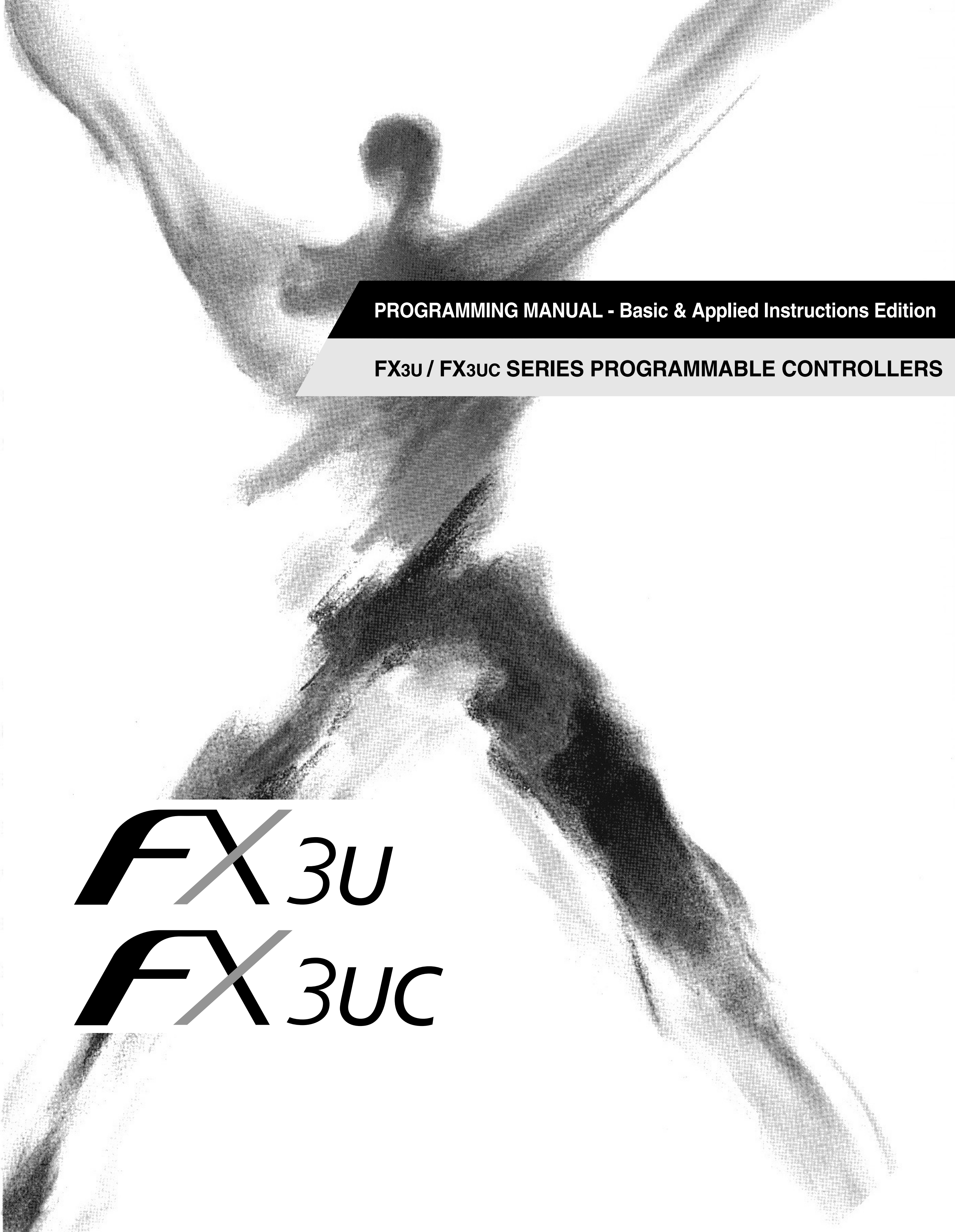
Changes for the Better

MITSUBISHI

PROGRAMMABLE CONTROLLERS

MELSEC-F



PROGRAMMING MANUAL - Basic & Applied Instructions Edition

FX3U / FX3UC SERIES PROGRAMMABLE CONTROLLERS

FX3U

FX3UC

FX3U/FX3UC Series Programmable Controllers

Programming Manual - Basic & Applied Instruction Edition

Manual No.	JY997D16601
Revision	B
Date	2006 / 2

Foreword

This manual contains text, diagrams and explanations which will guide the reader through the safe and correct installation, use, and operation of the FX3U/FX3UC Series programmable controller. It should be read and understood before attempting to install or use the unit.

Store this manual in a safe place so that you can take it out and read it whenever necessary. Always forward it to the end user.

Outline Precautions

- This manual provides information for the use of the FX3U Series Programmable Controllers. The manual has been written to be used by trained and competent personnel. The definition of such a person or persons is as follows;
  a) Any engineer who is responsible for the planning, design and construction of automatic equipment using the product associated with this manual should be of a competent nature, trained and qualified to the local and national standards required to fulfill that role. These engineers should be fully aware of all aspects of safety with regards to automated equipment.
  b) Any commissioning or service engineer must be of a competent nature, trained and qualified to the local and national standards required to fulfill that job. These engineers should also be trained in the use and maintenance of the completed product. This includes being completely familiar with all associated documentation for the said product. All maintenance should be carried out in accordance with established safety practices.
  c) All operators of the completed equipment should be trained to use that product in a safe and coordinated manner in compliance to established safety practices. The operators should also be familiar with documentation which is connected with the actual operation of the completed equipment.

  Note: the term 'completed equipment' refers to a third party constructed device which contains or uses the product associated with this manual
- This product has been manufactured as a general-purpose part for general industries, and has not been designed or manufactured to be incorporated in a device or system used in purposes related to human life.
- Before using the product for special purposes such as nuclear power, electric power, aerospace, medicine or passenger movement vehicles, consult with Mitsubishi Electric.
- This product has been manufactured under strict quality control. However when installing the product where major accidents or losses could occur if the product fails, install appropriate backup or failsafe functions in the system.
- When combining this product with other products, please confirm the standard and the code, or regulations with which the user should follow. Moreover, please confirm the compatibility of this product to the system, machine, and apparatus with which a user is using.
- If in doubt at any stage during the installation of the product, always consult a professional electrical engineer who is qualified and trained to the local and national standards. If in doubt about the operation or use, please consult the nearest Mitsubishi Electric distributor.
- Since the examples indicated by this manual, technical bulletin, catalog, etc. are used as a reference, please use it after confirming the function and safety of the equipment and system. Mitsubishi Electric will accept no responsibility for actual use of the product based on these illustrative examples.
- This manual content, specification etc. may be changed without a notice for improvement.
- The information in this manual has been carefully checked and is believed to be accurate; however, you have noticed a doubtful point, a doubtful error, etc., please contact the nearest Mitsubishi Electric distributor.

Registration

- Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- The company name and the product name to be described in this manual are the registered trademarks or trademarks of each company.

Table of Contents

5. How to Specify Devices and Constants to Instructions 126

6. What to Understand before Programming 137

7. Basic Instruction 154

8. Program Flow – FNC 00 to FNC 09 194

9. Move and Compare – FNC 10 to FNC 19 220

10. Arithmetic and Logical Operation (+, −, ×, ÷) – FNC 20 to FNC 29 248

11. Rotation and Shift Operation – FNC 30 to FNC 39 272

12. Data Operation – FNC 40 to FNC 49 296

13. High Speed Processing – FNC 50 to FNC 59 319

14. Handy Instruction – FNC 60 to FNC 69 367

24. Others – FNC181 to FNC189 571

25. Block Data Operation – FNC190 to FNC199 589

26. Character String Control – FNC200 to FNC209 601

27. Data Operation 3 – FNC210 to FNC219 635

28. Data Comparison – FNC220 to FNC249 647

29. Data Table Operation – FNC250 to FNC269 658

35. Interrupt Function and Pulse Catch Function 787

36. Operation of Special Devices (M8000 -, D8000 -) 806

37. Error Check Method and Error Code List 850

Related manuals

For detailed explanation of programming (basic instructions, applied instructions and step ladder instructions) in FX3U/FX3UC PLCs, refer to this manual.
For hardware information on the PLC main unit, special extension units, etc., refer to each associated manual.
For acquiring manuals, contact the distributor you have purchased the product from.

⦿Essential manual
○Manual required depending on application
△Manual with additional manual for detailed explanation

		Manual name	Manual number	Contents	Model name code
Manuals for PLC main unit					
■FX3U PLC main unit					
△	Supplied with product	FX3U Series Hardware Manual	JY997D18801	I/O specifications, wiring and installation of the PLC main unit FX3U extracted from the FX3U Series User's Manual - Hardware Edition. For detailed explanation, refer to the FX3U Series User's Manual - Hardware Edition.	–
⦿	Additional Manual	FX3U Series User's Manual - Hardware Edition	JY997D16501	Details about the hardware including I/O specifications, wiring, installation and maintenance of the FX3U PLC main unit.	09R516
■Programming					
⦿	Additional Manual	FX3U/FX3UC Series Programming Manual - Basic & Applied Instruction Edition (this manual)	JY997D16601	Items related to programming in PLCs including explanation of basic instructions, applied instructions and various devices in FX3U/FX3UC PLCs.	09R517
■Terminal block					
○	Supplied with product	FX INPUT AND OUTPUT TERMINAL BLOCKS	JY992D50401	Terminal block handling procedures.	–
Manuals for communication control					
■Common					
○	Additional Manual	FX Series User's Manual - Data Communication Edition	JY997D16901	Details about simple N : N link, parallel link, computer link and no-protocol communication (RS instruction and FX2N-232IF).	09R715
■Communication via RS-232C/RS-422/RS-485/USB **When using each product, refer also to the User's Manual - Hardware Edition of the PLC main unit to which each product is connected.**					
△	Supplied with product	FX3U-USB-BD User's Manual	JY997D13501	Items about the system configuration of USB communication expansion board and the driver installation method. For use, refer also to the FX Series User's Manual - Data Communication Edition.	–
△	Supplied with product	FX3U-232-BD Installation Manual	JY997D12901	Handling procedures of the RS-232C communication expansion board. For use, refer also to the FX Series User's Manual - Data Communication Edition.	–
△	Supplied with product	FX3U-232ADP Installation Manual	JY997D13701	Handling procedures of the RS-232C communication special adapter. For use, refer also to the FX Series User's Manual - Data Communication Edition.	–
△	Supplied with product	FX2N-232IF Hardware Manual	JY992D73501	Handling procedures of the RS-232C communication special function block. For use, refer also to the FX Series User's Manual - Data Communication Edition.	–

⊙Essential manual
○Manual required depending on application
△Manual with additional manual for detailed explanation

		Manual name	Manual number	Contents	Model name code
■Communication via RS-232C/RS-422/RS-485/USB When using each product, refer also to the User's Manual - Hardware Edition of the PLC main unit to which each product is connected.					
△	Supplied with product	FX3U-422-BD Installation Manual	JY997D13101	Handling procedures of the RS-422 communication expansion board. For use, refer also to the FX Series User's Manual - Data Communication Edition.	–
△	Supplied with product	FX3U-485-BD Installation Manual	JY997D13001	Handling procedures of the RS-485 communication expansion board. For use, refer also to the FX Series User's Manual - Data Communication Edition.	–
△	Supplied with product	FX3U-485ADP Installation Manual	JY997D13801	Handling procedures of the RS-485 communication special adapter. For use, refer also to the FX Series User's Manual - Data Communication Edition.	–
○	Supplied with product	FX-485PC-IF Hardware Manual	JY992D81801	Handling procedures of the RS-232C/RS-485 conversion interface. For use, refer also to the FX Series User's Manual - Data Communication Edition.	–
■CC-Link, MELSEC I/O LINK and AS-i system When using each product, refer also to the User's Manual - Hardware Edition of the PLC main unit to which each product is connected.					
△	Supplied with product	FX2N-16CCL-M Hardware Manual	JY992D93201	Handling procedures of the CC-Link master special function block. For use, refer to the FX2N-16CCL-M User's Manual.	–
○	Additional Manual	FX2N-16CCL-M User's Manual	JY992D93101	Details about the CC-Link master special function block.	09R710
○	Supplied with product	FX2N-32CCL User's Manual	JY992D71801	Handling procedures of the CC-Link remote device station special function block.	09R711
○	Supplied with product	Remote I/O Stations and remote device stations for CC-Link	For remote I/O stations and remote device stations for CC-Link, refer to each manual and the related data.		
△	Supplied with product	FX2N-64CL-M User's Manual [Hardware Volume]	JY997D05401	Handling procedures of the CC-Link/LT master special function block. For use, refer also to the FX2N-64CL-M User's Manual [Detailed Volume].	–
○	Additional Manual	FX2N-64CL-M User's Manual [Detailed Volume]	JY997D08501	Details about the CC-Link/LT master special function block.	–
○	Supplied with product	Remote I/O Power Adapter	For remote I/O stations, power adapter and dedicated power supply for CC-Link/LT, refer to each manual and the related data.		
○	Supplied with product	FX2N-16LNK-M User's Manual	JY992D72101	Handling procedure of the master special function block for the MELSEC I/O LINK	09R703
○	Supplied with product	FX2N-32ASI-M User's Manual	JY992D76901	Handling procedure of the master special function block for the AS-i system.	–

⦿Essential manual
○Manual required depending on application
△Manual with additional manual for detailed explanation

		Manual name	Manual number	Contents	Model name code
Manuals for analog control					
■Common					
○	Additional Manual	FX3U/FX3UC User's Manual - Analog Control Edition	JY997D16701	Detaileds about the analog special function block (FX3UC-4AD) and analog special adapter (FX3U-∗∗∗∗-ADP).	09R619
■Analog input, temperature input and temperature control **When using each product, refer also to the User's Manual - Hardware Edition of the PLC main unit to which each product is connected.**					
○	Supplied with product	FX2N-2AD User's Guide	JY992D74701	Handling procedures of the 2-channel analog input special function block.	–
○	Supplied with product	FX3U-4AD-ADP User's Manual	JY997D13901	Handling procedures of the 4-channel analog input special adapter. For use, refer also to the FX3U/FX3UC Series User's Manual - Analog Control Edition.	–
○	Supplied with product	FX2N-4AD User's Guide	JY992D65201	Handling procedures of the 4-channel analog input special function block.	–
○	Supplied with product	FX2NC-4AD User's Manual	JY997D07801	Handling procedures of the 4-channel analog input special function block.	–
○	Supplied with product	FX2N-8AD User's Manual	JY992D86001	Handling procedures of the 8-channel analog input (and thermocouple input) special function block.	09R608
○	Supplied with product	FX3U-4AD-PT-ADP User's Manual	JY997D14701	Handling procedures of the 4-channel PT-100 temperature sensor input special adapter. For use, refer also to the FX3U/FX3UC Series User's Manual - Analog Control Edition.	–
○	Supplied with product	FX2N-4AD-PT User's Guide	JY992D65601	Handling procedures of the 4-channel PT-100 temperature sensor input special function block.	–
○	Supplied with product	FX3U-4AD-TC-ADP User's Manual	JY997D14801	Handling procedures of the 4-channel thermocouple input special adapter. For use, refer also to the FX3U/FX3UC Series User's Manual - Analog Control Edition.	–
○	Supplied with product	FX2N-4AD-TC User's Guide	JY992D65501	Handling procedures of the 4-channel thermocouple input special function block.	–
△	Supplied with product	FX2N-2LC User's Guide	JY992D85601	Handling procedures of the 2-channel temperature control special function block. For use, refer to the FX2N-2LC User's Manual	–
○	Additional Manual	FX2N-2LC User's Manual	JY992D85801	Details about the 2-channel temperature control special function block.	09R608

⦿Essential manual
○Manual required depending on application
△Manual with additional manual for detailed explanation

		Manual name	Manual number	Contents	Model name code
Manuals for analog control					
■Analog output **When using each product, refer also to the User's Manual - Hardware Edition of the PLC main unit to which each product is connected.**					
○	Supplied with product	FX2N-2DA User's Guide	JY992D74901	Handling procedures of the 2-channel analog output special function block.	–
○	Supplied with product	FX3U-4DA-ADP User's Manual	JY997D14001	Handling procedures of the 4-channel analog output special adapter. For use, refer also to the FX3U/FX3UC Series User's Manual - Analog Control Edition.	–
○	Supplied with product	FX2N-4DA User's Guide	JY992D65901	Handling procedures of the 4-channel analog output special function block.	–
○	Supplied with product	FX2NC-4DA User's Manual	JY997D07601	Handling procedures of the 4-channel analog output special function block.	–
■Analog I/O (mixed) **When using each product, refer also to the User's Manual - Hardware Edition of the PLC main unit to which each product is connected.**					
○	Supplied with product	FX0N-3A User's Guide	JY992D49001	Handling procedures of the 2-channel analog input/1-channel analog output special function block.	–
○	Supplied with product	FX2N-5A Users Manual	JY997D11401	Handling procedures of the 4-channel analog input/1-channel analog output special function block.	09R616
Manuals for positioning control					
■High speed counter **When using each product, refer also to the User's Manual - Hardware Edition of the PLC main unit to which each product is connected.**					
△	Supplied with product	FX3U-4HSX-ADP Installation Manual	JY997D16301	Handling procedure of the special high speed input adapter.	–
○	Supplied with product	FX2N-1HC User's Guide	JY992D64501	Handling procedures of the 1-channel high speed counter special function block.	–
Manuals for positioning control					
■Common					
○	Additional Manual	FX3U/FX3UC Series User's Manual - Positioning Edition	JY997D16801	Details about the positioning function built in the FX3U/FX3UC Series.	09R620
■Pulse output and positioning **When using each product, refer also to the User's Manual - Hardware Edition of the PLC main unit to which each product is connected.**					
△	Supplied with product	FX3U-2HSY-ADP Installation Manual	JY997D16401	Handling procedure of the special high speed output adapter. For use, refer also to the FX3U/FX3UC Series User's Manual - Positioning Edition.	–
○	Supplied with product	FX2N/FX-1PG-E User's Manual	JY992D65301	Handling procedures of the 1-axis pulse output special function block.	09R610
△	Supplied with product	FX2N-10PG Installation Manual	JY992D91901	Handling procedures of the 1-axis pulse output special function block. For use, refer to the FX2N-10PG User's Manual.	–
○	Additional Manual	FX2N-10PG User's Manual	JY992D93401	Details about the 1-axis pulse output special function block.	09R611

⊙Essential manual
○Manual required depending on application
△Manual with additional manual for detailed explanation

		Manual name	Manual number	Contents	Model name code
△	Supplied with product	FX2N-10GM User's Guide	JY992D77701	Handling procedures of the 1-axis positioning special extension unit. For use, refer to the FX2N-10GM/FX2N-20GM Hardware/Programming Manaul	–
△	Supplied with product	FX2N-20GM User's Guide	JY992D77601	Handling procedures of the 2-axis positioning special extension unit. For use, refer to the FX2N-10GM/FX2N-20GM Hardware/Programming Manual	–
○	Additional Manual	FX2N-10GM, FX2N-20GM Hardware/Programming Manual	JY992D77801	Details on the 1-axis/2-axis positioning special extension unit.	09R612
■Programmable cam switch **When using each product, refer also to the User's Manual - Hardware Edition of the PLC main unit to which each product is connected.**					
○	Supplied with product	FX2N-1RM-E-SET User's Manual	JY992D71101	Handling procedures of the programmable cam switch special extension unit.	09R614
Other manuals					
When using each product, refer also to the User's Manual - Hardware Edition of the PLC main unit to which each product is connected.					
△	Supplied with product	FX3U-CNV-BD Installation Manual	JY997D13601	Handling procedures of the connector conversion expansion board for special adapter connection.	–

Generic Names and Abbreviations Used in Manuals

Abbreviation/generic name	Name
Programmable controllers	
FX3U Series	Generic name of FX3U Series PLCs
FX3U PLC or main unit	Generic name of FX3U Series PLC main units
FX3UC Series	Generic name of FX3UC Series PLCs
FX3UC PLC or main unit	Generic name of FX3UC Series PLC main units Only manuals in Japanese are available for these products.
FX2N Series	Generic name of FX2N Series PLCs
FX2NC Series	Generic name of FX2NC Series PLCs
FX1N Series	Generic name of FX1N Series PLCs
FX1NC Series	Generic name of FX1NC Series PLC main units Only manuals in Japanese are available for these products.
FX1S Series	Generic name of FX1S Series PLCs
Expansion boards	
Expansion board	Generic name of expansion boards (The models shown below): FX3U-232-BD, FX3U-422-BD, FX3U-485-BD, FX3U-USB-BD, and FX3U-CNV-BD
Special adapters	
Special adapter	Generic name of special high speed I/O adapters, special communication adapters, and special analog adapters Connectable equipment may vary depending on the main unit. For connectable equipment, refer to the User's Manual - Hardware Edition of the main unit.
Special high speed I/O adapter	Generic name of special high speed I/O adapters (The models shown below): FX3U-2HSY-ADP and FX3U-4HSX-ADP
Special communication adapter	Generic name of special communication adapters (The models shown below): FX3U-232ADP and FX3U-485ADP
Special analog adapter	Generic name of special analog adapters (The models shown below): FX3U-4AD-ADP, FX3U-4DA-ADP, FX3U-4AD-PT-ADP, and FX3U-4AD-TC-ADP
Extension equipment	
Extension equipment	Generic name of I/O extension equipment and special extension equipment Connectable equipment may vary depending on the main unit. For connectable equipment, refer to the User's Manual - Hardware Edition of the main unit.
I/O extension equipment	Generic name of FX2N Series I/O extension units, FX2N Series I/O extension blocks, FX2NC Series I/O extension blocks, and FX0N Series I/O extension blocks Connectable equipment may vary depending on the main unit. For connectable equipment, refer to the User's Manual - Hardware Edition of the main unit.
Special function block/unit or special extension equipment	Generic name of special extension units and special function blocks Connectable equipment may vary depending on the main unit. For connectable equipment, refer to the User's Manual - Hardware Edition of the main unit.
Special extension unit	Generic name of special extension units
Special function block	Generic name of special function blocks Connectable equipment may vary depending on the main unit. For connectable equipment, refer to the User's Manual - Hardware Edition of the main unit.
FX3UC Series special function block	FX3UC-4AD Only manuals in Japanese are packed together with this product.

Abbreviation/generic name		Name
Open field networks CC-Link and CC-Link/LT		
CC-Link equipment		Generic name of CC-Link master station and CC-Link remote device stations
	CC-Link master (station)	Generic name of CC-Link master station (having following model name): FX2N-16CCL-M
	CC-Link remote station	Generic name of remote I/O stations and remote device stations
CC-Link/LT equipment		Generic name of CC-Link/LT master station, CC-Link/LT remote I/O stations, power supply adapters, and dedicated power supplies
	CC-Link/LT master	Generic name of built-in type CC-Link/LT master and (additional) CC-Link/LT master
	Built-in type CC-Link/LT master	Generic name of built-in type CC-Link/LT master built in FX3UC-32MT-LT
	(Additional) CC-Link/LT master	Generic name of CC-Link/LT master station (having following model name): FX2N-64CL-M
	Power supply adapter	Generic name of units connected to supply the power to the CC-Link/LT system
	Dedicated power supply	Generic name of power supplies connected to supply the power to the CC-Link/LT system
AS-i system		
AS-i master		Generic name of AS-i system master station (having following model name): FX2N-32ASI-M
MELSEC I/O LINK		
MELSEC I/O LINK master		Generic name of MELSEC I/O LINK master station (having following model name): FX2N-16LNK-M
Options		
Extension power supply unit		FX3UC-1PS-5V
Memory cassette		FX3U-FLROM-16, FX3U-FLROM-64, and FX3U-FLROM-64L
Battery		FX3U-32BL
Peripheral equipment		
Peripheral equipment		Generic name of programming software, handy programming panels, and display units
Programming tools		
Programming tool		Generic name of programming software and handy programming panels
Programming software		Generic name of programming software
	GX Developer	Generic name of programming software packages SW□-D5C-GPPW-J and SW□-D5C-GPPW-E
	FX-PCS/WIN(-E)	Generic name of programming software packages FX-PCS/WIN and FX-PCS/WIN-E
Handy programming panel (HPP)		Generic name of programming panels FX-20P(-E) and FX-10P(-E)
RS-232C/RS-422 converter		FX-232AW, FX-232AWC, and FX-232AWC-H
RS-232C/RS-485 converter		FX-485PC-IF-SET and FX-485PC-IF
Display units		
GOT1000 Series		Generic name of GT15 and GT11
GOT-900 Series		Generic name of GOT-A900 and GOT-F900 Series
GOT-A900 Series		Generic name of GOT-A900 Series
GOT-F900 Series		Generic name of GOT-F900 Series
ET-940 Series		Generic name of ET-940 Series Only manuals in Japanese are available for these products.
Manuals		
FX3U Hardware Edition		FX3U Series User's Manual - Hardware Edition
FX3UC Hardware Edition		FX3UC Series User's Manual - Hardware Edition (Japanese only)
Programming Manual		FX3U/FX3UC Series Programming Manual - Basic & Applied Instruction Edition
Communication Control Edition		FX Series User's Manual - Data Communication Edition
Analog Control Edition		FX3U/FX3UC Series User's Manual - Analog Control Edition
Positioning Control Edition		FX3U/FX3UC Series User's Manual - Positioning Edition

1. Introduction

This chapter explains basic items related to programming in FX3U and FX3UC programmable controllers (PLCs).

1.1 Programming Language in PLCs

This section explains the features of programming in FX3U and FX3UC PLCs.

1.1.1 Types of programming languages

FX3U and FX3UC PLCs support the following three types of programming languages:

1. List programming

This method is the basis of programs.

1) Features
In this method, sequence instructions are input in the form of instruction words such as "LD", "AND" and "OUT".
This input method is the basis of sequence programs.

2) Example of list display

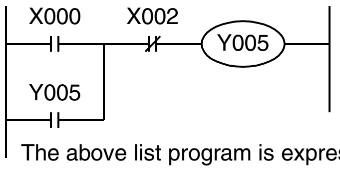
Step	Instruction	Device number
0000	LD	X000
0001	OR	Y005
0002	ANI	X002
0003	OUT	Y005
⋮	⋮	⋮

2. Circuit programming

In this method, ladder formats are drawn on the graphic screen.

1) Features
In a circuit program, a sequence circuit is drawn on the graphic screen by sequence formats and device numbers. Because a sequence circuit is expressed with contact symbols and coil symbols, the contents of a program can be understood easily.
In the circuit display status, the PLC operations can be monitored.

2) Example of circuit display



The above list program is expressed in the circuit diagram.

3. SFC (STL <step ladder>) programming

This input method allows sequence design in accordance with the flow of machine operations.

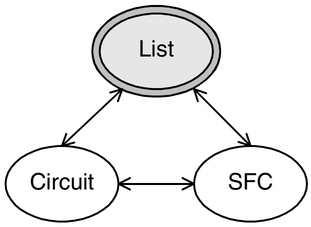
1) Features
In an SFC (sequential function chart) program, sequences can be designed in accordance with the flow of machine operations.

2) Compatibility between SFC programs and other programs
SFC programs can be converted into another program format. And when list programs and circuit programs are created according to certain rules, they can be converted inversely into SFC programs.

1.1.2 Compatibility among programs

All sequence programs created by the aforementioned three types are stored in the form of instruction words (contents as at the time of list program) in the program memory inside the PLC.

- Programs created by these three types of input methods can be converted mutually, and then displayed and edited as shown in the figure below.



2. Overview (Sequence Program)

This chapter explains the basic functions of FX3U/FX3UC PLCs.
This chapter includes not only the features of PLCs but also introduction of representative functions, parameters and memory to utilize the functions of PLCs. Read this chapter before designing sequences.

2.1 Introduction of Convenient Functions

FX3U/FX3UC PLCs have the following instruction functions.

2.1.1 Convenient functions for input processing

1. "High speed counter" function of one phase or two phases for counting high speed inputs

One-phase high speed counters can execute counting at up to 100 kHz (or 200 kHz when a special high speed input adapter is used) regardless of the operation time because they process high speed pulses from specific input relays as interrupts. (Two-phase counters can execute counting at up to 50 kHz (or 100 kHz when a special high speed input adapter is used).)
The counting result can be immediately handled as high speed counter output interrupts by specific program processing and high speed counter counted values by comparison instructions dedicated to high speed counters.

→ Related instructions: High speed counter compare; HSCS (FNC 53), HSCR (FNC 54) and HSZ (FNC 55)

If the number of high speed counters is insufficient, special extension equipment (high speed counter blocks) can be connected.
By extending hardware counters in the high speed counter block, high speed pulses at up to 50 kHz can be received (except 1 and 4 edge count).

→ FX2N-1HC high speed counter block

2. "I/O refresh" function for receiving the latest input information

The input terminal information of the PLC in the batch refresh method is input all at once by the input image memory before step 0. The output information is output at one time when END instruction is executed.
I/O refresh instruction can get the latest input information and immediately output the operation result during sequence operation.

→ Related instruction: Refresh REF (FNC 50)

3. "Input filter adjustment" function for changing the time constant of input relays

Input relays in the PLC are equipped with a C-R filter of approximately 10 ms as countermeasures against chattering and noise in input signals. Because a digital filter is adopted for the input relays X000 to X017*1, however, the filter value can be changed in sequence programs.

→ Related instruction: Refresh and filter adjust instruction REFF (FNC 51)

*1. X000 to X007 in the FX3U-16M□

4. "Pulse catch" function

The pulse catch function is provided as a method to receive short-time pulse signals.
The pulse catch function monitors signals from specific input relays, and sets special auxiliary relays in the interrupt processing as soon as signals are input.
The pulse catch function can be used in a wide range of applications because even narrow pulses can be easily received.
When complicated operations should be processed with high priority as interrupt by using specific trigger signals, the "interrupt" function described later is suitable.

→ Refer to Section 35.7.

5. Three types of "interrupt" functions for receiving short-period pulses and priority processing

→ Refer to Chapter 35.

1) Input interrupt
Signals from specific input relays are monitored. At the rising edge or falling edge of the monitored input, a specified interrupt routine is executed with highest priority.

2) Timer interrupt
Specified interrupt routines are executed with highest priority at every specified time.

3) Counter interrupt
Depending on the present value of a high speed counter, a specified interrupt routine is executed with highest priority.

2.1.2 Convenient functions for output processing

1. "I/O refresh" function for outputting the latest input information

The input terminal information of the PLC in the batch refresh method is input at one time by the input image memory before operation in the step 0. The output information is output at one time when END instruction is executed.
I/O refresh instruction can get the latest input information and immediately output the operation result during sequence operation.

→ Related instruction: Refresh REF (FNC 50)

2. "Pulse output" function for pulse train output control

→ Related instructions: Pulse Y Output PLSY (FNC 57) and Acceleration/Deceleration Setup PLSR (FNC 59)

3. "Positioning" function for positioning control

→ Related instructions: DOG Search Zero Return DSZR (FNC150), Interrupt Positioning DVIT (FNC151), Zero Return (FNC156), Variable Speed Pulse Output PLSV (FNC157), Drive to Increment DRVI (FNC158) and Drive to Absolute DRVA (FNC159)

2.1.3 Functions for supporting sequence control

1. "Constant scan" mode for making the operation cycle of the PLC constant

The operation cycle in the PLC adopting the cyclic operation method varies depending on the contents of the program execution.
In the constant scan mode (M8039 and D8039), the operation cycle can be made constant. As a result, instructions executed in synchronization with the operation can be processed in a constant cycle.

2. "All outputs disable" mode for turning OFF all output signals

When the special auxiliary relay M8034 is driven, the output latch memory is cleared. Accordingly, all output relays (Y) turn OFF while the PLC is continuing its operation.
However, the status of output relays (Y) in each device image memory is not cleared. As a result, when devices are monitored using a programming tool, they may be regarded as the ON status.

3. "Memory hold stop" function for holding the output status during the RUN mode even in the STOP mode

When the special auxiliary relay M8033 is driven, the PLC is stopped while the output status during the RUN mode is held.

4. Registration of "entry code" for protecting programs

The entry code can be registered to prevent erroneous wiring to and plagiarism of created sequence programs.
With regard to online operations from GX Developer (Ver.8.23Z or later) and handy programming panels, the program protection level can be set by the entry code specification method. In this case, such specification that "changes of a program are disabled, but monitoring and changes of present values are enabled" is available.

→ **Refer to the manual of the used programming tool.**

5. Addition of "comments" for a sequence program

By setting parameters, the device comment area (where Katakana, Kanji and alphanumeric characters are available) can be secured in the program memory.

→ **Refer to the manual of the used programming tool.**

6. Writing programs in the RUN mode

Programs can be changed while the PLC is operating (RUN mode).
By this function, programs can be adjusted and changed efficiently without stopping the machine.

→ **Refer to the manual of the used programming tool.**

2.2 Introduction of Applied Instructions

1. Excellent fundamental performance

FX3U/FX3UC PLCs are equipped with not only fundamental applied instructions for data transfer, data comparison, arithmetic operations, logical operations, data rotation and data shift but also with high speed processing instructions for I/O refresh, interrupt, comparison dedicated to high speed counters and high speed pulse output as well as initial state instructions by which standard operations for machine control are made into packages in the SFC control. FX PLCs have the specifications offering fundamental functions, high speed processing and good operability.

2. Advanced control available easily

In addition, FX PLCs offer many handy instructions by which complicated sequence control is made into packages to mitigate the load for creating sequence programs and save the number of I/O points.
FX PLCs also offer floating point arithmetic operations and PID operations to cope with more advanced control.

2.2.1 Major applied instructions

This subsection introduces representative ones among many applied instructions provided in FX3U/FX3UC PLCs.

1. Program flow

- Conditional jump (CJ/FNC 00)
- Call subroutine (CALL/FNC 01)
- Enable interrupt (EI/FNC 04)
- Disable interrupt (DI/FNC 05)
- Start a FOR/NEXT loop (FOR/FNC 08)

→ **Refer to Chapter 8.**

2. Move and compare

- Compare (CMP/FNC 10)
- Data comparison (FNC224 to FNC246)
- Floating point compare (ECMP/FNC110 and EZCP/FNC111)
- Zone compare (ZCP/FNC 11)
- High speed counter compare (FNC 53 to FNC 55)
- High speed counter compare with data table (HSCT/FNC280)
- Move (MOV/FNC 12)
- Floating point move (EMOV/FNC112)
- High speed counter move (HCMOV/FNC189)
- Conversion to binary-coded decimal (BCD/FNC 18)
- Conversion to binary (BIN/FNC 19)
- Decimal to gray code conversion (FNC170) and gray code to decimal conversion (FNC171)

→ **Refer to Chapter 9, Chapter 13, Chapter 18, Chapter 22, Chapter 24, Chapter 28 and Chapter 32.**

3. Arithmetic and logical operations

- Addition (ADD/FNC 20)
- Subtraction (SUB/FNC 21)

1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

- Multiplication (MUL/FNC 22)
- Division (DIV/FNC 23)
- Increment (INC/FNC 24)
- Square root (SQR/FNC 48)
- Trigonometry (FNC130 to FNC135)
- Conversion from/to floating point (FNC 49, FNC118, FNC119 and FNC129)
- Floating point arithmetic operations (FNC120 to FNC123)
- Floating point square root (ESQR/FNC127)

→ **Refer to Chapter 10, Chapter 12 and Chapter 18.**

4. Rotation and shift operation

- Rotation right (ROR/FNC 30)
- Rotation left (ROL/FNC 31)
- Rotation right with carry (RCR/FNC 32)
- Rotation left with carry (RCL/FNC 33)
- Bit shift right (SFTR/FNC 34)
- Bit shift left (SFTL/FNC 35)
- Word shift right (WSFR/FNC 36)
- Word shift left (WSFL/FNC 37)

→ **Refer to Chapter 11.**

5. Data operation

- Zone reset (ZRST/FNC 40)
- Decode (DECO/FNC 41)
- Encode (ENCO/FNC 42)
- Sum of active bits (SUM/FNC 43)
- Mean (MEAN/FNC 45)
- Word to byte (WTOB/FNC141) and byte to word (BTOW/FNC142)
- 4-bit linking/grouping of word data (FNC143 and FNC144)
- Limit control (LIMIT/FNC256)
- Dead band control (BAND/FNC257)
- Zone control (ZONE/FNC258)
- Block data operation (FNC192 to FNC199)
- Character string control (FNC200 to FNC209)

→ **Refer to Chapter 12, Chapter 19, Chapter 25, Chapter 26 and Chapter 29.**

6. High speed processing

- Refresh (REF/FNC 50)
- Refresh and filter adjust (REFF/FNC 51)
- Speed detection (SPD/FNC 56)
- Pulse Y output (PLSY/FNC 57)
- Pulse ramp (PLSR/FNC 59)

→ **Refer to Chapter 13.**

7. Handy instructions and instructions for external devices

- Initial state (IST/FNC 60)
- Teaching timer (TTMR/FNC 64)
- Alternate state (ALT/FNC 66)
- Ramp variable value (RAMP/FNC 67)
- Rotary table control (ROTC/FNC 68)
- Ten-key input (TKY/FNC 70)
- Digital switch (thumbwheel input) (DSW/FNC 72)
- Seven-segment decoder (SEGD/FNC 73)
- Seven-segment with latch (SEGL/FNC 74)
- ASCII code data input (ASC/FNC 76)
- BFM Read, BFM Write(FNC 78, FNC 79, FNC278, and FNC279)
- Serial communication (FNC 80 and FNC 87)
- Inverter communication (FNC270 to FNC274)
- Hexadecimal to ASCII conversion (ASCI/FNC 82)
- ASCII to hexadecimal conversion (HEX/FNC 83)
- Cyclic redundancy check (CRC/FNC188)
- Random number generation (RND/FNC184)
- Real time clock control (FNC160 to FNC167)
- Hour meter (HOUR/FNC 169)
- Timing pulse generation (DUTY/FNC186)
- Logging R and ER (LOGR/FNC293)

→ **Refer to Chapter 14, Chapter 15, Chapter 16, Chapter 21, Chapter 24, Chapter 30, Chapter 31 and Chapter 33.**

8. Complicated control

- Search a data stack (SER/FNC 61)
- Sort tabulated data (FNC 69 and FNC149)
- PID control loop (PID/FNC 88)

→ **Refer to Chapter 14, Chapter 16 and Chapter 19.**

9. Positioning control

- Dog search zero return (DSZR/FNC150)
- Interrupt positioning (DVIT/FNC151)
- Batch data positioning mode (TBL/FNC152)
- Absolute present value read (ABS/FNC155)
- Zero return (ZRN/FNC156)
- Variable speed pulse output (PLSV/FNC157)
- Drive to increment (DRVI/FNC158)
- Drive to absolute (DRVA/FNC159)

→ **Refer to Chapter 20.**
→ **Refer to the Positioning Control Manual.**

2.3 Analog/Positioning Special Control

For the details, refer to the manual of each product.

1. Analog I/O control

- Analog input
- Analog output
- Pt100 temperature sensor input
- Thermocouple temperature sensor input
- Block dedicated to temperature control

→ **Refer to the manual of each product.**

2. Positioning control

- Pulse output block (controlled by sequence program)
- Positioning unit (controlled by instructions dedicated to positioning)
- Cam switch (resolver detection)

→ **Refer to the manual of each product.**

3. High speed counter

- High speed counter (hardware counter equipped with multiplication function)

→ **Refer to the manual of each product.**

2.4 Link and Communication

FX3U/FX3UC PLCs support the following communication functions:

1. CC-Link

The CC-Link system can be constructed with an FX3U /FX3UC PLC working as the master station. Or an A or QnA PLC can work as the master station, and FX PLCs can be connected as slave stations (remote device stations).
The CC-Link is an open network allowing connection of not only FX PLCs but also inverters, AC servo systems, and sensors.

→ **Refer to each manual packed together.**

2. CC-Link/LT

The CC-Link/LT system can be constructed with an FX3U/FX3UC PLC working as the master station.
General X (input) and Y (output) devices are assigned to remote I/O units, and operated by programs for general-purpose I/O.

→ **Refer to the FX3UC Hardware Edition for the built-in type CC-Link/LT master.**
→ **Refer to the product manual for the FX2N-64CL-M.**

3. MELSEC I/O LINK

The MELSEC I/O LINK is a remote I/O system whose master station is an FX3U/FX3UC PLC.
Units for MELSEC I/O LINK remote I/O system (A PLCs) can be used as remote units.

→ **Refer to each manual packed together.**

4. AS-i system

A network system at the actuator or sensor level can be constructed with an FX3U/FX3UC PLC working as the master station in the AS-i system.

→ **Refer to each manual packed together.**

5. Simple N : N link

Up to eight FX3U/FX3UC PLCs are connected, and data are automatically transferred among them.

→ **Refer to the Data Communication Edition.**

6. Parallel link

Two PLCs are connected, and data are automatically transferred between them.

→ **Refer to the Data Communication Edition.**

7. Computer link

A computer such as personal computer works as the master station, up to sixteen FX and A PLCs are connected to the master station, the master station directly specifies devices in the PLCs, and then data are transferred.
Protocols in the computer link support the formats 1 and 4.
By using MX Component and MX Sheet, monitoring and logging for the PLC system can be easily set by Microsoft Excel.

→ **Refer to the Data Communication Edition.**
→ **For MX Component and MX Sheet, refer to the manual of each product.**

1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

8. No-protocol communication

No-protocol serial communication is available between an FX PLC and interface equipment in accordance with RS-232C/RS-485 such as bar code reader, printer, personal computer and measuring instrument.

→ **Refer to the Data Communication Edition.**

9. Inverter communication

An FX PLC can control up to eight inverters via communication in accordance with RS-485.

→ **Related instructions: IVCK (FNC270)**
IVDR (FNC271)
IVRD (FNC272)
IVWR (FNC273)
IVBWR (FNC274)

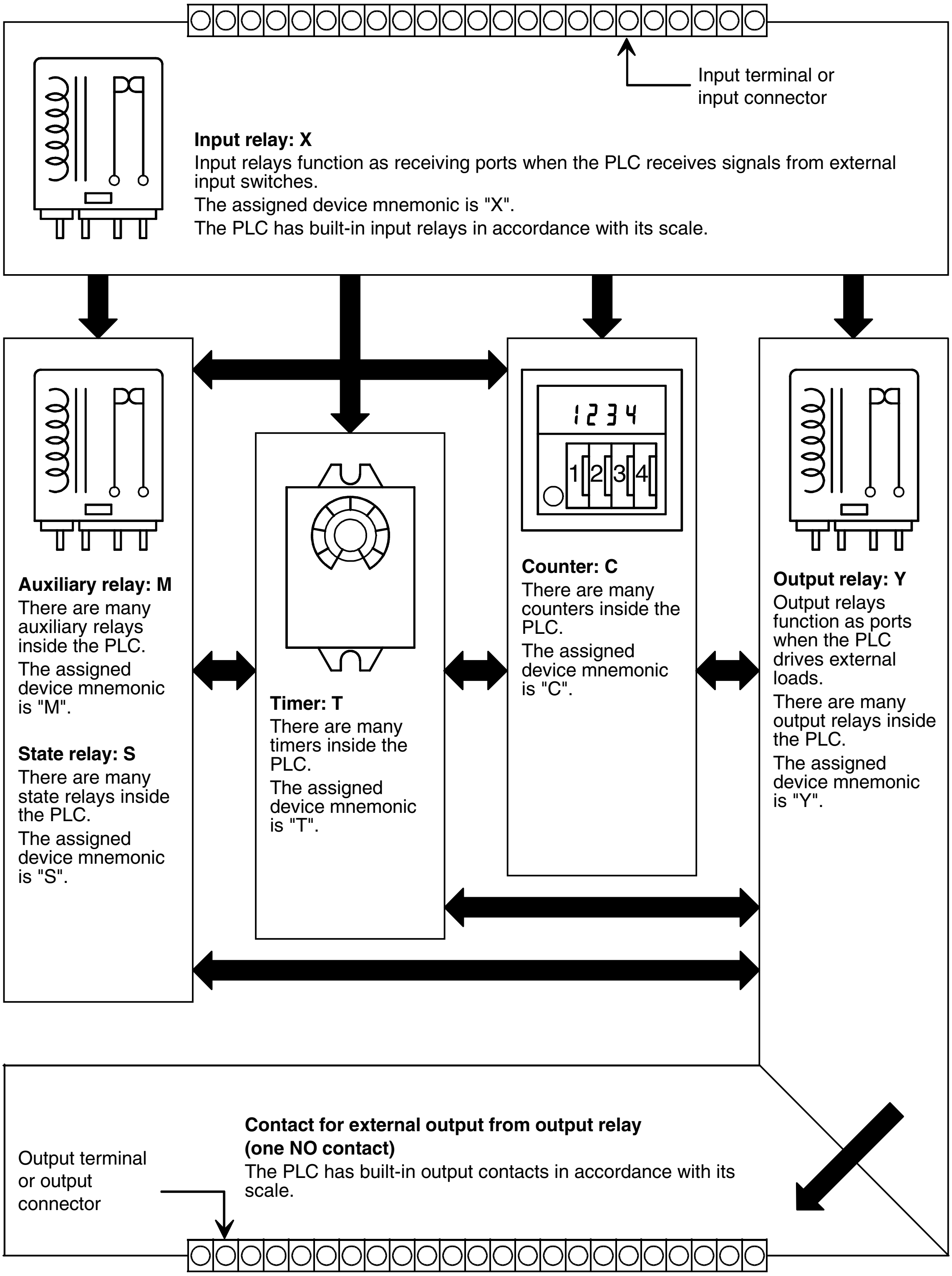
→ **Refer to the Data Communication Edition.**

2.5 Introduction of Devices Constructing PLC

Many relays, timers, and counters are built into an FX3U/FX3UC PLC, with many NO (normally open) contacts and NC (normally closed) contacts.
These contacts and coils are connected to make a sequence circuit.
A PLC is also equipped with data registers (D) and extension data registers (R) functioning as memory devices to store numeric data values.

2.5.1 Relationship among devices

Arrows show transfer of signals.



2.5.2 Device list

1. Input relays (X) and output relays (Y)

→ **Refer to Section 4.2.**

- Input relay numbers and output relay numbers in octal are assigned to each main unit in the way "X000 to X007, X010 to X017 …, Y000 to Y007, Y010 to Y017 …"
  The input relay (X) numbers and output relay (Y) numbers in extension units and extension blocks are also serial numbers in octal respectively in the order of connection to the main unit.
- A digital filter is applied to the input filter of specific input relays, and the filter value can be changed by a program. Accordingly, for a purpose requiring high speed receiving, assign such input relay numbers.
  (Refer to explanation of filter adjustment, input interrupt, high speed counter, various applied instructions, etc.)

2. Auxiliary relays (M)

→ **Refer to Section 4.3.**

- Relays built into the PLC are auxiliary relays, and are used for programs. Different from I/O relays, auxiliary relays cannot receive external inputs or directly drive external loads.
- There are latched (battery backed) type relays whose ON/OFF status is stored even if the PLC turns OFF.

3. State relays (S)

→ **Refer to Section 4.4.**

- State relays are used in the step ladder or as process numbers in the SFC expression.
- When a state relay is not used as a process number, it can be programmed as a general contact/coil in the same way as an auxiliary relay.
- State relays can be used as annunciators for external fault diagnosis.

4. Timers (T)

→ **Refer to Section 4.5.**

- A timer adds and counts clock pulses of 1, 10 or 100 ms, and its output contact turns ON or OFF when the counted result reaches a specified set value.
  A timer can count from 0.001 to 3276.7 seconds depending on the clock pulse.
- The timers T192 to T199 are dedicated to subroutines and interrupt routines.
  The timers T250 to T255 are retentive type base clock timers for 100 ms pulses. It means that the present value is retained even after the timer coil drive input turns OFF. And when the drive input turns ON again, a retentive type timer will continue its counting from where it left off.

5. Counters (C)

The following types of counters are provided, and can be used in accordance with the purpose or application.

1) For latched (battery backed up) counters
   → **Refer to Section 4.6.**
   Counters are provided for internal signals of the PLC, and their response speed is usually tens of Hz or less.
   - 16-bit counter: Provided for up-counting, counting range: 1 to 32767
   - 32-bit counter: Provided for up-counting and down-counting, counting range: –2,147,483,648 to +2,147,483,647
2) For latched (battery backed up) high speed counters
   → **Refer to Section 4.7.**
   High speed counters can execute counting at several kHz without regard to operations in the PLC.
   - 32-bit counter: Provided for up-counting and down-counting, counting range: –2,147,483,648 to +2,147,483,647 (1-phase 1-counting, 1-phase 2-counting and 2-phase 2-counting), assigned to specific input relays

6. Data registers (D)

→ Refer to Section 4.8.

Data registers store numeric data values.
All data registers in FX PLCs are 16-bit type (whose most significant bit is positive or negative). When two registers are combined, they can handle 32-bit numeric value (whose most significant bit is positive or negative).
(For the numeric value range, refer to "Counter" on the previous page.)
In the same way as other devices, data registers are classified into general type and latched type (battery backed).

7. Extension resistors (R) and extension file registers (ER)

→ Refer to Section 4.9.

Extension registers (R) are the extended form of data registers (D). They are protected by the battery against power failure.
While a memory cassette is mounted, the contents of extension resisters (R) can be stored in extension file registers (ER). Extension file registers (ER) can be used only while a memory cassette is mounted.

8. Index registers (V)(Z)

→ Refer to Section 4.10.

Among registers, there are index type registers V and Z used for modification.
A data register V or Z is added to another device as follows:
[In the case of "V0, Z0 = 5"]
D100V0 = D105, C20Z0 = C25 ← Device number + V□ or Z□ value
Data registers and index registers are used for indirectly specifying the set value of timers and counters, or used in applied instructions.

9. Pointers (P)(I)

→ Refer to Section 4.11.

Pointers are classified into branch pointers and interrupt pointers.

- A branch pointer (P) specifies the jump destination of the conditional jump CJ (FNC 00) or the call subroutine CALL (FNC 01) instruction.
- An interrupt pointer (I) specifies the interrupt routine of an input interrupt, timer interrupt or counter interrupt.

10.Constants (K)(H)(E)

→ Refer to Chapter 5.

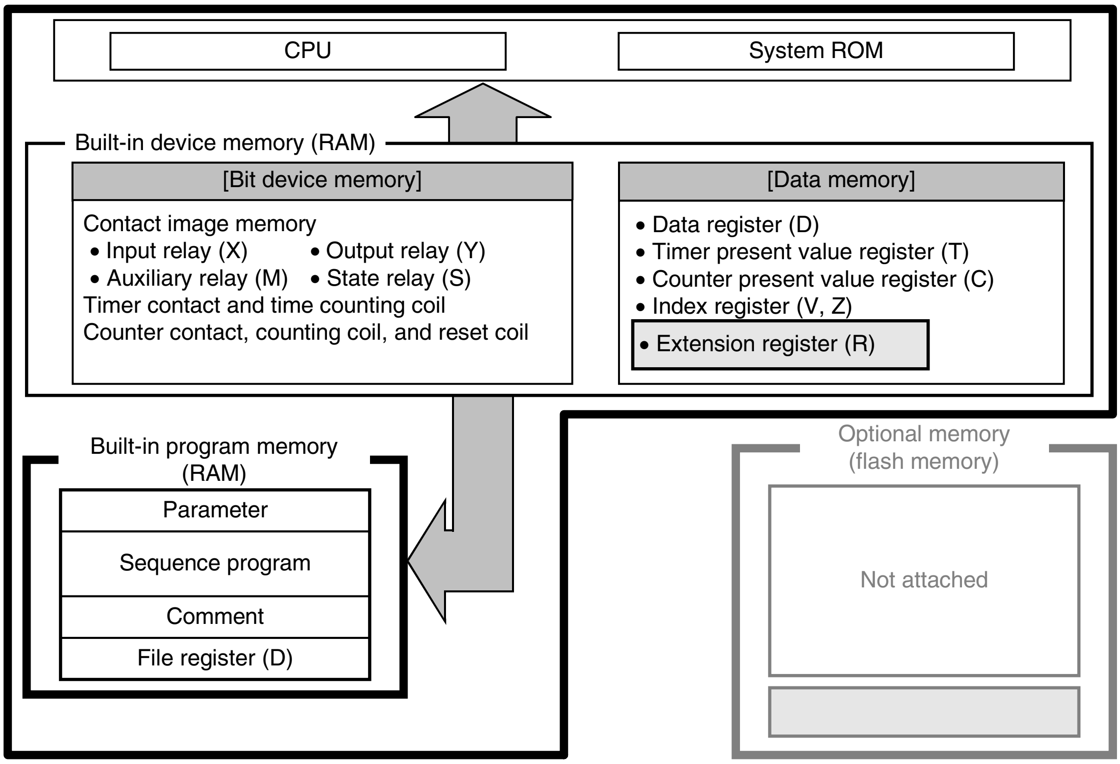
Among various numeric values used in the PLC, "K" indicates a decimal integer, value "H" indicates a hexadecimal numeric value, and "E" indicates a real number (floating point data).
They are used as the set value or present value of timers and counters, or operands of applied instructions.

2.6 Program Memory and Devices (Battery Backed)

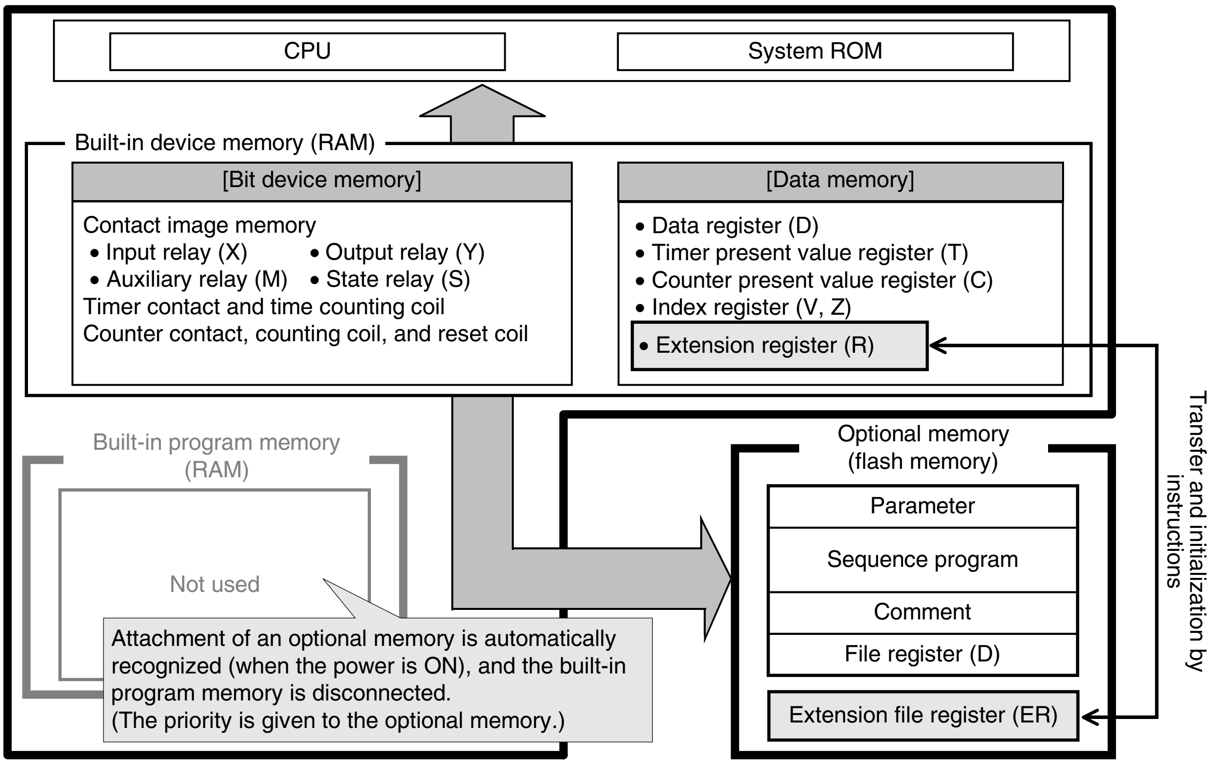
2.6.1 Memory structure

FX3U/FX3UC PLCs are supplied with RAM memory.
By mounting an optional memory device, the memory type can be changed.

1. When using the built-in memory (without attached optional memory)



2. When using an attached optional memory (without using the built-in memory)



2.6.2 Memory operations and latched (battery backed) (power ON/OFF and RUN/STOP)

1. Backup operation

The operations of the data memory, bit device memory and program memory in FX3U/FX3UC PLCs are classified as shown below:

1) Types of program memory

<table>
<tr><th colspan="2">Item</th><th>Power OFF</th><th>Power OFF→ON</th><th>STOP→RUN</th><th>RUN→STOP</th></tr>
<tr><td colspan="2">Parameter</td><td colspan="4">Does not change.</td></tr>
<tr><td colspan="2">Sequence program</td><td colspan="4">Does not change.</td></tr>
<tr><td>Comment</td><td rowspan="2">Can be secured by parameter setting.</td><td colspan="4">Does not change.</td></tr>
<tr><td>File register</td><td colspan="4">Does not change.</td></tr>
</table>

2) Types of word device memory

<table>
<tr><th colspan="2">Item</th><th>Power OFF</th><th>Power OFF→ON</th><th>STOP→RUN</th><th>RUN→STOP</th></tr>
<tr><td rowspan="5">Data register (D)</td><td rowspan="2">General type</td><td colspan="2" rowspan="2">Cleared.</td><td>Does not change.</td><td>Cleared.</td></tr>
<tr><td colspan="2">Does not change while M8033 is ON.</td></tr>
<tr><td>latched (battery backed) type</td><td colspan="4">Does not change.</td></tr>
<tr><td>File type</td><td colspan="4">Does not change.</td></tr>
<tr><td>Special type</td><td>Cleared.</td><td>Set to initial values.*1</td><td colspan="2">Does not change.*1</td></tr>
<tr><td>Extension register (R)</td><td>latched (battery backed) type</td><td colspan="4">Does not change.</td></tr>
<tr><td>Extension file register (ER)*2</td><td>File type</td><td colspan="4">Does not change.</td></tr>
<tr><td>Index register (V, Z)</td><td>V, Z</td><td colspan="2">Cleared.</td><td colspan="2">Does not change.</td></tr>
<tr><td rowspan="6">Timer present value register (T)</td><td rowspan="2">For 100 ms</td><td colspan="2" rowspan="2">Cleared.</td><td>Does not change.</td><td>Cleared.</td></tr>
<tr><td colspan="2">Does not change while M8033 is ON.</td></tr>
<tr><td rowspan="2">For 10 ms</td><td colspan="2" rowspan="2">Cleared.</td><td>Does not change.</td><td>Cleared.</td></tr>
<tr><td colspan="2">Does not change while M8033 is ON.</td></tr>
<tr><td>Retentive type for 100 ms</td><td colspan="4">Does not change.</td></tr>
<tr><td>Retentive type for 1 ms</td><td colspan="4">Does not change.</td></tr>
<tr><td rowspan="4">Counter present value register (C)</td><td rowspan="2">General type</td><td colspan="2" rowspan="2">Cleared.</td><td>Does not change.</td><td>Cleared.</td></tr>
<tr><td colspan="2">Does not change while M8033 is ON.</td></tr>
<tr><td>latched (battery backed) type</td><td colspan="4">Does not change.</td></tr>
<tr><td>High speed type</td><td colspan="4">Does not change.</td></tr>
<tr><td>Clock data</td><td>Present value</td><td colspan="4">Does not change.</td></tr>
</table>

*1. Some devices are cleared when the PLC status switches from STOP to RUN.

→ **For special data registers, refer to Chapter 36.**

*2. An optional memory cassette is required.

3) Types of bit device memory

Item		Power OFF	Power OFF→ON	STOP→RUN	RUN→STOP
Contact image memory (X, Y, M, S)	**Input relay (X)**	Cleared.		Does not change.	Cleared.
				Does not change while M8033 is ON.	
	Output relay (Y)	Cleared.		Does not change.	Cleared.
				Does not change while M8033 is ON.	
	General type auxiliary relay (M)	Cleared.		Does not change.	Cleared.
				Does not change while M8033 is ON.	
	latched (battery backed) type auxiliary relay (M)	Does not change.			
	Special type auxiliary relay (M)	Cleared.	Set to initial values.*1	Does not change.*1	
	General type state relay (S)	Does not change.			
	latched (battery backed) type state relay (S)	Does not change.			
	Annunciator (S)	Does not change.			
Timer contact Time counting coil (T)	**For 100 ms**	Cleared.		Does not change.	Cleared.
				Does not change while M8033 is ON.	
	For 10 ms	Cleared.		Does not change.	Cleared.
				Does not change while M8033 is ON.	
	Retentive type for 100 ms	Does not change.			
	Retentive type for 1 ms	Does not change.			
Counter contact Counting coil Reset coil (C)	**General type**	Cleared.		Does not change.	Cleared.
				Does not change while M8033 is ON.	
	latched (battery backed) type	Cleared.		Does not change.	Cleared.
				Does not change while M8033 is ON.	
	High speed type	Cleared.		Does not change.	Cleared.
				Does not change while M8033 is ON.	

*1. Some devices are cleared when the PLC status switches from STOP to RUN.

→ **For special auxiliary relays, refer to Chapter 36.**

2.6.3 Types of backup methods against power failure

There are following types of latch (battery backup) for the program memory and built-in devices in the PLC.

1. Battery backup method

Item	Description
Latched (battery backed) contents	A lithium battery backs up the RAM memory, devices (battery backed) and clock data built in the PLC.
Maintenance	Periodic replacement of the battery once in four or five years.
Cautions	1) When the battery life is expired, sequence programs and other latched (battery backed) contents are lost. 2) When an optional memory cassette (flash memory) is mounted, it is not necessary to back up sequence programs by the battery.

2. Flash memory backup method

Item	Description
Latched (battery backed) contents	1) The flash memory built into the memory cassette backs up sequence programs. 2) A battery is required to back up latched (battery backed) devices and clock data from failure.
Maintenance	Maintenance is not necessary.
Cautions	The upper limit is set to the number of times for overwriting. (Refer to the Hardware Edition of the main unit.)

2.6.4 Change between general devices and latched (battery backed) devices

1. When using latched (battery backed) type devices as non-latch type devices

In FX3U/FX3UC PLCs, some latched (battery backed) type devices can be changed into non-latch type devices by the parameter settings described later.
Devices dedicated to latched type cannot be changed into non-latch type devices even by the parameter settings. Such devices can be handled as non-latch type devices by clearing all latched (battery backed) type devices by the initial pulse (M8002) in a program.

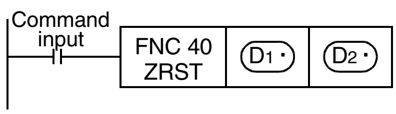
2. When using non-latch type devices as latched (battery backed) type devices

In FX3U/FX3UC PLCs, non-latch type devices can be changed into latched (battery backed) type devices by the parameter settings described later.

2.6.5 How to initialize devices (battery backed)

1. ZRST (FNC 40) instruction (zone reset)

By ZRST (FNC 40) instruction, devices in a specified range can be cleared at one time.



Devices D1· through D2· are reset at one time.

→ For details on ZRST (FNC 40) instruction, refer to Section 12.1.
→ In addition to ZRST instruction, FMOV (FNC 16) instruction is available also.
For details on FMOV (FNC 16) instruction, refer to Section 9.7.

2. M8032 (latch memory all clear)

When M8032 is turned ON, all latched (battery backed) type devices are cleared.

→ For details, refer to Subsection 36.2.11.

2.7 Types and Setting of Parameters

Setting of parameters means setting the environment where the PLC operates.
Almost all FX3U/FX3UC PLCs can be used with factory default values. When it is necessary to add optional memory, set the comment capacity, set the communication condition for serial ports, etc., change the parameter settings by a programming tool such as personal computer.

2.7.1 Parameter list

The following items may be set in the parameter settings.

Classification	Item	Description
Memory capacity	Memory capacity	This parameter specifies the maximum value for the number of steps to which a sequence program can be input. 1) The upper limit is determined by the capacity of the built-in memory or optional memory. 2) The program memory, file register, comment area, and other special setting capacities are contained in this memory capacity.
	Comment area	This parameter incorporates comments into the program memory. 1) Because comments remain in the PLC, the contents can be easily understood at the time of maintenance. 2) Up to 50 comments can be input when one block is specified, but the program memory capacity is reduced because the comment area requires 500 steps in the memory capacity.
	File register	This parameter incorporates data registers into the program memory. 1) A sequence program and control data such as machining set values can be handled together, which is convenient. 2) Up to 500 file registers can be created when one block is specified, but the program memory capacity is reduced because file registers require 500 steps in the memory capacity.
	Other special setting capacity	1) This parameter sets whether or not the special block/unit initial value setting function is used. When this function is used, the program memory capacity is reduced because this function requires 4000 steps (8 blocks) in the memory capacity. 2) This parameter sets whether or not the positioning setting (constants and setting table) in TBL (FNC152) instruction is used. When this setting is used, the program memory capacity is reduced because this setting requires 9000 steps (18 blocks) in the memory capacity.
Device setting	Latch range setting	This parameter enables to change the latched (battery backed) device range and the non-latch device range inside the PLC.
I/O assignment setting	I/O assignment setting	This setting is not written to the PLC. When the I/O range is set according to the system configuration, however, inputs and outputs are checked by the program check in GX Developer.
	Special unit setting	This parameter sets the initial values of the buffer memory (BFM) for each special block/unit number. It is necessary to set the memory capacity.
PLC system setting (1) [PLC mode]	Batteryless mode	This parameter sets the PLC operation mode without a battery. When the baterryless mode is set, detection of battery voltage low level error is stopped automatically, and consequently, contents of latched (battery backed) devices becomes inconsistent and are initialized automatically.
	Modem initialization	This parameter automatically sends a specified AT command as an initialization command to a modem connected to the serial port.
	RUN terminal input setting	This parameter sets whether one input terminal in the PLC is used for RUN input.
	RUN terminal input number	This parameter specifies the input number of the RUN input described above within the range from X000 to X017.
PLC system setting (2) [Serial communication]	Serial port operation setting	This parameter corresponds to the following settings by specifying each contents on the PC screen: Setting of communication format (D8120, D8400 and D8420) Setting of station number (D8121 and D8421) Setting of timeout check (D8129, D8409 and D8429)

Classification	Item	Description
Positioning setting	**Constant setting**	This parameter sets interrupt inputs for the maximum speed, bias speed, creep speed, zero return speed, acceleration time, deceleration time, and DVIT instruction. It is necessary to set the memory capacity.
	Detailed setting	This parameter sets the operation table. It is necessary to set the memory capacity.
Others	**Entry code**	This parameter sets protection to prevent erroneous writing and plagiarism of a sequence program. The entry code can be specified in 8 hexadecimal characters among A to F and 0 to 9. In FX3U and FX3UC PLCs Ver.2.20 or later, the second entry code (in 8 characters) can be added to allow specification of the entry code in 16 characters.
	Program title	This parameter enables to set a character string to be used as the program title.

