SHARP®

Version 1.0
Produced in Dec. 2003

Sharp Programmable Controller

New Satellite *JW300*

Programiming Manual - Ladder Instruction Version

We thank you for your purchase of the SHARP programmable controller JW300.
This manual (Programming Manual - Ladder Instruction Version) mainly describes software elements of the JW300 such as data memory, system memory, and instruction words.
Carefully read this user's manual, hardware version and the instruction manual attached to each module so that you are able to operate JW300 properly, having thoroughly familiarized yourself with the functions of the system module and their operation method.
As for the description about hardware elements such as system configuration, etc. of the JW300, see the "JW300 User's Manual - Hardware Version."

Precautions

- When you plan to use SHARP programmable controllers (hereafter referred to as "PLCs"), you are requested to design each system so that even if a fault or malfunction occurs within the PLC, it will not lead to a serious accident in your system. You should incorporate back-up measures and fail-safe features in your system that will thoroughly protect your system from malfunctions if a fault or error occurs in the PLC.
- SHARP PLCs are designed and manufactured with the idea that they will be used in general applications in ordinary industries. Therefore, they must not be used in specific applications that can affect the health or safety of the public, such as nuclear power plants and other power generating plants. Such applications require a special warranty of quality that SHARP explicitly does NOT offer for these PLCs. However, if a user will certify that he/she does not requires a special quality warranty on the PLC, and will limit the use of the PLC to non critical areas of these applications, SHARP will agree to such use.
  If you are planning to use SHARP PLCs for applications that may affect the lives of human beings and property, and you need particularly high reliability performance, such as in the fields of aviation, medicine, transportation, combustion and fuel processing equipment, passenger cars, amusement park rides, and safety equipment, please contact our sales division so that we can confirm the required specifications.

Notes

- Though this manual is produced with the almost care, if you have any questions and inquiries, please feel free to contact our dealers.
- The whole or partial photocopy of this booklet is prohibited.
- Contents of this booklet may be revised for improvement without notice.

Table of contents

Chapter 1 Outline

The New Satellite JW300 series are high-speed, high-performance programmable controllers for medium and large-scale control systems. These are high-level models of the JW30H series.

■ Features

(1) High-speed processing and large memory capacity

- High processing speeds of 33 ns for basic instructions, and 132 ns for application instructions (The overall processing speed will be approximately 20% faster than our conventional JW30H model.)
- Large, 256 K-word maximum capacity for program memory and 8 M-bytes maximum for file registers (both approximately 4 times larger than that of the JW30H).

(2) Compatible with memory cards

- Programs and parameters can be backed up on CF cards.
- Extensions to file memory, logging data, etc. can be stored on SRAM cards.

(3) Equipped with a USB port

The JW300 can exchange commands and data with PCs, at high speeds, through its USB port.

(4) Equipped with three ports for communication

There are two ports on the control module (one port on the JW-311CU/312CU) and one port on the I/O bus expansion adapter (JW-32EA), which can be used for communication. These make it easy to connect to a control terminal or image sensor camera.

(5) Structured programs / block operation

- You can separate the programs that run on the JW300 into a few blocks so that the PLC can operate various machines independently, for trial operations and other purposes.
- Each program block can be further separated into sub programs to save programming effort. This will make it possible to design some programs in parallel. These sub programs can be handled as standardized modules and can be reused.

(6) Built in faulty equipment diagnosis function

Just program relay numbers and monitor times and the PLC can monitor facilities. This feature makes for significant savings in writing ladder programs and for detecting errors.

(7) A variety of models

- Ten control module models are available. You can choose the one that best matches your system's control scale and budget.
- All of the I/O modules and special I/O modules available for the JW20H/30H series can be used with the JW300. Optional modules for the JW20H/30H also can be used if they are labeled as "compatible with the JW300."
- A Windows version of our ladder logic programming software, the JW-300SP, is available to support editing of structured programs.
  The JW-15PG hand-held programmer is available to modify and monitor programs on site.

(8) Compatible with various open networks

The JW300 series is Ethernet compatible for communication, FL-net compatible for control, and DeviceNet compatible for the field. It is also AS-I compatible for sensor applications. These devices can exchange data with various layers, without any barriers.

Chapter 2 Data memory

2-1 File address

Data memory of the control module (JW-3**CU) varies with each mode.

File address (capacity)					
JW-311CU JW-312CU	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
$00000000_{(8)}$ to $00073777_{(8)}$ (30 K-bytes)	$00000000_{(8)}$ to $00105777_{(8)}$ (35 K-bytes)	$00000000_{(8)}$ to $00177777_{(8)}$ (64 K-bytes)			
- Relay 30720 points (3.75 K-bytes) - TMR/CNT contact points 1024 points (0.25 K-bytes) - TMR/CNT/MD current value (2 K-bytes) - Register (24 K-bytes)	- Relay 53248 points (6.5 K-bytes) - TMR/CNT contact points 2048 points (0.5 K-bytes) - TMR/CNT/MD current value (4 K-bytes) - Register (24 K-bytes)	- Relay 180224 points (22 K-bytes) - TMR/CNT contact points 8192 points (2 K-bytes) - TMR/CNT/MD current value (16 K-bytes) - Register (24 K-bytes)			
File register ➡	$00200000_{(8)}$ to $00277777_{(8)}$ (32 K-bytes)	$00200000_{(8)}$ to $00577777_{(8)}$ (128 K-bytes)	$00200000_{(8)}$ to $02177777_{(8)}$ (512 K-bytes)	$00200000_{(8)}$ to $10177777_{(8)}$ (2048 K-bytes)	$00200000_{(8)}$ to $40177777_{(8)}$ (8192 K-bytes)

● Memory map => See page 2-2 to 2-5.

Reference

- The JW30H has a file register from file 1 to 3, 10 to $2C_{(H)}$. 64K bytes of each file (16 K bytes for file 1) are independent. On the contrary, the JW300 has continuous addresses up to 8192 K-bytes.
  In addition, if a program of the JW30H is converted to one for the JW300 using the JW-300SP (ladder logic programming software), files 1 to 3 and 10 to $2C_{(H)}$ of the JW30H are converted to "file registers of the JW300."

2-2 Memory map

This section describes the memory map of the data memory in order of byte address (page 2-2 to 2-4) and in order of file address (page 2-5).

[1] In order of byte address

Byte address (capacity) of "relay, TMR/CNT contact points, TMR/CNT/MD current value" and "file registers" varies with model of control module (JW-3**CU). => (1) and (3).

Byte address of "registers" is common for all the models of control module. => (2).

(1) Byte address (file address) of relay, TMR/CNT contact points, TMR/CNT/MD current value.

● **JW-311CU/312CU**

Area	Byte address(8) (Relay number(8))	File address(8)	Capacity (bytes)
Relay * (30720 points)	ㅋ00000 to ㅋ01577(000000 to 015777)	00000000 to 00001577	3840 (3.75 K)
	ㅋ02000 to ㅋ07577(020000 to 075777)	00030000 to 00035577	
TMR/CNT contacts (1024 points)	T or C 00000 to 00777	00001600 to 00001777	256 (0.25 K)
	T or C 01000 to 01777	00035600 to 00035777	
TMR/CNT/MD current value	b00000 to b01777	00002000 to 00003777	2048 (2 K)
	b02000 to b03777	00026000 to 00027777	

● **JW-321CU/322CU**

Area	Byte address(8) (Relay number(8))	File address(8)	Capacity (bytes)
Relay * (53248 points)	ㅋ00000 to ㅋ01577(000000 to 015777)	00000000 to 00001577	6656 (6.5 K)
	ㅋ02000 to ㅋ07577(020000 to 075777)	00030000 to 00035577	
	ㅋ10000 to ㅋ15377(100000 to 153777)	00074000 to 00101377	
TMR/CNT contact point (2048 points)	T or C 00000 to 00777	00001600 to 00001777	512 (0.5 K)
	T or C 01000 to 01777	00035600 to 00035777	
	T or C 02000 to 03777	00101400 to 00101777	
TMR/CNT/MD current value	b00000 to b01777	00002000 to 00003777	4096 (4 K)
	b02000 to b03777	00026000 to 00027777	
	b04000 to b07777	00102000 to 00105777	

● **JW-331CU/332CU,JW-341CU/342CU,JW-352CU,JW362CU**

Area	Byte address(8) (Relay number(8))	File address(8)	Capacity (bytes)
Relay * (180224 points)	ㅋ00000 to ㅋ01577(000000 to 015777)	00000000 to 00001577	22528 (22 K)
	ㅋ02000 to ㅋ07577(020000 to 075777)	00030000 to 00035577	
	ㅋ10000 to ㅋ54377(100000 to 543777)	00074000 to 00140377	
TMR/CNT contacts (8192 points)	T or C 00000 to 00777	00001600 to 00001777	2048 (2 K)
	T or C 01000 to 01777	00035600 to 00035777	
	T or C 02000 to 17777	00140400 to 00143777	
TMR/CNT/MD current value	b00000 to b01777	00002000 to 00003777	16384 (16 K)
	b02000 to b03777	00026000 to 00027777	
	b04000 to b37777	00144000 to 00177777	

*In the relay areas, the following fixed areas are arranged.

Type (inside relay area)		Byte address(8)	Relay number(8)	File address(8)	Capacity
(1)	Relay for I/O link [mode 7, 8]	ㅋ00100 to ㅋ00177	001000 to 001777	00000100 to 00000177	512 points (64 bytes)
(2)	Special relay	ㅋ00730 to ㅋ00737	007300 to 007377	00000730 to 00000737	64 points (8 bytes)
(3)	Relay for option module	ㅋ01000 to ㅋ01477	010000 to 014777	00001000 to 00001477	2560 points (320 bytes)
(4)	Flag for option module	ㅋ01500 to ㅋ01567	015000 to 015677	00001500 to 00001567	448 points (56 bytes)
(5)	Flag for I/O link	ㅋ01570 to ㅋ01577	015700 to 015777	00001570 to 00001577	64 points (8 bytes)
(6)	Relay for I/O link [mode 1 to 6]	ㅋ02000 to ㅋ02377	020000 to 023777	00030000 to 00030377	2048 points (256 bytes)
(7)	Relay 1 for special I/O module [basic system]	ㅋ03000 to ㅋ03777	030000 to 037777	00031000 to 00031777	4096 points (512 bytes)
(8)	Relay for special I/O module [remote I/O slave station]	ㅋ04000 to ㅋ04177	040000 to 041777	00032000 to 00032177	1024 points (128 bytes)
(9)	Relay 2 for special I/O module [basic system]	ㅋ04200 to ㅋ05177	042000 to 051777	00032200 to 00033177	4096 points (512 bytes)

(2) Register byte address (file address): Common for all models

Byte address (capacity) of "register" is common for all the models (JW-3**CU).

Area	Byte address(8)	File address(8)	Capacity (bytes)	
Register	009000 to 009777	00004000 to 00004777	512	24K
	019000 to 019777	00005000 to 00005777	512	
	029000 to 029777	00006000 to 00006777	512	
	039000 to 039777	00007000 to 00007777	512	
	049000 to 049777	00010000 to 00010777	512	
	059000 to 059777	00011000 to 00011777	512	
	069000 to 069777	00012000 to 00012777	512	
	079000 to 079777	00013000 to 00013777	512	
	089000 to 089777	00014000 to 00014777	512	
	099000 to 099777	00015000 to 00015777	512	
	E0000 to E0777	00016000 to 00016777	512	
	E1000 to E1777	00017000 to 00017777	512	
	E2000 to E2777	00020000 to 00020777	512	
	E3000 to E3777	00021000 to 00021777	512	
	E4000 to E4777	00022000 to 00022777	512	
	E5000 to E5777	00023000 to 00023777	512	
	E6000 to E6777	00024000 to 00024777	512	
	E7000 to E7777	00025000 to 00025777	512	
	109000 to 109777	00036000 to 00036777	512	
	119000 to 119777	00037000 to 00037777	512	
	129000 to 129777	00040000 to 00040777	512	
	139000 to 139777	00041000 to 00041777	512	
	149000 to 149777	00042000 to 00042777	512	
	159000 to 159777	00043000 to 00043777	512	
	169000 to 169777	00044000 to 00044777	512	
	179000 to 179777	00045000 to 00045777	512	
	189000 to 189777	00046000 to 00046777	512	
	199000 to 199777	00047000 to 00047777	512	
	209000 to 209777	00050000 to 00050777	512	
	219000 to 219777	00051000 to 00051777	512	
	229000 to 229777	00052000 to 00052777	512	
	239000 to 239777	00053000 to 00053777	512	
	249000 to 249777	00054000 to 00054777	512	
	259000 to 259777	00055000 to 00055777	512	
	269000 to 269777	00056000 to 00056777	512	
	279000 to 279777	00057000 to 00057777	512	
	289000 to 289777	00060000 to 00060777	512	
	299000 to 299777	00061000 to 00061777	512	
	309000 to 309777	00062000 to 00062777	512	
	319000 to 319777	00063000 to 00063777	512	
	329000 to 329777	00064000 to 00064777	512	
	339000 to 339777	00065000 to 00065777	512	
	349000 to 349777	00066000 to 00066777	512	
	359000 to 359777	00067000 to 00067777	512	
	369000 to 369777	00070000 to 00070777	512	
	379000 to 379777	00071000 to 00071777	512	
	389000 to 389777	00072000 to 00072777	512	
	Z000 to Z377 *	00073000 to 00073777	512	

* Z register is arranged in units of two bytes.

(3) Byte address (file address) of file register

● **JW-311CU/312CU**

These models do not have a file register.

● **JW-321CU/322CU**

Area	Byte address(8)	File address(8)	Capacity(bytes)
File register	00000000 to 00077777	00200000 to 00277777	32 K

● **JW-331CU/332CU**

Area	Byte address(8)	File address(8)	Capacity(bytes)
File register	00000000 to 00377777	00200000 to 00577777	128 K

● **JW-341CU/342CU**

Area	Byte address(8)	File address(8)	Capacity(bytes)
File register	00000000 to 01777777	00200000 to 02177777	512 K

● **JW-352CU**

Area	Byte address(8)	File address(8)	Capacity(bytes)
File register	00000000 to 07777777	00200000 to 10177777	2048 K

● **JW-362CU**

Area	Byte address(8)	File address(8)	Capacity(bytes)
File register	00000000 to 37777777	00200000 to 40177777	8192 K

[2] File address order arrangement

Data memory is arranged as follows in the order of file address.

File address(8)	JW-311CU/312CU	JW-321CU/322CU	JW-331CU/332CU JW-341CU/342CU JW-352CU JW-362CU	Byte address
00000000		Relay (7168 points)		⊐00000
00001577				⊐01577
00001600		TMR/CNT contents of 00000 to 00777 (512 points)		
00001777				
00002000		Current value of TMR/CNT/MD 00000 to 00777 (1024 bytes)		b00000
00003777				b01777
00004000		Register 009000 to 099777 (5120 bytes)		009000
00015777				099777
00016000		Register E0000 to E7777 (4096 bytes)		E0000
00025777				E7777
00026000		Current value of TMR/CNT 01000 to 01777 (1024 bytes)		b02000
00027777				b03777
00030000		Relay (23552 points)		⊐02000
00035577				⊐07577
00035600		TMR/CNT contents of 01000 to 01777 (512 points)		
00035777				
00036000		Register 019000 to 199777 (5120 bytes)		109000
00047777				199777
00050000		Register 209000 to 299777 (5120 bytes)		209000
00061777				299777
00062000		Register 309000 to 389777 (4608 bytes)		309000
00072777				389777
00073000		Register Z000 to Z377 (512 bytes)		Z000
00073777				Z377
		00074000 ⊐10000 Relay (22528 points) 00101377 ⊐15377	00074000 ⊐10000 Relay (149504 points) 00140377 ⊐54377	
		00101400 Contact of TMR/CNT 02000 to 03777(1024 points) 00101777	00140400 Contents of TMR/CNT 02000 to 17777 (7168 points) 00143777	
		00102000 b04000 Current value of TMR/CNT/MD 02000 to 03777 (2048 bytes) 00105777 b07777	00144000 b04000 Current of TMR/CNT/MD 02000 to 17777 (14336 bytes) 00177777 b37777	
		File register *	File register *	

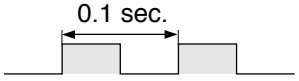
* As for the file register, see page 2-4.

2-3 Relay area

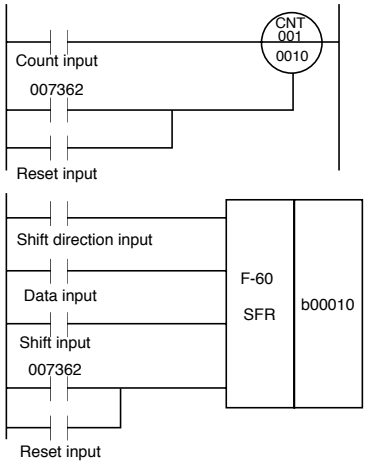
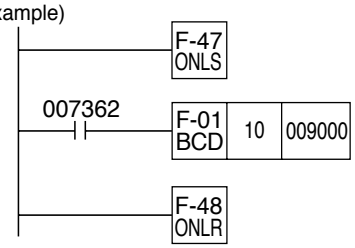
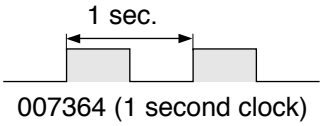
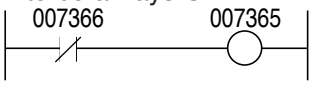
[1] Special relay

Sixty-four points of relay numbers 007300 to 007377 (Ⅎ00730 to Ⅎ00737) are a special relay area.

- Use the special relays (except 007365) in the area written by a CPU. With the user program, use them as contact points and a source of application instructions.
  These cannot be used as destination of the OUT and application instructions. Especially, be careful for instructions, distribution, extraction instructions, package data transfer instructions that handle data memory of two bytes or more.
- Do not use the reserved area (see the table below) for the user program.

Relay No.	Description
007300 to 007327	Reserved area
007330	**MW flag** - This flag goes ON when the program memory is changed, and the MW lamp of the control module (JW-3**CU) blinks.
007331	**MW reset** - Use to reset the MW flag (007330). While the MW flag is ON, turn the MW reset (007331) from OFF to ON; the MW flag goes OFF and the MW lamp of the control module goes OFF. At the same time, the MW reset goes OFF, too. In order to disable (turn OFF) the MW flag and MW lamp, always turn OFF the MW reset relay (007331) using the user program.
007332 to 007337	Reserved area
007340 to 007347	**Store error code** - As a result of the self-diagnosis, if an error is found, the JW300 store its error code (page 5-2). - These are handled as byte address "Ⅎ00734" of special register. - When the JW300 recovers from the error status, the error code will be cleared.
007350 to 007353	Reserved area
007354	Non carry flag - Flag will be set according to the kind of operation when an application instruction is executed that may affect the flag.
007355	Error flag
007356	Carry flag
007357	Zero flag
007360	0.1 second clock - Used for the clock of the CNT instruction and other application instruction. (1 sec. clock => 007364)  007360 (0.1 second clock)
007361	Reserved area

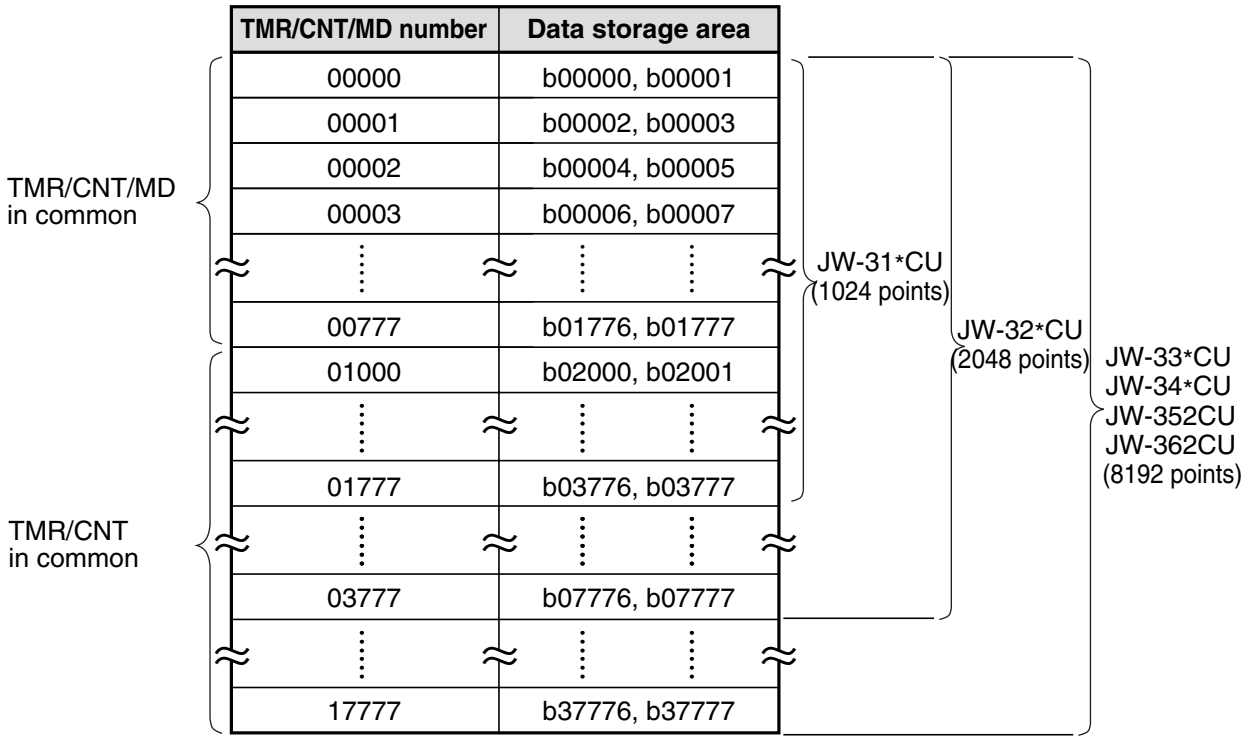
To next page

Relay No.	Contents
007362	**Initialize pulse** - Turns ON during one operation cycle immediately after the run mode of the control module is started. - Used to initially reset (initialize) a counter or shift register.  - When using the initializing pulse as the input of rise operation command, it is necessary to be used within the level operation command. (Example) 
007363	**Fuse blow** - "ON" at fuse blown detection of JW-262S (64-pointoutput module).
007364	**1-second clock** - Used for the clock of the CNT instruction and other application instruction.  (0.1 sec. clock => 007360)
007365	**Setting value change switch** - 07365 must be set ON by the program with the support tool such as programmer to set or reset a relay other than latched relay area (initially conditions is 007000 to 015777, 020000 to 075777). - Program to be always ON  Use the normally OFF contact 007366.
007366	**Normally OFF contact** - Used for the contact that programmed to be normally OFF (a-contact) or normally ON (b-contact).
007367	Reserved area
007370	Memory error
007371	CPU error
007372	Battery error
007373	I/O error
007374	Option error
007375	Special I/O module error flag
007376	Expansion power supply error flag
007377	Power source error

Self-diagnostic result
- If an error was met in a course of diagnosis, the relevant contact will be set ON.
- For details, refer to page 5-2, "Self-diagnosis."

2-4 TMR/CNT/MD data storage area

Registers having "b" in their top are areas to store current value of TMR/CNT and MD data of the MD instruction. The TMR/CNT/MD use two bytes per point. The TMR/CNT/MD numbers and "b*****" are as follows.



If any of "b*****" is specified for data processing instruction (F-00 etc.), current value of the TMR/CNT can be used for calculation.

The TMR/CNT/MD have three formats.

Type 1	Type 2	Type 3
—	TMR DTMR (BCD) UTMR (BCD)	DTMR (BIN) UTMR (BIN)
—	CNT DCNT (BCD) UCNT (BCD)	DCNT (BIN) UCNT (BIN)
MD	—	—

● **Data format for type 1**

	7	6	5	4	3	2	1	0	
MD	($x10^1$)				($x10^0$)				n
	8	4	2	1	8	4	2	1	
	Input information				($x10^2$)				n+1
	1	S_1	S_2	S_3	8	4	2	1	

- Numerical data is handled in BCD.
- n and n+1 represent the order of addresses.

● **Data format for type 2**

The TMR and CNT have a setting value range of 0 to 7999. "D" means down or decrement : "U" means up or increment.

	7	6	5	4	3	2	1	0	
TMR	($X10^0$)				($X10^{-1}$)				n
	8	4	2	1	8	4	2	1	
	Reset *	($X10^2$)			($X10^1$)				n+1
		4	2	1	8	4	2	1	
CNT	($X10^1$)				($X10^0$)				n
	8	4	2	1	8	4	2	1	
	Reset *	($X10^3$)			($X10^2$)				n+1
		4	2	1	8	4	2	1	
DTMR (BCD)	($X10^0$)				($X10^{-1}$)				n
	8	4	2	1	8	4	2	1	
	*	($X10^2$)			($X10^1$)				n+1
		4	2	1	8	4	2	1	
UTMR (BCD)	($X10^0$)				($X10^{-1}$)				n
	8	4	2	1	8	4	2	1	
	Reset *	($X10^2$)			($X10^1$)				n+1
		4	2	1	8	4	2	1	
DCNT (BCD)	($X10^1$)				($X10^0$)				n
	8	4	2	1	8	4	2	1	
	Reset *	($X10^3$)			($X10^2$)				n+1
		4	2	1	8	4	2	1	
UCNT (BCD)	($X10^1$)				($X10^0$)				n
	8	4	2	1	8	4	2	1	
	Reset *	($X10^3$)			($X10^2$)				n+1
		4	2	1	8	4	2	1	

- Numerical data is handled in BCD.
- n and n+1 represent the order of addresses.
- The JW300 program distinguishes between TMR and CNT, and between U and D.

● **Data format for type 3**

The TMR and CNT have a set value range of 0 to 32767. "D" means down or decrement : "U" means up or increment.

	7	6	5	4	3	2	1	0	
DTMR (BIN)	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	n
	Reset *	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	n+1
UTMR (BIN)	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	n
	Reset *	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	n+1
DCNT (BIN)	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	n
	Reset *	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	n+1
UCNT (BIN)	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	n
	Reset *	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	n+1

- Numerical data is handled in BIN (binary).

* While the TMR/CNT are operating, these bits go "1" (ON). If they do not measure, this bit goes "1" even resetting.

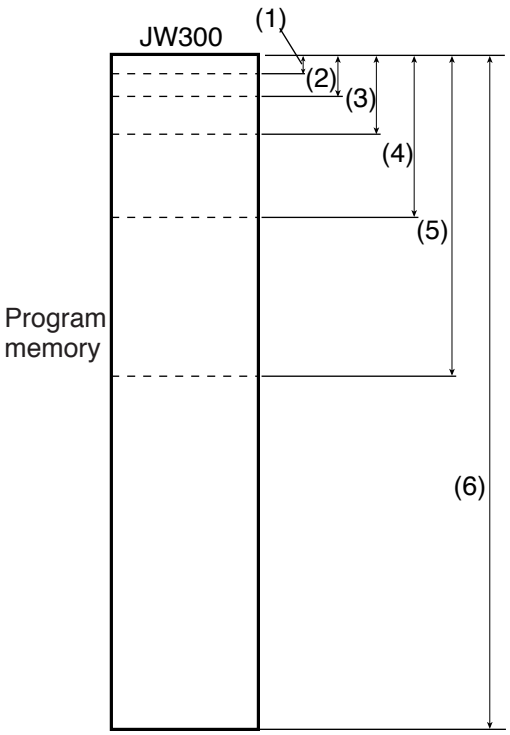
Chapter 3 Program memory, Parameter memory

3-1 Program memory

The program memory is the areas in which the user program is stored. As the JW300 begins to operate, the program is stated to read from the top address to do operation according to the program.

[1] Program addresses

In the JW300, the program capacity varies with model of control module (JW-3**CU).



(1) JW-311CU/312CU (8K words)
(2) JW-321CU/322CU (16K words)
(3) JW-331CU/332CU (32K words)
(4) JW-341CU/342CU (64K words)
(5) JW-352CU (128K words)
(6) JW-362CU (256K words)

- The program addresses are as follows.

Control module	Program memory capacity	Program address		
		Octal	Decimal	Hexadecimal
JW-311CU/312CU	8K words	000000 to 017777	000000 to 008191	00000 to 01FFF
JW-321CU/322CU	16K words	000000 to 037777	000000 to 016383	00000 to 03FFF
JW-331CU/332CU	32K words	000000 to 077777	000000 to 032767	00000 to 07FFF
JW-341CU/342CU	64K words	000000 to 177777	000000 to 065535	00000 to 0FFFF
JW-352CU	128K words	000000 to 377777	000000 to 131071	00000 to 1FFFF
JW-362CU	256K words	000000 to 777777	000000 to 262143	00000 to 3FFFF

There are 1-word, 2-word, 3-word and 4-word instructions.

	Instruction words
1-word instruction	STR, AND, etc.
2-word instruction	TMR, CNT, etc.
3-word instruction	F-00, F-01, etc.
4-word instruction	F-10, F-11, etc.

[2] Structuring program

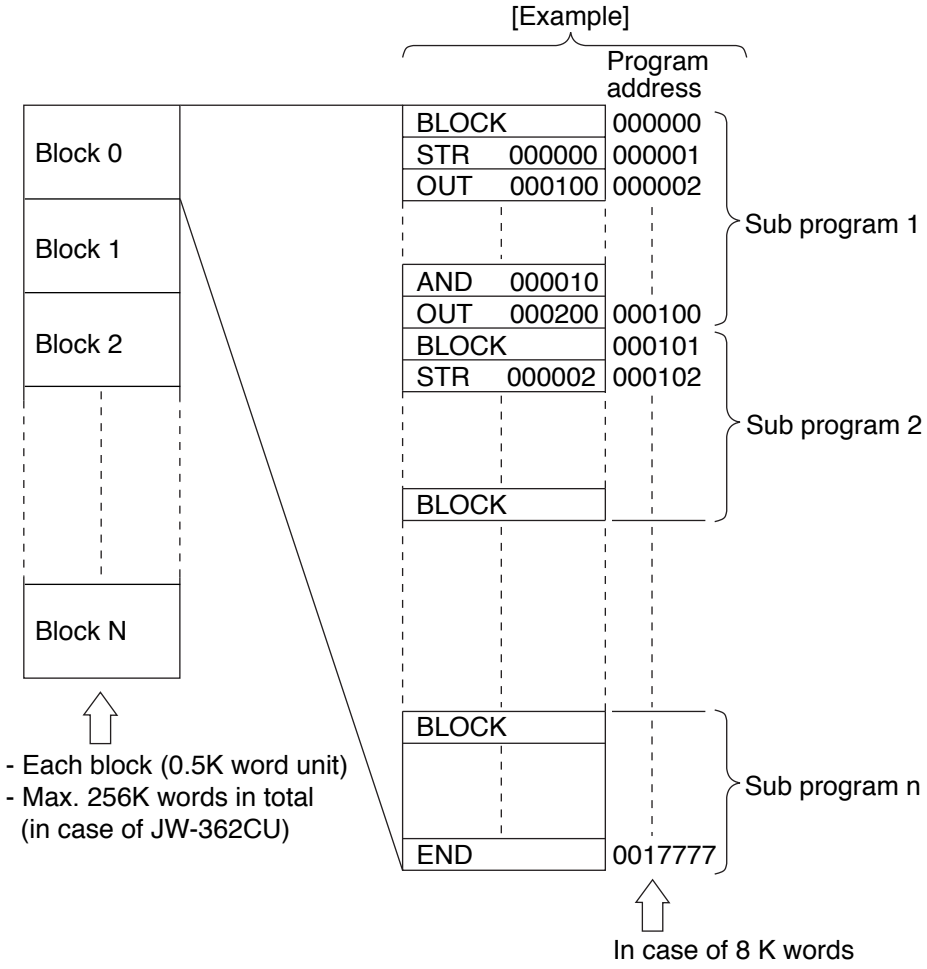
The JW300 can store multiple programs (from now called "blocks") in one control module.

The program capacity of each block is set in units of 0.5 K words using the JW-300SP (ladder logic programming software).

=> The last address of each block is written END instruction (F-40).

Each block can be programmed by separating it into multiple sub programs using the JW-300SP.

=> The top address of each sub program is written identification instruction (BLOCK).



The number of blocks and sub programs settable depend upon models of control module (JW-3**CU).

	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
No. of blocks (max.)	16	32	64	128	256	512
No. of sub programs(max.)	256	512	1024	2048	4096	8192

As for block operation, see "6-1 Block operation function."

Remarks

- The JW-15PG (hand-held programmer) cannot be used for separation into blocks and sub programs. It only can edit blocks and sub programs after they are separated using the JW-300SP.
- If a memory is cleared in units of block, the top address of a block is overwritten by BLOCK, the last address is written END instruction. Other addresses are written NOP instruction.

3-2 Parameter memory

This section describes parameter areas for special I/O modules and option modules on the JW300 system.

[1] Parameter for special I/O module

For the models of the special I/O modules listed below, set operation conditions to the special I/O parameter of the control module (JW-3**CU) using a support tools (JW-15PG, JW-300SP).

● JW-21HC, JW-22HC, JW-24AD, JW-22DA, JW-21PS, JW-21SU,

- The special I/O parameter area is determined by the special I/O module number switch (set value 0 to 7). One module uses 256 bytes.
- As for the details of the special I/O parameters, see respective user's manual of the special I/O module (JW-21HC etc.).

Setting value of module No. switch	Special I/O parameter address(8)							
	Rack 0		Rack 1		Rack 2		Rack 3	
0	T-00	000 to 377	T-10	000 to 377	T-20	000 to 377	T-30	000 to 377
1	T-01	000 to 377	T-11	000 to 377	T-21	000 to 377	T-31	000 to 377
2	T-02	000 to 377	T-12	000 to 377	T-22	000 to 377	T-32	000 to 377
3	T-03	000 to 377	T-13	000 to 377	T-23	000 to 377	T-33	000 to 377
4	T-04	000 to 377	T-14	000 to 377	T-24	000 to 377	T-34	000 to 377
5	T-05	000 to 377	T-15	000 to 377	T-25	000 to 377	T-35	000 to 377
6	T-06	000 to 377	T-16	000 to 377	T-26	000 to 377	T-36	000 to 377
7	T-07	000 to 377	T-17	000 to 377	T-27	000 to 377	T-37	000 to 377

Setting value of module No. switch	Special I/O parameter adddress(8)									
	Rack 4		Rack 5		Rack 6		Rack 7		Remote I/O slave station	
0	T-40	000 to 377	T-50	000 to 377	T-60	000 to 377	T-70	000 to 377	T-80	000 to 377
1	T-41	000 to 377	T-51	000 to 377	T-61	000 to 377	T-71	000 to 377	T-81	000 to 377
2	T-42	000 to 377	T-52	000 to 377	T-62	000 to 377	T-72	000 to 377	T-82	000 to 377
3	T-43	000 to 377	T-53	000 to 377	T-63	000 to 377	T-73	000 to 377	T-83	000 to 377
4	T-44	000 to 377	T-54	000 to 377	T-64	000 to 377	T-74	000 to 377	T-84	000 to 377
5	T-45	000 to 377	T-55	000 to 377	T-65	000 to 377	T-75	000 to 377	T-85	000 to 377
6	T-46	000 to 377	T-56	000 to 377	T-66	000 to 377	T-76	000 to 377	T-86	000 to 377
7	T-47	000 to 377	T-57	000 to 377	T-67	000 to 377	T-77	000 to 377	T-87	000 to 377

Reference

- With the JW30H, the special I/O modules cannot be inserted to rack 4 to 7.
  The special I/O parameter area uses 128 bytes per module.

[2] Parameter for option module

For the option module of the following models, specify operation conditions for them to the control modules (JW-3**CU) option parameter using a support tool (JW-15PG, JW-300SP).

● JW-21CM, JW-22SU, JW-20FL5/20FLT, JW-22FL5/22FLT,

JW-255CM, JW-25TCM, JW-22CM, JW-21MN

- Using the option module No. switch (set value 0 to 7), determine option parameter. This setting occupies 2 K bytes per one module.
- As for the detail of the option paramter, see each user's manual of the option module (JW-21CM etc.).

Setting value of module No. switch	Address of option parameter(8)
0	00000 to 03777
1	00000 to 03777
2	00000 to 03777
3	00000 to 03777
4	00000 to 03777
5	00000 to 03777
6	00000 to 03777
7	00000 to 03777

Reference

- With the JW30H, set parameters of the JW-255CM, JW-25TCM, JW-22CM, and JW-21MN to each module. The option parameter area uses 64 bytes per module.

Chapter 4 System memory

The system memory is used to set each function of the JW300 and monitor error detail of the JW300.

4-1 List of system memories

The system memory has a capacity of 1.5K-byte which occupy an address area from #0000 to #2777. It has battery back-up. This section describes memory numbers that are released to use for users. Other memory numbers are reserved region, and data should not be written in.

System memory address(8)	Initial value(H)	Contents		See page	* JW30H
#0010	Calendar clock	Seconds	Monitor the clock (BCD)	4-5	○
#0011		Minutes			
#0012		Hours			
#0013		Date			
#0014		Month			
#0015		Year			
#0016		Day of week			
#0017	00	Control			
#0030	00	Lower byte	Monitor the minimum value of scan time (BCD)	4-6	
#0031		Upper byte			
#0032		Lower byte	Monitor the current value of every scan time (BCD)		
#0033		Upper byte			
#0034		Lower byte	Monitor the maximum value of scan time (BCD)		
#0035		Upper byte			
#0046	00	Monitor the rack and slot of an I/O error address (HEX)		4-7	
#0050	00	Lower byte	Monitor the program's error address (OCT)	4-8	
#0051		⋮			
#0052					
#0053		Upper byte			
#0054	00	User program source sum check code		—	
#0055					
#0056					
#0057					
#0060	00	User program source sum check error (Error = 01(H), Normal = 00(H))			
#0064	—	Monitor boot program version (sub version)			
#0065	—	Monitor boot program version			
#0066	—	Model code of boot program			
#0067					
#0150	00	Monitor switch set value of an option module in which an error occurred		4-9	
#0152	00	Monitor switch setting of a device net module in which an error occurred			

*○: Same function as the JW30H
△: Changed function
×: JW30H does not have this function

System memory address(8)	Initial value(H)	Contents	See page	* JW30H
#0160	00	Error codes resulting from self-diagnosis.	4-10	○
#0161				
#0162				
#0163				
#0164				
#0165				
#0166				
#0167				
#0170	00	Error codes of option module.		
#0171				
#0172				
#0173				
#0174				
#0175				
#0176				
#0177				
#0201	00	Set the resetting condition of TMR		
#0202		Set the resetting condition of CNT		
#0206	00	Set the operation continue/stop at fuse blown detection		
#0207	00	Set the operation continue/stop at option error	4-11	
#0211	00	Select whether to continue or stop operation when a device net module has an error.		
#0213	00	Set the top address number of comment memory using area		
#0222	02	Set the communication method used by the PG/COMM 2 port.		
#0223	00	Select the clock feature		
#0225	00	Set the 1ms timer function	4-12	×
#0226	00	Set the constant scan time		○
#0227	00	Set the 10 ms timer function		
#0230	C0	Lower byte Setting the range of a latched relay	4-13	
#0231	01	Upper byte (OCT)		
#0232	00	Lower byte Setting output hold address	4-14	
#0233		Upper byte (OCT)		
#0234	00	Set the PG/COMM1 port	4-15	
#0235	00			
#0236	00	Set the PG/COMM2 port		
#0237	00			
#0240	00	Set interrupt process	4-16	
#0241	FF			
#0242	00			
#0243	00			
#0244	00			×
#0245	00			
#0246	0A	Set extend for allowable voltage interruption time	4-18	○
#0247	00	Set the I/O addresses registration method		△
#0250	00	Lower byte Set the range of a latched relay	4-13	○
#0251	30	Upper byte (expansion area)		

System memory address(8)	Initial value(H)	Contents	See page	* JW30H
#0252	00	Lower byte Set output holding address	4-14	○
#0253	30	Upper byte (expansion area)		
#0255	00	Set the ROM operation mode	4-19	
#0256	80	Set the ROM storage area		△
#0266	00	Set EA-PG port.	4-20	×
#0267	00			
#0270	FF	Transfer register of system memory #0000 to #2777.		
#0271	FF			
#0272	00			
#0277	No fixed	BCC check code		△
#0300 ~ #0377	00 OCT	Set device net JW-20DN (Module No. 0).	4-22	○
#0410 ~ #0414	—	Specify symbol/comment use area. (When using the ladder logic programming software JW-300SP)	4-21	△
#0500 ~ #0577	00 OCT	Set device net JW-20DN (Module No. 1).	4-22	○
#1600	00	Error code	4-24	×
#1601	30	The number of errors that occurred — The number of errors that occurred for each error code: No. 1		
#1602	00	Error code		
#1603	80	The number of errors that occurred — The number of errors that occurred for each error code: No. 2		
#1604 ~ #1675	00 OCT	The number of errors that occurred for each error code: No. 3 ~ The number of errors that occurred for each error code: No. 31		
#1676	00	Error code		
#1677	00	The number of errors that occurred — The number of errors that occurred for each error code: No. 32		

System memory address(8)	Initial value(H)	Contents	See page	* JW30H
#2100 #2101 #2102 #2103	00	Top address of an area that stores logging data (File address)	4-25	×
#2104	00	Number of logging data storing		
#2105	00	Time stamp format		
#2106 #2107 #2110 #2111	FF	Register address of logging specified register 1 (File address)		
#2112 #2113 #2114 #2115	FF	Register address of logging specified register 2 (File address)		
#2116 〜 #2151	FF each	Register address of logging specified register 3 (File address) 〜 Register address of logging specified register 9 (File address)		
#2152 #2153 #2154 #2155	FF	Register address of logging specified register 10 (File address)		
#2156 〜 #2161	00 each	Storage counter of next logging data	4-26	
#2162 〜 #2165	00 each	The number of storages of logging data		
#2220 #2221 #2222 〜 #2226 #2230 〜 #2236	00 1F 00 each 00 each	Set PC card		
#2200	00	Set fault diagnosis (Enable = $01_{(H)}$, Disable = Other than $01_{(H)}$).	6-14	
#2300 〜 #2377	00 each	Set device net JW-20DN (module No. 2)	4-23	
#2400 〜 #2477	00 each	Set device net JW-20DN (module No. 3)		

4-2 Contents of system memory

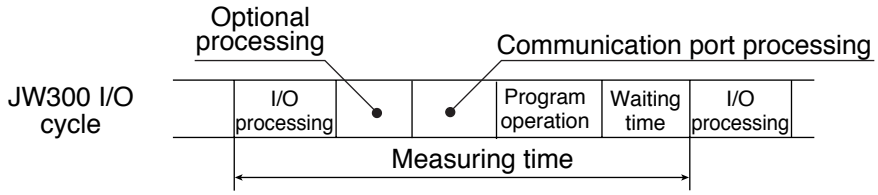
This section describes details of each item that is listed on the system memory table (page 4-1 to 4-4).

<table>
<tr><th>System memory No. (OCT)</th><th>Setting item</th><th>Contents</th></tr>
<tr><td>#0010</td><td rowspan="9">Clock feature</td><td>Seconds: 00 to 59 (BCD)</td></tr>
<tr><td>#0011</td><td>Minutes: 00 to 59 (BCD)</td></tr>
<tr><td>#0012</td><td>Hours: 00 to 23 (BCD)</td></tr>
<tr><td>#0013</td><td>Date: 01 to 31 (BCD)
The control module automatically distinguishes 30-day months from 31-day months and leap years.</td></tr>
<tr><td>#0014</td><td>Month: 01 to 12 (BCD)</td></tr>
<tr><td>#0015</td><td>Year: 00 to 99 (BCD)
Year is represented by the lower 2 digits of the Gregorian calendar year. Leap years are assumed to occur every 4 years.
(’96, ’00, and ’04 are identified as leap years.)</td></tr>
<tr><td>#0016</td><td>Day-of-week: 00 to 06 (BCD)
Set the day of the week when you adjust the present time. It changes from 00 to 06 each time date data is incremented. Day-of-week data is not computed from the year/month/date setting.
<table>
<tr><td>Day of week</td><td>SUN</td><td>MON</td><td>TUE</td><td>WED</td><td>THU</td><td>FRI</td><td>SAT</td></tr>
<tr><td>BCD value</td><td>00</td><td>01</td><td>02</td><td>03</td><td>04</td><td>05</td><td>06</td></tr>
</table></td></tr>
<tr><td>#0017</td><td>Control : Turn ON/OFF bits (D0, D3, and D7) for time adjustment/time monitor, 30 minutes correction, and stop/run clock.
<table>
<tr><th>Contents</th><th>D7</th><th>D6</th><th>D5</th><th>D4</th><th>D3</th><th>D2</th><th>D1</th><th>D0</th></tr>
<tr><td rowspan="2">ON</td><td>Time adjust</td><td colspan="3" rowspan="2">Not used.</td><td>30-sec. correction</td><td colspan="2" rowspan="2">Not used.</td><td>Stop clock</td></tr>
<tr><td>OFF</td><td>Time monitor</td><td>——</td><td>Start clock</td></tr>
</table>
- 30-sec. correction
If $010_{(8)}$ is set, 0 to 29 sec. is reset to 0 sec. ; 30 to 59 sec. is reset to 0 sec. with a carry to the minutes digit.
- Time adjustment
Set $001_{(8)}$ to stop the clock, then write the present time to #0010 to #0016. Next, write $200_{(8)}$ to set the time into the clock, whereupon bits D_0 and D_7 are reset and the clock starts working.
Note: By setting $001_{(8)}$, the display of system memory #0010 to #0016 and registers 099770 to 099776 is not updated (the clock is stopped apparently), but the internal clock is working.</td></tr>
</table>

- When the value of system memory location #0223 is $00_{(H)}$, time setting can be done from the JW300 program by using registers 099770 to 099777.
- The clock feature is backed up by battery.

#0030 **#0031**	Monitor scan time minimum value	Scan time minimum value is stored in a BCD value. [Example]: If the BCD value monitoring was 0020, the scan time minimum value is 20 ms. 00 20 Monitored with #0030 (Lower digit) Monitored with #0031 (Upper digit)
#0032 **#0033**	Monitor current value of each scan time	Scan time current value is stored in a BCD value. [Example]: If the BCD value monitoring was 0050, the current scan time is 50 ms. 00 50 Monitored with #0032 (Lower digit) Monitored with #0033 (Upper digit)
#0034 **#0035**	Monitor scan time maximum value	Scan time maximum value is stored in a BCD number. [Example]: If the BCD value monitoring was 0100, the scan time maximum value is 100 ms. 01 00 Monitored with #0034 (Lower digit) Monitored with #0035 (Upper digit)

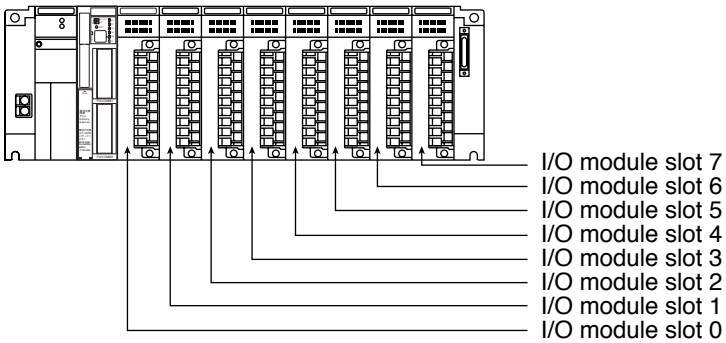
- The scan time measurement starts when the power is turned ON.
  When the operation is changed from run to stop (program mode), the latest scan time minimum and maximum values are stored. When the operation is changed from stop to run, the existing minimum and maximum value are cleared and newly detected values are stored.
  A scan error allowance is ±1 ms.
- Measurement of PC scan time is started at the beginning of I/O processing.



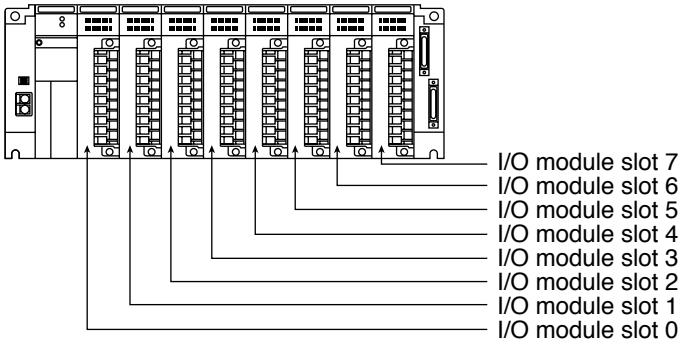
- The scan time is the total of the execution time from program block 0 to block N. Execution time of the waiting block is treated as 0.

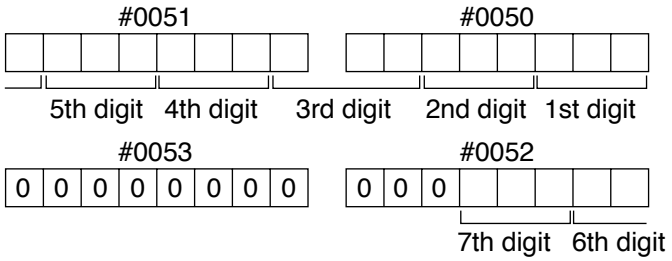
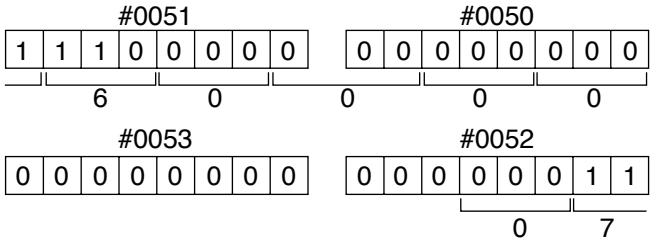
#0046	Monitor the I/O slot number where an error is detected	The rack and slot errors listed below can be monitored. The rack number and slot number where the error is detected are stored during PC I/O processing. As reference data in determining the error module. Error code (BCD) - I/O data bus error 44 - Output data check error 42 - Installed module error 40 - I/O rack panel error 48 - Table verify error 60 - Module No. switch verify error 61 - Table registration error 70 - Missing module error 71 - I/O point overflow error 72 - Module No. switch set error 73 ● **Bit arrangement of rack number, slot number** Bit → 6 5 4 3 2 1 0 #0046 | | 4 | 2 | 1 | 0 | 4 | 2 | 1 | Rack No. (0 to 7) Slot No. (0 to 7) - The rack number becomes 0 for the basic rack panel. For the expansion rack panel becomes setting value of the rack number switch (1 to 7). Therefore, when using the I/O bus expansion adapter, the rack number is decided by the combination of rack no. Switch of expansion rack panel (JW-34ZB/36ZB/38ZB) and I/O bus expansion adapter (JW-32EA). - For the slot number, the number of the I/O module slot in the basic/expansion rack panel (0 to 7) is stored.

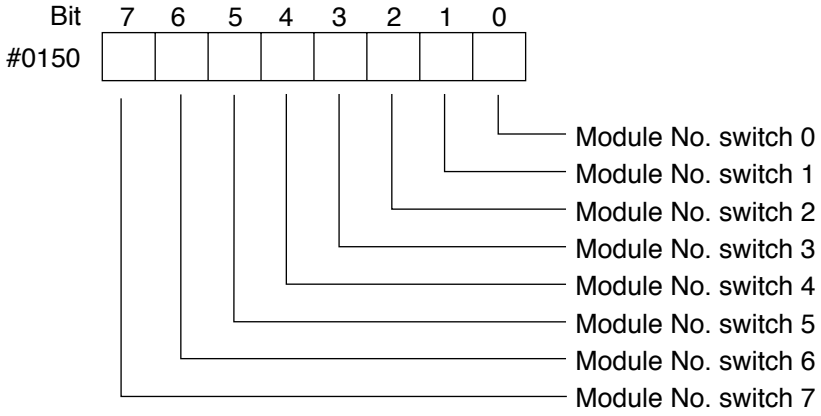
[Example] Basic rack panel: JW-318KB



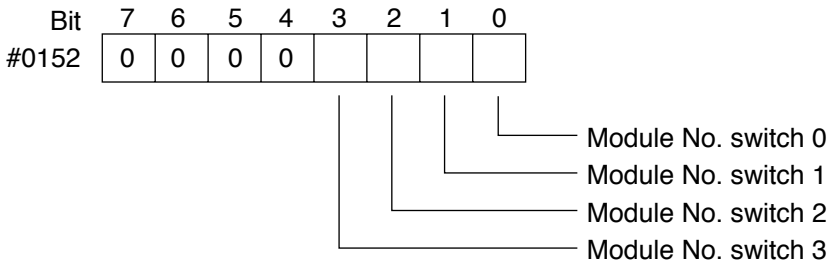
Expansion rack panel: JW-38ZB



#0050 **#0051** **#0052** **#0053**	Monitor user program error address	If there is the error code "24" (instruction code check error) stored in the system memory #0160 after diagnosis, you will know the address in error in the user program when the system memory #0050 to #0053 is monitored. The address in error is represented by a bit pattern (octal number), which will be as shown below. □ --- 0 (OFF), ■ --- 1 (ON)  [Example]: When address $0760000_{(8)}$ during programming by user is faulty 

<table>
<tr>
<td>#0150</td>
<td>Monitor switch set value in error option module</td>
<td>If there is the error code "53" (option error) stored in the system memory #0160 after diagnosis, you will know the error option module involving switch number when the system memory #0150 is monitored. The slot number in error is represented by a bit pattern, which will be as shown below.
The module No. switch numbers are expressed by bit pattern
□ --- Normal (0: OFF)
■ --- Abnormal (1: ON)
 

[Example] : When an option module, of which the module No. switch is set to 2, has an error
 

- A maximum of 8 option modules can be installed to a basic rack panel.
- The option module cannot be installed to an expansion rack panel.</td>
</tr>
<tr>
<td>#0152</td>
<td>Monitor switch set value in error device net and error I/O link module</td>
<td>If there is the error code "53" (option error) stored in the system memory #0160 after diagnosis, you will know the the module number switch set value of "error occurred device net masten module (JW-20DN) or error occurred I/O link master module (JW-23LMH) when the system memory #0152 is monitored.
The slot number in error is represented by a bit pattern, which will be as shown below.
□ --- Normal (0: OFF)
■ --- Abnormal (1: ON)
 

[Example] : When "JW-20DN or JW-23LMH," on which the module No. switch is set to 2, has an error.
 

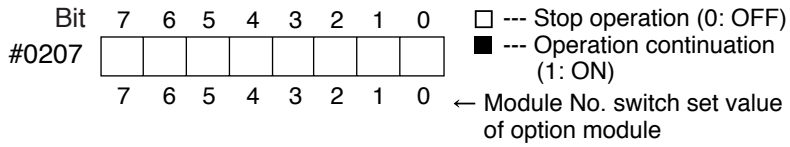
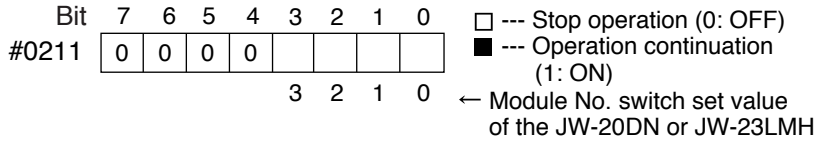
- A maximum of 4 JW-20DN and JW-23LMH can be installed to a basic rack panel.
- The JW-20DN and JW-23LMH cannot be installed to an expansion rack panel.</td>
</tr>
</table>

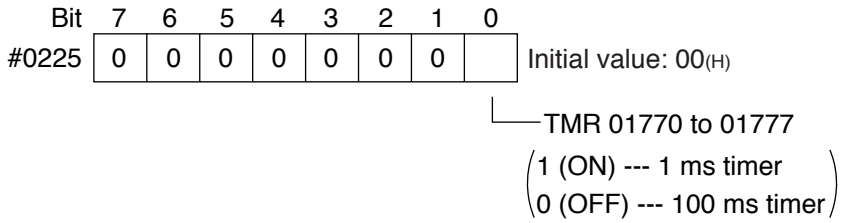
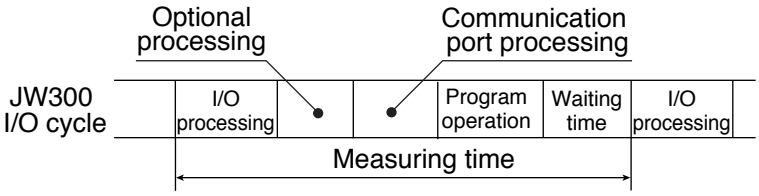
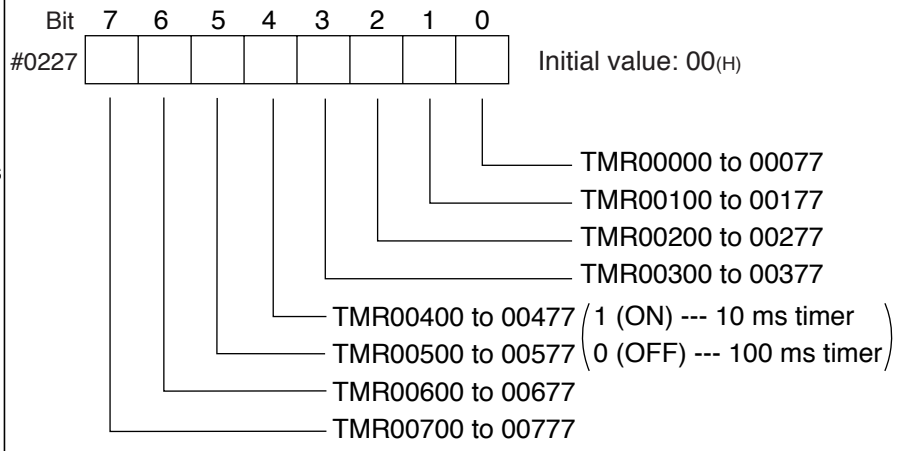
#0160 to #0167	Diagnostic error code	The error code will be stored when an error is encountered as a result of diagnosis. - #0160 to #0167 functions as a shift register which will be able to store 8 errors. For details, refer to "Self-diagnosis." - Error information including the time of error generation is stored in E7600 to E7777. - Error codes remain in system memory after the cause of the error is removed. To clear the error code, write "$00_{(H)}$" from support tool.

#0170 to #0177	Option module's error code	If an error is detected by self-diagnosis on option module, the corresponding error code is stored in these locations. - #0170 to #0177 function as a shift registers and can memorize occurrences of 8 errors. See each user's manual for the option module (JW-21CM etc.) for details on error code. - Information which includes the time of the error occurrence is stored in E5600 to E7577 for each setting of the module No. switch (0 to 7) of the option module. - All option module share system memory locations #0170 to #0177. - Error code remain in system memory after the cause of the error is removed. To clear the error code, write $00_{(H)}$ from support tool.

#0201	TMR reset condition	Used to program the state of the TMR instruction upon power recovery. $00_{(H)}$ --- Reset upon power recovery. (initial condition) $01_{(H)}$ --- Retains the state at power down. - The TMR instruction includes the DTMR (BCD), DTMR (BIN), UTMR (BCD), and UTMR (BIN) instructions.
#0202	CNT reset condition	Used to program the reset input condition for CNT instruction, application instruction (as follow). $00_{(H)}$ --- Reset when ON. (initial condition) $01_{(H)}$ --- Retains when OFF. - The CNT instruction includes the DCNT (BCD), DCNT (BIN), UCNT (BCD), and UCNT (BIN) instructions. - The following application instructions are available : F-60 (F/B SFR) F-160 (NSFR) F-60w (F/B SFR) Fc160 (NSFR) F-60d (F/B SFR) F-261 (RCNT) F-62 (U/DC) Fc261 (RCNT) F-62w (U/DC) F-62d (U/DC)

#0206	Setting the operation continuation/ halt at fuse blown detection	To set to continue or stop operation of the JW300 when fuse blow (failure code 49) is detected in the output module (JW-262S). $00_{(H)}$ --- Operation continuation (initial setting) $08_{(H)}$ --- Stop operation

#0207	Set the operation continuation/ stop operation at option module error	To set to continue or stop operation of the JW300 when module error (failure code 53) is detected in the option module. When the bit of the module No. switch (0 to 7) of option module is turned ON, the operation continues.  Initial state is stop state(□) in $00_{(H)}$ for all bit.
#0211	Set the operation continuation/ stop operation at device net, I/O link module error	To set to continue or stop operation of the JW300 when module error (failure code 53) is detected in the device net master module (JW-20DN), I/O link master module (JW-23LMH). When the bit of the module No. switch set value of the JW-20DN or JW-23LMH is turned ON, the operation continues.  Initial state is stop state(□) in $00_{(H)}$ for all bit.
#0213	Select error storage area	Select register area to store error codes. $00_{(H)}$ --- Not used $02_{(H)}$ --- Register E5600 to $E7777_{(8)}$ (Initial setting)
#0222	Set the communi-cation method for PG/COMM2	Sets the communication method used by the PG/COMM 2 port. **Set value** / **Communication method** $00_{(H)}$: RS-232C (RTS signal is ON while power is present) RS-422A (only 4-line systems at 1:1 allowed) $02_{(H)}$: RS-232C connection only (RTS signal is OFF while sending data, and ON while not sending data) - Use this setting if you want to control the data flow using the RTS signal. Note: The support tool cannot be used. $04_{(H)}$: RS-422A connection only (4-line system with 1: N connections is possible)
#0223	Selects clock feature	This byte allows to control the clock feature on register. It enables time adjustment from the PC program. When "Use register" is selected, you can adjust time using the JW300's program. $00_{(H)}$ --- Used register (initial state) $01_{(H)}$ --- Not used Registers 099770 to 099777 are used for clock control.

#0225	Select the 1 ms timer function	Each area of TMR 01770 to 01777 can be set in 1 ms/100 ms timer by bit ON/OFF switching.  DTMR or UTMR are only used for 100 ms timer.
#0226	Constant scan	JW300's scan time can set at your disposal. Setting is 01 to 99 ms in the BCD value. #0226 set value / JW300 scan time 00 (BCD) / Minimum scan 01 to 99 (BCD) / Scan time of 01 to 99 ms Initial value is 00 (BCD). - If the program operation takes a long time than setting of scan time, the JW300 scan time is determined by the program operation time. - The JW300 scan time measures the time simultaneously with start of input and output processing.  Setting time n accuracy=Setting value $^{-0}_{+1}$ ms.
#0227	Select the 10 ms timer function	Each area of TMR 00000 to 00777 can be set in 10 ms/100 ms timer by bit ON/OFF.  - DTMR or UTMR are only used for 100 ms timer.

Used to increase/decrease the latched relay area from the initial state (below).

Set with 8 points as unit. Set numeric values by setting file address with octal.

	Setting area (word, octal)	Byte address	Relay No.
#0230, #0231	00000000 to 00001577	⊐00000 to ⊐01577	000000 to 015777
#0250, #0251	00030000 to00035577	⊐02000 to ⊐07577	020000 to 075777

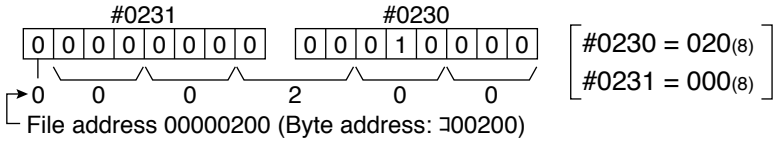
#0230
#0231

Set latched relay area

The latched relay area occupies from the assigned byte address to the final addresses (⊐01577, ⊐07577).

⊐10000 to ⊐54377 (relay number 100000 to 543777) are always latched relay area.

[Example] To set ⊐00200 and after (relay number 002000 to 015777) to the latched relay.



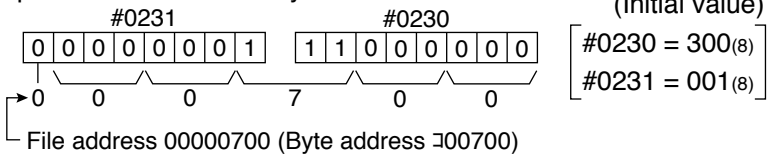
#0250
#0251

Set latched relay area (expansion relay area)

■ **Initial value**

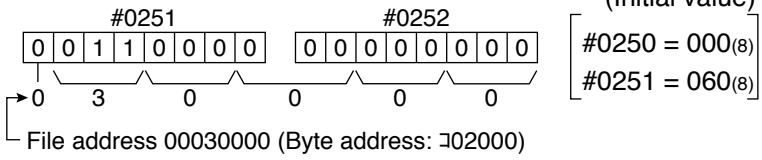
● Initial value of #0230, #0231

Area from ⊐00700 and after (relay No. 007000 to 015777) are specified as latched relay.



● Initial value of #0250, #0251

Area from ⊐02000 and after (relay No. 020000 to 075777, 100000 to 543777) are specified as latched relay.



#0232
#0233

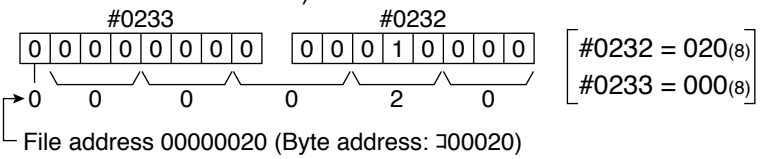
Set output retention address

Used to set the top address for the output relay which employed to retain the output module's output when the control module stops. Set with 8 points as unit. Set numeric values by setting file address with octal.

	Setting area (word, octal)	Byte address	Relay No.
#0232, #0233	00000000 to 00001577	⊐00000 to ⊐01577	000000 to 015777
#0252, #0253	00030000 to 00035577	⊐02000 to ⊐07577	020000 to 075777

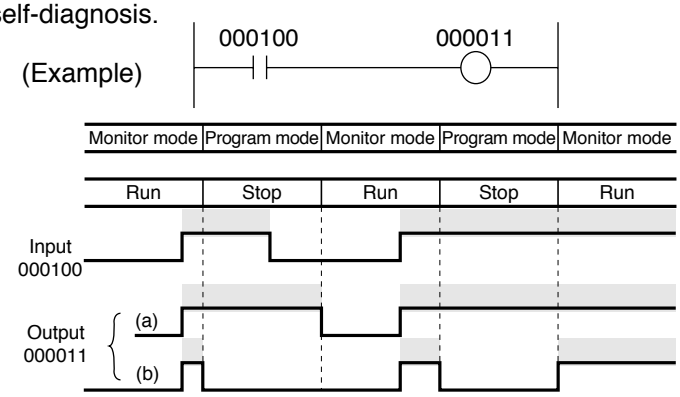
The output hold area occupies from the assigned byte address to the final addresses (⊐ 01577, ⊐ 07577).
Addresses ⊐10000 to ⊐54377 (relay No. 100000 to 543777) are always output hold area.

[Example] To assign it to the output hold following ⊐00020 (Relay No. 000200 to 015777)



File address 00000020 (Byte address: ⊐00020)

- The output turns OFF for the area the output retention is canceled as the latch of the output module is reset when the JW300 is switched to the program mode or the JW300 stopped its operation as a result of self-diagnosis.



(a) --- When 000011 is within the output retention area.
(b) --- When 000011 is within the output retention canceled area.

- When the JW300 stops because of an error detected as a result of self-diagnosis, it may not turn OFF the output module within the area that the output retention is canceled, depending on the nature of the error. Output that needs to turn OFF when the JW300 meets the error should be connected serial to the power supply module halt output.

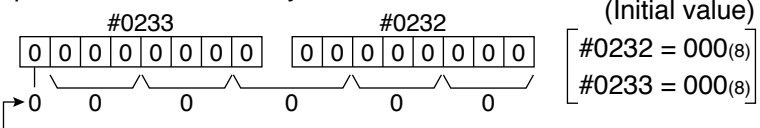
#0252
#0253

Set output latch address (expansion relay area)

■ **Initial value**

● **Initial value of #0232, #0233**

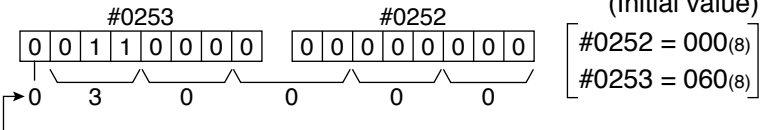
Area from ⊐00000 and after (relay No. 000000 to 015777) are specified as latched relay.



File address 00000000 (Byte address ⊐00000)

● **Initial value of #0252, #0253**

Area from ⊐02000 and after (relay No. 020000 to 075777) are specified as latched relay.

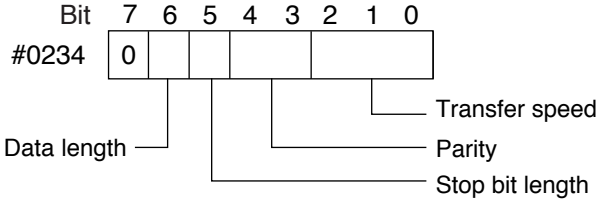


File address 00030000 (Byte address: ⊐02000)

#0234
#0235

Set PG/COMM1 port

- Specify communication condition of the PG/COMM1 port (communication port 1) to bits (D_0 to D_6) of #0234.



D_6	Data
0	7 bits
1	8 bits

D_5	Stop
0	1 bit
1	2 bits

D_4	D_3	Parity
0	0	None
0	1	Odd
1	0	Even
1	1	Disable

D_2	D_1	D_0	Transfer speed
0	0	0	9600 bits/s
0	0	1	19200
0	1	0	38400
0	1	1	76800
1	0	0	115200
1	0	1	230400

- Communication port requires to set station number (001 to $037_{(8)}$) to #0235 since the contents of communication are the same as those of the link module (JW-21CM) placed in command mode.

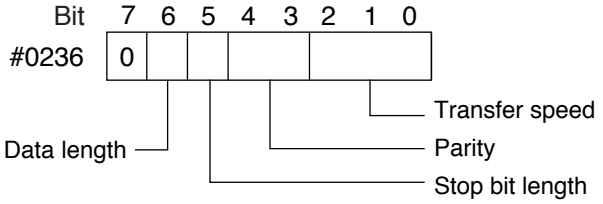
#0235 | Station number |

- The initail values both #0234 and #0235 are $00_{(H)}$.

#0236
#0237

Set PG/COMM2 port

- Specify communication condition of the PG/COMM2 port (communication port 2) to bits (D_0 to D_6) of #0236.



D_6	Data
0	7 bits
1	8 bits

D_5	Stop
0	1 bit
1	2 bits

D_4	D_3	Parity
0	0	None
0	1	Odd
1	0	Even
1	1	Disable

D_2	D_1	D_0	Transfer speed
0	0	0	9600 bits/s
0	0	1	19200
0	1	0	38400
0	1	1	76800
1	0	0	115200
1	0	1	230400

- Communication port requires to set station number (001 to $037_{(8)}$) to #0237 since the contents of communication are the same as those of the link module (JW-21CM) placed in command mode.

#0237 | Station number |

- The initial values of both #0236 and #0237 are $00_{(H)}$.

Reference (Set PG/COMM port).

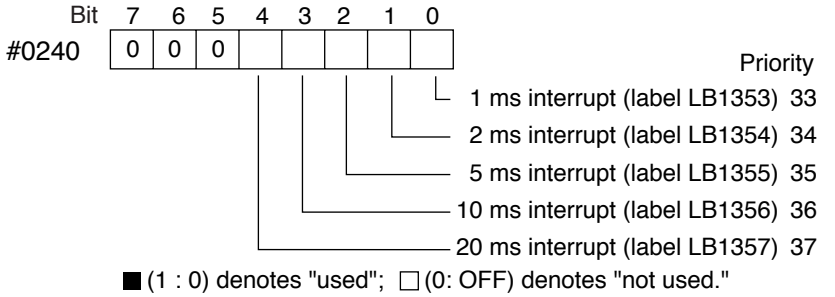
- With the JW30H, data length is "fix to 7 bits," and data transfer speed (bps) shall be select from "1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200".

#0240 〜 #0245

Set interrupt processing
- #0240 Specify TMR interruption.
- #0241 Specify input rack/slot.
- #0242 to #0245 Specify input interrupt conditions.

These bytes are used to define the JW300 interrupt service conditions. A total of 37 interrupts are available including timer and input interrupts. Interrupts can be serviced even when the JW300 is busy with internal operation or I/O processing. Once an interrupt is received, the PC calls the subroutine with the specified label.

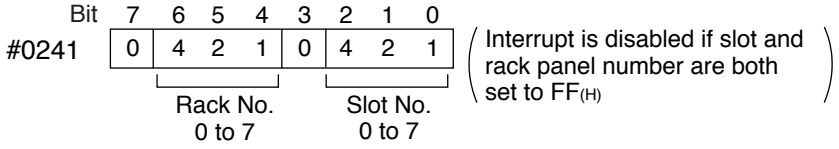
● Each bit of #0240 has a different timer interrupt time. Label numbers represent subroutine labels to which control branches on interrupt. All 5 timer interrupts are available.

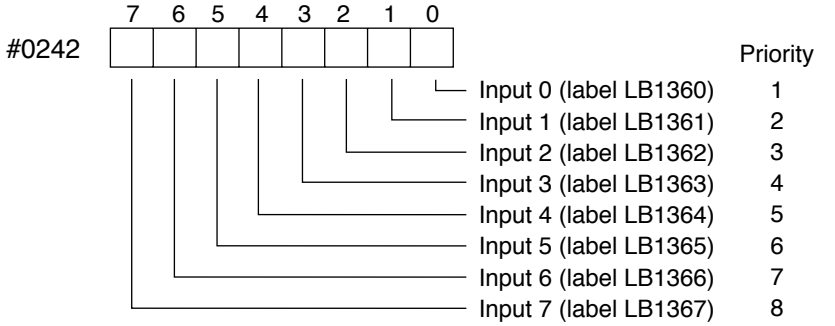


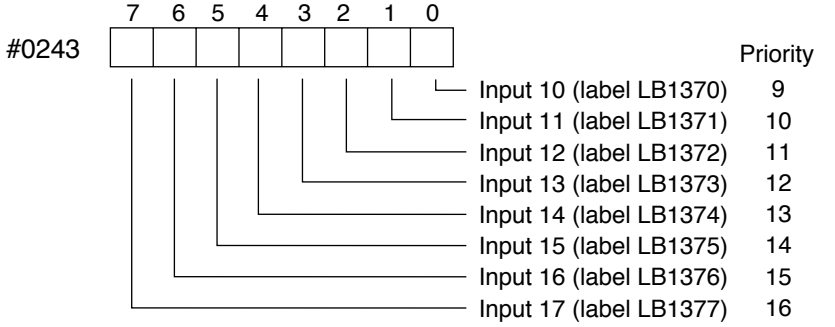
■ (1 : 0) denotes "used"; □ (0: OFF) denotes "not used."

● Locations #0241 to #0245 are used to specify input interrupt service conditions.
The JW300 monitors the input module specified by its slot and rack panel number at 1ms interval, detects an interrupt request at the rising (OFF to ON) or falling (ON to OFF) edge of an input signal, and then calls the subroutine with the specified label (F-140).

- Locations #0241 holds the slot and rack panel numbers of the input module which is the source of the interrupt. When the input module is 8/16-point module, input 20 to 37 that are set at #0244 and #0245 are invalid. In addition, when a 64-point input module is used, these addresses cannot be used for interrupt input.
- Locations #0242 to #0245 specify whether the rising or falling edge of the 32 input signals specified by #0241 is to be used to call the interrupt service routine.
- Each specified bit at #0242 to #0245 correspond to 32 points of the input module. Each bit is also assigned a unique subroutine label.





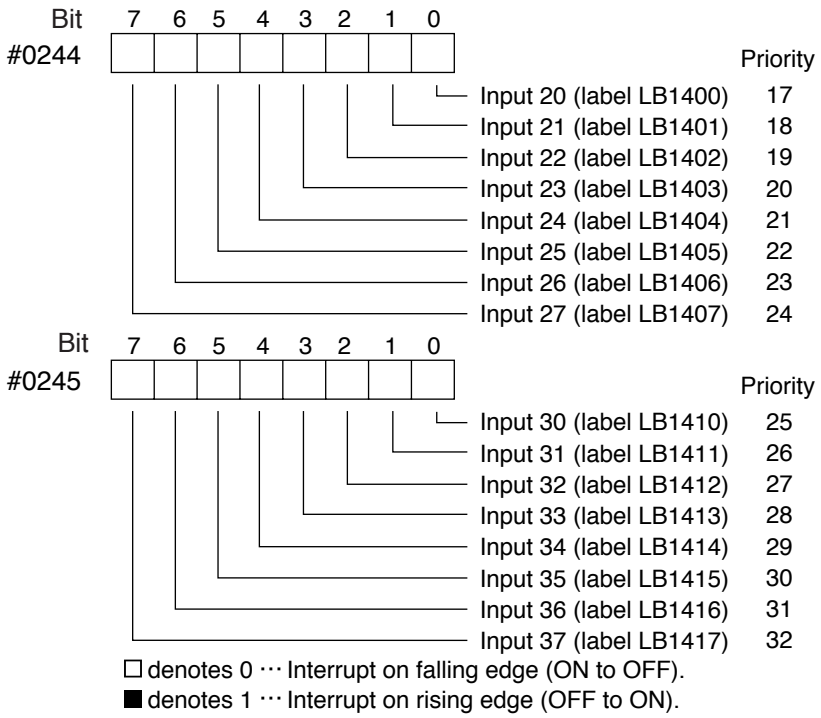


□ denotes 0 --- Interrupt on falling edge (ON to OFF).
■ denotes 1 --- Interrupt on rising edge (OFF to ON).

- The initial value of bytes #0240, #0242, #0243 is $00_{(H)}$, the initial value of bytes #0241 is $FF_{(H)}$.

#0240
〉
#0245

Set interrupt processing



□ denotes 0 ··· Interrupt on falling edge (ON to OFF).
■ denotes 1 ··· Interrupt on rising edge (OFF to ON).

- The initial value of bytes #0244, #0245, is $00_{(H)}$.

Remarks

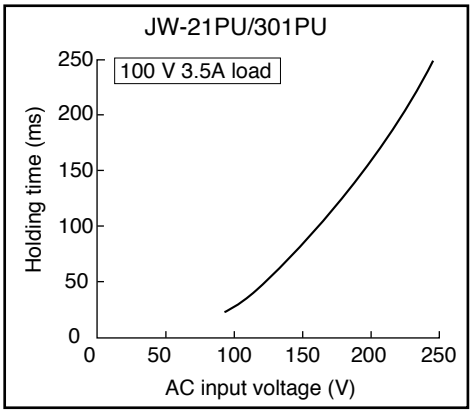
- Interrupt requests are treated in much the same way as the F-142 (CALL) instruction and F-148 (CAL+) instruction. Notes for interrupts are also the same as for those instructions.
- Locate the subroutine programs by placing the subroutine labels following the F-40 (END) instruction in the main program.
- Set an interrupt program to operate within 1ms. If longer than 1ms, interruption may be ignored.
- Interrupts are effective if an output or special I/O module is installed in the slot and rack panel number where input interrupts are available. Pay attention to the correct module installation.
- Interrupts on an input module installed in the remote I/O slave module (JW-21RS) is disabled.
- Timer and input interrupts are ignored if no jump destination label is specified (by F-140).
- If more than one interrupt has simultaneously occurred, the one with the highest priority (with smaller number) is serviced first.
- The controller will handle interrupts that occur while exchanging data with an I/O module as follows:

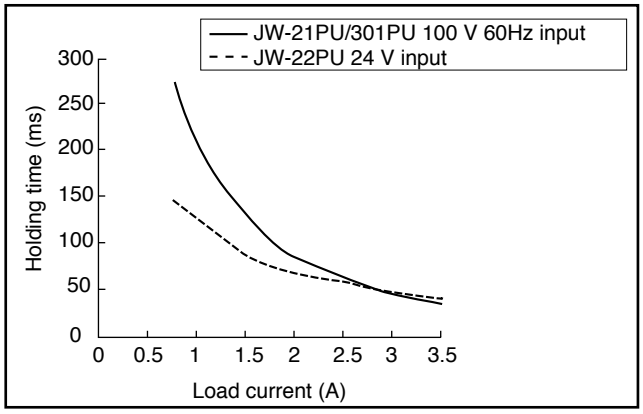
I/O module Special I/O module	Executes the interrupt when data exchange is complete.
Option module	Executes the interrupt during data exchange.

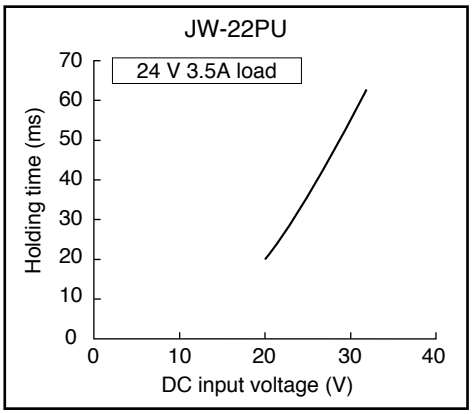
- The break function is invalid while interruption is being set.
- The subroutine call instruction (F-142 [CALL], F-148 [CAL+1]) cannot be used within an interrupt handling program.

#0246	Setting the allowable voltage interruption time length	Set when varying the momentary power failure detection time from 0 to 255 ms. Setting value is decimal. Initial setting is $010_{(D)}$ (10ms).

- When varying the setting of system memory #0246 (setting of extension of momentary power failure detection time), the variable range is determined by the power source input voltage and load current. Extend the momentary power failure detection time by referring to the following graphs of holding time characteristics and output holding time characteristics.
- During momentary power failure, all actions of control module are stopped, and the operation continues after power recovery. If longer than the momentary power failure detection time setting (power failure), after recovery, the operation must be started by turning on the power again.
- Be careful sufficiently in consideration of the capacity of 5V in the basic rack panel and expansion rack panel.







#0247	Set the I/O address	Setting registration method of I/O address. $00_{(H)}$ ----- Auto I/O registration at power ON. - Relay numbers are assigned continuously from $00000_{(8)}$, from the basic rack panel to the expansion rack panel (rack panel number in order). $03_{(H)}$ ----- Auto I/O registration at power ON is prohibited. - Write $03_{(H)}$ after automatic I/O registration, and operate by prohibiting automatic registration. $04_{(H)}$ ----- Available for setting rack top address. - Specify top address to each rack (1 to 7) using the support tool (JW-300SP). $C0_{(H)}$ ----- Settable slot top address - Specify the top address to each slot (0 to 7) of each lack (1 to 7) using the support tool (JW-300SP). **Reference** With the JW30H, setting of $C0_{(H)}$ is not available.

#0255 Set ROM operation mode

Setting ROM operation mode.

Setting value		Transfer ROM to RAM (at power supply ON)	Data Memory after transfer power supply ON	Mode after transfer power supply ON	Transfer ROM to RAM by tool
OCT	HEX				
000	00	No	—	—	Available
021	11	Yes	Hold	Mode at power supply OFF	Available
042	22	Yes	Clear *1	Stop	Available
104	44	Yes	Clear *1	Run	Available

* Stored data within ROM is holding.

The initial value is $00_{(H)}$.
See the section on "ROM operation" in the JW300 user's manual hardware version for details.

#0256 Set contents into ROM

Set the storing contents into a ROM to be used for operation using the ROM.

Set value		Resisterable contents to ROM [○: Yes, × : No]			
OCT	HEX	System memory [#200 to #2777]	Program *2	Register *3	File register
200	80	○	○	×	×
201	81	○	○	○	×
202	82	○	○	×	○ *4
203	83	○	×	○	×
204	84	○	×	×	○ *5
205	85	○	○	○	○ *4
206	86	○	×	○	○ *5

Initial value is $80_{(H)}$.

*2 ROM area of program

*3 ROM area of register

JW-311CU/312CU	8 K words	009000 to 099777
JW-321CU/322CU	16 K words	109000 to 199777
JW-331CU/332CU	32 K words	209000 to 299777
JW-341CU/342CU	64 K words	309000 to 389777
JW-352CU	128 K words	- E0000 to E7777 cannot be stored as ROM.
JW-362CU	256 K words	

*4 ROM area of file register

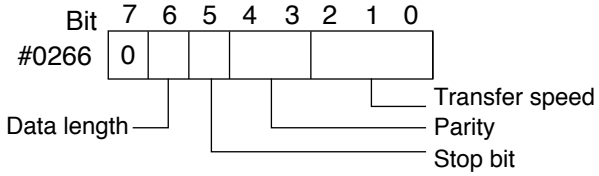
*5 ROM area of file register

JW-311CU/312CU	None	None
JW-321CU/322CU	00000000 to $00077777_{(8)}$ (32 K bytes)	00000000 to $00077777_{(8)}$ (32 K bytes)
JW-331CU/332CU	00000000 to $00077777_{(8)}$ (32 K bytes)	00000000 to $00377777_{(8)}$ (128 K bytes)
JW-341CU/342CU	00000000 to $00777777_{(8)}$ (256 K bytes)	00000000 to $01777777_{(8)}$ (512 K bytes)
JW-352CU	00000000 to $00777777_{(8)}$ (256 K bytes)	00000000 to $01777777_{(8)}$ (512 K bytes)
JW-362CU	00000000 to $00777777_{(8)}$ (256 K bytes)	00000000 to $03777777_{(8)}$ (1 M bytes)

#0266
#0267

Set EA-PG port

- Place communication conditions of the EA-PG port (communication port 3) to bits (D_0 to D_6) of #0266.



D_6	Data
0	7 bits
1	8 bits

D_5	Stop
0	1 bit
1	2 bits

D_4	D_3	Parity
0	0	None
0	1	Odd
1	0	Even
1	1	Disable

D_2	D_1	D_0	Transfer speed (bps)
0	0	0	9600
0	0	1	19200
0	1	0	38400
0	1	1	76800
1	0	0	115200

- Communication port requires the station number 001 to $037_{(8)}$ since the contents of communication are the same as those of the link module (JW-21CM) placed in command mode.

#0267 | Station number |

- The initial values both #0266 and #0267 are $00_{(H)}$.

#0270
#0271
#0272

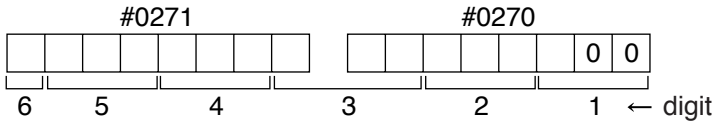
Transfer system memory #0000 to #2777 to register

When to transfer system memory #0000 to #2777 to a specified register (except file registers), place the setting meaning to "transfer" to #0272, and top address to destination to #0270 to #0271 with file address.
In addition, enter a "number multiplied by 4" (such as 009000, 009004) for the specified register. If other value than "number multiplied by 4" is entered, bits D_0 and D_1 of #0270 are set to 0. For example, if 009001 (file address $004001_{(8)}$) is entered, it will be changed to 009000.

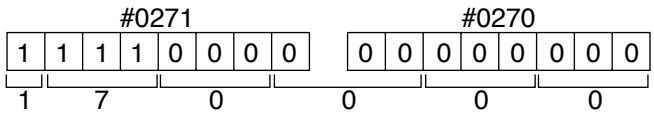
● **#0272**

Set value(H)	Content
01	Transfer
Except 01	Does not transfer

● **#0270 to #0271**



[Example] File address $00170000_{(8)}$ is top address

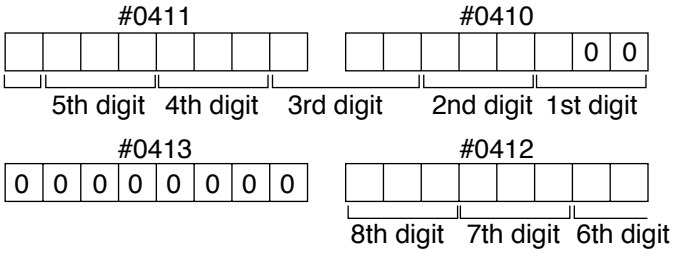
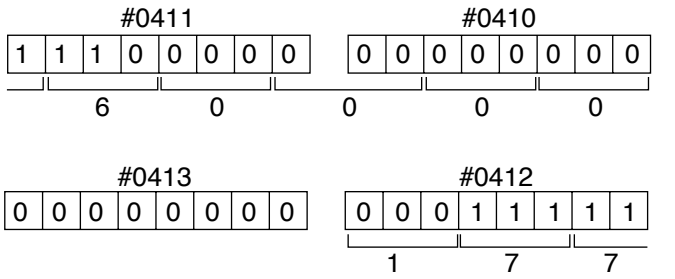
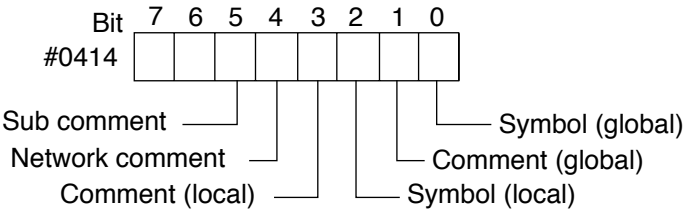


Initial value is #0270, #0271 = $FF_{(H)}$, #0272 = $00_{(H)}$

#0277

BCC check

The JW300 automatically computes and registers BCC check code for the contents of system memory actions #0200 to #0276.

#0410 to #0414	Specify use area (type) of symbol/comment	Specify top addresses and type of symbol/comment for relay, timer/counter, register, file register, and F-90 (REM) instruction that are written to the JW300 (read from the JW300) using the ladder logic programming software JW-300SP. ● **Top address (#0410 to #0413)** Specify file address of the file register to #0410 to #0413.  [Ex.] When file register $17560000_{(8)}$ (file address $17760000_{(8)}$) is top address.  ● **Type of symbol/comment (#0414)** Turn ON (1) the bit (0 to 5) of #0414 and select types. 

■ **Set device net**

#0300 to #0377	Set device net module (Module No. 0)	Set the following when the device net master module (JW-20DN) is used with its module No. switch set to "0." ● **When it is in master mode** - #0300 to #0303: Top address of the I/O table (*) - #0304 to #0307: Top address of the diagnosis table (*) - #0310 to #0313: Top address of the explicit message table (*) - #0314 to #0317: Top address of the scan list table (*) - #0321: Data length when editing the scan list - #0324, #0325: ISD (communication monitor time) - #0326, #0327: EPR (communication monitor time) - #0330: Slave output status when the JW300 is stopped operation ● **When it is in slave mode** - #0360 to #0363: Top address of the I/O table (*) - #0364 to #0367: The number of I/O bytes - #0370: Latch/clear the slave area when a communication error occurs. - #0371, #0372: Response time to the master module
#0500 to #0577	Set device net module (Module No. 1)	Set the following when the device net master module (JW-20DN) is used with its module No. switch set to "1." ● **When it is in master mode** - #0500 to #0503: Top address of the I/O table (*) - #0504 to #0507: Top address of the diagnosis table (*) - #0510 to #0513: Top address of the explicit message table (*) - #0514 to #0517: Top address of the scan list table (*) - #0521: Data length when editing the scan list - #0524, #0525: ISD (communication monitor time) - #0526, #0527: EPR (communication monitor time) - #0530: Slave output status when the JW300 is stopped operation ● **When it is in slave mode** - #0560 to #0563: Top address of the I/O table (*) - #0564 to #0567: The number of I/O bytes - #0570: Latch/clear the slave area when a communication error occurs. - #0571, #0572: Response time to the master module

* When the free allocation is selected.

- For the details of the #0300 to #0377 and #0500 to #0577, see the JW-20DN user's manual.
  For your information, #0320, #0331 to #0377, and #0520, #0531 to #0577 are not used areas.

#2300 to #2377	Set device net module (Module No. 2)	Set the following when the device net master module (JW-20DN) is used with its module No. switch set to "2." ● **When it is in master mode** - #2300 to #2303: Top address of the I/O table (*) - #2304 to #2307: Top address of the diagnosis table (*) - #2310 to #2313: Top address of the explicit message table (*) - #2314 to #2317: Top address of the scan list table (*) - #2321: Data length when editing the scan list - #2324, #2325: ISD (communication monitor time) - #2326, #2327: EPR (communication monitor time) - #2330: Slave output status when the JW300 is stopped operation ● **When it is in slave mode** - #2360 to #2363: Top address of the I/O table (*) - #2364 to #2367: The number of I/O bytes - #2370: Latch/clear the slave area when a communication error occurs. - #2371, #2372: Response time to the master module
#2400 to #2477	Set device net module (Module No. 3)	Set the following when the device net master module (JW-20DN) is used with its module No. switch set to "3." ● **When it is in master mode** - #2400 to #2403: Top address of the I/O table (*) - #2404 to #2407: Top address of the diagnosis table (*) - #2410 to #2413: Top address of the explicit message table (*) - #2414 to #2417: Top address of the scan list table (*) - #2421: Data length when editing the scan list - #2424, #2425: ISD (communication monitor time) - #2426, #2427: EPR (communication monitor time) - #2430: Slave output status when the JW300 is stopped operation ● **When it is in slave mode** - #2460 to #2463: Top address of the I/O table (*) - #2464 to #2467: The number of I/O bytes - #2470: Latch/clear the slave area when a communication error occurs. - #2471, #2472: Response time to the master module

* When the free allocation is selected.

- For the details of the #2300 to #2377 and #2400 to #2477, see the JW-20DN user's manual. For your information, #2320, #2331 to #2377, and #2420, #2431 to #2477 are not used areas.

■ **Number of error occurrence for each error code**
The number of errors occurred is stored for each error code (max. 32 types).

#1600, #1601	The number of errors that occurred for each error code: No. 1	#1600: Error code(H) #1601: The number of errors that occurred (000 to 377(8))
#1602, #1603	The number of errors that occurred for each error code: No. 2	
#1604, #1605	,, No. 3	
#1606, #1607	,, No. 4	
#1610, #1611	,, No. 5	
#1612, #1613	,, No. 6	
#1614, #1615	,, No. 7	
#1616, #1617	,, No. 8	
#1620, #1621	,, No. 9	
#1622, #1623	,, No. 10	
#1624, #1625	,, No. 11	
#1626, #1627	,, No. 12	
#1630, #1631	,, No. 13	
#1632, #1633	,, No. 14	
#1634, #1635	,, No. 15	
#1636, #1637	,, No. 16	
#1640, #1641	,, No. 17	
#1642, #1643	,, No. 18	
#1644, #1645	,, No. 19	
#1646, #1647	,, No. 20	
#1650, #1651	,, No. 21	
#1652, #1653	,, No. 22	
#1654, #1655	,, No. 23	
#1656, #1657	,, No. 24	
#1660, #1661	,, No. 25	
#1662, #1663	,, No. 26	
#1664 #1665	,, No. 27	
#1666, #1667	,, No. 28	
#1670, #1671	,, No. 29	
#1672, #1673	,, No. 30	
#1674, #1675	,, No. 31	
#1676, #1677	,, No. 32	

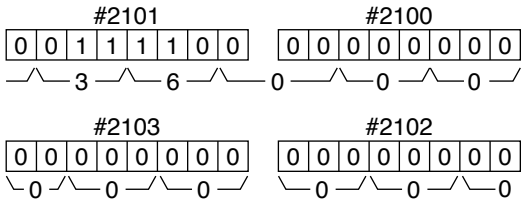
Address	Content
n	Error code(H)
n + 1	The number of errors that occurred (000 to 377(8))

- No.1 to No. 32 are stored error codes in order of occurrence of errors.

■ Set logging

Set #2100 to #2165 for using the logging function (F-403 instruction). => See page 6-7.

<table>
<tr><td>#2100 to #2103</td><td>Top address of logging data storage area</td><td>Specify top address to store logging data into file address #2100 to #2103. These also can be set to extension memory (file address 20000000000(8) and up) using a SRAM card. => See page 6-19.

[Ex.] When setting register 109000
 

- File address of register 109000 is 00036000(8).</td></tr>
<tr><td>#2104</td><td>Number of storages of logging</td><td>Specify the number of storages (set value x 10) of logging data.
[Ex.] when 050(D) is specified as the set value, the number of storages is 50 x 10 = 500 times.</td></tr>
<tr><td>#2105</td><td>Time stamp format</td><td>Specify format of time stamp that is added to logging data.
<table><tr><th>Set value(H)</th><th>Format</th></tr><tr><td>00</td><td>None</td></tr><tr><td>01</td><td>Year, month, day, hour, minute, second (add 6 bytes to logging data)</td></tr><tr><td>02</td><td>Hour, minute, and second (add 3 bytes to logging data)</td></tr></table></td></tr>
<tr><td>#2106 to #2111</td><td>Logging specifying register 1</td><td>Set register address to execute logging using file address. The setting method is the same as for the #2100 to #2103.</td></tr>
<tr><td>#2112 to #2115</td><td>Logging specifying register 2</td><td rowspan="9">Set logging specification register 2 to 10 with the same method as for logging specification register 1 (#2106 to #2111).</td></tr>
<tr><td>#2116 to #2121</td><td>Logging specifying register 3</td></tr>
<tr><td>#2122 to #2125</td><td>Logging specifying register 4</td></tr>
<tr><td>#2126 to #2131</td><td>Logging specifying register 5</td></tr>
<tr><td>#2132 to #2135</td><td>Logging specifying register 6</td></tr>
<tr><td>#2136 to #2141</td><td>Logging specifying register 7</td></tr>
<tr><td>#2142 to #2145</td><td>Logging specifying register 8</td></tr>
<tr><td>#2146 to #2151</td><td>Logging specifying register 9</td></tr>
<tr><td>#2152 to #2155</td><td>Logging specifying register 10</td></tr>
</table>

#2156 to #2161	These function as a counter to store the next logging data. 0 to [the number of storages of logging (#2104) - 1]
#2162 to #2165	The number of storage times of logging data 0 to FFFFFFF(H)

■ Setting of PC card

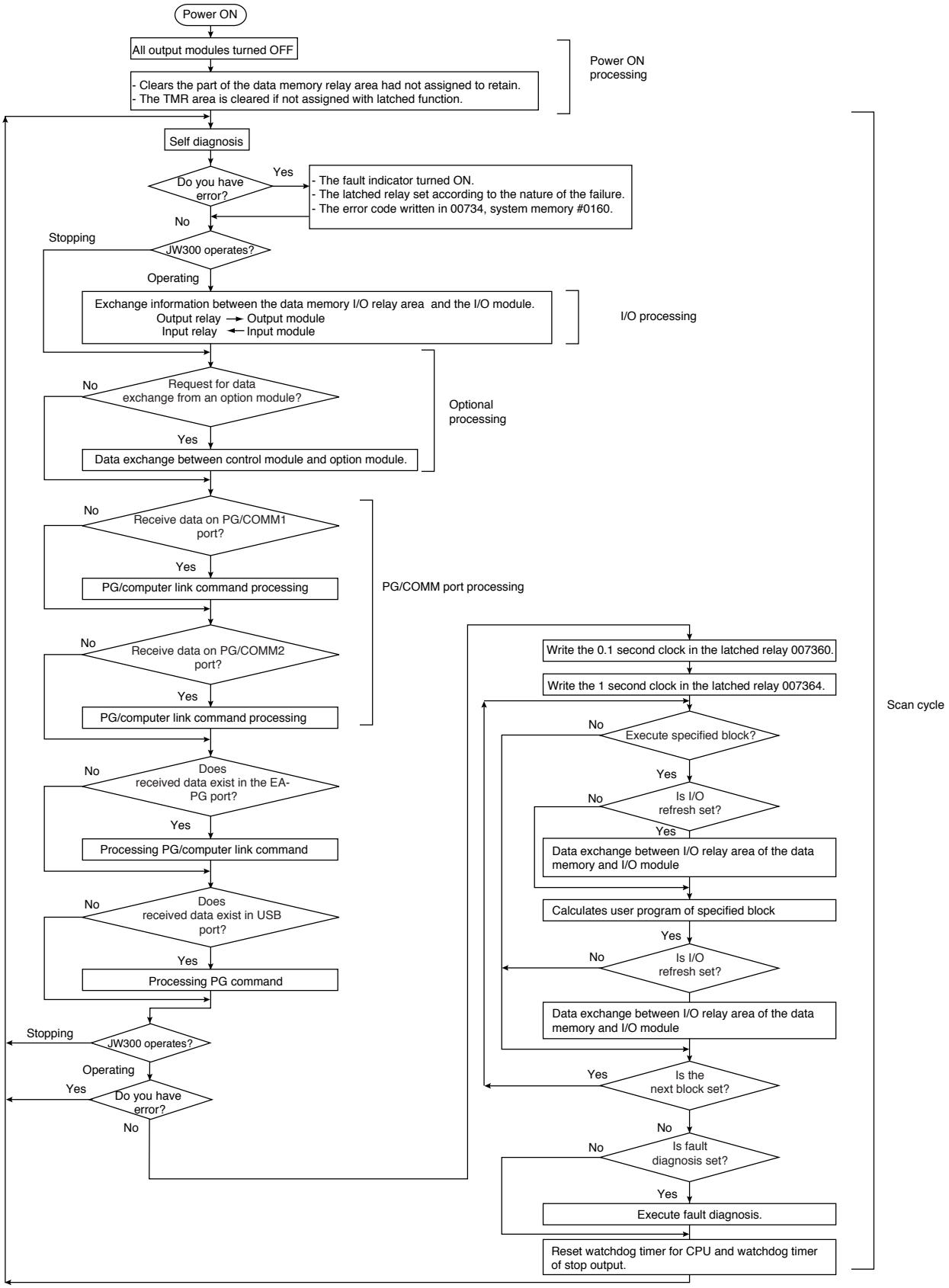
If the JW-3*2CU is used as a control module and you will use a PC card, set #2220 to #2226 and #2230 to #2236.

#2220	Writing to a PC card is prohibited.
#2221	Select file type to store in a PC card.
#2222	Set data transfer between a PC card and the JW-3*2CU.
#2223 to #2225	Automatically save data to a PC card for each time of a specified interval.
#2226	When an error occurs for reading/writing to a PC card, the respective error code is stored.
#2230 to #2236	If the AUTO LD switch on the JW-3*2CU is OFF, specify a file name to save on a PC card when it is inserted.

Chapter 5 Operation of control module

5-1 Operation cycle

[1] Operational flow chart



- The F-80 instruction allows I/O operations when the PC is busy for internal processing.
- Program can be executed using interrupts service (#0240 to #0245) even when the PC is busy for I/O, option, or PG/COMM port processing.

5-2 Self-diagnosis

The JW30H self-diagnosis each module such as control module, and falls in the following state when abnormality occurs. Seek cause of abnormality according to this table, and take proper countermeasure

=> Also see "Troubleshooting" in the JW300 User's Manual Hardware Version.

Item			Contents	JW300 operating state	Halt output	Control module	Power supply module's indicator		Special relay *3	Error code (BCD)		
										Special register	System memory	
						FAULT	POWER	RUN		□00734	#0160 to 0167	
Self diagnosis	Memory error		Instruction code check	Stop	Open	ON	ON	OFF	007370	20	24	(1)
			System memory setup check								23	(2)
			Program ROM check								25	(3)
			Program sum check								26	(4)
			I/O registration table check								28	(5)
	CPU error		Watchdog timer			OFF		Blinks	—	00	31	(6)
			RAM check (R/W)			ON		OFF	007371	30	32	(7)
			Hardware check								35	(8)
	I/O error	During refresh	I/O data bus						007373	40	44	(9)
			Output data check								42	(10)
			Installed module check								40	(11)
			I/O rack panel error								48	(12)
			Number of module bytes check								45	(13)
		During table verify	Table verify error							60	60	(14)
			Switch verify error								61	(15)
		During table registration	Table registration error							70	70	(16)
			Missing module error								71	(17)
			I/O points overflow error								72	(18)
			Switch set error								73	(19)
	Special I/O error		Hardware error						007375	40	46	(20)
			Parameter error								47	(21)
			*1 *2 Fuse blown of JW-262S	Run	Close	OFF	ON	ON	007363		49	(22)
				Stop	Open	ON	ON	OFF				
	Option error		*1 Hardware error	Run	Close	OFF	ON	ON	007374	50	53	(23)
				Stop	Open	ON	ON	OFF				
			Option command error	Run	Close	OFF	ON	ON			54	(24)
			System protect error	Run	Close	OFF	ON	ON			55	(25)
	Power supply error		Power failure or voltage drop	Stop	Open	OFF	OFF	OFF	007377	10	13 *4	(26)
	Expansion power supply error		Power failure or voltage drop			ON	ON		007376	40	43	(27)
	Battery error		Battery voltage drop or battery not inserted	Run	Close			ON	007372	20	22	(28)
Halt output			Relay output, 100/200 VAC, 30 VDC, 1A, Turn ON (closed) during JW300 operation.									

See the page 5-3 to 5-5

(Setting)	(State)
Continue operation	→ Upper column
Halt	→ Lower column

*1: The upper or lower state of each item may occur when the fuse is melted down in the system memory #0206 or #0207, or by the setting in the case of option abnormality.

*2: When external power is not additionally supplied to the JW-262S, a blown fuse error occurs. Especially be careful when to set #0206 as halt operation.

*3: The special relays 07370 to 07377 turn ON when the self-diagnosis is detected.
When an error occurs, monitor the special relays; using the support tool, host communication, or data link, you can check the error details. (Though the special relays turn ON when an error occurs, the module does not process I/O during occurrence of the error, so you cannot take out data from the output.)

*4: A power supply error is stored when the power is ON, even if the module is normal.

Note: When the control module detects an error while running a self-diagnostic test, an error code will be stored. However, the condition of the operating status, output halted indicator (such as a FAULT), and the special relay will be as follows:
- If the error is released within the watchdog timer cycle time (300 ms), the machine condition will be different from the table above.
- If the error continues, exceeding the watchdog timer cycle time (300 ms), the machine condition will be as shown in the table above.

[1] Description

Error details of (1) to (28) in the previous page after the self-diagnosis are as follows:

(1) Instruction code check

Program memory code is checked when an instruction is executed. Program addresses where instruction code errors are found are stored in system memory locations #0052 and #0053.

(2) System memory setup check

Sum check is done to the system memory #0200 to #0256.

(3) Program ROM check

When operated in the ROM mode, the ROM is checked when transferring program from ROM to RAM. An error is established if a failure is found in the flash ROM.

(4) Program sum check

When the user's program is written or revised, a checksum will be created automatically. When power is supplied, the CPU can use the checksum to detect whether the content of the user program has changed. However, the checksum function cannot be used to identify which part of the program has changed. If a program checksum error occurs, you have to retransmit the system memory settings, program, and data (if required) in order to restore the system.

(5) I/O registration table check

When registering I/O modules, a checksum will be created for the stored data. When power is supplied, the CPU can use the checksum to detect whether the data has changed. If an I/O module table checksum error occurs, you have to do again the I/O module registration.

(6) Watchdog timer

When the watchdog timer in the CPU module times out, this error will occur (the status is the same as in the program mode).

(7) RAM check

Prior to going into every operational cycle, the data memory RAM is checked if read/write is possible.

(8) Hardware operation

Before going into the operation cycle, correct setup of the accumulator and stack is checked.

(9) I/O data bus

Prior to I/O processing, the I/O data bus is examined if it is in the floating state. The faulty module's slot number is stored in system memory location #0046.

(10) Output data check

Rereads and verifies, in I/O processing, the data output to the output module. If verification fails, an error occurs.

- The special I/O module does not perform this check.

(11) Installed module check

Verifies, in I/O processing, the installation state of a module registered in the I/O table when the CPU performs a data exchange with each module. If verification fails, an error occurs.

(12) I/O rack panel error

Checks that all I/O port gates built into the rack panel are closed before I/O processing. If an open port is detected, an error occurs.

(13) Module number of bytes check error

When the PC is processing a 16/32/64 point I/O module, and if the type (number of bytes) registered in the I/O table does not match the internal byte counter, this error will occur.

(14) Table verify error

Verifies, when the power is turned ON or when the mode changes (stop mode to run mode), the installation state of each module with the contents of the I/O table which has a already been registered. If verification fails, an error occurs. This error also occurs when the setting of the rack number switch on the expansion rack panel or the connection state of the I/O expansion cable changes.

(15) Switch verify error (module no. switch verify error)

Additionally verifies the module no. switch setting for the special I/O, option, Device Net or I/O link master module during I/O table verification when the power is turned ON or when the mode changes (stop mode to run mode). If verification fails, an error occurs.

(16) Table registration error

This error occurs when an initial error is generated (e. g. due to an incorrect setting of the rack number switch on the expansion rack panel or an incorrect connection of the I/O expansion cable) during I/O table registration.

(17) Missing module error

An error occurs when the contents of the table data are registered as a "state without any module installation space" during I/O table registration. It is not the state in which no module have been installed.

(18) I/O point overflow error

An error occurs if the number of installed module is too great and the number of I/O points exceeds the number of control I/O points for the control module during I/O table registration.

(19) Switch set error (module no. switch set error)

This error occurs if the settings of the module no. switches on the special I/O or option modules conflict during I/O table registration.

(20) Special I/O hardware error

This error occurs when the watchdog timer for the CPU built into the special I/O module is activated due to an error in the special I/O module itself.

(21) Special I/O parameter error

This error occurs if parameter verification fails when the control module transfers parameters to the special I/O module.

(22) Fuse blown of special I/O (JW-262S)

This error occurs when fuse of JW-262S is OFF.

When no external power is supplied to the JW-262S, a blown fuse error will also occur.

Take special precautions to allow for this case when system memory location #0206 is set to "halt operation."

(23) Option hardware error

This error occurs when the watchdog timer for the CPU built into the module is activated due to an error in the option, device net, or I/O link master module.

(24) Option command error

The data exchange commands between an option module and a control module are checked. If the command is not appropriate, due to an external cause such as electrical noise, this error will occur.

(25) System protection error

When a data exchange command between an option module and control module requests a write to the system area of the control module, this error will occur.

(26) Power supply error

The JW300 ignores instantaneous power failures of 10ms or less and continues operation.

But, a power failure above the limit causes the CPU to halt and the halt output is forced open. The operation is resumed automatically as soon as it recovers from the power interrupt.

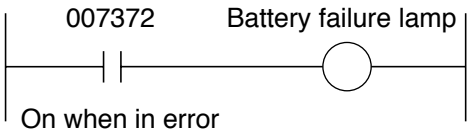
- As a gradual supply voltage drop occurs, the CPU stops when it goes below 85% of the rated voltage and the halt output turned open.
  But, the operation restores automatically when the normal power supply is recovered.

(27) Expansion power supply error

An error occurs when the voltage of the expansion power supply (5 VDC) drops below 4.5 V. When this error occurs, the output module is reset for the rack panel to which is mounted the power supply where the error occurred.

- If another I/O error occurs simultaneously with an expansion power supply error, it may be stored priority expansion power supply module error as error history.

(28) Battery error



Proper battery supply voltage is checked.

Choice is possible using the special relay 007372 to actuate the battery failure lamp or buzzer.

Nothing affects the JW300 operation as long as power is ON to the JW300, even if the battery is in a fault condition. In this event, the battery needs to be replaced with the fresh one to avoid an incidental power supply failure.

[2] Halt output

Though normally closed, the line goes open when an error is detected after diagnosis.

Connection of JW300's halt output to the system's emergency stop circuit enables to shut OFF the system when an error occurred within the programmable controller.

Note: If the error is recovered within the time of the watchdog timer (300 ms), the halt output does not turn ON. If the error is left exceeding the watchdog timer interval (300 ms), the halt output will open.

[3] Special relay

Result of diagnosis is written in the special relay area of the data memory.

When the JW300 stops after an error as a result of diagnosis, the nature of the error may be known when the special relay (007370 to 007377) is examined using the support tool.

- Because the diagnosis runs at every scan cycle, the halt output closes upon recovery from the fault and the JW300 resumes its operation. The diagnostic special relay is also reset.
- Of the special relays, only 007372 (battery error), and 007374 (option module error) can be read from output module by JW300's execution. To read the status of other relays, use the JW-21CM computer link or a support tool.

  The status of special relays cannot be read with the data link either.
- The contacts of relay 007377 (power supply error) is closed for 1 scan at power ON.

[4] Error code

(1) Special register

When an error is established after the diagnosis, the error code is written in the data memory special register (byte address ⊐00734).

- When another error occurred apart from the current error, the error code of higher priority takes preference over other error codes.
- As soon as the error cause is removed, the error code is canceled.

(2) System memory

When an error is established after the diagnosis, the error code is written in the system memory (#0160 to #0167).

#0160 to #0167 acts as a shift register and can store eight errors at a maximum. If more than eight errors occurred, the error code first stored is canceled and so on.

	#0167	#0166	#0165	#0164	#0163	#0162	#0161	#0160	
System memory clear	00	00	00	00	00	00	00	00	
	00	00	00	00	00	00	00	24	← Instruction code error
	00	00	00	00	00	00	24	44	← I/O data bus error
24 (Lost) ←	44	xx	xx	xx	xx	xx	xx	22	← Battery error

- While major error codes are stored in the special register, minor error codes are stored in the system memory.
- The error code in the system memory would not be cleared after recovery from the error. To clear the error code, "00" must be written in the system memory (#0160 to #0167) using the support tool such as the programmer.
- If multiple errors should occur in succession, no error will be written.

(3) Registers

- Registers E7600 to E7777 hold error codes at the time of error occurrence.
  (Setting the $02_{(H)}$ in the system memory #0213)

[5] ON/OFF state of the output module during error

Depending on how the system memory #0232, #0233 (000000 to 015777) and #0252, #0253 (020000 to 075777) output retention address was programmed, ON/OFF state of the output module is determined when the JW300 stops after self-diagnosis.

- Output module before the output retention address --- OFF
- Output module after the output retention address --- Retains ON/OFF state before stop.

But, the output module before the output retention address may not be turned OFF, depending on the nature of the error. The output that needs to be turned OFF at a time of JW300 error must be connected serial to the halt output of the control module. Refer to JW300's user's manual, hardware version.

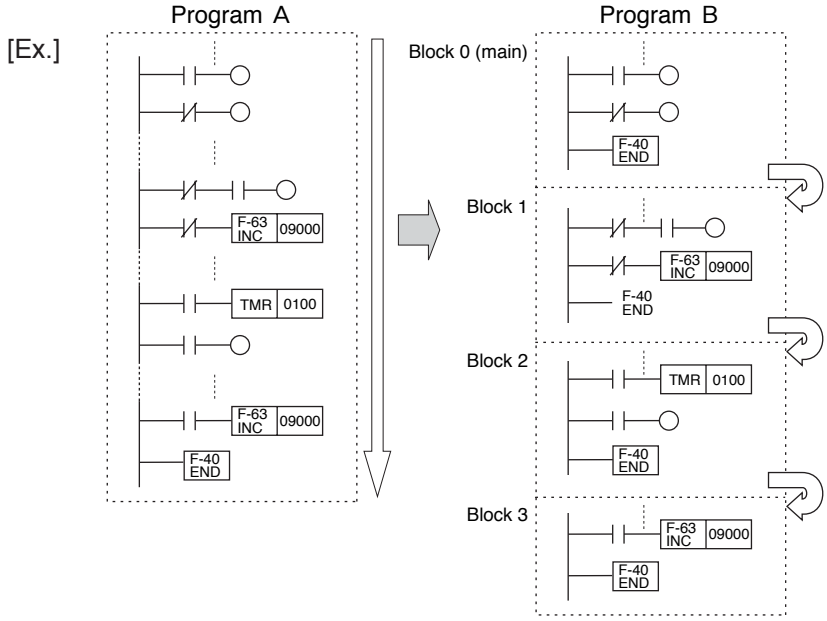
Chapter 6 Block operation, logging, fault diagnostic, and PC card

6-1 Block operation function

The JW300 has a "block operation function" that facilitates programming user programs after dividing into processing blocks (minimum 0.5 K-word) as processing units.

Using the block operation function, you can manage the programs in units of blocks, and handle each block as an independent JW300 so that you can manage the user programs easily. Further, this function is effective for shortening scan time, and convenient for debugging and test operation.

[Ex.]



- This function divides Program A into blocks (as seen in Program B) for each functional purpose and makes management easy. You can operate and wait program for each block, so this is effective for test operation and debugging operation.
- Normally, execute only the needed blocks. For each scan, un-needed blocks are not scanned, so that total scan time can be shortened.

■ Processes available to execute in block units

Process	Description
Operation/wait	Using the support tool (JW-15PG, JW-300SP) and block start relay, you can run (start) and wait (halt) program operation for each block. The waiting blocks do not execute programs and remain in stop mode status.
I/O refresh	I/O refresh operation can be set and executed for each block.
Program management	Including program changes of the currently running program, each block can be handled as an independent program.

■ Number of blocks and sub programs

The numbers of blocks and sub-programs that can be used with the block operation vary with the model of the JW300 control modules. => See the structuring program (page 4-2).

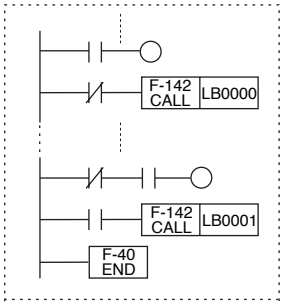
	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
Number of blocks (max.)	16	32	64	128	256	512
Number of sub programs (max.)	256	512	1024	2048	4096	8192

■ Jump between blocks

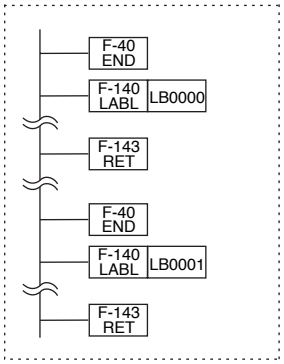
The cursor can be jumped between blocks.
In the example below, Block 0 (main) is used as the normal user program and Block 1 as the subroutine block so that you can easily manage the blocks.
In this case, even though the block is in waiting status, the JW300 executes calculation of the jumped-to target block.

[Ex.]

Block 0 (main)



Block 1

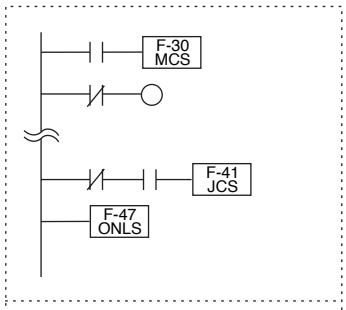


■ MCS and ONLS instructions

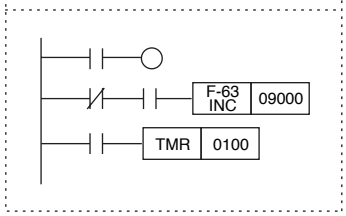
The MCS (F-30) and ONLS (F-47) instructions do not need to be reset, as these operations do not affect the next blocks.
The contents of the accumulator (ACC) and stack register (SR), which were stored by the PUSH instruction, are all initialized when starting the operation of each block.

[Ex.]

Block 0 (main)



Block 1



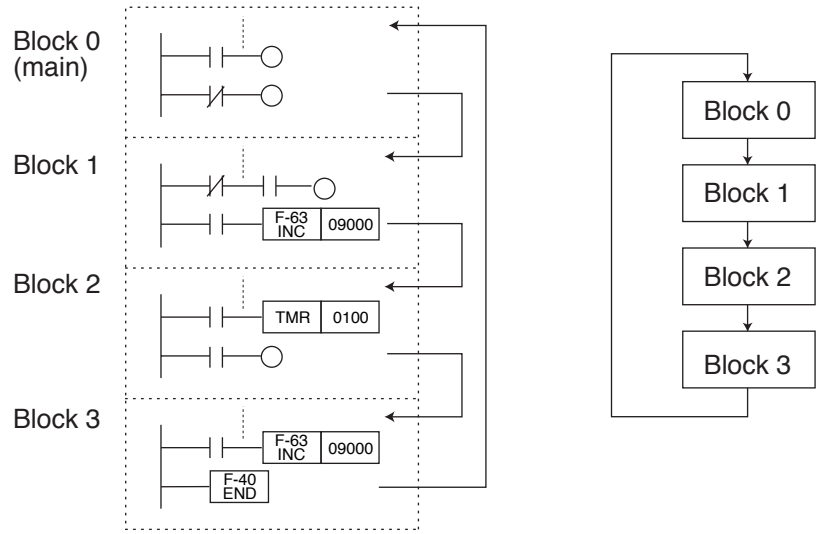
- The MCS (F-30) and ONLS (F-47) instructions do not need to be reset at Block 0, as these operations do not affect Block 1.

[1] Set block status

Each block status (run/wait) can be set using support tools (JW-15PG, JW-300SP). This is convenient for debugging of each device and partial debugging.

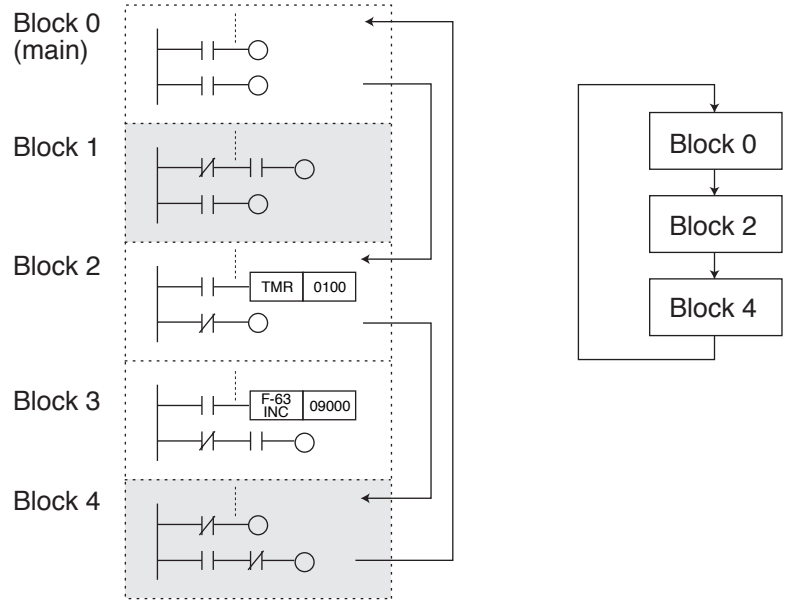
[Ex. 1]

- Number of blocks = 4
- Status setting = Run all blocks



[Ex. 2]

- Number of blocks = 5
- Status setting
  - Block 0, 2, 4 = Run
  - Block 1, 3 = Wait



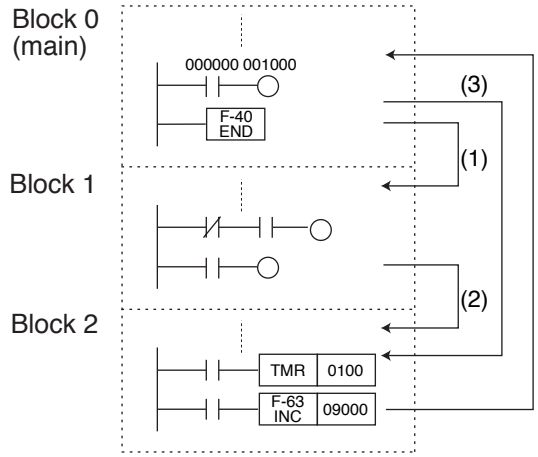
[2] Block start relay

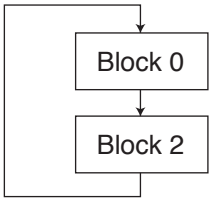
Set block start relays to each block (Block 1 and after) using the support tool (JW-15PG, JW-300SP), and you can control run (start) and wait (halt) each block using the ladder program.

[Ex. 1]

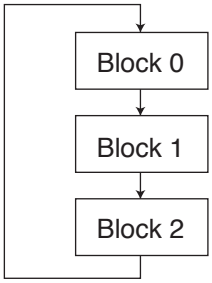
- Number of blocks = 3
- Setting block start relay

  Block 1 = Run when relay 001000 is ON
  Block 2 = No block start relay



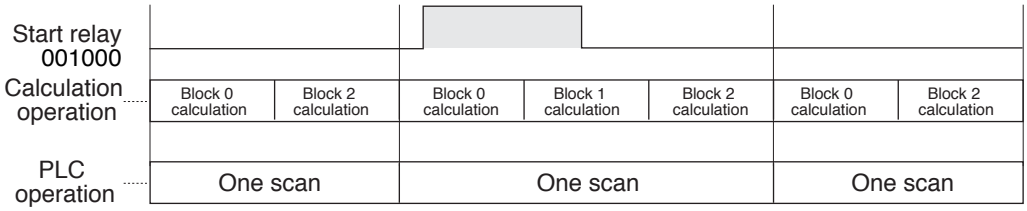


Run when relay 001000 is OFF



Run when relay 001000 is ON

(1) When relay 001000 is ON
(2) When block 1 is executing
(3) When relay 001000 is OFF

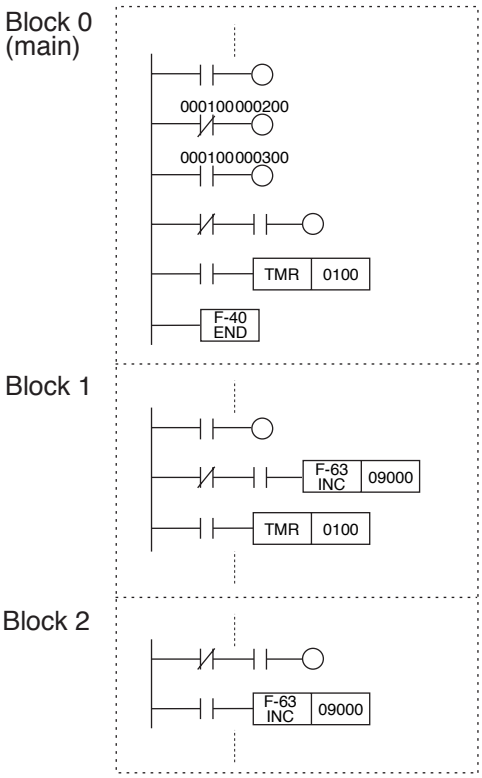


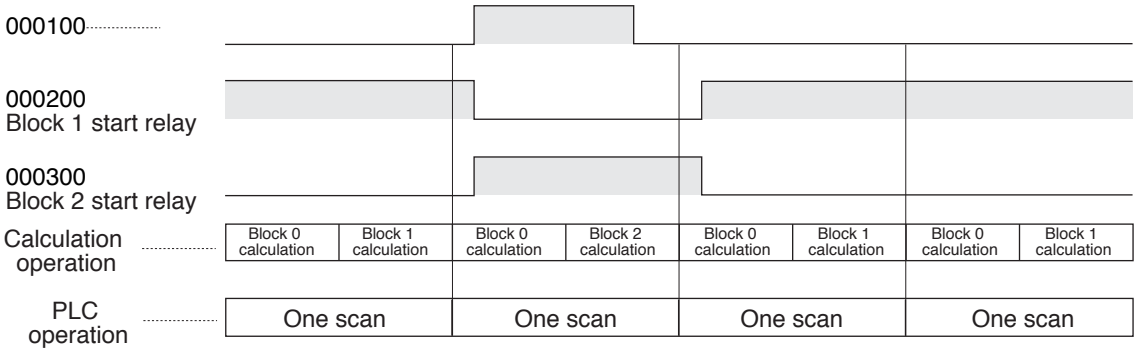
[Ex. 2]

Using the block start relay, you can branch blocks.
Below is an example that when relay 000200 is ON, Block 1 runs, and when relay 000300 is ON, block 2 runs (starts).

- Number of blocks = 3
- Setting block start relay

  Block 1 = Runs when relay 000200 is ON
  Block 2 = Runs when relay 000300 is ON



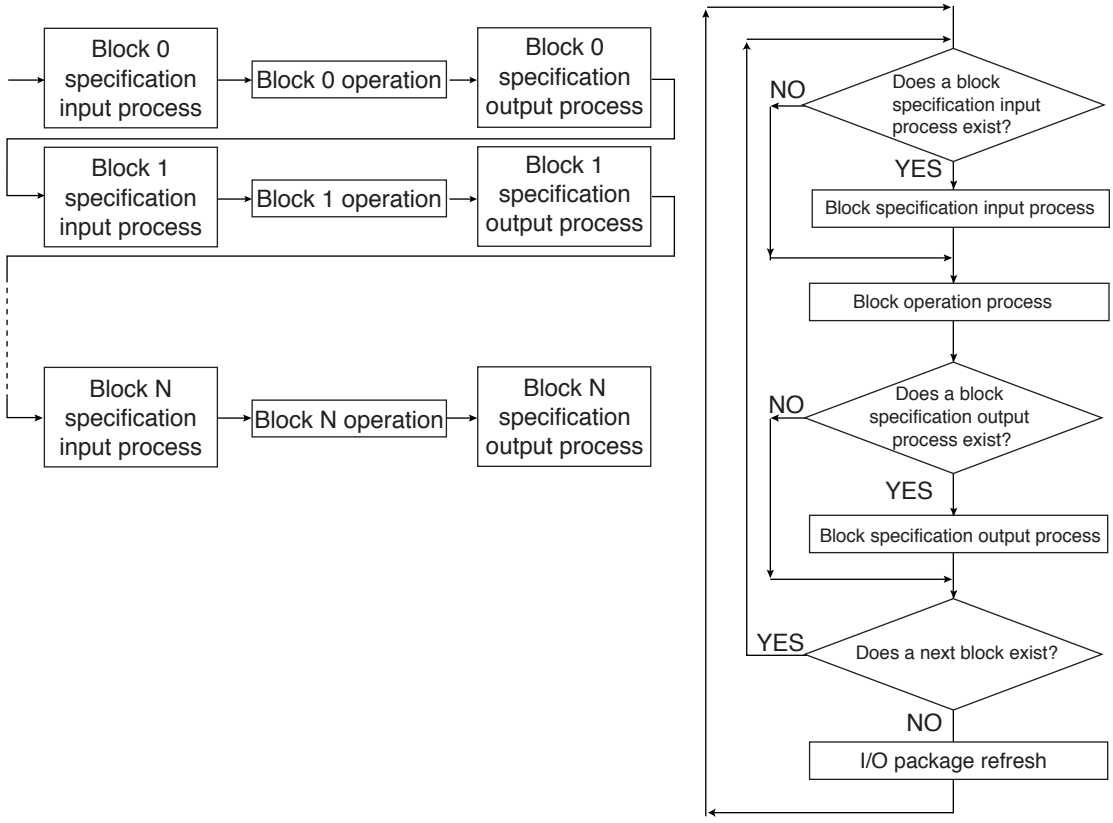


[3] I/O refresh

Specified I/O can be refreshed at each block.

