32-bit data to hexadecimal ASCII data

    Specified range: H1 to H8

    When less than H8, the specified number of digits is stored from the lower bytes. If the digit number of the original data is larger with a specification less than H8, this is an error

  - When converting to decimal ASCII data

    Specified range: H1 to HF

    Source data is treated as signed binary data. When it is a negative number, the minus sign "-" is added.

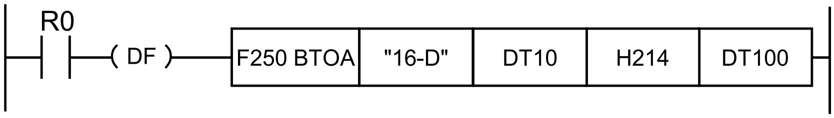
  About normal direction and reverse direction (only when converting to hexadecimal ASCII data)

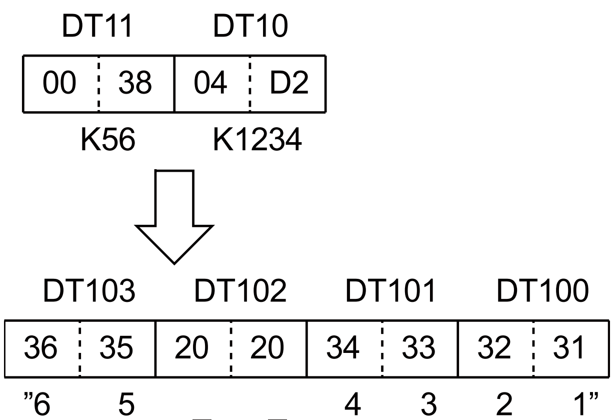
Conversion example

- Converting 16-bit data (K1234 and K56) to decimal ASCII data

```
DT10 = K 1234 → "1234__56"
DT11 = K 56
```

Number of converted data is "2", starting position for storage is "0", and size of the storage area is "4"



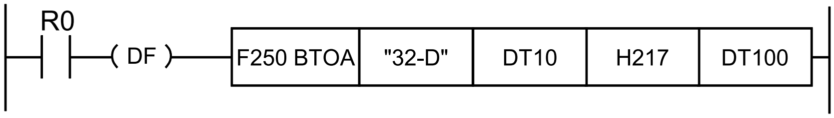


- Converting 32-bit data (K1234 and K56789) to decimal ASCII data

```
DT10,11 = K 1234 → "__1234__56789"
```

DT12,13 = K 56789

Number of converted data is "2", starting position for storage is "1", and size of the storage area is "7"



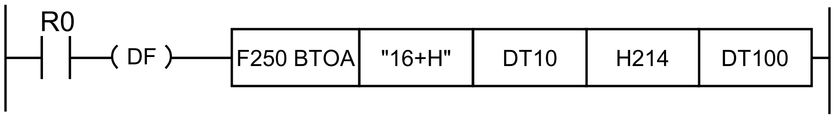
DT13		DT12		DT11		DT10	
00	00	DD	D5	00	38	04	D2
K56789				K1234			

DT107		DT106		DT105		DT104		DT103		DT102		DT101		DT100	
	39	38	37	36	35	20	20	34	33	32	31	20	20	20	
	”9	8	7	6	5	_	_	4	3	2	1	_	_	_”	

- Converting 16-bit data (H0123 and H89AB) to hexadecimal ASCII data

DT10 = H 123 → "2301AB89" DT11 = H 89AB

Number of converted data is "2", starting position for storage is "1", and size of the storage area is "4" (normal direction)



DT11		DT10	
89	AB	01	23

DT104		DT103		DT102		DT101		DT100	
	39	38	42	41	31	30	33	32	
	”9	8	B	A	1	0	3	2”	

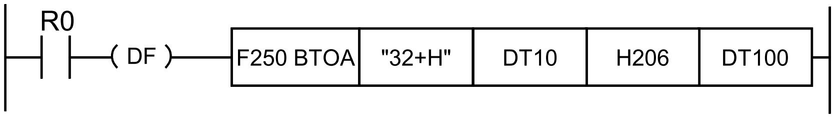
For the reverse direction (when "16+H" is "16-H")

DT104		DT103		DT102		DT101		DT100	
	42	41	39	38	33	32	31	30	
	”B	A	9	8	3	2	1	0”	

- Converting 32-bit data (H00000123 and H0089ABCD) to hexadecimal ASCII data (normal direction)

```
DT10,11 = H 123 → "230100CDAB89"
DT12,13 = H 89ABCD
```

Number of converted data is "2", starting position for storage is "0", and size of the storage area is "6"



DT13		DT12		DT11		DT10	
00	89	AB	CD	00	00	01	23

↓

DT105		DT104		DT103		DT102		DT101		DT100	
39	38	42	41	44	43	30	30	31	30	33	32
”9	8	B	A	D	C	0	0	1	0	3	2”

For the reverse direction (when "32+H" is "32-H")

DT105		DT104		DT103		DT102		DT101		DT100	
44	43	42	41	39	38	33	32	31	30	30	30
”D	C	B	A	9	8	3	2	1	0	0	0”

■ Flag operations

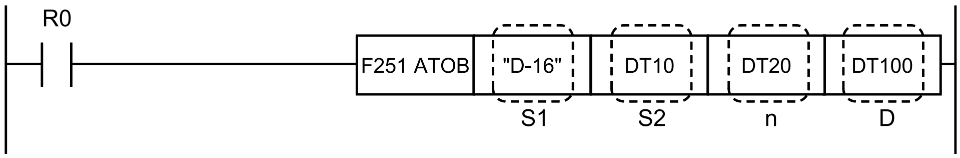
Name	Description
R9007 R9008 (ER)	When there is an error in the control string specified by S1
	When the conversion format specified by S1 is in decimal, and the direction of converted data is changed to the normal direction
	When the conversion format specified by S1 is in hexadecimal, and the size of the area for storing ASCII codes specified by N exceeds the rated value (Rated value for 16-bit data: 4) (Rated value for 32-bit data: 8)

Name	Description
	When the number of the conversion data specified by N is 0
	When the converted result exceeds the size of the area for storing ASCII codes specified by N
	When the converted result exceeds the area
	When the area is exceeded in index modification

24.3 F251 ATOB (Multiple ASCII Data Strings to Binary Data Conversion)

Converts ASCII code character strings to 16-bit/32-bit binary data.

■ Instruction format



■ Operands

Items	Settings
S1	Control character string
S2	Starting number of the area storing the ASCII code
n	Conversion method
D	Starting number of the area for storing the binary data of the converted result

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier (Note 1)	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●			●		●	
S2	●	●	●	●	●	●	●	●	●	●	●					●	
n	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

(Note 1) A character constant cannot be specified.

■ Outline of operation

Converts the ASCII data stored in the area specified by S2 to binary data using the conversion method in n, according to the four control characters specified in S1. The converted result is stored in the area specified by D.

■ Specifying each item

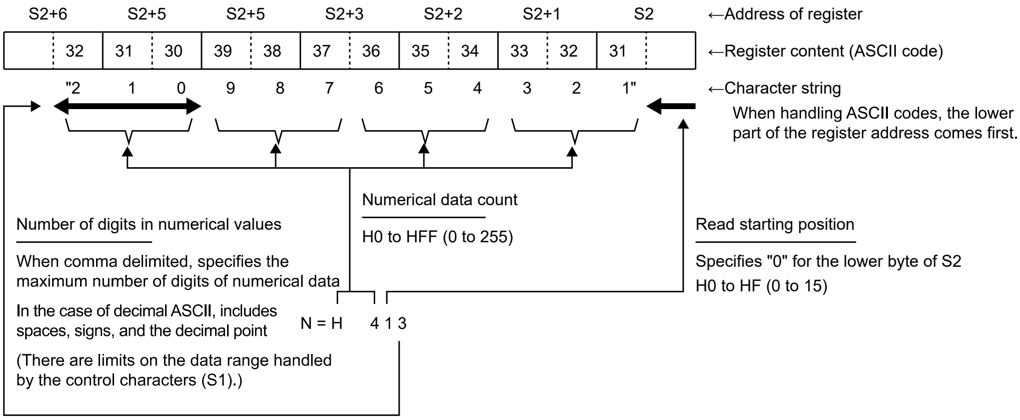
● Specifying control character strings and their meanings [S1]

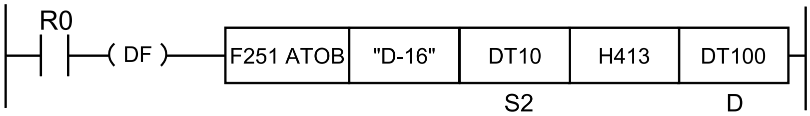
		Range of data that can be handled
"D-16"	Convert decimal ASCII data to 16-bit data	−32,768 to +32767
"D-32"	Convert decimal ASCII data to 32-bit data	-2,147,483,648 to +2,147,483,647
"H+16"	Convert hexadecimal ASCII data to 16-bit data (forward direction)	0 to FFFF

		Range of data that can be handled
"H+32"	Convert hexadecimal ASCII data to 32-bit data (forward direction)	0 to FFFFFFFF
"H-16"	Convert hexadecimal ASCII data to 16-bit data (reverse direction)	0 to FFFF
"H-32"	Convert hexadecimal ASCII data to 32-bit data (reverse direction)	0 to FFFFFFFF

(Note 1) Details of normal and reverse directions are described later

- Specifying the conversion method [n]
  - Example of converting the ASCII data string "123456789012" to four sets of three decimal digits





  - When converting via the above program

D+3		D+2		D+1		D		
00	0C	03	15	01	C8	00	7B	←Register content (hexadecimal number)
K12		K789		K456		K123		←Value in decimal

←Address of register

- About normal direction and reverse direction (only when converting to hexadecimal ASCII data)

  For hexadecimal ASCII data, conversions in the forward and reverse directions are possible. Example of converting"0123456789ABCDEF"

	S2+7		S2+6		S2+5		S2+5		S2+3		S2+2		S2+1		S2+0		
←Address of register																	
←Register content (ASCII code)	46	45	44	43	42	41	39	38	37	36	35	34	33	32	31	30	
←Character string	"F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0"	

⇩

Control character string		Conversion method (representative example)		DT13		DT12		DT11		DT10	
H+16	(Forward direction)	H404	(4 digits × 4)	EF	CD	AB	89	67	45	23	01
H-16	(Reverse direction)	H404	(4 digits × 4)	CD	EF	89	AB	45	67	01	23
H+16	(Forward direction)	H403	(3 digits × 4)	* B	9A	* 8	67	* 5	34	* 2	01
H-16	(Reverse direction)	H403	(3 digits × 4)	* 9	AB	* 6	78	* 3	45	* 0	12
H+32	(Forward direction)	H208	(8 digits × 2)	EF	CD	AB	89	67	45	23	01
H-32	(Reverse direction)	H208	(8 digits × 2)	89	AB	CD	EF	01	23	45	67
H+32	(Forward direction)	H205	(5 digits × 2)	* *	* 9	78	56	* *	* 4	23	01
H-32	(Reverse direction)	H205	(5 digits × 2)	* *	* 5	67	89	* *	* 0	12	34

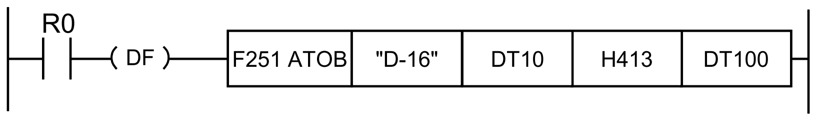
* "0" is entered to unused digits.

■ Conversion example

● Example of converting to four sets of three decimal digits (when there is no comma",")

	Converts to 16-bit data
"123456789012" →	DT100 = K 123 DT101 = K 456 DT102 = K 789 DT103 = K 12

- When the number of numeric data items is"4", starting position for reading is"1", number of digits is"3"



DT16		DT15		DT14		DT13		DT12		DT11		DT10	
	32	31	30	39	38	37	36	35	34	33	32	31	
	"2	1	0	9	8	7	6	5	4	3	2	1"	

⇩

DT103		DT102		DT101		DT100	
00	0C	03	15	01	C8	00	7B
K12		K789		K456		K123	

- When converting to 32-bit data (when"D-16"is"D-32")

DT107	DT106	DT105	DT104	DT103	DT102	DT101	DT100
00 00	00 0C	00 00	03 15	00 00	01 C8	00 00	00 7B
K12		K789		K456		K123	

- Example of converting to three sets of four hexadecimal digits

	Converts to 16-bit data in the forward direction
“001209AB000E” →	DT100 = K 1200 DT101 = K AB09 DT102 = K 0E00

- When the number of numeric data items is"3", starting position for reading is"1", number of digits to be converted is"4"

R0 (DF)	F251 ATOB	"H+16"	DT10	H314	DT100

DT16	DT15	DT14	DT13	DT12	DT11	DT10
45	30 30	30 42	41 39	30 32	31 30	30
”E	0 0	0 B	A 9	0 2	1 0	0”

⇩

DT102	DT101	DT100
0E 00	AB 09	12 00

- When converting to 16-bit data in the reverse direction (when"H+16"is"H-16")

DT102	DT101	DT100
00 0E	09 AB	00 12

- When converting to 32-bit data in the forward direction (when"H+16"is"H+32")

DT105	DT104	DT103	DT102	DT101	DT100
00 00	0E 00	00 00	AB 09	00 00	12 00

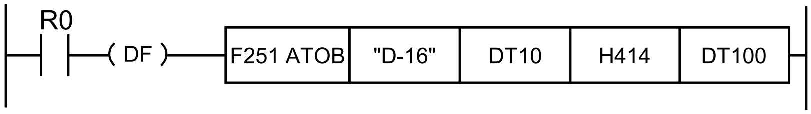
- When converting to 32-bit data in the reverse direction (when"H+16"is"H-32")

DT105	DT104	DT103	DT102	DT101	DT100
00 00	00 0E	00 00	09 AB	00 00	00 12

- Example of converting to four sets of decimal numbers (when there is a comma","separator)

“12,345,6789,0,” → The character string ends in a comma	DT100 = K 12 DT101 = K 345 DT102 = K 6789 DT103 = K 0

- When the number of numeric data items is"4", starting position for reading is"1", number of digits is"4"(Converts to 16-bit data)



DT17		DT16		DT15		DT14		DT13		DT12		DT11		DT10	
	2C	30	2C	31	30	39	38	2C	35	34	33	2C	32	31	
	”,	0	,	9	8	7	6	,	5	4	3	,	2	1”	

⇩

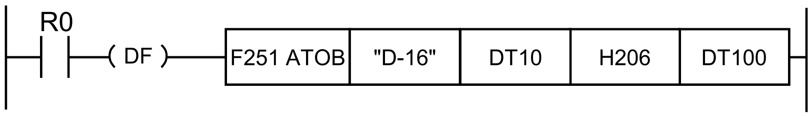
DT103		DT102		DT101		DT100	
00	00	1A	85	01	59	00	0C
K0		K6789		K345		K12	

(Note 1) Specify the maximum number of digits.

- Example of converting to two sets of five decimal digits with decimal points (when there is no comma",")

“1234.50006.7” →	DT100 = K 12345 DT101 = K 67

- When the number of numeric data items is"2", starting position for reading is"0", number of digits is"6", when converting to 16-bit data



DT15		DT14		DT13		DT12		DT11		DT10	
37	2E	36	30	30	30	35	2E	34	33	32	31
”7	.	6	0	0	0	5	.	4	3	2	1”

⇩

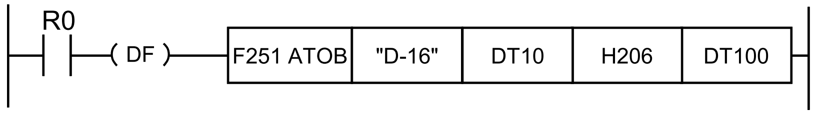
DT101		DT100	
00	43	30	39
K67		K12345	

(Note 1) A decimal point is also counted as a digit

- Example of converting to two sets of decimal digits with decimal points (when there is a comma","separator)

“1234.5,6.7” → The character string ends in a comma	DT100 = K 12345 DT101 = K 67

- When the number of numeric data items is"2", starting position for reading is"0", number of digits is"6", when converting to 16-bit data



DT15		DT14		DT13		DT12		DT11		DT10	
	2C	37	2E	36	2C	35	2E	34	33	32	31
	,	7	.	6	,	5	.	4	3	2	1”

DT101		DT100	
00	43	30	39
K67		K12345	

(Note 1) A decimal point is also counted as a digit

■ Particular examples

- If there is numeric data larger than the specified number of digits between commas (example: four sets of decimal numbers, and number of digits is four)

“1234, 567890, 12, 345” →	K 1234 K 5678 K90: The overflowed numbers become one numeric data K12 K345: Ignored

- If there is no value between commas (example: four sets of decimal numbers)

“123, 456,, 78” →	Operation error

- If there is only a decimal point between commas (example: three sets of decimal numbers with decimal points)

“1234. 5,.,6.7” →	Operation error *If there is any number, for example "2." or ".2", it is converted

■ Flag operations

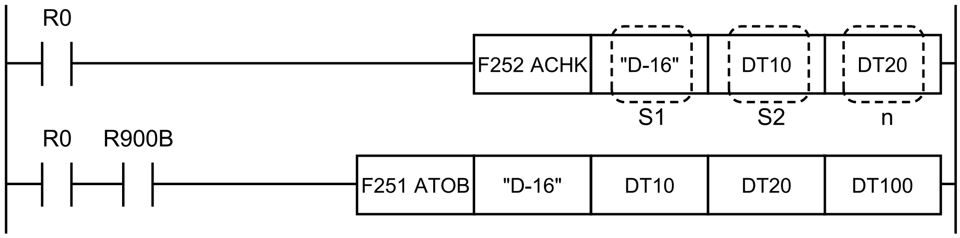
Name	Description
R9007 R9008 (ER)	When there is an error in the control string specified by S1
	When the conversion format specified by S1 is in decimal, and the direction of converted data is changed to the normal direction
	When the conversion format specified by S1 is hexadecimal, and the size of the area for storing ASCII codes specified by n exceeds the rated value (Rated value for 16-bit data: 4)

Name	Description
	(Rated value for 32-bit data: 8)
	The ASCII code specified by S2 contains any code other than 0 to F, a sign, a space, a dot, or a comma
	The number of converted blocks specified by n is 0
	The size of the area for storing ASCII codes specified by n is 0
	The ASCII code to be converted exceeds the area
	When the converted result exceeds the area
	The converted result exceeds the converted data scale specified by n
	When the area is exceeded in index modification

24.4 F252 ACHK (Multiple ASCII Data Strings ASCII Code Check)

Checks whether the specified ASCII data is correct.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the control character string, or character string data
S2	Starting number of the area storing the ASCII code
n	Area storing the conversion method, or constant data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier (Note 1)	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•			•		•	
S2	•	•	•	•	•	•	•	•	•	•	•					•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

(Note 1) A character constant cannot be specified.

■ Outline of operation

- This instruction checks whether the ASCII code stored in the area specified by S2 can be correctly converted using the conversion method specified by n in accordance with the 4-character control characters specified by S1.
- It checks whether the character string to be converted by the F251 ATOB instruction can be converted.

  This instruction can be executed before the character string is converted by the F251 ATOB instruction and if an error is found in the data, can control to not execute the F251 ATOB instruction. Specify S1, S2, and n to be the same values as in the F251 ATOB instruction. As a result of the check, the special relay R900B turns ON if the data is correct and OFF if there is an error.

■ Specifying each item

The method to specify S1, S2, and n is the same as for the F251 ATOB instruction, so refer to the description of F251 ATOB ASCII to Binary Conversion.

■ Flag operations

Name	Description
R9007 R9008 (ER)	When there is an error in the control string specified by S1
	When the conversion format specified by S1 is in decimal, and the direction of converted data is changed to the normal direction
	When the conversion format specified by S1 is hexadecimal, and the size of the area for storing ASCII codes specified by n exceeds the rated value (Rated value for 16-bit data: 4) (Rated value for 32-bit data: 8)
	The number of converted blocks specified by n is 0
	The size of the area for storing ASCII codes specified by n is 0
	The ASCII code to be converted exceeds the area
	When the area is exceeded in index modification

24.5 Overview of String Instructions F257 SCMP to F265 SREP

■ Data table structure

The character string data table sets the character string size, number of characters, and character data.

	H	L	
S:	String size		Maximum number of characters that can be stored
S+1:	Character count		Number of characters stored in the string.
S+2:	Character 2 (byte 1)	Character 1 (byte 0)	String data
S+3:	Character 4 (byte 3)	Character 3 (byte 2)	
S+4:	Character 6 (byte 5)	Character 5 (byte 4)	
.	.	.	
.	.	.	

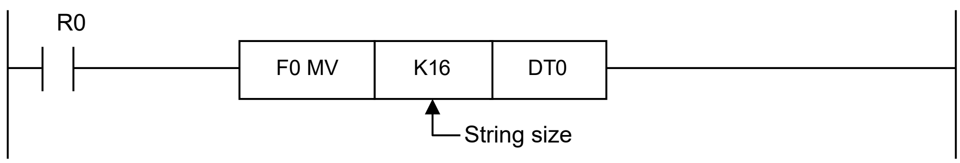
e.g. To set the following data table: Character string size (10), number of character (5), character data "ABCDE"

	H	L
DT0	10	
DT1	5	
DT2	"B"	"A"
DT3	"D"	"C"
DT4		"E"
DT5		
DT6		

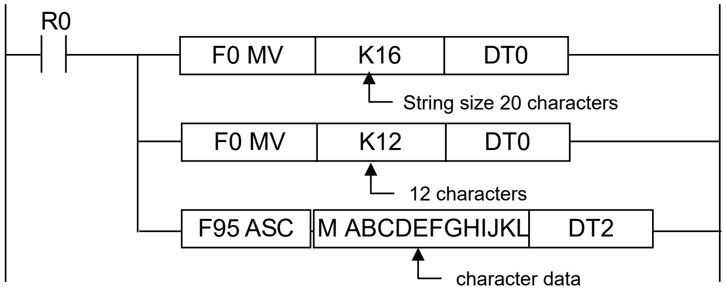
■ How to set a data table

- Set the character string size and the number of characters. The F0 (MV) instruction is used to set the values.
- Set the character string. The F95 (ASC) instruction is used to set the characters.

e.g. When the following data table is specified in DT0: String size (16), no characters specified



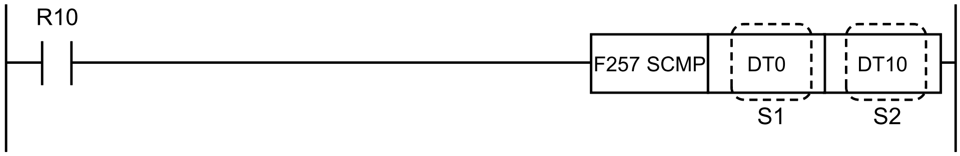
e.g. When the following data table is specified in DT0: String size (20), number of characters (12), character data "ABCDEFGHIJKL"



24.6 F257 SCMP (Comparing Character Strings)

Compares two specified character strings, and outputs the judgment result to a special internal relay.

■ Instruction format



■ Operands

Items	Settings
S1	Character string 1 for comparison
S2	Character string 2 for comparison

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S1	•	•	•	•	•	•	•	•	•	•	•					•	
S2	•	•	•	•	•	•	•	•	•	•	•					•	

■ Outline of operation

- The character string specified for [S1] is compared to the character string specified for [S2],and the judgment result is output to special internal relays R9009 to R900C (judgment flags for comparison instructions).
- R9009 to R900C are assigned based on whether [S1] or [S2] is larger or smaller, as shown in the table below.

Relationship of S1 and S2	Flag			
	R900A	R900B	R900C	R9009
	>	=	<	Carry
S1 < S2	OFF	OFF	ON	Indefinite
S1 = S2	OFF	ON	OFF	OFF
S1 > S2	ON	OFF	OFF	Indefinite

■ Operation example

Operation of instruction format description program

When internal relay R10 is ON, data registers DT1 and DT11 are compared. In this case, it is determined that [S1] < [S2], and R900C turns ON.

	H	L
DT0	10 (string size)	
DT1	4 (character count)	
DT2	"B" (Byte 1)	"A" (Byte 0)
DT3	"0" (Byte 3)	"C" (Byte 2)
DT4	(Byte 5)	(Byte 4)
DT5	(Byte 7)	(Byte 6)
DT6	(Byte 9)	(Byte 8)

	H	L
DT10	8 (string size)	
DT11	5 (character count)	
DT12	"B" (Byte 1)	"A" (Byte 0)
DT13	"0" (Byte 3)	"C" (Byte 2)
DT14	(Byte 5)	"E" (Byte 4)
DT15	(Byte 7)	(Byte 6)

■ Precautions for programming

- If the number of characters is different, the greater/lesser relationship is as shown below.

S1	Greater/lesser	S2
“ABCDE”	=	“ABCDE”
“ABCD”	<	“ABCDE”
“B”	>	“ABCDE”

- Comparison of character strings is performed in sequence from byte 0, one character at a time.
- If one character string has fewer characters than the other, it may still be handled as larger if a large character code is used when the comparison is made.

  e.g. "B">"ABCDE"
- To specify a character string, indicate the number of the area in which the character string size and number of characters have been specified. For detailed information about the table configuration of the data area, refer to"24.5 Overview of String Instructions F257 SCMP to F265 SREP".

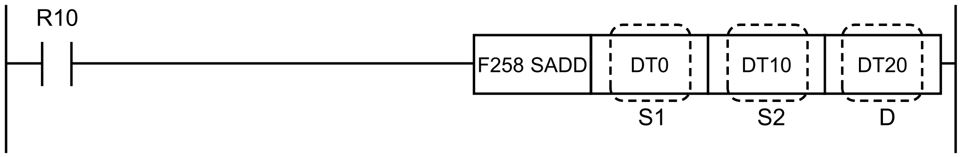
■ Flag operations

Name	Description
R9007	Turns ON when the area is exceeded in index modification.
R9008 （ER）	Turns ON when number of characters is greater than the character string size

24.7 F258 SADD (Character String Addition)

Concatenates one character string with another.

■ Instruction format



■ Operands

Items	Settings
S1	Character string to be concatenated
S2	Character string to be concatenated
D	Area in which the concatenated character strings are stored

■ Devices that can be specified (indicated by ●)

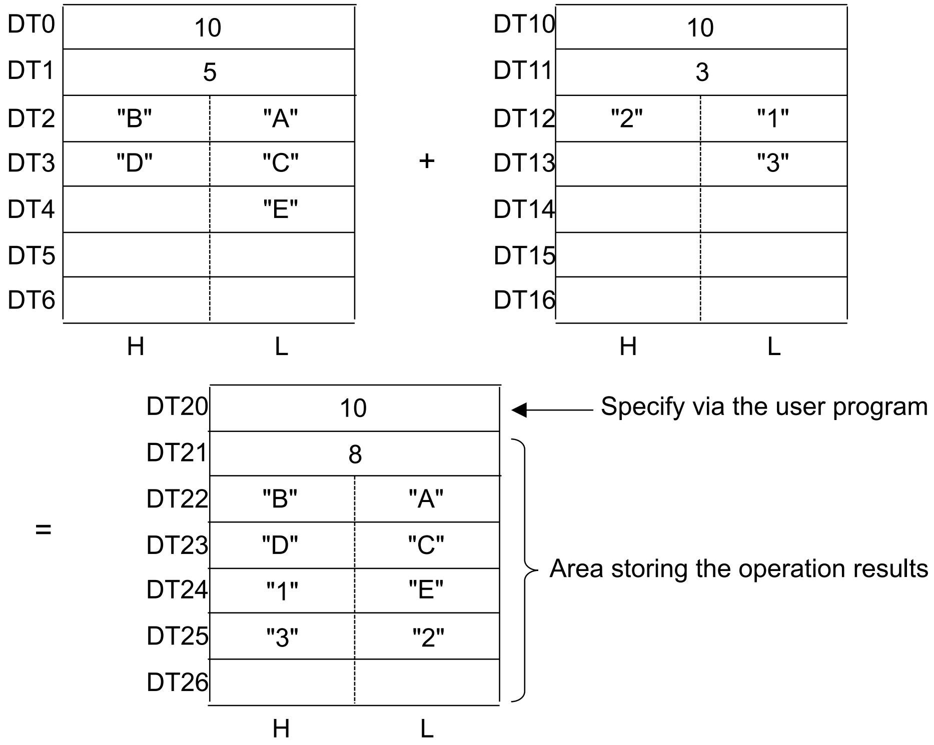
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●					●	
S2	●	●	●	●	●	●	●	●	●	●	●					●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The character string specified by [S1] is concatenated with the character string specified by [S2], and the result is stored in the character string specified by [D].
- At the start of the area for storing results [D], specify the character string size via the user program.

■ Operation example

Operation of instruction format description program



■ Precautions for programming

If the result of the concatenation operation is larger than the character string size of [D], only as many characters as will fit in [D] are stored.

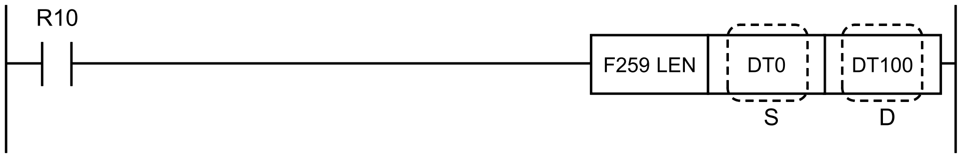
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when number of characters is greater than the character string size
R9009 (CY)	Turns ON when the operation result is greater than the size of the character string specified by [D]

24.8 F259 LEN (Character String Length)

Determines the number of characters stored in a character string.

■ Instruction format



■ Operands

Items	Settings
S	Character string
D	Area that stores the number of characters in the calculation result

■ Devices that can be specified (indicated by ●)

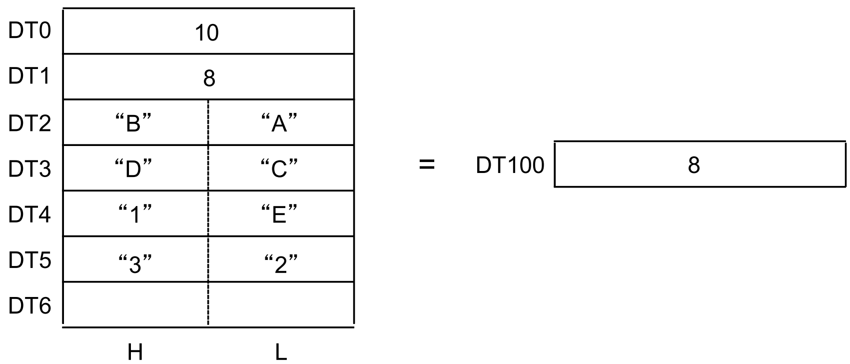
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S	●	●	●	●	●	●	●	●	●	●	●					●	
D	●	●	●	●	●	●	●	●	●							●	

■ Outline of operation

The number of characters in the character string specified by [S] is determined, and the result is stored in [D].

■ Operation example

Operation of instruction format description program



■ Precautions for programming

If the number of characters is greater than the character string size, an operation error occurs.

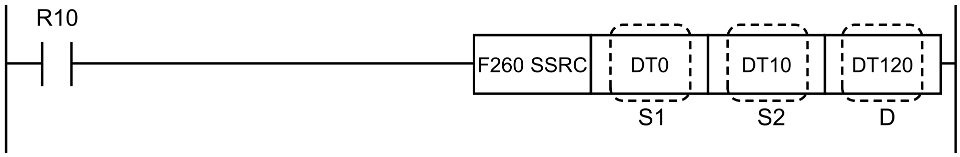
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when number of characters is greater than the character string size

24.9 F260 SSRC (Search for Character String)

Searches for the specified character string.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the character data to be searched (character string or character constant)
S2	Character string to be searched
D	Area storing the search result

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●					●	
S2	●	●	●	●	●	●	●	●	●	●	●					●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The character data specified by [S1] is searched for the character string specified by [S2].
- The number of characters that are the same based on the search result is stored in [D] and the first matching relative position (in byte units) is stored in [D+1].

■ Operation example

Operation of instruction format description program

The characters in DT0 are searched from the character string in DT10 and the result is stored in DT120.

Characters to search for

	H	L
DT0	10 (string size)	
DT1	3 (character count)	
DT2	"F" (Byte 1)	"E" (Byte 0)
DT3	(Byte 3)	"G" (Byte 2)
DT4	(Byte 5)	(Byte 4)
DT5	(Byte 7)	(Byte 6)
DT6	(Byte 9)	(Byte 8)

Search →

Character table to be searched

	H	L
DT10	10 (string size)	
DT11	8 (character count)	
DT12	"B" (Byte 1)	"A" (Byte 0)
DT13	"D" (Byte 3)	"C" (Byte 2)
DT14	"F" (Byte 5)	"E" (Byte 4)
DT15	"H" (Byte 7)	"G" (Byte 6)
DT16	(Byte 9)	(Byte 8)

→

DT120	1	Quantity of matching character data
DT121	5	Position of first match

■ Precautions for programming

- Specify a number of characters so that [S1] is less than or equal to [S2].
- For [S1+1], the number of characters in the character string on the search side, specify a value for the number of characters to be searched.

e.g.

4 (string size)	
1 (character count)	
"B"	"A"
"D"	"C"

← When the character count is 1, searches for the character "
When the character count is 2, searches for the characters "AB"as a single block

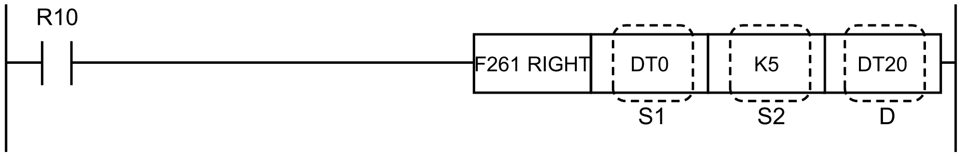
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when number of characters is greater than the character string size

24.10 F261 RIGHT (Right Retrieve from Character String)

Retrieves a character string with the specified number of characters from the right side of a character string.

■ Instruction format



■ Operands

Items	Settings
S1	Character string
S2	Area storing the number of characters, or constant data
D	Area storing the character string

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●					●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

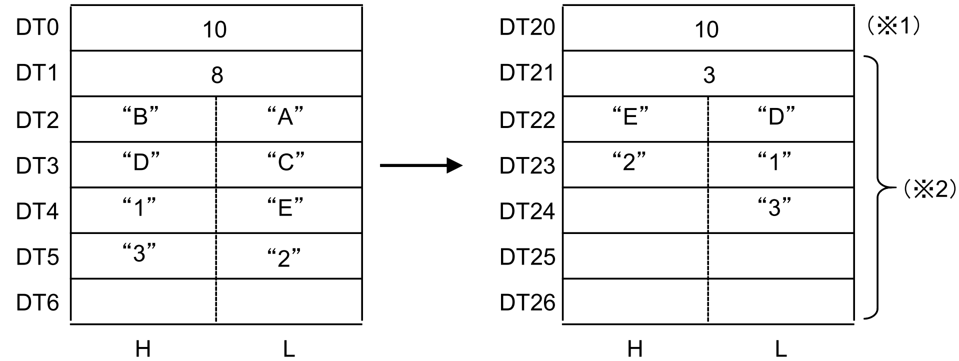
■ Outline of operation

- The number of characters specified by [S2] are retrieved from the right side of the character string (the end of character data) specified by [S1], and are transferred to the character string specified by [D].
- At the start of the area for storing results [D], specify the character string size via the user program.

■ Operation example

Operation of instruction format description program

Five characters are retrieved from the end of character string DT0 and transferred to DT20.



(*1): Specify via the user program

(*2): Area storing the operation results

■ Precautions for programming

- The character data of [D] prior to the operation is cleared.
- 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.
- If the number of characters specified by [S2] is greater than the size of the character string specified by [D], then the number of characters equal to the size of the character string specified by [D] are transferred.

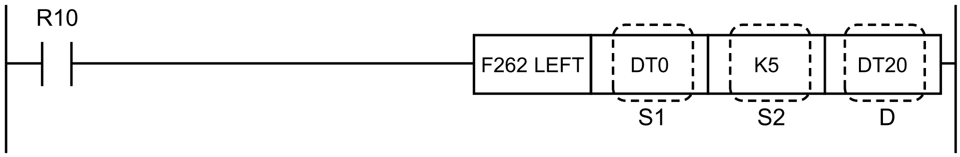
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when number of characters is greater than the character string size
R9009 (CY)	Turns ON when the operation result is greater than the size of the character string specified by [D]

24.11 F262 LEFT (Left Retrieve from Character String)

Retrieves the specified number of characters from the left side of a character string.

■ Instruction format



■ Operands

Items	Settings
S1	Character string
S2	Area storing the number of characters, or constant data
D	Area storing the character string

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•					•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

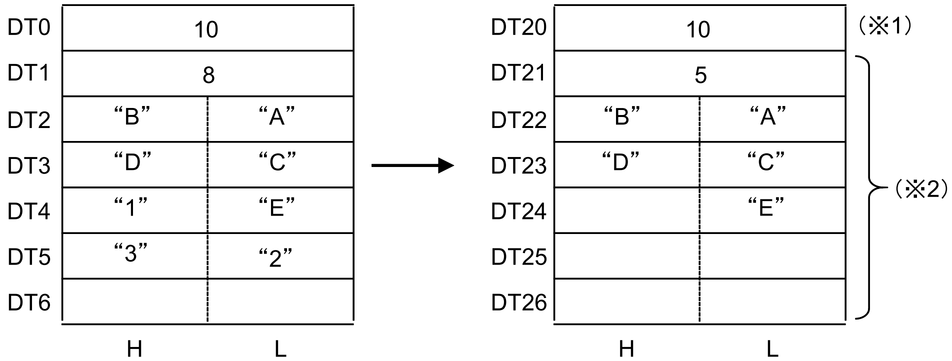
■ Outline of operation

- The number of characters specified by [S2] are retrieved from the left side of the character string (the start of character data) specified by [S1], and are transferred to the character string specified by [D].
- At the start of the area for storing results [D], specify the character string size via the user program.

■ Operation example

Operation of instruction format description program

Five characters are retrieved from the start of the character string in DT0 and transferred to DT20.



(*1): Specify via the user program

(*2): Area storing the operation results

■ Precautions for programming

- The character data of [D] prior to the operation is cleared.
- If the number of characters specified by [S2] is greater than the number of characters in the character string specified by [S1], then the number of characters in the character string specified by [S1] are transferred.
- If the number of characters specified by [S2] is greater than the size of the character string specified by [D], then the number of characters equal to the size of the character string specified by [D] are transferred.

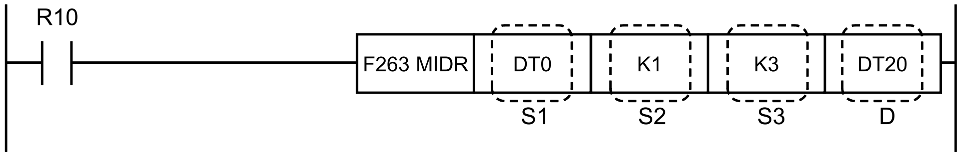
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when number of characters is greater than the character string size
R9009 (CY)	Turns ON when the operation result is greater than the size of the character string specified by [D]

24.12 F263 MIDR (Read from Any Position in Character String)

Retrieves a character string of the specified number of characters from the specified position in a character string.

■ Instruction format



■ Operands

Items	Settings
S1	Character string
S2	Area storing the character string position, or constant data
S3	Area storing the number of characters, or constant data
D	Area storing the character string

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•					•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S3	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

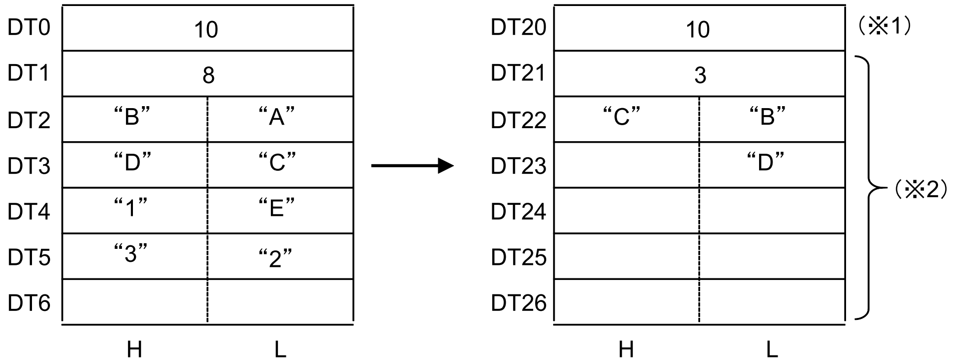
■ Outline of operation

- The number of characters specified by [S3] is retrieved from the position specified by [S2] in the character string specified by [S1], and is transferred to the character string specified by [D].
- At the start of the area for storing results [D], specify the character string size via the user program.

■ Operation example

Operation of instruction format description program

Three characters are retrieved from position byte 1 (2nd character) of the DT0 character string, and are transferred to DT20.



(*1): Specify via the user program

(*2): Area storing the operation results

■ Precautions for programming

- The character data of [D] prior to the operation is cleared.
- If the number of characters specified by [S3] is greater than the number of characters in the character string specified by [S1] from the position specified by [S2], then the number of characters in the character string specified by [S1] are transferred.
- If the number of characters of the operation result is greater than the size of the character string specified by [D], then the number of characters equal to the size of the character string specified by [D] are transferred.
- The position specified by [S2] has K0 specified for the least significant byte (byte 0), and the positions are counted in the order of 0, 1, 2, etc., starting from the least significant byte.

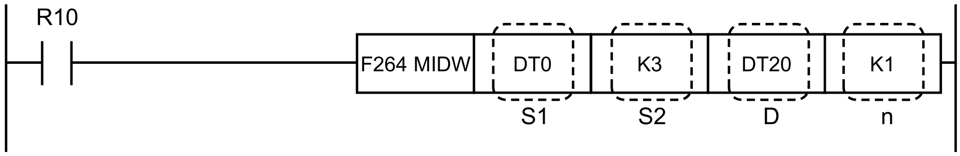
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when number of characters is greater than the character string size
	Turns ON when the number of characters specified by [S1] is less than [S2]
R9009 (CY)	Turns ON when the operation result is greater than the size of the character string specified by [D]

24.13 F264 MIDW (Write to Any Position in Character String)

These instructions write a specified number of characters from a character string to a specified position in the character string.

■ Instruction format



■ Operands

Items	Settings
S1	Character string
S2	Area storing the number of characters, or constant data
D	Starting address of the area storing a character string
n	Area storing the character string position, or constant data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier (Note 1)	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•			•		•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	
n	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

(Note 1) A character constant cannot be specified.

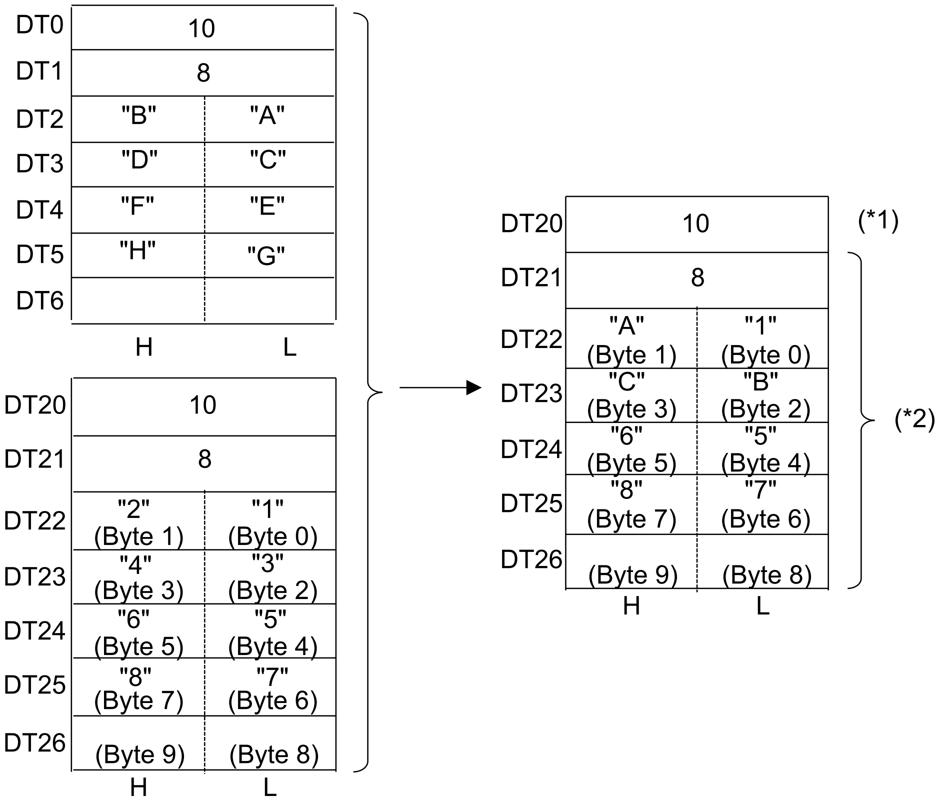
■ Outline of operation

The number of characters specified by [S2] is retrieved from the character string specified by [S1], and is transferred to the [n] position of the character string specified by [D].

■ Operation example

Operation of instruction format description program

Retrieves 3 characters from the DT0 character string, and transfers these to the byte 1 position (second character) of the DT20 character string block.



(*1): Specify via the user program

(*2): Area storing the operation results

■ Precautions for programming

- The [D] character data before calculation is not cleared. (This 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.
- If the position of [n] is greater than number of characters in the [D] character string, an operation error occurs.
- If the number of characters in the operation result is greater than the size of the [D] character string, then replacement is done only within a range the size of the [D] character string.
- The [n] position sets the least significant byte as K0 (byte 0), counting up in the order of 0, 1, 2, etc. starting from the least significant byte.

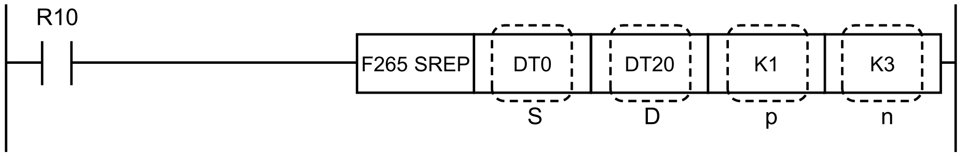
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when number of characters is greater than the character string size
	Turns ON when the number of characters of [D] < [n]
R9009 (CY)	Turns ON when the operation result is greater than the size of the character string specified by [D]

24.14 F265 SREP (Replace Character Strings)

Replaces the specified number of characters in a character string with the same number of different characters, starting from the specified position.

■ Instruction format



■ Operands

Items	Settings
S	Replacement character string
D	Starting address of the area storing a character string
p	Area storing the first byte position of the characters to be replaced, or constant data
n	Area storing the number of characters to be replaced from the source data, or constant data

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier (Note 1)	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●			●		●	
D		●	●	●	●	●	●	●	●							●	
p	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
n	●	●	●	●	●	●	●	●	●	●	●	●	●			●	

(Note 1) A character constant cannot be specified.

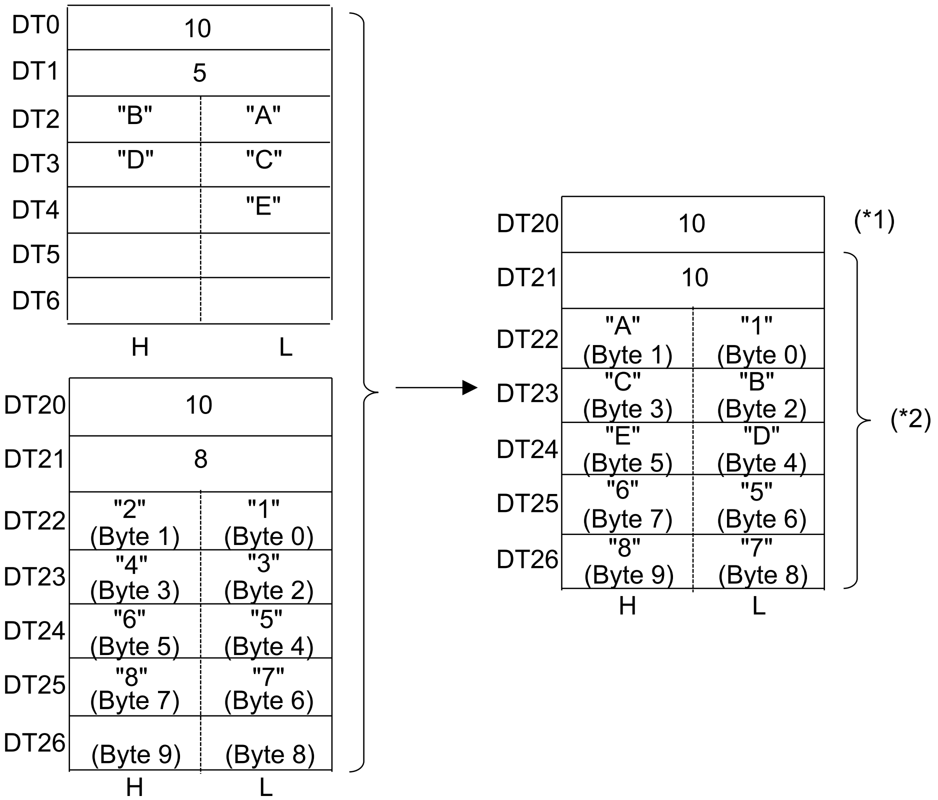
■ Outline of operation

The number of characters specified by [n] are replaced with the character string specified by [S], starting from position [p] in the character string specified by [D].

■ Operation example

Operation of instruction format description program

The DT0 character string is replaced with the number of characters in DT1 (five characters) from byte p = 1 in DT20. In this case, n = 3 characters of the data stored in the source are deleted and replaced.



(*1): Specify via the user program

(*2): Area storing the operation results

■ Precautions for programming

- The character data from [D] prior to the operation is not cleared. (This is overwritten.)
- If the number of characters in [n] is larger than the number of characters in the character string [S] subsequent to the point specified by [p], the number of characters in character string [S] subsequent to the point specified by [p] are replaced.
- If the position specified by [p] exceeds the number of characters in the character string specified by [D], an operation error occurs.
- The position specified by [p] sets the low byte as K0 (byte 0), and the positions are counted in the order 0, 1, 2, ... starting from the low byte.

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when number of characters is greater than the character string size
	Turns ON when the number of characters of [D] is less than [n]
R9009 (CY)	Turns ON when the operation result is greater than the size of the character string specified by [D]

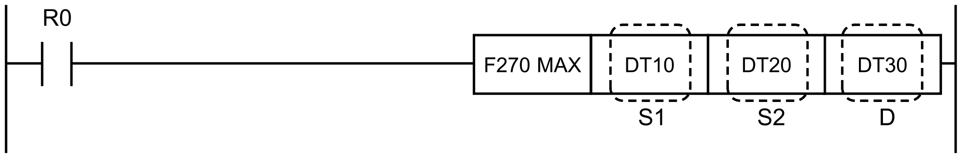
(MEMO)

25 Data Manipulation Instructions

25.1 F270 MAX (Search Maximum Value from 16-bit Data Block)

Finds the maximum value in the specified memory area range (word data table).

■ Instruction format



■ Operands

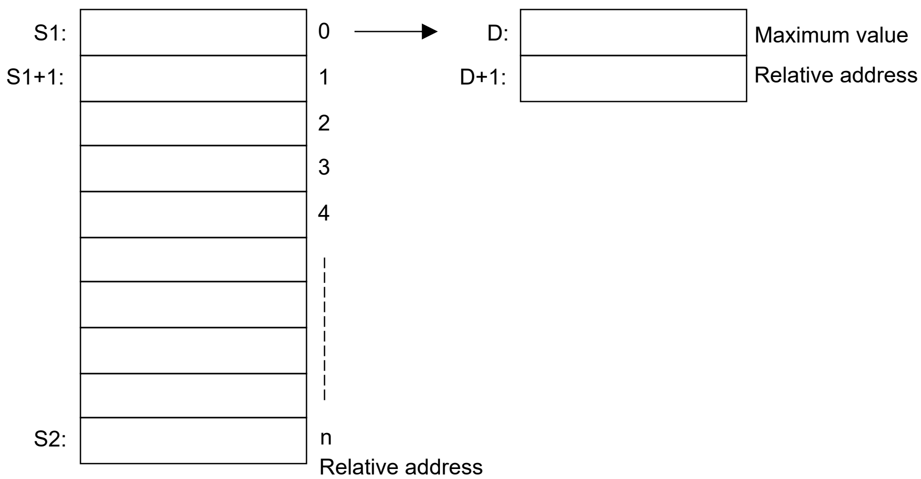
Items	Settings
S1	Starting area that stores word data
S2	Ending area that stores word data
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S1	•	•	•	•	•	•	•	•	•	•	•					•	
S2	•	•	•	•	•	•	•	•	•	•	•					•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- Searches for the maximum value in the word data tables from the area specified by [S1] to the area specified by [S2], stores the result in the area specified by [D], and stores the relative address value from [S1] in [D+1].



- If there is multiple data with the same value as the maximum value, the relative address of the first value found searching from [S1] is stored in [D+1].

■ Precautions for programming

[D+1] will stored even if it overflows the specified device area, so it may corrupt the start of other device areas. (Area overflow checks are not performed.)

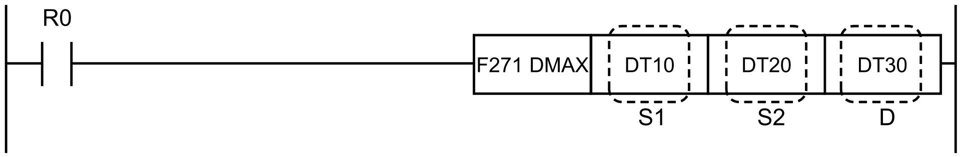
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	ON when [S1] is greater than [S2]
	Turns ON when S1 and S2 are different devices

25.2 F271 DMAX (Search Maximum Value from 32-bit Data Block)

Calculates the maximum value of the specified memory area range (double word data table).