2.7.2 Parameter initial values and available tools for changing parameter values

Item		GX Developer Initial value	Setting range	Initial value of non-applicable programming tool		Display unit
				FX-10P(-E)[*1] FX-20P(-E)[*1]	FX-PCS/WIN(-E)[*1]	GOT-F900 Series[*2] ET-940 Series[*3]
Memory capacity (steps)	**Program capacity**	16000[*4]	Refer to Subsection 2.7.3.	2000	8000	8000
	Katakana character comment capacity	0		0	0	–
	File register capacity	0		0	0	0
	Special unit initial value setting[*5]	Not used		–	–	–
	Positioning setting[*5]	Not used		–	–	–
Latch range (battery backed)	**Auxiliary relay [M]**	500 to 1023	0 to 1023	500 to 1023		
	State relay [S]	500 to 999	0 to 999	500 to 999		
	Counter [C] (16 bits)	100 to 199	0 to 199	100 to 199		
	Counter [C] (32 bits)	220 to 255	200 to 255	220 to 255		
	Data register [D]	200 to 511	0 to 511	200 to 511		
Program title		Not registered	Refer to Subsection 2.7.1.	–	Not registered	–
Entry code		Not registered		Not registered	Not registered	–
Batteryless mode		OFF		–	OFF	–
Modem initialization specification		Not set		–	Not set	–
RUN terminal input		OFF		Not used	Not used (X0)	–
Serial port operation setting		Not set		–	Not set	–

*1. These programming tools are not applicable to FX3U/FX3UC PLCs. The initial values in FX2N PLCs are shown above.

*2. Parameter values can be changed only by the F940WGOT, F94□GOT and F94□ handy GOT.

*3. Only manuals in Japanese are available for the ET-940 Series.

*4. The initial value is 8000 steps in GX Developer Ver.8.13P to Ver.8.22Y.

*5. GX Developer Ver.8.23Z or later is applicable.

2.7.3 Memory capacity setting range

⊙ Built-in memory capacity ✓ Can be set by changing parameter.

Memory capacity setting		Comment capacity setting Unit: Block	File register capacity setting Unit: Block	Special unit initial value setting*1 Unit: Block	Positioning setting*1 Unit: Block
2000 steps	✓	0 to 3	0 to 3	–	–
4000 steps	✓	0 to 7	0 to 7	–	–
8000 steps	✓	0 to 15	0 to 14	8	–
16000 steps	✓	0 to 31	0 to 14	8	18
32000 steps	✓	0 to 63	0 to 14	8	18
64000 steps	⊙	0 to 127	0 to 14	8	18

Cautions on setting the memory capacity

When one block is set in each capacity setting, the memory capacity is reduced by 500 steps.
Each setting should satisfy the following expression:

Memory capacity setting range	≥	Comment capacity setting	+	File register capacity setting	+	Special unit initial value setting*1	+	Positioning setting*1

1) With regard to the comment capacity, up to 50 device comments can be set in one block.
2) With regard to the file register capacity, up to 500 (16-bit) file registers can be set in one block.
3) In the special unit initial value setting*1, 8 blocks (4000 steps) are used.
4) In the positioning setting*1, 18 blocks (9000 steps) are used.

*1. GX Developer Ver.8.23Z or later is applicable.

Caution

After changing the memory capacity setting, make sure to write both the programs and parameters to the PLC.
If only the parameters are written to the PLC, program errors (such as parameter error, circuit error and grammar error) may occur in the PLC.

2.7.4 Compatible optional memory model

Model name	Maximum number of steps	Memory type	Allowable number of times of writing	Remarks
FX3U-FLROM-64	64000	Flash memory	10000 times	Write-protect switch is provided.
FX3U-FLROM-16	16000	Flash memory	10000 times	Write-protect switch is provided.
FX3U-FLROM-64L	64000	Flash memory	10000 times	Write-protect switch and loader function are provided.

2.7.5 Entry code

By registering the entry code in a PLC, the functions of programming tools, display modules, and display units to change programs, monitor devices, and current value changing function in the PLC can be restricted (access restriction).

→ **For the operations and restricted functions of display modules, refer to the Hardware Edition of the PLC main unit.**
→ **For the operations and restricted functions of display units, refer to the manual of each display unit.**

1. Correspondence between PLC and programming tool

	Available characters	Number of registered characters	Applicable PLC versions	GX Developer	Applicability of programming tools and display units to FX3U/FX3UC		
					FX-10P(-E) FX-20P(-E)	FX-PCS/WIN(-E)	GOT-F900 Series ET-940 Series*2
Entry code	0 to 9 and A to F	8	Ver.1.00 or later	Ver.2.00A or later*1	Applicable	Applicable	Only reset of entry code is allowed
Second entry code	0 to 9 and A to F	8 (16 characters which combined keyword <entry code> and 2nd keyword)	Ver.2.20 or later	Ver.8.23Z or later	Not applicable	Not applicable	Not applicable

*1. Ver.2.00A or later support FX series PLC, however, Ver.8.13P or later is required for FX3UC series PLCs and Ver.8.23Z or later is required for FX3U PLCs.

*2. Only manuals in Japanese are available for the ET-940 Series.
The security by keyword (entry code) and 2nd keyword has its limits and is not always perfect.

1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

2. Entry code setting and access restriction

<table>
<tr><th>Setting status</th><th>Programming tool and access restriction</th></tr>
<tr><td>When both entry code and second entry code are set</td><td>
1) When a programming tool supports the second entry code
The programming tool performs the following operations in accordance with the selected registration condition:
<table>
<tr><th rowspan="2">Registration condition</th><th colspan="2">Program</th><th rowspan="2">Monitoring</th><th rowspan="2">Present value change</th></tr>
<tr><th>Read</th><th>Write</th></tr>
<tr><td>Write prohibited</td><td>✓</td><td>–</td><td>✓</td><td>✓</td></tr>
<tr><td>Read and write prohibited</td><td>–</td><td>–</td><td>✓</td><td>✓</td></tr>
<tr><td>All online operations prohibited</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
</table>
2) When a programming tool does not support the second entry code
All online operations are disabled. The entry code cannot be changed or reset, and the PLC memory cannot be cleared.
</td></tr>
<tr><td>When only entry code is set</td><td>
1) In the case of handy programming panel FX-10P(-E)/FX-20P(-E)
The programming tool performs the following operations in accordance with the head character of the entry code (in 8 characters):
<table>
<tr><th rowspan="2"></th><th rowspan="2">Head character of entry code</th><th colspan="2">Program</th><th rowspan="2">Monitoring</th><th rowspan="2">Present value change</th></tr>
<tr><th>Read</th><th>Write</th></tr>
<tr><td>All operations prohibited</td><td>A,D to F,0 to 9</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>Anti-plagiarism</td><td>B</td><td>–</td><td>–</td><td>✓</td><td>✓</td></tr>
<tr><td>Erroneous write prohibited</td><td>C</td><td>✓</td><td>–</td><td>✓</td><td>✓</td></tr>
</table>
2) In the case of any programming tool other than FX-10P(-E)/FX-20P(-E)
Anti-plagiarism is set for all entry codes.
</td></tr>
<tr><td>When both entry code and second entry code are not set</td><td>All operations are enabled.</td></tr>
</table>

Caution on registering the entry code

- The entry codes are provided to restrict access from peripheral equipment to programs created by the users. Keep the entry codes carefully.
  If a registered entry code is forgotten, the online operations from the programming tool to the PLC are disabled depending on the programming tool type and the contents of the registered entry code.

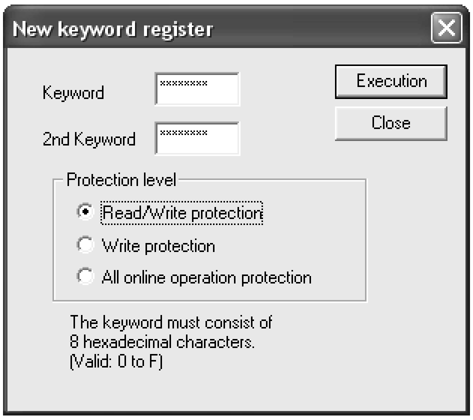
Registering and changing the entry codes

This section explains the operating procedure of GX Developer (Ver.8.23Z)

→ **For the entry code registration/change procedure in FX-10P(-E), FX-20P(-E), and FX-PCS/WIN(-E), refer to the manual of each product.**

1. Select [Online]-[Keyword setup]-[Register...] to open "New keyword register" dialog box.

2. Set the entry code, second entry code*1, and registration condition*1.



Set item	Contents of setting	Remarks
Keyword	Input 8 characters. Available characters are A to F and 0 to 9.	
2nd Keyword*1	Input 8 characters. Available characters are A to F and 0 to 9.	Before setting the second entry code, set the entry code first.
Protection level*1	Select either of the following: • Read/Write protection • Write protection • All online operation protection	Before setting the registration condition, set the 2ed entry code first.

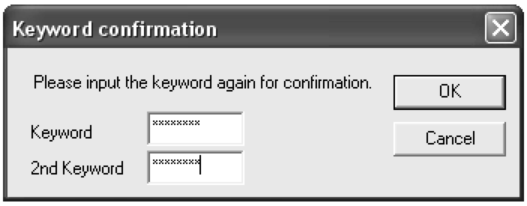
*1. The second entry code and registration condition can be set in FX3U and FX3UC PLCs Ver.2.20 or later.

Caution on registering the entry code

- The entry codes are provided to restrict access from peripheral equipment to programs created by the users. Keep the entry codes carefully.
  If a registered entry codes is forgotten, the online operations from a programming tool to the PLC are disabled depending on the programming tool type and the contents of the registered entry code.

3. Click [Execution] button to open "Keyword confirmation" dialog box.

4. Input the entry codes again.



5. Click [OK] to register the entry codes to the PLC.

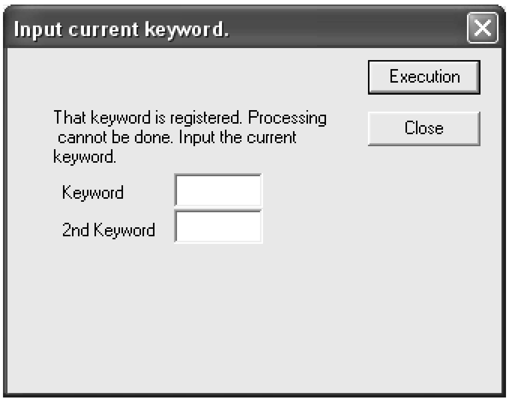
Reading/writing a program from/to a PLC with the entry codes registered

This section explains the operating procedure of GX Developer (Ver.8.23Z)

→ **For the program reading/writing procedure in FX-10P(-E), FX-20P(-E), and FX-PCS/WIN(-E), refer to the manual of each product.**

1. Select [Online]-[Read from PLC...]/[Write to PLC...] to open "Input current keyword." dialog box.

2. Input the entry code and second entry code[*1] currently registered in the PLC.



Set item	Contents of setting	Remarks
Keyword	Input 8 characters. Available characters are A to F and 0 to 9.	
2nd Keyword[*1]	Input 8 characters. Available characters are A to F and 0 to 9.	

*1. The second entry code can be set in FX3U and FX3UC PLCs Ver.2.20 or later.

3. Click [Execution] button to verify the entry codes you have input with the entry codes currently registered in the PLC.

- When the entry code inputs are verified, the PLC executes "Read from PC" or "Write to PC".
- When the entry code inputs are not verified, the PLC does not execute "Read from PC" or "Write to PC".

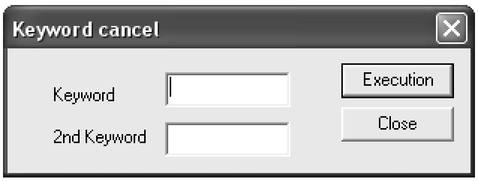
Canceling the entry codes

This section explains the operating procedure of GX Developer (Ver.8.23Z)

→ **For the entry code canceling (deletion) procedure in FX-10P(-E), FX-20P(-E), and FX-PCS/WIN(-E), refer to the manual of each product.**

1. Select [Online]-[Keyword setup]-[Delete...] to open "Keyword cancel" dialog box.

2. Input the entry code and second entry code currently registered in the PLC.



Set item	Contents of setting	Remarks
Keyword	Input 8 characters. Available characters are A to F and 0 to 9.	
2nd Keyword	Input 8 characters. Available characters are A to F and 0 to 9.	

3. Click [Execution] button to verify the entry codes you have input with the entry codes currently registered in the PLC.

- When the entry code inputs are verified, the PLC executes "Keyword Cancel".
- When the entry code inputs are not verified, the PLC does not execute "Keyword Cancel".

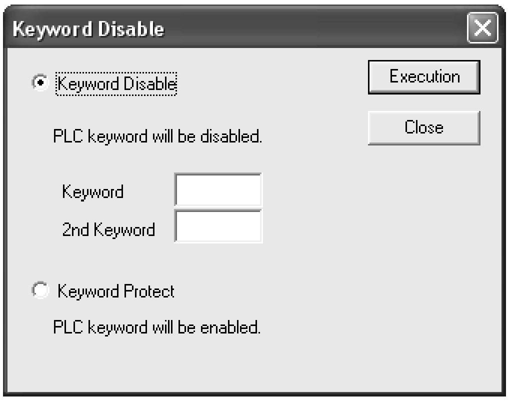
Resetting the entry codes, and validating the reset entry codes (Keyword Protect)

This section explains the operating procedure of GX Developer (Ver.8.23Z)

→ For the entry code reset procedure in FX-10P(-E), FX-20P(-E), and FX-PCS/WIN(-E), refer to the manual of each product.

1. Select [Online]-[Keyword setup]-[Disable...] to open "Keyword Disable" dialog box.

2. Input the entry code and second entry code.



Set item		Contents of setting
Keyword Disable	Keyword	Input 8 characters. Available characters are A to F and 0 to 9.
	2nd Keyword*1	Input 8 characters. Available characters are A to F and 0 to 9.
Keyword Protect*1		Reset entry codes are made valid again.

*1. The second entry code and entry code protect function can be set in FX3U/FX3UC PLCs Ver.2.20 or later.

3. Click [Execution] button to reset the entry codes or validate the reset entry codes again.

2.7.6 Special unit initial value setting [GX Developer Ver.8.23Z or later]

FX3UC Ver.2.20 → FX3U Ver.2.20 →

The initial values of the buffer memory (BFM) in special function blocks/units connected to an FX3U/FX3UC PLC (Ver.2.20 or later) can be set as a parameter in GX Developer (Ver.8.23Z or later).
When this parameter is used, it is not necessary to execute initial setting in a user program for special function blocks/units requiring initial setting. The special unit initial value setting uses 4000 steps (8 blocks) in the memory capacity.

→ For the setting procedure, refer to Subsection 2.7.8.

2.7.7 Positioning setting [for TBL (FNC152) instruction] [GX Developer Ver.8.23Z or later]

FX3UC Ver.2.20 → FX3U Ver.2.20 →

In the positioning setting in an FX3U/FX3UC PLC (Ver.2.20 or later), table and constants for added TBL (FNC152) instruction can be set. Make sure to set this parameter when using TBL (FNC152) instruction.
The positioning setting for TBL (FNC152) instruction uses 9000 steps (18 blocks) in the memory capacity.

→ For details on TBL (FNC152) instruction, refer to the Positioning Control Manual.
→ For the setting procedure, refer to the Positioning Control Manual or Subsection 2.7.8.

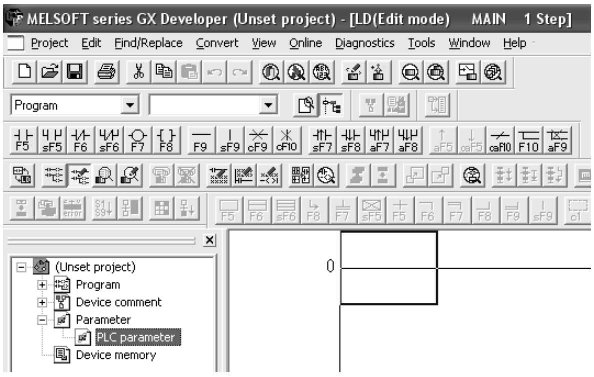
1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

2.7.8 Parameter settings by GX Developer

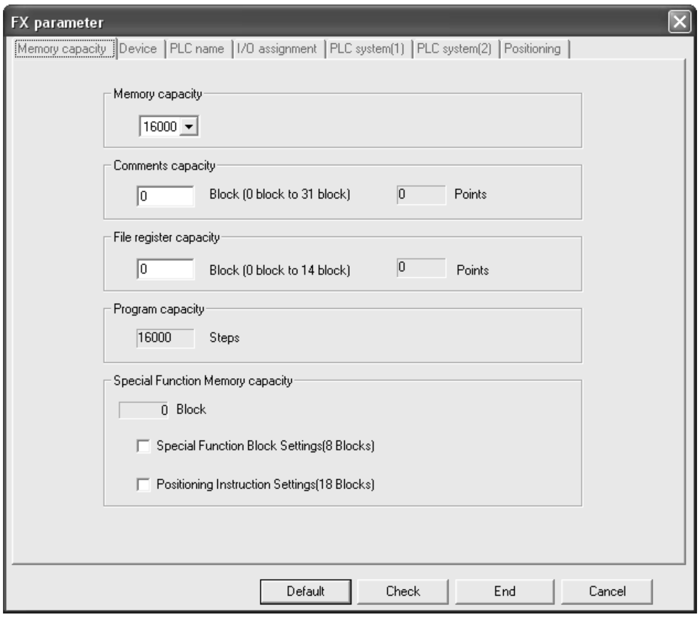
This subsection explains the parameter setting procedures by GX Developer (Ver.8.23Z).

1 Opening the parameter setting screen

In the project tree area provided on the left end of the screen, double-click [Parameter] → [PLC parameter].
If the project tree is not displayed, select [View] → [Project data list] from the menu bar.



2 Setting memory capacity

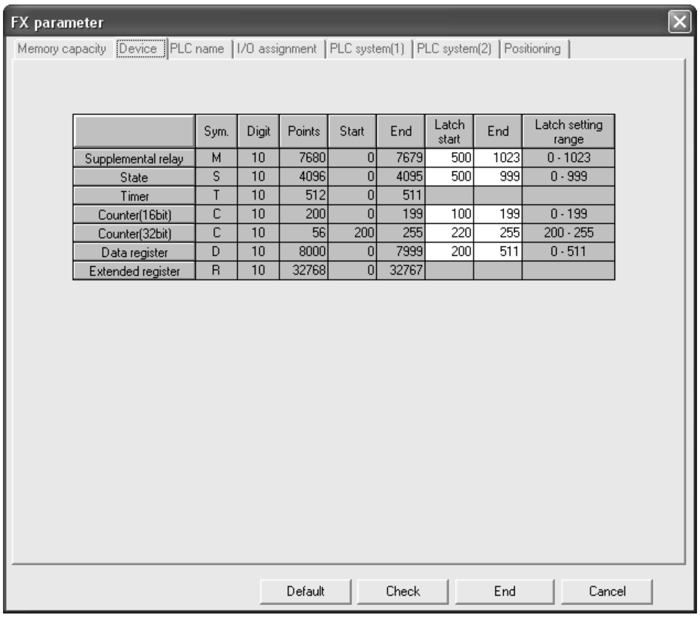


Set item	Contents of setting	Setting range
Memory capacity	Set the program memory capacity. Initial value: 16000*1	Refer to Subsection 2.7.3.
Comments capacity	Set the capacity of comments to be stored in the PLC. Initial value: 0 50 device comments/block (500 steps)	
File register capacity	Set the file register capacity. Initial value: 0 500 file registers/block (500 steps)	
Program capacity	The number of steps available for sequence program is displayed here.	
Special Function Memory capacity	Set whether the special unit initial value setting and positioning setting are valid or invalid.	–
Special Function Block Settings (8 Blocks)	Put a check mark to the initial value setting function for special function block/unit. (When this function is valid, the special unit setting is displayed on "I/O assignment" tab.)	–
Positioning Instruction Settings (18 Blocks)	Put a check mark to make valid TBL (FNC152) instruction setting function. (When this function is valid, "Positioning" tab is displayed.)	–

*1. The initial value is 8000 steps in GX Developer Ver.8.22Y and earlier.

3 Setting devices

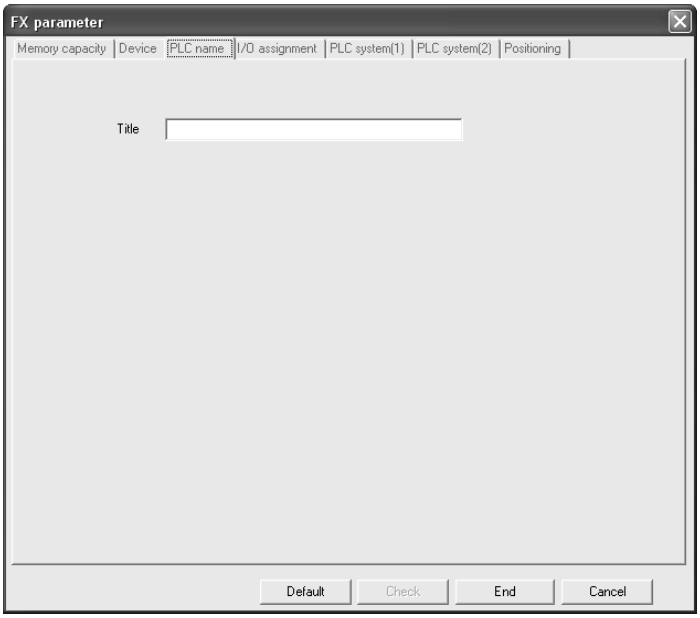
1. **Click "Device" tab, and set devices.**



Set item	Contents of setting	Setting range
Supplemental relay	Set the latched (battery backed) auxiliary relay range. Initial value: 500 to 1023	0 to 1023
State	Set the latched (battery backed) state relay range. Initial value: 500 to 999	0 to 999
Timer	The setting displayed here cannot be changed.	–
Counter (16bit)	Set the latched (battery backed) 16-bit counter range. Initial value: 100 to 199	0 to 199
Counter (32bit)	Set the latched (battery backed) 32-bit counter range. Initial value: 220 to 255	220 to 255
Data register	Set the data register range (battery backed). Initial value: 200 to 511	0 to 511
Extended register	All extension registers are latched (battery backed). This setting is fixed, and cannot be changed.	–

4 Setting the PC name

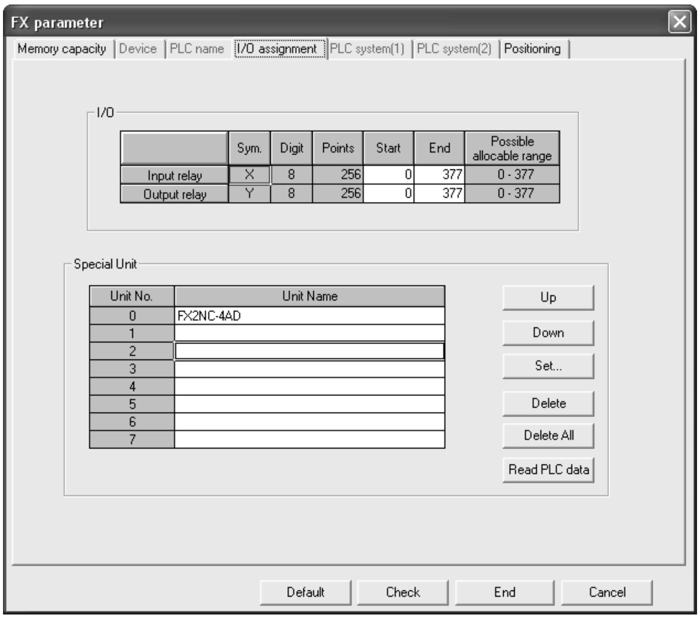
1. **Click "PLC name" tab, and input the program title.**



Set item	Contents of setting	Setting range
Title	Input the program title in up to 32 half-width characters (or 16 full-width characters).	32 half-width characters (or 16 full-width characters)

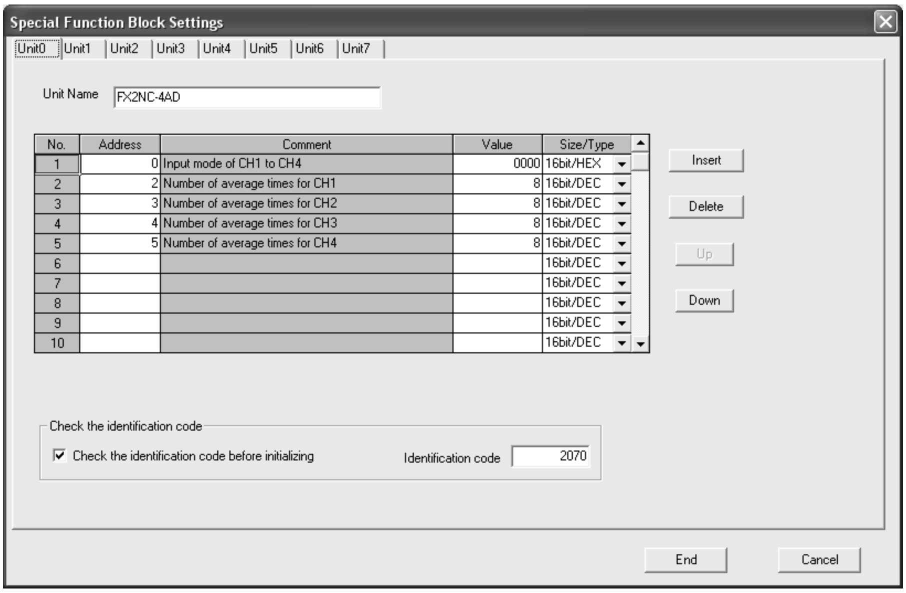
5 Assigning I/Os and setting the initial values for special unit

1. Click "I/O assignment" tab, and then set the I/O assignment and special function blocks/ units.
When inputting "Special Unit" field, it is necessary to put a check mark to "Special Function Block Settings" on "Memory capacity" tab.



Set item		Contents of setting	Setting range
I/O	Input relay	When the I/O range is set here according to the system configuration, those inputs and outputs are checked in programming by GX Developer.	0 to 377
	Output relay		0 to 377
Special Unit (It is necessary to set the memory capacity.)	Unit No.	This is the unit number of each special function block/unit.	–
	Unit Name	Set the name of each special function block/unit whose initial values are to be set.	32 half-width characters (or 16 full-width characters)
	Up	This button moves the cursor to the upper line (transposes the upper line).	–
	Down	This button moves the cursor to the lower line (transposes the lower line).	–
	Set...	This button displays "Special Function Block Settings" dialog box of the selected unit number. →Refer to the next page.	–
	Delete	This button deletes the setting of the selected unit number.	–
	Delete All	This button deletes all existing setting in "Special Unit" field.	–
	Read PLC data	This button reads "Special Unit" field from the connected PLC.	–

2. On "Special Function Block Settings" dialog box, set the initial values of special function blocks and units.



Set item	Contents of setting	Setting range
"Unit No." tab	Select the unit number of a special function block/unit to be set.	–
Unit Name	Set the name of a special function block/unit whose initial values are to be set. (The contents set on "I/O assignment" tab are displayed.)	32 half-width characters (or 16 full-width characters)
No.	This column indicates the order of initial value setting in the selected unit number. Numbers 1 to 98 can be set.	–
Address	Set the buffer memory address (BFM number) in a decimal value whose initial value is to be set.	*1
Comment	This column is displayed when device comments are registered. On the above screen, "Input mode of CH1 to CH4" is registered as the device comment for "U0\G0" (unit No. 0, BFM #0).	–
Value	Set a value to be set as the initial value of the buffer memory address (BFM number). Set the data length and type of the set value in "Size/Type" column.	*2
Size/Type	Select the size and type of a value set to the buffer memory among the following: 16bit/DEC 32bit/DEC 16bit/HEX 32bit/HEX	–
Insert	This button inserts a line in the currently selected position.	–
Delete	This button deletes the currently selected line.	–
Up	This button moves the cursor to the upper line (transposes the upper line).	–
Down	This button moves the cursor to the lower line (transposes the lower line).	–
Check the identification code before initializing	Put a check mark to check the model code of the special function block/unit before initialization.	–
Identification code	Set the model code of the special function block/unit.	*3

*1. Input buffer memory addresses (BFM numbers) that in the connected special function block/unit hold.

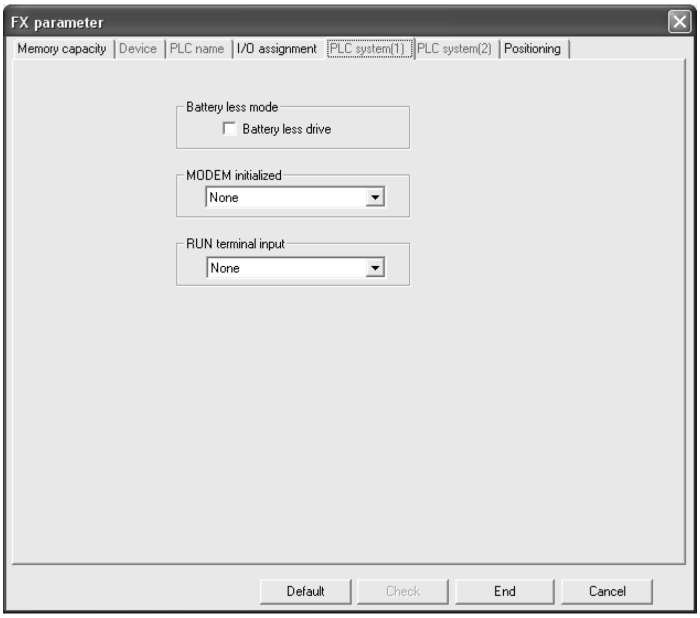
*2. To each buffer memory address (BFM number), set a value within the allowed range in the connected special function block/unit.

*3. Refer to the manual of the connected special function block/unit.

3. Click [End] button to finish the setting and close "Special Function Block Settings" dialog box.

6 Setting the PLC system (1)

Click "PLC system (1)" tab, and then set "Battery less mode", "MODEM initialized", and "RUN terminal input."



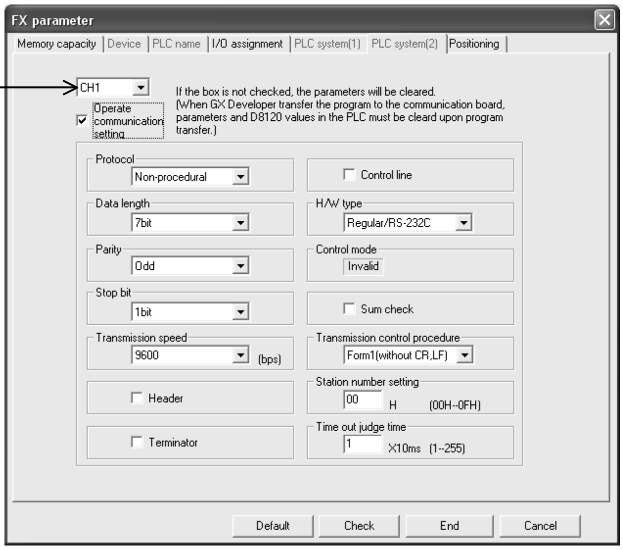
Set item	Contents of setting	Setting range
Battery less mode	Put a check mark to operate the PLC without using the battery. When a check mark is put here, the battery error indicator lamp is automatically turned off, and devices in the latch (battery backed) area are automatically cleared.	–
MODEM initialized	Set this item for automatically initializing a connected modem when the power of the PLC is turned ON.	–
RUN terminal input	Set this item to use one input terminal (X) for RUN input.	None X000 to X017[*1]

*1. X000 to X007 in the FX3U-16M□

7 Setting the PLC system (2)

1. **Click "PLC system (2)" tab.**
2. **Only when a latch (battery backed) area for a serial port exists through an extended PLC, select a channel to be set and put a check mark to "Operate communication setting." When not performing the communication setting for a serial port, do not put a check mark to "Operate communication setting."**

Channel selection



Set item	Contents of setting	Setting range
Channel selection	Select a channel in which a serial port is set.	CH1,CH2
Operate communication setting	Put a check mark when using the selected serial port in "computer link", "no-protocol communication" or "inverter communication". Do not put a check mark when transferring and monitoring sequence programs in GX Developer or when using the selected serial port in simple N : N link or parallel link.	–
Protocol	Set each item in accordance with application. → **For details on each item, refer to the communication control manual.**	
Data length		
Parity		
Stop bit		
Transmission speed		
Header		
Terminator		
Control line		
H/W type		
Control mode		
Sum check		
Transmission control procedure		
Station number setting		
Time out judge time		

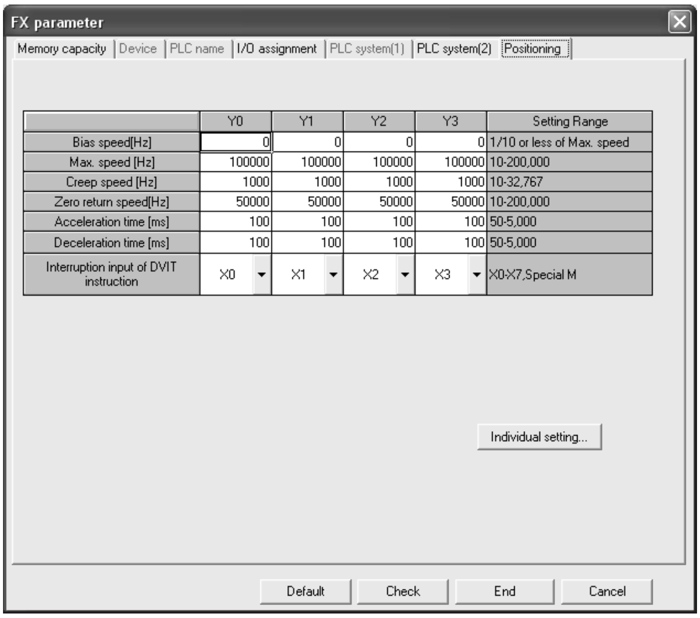
8 Setting positioning

The positioning setting function is available in FX3U/FX3UC PLCs Ver.2.20 or later.

1. Click "Positioning" tab.
"Positioning" tab is displayed when a check mark is put to "Positioning" on "Memory capacity" tab.

2. Set the positioning constants in TBL (FNC152) instruction.

→ For TBL (FNC152) instruction, refer to the Positioning Control Manual.



Set item	Contents of setting	Set range
Bias speed [Hz]	Set the bias speed for each output number of pulse. Initial value: 0	1/10 or less of the maximum speed
Max. speed [Hz]	Set the maximum speed for each output number of pulse. Initial value: 100,000	*1
Creep speed [Hz]	Set the creep speed in DSZR (FNC150) instruction for each output number of pulse. Initial value: 1000	10 to 32767*2
Zero return speed [Hz]	Set the zero point return speed in DSZR (FNC150) instruction for each output number of pulse. Initial value: 50000	*1
Acceleration time [ms]	Set the acceleration time for each output number of pulse. Initial value: 100	50 to 5000
Deceleration time [ms]	Set the deceleration time for each output number of pulse. Initial value: 100	50 to 5000
Interruption input of DVIT instruction	Set the interrupt input*3 for DVIT (FNC151) instruction for each output number of pulse. Specify a user interrupt command device (M) for a pulse output destination device not used in DVIT instruction. Initial setting: / Setting range: Pulse output destination Y000: X000 / X000 to X007, M8460 Pulse output destination Y001: X001 / X000 to X007, M8461 Pulse output destination Y002: X002 / X000 to X007, M8462 Pulse output destination Y003*4: X003 / X000 to X007, M8463	As shown on the left
Y0	They are set items for the pulse output destination Y000.	–
Y1	They are set items for the pulse output destination Y001.	–
Y2	They are set items for the pulse output destination Y002.	–
Y3*4	They are set items for the pulse output destination Y003.	–
Individual setting	This button displays "Positioning instruction settings" dialog box for setting the table used in TBL (FNC152) instruction. **→For the setting procedure, refer to the next step.**	–

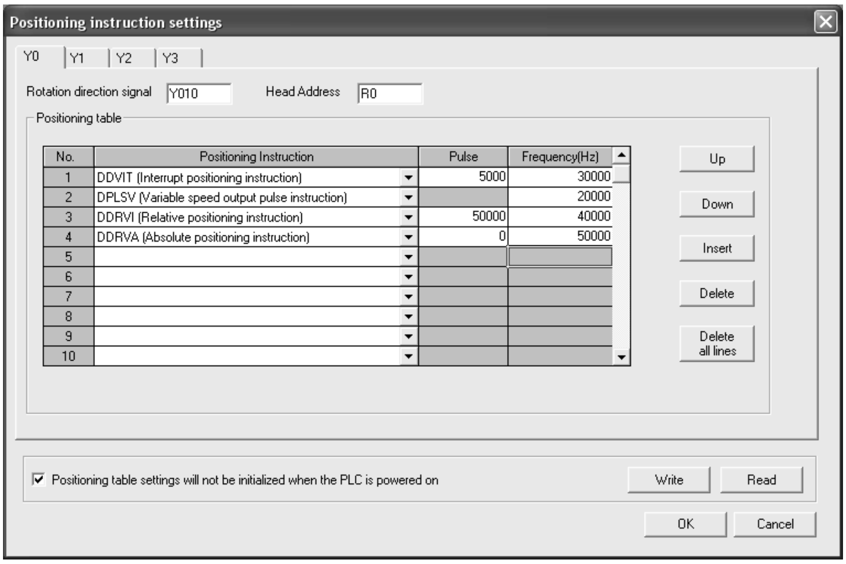
*1. The setting range is from 10 to 100,000 Hz in FX3UC PLCs.
The setting range is from 10 to 200,000 Hz in FX3U PLCs when the pulse output destination is the FX3U-2HSY-ADP.

*2. The creep speed should satisfy the relationship "Bias speed ≤ Creep speed ≤ Maximum speed."

*3. An interrupt input set here cannot be used jointly with a high speed counter, input interrupt, pulse catch input, input in SPD (FNC 67) instruction, or interrupt input in DVIT (FNC151) instruction.

*4. Y003 cannot be set in FX3U PLCs except when two FX3U-2HSY-ADP units are connected.

3. Click [Individual setting] button to display "Positioning instruction settings" dialog box. In this dialog box, set the positioning table for each pulse output destination.



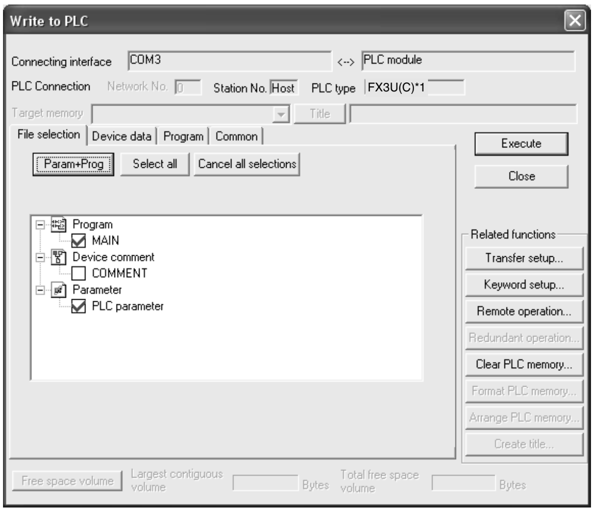
Set item	Contents of setting	Setting range
Y0	Set the positioning table for the pulse output destination Y000.	–
Y1	Set the positioning table for the pulse output destination Y001.	–
Y2	Set the positioning table for the pulse output destination Y002.	–
Y3*1	Set the positioning table for the pulse output destination Y003.	–
Rotation direction signal	Set the relay number of the rotation direction output signal. Initial setting: Pulse output destination Y000: Y010 Pulse output destination Y001: Y011 Pulse output destination Y002: Y012 Pulse output destination Y003*1: Y013 →**Refer to the Positioning Control Manual.**	Y000 to Y357 M0 to M7679 S0 to S4095
Head Address	Set the head number of devices storing the set data (pulse number and frequency). 1600 devices are occupied starting from the head device number set here without regard to the number of used axes. Initial setting: R0 →**Refer to the Positioning Control Manual.**	D0 to D6400 R0 to R31168
No.	This column shows the table number. Numbers 1 to 100 can be set.	–
Positioning Instruction	Select the positioning type among the following: DDVIT (Interrupt positioning instruction) DPLSV (Variable speed output pulse instruction) DDRVI (Relative positioning instruction) DDRVA (Absolute positioning instruction) →**Refer to the Positioning Control Manual.**	–
Pulse	Set the pulse number output by the operation (instruction) set in "Positioning Instruction" column. →**Refer to the Positioning Control Manual.**	Refer to the Positioning Control Manual.
Frequency [Hz]	Set the speed (pulse frequency) output by the operation (instruction) set in "Positioning Instruction" column. →**Refer to the Positioning Control Manual.**	Refer to the Positioning Control Manual.
Up	This button transposes the selected line to the upper line.	–
Down	This button transposes the selected line to the lower line.	–
Insert	This button inserts a line in the currently selected position.	–
Delete	This button deletes the currently selected line.	–
Delete all lines	This button deletes the entire setting of the positioning table for the selected pulse output destination.	–

Set item	Contents of setting	Setting range
Positioning table settings will not be initialized when the PLC is powered on	A check mark here means not to transfer the positioning setting when PLC turns ON. Put a check mark when changing the positioning setting from a display unit, etc., and then using the changed contents even after restoring the power. At this time, set a latched (battery backed) type device to "Head Address".	–
Write	This button writes from "Head Address" up to 1600 devices, the contents of the positioning table created here	–
Read	This button reads the existing positioning table contents starting from "Head Address" up to 1600 devices but does not read the device numbers without "positioning instruction" set.	–

*1. Y003 cannot be set in FX3UC PLCs except when two FX3U-2HSY-ADP units are connected to FX3U PLCs.

9 Transferring parameters (and sequence program) to the PLC

1. Select [Online]-[Write to PLC...] from the tool menu to display "Write to PLC" dialog box.



*1. For Ver. 8.13P to 8.24A of GX Developer, the PLC type is FX3UC.

2. Put a check mark to "Parameter", and click [Execute] button.

The selected contents are transferred to the PLC.
The transferred parameters become valid when the PLC switches from RUN to STOP.
When the communication setting is changed in step 7 "PLC system (2)", restore the PLC power.

Caution

After changing the memory capacity setting, make sure to write both the programs and parameters to the PLC.
If only the parameters are written to the PLC, program errors (such as parameter error, circuit error and grammar error) may occur in the PLC.

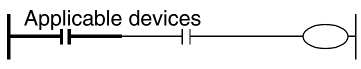
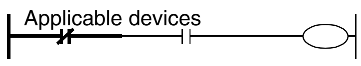
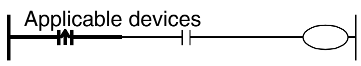
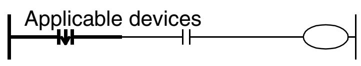
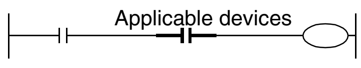
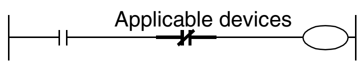
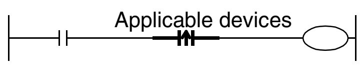
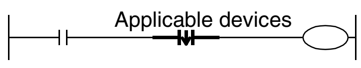
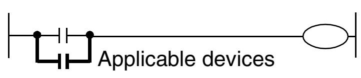
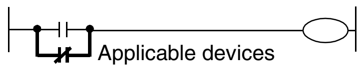
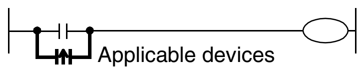
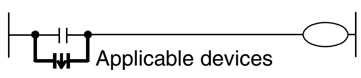
3. Instruction List

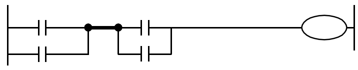
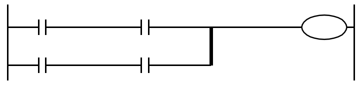
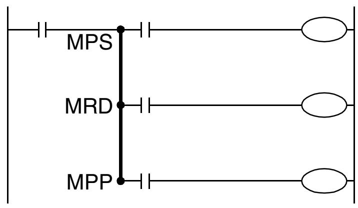
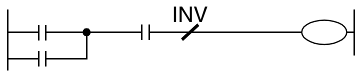
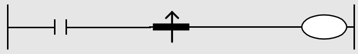
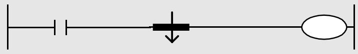
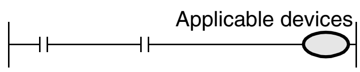
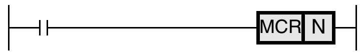
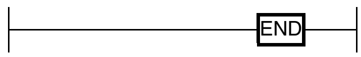
This chapter introduces a list of instructions available in programming.
Instructions added in FX3U and FX3UC PLCs are shaded in the list.

3.1 Basic Instructions

The basic instructions are provided in the following series. The table below shows differences in applicable devices.

Applicable PLC	FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC
All basic instructions	✓	✓	✓	✓	✓	✓	✓
Absence/presence of applicable devices (D□. b, R)	✓	✓	–	–	–	–	–

Mnemonic	Name	Symbol	Function	Applicable devices	Reference
Contact Instruction					
LD	Load		Initial logical operation contact type NO (normally open)	X,Y,M,S,D□.b,T,C	Section 7.1
LDI	Load Inverse		Initial logical operation contact type NC (normally closed)	X,Y,M,S,D□.b,T,C	Section 7.1
LDP	Load Pulse		Initial logical operation of Rising edge pulse	X,Y,M,S,D□.b,T,C	Section 7.5
LDF	Load Falling Pulse		Initial logical operation of Falling/trailing edge pulse	X,Y,M,S,D□.b,T,C	Section 7.5
AND	AND		Serial connection of NO (normally open) contacts	X,Y,M,S,D□.b,T,C	Section 7.3
ANI	AND Inverse		Serial connection of NC (normally closed) contacts	X,Y,M,S,D□.b,T,C	Section 7.3
ANDP	AND Pulse		Serial connection of Rising edge pulse	X,Y,M,S,D□.b,T,C	Section 7.5
ANDF	AND Falling Pulse		Serial connection of Falling/trailing edge pulse	X,Y,M,S,D□.b,T,C	Section 7.5
OR	OR		Parallel connection of NO (normally open) contacts	X,Y,M,S,D□.b,T,C	Section 7.4
ORI	OR Inverse		Parallel connection of NC (normally closed) contacts	X,Y,M,S,D□.b,T,C	Section 7.4
ORP	OR Pulse		Parallel connection of Rising edge pulse	X,Y,M,S,D□.b,T,C	Section 7.5
ORF	OR Falling Pulse		Parallel connection of Falling/trailing edge pulse	X,Y,M,S,D□.b,T,C	Section 7.5

Mnemonic	Name	Symbol	Function	Applicable devices	Reference
Connection Instruction					
ANB	AND Block		Serial connection of multiple parallel circuits	–	Section 7.7
ORB	OR Block		Parallel connection of multiple contact circuits	–	Section 7.6
MPS	Memory Point Store		Stores the current result of the internal PLC operations	–	Section 7.8
MRD	Memory Read		Reads the current result of the internal PLC operations	–	Section 7.8
MPP	Memory POP		Pops (recalls and removes) the currently stored result	–	Section 7.8
INV	Inverse		Invert the current result of the internal PLC operations	–	Section 7.10
MEP	MEP		Conversion of operation result to leading edge pulse	–	Section 7.11
MEF	MEF		Conversion of operation result to trailing edge pulse	–	Section 7.11
Out Instruction					
OUT	OUT		Final logical operation type coil drive	Y,M,S,D□.b,T,C	Section 7.2
SET	SET		SET Bit device latch ON	Y,M,S,D□.b	Section 7.13
RST	Reset		RESET Bit device OFF	Y,M,S,D□.b,T,C, D,R,V,Z	Section 7.13
PLS	Pulse		Rising edge pulse	Y,M	Section 7.12
PLF	Pulse Falling		Falling/trailing edge pulse	Y,M	Section 7.12
Master Control Instruction					
MC	Master Control		Denotes the start of a master control block	Y,M	Section 7.9
MCR	Master Control Reset		Denotes the end of a master control block	–	Section 7.9
Other Instruction					
NOP	No Operation		No operation or null step	–	Section 7.14
End Instruction					
END	END		Program END, I/O refresh and Return to Step 0	–	Section 7.15

3.2 Step Ladder Instructions

Mnemonic	Name	Symbol	Function	Applicable devices	Reference
STL	Step Ladder	STL | Applicable devices	Starts step ladder	S	Chapter 34
RET	Return	RET	Completes step ladder	–	Chapter 34

3.3 Applied Instructions ... in Ascending Order of FNC Number

Applied instructions such as Arithmetic operation, Rotation and Shift, Handy instructions etc. are used especially when numeric data is handled.
Instructions added in FX3U and FX3UC PLCs are shaded in the list.

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	FX3U	FX3UC	Applicable PLC: FX1S	FX1N	FX2N	FX1NC	FX2NC	Reference
Program Flow											
00	CJ	CJ | Pn	Conditional Jump	✓	✓	✓	✓	✓	✓	✓	Section 8.1
01	CALL	CALL | Pn	Call Subroutine	✓	✓	✓	✓	✓	✓	✓	Section 8.2
02	SRET	SRET	Subroutine Return	✓	✓	✓	✓	✓	✓	✓	Section 8.3
03	IRET	IRET	Interrupt Return	✓	✓	✓	✓	✓	✓	✓	Section 8.4
04	EI	EI	Enable Interrupt	✓	✓	✓	✓	✓	✓	✓	Section 8.5
05	DI	DI	Disable Interrupt	✓	✓	✓	✓	✓	✓	✓	Section 8.6
06	FEND	FEND	Main Routine Program End	✓	✓	✓	✓	✓	✓	✓	Section 8.7
07	WDT	WDT	Watchdog Timer Refresh	✓	✓	✓	✓	✓	✓	✓	Section 8.8
08	FOR	FOR | S	Start a FOR/NEXT Loop	✓	✓	✓	✓	✓	✓	✓	Section 8.9
09	NEXT	NEXT	End a FOR/NEXT Loop	✓	✓	✓	✓	✓	✓	✓	Section 8.10
Move and Compare											
10	CMP	CMP | S1 | S2 | D	Compare	✓	✓	✓	✓	✓	✓	✓	Section 9.1

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	Applicable PLC							Reference
				FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC	
11	ZCP	ZCP S1 S2 S D	Zone Compare	✓	✓	✓	✓	✓	✓	✓	Section 9.2
12	MOV	MOV S D	Move	✓	✓	✓	✓	✓	✓	✓	Section 9.3
13	SMOV	SMOV S m1 m2 D n	Shift Move	✓	✓	–	–	✓	–	✓	Section 9.4
14	CML	CML S D	Complement	✓	✓	–	–	✓	–	✓	Section 9.5
15	BMOV	BMOV S D n	Block Move	✓	✓	✓	✓	✓	✓	✓	Section 9.6

1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	Applicable PLC							Reference
				FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC	
Move and Compare											
16	FMOV	FMOV S D n	Fill Move	✓	✓	–	–	✓	–	✓	Section 9.7
17	XCH	XCH D1 D2	Exchange	✓	✓	–	–	✓	–	✓	Section 9.8
18	BCD	BCD S D	Conversion to Binary Coded Decimal	✓	✓	✓	✓	✓	✓	✓	Section 9.9
19	BIN	BIN S D	Conversion to Binary	✓	✓	✓	✓	✓	✓	✓	Section 9.10
Arithmetic and Logical Operation (+, –, ×, ÷)											
20	ADD	ADD S1 S2 D	Addition	✓	✓	✓	✓	✓	✓	✓	Section 10.1
21	SUB	SUB S1 S2 D	Subtraction	✓	✓	✓	✓	✓	✓	✓	Section 10.2
22	MUL	MUL S1 S2 D	Multiplication	✓	✓	✓	✓	✓	✓	✓	Section 10.3
23	DIV	DIV S1 S2 D	Division	✓	✓	✓	✓	✓	✓	✓	Section 10.4
24	INC	INC D	Increment	✓	✓	✓	✓	✓	✓	✓	Section 10.5
25	DEC	DEC D	Decrement	✓	✓	✓	✓	✓	✓	✓	Section 10.6
26	WAND	WAND S1 S2 D	Logical Word AND	✓	✓	✓	✓	✓	✓	✓	Section 10.7
27	WOR	WOR S1 S2 D	Logical Word OR	✓	✓	✓	✓	✓	✓	✓	Section 10.8
28	WXOR	WXOR S1 S2 D	Logical Exclusive OR	✓	✓	✓	✓	✓	✓	✓	Section 10.9
29	NEG	NEG D	Negation	✓	✓	–	–	✓	–	✓	Section 10.10
Rotation and Shift Operation											
30	ROR	ROR D n	Rotation Right	✓	✓	–	–	✓	–	✓	Section 11.1
31	ROL	ROL D n	Rotation Left	✓	✓	–	–	✓	–	✓	Section 11.2
32	RCR	RCR D n	Rotation Right with Carry	✓	✓	–	–	✓	–	✓	Section 11.3
33	RCL	RCL D n	Rotation Left with Carry	✓	✓	–	–	✓	–	✓	Section 11.4
34	SFTR	SFTR S D n1 n2	Bit Shift Right	✓	✓	✓	✓	✓	✓	✓	Section 11.5

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	Applicable PLC FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC	Reference
Rotation and Shift Operation											
35	SFTL	SFTL S D n1 n2	Bit Shift Left	✓	✓	✓	✓	✓	✓	✓	Section 11.6
36	WSFR	WSFR S D n1 n2	Word Shift Right	✓	✓	–	–	✓	–	✓	Section 11.7
37	WSFL	WSFL S D n1 n2	Word Shift Left	✓	✓	–	–	✓	–	✓	Section 11.8
38	SFWR	SFWR S D n	Shift write [FIFO/FILO control]	✓	✓	✓	✓	✓	✓	✓	Section 11.9
39	SFRD	SFRD S D n	Shift Read [FIFO Control]	✓	✓	✓	✓	✓	✓	✓	Section 11.10
Data Operation											
40	ZRST	ZRST D1 D2	Zone Reset	✓	✓	✓	✓	✓	✓	✓	Section 12.1
41	DECO	DECO S D n	Decode	✓	✓	✓	✓	✓	✓	✓	Section 12.2
42	ENCO	ENCO S D n	Encode	✓	✓	✓	✓	✓	✓	✓	Section 12.3
43	SUM	SUM S D	Sum of Active Bits	✓	✓	–	–	✓	–	✓	Section 12.4
44	BON	BON S D n	Check Specified Bit Status	✓	✓	–	–	✓	–	✓	Section 12.5
45	MEAN	MEAN S D n	Mean	✓	✓	–	–	✓	–	✓	Section 12.6
46	ANS	ANS S m D	Timed Annunciator Set	✓	✓	–	–	✓	–	✓	Section 12.7
47	ANR	ANR	Annunciator Reset	✓	✓	–	–	✓	–	✓	Section 12.8
48	SQR	SQR S D	Square Root	✓	✓	–	–	✓	–	✓	Section 12.9
49	FLT	FLT S D	Conversion to Floating Point	✓	✓	–	–	✓	–	✓	Section 12.10
High Speed Processing											
50	REF	REF D n	Refresh	✓	✓	✓	✓	✓	✓	✓	Section 13.1
51	REFF	REFF n	Refresh and Filter Adjust	✓	✓	–	–	✓	–	✓	Section 13.2
52	MTR	MTR S D1 D2 n	Input Matrix	✓	✓	✓	✓	✓	✓	✓	Section 13.3
53	HSCS	HSCS S1 S2 D	High Speed Counter Set	✓	✓	✓	✓	✓	✓	✓	Section 13.4

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	Applicable PLC FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC	Reference
High Speed Processing											
54	HSCR	HSCR S1 S2 D	High Speed Counter Reset	✓	✓	✓	✓	✓	✓	✓	Section 13.5
55	HSZ	HSZ S1 S2 S D	High Speed Counter Zone Compare	✓	✓	–	–	✓	–	✓	Section 13.6
56	SPD	SPD S1 S2 D	Speed Detection	✓	✓	✓	✓	✓	✓	✓	Section 13.7
57	PLSY	PLSY S1 S2 D	Pulse Y Output	✓	✓	✓	✓	✓	✓	✓	Section 13.8
58	PWM	PWM S1 S2 D	Pulse Width Modulation	✓	✓	✓	✓	✓	✓	✓	Section 13.9
59	PLSR	PLSR S1 S2 S3 D	Acceleration/Deceleration Setup	✓	✓	✓	✓	✓	✓	✓	Section 13.10
Handy Instruction											
60	IST	IST S D1 D2	Initial State	✓	✓	✓	✓	✓	✓	✓	Section 14.1
61	SER	SER S1 S2 D n	Search a Data Stack	✓	✓	–	–	✓	–	✓	Section 14.2
62	ABSD	ABSD S1 S2 D n	Absolute Drum Sequencer	✓	✓	✓	✓	✓	✓	✓	Section 14.3
63	INCD	INCD S1 S2 D n	Incremental Drum Sequencer	✓	✓	✓	✓	✓	✓	✓	Section 14.4
64	TTMR	TTMR D n	Teaching Timer	✓	✓	–	–	✓	–	✓	Section 14.5
65	STMR	STMR S m D	Special Timer	✓	✓	–	–	✓	–	✓	Section 14.6
66	ALT	ALT D	Alternate State	✓	✓	✓	✓	✓	✓	✓	Section 14.7
67	RAMP	RAMP S1 S2 D n	Ramp Variable Value	✓	✓	✓	✓	✓	✓	✓	Section 14.8
68	ROTC	ROTC S m1 m2 D	Rotary Table Control	✓	✓	–	–	✓	–	✓	Section 14.9
69	SORT	SORT S m1 m2 D n	SORT Tabulated Data	✓	✓	–	–	✓	–	✓	Section 14.10
External FX I/O Device											
70	TKY	TKY S D1 D2	Ten Key Input	✓	✓	–	–	✓	–	✓	Section 15.1
71	HKY	HKY S D1 D2 D3	Hexadecimal Input	✓	✓	–	–	✓	–	✓	Section 15.2
72	DSW	DSW S D1 D2 n	Digital Switch (Thumbwheel Input)	✓	✓	✓	✓	✓	✓	✓	Section 15.3

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	Applicable PLC: FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC	Reference
External FX I/O Device											
73	SEGD	SEGD S D	Seven Segment Decoder	✓	✓	–	–	✓	–	✓	Section 15.4
74	SEGL	SEGL S D n	Seven Segment With Latch	✓	✓	✓	✓	✓	✓	✓	Section 15.5
75	ARWS	ARWS S D1 D2 n	Arrow Switch	✓	✓	–	–	✓	–	✓	Section 15.6
76	ASC	ASC S D	ASCII Code Data Input	✓	✓	–	–	✓	–	✓	Section 15.7
77	PR	PR S D	Print (ASCII Code)	✓	✓	–	–	✓	–	✓	Section 15.8
78	FROM	FROM m1 m2 D n	Read From A Special Function Block	✓	✓	–	✓	✓	✓	✓	Section 15.9
79	TO	TO m1 m2 S n	Write To A Special Function Block	✓	✓	–	✓	✓	✓	✓	Section 15.10
External FX Device											
80	RS	RS S m D n	Serial Communication	✓	✓	✓	✓	✓	✓	✓	Section 16.1
81	PRUN	PRUN S D	Parallel Run (Octal Mode)	✓	✓	✓	✓	✓	✓	✓	Section 16.2
82	ASCI	ASCI S D n	Hexadecimal to ASCII Conversion	✓	✓	✓	✓	✓	✓	✓	Section 16.3
83	HEX	HEX S D n	ASCII to Hexadecimal Conversion	✓	✓	✓	✓	✓	✓	✓	Section 16.4
84	CCD	CCD S D n	Check Code	✓	✓	✓	✓	✓	✓	✓	Section 16.5
85	VRRD	VRRD S D	Volume Read	–	–	✓	✓	✓	✓	✓	–
86	VRSC	VRSC S D	Volume Scale	–	–	✓	✓	✓	✓	✓	–
87	RS2	RS2 S m D n n1	Serial Communication 2	✓	✓	–	–	–	–	–	Section 16.6
88	PID	PID S1 S2 S3 D	PID Control Loop	✓	✓	✓	✓	✓	✓	✓	Section 16.7
89 to 99	–										–
Data Transfer 2											
100, 101	–										–
102	ZPUSH	ZPUSH D	Batch Store of Index Register	✓	*5	–	–	–	–	–	Section 17.1

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	Applicable PLC							Reference
				FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC	
Data Transfer 2											
103	ZPOP	ZPOP D	Batch POP of Index Register	✓	*5	–	–	–	–	–	Section 17.2
104 to 109	–										–
Floating Point											
110	ECMP	ECMP S1 S2 D	Floating Point Compare	✓	✓	–	–	✓	–	✓	Section 18.1
111	EZCP	EZCP S1 S2 S D	Floating Point Zone Compare	✓	✓	–	–	✓	–	✓	Section 18.2
112	EMOV	EMOV S D	Floating Point Move	✓	✓	–	–	–	–	–	Section 18.3
113 to 115	–										–
116	ESTR	ESTR S1 S2 D	Floating Point to Character String Conversion	✓	✓	–	–	–	–	–	Section 18.4
117	EVAL	EVAL S D	Character String to Floating Point Conversion	✓	✓	–	–	–	–	–	Section 18.5
118	EBCD	EBCD S D	Floating Point to Scientific Notation Conversion	✓	✓	–	–	✓	–	✓	Section 18.6
119	EBIN	EBIN S D	Scientific Notation to Floating Point Conversion	✓	✓	–	–	✓	–	✓	Section 18.7
120	EADD	EADD S1 S2 D	Floating Point Addition	✓	✓	–	–	✓	–	✓	Section 18.8
121	ESUB	ESUB S1 S2 D	Floating Point Subtraction	✓	✓	–	–	✓	–	✓	Section 18.9
122	EMUL	EMUL S1 S2 D	Floating Point Multiplication	✓	✓	–	–	✓	–	✓	Section 18.10
123	EDIV	EDIV S1 S2 D	Floating Point Division	✓	✓	–	–	✓	–	✓	Section 18.11
124	EXP	EXP S D	Floating Point Exponent	✓	✓	–	–	–	–	–	Section 18.12
125	LOGE	LOGE S D	Floating Point Natural Logarithm	✓	✓	–	–	–	–	–	Section 18.13
126	LOG10	LOG10 S D	Floating Point Common Logarithm	✓	✓	–	–	–	–	–	Section 18.14
127	ESQR	ESQR S D	Floating Point Square Root	✓	✓	–	–	✓	–	✓	Section 18.15
128	ENEG	ENEG D	Floating Point Negation	✓	✓	–	–	–	–	–	Section 18.16
129	INT	INT S D	Floating Point to Integer Conversion	✓	✓	–	–	✓	–	✓	Section 18.17

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	Applicable PLC							Reference
				FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC	
Floating Point											
130	SIN	SIN S D	Floating Point Sine	✓	✓	–	–	✓	–	✓	Section 18.18
131	COS	COS S D	Floating Point Cosine	✓	✓	–	–	✓	–	✓	Section 18.19
132	TAN	TAN S D	Floating Point Tangent	✓	✓	–	–	✓	–	✓	Section 18.20
133	ASIN	ASIN S D	Floating Point Arc Sine	✓	✓	–	–	–	–	–	Section 18.21
134	ACOS	ACOS S D	Floating Point Arc Cosine	✓	✓	–	–	–	–	–	Section 18.22
135	ATAN	ATAN S D	Floating Point Arc Tangent	✓	✓	–	–	–	–	–	Section 18.23
136	RAD	RAD S D	Floating Point Degrees to Radians Conversion	✓	✓	–	–	–	–	–	Section 18.24
137	DEG	DEG S D	Floating Point Radians to Degrees Conversion	✓	✓	–	–	–	–	–	Section 18.25
138, 139	–										–
Data Operation 2											
140	WSUM	WSUM S D n	Sum of Word Data	✓	*5	–	–	–	–	–	Section 19.1
141	WTOB	WTOB S D n	WORD to BYTE	✓	*5	–	–	–	–	–	Section 19.2
142	BTOW	BTOW S D n	BYTE to WORD	✓	*5	–	–	–	–	–	Section 19.3
143	UNI	UNI S D n	4-bit Linking of Word Data	✓	*5	–	–	–	–	–	Section 19.4
144	DIS	DIS S D n	4-bit Grouping of Word Data	✓	*5	–	–	–	–	–	Section 19.5
145, 146	–										–
147	SWAP	SWAP S	Byte Swap	✓	✓	–	–	✓	–	✓	Section 19.6
148	–										–
149	SORT2	SORT2 S m1 m2 D n	Sort Tabulated Data 2	✓	*5	–	–	–	–	–	Section 19.7

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	Applicable PLC FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC	Reference
Positioning Control											
150	DSZR	DSZR S1 S2 D1 D2	DOG Search Zero Return	✓	*4	–	–	–	–	–	Section 20.1
151	DVIT	DVIT S1 S2 D1 D2	Interrupt Positioning	✓	*2, 4	–	–	–	–	–	Section 20.2
152	TBL	TBL D n	Batch Data Positioning Mode	✓	*5	–	–	–	–	–	Section 20.3
153, 154	–										–
155	ABS	ABS S D1 D2	Absolute Current Value Read	✓	✓	✓	✓	*1	✓	*1	Section 20.4
156	ZRN	ZRN S1 S2 S3 D	Zero Return	✓	*4	✓	✓	–	✓	–	Section 20.5
157	PLSV	PLSV S D1 D2	Variable Speed Pulse Output	✓	✓	✓	✓	–	✓	–	Section 20.6
158	DRVI	DRVI S1 S2 D1 D2	Drive to Increment	✓	✓	✓	✓	–	✓	–	Section 20.7
159	DRVA	DRVA S1 S2 D1 D2	Drive to Absolute	✓	✓	✓	✓	–	✓	–	Section 20.8
Real Time Clock Control											
160	TCMP	TCMP S1 S2 S3 S D	RTC Data Compare	✓	✓	✓	✓	✓	✓	✓	Section 21.1
161	TZCP	TZCP S1 S2 S D	RTC Data Zone Compare	✓	✓	✓	✓	✓	✓	✓	Section 21.2
162	TADD	TADD S1 S2 D	RTC Data Addition	✓	✓	✓	✓	✓	✓	✓	Section 21.3
163	TSUB	TSUB S1 S2 D	RTC Data Subtraction	✓	✓	✓	✓	✓	✓	✓	Section 21.4
164	HTOS	HTOS S D	Hour to Second Conversion	✓	✓	–	–	–	–	–	Section 21.5
165	STOH	STOH S D	Second to Hour Conversion	✓	✓	–	–	–	–	–	Section 21.6
166	TRD	TRD D	Read RTC data	✓	✓	✓	✓	✓	✓	✓	Section 21.7
167	TWR	TWR S	Set RTC data	✓	✓	✓	✓	✓	✓	✓	Section 21.8
168	–										–
169	HOUR	HOUR S D1 D2	Hour Meter	✓	✓	✓	✓	*1	✓	*1	Section 21.9

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	Applicable PLC							Reference
				FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC	
External Device											
170	GRY	GRY S D	Decimal to Gray Code Conversion	✓	✓	–	–	✓	–	✓	Section 22.1
171	GBIN	GBIN S D	Gray Code to Decimal Conversion	✓	✓	–	–	✓	–	✓	Section 22.2
172 to 175	–										–
176	RD3A	RD3A m1 m2 D	Read form Dedicated Analog Block	✓	✓	–	✓	*1	✓	*1	Section 22.3
177	WR3A	WR3A m1 m2 S	Write to Dedicated Analog Block	✓	✓	–	✓	*1	✓	*1	Section 22.4
178, 179	–										–
Extension Function											
180	EXTR	EXTR S SD1 SD2 SD3	External ROM Function (FX2N/FX2NC)	–	–	–	–	*1	–	*1	–
Others											
181	–										–
182	COMRD	COMRD S D	Read Device Comment Data	✓	*5	–	–	–	–	–	Section 24.1
183	–										–
184	RND	RND D	Random Number Generation	✓	✓	–	–	–	–	–	Section 24.2
185	–										–
186	DUTY	DUTY n1 n2 D	Timing Pulse Generation	✓	*5	–	–	–	–	–	Section 24.3
187	–										–
188	CRC	CRC S D n	Cyclic Redundancy Check	✓	✓	–	–	–	–	–	Section 24.4
189	HCMOV	HCMOV S D n	High Speed Counter Move	✓	*4	–	–	–	–	–	Section 24.5
Block Data Operation											
190, 191	–										–
192	BK+	BK+ S1 S2 D n	Block Data Addition	✓	*5	–	–	–	–	–	Section 25.1
193	BK-	BK- S1 S2 D n	Block Data Subtraction	✓	*5	–	–	–	–	–	Section 25.2

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	Applicable PLC FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC	Reference
Block Data Subtraction											
194	BKCMP=	BKCMP= S1 S2 D n	Block Data Compare S1 = S2	✓	*5	–	–	–	–	–	Section 25.3
195	BKCMP>	BKCMP> S1 S2 D n	Block Data Compare S1 > S2	✓	*5	–	–	–	–	–	Section 25.3
196	BKCMP<	BKCMP< S1 S2 D n	Block Data Compare S1 < S2	✓	*5	–	–	–	–	–	Section 25.3
197	BKCMP<>	BKCMP<> S1 S2 D n	Block Data Compare S1 ≠ S2	✓	*5	–	–	–	–	–	Section 25.3
198	BKCMP<=	BKCMP<= S1 S2 D n	Block Data Compare S1 ≤ S2	✓	*5	–	–	–	–	–	Section 25.3
199	BKCMP>=	BKCMP>= S1 S2 D n	Block Data Compare S1 ≥ S2	✓	*5	–	–	–	–	–	Section 25.3
Character String Control											
200	STR	STR S1 S2 D	BIN to Character String Conversion	✓	*5	–	–	–	–	–	Section 26.1
201	VAL	VAL S D1 D2	Character String to BIN Conversion	✓	*5	–	–	–	–	–	Section 26.2
202	$+	$+ S1 S2 D	Link Character Strings	✓	✓	–	–	–	–	–	Section 26.3
203	LEN	LEN S D	Character String Length Detection	✓	✓	–	–	–	–	–	Section 26.4
204	RIGHT	RIGHT S D n	Extracting Character String Data from the Right	✓	✓	–	–	–	–	–	Section 26.5
205	LEFT	LEFT S D n	Extracting Character String Data from the Left	✓	✓	–	–	–	–	–	Section 26.6
206	MIDR	MIDR S1 D S2	Random Selection of Character Strings	✓	✓	–	–	–	–	–	Section 26.7
207	MIDW	MIDW S1 D S2	Random Replacement of Character Strings	✓	✓	–	–	–	–	–	Section 26.8
208	INSTR	INSTR S1 S2 D n	Character string search	✓	*5	–	–	–	–	–	Section 26.9
209	$MOV	$MOV S D	Character String Transfer	✓	✓	–	–	–	–	–	Section 26.10
Data Operation 3											
210	FDEL	FDEL S D n	Deleting Data from Tables	✓	*5	–	–	–	–	–	Section 27.1
211	FINS	FINS S D n	Inserting Data to Tables	✓	*5	–	–	–	–	–	Section 27.2
212	POP	POP S D n	Shift Last Data Read [FILO Control]	✓	✓	–	–	–	–	–	Section 27.3

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	Applicable PLC FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC	Reference
Data Operation 3											
213	SFR	SFR D n	Bit Shift Right with Carry	✓	✓	–	–	–	–	–	Section 27.4
214	SFL	SFL D n	Bit Shift Left with Carry	✓	✓	–	–	–	–	–	Section 27.5
215 to 219	–										–
Data Comparison											
220 to 223	–										–
224	LD=	LD= S1 S2	Load Compare S1 = S2	✓	✓	✓	✓	✓	✓	✓	Section 28.1
225	LD>	LD> S1 S2	Load Compare S1 > S2	✓	✓	✓	✓	✓	✓	✓	Section 28.1
226	LD<	LD< S1 S2	Load Compare S1 < S2	✓	✓	✓	✓	✓	✓	✓	Section 28.1
227	–										–
228	LD<>	LD<> S1 S2	Load Compare S1 ≠ S2	✓	✓	✓	✓	✓	✓	✓	Section 28.1
229	LD<=	LD<= S1 S2	Load Compare S1 ≤ S2	✓	✓	✓	✓	✓	✓	✓	Section 28.1
230	LD>=	LD>= S1 S2	Load Compare S1 ≥ S2	✓	✓	✓	✓	✓	✓	✓	Section 28.1
231	–										–
232	AND=	AND= S1 S2	AND Compare S1 = S2	✓	✓	✓	✓	✓	✓	✓	Section 28.2
233	AND>	AND> S1 S2	AND Compare S1 > S2	✓	✓	✓	✓	✓	✓	✓	Section 28.2
234	AND<	AND< S1 S2	AND Compare S1 < S2	✓	✓	✓	✓	✓	✓	✓	Section 28.2
235	–										–
236	AND<>	AND<> S1 S2	AND Compare S1 ≠ S2	✓	✓	✓	✓	✓	✓	✓	Section 28.2
237	AND<=	AND<= S1 S2	AND Compare S1 ≤ S2	✓	✓	✓	✓	✓	✓	✓	Section 28.2
238	AND>=	AND>= S1 S2	AND Compare S1 ≥ S2	✓	✓	✓	✓	✓	✓	✓	Section 28.2
239	–										–

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	Applicable PLC							Reference
				FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC	
Data Comparison											
240	OR=	OR= S1 S2	OR Compare S1 = S2	✓	✓	✓	✓	✓	✓	✓	Section 28.3
241	OR>	OR> S1 S2	OR Compare S1 > S2	✓	✓	✓	✓	✓	✓	✓	Section 28.3
242	OR<	OR< S1 S2	OR Compare S1 < S2	✓	✓	✓	✓	✓	✓	✓	Section 28.3
243	–										–
244	OR<>	OR<> S1 S2	OR Compare S1 ≠ S2	✓	✓	✓	✓	✓	✓	✓	Section 28.3
245	OR<=	OR<= S1 S2	OR Compare S1 ≤ S2	✓	✓	✓	✓	✓	✓	✓	Section 28.3
246	OR>=	OR>= S1 S2	OR Compare S1 ≥ S2	✓	✓	✓	✓	✓	✓	✓	Section 28.3
247 to 249	–										–
Data Table Operation											
250 to 255	–										–
256	LIMIT	LIMIT S1 S2 S3 D	Limit Control	✓	✓	–	–	–	–	–	Section 29.1
257	BAND	BAND S1 S2 S3 D	Dead Band Control	✓	✓	–	–	–	–	–	Section 29.2
258	ZONE	ZONE S1 S2 S3 D	Zone Control	✓	✓	–	–	–	–	–	Section 29.3
259	SCL	SCL S1 S2 D	Scaling (Coordinate by Point Data)	✓	✓	–	–	–	–	–	Section 29.4
260	DABIN	DABIN S D	Decimal ASCII to BIN Conversion	✓	*5	–	–	–	–	–	Section 29.5
261	BINDA	BINDA S D	BIN to Decimal ASCII Conversion	✓	*5	–	–	–	–	–	Section 29.6
262 to 268	–										–
269	SCL2	SCL2 S1 S2 D	Scaling 2 (Coordinate by X/Y Data)	✓	*3	–	–	–	–	–	Section 29.7

*1: The instruction is provided in the FX2N/FX2NC Series Ver.3.00 or later.
*2: The function is changed in the FX3UC Series Ver.1.30 or later.
*3: The instruction is provided in the FX3UC Series Ver.1.30 or later.
*4: The function is changed in the FX3UC Series Ver.2.20 or later.
*5: The instruction is provided in the FX3UC Series Ver.2.20 or later.