I/O processing for each rack/slot can be set for each block, and you can process each I/O quickly.

■ **I/O refresh process**



[Precautions]

- When the I/O refresh is not set for each block, the JW300 does not I/O refresh the not yet set blocks.
- I/O refresh is not executed for blocks in waiting status.
- Do not use addresses of input modules that are used for input interrupt (system memory #0240 to #0245).
- The upper limit of the slot number varies with the rack panel used.

6-2 Logging function

Taking specified registers as the trigger condition, the data memory can be logged. This logging function is useful for analyzing devices when an error occurs, and data is collected at the rated time.

To use the logging function, you have to program application instruction F-403 (LOG) and set system memory #2100 to #2155.

- Application instruction F-403 (LOG) —— Trigger logging => [1]
- System memory

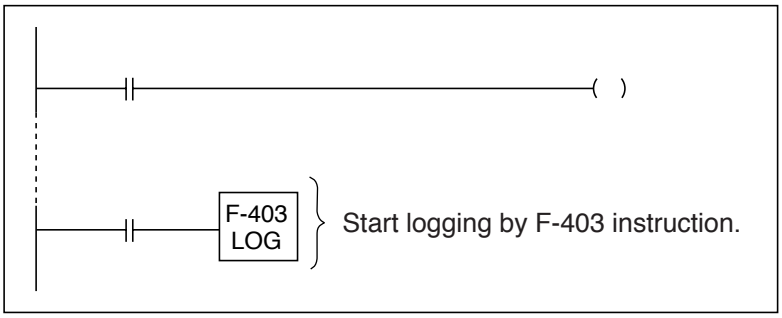
#2100 to #2155
- Top address of storage area (#2100 to #2103)
- Number of storages (#2104)
- Time stamp (#2105)
- Specify register address (#2106 to #2155)

=> [2]

[1] Programming application instruction F-403 (LOG)

Triggered by the application instruction F-403 (LOG) for logging, the JW300 starts logging.

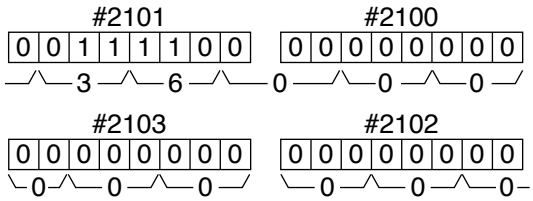
- Operation condition of F-403 (LOG) is the rising edge of the input signal (OFF to ON).



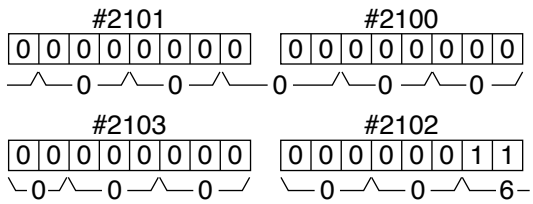
[2] Setting system memory #2100 to #2165

Specify top address of area to store (#2100 to #2103), number of storages (#2104), time stamp (#2105), and register address for logging (#2106 to #2155) to each system memory.

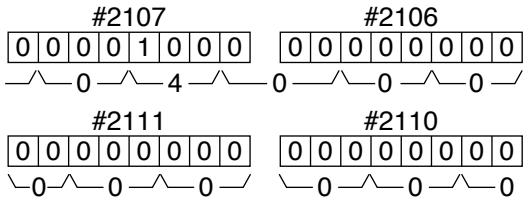
<table>
<tr><th>System memory No.(8)</th><th>Set item</th><th>Description</th></tr>
<tr><td>#2100
to #2103</td><td>Top address of logging data storage area</td><td>Set top address of area to store logging data to file address #2100 to #2103.

Ex. 1: When setting register 109000
 

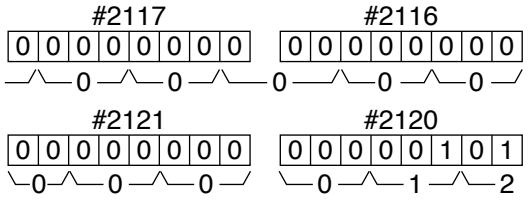
- File address of register 109000 is 00036000(8).

Ex. 2: When setting file register 00400000(8)
 

- File address of file register 00400000(8) is 00600000(8)</td></tr>
<tr><td>#2104</td><td>Number of storage of logging</td><td>Specify the number of storages of logging data (set value x 10).
Set range is 001 to 225(D).
Ex.: When the set value is 050(D), the JW300 store logging data for 50 x 10 = 500 times.</td></tr>
<tr><td>#2105</td><td>Time stamp format</td><td>Select format of time stamp that is added to logging data.
<table>
<tr><th>Set value (H)</th><th>Format</th><th>No. of bytes used</th></tr>
<tr><td>00 (Initial value)</td><td>None</td><td>0</td></tr>
<tr><td>01</td><td>Year, month, day, hour, minute second</td><td>6</td></tr>
<tr><td>02</td><td>Hour, minute second</td><td>3</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>System memory No.(8)</th><th>Set item</th><th>Description</th></tr>
<tr><td>#2106
to #2155</td><td>Logging specification register</td><td>
Up to 10 register addresses for logging can be set using file address.
<table>
<tr><th></th><th>System memory number</th></tr>
<tr><td>Logging register 1</td><td>#2106 to #2111</td></tr>
<tr><td>Logging register 2</td><td>#2112 to #2115</td></tr>
<tr><td>Logging register 3</td><td>#2116 to #2121</td></tr>
<tr><td>Logging register 4</td><td>#2122 to #2125</td></tr>
<tr><td>Logging register 5</td><td>#2126 to #2131</td></tr>
<tr><td>Logging register 6</td><td>#2132 to #2135</td></tr>
<tr><td>Logging register 7</td><td>#2136 to #2141</td></tr>
<tr><td>Logging register 8</td><td>#2142 to #2145</td></tr>
<tr><td>Logging register 9</td><td>#2146 to #2151</td></tr>
<tr><td>Logging register 10</td><td>#2152 to #2155</td></tr>
</table>
- Logging registers without a specified system memory number shall be set to FFFFFFFF(H) (initial value).
Ex. 1: When setting register 009000 to logging register 1


- File address of register 009000 is 00004000(8).

Ex. 2: When setting file register 01000000(8) to logging register 3


- File address of file register 01000000(8) is 01200000(8).
</td></tr>
</table>

System memory No.(8)	Set detail
#2156 to #2161	To be a counter to store next logging data. 0 to [Number of logging storages (#2104) -1]
#2162 to #2165	Number of times to store logging data. 0 to $FFFFFFFF_{(H)}$

[3] Use examples

This section describes examples of using of the logging function.

(1) Setting example of system memory (#2100 to #2155)

Below shows logging data storage examples if the following are set to system memory (#2100 to #2155) of the logging function.

Item	System memory No.	Set value	Description
Top address of logging data storage area	#2100 to #2103	$00200000_{(8)}$	File register $00000000_{(8)}$ => File address $00200000_{(8)}$
Number of logging storages	#2104	$20_{(D)}$	200 sets ($20_{(D)}$ x 10)
Time stamp	#2105	$01_{(H)}$	Year, month, day, hour, minute, second
Logging register 1	#2106 to #2111	$00000000_{(8)}$	↴00000 => File address $00000000_{(8)}$
Logging register 2	#2112 to #2115	$00000001_{(8)}$	↴00001 => File address $00000001_{(8)}$
Logging register 3	#2116 to #2121	$00000002_{(8)}$	↴00002 => File address $00000002_{(8)}$
Logging register 4	#2122 to #2125	$00000003_{(8)}$	↴00002 => File address $00000003_{(8)}$
Logging register 5	#2126 to #2131	$00004000_{(8)}$	009000 => File address $00004000_{(8)}$
Logging register 6	#2132 to #2135	$00004001_{(8)}$	009001 => File address $00004001_{(8)}$
Logging register 7	#2136 to #2141	$00004002_{(8)}$	009002 => File address $00004002_{8)}$
Logging register 8	#2142 to #2145	$00004003_{(8)}$	009003 => File address $00004003_{(8)}$
Logging register 9	#2146 to #2151	$00016000_{(8)}$	E0000 => File address $00016000_{(8)}$
Logging register 10	#2152 to #2155	$00016001_{(8)}$	E0001 => File address $00016001_{(8)}$

■ Storage example of logging data

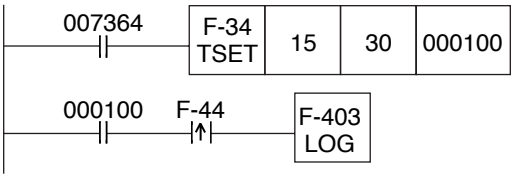
When the system memory is set as the example in the previous page, the JW300 stores logging data to area starting from file register 00000000(8).

File register(8)	Data (Ex.)	Description	
00000000	03	Year	1st logging data
00000001	11	Month	
00000002	20	Day	
00000003	13	Hour	
00000004	10	Minute	
00000005	35	Second	
00000006	02	Logging specification register 1 (data of ⊐00000)	
00000007	05	Logging specification register 2 (data of ⊐00001)	
00000008	0A	Logging specification register 3 (data of ⊐00002)	
00000009	A5	Logging specification register 4 (data of ⊐00003)	
0000000A	F1	Logging specification register 5 (data of 009000)	
0000000B	E0	Logging specification register 6 (data of 009001)	
0000000C	09	Logging specification register 7 (data of 009002)	
0000000D	03	Logging specification register 8 (data of 009003)	
0000000E	24	Logging specification register 9 (data of E00000)	
0000000F	57	Logging specification register 10 (data of E00001)	
00000C70	03	Year	200th logging data
00000C71	11	Month	
00000C72	20	Day	
00000C73	18	Hour	
00000C74	25	Minute	
00000C75	14	Second	
00000C76	02	Logging specification register 1 (data of ⊐00000)	
00000C77	05	Logging specification register 2 (data of ⊐00001)	
00000C78	0A	Logging specification register 3 (data of ⊐00002)	
00000C79	A5	Logging specification register 4 (data of ⊐00003)	
00000C7A	F1	Logging specification register 5 (data of 009000)	
00000C7B	E0	Logging specification register 6 (data of 009001)	
00000C7C	09	Logging specification register 7 (data of 009002)	
00000C7D	03	Logging specification register 8 (data of 009003)	
00000C7E	24	Logging specification register 9 (data of E00000)	
00000C7F	57	Logging specification register 10 (data of E00001)	

- From the 201st, logging data are overwritten on the 1st data and after.

(2) Program example of application instruction F-403 (LOG)

If the application instruction F-403 (LOG) for logging is programmed as follows, the JW300 stores data of the specified register at 15:30 as logging data.



Remarks

- If the number of logging storages (system memory #2104) is set to $000_{(D)}$, the JW300 logs data only one time.
- If an address out of the set range is specified in the logging specification register (system memory #2106 to #2155), the JW300 will not store its address data. It will store only data in the setting range.
- When the number of logging times exceeds the specified number of storages, the JW300 overwrites the 1st logging data and after. Another exceeding of the number of logging times also overwrites from the 1st logging data.
- When you will use an extension RAM using a PC card, the last address of the logging data storage area can be set up to file address $20077777777_{(8)}$.
- When the logging system memory (#2100 to #2155) is changed during operation of the JW300, the set value when the application instruction is F-403 (LOG) will be effective.

6-3 Fault diagnosis function

By monitoring the execution time of each cycle, the relays can be diagnosed. Monitoring using conventional user programs can be performed by this fault diagnosis function. By using this function, you can save the user program and obtain error relay status immediately, so that maintenance ability can be improved.

To enable/disable the fault diagnosis function, set system memory #2200.

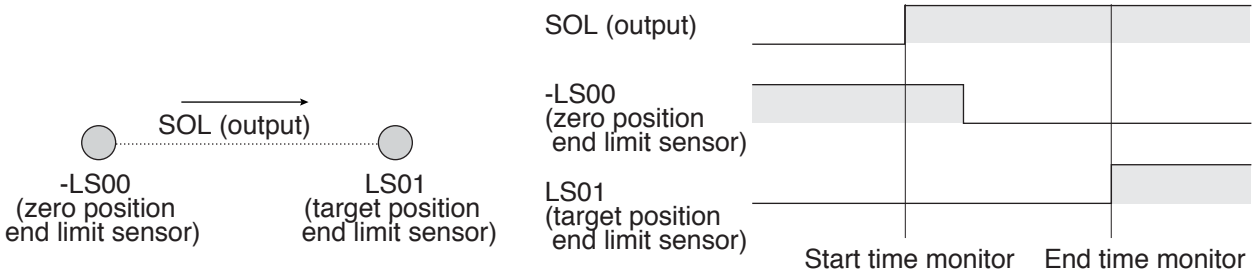
Address	Data	Description
#2200	00(H)	Initial value
	01(H)	Enable
	Other than above	Disable

Setting available number of cycles for each model (control module) shall be as follows:

	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
Number of I/O cycle	128	256	512	1024	2048	4096
Number of always monitoring relays	128	256	512	1024	2048	4096
Monitoring time	0 to 32767 ms					

[1] Faulty diagnosis principle

(1) I/O cycle



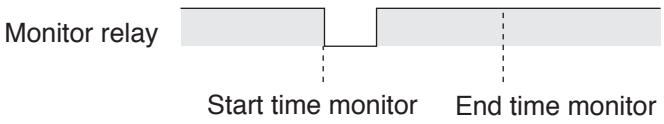
When a machine moves from zero position end limit sensor (-LS00) to target position end limit sensor (LS01), the JW300 monitors the zero position end limit sensor (-LS00) and the target position end limit sensor (LS01) for a period after the output (SOL) turns ON until the target position end limit sensor turns ON, and detects following errors.

Monitor output	Machine status
The target position end limit sensor does not turn ON.	The zero position end limit sensor is OFF and the target position end limit sensor does not change from OFF to ON.
No output operation	The zero position end limit sensor does not change from ON to OFF. The target position end limit sensor does not change from OFF to ON.
Zero position end limit sensor does not turn OFF.	The target position end limit sensor is ON and the zero position end limit sensor does not change from ON to OFF.
Zero position is ON.	Zero position end limit sensor of the previous output is turned ON.
Target position end limit sensor is OFF.	Target position end limit sensor of the previous output is turned OFF.

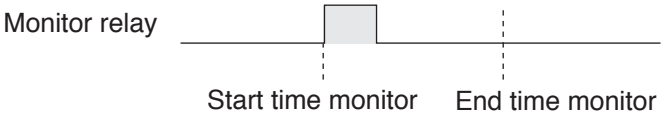
(2) Always monitor relay

This is a function to always monitor a specified relay.
This function monitors whether a relay will go OFF within a specified interval, and whether a relay will go ON within a specified interval.

- Monitor to be ON



- Monitor to be OFF

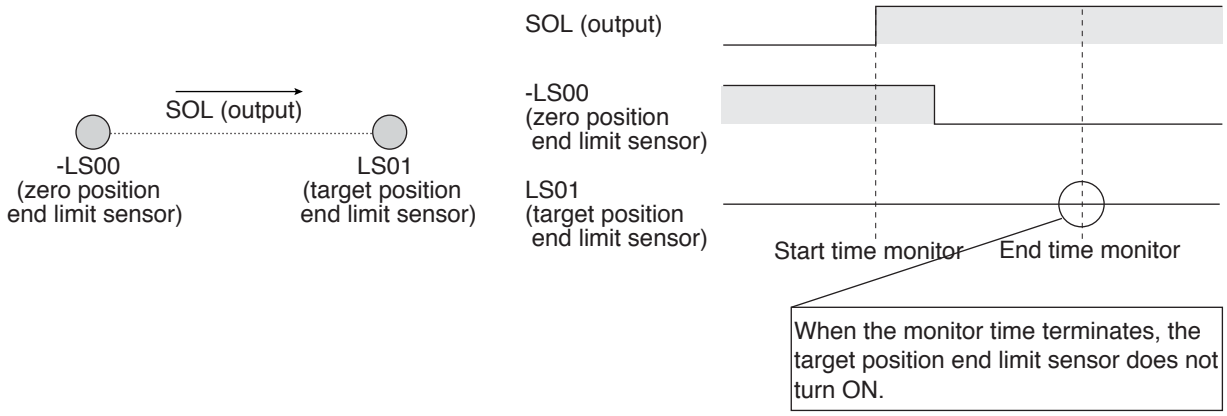


[2] Error details of fault diagnosis

(1) Error on I/O cycle

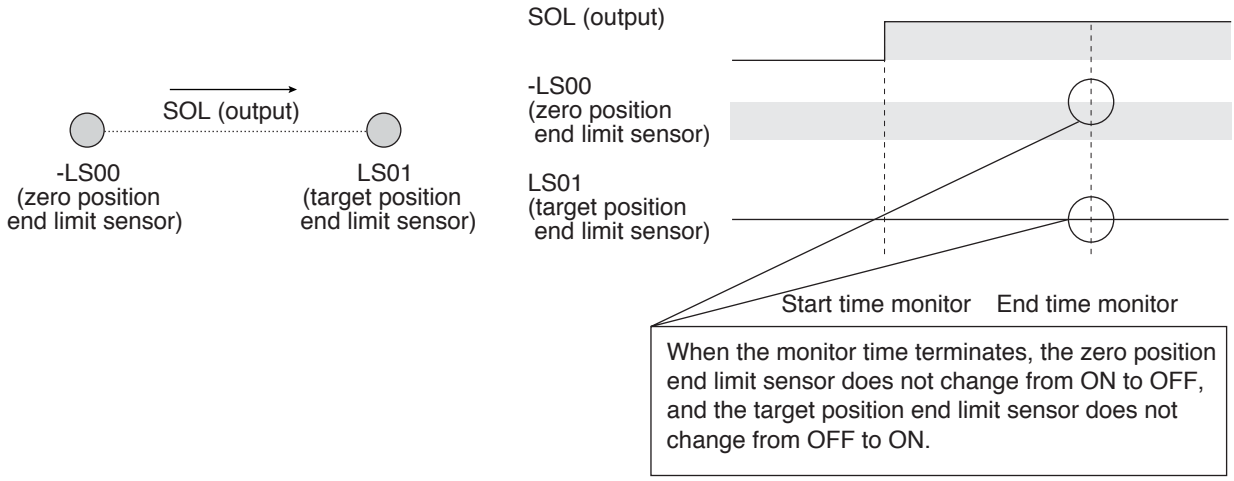
1) Does not turn ON the target position end limit sensor

The zero position end limit sensor is OFF and the target position end limit sensor does not change from OFF to ON



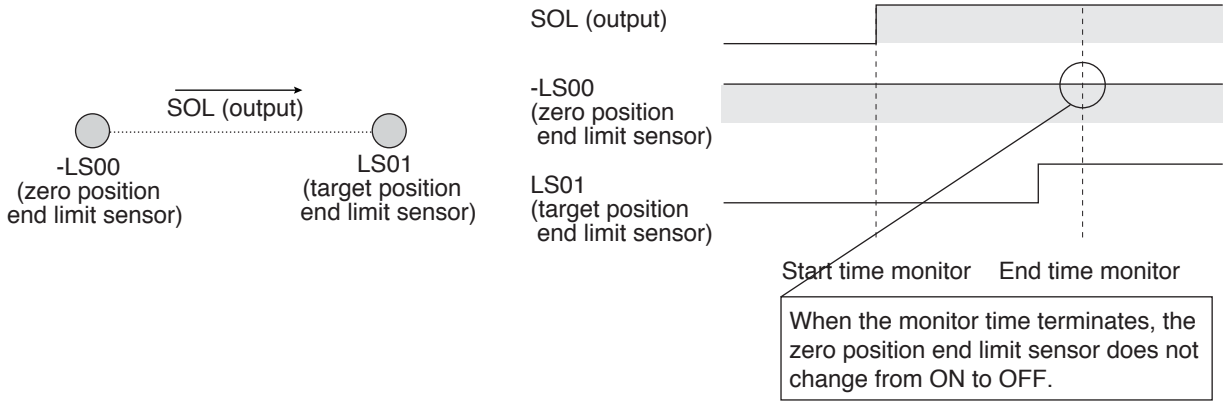
2) Does not output

The zero position end limit sensor does not change from ON to OFF, and the target position end limit sensor does not change from OFF to ON.



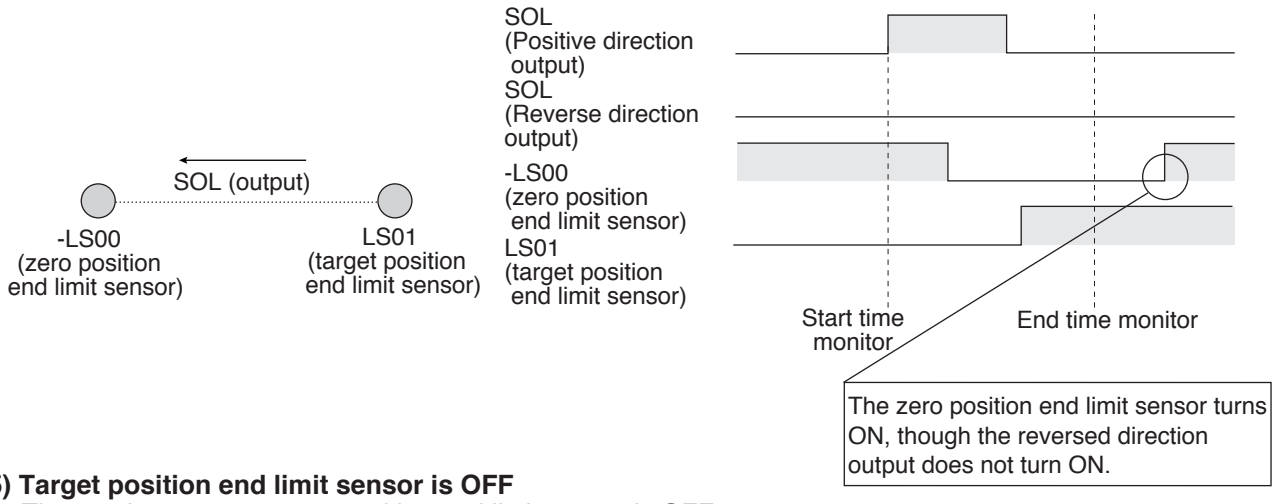
3) The zero position end limit sensor does not turn OFF.

The target position end limit sensor is ON and the zero position end limit sensor does not change from ON to OFF.



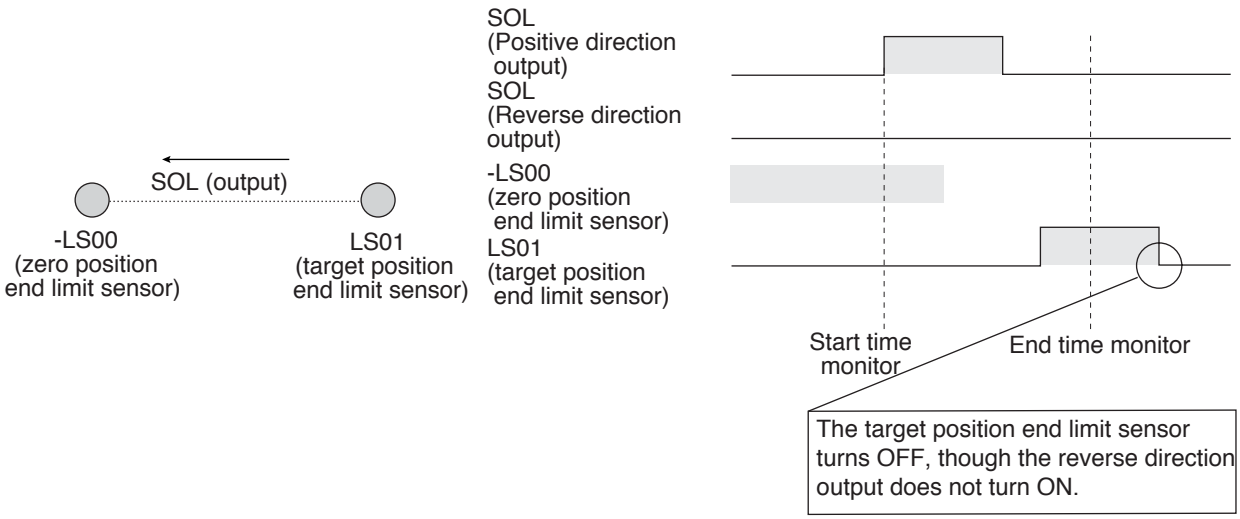
4) Zero position end limit sensor turns ON

The previous output zero position end limit sensor turns ON.



5) Target position end limit sensor is OFF

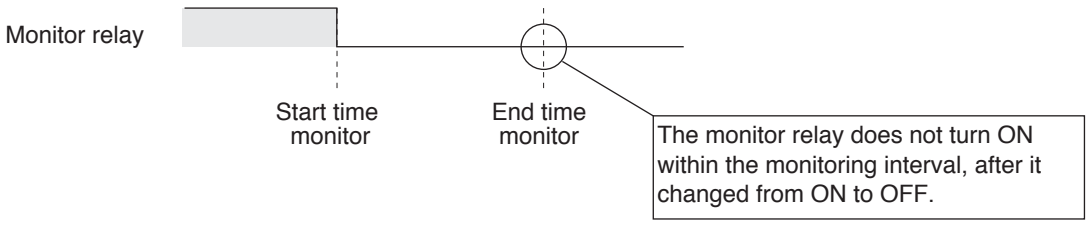
The previous output target position end limit sensor is OFF.



(2) Error of always monitor relay

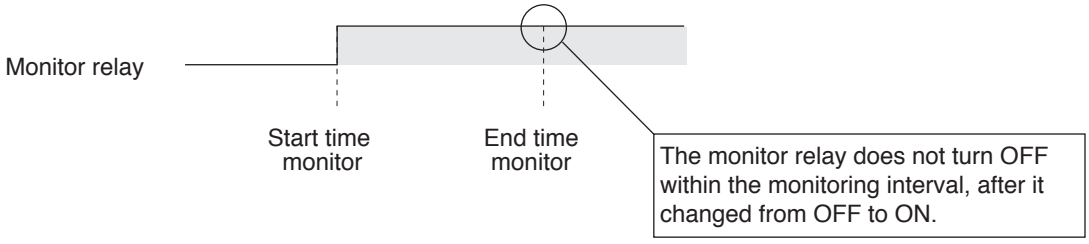
1) ON monitor error

An ON monitoring relay is OFF.



2) OFF monitor error

When an OFF monitoring relay turns ON



[3] Setting details

To use the fault diagnosis function, set the following items using the support tool (JW-300SP).

Set item	Description
Diagnosis approval relay	A relay to start the fault diagnosis function. When this is OFF, the JW300 does not diagnose faults.
Diagnosis result reset relay	When this is set to ON, clear all the diagnosis results.
History setting	Store the history of error occurrence to data or file memory.
Output relay when diagnosis result has an error	If an error is detected after fault diagnosis, this relay turns ON.

Remarks

- When the fault diagnosis is selected, total scan time will be longer. The execution time of the fault diagnosis is as follow:
  24 μs + (3 μs x N cycles)

6-4 How to use the PC card

The JW300 control module (JW-3*2CU) is equipped with a memory card interface. Usage of the memory card is as follows:

(1) Save and load files

You can save and load user programs and data into a CF card. If there is no support tool, you can use the card for saving data on site. => See page 6-20 to 23.

(2) Memory capacity extensions

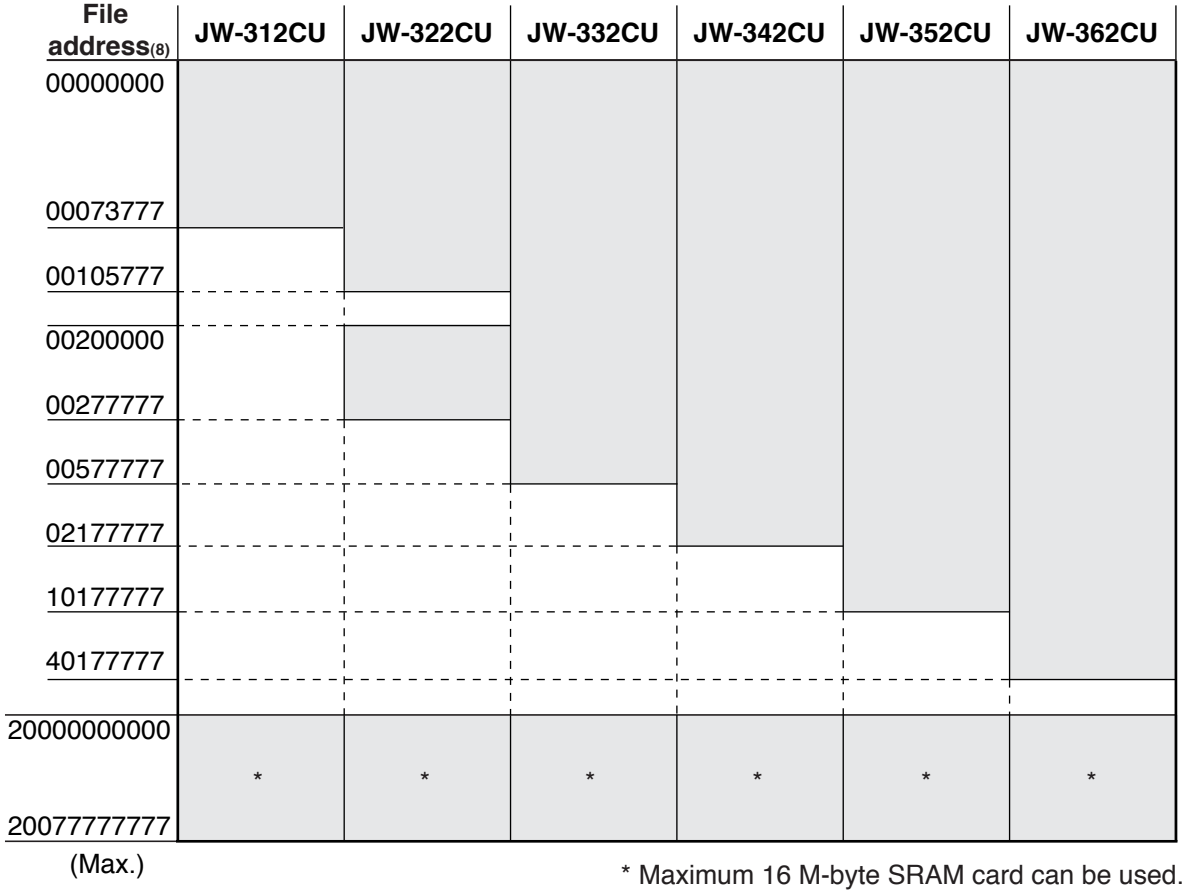
Using a SRAM card, data memory capacity can be extended (up to 16 M bytes. => See table below).

- The card can be used to store symbol/comment data of a user program.
- Using the card, your can log in a large volume of operations and it can be used for trouble analysis and examination for shortening tact time.

● **Memory card interface of JW300 (JW-3*2CU)**

Item	Specifications
Interface (connector)	Connector for PC card type I/II (68 pins)
Power supply voltage	3.3 V / 5 V power supply
Usable memory card	CF card (need a conversion adapter) SRAM card (small size types need a conversion adapter)
Objective data to save	1) When used as save and load files - User program - Parameter memory - System memory - Data memory 2) When used as memory extension - Data memory (symbol, comment data, logging data)

● **Memory extension using a SRAM card**



* Maximum 16 M-byte SRAM card can be used.

[1] Objective files for save and load

To save and load a user program and data, use a CF card.

(1) CF card applicable format

FAT12
FAT16
VFAT
FAT32

- When VFAT and FAT32, long file name cannot be used.

(2) Type and extension of file

Files that can be saved/loaded into/from a CF card are as follows. The saved file has the following extension.

File type	Extension	Description
User program	ppg	Use as a set of block data and titles of block/sub program.
Block data	pbk	Data concerning blocks. When the user program is saved, the block data is created. Use as a set of user program and titles of block/sub program.
Title of block and sub program	ptl	Title data of block/sub program. It is created to save a user program. Use as a set of user program and block data.
System memory	psm	System memory data are loaded and saved. - When saving, the range is #0000 to #2777. When loading, the range is #00200 to #02777.
Parameter memory	ppa	All of parameter memory data are saved/loaded.
Data memory (except file register)	pdt	All of data in the data memory (except file register) are saved/loaded.
File register	pfl	All of data of the file register are saved/loaded.

(3) File name

Set the file name with a maximum of 8 characters.
The following characters cannot be used.

「¥」 「/」 「:」 「,」 「;」 「*」 「?」 「‘‘」 「>」 「<」 「|」

(4) Directory

The JW300 cannot use files in the directory. Make sure to enter into root directory of CF card.

(5) Files created by other model

Files that were saved by another model of control module (JW-3*2CU) cannot be used. However, after the file has been converted using the support tool (JW-300SP), it can be used again.

[2] Setting system memory

Setting details for the CF card are as follows.

<table>
<tr><th>System memory No. (8)</th><th>Set item</th><th>Description</th></tr>
<tr><td>#2220</td><td>Prohibit to save into CF card</td><td>Specify prohibit/allowed saving data to CF card.
Bit 7 6 5 4 3 2 1 0
#2220 0 0 0 0 0 0 0 [ ] Initial value: 00(H)
Write (1 (ON) --- Prohibit
0 (OFF) --- Allowed)</td></tr>
<tr><td>#2221</td><td>File type selection</td><td>Select file type to save and load to CF card.
Bit 7 6 5 4 3 2 1 0
#2221 0 0 0 [ ] [ ] [ ] [ ] [ ] Initial value: 1F(H)
User program (including block data and tile of block and sub program.)
System memory
Parameter memory
Data memory (except file register)
File register
- When the JW-312CU is used, file register cannot be selected.</td></tr>
<tr><td>#2222</td><td>Select save/load to/from CF card</td><td>For a file selected by #2221, select save (write data from JW300 to CF card) or load (read data from CF card to the JW300).
<table><tr><th>Setting (H)</th><th>Description</th></tr><tr><td>11</td><td>Save</td></tr><tr><td>22</td><td>Load</td></tr></table>Initial value: 00(H)
- File name that can be handed by #2222 is "SHARP.***" only.
- While the JW300 is operating, data cannot be saved/load to/from a CF card.
- Before saving data, make sure to format the CF card.
- After the data is saved/loaded, the set value returns to 00(H). (Return to 00(H) also when an error occurs)</td></tr>
<tr><td>#2223
#2224
#2225</td><td>Setting auto save of CF card</td><td>With this setting, the JW300 automatically saves data to the CF card at a specified interval. Set the file (type) to save in #2223.
7 6 5 4 3 2 1 0
#2223 [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] Initial value: 00(H)
Execution unit (1 to 127 days)
Execution flag (1 (ON) --- Execute
0 (OFF) --- Does not execute)
7 6 5 4 3 2 1 0
#2224 [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] Initial value: 00(H)
Hour (0 to 23)
7 6 5 4 3 2 1 0
#2225 [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] Initial value: 00(H)
Minute (0 to 59)
- File names are "AT Year Year Month Month Month. ***"
- When 00(H) is set as execution unit, it will be 128 days.
- Set "2224 and #2225 with BCD.</td></tr>
<tr><td>#2226</td><td>CF card error code</td><td>When an error is found during saving/loading data, an error code corresponding to the error will be stored. => See page 6-23.
Even if the error is released, the error code is not cleared. To clear the data, use the support tool (JW-15PG, JW-100SP).</td></tr>
<tr><td>#2230
to #2236</td><td>Automatic save file name</td><td>When the AUTO LD switch on the JW-3*2CU is OFF and address #2230 or up is specified, the file is automatically saved when inserting the CF card.
Specify file types to save on the "#2221."
File name will be "@*******" Initial value: 00(H)</td></tr>
</table>

[3] Operation procedure to save and load to/from CF card

This chapter describes each operation procedures to save/load files to/from a CF card.
Note: While the JW300 is in operation, you cannot save/load files to/from the CF card.

(1) File save procedure

To save files by system memory setting (write from the JW300 to the CF card), follow the procedures below.

[Ex.] Save user program

1. Insert an already formatted CF card into the PC card slot of the JW300 control module (JW-3*2CU).
2. Set $01_{(H)}$ to system memory #2221.
3. Set $11_{(H)}$ to system memory #2222.
4. When system memory changes to $00_{(H)}$, the save file is complete.

   => Files with file name "SHARP.ppg" "SHART.pbk" "SHART.ptl" are written to the CF card.

If there is a file having the same name in the CF card, it will be overwritten.

(2) File load procedure

To load files by system memory setting (read from the CF card to the JW300), follow the procedures below

Ex.: Load user program.

1. Insert an already formatted CF card into the PC card slot of the JW300 control module (JW-3*2CU).
2. Stop operation of the JW300 control module.
2. Set $01_{(H)}$ to system memory #2221.
3. Check that files with file name "SHARP.ppg", "SHARP.pbk" and "SHARP.ptl" exist in the CF card. (For handling program memory, if all of these files do not exist, the files cannot be loaded.)
5. Set $22_{(H)}$ to system memory #2222.
6. When system memory #2222 changes to $00_{(H)}$, the load is complete.

(3) File auto save procedure

With a specified time, files are saved/loaded to /from the CF card. Time interval is after D7 of system memory #2223 turns ON. Set as follows:

Ex.: Save user program at 12:30 every 7days.

1. Insert an already formatted CF card into the PC card slot of the JW300 control module (JW-3*2CU).
2. Check the clock of the JW300 control module.
3. Set $01_{(H)}$ to system memory #2221.
4. Set $12_{(BCD)}$ to system memory #2224, and $30_{(BCD)}$ to #2225.
5. Set $87_{(H)}$ to system memory #2223.
6. For 7 days from step 5 above, files of file name "AT030601.ppg" "AT030601.pbk" "AT030601.ptl" are saved to the CF card. (The file name above is true when June 1st, 2003.)

(4) Auto save/load procedure

1) To load files in the CF card into the JW300 when turning ON the power

While the CF card is inserted into the JW300 control module (JW-3*2CU), if the power is input, if a file having file name "AUTOLOAD" exists in the CF card, and if the AUTO LD switch of the JW-3*2CU is turned ON, the files are loaded from the CF card to the JW300. There is no special setting. All of the files having file name "AUTOLOAD" in the CF card are loaded.

Remarks

When loading the "user program," files of "block data" and "titles of block and sub program" are needed. The user program will not load without them, unless these three files do no exist.

2) When inserting the CF card, save the files of the JW300 into the CF card

When the CF card is inserted into the JW300 control module (JW-3*2CU), and if the AUTO LD switch of the JW-3*2CU is OFF, the files of the JW300 are saved to the CF card. The files to save shall be set in system memory #2221. File names will be those set in system memory "2230 to "2236 and "@******* " If the $00_{(H)}$ is set to system memory #2221, the auto save is not executed.
If a file with the same name exists in the CF card, it will be overwritten.

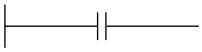
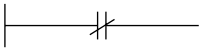
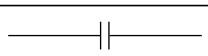
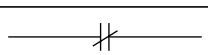
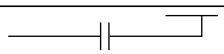
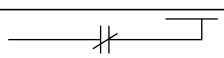
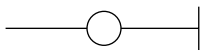
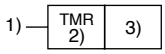
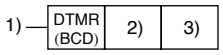
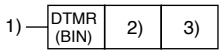
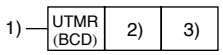
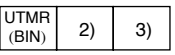
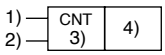
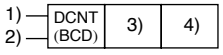
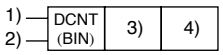
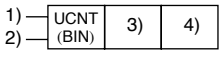
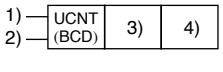
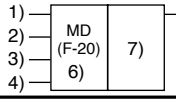
[4] Error code

If an error occurs on the CF card operation, any of the following codes in the table below will be stored in system memory #2226.

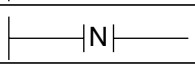
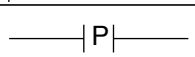
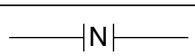
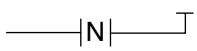
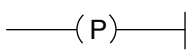
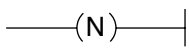
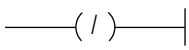
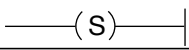
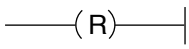
Error code(H)	Error detail
00	Normal complete
01	Directory name error of specified file
02	Same file or directory name exists.
03	The specified file does not exist.
05	Specified mode error.
06	File open error.
07	The specified file is already opened.
08	Device read error
09	Device write error
0A	The number of files opened at the same time exceeds the limit.
0B	Disk full.
0D	Specified parameter error.
0E	End of the file was reached.
10	Not supported.
11	Not initialized (relation with the driver and file system).
12	Not initialized (initialization of file system).
13	Is not mounted.
16	Device initialization error.
17	Device mount error.
18	Shorted cache size.
19	Write protected.
63	Others

Chapter 7 List of instructions

7-1 List of basic instructions

Mnemonics	Symbol	Words	Function	Execution condition	Flag Zero 007357	Flag Carry 007356	Flag Error 007355	Flag Non-carry 007354	See page
STR		1	Starts at normally open contact and intermediate result is stored.						8-2
STR NOT		1	Starts at normally closed contact and intermediate result is stored.						3
AND		1	AND						4
AND NOT		1	AND NOT						4
OR		1	OR						5
OR NOT		1	OR NOT						5
AND STR		1	AND with the intermediate result						6
OR STR		1	OR with the intermediate result						7
OUT		1	Outputs result						8
TMR		2	Timer (decrement) 1) Start input (starts with ON) 2) TMR number (00000 to 17777) Internal clock 0.1 or 0.01 sec. 3) Setting value 0.1 to 799.9 sec. 0.01 to 79.99 sec. 0.001 to 7.999 sec.	Start input ON					9
DTMR (BCD)		3	Timer (decrement) 1) Start input (starts with ON) 2) TMR number (00000 to 17777) 3) Setting value (0.1 to 799.9 sec.)	Start input ON					9
DTMR (BIN)		3	Timer (decrement) 1) Start input (starts with ON) 2) TMR number (00000 to 17777) 3) Setting value (0.1 to 3276.7 sec.)	Start input ON					9
UTMR (BCD)		3	Timer (increment) 1) Start input (starts with ON) 2) TMR number (00000 to 17777) 3) Setting value (0.1 to 799.9 sec.)	Start input ON					9
UTMR (BIN)		3	Timer (increment) 1) Start input (starts with ON) 2) TMR number (00000 to 17777) 3) Setting value (0.1 to 3276.7 sec.)	Start input ON					9
CNT		2	Counter (decrement) 1) Counter input 3) CNT number (00000 to 17777) 2) Reset input 4) Setting value (1 to 7999)	Counter input					12
DCNT (BCD)		3	Counter (decrement) 1) Counter input 3) CNT number (00000 to 17777) 2)Reset input 4)Setting value (1 to 7999)	Counter input					12
DCNT (BIN)		3	Counter (decrement) 1) Counter input 3) CNT number (00000 to 17777) 2) Reset input 4) Setting value (1 to 32767)	Counter input					12
UCNT (BCD)		3	Counter (increment) 1) Counter input 3) CNT number (00000 to 17777) 2) Reset input 4) Setting value (1 to 7999)	Counter input					12
UCNT (BIN)		3	Counter (increment) 1) Counter input 3) CNT number (00000 to 17777) 2) Reset input 4) Setting value (1 to 32767)	Counter input					12
MD		2	Maintenance display 1), 2), 3) input information 6) MD number 4) Output direct terminal (00000 to 17777) 5) Expansion output 7) MD data (000 to 999)	Output display terminal ON					16

The JW300 has the following additional instructions (compared with the JW30H conventional model) related to differentiation, output instructions, and bit operation.

Mnemonics	Symbol	Words	Function	Execution condition	Flag				See page
					Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
STR POS		1	"a" contact point load rising edge						8-17
STR NEG		1	"a" contact point load falling edge						18
AND POS		1	"a" contact AND rising edge						19
AND NEG		1	"a" contact AND falling edge						20
OR POS		1	"a" contact OR rising edge						21
OR NEG		1	"a" contact OR falling edge						22
OUT POS		1	Rising coil						23
OUT NEG		1	Falling coil						24
OUT NOT		1	Inverse output input conditions						25
SET		1	Set at rising input signal						26
RST		1	Reset at rising input signal						27
PUSH		1	Makes a protected area in the internal memory for the contents of the accumulator and stack.						29
POP		1	Recovers the contents of the accumulator and stack from the internal memory area.						
MRD		1	Temporarily reads the contents of accumulator and stack in the internal memory area.						

7-2 List of application instructions

[1] In order of numbers

Mne-monics	Symbol	Words	Function	Execution condition	Flag				See page
					Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
F-00	F-00 XFER S D	3	Transfers data register to data register (1 byte)	↑					10-1
F-00w	F-00w XFER S D	3	Transfers data register to data register (1 word)	↑					1
F-00d	F-00d XFER S D	3	Transfers data register to data register (2 words)	↑					2
F-01	F-01 BCD n D	3	Transfers BCD constant (2 digits)	↑					3
F-01w	F-01w BCD n D	3	Transfers BCD constant (4 digits)	↑					3
F-01d	F-01d BCD n D	3	Transfers BCD constant (8 digits)	↑					4
F-02	F-02 XCHG D_1 D_2	3	Exchanges registers (1 byte)	↑					5
F-02w	F-02w XCHG D_1 D_2	3	Exchanges registers (1 word)	↑					5
F-02d	F-02d XCHG D_1 D_2	3	Exchanges registers (2 words)	↑					6
F-03	F-03 →BIN S D	3	Converts 2-digit BCD to 8-bit binary	↑	0	0	↕	0	7
F-03w	F-03w →BIN S D	3	Converts 4-digit BCD to 16-bit binary	↑	0	0	↕	0	7
F-03d	F-03d →BIN S D	3	Converts 8-digit BCD to 32-bit binary	↑	0	0	↕	0	8
F-04	F-04 →BCD S D	3	Converts 8-bit binary to 2-digit BCD	↑					9
F-04w	F-04w −BCD S D	3	Converts 16-bit binary to 6-digit BCD	↑					9
F-04d	F-04d −BCD S D	3	Converts 32-bit binary to 12-digit BCD	↑					10
F-05	F-05 DMPX S D	3	Demultiplexes 1-byte	↑					11
F-05w	F-05w DMPX S D	3	Demultiplexes 1-word	↑					12
F-05d	F-05d DMPX S D	3	Demultiplexes 2-word	↑					13
F-06	F-06 MPX S D	3	Multiplexes 1-byte	↑					14
F-06w	F-06w MPX S D	3	Multiplexes 1-word	↑					15
F-06d	F-06d MPX S D	3	Multiplexes 2-word	↑					16
F-07	F-07 DCML n D	3	Transfers 1-byte decimal constant	↑					17
F-07w	F-07w DCML n D	3	Transfers 1-word decimal constant	↑					17
F-07d	F-07d DCML n D	3	Transfers 2-word decimal constant	↑					18
F-08	F-08 OCT n D	3	Transfers 1-byte octal constant	↑					19
F-08w	F-08w OCT n D	3	Transfers 1-word octal constant	↑					19
F-08d	F-08d OCT n D	3	Transfers 2-word octal constant	↑					20
F-09	F-09 INV S D	3	Complements 8-bit data	↑					21
F-09w	F-09w INV S D	3	Complements 16-bit data	↑					21
F-09d	F-09d INV S D	3	Complements 32-bit data	↑					22

Mnemonics	Symbol	Words	Function	Execution condition	Flag				See page
					Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
F-10	F-10 ADD S1 S2 D	4	Adds register and register (BCD 2 digits)	↑	↕	↕	↕	↕	10-23
F-10w	F-10w ADD S1 S2 D	4	Adds register and register (BCD 4 digits)	↑	↕	↕	↕	↕	24
F-10d	F-10d ADD S1 S2 D	4	Adds register and register (BCD 8 digits)	↑	↕	↕	↕	↕	25
Fc10	Fc10 ADD S1 n D	4	Adds register (BCD 2 digits) and constant (2 digits)	↑	↕	↕	↕	↕	26
Fc10w	Fc10w ADD S1 n D	4	Adds register (BCD 4 digits) and constant (4 digits)	↑	↕	↕	↕	↕	27
Fc10d	Fc10d ADD S1 n D	4	Adds register (BCD 8 digits) and constant (4 digits)	↑	↕	↕	↕	↕	28
F-11	F-11 SUB S1 S2 D	4	Subtracts register from register (BCD 2 digits)	↑	↕	↕	↕	↕	29
F-11w	F-11w SUB S1 S2 D	4	Subtracts register from register (BCD 4 digits)	↑	↕	↕	↕	↕	30
F-11d	F-11d SUB S1 S2 D	4	Subtracts register from register (BCD 8 digits)	↑	↕	↕	↕	↕	31
Fc11	Fc11 SUB S1 n D	3	Subtracts constant (BCD 2 digits) from register (2 digits)	↑	↕	↕	↕	↕	32
Fc11w	Fc11w SUB S1 n D	3	Subtracts constant (BCD 4 digits) from register (4 digits)	↑	↕	↕	↕	↕	33
Fc11d	Fc11d SUB S1 n D	3	Subtracts constant (BCD 8 digits) from register (4 digits)	↑	↕	↕	↕	↕	34
F-12	F-12 CMP S1 S2	3	Compares register with register (1 byte)	ON	↕	↕	0	↕	35
F-12w	F-12w CMP S1 S2	3	Compares register with register (1 word)	ON	↕	↕	0	↕	36
F-12d	F-12d CMP S1 S2	3	Compares register with register (2 words)	ON	↕	↕	0	↕	37
Fc12	Fc12 CMP S1 n	3	Compares register with octal constant (1 byte)	ON	↕	↕	0	↕	38
Fc12w	Fc12w CMP S1 n	3	Compares register with octal constant (1 word)	ON	↕	↕	0	↕	38
Fc12d	Fc12d CMP S1 n	3	Compares register with octal constant (2 words)	ON	↕	↕	0	↕	39
Fx12	Fx12 CMP S1 n	3	Compares register with hexadecimal constant (1 byte)	ON	↕	↕	0	↕	40
Fx12w	Fx12w CMP S1 n	3	Compares register with hexadecimal constant (1 word)	ON	↓	↓	0	↓	40
Fx12d	Fx12d CMP S1 n	3	Compares register with hexadecimal constant (2 words)	ON	↑	↑	0	↑	41
F-13	F-13 AND S D	3	ANDs register with register (1 byte)	↑					42
F-13w	F-13w AND S D	3	ANDs register with register (1 word)	↑					42
F-13d	F-13d AND S D	3	ANDs register with register (2 words)	↑					43
Fc13	Fc13 AND n D	3	ANDs register with octal constant (1 byte)	↑					44
Fc13w	Fc13w AND n D	3	ANDs register with register (1 word)	↑					44
Fc13d	Fc13d AND n D	3	ANDs register with octal constant (2 words)	↑					45
Fx13	Fx13 AND n D	3	ANDs register with hexadecimal constant (1 byte)	↑					46
Fx13w	Fx13w AND n D	3	ANDs register with hexadecimal constant (1 word)	↑					46
Fx13d	Fx13d AND n D	3	ANDs register with hexadecimal constant (2 words)	↑					47

Mnemonics	Symbol	Words	Function	Execution condition	Flag				See page
					Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
F-14	F-14 OR S D	3	ORs register with register (1 byte)	↑					10-48
F-14w	F-14w OR S D	3	ORs register with register (1 word)	↑					48
F-14d	F-14d OR S D	3	ORs register with register (2 words)	↑					49
Fc14	Fc14 OR n D	3	ORs register with octal constant (1 byte)	↑					50
Fc14w	Fc14w OR n D	3	ORs register with octal constant (1 word)	↑					50
Fc14d	Fc14d OR n D	3	ORs register with octal constant (2 words)	↑					51
Fx14	Fx14 OR n D	3	ORs register with hexadecimal constant (1 byte)	↑					52
Fx14w	Fx14w OR n D	3	ORs register with hexadecimal constant (1 word)	↑					52
Fx14d	Fx14d OR n D	3	ORs register with hexadecimal constant (2 words)	↑					53
F-15	F-15 MUL S_1 S_2 D	4	Multiplies register by register (BCD 4 digits)	↑	0	0	↕	0	11-1
F-15d	F-15d MUL S_1 S_2 D	4	Multiplies register by register (BCD 8 digits)	↑	0	0	↕	0	2
Fc15	Fc15 MUL S_1 n D	4	Multiplies register (BCD 4 digits) by constant (BCD 3 digits)	↑	0	0	↕	0	3
Fc15d	Fc15d MUL S_1 n D	4	Multiplies register (BCD 8 digits) by constant (BCD 4 digits)	↑	0	0	↕	0	4
F-16	F-16 DIV S_1 S_2 D	4	Divides register (BCD 4 digits) by register (BCD 2 digits)	↑	0	0	↕	0	5
F-16d	F-16d DIV S_1 S_2 D	4	Divides register (BCD 8 digits) by register (BCD 8 digits)	↑	0	0	↕	0	6
Fc16	Fc16 DIV S_1 n D	4	Divides register (BCD 4 digits) by constant (BCD 2 digits)	↑	0	0	↕	0	7
Fc16d	Fc16d DIV S_1 n D	4	Divides register (BCD 8 digits) by constant (BCD 4 digits)	↑	0	0	↕	0	8
F-17	F-17 XNR S D	3	Exclusive NORs register with register (1 byte)	↑					9
F-17w	F-17w XNR S D	3	Exclusive NORs register with register (1 word)	↑					9
F-17d	F-17d XNR S D	3	Exclusive NORs register with register (2 words)	↑					10
Fc17	Fc17 XNR n D	3	Exclusive NORs register with octal constant (1 byte)	↑					11
Fc17w	Fc17w XNR n D	3	Exclusive NORs register with octal constant (1 word)	↑					11
Fc17d	Fc17d XNR n D	3	Exclusive NORs register with octal constant (2 words)	↑					12
Fx17	Fx17 XNR n D	3	Exclusive NORs register with hexadecimal constant (1 byte)	↑					13
Fx17w	Fx17w XNR n D	3	Exclusive NORs register with hexadecimal constant (1 word)	↑					13
Fx17d	Fx17d XNR n D	3	Exclusive NORs register with hexadecimal constant (2 words)	↑					14
F-18	F-18 XOR S D	3	Exclusive ORs register with register (1 byte)	↑					15
F-18w	F-18w XOR S D	3	Exclusive ORs register with register (1 word)	↑					15
F-18d	F-18d XOR S D	3	Exclusive ORs register with register (2 words)	↑					16
Fc18	Fc18 XOR n D	3	Exclusive ORs register with octal constant (1 byte)	↑					17
Fc18w	Fc18w XOR n D	3	Exclusive ORs register with octal constant (1 word)	↑					17
Fc18d	Fc18d XOR n D	3	Exclusive ORs register with octal constant (2 words)	↑					18

Mnemonics	Symbol	Words	Function	Execution condition	Flag Zero 007357	Flag Carry 007356	Flag Error 007355	Flag Non-carry 007354	See page
Fx18	Fx18 XOR n D	3	Exclusive ORs register with hexadecimal constant(1 byte)	↑					11-19
Fx18w	Fx18w XOR n D	3	Exclusive ORs register with hexadecimal constant(1 word)	↑					19
Fx18d	Fx18d XOR n D	3	Exclusive ORs register with hexadecimal constant(2 words)	↑					20
F-20	1) 2) 3) 4) MD (F-20) 6) 7) 5)	2	Maintenance display 1), 2), 3) Input information 4) Output direct terminal 5) Expansion output 6) MD number (000 to 777) 7) MD data (000 to 999)	Output display terminal ON					21
F-21	F-21 SQRT S D	3	Obtains square root of register (BCD 8 digits)	↑	0	0	↕	0	22
F-22	F-22 SIN S D	3	Executes trigonometric function (SIN)	↑	0	↕	↕	↕	22
F-23	F-23 COS S D	3	Executes trigonometric function (COS)	↑	0	↕	↕	↕	23
F-24	F-24 TAN S D	3	Executes trigonometric function (TAN)	↑	0	↕	↕	↕	24
F-25	F-25 ASIN S D	3	Executes trigonometric function (SIN^{-1})	↑	0	↕	↕	↕	25
F-26	F-26 ACOS S D	3	Executes trigonometric function (COS^{-1})	↑	0	↕	↕	↕	26
F-27	F-27 ATAN S D	3	Executes trigonometric function (TAN^{-1})	↑	0	↕	↕	↕	27
F-28	F-28 XY→ S D	3	Exchanges the rectangular coordinate system (X,Y) data with polar coordinate system (γ, θ)	↑	0	0	↕	0	28
F-29	F-29 →XY S D	3	Exchanges the polar coordinate system (γ, θ) data with rectangular coordinate system(X, Y)	↑	0	0	↕	0	29
F-30	F-30 MCS	1	Sets master control	ON					30
F-31	F-31 MCR	1	Resets master control						30
F-32	F-32 SET OUT	2	Sets coil	ON					33
F-33	F-33 RST OUT	2	Resets coil	ON					34
F-34	F-34 TSET n_1 n_2 BIT	4	Compares with current value of clock (specified relay set)	ON					36
F-35	F-35 TRST n_1 n_2 BIT	4	Compares with current value of clock (specified relay reset)	ON					37
F-36	F-36 TADD S_1 S_2 D	4	Adds time	↑	↑	↑	↑	↑	38
F-37	F-37 TSUB S_1 S_2 D	4	Subtracts time	↑	↕	↕	↕	↕	39
F-38	F-38 TXFR D	2	Transfers current value of clock	↑					40
F-40	F-40 END	1	END instruction						41
F-41	F-41 JCS	1	Sets jump control	OFF					42
F-42	F-42 JCR	1	Resets jump control						42
F-43	F-43	1	Complements bit (ACC contents)						44
F-44	F-44	1	Differentiates at ON	↑					45
F-45	F-45	1	Differentiates at OFF	↓					46
F-47	F-47 ONLS	1	Sets level operating condition						47
F-48	F-48 ONLR	1	Resets level operating condition						47
F-49	F-49 ENDC	1	Conditional END	OFF					48

Mnemonics	Symbol	Words	Function	Execution condition	Flag: Zero 007357	Flag: Carry 007356	Flag: Error 007355	Flag: Non-carry 007354	See page
F-50	F-50 4→16 S D	3	Decodes from 4 to 16	↑					12-1
F-51	F-51 16→4 S D	3	Encodes from 16 to 4	↑					1
F-52	F-52 7SEG S D	3	Decodes to 7-segment data	↑					2
F-53	F-53 →BIN S D	3	Converts 4-digit BCD to 16-bit binary	↑	0	0	↕	0	3
F-54	F-54 →BCD S D	3	Converts 16-bit binary to 6-digit BCD	↑					3
F-55	F-55 SWAP S D	3	Swaps upper 4 bits with lower 4 bits	↑					4
F-56	F-56 NEG S D	3	Complement of 10 of 1-byte data	↑	0	0	↕	0	5
F-56w	F-56w NEG S D	3	Complement of 10 of 1-word data	↑	0	0	↕	0	5
F-56d	F-56d NEG S D	3	Complement of 10 of 2-word data	↑	0	0	↑↓	0	6
F-57	F-57 2NEG S D	3	Complement of 2 of 1-byte data	↑					7
F-57w	F-57w 2NEG S D	3	Complement of 2 of 1-word data	↑					7
F-57d	F-57d 2NEG S D	3	Complement of 2 of 2-word data	↑					8
F-58	F-58 Σ BIT n S D	4	Total of ON bits	↑					9
F-60	1) 2) 3) 4) F-60 SFR D	2	Shifts register bi-directionally (1 byte) 1) Shift direction input 3) Shift input 2) Data input 4) Reset input	Shift input ↑	↕	↕	0	↕	10
F-60w	1) 2) 3) 4) F-60w SFR D	2	Shifts register bi-directionally (1 word) 1) Shift direction input 3) Shift input 2) Data input 4) Reset input	Shift input ↑	↕	↕	0	↕	12
F-60d	1) 2) 3) 4) F-60d SFR D	2	Shifts register bi-directionally (2 words) 1) Shift direction input 3) Shift input 2)Data input 4) Reset input	Shift input ↑	↕	↕	0	↕	13
F-61	1) 2) F-61 ASFR D	2	Shifts register asynchronously (1 byte) 1) Shift direction input 2) Shift input	Shift input ON	0	↕	0	↕	14
F-61w	1) 2) F-61w ASFR D	2	Shifts register asynchronously (1 word) 1) Shift direction input 2) Shift input	Shift input ON	0	↕	0	↕	15
F-61d	1) 2) F-61d ASFR D	2	Shifts register asynchronously (2 words) 1) Shift direction input 2)Shift input	Shift input ON	0	↕	0	↕	16
F-62	1) 2) 3) F-62 U/DC D	2	2-digit BCD up/down counter 1) Up/down counter direction input 2) Counter input 3) Reset input	Counter input ↑	↕	↕	↕	↕	17
F-62w	1) 2) 3) F-62w U/DC D	2	4-digit BCD up/down counter 1) Up/down counter direction input 2) Counter input 3) Reset input	Counter input ↑	↕	↕	↕	↕	18
F-62d	1) 2) 3) F-62d U/DC D	2	8-digit BCD up/down counter 1) Up/down counter direction input 2) Counter input 3) Reset input	Counter input ↑	↕	↕	↕	↕	19
F-63	F-63 INC D	2	Increments binary counter (1 byte)	↑	↕	↕	0	↕	20
F-63w	F-63w INC D	2	Increments binary counter (1 word)	↑	↕	↕	0	↕	20
F-63d	F-63d INC D	2	Increments binary counter (2 words)	↑	↕	↕	0	↕	21
F-64	F-64 DEC D	2	Decrements binary counter (1 byte)	↑	↕	↕	0	↕	22
F-64w	F-64w DEC D	2	Decrements binary counter (1 word)	↑	↕	↕	0	↕	22
F-64d	F-64d DEC D	2	Decrements binary counter (2 words)	↑	↕	↕	0	↕	23

Mnemonics	Symbol	Words	Function	Execution condition	Flag				See page
					Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
F-65	F-65 BCDI D	2	BCD increment counter (1 byte)	↑	↕	↕	↕	↕	12-24
F-65w	F-65w BCDI D	2	BCD increment counter (1 word)	↑	↕	↕	↕	↕	24
F-65d	F-65d BCDI D	2	BCD increment counter (2 words)	↑	↕	↕	↕	↕	25
F-66	F-66 BCDD D	2	BCD decrement counter (1 byte)	↑	↕	↕	↕	↕	26
F-66w	F-66w BCDD D	3	BCD decrement counter (1 word)	↑	↕	↕	↕	↕	26
F-66d	F-66d BCDD D	3	BCD decrement counter (2 words)	↑	↕	↕	↕	↕	27
F-67	F-67 NSFH n D	3	Digit shift (up)	↑					28
F-68	F-68 NSFL n D	4	Digit shift (down)	↑					28
F-69	F-69 NXFR S D	4	Digit transfer	↑					29
F-70	F-70 FILE n S D	4	Transfers "n" bytes in block	↑					30
F-70w	F-70w FILE n S D	4	Transfers "n" words in block	↑					31
F-70d	F-70d FILE n S D	4	Transfers "n" double word in block	↑					32
F-71	F-71 CONS n D_1 D_2	4	Transfers octal constant in block (1 byte)	↑					33
F-71w	F-71w CONS n D_1 D_2	4	Transfers octal constant in block (1 word)	↑					34
F-71d	F-71d CONS n D_1 D_2	4	Transfers octal constant block (2 words)	↑					34
F-72	F-72 DMPX n S D	4	Demultiplexes "n" bytes to file 1 register	↑					35
F-72w	F-72w DMPX n S D	4	Demultiplexes "n" words to file 1 register	↑					36
F-72d	F-72d DMPX n S D	4	Demultiplexes "n" double word to file 1 register	↑					37
F-73	F-73 MPX n S D	4	Multiplexes "n" bytes from file 1 register	↑					38
F-73w	F-73w MPX n S D	4	Multiplexes "n" words from file 1 register	↑					39
F-73d	F-73d MPX n S D	4	Multiplexes "n" double word from file 1 register	↑					40
F-74	F-74 nXFR n S D	4	Transfers "n" bytes	↑					41
F-74w	F-74w nXFR n S D	4	Transfers "n" words	↑					41
F-74d	F-74d nXFR n S D	4	Transfers "n" double word	↑					42
F-76	F-76 FILR S_1 S_2 D	4	Transfers "n" bytes in block	↑					43
F-76w	F-76w FILR S_1 S_2 D	4	Transfers "n" words in block	↑					44
F-76d	F-76d FILR S_1 S_2 D	4	Transfers "n" double word in block	↑					45
F-77	F-77 CHKC S_1 S_2 D	4	Creates sum check code	↑					46
F-78	F-78 CHK S_1 S_2 S_3	4	Checks data	↑	0	0	↕	0	47
F-79	F-79 SORT S_1 n_1 n_2	4	Sorts register (1 byte) data	↑					48
F-79w	F-79w SORT S_1 n_1 n_2	4	Sorts register (1 word) data	↑					49
F-79d	F-79d SORT S_1 n_1 n_2	4	Sorts register (2 words) data	↑					50

Mne-monics	Symbol	Words	Function	Execution condition	Flag				See page
					Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
F-80	F-80 IORF R,S	2	I/O refresh	ON	0	↕	↕	↕	13-1
F-82	F-82 IORF SW	2	Special I/O refresh	ON	0	↕	↕	↕	2
F-85	F-85 PRRD n_1 SW,n_2 D	4	Read from special I/O	↑	0	↕	↕	↕	3
F-86	F-86 PRWR n_1 D SW,n_2	4	Write to special I/O	↑	0	↕	↕	↕	3
F-90	F-90 REM n	2	Remark n = 0000 to 3777						4
F-91	F-91 BCD8 n_1 n_2 D	4	Transfers BCD constant (8 digits)	↑					5
F-97	F-97 DML8 n_1 n_2 D	4	Transfers a decimal constant (8 digits)	↑					6
F-100	F-100 ADRS S D	3	Sets indirect address	↑					7
F-101	F-101 SEGM n file N D	4	Sets indirect address	↑					8
F-102	F-102 MRD n file N D	4	Reads from the register of a direct address (1 byte)	↑					9
F-102w	F-102w MRD n file N D	4	Reads from the register of a direct address (1 word)	↑					9
F-102d	F-102d MRD n file N D	4	Reads from the register of a direct address (2 words)	↑					10
F-103	F-103 MWR S n file N	4	Writes in the register of a direct address (1 byte)	↑					11
F-103w	F-103w MWR S n file N	4	Writes in the register of a direct address (1 word)	↑					11
F-103d	F-103d MWR S n file N	4	Writes in the register of a direct address (2 words)	↑					12
F-112	F-112 NCMP S_1 S_2 S_3	4	Compares "n" bytes (between 1-byte registers)	ON	↕	↕	0	↕	13
F-112w	F-112w NCMP S_1 S_2 S_3	3	Compares "n" words	ON	↕	↕	0	↕	14
F-112d	F-112d NCMP S_1 S_2 S_3	3	Compares "n" double word	ON	↕	↕	0	↕	15
F-116	F-116 DIV S_1 S_2 D	3	Divides register value (BCD 8 digits) by another register value (BCD 8 digits) (4 decimal places)	↑	0	0	↕	0	16
F-130	F-130 BIT→ S_1 S_2	3	Multiplexes bit (indirect address)	ON	0	↕	0	0	17
F-131	F-131 BIT→ n S	2	Multiplexes bit (direct address)	ON	0	↕	0	0	17
F-132	1) 2) F-132 S/R S D	2	Sets/resets bits (indirect address) 1) Sets/resets direction input 2) Input condition	ON					18
F-133	1) 2) F-133 S/R n D	2	Sets/resets bits (direct address) 1) Sets/resets direction input 2) Input condition	ON					18
F-140	F-140 LABL LBn	1	Sets label LB0000 to LB1377						19
F-141	F-141 JMP LBn	2	Jumps to label	ON					20
F-142	F-142 CALL LBn	1	Calls labeled subroutine	↑					22
F-143	F-143 RET	2	Returns from subroutine						22
F-144	F-144 FOR n	1	Sets loop count	↑					24
F-145	F-145 NEXT	2	END of loop						24
F-146	F-146 FORR S	1	Sets loop count register	↑					26
F-147	F-147 EXIT		Exits loop (conditional)	OFF					27
F-148	F-148 CAL+ LBn S		Calls subroutine by label	↑					28
F-149	F-149 RETC		Retuens from subroutine (conditional)	OFF					29

Mnemonics	Symbol	Words	Function	Execution condition	Flag				See page
					Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
F-151	F-151 JMP+ LBn S	3	Jumps to labeled program address	ON					13-30
F-153	F-153 → BIN S D	3	Converts 8-digit BCD to 32-bit binary	↑	0	0	↕	0	31
F-154	F-154 →BCD S D	3	Converts 32-bit binary to 10-digit BCD	↑					31
F-155	F-155 →SEC S D	3	Convert hours (BCD 4 digits), minutes (BCD 2 digits) and seconds (BCD 2 digits) into seconds (BCD 8 digits)	↑	0	0	↕	0	32
F-156	F-156 →HMS S D	3	Convert seconds (BCD 8 digits) into hours (BCD 4 digits), minutes (BCD 2 digits), and seconds (BCD 2 digits)	↑	0	0	↕	0	33
F-160	1) 2) 3) 4) F-160 NSFR S_1 S_2 D	4	Shifts register bi-directionally (n bit) 1) Shift direction input 3) Shift input 2) Data input 4) Reset input (Shift is 1 bit) (S_1)=0 to 256 (S_2)=0 to 7	Shift input ↑	↕	↕	0	↕	34
Fc160	1) 2) 3) 4) Fc160 NSFR n_1 n_2 D	4	Shift register bi-directionally (n bit) 1) Shift direction input 3) Shift input 2) Data input 4) Reset input (Shift is 1 bit) n_1=0 to 377 n_2=0 to 7	Shift input ↑	↕	↕	0	↕	35
F-161	1) 2) F-161 NASR D n	3	Shifts register asynchronously (n bytes) 1) Shift direction input 2) Shift input	Shift input ON	0	↕	0	↕	36
F-161w	1) 2) F-161w NASR D n	3	Shifts register asynchronously (n words) 1) Shift direction input 2) Shift input	Shift input ON	0	↕	0	↕	38
F-161d	1) 2) F-161d NASR D n	3	Shifts register asynchronously (n double words)	Shift input ON	0	↕	0	↕	39
F-163	F-163 INC2 D	2	Adds binary (+2) counter (1 byte)	↑	↕	↕	0	↕	40
F-163w	F-163w INC2 D	2	Adds binary (+2) counter (1 word)	↑	↕	↕	0	↕	40
F-163d	F-163d INC2 D	2	Adds binary (+2) counter (2 words)	↑	↕	↓	0	↕	41
F-164	F-164 DEC2 D	2	Subtracts binary (-2) counter (1 byte)	↑	↕	↕	0	↕	42
F-164w	F-164w DEC2 D	2	Subtracts binary (-2) counter (1 word)	↑	↕	↕	0	↕	42
F-164d	F-164d DEC2 D	2	Subtracts binary (-2) counter (2 words)	↑	↕	↓	0	↕	43
F-170	F-170 INS S D_1 D_2	4	Inserts data (1 byte)	↑	0	0	↕	0	44
F-170w	F-170w INS S D_1 D_2	4	Inserts data (1 word)	↑	0	0	↕	0	45
F-170d	F-170d INS S D_1 D_2	4	Inserts data (2 words)	↑	0	0	↕	0	46
F-171	F-171 DEL S_1 S_2 S_3	4	Deletes data (1 byte)	↑	0	0	↕	0	47
F-171w	F-171w DEL S_1 S_2 S_3	4	Deletes data (1 word)	↑	0	0	↕	0	48
F-171d	F-171d DEL S_1 S_2 S_3	4	Deletes data (2 words)	↑	0	0	↕	0	49
F-172	F-172 SRCH S D_1 D_2	4	Searches data (1 byte)	↑	0	0	↕	0	50
F-172w	F-172w SRCH S D_1 D_2	4	Searches data (1 word)	↑	↕	↕	0	0	51
F-172d	F-172d SRCH S D_1 D_2	4	Searches data (2 words)	↑	↕	↕	0	0	52
F-173	1) 2) F-173 CHNG S D_1 D_2	4	Changes data (1 byte) 1) Mode direction 2) Execution input	Execution input ↑	↕	↕	0	0	53
F-173w	1) 2) F-173w CHNG S D_1 D_2	4	Changes data (1 word) 1) Mode direction 2) Execution input	Execution input ↑	↕	↕	0	0	54
F-173d	1) 2) F-173d CHNG S D_1 D_2	4	Changes data (2 words)	Execution input ↑	↕	↕	0	0	55
F-174	F-174 VREV D n	3	Reverse order of register data (1 byte)	↑					14-1
F-175	F-175 NSWP D n	3	Swaps upper 4 bits with lower 4 bits of registers	↑					1

Mnemonics	Symbol	Words	Function	Execution condition	Flag				See page
					Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
F-176	F-176 DFRD S file N D	4	Reads data from register of specified address (256 bytes)	↑					14-2
F-177	F-177 DFWR S D file N	4	Writes data into register of specified address (256 bytes)	↑					3
F-180	F-180 CP> S_1 S_2 BIT	4	Compare between register and register (1 byte)(>, with relay output)	ON	0	0	0	0	4
F-180w	F-180w CP> S_1 S_2 BIT	4	Compare between register and register (1 word)(>, with relay output)	ON	0	0	0	0	5
F-180d	F-180d CP> S_1 S_2 BIT	4	Compare between register and register (2 words)(>, with relay output)	ON	0	0	0	0	6
Fc180	Fc180 CP> S n BIT	4	Compare register with constant (1 byte)(>, with relay output)	ON	0	0	0	0	7
Fc180w	Fc180w CP> S n BIT	4	Compare register with constant (1 word)(>, with relay output)	ON	0	0	0	0	8
Fc180d	Fc180d CP> S n BIT	4	Compare register with constant (2 words)(>, with relay output)	ON	0	0	0	0	9
F-181	F-181 CP< S_1 S_2 BIT	4	Compare between register and register (1 byte)(<, with relay output)	ON	0	0	0	0	4
F-181w	F-181w CP< S_1 S_2 BIT	4	Compare between register and register (1 word)(<, with relay output)	ON	0	0	0	0	5
F-181d	F-181d CP< S_1 S_2 BIT	4	Compare between register and register (2 words)(<, with relay output)	ON	0	0	0	0	6
Fc181	Fc181 CP< S n BIT	4	Compare register with constant (1 byte)(<, with relay output)	ON	0	0	0	0	7
Fc181w	Fc181w CP< S n BIT	4	Compare register with constant (1 word)(<, with relay output)	ON	0	0	0	0	8
Fc181d	Fc181d CP< S n BIT	4	Compare register with constant (2 words)(<, with relay output)	ON	0	0	0	0	9
F-182	F-182 CP= S_1 S_2 BIT	4	Compare between register and register (1 byte)(<, with relay output)	ON	0	0	0	0	4
F-182w	F-182w CP= S_1 S_2 BIT	4	Compare between register and register (1 word)(<, with relay output)	ON	0	0	0	0	5
F-182d	F-182d CP= S_1 S_2 BIT	4	Compare between register and register (2 words)(<, with relay output)	ON	0	0	0	0	6
Fc182	Fc182 CP= S n BIT	4	Compare register with constant (1 byte)(<, with relay output)	ON	0	0	0	0	7
Fc182w	Fc182w CP= S n BIT	4	Compare register with constant (1 word)(<, with relay output)	ON	0	0	0	0	8
Fc182d	Fc182d CP= S n BIT	4	Compare register with constant (2 words)(<, with relay output)	ON	0	0	0	0	9
F-183	F-183 CP>= S_1 S_2 BIT	4	Compare between register and register (1 byte)(=, with relay output)	ON	0	0	0	0	4
F-183w	F-183w CP>= S_1 S_2 BIT	4	Compare between register and register (1 word)(=, with relay output)	ON	0	0	0	0	5
F-183d	F-183d CP>= S_1 S_2 BIT	4	Compare between register and register (2 words)(=, with relay output)	ON	0	0	0	0	6
Fc183	Fc183 CP>= S n BIT	4	Compare register with constant (1 byte)(=, with relay output)	ON	0	0	0	0	7
Fc183w	Fc183w CP>= S n BIT	4	Compare register with constant (1 word)(=, with relay output)	ON	0	0	0	0	8
Fc183d	Fc183d CP>= S n BIT	4	Compare register with constant (2 words)(=, with relay output)	ON	0	0	0	0	9
F-184	F-184 CP<= S_1 S_2 BIT	4	Compare between register and register (1 byte)(≧, with relay output)	ON	0	0	0	0	4
F-184w	F-184w CP<= S_1 S_2 BIT	4	Compare between register and register (1 word)(≧, with relay output)	ON	0	0	0	0	5
F-184d	F-184d CP<= S_1 S_2 BIT	4	Compare between register and register (2 words)(≧, with relay output)	ON	0	0	0	0	6
Fc184	Fc184 CP<= S n BIT	4	Compare register with constant (1 byte)(≧, with relay output)	ON	0	0	0	0	7
Fc184w	Fc184w CP<= S n BIT	4	Compare register with constant (1 word)(≧, with relay output)	ON	0	0	0	0	8
Fc184d	Fc184d CP<= S n BIT	4	Compare register with constant (2 words)(≧, with relay output)	ON	0	0	0	0	9

Mnemonics	Symbol	Words	Function	Execution condition	Flag				See page
					Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
F-185	F-185 CP<> S1 S2 BIT	4	Compare between register and register (1 byte)(≦, with relay output)	ON	0	0	0	0	14-4
F-185w	F-185w CP<> S1 S2 BIT	4	Compare between register and register (1 word)(≦, with relay output)	ON	0	0	0	0	5
F-185d	F-185d CP<> S1 S2 BIT	4	Compare between register and register (2 words)(≦, with relay output)	ON	0	0	0	0	6
Fc185	Fc185 CP<> S n BIT	4	Compare register with constant (1 byte)(≦, with relay output)	ON	0	0	0	0	7
Fc185w	Fc185w CP<> S n BIT	4	Compare register with constant (1 word)(≦, with relay output)	ON	0	0	0	0	8
Fc185d	Fc185d CP<> S n BIT	4	Compare register with constant (2 words)(≦, with relay output)	ON	0	0	0	0	9
F-202	F-202 OPCH UN,CH, ST file N n	4	Open channel (with octal station number)	ON					10
F-203	F-203 OPCH UN,CH, ST file N n	4	Open channel (with hex. station number)	ON					10
F-204	F-204 SEND n S	3	Sends data	↑	↕	↕	↕	↕	11
F-205	F-205 RCV n D	3	Receives data	↑	↕	↕	↕	↕	12
F-206	F-206 EOP1 UN1, CH ST1 UN2	4	Open channel 1 (set the hierarchical communication)	ON					13
F-207	F-207 EOP2 ST2 file N n	4	Open channel 2 (set the hierarchical communication)	ON					13
F-210	F-210 ADD S1 S2 D	4	Adds register and register in binary (8 bits + 8 bits)	↑	↕	↕	0	↕	14
F-210w	F-210w ADD S1 S2 D	4	Adds register and register in binary (16 bits + 16 bits)	↑	↕	↕	0	↕	15
F-210d	F-210d ADD S1 S2 D	4	Adds register and register in binary (32 bits + 32 bits)	↑	↕	↕	0	↕	16
Fc210	Fc210 ADD S1 n D	4	Adds register and constant in binary (8 bits + 8 bits)	↑	↕	↕	0	↕	17
Fc210w	Fc210w ADD S1 n D	4	Adds register and constant in binary (16 bits + 16 bits)	↑	↕	↕	0	↕	17
Fc210d	Fc210d ADD S1 n D	4	Adds register and constant in binary (32 bits + 16 bits)	↑	↕	↕	0	↕	18
F-211	F-211 SUB S1 S2 D	4	Subtracts register from register in binary (8 bits - 8 bits)	↑	↕	↕	0	↕	19
F-211w	F-211w SUB S1 S2 D	4	Subtracts register from register in binary (16 bits - 16 bits)	↑	↕	↕	0	↕	20
F-211d	F-211d SUB S1 S2 D	4	Subtracts register from register in binary (32 bits - 32 bits)	↑	↕	↕	0	↕	20
Fc211	Fc211 SUB S1 n D	4	Subtracts constant from register in binary (8 bits - 8 bits)	↑	↕	↕	0	↕	21
Fc211w	Fc211w SUB S1 n D	4	Subtracts constant from register in binary (16 bits - 16 bits)	↑	↕	↕	0	↕	21
Fc211d	Fc211d SUB S1 n D	4	Subtracts constant from register in binary (32 bits - 16 bits)	↑	↕	↕	0	↕	22
F-212	F-212 WNDW S1 S2 S3	4	Window comparator (between 1-byte registers)	ON	↕	↕	↕	↕	23
F-212w	F-212w WNDW S1 S2 S3	4	Window comparator (between 1-word registers)	ON	↕	↕	↕	↕	24
F-212d	F-212d WNDW S1 S2 S3	4	Window comparator (between 2-word registers)	ON	↕	↕	↕	↕	25
Fc212	Fc212 WNDW S1 n1 n2	4	Window comparator (between 1-byte octal constants)	ON	↕	↕	↕	↕	26
Fc212w	Fc212w WNDW S1 n1 n2	4	Window comparator (between 1-word octal constants)	ON	↕	↕	↕	↕	26
Fc212d	Fc212d WNDW S1 n1 n2	4	Window comparator (between 2-word octal constants)	ON	↕	↕	↕	↕	27
Fx212	Fx212 WNDW S1 n1 n2	4	Window comparator (between 1-byte hexadecimal constants)	ON	↕	↕	↕	↕	28
Fx212w	Fx212w WNDW S1 n1 n2	4	Window comparator (between 1-word hexadecimal constants)	ON	↕	↕	↕	↕	28
Fx212d	Fx212d WNDW S1 n1 n2	4	Window comparator (between 2-word hexadecimal constants)	ON	↕	↕	↕	↕	29

Mnemonics	Symbol	Words	Function	Execution condition	Flag				See page
					Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
F-215	F-215 MUL S1 S2 D	4	Multiplies register by register in binary (8 bits x 8 bits)	↑	0	0	0	0	14-30
F-215w	F-215w MUL S1 S2 D	4	Multiplies register by register in binary (16 bits x 16 bits)	↑	0	0	0	0	30
F-215d	F-215d MUL S1 S2 D	4	Multiplies register by register in binary (32 bits x 32 bits)	↑	0	0	0	0	31
Fc215	Fc215 MUL S1 n D	4	Multiplies register by constant in binary (8 bits x 8 bits)	↑	0	0	0	0	32
Fc215w	Fc215w MUL S1 n D	1	Multiplies register by constant in binary (16 bits x 16 bits)	↑	0	0	0	0	32
Fc215d	Fc215d MUL S1 n D	1	Multiplies register by constant in binary (32 bits x 16 bits)	↑	0	0	0	0	33
F-216	F-216 DIV S1 S2 D	4	Divides register by register in binary (8 bits ÷ 8 bits)	↑	0	0	↕	0	34
F-216w	F-216w DIV S1 S2 D	4	Divides register by register in binary (15 bits ÷ 15 bits)	↑	0	0	↕	0	35
F-216d	F-216d DIV S1 S2 D	4	Divides register by register in binary (31 bits ÷ 31 bits)	↑	0	0	↕	0	36
Fc216	Fc216 DIV S1 n D	4	Divides register by constant in binary (8 bits ÷ 8 bits)	↑	0	0	↕	0	37
Fc216w	Fc216w DIV S1 n D	4	Divides register by constant in binary (15 bits ÷ 15 bits)	↑	0	0	↕	0	38
Fc216d	Fc216d DIV S1 n D	1	Divides register by constant in binary (31 bits ÷ 15 bits)	↑	0	0	↕	0	38
F-231	F-231 MCRN	1	Master control reset nesting						39
F-242	F-242 JCRN	4	Jump control reset nesting						40
F-252	F-252 →ASC S n D	4	Converts HEX code into ASCII code	↑					41
F-253	F-253 →HEX S n D	4	Converts ASCII code into HEX code	↑	0	0	↕	0	42
F-260	F-260 RTMR S D BIT	4	Subtracts timer (setting value, register address)	ON	0	0	↕	0	43
Fc260	Fc260 RTMR n D BIT	4	Subtracts timer (constant, register address)	ON	0	0	↕	0	44
F-261	1) 2) F-261 RCNT S D BIT	4	Subtracts counter (setting value, register address) 1) Counter input 2) Reset input	Counter input ↑	0	0	↕	0	45
Fc261	1) 2) Fc261 RCNT n D BIT	4	Subtracts counter (constant, register address) 1) Counter input 2)Reset input	Counter input ↑	0	0	↕	0	46
F-263	F-263 INC4 D	2	Increments counter by 4 (1-byte binary)	↑	↕	↕	0	↕	47
F-263w	F-263w INC4 D	2	Increments counter by 4 (1-word binary)	↑	↕	↕	0	↕	47
F-263d	F-263d INC4 D	2	Increments counter by 4 (2-word binary)	↑	↕	↕	0	↕	48
F-264	F-264 DEC4 D	2	Decrements counter by 4 (1-byte binary)	↑	↕	↕	0	↕	49
F-264w	F-264w DEC4 D	2	Decrements counter by 4 (1-word binary)	↑	↕	↕	0	↕	49
F-264d	F-264d DEC4 D	2	Decrements counter by 4 (2-word binary)	↑	↕	↕	0	↕	50

Mnemonics	Symbol	Words	Function	Execution condition	Flag				See page
					Zero 07357	Carry 07356	Error 07355	Non-carry 07354	
F-300	F-310 XFER S D	3	Transfers 1-byte data	ON					14-51
F-300w	F-300w XFER S D	3	Transfers 1-word data	ON					51
F-300d	F-300d XFER S D	3	Transfers 2-word data	ON					52
F-310	F-310 SADD S_1 S_2 D	4	Adds registers in binary with sign (31 bits + 31 bits)	↑	↕	↕	↕	↕	53
F-311	F-311 SSUB S_1 S_2 D	4	Subtracts registers in binary with sign (31 bits - 31 bits)	↑	↕	↕	↕	↕	54
F-315	F-315 SMUL S_1 S_2 D	4	Multiplies registers in binary with sign (31 bits x 31 bits)	↑	0	0	0	0	55
F-316	F-316 SDIV S_1 S_2 D	4	Divides registers in binary with sign (31 bits ÷ 31 bits)	↑	0	0	↕	0	56
F-403	F-403 LOG	1	Logging instruction						56
NOP		1	Non-operation instruction						——

[2] Classification by operation

Type			Mnemonics	See page
Transfer instructions	Register to register transfer	1 byte	F-00	10-1
			F-300	14-51
		1 word	F-00w	10-1
			F-300w	14-51
		2 words	F-00d	10-2
			F-300d	14-52
		n bytes	F-70	12-30
		n words	F-70w	31
		n double word	F-70d	32
		n bytes (Indirect assignment)	F-76	43
		n words (Indirect assignment)	F-76w	44
		n double word (Indirect assignment)	F-76d	45
		n bytes (Same data)	F-74	41
		n words (Same data)	F-74w	41
		n double word (Same data)	F-74d	42
	BCD constant transfer	2 digits	F-01	10-3
		4 digits	F-01w	3
		8 digits	F-01d	4
			F-91	13-5
	Decimal constant transfer	1 byte	F-07	10-17
		1 word	F-07w	17
		2 words	F-07d	18
		8 digits	F-97	13-6
	Octal constant transfer	1 byte	F-08	10-19
		1 word	F-08w	19
		2 words	F-08d	20
		n bytes	F-71	12-33
		n words	F-71w	34
		n double word	F-71d	34
	Demultiplex	1 byte	F-05	10-11
		1 word	F-05w	12
		2 words	F-05d	13
		n bytes (file 1)	F-72	12-35
		n words (file 1)	F-72w	36
		n double word (file 1)	F-72d	37
	Multiplex	1 byte	F-06	10-14
		1 word	F-06w	15
		2 words	F-06d	16
		n bytes (file 1)	F-73	12-38
		n words (file 1)	F-73w	39
		n double word (file 1)	F-73d	40
	Digit transfer	4 bits	F-69	29
	Read out the file	1 byte	F-102	13-9
		1 word	F-102w	9
		2 words	F-102d	10
		256 bytes	F-176	14-2
	Write to file	1 byte	F-103	13-11
		1 word	F-103w	11
		2 words	F-103d	12
		256 bytes	F-177	14-3

Type				Mnemonics	See page
Arithmetic operation instructions	BCD addition	Register from register	2 digits + 2 digits	F-10	10-23
			4 digits + 4 digits	F-10w	24
			8 digits + 8 digits	F-10d	25
		Constant from register	2 digits + 2 digits	Fc10	26
			4 digits + 4 digits	Fc10w	27
			8 digits + 4 digits	Fc10d	28
	BCD subtraction	Register from register	2 digits - 2 digits	F-11	29
			4 digits - 4 digits	F-11w	30
			8 digits - 8 digits	F-11d	31
		Constant from register	2 digits - 2 digits	Fc11	32
			4 digits - 4 digits	Fc11w	33
			8 digits - 4 digits	Fc11d	34
	BCD multiplication	Register from register	4 digits x 4 digits	F-15	11-1
			8 digits x 8 digits	F-15d	2
		Constant from register	4 digits x 3 digits	Fc15	3
			8 digits x 4 digits	Fc15d	4
	BCD division	Register from register	4 digits ÷ 2 digits	F-16	5
			8 digits ÷ 8 digits	F-16d	6
			8 digits ÷ 8 digits (Decimal fraction 4 digits)	F-116	13-16
		Constant from register	4 digits ÷ 2 digits	Fc16	11-7
			8 digits ÷ 4 digits	Fc16d	8
	Binary addition	Register from register	8 bits + 8 bits	F-210	14-14
			16 bits + 16 bits	F-210w	15
			32 bits + 32 bits	F-210d	16
		Constant from register	8 bits + 8 bits	Fc210	17
			16 bits + 16 bits	Fc210w	17
			32 bits + 16 bits	Fc210d	18
		Register from register (with code)	31 bits + 31 bits	F-310	53
	Binary subtraction	Register from register	8 bits - 8 bits	F-211	19
			16 bits - 16 bits	F-211w	20
			32 bits - 32 bits	F-211d	20
		Constant from register	8 bits - 8 bits	Fc211	21
			16 bits - 16 bits	Fc211w	21
			32 bits - 16 bits	Fc211d	22
		Register from register (with code)	31 bits - 31 bits	F-311	54
	Binary division	Register from register	8 bits x 8 bits	F-215	30
			16 bits x 16 bits	F-215w	30
			32 bits x 32 bits	F-215d	31
		Constant from register	8 bits x 8 bits	Fc215	32
			16 bits x 16 bits	Fc215w	32
			32 bits x 16 bits	Fc215d	33
		Register from register (with code)	31 bits x 31 bits	F-315	55
	Binary multiplication	Register from register	8 bits ÷ 8 bits	F-216	34
			15 bits ÷ 15 bits	F-216w	35
			31 bits ÷ 31 bits	F-216d	36
		Constant from register	8 bits ÷ 8 bits	Fc216	37
			15 bits ÷ 15 bits	Fc216w	37
			31 bits ÷ 15 bits	Fc216d	38
		Register from register (with code)	31 bits ÷ 31 bits	F-316	56

Type				Mnemonics	See page
Logical operation instruction	Binary multiplication	Register from register	8 bits	F-13	10-42
			16 bits	F-13w	42
			32 bits	F-13d	43
		Register with octal constant	8 bits	Fc13	44
			16 bits	Fc13w	44
			32 bits	Fc13d	45
		Register with hexadecimal constant	8 bits	Fx13	46
			16 bits	Fx13w	46
			32 bits	Fx13d	47
	Binary division	Register from register	8 bits	F-14	48
			16 bits	F-14w	48
			32 bits	F-14d	49
		Register with octal constant	8 bits	Fc14	50
			16 bits	Fc14w	50
			32 bits	Fc14d	51
		Register with hexadecimal constant	8 bits	Fx14	52
			16 bits	Fx14w	52
			32 bits	Fx14d	53
	AND	Register from register	8 bits	F-17	11-9
			16 bits	F-17w	9
			32 bits	F-17d	10
		Register with octal constant	8 bits	Fc17	11
			16 bits	Fc17w	11
			32 bits	Fc17d	12
		Register with hexadecimal constant	8 bits	Fx17	13
			16 bits	Fx17w	13
			32 bits	Fx17d	14
	OR	Register from register	8 bits	F-18	15
			16 bits	F-18w	15
			32 bits	F-18d	16
		Register with octal constant	8 bits	Fc18	17
			16 bits	Fc18w	17
			32 bits	Fc18d	18
		Register with hexadecimal constant	8 bits	Fx18	19
			16 bits	Fx18w	19
			32 bits	Fx18d	20
	Complements		8 bits	F-09	10-21
			16 bits	F-09w	21
			32 bits	F-09d	22

Type					Mnemonics	See page
Compare instructions	Compare	Register from register		1 byte	F-12	10-35
				1 word	F-12w	36
				2 words	F-12d	37
				n bytes	F-112	13-13
				n words	F-112w	14
				n double word	F-112d	15
		Register with octal constant		1 byte	Fc12	10-38
				1 word	Fc12w	38
				2 words	Fc12d	39
		Register with hexadecimal constant		1 byte	Fx12	40
				1 word	Fx12w	40
				2 words	Fx12d	41
	Compare (with relay output)	>	Register from register	1 byte	F-180	14-4
				1 word	F-180w	5
				2 words	F-180d	6
			Register with octal constant	1 byte	Fc180	7
				1 word	Fc180w	8
				2 words	Fc180d	9
		<	Register from register	1 byte	F-181	4
				1 word	F-181w	5
				2 words	F-181d	6
			Register with octal constant	1 byte	Fc181	7
				1 word	Fc181w	8
				2 words	Fc181d	9
		=	Register from register	1 byte	F-182	4
				1 word	F-182w	5
				2 words	F-182d	6
			Register with octal constant	1 byte	Fc182	7
				1 word	Fc182w	8
				2 words	Fc182d	9
		≠	Register from register	1 byte	F-183	4
				1 word	F-183w	5
				2 words	F-183d	6
			Register with octal constant	1 byte	Fc183	7
				1 word	Fc183w	8
				2 words	Fc183d	9
		≤	Register from register	1 byte	F-184	4
				1 word	F-184w	5
				2 words	F-184d	6
			Register with octal constant	1 byte	Fc184	7
				1 word	Fc184w	8
				2 words	Fc184d	9
		≥ <	Register from register	1 byte	F-185	4
				1 word	F-185w	5
				2 words	F-185d	6
			Register with octal constant	1 byte	Fc185	7
				1 word	Fc185w	8
				2 words	Fc185d	9
	Window comparator		Register from register	1 byte	F-212	23
				1 word	F-212w	24
				2 words	F-212d	25
			Register with octal constant	1 byte	Fc212	26
				1 word	Fc212w	26
				2 words	Fc212d	27
			Register with hexadecimal constant	1 byte	Fx212	28
				1 word	Fx212w	28
				2 words	Fx212d	29

Type			Mnemonics	See page
Convert instructions	Convert BCD to BIN	2 digits → 8 bits	F-03	10-7
		4 digits → 16 bits	F-03w	7
			F-53	12-3
		8 digits → 32 bits	F-03d	10-8
			F-153	13·31
	Convert BIN to BCD	8 bits → 2 digits	F-04	10-9
		16 bits → 6 digits	F-04w	9
			F-54	12-3
		32 bits → 10 digits	F-04d	10-10
			F-154	13-31
	Convert HEX to ASCII		F-252	14-41
	Convert ASCII to HEX		F-253	42
	Convert second to hour·minute·second		F-155	13-32
	Convert hour·minute·second to second		F-156	13-33
	Decode from 4 to 16		F-50	12-1
	Encode from 16 to 4		F-51	1
	Decode 7 SEG		F-52	2
	10's complement	2 digits	F-56	5
		4 digits	F-56w	5
		8 digits	F-56d	6
	2's complement	8 bits	F-57	7
		16 bits	F-57w	7
		32 bits	F-57d	8
	Total of ON bit		F-58	9
	Polar coordinates system		F-28	11-28
	Rectangular coordinate system		F-29	11-29
Exchange instructions	Exchange data	1 byte	F-02	10-5
		1 word	F-02w	5
		2 words	F-02d	6
		n bytes	F-174	14-1
	Swap upper 4 bits with lower 4 bits	1 byte	F-55	12-4
		n bytes	F-175	14-1

Type			Mnemonics	See page
Data processing instructions	Insert data	1 byte	F-170	13-44
		1 word	F-170w	45
		2 words	F-170d	46
	Delete data	1 byte	F-171	47
		1 word	F-171w	48
		2 words	F-171d	49
	Search data	1 byte	F-172	50
		1 word	F-172w	51
		2 words	F-172d	52
	Change data	1 byte	F-173	53
		1 word	F-173w	54
		2 words	F-173d	55
	Sequencing data	1 byte	F-79	12-48
		1 word	F-79w	49
		2 words	F-79d	50
	SIN function		F-22	11-22
	COS function		F-23	23
	TAN function		F-24	24
	ASIN function		F-25	25
	ACOS function		F-26	26
	ATAN function		F-27	27
Bit processing instructions	Complement bit		F-43	44
	Differentiate at ON		F-44	45
	Differentiate at OFF		F-45	46
	Set coil		F-32	33
	Reset coil		F-33	34
	Multiplex bit	Indirect address	F-130	13-17
		Direct address	F-131	17
	Set/reset bit	Indirect address	F-132	18
		Direct address	F-133	18

Type				Mnemonics	See page
Timer/ counter instruc-tions	BCD up/down counter		2 digits	F-62	12-17
			4 digits	F-62w	18
			8 digits	F-62d	19
	Add BCD counter		2 digits	F-65	24
			4 digits	F-65w	24
			8 digits	F-65d	25
	Subtract BCD counter		2 digits	F-66	26
			4 digits	F-66w	26
			8 digits	F-66d	27
	Add binary counter	+ 1	1 byte	F-63	20
			1 word	F-63w	20
			2 words	F-63d	21
		+ 2	1 byte	F-163	13-40
			1 word	F-163w	40
			2 words	F-163d	41
		+ 4	1 byte	F-263	14-47
			1 word	F-263w	47
			2 words	F-263d	48
	Subtract binary counter	— 1	1 byte	F-64	12-22
			1 word	F-64w	22
			2 words	F-64d	23
		— 2	1 byte	F-164	13-42
			1 word	F-164w	42
			2 words	F-164d	43
		— 4	1 byte	F-264	14-49
			1 word	F-264w	49
			2 words	F-264d	50
	Expansion timer	Subtract timer (setting value, register address)		F-260	43
		Subtract timer (setting value, register address)		Fc260	44
	Expansion counter	Subtract counter (setting value, register address)		F-261	45
		Subtract counter (constant , register address)		Fc261	46
Shift instruc-tions	Reversible shift register		8 bits	F-60	12-10
			16 bits	F-60w	12
			32 bits	F-60d	13
			n bits (register address)	F-160	13-34
			n bits (constant address)	Fc160	35
	Asynchronous reversible shift register		1 byte	F-61	12-14
			1 word	F-61w	15
			2 words	F-61d	16
			n bytes	F-161	13-36
			n words	F-161w	38
			n double word	F-161d	39
	Digit shift (upper shift)			F-67	12-28
	Digit shift (lower shift)			F-68	28

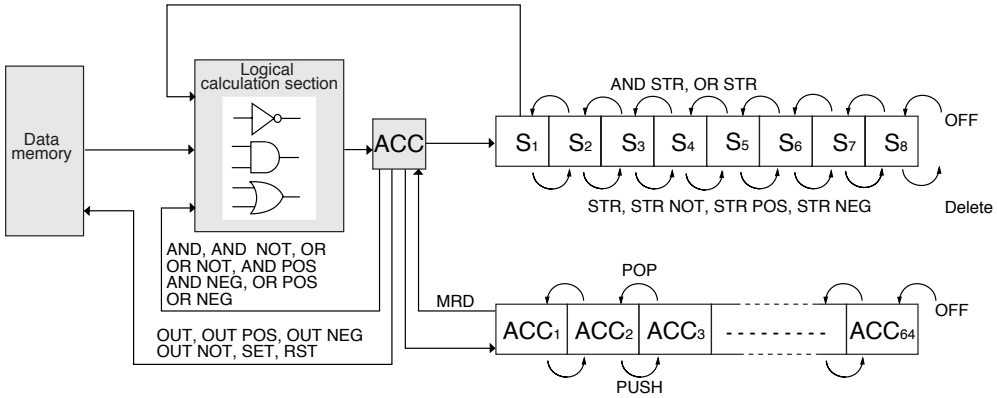
Type			Mnemonics	See page
Operational condition instructions	Set master control		F-30	11-30
	Reset master control		F-31	30
	Master control reset nesting		F-231	14-39
	Set jump control		F-41	11-42
	Reset jump control		F-42	42
	Jump control reset nesting		F-242	14-40
	Set level operating condition		F-47	11-47
	Reset level operating condition		F-48	47
	End	Unconditional end	F-40	41
		Conditional end	F-49	48
Branch instructions	Label		F-140	13-19
	Jump	Direct address	F-141	20
		Indirect address	F-151	30
	Call subroutine	Direct address	F-142	22
		Indirect address	F-148	28
	Return from subroutine	Unconditional return	F-143	22
		Conditional return	F-149	29
	Set loop count	Direct address	F-144	24
		Indirect address	F-146	26
	End of loop		F-145	24
	Forced end of loop		F-147	27

Type			Mnemonics	See page
Clock instructions	Comparison with current value of clock (specified relay set)		F-34	11-36
	Comparison with current value of clock (specified relay reset)		F-35	37
	Addition of time		F-36	38
	Subtraction of time		F-37	39
	Transfer of current value of clock		F-38	40
Communication instructions	Open channel instruction		F-202 F-203	14-10
	Open channel 1 (Set hierarchy communication)		F-206	13
	Open channel 2 (Set hierarchy communication)		F-207	13
	Sending instruction		F-204	14-11
	Receiving instruction		F-205	12
Other instructions	Maintenance display (MD)		F-20	11-21
	Square root		F-21	22
	Formation of sum check code of data		F-77	12-46
	Check data		F-78	47
	Refresh I/O		F-80	13-1
	Special I/O module	Refresh	F-82	2
		Read out	F-85	3
		Write	F-86	3
	Set the indirect address		F-100	7
			F-101	8
	Remark (instruction for comment identification)		F-90 (REM)	4
	Logging function		F-403	14-56

Chapter 8. Description of basic instructions

8-1 Calculation of basic instructions

For calculation of the basic instructions, the JW300 uses the data memory section, logical calculation section, accumulator (ACC), and stack registers (S_1 to S_8).



(1) Data memory

- This is a memory specified by relay numbers (input relay, output relay, auxiliary relay, timer/counter contact point). It stores ON/OFF data.
- With the basic instructions, the JW300 reads out ON/OFF data of the data memory with the following instructions.

  STR, STR NOT, AND, AND NOT, OR, OR NOT

  STR POS, STR NEG, AND POS, AND NEG, OR POS, OR NEG

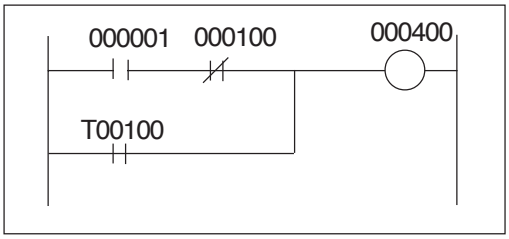
The JW300 writes data to the data memory with the following instructions.

OUT, OUT POS, OUT NEG, OUT NOT, SET, RST, timer/counter

[Example of writing instructions]

STR	000001
AND NOT	000100
OR	T00100
OUT	000400
↑ Instructions	↑ Relay number

[Example of ladder chart]



(2) Logical calculation section

According to instruction details, the JW300 carries out logical operations.

(3) Accumulator(ACC)

- This is a one-bit register that stores logical operation results.
- The register changes with the following instructions.

  STR, STR NOT, AND, AND NOT, OR, OR NOT, AND STR, OR STR

  STR POS, STR NEG, AND POS, AND NEG, OR POS, OR NEG, PUSH, POP

(4) Stack register (S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8)

- This is an 8-bit register in which the intermediate result is stored during processing of the serial/parallel circuit or counter instruction, and application instruction (F-60, F-60w, F-62, F-62w etc.) that has a plural number of input conditions.
- The register changes with the following instructions.

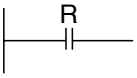
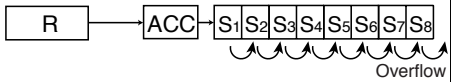
  STR, STR NOT, AND STR, OR STR, STR POS, STR NEG

(5) Clear internal memory area (ACC_1, ACC_2, ACC_3, - - - - -, ACC_{64})

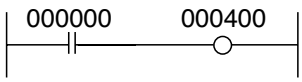
- These are areas to memorize contents of the accumulator and stack.
- The areas change with the PUSH and POP instructions.

8-2 Description of each basic instruction

STR (Store)

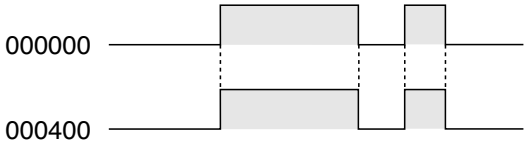
Symbol		
Function		Use when a 1st contact point from a bus line, or a 1st contact point of a circuit block is "a" contact (normally OFF).
Operation details		JW300 stores the data memory content (ON/OFF data) having relay number R to the accumulator (ACC). The previously located ON/OFF data in the ACC shifts to stack register S_1. The previous S_1 data shift to S_2. As such, other stack register data shift as S_2 -> S_3, S_3 => S_4, S_4 -> S_5, S_5 -> S_6, S_6 -> S_7, and S_7 -> S_8. The data in the S_8 stack register is deleted. 
After the operation	**R detail**	Latch
	ACC	Content of R data memory
	S_1	Content of the ACC before the operation
	S_2	Content of S_1 before the operation
	S_3	Content of S_2 before the operation
	S_4	Content of S_3 before the operation
	S_5	Content of S_4 before the operation
	S_6	Content of S_5 before the operation
	S_7	Content of S_6 before the operation
	S_8	Content of S_7 before the operation

[Use example 1]

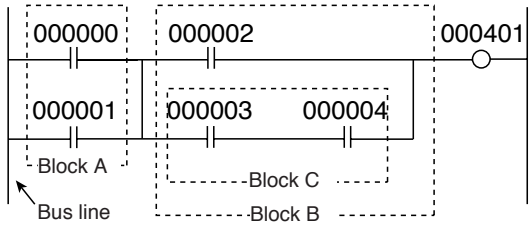


Instruction	
STR	000000
OUT	000400

When input relay 000000 is ON, output relay 000400 goes ON.



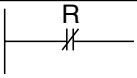
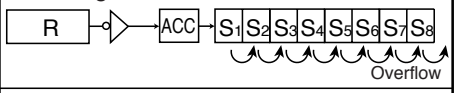
[Use example 2]



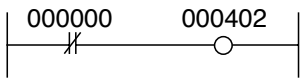

```
STR   000000 ------ 1st contact point of the bus line (block A)
OR    000001
STR   000002 ------ 1st contact point of block B
STR   000003 ------ 1st contact point of block C
AND   000004
OR    STR
AND   STR
OUT   000401
```

Use range of R	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 to 015777 020000 to 075777 T00000 to T01777 C00000 to C01777	000000 to 015777 020000 to 075777 100000 to 153777 T00000 to T03777 C00000 to C03777	000000 to 015777 020000 to 075777 100000 to 543777 T00000 to T17777 C00000 to C17777			

STR NOT (Store not)

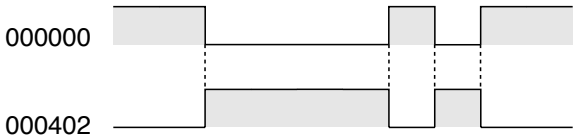
Symbol		
Function		Use when a 1st contact point from a bus line, or a 1st contact point of a circuit block is "a" contact (normally OFF).
Operation details		Inverts the contents (ON/OFF data) of the data memory having relay number R, and stores them in the accumulator (ACC). The previously located ON/OFF data in the ACC shifts to stack register S_1. The previous S_1 data shift to S_2. As such, other stack register data shift as S_2 -> S_3, S_3 => S_4, S_4 -> S_5, S_5 -> S_6, S_6 -> S_7, and S_7 -> S_8. The data in the S_8 stack register is deleted. 
After the operation	**R detail**	Latch
	ACC	The value after inverting the content of R
	S_1	Content of the ACC before the operation
	S_2	Content of S_1 before the operation
	S_3	Content of S_2 before the operation
	S_4	Content of S_3 before the operation
	S_5	Content of S_4 before the operation
	S_6	Content of S_5 before the operation
	S_7	Content of S_6 before the operation
	S_8	Content of S_7 before the operation

[Use example 1]

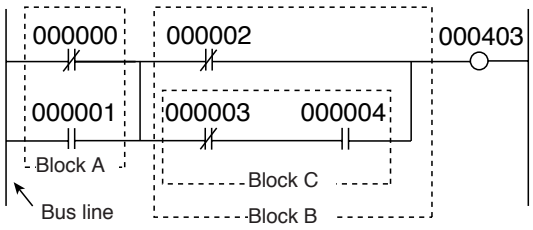


Instruction	
STR NOT	000000
OUT	000402

When input relay 000000 is ON, output relay 000402 goes ON.



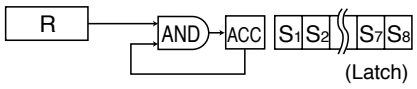
[Use example 2]



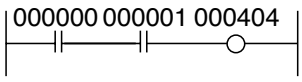

```
STR NOT 000000 ····· 1st contact point of the bus line (block A)
OR      000001
STR NOT 000002 ····· 1st contact point of block B
STR NOT 000003 ····· 1st contact point of block C
AND     000004
OR  STR
AND STR
OUT     000403
```

Use range of R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777 020000 to 075777 T00000 to T01777 C00000 to C01777	000000 to 015777 020000 to 075777 100000 to 153777 T00000 to T03777 C00000 to C03777	000000 to 015777 020000 to 075777 100000 to 543777 T00000 to T17777 C00000 to C17777			

AND (AND)

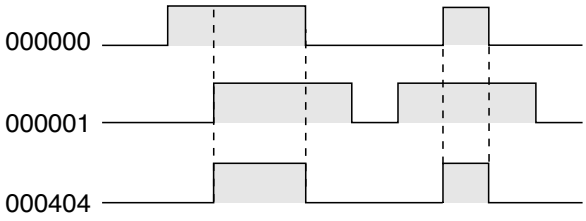
Symbol		R
Function		Use when the serial contact point is "a" contact (normally OFF).
Operation details		Executes AND operation between the content of the R data memory (ON/OFF data) and the content of the accumulator (ACC), and stores the result to the ACC. 
After the operation	**R detail**	Latch
	ACC	Value after AND operated the R content and ACC content before the operation.
	S1 to S8	Latch

[Use example]



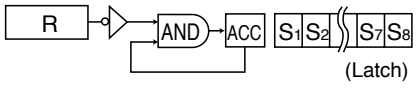
Instruction	
STR	000000
AND	000001
OUT	000404

If both the input relays 000000 and 000001 are ON, output relay 000404 goes ON.

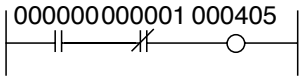


Use range of R	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 to 015777 020000 to 075777 T00000 to T01777 C00000 to C01777	000000 to 015777 020000 to 075777 100000 to 153777 T00000 to T03777 C00000 to C03777	000000 to 015777 020000 to 075777 100000 to 543777 T00000 to T17777 C00000 to C17777			

AND NOT (AND NOT)

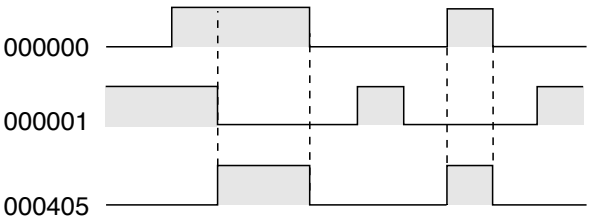
Symbol		R
Function		Use when the serial contact point is "b" contact (normally OFF).
Operation details		After inverting the content (ON/OFF data) of the data memory having relay number R, executes AND operation between the inverted data and the content of the accumulator (ACC), and store the result to the ACC. 
After the operation	**R detail**	Latch
	ACC	The value after the AND operation of the inversed content of the R and ACC content before the operation
	S1 to S8	Latch

[Use example]



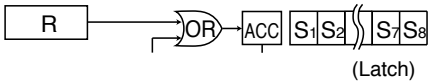
Instruction	
STR	000000
AND NOT	000001
OUT	000405

When input relay 000000 is ON and 000001 is OFF, output relay 000405 goes ON.

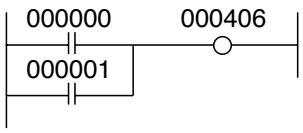


Use range of R	Same as the above (AND).

OR (OR)

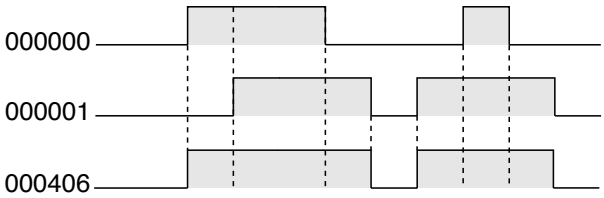
Symbol		R
Function		Use when a parallel contact point is "a" contact (normal OFF).
Operation details		Executes OR operation between the content of the R data memory (ON/OFF data) and the content of the accumulator (ACC), and stores the result to the ACC. 
After the operation	**R detail**	Latch
	ACC	Value after OR operated the R content and ACC content before the operation.
	S_1 to S_8	Latch

[Use example]



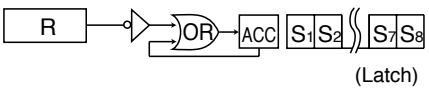
Instruction	
STR	000000
OR	000001
OUT	000406

If both the input relays 000000 and 000001 are ON, output relay 000406 goes ON.

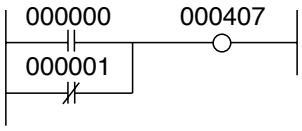


Use range of R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777 020000 to 075777 T00000 to T01777 C00000 to C01777	000000 to 015777 020000 to 075777 100000 to 153777 T00000 to T03777 C00000 to C03777	000000 to 015777 020000 to 075777 100000 to 543777 T00000 to T17777 C00000 to C17777			

OR NOT (OR NOT)

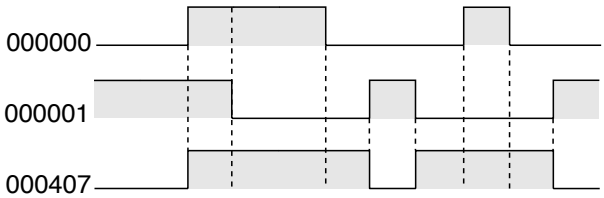
Symbol		R
Function		Use when a parallel contact point is "b" contact (normal OFF).
Operation details		After inverting the content (ON/OFF data) of the data memory having relay number R, executes OR operation between the inverted data and the content of the accumulator (ACC), and store the result to the ACC. 
After the operation	**R detail**	Latch
	ACC	The value after the OR operation of the inversed content of the R and ACC content before the operation
	S_1 to S_8	Latch

[Use example]



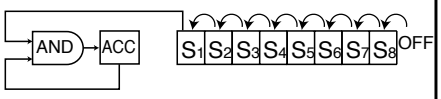
Instruction	
STR	000000
OR NOT	000001
OUT	000407

When input relay 000000 is ON or 000001 is OFF, output relay 000407 goes ON.

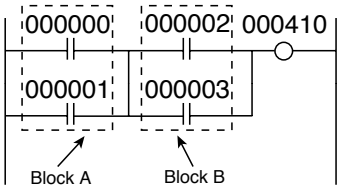


Use range of R	Same as the above (OR).

AND STR (AND STORE)

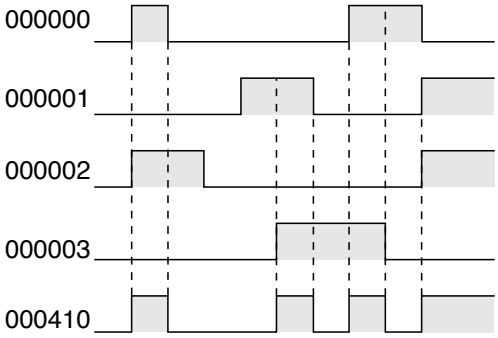
Function		Use to connect circuit blocks to each other in series.
Operation details		Executes AND operation between stack register S_1 and content (ON/OFF data) of the accumulator (ACC), and stores the result to the ACC. The previously located ON/OFF data in the ACC shifts to stack register S_1. The previous S_1 data shifts to S_2. Then, other stack register data shift as S_2 -> S_3, S_3 -> S_4, S_4 -> S_5, S_5 -> S_6, S_6 -> S_7, and S_7 -> S_8. The data in the S_8 stack register is entered as OFF data. 
After the operation	**ACC**	AND operation result between the S_1 content before calculation and ACC content
	S_1	Content of S_2 before the operation
	S_2	Content of S_3 before the operation
	S_3	Content of S_4 before the operation
	S_4	Content of S_5 before the operation
	S_5	Content of S_6 before the operation
	S_6	Content of S_7 before the operation
	S_7	Content of S_8 before the operation
	S_8	OFF (0)

[Use example]

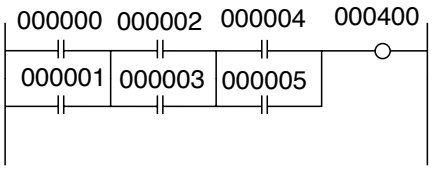


Instruction	
STR	000000
OR	000001
STR	000002
OR	000003
AND STR	
OUT	000410

Connect block A and block B in a series.
When input relay 000000 or 000001 is ON and 000002 or 000003 is ON, output relay 000410 goes ON.



Note 1: For programming the following ladder chart, two methods ("a" and "b") are available.



(a)

STR	000000
OR	000001
STR	000002
OR	000003
AND STR	
STR	000004
OR	000005
AND STR	
OUT	000400

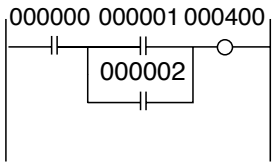
(b)

STR	000000
OR	000001
STR	000002
OR	000003
STR	000004
OR	000005
AND STR	
AND STR	
OUT	000400

The same calculation result can be obtained from both (a) and (b). However, (a) uses the S_1 stack register only and (b) uses stack registers S_1 and S_2. As the JW300 has 8 stack registers, programming (b) will restrict connection of blocks up to 9 blocks.

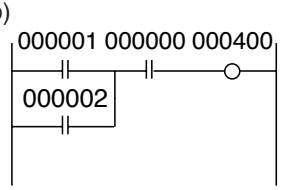
Note 2: The following examples (a) and (b) bring the identical operation.

(a)



STR	000000
STR	000001
OR	000002
AND STR	
OUT	000400

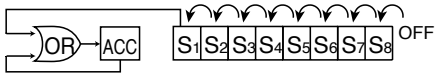
(b)



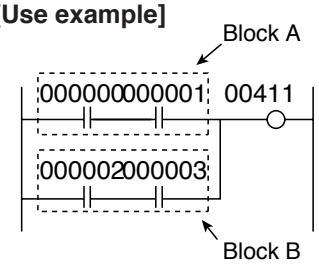
STR	000001
OR	000002
AND	000000
OUT	000400

The number of steps of (b) is small by one than (a).

OR STR (OR STORE)

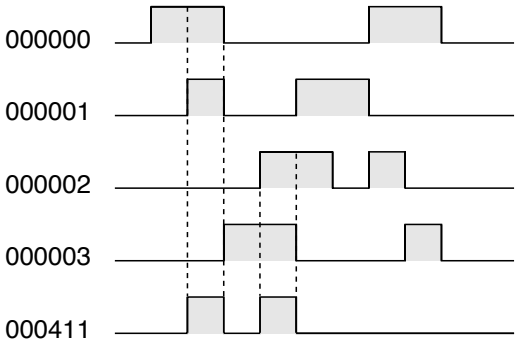
Function		Use to connect circuit blocks to each other in parallel.
Operation details		Executes OR operation between stack register S_1 and content (ON/OFF data) of the accumulator (ACC), and stores the result to the ACC. The previously located ON/OFF data in the ACC shifts to stack register S_1. The previous S_1 data shifts to S_2. Then, other stack register data shift as S_2 -> S_3, S_3 -> S_4, S_4 -> S_5, S_5 -> S_6, S_6 -> S_7, and S_7 -> S_8. The data in the S_8 stack register is entered as OFF data. 
After the operation	**ACC**	OR operation result between the S_1 content before calculation and ACC content
	S_1	Content of S_2 before the operation
	S_2	Content of S_3 before the operation
	S_3	Content of S_4 before the operation
	S_4	Content of S_5 before the operation
	S_5	Content of S_6 before the operation
	S_6	Content of S_7 before the operation
	S_7	Content of S_8 before the operation
	S_8	OFF (0)

[Use example]

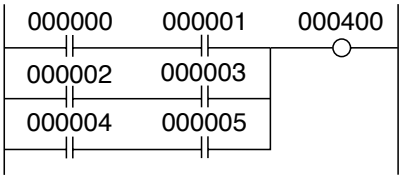


Instruction	
STR	000000
AND	000001
STR	000002
AND	000003
OR STR	
OUT	000411

Conned block A and B in parallel.
When input relay 000000 and 000001, or 000002 and 000003 are ON, output relay 000411 goes ON.



Note1: For programming the following ladder chart, two methods ("a" and "b") are available.



(a)

STR	000000
AND	000001
STR	000002
AND	000003
OR STR	
STR	000004
AND	000005
OR STR	
OUT	000400

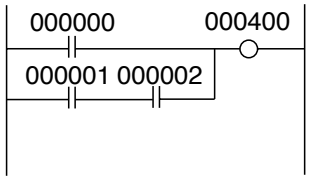
(b)

STR	000000
AND	000001
STR	000002
AND	000003
STR	000004
AND	000005
OR STR	
OR STR	
OUT	000400

The same calculation result can be obtained from both (a) and (b). However, (a) uses the S1 stack register only and (b) uses stack registers S1 and S2. As the JW300 has 8 stack registers, programming (b) will restrict connection of blocks up to 9 blocks.

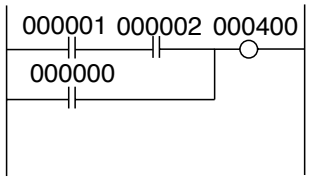
Note 2: The following examples (a) and (b) bring the identical operation.

(a)



STR	000000
STR	000001
AND	000002
OR STR	
OUT	000400

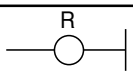
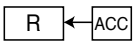
(b)



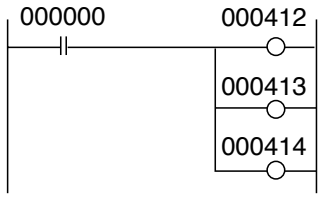
STR	000001
AND	000002
OR	000000
OUT	000400

The number of steps of (b) is small by one than (a).

OUT (OUT)

Symbol		
Function		Use to output calculation result.
Operation details		Write the content of the accumulator (ACC) to the data memory having relay number R. 
After the operation	**R detail**	ACC detail
	ACC	Latch
	S_1 to S_8	Latch

[Use example]



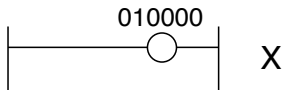
Instruction	
STR	000000
OUT	000412
OUT	000413
OUT	000414

When input relay 000000 is ON, output relay 000412, 000413, and 000414 go ON.
(With the OUT instruction, the ACC content does not change, so that continuous use of the OUT instruction is possible.)

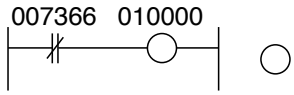
Use range of R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777 020000 to 075777	000000 to 015777 020000 to 075777 100000 to 153777	000000 to 015777 020000 to 075777 100000 to 543777			

Note 1: The special relay areas (007300 to 007377) cannot be used for output.

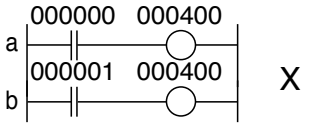
Note 2: The OUT instruction cannot be started from a bus line.



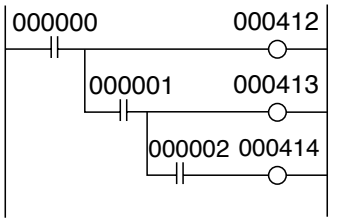
For a relay you want to normally be ON, use a normally OFF contact point (007366).



Note 3: If the same relay number is used double with the OUT instruction, the error will be shown with program check using support tool JW-15PG.



Note 4: After calculation of the OUT instruction, the content of the ACC does not change, so that the following program is also available.



STR	000000
OUT	000412
AND	000001
OUT	000413
AND	000002
OUT	000414

TMR (Timer instruction)

The TMR instructions include five types: decrement, increment (these decrement or increment installed 0.1 second clock), handle calculated values with BCD or binary.

It also can be used by specifying a register after changing the set value to a fixed value. Using this instruction by combining it with a write instruction to registers, you can easily change set values.

(1) TMR instruction types

Instruction	Operation	Notation	TMR number *1	Setting area *2
TMR	Decrement	BCD	00000 to 17777	Specify 0000 to 7999 registers
DTMR (BCD)			00000 to 17777	Specify 0000 to 7999 registers
DTMR (BIN)		Binary	00000 to 17777	Specify 0000 to 32767 registers
UTMR (BCD)	Increment	BCD	00000 to 17777	Specify 0000 to 7999 registers
UTMR (BIN)		Binary	00000 to 17777	Specify 0000 to 32767 registers

*1: The TMR number varies with the control module model used.

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
00000 to 01777	00000 to 03777	00000 to 17777			

The TMR number is shared between CNT and MD.

*2: For specifying registers, all of the byte addresses in the data memory can be specified. For specifying byte addresses, make sure to specify even address numbers using 2 bytes.

(2) Decrement TMR instructions

- Counting does not take place when the start input is OFF and the setting value keeps to be the current value, and the TMR contact is OFF.
- When the start input turns ON, the current value is decremented one step at every 0.1 second and come value=0, then the TMR contact is closed, and the condition is maintained as long as the start input is ON.

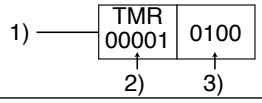
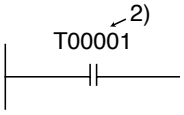
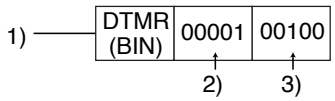
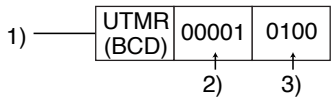
Start input	Current value	TMR contact
OFF	Setting value	OFF
ON (current value > 0)	Decremented one step at every 0.1 second	OFF
ON (current value = 0)	0	ON

(3) Increment TMR instructions

- While the start input is inactive (OFF), no count operation occurs, with the counter held at the current value=0 and the TMR contacts left open.
- When the start input is activated, the counter is incremented by one at 0.1 sec. intervals. When the setting value is reached, the TMR contacts are closed. This state is maintained as long as the start input remains active.

Start input	Current value	TMR contact
OFF	0	OFF
ON (current value < setting value)	Incremented one step at every 0.1 second	OFF
ON (current value = setting value)	Setting value	ON

(4) Symbol mark

Instruction	Symbol of TMR	Symbol of TMR contact
TMR		
DTMR (BCD)		
DTMR (BIN)		
UTMR (BCD)		
UTMR (BIN)		

1) Start input
Start with ON state signal

2) TMR number
00000 to 17777(8) ··· { 00000 to 00777: Common among TMR, CNT, and MD
01000 to 17777: Common between TMR and CNT

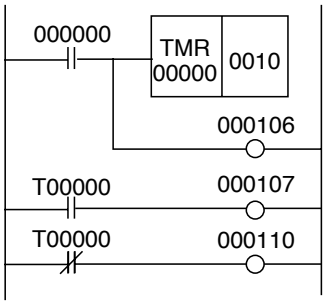
3) Setting value
0.1 second (100ms) unit
0.01 second (10ms) unit*
0.001 second (1ms) unit*

*Setting system memory #0227 can set TMR00000 to 00777 to 10 ms timer. Setting system memory #0225 can set TMR01770 to 01777 to 1 ms timer.
However, the DTMR and UTMR function as 100ms only.