■ Instruction format



■ Operands

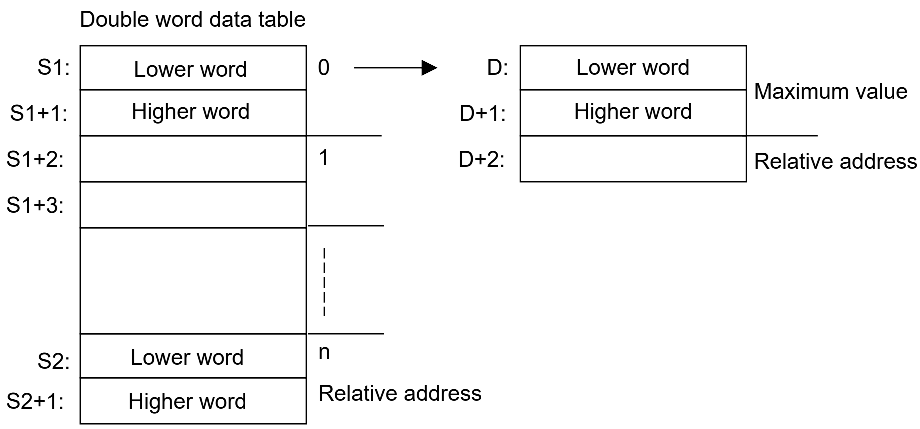
Items	Settings
S1	Starting area storing double word data
S2	Ending area storing double word data
D	Area storing the result of the operation (three words)

■ Devices that can be specified (indicated by ●)

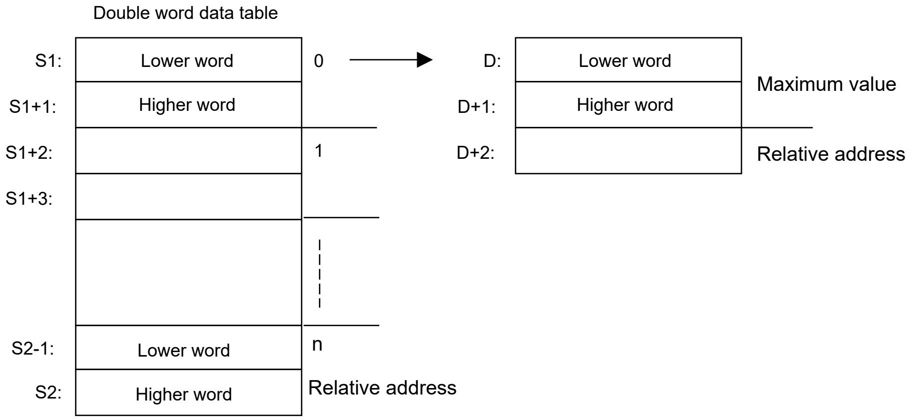
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●					●	
S2	●	●	●	●	●	●	●	●	●	●	●					●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The maximum value is searched for in the double word data table between the area specified by [S1] and the area specified by [S2] and the result is stored in the area specified by [D]. The address relative to [S1] is stored in [D+2].



- If [S2] specifies a high word of double word data, processing will take place over the same area as if a low word had been specified.



- If there is multiple data with the same value as the maximum value, the relative address of the first value found searching from [S1] is stored in [D+2].

■ Precautions for programming

- [D+2 will stored even if it overflows the specified device area, so it may corrupt the start of other device areas. (Area overflow checks are not performed.)
- The stored relative address value is counted in 32-bit units.

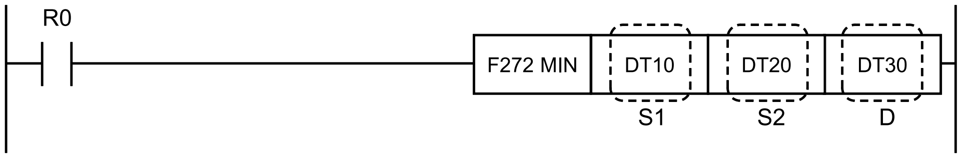
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	ON when [S1] is greater than [S2]
	Turns ON when S1 and S2 are different devices

25.3 F272 MIN (Search Minimum Value from 16-bit Data Block)

Finds the minimum value in the specified memory area range (word data table).

■ Instruction format



■ Operands

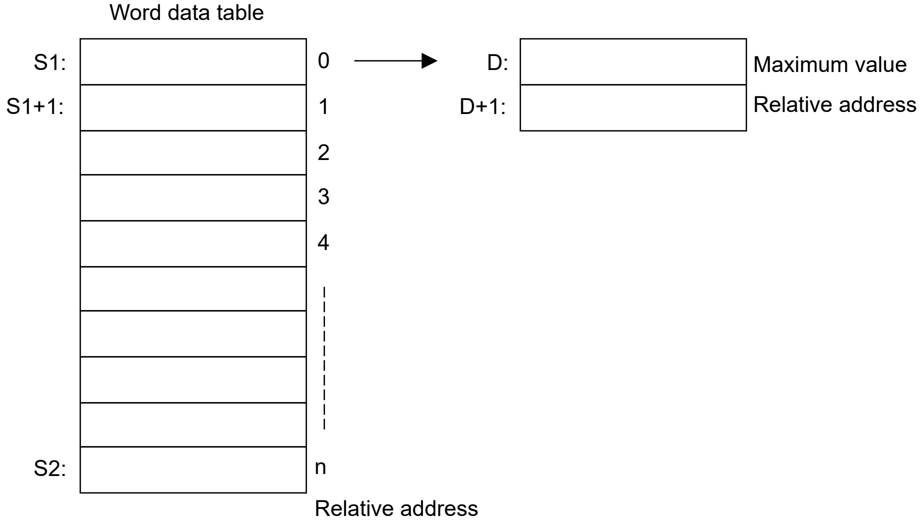
Items	Settings
S1	Starting area that stores word data
S2	Ending area that stores word data
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•					•	
S2	•	•	•	•	•	•	•	•	•	•	•					•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- Searches for a minimum value in the word data table from the area specified in [S1] to the area specified in [S2], stores the result in the area specified in [D], and stores the relative address value from [S1] in [D+1].



- When there is multiple data sharing the same minimum value, the relative address of the first result found searching from [S1] is stored in [D+1].

■ Precautions for programming

[D+1] will stored even if it overflows the specified device area, so it may corrupt the start of other device areas. (Area overflow checks are not performed.)

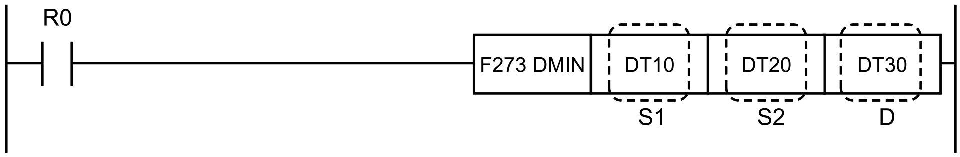
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	ON when [S1] is greater than [S2]
	Turns ON when S1 and S2 are different devices

25.4 F273 DMIN (Search Minimum Value from 32-bit Data Block)

Finds the minimum value of the specified memory area range (double word data table).

■ Instruction format



■ Operands

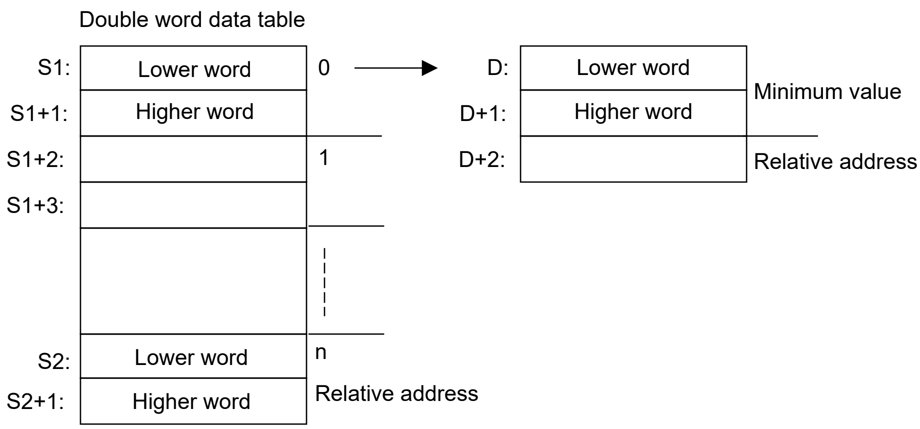
Items	Settings
S1	Starting area storing double word data
S2	Ending area storing double word data
D	Area storing the result of the operation (three words)

■ Devices that can be specified (indicated by •)

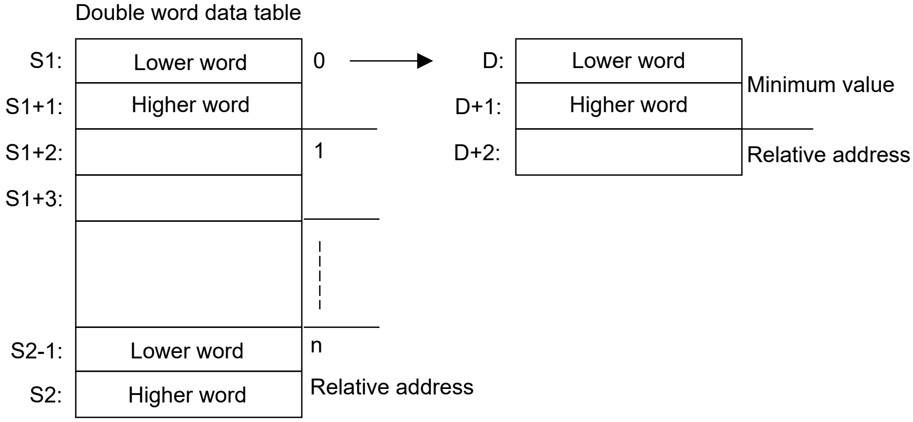
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S1	•	•	•	•	•	•	•	•	•	•	•					•	
S2	•	•	•	•	•	•	•	•	•	•	•					•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- Searches for the minimum value in the double word data table between the area specified by [S1] and the area specified by [S2] and stores the result in the area specified by [D]. The relative address value relative to [S1] is stored in [D+2].



- If [S2] specifies a high word of double word data, processing will take place over the same area as if a low word had been specified.



- When there is multiple data sharing the same minimum value, the relative address of the first result found searching from [S1] is stored in [D+2].

■ Precautions for programming

- [D+2 will stored even if it overflows the specified device area, so it may corrupt the start of other device areas. (Area overflow checks are not performed.)
- The stored relative address value is counted in 32-bit units.

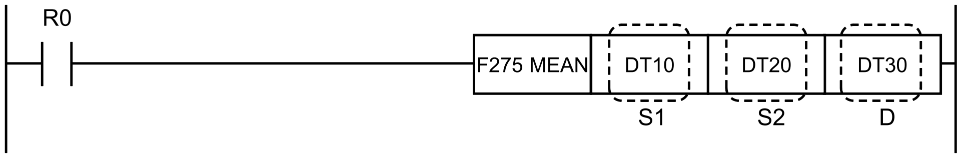
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	ON when [S1] is greater than [S2]
	Turns ON when S1 and S2 are different devices

25.5 F275 MEAN (16-bit Data Sum and Average)

Calculates the total value and mean value of the specified memory area range (word data).

■ Instruction format



■ Operands

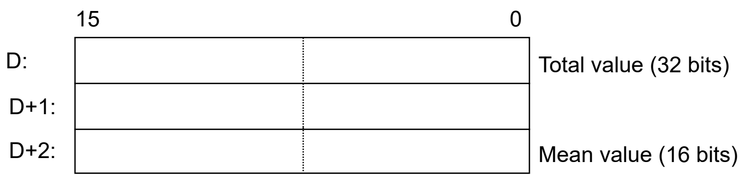
Items	Settings
S1	Starting area that stores word data
S2	Ending area that stores word data
D	Area storing the result of the operation (three words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•					•	
S2	•	•	•	•	•	•	•	•	•	•	•					•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The total value and mean value of the word data (signed) from the area specified by [S1] to the area specified by [S2] are obtained and stored in the area specified by [D].



- For the mean value, the decimal is rounded down to make an integer.

■ Precautions for programming

[D+2 will stored even if it overflows the specified device area, so it may corrupt the start of other device areas. (Area overflow checks are not performed.)

■ Flag operations

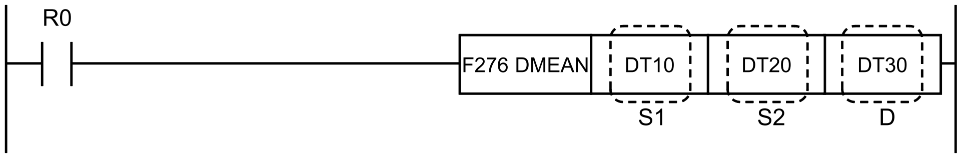
Name	Description
R9007	Turns ON when the area is exceeded in index modification.

Name	Description
R9008 (ER)	ON when [S1] is greater than [S2]
	Turns ON when S1 and S2 are different devices
R9009 (CY)	Turns ON when overflow/underflow occurs during calculation

25.6 F276 DMEAN (32-bit Data Sum and Average)

Calculates the total and mean values of the specified memory area range (double word data).

■ Instruction format



■ Operands

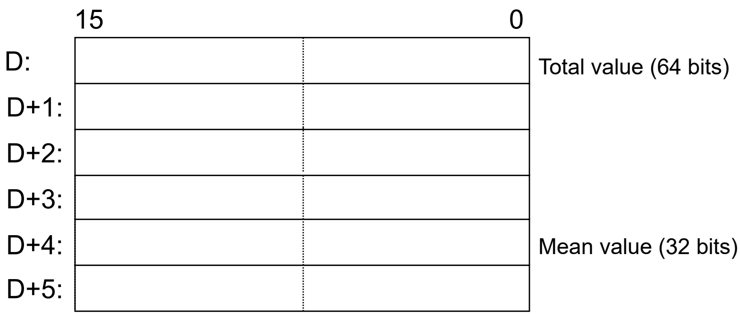
Items	Settings
S1	Starting area storing double word data
S2	Ending area storing double word data
D	Area storing the operation results (6 words)

■ Devices that can be specified (indicated by ●)

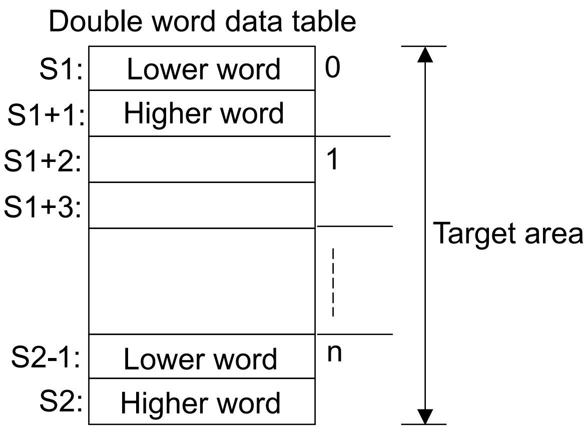
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S1	●	●	●	●	●	●	●	●	●	●	●					●	
S2	●	●	●	●	●	●	●	●	●	●	●					●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The total and mean values of the double word data (signed) from the area specified by [S1] to the area specified by [S2] are stored in the area specified by [D].



- If [S2] specifies a high word of double word data, processing will take place over the same area as if a low word had been specified.



- For the mean value, the decimal is rounded down to make an integer.

■ Precautions for programming

[D+5] will stored even if it overflows the specified device area, so it may corrupt the start of other device areas. (Area overflow checks are not performed.)

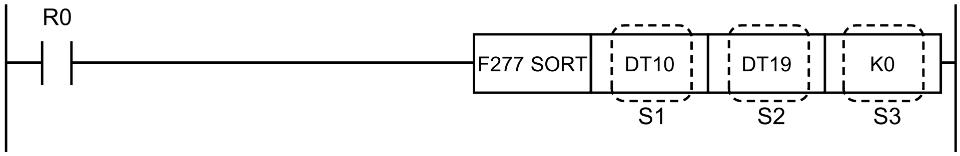
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	ON when [S1] is greater than [S2]
	Turns ON when S1 and S2 are different devices
R9009 (CY)	Turns ON when overflow/underflow occurs during calculation

25.7 F277 SORT (16-bit Data Block Sort)

Sorts the strings (word data) in the specified memory area range into ascending or descending order.

■ Instruction format



■ Operands

Items	Settings
S1	Starting area storing sort data
S2	Ending area storing sort data
S3	Area storing sort conditions, or constant data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1		•	•	•	•	•	•	•	•							•	
S2		•	•	•	•	•	•	•	•							•	
S3	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

- The word data (signed) from the area specified by [S1] to the area specified by [S2] is sorted into ascending or descending order.
- When S1 = S2, no operation takes place.
- The sort conditions are specified in [S3].

  K0: Ascending order

  K1: Descending order
- During sorting, the data from [S1] to [S2] is sorted in sequential order in accordance with the sort procedure. Note that the number of times data is compared increases proportionally to the square of the number of data words, and therefore the operation time will increase if there is a large number of data words to be sorted.

■ Operation example

Operation of instruction format description program

- Ascending order

  If data is stored in DT10 to DT19 as shown below and [S3] = K0, the following operation is performed.

DT10:	K300	⇨	DT10:	K-30
11:	K10		11:	K-3
12:	K3		12:	K-1
13:	K-1		13:	K1
14:	K1000		14:	K3
15:	K-30		15:	K10
16:	K100		16:	K30
17:	K30		17:	K1000
18:	K1		18:	K300
19:	K-3		19:	K1000

- Descending order

  If data is stored in DT10 to DT19 as shown below and [S3] = K1, the following operation is performed.

DT10:	K300	⇨	DT10:	K1000
11:	K10		11:	K300
12:	K3		12:	K100
13:	K-1		13:	K30
14:	K1000		14:	K10
15:	K-30		15:	K3
16:	K100		16:	K1
17:	K30		17:	K-1
18:	K1		18:	K-3
19:	K-3		19:	K-30

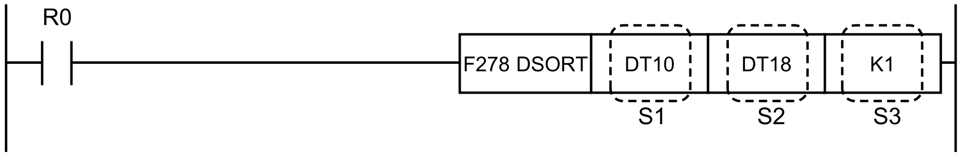
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	ON when [S1] is greater than [S2]
	Turns ON when S1 and S2 are different devices

25.8 F278 DSORT (32-bit Data Block Sort)

Sorts strings (double word data) in the specified memory area in ascending or descending order.

■ Instruction format



■ Operands

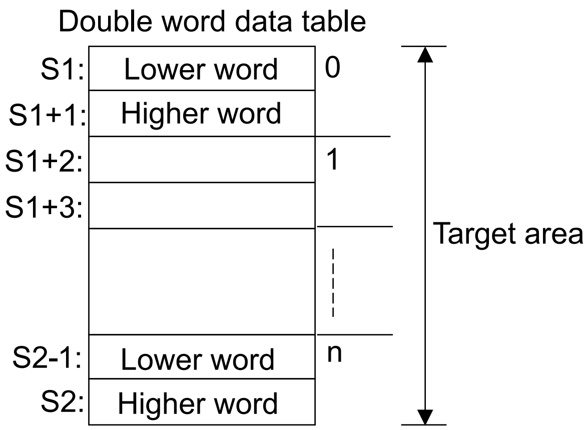
Items	Settings
S1	Starting area storing sort data
S2	Ending area storing sort data
S3	Area storing sort conditions, or constant data

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1		•	•	•	•	•	•	•	•							•	
S2		•	•	•	•	•	•	•	•							•	
S3	•	•	•	•	•	•	•	•	•	•	•	•	•			•	

■ Outline of operation

- Sorts the double word data (signed) in the areas specified by [S1] and [S2] into ascending or descending order.
- When S1 = S2, no operation takes place.
- The sort conditions are specified in [S3].
  K0: Ascending order
  K1: Descending order
- During sorting, the data from [S1] to [S2] is sorted in sequential order in accordance with the sort procedure. Note that the number of times data is compared increases proportionally to the square of the number of data words, and therefore the operation time will increase if there is a large number of data words to be sorted.
- If [S2] specifies a high word of double word data, processing will take place over the same area as if a low word had been specified.



■ Operation example

Operation of instruction format description program

- Ascending order

  If data is stored in DT10 to DT19 as below and [S3] = K0, the following operation will be performed.

	Before		After
DT10, 11:	K25000	DT10, 11:	K-4000
12, 13:	K-4000	12, 13:	K-2600
14, 15:	K1500	14, 15:	K1500
16, 17:	K-2600	16, 17:	K25000
18, 19:	K100000	18, 19:	K100000

- Descending order

  If data is stored in DT10 to DT19 as shown below and [S3] = K1, the following operation is performed.

	Before		After
DT10, 11:	K25000	DT10, 11:	K100000
12, 13:	K-4000	12, 13:	K25000
14, 15:	K1500	14, 15:	K1500
16, 17:	K-2600	16, 17:	K-2600
18, 19:	K100000	18, 19:	K-4000

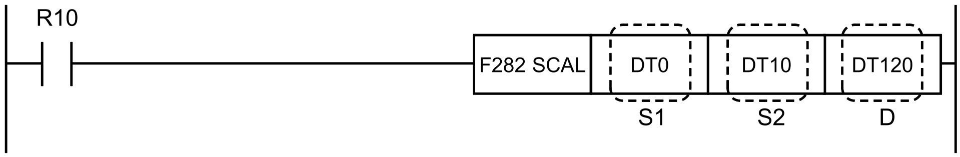
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	ON when [S1] is greater than [S2]
	Turns ON when S1 and S2 are different devices

25.9 F282 SCAL (16-bit Data Linearization)

Performs scaling of the given data table and finds output value Y with regards to input value X.

■ Instruction format



■ Operands

Items	Settings
S1	Source 16-bit data equivalent to input value X, or the area where it is stored
S2	Starting address of the data table used for scaling (linearization)
D	Area where output result Y is stored

■ Devices that can be specified (indicated by •)

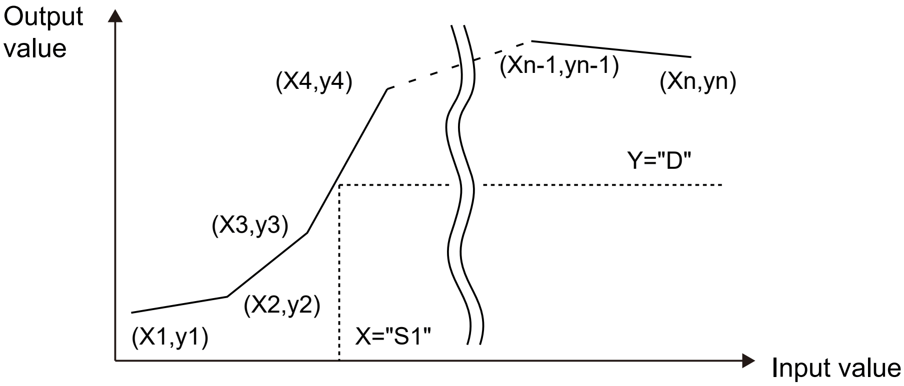
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•					•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The 16-bit data specified by [S1] is scaled in accordance with the data table specified by [S2], and the output value for input value X is calculated.
- The number of items in the data table n is determined by the value [n] specified in [S2] at the top of the data table.

Structure of the data table used in scaling (linearization) (if S2 = DT10 and n = K10)

S2:	n	DT10
S2+1:	x1	DT11
S2+2:	x2	DT12
S2+3:	x3	DT13
S2+n-1:	xn-1	DT19
S2+n:	xn	DT20
S2+n+1:	y1	DT21
S2+n+2:	y2	DT22
S2+n+3:	y3	DT23
S2+2n-1:	yn-1	DT29
S2+2n:	yn	DT30



■ Operation example

Operation of instruction format description program

The data table is referenced starting from DT10, output value Y for the input value stored in DT0 is calculated, and the result is stored in DT120.

■ Precautions for programming

- Make $X_{t\_1}<X_t$.
- Create xt and yt as signed 16-bit data.
- If X(S1) < x1, then Y(D) = y1.
- If X(S1) > xn, then Y(D) = yn. n has a maximum of 99.

■ Flag operations

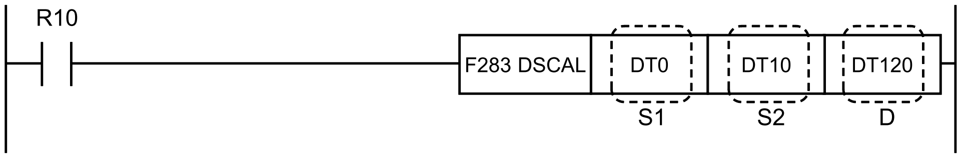
Name	Description
R9007	Turns ON when the area is exceeded in index modification.

Name	Description
R9008 (ER)	Turns ON when n < 2 or n > 99 in [S2]
	Turns ON when data table in [S2] exceeds area
	Turns ON when Xn is not in ascending order

25.10 F283 DSCAL (32-bit Data Linearization)

Performs scaling of the given data table and finds output value Y with regards to input value X.

■ Instruction format



■ Operands

Items	Settings
S1	Original 32-bit data corresponding to input value X, or storage area
S2	Starting address of the data table used for scaling (linearization)
D	Area where output result Y is stored

■ Devices that can be specified (indicated by ●)

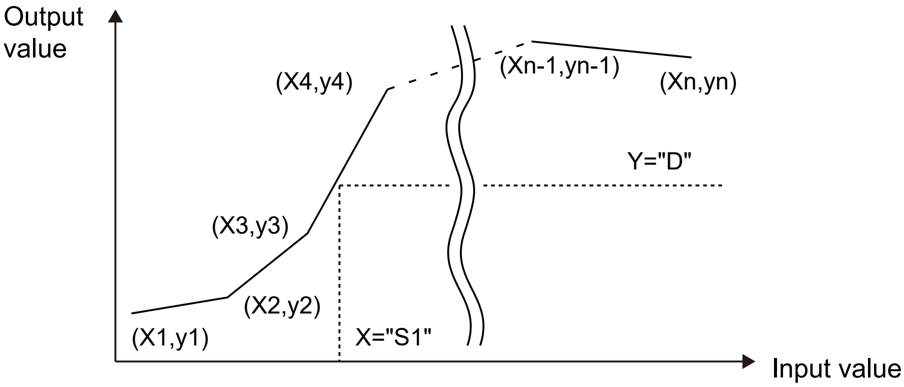
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●					●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- Performs scaling of the 32-bit data specified in [S1] according to the data table specified in [S2], and finds output value Y with regards to input value X.
- The number of items in the data table n is determined by the value [n] specified in [S2] at the top of the data table.

Structure of the data table used in scaling (linearization) (if S2 = DT10 and n = K10)

S2:	n	DT10
S2+1:	x1	DT11
S2+2:		DT12
S2+3:	x2	DT13
S2+4:		DT14
S2+2n-1:	xn	DT29
S2+2n:		DT30
S2+2n+1:	y1	DT31
S2+2n+2:		DT32
S2+2n+3:	y2	DT33
S2+2n+4:		DT34
S2+4n-1:	yn	DT49
S2+4n:		DT50



■ Operation example

Operation of instruction format description program

Finds output value Y with regards to input value X stored in DT0, with reference to the data table starting from DT10, and stores the result in DT120 to DT121.

■ Precautions for programming

- Make $X_{t\_1}<X_t$.
- Create xt and yt as signed 32-bit data.
- If X(S1) < x1, then Y(D) = y1.
- If X(S1) > xn, then Y(D) = yn. n has a maximum of 99.

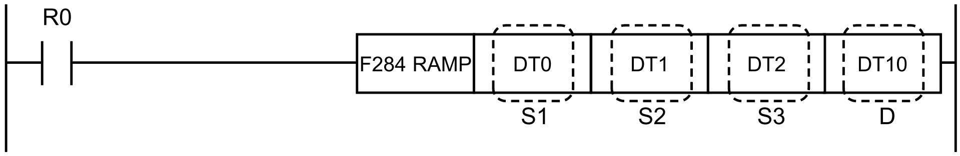
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when n<2 or n>99 in [S2]
	Turns ON when data table in [S2] exceeds area
	Turns ON when Xn is not in ascending order

25.11 F284 RAMP (16-bit Data Ramp Output)

Linear output is executed based on the elapsed time from the start of execution, by performing scaling from the output default value, target value, and time width.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the default value, or constant data
S2	Area storing the target value, or constant data
S3	Area storing the time width, or constant data
D	Data output area

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S3	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

Scaling is performed from the 16-bit output default value of the area specified by [S1], the 16-bit output target value of the area specified by [S2], and the 16-bit output time width (in ms units) of the area specified by [S3], and linear output is performed according to the elapsed time from the start of execution.

■ Precautions for programming

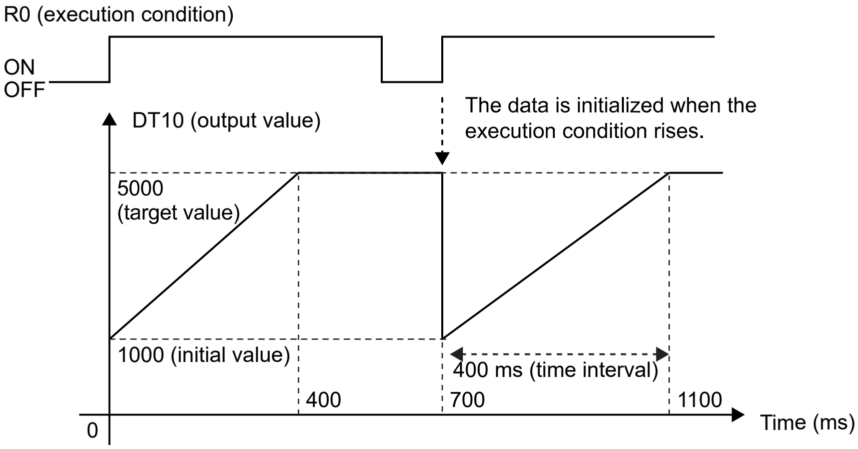
It is possible that a maximum error of 1 scan may occur in the output time width.

<Example> If the following values are set in a program

DT0 (default value) = K1000

DT1 (target value) = K5000

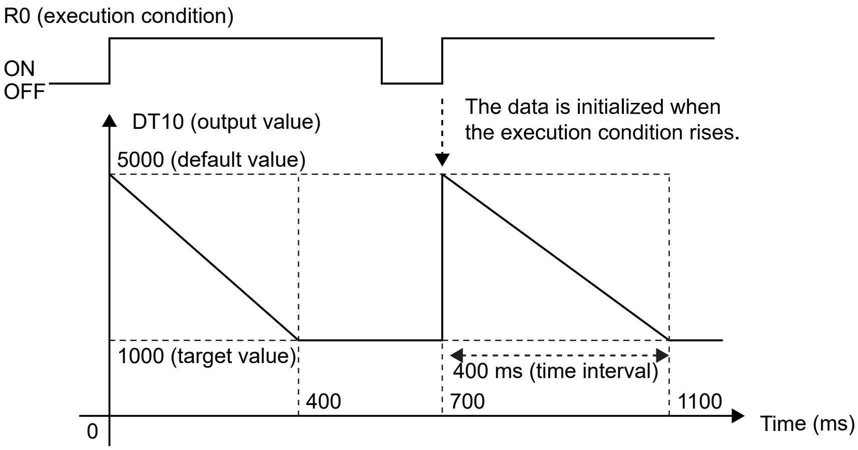
DT2 (time width) = K400



DT0 (default value) = K5000

DT1 (target value) = K1000

DT2 (time width) = K400



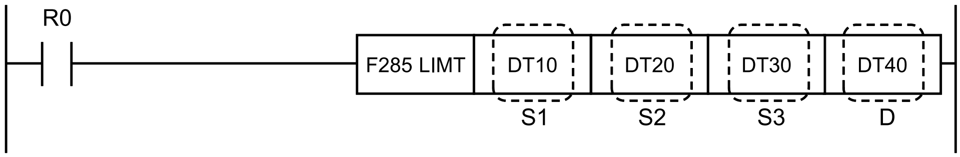
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	If the output time width specified in S3 is out of range of K1 to K30000

25.12 F285 LIMT (16-bit Data Upper and Lower Limit Control)

Performs upper and lower limit control (word data).

■ Instruction format



■ Operands

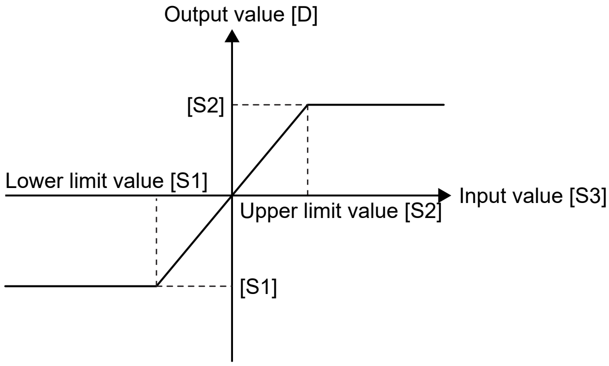
Items	Settings
S1	Area storing the lower limit or lower limit data
S2	Area storing the upper limit or upper limit data
S3	Area storing the input value or input value data
D	Area storing the output value

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S3	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The output value (word data) stored in the area specified by [D] is controlled according to whether or not the input value (word data) specified by [S3] falls within the range bounded by the upper and lower limits specified by [S1] and [S2].
- The output value is determined based on the following conditions.
  - If lower limit value [S1] is greater than input value [S3], then lower limit value [S1] becomes output value [D]
  - If upper limit value [S2] is less than input value [S3], then upper limit value [S2] becomes output value [D]
  - If lower limit value [S1] is equal to or less than input value [S3], which is equal to or less than upper limit value [S2], then input value [S3] becomes output value [D]



- For control using only the upper limit value
  Specify K-32768 (or H8000) for the lower limit value [S1].
- For control using only the lower limit value
  Specify K32767 (or H7FFF) for the upper limit value [S2].

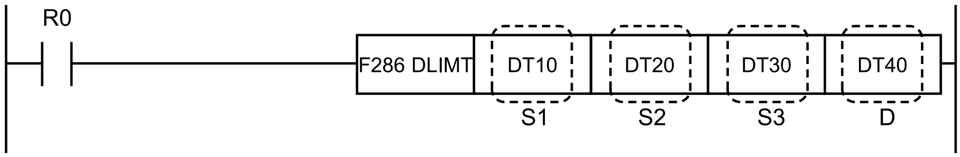
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification. ON when [S1] is greater than [S2]
R900B (=)	Turns ON when the operation result falls within the upper/lower limit range

25.13 F286 DLIMT (32-bit Data Upper and Lower Limit Control)

Performs upper and lower limit control (double word).

■ Instruction format



■ Operands

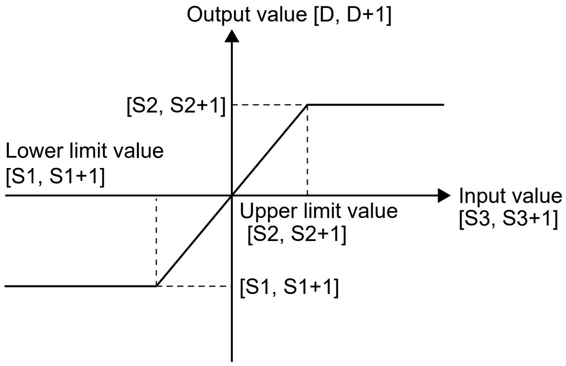
Items	Settings
S1	Area storing the lower limit, or lower limit data (two words)
S2	Area storing the upper limit, or upper limit data (two words)
S3	Area storing the input value, or input value data (two words)
D	Area storing the output value (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S3	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The output value (double-word data) stored in the area specified by [D] is controlled according to whether or not the input value (double-word data) specified by [S3] falls within the range bounded by the upper and lower limits specified by [S1] and [S2].
- The output value is determined based on the following conditions.
  - If lower limit value [S1, S1+1] is greater than input value [S3, S3+1], then lower limit value [S1, S1+1] becomes output value [D, D+1]
  - If upper limit value [S2, S2+1] is less than input value [S3, S3+1], then upper limit value [S2, S2+1] becomes output value [D, D+1]
  - If lower limit value [S1, S1+1] is equal to or less than input value [S3, S3+1], which is equal to or less than upper limit value [S2, S2+1], then input value [S3, S3+1] becomes output value [D, D+1]



- For control using only the upper limit value
  Set K-2147483648 (or H80000000) for lower limit [S1, S1+1].
- For control using only the lower limit value
  Set K2147483647 (or H7FFFFFFF) for upper limit [S2, S2+1].

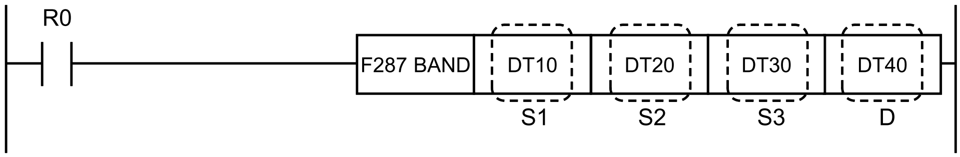
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	ON when [S1] is greater than [S2]
R900B (=)	Turns ON when the operation result falls within the upper/lower limit range

25.14 F287 BAND (16-bit Data Deadband Control)

Performs deadband control (word).

■ Instruction format



■ Operands

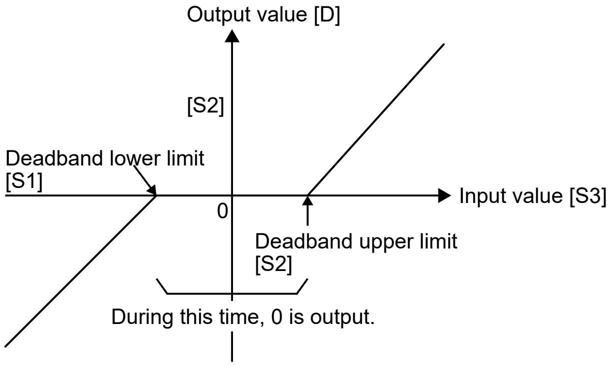
Items	Settings
S1	Area storing the lower limit or lower limit data
S2	Area storing the upper limit or upper limit data
S3	Area storing the input value or input value data
D	Area storing the output value

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S3	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The output value (word data) stored in the area specified by [D] is controlled based in whether or not the input value (word data) specified by [S3] is inside or outside of the deadband bounded by the upper and lower limits specified by [S1] and [S2].
- The output value is determined based on the following conditions.
  - When the lower limit [S1] is greater than the input value [S3], input value [S3] minus lower limit [S1] equals output value [D]
  - When the upper limit [S2] is less than the input value [S3], input value [S3] minus upper limit [S2] equals output value [D]
  - When the lower limit [S1] is equal to or less than the input value [S3] that is equal to or less than the upper limit [S2], 0 equals output value [D]



■ Operation example

Operation of instruction format description program

When K-100 is stored in DT10 and K100 in DT20, the following operation will be performed.

Value of DT30	Value stored in DT40
K-300	K-200
K-200	K-100
K-100 to K100	K0
K200	K100
K300	K200

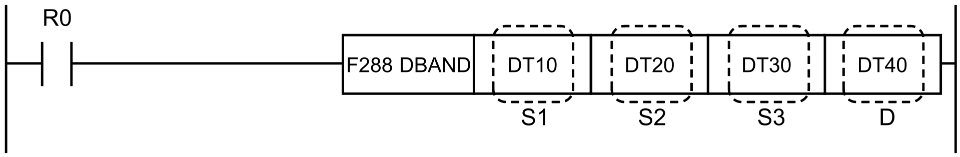
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	ON when [S1] is greater than [S2]
R9009 (CY)	ON when the calculation result overflows or underflows
R900B (=)	Turns ON when the calculation result is"0"

25.15 F288 DBAND (32-bit Data Deadband Control)

Carries out deadband control (double word).

■ Instruction format



■ Operands

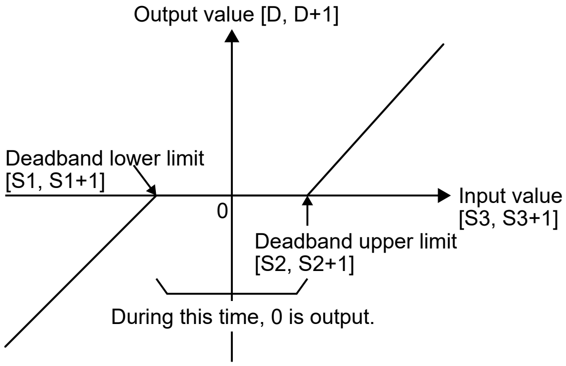
Items	Settings
S1	Area storing the lower limit, or lower limit data (two words)
S2	Area storing the upper limit, or upper limit data (two words)
S3	Area storing the input value, or input value data (two words)
D	Area storing the output value (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S3	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The output value (double word data) stored in the area specified by [D] is controlled according to whether or not the input value (double word data) specified by [S3] is inside the range of the upper and lower limits of the deadband specified by [S1] and [S2].
- The output value is determined based on the following conditions.
  - When the lower limit [S1, S1+1] > input value [S3, S3+1], the input value [S3, S3+1] - the lower limit [S1, S1+1] becomes the output value [D, D+1]
  - When the upper limit [S2, S2+1] < input value [S3, S3+1], the input value [S3, S3+1] - the upper limit [S2, S2+1] becomes the output value [D, D+1]
  - When the lower limit [S1, S1+1] ≤ input value [S3, S3+1] ≤ the upper limit [S2, S2+1], 0 becomes the output value [D, D+1]



■ Operation example

Operation of instruction format description program

If K-10000 is stored in DT10 and DT11, and K10000 is stored in DT20 and DT21, the following operation is performed.

Values of DT30, and DT31	Values stored in DT40 and DT41
K-30000	K-20000
K-20000	K-10000
K-10000 to K10000	K0
K20000	K10000
K30000	K20000

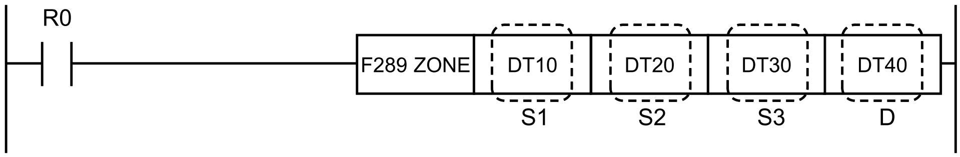
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	ON when [S1] is greater than [S2]
R9009 (CY)	ON when the calculation result overflows or underflows
R900B (=)	Turns ON when the calculation result is"0"

25.16 F289 ZONE (16-bit Data Zone Control)

Performs zone control (word).

■ Instruction format



■ Operands

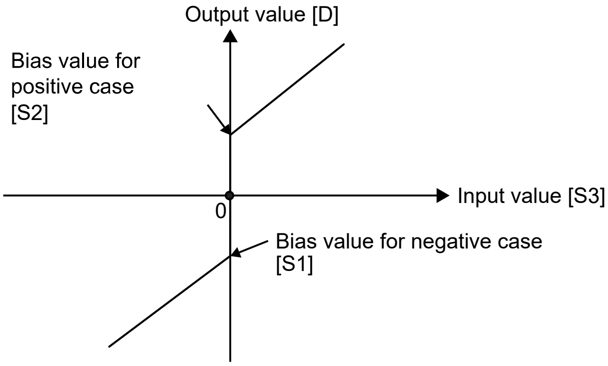
Items	Settings
S1	Area where negative bias value is stored, or negative bias value data
S2	Area where positive bias value is stored, or positive bias value data
S3	Area storing the input value or input value data
D	Area storing the output value

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S2	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
S3	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

- The bias value specified by [S1] or [S2] is added to the input value (word data) specified by [S3], and the output value is stored in the area specified by [D].
- The output value is determined based on the following conditions.
  - When input value [S3] < 0, input value [S3] + negative bias value [S1] → output value [D]
  - When input value [S3] = 0, 0 → output value [D]
  - When input value [S3] > 0, input value [S3] + positive bias value [S2] → output value [D]



Operation example

Operation of instruction format description program

When K-100 is stored in DT10, and K100 is stored in DT20

Value of DT30	Value stored in DT40
K-300	K-400
K-200	K-300
K-100	K-200
K0	K0
K100	K200
K200	K300
K300	K400

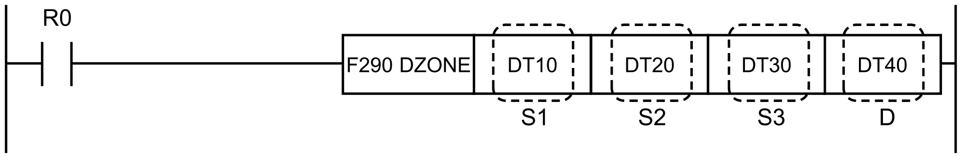
Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	ON when the calculation result overflows or underflows
R900B (=)	Turns ON when the input value is"0"

25.17 F290 DZONE (32-bit Data Zone Control)

Carries out zone control (double word).

■ Instruction format



■ Operands

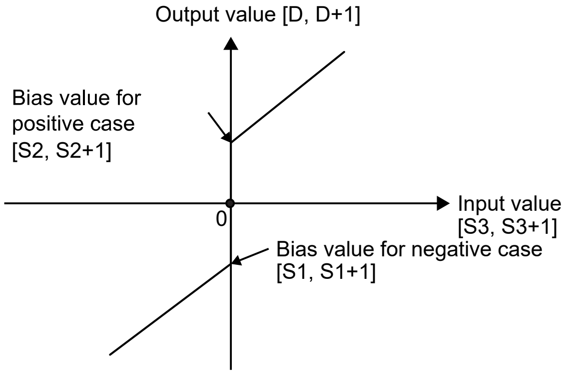
Items	Settings
S1	Area storing negative bias values, or negative bias value data (two words)
S2	Area storing positive bias values, or positive bias value data (two words)
S3	Area storing the input value, or input value data (two words)
D	Area storing the output value (two words)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S2	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
S3	●	●	●	●	●	●	●	●	●	●	●	●	●			●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The bias value specified by [S1] or [S2] is added to the input value (double-word data) specified by [S3], and stored in the area specified by [D].
- The output value is determined based on the following conditions.
  - When the input value [S3, S3+1] is less than 0, the input value [S3, S3+1] + the negative bias value [S1, S1+1] is the output value [D, D+1]
  - When the input values [S3, S3+1] equal zero, zero is stored in [D, D+1] as the output values
  - When the input values [S3, S3+1] are greater than zero, the input values [S3, S3+1] plus the positive bias values [S2, S2+1] are stored in [D, D+1] as the output values



■ Operation example

Operation of instruction format description program

If K-10000 is stored in DT10 and DT11, and K10000 is stored in DT20 and DT21, the following operation is performed.

Values of DT30, and DT31	Values stored in DT40 and DT41
K-30000	K-40000
K-20000	K-30000
K-10000	K-20000
K0	K0
K10000	K20000
K20000	K30000
K30000	K40000

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R9009 (CY)	ON when the calculation result overflows or underflows
R900B (=)	Turns ON when the input value is"0"

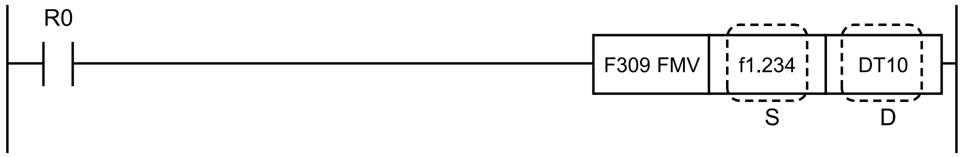
(MEMO)

26 Floating-point Instruction

26.1 F309 FMV (Floating Point Data Move)

Transfers the specified real number data to the specified area.

■ Instruction format



■ Operands

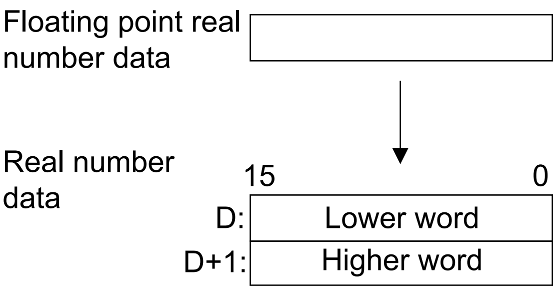
Items	Settings
S	Transfer data: Area storing real number data (32-bit), or constant data
D	Destination: Data transfer destination area

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S															●		
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- The floating-point type real number data (32-bit) specified by [S] is transferred to the memory area specified by [D].
  Specify a lower 16-bit memory area for the memory area.



- The range of constants that can be specified in [S] is as follows.
  Positive numbers f 0.0000001 to f 9999999
  Negative numbers f -9999999 to f -0.000001

■ Operation example

Operation of instruction format description program

When the execution condition R0 is ON, the floating-point type constant value f 1.234 is transferred to data registers DT10 to DT11.

DT10:	(f1.234)
DT11:	

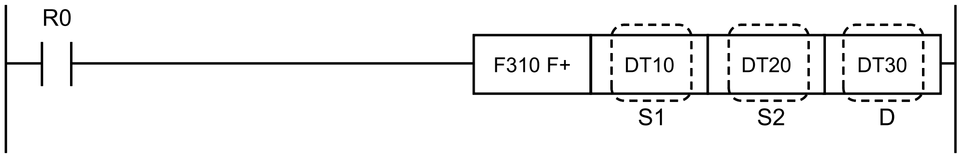
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

26.2 F310 F+ (Floating Point Data Addition)

Adds real number data.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing augend data, or augend data (two words)
S2	Area storing addend data, or addend data (two words)
D	Area storing the addition result (two words)

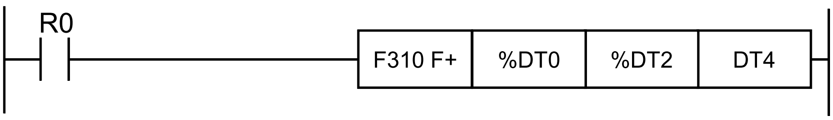
■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
D		●	●	●	●	●	●	●	●							●	●

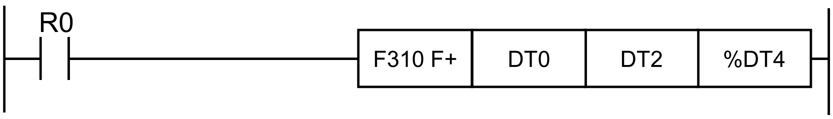
■ Outline of operation

- The real number data specified by [S1, S1+1] and [S2, S2+1] is added, and the result is stored in [D, D+1].

  [S1, S1+1] + [S2, S2+1] → [D, D+1]
- If [S1] and [S2] are specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



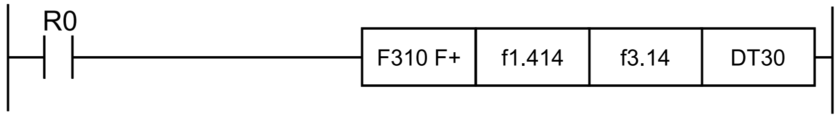
- If [D] is specified with an integer device, the real number is converted to integer data and stored.



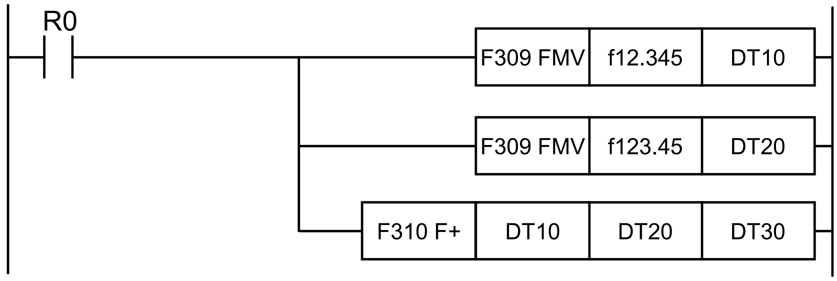
- If a K constant is specified for [S1] or [S2], the same process is performed as if an integer device was specified.

■ Program example

- When R0 is turned ON, f 4.554 is stored in DT30 and DT31.



- When R0 is turned ON, f 135.795 is stored in DT30 and DT31.



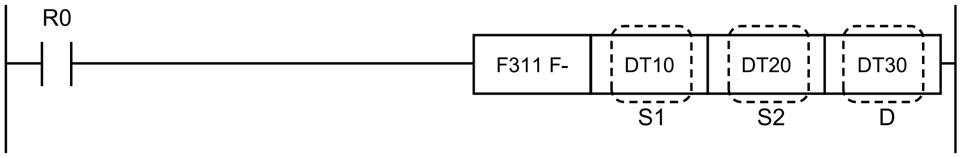
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real-number data is specified in [S1, S1+1] or [S2, S2+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows

26.3 F311 F- (Floating Point Data Subtraction)

Subtracts real number data.

■ Instruction format



■ Operands

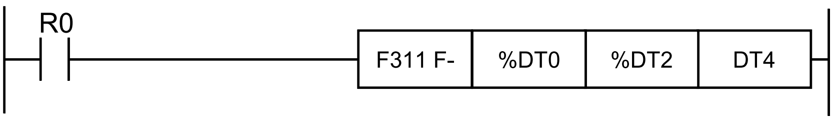
Items	Settings
S1	Area storing the minuend data, or the minuend data (two words)
S2	Area storing the subtrahend data, or the subtrahend data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by ●)

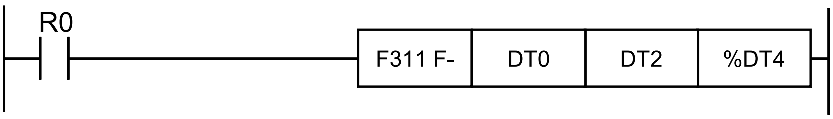
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
D		●	●	●	●	●	●	●	●							●	●

■ Outline of operation

- The subtrahend data specified by [S2, S2+1] is subtracted from the minuend data specified by [S1, S1+1], and the result is stored in [D, D+1].
  [S1, S1+1] - [S2, S2+1] → [D, D+1]
- If [S1] and [S2] are specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



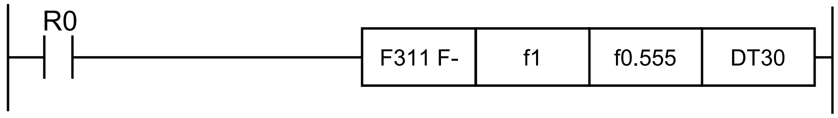
- If [D] is specified with an integer device, the real number is converted to integer data and stored.



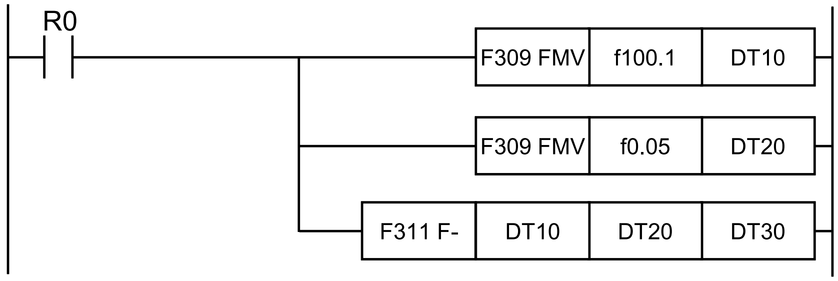
- If a K constant is specified for [S1] or [S2], the same process is performed as if an integer device was specified.

■ Program example

- When R0 turns ON, f 0.445 is stored in DT30 and DT31.



- When R0 turns ON, f 100.05 is stored in DT30 and DT31.