FNC No.	Mnemonic	Symbol	Function	Applicable PLC FX3U	FX3UC	FX1S	FX1N	FX2N	FX1NC	FX2NC	Reference
External Device Communication (Inverter Communication)											
270	IVCK	IVCK S1 S2 D n	Inverter Status Check	✓	✓	–	–	–	–	–	Section 30.1
271	IVDR	IVDR S1 S2 S3 n	Inverter Drive	✓	✓	–	–	–	–	–	Section 30.2
272	IVRD	IVRD S1 S2 D n	Inverter Parameter Read	✓	✓	–	–	–	–	–	Section 30.3
273	IVWR	IVWR S1 S2 S3 n	Inverter Parameter Write	✓	✓	–	–	–	–	–	Section 30.4
274	IVBWR	IVBWR S1 S2 S3 n	Inverter Parameter Block Write	✓	✓	–	–	–	–	–	Section 30.5
275 to 277	–										–
Data Transfer 3											
278	RBFM	RBFM m1 m2 D n1 n2	Divided BFM Read	✓	*5	–	–	–	–	–	Section 31.1
279	WBFM	WBFM m1 m2 S n1 n2	Divided BFM Write	✓	*5	–	–	–	–	–	Section 31.2
High Speed Processing 2											
280	HSCT	HSCT S1 m S2 D n	High Speed Counter Compare With Data Table	✓	✓	–	–	–	–	–	Section 32.1
281 to 289	–										–
Extension File Register Control											
290	LOADR	LOADR S n	Load From ER	✓	✓	–	–	–	–	–	Section 33.1
291	SAVER	SAVER S m D	Save to ER	✓	✓	–	–	–	–	–	Section 33.2
292	INITR	INITR S n	Initialize R and ER	✓	✓	–	–	–	–	–	Section 33.3
293	LOGR	LOGR S m D1 n D2	Logging R and ER	✓	✓	–	–	–	–	–	Section 33.4
294	RWER	RWER S n	Rewrite to ER	✓	*3	–	–	–	–	–	Section 33.5
295	INITER	INITER S n	Initialize ER	✓	*3	–	–	–	–	–	Section 33.6
296 to 299	–										–

4. Devices in Detail

This chapter explains how numeric values are handled in the PLC as well as the roles and functions of various built-in devices including I/O relays, auxiliary relays, state relays, counters and data registers.
The following content provides a basis for handling the PLC.

4.1 Device Number List

Device numbers are assigned as shown below.
For input relay numbers and output relay numbers when I/O extension equipment and special extension equipment are connected to the PLC main unit, refer to the FX3U Hardware Edition.

Device name	Description			Reference
I/O relay				
Input relay	X000 to X367*1	248 points	Device numbers are octal. The total number of inputs and outputs is 256.	Section 4.2
Output relay	Y000 to Y367*1	248 points		
Auxiliary relay				
General type [variable]	M0 to M499	500 points	The setting can be changed between the latched (battery backed) type and the non-latched type using parameters.	Section 4.3
Latched (battery backed) type [variable]	M500 to M1023	524 points		
Latched (battery backed) type [fixed]	M1024 to M7679	6656 points		
Special type*2	M8000 to M8511	512 points		Chapter 36
State relay				
Initial state (general type [variable])	S0 to S9	10 points	The setting can be changed between the latched (battery backed) type and the non-latched type using parameters.	Section 4.4
General type [variable]	S10 to S499	490 points		
Latched (battery backed) type [variable]	S500 to S899	400 points		
Annunciator (latched (battery backed) type [variable])	S900 to S999	100 points		
Latched (battery backed) type [fixed]	S1000 to S4095	3096 points		
Timer (on-delay timer)				
100 ms	T0 to T191	192 points	0.1 to 3,276.7 sec	Section 4.5
100 ms [for subroutine or interrupt routine]	T192 to T199	8 points	0.1 to 3,276.7 sec	
10 ms	T200 to T245	46 points	0.01 to 327.67 sec	
Retentive type for 1 ms	T246 to T249	4 points	0.001 to 32.767 sec	
Retentive type for 100 ms	T250 to T255	6 points	0.1 to 3,276.7 sec	
1 ms	T256 to T511	256 points	0.001 to 32.767 sec	

Device name	Description			Reference
Counter				
General type up counter (16 bits) [variable]	C0 to C99	100 points	Counts 0 to 32,767 The setting can be changed between the latched (battery backed) type and the non-latched type using parameters.	Section 4.6
Latched (battery backed) type up counter (16 bits) [variable]	C100 to C199	100 points		
General type bi-directional counter (32 bits) [variable]	C200 to C219	20 points	–2,147,483,648 to +2,147,483,647 counts The setting can be changed between the latched (battery backed) type and the non-latched type using parameters.	
Latched (battery backed) type bi-directional counter (32 bits) [variable]	C220 to C234	15 points		

*1. Available device numbers vary depending on the PLC. For details, refer to Section 4.2.

*2. For supported functions, refer to Chapter 36.
For handling the battery latched (battery backed) area, refer to Section 2.6.

Device name	Description			Reference
High speed counter				
1-phase 1-counting input Bi-directional (32 bits)	C235 to C245	8 points maximum can be used among C235 to C255 [latched (battery backed) type]. The setting can be changed between the latched (battery backed) type and the non-latch type using parameters. –2,147,483,648 to 2,147,483,647 counts Hardware counter*1 1 phase: 100 kHz × 6 points, 10 kHz × 2 points 2 phases: 50 kHz (1 edge count), 50 kHz (4 edge count) Software counter 1 phase: 40 kHz 2 phases: 40 kHz (1 edge count), 10 kHz (4 edge count)		Section 4.7
1-phase 2-counting input Bi-directional (32 bits)	C246 to C250			
2-phase 2-counting input Bi-directional (32 bits)	C251 to C255			
Data register (32 bits when used in pair form)				
General type (16 bits) [variable]	D0 to D199	200 points	The setting can be changed between the latched (battery backed) type and the non-latched type using parameters.	Section 4.8
latched (battery backed) type (16 bits) [variable]	D200 to D511	312 points		
latched (battery backed) type (16 bits) [fixed] <file register>	D512 to D7999 <D1000 to D7999>	7488 points <7000 points>	Among the 7488 fixed latched (battery backed) type data registers, D1000 and later can be set as file registers in units of 500 points.	
Special type (16 bits)*2	D8000 to D8511	512 points		Chapter 36
Index type (16 bits)	V0 to V7, Z0 to Z7	16 points		Section 4.10
Extension register/Extension file register				
Extension register (16 bits)	R0 to R32767	32768 points	latched (battery backed)	Section 4.9
Extension file register (16 bits)	ER0 to ER32767	32768 points	Available only while a memory cassette is mounted	
Pointer				
For jump and branch call	P0 to P4095	4096 points	For CJ and CALL instructions	Section 4.11
Input interrupt Input delay interrupt	I0□□ to I5□□	6 points		
Timer interrupt	I6□□ to I8□□	3 points		
Counter interrupt	I010 to I060	6 points	For HSCS instruction	
Nesting				
For master control	N0 to N7	8 points	For MC instruction	

Device name	Description		Reference
Constant			
Decimal (K)	16 bits	–32768 to +32767	Chapter 5
	32 bits	–2,147,483,648 to +2,147,483,647	
Hexadecimal (H)	16 bits	0 to FFFF	
	32 bits	0 to FFFFFFFF	
Real number (E)	32 bits	-1.0×2^{128} to -1.0×2^{-126}, 0, 1.0×2^{-126} to -1.0×2^{128} Both the decimal point expression and the exponent expression are available.	
Character string (" ")	Character string	Specify characters by quotation marks. In a constant of an instruction, up to 32 half-width characters are available.	

*1. When the FX3U-4HSX-ADP is connected to an FX3U PLC, the maximum input frequency is set as follows:
1 phase: 200 kHz
2 phases: 100 kHz (1 edge count). 100 kHz (4 edge count)

*2. For supported functions, refer to Chapter 36.
For handling of the latched (battery backed) area, refer to Section 2.6.

4.2 I/O Relays [X, Y]

Some numbers of input relays and output relays are specific ones secured in the main unit, and other numbers are assigned to extension devices in the connection order. Because these numbers are octal, numeric values such as "8" and "9" do not exist.

4.2.1 Numbers of I/O relays

The table below shows input relay (X) numbers and output relays (Y) numbers (Numbers are assigned in octagon.)

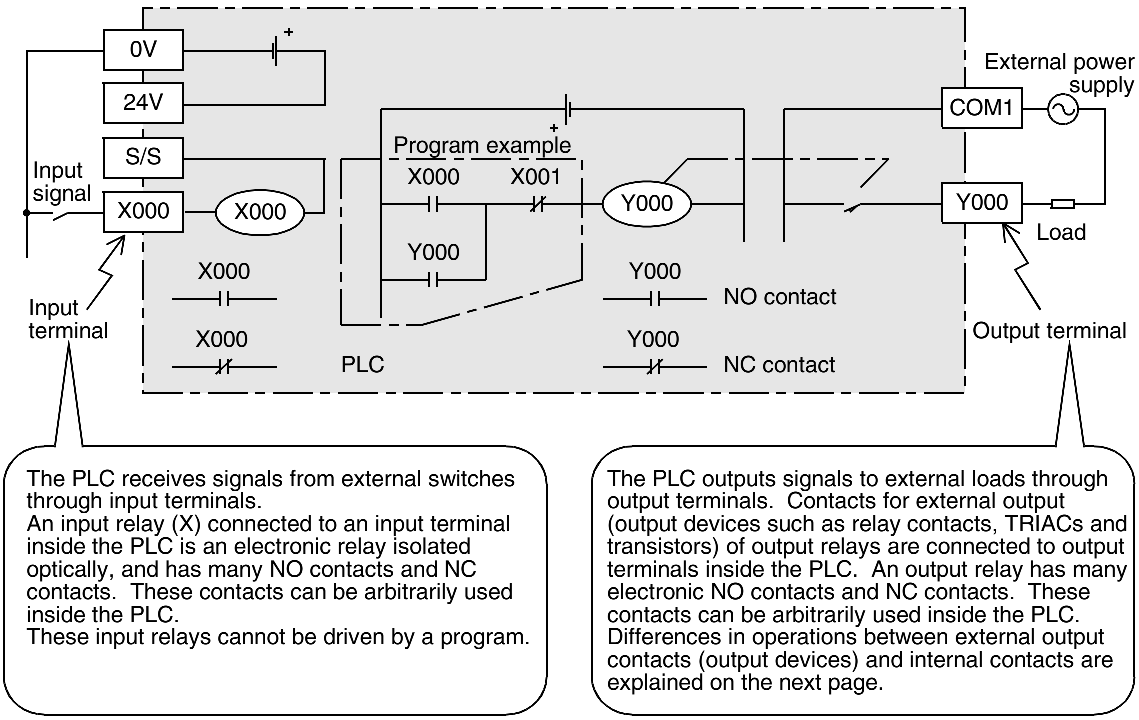
	Model name	FX3UC-32MT-LT	When extended	
FX3UC PLC	Input	X000 to X017 16 points	X000 to X357 240 points	256 points in total
	Output	Y000 to Y017 16 points	Y000 to Y357 240 points	

	Model name	FX3U-16M	FX3U-32M	FX3U-48M	FX3U-64M	FX3U-80M	FX3U-128M
FX3U PLC	Input	X000 to X007 8 points	X000 to X017 16 points	X000 to X027 24 points	X000 to X037 32 points	X000 to X047 40 points	X000 to X077 64 points
	Output	Y000 to Y007 8 points	Y000 to Y017 16 points	Y000 to Y027 24 points	Y000 to Y037 32 points	Y000 to Y047 40 points	Y000 to Y077 64 points

	Model name	When extended	
FX3U PLC	Input	X000 to X367 248 points	256 points in total
	Output	Y000 to Y367 248 points	

4.2.2 Functions and roles

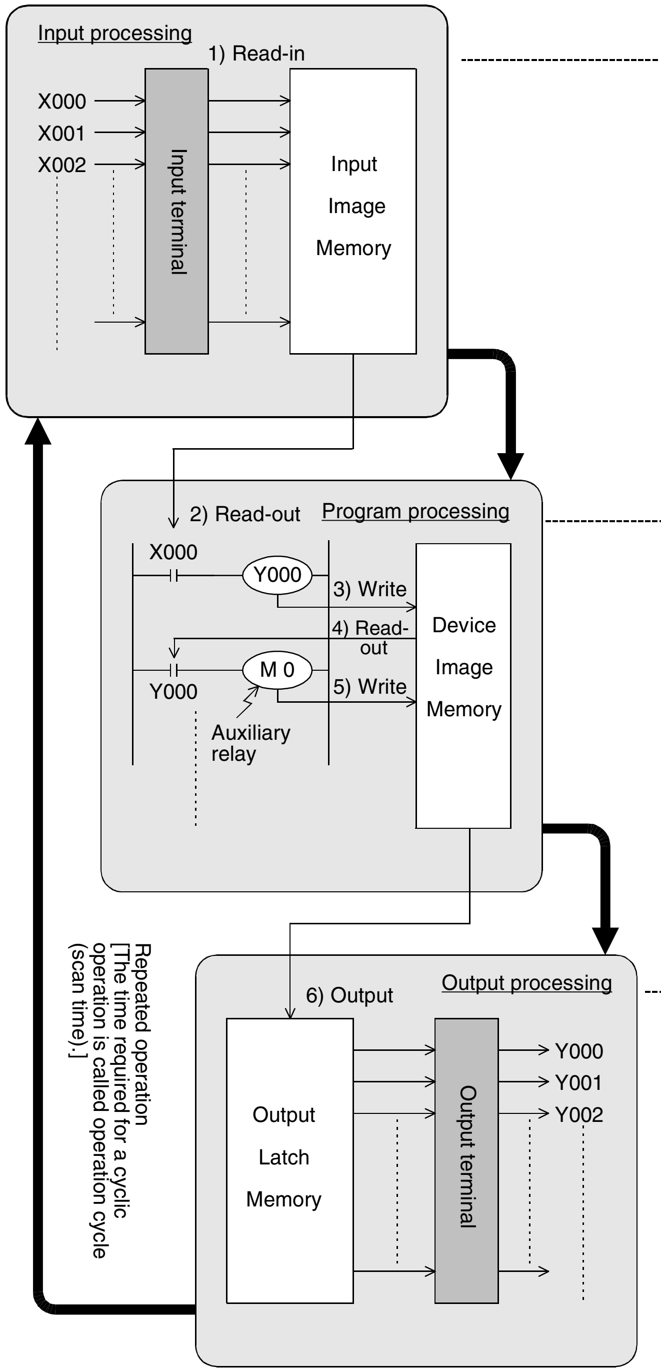
Examples of terminal names and wiring (sink input) are for the FX3U Series PLC.



4.2.3 Operation timing of input relays

The PLC executes sequence control by repeatedly executing the following processing procedure. In this batch I/O method, not only are there driving times of input filters and output devices but also response delays caused by operation cycles. (Refer to Section 6.3.)

1) 2) 3) .... 6) indicate the processing order.



Input processing
Before executing a program, the PLC reads the ON/OFF status of all input terminals inside the PLC into the input image memory.
Even if inputs change while the program is executed, the contents of the input image memory remain unchanged, but the changes in inputs are read during the input processing in the next cycle.
Even if an input contact changes from ON to OFF or from OFF to ON, its ON/OFF status is judged after the response delay (approximately 10 ms) caused by the input filter.
(When the input filter is a digital type input terminal , its value can be overwritten by a sequence program.)

Program processing
The PLC reads the ON/OFF status of each device from the input image memory and other device image memories according to the contents of instructions in the program memory, executes operations in sequence from step 0, and then writes the operation result to the image memory.
Accordingly, the contents of the image memory for each device change as the program is executed.
The operation of a contact inside an output relay is determined by the contents of the output image memory.

Output processing
When execution of all instructions is finished, the ON/OFF status of the image memory of outputs (Y) is transferred to the output latch memory. This is the actual output of the PLC.
External output contacts inside the PLC operate after the response delay time of the output devices.

The above method is called the batch I/O method (or refresh method).

4.3 Auxiliary Relay [M]

There are many auxiliary relays inside the PLC. Coils of auxiliary relays are driven by contacts of various devices inside the PLC in the same way as output relays.
Auxiliary relays have many electronically NO contacts and NC contacts which can be used arbitrarily inside the PLC. However, external loads cannot be driven directly by these contacts. External loads should be driven by output relays.

4.3.1 Numbers of auxiliary relays

The table below shows auxiliary relay (M) numbers. (Numbers are assigned in decimal.)

	General type	latched (battery backed) type	Fixed latched (battery backed) type	Special type
FX3U/FX3UC PLC	M0 to M499 500 points*1	M500 to M1023 524 points*2	M1024 to M7679 6656 points*3	M8000 to M8511 512 points

*1. This area is not latched (battery backed). It can be changed to a latched (battery backed) area by setting the parameters.

*2. This area is latched (battery backed). It can be changed to a non-latched (non-battery-backed) area by setting the parameters.

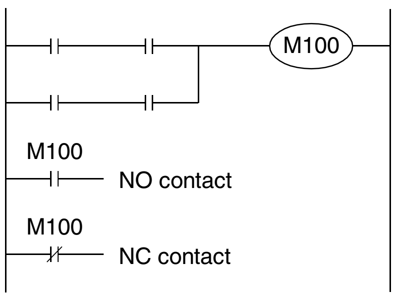
*3. The characteristics of latch (battery backup) cannot be changed in the parameters.

When simple N : N link or parallel link is used, some auxiliary relays are occupied for the link.

→ **Refer to the Communication Control Manual.**

4.3.2 Functions and operation examples

1. General type



Auxiliary relay circuit

All of general type auxiliary relays turn OFF when the PLC turns OFF.
When the ON/OFF status of auxiliary relays just before power failure is required in control, use latched (battery backed) type auxiliary relays.

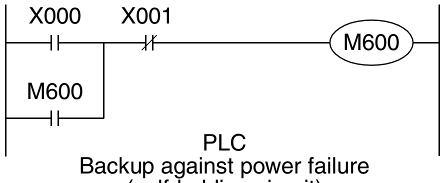
2. Latched (battery backed) type

When the power is turned OFF while the PLC is operating, all of the output relays and general type auxiliary relays turn OFF.

When restoring the power again, all of the output relays and general type auxiliary relays remain OFF except those whose input condition is ON. In some output relays and auxiliary relays, however, the ON/OFF status just before power failure should be stored and then replicated when restoring the power, depending on control targets. In such a case, use latched (battery backed) type auxiliary relays.

Latch type devices are backed up by the battery built in to the PLC.

→ **For details on backup method against power failure, refer to Section 2.6.**

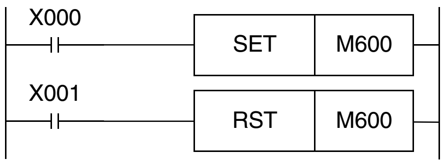


PLC
Backup against power failure
(self-holding circuit)

The figure on the left shows an operation example of M600 (latched [battery backed] type device) in a self-holding circuit.

When X000 turns ON and M600 turns ON in this circuit, M600 holds its operation by itself even if X000 is opened.

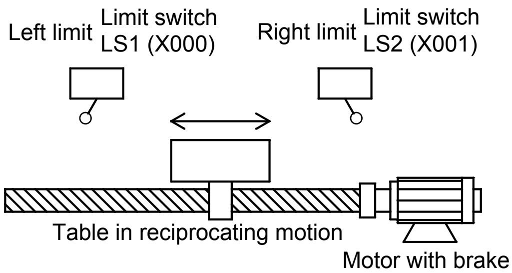
Because M600 is a latched (battery backed) type device, it remains activated when the operation is restarted even after X000 has turned OFF due to power failure. If an NC contact of X001 is opened when the operation is restarted, however, M600 is deactivated.



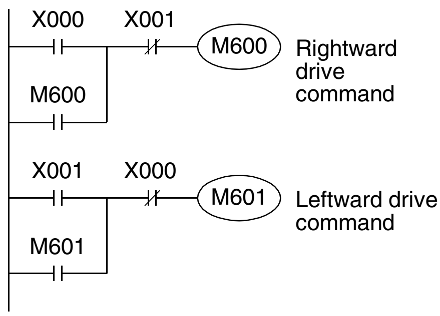
PLC
Backup against power failure
(set/reset circuit)

The figure on the left shows a circuit using the SET and RST instructions.

1) Application example of latched (battery backed) type auxiliary relays



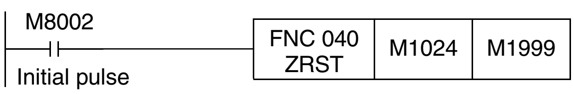
In some cases, the table should be restarted in the same direction as the direction selected just before power failure.



X000 = ON (at the left limit) → M600 = ON→ The table is driven rightward. → The power is turned OFF. → The table is stopped in an intermediate position. → The table is restarted (M600 = ON). → X001 = ON (at the right limit) → M600 = OFF, M601 = ON → The table is driven leftward.

2) Method for using a fixed latched (battery backed) type auxiliary relay as a general type auxiliary relay

When using a fixed latched (battery backed) type auxiliary relay as a general type auxiliary relay, provide a reset circuit shown in the figure below around the head step in the program.



M1024 to M1999 are initialized.

4.4 State Relay [S]

State relays (S) are important devices to program stepping type process control simply, and combined with the step ladder instruction STL.
State relays can be used in the SFC (sequential function chart) programming method.

→ **For programming by the step ladder instruction and SFC method, refer to Chapter 34.**

4.4.1 Numbers of state relays

The table below shows state relay (S) numbers. (Numbers are assigned in decimal.)

	General type	Latched (battery backed) type	Fixed latched (battery backed) type	Annunciator type
FX3U/FX3UC PLC	S0 to S499 500 points (S0 to S9 are provided initially.)*1	S500 to S899 400 points*2	S1000 to S4095 3096 points*3	S900 to S999 100 points*2

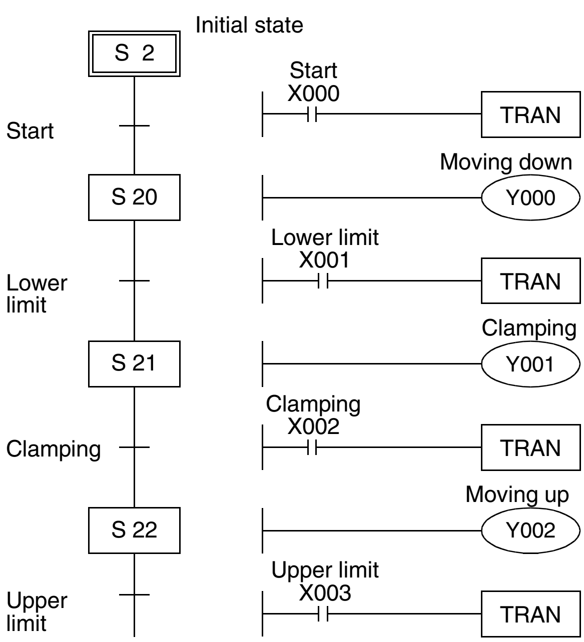
*1. This area is not latched (battery backed). It can be changed to a latched (battery backed) area by setting the parameters.

*2. This area is latched (battery backed). It can be changed to a non-latched (non-battery-backed) area by setting the parameters.

*3. The characteristics of latch (battery backup) cannot be changed in the parameters.

4.4.2 Functions and operation examples

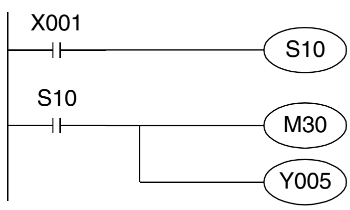
1. General type



In the stepping type process control shown in the left figure, when the start signal X000 turns ON, the state relay S20 is set (turned ON) and the solenoid valve Y000 for moving down turns on.
When the lower limit switch X001 turns ON the state relay S21 is set (turned ON) and the solenoid valve Y001 for clamping turns on.
When the clamp confirmation limit switch X002 turns ON, the state relay S22 is set (turned ON).
When the operation proceeds to the next step, the state relay in the preceding step is automatically reset (turned OFF).

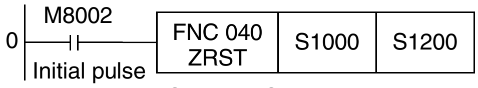
When the PLC turns OFF, all of general type state relays are turned OFF.
When the ON/OFF status just before power failure is required, use latched (battery backed) type state relays.

State relays have many NO contacts and NC contacts in the same way as auxiliary relays, and such contacts can be used arbitrarily in sequence programs.
When state relays (S) are not used for step ladder instructions, they can be used in general sequences in the same way as auxiliary relays (M) (as shown in the figure on the right).



2. Latched (battery backed) type

- Latched (battery backed) type state relays store their ON/OFF status even if the power is shut down while the PLC is operating, so the operation can be restarted from the last point in the process.
  Latched (battery backed) type state relays are backed up by the battery built into the PLC.
  → **For details on each backup method, refer to Section 2.6.**
- When using latched (battery backed) type state relays as general type state relays, provide a reset circuit shown in the right figure around the head step in the program.

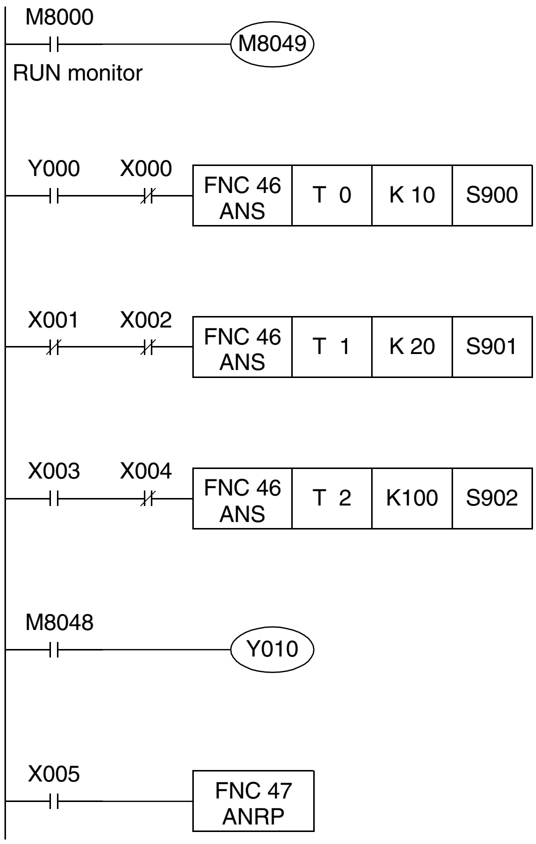


S1000 to S1200 are initialized.

3. Annunciator type

Annunciator type state relays can be used as outputs for external fault diagnosis.

For example, when an external fault diagnosis circuit shown in the figure below is created and the contents of the special data register D8049 are monitored, the smallest number out of the active state relays S900 to S999 is stored in D8049.
If two or more faults have occurred, the smallest state number having a fault is displayed at first. When the fault is cleared, the next smallest state number having a fault is stored.



- When the special auxiliary relay M8049 is driven, monitoring becomes valid.
- If the forward end detection input X000 is not activated within 1 second after the forward output Y000 is driven, S900 is activated.
- If both the upper limit detection input X001 and the lower limit detection input X002 are deactivated at the same time for 2 seconds or more, S901 is activated.
- In a machine whose tact time is less than 10 seconds, if the switch X004 which is designed to be activated during one-cycle operation of the machine is not activated while the continuous operation mode input X003 is ON, S902 is activated.
- When any annunciator among S900 to S999 turns ON, the special auxiliary relay M8048 is activated and the fault display output Y010 is activated.
- The state relays activated by the external fault diagnosis program can be turned OFF by the reset button X005.
  Every time X005 is set to ON, the active annunciator with the smallest number is reset in turn.

While the special auxiliary relay M8049 is not driven, annunciator type state relays can be used as latched (battery backed) type state relays in sequence programs in the same way as general type state relays.
In the SFC programming mode in the FX-PCS/WIN(-E), however, S900 to S999 cannot be programmed as a processes flow in SFC diagrams.

4.5 Timer [T]

Timers add and count clock pulses of 1 ms, 10 ms, 100 ms, etc. inside the PLC. When the counted value reaches a specified set value, the output contact of the timer turns on.
A set value can be directly specified by a constant (K) in the program memory, or indirectly specified by the contents of a data register (D).

4.5.1 Numbers of timers

The table below shows timer (T) numbers. (The numbers are assigned in decimal.)

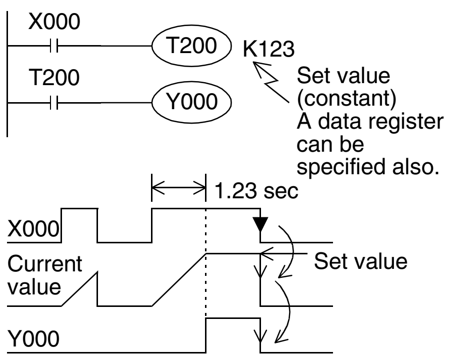
	For 100 ms pulses 0.1 to 3276.7 sec	For 10 ms pulses 0.01 to 327.67 sec	Retentive type for 1 ms pulses*1 0.001 to 32.767 sec	Retentive type for 100 ms pulses*1 0.1 to 3276.7 sec	For 1 ms pulses 0.001 to 32.767 sec
FX3U/ FX3UC PLC	T 0 to T199 200 points ------------- Routine program type T192 to T199	T200 to T245 46 points	T246 to T249 4 points for Interrupt execution Latched (battery backed) type*1	T250 to T255 6 points Latched (battery backed) type*1	T256 to T511 256 points

Timer numbers not used for timers can be used as data registers for storing numeric values.

*1. Retentive timers are backed up by battery.

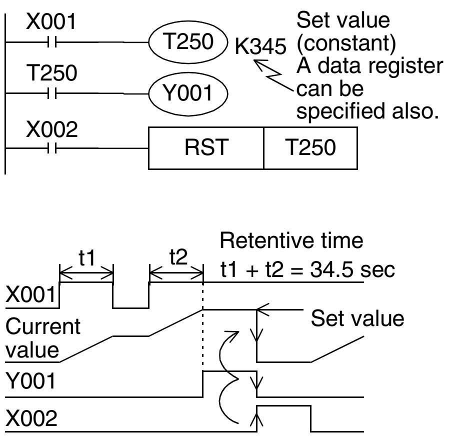
4.5.2 Functions and operation examples

General type



When the drive input X000 of the timer coil T200 turns ON, the current value counter for T200 adds and counts clock pulses of 10 ms. When the counted value becomes equivalent to the set value K123, the output contact of the timer turns on.
In other words, the output contact turns on 1.23 seconds after the coil is driven.
When the drive input X000 turns OFF or when the power is turned off the timer is reset and the output contact returns.

Retentive type

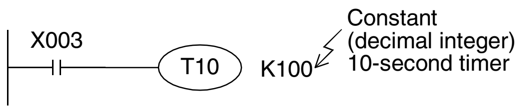


When the drive input X001 of the timer coil T250 turns ON, the current value counter for T250 adds and counts clock pulses of 100 ms. When the counted value becomes equivalent to the set value K345, the output contact of the timer turns on.
Even if the drive input X001 turns OFF or the power is turned off during counting, the timer continues counting when the operation restarts. The retentive operating time is 34.5 seconds.
When the reset input X002 turns ON, the timer is reset and the output contact is returned.

1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

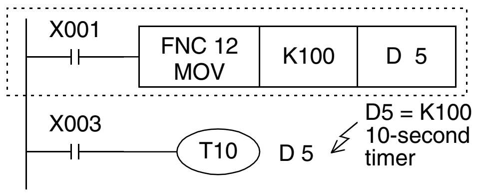
4.5.3 Set value specification method

1. Specifying a constant (K)



T10 is a 100 ms (0.1 sec) type timer.
When the constant "100" is specified, T10 works as a 10-second timer (0.1 sec × 100 = 10 sec).

2. Indirectly specifying a data register



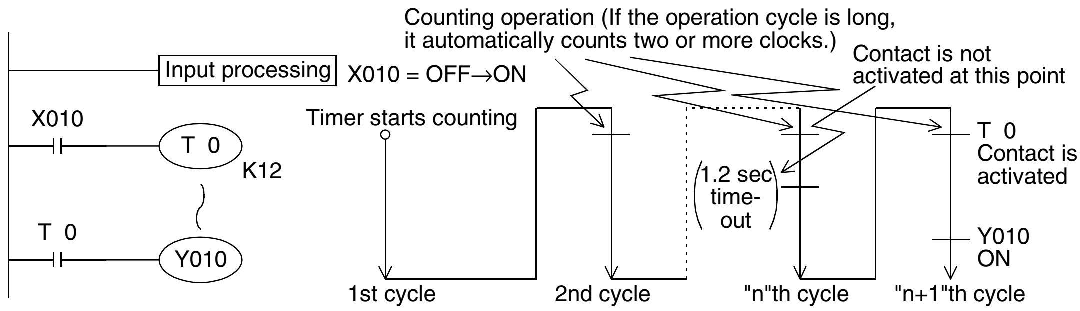
Turns on when T10 reaches the indirectly specified value of the defined data register, previously set by a digital switch.
Note that the set value of a latched (battery backed) type register can change randomly indefinite when the battery voltage is low.

4.5.4 Cautions on routines

1) Use timers T192 to T199 in subroutines and interrupt routines. These timers execute counting when a coil instruction or END instruction is executed.
When such a timer reaches the set value, its output contact turns on when a coil instruction or END instruction is executed.
Because general type timers execute counting only when a coil instruction is executed (Refer to "4.5.5 Details of timer operation and timer accuracy" below.), they do not execute counting and do not operate normally if they are used in subroutines or interrupt routines in which a coil instruction is executed only in a certain condition.
2) When a retentive timer for 1 ms pulses (T246 to T249) is used in a subroutine or interrupt routine, note that its output contact turns on when the first coil instruction is executed after the retentive timer has reached the set value.

4.5.5 Details of timer operation and timer accuracy

A timer (except interrupt execution type) starts counting when a coil is driven, and its output contact turns on when the first coil instruction is executed after the timer has reached timeout.



As shown in the above operation diagram, the accuracy of operation of the timer contact after the coil is driven until the contact turns on is shown in the following outline:

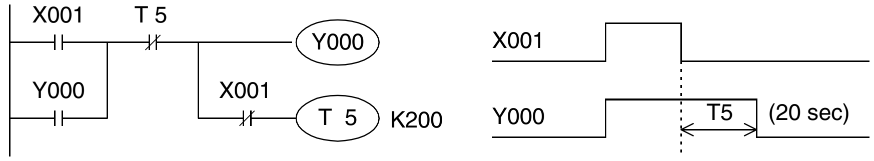
$$T^{+T_0}_{-\alpha}$$

α : 0.001 sec (timer for 1 ms), 0.01 sec (timer for 10 ms) or 0.1 sec (timer for 100 ms)
T : Timer set value (sec)
T_0: Operation cycle (sec)

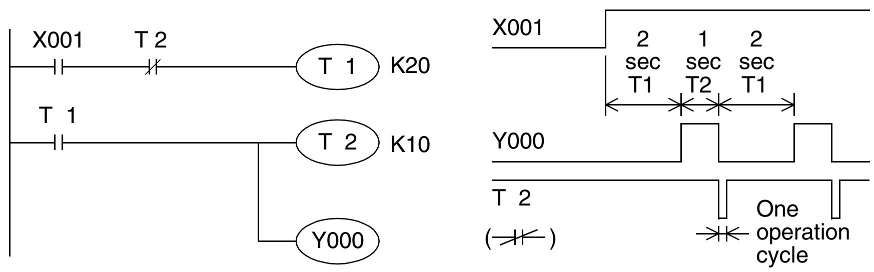
If the contact is programmed before the timer coil, "+2T_0" is obtained in the worst case.
When the timer set value is "0", the output contact turns on when a coil instruction is executed in the next cycle. An interrupt execution type timer for 1 ms pulses counts clock pulses of 1 ms as an interrupt processing after a coil instruction has been executed.

4.5.6 Program examples [off-delay timer and flicker timer]

Off-delay timer



Flicker timer (blink)



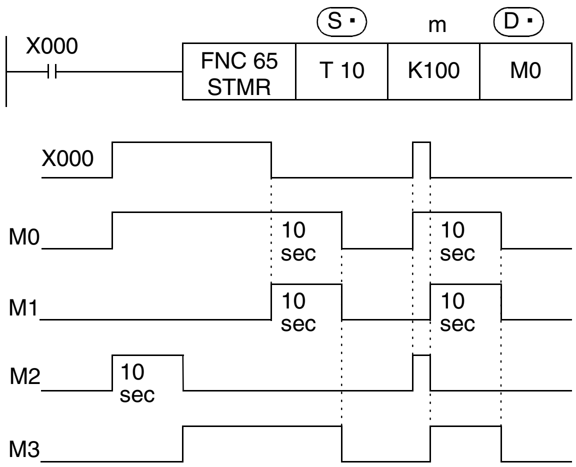
In addition, the flicker operation can be performed by the ALT (FNC 66) instruction.

Multi-timer by the applied instruction STMR (FNC 65)

By this instruction, off-delay timers, one-shot timers and flicker timers can be easily created.

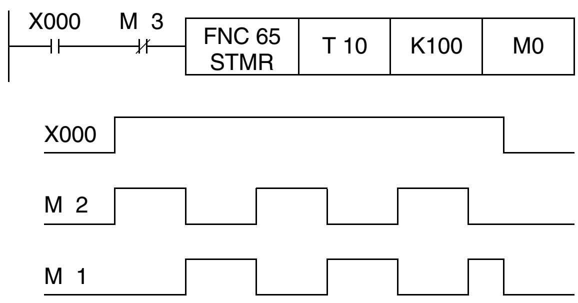
→ **For details, refer to Section 14.6.**

Off-delay timer and one-shot timer



- A value specified by "m" becomes the set value of the timer specified by (S·). 10-second in this example.
- M0 is an off-delay timer.
- M1 is a one-shot timer after "ON → OFF" operation.
- M2 and M3 are provided for a flicker timer, and connected as shown in the program example for flicker timer (below).

Flicker timer



- When M3 is connected as shown in the left figure, M2 and M1 become flicker outputs.
- When X000 is set to OFF, M0, M1 and M3 are turned OFF and T10 is reset after the set time.
- Do not use the timers here in other general circuits again.

In addition, the timer time can be set according to the switch input time by the teaching timer instruction TTMR (FNC 64).

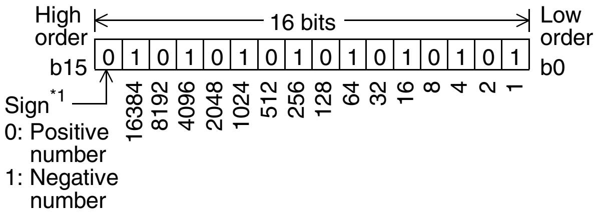
1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

4.5.7 Handling timers as numeric devices

In timers, the output contact operating in accordance with the set value is used in some cases, and the present value is used as numeric data for control in other cases.
The figures below show the structure of the timer present value registers. When a timer number is specified in an operand of an applied instruction, the timer is handled as a device storing 16-bit or 32-bit data in the same way as data registers.

1. Structure of timer present value register

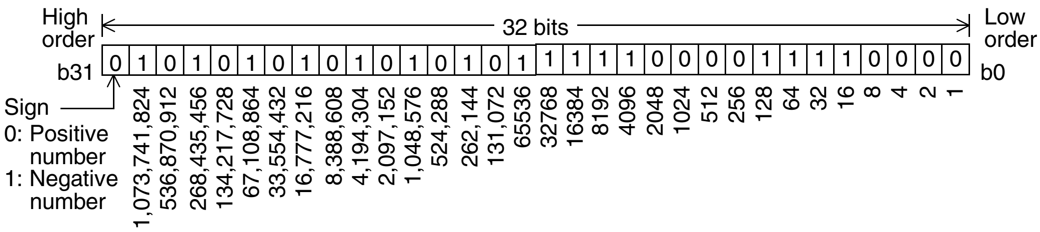
1) 16-bit



Available numeric value range
16-bit : 0 to 32,767
32-bit : -2,147,483,648 to +2,147,483,647

*1 The sign is valid only when a timer is handled as a substitute for data register.

2) 32-bit



2. Use examples in applied instructions

For the full use of timers as numeric devices, refer to the explanation of applied instructions later.

4.6 Counter [C]

4.6.1 Numbers of counters

The table below shows counter (C) numbers. (Numbers are assigned in decimal.)

→ For high speed counters, refer to Section 4.7.

	16-bit up counter Counting range: 0 to 32767		32-bit bi-directional counter Counting range: –2,147,483,648 to +2,147,483,647	
	General type	Latched (battery backed) type (protected by battery against power failure)	General type	Latched (battery backed) type (protected by battery against power failure)
FX3U/ FX3UC PLC	C 0 to C 99 100 points*1	C100 to C199 100 points*2	C200 to C219 20 points*1	C220 to C234 15 points*2

*1. This area is not latched (battery backed). It can be changed to a latched (battery backed) area by setting the parameters.

*2. This area is latched (battery backed). It can be changed to a non-latched (non-battery-backed) area by setting the parameters.

Counter numbers not used as counters can be converted as data registers for storing numeric values.

4.6.2 Features of counters

The table below shows the features of 16-bit counters and 32-bit counters. They can be used in accordance with the operating condition such as the counting direction switching and counting range, etc.

Item	16-bit counter	32-bit counter
Counting direction	Up-counting	Up-counting and down-counting can be switched (as shown in Subsection 4.6.3)
Set value	1 to 32767	–2,147,483,648 to +2,147,483,647
Set value specification	Constant (K) or data register	Constant (K) or a pair of data registers
Current value change	Does not change after counting up	Changes even after counting up (ring counter)
Output contact	Latches after counting up	Latches (in up-counting), or reset (in down-counting)
Reset operation	When RST instruction is executed, current value of counter is reset to "0" and output contact returns	
Current value register	16 bits	32 bits

4.6.3 Related devices (to specify counting direction) [32-bit counter]

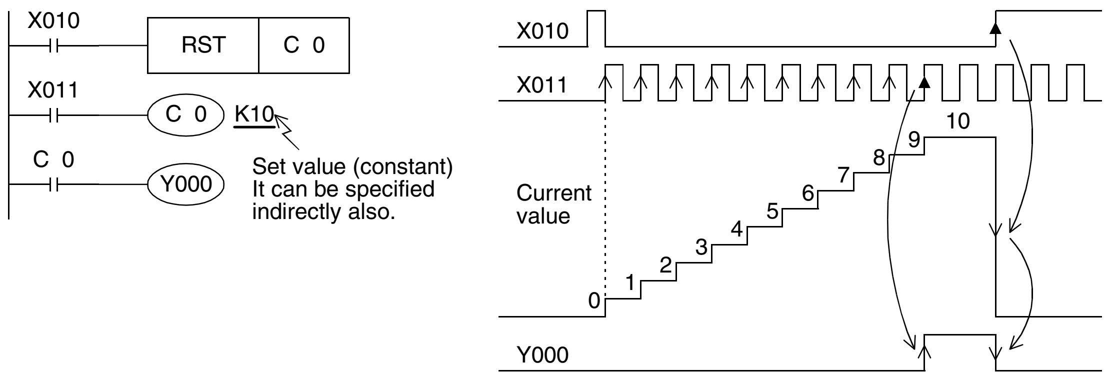
When an auxiliary relay for switching the counting direction is set to ON, the counter executes down-counting, and when set to OFF, the counter executes up-counting.

Counter No.	Counting direction switching relay	Counter No.	Counting direction switching relay	Counter No.	Counting direction switching relay	Counter No.	Counting direction switching relay
C200	M8200	C210	M8210	C220	M8220	C230	M8230
C201	M8201	C211	M8211	C221	M8221	C231	M8231
C202	M8202	C212	M8212	C222	M8222	C232	M8232
C203	M8203	C213	M8213	C223	M8223	C233	M8233
C204	M8204	C214	M8214	C224	M8224	C234	M8234
C205	M8205	C215	M8215	C225	M8225		
C206	M8206	C216	M8216	C226	M8226		
C207	M8207	C217	M8217	C227	M8227		
C208	M8208	C218	M8218	C228	M8228		
C209	M8209	C219	M8219	C229	M8229		

4.6.4 Functions and operation examples

1. General type and latched (battery backed) type 16-bit up counters

- The valid set range of 16-bit binary up counter is from K1 to K32767 (decimal constant).
  K0 provides the same operation as K1, and the output contact turns on at the first counting.
- In general type counters, the counter value is cleared when the PLC turns off. In latch type counters, however, the counter value just before power failure is stored (backed up by the battery); The counter value in the subsequent operations can be added to the last counter value.
- Every time the coil C0 is driven by the counting input X011, the current value of the counter increases.
  When a coil instruction is executed 10 times, the output contact turns on.
  After that, the current value of the counter does not change even if the counting input X011 turns on after that.
  When the RST input X010 turns ON and then RST instruction is executed, the current value of the counter is reset to "0" and the output contact returns.



- The counter set value can be set by a constant (K) as shown above, or indirectly specified by a data register number. For example, when D10 is specified and the contents of D10 are "123", it is equivalent to "K123".
- If data beyond the set value is written to the current value register by MOV instruction, etc., the OUT coil turns ON and the current value register becomes the set value when the next counting input is received.
- In a latch type counter, the current value, output contact operation and reset status of the counter are latched (battery backed) by the battery built in the PLC.

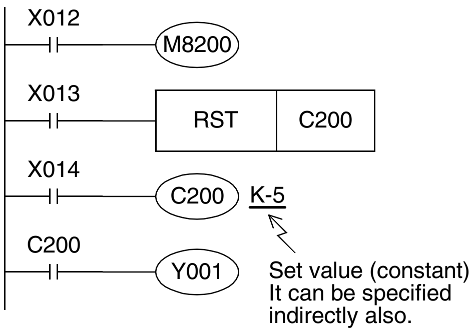
→ **For details on backup methods against power failure, refer to Section 2.6.**

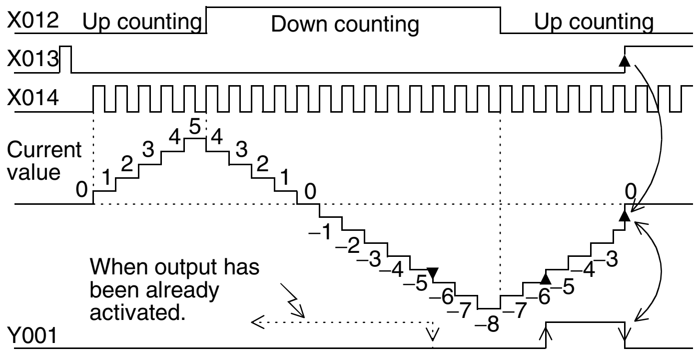
2. General type and latched (battery backed) type 32-bit bi-directional counters

The valid set range of 32-bit binary bi-directional counters is from –2,147,483,648 to +2,147,483,647 (decimal constant). The counting direction (up or down) is specified by special auxiliary relays M8200 to M8234.

- When M8△△△ is driven for C△△△, a counter executes down-counting. When M8△△△ is not driven, a counter executes up-counting. (Refer to the previous page.)
- The set value (positive or negative) can be specified by a constant (K) or the contents of data registers (D).
  When data registers are used, 32-bit data composed of paired serial devices are treated as set values.
  For example, when D0 is specified, D1 and D0 provide a 32-bit set value.

- When the coil C200 is driven by the counting input X014, a counter starts up-counting or down-counting. When the current value of a counter increases from "–6" to "–5", the output contact is set. When the current value decreases from "–5" to "–6", the output contact is reset.





- The current value increases or decreases regardless of the operation of the output contact. When a counter executes up-counting from "+2,147,483,647", the counter value becomes "–2,147,483,648". In the same way, when a counter executes down-counting from "–2,147,483,648", the counter value becomes "+2,147,483,647". (This type of counter is called ring counter.)
- When the reset input X013 turns ON and then RST instruction is executed, the current value of the counter is reset to "0" and the output contact returns.
- In a latch type counter, the current value, output contact operation and reset status of the counter are latched (battery backed) by the battery built in the PLC.

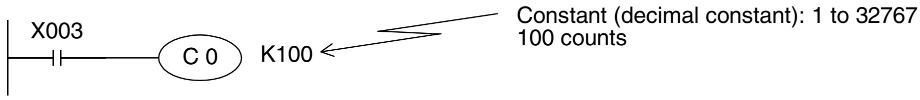
→ **For details on backup methods against power failure, refer to Section 2.6.**

- A 32-bit counter can be used as a 32-bit data register. 32-bit counters cannot be handled as target devices in 16-bit applied instructions.
- If data beyond the set value is written to the current value register by DMOV instruction, etc., the counter continues counting and the contact does not change when the next counting input is received.

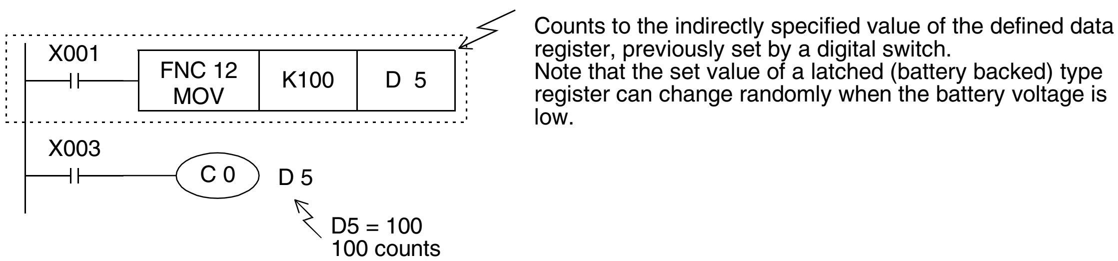
4.6.5 Set value specification method

1. 16-bit counter

1) Specification by constant (K)

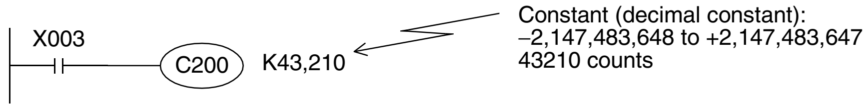


2) Indirect specification (D)

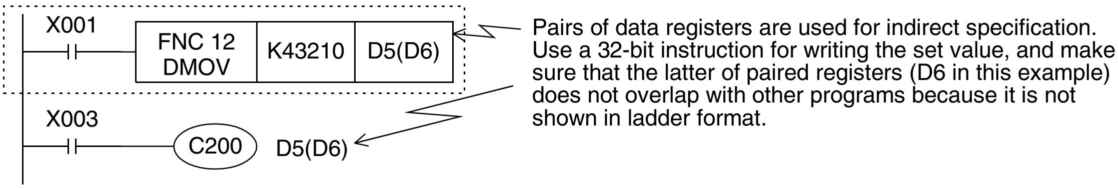


2. 32-bit counter

1) Specification by constant (K)



2) Indirect specification (D)



4.6.6 Response speed of counters

Counters execute counting by cyclic operating for contact operations of internal signals X, Y, M, S, C, etc. inside the PLC.
For example, when X011 is specified as counting input, its ON duration and OFF duration should be longer than the cycle time of the PLC (which is tens of Hz or less usually).
On the other hand, high speed counters described later execute counting as an interrupt processing for specific input, and can execute counting at 5 k to 6 kHz regardless of the cycle time.

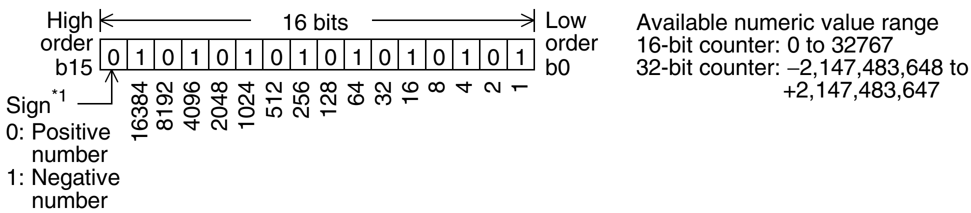
→ **For high speed counters, refer to Section 4.7.**

4.6.7 Handling counters as numeric devices

Counters use output contacts operating in accordance with the set value or use the counter value (current value) as numeric data for control.
The figure below shows the structure of the current value register of a counter. When a counter number is specified in an operand of an applied instruction in execution, the counter is handled as a device storing 16-bit or 32-bit data in the same way as data register.
A 32-bit counter is handled as 32-bit data.

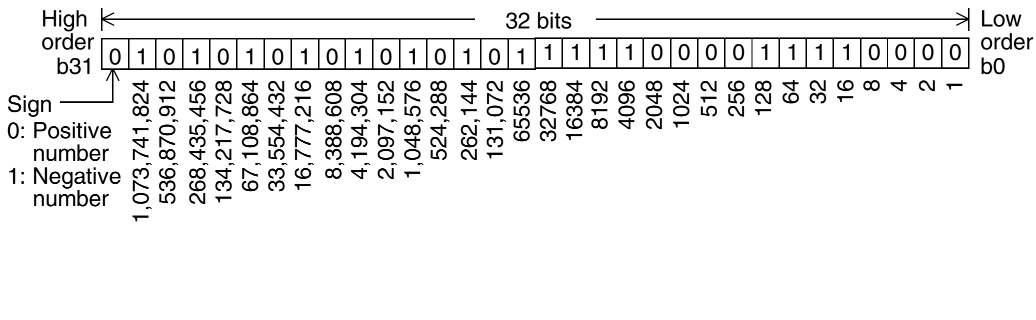
1. Structure of register storing current value of counter

1) 16-bit



*1. The sign is valid only when a counter is handled as a substitute for data register.

2) 32-bit



2. Examples in applied instructions

For the full use of counters as numeric devices, refer to the explanation of applied instructions later.

	FNC 12 MOV	C 20	D 10			C20 (current value) is transferred to D10.
	FNC 10 CMP	K100	C 30	M 0		A decimal integer "100" is compared with C30 (current value), and the result is output to M0 to M2.
	FNC 18 BCD	C 10	K2Y000			The contents of C10 (current value) are converted into BCD, and output to Y000 to Y007. (Seven-segment display unit is controlled.)
	FNC 22 MUL	C 5	K 2	D4(D5)		C5 (current value) is multiplied by 2, and transferred to (D5, D4).
*1	FNC 12 DMOV	C200	D0(D1)			C 200 (current value) is transferred to (D1, D0).
*1	FNC 11 DZCP	K100	K20000	C200	M10	C200 (current value) is compared with a decimal integer zone 100 to 20000, and the result is output to M10 and M11.

*1. Make sure to use 32-bit operation instructions for 32-bit counters.

4.7 High Speed Counter [C]

4.7.1 Types and device numbers of high speed counters

1. Types of high speed counters

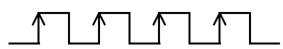
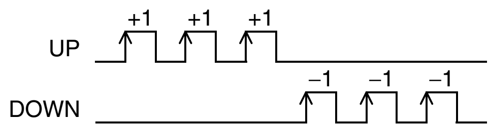
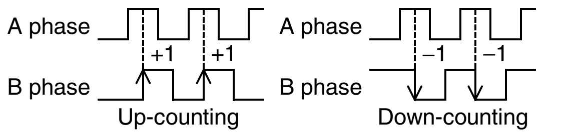
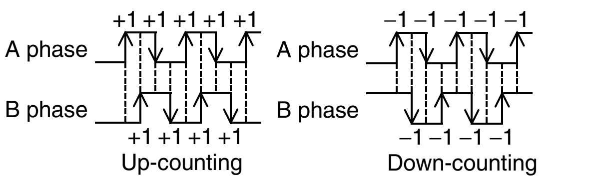
The main unit has built-in 32-bit high speed bi-directional counters (1-phase 1-count, 1-phase 2-count and 2-phase 2-count). These high speed counters are classified into hardware type or software type according to the counting method. Some high speed counters are capable of using an external reset input terminal and an external start input terminal (for counting start).

2. Classification of high speed counters according to counting method

- Hardware counters: These types of counters execute counting by hardware, but may be switched to software counters depending on the operating condition.
  → **For the condition handled as software counters, refer to Subsection 4.7.9.**
- Software counters: These types of counters execute counting as CPU interrupt processing.
  It is necessary to use each software counter within both limitations of maximum response frequency and total frequency.
  → **For the limitation of response frequency depending on the total frequency, refer to Subsection 4.7.10.**

3. Types of high speed counters and input signal forms

The table below shows the types (1-phase 1-count, 1-phase 2-count and 2-phase 2-count) and input signals (waveforms) of high speed counters.

		Input signal form	Counting direction
1-phase 1-count input			Down-count or up-count is specified by turning on or off M8235 to M8245. ON: Down-counting OFF: Up-counting
1-phase 2-count input			A counter executes up-count or down-count as shown on the left. The counting direction can be checked with M8246 to M8250. ON: Down-counting OFF: Up-counting
2-phase 2-count input	1 edge count		A counter automatically executes up-count or down-count according to changes in the input status of the A/B phase as shown on the left. The counting direction can be checked with M8251 to M8255. ON: Down-counting OFF: Up-counting
	4 edge count		

4. Cautions on counterpart equipment connected to high speed counter inputs

General-purpose inputs X000 to X007 are used for high speed counter inputs. An encoder[*1] adopting the output method shown in the table below can be connected depending on the connected terminal.
Encoders adopting the voltage output method and absolute encoders cannot be connected to high speed counter inputs.

→ **For the wiring, refer to the Hardware Edition of the main unit.**

Output method of encoder which can be directly connected to input terminal in main unit	Open collector transistor output method compatible with 24V DC
Output method of encoder which can be directly connected to input terminal in FX3U-4HSX-ADP	Differential line driver output method (output voltage: 5V DC or less)

*1. A rotary encoder adopting the output method shown above may not operate correctly depending on the electrical compatibility. Check the specifications before connecting an encoder.

5. High speed counter device list

	Classification	Counter No.	Edge count	Data length	External reset input terminal	External start input terminal
1-phase 1-count input	Hardware counter*1	C235*2 C236*2 C237*2 C238*2 C239*2 C240*2	–	32-bit bi-directional counter	Not provided	Not provided
		C244(OP)*3 C245(OP)*3	–			
	Software counter	C241 C242 C243	–		Provided*5	Not provided
		C244*3 C245*3	–		Provided*5	Provided
1-phase 2-count input	Hardware counter*1	C246*2 C248(OP)*2*3	–	32-bit bi-directional counter	Not provided	Not provided
	Software counter	C247 C248*3	–		Provided*5	Not provided
		C249 C250	–		Provided*5	Provided
2-phase 2-count input	Hardware counter*1	C251*2	1*4	32-bit bi-directional counter	Not provided	Not provided
			4*4			
		C253*2	1*4		Provided*5	
			4*4			
	Software counter	C252	1*4		Provided*5	Not provided
			4*4			
		C253(OP)*6	1*4		Not provided	
			4*4			
		C254 C255	1*4		Provided*5	Provided
			4*4			

*1. They are handled as software counters depending on the operating condition. When they are handled as software counters, they have limitations on both maximum response frequency and total frequency.