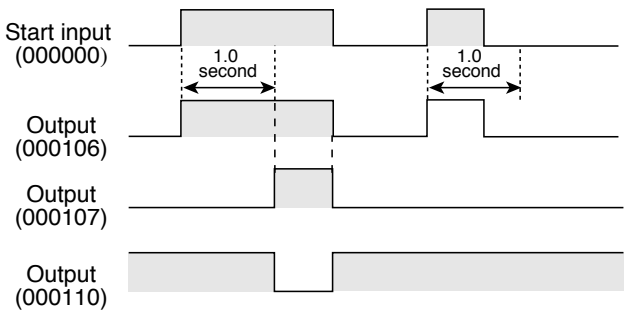
4) Precision
100 ms (setting value $^{+0}_{-0.1s}$) + scan time (second)
10 ms (setting value $^{+0}_{-0.01s}$) + scan time (second)
1 ms (setting value $^{+0}_{-0.001s}$) + scan time (second)

(5) Use example

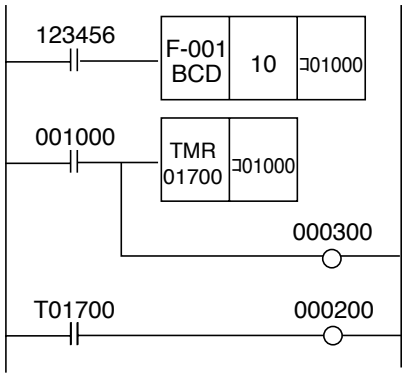
● **Example 1**



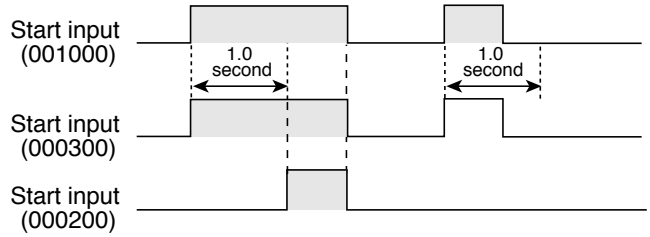
Instruction	
STR	000000
TMR	00000
	0010
OUT	000106
STR	T00000
OUT	000107
STR NOT	T00000
OUT	000110



● **Example 2**

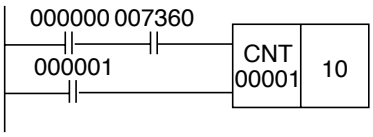


Instruction	
STR	123456
F-001	
	10
	⊐01000
STR	001000
TMR	01700
	⊐01000
OUT	000300
STR	T01700
OUT	000200

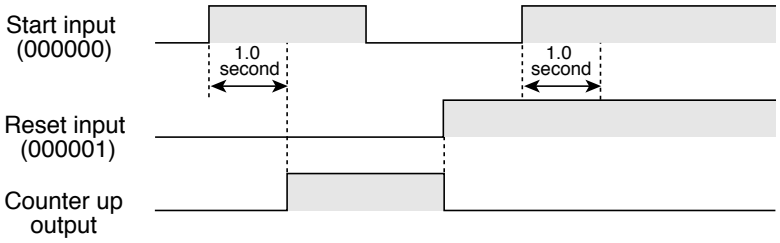


Remarks

- Because the TMR number is used common with CNT and MD, do not use the number used for a CNT and MD number for the TMR number. Also, do not use the same number. If the same number be used, an error alert will be issued by the program in a support tool.
- Specify the same number as the TMR number for the TMR contact, and you may use as many a-contact and b-contacts as required.
- Current value of TMR is stored in "b*****." => See page 2-8
- The timer is reset when the JW300 is turned ON. Therefore, the reset function activates and results that current value becomes setting value even if the JW300 is turned ON with the timer's start in the ON state.
- If the timer reset condition is set in the system memory (#0201) it is possible to store the state at the time of a power interrupt.
- It is possible by using the contact 007360 (0.1 second clock) and the CNT instruction to implement the power failure store timer and the timers that differ in their start and reset condition.



Instruction	
STR	000000
AND	007360
STR	000001
CNT	00001
	10



CNT (Counter instruction)

The CNT instructions include five types: decrement, increment (these decrement or increment the counter at the rising edge of the counter input), handle calculated values with BCD or binary.
It also can be used by specifying a register after changing the set value to a fixed value. Using this instruction by combining it with a write instruction to registers, you can easily change set values.

(1) CNT instruction types

Instruction	Operation	Notation	CNT number *1	Setting area *2
CNT	Decrement	BCD	00000 to 17777	Specify 0000 to 7999 registers
DCNT (BCD)			00000 to 17777	Specify 0000 to 7999 registers
DCNT (BIN)		Binary	00000 to 17777	Specify 0000 to 32767 registers
UCNT (BCD)	Increment	BCD	00000 to 17777	Specify 0000 to 7999 registers
UCNT (BIN)		Binary	00000 to 17777	Specify 0000 to 32767 registers

*1: The CNT number varies with the control module model used.

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
00000 to 01777	00000 to 03777	00000 to 17777			

The CNT number is shared between CNT and MD.

*2: For specifying registers, all of the byte addresses in the data memory can be specified. For specifying byte addresses, make sure to specify even address numbers using 2 bytes.

(2) Decrement CNT instructions

- No counting takes place even if the counter input changes its state from OFF to ON as long as the reset input is ON and the setting value is retained for the current value and the CNT contact is OFF.
- When the counter changes from OFF to ON while the reset input is OFF, the current value is decremented by one. When the current value becomes 0, the CNT contact returns ON and its state is maintained until the reset input is turned OFF.

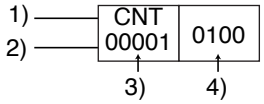
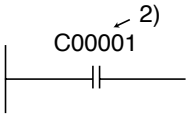
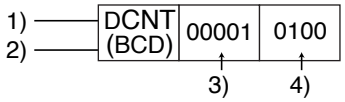
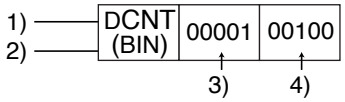
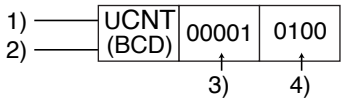
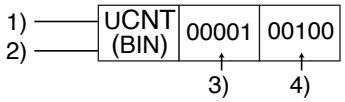
Reset input	Current value	CNT contact
ON	Setting value	OFF
OFF (current value > 0)	Decremented one step each time the counter input changes from OFF to ON	OFF
OFF (current value = 0)	0	ON

(3) Increment CNT instructions

- No counting takes place even if the counter input changes its state from OFF to ON as long as the reset input is ON and the 0 is retained for the current value and the CNT contact is OFF.
- While the reset input is inactive, the counter is incremented by one each time the count input is set to ON. When the preset value is reached, the CNT contacts are closed. This state is maintained as long as the reset input remains inactive.

Reset input	Current value	CNT contact
ON	0	OFF
OFF (current value < 0)	Incremented one step each time the counter input changes from OFF to ON	OFF
OFF (current value = 0)	Setting value	ON

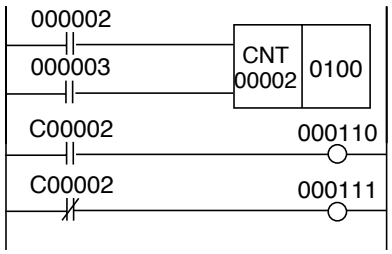
(4) Symbol mark

Instruction	Symbol of CNT	Symbol of CNT contact
CNT		
DCNT (BCD)		
DCNT (BIN)		
UCNT (BCD)		
UCNT (BIN)		

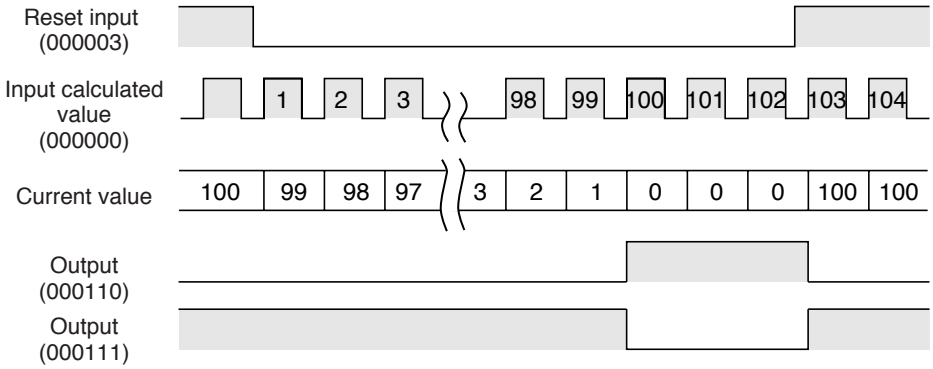
1) Counter input
   Senses OFF to ON transition
2) Reset input
   Reset with ON
3) CNT number
   00000 to $17777_{(8)}$ ⋯ { 00000 to 00777: Common among TMR, CNT, and MD
   01000 to 17777: Common between TMR and CNT
4) Setting value

(5) Use example

● Example 1

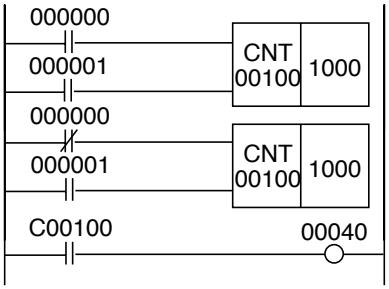


Instruction	
STR	000002
STR	000003
CNT	00002
	0100
STR	C00002
OUT	00110
STR NOT	C00002
OUT	00111



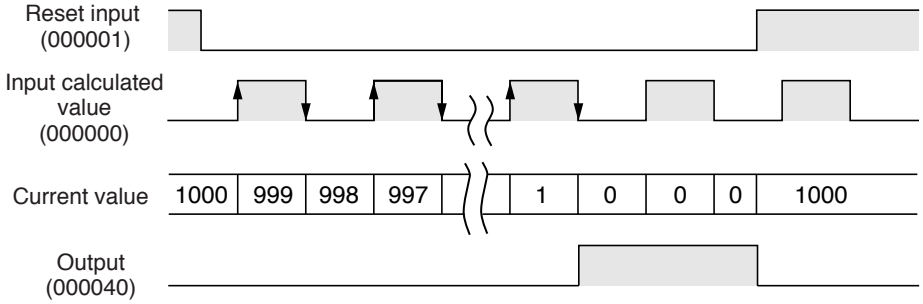
● Example 2

Counters that count at rising and falling edges of count input

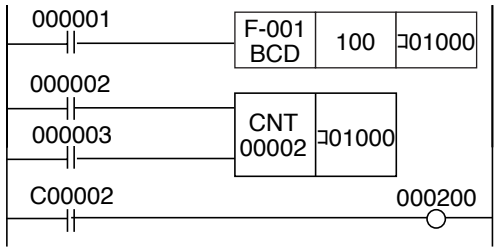


Instruction	
STR	000000
STR	000001
CNT	00100
	1000
STR NOT	000000
STR	000001
CNT	00100
	1000
STR	C00100
OUT	000040

- These are counters that subtract with both cases when a count input changes from OFF to ON and from ON to OFF.

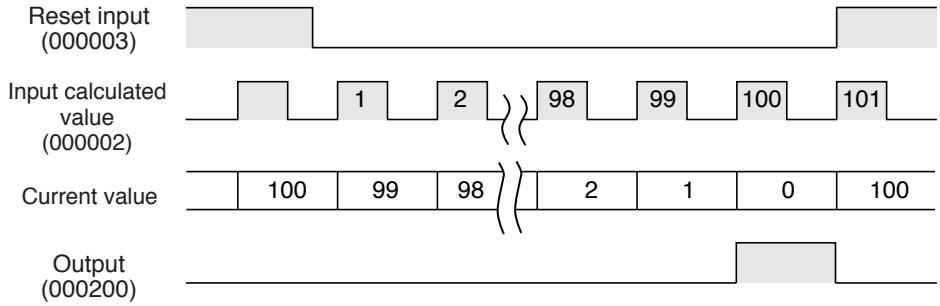


● **Example 3**



Instruction	
STR	000001
F-001	
	100
	⊐01000
STR	000002
STR	000003
CNT	00002
	⊐01000
STR	C00002
OUT	000200

- These are counters that subtract with both cases when a count input changes from OFF to ON and from ON to OFF.

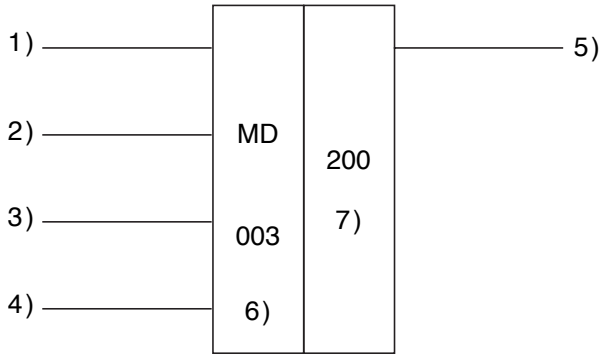


Remarks

- Because the CNT number is used common with TMR and MD, do not use the number used for a TMR and MD number for the CNT number. Also, do not use the same number. If the same number be used, an error alert will be issued by the program in a support tool.
  Also, do not use the same CNT number. If the same number be used, an error alert will be issued by the program in the programmer. You may ignore this warning if it is used intentionally.
- Specify the same number as the CNT number for the CNT contact, and you may use as many contacts and b-contacts as required.
- As successive inputs are disregarded when the counter has reached zero, the reset input must be set ON and OFF or forced to reset using a support tool, in order to start counting again.
- If both the counter input and the reset input go ON simultaneously, the reset input takes preference over the counter input.
- Current value of TMR is stored in "b*****." => See page 2-8
- When a power interrupt is encountered, the counter retains the current value. But, the current value will be reset if the reset input was programmed to turn ON at power ON. If it is necessary to retains the current value after the power interrupt, use the reset input that turns OFF at power ON.
- It would be possible to reset with OFF, if the reset condition is given in the system memory (#0202).

MD (Maintenance display)

The MD (Maintenance Display) instruction is used to monitor the operating condition of the controlled device and to trace a trouble cause in the support tool such as the programmer or send the output signal to an external source.

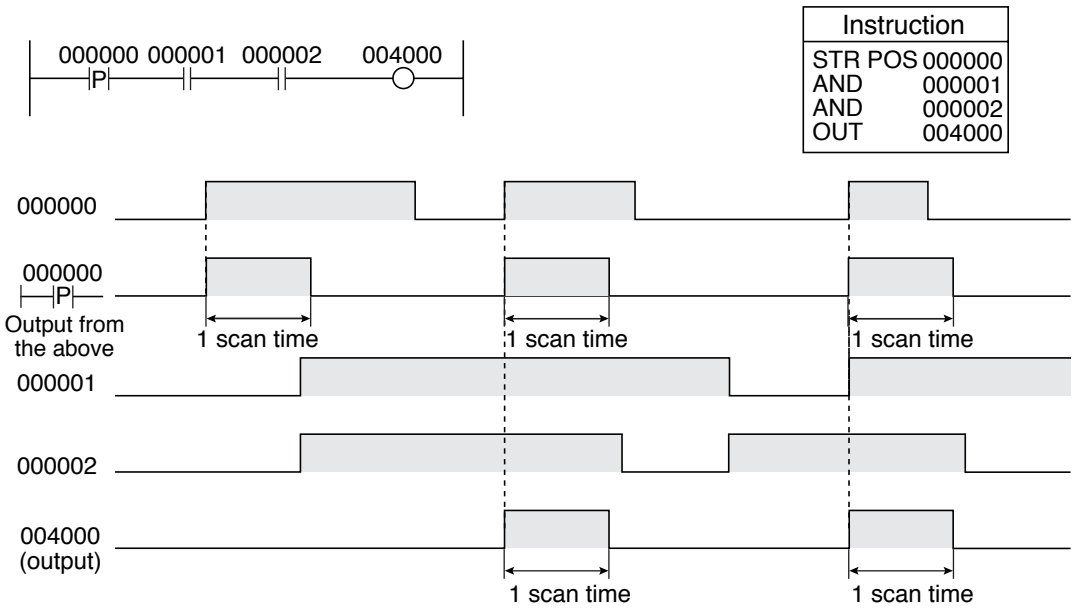


1) 2) 3)	**Input information**	This is an external output contact information used in conjunction with the MD data of 7). Relay, TMR, and CNT contact may be used.
4)	**Output direct condition**	This is an input used to direct whether the contact information of 1), 2), 3), and MD data of 7) be output to the data memory or relay area of the MD number specified by 6). Relay, TMR, and CNT contact may be used. Output becomes active when ON. No change occurs in the contact information and MD information even if it was turned OFF.
5)	**MD expansion output**	There is no need of programming the MD instruction with the condition of 4) above, when using the MD instruction continuously with the same output direct condition.
6)	**MD number**	The MD instruction uses the TMR and CNT current value storage area or relay area for the storage of contact information of 1), 2), and 3) and MD data of 7). (1) To use the TMR or CNT area Similar as TMR and CNT, it should be programmed using the number of 000 to 777 and the information must be monitored on the programmer, etc. Note: It is not possible to use the same number used for the TMR and CNT. (2) When the relay area is used It should be programmed in terms of byte address ∃xxxxx. As an example, program ∃00000, 2 bytes of ∃00000 and ∃00001 will be MD area. Use of the area connected with the output module will permit external output (display) of the contact information of 1), 2), 3) and the MD data of 7).
7)	**MD data**	Any number of 000 to 999 may be used in the BCD code. It should be programmed in connection with the process number, relay number, external device number, etc.

STR POS "a" contact load rising

Symbol	R P					
Function	When a relay number R that was specified by STR POS instruction changes from OFF to ON, this contact turns ON for one scan time. - This instruction operates only for one scan even it is entered between F-47 (set level calculation condition) and F-48 (reset level calculation condition). - This instruction does not operate for one scan interval soon after turning ON the power and soon after start operation.					
Use range of R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777	000000 to 015777	000000 to 015777			
	020000 to 075777	020000 to 075777	020000 to 075777			
	T00000 to T01777	100000 to 153777	100000 to 543777			
	C00000 to C01777	T00000 to T03777	T00000 to T17777			
		C00000 to C03777	C00000 to C17777			

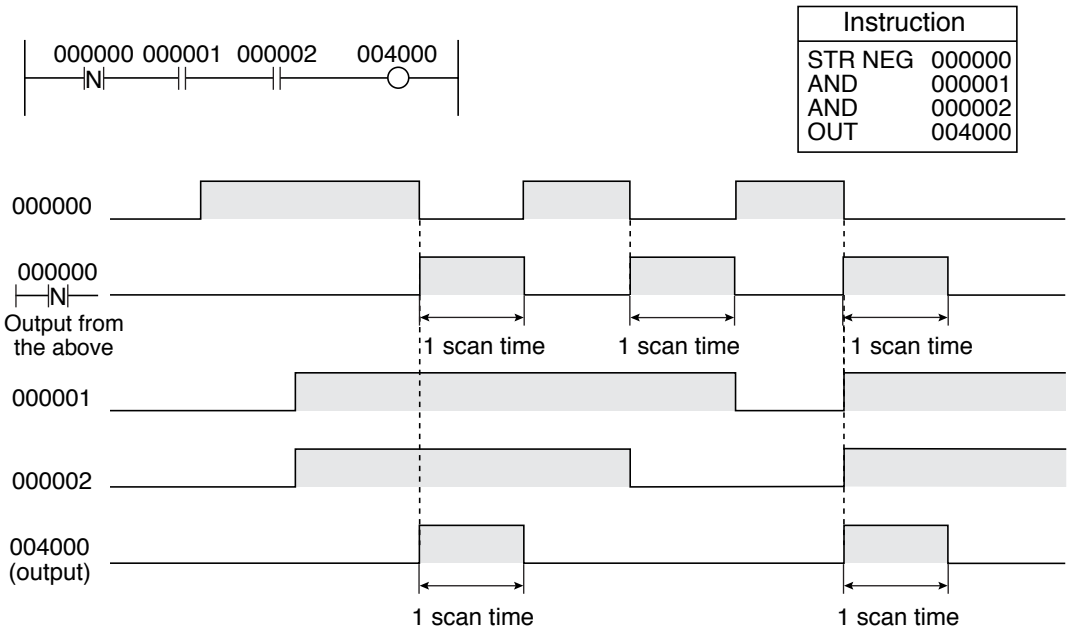
[Use example]



STR NEG "a" contact load falling

Symbol	R N					
Function	When a relay number R that was specified by STR NEG instruction changes from OFF to ON, this contact turns ON for one scan time. - This instruction operates only for one scan even it is entered between F-47 (set level calculation condition) and F-48 (reset level calculation condition). - This instruction does not operate for one scan interval soon after turning ON the power and soon after start operation.					
Use range of R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777	000000 to 015777	000000 to 015777			
	020000 to 075777	020000 to 075777	020000 to 075777			
	T00000 to T01777	100000 to 153777	100000 to 543777			
	C00000 to C01777	T00000 to T03777	T00000 to T17777			
		C00000 to C03777	C00000 to C17777			

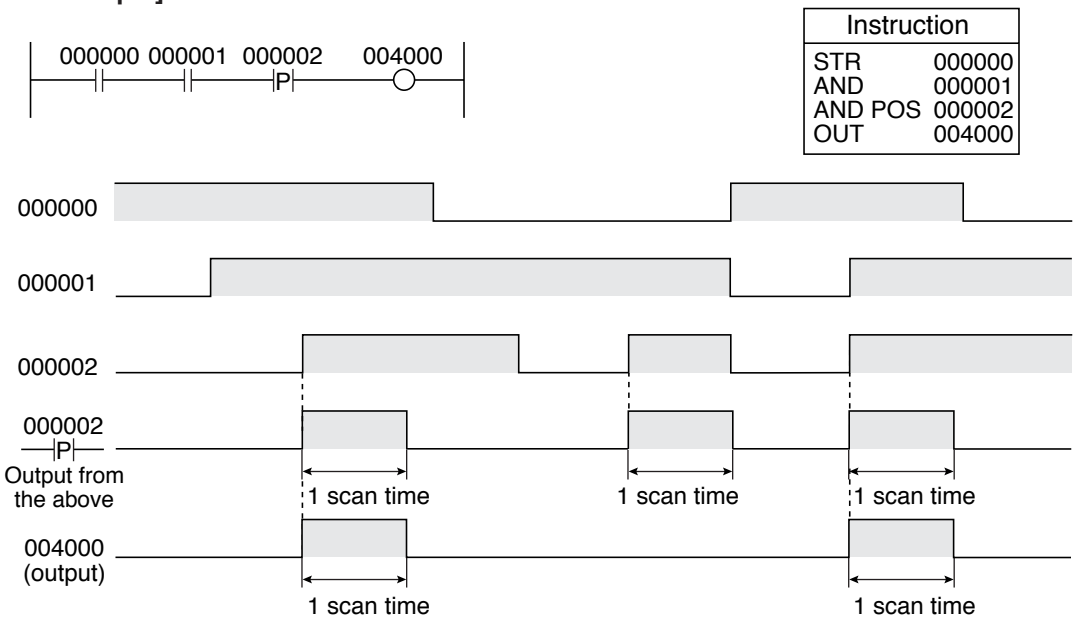
[Use example]



AND POS "a" contact AND rising

Symbol	R P					
Function	When relay number R that was specified by AND POS instruction changes from OFF to ON, this contact generates pulses of one scan time length, and executes AND operation with the ACC content, and stores its result to the ACC. - This instruction operates only for one scan even it is entered between F-47 (set level calculation condition) and F-48 (reset level calculation condition). - This instruction does not operate for one scan interval soon after turning ON the power and soon after start operation.					
Use range of R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777	000000 to 015777	000000 to 015777			
	020000 to 075777	020000 to 075777	020000 to 075777			
	T00000 to T01777	100000 to 153777	100000 to 543777			
	C00000 to C01777	T00000 to T03777	T00000 to T17777			
		C00000 to C03777	C00000 to C17777			

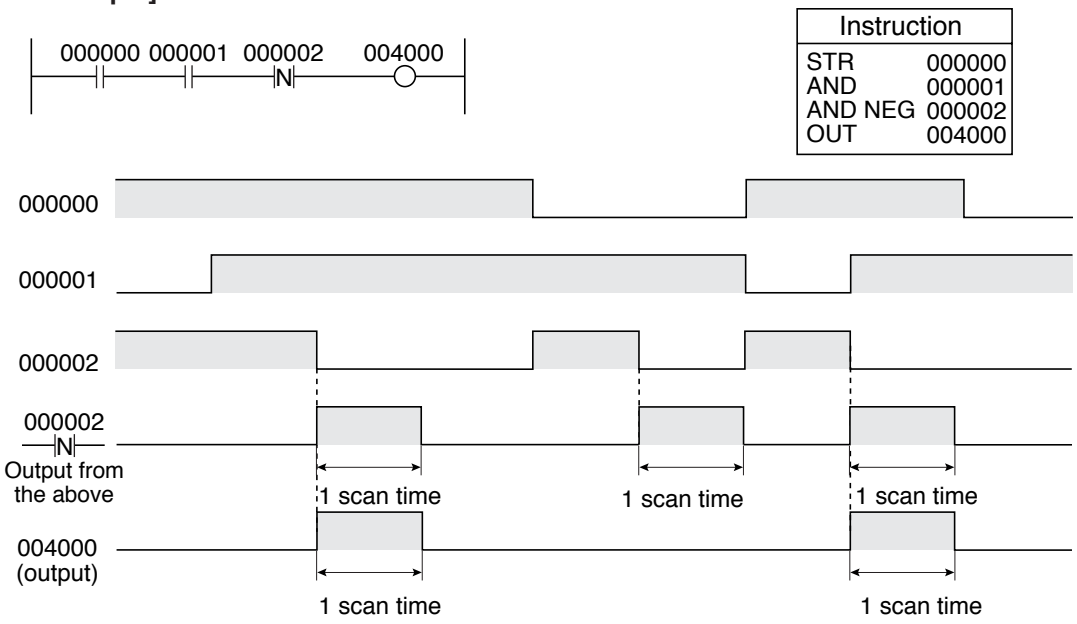
[Use example]



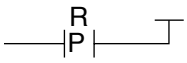
AND NEG "a" contact AND falling

Symbol	R N					
Function	When relay number R that was specified by AND NEG instruction changes from OFF to ON, this contact generates pulses of one scan time length, and executes AND operation with the ACC content, and stores its result to the ACC. - This instruction operates only for one scan even it is entered between F-47 (set level calculation condition) and F-48 (reset level calculation condition). - This instruction does not operate for one scan interval soon after turning ON the power and soon after start operation.					
Use range of R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777	000000 to 015777	000000 to 015777			
	020000 to 075777	020000 to 075777	020000 to 075777			
	T00000 to T01777	100000 to 153777	100000 to 543777			
	C00000 to C01777	T00000 to T03777	T00000 to T17777			
		C00000 to C03777	C00000 to C17777			

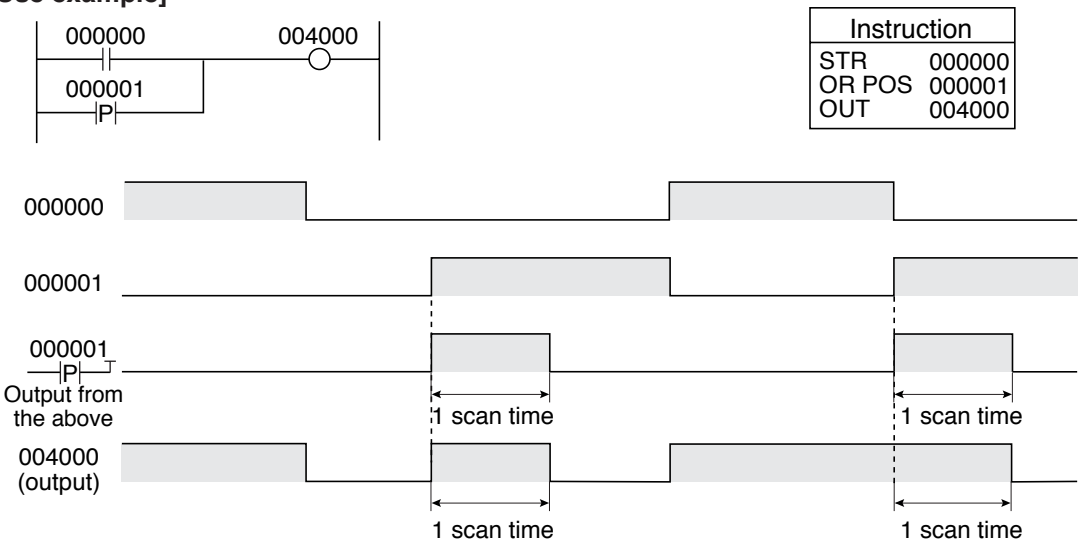
[Use example]



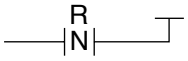
OR POS "a" contact OR rising

Symbol						
Function	When relay number R that was specified by OR POS instruction changes from OFF to ON, this contact generates pulses of one scan time length, and executes AND operation with the ACC content, and stores its result to the ACC. - This instruction operates only for one scan even it is entered between F-47 (set level calculation condition) and F-48 (reset level calculation condition). - This instruction does not operate for one scan interval soon after turning ON the power and soon after start operation.					
Use range of R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777	000000 to 015777	000000 to 015777			
	020000 to 075777	020000 to 075777	020000 to 075777			
	T00000 to T01777	100000 to 153777	100000 to 543777			
	C00000 to C01777	T00000 to T03777	T00000 to T17777			
		C00000 to C03777	C00000 to C17777			

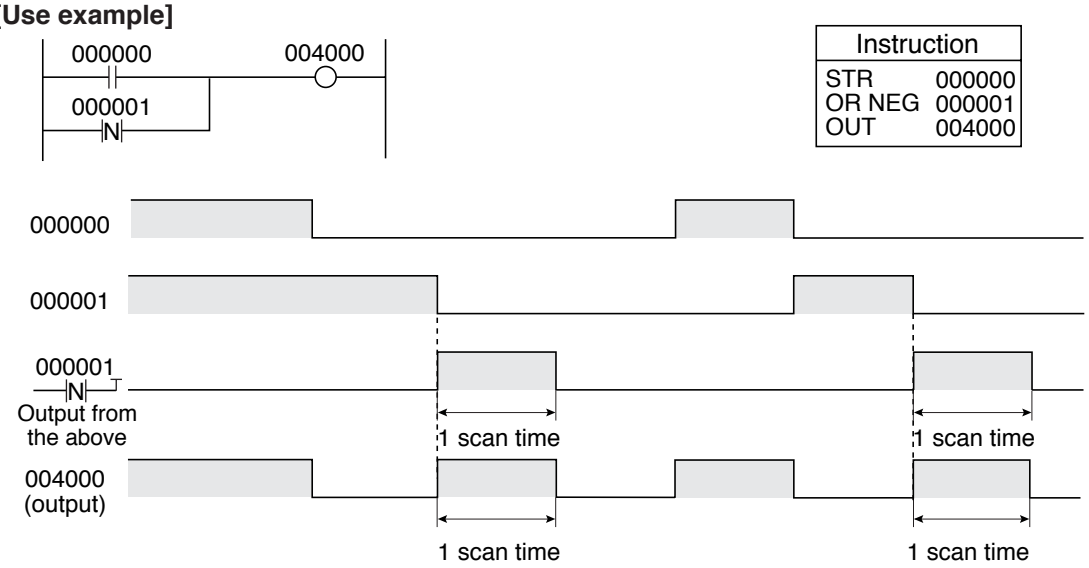
[Use example]



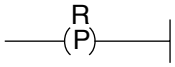
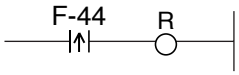
OR NEG "a" contact OR falling

Symbol						
Function	When relay number R that was specified by OR NEG instruction changes from OFF to ON, this contact generates pulses of one scan time length, and executes AND operation with the ACC content, and stores its result to the ACC. - This instruction operates only for one scan even it is entered between F-47 (set level calculation condition) and F-48 (reset level calculation condition). - This instruction does not operate for one scan interval soon after turning ON the power and soon after start operation.					
Use range of R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777	000000 to 015777	000000 to 015777			
	020000 to 075777	020000 to 075777	020000 to 075777			
	T00000 to T01777	100000 to 153777	100000 to 543777			
	C00000 to C01777	T00000 to T03777	T00000 to T17777			
		C00000 to C03777	C00000 to C17777			

[Use example]

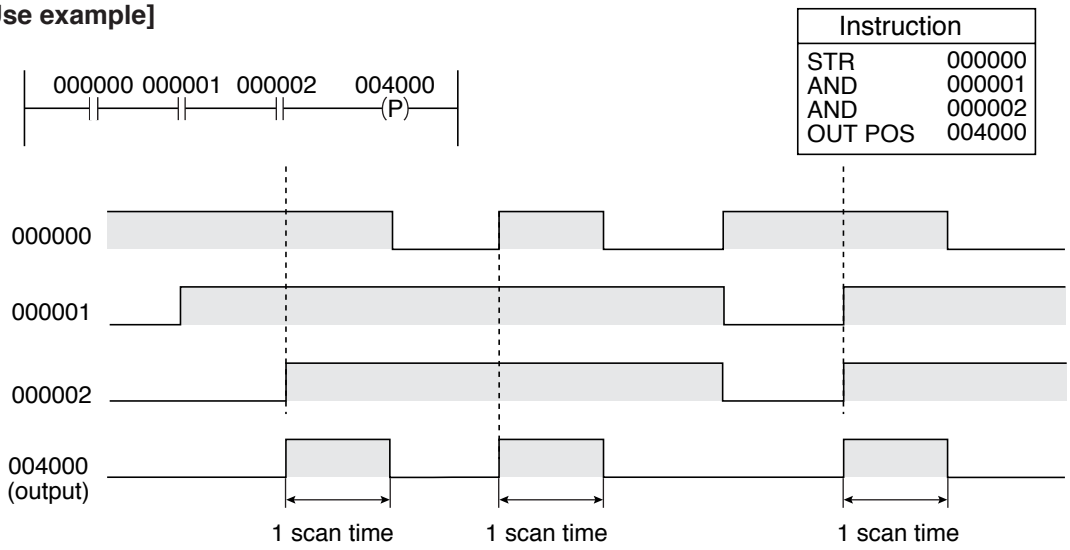


OUT POS Rising coil

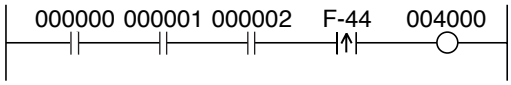
Symbol	
Function	When the ACC (accumulator) status just before the OUT POS instruction changes from OFF to ON, this contact outputs pulses of one scan time, and transfer the ACC content to the data memory. The operation is the same as the below operation.  - This instruction operates only for one scan even it is entered between F-47 (set level calculation condition) and F-48 (reset level calculation condition). - This instruction does not operate for one scan interval soon after turning ON the power and soon after start operation.

Use range of R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777 020000 to 075777	000000 to 015777 020000 to 075777 100000 to 153777	000000 to 015777 020000 to 075777 100000 to 543777			

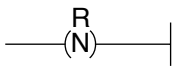
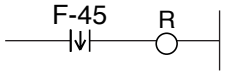
[Use example]



With the example above, the operation will be same as below.

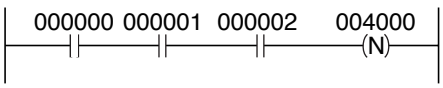


OUT NEG Falling coil

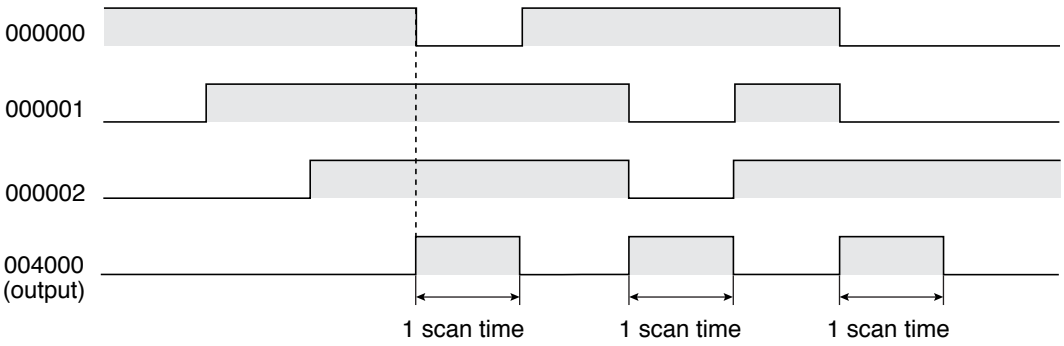
Symbol	
Function	When the ACC (accumulator) status just before the OUT NEG instruction changes from OFF to ON, this contact outputs pulses of one scan time, and transfer the ACC content to the data memory. The operation is the same as the below operation.  - This instruction operates only for one scan even it is entered between F-47 (set level calculation condition) and F-48 (reset level calculation condition). - This instruction does not operate for one scan interval soon after turning ON the power and soon after start operation.

Use range of R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777 020000 to 075777	000000 to 015777 020000 to 075777 100000 to 153777	000000 to 015777 020000 to 075777 100000 to 543777			

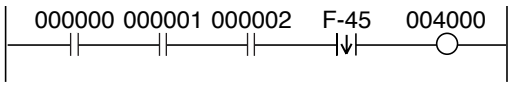
[Use example]



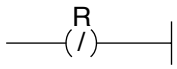
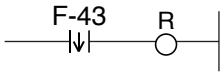
Instruction	
STR	000000
AND	000001
AND	000002
OUT NEG	004000



- With the example above, the operation will be same as below.

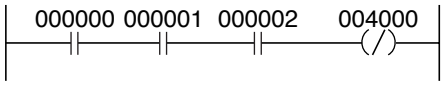


OUT NOT Inverse output input conditions

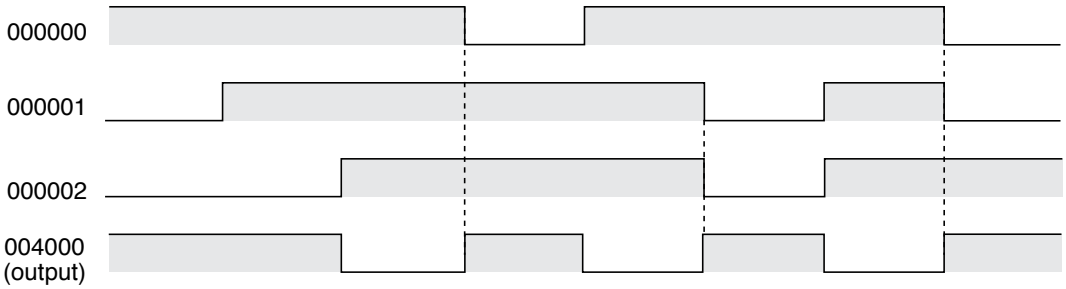
Symbol	
Function	Inverts the ACC content and outputs to a specified data memory. The operation is the same as the below operation. 

Use range of R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777 020000 to 075777	000000 to 015777 020000 to 075777 100000 to 153777	000000 to 015777 020000 to 075777 100000 to 543777			

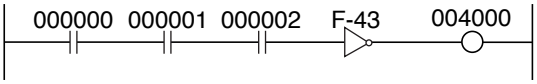
[Use example]



Instruction		
STR		000000
AND		000001
AND		000002
OUT	NOT	004000



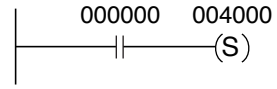
- With the example above, the operation will be same as below.



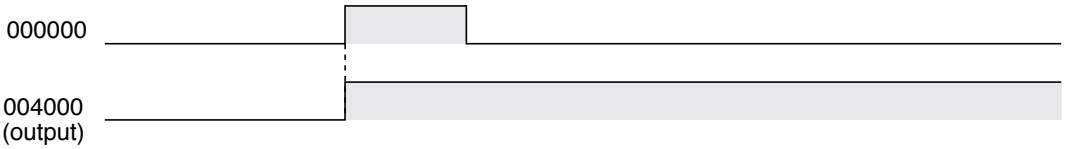
SET Set when the input signal is rising

Symbol	R (S)					
Function	When the set input goes ON, this instruction turns ON specified OUT and latches.					
Use range of R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777 020000 to 075777	000000 to 015777 020000 to 075777 100000 to 153777	000000 to 015777 020000 to 075777 100000 to 543777			

[Use example]

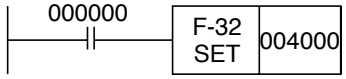


Instruction	
STR	000000
SET	004000



When set input 000000 changes from OFF to ON, OUT004000 goes ON.
The OUT004000 keeps its ON status even if set input 000000 goes OFF.
When set input 000000 is OFF, OUT004000 does not change.

With the example above, the operation will be same as below.

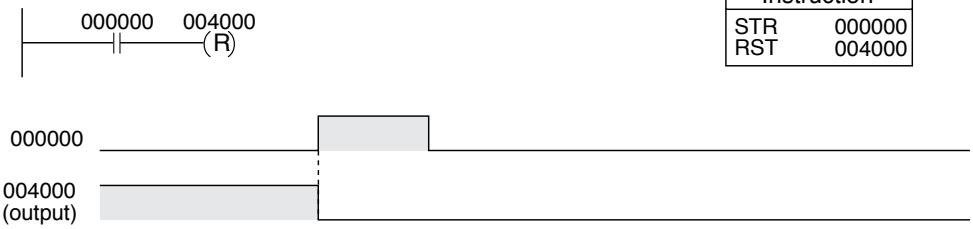


- If this instruction is located in F-30 (MCS) instruction, the OUT that was turned ON by this instruction keeps ON status even if the F-30 instruction goes OFF.
- Using this instruction, you can control one OUT on multiple circuits.
- If the OUT that is specified by this instruction locates inside a latch specified area, this OUT will be latched after recovery from a power failure. If the OUT is out of the latch specified area, it will be reset after recovery from a power failure.
- If the OUT that was specified by this instruction is in the area to latch output when the JW300 stops operation, the status before stopping the JW300 will be kept. If the OUT is out of the latch output area, it will be reset when the JW300 operation is stopped.
  => See system memory #0232, #0233, #0252, and #0253.
- Use this instruction as a pair with RST instruction (next page).
- The SET instruction and RST instruction that are between MCS (F-30) and MCR (F-31) do not function when MCS (F-30) instruction condition is OFF.

RST Reset when input is rising

Symbol	R (R)					
Function	When the set input goes ON, this instruction turns OFF specified OUT and latches. 000000 — F-33 RST 004000					
Use range of relay number R	**JW-311CU JW-312CU**	**JW-321CU JW-322CU**	**JW-331CU JW-332CU**	**JW-341CU JW-342CU**	**JW-352CU**	**JW-362CU**
	000000 to 015777 020000 to 075777	000000 to 015777 020000 to 075777 100000 to 153777	000000 to 015777 020000 to 075777 100000 to 543777			

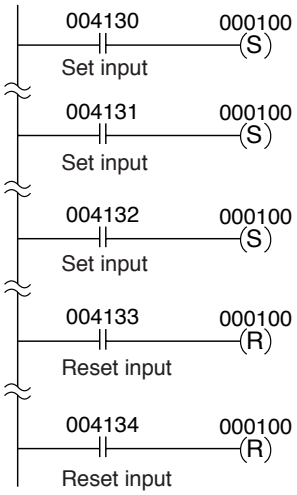
[Use example]



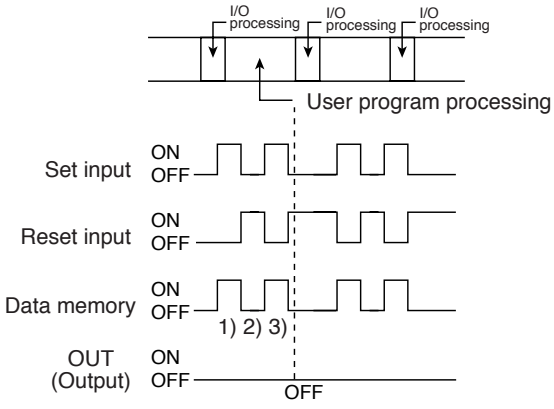
When reset input 000000 changes from OFF to ON, OUT004000 goes ON.
The OUT004000 keeps its ON status even if reset input 000000 goes OFF.
When reset input 000000 is OFF, OUT004000 does not change.

- If the OUT that is specified by this instruction locates inside a latch specified area, this OUT will be latched after recovery from a power failure. If the OUT is out of the latch specified area, it will be reset after recovery from a power failure.
- If the OUT that was specified by this instruction is in the area to latch output when the JW300 stops operation, the status before stopping the JW300 will be kept. If the OUT is out of the latch output area, it will be reset when the JW300 operation is stopped.
  => See system memory #0232, #0233, #0252, and #0253.
- Use this instruction as a pair with RST instruction (previous page).

- Using the SET instruction and RST instruction, you can control one OUT with multiple conditions.



- When the SET input and RESET input are turned ON/OFF more than one time within one scan, data memory used as OUT repeats ON/OFF within one scan cycle. However, the output terminal of an output module outputs an OUT result (ON or OFF) just before I/O processing.

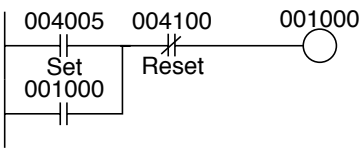


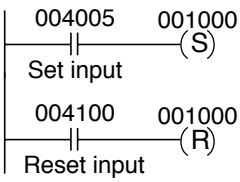
Unable to monitor at 1), 2), and 3).

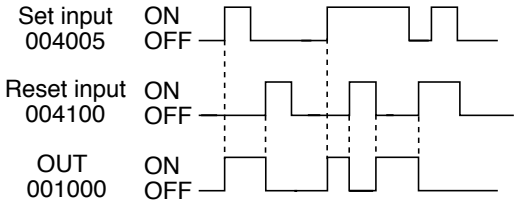
Even if the data memory turns ON/OFF several times while processing user program, the JW300 monitors only the result just before I/O processing.

- Using SET and RST instructions as a pair will allow simplification of self-latch circuit, etc.

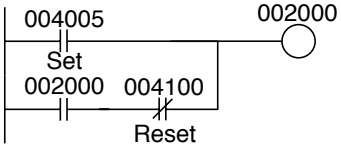
Self-latch circuit with reset priority

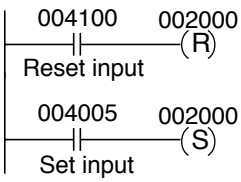


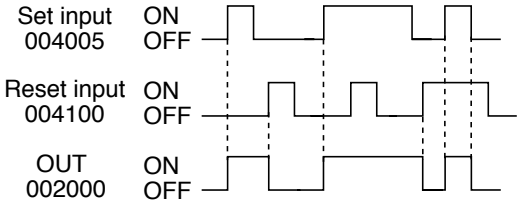




Self-latch circuit with set priority







PUSH **POP** **MRD**

Instruction word	Function
PUSH	Stores the accumulator (ACC) and stack contents to the protected area in the internal memory area.
POP	Recovers the saved accumulator (ACC) and stack contents using PUSH instruction from the memory area. After this operation, clears the contents stored by the PUSH instruction.
MRD	Temporarily reads the saved accumulator (ACC) and stack contents using PUSH instruction from the memory area.

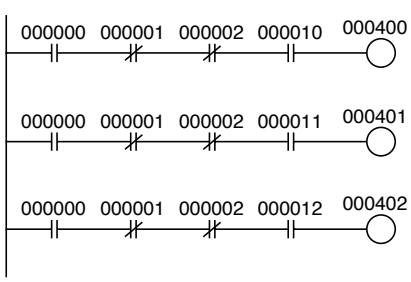
- PUSH instruction shall be less than 64 in one circuit.
- The number of times used shall be the same for PUSH and POP.
  After PUSH instruction, enter POP instruction.
- MRD instruction reads only the area stored just before.

● Use example

=> Page 8-30 to 8-31

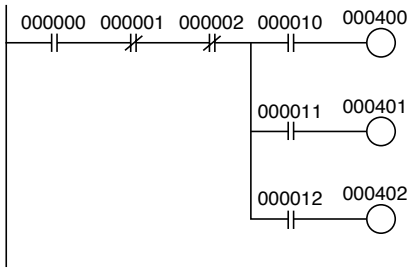
● Use example 1

- When to not use PUSH, POP, and MRD instructions




```
STR  000000
AND NOT  000001
AND NOT  000002
AND  000010
OUT  000400
STR  000000
AND NOT  000001
AND NOT  000002
AND  000011
OUT  000401
STR  000000
AND NOT  000001
AND NOT  000002
AND  000012
OUT  000402
```

- When using PUSH, POP, and MRD instructions




```
STR  000000
AND NOT  000001
AND NOT  000002
PUSH
AND  000010
OUT  000400
MRD
AND  000011
OUT  000401
POP
AND  000012
OUT  000402
```

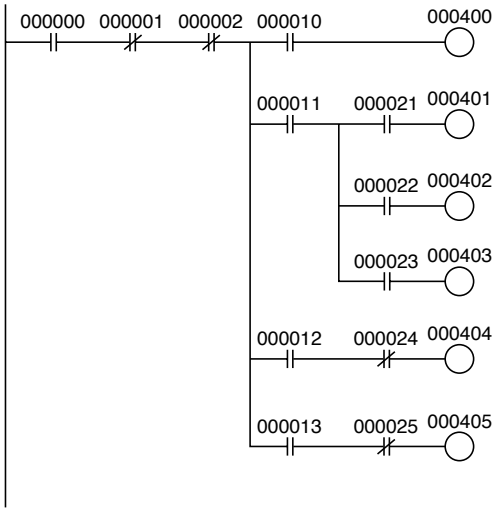
「PUSH」
Make a protected area for the contents of ACC and stack.

「MRD」
Reads the data in the protected area taken from the ACC and stack contents.

「POP」
Recovers the data in the protected area taken from the ACC and stack contents to the previous conditions.

Instruction	Accumulator ACC	Internal memory area ACC_1
STR 000000	000000	
AND NOT 000001	000000 000001	
AND NOT 000002	000000 000001 000002	
PUSH	000000 000001 000002	000000 000001 000002
AND 000010	000000 000001 000002 000010	000000 000001 000002
OUT 000400	000000 000001 000002 000010	000000 000001 000002
MRD	000000 000001 000002	000000 000001 000002
AND 000011	000000 000001 000002 000011	000000 000001 000002
OUT 000401	000000 000001 000002 000011	000000 000001 000002
POP	000000 000001 000002	
AND 000012	000000 000001 000002 000012	
OUT 000402	000000 000001 000002 000012	

● Use example 2




```
STR  000000
AND NOT  000001
AND NOT  000002
PUSH
AND  000010
OUT  000400
MRD
AND NOT  000011
PUSH
AND  000021
OUT  000401
MRD
AND  000022
OUT  000402
POP
AND  000023
OUT  000403
MRD
AND  000012
AND NOT  000024
OUT  000404
POP
AND  000013
AND NOT  000025
OUT  000405
```

Instruction	Accumulator ACC	Internal area	
		ACC_1	ACC_2
STR 000000	000000		
AND NOT 000001	000000 000001		
AND NOT 000002	000000 000001 000002		
PUSH	000000 000001 000002	000000 000001 000002	
AND 000010	000000 000001 000002 000010	000000 000001 000002	
OUT 000400	000000 000001 000002 000010	000000 000001 000002	
MRD	000000 000001 000002	000000 000001 000002	
AND 000011	000000 000001 000002 000011	000000 000001 000002	
PUSH	000000 000001 000002 000011	000000 000001 000002 000011	000000 000001 000002
AND 000021	000000 000001 000002 000011 00021	000000 000001 000002 000011	000000 000001 000002
OUT 000401	000000 000001 000002 000011 00021	000000 000001 000002 000011	000000 000001 000002
MRD	000000 000001 000002 000011	000000 000001 000002 000011	000000 000001 000002
AND 000022	000000 000001 000002 000011 00022	000000 000001 000002 000011	000000 000001 000002
OUT 000402	000000 000001 000002 000011 00022	000000 000001 000002 000011	000000 000001 000002
POP	000000 000001 000002 000011	000000 000001 000002	
AND 000023	000000 000001 000002 000011 00023	000000 000001 000002	
OUT 000403	000000 000001 000002 000011 00023	000000 000001 000002	
MRD	000000 000001 000002	000000 000001 000002	
AND 000012	000000 000001 000002 000012	000000 000001 000002	
AND NOT000024	000000 000001 000002 000012 00024	000000 000001 000002	
OUT 000404	000000 000001 000002 000012 00024	000000 000001 000002	
POP	000000 000001 000002		
AND 000013	000000 000001 000002 000013		
AND NOT 000025	000000 000001 000002 000013 00025		
OUT 000405	000000 000001 000002 000013 00025		

Chapter 9 Description of application instructions

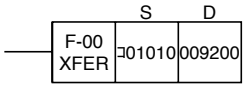
9-1 Application instruction hints and tips

[1] Source and destination

Data processing instruction, such as data transfer and calculation, is handled in terms of a byte or word.

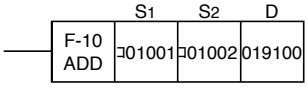
The register that contains the data before the operation is called source (S) and the register that contains the result is called destination (D).

Example 1



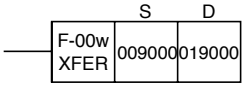
- The contents of ⊐01010(S) are transferred to 009200(D).

Example 2



- The contents of ⊐01001(S_1) are added with the contents of ⊐01002(S_2) and its result is stored in 019100(D).

Example 3



- The contents of 009000(S) and 009001(S+1) are transferred to 019000(D) and 019001(D+1).

- Make sure to set even address to source and destination for word processing instructions (instructions with "w") and two word processing instructions (instructions with "d").
- For instructions (F-70 etc.) that operate with data memory of 2 bytes or more, and if the source (s) or destination (D) exceeds each area, the JW300 allocates the next file address as shown in the table below. => See page 2-5.

S, D	S+1, D+1
⊐01577	TMR/CNT contact area (File address 00001600) *
⊐07577	TMR/CNT contact area (File address 00035600) *
b01777	009000
b03777	⊐02000
b37777	109000
009777	019000
019777	029000
⋮	⋮
099777	E0000

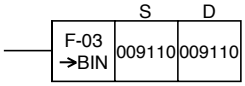
S, D	S+1, D+1
E0777	E1000
⋮	⋮
E7777	b02000
109777	119000
119777	129000
⋮	⋮
199777	209000
209777	219777
⋮	⋮
299777	309000
309777	319777
⋮	⋮
389777	Z000

Note: Set the source and destination so that S+1 and D+1 do not enter the following areas.

1. Contact point area of TMR/CNT => See * above.
   File addresses 00001600 to $00001777_{(8)}$, 00035600 to $00035777_{(8)}$, 00101400 to $00101777_{(8)}$ (JW-32*CU), 00140400 to $00143777_{(8)}$ (JW-33*CU to 362CU).
2. Out of the data memory area of each model (JW-3**CU)
   [Ex. 5] File address $00074000_{(8)}$ or after when JW-31*CU is used.
3. When JW-33*CU to 362CU is used, entering from file address $00177777_{(8)}$ to $00200000_{(8)}$ (File register: FILE 1) is prohibited.

● The contents of the register on the source side is not affected after the operation. Though it is possible to use the same register for the source (and destination), the contents of the source (destination, in other words) will be affected depending on the instruction used.

[Example 4]



- The contents of 009110(S) are converted into BCD 2 digits and stored in 009110(D).

● The special relay area ⊐00730 to ⊐00737 is an area that is written by the CPU. Do not specify this area as the destination (D).

■ Use range of S and D

Use range of the source (S) and destination (D) are described in "A to C, E to H, J, and K" in the description for the application instruction. For the range of each, see below.

(1) Use range A

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
ㅋ00000 to ㅋ01577 ㅋ02000 to ㅋ07577					
	ㅋ10000 to ㅋ15377	ㅋ10000 to ㅋ54377			
b00000 to b01777 b02000 to b03777					
	b04000 to b07777	b04000 to b37777			
009000 to 099777 E00000 to E07777 109000 to 199777 209000 to 299777 309000 to 389777 Z000 to Z377 ———— *					
File register (FILE 1) ⇒	00000000 to 00077777	00000000 to 00377777	00000000 to 01777777	00000000 to 07777777	00000000 to 37777777

* Z000 to Z377 can be used only for word instruction (F □ w).

● When specifying an indirect address

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@ㅋ00000 to @ㅋ01574 @ㅋ02000 to @ㅋ07574					
	@ㅋ10000 to @ㅋ15374	@ㅋ10000 to @ㅋ54374			
@b00000 to @b01774 @b02000 to @b03774					
	@b04000 to @b07774	@b04000 to b37774			
@009000 to @099774 @E00000 to @E07774 @109000 to @199774 @209000 to @299774 @309000 to @389774					
File register ⇒ (FILE 1)	@00000000 to @00077774	@00000000 to @00377774	@00000000 to @01777774	@00000000 to @07777774	@00000000 to @37777774

[Applicable instruction words]

F-00 (S, D), F-01 (D), F-02 (D1, D2), F-03 (S, D), F-04 (S, D), etc.

(2) Use range B

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
□00000 to □01576 □02000 to □07576					
	□10000 to □15376	□10000 to □54376			
b00000 to b01776 b02000 to b03776					
	b04000 to b07776	b04000 to b37776			
009000 to 099776 E00000 to E07776 109000 to 199776 209000 to 299776 309000 to 389776 Z000 to Z377 ——— *					
File register (FILE 1) ⇒	00000000 to 00077776	00000000 to 00377776	00000000 to 01777776	00000000 to 07777776	00000000 to 37777776

* Z000 to Z377 can be used only for word instruction (F □w).

● **When specifying an indirect address**

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@□00000 to @□01574 @□02000 to @□07574					
	@□10000 to @□15374	@□10000 to @□54374			
@b00000 to @b01774 @b02000 to @b03774					
	@b04000 to @b07774	@b04000 to b37774			
@009000 to @099774 @E00000 to @E07774 @109000 to @199774 @209000 to @299774 @309000 to @389774					
File register (FILE 1) ⇒	@00000000 to @00077774	@00000000 to @00377774	@00000000 to @01777774	@00000000 to @07777774	@00000000 to @37777774

[Applicable instruction words]

F-00w (S, D), F-01w (D), F-02w (D1, D2), F-03w (S, D), F-04w (S), etc.

(3) Use range C

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
ㅋ00000 to ㅋ01574 ㅋ02000 to ㅋ07574					
	ㅋ10000 to ㅋ15374	ㅋ10000 to ㅋ54374			
b00000 to b01774 b02000 to b03774					
	b04000 to b07774	b04000 to b37774			
009000 to 099774 E00000 to E07774 109000 to 199774 209000 to 299774 309000 to 389774 Z000 to Z376 ———— *					
File register (FILE 1) ⇒	00000000 to 00077774	00000000 to 00377774	00000000 to 01777774	00000000 to 07777774	00000000 to 37777774

* Z000 to Z376 can be used only for word instruction (F □ w).

● **When specifying an indirect address**

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@ㅋ00000 to @ㅋ01574 @ㅋ02000 to @ㅋ07574					
	@ㅋ10000 to @ㅋ15374	@ㅋ10000 to @ㅋ54374			
@b00000 to @b01774 @b02000 to @b03774					
	@b04000 to @b07774	@b04000 to b37774			
@009000 to @099774 @E00000 to @E07774 @109000 to @199774 @209000 to @299774 @309000 to @389774					
File register (FILE 1) ⇒	@00000000 to @00077774	@00000000 to @00377774	@00000000 to @01777774	@00000000 to @07777774	@00000000 to @37777774

[Applicable instruction words]

F-00d (S, D), F-02d (D1, D2), F-05w (S), F-06w (D), F-09d (S, D), etc.

(4) Use range E

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
□00000 to □01575 □02000 to □07575					
	□10000 to □15375	□10000 to □54375			
b00000 to b01775 b02000 to b03775					
	b04000 to b07775	b04000 to b37775			
009000 to 099775 E00000 to E07775 109000 to 199775 209000 to 299775 309000 to 389775 Z000 to Z376 ———— *					
File register (FILE 1) ⇒	00000000 to 00077775	00000000 to 00377775	00000000 to 01777775	00000000 to 07777775	00000000 to 37777775

* Z000 to Z376 can be used only for word instruction (F□w).

● **When specifying an indirect address**

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@□00000 to @□01574 @□02000 to @□07574					
	@□10000 to @□15374	@□10000 to @□54374			
@b00000 to @b01774 @b02000 to @b03774					
	@b04000 to @b07774	@b04000 to b37774			
@009000 to @099774 @E00000 to @E07774 @109000 to @199774 @209000 to @299774 @309000 to @389774					
File register (FILE 1) ⇒	@00000000 to @00077774	@00000000 to @00377774	@00000000 to @01777774	@00000000 to @07777774	@00000000 to @37777774

[Applicable instruction words]

F-04w (S, D), F-16 (D), Fc16 (D), F-22 (S), F-23 (S), F-24 (S),etc.

(5) Use range F

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
ㅋ00000, 00400, ㅋ01000, ㅋ01400 ㅋ02000, ㅋ02400, ㅋ03000, - - - - - -, ㅋ07000, ㅋ07400					
	@ㅋ10000, @ㅋ10400, ⋮ @ㅋ15000	ㅋ10000, ㅋ10400, ㅋ11000, - - - - - , ㅋ53400, ㅋ54000			
b00000, b00400, b01000, b01400 b02000, b02400, b03000, b03400					
	b04000, b04400, ⋮ b07400	b04000, b04400, b05000, - - - - - , b37000, b37400			
009000, 009400, - - - - - , 099000, 099400 E00000, E00400, - - - - - , E07000, E07400 109000, 109400, - - - - - , 199000, 199400 209000, 209400, - - - - - , 299000, 299400 309000, 309400, - - - - - , 389000, 389400 Z000, Z200 ———— *					
File register (FILE 1) ⇒	00000000, 00000400, 00001000, ⋮ 00077400	00000000, 00000400, 00001000, ⋮ 00377400	00000000, 00000400, 00001000, ⋮ 01777400	00000000, 00000400, 00001000, ⋮ 01777400	00000000, 00000400, 00001000, ⋮ 00377400

* Z000 to Z200 can be used only for word instruction (F □ w).

● **When specifying an indirect address**

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@ㅋ00000, @ㅋ00400, @ㅋ01000, @ㅋ01400 @ㅋ02000, @ㅋ02400, @ㅋ03000, - - - - - , @ㅋ07000, @ㅋ07400					
	@ㅋ10000, @ㅋ10400, ⋮ @ㅋ15000	@ㅋ10000, @ㅋ10400, @ㅋ11000, - - - - - , @ㅋ53400, @ㅋ54000			
@b00000, @b00400, @b01000, @b01400 @b02000, @b02400, @b03000, @b03400					
	@b04000, @b04400, ⋮ @b07400	@b04000, @b04400, @b05000、 - - - - - , @b37000, @b37400			
@009000, @ㅋ009400, - - - - - , @ㅋ099000, @ㅋ099400 @E00000, @ㅋE00400, - - - - - , @ㅋE07000, @ㅋE07400 @109000, @ㅋ109400, - - - - - , @ㅋ199000, @ㅋ199400 @209000, @ㅋ209400, - - - - - , @ㅋ299000, @ㅋ299400 @309000, @ㅋ309400, - - - - - , @ㅋ389000, @ㅋ389400					
File register (FILE 1) ⇒	@00000000, @00000400, @00001000, ⋮ @ 00077400	@00000000, @00000400, @00001000, ⋮ @00377400	@00000000, @00000400, @00001000, ⋮ @01777400	@00000000, @00000400, @00001000, ⋮ @07777400	@00000000, @00000400, @00001000, , ⋮ @37777400

[Applicable instruction words]

F-05 (D), F-05w (D), F-06 (S), F-06w (S)

(6) Use range G

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
ㅋ00000 to ㅋ01570 ㅋ02000 to ㅋ07570					
	ㅋ10000 to ㅋ15370	ㅋ10000 to ㅋ54370			
b00000 to b01770 b02000 to b03770					
	b04000 to b07770	b04000 to b37770			
009000 to 099770 E00000 to E07770 109000 to 199770 209000 to 299770 309000 to 389770					
File register (FILE 1) ⇒	00000000 to 00077770	00000000 to 00377770	00000000 to 01777770	00000000 to 07777770	00000000 to 37777770

● **When specifying an indirect address**

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@ㅋ00000 to @ㅋ01574 @ㅋ02000 to @ㅋ07574					
	@ㅋ10000 to @ㅋ15374	@ㅋ10000 to @ㅋ54374			
@b00000 to @b01774 @b02000 to @b03774					
	@b04000 to @b07774	@b04000 to b37774			
@009000 to @099774 @E00000 to @E07774 @109000 to @199774 @209000 to @299774 @309000 to @389774					
File register (FILE 1) ⇒	@00000000 to @00077774	@00000000 to @00377774	@00000000 to @01777774	@00000000 to @07777774	@00000000 to @37777774

[Applicable instruction words]

F-15d (D), F-16d (D), Fc16d (D), F-28 (S,D), etc.

(7) Use range H

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
ㅋ00000 to ㅋ01572 ㅋ02000 to ㅋ07572					
	ㅋ10000 to ㅋ15372	ㅋ10000 to ㅋ54372			
b00000 to b01772 b02000 to b03772					
	b04000 to b07772	b04000 to b37772			
009000 to 099772 E00000 to E07772 109000 to 199772 209000 to 299772 309000 to 389772					
File register (FILE 1) ⇒	00000000 to 00077772	00000000 to 00377772	00000000 to 01777772	00000000 to 07777772	00000000 to 37777772

● **When specifying an indirect address**

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@ㅋ00000 to @ㅋ01574 @ㅋ02000 to @ㅋ07574					
	@ㅋ10000 to @ㅋ15374	@ㅋ10000 to @ㅋ54374			
@b00000 to @b01774 @b02000 to @b03774					
	@b04000 to @b07774	@b04000 to b37774			
@009000 to @099774 @E00000 to @E07774 @109000 to @199774 @209000 to @299774 @309000 to @389774					
File register (FILE 1) ⇒	@00000000 to @00077774	@00000000 to @00377774	@00000000 to @01777774	@00000000 to @07777774	@00000000 to @37777774

[Applicable instruction words]

F-05d (S), F-116 (D), F-154(D)

(8) Use range J

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
J00000 to J01200 J02000 to J07200					
	J10000 to J15000	J10000 to J54000			
b00000 to b01400 b02000 to b03400					
	b04000 to b07400	b04000 to b37400			
009000 to 099400 E00000 to E07400 109000 to 199400 209000 to 299400 309000 to 389400					
File register (FILE 1) ⇒	00000000 to 00077400	00000000 to 00377400	00000000 to 01777400	00000000 to 07777400	00000000 to 37777400

● **When specifying an indirect address**

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@J00000 to @J01574 @J02000 to @J07574					
	@J10000 to @J15374	@J10000 to @J54374			
@b00000 to @b01774 @b02000 to @b03774					
	@b04000 to @b07774	@b04000 to b37774			
@009000 to @099774 @E00000 to @E07774 @109000 to @199774 @209000 to @299774 @309000 to @389774					
File register (FILE 1) ⇒	@00000000 to @00077774	@00000000 to @00377774	@00000000 to @01777774	@00000000 to @07777774	@00000000 to @37777774

[Applicable instruction words]

F-176 (D)

(9) Use range K

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
000000 to 015777 020000 to 075777	000000 to 015777 020000 to 075777 100000 to 153777	000000 to 015777 020000 to 075777 100000 to 543777			

[Applicable instruction words]

F-32 (OUT), F-33 (OUT), F-34 (BIT), F-35 (BIT)

[2] When specifying an indirect address

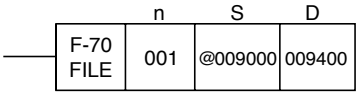
When the source (S) or destination (D) is specified using an indirect address, the file address of 3 bytes starting from the top address "file N, address n" specified in the indirect address executes operation, not the specified byte address itself. As for "file N and address n," see the next page.

File N: 00 to $80_{(H)}$

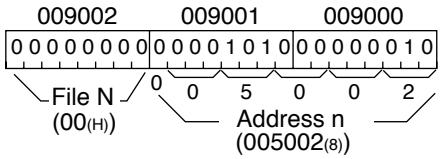
Address n: 000000 to $177777_{(8)}$

To specify an indirect address, put an "@" (at mark) in front of the byte address.

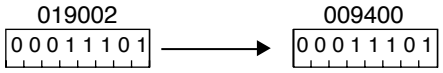
[Example]



- The content of file addresses 009000, 009001, and 009002 specified by an indirect address is transferred to 009400.



Since file address $005002_{(8)}$ of file $00_{(H)}$ in the example above is $00005002_{(8)}$ (register 019002), the result is that @009000 indicates 019002.

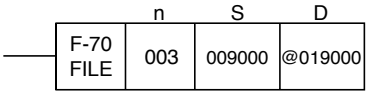


- Use an even address to specify an indirect address. If an odd address is used, the address will be decrement minus one so that the same action as the odd address will be executed.

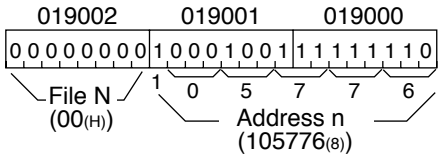
(When @009003 is set, it is treated as @009002.)

- Addresses to be specified by an indirect address specification shall be within the rated data memory area of each model (JW-3**CU). If the addresses are out of the range, the JW300 does not operate.

[Prohibited example of JW-32*CU]



- The content of 3 bytes starting from 009000 shall be transferred to 3-bytes starting from a file address specified by an indirect address.



With the example above, address $105776_{(8)}$ of file $00_{(H)}$ is file address $00105776_{(8)}$. After the operation, they are transferred to 3 bytes of file address 00105776 to $00106000_{(8)}$. However, $00106000_{(8)}$ is out of the setting range (prohibited area).

- Do not specify contact point areas of TMR/CNT (file address 00001600 to $00001777_{(8)}$ with an indirect address.
- With the word processing instruction and double word processing instruction, specify even addresses for indirect specification file addresses. Setting an odd address will automatically subtract "1" and make it into an even address.
- With control modules JW-33*CU, JW-34*CU, JW-352CU, and JW-362CU, when the specified address exceeds the last address $177777_{(8)}$ of file $00_{(H)}$ they will not operate. From file $01_{(H)}$, they will continue to the top address of the next file number, and execute operation.

■ **Indirect address "file N, address n"**

"file N, address n" that is set by an indirect address specification and application instruction (F-100, F-101) has the following relation with the file address.

[Ex.] File address $03100000_{(8)}$ will be n = $100000_{(8)}$ of file = $0C_{(H)}$.

File address (8)	fileN		n(8)	Capacity (byte)	
	Decimal	Hex.			
00000000 to 00073777	0	00	000000 to 073777	30K	64K
00074000 to 00105777			074000 to 105777	5K	
00106000 to 00177777			106000 to 177777	29K	
00200000 to 00277777	1	01	000000 to 077777	32K	64K
00300000 to 00377777			100000 to 177777	32K	
00400000 to 00577777	2	02	000000 to 177777	64K	
00600000 to 00777777	3	03	000000 to 177777	64K	
01000000 to 01177777	4	04	000000 to 177777	64K	
01200000 to 01377777	5	05	000000 to 177777	64K	
01400000 to 01577777	6	06	000000 to 177777	64K	
01600000 to 01777777	7	07	000000 to 177777	64K	
02000000 to 02177777	8	08	000000 to 177777	64K	
02200000 to 02377777	9	09	000000 to 177777	64K	
02400000 to 02577777	10	0A	000000 to 177777	64K	
02600000 to 02777777	11	0B	000000 to 177777	64K	
03000000 to 03177777	12	0C	000000 to 177777	64K	
03200000 to 03377777	13	0D	000000 to 177777	64K	
03400000 to 03577777	14	0E	000000 to 177777	64K	
03600000 to 03777777	15	0F	000000 to 177777	64K	
04000000 to 04177777	16	10	000000 to 177777	64K	
04200000 to 04377777	17	11	000000 to 177777	64K	
04400000 to 04577777	18	12	000000 to 177777	64K	
04600000 to 04777777	19	13	000000 to 177777	64K	
05000000 to 05177777	20	14	000000 to 177777	64K	
05200000 to 05377777	21	15	000000 to 177777	64K	
05400000 to 05577777	22	16	000000 to 177777	64K	
05600000 to 05777777	23	17	000000 to 177777	64K	
06000000 to 06177777	24	18	000000 to 177777	64K	
06200000 to 06377777	25	19	000000 to 177777	64K	
06400000 to 06577777	26	1A	000000 to 177777	64K	
06600000 to 06777777	27	1B	000000 to 177777	64K	
07000000 to 07177777	28	1C	000000 to 177777	64K	
07200000 to 07377777	29	1D	000000 to 177777	64K	
07400000 to 07577777	30	1E	000000 to 177777	64K	
07600000 to 07777777	31	1F	000000 to 177777	64K	
10000000 to 10177777	32	20	000000 to 177777	64K	

1) JW-31*CU 2) JW-32*CU 3) JW-33*CU 4) JW-34*CU 5) JW-352CU 6) JW-362CU

Next page

From previous page

File address (8)	fileN		n(8)	Capacity (byte)
	Decimal	Hex.		
10200000 to 10377777	33	21	000000 to 177777	64K
10400000 to 10577777	34	22	000000 to 177777	64K
10600000 to 10777777	35	23	000000 to 177777	64K
11000000 to 11177777	36	24	000000 to 177777	64K
11200000 to 11377777	37	25	000000 to 177777	64K
11400000 to 11577777	38	26	000000 to 177777	64K
11600000 to 11777777	39	27	000000 to 177777	64K
12000000 to 12177777	40	28	000000 to 177777	64K
12200000 to 12377777	41	29	000000 to 177777	64K
12400000 to 12577777	42	2A	000000 to 177777	64K
12600000 to 12777777	43	2B	000000 to 177777	64K
13000000 to 13177777	44	2C	000000 to 177777	64K
13200000 to 13377777	45	2D	000000 to 177777	64K
13400000 to 13577777	46	2E	000000 to 177777	64K
13600000 to 13777777	47	2F	000000 to 177777	64K
14000000 to 14177777	48	30	000000 to 177777	64K
14200000 to 14377777	49	31	000000 to 177777	64K
14400000 to 14577777	50	32	000000 to 177777	64K
14600000 to 14777777	51	33	000000 to 177777	64K
15000000 to 15177777	52	34	000000 to 177777	64K
15200000 to 15377777	53	35	000000 to 177777	64K
15400000 to 15577777	54	36	000000 to 177777	64K
15600000 to 15777777	55	37	000000 to 177777	64K
16000000 to 16177777	56	38	000000 to 177777	64K
16200000 to 16377777	57	39	000000 to 177777	64K
16400000 to 16577777	58	3A	000000 to 177777	64K
16600000 to 16777777	59	3B	000000 to 177777	64K
17000000 to 17177777	60	3C	000000 to 177777	64K
17200000 to 17377777	61	3D	000000 to 177777	64K
17400000 to 17577777	62	3E	000000 to 177777	64K
17600000 to 17777777	63	3F	000000 to 177777	64K
20000000 to 20177777	64	40	000000 to 177777	64K
20200000 to 20377777	65	41	000000 to 177777	64K
20400000 to 20577777	66	42	000000 to 177777	64K
20600000 to 20777777	67	43	000000 to 177777	64K
21000000 to 21177777	68	44	000000 to 177777	64K
21200000 to 21377777	69	45	000000 to 177777	64K
21400000 to 21577777	70	46	000000 to 177777	64K
21500000 to 21777777	71	47	000000 to 177777	64K
22000000 to 22177777	72	48	000000 to 177777	64K
22200000 to 22377777	73	49	000000 to 177777	64K
22400000 to 22577777	74	4A	000000 to 177777	64K

6)

Next page

File address (8)	fileN		n(8)	Capacity (byte)
	Decimal	Hex.		
22600000 to 22777777	75	4B	000000 to 177777	64K
23000000 to 23177777	76	4C	000000 to 177777	64K
23200000 to 23377777	77	4D	000000 to 177777	64K
23400000 to 23577777	78	4E	000000 to 177777	64K
23600000 to 23777777	79	4F	000000 to 177777	64K
24000000 to 24177777	80	50	000000 to 177777	64K
24200000 to 24377777	81	51	000000 to 177777	64K
24400000 to 24577777	82	52	000000 to 177777	64K
24600000 to 24777777	83	53	000000 to 177777	64K
25000000 to 25177777	84	54	000000 to 177777	64K
25200000 to 25377777	85	55	000000 to 177777	64K
25400000 to 25577777	86	56	000000 to 177777	64K
25600000 to 25777777	87	57	000000 to 177777	64K
26000000 to 26177777	88	58	000000 to 177777	64K
26200000 to 26377777	89	59	000000 to 177777	64K
26400000 to 26577777	90	5A	000000 to 177777	64K
26600000 to 26777777	91	5B	000000 to 177777	64K
27000000 to 27177777	92	5C	000000 to 177777	64K
27200000 to 27377777	93	5D	000000 to 177777	64K
27400000 to 27577777	94	5E	000000 to 177777	64K
27600000 to 27777777	95	5F	000000 to 177777	64K
30000000 to 30177777	96	60	000000 to 177777	64K
30200000 to 30377777	97	61	000000 to 177777	64K
30400000 to 30577777	98	62	000000 to 177777	64K
30600000 to 30777777	99	63	000000 to 177777	64K
31000000 to 31177777	100	64	000000 to 177777	64K
31200000 to 31377777	101	65	000000 to 177777	64K
31400000 to 31577777	102	66	000000 to 177777	64K
31600000 to 31777777	103	67	000000 to 177777	64K
32000000 to 32177777	104	68	000000 to 177777	64K
32200000 to 32377777	105	69	000000 to 177777	64K
32400000 to 32577777	106	6A	000000 to 177777	64K
32600000 to 32777777	107	6B	000000 to 177777	64K
33000000 to 33177777	108	6C	000000 to 177777	64K
33200000 to 33377777	109	6D	000000 to 177777	64K
33400000 to 33577777	110	6E	000000 to 177777	64K
33600000 to 33777777	111	6F	000000 to 177777	64K
34000000 to 34177777	112	70	000000 to 177777	64K
34200000 to 34377777	113	71	000000 to 177777	64K
34400000 to 34577777	114	72	000000 to 177777	64K
34600000 to 34777777	115	73	000000 to 177777	64K
35000000 to 35177777	116	74	000000 to 177777	64K

From previous page

6)

Next page

File address (8)	fileN		n(8)	Capacity (byte)
	Decimal	Hex.		
35200000 to 35377777	117	75	000000 to 177777	64K
35400000 to 35577777	118	76	000000 to 177777	64K
35600000 to 35777777	119	77	000000 to 177777	64K
36000000 to 36177777	120	78	000000 to 177777	64K
36200000 to 36377777	121	79	000000 to 177777	64K
36400000 to 36577777	122	7A	000000 to 177777	64K
36600000 to 36777777	123	7B	000000 to 177777	64K
37000000 to 37177777	124	7C	000000 to 177777	64K
37200000 to 37377777	125	7D	000000 to 177777	64K
37400000 to 37577777	126	7E	000000 to 177777	64K
37600000 to 37777777	127	7F	000000 to 177777	64K
40000000 to 40177777	128	80	000000 to 177777	64K

From previous page

6)

9-2 Index modification function

When the index modification function is applied to relays and registers that are directly specified by the basic instructions and application instructions using index register Z000 to Z377 (hereafter called to "Z_xxx"), the JW-300 operates with addresses added or subtracted by Z_xxx.

[1] Programming of index modification

For programming index modification using Z_xxx, "normal modification" and "auto modification" are available.

(1) Normal modification

Program as "relay/register, Z_xxx," and when it executes its instruction, the JW300 operates with an address adding Z_xxx to the relay or register address.
Whether addition or subtraction is determined by the upper most bit (sign). When this bit is 0, "add", and when this bit is 1, "subtract."

● **Detail of Z_xxx**

The Z_xxx is one register and has two bytes. Set it with binary notation with sign (-77777 to +777777(8)). The upper most bit is sign (+/-).

Detail of Z_xxx

Binary with sign	Octal	Decimal
0111111111111111	+77777	+32767
0111111111111110	+77776	+32766
⋮		
0000000011111111	+377	+255
⋮		
0000000000000010	+2	+2
0000000000000001	+1	+1
0000000000000000	0	0
1111111111111111	-1	-1
1111111111111110	-2	-2
⋮		
1111111100000000	-400	-256
⋮		
1000000000000001	-77776	-32766
1000000000000000	-77777	-32767

For negative value, take complement of 2.

Sign bit (1= "-", 0 = "+")

- The "complement of 2" means a value to invert binary data (0 to 1 and 1 to 0) and add "1."

[Ex.] When 11111111 00000000,

11111111 00000000
↓ Bit inversion
00000000 11111111
↓ Add 1
00000001 00000000
↓ Convert to octal notation (Sign: -)
$-400_{(8)}$

(2) Auto modification

When "relay/register, Z_xxx + yy" or "relay/register, Z_xxx-yy" is programmed, after executing the same operation as the "normal modification," the JW300 adds or subtracts specified value yy (0 to 255) on the Z_xxx content (shown above).

- When yy is added to a basic instruction, the JW300 adds or subtracts an address number regardless of the accumulator (ACC) content.
- When yy is added to an application instruction, the JW300 adds or subtracts an address number only when its instruction is executed. Whether addition or subtraction is determined by the upper most bit (sign). When this bit is 0, "add", and when this bit is 1, "subtract."

[2] Index modification application area

The Z_xxx can execute index modification for the areas of data memory, TMR/CNT number and label number shown in the table below.

Area			JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
Data memory	*1 Relay	1)	ㅋ00000 to ㅋ01577(000000 to 015777)					
		2)	ㅋ02000 to ㅋ07577(020000 to 075777)					*2
		3)		ㅋ10000 to ㅋ15377 (100000 to 153777)	ㅋ10000 to ㅋ54377(100000 to 543777)			
	TMR, CNT, contact	4)	T/C 00000 to 00777					
		5)	T/C 01000 to 01777					
		6)		T/C 02000 to 03777	T/C 02000 to 17777			
	TMR, *1 CNT, MD current value	7)	b00000 to b01777					
		8)	b02000 to b03777					
		9)		b04000 to b07777	b04000 to b37777			
	*1 Register	10)	009000 to E07777					
		11)	109000 to 389777					
TMR, CNT number		12)	00000 to 01777	00000 to 03777	00000 to 17777			
Label number		13)	LB0000 to LB1777					

*1: Index modification to indirect address applicable to relay, TMR/CNT/MD current value, and register.
*2: Numbers in ( ) parentheses are relay numbers.

Note: With the Index modification, JW300 adds and subtracts within the areas (1) to (13) of the table above.
- When the Z_xxx added/subtracted address number is out of the range, the JW300 does not operate it.
- With the "auto modification," if the specified value yy is repeatedly added/subtracted and if the address number will be out of the rated range, the JW300 operates within the range of the address number.

Remarks

- Do not apply the index modification to the special relays and index register Z-xxx.
- To apply the index modification to registers for application instructions having the double length calculation function, make sure to program so that the addresses after adding/subtracting shall be odd number address.
- The index modification cannot be applied to the constants of the application instructions.
- For the basic instructions having no set address such as AND STR and PUSH, the index modification cannot be applied.
- For labels that applied the index modification, if a label to jump does not exist, the JW300 does not operate.

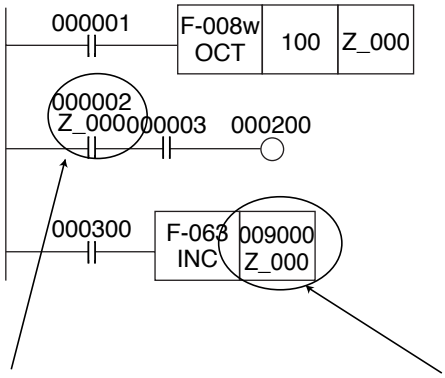
[3] Program examples of index modification

(1) Normal modification

Program "Z_xxx and relay": an address of the relay is added to or subtracted from the content of Z_xxx when executing this instruction.

Whether to add or subtract is determined by the uppermost bit (sign). When this bit is 0, it will "add." When this bit is 1, it will "subtract."

■ **Example of addition (normal modification)**




```
STR          000001
F-008w
                100
              Z_000
STR    000002, Z_000
AND          000003
OUT          000200
STR          000300
F-063
       009000, Z_000
```

Execute STR 000102, that adds Z_000(+100(8)) to relay number 000002.

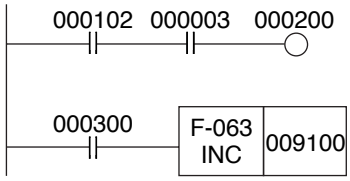
Store the calculated result to register 009100, with -000(+100(8)) added, to register 009000.

● **Detail of Z_000**

Binary value with sign	Octal
00000000 01000000 →	+100

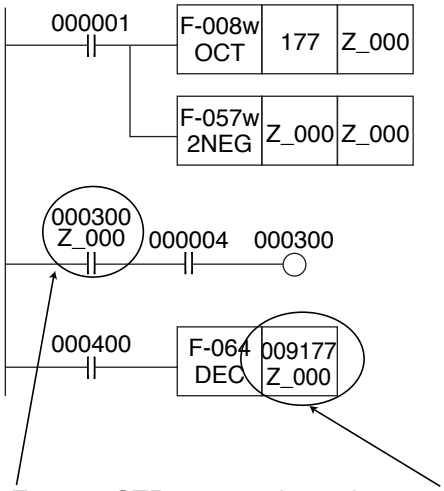
Sign bit (0 = +)

- With the example above, the JW300 operates as shown below.




```
STR   000002
AND   000003
OUT   000200
STR   000300
F-063
      009100
```

■ Example of subtraction (normal modification)




```
STR          000001
F-008w
                 177
             Z_000
F-057w
             Z_000
             Z_000
STR  000300, Z_000
AND          000004
OUT          000300
STR          000400
F-064
       009177, Z_000
```

Execute STR 000101 that subtracts Z_000(-177(8)) from relay number 000300.

Store the calculated result to register 009000, with Z_000(-177(8)) subtracted, to register 009177.

● Detail of Z_000

Binary value with sign	Octal
11111111 10000001	-177

Sign bit (1= "-")

For negative value, take complement of 2.

↓ Bit inversion

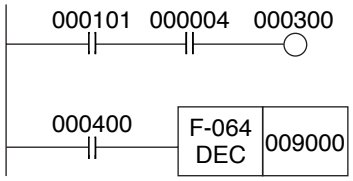
00000000 01111110

↓ Add 1

00000000 01111111

Convert to octal notation

- With the example above, the JW300 operates as shown below.



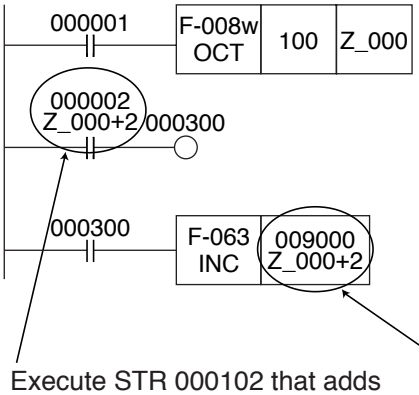

```
STR  000002
AND  000003
OUT  000200
STR  000300
F-064
     009000
```

(2) In case of auto modification

Program "relays, Z_xxx + yy" or "relay, Z_xxx - yy" after operation; the specified value is added to the Z_xxx content.

- With the basic instruction, yy is added/subtracted regardless of execute or not execute operation.
- With the application instruction, yy is added/subtracted when this instruction is executed.

■ **Example of addition (auto)**

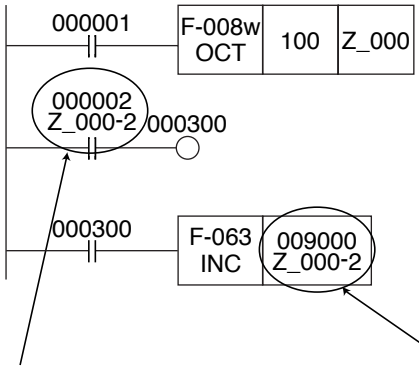



```
STR                    000001
F-008w
                          100
                        Z_000
STR       000002, Z_000+2
AND NOT              000003
STR                  000300
F-063
          009000, Z_000+2
```

Execute STR 000102 that adds Z_000(+$100_{(8)}$) to relay number 000002. After execution of the instruction, the content of Z_000 has 2 added and becomes +$102_{(8)}$.

Store the calculation contents to register 09102, that has Z_000(+$102_{(8)}$) added to register 009000. After executing the instruction, the content of Z_000 has 2 added and becomes +$104_{(4)}$.[Loew1]

■ **Example of subtraction (auto)**

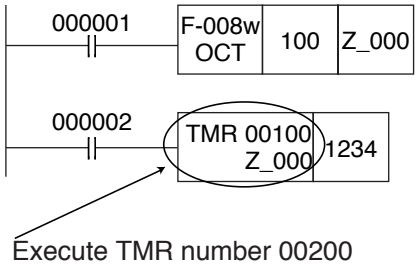



```
STR                    000001
F-008w
                          100
                        Z_000
STR       000002, Z_000-2
AND NOT              000003
STR                  000300
F-063
          009000, Z_000-2
```

Execute STR 000102 that adds Z_000(+$100_{(8)}$) to relay number 000002. After execution of the instruction, the content of Z_000 has 2 subtracted and becomes +$076_{(8)}$.

Store the calculation contents to register 009076, that has Z_000(+$076_{(8)}$) added to register 009076. After executing the instruction, the content of Z_000 has 2 subtracted and becomes +$074_{(8)}$.

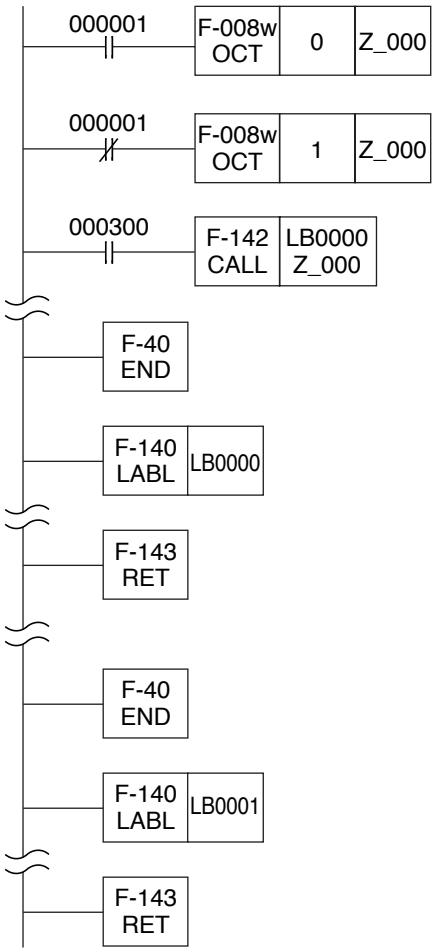
(3) Example to apply index modification to TMR/CNT number




```
STR          000001
F-008w
                 100
               Z_000
STR          000002
TMR  00100, Z_000
                1234
```

Execute TMR number 00200 that has Z_000(+$100_{(8)}$) added to TMR number 00100.

(4) Example to apply index modification to label number




```
STR              000001
F-008w
                    000
                  Z_000
STR NOT          000001
F-008w
                    001
                  Z_000
STR              000300
F-142
          LB0000, Z_000
    ⋮
F-40
F-140
                 LB0000
    ⋮
F-143
    ⋮
F-40
F-140
                 LB0000
    ⋮
F-143
```

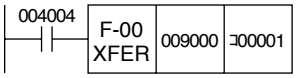
- When the relay 000001 is ON, the JW300 shifts to subroutine of label number LB0000. When relay 000001 is OFF, it shifts to subroutine of label number LB0001.

Chapter 10 Application instructions (F-00 to Fx14d)

F-00 XFER Transfers 1-byte data

Symbol	F-00 XFER | S | D	
Function	Transfers the contents of the register (1 byte) to the register D.	
Operation	S -> D	
S	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Contents of register S
	Flag	Unchanged

[Explanation]



Instruction	
STR	004004
F-00	
	009000
	⊐00001

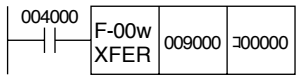
When the input condition 004004 changes from OFF to ON, the contents of the register 009000 are transferred to the register ⊐00001.

Resembled instructions: F-00w, F-00d, F-70, F-70w, F-70d, F-74, F-74w, F-74d, F-76, F-76w, and F-76d

F-00w XFER Transfers 1-word data

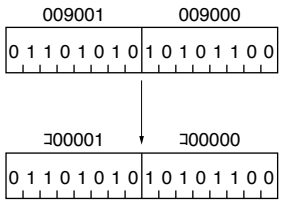
Symbol	F-00w XFER | S | D	
Function	Transfers the contents of the registers S, S+1 (1 word) to the register D.	
Operation	S, S+1 -> D, D+1	
S	Use range B *	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1	Unchanged
	D	Contents of register S
	D+1	Contents of register S+1
	Flag	Unchanged

[Explanation]



Instruction	
STR	004000
F-00w	
	009000
	⊐00000

When the input condition 004000 changes from OFF to ON, the contents of the registers 009000 and 009001 (1 word) are transferred to the registers ⊐00000 and ⊐00001.

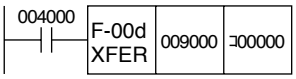


* Be sure to use even addresses for registers S and D.
(Odd address such as 019003 etc. are prohibited to use.)
Resembled instructions: F-00, F-00d, F-70, F-70w, F-70d, F-74, F-74w, F-74d, F-76, F-76w, and F-76d

F-00d XFER — Transfer 2-word data

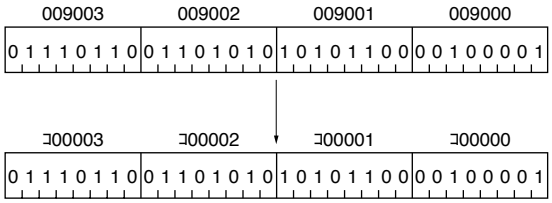
Symbol	F-00d XFER | S | D	
Function	Transfers the contents of the registers S to S+3 (2 words) to the registers D to D+3.	
Operation	S to S+3 -> D to D+3	
S	Use range C *	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S to S+3	Unchanged
	D to D+3	Contents of registers S to S+3
	Flag	Unchanged

[Explanation]



Instruction	
STR	004000
F-00d	
	009000
	□00000

When the input condition 004000 changes from OFF to ON, the contents of the registers 009000 to 009003 (2 words) are transferred to the registers □00000 to □00003.

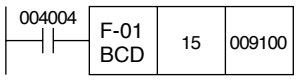


* Be sure to use even addresses for registers S and D.
(Odd address such as 019003 etc. are prohibited to use.)
Resembled instructions: F-00, F-00w, F-70, F-70w, F-70d, F-74, F-74w, F-74d, F-76, F-76w, and F-76d

F-01 BCD — Transfers BCD (2 digits) constant

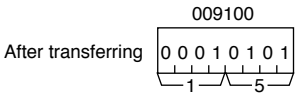
Symbol	F-01 BCD | n | D	
Function	Transfers a 2-digit BCD constant "n" to the register D.	
Operation	n -> D	
n	Use range 00 to 99	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D	n
	Flag	Unchanged

[Explanation]



Instruction	
STR	004001
F-01	
	15
	009100

When the input condition 004004 changes from OFF to ON, the BCD constant "15" is transferred to the register 009100.

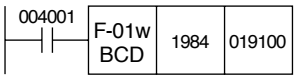


Resembled instructions: F-01w, F-01d, F-91

F-01w BCD — Transfers BCD (4 digits) constant

Symbol	F-01w BCD | n | D	
Function	Transfers a 4-digit BCD constant "n" to the registers D, D+1.	
Operation	n -> D, D+1	
n	Use range 0000 to 9999	
D	Use range B - Be sure to use even addresses for register D. (Odd address such as 019003 etc. are prohibited to use.)	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D, D+1	n
	Flag	Unchanged

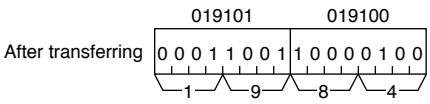
[Explanation]



Instruction	
STR	004001
F-01w	
	1984
	019100

When the input condition 004001 changes from OFF to ON, the BCD constant "1984" is transferred to the registers 019100 and 019101.

The following value is contained in the registers 019100 and 019101 after the transfer.

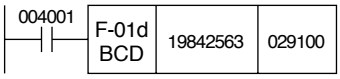


Resembled instructions: F-01, F-01d, F-91

F-01d BCD — Transfers BCD (8 digits) constant

Symbol	F-01d BCD | n | D	
Function	Transfers a 8 digits BCD constant "n" to the registers D to D+3.	
Operation	n -> D to D+3	
n	Use range 00000000 to 99999999	
D	Use range C - Be sure to use even addresses for register D. (Odd address such as 019003 etc. are prohibited to use.)	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D to D+3	n
	Flag	Unchanged

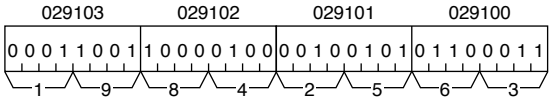
[Explanation]



Instruction	
STR	004001
F-01d	
	19842563
	029100

When the input condition 004001 changes from OFF to ON, the BCD constant "19842563" is transferred to the registers 029100 to 029103.

After transferring

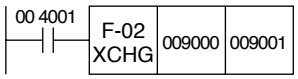


Resembled instructions: F-01, F-01w, F-91

F-02 XCHG — Exchange 1-byte data between registers (eXCHanGe)

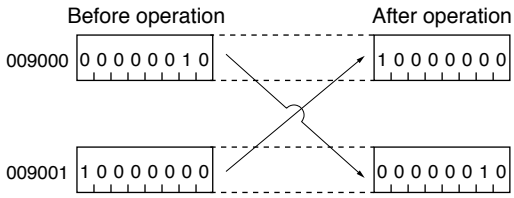
Symbol	F-02 XCHG | D_1 | D_2	
Function	The contents of the register D_1 are exchanged with the contents of the register D_2.	
Operation	D_1 <-> D_2	
D_1	Use range A	
D_2	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D_1	Contents of register D_2
	D_2	Contents of register D_1
	Flag	Unchanged

[Explanation]



Instruction	
STR	004001
F-02	
	009000
	009001

When the input condition 004001 changes from OFF to ON, the contents of the register 009000 are exchanged with the contents of the register 009001.

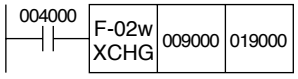


Resembled instructions: F-02w, F-02d, F-174

F-02w XCHG — Exchange 1-word data between registers (eXCHanGe)

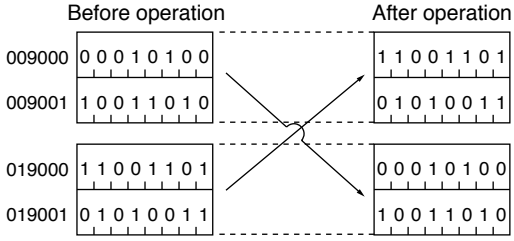
Symbol	F-02w XCHG | D_1 | D_2	
Function	The contents of the registers D_1, D_1+1 (1 word) are exchanged with the contents of the registers D_2, D_2+1.	
Operation	D, D_1+1 <-> D_2, D_2+1	
D_1	Use range B - Be sure to use even addresses for registers D_1.	
D_2	Use range B - Be sure to use even addresses for registers D_2.	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D1	Contents of the register D_2
	D1+1	Contents of register D_2+1
	D2	Contents of the register D_1
	D2+1	Contents of register D_1+1
	Flag	Unchanged

[Explanation]



Instruction	
STR	004000
F-02w	
	009000
	019000

When the input condition 004000 changes from OFF to ON, the contents of the registers 009000 and 009001 (1 word) are exchanged with the contents of the registers 019000 and 019001.



Resembled instructions: F-02, F-02d, F-174

F-02d
XCHG

Exchange 2-word data between registers (eXCHanGe)

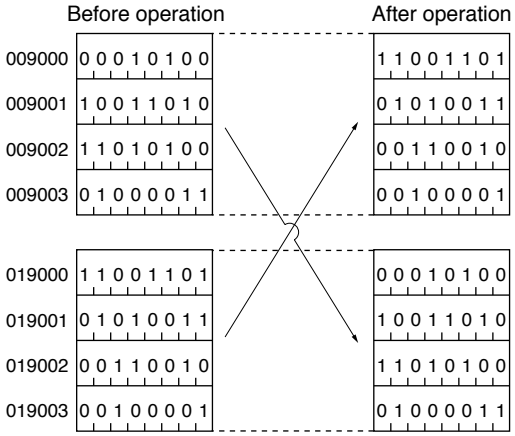
Symbol	F-02d XCHG | D_1 | D_2	
Function	The contents of the registers D_1 to D_1+3 (2 words) are exchanged with the contents of the registers D_2 to D_2+3 (2 words).	
Operation	D_1 to D_1+3 <-> D_2 to D_2+3	
D_1	Use range C - Be sure to use even addresses for registers D_1.	
D_2	Use range C - Be sure to use even addresses for registers D_2.	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D_1 to D_1+3	Contents of registers D_2 to D_2+3
	D_2 to D_2+3	Contents of registers D_1 to D_1+3
	Flag	Unchanged

[Explanation]

Instruction	
STR	004000
F-02d	
	009000
	019000

004000 — F-02d XCHG | 009000 | 019000

When the input condition 004000 changes from OFF to ON, the contents of the registers 009000 to 009003 (2 words) are exchanged with the contents of the registers 019000 to 019003 (2 words) .



Resembled instructions: F-02, F-02w, and F-174

F-03 →BIN Convert 2-digit BCD to 8-bit binary

Symbol		F-03 →BIN | S | D
Function		The contents of the register S (8 bits) are assumed as a BCD code, converts into a binary equivalent, then store the result in the register D.
Operation		S ->D
S		Use range A
D		Use range A
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S	Unchanged
	D	Calculation result - Unchanged when the contents of the register S are not a BCD code.
	Flag	Contents of register S: BCD code / Not BCD code; Zero 007357: 0; Carry 007356: 0; Error 007355: 0 / 1; Non-carry 007354: 1

Resembled instructions: F-03, F-03w, F-53, F-153

[Explanation]

Instruction	
STR	004006
F-03	
	009300
	009310

004006 | F-03 →BIN | 009300 | 009310

When the input condition 004006 changes from OFF to ON, the contents of the register 009300 (8 bits) are assumed as a BCD code, converts into a binary equivalent, and its result is transferred to the register 009310.
The contents of the register 009300 remain unchanged. If the contents of the register 009310 stay the same as before and the error flag is set to "1."

- Ex.1

009300: 1 0 0 0 0 1 0 1 (8 | 5) → 009310: 0 1 0 1 0 1 0 1

$2^6+2^4+2^2+2^0=64+16+4+1=85$

Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	0	0	0

- Ex.2

009300: 1 1 0 0 0 1 0 1 (C | 5) → 009310

The contents of 009310 stay the same.

Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	0	1	0

F-03w →BIN Convert 4-digit BCD to 16-bit binary

Symbol		F-03w →BIN | S | D
Function		The 2-byte BCD contents of the registers S, S+1 are converted into binary, and the result is stored in the 2-byte area of the registers D, D+1.
Operation		S_1, S+1 -> D, D+1
S		Use range B
D		Use range B
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S, S+1	Unchanged
	D	Result (0 to 255) — Unchanged when the contents of the registers S, S+1 are not a BCD code.
	D+1	Result (256 to 9999)
	Flag	Contents of registers S,S+1: BCD code / Not BCD code; Zero 007357: 0; Carry 007356: 0; Error 007355: 0 / 1; Non-carry 007354: 0

[Explanation]

Instruction	
STR	004001
F-03w	
	ㅋ01000
	019000

004001 | F-03w →BIN | ㅋ01000 | 019000

When the input condition 004001 changes from OFF to ON, the contents of 4-digit BCD in registers ㅋ01000 and ㅋ01001 are converted into a binary equivalent, and its result is transferred and stored in 2-byte area of registers 019000 and 019001.

Before operation → After operation

ㅋ01000: Tens Ones — 1 0 0 1 | 0 1 1 0 (9 | 6)
ㅋ01001: Thousands Hundreds — 0 1 0 0 | 0 0 0 0 (4 | 0)
BCD 4096

019000: 0 0 0 0 0 0 0 0 (2^7 … 2^0)
019001: 0 0 0 1 0 0 0 0 (2^{15} … 2^8)
BIN (Binary) 2^{12}=4096

- If the F-53 instruction is used for programming, the F-03w instruction displays the program during monitoring.
  Resembled instructions: F-03, F-03d, F-53, F-153

F-03d
→BIN

Convert 8-digit BCD to 32-bit binary

Symbol	F-03d →BIN | S | D					
Function	The 4-byte BCD 8-digit contents of the registers S to S+3 are converted into binary, and the result is stored in the 4-byte area of the registers D to D+3.					
Operation	S to S+3 -> D to D+3					
S	Use range C					
D	Use range C					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	S to S+3	Unchanged				
	D	0 to 255	Unchanged when the contents of the registers S to S+3 are not a BCD code.			
	D+1	256 to 65280				
	D+2	65536 to 16711680				
	D+3	16777216 to 99999999				
	Flag	Contents of registers S to S+3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		BCD code	0	0	0	0
		Not BCD code			1	

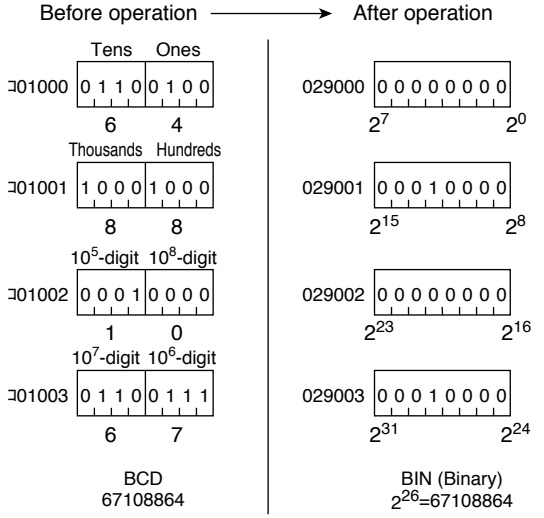
Resembled instructions: F-03, F-03w, F-53, F-153

[Explanation]

Instruction	
STR	004001
F-03d	
	⊐01000
	029000

004001 | F-03d →BIN | ⊐01000 | 029000

When the input condition 004001 changes from OFF to ON, the contents of 8-digit BCD in registers ⊐01000 to ⊐01003 are converted into a binary equivalent, and its result is transferred and stored in 4-byte area of registers 029000 to 029003.



F-04 -> BCD — Convert 8-bit binary to 2-digit BCD

Symbol	F-04 -> BCD | S | D	
Function	The contents of the register S (8 bits) are assumed as a binary code, converts into a BCD equivalent, then store the result in the register D.	
Operation	S -> D	
S	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Result
	Flag	Unchanged

Resembled instructions: F-04w, F-04d, F-54, F-154

[Explanation]

Instruction	
STR	004006
F-04	
	009320
	009330

004006 — F-04 -> BCD | 009320 | 009330

When the input condition 004006 changes from OFF to ON, the contents of the register 009320 (8 bits) are assumed as a binary code, converted into a BCD code, and its result is transferred to the register 009330. The contents of the register 009320 remain unchanged.
If the BCD number converted should exceed "100," the digit of hundreds will be ignored.

- Ex.1

009320: 0 0 1 0 1 1 0 1
$2^5 + 2^3 + 2^2 + 2^0$
$(32 + 8 + 4 + 1)$
=
45

009330: 0 1 0 0 | 0 1 0 1
4 | 5

- Ex.2

009320: 1 1 0 1 1 0 1 1
$2^7+2^6+2^4+2^3+2^1+2^0$
=
219

009330: 0 0 0 1 | 1 0 0 1
1 | 9

F-04w -> BCD — Convert 16-bit binary to 6-digit BCD

Symbol	F-04w -> BCD | S | D	
Function	The contents of 2-byte binary in the registers S, S+1 converted into a 6-digit BCD, and the result is stored in 3 bytes of the registers D, D+1, and D+2.	
Operation	S, S+1 ->D, D+1, D+2	
S	Use range B	
D	Use range E	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1	Unchanged
	D	Result (ones and tens)
	D+1	Result (hundreds and thousands)
	D+2	Result (ten thousands)
	Flag	Unchanged

[Explanation]

Instruction	
STR	004001
F-04w	
	⊐01000
	019000

004001 — F-04w -> BCD | ⊐01000 | 019000

When the input condition 004001 changes from OFF to ON, the contents of 2-byte binary in registers ⊐01000 and ⊐01001 are converted into a 6-digit BCD and stored in 3-byte area of registers 019000 to 019002.

Before operation → After operation

⊐01000: 0 0 0 0 0 0 0 0 (2^7 … 2^0)
⊐01001: 1 0 0 0 0 0 0 0 (2^{15} … 2^8)

BIN

019000: 0 1 1 0 | 1 0 0 0 (Tens | Ones) = 6 | 8
019001: 0 0 1 0 | 0 1 1 1 (Thousands | Hundreds) = 2 | 7
019002: 0 0 0 0 | 0 0 1 1 (Ten thousands) = 0 | 3

6 digits BCD
032768

- If the F-54 instruction is used for programming, the F-04w instruction displays the program during monitoring.
  Resembled instructions: F-04, F-04d, F-54, F-154

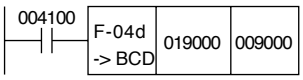
F-04d
-> BCD

Convert 32-bit binary to 10-digit BCD

Symbol	F-04d -> BCD | S | D	
Function	The contents of 4-byte 32-bit binary in the registers S to S+3 converted into a BCD, and the result is stored in 5 bytes of the registers D to D+4.	
Operation	S to S+3 -> D to D+4	
S	Use range C *	
D	Use range H *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S to S+3	Unchanged
	D	Result (ones and tens)
	D+1	Result (hundreds and thousands)
	D+2	Result (10^4 and 10^5 digits)
	D+3	Result (10^6 and 10^7 digits)
	D+4	Result (10^8 and 10^9 digits)
	Flag	Unchanged

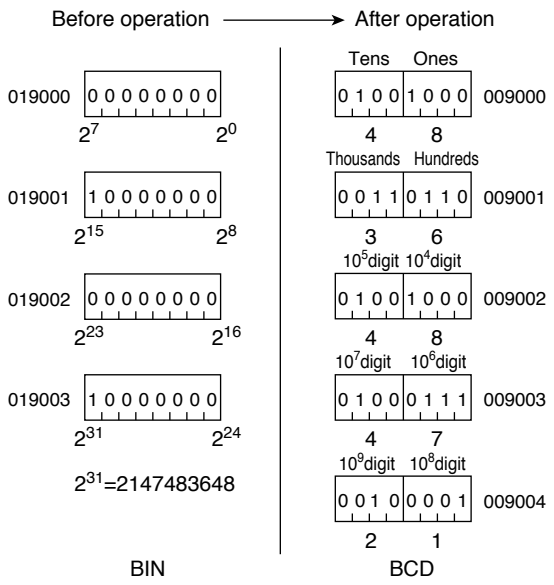
* Be sure to use even addresses for registers S and D.
Resembled instructions: F-04, F-04w, F-54, F-154

[Explanation]



Instruction	
STR	004100
F-04d	
	019000
	009000

When the input condition 004100 changes from OFF to ON, the contents of binary in registers 019000 to 019003 (32-bit) are converted into a BCD and stored in area of registers 009000 to 009004.

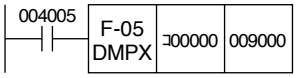


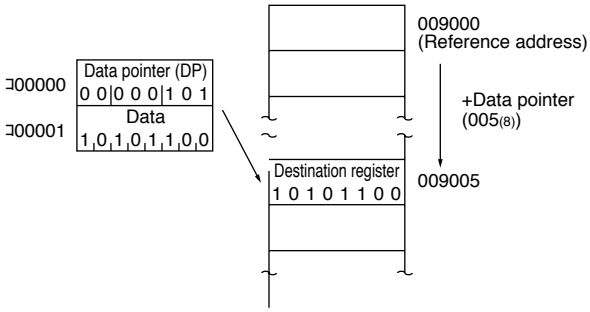
F-05 DMPX — Demultiplexes 1-byte data (DeMultiPleXer)

Symbol	F-05 DMPX | S | D	
Function	The contents of the register S+1 are transferred to the register of which address is implied by the register D (reference address) modified by the contents of the register S (data pointer).	
Operation	S+1 -> D+〈S〉 (Data pointer (DP); Reference address)	
S	Use range B	
D	Use range F	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1	Unchanged
	D	Unchanged
	D+<S>	Contents of the register S+1
	Flag	Unchanged

[Explanation]

Instruction	
STR	004005
F-05	
	⊐00000
	009000





The following transfer takes place when the input condition 004005 changes from OFF to ON.
The data in ⊐00000+1, that is , the data in ⊐00001 is transferred to the address 009005 which obtained after modifying the reference address (009000) by the contents of the data pointer ⊐00000 (005(8)).

- Since the data pointer is represented by an octal number it may assume a number between 000 to 377(8). In the above example, it would be possible to demultiplex data to address, 009000 to 009377, by varying the data pointer value in reference to the reference address.

Resembled instructions: F-05w, F-05d, F-73, F-73w

- Although it is possible to program other than the block top address to D (reference address), of that address contained block for the reference address.

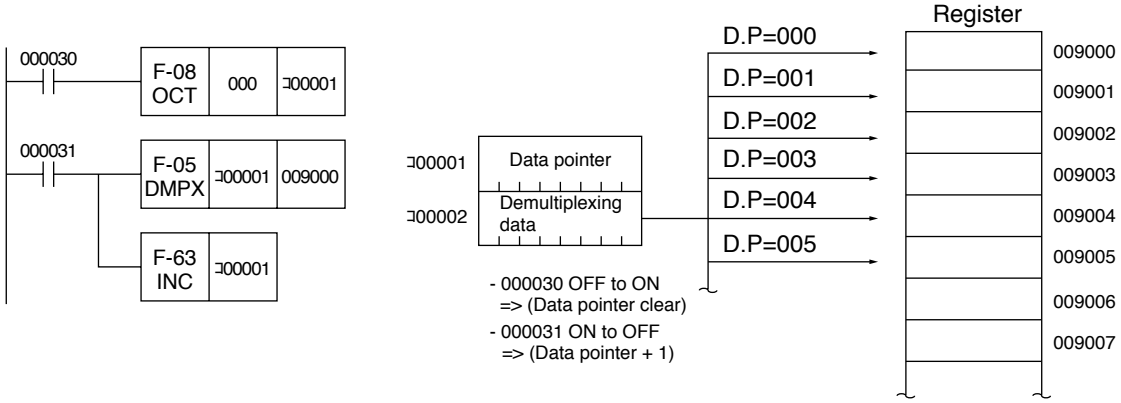
[Example]

D setting	Operational reference address
⊐00050	⊐00000
b00210	b00000
009105	009000
00033210	00033000

⇒ See page "Data memory block and reference address."

Reference

A programming example to change the demultiplexing address by varying the data pointer.



(1) When 000030 is changed from OFF to ON, the octal constant 000(8) is transferred to ⊐00001. (Data pointer 000(8))
(2) When 000031 is changed from OFF to ON, the contents of ⊐00002 are transferred to (009000+000(8)=009000). The contents of ⊐00001 changed to 001(8) as it is incremented plus 1 by the F-63 instruction.
(3) When 000031 is changed from OFF to ON again, the contents of ⊐00002 are transferred to (009000+001(8) =009001). The contents of ⊐00001 change to 002(8) as it is incremented plus 1 by the F-63 instruction.

Thereafter, the contents of ⊐00002 are demultiplexed to registers up to 009377.

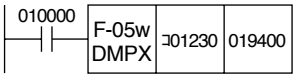
F-05w DMPX

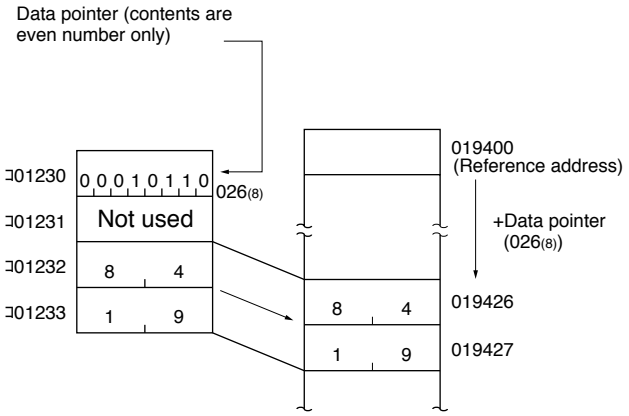
Demultiplexs 1-word data (DeMultiPleXer)

Symbol	F-05w DMPX | S | D	
Function	The contents of the registers S+2, S+3 are transferred to 2-byte area of the registers of which address is implied by the register D (reference address) modified by the contents of the register S (data pointer).	
Operation	S+2, S+3 -> D+<S> D+ <S> +1 Data pointer (DP) Reference address	
S	Use range C - Be sure to use even addresses for register S. (Odd address such as ⊐00011 etc. are prohibited to use.) - Set word addresses (000 to 376(8)) to register S.	
D	Use range F	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1, S+2, S+3	Unchanged
	D	Unchanged
	D+ <S>	Contents of the register S+2
	D+ <S> +1	Contents of the register S+3
	Flag	Unchanged

[Explanation]

Instruction	
STR	010000
F-05w	
	⊐01230
	019400





The following transfer takes place when the input condition 010000 changes from OFF to ON.

- The data in ⊐01230+2, ⊐01230+3, that is, the data in ⊐01232 and ⊐01233 are transferred to the 2-byte area from the address 019426 which obtained after modifying the reference address 019400 by the contents of the data pointer ⊐01230 (026(8)).

Resembled instructions: F-05, F-05d, F-72, F-72w, F-72d

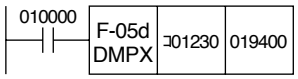
F-05d DMPX

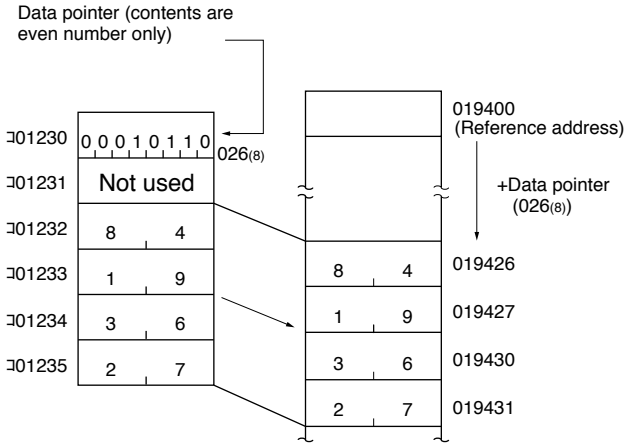
Demultiplexs 2-word data (DeMultiPleXer)

Symbol	F-05d DMPX | S | D	
Function	The contents of the registers S+2 to S+5 are transferred to 4-byte area of the registers of which address is implied by the register D (reference address) modified by the contents of the register S (data pointer).	
Operation	S+2 to S+5 -> D+ <S> to D+ <S> +1 Data pointer (DP) Reference address	
S	Use range H - Be sure to use even addresses for register S. (Odd address such as ⊐00011 etc. are prohibited to use.) - Set word addresses (000 to 376(8)) to register S.	
D	Use range F	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S to S+5	Unchanged
	D	Unchanged
	D+<S>	Contents of the register S+2
	D+<S>+1	Contents of the register S+3
	D+<S>+2	Contents of the register S+4
	D+<S>+3	Contents of the register S+5
	Flag	Unchanged

[Explanation]

Instruction	
STR	010000
F-05d	
	⊐01230
	019400





The following transfer takes place when the input condition 010000 changes from OFF to ON.

- The data in ⊐01230+2 to ⊐01230+5, that is , the data in ⊐01232 to ⊐01235 are transferred to the 4-byte area from the address 019426 which obtained after modifying the reference address 019400 by the contents of the data pointer ⊐01230 (026(8)).

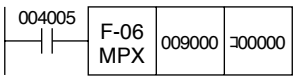
Resembled instructions: F-05, F-05w, F-72, F-72w, F-72d

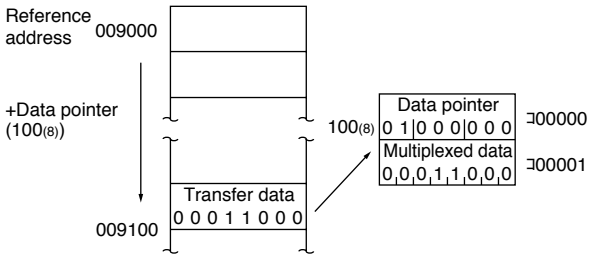
F-06 MPX — Multiplex 1-byte data (MultiPleXer)

Symbol	F-06 MPX | S | D	
Function	The contents of the register whose address is implied by the register S (reference address) modified by the contents of the register D (data pointer), are transferred to the register D+1.	
Operation	S+<D> -> D+1 Data pointer (DP) Reference address	
S	Use range F	
D	Use range B	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Unchanged (data pointer)
	D+1	Contents of the register S+<D>
	Flag	Unchanged

[Explanation]

Instruction	
STR	004005
F-06	
	009000
	⊐00000





The following transfer takes place when the input condition 004005 changes from OFF to ON.
The contents of the address 009100 is transferred to ⊐00000+1 (⊐00001) which is obtained by modifying the reference address 009000 by the contents of the data pointer ⊐00000 (100(8)).

- Since the data pointer is represented by an octal number it can assume a number between 000 to 377(8). In the above example, it would be possible to multiple data to address 009000 to 009377, by varying the data pointer value in reference to the reference address.

Resembled instructions: F-06w, F-06d, F-73, F-73w

- Although it is possible to program other than the block top address to S (reference address), the programmable controller assumes for its operation the top address of that address contained block for the reference address.

[Ex.]

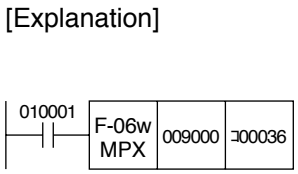
S setting	Operational reference address
⊐00051	⊐00000
b00106	b00000
009023	009000
00031257	00031000

⇒ See page "Data memory block and reference address."

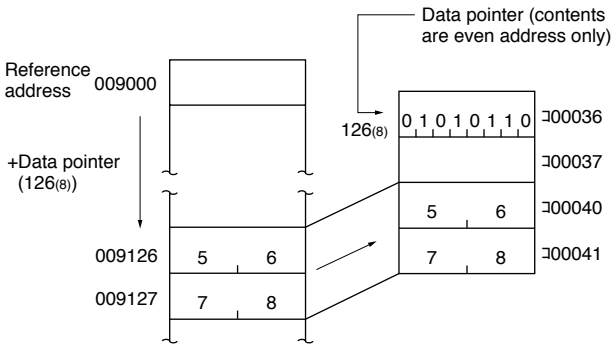
F-06w MPX — Multiplex 1-word data (MultiPleXer)

Symbol	F-06w MPX | S | D	
Function	The 2-byte contents of the register whose address is implied by the register S (reference address) modified by the contents of the register D (data pointer), are transferred to the registers D+2, D+3.	
Operation	S+<D>, S+<D> +1 -> D+2, D+3 Data pointer (DP) Reference address	
S	Use range F	
D	Use range C - Be sure to use even addresses for register D. (Odd address such as ⊐00011 etc. are prohibited to use.) - Set word addresses (000 to 376(8)) to register D.	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D, D+1	Unchanged
	D+2	Contents of the register S+<D>
	D+3	Contents of the register S+<D>+1
	Flag	Unchanged

[Explanation]



Instruction	
STR	010001
F-06w	
	009000
	⊐00036



The following transfer takes place when the input condition 010001 changes from OFF to ON.

- The contents of the address 009126 and 009127 are transferred to ⊐00040 (⊐00036+2) and ⊐00041 (⊐00036+3) which is obtained by modifying the reference address 009000 by the contents of the data pointer ⊐00036 (126(8)).

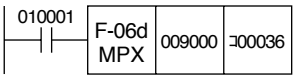
Resembled instructions: F-06, F-06w, F-73, F-73w, F-73d

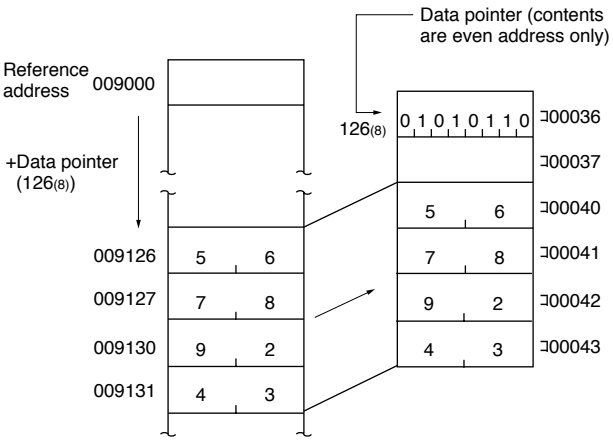
F-06d MPX — Multiplex 2-word data (MultiPleXer)

Symbol	F-06d MPX | S | D	
Function	The 4-byte contents of the register whose address is implied by the register S (reference address) modified by the contents of the register D (data pointer), are transferred to the registers D+2 to D+5.	
Operation	S+<D>to S+<D>+3 -> D+2 to D5 Data pointer (DP) Reference address	
S	Use range F	
D	Use range H - Be sure to use even addresses for register D. (Odd address such as ⊐00011 etc. are prohibited to use.) - Set word addresses (000 to 376(8)) to register D.	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D, D+1	Unchanged
	D+2	Contents of the register S+<D>
	D+3	Contents of the register S+<D>+1
	D+4	Contents of the register S+<D>+2
	D+5	Contents of the register S+<D>+3
	Flag	Unchanged

[Explanation]

Instruction	
STR	010001
F-06d	
	009000
	⊐00036





The following transfer takes place when the input condition 010001 changes from OFF to ON.

- The contents of the address 009126 to 009131 are transferred to ⊐00040 (⊐00036+2) to ⊐00043 (⊐00036+5) which is obtained by modifying the reference address 009000 by the contents of the data pointer ⊐00036 (126(8)).

Resembled instructions: F-06, F-06w, F-73, F-73w, F-73d

F-07 DCML — Transfers decimal (1 byte) constant (DeCiMaL)

Symbol	F-07 DCML | n | D	
Function	A decimal constant "n" is transferred to the register D.	
Operation	n → D	
n	Use range 000 to 255	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
After operation	D	n (000 to 255)
	Flag	Unchanged

[Explanation]

Instruction	
STR	004004
F-07	
	015
	009100

004004 — F-07 DCML | 015 | 009100

When the input condition 004004 changes from OFF to ON, the decimal constant "15" is transferred to the register 009100.
Register 009100 becomes the following values of binary codes.

009100 | 0 0 0 0 1 1 1 1 |

$2^3+2^2+2^1+2^0=15$

Resembled instructions: F-07w, F-07d, F-97

F-07w DCML — Transfer decimal (1 word) constant (DeCiMaL)

Symbol	F-07w DCML | n | D	
Function	A decimal constant "n" is transferred to the registers D, D+1.	
Operation	n-> D, D+1	
n	Use range 00000 to 65535	
D	Use range B - Be sure to use even addresses for register D. (Odd address such as 019003 etc. are prohibited to use.)	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D, D+1	n
	Flag	Unchanged

[Explanation]

Instruction	
STR	004001
F-07w	
	22659
	019100

004001 — F-07w DCML | 22659 | 019100

When the input condition 004001 changes from OFF to ON, the decimal constant 22659 is transferred to the registers 019100 and 019101.

	019101	019100
After transferring	0 1 0 1 1 0 0 0	1 0 0 0 0 0 1 1

$2^{14}+2^{12}+2^{11}+2^7+2^1+2^0=22659$

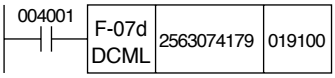
Resembled instructions: F-07, F-07d, F-97

F-07d DCML

Transfer decimal (2 word) constant (DeCiMaL)

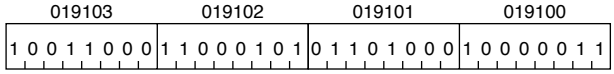
Symbol	F-07d DCML | n | D	
Function	A decimal constant "n" is transferred to the registers D to D+3.	
Operation	n -> D to D+3	
n	Use range 0000000000 to 4294967295	
D	Use range C - Be sure to use even addresses for register D. (Odd address such as 019003 etc. are prohibited to use.)	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D to D+3	n
	Flag	Unchanged

[Explanation]



Instruction	
STR	004001
F-07d	
	2563074179
	019100

When the input condition 004001 changes from OFF to ON, the decimal constant 2563074179 is transferred to the registers 019100 to 019103. 019100 to 019103 become the following values of binary codes.



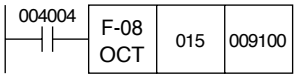
$2^{31}+2^{28}+2^{27}+2^{23}+2^{22}+2^{18}+2^{16}+2^{14}+2^{13}+2^{11}+2^{7}+2^{1}+2^{0}$
$=2563074179$

Resembled instructions: F-07, F-07w, F-97

F-08 OCT — Transfer octal (1 byte) constant (OCTal)

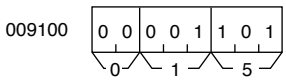
Symbol	F-08 OCT | n | D	
Function	An octal constant "n" is transferred to the register D.	
Operation	n → D	
n	Use range 000 to 377(8)	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D	n (000 to 377(8))
	Flag	Unchanged

[Explanation]



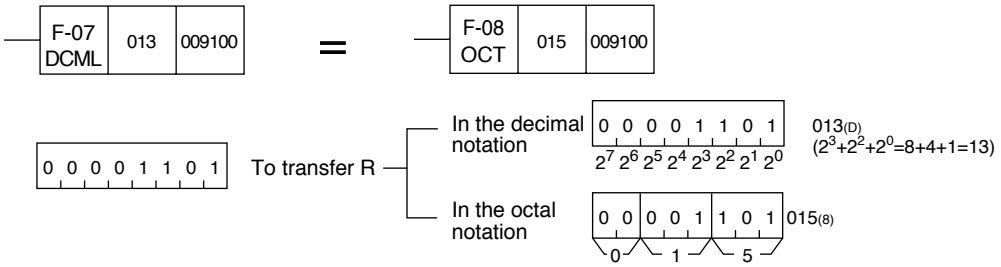
Instruction	
STR	004004
F-08	
	015
	009100

When the input condition 004004 changes from OFF to ON, the octal constant 015 is transferred to the register 009100.
The register 009100 is in the following code representation.



Resembled instructions: F-08w, F-08d, F-71, F-71w

Reference — Though F-07 (transfer decimal constant) handles a decimal number and F-08 (transfer octal constant) an octal number, the contents of the register after the transfer are represented in the binary code for both instructions.