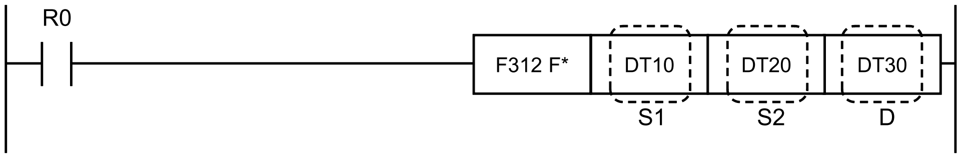
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real-number data is specified in [S1, S1+1] or [S2, S2+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows

26.4 F312 F* (Floating Point Data Multiplication)

Multiplies real number data items.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the multiplicand data, or the multiplicand data (two words)
S2	Area storing the multiplier data, or the multiplier data (two words)
D	Area storing the operation results (two words)

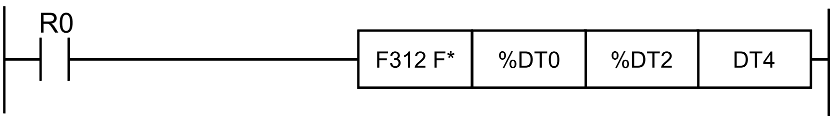
■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
D		●	●	●	●	●	●	●	●							●	●

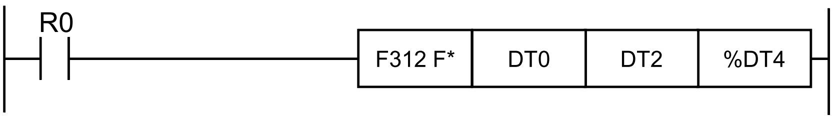
■ Outline of operation

- Multiplies the multiplicand data specified by [S1, S1+1] and the multiplier data specified by [S2, S2+1], and stores the result in [D, D+1].

  [S1, S1+1] × [S2, S2+1] → [D, D+1]
- If [S1] and [S2] are specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



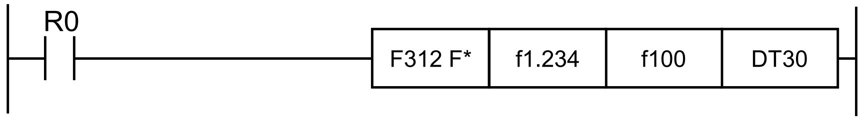
- If [D] is specified with an integer device, the real number is converted to integer data and stored.



- If a K constant is specified for [S1] or [S2], the same process is performed as if an integer device was specified.

■ Program example

The f123.4000 is stored to DT30 and DT31 when the R0 turns ON.



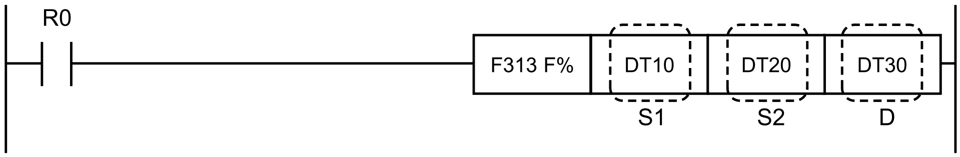
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real-number data is specified in [S1, S1+1] or [S2, S2+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows

26.5 F313 F% (Floating Point Data Division)

Divides real number data.

■ Instruction format



■ Operands

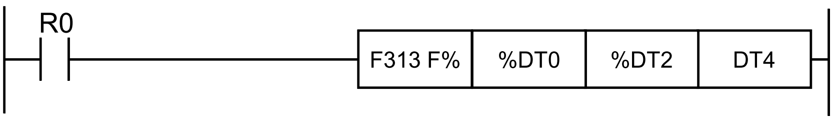
Items	Settings
S1	Area storing the dividend data, or dividend data (two words)
S2	Area storing the divisor data, or divisor data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by ●)

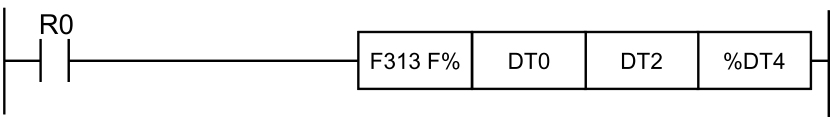
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
D		●	●	●	●	●	●	●	●							●	●

■ Outline of operation

- Divides the dividend data specified by [S1, S1+1] by the divisor data specified by [S2, S2+1], and stores the result in [D, D+1].
  [S1, S1+1]÷[S2, S2+1] → [D, D+1]
- If [S1] and [S2] are specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



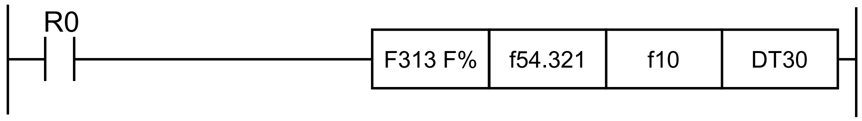
- If [D] is specified with an integer device, the real number is converted to integer data and stored.



- If a K constant is specified for [S1] or [S2], the same process is performed as if an integer device was specified.

■ Program example

When R0 turns ON, f5.432100 is stored to DT30 to DT31.



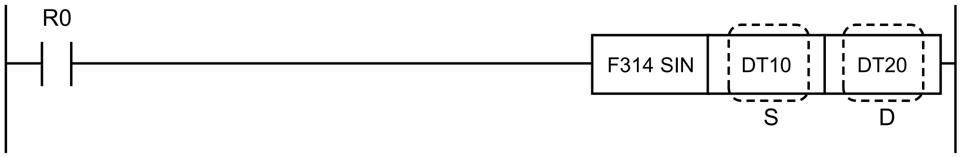
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real-number data is specified in [S1, S1+1] or [S2, S2+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
	Turns ON when divided by 0.0
R9009 (CY)	Turns ON when operation result overflows

26.6 F314 SIN (Floating Point Data Sine Operation)

Calculates the trigonometric function sin().

■ Instruction format



■ Operands

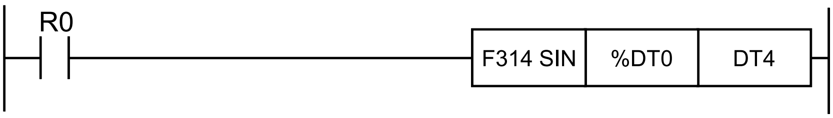
Items	Settings
S	Area storing angle data, or angle data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by ●)

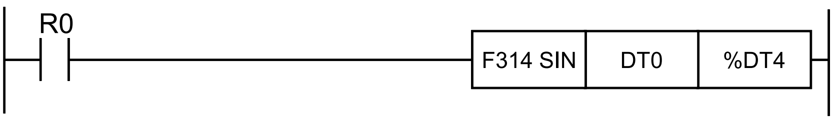
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
D		●	●	●	●	●	●	●	●							●	●

■ Outline of operation

- sin([S, S+1]) of the angle data specified by [S, S+1] (unit: radian) is calculated, and the result is stored in [D, D+1].
  sin([S, S+1]) → [D, D+1]
- If [S] is specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



- If [D] is specified with an integer device, the real number is converted to integer data and stored.



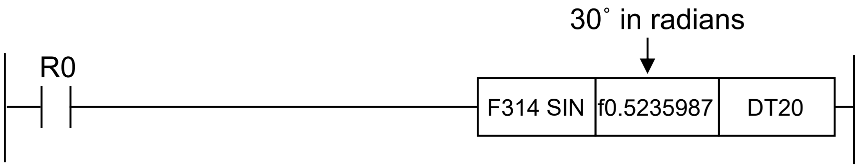
- If a K constant is specified for [S], the same process is performed as if an integer device was specified.

■ Precautions for programming

The accuracy decreases as the absolute value of the input value increases. Where possible, use angle data within the range -2π radians ≤ input ≤ 2π radians.

■ Program example

When R0 turns ON, f0.4999999 is stored in DT20 and DT21.



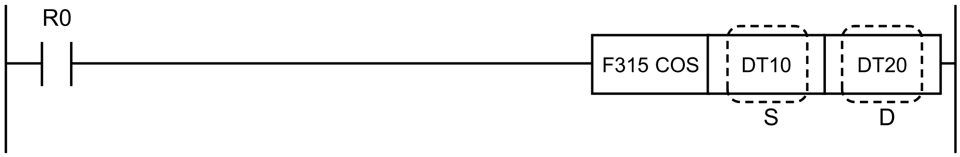
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
	Turns ON when the absolute value of the input value is 52707176 or higher
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.7 F315 COS (Floating Point Data Cosine Operation)

Operates the trigonometric function cos().

■ Instruction format



■ Operands

Items	Settings
S	Area storing angle data, or angle data (two words)
D	Area storing the operation results (two words)

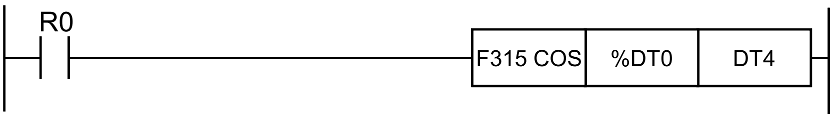
■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
D		●	●	●	●	●	●	●	●							●	●

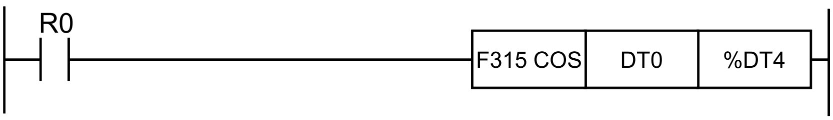
■ Outline of operation

- Calculates cos ([S, S+1]) of angle data (unit: radians) specified in [S, S+1], and stores the result in [D, D+1].

  cos([S, S+1]) → [D, D+1]
- If [S] is specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



- If [D] is specified with an integer device, the real number is converted to integer data and stored.



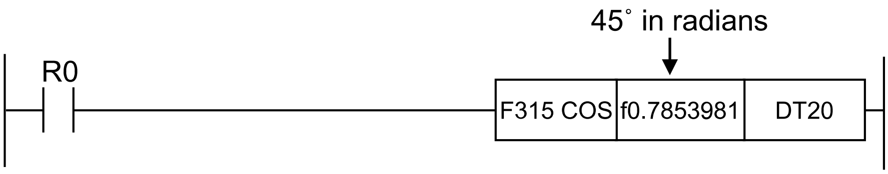
- If a K constant is specified for [S], the same process is performed as if an integer device was specified.

■ Precautions for programming

The accuracy decreases as the absolute value of the input value increases. Where possible, use angle data within the range -2π radians ≤ input ≤ 2π radians.

■ Program example

When R0 is ON, f 0.7071068 is stored in DT20 to DT21.



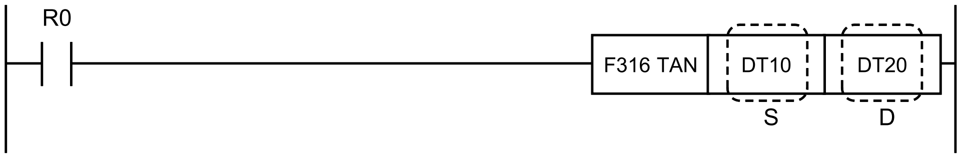
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
	Turns ON when the absolute value of the input value is 52707176 or higher
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.8 F316 TAN (Floating Point Data Tangent Operation)

Calculates the trigonometrical function tan().

■ Instruction format



■ Operands

Items	Settings
S	Area storing angle data, or angle data (two words)
D	Area storing the operation results (two words)

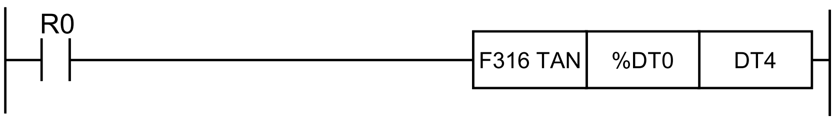
■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
D		●	●	●	●	●	●	●	●							●	●

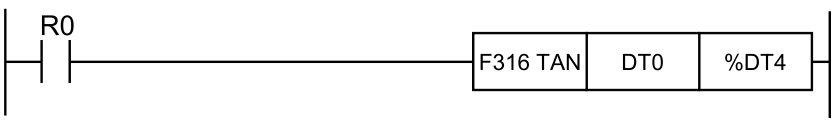
■ Outline of operation

- The tan([S and S+1]) of angle data (unit: radians) specified by S and S+1 is calculated and the result stored in D and D+1.

  tan([S, S+1]) -> [D, D+1]
- If [S] is specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



- If [D] is specified with an integer device, the real number is converted to integer data and stored.



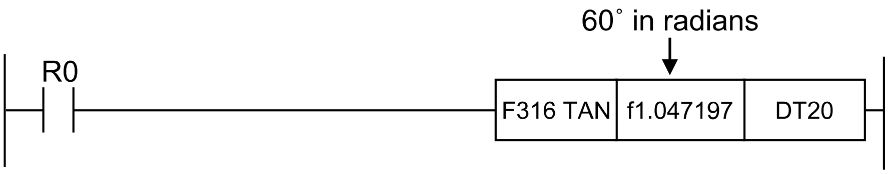
- If a K constant is specified for [S], the same process is performed as if an integer device was specified.

■ Precautions for programming

The accuracy decreases as the absolute value of the input value increases. Where possible, use angle data within the range -2π radians ≤ input ≤ 2π radians.

■ Program example

f 1.732048 is stored in DT20 and DT21 when R0 turns ON.



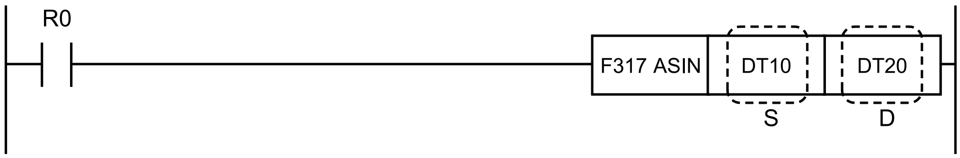
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
	Turns ON when the absolute value of the input value is 52707176 or higher
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.9 F317 ASIN (Floating Point Data Arcsine Operation)

Calculates the trigonometric function $SIN^{-1}()$.

■ Instruction format



■ Operands

Items	Settings
S	Area storing angle data, or angle data (two words)
D	Area storing the operation results (two words)

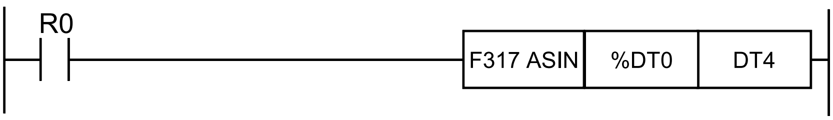
■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
D		•	•	•	•	•	•	•	•							•	•

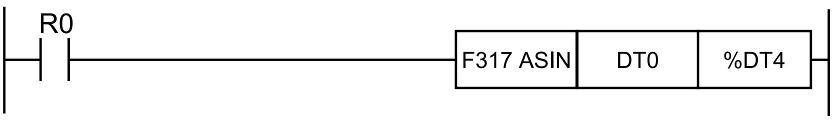
■ Outline of operation

- Calculates an angle from the SIN value specified in [S, S+1] and stores the result in [D, D+1] (in radians).

  SIN^{-1}([S, S+1])→[D, D+1]
- If [S] is specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



- If [D] is specified with an integer device, the real number is converted to integer data and stored.



- If a K constant is specified for [S], the same process is performed as if an integer device was specified.

26.9 F317 ASIN (Floating Point Data Arcsine Operation)

■ Precautions for programming

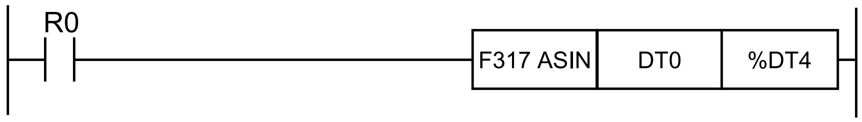
[D, D+1] is stored in the following range:

$-\pi/2 \leq$ [D, D+1] $\leq \pi/2$

[radians]   [radians]

■ Program example

f0.5235986 (30° radians) is stored in DT20 to DT21 when R0 turns ON.



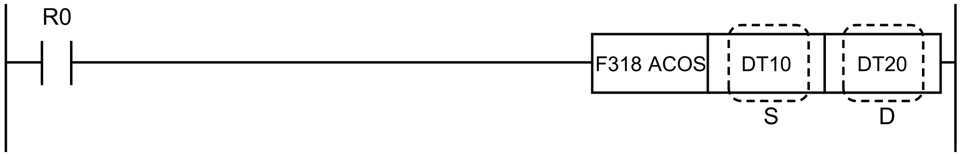
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when [S, S+1] is not within the range -1.0 ≤ [S, S+1] ≤ 1.0
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.10 F318 ACOS (Floating Point Data Arccosine Operation)

Calculates the trigonometric function $COS^{-1}()$.

■ Instruction format



■ Operands

Items	Settings
S	Area storing angle data, or angle data (two words)
D	Area storing the operation results (two words)

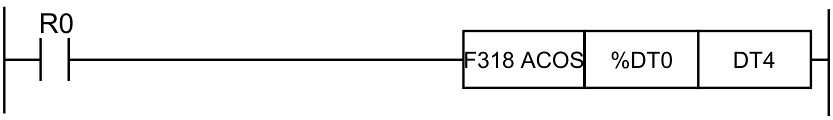
■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
D		•	•	•	•	•	•	•	•							•	•

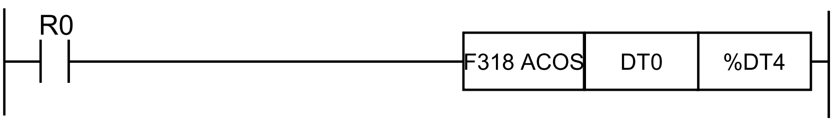
■ Outline of operation

- The angle from the COS value specified by [S, S+1] is calculated and the result (unit: radian) is stored in [D, D+1].

  $COS^{-1}([S, S+1]) \rightarrow [D, D+1]$
- If [S] is specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



- If [D] is specified with an integer device, the real number is converted to integer data and stored.



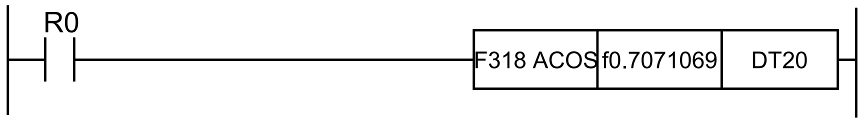
- If a K constant is specified for [S], the same process is performed as if an integer device was specified.

■ Precautions for programming

[D, D+1] is stored in the following range:
0.0 ≤ [D, D+1] ≤ π
[radians] [radians]

■ Program example

When R0 turns ON, f0.7853980 (45° in radians) is stored in DT20 and DT21.



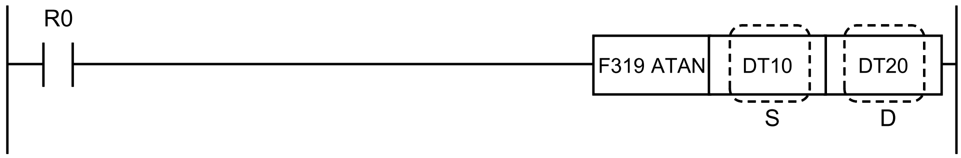
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when [S, S+1] is not -1.0 ≤ [S, S+1] ≤ 1.0
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.11 F319 ATAN (Floating Point Data Arctangent Operation)

Calculates the trigonometrical function $TAN^{-1}()$.

■ Instruction format



■ Operands

Items	Settings
S	Area storing angle data, or angle data (two words)
D	Area storing the operation results (two words)

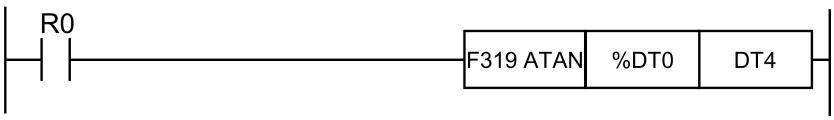
■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
D		•	•	•	•	•	•	•	•							•	•

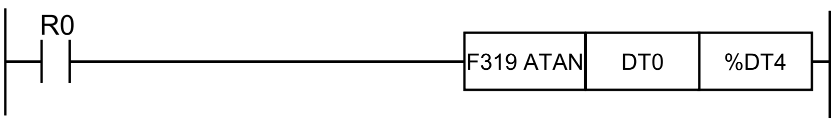
■ Outline of operation

- The angle from the TAN value specified by [S, S+1] is calculated and the result (unit: radian) is stored in [D, D+1].

  TAN^{-1}([S, S+1]) → [D, D+1]
- If [S] is specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



- If [D] is specified with an integer device, the real number is converted to integer data and stored.



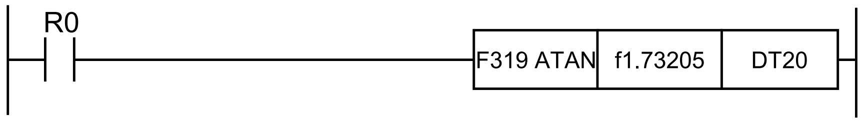
- If a K constant is specified for [S], the same process is performed as if an integer device was specified.

■ Precautions for programming

[D, D+1] is stored in the following range:
-π/2 < [D, D+1] < π/2
[radians] [radians]

■ Program example

f1.047197 (60° in radians) is stored in DT20 to DT21 when R0 turns ON.



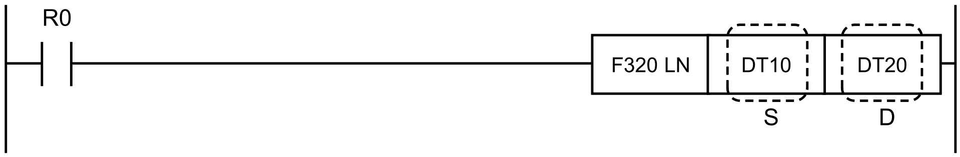
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification. Turns ON when non-real number data is specified in [S, S+1] Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.12 F320 LN (Floating Point Data Natural Logarithmic Operation)

Calculates the natural logarithm LN().

■ Instruction format



■ Operands

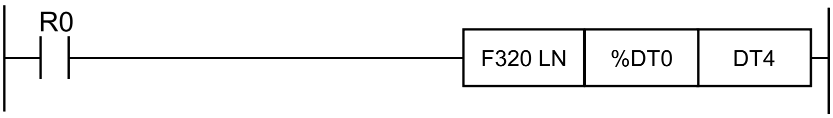
Items	Settings
S	Area storing angle data, or angle data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by ●)

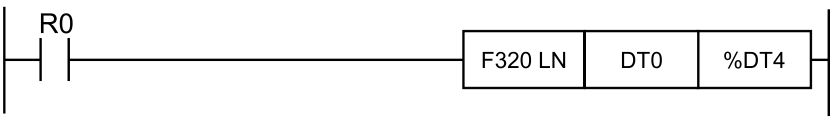
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
D		●	●	●	●	●	●	●	●							●	●

■ Outline of operation

- The natural logarithm LN ([S, S+1]) is calculated from the operation data specified by [S, S +1], and the result is stored in [D, D+1].
  LN([S, S+1]) → [D, D+1]
- If [S] is specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.

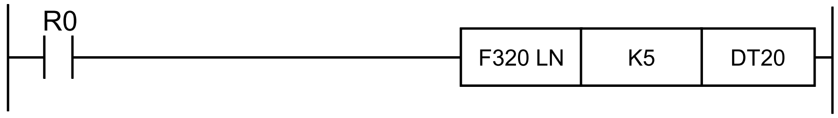


- If [D] is specified with an integer device, the real number is converted to integer data and stored.

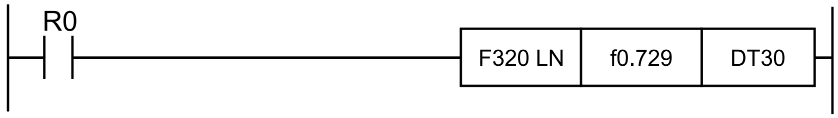


■ Program example

- When R0 turns ON, f1.6094379 is stored in DT20 and DT21.



- When R0 turns ON, f-0.3160815 is stored in DT30 and DT31.



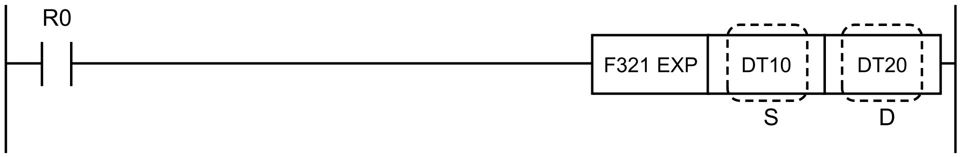
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when [S, S+1] is not 0 < [S, S+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.13 F321 EXP (Floating Point Data Exponent Operation)

Calculates the exponent EXP().

■ Instruction format



■ Operands

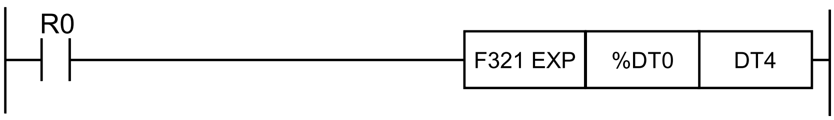
Items	Settings
S	Area storing angle data, or angle data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by ●)

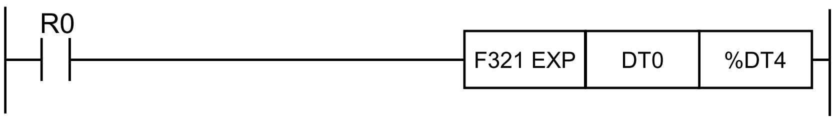
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
D		●	●	●	●	●	●	●	●							●	●

■ Outline of operation

- The exponent EXP ([S, S+1]) is calculated from the operation data specified by [S, S+1], and the result is stored in [D, D+1].
  EXP([S, S+1]) → [D, D+1]
  The calculation is performed with exponent base (e) equal to"2.718282".
- If [S] is specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.

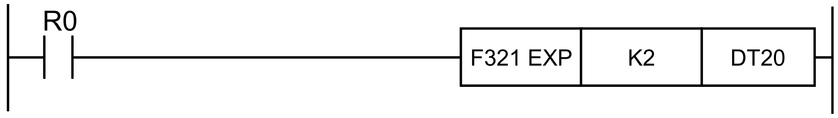


- If [D] is specified with an integer device, the real number is converted to integer data and stored.

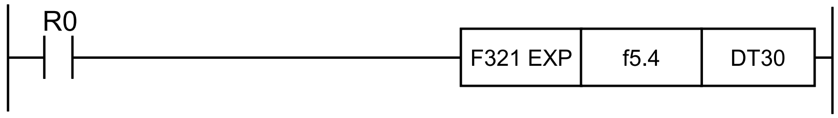


■ Program example

- When R0 turns ON, f7.389056 is stored in DT20 and DT21.



- When R0 turns ON, f221.406402 is stored in DT30 and DT31.



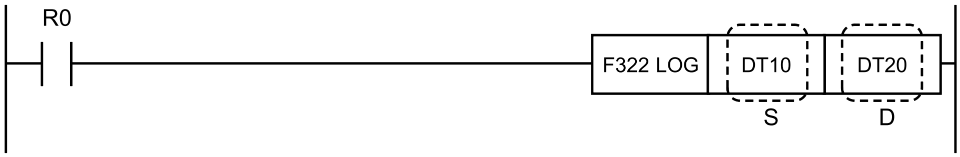
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.14 F322 LOG (Floating Point Data Logarithm Operation)

Calculates the logarithm LOG().

■ Instruction format



■ Operands

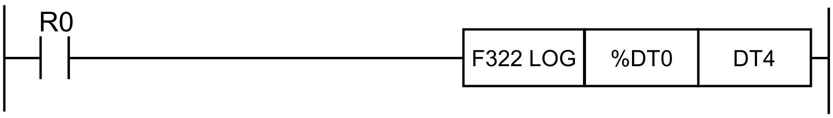
Items	Settings
S	Area storing angle data, or angle data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by ●)

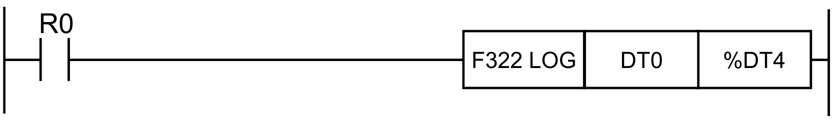
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
D		●	●	●	●	●	●	●	●							●	●

■ Outline of operation

- The logarithm LOG (S and S+1) is calculated using the data specified by S and S+1 and the result stored in D and D+1.
  LOG([S, S+1]) -> [D, D+1]
- If [S] is specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



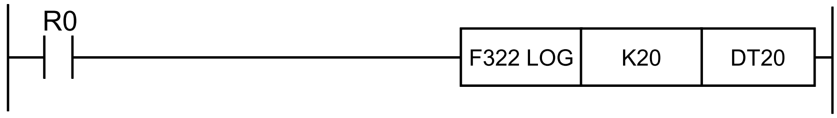
- If [D] is specified with an integer device, the real number is converted to integer data and stored.



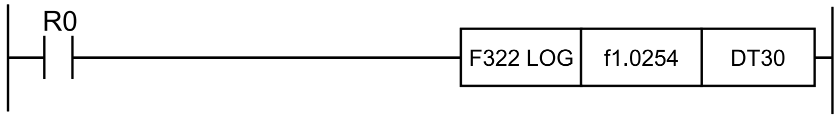
- If a K constant is specified for [S], the same process is performed as if an integer device was specified.

■ Program example

- f 1.30103 is stored in DT20 and DT21 when R0 turns ON.



- f 0.0108932 is stored in DT30 and DT31 when R0 turns ON.



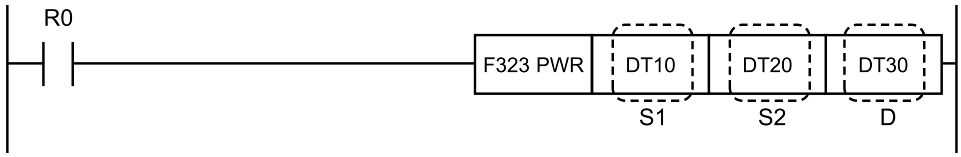
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when [S, S+1] is not 0 < [S, S+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.15 F323 PWR (Floating Point Data Power Operation)

Calculates powers for real number data.

■ Instruction format



■ Operands

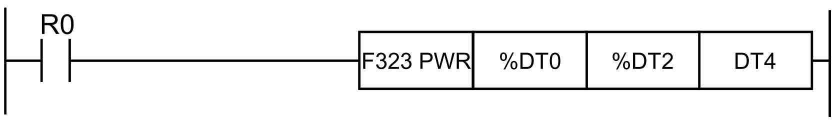
Items	Settings
S1	Area storing the base data, or base data (two words)
S2	Area storing the power data, or power data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

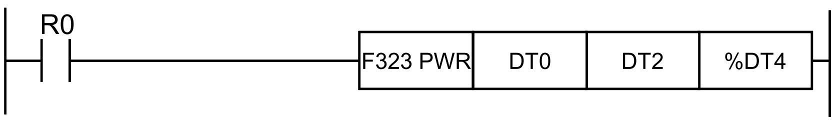
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
S2	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
D		•	•	•	•	•	•	•	•							•	•

■ Outline of operation

- Raises the base data specified by [S1, S1+1] to the power data specified by [S2, S2+1], and stores the result in [D, D+1].
  [S1, S1+1]^[S2, S2+1] → [D, D+1]
- If [S1] and [S2] are specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



- If [D] is specified with an integer device, the real number is converted to integer data and stored.



- If a K constant is specified for [S1] or [S2], the same process is performed as if an integer device was specified.

■ Program example

- When R0 turns ON, f 625.0 is stored to DT20 to DT21.

- When R0 turns ON, f 30.51758 is stored to DT30 to DT31.

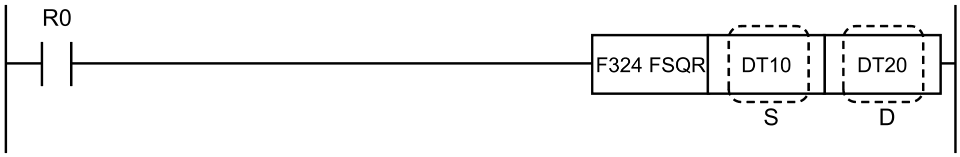
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real-number data is specified in [S1, S1+1] or [S2, S2+1]
	Turns ON when the power of negative number data is not an integer
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.16 F324 FSQR (Floating Point Data Square Root Operation)

Calculates the square root of real number data.

■ Instruction format



■ Operands

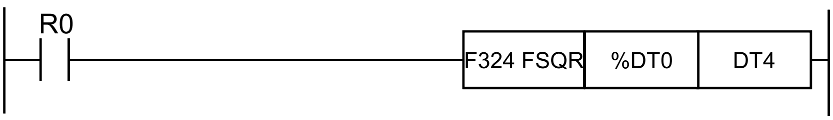
Items	Settings
S	Area storing operation data, or operation data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

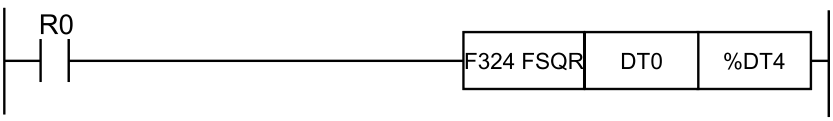
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
D		•	•	•	•	•	•	•	•							•	•

■ Outline of operation

- The square root of the operation data specified by [S, S+1] is calculated and the result is stored in [D, D+1].
  √[S, S+1] → [D, D+1]
- If [S] is specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



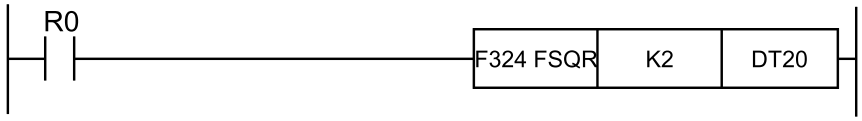
- If [D] is specified with an integer device, the real number is converted to integer data and stored.



- If a K constant is specified for [S], the same process is performed as if an integer device was specified.

■ Program example

When R0 turns ON, f1.41421 is stored in DT20 and DT21.



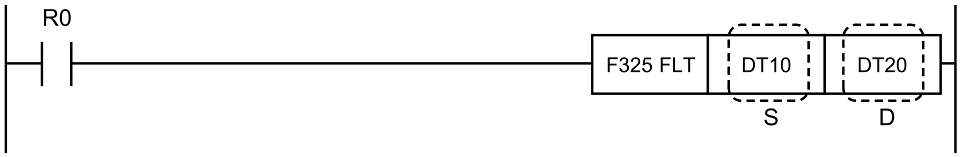
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when [S, S+1] is not 0 ≤ [S, S+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.17 F325 FLT (16-bit Integer to Floating Point Data Conversion)

Converts 16-bit integer data to real number data.

■ Instruction format



■ Operands

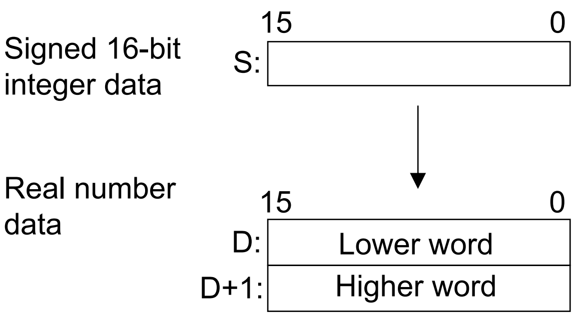
Items	Settings
S	Area storing operation data, or operation data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

Converts the operation data (signed 16-bit integer data) specified by [S] to real number data, and stores this in [D].



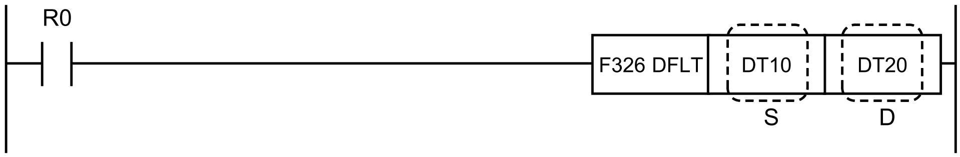
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
R900B (=)	Turns ON when the calculation result is"0"

26.18 F326 DFLT (32-bit Integer to Floating Point Data Conversion)

Converts 32-bit integers to real number data.

■ Instruction format



■ Operands

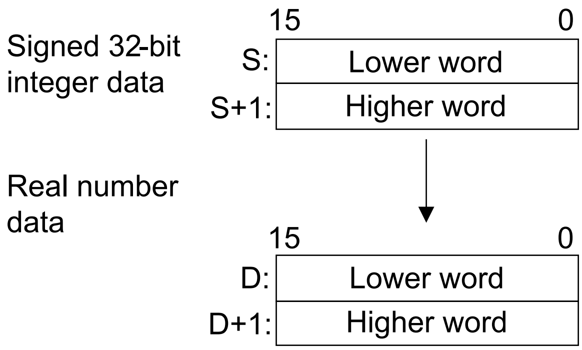
Items	Settings
S	Area storing operation data, or operation data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S	•	•	•	•	•	•	•	•	•	•	•	•	•			•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

The operation data (signed 32-bit integer data) specified by [S, S+1] is converted to real number data and stored in [D, D+1].



■ Flag operations

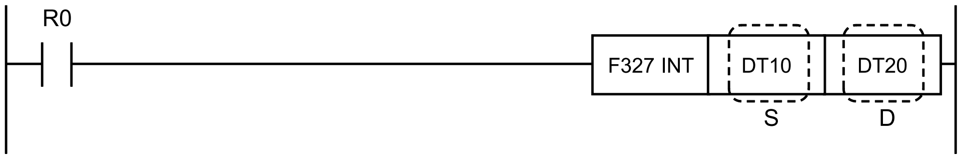
Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.

Name	Description
R9009 (CY)	Turns ON when the significant digits of the mantissa for the operation result real number data cannot be obtained
R900B (=)	Turns ON when the calculation result is"0"

26.19 F327 INT [Floating Point Data to 16-bit Integer Conversion (Largest Integer Not Exceeding the Floating-point Data)]

Converts real number data to 16-bit integers (largest integer not exceeding floating point real number).

■ Instruction format



■ Operands

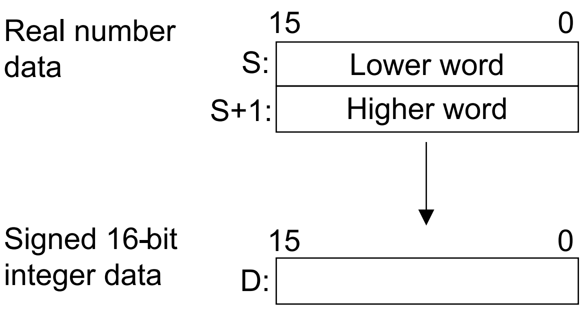
Items	Settings
S	Area storing operation data, or operation data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•		•		•	•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

The real number data (-32767.99 to +32767.99) specified by [S, S+1] is converted to signed 16-bit integers (largest integer not exceeding floating point real number) and stored in [D].



■ Operation example

Operation of instruction format description program

- If the real number 1.234 is stored in DT10 and DT11, the following operation is performed.

DT10:
DT11: (f1.234)

↓

DT20: (K1)

- If the real number -1.234 is stored in DT10 and DT11, the following operation is performed.

DT10:
DT11: (f-1.234)

↓

DT20: (K-2)

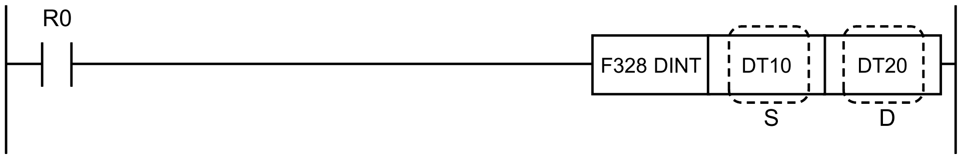
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when [D] exceeds the 16-bit integer range
R900B (=)	Turns ON when the calculation result is"0"

26.20 F328 DINT [Floating Point Data to 32-bit Integer Conversion (Largest Integer Not Exceeding the Floating-point Data)]

Converts real number data to 32-bit integers (largest integer not exceeding floating point real number).

■ Instruction format



■ Operands

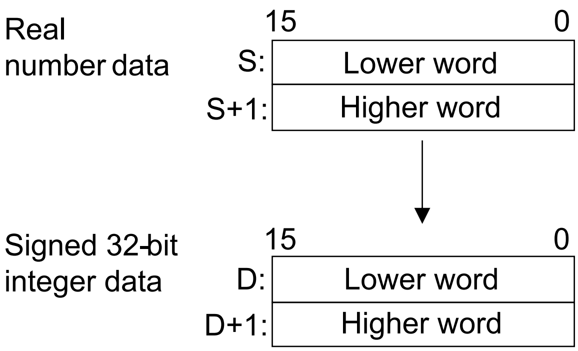
Items	Settings
S	Area storing operation data, or operation data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S	•	•	•	•	•	•	•	•	•	•	•		•		•	•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

The real number data (-2,147,483,000 to +2,147,483,000) specified by [S, S+1] is converted to signed 32-bit integers (largest integer not exceeding floating point real number) and stored in [D, D+1].



■ Operation example

Operation of instruction format description program

- If the real number 12345.67 is stored in DT10 and DT11, the following operation is performed.

DT10:
DT11: (f12345.67)

↓

DT20:
DT21: (K12345)

- If the real number -12345.67 is stored in DT10 and DT11, the following operation is performed.

DT10:
DT11: (f-12345.67)

↓

DT20:
DT21: (K-12346)

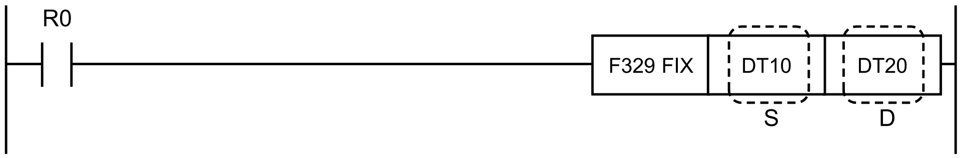
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when [D, D+1] exceeds the 32-bit integer range
R900B (=)	Turns ON when the calculation result is"0"

26.21 F329 FIX [Floating Point Data to 16-bit Integer Conversion (Round-down)]

Converts real number data to a 16-bit integer (rounded down to the nearest integer).

■ Instruction format



■ Operands

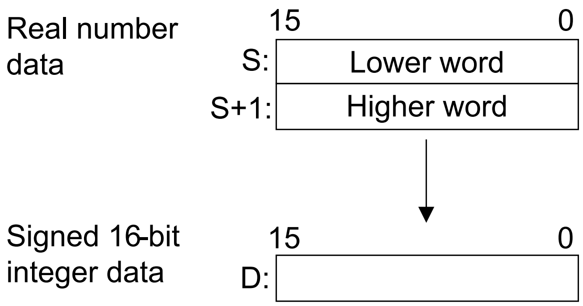
Items	Settings
S	Area storing operation data, or operation data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S	•	•	•	•	•	•	•	•	•	•	•		•		•	•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

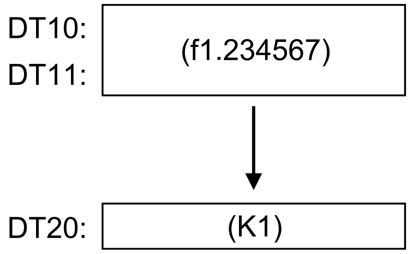
Converts real number data specified in [S, S+1] (-32767.99 to +32767.99) to a signed 16-bit integer (rounded down to the nearest integer), and stores it in [D].



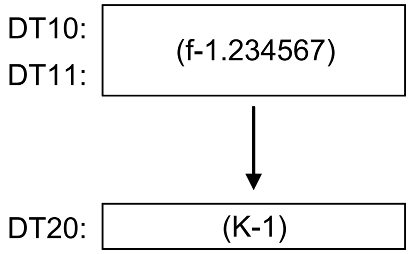
■ Operation example

Operation of instruction format description program

- When the real number 1.234567 is stored in DT10 and DT11, the following operation is performed.



- When the real number -1.234567 is stored in DT10 and DT11, the following operation is performed.



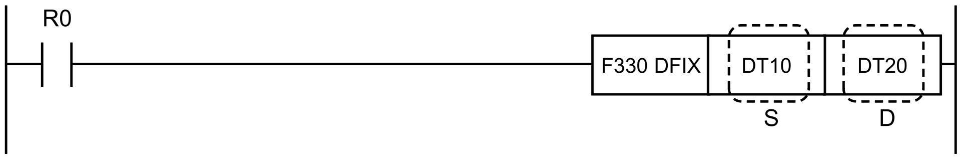
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when [D] exceeds the 16-bit integer range
R900B (=)	Turns ON when the calculation result is"0"

26.22 F330 DFIX [Floating Point Data to 32-bit Integer Conversion (Round-down)]

Converts real number data to 32-bit integers (rounding down the decimal point).

■ Instruction format



■ Operands

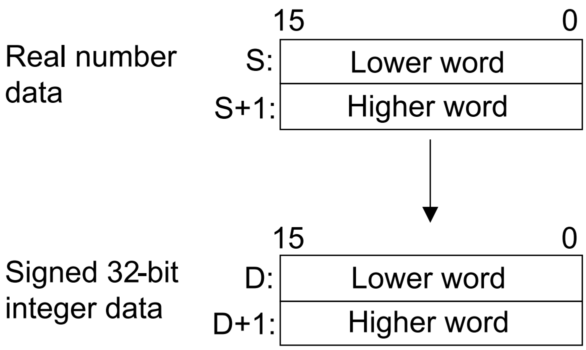
Items	Settings
S	Area storing operation data, or operation data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•		•		•	•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

The real number data (-2,147,483,000 to +2,147,483,000) specified by [S, S+1] is converted to signed 32-bit integers (rounding down the decimal point), and stored in [D, D+1].



■ Operation example

Operation of instruction format description program

- If the real number 123456.7 is stored in DT10 to DT11, the following operation is performed.

DT10:
DT11: (f123456.7)

↓

DT20:
DT21: (K123456)

- If the real number -123456.7 is stored in DT10 to DT11, the following operation is performed.

DT10:
DT11: (f-123456.7)

↓

DT20:
DT21: (K-123456)

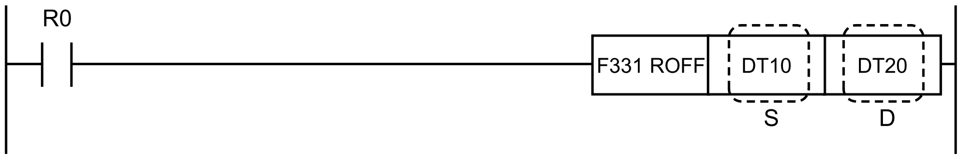
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when [D, D+1] exceeds the 32-bit integer range
R900B (=)	Turns ON when the calculation result is"0"

26.23 F331 ROFF [Floating Point Data to 16-bit Integer Conversion (Round-off)]

Converts real number data to a 16-bit integer (rounded off to the nearest integer).

■ Instruction format



■ Operands

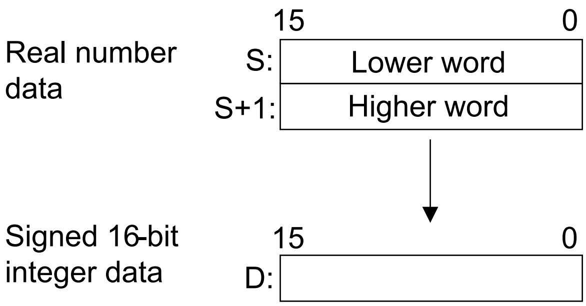
Items	Settings
S	Area storing operation data, or operation data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•		•		•	•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

Converts the real number data (-32767.99 to +32767.99) specified in [S, S+1] to a signed 16-bit integer (rounded off to the nearest integer) and stores it in [D].



■ Operation example

Operation of instruction format description program

- When the real number 1234.567 is stored in DT10 and DT11, the following operation will be performed.

DT10: DT11: (f1234.567)

↓

DT20: (K1235)

- When the real number -1234.567 is stored in DT10 and DT11, the following operation will be performed.

DT10: DT11: (f-1234.567)

↓

DT20: (K-1235)

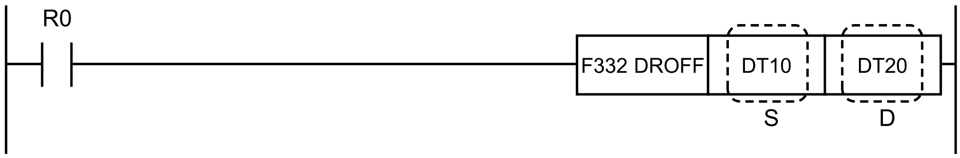
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when [D] exceeds the 16-bit integer range
R900B (=)	Turns ON when the calculation result is"0"

26.24 F332 DROFF [Floating Point Data to 16-bit Integer Conversion (Round-off)]

Converts real number data to 32-bit integers (rounding off at the decimal point).

■ Instruction format



■ Operands

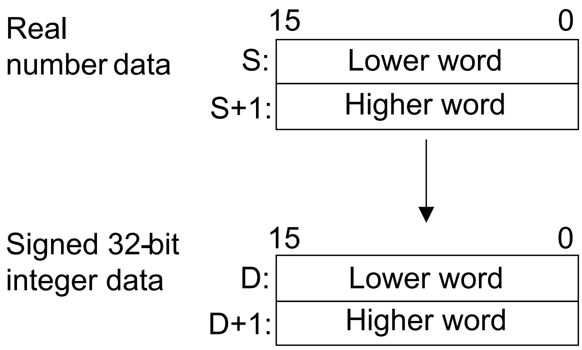
Items	Settings
S	Area storing operation data, or operation data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•		•		•	•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

The real number data specified by [S, S+1] (-2,147,483,000 to +2,147,483,000) is converted to signed 32-bit integers (rounding off at the decimal point) and stored in [D, D+1].



■ Operation example

Operation of instruction format description program

- If the real number 45678.51 is stored in DT10 and DT11, the following operation is performed.

DT10:
DT11: (f45678.51)

↓

DT20:
DT21: (K45679)

- If the real number -45678.51 is stored in DT10 and DT11, the following operation is performed.

DT10:
DT11: (f-45678.51)

↓

DT20:
DT21: (K-45679)

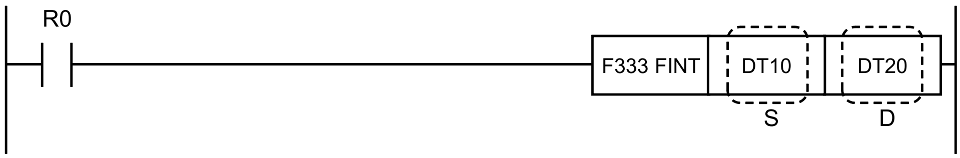
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when [D, D+1] exceeds the 32-bit integer range
R900B (=)	Turns ON when the calculation result is"0"

26.25 F333 FINT (Floating Point Data Round-down)

Rounds down real number data at the decimal point. (The largest integer not exceeding the floating point type data)

■ Instruction format



■ Operands

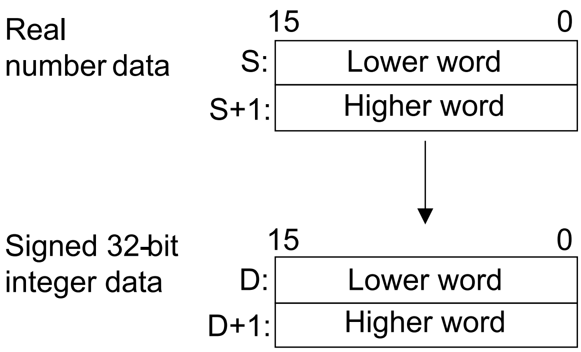
Items	Settings
S	Area storing operation data, or operation data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•		•		•	•	
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

The real number data specified by [S, S+1] is rounded down at the decimal point and the result is stored in [D, D+1].



■ Operation example

Operation of instruction format description program

- When the real number 1234.567 is stored in DT10 and DT11, the following operation will be performed.

DT10:
DT11: (f1234.567)

↓

DT20:
DT21: (f1234.000)

- When the real number -1234.567 is stored in DT10 and DT11, the following operation will be performed.

DT10:
DT11: (f-1234.567)

↓

DT20:
DT21: (f-1235.000)

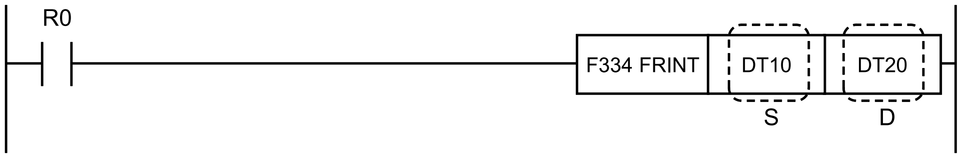
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.26 F334 FRINT (Floating Point Data Round-off)

Rounds off real number data to the first decimal place.

■ Instruction format



■ Operands

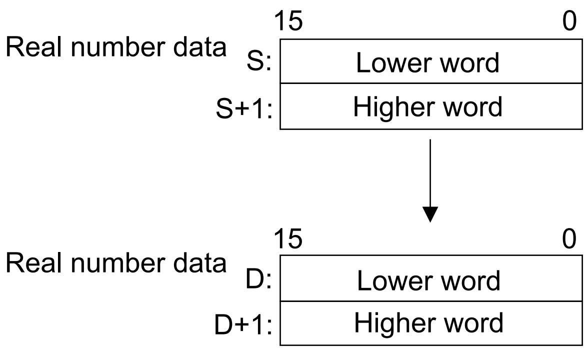
Items	Settings
S	Area storing operation data, or operation data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●		●		●	●	
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

The decimal part of the real number data specified by [S, S+1] is rounded off to the first decimal place, and the result is stored in [D, D+1].



■ Operation example

Operation of instruction format description program

- When the real number 1234.567 is stored in DT10 and DT11, the following operation will be performed.

DT10:
DT11: (f1234.567)

↓

DT20:
DT21: (f1235.000)

- When the real number -1234.567 is stored in DT10 and DT11, the following operation will be performed.

DT10:
DT11: (f-1234.567)

↓

DT20:
DT21: (f-1235.000)

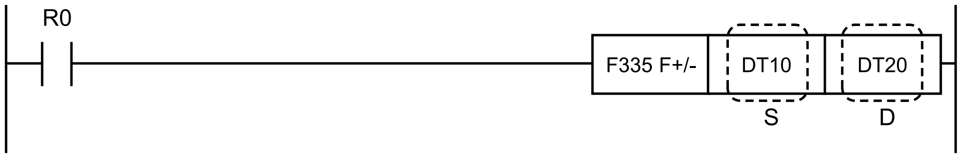
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.27 F335 F+/- (Floating Point Data Sign Conversion)

Changes the sign of real number data.

■ Instruction format



■ Operands

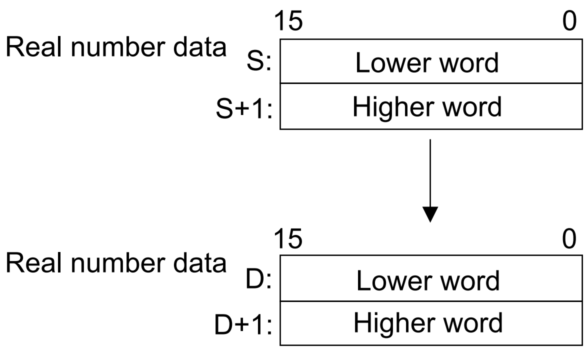
Items	Settings
S	Area storing operation data, or operation data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

The sign for the real number data specified by [S, S+1] is changed and the result stored in [D, D +1].



■ Operation example

Operation of instruction format description program

- If the real number "-60000.00" is stored in DT10 to DT11, the following operation will be performed.

DT10:
DT11: (f-60000.00)

↓

DT20:
DT21: (f60000.00)

- If the real number "-30000.00" is stored in DT10 to DT11, the following operation will be performed.