→ For the condition handled as software counters, refer to Subsection 4.7.9.
→ For the total frequency, refer to Subsection 4.7.10.

*2. Cautions on wiring should be considered for these high speed counters.

→ For the wiring, refer to the Hardware Edition of the main unit.

*3. C244, C245 and C248 are usually used as software counters, but can be used as hardware counters C244 (OP), C245 (OP) and C248 (OP) by combining a special auxiliary relay (M8388, M8390 to M8392).

→ For the method to switch the counter function, refer to Subsection 4.7.7.

*4. 2-phase 2-input counter is usually 1 edge count counter, but can be used as a 4 edge count counter by combining a special auxiliary relay (M8388, M8198 or M8199).

→ For the method to use a 2-phase 2-input 4 edge count counter, refer to Subsection 4.7.8.

*5. The external reset input is usually reset by turning ON, but can be changed to be reset by turning OFF by combining special auxiliary relays (M8388 and M8389).

→ For the method to change the logic of the external reset input, refer to Subsection 4.7.6.

*6. The counter C253 is usually used as a hardware counter, but can be used as the counter C253 (OP) not equipped with reset input by combining special auxiliary relays (M8388 and M8392).
In this case, C253 (OP) is handled as a software counter.

Notation of high speed counter devices

For some high speed counters in FX3U and FX3UC PLCs, the assignment of input terminals will switch when special auxiliary relays are used.
Such high speed counter devices are classified below. Note that description as (OP) is not available in programming.

Standard Device Numbers	Switched Device Numbers
C244	C244(OP)
C245	C245(OP)

Standard Device Numbers	Switched Device Numbers
C248	C248(OP)
C253	C253(OP)

4.7.2 Input assignment for high speed counters

Inputs X000 to X007 are assigned as shown in the table below according to each high speed counter number. When a high speed counter is used, the filter constant of a corresponding input number in the main unit automatically changes (X000 to X005: 5 μs, X006 and X007: 50 μs). Input terminals not used for high speed counters, however, can be used as general inputs.
When FX3U-4HSX-ADP unit is connected to an FX3U PLC, input terminals inside bold-line frames in the table below are assigned to the first FX3U-4HSX-ADP unit, and other input terminals are assigned to the second FX3U-4HSX-ADP unit.

→ For the input specifications of the FX3U-4HSX-ADP, refer to the FX3U Hardware Edition.
→ For the input specifications of the main unit, refer to the Hardware Edition of the main unit.

	Counter No.	Classification	Input terminal assignment							
			X000	X001	X002	X003	X004	X005	X006	X007
1-phase 1-count input	C235*1	H/W*2	U/D							
	C236*1	H/W*2		U/D						
	C237*1	H/W*2			U/D					
	C238*1	H/W*2				U/D				
	C239*1	H/W*2					U/D			
	C240*1	H/W*2						U/D		
	C241	S/W	U/D	R						
	C242	S/W			U/D	R				
	C243	S/W					U/D	R		
	C244	S/W	U/D	R					S	
	C244(OP)*3	H/W*2							U/D	
	C245	S/W			U/D	R				S
	C245(OP)*3	H/W*2								U/D
1-phase 2-count input	C246*1	H/W*2	U	D						
	C247	S/W	U	D	R					
	C248	S/W				U	D	R		
	C248(OP)*1*3	H/W*2				U	D			
	C249	S/W	U	D	R				S	
	C250	S/W				U	D	R		S
2-phase 2-count input*4	C251*1	H/W*2	A	B						
	C252	S/W	A	B	R					
	C253*1	H/W*2				A	B	R		
	C253(OP)*3	S/W				A	B			
	C254	S/W	A	B	R				S	
	C255	S/W				A	B	R		S

H/W: Hardware counter S/W: Software counter U: Up-counting input D: Down-counting input
A: A phase input B: B phase input R: External reset input S: External start input

*1. Cautions on wiring should be considered for these high speed counters.

→ For the wiring, refer to the Hardware Edition of the main unit.

*2. Hardware counters are switched to software counters when a comparison set/reset instruction for high speed counter (DHSCS, DHSCR, DHSZ or DHSCT) is used.
The counter C253 is switched to a software counter when the logic of the external reset input signal is reversed.

→ For the condition under which it is handled as a software counter, refer to Subsection 4.7.9.

*3. When a special auxiliary relay is driven in a program, the input terminals and their associated functions are switched.

→ For the method to use a software counter as a hardware counter, refer to Subsection 4.7.7.

*4. In a 2-phase 2-count input counter, the edge count is usually 1. But the edge count can be set to 4 by combining a special auxiliary relay.

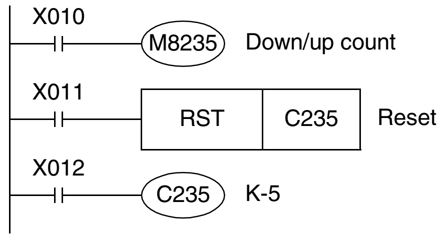
→ For the method on how to use a 2-phase 2-count input counter with on edge count of 4, refer to Subsection 4.7.8.

Restriction to overlap input numbers

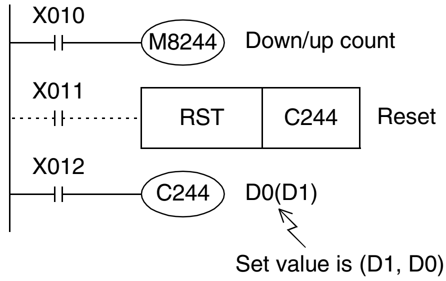
- Inputs X000 to X007 are used for high speed counters, input interrupt, pulse catch, SPD/ZRN/DSZR/DVIT instructions and general-purpose inputs. When assigning functions, there should be no overlap between those input terminals.
  For example, when C251 is used, X000 and X001 are occupied. As a result, "C235, C236, C241, C244, C246, C247, C249, C252 and C254", "input interrupt pointers I000 and I101", "pulse catch contacts M8170 and M8171" and "SPD, ZRN, DSZR and DVIT instructions using X000 and/or X001" cannot be used.
- Since the FX3U-4HSX-ADP and FX3UC PLC main unit share the same assigned input terminal numbers, only one of them may be used in operation. If both input terminals are used, intended operation is not enabled because the inputs of the FX3U-4HSX-ADP and PLC main unit operate in an "OR" relationship.

4.7.3 Handling of high speed counters

1. 1-phase 1-count input



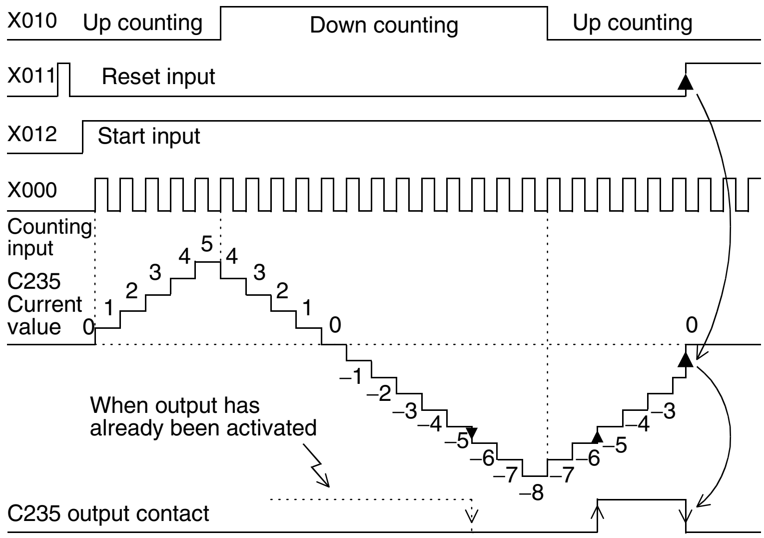
- C235 counts "OFF→ ON" of the input X000 while X012 is ON.
- When X011 turns ON and then RST instruction is executed, C235 is reset.
- The counting direction of the counters C235 to C245 is switched to down-count or up-count when M8235 to M8245 turns ON or OFF.



- C244 immediately starts counting when the input X006 turns ON while X012 is ON. The counting input is X000. In this example, the set value is indirectly specified by the contents of data registers (D1 and D0).
- A high speed counter can be reset using X011 in a sequence as shown in the figure, but C244 immediately reset without any program when X001 is closed. So a program with X011 is not necessary.
- The counting direction of the counters C235 to C245 is switched to down-count or up-count when M8235 to M8245 turns ON or OFF.

Operation example

The counter C235 shown above operates as follows:



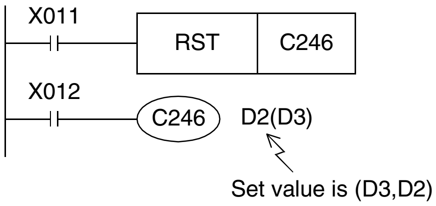
When counting with input X000, C235 executes up-count or down-count as an interrupt.

- When the current value of a counter increases from "–6" to "–5", the output contact is set. When the current value decreases from "–5" to "–6", the output contact is reset.

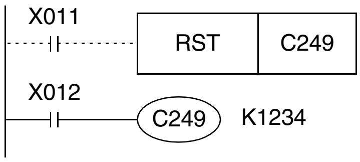
- The current value increases or decreases without regard to the operation of the output contact. When a counter executes up-count from "+2,147,483,647", the counter value becomes "–2,147,483,648". In the same way, when a counter executes down-count from "–2,147,483,648", the counter value becomes "+2,147,483,647". (This type of counter is called a ring counter.)
- When the reset input X011 turns ON and RST instruction is executed, the current value of the counter is reset to "0" and the output contact is restored.
- In a latch type high speed counter, the current value, output contact operation and reset status of the counter are latched (battery backed) by the backup battery built in the PLC.

2. 1-phase 2-count input

These counters are 32-bit binary bi-directional counters, and the operation of the output contact for the current value is equivalent to that in 1-phase 1-count input type high speed counters described above.



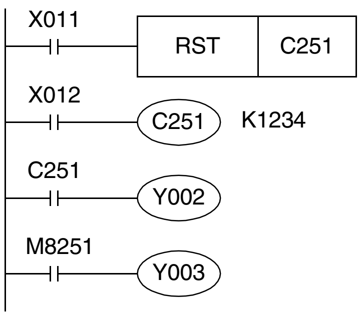
- While X012 is ON, C246 executes up-count when the input X000 turns from OFF to ON, and executes down-count when the input X001 turns from OFF to ON.
- The up/down-count operation of C246 to C250 can be checked with M8246 to M8250.
  ON status: Down-counting
  OFF status: Up-counting



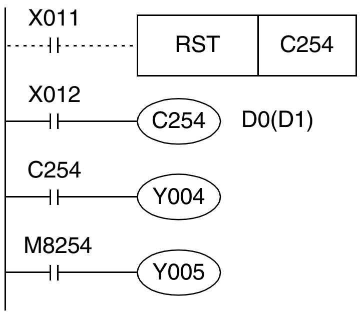
- While X012 is ON, C249 immediately starts counting when the input X006 turns ON.
  The up-count input is X000, and the down-count input is X001.
- A high speed counter can be reset by X011 in a sequence as shown in the figure, but C249 is immediately reset without any program when X002 is closed. So a program with X011 is not necessary.
- The up/down-count operation of C246 to C250 can be checked with M8246 to M8250.
  ON status: Down-counting
  OFF status: Up-counting

3. 2-phase 2-count input

These counters are 32-bit binary bi-directional counters, and the operation of the output contact for the current value is equivalent to that in 1-phase high speed counters described above.

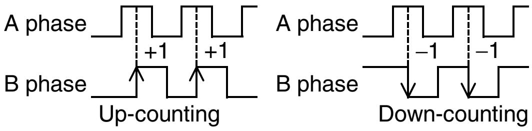


- While X012 is ON, C251 counts the operation of the inputs X000 (A phase) and X001 (B phase) as interrupt.
  When X011 turns ON a RST instruction is executed and C251 is reset.
- When the current value becomes equivalent to or larger than the set value, Y002 turns ON. When the current value becomes equivalent to or smaller than the set value, Y002 turns OFF.
- Y003 turns ON (for down-count) or OFF (for up-count) according to the counting direction.

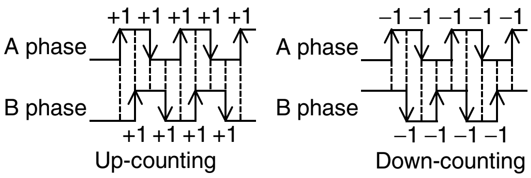


- When X006 turns ON while X012 is ON, C254 immediately starts counting. Its counting inputs are X000 (A phase) and X001 (B phase).
- In addition to reset by X011 in a sequence, C254 is reset immediately when X002 turns ON.
- When the current value becomes equivalent to or larger than the set value (D1, D0), Y004 turns ON. When the current value becomes equivalent to or smaller than the set value, Y004 turns OFF.
- Y005 turns ON (for down-count) or OFF (for up-count) according to the counting direction.

- A 2-phase encoder generates outputs for the A phase and B phase by a phase difference of 90°. With these outputs, a high speed counter automatically executes up-count and down-count as shown in the figure below.
  - When the counter is operating at the 1 edge count



  - When the counter is operating at the 4 edge count



- The down/up-count operation of C251 to C255 can be checked with M8251 to M8255.
  ON status: Down-counting
  OFF status: Up-counting

4.7.4 Current value update timing and comparison of current value

1. Current value update timing

A high speed counter executes up-count or down-count when a pulse is input to its input terminal, but the current value is updated at the timing shown in the table below. When using the current value of a hardware counter in a MOV, CMP or applied instruction such as the comparison instruction, special care must be taken since the current value update timing is affected by the ladder scans as shown in the table.

	Current value update timing
Hardware counter	When OUT or HCMOV instruction is executed for the counter
Software counter	Every time a pulse is input

2. Comparison of the Current value

The following two methods are available to compare and output the current value of a high speed counter.

1) Using the comparison instruction (CMP), zone comparison instruction (ZCP) or comparison contact instruction
   When the comparison result is necessary during counting operation[*1], comparison may be executed in the main program if the HCMOV instruction is used just before the comparison instruction (CMP or ZCP) or comparison contact instruction.

   *1. If it is necessary to execute comparison to update an output contact with the high-speed counter's changing value, use comparison instructions for high speed counters (HSCS, HSCR, HSZ or HSCT).

2) Using comparison instructions for high speed counters (HSCS, HSCR, HSZ or HSCT)
   The comparison instructions for high speed counters (HSCS, HSCR, HSZ and HSCT) execute a comparison and output the comparison result during high speed counting. The number of times these instructions can be used is limited as shown in the table below.
   When an output relay is specified for the comparison result, the comparison result is directly updated at the ON/OFF status of the output regardless of the output refresh by END instruction.
   Mechanical operation delay (about 10 ms) cannot be avoided in a relay output type PLC. Use a transistor output type PLC.

Instruction	Limitation in number of instruction
HSCS	Can be used up to 32 times including HSCT instruction.
HSCR	
HSZ[*1]	
HSCT[*1]	Can be used only once.

   *1. When HSZ or HSCT instruction is used, the maximum response frequency and total frequency of all software counters are affected.

→ **For the maximum response frequency and total frequency of software counters, refer to Subsection 4.7.10.**

4.7.5 Related devices

1. Devices used to switch the counting direction of 1-phase 1-count input counters

Type	Counter No.	Specifying device	Up-counting	Down-counting
1-phase 1-counting input	C235	M8235	OFF	ON
	C236	M8236		
	C237	M8237		
	C238	M8238		
	C239	M8239		
	C240	M8240		
	C241	M8241		
	C242	M8242		
	C243	M8243		
	C244	M8244		
	C245	M8245		

2. Devices used to check the counting direction of 1-phase 2-count input counters and 2-phase 2-count input counters

Type	Counter No.	Monitoring device	OFF	ON
1-phase 2-counting input	C246	M8246	Up-counting	Down-counting
	C247	M8247		
	C248	M8248		
	C249	M8249		
	C250	M8250		
2-phase 2-counting input	C251	M8251		
	C252	M8252		
	C253	M8253		
	C254	M8254		
	C255	M8255		

3. Devices used to switch the high speed counter function

Device No.	Name	Description
M8388	Contact for changing function of high speed counter	Changes the function of high speed counter.
M8389	Function switching device	Switches the logic of the external reset input. (For details, refer to Subsection 4.7.6.)
M8390		Switches the function of C244. (For details, refer to Subsection 4.7.7.)
M8391		Switches the function of C245. (For details, refer to Subsection 4.7.7.)
M8392		Switches the function of C248 and C253. (For details, refer to Subsection 4.7.7.)
M8198		Switches the edge count (between 1 and 4) of C251, C252 and C254. (For details, refer to Subsection 4.7.8.)
M8199		Switches the edge count (between 1 and 4) of C253, C255 and C253 (OP). (For details, refer to Subsection 4.7.8.)

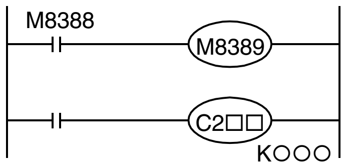
4. Operation status of hardware counters and software counters

Device No.	Name	Description	ON	OFF
M8380*1	Operation status flag	Operation status of C235, C241, C244, C246, C247, C249, C251, C252 or C254	Software counter	Hardware
M8381*1		Operation status of C236		
M8382*1		Operation status of C237, C242 or C245		
M8383*1		Operation status of C238, C248, C248(OP), C250, C253 or C255		
M8384*1		Operation status of C239 or C243		
M8385*1		Operation status of C240		
M8386*1		Operation status of C244(OP)		
M8387*1		Operation status of C245(OP)		

*1. Cleared when the PLC mode switches from STOP to RUN.

4.7.6 Changing the logic of external reset input signal

The counters C241 to C245, C247 to C250 and C252 to C255 are usually reset when the external reset input turns ON.
By using the program shown below, the logic can be inverted so that these counters are reset when the external reset input turns OFF.

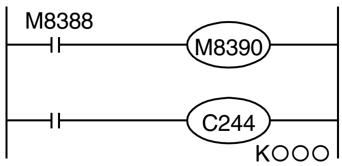
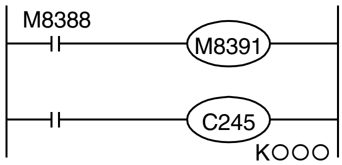
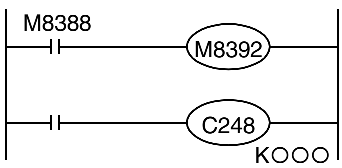
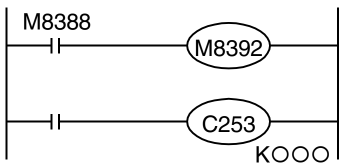
Counter No.	When inverting logic of external reset input signal	Description
C241 to C245 C247 to C250 C252 to C255		The logic of the external reset input is inverted so that the counters are reset when the input turns OFF. (The logic is inverted for all target counters.)

Caution

The counter C253 is switched to a software counter when the logic of the external reset input signal is inverted.

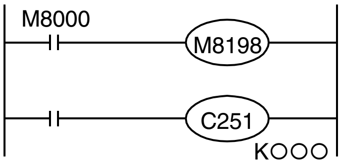
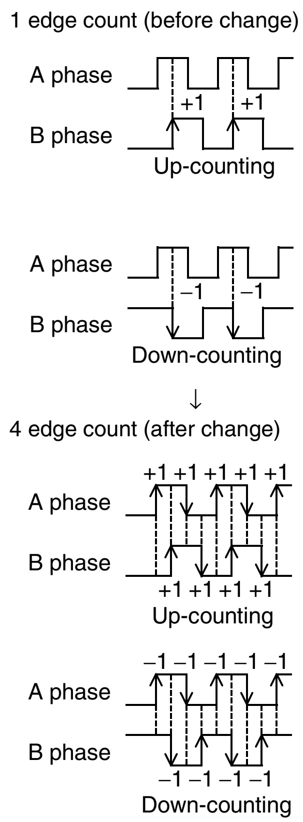
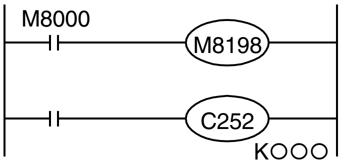
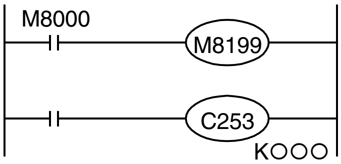
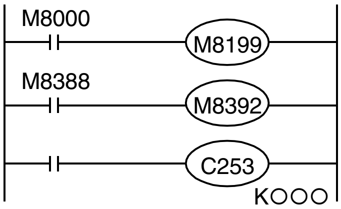
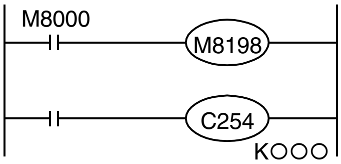
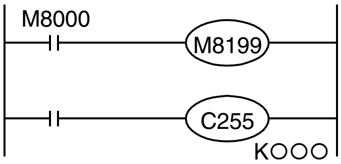
4.7.7 Assignment of counter input terminal and switching of function

The assignment of the input terminal and the function of the software counters C244, C245, C248 and C253 are changed as shown below when combined with the following special auxiliary relays.
In a program, put a special auxiliary relay just before a target counter.

Counter No.	When using software counter as hardware counter	Description
C244(OP)		• The counting input is changed from X000 to X006. • Reset input is not provided. • Start input is not provided. • It operates as a hardware counter.
C245(OP)		• The counting input is changed from X002 to X007. • Reset input is not provided. • Start input is not provided. • It operates as a hardware counter.
C248(OP)		• Reset input is not provided. • It operates as a hardware counter.
C253(OP)		• Reset input is not provided. • It operates as a software counter.

4.7.8 How to use 2-phase 2-count input counter C251 to C255 at 4 edge count.

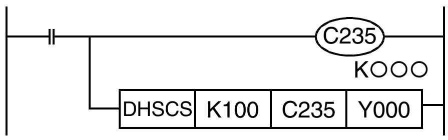
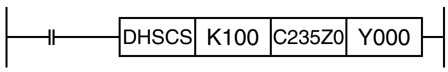
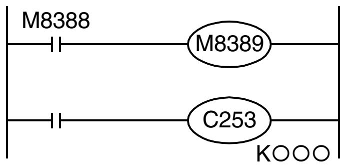
For the 2-phase 2-count input counters C251 to C255, the edge count is usually set to 1. By using the programs shown in the table below, the edge count may be set to 4.

<table>
<tr><th>Counter No.</th><th>When using 2-phase 2-counting input counter at edge count of 4</th><th>Description</th></tr>
<tr><td>C251</td><td>
</td><td rowspan="6">
</td></tr>
<tr><td>C252</td><td>
</td></tr>
<tr><td>C253</td><td>
</td></tr>
<tr><td>C253(OP)</td><td>
</td></tr>
<tr><td>C254</td><td>
</td></tr>
<tr><td>C255</td><td>
</td></tr>
</table>

4.7.9 Conditions for hardware counters to be handled as software counters

High speed counters are classified into hardware counters and software counters. In some conditions, however, hardware counters are handled as software counters.
In this case, use hardware counters within the range of maximum response frequency and total frequency as determined for software counters.

Conditions under which counters are handled as software counters

Hardware counter No.	Condition in which hardware counters are handled as software counters
C235 C236 C237 C238 C239 C240 C244(OP) C245(OP) C246 C248(OP) C251 C253	Because hardware counters execute counting at the hardware level of the FX3U/FX3UC, they can execute counting regardless of the total frequency. However, when hardware counters are handled as software counters with the following conditions, their maximum response frequency and total frequency are limited in the same way as the software counters. Use M8380 to M8387 to check whether high speed counters are handled as hardware counters or software counters. • When DHSCS (FNC 53), DHSCR (FNC 54), DHSZ (FNC 55) or DHSCT (FNC280) instruction is used for a hardware counter number, the hardware counter is handled as a software counter. Example: C235  In this case, C235 is handled as a software counter. • When an index register is used for a counter number specified in DHSCS (FNC 53), DHSCR (FNC 54), DHSZ (FNC 55) or DHSCT (FNC280) instruction, all hardware counters are handled as software counters. Example: C235Z0  • C253 (hardware counter) is handled as a software counter by inverting the logic using the external reset input signal logic changing function. Example: The logic of the C253 external reset input signal is inverted. →**For logic inversion of the external reset input signal, refer to Subsection 4.7.6.** 

4.7.10 Response frequency of high speed counters

1. Response frequency of hardware counters

The table below shows the maximum response frequency of hardware counters.
When hardware counters are handled as software counters in some operating conditions, their maximum response frequency becomes equivalent to that of software counters, and thus hardware counters are sometimes subject to restrictions in total frequency.

→ **For the conditions in which hardware counters are handled as software counters, refer to the previous page.**

<table>
<tr><th rowspan="2" colspan="2">Counter type</th><th rowspan="2">Counter No.</th><th colspan="2">Maximum response frequency</th></tr>
<tr><th>Main unit</th><th>FX3U-4HSX-ADP</th></tr>
<tr><td rowspan="2" colspan="2">1-phase 1-counting input</td><td>C235, C236, C237, C238, C239, C240</td><td>100 kHz</td><td rowspan="3">200 kHz</td></tr>
<tr><td>C244(OP), C245(OP)</td><td>10 kHz</td></tr>
<tr><td colspan="2">1-phase 2-counting input</td><td>C246, C248(OP)</td><td>100 kHz</td></tr>
<tr><td rowspan="2">2-phase 2-counting input</td><td>1 edge count</td><td rowspan="2">C251, C253</td><td>50 kHz</td><td>100 kHz</td></tr>
<tr><td>4 edge count</td><td>50 kHz</td><td>100 kHz</td></tr>
</table>

2. Response frequency and total frequency of software counters

The table below shows the maximum response frequency and total frequency of software counters.
When using the HSZ or HSCT instruction in a program, both the maximum response frequency and the total frequency are limited for all software counters without regarding the operands of the instruction. When examining a system or creating a program, consider the limitations, and use software counters within the allowable range of maximum response frequency and total frequency.

→ **For the conditions handled as software counters, refer to the previous page.**

1) When special analog adapters and FX3U/FX3UC Series special function blocks/units are not used

<table>
<tr><th rowspan="3" colspan="2">Counter type</th><th rowspan="3">Software counter</th><th rowspan="3">Following software counter with HSCS, HSCR, HSZ or HSCT instruction*1</th><th rowspan="3">Magnification for calculating total frequency</th><th colspan="8">Response frequency and total frequency according to instructions used</th></tr>
<tr><th colspan="2">When HSZ and HSCT instructions are not used</th><th colspan="2">When only HSCT instruction is used</th><th colspan="2">When only HSZ instruction is used</th><th colspan="2">When both HSZ and HSCT instructions are used</th></tr>
<tr><th>Maximum response frequency (kHz)</th><th>Total frequency (kHz)</th><th>Maximum response frequency (kHz)</th><th>Total frequency (kHz)</th><th>Maximum response frequency (kHz)</th><th>Total frequency (kHz)</th><th>Maximum response frequency (kHz)</th><th>Total frequency (kHz)</th></tr>
<tr><td rowspan="2" colspan="2">1-phase 1-counting input</td><td>C241, C242, C243, C244, C245</td><td>C235, C236, C237, C238, C239, C240</td><td>×1</td><td>40</td><td rowspan="5">80</td><td>30</td><td rowspan="5">60</td><td rowspan="4">40 - (Number of instruction)*2</td><td rowspan="5">80–1.5×(Number of instruction)</td><td rowspan="4">30-(Number of instruction)*2</td><td rowspan="5">60–1.5×(Number of instruction)</td></tr>
<tr><td>–</td><td>C244(OP), C245(OP)</td><td>×1</td><td>10</td><td>10</td></tr>
<tr><td colspan="2">1-phase 2-counting input</td><td>C247, C248, C249, C250</td><td>C246, C248(OP)</td><td>×1</td><td>40</td><td>30</td></tr>
<tr><td rowspan="2">2-phase 2-counting input</td><td>1 edge count</td><td rowspan="2">C252, C253(OP), C254, C255</td><td rowspan="2">C251, C253</td><td>×1</td><td>40</td><td>30</td></tr>
<tr><td>4 edge count</td><td>×4</td><td>10</td><td>7.5</td><td>(40-Number of instruction) ÷ 4</td><td>(30-Number of instruction) ÷ 4</td></tr>
</table>

*1. When an index register is added to a counter number specified by a HSCS, HSCR, HSZ or HSCT instruction, all hardware counters are switched to software counters.

*2. The high speed counters C244 (OP) and C245 (OP) can count up to 10 kHz.

2) When special analog adapters and FX3U/FX3UC Series special function blocks/units are used

Counter type				Magnification for calculating total frequency	Response frequency and total frequency according to instruction use condition							
		Software counter	Following software counter with HSCS, HSCR, HSZ or HSCT instruction*1		When HSZ and HSCT instructions are not used		When only HSCT instruction is used		When only HSZ instruction is used		When both HSZ and HSCT instructions are used	
					Maximum response frequency (kHz)	Total frequency (kHz)	Maximum response frequency (kHz)	Total frequency (kHz)	Maximum response frequency (kHz)	Total frequency (kHz)	Maximum response frequency (kHz)	Total frequency (kHz)
1-phase 1-counting input		C241, C242, C243, C244, C245	C235, C236, C237, C238, C239, C240	×1	30	60	25	50	30 - (Number of instruction)*2	50–1.5×(Number of instruction)	25-(Number of instruction)*2	50–1.5×(Number of instruction)
		–	C244(OP), C245(OP)	×1	10		10					
1-phase 2-counting input		C247, C248, C249, C250	C246, C248(OP)	×1	30		25					
2-phase 2-counting input	1 edge count	C252, C253(OP), C254, C255	C251, C253	×1	30		25					
	4 edge count			×4	7.5		6.2		(30-Number of instruction) ÷ 4		(25-Number of instruction) ÷ 4	

*1. When an index register is added to a counter number specified by a HSCS, HSCR, HSZ or HSCT instruction, all hardware counters are switched to software counters.

*2. The high speed counters C244 (OP) and C245 (OP) can count up to 10 kHz.

3. Calculation of the total frequency

Total frequency ≥ Sum of "Response frequency of high speed counter × Magnification for calculating total frequency"

→ **For a calculation example, see below.**

4. Calculation example

When only HSZ instruction is used 6 times in a program, the total frequency and response frequency are calculated as follows in accordance with the columns for "When only HSZ instruction is used" shown above.
This calculation example is provided for a system configuration not including special analog adapters and FX3U/FX3UC Series special function blocks/units.

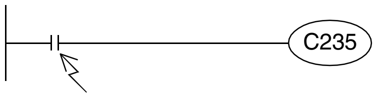
Used high speed counter No.		Input frequency	Maximum response frequency calculation	Magnification for calculating total frequency	Used instruction
C237	Operates as software counter	30 kHz	40 – 6 (times) = 34 kHz	× 1	HSZ instruction × 6 times
C241	Software counter	20 kHz	40 – 6(times) = 34 kHz	× 1	
C253(OP) [4 edge count]		4 kHz	{40 – 6(times)} ÷ 4 = 8.5 kHz	× 4	

1) The total frequency is calculated as follows because HSZ instruction is used 6 times:
Total frequency = 80 – 1.5 × 6 = 71 kHz
2) The sum of the response frequencies of the high speed counters being used is culculated of follows:
"30 kHz × 1[C237]" + "20 kHz × 1[C241]" + "4 × 4[C253(OP)]" = 66 kHz ≤ 71 kHz

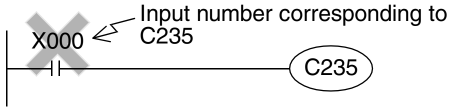
4.7.11 Cautions on use

- For a contact to drive the coil of a high speed counter, use a contact which is normally ON during high speed counting.

Example: M8000 (RUN monitor NO contact)

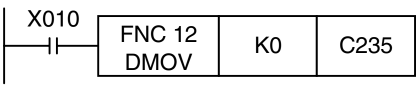


Program a contact which is normally ON during counting.



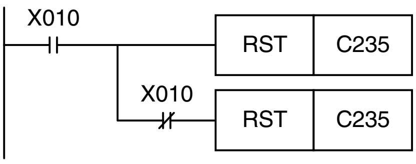
If a number of input relay for counting is specified, high speed counter cannot execute accurate counting.

- If the operation of a high speed counter is triggered by a device such as a switch, the counter may malfunction due to extra noise from switch chattering or contact bounce.
- The input filter of an input terminal for a high speed counter in the main unit is automatically set to 5 μs (X000 to X005) or 50 μs (X006 and X007).
  Accordingly, it is not necessary to use REFF instruction or special data register D8020 (input filter adjustment).
  The input filter for input relays not being used for high speed counters remains 10 ms (initial value).
- The inputs X000 to X007 are used for high speed counters, input interrupt, pulse catch, SPD/DSZR/DVIT/ZRN instructions and general-purpose inputs. There should be no overlap between each input number. For example, when C251 is used, X000 and X001 are occupied. As a result, "C235, C236, C241, C244, C246, C247, C249, C252 and C254", "input interrupt pointers I00* and I10*", "pulse catch contacts M8170 and M8171" and "SPD instruction using X000 and/or X001" cannot be used.
- When a counting pulse is not provided, none of the high speed counter output contacts will turn ON, even if the PLC executes an instruction where "present value = set value."
- Counting may be started or stopped for a high speed counter when the output coil (OUT C***) is set to ON or OFF. Program this output coil in the main routine. If the output coil is programmed in a step ladder (SFC) circuit, subroutine or interrupt routine, counting cannot be started or stopped until the step ladder or routine is executed.
- Make sure that the signal speed for high speed counters does not exceed the response frequency described above. If an input signal exceeds the response frequency, a WDT error or parallel link malfunction may occur.
- When a high speed counter is reset by RST instruction
  The high speed counter cannot start counting unless the drive contact for the RST instruction is set to OFF.
  When clearing only the current counter value or resetting both the OFF status of a contact and the current value, use the following programs:
  - When only the current value should be cleared



The current value of C235 is cleared (to "0").

  - When both the OFF status of a contact and the current value should be reset



C235 is reset.
Contact: Set to OFF, Current value: Cleared to "0"

Reset image of C235 is reset.

4.8 Data Register and File Register [D]

Data registers are devices for storing numeric data. File registers are handled as the initial values of data registers.
Each data register or file register stores 16-bit data (whose most significant bit specifies the positive or negative sign). Combined two data registers or file registers can store 32-bit numeric data (whose most significant bit specifies the positive or negative sign).

→ **For the functions and operations of file registers, refer to Subsection 4.8.4.**

4.8.1 Numbers of data registers and file registers

The table below shows numbers of data registers and file registers. (Numbers are assigned in decimal.)

	Data registers				File registers (latched (battery backed) type)
	General type	**Latched (battery backed) type (backed up by battery against power failure)**	**Fixed latched (battery backed) type (backed up by battery against power failure)**	**Special type**	
FX3U/ FX3UC PLC	D0 to D199 200 points*1	D200 to D511 312 points*2	D512 to D7999 7488 points*3*4	D8000 to D8511 512 points	D1000*4 and later 7000 points maximum

*1. This area is not latched (battery backed). It can be changed to the latched (battery backed) area by setting parameters.

*2. This area is latched (battery backed). It can be changed to the non-latched (non-battery-backed) area by setting parameters.

*3. The latch (battery backup) characteristics cannot be changed using parameters.

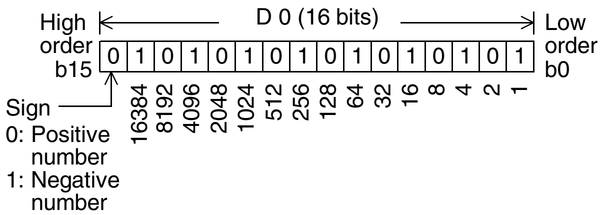
*4. Data registers D1000 and later can be used as file registers in units of 500 points by setting parameters.

When computer link PCs or parallel link is used, some data registers are occupied for the link.

→ **Refer to the Data Communication Edition.**

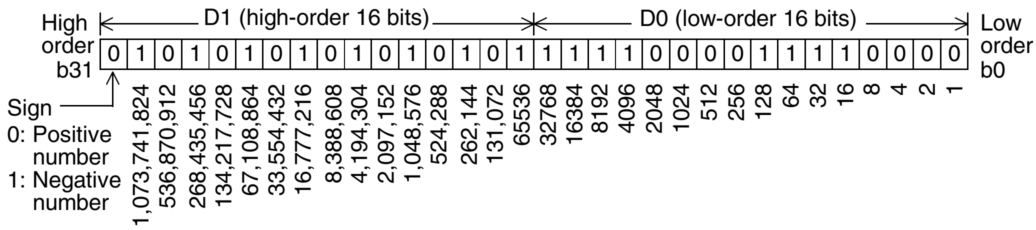
4.8.2 Structures of data registers and file registers

1) 16-bit type
One (16-bit) data register or file register can store a numeric value within the range from -32768 to +32767.



A numeric value can be read from or written to a data register by an applied instruction usually.
Or a numeric value can be directly read from or written to a data register from a display unit, display module, or programming tool.

2) 32-bit type
Two serial data registers or file registers can express 32-bit data.
- A data register with a larger device number handles high-order bits, and a data register with a smaller device number handles low-order bits.
- In the index type, V handles high-order bits, and Z handles low-order bits.
Two data registers or file registers can store a numeric value within the range from -2,147,483,648 to +2,147,483,647.



In the case of 32-bit type, when a data register or file register on the low-order side (example: D0) is specified, the subsequent number on the high-order side (example: D1) is automatically occupied.
Either an odd or even device number can be specified for the low-order side, but it is recommended to specify an even device number for the low-order side under consideration of the monitoring function of display units, display modules, and programming tools.

4.8.3 Functions and operation examples of data registers

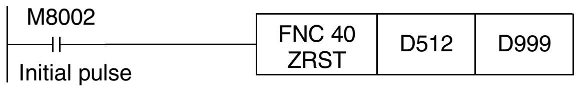
Data registers are devices for storing numeric data.
Each data register stores 16-bit data (whose most significant bit specifies the positive or negative sign). Two data registers combined can store 32-bit numeric data (whose most significant bit specifies the positive or negative sign).

1. General type and latched (battery backed) type data registers

- Once data is written to a data register it does not change unless other data overwrite it.
  When the PLC mode switches from "RUN" to "STOP" or when the power is interrupted, however, all data stored in general type data registers is cleared to "0".
  If the special auxiliary relay M8033 has been driven in advance, data is held even when the PLC mode switches from "RUN" to "STOP".
- Latched (battery backed) type data registers hold their contents even when the PLC mode switches from "RUN" to "STOP" or when the power is interrupted.
- The contents of data registers are backed up by the battery built in the PLC.

→ **For details on each backup method, refer to Section 2.6.**

- When using fixed latched (battery backed) type data registers as general type data registers, provide the following reset circuit by RST or ZRST instruction at the head step in a program.

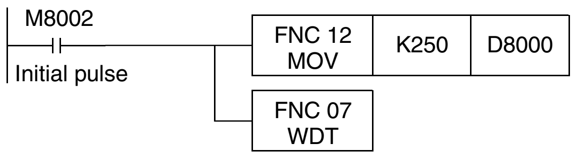


Data stored in D512 to D999 are cleared to "0".

→ **For file registers, refer to Subsection 4.8.4.**

2. Special type data registers

- Special type data registers contain informative, special purpose data and are sometimes written to during program operation.
  The contents of special type data registers are cleared to their initial values when restoring the power. (Generally, these data registers are cleared to "0" at first, and then the initial values (if there are any) are written by the system ROM.)
- For example, the watchdog timer time is set initially to D8000 by the system ROM. When changing the contents, write a desired time to D8000 by transfer instruction MOV (FNC 12).



The watchdog timer is set to 250 ms.

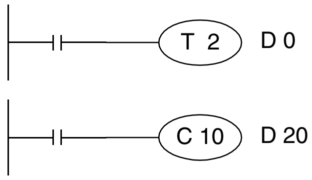
The watchdog timer is refreshed.

→ **For the data backup characteristics of special data registers, refer to Section 2.6 and Chapter 36.**
→ **For the types and functions of special data registers, refer to Chapter 36.**

3. Operation examples

Data registers can be used in various control with numeric data.
This section explains the operations of representative basic instructions and applied instructions among various applications.
For the full use of data registers, refer to the explanation of applied instructions later.

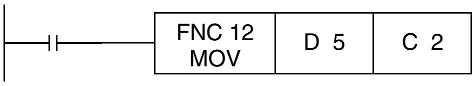
1) Data registers in basic instructions
Specifying the set value of a timer or counter



A counter or timer operates while regarding the contents of a specified data register as its set value.

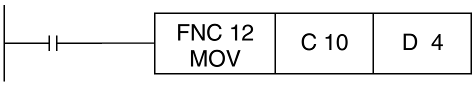
2) Data registers in applied instructions
Operation examples in FNC 12 (MOV) instruction

a) Changing the current value of a counter



The current value of the counter C2 is changed to the contents stored in D5.

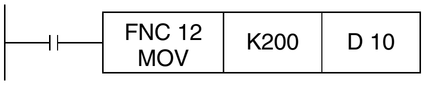
b) Reading the current value of a timer or counter to a data register



The current value of the counter C10 is transferred to D4.

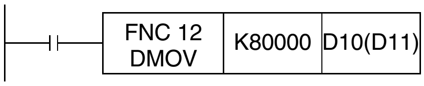
c) Storing a numeric value to data registers

16-bit



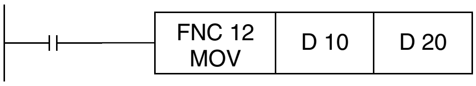
"200 (decimal value)" is transferred to D10.

32-bit



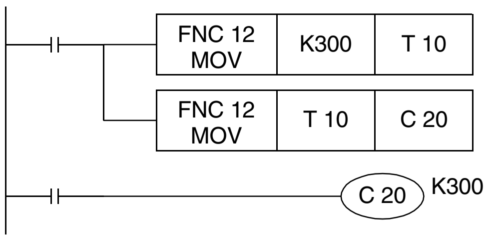
"80000 (decimal value)" is transferred to D10 and D11. Because a numeric value larger than 32767 is 32-bit data, a 32-bit operation is required. When a data register on the low-order side (D10) is specified, a data register on the high-order side (D11) is automatically occupied.

d) Transferring the contents of a data register to another data register



The contents of D10 are transferred to D20.

3) Using unoccupied timers and counters as data registers
Operation examples in FNC 12 (MOV) instruction
Timers and counters not in a program can be used as devices for storing 16-bit or 32-bit numeric values (data registers).



"300 (decimal value)" is transferred to T10.

The contents of T10 are transferred to the current value register of C20.
In this case, T10 is not working as a timer, but is working as a data register.

As in the case of data registers, when 16-bit timers or counters are used as 32-bit devices, two timers or two counters (example: C1 and C0) store 32-bit numeric data.
One 32-bit counter (example: C200) can store 32-bit numeric data.

4.8.4 Functions and operation examples of file registers

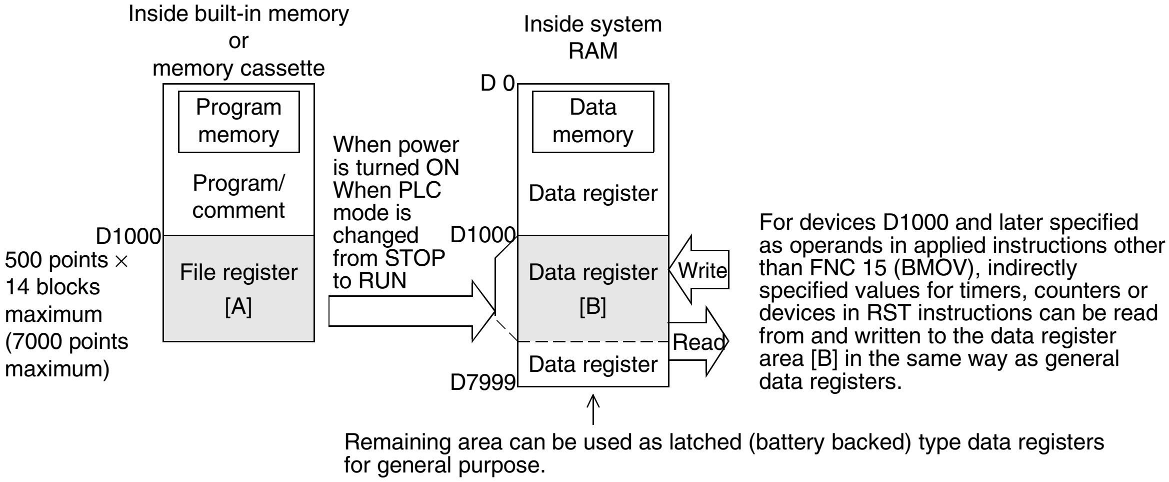
A file register is a device for setting the initial value of a data register with the same number.
Each file register stores 16-bit data (whose most significant bit specifies the positive or negative sign). Two file registers combined can store 32-bit numeric data (whose most significant bit specifies the positive or negative sign).
Fixed latched (battery backed) type data registers D1000 to D7999 can be specified as file registers using parameters.

- In parameter settings, 1 to 14 blocks can be specified. One block secures 500 file registers, but uses the program memory area by 500 steps.
- When some of file registers D1000 and later are set as file registers, the remaining registers can be used as latched (battery backed) type data registers.

This section explains how to handle file registers.

1. Operation of file registers

- The file register area [A] set inside the built-in memory or optional memory is batch-transferred to the data memory area [B] inside the system RAM when the power to the PLC is turned ON or when the PLC mode switches from STOP to RUN.
  The data register area [B] is latched (battery backed). When data registers are set as file registers by parameters, however, the file register area [A] inside the program memory is transferred when the power of the PLC is turned ON or when the PLC mode switches from STOP to RUN. This means that the contents changed in the data memory are reinitialized when the PLC turns ON or when the PLC mode switches from STOP to RUN.
  When it is necessary to save data changed in the data memory using a sequence program, update the file register area [A] to the changed values by the same-number register update mode in BMOV (FNC 15) instruction described later.



- Difference between BMOV (FNC 15) instruction and other instructions
  The table below shows the differences between the BMOV (FNC 15) instruction and other applied instructions.

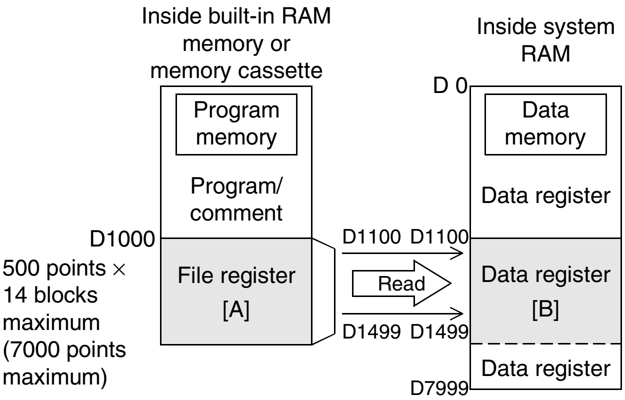
Instruction	Transferred contents	Remarks
BMOV instruction	Data can be read from and written to the file register area [A] inside the program memory.	–
Applied instructions other than BMOV instruction	Data can be read from and written to the data register area [B] inside the image memory in the same way as general data registers.	Because the data register area [B] is provided inside the system RAM in the PLC, its contents can be arbitrarily changed without being limited by the optional memory format.

The data stored in data registers set as file registers are automatically copied from the file register area [A] to the data register area [B] when restoring the power.

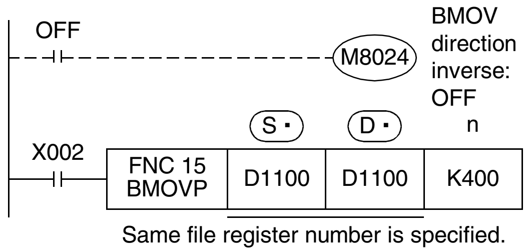
- When a file register is monitored from peripheral equipment, the data register area [B] inside the data memory is read.
  When "file register device current value change", "file register device forced reset" or "PLC memory all clear" is executed from peripheral equipment, the file register area [A] inside the program memory is changed, and then the data is automatically transferred to the data register area [B].
  Accordingly, when file register devices are overwritten, the program memory should be "built-in memory (RAM)" or "memory cassette (flash memory) whose protect switch is set to OFF". (The memory cassette (flash memory) cannot be overwritten from peripheral equipment if its protect switch is set to ON.)

2. File register ↔ Data register <updating the same number registers by BMOV (FNC 15) instruction>

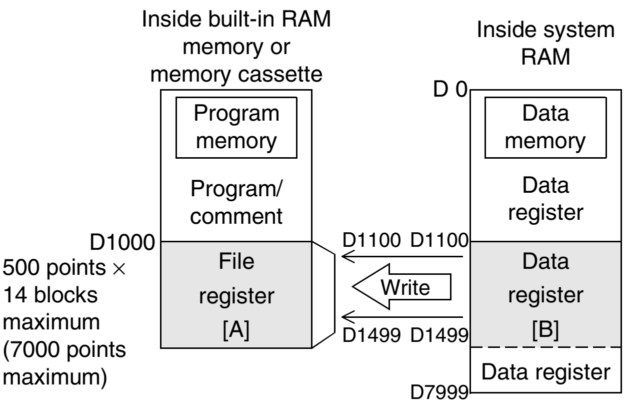
When the same file register is specified for both S· and D· in BOMV (FNC 15) instruction, this instruction specifies the same-number register update mode and executes the following operation:



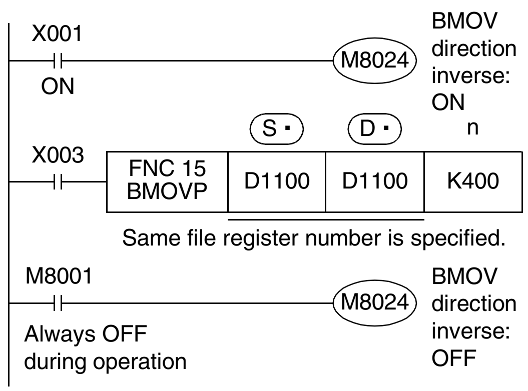
Read



•When X002 is set to ON while BMOV instruction direction reverse flag M8024 is OFF, the contents of a file register are transferred to the data register area [B] inside the data memory as shown in the figure on the left.



Write

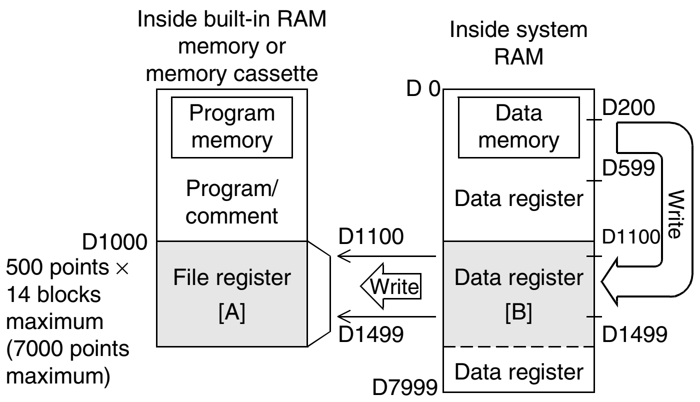


•When X003 is set to ON while BMOV instruction direction reverse flag M8024 is ON, the contents of a data register inside the data memory are written to the file register area inside the program memory as shown in the figure on the left.

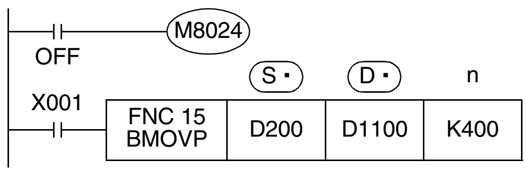
- When updating the contents of a file register in the same-number update mode, make sure that the file register numbers at S· and D· are equal to each other. Also make sure that the number of transfer points specified by "n" does not exceed the file register area. If the number of transfer points exceeds the file register area, an operation error occurs and the instruction is not executed.
- When S· and D· are indexed, the instruction is executed if the actual device number is within the file register area and if the number of transfer points is within the file register area also.

3. Data register → File register <writing by BMOV (FNC 15) instruction>

When a file register (D1000 or later) is specified for the destination of BMOV (FNC 15) instruction, it is possible to directly write data to the file register area [A] inside the program memory.



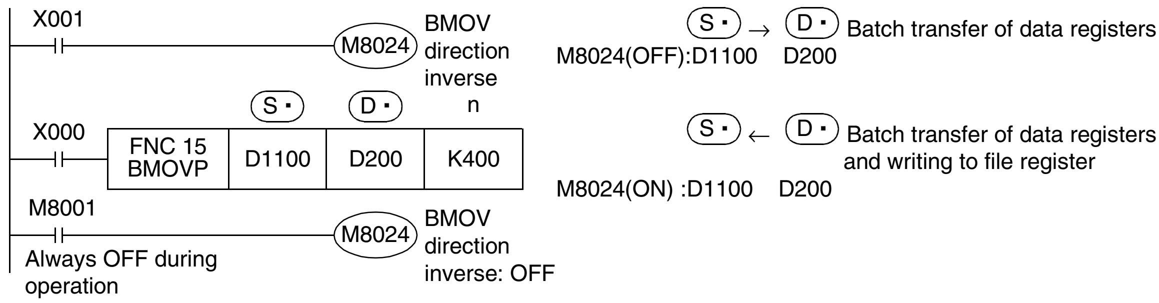
Write



- When X001 is set to ON, data is transferred to the data register area [B] and file register area [A] as shown in the figure on the left.
  If data cannot be written to the file register area [A] because the protect switch of the memory cassette (flash memory) is ON, data is written to only the data register area [B].
  When a file register device is specified for (D·) in a general applied instruction, data is transferred to only the data register area [B].
- A file register can be specified for (S·). If (D·) is the same as (S·), the same-number register update mode is selected.

→ **For the same-number register update mode, refer to the previous page.**

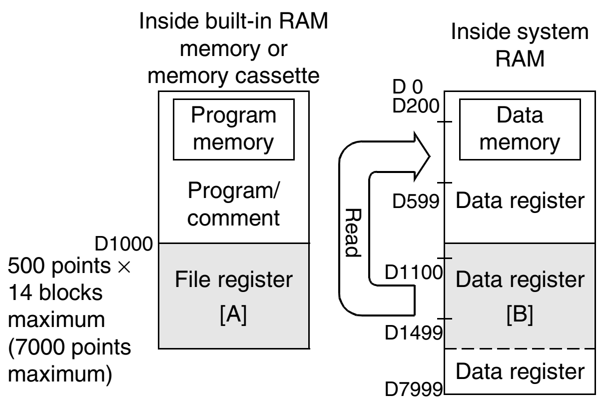
- By controlling BMOV instruction direction reverse flag M8024 for BMOV (FNC 15) instruction, data can be transferred in both directions in one program (as shown in the figure below).



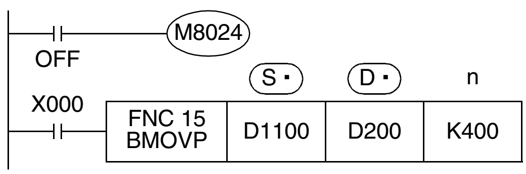
Cautions on reading

When a file register (D1000 or later) is specified for the source of BMOV (FNC 15) instruction and the same number file register is not specified for the destination, the contents of the file register area [A] inside the program memory are not read.

1) When a file register is specified for the source and a data register is specified for the destination



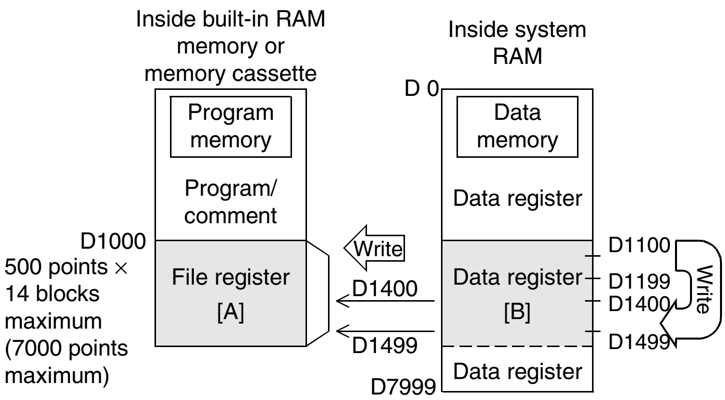
Batch transfer of data registers



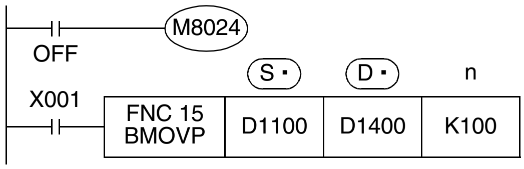
- When X000 is set to ON, the data register area [B] is read as shown in the figure on the left.
- A file register can be specified for (D·). If (S·) is the same as (D·), the same-number register update mode is selected.

→ **For the same-number register update mode, refer to the previous page.**

2) When file registers of different device numbers are specified for the source and destination



Write



- When X001 is set to ON, the contents of the data register area [B] are transferred to the data register area [B] and file register area [A] as shown in the figure on the left.
  If data cannot be written to the file register area [A] because the protect switch of the memory cassette (flash memory) is ON, data is written to only the data register area [B].

4.8.5 Cautions on using file registers

1. Cautions on using a memory cassette

When changing the contents of file registers stored in the flash memory inside the memory cassette, observe the following condition:

- Set to OFF the protect switch in the memory cassette.
- Allowable writing to a flash memory is up to 10000 times.
  When writing data using a continuous operation type instruction in a program, data is written to the flash memory in every operation cycle of the PLC.
  For preventing this, make sure to use a pulse operation type instruction (BMOVP).
- It takes 66 to 132 ms to write data to one serial block (500 points) in the flash memory. Execution of the program is paused during this period. Because the watchdog timer is not refreshed at this time, it is necessary to take proper countermeasures such as insertion of WDT instruction in a user program.

2. Cautions on handling file registers in the same-number register update mode in BMOV (FNC 15) instruction

- When updating the contents of the same number file register, make sure that the file register number at S· and D· are equal to each other.
- Make sure that the number of transfer points specified by "n" does not exceed the file register area.
- If the number of transfer points specified by "n" exceeds the file register area, an operation error (M8067) occurs and the instruction is not executed.
- In the case of indexing
  When S· and D· are indexed, the instruction is executed if the actual device number is within the file register area and the number of transfer points is within the file register area also.

4.9 Extension Register [R] and Extension File Register [ER]

Extension registers (R) are devices for extending data registers (D), and are backed up by the battery.
While a memory cassette is mounted, the contents of extension registers (R) can be stored in extension file registers (ER). However, extension file registers (ER) are available only while a memory cassette is mounted.

4.9.1 Numbers of extension registers and extension file registers

The table below shows numbers of extension registers (R) and extension file registers (ER).
(Numbers are assigned in decimal.)

	Extension register (R) (latched [battery backed] type)	Extension file register (ER) (file type)
FX3U/FX3UC PLC	R0 to R32767 32768 points	ER0 to ER32767 32768 points*1

*1. Available only while a memory cassette is mounted (because they are stored in the flash memory inside a memory cassette.)

4.9.2 Data storage destination and access method

Because the memory for storing data is different between extension registers and extension file registers, the access method is different as shown in the table below:

Data storage destination

Device	Data storage destination
Extension register	Built-in RAM (latched [battery backed] area)
Extension file register	Memory cassette (flash memory)

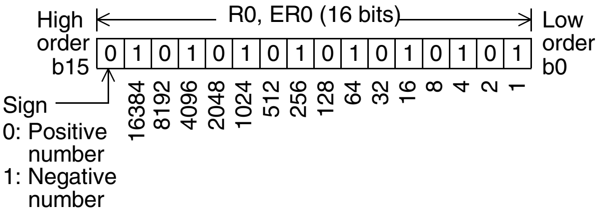
Difference in the access method

Access method		Extension register	Extension file register
Reading in program		✓	△Only dedicated instructions are enabled
Writing in program		✓	△Only dedicated instructions are enabled
Display module		✓	✓
Data change method	Test operation in online mode of GX Developer	✓	×
	Batch writing by GX Developer	✓	✓
	Computer link function	✓	×

4.9.3 Structures of extension registers and extension file registers

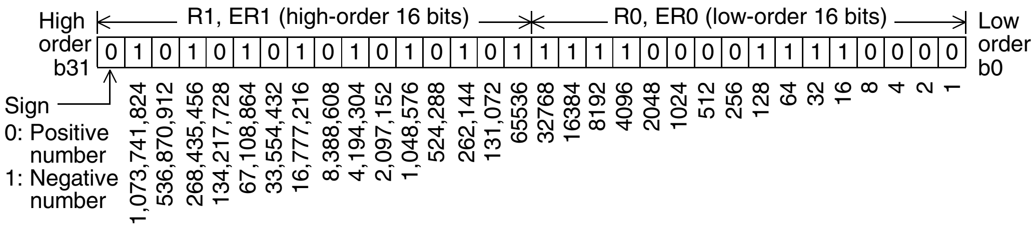
One extension register consists of 16 bits. Extension registers can be used in 16-bit and 32-bit applied instructions in the same way as data registers.

1) 16-bit type
One extension register (consisting of 16 bits) can handle a numeric ranging from –32768 to +32767.



A numeric value is usually read from and written to an extension register by applied instructions. However, a numeric value can also be directly read from and written to an extension register from a display unit, display module, or programming tool.

2) 32-bit type
Two serial extension registers (consisting of 32 bits) can express a 32-bit numeric value ranging from –2,147,483,648 to +2,147,483,647. (A larger number register handles high-order 16 bits, and a smaller number register handles low-order 16 bits.)



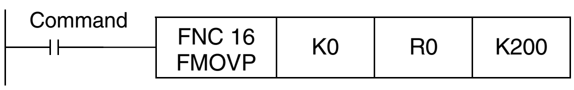
- In the case of 32 bit type, when an extension register on the low-order side (example: R0) is specified, the subsequent serial number on the high-order side (example: R1) is automatically occupied.
  Either an odd or even device number can be specified for the low-order side, but it is recommended to specify an even device number for the convenience of the monitoring function for display units, display modules, and programming tools.

4.9.4 Initialization of extension registers and extension file registers

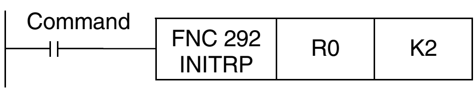
The contents of extension registers are backed up by the built-in battery even when the power is turned OFF or when the PLC mode switches from STOP to RUN.
When initializing the contents of extension registers, clear them using a sequence program or GX Developer.

1. When clearing the data using a program

- When initializing some extension registers (R)
  Example: When initializing (clearing) R0 to R199



- When initializing extension registers and extension file registers in sector units
  Example: When initializing R0 to R4095 and ER0 to ER4095 (initializing two sectors starting from R0 and ER0)



The current value in Ro to R4095 is initialized to "FFFFH".

2. When clearing the data using GX Developer

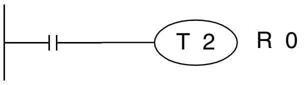
Select [Online] → [Clear PLC memory...] in GX Developer, and clear [Data device].
This operation initializes the contents of timers, counters, data registers, file registers and extension registers.

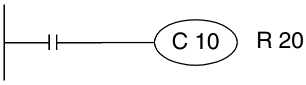
4.9.5 Functions and operation examples of extension registers

Extension registers can be used in various controls with numeric data the same as data registers.
This subsection explains operations in representative basic instructions and applied instructions among various applications.
For the full use of extension registers, refer to the explanation of applied instructions described later.

1. Extension registers in basic instructions

- Specifying an extension register as the set value of a timer or counter



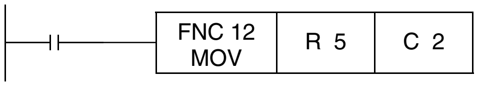


A counter or timer operates while regarding the contents of a specified data register as the set value.

2. Extension registers in applied instructions

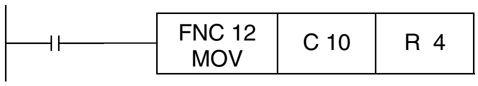
Operation examples in FNC 12 (MOV) instruction

- Changing the current value of a counter



The current value of the counter C2 is changed to the contents of R5.

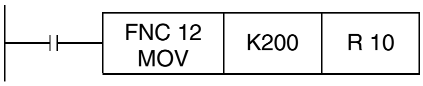
- Reading the current value of a timer or counter to an extension register



The current value of the counter C10 is transferred to R4.

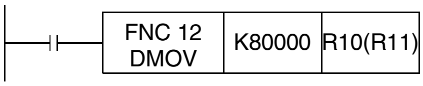
- Storing a numeric value to extension registers

16 bits



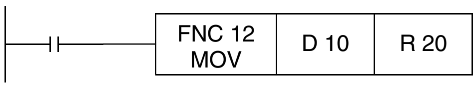
"200 (decimal value)" is transferred to R10.

32 bits



"80000 (decimal value)" is transferred to R10 and R11.
Because a numeric value larger than 32767, the 32-bit operation (double D instruction) is required. When a data register on the low-order side (R10) is specified, a data register on the high-order side (R11) is automatically occupied.

- Transferring the contents of a data register to another extension register



The contents of D10 are transferred to R20.