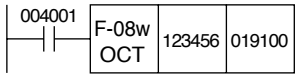


When the F-08 instruction is used to preset the data pointer of F-05 (demultiplex) and F-06 (multiplex), you will be able to make direct recognition of the data memory address (octal).

F-08w OCT — Transfer octal (1 word) constant (OCTal)

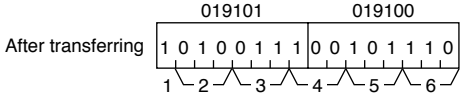
Symbol	F-08w OCT | n | D	
Function	An octal constant "n" is transferred to the register D, D+1.	
Operation	n -> D, D+1	
n	Use range 000000 to 177777(8)	
D	Use range B - Be sure to use even addresses for register D. (Odd address such as 019003 etc. are prohibited to use.)	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D, D+1	n
	Flag	Unchanged

[Explanation]



Instruction	
STR	004001
F-08w	
	123456
	019100

When the input condition 004001 changes from OFF to ON, the octal constant 123456 is transferred to the registers 019100 and 019101.

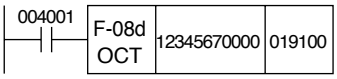


Resembled instructions: F-08, F-08d, F-71, F-71w

F-08d OCT — Transfer decimal (2 words) constant (OCTal)

Symbol	F-08d OCT | n | D	
Function	An octal constant "n" is transferred to the register D to D+3.	
Operation	n -> D to D+3	
n	Use range 00000000000 to 37777777777(8)	
D	Use range C - Be sure to use even addresses for register D. (Odd address such as 019003 etc. are prohibited to use.)	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D to D+3	n
	Flag	Unchanged

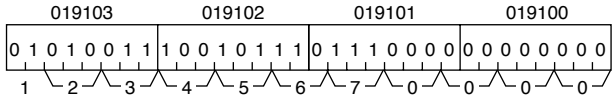
[Explanation]



Instruction	
STR	004001
F-8d	
	12345670000
	019100

When the input condition 004001 changes from OFF to ON, the octal constant 12345670000 is transferred to the registers 019100 to 019103.

After transferring



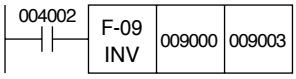
Resembled instructions: F-08, F-08w, F-71, F-71w

F-09 INV — Complements 8-bit data (INVerter)

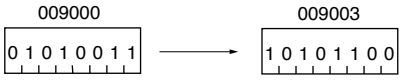
Symbol	F-09 INV | S | D	
Function	The contents of the register S are inverted and stored in the register D.	
Operation	S → D	
S	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Inversion of register S contents
	Flag	Unchanged

[Explanation]

Instruction	
STR	004002
F-09	
	009000
	009003



When the input condition 004002 changes from OFF to ON, the 8 bits contents of the register 009000 are inverted and its result is stored in the register 009003.



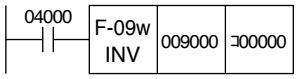
The contents of the register 009000 remain unchanged.

F-09w INV — Complements 16-bit data (INVerter)

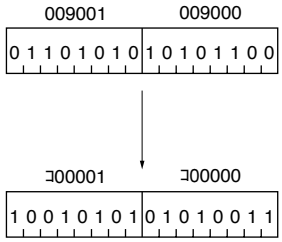
Symbol	F-09w INV | S | D	
Function	The contents of the registers S, S+1 (16 bits) are inverted and stored in the registers D, D+1.	
Operation	$\overline{S, S+1}$ -> D, D+1	
S	Use range B *	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1	Unchanged
	D	Inversion of register S contents
	D+1	Inversion of register S+1
	Flag	Unchanged

[Explanation]

Instruction	
STR	004000
F-09w	
	009000
	⊐00000



When the input condition 004000 changes from OFF to ON, the contents of the registers 009000 and 009001 (16 bits) are inverted and its result is stored in the registers ⊐00000 and ⊐00001.



The contents of the registers 009000 and 009001 remain unchanged.

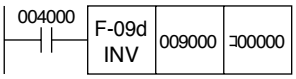
* Be sure to use even addresses for registers S and D.
(Odd address such as 019003 etc. are prohibited to use.)

F-09d INV — Complements 32-bit data (INVerter)

Symbol	F-09d INV | S | D	
Function	The contents of the registers S to S+3 (32 bits) are inverted and stored in the registers D to D+3.	
Operation	$\overline{\text{S to S+3}}$ -> D to D+3	
S	Use range C *	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S to S+3	Unchanged
	D to D+3	Inversion of registers S to S+3
	Flag	Unchanged

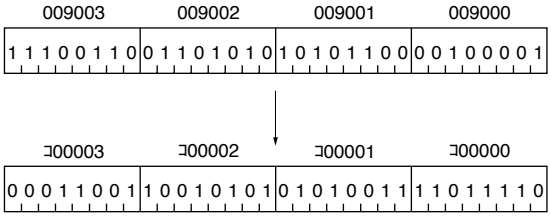
* Be sure to use even addresses for registers S and D. (Odd address such as 019003 etc. are prohibited to use.)

[Explanation]



Instruction	
STR	004000
F-09d	
	009000
	⊐00000

When the input condition 004000 changes from OFF to ON, the contents of the registers 009000 to 009003 (32 bits) are inverted and its result is stored in the registers ⊐00000 to ⊐00003.



The contents of the registers 009000 to 009003 remain unchanged.

F-10 ADD — Add register and register (BCD 2 digits) (ADD)

Symbol	F-10 ADD | S_1 | S_2 | D
Function	The contents of the register S_1 are added with the contents of the register S_2 in BCD 2 digits and its result is stored in the register D.
Operation	S_1+S_2 -> D
S_1	Use range A
S_2	Use range A
D	Use range A
Condition	Rising edge of input signal (OFF to ON)

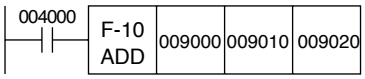
Contents after operation

S_1	Unchanged
S_2	Unchanged
D	Lower 2 digits of the result. - Unchanged when the contents of registers S_1 and S_2 are not BCD code. (Does not calculate.)

Flag

Result	Zero 07357	Carry 07356	Error 07355	Non-carry 07354
0	1	0	0	1
1 to 99	0	0	0	1
100	1	1	0	0
101 and above	0	1	0	0
S_1 and S_2 are not BCD code.	0	0	1	0

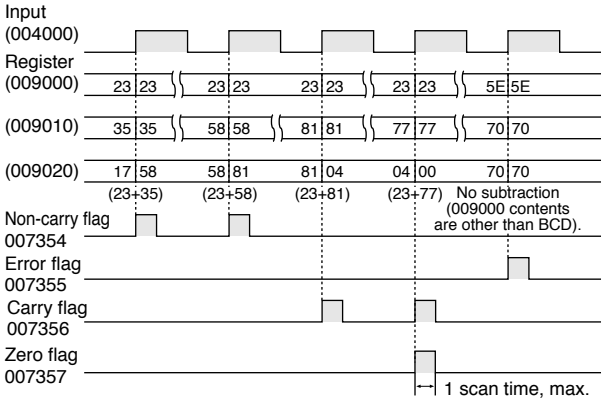
[Explanation]



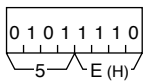
Instruction	
STR	004000
F-10	
	009000
	009010
	009020

When the input condition 004000 changes from OFF to ON, the contents of the register 009000 are added with the contents of the register 009010 and its result is stored in the register 009020. The contents of registers 009000 and 009010 remain unchanged.

- Transition of result and flags

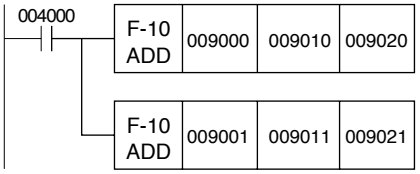


- If the contents of S_1 and S_2 are numbers other than BCD code, the error flag (007355) is set ON and no addition will be done.

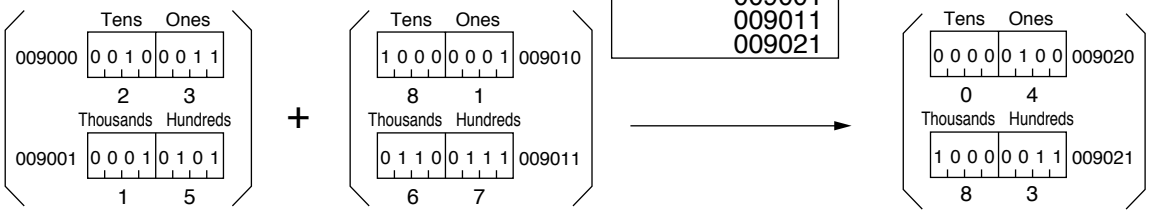
(Example) S_1 

1110 is prohibited in the BCD mode.

Reference In case 3 BCD digits or more have to be added, the F-10 instruction must be provided successively.
When the F-10 instruction is programmed repeatedly, the contents of the carry flag (007356) are also added after the second instruction. For the F-10 instruction that appears first in succession to the STR instruction, the contents of the carry flag (007356) are not added.



Instruction	
STR	004000
F-10	
	009000
	009010
	009020
F-10	
	009001
	009011
	009021



- The above example shows the case of 1523+6781=8304.
- If programmed from lower digit, the carry information will be carried on to upper digit.
  => See the "Double length operation."

F-10w ADD

Add register and register (BCD 4 digits) (ADD)

Symbol	F-10w ADD | S_1 | S_2 | D
Function	The contents of the registers S_1, S_1+1 are added with the contents of the registers S_2, S_2+1 in BCD 4 digits and its result is stored in the registers D, D+1.
Operation	(S_1, S_1+1) + (S_2, S_2+1) -> D, D+1
S_1	Use range B *
S_2	Use range B *
D	Use range B *
Condition	Rising edge of input signal (OFF to ON)

Contents after operation

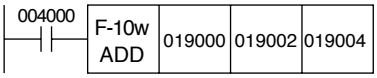
S_1,S_1+1	Unchanged
S_2,S_1+2	Unchanged
D	Lower 2 digits of the result
D+1	Upper 2 digits of the result

Unchanged when the contents of registers S_1, S_1+1, S_2 and S_2+1 are not BCD code.

Flag

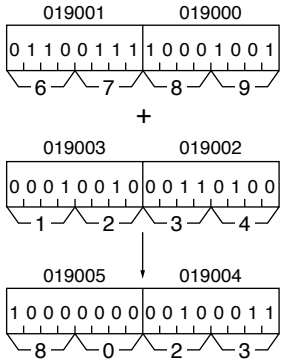
Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0	1
1 to 9999	0	0	0	1
10000	1	1	0	0
10001 and above	0	1	0	0
Not BCD code	0	0	1	0

[Explanation]



Instruction	
STR	004000
F-10w	
	019000
	019002
	019004

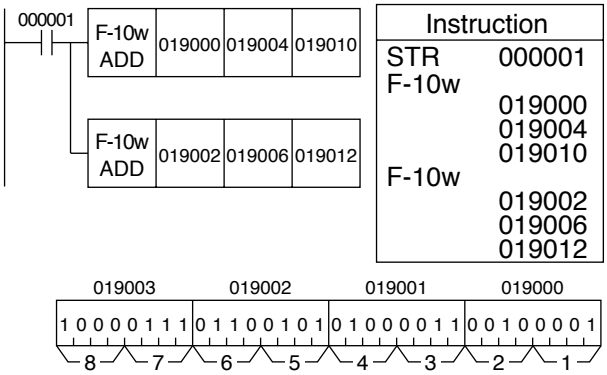
When the input condition 004000 changes from OFF to ON, the 4 digits BCD contents of the registers 019000 and 019001 are added with the contents of the registers 019002 and 019003 and its result is stored in the registers 019004 and 019005.



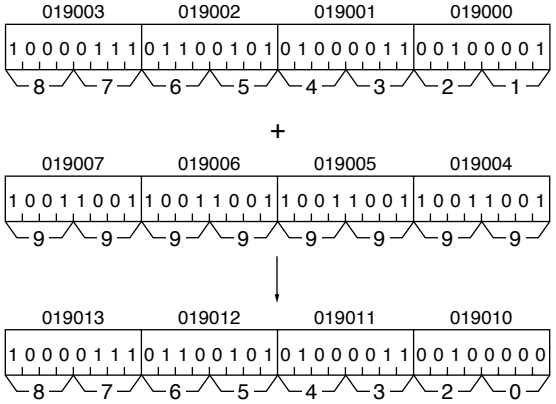
* Be sure to use even addresses for registers S_1, S_2 and D.
(Odd address such as 019003 etc. are prohibited to use.)

Reference

As double-length operation is possible for the F-10w instruction, same as the F-10 instruction, write F-10w instruction in succession to add 8 digits or more BCD.



Instruction	
STR	000001
F-10w	
	019000
	019004
	019010
F-10w	
	019002
	019006
	019012

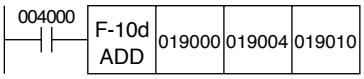


F-10d ADD — Adds register and register (BCD 8 digits) (ADD)

Symbol	F-10d ADD | S_1 | S_2 | D				
Function	The 8-digit BCD contents of the registers S_1 to S_1+3 are added by the contents of the registers S_2 to S_2+3 and its result is stored in the registers D to D+3.				
Operation	(S_1 to S_1+3) + (S_2 to S_2+3) -> D to D+3				
S_1	Use range C *				
S_2	Use range C *				
D	Use range C *				
Condition	Rising edge of input signal (OFF to ON)				
Contents after operation	S_1 to S_1+3	Unchanged			
	S_2 to S_2+3	Unchanged			
	D to D+3	Result (8-digit BCD)	Unchanged when the contents of registers S_1 to S_1+3, S_2 to S_2+3 are not BCD code.		

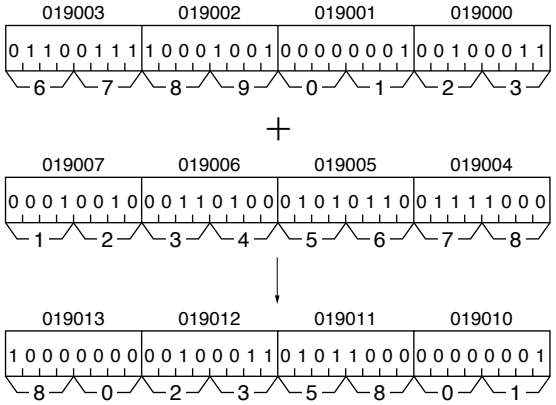
Contents after operation — Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	0	1	0	0	1
	1 to 99999999	0	0	0	1
	100000000	1	1	0	0
	100000001 and above	0	1	0	0
	Not BCD code	0	0	1	0

[Explanation]



Instruction	
STR	004000
F-10d	
	019000
	019004
	019010

When the input condition 004000 changes from OFF to ON, the 8 digits BCD contents of the registers 019000 to 019003 are subtracted by the contents of the registers 019004 to 019007 and its result is stored in the registers 019010 to 019013.



* Be sure to use even addresses for registers S_1, S_2 and D.
(Odd address such as 019003 etc. are prohibited to use.)

- If the contents of registers S_1 to S_1+3 or S_2 to S_2+3 are not BCD codes, no addition occurs, with the error flag

Reference

As double-length operation is possible with the F-10d instruction, same as the F-10w instruction, write F-10d instruction in succession to subtract 16 digits or more of BCD.

Fc10 ADD — Add register (BCD 2 digits) and constant (2 digits) (ADD)

Symbol	Fc10 ADD | S1 | n | D
Function	The contents of the register S1 are added with the 2-digit BCD constant "n" and its result is stored in the register D.
Operation	S1+n→D
S1	Use range A
n	Use range 00 to 99
D	Use range A
Condition	Rising edge of input signal (OFF to ON)

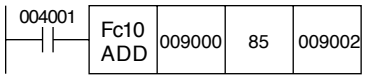
Contents after operation

S1	Unchanged
D	Lower 2 digits of the result - Unchanged when the contents of the register S1 are not a BCD code. (Does not calculate.)

Flag

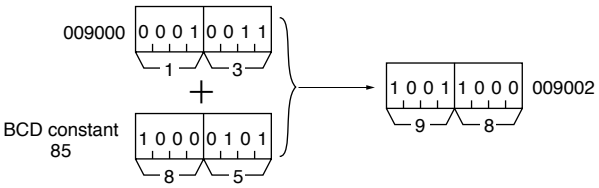
Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0	1
1 to 99	0	0	0	1
100	1	1	0	0
101 and above	0	1	0	0
S1 is not BCD code	0	0	1	0

[Explanation]

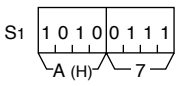


Instruction	
STR	004001
Fc10	
	009000
	85
	009002

When the input condition 004001 changes from OFF to ON, the contents of the register 009000 are added with the BCD constant 85 and its result is stored in the register 009002.
If operates in the same timing as the F-10 instruction.



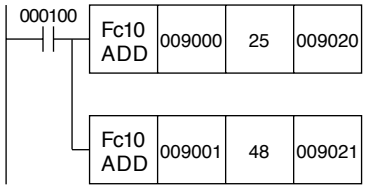
- If the contents of S1 is a number other than BCD code, the error flag (007355) is set ON and no addition will be done.

[Example] 

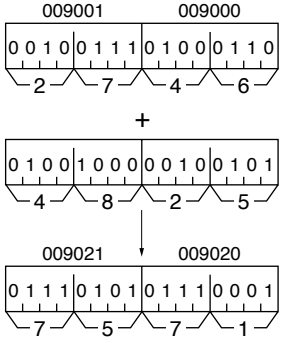
1010 is the code prohibited to use in the BCD code.

Reference

Similar as the F-10 instruction, it is possible to add 3 digits or more of BCD value.



Instruction	
STR	000100
Fc10	
	009000
	25
	009020
Fc10	
	009001
	48
	009021



Fc10w ADD — Add register (BCD 4 digits) and constant (4 digits) (ADD)

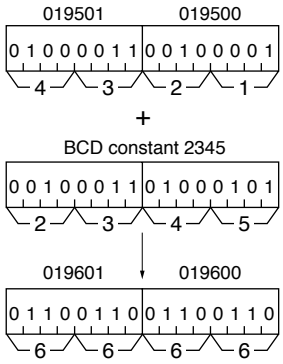
Symbol	Fc10w ADD | S_1 | n | D					
Function	The BCD 4-digit contents of the registers S_1, S_1+1 are added with the 4-digit BCD constant "n" and its result is stored in the registers D, D+1.					
Operation	(S_1, S_1+1) +n -> D, D+1					
S_1	Use range B *					
n	Use range 0000 to 9999					
D	Use range B *					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	S_1,S_1+1	Unchanged				
	D	Lower 2 digits of the result	Unchanged when the contents of registers S_1 and S_1+1 are not BCD code. (Does not calculate.)			
	D+1	Upper 2 digits of the result				
	Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		0	1	0	0	1
		1 to 9999	0	0	0	1
		10000	1	1	0	0
		10000 and above	0	0	0	0
		Not BCD code	0	0	1	0

[Explanation]

Instruction	
STR	002001
Fc10w	
	019500
	2345
	019600

002001 | Fc10w ADD | 019500 | 2345 | 019600

When the input condition 002001 changes from OFF to ON, the 4-digit BCD contents of the registers 019500 and 019501 are added with the BCD constant 2345 and its result is stored in the registers 019600 and 019601.



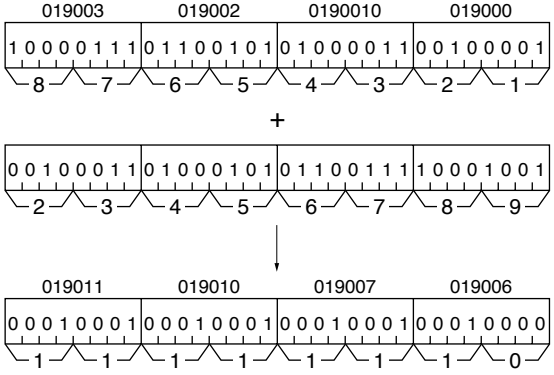
* Be sure to use even addresses for registers S_1 and D.
(Odd address such as 019003 etc. are prohibited to use.)

Reference

As double-length operation is possible with the Fc10w instruction, same as the Fc10 instruction, write Fc10w instruction in succession to add 8 digits or more of BCD.

000001 | Fc10w ADD | 019000 | 6789 | 019006
000001 | Fc10w ADD | 019002 | 2345 | 019010

Instruction	
STR	000001
Fc10w	
	019000
	6789
	019006
Fc10w	
	019002
	2345
	019010



Fc10d ADD — BCD register (BCD 8 digits) and constant (4 digits) (ADD)

Symbol		Fc10d ADD | S_1 | n | D
Function		The BCD 4-digit contents of the registers S_1 to S_1+3 are added with the 4-digit BCD constant "n" and its result is stored in the registers D to D+3.
Operation		(S_1 to S_1+3) +n -> D to D+3
S_1		Use range C *
n		Use range 0000 to 9999
D		Use range C *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S to S+3	Unchanged
	D to D+3	Result (8-digit BCD) Unchanged when the contents of registers S_1 to S_1+3 are not BCD code. (Does not calculate.)

Flag

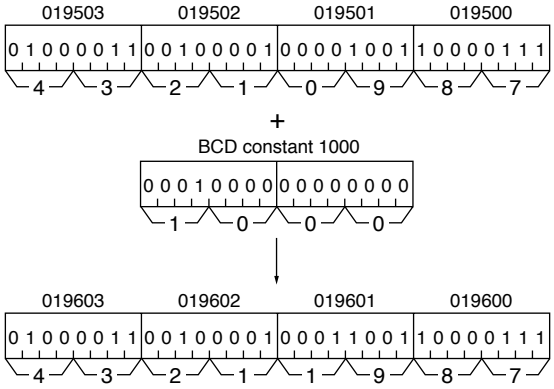
Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0	1
1 to 99999999	0	0	0	1
100000000	1	1	0	0
100000001 and above	0	1	0	0
Not BCD code	0	0	1	0

[Explanation]

002001 — Fc10d ADD | 019500 | 1000 | 019600

Instruction	
STR	002001
Fc10d	
	019500
	1000
	019600

When the input condition of 002001 changes from OFF to ON, this instruction adds the 8-digit BCD contents of registers 019500 to 019503 to BCD constant 1000, and stores the result in registers 019600 to 019603.



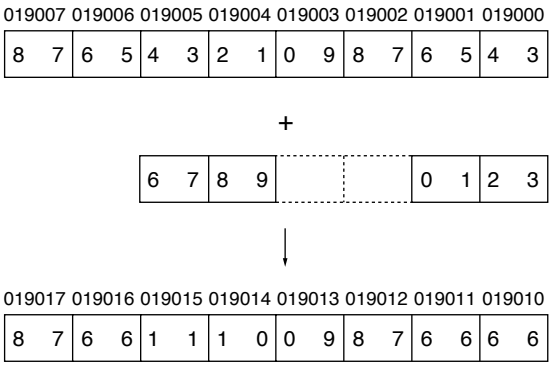
* Be sure to use even addresses for registers S_1 and D.
(Odd address such as 019003 etc. are prohibited to use.)

Reference

As double-length operation is possible with the Fc10d instruction, same as the Fc10 instruction, write Fc10d instruction in succession to subtract 16 digits or more of BCD.

000001 — Fc10d ADD | 019000 | 0123 | 019010
000001 — Fc10d ADD | 019004 | 6789 | 019014

Instruction	
STR	000001
Fc10d	
	019000
	0123
	019010
Fc10d	
	019004
	6789
	019014



F-11 SUB — Subtracts register from register (BCD 2 digits) (SUBtract)

Symbol	F-11 SUB | S_1 | S_2 | D
Function	The contents of the register S_1 are subtracted from the contents of the register S_2 in BCD 2 digits and its result is stored in the register D.
Operation	S_1-S_2 -> D
S_1	Use range A
S_2	Use range A
D	Use range A
Condition	Rising edge of input signal (OFF to ON)

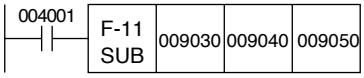
Contents after operation:

S_1	Unchanged
S_2	Unchanged
D	Unchanged when the contents of registers S_1 and S_2 are not BCD code. (Does not calculate.)

Flag:

Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0	1
1 to 99	0	0	0	1
Negative value	0	1	0	0
S_1 and S_2 are not BCD code	0	0	1	0

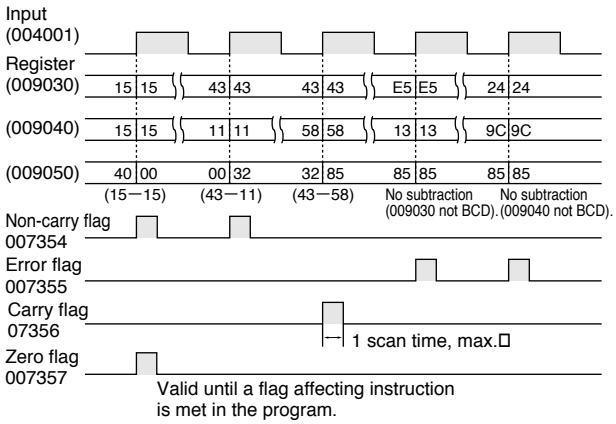
[Explanation]



Instruction	
STR	004001
F-11	
	009030
	009040
	009050

When the input condition 004001 changes from OFF to ON, the contents of the register 009030 are subtracted by the contents of the register 009040 and its result is stored in the register 009050. The contents of registers 009030 and 009040 remain unchanged.

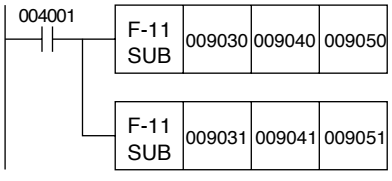
- Transition of operational result and flags



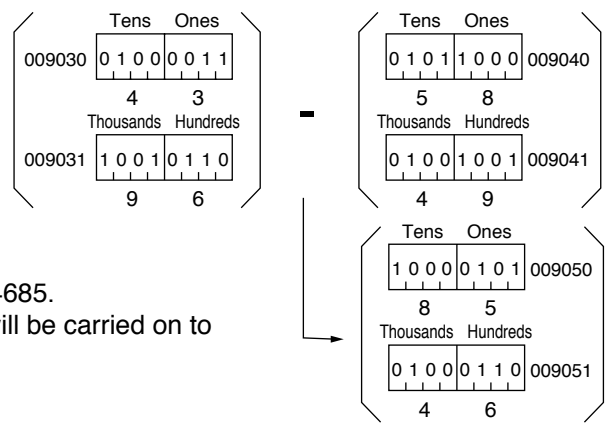
- If (contents of S_1) < (contents of S_2) is calculated, the answer will be produced in the complement of 100.
  [Example]
  23−85= -62 will produce the answer of 38 which is the complement of 100 of 62.
  (Assume it to be 123-85=38.)
- If the contents of S_1 and S_2 are numbers other than BCD code, the error flag (007355) is set ON and no addition will be done. (D remains unchanged.)

[Example] S_1 1 1 0 0 | 0 1 0 1 — C (H) — 5 — 1100 is prohibited in the BCD code.

Reference In case 3 BCD digits or more have to be subtracted, the F-11 instruction must be provided successively.
When the F-11 instruction is programmed repeatedly, the contents of the carry flag (007356) is also subtracted after the second instruction. For the F-11 instruction that appears first in succession to the STR instruction, the contents of the carry flag (007356) is not subtracted.



Instruction	
STR	004001
F-11	
	009030
	009040
	009050
F-11	
	009031
	009041
	009051



- The above example shows the case of 9643 - 4958 = 4685.
- If programmed from lower digit, the carry information will be carried on to upper digit. (See page "Double length operation.")

F-11w SUB — Subtract register from register (BCD 4 digits) (SUBtract)

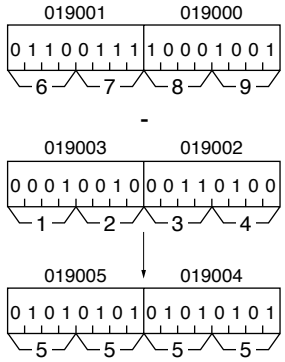
Symbol	F-11w SUB | S_1 | S_2 | D				
Function	The 4-digit BCD contents of the registers S_1 and S_1+1 are subtracted by the contents of the registers S_2, S_2+1 and its result is stored in the registers D, D+1.				
Operation	(S_1, S_1+1) - (S_2, S_2+1) -> D, D+1				
S_1	Use range B *				
S_2	Use range B *				
D	Use range B *				
Condition	Rising edge of input signal (OFF to ON)				
Contents after operation	S_1, S_1+1	Unchanged			
	S_2, S_2+1	Unchanged			
	D	Lower 2 digits of the result	Unchanged when the contents of registers S_1, S_1+1, S_2 and S_2+1 are not BCD code.		
	D+1	Upper 2 digits of the result			
	Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		0	1	0	0	1
		1 to 9999	0	0	0	1
		Negative value	0	1	0	0
		Not BCD code	0	0	1	0

[Explanation]

004000 — F-11w SUB | 019000 | 019002 | 019004

Instruction	
STR	004000
F-11w	
	019000
	019002
	019004

When the input condition 004000 changes from OFF to ON, the 4-digit BCD contents of the registers 019000 and 019001 are subtracted by the contents of the registers 019002 and 019003 and its result is stored in the registers 019004 and 019005.

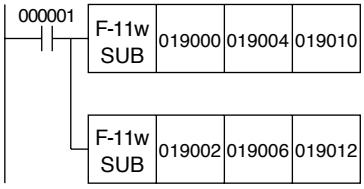


* Be sure to use even addresses for registers S_1, S_2, and D.
(Odd address such as 019003 etc. are prohibited to use.)

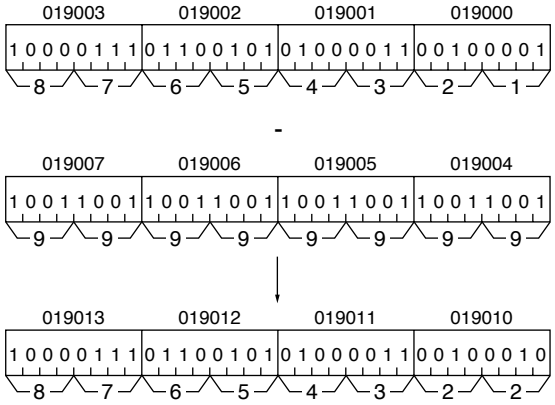
- Result will be produced as a complement of 10000, if (contents of S_1, S_1+1) < (contents of S_2, S_2+1) is carried out.
[Example]
2578-7890= -5312 will produce the answer of 4688 which is the complement of 10000 of 5312.
(Assume it to be 12578-7890=4688.)

Reference

As double-length operation is possible with the F-11w instruction, same as the F-11 instruction, write F-11w instruction in succession to subtract 16 digits or more of BCD.



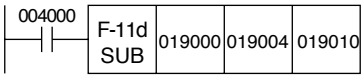
Instruction	
STR	000001
F-11w	
	019000
	019004
	019010
F-11w	
	019002
	019006
	019012



F-11d SUB — Subtract register from register (BCD 8 digits) (SUBtract)

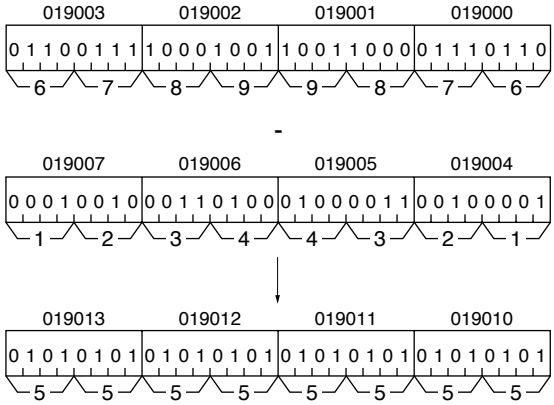
Symbol	F-11d SUB | S_1 | S_2 | D					
Function	The 8-digit BCD contents of the registers S_1 to S_1+3 are subtracted by the contents of the registers S_2 to S_2+3 and its result is stored in the registers D to D+3.					
Operation	(S_1 to S_1+3) - (S_2 to S_2+3) -> D to D+3					
S_1	Use range C *					
S_2	Use range C *					
D	Use range C *					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	S_1 to S_1+3	Unchanged				
	S_2 to S_2+3	Unchanged				
	D to D+3	Result (8-digit BCD)	Unchanged when the contents of registers S_1 to S_1+3, S_2 to S_2+3 are not BCD code.			
	Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		0	1	0	0	1
		1 to 99999999	0	0	0	1
		Negative value	0	1	0	0
		Not BCD code	0	0	1	0

[Explanation]



Instruction	
STR	004000
F-11d	
	019000
	019004
	019010

When the input condition 004000 changes from OFF to ON, the 8-digit BCD contents of the registers 019000 to 019003 are subtracted by the contents of the registers 019004 to 019007 and its result is stored in the registers 019010 to 019013.

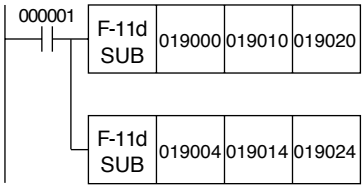


* Be sure to use even addresses for registers S_1, S_2, and D.
(Odd address such as 019003 etc. are prohibited to use.)

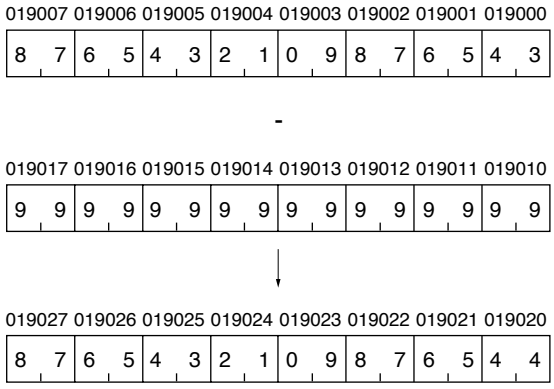
- Result will be produced as a complement of 100000000, if (contents of S_1 to S_1+3) < (contents of S_2 to S_2+3) is carried out.
  [Example]
  25780000-78900000= -53120000 will produce the answer of 46880000 which is the complement of 100000000 of 53120000. (Assume it to be 125780000-78900000=46880000.)
- If the contents of S_1 to S_1+3 and S_2 to S_2+3 are numbers other than BCD code, the error flag (007355) is set ON and no addition will be done. (D to D+3 remain unchanged.)

Reference

As double-length operation is possible with the F-11d instruction, same as the F-11 instruction, write F-11d instruction in succession to subtract 16 digits or more of BCD.



Instruction	
STR	000001
F-11d	
	019000
	019010
	019020
F-11d	
	019004
	019014
	019024



Fc11 SUB — Subtracts constant (2 digits) from register (BCD 2 digits) (SUBtract)

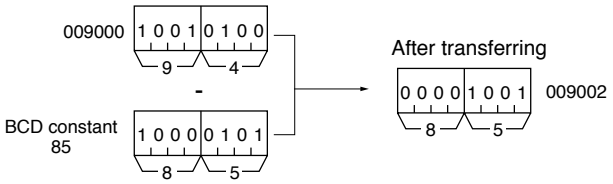
Symbol	Fc11 SUB | S_1 | n | D						
Function	2-digit BCD constant "n" is subtracted from the contents of the register S_1 and its result is stored in the register D.						
Operation	S_1-n -> D						
S_1	Use range A						
n	Use range 00 to 99						
D	Use range A						
Condition	Rising edge of input signal (OFF to ON)						
Contents after operation	S_1	Unchanged					
	D	Unchanged when the contents of the register S_1 are not a BCD code. (Does not calculate.)					
	Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
		0	1	0	0	1	
		1 to 99	0	0	0	1	
		Negative value	0	1	0	0	
		S_1 is not BCD code	0	0	1	0	

[Explanation]

Instruction	
STR	004001
Fc11	
	009000
	85
	009002

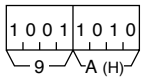
004001 | Fc11 SUB | 009000 | 85 | 009002

When the input condition 004001 changes from OFF to ON, the BCD constant 85 is subtracted from the contents of the register 009000 and its result is stored in the register 009002.
If operates in the same timing as the F-11 instruction.



- If (contents of S_1) < n is calculated, the answer will be produced in the complement of 100.
  [Example]
  23-85= -62 will produce the answer of 38 which is the complement of 100 of 62.
  (Assume it to be 123-85=38.)
- If the contents of S_1 is a number other than BCD code, the error flag (007355) is set ON and no subtraction will be done. (D remains unchanged.)

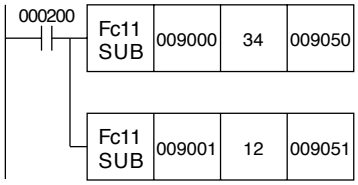
[Example]



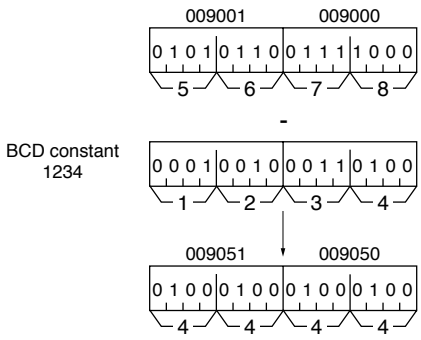
1010 is the code prohibited to use in the BCD code.

Reference

Similar as the F-11 instruction, it is possible to subtract 3 digits or more of BCD value.



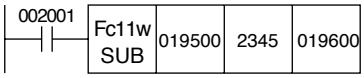
Instruction	
STR	000200
Fc11	
	009000
	34
	009050
Fc11	
	009001
	12
	009051



Fc11w SUB — Subtract constant (4 digits) from register (BCD 4 digits) (SUBtract)

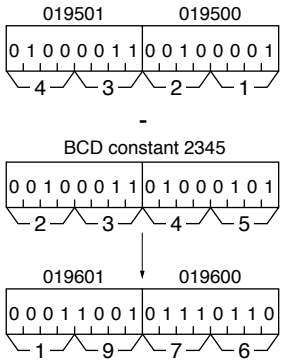
Symbol	Fc11w SUB	S_1	n	D		
Function	The 4-digit BCD contents of "n" is subtracted from the 4-digit BCD contents of the registers S_1, S_1+1 and its result is stored in the registers D, D+1.					
Operation	(S_1, S_1+1) -n -> D, D+1					
S_1	Use range B *					
n	Use range 0000 to 9999					
D	Use range B *					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	S_1, S_1+1	Unchanged				
	D	Lower 2 digits of the result	Unchanged when the contents of registers S_1 and S_1+1 are not BCD code.			
	D+1	Upper 2 digits of the result				
	Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		0	1	0	0	1
		1 to 9999	0	0	0	1
		Negative value	0	1	0	0
		Not BCD code	0	0	1	0

[Explanation]



Instruction	
STR	002001
Fc11w	
	019500
	2345
	019600

When the input condition 002001 changes from OFF to ON, the BCD constant 2345 is subtracted from the 4-digit BCD contents of the registers 019500 and 019501 and its result is stored in the registers 019600 and 019601.

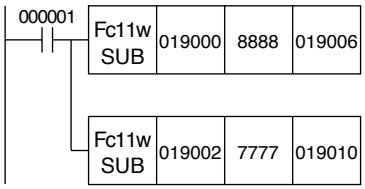


* Be sure to use even addresses for registers S_1 and D.
(Odd address such as 019003 etc. are prohibited to use.)

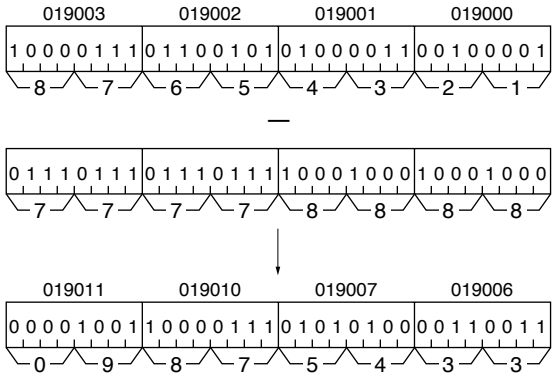
- If (contents of S_1, S_1+1) < n is calculated, the answer will be produced in the complement of 10000.
〔Example〕
4568−7890=−3322 will produce the answer of 6678 which is the complement of 10000 of 3322.
(Assume it to be 14568−7890=6678.)

Reference

As double-length operation is possible for the Fc11w instruction, same as the Fc11 instruction, write Fc11w instruction in succession to subtract 9 digits or more of BCD.



Instruction	
S T R	000001
F c11w	
	019000
	8888
	019006
F c11w	
	019002
	7777
	019010



Fc11d SUB — Subtract constant (4 digits) from register (BCD 8 digits) (SUBtract)

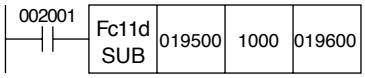
Symbol	Fc11d SUB | S_1 | n | D
Function	The 4-digit BCD constant "n" is subtracted from the 8-digit BCD contents of the registers S_1 to S_1+3 and its result is stored in the registers D to D+3.
Operation	(S_1 to S_1+3) −n→D to D+3
S_1	Use range C *
n	Use range 0000 to 9999
D	Use range C *
Condition	Rising edge of input signal (OFF to ON)

Contents after operation

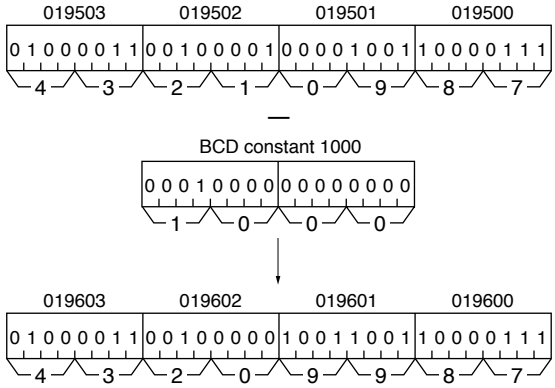
S_1 to S_1+3	Unchanged				
D to D+3	Operational result (8-digit BCD)	Unchanged when the contents of registers S_1 to S_1+3 are not BCD code. (Does not calculate.)			
Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	0	1	0	0	1
	1 to 99999999	0	0	0	1
	Negative value	0	1	0	0
	Not BCD code	0	0	1	0

[Explanation]

Instruction	
S T R	002001
F c11d	
	019500
	1000
	019600



When the input condition 002001 changes from OFF to ON, the BCD constant 1000 is subtracted from the 8-digit BCD contents of the registers 019500 to 019503 and its result is stored in the registers 019600 to 019603.

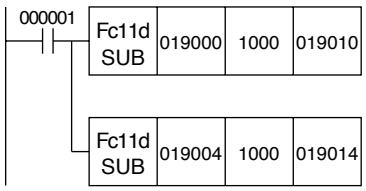


* Be sure to use even addresses for registers S_1 and D.
(Odd address such as 019003 etc. are prohibited to use.)

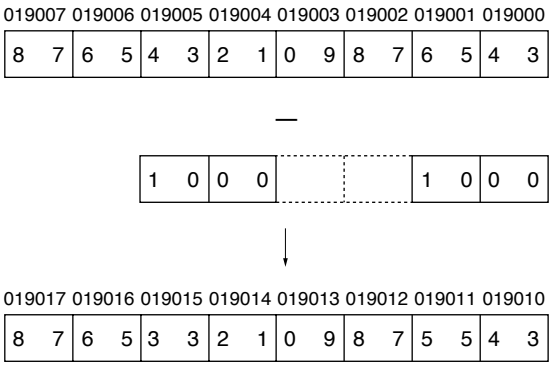
- If (contents of S_1 to S_1+3) < n is calculated, the answer will be produced in the complement of 100000000.
〔Example〕
4568−7890=−3322 will produce the answer of 6678 which is the complement of 100000000 of 3322.
(Assume it to be 100004568−7890=99996678.)

Reference

As double-length operation is possible with the Fc11d instruction, same as the Fc11 instruction, write Fc11d instruction in succession to subtract 16 digits or more of BCD.



Instruction	
S T R	000001
F c11d	
	019000
	1000
	019010
F c11d	
	019004
	1000
	019014

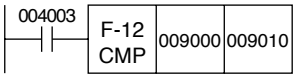


F-12 CMP — Compare register with register (1 byte) (CoMPare)

Symbol	F-12 CMP | S_1 | S_2					
Function	The contents of the register S_1 are compared with the contents of the register S_2.					
Operation	S_1< = >S_2→Flag					
S_1	Use range A					
S_2	Use range A					
Condition	When the input signal is ON (not limited to OFF to ON change)					
Contents after operation	S_1	Unchanged				
	S_2	Unchanged				
	Flag*	Register contents	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		$S_1 > S_2$	0	0	0	1
		$S_1 = S_2$	1	0	0	1
		$S_1 < S_2$	0	1	0	0

* Error flag (007355) will be already "0."
Resembled instructions: F-12w, F-12d, Fc12, Fc12w, Fc12d, F-112, F-112w, F-112d

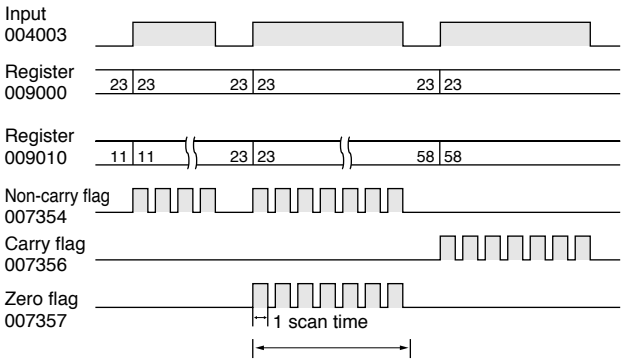
[Explanation]



Instruction	
S T R	004003
F -12	
	009000
	009010

When the input condition 04003 is ON, the contents of the register 009000 are compared with the contents of the register 009010 and its results are set in the non-carry flag (007354), carry flag (007356), and the zero flag (007357). registers 009000 and 009010 remain unchanged after this operation.

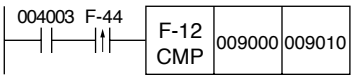
- Transition of register contents and flags



So long as the input condition is ON, comparison is done at every scan and the zero flag is tuned ON.

Reference

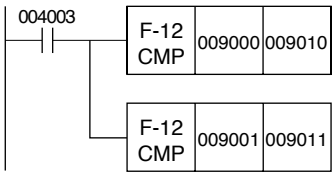
In case comparison is to be done only at a OFF to ON transition of the input condition, use the differentiate instruction in conjunction with the input condition.



Instruction	
S T R	004003
F -44	
F -12	009000
	009010

Reference

To compare data of 2 bytes or more, it must be so programmed that comparison should start from a lower digit, as in addition (F-10) and subtraction (F-11). If the F-12 is programmed in succession, the contents of the carry flag (007356) are also compared after the second F-12 instruction. For the F-12 instruction that first follows the STR instruction, the contents of the carry flag (007356) are exempted from comparison.



Instruction	
S T R	004003
F -12	
	009000
	009010
F -12	
	009001
	009011

When programmed from a lower digit, the carry-down information is forwarded to upper digit.
=> See page "Double length operation."

F-12w CMP — Compare register with register (1 word) (CoMPare)

Symbol	F-12w CMP | S_1 | S_2
Function	The word contents of the registers S_1, S_1+1 are compared with the word contents of the registers S_2, S_2+1.
Operation	S_1, S_1+1< = >S_2, S_2+1→Flag
S_1	Use range B *
S_2	Use range B *
Condition	When the input signal is ON state (not limited to OFF to ON change)

Contents after operation

S_1, S_1+1	Unchanged
S_2, S_2+1	Unchanged

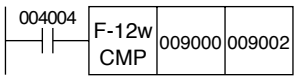
Flag

Contents of register	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
S_1, S_1+1>S_2, S_2+1	0	0	0	1
S_1, S_1+1=S_2, S_2+1	1	0	0	1
S_1, S_1+1<S_2, S_2+1	0	1	0	0

* Be sure to use even addresses for registers S_1 and S_2.

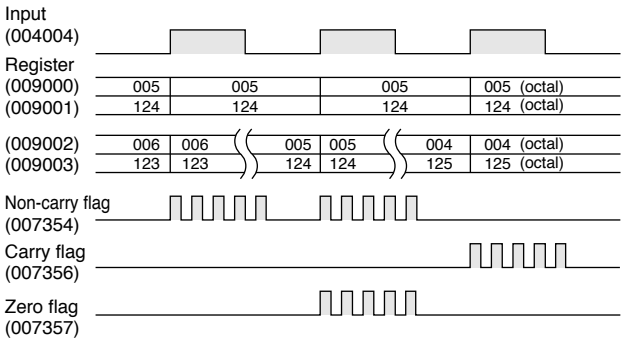
Resembled instructions: F-12, F-12d, Fc12, Fc12w, Fc12d, F-112, F-112w, F-112d

[Explanation]



Instruction	
S T R	004004
F -12w	
	009000
	009002

When the input condition 004004 is ON, the word contents of the registers 009000 and 009001 are compared with the word contents of the registers 009002 and 09003 and its results are set in the non-carry flag (007354), carry flag (007356), and the zero flag (007357). registers 009000 to 009003 remain unchanged.



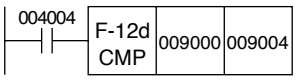
F-12d CMP — Compare register with register (2 words) (CoMPare)

Symbol		F-12d CMP | S_1 | S_2
Function		The word contents of the registers S_1 to S_1+3 are compared with the word contents of the registers S_2 to S_2+3.
Operation		S_1 to S_1+3< = >S_2 to S_2+3→Flag
S_1		Use range C *
S_2		Use range C *
Condition		When the input signal is ON (not limited to OFF to ON change)
Contents after operation	S_1 to S_1+3	Unchanged
	S_2 to S_2+3	Unchanged

Contents after operation: Flag — Contents of register	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
S_1 to S_1+3>S_2 to S_2+3	0	0	0	1
S_1 to S_1+3=S_2 to S_2+3	1	0	0	1
S_1 to S_1+3<S_2 to S_2+3	0	1	0	0

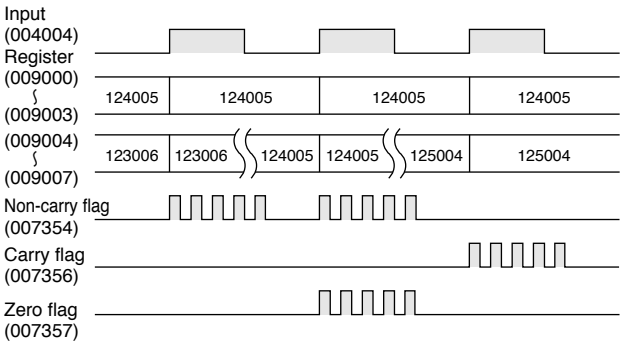
* Be sure to use even addresses for registers S_1 and S_2.
Resembled instructions: F-12, F-12w, Fc12, Fc12w, Fc12d, F-112, F-112w, F112d

[Explanation]



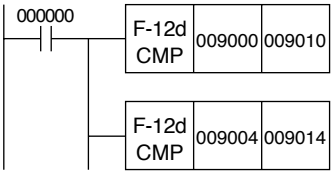
Instruction	
S T R	004004
F -12d	
	009000
	009004

When the input condition 004004 is ON, the 2 words contents of the registers 009000 to 009003 are compared with the 2-word contents of the registers 009004 to 009007 and its results are set in the non-carry flag (007354), carry flag (007356), and the zero flag (007357). Registers 009000 to 009007 remain unchanged after this operation.



Reference

If the F-12d instruction is used in succession, 8 bytes or more data comparison may be done.



Instruction	
S T R	000000
F -12d	
	009000
	009010
F -12d	
	009004
	009014

Fc12 CMP — Compare register with octal constant (1 byte) (CoMPare)

Symbol	Fc12 CMP | S_1 | n
Function	The contents of the register S_1 are compared with an octal constant "n".
Operation	S_1＜＝＞n→Flag
S_1	Use range A
n	Use range 000 to 377(8)
Condition	When the input signal is ON (not limited to OFF to ON change)

Contents after operation:

		Contents of register	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
S_1		Unchanged				
Flag *		S_1＞n	0	0	0	1
		S_1＝n	1	0	0	1
		S_1＜n	0	1	0	0

* Error flag (007355) will be already "0."

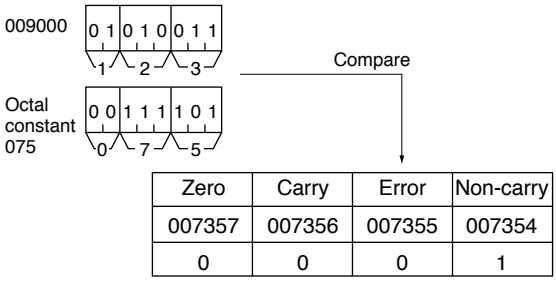
Resembled instructions: F-12, F-12w, F-12d, Fc12w, Fc12d, F-112, F-112w, F-112d

[Explanation]

Instruction	
S T R	04001
F c12	
	09000
	075

004001 | Fc12 CMP | 009000 | 075

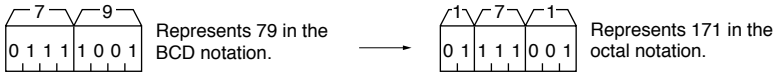
When the input condition 004001 is ON, the contents of the register 009000 are compared with the octal constant 075 and its results are set in the non-carry flag (007354) , carry flag (007356) , and zero flag (007357). The contents of the register 009000 remain unchanged after this operation.



Zero	Carry	Error	Non-carry
007357	007356	007355	007354
0	0	0	1

Reference

Use an octal number in writing program with the Fc12 instruction. Octal number can express any bit pattern and it dose not require annoying weight calculation. To compare with a BCD constant, convert it to an octal equivalent before writing it in the program.



Fc12w CMP — Compare register with octal constant (1 word) (CoMPare)

Symbol	Fc12w CMP | S_1 | n
Function	The word contents of the registers S_1 and S_1+1 are compared with an octal constant "n".
Operation	S_1, S_1+1＜＝＞n→Flag
S_1	Use range B *
n	Use range 000000 to 177777(8)
Condition	When the input signal in the ON (not limited to OFF to ON change)

Contents after operation:

		Contents of register	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
S_1, S_1+1		Unchanged				
Flag		S_1, S_1+1＞n	0	0	0	1
		S_1, S_1+1＝n	1	0	0	1
		S_1, S_1+1＜n	0	1	0	0

* Be sure to use even addresses for register S_1.
(Odd address such as ㅋ00011 etc. are prohibited to use.)

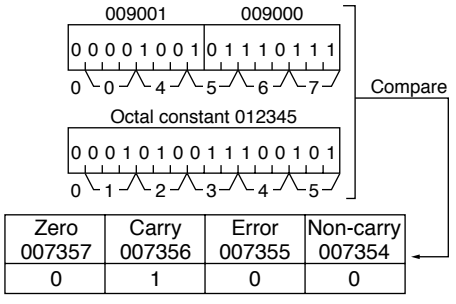
Resembled instructions: F-12, F-12w, F-12d, Fc12, Fc12d, F-112, F-112w

[Explanation]

Instruction	
S T R	002000
F c12w	
	009000
	012345

002000 | Fc12w CMP | 009000 | 012345

When the input condition 002000 is ON, the word contents of the registers 009000 and 009001 are compared with the octal constant 012345 and its results are set in the non-carry flag, carry flag, and zero flag. The contents of the registers 009000, 009001 remain unchanged. Operation takes place in the same timing as F-12w.



Zero	Carry	Error	Non-carry
007357	007356	007355	007354
0	1	0	0

Fc12d CMP — Compare register with octal constant (2 words) (CoMPare)

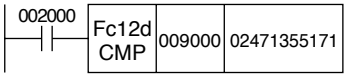
Symbol		Fc12d CMP | S_1 | n
Function		The 2-words contents of the registers S_1 to S_1+3 are compared with an octal constant "n."
Operation		S_1 to S_1+3＜＝＞n→Flag
S_1		Use range C *
S_2		Use range 000000000000 to 37777777777(8)
Condition		When the input signal in the ON (not limited to OFF to ON change)
Contents after operation	S_1 to S_1+3	Unchanged

Contents after operation: Flag	Contents of register	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	S_1 to S_1+3＞n	0	0	0	1
	S_1 to S_1+3＝n	1	0	0	1
	S_1 to S_1+3＜n	0	1	0	0

* Be sure to use even addresses for register S_1.
(Odd address such as ㅋ00011 etc. are prohibited to use.)

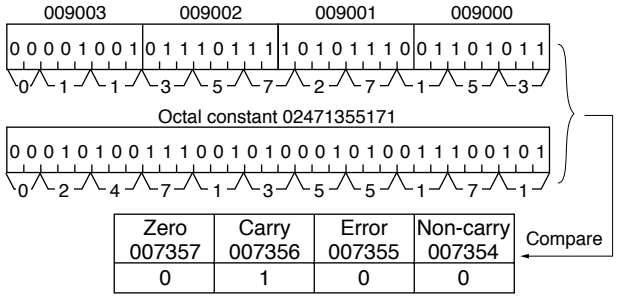
Resembled instructions: F-12, F-12w, F-12d, Fc12, Fc12w, F-112, F-112w

[Explanation]



Instruction	
S T R	002000
F c12d	
	009000
	02471355171

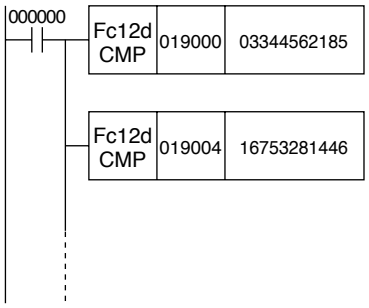
When the input condition 002000 is ON, the word contents of the registers 009000 to 009003 are compared with the octal constant 02471355171 and its results are set in the non-carry flag (007354), carry flag (007356), and zero flag (007357). The contents of the registers 009000 to 009003 remain unchanged. Operation takes place in the same timing as F-12d.



Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0

Reference

If the Fc12d instruction is used in succession, 8 bytes or more data comparison may be done.



Instruction	
S T R	000000
F c12d	
	019000
	03344562185
F c12d	
	019002
	16753281446

Fx12 CMP — Compare register with hexadecimal constant (1 byte) (CoMPare)

Symbol	Fx12 CMP | S_1 | n
Function	The contents of the register S_1 are compared with a hexadecimal constant "n."
Operation	S_1＜＝＞n→Flag
S_1	Use range A
n	Use range 00 to FF
Condition	When the input signal is ON (not limited to an OFF to ON change)

Contents after operation:

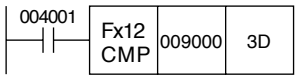
S_1	Unchanged					
Flag *	Contents of register	Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
	S_1＞n	0	0	0	1	
	S_1＝n	1	0	0	1	
	S_1＜n	0	1	0	0	

* Error flag (007355) will be always "0."

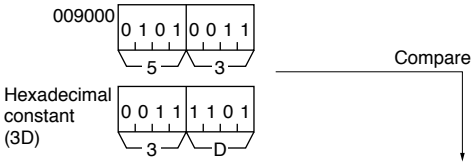
Resembled instructions: F-12, F-12w, F-12d, Fc12w, Fc12d, Fx12, Fx12w, Fx12d, F-112, F-112w, F-112d

[Explanation]

Instruction	
S T R	004001
F x12	
	009000
	3D



When the input condition 004001 is ON, the contents of the register 009000 are compared with the hexadecimal constant 3D and its results are set in the non-carry flag, carry flag, and zero flag. The contents of the registers 009000 remain unchanged after this operation.



Zero	Carry	Error	Non-carry
007357	007356	007355	007354
0	0	0	1

Fx12w CMP — Compare register with hexadecimal constant (1 word) (CoMPare)

Symbol	Fx12w CMP | S_1 | n
Function	The word contents of the register S_1 and S_1+1 are compared with a hexadecimal constant "n."
Operation	S_1, S_1+1＜＝＞n→Flag
S_1	Use range B - Be sure to use even addresses for register S_1. (Odd address such as ㅋ00011 and etc. are prohibited to use.)
n	Use range 0000 to FFFF(H)
Condition	When the input signal is ON (not limited to an OFF to ON change)

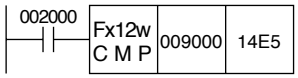
Contents after operation:

S_1, S_1+1	Unchanged					
Flag	Contents of register	Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
	S_1, S_1+1＞n	0	0	0	1	
	S_1, S_1+1＝n	1	0	0	1	
	S_1, S_1+1＜n	0	1	0	0	

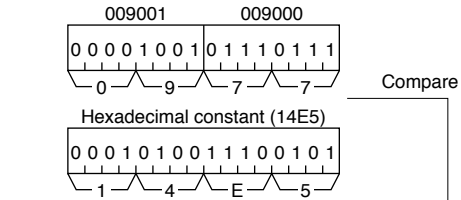
Resembled instructions: F-12, F-12w, F-12d, Fc12, Fc12w, Fc12d, Fx12, Fx12d, F-112, F-112w

[Explanation]

Instruction	
S T R	002000
F x12w	
	009000
	14E5



When the input condition 002000 is ON, the word contents of the register 009000 and 009001 are compared with the hexadecimal constant 14E5 and its results are set in the non-carry flag, carry flag, and zero flag. The contents of the registers 009000, 009001 remain unchanged. Operation takes place in the same timing as F-12w.



Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0

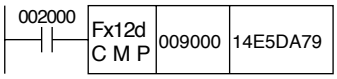
Fx12d CMP — Compare register with hexadecimal constant (2 words) (CoMPare)

Symbol		Fx12d CMP | S1 | n
Function		The 2-words contents of the register S1 to S1+3 are compared with a hexadecimal constant "n."
Operation		S1, S1+3＜＝＞n→Flag
S1		Use range C - Be sure to use even addresses for register S1. (Odd address such as ㅋ00011 and etc. are prohibited to use.)
n		Use range 00000000 to FFFFFFFF(H)
Condition		When the input signal is ON (not limited to an OFF to ON change)
Contents after operation	S1 to S1+3	Unchanged

Contents after operation: Flag — Contents of register	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
S1 to S1+3＞n	0	0	0	1
S1 to S1+3＝n	1	0	0	1
S1 to S1+3＜n	0	1	0	0

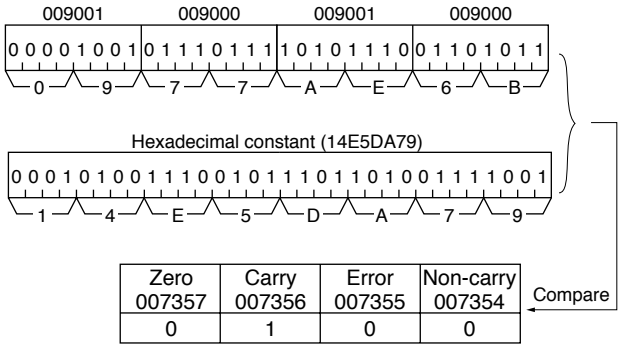
The following F instructions have similar functions:
F-12, F-12w, F-12d, Fc12, Fc12w, Fc12d, Fx12, Fx12w, F-112, F-112w

[Explanation]



Instruction	
S T R	002000
F x12d	
	009000
	14E5DA79

When the input condition 002000 is ON, the 2 word contents of the register 009000 to 009003 are compared with the hexadecimal constant 14E5DA79 and its results are set in the non-carry flag, carry flag, and zero flag. The contents of the registers 009000 to 009003 remain unchanged. Operation takes place in the same timing as F-12d.

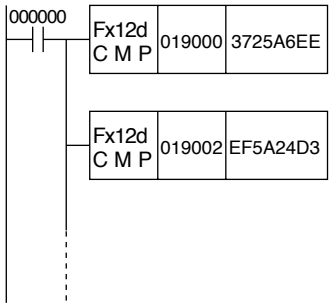


Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0

Compare

Reference

If the Fx12d instruction is used in succession, 8 bytes or more data comparison may be done.

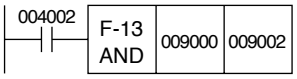


Instruction	
S T R	000000
F x12d	
	019000
	3725A6EE
F x12d	
	019002
	EF5A24D3

F-13 AND — ANDs register with register (1 byte) (AND)

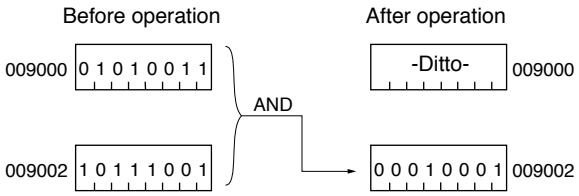
Symbol	F-13 AND | S | D	
Function	The contents of the register S (8 bits) are ANDed with the contents of the register D (8 bits) and its result is stored in the register D.	
Operation	S∩D→D	
S	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Result
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004002
F -13	
	009000
	009002

When the input condition 004002 changes from OFF to ON, the 8-bit contents of the register 009000 are ANDed with the 8-bit contents of the register 009002 and its result is stored in the register 009002. The contents of the register 009000 remain unchanged.



- AND truth table

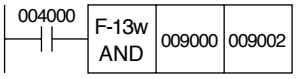
Symbol	A	B	C
A, B → C	0	0	0
	1	0	0
	0	1	0
	1	1	1

F-13w AND — AND register with register (1 word) (AND)

Symbol	F-13w AND | S | D	
Function	The 16-bit contents of the registers S, S+1 are ANDed with the 16-bit contents of the registers D, D+1 and its result is stored in the registers D, D+1.	
Operation	S, S+1∩D, D+1→D, D+1	
S	Use range B *	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1	Unchanged
	D, D+1	Result
	Flag	Unchanged

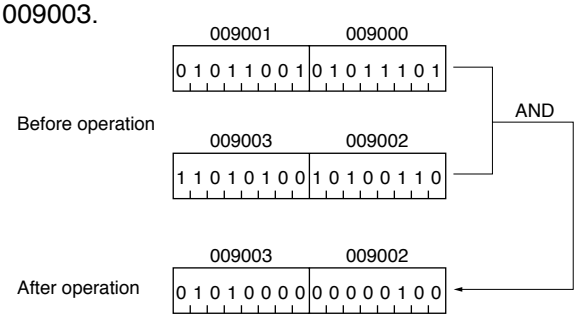
* Be sure to use even addresses for registers S and D. (Odd address such as 019003 etc. are prohibited to use.)

[Explanation]



Instruction	
S T R	004000
F -13w	
	009000
	009002

When the input condition 004000 changes from OFF to ON, the 16 bits contents of the registers 009000 and 009001 are ANDed with the 16-bit contents of the registers 009002 and 009003 and its result is stored in the registers 009002 and 009003.



The contents of the registers 009000 and 009001 remain unchanged.

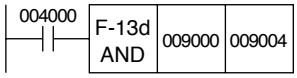
F-13d
AND

ANDs register with register (2 words) (AND)

Symbol	F-13d AND | S | D	
Function	The contents of the registers S to S+3 (32 bits) are ANDed with the contents of the registers D to D+3 (32 bits) and its result is stored in the registers D to D+3.	
Operation	S to S+3 ∩ D to D+3→D to D+3	
S	Use range C *	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S to S+3	Unchanged
	D to D+3	Result
	Flag	Unchanged

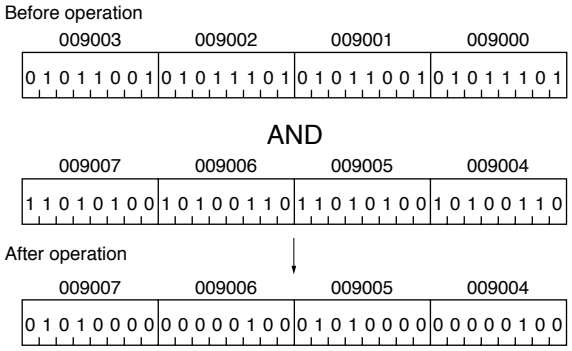
* Be sure to use even addresses for registers S and D. (Odd address such as 019003 etc. are prohibited to use.)

[Explanation]



Instruction	
S T R	004000
F -13d	
	009000
	009004

When the input condition 004000 changes from OFF to ON, the 32 bits contents of the registers 009000 to 009003 are ANDed with the 32-bit contents of the registers 009004 to 009007 and its result is stored in the registers 009004 to 009007.



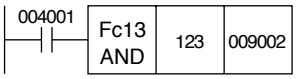
The contents of the registers 009000 to 009003 remains unchanged.

Fc13 AND — AND register with octal constant (1 byte) (AND)

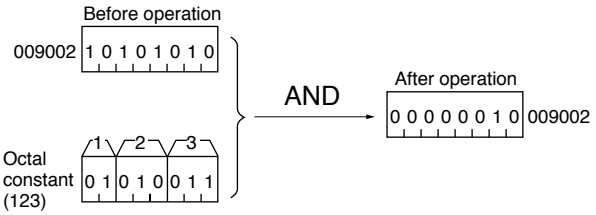
Symbol	Fc13 AND | n | D	
Function	An octal constant "n" is ANDed with the contents of the register D and its result is stored in the register D.	
Operation	n∩D→D	
n	Use range 000 to 377	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D	Result
	Flag	Unchanged

[Explanation]

Instruction	
S T R	004001
F c13	
	123
	009002



When the input condition 004001 changes from OFF to ON, the octal constant 123 is ANDed with the contents of the register 009002 and its result is stored in the register 009002.



- AND truth table

Symbol	A	B	C
A, B → C	0	0	0
	1	0	0
	0	1	0
	1	1	1

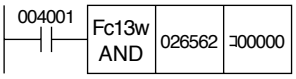
Fc13w AND — AND register with octal constant (1 word) (AND)

Symbol	Fc13w AND | n | D	
Function	An octal constant "n" is ANDed with the 16-bit contents of the registers D and D+1 and its result is stored in the registers D and D+1.	
Operation	n∩D, D+1→D, D+1	
n	Use range 000000 to 177777(8)	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D, D+1	Result
	Flag	Unchanged

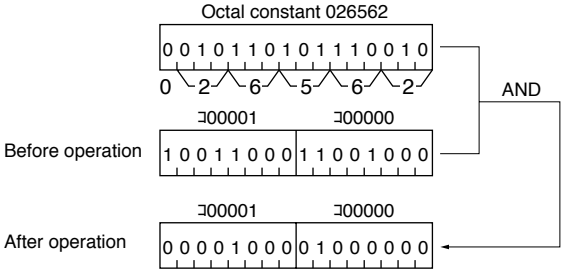
* Be sure to use even addresses for register D.
(Odd address such as ⊐00011 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	004001
F c13w	
	026562
	⊐00000



When the input condition 004001 changes from OFF to ON, the octal constant 026562 is ANDed with the16 bits contents of the registers ⊐00000 and ⊐00001 and its result is stored in the registers ⊐00000 and ⊐00001.

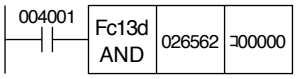


Fc13d AND — AND register with octal constant (2 words) (AND)

Symbol	Fc13d AND | n | D	
Function	An octal constant "n" is ANDed with the 32-bit contents of the registers D to D+3 and its result is stored in the registers D to D+3.	
Operation	n∩D to D+3→D to D+3	
n	Use range 00000000000 to 37777777777(8)	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D to D+3	Result
	Flag	Unchanged

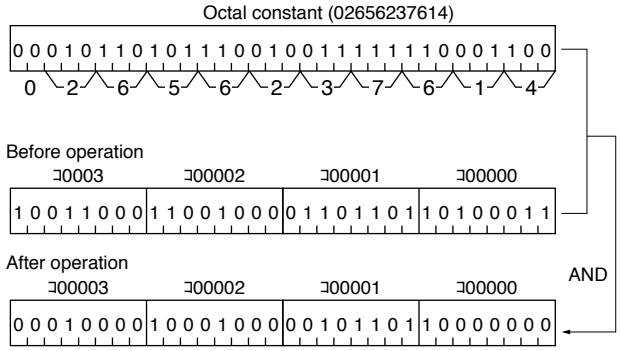
* Be sure to use even addresses for register D.
(Odd address such as ⊐00011 etc. are prohibited to use.)

[Explanation]



Instruction	
S T R	004001
F c13d	
	02656237614
	⊐00000

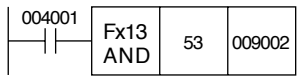
When the input condition 004001 changes from OFF to ON, the octal constant 02656237614 is ANDed with the 32 bits contents of the registers ⊐00000 to ⊐00003 and its result is stored in the registers ⊐00000 to ⊐00003.



Fx13 AND — AND register with hexadecimal constant (1 byte) (AND)

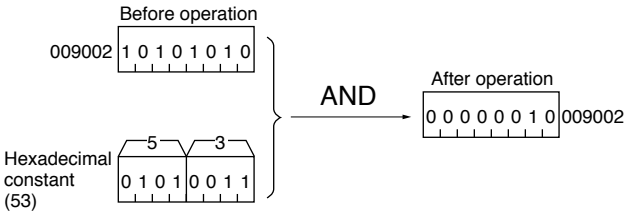
Symbol	Fx13 AND | n | D	
Function	A hexadecimal constant "n" is ANDed with the contents of the register D and its result is stored in the register D.	
Operation	n ∩ D→D	
n	Use range 00 to FF	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D	Result
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004001
F x13	
	53
	009002

When the input condition 004001 changes from OFF to ON, the hexadecimal constant 53 is ANDed with the contents of the register 009002 and its result is stored in the register 009002.



- AND truth table

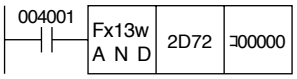
Symbol	A	B	C
A, B → C	0	0	0
	1	0	0
	0	1	0
	1	1	1

Fx13w AND — AND register with hexadecimal constant (1 word) (AND)

Symbol	Fx13w AND | n | D	
Function	A hexadecimal constant "n" is ANDed with the 16-bit contents of the registers D, D+1 and its result is stored in the registers D, D+1.	
Operation	n ∩ D, D+1→D, D+1	
n	Use range 0000 to FFFF(H)	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D, D+1	Result
	Flag	Unchanged

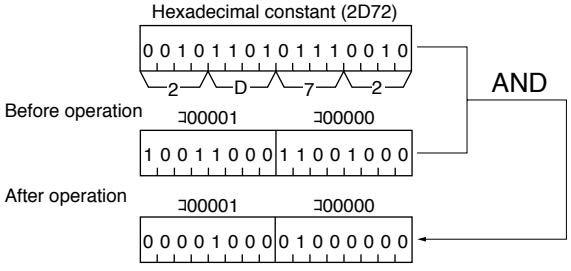
* Be sure to use even addresses for register D.
(Odd address such as ↄ00011 etc. are prohibited to use.)

[Explanation]



Instruction	
S T R	004001
F x13w	
	2D72
	ↄ00000

When the input condition 004001 changes from OFF to ON, the hexadecimal constant 2D72 is ANDed with the 16 bits contents of the registers ↄ00000 and ↄ00001 and its result is stored in the registers ↄ00000 and ↄ00001.



Fx13d
AND

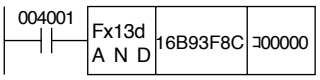
AND register with hexadecimal constant (2 words) (AND)

Symbol	Fx13d A N D | n | D	
Function	A hexadecimal constant "n" is ANDed with the 32-bit contents of the registers D to D+3 and its result is stored in the registers D to D+3.	
Operation	n ∩ D to D+3→D to D+3	
n	Use range 00000000 to FFFFFFFF(H)	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D to D+3	Result
	Flag	Unchanged

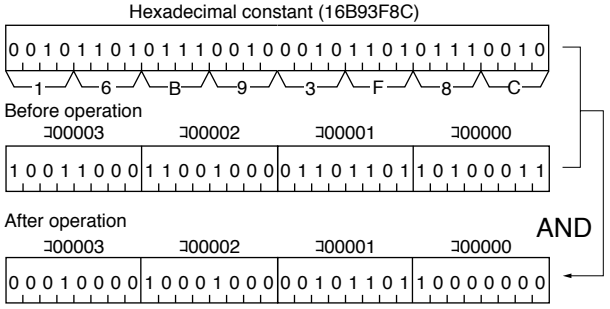
* Be sure to use even addresses for register D.
(Odd address such as ↄ00011 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	004001
F x13d	
	16B93F8C
	ↄ00000



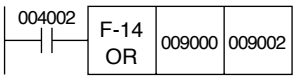
When the input condition 004001 changes from OFF to ON, the hexadecimal constant 16B93F8C is ANDed with the 32 bits contents of the registers ↄ00000 to ↄ00003 and its result is stored in the registers ↄ00000 to ↄ00003.



F-14 OR — OR register with register (1 byte) (OR)

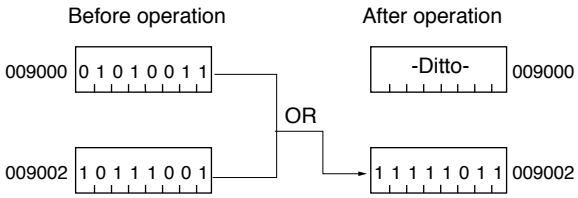
Symbol	F-14 OR | S | D	
Function	The contents of the register S (8 bits) are ORed with the contents of the register D (8 bits) and its result is stored in the register D.	
Operation	S∪D→D	
S	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Result
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004002
F -14	
	009000
	009002

When the input condition 004002 changes from OFF to ON, the 8 bits contents of the register 009000 are ORed with the 8 bits contents of the register 009002 and its result is stored in the register 009002. The contents of the register 009000 remain unchanged.



- OR truth table

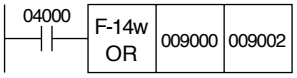
Symbol	A	B	C
A, B → C	0	0	0
	1	0	1
	0	1	1
	1	1	1

F-14w OR — OR register with register (1 word) (OR)

Symbol	F-14w OR | S | D	
Function	The 16-bit contents of the registers S, S+1 are ORed with the 16-bit contents of the registers D, D+1 and its result is stored in the registers D, D+1.	
Operation	S, S+1∪D, D+1→D, D+1	
S	Use range B *	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1	Unchanged
	D, D+1	Result
	Flag	Unchanged

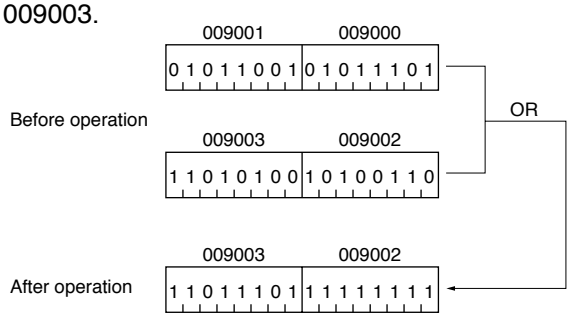
* Be sure to use even addresses for registers S and D. (Odd address such as 019003 etc. are prohibited to use.)

[Explanation]



Instruction	
S T R	004000
F -14w	
	009000
	009002

When the input condition 004000 changes from OFF to ON, the 16-bit contents of the registers 009000 and 009001 are ORed with the 16-bit contents of the registers 009002 and 009003 and its result is stored in the registers 009002 and 009003.



The contents of the registers 009000 and 009001 remain unchanged.

F-14d
OR

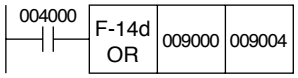
OR register with register (2 words) (OR)

Symbol	F-14d OR | S | D	
Function	The contents of the registers S to S+3 (32 bits) are ORed with the contents of the registers D to D+3 (32 bits) and its result is stored in the registers D to D+3.	
Operation	S to S+3 ∪ D to D+3 → D to D+3	
S	Use range C *	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S to S+3	Unchanged
	D to D+3	Result
	Flag	Unchanged

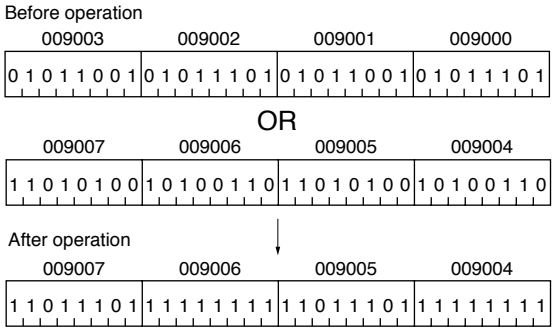
* Be sure to use even addresses for registers S and D. (Odd address such as 019003 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	004000
F -14d	
	009000
	009004



When the input condition 004000 changes from OFF to ON, the 32-bit contents of the registers 009000 to 009003 are ORed with the 32-bit contents of the registers 009004 to 009007 and its result is stored in the registers 009004 to 009007.



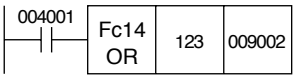
The contents of the registers 009000 to 009003 remain unchanged.

Fc14 OR — OR register with octal constant (1 byte) (OR)

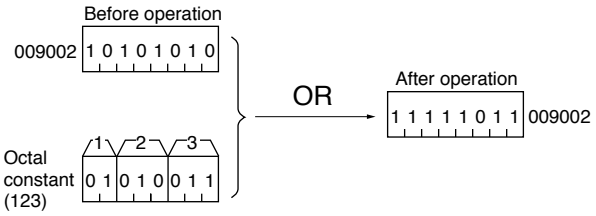
Symbol	Fc14 OR | n | D	
Function	An octal constant "n" is ORed with the contents of the register D and its result is stored in the register D.	
Operation	n ∪ D→D	
n	Use range 000 to 377	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D	Result
	Flag	Unchanged

[Explanation]

Instruction	
S T R	004001
F c14	
	123
	009002



When the input condition 004001 changes from OFF to ON, the octal constant 123 is ORed with the contents of the register 009002 and its result is stored in the register 009002.



- OR truth table

Symbol	A	B	C
A, B → C	0	0	0
	1	0	1
	0	1	1
	1	1	1

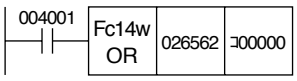
Fc14w OR — OR register with octal constant (1 word) (OR)

Symbol	Fc14w OR | n | D	
Function	An octal constant "n" is ORed with the 16-bit contents of the registers D and D+1 and its result is stored in the registers D and D+1.	
Operation	n ∪ D, D+1→D, D+1	
n	Use range 000000 to 177777(8)	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D, D+1	Result
	Flag	Unchanged

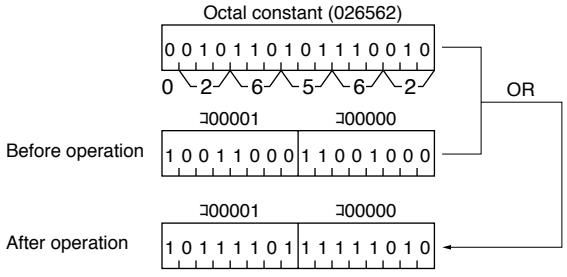
* Be sure to use even addresses for register D.
(Odd address such as ⊐00011 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	004001
F c14w	
	026562
	⊐00000



When the input condition 004001 changes from OFF to ON, the octal constant 026562 is ORed with the 16 bits contents of the registers ⊐00000 and ⊐00001 and its result is stored in the registers ⊐00000 and ⊐00001.



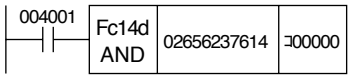
Fc14d
OR

OR register with octal constant (2 words) (OR)

Symbol	Fc14d OR | n | D	
Function	An octal constant "n" is ORed with the 32-bit contents of the registers D to D+13 and its result is stored in the registers D to D+3.	
Operation	n ∪ D to D+3→D to D+3	
n	Use range 00000000000 to 37777777777(8)	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D to D+3	Result
	Flag	Unchanged

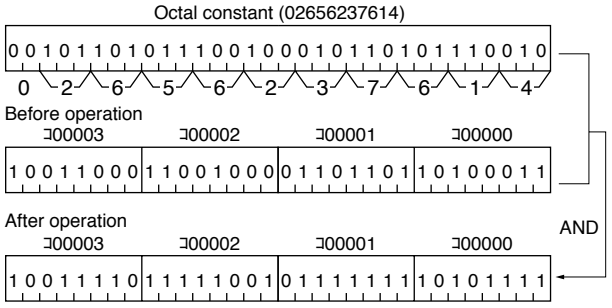
* Be sure to use even addresses for register D.
(Odd address such as ⊐00011 etc. are prohibited to use.)

[Explanation]



Instruction	
S T R	004001
F c14d	
	02656237614
	⊐00000

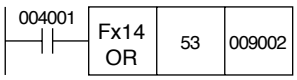
When the input condition 004001 changes from OFF to ON, the octal constant 02656237614 is ORed with the 16 bits contents of the registers ⊐00000 to ⊐00003 and its result is stored in the registers ⊐00000 to ⊐00003.



Fx14 OR — OR register with hexadecimal constant (1 byte) (OR)

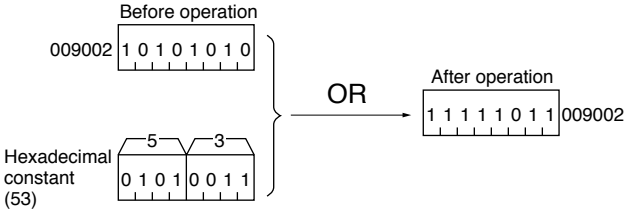
Symbol	Fx14 OR | n | D	
Function	A hexadecimal constant "n" is ORed with the contents of the register D and its result is stored in the register D.	
Operation	n ∪ D → D	
n	Use range 00 to FF	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D	Result
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004001
F x14	
	53
	009002

When the input condition 004001 changes from OFF to ON, the hexadecimal constant 53 is ORed with the contents of the register 009002 and its result is stored in the register 009002.



- OR truth table

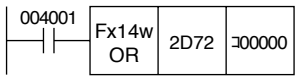
Symbol	A	B	C
A, B → C	0	0	0
	1	0	1
	0	1	1
	1	1	1

Fx14w OR — OR register with hexadecimal constant (1 word) (OR)

Symbol	Fx14w OR | n | D	
Function	A hexadecimal constant "n" is ORed with the 16-bit contents of the registers D and D+1 and its result is stored in the registers D and D+1.	
Operation	n ∪ D, D+1→D, D+1	
n	Use range 0000 to FFFF(H)	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D, D+1	Result
	Flag	Unchanged

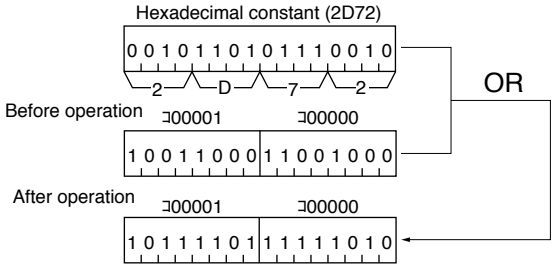
* Be sure to use even addresses for registers D.
(Odd address such as ⊐00011 etc. are prohibited to use.)

[Explanation]



Instruction	
S T R	004001
F x14w	
	2D72
	⊐00000

When the input condition 004001 changes from OFF to ON, the hexadecimal constant 2D72 is ORed with the 16-bit contents of the registers ⊐00000 and ⊐00001 and its result is stored in the registers ⊐00000 and ⊐00001.

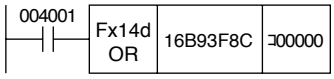


Fx14d OR — OR register with hexadecimal constant (2 words) (OR)

Symbol	Fx14d OR | n | D	
Function	A hexadecimal constant "n" is ORed with the 32-bit contents of the registers D to D+3 and its result is stored in the registers D to D3.	
Operation	n ∪ D to D+3→D to D+3	
n	Use range 00000000 to FFFFFFFF(H)	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D to D+3	Result
	Flag	Unchanged

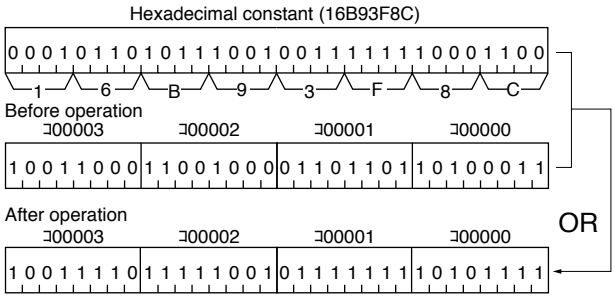
* Be sure to use even addresses for registers D.
(Odd address such as ㅋ00011 etc. are prohibited to use.)

[Explanation]



Instruction	
S T R	004001
F x14d	
	16B93F8C
	ㅋ00000

When the input condition 004001 changes from OFF to ON, the hexadecimal constant 16B93F8C is ORed with the 32-bit contents of the registers ㅋ00000 to ㅋ00003 and its result is stored in the registers ㅋ00000 to ㅋ00003.

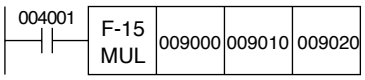


Chapter 11 Application instructions (F-15 to F-49)

F-15 MUL — Multiplies register by register (BCD 4 digits) (MULtiply)

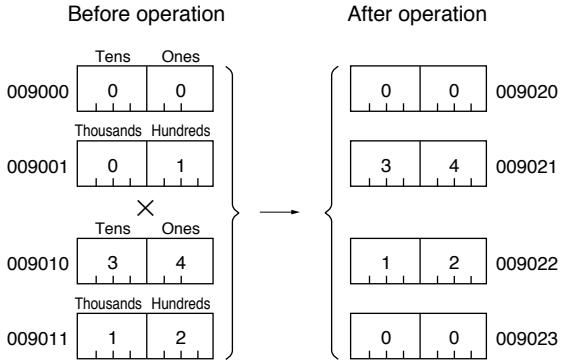
Symbol	F-15 MUL | S_1 | S_2 | D					
Function	The contents of the registers S_1 and S_1+1 (BCD 4 digits) are multiplied by the contents of the registers S_2 and S_2+1 (BCD 4 digits) and its result is stored in the 4-byte area starting from the register D.					
Operation	(S_1, S_1+1) × (S_2, S_2+1) →D, D+1, D+2, D+3					
S_1	Use range B					
S_2	Use range B					
D	Use range C					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	S_1, S_1+1	Unchanged				
	S_2, S_2+1	Unchanged				
	D	Result in ones and tens	Unchanged if the contents of registers S_1, S_1+1, S_2, and S_2+1 are not BCD code. (Does not calculate.)			
	D+1	Result in hundreds and thousands				
	D+2	Result in ten thousands and hundred thousands				
	D+3	Result in millions and ten millions				
	Flag	Register S_1, S_1+1, S_2, S_2+1	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		BCD code	0	0	0	0
		Not BCD code			1	

[Explanation]



Instruction	
S T R	004001
F -15	
	009000
	009010
	009020

When the input condition 004001 changes from OFF to ON, the BCD 4 digits in registers 009000 and 009001 are multiplied by the BCD 4 digits of registers 009010 and 009011 and its result is stored in the 4-byte area starting from the register 009020.



The above example shows the operation of 100×1234=123400.

F-15d MUL — Multiplies register by register (BCD 8 digits) (MULtiply)

Symbol	F-15d MUL | S_1 | S_2 | D
Function	The contents of the registers S_1 to S_1+3 (BCD 8 digits) are multiplied by the contents of the registers S_2 to S_2+3 (BCD 8 digits) and its result is stored in the 8-byte area starting from the register D.
Operation	(S_1 to S_1+3) × (S_2 to S_2+3)→D to D+7
S_1	Use range C *
S_2	Use range C *
D	Use range G *
Condition	Rising edge of input signal (OFF to ON)

Contents after operation		
	S_1 to S_1+3	Unchanged
	S_2 to S_2+3	Unchanged
	D to D+7	Operation result (MSB | LSB): D: 10^1 | 10^0; D+1: 10^3 | 10^2; D+2: 10^5 | 10^4; D+3: 10^7 | 10^6; D+4: 10^9 | 10^8; D+5: 10^{11} | 10^{10}; D+6: 10^{13} | 10^{12}; D+7: 10^{15} | 10^{14}. Unchanged when the contents of registers S_1 to S_1+3, S_2 to S_2+3 are not BCD code. (Does not calculate.)

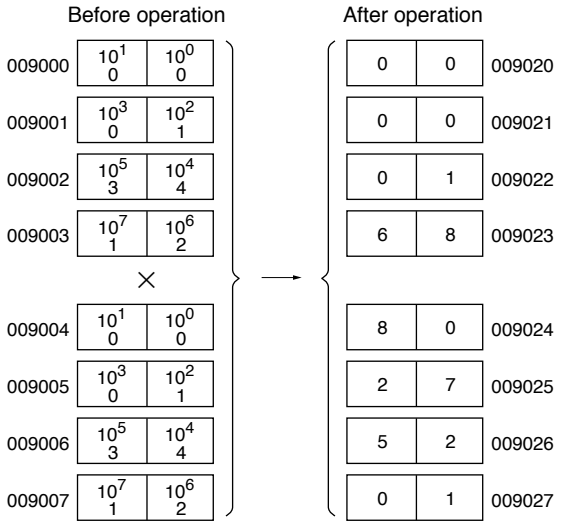
Flag	S_1 to S_1+3, S_2 to S_2+3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	BCD code	0	0	0	0
	Not BCD code			1	

[Explanation]

Instruction	
S T R	004001
F -15d	009000 009004 009020

004001 — F-15d MUL | 009000 | 009004 | 009020

When the input condition 004001 changes from OFF to ON, the BCD 8 digits in registers 009000 to 009003 are multiplied by the BCD 8 digits of registers 009004 to 009007 and its result is stored in the 8-byte area starting from the registers 009020.



The above example shows the operation of 12340100×12340100=152278068010000.

* Be sure to use even addresses for registers S_1, S_2, and D.
(Odd address such as 019003 etc. are prohibited to use.)

Fc15 MUL — Multiplies register (BCD 4 digits) by BCD constant (3 digits) (MULtiply)

Symbol	Fc15 MUL | S_1 | n | D
Function	The contents of the registers S_1 and S_1+1 (BCD 4 digits) are multiplied by a 3-digit BCD constant "n" and its result is stored in the 4-byte area starting from the register D.
Operation	(S_1, S_1+1) × n→D, D+1, D+2, D+3
S_1	Use range B
n	Use range 000 to 999
D	Use range C
Condition	Rising edge of input signal (OFF to ON)

Contents after operation			
	S_1, S_1+1	Unchanged	
	D	Result in ones and tens	Unchanged when the contents of registers S_1 and S_1+1 are not BCD code. (Does not calculate.)
	D+1	Result in hundreds and thousands	
	D+2	Result in ten thousands and hundred thousands	
	D+3	Result in millions and ten millions	

Flag					
S_1, S_1+1	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
BCD code	0	0	0	0
Not BCD code			1	

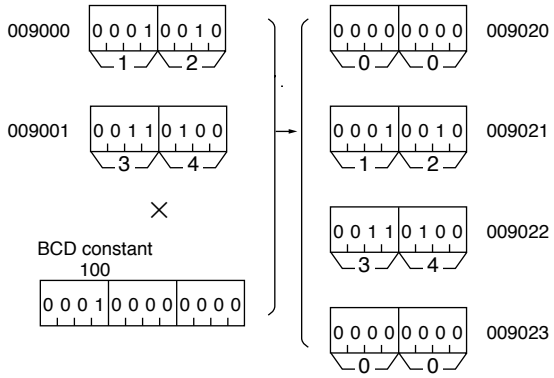
[Explanation]

Instruction	
S T R	004001
F c15	009000
	100
	009020

004001 | Fc15 MUL | 009000 | 100 | 009020

When the input condition 004001 changes from OFF to ON, the BCD 4 digits in registers 009000 and 009001 are multiplied by the 3-digit BCD constant 100 and its result is stored in the 4-byte area starting from the registers 009020.

3412×100=341200



Fc15d MUL — Multiplies register (BCD 8 digits) by BCD constant (4 digits) (MULtiply)

Symbol	Fc15d MUL | S_1 | n | D
Function	The contents of the registers S_1 to S_1+3 (BCD 8 digits) are multiplied by a 4-digit BCD constant "n" and its result is stored in the 8-byte area starting from the register D.
Operation	(S_1 to S_1+3) × n→D to D+5
S_1	Use range C
n	Use range 0000 to 9999
D	Use range G
Condition	Rising edge of input signal (OFF to ON)

Contents after operation

S_1 to S_1+3	Unchanged
D to D+5	Operation result: D: 10^1 10^0; D+1: 10^3 10^2; D+2: 10^5 10^4; D+3: 10^7 10^6; D+4: 10^9 10^8; D+5: 10^{11} 10^{10}; D+6: 0 0; D+7: 0 0. Unchanged when the contents of registers S_1 to S_1+3 are not BCD code. (Does not calculate.)

Flag

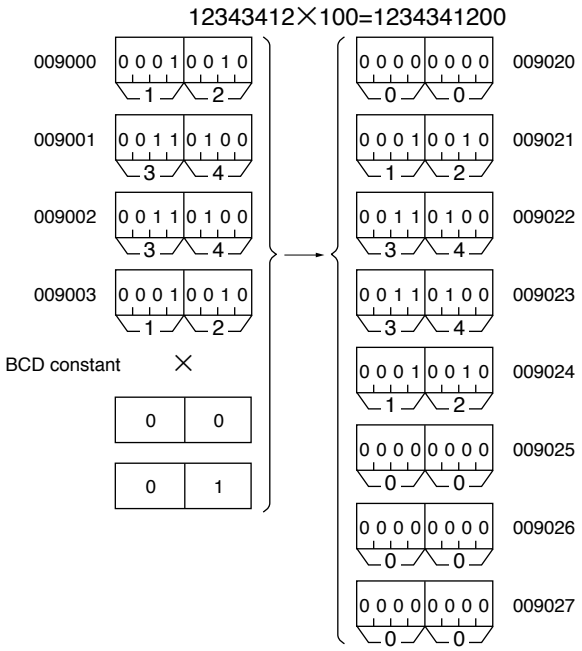
S_1 to S_1+3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
BCD code	0	0	0	0
Not BCD code			1	

[Explanation]

004001 — Fc15d MUL | 009000 | 0100 | 009020

Instruction	
S T R	004001
F c15d	009000
	0100
	009020

When the input condition 004001 changes from OFF to ON, the BCD 8 digits in registers 009000 and 009003 are multiplied by the 4-digit BCD constant 0100 and its result is stored in the 8-byte area starting from the registers 009020.



F-16 DIV — Divide register (BCD 4 digits) by register (BCD 2 digits) (DIVide)

Symbol	F-16 DIV | S_1 | S_2 | D
Function	The contents of the registers S_1 and S_1+1 (BCD 4 digits) are divided by the contents of the register S_2 (BCD 2 digits) and the quotient is stored in the 2-byte area starting from the register D and the remainder in the third byte area.
Operation	(S_1, S_1+1) ÷ S_2 → D, D+1, D+2
S_1	Use range B
S_2	Use range A
D	Use range E
Condition	Rising edge of input signal (OFF to ON)

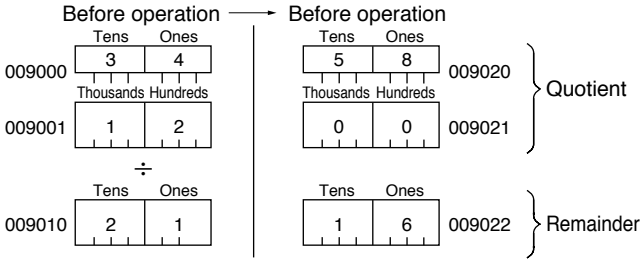
Contents after operation						
S_1, S_1+1	Unchanged					
S_2	Unchanged					
D	Quotient (in ones and tens)	Unchanged when the contents of registers, S_1, S_1+1, and S_2 are not BCD code or the contents of S_2 is 00. (Does not calculate.)				
D+1	Quotient (in hundreds and thousands)					
D+2	Remainder					
Flag	S_1, S_1+1, S_2	Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
	BCD code	0	0	0	0	
	Not BCD code If S_2 is 00			1		

[Explanation]

Instruction	
S T R	004001
F -16	
	009000
	009010
	009020

004001 — F-16 DIV | 009000 | 009010 | 009020

When the input condition 004001 changes from OFF to ON, the BCD 4 digits in registers 009000 and 009001 are divided by the BCD 2 digits in registers 009010 and the quotient is stored in the 2-byte area starting from the registers 009020 with the remainder in the third byte area.



The above example shows the operation of 1234 divided by 21 equals 58 with the remainder of 16.

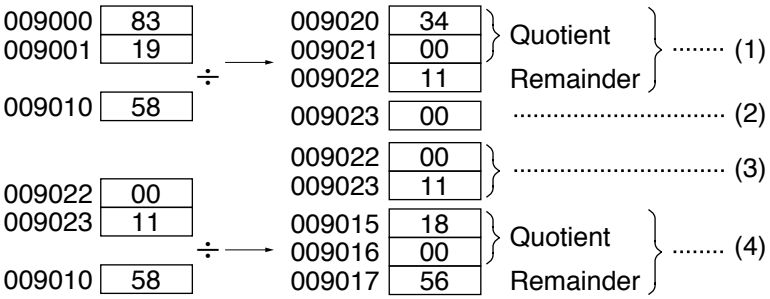
- If denominator is not greater than numerator (S_1<S_2, S_1+1=0), the quotient (contents of D, D+1) is 0 and remainder (contents of D+2) is numerator (contents of S_1). For instance, 20/30 will produce the result of 0 with a remainder of 20.

Reference To obtain a result of 2 places under the decimal point, the following programming is suggested.

Example：1983÷58=34.18 … remainder 0.56

004001

F-16 DIV	009000	009010	009020	..... (1)
F-01 BCD	00	009023		..... (2)
F-02 XCH	009022	009023		..... (3)
F-16 DIV	009022	009010	009015	..... (4)



Instruction	
S T R	004001
F -16	
	009000
	009010
	009020
F -01	
	00
	009023
F -02	
	009022
	009023
F -16	
	009022
	009010
	009015

(1) When the input condition 004001 changes from OFF to ON, the contents of registers 009000 and 009001 are divided by the contents of the register 009010 and the quotient is stored in 009020 and 009021 and remainder is stored in 009022.
(2) Enter data 00(H) in 009023.
(3) The contents of 009022 are exchanged with 009023 and a remainder is converted into thousands and hundreds.
(4) The data in (3) is divided by the contents of 009010 again, and a quotient is stored in 009015 and 009016 with a remainder stored in 009017.
The result under 2 digits of decimal place is then stored in 009015.

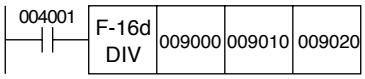
F-16d DIV — Divide register (BCD 8 digits) by register (BCD 8 digits) (DIVide)

Symbol	F-16d DIV	S_1	S_2	D
Function	The contents of the registers S_1 to S_1+3 (BCD 8 digits) are divided by the contents of the registers S_2 to S_2+3 (BCD 8 digits) and the quotient is stored in the register D to D+3 and the remainder in the D+4 to D+7.			
Operation	(S_1 to S_1+3) ÷ S_2 to S_2+3 → D to D+7			
S_1	Use range C *			
S_2	Use range C *			
D	Use range G *			
Condition	Rising edge of input signal (OFF to ON)			

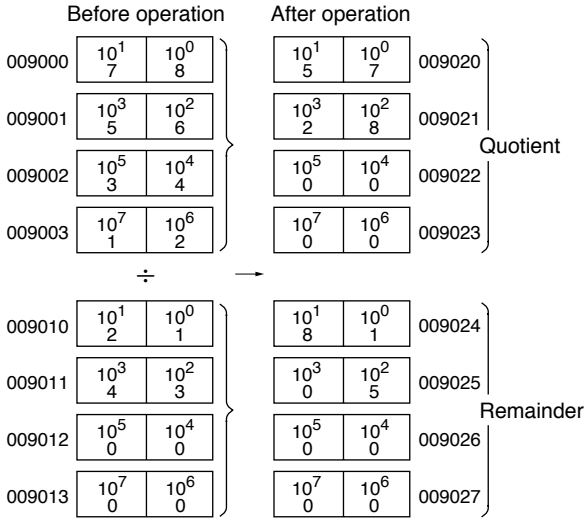
Contents after operation	S_1 to S_1+3	Unchanged				
	S_2 to S_2+3	Unchanged				
	D to D+3	Quotient (8-digit BCD)	Unchanged when the contents of registers S_1 to S_1+3, and S_2 to S_2+3 are not BCD code or the contents of S_2 to S_2+3 are 0. (Does not calculate.)			
	D+4 to D+7	Remainder (8-digit BCD)				
	Flag	S_1 to S_1+3, S_2 to S_2+3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		BCD code	0	0	0	0
		Not BCD code If S_2 to S_2+3 are 00000000(H)			1	

[Explanation]

Instruction	
S T R	004001
F -16d	009000
	009010
	009020



When the input condition 004001 changes from OFF to ON, the BCD 8 digits in registers 009000 to 009003 are divided by the BCD 4 digits in registers 009010 to 009013 and the quotient is stored in the registers 009020 to 009023 with the remainder in the 009024 to 009027.



The above example shows the operation of 12345678 divided by 4321 equals 2857 with the remainder of 581.

* Be sure to use even addresses for registers S_1 , S_2, and D.
(Odd address such as 019003 etc. are prohibited to use.)

- If denominator is not greater than numerator (S_1 to S_1+3 < S_2 to S_2+3), the quotient (contents of D to D+3) is 0 and remainder (contents of D+4 to D+7) is numerator (contents of S_1 to S_1+3).
  For instance, 20/30 will produce the result of 0 with a remainder of 20.

Fc16 DIV — Divide register (BCD 4 digits) by BCD constant (2 digits) (DIVide)

Symbol	Fc16 DIV | S_1 | n | D					
Function	The contents of the registers S_1 and S_1+1 (BCD 4 digits) are divided by a 2-digit BCD contents "n" and the quotient is stored in the 2-byte area starting from the register D and the remainder in the third byte area.					
Operation	(S_1, S_1+1) ÷ n → D, D+1, D+2					
S_1	Use range B					
n	Use range 00 to 99					
D	Use range E					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	S_1, S_1+1	Unchanged				
	D	Quotient (in ones and tens)	Unchanged when the contents of registers S_1 and S_1+1 are not BCD code or the contents of "n" is 00. (Does not calculate.)			
	D+1	Quotient (in hundreds and thousands)				
	D+2	Remainder				
	Flag	S_1, S_1+1, "n"	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		BCD code	0	0	0	0
		Not BCD code If "n" is 00		1		

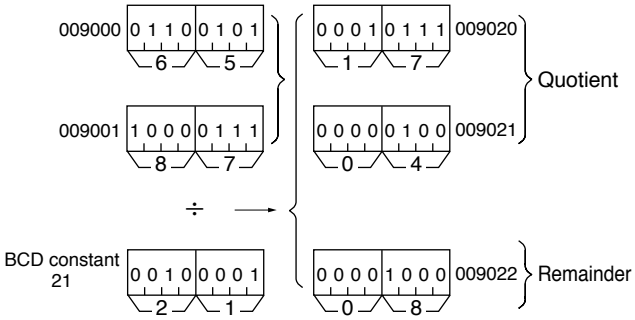
[Explanation]

Instruction	
S T R	004001
F c16	009000
	21
	009020

004001 | Fc16 DIV | 009000 | 21 | 009020

When the input condition 004001 changes from OFF to ON, the BCD 4 digits in registers 009000 and 009001 are divided by the BCD constant 21 and the quotient is stored in the 2-byte area starting from the register 009020 with the remainder in the third byte area.

8765÷21=417 remainder 8



- If denominator is not greater than numerator (S_1 < n, S_1+1=0), the quotient (contents of D, D+1) is 0 and remainder (contents of D+2) is numerator (contents of S_1). For instance, 20/30 will produce the result of 0 with a remainder of 20.

Fc16d DIV — Divide register (BCD 8 digits) by BCD constant (4 digits) (DIVide)

Symbol	Fc16d DIV | S_1 | n | D
Function	The contents of the registers S_1 to S_1+3 (BCD 8 digits) are divided by a 4-digit BCD contents "n" and the quotient is stored in the register D to D+3 with the remainder in the registers D+4 to D+7.
Operation	(S_1 to S_1+3) ÷ n → D to D+7
S_1	Use range C *
n	Use range 0000 to 9999
D	Use range G *
Condition	Rising edge of input signal (OFF to ON)

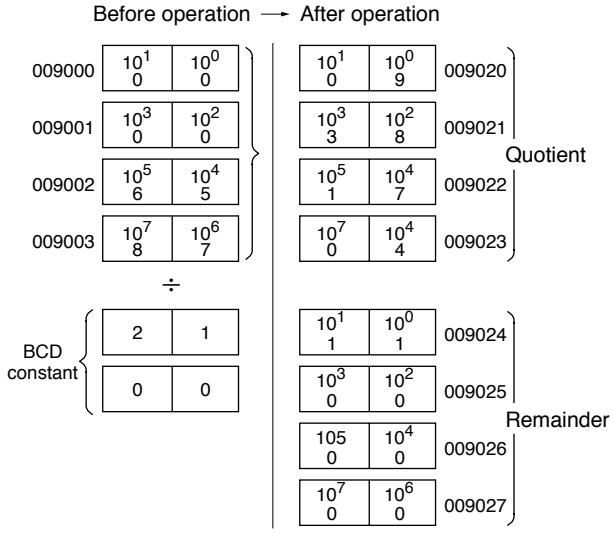
Contents after operation						
	S_1 to S_1+3	Unchanged				
	D to D+3	Quotient (8-digit BCD)	Unchanged when the contents of registers S_1 to S_1+3 are not BCD code or the contents of "n" is 0000. (Does not calculate.)			
	D+4 to D+7	Remainder (8-digit BCD)				
	Flag	S_1 to S_1+3, "n"	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		BCD code	0	0	0	0
		Not BCD code If "n" is 00			1	

[Explanation]

Instruction	
S T R	004001
F c16d	009000
	0021
	009020

004001 | Fc16d DIV | 009000 | 0021 | 009020

When the input condition 004001 changes from OFF to ON, the BCD 8 digits in registers 009000 and 009003 are divided by the BCD constant 0021 and the quotient is stored in the registers 009020 to 009023 with the remainder in the 009024 to 009027.



The above example shows the operation of 87650000 divided by 21 equals 4173809 with the remainder of 11.

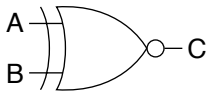
* Be sure to use even addresses for registers S_1 and D. (Odd address such as 019003 etc. are prohibited to use.)

- If denominator is not greater than numerator (S_1 to S_1+3<n), the quotient (contents of D to D+3) is 0 and remainder (contents of D+4 to D+7) is numerator (contents of S_1 to S_1+3). For instance, 20/30 will produce the result

F-17 XNR — Exclusive NORs register with register (1 byte) (eXclusive NoR)

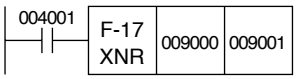
Symbol	F-17 XNR | S | D	
Function	The contents of the register S are XNRed with the contents of the register D and its result is stored in the register D.	
Operation	$\overline{S \oplus D} \rightarrow D$	
S	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Result
	Flag	Unchanged

- Exclusive NOR truth table

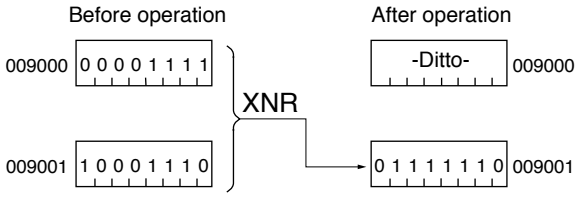
Symbol	A	B	C
	0	0	1
	1	0	0
	0	1	0
	1	1	1

[Explanation]

Instruction	
S T R	004001
F -17	
	009000
	009001



When the input condition 004001 changes from OFF to ON, the contents of the register 009000 are XNRed with the contents of the register 009001 and its result is stored in the register 009001. The contents of the register 009000 remains unchanged.



Bit matched in 009000 and 009001 (0 and 0, 1 and 1) is turned to 1 and unmatched bit (0 and 1) is turned to 0.

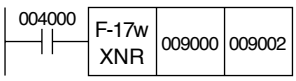
F-17w XNR — Exclusive NORs register with register (1 word) (eXclusive NoR)

Symbol	F-17w XNR | S | D	
Function	The 16-bit contents of the registers S, S+1 are XNRed with the 16-bit contents of the registers D, D+1 and its result is stored in the registers D, D+1.	
Operation	$\overline{S, S+1 \oplus D, D+1} \rightarrow D, D+1$	
S	Use range B *	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1	Unchanged
	D, D+1	Result
	Flag	Unchanged

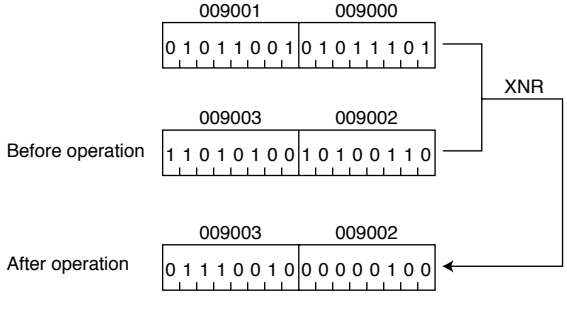
* Be sure to use even addresses for registers S and D. (Odd address such as 019003 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	004000
F -17w	
	009000
	009002



When the input condition 004000 changes from OFF to ON, the 16-bit contents of the registers 009000 and 009001 are XNRed with the 16-bit contents of the registers 009002 and 009003 and its result is stored in the registers 009002 and 009003. The contents of the registers 009000 and 009001 remain unchanged.



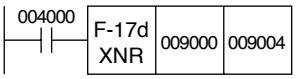
F-17d XNR — Exclusive NORs register with register (2 words) (eXclusive NoR)

Symbol	F-17d XNR | S | D	
Function	The contents of the registers S to S+3 (32 bits) are XNRed with the contents of the registers D to D+3 (32 bits) and its result is stored in the registers D to D+3.	
Operation	$\overline{\text{S to S+3} \oplus \text{D to D+3}} \rightarrow$ D to D+3	
S	Use range C *	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S to S+3	Unchanged
	D to D+3	Result
	Flag	Unchanged

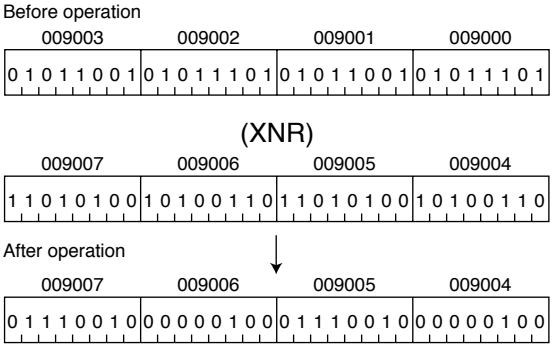
* Be sure to use even addresses for registers S and D. (Odd address such as 019003 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	004000
F -17d	
	009000
	009004



When the input condition 004000 changes from OFF to ON, the 32-bit contents of the registers 009000 to 009003 are XNRed with the 32-bit contents of the registers 009004 to 009007 and its result is stored in the registers 009004 to 009007. The contents of the registers 009000 to 009003 remain unchanged.

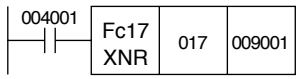


Fc17 XNR — Exclusive NORs register with octal constant (1 byte) (eXclusive NOR)

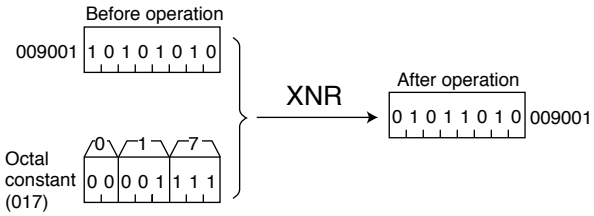
Symbol	Fc17 XNR | n | D	
Function	An octal constant "n" is XNRed with the contents of the register D and its result is stored in the register D.	
Operation	$\overline{n \oplus D} \rightarrow D$	
n	Use range 000 to 377	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D	Result
	Flag	Unchanged

[Explanation]

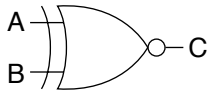
Instruction	
S T R	004001
F c17	
	017
	009001



When the input condition 004001 changes from OFF to ON, the octal constant 017 is XNRed with the contents of the register 009001 and its result is stored in the register 009001.



- Exclusive NOR truth table

Symbol	A	B	C
	0	0	1
	1	0	0
	0	1	0
	1	1	1

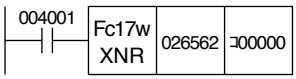
Fc17w XNR — Exclusive NORs register with octal constant (1 word) (eXclusive NoR)

Symbol	Fc17w XNR | n | D	
Function	An octal constant "n" is XNRed with the 16 bits contents of the registers D and D+1 and its result is stored in the registers D and D+1.	
Operation	$\overline{n \oplus D, D+1} \rightarrow D, D+1$	
n	Use range 000000 to 177777(8)	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D, D+1	Result
	Flag	Unchanged

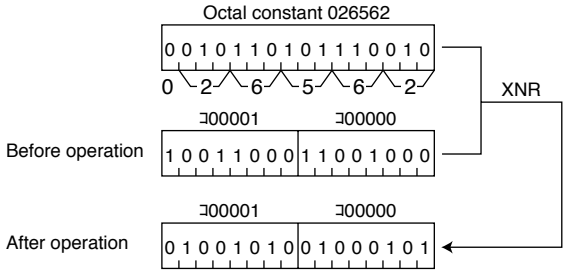
* Be sure to use even addresses for registers D.
(Odd address such as ⊐00011 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	004001
F c17w	
	026562
	⊐00000



When the input condition 004001 changes from OFF to ON, the octal constant 026562 is XNRed with the16-bit contents of the registers ⊐00000 and ⊐00001 and its result is stored in the registers ⊐00000 and ⊐00001.



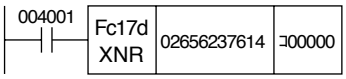
Fc17d XNR — Exclusive NORs register with octal constant (2 words) (eXclusive NoR)

Symbol	Fc17d XNR | n | D	
Function	An octal constant "n" is XNRed with the 32 bits contents of the registers D to D+3 and its result is stored in the registers D to D+3.	
Operation	$\overline{n \oplus D \text{ to } D+3} \rightarrow D \text{ to } D+3$	
n	Use range 00000000000 to 17777777777(8)	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D to D+3	Result
	Flag	Unchanged

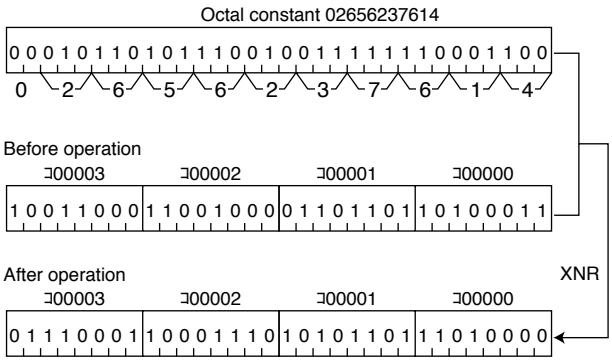
* Be sure to use even addresses for registers D.
(Odd address such as ⊐00011 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	004001
F c17d	
02656237614	
⊐00000	



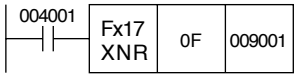
When the input condition 004001 changes from OFF to ON, the octal constant 02656237614 is XNRed with the 32-bit contents of the registers ⊐00000 to ⊐00003 and its result is stored in the registers ⊐00000 to ⊐00003.



Fx17 XNR — Exclusive NORs register with hexadecimal constant (1 byte) (eXclusive NoR)

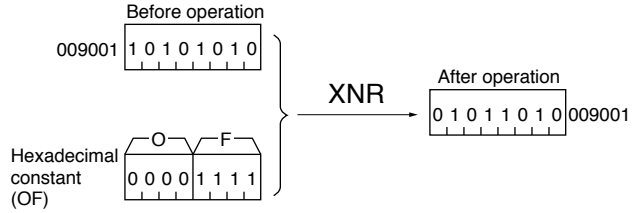
Symbol	Fx17 XNR | n | D	
Function	A hexadecimal constant "n" is XNRed with the contents of the register D and its result is stored in the register D.	
Operation	$\overline{n \oplus D} \rightarrow D$	
n	Use range 00 to FF	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D	Result
	Flag	Unchanged

[Explanation]

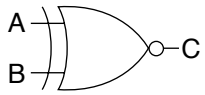


Instruction	
S T R	004001
F x17	
	0F
	009001

When the input condition 004001 changes from OFF to ON, the hexadecimal constant 0F is XNRed with the contents of the register 009001 and its result is stored in the register 009001.



- Exclusive NOR truth table

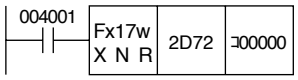
Symbol	A	B	C
	0	0	1
	1	0	0
	0	1	0
	1	1	1

Fx17w XNR — Exclusive NORs register with hexadecimal constant (1 word) (eXclusive NoR)

Symbol	Fx17w X N R | n | D	
Function	A hexadecimal constant "n" is XNRed with the 16-bit contents of the registers D and D+1 and its result is stored in the registers D and D+1.	
Operation	$\overline{n \oplus D, D+1} \rightarrow D, D+1$	
n	Use range 0000 to FFFF(H)	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D, D+1	Result
	Flag	Unchanged

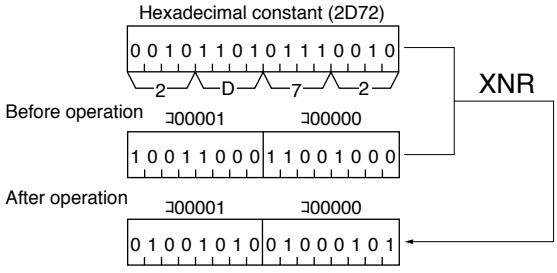
* Be sure to use even addresses for register D.
(Odd address such as ⊐00011 etc. are prohibited to use.)

[Explanation]



Instruction	
S T R	004001
F x17w	
	2D72
	⊐00000

When the input condition 004001 changes from OFF to ON, the hexadecimal constant 2D72 is XNRed with the 16-bit contents of the registers ⊐00000 and ⊐00001 and its result is stored in the registers ⊐00000 and ⊐00001.

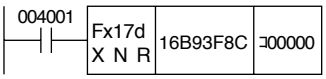


Fx17d XNR — Exclusive NORs register with hexadecimal constant (2 words) (eXclusive NoR)

Symbol	Fx17d X N R | n | D	
Function	A hexadecimal constant "n" is XNRed with the 32-bit contents of the registers D to D+3 and its result is stored in the registers D to D+3.	
Operation	$\overline{n \oplus D \text{ to } D+3} \rightarrow D \text{ to } D+3$	
n	Use range 00000000 to FFFFFFFF(H)	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D to D+3	Result
	Flag	Unchanged

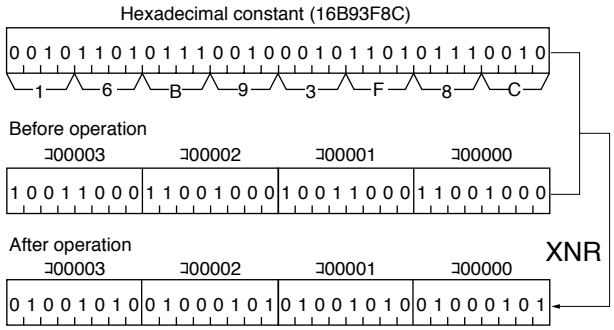
* Be sure to use even addresses for register D.
(Odd address such as ㅋ00011 etc. are prohibited to use.)

[Explanation]



Instruction	
S T R	004001
F x17d	
	16B93F8C
	ㅋ00000

When the input condition 004001 changes from OFF to ON, the hexadecimal constant 16B93F8C is XNRed with the 32-bit contents of the registers ㅋ00000 to ㅋ00003 and its result is stored in the registers ㅋ00000 to ㅋ00003.



F-18 XOR — Exclusive ORs register with register (1 byte) (eXclusive OR)

Symbol	F-18 XOR | S | D	
Function	The contents of the register S are XORed with the contents of the register D and its result is stored in the register D.	
Operation	S ⊕ D → D	
S	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Result
	Flag	Unchanged

- Exclusive OR truth table

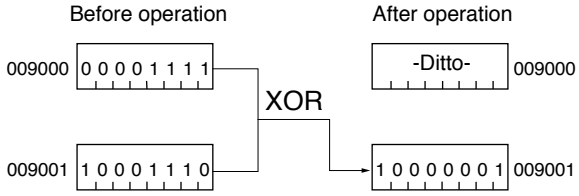
Symbol	A	B	C
A, B → C	0	0	0
	1	0	1
	0	1	1
	1	1	0

[Explanation]

Instruction	
S T R	004001
F -18	
	009000
	009001

004001 — F-18 XOR | 009000 | 009001

When the input condition 004001 changes from OFF to ON, the contents of the register 009000 are XORed with the contents of the register 009001 and its result is stored in the register 009001.
The contents of the register 009000 remain unchanged.



Bit matched in 009000 and 009001 (0 for 0, 1 for 1) is turned to 0 and unmatched bit (0 and 1) is turned to 1.

F-18w XOR — Exclusive ORs register with register (1 word) (eXclusive OR)

Symbol	F-18w XOR | S | D	
Function	The 16-bit contents of the registers S, S+1 are XORed with the 16-bit contents of the registers D, D+1 and its result is stored in the registers D, D+1.	
Operation	S, S+1 ⊕ D, D+1 → D, D+1	
S	Use range B *	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1	Unchanged
	D, D+1	Result
	Flag	Unchanged

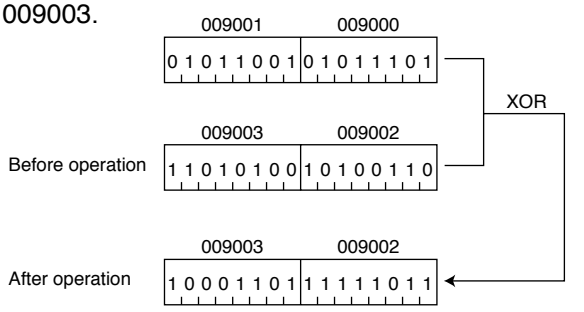
* Be sure to use even addresses for register S and D.
(Odd address such as 019003 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	004000
F -18w	
	009000
	009002

004000 — F-18w XOR | 009000 | 009002

When the input condition 004000 changes from OFF to ON, the 16-bit contents of the registers 009000 and 009001 are XORed with the 16 bits contents of the registers 009002 and 009003 and its result is stored in the registers 009002 and 009003.



The contents of the registers 009000 and 009001 remain unchanged.

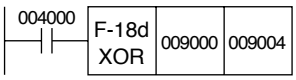
F-18d XOR — Exclusive OR register with register (2 words) (eXclusive OR)

Symbol	F-18d XOR | S | D	
Function	The contents of the registers S to S+3 (32 bits) are XORed with the contents of the registers D to D+3 (32 bits) and its result is stored in the registers D to D+3.	
Operation	S to S+3 ⊕ D to D+3 → D to D+3	
S	Use range C *	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S to S+3	Unchanged
	D to D+3	Result
	Flag	Unchanged

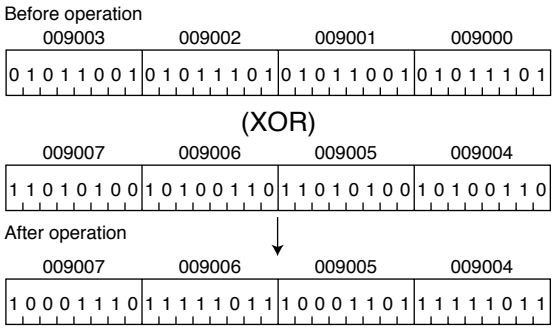
* Be sure to use even addresses for registers S and D. (Odd address such as 019003 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	004000
F -18d	
	009000
	009004



When the input condition 004000 changes from OFF to ON, the 32-bit contents of the registers 009000 to 009003 are XORed with the 32-bit contents of the registers 009004 to 009007 and its result is stored in the registers 009004 to 009007.



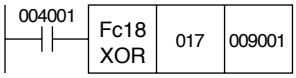
The contents of the registers 009000 to 009003 remain unchanged.

Fc18 XOR — Exclusive ORs register with octal constant (1 byte) (eXclusive OR)

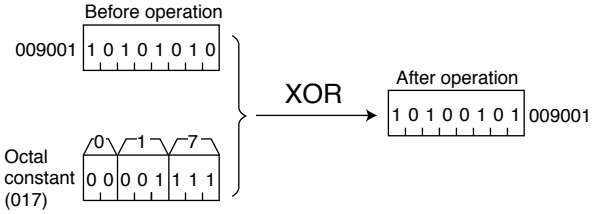
Symbol	Fc18 XOR | n | D	
Function	An octal constant "n" is XORed with the contents of the register D and its result is stored in the register D.	
Operation	n⊕D→D	
n	Use range 000 to 377	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D	Result
	Flag	Unchanged

[Explanation]

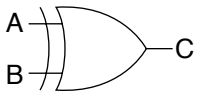
Instruction	
S T R	004001
F c18	
	017
	009001



When the input condition 004001 changes from OFF to ON, the octal constant 017 is XORed with the contents of the register 009001 and its result is stored in the register 009001.



- Exclusive OR truth table

Symbol	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	0

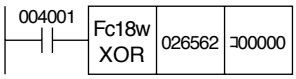
Fc18w XOR — Exclusive ORs register with octal constant (1 word) (eXclusive OR)

Symbol	Fc18w XOR | n | D	
Function	An octal constant "n" is XORed with the 16-bit contents of the registers D and D+1 and its result is stored in the registers D and D+1.	
Operation	n⊕D, D+1→D, D+1	
n	Use range 000000 to 177777(8)	
D	Use range 000000 B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D, D+1	Result
	Flag	Unchanged

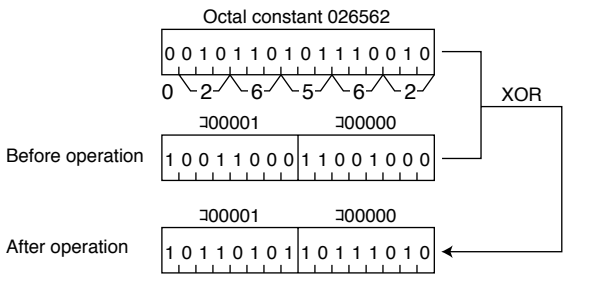
* Be sure to use even addresses for register D.
(Odd address such as ⊐00011 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	004001
F c18w	
	026562
	⊐00000



When the input condition 004001 changes from OFF to ON, the octal constant 026562 is XORed with the16-bit contents of the registers ⊐00000 and ⊐00001 and its result is stored in the registers ⊐00000 and ⊐00001.