DT10:
DT11: (f-30000.00)

↓

DT20:
DT21: (f30000.00)

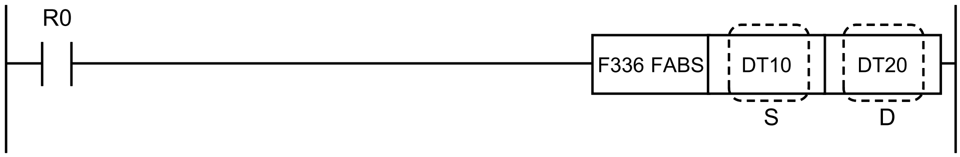
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
R9009 (CY)	Turns ON when operation result overflows

26.28 F336 FABS (Floating Point Data Absolute Value Conversion)

Calculates the absolute value of real number data.

■ Instruction format



■ Operands

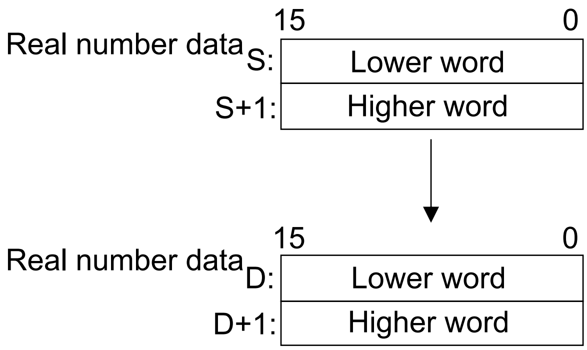
Items	Settings
S	Area storing operation data, or operation data (two words)
D	Area storing the operation results (two words)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
D		●	●	●	●	●	●	●	●							●	

■ Outline of operation

- Calculates the absolute value of the real number data specified in [S, S+1], then stores the result in [D, D+1].



- If [S] is specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.
- If a K constant is specified for [S], the same process is performed as if an integer device was specified.

■ Operation example

Operation of instruction format description program

- When the real number 1234.567 is stored in DT10 and DT11, the following operation will be performed.

DT10:
DT11: (f1234.567)

↓

DT20:
DT21: (f1234.567)

- When the real number -1234.567 is stored in DT10 and DT11, the following operation will be performed.

DT10:
DT11: (f-1234.567)

↓

DT20:
DT21: (f1234.567)

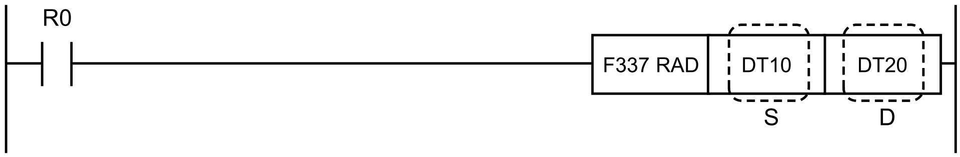
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.29 F337 RAD (Degree to Radian Conversion)

Converts the unit of an angle from [degrees] to [radians].

■ Instruction format



■ Operands

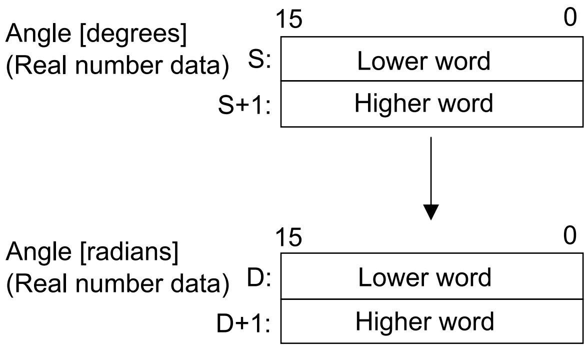
Items	Settings
S	Area storing angle [degrees] data, or angle [degrees] (two words)
D	Area (two word) to store the conversion result

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
D		•	•	•	•	•	•	•	•							•	

■ Outline of operation

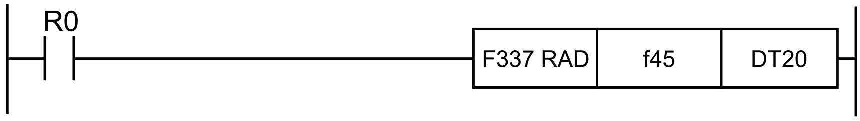
- The angle [degrees] specified by [S, S+1] is converted into an angle [radians] (real number data), and the result is stored in [D, D+1].



- If [S] is specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.
- If a K constant is specified for [S], the same process is performed as if an integer device was specified.

■ Program example

When R0 turns ON, f0.7853981 is stored in DT20 and DT21.



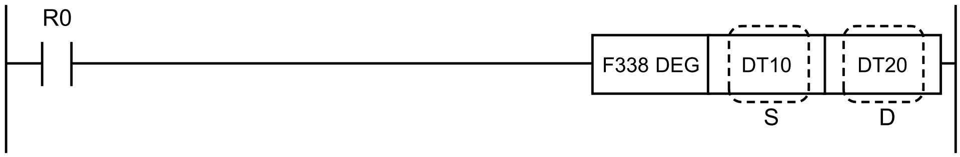
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

26.30 F338 DEG (Radian to Degree Conversion)

Converts the unit of an angle from radians to degrees.

■ Instruction format



■ Operands

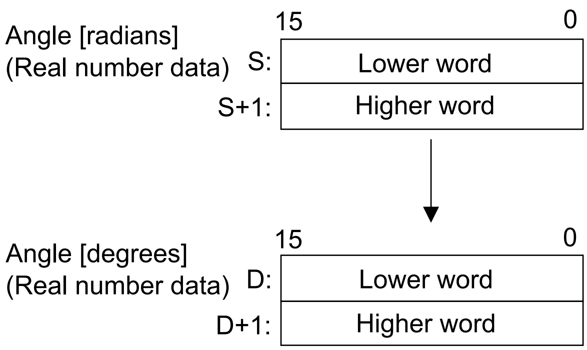
Items	Settings
S	Area storing angle data (radians), or angle data (radians) (two words)
D	Area (two words) to store the conversion result

■ Devices that can be specified (indicated by •)

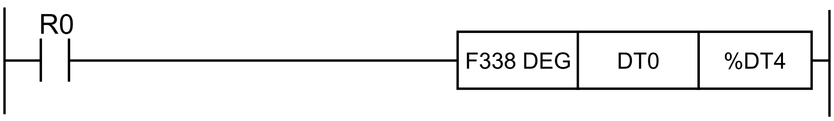
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
D		•	•	•	•	•	•	•	•							•	•

■ Outline of operation

- The angle data in radians (real number data) specified by [S, S+1] is converted to angle data in degrees, and the result is stored in [D, D+1].



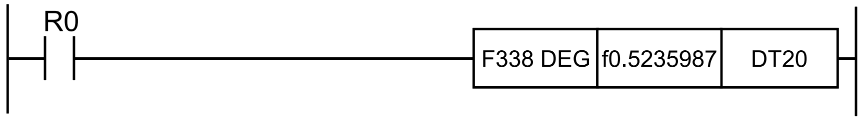
- If [D] is specified with an integer device, the real number is converted to integer data and stored.



- If a K constant is specified for [S], the same process is performed as if an integer device was specified.

■ Program example

When R0 turns to ON, f30.00000 is stored in DT20 and DT21.



■ Precautions for programming

When a constant is specified for [S], an integer device cannot be specified for [D].

■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real number data is specified in [S, S+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the calculation result is"0"

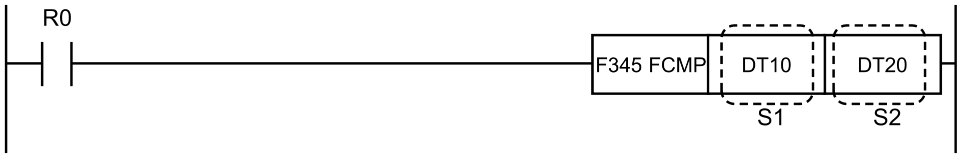
(MEMO)

27 Real Number Data Processing Instructions

27.1 F345 FCMP (Floating Point Data Comparison)

Compares real number data and outputs the judgment result to special internal relays.

■ Instruction format



■ Operands

Items	Settings
S1	Area storing the real number data, or real number data (comparison data 1) (two words)
S2	Area storing the real number data, or real number data (comparison data 2) (two words)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●

■ Outline of operation

- The real number data specified by [S1, S1+1] is compared with the real number data specified by [S2, S2+1], and the judgment result is output to the special internal relay flags (R9009 to R900C).
- The size relationship between [S1, S1+1] and [S2, S2+1] affects R9009 to R900C as follows.

Relationship between [S1, S1+1] and [S2, S2+1]	Flag			
	R900A	R900B	R900C	R9009
	>	=	<	Carry
[S1, S1+1]<[S2, S2+1]	OFF	OFF	ON	Indefinite
[S1, S1+1]=[S2, S2+1]	OFF	ON	OFF	OFF
[S1, S1+1]>[S2, S2+1]	ON	OFF	OFF	Indefinite

- If [S1] and [S2] are specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.
- If a K constant is specified for [S1] or [S2], the same processing is performed as when an integer device is specified.

■ Flag operations

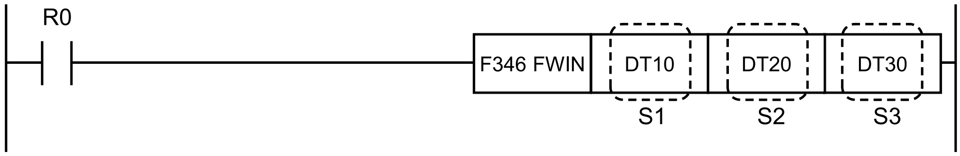
Name	Description
R9007 R9008	Turns ON when the area is exceeded in index modification.

Name	Description
(ER)	Turns ON when non-real-number data is specified in [S1, S1+1] or [S2, S2+1]

27.2 F346 FWIN (Floating Point Data Band Comparison)

Compares real number data with a band and outputs the judgment result to special internal relays.

■ Instruction format



■ Operands

Items	Settings
S1	Comparison data: Area storing real number data, or real number data (two words)
S2	Lower limit data: Area storing real number data, or real number data (two words)
S3	Upper limit data: Area storing real number data, or real number data (two words)

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●
S3	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●

■ Outline of operation

- A band comparison is performed on real number data.
  The real number data specified by [S1, S1+1] is compared with the range specified by [S2, S2+1] (lower limit value) and [S3, S3+1] (upper limit value) to determine whether it falls in that range, and the comparison result is output to the special internal relays R9009 to R900C (comparison instruction judgment flags).
- The relationship between [S1, S1+1], [S2, S2+1], and [S3, S3+1] affects R9009 to R900C as follows.
  ×: Does not change.

Relationship between [S1, S1+1], [S2, S2+1], [S3, S3+1]	Flag			
	R900A	R900B	R900C	R9009
	>	=	<	Carry
[S1, S1+1] < [S2, S2+1]	OFF	OFF	ON	×
[S2, S2+1] ≤ [S1, S1+1] ≤ [S3, S3+1]	OFF	ON	OFF	×
[S3, S3+1] < [S1, S1+1]	ON	OFF	OFF	×

- If [S1] to [S3] are specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.
- If a K constant is specified for [S1], [S2], or [S3], the same process is performed as if an integer device was specified.

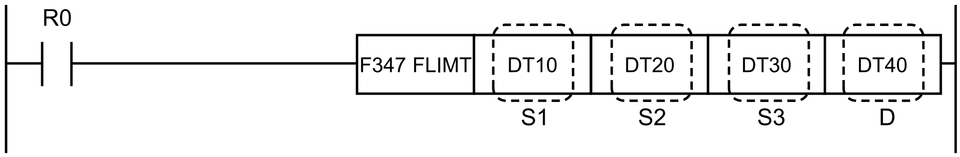
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real-number data is specified in [S1, S1+1], [S2, S2+1] or [S3, S3+1]
	Turns ON when [S2, S2+1] is greater than [S3, S3+1]

27.3 F347 FLIMT (Floating Point Data Upper/Lower Limit Control)

Performs upper and lower limit control (real number data).

■ Instruction format



■ Operands

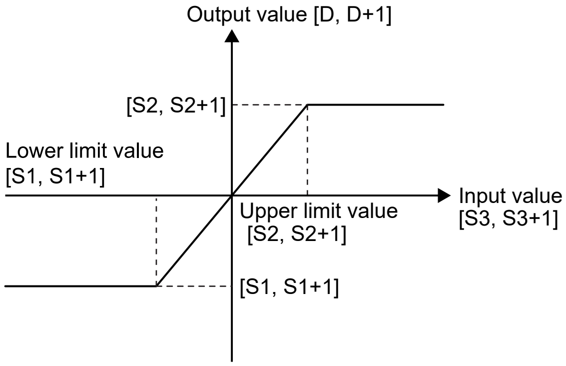
Items	Settings
S1	Area storing the lower limit, or lower limit data (two words)
S2	Area storing the upper limit, or upper limit data (two words)
S3	Area storing the input value, or input value data (two words)
D	Area storing the output value (two words)

■ Devices that can be specified (indicated by •)

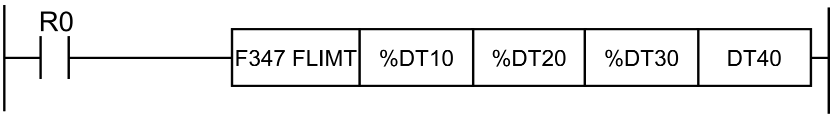
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
S2	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
S3	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
D		•	•	•	•	•	•	•	•							•	•

■ Outline of operation

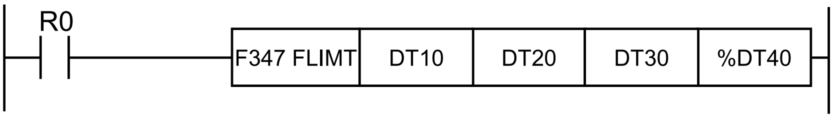
- The output value (real number data) stored in the area specified by [D] is controlled according to whether or not the input value (real number data) specified by [S3] falls within the range bounded by the upper and lower limits (real number data) specified by [S1] and [S2].
- The output value is determined based on the following conditions.
  - If lower limit value [S1, S1+1] is greater than input value [S3, S3+1], then lower limit value [S1, S1+1] becomes output value [D, D+1]
  - If upper limit value [S2, S2+1] is less than input value [S3, S3+1], then upper limit value [S2, S2+1] becomes output value [D, D+1]
  - If lower limit value [S1, S1+1] is equal to or less than input value [S3, S3+1], which is equal to or less than upper limit value [S2, S2+1], then input value [S3, S3+1] becomes output value [D, D+1]



- If [S1] to [S3] are specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



- If [D] is specified with an integer device, the real number is converted to integer data and stored.



- If a K constant is specified for [S1], [S2], or [S3], the same process is performed as if an integer device was specified.

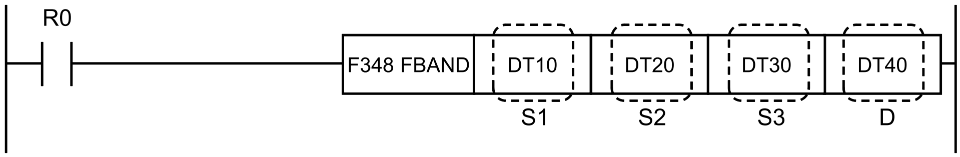
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real-number data is specified in [S1, S1+1], [S2, S2+1] or [S3, S3+1]
	Turns ON when [S1, S1+1] is greater than [S2, S2+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R900B (=)	Turns ON when the operation result falls within the upper/lower limit range

27.4 F348 FBAND (Floating Point Data Deadband Control)

Performs dead-band control (real number data).

■ Instruction format



■ Operands

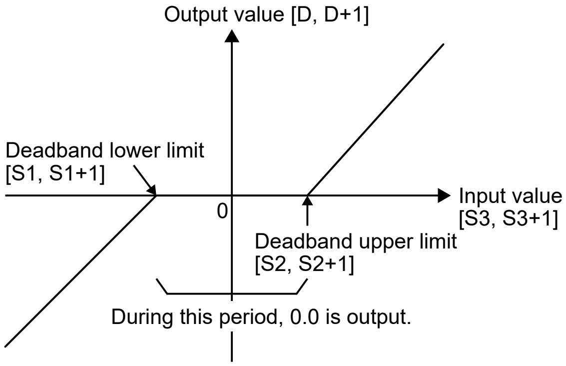
Items	Settings
S1	Area storing the lower limit, or lower limit data (two words)
S2	Area storing the upper limit, or upper limit data (two words)
S3	Area storing the input value, or input value data (two words)
D	Area storing the output value (two words)

■ Devices that can be specified (indicated by •)

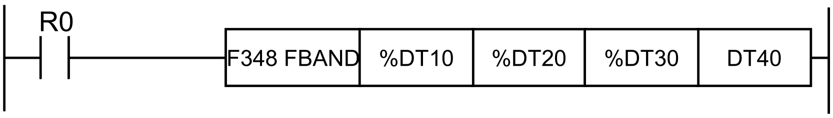
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S1	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
S2	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
S3	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
D		•	•	•	•	•	•	•	•							•	•

■ Outline of operation

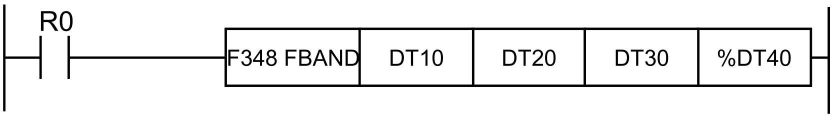
- The output value (real number data) stored in the area specified by [D] is controlled according to whether the input value (real number data) specified by [S3] is within the range of the upper and lower limits (real number data) of the dead-band specified by [S1] and [S2].
- The output value is determined based on the following conditions.
  - When the lower limit [S1, S1+1] > input value [S3, S3+1], the input value [S3, S3+1] - the lower limit [S1, S1+1] becomes the output value [D, D+1]
  - When the upper limit [S2, S2+1] < input value [S3, S3+1], the input value [S3, S3+1] - the upper limit [S2, S2+1] becomes the output value [D, D+1]
  - When the lower limit [S1, S1+1] ≤ input value [S3, S3+1] ≤ the upper limit [S2, S2+1], 0.0 becomes the output value [D, D+1]



- If [S1] to [S3] are specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



- If [D] is specified with an integer device, the real number is converted to integer data and stored.



- If a K constant is specified for [S1], [S2], or [S3], the same process is performed as if an integer device was specified.

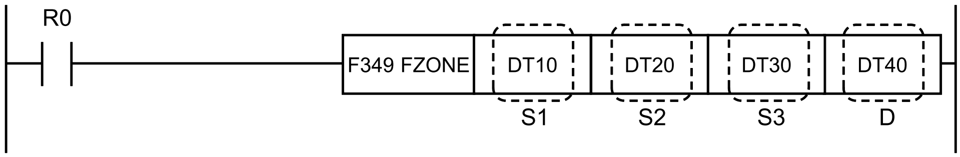
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real-number data is specified in [S1, S1+1], [S2, S2+1] or [S3, S3+1]
	Turns ON when [S1, S1+1] is greater than [S2, S2+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the operation result falls within the upper/lower limit range

27.5 F349 FZONE (Floating Point Data Zone Control)

Performs zone control (real number data).

■ Instruction format



■ Operands

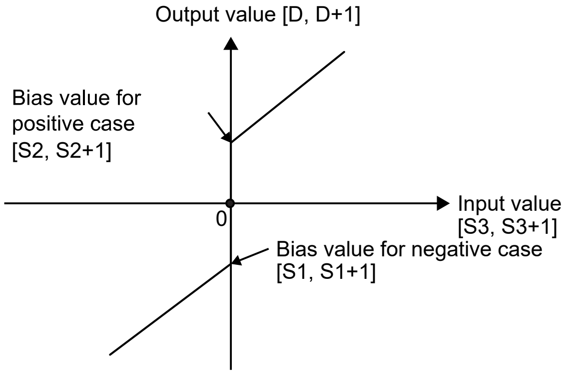
Items	Settings
S1	Area storing negative bias values, or negative bias value data (two words)
S2	Area storing positive bias values, or positive bias value data (two words)
S3	Area storing the input value, or input value data (two words)
D	Area storing the output value (two words)

■ Devices that can be specified (indicated by •)

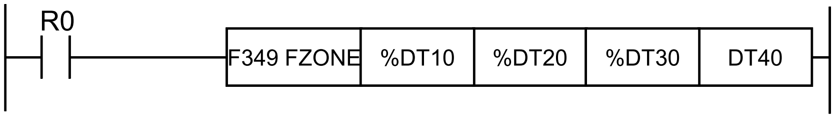
Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
S2	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
S3	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
D		•	•	•	•	•	•	•	•							•	•

■ Outline of operation

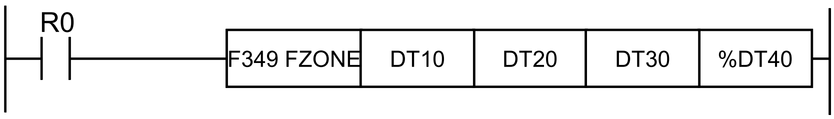
- The bias value specified in [S1] or [S2] is added to the input value (real number data) specified in [S3], and the result is stored in the area specified in [D].
- The output value is determined based on the following conditions.
  - When the input value [S3, S3+1] is less than 0.0, the input value [S3, S3+1] + the negative bias value [S1, S1+1] is the output value [D, D+1]
  - When the input value [S3, S3+1] is equal to 0.0, 0.0 is the output value [D, D+1]
  - When the input value [S3, S3+1] is more than 0.0, the input value [S3, S3+1] + the positive bias value [S2, S2+1] is the output value [D, D+1]



- If [S1] to [S3] are specified with an integer device, the operation occurs after the integer data is internally converted to real numbers.



- If [D] is specified with an integer device, the real number is converted to integer data and stored.



- If a K constant is specified for [S1], [S2], or [S3], the same process is performed as if an integer device was specified.

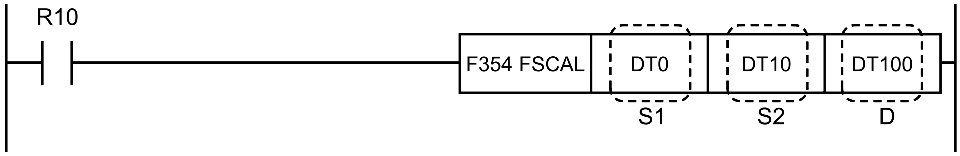
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when non-real-number data is specified in [S1, S1+1], [S2, S2+1] or [S3, S3+1]
	Turns ON when the operation result exceeds the integer range when an integer device is specified in [D, D+1]
R9009 (CY)	Turns ON when operation result overflows
R900B (=)	Turns ON when the input value is"0"

27.6 F354 FSCAL (Scaling of real number data)

Performs scaling (linearization) using a real number data table and calculates the output (Y) for the input value (X).

■ Instruction format



■ Operands

Items	Settings
S1	Real value or area representing the input value (X)
S2	Starting area of data table used for scaling
D	Area storing output value (Y)

■ Devices that can be specified (indicated by •)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
S2	•	•	•	•	•	•	•	•	•	•	•					•	
D		•	•	•	•	•	•	•	•							•	•

■ Outline of operation

- The input real value [S1] is scaled (linearized) according to the real number data table specified by [S2], and the output value is stored in [D].
- The section corresponding to the input value [S1] is searched from the table specified by [S2], the linear interpolation between these two points is calculated, and the output value is obtained.

  When the specified input value is outside the registration range in the table, the start point (x0) or end point (xn) is stored for the output value (Y0 or Yn).

  [S1] ≤ x0 [D] ← y0

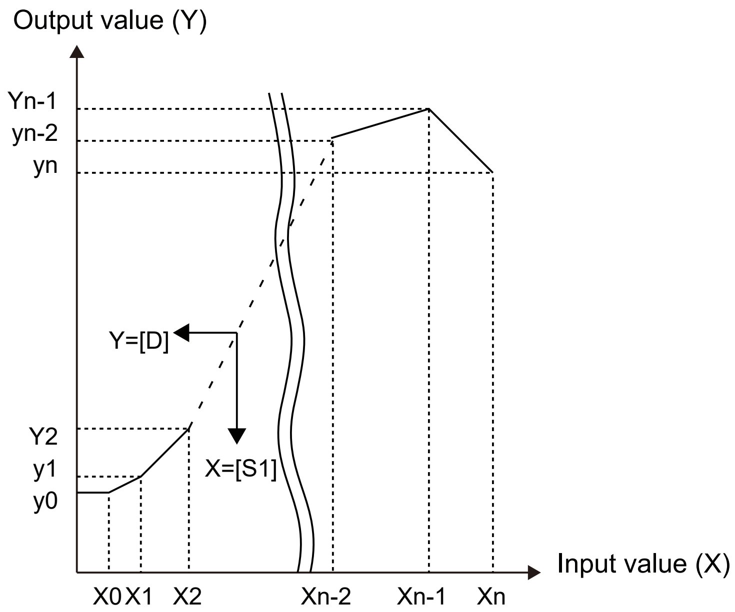
  [S1] ≥ xn [D] ← yn

■ Operation example

Operation of instruction format description program

The output value Y for the input value stored in DT0 is obtained by referring to the data table starting from DT10, and the result is stored in DT100.

S2:	Registration point count (m←n+1)	DT10
S2+1:	x0	DT11
S2+2:	(Real number value)	DT12
S2+3:	x1	DT13
S2+4:	(Real number value)	DT14
S2+2n+1:	xn	DT29
S2+2n+2:	(Real number value)	DT30
S2+2n+3:	y0	DT31
S2+2n+4:	(Real number value)	DT32
S2+2n+5:	y1	DT33
S2+2n+6:	(Real number value)	DT34
S2+4n+3:	yn	DT49
S2+4n+4:	(Real number value)	DT50



- The data table [S2] used for scaling must have two or more sections registered. In addition, the points must be registered in order from the smallest number on the X axis to the largest number.

  2 ≤ Number of registered points (m) ≤ 99 [Number of registered points (m) = n + 1]

  $x_{t-1} < x_t$ ($1 \le t \le n$)
- When the distance between two points on the data table is very large, an operation error will occur.

(This occurs when the distance between two points cannot be represented by a real number.)

e.g.

First point: (x0, y0) = (HFF000000, HFF000000) = ($-1.7*10^{34}$, $-1.7*10^{34}$)

Second point: (x1,y1) = (H7F000000, H7F000000) = ($+1.7*10^{34}$, $+1.7*10^{34}$)

- The measurement error of the output result is proportional to the distance between two points of the data table.
- When an integer device is specified for the input value [S1], scaling is performed after converting it to a real value.
- When an integer device is specified for the output value [S2], the output result is converted to an integer value and stored.

■ Flag operations

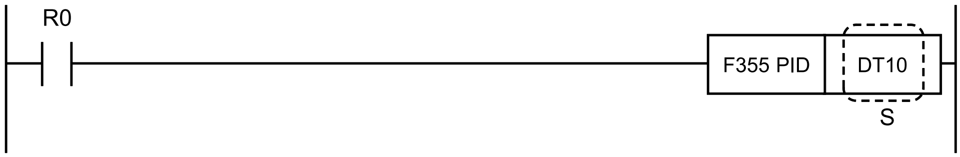
Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when a non-real value is entered in [S1]
	Turns ON when m < 2 or m > 99 in the registered points of [S2]
	Turns ON when a non-real value is specified for the real value (xt, yt) specified in [S2]
	Turns ON when the data table of [S2] is not registered in ascending order of the X axis
	Turns ON when data table in [S2] exceeds area
	Turns ON when an overflow (calculation not possible) occurs in the scaling calculation
	Turns ON when the output result exceeds the integer range when an integer device is specified in [D]

28 Process Control Instructions

28.1 F355 PID (PID Operation)

PID operation is performed.

■ Instruction format



■ Operands

Items	Settings
S	Starting number of parameter area (30 word) for PID operation

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant K	Constant H	Constant M	Constant f	Index modifier	Integer Device
S							●										

■ Outline of operation

- PID operation is performed to match and hold the measurement value [S+2] at the setting value [S+1], and the result is output to [S+3].
- Derivative priority type or proportional-derivative priority type can be selected for PID operation.
- Set the coefficients (proportional gain, integral time, derivative time) used for PID operation and the operation type/interval in the parameter table. PID operation will be performed according to the specified content.

■ Types of PID operation

(1) Reverse operation / Forward operation

The vertical direction of output when there is a change to the process can be selected.

- Specify"Reverse operation"if increasing the output when the measured value falls. (Heating, etc.)
- Specify"Forward operation"if decreasing the output when the measured value rises. (Cooling, etc.)

(2) Derivative priority type PID / Proportional-derivative priority type PID

- In general, with"Derivative priority type PID control", there is increased fluctuation in the output when the set value changes, but convergence is faster.
- In general, with"Proportional-derivative priority type PID control", there is less output fluctuation when the set value changes, but convergence is slower.

■ Parameter table settings

[S]	Control mode
[S+1]	Set point value (SP)
[S+2]	Measured process value (PV)
[S+3]	Output value (MV)
[S+4]	Output lower limit
[S+5]	Output upper limit
[S+6]	Proportional gain (Kp)
[S+7]	Integral time (Ti)
[S+8]	Derivative time (Td)
[S+9]	Control interval (Ts)
[S+10]	Auto-tuning progress status
[S+11] ~ [S+29]	Work area for PID operation

■ Description of each parameter

(1) Control mode [S]

Specify the PID operation type and auto-tuning with H constants.

Control mode		[S] value	
		When not executing auto-tuning	When executing auto-tuning
Derivative type	Reverse	H0	H8000
	Forward	H1	H8001
Proportional-derivative type	Reverse	H2	H8002
	Forward	H3	H8003

- Auto-tuning

  The optimal values for the PID parameters Kp, Ti, and Td are measured by measuring the process response.

  When auto-tuning is executed, the estimated results are reflected in the parameter area after auto-tuning is complete. (Depending on the process, execution of auto-tuning may not be possible. In such cases, the process will return to the original parameter operation.)

  For precautions regarding the execution of auto-tuning, please refer to"P.28-5".

- Reverse operation, forward operation

  The vertical direction of output when there is a change to the process is determined.

Reverse	The output is increased if the measured value of the process falls. (e.g. heating)
Forward	The output is increased if the measured value of the process rises. (e.g. cooling)

- Derivative priority type, proportional-derivative priority type PID
  There is a change in output when the setting value is changed.

Derivative type	Generally, there is significant fluctuation when the setting value is changed, but convergence is fast.
Proportional-derivative type	Generally, there is less fluctuation when the setting value is changed, but convergence is slow.

(2) Set value (SP) [S+1]

Set the target value for the process control within the following range.

K0 to K10000

(3) Measured value (PV) [S+2]

Use an A/D conversion unit, etc., to input the current value of process control. Make sure it is within the following range.

K0 to K10000

(4) Output value (MV) [S+3]

The value from PID processing is stored. Use a D/A conversion unit, etc., to output to the process.

K0 to K10000

(5) Output lower limit [S+4]

K0 to K9999 (< upper limit)

(6) Output upper limit [S+5]

K1 to K10000 (> lower limit)

Specify the output value (MV) range. Values for the specified range are output.

Make sure that 0 ≤ output lower limit < output upper limit ≤ 10000.

(7) Proportional gain (Kp) [S+6]

Specify the coefficient used for PID operation.

The setting value × 0.1 is the actual proportional gain.

The setting value range is K1 to K9999 (0.1 to 999.9, specified in units of 0.1).

If auto-tuning is specified in the operation mode specifications, the setting value is automatically adjusted and rewritten.

(8) Integral time (Ti) [S+7]

Specify the coefficient used for PID operation.

Actual integral time is set point value × 0.1.

The setting value range is K1 to K30000 (0.1 to 3000 seconds, specified in units of 0.1 second).

If 0 is specified, integration will not be executed.

If auto-tuning is specified in the operation mode specifications, the setting value is automatically adjusted and rewritten.

(9) Derivative time (Td) [S+8]

Specify the coefficient used for PID operation.

Actual derivative time is set point value × 0.1.

The setting value range is K0 to K10000 (0 to 1000 seconds, specified in units of 0.1 second).

If auto-tuning is specified in the operation mode specifications, the setting value is automatically adjusted and rewritten.

(10) Control interval (Ts) [S+9]

Specify the interval for executing the PID operation. The setting value × 0.01 is the actual control interval.

The setting value range is K1 to K6000 (0.01 to 60.0 seconds, specified in units of 0.01 second).

(11) Auto-tuning progress status [S+10]

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.

(12) Work area for PID operation [S+11] to [S+29]

The work area used by the system that is required for operations.

■ Precautions when executing auto-tuning

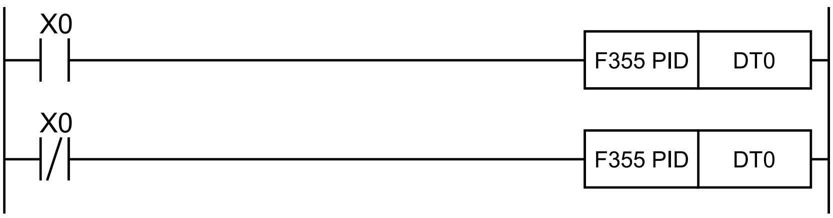
Note the following points if"Auto-tuning Execution"is set in the parameter table (control mode [S]).

- After auto-tuning is complete, the area of control mode [S] is automatically rewritten from H8000 to H8003, to H0 to H3. Make sure that it is not rewritten again by the program, etc.
- After auto-tuning is complete, the optimal values for proportional gain [Kp], integral time [Ti], and derivative time [Td] are stored, but it is necessary to specify appropriate values within the setting range (for example, the lower limit) before execution.
- After auto-tuning is complete, the optimal values for proportional gain [Kp], integral time [Ti], and derivative time [Td] are stored. Be careful that the stored values are not rewritten.
- The optimal values for Kp, Ti, and Td are calculated by auto-tuning determining the set point value (SP) by measuring the change of the measured value (PV) when the output value (MV) is set to the upper limit, causing the measured value (PV) to fluctuate, and then measuring the change of the measured value (PV) when the output value (MV) is set to the lower limit.
- The change of the output value (MV) for auto-tuning is completed after a minimum of 3 changes: upper limit output -> lower limit output -> upper limit output. If the auto-tuning progress status is still at 0 after several changes, shorten the control synchronization Ts and execute auto-tuning again.

■ Precautions for programming

- Including the work area for operation, a 30 word area is required for the parameter table. Take care that the values in this area are not rewritten by other instructions.
- Even if the parameter table exceeds the area, an error will not be detected. When specifying [S], specify a number that is within a minimum of 30 words from the last number.
- Take care that the area is not exceeded by index modification. Even if the area is exceeded, an error will not be detected.
- Use an A/D conversion unit, etc., to input the current value of the measured value [S+2].
- Use a D/A conversion unit, etc., to output the result of PID processing [S+3} to the process.
- If two or more PID instructions specifying the same table are included in the program, it may not operate correctly.

<Example>



(Reason) This is because the F355 PID instruction operates internally using the specified table, even when the execution condition is not met.

In such cases, set the tables to separate addresses.

Info.

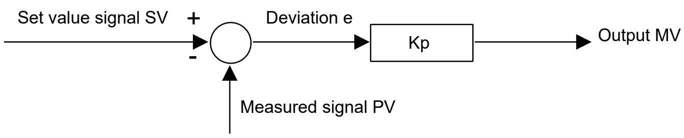
- See the following operational expressions regarding PID operation.

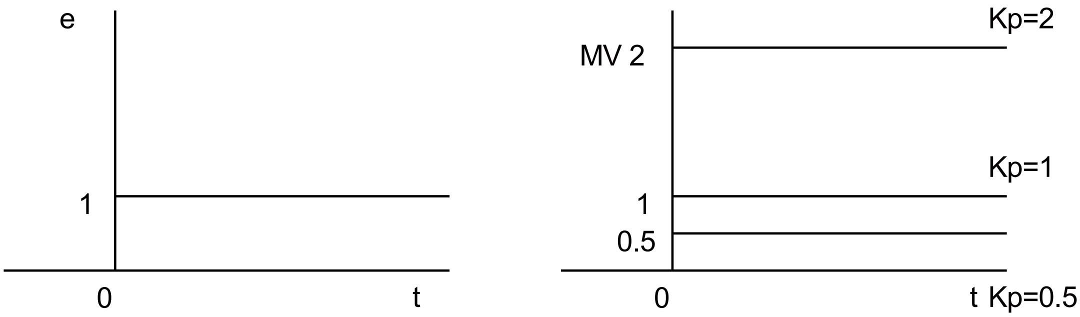
■ Outline of operation of PID control

PID control is a feedback control method widely used in the instrumentation field to control process quantities such as temperature, pressure, flow rate, and fluid levels.

(1) Proportional operation

Control operation that produces an output proportional to the size of the input





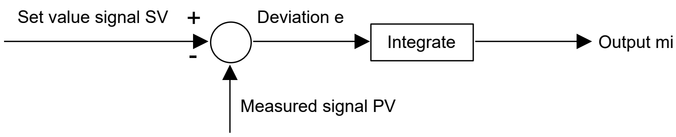
A constant control quantity is maintained.

An offset (regular deviation) remains.

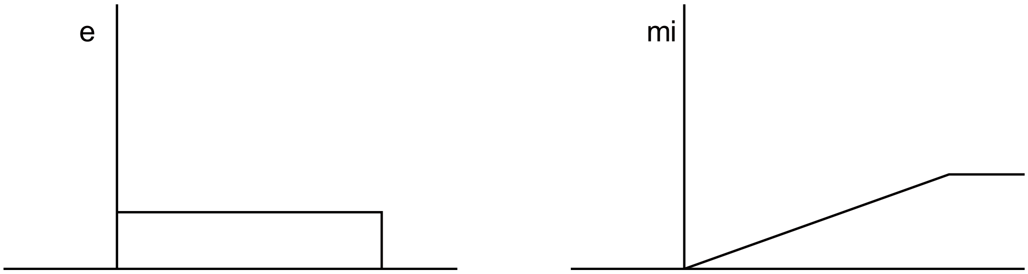
The larger the Kp value, the stronger the action of the proportional operation.

(2) Integral operation

Control operation that produces an output proportional to the integral time of the input.



$mi = 1/Ti \int e\,dt$

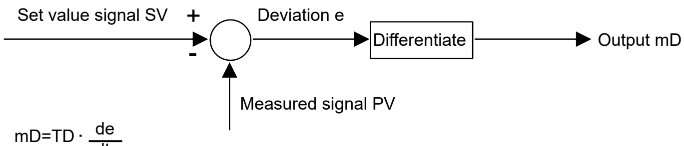


The resulting offset is removed by combining with proportional operation or proportional-derivative operation.

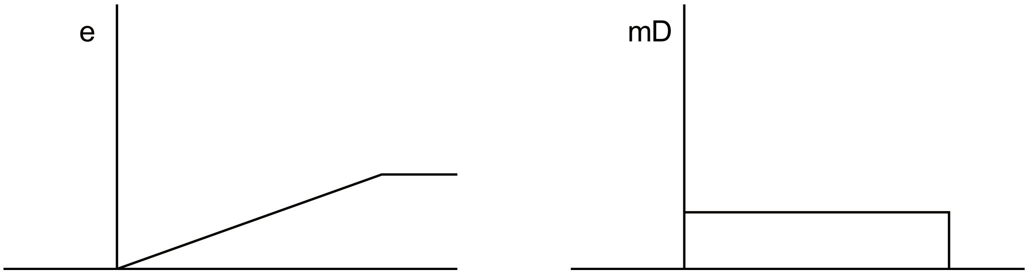
The smaller the Ti value, the stronger the action of the integral operation.

(3) Derivative operation

Control operation that produces an output proportional to the time derivative value of the input.



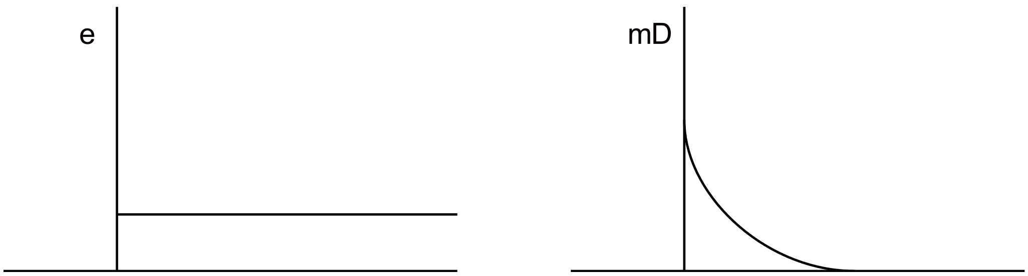
$mD=TD\cdot\frac{de}{dt}$



The advancing property of derivative operation reduces the negative effects that the delaying property of the process has on control.

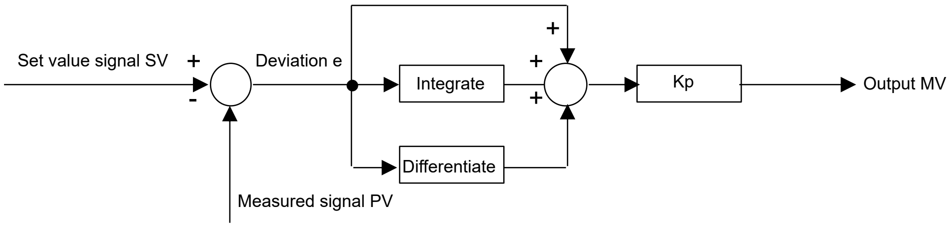
The larger the Td value, the stronger the action of the derivative operation.

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.

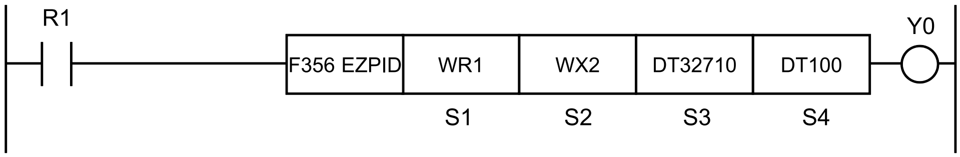
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the parameter setting value is out of range
	Turns ON when the area is exceeded in index modification.

28.2 F356 EZPID (PID Operation: PWM Output Possible)

Temperature control (PID) can be easily performed using the image of a temperature controller.

■ Instruction format



■ Operands

Items	Settings
S1	Control data
S2	Measured process value (PV)
S3	Starting No. of area storing PID control parameters
S4	Starting No. of calculation work area

■ Devices that can be specified (indicated by ●)

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	SWR	SDT	Constant				Index modifier	Integer Device
												K	H	M	f		
S1		●	●	●	●	●	●	●	●								
S2	●	●	●	●	●	●	●	●	●	●	●						
S3		●	●	●	●	●	●	●									
S4		●	●	●	●	●	●	●									

■ Outline of operation

- PID processing is performed to hold the measured process value (PV) at the set point value (SP).
  Writing the OUT instruction immediately after this instruction enables the PWM output (ON-OFF output) similar to a temperature controller.
  An auto-tuning function is also available to calculate the PID control parameters automatically.
  It can also be used with analog output as it outputs numerical values as well as PWM output.

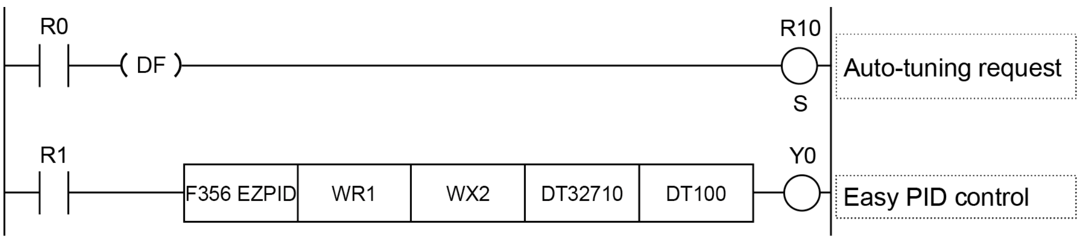
■ General explanation of the memory areas used

S1	Starts auto-tuning of the control data (one word) and reports its completion. Specifying a non-hold type area (e.g. WR) is recommended to allow operation on a per-bit basis.	
	When bit 0 is 1	Auto-tuning request. This instruction resets the bit if auto-tuning is completed. Reset this bit to cancel auto-tuning.
	When bit 0 is 0	PID control
	Bit 1	When auto-tuning has completed successfully, 1 is set.

	Bit 2	Turn this bit ON to hold the output MV (S4) when the execution condition of this instruction changes from OFF to ON. When this bit is OFF, MV is cleared.
	When bit 3 is 0	Specifies PWM output
	When bit 3 is 1	Specifies analog output
	When bit 4 is 0	The maximum value and minimum value of the internal output are +20% and -20% of the output range (output upper limit value - output lower limit value) respectively.
	When bit 4 is 1	The maximum value and minimum value of the internal output are the output upper limit value and output lower limit value respectively. *The output lower limit value is specified by S4+1, and the output upper limit value is specified by S4+2.
	Bits 5 to F	Reserved bits. Normally use 0.
S2	Area storing the measured process value (PV) (one word) The input WXn of a temperature input unit can be directly specified. Effective range: K-30000 to K+30000	
S3	Area to specify the target value (SP) and control parameters. (Four words) It is recommended that this area is allocated to hold-type operation memory.	
	S3	Stores the set point value (SP). Must be set from the instruction or a display. Setting range: K-30000 to K+30000
	S3+1	Stores the proportional gain (KP). Actual gain is set point value × 0.1. Automatically set after auto-tuning is completed. Setting range: K1 to K9999 (0.1 to 999.9)
	S3+2	Stores the integral time (TI) Actual integral time is set point value × 0.1. Automatically set after auto-tuning is completed. Setting range: K0 to K30000 (0 to 3000 s)
	S3+3	Stores the derivative time (TD). Actual derivative time is set point value × 0.1. Automatically set after auto-tuning is completed. Setting range: K0 to K10000 (0 to 1000 s)
S4	Divided into output (MV), specified area of control mode, auto-tuning related area, and operation work area. The area in the range of S4 to S4+29 is necessary for the instruction. (See below for details.) It is recommended to allocate it in the non-hold area. Also, do not use the data in this area for other purposes.	

■ Easy usage

<PWM output in reverse operation (heating)>



- Specify the set point value (SP) with the instruction or a display before the operation.
- If auto-tuning is requested with a device such as a display, the above auto-tuning request program is not necessary.
- Work areas DT100 to DT129 return to the default value when R1 turns on. (However, only DT100 (MV) can be held.)
- The control conditions are as follows: operation cycle 1 s, derivative-type reverse operation (heating), PWM resolution = 1000.
- PID control starts from the next scan, and PWM output is executed for Y0.
- Program as described above to start auto-tuning with the instruction, and turn ON R1 after turning ON R0.
- When auto-tuning has completed successfully, R11 turns ON and KP, TI, and TD are set.
- After that, if R1 is ON continuously, it will change to PID control automatically, and PWM output will be executed for Y0.

Note

- If execution condition R1 has turned OFF during PID control, PWM output Y0 also turns OFF. However, the output manipulated value MV is held.

■ When changing control conditions

- The area S4+1 to S4+9 must be changed to change control conditions. Change it before the second execution of the F356 EZPID instruction.

<Details of S4>

S4: Divided into output (MV), specified area of control mode, auto-tuning related area, and operation work area. It is recommended to allocate it in the non-hold area. Also, do not use the data in this area for other purposes.

Output (MV) and control mode area (Used with the normal default values.)

<table>
<tr><th>Memory</th><th>Function</th><th>Default</th><th>Range:</th></tr>
<tr><td>S4</td><td>The output manipulated value (MV) of the calculation result is stored</td><td>K0</td><td>K-10000 to K10000</td></tr>
<tr><td>S4+1</td><td>Specify the lower limit of the output manipulated value (MV)</td><td>K0</td><td>Minimum K-10000</td></tr>
<tr><td>S4+2</td><td>Specify the upper limit of the manipulated value (MV)</td><td>K10000</td><td>Maximum K+10000</td></tr>
<tr><td>S4+3</td><td>Specify the 100% output band (range where PID control is not performed)</td><td>K0</td><td>K0 to K80 (%)</td></tr>
<tr><td>S4+4</td><td>Specify the control cycle (TS). Setting unit = 10 ms, default value = 1 s</td><td>K100</td><td>K1 to K3000
(0.01 to 30 s)</td></tr>
<tr><td>S4+5</td><td>Specify the control mode (see table below)
<table>
<tr><th colspan="2">Control mode</th><th>Value</th><th>Example</th></tr>
<tr><td rowspan="2">Derivative type</td><td>Reverse</td><td>K0</td><td>Heating</td></tr>
<tr><td>Forward</td><td>K1</td><td>Cooling</td></tr>
<tr><td rowspan="2">Proportional-derivative type</td><td>Reverse</td><td>K2</td><td>Heating</td></tr>
<tr><td>Forward</td><td>K3</td><td>Cooling</td></tr>
</table>
Reverse operation and forward operation
Reverse operation: If the measured process value drops, the output is increased (example: heating)</td><td>K0</td><td>K0 to K3</td></tr>
</table>

Memory	Function	Default	Range:
	Forward operation: If the measured process value increases, the output is increased (example: cooling) Derivative-type and proportional-derivative type Derivative type: Approaches the set point value faster, but is more likely to overshoot. Proportional-derivative type: Approaches the set point value slower, but is less likely to overshoot.		

Auto-tuning related area (Used with the normal default values.)

Memory	Function	Default	Range:
S4+6	Specify the bias value for performing auto-tuning.	K0	From K0
S4+7	Specify the correction data (a1) of the auto-tuning result (KP).	K125	K50 to K500%
S4+8	Specify the correction data (a2) of the auto-tuning result (TI).	K200	K50 to K500%
S4+9	Specify the correction data (a3) of the auto-tuning result (TD).	K100	K50 to K500%
S4+10	Stores the status while auto-tuning is being performed.	K0	K0 to K5

Operation work area

Memory	Function	Default	Range:
S4+11 to S4+29	The area up to S4+29 is the work area for the PID and auto-tuning operations.	0	

(Note 1) The default value is written when the execution condition turns on.

The output manipulated value (MV) is output only within the range of the upper limit value and lower limit value.

Configure the settings so that -10000 ≤ lower limit value < upper limit value ≤ 10000.

■ How to output PWM

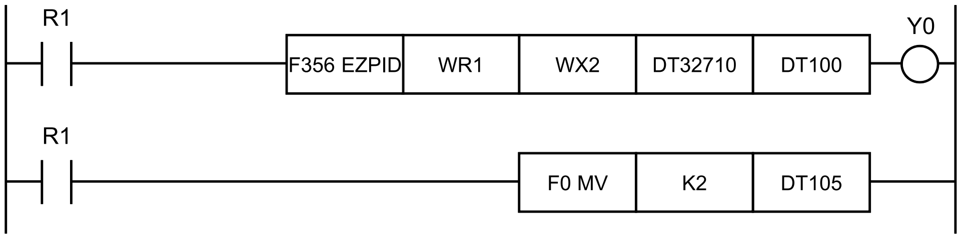
- The PWM output cycle is determined by the value set for S4+4. The default value is a cycle of 1 s. The PWM duty cycle is determined by what percentage of K0 to K10000 is comprised of the output MV (S4).
- When either the lower or upper limit value of output MV, specified by S4+1 and S4+2, is a negative value, the PMW output is always OFF.
- The PWM output is always OFF when the output MV is K0, and it is always ON when the output MV is K10000.

■ Explanation of specific usage

1. Only changing control mode with PWM output

- Change the content of the control mode (S4+5) to K1 to K3, using an instruction such as F0 MV.

Example: Change the control mode from the default = derivative type to the proportional-derivative type.



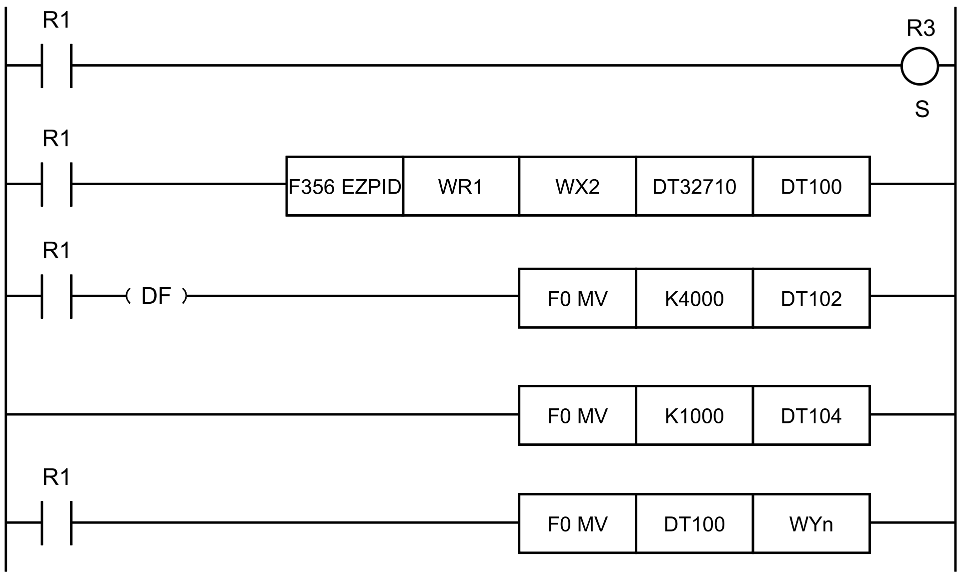
2. Using an analog output unit for output

1. Set the analog output flag (bit 3 of S1) to 1.
2. Set the output lower limit value (S4+1) and the output upper limit value (S4+2) according to the output range of the analog output unit.
   e.g. <Lower limit value=K0, upper limit value=K2000>, <lower limit value=K0, upper limit value=K4000>
3. Control cycle (TS): Change the value of (S4+4) according to the input update cycle of the temperature input unit (normally 0.1 s or more)
   e.g. TS=K10 (100 ms)
4. Change the control mode if necessary.
5. Transmit the output manipulated value (MV) to WY on the analog output unit.

Note

- When analog output is used for the output, it is not necessary to write an OUT instruction immediately after this instruction.
  Also, when using analog output, PWM output is fixed to OFF.

Example: Control with the output upper limit value (S4+2) set to K4000 and the control cycle (S4+4) set to 10 s



■ More details on setting methods

1. Setting the 100% output band (S4+3)

The 100% output band specifies the percentage of the set value for the measured process value (PV) to be above when PID control is started.

100% output is performed in the area up to the specified process value.

If the measured process value (PV) is less than the set point value (SP) × this setting, it has the effect of shortening the time to reach the set point value (SP), during which 100% output is performed.

For example, if this setting is set to K80, 100% output is performed up to 80% of the set point value (SP), and PID control starts from there.

If this setting has K0=the default value, PID control is performed from the beginning.

2. Fine adjustment of auto-tuning

1. Correction of auto-tuning results (S4+7, S4+8, and S4+9)

   When auto-tuning has completed, the parameters KP, TI, and TD are stored in (S3+1, S3+2, and S3+3). The result can be corrected with these parameters at this time.

   e.g. To correct KP to 2 times its value, set S4+7 to K200 (meaning 200%) and perform auto-tuning.