4.9.6 Functions and operation examples of extension file registers

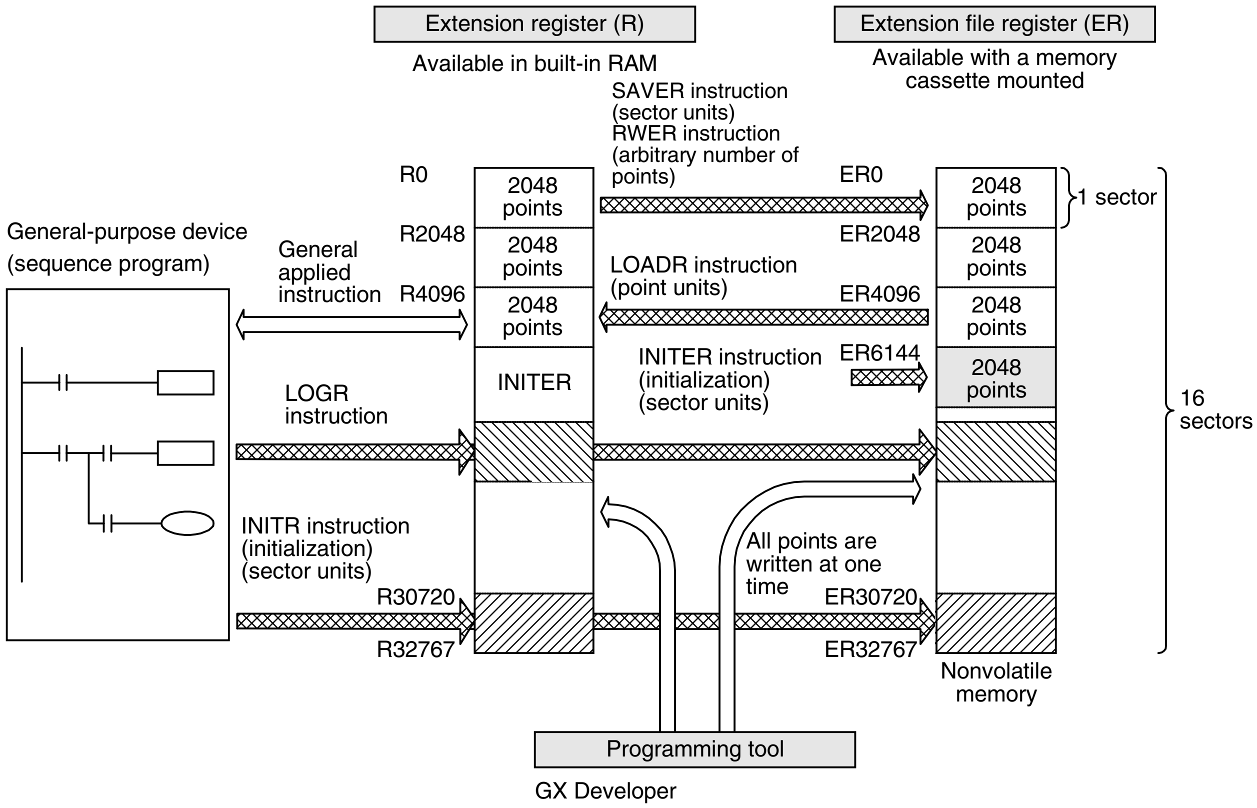
Extension file registers (ER) are usually used as log data storage destinations and set data storage destinations.
Extension file registers can be handled only with dedicated instructions shown in the table below. When using data contents with other instructions, transfer them to an extension register of the same device number, and then use the extension register.
However, note that extension file registers are available only while a memory cassette is mounted.

Instruction	Description
LOADR(FNC290)	This (transfer) instruction reads data of extension file registers (ER) to extension registers (R).
SAVER(FNC291)	This (transfer) instruction writes data of extension registers (R) to extension file registers (ER) in 2048 point (1 sector) units. Use this instruction to store newly created sectors (2048 points) of data to extension file registers (ER)[*1].
INITR(FNC292)	This instruction initializes extension registers (R) and extension file registers (ER)[*1] in 2048 point (1 sector) units. Use this instruction to initialize extension registers (R) and extension file registers (ER)[*1] before starting to log data by the LOGR instruction.
LOGR(FNC293)	This instruction logs specified data, and writes it to extension registers (R)[*1] and extension file registers (ER).
RWER(FNC294)	This (transfer) instruction writes specified extension registers (R) to extension file registers (ER)[*1].This instruction is supported in FX3UC PLCs Ver.1.30 or later.Use this instruction to store the contents of any extension register (R) to extension file register (ER)[*1].
INITER(FNC295)	This instruction initializes extension file registers (ER)[*1] in 2048 point (1 sector) units. This instruction is supported in FX3UC PLCs Ver.1.30 or later.Use this instruction to initialize extension file registers (ER)[*1] before executing SAVER instruction.

*1. Extension file registers are only accessible when a memory cassette is mounted.

1. Relationship between extension file registers and extension registers

Extension file registers and extension registers have the following positional relationship inside the PLC.



2. Sectors of extension registers and extension file registers

According to the data structure, extension registers and extension file registers are divided into sectors. One sector consists of 2048 devices. The table below shows the head device in each sector.

Sector No.	Head device No.	Device range
Sector 0	R0	ER0 to ER2047, R0 to R2047
Sector 1	R2048	ER2048 to ER4095, R2048 to R4095
Sector 2	R4096	ER4096 to ER6143, R4096 to R6143
Sector 3	R6144	ER6144 to ER8191, R6144 to R8191
Sector 4	R8192	ER8192 to ER10239, R8192 to R10239
Sector 5	R10240	ER10240 to ER12287, R10240 to R12287
Sector 6	R12288	ER12288 to ER14335, R12288 to R14335
Sector 7	R14336	ER14336 to ER16383, R14336 to R16383

Sector No.	Head device No.	Device range
Sector 8	R16384	ER16384 to ER18431, R16384 to R18431
Sector 9	R18432	ER18432 to ER20479, R18432 to R20479
Sector 10	R20480	ER20480 to ER22527, R20480 to R22527
Sector 11	R22528	ER22528 to ER24575, R22528 to R24575
Sector 12	R24576	ER24576 to ER26623, R24576 to R26623
Sector 13	R26624	ER26624 to ER28671, R26624 to R28671
Sector 14	R28672	ER28672 to ER30719, R28672 to R30719
Sector 15	R30720	ER30720 to ER32767, R30720 to R32767

4.9.7 Cautions on using extension file registers

1. Cautions on writing data to extension file registers

Because extension file registers are stored in the flash memory inside a memory cassette, pay attention to the following points:

- When writing data to extension file registers by SAVER instruction
  Initialize sectors to be written before executing this instruction. After initialization, store data to be written to extension registers.
  In FX3UC PLCs Ver.1.30 or later, it is not necessary to initialize sectors to be written when using RWER instruction.
- When writing data to extension file registers by LOGR instruction
  Initialize sectors to be written before starting to log data.
- When using INITR instruction
  This instruction initializes the contents of specified extension registers and extension file registers.
  When initializing only extension file registers by this instruction, make sure to temporarily move the contents of extension registers to unused extension registers or unused data registers before executing this instruction.
  When initializing only extension file registers in FX3UC PLCs Ver.1.30 or later, use INITER instruction.

2. Initialization of extension file registers

Because the contents of extension file registers are stored in the flash memory of a memory cassette, use the data clear operation in a sequence program or GX Developer to initialize them.

1) When initializing extension file registers in a program
   a) Initializing only extension file registers in sector units [Ver.1.30 or later]
      Example: When initializing ER0 to ER4095 (initializing two sectors starting from ER0)

Command input	FNC 295 INITERP	R0	K2

The current value is initialized to "FFFFH" in ER0 to ER4095.

   b) Initializing extension registers and extension file registers in sector units
      Example: When initializing R0 to R4095 and ER0 to ER4095 (initializing two sectors starting from R0 and ER0)

Command input	FNC 292 INITRP	R0	K2

The current value is initialized to "FFFFH" in R0 to R4095 and ER0 to ER4095.

2) When initializing extension file registers using GX Developer
   Select [Online] → [Clear PLC memory...] in GX Developer, and clear [Data device].
   This operation initializes the contents of timers, counters, data registers, file registers and extension registers.

4.9.8 Registration of data in extension registers and extension file registers

This subsection explains the operating procedure of GX Developer (Ver.8.23Z or later).

→ **For details on operating procedures of GX Developer, refer to the manual of GX Developer.**

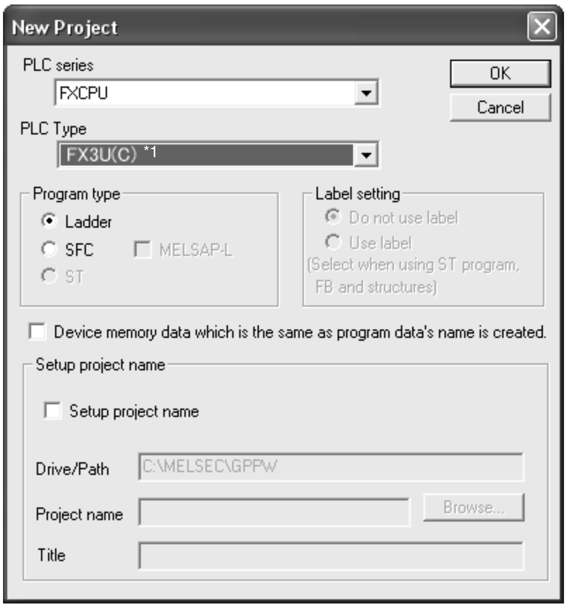
1 Starting up GX Developer (GPPW)

Click [Start]-[All Programs]-[MELSOFT Application]-[GX Developer] in Microsoft Windows.

2 Setting the PLC model

Set the PLC Series and PLC type as shown below.

Essential set item	Contents of setting
PLC series	FXCPU
PLC Type	FX3U(C)*1

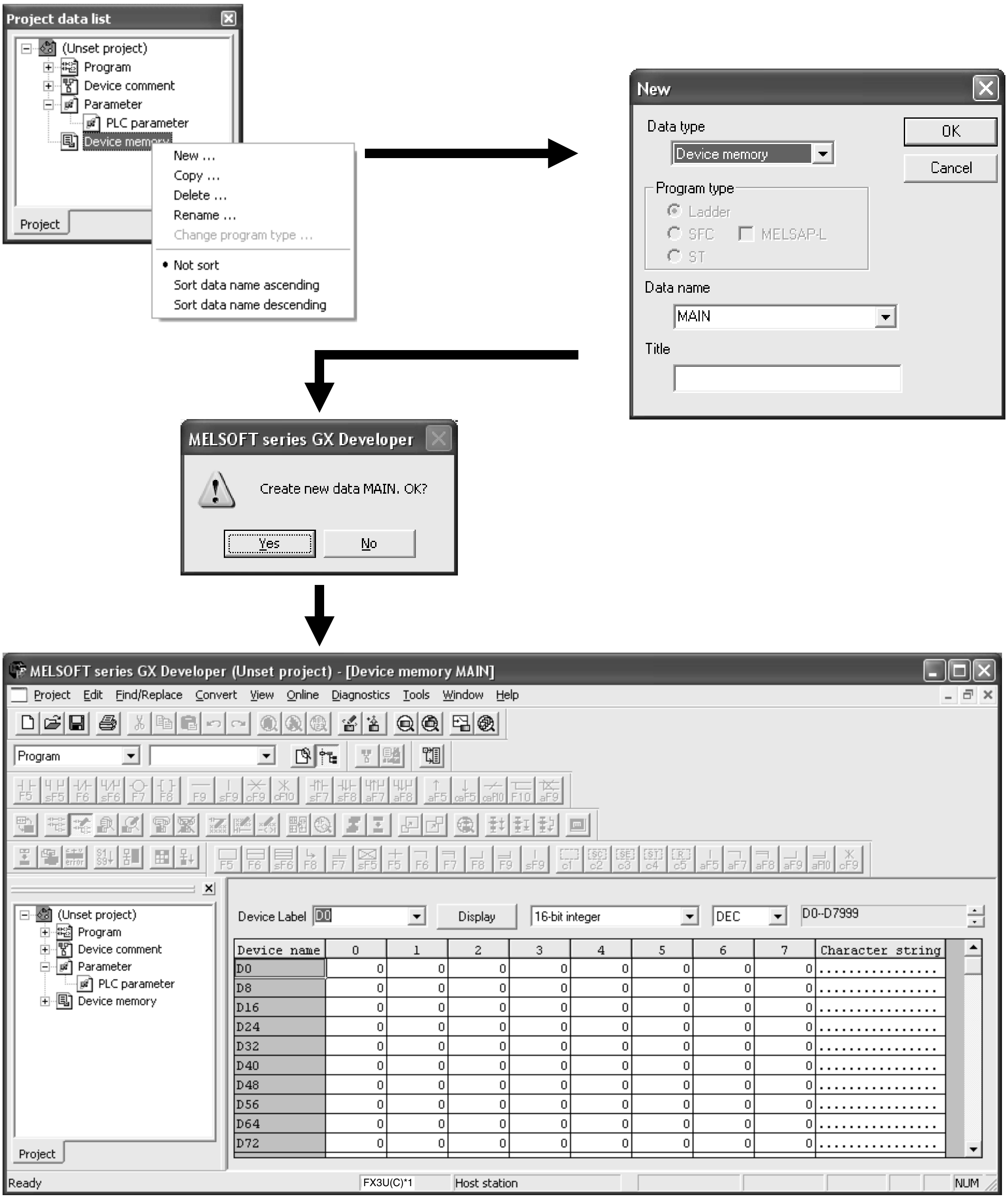


*1. For Ver. 8.23Z or 8.24A of GX Developer, select FX3UC for the PC type.

1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

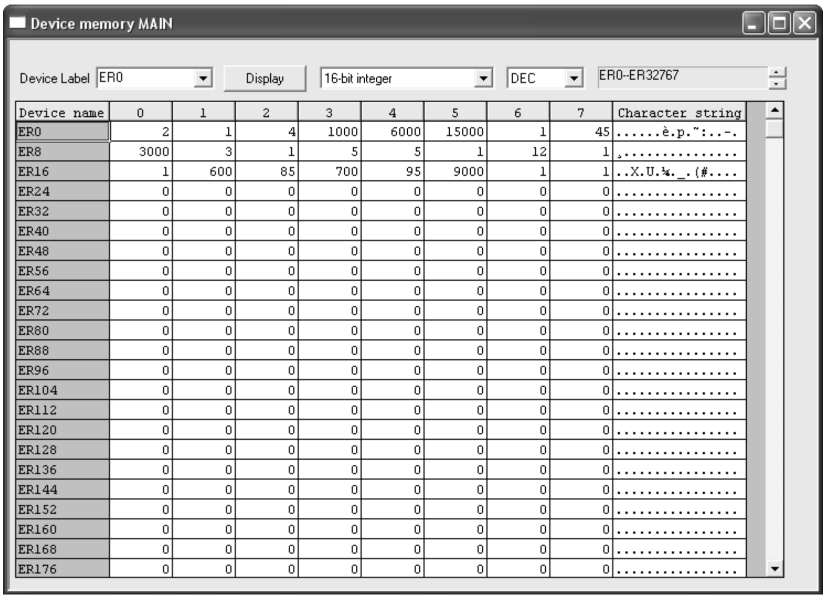
3 Setting the data

1. **Right-click [Device memory] in the project data list to open the submenu.**
2. **Click [New] on the submenu to display "New" dialog box.**
3. **Click the [OK] button to display the dialog box for confirmation.**
4. **Click the [Yes] button.**



*1. For Ver. 8.23Z or 8.24A of GX Developer, the PLC type is FX3UC.

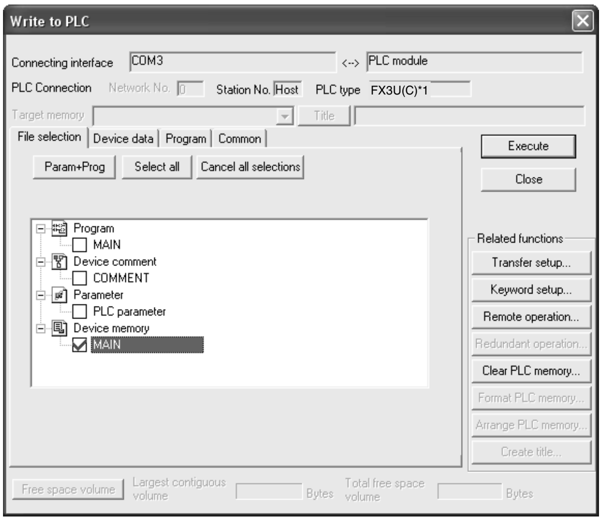
5. **Input a device number to be set to "Device Label", and click the [Display] button.**
6. **Select the data type to be set in the two selection boxes to the right of the [Display] button.**
7. **Input data or character string to each device accordingly.**



Device name	0	1	2	3	4	5	6	7	Character string
ER0	2	1	4	1000	6000	15000	1	45	......è.p.˜:..-.
ER8	3000	3	1	5	5	1	12	1	¸...............
ER16	1	600	85	700	95	9000	1	1	..X.U.¾._.(#....
ER24	0	0	0	0	0	0	0	0	....
ER32	0	0	0	0	0	0	0	0	....
ER40	0	0	0	0	0	0	0	0	....
ER48	0	0	0	0	0	0	0	0	....
ER56	0	0	0	0	0	0	0	0	....
ER64	0	0	0	0	0	0	0	0	....
ER72	0	0	0	0	0	0	0	0	....
ER80	0	0	0	0	0	0	0	0	....
ER88	0	0	0	0	0	0	0	0	....
ER96	0	0	0	0	0	0	0	0	....
ER104	0	0	0	0	0	0	0	0	....
ER112	0	0	0	0	0	0	0	0	....
ER120	0	0	0	0	0	0	0	0	....
ER128	0	0	0	0	0	0	0	0	....
ER136	0	0	0	0	0	0	0	0	....
ER144	0	0	0	0	0	0	0	0	....
ER152	0	0	0	0	0	0	0	0	....
ER160	0	0	0	0	0	0	0	0	....
ER168	0	0	0	0	0	0	0	0	....
ER176	0	0	0	0	0	0	0	0	....

4 Writing (transferring) data to the PLC

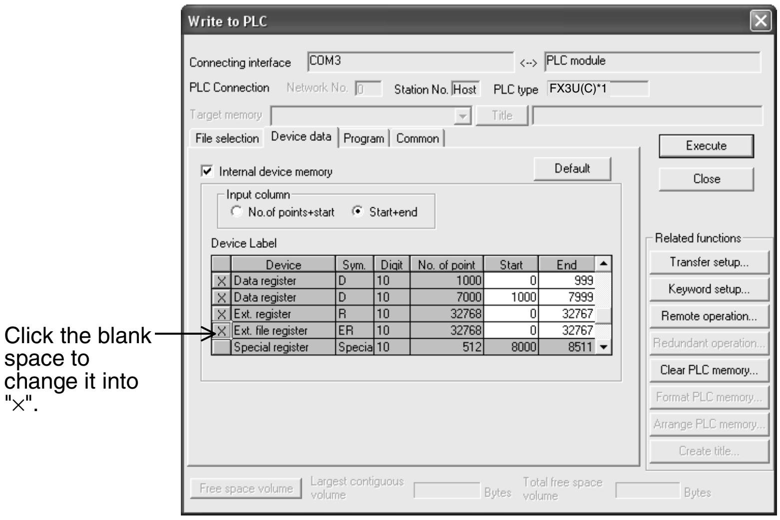
1. Select [Online] → [Write to PLC...] to open the [Write to PLC] dialog box.



*1. For Ver. 8.23Z or 8.24A of GX Developer, the PLC type is FX3UC.

2. Put a check mark to "MAIN (prepared device memory name)" under "Device memory."

3. Click "Device data" tab, and add "Ext. file register" to target devices.
In the default setting, "Ext. file register" is not included for reading/writing.
To add it, click the blank space on the left side of "Ext. file register" to change it to "×".
In GX Developer Ver.8.18U or later, the range of extension file registers to be written can be specified.



*1. For Ver. 8.23Z or 8.24A of GX Developer, the PLC type is FX3UC.

4. Click the [Execute] button to execute writing (transfer).

4.10 Index Register [V and Z]

Index registers can be used in the same way as of data registers. But they are special registers since they can change the contents of device numbers and numeric values by program when combined with another device number or numeric value in operands of applied instructions.

4.10.1 Numbers of index registers

The table below shows numbers of index registers (V and Z). (Numbers are assigned in decimal.)
When only "V" or "Z" is specified, it is handled as "V0" or "Z0" respectively.

	Index type
FX3U/FX3UC PLC	V0 (V) to V7, Z0 (Z) to Z7 16 points*1

*1. The characteristics related to protection against power failure cannot be changed by parameters.

4.10.2 Functions and structures

1. 16-bit type

Index registers have the same structures as data registers.

← 16 bits →	← 16 bits →
V	Z
V0 to V7: 8 points	Z0 to Z7: 8 points

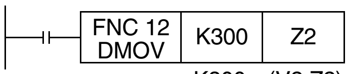
2. 32-bit type

Make sure to use Z0 to Z7 when indexing a device in a 32-bit applied instruction or handling a numeric value outside the 16-bit range.

← 32 bits →	
V0 (high-order side)	Z0 (low-order side)
V1 (high-order side)	Z1 (low-order side)
V2 (high-order side)	Z2 (low-order side)
V3 (high-order side)	Z3 (low-order side)
V4 (high-order side)	Z4 (low-order side)
V5 (high-order side)	Z5 (low-order side)
V6 (high-order side)	Z6 (low-order side)
V7 (high-order side)	Z7 (low-order side)

This is because FX PLCs handle Z as the low-order side of a 32-bit register as shown in combinations of V and Z in the figure on the left. Even if V0 to V7 on the high-order side is specified, indexing is not executed.
When index registers are specified as a 32-bit device, both V (high-order side) and Z (low-order side) are referred to at the same time.
If a numeric value for another purpose remains in V (high-order side), consequently the numeric value here becomes extremely large, thus an operation error occurs.

Example of writing to 32-bit index registers



K300→(V2,Z2)

Even if an index value in a 32-bit applied instruction does not exceed the 16-bit numeric range, use a 32-bit operation instruction such as DMOV instruction for writing a numeric value to Z as shown in the figure on the left so that both V (high-order side) and Z (low-order side) are overwritten at the same time.

4.10.3 Indexing of devices

Available devices and the contents of indexing are as described below:

→ **For indexing method and cautions, refer to Section 5.7.**

Decimal devices/numeric values: M, S, T, C, D, R, KnM, KnS, P and K

For example, when "V0 = K5" is specified and "D20V0" is executed, an instruction is executed for the device number D25 (D20 + 5).

Constants can be indexed also. When "K30V0" is specified, an instruction is executed for decimal value K35 (30 + 5).

Octal devices: X, Y, KnX and KnY

For example, when "Z1 = K8" is specified and "X0Z1" is executed, an instruction is executed for the device number X10 (X0 + 8: addition of octal value). When indexing for a device whose device number is handled in octal, a numeric value converted into octal is added for the contents of V and Z.

Accordingly, note that when "Z1 = K10" is specified "X0Z1" indicates that X12 is specified, and X10 is not specified.

Hexadecimal numeric values: H

For example, when "V5 = K30" is specified and a constant "H30V5" is specified, it is handled as H4E (30H + K30). When "V5 = H30" is specified and a constant "H30V5" is specified, it is handled as H60 (30H + 30H).

4.11 Pointer [P and I]

4.11.1 Numbers of pointers

The table below shows numbers of pointers (P and I). (Numbers are assigned in decimal.)
When using a pointer for input interrupt, an input number assigned to it cannot be used together with a "high speed counter" or "speed detection (FNC 56)" which uses the same input range.

	For branch	For branch: For jump to END step	For input interrupt/input delay interrupt	For timer interrupt	For counter interrupt
FX3U/ FX3UC PLC	P0 to P62 P64 to P4095 4095 points	P63 1 point	I00□(X000) I30□(X003) I10□(X001) I40□(X004) I20□(X002) I50□(X005) 6 points	I6□□ I7□□ I8□□ 3 points	I010 I040 I020 I050 I030 I060 6 points

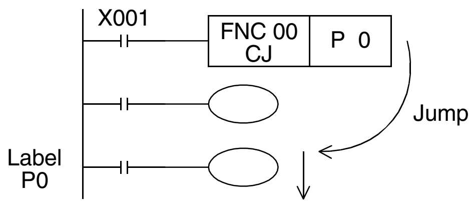
4.11.2 Functions and operation examples of pointers for branch

The roles and operations of pointers for branch are as described below.
Because all of these pointers are combined with applied instructions, refer to the explanation of each instruction for the detailed method.

→ For details on interrupt function, refer to Chapter 35.

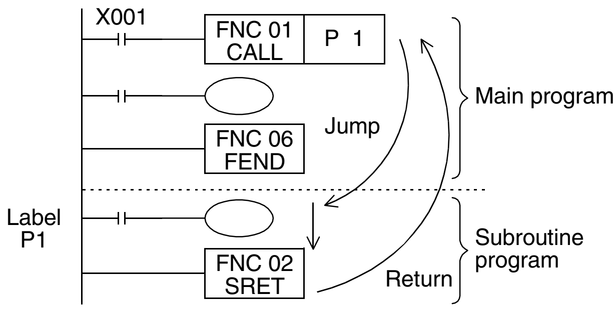
1. Applied instructions using pointers for branch (P)

- CJ (FNC 00) (conditional jump)



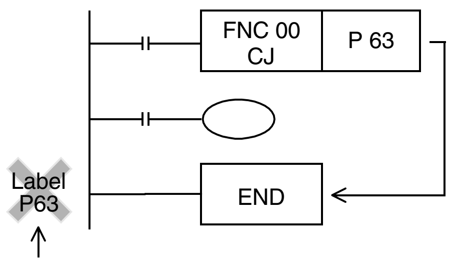
When X001 turns ON, the PLC jumps to a label position specified by CJ (FNC 00) instruction, and executes the subsequent program.

- CALL (FNC 01) call subroutine



When X001 turns ON, the PLC executes a subroutine in the label position specified by CALL (FNC 01) instruction, and then returns to the original position by SRET (FNC 02) instruction.

- Role of pointer P63 for jump to the END step



Should not be programmed.

P63 is a special pointer for jumping to the END step when the CJ (FNC 00) instruction is executed.
Note that a program error will accur when P63 is programmed as a label.

→ Refer to "5. Label unnecessary for the pointer P63" in Section 8.1.

4.11.3 Functions and operation examples of pointers for interrupt

→ **For details on interrupt function, refer to Chapter 35.**

There are three types of pointers for interrupt. When in use, they are combined with IRET(FNC03), EI(FNC04) and DI(FNC05) for interrupt return, enabling interrupt and disabling interrupt.

1. Pointers for input interrupt (delay interrupt): 6 points

→ **For details on input interrupt function, refer to Section 35.3 and Section 35.4.**

The PLC can receive input signals from specific input numbers without influence of the operation cycle of the PLC. By using these input signals as triggers, the PLC executes interrupt routine programs.
Because pointers for input interrupt can handle signals shorter than the operation cycle, use them for high priority processing during sequence control and for control handling short pulses.

Input	Pointer for input interrupt		Interrupt disabling flag	ON duration or OFF duration of input signal
	Interrupt at rising edge	Interrupt at falling edge		
X000	I001	I000	M8050*1	5 μs or more
X001	I101	I100	M8051*1	
X002	I201	I200	M8052*1	
X003	I301	I300	M8053*1	
X004	I401	I400	M8054*1	
X005	I501	I500	M8055*1	

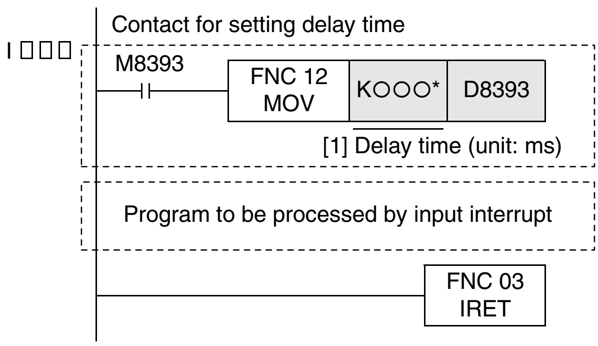
*1. Cleared when the PLC mode switches from RUN to STOP.

Non-overlap of input numbers

Inputs X000 to X007 are used for high speed counters, input interrupt, pulse catch, SPD/ZRN/DSZR/DVIT instructions and general-purpose inputs. When assigning functions, there should be no overlap between those input terminals.
For example, when the input interrupt pointer I001 is used, X000 is occupied. As a result, "C235, C241, C244, C246, C247, C249, C251, C252 and C254", "input interrupt pointer I000", "pulse catch contact M8170" and "SPD instruction using X000" cannot be used.

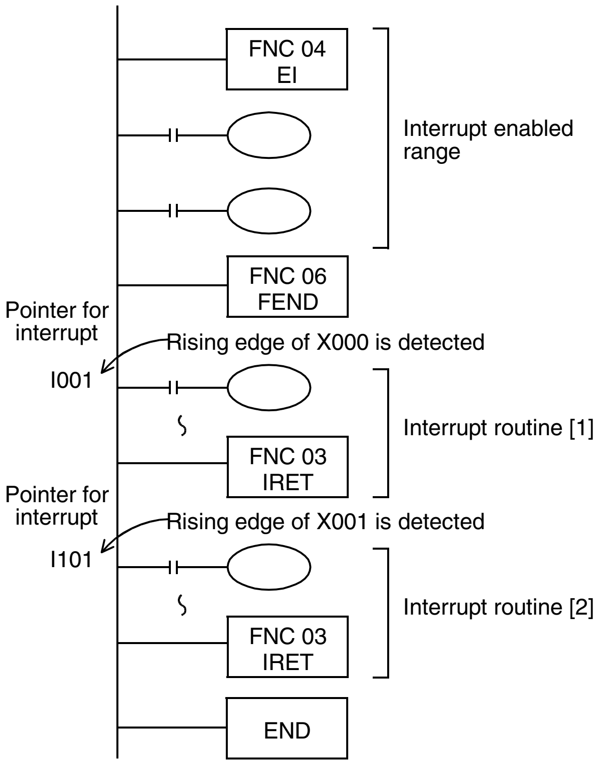
Delay function of input interrupt

This input interrupt has a function to delay the execution of interrupt routine in units of 1ms.
The delay time is specified by the following pattern program.
This delay function can electrically adjust the mounting position of sensors for input interrupts without shifting the actual position.



- Delay time specifying program
  Make sure to describe the delay time specifying program shown on the left at the head of an interrupt routine program.
  Because this is a pattern program, change only the delay time [1].
  Only a constant (K) or data register (D) can be used to specify the delay time*.
- Interrupt program is finished

Operations



- Interrupt is usually disabled in the PLC.
  If interrupt is enabled by EI instruction, when X000 or X001 turns ON while a program is scanned, the PLC executes the interrupt routine [1] or [2], and then returns to the main program by IRET instruction.
- Make sure to program a pointer for interrupt (I***) as a label after FEND instruction.

2. Pointers for timer interrupt: 3 points

→ **For details on timer interrupt function, refer to Section 35.5.**

The PLC executes an interrupt routine program at every specified interrupt cycle time (10 to 99 ms).
Use these pointers for control requiring cyclic processing regardless of the operation cycle of the PLC.

Input No.	Interrupt cycle (ms)	Interrupt disabling flag
I6□□	An integer in the range from 10 to 99 is put in "□□" portion of the pointer name. Ex: I610 = Timer interrupt at every 10 ms	M8056*1
I7□□		M8057*1
I8□□		M8058*1

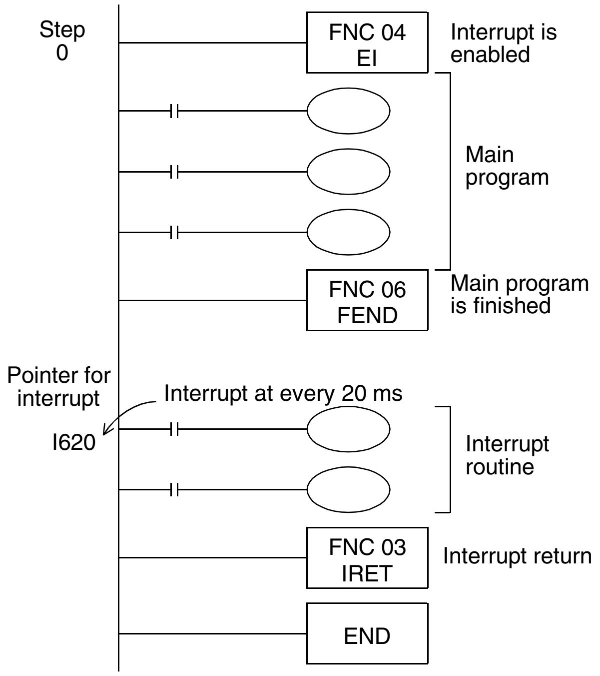
*1. Cleared when the PLC mode switches from RUN to STOP.

Caution

It is recommended to set the timer interrupt time to 10 ms or more. When the timer interrupt time is set to 9 ms or less, the timer interrupt processing may not be executed at an accurate cycle in the following cases:

- When the processing time of the interrupt program is long
- When an instruction requiring long processing time is used in the main program

Operations



- Timer interrupt is enabled after EI instruction. It is not necessary to program DI (disable interrupt) instruction when no zone to disable timer interrupt is needed.
- "FEND" indicates the end of the main program. Make sure to describe an interrupt routine after "FEND".
- The PLC executes an interrupt routine at every 20 ms. The PLC returns to main program by IRET instruction.

3. Pointers for counter interrupt: 6 points

→ **For details on counter interrupt function, refer to Section 35.6.**

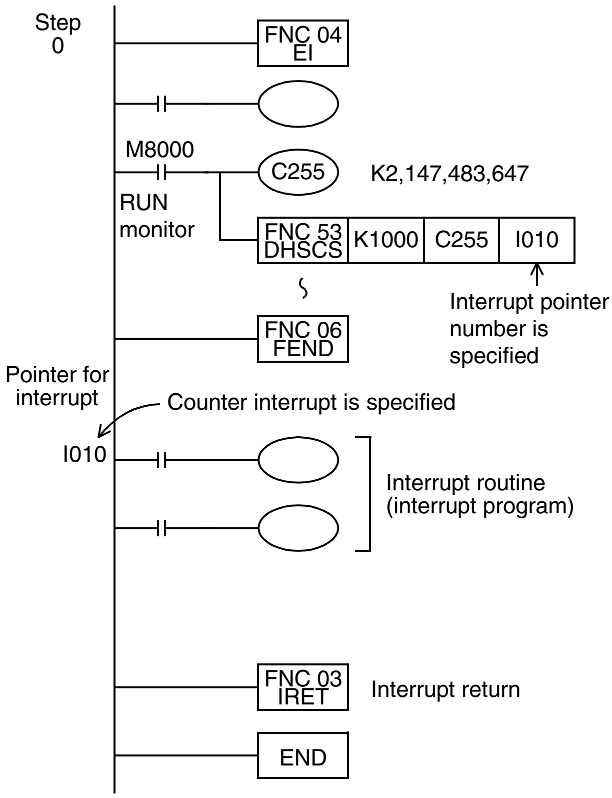
The PLC executes an interrupt routine based on the comparison result obtained by the comparison set instruction for high speed counter (DHSCS instruction).
Use these pointers for control requiring an interrupt routine based on the counting result from high speed counters.

Pointer No.	Interrupt disabling flag
I010	M8059*1
I020	
I030	

Pointer No.	Interrupt disabling flag
I040	M8059*1
I050	
I060	

*1. Cleared when the PLC mode switches from RUN to STOP.

Operations



- Enable interrupt after EI instruction, and describe the main program.
- Drive the coil of a high speed counter, and specify an interrupt pointer in DHSCS (FNC 53) instruction.
- When the current value of C255 changes from "999" to "1000" or from "1001" to "1000", the interrupt routine is executed.
  For example of interrupt program, refer to an input interrupt described above.

1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

5. How to Specify Devices and Constants to Instructions

This chapter explains how to specify sources and destinations in sequence instructions which are the basis for handling PLC instructions.

- Specifying constants as decimal, hexadecimal and real numbers
- Specifying digits of bit devices
- Specifying bit positions in data registers
- Directly specifying BFM (buffer memory) in special function blocks/units
- Indexing with index registers

5.1 Numeric Values Handled in PLCs (Octal, Decimal, Hexadecimal and Real Numbers)

FX PLCs handle five types of numeric values according to the application and purpose.
This section explains the roles and functions of these numeric values.

5.1.1 Types of numeric values

1. Decimal numbers (DEC)

- Set value (constant K) of timers and counters
- Device numbers of auxiliary relays (M), timers (T), counters (C), state relays (S), etc.
- Numeric values in operands and instruction operations in applied instructions (constant K)

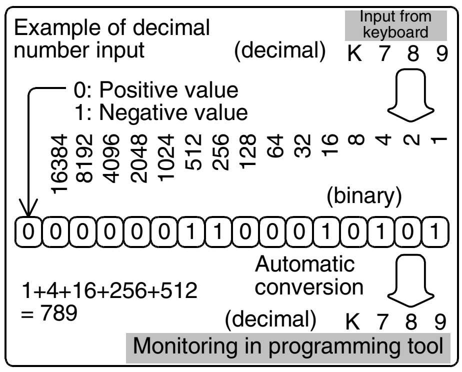
2. Hexadecimal numbers (HEX)

- Numeric values in operands and instruction operations in applied instructions (constant H)

3. Binary numbers (BIN)

For a timer, counter or data register, a numeric value is specified in decimal or hexadecimal as described above. But all of these numeric values are handled in the binary format inside PLCs.
When these devices are monitored in peripheral equipment, they are automatically converted into the decimal format as shown in the figure on the right (or can be converted into the hexadecimal format).

- Handling of negative value
  A negative value is expressed in complement of PLCs.
  For details, refer to the explanation of NEG (FNC 29) instruction.



4. Octal numbers (OCT)

In FX PLCs, device numbers of input relays and output relays are assigned in octal.
Because "8" and "9" do not exist in octal, device numbers are carried in the way "0 to 7, 10 to 17, 70 to 77, 100 to 107".

5. Binary coded decimal (BCD)

BCD format expresses each numeric value from 0 to 9 constructing each digit of a decimal number in a 4-bit binary number.
Because handling of each digit is easy, this format is adopted in controlling digital switches of BCD output type and seven-segment display units.

6. Real numbers (floating point data)

FX3U and FX3UC PLCs have the floating point operation function to achieve high accuracy operation.
In floating point operations, binary floating points (real numbers) are used, and scientific notation (real numbers) are used for monitoring them.

5.1.2 Conversion of numeric values

Numeric values handled in FX PLCs can be converted as shown in the table below:

Decimal number (DEC)	Octal number (OCT)	Hexadecimal number (HEX)	Binary number (BIN)		BCD	
0	0	00	0000	0000	0000	0000
1	1	01	0000	0001	0000	0001
2	2	02	0000	0010	0000	0010
3	3	03	0000	0011	0000	0011
4	4	04	0000	0100	0000	0100
5	5	05	0000	0101	0000	0101
6	6	06	0000	0110	0000	0110
7	7	07	0000	0111	0000	0111
8	10	08	0000	1000	0000	1000
9	11	09	0000	1001	0000	1001
10	12	0A	0000	1010	0001	0000
11	13	0B	0000	1011	0001	0001
12	14	0C	0000	1100	0001	0010
13	15	0D	0000	1101	0001	0011
14	16	0E	0000	1110	0001	0100
15	17	0F	0000	1111	0001	0101
16	20	10	0001	0000	0001	0110
⋮	⋮	⋮	⋮	⋮	⋮	⋮
99	143	63	0110	0011	1001	1001
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Major applications

Decimal number (DEC)	Octal number (OCT)	Hexadecimal number (HEX)	Binary number (BIN)	BCD
Constants (K) and numbers of internal devices except I/O relays	Numbers of internal I/O relays	Constants (H)	Processing inside PLC	BCD digital switches and seven-segment display units

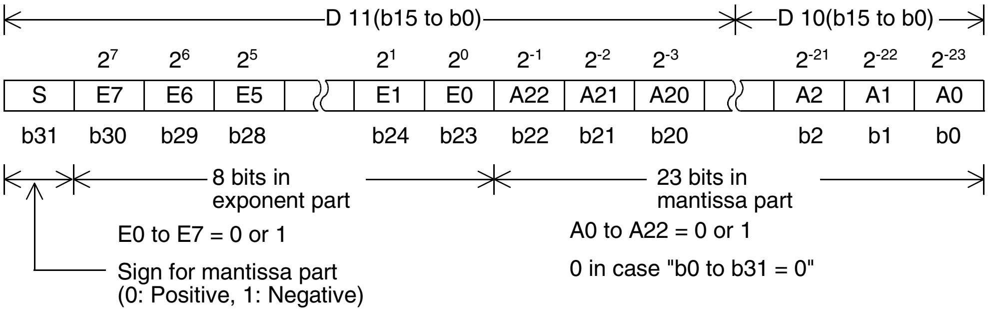
5.1.3 Handling of numeric values in floating point operations

Handling of numeric values in floating point operations

Binary integers are handled inside PLCs.
During division of integers, the answer "40 ÷ 3 = 13 ... 1" is obtained, for example.
During square root extraction operations, decimal points are ignored.
In FX3U and FX3UC PLCs, floating point operations are available to achieve higher accuracy in such operations.

Binary floating point (real number)

When handling a binary floating point (real number) in data registers, use a pair of data registers having consecutive device numbers.
When D11 and D10 are used, for example, a binary floating point is handled as shown below:



Binary floating point (real number) $= \pm (2^0 + A22 \times 2^{-1} + A21 \times 2^{-2} + ... + A0 \times 2^{-23})$
$\times 2^{(E7 \times 2^7 + E6 \times 2^6 + ... + E0 \times 2^0)} / 2^{127}$

Example: A22=1 , A21=0, A20=1, A19 to A0=0, E7=1, E6 to E1=0, E0=1

Binary floating point (real number) $= \pm (2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + ... + 0 \times 2^{-23})$
$\times 2^{(1 \times 2^7 + 0 \times 2^6 + ... + 1 \times 2^0)} / 2^{127}$
$= \pm 1.625 \times 2^{129}/2^{127} = \pm 1.625 \times 2^2$

The sign bit b31 states whether data is positive or negative, but is not handled as a complement.

Handling of the zero (M8020), borrow (M8021) and carry (M8022) flags
These flags operate as follows in floating point operations.

- Zero flag : 1 when the result is 0
- Borrow flag : 1 when the result does not reach the minimum unit but is not 0
- Carry flag : 1 when the absolute value of the result exceeds the available numeric value range.

Monitoring of binary floating point (real number)
A programming software supporting the display of floating point such as GX Developer can directly monitor binary floating point (real number).
A programming tool not supporting the display of floating point can monitor binary floating point (real number) when it is converted into scientific notation (real number).

Scientific notation (real number)

Because binary floating point (real number) is difficult to understand for users, it can be converted into scientific notation (real number). But internal operations are executed using binary floating point (real number).
Scientific notation (real number) is handled by a pair of data registers having serial device numbers. Different from binary floating point (real number), a data register having a smaller device number handles the mantissa part, and the other data register having a larger device number handles the exponent part.
For example, when data registers D1 and D0 are used, they handle scientific notation as shown below. Data can be written to D0 and D1 by MOV instruction.

Scientific notation (real number) = [Mantissa D0] $\times$ 10 [Exponent D1]
Mantissa D0 = $\pm$ (1000 to 9999) or 0
Exponent D1 = -41 to $+35$

The most significant bit of D0 and D1 specifies the positive or negative sign respectively, and is handled as the complement of 2 respectively.
The mantissa D0 does not allow "100", for example. In the case of "100", it is handled as "1000×10^{-1}".
The scientific notation (real number) range is as follows:

- Minimum absolute value: 1175×10^{-41}
- Maximum absolute value: 3402×10^{35}

Scientific notation (real number) is valid in the following instructions:

- Conversion from binary floating point (real number) into scientific notation (real number): FNC118 ([D]EBCD)
- Conversion from scientific notation (real number) into binary floating point (real number): FNC119 ([D]EBIN)

5.2 Specification of Constants K, H and E (Decimal, Hexadecimal and Real Number)

When handling constants in a sequence program, use constant K (decimal), H (hexadecimal) or E (floating point).
In peripheral equipment for programming, add "K" to a decimal number, "H" to a hexadecimal number and "E" to a floating point (real number) for operations associated with numeric values in instructions. (Examples: K100 (decimal number), H64 (hexadecimal number) and E1.23 (or E1.23 + 10) (real number))
The roles and functions of constants are described below.

5.2.1 Constant K (decimal number)

"K" indicates a decimal integer, and is mainly used to specify the set value of timers and counters and numeric values as operands in applied instructions. (Example: K1234)
The decimal constant specification range is as follows:

- When word data (16 bits) is used ... K-32768 to K32767
- When double data (32 bits) is used ... K-2,147,483,648 to K2,147,483,647

5.2.2 Constant H (hexadecimal number)

"H" indicates a hexadecimal number, and is mainly used to specify numeric values as operands in applied instructions. (Example: H1234)
When using digits 0 to 9, the bit status (1 or 0) of each bit is equivalent to the BCD code, so BCD data can be specified also.
(Example: H1234 ... When specifying BCD data, specify each digit of hexadecimal number in 0 to 9.)
The hexadecimal constant setting range is as follows:

- When word data (16 bits) is used ... H0 to HFFFF (H0 to H9999 in the case of BCD data)
- When double data (32 bits) is used ... H0 to HFFFFFFFF (H0 to H99999999 in the case of BCD data)

5.2.3 Constant E (real number)

"E" indicates a real number (floating point data), and is mainly used to specify numeric values as operands in applied instructions. (Example: E1.234 or E1.234 + 3)
The real number setting range is from -1.0×2^{128} to -1.0×2^{-126}, 0 and 1.0×2^{-126} to 1.0×2^{128}.
In a sequence program, a real number can be specified in two methods, "normal expression" and "exponent expression".

- Normal expression: .... Specify a numeric value as it is.
  For example, specify "10.2345" in the form "E10.2345".
- Exponent expression: .... Specify a numeric value in the format "(numeric value) $\times 10^n$".
  For example, specify "1234" in the form "E1.234 + 3".
  "+3" in "E1.234 + 3" indicates "10^3".

5.3 Character Strings

Character strings are classified into character string constants which directly specify character strings in operands in applied instructions and character string data.

5.3.1 Character string constant ("ABC")

A device "character string" directly specifies a character string in a sequence program.
Put half-width characters inside quotation marks (example: "ABCD1234") in specification.
JIS8 code is available.
Up to 32 characters can be specified as a character string.

5.3.2 Character string data

With regard to character string data, a specified device to the NUL code (00H) is handled as one character string in 1-byte units.
When expressing (recognizing) character string data by bit devices with digit specification, however, 16 bits are required for data including the NUL code (00H) specifying the end of the character string data because the instruction length is 16 bits. (Refer to Example 2 in the step 2 below.)
In the following cases, an operation error occurs in the applied instruction (error code: K6706):

- When "00H" is not specified in the corresponding device range after the source device number specified in an applied instruction
- When there are insufficient devices for storing character string data (including "00H" or "0000H" indicating the end of the character string data) in the destination devices specified in an applied instruction

1) Character string data stored in word devices

- Example of data which can be recognized as character string data

	b15 — b8	b7 — b0
D100	2nd character	1st character
D101	4th character	3rd character
D102	6th character	5th character
⋮		
D110	00H	21st character

"00H" indicating the end of the character string can be detected.

- Example of data which cannot be recognized as character string data

	b15 — b8	b7 — b0
D100	2nd character	1st character
D101	4th character	3rd character
D102	6th character	5th character
⋮		
D7999	"n"th character	"(n-1)"th character

"00H" indicating the end of character string cannot be detected from the specified device to the end device number.

2) Character string data stored in bit devices with digit specification

- Example of data which can be recognized as character string data

	16 bits	
M115 to M100	2nd character	1st character
M131 to M116	4th character	3rd character
M147 to M132	6th character	5th character
⋮		
M211 to M196	00H	13th character

"00H" indicating the end of the character string can be detected.

- Examples of data which cannot be recognized as character string data

<Example 1>

	16 bits	
M115 to M100	2nd character	1st character
M131 to M116	4th character	3rd character
M147 to M132	6th character	5th character
⋮		
M7679 to M7664	"n"th character	"(n-1)"th character

"00H" indicating the end of character string cannot be detected from the specified device to the end device number.

<Example 2>

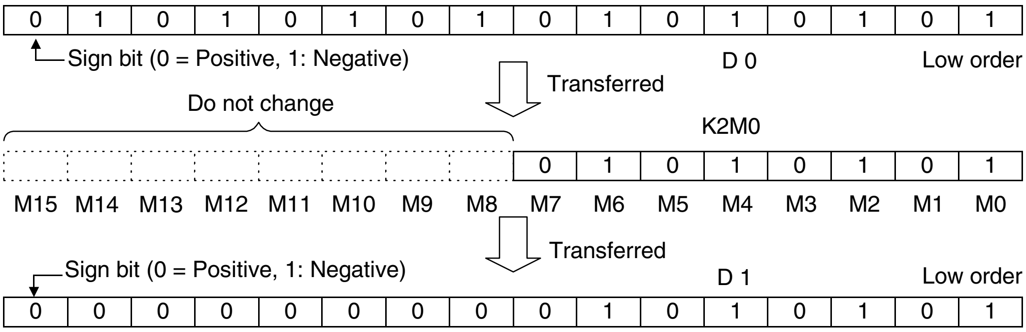
	16 bits	
M7623 to M7608	2nd character	1st character
M7639 to M7624	4th character	3rd character
M7655 to M7640	6th character	5th character
M7671 to M7656	8th character	7th character
M7679 to M7672		00H

Because the data "00H" indicating the end of the character string does not reach 16 bits, the end of the character string cannot be recognized.

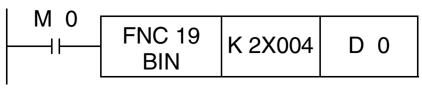
5.4 Specification of Digits for Bit Devices (Kn[ ]***)

Handling of bit devices

Devices which handle only the ON/OFF information such as X, Y, M and S are called bit devices.
On the other hand, devices handling numeric values such as T, C, D and R are called word devices.
Even bit devices can handle a numeric value when they are combined. In this case, the number of digits Kn and the head device number are combined.
The number of digits is expressed in 4 bit units (digits); K1 to K4 are used for 16-bit data, and K1 to K8 are used for 32-bit data.
For example, "K2M0" indicates two-digit data expressed by M0 to M7.



When 16-bit data is transferred to K1M0 to K3M0, the highest-order bits are not transferred due to insufficient data length.
32-bit data is transferred in the same way.
When the number of digits specified for bit devices is K1 to K3 (or K1 to K7) in a 16-bit (or 32-bit) operation, the insufficient high-order bits are always regarded as "0". It means that such data is always positive.



Two-digit BCD data expressed by X004 to X013 is converted into binary data, and then transferred to D0.

A bit device number can be specified arbitrarily, but it is recommended to set the least significant digit to "0" for X or Y. (In other words, it is recommended to specify "X000, X010, X020 ... Y000, Y010, Y020 ...")
For M and S, multiples of "8" are ideal, but it is recommended to specify "M0, M10, M20 ..." to prevent confusion.

Specification of consecutive words

A series of data registers starting from D1 means "D1, D2, D3, D4 ....."
In the case of word devices with digit specification, when such word devices are handled as a series, they are specified as shown below:

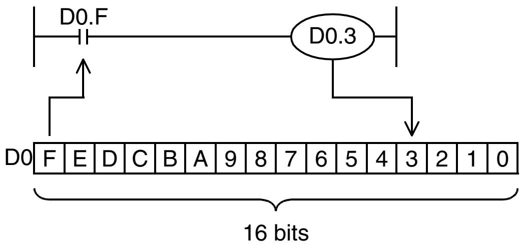
- K1X000, K1X004, K1X010, K1X014 .....
- K2Y010, K2Y020, K2X030 .....
- K3M0, K3M12, M3M24, K3M36 .....
- K4S16, K4S32, K4S48 .....

Use the above devices in digit units so that devices are not skipped.
When "K4Y000" is used in a 32-bit operation, the high-order 16 bits register as "0".
It is necessary to use "K8Y000" when 32-bit data is required.

5.5 Bit Specification of a Word Device (D[ ].b)

By specifying a bit of a word device, the specified bit can be used as bit data.
When specifying a bit of a word device, use a word device number and bit number (hexadecimal).
(Example: D0.0 ... Indicates the bit 0 of data register (D).)
Indexing is not available for both device numbers and bit numbers.

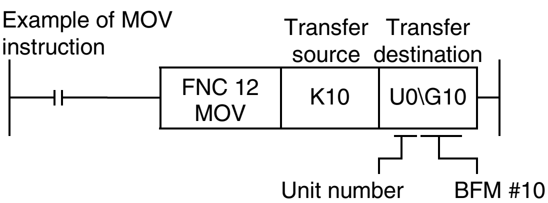
Target word device : Data register or special data register
Bit number : 0 to F (hexadecimal)

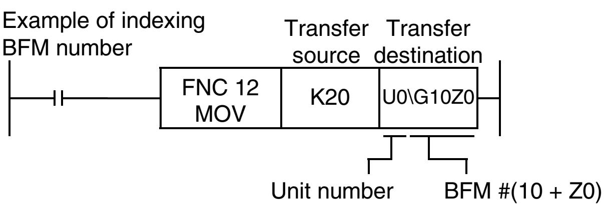


5.6 Direct Specification of Buffer Memory (U[ ]\G[ ])

A buffer memory (BFM) of a special function block or special extension unit can be specified directly.
BFM is 16-bit or 32-bit word data, and is mainly used for operands in applied instructions.
For specifying a BFM, specify the unit number (U) of a special function block or special extension unit and the BFM number (\G) consecutively.
(Example: U0\G0 ... Indicates the BFM #0 in the special function block or special extension unit whose unit number is 0.)
Indexing is available for BFM numbers.
The specification range is as follows:

Unit number (U)................. 0 to 7
BFM number (\G) .... 0 to 32766





5.7 Indexing

The functions and structures of index registers are explained in detail in "4.10 Index Register [V and Z]". Refer to Section 4.10 in advance.

5.7.1 Indexing in basic instructions

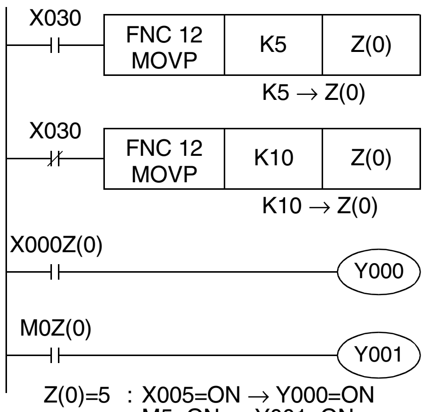
In the case of bit devices

Bit devices [X, Y, M (except special auxiliary relays), T, and C (C0 to C199)] used in LD, LDI, AND, ANI, OR, ORI, OUT, SET, RST, PLS, and PLF instructions can be indexed with index registers.

The figure shown on the right explains an indexing operation with the index register Z(0) for X000 and M0 in the LD instruction.
Transfer K5 or K10 to the index register Z(0) in advance.
If Z(0) is "5", "X(0+5) = X005". When X005 turns ON, Y000 turns ON and "M(0+5) = M5". When M5 turns ON, Y001 turns ON.
If Z(0) is "10", "X(0+10) = X012*1". When X012*1 turns ON, Y000 turns ON and "M(0+10) = M10". When M10 turns ON, Y001 turns ON.

*1. Refer to the caution 3) below.

- The index registers Z0 to Z7 and V0 to V7 can be used for indexing.
- In OUT instruction for a timer or counter, the timer number (or counter number) and the device specified for the set value can be indexed.



Z(0)=5 : X005=ON → Y000=ON
M5=ON → Y001=ON
Z(0)=10 : X012*2=ON → Y000=ON
M10=ON → Y001=ON

*2 Refer to the caution 3).

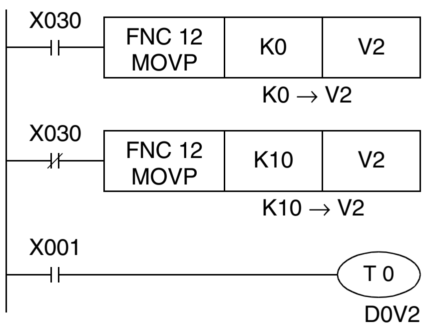
Cautions

1) 32-bit counters and special auxiliary relays cannot be indexed with index registers.
2) It is not permitted to use 16-bit counters as 32-bit counters by executing indexing.
3) When an octal device number of X or Y is indexed with an index register, the contents of the index register are converted into octal, and then added to the device number.
For example, when the value of an index register added to the input X000 is changed in the order "K0 → K8 → K16", the device number converted into octal is added to the input X000 and the input number is changed in the order "X(000+0) = X000 → X(000+8) = X10 → X(000+16) = X20".

In the case of word devices and constants

The set value of bit devices [T and C] used in OUT instruction can be indexed with index registers.

The indexing operation is explained in an example in which the set value D0 of T0 used in the index register V2 indexes OUT instruction(as shown in the right figure).
Transfer K0 or K10 to the index register V2 in advance.
When X001 is set to ON, "D(0+0) = D0" if V2 is "0", and T0 operates with the set value D0.
When X001 is set to ON, "D(0+10) = D10" if V2 is "10", and T0 operates with the set value D10.



V2 = 0 : The set value of T0 is the present value of D0.
V2 = 10 : The set value of T0 is the present value of D10.

Caution

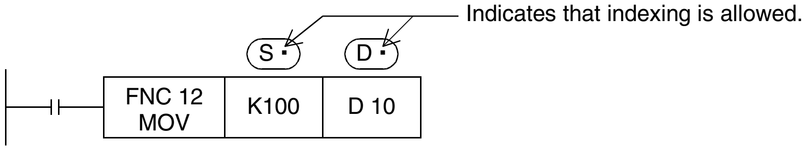
1) When a 32-bit counter is used in OUT instruction, the set value cannot be indexed with an index register.

1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

5.7.2 Indexing in applied instructions

Expression of applied instructions allowing indexing

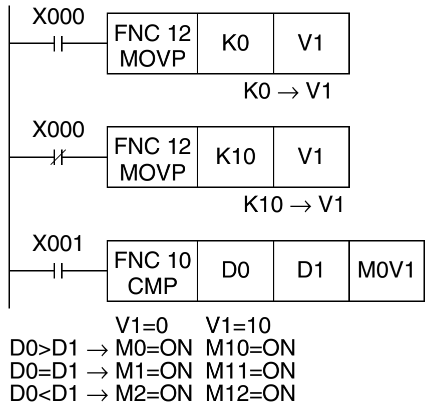
In the explanation of applied instructions, " ▪ " is added to the source (S) or destination (D) symbol to indicate operands allowing indexing as shown in the figure below so that such operands can be discriminated from operands not allowing indexing.



In the case of bit devices

The indexing operation is explained in an example in which the comparison result M0 in CMP (FNC 10) instruction is indexed with the index register V1 (as shown in the figure on the right).
Transfer K0 or K10 to the index register V1 in advance.
When X001 is set to ON, "M(0+0) = M0" and the comparison result is output to M0 to M2 if V1 is "0".
On the other hand, "M(0+10) = M10" and the comparison result is output to M10 to M12 if V1 is "10".

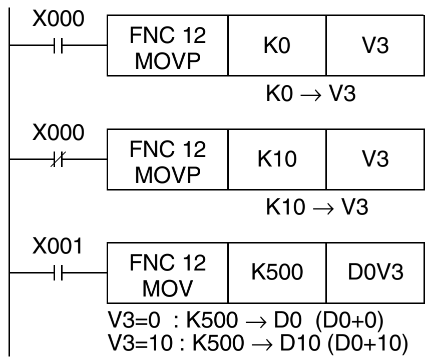
- The index registers Z0 to Z7 and V0 to V7 can be used for indexing.



In the case of word devices

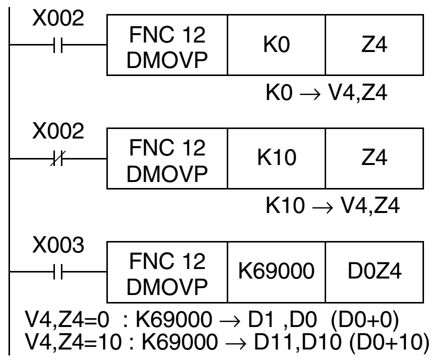
1. **indexing operands in 16-bit instructions**

   The indexing operation is explained in an example in which the transfer destination D0 in MOV instruction is indexed with the index register V3 (as shown in the figure on the right).
   Transfer K0 or K10 to the index register V3 in advance.
   When X001 is set to ON, "D(0+0) = D0" if V3 is "0", and K500 is transferred to D0.
   When X001 is set to ON, "D(0+10) = D10" if V3 is "10", and K500 is transferred to D10.



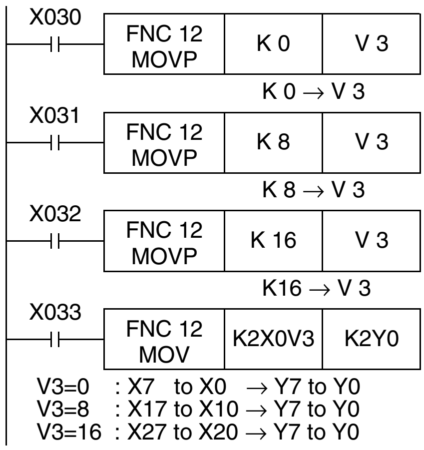
2. **indexing operands in 32-bit instructions**

   In a 32-bit instruction, it is also necessary to specify a 32-bit index register in the instruction.
   When an index register Z (Z0 to Z7) is specified in a 32-bit instruction, the specified Z and its counterpart V (V0 to V7) work together as 32-bit registers.
   The indexing operation is explained in an example in which the transfer destinations [D1, D0] in DMOV instruction are indexed with the index registers [V4, Z4] (as shown in the figure on the right).
   Transfer K0 or K10 to the index registers [V4, Z4] in advance.
   When X003 is set to ON, "[D(1+0), D(0+0)] = [D1, D0]" is realized if [V4, Z4] is "0", and K69000 is transferred to [D1, D0].
   When X003 is set to ON, "[D(1+10), D(0+10)] = [D11, D10]" is realized if [V4, Z4] is "10", and K69000 is transferred to [D11, D10].



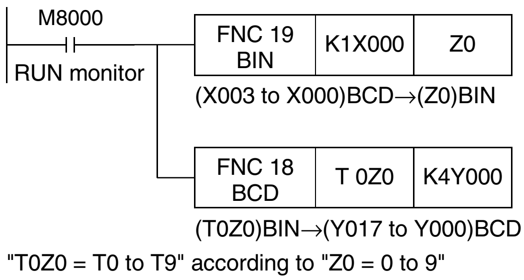
Cautions

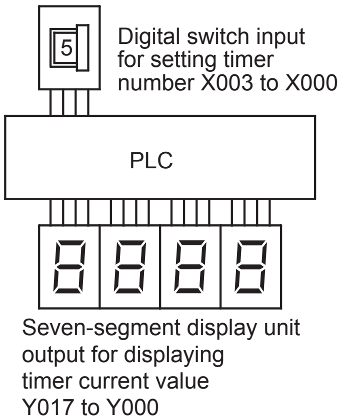
1) When even if a numeric value written to index registers does not exceed the 16-bit numeric value range (0 to 32767), make sure to overwrite both V and Z using a 32-bit instruction. If only Z is overwritten and another numeric value remains in V, the numeric value will be extremely large. Thus an operation error occurs.
2) It is not permitted to use 16-bit counters as 32-bit counters by executing indexing.
   When 32-bit counters are required, add Z0 to Z7 to counters C200 and later.
3) It is not permitted to index V and Z themselves.
4) Direct specification of buffer memory in special function blocks/units
   In the direct specification of buffer memory "U□\G□", the buffer memory number can be indexed with index registers.
   The unit number cannot be indexed with index registers.
   ("U0\G0Z0" is valid, but "U0Z0\G0" is invalid.)
5) Indexing in bit digit specification
   It is not permitted to index "n" in "Kn" used for digit specification.
   ("K4M0Z0" is valid, but "K0Z0M0" is invalid.)
6) Indexing of I/O relays (octal device numbers)
   When octal device numbers of X, Y, KnX, and KnY are indexed with index register, the contents of an index register are converted into octal, and then added to the device number.
   In the example shown in the figure on the right, Y007 to Y000 are output by MOV instruction, and inputs are switched by indexing X007 to X000, X017 to X010, and X027 to X020.
   When rewriting the index value as "K0", "K8", "K16", the device number converted into octal is added "X000 + 0 = X000", "X000 + 8 = X10", "X000 + 16 = X20", and the input terminal working as the source is changed accordingly.



Display example of timer present value

A sequence to display the present value of the timers T0 to T9 can be programmed index registers.

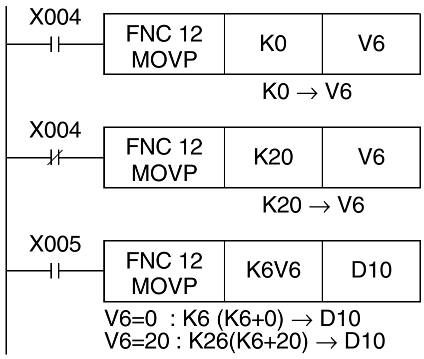




1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

In the case of constants

The indexing operation is explained in an example in which the transfer destination in MOV instruction is indexed with the index register V6 (as shown in the figure on the right).
Transfer K0 or K20 to the index register V6 in advance.
When X005 is set to ON, "K(6+0) = K6" if V6 is "0", and K6 is transferred to D10.
When X005 is set to ON, "K(6+20) = K26" if V6 is "20", and K26 is transferred to D10.

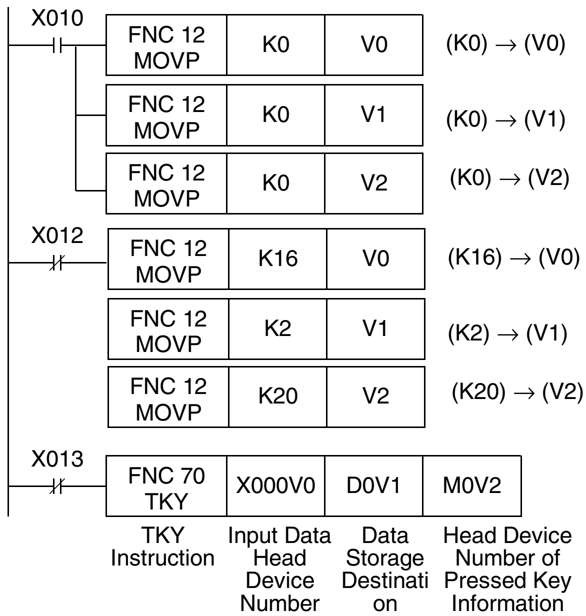


5.7.3 Indexing example for instruction with limited number of use.

By modifying the target device numbers using index registers V and Z, the target device numbers can be changed using the program. In this way, an instruction with a limited number of uses per program can be used with multiple devices.

Example using the TKY instruction (FNC 70)

Two groups of key entries (numeric keypad from 0 to 9) store the input data to D0 and D2. Although the TKY instruction (FNC 70) can only be programmed once, modifying the head device number of the input data, storage destination and pressed key information, the information can be input from the two groups of keys (numeric keypad from 0 to 9). Furthermore, even if V is changed while this instruction is being executed, this change is invalid. The change is invalid until the instruction is no longer being driven.