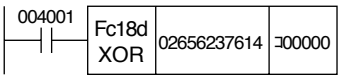
Fc18d XOR — Exclusive ORs register with octal constant (2 words) (eXclusive OR)

Symbol	Fc18d XOR | n | D	
Function	An octal constant "n" is XORed with the 32-bit contents of the registers D to D+3 and its result is stored in the registers D to D+3.	
Operation	n⊕D to D+3→D to D+3	
n	Use range 00000000000 to 37777777777(8)	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D to D+3	Result
	Flag	Unchanged

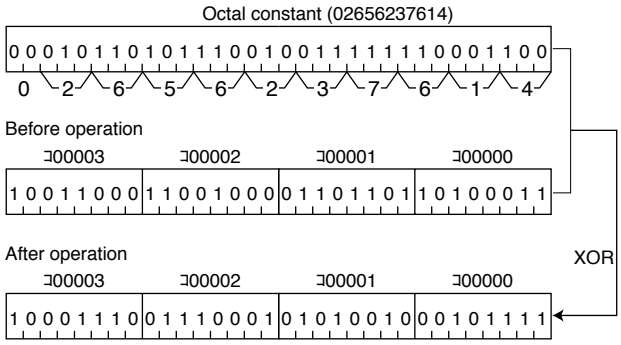
* Be sure to use even addresses for register D.
(Odd address such as ㅋ00011 etc. are prohibited to use.)

[Explanation]

Instruction
S T R 004001
F c18d
02656237614
ㅋ00000



When the input condition 004001 changes from OFF to ON, the octal constant 02656237614 is XORed with the 32-bit contents of the registers ㅋ00000 to ㅋ00003 and its result is stored in the registers ㅋ00000 to ㅋ00003.

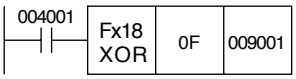


Fx18 XOR — Exclusive ORs register with hexadecimal constant (1 byte) (eXclusive OR)

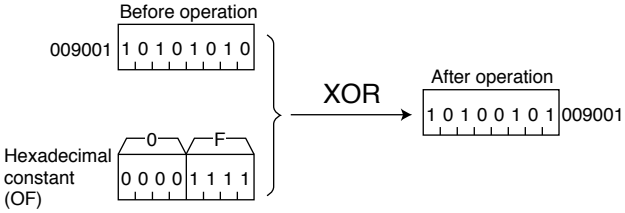
Symbol	Fx18 XOR | n | D	
Function	A hexadecimal constant "n" is XORed with the contents of the register D and its result is stored in the register D.	
Operation	n⊕D→D	
n	Use range 00 to FF	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D	Result
	Flag	Unchanged

[Explanation]

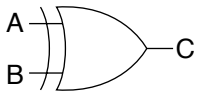
Instruction	
S T R	004001
F x18	
	0F
	009001



When the input condition 004001 changes from OFF to ON, the hexadecimal constant 0F is XORed with the contents of the register 009001 and its result is stored in the register 009001.



- Exclusive OR truth table

Symbol	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	0

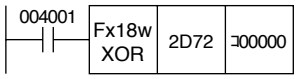
Fx18w XOR — Exclusive ORs register with hexadecimal constant (1 word) (eXclusive OR)

Symbol	Fx18w XOR | n | D	
Function	A hexadecimal constant "n" is XORed with the 16-bit contents of the registers D and D+1 and its result is stored in the registers D and D+1.	
Operation	n⊕D, D+1→D, D+1	
n	Use range 0000 to FFFF(H)	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D, D+1	Result
	Flag	Unchanged

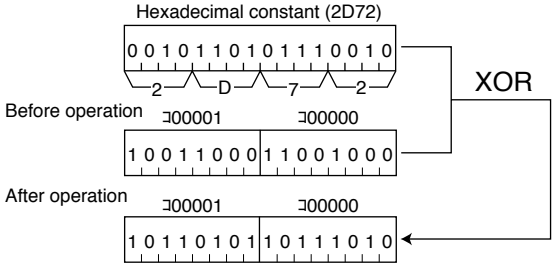
* Be sure to use even addresses for register D.
(Odd address such as ⊐00011 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	004001
F x18w	
	2D72
	⊐00000



When the input condition 004001 changes from OFF to ON, the hexadecimal constant 2D72 is XORed with the 16-bit contents of the registers ⊐00000 and ⊐00001 and its result is stored in the registers ⊐00000 and ⊐00001.

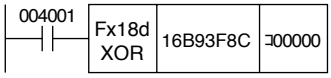


Fx18d XOR — Exclusive ORs register with hexadecimal constant (2 words) (eXclusive OR)

Symbol	Fx18d XOR | n | D	
Function	A hexadecimal constant "n" is XORed with the 32-bit contents of the registers D to D+3 and its result is stored in the registers D to D3.	
Operation	n⊕D to D+3→D to D+3	
n	Use range 00000000 to FFFFFFFF(H)	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D to D+3	Result
	Flag	Unchanged

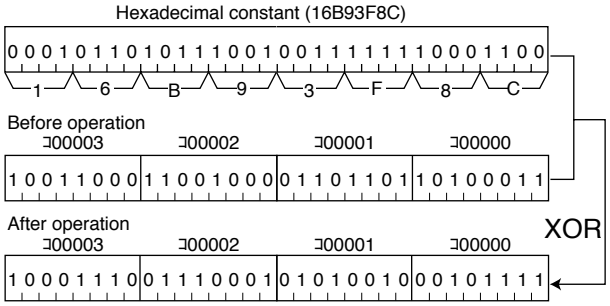
* Be sure to use even addresses for register D.
(Odd address such as ↄ00011 etc. are prohibited to use.)

[Explanation]



Instruction	
S T R	004001
F x18d	
	16B93F8C
	ↄ00000

When the input condition 004001 changes from OFF to ON, the hexadecimal constant 16B93F8C is XORed with the 32-bit contents of the registers ↄ00000 to ↄ00003 and its result is stored in the registers ↄ00000 to ↄ00003.



F-20
(MD)

Maintenance display

This instruction is identical to the F-20 instruction.
=> See MD "Maintenance display."

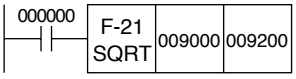
F-21 SQRT — Obtains square root of register (BCD 8 digits) (SQuare RooT)

Symbol	F-21 SQRT | S | D
Function	Determines the square root of the 8-digit BCD contents of registers S to S+3, and stores the result in register D. Truncates any decimal places.
Operation	$\sqrt{(S \text{ to } S+3)} \rightarrow D, D+1$
S	Use range C *
D	Use range B *
Condition	Rising edge of input signal (OFF to ON)
Contents after operation: S to S+3	Unchanged
Contents after operation: D, D+1	Unchanged when the contents of registers S to S+3 are not BCD code. (Does not calculate.)

Contents after operation — Flag

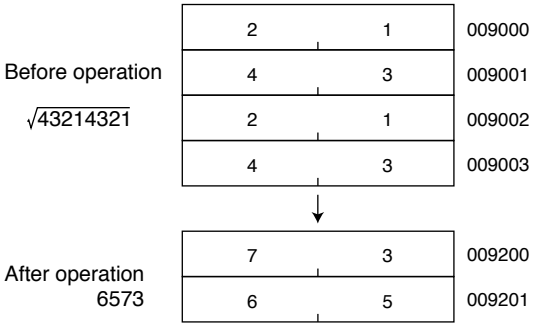
S to S+3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
BCD code	0	0	0	0
Not BCD code			1	

[Explanation]



Instruction	
S T R	000000
F -21	
	009000
	009200

When the input condition of 000000 changes from OFF to ON, this instruction obtains the square root of the 8-digit BCD contents of registers 009000 to 009003 and stores the result in registers 009200 and 009201.



All decimal places in the operation result are truncated.

* Be sure to use even addresses for registers S and D. (Odd address such as 019003 etc. are prohibited to use.)

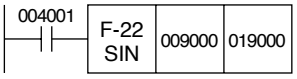
F-22 SIN — Executes trigonometric function (SIN)

Symbol	F-22 SIN | S | D
Function	Determines the SIN of the 6-digit BCD contents of registers S to S+2 and its result is stored in the registers D to D+3 (8-digit BCD).
Operation	SIN (S to S+2) → D to D+3
S	Use range E
D	Use range C
Condition	Rising edge of input signal (OFF to ON)
Contents before operation: S	Decimal fraction of an angle. (2-digit BCD) — Area of an angle is 0 to 9999.99°
Contents before operation: S+1, S+2	Integer of an angle. (4-digit BCD)
Contents after operation: S to S+2	Unchanged
Contents after operation: D, D+1	Decimal fraction of result. (4-digit BCD) *
Contents after operation: D+2	Integer of result. (2-digit BCD) *
Contents after operation: D+3	A sign of result. (2-digit BCD) 00(H): Positive (+) 80(H): Negative (−) *

Contents after operation — Flag

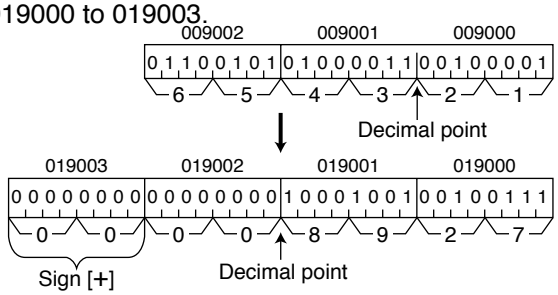
Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
Positive (+)	0	0	0	1
Negative (−)	0	1	0	0
S to S+2 are not BCD code.	0	0	1	0

[Explanation]



Instruction	
S T R	004001
F -22	
	009000
	019000

When the input condition of 004001 changes from OFF to ON, this instruction obtains the SIN of the 6-digit BCD contents of registers 009000 to 009002, and stores the result in registers 019000 to 019003.



The above example shows the operation of SIN 6543.21°≒0.8927.

* D to D+3 (Contents after calculation.)
- The calculation result will be rounded off below the 5th digit to the right of the decimal, area of result is −1.0000 to 1.0000.
- Unchanged when the contents of registers S to S+2 are not BCD code.

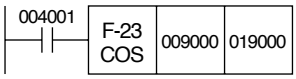
F-23 COS — Executes trigonometric function (COS)

Symbol	F-23 COS | S | D		
Function	Determines the COS of the 6-digit BCD contents of registers S to S+2 and its result is stored in the registers D to D+3 (8-digit BCD).		
Operation	COS (S to S+2) → D to D+3		
S	Use range E		
D	Use range C		
Condition	Rising edge of input signal (OFF to ON)		
Contents before operation	S	Decimal fraction of an angle. (2-digit BCD)	Area of an angle is 0 to 9999.99°
	S+1, S+2	Integer of an angle. (4-digit BCD)	
Contents after operation	S to S+2	Unchanged	
	D, D+1	Decimal fraction of result. (4-digit BCD)	*
	D+2	Integer of result. (2-digit BCD)	
	D+3	A sign of result. (2-digit BCD) 00(H): Positive (+) 80(H): Negative (−)	

Flag

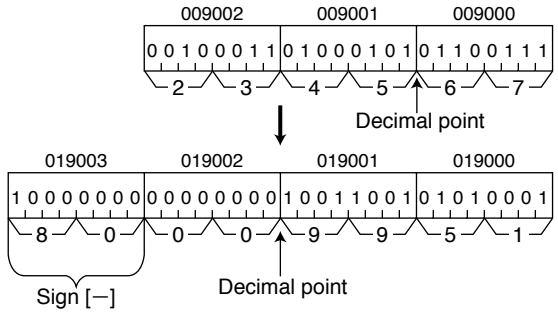
Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
Positive (+)	0	0	0	1
Negative (−)	0	1	0	0
S to S+2 are not BCD code.	0	0	1	0

[Explanation]



Instruction	
S T R	004001
F -23	
	009000
	019000

When the input condition of 004001 changes from OFF to ON, this instruction obtains the COS of the 6-digit BCD contents of registers 009000 to 009002, and stores the result in registers 019000 to 019003.



The above example shows the operation of COS 2345.67°≒−0.9951.

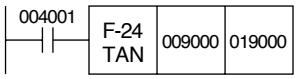
* D to D+3 (Contents after calculation.)
- The calculation result will be rounded off below the 5th digit to the right of the decimal, area of result is −1.0000 to 1.0000.
- Unchanged when the contents of registers S to S+2 are not BCD.

F-24 TAN — Executes trigonometric function (TAN)

Symbol	F-24 TAN | S | D		
Function	Determines the TAN of the 6 digits BCD contents of registers S to S+2 and its result is stored in the registers D to D+3 (8-digit BCD).		
Operation	TAN (S to S+2) → D to D+3		
S	Use range E		
D	Use range C		
Condition	Rising edge of input signal (OFF to ON)		
Contents before operation	S	Decimal fraction of an angle. (2-digit BCD)	Area of an angle is 0 to 9999.99°.
	S+1, S+2	Integer of an angle. (4-digit BCD)	
Contents after operation	S to S+2	Unchanged	
	D, D+1	Decimal fraction of result. (4-digit BCD)	*
	D+2	Integer of result. (2-digit BCD)	
	D+3	A sign of result. (2-digit BCD) 00(H): Positive (+) 80(H): Negative (−)	

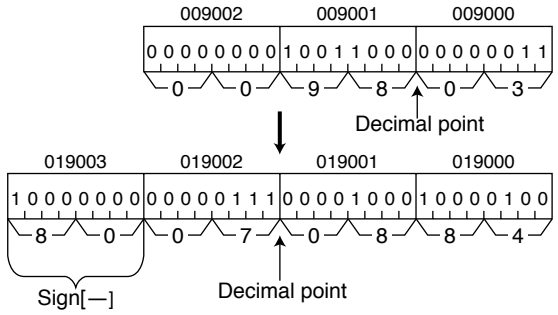
Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	Positive (+)	0	0	0	1
	Negative (−)	0	1	0	0
	More than +100 or less than −100.	0	0	1	0
	S to S+2 are not BCD code.				

[Explanation]



Instruction	
S T R	004001
F -24	
	009000
	019000

When the input condition of 004001 changes from OFF to ON, this instruction obtains the TAN of the 6-digit BCD contents of registers 009000 to 009002, and stores the result in registers 019000 to 019003.



- The above example shows the operation of TAN 98.03°≒−7.0884.

* D to D+3 (Contents after calculation.)

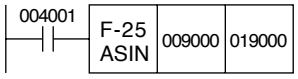
- The calculation result will be rounded off below the 5th digit to the right of the decimal, area of result is −99.9999 to 99.9999.
- Unchanged when the contents of registers S to S+2 are not BCD.
- When operation of result is contents of S becomes more than 100.0000 or less than −100.0000, the PC turns ON an error flag, and does not operate.

F-25 ASIN — Executes trigonometric function (SIN^{-1})

Symbol	F-25 ASIN | S | D		
Function	Determines the SIN^{-1} of the 8-digit BCD contents of registers S to S+3 and its result is stored in the registers D to D+3 (8-digit BCD).		
Operation	SIN^{-1} (S to S+3) → D to D+3		
S	Use range C		
D	Use range C		
Condition	Rising edge of input signal (OFF to ON)		
Contents before operation	S, S+1	Decimal fraction of SIN^{-1} data. (4-digit BCD)	Area of SIN^{-1} data is −1.0000 to 1.0000.
	S+2	Integer of SIN^{-1} data. (2-digit BCD)	
	S+3	A sign of SIN^{-1} data. (2-digit BCD) 00(H): Positive (+) 80(H): Negative (−)	
Contents after operation	S to S+3	Unchanged	
	D	Decimal fraction of result. (2-digit BCD)	*
	D+1, D+2	Integer of result. (4-digit BCD)	
	D+3	A sign of result. (2-digit BCD) 00(H): Positive (+) 80(H): Negative (−)	

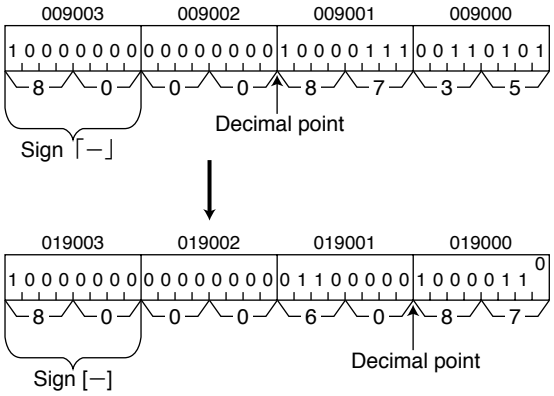
Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	Positive (+)	0	0	0	1
	Negative (−)	0	1	0	0
	S to S+3 are not BCD code.	0	0	1	0
	S to S+3 is bigger than +1 or smaller than −1.				

[Explanation]



Instruction	
S T R	004001
F -25	
	009000
	019000

When the input condition of 004001 changes from OFF to ON, this instruction obtains the SIN^{-1} of the 8-digit BCD contents of registers 009000 to 009003, and stores the result in registers 019000 to 019003.



The above example shows the operation of SIN^{-1} (−0.8735)≒−60.87°.

* D to D+3 (Contents after calculation.)
- The calculation result will be rounded off below the 5th digit to the right of the decimal, area of result is −90.00° to 90.00°.
- Unchanged when the contents of registers S to S+3 are not BCD.

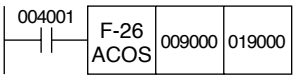
F-26 ACOS — Executies trigonometric function (COS^{-1})

Symbol	F-26 ACOS | S | D		
Function	Determines the COS^{-1} of the 8 digits BCD contents of registers S to S+3 and its result is stored in the registers D to D+3 (8-digit BCD).		
Operation	COS^{-1} (S to S+3) → D to D+3		
S	Use range C		
D	Use range C		
Condition	Rising edge of input signal (OFF to ON)		
Contents before operation	S, S+1	Decimal fraction of COS^{-1} data. (4-digit BCD)	Area of COS^{-1} data is −1.0000 to 1.0000.
	S+2	Integer of COS^{-1} data. (2-digit BCD)	
	S+3	A sign of COS^{-1} data. (00(H): Positive (+), 80(H): Negative (−)) (2-digit BCD)	
Contents after operation	S to S+3	Unchanged	
	D	Decimal fraction of result. (2-digit BCD)	*
	D+1, D+2	Integer of result. (4-digit BCD)	
	D+3	A sign of result. (00(H): Positive (+), 80(H): Negative (−)) (2-digit BCD)	

Flag

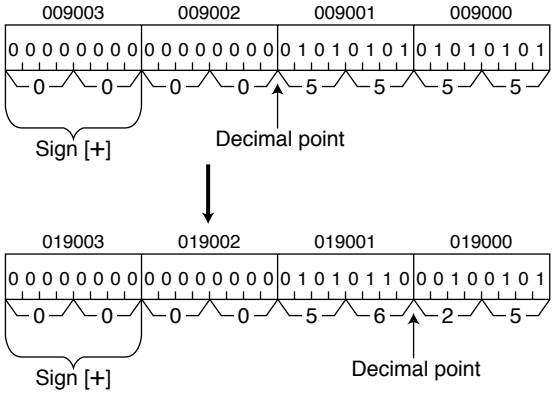
Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
Positive (+)	0	0	0	1
Negative (−)	0	1	0	0
S to S+3 are not BCD code.	0	0	1	0
S to S+3 is bigger than +1 or smaller than −1.				

[Explanation]



Instruction	
S T R	004001
F -26	
	009000
	019000

When the input condition of 004001 changes from OFF to ON, this instruction obtains the COS^{-1} of the 8-digit BCD contents of registers 009000 to 09003, and stores the result in registers 019000 to 019003.



The above example shows the operation of COS^{-1} (0.5555)≒56.25°.

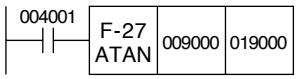
* D to D+3 (Contents after calculation.)
- Area of result is 0 to 180.00°
- Unchanged when the contents of registers S to S+3 are not BCD.

F-27 ATAN — Executes trigonometric function (TAN^{-1})

Symbol	F-27 ATAN | S | D		
Function	Determines the TAN^{-1} of the 8-digit BCD contents of registers S to S+3 and its result is stored in the registers D to D+3 (8-digit BCD).		
Operation	TAN^{-1} (S to S+3) → D to D+3		
S	Use range C		
D	Use range C		
Condition	Rising edge of input signal (OFF to ON)		
Contents before operation	S, S+1	Decimal fraction of TAN^{-1} data. (4-digit BCD)	Area of TAN^{-1} data is −98.9999 to 98.9999
	S+2	Integer of TAN^{-1} data. (2-digit BCD)	
	S+3	A sign of TAN^{-1} data. 00(H): Positive (+) 80(H): Negative (−) (2-digit BCD)	
Contents after operation	S to S+3	Unchanged	
	D	Decimal fraction of result. (2-digit BCD)	*
	D+1, D+2	Integer of result. (4-digit BCD)	
	D+3	A sign of result. (2-digit BCD) 00(H): Positive (+) 80(H): Negative (−)	

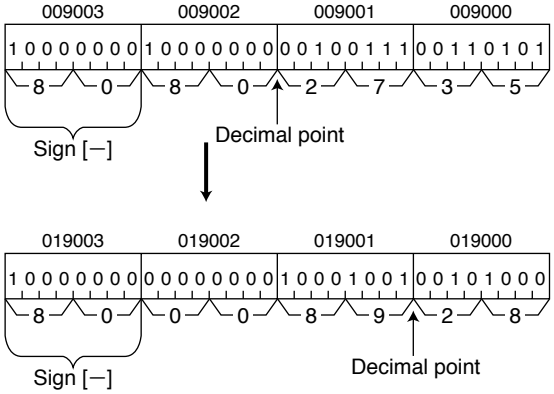
Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	Positive (+)	0	0	0	1
	Negative (−)	0	1	0	0
	S to S+3 are not BCD code.	0	0	1	0
	S to S+3 is bigger than +99.0000 or smaller than −99.0000.				

[Explanation]



Instruction	
S T R	004001
F -27	
	009000
	019000

When the input condition of 004001 changes from OFF to ON, this instruction obtains the TAN^{-1} of the 8-digit BCD contents of registers 009000 to 009003, and stores the result in registers 019000 to 019003.



The above example shows the operation of TAN^{-1} (−80.2735)≒−89.29°.

* D to D+3 (Contents after calculation.)
- Area of result is −89.42 to 89.42, the calculation result will be rounded off below the 3rd digit to the right of the decimal.
- Unchanged when the contents of registers S to S+3 are not BCD.

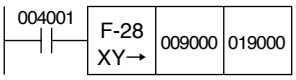
F-28
XY→

Exchanges the rectangular coordinate system (X, Y) data with polar coordinate system (γ , θ)

Symbol	F-28 XY→ | S | D		
Function	Exchanges the rectangular coordinate system (X, Y) of registers S to S+3 and S+4 to S+7 to polar coordinate system (γ , θ) and its result is in the registers D to D+3 and the registers D+4 to D+7.		
Operation	X (S to S+3), Y (S+4 to S+7) → γ (D to D+3), θ (D+4 to D+7)		
S	Use range G		
D	Use range G		
Condition	Rising edge of input signal (OFF to ON)		
Contents before operation	S to S+2	Integer of X data. (5-digit BCD)	- Area of X data is −99999 to 99999
	S+3	A sign of X data. (2-digit BCD) 00(H): Positive (+) 80(H): Negative (−)	
	S+4 to S+6	Integer of Y data. (5-digit BCD)	- Area of Y data is −99999 to 99999
	S+7	A sign of Y data. 00(H): Positive (+) 80(H): Negative (−)	
Contents after operation	S to S+7	Unchanged	
	D to D+3	Integer of γ data. (8-digit BCD)	Unchanged when X, Y data isn't BCD code
	D+4	Decimal fraction of θ data. (2-digit BCD)	- Area of θ data is −179.99° to 180.00° - Unchanged when X, Y data isn't BCD code.
	D+5, D+6	Integer of θ data. (3-digit BCD)	
	D+7	A sign of θ data. (2-digit BCD) 00(H): Positive (+) 80(H): Negative (−)	

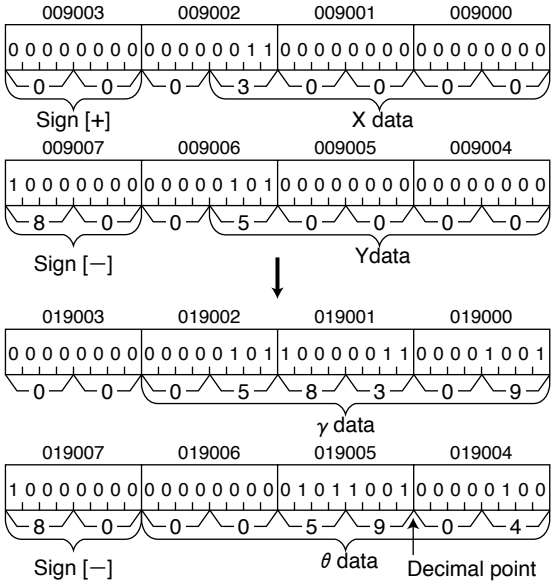
Flag	Contents of register S	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	When X, Y data is BCD code.	0	0	0	0
	When X, Y data is not BCD code.	0	0	1	0

[Explanation]

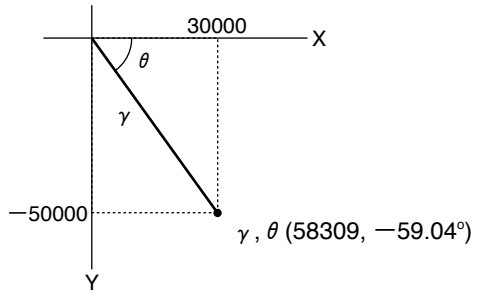


Instruction	
S T R	004001
F -28	
	009000
	019000

When the input condition of 004001 changes from OFF to ON, the PC exchange the rectangular coordinate system (X, Y) of registers 009000 to 009007 to polar coordinate system (γ , θ), and stores the result in registers 019000 to 019007.



- The above example shows the operation of X (30000), Y (−50000) → γ (58309), θ (−59.04°).



F-29 →XY — Exchanges the polar coordinate system (γ , θ) data with rectangular coordinate system (X, Y)

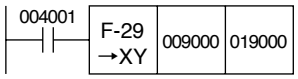
Symbol	F-29 →XY | S | D		
Function	Exchanges the rectangular coordinate system (X, Y) of registers S to S+3 and S+4 to S+7 to polar coordinate system (γ , θ) and its result is in the registers D to D+3 and the registers D+4 to D+7.		
Operation	γ (S to S+3), θ (S+4 to S+7) →X (D to D+3), Y (D+4 to D+7)		
S	Use range G		
D	Use range G		
Condition	Rising edge of input signal (OFF to ON)		
Contents before operation	S to S+3	Integer of γ data. (8-digit BCD)	- Area of γ data is 0 to 999999
	S+4	Decimal fraction of θ data. (2-digit BCD) 00(H): Positive (+) 80(H): Negative (−)	
	S+5, S+6	Integer of θ data. (3-digit BCD)	- Area of θ data is −179.99 to 180.00
	S+7	A sign of θ data. (2-digit BCD) 00(H): Positive (+) 80(H): Negative (−)	
Contents after operation	S to S+7	Unchanged	
	D to D+2	Integer of X data. (5-digit BCD)	- Unchanged when γ , θ data isn't BCD code
	D+3	A sign of X data. (2-digit BCD) 00(H): Positive (+) 80(H): Negative (−)	
	D+4 to D+6	Integer of Y data. (5-digit BCD)	
	D+7	A sign of Y data. (2-digit BCD) 00(H): Positive (+) 80(H): Negative (−)	

Flag	Contents of register S	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	When γ , θ data is BCD	0	0	0	0
	When γ , θ data is not BCD	0	0	1	0

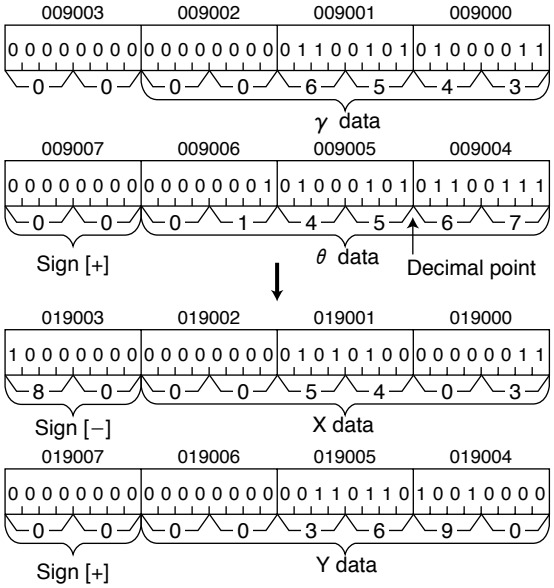
- When θ data of S+4 to S+7 is less than −180.00 or more than 180.01, it doesn't operate.

[Explanation]

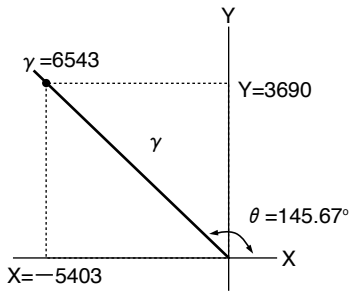
Instruction	
STR	004001
F-29	
	009000
	019000



When the input condition of 004001 changes from OFF to ON, the PC exchanges the polar coordinate system (γ , θ) of registers 009000 to 009007 to rectangular coordinate system (X, Y), and stores the result in registers 019000 to 019007.



- The above example shows the operation of γ (6543), θ (145.67°) →X (−5403), Y (3690)

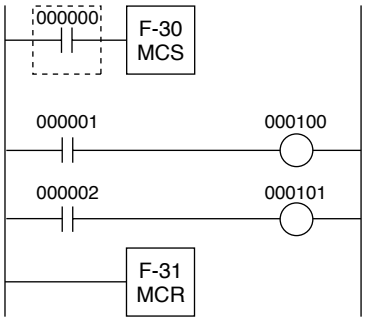


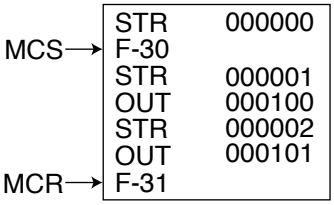
F-30 MCS Set master control (Master Control Set)

F-31 MCR Reset master control (Master Control Reset)

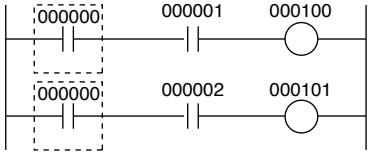
MCS and MCR can be used when the circuit is branched to a multiple number of outputs after common operational condition.

- In the case of MCS and MCR in use

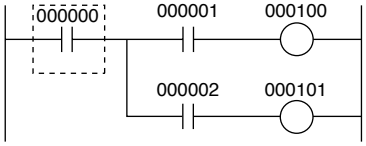




- In the case of MCS and MCR not in use



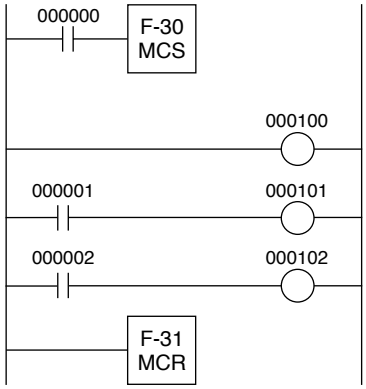
- In the case of a relay board

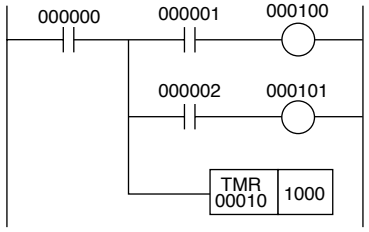


- When the F-30 (MCS) instruction is programmed, the contents of the accumulator are stored in the CPU internal register, and succeeding instructions are ANDed with the contents of the CPU internal register until the F-31 (MCR) instruction is met.
- The F-31 (MCR) instruction indicates the end of AND operation.
- It will help to simplify the program when the operational condition indicated in a block is complicated or many branches are set after the common operational condition.

- Do not directly connect the F-30 (MCS) derived bus line with OUT, TMR, and CNT instructions or application instruction.

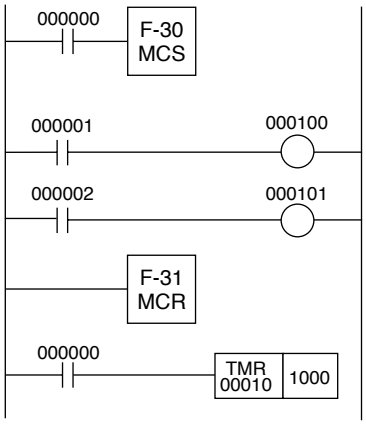
- MCS,MCR prohibited program



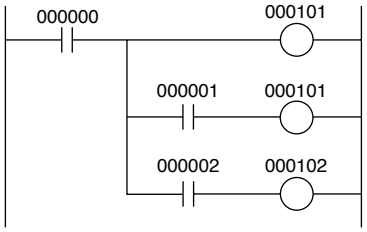


Instruction	
STR	000000
F-30	
OUT	000100
STR	000001
OUT	000101
STR	000002
OUT	000102
F-31	

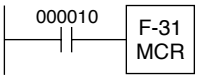
Must be programmed in the following manner.



(Reference) In the case of the relay board



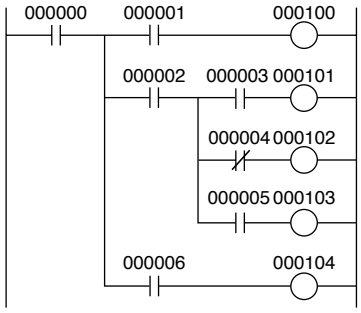
- The F-31 (MCR) instruction is an unconditional instruction.



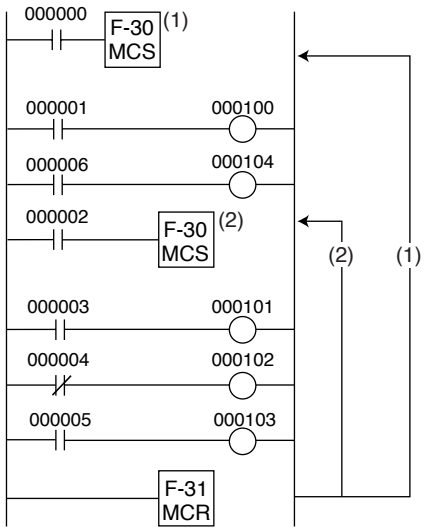
It prohibits the program like shown above.

- Another MCS may be used in between MCS and MCR.

- In the case of the relay board



- In the case of MCS and MCR in use



- The relay board ladder chart shown in left can be programmed in the following manner using MCS and MCR. However, there may be a need of changing the program sequence in the example (*).

Instruction	
S T R	000000
F -30	
S T R	000001
O U T	000100
S T R	000006 }
O U T	000104 } *
S T R	000002
F-30	
S T R	000003
O U T	000101
S T R N O T	000004
O U T	000102
S T R	000005
O U T	000103
F -31	

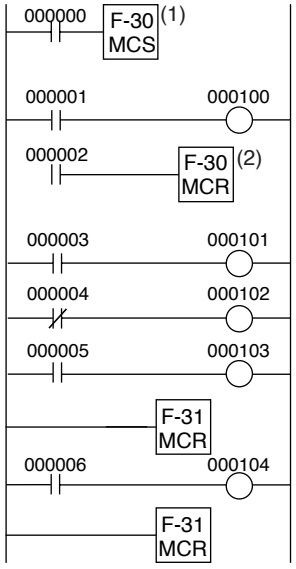
- The F-31 (MCR) instruction indicates the termination of the preceding F-30 (MCS) instruction ; (1) and (2) in the example.

- The desired circuit would not be established if programmed in the following way.

	Instruction	
M C S →	S T R	000000
	F -30	
	S T R	000001
	O U T	000100
	S T R	000002
M C S →	F-30	
	S T R	000003
	O U T	000101
	S T R N O T	000004
	O U T	000102
	S T R	000005
	O U T	000103
M C R →	F -31	
	S T R	000006 }
	O U T	000104 } *
M C R →	F -31	

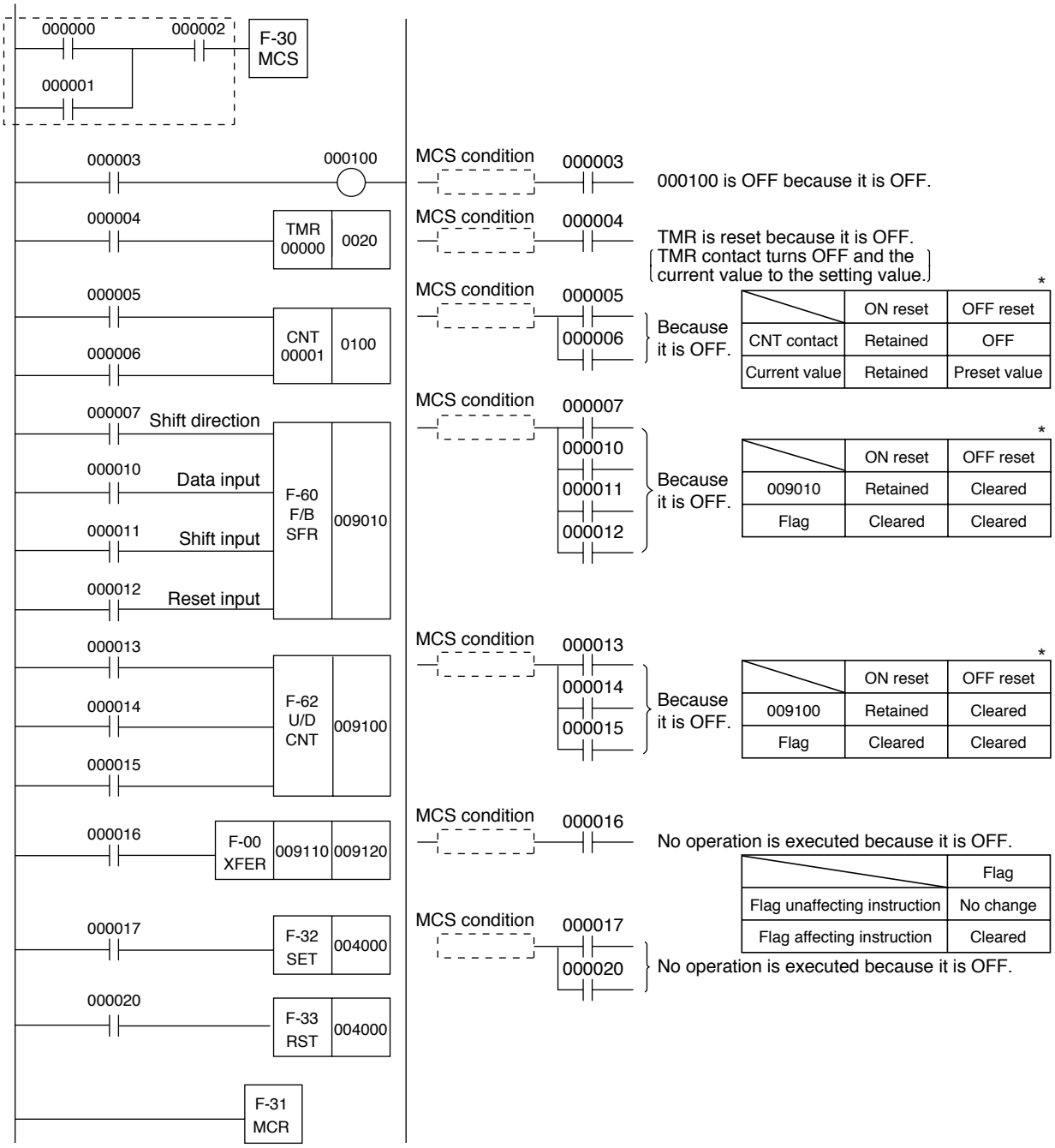
This MCR is insignificant.

MCS of (1) and (2) has been terminated with this MCR.



- Although it is possible to insert MCS as many times as required between MCS and MCR (*), the range of MCS terminates with MCR marked with an asterisk (*).
When nesting like above shows, we use F-231 (MCRN) "Reset master control nesting" instruction instead of F-31 (MCR).

- When the MCS condition (enclosed with dotted line) is OFF instructions existing between MCS and MCR will be handled in the following manner.

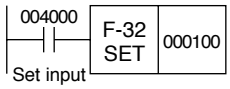


* With a CNT, F-60, or F-62 instruction, it is possible to ON reset or OFF reset the reset condition using the system memory #0202. In the case of the OFF reset, reset is done by MCS.

F-32 SET Set coil

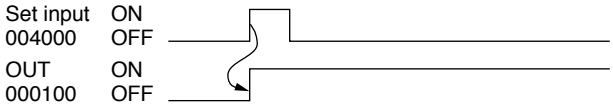
Symbol	F-32 SET | OUT	
Function	Activates the OUT specified by F-32 when the set input is turned ON.	
Operation	Activate the OUT specified by F-32.	
OUT	Use range K	
Condition	When set input is ON (not limited to OFF to ON change)	
Contents after operation	OUT	ON
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004000
F -32	
	000100

When the set input of 004000 is ON, this instruction sets ON OUT000100. Once sets ON, OUT000100 remains active after the set input is set to OFF. When the set input of 004000 is OFF, this instruction does not affect the state of OUT000100.

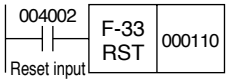


- When the F-32 (SET) instruction is contained in the F-30 (MCS) instruction, the OUT, once activated, remains active after the F-30 instruction has been executed.
- The F-32 (SET) instruction can control a specific OUT on more than 1 circuit.
- If the OUT specified by the F-32 (SET) instruction is within the latched relay area, it is kept the last state when the power is recovered from power failure. If the specified OUT is outside the latched relay area, it is reset at power recovery.
- If the OUT specified by the F-32 (SET) instruction is within the output hold area, it is kept the last state after the JW300's operation halts. If the specified OUT is outside the output hold area, it is reset when the JW300's operation halts. => See page "System memory #0232, #0233, #0252, and #0253."
- It is recommended that the F-32 (SET) instruction be used in conjunction with the F-33 instruction described on the next page.
- The F-32 (SET) and F-33 (RST) instructions placed between the F-30 (MCS) and F-31 (MCR) instructions are not operative if the operational condition of the F-30 (MCS) is OFF.

F-33 RST — Resets coil

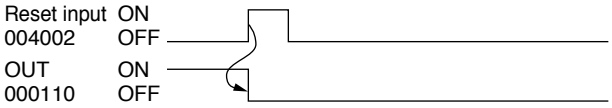
Symbol	F-33 RST | OUT	
Function	Deactivates the OUT specified by F-33 instruction when the reset input is turned ON.	
Operation	Deactivates the OUT specified by F-33.	
OUT	Use range K	
Condition	When reset input is ON (not limited to OFF to ON change)	
Contents after operation	OUT	OFF
	Flag	Unchanged

[Explanation]



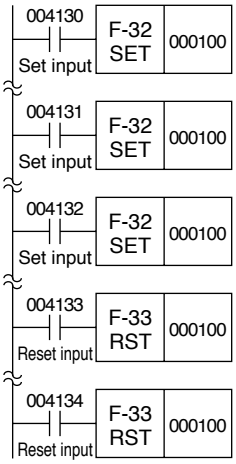
Instruction	
S T R	004002
F -33	
	000110

When the reset input of 004002 is ON, this instruction sets OFF OUT000110. Once sets OFF, OUT000110 remains inactive after the reset input is set to OFF. When the reset input of 004002 is OFF, this instruction does not affect the state of OUT000110.

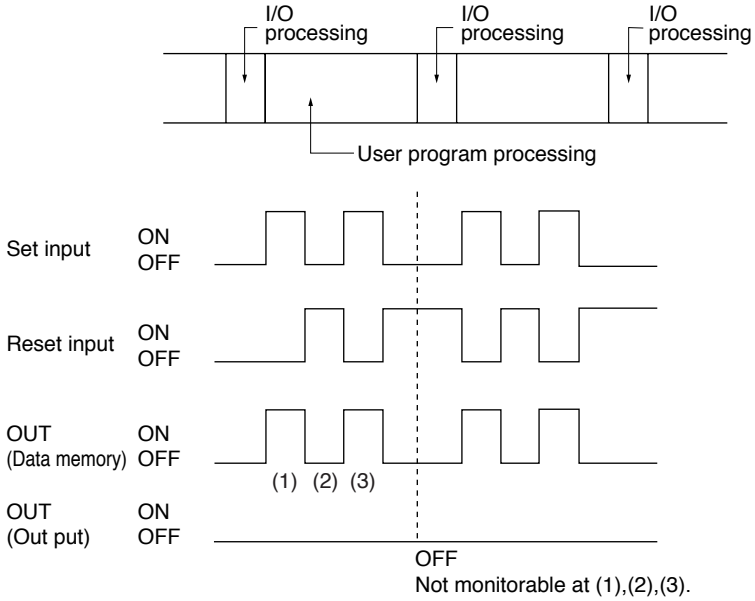


- It is the OUT specified by the F-33 (RST) instruction is within the output hold area, it is kept the last state after JW300 operation halts.
- If the specified OUT is outside the output hold area, it is reset when JW300 operation halts.
=> See page "System memory #0232, #0233, #0252, and #0253."

- The F-32 (SET) and F-33 (RST) instructions allow 1 OUT to be controlled under more than 1 condition.



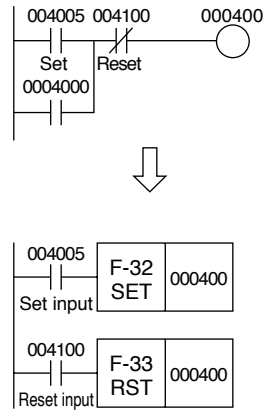
- If the set and reset inputs are activated or deactivated more than 1 time in 1 scan cycle, the data memory used as the OUT is set and reset repeatedly in 1 scan cycle. In case of an output terminal of an output module, however, the PC outputs the OUT results before the I/O processing. (ON or OFF state).



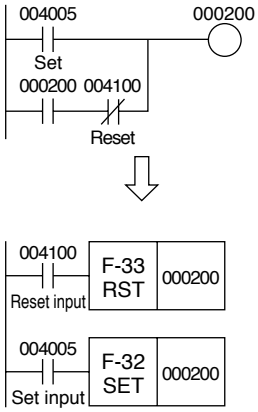
Even if the data memory turns ON/OFF several times during processing a user program, the JW300 can monitor a result just before the I/O processing.

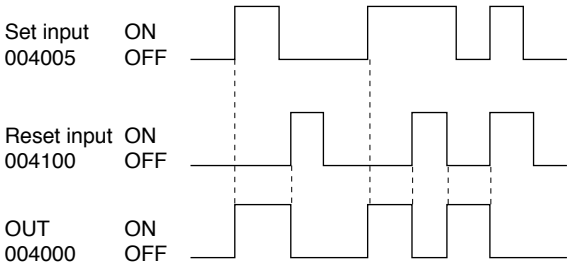
- It is recommended that the F-32 (SET) and F-33 (RST) instruction be used in pair.
  It will simplify the self holding circuit.

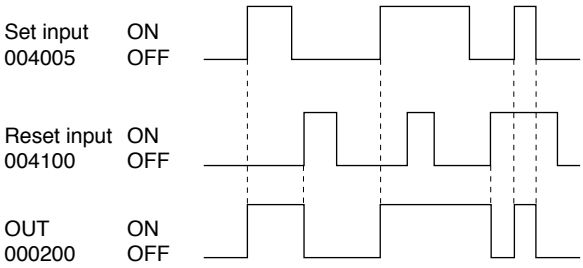
Reset priority self holding circuit



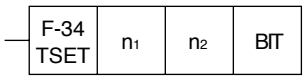
Set priority self holding circuit



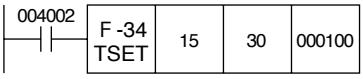




F-34 TSET — Compares with current value of clock (specified relay set)

Symbol		
Function	Compares the constants n_1 (hours) and n_2 (minutes) with the current value of the clock and sets (turns ON) the specified BIT (relay) if they match.	
Operation	Compares the current value of the clock with n_1 and n_2, and turns ON the relay if the comparison result shows a match.	
n_1	Use range 00 to 23 (decimal)	
n_2	Use range 00 to 59 (decimal)	
BIT	Use range K	
Condition	When the input signal is ON (not limited to an OFF to ON change)	
Contents after operation	n_1	Unchanged
	n_2	Unchanged
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004002
F-34	
	15
	30
	000100

When input condition 004002 is ON and the specified time and the current value of the clock match, relay 000100 turns ON.
Relay 000100 which is turned ON remains ON even if input condition 004002 turns OFF.
If the current value of the clock and the specified time do not match, the state of relay 000100 does not change.

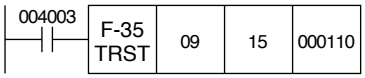
- Specified time ≧ Current value of clock
  → NOP
- Specified time = Current value of clock
  → Specified relay ON

- If the F-34 (TSET) instruction is within the F-30 (MCS) instruction, the relay which is turned ON remains ON even though the F-30 instruction turns OFF.
- Use of the F-34 (TSET) instruction can control 1 relay on multiple circuits.
- If the relay specified by the F-34 (TSET) instruction is within the latch specified area, it retains its state that existed before the power failure even after power is restored. If the specified Relay is outside the latch specified area, it is reset when power is recovered.
- If the relay specified by the F-34 (TSET) instruction is within the area which retains the output while the JW300 is stopped, it retains its state that existed before the stop when the JW300 is stopped. If the specified relay is outside the area which retains the output while the JW300 is stopped, it is reset when the JW300 is stopped.
- Use the F-34 (TSET) instruction as a pair with the F-35 (TRST) instruction on the next page.
- F-34 (TSET) and F-35 (TRST) instructions between the F-30 (MCS) and F-31 (MCR) instructions do not function when the operational condition for the F-30 (MCS) instruction is OFF.

F-35 TRST — Compares with current value of clock (specified relay reset)

Symbol	F-35 TRST | n_1 | n_2 | BIT	
Function	Compares the constants n_1 (hours) and n_2 (minutes) with the current value of the clock and resets (turns OFF) the specified BIT (Relay) if they match.	
Operation	Compares the current value of the clock with n_1 and n_2, and resets the relay if the comparison result shows a match.	
n_1	Use range 00 to 23 (decimal)	
n_2	Use range 00 to 59 (decimal)	
BIT	Use range K	
Condition	When the input signal is ON (not limited to an OFF to ON change)	
Contents after operation	n_1	Unchanged
	n_2	Unchanged
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004003
F -35	
	09
	15
	000110

When input condition 004003 is ON and the specified time and the current value of the clock match, relay 000110 turns OFF.
The relay which is turned OFF remains OFF even if input condition 004003 turns OFF.
If the current value of the clock and the specified time do not match, the state of relay 000110 does not change.

- Specified time $\gtrless$ Current value of clock
  → NOP
- Specified time = Current value of clock
  → Specified Relay OFF

- If the relay specified by the F-35 (TRST) instruction is within the latch specified area, it retains its state that existed before the power failure even after power is restored. If the specified relay is outside the latch specified area, it is reset when power is recovered.
- If the relay specified by the F-35 (TRST) instruction is within the area which retains the output while the JW300 is stopped, it retains its state that existed before the stop when the JW300 is stopped. If the specified relay is outside the area which retains the output while the JW300 is stopped, it is reset when the JW300 is stopped.
- Use the F-35 (TRST) instruction as a pair with the F-34 (TSET) instruction.

F-36 TADD — Adds time

Symbol	F-36 TADD | S_1 | S_2 | D					
Function	Adds the contents of registers S_1 to S_1+2 and the contents of S_2 to S_2+2 for time as hours, minutes and seconds, and stores the results to D to D+2.					
Operation	(S_1 to S_1+2) + (S_2 to S_2+2) → (D to D+2)					
S_1	Use range E					
S_2	Use range E					
D	Use range E					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	S_1 to S_1+2	Unchanged				
	S_2 to S_2+2	Unchanged				
	D to D+2	Result				
	Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		0	1	0	0	1
		000001 to 235959	0	0	0	1
		000000 (tomorrow)	1	1	0	0
		000000 or more	0	1	0	0
		Excepting time	0	0	1	0

- If 24 hours are exceeded and the time changes to the next day, carry flag (007356) turns ON.

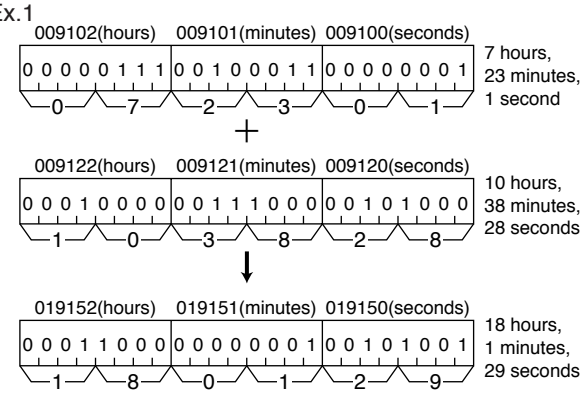
[Explanation]

Instruction	
S T R	004001
F -36	009100
	009120
	009150

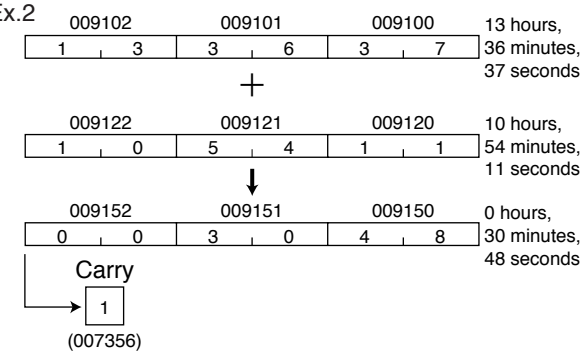
004001 | F-36 TADD | 009100 | 009120 | 009150

When input condition 004001 changes from OFF to ON, the contents (6-digit BCD) of registers 009100 to 009102 are added to the contents (6-digit BCD) of registers 009120 to 009122, and the results are stored to registers 009150 to 009152.

- Ex.1



- Ex.2



F-37 TSUB — Subtracts time

Symbol		F-37 TSUB | S_1 | S_2 | D
Function		Subtracts the contents of registers S_2 to S_2+2 from the contents of S_1 to S_1+2 for time as hours, minutes and seconds, and stores the results to D to D+2.
Operation		(S_1 to S_1+2) + (S_2 to S_2+2) → D to D+2
S_1		Use range E
S_2		Use range E
D		Use range E
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1 to S_1+2	Unchanged
	S2 to S2+2	Unchanged
	D to D+2	Result

Flag

Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0	1
000001 to 235959	0	0	0	1
Negative number	0	1	0	0
Excepting time	0	0	1	0

- If 0 hour is exceeded and the time changes to the previous day, carry flag (007356) turns ON.

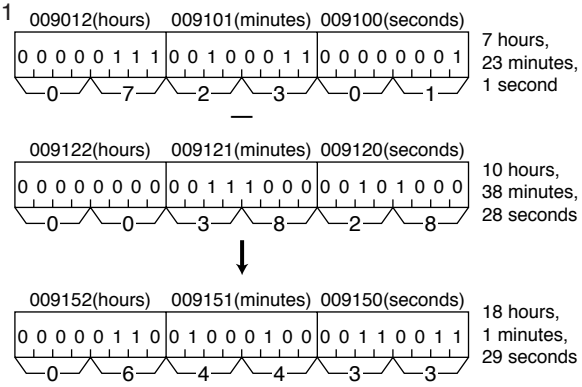
[Explanation]

Instruction	
S T R	004002
F -37	009100
	009120
	009150

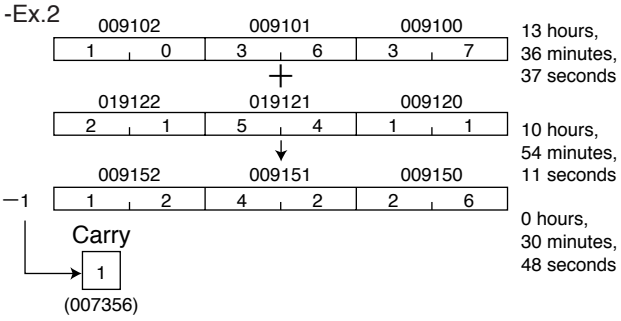
004002 | F-37 TSUB | 009100 | 009120 | 009150

When input condition 004002 changes from OFF to ON, the contents (6-digit BCD) of registers 009100 to 009102 are subtracted from the contents (6-digit BCD) of registers 009120 to 009122, and the results are stored to registers 009150 to 009152.

-Ex.1



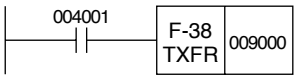
-Ex.2



F-38 TXFR Transfers current value of clock

Symbol	F-38 TXFR | D	
Function	Transfers the current value of the clock (hours, minutes, seconds) to registers D to D+2.	
Operation	Current value of clock → D to D+2	
D	Use range E	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D to D+2	Clock data
	Flag	Unchanged

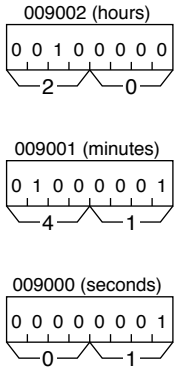
[Explanation]



Instruction	
S T R	004001
F -38	
	009000

When input condition 004001 changes from OFF to ON, the current value of the clock is transferred to registers 009000, 009001 and 009002.

Current value of clock 20 hours, 41 minutes, 1 second →



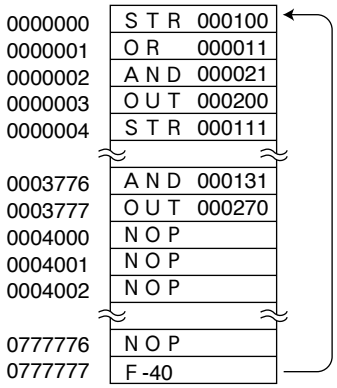
F-40
END

End instruction (END)

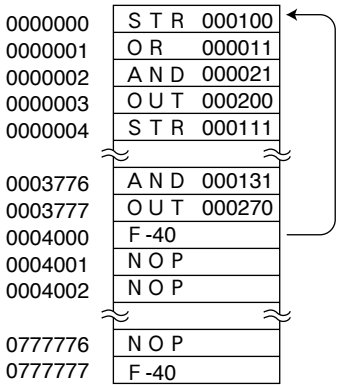
The F-40 instruction indicates the end of the program.
There is no need of writing the END instruction, except for the following cases, because it will be automatically set in the last address of the program memory when the memory is cleared after specifying the program memory capacity.

(1) Saving scan time

- The scan time is a time the I/O processing time added with the user program execution time. The user program execution time is the total time required to execute all instructions from the program address 0000000 to the END instruction.
- The location of the END instruction automatically written after memory clear is the position of the END instruction, which is automatically written by clearing the program memory, will be 0777777 (256th word) of the main block if the control module is JW-362CU.
- Assume now if the last address is 0003777 (2048th word) when the ladder chart has written by the programmer, 0004000 to 0777776 are padded with NOP instruction with the END instruction in 0777777, so that it requires an unwanted time in order to execute NOP instructions.
- If F-40 is written in 0004000, it saves the processing time without executing those unwanted NOP instructions, and the control can proceed to a next scan cycle after termination of the user program.



(a) END (0777777) after mere memory clear



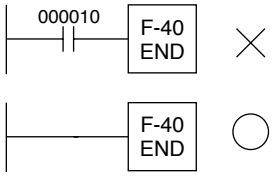
(b) F-40 (END) written in 0004000

- If you are using multiple blocks, write F-40 to each block.

(2) To perform a partial program execution during trial run

By inserting the F-40 instruction at an end of a sequence block, you will be able to execute only the required portion of the program. If the result was successful, that F-40 may then be deleted.

- A multiple number of F-40 may exist when the END instruction is written in (1) and (2). In this case, the user program terminates at the first F-40. So, it would be necessary to check the location of the END instruction before going into the actual operation.
- The F-40 (END) instruction is unconditional, and it prohibits the following kind of programming.

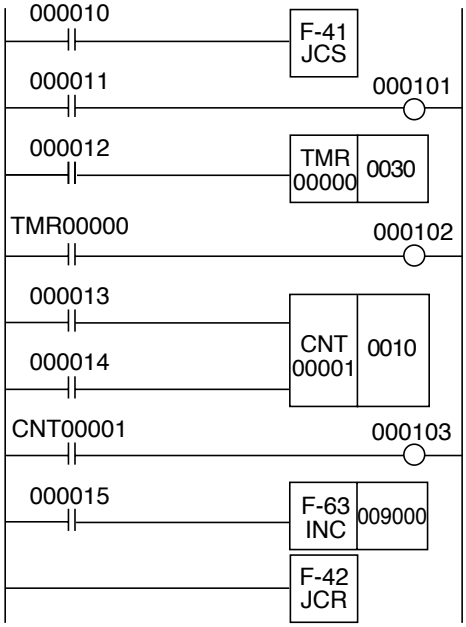


- Although the F-40 instruction has the highest priority, it will be ignored if an END instruction exists between F-141 (JMP) and F-140 (LABL) or between F-142 (CALL) and F-140 (LABL) and that F-141 or F-142 was executed.

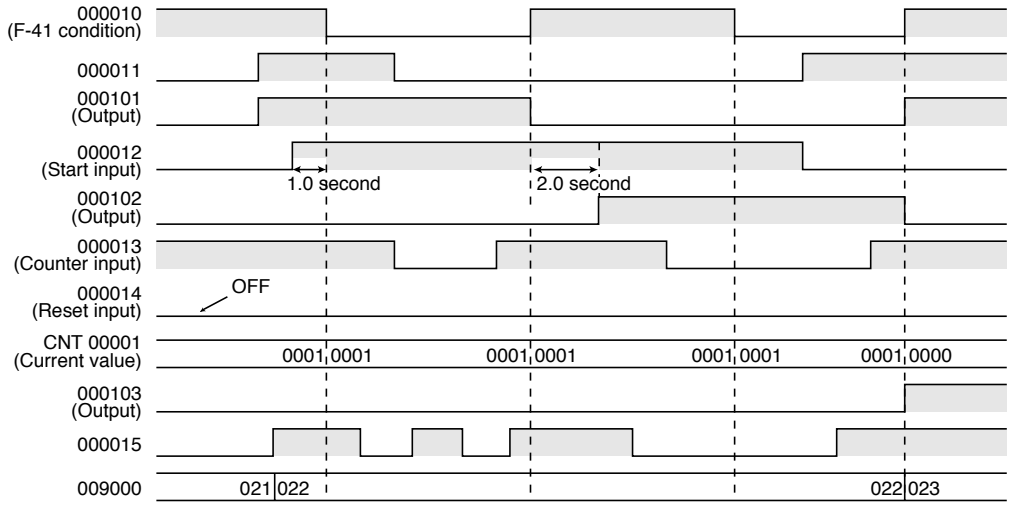
F-41 JCS — Set jump control (Jump Control Set)

F-42 JCR — Reset jump Control (Jump Control Reset)

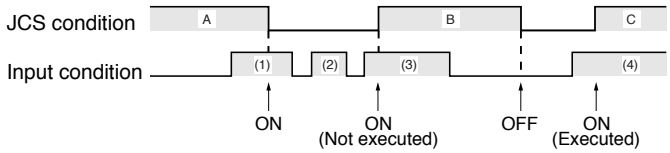
When the F-41 (JCS) condition is OFF, all instruction will not be executed, except when the F-42 (JCR) instruction that exists before the END instruction. Therefore, it does affect the contents of the data memory, even if there is an instruction that writes the result in the data memory, such as OUT, TMR, CNT, MD, and application instruction. And, it retains the state when JCS condition is ON.



Instruction	
→ S T R	000010
F -41	
S T R	000011
O U T	000101
S T R	000012
T M R	00000
	0030
S T R	T M R 00000
O U T	000102
S T R	000013
S T R	000014
C N T	00001
	0010
S T R	C N T 00001
O U T	000103
S T R	000015
F -63	
	009000
→ F -42	

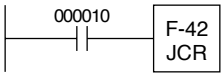


- Attention must be paid to the TMR internal clock (0.1 second clock), CNT counter input and application instruction input condition (that which the operation takes place at an OFF to ON transition in the input condition), and F-41 (JCS) condition ON/OFF timing.

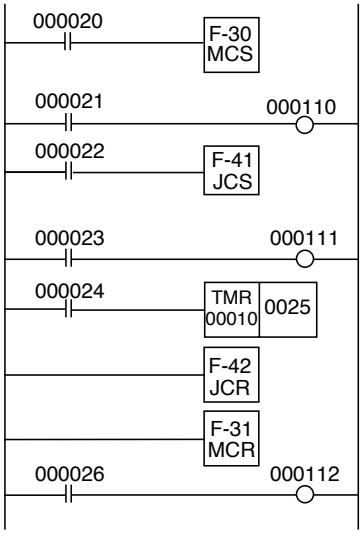


  - Operation takes place at a rising of (1) because the JCS condition is ON.
  - Operation does not take place at a rising of (2) because the JCS condition is OFF.
  - Operation does not take place at a rising of (3) because the JCS condition is OFF.
  - The JCS condition turns to ON while (3) is ON, but operation dose not take place as it does not recognize that the input condition has changed from OFF to ON, because the input condition is ON with which the JCS condition of Ⓐ changes from ON to OFF and the input condition is ON with which the JCS condition of Ⓑ changes from OFF to ON.
  - Operation does not take place at a rising of (4) because the JCS condition is OFF.
  - The JCS condition becomes ON while (4) is ON. Operation takes place immediately after the JCS condition of Ⓑ changes from OFF to ON because there was no change in that the input condition is OFF with which the JCS condition of Ⓒ changes from ON to OFF and the input condition is ON with which the JCS condition changes from OFF to ON.

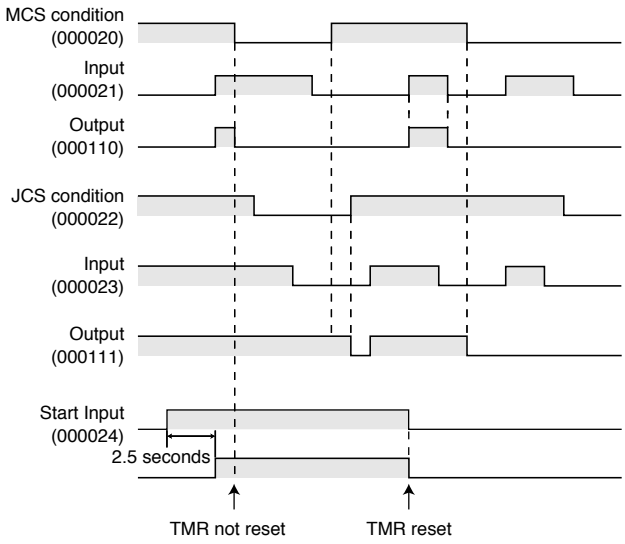
- The END instruction will be executed regardless whether the JCS condition be ON or OFF when there is a F-40 (END) instruction between F-41 (JCS) and F-42 (JCR). And the user program execution is terminated and the control proceeds to a next scan cycle.
- F-41 (JCS) and F-42 can be nested between F-30 (MCS) and F-31 (MCR). When the MCS condition becomes OFF, the instructions between JCS and JCR will be not executed regardless whether the JCS condition be ON or OFF. It is not possible to insert another F-40 and F-41 between F-41 (JCS) and F-42 (JCR). It will evoke "JCS ERROR" on the hand-held programmer during program check, if such a program was written. When nesting like above shows, we use F-242 (JCRN) "Reset jump control nesting" instruction instead of F-42 (JCR).
- The following kind programming is not possible as F-42 (JCR) is an unconditional instruction.



- To use an application instruction that operates at a rising edge of F-41 (JCS) and F-42 (JCR), the input condition must be different from F-41 (JCS). No operation will take place if the condition is same.
- F-41 (JCS) and F-42 (JCR) can be nested between F-30 (MCS) and F-31 (MCR). When the MCS condition becomes OFF, the instructions between JCS and JCR will be not executed regardless whether the JCS condition be ON or OFF.



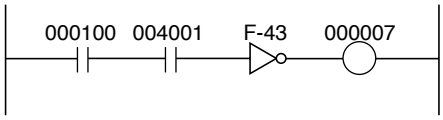
Instruction	
S T R	000020
F -30	
S T R	000021
O U T	000110
S T R	000022
F -41	
S T R	000023
O U T	000111
S T R	000024
T M R	00010
	0025
F -42	
F -31	
S T R	000026
O U T	000112



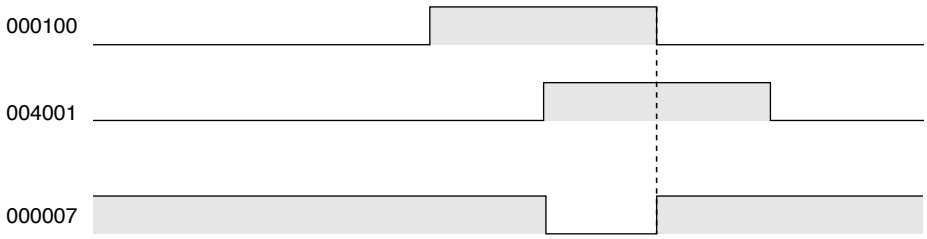
F-43
CPL

Complement bit (ComPLement)

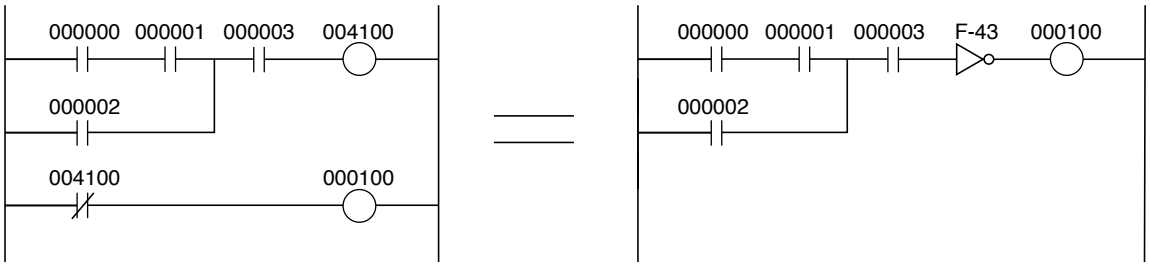
The F-43 instruction complements the bit in the accumulator.



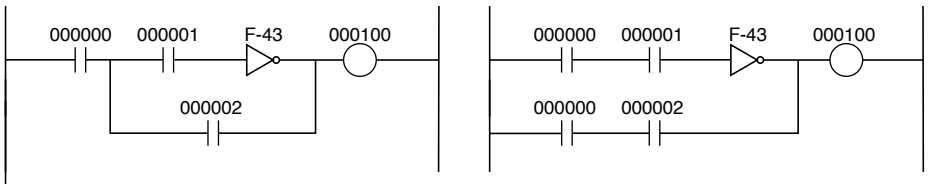
instruction	
S T R	000100
A N D	004001
F -43	
O U T	000007



- Results from the STR instruction to the F-43 instruction are complemented and sent to the output relay 000007.
- Use of F-43 permits to obtain the complemented output without the use of auxiliary contact.



- The F-43 instruction may be used for a single or a multiple number of contacts.
- Pay attention that the following programs (a) and (b) will not produce the same result because the F-43 instruction is the instruction that complement the contents of the accumulator.

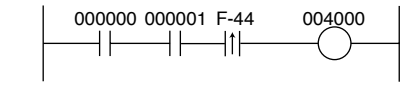


(a) (b)

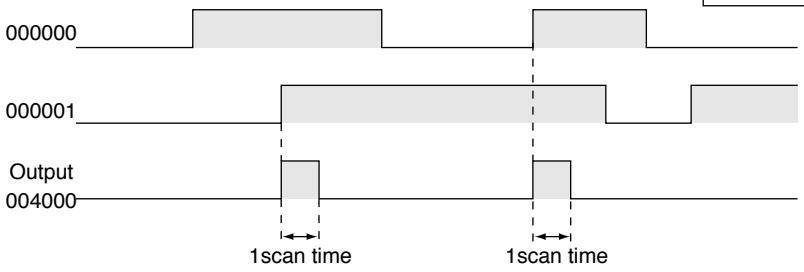
F-44 Differentiate at ON

The F-44 instruction sends 1 scan time pulse when the immediate state of the accumulator changes from OFF to ON.

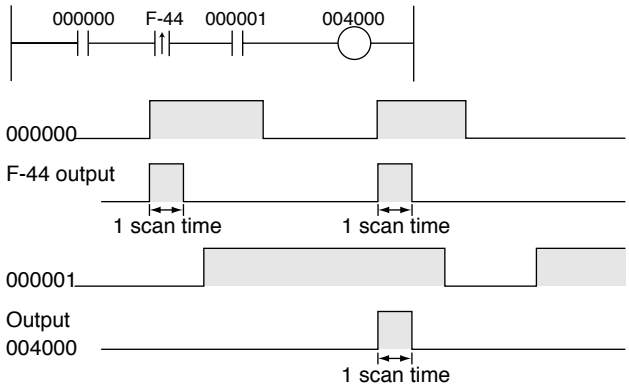
[Use example]



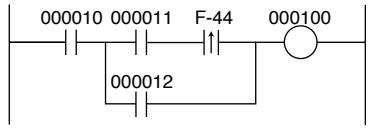
Instruction	
S T R	000000
A N D	000001
F -44	
O U T	004000



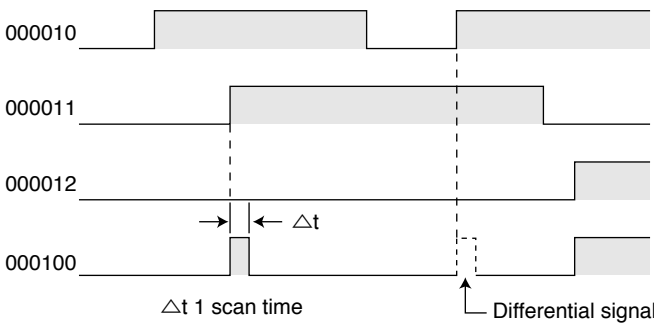
- Note that a different result is produced when the sequence of the F-44 program is changed in the above ladder chart. (Identical in the case of F-45.)



- F-44 condition may be a single or a multiple number of contacts.
- In the below example, a differential signal is not issued even if 000010 has changed from OFF to ON when 000011 is ON, because 000010 is ANDed by the AND STR instruction.



S T R	000010
S T R	000011
F -44	
O R	000012
A N D S T R	
O U T	000100



	Accumulator ACC	Stack register S_1
S T R 000010	000010	
S T R 000011	000011	000010
F -44	000011 F-44	000010
O R 000012	000011 F-44 000012	000010
A N D S T R	000010 000011 F-44 00012	
O U T 000100	000010 000011 F-44 000012	

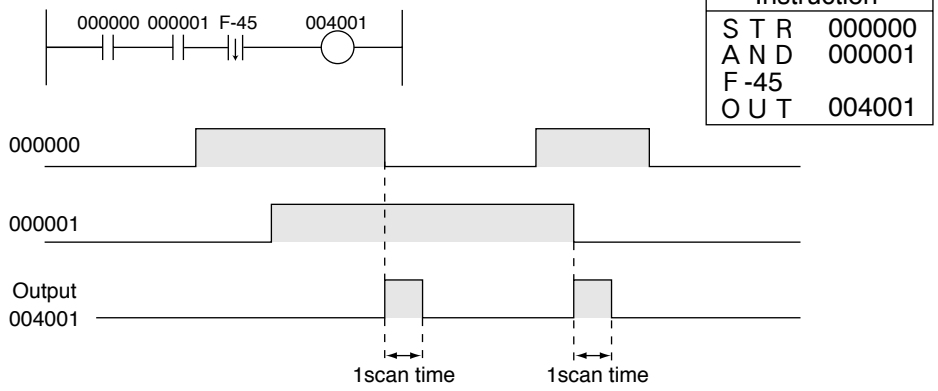
The accumulator turns ON for a period of a scan time that 000011 changed from OFF to ON.

- The F-44 instruction scans only once even if inserted between an F-47 (level operation condition set) and a F-48 (level operation condition reset).

F-45 Differentiate at OFF

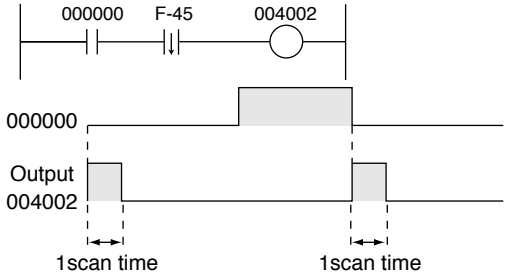
The F-45 instruction sends 1 scan time pulse when the immediate state of the accumulator changes from ON to OFF.

[Use example]



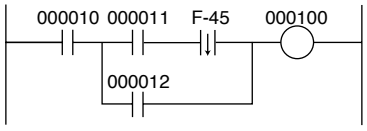
Instruction	
S T R	000000
A N D	000001
F -45	
O U T	004001

- You may use differential instruction (F-44, F-45) as many times as required in the program.



Output (004002) turns ON if input (000000) is OFF at the start of operation immediately after write of program.

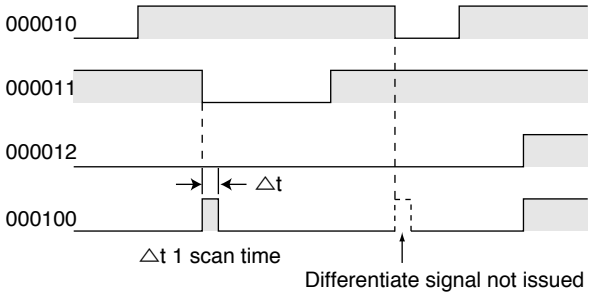
- In the below example, differentiate signal is not issued even if 000010 changed from ON to OFF when 000011 is ON, because 000010 is ANDed by the AND STR instruction.



S T R	000010
S T R	000011
F -45	
O R	000012
A N D S T R	
O U T	000100

	Accumulator ACC	Stack register S_1
S T R 000010	000010	
S T R 000011	000011	000010
F -45	000011 F-45	000010
O R 000012	000010 F-45 000012	000010
A N D S T R	000010 000011 F-45 000012	
O U T 000100	000010 000011 F-45 000012	

The accumulator turns ON for a period of a scan time that 000011 changed from ON to OFF.

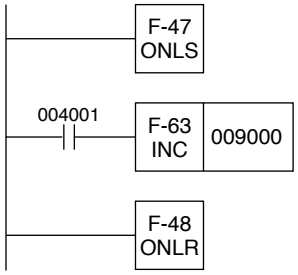


- An F-45 instruction can only be executed for a single scan time, if inserted between an F-47 instruction (level operation condition set) and an F-48 instruction (level operation condition reset).

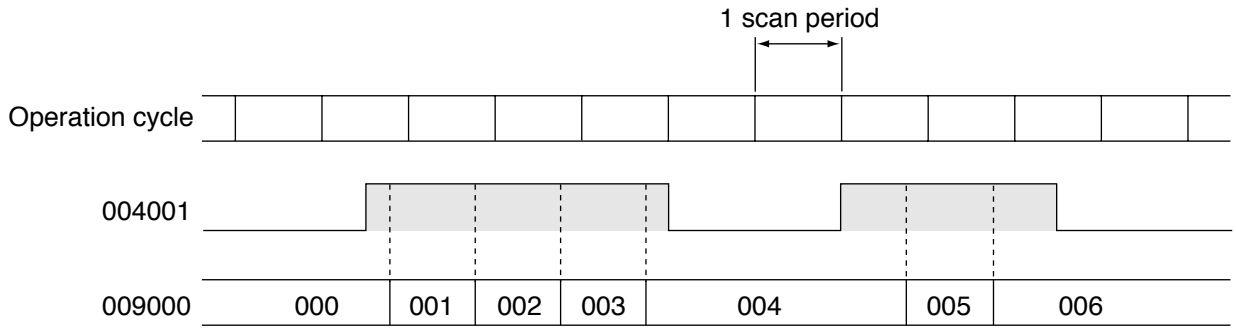
F-47 ONLS — ON level set (ON Level Set)

F-48 ONLR — ON level reset (ON Level Reset)

Rising of a signal between F-47 (ONLS) and F-48 (ONLR) is used for an level operational condition (started with a rising edge of the signal).

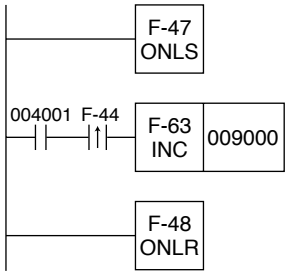


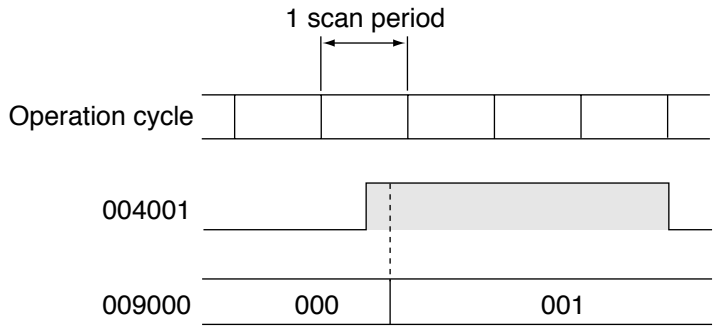
Instruction	
F -47	
S T R	004001
F-63	
	009000
F -48	



- Another F-47 may not be nested between already existing F-47 (ONLS) and F-48 (ONLR).
- Only 1 scan will take place at a rising edge of 004001, if it contains a differentiate instruction (F-44, F-45).

[Example for F-44]



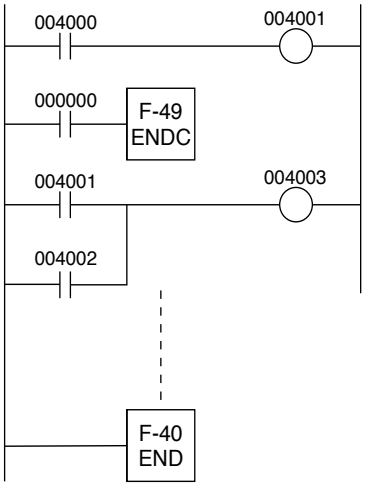


F-49
ENDC

Conditional end

When the condition of F-49 is OFF, the sequential operation terminates.

[Use example]



Address	Instruction	
0000010	S T R	004000
0000011	O U T	004001
0000012	S T R	000000
0000013	F -49	
0000014	S T R	004001
0000015	O R	004002
0000016	O U T	004003
0036777	F -40	

- When the input condition 000000 is ON, instructions down to F-40 are executed (address 0036777).
- When the input condition 000000 is OFF, instructions after the address 0000014 are not executed.

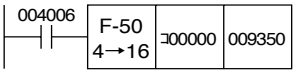
Chapter 12 Application instructions (F-50 to F-79d)

F-50 4→16 Decode 4 to 16

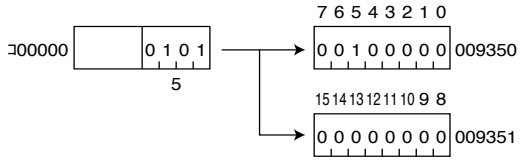
Symbol	F-50 4→16 | S | D	
Function	The lower 4 bits of the register S are decoded and stored in the 2-byte area of the registers D and D+1.	
Operation	S→D, D+1	
S	Use range A - Upper 4 bits of register S are ignored for calculation.	
D	Use range B	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Result (0 to 7)
	D+1	Result (8 to 15)
	Flag	Unchanged

[Explanation]

Instruction	
S T R	004006
F -50	
	⊐00000
	009350



When the input condition 004006 changes from OFF to ON, the lower 4 bits of ⊐00000 are decoded and stored as 16-bit data in the 2-byte area consisting of registers 009350 and 009351.



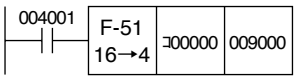
Only the bit position that corresponds to the lower 4 bits representing 0 to 15 is turned ON, and other bits are turned OFF.

F-51 16→4 Encode 16 to 4

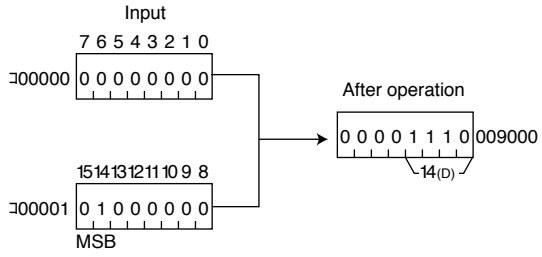
Symbol	F-51 16→4 | S | D	
Function	The 2-byte data in registers S and S+1 are encoded and stored in the register D.	
Operation	S, S+1→D	
S	Use range B	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1	Unchanged
	D	Result
	Flag	Unchanged

[Explanation]

Instruction	
S T R	004001
F -51	
	⊐00000
	009000

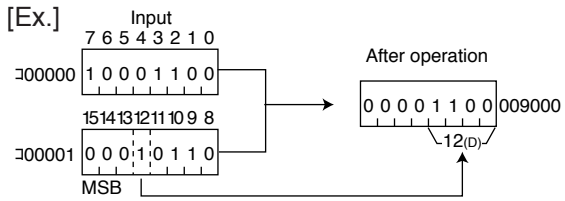


When the input condition 004001 changes from OFF to ON, the 2-byte data consisting of registers ⊐00000 and ⊐00001 are encoded and stored in the register 009000.



- After the operation, the upper 4 bits of D (009000 in the example) becomes 0 at all times.
- The MSB of the encoder input takes the highest priority.

[Ex.]



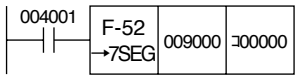
F-52
→7SEG

Decode to 7-segment data

Symbol	F-52 →7SEG | S | D	
Function	The lower 4 bits data in register S is decoded into 7-segment display data.	
Operation	S→D	
S	Use range A	
D	Use range A *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Result => "7-segment decoder chart"
	Flag	Unchanged

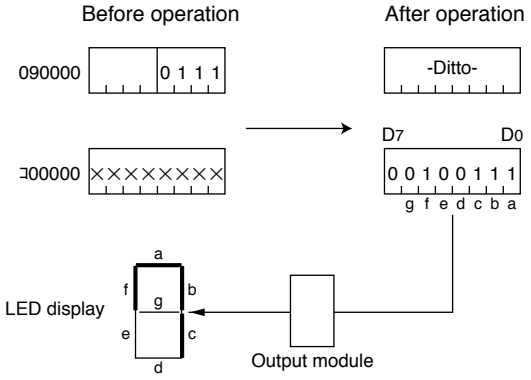
* Output data D0 to D6 to register D correspond to "a" to "g" of the 7-segment display. D_7 is already "0."

[Explanation]



Instruction	
S T R	004001
F -52	
	009000
	□00000

When the input condition 004001 changes from OFF to ON, the lower 4 bits of the register 009000 are decoded into the 7-segment display data. See "7-segment decoder chart" relation between input data and display output.



7-segment decoder chart

Input data	Output data g f e d c b a	Display output
0 0 0 0 0 0 0 0	0 0 1 1 1 1 1 1	0
0 0 0 0 0 0 0 1	0 0 0 0 0 1 1 0	1
0 0 0 0 0 0 1 0	0 1 0 1 1 0 1 1	2
0 0 0 0 0 0 1 1	0 1 0 0 1 1 1 1	3
0 0 0 0 0 1 0 0	0 1 1 0 0 1 1 0	4
0 0 0 0 0 1 0 1	0 1 1 0 1 1 0 1	5
0 0 0 0 0 1 1 0	0 1 1 1 1 1 0 1	6
0 0 0 0 0 1 1 1	0 0 1 0 0 1 1 1	7
0 0 0 0 1 0 0 0	0 1 1 1 1 1 1 1	8
0 0 0 0 1 0 0 1	0 1 1 0 1 1 1 1	9
0 0 0 0 1 0 1 0	0 1 1 1 0 1 1 1	A
0 0 0 0 1 0 1 1	0 1 1 1 1 1 0 0	b
0 0 0 0 1 1 0 0	0 0 1 1 1 0 0 1	C
0 0 0 0 1 1 0 1	0 1 0 1 1 1 1 0	d
0 0 0 0 1 1 1 0	0 1 1 1 1 0 0 1	E
0 0 0 0 1 1 1 1	0 1 1 1 0 0 0 1	F

F-53 →BIN Convert 4 digits BCD to 16 bits binary

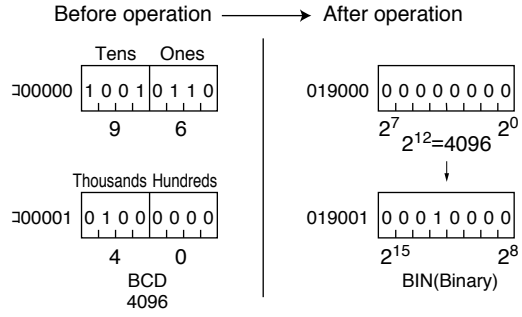
Symbol	F-53 →BIN S D				
Function	4-digit BCD data of 2 bytes in registers, S and S+1, are converted into a binary equivalent and stored in 2 bytes in registers D and D+1.				
Operation	S, S+1→D, D+1				
S	Use range B				
D	Use range B				
Condition	Rising edge of input signal (OFF to ON)				
Contents after operation	S, S+1	Unchanged			
	D	Result (0 to 255)	Unchanged when the contents of registers, S and S+1, are not BCD code.		
	D+1	Result (256 to 9999)			
	Flag	S, S+1	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		BCD code	0	0	0	0
		Not BCD code			1	

[Explanation]

004001 | F-53 →BIN | ⊐00000 | 019000

Instruction	
S T R	004001
F -53	
	⊐00000
	019000

When the input condition 004001 changes from OFF to ON, the contents of the 4-digit BCD data in registers ⊐01000 and ⊐019001 are converted into a binary equivalent and stored in 2-byte area of registers 019000 and 019001.



Note: When the F-53 instruction is used for programming, the F-03w instruction provides program monitor display.

Resembled instructions: F-03, F-03w, F-03d, F-153

F-54 →BCD Convert 16 bits binary to 6 digits BCD

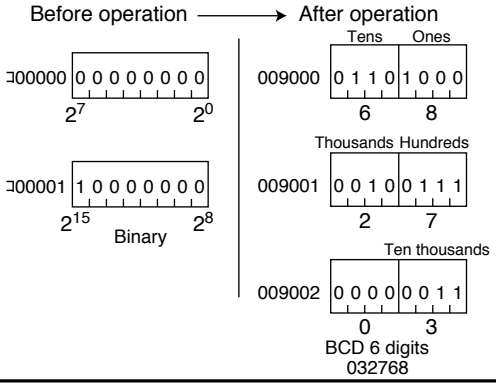
Symbol	F-54 →BCD S D	
Function	2-byte binary data in registers S, S+1 are converted into BCD 6 digits equivalent and stored in 3 bytes in registers D, D+1, and D+2.	
Operation	S, S+1→D, D+1, D+2	
S	Use range B	
D	Use range E	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1	Unchanged
	D	Result (ones and tens)
	D+1	Result (hundreds and thousands)
	D+2	Result (ten thousands)
	Flag	Unchanged

[Explanation]

004001 | F-54 →BCD | ⊐00000 | 009000

Instruction	
S T R	004001
F -54	
	⊐00000
	009000

When the input condition 004001 changes from OFF to ON, the 2-byte contents of binary data in registers, ⊐00000 and ⊐00001, are converted into BCD 6 digits and stored in 3-byte area of registers 009000 to 009002.



Note: When the F-54 instruction is used for programming, the F-04w instruction provides program monitor display.

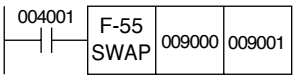
Resembled instructions: F-04, F-04w, F-04d, F-154

F-55
SWAP

Swap upper 4 bits with lower 4 bits

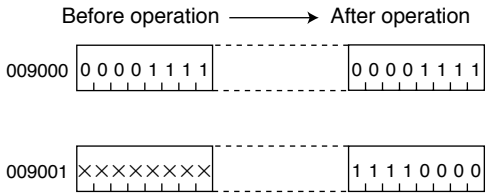
Symbol	F-55 SWAP | S | D	
Function	Upper 4 bits are swapped with lower 4 bits of the register S and stored in the register D.	
Operation	S→D	
S	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Result
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004001
F -55	
	009000
	009001

When the input condition 004001 changes from OFF to ON, upper 4 bits are swapped with lower 4 bits of the register 009000 and its result is stored in the register 009001.
The contents of the register 009000, however, remains unchanged.



Reference

- F-55 instruction becomes valid in the following case.
- With the F-52 (7 SEG decode) instruction, the lower 4 bits are decoded into the 7-segment display data. To display a multiple number of digits on the display, the upper 4 bits need to be swapped with the lower 4 bits, then F-52 should be executed.

Resembled instructions: F-175

F-56 NEG — Complement of 10 of 1-byte data

Symbol	F-56 NEG | S | D
Function	The 1 byte contents of the register S are assumed to be a 2-digit BCD code. Complement of 10 of that value is obtained and stored in the register D.
Operation	100−S→D
S	Use range A
D	Use range A
Condition	Rising edge of input signal (OFF to ON)

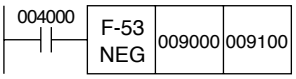
Contents after operation

S	Unchanged
D	Result - Unchanged when the contents of the register S is not a BCD code.

Flag

Contents of register S	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
BCD code	0	0	0	0
Not BCD code			1	

[Explanation]



Instruction	
S T R	004000
F -56	
	009000
	009100

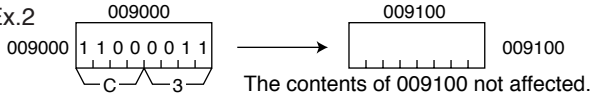
When the input condition 004000 changes from OFF to ON, the contents of the register 009000 are assumed to be 2-digit BCD code and complement of 10 of that value is obtained and stored in the register 009100.
If the contents of the register 009000 are not a BCD code, the contents of 009100 are unchanged and the error flag is set 1.

- Ex.1

100−58=42

Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	0	0	0

- Ex.2



The contents of 009100 not affected.

Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	0	1	0

F-56w NEG — Complement of 10 of 1-word data

Symbol	F-56w NEG | S | D
Function	The word contents of the registers S and S+1 are assumed to be a 4-digit BCD code. Complement of 10 of that value is obtained and stored in the registers D and D+1.
Operation	10000− (S, S+1) →D, D+1
S	Use range B *
D	Use range B *
Condition	Rising edge of input signal (OFF to ON)

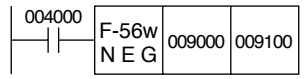
Contents after operation

S, S+1	Unchanged	
D	Lower digits of result	Unchanged when the contents of the register S, S+1 are not a BCD code.
D+1	Upper digits of result	

Flag

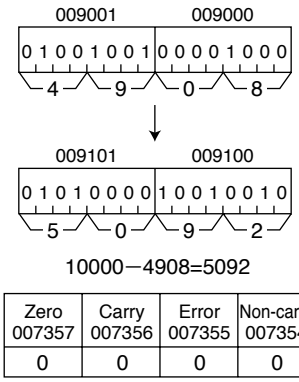
Contents of register S, S+1	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
BCD code	0	0	0	0
Not BCD code			1	

[Explanation]



Instruction	
S T R	004000
F -56w	
	009000
	009100

When the input condition 004000 changes from OFF to ON, the contents fo the registers 009000 and 009001 are assumed to be 4-digit BCD code and complement of 10 of that value is obtained and stored in the registers 009100 and 009101.



10000−4908=5092

Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	0	0	0

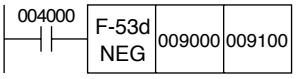
* Be sure to use even addresses for registers S and D. (Such as 009003 etc. are prohibited.)

F-56d NEG Complement of 10 of 2-word data

Symbol	F-56d NEG | S | D	
Function	Obtain the 10's complement of the 8-digit BCD contents (2-word data) of registers S to S+3, and store the result in registers D to D+3.	
Operation	100000000 − (S to S+3) →D to D+3	
S	Use range B *	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S to S+3	Unchanged
	D to D+3	Result - Unaffected if the contents of registers S to S+3 are not BCD.

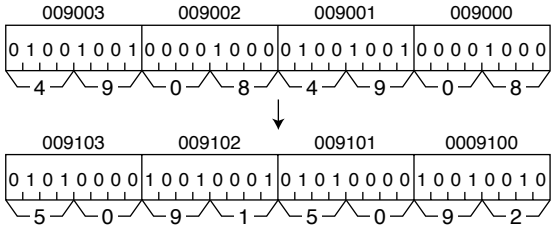
Flag	Contents of registers S to S+3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	BCD code	0	0	0	0
	Not BCD code			1	

[Explanation]



Instruction	
S T R	004000
F -56d	
	009000
	009100

When the input condition 004000 changes from OFF to ON, the contents of the registers 009000 to 009003 are assumed to be 8-digit BCD code and complement of 10 of that value is obtained and stored in the registers 009100 to 009103.



100000000−49084908=50915092

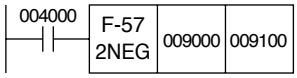
Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	0	0	0

* Be sure to use even addresses for registers S and D.
(Odd addresses such as 009003 etc. are prohibited to use.)

F-57 2NEG Complement of 2 of 1-byte data

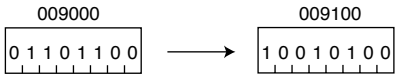
Symbol	F-57 2NEG | S | D	
Function	The 8-bit contents of the register S are complemented to 2 and its result is stored in the register D.	
Operation	0−S→D	
S	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Result
	Flag	Unchanged

[Explanation]



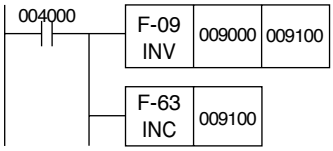
Instruction	
S T R	004000
F -57	
	009000
	009100

When the input condition 004000 changes from OFF to ON, the 8-bit contents of the register 009000 are complemented of 2 and stored in the register 009100.



- How to obtain the complement of 2.
  (1) Complement all bits; 0 to 1 and 1 to 0.
  (2) Add 1 to the above result.

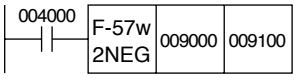
- Operation similar to the following instructions will take place for the above.



F-57w 2NEG Complement of 2 of 1-word data

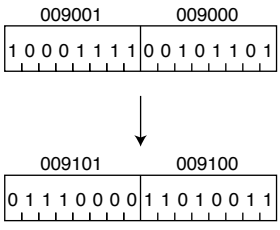
Symbol	F-57w 2NEG | S | D	
Function	The word contents of the registers S and S+1 are complemented to 2 and stored in the registers D, D+1.	
Operation	0− (S, S+1) →D, D+1	
S	Use range B	
D	Use range B	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1	Unchanged
	D	Lower digits of result
	D+1	Upper digits of result
	Flag	Unchanged

[Explanation]

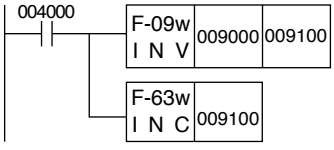


Instruction	
S T R	004000
F -57w	
	009000
	009100

When the input condition 004000 changes from OFF to ON, the 16-bit contents of the registers 009000 and 009001 are complemented to 2 and stored in the registers 009100 and 009101.



- Operation similar to the following instructions will take place for the above.

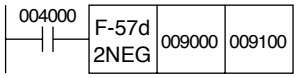


F-57d
2NEG

Complement of 2 of 2-word data

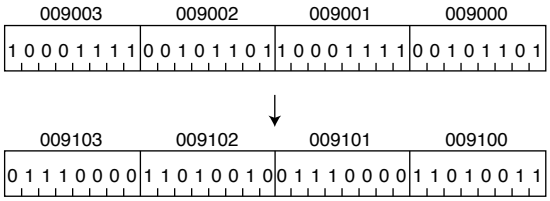
Symbol	F-57d 2NEG | S | D	
Function	Obtains the 2 complement of the contents (2-word data) of registers S to S+3, and store the result in registers D to D+3.	
Operation	0− (S to S+3) →D to D+3	
S	Use range C *	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S to S+3	Unchanged
	D	Lower digits of result
	D to D+3	Upper digits of result
	Flag	Unchanged

[Explanation]

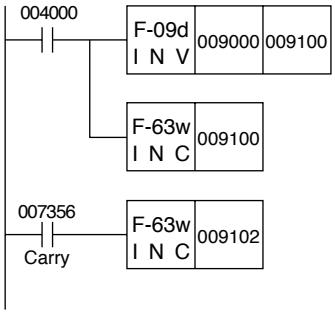


Instruction	
S T R	004000
F -57d	
	009000
	009100

When the input condition of 004000 changes from OFF to ON, this instruction obtains the complement of 2 of the 32-bit content of registers 009000 to 009003, and stores the result in registers 009100 to 009103.



* Be sure to use even addresses for registers S and D. - Operation similar to the following instructions will take place for the above.

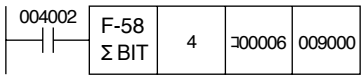


F-58
Σ BIT

Total ON bits

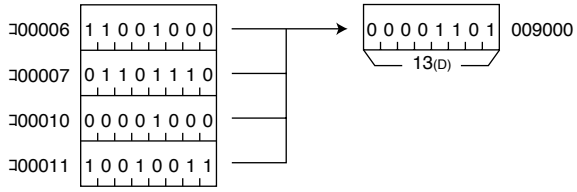
Symbol	F-58 Σ BIT | n | S | D	
Function	All active bits in the "n" byte register having the register S at its top are stored in the register D.	
Operation	ON bits→D	
n	Use range 0 to 7 (8 bytes for 0)	
S	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1,··· S+n−1	Unchanged
	D	Result
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004002
F -58	
	4
	⊐00006
	009000

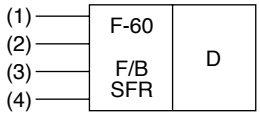
When the input condition 004002 changed from OFF to ON, the total number of active bits in a 4-byte register heralded by the register ⊐00006 is stored in the register 009000.



- Among 32 bits in ⊐00006 to ⊐00011, 13 bits are ON.

F-60 SFR — Shifts register bi-directionally (1 byte) (Forward/Backward Shift Register)

Symbol



(1) Shift direction indication input
(2) Data input
(3) Shift input
(4) Reset input

Function

The 8-bit data in the register D is shifted to upper or lower bit positions according to the shift direction input (1).

Operation

- When the shift direction input is ON:

- When the shift direction input is OFF:

D

Use range A

Condition

When the reset input (4) is OFF, bits are shifted at a rising edge (OFF to ON) of the shift input (3).

Contents after operation

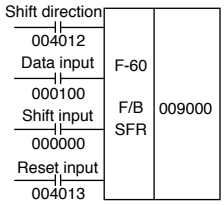
D

- Result is produced when the reset input (4) is OFF.
- All bits are turned OFF when the reset input (4) is ON.

Flag

Reset input (4)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
OFF	0 or 1	0 or 1	0	1 or 0
ON	0	0		0

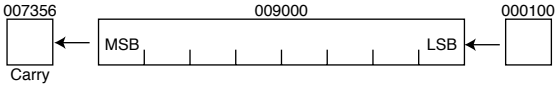
[Explanation]



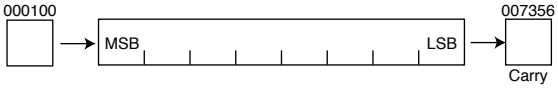
Instruction	
S T R	004012
S T R	000100
S T R	000000
S T R	004013
F -60	009000

When the shift input 000000 changes from OFF to ON, shift takes place in the following way according to the state of the shift direction input 004012.

- When 004012 is ON



- When 004012 is OFF



Input condition	009000 (before operation) 7 6 5 4 3 2 1 0	009000 (after operation) 7 6 5 4 3 2 1 0	Zero 007357	Carry 007356	Non-carry 007354
004012 ○	○ ○ ○ ○ ○ ○ ○ ○	○ ○ ○ ○ ○ ○ ○ ○	●	○	●
000100 ○	● ○ ○ ○ ● ○ ● ○	○ ● ○ ○ ○ ● ○ ●	○	○	●
000000 ↓	○ ● ○ ○ ● ○ ○ ●	○ ○ ● ○ ○ ● ○ ○	○	●	○
004013 ○	○ ○ ○ ○ ○ ○ ○ ●	○ ○ ○ ○ ○ ○ ○ ○	●	●	○
004012 ●	○ ○ ○ ○ ○ ○ ○ ○	○ ○ ○ ○ ○ ○ ○ ○	●	○	●
000100 ○	○ ● ○ ● ○ ○ ○ ○	● ○ ● ○ ○ ○ ○ ○	○	○	●
000000 ↓	● ○ ○ ○ ● ○ ○ ○	○ ○ ○ ● ○ ○ ○ ○	○	●	○
004013 ○	● ○ ○ ○ ○ ○ ○ ○	○ ○ ○ ○ ○ ○ ○ ○	●	●	○
004012 ○ 000100 ●	● ○ ○ ○ ○ ● ○ ○	● ● ○ ○ ○ ○ ● ○	○	○	●
000000 ↓ 004013 ○	○ ○ ○ ● ○ ○ ○ ●	● ○ ○ ○ ● ○ ○ ○	○	●	○
004012 ● 000100 ●	○ ● ○ ○ ○ ○ ○ ○	● ○ ○ ○ ○ ○ ○ ●	○	○	●
000000 ↓ 004013 ○	● ○ ○ ○ ● ○ ○ ○	○ ○ ○ ● ○ ○ ○ ●	○	●	○
004013 ●	○ ○ ● ○ ● ○ ○ ○	○ ○ ○ ○ ○ ○ ○ ○	○	○	○

- Error flag (007355) is OFF at all times.

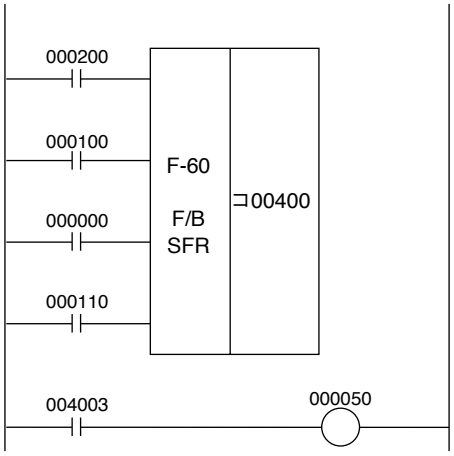
○ OFF ● ON

- By setting the reset condition in the system memory (#0202) for reset input (4), it permits to reset with OFF.

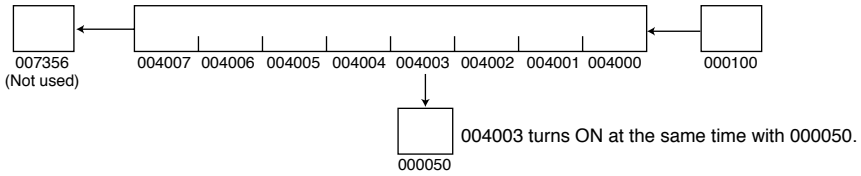
Resembled instructions: F-60w, F-60d, F-160, Fc160

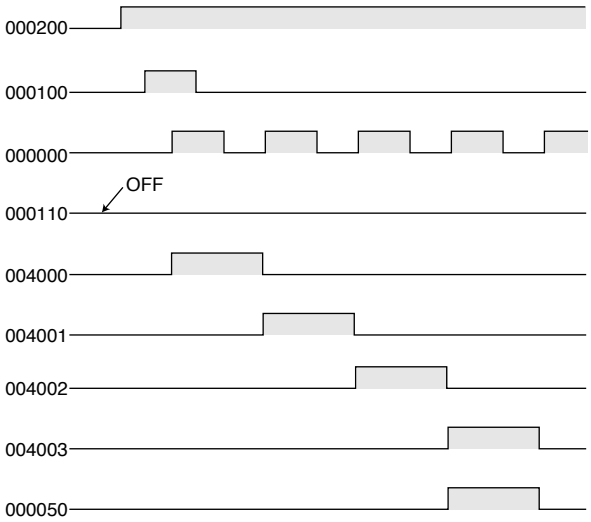
Reference

When ⊐XXXXX is assigned for D, it allows to constitute a "n" bit (n<8) shift register.

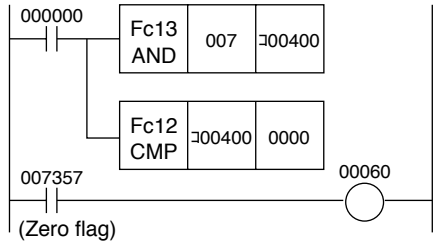


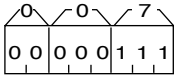
- When 000200 is ON





- Data are shifted in 004004 to 004007.
- When all of 004000 to 004007 are 0, the zero flag turns to 1.
  The following program must be used to check that 004000 to 004002 are 0.



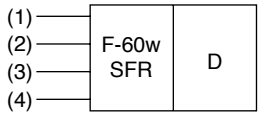


By ANDing this way, 004003 to 004007 are masked (all 0).

F-60w SFR

Shift register bi-directionally (1 word) (Forward/Backward Shift Register)

Symbol



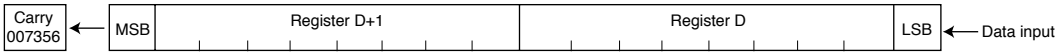
(1) Shift direction indication input
(2) Data input
(3) Shift input
(4) Reset input

Function

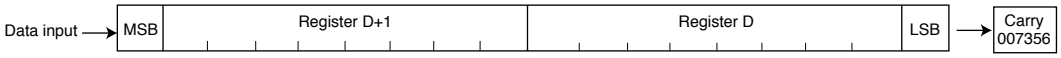
The 16-bit contents of the registers D, D+1 are shifted to upper or lower bit positions according to the direction indication input (1).

Operation

- When the shift direction indication input (1) is ON:



- When the shift direction indication input (1) is OFF:



D

Use range B

- Be sure to use even addresses for register D.
  (Odd addresses such as 019003 etc. are prohibited to use.)

Condition

When the reset input (4) is OFF, bits are shifted at a rising edge (OFF to ON) of the shift input (3).

Contents after operation

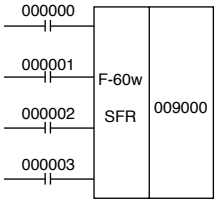
D, D+1

- Result is produced when the reset input (4) is OFF.
- All bits are turned OFF when the reset input (4) is ON.

Flag

Reset input (4)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
OFF	0 or 1	0 or 1	0	1 or 0
ON	0	0		0

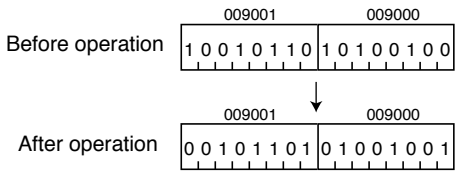
[Explanation]



Instruction	
S T R	000000
S T R	000001
S T R	000002
S T R	000003
F -60w	
	009000

- Input conditions
  000000 (1) ON ---------- Shifts towards MSB
  000001 (2) ON ---------- Data input ON
  000002 (3) OFF→ON ---------- Shift direction
  000003 (4) OFF ---------- No reset function

The following result will be produced for the above input condition.



Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0

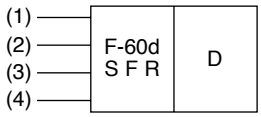
- By setting the reset input (4) in the system memory (#0202), it permits to reset with OFF.

Resembled instructions: F-60, F-60d, F-160, Fc160

F-60d SFR

Shift register bi-directionally (2 words) (Forward/Backward Shift Register)

Symbol



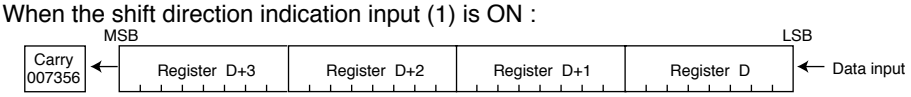
(1) Shift direction indication input
(2) Data input
(3) Shift input
(4) Reset input

Function

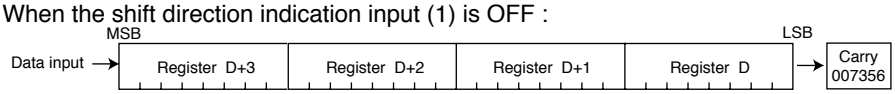
Shift the 32-bit contents of registers D to D+3 toward the MSB or LSB according to the shift direction indication input (1).

Operation

- When the shift direction indication input (1) is ON :



- When the shift direction indication input (1) is OFF :



D

Use range C

- Be sure to use even addresses for register D.
  (Odd addresses such as 019003 etc. are prohibited to use.)

Condition

When the reset input (4) is OFF, shift occurs at the rising edge (OFF to ON) of the shift clock input (3).

Contents after operation

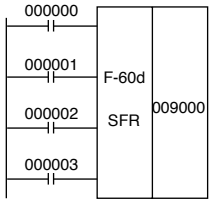
D to D+3

- Result of operation when reset input (4) is OFF.
- All bits OFF when reset input (4) is ON.

Flag

Reset input (4)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
OFF	0 or 1	0 or 1	0	1 or 0
ON	0	0		0

[Explanation]

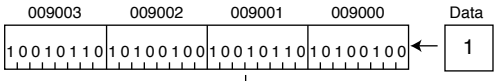


Instruction	
S T R	000000
S T R	000001
S T R	000002
S T R	000003
F -60d	
	009000

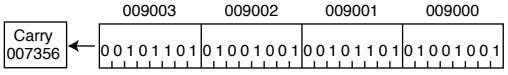
- Input conditions
  000000 (1) ON ---------- Shift towards MSB.
  000001 (2) ON ---------- Data input ON
  000002 (3) OFF→ON ------ Shift direction
  000003 (4) OFF ---------- No reset function

The following result will be produced for the above input condition.

Before operation



After operation



Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0

- By setting the reset condition in the system memory (#0202), it permits to reset input (4) with OFF.

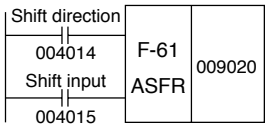
Resembled instructions: F-60, F-60w, F-160, Fc160

F-61 ASFR — Shift register asynchronously (1 byte) (Asynchronous ShiFt Register)

Symbol	(1) — F-61 ASFR — D (2) — (1) Shift direction indication input (2) Shift input
Function	1-byte data in the register D−1 ((1) ON) or the register D+1 ((1) OFF) are shifted to the register D according to the shift direction indication input (1).
Operation	- When the shift direction indication input is ON D−1→D - When the shift direction indication input is OFF D+1→D - The contents of the shift register (D−1 or D+1) are cleared upon execution. - Execution will not take place unless the contents of D are not 0.
D	Use range B
Condition	When the contents of the register are 0, shift is done with the ON state of the shift input (2) (not limited to the OFF to ON change).

		(1) ON		(1) OFF		(1) ON/OFF	
		Before operation	After operation	Before operation	After operation	Before operation	After operation
Contents of register	D−1	D1	0	D1	Same as left	D1	Same as left
	D	0	D1	0	D2	Other than 0	Same as left
	D+1	D2	Same as left	D2	0	D2	Same as left
Flag ★	Non-carry 007354	1 (D1=0) 0 (D1≠0)		1 (D2=0) 0 (D2≠0)		1	
	Error 007355	0		0		0	
	Carry 007356	0 (D1=0) 1 (D1≠0)		0 (D2=0) 1 (D2≠0)		0	
	Zero 007357	0		0		0	

[Explanation]

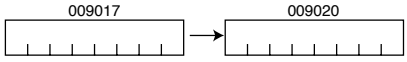


Instruction	
S T R	004014
S T R	004015
F -61	009020

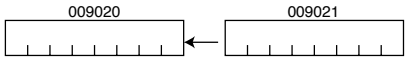
- Shift does not take place unless the contents of 09020 are 0 before the execution.
- The contents of the shifted register (09017 or 09021) are cleared.

A 1 byte data will be shifted in the following way while the shift input 004015 is ON, according to the state of the shift direction input 004014.

- When 004014 is ON



- When 004014 is OFF



Input condition		Before operation 7 6 5 4 3 2 1 0	After operation 7 6 5 4 3 2 1 0	Zero 007357	Carry 007356	Non-carry 007354
004014	●	009017 ○ ○ ● ○ ○ ○ ○ ●	○ ○ ● ○ ○ ○ ○ ●	○	○	●
004015	●	009020 ○ ● ○ ○ ○ ○ ● ○	○ ● ○ ○ ○ ○ ● ○			
		009021 ● ○ ○ ○ ○ ● ○ ○	● ○ ○ ○ ○ ● ○ ○			
004014	●	009017 ○ ○ ○ ○ ● ○ ● ○	○ ○ ○ ○ ○ ○ ○ ○	○	●	○
004015	●	009020 ○ ○ ○ ○ ○ ○ ○ ○	○ ○ ○ ○ ● ○ ● ○			
		009021 ● ● ○ ○ ○ ○ ○ ○	● ● ○ ○ ○ ○ ○ ○			
004014	○	009017 ○ ○ ○ ○ ● ○ ● ○	○ ○ ○ ○ ● ○ ● ○	○	●	○
004015	●	009020 ○ ○ ○ ○ ○ ○ ○ ○	● ● ○ ○ ○ ○ ○ ○			
		009021 ● ● ○ ○ ○ ○ ○ ○	○ ○ ○ ○ ○ ○ ○ ○			
004014	○	009017 ○ ○ ○ ○ ● ○ ● ○	○ ○ ○ ○ ● ○ ● ○	○	○	●
004015	●	009020 ○ ○ ○ ○ ○ ○ ○ ○	○ ○ ○ ○ ○ ○ ○ ○			
		009021 ○ ○ ○ ○ ○ ○ ○ ○	○ ○ ○ ○ ○ ○ ○ ○			

- The error flag (007355) will be OFF at all times.

○ OFF ● ON

★ The carry flag (007356) turns ON only when data other than 0 are shifted from D−1 or D+1.

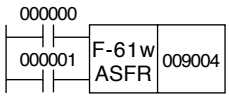
Resembled instructions: F-61w, F-61d, F-161, F-161w, F-161d

F-61w ASFR Shift register asynchronously (1 word) (Asynchronous Shift Register)

Symbol	(1) F-61w ASFR D (2) (1) Shift direction indication input (2) Shift input
Function	1 word in the registers D−2 and D−1 ((1) ON) or the register D+2 and D+3 ((1) OFF) are shifted to the registers D and D+1, according to the shift direction indication input (1).
Operation	- When the shift direction indication input is ON D−2, D−1→D, D+1 - When the shift direction indication input is OFF D+2, D+3→D, D+1 - When the operation is executed, the contents of shifted registers (D-2 to D-1 or D+2 to D+3) will be cleared. - When the contents of registers D and D+1 are other than 0, the JW300 does not operate.
D	Use range C (Be sure to use even addresses for register D.)
Condition	When the contents of the registers D, D+1 are 0, shift is done with the ON state of the shift input (2). (not limited to the OFF to ON change)

		(1) ON		(1) OFF		(1) ON/OFF	
		Before operation	After operation	Before operation	After operation	Before operation	After operation
Contents of register	D−2, D−1	D1	0	D1	Same as left	D1	Same as left
	D, D+1	0	D1	0	D2	Other than 0	Same as left
	D+2, D+3	D2	Same as left	D2	0	D2	Same as left
Flag	Non-carry 007354	1 (D1=0) 0 (D1≠0)		1 (D2=0) 0 (D2≠0)		1	
	Error 007355	0		0		0	
	Carry 007356	0 (D1=0) 1 (D1≠0)		0 (D2=0) 1 (D2≠0)		0	
	Zero 007357	0		0		0	

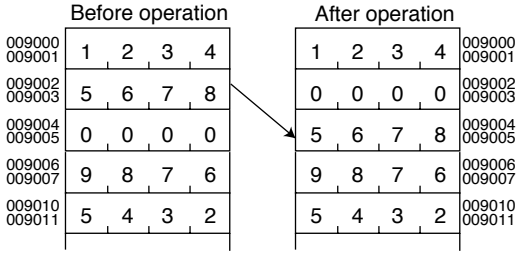
[Explanation]



Instruction	
S T R	000000
S T R	000001
F -61w	009004

00000 (1) ON --------Shifts from 009002 and 009003.
00001 (2) ON --------Shift direction
Contents of 009004 and 009005 are 0000(H).

The following result will be produced for the above input condition.



Only the carry flag (007356) is set ON.

Resembled instructions: F-61, F-61d, F-161, F-161w, F-161d

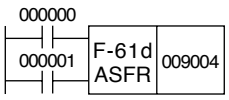
F-61d ASFR

Shift register asynchronously (2 words) (Asynchronous Shift Register)

Symbol	(1) — F-61d ASFR D (2) — (1) Shift direction indication input (2) Shift input
Function	Shift the contents (2-words data) of registers D−4 to D−1 (when input (1) is ON) or registers D+4 to D+7 (when input (1) is OFF) into registers D to D+3 according to the shift direction indication input (1).
Operation	- When the shift direction input is ON D−4 to D−1→D to D+3 - When the shift direction input is OFF D+4 to D+7→D to D+3 - When the operation is executed, the contents of shifted registers (D-4 to D-1 or D+4 to D+7) will be cleared. - When the contents of registers D to D+3 are other than 0, the JW300 does not operate.
D	Use range G (Be sure to use even addresses for register D.)
Condition	If the contents of registers D to D+3 are 0, shift is done with the ON state of the shift input (2) (not limited to OFF to ON change)

		(1) ON		(1) OFF		(1) ON/OFF	
		Before operation	After operation	Before operation	After operation	Before operation	After operation
Contents of register	D−4 to D−1	D1	0	D1	Same as left.	D1	Same as left.
	D to D+3	0	D1	0	D2	Other than 0	Same as left.
	D+4 to D+7	D2	Same as left.	D2	0	D2	Same as left.
Flag	Non-carry (007354)	1 (D1=0) 0 (D1≠0)		1 (D2=0) 0 (D2≠0)		1	
	Error (007355)	0		0		0	
	Carry (007356)	0 (D1=0) 1 (D1≠0)		0 (D2=0) 1 (D2≠0)		0	
	Zero (007357)	0		0		0	

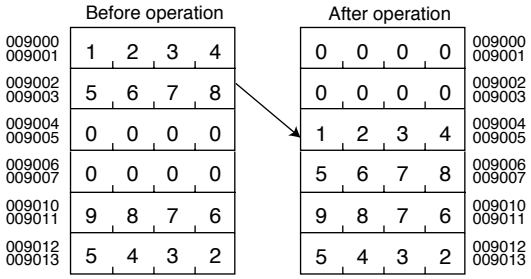
[Explanation]



Instruction	
S T R	000000
S T R	000001
F -61d	009004

000000 (1) ON ---------- Shift from 009000 to 009003.
000001 (2) ON ---------- Shift direction
Contents of 009004 to 009007 are 00000000(H).

The following result will be produced for the above input condition.



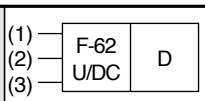
Only the carry flag (007356) is set.

Resembled instructions: F-61, F-61w, F-161, F-161w, F-161d

F-62
U/DC

2-digit BCD up/down counter (Up/Down Counter)

Symbol



(1) Up/down direction input
(2) Counter input
(3)Reset input

Function

The contents of the register D (BCD 2 digits) are added ((1) ON) or subtracted ((1) OFF), according to the up/down counter direction input.

Operation

- When the up/down direction input (1) is ON: 〈D〉+1→D
- When the up/down direction input (1) is OFF: 〈D〉−1→D

D

Use range A

Condition

At a rising of the counter input (2) (OFF to ON) when the reset input (3) is OFF

Contents after operation

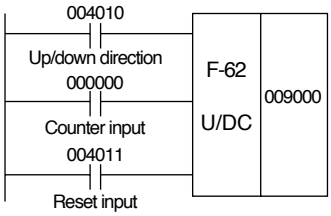
D

- Result (BCD code) is contained when the reset input (3) is OFF.
- Result (BCD code) is contained when the reset input (3) is OFF.

Flag

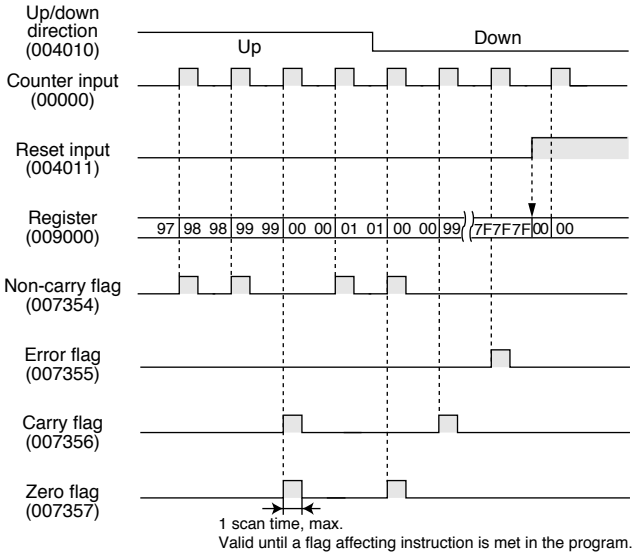
Up/down direction input (1)	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
ON	99+1 →00	1	1	0	0
	00 to 98+1 →01 to 99	0	0	0	1
	Not BCD code	0	0	1	0
OFF	00−1 →99	0	1	0	0
	01−1 →00	1	0	0	1
	02 to 99−1 →01 to 98	0	0	0	1
	Not BCD code	0	0	1	0
Reset input (3) ON		0	0	0	0

[Explanation]



Instruction	
S T R	004010
S T R	000000
S T R	004011
F -62	009000

When the reset input 0004011 is OFF, the counter is enabled, if set to ON reset mode. When the up/down counter direction input 004010 is ON, it acts as an increment counter. When OFF, it acts as a decrement counter. If the contents of 009000 were other than the BCD code, the error flag (007355) is turned ON and no counter operation takes place. (7F in the example.)



- If the contents of D is number other than BCD code, the error flag (007355) is set ON and no addition will be done. (D remain unchanged.)
- By setting the reset condition in the system memory (#0202), it is possible to reset the reset input (3) with OFF.

Resembled instructions: F-62w, F-62d, F-65, F-65w, F-65d, F-66, F-66w

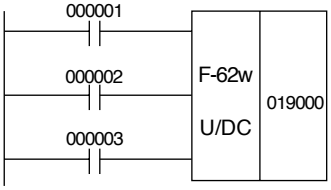
F-62w U/DC

4-digit BCD up/down counter (Up/Down Counter)

Symbol	(1) F-62w U DC D (2) (3) (1) Up/down direction input (2) Counter input (3) Reset input
Function	The 4-digit BCD contents of the register D, D+1 are added ((1) ON) or subtracted((1) OFF), according to the up/down counter direction input (1).
Operation	- When the up/down direction input (1) is ON: 〈D, D+1〉+1→D, D+1 - When the up/down direction input (1) is OFF: 〈D, D+1〉−1→D, D+1
D	Use range B - Be sure to use even addresses for register D.
Condition	At a rising of the counter input (2) (OFF to ON) when the reset input (3) is OFF

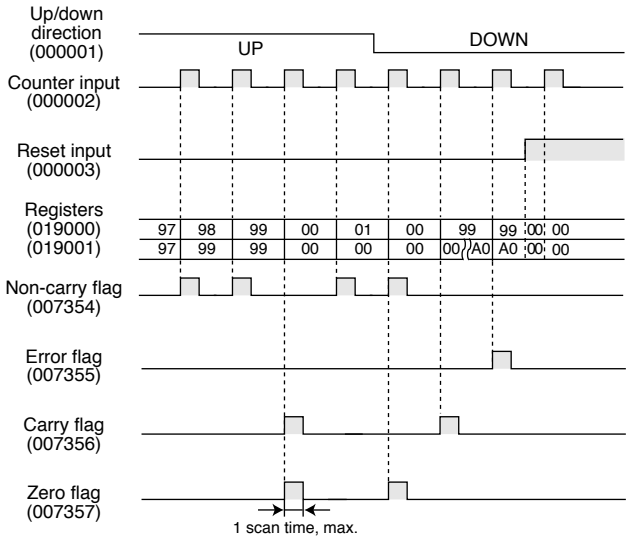
Contents after operation	D	Lower 2 digits of result		All bits are OFF when the reset input (3) is ON.				
	D+1	Upper 2 digits of result						
	Flag	Up/down direction input (1)	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
		ON	9999+1	1	1	0	0	
			0000 to 9998+1	0	0	0	1	
			Not BCD code	0	0	1	0	
		OFF	0000−1	0	1	0	0	
			0001−1	1	0	0	1	
			0002 to 9999−1	0	0	0	1	
			Not BCD code	0	0	1	0	
		Reset input (3) ON		0	0	0	0	

[Explanation]



Instruction	
S T R	000001
S T R	000002
S T R	000003
F -62w	019000

When the reset input 000003 is OFF, the counter becomes ready to count, if set to ON reset mode. When the up/down counter direction input 000001 is ON, it acts as an increment counter. When OFF, it acts as a decrement counter. If the contents of the register 019000 or 019001 are other than the BCD code, the error flag is turned ON and no counter operation takes place.



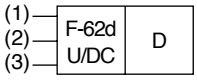
- By setting the reset condition in the system memory (#0202), it is possible to reset input (3) with OFF.

Resembled instructions: F-62, F-62d, F-65, F-65w, F-65d, F-66, F-66w

F-62d
U/DC

8-digit BCD up/down counter (Up/Down Counter)

Symbol



(1) Up/down direction input
(2) Counter input
(3) Reset input

Function

Increments (when input (1) is ON) or decrements (when input (1) is OFF) the contents of registers D to D+3 (BCD 8 digits) according to the up/down input (1) state.