   To correct TI to 1.25 times its value, set S4+8 to K125 (meaning 125%) and perform auto-tuning.

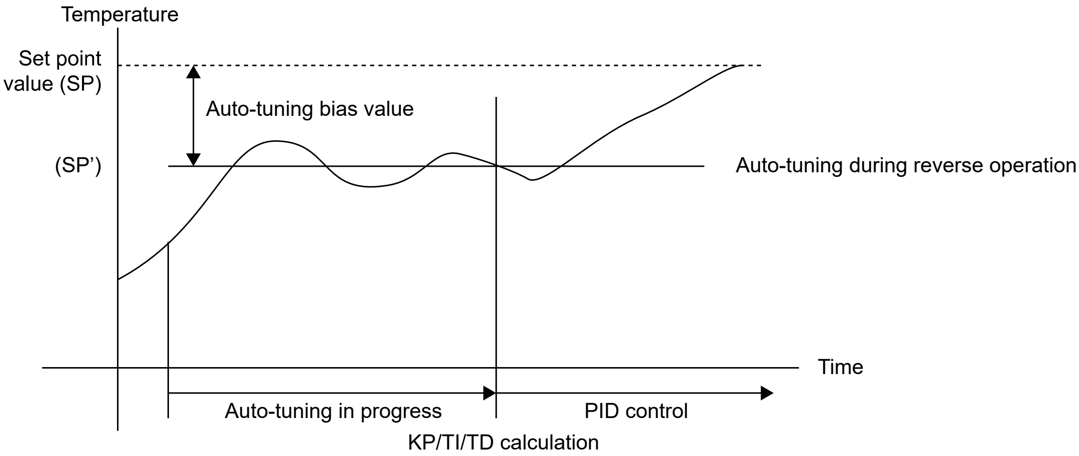
   To correct TD to 0.75 times its value, set S4+9 to K75 (meaning 75%) and perform auto-tuning.

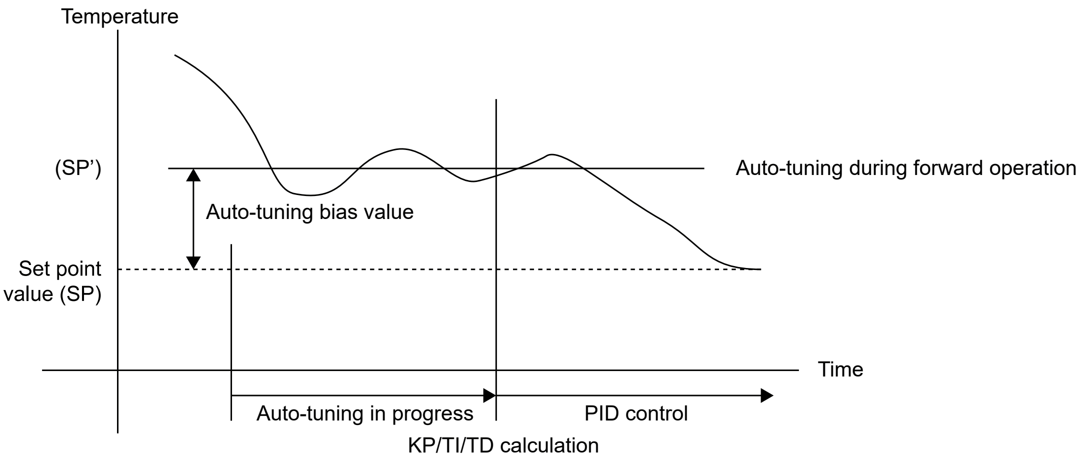
2. Auto-tuning bias value (S4+6)

   Auto-tuning is executed with the set point value (SP’) as [set point value (SP) - auto-tuning bias value].

   This is used to control excessive temperature rise while auto-tuning is performed.

   For the forward operation, auto-tuning is executed with the value set to [set point value (SP) + this set value].

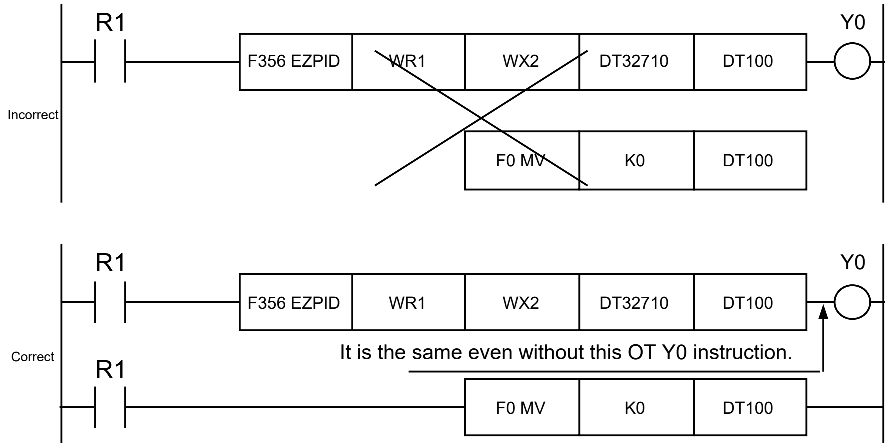




(Note 1) Even if auto-tuning is started when the measured process value (PV) is close to the set point value (SP), auto-tuning is performed with the above SP'.

■ Precautions for programming

- When the execution condition turns on, the area S4 to S4+29 is initialized.
  If the values are set to non-default values, write using the always-ON relay R9010 as the execution condition, with an instruction such as the F0 MV instruction.
- The PID operation instruction always calculates the operation cycle and PWM output timing internally, so be sure to perform only one operation during a single scan. Additionally, do not attempt to execute it during a subroutine or interrupt program. This instruction cannot be written more than once with the same operand specified.
- Do not turn OFF the execution condition during PID processing. 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.
- After executing this instruction, the execution conditions will change. This means that subsequent instructions will not work correctly in the program shown below.



■ Conditions when operation errors occur

- When the following parameters are out of the setting range: S2: measured process value (PV), S3: set point value (SP), S3+1: KP, S3+2: TI, S3+3: TD, S4+1 to S4+9
- When the area specified by S3 or S4 exceeds the upper limit of the specified operation device

■ Internal operation specifications

- When the execution condition turns on, the operation work is initialized.
- If the parameters KP, TI, and TD are all 0 when PID operation starts, they are initialized at 1, 0, and 0 respectively, and the operation is continued.
- At the rising edge of the AT signal, the AT successful completion flag and AT completion code are cleared.
- The AT set value operates with <set point value (SP) - bias value> as the target value. The default bias value is 0.
- When AT successfully completes, it stores the result obtained by multiplying the calculation results KP, TI, and TD by correction data a1, a2, and a3. The default value is 100%.
- When AT successfully completes, the AT successful completion flag is set, and the AT completion code is stored in AT step.
- If AT terminates abnormally, the parameters KP, TI, and TD are unchanged.
- PWM output is output at the duty cycle when the MV output range is 0 to 10000.
- For analog output (when bit 3 of S1 is 1), the internal calculated value is output in the range 0 to 10000 and converted to the specified range.
- Conversion formula: (upper limit value - lower limit value) × internal calculated value / 10000 + lower limit value

  e.g. When upper limit value = 40000, lower limit value = 0, and internal calculated value = 5000: output manipulated value MV = 2000

■ Precautions when using MV holding function

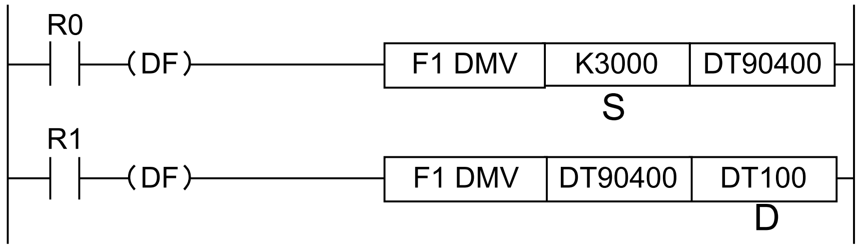
- When using the MV holding function, use the default upper and lower limit values.

29 Positioning Control Instructions

29.1 F1 DMV (Elapsed Value Write/Read)

Writes and reads the elapsed value of the high-speed counter.

■ Instruction format



■ Operands

Operand	Setting
S	When setting: Area storing the elapsed value (32-bit) set in the high-speed counter or constant data K-2,147,483,648 to K2,147,483,647
D	When reading: Area reading the elapsed value of the high-speed counter

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S	●	●	●	●	●	●	●	●	●	●	●	●
D	-	●	●	●	●	●	●	●	●	-	-	●

■ Outline of operation (Reading elapsed value)

- Reads the content of the special data register storing the elapsed value of the high-speed counter and writes to the area specified by [D].

■ Outline of operation (Setting elapsed value)

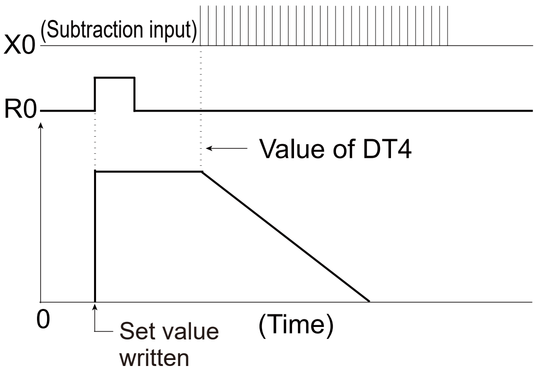
- At the same time as writing the value to the elapsed value area of the high-speed counter which uses 32-bit data specified by [S], sets it in the elapsed value area of the high-speed counter used within the system.

■ Precautions for programming

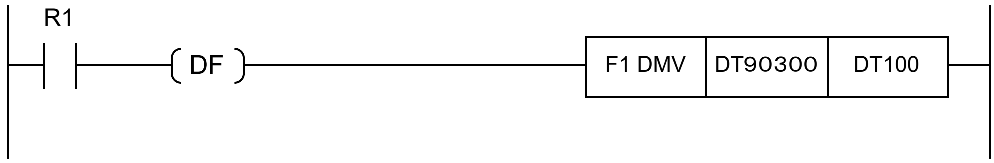
- Only F1 (DMV) instruction can perform the writing. The writing cannot be performed by other high-level instructions such as transfer instruction F0 (MV) and arithmetic instructions.
- Specify the memory area of [S] or [D] with the memory area number for the lower 16 bits.

■ Program example

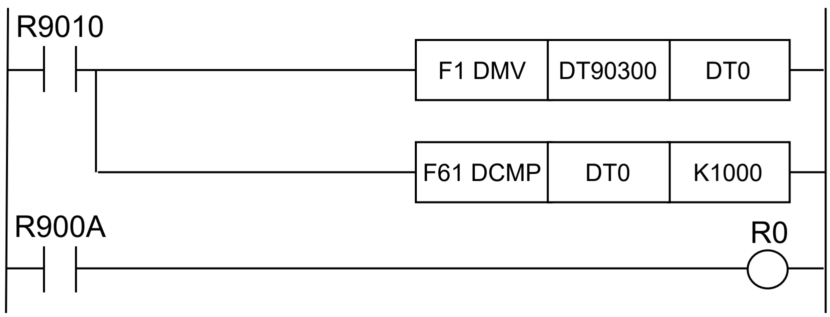
<Example 1> Set the value of the data register DT4 as the set value in the elapsed value area of CH0 with the input of R0.



<Example 2> Store the elapsed value of CH0 in the data register DT100 with the input of R1.



<Example 3> When the elapsed value of CH0 is greater than "K1000", the internal relay R0 is turns ON.



■ Table of correspondence between channel No. and elapsed value area

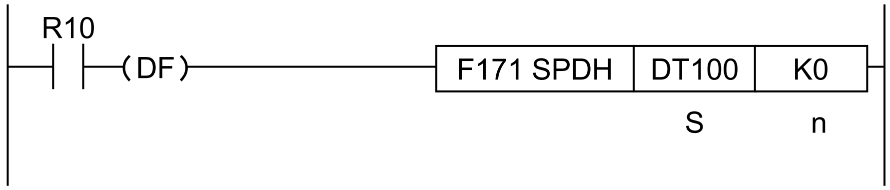
High-speed counter channel No.	Pulse output channel No.	Elapsed value area
ch0	–	DT90300 to DT90301

High-speed counter channel No.	Pulse output channel No.	Elapsed value area
ch1	–	DT90304 to DT90305
ch2	–	DT90308 to DT90309
ch3	–	DT90312 to DT90313
ch4	–	DT90316 to DT90317
ch5	–	DT90320 to DT90323
–	ch0	DT90400 to DT90401
–	ch1	DT90410 to DT90411
–	ch2	DT90420 to DT90421
–	ch3	DT90430 to DT90431

29.2 F171 SPDH [Pulse Output (Trapezoidal Control)]

Outputs pulses from a specified pulse output channel according to specified parameters.

■ Instruction format



■ Operands

Operand	Setting
S	Starting number of the area in which the data table is registered
n	Target channel for pulse output (n: 0 to 3)

■ Memory area type that can be specified

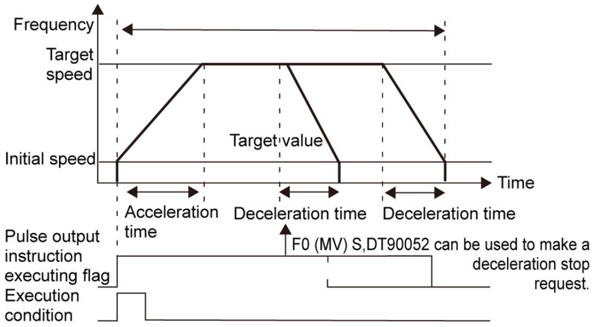
Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S	-	-	-	-	-	-	●	-	-	-	-	●
n	-	-	-	-	-	-	-	-	-	●	●	-

■ Outline of operation

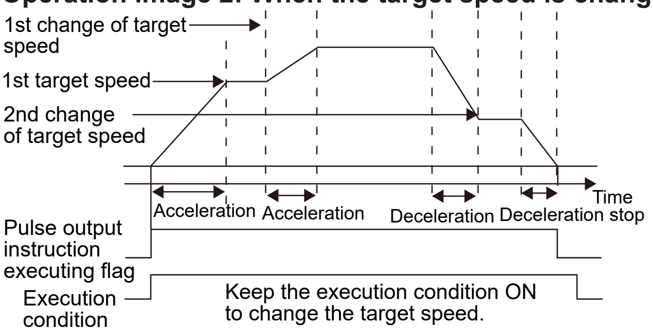
- This instruction outputs pulses from a specified channel to perform trapezoidal control when the execution condition is ON.
- The control code, initial speed, target speed, acceleration/deceleration time, and target value are specified by creating a data table [S] to [S+11] by using a user program.
- At the time of acceleration, the frequency is switched from the initial speed to the target speed in the specified acceleration time.
- A the time of deceleration, the frequency is switched from the target speed in the specified deceleration time.
- The control data (bit 5) of DT90052 can be used to make a deceleration stop request. (Example) F0 (MV) H120, DT90052
- If the contents of the table are the same as those at the previous startup, calculations can be omitted for faster startup.
- While pulses are being output, the pulse output instruction executing flag corresponding to the channel is ON.
- Regarding the acceleration/deceleration method and initial speed, if a deceleration stop request is made during acceleration, the deceleration is performed at the same slope as the deceleration time from the target speed. This instruction performs the processing with priority given to acceleration/deceleration time. Therefore, the initial speed may be corrected so that acceleration/deceleration can be performed in the specified time.
- The pulse output frequency can be changed by rewriting the target speed while pulses are being output. There are two types of control methods: type 0 and type 1. Type 0 allows the

speed to be changed in the range of the target speed specified at first. Type 1 allows acceleration/deceleration in the range up to the maximum frequency, regardless of the first target speed.

Operation image 1: When the target speed is not changed



Operation image 2: When the target speed is changed



■ Precautions for programming

- When describing the same channel in both the normal program and the interrupt program, be sure to program not to execute them simultaneously.
- This instruction cannot be executed when the pulse output instruction executing flag corresponding to the channel to be started is ON.
- If rewriting during RUN is performed, the pulse output stops.
- While the deceleration stop request flag is ON, this instruction cannot be started.
- To restart this instruction after stopping it, once turn OFF the execution condition and then turn it ON.
- To start this instruction in an interrupt program, specify "execution in interrupt program" with the control code. When executed in an interrupt program, this instruction cannot perform speed changes.

■ Pulse output channels and areas used

Channel No.	Output	Output method	
ch0	Y0	CW	PLS
	Y1	CCW	SIGN
ch1	Y2	CW	PLS
	Y3	CCW	SIGN

Channel No.	Output	Output method	
ch2	Y4	CW	PLS
	Y5	CCW	SIGN
ch3	Y6	CW	PLS
	Y7	CCW	SIGN

Channel No.	Pulse output instruction executing flag	Elapsed value area	Target value area
ch0	R9120	DT90400 DT90401	DT90402 DT90403
ch1	R9121	DT90410 DT90411	DT90412 DT90413
ch2	R9122	DT90420 DT90421	DT90422 DT90423
ch3	R9123	DT90430 DT90431	DT90432 DT90433

Channel No.	Initial speed correction speed	Deceleration lower limit speed	Acceleration disabled area start position
ch0	DT90406	DT90407	DT90408 DT90409
ch1	DT90416	DT90417	DT90418 DT90419
ch2	DT90426	DT90427	DT90428 DT90429
ch3	DT90436	DT90437	DT90438 DT90439

■ Data table settings

Table of data 0

S S+1	Control code	
S+2 S+3	Initial speed (Hz)	Speed range (Frequency) (Hz) 1 Hz to 50 kHz [K1 to K50000 (Unit: Hz)]
S+4 S+5	Target speed (Hz)	
S+6 S+7	Acceleration time (ms)	Acceleration time to target speed Acceleration time range (ms) [K1 to K32760 (Unit: ms)]
S+8 S+9	Deceleration time (ms)	Deceleration time to target speed Deceleration time range (ms) [K1 to K32760 (Unit: ms)]
S+10 S+11	Target value (No. of pulses)	Target value range [K-2,147,483,648 to K2,147,483,647]

Table of data 1

S S+1	Control code	
S+2 S+3	Initial speed (Hz)	Speed range (Frequency) (Hz) 1 Hz to 50 kHz [K1 to K50000 (Unit: Hz)]
S+4 S+5	Target speed (Hz)	
S+6 S+7	Acceleration time (ms)	Acceleration time to max. speed 50 kHz Acceleration time range (ms) [K1 to K32760 (Unit: ms)]
S+8 S+9	Deceleration time (ms)	Deceleration time to max. speed 50 kHz Deceleration time range (ms) [K1 to K32760 (Unit: ms)]
S+10 S+11	Target value (No. of pulses)	Target value range [K-2,147,483,648 to K2,147,483,647]

Info.

When performing a speed change during startup, if a value greater than 50 kHz is specified, this instruction corrects the speed to 50 kHz.

- Note that the instruction has the following characteristics depending on the specified initial speed.
  1. When the initial speed is equal to or greater than 1 Hz and less than 46 Hz, the maximum frequency can be controlled up to approximately 10 kHz. At higher frequencies, the speed error becomes larger.

2. When the initial speed is equal to or greater than 46 Hz and less than 184 Hz, the maximum frequency can be controlled up to 50 kHz.
3. When the initial speed is equal to or greater than 184 Hz, the maximum frequency can be controlled up to 50 kHz. The error in speed is the smallest around 50 kHz.

- Speed change during pulse output
  1. In type 0 control, if a value greater than the target speed at startup is specified, the speed is corrected to the target speed at startup. In type 1 control, if a value greater than 50 kHz is specified for the target speed, the speed is corrected to 50 kHz.
  2. For acceleration, the speed cannot be increased if it exceeds the acceleration disabled area start position. For the acceleration disabled area start position, refer to the description of the corresponding special register.
  3. For deceleration, the speed can only be decreased up to the deceleration lower limit speed. For the deceleration lower limit speed, refer to the description of the corresponding special register.

■ Control code setting

(Specify with an H constant.)

H □□□□□□□□

10: Fixed

Control specification
0: Trapezoidal

Type specification
0: Type 0 1: Type 1

Interrupt execution specification
0: Execution in main program
1: Execution in interrupt program
(Execution condition is level type)

Output specification
0: Pulse output
1: Calculation only

Operation mode specification
0: Incremental
1: Absolute

Output type specification
0:CW/CCW
1: PLS+SIGN (Forward OFF/Reverse ON)
2: PLS+SIGN (Forward ON/Reverse OFF)

Info.

- About output specification

  If this instruction is started with "1: Calculation only" specified, no pulses are output. If this instruction is executed once for a channel, and then executed again for the same channel with the same parameters, it can start up faster. (This applies whether "Pulse output" or "Calculation only" is specified.)

  However, if parameters that are different from those of the previous execution are specified, it does not start up faster.

  * The same parameter means that all parameters except the output specification are the same.

■ Output type

Incremental <Relative value control>

Outputs the pulses set with the target value.

Selection/ Target value	CW/CCW	PLS+SIGN Forward OFF/Reverse ON	PLS+SIGN Forward ON/Reverse OFF	Elapsed value
Positive value	Pulse output from CW	Pulse output when direction output is OFF	Pulse output when direction output is ON	Addition
Negative value	Pulse output from CCW	Pulse output when direction output is ON	Pulse output when direction output is OFF	Subtraction

Absolute <Absolute value control>

Outputs the pulses of the difference between the set target value and the current value.

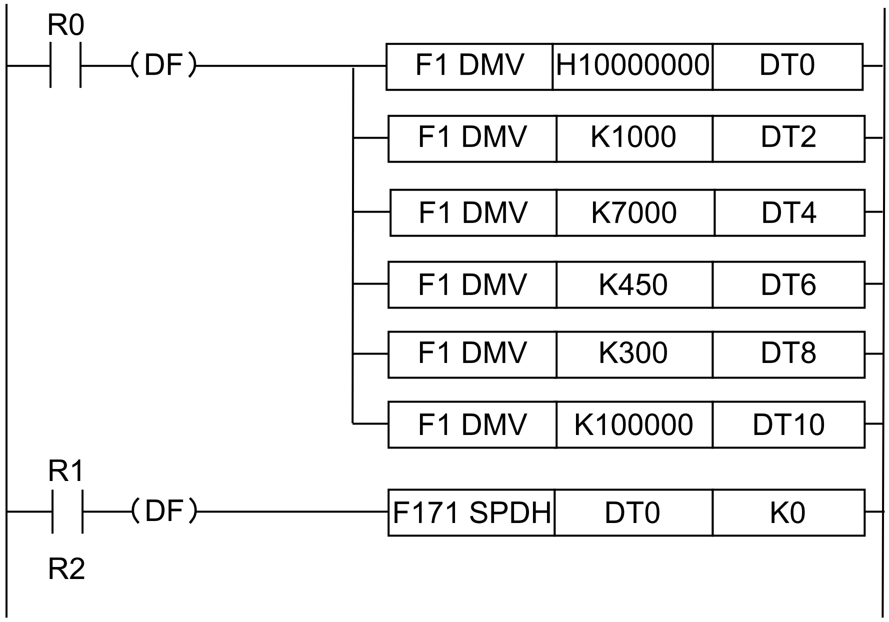
Selection/ Target value	CW/CCW	PLS+SIGN Forward OFF/Reverse ON	PLS+SIGN Forward ON/Reverse OFF	Elapsed value
When target value is larger than current value	Pulse output from CW	Pulse output when direction output is OFF	Pulse output when direction output is ON	Addition
When target value is smaller than current value	Pulse output from CCW	Pulse output when direction output is ON	Pulse output when direction output is OFF	Subtraction

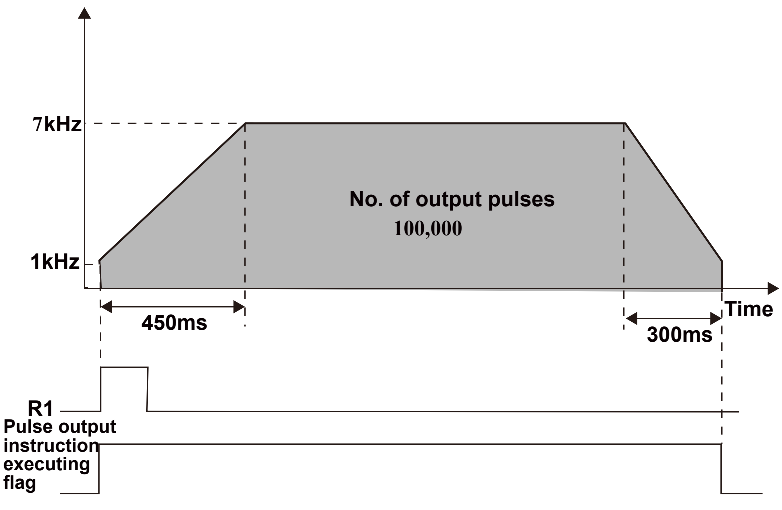
[Description of pulse output operation]

Pulses are output with duty fixed at 25%. When the PLS+SIGN output type is specified, pulse output starts approximately 300 μs after the direction signal is output (taking into account the characteristics of the motor driver).

■ Program example

<Example 1> Trapezoidal control type 0, no deceleration stop request, no speed change



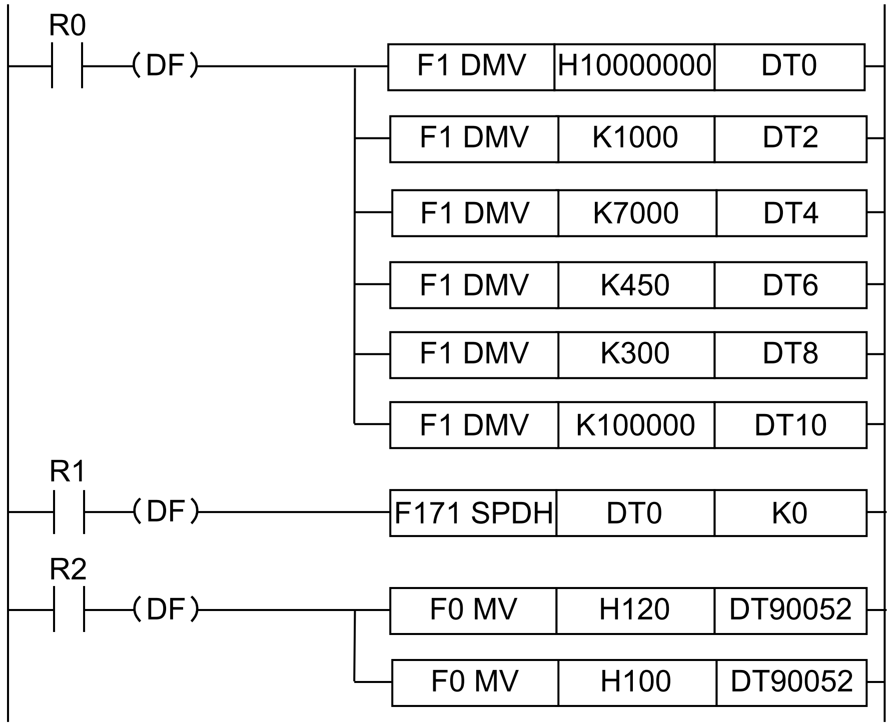


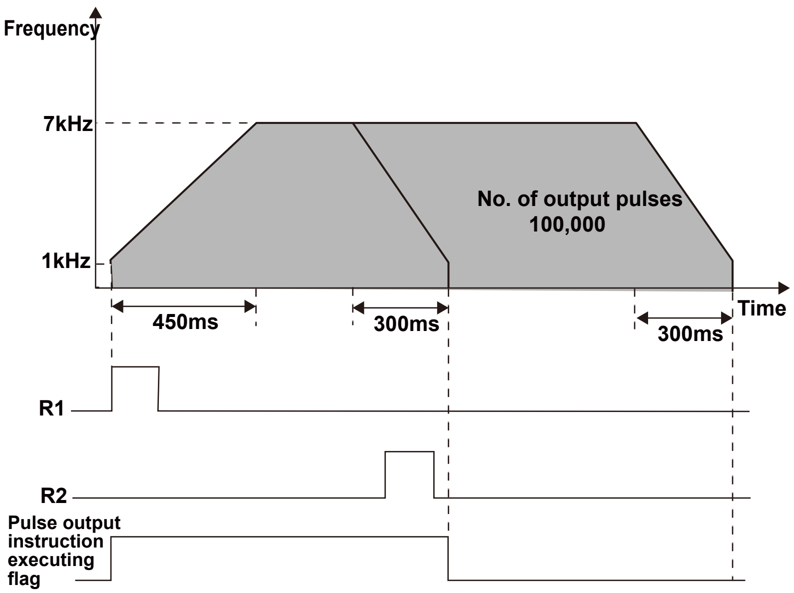
Data table

DT0	Control information	Trapezoidal control incremental CW/CCW
DT2	Initial speed (Hz)	1000 Hz
DT4	Target speed (Hz)	7000 Hz
DT6	Acceleration time (ms)	450 ms

DT8	Deceleration time (ms)	300 ms
DT10	Target value (No. of pulses)	100,000 pulses

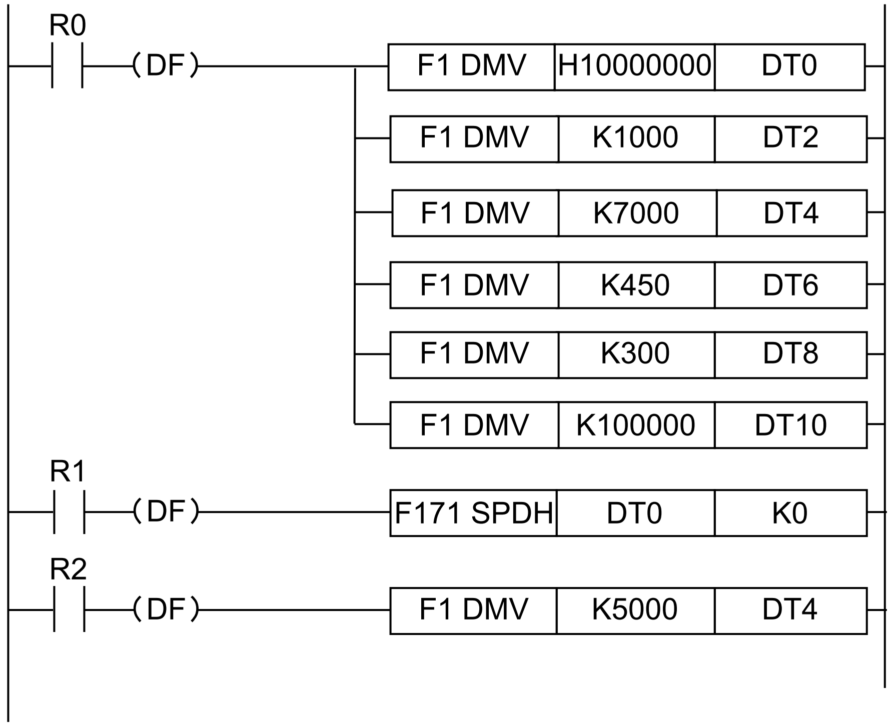
<Example 2> Trapezoidal control type 0, deceleration stop request, no speed change

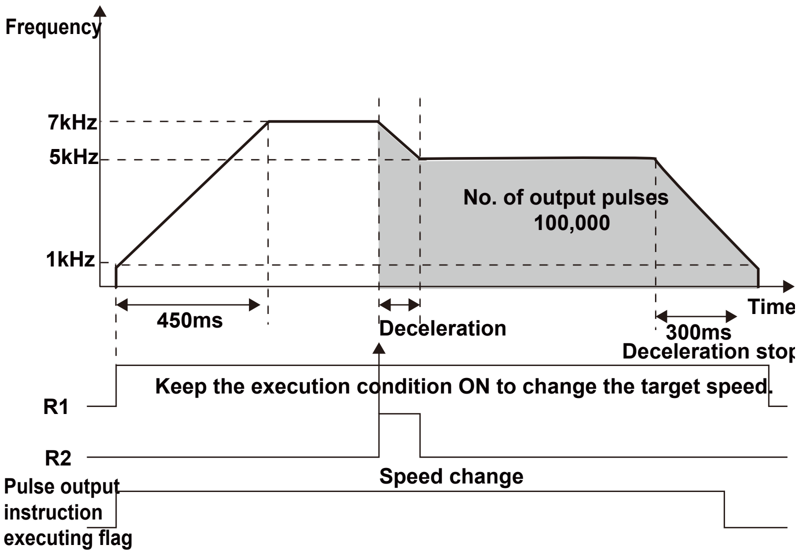




- Deceleration stop is performed according to the deceleration time after detection of a deceleration stop request.
- The data table is the same as in Program example 1.

<Example 3> Trapezoidal control type 0, speed change



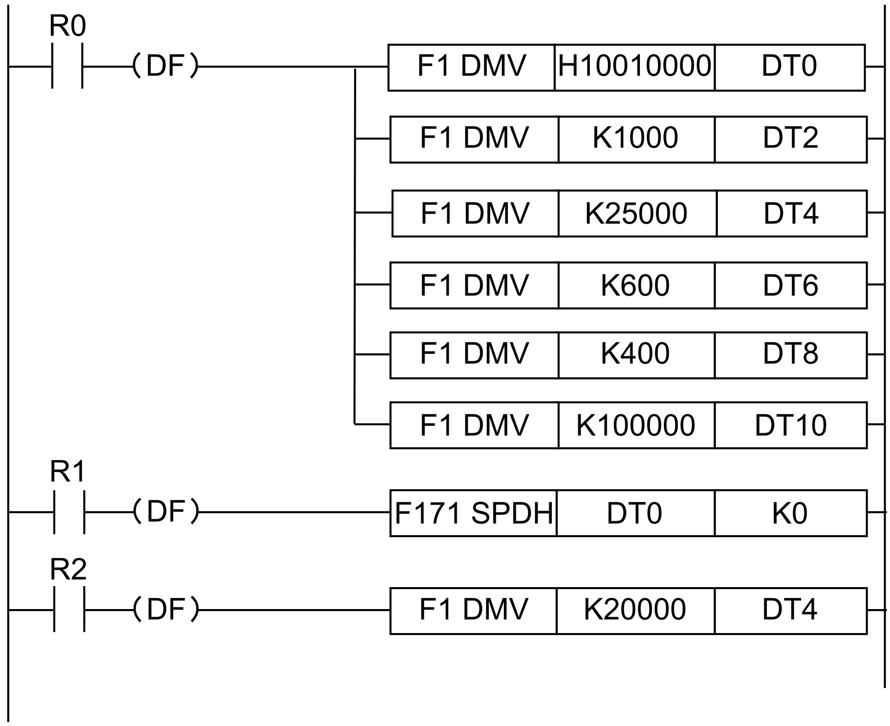


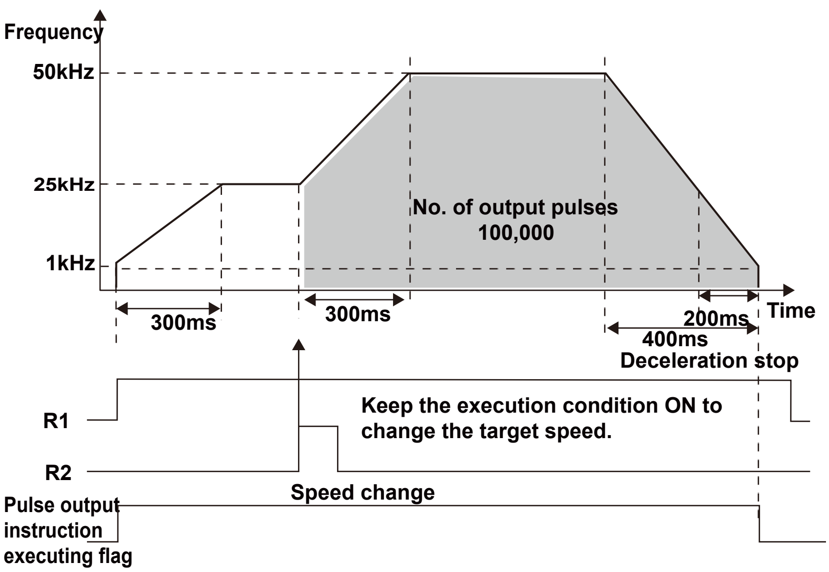
Data table

DT0	Control information	Trapezoidal control incremental CW/CCW
DT2	Initial speed (Hz)	1000 Hz

DT4	Target speed (Hz)	7000 Hz
DT6	Acceleration time (ms)	450 ms
DT8	Deceleration time (ms)	300 ms
DT10	Target value (No. of pulses)	100,000 pulses

<Example 4> Trapezoidal control type 1, speed change





Data table

DT0	Control information	Trapezoidal control incremental CW/CCW	—

DT2	Initial speed (Hz)	1000 Hz	—
DT4	Target speed (Hz)	25000 Hz	—
DT6	Acceleration time (ms)	600 ms	Acceleration time to 50 kHz
DT8	Deceleration time (ms)	400 ms	Deceleration time to 50 kHz
DT10	Target value (No. of pulses)	100,000 pulses	—

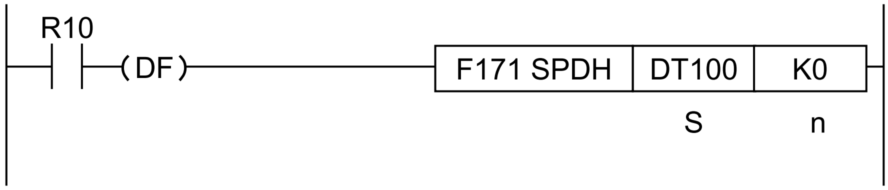
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when [n] is outside the specified range.
	Turns ON when each of [S, S+1] to [S+4, S+5] is outside the specified range.
	Turns ON when [S+2, S+3] is greater than [S+4, S+5].
	Turns ON when [S+10, S+11] is outside the specified range.
	Turns ON when the pulse output is not set in the system register.
	Turns ON when interrupt execution is specified in the main program.

29.3 F171 SPDH [Pulse Output (JOG Positioning Type 0)]

Outputs a specified number of pulses for deceleration stop after position control start input while pulses are being output.

■ Instruction format



■ Operands

Operand	Setting
S	Starting number of the area in which the data table is registered
n	Target channel for pulse output (n: 0 to 3)

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S	-	-	-	-	-	-	●	-	-	-	-	●
n	-	-	-	-	-	-	-	-	-	●	●	-

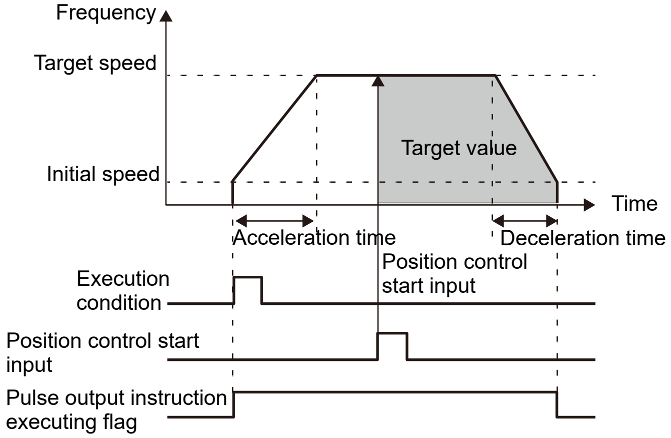
■ Outline of operation

- This instruction outputs pulses from a specified channel when the execution condition is ON.
- The control code, initial speed, target speed, acceleration/deceleration time, and target value after position control start input are specified by creating a data table [S] to [S+11] by using a user program.
- At the time of acceleration, the frequency is switched from the initial speed to the target speed in the specified acceleration time.
- After the target speed is reached, pulses are output continuously until position control start input turns ON.
- After position control start input turns ON, pulses are output for deceleration stop until the target value is reached.
- To use position control start input (X0, X1, X2, X3), set system register 402.
- A the time of deceleration, the frequency is switched from the target speed in the specified deceleration time.
- The control data in DT90052 (bit 6) can also be used to start positioning control.
  (Example) F0 (MV) H140, DT90052
- The control data (bit 5) of DT90052 can be used to make a deceleration stop request.
  (Example) F0 (MV) H120, DT90052
- If the contents of the table are the same as those at the previous startup, calculations can be omitted for faster startup.

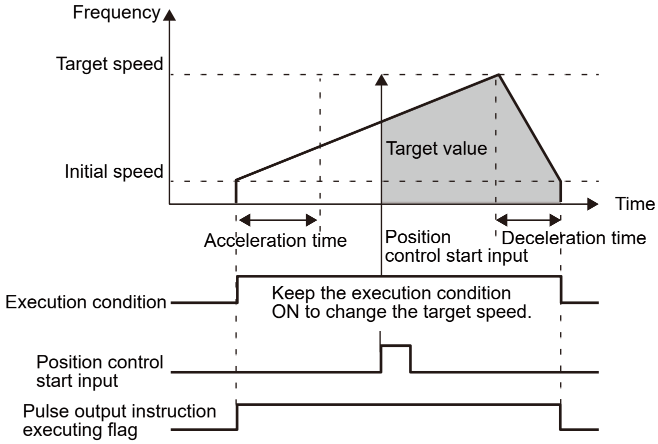
- While pulses are being output, the pulse output instruction executing flag corresponding to the channel is ON.
- About the acceleration/deceleration method and initial speed

  If a deceleration stop request is made during acceleration, the deceleration is performed at the same slope as the deceleration time from the target speed. This instruction performs the processing with priority given to acceleration/deceleration time. Therefore, the initial speed may be corrected so that acceleration/deceleration can be performed in the specified time.

Operation image: When the target speed is not changed



Operation image: When the target speed is changed



■ Precautions for programming

- When describing the same channel in both the normal program and the interrupt program, be sure to program not to execute them simultaneously.
- This instruction cannot be executed when the pulse output instruction executing flag corresponding to the channel to be started is ON.
- If rewriting during RUN is performed, the pulse output stops.
- Position control start input detects only the leading edge of the ON signal.
- While the deceleration stop request flag is ON, this instruction cannot be started.

- Note that the only methods that can be used to stop the pulses being output by this instruction are turning ON position control start input (position control start flag), deceleration stop request, and emergency stop.
- To restart this instruction after stopping it, once turn OFF the execution condition and then turn it ON.
- To start this instruction in an interrupt program, specify "execution in interrupt program" with the control code. When executed in an interrupt program, this instruction cannot perform speed changes.

■ Pulse output channels and areas used

Channel No.	Output	Output method		Position control start input
ch0	Y0	CW	PLS	X0 DT90052 bit6
	Y1	CCW	SIGN	
ch1	Y2	CW	PLS	X1 DT90052 bit6
	Y3	CCW	SIGN	
ch2	Y4	CW	PLS	X2 DT90052 bit6
	Y5	CCW	SIGN	
ch3	Y6	CW	PLS	X3 DT90052 bit6
	Y7	CCW	SIGN	

Channel No.	Pulse output instruction executing flag	Elapsed value area	Target value area
ch0	R9120	DT90400 DT90401	DT90402 DT90403
ch1	R9121	DT90440 DT90411	DT90412 DT90413
ch2	R9122	DT90420 DT90421	DT90422 DT90423
ch3	R9123	DT90430 DT90431	DT90432 DT90433

Channel No.	Initial speed correction speed	Deceleration lower limit speed	Acceleration disabled area start position
ch0	DT90406	DT90407	DT90408 DT90409
ch1	DT90416	DT90417	DT90418 DT90419
ch2	DT90426	DT90427	DT90428 DT90429
ch3	DT90436	DT90437	DT90438 DT90439

■ Data table settings

S S+1	Control code	
S+2 S+3	Initial speed (Hz)	Speed range (Frequency) (Hz) 1 Hz to 50 kHz [K1 to K50000 (Unit: Hz)]
S+4 S+5	Target speed (Hz)	
S+6 S+7	Acceleration time (ms)	Acceleration time to max. speed 50 kHz Acceleration time range (ms) [K1 to K32760 (Unit: ms)]
S+8 S+9	Deceleration time (ms)	Deceleration time to max. speed 50 kHz Deceleration time range (ms) [K1 to K32760 (Unit: ms)]
S+10 S+11	Target value (No. of pulses)	Target value range [K-2,147,483,648 to K2,147,483,647]

- Note that the instruction has the following characteristics depending on the specified initial speed.
  1. When the initial speed is equal to or greater than 1 Hz and less than 46 Hz, the maximum frequency can be controlled up to approximately 10 kHz. At higher frequencies, the speed error becomes larger.
  2. When the initial speed is equal to or greater than 46 Hz and less than 184 Hz, the maximum frequency can be controlled up to 50 kHz.
  3. When the initial speed is equal to or greater than 184 Hz, the maximum frequency can be controlled up to 50 kHz. The error in speed is the smallest around 50 kHz.
- Speed change during pulse output
  1. If a value greater than 50 kHz is specified for the target speed, the speed is corrected to 50 kHz.
  2. For acceleration, the speed cannot be increased if it exceeds the acceleration disabled area start position. For the acceleration disabled area start position, refer to the description of the corresponding special register.
  3. For deceleration, the speed can only be decreased up to the deceleration lower limit speed. For the deceleration lower limit speed, refer to the description of the corresponding special register.

■ Control code setting

(Specify with an H constant.)

H □□□□□□□□

10: Fixed

Control specification
1: JOG positioning

Control specification 2
0: Type 0

Interrupt execution specification
0: Execution in main program
1: Execution in interrupt program
(Execution condition is level type)

Output specification
0: Pulse output
1: Calculation only

Operation mode specification
0: Incremental

Output type specification
0:CW/CCW
1: PLS+SIGN (Forward OFF/Reverse ON)
2: PLS+SIGN (Forward ON/Reverse OFF)

- When 0 is specified for the target value, this instruction stops when position control start input turns ON. (V1.06 or later)

  To perform reverse output when 0 is specified for the target value, specify 4, 5, or 6, instead of 0, 1, or 2, as the output type of the control code.

Info.

- About output specification

  If this instruction is started with "1: Calculation only" specified, no pulses are output. If this instruction is executed once for a channel, and then executed again for the same channel with the same parameters, it can start up faster. (This applies whether "Pulse output" or "Calculation only" is specified.)

  However, if parameters that are different from those of the previous execution are specified, it does not start up faster.

  * The same parameter means that all parameters except the output specification are the same.

■ Output type

Selection/ Target value	CW/CCW	PLS+SIGN Forward OFF/Reverse ON	PLS+SIGN Forward ON/Reverse OFF	Elapsed value
Positive value	Pulse output from CW	Pulse output when direction output is OFF	Pulse output when direction output is ON	Addition
Negative value	Pulse output from CCW	Pulse output when direction output is ON	Pulse output when direction output is OFF	Subtraction

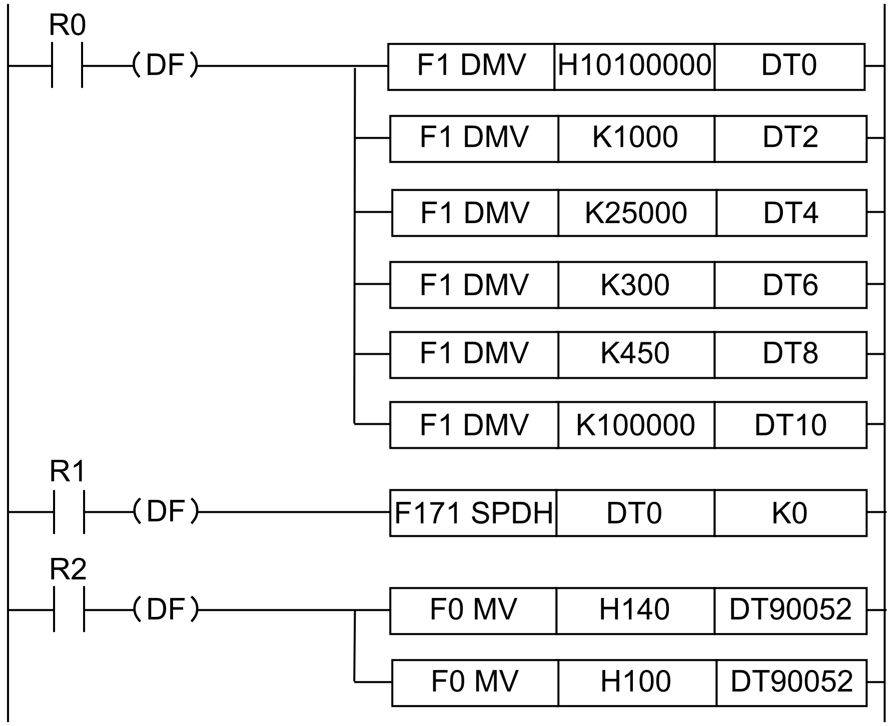
When 0 is specified for the target value, specifying 0, 1, or 2 for the output format results in forward output.

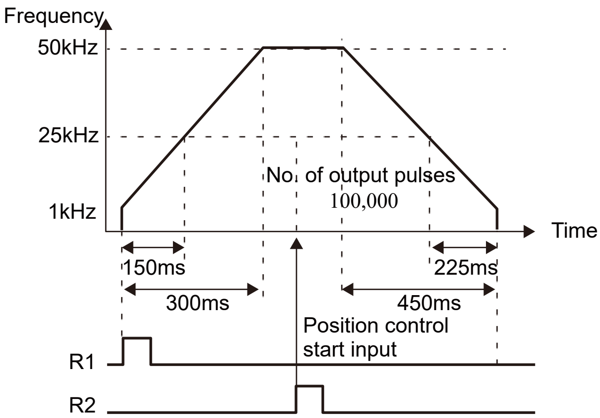
To perform reverse output, specify 4, 5, or 6, instead of 0, 1, or 2. (V1.06 or later)

[Description of pulse output operation]

Pulses are output with duty fixed at 25%. When the PLS+SIGN output type is specified, pulse output starts approximately 300 μs after the direction signal is output. (The characteristics of the motor driver are considered.)

■ Program example





■ Flag operations

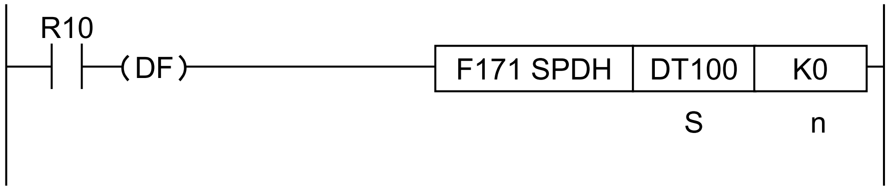
Name	Description
Hold error (R9007) Latest error (R9008)	Turns ON when the area is exceeded in index modification.
	Turns ON when [n] is outside the specified range.

Name	Description
	Turns ON when each of [S, S+1] to [S+4, S+5] is outside the specified range.
	Turns ON when [S+2, S+3] is greater than [S+4, S+5].
	Turns ON when [S+10, S+11] is outside the specified range.
	Turns ON when the pulse output is not set in the system register.
	Turns ON when interrupt execution is specified in the main program.

29.4 F171 SPDH [Pulse Output (JOG Positioning Type 1)]

Outputs a specified number of pulses for deceleration stop while changing the target speed again after position control start input while pulses are being output.

■ Instruction format



■ Operands

Operand	Setting
S	Starting number of the area in which the data table is registered
n	Target channel for pulse output (n: 0 to 3)

■ Memory area type that can be specified

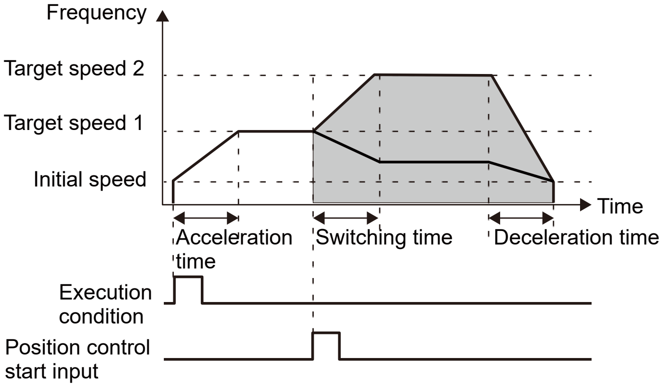
Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S	-	-	-	-	-	-	●	-	-	-	-	●
n	-	-	-	-	-	-	-	-	-	●	●	-

■ Outline of operation

- This instruction outputs pulses from a specified channel when the corresponding pulse output instruction executing flag is OFF and the execution condition is ON.
- The control code, initial speed, target speed 1, acceleration time, target speed 2 after position control start input, switching time, deceleration time, and target value are specified by creating a data table [S] to [S+15] by using a user program.
- At the time of acceleration, the frequency is switched from the initial speed to the target speed in the specified acceleration time.
- After the target speed is reached, pulses are output continuously until position control start input turns ON.
- After position control start input turns ON, pulses are output for deceleration stop until the target value is reached.
- To use position control start input (X0, X1, X2, X3), set system register 402.
- A the time of deceleration, the frequency is switched from the target speed in the specified deceleration time.
- The control data in DT90052 (bit 6) can also be used to start positioning control.
  (Example) F0 (MV) H140, DT90052
- The control data (bit 5) of DT90052 can be used to make a deceleration stop request.
  (Example) F0 (MV) H120, DT90052

- If the contents of the table are the same as those at the previous startup, calculations can be omitted for faster startup.
- About the acceleration/deceleration method and initial speed
  If a deceleration stop request is made during acceleration, the deceleration is performed at the same slope as the deceleration time from the target speed. This instruction performs the processing with priority given to acceleration/deceleration time. Therefore, the initial speed may be corrected so that acceleration/deceleration can be performed in the specified time.

Operation image



(Note 1) Be aware that position control start input is ignored even if it turns ON during the acceleration time period.

■ Precautions for programming

- When describing the same channel in both the normal program and the interrupt program, be sure to program not to execute them simultaneously.
- This instruction cannot be executed when the pulse output instruction executing flag corresponding to the channel to be started is ON.
- If rewriting during RUN is performed while pulses are being output, more pulses than the setting may be output.
- Position control start input detects only the leading edge of the ON signal.
- While the deceleration stop request flag is ON, this instruction cannot be started.
- Note that the only methods that can be used to stop the pulses being output by this instruction are turning ON position control start input (position control start flag), deceleration stop, and emergency stop.
- The target speed cannot be changed with this instruction.
- To start this instruction in an interrupt program, specify "execution in interrupt program" with the control code.