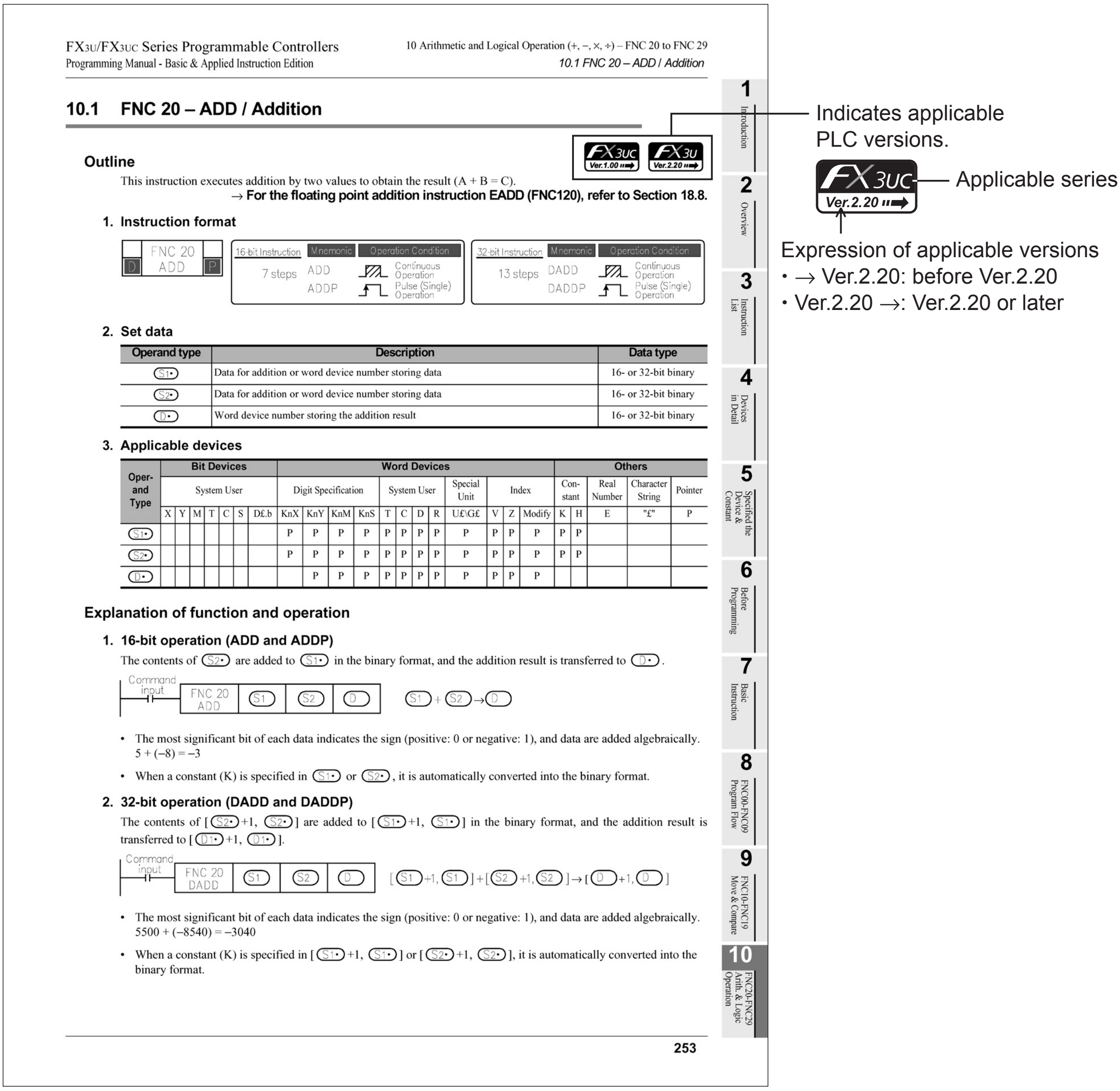


6. What to Understand before Programming

This chapter explains the I/O processing, relationship among instructions and programming method which should be understood before creating sequence programs.

6.1 How to Read Explanation of Instructions

In this manual, applied instructions are explained in the following form.
For the expression methods and basic rules for applied instructions, read in advance "6.5 General rules for applied instructions" described later.



FX3U/FX3UC Series Programmable Controllers
Programming Manual - Basic & Applied Instruction Edition

10 Arithmetic and Logical Operation (+, −, ×, ÷) – FNC 20 to FNC 29
10.1 FNC 20 – ADD / Addition

10.1 FNC 20 – ADD / Addition

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction executes addition by two values to obtain the result (A + B = C).
→ **For the floating point addition instruction EADD (FNC120), refer to Section 18.8.**

1. Instruction format

D FNC 20 ADD P	16-bit Instruction	Mnemonic	Operation Condition	32-bit Instruction	Mnemonic	Operation Condition
	7 steps	ADD	Continuous Operation	13 steps	DADD	Continuous Operation
		ADDP	Pulse (Single) Operation		DADDP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S1•	Data for addition or word device number storing data	16- or 32-bit binary
S2•	Data for addition or word device number storing data	16- or 32-bit binary
D•	Word device number storing the addition result	16- or 32-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D£.b	KnX	KnY	KnM	KnS	T	C	D	R	U£\G£	V	Z	Modify	K	H	E	"£"	P
S1•								P	P	P	P	P	P	P	P	P	P	P	P	P	P			
S2•								P	P	P	P	P	P	P	P	P	P	P	P	P	P			
D•									P	P	P	P	P	P	P	P	P	P	P					

Explanation of function and operation

1. 16-bit operation (ADD and ADDP)

The contents of S2• are added to S1• in the binary format, and the addition result is transferred to D•.

Command input — FNC 20 ADD | S1 | S2 | D   S1 + S2 → D

- The most significant bit of each data indicates the sign (positive: 0 or negative: 1), and data are added algebraically.
5 + (−8) = −3
- When a constant (K) is specified in S1• or S2•, it is automatically converted into the binary format.

2. 32-bit operation (DADD and DADDP)

The contents of [S2•+1, S2•] are added to [S1•+1, S1•] in the binary format, and the addition result is transferred to [D•+1, D•].

Command input — FNC 20 DADD | S1 | S2 | D   [S1+1, S1] + [S2+1, S2] → [D+1, D]

- The most significant bit of each data indicates the sign (positive: 0 or negative: 1), and data are added algebraically.
5500 + (−8540) = −3040
- When a constant (K) is specified in [S1•+1, S1•] or [S2•+1, S2•], it is automatically converted into the binary format.

253

Outline

1. Instruction format

1) The applied instruction number (FNC No.) and instruction mnemonic are indicated. The table below shows the meaning of simplified expression.

Mark	Description	Applicable instruction (example)
FNC No. Instruction name	Dotted lines on the upper left and lower left sides indicate an independent instruction not associated with the 16-bit or 32-bit type.	WDT(FNC 07)
D FNC No. Instruction name	Continuous lines on the upper left side indicates that 16-bit type is available. "D" on the lower left side indicates that the 32-bit type is available.	MOV(FNC 12)
FNC No. Instruction name	Dotted lines on the lower left side indicate that the 32-bit type does not exist. Continuous lines on the upper left side indicate that only the 16-bit type is available.	CJ(FNC 00)
D FNC No. Instruction name	Dotted lines on the upper left side indicate that the 16-bit type does not exist. "D" on the lower left side indicates that only the 32-bit type is available.	HSCS(FNC 53)
FNC No. Instruction name P	Continuous lines on the upper right side indicate that the continuous operation type is available. "P" on the lower right side indicates that the pulse operation type is available.	CMP(FNC 10)
FNC No. Instruction name	Dotted lines on the lower right side indicate that the pulse operation type does not exist. Continuous line on the upper right side indicate that only the continuous operation type is available.	MTR(FNC 52)
FNC No. Instruction name P	" "on the upper right side indicates that the contents of the destination change in every operation cycle when the continuous operation type is used. When operation should be executed only during the driving of an instruction, use the pulse operation type indicated by "P" on the lower right side.	INC(FNC 24)

2. Set data

The contents of devices that can be specified as operands in instructions and available data types are described below:

1) Contents
   The contents of operands in each instruction are described below.
2) Indexing of the source and destination
   In operands to which " · " is added such as S· and S1·, indexing is available.
   Operands not allowing indexing are expressed as S and S1.
3) Data types
   - Bit : Bit device
   - 16-bit BIN : 16-bit binary code
   - 32-bit BIN : 32-bit binary code
   - 64-bit BIN : 64-bit binary code
   - 16/32-bit BIN : 16-bit or 32-bit binary code
   - 32/64-bit BIN : 32-bit or 64-bit binary code
   - 4-digit BCD : 4-digit (16-bit) BCD code
   - 8-digit BCD : 8-digit (32-bit) BCD code
   - 4/8-digit BCD : 4-digit (16-bit) or 8-digit (32-bit) BCD code
   - Character string : Character code such as ASCII code and shift JIS code
   - Character string (only ASCII) : ASCII code
   - Real number (binary) : Binary floating point
   - Real number (decimal) : Scientific notation

Applicable devices

Devices which can be specified in operands of instructions are shown.
When a device supports an instruction, "✓" is added to the device.

1) Bit devices
- •X : Input relay (X)
- •Y : Output relay (Y)
- •M : Auxiliary relay (M)
- •S : State relay (S)

etc.

2) Word devices
- •K : Decimal integer
- •H : Hexadecimal integer
- •KnX: Input relay (X) with digit specification*1
- •KnY: Output relay (Y) with digit specification*1
- •KnM: Auxiliary relay (M) with digit specification*1
- •KnS: State relay (S) with digit specification*1
- •T : Timer (T) current value
- •C : Counter (C) current value
- •D : Data register (file register)
- •V, Z: Index register
- •Modify: Availability of indexing using index register

etc.

*1. Kn without specification indicates K1 to K4 for 16 bits, and K1 to K8 for 32 bits.

Explanation of function and operation

The function of each instruction is explained.

Cautions

Cautions on using each instruction are described.

Errors

Major errors that are possible to occur in each instruction are described.
For details on errors, refer to "Chapter 37. Errors and Error Code List".

Program examples

Concrete program examples using each instruction are described.

6.2 Cautions on Creation of Fundamental Programs

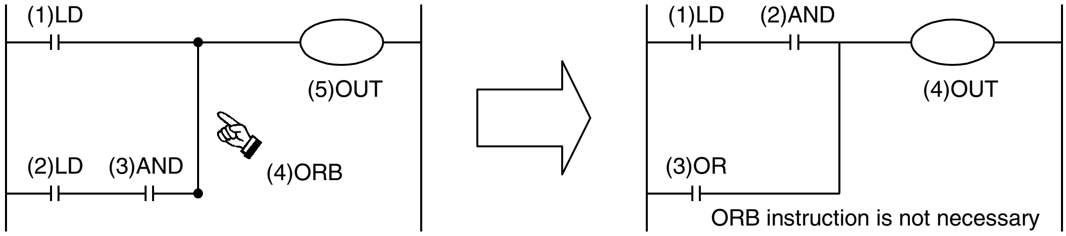
This section explains cautions on programming.

6.2.1 Programming procedure and execution order

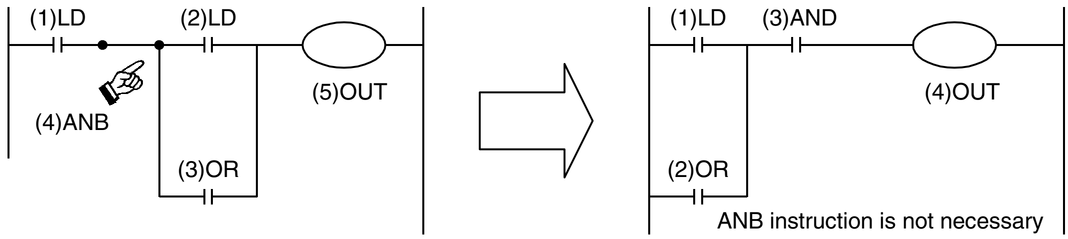
1. Contact configuration and steps

Even for a sequence circuit offering a same operation, the program can be simplified and the number of steps can be saved depending on the contact configuration method.

1) It is recommended to write a circuit with many serial contacts in an upper position.

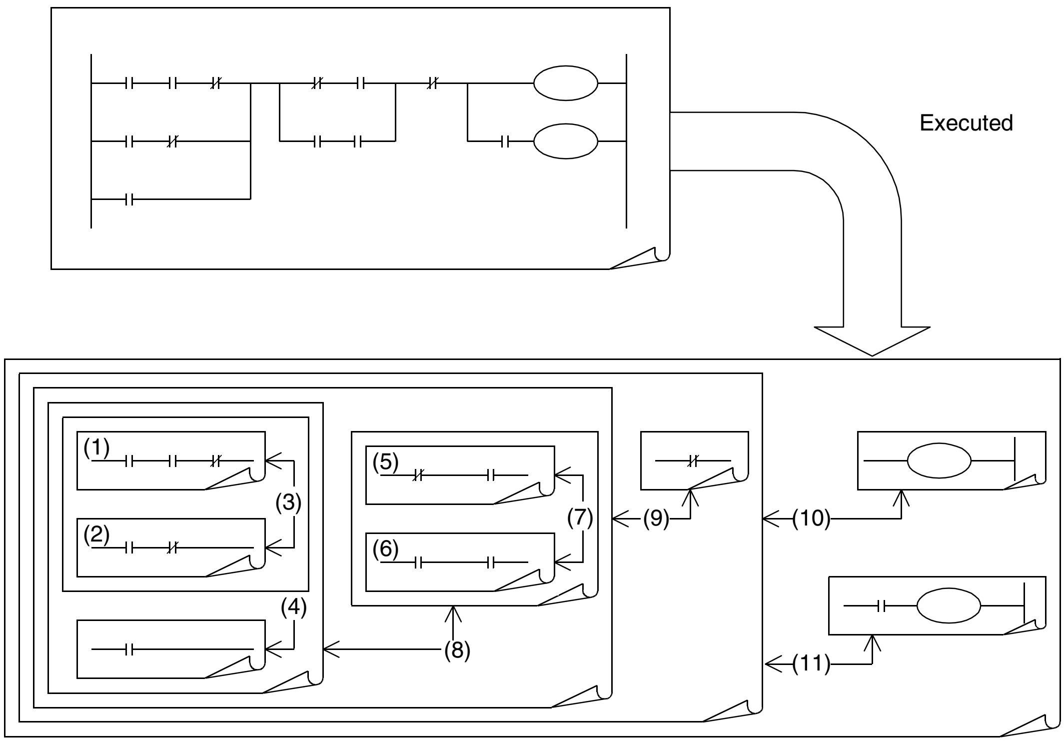


2) It is recommended to write a circuit with many parallel contacts in a left position.



2. Program execution and programming order

A sequence program is executed "from top to bottom" and "from left to right".
Code the sequence instruction list according to this rule.



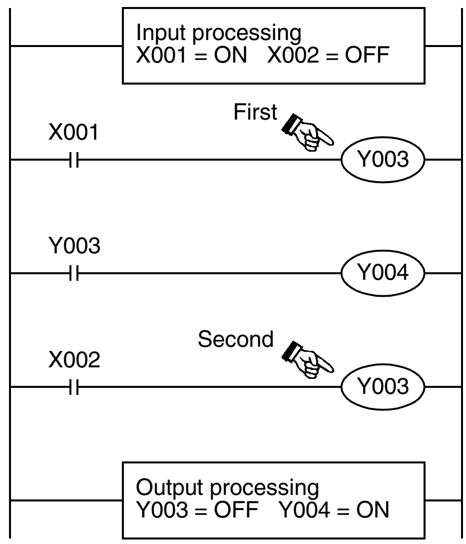
6.2.2 Double output (double coil) operation and countermeasures

1. Operation of double outputs

When a coil gives double outputs (double coils) in a sequence program, the priority is given to the latter one.

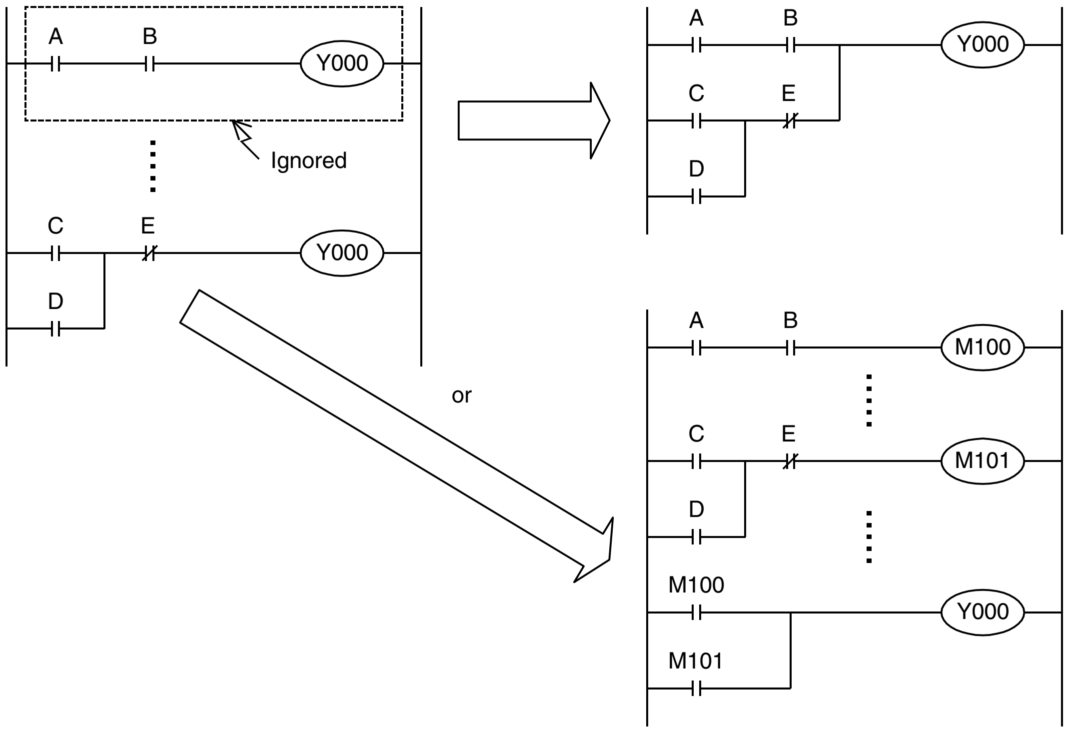
Suppose that the same coil Y003 is used in two positions as shown in the figure on the right.
For example, suppose the X001 is ON and X002 is OFF.

In the first coil Y003, the image memory turns ON and the output Y004 turns ON also because the input X001 is ON.
In the second coil Y003, however, the image memory is set to OFF because the input X002 is OFF.
Accordingly, the actual output to the outside is "Y003 = OFF, Y004 = ON".



2. Countermeasures against double outputs

Double outputs (double coils) do not cause illegal input (program error), but the operation is disrupted as described above.
Change the program as shown in the example below.

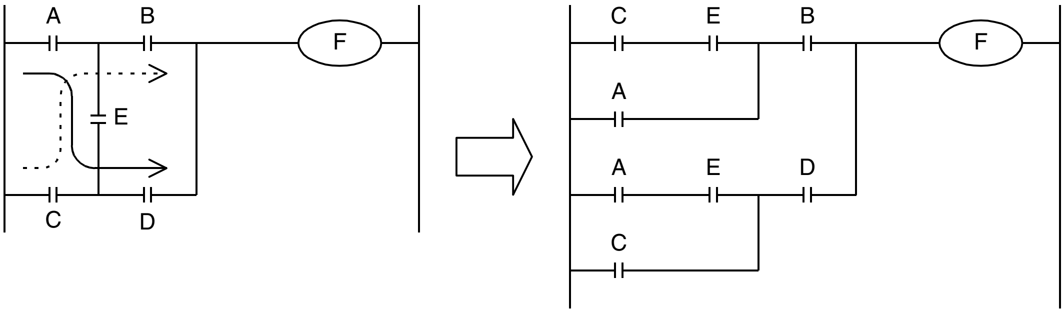


SET, RST or jump instruction can be used instead, or a same output coil can be programmed at each state by step ladder instructions.
When step ladder instructions are used, if an output coil located in the main routine is also used in a state, it is handled as a double coil. It is better to avoid such programming.

6.2.3 Circuits which cannot be programmed and countermeasures

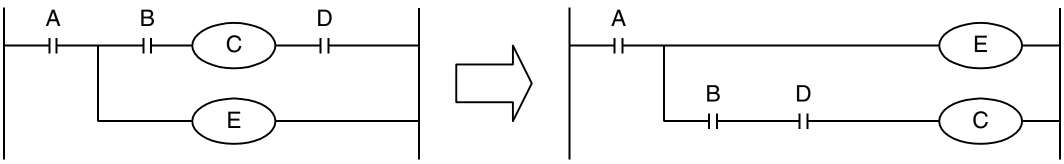
1. Bridge circuit

A circuit in which the current flows in both directions should be changed as shown in the figure on the right (so that a circuit without D and a circuit without B are connected in parallel).



2. Coil connection position

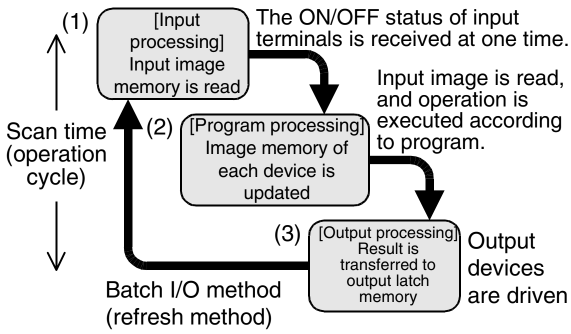
- Do not write a contact on the right side of a coil.
- It is recommended to program a coil between contacts first.
  The number of steps can be saved when a coil (E) between the contacts A and B is programmed first.



6.3 I/O Processing and Response Delay

1. Operation timing of I/O relays and response delay

FX PLCs execute the I/O processing by repeating the process (1) to process (3).
Accordingly, the control executed by PLCs contains not only the drive time of input filters and output devices but also the response delay caused by the operation cycle.



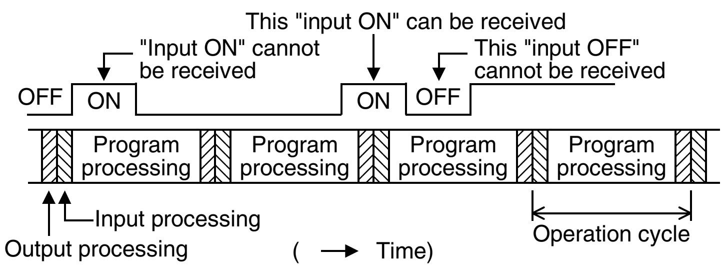
Acquiring the latest I/O information
For acquiring the latest input information or immediately outputting the operation result in the middle of the operation cycle shown above, the I/O refresh instruction is available.

2. Short pulses cannot be received.

The ON duration and OFF duration of inputs in PLCs require longer time than "PLC cycle time + Input filter response delay".
When the response delay of the input filter "10 ms" is considered and the cycle time is supposed as "10 ms", the ON duration and OFF duration should be at least 20 ms respectively.
Accordingly, PLCs cannot handle input pulses at 25 Hz (1000 / (20 + 20) = 25) or more. However, the situation can be improved by PLC special functions and applied instructions.

Convenient functions for improvement
By using the following functions, PLCs can receive pulses shorter than the operation cycle:

- High speed counter function
- Input interrupt function
- Pulse catch function
- Input filter value adjustment function



6.4 Mutual Relationship Among Program Flow Control Instructions

The table below shows the mutual relationship among various program flow control instructions.

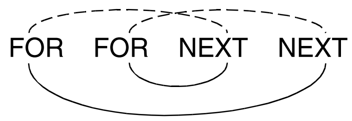
In the table below, " ◎ " indicates containment relationship, and " ⚭ " indicates that zones are partially overlapped.

Instruction in top line
Instruction in left line ◎
Example 1: MC — CJ — P — MCR

Instruction in top line
Instruction in left line ⚭
Example 2: MC — CJ — MCR — P

Left line \ Top line		MC-MCR	CJ-P	EI-DI	FOR-NEXT	STL-RET
MC-MCR	◎	✓ octet	✓ Example 1	✓	✓	✓
	⚭	×	△ Example 2	✓	×	×
CJ-P	◎	✓	✓	✓	✓	✓
	⚭	△	△	✓	△	△
EI-DI	◎	✓	✓	✓	✓	✓
	⚭	✓	✓	✓	✓	✓
FOR-NEXT	◎	×	✓	✓	✓ quintet	×
	⚭	×	△	✓	*2	×
STL-RET	◎	×	△	✓	✓ (inside one STL)	×
	⚭	×	△	✓	×	×
P-SRET	◎	×	✓	✓	✓	×
	⚭	×	△	✓	×	×
I-IRET	◎	×	✓	✓	✓	×
	⚭	×	△	✓	×	×
FEND-END	◎	✓	✓	✓	✓	△
	⚭	×	×	*1	×	×
O-FEND	◎	✓	✓	✓	✓	✓
	⚭	×	✓	✓	×	×
O-END (no FEND)	◎	✓	✓	✓	✓	✓
	⚭	×	×	*1	×	×

✓:This combination can be used without any problem.
×:This combination is not allowed; Operation error will be occurs.
△:This combination is allowed, but is better not to be used because the operation will be complicated.

P-SRET	I-IRET	FEND-END	Remarks
◎ ×	◎ ×	◎ ×	*1 The DI skip status occurs, but this is not an error.
⚭ ×	⚭ ×	⚭ ×	
◎ △	◎ △	◎ ×	*2 
⚭ △	⚭ △	⚭ ✓	
◎ ✓	◎ ✓	◎ *1	The operation indicated by continuous lines is discribed.
⚭ ✓	⚭ ✓	⚭ ✓	*3 The first FEND or END is valid, but the intended processes will not occur. But this is not an error.
◎ ×	◎ ×	◎ ×	
⚭ ×	⚭ ×	⚭ ×	
◎ ×	◎ ×	◎ ×	
⚭ ×	⚭ ×	⚭ ×	
◎ ×	◎ ×	◎ ×	Instructions having containment relationship can be combined except some combinations as follows:
⚭ ×	⚭ ×	⚭ ×	1) MC-MCR cannot be used in FOR-NEXT, STL-RET, P-SRET and I-RET.
◎ ×	◎ ×	◎ ×	
⚭ ×	⚭ ×	⚭ ×	2) STL-RET cannot be used in FOR-NEXT, P-SRET and I-IRET.
◎ ✓	◎ ✓	◎ *3	3) MC-MCR, FOR-NEXT, P-SRET and I-IRET cannot be interrupted by I, IRET, SRET, FEND, END, etc.
⚭ ×	⚭ ×	⚭ *3	
◎ ×	◎ ×	◎ *3	
⚭ ×	⚭ ×	⚭ *3	
◎ ×	◎ ×	◎ *3	
⚭ ×	⚭ ×	⚭ *3	

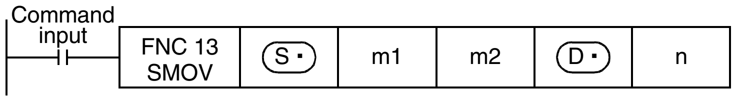
1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

6.5 General Rules for Applied Instructions

6.5.1 Expression and operation type of applied instructions

Instructions and operands

- Both a function number FNC 00 to FNC □□□ and a symbol (mnemonic) indicating the contents are given to each applied instruction.
  For example, a mnemonic "SMOV (shift move)" is assigned to FNC 13 instruction.
- Some applied instructions function only with their instruction part, but many instructions consist of the instruction part and following operands.



S : An operand whose contents do not change by execution of the instruction is called "source", and is indicated by this symbol.
When a device number can be indexed with index registers, the source is expressed as S· with addition of " · ".
When there are two or more sources, they are expressed as S1·, S2·, etc.

D : An operand whose contents change by execution of the instruction is called "destination", and indicated by this symbol.
When indexing is allowed and there are two or more destinations, they are expressed as D1·, D2·, etc. in the same way as sources.

m, n : Operands not falling under source or destination are expressed as m and n.
When indexing is allowed and there are two or more such operands, they are expressed as m1 ·, m2 ·, n1 ·, n2 ·,etc. in the same way as sources and destinations.

- In applied instructions, the program step of the instruction part always occupies 1 step, but each operand occupies 2 or 4 steps depending on whether the instruction is 16-bit type or 32-bit type.

Devices handled as operands

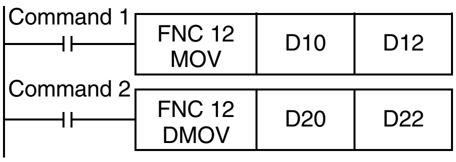
- Bit devices themselves such as X, Y, M and S may be handled.
- Combined bit devices, KnX, KnY, KnM, KnS, etc, may be handled as numeric value data.
  → **Refer to Section 5.4.**
- Data registers D and current value registers for timers T and counters C may be handled.
- Though data registers D are the 16-bit type, two serial data registers are combined when 32-bit data is handled.
  For example, when a data register D0 is specified as an operand in a 32-bit instruction, D1 and D0 are combined to handle 32-bit data. (D1 handles high-order 16 bits, and D0 handles low-order 16 bits.)
  When current value registers for T and C are used as general data registers, they are handled in the same way.
  However, each of 32-bit counters C200 to C255 can handle 32-bit data, and cannot be specified as an operand in a 16-bit instruction.

Instruction form and operation type

Applied instructions are classified into "16-bit type" or "32-bit type" by the size of handled numeric values. And by the operation type, applied instructions are classified into "continuous operation type" or "pulse operation type".
Some applied instructions have every combination of this form and type, and others do not.

1. 16-bit type and 32-bit type

- Applied instructions handling numeric values are classified into the 16-bit type or the 32-bit type by the bit length of the numeric value data.



This instruction transfers the contents of D10 to D12.

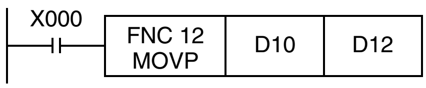
This instruction transfers the contents of D21 and D20 to D23 and D22.

- In a 32-bit type instruction, the symbol "D" is added (example: DMOV).
- Either an odd or even device number can be specified, and a specified device is combined with a device having the subsequent larger number (in the case of word devices such as T, C and D).
  For avoiding confusion, it is recommended to specify an even device number (which will be the low-order side) for an operand in a 32-bit instruction.
- 32-bit counter (C200 to C255) is regarded as 32 bits, and cannot be used as an operand in a 16-bit instruction.

2. Pulse operation type and continuous operation type

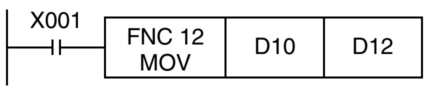
Pulse operation type

In the example shown in the figure on the right, when X000 turns ON from OFF, the instruction is executed only once, and is not executed in any other case.
When it is not necessary to continually execute an instruction, use the pulse operation type.
The symbol "P" indicates the pulse operation type.
"DMOVP" indicates also the pulse operation type.



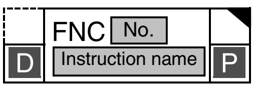
Continuous operation type

The figure on the right shows a continuous operation type instruction. While X001 is ON, the instruction is executed in every operation cycle.



In the continuous operation type of some instructions such as FNC 24 (INC) and FNC 25 (DEC), the contents of the destination change in every operation cycle.

For applied instructions requiring attention in using the continuous operation type, the symbol " ◥ " is added to the title of the explanation of such instructions as shown in the figure below.



In any case, instructions are not executed while the drive input X000 or X001 is OFF. And the destinations do not change except when instructions specify otherwise.

6.5.2 Handling of general flags

In some types of applied instructions, the following flags operate:
Examples: M8020: Zero flag M8021: Borrow flag M8022: Carry flag
M8029: Instruction execution complete flag
M8090: Block comparison signal M8328: Instruction non-execution flag
M8329: Instruction execution abnormal complete flag
M8304: Zero Flag M8306:Carry Flag

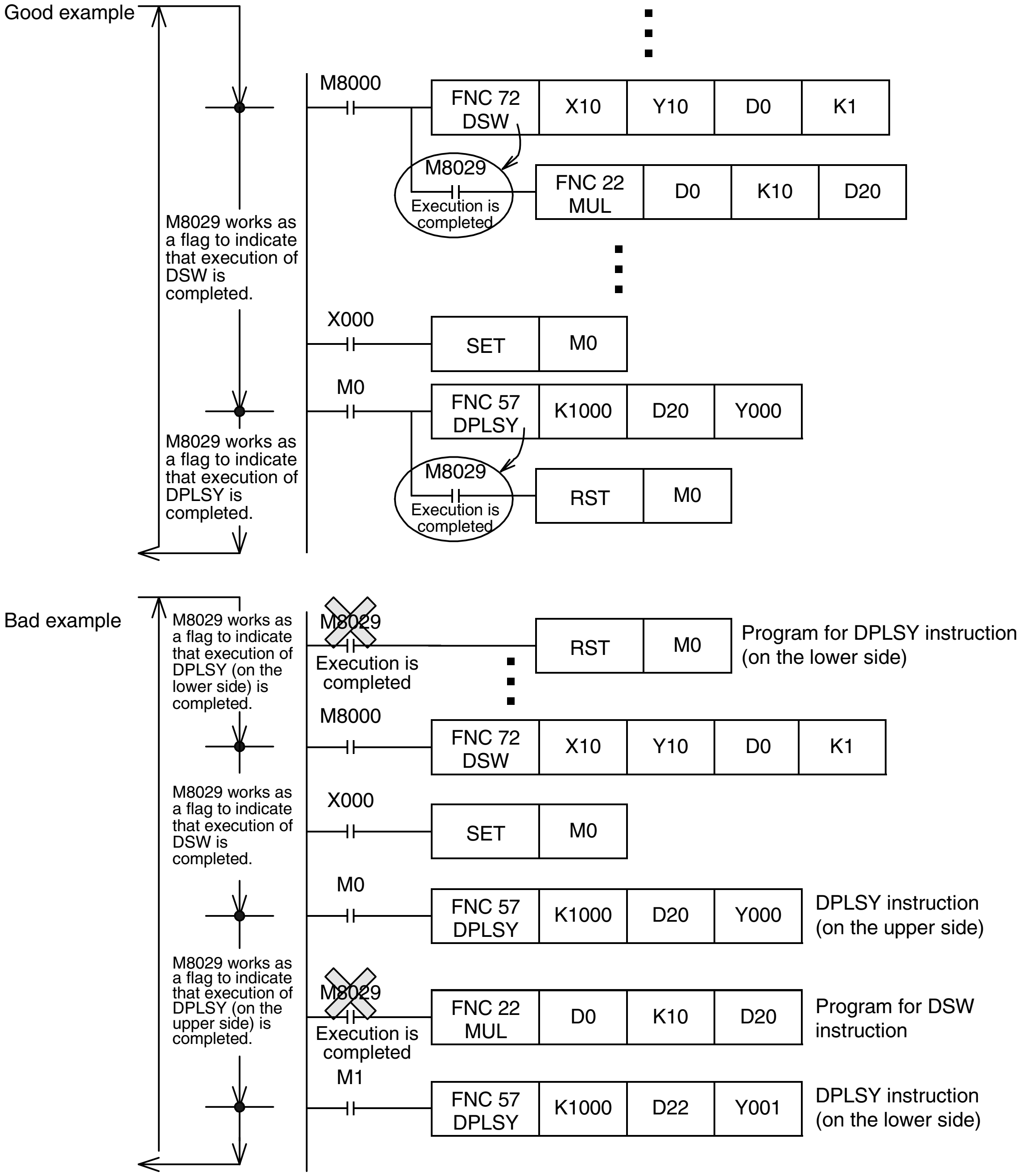
These flags turn ON or OFF every time various instructions turn ON, but do not change when various instructions turn OFF not driven or when errors have occurred.

Because these flags turn ON or OFF in many instructions, the ON/OFF status of flags change every time such instructions are executed.

Program flag contacts directly under each instruction while referring to the examples below.

1. Program containing many flags (example of instruction execution complete flag M8029)

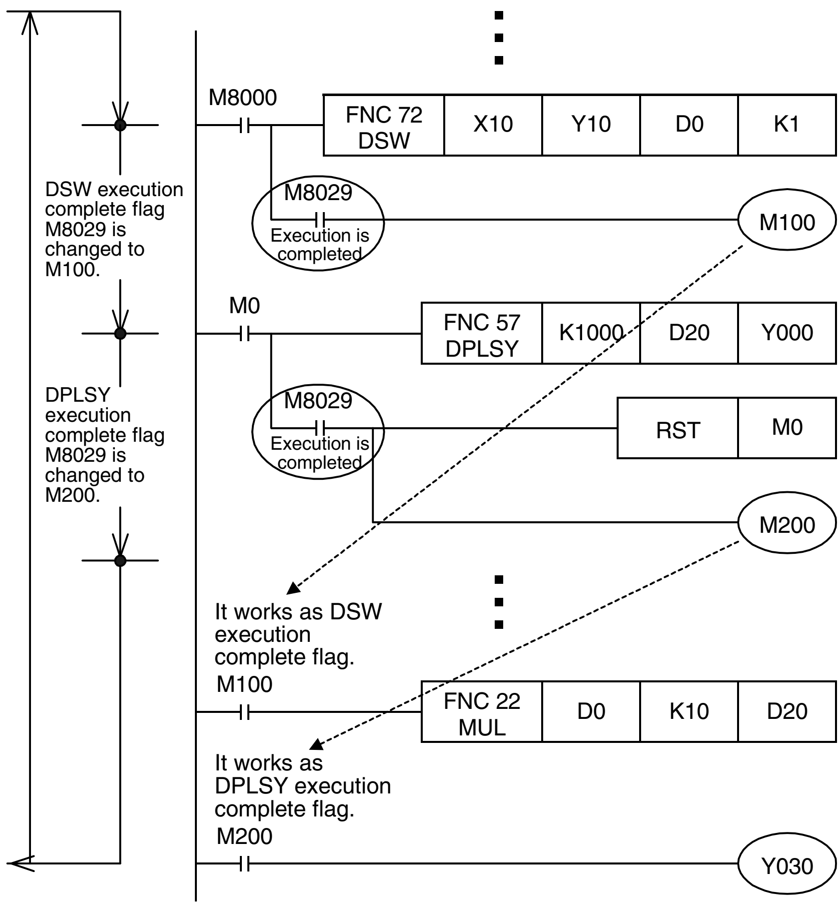
When two or more instruction execution complete flags M8029 are programmed together for applied instructions, it is difficult to determine which instruction executes which flag.
For using flags in any positions other than directly under applied instructions, refer to the next page.



2. Introduction of method for using flags in any positions other than directly under applied instructions

When two or more applied instructions are programmed, general flags turn ON or OFF when each applied instruction turns ON.

Accordingly, when using a flag in any position other than directly under an applied instruction, set to ON or OFF another device just under the applied instruction, and then use the contact of the device as the command contact.



6.5.3 Handling of operation error flag

When there is an error in the applied instruction configuration, target device or target device number range and an error occurs while operation is executed, the following flag turns ON and the error information is stored.

1. Operation error

Error flag	Device storing error code	Device storing error occurrence step	Device storing error occurrence step*1
M8067	D8067	D8315, D8314	D8069

*1. Compatible with the FX0, FX0S, FX0N, FX1, FX2, FX2C, FX1S, FX1N, FX2N, FX1NC and FX2NC Series.

- When an operation error has occurred, M8067 is set, D8067 stores the operation error code number, and D8315 and D8314 (32 bits in total) store the step number in which the error has occurred.
- When the error occurrence step is up to the 32767th step, the error occurrence step can be checked in D8069 (16 bits).
- If another error occurs in another step, the stored data is updated in turn to the error code and step number of the new error. (These devices are set to OFF when errors are cleared.)
- When the PLC mode switches from STOP to RUN, these devices are cleared instantaneously, and then set to ON again if errors have not been cleared.

2. Operation error latch

Error flag	Error code storing device	Error occurrence step storing device (32 bits)	Error occurrence step storing device*1
M8068	–	D8313, D8312	D8068

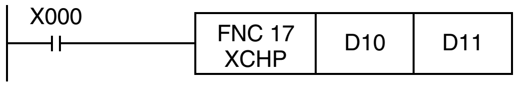
*1. Compatible with the FX0, FX0S, FX0N, FX1, FX2, FX2C, FX1S, FX1N, FX2N, FX1NC and FX2NC Series.

- When an operation error has occurred, M8068 is set, and D8313 and D8312 store the step number in which the error has occurred.
- Even if another error has occurred in another step, the stored data is not updated, and remains held until these devices are forcibly reset or until the power turns OFF.
- When the error occurrence step is up to the 32767th step, the error occurrence step can be checked in D8068 (16 bits).

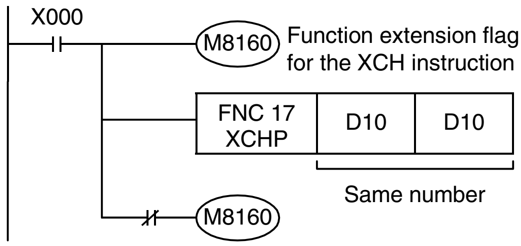
6.5.4 Handling functions of extension flag

In some applied instructions, the function can be extended by combining a specific special auxiliary relay determined for each applied instruction. An example is explained below.

- When X000 turns ON, this instruction exchanges the contents of D10 and D11 with each other.



- If M8160 has been driven before the XCH instruction and the source and destination of the XCH instruction are specified to the same device, high-order 8 bits and low-order 8 bits are exchanged with each other inside the device.
- For returning this XCH instruction to the normal XCH instruction, it is necessary to set M8160 to OFF.



When using an instruction requiring the function extension flag in an interrupt program, program DI instruction (for disabling interrupt) before driving the function extension flag, and program EI instruction (for enabling interrupt) after turning OFF the function extension flag.

6.5.5 Limitation in number of instructions

Limitation in the number of instructions and limitation in simultaneous driving

Some applied instructions can be used only up to the specified number of times.

Instruction name	Allowable number of times of use	Remarks
FNC 52(MTR)	1	MTR instruction can only be used once in program.
FNC 56(SPD)	8 (1 instruction/1 input or less)	Pay attention so that this instruction does not overlap the input numbers of in DVIT instruction, DOG inputs in ZRN instruction, zero point signal in DSZR instruction, input interrupt numbers and high speed counter input numbers.
FNC 60(IST)	1	–
FNC 69(SORT)	1	–
FNC 70(TKY)	1	–
FNC 71(HKY)	1	–
FNC 75(ARWS)	1	–
FNC 77(PR)	2	PR instruction can only be used twice in a program.
FNC149(SORT2)	2	–
FNC186(DUTY)	5 (1 instruction/1 input or less)	–
FNC280(HSCT)	1	–

When using above instructions beyond the allowable number of times of use

For instructions whose operands allow indexing, device numbers and numeric values in such instructions can be changed by index registers.

By indexing, when driving multiple instances simultaneously is not required, such instruction can be used as if they were used beyond the allowable number of times.

→ **Refer to "Subsection 5.7.3. Indexing example in instruction with limited number of use".**

Limitation in simultaneous instances of instructions

Some applied instructions can be programmed two or more times, but the number of simultaneous instances is limited.
Even in instructions not shown below, if two or more instructions are driven at the same time for a same I/O number, it is regarded as double outputs. In some combinations of instructions, the operation may be disrupted, or the instructions cannot be executed.
For details, refer to the caution described in each instruction page.
For combinations of instructions, refer to "6.4 Mutual relationship among program flow control instructions".

1. Positioning instructions

Do not drive FNC 57 (PLSY), FNC 58 (PWM), FNC 59 (PLSR), FNC150 (DSZR), FNC151 (DVIT), FNC156 (ZRN), FNC157 (PLSV), FNC158 (DRVI) and FNC159 (DRVA) instructions at the same time for the same output number.

2. High speed processing instructions

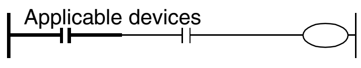
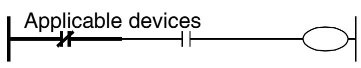
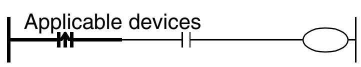
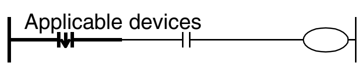
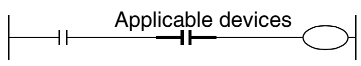
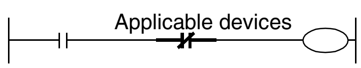
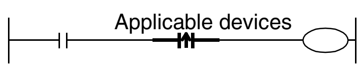
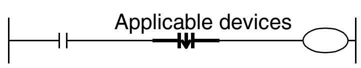
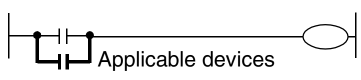
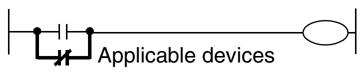
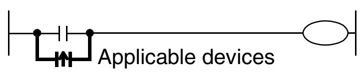
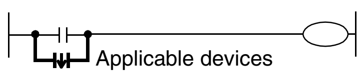
In FNC 53 (HSCS), FNC 54 (HSCR) and FNC 55 (HSZ) instructions (including FNC280 (HSCT) instruction), make sure that up to 32 instructions are driven at the same time. [FNC280 (HSCT) instruction can be used only once.]
Note that "FNC280 (HSCT) instruction", "table high speed comparison mode of FNC 55 (HSZ) instruction)" and "frequency control mode of FNC 55 (HSZ) instruction" can each only be used once.

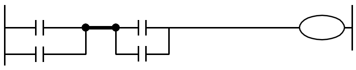
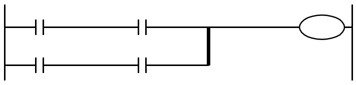
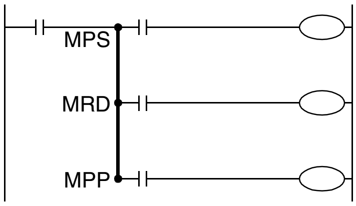
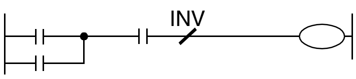
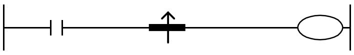
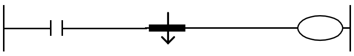
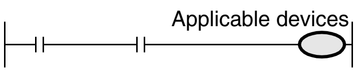
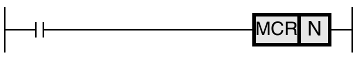
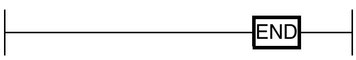
3. External device communication instructions

- In FNC 80 (RS) and FNC 87 (RS2) instructions, do not drive two or more instructions at the same time for the same port.
- It is impossible to combine and use FNC 80 (RS), FNC 87 (RS2), FNC270 (IVCK), FNC271 (IVDR), FNC272 (IVRD), FNC273 (IVWR) and FNC274 (IVBWR) instructions for the same port.
- In FNC270 (IVCK), FNC271 (IVDR), FNC272 (IVRD), FNC273 (IVWR) and FNC274 (IVBWR) instructions, two or more instructions can be driven at the same time for the same port.

7. Basic Instruction

This chapter explains types and functions of basic sequence instructions.
For beginners to sequence control, we offer "Introduction Course" and "Relay Ladder Course" learning texts for reference.
We can also offer the PLC learning software "Beginner Course".

Mnemonic	Name	Symbol	Function	Applicable devices	Reference
Contact Instruction					
LD	Load		Initial logical operation contact type NO (normally open)	X,Y,M,S,D□.b,T,C	Section 7.1
LDI	Load Inverse		Initial logical operation contact type NC (normally closed)	X,Y,M,S,D□.b,T,C	Section 7.1
LDP	Load Pulse		Initial logical operation of rising edge pulse	X,Y,M,S,D□.b,T,C	Section 7.5
LDF	Load Falling Pulse		Initial logical operation of falling/trailing edge pulse	X,Y,M,S,D□.b,T,C	Section 7.5
AND	AND		Serial connection of NO (normally open) contacts	X,Y,M,S,D□.b,T,C	Section 7.3
ANI	AND Inverse		Serial connection of NC (normally closed) contacts	X,Y,M,S,D□.b,T,C	Section 7.3
ANDP	AND Pulse		Serial connection of rising edge pulse	X,Y,M,S,D□.b,T,C	Section 7.5
ANDF	AND Falling Pulse		Serial connection of falling/trailing edge pulse	X,Y,M,S,D□.b,T,C	Section 7.5
OR	OR		Parallel connection of NO (normally open) contacts	X,Y,M,S,D□.b,T,C	Section 7.4
ORI	OR Inverse		Parallel connection of NC (normally closed) contacts	X,Y,M,S,D□.b,T,C	Section 7.4
ORP	OR Pulse		Parallel connection of rising edge pulse	X,Y,M,S,D□.b,T,C	Section 7.5
ORF	OR Falling Pulse		Parallel connection of falling/trailing edge pulse	X,Y,M,S,D□.b,T,C	Section 7.5

Mnemonic	Name	Symbol	Function	Applicable devices	Reference
Connection Instruction					
ANB	AND Block		Serial connection of multiple parallel circuits	–	Section 7.7
ORB	OR Block		Parallel connection of multiple contact circuits	–	Section 7.6
MPS	Memory Point Store		Stores the current result of the internal PLC operations	–	Section 7.8
MRD	Memory Read		Reads the current result of the initial PLC operations		Section 7.8
MPP	Memory POP		Pops (recalls and removes) the currently stored result		Section 7.8
INV	Inverse		Invert the current result of the internal PLC operations	–	Section 7.10
MEP	MEP		Conversion of operation result to leading edge pulse	–	Section 7.11
MEF	MEF		Conversion of operation result to trailing edge pulse	–	Section 7.11
Out Instruction					
OUT	OUT		Final logical operation type coil drive	Y,M,S,D□.b,T,C	Section 7.2
SET	SET		Set bit device latch ON	Y,M,S,D□.b	Section 7.13
RST	Reset		Reset bit device OFF	Y,M,S,D□.b,T,C, D,R,V,Z	Section 7.13
PLS	Pulse		Rising edge pulse	Y,M	Section 7.12
PLF	Pulse Falling		Falling/trailing edge pulse	Y,M	Section 7.12
Master Control Instruction					
MC	Master Control		Denotes the start of a master control block	Y,M	Section 7.9
MCR	Master Control Reset		Denotes the end of a master control block	–	Section 7.9
Other Instruction					
NOP	No Operation		No operation or null step	–	Section 7.14
End Instruction					
END	END		Program end, I/O refresh and return to step 0	–	Section 7.15

7.1 LD, LDI

Outline

LD and LDI instructions are contacts connected to bus lines.
When combined with ANB instruction described later, LD and LDI instructions can be used for the start of branches.

1. Instruction format

LD
Load

Basic Instruction	Mnemonic	Operation Condition
	LD	Continuous Operation
	–	Pulse (Single) Operation

LDI
Load Inverse

Basic Instruction	Mnemonic	Operation Condition
	LDI	Continuous Operation
	–	Pulse (Single) Operation

→ For the number of instruction steps, refer to Section 7.15.

2. Applicable devices

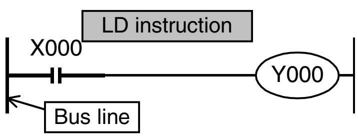
Instruction	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
LD	✓	✓	▲1	✓	▲1	▲2	▲2												✓					
LDI	✓	✓	▲1	✓	▲1	▲2	▲2												✓					

▲1: Special auxiliary relays (M) and 32-bit counters (C) cannot be indexed with index registers (V and Z).
▲2: State relays (S) and "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

1. LD instruction (initial logical operation, NO contact type)

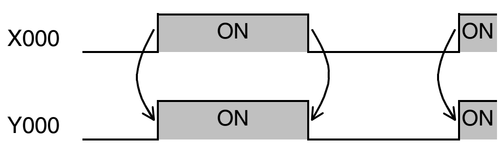
Circuit program



List program

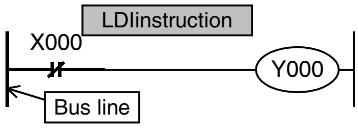
```
0000  LD   X000  ←  Connection to bus line
0001  OUT  Y000
```

Timing chart



2. LDI instruction (initial logical operation, NC contact type)

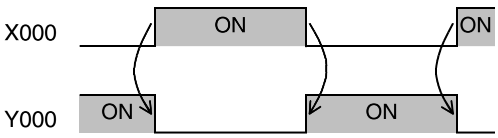
Circuit program



List program

```
0000 LDI  X000  ← Connection to bus line
0001 OUT  Y000
```

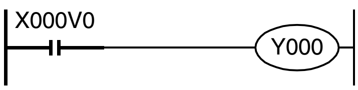
Timing chart



3. Indexing

Devices used in LD and LDI instructions allow indexing with index registers (V and Z).
(State relays (S), special auxiliary relays (M), 32-bit counters (C), and "D□.b" cannot be indexed.)

Circuit program



List program

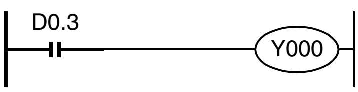
```
0000 LD   X000V0
0001 OUT  Y000
```

V0 to V7 and Z0 to Z7 are available in indexing.
When used devices are inputs (X) and outputs (Y), values of index registers (V and Z) are converted into octal numbers, and then added.
Example:When the value of V0 is 10, LD contact is set to ON (becomes conductive) or OFF (becomes non-conductive) by X012.

4. Bit specification of data register (D)

A bit in data register (D) can be specified as a device used in LD and LDI instructions.

Circuit program



List program

```
0000 LD   D0.3
0001 OUT  Y000
```

When specifying a bit in data register, input "." after a data register (D) number, and then input a bit number (0 to F) consecutively.
Only 16-bit data registers are available.
Specify a bit number as "0, 1, 2, ... 9, A, B, ... F" from the least significant bit.
Example:In the example shown on the left, LD contact is set to ON (becomes conductive) or OFF (becomes non-conductive) by the bit 3 of D0.

Errors

- When an I/O number used in LD or LDI instruction does not exist due to indexing, M8316 (Non-existing I/O specification error) turns ON.
- When the device number of a device (M, T or C) other than I/O used in LD or LDI instruction does not exist due to indexing, an operation error (error code: 6706) occurs.

7.2 OUT

Outline

OUT instruction drives coils of output relays (Y), auxiliary relays (M), state relays (S), timers (T) and counters (C).

1. Instruction format

OUT

OUT

Basic Instruction	Mnemonic	Operation Condition
	OUT	Continuous Operation
	–	Pulse (Single) Operation

→ **For the number of instruction steps, refer to Section 7.15.**

2. Applicable devices

Instruction	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
OUT		✓	▲1	✓	▲1	▲2	▲2												✓					
Set value														✓	✓				✓	✓	✓			

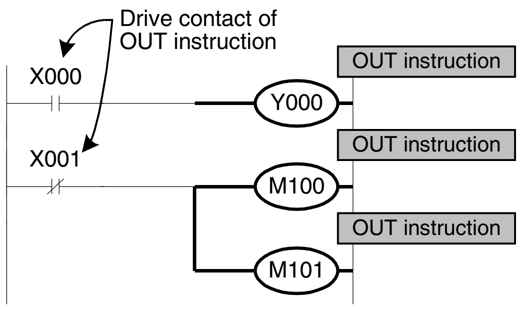
▲1: Special auxiliary relays (M) and 32-bit counters (C) cannot be indexed with index registers (V and Z).
▲2: State relays (S) and "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

1. When a bit device is used

A device described in OUT instruction turns ON or OFF according to the driven contact status.
Parallel OUT instructions can be used consecutively as many times as necessary.
In the program example shown below, OUT M100 and OUT M101 are parallel.
If two or more OUT instructions are executed for a same device number, however, the double output (double coil) operation is resulted.

Circuit program

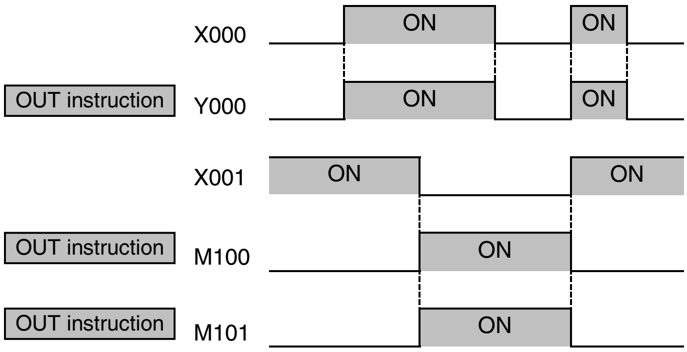


List program

```
0000 LD   X000
0001 OUT  Y000
0002 LDI  X001
0003 OUT  M100
0004 OUT  M101
```

Program step numbers are automatically controlled.

Timing chart

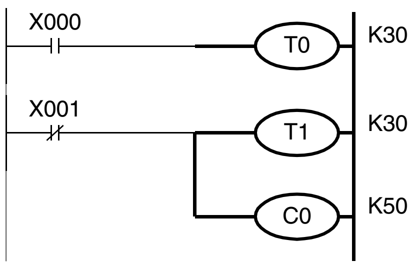


2. When a timer or counter is used

The set value is required after OUT instruction for the counting coil of a timer or counter.
The set value can be specified directly by a decimal number (K) or indirectly using a data register (D) or extension register (R).

1) Direct specification

Circuit program



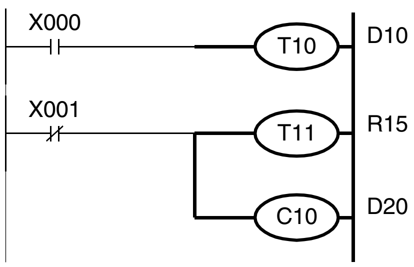
List program

```
0000 LD    X000
0001 OUT   T0
     (SP)  K30
0004 LDI   X001
0005 OUT   T1
     (SP)  K30
0008 OUT   C0
     (SP)  K50
```

The set value of a timer or counter can be specified directly by a decimal number (K).

2) Indirect specification

Circuit program



List program

```
0000 LD    X000
0001 OUT   T10
     (SP)  D10
0004 LDI   X001
0005 OUT   T11
     (SP)  R15
0008 OUT   C10
     (SP)  D20
```

The set value of a timer or counter can be set by a data register (D) or extension data register (R). At this time, the current value of the data register (D) or extension register (R) is regarded as the set value of the timer or counter.
It is necessary to write in advance the set value to a data register (D) or extension register (R) used for the set value by MOV instruction, DSW instruction or display unit before driving the timer or counter.

3) Setting range of timers and counters
The table below shows the set value range of timers and counters, the actual timer constants and the number of program steps (including the set value) for OUT instruction.

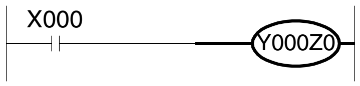
Timer/counter	Setting range (Value of K or current value of D or R)	Actual set value	Number of steps
1 ms timer	1 to 32767	0.001 to 32.767 sec	3
10 ms timer	1 to 32767	0.01 to 327.67 sec	3
100 ms timer		0.1 to 3276.7 sec	
16-bit counter	1 to 32767	Same as left	3
32-bit counter	−2,147,483,648 to +2,147,483,647	Same as left	5

1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

3. Indexing

Devices used in OUT instruction can be indexed with index registers (V and Z).
(State relays (S), special auxiliary relays (M), 32-bit counters (C), and "D□.b" cannot be indexed.)

Circuit program



List program

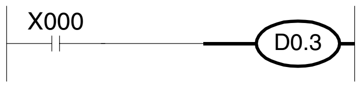
```
0000 LD  X000
0001 OUT Y000Z0
```

The index registers V0 to V7 and Z0 to Z7 are available for indexing.
When a used device is an input (X) or output (Y), the value of an index register (V or Z) is converted into an octal number, and then added.
Example: When the value of Z0 is "20", Y024 turns ON or OFF.

4. Bit specification of data register (D)

A bit in data register (D) can be specified as a device used in OUT instruction.

Circuit program



List program

```
0000 LD  X000
0001 OUT D0.3
```

When specifying a bit in data register, input "." after a data register (D) number, and then input a bit number (0 to F) consecutively.
Only 16-bit data registers are available.
Specify a bit number as "0, 1, 2, ... 9, A, B, ... F" from the least significant bit.
Example: In the example shown on the left, the bit 3 of D0 turns ON or OFF when X000 turns ON or OFF.

Caution

When a special internal relay (M), timer or counter is used, program steps increase as described in "Setting range of timers and counters" on the previous page.

Errors

- When an I/O number used in OUT instruction does not exist due to indexing, M8316 (Non-existing I/O specification error) turns ON.
- When the device number of a device (M, T or C) other than I/O used in OUT instruction does not exist due to indexing, an operation error (error code: 6706) occurs.

7.3 AND, ANI

Outline

AND and ANI instructions connect one contact in series.
The number of contacts connected in series is not limited, so AND and ANI instructions can be used consecutively as many times as necessary.
Output to another coil by way of a contact after OUT instruction is called cascade output.
Such cascade output can be repeated as many times as necessary as far as the order is correct.

1. Instruction format

AND

AND

Basic Instruction	Mnemonic	Operation Condition
	AND	Continuous Operation
	–	Pulse (Single) Operation

ANI

AND Inverse

Basic Instruction	Mnemonic	Operation Condition
	ANI	Continuous Operation
	–	Pulse (Single) Operation

→ For the number of instruction steps, refer to Section 7.15.

2. Applicable devices

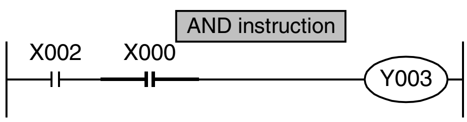
Instruction	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
AND	✓	✓	▲1	✓	▲1	▲2	▲2												✓					
ANI	✓	✓	▲1	✓	▲1	▲2	▲2												✓					

▲1: Special auxiliary relays (M) and 32-bit counters (C) cannot be indexed with index registers (V and Z).
▲2: State relays (S) and "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

1. AND instruction (serial connection of NO (normally open) contacts)

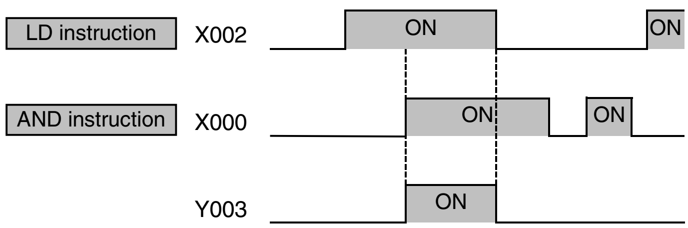
Circuit program



List program

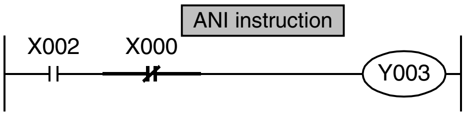
```
0000  LD   X002
0001  AND  X000   ← Contact connected in series
0002  OUT  Y003
```

Timing chart



2. ANI instruction (serial connection of NC (normally closed) contacts)

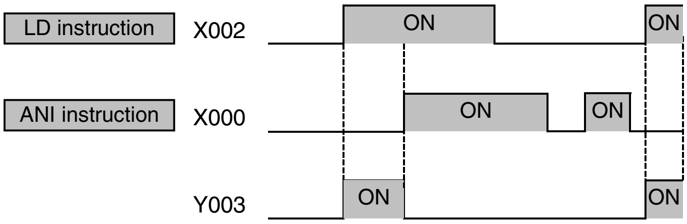
Circuit program



List program

```
0000 LD  X002
0001 ANI X000 ←Contact connected in series
0002 OUT Y003
```

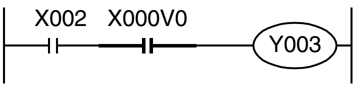
Timing chart



3. Indexing

Devices used in AND and ANI instruction can be indexed with index registers (V and Z).
(State relays (S), special auxiliary relays (M), 32-bit counters (C), and "D□.b" cannot be indexed.)

Circuit program



List program

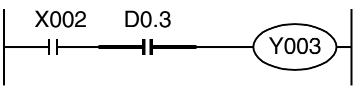
```
0000 LD  X002
0001 AND X000V0
0002 OUT Y003
```

The index registers V0 to V7 and Z0 to Z7 are available for indexing.
When a used device is an input (X) or output (Y), the value of an index register (V, Z) is converted into an octal number, and then added.
Example: When the value of V0 is "10", AND contact is set to ON or OFF by X012.

4. Bit specification of data register (D)

A bit in data register (D) can be specified as a device used in AND and ANI instructions.

Circuit program



List program

```
0000 LD  X002
0001 AND D0.3
0002 OUT Y003
```

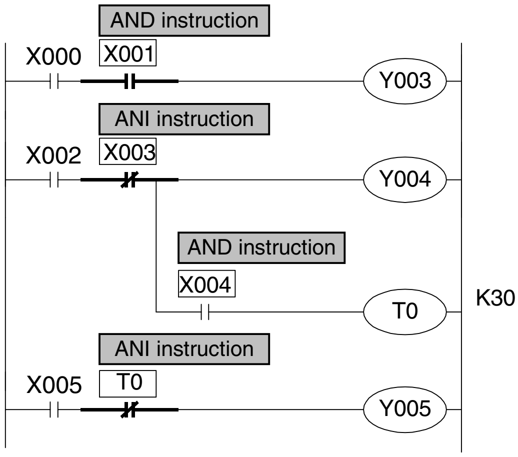
When specifying a bit in data register, input "." after a data register (D) number, and then input a bit number (0 to F) consecutively.
Only 16-bit data registers are available.
Specify a bit number as "0, 1, 2, ... 9, A, B, ... F" from the least significant bit.
Example: In the example shown on the left, AND contact turns ON (becomes conductive) when the bit 3 of D0 turns ON.

Errors

- When an I/O number used in AND or ANI instruction does not exist due to indexing, M8316 (Non-existing I/O specification error) turns ON.
- When the device number of a device (M, T or C) other than I/O used in AND or ANI instruction does not exist due to indexing, an operation error (error code: 6706) occurs.

Program examples

Circuit program



List program

```
0000  LD    X000
0001  AND   X001  <---- Contact connected in series
0002  OUT   Y003
0003  LD    X002
0004  ANI   X003  <---- Contact connected in series
0005  OUT   Y004
0006  AND   X004  <---- Contact connected in series
0007  OUT   T0
      (SP)  K30
0010  LD    X005  <---- Contact connected in series
0011  ANI   T0
0012  OUT   Y005
```

7.4 OR, ORI

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

OR and ORI instructions are used to connect one contact in parallel.
If two or more contacts are connected in series, use ORB instruction described later to connect such a serial circuit block to another circuit in parallel.
A step containing OR or ORI instruction is connected in parallel to a preceding step containing LD or LDI instruction. There is no limitation in the number of times of parallel connection.