Operation

- When the up/down direction input (1) is ON: 〈D to D+3〉+1 → D to D+3.
- When the up/down direction input (1) is OFF: 〈D to D+3〉−1 → D to D+3.

D

Use range C
- Be sure to use even addresses for register D.

Condition

Reset input (3) is OFF, and at the rising edge (OFF to ON) of the counter input (2).

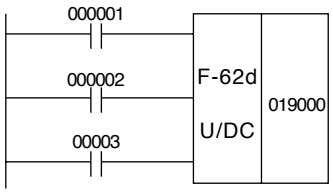
Contents after operation

D to D+3: Result (8 digits) — All bits turned OFF when reset input (3) is ON.

Flag

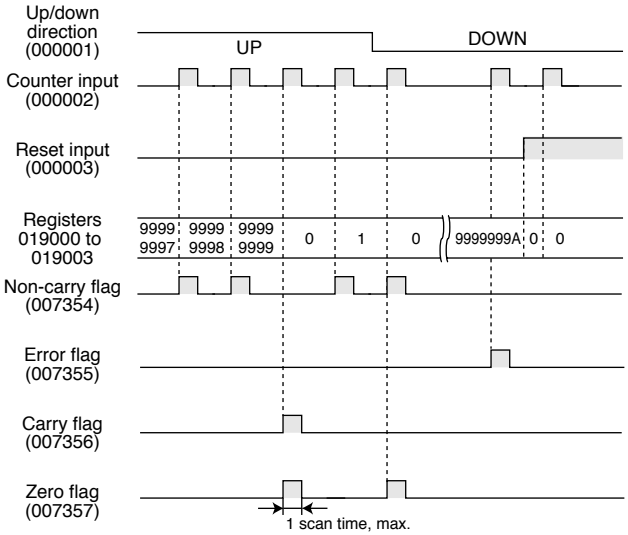
UP/DOWN input (1)	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
ON	99999999+1	1	1	0	0
	00000000 to 99999998+1	0	0	0	1
	Not BCD code	0	0	1	0
OFF	00000000-1	0	1	0	0
	00000001-1	1	0	0	1
	00000002 to 99999999-1	0	0	0	1
	Not BCD code	0	0	1	0
When reset input (3) is ON		0	0	0	0

[Explanation]



Instruction	
S T R	000001
S T R	000002
S T R	000003
F -62d	019000

When the reset input 000003 is OFF, the counter becomes ready to count, if set to ON reset mode. When the up/down counter direction input 000001 is ON, it acts as an increment counter. When OFF, it acts as a decrement counter. If the contents of the register 019000 to 019003 are other than the BCD code, the error flag is turned ON and no counter operation takes place.



- If the contents of registers D to D+3 are not BCD codes, the counter remains inoperative, and the error flag (007355) is set (the contents of D to D+3 are unaffected).
- By setting the reset condition in the system memory (#0202), it is possible to reset input (3) with OFF.

Resembled instructions: F-62, F-62w, F-65, F-65w, F-65d, F-66, F-66w

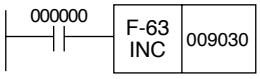
F-63 INC — Increments counter (1 byte) (INCrement)

Symbol	F-63 INC D					
Function	The contents of the register D (binary data) are incremented.					
Operation	〈D〉+1 → D					
D	Use range A					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	D	Result (binary)				
	Flag	Result (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		377 → 000	1	1	0	0
		Other than above	0	0	0	1

Resembled instructions: F-63, F-63w, F-163, F-163w, F-163d, F-263, F-263w, F-263d

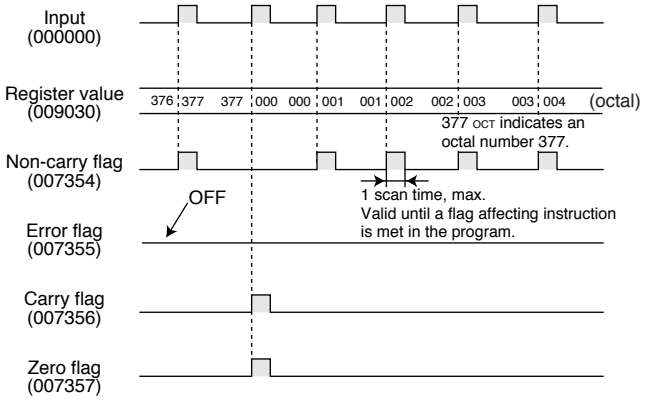
- The contents of D are represented by a binary number, which can assume a number of 000 to 255 in the decimal notation or 000 to 377(8) in the octal notation.

[Explanation]



Instruction	
S T R	000000
F-63	
	009030

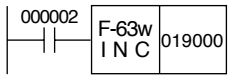
When the input condition 000000 changes from OFF to ON, the JW300 counts up the content of register 009030.



F-63w INC — Increments counter (1 word) (INCrement)

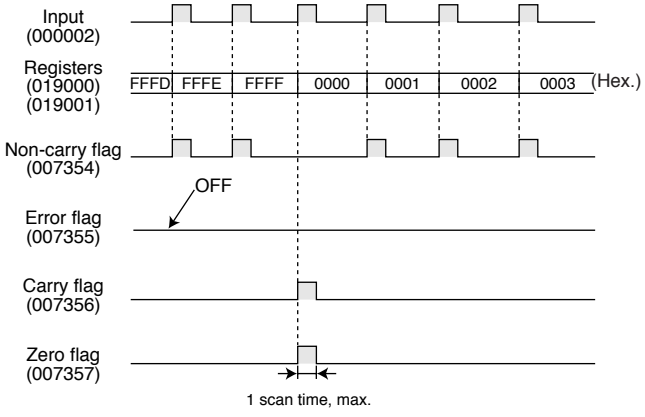
Symbol	F-63w INC D					
Function	The binary contents of the registers D and D+1 are incremented.					
Operation	〈D, D+1〉+1→D, D+1					
D	Use range B - Be sure to use even addresses for register D. (Odd addresses such as 019003 etc. are prohibited to use.)					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	D	Lower digits of result				
	D+1	Upper digits of result				
	Flag	Result (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		177777 →000000	1	1	0	0
		Other than above	0	0	0	1

[Explanation]



Instruction	
S T R	000002
F-63w	
	019000

When the input condition 000002 changes from OFF to ON, the JW300 counts up the content of register 019000 and 019001.



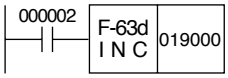
Resembled instructions: F-63, F-63d, F-163, F-163w, F-163d, F-263, F-263w, F-263d

F-63d INC — Increments counter (2 words) (INCrement)

Symbol		F-63d INC | D
Function		The binary contents of the registers D to D+3 are incremented.
Operation		〈D to D+3〉+1→D to D+3
D		Use range A - Be sure to use even addresses for register D. (Odd addresses such as 019003 etc. are prohibited to use.)
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	D to D+3	D : Lower digits of result D+3 : Upper digits of result

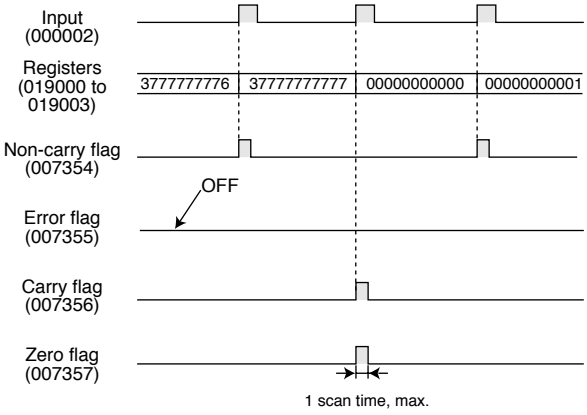
Contents after operation: Flag	Result (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	377777777777 →00000000000	1	1	0	0
	Other than above	0	0	0	1

[Explanation]



Instruction	
S T R	000002
F -63d	
	019000

When the input condition 000002 changes from OFF to ON, the JW300 counts up the content of register 019000 to 019003.

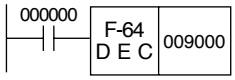


Resembled instructions: F-63, F-63w, F-163, F-163w, F-163d, F-263, F-263w, F-263d

F-64 DEC — Decrements counter (1 byte) (DECrement)

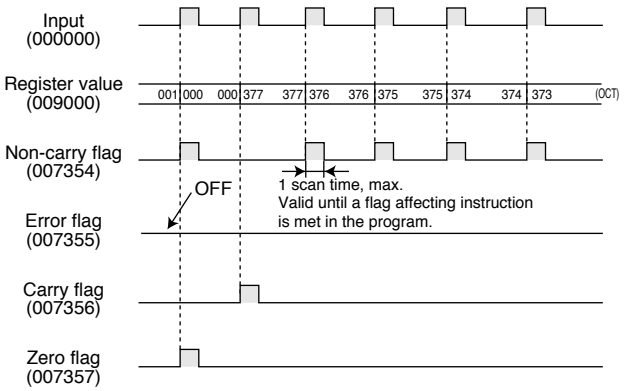
Symbol	F-64 DEC D					
Function	The contents of the register D (binary data) are decremented.					
Operation	〈D〉—1→D					
D	Use range A					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	D	Result (binary)				
	Flag	Result (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		001 → 000	1	0	0	1
		000 → 377	0	1	0	0
		Other than above	0	0	0	1

[Explanation]



Instruction	
S T R	000000
F -64	
	009000

When the input condition 000000 changes from OFF to ON, the JW300 counts down the content of register 009000.



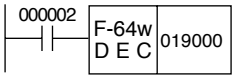
- The contents of D are represented by a binary number, which can assume a number of 000 to 255 in the decimal notation or 000 to 377(8) in the octal notation.

Resembled instructions: F-64w, F-64d, F-164, F-164w, F-164d, F-264, F-264w, F-264d

F-64w DEC — Decrements counter (1 word) (DECrement)

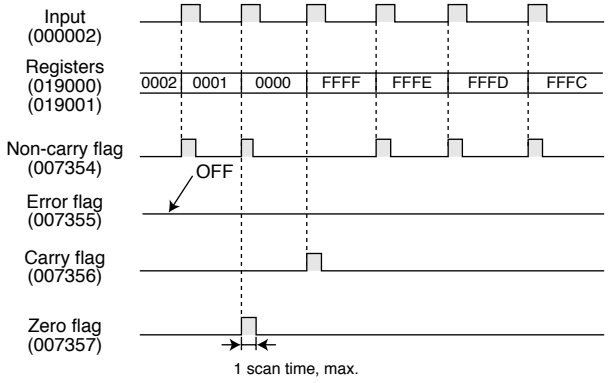
Symbol	F-64w DEC D					
Function	The binary contents of the registers D, D+1 are decremented.					
Operation	〈D, D+1〉—1→D, D+1					
D	Use range B *					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	D	Lower digits of result				
	D+1	Upper digits of result				
	Flag	Result (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		000001 →000000	1	0	0	1
		000000 →177777	0	1	0	0
		Other than above	0	0	0	1

[Explanation]



Instruction	
S T R	000002
F -64w	
	019000

When the input condition 000002 changes from OFF to ON, the JW300 counts down the content of register 019000 and 019001.



* Be sure to use even addresses for register D.
(Odd addresses such as 019003 etc. are prohibited to use.)

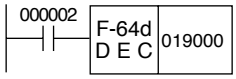
Resembled instructions: F-64, F-64d, F-164, F-164w, F-164d, F-264, F-264w, F-264d

F-64d DEC — Decrements counter (2 words) (DECrement)

Symbol		F-64d DEC | D
Function		The binary contents of the registers D to D+3 are decremented.
Operation		〈D to D+3〉−1→D to D+3
D		Use range C - Be sure to use even addresses for register D. (Odd addresses such as 019003 etc. are prohibited to use.)
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	D to D+3	D : Lower digits of result D+3 : Upper digits of result

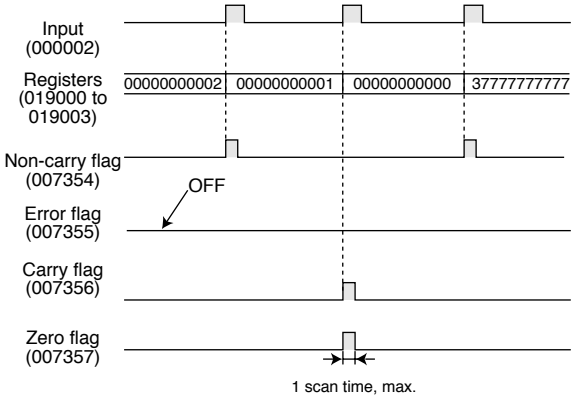
Flag: Result (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
00000000001 →00000000000	1	0	0	1
00000000000 →37777777777	0	1	0	0
Other than above	0	0	0	1

[Explanation]



Instruction	
S T R	000002
F -64d	
	019000

When the input condition 000002 changes from OFF to ON, the JW300 counts down the content of register 019000 to 019003.

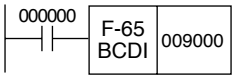


Resembled instructions: F-64, F-64w, F-164, F-164w, F-164d, F-264, F-264w, F-264d

F-65 BCDI — BCD increment counter (1 byte) (BCD Increment)

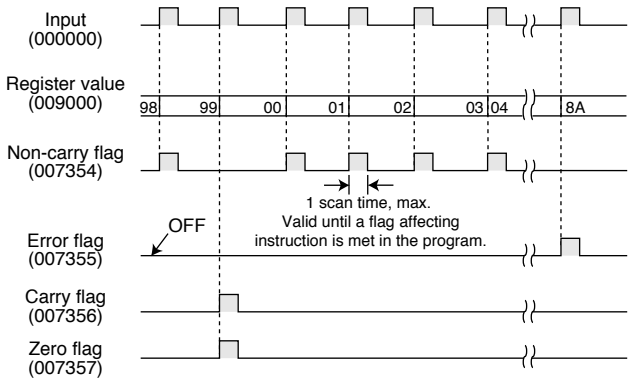
Symbol	F-65 BCDI D					
Function	Increment the contents (BCD) of register D by one.					
Operation	〈D〉+1→D					
D	Use range A					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	D	Result (BCD) - Unaffected if the contents of register D are not BCD code.				
	Flag	Result (BCD)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		99 to 00	1	1	0	0
		Other than above	0	0	0	1
		Not BCD code	0	0	1	0

[Explanation]



Instruction	
S T R	000000
F -65	
	009000

When the input condition 00000 changes from OFF to ON, the JW300 counts up (+1) the content of register 009000.



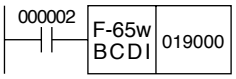
- If the contents of register D are not BCD code, the error flag (007355) is set ON and no division will be done. (D remains unchanged.)

Resembled instructions: F-62, F-62w, F-62d, F-65w, F-65d

F-65w BCDI — BCD increment counter (1 word) (BCD Increment)

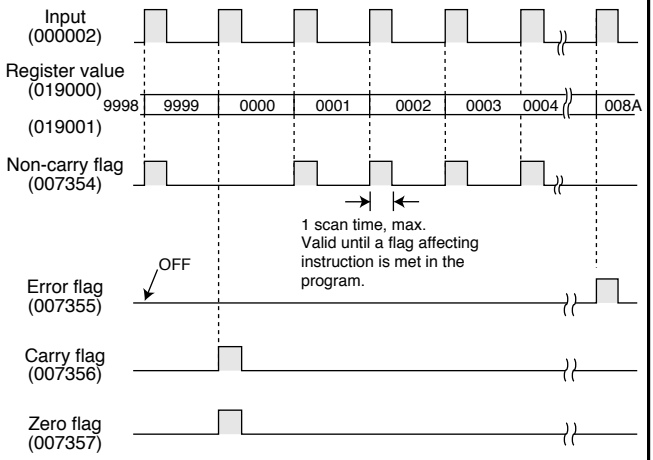
Symbol	F-65w BCDI D					
Function	Increment the contents (BCD) of registers D, D+1 by one.					
Operation	〈D, D+1〉+1→D, D+1					
D	Use range B - Be sure to use even addresses for register D. (Odd addresses such as 019003 etc. are prohibited to use.)					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	D	Lower digits of result	- Unaffected if the contents of registers D and D+1 are not BCD code.			
	D+1	Upper digits of result				
	Flag	Result (BCD)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		9999 →0000	1	1	0	0
		Other than above	0	0	0	1
		Not BCD code	0	0	1	0

[Explanation]



Instruction	
S T R	000002
F -65w	
	019000

When the input condition 000002 changes from OFF to ON, the JW300 counts up the content of register 019000 and 019001.



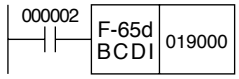
- If the contents of registers D and D+1 are not BCD code, the error flag (007355) is set ON and no division will be done. (D, D+1 remains unchanged.)

Resembled instructions: F-62, F-62w, F-62d, F-65, F-65w

F-65d BCDI — BCD increment counter (2 words) (BCD Increment)

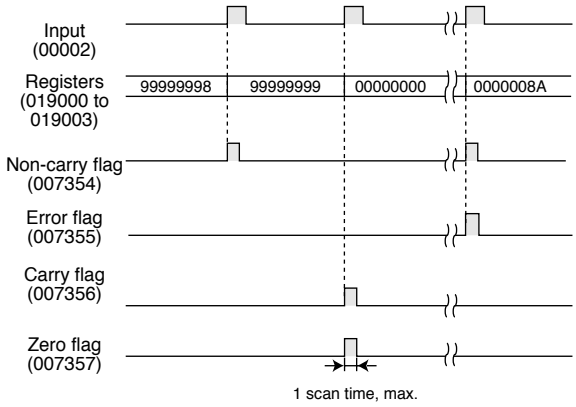
Symbol	F-65d BCDI | D					
Function	Increment the contents (BCD) of registers D to D+3 by one.					
Operation	〈D to D+3〉+1→D to D+3					
D	Use range C - Be sure to use even addresses for register D. (Odd addresses such as 019003 etc. are prohibited to use.)					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	D to D+3	D: Lower digits of result D+3: Upper digits of result *				
	Flag	Result (BCD)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		99999999 →00000000	1	1	0	0
		Other than above	0	0	0	1
		Not BCD code	0	0	1	0

[Explanation]



Instruction	
S T R	000002
F -65d	
	019000

When the input condition 000002 changes from OFF to ON, the JW300 counts up the content of register 019000 to 019003.



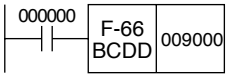
* If the contents of registers D to D+3 are not BCD code, the error flag (007355) is set ON and no division will be done. (D to D+3 remains unchanged.)

Resembled instructions: F-62, F-62w, F-62d, F-65, F-65w

F-66 BCDD — BCD decrement counter (1 byte) (BCD Decrement)

Symbol	F-66 BCDD D					
Function	Decrement the contents (BCD) of register D by one.					
Operation	〈D〉−1→D					
D	Use range A					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	D	Result (BCD code) - Unaffected if the contents of register D are not BCD code.				
	Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		01→00	1	0	0	1
		00→99	0	1	0	0
		Other than above	0	0	0	1
		Not BCD code	0	0	1	0

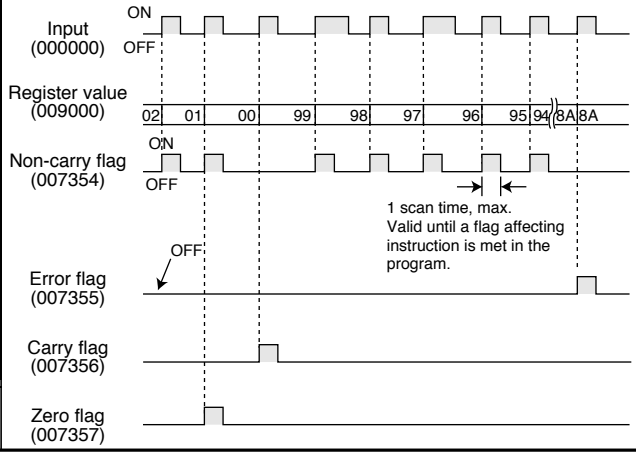
[Explanation]



Instruction

S T R 000000
F -66
009000

When the input condition 000000 changes from OFF to ON, the JW300 counts down (−1) the content of register 019000.



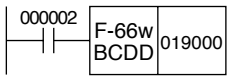
- If the contents of registers D are not BCD code, the error flag (007355) is set ON and no division will be done. (D remains unchanged.)

Resembled instructions: F-62, F-62w, F-62d, F-66w, F-66d

F-66w BCDD — BCD decrement counter (1 word) (BCD Decrement)

Symbol	F-66w BCDD D					
Function	Decrement the contents (BCD) of registers D, D+1 by one.					
Operation	〈D, D+1〉−1→D, D+1					
D	Use range B - Be sure to use even addresses for register D. (Odd addresses such as 019003 etc. are prohibited to use.)					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	D	Lower digits of result	- If the contents of registers D and D+1 are not BCD code, D and D+1 remains unchanged.			
	D+1	Upper digits of result				
	Flag	Result (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		0001 →0000	1	0	0	1
		0000 →9999	0	1	0	0
		Other than above	0	0	0	1
		Not BCD code	0	0	1	0

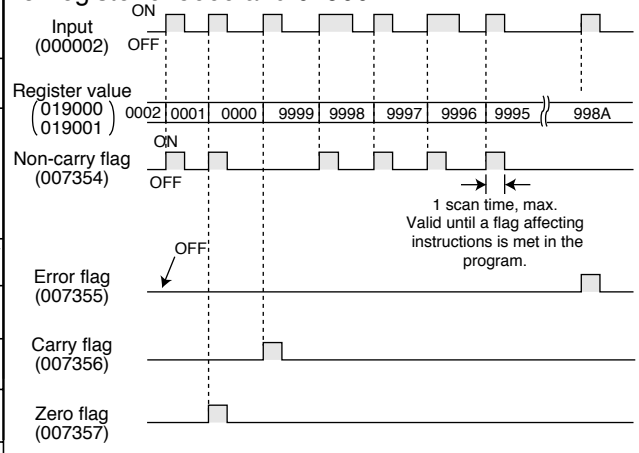
[Explanation]



Instruction

S T R 000002
F -66w
019000

When the input condition 000002 changes from OFF to ON, the JW300 counts down the content of register 019000 and 019001.



- If the contents of registers D and D+1 are not BCD code, the error flag (07355) is set ON and no division will be done. (D, D+1 remains unchanged.)

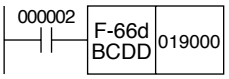
Resembled instructions: F-62, F-62w, F-62d, F-66, F-66d

F-66d BCDD — BCD decrement counter (2 words) (BCD Decrement)

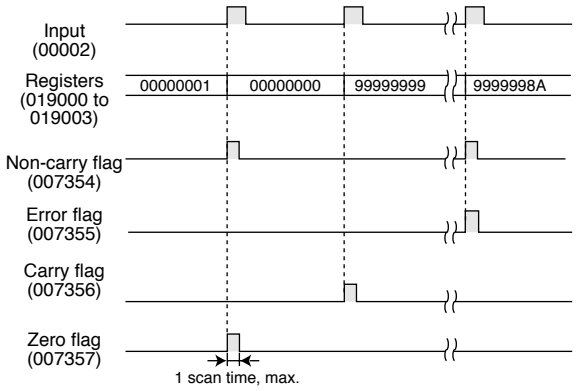
Symbol	F-66d BCDD | D					
Function	Decrement the contents (BCD) of registers D to D+3 by one.					
Operation	〈D to D+3〉−1→D to D+3					
D	Use range C - Be sure to use even addresses for register D. (Odd addresses such as 019003 etc. are prohibited to use.)					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	D to D+3	D: Lower digits of result D+3: Upper digits of result *				
	Flag	Result (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		00000001 →00000000	1	0	0	1
		00000000 →99999999	0	1	0	0
		Other than above	0	0	0	1
		Not BCD code	0	0	1	0

[Explanation]

Instruction	
S T R	000002
F -66d	
	019000



When the input condition 000002 changes from OFF to ON, the JW300 counts down the content of register 019000 to 019003.



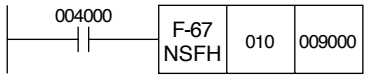
* If the contents of registers D to D+3 are not BCD code, the error flag (007355) is set ON and no division will be done. (D to D+3 remains unchanged.)

Resembled instructions: F-62, F-62w, F-62d, F-66, F-66w

F-67 NSFH — Digit shift (up)

Symbol	F-67 NSFH | n | D	
Function	Shifts the registers upper 4 bits for the "n" bytes starting with register D.	
Operation	Shifts D through D+n−1 up 4 bits each - After the operation, the PC stores 0 to the lower 4 bits of register D. After the shift operation, the upper 4 bits "n" bytes registers are cleared.	
n	000 to 377(8) (256 bytes for 000(8))	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D	Result (shift result)
	Flag	Unchanged

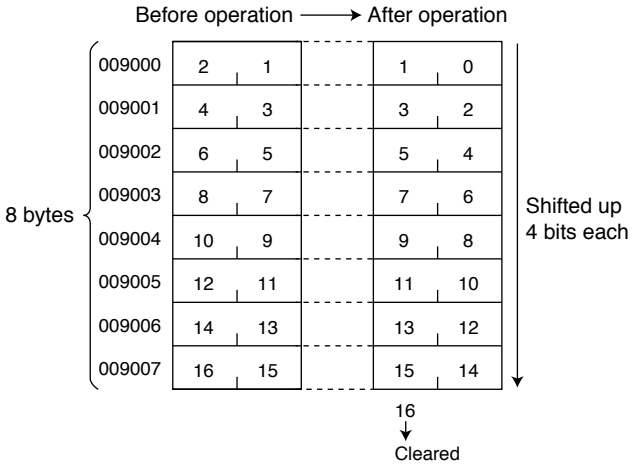
[Explanation]



Instruction

```
S T R   004000
F -67
          010
       009000
```

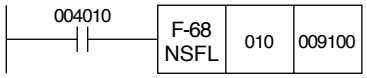
When input condition 004000 changes from OFF to ON, the 8 (010(8)) bytes of data in registers 009000 to 009007 are shifted upper 4 bits.



F-68 NSFL — Digit shift (down)

Symbol	F-68 NSFL | n | D	
Function	Shifts the registers to lower 4 bits for the "n" bytes starting with register D.	
Operation	Shifts D through D + n−1 down 4 bits each - After the operation, the PC stores 0 to the upper 4 bits of register D + n−1 which is at the beginning, store 0. After the shift operation, the lower 4 bits D registers are cleared.	
n	000 to 377(8) (256 bytes for 000(8))	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D	Result (shift result)
	Flag	Unchanged

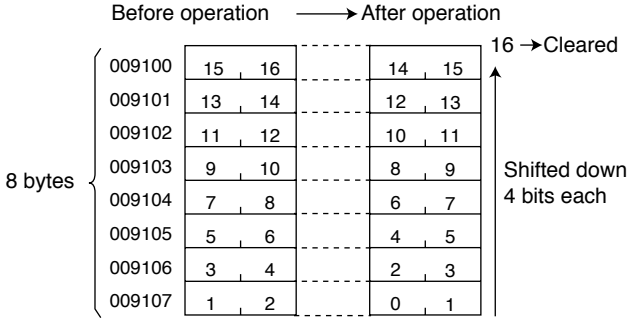
[Explanation]



Instruction

```
S T R   004010
F -68
          010
       009100
```

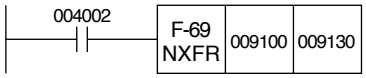
When input condition 004010 changes from OFF to ON, the 8 (010(8)) bytes of data in registers 009100 to 009107 are shifted to lower 4 bits.



F-69 NXFR Digit transfer

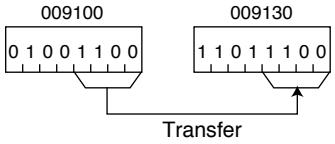
Symbol	F-69 NXFR | S | D	
Function	Transfers the lower 4 bits of register S to the lower 4 bits of register D.	
Operation	Lower 4 bits of S → Lower 4bits of D	
S	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Contents of lower 4 bits of register S ; the upper 4 bits remain unchanged
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004002
F -69	
	009100
	009130

When input condition 004002 changes from OFF to ON, the contents of the lower 4 bits of register 009100 are transferred to the lower 4 bits of register 009130.



F-70 FILE — Transfer "n" byte in batch (FILE)

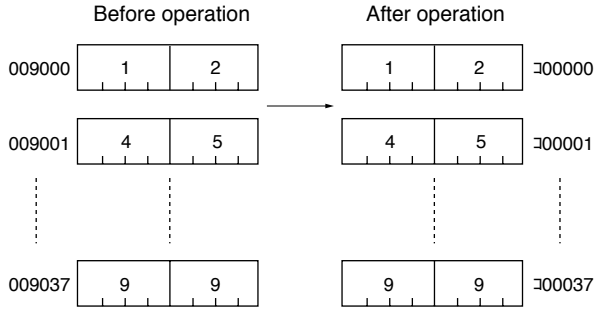
Symbol	F-70 FILE | n | S | D	
Function	Transfer data of S+n-1 data (n bytes) from register S to D+n-1 (n bytes) of register D.	
Operation	S, S+1, ·····, S+n−2, S+n−1 →D, D+1, ·····, D+n−2, D+n−1	
n	Use range 000 to 377(8) (256 bytes for 000(8))	
S	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+n−1	Unchanged
	D	Contents of the register S
	D+1	Contents of the register S+1
	⋮	⋮
	D+n−2	Contents of the register S+n−2
	D+n−1	Contents of the register S+n−1
	Flag	Unchanged

[Explanation]

Instruction	
S T R	004001
F -70	
	040
	009000
	⊐00000

004001 | F-70 FILE | 040 | 009000 | ⊐00000

When the input condition 004001 changes from OFF to ON, 040(8) -byte data (32 bytes in decimal) in register 009000 through 009037 are transferred in batch to the 32-byte area of ⊐00000 through ⊐00037.
The contents of registers, 009000 through 009037, remain unaffected.



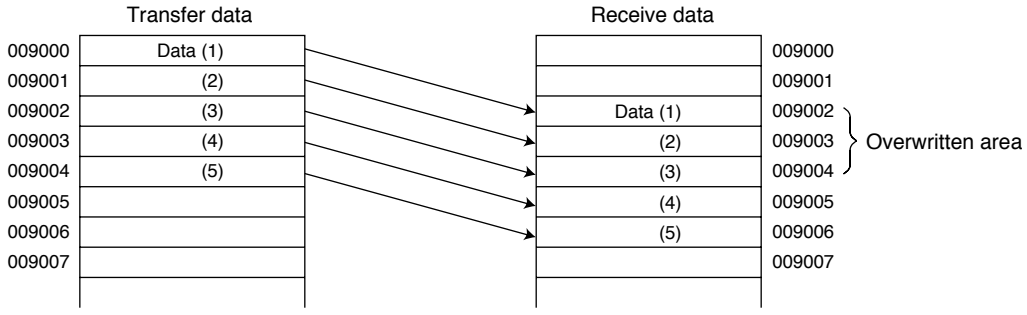
- Do not transfer the contact point area (file address 00001600 to 00001777 etc.) of the timer/counter of file number 0.

F-70 FILE | 5 | 019000 | ⊐01575

If programmed in the above manner, the contents of 019003 and 019004 will be transferred to 00001600 and 00001601.

- Setting n, S, and D which may overwrite the destination, is available.

004002 | F-70 FILE | 005 | 009000 | 009002



Resembled instructions: F-00, F-00w, F-00d, F-70w, F-70d, F-74, F-74w, F-74d, F-76, F-76w, F-76d

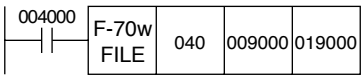
F-70w FILE

Transfers "n" words in batch (FILE)

Symbol		F-70w FILE | n | S | D
Function		Transfer data of S+2n−1 data (n words) from register D to D+2n-1 (n words) of register D.
Operation		S, S+1 ·····, S+2n−2, S+2n−1 →D, D+1, ·····, D+2n−2, D+2n−1
n		Use range 000 to 377(8) (256 words for 000(8))
S		Use range B - Be sure to use even addresses for registers S.
D		Use range B - Be sure to use even addresses for registers D.
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S, ···, S+2n−1	Unchanged
	D D+1 ⋮ D+2n−2 D+2n−1	Contents of the register S Contents of the register S+1 ⋮ Contents of the register S+2n−2 Contents of the register S+2n−1
	Flag	Unchanged

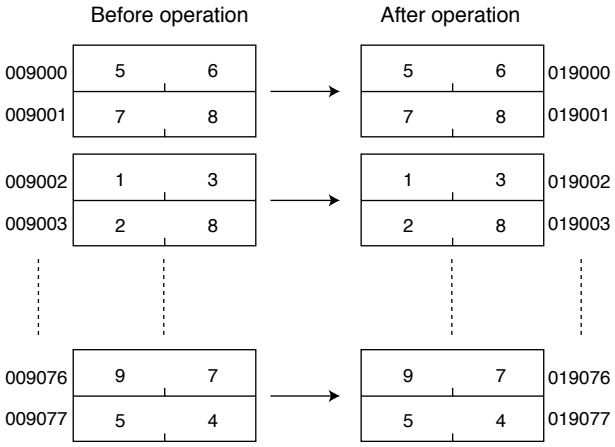
Resembled instructions: F-00, F-00w, F-00d, F-70, F-70d, F-74, F-74w, F-74d, F-76, F-76w, F-76d

[Explanation]



Instruction	
S T R	004000
F -70w	
	040
	009000
	019000

When the input condition 004000 changes from OFF to ON, 040(8)-words data (32 words in decimal) in registers 009000 to 009077 are transferred in batch to the 32-words area of 019000 to 019077.
The contents of registers, 009000 to 009077, remain unaffected.



F-70d FILE — Transfers "n" double words in batch (FILE)

Symbol	F-70d FILE | n | S | D	
Function	Transfer data of S to S+4n-1 data (n-double words) from register D to D+4n-1 (n-double words) of register D.	
Operation	S to S+3, ·····, S+4n−4 to S+4n−1 →D to D+3, ·····, D+4n−4 to D+4n−1	
n	Use range 000 to 377(8) (256 double words for 000(8))	
S	Use range C *1	
D	Use range C *2	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, ···, S+4n−1	Unchanged
	D to D+3 D+4 to D+7 ⋮ D+4n−4 to D+4n−1	Contents of the register S to S+3 Contents of the register S+4 to S+7 ⋮ Contents of the register S+4n−4 to S+4n−1
	Flag	Unchanged

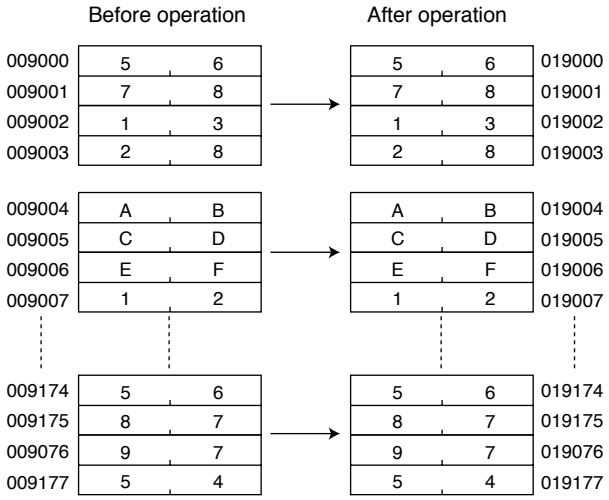
*1 Be sure to use even addresses for register S.
*2 Be sure to use even addresses for register D.
Resembled instructions: F-00, F-00w, F-00d, F-70, F-70w, F-74, F-74w, F-74d, F-76, F-76w, F-76d

[Explanation]

Instruction	
S T R	004000
F -70d	
	040
	009000
	019000

004000 — F-70d FILE | 040 | 009000 | 019000

When the input condition 004000 changes from OFF to ON, 040(8)-double words data (32-double words in decimal) in registers 009000 to 009077 are transferred in batch to the 32-double words area of 019000 to 019177.
The contents of registers, 009000 to 009177, remain unaffected.



F-71 CONS — Transfers octal constant in batch (1 byte) (CONStant)

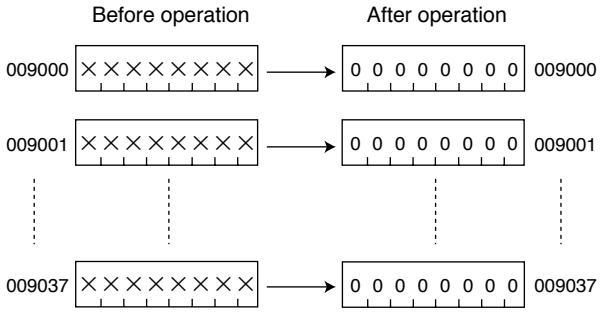
Symbol	F-71 CONS | n | D_1 | D_2	
Function	An octal constant "n" is transferred in batch from the register D_1 to the register D_2.	
Operation	$n \rightarrow D_1, \cdots\cdots, D_2$	
n	Use range 000 to $377_{(8)}$	
D_1	Use range A	
D_2	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D_1 D_1+1 ⋮ D_2-1 D_2	Constant "n"
	Flag	Unchanged

[Explanation]

Instruction	
S T R	004001
F -71	000
	009000
	009037

004001 | F-71 CONS | 000 | 009000 | 009037

When the input condition 004001 changes from OFF to ON, the octal constant 000 is transferred in batch from registers 009000 to 009037.



- No operation takes place if D_1 or D_2 is programmed that may override (1) to (9), as shown below.

Range		JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
Relay	(1)	ㅋ00000 to ㅋ01577					
	(2)	ㅋ02000 to ㅋ07577					
	(3)		ㅋ10000 to ㅋ15377	ㅋ10000 to ㅋ54377			
Value of TMR, CNT, MD	(4)	b00000 to b01777					
	(5)	b02000 to b03777					
	(6)		b04000 to b07777	b04000 to b37777			
Register	(7)	009000 to E07777					
	(8)	109000 to Z377					
FILE 1	(9)		00000000 to 00077777	00000000 to 00377777	00000000 to 01777777	00000000 to 07777777	00000000 to 37777777

- No operation takes place if the address D_1 is greater than D_2.

F-71 CONS | 050 | 009200 | 009000 $D_1(009200) > D_2(009000)$

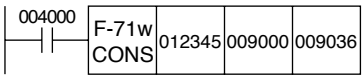
Resembled instructions: F-08, F-08w, F-08d, F-71w, F-71d

F-71w CONS — Transfers octal constant in batch (1 word) (CONStant)

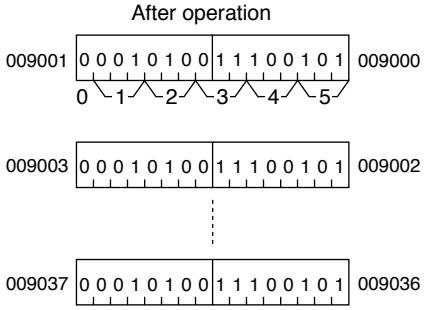
Symbol	F-71w CONS | n | D_1 | D_2	
Function	An octal constant "n" is transferred in batch in registers D_1, D_1+1, to D_2, D_2+1.	
Operation	n→ (D_1, D_1+1), ····· (D_2, D_2+1)	
n	Use range 000000 to 177777(8)	
D_1	Use range B *1	
D_2	Use range B *2	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D_1, D_1+1 D_1+2, D_1+3 ⋮ D_2−2, D_2−1 D_2, D_2+1	Constant "n"
	Flag	Unchanged

[Explanation]

Instruction
S T R 004000
F -71w
012345
009000
009036



When the input condition 004000 changes from OFF to ON, the octal constant 012345 is transferred in batch from registers 009000, 009001 to 009036, 009037.



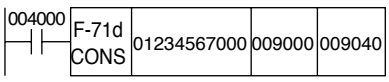
*1 Be sure to use even addresses for register D_1.
*2 Be sure to use even addresses for register D_2.
Resembled instructions: F-08, F-08w, F-08d, F-71, F-71d

F-71d CONS — Transfers octal constant in batch (2 words) (CONStant)

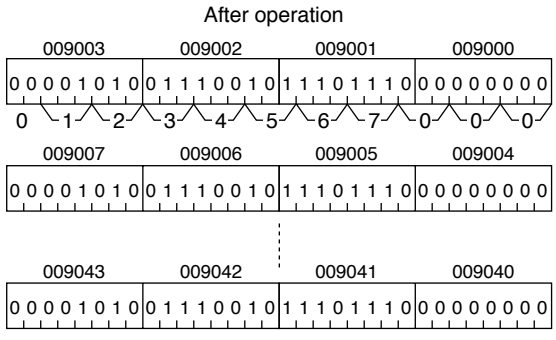
Symbol	F-71d CONS | n | D_1 | D_2	
Function	An octal constant "n" is transferred in batch in registers D_1 to D_1+3, to D_2 to D_2+3.	
Operation	n→ (D_1 to D_1+3), ····· (D_2 to D_2+3)	
n	Use range 00000000000 to 3777777777(8)	
D_1	Use range C *1	
D_2	Use range C *2	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D_1 to D_1+3 D_1+4 to D_1+7 ⋮ D_2−4 to D_2−1 D_2 to D_2+3	Constant "n"
	Flag	Unchanged

[Explanation]

Instruction
S T R 004000
F -71d
01234567000
009000
009040



When the input condition 004000 changes from OFF to ON, the octal constant 012345 is transferred in batch from registers 009000, 009001 to 009036, 009037.



*1 Be sure to use even addresses for register D_1.
*2 Be sure to use even addresses for register D_2.
Resembled instructions: F-08, F-08w, F-08d, F-71, F-71w

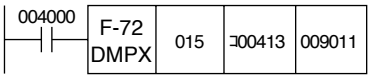
F-72 DMPX — Demultiplex "n" bytes to file 1 register (DeMultiPleXer)

Symbol	F-72 DMPX | n | S | D	
Function	The contents of an "n" byte register group heralded by the register which is modified by the contents of S (data pointer) from the register S contained data memory block top address (reference address) are transferred to the file 1 register which address is determined by the contents of the registers D to D+2.	
Operation	X+⟨S⟩, ···, X+⟨S⟩+n−1 →⟨D to D+2⟩, ···, ⟨D to D+2⟩+n−1 X: Top address of the data memory block in which S is contained (reference address) ⟨S⟩: Data pointer	
n	Use range 000 to 377(8) (256 bytes for 000(8))	
S	Use range A	
D	Use range E	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	Other than file 0 Register	Unchanged
	⟨D to D+2⟩ ⟨D to D+2⟩+1 ⋮ ⟨D to D+2⟩+n−1	Contents of register X+⟨S⟩ Contents of register X+⟨S⟩+1 ⋮ Contents of register X+⟨S⟩+n−1
	Flag	Unchanged

Resembled instructions: F-05, F-05w, F-05d, F-72w, F-72d

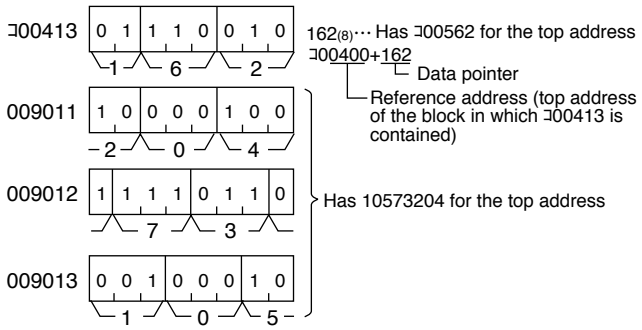
- The reference address of the register side is ⊐00000, ⊐00400, ⊐01000···019000, 019400 which is the top address of the data memory block in which S is contained. => "Data memory block and reference address."

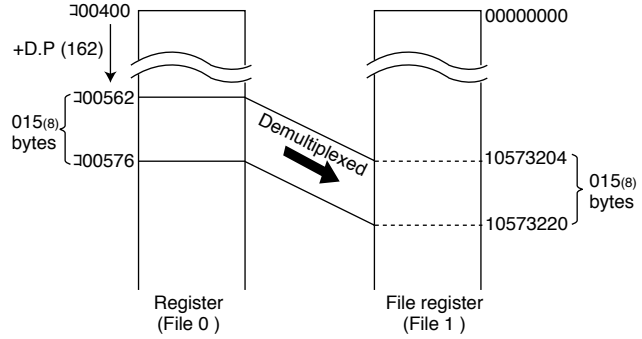
[Explanation]



Instruction	
S T R	004000
F -72	
	015
	⊐00413
	009011

When the input condition 004000 changes from OFF to ON, the contents of a 015(8) bytes register group heralded by the register which is modified by the contents of the register ⊐00413 (data pointer) from ⊐00400 (top address of the register ⊐00413 contained data memory block···reference address) are transferred to a 015(8) register group heralded by the file 1 register which address is determined by the contents of the registers 009011 and 009012.





F-72w DMPX — Demultiplexes "n" words to file 1 register (DeMultiPleXer)

Symbol	F-72w DMPX | n | S | D	
Function	The contents of an "n"-word register group which address is determined by the reference address (the top address of the register S included data memory block) modified by the contents of S (data pointer) are transferred to the file 1 register whose address implied by the contents of the registers D to D+2.	
Operation	X+⟨S⟩, ···, X+⟨S⟩+2n−1 →⟨D to D+2⟩, ···, ⟨D to D+2⟩+2n−1 X: Top address of the data memory block in which S is contained (reference address) ⟨S⟩: Data pointer	
n	Use range 000 to 377(8) (256 words for 000(8))	
S	Use range B *1	
D	Use range C *2	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	Other than file 0 register	Unchanged
	⟨D to D+2⟩	Contents of register X+⟨S⟩
	⟨D to D+2⟩+1	Contents of register X+⟨S⟩+1
	⋮	⋮
	⟨D to D+2⟩+2n−1	Contents of register X+⟨S⟩+2n−1
	Flag	Unchanged

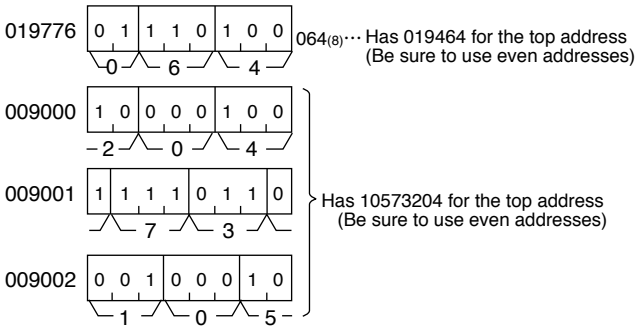
*1 Be sure to use even addresses for register S.
*2 Be sure to use even addresses for register D.
Resembled instructions: F-05, F-05w, F-05d, F-72, F-72d

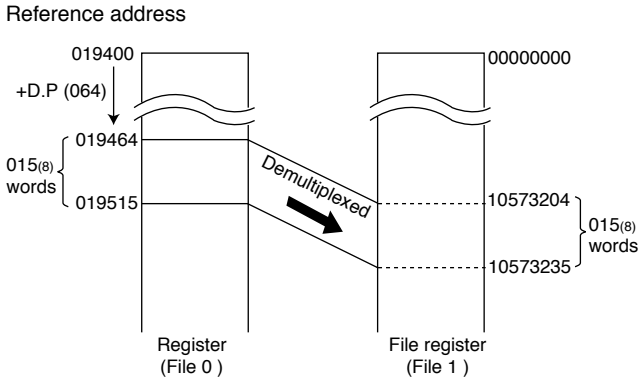
[Explanation]

Instruction	
S T R	004000
F -72w	
	015
	019776
	009000

004000 — F-72w DMPX | 015 | 019776 | 009000

When the input condition 004000 changes from OFF to ON, the contents of a 015(8)-words register group heralded by the register which is modified by the contents of the register 019776 (data pointer) from 019400 (the top address of the data block in which contained the register 019776 ···reference address) are transferred to a 015(8)-words register group heralded by the file 1 register of which address is determined by the contents of the registers 009000 to 009002.





Reference

Operation similar to the following instructions will take place for the above.

004000 — F-101 SEGM | 10573204 | 1 | 009000

F-70w FILE | 015 | 019464 | @009000

- The reference address of the register side is ⊐00000, ⊐00400, ⊐01000···019000, 019400 which is the top address of the data memory block in which S is contained. => "Data memory block and reference address."

F-72d DMPX — Demultiplexes "n" double words to file 1 register (DeMultiPleXer)

Symbol	F-72d DMPX | n | S | D	
Function	The contents of an "n"-double words register group which address is determined by the reference address (the top address of the register S included data memory block) modified by the contents of S (data pointer) are transferred to the file 1 register whose address implied by the contents of the registers D to D+2.	
Operation	X+⟨S⟩, ···, X+⟨S⟩+4n−1 →⟨D to D+2⟩, ···, ⟨D to D+2⟩+4n−1 X: Top address of the data memory block in which S is contained (reference address) ⟨S⟩: Data pointer	
n	Use range 000 to 377(8) (256 double words for 000(8))	
S	Use range B *1	
D	Use range C *2	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	Other than file 0 register	Unchanged
	⟨D to D+2⟩	Contents of register X+⟨S⟩
	⟨D to D+2⟩+1	Contents of register X+⟨S⟩+1
	⋮	⋮
	⟨D to D+2⟩+4n−1	Contents of register X+⟨S⟩+4n−1
	Flag	Unchanged

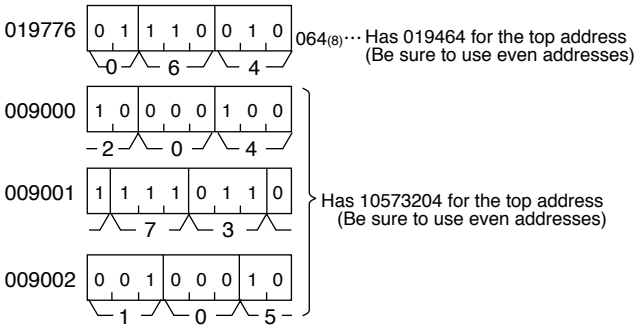
*1 Be sure to use even addresses for register S.
*2 Be sure to use even addresses for register D.
Resembled instructions: F-05, F-05w, F-05d, F-72, F-72w

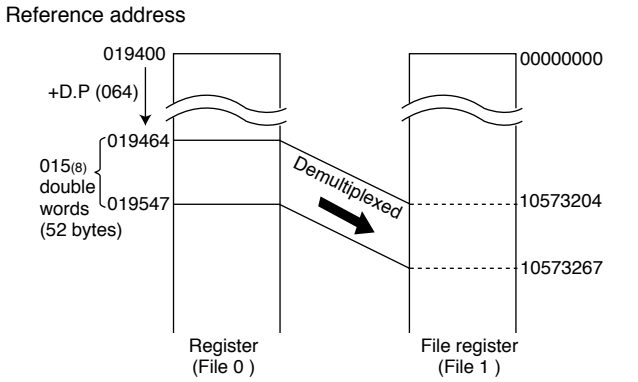
[Explanation]

Instruction	
S T R	004000
F -72d	
	015
	019776
	009000

004000 — F-72d DMPX | 015 | 019776 | 009000

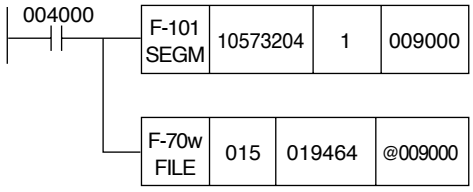
When the input condition 004000 changes from OFF to ON, the contents of a 015(8)-double words register group heralded by the register which is modified by the contents of the register 019776 (data pointer) from 019400 (the top address of the data block in which contained the register 019776 ···reference address) are transferred to a 015(8)-double words register group heralded by the file 1 register of which address is determined by the contents of the registers 009000 to 009002.





Reference

Operation similar to the following instructions will take place for the above.



- The reference address of the register side is ⊐00000, ⊐00400, ⊐01000···019000, 019400 which is the top address of the data memory block in which S is contained. => "Data memory block and reference address."

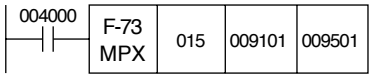
F-73 MPX — Multiplex "n" byte from file 1 register (MultiPleXer)

Symbol	F-73 MPX | n | S | D	
Function	Transfer data with the following conditions. Source: n bytes data starting from a file register specified by register S to S+2. Destination: n bytes from a register displaced by D counted from the top address of the data memory block having register D.	
Operation	〈S to S+2〉, ··· , 〈S to S+2〉+n−1 →X+〈D〉, ··· , X+〈D〉+n−1 X: Top address of the data memory in which D is contained (reference address) 〈D〉: Data pointer	
n	Use range 000 to 377(8) (256 bytes for 000)	
S	Use range E	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	File 1 register	Unchanged
	X+〈D〉 X+〈D〉+1 ⋮ X+〈D〉+n−1	Contents of file 1 register 〈S to S+2〉 Contents of file 1 register 〈S to S+2〉+1 ⋮ Contents of file 1 register 〈S to S+2〉+n−1
	Flag	Unchanged

Resembled instructions: F-06, F-06w, F-06d, F-73w, F-73d

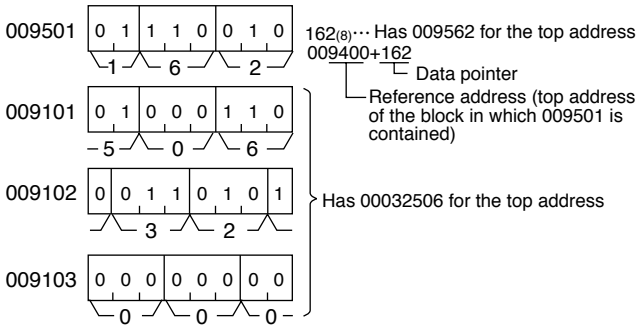
- The reference address of the register side is ⊐00000, ⊐00400, ⊐01000···019000, 019400 which is the top address of the data memory block in which S is contained. => "Data memory block and reference address."

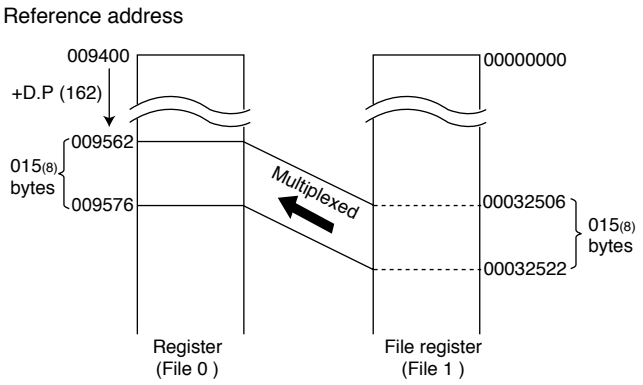
[Explanation]



Instruction	
S T R	004000
F -73	
	015
	009101
	009501

When the input condition 004000 changes from OFF to ON, the contents of a 015(8) bytes register group headed by the file 1 register of which address is determined by the contents of the registers 009101and 009103 are transferred to 015(8) bytes register group headed by the register which is modified by the contents of 009501 (data pointer) from 009400 (top address of the 009501 contained data memory block···reference address).





F-73w MPX — Multiplexes "n" words from file 1 register (MultiPleXer)

Symbol	F-73w MPX | n | S | D	
Function	Transfer data with the following conditions. Source: n bytes data starting from a file register specified by register S to S+2. Destination: n words from a register displaced by D counted from the top address of the data memory block having register D.	
Operation	〈S to S+2〉, ···〈S to S+2〉+2n−1 →X+〈D〉, ···X+〈D〉+2n−1 X: Top address of the data memory in which D is contained (reference address) 〈D〉: Data pointer	
n	Use range 000 to 377(8) (256 bytes for 000(8))	
S	Use range C *1	
D	Use range B *2	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	File 1 register	Unchanged
	X+〈D〉 X+〈D〉+1 ⋮ X+〈D〉+2n−1	Contents of file 1 register 〈S to S+2〉 Contents of file 1 register 〈S to S+2〉+1 ⋮ Contents of file 1 register 〈S to S+2〉+ 2n-1
	Flag	Unchanged

Resembled instructions: F-06, F-06w, F-06d, F-73, F-73d

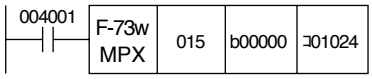
*1 Be sure to use even addresses for register S.

*2 Be sure to use even addresses for register D.

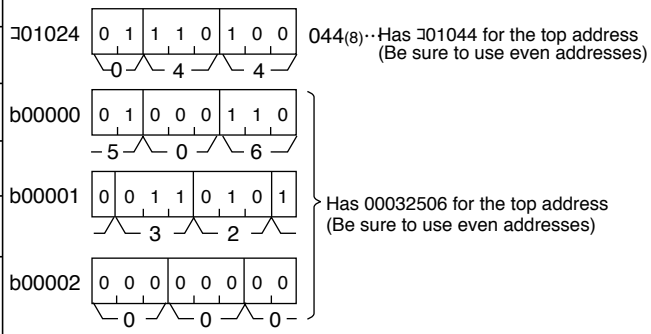
- The reference address of the register side is ⊐00000, ⊐00400, ⊐01000···019000, 019400 which is the top address of the data memory block in which S is contained. => "Data memory block and reference address."

[Explanation]

Instruction	
S T R	004001
F -73w	
	015
	b00000
	⊐01024



When the input condition 004001 changes from OFF to ON, the contents of a 015(8)-words register group headed by the file 1 register of which address is determined by the contents of the registers b00000 to b00002 are transferred to an "n"- words register group headed by the register which is modified by the contents of the register ⊐01024 (data pointer) from ⊐01000 (top address of the data memory block in which contained the register ⊐01024･･･reference address).



F-73d MPX — Multiplexes "n" double words from file 1 register (MultiPleXer)

Symbol	F-73d MPX | n | S | D	
Function	Transfer data with the following conditions. Source: n double words data starting from a file register specified by register S to S+2. Destination: n double words from a register displaced by D counted from the top address of the data memory block having register D.	
Operation	⟨S to S+2⟩, ···⟨S to S+2⟩+4n−1 →X+⟨D⟩, ···X+⟨D⟩+4n−1 X: Top address of the data memory in which D is contained (reference address) ⟨D⟩: Data pointer	
n	Use range 000 to 377(8) (256 double words for 000(8))	
S	Use range C *1	
D	Use range B *2	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	File 1 register	Unchanged
	X+⟨D⟩ X+⟨D⟩+1 ⋮ X+⟨D⟩ +4n−1	Contents of file 1 register ⟨S to S+2⟩ Contents of file 1 register ⟨S to S+2⟩+1 ⋮ Contents of file 1 register ⟨S to S+2⟩+ 4n-1
	Flag	Unchanged

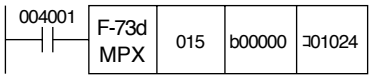
Resembled instructions: F-06, F-06w, F-06d, F-73, F-73w

*1 Be sure to use even addresses for register S.

*2 Be sure to use even addresses for register D.

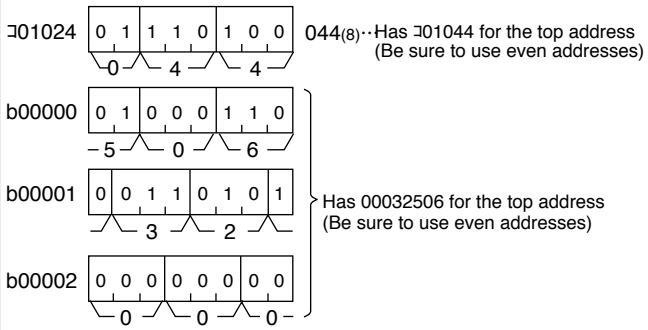
- The reference address of the register side is ⊐00000, ⊐00400, ⊐01000···019000, 019400 which is the top address of the data memory block in which S is contained. => "Data memory block and reference address."

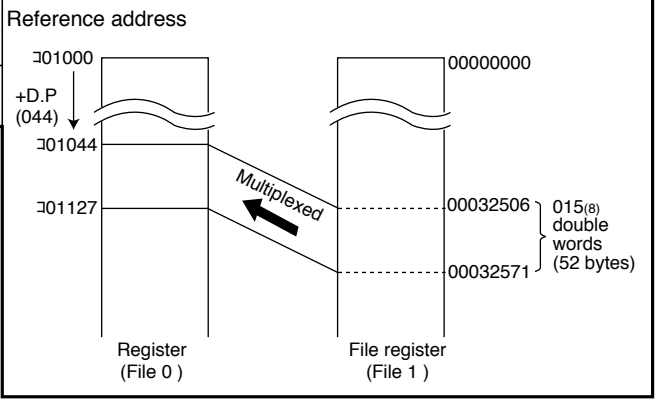
[Explanation]



Instruction	
S T R	004001
F -73d	
	015
	b00000
	⊐01024

When the input condition 004001 changes from OFF to ON, the contents of a 015(8)-double words register group headed by the file 1 register of which address is determined by the contents of the registers b00000 to b00002 are transferred to a 015(8)- double words register group headed by the register which is modified by the contents of the register ⊐01024 (data pointer) from ⊐01000 (top address of the data memory block in which contained the register ⊐01024・・・reference address).



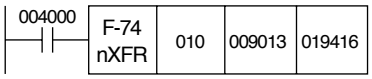


F-74 nXFR — Transfers "n" bytes

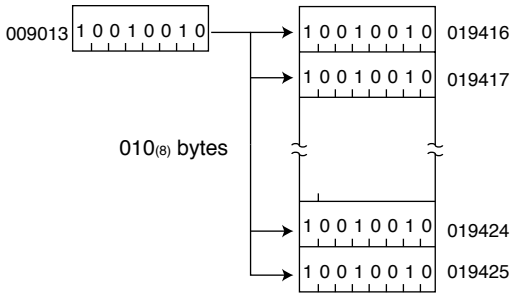
Symbol	F-74 nXFR | n | S | D	
Function	The contents of the register S are transferred to "n" bytes registers headed by the register D.	
Operation	S→D, D+1,···, D+n−1	
n	Use range 000 to 377(8) (256 bytes for 000)	
S	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D D+1 ⋮ D+n−1	Contents of the register S
	Flag	Unchanged

[Explanation]

Instruction	
S T R	004000
F-74	
	010
	009013
	019416



When the input condition 004000 changes from OFF to ON, the contents of the register 009013 are transferred to a 010(8) bytes registers headed by the register 019416.



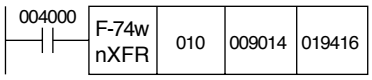
Resembled instructions: F-00, F-00w, F-00d, F-70, F-70w, F-70d, F-74w, F-74d, F-76, F-76w, F-76d

F-74w nXFR — Transfers "n" words

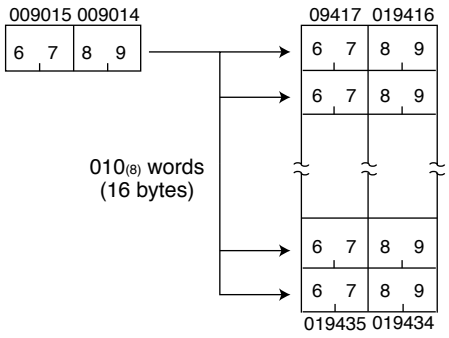
Symbol	F-74w nXFR | n | S | D	
Function	The contents of the registers S, S+1 are transferred to "n" words registers headed by the registers D, D+1.	
Operation	S, S+1→D, D+1···D+2n−2, D+2n−1	
n	Use range 000 to 377(8) (256 words for 000)	
S	Use range B *	
D	Use range B *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1	Unchanged
	D D+1 ⋮ D+2n−2 D+2n−1	Contents of the register S Contents of the register S+1 ⋮ Contents of the register S Contents of the register S+1
	Flag	Unchanged

[Explanation]

Instruction	
S T R	004000
F-74w	
	010
	009014
	019416



When the input condition 004000 changes from OFF to ON, the contents of the registers 009014 and 009015 are transferred to a 010(8) words registers headed by the registers 019416 and 019417.



Resembled instructions: F-00, F-00w, F-00d, F-70, F-70w, F-74, F-74d, F-76, F-76w, F-76d

* Be sure to use even addresses for registers S and D. (Odd addresses such as 019003 etc. are prohibited to use.)

F-74d
nXFR

Transfers "n" double words

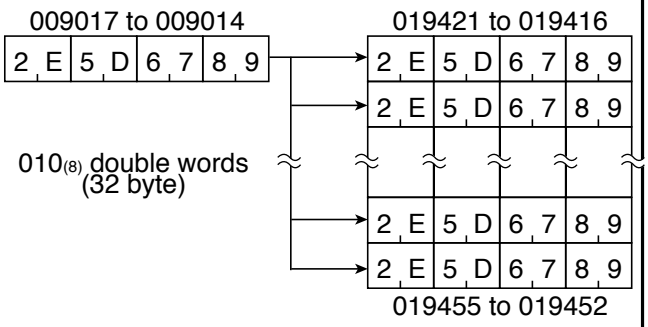
Symbol	F-74d nXFR | n | S | D	
Function	The contents of the registers S to S+3 are transferred to "n" double words registers headed by the registers D to D+3.	
Operation	S to S+3 →D to D+3,···,D+4n−4 to D+4n−1	
n	Use range 000 to 377(8) (256 double words when 000(8))	
S	Use range C *	
D	Use range C *	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S to S+3	Unchanged
	D to D+3 ⋮ D+4n−3 to D+4n−1	Contents of the register S to S+3 ⋮ Contents of the register S to S+3
	Flag	Unchanged

[Explanation]

Instruction	
S T R	004000
F-74d	
	010
	009014
	019416

004000 — F-74d nXFR | 010 | 009014 | 019416

When the input condition 004000 changes from OFF to ON, the contents of the registers 009014 to 009017 are transferred to a 010(8) double words registers headed by the registers 019416 to 019421.



* Be sure to use even addresses for registers S and D.
Resembled instructions: F-00, F-00w, F-00d, F-70, F-70w, F-70d, F-74, F-74w, F-76, F-76w, F-76d

F-76 FILR — Transfer "n" bytes in batch (FILR)

Symbol	F-76 FILR | S_1 | S_2 | D	
Function	Transfer data from "S_2 to S_2+(S_1) -1" to register "D to D+(S_1)-1" as a package. - (S_1) is the number of bytes specified by register S_1.	
Operation	S_2, ⋯, S_2+ (S_1) −1→D, ⋯, D+ (S_1) −1	
S_1	Use range A - The contents of register S_1 are 000(8) to 377(8). If set to 000(8), 256 bytes of data are transferred.	
S_2	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S_1	Unchanged
	S_2 to S_2+ (S_1) −1	Unchanged
	D, D+1, ⋮ D+ (S_1)−2 D+ (S_1)−1	Contents of register S_2, Contents of register S_2+1, ⋮ Contents of register S_2+ (S_1) −2 Contents of register S_2+ (S_1) −1
	Flag	Unchanged

[Explanation]

Instruction	
S T R	004000
F -76	
	009000
	⊐00200
	019200

004000 | F-76 FILR | 009000 | ⊐00200 | 019200

When the input condition of 004000 changes from OFF to ON, this instruction transfers data with the byte count specified by register 009000, from register ⊐00200 to the register area that begins with register 019200.

- When the contents of register 009000 are 012 (D) (014 (8))

	Before operation			After operation		
12 bytes	⊐00200	0	1	019200	0	1
	⊐00201	2	3	019201	2	3
	⊐00202	4	5	019202	4	5
	⊐00203	6	7	019203	6	7
	⊐00204	8	9	019204	8	9
	⊐00205	1	1	019205	1	1
	⊐00206	2	2	019206	2	2
	⊐00207	3	3	019207	3	3
	⊐00210	4	4	019210	4	4
	⊐00211	5	5	019211	5	5
	⊐00212	6	6	019212	6	6
	⊐00213	7	7	019213	7	7

- When the contents of register 009000 are 003 (D) (003 (8)) :

				After operation		
3 bytes	⊐00200	0	1	019200	0	1
	⊐00201	2	3	019201	2	3
	⊐00202	4	5	019202	4	5

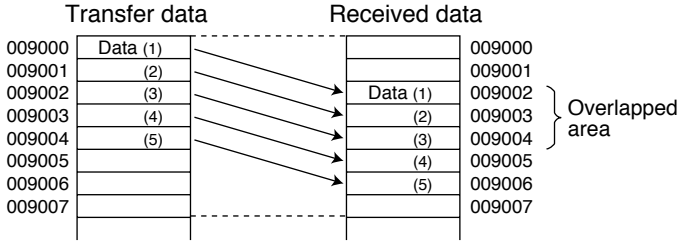
- Do not transfer the contact point area (file address 00001600 to 00001777(8) etc.) of the timer/counter of file number 0.

F-76 FILE | 009000 | 019000 | ⊐01575 Contents of 009000 = 005(8)

- If programmed in the above manner, the contents of 019003 and 019004 will be transferred to 00001600 and 00001601.

- Registers S_1, S_2, and D may be set to values which cause part of the source area to be overlapped with the destination area :

004002 | F-76 FILR | 019000 | 009000 | 009002



Resembled instructions: F-00, F-00w, F-00d, F-70, F-70w, F-70d, F-74, F-74w, F-74d, F-76w, F-76d

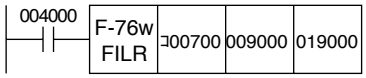
F-76w FILR

Transfer "n" words in batch (FILR)

Symbol	F-76w FILR | S_1 | S_2 | D	
Function	Transfer data from "S_2 to S_2+2(S_1)−1" to register "D to D+2(S_1)−1" as a package. - (S_1) is the number of words specified by register S1.	
Operation	S_2, S_2+1, ⋯, S_2+ 2 (S_1) −1 →D,D+1, ⋯, D+ 2 (S_1) −1	
S_1	Use range A - The contents of register S_1 are 000 to 377(8). If set to 000(8), 256 words of data are transferred.	
S_2	Use range B *1	
D	Use range B *2	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S_1	Unchanged
	S_2 to S_2+ (S_1) −1	Unchanged
	D, D+1, ⋮ D+ 2 (S_1)−2 D+2 (S_1)−1	Contents of register S_2, Contents of register S_2+1, ⋮ Contents of register S_2+2 (S_1) −2 Contents of register S_2+2 (S_1) −1
	Flag	Unchanged

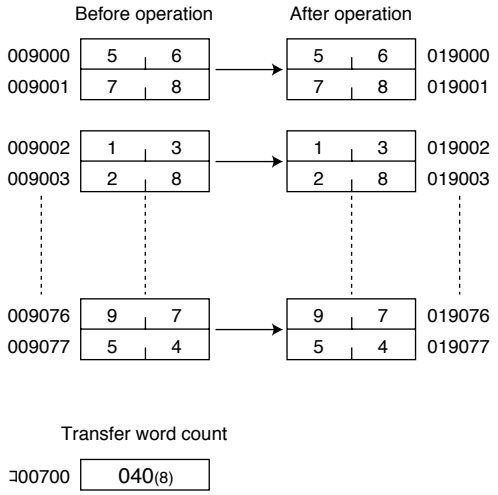
*1 Be sure to use even addresses for register S_2.
*2 Be sure to use even addresses for register D.
Resembled instructions: F-00, F-00w, F-00d, F-70, F-70w, F-70d, F-74, F-74w, F-74d, F-76, F-76d

[Explanation]



Instruction	
S T R	004000
F -76w	
	⊐00700
	009000
	019000

If the contents of location ⊐00700 are 040(8), this instruction transfers 040(8) words (32 words) of data in registers 009000 to 009077 to registers 019000 to 019077. The contents of registers 009000 to 009077 are unaffected.



F-76d
FILR

Transfer "n" double words in batch (FILR)

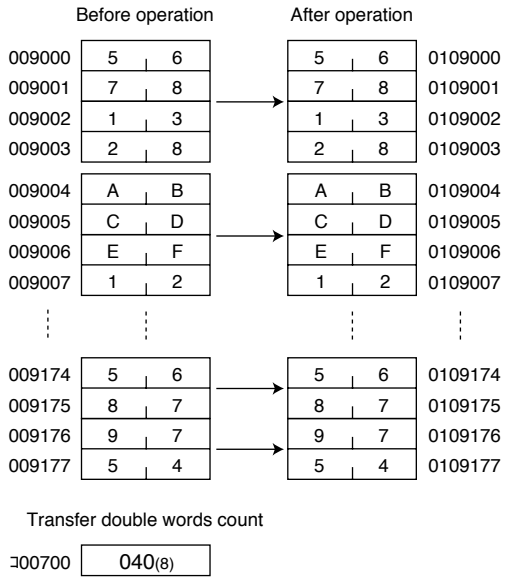
Symbol	F-76d FILR | S_1 | S_2 | D	
Function	Transfer data from "S_2 to S_2+4(S_1)−1" to register "D to D+4(S_1)−1" as a package. - (S_1) is the number of double words that is specified by the register S_1.	
Operation	S_2 to S_2+3, ·····, S_2+ 4 (S_1) −4 to S+4(S_1)−1 →D to D+3, ·····, D+ 4(S_1) −4 to D+4(S_1)−1	
S_1	Use range A - The contents of register S_1 are 000 to 377(8). If set to 000(8), 256 double words of data are transferred.	
S_2	Use range C *1	
D	Use range C *2	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S_1	Unchanged
	S_2,···, to S_2 +4 (S_1)−1	Unchanged
	D to D+3, D+4 to D+7 ⋮ D+ 4n−4 to D+4n−1	Contents of register S_2 to S_2+3 Contents of register S_2+4 to S_2+7 ⋮ Contents of register S_2+4 (S_1) −4 Contents of register S_2+4 (S_1) −1
	Flag	Unchanged

[Explanation]

Instruction	
S T R	004000
F -76d	⊐00700
	009000
	019000

004000 | F-76d FILR | ⊐00700 | 009000 | 019000

If the contents of location ⊐00700 are 040(8), this instruction transfers 040(8) double words (128 bytes) of data in registers 009000 to 009177 to registers 019000 to 019177 (128 bytes).
The contents of registers 009000 to 009077 are unaffected.

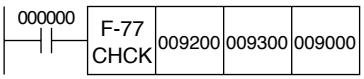


*1 Be sure to use even addresses for register S_2.
*2 Be sure to use even addresses for register D.
Resembled instructions: F-00, F-00w, F-00d, F-70, F-70w, F-70d, F-74, F-74w, F-74d, F-76, F-76w

F-77 CHCK — Generates sum check code (CHecK Code)

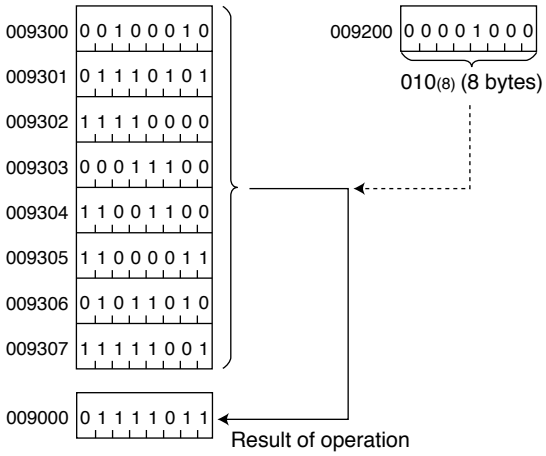
Symbol	F-77 CHCK | S_1 | S_2 | D	
Function	Generates a sum check code for the contents of registers S_2 through S_2+ (S_1) −1 (the byte count is specified by the contents of register S_1), and store the result in register D.	
Operation	0− Σ (S_2 to S_2+ (S_1) −1) →D	
S_1	Use range A - The contents of register S_1 are 000 to 377(8). If set to 000(8), 256 bytes of data are transferred.	
S_2	Use range A	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S_1	Unchanged
	S_2 to S_2+ n−1	Unchanged
	D	Result of operation
	Flag	Unchanged

[Explanation]



Instruction	
S T R	000000
F -77	
	009200
	009300
	009000

When the input condition of 000000 changes from OFF to ON, this instruction generates a sum check code for register 009300 and subsequent registers, for the number of bytes specified by the contents of register 009200, and stores the result in register 009000.
If register 009200 specifies 8 bytes : A sum check code for registers 009300 to 009307 is generated and stored in register 009000:



- Sum check code is calculated as follows :

```
  22(H)
  75
  F0
  1C
  CC
  C3
  5A
+ F9
 485
  85   ------ Obtains the complement of the least significant 2 digits.
  7B   ------ Two's complement (100(H)−85(H))
```

The value of checksum code is 7B(H).

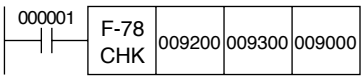
F-78 CHK — Check data (CHecK)

Symbol	F-78 CHK | S_1 | S_2 | S_3
Function	Generate a sum check code for the contents of registers S_2 through S_2+ (S_1) −1 (by the number of bytes specified by the contents of register S_1), compare it with another sum check code (generated by the F-77 instruction) stored in register S_3, and set or reset the flag according to the comparison result. - (S_1) is the number of bytes specified by register S_1.
Operation	〔0− Σ (S_2 to S_2+ (S_1) −1)〕 ⟷ S_3 (Compared) → Flag
S_1	Use range A - The contents of register S_1 are 000 to 377(8). If set to 000(8), 256 bytes of data are transferred.
S_2	Use range A
D	Use range A
Condition	Rising edge of input signal (OFF to ON)

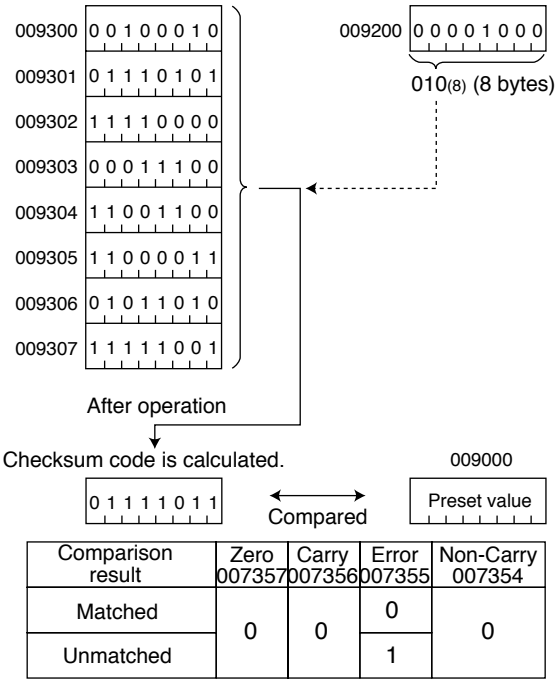
Contents after operation						
	S_1	Unchanged				
	S_2	Unchanged				
	S_3	Unchanged				
	Flag	Result of operation	Zero 007357	Carry 007356	Error 007355	Non-Carry 007354
		No error	0	0	0	0
		Sumcheck error	0	0	1	0

[Explanation]

Instruction	
S T R	000001
F-78	009200
	009300
	009000



When the input condition of 000001 changes from OFF to ON, this instruction computes the sum check code for register 009300 and subsequent registers, for the number of bytes specified by the contents of register 009200, and compares it with another sum check code (generated using F-77) stored in register 009000.



Comparison result	Zero 007357	Carry 007356	Error 007355	Non-Carry 007354
Matched	0	0	0	0
Unmatched			1	

F-79 SORT Sort register (1 byte) data

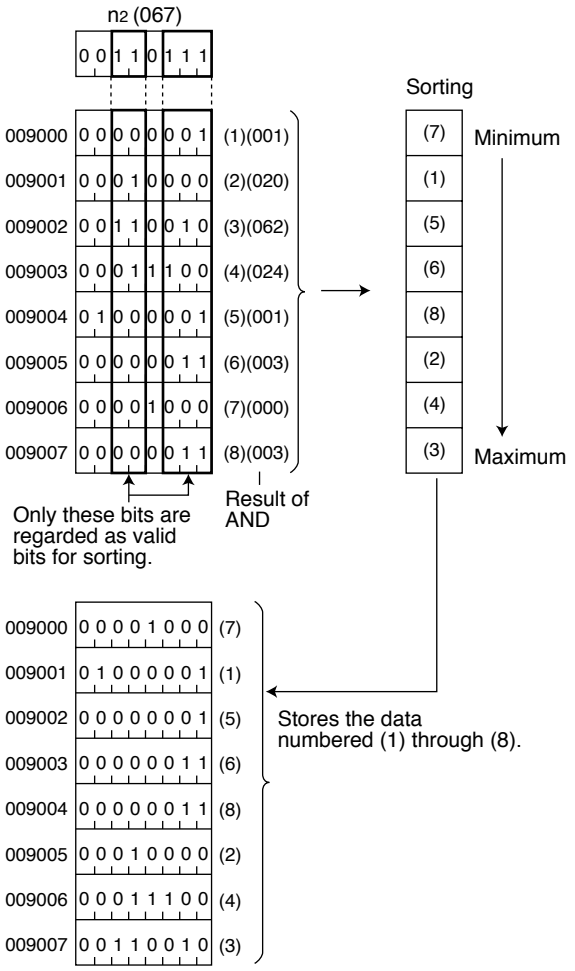
Symbol	F-79 SORT | S | n_1 | n_2	
Function	Sorts the contents of registers S to S+n_1−1 (1 byte each) in ascending order of magnitude. n_2 is a mask value for the data bits to be compared.	
Operation	S < S+1 < ⋯< S+n_1−2 < S+n_1−1	
S	Use range A	
n_1	Use range 000 to 377(8) (256 bytes for 000)	
n_2	Use range 000 to 377(8) - Values 000 and 377 for n_2 have the identical functions, making all bits valid.	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S to S+n_1-1	Result (sorted in ascending order)
	Flag	Unchanged

[Explanation]

Instruction	
S T R	005001
F -79	
	009000
	010
	067

005001 — F-79 SORT | 009000 | 010 | 067

When the input condition of 005001 changes from OFF to ON, this instruction ANDs the contents of registers 009000 to 009007 (8 bytes specified by n_1) with 067(8) (the value specified by n_2), compares and sorts the results in the ascending order of magnitude, and places them in register 009000 subsequent registers.



Data values are not affected by sorting.

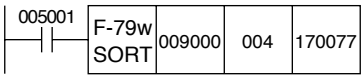
F-79w SORT — Sort register (1 word) data

Symbol	F-79w SORT | S | n_1 | n_2	
Function	Sort the details (1 word each) of register S, S+1 to S+2n_1-2, S+2n_1-1 in order of small to large. n_2 is a mask value for the data bits to be compared.	
Operation	S・S+1<…< S+ 2n_1−2・S+2n_1−1	
S	Use range B *	
n_1	Use range 000 to 377(8) (256 words for 000)	
n_2	Use range 000000 to 177777(8) - Values 000000 and 177777 for n_2 have the identical functions, making all bits valid.	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S to S + 2n_1-1	Result (sorted in ascending order)
	Flag	Unchanged

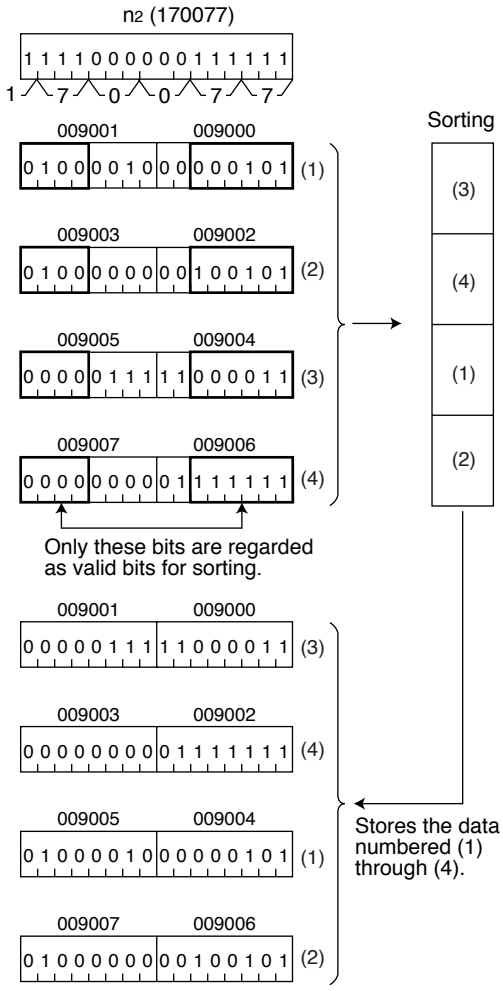
* Be sure to use an even address for register S.
(Such as 019003 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	005001
F -79w	
	009000
	004
	170077



When the input condition of 005001 changes from OFF to ON, this instruction ANDs the contents of registers 009000 to 009007 (4 words specified by n_1) with 170077(8) (the contents of of n_2), compares and sorts the results in the ascending order of magnitude, and places them in register 009000, 009001 and subsequent registers.

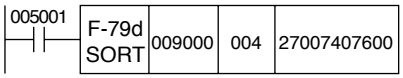


Data values are not affected by sorting.

F-79d SORT Sort register (2 words) data

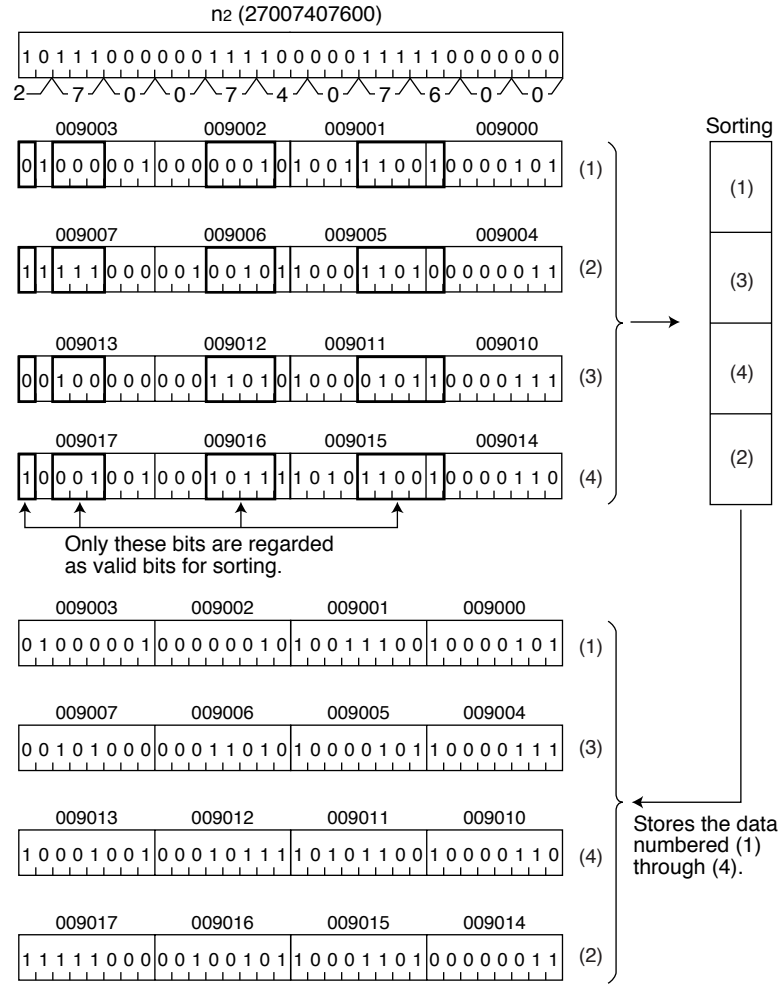
Symbol	F-79d SORT | S | n_1 | n_2
Function	Sort the details (2 words each) of register S to S+3 to S+4n1−4 to S+4n1−1 in order of small to large. n_2 is a mask value for the data bits to be compared.
Operation	S to S+3 <···< S+4n_1−4 to S+4n_1−1
S	Use range C - Be sure to use an even address for register S. (Such as 019003 etc. are prohibited to use.)
n_1	Use range 000 to 377(8) (256 double words for 000(8))
n_2	Use range 00000000000 to 37777777777(8) - Values 00000000000 and 37777777777 for n_2 have the identical functions, making all bits valid.
Condition	Rising edge of input signal (OFF to ON)
Contents after operation	S to S+4n_1−1: Result (2 words each sorted in ascending order) Flag: Unchanged

[Explanation]



Instruction
S T R 005001
F -79d 009000
004
27007407600

When the input condition of 005001 changes from OFF to ON, this instruction ANDs the contents of registers 009000 to 009017 (4 double words specified by n_1) with 27007407600(8) (the contents of of n_2), compares and sorts the results in the ascending order of magnitude, and places them in register 009000 to 009003 and subsequent registers.



Data values are not affected by sorting.

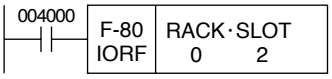
Chapter 13 Application instructions (F-80 to F-173d)

F-80
IORF

I/O refresh (I/O ReFresh)

Symbol	F-80 IORF | RACK·SLOT					
Function	Exchanges one byte of data between the PC's data memory and all data of the I/O module specified by the rack and slot numbers.					
Operation	Input module → data memory Output module ← data memory					
RACK	Use range 0 to 7					
SLOT	Use range 0 to 7					
Condition	When the input signal is ON (not limited to an OFF to ON change)					
Contents after operation	Input module	Data memory update	If a bus error occurs, the data memory or output state is not updated.			
	Output module	Output state update				
	Flag	After I/O refresh	Zero 007357	Carry 007356	Error 007355	Non-Carry 007354
		Bus error	0	0	1	0
		No module	0	0	0	1
		Transfer complete	0	1	0	0
		Non execution	0	0	0	0

[Explanation]



Instruction	
S T R	04000
F -80	
	R 0
	S 2

When the input condition of 004000 is ON, this instruction exchanges data between the data memory and the 1st byte of the I/O module installed in slot No. 2, rack No. 0.

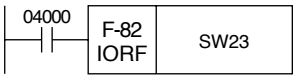
- This instruction may be used any time repeatedly in a program. The rack, slot need not be assigned the same number.
- The I/O module subjected to I/O refreshing by this instruction is also subjected to data refresh during normal I/O cycles.
- The I/O relay which is being refreshed by this instruction cannot be forcibly set or reset from the hand-held programmer (JW-15PG) or other programming tool. (For details of forced set/reset, see the instruction manual for the hand-held programmer.)
- The addresses on an I/O module that are used for input interrupt (defined in system memory locations #0240 to #0243) cannot be used for I/O refreshing.
- The upper limit of the slot number (SLOT) depends on the rack panel being used.
- Non-execution occurs when the input condition is inactive. All flags are reset.

F-82 IORF — Special I/O refresh (I/O ReFresh)

Symbol	F-82 IORF — SW
Function	Exchanges data (16 bytes) and the control relay between the special I/O module specified by the SW (It is fixed number by rack no. and module no. switch) number and the PLC data memory.
Operation	Special input module → Data memory Special output module ← Data memory
SW	Set up range 00 to 77(8) (Upper digit: Rack no. (0 to 7) Lower digit: Module no. switch (0 to 7))
Condition	When the input signal is ON (not limited to an OFF to ON change)

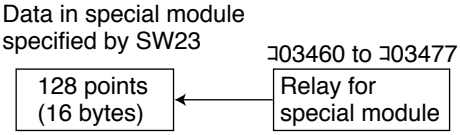
Contents after operation	Special Input module	Update of data memory	Data memory and output state are not updated during a bus error or special I/O error			
	Special Output module	Update of output state				
	Flag	Result of operation	Zero 0007357	Carry 07356	Error 007355	Non-carry 007354
		Special module error or bus error	0	0	1	0
		Transfer complete	0	1	0	0
		Missing transfer data or special module	0	0	0	1
		During non -execution	0	0	0	0

[Explanation]



Instruction
S T R 004000
F -82
SW23

When input condition 004000 is ON, the data (16 bytes) in the special module for rack 2, module no. switch 3 and the PC's data are refreshed.



- This instruction can be used any number of times during program operation.
- This instruction is for the exchange of data between the special I/O module and the special I/O module relay. The number of data bytes transferred is 16.
- The special I/O module which is data refreshed with this instruction performs the data refresh operation also during an ordinary I/O cycle.
- If this instruction is executed when the data conversion for the special I/O module has not been completed, the non-carry flag (007354) turns ON.
- The error flag (007355) turns ON when a bus error is detected during I/O refresh or when a data error or bus error due to noise is detected.
- Non-execution refers to the state when the input condition is OFF. All flags turn OFF.
- Special I/O error refers to the error signal that is output from the special I/O module. Some module do not provide this signal.
- Special I/O module that is installed in remote I/O slave station (JW-21RS) can't refresh.

F-85 PRRD Read from special I/O

F-86 PRWD Write to special I/O

	F-85 PRRD	F-86 PRWD
Symbol	F-85 PRRD | n_1 | SW,n_2 | D	F-86 PRWD | n_1 | D | SW,n_2
Function	Reads n_1 bytes in the special I/O dedicated F instruction area (n_2) of the special I/O module specified by the SW (It is fixed number by rack no. and module no. switch) and loads them to D + n_2−1 starting with register D.	Transfers the contents of n_1 bytes starting with register D to the special I/O dedicated F instruction area (n_2) of the special module specified by the SW (It is fixed number by rack no. and module no. switch).
n_1	Use range 000 to 377(8) (When "000(8) " is specified, the area will be 256 bytes.)	Use range 000 to 377(8) (When "000(8) " is specified, the area will be 256 bytes.)
SW	Use range 00 to 77(8) 〈Upper digit: Rack No. (0 to 7) 〈Lower digit: Module No. switch (0 to 7)	Use range 00 to 77(8) 〈Upper digit: Rack No. (0 to 7) 〈Lower digit: Module No. switch (0 to 7)
n_2	Special I/O dedicated F instruction area 0 to 3	Special I/O dedicated F instruction area 0 to 3
D	Use range A	Use range A
Condition	Rising edge of input signal (OFF to ON)	Rising edge of input signal (OFF to ON)
Contents after operation: n_1	Unchanged	Unchanged
Contents after operation: SW	Unchanged	Unchanged
Contents after operation: n_2	Unchanged	Unchanged
Contents after operation: D to D+n_1−1	Contents of n_2	Unchanged

Contents after operation — Flag

Result of operation	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
No response from special module	0	0	1	0
When waiting for transfer	0	0	0	1
When transfer completed	0	1	0	0
None of the above	0	0	0	0

Contents after operation — Flag: Same as left

- The F-85 (PRRD) and F-86 (PRWD) instructions are used to read and write the parameter contents for the special I/O module (JW-21SU, JW-21PS, JW-21DU/22DU).
- Do not use the two instructions above when the special module is not being used. Or this may cause the malfunction.
- Each block for the special I/O dedicated F instruction area (n_2) is specified by a numeric value 0 to 3.
  - Special I/O dedicated F instruction A block → 0
  - Special I/O dedicated F instruction B block → 1
  - Special I/O dedicated F instruction C block → 2
  - Special I/O dedicated F instruction D block → 3

  Each block contains 256 bytes.

F-90
REM

Remark (REMark)

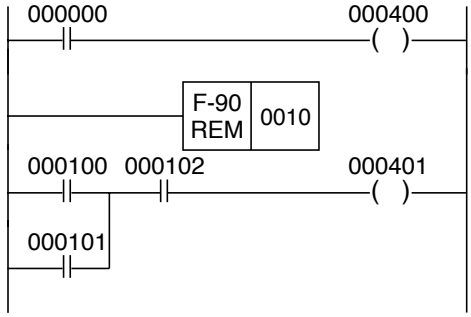
Symbol	F-90 REM n
Function	When printing ladder chart and instruction words, executed printing line comment on the ladder logic programming software (JW-300SP).
Operation	NOP (This instruction does not cause the PC to perform an operation)
n	Use range 0000 to 3777(8)
Contents after operation	Data memory of flag etc. is unchanged.

[Explanation]

- Symbols and comments for JW-300SP are registered with "symbol/comment setting" by multipurpose programmer or ladder software. (Symbol: 16 half characters, Comments: 28 half characters)
- When printing ladder chart, the multipurpose programmer or ladder chart prints symbol and comment contents. It does not print F-90 instruction in this case.
  If @ is registered at first character of the symbol, it is paged and symbol comment contents are not printed.
- When printing instructions, the multipurpose programmer or ladder chart prints each of F-90 instruction and symbol/comment contents.
  Even if @ is registered at first character of the symbol contents, paging is not executed and the registered contents are printed.

[Example for use]

- Ladder chart programming

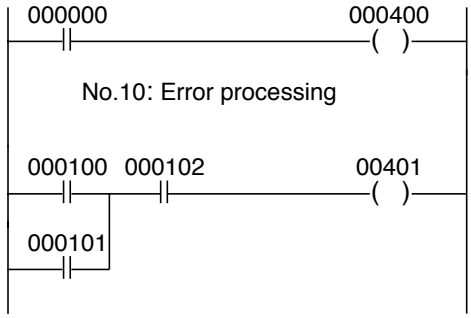


- Symbol/comment setting

	Address	Symbol	Comment
F-90	0010	No.10	Error processing
F-90	0011	@	

↑ Paging when printing ladder chart

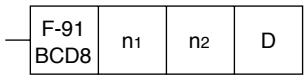
- Ladder chart printing



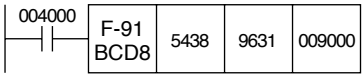
- Instruction printing

```
STR        000000
OUT        000400
F-90[REM]
             0010   No.10: Error processing
STR        000100
OR         000101
AND        000102
OUT        000401
```

F-91 BCD8 Transfer BCD constant (8 digits)

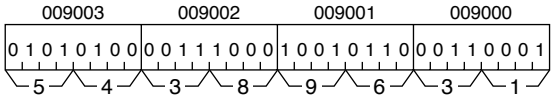
Symbol		
Function		Transfer an 8-digit BCD constant n_1, n_2 (upper and lower 4 digits, respectively) to registers D to D+3.
Operation		n_1→D+3, D+2 n_2→D+1, D
n_1		Use range 0000 to 9999 ($FFFF_{(H)}$ max.)
n_2		Use range 0000 to 9999 ($FFFF_{(H)}$ max.)
D		Use range C
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	D to D+3	Result D: 10^1 10^0 } n_2 D+1: 10^3 10^2 } n_2 D+2: 10^5 10^4 } n_1 D+3: 10^7 10^6 } n_1
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004000
F -91	
	5438
	09631
	09000

When the input condition of 004000 changes from OFF to ON, this instruction transfers the BCD constant 54389631 to registers 009000 to 009003.
After being transferred, the data will be the next value as BCD code.



Resembled instructions: F-01, F01w, F-01d

F-97
BML8

Transfers a decimal constant (8 digits)

Symbol	F-97 DML8 | n_1 | n_2 | D	
Function	Transfer an 8-digit decimal constant n_1, n_2 (n_1 x 10000+ n_2) to registers D to D+3.	
Operation	n_1 x 10000+ n_2 → D to D+3	
n_1	Use range 0000 to 9999	
n_2	Use range 0000 to 9999	
D	Use range C	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D to D+3	Result (Decimal constant: 00000000 to 99999999)
	Flag	Unchanged

[Explanation]

Instruction	
S T R	004000
F -97	
	5438
	9631
	009000

004000 | F-97 DML8 | 5438 | 9631 | 009000

When the input condition of 004000 changes from OFF to ON, This instruction transfers the decimal constant 54389631 to registers 009000 to 009003. After being transferred, the data will be the next value as binary code.

009001 | 009000

1	1	1	0	1	0	1	1	0	1	1	1	1	1	1	1
2^{15}	2^{14}	2^{13}		2^{11}		2^9	2^8	2^6	2^5	2^4	2^3	2^2	2^1	2^0	

009003 | 009002

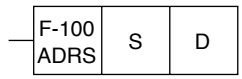
0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	1
						2^{25}	2^{24}	2^{21}	2^{20}	2^{19}	2^{18}		2^{16}		

1+2+4+8+16+32+64+256+512+2048+8192+16384+
32768+65536+262144+524288+1048576+2097152+
16777216+33554432=54389631

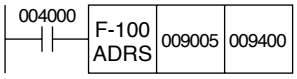
- Binary code bits have the following weights:

	7	6	5	4	3	2	1	0
D	128 (2^7)	64 (2^6)	32 (2^5)	16 (2^4)	8 (2^3)	4 (2^2)	2 (2^1)	1 (2^0)
D+1	32768 (2^{15})	16384 (2^{14})	8192 (2^{13})	4096 (2^{12})	2048 (2^{11})	1024 (2^{10})	512 (2^9)	256 (2^8)
D+2	8388608 (2^{23})	4194304 (2^{22})	2097152 (2^{21})	1048576 (2^{20})	524288 (2^{19})	262144 (2^{18})	131072 (2^{17})	65536 (2^{16})
D+3	———	———	———	———	———	67108864 (2^{26})	33554432 (2^{25})	16777216 (2^{24})

F-100 ADRS Set indirect address

Symbol		
Function	Convert the file address of register S to file N and address n for the indirect address, and set register D to D+2.	
Operation	S→D, D+1, D+2	
S	Use range A	
D	Use range C - Be sure to use even addresses for registers D.	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	D	Lower address n of the register S
	D+1	Lower address n of the register S
	D+2	File number N of register S
	Flag	Unchanged

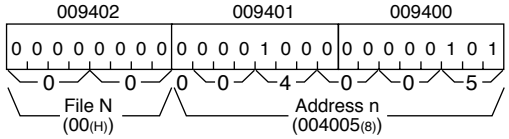
[Explanation]



Instruction	
S T R	004000
F -100	
	009005
	009400

When input condition 004000 changes from OFF to ON, the JW300 converts register 009005 (file address 00004005(8)) to file N and address n for indirect address, and sets them to registers 009400 to 009402.

- Register after operation



- Register 009005 is address 004005(8) of file 00(H).

The set file N and address n will be a direct address when specifying indirect address. (@009400)

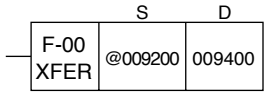
- As for "file N" and "address N" see "indirect address specification."

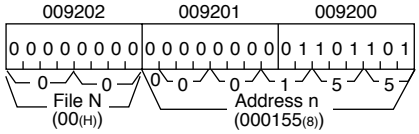
Reference

Indirect addressing:

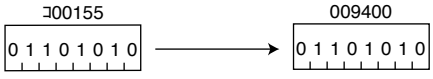
Some data processing instructions allow the use of indirect addresses for the source and destination. For indirect addressing, the operation is carried out between the registers with file addresses specified by the 3-byte contents of the source and destination areas, each which begin with the source and destination registers, respectively. Indirect addressing is identified by symbol "@" prefixing the register address.

Example: This instruction transfers the contents of the registers addressed by the contents of registers 009200 to 009202 to register 009400.





In this example, file 00(H)'s file address, 000155(8), is ⊐00155 in byte address. Consequently address @009200 points to address ⊐00155.



F-101 SEGM Set indirect address

Symbol	F-101 SEGM | n | FILE F | D	
Function	Convert file F and file address n to file N and address n that specifies indirect address, and set file N to register D+2 and address n to register D, D+1.	
Operation	n (For indirect address)→D, D+1 File N (For indirect address)→D+2	
n	Use range 00000000 to 37777777(8) (When this is the file register, it shall be a byte address)	
F	0 (Data memory except file register) 1 (File register)	
D	Use range C - Be sure to use even addresses for register D.	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	D	Lower digits of "n"
	D+1	Upper digits of "n"
	D+2	N
	Flag	Unchanged

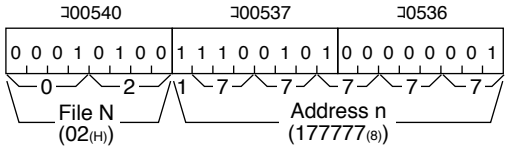
[Explanation]

Instruction	
S T R	004500
F -101	
	377777
	FILE 1
	⊐00536

004500 | F-101 SEGM | 377777 | FILE 1 | ⊐00536

When input condition 004500 changes from OFF to ON, the JW300 converts file register 377777(8) (file number 1) to file N and address n for indirect address, and sets them to register ⊐00536 to ⊐00540.

- Register after operation

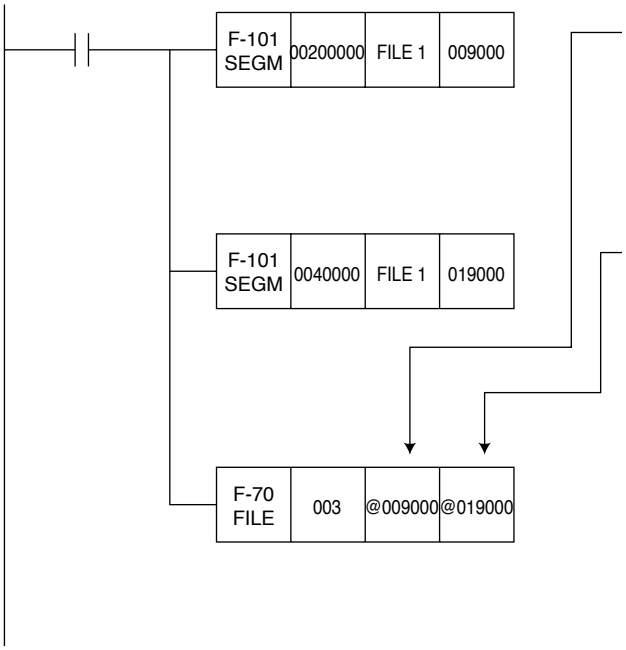


- File register 377777(8) is address 177777(8) of file address 00577777(8) and file 02(H).

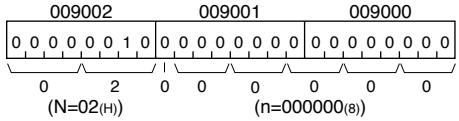
The file address thus set becomes an indirect address when the indirect address is set. (@⊐00536)

- As for "file N" and "address n" see "indirect address specification."

E.g.: A program used to transmit the first 3 bytes, starting from 00200000(8), to the 3 bytes starting from 00400000(8) in file 1.

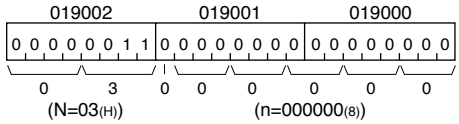


- Set the indirect address (Source side of F-70)



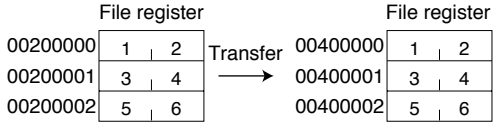
File register 00200000(8) is address 000000(8) of file address 00400000(8) and file 02(H).

- Set the indirect address (Destination side of F-70)



File register 00400000(8) is address 000000(8) of file address 00600000(8) and file 03(H).

- Transfer the 3 bytes data



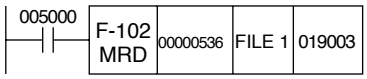
F-102 MRD — Read from the register of a direct address (1 byte)

Symbol	F-102 MRD | n | FILE F | D	
Function	The contents of the register in the file address "n" of the file number F are transferred to the register D.	
Operation	n of file number F → D	
n	Use range 00000000 to 37777777(8) (When this is the file register, it shall be a byte address)	
F	0 (Data memory except file register) 1 (File register)	
D	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	n	Unchanged
	D	Contents of the register "n"
	Flag	Unchanged

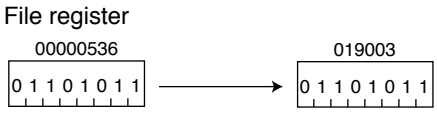
Resembled instructions: F-102w, F-176

[Explanation]

Instruction	
S T R	005000
F -102	000536 FILE 1 019003



When the input condition 005000 changes from OFF to ON, the contents of the file register 00000536(8) are transferred to the register 019003.



■ Relation between F and n

Data memory	F	Use range of "n"
Other than file register	0	00000000 to 00177777(8)
File register	1	00000000 to 37777777(8)

- Depending on the model of control module used (JW-3**CU), use range varies.

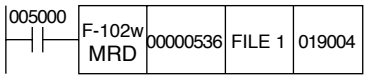
F-102w MRD — Reads from the register of a direct address (1 word)

Symbol	F-102w MRD | n | FILE F | D	
Function	The contents of the registers in the file addresses "n" and n+1 in the file number F are transferred to the registers D and D+1.	
Operation	n of file number F, n+1→ D, D+1	
n	Use range 0000000 to 37777776(8) (When this is the file register, it shall be a byte address) * Be sure to use even addresses for registers n. (Odd address such as 00000003 etc. are prohibited to use.)	
F	0 (Data memory except file register) 1 (File register)	
D	Use range B * Be sure to use even addresses for register D. (Odd address such as 019003 etc. are prohibited to use.)	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	n	Unchanged
	D	Contents of the register "n"
	D+1	Contents of the register n+1
	Flag	Unchanged

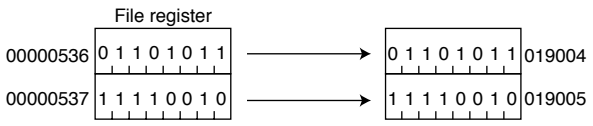
Resembled instructions: F-102, F-102d, F-176

[Explanation]

Instruction	
S T R	005000
F -102w	00000536 FILE 1 019004



When the input condition 005000 changes from OFF to ON, the 1 word contents of the file registers 00000536 and 00000537 are transferred to the registers 019004 and 019005.

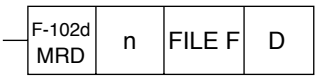


■ Relation between F and n

Data memory	F	Use range of "n"
Other than file register	0	00000000 to 00177776(8)
File register	1	00000000 to 37777776(8)

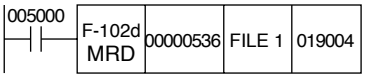
- Depending on the model of control module used (JW-3**CU), use range varies.

F-102d MRD Reads from the register of a direct address (2 words)

Symbol		
Function	The contents of the registers in the file addresses "n" to n+3 in the file number F are transferred to the registers D to D+3.	
Operation	n to n+3 of file number F → D to D+3	
n	Use range 00000000 to 37777774(8) (When this is the file register, it shall be a byte address) * Be sure to use even addresses for register n. (Odd address such as 00000003 etc. are prohibited to use.)	
F	0 (When this is the file register, it shall be a byte address) 1 (File register)	
D	Use range C * Be sure to use even addresses for register D. (Odd address such as 019003 etc. are prohibited to use.)	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	n	Unchanged
	D	Contents of the register "n"
	D+1	Contents of the register n+1
	D+2	Contents of the register n+2
	D+3	Contents of the register n+3
	Flag	Unchanged

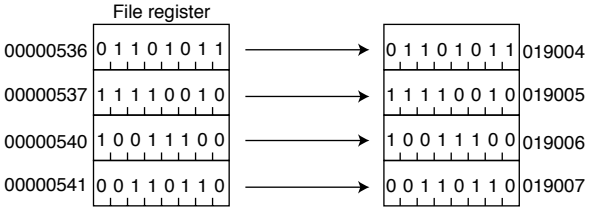
Resembled instructions: F-102, F-102w, F-176

[Explanation]



Instruction
S T R 005000
F -102d
00000536
FILE 1
019004

When the input condition 005000 changes from OFF to ON, the 1 word contents of the file registers 00000536 and 00000541(8) (2 words data) are transferred to the registers 019004 and 019007.

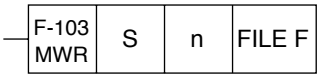


■ Relation between F and n

Data memory	F	Use range of "n"
Other than file register	0	00000000 to 00177777(8)
File register	1	00000000 to 37777774(8)

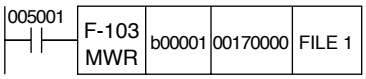
- Depending on the model of control module used (JW-3**CU), use range varies.

F-103 MWR — Write in the register of a direct address (1 byte)

Symbol		
Function	The contents of the register S are transferred to the register in the file address "n" of the file number F.	
Operation	S → n of file number F	
S	Use range A	
n	Use range 0000000 to 37777777(8) (When this is the file register, it shall be a byte address)	
F	0 (Data memory except file register) 1 (File register)	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	n	Contents of register S
	Flag	Unchanged

Resembled instructions: F-103, F-103w, F-177

[Explanation]



Instruction
S T R 005001
F -103
b00001
00170000
FILE 1

When the input condition 005001 changes from OFF to ON, the contents of the register b00001 are transferred to the file register 00170000(8).

■ Relation between F and n

Data memory	F	Use range of "n"
Other than file register	0	00000000 to 00177777(8)
File register	1	00000000 to 37777777(8)

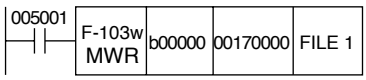
- Depending on the model of control module used (JW-3**CU), use range varies.

F-103w MWR — Write in the register of a direct address (1 word)

Symbol		
Function	The contents of the registers S and S+1 are transferred to the register in the file address "n" and n+1 of the file F.	
Operation	S, S+1 → n, n+1 of file number F	
S	Use range B - Be sure to use even addresses for register S. (Odd addresses such as 019003 etc. are prohibited to use.)	
n	Use range 00000000 to 37777776(8) (When this is the file register, it shall be a byte address) - Be sure to use even addresses for register n. (Odd addresses such as 00000003 etc. are prohibited to use.)	
F	0 (Data memory except file register) 1 (File register)	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S, S+1	Unchanged
	n	Contents of the register S
	n+1	Contents of the register S+1
	Flag	Unchanged

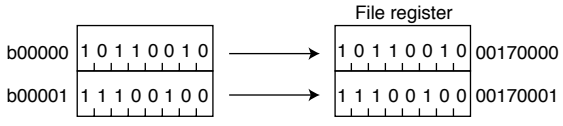
Resembled instructions: F-103, F-103d, F-177

[Explanation]



Instruction
S T R 005001
F -103w
b00000
00170000
FILE 1

When the input condition 005001 changes from OFF to ON, the word contents of the registers b00000 and b00001 are transferred to the registers of 00170000 and 00170001(8).



■ Relation between F and n

Data memory	F	Use range of "n"
Other than file register	0	00000000 to 00177776(8)
File register	1	00000000 to 37777776(8)

- Depending on the model of control module used (JW-3**CU), use range varies.

F-103d MWR

Write in the register of a direct address (2 words)

Symbol		F-103d MWR | S | n | FILE F
Function		The contents of the registers S to S+3 are transferred to the register in the file address n to n+3 of the file number F.
Operation		S to S+3 → n to n+3 of file number F
S		Use range C - Be sure to use even addresses for register S. (Odd addresses such as 019003 etc. are prohibited to use.)
n		Use range 00000000 to 37777774(8) (When this is the file register, it shall be a byte address) - Be sure to use even addresses for register n. (Odd addresses such as 00000003 etc. are prohibited to use.)
F		0 (Data memory except file register) 1 (File register)
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S to S+3	Unchanged
	n	Contents of the register S
	n+1	Contents of the register S+1
	n+2	Contents of the register S+2
	n+3	Contents of the register S+3
	Flag	Unchanged

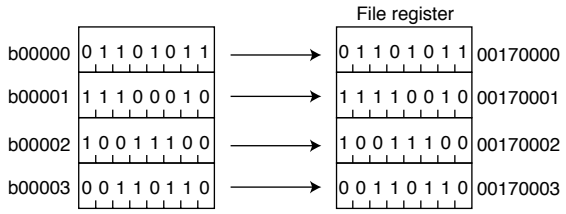
Resembled instructions: F-103, F-103w, F-177

[Explanation]

Instruction	
S T R	005001
F -103d	b00000
	00170000
	FILE 1

005001 | F-103d MWR | b00000 | 00170000 | FILE 1

When the input condition 005001 changes from OFF to ON, the 2 words data contents of the registers b00000 to b00003 are transferred to the registers of 00170000 and 00170003(8) of the file register.



■ Relation between F and n

Data memory	F	Use range of "n"
Other than file register	0	00000000 to 00177774(8)
File register	1	00000000 to 37777774(8)

- Depending on the model of control module used (JW-3**CU), use range varies.

F-112 NCMP — Compare "n" bytes (between 1-byte registers)

Symbol		F-112 NCMP S1 S2 S3
Function		Compare two data sets in size: compare data from register S_1 to a specified number of bytes (detail of register S_3) and data from register S_2 to a specified number of bytes (detail of register S_3).
Operation		Result of comparison→flag
S_1		Use range A
S_2		Use range A
S_3		Use range A - The contents of register S_3 are 000 to 377(8). If set to 000(8), 256 bytes of data are transferred.
Condition		When the input signal is ON (not limited to OFF to ON change)
Contents after operation	S_1 to $S_1+(S_3)-1$	Unchanged
	S_2 to $S_2+(S_3)-1$	Unchanged
	S3	Unchanged

Flag

Contents of register	Zero 007357	Carry 007356	Error 007355	Non-Carry 007354
S_1 to $S_1+(S_3)-1$ > S_2 to $S_2+(S_3)-1$	0	0	0	1
S_1 to $S_1+(S_3)-1$ = S_2 to $S_2+(S_3)-1$	1	0	0	1
S_1 to $S_1+(S_3)-1$ < S_2 to $S_2+(S_3)-1$	0	1	0	0

- This instruction is not capable of double length operations.

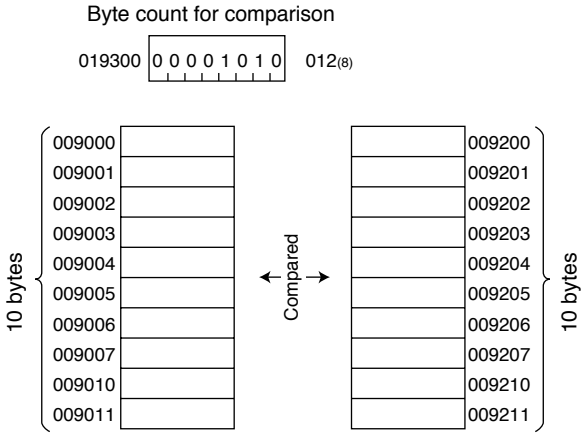
Resembled instructions: F-12, F-12w, F-12d, F-112w, F-112d, Fc12, Fc12w, Fc-12d

[Explanation]

Instruction	
S T R	004000
F-112	
	009000
	009200
	019300

004000 F-112 NCMP 009000 009200 019300

Assume that the value of register 019300 is 012(8) (10 bytes). When the input condition or 004000 is ON, this instruction compares the magnitude of the 10 bytes contents of registers 009000 to 009011 with the magnitude of the contents of registers 009200 to 009211, and sets the result in the non-carry flag , carry flag, or zero flag .



- 009000 to 009011>009200 to 009211→Non-carry ON
- 009000 to 009011=009200 to 009211→Zero ON, Non-carry ON
- 009000 to 009011<009200 to 009211→Carry ON

F-112w NCMP

Compare "n" words

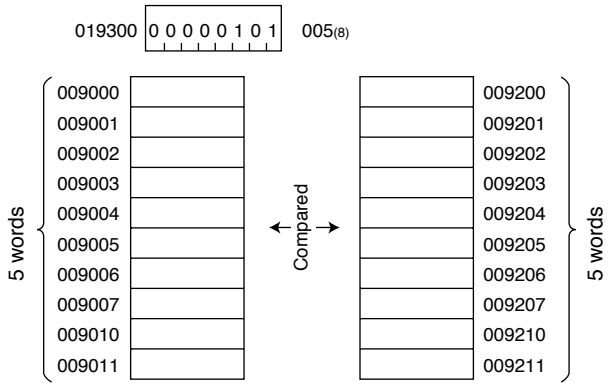
Symbol	F-112w NCMP | S_1 | S_2 | S_3					
Function	Compare two data sets in size: compare data from register S_1 to a specified number of words (detail of register S_3) and data from register S_2 to a specified number of words (detail of register S_3).					
Operation	Result of comparison→flag					
S_1	Use range B - Be sure to use even addresses for register S_1.					
S_2	Use range B - Be sure to use even addresses for register S_2.					
S_3	Use range A - The contents of register S_1 are 000 to 377(8). If set to 000(8), 256 words of data are transferred.					
Condition	When the input signal is ON (not limited to OFF to ON change)					
Contents after operation	S_1 to $S_1+2(S_3)-1$	Unchanged				
	S_2 to $S_2+2(S_3)-1$	Unchanged				
	S_3	Unchanged				
	Flag	Contents of Register	Zero 007357	Carry 007356	Error 007355	Non-Carry 007354
		S_1 to S_1+2 (S_3) -1> S_2 to S_2+2 (S_3) -1	0	0	0	1
		S_1 to S_1+2 (S_3) -1= S_2 to S_2+2 (S_3) -1	1	0	0	1
		S_1 to S_1+2 (S_3) -1< S_2 to S_2+2 (S_3) -1	0	1	0	0

[Explanation]

Instruction	
S T R	002000
F -112w	009000 009200 019300

002000 | F-112w NCMP | 009000 | 0092000 | 019300

Assume that the value of register 019300 is 005(8) (5 words). When the input condition or 002000 is ON, this instruction compares the magnitude of the 5 words contents of registers 009000 to 009011 with the magnitude of the contents of registers 009200 to 009211, and sets the result in the non-carry flag, carry flag, or zero flag.



- 009000 to 009011>009200 to 009211→Non-carry ON
- 009000 to 009011=009200 to 009211→Zero ON
  Non-carry ON
- 009000 to 009011<009200 to 009211→Carry ON

- This instruction is not capable of double length operations.

Resembled instructions: F-12, F-12w, F-12d, F-112, F-112d, Fc12, Fc12w, Fc12d

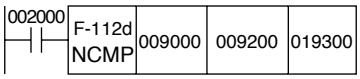
F-112d NCMP Compare "n" double words

Symbol	F-112d NCMP | S_1 | S_2 | S_3
Function	Compare two data sets in size: compare data from register S_1 to a specified number of double words (detail of register S_3) and data from register S_2 to a specified number of words (detail of register S_3).
Operation	Result of comparison → flag
S_1	Use range C - Be sure to use even addresses for register S_1.
S_2	Use range C - Be sure to use even addresses for register S_2.
S_3	Use range A - The contents of register S_3 are 000 to 377(8). If set to 000(8), 256 double words of data are transferred.
Condition	When the input signal is ON (not limited to OFF to ON change)

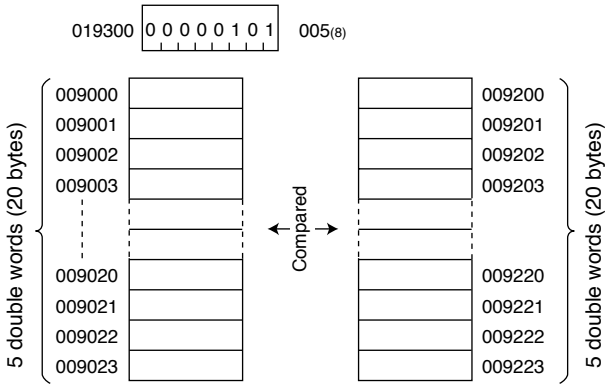
Contents after operation						
S_1 to $S_1+4(S_3)-1$	Unchanged					
S_2 to $S_2+4(S_3)-1$	Unchanged					
S_3	Unchanged					
Flag	Contents of register	Zero 007357	Carry 007356	Error 007355	Non-Carry 007354
	S_1 to S_1+4 (S_3) −1> S_2 to S_2+4 (S_3) −1	0	0	0	1
	S_1 to S_1+4 (S_3) −1= S_2 to S_2+4 (S_3) −1	1	0	0	1
	S_1 to S_1+4 (S_3) −1< S_2 to S_2+4 (S_3) −1	0	1	0	0

[Explanation]

Instruction	
S T R	002000
F -112d	009000
	009200
	019300



Assume that the value of register 019300 is 005(8) (5 double words). When the input condition or 002000 is ON, this instruction compares the magnitude of the 5 double words contents of registers 009000 to 009023 with the magnitude of the contents of registers 009200 to 009223, and sets the result in the non-carry flag, carry flag, or zero flag.



- 009000 to 009023>009200 to 009223→Non-carry ON
- 009000 to 009023=009200 to 009223→Zero ON
  Non-carry ON
- 009000 to 009023<009200 to 009223→Carry ON

- This instruction is not capable of double length operations.

Resembled instructions: F-12, F-12w, F-12d, F-112, F-112w, Fc12, Fc12w, Fc12d

F-116 DIV

Divides register value (BCD 8 digits) by another register value (BCD 8 digits) (DIVide) (Quotient has 8-digit integer part and 4 decimal fraction.)

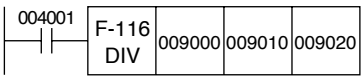
Symbol	F-116 DIV | S_1 | S_2 | D					
Function	Divide the 8 digits BCD contents of registers S_1 to S_1+3 by the 8-digit BCD contents of registers S_2 to S_2+3, and store the quotient as an 8-digit integer and 4 fractional digits to the 6 bytes register area hat begins with register D.					
Operation	(S_1 to S_1+3) ÷ (S_2 to S_2+3) →D to D+5					
S_1	Use range C *					
S_2	Use range C *					
D	Use range H *					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	S_1 to S_1+3	Unchanged				
	S_2 to S_2+3	Unchanged				
	D to D+1	Quotient (4 fractional digits)	Unaffected if the contents of registers S_1 to S_1+3 or S_2 to S_2+3 are not BCD code, or the contents of S_2 to S_2+3 are 00. (Does not calculate.)			
	D+2 to D+5	Quotient (8-digit integer)				
	Flag	Contents of registers S_1 to S_1+3 and S_2 to S_2+3	Zero 007357	Carry 007356	Error 007355	Non-Carry 007354
		BCD code	0	0	0	0
		- Not BCD code - S_2 to S_2+3 value is 0			1	

* Be sure to use even addresses for register S_1, S_2, and D.

- If the numerator is smaller than the denominator (S_1 to S_1+3<S_2 to S_2+3), the quotient (contents of D+2 to D+5) becomes 0. The 5th and subsequent decimal places in the contents of registers D and D+1 are truncated.

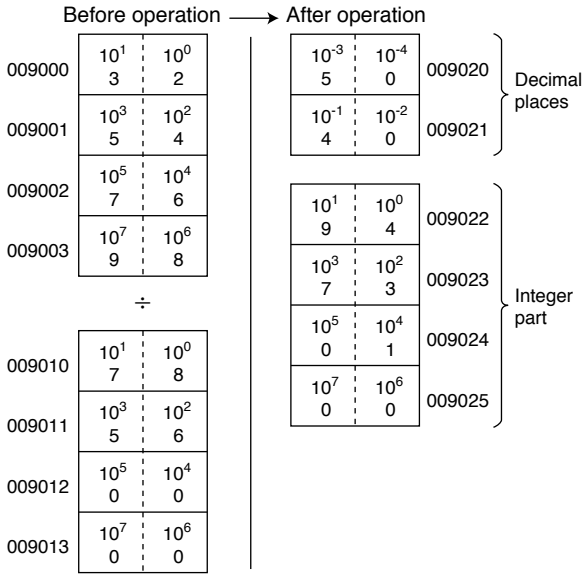
Resembled instructions: F-16, F-16d, Fc16, Fc16d

[Explanation]



Instruction	
S T R	004001
F -116	
	009000
	009010
	009020

When the input condition of 004001 changes from OFF to ON, this instruction divides the 8-digit BCD contents of registers 009000 to 009003 by the 8-digit BCD contents of registers 009010 to 009013, and stores the 4 fractional digits of the quotient to registers 009020, 009021 and the 8-digit integer of the quotient to 009022 to 009025.



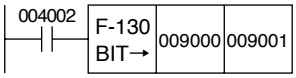
In this example, "98765432÷5678=17394.4050" is evaluated.

F-130 BIT→ Multiplexes bit (indirect address)

Symbol		F-130 BIT→ | S_1 | S_2
Function		The contents of a bit in the register S_2 determined by the contents of the register S_1 are transferred to the carry flag (007356).
Operation		Bit of $S_2\langle S_1\rangle$ → Carry flag
S_1		Use range A
S_2		Use range A
Condition		When the input signal is ON (not limited to OFF to ON change)
Contents after operation	S_1	Unchanged
	S_2	Unchanged

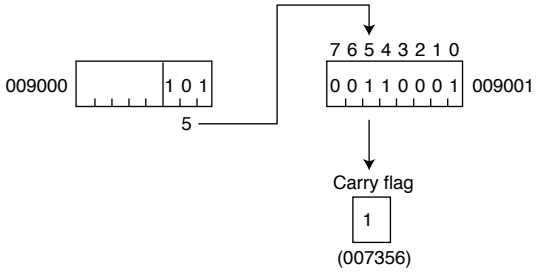
Contents after operation: Flag	State of the specific bit	Zero 007357	Carry 007356	Error 007355	Non-Carry 007354
	0 (OFF)	0	0	0	0
	1 (ON)		1		

[Explanation]



Instruction	
S T R	004002
F -130	
	009000
	009001

When the input condition of 004002 is ON, the contents of a bit in the register 009001 determined by the lower 3 bits of the register 009000 are transferred to the carry flag (007356).

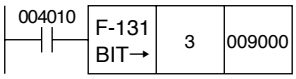


F-131 BIT→ Multiplexes bit (direct address)

Symbol		F-131 BIT→ | n | S
Function		The contents of a bit "n" the register S determined by the contents of the register S_1 are transferred to the carry flag.
Operation		Bit n of S → Carry flag
n		Use range 0 to 7
S		Use range A
Condition		When the input signal is ON (not limited to OFF to ON change)
Contents after operation	S	Unchanged

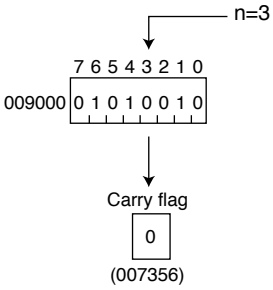
Contents after operation: Flag	State of the specific bit	Zero 007357	Carry 007356	Error 007355	Non-Carry 007354
	0 (OFF)	0	0	0	0
	1 (ON)		1		

[Explanation]



Instruction	
S T R	004010
F -131	
	3
	009000

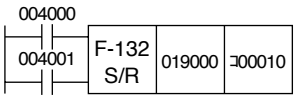
When the input condition 004010 is ON, the contents of a bit in the register 009000 are transferred to the carry flag (007356).