■ Pulse output channels and areas used

Channel No.	Output	Output method		Position control start input
ch0	Y0	CW	PLS	X0 DT90052 bit6
	Y1	CCW	SIGN	
ch1	Y2	CW	PLS	X1 DT90052
	Y3	CCW	SIGN	

Channel No.	Output	Output method		Position control start input
				bit6
ch2	Y4	CW	PLS	X2 DT90052 bit6
	Y5	CCW	SIGN	
ch3	Y6	CW	PLS	X3 DT90052 bit6
	Y7	CCW	SIGN	

Channel No.	Pulse output instruction executing flag	Elapsed value area	Target value area
ch0	R9120	DT90400 DT90401	DT90402 DT90403
ch1	R9121	DT90440 DT90411	DT90412 DT90413
ch2	R9122	DT90420 DT90421	DT90422 DT90423
ch3	R9123	DT90430 DT90431	DT90432 DT90433

■ Data table settings

S S+1	Control code	
S+2 S+3	Initial speed (Hz)	Speed range (Frequency) (Hz) 1 Hz to 50 kHz [K1 to K50000 (Unit: Hz)]
S+4 S+5	Target speed 1 (Hz)	
S+6 S+7	Acceleration time (ms)	Acceleration/deceleration time range (ms) [K1 to K32760 (Unit: ms)]
S+8 S+9	Target speed 2 (Hz)	Speed range (Frequency) (Hz) 1 Hz to 50 kHz [K1 to K50000 (Unit: Hz)]
S+10 S+11	Switching time (ms)	[K1 to K32760 (Unit: ms)]
S+12 S+13	Deceleration time (ms)	[K1 to K32760 (Unit: ms)]
S+14 S+15	Target value (No. of pulses)	Target value range [K-2,147,483,648 to K2,147,483,647]

- Note that the instruction has the following characteristics depending on the specified initial speed.
  1. When the initial speed is equal to or greater than 1 Hz and less than 46 Hz, the maximum frequency can be controlled up to approximately 10 kHz. At higher frequencies, the speed error becomes larger.
  2. When the initial speed is equal to or greater than 46 Hz and less than 184 Hz, the maximum frequency can be controlled up to 50 kHz.
  3. When the initial speed is equal to or greater than 184 Hz, the maximum frequency can be controlled up to 50 kHz. The error in speed is the smallest around 50 kHz.

■ Control code setting

(Specify with an H constant.)

H □□□□□□□□

10: Fixed

Control specification
1: JOG positioning

Control specification 2
0: Type 1

Interrupt execution specification
0: Execution in main program
1: Execution in interrupt program
(Execution condition is level type)

Output specification
0: Pulse output
1: Calculation only

Operation mode specification
0: Incremental

Output type specification
0:CW/CCW
1: PLS+SIGN (Forward OFF/Reverse ON)
2: PLS+SIGN (Forward ON/Reverse OFF)

Info.

- About output specification

  If this instruction is started with "1: Calculation only" specified, no pulses are output.

  If this instruction is executed once for a channel, and then executed again for the same channel with the same parameters, it can start up faster. (This applies whether "Pulse output" or "Calculation only" is specified.)

  However, if parameters that are different from those of the previous execution are specified, it does not start up faster.

  * The same parameter means that all parameters except the output specification are the same.

■ Output type

Incremental <Relative value control>

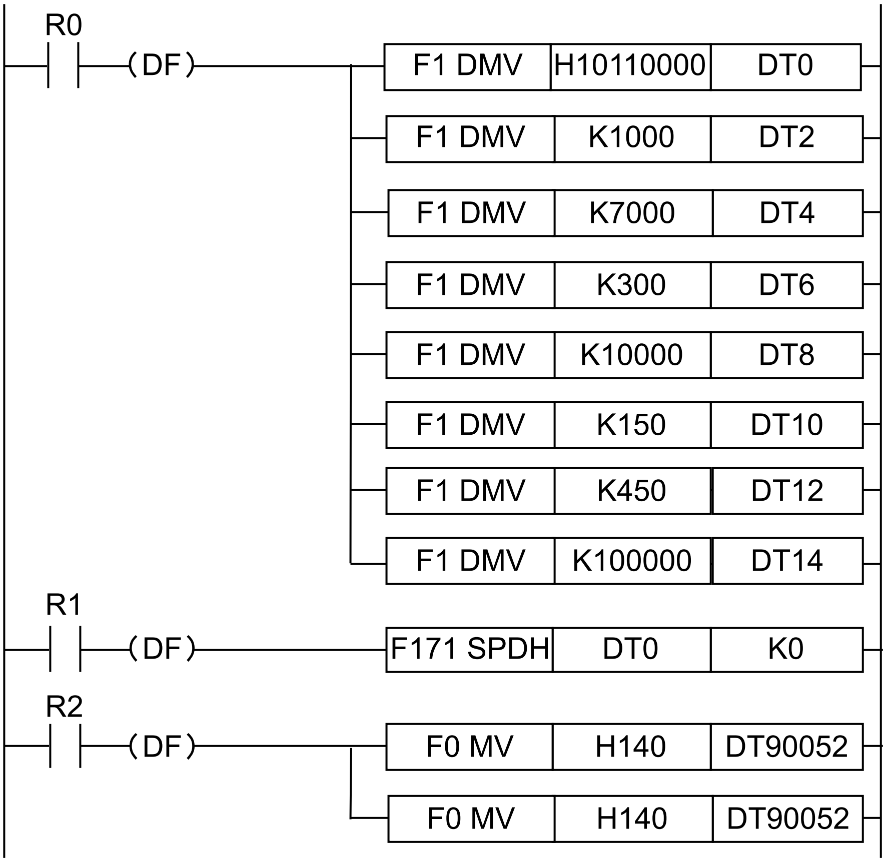
Outputs the pulses set with the target value.

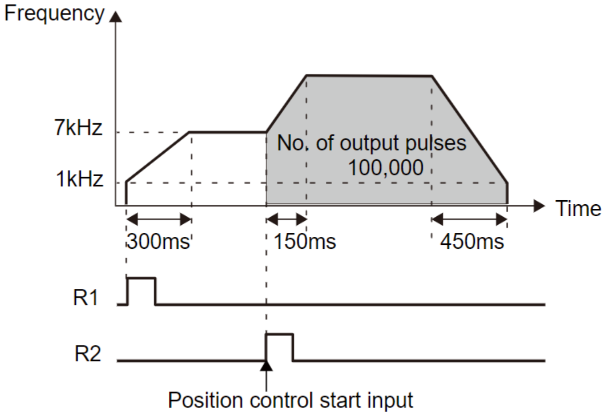
Selection/ Target value	**CW/CCW**	**PLS+SIGN** **Forward OFF/Reverse ON**	**PLS+SIGN** **Forward ON/Reverse OFF**	**Elapsed value**
Positive value	Pulse output from CW	Pulse output when direction output is OFF	Pulse output when direction output is ON	Addition
Negative value	Pulse output from CCW	Pulse output when direction output is ON	Pulse output when direction output is OFF	Subtraction

[Description of pulse output operation]

Pulses are output with duty fixed at 25%. When the PLS+SIGN output type is specified, pulse output starts approximately 300 μs after the direction signal is output. (The characteristics of the motor driver are considered.)

■ Program example





■ Flag operations

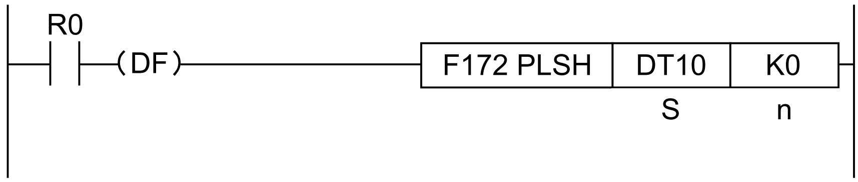
Name	Description
Hold error (R9007) Latest error (R9008)	Turns ON when the area is exceeded in index modification.
	Turns ON when [n] is outside the specified range.

Name	Description
	Turns ON when each of [S, S+1] to [S+4, S+5] is outside the specified range.
	Turns ON when [S+8, S+9] is outside the specified range.
	Turns ON when [S+2, S+3] is greater than [S+4, S+5].
	Turns ON when [S+2, S+3] is greater than [S+8, S+9].
	Turns ON when [S+14, S+15] is outside the specified range.
	Turns ON when the pulse output is not set in the system register.

29.5 F172 PLSH [Pulse Output (JOG Operation Type 0/1)]

Outputs pulses according to specified parameters from a specified pulse output channel.

■ Instruction format



■ Operands

Operand	Setting
S	Starting number of the area in which the data table is registered
n	Target channel for pulse output

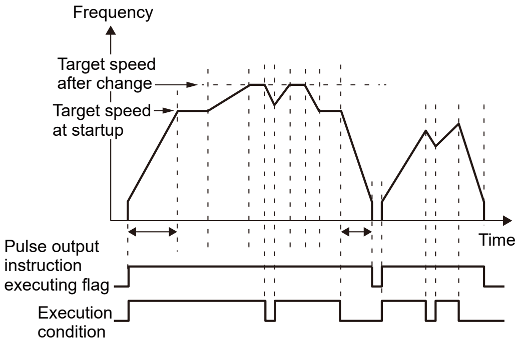
■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S	-	-	-	-	-	-	●	-	-	-	-	●
n	-	-	-	-	-	-	-	-	-	●	●	-

■ Outline of operation

- This instruction outputs pulses from a specified channel when the corresponding pulse output instruction executing flag is OFF and the execution condition is ON.
- For JOG operation type 0, the control code, initial speed, target speed, acceleration/deceleration time, and target value are specified by creating a data table [S] to [S+9] by using a user program.

  For JOG operation type 1 (with target value), in addition to the above, specify target values in [S+10] to [S+11].
- At the time of acceleration, the frequency is switched from the initial speed to the target speed in the specified acceleration time.
- When the execution condition turns OFF after the instruction is started, this instruction performs deceleration stop.
- A the time of deceleration, the frequency is switched from the target speed in the specified deceleration time.
- If the execution condition turns ON during deceleration, this instruction switches from deceleration to acceleration again.
- Operation image



■ Data table settings

S S+1	Control code	
S+2 S+3	Initial speed (Hz)	Speed range (Frequency) (Hz) 1 Hz to 50 kHz [K1 to K50000 (Unit: Hz)]
S+4 S+5	Target speed (Hz)	Acceleration time to max. speed 50 kHz Acceleration time range (ms) [K1 to K32760 (Unit: ms)]
S+6 S+7	Acceleration time (ms)	Deceleration time from max. speed 50 kHz Deceleration time range (ms) [K1 to K32760 (Unit: ms)]
S+8 S+9	Deceleration time (ms)	Target value range Note: Valid only for JOG type 1 (with target value) K-2,147,483,648 to K2,147,483,647
	Target value	Valid only for JOG operation type 1

- Note that the instruction has the following characteristics depending on the specified initial speed.
  1. When the initial speed is equal to or greater than 1 Hz and less than 46 Hz, the maximum frequency can be controlled up to approximately 10 kHz. At higher frequencies, the speed error becomes larger.
  2. When the initial speed is equal to or greater than 46 Hz and less than 184 Hz, the maximum frequency can be controlled up to 50 kHz.
  3. When the initial speed is equal to or greater than 184 Hz, the maximum frequency can be controlled up to 50 kHz.

     The error in speed is the smallest around 50 kHz.
- Speed change during pulse output
  1. In type 0 control, if a value greater than the target speed at startup is specified, the speed is corrected to the target speed at startup.

     In type 1 control, if a value greater than 50 kHz is specified for the target speed, the speed is corrected to 50 kHz.
  2. For acceleration, the speed cannot be increased if it exceeds the acceleration disabled area start position. For the acceleration disabled area start position, refer to the description of the corresponding special register.
  3. For deceleration, the speed can only be decreased up to the deceleration lower limit speed. For the deceleration lower limit speed, refer to the description of the corresponding special register.

Supplement to pulse output operation

1. Pulses are output with duty fixed at 25%.
2. When the PLS+SIGN output type (direction output) is specified, pulse output starts approximately 300 μs after the direction signal (SIGN) is output. (The characteristics of the motor driver are considered.)

Precautions for programming

- When describing the same channel in both the normal program and the interrupt program, be sure to program not to execute them simultaneously.
- This instruction cannot be executed when the pulse output instruction executing flag corresponding to the channel to be started is ON.
- When rewriting during RUN is performed during pulse output, the pulse output stops while a program is being rewritten.
- Even if the control code is changed after the instruction is started, the change is invalid. It does not affect the operation.

Pulse output channels and areas used

Channel No.	Output	Output method	
ch0	Y0	CW	PLS
	Y1	CCW	SIGN
ch1	Y2	CW	PLS
	Y3	CCW	SIGN
ch2	Y4	CW	PLS
	Y5	CCW	SIGN
ch3	Y6	CW	PLS
	Y7	CCW	SIGN

Channel No.	Pulse output instruction executing flag	Elapsed value area	Target value area
ch0	R9120	DT90400 DT90401	DT90402 DT90403
ch1	R9121	DT90410 DT90411	DT90412 DT90413
ch2	R9122	DT90420 DT90421	DT90422 DT90423
ch3	R9123	DT90430 DT90431	DT90432 DT90433

Channel No.	Initial speed correction speed	Deceleration lower limit speed	Acceleration disabled area start position
ch0	DT90406	DT90407	DT90408 DT90409
ch1	DT90416	DT90417	DT90418 DT90419
ch2	DT90426	DT90427	DT90428

Channel No.	Initial speed correction speed	Deceleration lower limit speed	Acceleration disabled area start position
			DT90429
ch3	DT90436	DT90437	DT90438 DT90439

■ Specify the control code with an H constant.

H□□□□□□□□

10: Fixed

Control specification
0:JOG

Control specification 2
0: Type 0 (without target value)
1: Type 1 (with target value)

0: Fixed

Output specification
0: Pulse output
1: Calculation only

Motion direction
For type 0 (without target value)
0: Forward
1: Reverse
For type 1 (with target value)
0: Incremental
1: Absolute

Output type specification
0:CW/CCW
1: PLS+SIGN (Forward OFF/Reverse ON)
2: PLS+SIGN (Forward ON/Reverse OFF)

Info.

- About output specification

  If this instruction is started with "1: Calculation only" specified, no pulses are output. If this instruction is executed once for a channel, and then executed again for the same channel with the same parameters, it can start up faster. (This applies whether "Pulse output" or "Calculation only" is specified.)

  However, if parameters that are different from those of the previous execution are specified, it does not start up faster.

  * The same parameter means that all parameters except the output specification are the same.

■ For output type 0 (without target value)

Selection/Target value	CW/CCW	PLS+SIGN Forward OFF/ Reverse ON	PLS+SIGN Forward ON/ Reverse OFF	Elapsed value
Forward	Pulse output from CW	Pulse output when direction output is OFF	Pulse output when direction output is ON	Addition
Reverse	Pulse output from CCW	Pulse output when direction output is ON	Pulse output when direction output is OFF	Subtraction

■ For output type 1 (with target value)

Incremental <Relative value control>

Outputs the pulses set with the target value.

Selection/Target value	CW/CCW	PLS+SIGN Forward OFF/ Reverse ON	PLS+SIGN Forward ON/ Reverse OFF	Elapsed value
Positive value	Pulse output from CW	Pulse output when direction output is OFF	Pulse output when direction output is ON	Addition
Negative value	Pulse output from CCW	Pulse output when direction output is ON	Pulse output when direction output is OFF	Subtraction

Absolute <Absolute value control>

Outputs the pulses of the difference between the set target value and the current value.

Selection/Target value	CW/CCW	PLS+SIGN Forward OFF/ Reverse ON	PLS+SIGN Forward ON/ Reverse OFF	Elapsed value
When target value is larger than current value	Pulse output from CW	Pulse output when direction output is OFF	Pulse output when direction output is ON	Addition
When target value is smaller than current value	Pulse output from CCW	Pulse output when direction output is ON	Pulse output when direction output is OFF	Subtraction

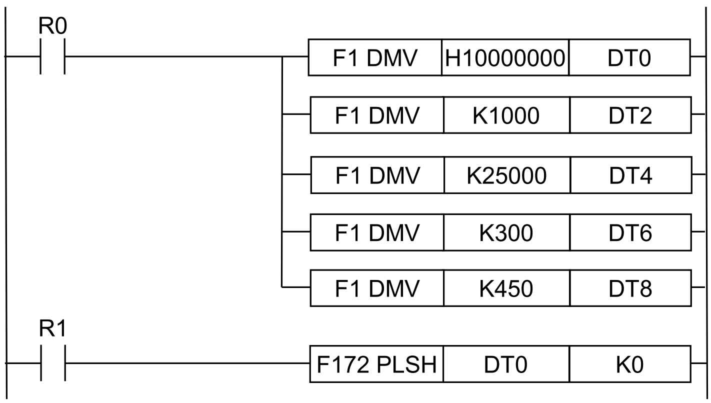
- About the acceleration/deceleration method and initial speed

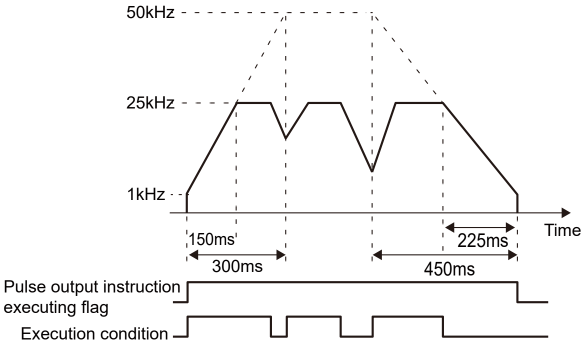
  If a deceleration stop request is made during acceleration, the deceleration is performed at the same slope as the deceleration time from the target speed.

  This instruction performs the processing with priority given to acceleration/deceleration time.

  Therefore, the initial speed may be corrected so that acceleration/deceleration can be performed in the specified time.

■ Program example





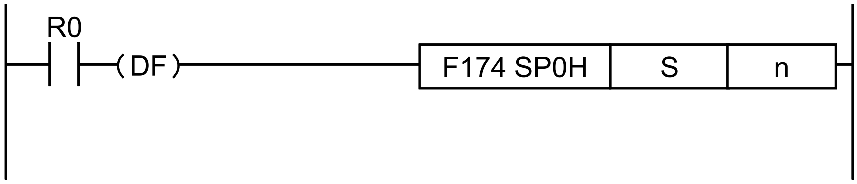
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the control code or frequency is outside the settable range (at startup of the instruction).
	Turns ON when the area is exceeded in index modification.
	Turns ON when [n] is outside the specified range.
	Turns ON when the pulse output for the specified channel is not set in the system register.

29.6 F174 SP0H [Pulse Output (Selectable Data Table Control Operation)]

Outputs pulses from a specified pulse output channel according to a specified data table.

■ Instruction format



■ Operands

Operand	Setting
S	Starting number of the area in which the data table is registered
n	Target channel for pulse output

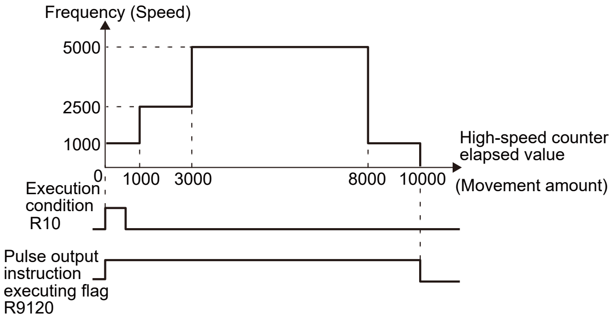
■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S	-	-	-	-	-	-	●	-	-	-	-	●
n	-	-	-	-	-	-	-	-	-	●	●	-

■ Outline of operation

- This instruction outputs pulses from a specified channel according to the settings specified in the data table starting with the address specified by [S] when the corresponding pulse output instruction executing flag is OFF and the execution condition is ON.
- The pulse frequency is switched when the elapsed value of the high-speed counter reaches the target value set in the data table. (This operation is executed as interrupt processing.)
- The pulse output stops when the elapsed value reaches the final target value.

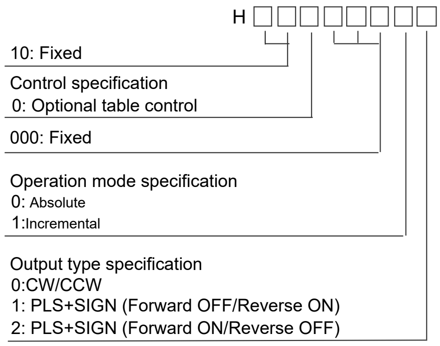
Operation image



■ Data table settings

S S+1	Control code (Note 1)
S+2 S+3	Frequency 1 (Hz) (Note 2)
S+4 S+5	Target value 1 (No. of pulses) (Note 3)
S+2 S+3	Frequency 2 (Hz)
S+4 S+5	Target value 2 (No. of pulses)
≈	≈
S+2n S+2n+1	Frequency n (Hz)
S+2(n+1) S+2(n+1)+1	Target value n (No. of pulses)
S+2(n+2) S+2(n+2)+1	K0 fixed: End of table

(Note 1) Specify the control code with an H constant.



(Note 2) Speed range (Frequency) (Hz) <K constant>
1 Hz to 50 kHz [K1 to K50000 (Unit: Hz)

(Note 3) Target value range <K constant>
K-2,147,483,648 to K2,147,483,647

Info.

- If a value greater than 50 kHz is specified for the frequency n, the speed is corrected to 50 kHz.
- Note that the instruction has the following characteristics depending on the specified frequency 1 value.
  1. When the frequency 1 is equal to or greater than 1 Hz and less than 46 Hz, the maximum frequency can be controlled up to approximately 10 kHz. If a value of 6 Hz or less is specified for the frequency n, the speed is corrected to 6 Hz.

2. When the frequency 1 is equal to or greater than 46 Hz and less than 184 Hz, the maximum frequency can be controlled up to 50 kHz. If a value of 46 Hz or less is specified for the frequency n, the speed is corrected to 46 Hz.
3. When the frequency 1 is equal to or greater than 184 Hz, the maximum frequency can be controlled up to 50 kHz. If a value of 184 Hz or less is specified for the frequency n, the speed is corrected to 184 Hz.

- The values of 32-bit data specified as target values should be within the range as shown in the table below.

Control code setting		Settable range of target value
Operation mode	**Output type**	
Incremental	Addition counting	Positive values
	Subtraction counting	Negative values
Absolute	Addition counting	Values larger than the current value
	Subtraction counting	Values smaller than the current value

■ Program example

[Operation]

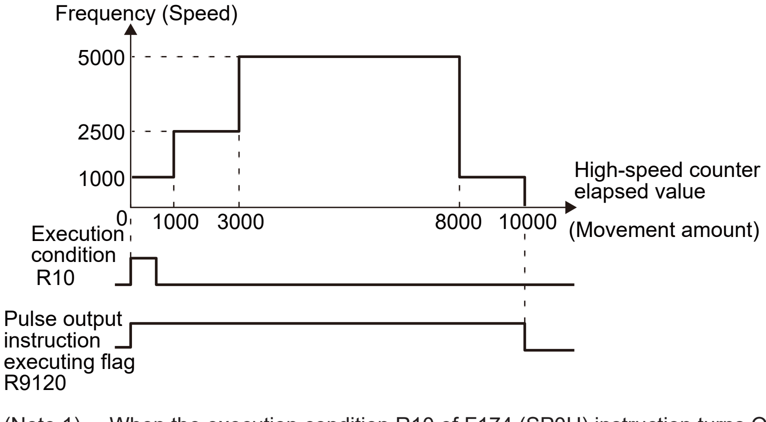
(1) This instruction starts the pulse output at 1000 Hz from the specified channel ch0 when the execution condition R10 of F174 (SP0H) instruction turns ON.

(2) The frequency is switched to 2500 Hz when 1000 pulses are counted at 1000 Hz.

(3) The frequency is switched to 5000 Hz when 3000 pulses are counted at 2500 Hz.

(4) The frequency is switched to 1000 Hz when 8000 pulses are counted at 5000 Hz.

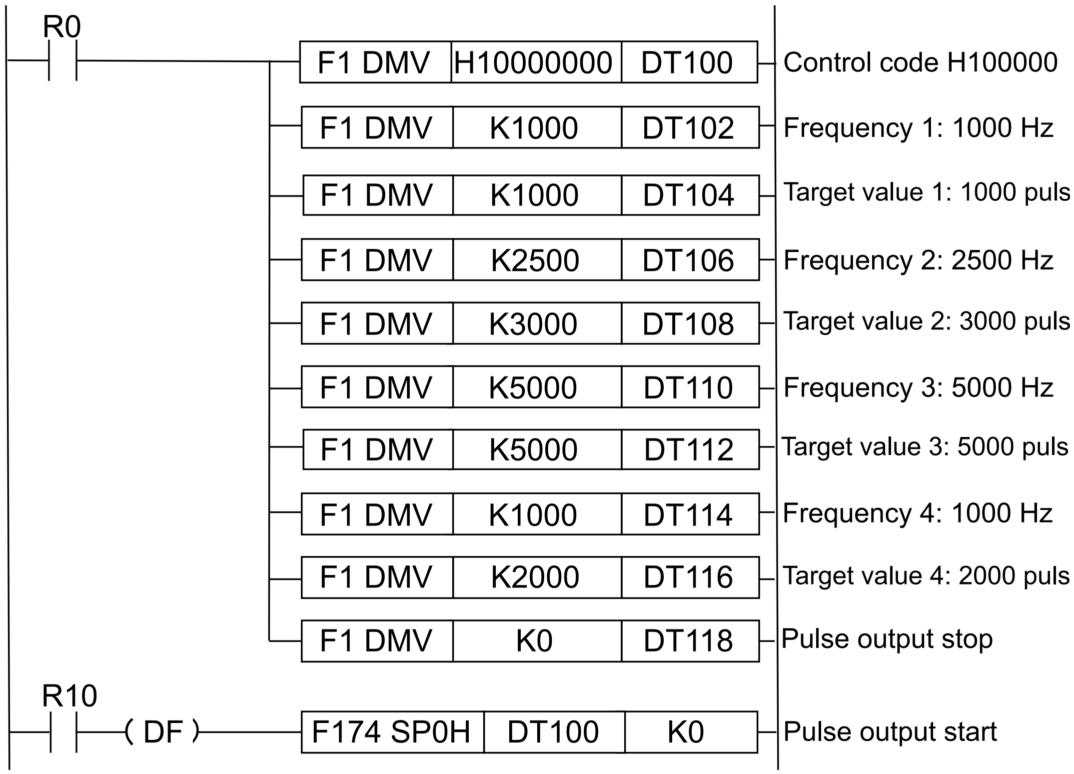
(5) The pulse output stops when 10000 pulses are counted.



(Note 1) When the execution condition R10 of F174 (SP0H) instruction turns ON, the pulse output instruction executing flag turns ON. When the elapsed value reaches 10000 and the pulse output stops, the high-speed counter control active flag turns OFF.

[Settings and program]

Set the instruction (Control specification: Optional table control, Operation mode: Incremental, Output type: CW/CCW).



■ Supplement to pulse output operation

When the PLS+SIGN output type (direction output) is specified, pulse output starts approximately 300 μs after the direction signal (SIGN) is output. (The characteristics of the motor driver are considered.)

■ Precautions for programming

- The pulse output instruction executing flag is ON until the pulse output stops after the execution condition of F174 (SP0H) instruction has turned ON.
- When the frequency 1 is any value outside of the settable range, an operation error occurs. (When the data of the frequency 1 is 0, nothing is executed and the operation ends.)
- When the frequency after the second step is 0 or outside of the specified range, the pulse output stops.
- If the direction is reversed when this instruction is executed at a specified target value, the pulse output stops.
- Do not execute this instruction simultaneously in a normal program and an interrupt program.
- If the table pointer exceeds the area of data registers DT during the pulse output, the pulse output stops and the high-speed counter control active flag turns OFF.
- Be sure to specify the target values within the specified range. If a value outside of the range is specified, the number of pulses different from the specified value is output.

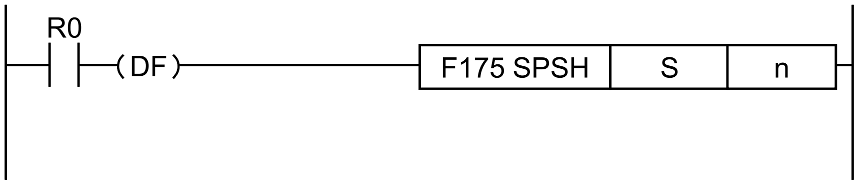
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when [n] is outside the specified range.
	Turns ON when the control code or frequency 1 is outside the settable range.

29.7 F175 SPSH [Pulse Output (Linear Interpolation)]

Pulses are output from two pulse output channels in accordance with the parameters in the designated data table, so that the path to the target position forms a straight line.

■ Instruction format



■ Operands

Operand	Setting
S	Starting number of the area in which the data table is registered
n	Target channel for pulse output (n: 0 to 3)

■ Memory area type that can be specified

Operands	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S	-	-	-	-	-	-	●	-	-	-	-	●
n	-	-	-	-	-	-	-	-	-	●	●	-

■ Outline of operation

- This instruction outputs pulses from the channels ch0 (X-axis) and ch1 (Y-axis) or ch2 (X-axis) and ch3 (Y-axis) when the corresponding pulse output instruction execution flag is OFF and the execution condition is ON.

Channel No.	Output	Output method	
ch0 (X-axis)	Y0	CW	PLS
	Y1	CCW	SIGN
ch1 (Y-axis)	Y2	CW	PLS
	Y3	CCW	SIGN
ch2 (X-axis)	Y4	CW	PLS
	Y5	CCW	SIGN
ch3 (Y-axis)	Y6	CW	PLS
	Y7	CCW	SIGN

- The control code, initial speed, maximum speed, acceleration/deceleration time, and target value are specified by creating a data table [S] to [S+11] using a user program, as described on the next page.
- The pulse output stops when the elapsed value reaches the final target value.

■ List of used areas

Pulse output channel No.	Pulse output instruction executing flag	Elapsed value area	Target value area	Target value area for match ON/OFF	Initial speed correction speed
ch0	R9120	DT90400 to DT90401	DT90402 to DT90403	DT90404 to DT90405	DT90406
ch1	R9121	DT90410 to DT90411	DT90412 to DT90413	DT90414 to DT90415	DT90416
ch2	R9122	DT90420 to DT90421	DT90422 to DT90423	DT90424 to DT90425	DT90426
ch3	R9123	DT90430 to DT90431	DT90432 to DT90433	DT90434 to DT90435	DT90436

■ Data table settings

Linear interpolation can be performed for each combination, (CH0 and CH1) or (CH2 and CH3), where acceleration time and deceleration time can be specified individually.

Address	Content	Note	Area
S S+1	Control code	(Note 1)	Setting area Set by the user program.
S+2 S+3	Composite speed: Initial speed (Hz)	(Note 2)	
S+4 S+5	Composite speed: Target speed (Hz)	(Note 2)	
S+6 S+7	Acceleration time (ms)	(Note 3)	
S+8 S+9	Deceleration time (ms)	(Note 3)	
S+10 S+11	X-axis target value	(Note 4)	
S+12 S+13	Y-axis target value	(Note 4)	
S+14 S+15	X-axis component speed: Initial speed (Hz)	(Note 5)	Operation result storage area Parameters of each axis component calculated by execution of the instruction are stored.
S+16 S+17	X-axis component speed: Target speed (Hz)		
S+18 S+19	Y-axis component speed: Initial speed (Hz)		
S+20 S+21	Y-axis component speed: Target speed (Hz)		

(Note 1) Specify the control code with an H constant.

H□□□□□□□□

10: Fixed

Control specification
0: Interpolation

Type specification
0: Linear interpolation

0: Fixed

Output specification
0: Pulse output
1: Calculation only

Operation mode specification
0: Incremental
1: Absolute

Output type specification
0:CW/CCW
1: PLS+SIGN (Forward OFF/Reverse ON)
2: PLS+SIGN (Forward ON/Reverse OFF)

(Note 2) About output specification

If this instruction is started with"1 : Calculation only"specified, no pulses are output.

If this instruction is executed once for a channel, and then executed again for the same channel with the same parameters, it can start up faster. (This applies whether "Pulse output" or "Calculation only" is specified.)

However, if parameters that are different from those of the previous execution are specified, it does not start up faster.

* The same parameter means that all parameters except the output specification are the same.

(Note 3) Composite speed range (Initial speed, target speed) (Hz) <K constant>

6.0 Hz to 50 kHz [K6 to K50000] (However, for 6.0 Hz, the angle is 0 or 90 degrees only. Also, for specifying 6.0 Hz, specify K6.)

- When K1 to K5 is specified, the composite speed range is the same as for 6.0 Hz (K6).
- When the initial speed is set to the target speed, the pulse output is performed without acceleration and deceleration.
- Specify the composite speed so that the component speed of each axis is 6 Hz or more.
- Composite speed (Initial speed): 30 kHz or less

  Notes on the specification of composite speed (initial speed)

  When the initial component speed for each of CH0 and CH2 is not 6.0 Hz or more, the path may not be linear.

  (This is because the following formula is not satisfied.)

$$\boxed{f \geqq \frac{6.0\sqrt{(\Delta x^2+\Delta y^2)}}{\Delta x}}$$

Δx: Channel whose distance of (target value - current value) is short

Δy: Channel whose distance of (target value - current value) is long

(Note 4) Acceleration time (ms)/Deceleration time (ms) range < K constant> K0 to K32767

In the case of 0, the pulse output is performed at the initial speed (composite speed) without acceleration and deceleration.

(Note) Specify the same value for acceleration time and deceleration time.

(Note 5) Target value (Movement amount)

K-8388608 to K8388607

When only one axis is activated:

- For the incremental mode, set the target value of the axis that is not activated to 0.
- For the absolute mode, set the target value of the axis that is not activated to the same as the current value.

(Note) In the case of linear interpolation, infinite rotation cannot be performed.

(Note 6) Component speed (initial speed and target value of each axis)

The component speed is stored as 2 words in real type.

$$\text{X-axis component speed} = \frac{\text{(Total speed) x (X-axis movement amount)}}{\sqrt{(\text{(X-axis movement amount)}^2 + \text{(Y-axis movement amount)}^2)}}$$

$$\text{Y-axis component speed} = \frac{\text{(Total speed) x (Y-axis movement amount)}}{\sqrt{(\text{(Y-axis movement amount)}^2 + \text{(Y-axis movement amount)}^2)}}$$

■ Component speed and correction

Note that the instruction has the following characteristics depending on the initial component speed calculated by the formula in (Note 5).

1. When the initial speed is equal to or greater than 1 Hz and less than 46 Hz, the maximum frequency can be controlled up to approximately 10 kHz. At higher frequencies, the speed error becomes larger.
2. When the initial speed is equal to or greater than 46 Hz and less than 184 Hz, the maximum frequency can be controlled up to 50 kHz.
3. When the initial speed is equal to or greater than 184 Hz, the maximum frequency can be controlled up to 50 kHz.

   The error in speed is the smallest around 50 kHz.

Note also that the initial speed may be corrected as a result of the calculation.

If corrected, the composite speed vector may be shifted at the start and end of pulse output.

To check if the specified initial speed is corrected, refer and compare to the initial speed correction speed value in the special register.

■ Output type

Incremental <Relative value control>

Outputs the pulses set with the target value.

Selection/Target value	CW/CCW	PLS+SIGN Forward OFF/ Reverse ON	PLS+SIGN Forward ON/ Reverse OFF	Elapsed value
Positive value	Pulse output from CW	Pulse output when direction output is OFF	Pulse output when direction output is ON	Addition
Negative value	Pulse output from CCW	Pulse output when direction output is ON	Pulse output when direction output is OFF	Subtraction

Absolute <Absolute value control>

Outputs the pulses of the difference between the set target value and the current value.

Selection/Target value	CW/CCW	PLS+SIGN Forward OFF/ Reverse ON	PLS+SIGN Forward ON/ Reverse OFF	Elapsed value
When target value is larger than current value	Pulse output from CW	Pulse output when direction output is OFF	Pulse output when direction output is ON	Addition
When target value is smaller than current value	Pulse output from CCW	Pulse output when direction output is ON	Pulse output when direction output is OFF	Subtraction

■ Supplement to pulse output operation

Pulses are output with duty fixed at 25%.

When the PLS+SIGN output type (direction output) is specified, pulse output starts approximately 300 μs after the direction signal (SIGN) is output. (The characteristics of the motor driver are considered.)

■ Precautions for programming

- Set the target value or movement amount to be within the following range.
  K-8,388,608 to +8,388,607
  When using this instruction in combination with other positioning instructions such as the F171 (SPDH) instruction, also set the target values for those instructions to be within the above range.
- When using this instruction for a purpose for which high accuracy is required, confirm the operation using a real machine.
- When describing the same channel in both the normal program and the interrupt program, be sure to program not to execute them simultaneously.

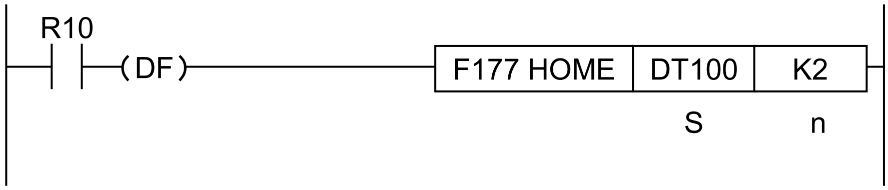
■ Flag operations

Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when [n] is other than 0.
	Turns ON when data in each of [S, S+1] to [S+10, S+11] of the data table is outside the settable range.
	Turns ON when the composite speed is specified as the initial speed [S+2, S+3] is greater than the target value [S+4, S+5].
	Turns ON when the composite speed is specified as the maximum speed [S+4, S+5] is greater than 50 kHz.
	Turns ON when the incremental mode is specified and the value of "current value + movement amount" is outside the range of -8388608 to +8388607.
	Turns ON when the absolute mode is specified and the value of target value is outside the range of -8388608 to +8388607.
	Turns ON when the same value is not specified for acceleration time and deceleration time.

29.8 F177 HOME [Pulse Output (Home Return)]

Performs home return in a specified pulse output channel.

■ Instruction format



■ Operands

Operand	Setting
S	Starting number of the area in which the data table is registered
n	Target channel for pulse output

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S	-	-	-	-	-	-	●	-	-	-	-	●
n	-	-	-	-	-	-	-	-	-	●	●	-

■ Outline of operation

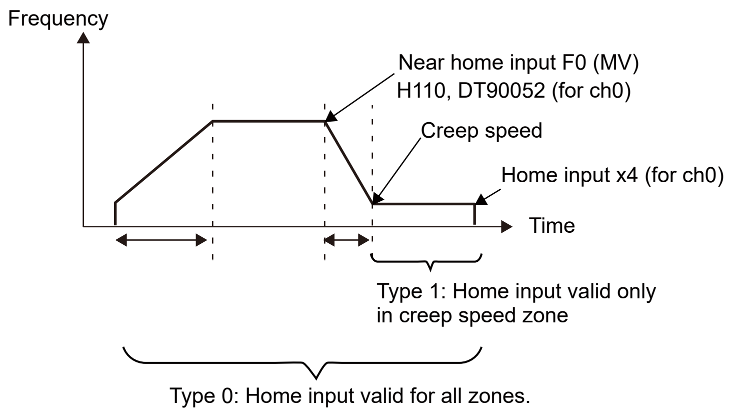
- This instruction outputs pulses for home return operation from a specified channel when the corresponding pulse output instruction executing flag is OFF and the execution condition is ON.

Info.

- To use the pulse output function, the usage of I/O must be set in the corresponding system register.
- C10 and C14 are for relay output and cannot be used for pulse output.

Explanation of operation mode

- Home return (Type 0): Home input is valid for all zones.
- Home return (Type 1): Home input is valid in the creep speed zone.



List of used areas

Channel No.	Output	Output method	
ch0	Y0	CW	PLS
	Y1	CCW	SIGN
ch1	Y2	CW	PLS
	Y3	CCW	SIGN
ch2	Y4	CW	PLS
	Y5	CCW	SIGN
ch3	Y6	CW	PLS
	Y7	CCW	SIGN

Channel No.	Near home input	Home input	Deviation counter clear	
			C16(Note 1)	C32, T32, F32(Note 2)
ch0	DT90052 bit4	X4	Y6	Y8
ch1		X5	Y7	Y9
ch2		X6	-	YA
ch3		X7	-	YB

(Note 1) Y6 and Y7 of CH3 are dual-use outputs for the deviation counter clear output of CH0 and CH1, and can be used only for one of the functions.

(Note 2) X4, X5, X6, and X7 for home position input are dual-use inputs for the high-speed counter and can be used for only one of these functions.

Channel No.	Pulse output instruction executing flag	Elapsed value area	Target value area
ch0	R9120	DT90400 DT90401	DT90402 DT90403
ch1	R9121	DT90410 DT90411	DT90412 DT90413
ch2	R9122	DT90420 DT90421	DT90422 DT90423

Channel No.	Pulse output instruction executing flag	Elapsed value area	Target value area
ch3	R9123	DT90430 DT90431	DT90432 DT90433

■ Data table settings

S S+1	Control code	
S+2 S+3	Initial speed (Hz)	Speed range (Frequency) (Hz) 1 Hz to 50 kHz [K1 to K50000 (Unit: Hz)]
S+4 S+5	Target speed (Hz)	
S+6 S+7	Acceleration time (ms)	Acceleration time to target speed Acceleration time range (ms) [K1 to K32760 (Unit: ms)]
S+8 S+9	Deceleration time (ms)	Deceleration time from target speed Deceleration time range (ms) [K1 to K32760 (Unit: ms)]
S+10 S+11	Creep speed (Hz)	Speed range (Frequency) (Hz) 1 Hz to 50 kHz [K1 to K50000 (Unit: Hz)]
S+12 S+13	Deviation counter clear signal output time	k0 to k200 k0=Deviation counter clear signal not output Kn = n* 0.5 ms

- Note that the instruction has the following characteristics depending on the specified initial speed.
  1. When the initial speed is equal to or greater than 1 Hz and less than 46 Hz, the maximum frequency can be controlled up to approximately 10 kHz. At higher frequencies, the speed error becomes larger.
  2. When the initial speed is equal to or greater than 46 Hz and less than 184 Hz, the maximum frequency can be controlled up to 50 kHz.
  3. When the initial speed is equal to or greater than 184 Hz, the maximum frequency can be controlled up to 50 kHz. The error in speed is the smallest around 50 kHz.

■ Control code setting

(Specify with an H constant.)

H □□□□□□□□

- 10: Fixed
- Control specification
  0: Home return
- Control specification type
  0: Home return type 0
  1: Home return type 1
- 00: Fixed
- Operation mode specification
  0: Forward
  1: Reverse
- Output type specification
  0:CW/CCW
  1: PLS+SIGN (Forward OFF/Reverse ON)
  2: PLS+SIGN (Forward ON/Reverse OFF)

■ Output type

Selection/ Target value	**CW/CCW**	**PLS+SIGN Forward OFF/Reverse ON**	**PLS+SIGN Forward ON/Reverse OFF**	**Elapsed value**
Forward	Pulse output from CW	Pulse output when direction output is OFF	Pulse output when direction output is ON	Addition
Reverse	Pulse output from CCW	Pulse output when direction output is ON	Pulse output when direction output is OFF	Subtraction

[Description of pulse output operation]

Pulses are output with duty fixed at 25%. When the PLS+SIGN output type is specified, pulse output starts approximately 300 μs after the direction signal is output. (The characteristics of the motor driver are considered.)

■ Precautions for programming

- Even when the home input is enabled, the pulse output starts when this instruction is executed.
- When the near home input is enabled during acceleration, the deceleration operation starts.
- When describing the same channel in both the normal program and the interrupt program, be sure to program not to execute them simultaneously.
- This instruction cannot be executed when the pulse output instruction executing flag corresponding to each channel is ON.
- If rewriting during RUN is performed while pulses are being output, more pulses than the setting may be output.
- For performing the software reset, disabling the counting, stopping the pulse output or near home processing, refer to "23.2 F0 MV (Pulse Output Control)".

■ Flag operations

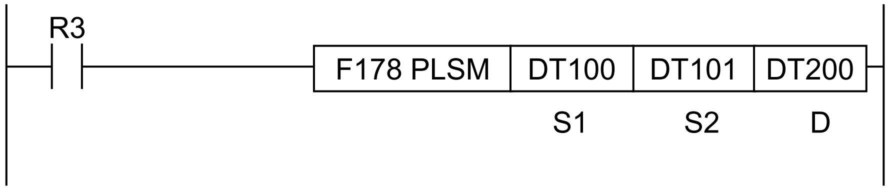
Name	**Description**
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification. Turns ON when n is other than 0.

Name	Description
	Turns ON when data in each of [S, S+1] to [S+4, S+5] of the data table is outside the settable range.
	Turns ON when the initial speed [S+2, S+3] is greater than the target value [S+4, S+5].

29.9 F178 PLSM (Input Pulse Measurement)

Measures the number of pulses and cycle of the specified high-speed counter channel when the high-speed counter function is used.

■ Instruction format



■ Operands

Operand	Setting
S1	Channel number and number of moving averages
S2	Count cycle
D	Starting number of output destination area

■ Memory area type that can be specified

Operand	WX	WY	WR	WL	SV	EV	DT	LD	I	Constant		Index modifier
										K	H	
S1	●	●	●	●	●	●	●	●	●	●	●	●
S2	●	●	●	●	●	●	●	●	●	●	●	●
D	-	●	●	●	●	●	●	●	●	-	-	●

■ Outline of operation

- This instruction measures the number of pulses or cycle of a specified high-speed counter channel based on the control data specified in [S1].
- In the pulse count measurement, the number of pulses for a specified high-speed counter is counted during the cycle specified in S2. The average value for the number of moving averages is calculated for each specified cycle and stored in D and D+1. After execution of the instruction, -1 is output for a time period of (n × count cycle), where n is the average count.
- For 1-μs unit pulse cycle measurement, the cycle of one pulse immediately after execution of this instruction is counted and stored in D+2 and D+3.
- For 1-ms unit pulse cycle measurement, the measured value is stored in D+4 and D+5 at every completion of 1-pulse cycle measurement.
- The same channel cannot be specified simultaneously for other high-speed counter control instructions [F165 (CAM0), F166 (HC1S), and F167 (HC1R)]. Execution of these instructions is controlled exclusively by the high-speed counter control active flag (R9110 to R9115).
- The number of channels that can be executed simultaneously is 2.
- During pulse measurement with this instruction, the execution condition should be always ON. The measurement is canceled when the execution condition turns OFF.

■ Specifying each item

Specifying the channel number and the number of moving averages [S1]: Specify the channel number of the high-speed counter and the number of moving averages. If necessary, also specify whether to perform pulse cycle measurement.

Measurement limit setting for periodic measurement in units of 1 ms

0:No measurement limit processing
1: 100ms
2: 200ms
3: 300ms
4: 500ms
5: 1s
6: 2s
7: 5s
8: 10s
9: 60s
A: Undefined

S1	n			

Pulse period measurement H0 ～ H1

Number of moving averages H1 ～ H5

Channel No. H0 ～ H5

0: No pulse period measurement
1: Pulse period measurement in 1 μs units
2: Pulse period measurement in 1 ms units
3: Pulse period measurement in 1 μs/ms units

[Measurement limit processing for cycle measurement]

Measurement limit processing is a function that sets the measured value to -1 if the measurement of a cycle is not completed within a certain time period.

- 1-μs unit cycle measurement

  In case of measurement timer overflow

  The judgment condition is that a short cycle can be measured although 174 ms or more have elapsed since the previous measurement request and, if met, the measured value is set to -1.

  In case of measurement not completed

  The judgment condition is that measurement is not yet completed although 350 ms or more have elapsed since the previous request and, if met, the measured value is set to -1. Even if the measurement is completed later, a measurement request is made again with the result discarded.

- 1-ms unit cycle measurement

  The cycle measurement counter _plsCycleTime0 is checked and, if the measurement limit specified by n above is exceeded, the elapsed value is set to -1.

  Even if the measurement is completed later, a measurement request is made again with the measured data ignored.

- Specifying the pulse count cycle [S2]

  Specify this in units of 1 ms.

  K1 to K5000 (1 ms to 5 sec.)

- Specifying the starting number of output destination area [D]
  Specify the starting number of output destination area.

D D+1	No. of pulses (Moving average value)	The latest value is stored for each measurement cycle specified by S2.
D+2 D+3	Pulse period (1 μs units)	1 pulse cycle immediately after execution of this instruction is stored.
D+4 D+5	Pulse period (1 ms units)	After execution of this instruction, the latest value is updated every time 1 pulse cycle measurement is completed.

1-μs unit measurement allows measurement up to approximately 174.7 ms.
1-ms unit measurement allows measurement up to approximately 49.7 days.

[About cycle measurement data]

At the start of measurement, "-1" is set.
When the measurement limit is exceeded, "-1" is set.

■ Precautions for programming

- Once the instruction is executed, the pulse measurement function is enabled until the control is canceled by the F0 (MV) S, DT90052 instruction.
- The F178 (PLSM) instruction cannot be executed when the high-speed counter function is not used.
- Do not execute this instruction simultaneously in a normal program and an interrupt program.

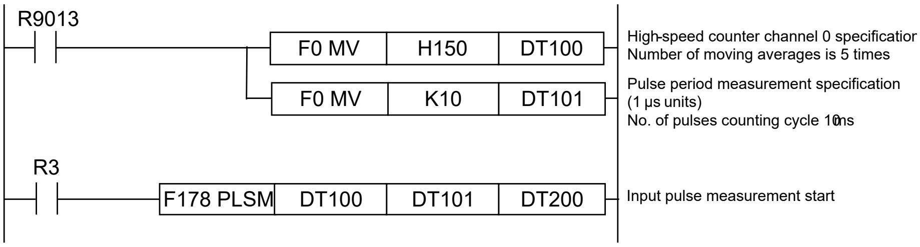
High-speed counter channel No.	Control active flag	Elapsed value	Target value
ch0	R9110	From DT90300 DT90301	From DT90302 DT90303
ch1	R9111	From DT90304 DT90305	From DT90306 DT90307
ch2	R9112	From DT90308 DT90309	From DT90310 DT90311
ch3	R9113	From DT90312 DT90313	From DT90314 DT90315
ch4	R9114	From DT90316 DT90317	From DT90318 DT90319
ch5	R9115	From DT90320 DT90321	From DT90322 DT90323

■ Setting example for input pulse measurement

[Conditions]

1. The channel number is 0 and the number of moving averages is set to 5.
   1-μs unit pulse period measurement is specified.
2. The count cycle is set to 10 ms.

[Program]



[Executing the program]

When internal relay R3 is ON, the operation is as follows.

Assume that pulses are input at a frequency of 10 kHz.

- DT200 to DT201 No. of pulses (Moving average value) → 100 pulses
  The number of input pulses is calculated every 10 ms to obtain the average value of the last 5 pulses for each count cycle.
- DT202 to DT203 1-μs unit pulse cycle → 100 μs (As the value, k100 is set.)
- DT204 to DT205 1-ms unit pulse cycle → 0 ms

Note

- The last digits of actual measured values may vary due to measurement errors.

■ Flag operations

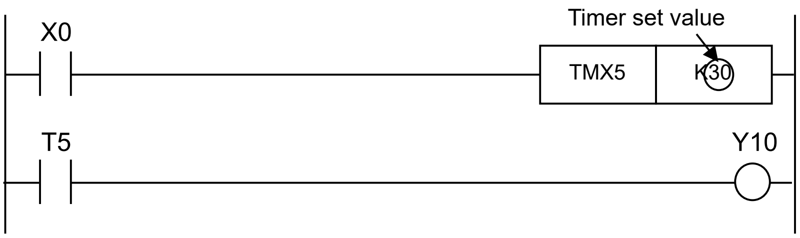
Name	Description
R9007 R9008 (ER)	Turns ON when the area is exceeded in index modification.
	Turns ON when the specified channel in [S1] is outside the range.
	Turns ON when the number of moving averages in [S1] is outside the specified range.
	Turns ON when the count cycle [S2] is outside the specified range.
	Turns ON when the data storage range [D] exceeds the area.
	Turns ON when the same channel is already controlled by an instruction of the same type.
	Turns ON when the number of execution channels is 3 or more.
	Turns ON when the high-speed counter setting for the specified channel is not made in the system register.

(MEMO)

30 Precautions for Programming

30.1 Changing the Set Value of Timer/Counter During RUN

30.1.1 How to Rewrite Constants in the Program



Method using programming tool software

Here is an example of changing the set value of timer 5 from K30 to K50.

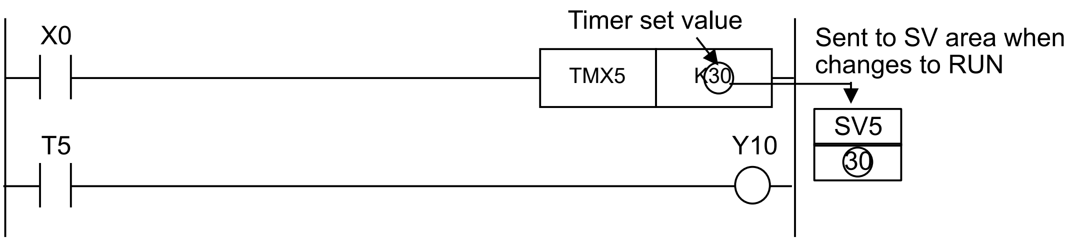
1 2 Procedure

1. Place the cursor on the timer 5 set value K30.
2. Enter the new constant K50, and press the Enter key.
3. Finalize the program by using [PB conversion] or [project conversion], and write it to the main unit.

   Operation and cautions after the change

   - When the program is changed using programming tool software, the timers and counters in operation will continue to operate unchanged. The program will start operating with the changed settings after the next execution condition changes from OFF to ON.
   - When the constants in the program are rewritten, the program itself is rewritten, so when the mode is switched and RUN again, or when the power is turned off and on, the program is preset with the changed settings.

30.1.2 Methods Used to Rewrite a Value in the Set Value Area



Changing a value in set value area SV

A value in set value area SV can be rewritten under the following conditions.

- Rewriting methods:
  1. Method using programming tool software
  2. Method using a program (high-level instruction)

Operation and cautions after the change

- After the change, the active timer/counter will continue to run. The program will start operating with the changed settings after the next execution condition changes from OFF to ON.
- With these methods, the value in set value area SV will change; however, the program itself is not rewritten. Therefore, when the mode is changed and then set back to RUN, or when the power is turned back on, the operation will be as follows.

  1. When the set value is specified by a K constant

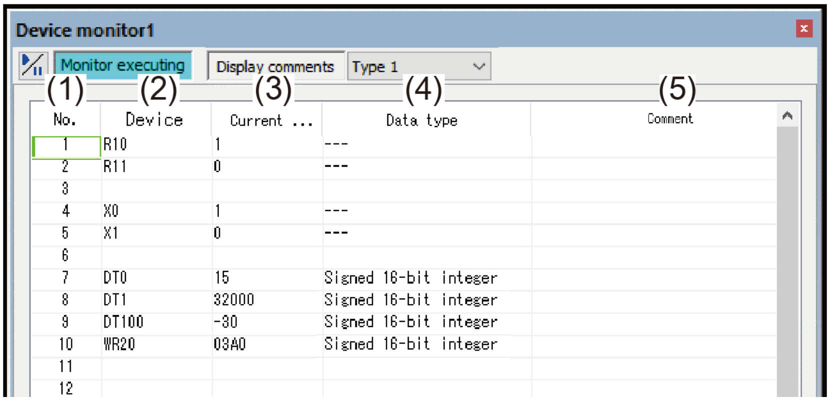
     The K constant is preset in set value area SV. After the change, the value will no longer be valid.

  2. When the set value is specified by a set value area number

     In the case of a non-hold-type timer/counter, 0 is preset in set value area SV. In the case of a hold-type timer/counter, the value changed by the method on the previous page is preset in set value area SV.

Method 1: Using programming tool software

From the menu bar, select:**Online**>**Device Monitor**.