1. Instruction format

OR
OR

Basic Instruction	Mnemonic	Operation Condition
	OR	Continuous Operation
	–	Pulse (Single) Operation

ORI
OR Inverse

Basic Instruction	Mnemonic	Operation Condition
	ORI	Continuous Operation
	–	Pulse (Single) Operation

→ **For the number of instruction steps, refer to Section 7.15.**

2. Applicable devices

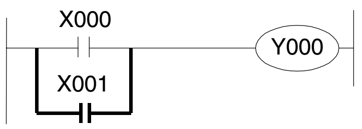
Instruction	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
OR	✓	✓	▲1	✓	▲1	▲2	▲2												✓					
ORI	✓	✓	▲1	✓	▲1	▲2	▲2												✓					

▲1: Special auxiliary relays (M) and 32-bit counters (C) cannot be indexed with index registers (V and Z).
▲2: State relays (S) and "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

1. OR instruction (parallel connection of NO (normally open) contacts)

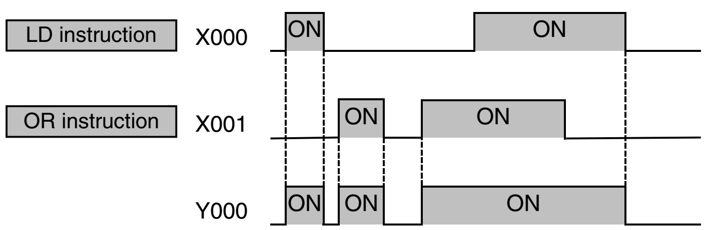
Circuit program



List program

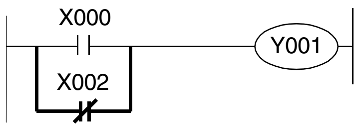
```
0000 LD   X000
0001 OR   X001
0002 OUT  Y000
```

Timing chart



2. ORI instruction (parallel connection of NC (normally closed) contacts)

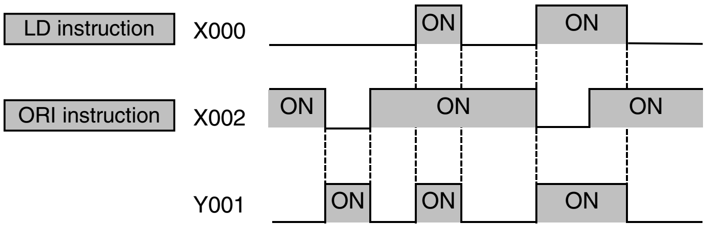
Circuit program



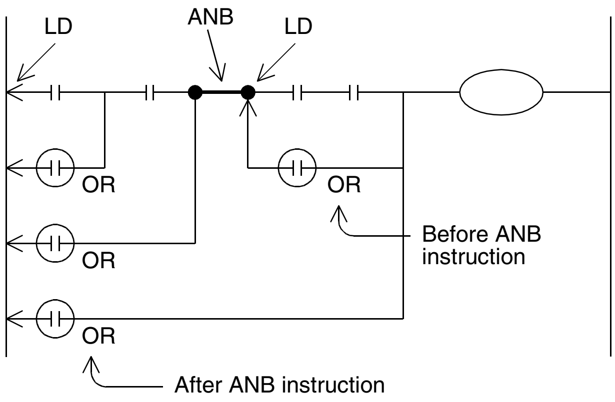
List program

```
0000 LD   X000
0001 ORI  X002
0002 OUT  Y001
```

Timing chart



3. Relationship with ANB instruction

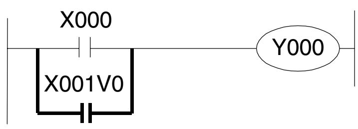


The parallel connection by OR or ORI instruction is connected to the preceding LD or LDI instruction in principle. After ANB instruction, however, the parallel connection by OR or ORI instruction is connected to the second preceding LD or LDI instruction.

4. Indexing

Devices used in OR and ORI instruction can be indexed with index registers (V and Z).
(State relays (S), special auxiliary relays (M), 32-bit counters, and "D□.b" cannot be indexed.)

Circuit program



List program

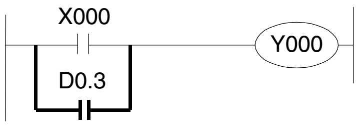
```
0000 LD   X000
0001 OR   X001V0
0002 OUT  Y000
```

The index registers V0 to V7 and Z0 to Z7 are available for indexing.
When a used device is an input (X) or output (Y), the value of an index register (V or Z) is converted into an octal number, and then added.
Example:When the value of V0 is "10", OR contact is set to ON (becomes conductive) or OFF (becomes non-conductive) by X013.

5. Bit specification of data register (D)

A bit in data register (D) can be specified as a device used in OR and ORI instructions.

Circuit program



List program

```
0000 LD   X000
0001 OR   D0.3
0002 OUT  Y000
```

When specifying a bit in data register, input "." after a data register (D) number, and then input a bit number (0 to F) consecutively.
Only 16-bit data registers are available.
Specify a bit number as "0, 1, 2, ... 9, A, B, ... F" from the least significant bit.
Example: In the example shown on the left, OR contact is set to ON (becomes conductive) or OFF (becomes non-conductive) by the bit 3 of D0.

Errors

- When an I/O number used in OR or ORI instruction does not exist due to indexing, M8316 (Non-existing I/O specification error) turns ON.
- When the device number of a device (M, T or C) other than I/O used in OR or ORI instruction does not exist due to indexing, an operation error (error code: 6706) occurs.

7.5 LDP, LDF, ANDP, ANDF, ORP, ORF

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

LDP, ANDP, and ORP instructions for contacts detect the rising edge, and become active during one operation cycle only at the rising edge of a specified bit device (that is, when the bit device turns ON from OFF).
Contact instructions LDF, ANDF and ORF detect the falling edge, and become active during one operation cycle only at the falling edge of a specified bit device (that is, when the bit device turns OFF from ON).

1. Instruction format

LDP — Load Pulse

Basic Instruction	Mnemonic	Operation Condition
	–	Continuous Operation
	LDP	Pulse (Single) Operation

LDF — Load Falling Pulse

Basic Instruction	Mnemonic	Operation Condition
	–	Continuous Operation
	LDF	Pulse (Single) Operation

ANDP — AND Pulse

Basic Instruction	Mnemonic	Operation Condition
	–	Continuous Operation
	ANDP	Pulse (Single) Operation

ANDF — AND Falling Pulse

Basic Instruction	Mnemonic	Operation Condition
	–	Continuous Operation
	ANDF	Pulse (Single) Operation

ORP — OR Pulse

Basic Instruction	Mnemonic	Operation Condition
	–	Continuous Operation
	ORP	Pulse (Single) Operation

ORF — OR Falling Pulse

Basic Instruction	Mnemonic	Operation Condition
	–	Continuous Operation
	ORF	Pulse (Single) Operation

→ For the number of instruction steps, refer to Section 7.15.

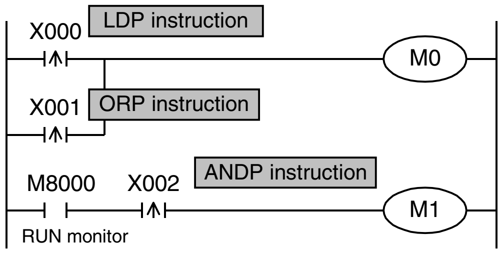
2. Applicable devices

Instruction	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
LDP	✓	✓	✓	✓	✓	✓	✓																	
LDF	✓	✓	✓	✓	✓	✓	✓																	
ANDP	✓	✓	✓	✓	✓	✓	✓																	
ANDF	✓	✓	✓	✓	✓	✓	✓																	
ORP	✓	✓	✓	✓	✓	✓	✓																	
ORF	✓	✓	✓	✓	✓	✓	✓																	

Explanation of function and operation

1. LDP, ANDP, and ORP instructions (initial logical operation of rising edge pulse, serial connection of rising edge pulse, and parallel connection of rising edge pulse)

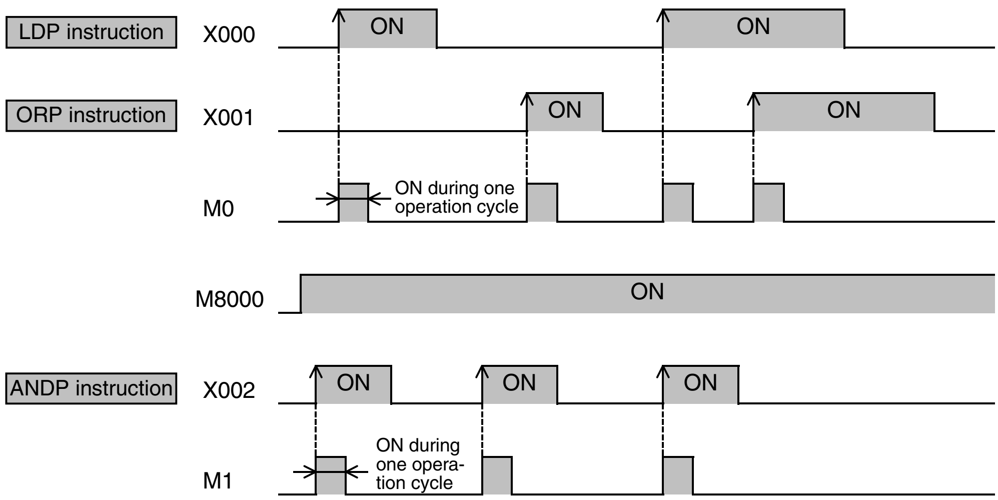
Circuit program



List program

```
0000 LDP  X000
0002 ORP  X001
0004 OUT  M0
0005 LD   M8000
0006 ANDP X002
0008 OUT  M1
```

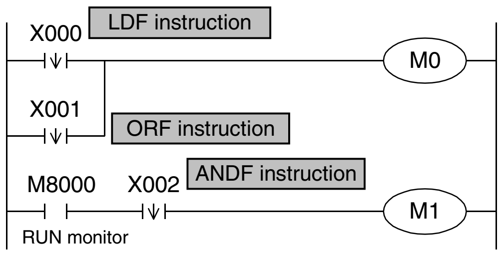
Timing chart



In the example shown above, M0 or M1 is ON during only one operation cycle when X000 to X002 turn ON from OFF.

2. LDF, ANDF, and ORF instructions (initial logical operation of falling/trailing edge pulse, serial connection of falling/trailing edge pulse, and parallel connection of falling/trailing edge pulse)

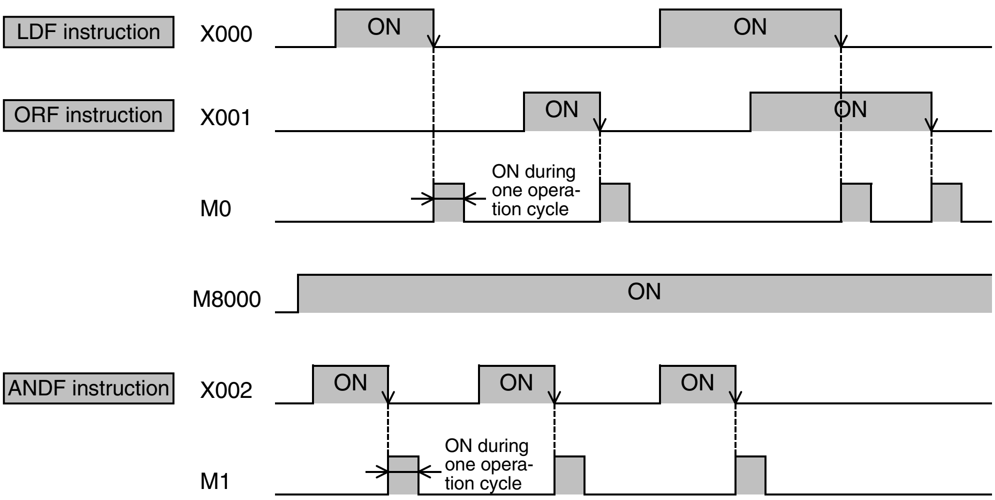
Circuit program



List program

```
0000 LDF  X000
0002 ORF  X001
0004 OUT  M0
0005 LD   M8000
0006 ANDF X002
0008 OUT  M1
```

Timing chart

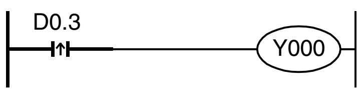


In the example shown above, M0 or M1 is ON during only one operation cycle when X000 to X002 turn OFF from ON.

3. Bit specification of a data register (D)

A bit in data register (D) can be specified as a device used in LDP, LDF, ANDP, ANDF, ORP and ORF instructions.

Circuit program



List program

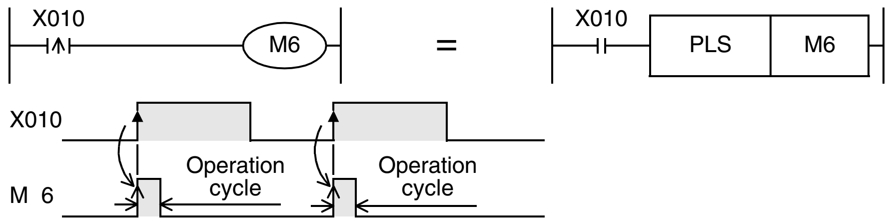
```
0000 LDP D0.3
0001 OUT Y000
```

When specifying a bit in data register, input "." after a data register (D) number, and then input a bit number (0 to F) consecutively.
Only 16-bit data registers are available.
Specify a bit number as "0, 1, 2, ... 9, A, B, ... F" from the least significant bit.
Example: In the example shown on the left, LDP contact turns ON (becomes conductive) or OFF (becomes non-conductive) when the bit 3 of D0 turns ON or OFF.

4. Output drive side

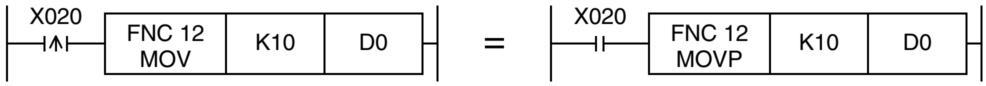
The following two circuits offer a same operation:

<OUT instruction> <Pulse instruction>



In each circuit, M6 is ON during only one operation cycle when X010 turns ON from OFF.

<Rising edge detection> <Pulse instruction (applied instruction)>

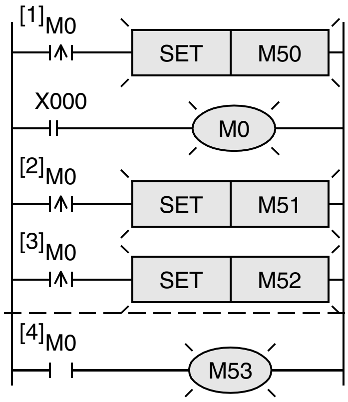


In each circuit, MOV instruction is executed only once when X020 turns ON from OFF.

5. Differences in the operation caused by auxiliary relay (M) numbers

When an auxiliary relay (M) is specified as a device in LDP, LDF, ANDP, ANDF, ORP and ORF instructions, the operation varies depending on the device number range as shown in the figure below.

<M0 to M2799, M3072 to M7679>

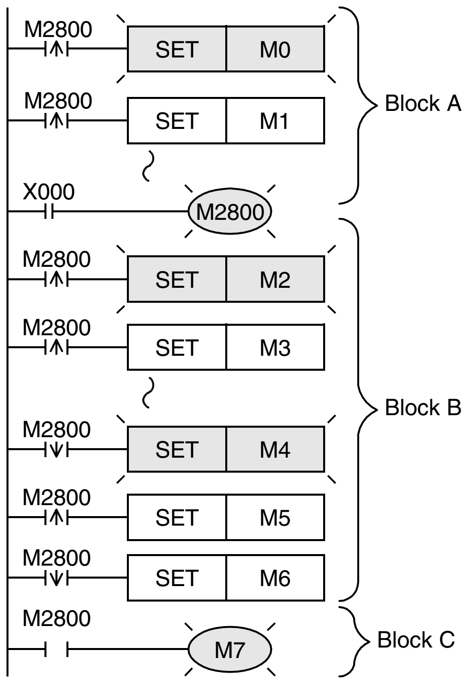


After M0 is driven by X000, all contacts [1] to [4] corresponding to M0 are activated.

•The contacts [1] to [3] detect the rising edge of M0.

•Because of LD instruction, the contact [4] is conductive while M0 is ON.

<M2800 to M3071>



From M2800 driven by X000, the program is divided into the upper block (block A) and the lower block (block B). In each of the blocks A and B, only the first contact which detects the rising or falling edge is activated.
Because of LD instruction, the contact in the block C is conductive while M2800 is ON.
By utilizing these characteristics, "transition of state by same signal" in a step ladder circuit can be efficiently programmed.

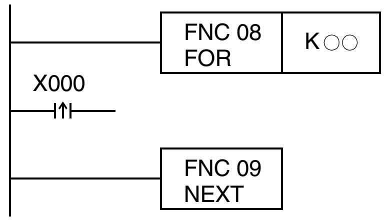
Cautions

1. Cautions when LDP, LDF, ANDP, ANDF, ORP, or ORF instruction programmed in a same step is executed two or more times within one operation cycle

When LDP, LDF, ANDP, ANDF, ORP or ORF instruction programmed in a same step is executed two or more times within one operation cycle, the following operation results:

Programs executed two or more times

- Program between FOR and NEXT instructions
- Program which executes a same subroutine program from two or more CALL instructions during one operation cycle
- Program which jumps to a label (P) in a smaller step number by CJ instruction



Operation

1) When a device turns ON from OFF
1st time: LDP, ANDP or ORP instruction turns ON.
2nd time and later: When the device status is same as the time when the instruction was executed last, the instruction turns OFF.
2) When a device turns OFF from ON
1st time: LDF, ANDF or ORF instruction turns ON.
2nd time and later: When the device status is same as the time when the instruction was executed last, the instruction turns OFF.

2. Cautions on write during RUN

1) Instructions for falling edge pulse
When write during RUN is completed for a circuit including an instruction for falling edge pulse (LDF, ANDF, or ORF instruction), the instruction for falling edge pulse is not executed without regard to the ON/OFF status of the target device of the instruction for falling edge pulse.
When write during RUN is completed for a circuit including an instruction for falling edge pulse (PLF instruction), the instruction for falling edge pulse is not executed without regard to the ON/OFF status of the operation condition device.
It is necessary to set to ON the target device or operation condition device once and then set it to OFF for executing the instruction for falling edge pulse.
2) Instructions for rising edge pulse
When write during RUN is completed for a circuit including an instruction for rising edge pulse, the instruction for rising edge pulse is executed if a target device of the instruction for rising edge pulse or the operation condition device is ON.
Target instructions for rising edge pulse: LDP, ANDP, ORP, and pulse operation type applied instructions (such as MOVP)

Contact ON/OFF status (while write during RUN is executed)	Instruction for rising edge pulse	Instruction for falling edge pulse
OFF	Not executed	Not executed
ON	Executed*1	Not executed

*1. PLS instruction is not executed.

7.6 ORB

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

A circuit in which two or more contacts are connected in series is called serial circuit block.

1. Instruction format

ORB
OR Block

Basic Instruction	Mnemonic	Operation Condition
1 step	ORB	Continuous Operation
	–	Pulse (Single) Operation

2. Applicable devices

Instruction	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
ORB	There are no applicable devices.																							

Explanation of function and operation

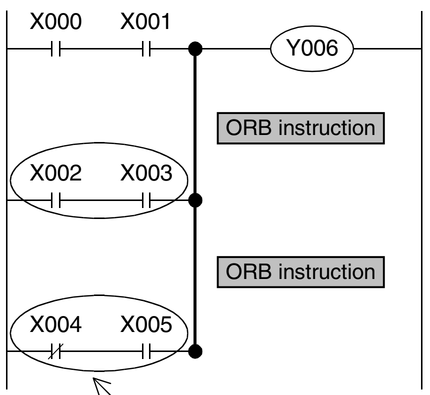
1. ORB instruction (parallel connection of multiple contact circuits)

When connecting serial circuit blocks in parallel, use LD or LDI instruction at the start of branch, and use ORB instruction at the end of branch.

ORB instruction is an independent instruction not associated with any device number in the same way as ANB instruction described later.

When there are many parallel circuits, ORB instruction can be used for each circuit block to connect them.

Circuit program



List program

Appropriate program				Inappropriate program			
0000	LD	X000		0000	LD	X000	
0001	AND	X001		0001	AND	X001	
0002	LD	X002		0002	LD	X002	
0003	AND	X003		0003	AND	X003	
0004	**ORB**		←	0004	LDI	X004	
0005	LDI	X004		0005	AND	X005	
0006	AND	X005		0006	**ORB**		←
0007	**ORB**		←	0007	**ORB**		←
0008	OUT	Y006		0008	OUT	Y006	

Caution

There is no limitation in the number of parallel circuits which can be connected by ORB instructions (in the case of appropriate program shown above).

Though ORB instructions can be used at one time, note that the repeated use of LD or LDI instruction is limited to 8 or less (in the case of inappropriate program shown above).

7.7 ANB

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Use ANB instruction to connect a branch circuit (parallel circuit block) to the preceding circuit in series.
Use LD or LDI instruction at the start of branch. After completing a parallel circuit block, connect the parallel circuit block to the preceding circuit in series by ANB instruction.
When there are many parallel circuits, ANB instruction can be used in each circuit block to connect them.

1. Instruction format

ANB

AND Block

Basic Instruction	Mnemonic	Operation Condition
1 step	ANB	Continuous Operation
	–	Pulse (Single) Operation

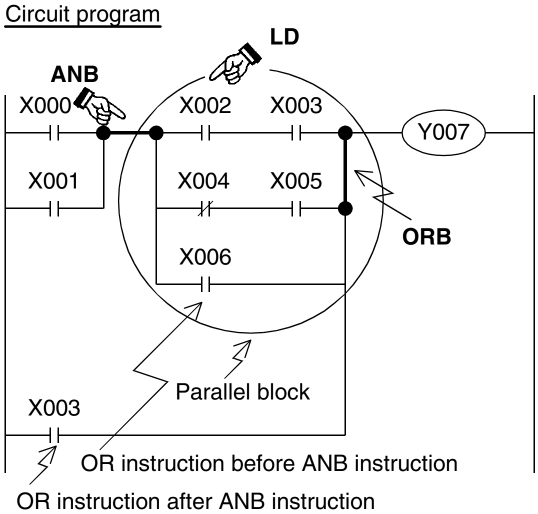
2. Applicable devices

Instruction	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
ANB	There are no applicable devices.																							

Explanation of function and operation

1. ANB instruction (serial connection of multiple parallel circuits)

Circuit program



List program

0000	LD	X000	
0001	OR	X001	
0002	LD	X002	← Branch start point
0003	AND	X003	
0004	LDI	X004	←
0005	AND	X005	
0006	ORB		← Parallel block is completed.
0007	OR	X006	←
0008	**ANB**		← It is connected to the preceding circuit in series.
0009	OR	X003	
0010	OUT	Y007	

Caution

There is no limitation in the number of ANB instruction.
Though ANB instructions can be used at one time, note that the repeated use of LD or LDI instruction is limited to 8 or less in the same way as ORB instruction.

7.8 MPS, MRD, MPP

Outline

FX3U and FX3UC PLCs have 11 memories called "Stack" which store the intermediate result (ON or OFF) of operations.

1. Instruction format

MPS — Memory Point Store

Basic Instruction	Mnemonic	Operation Condition
1 step	MPS	Continuous Operation
	–	Pulse (Single) Operation

MRD — Memory Read

Basic Instruction	Mnemonic	Operation Condition
1 step	MRD	Continuous Operation
	–	Pulse (Single) Operation

MPP — Memory POP

Basic Instruction	Mnemonic	Operation Condition
1 step	MPP	Continuous Operation
	–	Pulse (Single) Operation

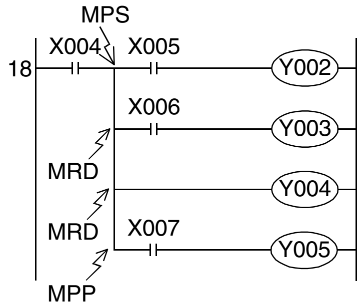
2. Applicable devices

Instruction	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
MPS	There are no applicable devices.																							
MRD	There are no applicable devices.																							
MPP	There are no applicable devices.																							

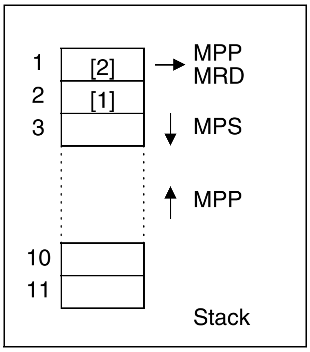
Explanation of function and operation

These instructions are convenient in programming branched multi-output circuits.

1. **MPS, MRD, and MPP instructions (stores the current result of the internal PLC operations, reads the current result of the internal PLC operations, and pops (recalls and removes) the currently stored result)**




```
0018  LD    X004
0019  MPS
0020  AND   X005
0021  OUT   Y002
0022  MRD
0023  AND   X006
0024  OUT   Y003
0025  MRD
0026  OUT   Y004
0027  MPP
0028  AND   X007
0029  OUT   Y005
0030  END
```



- Use MPS instruction to store the intermediate result of operation, and then drive the output Y002.
- Use MRD instruction to read the stored data, and then drive the output Y003.
  MRD instruction can be programmed as many times as necessary.
- In the final output circuit, use MPP instruction instead of MRD instruction.
  MPP instruction reads the stored data described above, and then resets it.

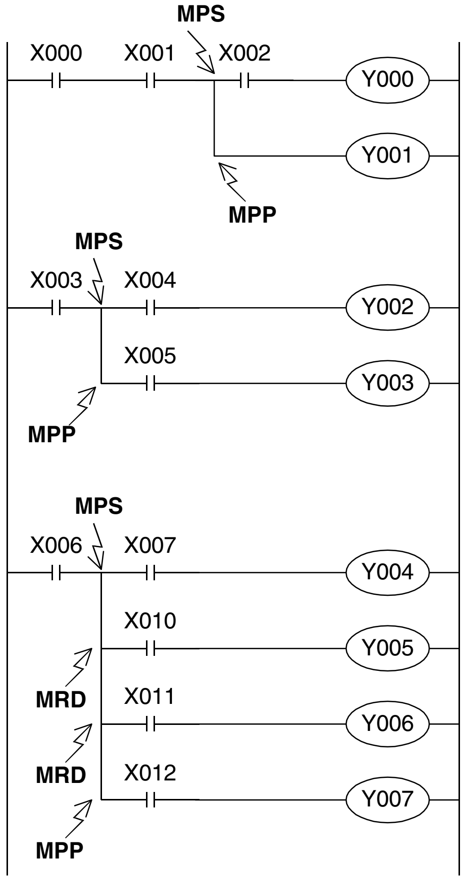
Error

MPS instruction can be used two or more times.
However, the difference between number of MPS instructions and the number of MPP instructions should be 11 or less, and should be 0 at the end.

Program examples

1) Program example 1: One stack
   Only one stack is used in this example.

Circuit program

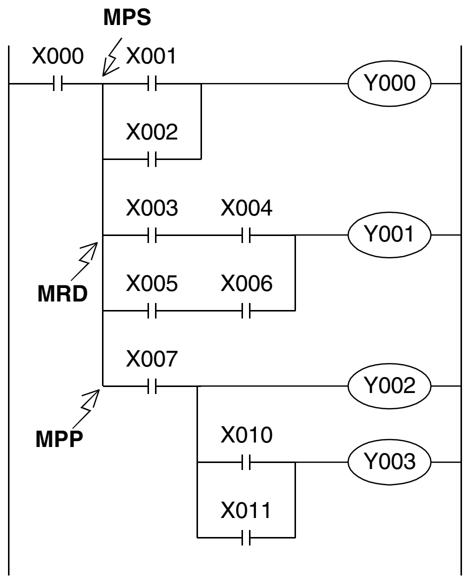


List program

```
0000  LD    X000
0001  AND   X001
0002  MPS
0003  AND   X002
0004  OUT   Y000
0005  MPP
0006  OUT   Y001
0007  LD    X003
0008  MPS
0009  AND   X004
0010  OUT   Y002
0011  MPP
0012  AND   X005
0013  OUT   Y003
0014  LD    X006
0015  MPS
0016  AND   X007
0017  OUT   Y004
0018  MRD
0019  AND   X010
0020  OUT   Y005
0021  MRD
0022  AND   X011
0023  OUT   Y006
0024  MPP
0025  AND   X012
0026  OUT   Y007
```

2) Program example 2: One stack with ANB and ORB instructions

Circuit program

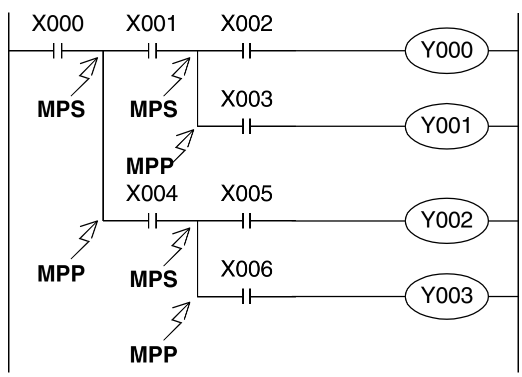


List program

```
0000 LD   X000
0001 MPS
0002 LD   X001
0003 OR   X002
0004 ANB
0005 OUT  Y000
0006 MRD
0007 LD   X003
0008 AND  X004
0009 LD   X005
0010 AND  X006
0011 ORB
0012 ANB
0013 OUT  Y001
0014 MPP
0015 AND  X007
0016 OUT  Y002
0017 LD   X010
0018 OR   X011
0019 ANB
0020 OUT  Y003
```

3) Program example 3: Two stacks

Circuit program

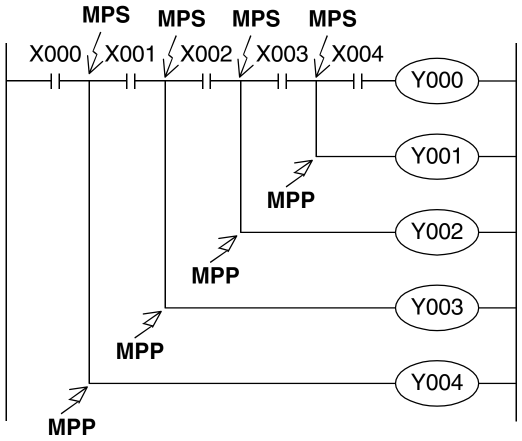


List program

```
0000 LD   X000
0001 MPS
0002 AND  X001
0003 MPS
0004 AND  X002
0005 OUT  Y000
0006 MPP
0007 AND  X003
0008 OUT  Y001
0009 MPP
0010 AND  X004
0011 MPS
0012 AND  X005
0013 OUT  Y002
0014 MPP
0015 AND  X006
0016 OUT  Y003
```

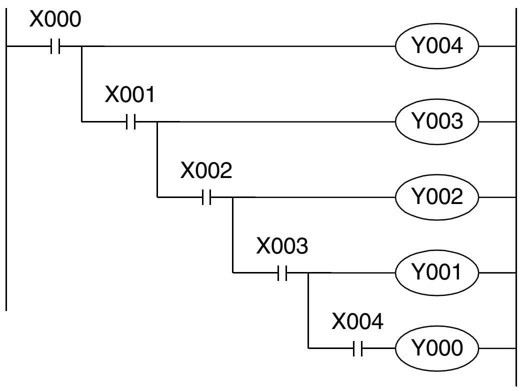
4) Program example 4: Four stacks

Circuit program



List program

0000	LD	X000	0009	OUT	Y000
0001	**MPS**		0010	**MPP**	
0002	AND	X001	0011	OUT	Y001
0003	**MPS**		0012	**MPP**	
0004	AND	X002	0013	OUT	Y002
0005	**MPS**		0014	**MPP**	
0006	AND	X003	0015	OUT	Y003
0007	**MPS**		0016	**MPP**	
0008	AND	X004	0017	OUT	Y004



0000	LD	X000
0001	OUT	Y004
0002	AND	X001
0003	OUT	Y003
0004	AND	X002
0005	OUT	Y002
0006	AND	X003
0007	OUT	Y001
0008	AND	X004
0009	OUT	Y000

In programming a circuit on the upper side, it is necessary to MPS instruction three times.
By changing the circuit on the upper side into the circuit on the lower side, the same contents can be programmed easily without MPS instruction.

7.9 MC, MCR

Outline

When MC instruction is executed, the bus line (LD or LDI point) is moved to a position after MC contact.
The bus line can be returned to the original position by MCR instruction.
By changing a device (Y or M) number, MC instruction can be used as many times as necessary.
If a same device number is used twice, however, it results in the double coil operation in the same way as OUT instruction.

1. Instruction format

MC	Basic Instruction	Mnemonic	Operation Condition
Master Control		MC	Continuous Operation
		–	Pulse (Single) Operation

MCR	Basic Instruction	Mnemonic	Operation Condition
Master Control Reset	2 steps	MCR	Continuous Operation
		–	Pulse (Single) Operation

→ **For the number of steps of MC instruction, refer to Section 7.15.**

2. Applicable devices

Instruction	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
MC		✓	▲																					
MCR	There are no applicable devices.																							

▲: Except special auxiliary relays (M)

Explanation of function and operation

1. MC and MCR instructions (denotes the start of a master control block and denotes the end of a master control block)

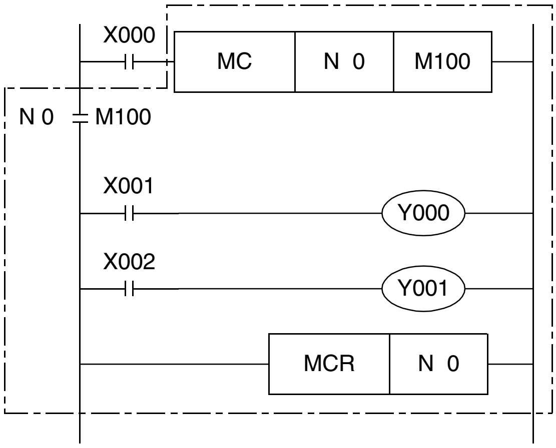
When MC instruction is executed, the bus line is moved to a position after MC contact.
Drive instructions connected to the bus line after MC contact execute each operation only when MC instruction is executed, and do not execute the operation when MC instruction is not executed.
In the program example below, the instructions from MC to MCR are executed as they are while the input X000 is ON.
However, while the input X000 is OFF, each driven device offers the following operation:

Timers (except retentive type timers) and devices driven by OUT instruction: Turn OFF.

Retentive type timers, counters and devices driven by SET/RST instruction: Hold the current status.

The expressions of circuit programs used to explain operations are circuits (for reading or monitoring) of GX Developer.

Circuit program



List program

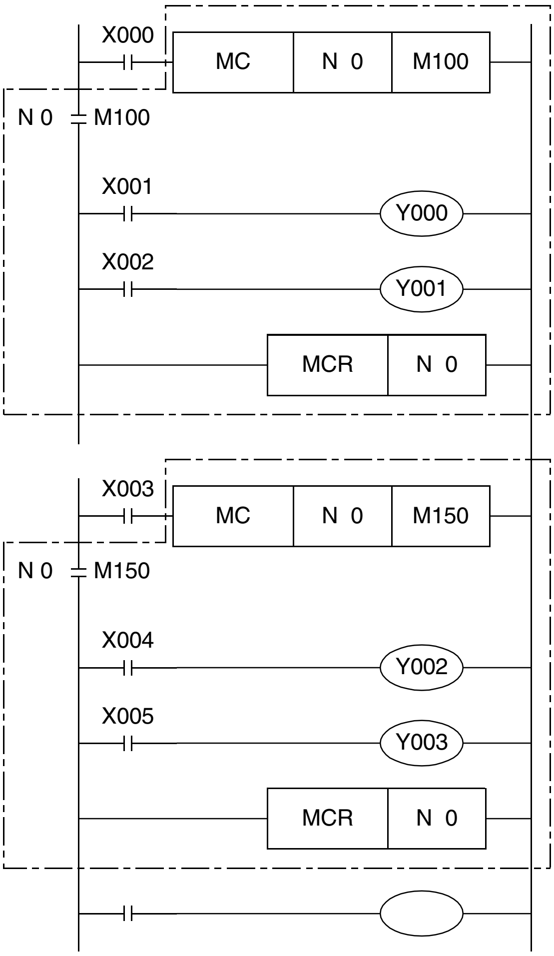
0000	LD	X000	
0001	**MC**	N 0	Three-step instruction
	SP	M100	
0004	LD	X001	
0005	OUT	Y000	
0006	LD	X002	
0007	OUT	Y001	
0008	**MCR**	N 0	← Two-step instruction

← Write MCR N0 instruction.

Program examples

1) When the nesting structure is not adopted

Circuit program



List program

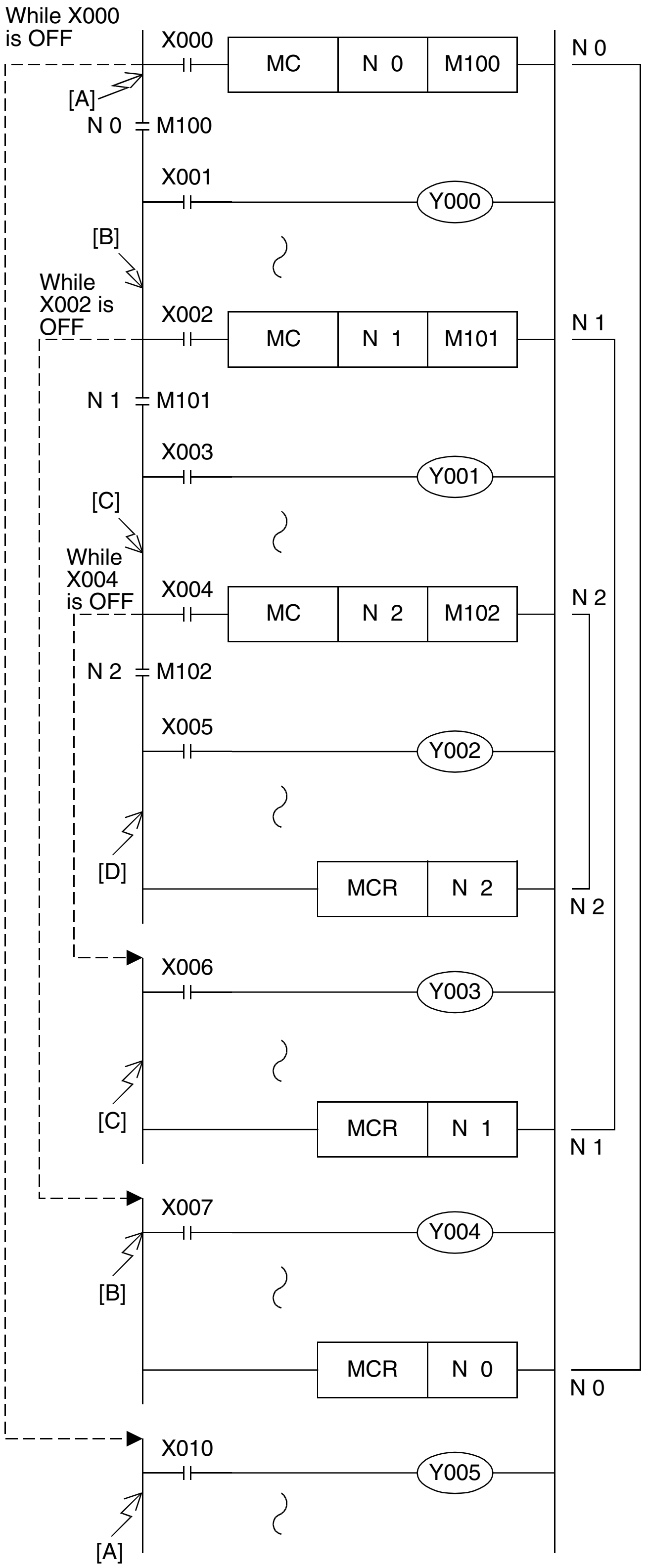
0000	LD	X000	
0001	**MC**	N 0	Three-step instruction
	SP	M100	
0004	LD	X001	
0005	OUT	Y000	
0006	LD	X002	
0007	OUT	Y001	
0008	**MCR**	N 0	← Two-step instruction

← Return to the bus line ("N0" shows the nest level.)

← When not adopting the nesting structure, use "N0" again. There is no limitation in the number of "N0".
Only in the nesting structure, increase the nest level "N" in the way "N0 → N1 ... N6 → N7" as shown in the example 2 on the next page.

2) When the nesting structure is adopted
When using MC instructions inside MC instruction, increase the nest level "N" in turn in the way "N0 → N1 → N2 → N3 → N4 → N5 → N6 → N7".
For returning from the nesting structure, reset the nest levels from the highest one in turn using MCR instruction in the way "N7 → N6 → N5 → N4 → N3 → N2 → N1 → N0".
For example, if "MCR N5" is programmed without programming "MCR N6" and "MCR N7", the nest level is returned to 5 at one time.
Available nest levels are from N0 to N7 (eight layers).

Circuit program



Level N0
The bus line B is active while X000 is ON.

Level N1
The bus line C is active while both X000 and X002 are ON.

Level N2
The bus line D is active while all of X000, X002 and X004 are ON.

Level N1
The bus line returns to the status of the bus line C by "MCR N2".

Level N0
The bus line returns to the status of the bus line B by "MCR N1".

Initial status
The bus line returns to the initial status of the bus line A by "MCR N0". Accordingly, Y005 turns ON or OFF by turning ON or OFF of X010 without regard to X000, X002 and X004.

7.10 INV

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

1. Instruction Format

INV instruction inverts the operation result up to just before INV instruction, and does not require device number specification.

INV

Inverse

Basic Instruction	Mnemonic	Operation Condition
1 step	INV	Continuous Operation
	–	Pulse (Single) Operation

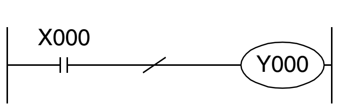
2. Applicable devices

Instruction	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
INV	There are no applicable devices.																							

Explanation of function and operation

1. INV instruction (inverts the result of operations)

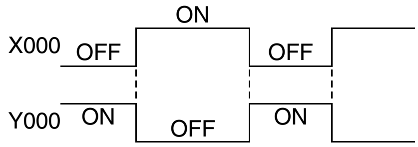
Circuit program



List program

```
0  LD   X000
1  INV
2  OUT  Y000
```

Timing chart



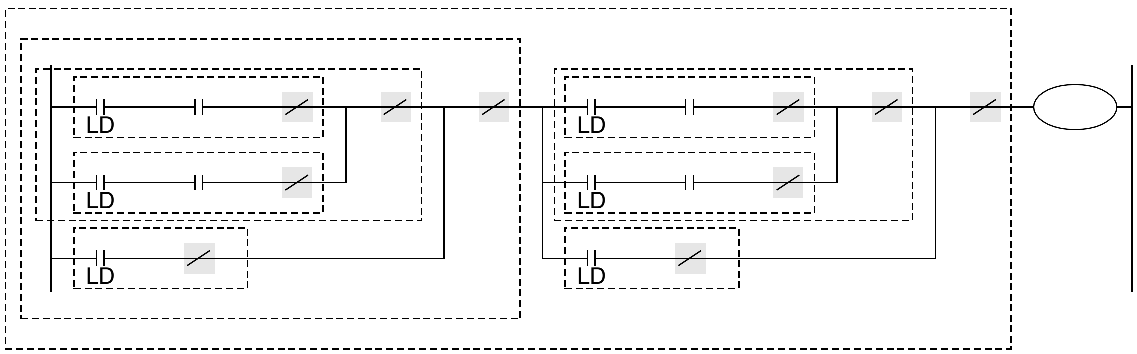
Operation result until just before INV instruction		Operation result after INV instruction is executed
OFF	→	ON
ON	→	OFF

Inverted

In the figure above, Y000 turns ON when X000 is OFF, and Y000 turns OFF when X000 is ON.
INV instruction can be used in a same position as serial contact instructions (AND, ANI, ANDP and ANDF).
Different from LD, LDI, LDP and LDF instructions shown in the list, INV instruction cannot execute connection to bus lines. Different from OR, ORI, ORP and ORF instructions, INV instruction cannot be used independently in parallel to a contact instruction.

2. Operation range of INV instruction

When INV instruction is used in a complicated circuit containing ORB and ANB instructions, the operation range of INV instruction is as shown in the figure below:



INV instruction inverts the operation result after LD, LDI, LDP or LDF instruction located before INV instruction.
Accordingly, if INV instructions are used inside ORB and ANB instructions, blocks after LD, LDI, LDP or LDF instruction seen from each INV instruction are regarded as the target of INV operation.

7.11 MEP, MEF

Outline

FX3UC Ver. 2.30 → FX3U Ver. 2.30 →

MEP and MEF commands are instructions that change the operation results to pulses so that device numbers do not have to be specified.

1) MEP
The operation results up to the MEP instruction become conductive when the driving contacts turn ON from OFF.
The use of MEP instructions simplifies the process of changing driving contacts to pulses when multiple contact points connect in a series.

2) MEF
The operation results up to the MEF instruction become conductive when the driving contacts turn OFF from ON.
The use of MEF instructions simplifies the process of changing driving contacts to pulses when multiple contact points connect in a series.

1. Instruction format

MEP	Basic Instruction	Mnemonic	Operation Condition
MEP	1 step	–	Continuous Operation
		MEP	Pulse (Single) Operation

MEF	Basic Instruction	Mnemonic	Operation Condition
MEF	1 step	–	Continuous Operation
		MEF	Pulse (Single) Operation

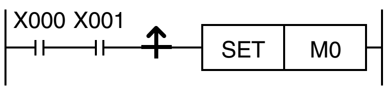
2. Applicable devices

Instruction	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
MEP	There are no applicable devices.																								
MEF	There are no applicable devices.																								

Explanation of function and operation

1. MEP instruction (ON during rising edge of driving contacts results)

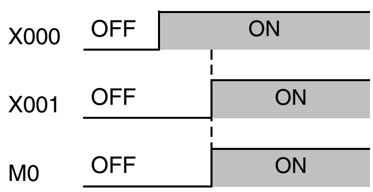
Circuit Program



List program

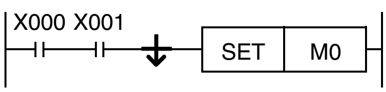
0	LD	X000
1	AND	X001
2	MEP	
3	SET	M0

Timing chart



2. MEF instruction (ON during falling edge of driving contacts results)

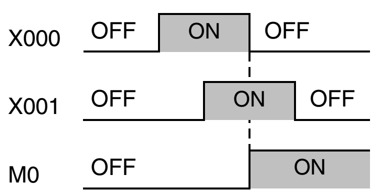
Circuit program



List program

0	LD	X000
1	AND	X001
2	MEF	
3	SET	M0

Timing chart



Caution

1. **MEP and MEF instructions may not operate normally if the indexed contact is modified and changed to pulses by sub-routine programs, the FOR and NEXT instructions, etc.**
2. **As the MEP and MEF instructions operate using the operation results immediately before them, use at the list program as the AND instruction.**
   The MEP and MEF instructions cannot be used at the list program as LD or OR.
3. **Caution on writing during RUN**
   1) Pulse command during rising edge of operation (MEP instruction) results
      After writing to the circuit with MEP instructions during RUN, the MEP instruction result turns ON (conductive) while the operation results up to the MEP instruction are ON.
   2) Pulse instruction during falling edge of operation (MEF command) results
      After writing to the circuit with MEF instructions during RUN , the MEF instruction result turns OFF (non-conductive), regardless of the operation results up to the MEF instruction. The operation results of MEF instruction turns ON (conductive) when the operation results up to the MEF instruction turn OFF.

Operation Results up to MEP/MEF Instruction (while writing is excuted during RUN)	MEP Instruction	MEF Instruction
OFF	OFF (non-conductive)	OFF (non-conductive)
ON	ON (conductive)	OFF (non-conductive)

Error

- There are no calculation errors in the MEP and MEF instructions.

7.12 PLS, PLF

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

When PLS instruction is executed, an applicable device is activated during only one operation cycle after a drive input turns ON.
When PLF instruction is executed, an applicable device is activated during only one operation cycle after a drive input turns OFF.
For example, when PLC mode is changed in the way "RUN → STOP → RUN" while a drive input remains ON, "PLS M0" operates, but "PLS M600 (backed up by the battery)" does not operate (when the PLC mode switches from STOP to RUN) because the status of M600 is latched even while the PLC is in the STOP mode.

1. Instruction format

PLS
Pulse

Basic Instruction	Mnemonic	Operation Condition
	–	Continuous Operation
	PLS	Pulse (Single) Operation

PLF
Pulse Falling

Basic Instruction	Mnemonic	Operation Condition
	–	Continuous Operation
	PLF	Pulse (Single) Operation

→ **For the number of instruction steps, refer to Section 7.15.**

2. Applicable devices

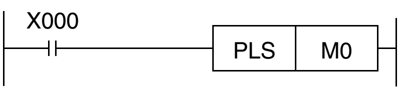
Instruction	Bit Devices: System User							Word Devices: Digit Specification				Word Devices: System User				Special Unit	Index			Others: Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
PLS		✓	▲																✓					
PLF		✓	▲																✓					

▲: Except special auxiliary relays (M)

Explanation of function and operation

1. PLS instruction (rising edge pulse)

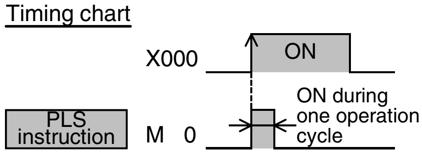
Circuit program



List program

```
0000  LD   X000
0001  PLS  M  0
```

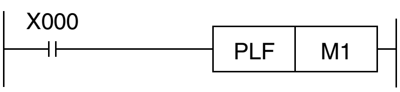
Timing chart



In the figure above, M0 is ON during only one operation cycle when X000 changes from OFF to ON.

2. PLF instruction (falling/trailing edge pulse)

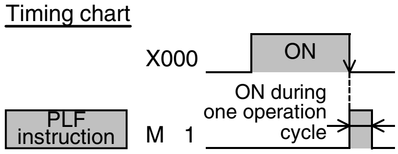
Circuit program



List program

```
0000  LD   X000
0001  PLF  M  1
```

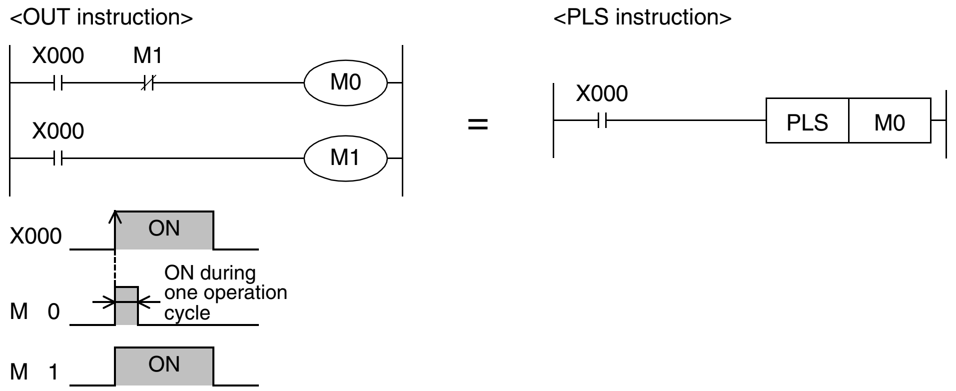
Timing chart



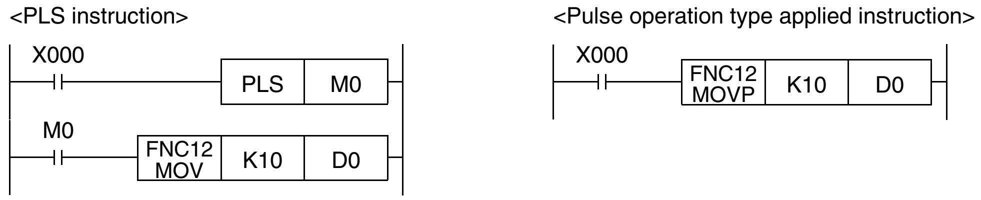
In the figure above, M1 is ON during only one operation cycle when X000 changes from ON to OFF.

3. Output drive side

The following two circuits cause a same operation.



In each case, M0 is ON during only one operation cycle when X000 changes from OFF to ON.



In each case, MOV instruction is executed only once when X000 changes from OFF to ON.

Caution

1. Cautions on write during RUN

1) Instructions for falling edge pulse
When write during RUN is completed for a circuit including an instruction for falling edge pulse (LDF, ANDF, or ORF instruction), the instruction for falling edge pulse is not executed without regard to the ON/OFF status of the target device of the instruction for falling edge pulse.
When write during RUN is completed for a circuit including an instruction for falling edge pulse (PLF instruction), the instruction for falling edge pulse is not executed without regard to the ON/OFF status of the operation condition device.
It is necessary to set to ON the target device or operation condition device once and then set it to OFF for executing the instruction for falling edge pulse.

2) Instructions for rising edge pulse
When write during RUN is completed for a circuit including an instruction for rising edge pulse, the instruction for rising edge pulse is executed if a target device of the instruction for rising edge pulse or the operation condition device is ON.
Target instructions for rising edge pulse: LDP, ANDP, ORP, and pulse operation type applied instructions (such as MOVP)

Contact ON/OFF status (while write during RUN is executed)	Instruction for rising edge pulse	Instruction for falling edge pulse
OFF	Not executed	Not executed
ON	Executed*1	Not executed

*1. PLS instruction is not executed.

1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

7.13 SET, RST

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

1) Setting a bit device (SET instruction (set bit device latch ON))
   When the command input turns ON, SET instruction sets to ON an output relay (Y), auxiliary relay (M), state relay (S) and bit specification of word device.
   Even if the command input turns OFF after that, the device which was set to ON by SET instruction remains ON.
2) Resetting a bit device (RST instruction (reset bit device OFF))
   RST instruction resets an output relay (Y), auxiliary relay (M), state relay (S), Timer (T), counter (C) or bit specification of a word device.
   Use the RST instruction to reset (set to OFF) a device which was set to ON by SET instruction.
3) Clearing the present value of a word device (RST instruction reset bit device OFF))
   RST instruction clears the current value of a timer (T), counter (C), data register (D), extension register (R) or index register (V)(Z).
   RST instruction can be used to clear to "0" the contents of a data register (D) or index register (V)(Z). (The same result can be obtained by MOV instruction which transfers the constant K0.)
   RST instruction can be used also to reset the current value and return the contact of retentive type timers T246 to T255.
   SET and RST instructions can be used for a same device as many times as necessary in an arbitrary order.

1. Instruction format

SET
SET

Basic Instruction	Mnemonic	Operation Condition
	SET	Continuous Operation
	–	Pulse (Single) Operation

RST
Reset

Basic Instruction	Mnemonic	Operation Condition
	RST	Continuous Operation
	–	Pulse (Single) Operation

→ **For the number of instruction steps, refer to Section 7.15.**

2. Applicable devices

Instruction	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
SET		✓	▲1			▲2	▲2												✓					
RST		✓	▲1	✓	▲1	▲2	▲2					▲2	▲2	▲2	▲2		▲2	▲2	✓					

▲1: Special auxiliary relays (M) and 32-bit counters (C) cannot be indexed with index registers (V and Z).
▲2: State relays (S) and "D□.b" cannot be indexed with index registers (V and Z).

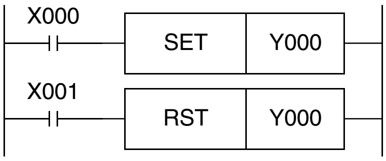
Explanation of function and operation

SET instruction drives the coil for an output relay (Y), auxiliary relay (M), state relay (S) and bit specification of data register (D).

1. When using a bit device

SET instructions located in parallel can be used consecutively as many times as necessary.
In the program example shown below, "RST Y000" after "SET Y000" corresponds to this usage.

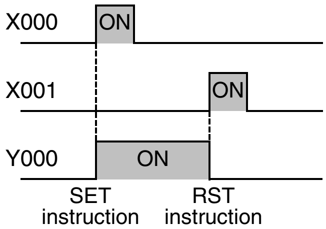
Circuit program



List program

0	LD	X000
1	**SET**	Y000
2	LD	X001
3	**RST**	Y000

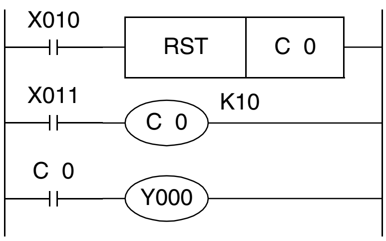
Timing chart



2. When using a word device (timer or counter)

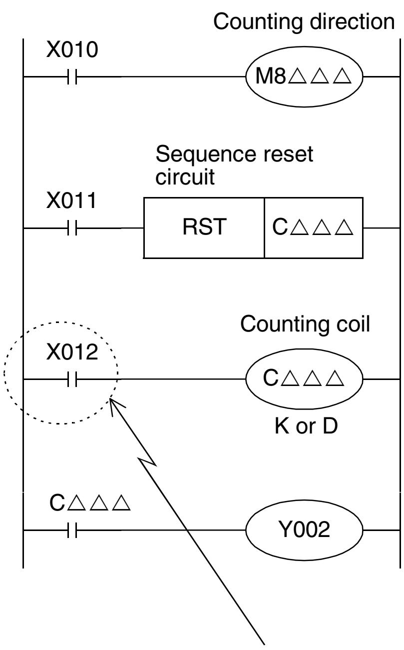
Use RST instruction to reset a counter or retentive type timer.

1) Program example of an internal counter



C0 up-counts the number of turning ON from OFF at X011. When the counting result reaches the set value K10, the output contact C0 is activated. Even if X011 changes from OFF to ON after that, the current value of the counter remains unchanged and the output contact remains activated.
For clearing the counter and returning the output contact, X010 is set to ON.
It is necessary to specify a constant K or data register number for indirect specification after OUT C instruction.
In the case of latched (battery backed) type counters, the current value and the operation status/reset status of the output contact are latched even after power failure.

2) Program example of a high speed counter



For 1-phase 1-input counters C235 to C245, use special auxiliary relays M8235 to M8245 for specifying the counting direction.
X010 in ON status: Specifies down counting.
X010 in OFF status: Specifies up counting.
When X011 turns ON, the output contact of the counter C△△△ is returned and the current value of the counter C△△△ is reset to "0".
In counters with reset input (C241, C242 ...), the same situation is achieved by interrupt operation when the corresponding reset input turns ON, but any program is not required for this operation.
When X012 turns ON, turning ON/OFF of a counting input X000 to X005 determined according to the counter number is counted.
In counters having start input (C244, C245 ...), counting is started only after the corresponding input turns ON.
When the current value of a counter increases and reaches the set value (K or contents of D), the output contact is set. When the current value decreases and reaches the set value, the output contact is reset.

As a contact driving the counting coil of a high speed counter, program a contact which is normally ON when high speed counting is executed.
If an input relay (X000 to X005) assigned for high speed counters is used for driving the counting coil, accurate counting cannot be achieved.

Cautions on using RST instruction for a jumped program, subroutine program or interrupt program

When RST instruction for a timer or counter is executed in a jumped program, subroutine program or interrupt program, the timer or counter may be kept in the reset status and the timer or counter may be disabled.

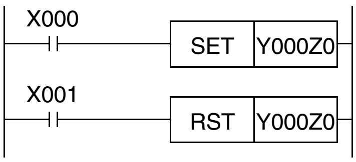
For details, refer to the following sections:

→ **For a jumped program, refer to Subsection 8.1.1.**
→ **For a subroutine program, refer to Subsection 8.2.1.**
→ **For an interrupt program, refer to Subsection 35.2.3.**

3. Indexing

Devices used in SET and RST instructions can be indexed with index registers (V)(Z).
(State relays (S), special auxiliary relays (M), 32-bit counters, "D□.b", and word devices cannot be indexed.)

Circuit program



List program

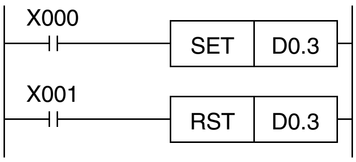
```
0000 LD   X000
0001 SET  Y000Z0
0004 LD   X001
0005 RST  Y000Z0
```

V0 to V7 and Z0 to Z7 are available for indexing.
When a used device is an input (X) or output (Y), the value of an index register (V or Z) is converted into octal, and then added.
Example: When Z0 is "20", Y024 turns ON or OFF.

4. Bit specification of a data register (D)

A bit in data register (D) can be specified as a device used in SET or RST instruction.

Circuit program



List program

```
0000 LD   X000
0001 SET  D0.3
0004 LD   X001
0005 RST  D0.3
```

When specifying a bit in data register, input "." after a data register (D) number, and then input a bit number (0 to F) consecutively.
Only 16-bit data registers are available.
Specify a bit number as "0, 1, 2, ... 9, A, B, ... F" from the least significant bit.
Example: In the example shown on the left, when X000 turns ON once, the bit 3 of D0 turns ON. When X001 turns ON, the bit 3 of D0 turns OFF.

Caution

When SET and RST instructions are executed for an output relay (Y) in a same operation, the result of the instruction located nearest the END instruction (which specifies the end of program) is output.

Errors

- When an I/O number used in SET or RST instruction does not exist due to indexing, M8316 (Non-existing I/O specification error) turns ON.
- When the device number of a device (M, T or C) other than I/O used in SET or RST instruction does not exist due to indexing, an operation error (error code: 6706) occurs.

7.14 NOP

Outline

NOP instruction specifies no operation.
When a program is erased completely, all steps are replaced with NOP instructions.
When NOP instruction is located between general instructions, PLCs ignore NOP instruction.
If NOP instructions are put in the middle of a program, fluctuation of step numbers is minimized when the program is changed or added. But excessive program steps are required.
Note that circuits are considerably changed if already written instructions are replaced with NOP instructions.

1. Instruction format

NOP

No Operation

Basic Instruction	Mnemonic	Operation Condition
1 step	NOP	Continuous Operation
	–	Pulse (Single) Operation

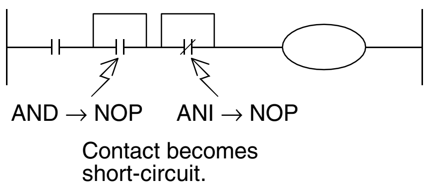
2. Applicable devices

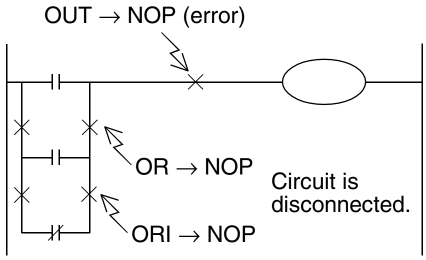
Instruction	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
NOP	There are no applicable devices.																							

Explanation of function and operation

1. NOP instruction (no operation or null step)

NOP instruction specifies no operation.
If NOP operation is written in the middle of a program, PLCs ignore it in executing the program.
When an existing program is replaced with NOP instructions, it means that former instructions are deleted.





1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

7.15 END

Outline

END instruction specifies the end of a program.
(Do not write the END instruction in the middle of a program.)

1. Instruction format

END
END

Basic Instruction	Mnemonic	Operation Condition
1 step	END	Continuous Operation
	–	Pulse (Single) Operation

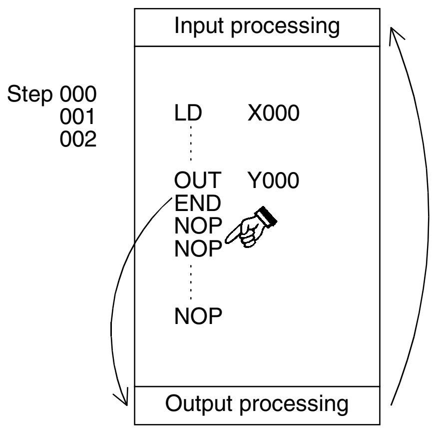
2. Applicable devices

Instruction	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
END	There are no applicable devices.																							

Explanation of function and operation

1. END instruction (program end, I/O refresh and return to step 0)

PLCs repeat "input processing → program execution → output processing". When END instruction is written at the end of a program, PLCs immediately execute the output processing without executing steps after END instruction.
If END instruction is not written at the end of a program, PLCs execute the program until the final step, and then execute the output processing.
At the first execution after the PLC mode was changed from STOP to RUN, PLCs start from END instruction.
When END instruction is executed, the watchdog timer (which checks the operation cycle) is refreshed.



Caution

Do not write END instruction in the middle of a program.
When a program is transferred from a programming tool, all steps after END instruction are replaced with NOP (no operation) instructions.

7.16 Number of Instruction Steps and Specified Devices

For ORB, ANB, MPS, MPD, MPP, MCR, INV, NOP and END instructions, refer to pages explaining these instructions.

Device		Instruction						
		LD, LDI, AND, ANI, OR, ORI	OUT	SET	RST	PLS, PLF	LDP, LDF, ANDP, ANDF, ORP, ORF	MC
Bit devices	X000 to X357	1	–	–	–	–	2	–
	Y000 to Y357	1	1	1	1	2	2	3
	M0 to M1535	1	1	1	1	2	2	3
	M1536 to M3583	2	2	2	2	2	2	3
	M3584 to M7679	3	3	3	3	3	3	4
	S0 to S1023	1	2	2	2	–	2	–
	S1024 to S4095	2	2	2	2	–	2	–
	T0 to T191, T200 to T245	1	3	–	2	–	2	–
	T192 to T199, T246 to T511	1	3	–	2	–	2	–
	C0 to C199	1	3	–	2	–	2	–
	C200 to C255	1	5	–	2	–	2	–
	Special auxiliary relays M8000 to M8255	1	2	2	2	–	2	–
	Special auxiliary relays M8256 to M8511	2	2	2	2	–	2	–
Bit devices with index	X000 to X357	3	–	–	–	–	–	–
	Y000 to Y357	3	3	3	3	3	–	–
	M0 to M7679	3	3	3	3	3	–	–
	T0 to T511	3	4	–	–	–	–	–
	S0 to S4095	–	–	–	–	–	–	–
	C0 to C199	3	4	–	3	–	–	–
	C200 to C255	–	–	–	–	–	–	–
	Special auxiliary relays M8000 to M8511	3	3	3	3	–	–	–
Word devices	D0 to D7999, Special data registers D8000 to D8511	–	–	–	3	–	–	–
	R0 to R32767	–	–	–	3	–	–	–
Word devices with index	D0 to D7999, Special data registers D8000 to D8511, R0 to R32767	–	–	–	–	–	–	–
Bit specification in word device	D□.b, Special auxiliary relays D□.b	3	3	3	3	–	3	–

1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

8. Program Flow – FNC 00 to FNC 09

FNC 00 to FNC 09 provide instructions mainly related to control flow of sequence programs such as conditional program execution and priority processing.

FNC No.	Mnemonic	Symbol	Function	Reference
00	CJ	CJ Pn	Conditional Jump	Section 8.1
01	CALL	CALL Pn	Call Subroutine	Section 8.2
02	SRET	SRET	Subroutine Return	Section 8.3
03	IRET	IRET	Interrupt Return	Section 8.4
04	EI	EI	Enable Interrupt	Section 8.5
05	DI	DI	Disable Interrupt	Section 8.6
06	FEND	FEND	Main Routine Program End	Section 8.7
07	WDT	WDT	Watchdog Timer Refresh	Section 8.8
08	FOR	FOR S	Start a FOR/NEXT Loop	Section 8.9
09	NEXT	NEXT	End a FOR/NEXT Loop	Section 8.10

8.1 FNC 00 – CJ / Conditional Jump

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

CJ or CJP instruction jumps to a pointer (P); The sequence program steps between CJ or CJP instruction and the pointer are not executed.
CJ and CJP instructions reduce the cycle time, and allow programs with double coils.