F-132 S/R — Sets/resets bit (indirect address) (Set/Reset)

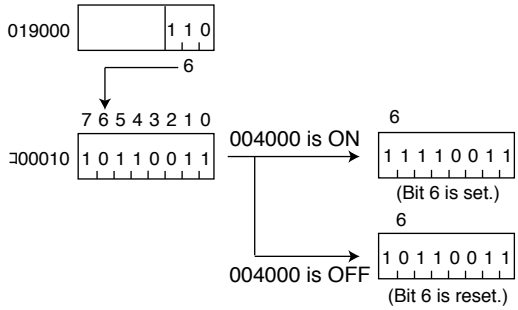
Symbol	(1) (2) — F-132 S/R | S | D (1) Set / reset direction input (2) Input condition	
Function	A bit in the register D determined by the contents of the register S (lower 3 bits) is set or reset according to the set/reset command input of (1).	
Operation	State of (1) → Bit of D〈S〉	
S	Use range A	
D	Use range A	
Condition	When the input condition (2) is ON (not limited to OFF to ON change)	
Contents after operation	S	Unchanged
	D	Only the specified bit changes
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004000
S T R	004001
F -132	
	019000
	Ⅎ00010

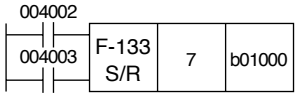
When the input condition 004001 is ON, the bit of Ⅎ00010 determined by the lower 3 bits of the register 019000 are set when 004000 is ON or reset when OFF.



F-133 S/R — Sets/resets bit (direct address) (Set/Reset)

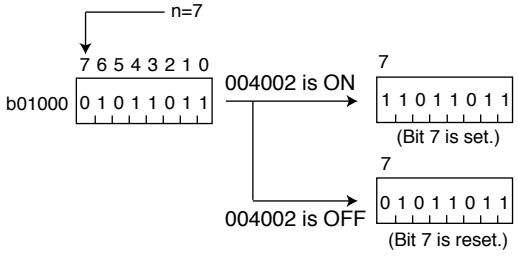
Symbol	(1) (2) — F-133 S/R | n | D (1) Set / reset direction input (2) Input condition	
Function	A bit "n" in the register D is set or reset according to the set/reset command input of (1).	
Operation	State of (1) → Bit D "n"	
n	Use range 0 to 7	
D	Use range A	
Condition	When the input condition (2) is ON (not limited to OFF to ON change)	
Contents after operation	D	Only the specified bit changes
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004002
S T R	004003
F -133	
	7
	b01000

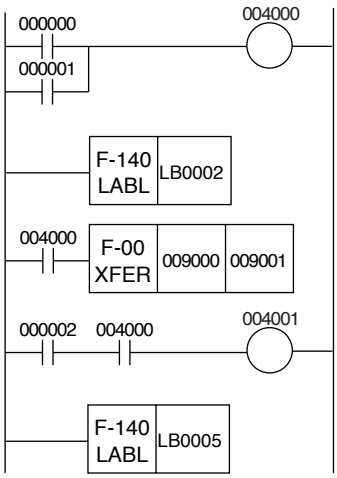
When the input condition of 004003 is ON, the bit 7 of the register b01000 is set when 004002 is ON or reset when OFF.



F-140 LABL — Set label (LABeL)

Symbol	F-140 LABL — LBn
Function	Use to set Jump destination for the F-141 (JMP) instruction. Branch destination for the F-142 (CALL) instruction. Branch destination for the F-148 (CAL+) instruction. Jump destination for the F-151 (JMP+) instruction. Branch destination for timer interrupt. Branch destination for input interrupt.
n	Use range 0000 to 1777(8)

[Example]



Address	Instruction	
0000000	S T R	000000
0000001	O R	000001
0000002	O U T	004000
0000003	F -140	
0000004		LB00002
0000005	S T R	004000
0000006	F -00	
0000007		009000
0000010		009001
0000011	S T R	000002
0000012	A N D	004000
0000013	O U T	004001
0000014	F -140	
0000015		LB00005

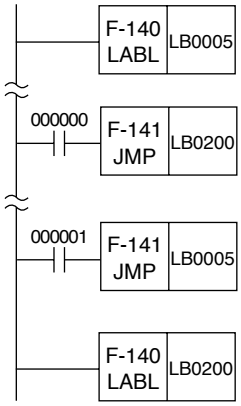
- F-140 is a label to indicate jump or subroutine destination, and does not execute the program. Therefore, the data memory is retained after execution of an F-140 instruction.

- Label numbers LB0000 to LB1777 may be used at any time, but the same number should not be used again within the program.
- Use labels LB1353 to LB1357 for timer interrupts. For usage, see the description of system memory location #0240 and of the F-143 (return from subroutine) instruction.
- Use labels LB1360 to LB1417 for input interrupts. For usage, see the description of system memory location #0241 to #0245 and of the F-143 (return from subroutine) instruction.

F-141 JMP — Jump to label (JUMP)

Symbol	F-141 JMP | LBn
Function	The control jumps to the label address LBn (F-140).
n	Use range 0000 to 1777(8)
Condition	When the input signal is ON (not to limited to OFF to ON change)

[Explanation]



Address	Instruction	
0000005	F -140	
0000006		LB0005
⋮		⋮
0000777	S T R	000000
0001000	F -141	
0001001		LB0200
⋮		⋮
0002002	S T R	000001
0002003	F -141	
0002004		LB0005
0002005	F -140	
0002006		LB0200

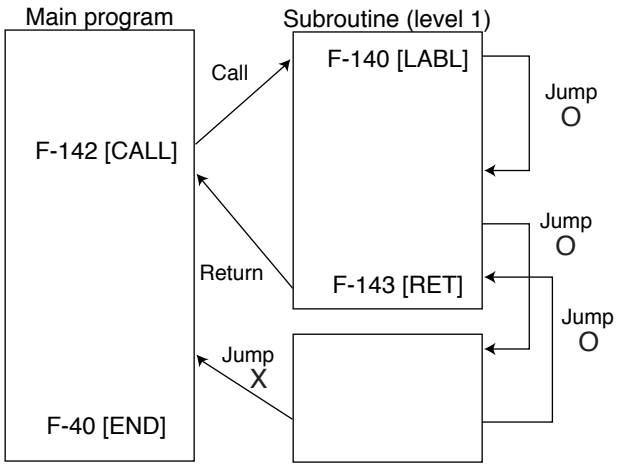
*

- When the input condition 000000 is ON, the control jumps to the program address 0002005 of the label LB0200 and instructions after the address 0002007 are executed.
- When the input condition 000001 is ON, the control jumps to the program address 0000005 of the label LB0005 and instructions after the address 0000007 are executed.

- The contents of the data memory are not affected after execution of an F-141 instruction.
- Execution may be done with F-41 (JCS) and F-42 (JCR) for the location indicated with an asterisk (*), the execution time may be saved if an F-141 (JMP) is used as it does not execute instruction down to F-140 (LABL).
- Same number may be used for the label of an F-141 instruction at any time.
- Because the control does not execute down to the jump address with an F-141 instruction, an F-40 (END instruction) will be disregarded even if there is an F-40 before the jump address.
- The jump address label (F-140) should be written in the program, in order to avoid malfunction that may occur if there is a jump address.
- Use labels LB1353 to LB1417 for timer interrupts (#0240) and for input interrupts (#0241 to #0245).

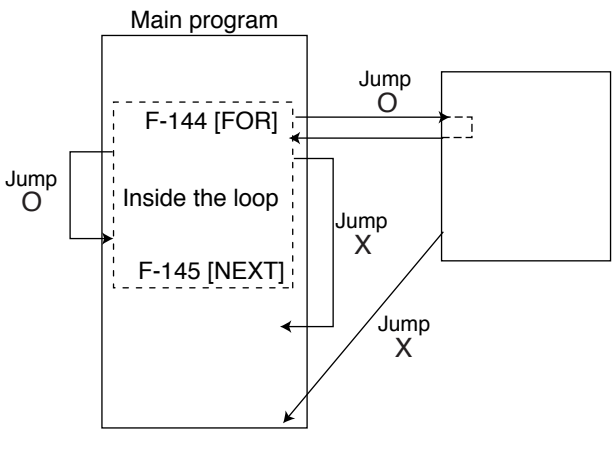
- When using the F-141 (JMP) instruction to cross multiple nested layers specified by the F-142 (CALL) and F-143 (RET) instructions.
  The F-141 (JMP) instruction can only be used when the program jumps to an address inside the same subroutine, or to jump back to a start line in the same subroutine.

[Example]



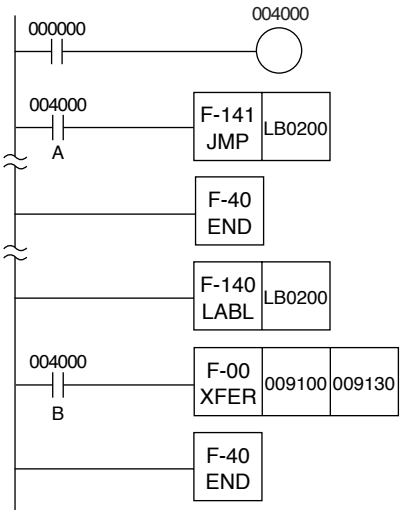
- When using the F-141 (JMP) instruction to cross multiple nested layers specified by the F-144 (FOR) and F-145 (NEXT) instructions.
  The F-141 (JMP) instruction can only be used when the program jumps to an address inside the same loop, or to jump back to a start line in the same loop.

[Example]

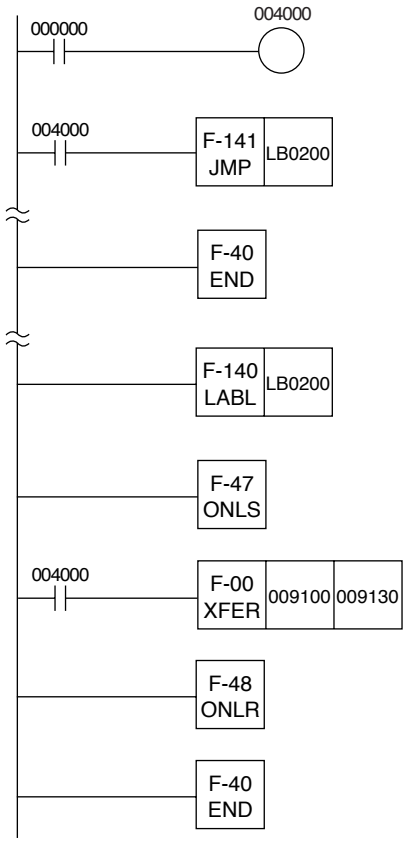


- When the following programs (ex. 1 and 2) are created using F-140 (LABL) and F-141 (JMP), F-00 (XFER) does not execute the target operation.
  Operation wise, the F-141 is executed and the control jumps to the destination specified by the F-141 instruction, when the contact 004000 (A contact) turns active for both the examples. And, the F-00 is executed when a next contact 004000 (B contact) is active.

Programming example 1 ✕



Programming example 2 ○



  - Regarding the programming example 1, the F-141 is executed in the first cycle the contact 004000 (A contact) had turned active and the control jumps to the destination specified by the F-141. For the next contact 004000 (B contact) has been ON after the jump was made, the F-00 is executed, but, the F-00 is not executed in the second cycle after the contact 004000 (a contact) has turned ON even if it is ON, because the F-00 does not recognize the rising edge of the signal as the contents of the accumulator one scan cycle before do not match the current contents of the accumulator.
  - Regarding the programming example 2, the F-00 after the jump is executed in the second cycle after the contact 004000 (A contact) has turned ON. This is because the level operation condition (F-17, F-48) is provided so as to execute the instruction after the jump at the time of ON.

Therefore, it has to be programmed like the example 2 in order to execute the program at every operational cycle after a jump.

Reference The F-00 instruction compares the contents of the previous contents of the accumulator with the current contents and executes the program when a rising edge is recognized.

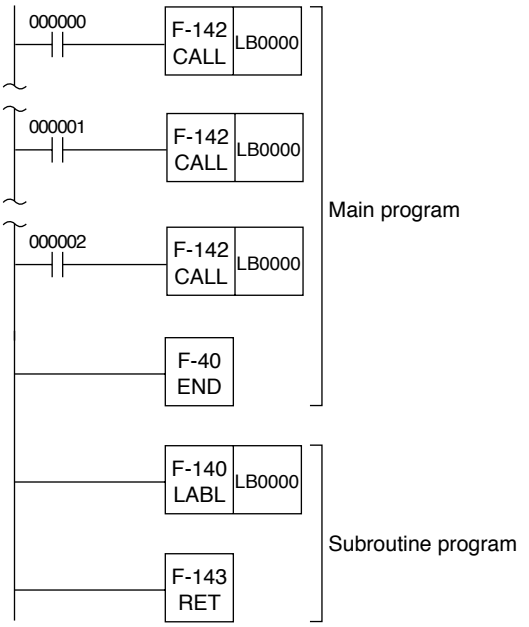
- Even if the label to jump exists in a waiting status block, the JW300 executes operation of the jump destination.

F-142 CALL — Call labeled subroutine (CALL)

F-143 RET — Return from subroutine (Return)

Symbol	F-142 CALL LBn F-143 RET
Function	The PC jumps to the subroutine of LBn (F-140) and returns with an F-143 instruction.
n	Use range 0000 to 1777(8)
Condition	Rising edge of input signal (OFF to ON)

[Explanation]

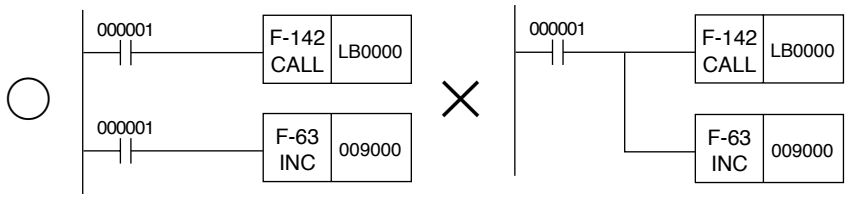


Address	Instruction	
0000010	S T R	000000
0000011	F -142	
0000012		LB0000
0000100	S T R	000001
0000101	F -142	
0000102		LB0000
0000200	S T R	000002
0000201	F -142	
0000202		LB0000
0000300	F -40	
0000301	F -140	
0000302		LB0000
0000315	F -143	

- Use of an F-142 (CALL) and F-143 (RET) instructions in the program not only eliminates need of writing the same program repeatedly within the program and saves the program size, but it will also help to make a simpler program structure.
- In the above example, when the input condition 000000 changes from OFF to ON, the control jumps from the main program to the subroutine to execute the addresses 0000303 through 0000314, and returns to resume executing the main program at the address 0000013 when an F-143 instruction is met.

- When a subroutine is called, instructions within the subroutine are set to level operation with an ON state of signal.
- The following instructions may not be used within a subroutine.
  TMR, CNT, F-30 (MCS), F-31 (MCR), F-40 (END), F-41 (JCS), F-42 (JCR), F-44 (⊣↑⊢), F-45 (⊣↓⊢), F-47 (ONLS), F-48 (ONLR).
- The F-148 (CAL+) or F-149 (RETC) instruction can also be used for subroutine calls.
- The F-143 (RET) instruction is also used to program a subroutine for timer interrupts (set with #0240) or input interrupts (set with #0241 to #0245).
- Use labels LB1353 to LB1357 for timer interrupts.
- Use labels LB1360 to LB1417 for input interrupts.
- Main program must be created first before subroutine is created and it must be affixed with an F-40 (END) instruction at the end of the subroutine.
- The F-142 instruction cannot be used in the interrupt program.
- For the F-143 does not move to execute the F-142, but moves to execute the instruction that follows the F-142.

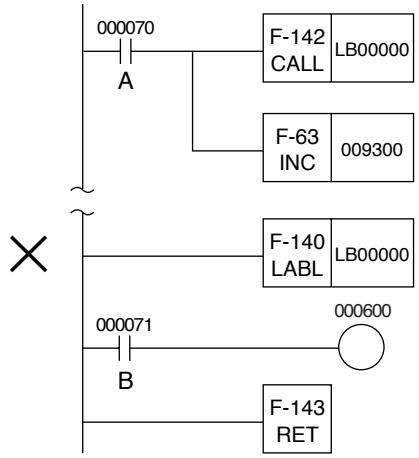
- A next step of an F-142 (CALL) instruction must begin with a contact input.



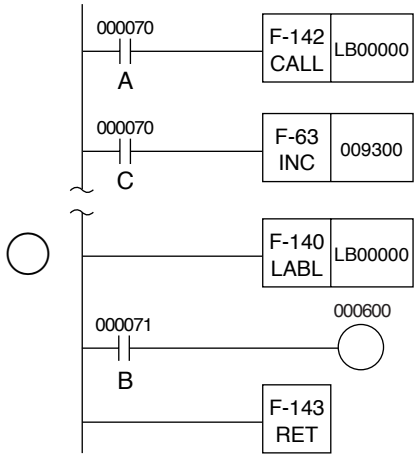
When the control returns from an F-143 (RET) instruction after executing an F-142 and its next instruction by a single contact point, a normal operation may not be expected because an input of a next step is dependent on the state within the subroutine.

- Either of programming examples 1 and 2 programmed using an F-142 (CALL) or F-143 (RET), has the same operation, but the F-63 (JNC) instruction of the programming example 1 will not operate with the condition for the contact 000070.For both the examples, the control jumps to subroutine specified by the F-142 has been executed when the contact 000070 (a contact) turned ON. And, It returns to the F-63 of the next step of the F-142 at the F-143, to execute the F-63 instruction.

Programming example 1



Programming example 2

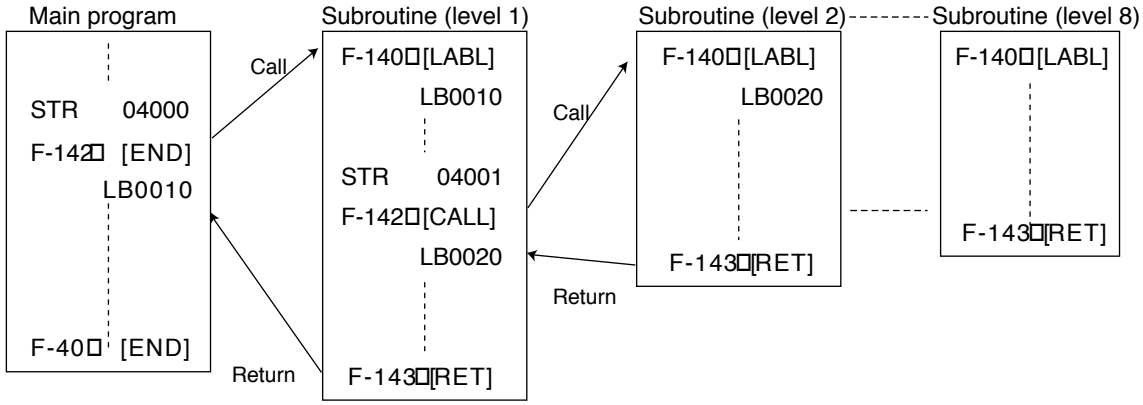


- Regarding the programming example 1, the F-142 is executed when the contact 000070 (A contact) has turned ON and the control jumps to the subroutine specified by the F-142. After executing the program steps until the F-143 is met, then moves to the step (F-63) that follows the F-142. In this event, the F-63 is executed under the condition of the contact 000071 (B contact). When the control with the F-143 returned to the step next to F-142, the F-63 may not be executed as programmed for execution as the F-63 operating condition is executed in the state of the accumulator immediately before, in this case, the state in the accumulator at the end of the subroutine.
- Regarding the programming example 2, execution does not take place under the state of the accumulator contents at the end of the subroutine after returning to the step next to the F-142, but, the execution takes place as programmed under the condition of the contact 000070 (C contact) after the return.

Therefore, it has to be as in the programming example 2 which the step immediately after the F-142 starts with a contact input.

- The F-142 (CALL) and F-143 (RET) instructions can be used to nest up to 8 subroutine calls.

[Example]



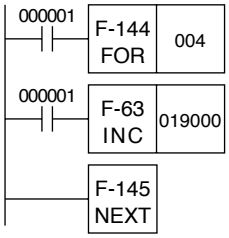
Resembled instructions: F-142, F-148, F-143, F-149

F-144 FOR — Sets loop count (FOR)

F-145 NEXT — End of loop (NEXT)

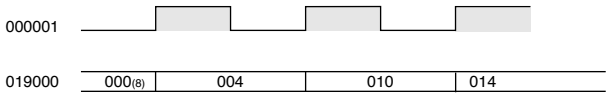
Symbol	F-144 FOR | n ; F-145 NEXT
Function	Repeats to execute the program "n" times between F-144 (FOR) and F-145 (NEXT).
n	Use range 000 to 377(8) (256 times if 000(8))
Condition	Rising edge of input signal (OFF to ON)

[Explanation]

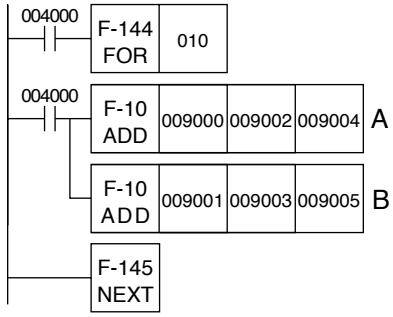


Instruction	
S T R	000001
F-144	
	004
S T R	000001
F-63	
	019000
F-145	

When the input condition 000001 changes from OFF to ON, the F-63 (INC) instruction is repeated for four times.

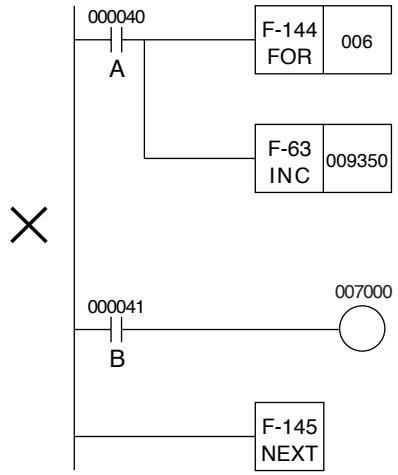


- The F-144 (FOR) must be used in conjunction with the F-145 (NEXT).
- When the F-144 (FOR) is executed, instructions between the F-144 (FOR) and F-145 (NEXT) are set ON level active.
- When the instruction is not execution, the contents of the data memory between the F-144 (FOR) and F-145 (NEXT) do not change.
- The following instruction may not be inserted between the F-144 (FOR) and F-145 (NEXT).
  TMR, CNT, F-30 (MCS), F-31 (MCR), F-40 (END), F-41 (JCS), F-42 (JCR), F-44 (⊣↑⊢), F-45 (⊣↓⊢), F-47 (ONLS), F-48 (ONLR), F-141 (JMP), F-144 (FOR), F-145 (NEXT)
- Instructions to be executed between the F-144 (FOR) and F-145 (NEXT) must be less as much as situation allows. The time required for the execution must be considered.
- Double-length operation is possible for the F-10 instruction. But the operation flag of the instruction of B will not affect the instruction of A.

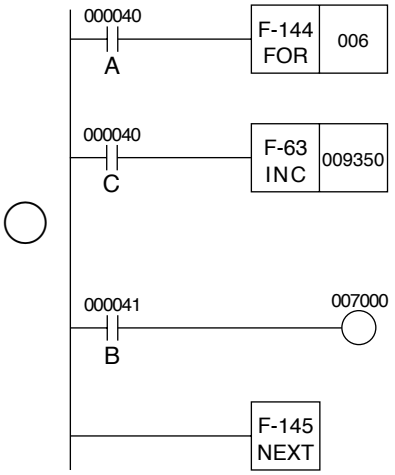


- The same operation takes place when the follow kind of program is programmed using an F-144 (FOR) and F-145 (NEXT) instructions (examples 1 and 2). But, the F-63 instruction (INC) will not be executed for the example 1.
  For both the examples 1 and 2, execution starts from the program steps that follows the F-144 through the F-145 for the times specified by the F-144, when the contact 000040 (A contact) has turned ON.

Programming example 1



Programming example 2

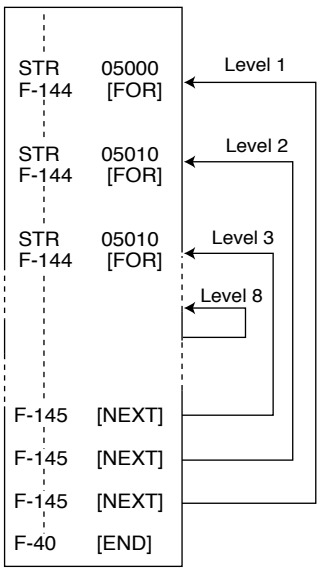


  - Regarding the programming example 1, execution takes place from the program steps that follows the F-144 through the F-145 for the times specified by the F-144 after executing the F-144 and the F-63, when the contact 000040 (A contact) turned On, but the F-63 operates in the condition of the contact 000041 (B contact). Proper program execution may not be expected because the execution is done for the F-63 execution condition according to the previous contents of the accumulator (in the state with the contents of the accumulator stored immediately before the F-145, in this instance), when it returned to the step immediately before the F-144 with the F-145.
  - Regarding the programming example 2, proper program execution is done as programmed as it is executed in the condition of the contact 000040 (C contact); not executing the program according to the accumulator contents before executing the F-145 after returning to the step immediately before the F-144 with the F-145.

  Whereas, it is has to be programmed as in the example 2 for the step that immediately follows the F-144.

- To change the loop count, use the F-146 (FORR) instruction. To exit an execution loop, use the F-147 (EXIT) instruction.
- The F-144 (FOR) and F-145 (NEXT) instructions can be used to nest up to 8 subroutine calls.

[Example]

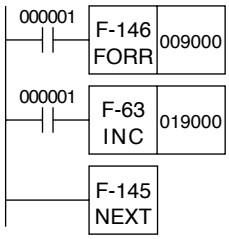


F-146 FORR Sets loop count register (FORR)

Symbol	F-146 FORR | S	
Function	Repeat the execution of the series of instructions in the loop between the F-146 (FORR) and F-145 (NEXT) instructions the number of times specified by the contents of register S.	
S	Use range A - The contents of register S is 000 to 377(8). If set to 000(8), 256 times of data are transferred.	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	Flag	Unchanged

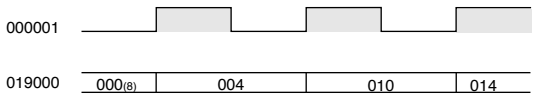
Resembled instructions: F-144

[Explanation]

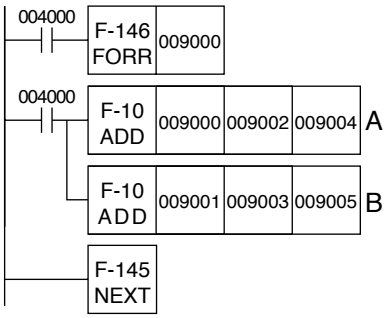


Instruction	
S T R	000001
F-146	009000
F-63	019000
F-145	

When the input condition of 000001 changes from OFF to ON, this instruction repeats the execution of the F-63 (INC) instruction the number of times specified by the contents of register 009000. (In the following example, the register 009000 contains 4.)



- The F-146 (FORR) must be used in conjunction with the F-145 (NEXT).
- When the F-146 (FORR) is executed, instructions between the F-146 (FORR) and F-145 (NEXT) are set ON level active.
- When the loop is not executed at all, the contents of the data variables in the loop remain unaffected.
- To abort loop execution before the preset loop count is reached, use the F-147 (EXIT) instruction.
- The following instructions cannot be used in the loop between the F-146 (FORR) and F-145 (NEXT) instruction : TMR, CNT, F-30 (MCS), F-31 (MCR), F-40 (END), F-41 (JCS), F-42 (JCR), F-44 (⊣↑⊢), F-45 (⊣↓⊢), F-47 (ONLS), F-48 (ONLR)
- The F-146 (FORR) and F-145 (NEXT) instructions can be used to nest up to 8 subroutine calls. The using method is same as F-144 (FOR) instruction.
- Instructions to be executed between the F-146 (FORR) and F-145 (NEXT) must be less as much as situation allows. The time required for the execution must be considered.
- While the F-10 instruction is capable of double length operations, the flags for instruction "B" above do not affect instruction "A".

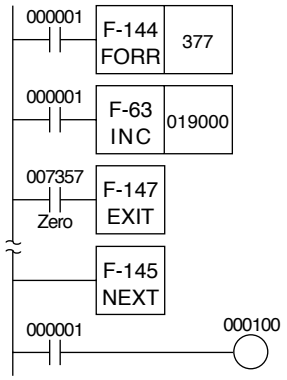


- For other notes, see those for the F-144 (FOR) instruction.

F-147 EXIT — Exits loop (conditional)

Symbol	F-147 EXIT
Function	Exit the execution loop that exists between (and including) the F-144 (FOR) or F-146 (FORR) and F-145 (NEXT) instructions before the preset loop count is reached, and pass control to the instruction following the F-145 (NEXT) instruction.
Condition	When the input signal is OFF (not limited to ON to OFF change)

[Explanation]



Instruction	
S T R	000001
F-144	
	377
F-63	
	019000
S T R	007357
F-147	
⋮	
F-145	
S T R	000001
O U T	000100

- When the input condition of 007357 is inactive, the instructions between the F-147 (EXIT) and F-145 (NEXT) instructions are executed as NOP instructions, control exits the loop, and control is passed to the instruction following the F-145 instruction.
- When the input condition of 007357 is active, the F-147 (EXIT) instruction does not affect the execution of the loop.

- The F-147 (EXIT) instruction must always be used in a loop between the F-144 (FOR) or F-146 (FORR) and F-145 (NEXT) instructions. Otherwise, execution may result in an error.
- F-147 may be used any number of times in a loop.

F-148
CAL+

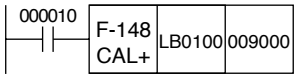
Call subroutine by Label (CAL+)

Symbol	F-148 CAL+ | LBn | S	
Function	Branch program execution to the subroutine with the label (F-140) whose value is the sum of LBn and the contents of register S, and return execution to the main program when the F-143 or F-149 instruction is encountered in the subroutine.	
LBn	0000 to 1370(8) (LSD is always treated as 0)	
S	Use range A	
Condition	Rising edge of input signal (OFF to ON)	
Contents after operation	S	Unchanged
	Flag	Unchanged

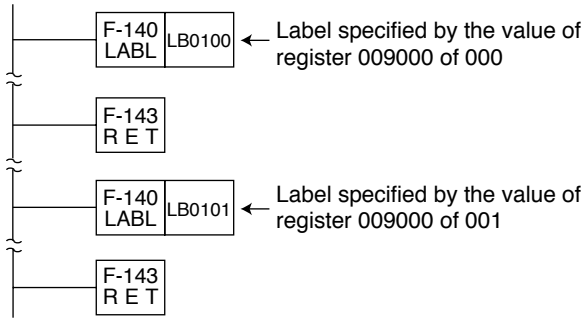
Resembled instructions: F-142

[Explanation]

Instruction	
S T R	000010
F -148	
	LB0100
	009000



When the input condition of 000010 changes from OFF to ON, this instruction calls the subroutine having the label whose value is the sum of LB0100 and the contents of register 009000.

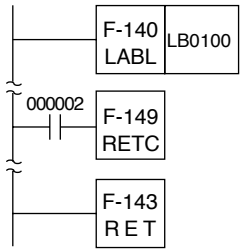


- Register S can have values from 000 to 007(8). If it is set to 000(8) the subroutine call is made to label LBn (LBn+0=LBn).
- The LSD of LBn is always regarded as "0" (e. g. if LBn=0001 is specified, it is treated as LB=0000).
- The active input condition is applied to all instructions within a subroutine (not limited to OFF to ON change).
- The following instructions cannot be used in subroutines: TMR, CNT, F-30 (MCS), F-31 (MCR), F-40 (END), F-41 (JCS), F-42 (JCR), F-44 (⊣↑⊢), F-45 (⊣↓⊢), F-47 (ONLS), F-48 (ONLR)
- Be sure to write the main program before programming a subroutine, and place the F-40 (END) instruction at the last address of the main program.
- If a subroutine is given no label, the program will malfunction.
- The F-148 instruction cannot be used in interrupt program.
- The F-148 (CAL+) , F-143 (RET), and F-149 (RETC) instructions can be used to nest up to 8 subroutine calls. The using method is same as F-142 (CALL) instruction.
- See the notes for the F-142 (CALL) instruction.

F-149 RETC Returns from subroutine (conditional)

Symbol	F-149 RETC
Function	Returns control from the subroutine, which is called by the F-142 (CALL) or F-148 (CAL@) instruction, to the main program.
Condition	When the input signal is OFF (not limited to ON to OFF change)

[Explanation]



Instruction	
F-140	LB0100
STR	000002
F-149	
F-143	

- When the input condition of 000002 is OFF, the instructions between the F-149 (RETC) and F-143 (RET) are executed as NOP instructions, and control is returned from F-143 to the main program.
- When the input condition of 000002 is ON, the F-149 (RETC) instruction is no effect.

- The F-149 instruction must always be used between the F-140 (LABL) and F-143 (RET) instructions. Otherwise, execution may result in an error.
- The F-149 instruction may be used any number of times in a subroutine.

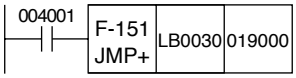
F-151
JMP+

Jump to labeled program address (JuMP+)

Symbol	F-151 JMP+ | LBn | S	
Function	Branch program execution to the program address with the label (F-140) of which value is the sum of LBn and the contents of register S.	
LBn	0000 to 1370(8) (LSD is always treated as 0)	
S	Use range A	
Condition	When the input signal is ON (not to limited to OFF to ON change)	
Contents after operation	S	Unchanged
	Flag	Unchanged

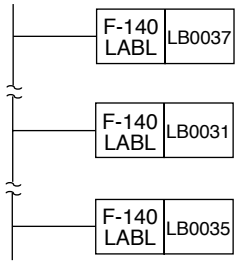
Resembled instructions: F-151

[Explanation]



Instruction	
S T R	004001
F-151	
	LB0030
	019000

When the input condition of 004001 is active, this instruction passes control to the program address having the label (F-140) of which value is the sum of LB0030 and the contents of register 019000.



Label specified by the value of register 019000 of 007(8)

Label specified by the value of register 019000 of 001(8)

Label specified by the value of register 019000 of 005(8)

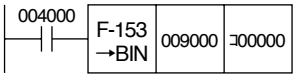
- Register S can have values from 000(8) to 007(8). If it is set to 000(8) the jump is made to label LBn (LBn+0=LBn), and the JW300 shifts to label LBn (F-140) program.
- The LSD of LBn is always regarded as "0" (e. g. if LB0031 is specified, it is treated as LB0030).
- Execution of the F-151 instruction does not affect the contents of the data memory.
- The same label number may be used any number of times for the F-151 instruction.
- Since the F-151 instruction causes control to skip all instructions down to the labeled address, an F-40 (END instruction) placed before the labeled address will be ignored.
- Be sure to give a label (F-140) to the jump destination address. Otherwise the program will malfunction.
- When using the F-151 (JMP+) instruction to cross multiple nested layers specified by the F-142 (CALL) and F-143 (RET) instructions.
  The F-151 (JMP+) instruction can only be used when the program jumps to an address inside the same subroutine, or to jump back to a start line in the same subroutine. The using method is same as F-141 (JMP) instruction.
- When using the F-151 (JMP+) instruction to cross multiple nested layers specified by the F-144 (FOR) and F-145 (NEXT) instructions.
  The F-151 (JMP+) instruction can only be used when the program jumps to an address inside the same subroutine, or to jump back to a start line in the same subroutine. The using method is same as F-141 (JMP) instruction.
- See the notes for the F-141(JMP) instruction.

F-153 →BIN Converts 8-digit BCD to 32-bit binary

Symbol	F-153 →BIN	S	D	
Function	Convert BCD 8 digits data in register S to S+3 (4 bytes) into binary codes, and store them into registers D to D+3 (4 bytes).			
Operation	S, S+1, S+2, S+3 → D, D+1, D+2, D+3			
S	Use range C *			
D	Use range C *			
Condition	Rising edge of input signal (OFF to ON)			
Contents after operation	S, S+1, S+2, S+3	Unchanged		
	D	0 to 255	Unchanged if the contents of S to S+3, are not BCD code.	
	D+1	256 to 65280		
	D+2	65536 to 16711680		
	D+3	16777216 to 99999999		

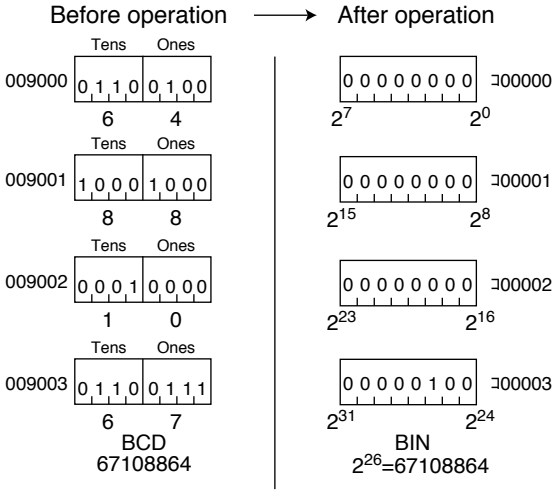
Flag	Contents of S, S+1, S+2, S+3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	BCD	0	0	0	0
	Not BCD code			1	

[Explanation]



Instruction	
S T R	004000
F -153	009000
	□00000

When input condition 004000 changes from OFF to ON, the JW300 converts BCD 8-digit data in register 009000 to 009003 to binary codes and stores them into registers □000000 to □00003.

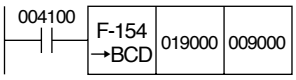


* Be sure to use even addresses for registers S and D. Resembled instructions: F-03, F-03w, F-03d, F-53

F-154 →BCD Converts 32-bit binary to 10-digit BCD

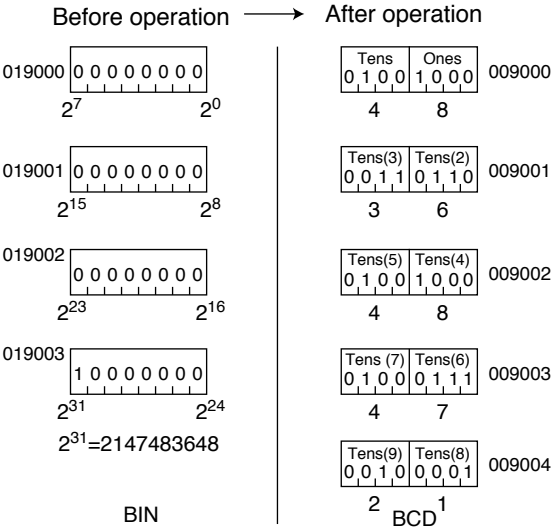
Symbol	F-154 →BCD	S	D	
Function	Convert binary data in register S to S+3 (4 bytes: 32 bits) into BCD codes, and store them into registers D to D+4 (5 bytes).			
Operation	S to S+3 → D to D+4			
S	Use range C *			
D	Use range H *			
Condition	Rising edge of input signal (OFF to ON)			
Contents after operation	S to S+3	Unchanged		
	D	Result (ones and tens)		
	D+1	Result (power 2 of tens and power 3 of tens)		
	D+2	Result (power 4 of tens and power 5 of tens)		
	D+3	Result (power 6 of tens and power 7 of tens)		
	D+4	Result (power 8 of tens and power 9 of tens)		
	Flag	Unchanged		

[Explanation]



Instruction	
S T R	004100
F -154	019000
	009000

When input condition 004100 changes from OFF to ON, the JW300 converts binary data in register 019000 to 019003 (32 bits) to BCD codes and stores them into registers 009000 to 009004.

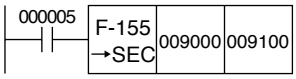


* Be sure to use even addresses for registers S and D. Resembled instructions: F-04, F-04w, F-04d, F-54, F-154

F-155 →SEC Convert hours (BCD 4 digits), minutes (BCD 2 digits), and seconds (BCD 8 digits)

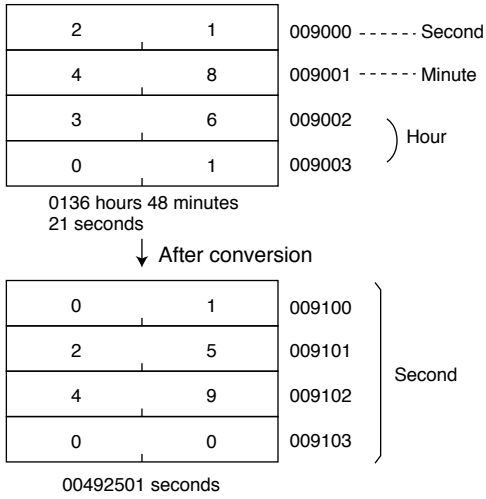
Symbol	F-155 →SEC | S | D					
Function	Convert the time data (4 bytes) in registers S (second), S+1(minute), S+2 (hour: lower 2 digits) and S+3 (hour: upper 2 digits) to second data, and store the result in registers D to D+3 (4 bytes).					
Operation	〈S (second), S+1 (minute), S+2, S+3 (hour)〉 → D to D+3 (second) Max. value = 9999 hours 59 minutes 59 seconds → 35999999 sec					
S	Use range C *					
D	Use range C *					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	S to S+3	Unchanged				
	D to D+3	Result of operation (hr, min., sec.)		If the contents of registers S to S+3 are not BCD or minute/second exceed 60.		
	Flag	Contents of registers S to S+3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		BCD code	0	0	0	0
		- Not BCD code - Greater than max. value			1	

[Explanation]



Instruction	
S T R	000005
F -155	009000
	009100

When the input condition of 000005 changes from OFF to ON, this instruction converts the every time data stored in registers 009100 to 009103 into 1 second data, and stores the result in registers 009100 to 009103 (all data BCD).

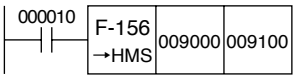


* Be sure to use even addresses for registers S and D.
(Odd addresses such as 019003 etc. are prohibited to use.)

F-156 →HMS Convert seconds (BCD 8 digits) into hours (BCD 4 digits), minutes (BCD 2 digits), and seconds (BCD 2 digits)

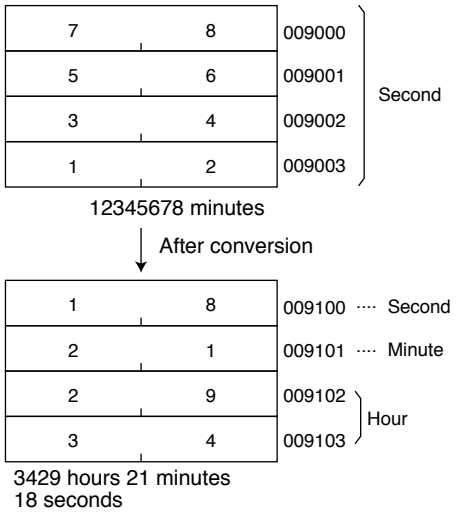
Symbol	F-156 →HMS | S | D					
Function	Convert the second data (BCD 8 digits) stored in registers S to S+3 into hour (BCD 4 digits), minute (BCD 2 digits), and second (BCD 2 digits) data, and store the result in registers D to D+3.					
Operation	(S to S+3) → D (second), D+1 (minute), D+2 and D+3 (hour) Max. value = 35999999 sec. → 9999 hours 59 minutes 59 seconds					
S	Use range C *					
D	Use range C *					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	S to S+3	Unchanged				
	D to D+3	Result of operation (hr, min., sec.)	D to D+3 are unaffected. if the contents of S to S+3 are not BCD or exceed the max. value. (Does not calculate.)			
	Flag	Contents of registers S to S+3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		BCD code	0	0	0	0
		- Not BCD code - Greater than max. value	0	0	1	0

[Explanation]



Instruction	
S T R	000010
F -156	009000
	009100

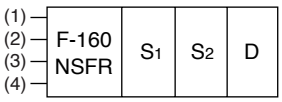
When the input condition of 000010 changes from OFF to ON, this instruction converts the second data stored in registers 009001 to 009003 into hour, minute, and second data, and stores the result in registers 009100 to 009103. (all data BCD)



* Be sure to use even addresses for registers S and D.
(Odd addresses such as 019003 etc. are prohibited to use.)

F-160 NSFR — Shift "n" bit register (N bit ShiFt Register)

Symbol



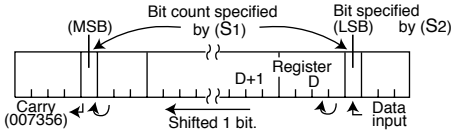
(1) Shift direction indication input
(2) Data input
(3) Shift input
(4) Reset input

Function

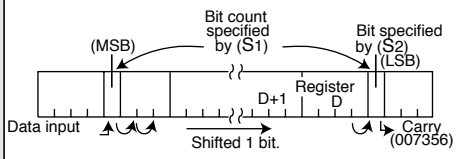
Shift the specified bits area (register S_1) from the specified bit by register D (register S_2) to 1 bit higher or lower in the direction specified by shift direction indication input (1).

Operation

- When the shift direction indication input (1) is ON:



- When the shift direction indication input (1) is OFF:



S_1

Use range A *
- The contents of register S_1 are 000 to 255(D). If set to 000(D), 256 bits of data are transferred.

S_2

Use range A *
- The contents of register S_2 are 000 to 007(D).

D

Use range A *

Condition

When the reset input (4) is OFF, shift occurs at the rising edge (OFF to ON) of the shift input (3).

Contents after operation

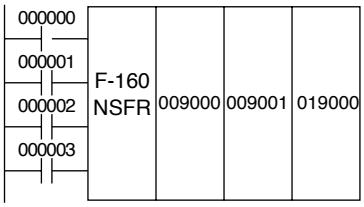
Shift area

- Result of operation when reset input (4) is OFF.
- All bits OFF when reset input (4) is ON.

Flag

Reset input (4)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
OFF	0	0 or 1	0	0
ON		0		

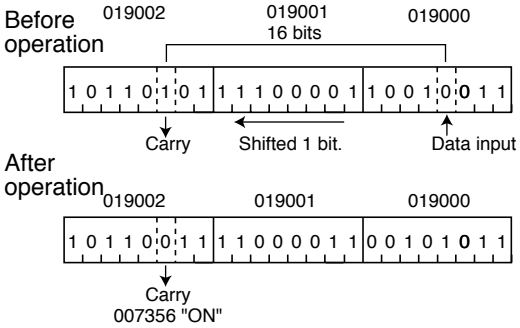
[Explanation]



Instruction	
S T R	000000
S T R	000001
S T R	000002
S T R	000003
F -160	
	009000
	009001
	019000

- The value of register 009000 is 016(D) bits
- The value of register 009001 is 003(D) bits
- Direction instruction input 000000 is ON
- Data input 0000001 is ON
- Reset input 000003 is OFF;

in this example, the following shift operation occurs when input (000002) changes from OFF to ON.

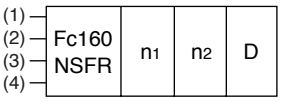


* Be careful that the shift area set by S_1, S_2, and D does not enter the contact point areas of the timer/counter (file address 00001600 to 00001777(8) etc.) or the last address or more of the registers and file registers.

Resembled instructions: F-60, F-60w, F-60d, Fc160

Fc160 NSFR — Shift "n" bit register (N bit ShiFt Register)

Symbol

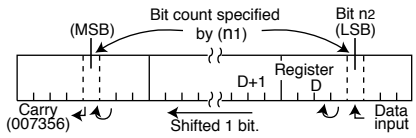


Function

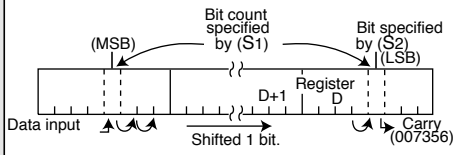
Shift the specified shift area (of which bit span is specified by n1 and whose LSB is specified by n2) of register D (bit n2), 1 bit in the direction specified by the shift direction indication input (1).

Operation

- When the shift direction indication input (1) is ON:



- When the shift direction indication input (1) is OFF:



n1

Use range 000 to 255(D)
- If set to 000(D), 256 bits of data are transferred.

n2

Use range 000 to 007(D)

D

Use range A *

Condition

When the reset input (4) is OFF, shift occurs at the rising edge (OFF to ON) of the shift input (3).

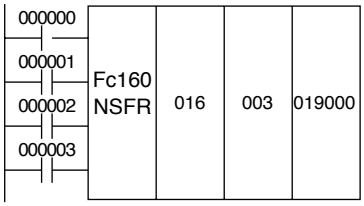
Contents after operation

Shift area:
- Result of operation when reset input (4) is OFF.
- All bits OFF when reset input (4) is ON.

Flag:

Reset input (4)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
OFF	0	0 or 1	0	0
ON		0		

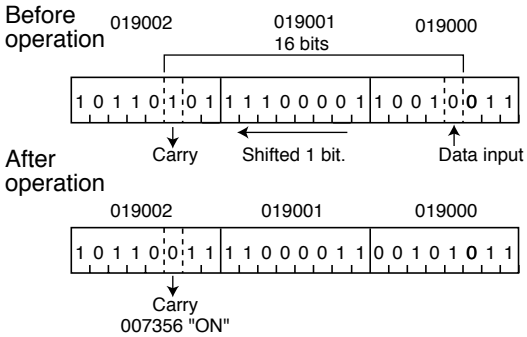
[Explanation]



Instruction	
S T R	000000
S T R	000001
S T R	000002
S T R	000003
F c160	
	016
	003
	019000

- The shift area is 016(D) bits
- The data input is at bit 003(D)
- Direction instruction input 000000 is ON
- Data input 0000001 is ON
- Reset input 000003 is OFF

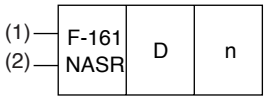
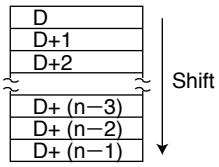
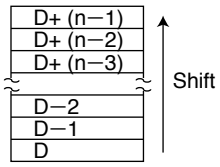
in this example, the following shift operation occurs when input (000002) changes from OFF to ON.



* Be careful that the shift area set by D is not enter the contact point areas of the timer/counter (file address 00001600 to 00001777(8) etc.) or the last address or more of the registers and file registers.

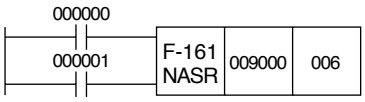
Resembled instructions: F-60, F-60w, F-60d, F-160

F-161 NASR — Bi-directional asynchronous shift register (N byte) (N byte Asynchronous Shift Register)

Symbol	 (1) Shift direction indication input (2) Shift input
Function	In the range of "register D to D+(n-1)" or "register D - (n-1) to D," shift data of just before or after register (1 byte) to a register (1 byte) having data $00_{(H)}$.
Operation	- When the shift direction indication input (1) is ON. Shift occurs in the direction from register D toward D+n.  - When the shift direction indication input (1) is OFF. Shift occurs in the direction from register D toward D−n.  - Once a shift operation is executed, the contents of register (1 byte) are cleared to "0". - No shift operation is performed if no register with all null data ($00_{(H)}$) exists in the shift area. - The shift area that is set by D and n shall not be before file address $00000000_{(8)}$, in the contact point area of counter/timer (00001600 to $00001777_{(8)}$ etc.), before file register $000000000_{(8)}$, or after the last address of the register and file register.
D	Use range A
n	Use range $000_{(8)}$ to $377_{(8)}$ bytes (256 bytes for $000_{(8)}$)
Condition	Shift occurs when the shift input (2) is ON (not limited to OFF to ON change) - Shift operation is performed in every scan cycle as long as the shift input (2) is ON.

		(1) ON		(1) OFF		(1) ON/OFF	
		Before operation	After operation	Before operation	After operation	Before operation	After operation
Contents of just before 0		DATA1	0	DATA1	Same as left	DATA1	Same as left
Contents of 0		0	DATA1	0	DATA2	Other than 0	Same as left
Contents of just after 0		DATA2	Same as left	DATA2	0	DATA2	Same as left
Flag * (After operation)	Non-Carry 007354	1 : D+ (n−1) =0 0 : D+ (n−1) ≠0		1 : D− (n−1) =0 0 : D− (n−1) ≠0		1	
	Error 007355	0		0		0	
	Carry 007356	0 : D+ (n−1) =0 1 : D+ (n−1) ≠0		0 : D− (n−1) =0 1 : D− (n−1) ≠0		0	
	Zero 007357	0		0		0	

[Explanation]

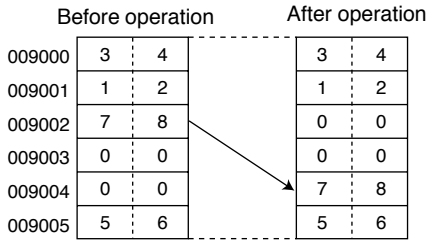


Instruction	
S T R	000000
S T R	000001
F -161	009000
	006

000000 (1) ON ··· Shift direction
000001 (2) ON ··· 009000 to 009005 are shifted.

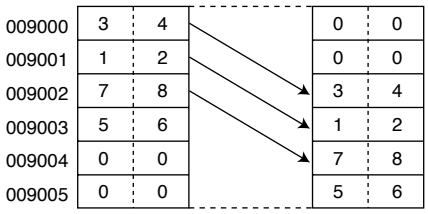
* Carry flag (007356) turns ON only when data other than 0 shift to "D+(n-1)" or "D-(n-1)."
Resembled instructions: F-61, F-61w, F-61d, F-161w, F-161d

(1) When the values of 009003 and 009004 are $00_{(H)}$:
With the above input conditions, this instruction yields the following shift result :

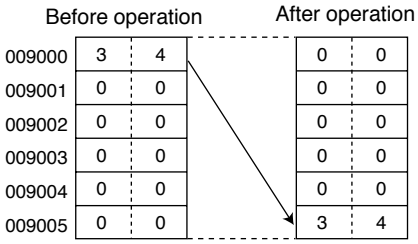


Only the non-carry flag (007354) is set ON.

- To shift all 3 bytes in example (1) above, 3 scan cycles are needed:

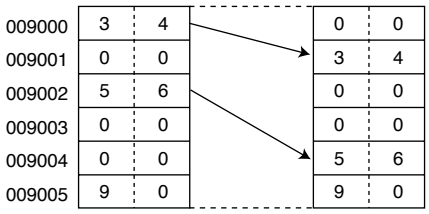


(2) When the contents of 009001 to 009005 are all $00_{(H)}$:
A single execution of this instruction shifts the data from register D (009000) to 009005.



Only the carry flag (007356) is set ON.

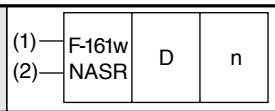
- If the data item that precedes another data item to be shifted is "$00_{(H)}$", a single execution cycle causes more than one data item to be shifted:



F-161w NASR

Bi-directional asynchronous shift register (N words) (N byte Asynchronous Shift Register)

Symbol



(1) Shift direction indication input
(2) Shift input

Function

In the range of "register D to D+2(n-1)+1" or "register D−2(n-1) to D+1," shift data of just before or after register (1 word) to a register (1 word) having data 0000(H).

Operation

- When the shift direction indication input (1) is ON. Shift occurs in the direction from register D toward D+2n.

D+1	D
D+3	D+2
D+5	D+4
≈	≈
D+2 (n−3) +1	D+2 (n−3)
D+2 (n−2) +1	D+2 (n−2)
D+2 (n−1) +1	D+2 (n−1)

Shift ↓

- When the shift direction indication input (1) is OFF. Shift occurs in the direction from register D toward D−2n.

D−2 (n−1) +1	D−2 (n−1)
D−2 (n−2) +1	D−2 (n−2)
D−2 (n−3) +1	D−2 (n−3)
≈	≈
D−3	D−4
D−1	D−2
D+1	D

Shift ↑

- Once a shift operation is executed, the contents of registers (1 word) is cleared to "0".
- No shift operation is performed if no register with all null data (0000(H)) exists in the shift area.
- The shift area that is set by D and n shall not be before file address 00000000(8), in the contact point area of counter/timer (00001600 to 00001777(8) etc.), before file register 000000000(8), or after the last address of the register and file register.

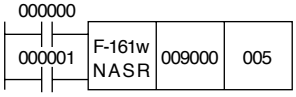
D: Use range B

n: Use range 000 to 377(8) (256 words for 000(8))

Condition: Shift occurs when the shift input (2) is ON (not limited to OFF to ON change)
- Shift operation is performed in every scan cycle as long as the shift input (2) is ON.

		(1) ON		(1) OFF		(1) ON/OFF	
		Before operation	After operation	Before operation	After operation	Before operation	After operation
Contents of just before 0		DATA1	0	DATA1	Same as left	DATA1	Same as left
Contents of 0		0	DATA1	0	DATA2	Other than 0	Same as left
Contents of just after 0		DATA2	Same as left	DATA2	0	DATA2	Same as left
Flag * (After operation)	Non-Carry 007354	1 (D+2 (n−1) +1, D+2 (n−1) =0) 0 (D+2 (n−1) +1, D+2 (n−1) ≠0)		1 (D−2 (n−1) +1, D−2 (n−1) =0) 0 (D−2 (n−1) +1, D−2 (n−1) ≠0)		1	
	Error 007355	0		0		0	
	Carry 007356	0 (D+2 (n−1) +1, D+2 (n−1) =0) 1 (D+2 (n−1) +1, D+2 (n−1) ≠0)		0 (D−2 (n−1) +1, D−2 (n−1) =0) 1 (D−2 (n−1) +1, D−2 (n−1) ≠0)		0	
	Zero 007357	0		0		0	

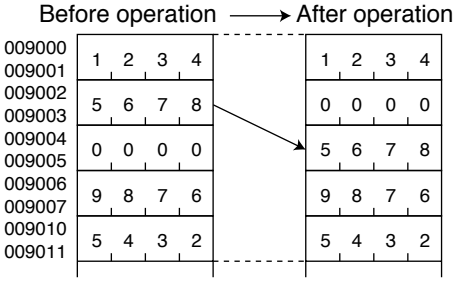
[Explanation]



Instruction	
S T R	000000
S T R	000001
F -161w	009000
	005

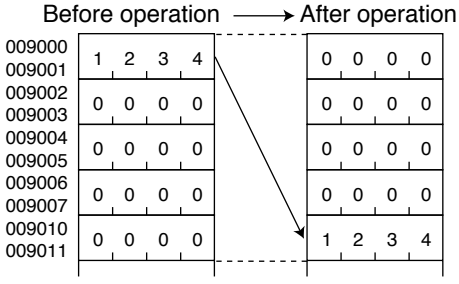
000000 (1) ON ... 009000 to 009011 are shifted.
000001 (2) ON ... Shift direction

(1) When the values of 009004 and 009005 are 0000(H): With the above input conditions, this instruction yields the following shift result:



Only the non-carry flag (007354) is set ON.

(2) When the contents of 009002 to 009011 are all 0000(H), a single execution cycle of this instruction shifts the data in 009000, 009001 to 009010 and 009011.

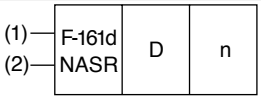


Only the carry flag (007356) is set ON.

* Carry flag (007356) turns ON only when data other than 0 shift to "D+2 (n-1)+1, D+2 (n-1)" or "D-2 (n-1)+1, D-2 (n-1)." Resembled instructions: F-61, F-61w, F-61d, F-161, F-161d

F-161d NASR Bi-directional asynchronous shift register (N double words) (N byte Asynchronous Shift Register)

Symbol



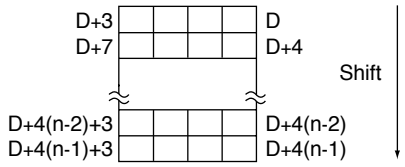
(1) Shift direction indication input
(2) Shift input

Function

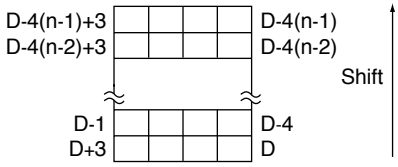
In the range of "register D to D+4(n-1)+3" or "register D-4 (n-1) to D+3," shift data of just before or after register (2 words) to a register (2 words) having data 0.

Operation

- When the shift direction input (1) is ON. Shift occurs in the direction from register D toward D+4n.



- When the shift direction input (1) is OFF. Shift occurs in the direction from register D toward D-4n.



- Once a shift operation is executed, the contents of registers (2 words) is cleared to "0".
- No shift operation is performed if no register with all null data 0 (2 words) exists in the shift area.
- The shift area that is set by D and n shall not be before file address 00000000(8), in the contact point area of counter/timer (00001600 to 00001777(8) etc.), before file register 000000000(8), or after the last address of the register and file register.

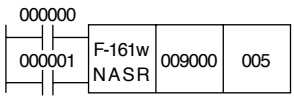
D: Use range C

n: Use range 000 to 377(8) double words (256 double words for 000(8))

Condition: Shift occurs when the shift input (2) is ON (not limited to OFF to ON change)
- Shift operation is performed in every scan cycle as long as the shift input (2) is ON.

		(1) ON		(1) OFF		(1) ON/OFF	
		Before operation	After operation	Before operation	After operation	Before operation	After operation
Contents of just before 0		DATA1	0	DATA1	Same as left	DATA1	Same as left
Contents of 0		0	DATA1	0	DATA2	Other than 0	Same as left
Contents of just after 0		DATA2	Same as left	DATA2	0	DATA2	Same as left
Flag * (After operation)	Non-Carry 007354	1 (D+4(n−1) to D+4(n−1)+3 =0) 0 (D+4(n−1) to D+4(n−1)+3 ≠0)		1 (D−4 (n−1) to D−4 (n−1)+3 =0) 0 (D−4 (n−1) to D−4 (n−1)+3 ≠0)		1	
	Error 007355	0		0		0	
	Carry 007356	0 (D+4(n−1) to D+4(n−1)+3 =0) 1 (D+4(n−1) to D+4(n−1)+3 ≠0)		0 (D−4 (n−1) to D−4 (n−1)+3 =0) 1 (D−4 (n−1) to D−4 (n−1)+3 ≠0)		0	
	Zero 007357	0		0		0	

[Explanation]



Instruction	
S T R	000000
S T R	000001
F -161d	
	009000
	005

000000 (1) ON ... 009000 to 009011 are shifted.
000001 (2) ON ... Shift direction

When the values of 009010 to 009017 are 0: With the above input conditions, this instruction yields the following shift result:

	Before operation								After operation							
009000 to 009003	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
009004 to 009007	9	A	B	C	D	E	F	1	0	0	0	0	0	0	0	0
009010 to 009013	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
009014 to 009017	0	0	0	0	0	0	0	0	9	A	B	C	D	E	F	1
009020 to 009023	7	6	5	4	3	2	1	1	7	6	5	4	3	2	1	1

Only the non-carry flag (007354) is set ON.

* Carry flag (007356) turns ON only when data other than 0 shift to "D+4(n-1) to D+4(n-1)+3" or "D-4(n-1) to D-4(n-1)+3." Resembled instructions: F-61, F-61w, F-61d, F-161, F-161w

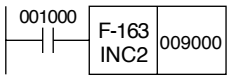
F-163 INC2 Adds binary (+2) counter (1 byte)

Symbol		F-163 INC2 D
Function		Binary contents of the register D are incremented by 2.
Operation		〈D〉+2 → D
D		Use range A
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	D	Result
	Flag	Result (OCT) / Zero 007357 / Carry 007356 / Error 007355 / Non-carry 007354

Result (OCT)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
376→000	1	1	0	0
377→001	0	1	0	0
Other than above	0	0	0	1

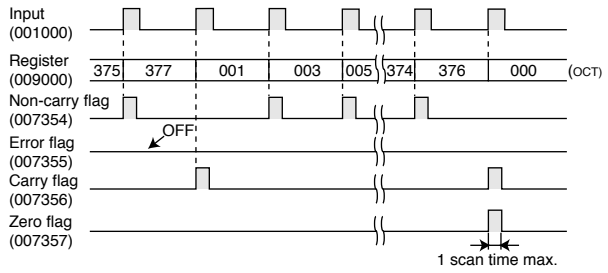
Resembled instructions: F-63, F-63w, F-63d, F-163w, F-163d, F-263, F-263w, F-263d

[Explanation]



Instruction	
S T R	001000
F -163	009000

When the input condition 001000 changes from OFF to ON, the binary contents of the register 009000 are incremented by 2.



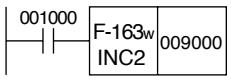
F-163w INC2 Adds binary (+2) counter (1 word)

Symbol		F-163w INC2 D
Function		Binary contents of the register D and D+1 are incremented by 2.
Operation		〈D, D+1〉 +2 → D, D+1
D		Use range B - Be sure to use even addresses for register D. (Odd addresses such as 019003 etc. are prohibited to use.)
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	D	Lower digits of result
	D+1	Upper digits of result
	Flag	Result (OCT) / Zero 007357 / Carry 007356 / Error 007355 / Non-carry 007354

Result (OCT)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
177776→00000	1	1	0	0
177777→00001	0	1	0	0
Other than above	0	0	0	1

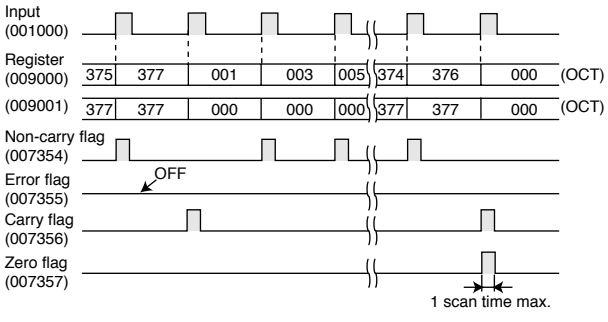
Resembled instructions: F-63, F-63w, F-63d, F-163, F-163d, F-263, F-263w, F-263d

[Explanation]



Instruction	
S T R	001000
F -163w	009000

When the input condition 001000 changes from OFF to ON, the 16-bit binary contents of the registers 009000 and 009001 are incremented by 2.



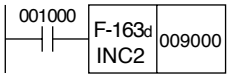
F-163d INC2 Adds binary (+2) counter (2 words)

Symbol	F-163d INC2 | D
Function	Binary contents of the register D to D+3 are incremented by 2.
Operation	〈D to D+3〉 +2 → D to D+3
D	Use range C - Be sure to use even addresses for register D. (Odd addresses such as 019003 etc. are prohibited to use.)
Condition	Rising edge of input signal (OFF to ON)

Contents after operation: D to D+3 — D: Lower digits, D+3: Upper digits

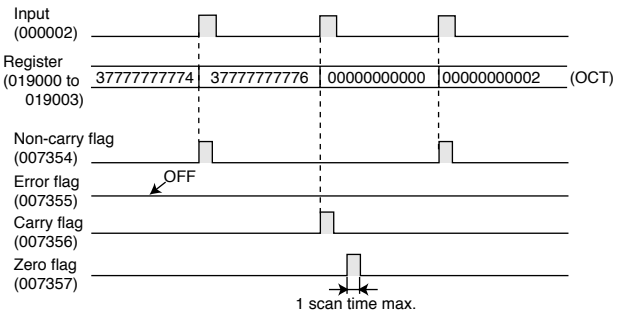
Flag: Result (OCT)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
37777777776 →00000000000	1	1	0	0
37777777777 →00000000001	0	1	0	0
Other than above	0	0	0	1

[Explanation]



Instruction	
S T R	001000
F -163d	009000

When the input condition 001000 changes from OFF to ON, the 32-bit binary contents of the registers 009000 to 009003 are incremented by 2.



Resembled instructions: F-63, F-63w, F-63d, F-163, F-163w, F-263, F-263w, F-263d

F-164 DEC2 Subtracts binary (-2) counter (1 byte)

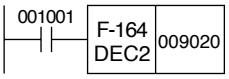
Symbol	F-164 DEC2 D
Function	Binary contents of the register D are decremented by 2.
Operation	〈D〉—2 → D
D	Use range A
Condition	Rising edge of input signal (OFF to ON)

Contents after operation

D	Result				
Flag	Result (OCT)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	002→000	1	0	0	1
	001→377 000→376	0	1	0	0
	Other than above	0	0	0	1

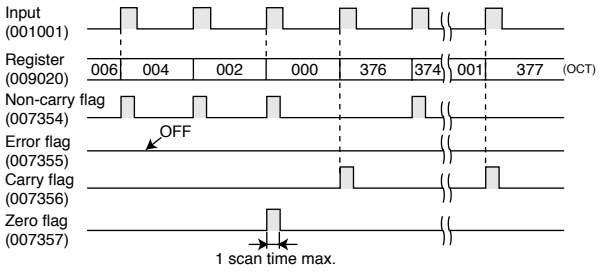
Resembled instructions: F-64, F-64w, F-64d, F-164w, F-164d, F-264, F-264w, F-264d

[Explanation]



Instruction	
S T R	001001
F -164	009020

When the input condition 001001 changes from OFF to ON, the binary contents of the register 009020 are decremented by 2.



F-164w DEC2 Subtracts binary (-2) counter (1 word)

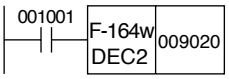
Symbol	F-164w DEC2 D
Function	Binary contents of the registers D and D+1 are decremented by 2.
Operation	〈D, D+1〉—2 → D, D+1
D	Use range B - Be sure to use even addresses for register D. (Odd addresses such as 019003 etc. are prohibited to use.)
Condition	Rising edge of input signal (OFF to ON)

Contents after operation

D	Lower digits of result				
D+1	Upper digits of result				
Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	000002→00000	1	0	0	1
	000001→17777 000000→17776	0	1	0	0
	Other than above	0	0	0	1

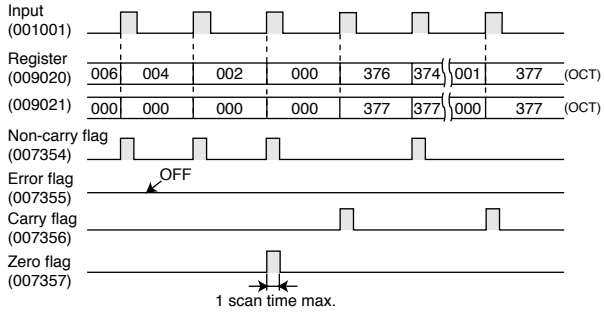
Resembled instructions: F-64, F-64w, F-64d, F-164, F-164d, F-264, F-264w, F264d

[Explanation]



Instruction	
S T R	001001
F -164w	009020

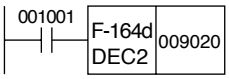
When the input condition 001001 changes from OFF to ON, the 16-bit binary contents of the registers 009020 and 009021 are decremented by 2.



F-164d DEC2 Subtracts binary (-2) counter (2 words)

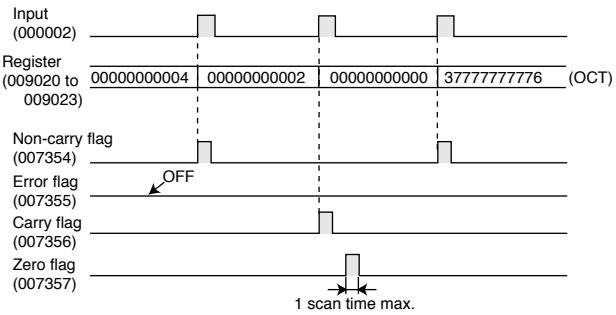
Symbol	F-164d DEC2 D					
Function	Binary contents of the registers D to D+3 are decremented by 2.					
Operation	〈D to D+3〉−2 → D to D+3					
D	Use range C - Be sure to use even addresses for register D. (Odd addresses such as 019003 etc. are prohibited to use.)					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	D to D+3	D : Lower digits, D+3 : Upper digits				
	Flag	Result (OCT)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		00000000002→ 00000000000	1	0	0	1
		00000000001→ 37777777777 00000000000→ 37777777776	0	1	0	0
		Other than above	0	0	0	1

[Explanation]



Instruction	
S T R	001001
F -164d	009020

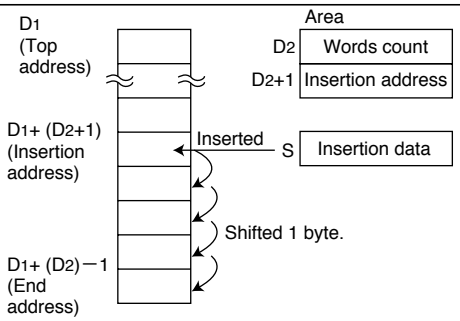
When the input condition 001001 changes from OFF to ON, the 32-bit binary contents of the registers 009020 to 009023 are decremented by 2.



Resembled instructions: F-64, F-64w, F-64d, F-164, F-164w, F-264, F-264w, F-264d

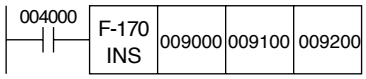
F-170
INS

Inserts of data (1 byte) (INSert)

Symbol	F-170 INS | S | D_1 | D_2
Function	Insert the contents of register S into register address D_1+ (D_2+1) in the shift register area whose first register address is D_1 and of which byte count is given by the contents of register D_2.
Operation	 - The shift register area that is set by D_1 and D_2 shall not be in the contact point area of counter/timer (file address 00001600 to 00001777(8) etc.), after the last address of the register and file register. - If the end address is not 00(H), or D_2 ≦ D_2+1, an error flag (007355) turns ON and JW300 does not operate. - If the contents of the last address are not 00(H), the instruction is not executed even if the contents of intermediate addresses are 00(H). Intermediate addresses 00(H) are treated as data.
S	Use range A
D_1	Use range A
D_2	Use range B - The contents of register D_2 are 000 to 377(8). 000(8) specifies 256 bytes. - The contents of register D_2+1 are 000 to 377(8). 000(8) specifies the top address.
Condition	Rising edge of input signal (OFF to ON)

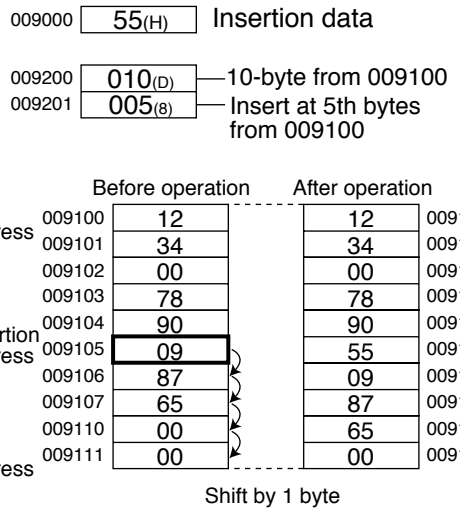
Contents after operation						
	S	Unchanged				
	D_2, D_2+1	Unchanged				
	D_1 to D_1+(D_2)−1	1 byte shifted and value of S inserted. - Does not chanded when the contents of the end address are not "00(H)" or D_2 ≦				
	Flag	Contents of end address, etc.	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		Contents of end address 00(H)	0	0	0	0
		Contents of end address not 00(H) or D_2 ≦ D_2+1			1	

[Explanation]

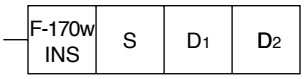
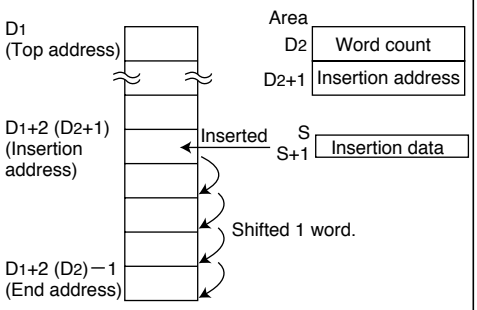


Instruction	
S T R	004000
F -170	
	009000
	009100
	009200

- Calculates when input condition 004000 changes from OFF to ON.
- The contents (55(H)) of registers 009000 is inserted. The top address is register 009100.
- Address to insert is one added 005(8) bytes (contents of 009201) to register 009100.
- The end address is 010(D) byte (contents of 009200) for 009100. The top address is included in the byte count.
- When the contents of the last address are 00(H), the data at and beyond the address of insertion are shifted down 1 byte, and the contents (55(H)) of address 009000 are stored in the insertion address.

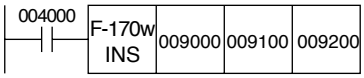


F-170w INS — Inserts data (1 word) (INSert)

Symbol	
Function	Insert the contents of register S, S+1 into register address D1+2 (D2+1), D1+2(D2+1)+1 in the shift register area whose first register address is D1 and of which word count is given by the contents of register D2.
Operation	 - The shift register area that is set by D1 and D2 shall not be in the contact point area of counter/timer (file address 00001600 to 00001777(8) etc.), after the last address of the register and file register. - If the end address is not 0000(H), or D2 ≦ D2+1, an error flag (007355) turns ON and JW300 does not operate. - If the contents of the last address are not 0000(H), the instruction is not executed even if the contents of intermediate addresses are 0000(H). Intermediate addresses 0000(H) are treated as data.
S	Use range B - Be sure to use even addresses for register S.
D1	Use range B - Be sure to use even addresses for register D1.
D2	Use range B - The contents of register D2 are 000 to 377(8). 000(8) specifies 256 words. - The contents of register D2+1 are 000 to 377(8). 000(8) specifies the top address.
Condition	Rising edge of input signal (OFF to ON)

Contents after operation						
S	Unchanged					
D1, D2+1	Unchanged					
D1 to D1+2 (D2) −1	1 word shifts and value of S, S+1 inserted. - Unaffected when the contents of the end address are not "0000(H)" or D2 ≦ D2+1.					
Flag	Contents of end address, etc.	Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
	Contents of end address 0000(H)	0	0	0	0	
	Contents of end address not 0000(H) or D2 ≦ D2+1			1		

[Explanation]

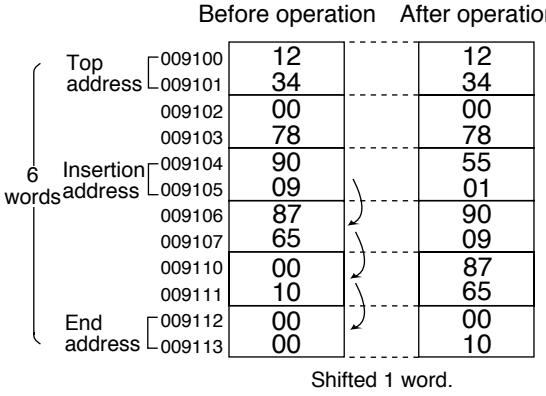


Instruction	
S T R	004000
F -170w	
	009000
	009100
	009200

- Calculates when input condition 004000 changes from OFF to ON.
- The contents (0155) of registers 009000 and 009001 are inserted. The top address is register 009100.
- Address to insert is one added 002(8) words (contents of 009201) to register 009100.
- The end address is 006(8) words (contents of 009200) for 009100. The top address is included in the word count.
- When the contents of the end address are 0000(H), the data at and beyond the address of insertion are shifted down 1 word, and the contents (0155) of address 009000 and 009001 are stored in the insertion address.

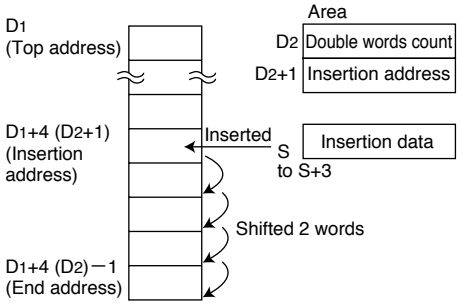
009000 55 (low)
009001 01 (high) Insertion data

009200 006 (8) → 6 words from 009100
009201 002 (8) → Insert at 2nd words from 009100



Shifted 1 word.

F-170d INS — Inserts data (2 words) (INSert)

Symbol	F-170d INS | S | D_1 | D_2
Function	Insert the contents of register S to S+3 into register address D_1+4 (D_2+1) to D_1+4 (D_2+1)+3 in the shift register area whose first register address is D_1 and of which double words count is given by the contents of register D_2.
Operation	 - The shift register area that is set by D_1 and D_2 shall not be in the contact point area of counter/timer (00001600 to 00001777(8) etc.), after the last address of the register and file register. - If the end address is not 00000000(H), or $D_2 \leqq D_2+1$, an error flag (007355) turns ON and JW300 does not operate. - If the contents of the end address are not 0, the instruction is not executed even if the contents of intermediate addresses are 0 at intermediate addresses are treated as data.
S	Use range C - Be sure to use even addresses for register S.
D_1	Use range C - Be sure to use even addresses for register D_1.
D_2	Use range B - The contents of register D_2 are 000 to 377(8). (000(8) specifies 256 double words.) - The contents of register D_2+1 are 000 to 377(8). (000(8) specifies the top address.)
Condition	Rising edge of input signal (OFF to ON)

Contents after operation						
	S to S+3	Unchanged				
	D2, D2+1	Unchanged				
	D1 to D1+4 (D2) −1	1 word shifts and value of S to S+3 contents inserted. - Unaffected when the contents of the end address are not "0" or $D_2 \leqq D_2+1$.				
	Flag	Contents of end address, etc.	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		Contents of end address 0	0	0	0	0
		Contents of end address not 0 or $D_2 \leqq D_2+1$			1	

[Explanation]

Instruction	
S T R	004000
F -170d	009000
	009100
	009200

004000 | F-170d INS | 009000 | 009100 | 009200

- Calculates when input condition 004000 changes from OFF to ON.
- The contents (46320155(H)) of registers 009000 to 009003 are inserted. The top address is register 009100.
- Address to insert is one added 002(8) double words (contents of 009201) to register 009100.
- The end address is 006(8) double words (contents of 009100) for 009100. The top address is included in the words count.
- When the contents of the end address are 0, the data at and beyond the address of insertion are shifted down 2 words, and the contents of address 009000 to 009003 are stored in the insertion address.

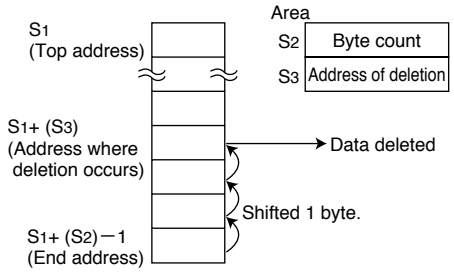
009000	55 (low)	Insertion data
009001	01	
009002	32	
009003	46 (high)	
009200	006 (8)	→ 6 double words from 009100
009201	002 (8)	→ Insert at 2nd double words from 009100

Before operation → After operation

			Before operation	After operation
Top address	009100		12	12
	009101		34	34
	009102		00	00
	009103		78	78
	009104		23	23
	009105		34	34
	009106		45	45
	009107		67	67
Insertion address	009110		52	55
	009111		76	01
	009112		38	32
	009113		14	46
6 double words	009114		37	52
	009115		52	76
	009116		18	38
	009117		45	14
	009120		27	37
	009121		68	52
	009122		51	18
	009123		39	45
End address	009124		00	27
	009125		00	68
	009126		00	51
	009127		00	39

Shifted 2 words.

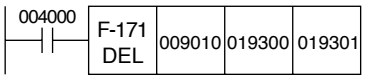
F-171 DEL — Deletes of data (1 byte) (DELete)

Symbol	F-171 DEL | S_1 | S_2 | S_3
Function	Delete 1 byte of data from address S_1+(S_3) in the shift register area of which top register address is S_1 and of which byte count is given by the contents of register S_2. All the data beyond the deleted data are shifted up 1 byte position.
Operation	 - The shift register area that is set by S_2 and S_3 shall not be in the contact point area of counter/timer (file addresses 00001600 to 00001777(8) etc.), after the last address of the register and file register. - If the contents of register is $S_2 \leqq S_3$, an error flag (007355) turns ON and JW300 does not operate. - The instruction is executed even if the contents of intermediate addresses are 00(H) at intermediate addresses are treated as data.
S_1	Use range A
S_2	Use range A - The contents of register S_2 are 000 to 377(8). (000(8) specifies 256 bytes.)
S_3	Use range A - The contents of register S_3 are 000 to 377(8). (000(8) specifies the top address.)
Condition	Rising edge of input signal (OFF to ON)

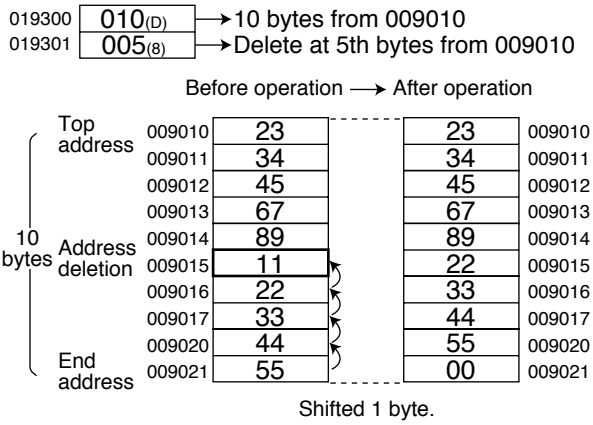
Contents after operation						
	S_2	Unchanged				
	S_3	Unchanged				
	S_1 to S_1+(S_2) −1	Data deleted and subsequent data shifted 1 byte. - Unaffected if $S_2 \leqq S_3$.				
	Flag	Contents of S_2, S_3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		If $S_2 > S_3$	0	0	0	0
		If $S_2 \leqq S_3$			1	

[Explanation]

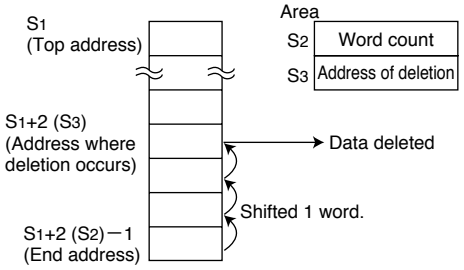
Instruction	
S T R	004000
F -171	009010
	019300
	019301



- Calculates when input condition 004000 changes from OFF to ON.
- The top address is register 009010.
- The address of deletion has an offset from address 009010 given by the contents (005(8) bytes) of register 019300.
- The end address has an offset from address 009010 given by the contents (010(D) bytes) of register 019300. The top address is counted in the byte count.
- When the JW300 executes calculation, it erases the deleted address, and shifts data from the deleted address to the end address. The data of the end address become 00(H).

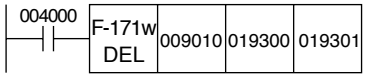


F-171w DEL — Deletes data (1 word) (DELete)

Symbol	F-171w DEL | S_1 | S_2 | S_3
Function	Delete 1 word of data from addresses S_1+2 (S_3) and S_1+2 (S_3) +1 in the shift register area of which top register address is S_1 and of which word count is given by the contents of register S_2. All the data beyond the deleted data are shifted up 1 word position.
Operation	 - The shift register area that is set by S_2 and S_3 shall not be in the contact point area of counter/timer (file addresses 00001600 to 00001777(8) etc.), after the last address of the register and file register. - If the contents of register is $S_2 \leqq S_3$, an error flag (007355) turns ON and JW300 does not operate. - The instruction is executed even if the contents of intermediate addresses are 0000(H) at intermediate addresses are treated as data.
S_1	Use range C - Be sure to use even addresses for register S_1.
S_2	Use range A - The contents of register S_2 are 000 to 377(8). (000(8) specifies 256 words.)
S_3	Use range A - The contents of register S_3 are 000 to 377(8). (000(8) specifies the top address.)
Condition	Rising edge of input signal (OFF to ON)

Contents after operation						
	S_2	Unchanged				
	S_3	Unchanged				
	S_1 to S_1+2 (S_2) −1	Data deleted and subsequent data shifted 1 word. - Unaffected if $S_2 \leqq S_3$.				
	Flag	Contents of S_2, S_3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		If $S_2 > S_3$	0	0	0	0
		If $S_2 \leqq S_3$			1	

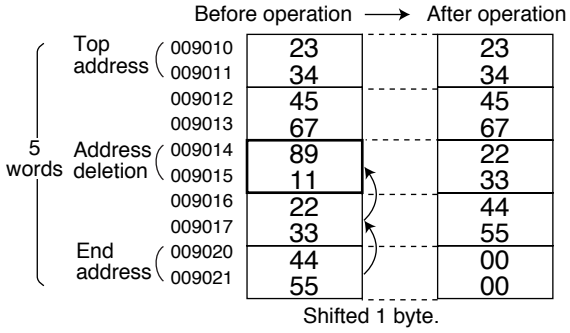
[Explanation]



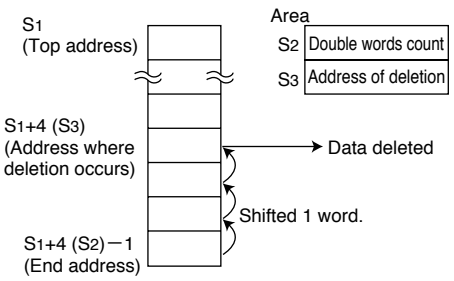
Instruction	
S T R	004000
F -171w	009010
	019300
	019301

- Calculates when input condition 004000 changes from OFF to ON.
- The top address is register 009010.
- The address of deletion has an offset from address 009010 given by the contents (002(8) words) of register 019301.
- The end address has an offset from address 009010 given by the contents (005(8) words) of register 019300. The top address is counted in the word count.
- When the JW300 executes calculation, it erases the deleted address, and shifts data from the deleted address to the end address. The word of the end address become 0000(H).

019300	005(D)	→ 5 words from 009010
019301	002(8)	→ Delete at 2nd word from 009010

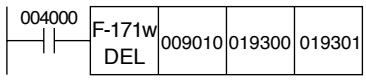


F-171d DEL — Deletes data (2 words) (DELete)

Symbol	F-171d DEL | S_1 | S_2 | S_3
Function	Delete 1 word of data from addresses S_1+4 (S_3) to S_1+4 (S_3) +3 in the shift register area of which top register address is S_1 and of which double words count is given by the contents of register S_2. All the data beyond the deleted data are shifted up 1 word position.
Operation	 - The shift register area that is set by S_2 and S_3 shall not be in the contact point area of counter/timer (file addresses 00001600 to 00001777(8) etc.), after the last address of the register and file register. - If the contents of register is $S_2 \leqq S_3$, an error flag (007355) turns ON and JW300 does not operate. - The instruction is executed even if the contents of intermediate addresses are 00000000(H) at intermediate addresses are treated as data.
S_1	Use range C - Be sure to use even addresses for register S_1.
S_2	Use range A - The contents of register S_2 are 000 to 377(8). (000(8) specifies the top address.)
S_3	Use range A - The contents of register S_3 are 000 to 377(8). (000(8) specifies the top address.)
Condition	Rising edge of input signal (OFF to ON)

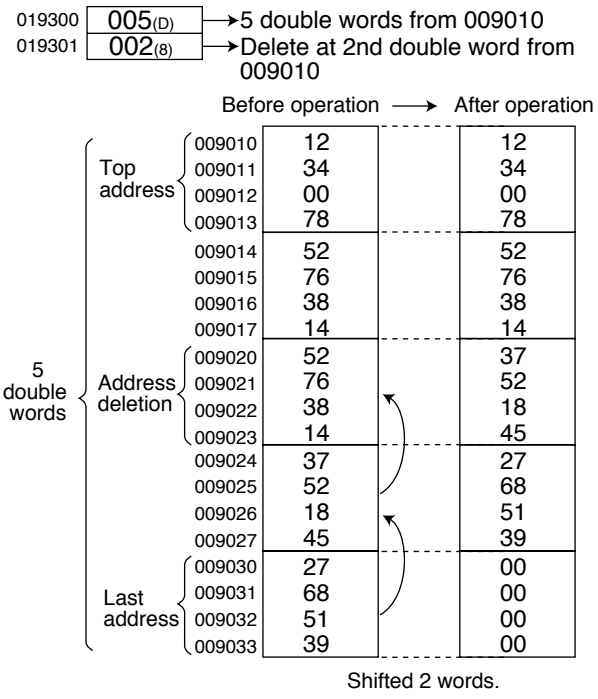
Contents after operation						
	S_2	Unchanged				
	S_3	Unchanged				
	S_1 to S_1+4 (S_2) −1	Data deleted and subsequent data shifted 2 words. - Unaffected if $S_2 \leqq S_3$.				
	Flag	Contents of S_2, S_3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		If $S_2 > S_3$	0	0	0	0
		If $S_2 \leqq S_3$			1	

[Explanation]

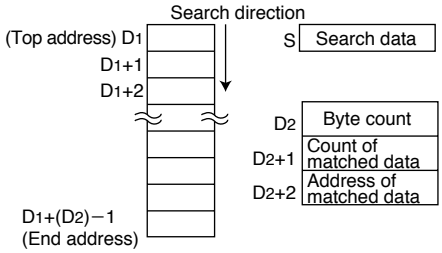


Instruction	
S T R	004000
F -171d	
	009010
	019300
	019301

- Calculates when input condition 004000 changes from OFF to ON.
- The top address is register 009010.
- The address of deletion has an offset from address 009010 given by the contents (002(8) double words) of register 019301.
- The end address has an offset from address 009010 given by the contents (005(8) double words) of register 019300. The top address is counted in the double words count.
- When the JW300 executes calculation, it erases the deleted address, and shifts data from the deleted address to the end address. Double words of the end address become 0.

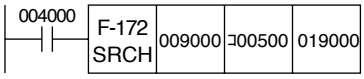


F-172 SRCH — Searches data (1 byte) (SeaRCH)

Symbol	F-172 SRCH | S | D_1 | D_2
Function	Take register D_1 as top address, searches in the range specified by the number of bytes (D_2). The register S detail is data to search. Searches the data and stores the first matched address position (from D_1) to D_2+2. The number of matches is stored in D_2+1.
Operation	 - The shift register area that is set by D_1 and D_2 shall not be in the contact point area of counter/timer (file addresses 00001600 to 00001777(8) etc.), after the last address of the register and file register.
S	Use range A
D_1	Use range A
D_2	Use range E - The contents of register D_2 are 000 to 377(8). (000(8) specifies the 256 bytes.) - Detail of the register D_2+1 are 000 to 377(8). When it is 000(8), the number of matches is 0 or 256. Use the carry flag (007356) to identify the presence of matched data. - Detail of the register D_2+2 is 000 to 377(8).The number of bytes, taking D_1 as standard, is stored. (When this is 000(8), the match data is the top address.)
Condition	Rising edge of input signal (OFF to ON)

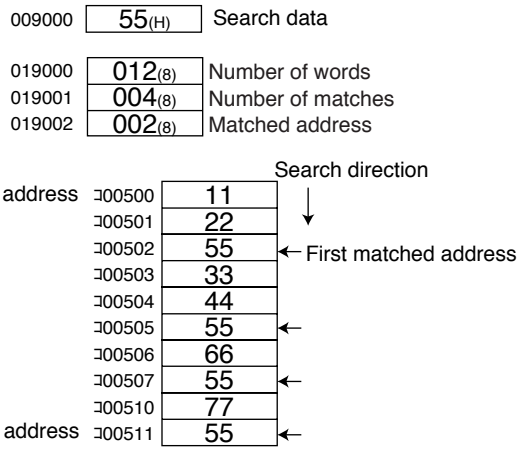
Contents after operation						
	S	Unchanged				
	D_1 to D_1+(D_2)−1	Unchanged				
	D_2	Unchanged				
	D_2+1, D_2+2	Result	- When the number of matches is 0 or 256, D_2+1 will be 000(8).			
	Flag	Search data	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		Found	0	1	0	0
		Not found	1	0		

[Explanation]



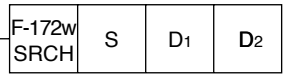
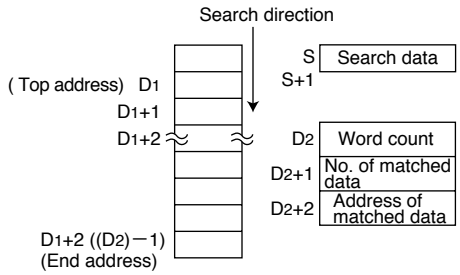
Instruction	
S T R	004000
F -172	
	009000
	⊐00500
	019000

- This instruction is executed when the input condition of 004000 changes from OFF to ON.
- The contents of register 009000 is search data.
- The top address is register ⊐00500.
- The end address has an offset from address ⊐00500 given by the contents (012(8) bytes) of register 019000. The top address is included in the byte count.
- The number of matched data items is stored to register 019001.
- The address (offset from D_1) of the first matched data found is stored in register 019002.



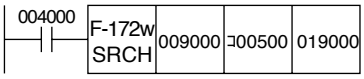
- When execution is complete, the carry flag (007356) is set ON.

F-172w SRCH — Searches data (1 word) (SeaRCH)

Symbol	
Function	Taking register D_1 as the top address, searches in the range specified by the number of words (D_2). Register S and S+1 details are data to search. Store the address (offset from D_1) of the first matched data in D_2+2 and the number of matched data items in D_2+1.
Operation	 - The shift register area that is set by D_1 and D_2 shall not be in the contact point area of counter/timer (file addresses 00001600 to 00001777(8) etc.), after the last address of the register and file register.
S	Use range B - Be sure to use even addresses for register S.
D_1	Use range B - Be sure to use even addresses for register D_1.
D_2	Use range E - The contents of register D_2 are 000 to 377(8). (000(8) specifies the 256 words.) - Detail of the register D_2+1 are 000 to 377(8). When it is 000(8), the number of matches is 0 or 256. Use the carry flag (007356) to identify the presence of matched data. - Detail of the register D_2+2 is 000 to 377(8).The number of words, taking D_1 as standard, is stored. (When this is 000(8), the match data is the top address.)
Condition	Rising edge of input signal (OFF to ON)

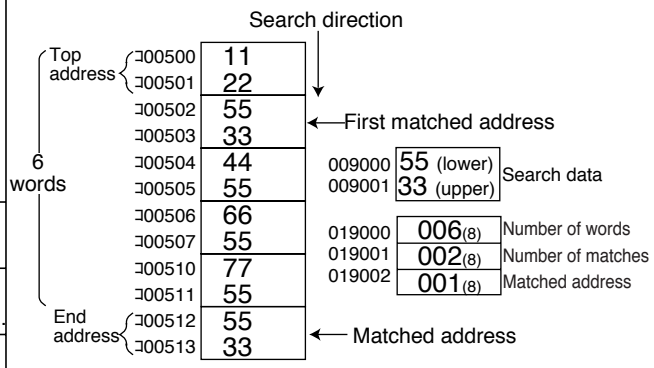
Contents after operation	S, S+1	Unchanged				
	D_1 to D_1+2 (D_2) −1	Unchanged				
	D_2	Unchanged				
	D_2+1, D_2+2	Result	- When the number of matches is 0 or 256, D_2+1 will be 000(8).			
	Flag	Search data	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		Found	0	1	0	0
		Not found	1	0		

[Explanation]



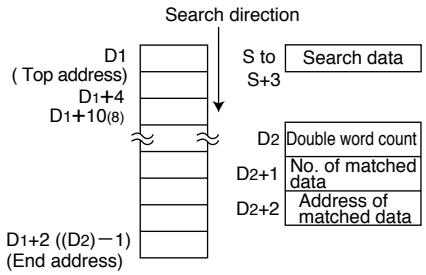
Instruction	
S T R	004000
F -172w	
	009000
	⊐00500
	019000

- This instruction is executed when the input condition of 004000 changes from OFF to ON.
- The contents (3355(H)) of registers 009000 and 009001 are the search data.
- The top address is register ⊐00500.
- The end address has an offset from address ⊐00500 given by the contents (006(8) words) of register 019000. The top address is included in the byte count.
- The number of matched data items is stored to register 019001.
- The address (offset from D_1) of the first matched data found is stored in register 019002.



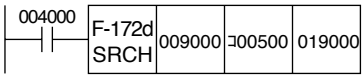
- When execution is complete, the carry flag (007356) is set ON.

F-172d SRCH — Searches data (2 words) (SeaRCH)

Symbol	F-172d SRCH | S | D_1 | D_2
Function	Taking register D_1 as the top address, searches in the range specified by the number of double words (D_2). Register S to S+3 details are data to search. Store the address (offset from D_1) of the first matched data in D_2+2 and the number of matched data items in D_2+1.
Operation	 - The shift register area that is set by D_1 and D_2 shall not be in the contact point area of counter/timer (file addresses 00001600 to 00001777(8) etc.), after the last address of the register and file register.
S	Use range C - Be sure to use even addresses for register S.
D_1	Use range C - Be sure to use even addresses for register D_1.
D_2	Use range E - The contents of register D_2 are 000 to 377(8). (000(8) specifies the 256 double words.) - Detail of the register D_2+1 are 000 to 377(8). When it is 000(8), the number of matches is 0 or 256. Use the carry flag (007356) to identify the presence of matched data. - Detail of the register D_2+2 is 000 to 377(8).The number of double words, taking D_1 as standard, is stored. (When this is 000(8), the match data is the top address.)
Condition	Rising edge of input signal (OFF to ON)

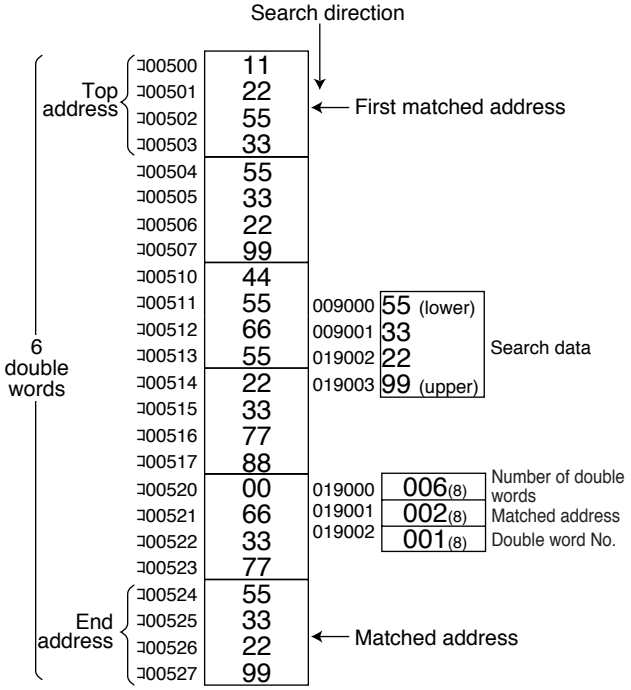
Contents after operation	S to S+3	Unchanged				
	D_1 to D_1+4 (D_2) −1	Unchanged				
	D_2	Unchanged				
	D_2+1, D_2+2	Result	- When the number of matches is 0 or 256, D_2+1 will be 000(8).			
	Flag	Search data	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		Found	0	1	0	0
		Not found	1	0		

[Explanation]



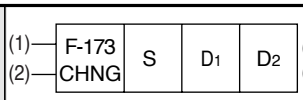
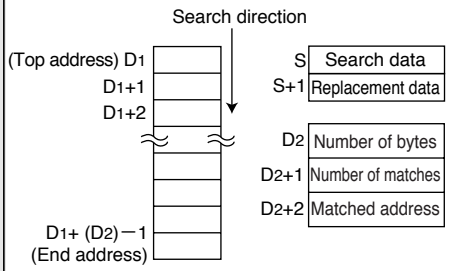
Instruction	
S T R	004000
F -172d	
	009000
	⊐00500
	019000

- This instruction is executed when the input condition of 004000 changes from OFF to ON.
- The contents (99223355(H)) of registers 009000 to 009003 are the search data.
- The top address is register ⊐00500.
- The end address has an offset from address ⊐00500 given by the contents (006(8) double words) of register 019000. The top address is included in the double words count.
- The number of matched data items is stored to register 019001.
- The address (offset from D_1) of the first matched data found is stored in register 019002.

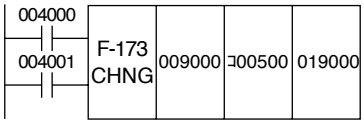


- When execution is complete, the carry flag (007356) is set ON.

F-173 CHNG — Changes data (1 byte) (CHaNG)

Symbol	
Function	Taking register D_1 as the top address, searches in the range specified by the number of byte (D_2). Register S details are data to search. Store the address (offset from D_1) of the first matched data in D_2+2 and the number of matched data items in D_2+1. Data to be overwritten is the detail of register S+1. The JW300 overwrites with mode specification (1) input condition. - When mode input (1) is OFF: Only the first matched data is replaced. - When mode input (1) is ON: All matched data are replaced.
Operation	 - The shift register area that is set by D_1 and D_2 shall not be in the contact point area of counter/timer (file addresses 00001600 to 00001777(8) etc.), after the last address of the register and file register.
S	Use range B
D_1	Use range A
D_2	Use range E - The contents of register D_2 are 000 to 377(8). (000(8) specifies the 256 bytes.) - Detail of the register D_2+1 are 000 to 377(8). When it is 000(8), the number of matches is 0 or 256. Use the carry flag (007356) to identify the presence of matched data. - Detail of the register D_2+2 is 000 to 377(8).The number of byte, taking D_1 as standard, is stored. (When this is 000(8), the match data is the top address.)
Condition	Rising edge of input signal ((2) execution input) (OFF to ON)

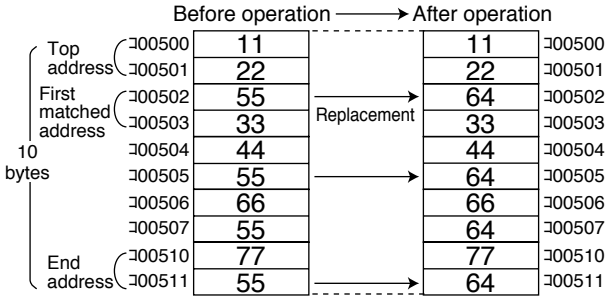
[Explanation]



Instruction	
S T R	004000
S T R	004001
F -173	
	009000
	ↄ00500
	019000

- This instruction is executed when the input condition of 004001 changes from OFF to ON.
- The contents (55(H)) of register 009000 is the search data.
- The top address is register ↄ00500.
- The end address has an offset from address ↄ00500 given by the contents (010(D) bytes) of register 019000. The top address is included in the byte count.
- The number of matched data items is stored to register 019001.
- The address (offset from D_1) of the first matched data found is stored in register 019002.
- The contents (64(H)) of register 009001 is the replacement data.
- Example: Below is an operation when 004000 is ON.

009000	55	Search data	019000	010(D)	Number of bytes
009001	64	Replacement data	019001	004(8)	Number of matches
			019002	002(8)	Byte No.

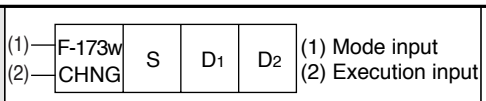
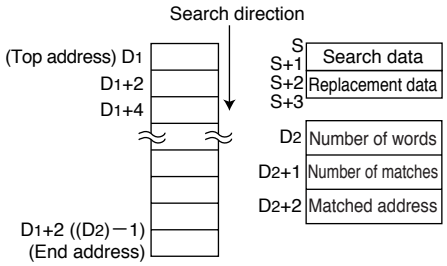


- When execution is complete, the carry flag (007356) is set ON.

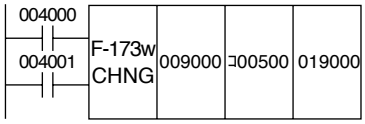
Contents after operation	S, S+1	Unchanged				
	D_2	Unchanged				
	D_2+1, D_2+2	Result	- When the number of matches is 0 or 256, D_2+1 will be 000(8).			
	Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		Found	0	1	0	0
		Not found	1	0		

F-173w CHNG

Changes data (1 word) (CHaNG)

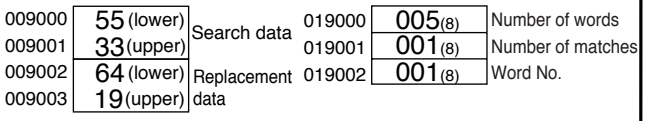
Symbol	
Function	Search for data matching the contents of registers S and S+1 in this register area whose top address is D_1 and whose word count is given by the contents of register D_2 and store the address (offset from D_1) of the first matched data in D_2+2 and the number of matched data items in D_2+1. Replace matched data with the contents of registers S+2 and S+3 depending on the state of mode input (1): - When mode input (1) is OFF: Only the first matched data is replaced. - When mode input (1) is ON: All matched data are replaced.
Operation	 - The shift register area that is set by D_1 and D_2 shall not be in the contact point area of counter/timer (file addresses 00001600 to 00001777(8) etc.), after the last address of the register and file register.
S	Use range C - Be sure to use even addresses for register S.
D_1	Use range B - Be sure to use even addresses for register D_1.
D_2	Use range E - The contents of register D_2 are 000 to 377(8). (000(8) specifies the 256 words.) - Detail of the register D_2+1 are 000 to 377(8). When it is 000(8), the number of matches is 0 or 256. Use the carry flag (007356) to identify the presence of matched data. - Detail of the register D_2+2 is 000 to 377(8).The number of words, taking D_1 as standard, is stored. (When this is 000(8), the match data is the top address.)
Condition	Rising edge of input signal ((2) execution input) (OFF to ON)

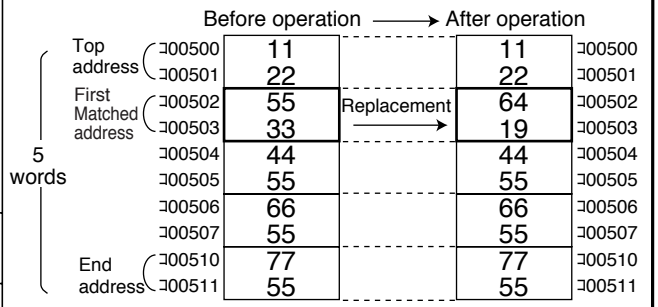
[Explanation]



Instruction	
S T R	004000
S T R	004001
F -173w	
	009000
	⊐00500
	019000

- This instruction is executed when the input condition of 004000 changes from OFF to ON.
- The contents (3355(H)) of register 009000, 009001 are the search data.
- The top address is register ⊐00500.
- The end address has an offset from address ⊐00500 given by the contents (005(8) words) of register 019000. The top address is included in the word count.
- The number of matched data items is stored to register 019001.
- The address (offset from D_1) of the first matched data found is stored in register 019002.
- The contents (1964(H)) of registers 009002 and 009003 are the replacement data.
- Example: Below is an operation when 00x000 is ON.

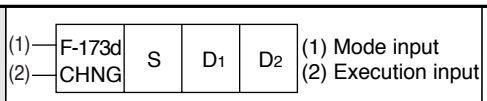




Contents after operation	S to S+3	Unchanged				
	D_2	Unchanged				
	D_2+1, D_2+2	Result	- If the result is 256 or no matched data.D_2+1 becomes 000.			
	Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		Found	0	1	0	0
		Not found	1	0		

F-173d CHNG — Changes data (2 words) (CHaNG)

Symbol

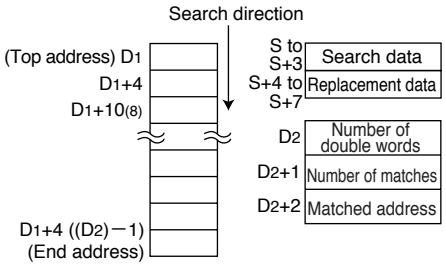


Function

Search for data matching the contents of registers S to S+3 in this register area whose top address is D_1 and whose word count is given by the contents of register D_2 and store the address (offset from D_1) of the first matched data in D_2+2 and the number of matched data items in D_2+1. Replace matched data with the contents of registers S+4 to S+7 depending on the state of mode input (1):

- When mode input (1) is OFF:
  Only the first matched data is replaced.
- When mode input (1) is ON:
  All matched data are replaced.

Operation



- The shift register area that is set by D_1 and D_2 shall not be in the contact point area of counter/timer (file addresses 00001600 to 00001777(8) etc.), after the last address of the register and file register.

S

Use range C

- Be sure to use even addresses for register S.

D_1

Use range B

- Be sure to use even addresses for register D_1.

D_2

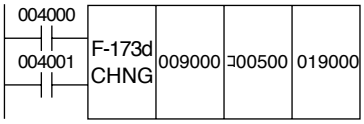
Use range E

- The contents of register D_2 are 000 to 377(8). (000(8) specifies the 256 double words.)
- Detail of the register D_2+1 are 000 to 377(8). When it is 000(8), the number of matches is 0 or 256. Use the carry flag (007356) to identify the presence of matched data.
- Detail of the register D_2+2 is 000 to 377(8).The number of double words, taking D_1 as standard, is stored. (When this is 000(8), the match data is the top address.)

Condition

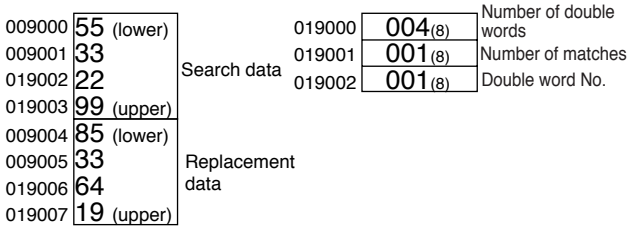
Rising edge of input signal ((2) execution input) (OFF to ON)

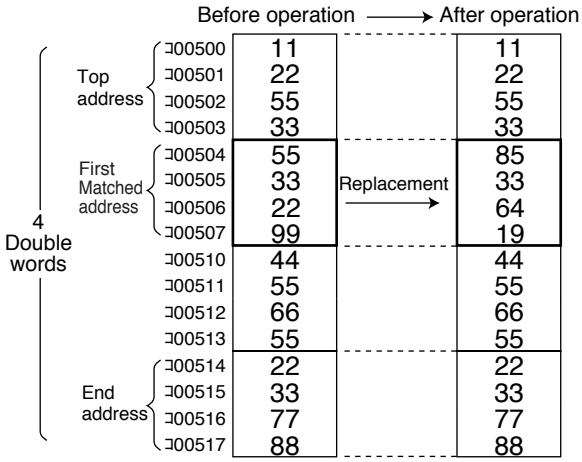
[Explanation]



Instruction	
S T R	004000
S T R	004001
F -173d	
	009000
	ↄ00500
	019000

- This instruction is executed when the input condition of 004001 changes from OFF to ON.
- The contents (99223355(H)) of register 009000 to 009003 are the search data.
- The top address is ↄ00500.
- The end address has an offset from address ↄ00500 given by the contents (004(8) double words) of register 019000. The top address is included in the double words count.
- The number of matched data items is stored to register 019001.
- The address (offset from D_1) of the first matched double words found is stored in register 019002.
- The contents (19643385(H)) of registers 009004 to 009007 are the replacement data.
- Example: Below is an operation when 004000 is ON.





Contents after operation

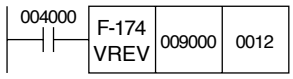
S to S+7	Unchanged					
D_2	Unchanged					
D_2+1, D_2+2	Result	- If the result is 256 or no matched data.D_2+1 becomes 000(8).				
Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
	Found	0	1	0	0	
	Not found	1	0			

Chapter 14 Application instructions (F-174 to F-403)

F-174 VREV — Reverse order of register data (1 byte) (Vertical REVerse)

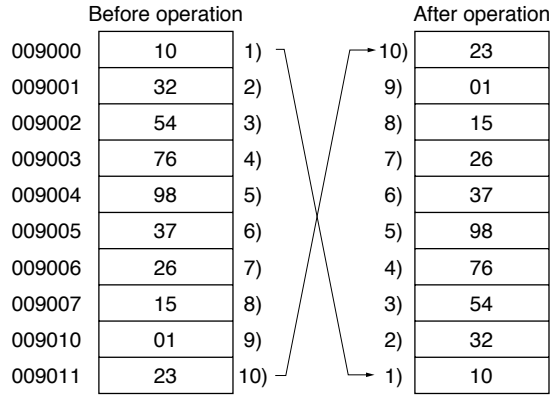
Symbol		F-174 VREV | D | n
Function		Within the n byte register that starts with register D, swap the contents of each register between "the lowest address to the upper most address" and "the upper most address to the lowest address."
Operation		D | D+1 | ... | D+n-2 | D+n-1 (Data swap)
D		Use range A *
n		Use range 0000 to 1777(8) (1024 bytes for 0000(8))
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	D to D+n-1	Result
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004000
F -174	
	009000
	0012

When the input condition of 004000 changes from OFF to ON, converts the contents of the 10 ($0012_{(8)}$) in registers 009000 to 009011 in reverse order. Swap between "009000 -> 009011" and "0090011 -> 009000."



* Be careful that the register areas set by D and n do not enter the contact point areas of the timer/counter (file address 00001600 to 00001777(8) etc.) and the last address or more of the registers and file registers.

Resembled instructions: F-02, F-02w, F-02d

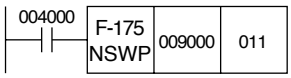
F-175 NSWP — Swap upper 4 bits with lower 4 bits of register (N byte SWaP)

Symbol		F-175 NSWP | D | n
Function		Swap the upper 4 bits with lower 4 bits of "n" bytes of register area that begins with register D.
Operation		Swap nibbles of registers D to D+n-1.
D		Use range A *
n		Use range 0000 to 1777(8) (1024 bytes for 0000(8))
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	D to D+n-1	Result
	Flag	Unchanged

* Be careful that the shift area set by D and n do not enter the contact point areas of the timer/counter (file address 00001600 to 00001777(8) etc.) and the last address or more of the registers and file registers.

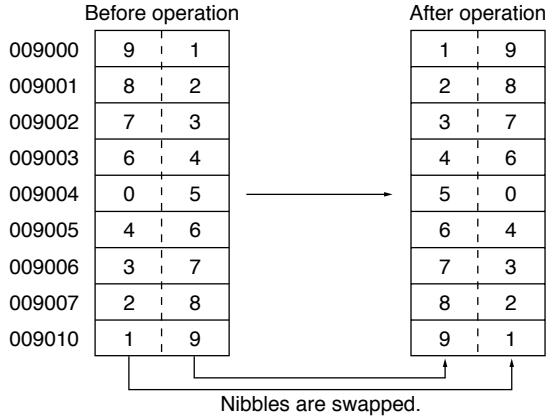
Resembled instructions: F-55

[Explanation]



Instruction	
S T R	004000
F -175	
	009000
	0011

When the input condition of 004000 changes from OFF to ON, this instruction swaps the upper nibbles with lower nibbles of the 9 ($0011_{(8)}$)-byte contents of registers 009000 to 009010 (1-byte register each).



F-176 DFRD — Read 256 bytes from register of specified address (Direct File Read)

Symbol		F-176 DFRD | S | FILE F | D
Function		In file number F, transfer data (256 bytes), that was specified by register S and S+1, to the address starting from register D.
Operation		Transfer 256 bytes from file F, S, S+1 (block No.).
S		Use range B - Be sure to use even addresses for register S.
F		0 (Data memory except file register) 1 (File register)
D		Use range J - Be sure to use even addresses for register D.
Condition		Rising edge of input signal (OFF to ON)
After operation	S to S+255	Unchanged
	D to D+255	Result
	Flag	Unchanged

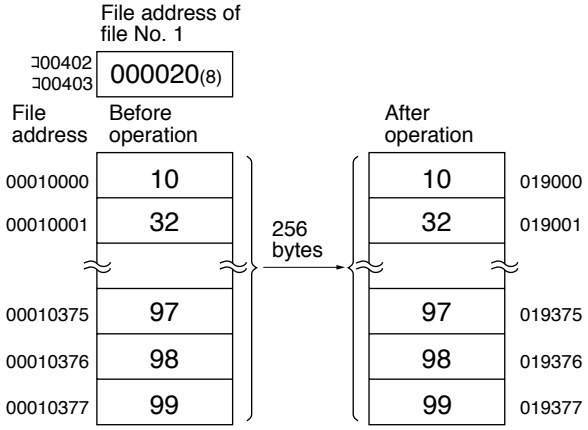
Resembled instructions: F-102, F-102w, F102d

[Explanation]

Instruction	
S T R	004000
F -176	
	⊐00402
	FILE 1
	019000

004000 | F-176 DFRD | ⊐00402 | FILE 1 | 019000

When the input condition of 004000 changes from OFF to ON, this instruction transfers 256 bytes of data from file No. 1's file area that begins with block number 000020(8) (contents of register ⊐00402, ⊐00403) to the register area that begins with register 019000.



- The top address of block No. 000020(8) of file No. 1 is 00010000(8).

- S and S+1 are block numbers (000000 to 100000(8)) separated file No. 0 (file address) and file number 1 (byte address) in 256 bytes each. => See "Data memory block number."
  This instruction performs block transfer of 256 bytes each.
- Be careful for the top address that is specified by D. 256 bytes from there will be a transfer area.

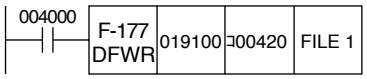
F-177 DFWR — Write data into registers of specified address (256 bytes) (Direct File WRite)

Symbol		F-177 DFWR | S | D | FILE F
Function		Transfers 256 bytes of data that starts from register S to the area that was specified by register D and D+1 in file No. F.
Operation		Transfers 256 bytes from S to D, D+1 (block No.) of file F.
S		Use range B - Be sure to use even addresses for register S.
D		Use range B
F		0 (Data memory except file register) 1 (File register)
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S to S+255	Unchanged
	D to D+255	Result
	Flag	Unchanged

Resembled instructions: F-103, F-103w, F103d

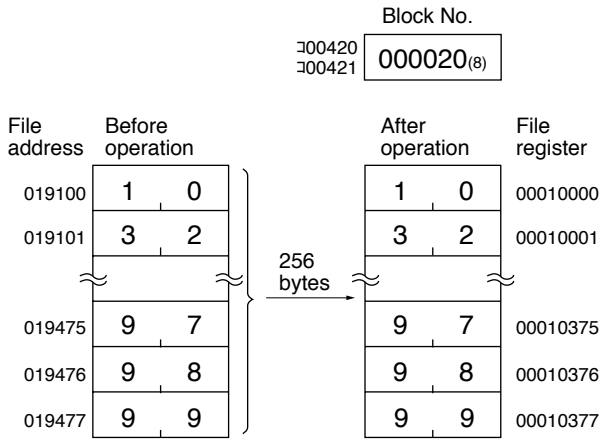
- The contents of D and D+1 use block No. (000000 to $100000_{(8)}$) that is separated file No. 0 (file address) and file No. 1 (byte address) in units of 256 bytes each. => See "Data memory block number."
  This instruction performs block transfer of 256 bytes each.
- For setting the block number to D, do not use block No. (3, 78, 301 to $307_{(8)}$) that contains the contact area of the TMR/CNT of file No. 0.

[Explanation]



Instruction	
S T R	004000
F -177	
	019100
	⊐00420
	FILE 1

When the input condition of 004000 changes from OFF to ON, this instruction transfers 256 bytes of data to the area that was specified by block No. $000020_{(8)}$ (register ⊐00420, ⊐00421) of file No. 1 (file register).

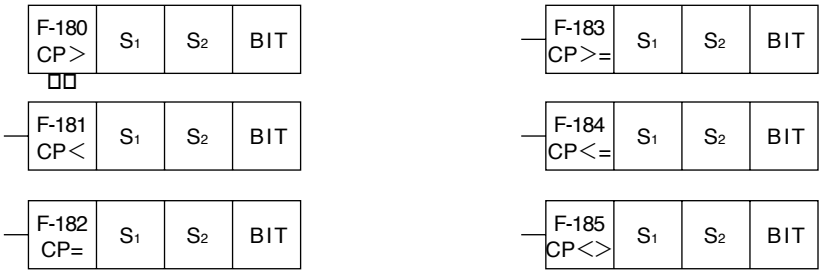


- The top address of block No. $000020_{(8)}$ of file No. 1 is $00010000_{(8)}$.

Compare between register and register (1 byte) (with relay output)

F-180 CP＞	(ComPare ＞)	F-183 CP＞＝	(ComPare ＞＝)
F-181 CP＜	(ComPare ＜)	F-184 CP＜＝	(ComPare ＜＝)
F-182 CP=	(ComPare ＝)	F-185 CP＜＞	(ComPare ＜＞)

Symbol



Function

Comparing the magnitude between the content of register S_1 and content of S_2, the bit is turned ON when the result of comparison is established.

Operation

F-180	$S_1 > S_2 \rightarrow$ BIT ON	F-183	$S_1 \geqq S_2 \rightarrow$ BIT ON
F-181	$S_1 < S_2 \rightarrow$ BIT ON	F-184	$S_1 \leqq S_2 \rightarrow$ BIT ON
F-182	$S_1 = S_2 \rightarrow$ BIT ON	F-185	$S_1 \neq S_2 \rightarrow$ BIT ON

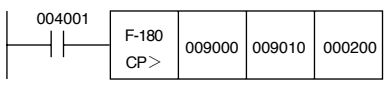
S_1	Use range A	S_2	Use range A
BIT	Use range K	Condition	When the input signal is ON (not limited to OFF to ON change)

Contents after operation

S_1, S_2	Unchanged				
BIT	F-180	ON ($S_1 > S_2$) OFF ($S_1 \leqq S_2$)	F-183	ON ($S_1 \leqq S_2$) OFF ($S_1 < S_2$)
	F-181	ON ($S_1 < S_2$) OFF ($S_1 \geqq S_2$)	F-184	ON ($S_1 \leqq S_2$) OFF ($S_1 > S_2$)
	F-182	ON ($S_1 = S_2$) OFF ($S_1 \neq S_2$)	F-185	ON ($S_1 \neq S_2$) OFF ($S_1 = S_2$)

Flag	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	0	0	0	0

[Explanation]



Instruction	
S T R	004001
F -180	
	009000
	009010
	000200

When the input condition 004001 is ON, the contents of register 009000 and register 009010 are compared, and the relay 000200 is ON if 009000 ＞ 009010.
When the input condition 004001 is OFF, and 009000 ≦ 009010, relay 000200 goes OFF.

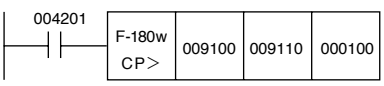
Compare between register and register (1 word) (with relay output)

F-180w CP>	(ComPare >)	F-183w CP>=	(ComPare >=)
F-181w CP<	(ComPare <)	F-184w CP<=	(ComPare <=)
F-182w CP=	(ComPare =)	F-185w CP<>	(ComPare <>)

Symbol	F-180w CP>	F-181w CP<	F-182w CP=	F-183w CP>= / F-184w CP<= / F-185w CP<>
	S1	S2	BIT	
Function	The contents (1-word data) of register S1, S1 + 1 and the contents (1-word data) of registers S2, S2 + 1 are compared in magnitude, BIT is turned ON when the result is properly obtained.			
Operation	F-180 w	S1, S1+1 > S2, S2+1 → BIT ON	F-183 w	S1, S1+1 ≧ S2, S2+1 → BIT ON
	F-181 w	S1, S1+1 < S2, S2+1 → BIT ON	F-184 w	S1, S1+1 ≦ S2, S2+1 → BIT ON
	F-182 w	S1, S1+1 = S2, S2+1 → BIT ON	F-185 w	S1, S1+1 ≠ S2, S2+1 → BIT ON
S1	Use range B		S2	Use range B
BIT	Use range K		Condition	When the input signal is ON (not limited to OFF to ON change)

Contents after operation					
S1, S1 + 1 S2, S2 + 1	Unchanged				
BIT	F-180w	ON (S1, S1+1 > S2, S2+1) OFF (S1, S1+1 ≦ S2, S2+1)	F-183w	ON (S1, S1+1 ≧ S2, S2+1) OFF (S1, S1+1 < S2, S2+1)	
	F-181w	ON (S1, S1+1 < S2, S2+1) OFF (S1, S1+1 ≧ S2, S2+1)	F-184w	ON (S1, S1+1 ≦ S2, S2+1) OFF (S1, S1+1 > S2, S2+1)	
	F-182w	ON (S1,S1+1 = S2, S2+1) OFF (S1,S1+1 ≠ S2, S2+1)	F-185w	ON (S1, S1+1 ≠ S2, S2+1) OFF (S1, S1+1 = S2, S2+1)	
Flag	Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
	0	0	0	0	

[Explanation]

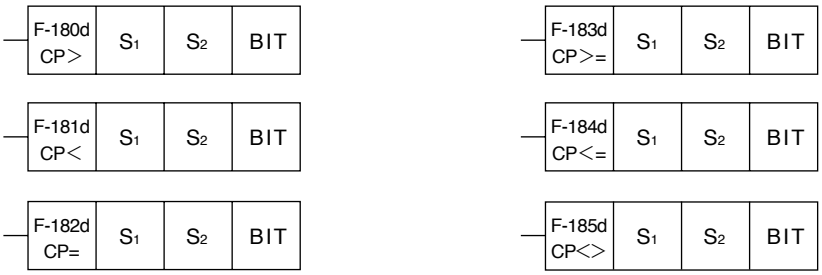


Instruction	
S T R	004201
F-180w	
	009100
	009110
	000100

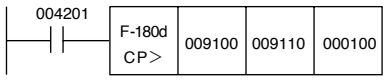
When the input condition 004201 is ON, the contents of registers 009100 and 009101, and the contents of registers 009110 and 009111 are compared, and the relay 000100 is ON if (009100, 009101) > (009110, 009111). When the input condition 04201 is OFF, and (009100, 090101) ≦ (009110, 009111), relay 000100 goes OFF.