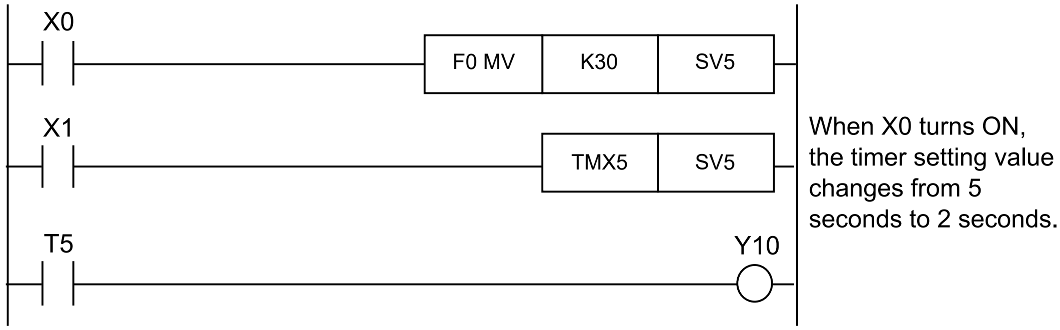
		Description
(1)	No.	Displays the line number.
(2)	Device	Pressing the<Enter>key or double-clicking in this field displays the device code and device number.
(3)	Current value	Displays the monitored data value. During online monitoring, data can be changed by pressing the<Enter>key or double-clicking in this field.
(4)	Data type	Pressing the<Enter>key or double-clicking in this field displays the number base (decimal, hexadecimal, binary, ASCII) and number of words to be monitored.
(5)	Comment s	Displays the I/O comments for each register. I/O comments can be added for each register by pressing the<Enter>key or double-clicking in this field.

(Note 1) For details, see the FPWIN GR7 help menu.

Method 2: Using a program (high-level instruction)

To change the set value of a timer/counter based on an input condition, etc., use a high-level instruction as shown below to rewrite the value in set value area SV of the relevant timer or counter.

Example: Changing the set value to K20 when input X0 turns ON



The SV area can also be specified directly in the set value area. The set value can be changed by changing the value to be transmitted, using the F0 instruction, etc.

30.2 Use of Duplicate Output

30.2.1 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 duplicate output.
- Even if the same output is used for **SET** instructions,**RST** instructions or high-level instructions (such as data transfer), it is not considered duplicate output.
- If you enter "RUN mode" while the duplicate output condition exists, under normal circumstances an error will result. (ERR/ALM LED blinks and the self-diagnostic flag R9000 turns ON.)

■ How to check for duplicate output

You can check for duplicate output in the program using a programming tool and the following method.

Execute the project's Total Check function from the menu.

If there is duplicate output, an error message [duplicate use (definition) error] and the address are displayed.

■ Enabling duplicate output

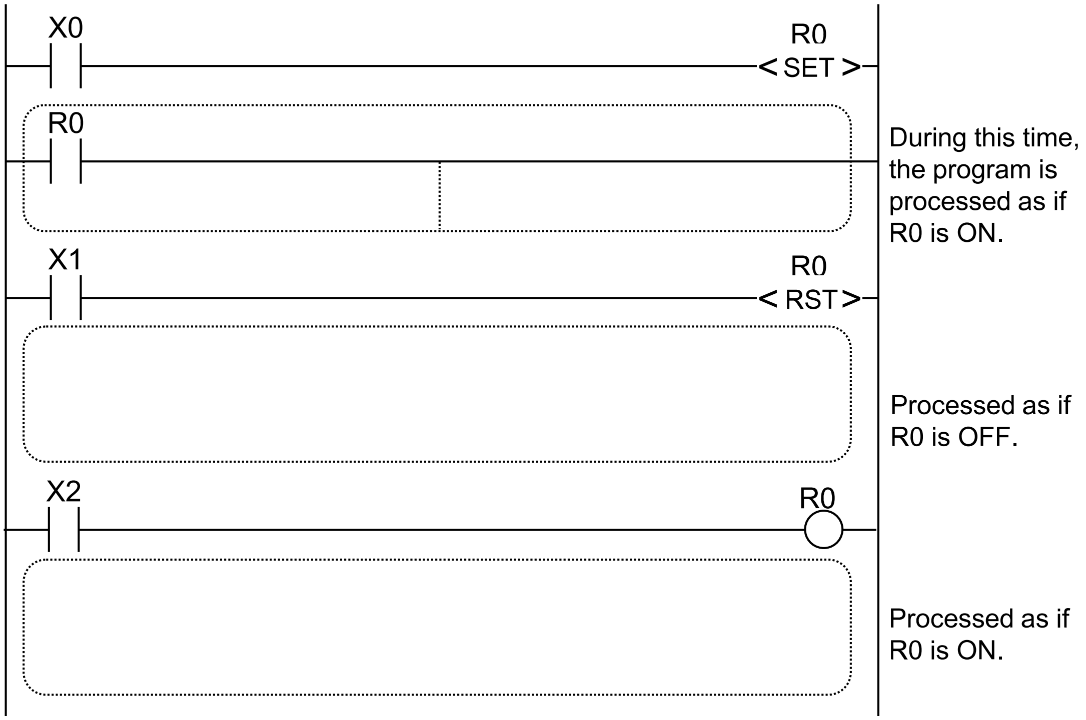
- If you need to use output repeatedly due to the content of the program, duplicate output can be enabled.
- To do this, change the content of system register No. 20 to "enable".
- In this case, an error will not result even if the program is executed.

30.2.2 Processing When Output Is Duplicated with OT, KP, SET, and RST Instructions

■ Status of internal and output relays during operation

If instructions that output to internal and output relays, such as the OT instruction, KP instruction, SET instruction, RST instruction, and transfer instructions, are executed in duplicate, the contents are rewritten at each step during operation.

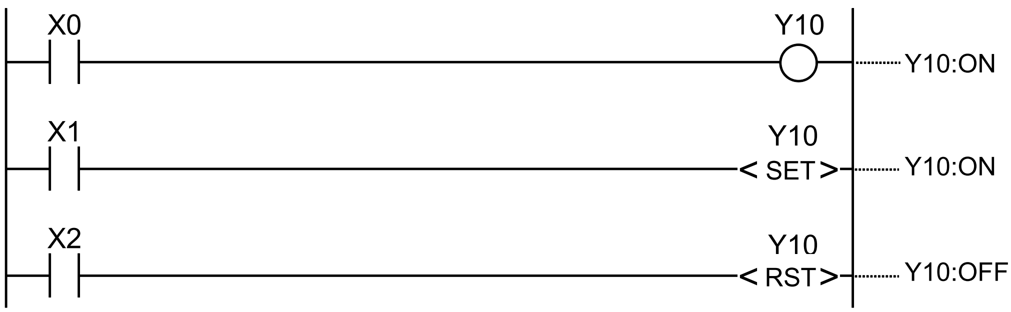
<Example> Processing when the SET instruction, RST instruction, and OT instruction are used (X0 to X2 are all ON)



■ **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> Output to the same output relay Y10 by the OT instruction, SET instruction, and RST instruction



When X0 to X2 are all ON, output occurs with Y10 OFF when I/O refresh is performed.

30.3 Rise Detection Method

30.3.1 Rise Detection Instructions

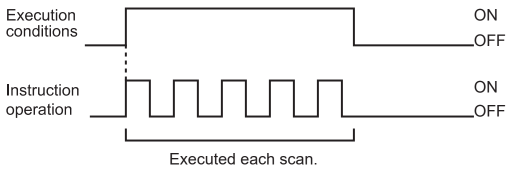
■ Instructions for which rise detection is performed

1. DF (rise differential)
2. CT (counter) count input
3. F118 UDC (up-down counter) count input
4. SR (shift register) shift input
5. F119 LRSR (left and right shift register) shift input
6. NSTP (next step)
7. Differential execution type high-level instruction (instruction specified by P and a number)

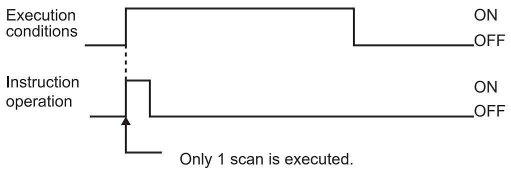
■ What is rise detection?

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 for instructions for which rise detection is performed

- 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. See below.
- 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.

  <Instructions that require caution when using instructions for which rise detection is performed>

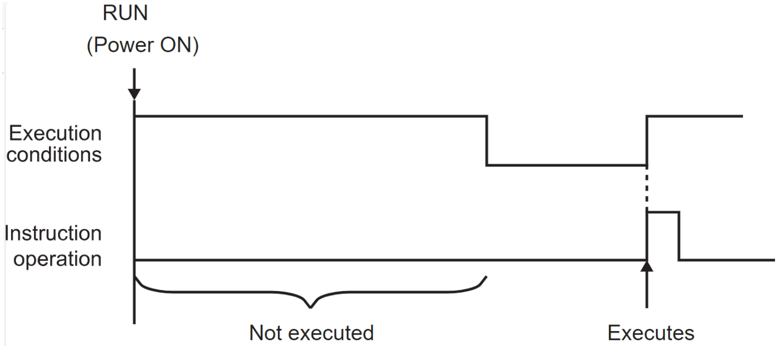
  1. MC to MCE instructions
  2. JP to LBL instructions

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

30.3.2 Operation and Precautions at Run Start Time

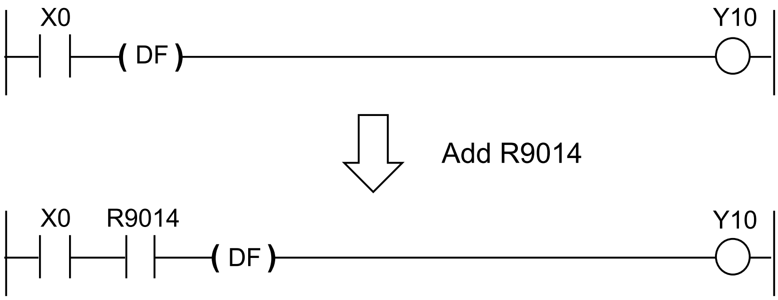
■ Operation of first scan after RUN begins

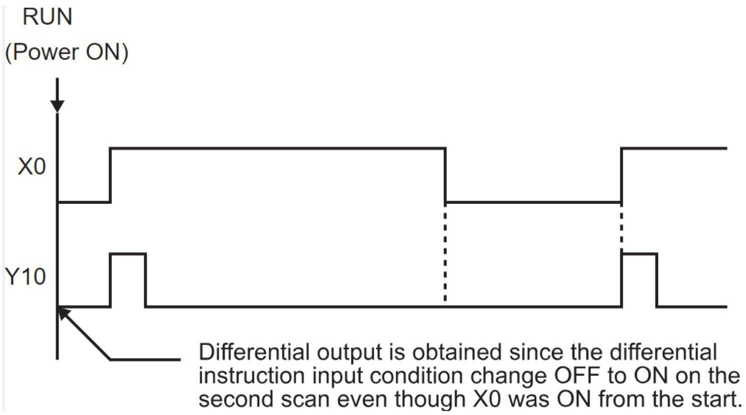
- The leading edge detection instruction is not executed when the mode has been switched to the"RUN mode", or when the power supply is booted in the"RUN mode", if the execution condition is already ON.



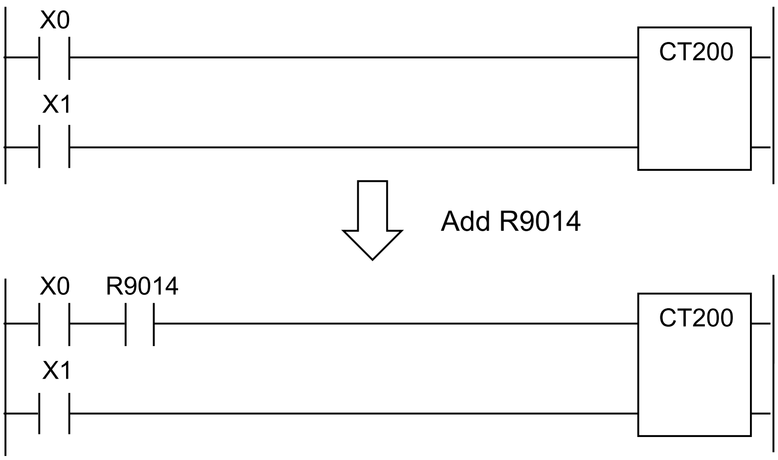
- If you need to execute an instruction when the execution condition is ON prior to switching to"RUN mode", use the special internal relay R9014 in your program as follows. (R9014 is a special internal relay which is OFF during the first scan and turns ON from the second scan onwards.)

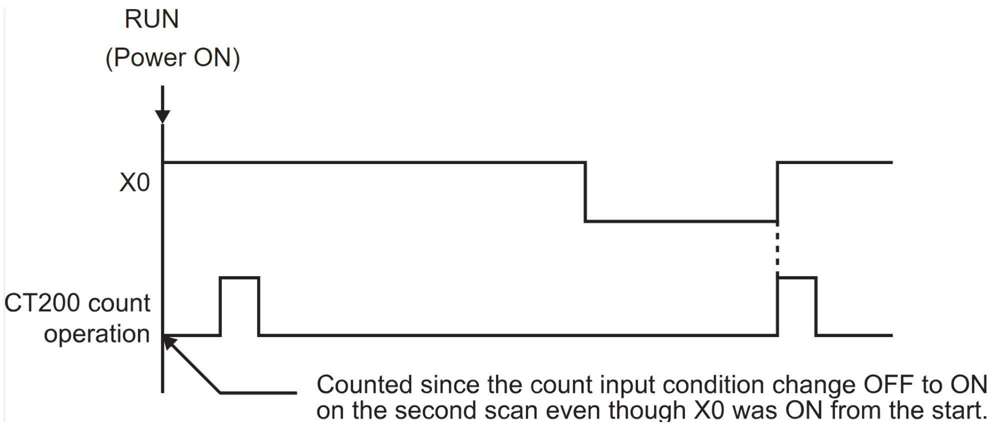
Example 1: DF (leading edge differential) instruction





Example 2: CT (counter) instruction



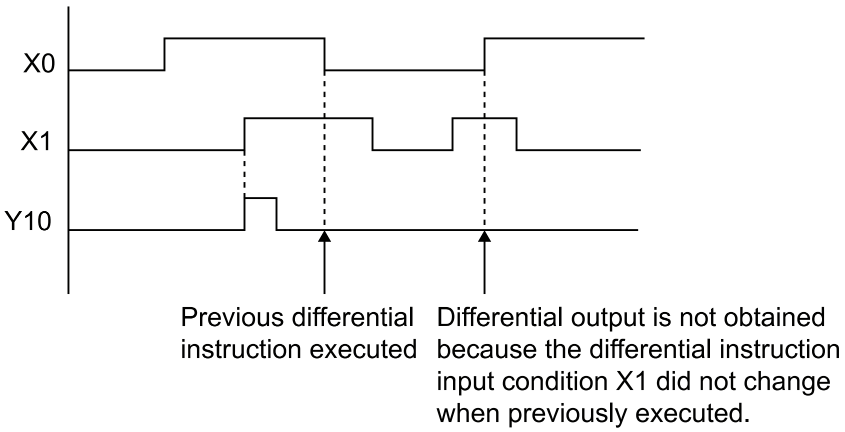


30.3.3 Precautions When Using Control Instructions

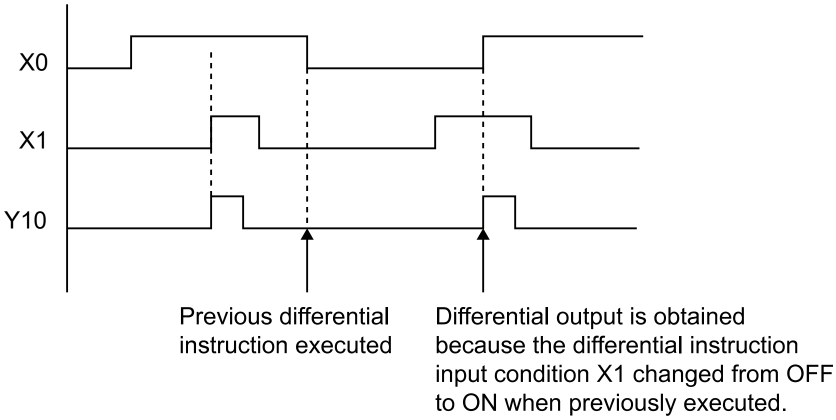
- Instructions that perform rise detection compare the execution condition from the last time that instruction was performed with current execution condition, and are only executed when the condition changes from OFF to ON. They are not executed in any other circumstance.
- When a rise detection instruction is used with an instruction that changes the order in which instructions are executed, such as MC and MCE, or JP and LBL, the operation of the instruction may change as follows depending on input timing.

Example 1: When using the differential instruction DF between MC and MCE

[Timing chart 1]

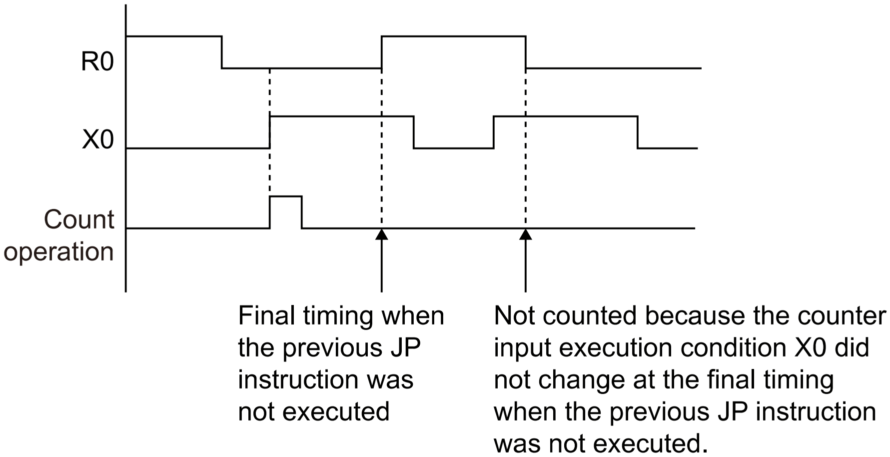


[Timing chart 2]

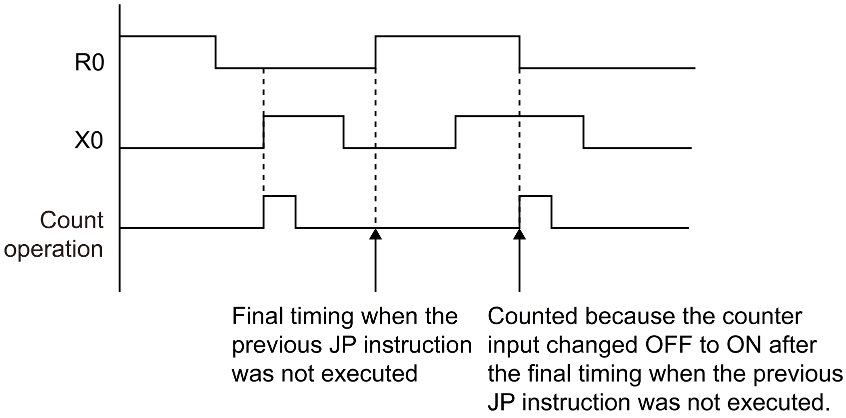


Example 2: When using the counter instruction between JP and LBL

[Timing chart 1]



[Timing chart 2]



30.4 Operation Errors

30.4.1 What is an operation error?

■ What is an operation error?

- It refers to invalid operation when executing operations using high-level instructions.
- When an operation error occurs, the ERR/ALM LED on the main unit flashes, and the operation error flags (R9007 and R9008) turn ON.
- The operation error code K45 is stored in the special data register DT90000.
- The address where the error was generated is stored in special data registers DT90017 to DT90018.

■ Operation error types

(1)	Address error	If, after using index modification, the memory address (number) specification exceeds the usable area
(2)	BCD error	If an attempt is made to calculate data other than BCD using an instruction that handles BCD data If the data for which BCD conversion is attempted exceeds the convertible range
(3)	Parameter error	If the data specified by an instruction for which specification of control data is required is out of range
(4)	Area exceeded error	If the target of a block instruction exceeds the memory range

30.4.2 Operation Mode when an Operation Error Occurs

Normally, the operation stops when an operation error occurs.
To have the operation continue even if an operation error occurs, change system register No.26 to "Continuation".
Implement this change as follows.

1 2 Procedure

1. Set the control unit to "PROG. mode".
2. Select "System register settings".
3. From the "System register settings" menu, select the "Action on error" screen to display system registers Nos. 20 to 28.
4. Clear the system register No.26 check box and change to "RUN".
5. Press [OK] to write the setting to the PLC.

30.4.3 Handling the Occurrence of Operation Errors

■ **Procedure**

1. Check the location where the error occurred

Refer to the error address stored in DT90017 and DT90018, then check the high-level instruction for that address.

2. Clear the error status

Clear the error by using the programming tool. (If the mode selection switch is set to RUN, the system will enter a RUN state when the error is cleared.)

Execute"Clear error"on the"Status display"menu of the programming tool software.

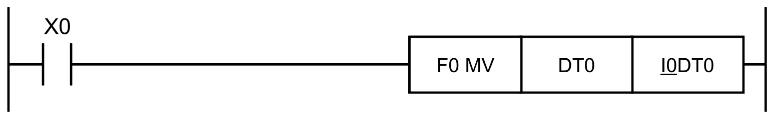
- The error can also be cleared by turning the power on and off in"PROG. mode". Note, however, that the content of the operation memory other than hold type data will be cleared.
- The error can also be cleared by using the self-diagnostic error set instruction (F148).

30.4.4 Points to Review in Program

Be sure to review your program by following the points below.

1. Check if an extraordinarily large value or negative value is stored in the index registers.

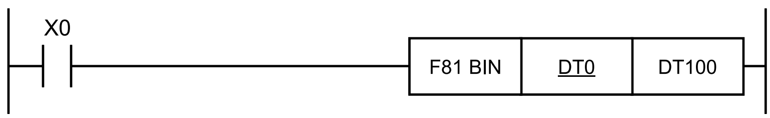
Example: When a data register is modified using an index register



In this case, the index register modifies the address of data register DT0. If the value of I0 is too large, it will exceed the specifiable range of the data register. If the data in I0 is larger than the final address of the data register, an operation error will occur. The same is true when the data in I0 is a negative value.

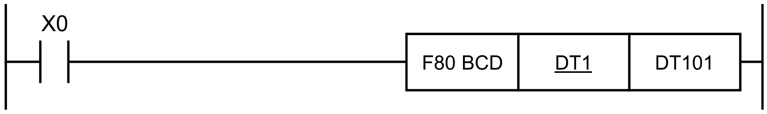
2. Check if there is any data that cannot be converted by BCD-BIN data conversion.

Example: When BCD-to-BIN conversion is attempted



In this case, if DT0 contains a hexadecimal number that includes one of the digits A through F such as"12A4", the data conversion will be impossible and an operation error will result.

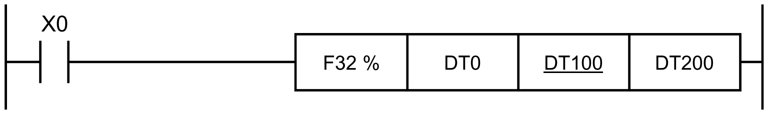
Example: When BIN-to-BCD conversion is attempted



In this case, if DT1 contains a negative value or a value greater than K9999, an operation error will occur.

3. Check if the divisor of a division instruction is"0".

<Example>



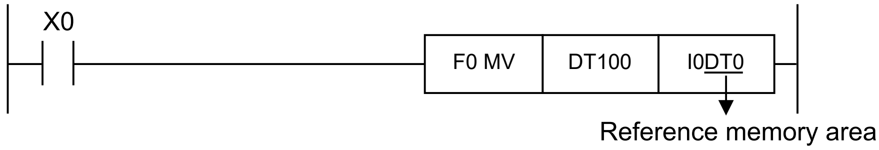
In this case, if the content of DT100 is"0", an operation error will occur.

30.5 How to Use the Index Register

30.5.1 Index Registers

- Index registers are used for indirect specification of values to numbers and operands in relays and memory areas. (This is called "index modification".)
- The range that can be specified is 14 points, and the numbers that can be specified are I0 to ID.
- Add the index register to the relay, memory area, or constant you want to modify, and then write the modifying value (16-bit data) to the index register.

<Example> Transferring the contents of data register DT100 to the number specified by the contents of an index register



In this example, the number of the destination data register varies depending on the contents of I0 with DT0 acting as a base. For example, when I0 is K10, the destination will be DT10, and when the I0 is K20, the destination will be DT20.

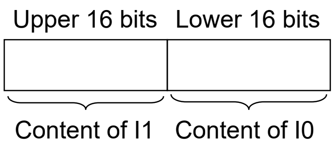
- In this way, index registers allow the specification of multiple memory areas with a single instruction, and thus index registers are very convenient when handling large amounts of data.

30.5.2 Index Modification Applicable Areas

- Index registers can be used to modify other types of memory areas in addition to data register DT.

  <Example> I0WX10, I2WY1, I3WR0, IASV0, IBEV2
- Constants can also be modified.

  <Example> I0K10, I0H1001
- When a 32-bit constant is modified, the index registers of the specified number and the following number are used in combination to handle the data as 32-bit data. The result of the modification is 32-bit data.

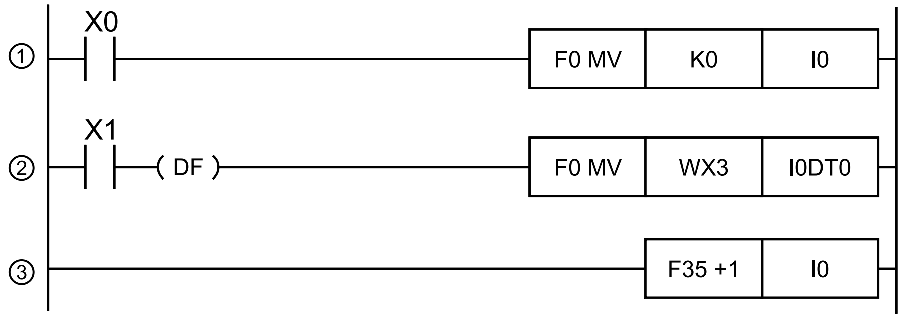


(Note 1) When modifying a 32-bit constant, do not specify the ID. Be aware that a syntax error will not occur even if this is specified.

30.5.3 Example of Using an Index Register

■ When external data is read successively

<Example> Writing the contents of input WX3 sequentially from data register DT0

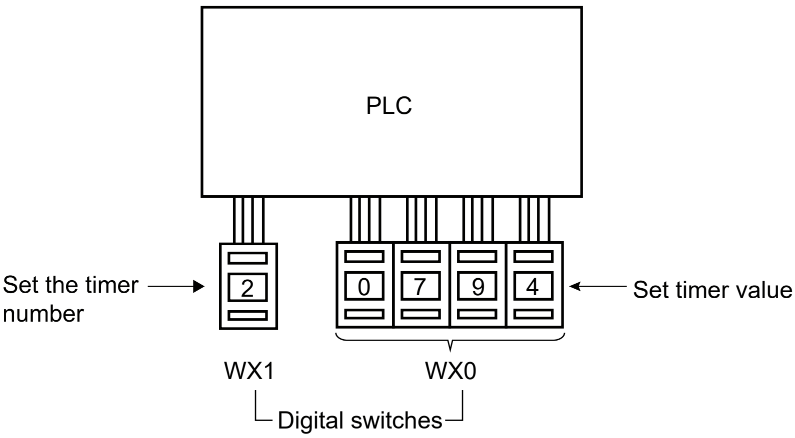


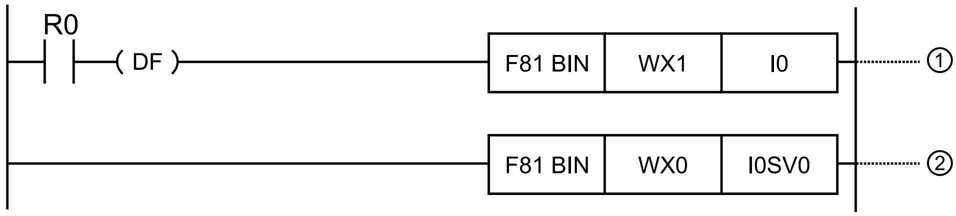
(1)	When X0 is ON, index register I0 is set to 0.
(2)	When X1 turns ON, the content of input WX3 is transferred to the data register specified by I0DT0.
(3)	Adds 1 to I0. In this case, the content of I0 changes in sequential order, so the write destination of the data register becomes as shown below.

Input of X1	Content of I0	Data writing destination
1st time	0	DT0
2nd time	1	DT1
3rd time	2	DT2

■ Inputting and outputting data according to the number specified by input

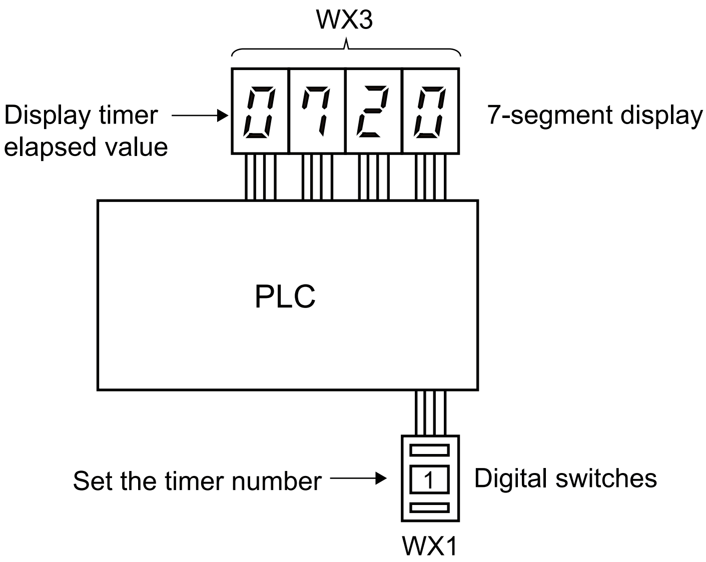
<Example 1> Setting a timer with a number specified by a digital switch

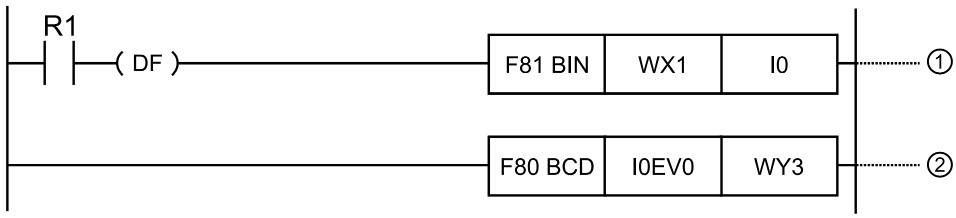




(1)	Timer number data WX1 is converted from BCD data to BIN data, and is set to index register I0.
(2)	Timer setting value data WX0 is converted from BCD data to BIN data, and is stored in the timer setting value area SV specified by the content of I0.

<Example 2> External output of the timer process value with the number specified by the digital switch





(1)	Timer number data WX1 is converted from BCD data to BIN data, and is set to index register I0.
(2)	The content of timer process value data EV specified by the content of I0 is converted to BCD data, and output to output WY3.

30.6 Handling BCD Data

(1) What is BCD?

BCD or binary coded decimal refers to a decimal number that is divided into single digits and expressed by binary numbers.

<Example> Decimal number expressed in BCD

Decimal	6	4	5
	↓	↓	↓ Each digit is converted to binary
BCD (binary coded decimal)	0110	0100	0101→H645

(2) Handling of BCD data in the PLC

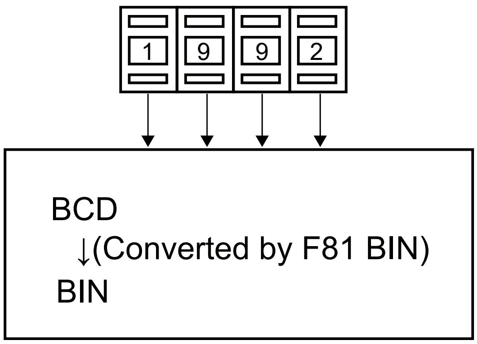
- When inputting data from a digital switch to the PLC or outputting data to a 7-segment display (with decoder), the input or output must be BCD data. In this case, use a data conversion instruction as shown in the examples below.
- BCD arithmetic instructions (F40 through F58) also exist that can operate directly on BCD data. However, since operations in the PLC are usually processed in BIN, it is more convenient to use BIN operation instructions (F20 through F38).

■ Inputting from a digital switch

Use the F81 BIN instruction as the BCD to BIN conversion instruction.

Digital switch

PLC



Data read into PLC (BCD data)

1				9				9				2			
0	0	0	1	1	0	0	1	1	0	0	1	0	0	1	0

↓

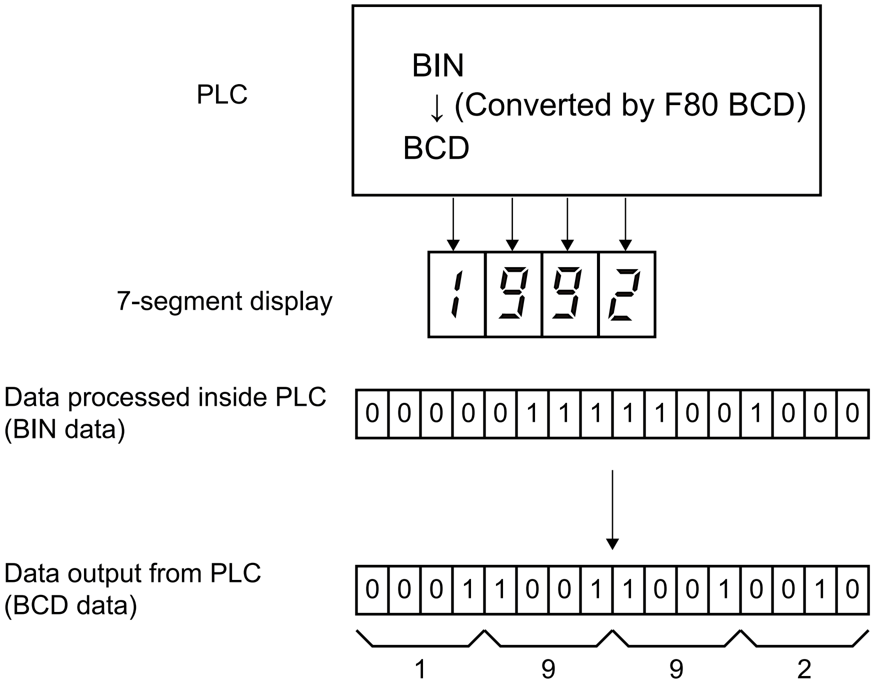
Data processed inside PLC (BIN data)

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

Converted to 1992 in decimal

■ **Outputting to a 7-segment display (with decoder)**

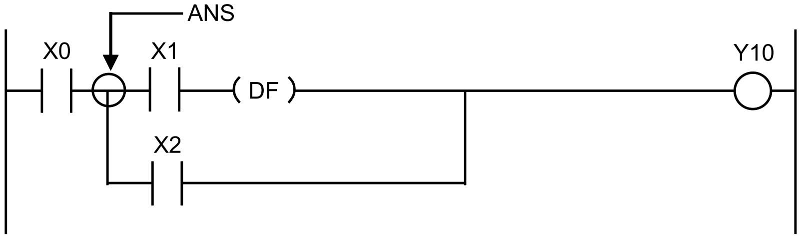
Use the F80 BCD instruction as the BIN to BCD conversion instruction.



30.7 Precautions for Programming

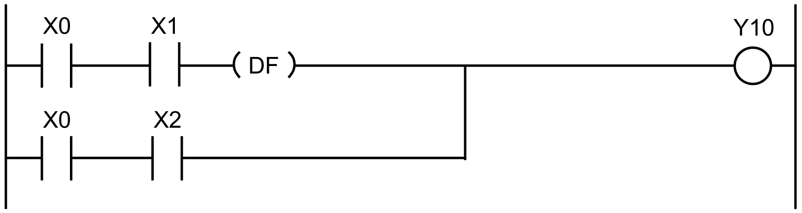
<Example 1>

Programs that do not execute correctly



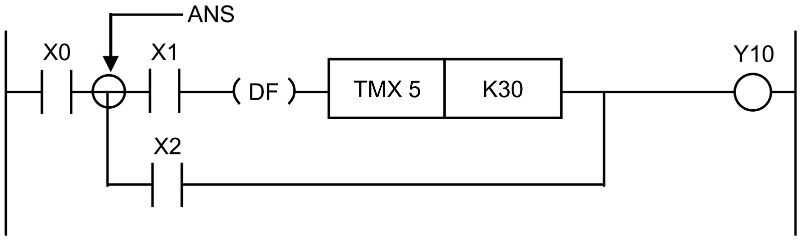
- If X1 turns ON first, Y10 does not turn ON even if X0 is ON.

Rewritten Program



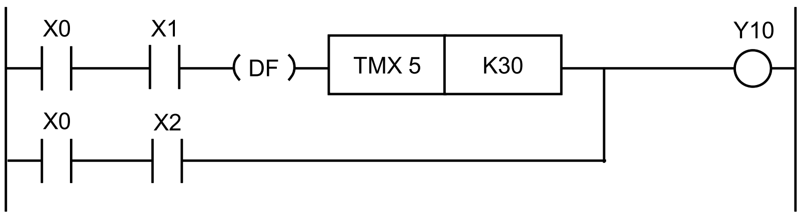
<Example 2>

Programs that do not execute correctly



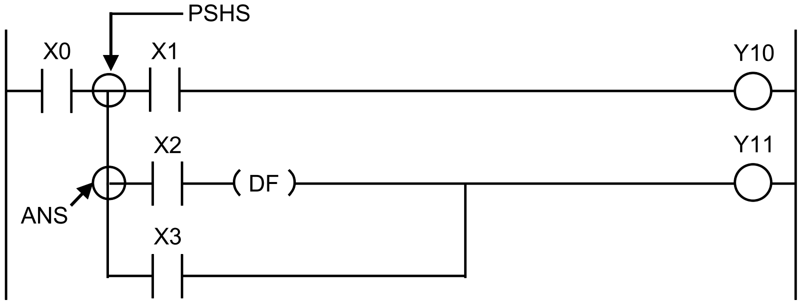
- Regardless of whether X0 is ON or OFF, if X1 is ON, TMX5 becomes active.

Rewritten program



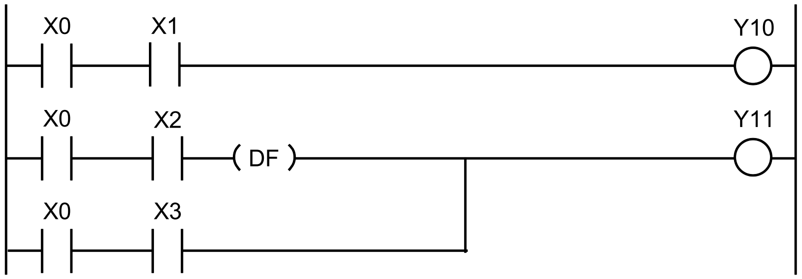
<Example 3>

Programs that do not execute correctly



- If X2 is ON first, even if X0 is ON, Y11 does not turn ON.

Rewritten program



- When a combination of contacts are set as the execution condition of a differential instruction (DF) or timer instruction, do not use an AND stack, push stack, read stack, or pop stack instruction.

30.8 Rewrite Function During RUN

30.8.1 Operation of Rewrite During RUN

■ How Rewrite During RUN Works

A program can be rewritten even during"RUN mode".

When attempting to rewrite a program during RUN, the tool service time is temporarily extended, the program rewritten, and operation is resumed without changing the mode.

For this reason, the scan time of one scan when rewriting during RUN is extended by several ms to several hundred ms.

■ Controller Operation During Rewrite

(1)	External output (Y) is held.
(2)	External input (X) is ignored.
(3)	Timer (T) stops the clock.
(4)	Rise and fall changes in the inputs of the differential instructions (DF), counter instructions (C), and right/ left shift registers are ignored.
(5)	Interrupt functions are stopped.
(6)	Internal clock relays (special internal relays) are also stopped.
(7)	Pulse output is stopped for the duration.

■ Setting Values for Timer/Counter Instructions

All set values specified with K constants for timer and counter instructions are preset to the set value SV area with corresponding numbers. (Values in elapsed value area EV do not change.)

■ Operation of the Rewrite During RUN Completion Flag

The rewrite during RUN completion flag (R9034) is a special internal relay that turns ON only for the first scan after rewrite during RUN is complete. It can be used instead of the initial pass relay following a change in a program.

30.8.2 When Rewriting During RUN is not Possible

■ When the timeout message is displayed

Even if the timeout message is displayed, it is likely that the PLC has been rewritten.

The ladder edit remains, so take the system offline, complete the program changes in the tool software, then change to online mode to check.

■ When timeout occurs using the GT series display unit through mode

Use GTWIN to extend the timeout period of the display unit. (The default value is 5 seconds.)

Select"Transfer"from**File**in the menu bar to open the data transfer screen. Select"Communication Conditions"from the data transfer screen to open the communication settings screen. The"Timeout"item displays the number of seconds, so change this value. Click the[OK]button to complete the setting change.

- When Rewriting During RUN is not Possible

1. When the result of rewriting is a syntax error, rewriting is not possible.
   [Specific example]
   When the rewriting would not form a pair of the following instructions
   a) Step ladder instructions (SSTP/STPE)
   b) Subroutine instructions (SUB/RET)
   c) Interrupt instructions (INT/IRET)
   d) JP/LBL
   e) LOOP/LBL
   f) MC/MCE
   Rewriting is not possible in the case of other syntax errors.
2. Rewriting during RUN is not possible during forced input/output operation.

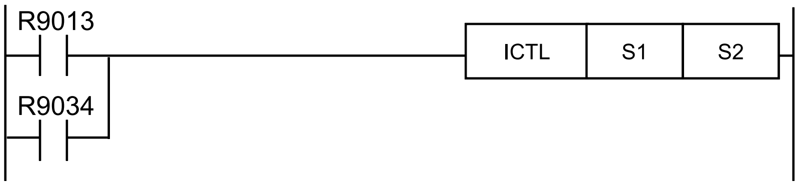
- Interrupt processing restrictions

Do not perform a rewrite during RUN when using interrupt, high-speed counter, pulse output, or PWM output functions.

Note that when executing a rewrite during RUN, the following operations will occur.

1. Interrupt programs will be disabled. Re-enable with an ICTL instruction.

e.g. When using R9034 (Completion flag for rewrite during RUN)

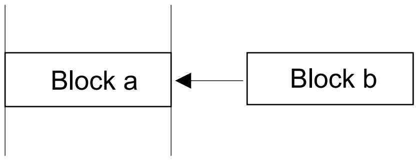


2. The high-speed counter will continue counting.
   Target value match ON/OFF instructions (F166 HC1S/F167 HC1R) will continue.
   Matching interrupt programs will be disabled during execution of the F166 HC1S/F167 HC1R instruction.
3. The pulse output and PWM output will be stopped.

Status	Instruction number	Name
Continue	F171 SPDH	Pulse output (with channel specification) (Home return)
Stop	F172 PLSH	Pulse output (with channel specification) (JOG operation)
Stop	F173 PLSH	PWM Output (with channel specification)
Continue	F174 SP0H	Pulse output (with channel specification) (Optional data table control operation)
Continue	F175 SPSH	Pulse output (linear interpolation)
Stop	F380 POSST	Positioning table start instruction
Stop	F381 JOGST	JOG operation start instruction
Stop	F383 MPOST	Positioning table simultaneous start instruction

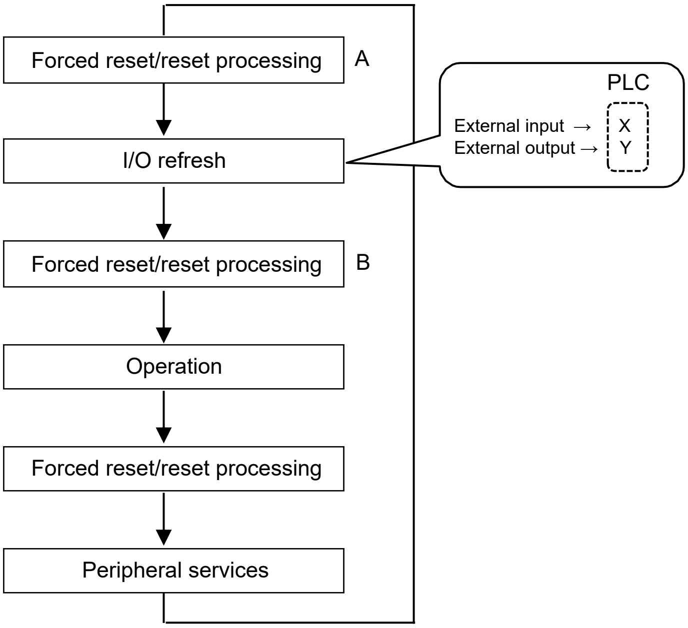
4. Fixed time sampling trace will not be stopped.

30.8.3 Method and Operation of Rewriting during RUN

<table>
<tr><th colspan="2">Items</th><th>FPWIN GR7 input</th></tr>
<tr><td colspan="2">Rewrite method</td><td>Up to 512 steps.
Changes are made in block units.
The program is rewritten online, when PB conversion is executed.

</td></tr>
<tr><td rowspan="8">Unique operation of each instruction</td><td>OT/KP</td><td>If instructions that were written in block a are deleted in block b, the status prior to the rewriting is held.</td></tr>
<tr><td>TM/CT</td><td>● If instructions that were written in block a are deleted in block b, the status prior to the rewriting is held.
● The set values specified by K constants in TM/CT instructions are preset to the SVs of all the corresponding numbers in the program.
(Elapsed value EV does not change)</td></tr>
<tr><td>Fun high-level instructions</td><td>If instructions that were written in block a are deleted in block b, the status prior to the rewriting is held.</td></tr>
<tr><td>MC/MCE</td><td>Always write MC/MCE instructions as a pair.</td></tr>
<tr><td>CALL/
SUB/
RET</td><td>A subroutine is a program that appears between SUBn and RET instructions.
Always write a subroutine to an address that comes after the ED instruction.</td></tr>
<tr><td>INT/IRET</td><td>An interrupt program is a program that appears between INTn and IRET instructions.
Always write a subroutine to an address that comes after the ED instruction.</td></tr>
<tr><td>SSTP/
STPE</td><td>Processes that have the same number cannot be defined in duplicate.
SSTP instructions cannot be written inside subprograms.</td></tr>
<tr><td>JP/
LOOP/
LBL</td><td>Always write the instruction for setting the number of loops before the LBL-LOOP instructions.</td></tr>
</table>

30.9 Processing During Forced Input/Output

■ Processing when forced input/output is initiated during RUN



1. Processing of external input (X)
   - For a contact for which forced input/output is specified, the forced ON/OFF operation takes precedence regardless of the state of the input from the input device in procedure B in the above flowchart. The input LED will not blink at this time; however, the area of input X in the operation memory will be overwritten.
   - For contacts for which forced input/output is not specified, the ON/OFF state is read according to the input state from the input device.
2. Processing of external output (Y)
   - For a contact for which forced input/output is specified, the forced ON/OFF operation takes precedence regardless of the operation result in procedure A in the above flowchart. The area of output Y in the operation memory will be forcibly overwritten at this time. External output will occur at the input/output refresh timing in the above figure.
   - For contacts for which forced input/output is not specified, the ON/OFF state is determined by the operation result.
3. Processing of timer (T)/counter (C)
   - For a contact for which forced input/output is specified, the forced ON/OFF operation takes precedence regardless of the input condition of the timer/counter. The contact of the timer (T)/counter (C) in the operation memory is overwritten at this time.

     Timing and counting will not be performed during control.
   - For contacts for which forced input/output is not specified, the ON/OFF state is determined by the contents of the operation result.

Operation during operation

Forcibly controlled internal relay R and output Y are overwritten according to the operation result.

31 Reference Material

31.1 Operation Memory Areas

<table>
<tr><th colspan="2" rowspan="2">Item</th><th colspan="2">Specifications</th></tr>
<tr><th>C10/C14/C16</th><th>C32/T32/F32</th></tr>
<tr><td rowspan="6">Relay</td><td>External input (X)(Note 1)</td><td colspan="2">1760 points (X0 to X109F)</td></tr>
<tr><td>External output (Y)(Note 1)</td><td colspan="2">1760 points (Y0 to Y109F)</td></tr>
<tr><td>Internal relay (R)(Note 2)</td><td colspan="2">4096 points (R0 to R255F)</td></tr>
<tr><td>Link relay (L)(Note 2)</td><td colspan="2">2048 points (L0 to L127F)</td></tr>
<tr><td>Timer/Counter (T/C)(Note 2)</td><td colspan="2">1024 points
(T0 to T1007/C1008 to C1023)(Note 3)</td></tr>
<tr><td>Special internal relay (R)</td><td colspan="2">224 points (R9000 onwards)</td></tr>
<tr><td rowspan="10">Memory area</td><td>External input (WX)(Note 1)</td><td colspan="2">110 words (WX0 to WX109)</td></tr>
<tr><td>External output (WY)(Note 1)</td><td colspan="2">110 words (WY0 to WY109)</td></tr>
<tr><td>Internal relay (WR)(Note 2)</td><td colspan="2">256 words (WR0 to WR255)</td></tr>
<tr><td>Link relay (WL)</td><td colspan="2">128 words (WL0 to WL127)</td></tr>
<tr><td>Data register (DT)(Note 2)</td><td>12315 words (DT0 to DT12314)</td><td>32765 words (DT0 to DT32764)</td></tr>
<tr><td>Link data register (LD)(Note 2)</td><td colspan="2">256 words (LD0 to LD255)</td></tr>
<tr><td>Timer/counter set value area (SV) (Note 2)</td><td colspan="2">1024 words (SV0 to SV1023)</td></tr>
<tr><td>Timer/counter elapsed value area (EV)(Note 2)</td><td colspan="2">1024 words (EV0 to EV1023)</td></tr>
<tr><td>Special data register (DT)(Note 3)</td><td colspan="2">440 words (DT90000 to DT90439)</td></tr>
<tr><td>Index register (I)</td><td colspan="2">14 words (I0 to ID)</td></tr>
<tr><td rowspan="5">Control Instruction points</td><td>Master control relay points (MCR) (MC)</td><td colspan="2">256 points</td></tr>
<tr><td>No. of labels (JP+LOOP) (LBL)</td><td colspan="2">256 points</td></tr>
<tr><td>No. of step ladders (SSTP)</td><td colspan="2">1000 processes</td></tr>
<tr><td>No. of subroutines (SUB)</td><td colspan="2">500 subroutines</td></tr>
<tr><td>No. of interrupt programs (INT)</td><td colspan="2">C10: 11 programs (6 points for external input, 1 point for periodic, 4 points for pulse matching)
Other than C10: 13 programs (8 points for external input, 1 point for periodic, 4 points for pulse matching)</td></tr>
<tr><td rowspan="6">Constants</td><td rowspan="2">Decimal constants (K)</td><td colspan="2">K-32,768 to K32,767 (for 16-bit operation)</td></tr>
<tr><td colspan="2">K-2,147,483,648 to K2,147,483,647 (for 32-bit operation)</td></tr>
<tr><td rowspan="2">Hexadecimal constants (H)</td><td colspan="2">H0 to HFFFF (for 16-bit operation)</td></tr>
<tr><td colspan="2">H0 to HFFFFFFFF (for 32-bit operation)</td></tr>
<tr><td rowspan="2">Floating point real numbers (f)</td><td colspan="2">F-1.175494 x 10^{-38} to F-3.402823 x 10^{38}</td></tr>
<tr><td colspan="2">F1.175494 x 10^{-38} to F3.402823 x 10^{-38}</td></tr>
</table>

(Note 1) The number of points shown here is the number of points provided as operation memory and the number of actual available points depends on the combination of hardware.

(Note 2) There are two types of operation memory areas: hold type that stores the last status even when the power is turned off or the mode is switched from RUN mode to PROG. mode and non-hold type that resets the memory area.

C10/C14/C16/C32: The hold-type and non-hold-type areas are fixed.

T32/F32: The division of the hold-type and non-hold-type areas can be changed by system register settings.

T32: If the built-in backup battery (secondary battery) runs out of charge and the hold area becomes indeterminate, the values in the hold area will be cleared to 0 when the power is turned on next time.

(Note 3) The points for the timer and counter can be changed by the setting of system register 5. The number given in the table are the numbers when system register 5 is at its default setting.

31.2 List of System Registers

■ Hold/Non-hold 1

No.	Name	Default	Setting range and description
5	Counter starting address	1008	0 to 1024
6	Hold-type area starting address for timer/counter (T32/F32)	1008	0 to 1024
7	Starting word No. of internal relay hold area (T32/F32)	248	0 to 256
8	Hold-type area starting address for data registers (T32/F32)	0	0 to 32765
14	Holding the step ladder (T32/F32)	Non-hold	Hold/Non-hold
4	Leading edge detection of the differential instruction during MC holds the previous value	Hold	Hold/Non-hold

■ Hold/Non-hold 2

No.	Name	Default	Setting range and description
10	Hold-type area starting word number for link relays for PLC link 0 (T32/F32)	0	0 to 64 64 to 128
11	Hold-type area starting word number for link relays for PLC link 1 (T32/F32)	64	
12	Hold type area starting number for link data registers for PLC link 0 (T32/F32)	0	0 to 128
13	Hold-type area starting word address setting for link data registers for PLC link 1 (T32/F32)	128	128 to 256

■ Action on error

No.	Name	Default	Setting range and description
20	Disable settings for duplicated output	Disable	Disable/Enable
23	Stop operation when an I/O verification error occurs	Stop	Stop/Run
26	Stop operation when calculating error occurs	Stop	Stop/Run

■ Time setting

No.	Name	Default	Setting range and description
31	Waiting time for managing multiple frame	6500 ms	10～81900 ms
32	SEND/RECV/RMRD/RMWT instruction communication timeout	10000 ms	10～81900 ms

No.	Name	Default	Setting range and description
34	Constant scan time	Normal scan	0：Normal scan 0～600 ms：Scans each specified time

■ PLC link 0 setting

No.	Name	Default	Setting range and description
40	Size of link relays	0	0 to 64 words
41	Size of link data registers	0	0 to 128 words
42	Send area starting word number of link relay	0	0 to 63
43	Size of link relays used for send area	0	0 to 64 words
44	Send area starting address of link data register	0	0 to 127
45	Size of link data registers used for send area	0	0 to 127 words
46	PLC link switch flag	Standard	Normal/Reverse
47	MEWNET-W0 PLC link max. station No.	16	1 to 16

■ PLC link 1 setting

No.	Name	Default	Setting range and description
50	Size of link relays	0	0 to 64 words
51	Size of link data registers	0	0 to 128 words
52	Send area starting word number of link relay	64	64 to 127
53	Size of link relays used for send area	0	0 to 64 words
54	Send area starting address of link data register	128	128 to 255
55	Size of link data registers used for send area	0	0 to 127 words
57	MEWNET-W0 PLC link max. station No.	16	1 to 16

■ Control Unit input settings 1 (High-speed counter)

No.	Name	Default	Setting range and description
400	High-speed counter operation mode settings (X0 to X2)	CH0: Not Set X0 as High Speed Counter	Not Set X0 as High Speed Counter 2 phase input (X0, X1) 2 phase input (X0, X1), Reset input (X2) Addition input (X0) Addition input (X0), Reset input (X2) Subtraction input (X0) Subtraction input (X0), Reset input (X2) One input (X0, X1) One input (X0, X1), Reset input (X2)

No.	Name	Default	Setting range and description
			Direction distinction (X0, X1) Direction distinction (X0, X1), Reset input (X2)
		CH1: Not Set X1 as High Speed Counter	Not Set X1 as High Speed Counter Addition input (X1) Addition input (X1), Reset input (X2) Subtraction input (X1) Subtraction input (X1), Reset input (X2)
400	High-speed counter operation mode settings (X3 to X5)	CH2: Not Set X3 as High Speed Counter	Not Set X3 as High Speed Counter 2 phase input (X3, X4) 2 phase input (X3, X4), Reset input (X5) Addition input (X3) Addition input (X3), Reset input (X5) Subtraction input (X3) Subtraction input (X3), Reset input (X5) One input (X3, X4) One input (X3, X4), Reset input (X5) Direction distinction (X3, X4) Direction distinction (X3, X4), Reset input (X5)
		CH3: Not Set X4 as High Speed Counter	Not Set X4 as High Speed Counter Addition input (X4) Addition input (X4), Reset input (X5) Subtraction input (X4) Subtraction input (X4), Reset input (X5)

(Note 1) When the operation mode is set to 2-phase, individual, or direction distinction, the settings of CH1 or CH3 in system register No. 400 are invalid.