Compare between register and register (2 words) (with relay output)

F-180d CP>	(ComPare >)	F-183d CP>=	(ComPare >=)
F-181d CP<	(ComPare <)	F-184d CP<=	(ComPare <=)
F-182d CP=	(ComPare =)	F-185d CP<>	(ComPare <>)

<table>
<tr><td>Symbol</td><td colspan="5"> 
</td></tr>
<tr><td>Function</td><td colspan="5">The contents (2-words data) of register S_1 to S_1 + 3 and the contents (2-words data) of registers S_2 to S_2 + 3 are compared in magnitude, and a bit is turned ON when the result is properly obtained.</td></tr>
<tr><td rowspan="3">Operation</td><td>F-180d</td><td>S_1 to S_1+3 > S_2 to S_2+3 → BIT ON</td><td>F-183d</td><td colspan="2">S_1 to S_1+3 ≧ S_2 to S_2+3 → BIT ON</td></tr>
<tr><td>F-181d</td><td>S_1 to S_1+3 < S_2 to S_2+3 → BIT ON</td><td>F-184d</td><td colspan="2">S_1 to S_1+3 ≦ S_2 to S_2+3 → BIT ON</td></tr>
<tr><td>F-182d</td><td>S_1 to S_1+3 = S_2 to S_2+3 → BIT ON</td><td>F-185d</td><td colspan="2">S_1 to S_1+3 ≠ S_2 to S_2+3 → BIT ON</td></tr>
<tr><td>S_1</td><td colspan="2">Use range C</td><td>S_2</td><td colspan="2">Use range C</td></tr>
<tr><td>BIT</td><td colspan="2">Use range K</td><td>Condition</td><td colspan="2">When the input signal is ON
(not limited to OFF to ON change)</td></tr>
<tr><td rowspan="5">Contents after operation</td><td>S_1 to S_1 + 3
S_2 to S_2 + 3</td><td colspan="5">Unchanged</td></tr>
<tr><td rowspan="3">BIT</td><td>F-180d</td><td>ON (S_1 to S_1+3 > S_2 to S_2+3)
OFF (S_1 to S_1+3 ≦ S_2 to S_2+3)</td><td>F-183d</td><td colspan="2">ON (S_1 to S_1+3 ≧ S_2 to S_2+3)
OFF (S_1 to S_1+3 < S_2 to S_2+3)</td></tr>
<tr><td>F-181d</td><td>ON (S_1 to S_1+3 < S_2 to S_2+3)
OFF (S_1 to S_1+3 ≧ S_2 to S_2+3)</td><td>F-184d</td><td colspan="2">ON (S_1 to S_1+3 ≦ S_2 to S_2+3)
OFF (S_1 to S_1+3 > S_2 to S_2+3)</td></tr>
<tr><td>F-182d</td><td>ON (S_1 to S_1+3 = S_2 to S_2+3)
OFF (S_1 to S_1+3 ≠ S_2 to S_2+3)</td><td>F-185d</td><td colspan="2">ON (S_1 to S_1+3 ≠ S_2 to S_2+3)
OFF (S_1 to S_1+3 = S_2 to S_2+3)</td></tr>
<tr><td>Flag</td><td colspan="5"><table><tr><td>Zero
007357</td><td>Carry
007356</td><td>Error
007355</td><td>Non-carry
007354</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
</table>

[Explanation]

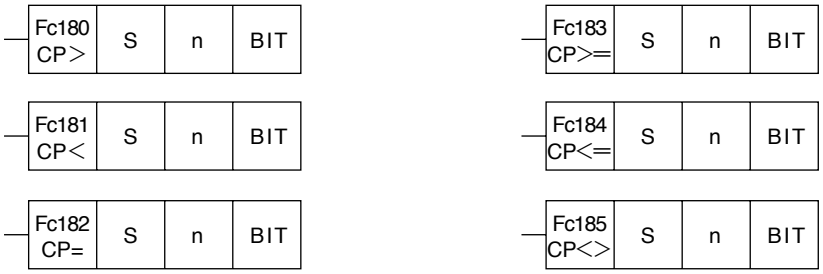


Instruction	
S T R	004201
F-180d	
	009100
	009110
	000100

When the input condition 004201 is ON, the contents of registers 0009100 and 009101, and the contents of registers 009110 and 009111 are compared, and the relay 000100 is ON if (009100, 009103) > (009110, 009113). When the input condition 04201 is OFF, and (009100, 090103) ≦ (009110, 009113), relay 000100 is OFF.

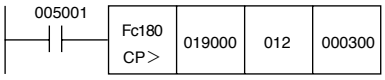
Compare register with constant (1 byte) (with relay output)

Fc180 CP>	(ComPare >)	Fc183 CP>=	(ComPare >=)
Fc181 CP<	(ComPare <)	Fc184 CP<=	(ComPare <=)
Fc182 CP=	(ComPare =)	Fc185 CP<>	(ComPare <>)

Symbol					
Function	Comparing the magnitude between the content of register S and content of n, and a bit is turned ON when the result is properly obtained.				
Operation	Fc180	S >n → BIT ON	Fc183	S ≧ n → BIT ON	
	Fc181	S <n → BIT ON	Fc184	S ≦ n → BIT ON	
	Fc182	S = n → BIT ON	Fc185	S ≠ n → BIT ON	
S	Use range A		n	Use range 000 to 377(8)	
BIT	Use range K		Condition	When the input signal is ON (not limited to OFF to ON change)	
After operation S	Unchanged				
After operation BIT	Fc180	ON (S>n) OFF (S≦n)	Fc183	ON (S≧n) OFF (S<n)	
	Fc181	ON (S<n) OFF (S≧n)	Fc184	ON (S≦n) OFF (S>n)	
	Fc182	ON (S=n) OFF (S≠n)	Fc185	ON (S≠n) OFF (S=n)	

After operation Flag	Zero 007357	Carry 007356	Error 070355	Non-carry 007354
	0	0	0	0

[Explanation]

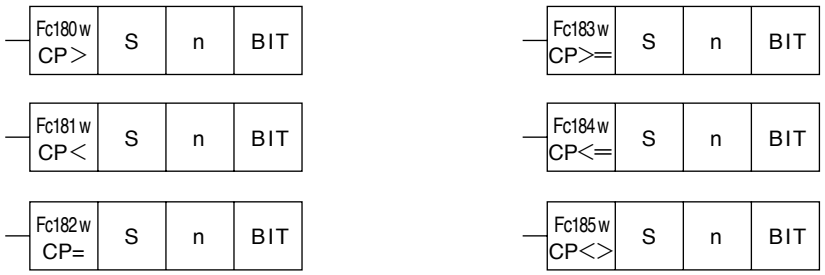


Instruction	
S T R	005001
Fc180	
	019000
	012
	000300

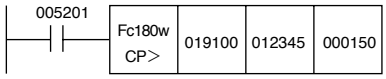
When the input condition 005001 is ON, the contents of register 019000 and octal constant 012 are compared, and the relay 000300 is ON if (019000) > 012(8).
When the input condition 005001 is OFF, and (019000) ≦ 012(8), relay 000300 goes OFF.

Compare register with constant (1 word) (with relay output)

Fc180w CP>	(ComPare >)	Fc183w CP>=	(ComPare >=)
Fc181w CP<	(ComPare <)	Fc184w CP>=	(ComPare <=)
Fc182w CP=	(ComPare =)	Fc185w CP<>	(ComPare <>)

<table>
<tr><td>Symbol</td><td colspan="5"> 
</td></tr>
<tr><td>Function</td><td colspan="5">The contents (1-word data) of register S, S + 1 and the content (1-word data) of a constant n are compared in magnitude, and a bit is turned ON when the result is properly obtained.</td></tr>
<tr><td rowspan="3">Operation</td><td>Fc180 w</td><td>S, S+1 > n → BIT ON</td><td>Fc183 w</td><td colspan="2">S, S+1 ≧ n → BIT ON</td></tr>
<tr><td>Fc181 w</td><td>S, S+1 < n → BIT ON</td><td>Fc184 w</td><td colspan="2">S, S+1 ≦ n→ BIT ON</td></tr>
<tr><td>Fc182 w</td><td>S, S+1 = n → BIT ON</td><td>Fc185 w</td><td colspan="2">S, S+1 ≠ n→ BIT ON</td></tr>
<tr><td>S</td><td colspan="2">Use range B</td><td>n</td><td colspan="2">Use range 000000 to 177777(8)</td></tr>
<tr><td>BIT</td><td colspan="2">Use range K</td><td>Condition</td><td colspan="2">When the input signal is ON
(not limited to OFF to ON change)</td></tr>
<tr><td rowspan="5">Contents after operation</td><td>S</td><td colspan="5">Unchanged</td></tr>
<tr><td rowspan="3">BIT</td><td>Fc180w</td><td>ON (S, S+1 > n)
OFF (S, S+1 ≦ n)</td><td>Fc183w</td><td colspan="2">ON (S, S+1 ≧ n)
OFF (S, S+1 < n)</td></tr>
<tr><td>Fc181w</td><td>ON (S, S+1 < n)
OFF (S, S+1 ≧ n)</td><td>Fc184w</td><td colspan="2">ON (S, S+1 ≦ n)
OFF (S, S+1 > n)</td></tr>
<tr><td>Fc182w</td><td>ON (S, S+1 = n)
OFF (S, S+1 ≠ n)</td><td>Fc185w</td><td colspan="2">ON (S, S+1 ≠ n)
OFF (S, S+1 = n)</td></tr>
<tr><td>Flag</td><td colspan="5">Zero 007357: 0 | Carry 007356: 0 | Error 007355: 0 | Non-carry 007354: 0</td></tr>
</table>

[Explanation]

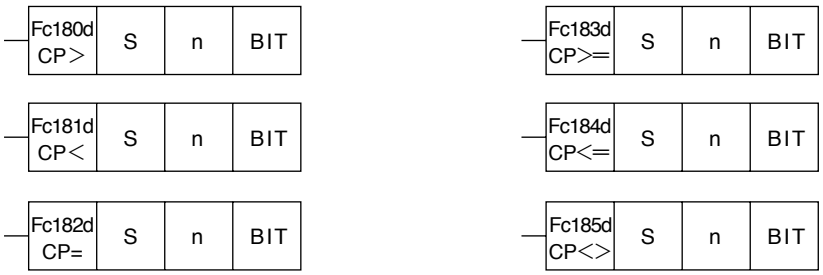


Instruction	
S T R	005201
Fc180w	
	019100
	012345
	000150

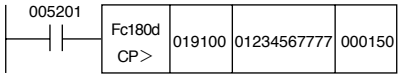
When the input condition 005201 is ON, the contents of registers (019100 and 019101), and the octal constant 012345 are compared, and the relay 000150 is ON if (019100, 019101) > 012345.
When the input condition 005201 is OFF, and (019100, 019101) ≦ 012345, relay 000150 goes OFF.

Compare register with constant (2 words) (with relay output)

Fc180d CP>	(ComPare >)	Fc183d CP>=	(ComPare >=)
Fc181d CP<	(ComPare <)	Fc184d CP>=	(ComPare <=)
Fc182d CP=	(ComPare =)	Fc185d CP<>	(ComPare <>)

<table>
<tr><td>Symbol</td><td colspan="5"> 
</td></tr>
<tr><td>Function</td><td colspan="5">The contents (2-words data) of register S to S + 3 and the constant n are compared in magnitude, and a bit is turned ON when the result is properly obtained.</td></tr>
<tr><td rowspan="3">Operation</td><td>Fc180d</td><td>S to S+3>n → BIT ON</td><td>Fc183d</td><td>S to S+3≧n → BIT ON</td></tr>
<tr><td>Fc181d</td><td>S to S+3<n → BIT ON</td><td>Fc184d</td><td>S to S+3≦ n→ BIT ON</td></tr>
<tr><td>Fc182d</td><td>S to S+3=n → BIT ON</td><td>Fc185d</td><td>S to S+3≠ n→ BIT ON</td></tr>
<tr><td>S</td><td colspan="2">Use range C</td><td>n</td><td>Use range
00000000000 to 37777777777(8)</td></tr>
<tr><td>BIT</td><td colspan="2">Use range K</td><td>Condition</td><td>When the input signal is ON
(not limited to OFF to ON change)</td></tr>
<tr><td rowspan="5">After operation</td><td>S, S+1</td><td colspan="4">Unchanged</td></tr>
<tr><td rowspan="3">BIT</td><td>Fc180d</td><td>ON (S to S+3>n)
OFF (S to S+3≦n)</td><td>Fc183d</td><td>ON (S to S+3≧n)
OFF (S to S+3<n)</td></tr>
<tr><td>Fc181d</td><td>ON (S to S+3<n)
OFF (S to S+3≧n)</td><td>Fc184d</td><td>ON (S to S+3≦n)
OFF (S to S+3>n)</td></tr>
<tr><td>Fc182d</td><td>ON (S to S+3=n)
OFF (S to S+3≠n)</td><td>Fc185d</td><td>ON (S to S+3≠n)
OFF (S to S+3=n)</td></tr>
<tr><td>Flag</td><td colspan="4"><table><tr><td>Zero
007357</td><td>Carry
007356</td><td>Error
007355</td><td>Non-carry
007354</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
</table>

[Explanation]



Instruction	
S T R	005201
Fc180d	
	019100
	01234567777
	000150

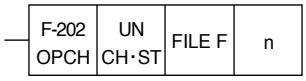
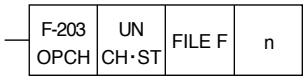
When the input condition 005201 is ON, the contents of registers (019100 to 019103), and the octal constant 01234567777 are compared, and the relay 000150 goes ON if (019100 to 019103) > 01234567777.
When the input condition 005201 is OFF, and (019100 to 019103) ≦ 01234567777, relay 000150 goes OFF.

F-202 OPCH

Open Channel (Signal layer: with Octal Station Number)

F-203 OPCH

Open Channel (Signal layer: with Hex Station Number)

Symbol		
Function	Specify the station for inter-PLC data communications to be performed across the satellite net (JW-22CM, JW-20CM). These instructions are used in conjunction with the F-204 (SEND) or F-205 (RCV) instructions.	
UN	Use range 0 to 7: Module number of JW-22CM.	
CH	Use range 0 to 3: Channel number for the specified module number.	
ST	Use range 000 to $377_{(8)}$, 00 to $FF_{(H)}$: Remote station number. F-202 uses octal notation. F-203 uses hexadecimal notation.	
F	Use range 0, 1: File number of communication target	*
n	Use range 00000000 to $77777777_{(8)}$: File address (data top address) of communication target.	
Condition	When input signal is ON (not limited to OFF to ON change).	
Flag	Unchanged.	

* When the communication target is the conventional model (JW30H etc.), convert these numbers to file number and file address for the JW300. => See "Indirection address assignment".

- If CH0, CH1, CH2, and CH3 are used as channel numbers, the JW-22CM with module number can be used up to 4 times in a PLC program.
- These instructions must always be used in conjunction with the F-204 (SEND) or F-205 (RCV) instruction.

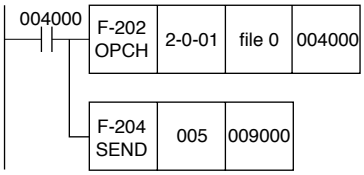
F-204 SEND — Sends data

Symbol		F-204 SEND | n | S
Function		Specify the top address and byte count of the sending data across the satellite net.
Operation		S to S+n−1 → sSpecified station
n		Use range 000 to 377(8) (256 bytes for 000(8))
S		Use range A
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S	Unchanged

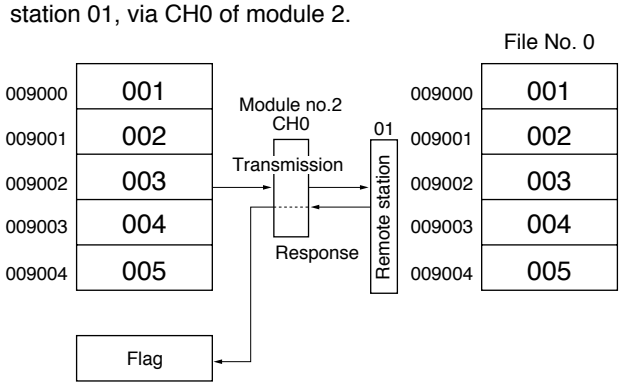
Contents after operation: Flag	Meaning	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	No response from port	0	0	1	0
	Communication congestion	0	0	0	1
	Communication busy (waiting for response from remote station)	1	0	0	1
	Normal end	0	1	0	0
	Abnormal end (communication error)	0	1	1	0
	Remote station write protected	1	1	1	0

[Explanation]

Instruction	
S T R	004000
F -202	
	2-0-01
	file 0
	004000
F -204	
	005
	009000



When the input condition of 004000 changes from OFF to ON, this instruction sends the contents (5-byte data) of registers 009000 to 009004 to file 0's file area beginning with address 004000 (register 009000) on satellite net station 01, via CH0 of module 2.



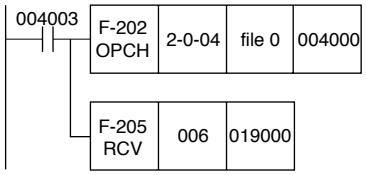
- For the usage of this instruction, see the user's manual for the network module (JW-22CM).
- 1 hierarchical communication must always be used in conjunction with the F-202 or F-203, 2 hierarchical communication must always be used in conjunction with the F-206 and F-207.

F-205 RCV — Receives data

Symbol		F-205 RCV | n | D
Function		Specify the top address and byte count of the receive data for transmission across the satellite net.
Operation		Specified station → D to D+n−1
n		Use range 000 to 377(8) (256 bytes for 000(8))
D		Use range A
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	D	Result

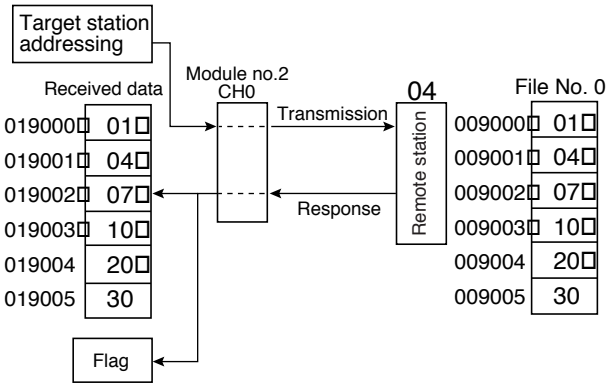
Flag	Meaning	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	No response from port	0	0	1	0
	Communication congestion	0	0	0	1
	Communication busy (waiting for response from remote station)	1	0	0	1
	Normal end	0	1	0	0
	Abnormal end (communication error)	0	1	1	0

[Explanation]



Instruction	
S T R	004003
F -202	
	2-0-04
	file 0
	004000
F -205	
	006
	019000

When the input condition of 004003 changes from OFF to ON, this instruction reads 6 bytes of data from file 0's file area beginning with file address 004000 on satellite net station 04,via CH0 of module 2. The data read is stored to registers 109000 to 019005.



- For the usage of this instruction, see the user's manual for the network module (JW-22CM).
- 1 hierarchical communication must always be used in conjunction with the F-202 or F-203, 2 hierarchical communication must always be used in conjunction with the F-206 and F-207.

F-206
EOP1

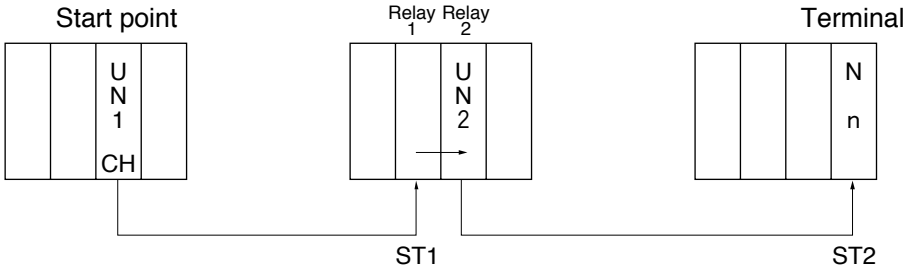
Open channel 1 (set the hierarchical communication)

F-207
EOP2

Open channel 2 (set the hierarchical communication)

Symbol	F-206 EOP1 | UN1、CH | ST1 | UN2	F-207 EOP2 | ST2 | FILE F | n
Function	Specify the station for inter-PLC data communications to be performed across the satellite net (JW-22CM, JW-20CM). The F-206 (EOP1) instructions are used in conjunction with the F-207 (EOP2),	
UN1	Use range 0 to 7: Module no. of module starts from SEND, RECEIVE instructions.	
UN2	Use range 0 to 7: Module no. of module from relay station 2 of SEND, RECEIVE instructions. Note: When the relay station 2 is JW50H/70H/100H, it becomes slot number.	
CH	Use range 0 to 3: Module no. of module starts from SEND, RECEIVE instructions.	
ST1	Use range 00 to $77_{(8)}$: The relay station 1's station number of SEND, RECEIVE instructions.	
ST2	Use range 00 to $77_{(8)}$: Terminal station number of SEND, RECEIVE instructions.	
F	Use range 0, 1: File number in the terminal station of SEND, RECEIVE instructions.	*
n	Use range 00000000 to $77777777_{(8)}$: File address in the terminal station of SEND, RECEIVE instructions.	
Condition	When input signal is ON (not limited to OFF to ON change).	
Flag	Unchanged.	

* When the communication target is the conventional model (JW30H etc.), convert these numbers to file number and file address for the JW300. => See "Indirection address assignment".



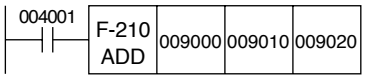
- These instructions must always be used in conjunction with the F-206, F-207, F-204 or F-206, F-207, F-205 instructions.

F-210 ADD — Add register and register in binary (8 bits + 8 bits) (ADD)

Symbol		F-210 ADD | S_1 | S_2 | D
Function		The contents of the register S_1 are added in binary with the contents of the register S_2 and its result is stored in the register D.
Operation		$S_1+S_2 \rightarrow D$
S_1		Use range A
S_2		Use range A
D		Use range A
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1	Unchanged
	S_2	Unchanged
	D	Result

Contents after operation	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
Flag	0	1	0	0	1
	001 to 377(8)	0	0	0	1
	400(8)	1	1	0	0
	Above 401(8)	0	1	0	0

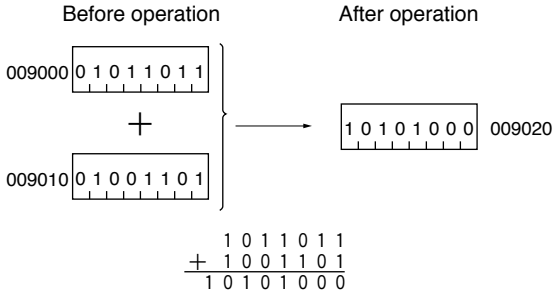
[Explanation]



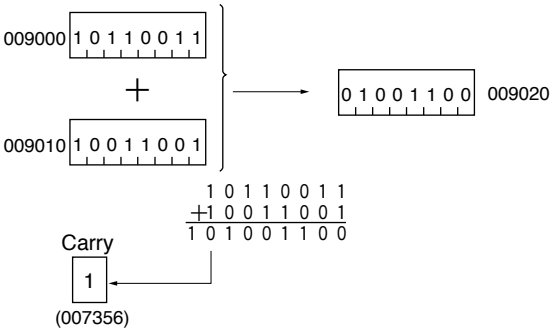
Instruction	
S T R	004001
F -210	
	009000
	009010
	009020

When the input condition 004001 changes from OFF to ON, the contents of the register 009000 and the register 009010 are added binary and stored in the register 009020.

- Example 1



- Example 2



F-210w ADD

Adds register and register in binary (16 bits + 16 bits) (ADD)

Symbol		F-210w ADD | S_1 | S_2 | D
Function		The contents of the registers S_1 and S_1+1 are added in binary with the contents of the registers S_2 and S_2+1 and its results are stored in the registers D and D+1.
Operation		(S_1, S_1+1) + (S_2, S_2+1) → D, D+1
S_1		Use range B *
S_2		Use range B *
D		Use range B *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1, S_1+1	Unchanged
	S_2, S_2+1	Unchanged
	D	Lower digits of result
	D+1	Upper digits of result

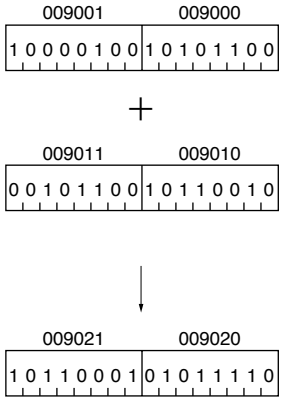
Flag	Result(8)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	0	1	0	0	1
	000001 to 177777	0	0	0	1
	200000	1	1	0	0
	Above 200001	0	1	0	0

[Explanation]

Instruction	
S T R	004001
F-210w	
	009000
	009010
	009020

004001 — F-210w ADD | 009000 | 009010 | 009020

When the input condition 004001 changes from OFF to ON, the contents of the registers 009000 and 009001 are added binary with the registers 009010 and 009011 and its results are stored in the registers 009020 and 009021.



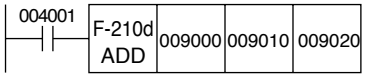
* Be sure to use even addresses for register S_1, S_2, and D (Odd addresses such as 019003 are prohibited).

F-210d ADD — Add register and register in binary (32 bits + 32 bits) (ADD)

Symbol		F-210d ADD | S_1 | S_2 | D
Function		The contents of the registers S_1 to S_1+3 are added in binary with the contents of the registers S_2 to S_2+3 and its result is stored in the registers D to D+3.
Operation		(S_1 to S_1+3) + (S_2 to S_2+3) → D to D+3
S_1		Use range C *
S_2		Use range C *
D		Use range C *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1 to S_1+3	Unchanged
	S_2 to S_2+3	Unchanged
	D to D+3	Result (32-bit binary)

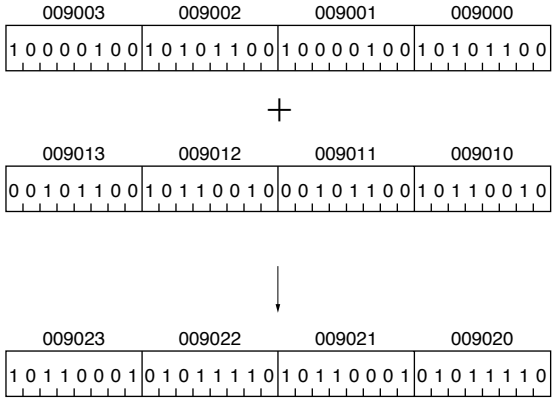
Flag: Result(8)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0	1
0 to 37777777777	0	0	0	1
40000000000	1	1	0	0
Above 40000000001	0	1	0	0

[Explanation]



Instruction	
S T R	004001
F -210d	
	009000
	009010
	009020

When the input condition 004001 changes from OFF to ON, the contents of the registers 009000 to 009003 are added binary with the registers 009010 to 009013 and its results are stored in the registers 009020 to 009023.



* Be sure to use even addresses for register S_1, S_2, and D (Odd addresses such as 019003 are prohibited).

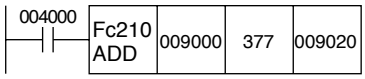
Fc210 ADD — Adds register and constant in binary (8 bits + 8 bits) (ADD)

Symbol		Fc210 ADD | S_1 | n | D
Function		Contents of the register S_1 are binary added with the contents of an octal constant and its result is stored in the register D.
Operation		S_1+n → D
S_1		Use range A *
n		Use range 000 to 377(8)
D		Use range A *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1	Unchanged
	D	Result
	Flag	Result (Octal) / Zero 007357 / Carry 007356 / Error 007355 / Non-carry 007354

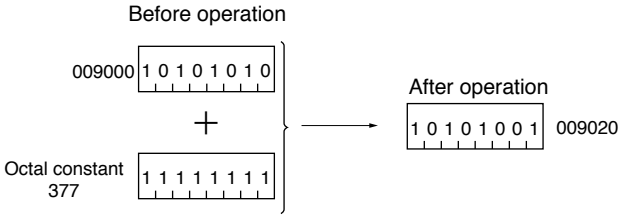
Result (Octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0	1
001 to 377	0	0	0	1
400	1	1	0	0
Above 401	0	1	0	0

[Explanation]

Instruction	
S T R	004000
Fc210	009000
	377
	009020



When the input condition 004000 changes from OFF to ON, the contents of the register 009000 are binary added with the octal constant 377 and its result is stored in the register 009020.



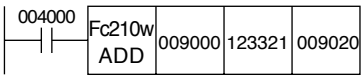
Fc210w ADD — Adds register and constant in binary (16 bits + 16 bits) (ADD)

Symbol		Fc210w ADD | S_1 | n | D
Function		Add octal constant n to register S_1 and S_1+1 contents and store the result to register D and D+1.
Operation		(S_1, S_1+1) + n → D, D+1
S_1		Use range B *
n		Use range 000000 to 177777(8)
D		Use range B *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1	Unchanged
	D	Lower digits of result
	D+1	Upper digits of result
	Flag	Result (octal) / Zero 007357 / Carry 007356 / Error 007355 / Non-carry 007354

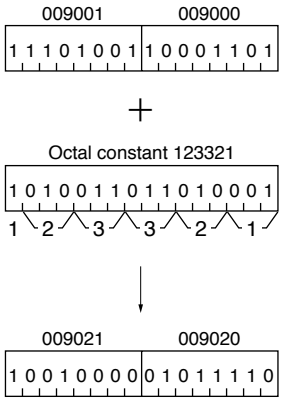
Result (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0	1
000001 to 177777	0	0	0	1
200000	1	1	0	0
Above 200001	0	1	0	0

[Explanation]

Instruction	
S T R	004000
Fc210w	009000
	123321
	009020



When the input condition 004000 changes from OFF to ON, the contents of the registers 009000 and 009001 are added in binary with the octal constant 123321 and its results are stored in the registers 009020 and 009021.



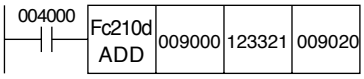
- Be sure to use even addresses for registers S_1 and D (Odd addresses such as 019003 are prohibited).

Fc210d ADD — Adds register and constant in binary (32 bits + 16 bits) (ADD)

Symbol		Fc210d ADD | S_1 | n | D
Function		Contents of the registers S_1 to S_1+3 are added in binary with an octal constant "n" and its result are stored in the registers D to D+3.
Operation		(S_1 to S_1+3) + n → D to D+3
S_1		Use range C *
S_2		Use range 000000 to 177777(8)
D		Use range C *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1 to S_1+3	Unchanged
	D to D+3	Result (32-bit binary)

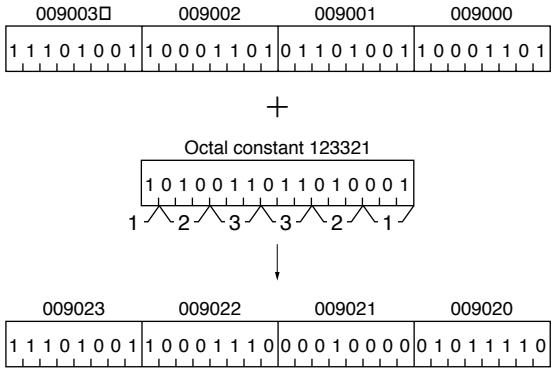
Flag: Result (Octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0	1
1 to 37777777777	0	0	0	1
40000000000	1	1	0	0
Above 40000000001	0	1	0	0

[Explanation]



Instruction	
S T R	004000
Fc210d	
	009000
	123321
	009020

When the input condition 004000 changes from OFF to ON, the contents of the registers 009000 to 009003 are added in binary with the octal constant 123321 and its result are stored in the registers 009020 to 009023.



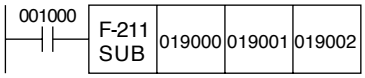
* Be sure to use even addresses for registers S_1 and D (Odd addresses such as 019003 are prohibited).

F-211 SUB — Subtracts register from register in binary (8 bits - 8 bits) (SUBtract)

Symbol		F-211 SUB | S_1 | S_2 | D
Function		Contents of the register S_1 are subtracted by the contents of the register S_2 and its result is stored in the register D.
Operation		$S_1 - S_2 \rightarrow D$
S_1		Use range B
S_2		Use range B
D		Use range B
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1	Unchanged
	S_2	Unchanged
	D	Result

Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	0	1	0	0	1
	001 to 377(8)	0	0	0	1
	Negative value	0	1	0	0

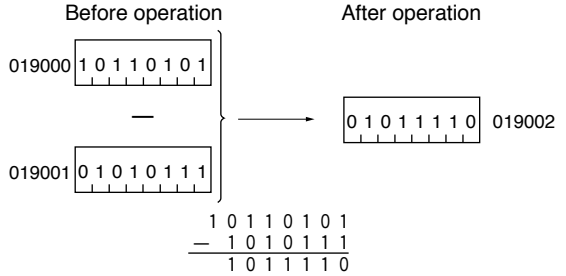
[Explanation]



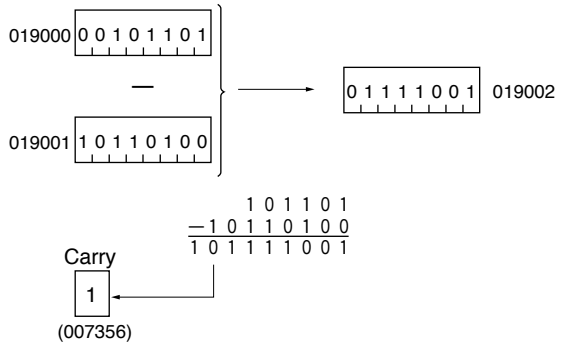
Instruction	
S T R	001000
F-211	
	019000
	019001
	019002

When the input condition 001000 changes from OFF to ON, the contents of the register 019000 are subtracted by the contents of the register 019001 and its result is stored in the register 019002.

- Example 1



- Example 2



F-211w SUB — Subtracts register from register in binary (16 bits - 16 bits) (SUBtract)

Symbol		F-211w SUB | S_1 | S_2 | D
Function		Contents of the registers S_1 and S_1+1 are subtracted from the registers S_2 and S_2+1 in the binary mode and its results are stored in the registers D and D+1.
Operation		(S_1, S_1+1) − (S_2, S_2+1) → D, D+1
S_1		Use range B *
S_2		Use range B *
D		Use range B *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1, S_1+1	Unchanged
	S_2, S_2+1	Unchanged
	D	Lower digits of result
	D+1	Upper digits of result

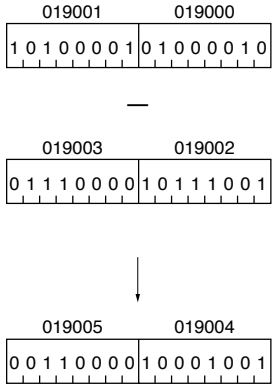
Flag: Result (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0	1
1 to 177777	0	0	0	1
Negative value	0	1	0	0

[Explanation]

Instruction	
S T R	001000
F-211w	019000
	019002
	019004

001000 | F-211w SUB | 019000 | 019002 | 019004

When the input condition 001000 changes from OFF to ON, the contents of the registers 019000 and 019001 are subtracted by the contents of the registers 019002 and 019003 and its results are stored in the registers 019004 and 019005.



- Be sure to use even addresses for registers S_1, S_2, and D (Odd addresses such as 019003 are prohibited).

F-211d SUB — Subtracts register from register in binary (32 bits - 32 bits) (SUBtract)

Symbol		F-211d SUB | S_1 | S_2 | D
Function		Contents of the registers S_1 to S_1+3 are subtracted from the registers S_2 to S_2+3 in the binary mode and its results are stored in the registers D and D+1.
Operation		(S_1 to S_1+3) − (S_2 to S_2+3) → D to D+3
S_1		Use range C *
S_2		Use range C *
D		Use range C *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1 to S_1+3	Unchanged
	S_2 to S_2+3	Unchanged
	D to D+3	Result (binary 32-bits)

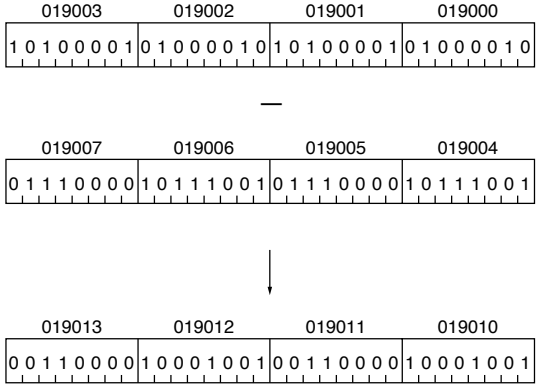
Flag: Result (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	1	0	0	1
1 to 37777777777	0	0	0	1
Negative value	0	1	0	0

[Explanation]

Instruction	
S T R	001000
F-211d	019000
	019004
	019010

001000 | F-211d SUB | 019000 | 019004 | 019010

When the input condition 001000 changes from OFF to ON, the contents of the registers 019000 to 019003 are subtracted by the contents of the registers 019004 to 019007 and its results are stored in the registers 019010 to 019013.



- Be sure to use even addresses for registers S_1, S_2, and D (Odd addresses such as 019003 are prohibited).

Fc211 SUB — Subtracts register from constant in binary (8 bits - 8 bits) (SUBtract)

Symbol		Fc210 ADD | S_1 | n | D
Function		Contents of the register S_1 are binary added with the contents of an octal constant and its result is stored in the register D.
Operation		S_1-n → D
S_1		Use range A *
n		Use range 000 to 377(8)
D		Use range A *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1	Unchanged
	D	Result

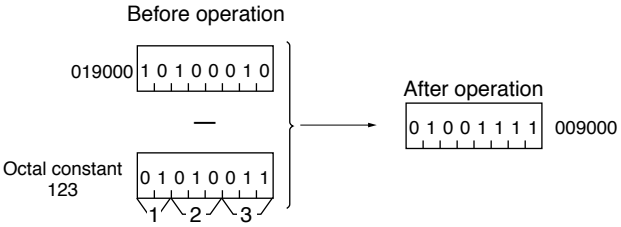
Contents after operation: Flag	Result (Octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	0	1	0	0	1
	1 to 377(8)	0	0	0	1
	Negative value	0	1	0	0

[Explanation]

Instruction	
S T R	004000
Fc211	
	019000
	123
	009000

004000 | Fc211 SUB | 019000 | 123 | 009000

When the input condition 004000 changes from OFF to ON, the octal constant 123 in the binary are subtracted from the contents of the register 019000 and its result is stored in the register 009000.



Fc211w SUB — Subtracts constant from register in binary (16 bits - 16 bits) (SUBtract)

Symbol		Fc211w SUB | S_1 | n | D
Function		Contents of the register S_1, S_1+1 are subtracted by the octal constant n and its result is stored in the register D, D+1.
Operation		S_1-S_2 → D
S_1		Use range B *
S_2		Use range B 000000 to 177777(8)
D		Use range B *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S1 to S1+1	Unchanged
	D	Lower digits of result
	D+1	Upper digits of result

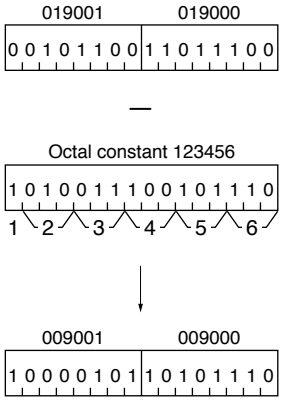
Contents after operation: Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	0	1	0	0	1
	1 to 177777(8)	0	0	0	1
	Negative value	0	1	0	0

[Explanation]

Instruction	
S T R	004000
Fc211w	
	019000
	123456
	009000

004000 | Fc211w SUB | 019000 | 123456 | 009000

When the input condition 004000 changes from OFF to ON, the contents of the octal constant 019000 and 019001 are subtracted from registers 123456 and its results are stored in the registers 009000 and 009001.



* Be sure to use even addresses for register S_1 and D (Odd addresses such as 019003 are prohibited).

Fc211d SUB — Subtracts constant from register in binary (16 bits - 16 bits) (SUBtract)

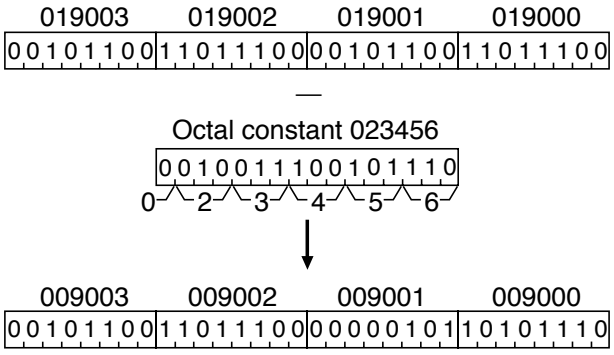
Symbol		Fc211d SUB | S_1 | n | D
Function		Contents of the register S_1 to S_{1+3} are subtracted by the contents of the register S_2 and its result is stored in the register D to D+3.
Operation		(S_1 to S_1+3) − n → D to D+3
S_1		Use range C *
S_2		Use range 000000 to 177777(8)
D		Use range C *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1 to S_1+3	Unchanged
	D to D+3	Result (32-bit binary)
	Flag	Result | Zero 007357 | Carry 007356 | Error 007355 | Non-carry 007354
		0 | 1 | 0 | 0 | 1
		1 to 37777777777(8) | 0 | 0 | 0 | 1
		Negative value | 0 | 1 | 0 | 0

[Explanation]

Instruction	
S T R	004000
Fc211d	
	019000
	023456
	009000

004000 | Fc211d SUB | 19000 | 123456 | 09000

When the input condition 004000 changes from OFF to ON, the contents of the octal constant 019000 and 019003 are subtracted from registers 023456 and its results are stored in the registers 009000 and 009003.



* Be sure to use even addresses for register S_1 and D (Odd addresses such as 019003 are prohibited).

F-212
WNDW

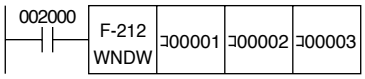
Window comparator (between 1-byte registers)

Symbol		F-212 WNDW | S_1 | S_2 | S_3
Function		Compares contents of register S_1 with the contents of the registers S_2 and S_3 and stores its results in the flags.
Operation		Compare result→flag
S_1		Use range A
S_2		Use range A
D		Use range A
Condition		When the input signal is ON (not limited to an OFF to ON change)
Contents after operation	S_1	Unchanged
	S_2	Unchanged
	D	Unchanged

Contents after operation: Flag	Contents of register	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	$S_1 < S_2$	1	1	0	0
	$S_2 \leqq S_1 \leqq S_3$	0	0	0	0
	$S_3 < S_1$	0	0	0	1
	$S_3 < S_2$	0	0	0	0

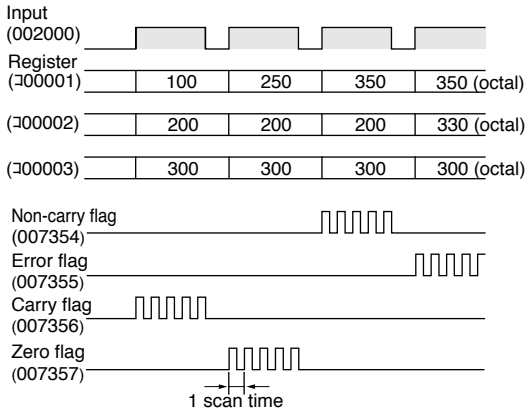
[Explanation]

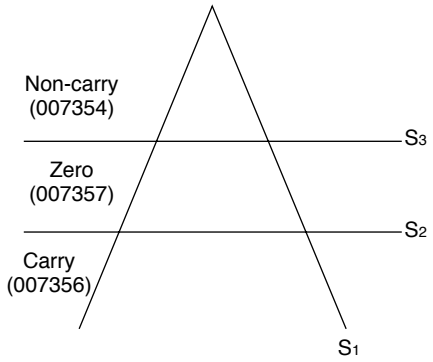
Instruction	
S T R	002000
F-212	
	⊐00001
	⊐00002
	⊐00003



When the input condition 002000 changes from OFF to ON, the contents of the register ⊐00001 are checked if ⊐00001 < ⊐00002, ⊐00002 ≦ ⊐ 00001 ≦ ⊐00003, and ⊐00003 < ⊐00001, its results are stored in the carry flag zero flag and non carry flag.

Operation takes place only when the condition is ⊐00002 ≦ ⊐00003. Operation does not take place and the error flag is set active, if ⊐00002 ≦ ⊐00003.





F-212w WNDW

Window comparator (between 1-word registers)

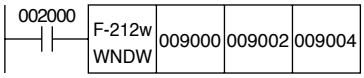
Symbol		F-212w WNDW | S_1 | S_2 | S_3
Function		Compares contents of the registers S_1, S_1+1 with registers S_2, S_2+1 and registers S_3, S_3+1 and stores their result in the flags.
Operation		Compare result → flag
S_1		Use range B *
S_2		Use range B *
D		Use range B *
Condition		When the input signal is ON (not limited to an OFF to ON change)
Contents after operation	S_1	Unchanged
	S_2	Unchanged
	D	Unchanged

Flag: Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
S_1, S_1+1 < S_2, S_2+1	1	1	0	0
S_2, S_2+1 ≦ S_1, S_1+1 ≦ S_3, S_3+1	0	0	0	0
S_3, S_3+1 < S_1, S_1+1	0	0	0	1
S_3, S_3+1 < S_2, S_2+1	0	0	1	0

* Be sure to use even addresses for registers S_1,S_2 and S_3 (Odd addresses such as 009003 are prohibit).

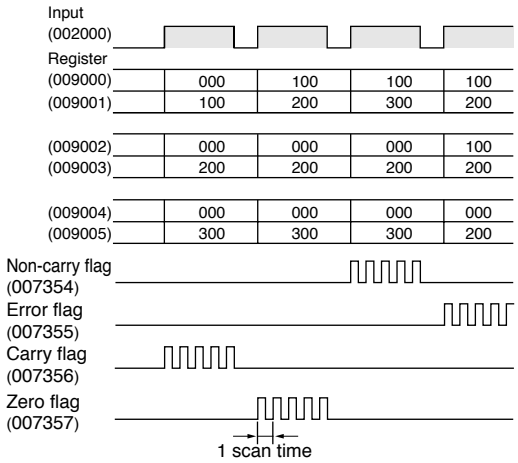
[Explanation]

Instruction	
S T R	002000
F-212w	
	009000
	009002
	009004



When the input condition 002000 changes from OFF to ON, the word contents of the registers 009000 and 009001 (1 word), are checked if (009000, 009001) < (090002, 009003) and (009002, 009003) ≦ (009000, 009001) ≦ (009004, 009005), and (009004, 009005) < (009000, 009001), and its results are stored in the carry flag, zero flag and non carry flag.

Operation takes place only when (009002, 009003) ≦ (009004, 009005) is established and operation stop with the error flag in activation if (009004, 009005) < (009002, 009003.)



F-212d WNDW Window comparator (between 2-word registers)

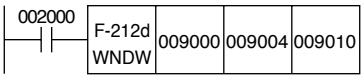
Symbol		F-212d WNDW | S_1 | S_2 | S_3
Function		Compares contents of the registers S_1 to S_1+3 with the registers S_2 to S_2+3, S_2 to S_2+3 and stores its result in the flags.
Operation		Compare result → flag
S_1		Use range C *
S_2		Use range C *
D		Use range C *
Condition		When the input signal is ON (not limited to an OFF to ON change)
Contents after operation	S_1 to S_1+3	Unchanged
	S_2 to S_2+3	Unchanged
	S_3 to S_3+3	Unchanged

Flag: Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
S_1, S_1+1 < S_2, S_2+1	1	1	0	0
S_2, S_2+1 ≦ S_1, S_1+1 ≦ S_3, S_3+1	0	0	0	0
S_3, S_3+1 < S_1, S_1+1	0	0	0	1
S_3, S_3+1 < S_2, S_2+1	0	0	1	0

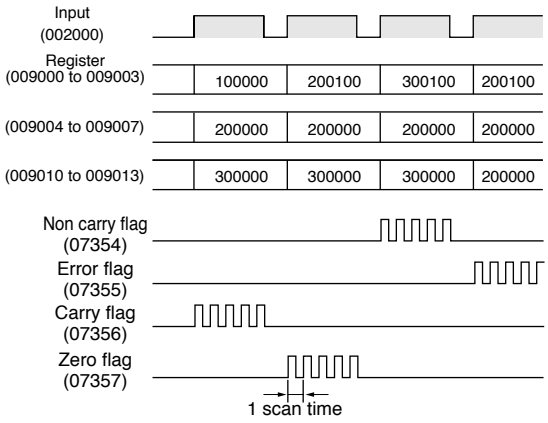
* Be sure to use even addresses for registers S_1,S_2 and S_3 (Odd addresses such as 009003 are prohibit).

[Explanation]

Instruction	
S T R	002000
F-212d	
	009000
	009004
	009010



When the input condition of 002000 is ON, this instruction checks the 2-word contents of registers 009000 to 009003 (2 words) whether they fall in the any of (009000 to 009003) < (009004 to 009007) or (009004 to 009007) ≦ (009000 to 009003) ≦ (009010 to 009013) or (009010 to 009013) < (009000 to 009003), and sets the carry flag accordingly.
This instruction is executed only if (009004 to 009007) ≦ (009010 to 009013). If (009010 to 009013) ≦ (009004 to 009007), execution is aborted, and the error flag is set.



Fc212 WNDW — Window comparator (between 1-byte octal constants)

Symbol		Fc212 WNDW | S | n_1 | n_2
Function		Compares contents of the register S_1 with the octal constants n_1 and n_2 and stores its results in the flags.
Operation		Result → flag
S_1		Use range A
n_1		Use range 000 to $377_{(8)}$
n_2		Use range 000 to $377_{(8)}$
Condition		When the input signal is ON (not limited to an OFF to ON change)
Contents after operation	S_1	Unchanged
	Flag	See table below

Contents of register	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
S1<n1	0	1	0	0
n1≦S1≦n2	1	0	0	0
n2<S1	0	0	0	1
n2<n1	0	0	1	0

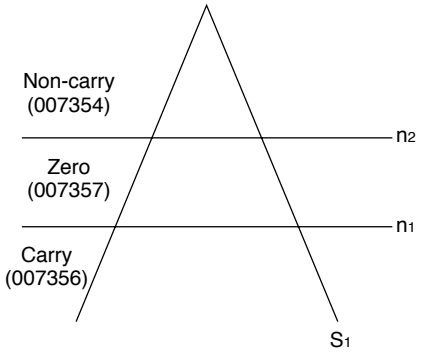
[Explanation]

Instruction	
S T R	004000
Fc212	
	019000
	200
	300

004000 — Fc212 WNDW | 019000 | 200 | 300

When the input condition 004000 changes from OFF to ON, the contents of the register 019000 are checked if (019000) < $200_{(8)}$, $200_{(8)}$ ≦ (019000) ≦ $300_{(8)}$, and $300_{(8)}$ < (019000), and its results are stored in the carry flag, zero flag and non-carry flag.

Contents of 019000 (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
150	0	1	0	0
250	1	0	0	0
350	0	0	0	1



Fc212w WNDW — Window comparator (between 1-word octal constants)

Symbol		Fc212w WNDW | S_1 | n_1 | n_2
Function		Compares contents of the registers S_1 and S_1+1 with the octal constants n_1 and n_2, and its results are stored in the flags.
Operation		Result → flag
S_1		Use range B
n_1		Use range 000000 to $177777_{(8)}$
n_2		Use range 000000 to $177777_{(8)}$
Condition		When the input signal is ON (not limited to an OFF to ON change)
Contents after operation	S_1, S_1+1	Unchanged
	Flag	See table below

Contents of register	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
S_1, S_1+1 < n_1	0	1	0	0
n_1 ≦ S_1, S_1+1 ≦ n_2	1	0	0	0
n_2 < S_1, S_1+1	0	0	0	1
n_2 < n_1	0	0	1	0

[Explanation]

Instruction	
S T R	004000
F c212w	
	019000
	020000
	030000

004000 — Fc212w WNDW | 019000 | 020000 | 030000

When the input condition 004000 changes from OFF to ON, the word contents of the registers 019000 and 019001 (1word) are checked if (019000, 019001) < 020000, 020000 ≦ (019000, 019001) ≦ 030000, and 030000 < (019000, 019001) and its results are stored in the carry flag, zero flag and non-carry flag.

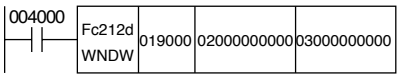
Contents of 019000,019001	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
015000	0	1	0	0
025000	1	0	0	0
035000	0	0	0	1

Fc212d WNDW Window comparator (between 2-word octal constants)

Symbol		Fc212d WNDW | S_1 | n_1 | n_2
Function		Compares contents of the registers S_1 to S_1+3 (2-word data) with the octal constants n_1 and n_2, and its results are stored in the flags.
Operation		Result → flag
S_1		Use range C
n_1		Use range 00000000000 to 37777777777(8)
n_2		Use range 00000000000 to 37777777777(8)
Condition		When the input signal is ON (not limited to an OFF to ON change)
Contents after operation	S_1 to S_1+3	Unchanged

Contents after operation: Flag	Contents of register	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	S_1 to S_1+3 < n_1	0	1	0	0
	n_1 ≦ S_1 to S_1+1 ≦ n_2	1	0	0	0
	n_2 < S_1, S_1+1	0	0	0	1
	n_2 < n_1	0	0	1	0

[Explanation]



Instruction
S T R 004000
F c212d 019000
02000000000
03000000000

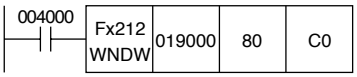
When the input condition 004000 changes from OFF to ON, the word contents of the registers 019000 to 019003 (2-word) are checked if (019000 to 019003) < 02000000000, 02000000000 ≦ (019000 to 019003) ≦ 03000000000, and 03000000000 < (019000 to 019003) and its results are stored in the carry flag, zero flag and non-carry flag.

Contents of 019000 to 019003	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
01500000000	0	1	0	0
02500000000	1	0	0	0
03500000000	0	0	0	1

Fx212 WNDW — Window comparator (between 1-byte hexadecimal constants)

Symbol		Fx212 WNDW, S_1, n_1, n_2
Function		Compares contents of the register S_1 with the hexadecimal constants n_1 and n_2 and stores its results in the flags.
Operation		Result → flag
S_1		Use range A
n_1		Use range 00 to $FF_{(8)}$
n_2		Use range 00 to $FF_{(8)}$
Condition		When the input signal is ON (not limited to an OFF to ON change)
Contents after operation	S_1	Unchanged
	Flag	Contents of register / Zero 007357 / Carry 007356 / Error 007355 / Non-carry 007354
		S1 < n1 — 0 / 1 / 0 / 0
		n1 ≦ S1 ≦ n2 — 1 / 0 / 0 / 0
		n2 < S1 — 0 / 0 / 0 / 1
		n2 < n1 — 0 / 0 / 1 / 0

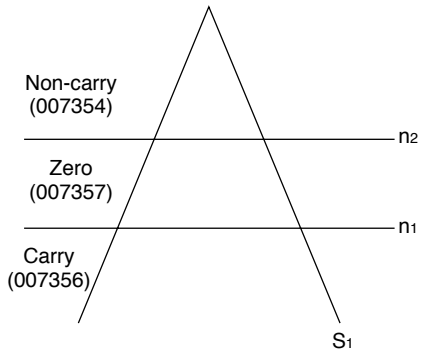
[Explanation]



Instruction	
S T R	004000
Fx212	019000
	80
	C0

When the input condition 004000 changes from OFF to ON, the contents of the register 019000 are checked if (019000) < $80_{(H)}$, $80_{(H)}$ ≦ (019000) ≦ $C0_{(H)}$, and $C0_{(H)}$ < (019000), and its results are stored in the carry flag, zero flag and non-carry flag.

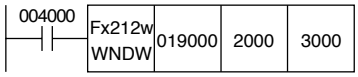
Contents of 019000 (hexadecimal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
70	0	1	0	0
90	1	0	0	0
D0	0	0	0	1



Fx212w WNDW — Window comparator (between 1-word hexadecimal constants)

Symbol		Fx212w WNDW, S_1, n_1, n_2
Function		Compares contents of the registers S_1 and S_1+1 (1-word) with the hexadecimal constants n_1 and n_2, and its results are stored in the flags.
Operation		Result → flag
S_1		Use range B
n_1		Use range 0000 to $FFFF_{(8)}$
n_2		Use range 0000 to $FFFF_{(8)}$
Condition		When the input signal is ON (not limited to an OFF to ON change)
Contents after operation	S_1, S_1+1	Unchanged
	Flag	Contents of register / Zero 007357 / Carry 007356 / Error 007355 / Non-carry 007354
		S_1, S_1+1 < n_1 — 0 / 1 / 0 / 0
		n_1 ≦ S_1, S_1+1 ≦ n_2 — 1 / 0 / 0 / 0
		n_2 < S_1, S_1+1 — 0 / 0 / 0 / 1
		n_2 < n_1 — 0 / 0 / 1 / 0

[Explanation]



Instruction	
S T R	004000
Fx212w	019000
	2000
	3000

When the input condition 004000 changes from OFF to ON, the word contents of the registers 019000 and 019001 (1 word) are checked if (019000, 019001) < 2000, 2000 ≦ (019000, 019001) ≦ 3000, and 3000 < (019000, 019001) and its results are stored in the carry flag, zero flag and non-carry flag.

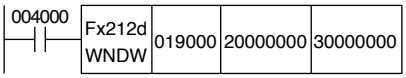
Contents of 019000,019001	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
1500	0	1	0	0
2500	1	0	0	0
3500	0	0	0	1

Fx212d WNDW — Window comparator (between 2-word hexadecimal constants)

Symbol		Fx212d WNDW | S_1 | n_1 | n_2
Function		Compares contents of the registers S_1 to S_1+3 (2-word) with the hexadecimal constants n_1 and n_2, and its results are stored in the flags.
Operation		Result → flag
S_1		Use range C
n_1		Use range 00000000 to FFFFFFFF(H)
n_2		Use range 00000000 to FFFFFFFF(H)
Condition		When the input signal is ON (not limited to an OFF to ON change)
Contents after operation	S_1 to S_1+3	Unchanged

Contents after operation: Flag	Contents of register	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	S_1 to S_1+3 < n_1	0	1	0	0
	$n_1 \leqq S_1$ to S_1+3 $\leqq n_2$	1	0	0	0
	$n_2 < S_1$ to S_1+3	0	0	0	1
	$n_2 < n_1$	0	0	1	0

[Explanation]



Instruction	
S T R	004000
Fx212d	
	019000
	20000000
	30000000

When the input condition 004000 changes from OFF to ON, the word contents of the registers 019000 to 019003 (2-word) are checked if (019000, 019003) < 20000000, 20000000 ≦ (019000, 019003) ≦ 30000000, and 30000000 < (019000 to 019003) and its results are stored in the carry flag, zero flag and non-carry flag.

Contents of 019000,019001	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
15000000	0	1	0	0
25000000	1	0	0	0
35000000	0	0	0	1

F-215 MUL — Multiply register by register in binary (8 bits × 8 bits) (MULtiply)

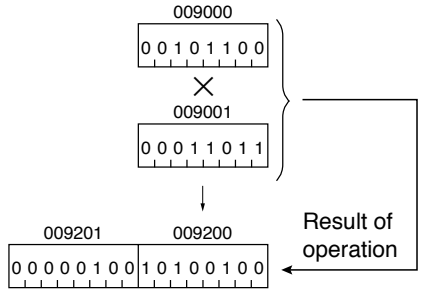
Symbol		F-215 MUL | S_1 | S_2 | D
Function		Binary multiply register S_1 content and register S_2 content and store the result to D and D+1.
Operation		$S_1 \times S_2 \rightarrow$ D, D+1
S_1		Use range A
S_2		Use range A
D		Use range B
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1	Unchanged
	S_2	Unchanged
	D	Lower digits of result
	D+1	Upper digits of result
	Flag	Zero 007357 | Carry 007356 | Error 007355 | Non-carry 007354
		0 | 0 | 0 | 0

[Explanation]

Instruction	
S T R	004000
F-215	009000
	009100
	009200

004000 — F-215 MUL | 09000 | 009100 | 009200

When the input condition 004000 changes from OFF to ON, the contents of 009000 are multiplied by the contents of the register 009100 in binary and its results are stored in the registers 009200 and 009201.




```
  1 0 1 1 0 0
×   1 1 0 1 1
  1 0 1 1 0 0
1 0 1 1 0 0
1 0 1 1 0 0
1 0 1 1 0 0
1 0 0 1 0 1 0 0 1 0 0
```

F-215w MUL — Multiply register by register in binary (16 bits × 16 bits) (MULtiply)

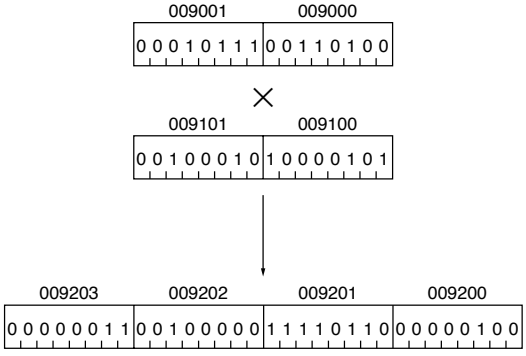
Symbol		F-215w MUL | S_1 | S_2 | D
Function		Binary multiply register S_1, S_1+1 content and register S_2, S_2+1 content and store the result to D to D+3.
Operation		(S_1, S_1+1) × (S_2, S_2+1) → D to D+3
S_1		Use range B *
S_2		Use range B *
D		Use range C *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1	Unchanged
	S_2	Unchanged
	D	Lower digits of result
	D+1	Result
	D+2	Result
	D+3	Upper digits of result
	Flag	Zero 007357 | Carry 007356 | Error 007355 | Non-carry 007354
		0 | 0 | 0 | 0

[Explanation]

Instruction	
S T R	004000
F-215w	009000
	009100
	009200

004000 — F-215w MUL | 009000 | 009100 | 009200

When the input condition 004000 changes from OFF to ON, the 16-bit contents of the registers 009000 and 009001(16-bit data) are multiplied by the contents of the registers 009100 and 009101 (16-bit data)in binary and its results are stored in the registers 009200 to 009203.



* Be sure to use even addresses for registers S_1, S_2 and D.

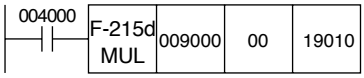
F-215d MUL — Multiplies register by register in binary (32 bits × 32 bits) (MULtiply)

Symbol		F-215d MUL | S_1 | S_2 | D			
Function		Binary multiply register S_1 to S_1+3 content and register S_2 to S_2+3 content and store the result to D to D+7.			
Operation		(S_1 to S_1+3) × (S_2 to S_2+3) → D to D+7			
S_1		Use range C *			
S_2		Use range C *			
D		Use range G *□			
Condition		Rising edge of input signal (OFF to ON)			
Contents after operation	S_1 to S_1+3	Unchanged			
	S_2 to S_2+3	Unchanged			
	D to D+7	Result (64-bit binary)			
	Flag	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		0	0	0	0

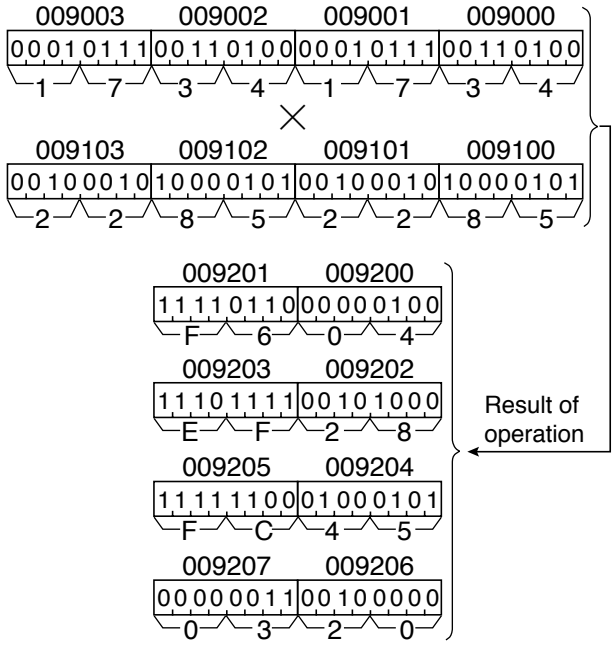
* Be sure to use even addresses for registers S_1, S_2 and D (Odd addresses such as 019003 are prohibited),

[Explanation]

Instruction	
S T R	004000
F-215d	
	009000
	009100
	009200



When the input condition 004000 changes from OFF to ON, instruction multiplies the 32-bit contents of registers 009000 to 009003 by the 32-bit contents of registers 009100 to 009103, and stored the result in registers 009200 to 009207.



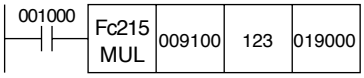
In this example, $17341734_{(H)} \times 22852285_{(H)} = 0320FC45EF25FR604_{(H)}$ is performed.

Fc215 MUL — Multiplies register by constant in binary (8 bits × 8 bits) (MULtiply)

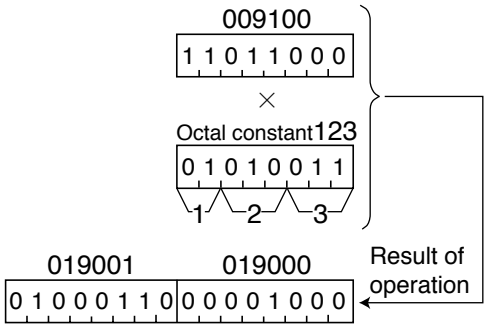
Symbol		Fc215 MUL | S_1 | S_2 | D			
Function		Binary multiply register S_1 content and octal constant n and store the result to D and D+1.			
Operation		$S_1 \times n \rightarrow D, D+1$			
S_1		Use range A			
S_2		Use range 000 to 377(8)			
D		Use range B			
Condition		Rising edge of input signal (OFF to ON)			
Contents after operation	S_1 to S_1+3	Unchanged			
	S_2 to S_2+3	Lower digits of result			
	D to D+7	Upper digits of result			
	Flag	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		0	0	0	0

[Explanation]

Instruction	
S T R	01000
Fc215	009100
	123
	019000



When the input condition 001000 changes from OFF to ON, the contents of the register 009100 are multiplied by the octal constant 123 in binary and its results are stored in the registers 019000 and 019001.

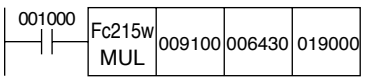


Fc215w MUL — Multiplies register by constant in binary (16 bits × 16 bits) (MULtiply)

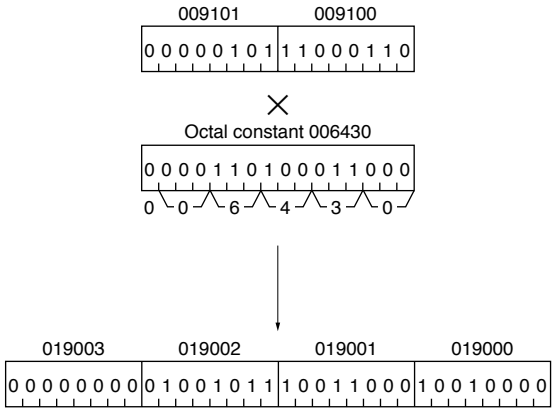
Symbol		Fc215w MUL | S_1 | S_2 | D			
Function		Binary multiply register S_1, S_1+1 content (16-bit data) and octal constant n and store the result to D to D+3.			
Operation		(S_1, S_1+1) × n → D to D+3			
S_1		Use range B *			
S_2		Use range 000000 to 177777(8)			
D		Use range C *			
Condition		Rising edge of input signal (OFF to ON)			
Contents after operation	S_1, S_1+1	Unchanged			
	D	Lower digits of result			
	D+1	Result			
	D+2	Result			
	D+3	Upper digits of result			
	Flag	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		0	0	0	0

[Explanation]

Instruction	
S T R	01000
Fc215w	009100
	006430
	019000



When the input condition 001000 changes from OFF to ON, the 16-bit contents of the registers 009100 and 009101 are multiplied by the octal constant 006430 in binary and its results are stored in the registers 019000 to 019003.



* Be sure to use even addresses for registers S_1 and D.

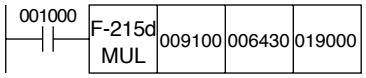
F-215d MUL — Multiplies register by register in binary (32 bits × 16 bits) (MULtiply)

Symbol		F-215d MUL | S_1 | S_2 | D			
Function		Binary multiply register S_1 to S_1+3 content (32-bit data) and octal constant n and store the result to D to D+7.			
Operation		(S_1 to S_1+3) × n → D to D+7			
S_1		Use range C *			
S_2		Use range 000000 to 177777(8)			
D		Use range G *			
Condition		Rising edge of input signal (OFF to ON)			
Contents after operation	S_1 to S_1+3	Unchanged			
	D to D+7	Result (64-bit binary)			
	Flag	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		0	0	0	0

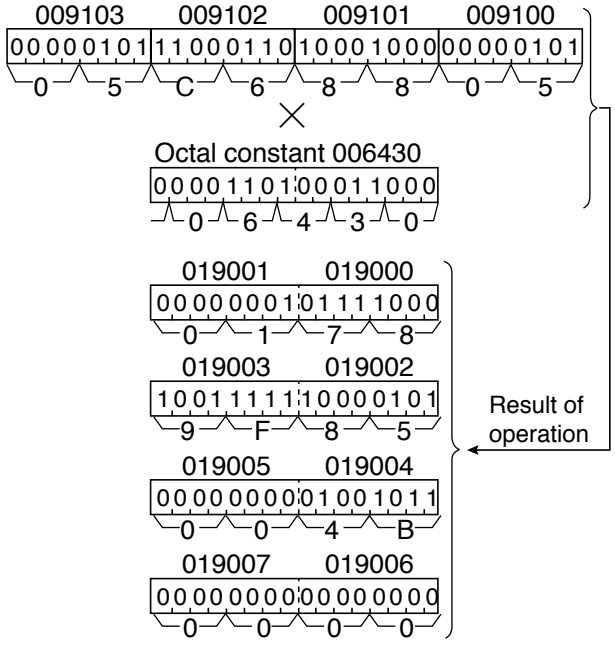
* Be sure to use even addresses for registers S_1 and D.(Odd addresses such as 019003 are prohibited).

[Explanation]

Instruction	
S T R	001000
F-215d	
	009100
	006430
	019000



When the input condition 001000 changes from OFF to ON, instruction multiplies the 32-bit contents of registers 009100 to 009003 by octal constant 006430, and stored the result in registers 019000 to 009007.

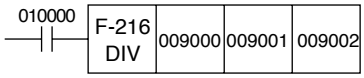


In this example, 05C68805(H) × 006430(8) = 4B9F850178(H) is performed.

F-216 DIV — Divides register by register in binary (8 bits ÷ 8 bits) (DIVide)

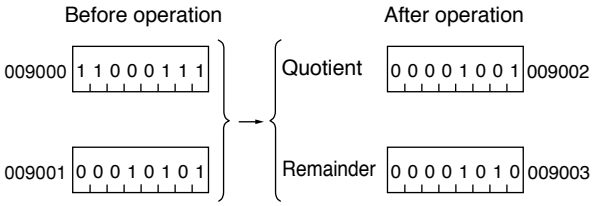
Symbol		F-216 DIV | S_1 | S_2 | D				
Function		Divides the 8-bit contents of the register S_1 by the 8-bit contents of the register S_2 in binary and stores its quotient in the register D and remainder in the register D+1.				
Operation		$S_1 \div S_2 \rightarrow$ D, D+1				
S_1		Use range A				
S_2		Use range A				
D		Use range B				
Condition		Rising edge of input signal (OFF to ON)				
Contents after operation	S_1	Unchanged				
	S_2	Unchanged				
	D	Quotient	Unchanged if the contents of the register S_2 is 000(8)			
	D+1	Remainder				
	Flag	Contents of S_2	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		000 (8)	0	0	0	0
		Other than above			0	

[Explanation]



Instruction	
S T R	010000
F-216	
	009000
	009001
	009002

When the input condition 010000 changes from OFF to ON, the contents of the register 009000 are divided by the contents of the register 009001 in binary and its quotient is stored in the registers 009002 and remainder in the register 009003.




```
          1 0 0 1
1 0 1 0 1 ) 1 1 0 0 0 1 1 1
            1 0 1 0 1
            1 1 1 1 1
            1 0 1 0 1
            1 0 1 0
```

F-216w DIV — Divides register by register in binary (15 bits ÷ 15 bits) (DIVide)

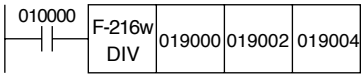
Symbol		F-216w DIV | S_1 | S_2 | D
Function		Divides the 15-bit contents of the register S_1, S_1 +1 by the 15-bit contents of the register S_2, S_2+1 in binary and stores its quotient in the register D, D+1 and remainder in the register D+2, D+3.
Operation		(S_1, S_1+1) ÷ (S_2, S_2+1) → D, D+1, D+2, D+3
S_1		Use range B *
S_2		Use range B *
D		Use range C *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1, S_1+1	Unchanged
	S_2, S_2+1	Unchanged
	D	Lower digits of quotient (lower) — Unchanged if the contents of the registers S_2 and S_2+1 are 000000(8)
	D+1	Upper digits of quotient (upper)
	D+2	Lower digits of remainder (lower)
	D+3	Upper digits of remainder (upper)

Flag

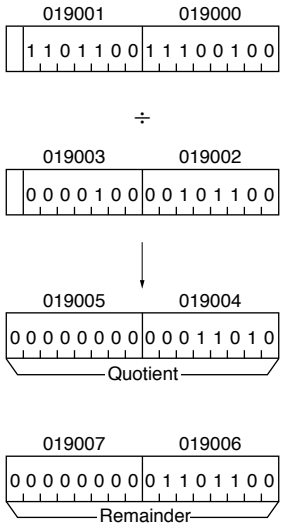
Contents of S_2, S_2+1	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
000000 (8)	0	0	0	0
Other than above			0	

[Explanation]

Instruction	
S T R	010000
F -216w	019000
	019002
	019004



When the input condition0 10000 changes from OFF to ON, the 15-bit contents of the registers 019000 and 019001 are divided by the 15-bit contents of the registers 019002 and 019003 in binary and its quotient is stored in the registers 019004 and 019005 and the remainder in 019006 and 019007.



- MSB (bit 7) of the registers 019001 and 019003 will be disregarded.

* Be sure to use even addresses for registers S_1, S_2 and D.

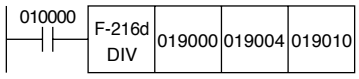
F-216d DIV — Divides register by register in binary (31 bits ÷ 31 bits) (DIVide)

Symbol		F-216d DIV | S_1 | S_2 | D				
Function		Divides the 31-bit contents of the registers S_1 to S_1+3 by the 31-bit contents of the registers S_2 to S_2+3 and stores its quotient in the registers D to D+3 and remainder in the registers D+4 to D+7.				
Operation		(S_1 to S_1+3) ÷ (S_2 to S_2+3) → D to D+7				
S_1		Use range C *				
S_2		Use range C *				
D		Use range G *				
Condition		Rising edge of input signal (OFF to ON)				
Contents after operation	S_1 to S_1+3	Unchanged				
	S_2 to S_2+3	Unchanged				
	D to D+3	Quotient (31-bit binary)	Unchanged if the contents of registers S_2 to S_2+3 are 0. (Does not calculate.)			
	D+4 to D+7	Remainder (31-bit binary)				
	Flag	Contents of S_2 to S_2+3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		000000 (8)	0	0	0	0
		Other than above			0	

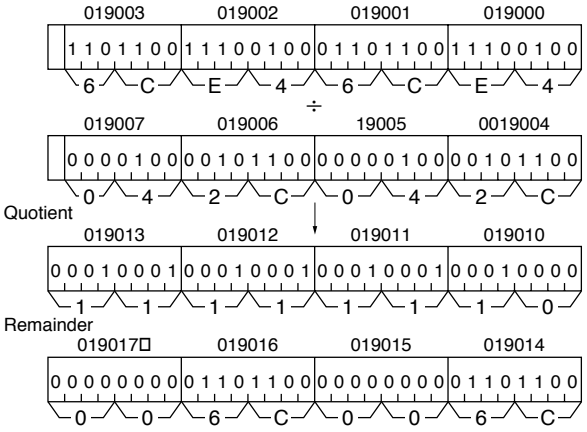
* Be sure to use even addresses for registers S_1, S_2, and D. (Odd addresses such as 019003 etc. are prohibited to use.)

[Explanation]

Instruction	
S T R	010000
F-216d	
	019000
	109004
	019010



When the input condition 010000 changes from OFF to ON, the 31-bit contents of the register 019000 to 019003 are divided by the 31-bit contents of the register 019004 to 019007 in binary and its quotient is stored in the registers 019010 to 019013 and the remainder in 019014 to 019017.
In this example, 6CE46CE4(H) ÷ 042C042C(H)=1A(H) remainder 6C006C is performed.



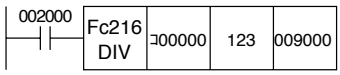
- MSB (bit 7) of the registers 019003 and 019007 will be ignored.

Fc216 DIV — Divides register by register in binary (8 bits ÷ 8 bits) (DIVide)

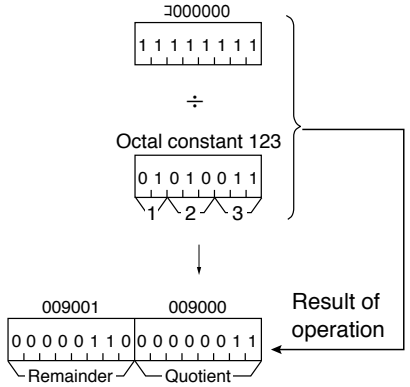
Symbol		Fc216 DIV | S_1 | n | D			
Function		Divides the 8-bit contents of the register S_1 by the 8-bit contents of the register S_2 in binary and stores its quotient in the register D and remainder in the register D+1.				
Operation		$S_1 \div n \rightarrow D, D+1$				
S_1		Use range A				
n		Use range 000 to $377_{(8)}$				
D		Use range B				
Condition		Rising edge of input signal (OFF to ON)				
Contents after operation	S	Unchanged				
	D	Quotient	Unchanged if "n" =$000_{(8)}$.			
	D+1	Remainder				
	Flag	Octal constant "n"	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		000 (8)	0	0	1	0
		Other than above			0	

[Explanation]

Instruction	
S T R	02000
Fc216	
	⊐00000
	123
	009000



When the input condition 002000 changes from OFF to ON, the contents of the register ⊐0000 are divided by the octal constant 123 and its quotient is stored in the register 009000 and the remainder in 009001.

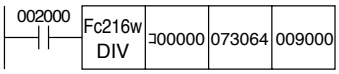


Fc216w DIV — Divides register by register in binary (15 bits ÷ 15 bits) (DIVide)

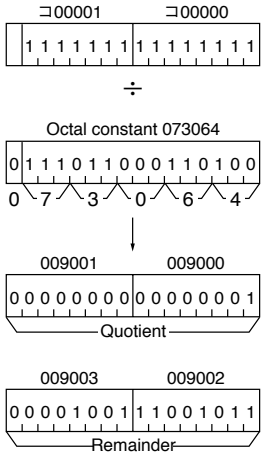
Symbol		Fc216w DIV | S_1 | S_2 | D				
Function		Divides the 8-bit contents of the register S_1 to S_1+1 by the 8-bit contents of the register S_2 in binary and stores its quotient in the register D, D+1 and remainder in the register D+2,D+3.					
Operation		(S_1, S_1+1) ÷ n → D, D+1, D+2, D+3					
S_1		Use range B *					
n		Use range 000000 to $177777_{(8)}$					
D		Use range C *					
Condition		Rising edge of input signal (OFF to ON)					
Contents after operation	S_1, S_1+1	Unchanged	Unchanged if "n" =$000000_{(8)}$.				
	D	Lower digits of quotient					
	D+1	Upper digits of quotient					
	D+2	Lower digits of remainder					
	D+3	Upper digits of remainder					
	Flag	Octal constant "n"	Zero 007357	Carry 007356	Error 007355	Non-carry 007354	
		000000(8)	0	0	0	0	
		Other than above			0		

[Explanation]

Instruction	
S T R	020000
Fc216w	
	⊐00000
	073064
	009000



When the input condition 002000 changes from OFF to ON, the 15-bit contents of the registers ⊐00000 and ⊐00001 are divided by the octal constant 073064 in binary and its quotient is stored in the registers 009000 and 009001 and the remainder in 009002 and 009003.



- MSB of the register ⊐00001 will be ignored.

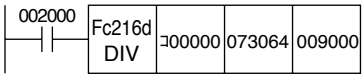
* Be sure to use even addresses for registers S_1 and D.

Fc216d DIV — Divides register by register in binary (31 bits ÷ 15 bits) (DIVide)

Symbol		Fc216d DIV | S1 | n | D				
Function		Divides the 31-bit contents of the registers S1 to S1+3 by the 31-bit contents of the registers S2 to S2+3 and stores its quotient in the registers D to D+3 and remainder in the registers D+4 to D+7.				
Operation		(S1 to S1+3) ÷ n → D to D+7				
S1		Use range C *				
n		Use range 000000 to 077777(8)				
D		Use range G *				
Condition		Rising edge of input signal (OFF to ON)				
Contents after operation	S1 to S1+3	Unchanged				
	D to D+3	Quotient (31-bit binary)	Unchanged if "n" =0.			
	D+4 to D+7	Remainder (31-bit binary)				
	Flag	Octal constant "n"	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		000000	0	0	0	0
		Other than above			0	

* Be sure to use even addresses for registers S1 and D.

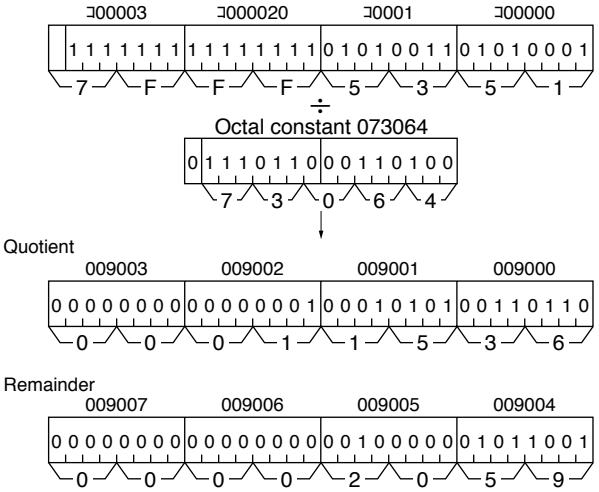
[Explanation]



Instruction	
S T R	002000
Fc216d	
	⊐00000
	073064
	009000

When the input condition 002000 changes from OFF to ON, the 31-bit contents of the registers ⊐00000 to ⊐00003 are divided by the octal constant 073064 in binary and its quotient is stored in the registers 009000 to 090003 and the remainder in 009004 to 009007.

In this example, 7FFF5351(H) ÷ 73064(8)=11536(H), remainder 2059(H) is performed.



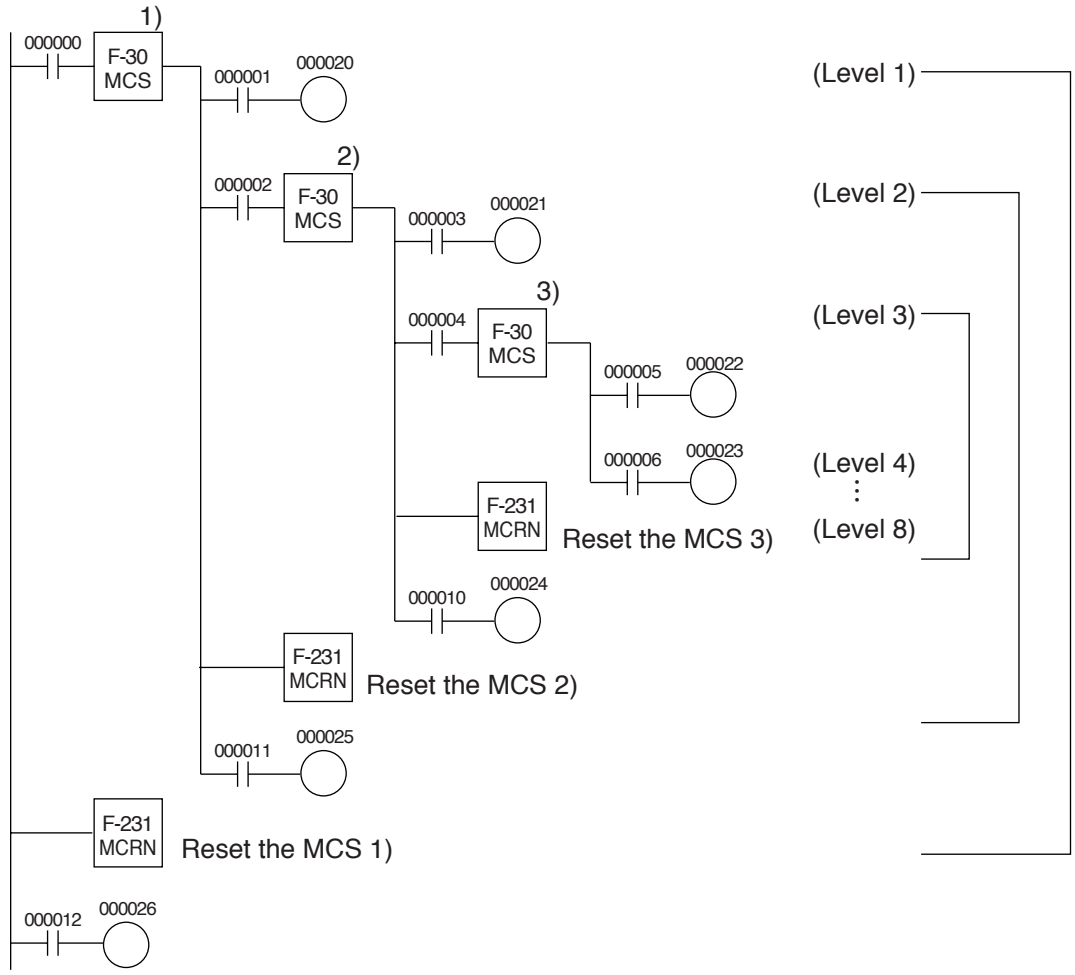
- MSB (bit 7) of the registers ⊐00003 will be ignored.

F-231
MCRN

Master control reset nesting (Master Control Reset Nesting)

Combined with F-30 (MCS), it is used when the circuit after the common operation condition is branched into plural outputs. Nesting is not enabled in F-31 (MCR), but nesting is enabled in up to 8 levels by using F-231 (MCRN).

[Example]



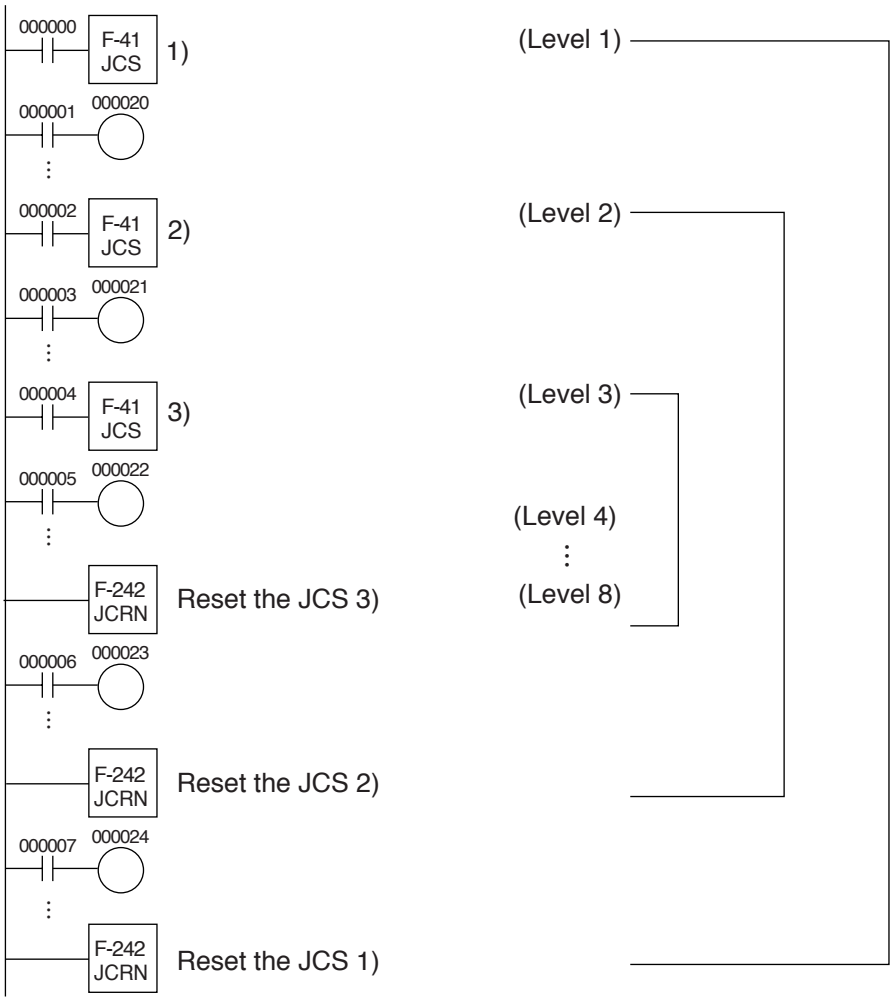
- Refer to the F-30 (MCS), F-31 (MCR).

F-242
JCRN

Jump control reset nesting (Jump Control Reset Nesting)

When the condition of F-41 (JCS) is OFF, any of the commands existing in up to F-242 (JCRN) except for END command is not executed. Nesting is not enabled in F-42 (JCR), but nesting is enabled in up to 8 levels by using F-242 (JCRN).

[Example]



- Refer to the F-41 (JCS), F-42 (JCR).

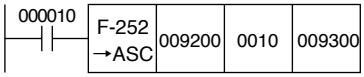
F-252 →ASC Converts HEX code into ASCII code (→ASCii)

Symbol		F-215d MUL | S1 | S2 | D
Function		In an area of n bytes starting from register S, converts hexadecimal codes to ASCII codes, and store the results in the area starting from register D. Converts from lower 4 bits of register S.
Operation		(S, S+1, ----, S+n-1) → ASCII → D, D+1, ----, D+2n-1
S1		Use range A
n		Use range 0000 to 1777(8) *1 (When 0000(8), 1024 bytes)
D		Use range A *1□
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S1 to S+n-1	Unchanged
	D to D+n-1	Result *2
	Flag	Unchanged

*1 Be careful for register area to specify by n and D. The number of bytes used in register of calculation results will be double of hexadecimal code area.

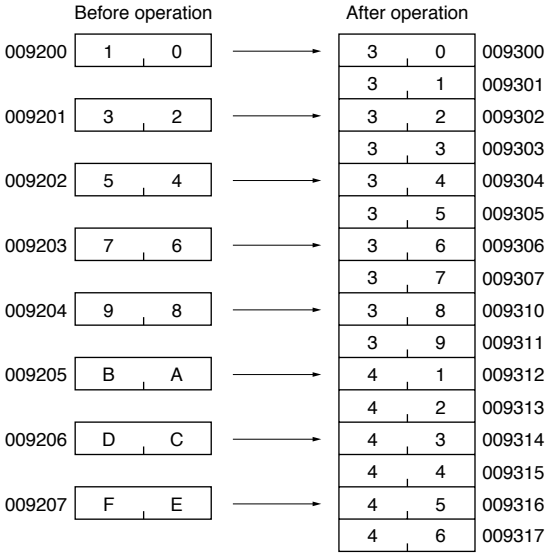
*2: Make sure that the calculation result does not enter the TMR/CNT contact area (file address 00001600 to 00001777(8) etc.) of file No. 0.

[Explanation]



Instruction	
S T R	000010
F-252	
	009200
	0010
	009300

When the input condition of 000010 changes from OFF to ON, this instruction converts the 8 (0010(8)) bytes of HEX data in registers 009200 to 009207 into their ASCII equivalents, and stores the results to the register area that begins with register 009300.



- Relationship between hexadecimal codes and ASCII codes.

Hex code	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ASCII code	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46

F-253 →HEX

Converts ASCII code into HEX code (→HEX)

Symbol	F-253 →HEX	S	n	D
Function	Converts the ASCII codes in the "n" byte register area that begins with register S into their hex equivalents, and stores the results to the register area that begins with register D. The lower 4 bits of converted data are stored first. If a hex code which is not convertible into ASCII code is found, the hex code is stored in the last address of the area that begins with register D, and conversion is aborted.			
Operation	(S, S+1,⋯, S+n-1) → HEX → D, D+1, ⋯, D+n/2-1 Unconvertible code → D+n/2-1 (If "n" is an odd number, the last address=D+(n-1)/2)			
S	Use range A			
n	Use range 0000 to 1777(8) (0000 specifies 1024 bytes)			
D	Use range A			
Condition	Rising edge of input signal (OFF to ON)			
Contents after operation	S to S+n-1	Unchanged		
	D to D+n/2-1	Result	If an unconvertible code is encountered, the code is stored in the last address of register D's area, and conversion is aborted.	
	D+n/2	Unchanged if normal.		

Flag	Conversion	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	Unconvertible code	0	0	1	0
	Other than above			0	

[Explanation]

000010	F-253 →HEX	009200	0017	009300

Instruction	
S T R	000010
F-253	
	009200
	0017
	009300

When the input condition of 00010 changes from OFF to ON, this instruction converts the 017(8) bytes (15) bytes of ASCII data in registers 009200 to 009216 into their HEX equivalents, and stores the results in the register area that begins with register 009300.

	Before operation			After operation		
009200	3	0	→	1	0	009300
009201	3	1				
009202	3	2	→	3	2	009301
009203	3	3				
009204	3	4	→	5	4	009302
009205	3	5				
009206	3	6	→	7	6	009303
009207	3	7				
009210	3	8	→	9	8	009304
009211	3	9				
009212	4	1	→	B	A	009305
009213	4	2				
009214	4	3	→	D	C	009306
009215	4	4				
009216	4	5	→	0	E	009307

- If an unconvertible ASCII code is found, the code is stored to the last register 009307.

* Make sure that the calculation result does not enter the TMR/CNT contact area (file address 00001600 to 00001777(8) etc.) of file No. 0.

- If "n" is an odd number, the upper 4 bits of the last data will become 0.
- Relationship between hexadecimal codes and ASCII codes.

Hex code	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ASCII code	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46

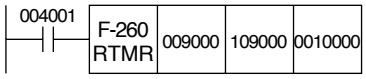
F-260 RTMR — Subtracts timer (setting value, register address)

Symbol		F-260 RTMR | S | D | BIT
Function		The contents of registers D and D+1 (the timer current value) are added by -1 at every 0.1 sec from the contents of registers S and S+1 (timer setting value), and when becoming 0, relay BIT is turned ON, and the input signal is held while it is ON.
Operation		(S, S+1) − Passage of time → (D, D+1) ↓ BIT (ON) for (D, D+1) = 0
S		Use range B
D		Use range B
BIT		Use range K
Condition		Starts counting on the rising edge of the input signal (OFF to ON), and continues as long as the input signal remains ON.
Contents before operation	S, S+1	Timer setting value 0000 to 9999 (BCD 4 digits, 0 to 999.9 seconds)
	D, D+1	Timer current value (Same as contents of S, S+1)
	BIT	OFF
Contents after operation	S, S+1	Unchanged
	D, D+1	Result the timer current value 0000 to 9999 (BCD 4 digits)
	BIT	ON (timer current value = 0)

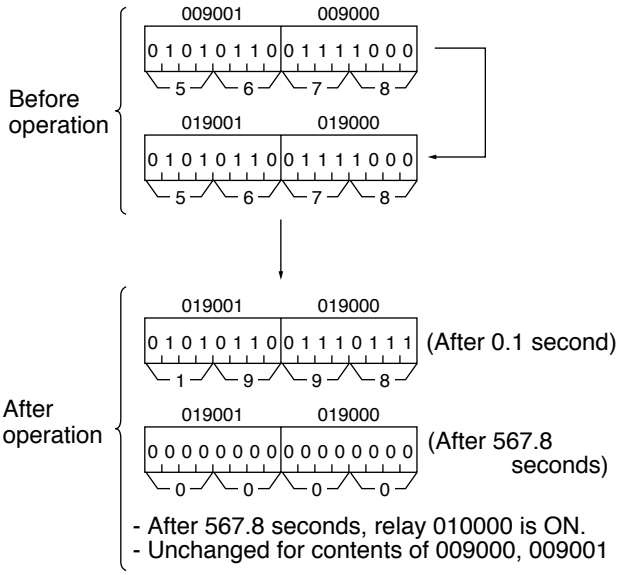
Flag	Contents of register D	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	BCD code	0	0	0	0
	Not BCD code			1	

[Explanation]

Instruction	
S T R	04001
F-260	
	009000
	019000
	010000



After input condition 004001 changes from OFF to ON, the contents in register 019000 and 019001 (current timer value) are incremented by 1, from the contents of register 009000 and 009001 (timer setting value), every 0.1 second as long as 004001 remains ON. When the current timer value reaches 0, relay 010000 turns ON. This relay will stay ON as long as 004001 remains ON.



- After 567.8 seconds, relay 010000 is ON.
- Unchanged for contents of 009000, 009001

The above operation shows timer setting value is 5678 (567.8 seconds).

- In case of using failure holding mode (refer to the system memory #0201), use latched relay area or register (after b00000 or after 009000) for register D.
- Function is same as decrement TMR instructions.

Input signal	Timer current value	Relay BIT
OFF	Timer setting value	OFF
ON (current value > 0)	Decrease−1 each 0.1 second	OFF
ON (current value = 0)	0	ON

Note: After program input, when the condition is ON and changed to the operation mode while the content of register D is 0, it must be noted that the output relay (BIT) is turned ON.

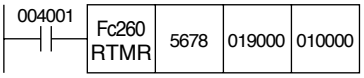
Fc260 RTMR — Subtracts timer (constant, register address)

Symbol		Fc260 RTMR | n | D | BIT
Function		The contents of registers D and D+1 (the timer current value) are added by−1 at every 0.1 sec. from "n" (timer setting value), and when becoming 0, relay BIT is turned ON, and the input signal is held while it is ON.
Operation		n−Passage of time → (D, D+1) ↓ BIT (ON) for (D, D+1) = 0
n		Timer setting value 0000 to 9999 (0 to 999,9 seconds)
D		Use range B
BIT		Use range K
Condition		Starts counting on the rising edge of the input signal (OFF to ON), and continues as long as the input signal remains ON.
Contents before operation	D, D+1	Timer current value Same as contents of n
	BIT	OFF
Contents after operation	D, D+1	Result the timer current value 0000 to 9999 (BCD 4 digits)
	BIT	ON (timer current value = 0)

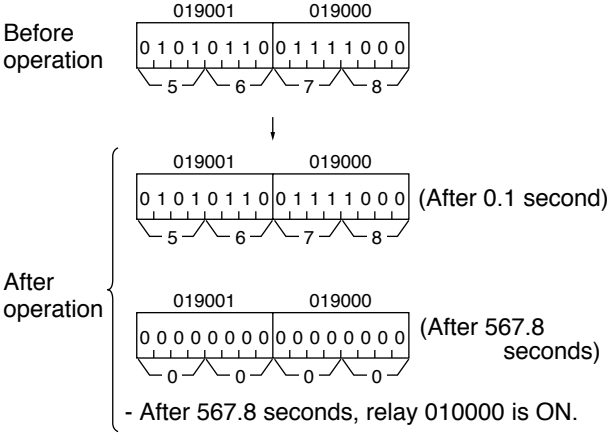
Flag	Contents of register D	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	BCD code	0	0	0	0
	Not BCD code	0	0	1	0

[Explanation]

Instruction	
S T R	004001
Fc260	
	5678
	019000
	010000



After input condition 004001 changes from OFF to ON, the contents in register 019000 and 019001 (current timer value) are incremented by 1, from the value of n 5678 (timer setting value567.8 seconds), every 0.1 second as long as 04001 remains ON. When the current timer value reaches 0, relay 010000 turns ON. This relay will stay ON as long as 004001 remains ON.



- After 567.8 seconds, relay 010000 is ON.

- In case of using failure holding mode (refer to the system memory #0201), use latched relay area or register (after b00000 or after 009000) for register D.
- Function is same as decrement TMR instructions.

Input signal	Timer current value	Relay BIT
OFF	Timer setting value	OFF
ON (current value > 0)	Decrease−1 each 0.1 second	OFF
ON (current value = 0)	0	ON

Note: After program input, when the condition is ON and changed to the operation mode while the content of register D is 0, it must be noted that the output relay (BIT) is turned ON.

F-261 RCNT Subtract counter (setting value, register address)

Symbol		1) 2) — F-261 RCNT | S | D | BIT
Function		During reset input 2) (OFF), contents of registers D and D+1 (counter current value) are added by −1 every time the count input 1) is changed from OFF to ON from the contents of registers S and S+1 (counter set value), and when becoming 0, the relay bit is turned ON and held. In the case of reset input 2) (ON), it follows that the counter current value = counter setting value and the relay BIT (OFF).
Operation		(S, S+1) -calculation input times → (D, D+1) ↓ BIT (ON) for (D, D+1) = 0
S		Use range B
D		Use range B
BIT		Use range K
Condition		Calculation input 1) at reset input 2) (OFF) (OFF → ON)
Contents before operation	S, S+1	Counter setting value 0000 to 9999 (BCD 4 digits)
	D, D+1	Counter current value (Same as contents of S, S+1)
	BIT	OFF
Contents after operation	S, S+1	Unchanged
	D, D+1	Result the counter current value 0000 to 9999 (BCD 4 digits)
	BIT	ON (counter current value = 0)

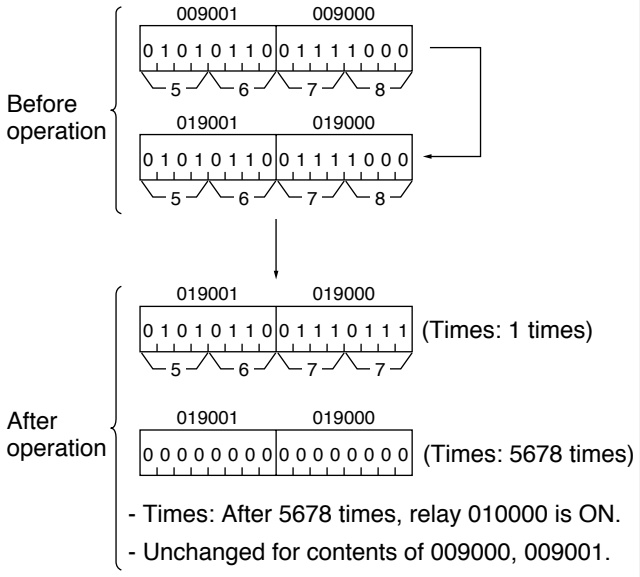
Flag	Contents of register D	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	BCD code	0	0	0	0
	Not BCD code	0	0	1	0

[Explanation]

Instruction	
S T R	004000
S T R	004001
F-261	009000 019000 010000

004000 / 004001 — F-261 RCNT | 009000 | 019000 | 010000

During reset input 004001 (OFF), contents of registers 019000 and 019001 (counter current value) are added by −1 every time the count input 004000 is changed from OFF to ON from the contents of registers 009000 and 009001 (counter setting value), and when becoming 0, the relay 010000 is turned ON and held. In the case of reset input 004001 (ON), it follows that (contents of registers 019000 and 019001) = (contents of registers 009000 and 009001) and the relay 010000 (OFF).
The below operation shows counter setting value is 5678 times.



- Times: After 5678 times, relay 010000 is ON.
- Unchanged for contents of 009000, 009001.

- Function is same as decrement CNT instructions.

Reset input 2)	Counter current value	Relay BIT
ON	Counter setting value	OFF
OFF (current value > 0)	Decrease −1 each calculation input 1) OFF to ON	OFF
ON (current value = 0)	0	ON

Note 1: After program input, when the reset input is OFF and changed to the operation mode while the content of register D, D+1 is 0, it must be noted that the output relay (BIT) is turned ON.

Note 2: Use the latched relay area or register (after 009000) for register D. In case of using the I/O relay area or ⊐xxxxx of auxiliary relay area, output relay (BIT) is ON at power OFF to ON.

Fc261 RCNT — Subtracts counter (constant, register address)

Symbol		1) Fc261 RCNT / 2) — n — D — BIT
Function		During reset input 2) (OFF), contents of registers D and D+1 (counter current value) are added by −1 every time the count input 1) is changed from OFF to ON from the contents of registers S and S+1 (counter set value), and when becoming 0, the relay bit is turned ON and held. In the case of reset input 2) (ON), it follows that the counter current value = counter setting value and the relay BIT (OFF).
Operation		n−calculation input times → (D, D+1) ↓ BIT (ON) for (D, D+1) = 0
n		Counter set value 0000 to 9999 (0 to 9999 times)
D		Use range B
BIT		Use range K
Condition		Calculation input 1) at reset input 2) (OFF) (OFF →ON)
Contents before operation	D, D+1	Counter current value (Same as contents of n)
	BIT	OFF
Contents after operation	D, D+1	Result the timer current value 0000 to 9999 (BCD 4 digits)
	BIT	ON (counter current value = 0)

Flag	Contents of register D	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	BCD code	0	0	0	0
	Not BCD code	0	0	1	0

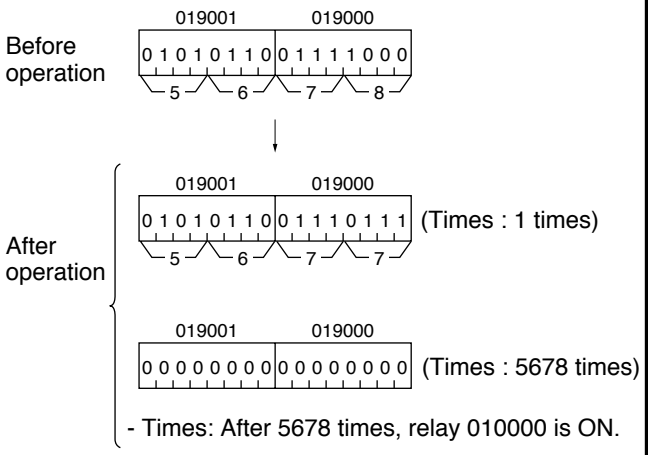
[Explanation]

Instruction	
S T R	004000
S T R	004001
Fc261	
	5678
	019000
	010000

004000 / 004001 — Fc261 RCNT | 5678 | 019000 | 010000

During reset input 004001 (OFF), contents of registers 019000 and 019001 (counter current value) are added by −1 every time the count input 004000 is changed from OFF to ON from value 5678 of "n" (counter setting value), and when becoming 0, the relay 010000 is turned ON and held. In the case of reset input 004001 (ON), it follows that (contents of registers 009000 and 009001) = (value 5678 of "n") and the relay 010000 (OFF).

The below operation shows counter setting value is 5678 times.



- Function is same as decrement CNT instructions.

Reset input 2)	Counter current value	Relay BIT
ON	Counter setting value	OFF
OFF (current value > 0)	Decrease−1 each calculation input 1) OFF to ON	OFF
OFF (current value = 0)	0	ON

Note 1: After program input, when the reset input is OFF and changed to the operation mode while the content of register D, D+1 is 0, it must be noted that the output relay (BIT) is turned ON.

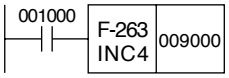
Note 2: Use the latched relay area or register (after 009000) for register D. In case of using the I/O relay area or ∃xxxxx of auxiliary relay area, output relay (BIT) is ON at power OFF to ON.

F-263 INC4 — Increments counter by (+4) (1-byte binary) (INCrement)

Symbol		F-263 INC4 | D
Function		Increment the binary contents of register D by (+4).
Operation		〈D〉+4 → D
D		Use range A
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	D	Result (binary code)
	Flag	Register (octal) / Zero 007357 / Carry 007356 / Error 007355 / Non-carry 007354
		374 → 000: 1 / 1 / 0 / 0
		375 → 001, 376 → 002, 377 → 003: 0 / 1 / 0 / 0
		Other than above: 0 / 0 / 0 / 1

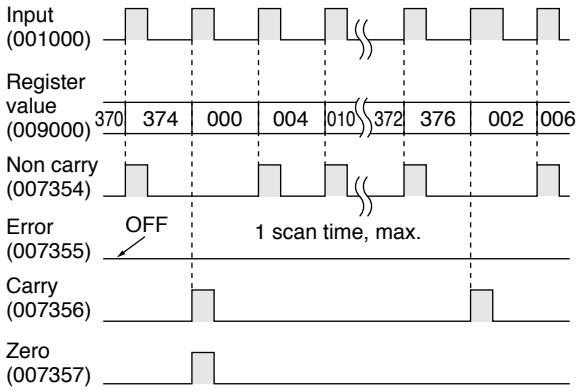
Resembled instructions: F-63, F-63w, F-63d, F-163, F-163w, F-163d, F-263d, F-263w

[Explanation]



Instruction	
S T R	001000
F-263	
	009000

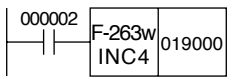
When the input condition of 001000 changes from OFF to ON, this instruction increments the binary contents of register 009000 by (+4).



F-263w INC4 — Increments counter by (+4) (1-word binary) (INCrement)

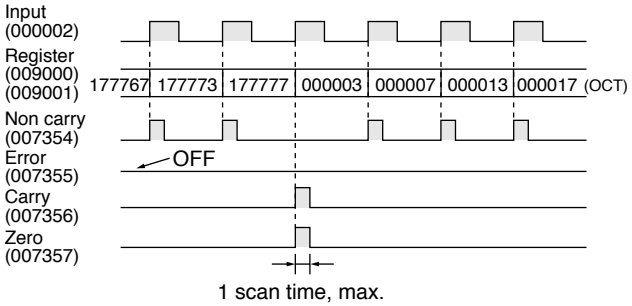
Symbol		F-263w INC4 | D
Function		Increment the binary contents of registers D and D+1 by (+4).
Operation		〈D, D+1〉+4 → D, D+1
D		Use range B * Be sure to use even addresses for registers D. (Odd addresses such as 019003 etc. are prohibited to use.)
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	D	Lower digits of result
	D+1	Upper digits of result
	Flag	Register (octal) / Zero 007357 / Carry 007356 / Error 007355 / Non-carry 007354
		177774 → 000000: 1 / 1 / 0 / 0
		177775 → 000001, 177776 → 000002, 177777 → 000003: 0 / 1 / 0 / 0
		Other than above: 0 / 0 / 0 / 1

[Explanation]



Instruction	
S T R	000002
F-263w	
	019000

When the input condition of 000002 changes from OFF to ON, this instruction increments the binary contents of registers 019000 and 019001 by (+4).

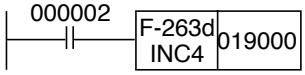


Resembled instructions: F-63, F-63w, F-63d, F-163, F-163w, F-163d, F-263, F-263d

F-263d INC4 Increments counter by (+4) (2-word binary) (INCrement)

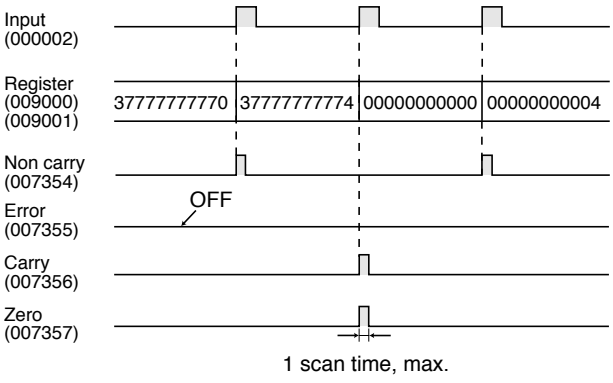
Symbol	F-263d INC4 D					
Function	Increment the binary contents of registers D and D+3 by (+4).					
Operation	〈D, D+3〉+4 → D, D+3					
D	Use range C * Be sure to use even addresses for registers D. (Odd addresses such as 019003 etc. are prohibited to use.)					
Condition	Rising edge of input signal (OFF to ON)					
Contents after operation	D to D+3	Result (D: lower to D+3: upper)				
	Flag	Register (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
		37777777774 → 00000000000	1	1	0	0
		37777777775 → 00000000001 37777777776 → 00000000002 37777777777 → 00000000003	0	1	0	0
		Other than above	0	0	0	1

[Explanation]



Instruction	
S T R	000002
F-263d	
	019000

When the input condition of 000002 changes from OFF to ON, this instruction increments the binary contents of registers 019000 to 019003 by (+4).

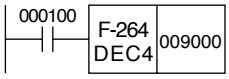


Resembled instructions: F-63, F-63w, F-63d, F-163, F-163w, F-163d, F-263, F-263w

F-264 DEC4 Decrements counter by (-4) (1-byte binary) (DECrement)

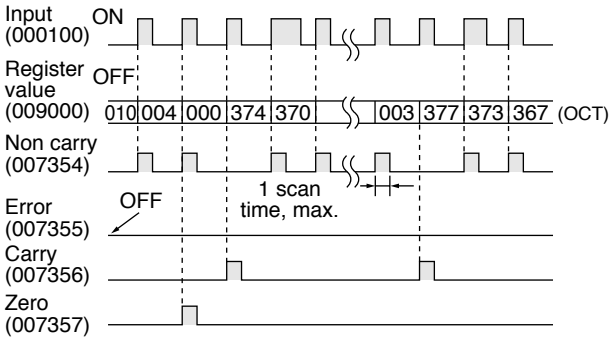
Symbol		F-264 DEC4 D
Function		Increment the binary contents of register D by (-4).
Operation		⟨D⟩-4 → D
D		Use range A
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	D	Result (binary code)
	Flag	Register (octal) / Zero 007357 / Carry 007356 / Error 007355 / Non-carry 007354
		004 → 000: 1 / 0 / 0 / 1
		003 → 377, 002 → 376, 001 → 375, 000 → 374: 0 / 1 / 0 / 0
		Other than above: 0 / 0 / 0 / 1

[Explanation]



Instruction	
S T R	000100
F-264	
	009000

When the input condition of 0001000 changes from OFF to ON, this instruction decrements the binary contents of register 009000 by (-4).

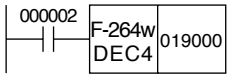


Resembled instructions: F-64, F-64w, F-64d, F-164, F-164w, F-164d, F-264w, F-264d

F-264w DEC4 Decrements counter by (-4) (1-word binary) (DECrement)

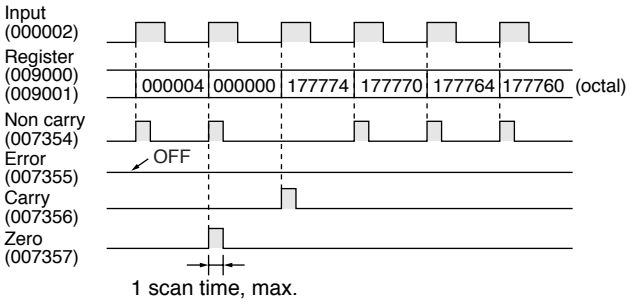
Symbol		F-264w DEC4 D
Function		Decrements the binary contents of registers D and D+1 by (-4).
Operation		⟨D, D+1⟩-4 → D, D+1
D		Use range B * Be sure to use even addresses for registers D. (Odd addresses such as 019003 etc. are prohibited to use.)
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	D	Lower digits of result
	D+1	Upper digits of result
	Flag	Register (octal) / Zero 007357 / Carry 007356 / Error 007355 / Non-carry 007354
		000004 → 000000: 1 / 0 / 0 / 1
		000003 → 177777, 000002 → 177776, 000001 → 177775, 000000 → 177774: 0 / 1 / 0 / 0
		Other than above: 0 / 0 / 0 / 1

[Explanation]



Instruction	
S T R	000002
F-264w	
	019000

When the input condition of 000002 changes from OFF to ON, this instruction decrements the binary contents of registers 019000 and 019001 by (-4).



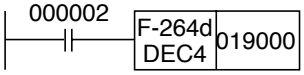
Resembled instructions: F-64, F-64w, F-64d, F-164, F-164w, F-164d, F-264, F-264d

F-264d DEC4 Decrements counter by (-4) (2-word binary) (DECrement)

Symbol		F-264d DEC4 D
Function		Decrements the binary contents of registers D to D+3 by (-4).
Operation		〈D to D+3〉-4 → D to D+3
D		Use range C * Be sure to use even addresses for registers D. (Odd addresses such as 019003 etc. are prohibited to use.)
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	D	Result (D: lower to D+3: upper)

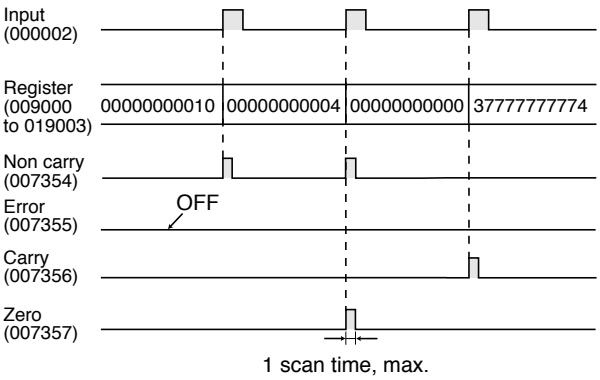
Contents after operation: Flag	Register (octal)	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	00000000004 → 00000000000	1	0	0	1
	00000000003 → 37777777777, 00000000003 → 37777777777, 00000000003 → 37777777777, 00000000003 → 37777777777	0	1	0	0
	Other than above	0	0	0	1

[Explanation]



Instruction	
S T R	000002
F-264d	
	019000

When the input condition of 000002 changes from OFF to ON, this instruction decrements the binary contents of registers 019000 to 019003 by (-4).

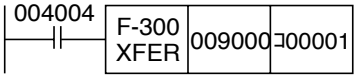


Resembled instructions: F-64, F-64w, F-64d, F-164, F-164w, F-164d, F-264, F-264w

F-300 XFER Transfer 1 byte data

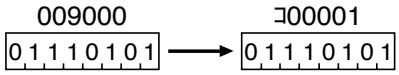
Symbol		F-300 XFER | S | D
Function		Transfer register S content (1 byte data) to register D.
Operation		S → D
S		Use range A *
D		Use range A *
Condition		When the input signal is ON
Contents after operation	S	Unchanged
	D	Contents of register S
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004004
F-300	
	009000
	00001

When the input condition 004004 is ON, the JW300 transfers register 009000 content to register 00001.



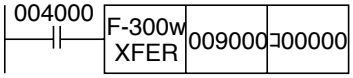
* This instruction cannot be used for the file register and indirect specification.

Resembled instructions: F-00, F-00w, F-00d, F-70, F-70w, F-70d, F-74, F-74w, F-74d, F-76, F-76w, F-76d

F-300w XFER Transfer 1 word data

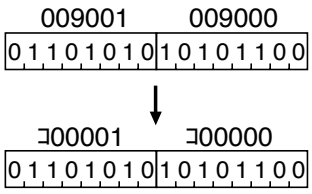
Symbol		F-300w XFER | S | D
Function		Transfer register S, S+1 content (1 word data) to register D, D+1.
Operation		S, S+1 → D, D+1
S		Use range B *
D		Use range B *
Condition		When the input signal is ON
Contents after operation	S, S+1	Unchanged
	D	Contents of register S
	D+1	Contents of register S+1
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004000
F-300w	
	009000
	00000

When the input condition 004000 is ON, the JW300 transfers register 009000,0090001 content (1-word data) to register 00000, 00001.



* This instruction cannot be used for the file register and indirect specification.

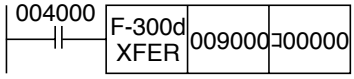
* Be sure to use even addresses for registers S and D.(Odd addresses such as 019003 are prohibited).

Resembled instructions: F-00, F-00w, F-00d, F-70, F-70w, F-70d, F-74, F-74w, F-74d, F-76, F-76w, F-76d

F-300d XFER Transfer 2 word data

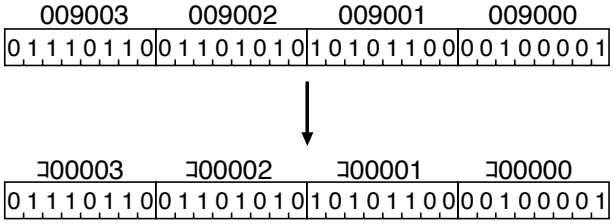
Symbol		F-300d XFER | S | D
Function		Transfer register S to S+3 content (2 word data) to register D to D+3.
Operation		S to S+3 → D to D+3
S		Use range C *
D		Use range C *
Condition		When the input signal is ON
Contents after operation	S to S+1	Unchanged
	D to D+3	Contents of register S to S+3
	Flag	Unchanged

[Explanation]



Instruction	
S T R	004000
F-300d	
	009000
	□00000

When the input condition 004000 is ON, the JW300 transfers register 009000 to 0090003 content (2-word data) to register □00000 to □00003.



* This instruction cannot be used for the file register and indirect specification.
* Make sure to set even addresses of 4 byte units for S and D (009000, 009004, 009010, etc.).

Resembled instructions: F-00, F-00w, F-00d, F-70, F-70w, F-70d, F-74, F-74w, F-74d, F-76, F-76w, F-76d

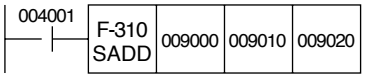
F-310 SADD — Add registers in binary with sign (31 bits + 31 bits) (Signed ADD)

Symbol		F-310 SADD | S_1 | S_2 | D
Function		Contents of the registers S_1 to S_1+3 are added as 31 bits with sign with the contents of the registers S_2 to S_2+3 and it's results are stored in the registers D to D+3.
Operation		(S_1 to S_1+3) + (S_2 to S_2+3) → (D to D+3)
S_1		Use range C *
S_2		Use range C *
D		Use range C *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1 to S_1+3	Unchanged
	S_2 to S_2+3	Unchanged
	D to D+3	Result (binary 31 bits with sign)

Flag	Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
	±0	1	0	0	1
	Overflow	0	1	1	0
	Underflow	0	0	1	1
	Other than above	0	0	0	1

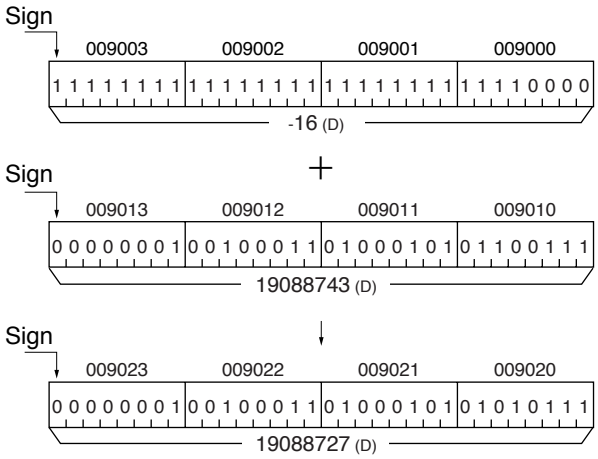
[Explanation]

Instruction	
S T R	004001
F-310	
	009000
	009010
	009020



When the input condition of 004001 changes from OFF to ON, this instruction adds the binary with sign contents of registers 009000 to 009003 to contents of registers 009010 to 009013, and stores the result in registers 009020 to 009023.

The below example shows -16＋19088743 ＝ 19088727.



* Be sure to use even addresses for registers S_1, S_2 and D.

- Range of treated numerical value is -2147483648 to 2147483647(D). => (See page binary with sign.)

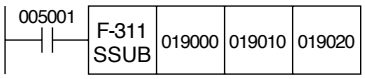
F-311 SSUB — Subtracts registers in binary with sign (31 bits − 31 bits) (Signed SUB tract)

Symbol		F-311 SSUB | S_1 | S_2 | D
Function		Contents of the registers S_1 to S_1+3 are subtracted as 31 bits with sign with the contents of the registers S_2 to S_2+3 and it's results are stored in the registers D to D+3.
Operation		(S_1 to S_1+3) + (S_2 to S_2+3) → (D to D+3)
S_1		Use range C *
S_2		Use range C *
D		Use range C *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1 to S_1+3	Unchanged
	S_2 to S_2+3	Unchanged
	D to D+3	Result (binary 31 bits with sign)

Flag: Result	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
±0	1	0	0	1
Overflow	0	1	1	0
Underflow	0	0	1	1
Other than above	0	0	0	1

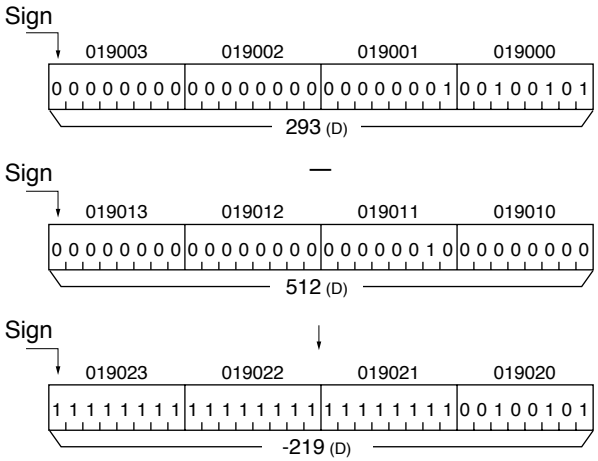
[Explanation]

Instruction	
S T R	005001
F-311	019000
	019010
	019020



When the input condition of 005001 changes from OFF to ON, this instruction subtracts the binary with sign contents of registers 019000 to 019003 to contents of registers 019010 to 019013, and stores the result in registers 019020 to 019023.

The below example shows 293-512=-219.



* Be sure to use even addresses for registers S_1, S_2 and D

- Range of treated numerical value is -2147483648 to 2147483647(D). => (See page binary with sign.)

F-315 SMUL — Multiplies registers in binary with sign (31 bits × 31 bits) (Signed MUL tiply)

Symbol		F-315 SMUL | S_1 | S_2 | D
Function		Contents of the registers S_1 to S_1+3 are multiplied as 31 bits with sign with the contents of the registers S_2 to S_2+3 and it's results are stored in the registers D to D+7.
Operation		(S_1 to S_1+3) × (S_2 to S_2+3) → (D to D+7)
S_1		Use range C *
S_2		Use range C *
D		Use range G *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1 to S_1+3	Unchanged
	S_2 to S_2+3	Unchanged
	D to D+7	Result (binary 63 bits with sign)
	Flag	Zero 007357 | Carry 007356 | Error 007355 | Non-carry 007354
		0 | 0 | 0 | 0

* Be sure to use even addresses for registers S_1, S_2 and D.

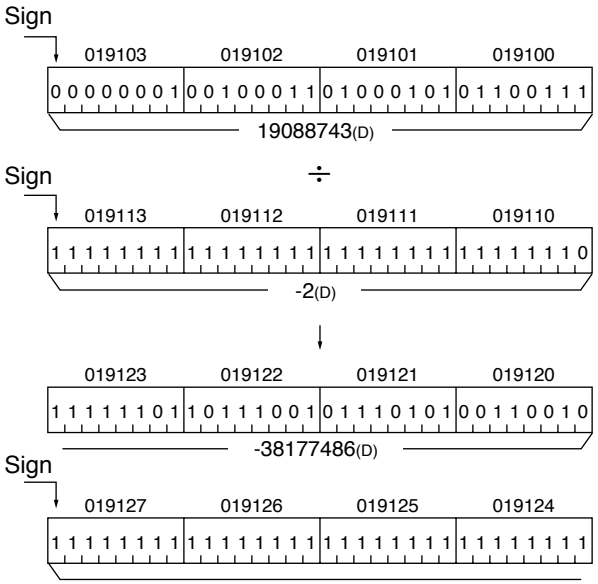
- Range of treated numerical value is -2147483648 to 2147483647(D). ⇒ (See page binary with sign.)

[Explanation]

Instruction	
S T R	004201
F-315	009100
	009110
	009120

004201 — F-315 SMUL | 009100 | 009110 | 009120

When the input condition of 004201 changes from OFF to ON, this instruction multiples the binary with sign contents of registers 009110 to 009113 and stores the result in registers 009120 to 009127. The above example shows 19088743×(-2)＝-38177486.



F-316 SDIV — Divides registers in binary with sign (31 bits ÷ 31 bits) (Signed DIV ide)

Symbol		F-316 SDIV | S_1 | S_2 | D
Function		Contents of the registers S_1 to S_1+3 are divided as 31 bits with sign with the contents of the registers S_2 to S_2+3 and it's quotient is stored in the registers D to D+3 and the remainder in the registers D+4 to D+7.
Operation		(S_1 to S_1+3) ÷ (S_2 to S_2+3) → (D to D+7)
S_1		Use range C *
S_2		Use range C *
D		Use range G *
Condition		Rising edge of input signal (OFF to ON)
Contents after operation	S_1 to S_1+3	Unchanged
	S_2 to S_2+3	Unchanged
	D to D+3	Result (binary 31 bits with sign)
	D+4 to D+7	Result (binary 31 bits with sign)

Flag

Contents of register S_2 to S_2+3	Zero 007357	Carry 007356	Error 007355	Non-carry 007354
0	0	0	1	0
Other than above	0	0	0	0

* Be sure to use even addresses for registers S_1, S_2 and D.
- Range of treated numerical value is -2147483648 to 2147483647(D). ⇒ (See page binary with sign.)

[Explanation]

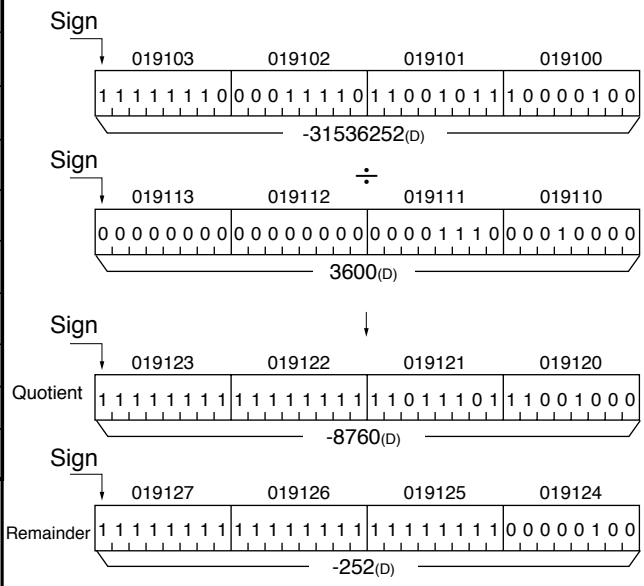
Instruction	
S T R	005201
F-316	019100
	019110
	019120

005201 | F-316 SDIV | 019100 | 019110 | 019120

When the input condition of 005201 changes from OFF to ON, this instruction divides the binary with sign contents of registers 019100 to 019103 to contents of registers 019110 to 019113, and stores the result in registers 019120 to 019127.

- Example 1

The above shows -31536252 ÷ 3600 = -8760's quotient and remainder -252.



- Example 2

31536252 ÷ (-3600) = -8760 remainder 252

- Example 3

-31536252 ÷ (-3600) = 8760 remainder -252

F-403 LOG — Logging instruction

F-403 instruction shall be used for the logging function.
=> See "Logging function".

SHARP

SHARP MANUFACTURING SYSTEMS CORPORATION

♦Information about Sharp programmable controller is available at our internet homepage **http://sharp-world.com/sms/**

1.0v 12.2003
Printed in Japan (0.1I.Y.Y)