(Note 2) When the reset input settings are overlapped, priority is given to the setting of CH1 in system register No.400.

(Note 3) When system registers Nos. 400 to 403 are set for the same input contact simultaneously, the priority order is as follows: 1. High-speed counter 2. Pulse catch 3. Interrupt input

<Example>

When the high-speed counter is used in the addition input mode, specifying X0 as interrupt input or pulse catch input will be invalid, and X0 will be activated as the counter input of the high-speed counter.

■ Control Unit input settings 2 (High-speed counter)

No.	Name	Default	Setting range and description
401	High-speed counter/pulse output setting (X6 to X7)	CH4: Not Set X6 as High Speed Counter	Not Set X6 as High Speed Counter Addition input (X6) Subtraction input (X6) 2 phase input (X6, X7) One input (X6, X7) Direction distinction (X6, X7)
		CH5: Not Set X7 as High Speed Counter	Not Set X7 as High Speed Counter Addition input (X7) Subtraction input (X7)

(Note 1) When the operation mode is set to 2-phase, individual, or direction distinction, the settings of CH5 in system register No. 401 are invalid.

(Note 2) When the reset input settings are overlapped, priority is given to the setting of CH3 in system register No.401.

(Note 3) When system registers Nos. 400 to 403 are set for the same input contact simultaneously, the priority order is as follows: 1. High-speed counter 2. Pulse catch 3. Interrupt input

<Example>

When the high-speed counter is used in the addition input mode, specifying X0 as interrupt input or pulse catch input will be invalid, and X0 will be activated as the counter input of the high-speed counter.

■ Control Unit output settings 2 (PLS/PWM) transistor type C16 or higher

No.	Name		Default	Setting range and description
402	Pulse/PWM output setting (Y0 to Y7)	CH0	Normal output (Y0, Y1)	Normal output (Y0, Y1) Pulse output (Y0, Y1) Pulse output (Y0, Y1)/Home input X4 Pulse output (Y0, Y1)/Home input X4/Position control start input X0 PWM output (Y0), Normal output (Y1)
		CH1	Normal output (Y2, Y3)	Normal output (Y2, Y3) Pulse output (Y2, Y3) Pulse output (Y2, Y3)/Home input X5 Pulse output (Y2, Y3)/Home input X5/Position control start input X1 PWM output (Y2), Normal output (Y3)
		CH2	Normal output (Y4, Y5)	Normal output (Y4, Y5) Pulse output (Y4, Y5) Pulse output (Y4, Y5)/Home input X6 Pulse output (Y4, Y5)/Home input X6/Position control start input X2 PWM output (Y4), Normal output (Y5)
		CH3	Normal output (Y6, Y7)	Normal output (Y6, Y7) Pulse output (Y6, Y7) Pulse output (Y6, Y7)/Home input X7 Pulse output (Y6, Y7)/Home input X7/Position control start input X3 PWM output (Y6), Normal output (Y7)

(Note 1) To use the pulse output or PWM output function, the Control Unit output setting must be set.

The output specified for the pulse output and PWM output cannot be used as normal output.

(Note 2) X4 to X7 can be also used as the home input of the pulse output CH0 to CH3.

To use the home return function of pulse output, be sure to set the home input. In that case, X4 to X7 cannot be set as the high-speed counter.

(Note 3) C16 type:

- To perform home return for pulse output CH0 with deviation counter clear, it is necessary to set the above Y6 to normal output since Y6 is fixed for use as the deviation counter clear signal.
- To perform home return for pulse output CH1 with deviation counter clear, it is necessary to set the above Y7 to normal output since Y7 is fixed for use as the deviation counter clear signal.

- Pulse output CH2 cannot be used for home return with deviation counter clear.

(Note 4) C32/T32/F32 type：

When home return with deviation counter clear is performed, the deviation counter clear signal corresponding to each CH is fixed for use, for example, CH0=Y8, CH1=Y9, CH2=YA, and CH3=YB.

To perform home return for each type, it is necessary to set the home input corresponding to each channel to be used for home return in system register No. 401.

Home input corresponding to each channel: CH0=X4, CH1=X5, CH2=X6, CH3=X7

To perform JOG positioning for each type, it is necessary to specify the use of the position control start input signal corresponding to each channel.

■ Interrupt/pulse catch settings

No.	Name	Default	Setting range and description
403	Pulse catch input setting	Not set	X0 X1 X2 X3 X4 X5 X6 X7 Controller input The pressed contact is set as pulse catch input.
404	Interrupt input settings	Not set	X0 X1 X2 X3 X4 X5 X6 X7 Controller input The pressed contact is set as interrupt input.

(Note 1) Set for each contact on the screen.

■ Interrupt edge setting

No.	Name	Default	Setting range and description
405	Control Unit input interrupt edge settings	Leading edge	X0 X1 X2 X3 X4 X5 X6 X7 Leading edge X0 X1 X2 X3 X4 X5 X6 X7 Trailing edge The pressed contacts are set as leading and trailing edges.

(Note 1) Set for each contact on the screen.

■ Tool port settings

No.	Name	Default	Setting range and description
410	Unit No.	1	1 to 99
412	Communication mode	Computer link	Computer link General-purpose Communication(Note 2)
	Modem connection	Not execute	Execute/Not execute
413	Transmission format	Data length: 8 bits Parity check: Odd Stop bit: 1	Data length: 7 bits/8 bits Parity check: None/Odd/Even Stop bit: 1/2 * The following settings are valid only when the communication mode in system register No. 412 is set to "general-purpose communication". Terminator: CR/CR+LF/None/ETX Header: No STX/STX

No.	Name	Default	Setting range and description
415	Baud rate(Note 2)	9600 bps	2400 bps, 4800 bps, 9600 bps, 19200 bps, 38400 bps, 57600 bps, 115200 bps
420	Receive buffer starting number during general-purpose communication	4096	0 to 32764
421	Receive buffer capacity during general-purpose communication	2048	0 to 2048

(Note 1) The transmission format when using PLC link is as follows: data length: 8 bits, parity: odd, stop bit: 1 bit (fixed).

Similarly, the baud rate is fixed at 115200 bps.

(Note 2) For the tool port, general-purpose communication is valid only in RUN mode. In PROG mode, the tool port is set to computer link mode regardless of the setting.

■ COM port settings

No.	Name	Default	Setting range and description
410	Unit No.	1	1 to 99
412	Communication mode	Computer link	Computer link General-purpose communication PLC link MODBUS RTU
	Modem connection	Not execute	Execute/Not execute
413	Transmission format	Data length: 8 bits Parity check: Odd Stop bit: 1	Data length: 7 bits/8 bits Parity check: None/Odd/Even Stop bit: 1/2 * The following settings are valid only when the communication mode in system register No. 412 is set to "general-purpose communication". Terminator: CR/CR+LF/None/ETX Header: No STX/STX
415	Baud rate(Note 2)	9600 bps	2400 bps, 4800 bps, 9600 bps, 19200 bps, 38400 bps, 57600 bps, 115200 bps
416	Receive buffer starting number during general-purpose communication	0	0 to 32764
417	Receive buffer capacity during general-purpose communication	2048	0 to 2048

(Note 1) The transmission format when using PLC link is as follows: data length: 8 bits, parity: odd, stop bit: 1 bit (fixed).

Similarly, the baud rate is fixed at 115200 bps.

(Note 2) For the tool port, general-purpose communication is valid only in RUN mode. In PROG mode, the tool port is set to computer link mode regardless of the setting.

■ Time constant setting of Control Unit input

(Note 1)

No.	Name	Default	Setting range and description
430	Time constant setting of Control Unit input 1 X0 to X3	1 ms	None/0.1 ms/0.5 ms/1 ms/2 ms/4 ms/8 ms/ 16 ms/32 ms/64 ms
431	Time constant setting of Control Unit input 1 X4 to X7		
432	Time constant setting of Control Unit input 2 X8 to XB (C32/T32/F32)		
433	Time constant setting of Control Unit input 2 XC to XF (C32/T32/F32)		

(Note 1) X6 and X7 are invalid for C10.

31.3 List of Special Relays

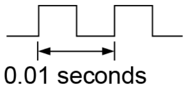
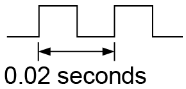
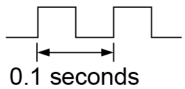
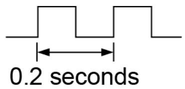
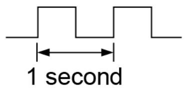
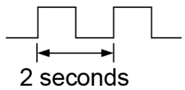
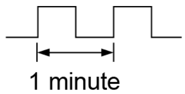
WR900 (Specified in units of words)

Relay No.	Name	Description
R9000	Self-diagnostic error flag	Turns ON when a self-diagnostic error occurs. → The self-diagnostic result is stored in DT90000.
R9001 to R9003	Reserved for system	-
R9004	I/O verification error flag	Turns ON when an I/O verification error is detected.
R9005 to R9006	Reserved for system	-
R9007	Operation error flag (hold type) (ER flag)	Turns ON when an operation error occurs after the unit has started operating, and remains ON while the unit operation continues. → The address where the error occurred is stored in DT90017. (It indicates the first operation error that has occurred.)
R9008	Operation error flag (new type) (ER flag)	Turns ON every time an operation error occurs. → The address where the operation error occurred is stored in DT90018. Every time a new error occurs, the data will be updated.
R9009	Carry flag (CY flag)	This flag is set when the operation result overflow or underflow occurs or when a shift system instruction is executed.
R900A	> flag	Turns ON if the execution result of a comparison instruction is "greater than the compared value".
R900B	= flag	Turns ON if the execution result of a comparison instruction is "equal to the compared value". Turns ON if the operation result of an operation instruction is "0".
R900C	< flag	Turns ON if the execution result of a comparison instruction is "less than the compared value".
R900D	Auxiliary timer contact	Turns ON when the specified time elapses after the auxiliary timer instruction (F137/F138) is executed. This flag turns OFF when the execution condition becomes unmatched.
R900E	Tool port communication error	Turns ON if a communication error is detected when using the tool port.
R900F	Constant scan error flag	Turns ON if the scan time exceeds the set time (system register no. 34) when the constant scan is executed. It also turns ON when 0 is set in the system register no. 34.

(Note 1) The same function is allocated to the special internal relay in parentheses.

WR901 (Specified in units of words)

Relay No.	Name	Description
R9010	Always ON relay	Always on.
R9011	Always OFF relay	Always off.
R9012	Scan pulse relay	Turns ON and OFF alternately at each scan.

Relay No.	Name	Description
R9013	Initial pulse relay (ON)	Turns ON for only the first scan after operation (RUN) has been started, and turns OFF for the second and subsequent scans.
R9014	Initial pulse relay (OFF)	Turns OFF for only the first scan after operation (RUN) has been started, and turns ON for the second and subsequent scans.
R9015	Step ladder Initial pulse relay (ON)	Turns ON in the first scan only, following startup of any single process, during stepladder control.
R9016 to R9017	Reserved for system	-
R9018	0.01-sec clock pulse relay	Clock pulse with a 0.01-second cycle. 
R9019	0.02-sec clock pulse relay	Clock pulse with a 0.02-second cycle. 
R901A	0.1-sec clock pulse relay	Clock pulse with a 0.1-second cycle. 
R901B	0.2-sec clock pulse relay	Clock pulse with a 0.2-second cycle. 
R901C	1-sec clock pulse relay	Clock pulse with a 1-second cycle. 
R901D	2-sec clock pulse relay	Clock pulse with a 2-second cycle. 
R901E	1-min clock pulse relay	Clock pulse with a 1-minute cycle. 
R901F	Reserved for system	-

WR902 (Specified in units of words)

Relay No.	Name	Description
R9020	RUN mode flag	Turns OFF while the mode selector is set to PROG. Turns ON while the mode selector is set to RUN.
R9021 to R9025	Reserved for system	-
R9026	Message flag	Turns ON when the message display instruction (F149) is executed.
R9027 to R9028	Reserved for system	-

Relay No.	Name	Description
R9029	Force flag	Turns ON during forced ON/OFF operation for input/output relays, timer/counter contacts, etc.
R902A	Interrupt enable flag	Turns ON while the external interrupt trigger is enabled.
R902B	Interrupt error flag	Turns ON when an interrupt error occurs.
R902C	Sample point flag	Sampling by instruction = 0, Sampling at constant time intervals = 1
R902D	Sampling trace end flag	When the sampling operation stops = 1, When the sampling operation starts = 0
R902E	Sampling stop trigger flag	When the sampling stop trigger occurs = 1, When the sampling stop trigger stops = 0
R902F	Sampling enable flag	When sampling starts = 1, When sampling stops = 0

WR903 (Specified in units of words)

Relay No.	Name	Description
R9030 to R9031	Reserved for system	-
R9032	COM port operation mode flag	Turns ON when the general-purpose communication function is used. Turns OFF when the computer link or PLC link function is used.
R9033	Print instruction execution flag	OFF: Instruction not active ON: Executing
R9034	Program edit flag in RUN mode	This is a special internal relay which turns ON for only the first scan following the completion of rewriting in RUN mode.
R9035 to R9036	Reserved for system	-
R9037	COM port communication error flag	Turns ON when a transmission error occurs during data communication. Turns OFF when transmission is requested using an F159 (MTRN) instruction.
R9038	COM port reception done flag during general-purpose communication	Turns ON when the terminator (end code) is received during general-purpose communication. Turns OFF when transmission is requested during general-purpose communication.
R9039	COM port transmission done flag during general-purpose communication	Turns ON when transmission is done during general-purpose communication. Turns OFF when transmission is requested during general-purpose communication.
R903A to R903D	Reserved for system	-
R903E	TOOL port reception done flag during general-purpose communication	Turns ON when the terminator (end code) is received during general-purpose communication. Turns OFF when transmission is requested during general-purpose communication.
R903F	TOOL port transmission done flag during general-purpose communication	Turns ON when transmission is done during general-purpose communication.

Relay No.	Name	Description
		Turns OFF when transmission is requested during general-purpose communication.

(Note 1) R9030 to R903F will change even during one scanning cycle.

WR904 (Specified in units of words)

Relay No.	Name	Description
R9040	TOOL port operation mode flag	Turns ON when the general-purpose communication function is used. Turns OFF when the computer link is used.
R9041	COM port PLC link flag	Turns ON when the PLC link function is used.
R9042 to R9043	Reserved for system	-
R9044	COM port SEND/RECV instruction execution flag	Indicates whether the F145 (SEND) or F146 (RECV) instruction can be executed or not for the COM port. OFF: Not executable (execution in progress) ON: Executable
R9045	COM port SEND/RECV instruction execution end flag	Indicates the execution state of the F145 (SEND) or F146 (RECV) instruction for the COM port. OFF: Normal end ON: Abnormal end (Communication error occurs) The error code is stored in DT90124.
R9046 to R904F	Reserved for system	-

(Note 1) R9040 to R904F will change even during one scanning cycle.

WR905 (Specified in units of words)

Relay No.	Name	Description
R9050	MEWNET-W0 PLC link transmission error flag	When using MEWNET-W0 Turns ON when there is a transmission error in the PLC link. Turns ON when there is an error in the setting for the PLC link area.
R9051 to R905F	Reserved for system	-

WR906 (Specified in units of words)

Relay No.	Name		Description
R9060	MEWNET-W0 PLC link 0 transmission assurance relay	Unit No. 1	Unit No. 1: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9061		Unit No. 2	Unit No. 2: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9062		Unit No. 3	Unit No. 3:

Relay No.	Name		Description
			is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9063		Unit No. 4	Unit No. 4: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9064		Unit No. 5	Unit No. 5: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9065		Unit No. 6	Unit No. 6: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9066		Unit No. 7	Unit No. 7: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9067		Unit No. 8	Unit No. 8: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9068		Unit No. 9	Unit No. 9: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9069		Unit No. 10	Unit No. 10: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R906A		Unit No. 11	Unit No. 11: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R906B		Unit No. 12	Unit No. 12: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R906C		Unit No. 13	Unit No. 13: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R906D		Unit No. 14	Unit No. 14: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF

Relay No.	Name		Description
R906E		Unit No. 15	Unit No. 15: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R906F		Unit No. 16	Unit No. 16: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF

WR907 (Specified in units of words)

Relay No.	Name		Description
R9070	MEWNET-W0 PLC link 0 operation mode relay	Unit No. 1	Turns ON when Unit No. 1 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9071		Unit No. 2	Turns ON when Unit No. 2 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9072		Unit No. 3	Turns ON when Unit No. 3 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9073		Unit No. 4	Turns ON when Unit No. 4 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9074		Unit No. 5	Turns ON when Unit No. 5 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9075		Unit No. 6	Turns ON when Unit No. 6 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9076		Unit No. 7	Turns ON when Unit No. 7 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9077		Unit No. 8	Turns ON when Unit No. 8 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9078		Unit No. 9	Turns ON when Unit No. 9 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9079		Unit No. 10	Turns ON when Unit No. 10 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R907A		Unit No. 11	Turns ON when Unit No. 11 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R907B		Unit No. 12	Turns ON when Unit No. 12 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R907C		Unit No. 13	Turns ON when Unit No. 13 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R907D		Unit No. 14	Turns ON when Unit No. 14 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R907E		Unit No. 15	Turns ON when Unit No. 15 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R907F		Unit No. 16	Turns ON when Unit No. 16 is in RUN mode. Turns OFF when the unit is in PROG. mode.

WR908 (Specified in units of words)

Relay No.	Name		Description
R9080	MEWNET-W0 PLC link 1 transmission assurance relay	Unit No. 1	Unit No. 1: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9081		Unit No. 2	Unit No. 2: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9082		Unit No. 3	Unit No. 3: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9083		Unit No. 4	Unit No. 4: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9084		Unit No. 5	Unit No. 5: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9085		Unit No. 6	Unit No. 6: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9086		Unit No. 7	Unit No. 7: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9087		Unit No. 8	Unit No. 8: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9088		Unit No. 9	Unit No. 9: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R9089		Unit No. 10	Unit No. 10: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R908A		Unit No. 11	Unit No. 11: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R908B		Unit No. 12	Unit No. 12: is normally communicating in PLC link mode: ON

Relay No.	Name		Description
			When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R908C		Unit No. 13	Unit No. 13: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R908D		Unit No. 14	Unit No. 14: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R908E		Unit No. 15	Unit No. 15: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF
R908F		Unit No. 16	Unit No. 16: is normally communicating in PLC link mode: ON When the unit is stopped, an error has occurred, or PLC link is not established: OFF

WR909 (Specified in units of words)

Relay No.	Name		Description
R9090	MEWNET-W0 PLC link 1 operation mode relay	Unit No. 1	Turns ON when Unit No. 1 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9091		Unit No. 2	Turns ON when Unit No. 2 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9092		Unit No. 3	Turns ON when Unit No. 3 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9093		Unit No. 4	Turns ON when Unit No. 4 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9094		Unit No. 5	Turns ON when Unit No. 5 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9095		Unit No. 6	Turns ON when Unit No. 6 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9096		Unit No. 7	Turns ON when Unit No. 7 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9097		Unit No. 8	Turns ON when Unit No. 8 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9098		Unit No. 9	Turns ON when Unit No. 9 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R9099		Unit No. 10	Turns ON when Unit No. 10 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R909A		Unit No. 11	Turns ON when Unit No. 11 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R909B		Unit No. 12	Turns ON when Unit No. 12 is in RUN mode.

Relay No.	Name		Description
			Turns OFF when the unit is in PROG. mode.
R909C		Unit No. 13	Turns ON when Unit No. 13 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R909D		Unit No. 14	Turns ON when Unit No. 14 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R909E		Unit No. 15	Turns ON when Unit No. 15 is in RUN mode. Turns OFF when the unit is in PROG. mode.
R909F		Unit No. 16	Turns ON when Unit No. 16 is in RUN mode. Turns OFF when the unit is in PROG. mode.

WR910（Specified in units of words)）

Relay No.	Name		Description
R9110	High-speed counter control active flag	HSC-CH0	● Turns ON when high-speed counter channels are controlled by the F165 (CAM0), F166 (HC1S), F167 (HC1R), or F178 (PLSM) instruction. ● This flag turns OFF when the control is canceled or the operation of this instruction is completed.
R9111		HSC-CH1	
R9112		HSC-CH2	
R9113		HSC-CH3	
R9114		HSC-CH4	
R9115		HSC-CH5	
R9116 to R911F	Reserved for system		-
R9120	Pulse output instruction executing flag	PLS-CH0	Turns ON when pulses are output by the F171 (SPDH), F172 (PLSH), F173 (PWMH), F174 (SP0H), F175 (SPSH), or F177 (HOME) instruction.
R9121		PLS-CH1	
R9122		PLS-CH2	
R9123		PLS-CH3	
R9124 to R912F	Reserved for system		-
R9130	Pulse output control flag	PLS-CH0	● Turns ON when high-speed counter channels are controlled by the F166 (HC1S) or F167 (HC1R) instruction. ● This flag turns OFF when the control is canceled or the operation of this instruction is completed.
R9131		PLS-CH1	
R9132		PLS-CH2	
R9133		PLS-CH3	
R9134 to R913F	Reserved for system		-

31.4 List of Special Data Registers

Register No.	Name	Description	R	W
DT90000	Self-diagnostic error code	When a self-diagnostic error occurs, the error code is stored.	●	
DT90001 to DT90009	Reserved for system	-		
DT90010	FP0/FP0R Expansion (Right expansion) Position of I/O verification mismatched unit [0 to 3]	When the installation state of FP0/FP0R Expansion I/O Unit changes from the state that it was in when the power was turned ON, the bit corresponding to the unit number turns ON (1). Monitor using binary display. 15 11 2 1 0 (Bit no.) 3 2 1 (Unit no.) ON(1): Abnormal OFF(0): Normal	●	
DT90011 to DT90013	Reserved for system	-		
DT90014	Operation auxiliary register for data shift instruction	As a result of the execution of data shift instruction F105 (BSR) or F106 (BSL), the overflowed 1-digit data is stored in bit 0 to bit 3. The value can be read and written by executing the F0 (MV) instruction.	●	●
DT90015 DT90016	Operation auxiliary register for division instruction	When executing the 16-bit division instruction F32 (%) or F52 (B%), the remainder of 16 bits is stored in DT90015. When executing the 32-bit division instruction F33 (D%) or F53 (DB%), the remainder of 32 bits is stored in DT90015 to DT90016. The value can be read and written by executing the F1 (DMV) instruction.	●	●
DT90017	Address with operation error (Hold)	The address where the first operation error occurred after startup of the operation is stored. Monitor using decimal display.	●	
DT90018	Address with operation error (Latest)	The address where the operation error occurred is stored. It will be updated every time an error occurs. Monitor using decimal display.	●	
DT90019	RING counter (2.5 ms)(Note 2)	The stored value is incremented by one every 2.5 ms. (H0 to HFFFF) Difference between the values of 2 points (absolute value) x 2.5 ms = Elapsed time between the 2 points	●	
DT90020	RING counter (10 μs) (Note 2) (Note 3)	The stored value is incremented by one every 10.67 μs. (H0 to HFFFF) Difference between the values of 2 points (absolute value) x 10.67 μs = Elapsed time between the 2 points (Note) The accurate figure is 10.67 μs.	●	
DT90021	Reserved for system	-		
DT90022	Scan time (Current value)(Note 1)	The current value of scan time is stored. [Stored value (decimal)] x 0.1 ms e.g. For K50, it is within 5 ms.	●	

Register No.	Name	Description	R	W
DT90023	Scan time (Minimum value) (Note 1)	The minimum value of scan time is stored. [Stored value (decimal)] x 0.1 ms e.g. For K50, it is within 5 ms.	●	
DT90024	Scan time (Maximum value) (Note 1)	The maximum value of scan time is stored. [Stored value (decimal)] x 0.1 ms e.g. For K125, it is within 12.5 ms.	●	
DT90025	Interrupt enable (mask) status (INT0 to 11)	The content set by the ICTL instruction is stored. Monitor using binary display. 15 13 11 7 3 0 (Bit no.) 7 3 0 (INT no.) 0: Interrupt disabled 1: Interrupt enabled INT0 to INT7: Interrupt input X0 to X7	●	
DT90026	Reserved for system	-		
DT90027	Periodical interrupt interval (INT24)	The content set by the ICTL instruction is stored. K0: Periodical interrupt is not used K1 to K3000: 0.5 ms to 1.5 sec. or 10 ms to 30 s	●	
DT90028	Interval of sampling trace	K0: Sampling by the SMPL instruction K1 to K3000(×10ms):10 ms to 30 sec.	●	
DT90029	Reserved for system	-		
DT90030	Character storage by F149 (MSG) instruction	The content (characters) set by the message display instruction (F149) is stored.	●	
DT90031				
DT90032				
DT90033				
DT90034				
DT90035				
DT90036	Reserved for system	-	●	

(Note 1) The scan time display shows the operation cycle time only in RUN mode. In PROG. mode, the scan time of operation is not displayed. The maximum and minimum values are cleared when switching the mode between RUN and PROG.

(Note 2) It is updated once at the beginning of every scan.

(Note 3) DT90020 is also updated when executing the F0 (MV), DT90020 and D instructions, therefore, it can be used for measuring a block time.

Register No.	Register No.	Name	R	W
DT90037	Work 1 for search instruction	When executing the F96 (SRC) instruction, the number that matches the search data is stored.	●	
DT90038	Work 2 for search instruction	When executing the F96 (SRC) instruction, the relative position that matches the search data is stored.	●	
DT90039 ～DT90051	Reserved for system			

<table>
<tr><th>Register No.</th><th>Register No.</th><th>Name</th><th>R</th><th>W</th></tr>
<tr><td rowspan="2">DT90052</td><td>High-speed counter control flag</td><td>Controls such as resetting the high-speed counter, disabling the count, and continuing and clearing a high-speed counter instruction can be performed by writing values with the F0 (MV) instruction.
bit no. 15 12 4 3 2 1 0
Channel specification
H0 to H5: CH0 to CH5
H00: Fixed
High-speed counter instruction 0: Continue 1: Clear
External reset input 0: Valid 1: Invalid
Count 0: Enable 1: Disable
Software reset 0: Disable 1: Enable</td><td>●</td><td>●</td></tr>
<tr><td>Pulse output control flag</td><td>Controls such as continuing and clearing a pulse output instruction can be performed by writing values with the F0 (MV) instruction.
bit no. 15 12 6 5 4 3 2 1 0
Channel specification
H0 to H3: CH0 to CH3
H1: Fixed
Position control start request 0: Disable 1: Enable
Deceleration stop request 0: Disable 1: Enable
Near home 0: Invalid 1: Valid
Pulse output 0: Continue 1: Stop
Pulse output control (Match ON/OFF) 0: Continue 1: Stop
Count 0: Enable 1: Disable
Software reset 0: Disable 1: Enable</td><td>●</td><td>●</td></tr>
<tr><td>DT90053</td><td>Clock/calendar monitor
(hour and minute)
(T32 only)</td><td>The hour and minute data of the clock/calendar monitor is stored.
You can only read and cannot write.
High byte Low byte
Hour dataH00 to H23 Minute dataH00 to H59</td><td>●</td><td></td></tr>
<tr><td>DT90054</td><td>Clock/calendar
(minute and second)
(T32 only)</td><td rowspan="4">The year, month, day, hour, minute, second, and day-of-the-week data of the clock/calendar is stored. The built-in clock/calendar is applicable until 2099 and supports leap years.
The clock/calendar can be set (time synch) by writing desired values using the programming tool or a program based on the transfer instruction (F0).</td><td rowspan="4">●</td><td rowspan="4">●</td></tr>
<tr><td>DT90055</td><td>Clock/calendar
(day and hour)
(T32 only)</td></tr>
<tr><td>DT90056</td><td>Clock/calendar
(year and month)
(T32 only)</td></tr>
<tr><td>DT90057</td><td>Clock/calendar</td></tr>
</table>

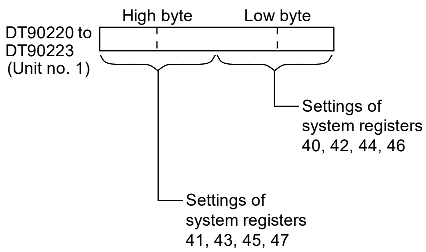
Register No.	Register No.	Name	R	W
	(day of week) (T32 only)	High byte / Low byte DT90054: Minute data (H00 to H59) / Second data (H00 to H59) DT90055: Day data (H01 to H31) / Hour data (H00 to H23) DT90056: Year data (H00 to H99) / Month data (H01 to H12) DT90057: – / Day of week data (H00 to H06) The day of the week is not set automatically. Allocate an arbitrary value in the range of H0 to H6 (0: Sun., 1: Mon., 2: Tue., 3: Wed., 4: Thu., 5: Fri., 6: Sat.).		
DT90058	Clock/calendar time setting and 30-second compensation register (T32 only)	It is used to adjust the time of the built-in clock/ calendar. ● Adjust the time by a program By setting the MSB of DT90058 to 1, the time is adjusted to that written to DT90054 to DT90057 by the F0 instruction. After the execution of the time adjustment, DT90058 will be cleared to zero. (It cannot be executed using any other instructions than F0 instruction.) <Example> Turn X0 ON to set the time to 12:00:00 on the 5th day. X0 ─(DF)─ [F0 MV, H 0, DT90054] Set 0 minute 0 second [F0 MV, H 512, DT90055] Set 12th hour 5th day. [F0 MV, H8000, DT90058] Set the time. ● Correct a difference within 30 seconds. By setting the LSB of DT90058 to 1, the time is moved up or down to be just 0 second. After the execution of the correction, DT90058 will be cleared to zero. <Example> Turn X0 ON to correct the time to be 0 second. X0 ─(DF)─ [F0 MV, H 1, DT90058] Correct to 0 second. When the time is 0 to 29 seconds when the correction is executed, it is moved down. When it is 30 to 59 seconds, it is moved up. In the above example, when the time is 5 minutes 29 seconds, it will be 5 minutes 0 seconds. When the time is 5 minutes 35 seconds, it will be 6 minutes 0 seconds.	●	●
DT90059	Communication error code	When a communication error occurs, the error code is stored.		
DT90060	Step ladder process (0 to 15)	Indicates the starting status of the step ladder process. When the process starts, the bit corresponding to its process number turns ON. Monitor using binary display. <Example> 15 11 7 3 0 (Bit no.) DT90060 15 11 7 3 0 (Process no.) 1：Starting 0：During stop The data can be written using the programming tool.	●	●
DT90061	Step ladder process (16 to 31)			
DT90062	Step ladder process (32 to 47)			
DT90063	Step ladder process (48 to 63)			

Register No.	Register No.	Name	R	W
DT90064	Step ladder process (64 to 79)			
DT90065	Step ladder process (80 to 95)			
DT90066	Step ladder process (96 to 111)			
DT90067	Step ladder process (112 to 127)			
DT90068	Step ladder process (128 to 143)			
DT90069	Step ladder process (144 to 159)			
DT90070	Step ladder process (160 to 175)			
DT90071	Step ladder process (176 to 191)			
DT90072	Step ladder process (192 to 207)			
DT90073	Step ladder process (208 to 223)			
DT90074	Step ladder process (224 to 239)			
DT90075	Step ladder process (240 to 255)			
DT90076	Step ladder process (256 to 271)			
DT90077	Step ladder process (272 to 287)			
DT90078	Step ladder process (288 to 303)			
DT90079	Step ladder process (304 to 319)			
DT90080	Step ladder process (320 to 335)			
DT90081	Step ladder process (336 to 351)			
DT90082	Step ladder process (352 to 367)			
DT90083	Step ladder process (368 to 383)			
DT90084	Step ladder process (384 to 399)			
DT90085	Step ladder process (400 to 415)			
DT90086	Step ladder process (416 to 431)			

Register No.	Register No.	Name	R	W
DT90087	Step ladder process (432 to 447)			
DT90088	Step ladder process (448 to 463)			
DT90089	Step ladder process (464 to 479)			
DT90090	Step ladder process (480 to 495)			
DT90091	Step ladder process (496 to 511)			
DT90092	Step ladder process (512 to 527)			
DT90093	Step ladder process (528 to 543)			
DT90094	Step ladder process (544 to 559)			
DT90095	Step ladder process (560 to 575)			
DT90096	Step ladder process (576 to 591)			
DT90097	Step ladder process (592 to 607)			
DT90098	Step ladder process (608 to 623)	Indicates the starting status of the step ladder process. When the process starts, the bit corresponding to its process number turns ON. Monitor using binary display. <Example> 15 11 7 3 0 (Bit no.) DT90100 655 651 647 643 640 (Process no.) 1: Starting 0: During stop The data can be written using the programming tool.	•	•
DT90099	Step ladder process (624 to 639)			
DT90100	Step ladder process (640 to 655)			
DT90101	Step ladder process (656 to 671)			
DT90102	Step ladder process (672 to 687)			
DT90103	Step ladder process (688 to 703)			
DT90104	Step ladder process (704 to 719)			
DT90105	Step ladder process (720 to 735)			
DT90106	Step ladder process (736 to 751)			
DT90107	Step ladder process (752 to 767)			
DT90108	Step ladder process (768 to 783)			
DT90109	Step ladder process (784 to 799)			

Register No.	Register No.	Name	R	W
DT90110	Step ladder process (800 to 815)			
DT90111	Step ladder process (816 to 831)			
DT90112	Step ladder process (832 to 847)			
DT90113	Step ladder process (848 to 863)			
DT90114	Step ladder process (864 to 879)			
DT90115	Step ladder process (880 to 895)			
DT90116	Step ladder process (896 to 911)			
DT90117	Step ladder process (912 to 927)			
DT90118	Step ladder process (928 to 943)			
DT90119	Step ladder process (944 to 959)			
DT90120	Step ladder process (960 to 975)			
DT90121	Step ladder process (976 to 991)			
DT90122	Step ladder process (992 to 999) (High-order byte is not used.)			
DT90123	Reserved for system	-		
DT90124	COM SEND/RECV instruction end code	Refer to the description of the F145 (SEND) and F146 (RECV) instructions.		
DT90125	Reserved for system	-		
DT90126	Forced I/O in-process unit number	Used by the system.		
DT90127 ～DT90139	Reserved for system	-		
DT90140	MEWNET-W0 PLC link 0 status	Number of receptions of PLC link 0	●	
DT90141		PLC link 0 reception interval (current value) (x 2.5 ms)		
DT90142		PLC link 0 reception interval (min. value) (x 2.5 ms)		
DT90143		PLC link 0 reception interval (max. value) (x 2.5 ms)		
DT90144		Number of transmission of PLC link 0		
DT90145		PLC link 0 transmission interval (current value) (x 2.5 ms)		
DT90146		PLC link 0 transmission interval (min. value) (x 2.5 ms)		

Register No.	Register No.		Name	R	W
DT90147			PLC link 0 transmission interval (max. value) (x 2.5 ms)		
DT90148	MEWNET-W0 PLC link 1 status		Number of receptions of PLC link 1	●	
DT90149			PLC link 1 reception interval (current value) (x 2.5 ms)		
DT90150			PLC link 1 reception interval (min. value) (x 2.5 ms)		
DT90151			PLC link 1 reception interval (max. value) (x 2.5 ms)		
DT90152			Number of transmission of PLC link 1		
DT90153			PLC link 1 transmission interval (current value) (x 2.5 ms)		
DT90154			PLC link 1 transmission interval (min. value) (x 2.5 ms)		
DT90155			PLC link 1 transmission interval (max. value) (x 2.5 ms)		
DT90156	MEWNET-W0 PLC link 0 status		PLC link 0 work for measuring reception interval	●	
DT90157			PLC link 0 work for measuring transmission interval		
DT90158	MEWNET-W0 PLC link 1 status		PLC link 1 work for measuring reception interval	●	
DT90159			PLC link 1 work for measuring transmission interval		
DT90160	MEWNET-W0 PLC link 0 unit number		PLC link 0 unit number is stored.	●	
DT90161	MEWNET-W0 PLC link 0 error flag		The content of error of PLC link 0 is stored.	●	
DT90162 to DT90169	Reserved for system		-		
DT90170	MEWNET-W0 PLC link 0 status		PLC link address duplicate destination	●	
DT90171			Number of lost tokens		
DT90172			Number of duplicate tokens		
DT90173			Number of no-signal states		
DT90174			Number of times of reception of undefined commands		
DT90175			Number of reception checksum errors		
DT90176			Number of received data format errors		
DT90177			Number of transmission errors		
DT90178			Number of procedure errors		
DT90179			Number of duplicate master units		
DT90180 to DT90218	Reserved for system		-		
DT90219	Unit number switching for DT90220 to DT90251		0: Unit Nos. 1 to 8, 1: Unit Nos. 9 to 16	●	
DT90220	PLC link unit No. 1 or 9	System registers 40	System register settings related to the PLC link function of each unit number are stored as below. <Example> When the content of DT90219 is 0:	●	

Register No.	Register No.		Name	R	W
		and 41	 If the content of the system register no. 46 of the master unit is the standard setting, the values in the master unit will be copied for system register nos. 46 and 47. If the content of the system register no. 46 of the master unit is the reverse setting, the nos. 40 to 45 and 47 corresponding to those of the master unit will be 50 to 55 and 57, and 46 will be set as it is. Also, the contents of the system register nos. 40 to 45 corresponding to those of other units will be the values obtained by correcting the received values, and the contents of nos. 46 and 57 of the master unit will be set for nos. 46 and 47.		
DT90221		System registers 42 and 43			
DT90222		System registers 44 and 45			
DT90223		System registers 46 and 47			
DT90224	PLC link unit No. 2 or 10	System registers 40 and 41			
DT90225		System registers 42 and 43			
DT90226		System registers 44 and 45			
DT90227		System registers 46 and 47			
DT90228	PLC link unit No. 3 or 11	System registers 40			

Register No.	Register No.		Name	R	W
		and 41			
DT90229		System registers 42 and 43			
DT90230		System registers 44 and 45			
DT90231		System registers 46 and 47			
DT90232	PLC link unit No. 4 or 12	System registers 40 and 41			
DT90233		System registers 42 and 43			
DT90234		System registers 44 and 45			
DT90235		System registers 46 and 47			
DT90236	PLC link unit No. 5 or 13	System registers 40			

Register No.	Register No.		Name	R	W
		and 41			
DT90237		System registers 42 and 43			
DT90238		System registers 44 and 45			
DT90239		System registers 46 and 47			
DT90240	PLC link unit No. 6 or 14	System registers 40 and 41			
DT90241		System registers 42 and 43			
DT90242		System registers 44 and 45			
DT90243		System registers 46 and 47			
DT90244	PLC link unit No. 7 or 15	System registers 40			

Register No.	Register No.			Name	R	W
			and 41			
DT90245			System registers 42 and 43			
DT90246			System registers 44 and 45			
DT90247			System registers 46 and 47			
DT90248	PLC link unit No. 8 or 16		System registers 40 and 41			
DT90249			System registers 42 and 43			
DT90250			System registers 44 and 45			
DT90251			System registers 46 and 47			
DT90252 ～DT90256	Reserved for system			-		
DT90300	Elapsed value area	Lower word	HSC-CH0	The counting area of the high-speed counter Control Unit input CH0 (X0) or (X0, X1)	•	•
DT90301		Higher word				

Register No.	Register No.			Name	R	W
DT90302	Target value area	Lower word		When executing the F166 (HC1S) or F167 (HC1R) instruction, the target value is set.		
DT90303		Higher word				
DT90304	Elapsed value area	Lower word	HSC-CH1	The counting area of the high-speed counter Control Unit input (X1).		
DT90305		Higher word				
DT90306	Target value area	Lower word		When executing the F166 (HC1S) or F167 (HC1R) instruction, the target value is set.		
DT90307		Higher word				
DT90308	Elapsed value area	Lower word	HSC-CH2	The counting area of the high-speed counter Control Unit input (X2) or (X2, X3).		
DT90309		Higher word				
DT90310	Target value area	Lower word		When executing the F166 (HC1S) or F167 (HC1R) instruction, the target value is set.		
DT90311		Higher word				
DT90312	Elapsed value area	Lower word	HSC-CH3	The counting area of the high-speed counter Control Unit input (X3).		
DT90313		Higher word				
DT90314	Target value area	Lower word		When executing the F166 (HC1S) or F167 (HC1R) instruction, the target value is set.		
DT90315		Higher word				
DT90316	Elapsed value area	Lower word	HSC-CH4	The counting area of the high-speed counter Control Unit input (X4) or (X4, X5).		
DT90317		Higher word				
DT90318	Target value area	Lower word		When executing the F166 (HC1S) or F167 (HC1R) instruction, the target value is set.		
DT90319		Higher word				
DT90320	Elapsed value area	Lower word	HSC-CH5	The counting area of the high-speed counter Control Unit input (X5).		
DT90321		Higher word				
DT90322	Target value area	Lower word		When executing the F166 (HC1S) or F167 (HC1R) instruction, the target value is set.		
DT90323		Higher word				
DT90324 ～DT90335	Reserved for system			-		

(Note 1) When the F166 (HC1S) or F167 (HC1R) instruction is used to control pulse output channels, the target value for match ON/OFF is stored.

Register No.	Register No.		Name	R	W
DT90336 ～DT90363	Reserved for system				
DT90370	Control flag monitor area	HSC-CH0	When the F0 (MV), DT90052 instruction is used for HSC control, the set value for the target channel is stored for the each channel.	•	
DT90371		HSC-CH1			
DT90372		HSC-CH2			
DT90373		HSC-CH3			

Register No.	Register No.			Name	R	W
DT90374			HSC-CH4			
DT90375			HSC-CH5			
DT90376 ～DT90379	Reserved for system					
DT90380	Control flag monitor area (Transistor output type only)		PLS-CH0	When the F0 (MV), DT90052 instruction is used for pulse output control, the set value for the target channel is stored for the each channel.	•	
DT90381			PLS-CH1			
DT90382			PLS-CH2			
DT90383			PLS-CH3			
DT90380 ～DT90389	Reserved for system					
DT90400	Elapsed value area	Lower word	PLS-CH0	Valid only for transistor operation type	•	•
DT90401		Higher word				
DT90402	Target value area	Lower word				
DT90403		Higher word				
DT90404	Target value area for match ON/OFF	Lower word				
DT90405		Higher word				
DT90406	Initial speed correction speed	Lower word		The initial speed from the calculation result is stored.		
DT90407	Deceleration lower limit speed			Lower limit speed for speed change		
DT90408	Acceleration disabled area start position	Lower word		Acceleration cannot be performed beyond this position during speed change.		
DT90409		Higher word				
DT90410	Elapsed value area	Lower word	PLS-CH1	Valid only for transistor operation type		•
DT90411		Higher word				
DT90412	Target value area	Lower word				
DT90413		Higher word				
DT90414	Target value	Lower word				

Register No.	Register No.			Name	R	W
DT90415	area for match ON/OFF	Higher word				
DT90416	Initial speed correction speed	Lower word		The initial speed from the calculation result is stored.		
DT90417	Deceleration lower limit speed			Lower limit speed for speed change		
DT90418	Acceleration disabled area start position	Lower		Acceleration cannot be performed beyond this position during speed change.		
DT90419		Higher				
DT90420	Elapsed value area	Lower word	PLS-CH2	Valid only for transistor operation type		•
DT90421		Higher word				
DT90422	Target value area	Lower word				
DT90423		Higher word				
DT90424	Target value area for match ON/OFF	Lower word				
DT90425		Higher word				
DT90426	Initial speed correction speed	Lower word		The initial speed from the calculation result is stored.		
DT90427	Deceleration lower limit speed			Lower limit speed for speed change		
DT90428	Acceleration disabled area start position	Lower		Acceleration cannot be performed beyond this position during speed change.		
DT90429		Higher				
DT90430	Elapsed value area	Lower word	PLS-CH3	Valid only for transistor operation type		•
DT90431		Higher word				
DT90432	Target value area	Lower word				
DT90433		Higher word				
DT90434	Target value	Lower word				

Register No.	Register No.			Name	R	W
DT90435	area for match ON/OFF	Higher word				
DT90436	Initial speed correction speed	Lower word		The initial speed from the calculation result is stored.		
DT90437	Deceleration lower limit speed			Lower limit speed for speed change		
DT90438	Acceleration disabled area start position	Lower		Acceleration cannot be performed beyond this position during speed change.		
DT90439		Higher				

31.5 Communication Commands

31.5.1 MEWTOCOL-COM

Type of instruction	Code	Description
Read contact area	RC (RCS) (RCP) (RCC)	Reads the ON/OFF status of contacts. ● Specifies only one point. ● Specifies multiple contacts. ● Specifies a range in word units.
Write contact area	WC (WCS) (WCP) (WCC)	Turns contacts ON or OFF. ● Specifies only one point. ● Specifies multiple contacts. ● Specifies a range in word units.
Read data area	RD	Reads the contents of a data area.
Write data area	WD	Writes data to a data area.
Read timer/counter set value area	RS	Reads the timer/counter set value.
Write timer/counter set value area	WS	Writes a timer/counter set value.
Read timer/counter elapsed value area	RK	Reads the timer/counter elapsed value.
Write timer/counter elapsed value area	WK	Writes the timer/counter elapsed value.
Register or Reset contacts monitored	MC	Registers the contact to be monitored.
Register or Reset data monitored	MD	Registers the data to be monitored.
Monitoring start	MG	Monitors a registered contact or data using MD and MC.
Preset contact area (fill command)	SC	Fills the area of a specified range with a 16-point ON/OFF pattern.
Preset data area (fill command)	SD	Writes the same contents to the data area of a specified range.
Read system register	RR	Reads the contents of a system register.
Write system register	WR	Specifies the contents of a system register.
Read the status of PLC	RT	Reads PLC specification, an error code when an error occurs, etc.
Remote control	RM	Switches the operation mode of the programmable controller.
Abort	AB	Aborts communication.

(Note 1) Some devices cannot be accessed due to format limitations of MEWTOCOL-COM communication commands.

31.6 Error code

31.6.1 List of Syntax Check Errors

Error codes 1 to 8

Code	Name	Operat ion	Error contents and steps to take
E1	Syntax error	Stop	A program with a syntax error has been written. Change to PROG. mode and correct the error.
E2	Duplicated output error	Stop	The relay is used in the ‘Out’ instruction or ‘Keep’ instruction more than once. It also occurs when using the same timer/ counter number. Change to PROG. mode and correct the program so that one relay is not used for two or more instructions. Or, set the duplicated output to "enable" in the system register no. 20. A timer/counter instruction double definition error will be detected even if double output permission has been selected.
E3	Not paired error	Stop	For instructions which must be used in a pair (such as JP and LBL), one instruction is either missing or in an incorrect position. Change to PROG. mode and enter the two instructions which must be used in a pair in the correct positions.
E4	Parameter mismatch error	Stop	An instruction has been written which does not agree with system register settings. The number setting in a program does not agree with the timer/counter range setting. Change to PROG. mode, check the system register settings, and change so that the settings and the instruction agree.
E5	Program area error (Note 1)	Stop	An instruction which must be written in a specific area (main program area or subprogram area) has been written to a different area (for example, a subroutine SUB to RET is placed before an ED instruction). Change to PROG. mode and enter the instruction into the correct area.
E6	Compile memory full	Stop	The program is too large to compile in the program memory. Change to PROG. mode and reduce the total number of steps for the program.
E7	High-level instruction type error	Stop	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.
E8	High-level instruction operand combination error	Stop	There is an incorrect operand in an instruction which requires a specific combination of operands (for example, the operands must all be of a certain type). Enter the correct combination of operands.

(Note 1) The error codes E2 and E5 are detected even when rewriting data with grammatical errors in RUN mode. In this case, nothing will be written into the control unit. The operation continues.

31.6.2 Self-diagnostic Errors

Code	Name	Operation	Error contents and steps to take
E26	User ROM error	Stop	There may be an error in the hardware. Please contact your dealer.
E27	Restrictions on the number of units installed	Stop	The number of the installed units exceeds the limitation. Turn off the power and check if the combinations of the units are within the limitation range.
E31	Interrupt error 1	Stop	An interrupt occurred although no interrupt request was made. There may be a malfunction due to a hardware error or some other cause. Turn off the power and check and maintain the noise environment.
E32	Interrupt error 2	Stop	There is no interrupt program for the interrupt that occurred. There may be a malfunction due to a hardware error or some other cause.
E34	I/O status error	Stop	An abnormal unit is installed. Check the slot number with DT90036, and replace the abnormal unit with a normal unit.
E42	I/O verification error	Select	The connection state of the I/O unit (Expansion Unit) has changed from the state that it was in when the power was turned ON. Use the special data registers DT90010 and DT90011 to check the I/O unit whose connection state has changed. Alternatively, check the fitting state of the expansion connector. The operation can be continued or stopped by selecting "1: Run" or "0: Stop" in system register No. 23. In the tool software, the error can also be checked by "Verification Error (I/O Verification Error)" in the status display function.
E45	Operation error occurred	Select	Inexecutable operation error occurs. The address of the operation error can be confirmed using either special data register DT90017 or DT90018. The operation can be continued or stopped by selecting "1: Run" or "0: Stop" in system register No. 26. In the tool software, the error can also be checked by clicking the [Operation errors] button in the status display dialog box.
From E100 E199	Self-diagnostic error set by F148	Stop	An error optionally set by the high-level instruction F148 has occurred. Take action according to the specified detection conditions.
From E200 E299		Operation continues	

31.6.3 List of MEWTOCOL-COM/DAT Communication Error Codes

Code	Name	Description of error
!21	NACK error	Link system error
!22	WACK error (Partner receive buffer overflow)	Link system error
!23	Unit number duplicate	Link system error
!24	Transmission format error	Link system error
!25	Link unit hardware error	Link system error
!26	Abnormal unit number setting	A command was received that cannot be used globally (station number FF).
!27	No support error	Link system error
!28	No response error (Waiting for response)	Link system error
!29	Buffer closed error	Link system error
!30	Timeout (Transmission not possible)	Link system error
!32	Transfer disabled error (Master unit buffer overflow)	Link system error
!33	Communication stop	Link system error
!36	No partner unit error	Link system error
!38	Other communication errors	Link system error
!40	BCC error	A transmission error has occurred in the received data.
!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.
!50	Link setting error	A nonexistent route number has been specified. Check the route number in the master unit specification.
!51	Transmission timeout error	Data cannot be sent to other units due to congestion in the send buffer.
!52	Transmission not possible error	Data cannot be sent to other units. (Link unit out of control, etc.)
!53	Busy error	Command processing cannot be accepted because multiple frames are being processed. Or the command being processed is stuck to prevent the acceptance.
!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.

Code	Name	Description of error
!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.
!70	SIM over error	Program area was exceeded during a program write process.
!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.
!93	Invalid server error	A command was executed when the target server was invalid.
!94	Certificate write error	Certificate write was executed using an incorrect procedure.
!97	Server connected error	A command was executed when the target server was connected.

31.7 BIN/HEX/BCD Code Correspondence Table

Decimal (Decimal)	Hexadecimal (Hexadecimal)	BIN Binary (Binary)		BCD Binary Coded Decimal (4-Digit) (Binary Coded Decimal)			
0	0000	00000000	00000000	0000	0000	0000	0000
1	0001	00000000	00000001	0000	0000	0000	0001
2	0002	00000000	00000010	0000	0000	0000	0010
3	0003	00000000	00000011	0000	0000	0000	0011
4	0004	00000000	00000100	0000	0000	0000	0100
5	0005	00000000	00000101	0000	0000	0000	0101
6	0006	00000000	00000110	0000	0000	0000	0110
7	0007	00000000	00000111	0000	0000	0000	0111
8	0008	00000000	00001000	0000	0000	0000	1000
9	0009	00000000	00001001	0000	0000	0000	1001
10	000A	00000000	00001010	0000	0000	0001	0000
11	000B	00000000	00001011	0000	0000	0001	0001
12	000C	00000000	00001100	0000	0000	0001	0010
13	000D	00000000	00001101	0000	0000	0001	0011
14	000E	00000000	00001110	0000	0000	0001	0100
15	000F	00000000	00001111	0000	0000	0001	0101
16	0010	00000000	00010000	0000	0000	0001	0110
17	0011	00000000	00010001	0000	0000	0001	0111
18	0012	00000000	00010010	0000	0000	0001	1000
19	0013	00000000	00010011	0000	0000	0001	1001
20	0014	00000000	00010100	0000	0000	0010	0000
21	0015	00000000	00010101	0000	0000	0010	0001
22	0016	00000000	00010110	0000	0000	0010	0010
23	0017	00000000	00010111	0000	0000	0010	0011
24	0018	00000000	00011000	0000	0000	0010	0100
25	0019	00000000	00011001	0000	0000	0010	0101
26	001A	00000000	00011010	0000	0000	0010	0110
27	001B	00000000	00011011	0000	0000	0010	0111
28	001C	00000000	00011100	0000	0000	0010	1000
29	001D	00000000	00011101	0000	0000	0010	1001
30	001E	00000000	00011110	0000	0000	0011	0000
31	001F	00000000	00011111	0000	0000	0011	0001
63	003F	00000000	00111111	0000	0000	0110	0011
255	00FF	00000000	11111111	0000	0010	0101	0101
9999	270F	00100111	00001111	1001	1001	1001	1001

31.8 ASCII Code Table, JIS8 Code Table

■ Reference ASCII code table

								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 JIS8 code table

b7	b6	b5	b4	b3	b2	b1	b0	Column / Row	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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
				0	0	0	0	0	NUL	TC$_7$ (DEL)	(SP)	0	@	P	`	p			Undefined	ー	タ	ミ		
				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	Undefined	Undefined	ヲ	カ	ニ	ヨ	Undefined	Undefined
				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 parts of the JIS8 code table.

(MEMO)

Record of Changes

The manual No. is written at the bottom of the cover page.

Date	Manual No.	Record of Changes
Apr. 2024	WUME-FP0RPGR-01	1st Edition

(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-FP0RPGR-01