1. Instruction format

FNC 00 CJ	P

16-bit Instruction	Mnemonic	Operation Condition
3 steps	CJ	Continuous Operation
	CJP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
Pn	Pointer number (P) indicating the label number for the jump destination (n = 0 to 4095) (P63 jumps to END instruction.)	Pointer number

3. Applicable devices

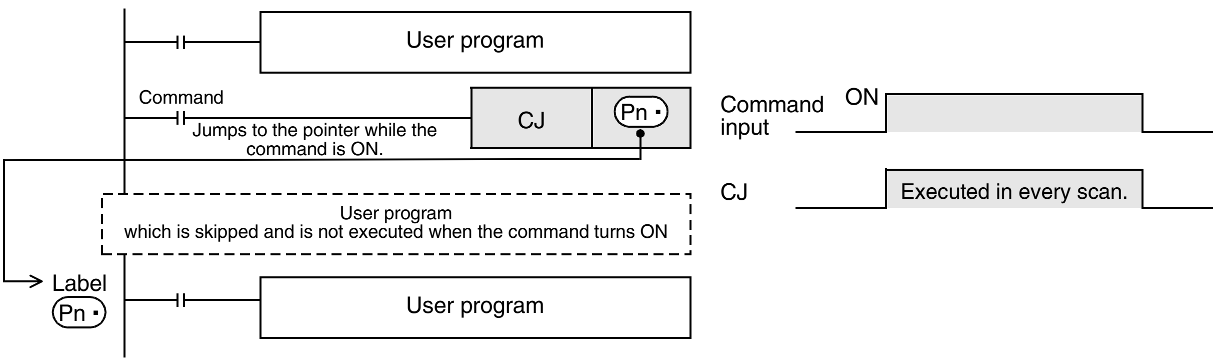
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
Pn																			✓					✓

Explanation of function and operation

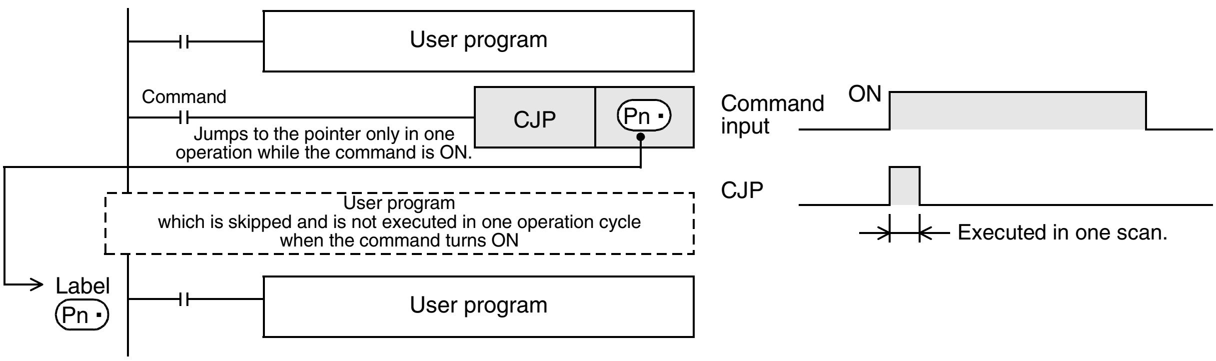
1. 16-bit operation (CJ and CJP)

While the command input is ON, CJ or CJP instruction executes a program with a specified label (pointer number).

1) In the case of CJ instruction



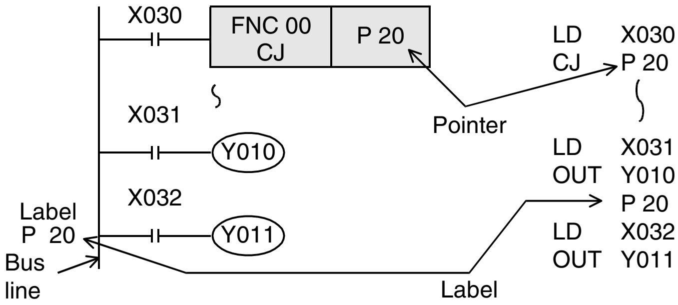
2) In the case of CJP instruction



Cautions

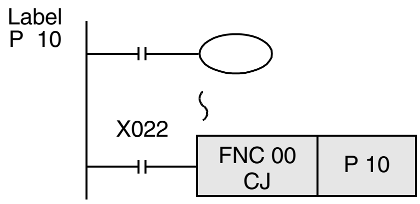
1. Relationship between the label input position and the list program

The figure below shows programming of a label.
When creating a circuit program, move the cursor to the left side of the bus line in the ladder diagram, and input a label (P) at the head of the circuit block.



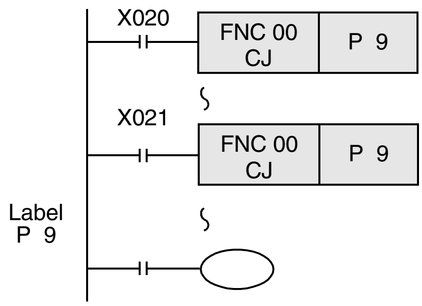
2. Programming a label in a smaller number step than CJ instruction

A label can be programmed in a smaller number step than CJ instruction. However, note that a watchdog timer error occurs when the scan time exceeds 200 ms (default setting).



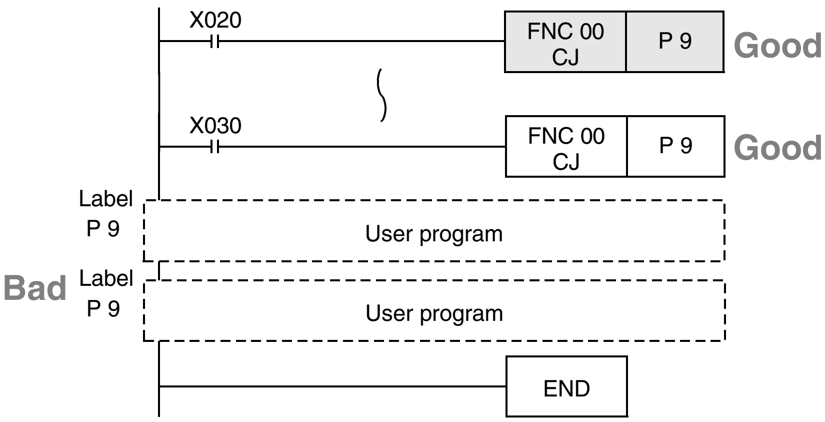
3. Jumping to one label from two or more CJ instructions

When the pointer number in operands is same and there is one label, the following operation is caused:
When X020 turns ON, the program execution jumps from CJ instruction corresponding to X020 to the label P9. When X020 turns OFF and X021 turns ON, the program execution jumps from CJ instruction corresponding to X021 to the label P9.



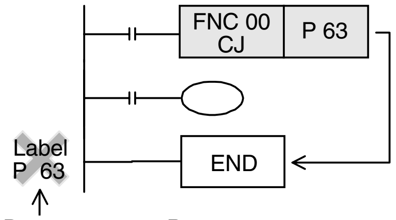
4. Using a label (P) two or more times

When a label number (including labels for CALL instructions described later) is used two or more times, an error is caused.



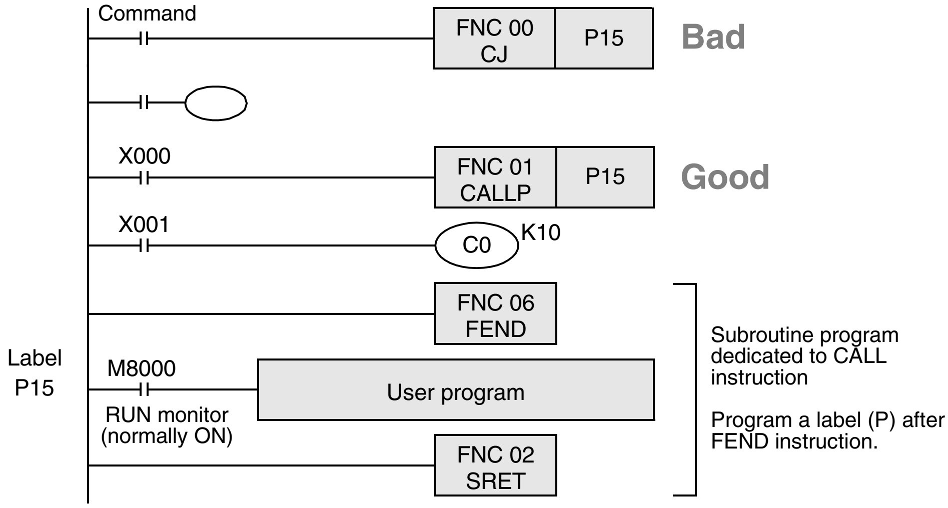
5. Label unnecessary for the pointer P63

The pointer P63 specifies jump to END step. Do not program P63.
If P63 is programmed, PLCs will display the error code 6507 (defective label definition) and stop.



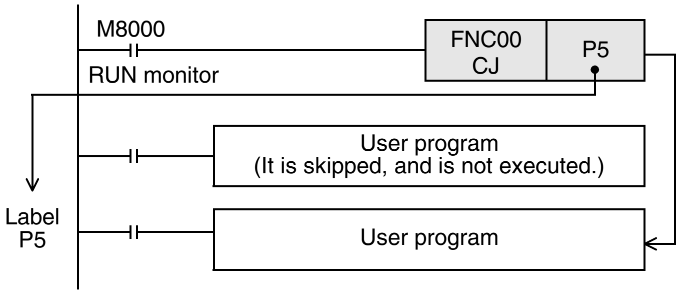
6. When jumping to a pointer for subroutine

Any label cannot be shared by CALL instruction and CJ instruction.



7. Unconditional jump if the command contact is normally ON

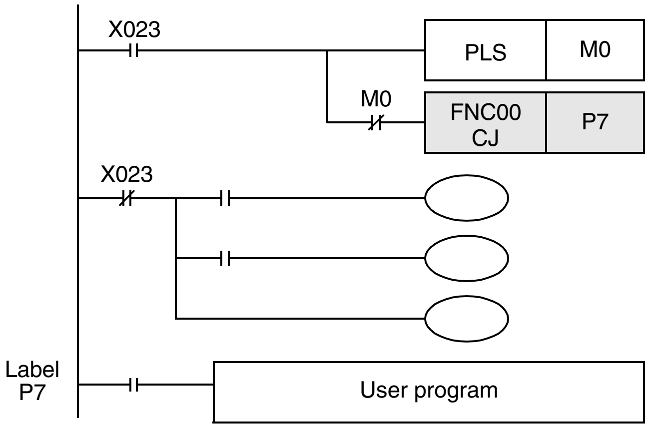
Because M8000 is normally ON while a PLC is operating, unconditional jump is specified when M8000 is used in the following example:



Program example

1. When jump is necessary after the OFF processing

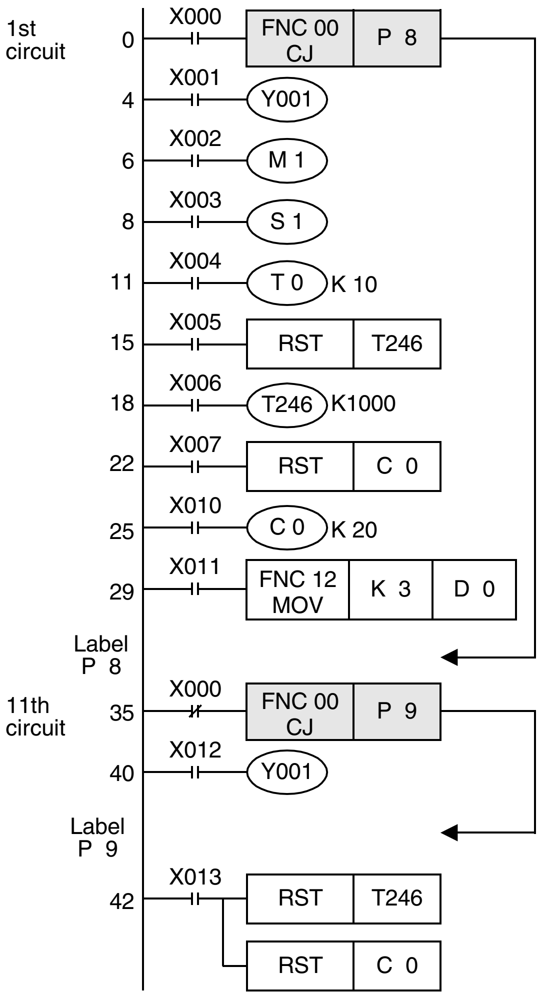
In one operation cycle after X023 changes to ON from OFF, CJ P7 instruction becomes valid.
By using this method, jump can be executed after all outputs between CJ P7 instruction and the label P7 turn OFF.



8.1.1 CJ instruction and operations of contact and coil

In the program example shown below, when X000 turns ON, the program execution jumps from CJ instruction in the first circuit to the label P8.
While X000 is OFF, jump is not executed; The program is executed from the 1st step in turn, and then the program execution jumps from CJ instruction in the 11th circuit to the label P9.
Instructions skipped by jump are not executed.

1. Circuit example 1 for explaining operations

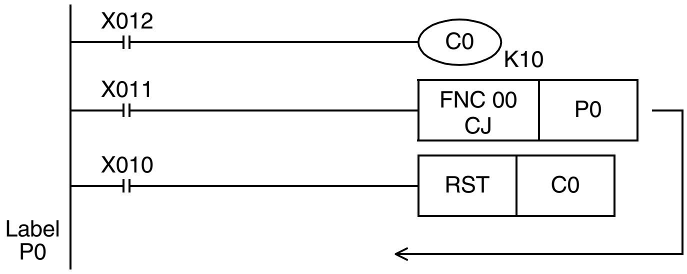


- Double coil operation of output Y001
  While X000 is OFF, output Y001 is activated by X001.
  While X000 is ON, output Y001 is activated by X012.
  Even in a program divided by conditional jumps, if a same coil (Y000 in this case) is programmed two or more times within the jump area or outside the jump area, such a coil is handled as double coil.
- When the reset (RST) instruction for the retentive type timer T246 is located outside the jump area
  Even if the counting coil (OUT T246) is jumped, reset (return of the contact and clearing of the current value) is valid.
- When the reset (RST) instruction for the counter C0 is located outside the jump area
  Even if the counting coil is jumped, reset (return of the contact and clearing of the current value) is valid.
- Operation of the routine timers T192 to T199
  A routine timer continues its operation even if it is jumped after the coil is driven, and the output contact is activated.
- Operation of the high speed counters C235 to C255
  A high speed counter continues its operation even if it is jumped after the coil is driven, and the output contact is activated.

When each input changes during jump in the above program, each coil executes the following operation:

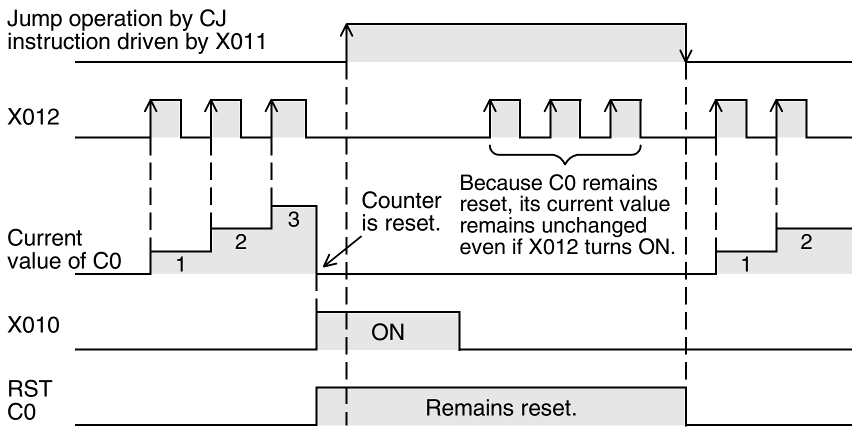
Classification	Contact status before jump	Coil operation during jump
Y, M, S (Y001, M1, S1)	X001, X002, X003 OFF	Y001, M1 and S1 turn OFF.
	X001, X002, X003 ON	Y001, M1 and S1 turn ON
10 ms timer and 100 ms timer (T0)	X004 OFF	Timer is not activated.
	X004 ON	Counting is paused (, and is restarted after X000 turns OFF).
1 ms timer (T246)	X005 OFF X006 OFF	Timer is not activated. The deactivation status is reset when X013 turns ON.
	X005 OFF X006 ON	Counting is continued (, and the contact is activated after X000 turns OFF).
Counter (C0)	X007 OFF X010 OFF	Counting is not activated. The deactivation status is reset when X013 turns ON.
	X007 OFF X010 ON	Counting is paused (, and is restarted after X000 turns OFF).
Applied instruction (MOV)	X011 OFF	FNC instruction is not executed during jump. But instructions FNC 52 to FNC 58 continue their operations.
	X011 ON	

2. Circuit example 2 for explaining operations (when only RST instruction for a timer or counter is jumped)

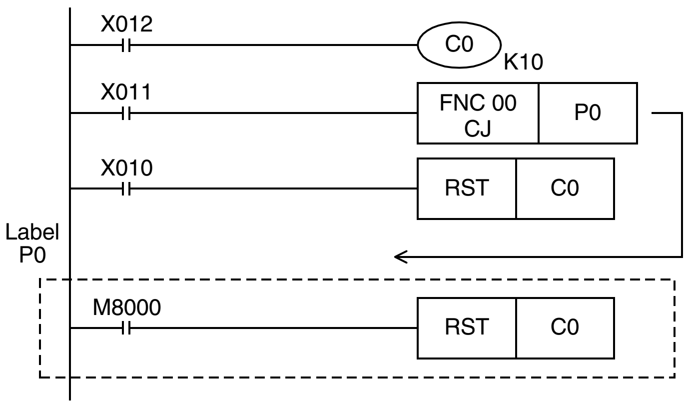


When X011 turns ON while RST instruction for the counter C0 is operating (X010 is ON), the program execution jumps after execution of the RST instruction due to CJ (FNC 00) instruction.
In this jump status, the counter C0 remains reset. Accordingly, the current value of C0 remains "0" even if X012 turns ON.
For clearing this reset status, it is necessary to turn OFF RST instruction for the counter C0. (Refer to the program shown below.)

Timing chart

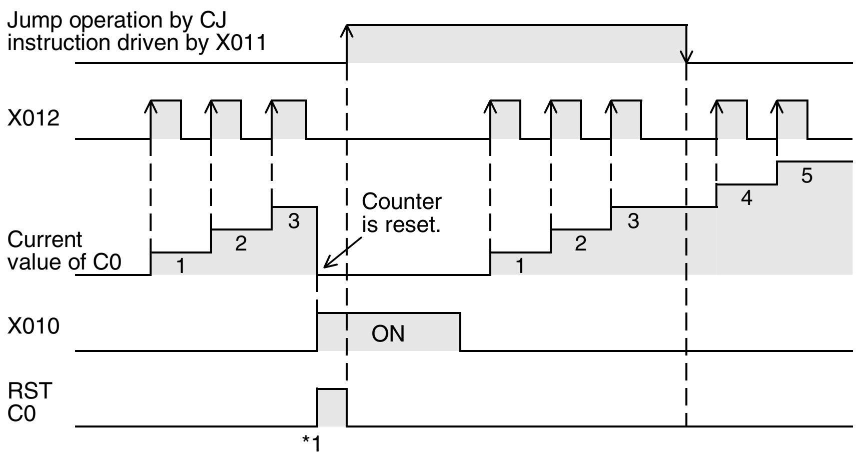


Program example for activating a timer and counter even during jump



Clears the reset status of the counter C0 during jump.

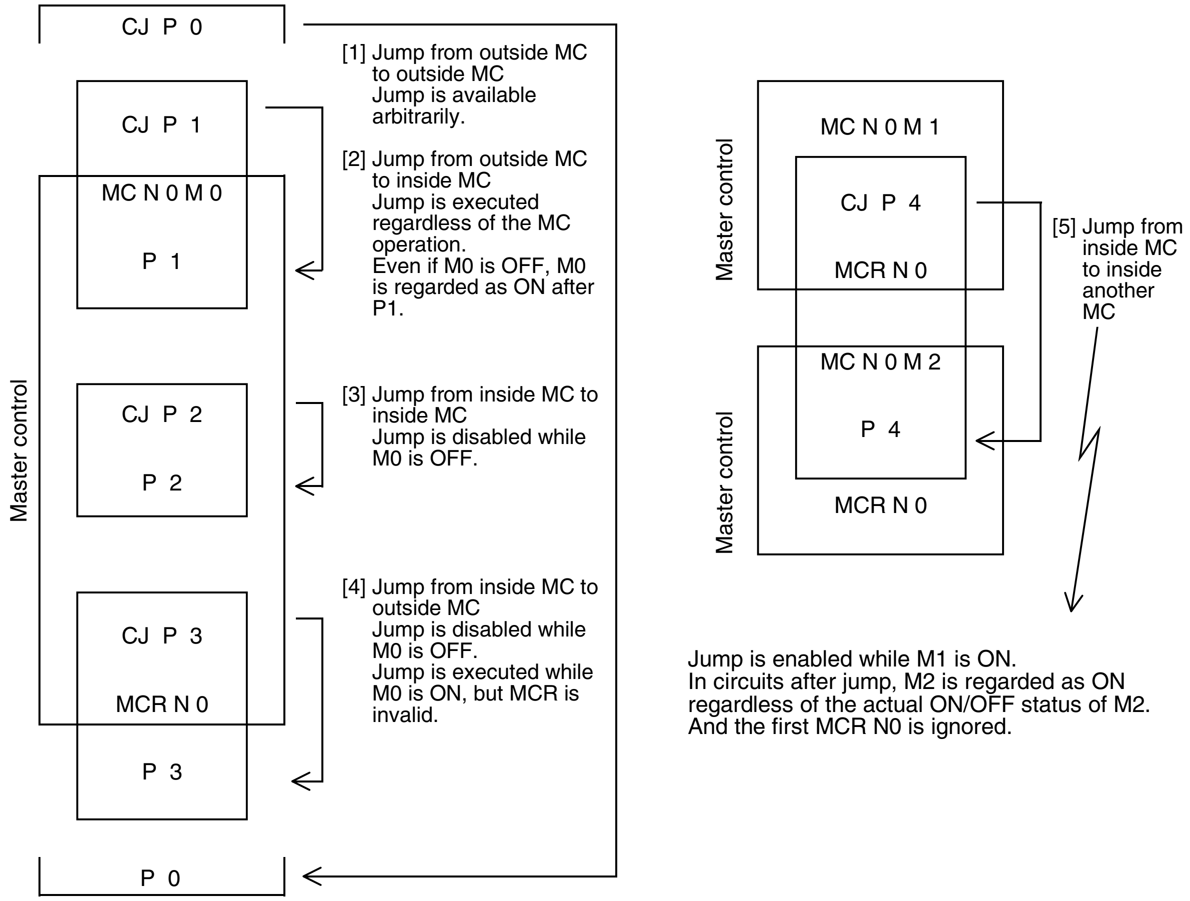
Timing chart



*1 In the same operation cycle as reset, the reset status of the counter C0 is cleared.

8.1.2 Relationship between master control instruction and jump instruction

The figure below shows the contents of operation and the relationship between the master control instruction. Avoid using [2], [4] and [5] because the operation will be complicated.



Jump is enabled while M1 is ON.
In circuits after jump, M2 is regarded as ON regardless of the actual ON/OFF status of M2.
And the first MCR N0 is ignored.

8.2 FNC 01 – CALL / Call Subroutine

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction calls and executes a program which should be processed commonly in a sequence program.
This instruction saves the number of program steps, and achieves efficient program design.
For creating a subroutine program, FEND (FNC 06) and SRET (FNC 02) instructions are required.

1. Instruction format

FNC 01 CALL P

16-bit Instruction	Mnemonic	Operation Condition
3 steps	CALL	Continuous Operation
	CALLP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
Pn•	Pointer number (P) indicating the label number for the jump destination (P0 to P62 and P64 to P4095)	Pointer number

For the pointer Pn• in CALL instruction, P0 to P62 and P64 to P4095 can be specified.
Because P63 is dedicated to CJ (FNC 00) instruction (for jump to END step), it cannot be used as a pointer for CALL (FNC 01) instruction.

3. Applicable devices

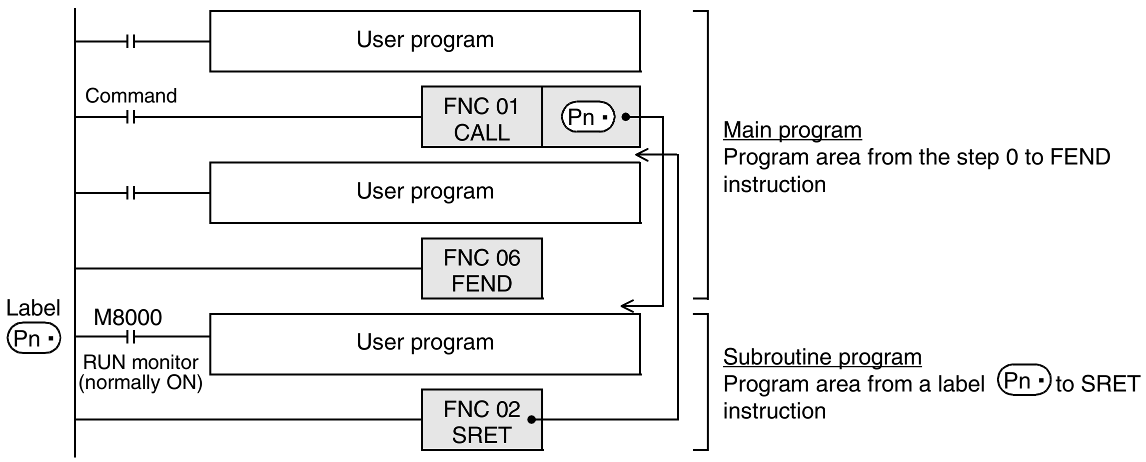
Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
Pn•																			✓					✓

Explanation of function and operation

1. 16-bit operation

While the command input is ON, CALL instruction is executed and the program execution jumps to a step with a label Pn•.
Then, a subroutine program with the label Pn• is executed.
When SRET (FNC 02) instruction is executed, the program execution returns to the step after CALL instruction.

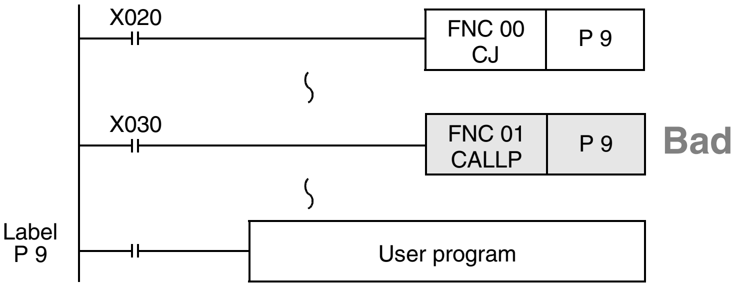
- At the end of the main program, put FEND instruction.
- Put a label (P) for CALL instruction after FEND instruction.



Caution

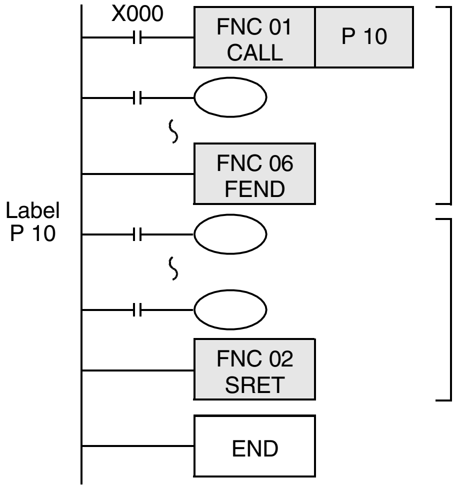
1. Using a label (P) number two or more times

In CALL instructions, a same number can be used two or more times in operands (P).
However, do not use a label (P) and number used in another instruction (CJ instruction).



Program examples

1. Example of fundamental use (no nesting)

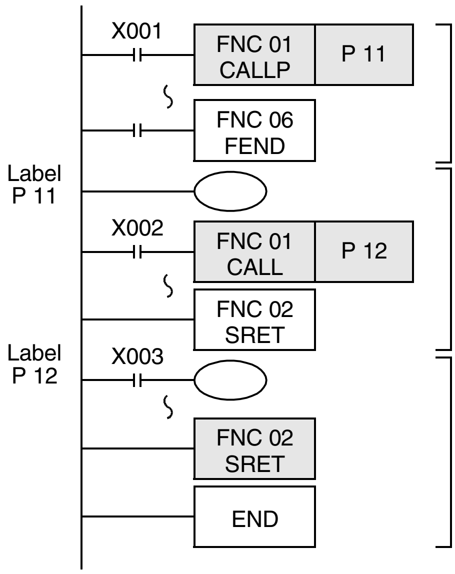


<u>Main program</u>
While X000 is ON, the program execution jumps to a step with the label P10.

<u>Subroutine program</u>
When SRET instruction is executed after the subroutine program has executed, the program execution returns to the original step +1.

2. Example of multiple CALL instructions in subroutines (multiple nesting)

CALL instruction can be used up to 4 times in subroutine programs. Nesting of up to five layers is allowed.



<u>Main program</u>
When X001 turns ON from OFF, the program execution jumps to the label P11 only once.

<u>Subroutine program 1</u>
When SRET instruction is executed, the program execution returns to the main program.
If X002 is ON while the subroutine program 1 is executed, the program execution jumps to a step with the label P12.

<u>Subroutine program 2</u>
The subroutine program with P12 is executed, and then the program execution returns to the subroutine program with P11 by SRET instruction.

8.2.1 Cautions on subroutines and interrupt routines

This section explains cautions on creating programs in subroutines and interrupt routines.
The explanation below is given for subroutines, but the situation also applies to interrupt routines.

1. When using timers in subroutines (or interrupt routines)

Use retentive type timers T192 to T199 in subroutines.
These timers execute counting when the coil instruction or END instruction is executed.
After a timer reaches the set value, the output contact is activated when the coil instruction or END instruction is executed.
Because general timers execute counting only when the coil instruction is executed, they do not execute counting if they are used in subroutines in which the coil instruction is executed only under some conditions.

2. When using retentive type 1 ms timers in subroutines (or interrupt routines)

If a retentive type 1 ms timer is used in a subroutine, note that the output contact is activated when the first coil instruction (or subroutine) is executed after the timer reaches its set value.

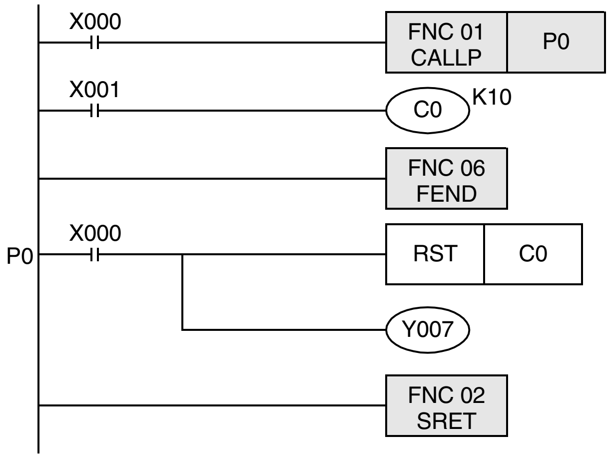
3. Countermeasures against latches of devices used in subroutines (or interrupt routines)

Devices which were set to ON in a subroutine are latched in the ON status even after the subroutine is finished. (Refer to the program example shown below.)
When RST instruction for a timer or counter is executed, the reset status of the timer or counter is latched also.
For turning OFF such a device latched in the ON status or for canceling such a timer or counter latched in the reset status, reset such a device in the main program after the subroutine is finished, or program a sequence for resetting such a device or for deactivating RST instruction in the subroutine. (Refer to the program example shown on the next page.)

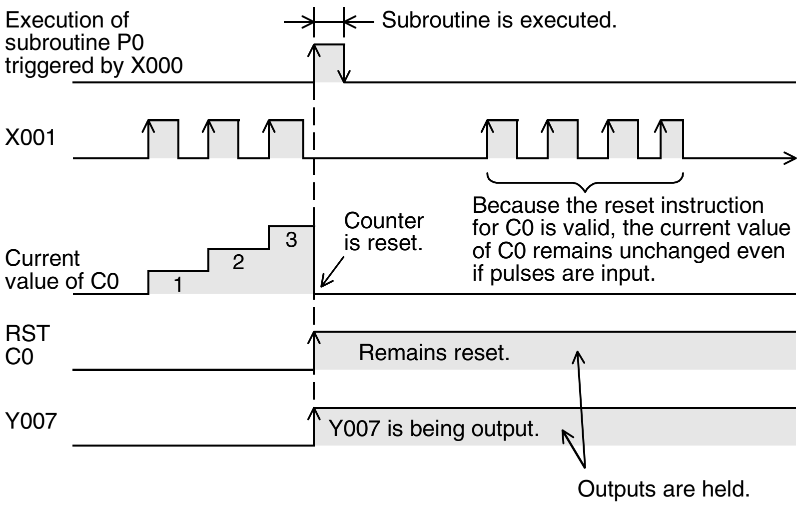
Example in which outputs are latched

In the following program example, the counter C0 is provided to count X001. When X000 is input, the subroutine P0 is executed only in one scan, and then the counter is reset and Y007 is output.

1) Program example

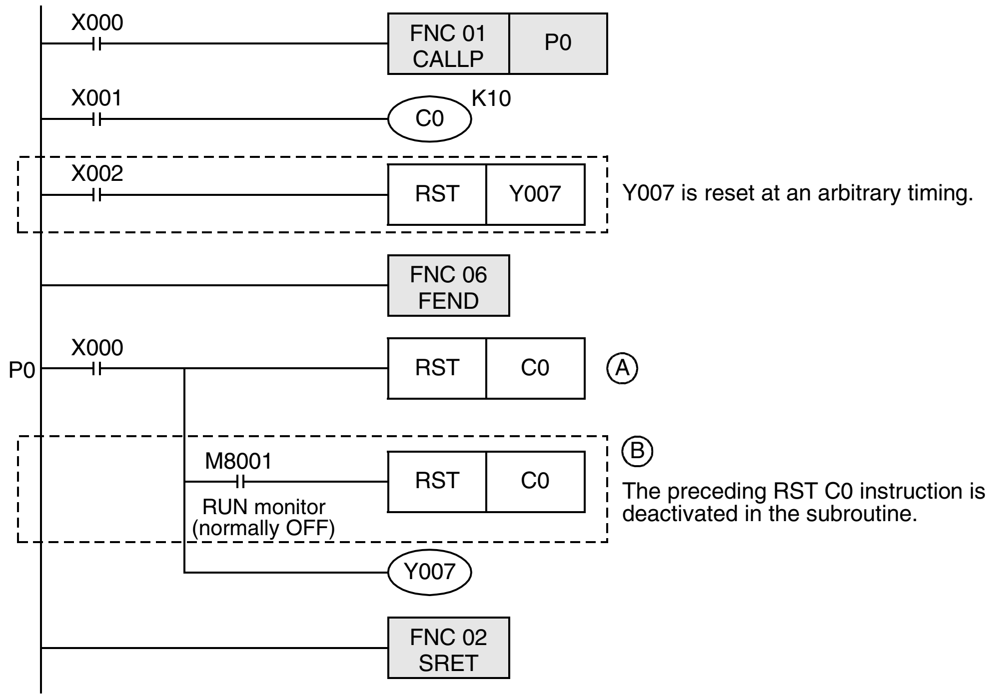


2) Timing chart

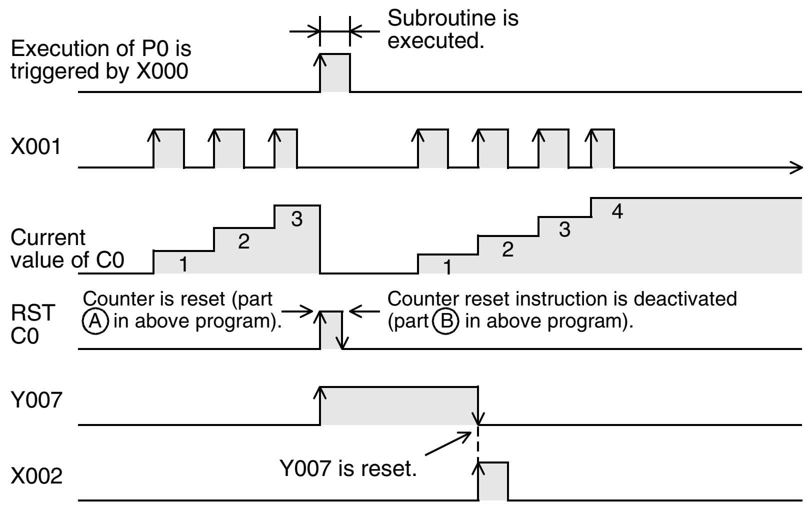


Example for resetting held outputs (countermeasures)

1) Program example



2) Timing chart



8.3 FNC 02 – SRET / Subroutine Return

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction returns the program execution from a subroutine to the main program.

1. Instruction format

FNC 02
SRET

Independent Inst.	Mnemonic	Operation Condition	
1 step	SRET	Continuous Operation	This instruction is the independent type, and does not require drive contact.

2. Set data

Operand type	Description	Data type
–	There is no set data.	–

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
–	There are no applicable devices.																							

Explanation of function and operation

When CALL instruction in the main program is executed, the program execution jumps to a subroutine.
SRET instruction returns the program execution to the main routine.

→ **Refer to Section 8.2.**

8.4 FNC 03 – IRET / Interrupt Return

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction returns the program execution from an interrupt routine to the main program.

1. Instruction format

FNC 03
IRET

Independent Inst.	Mnemonic	Operation Condition	
1 step	IRET	Continuous Operation	This instruction is the independent type, and does not require drive contact.

2. Set data

Operand type	Description	Data type
–	There is no set data.	–

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
–	There are no applicable devices.																							

Explanation of function and operation

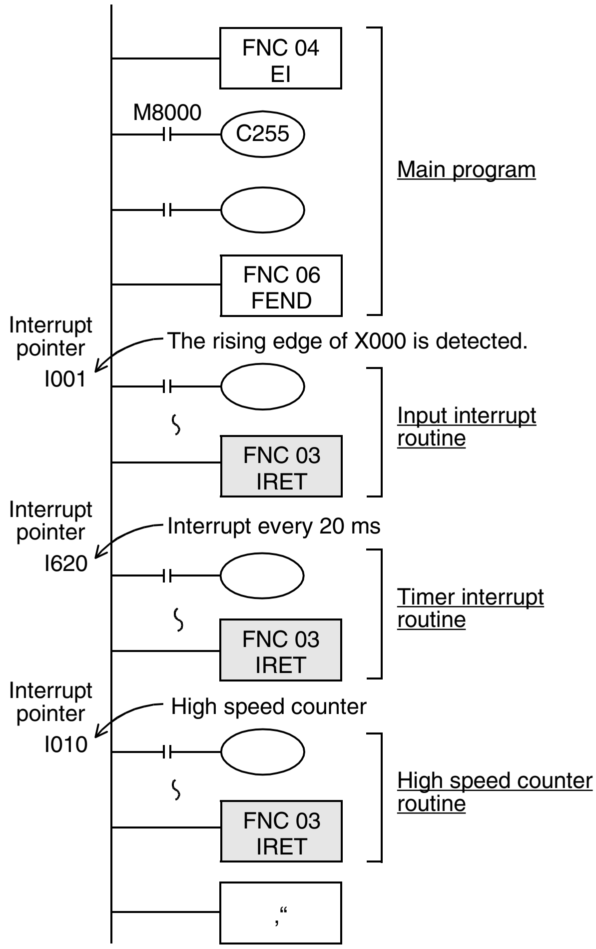
When an interrupt (input, timer or counter) is generated while the main program is executed, the program execution jumps to an interrupt (I) routine.
IRET instruction returns the program execution to the main routine.
The table below shows three types of jump to an interrupt routine.

1. Types of interrupt function

Function	Interrupt No.	Description	Reference
Input interrupt	I00* to I50*	Executes the interrupt processing when an input (X) signal turns ON or OFF.	Section 35.3 and Section 35.4
Timer interrupt	I6** to I8**	Executes the interrupt processing at a specified time interval (constant cycle).	Section 35.5
Counter interrupt	I010 to I060	Executes the interrupt processing when a high speed counter reaches it's set value.	Section 35.6

→ **For the interrupt function, refer to Chapter 35.**

Program example



Interrupts are usually disabled in PLCs.
Use EI instruction to enable interrupts.
When X000 turns ON while the main program is executed, instructions after the interrupt routine pointer I001 are executed, and the program execution returns to the original main program by IRET instruction.

The timer interrupt of the pointer I620 is executed every timer time of 20 ms, and the program execution is returned to the original main program by IRET instruction each time.

The high speed counter interrupt of the pointer I010 is executed when the current value of a high speed counter becomes equivalent to a value specified by DHSCS (FNC 53) instruction.
The program execution returns to the original main program by IRET instruction.

Make sure to program an interrupt pointer (I***) as a label after FEND instruction.

8.5 FNC 04 – EI / Enable Interrupt

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

Interrupts are usually disabled in PLCs.
This instruction enables interrupts in PLCs.
Use this instruction for using the input interrupt, timer interrupt and counter interrupt functions.

1. Instruction format

FNC 04
EI

Independent Inst.	Mnemonic	Operation Condition	
1 step	EI	Continuous Operation	This instruction is the independent type, and does not require drive contact.

2. Set data

Operand type	Description	Data type
–	There is no set data.	–

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
–	There are no applicable devices.																							

Explanation of function and operation

EI instruction is the independent type, and does not require command (drive) contact.

→ **For the interrupt function, refer to Chapter 35.**

8.6 FNC 05 – DI / Disable Interrupt

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction disables interrupts after interrupts were enabled by DI (FNC 05) instruction.

1. Instruction format

FNC 05
DI

Independent Inst.	Mnemonic	Operation Condition	
1 step	DI	Continuous Operation	This instruction is the independent type, and does not require drive contact.

2. Set data

Operand type	Description	Data type
–	There is no set data.	–

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
–	There are no applicable devices.																							

Explanation of function and operation

DI instruction is the independent type, and does not require command (drive) contact.

→ **For the interrupt function, refer to Chapter 35.**

Cautions

Interrupts (requests) generated after DI instruction are processed after EI (FNC 04) instruction is executed

8.7 FNC 06 – Main Routine Program End

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction indicates the end of the main program.

1. Instruction format

FNC 06 FEND	Independent Inst.	Mnemonic	Operation Condition	
	1 step	FEND	Continuous Operation	This instruction is the independent type, and does not require drive contact.

2. Set data

Operand type	Description	Data type
–	There is no set data.	–

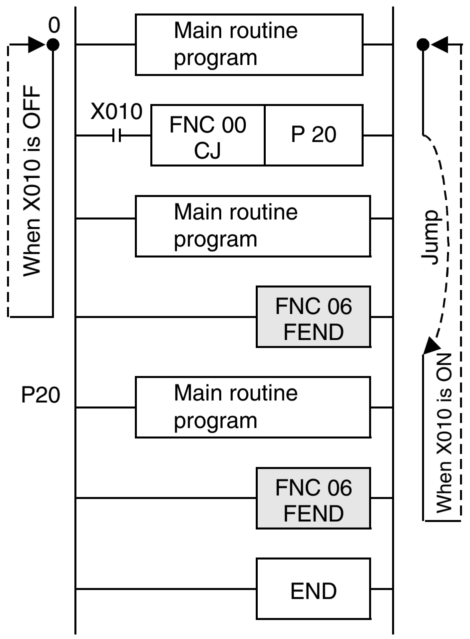
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
–	There are no applicable devices.																							

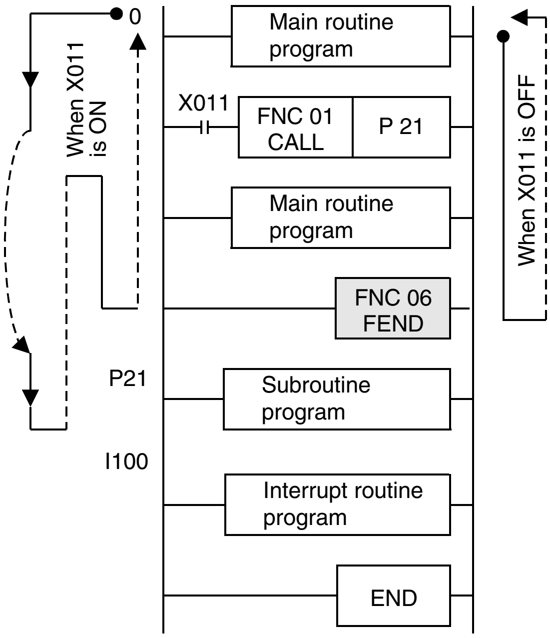
Explanation of function and operation

FEND instruction works in the same way as END instruction.
When FEND instruction is executed, output processing, input processing and watchdog timer refresh are executed, and then the program execution returns to the step 0.
FEND instruction is required in creating subroutine programs and interrupt programs.

1. In the case of CJ instruction



2. In the case of CALL instruction



Cautions

1. When FEND instruction is programmed two or more times
Put a subroutine program or interrupt routine program between last FEND instruction and END instruction.

2. When CALL or CALLP instruction is used
Put a label after FEND instruction. And the SRET instruction is required in every case.

3. When CALL or CALLP instruction is used
If FEND instruction is executed after CALL or CALLP instruction was executed and before SRET instruction is executed, an error is caused.

4. When FOR instruction is used
If FEND instruction is executed after FOR instruction was executed and before NEXT instruction is executed, an error is caused.

5. When the interrupt function (I) is used
Make sure to program an interrupt label (pointer) after FEND instruction. And IRET instruction is required in every case.

8.8 FNC 07 – WDT / Watchdog Timer Refresh

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction refreshes the watchdog timer in a sequence program.

1. Instruction format

FNC 07 WDT P

Independent Inst.	Mnemonic	Operation Condition
1 step	WDT	Continuous Operation
	WDTP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
–	There is no set data.	–

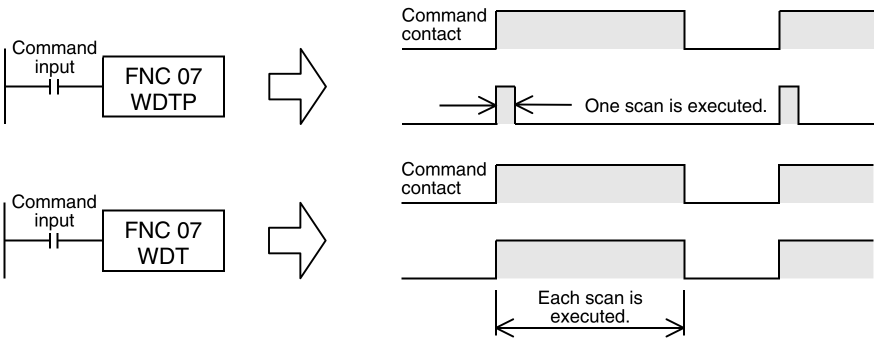
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
–	There are no applicable devices.																							

Explanation of function and operation

When the operation cycle (time until END or FEND instruction is executed after the step 0) of a PLC exceeds 200 ms, a watchdog timer error (indicating abnormal operation) occurs; The CPU error LED lights, and the PLC stops.

When the operation cycle is long, insert WDT instruction in the middle of the program to avoid the watchdog timer error.



Related device

Device	Name	Description
D8000	Watchdog timer time	Up to 32767 ms can be set in units of ms (initial value: 200 ms).

Cautions

1. When a watchdog timer error occurs

A watchdog timer error may occur in the following cases. To avoid the error, input a program shown below near the head step to extend the watchdog timer time, or shift FROM/TO instruction execution timing.

1) Caution when many special extension devices are connected
In such configuration that many special extension devices (such as positioning units, cam switches, analog units and link units) are connected, the buffer memory initialization time may become longer, thus the operation time may become longer, and a watchdog timer error may occur.

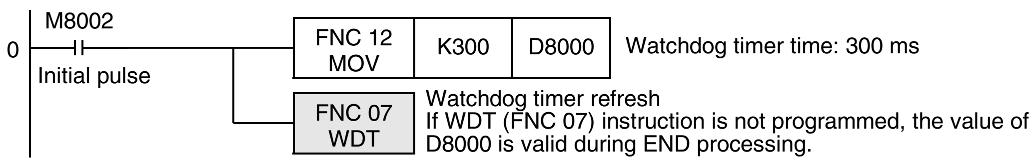
2) Caution when many FROM/TO instructions are driven at one time
When many FROM/TO instructions are executed or when many buffer memories are transferred, the extension time may become longer, and a watchdog timer error may occur.

3) Caution when there are many high speed counters (software counters)
When many high speed counters are provided and high frequency are counted at one time, the operation time may become longer, and a watchdog timer error may occur.

2. The watchdog timer time can be changed.

→ For details on changing watchdog timer time, refer to Subsection 36.2.2.

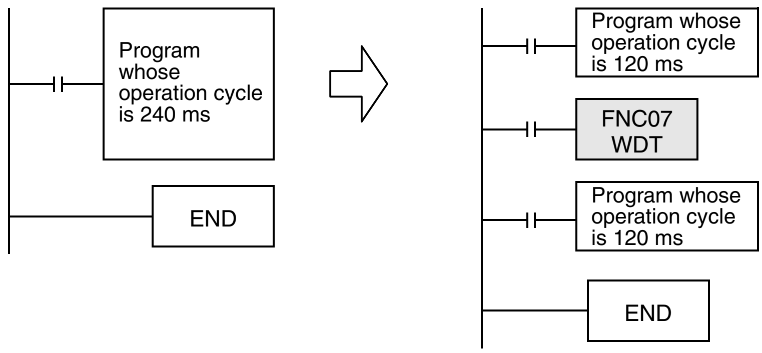
By overwriting the contents of D8000 (watchdog timer time), the watchdog timer detection time (initial value: 200 ms) can be changed.
By inputting the program shown below, the sequence program after this insertion is monitored by a new watchdog timer time.



Program examples

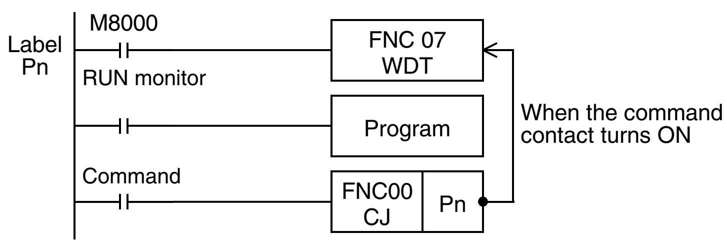
1. When the operation cycle is long and causes an error

For example, by dividing a program whose operation cycle is 240 ms into two portions and inserting WDT instruction between them, the operation cycle becomes less than 200 ms in both the former half portion and the latter half portion.



2. When a label (P) of CJ instruction is located in a step number smaller than the step number of CJ instruction

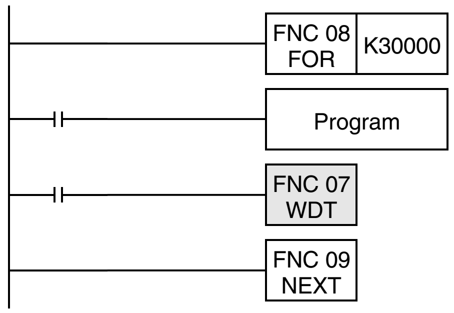
Put WDT instruction after the label (P).



If an input relay (X) is used as the command contact, input refresh is disabled, so the program execution cannot be returned from the area between P and CJ.
As the command contact, use such device that can be set to OFF in a program being jumped.

3. When FOR/NEXT instruction is repeated many times

Put WDT instruction between FOR and NEXT instructions.



8.9 FNC 08 – FOR / Start a FOR/NEXT Loop

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

FOR instruction specifies the number of repetition of the loop between FOR and NEXT (FNC 09) instructions.

1. Instruction format

FNC 08
FOR

16-bit Instruction	Mnemonic	Operation Condition
3 steps	FOR	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand type	Description	Data type
S•	Number of repetition of the loop between FOR and NEXT instructions [S• = K1 to K32767] (A value from –32768 to 0 is handled as "1".)	16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			

Explanation of function and operation

→ **For details, refer to NEXT (FNC 09) instruction.**

Related instruction

FOR instruction and NEXT (FNC 09) instruction are set as a pair in programming.

8.10 FNC 09 – NEXT / End a FOR/NEXT Loop

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

NEXT instruction specifies the end position of the loop.

1. Instruction format

FNC 09
NEXT

Independent Inst.	Mnemonic	Operation Condition	
1 step	NEXT	Continuous Operation	This instruction is the independent type, and does not require drive contact.

2. Set data

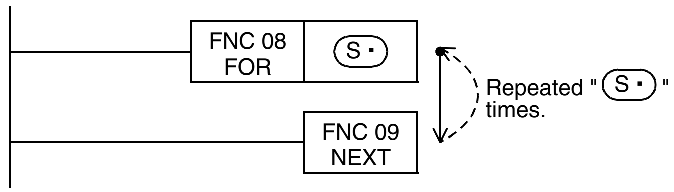
Operand type	Description	Data type
–	There is no set data.	–

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
–	There are no applicable devices.																							

Explanation of function and operation

The loop between FOR and NEXT instructions is repeated "n" times (which is specified by the source data).
After the loop is repeated by the specified number of times, steps after NEXT instruction are executed.



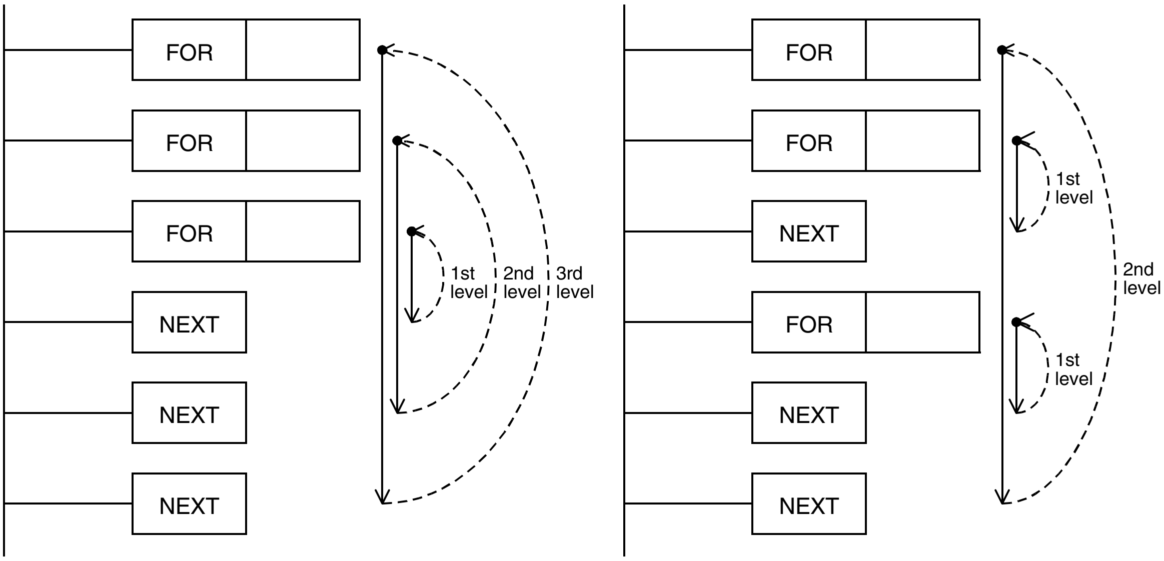
Related instruction

NEXT instruction and FOR (FNC 08) instruction are set as a pair in programming.

Caution

1. Limitation in the number of nesting

FOR-NEXT loop can be nested up to 5 levels.



Errors

1. Watchdog timer error

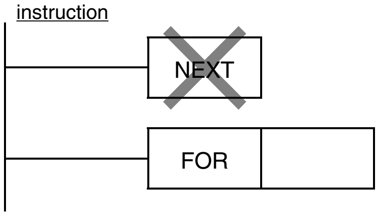
When FOR-NEXT loop is repeated many times, the operation cycle (D8010) is too long, and a watchdog timer error may occur. In such a case, change the watchdog timer time or reset the watchdog timer.

→ For details on changing the watchdog timer time, refer to Subsection 36.2.2.
→ For resetting the watchdog timer, refer to WDT (FNC 07) instruction.

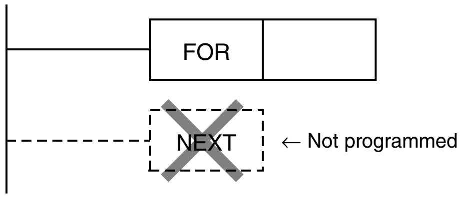
2. Examples of wrong programs

The following programs are regarded as errors.

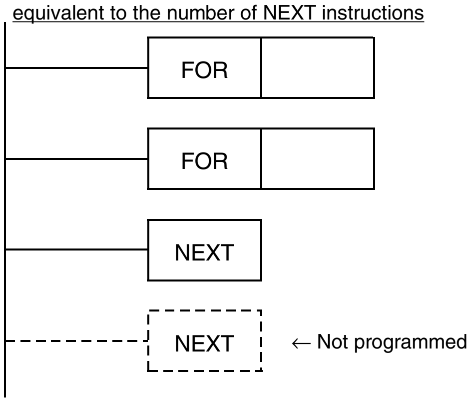
When NEXT instruction is located before FOR instruction



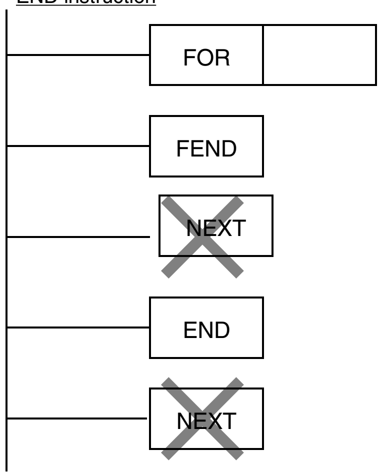
When NEXT instruction does not exist



When number of FOR instructions is not equivalent to the number of NEXT instructions

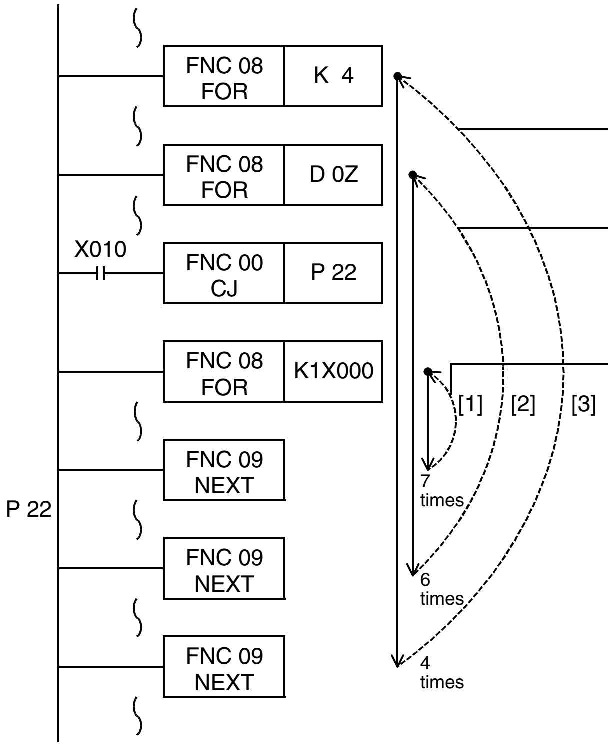


When NEXT instruction is located after FEND or END instruction



Program example

1. Program with three FOR-NEXT loops



The loop [3] is repeated 4 times.

When the data value (current value) of D0Z (D4 when Z is "4") is "6", the loop [2] is repeated 6 times.

- When X010 is OFF
  When K1X000 is "7", the loop [1] is repeated 7 times.
- When X010 is ON
  The program execution jumps to the pointer P22, and the loop [1] is skipped.

Number of times of repeating the loops [1], [2] and [3]

	X010 = OFF	X010 = ON
1]	7 × 6 × 4 =168 times	0 time
2]	6 × 4 = 24 times	24 times
3]	4 times	4 times

9. Move and Compare – FNC 10 to FNC 19

FNC 10 to FNC 19 provide fundamental data processing instructions such as data transfer and data comparison which are regarded as most important in applied instructions.

FNC No.	Mnemonic	Symbol	Function	Reference
10	CMP	CMP S1 S2 D	Compare	Section 9.1
11	ZCP	ZCP S1 S2 S D	Zone Compare	Section 9.2
12	MOV	MOV S D	Move	Section 9.3
13	SMOV	SMOV S m1 m2 D n	Shift Move	Section 9.4
14	CML	CML S D	Complement	Section 9.5
15	BMOV	BMOV S D n	Block Move	Section 9.6
16	FMOV	FMOV S D n	Fill Move	Section 9.7
17	XCH	XCH D1 D2	Exchange	Section 9.8
18	BCD	BCD S D	Conversion to Binary Coded Decimal	Section 9.9
19	BIN	BIN S D	Conversion to Binary	Section 9.10

9.1 FNC 10 – CMP / Compare

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction compares two values, and outputs the result (smaller, equal or larger) to bit devices (3 points).

→ For the contact comparison instruction, refer to Chapter 28.
→ For floating point comparison, refer to Section 18.1.

1. Instruction format

D	FNC 10 CMP	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	CMP	Continuous Operation
	CMPP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DCMP	Continuous Operation
	DCMPP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S1•	Data or device number handled as comparison value	16- or 32-bit binary
S2•	Date or device number handled as comparison source	16- or 32-bit binary
D•	Head bit device number to which comparison result is output	Bit

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D•		✓	✓			✓	▲												✓					

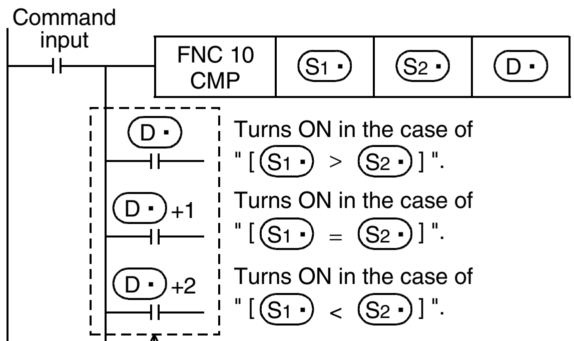
▲: "D□.b" cannot be indexing with index registers (V and Z).

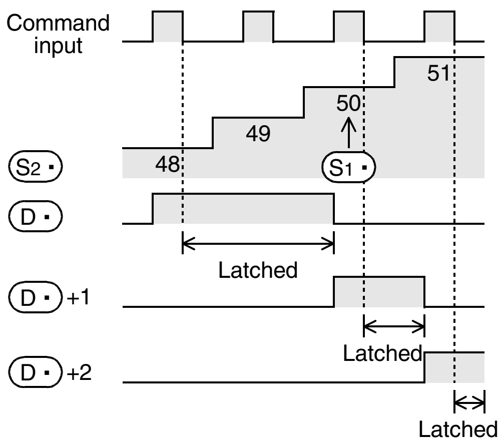
Explanation of function and operation

1. 16-bit operation (CMP and CMPP)

The comparison value S1• and the comparison source S2• are compared with each other. According to the result (smaller, equal or larger), either one among D•, D• +1 and D• +2 turns ON.

- The source data S1• S2• are handled as binary values.
- Comparison is executed algebraically. Example: $-10 < 2$





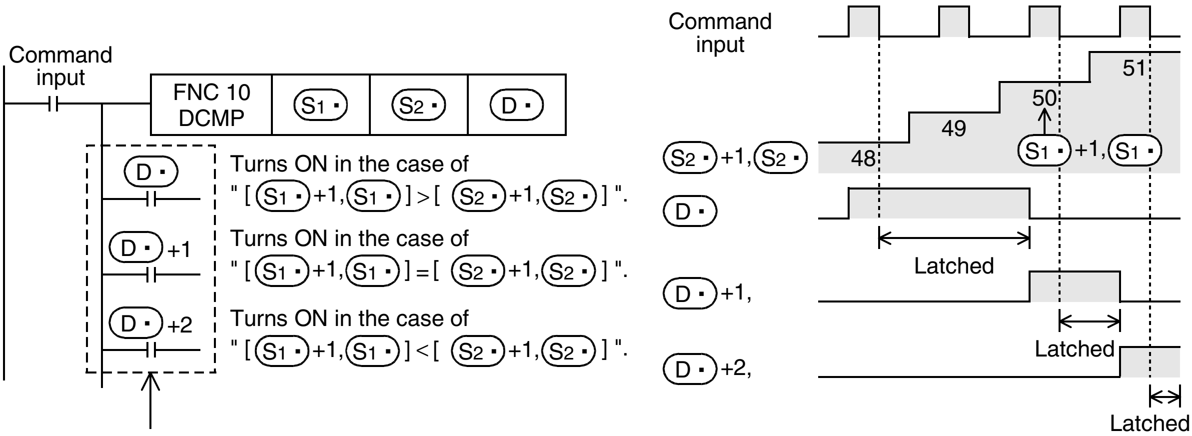
Even if the command input turns OFF and CMP instruction is not executed, D•, D• +1 and D• +2 latch the status just before the command input turns OFF from ON.

1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

2. 32-bit operation (DCMP and DCMPP)

The comparison value [S1·+1, S1·] and the comparison source [S2·+1, S2·] are compared with each other. According to the result (smaller, equal or larger), either one among D·, D·+1 and D·+2 turns ON.

- The source data [S1·+1, S1·] [S2·+1, S2·] are handled as binary values.
- Comparison is executed algebraically. Example: –125400 < 22466



Even if the command input turns OFF and DCMP instruction is not executed, D·, D·+1 and D·+2 latch the status just before the command input turns OFF from ON.

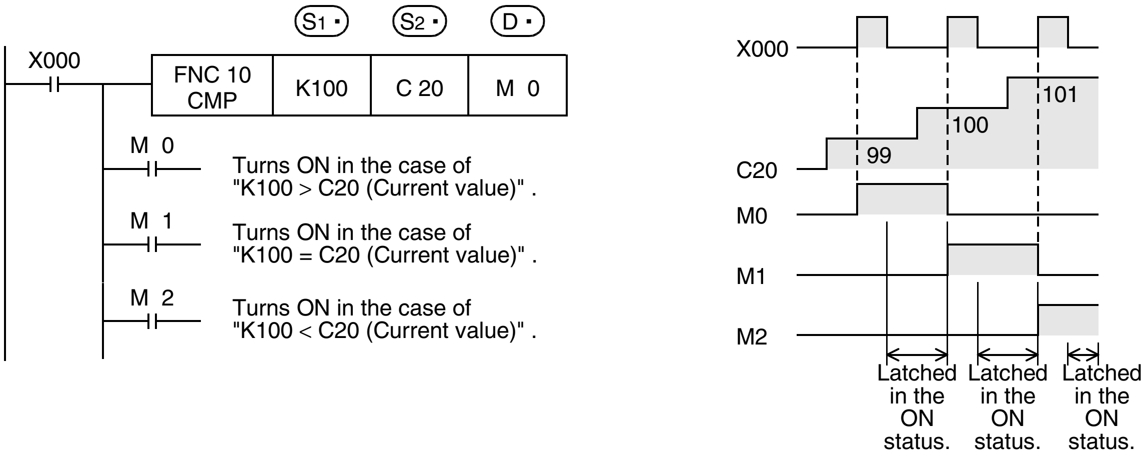
Caution

1. Number of occupied devices

From the device specified as D·, three devices are occupied. Make sure not to use those devices in another control.

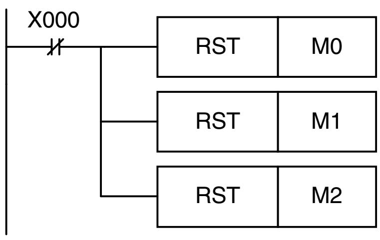
Program examples

1. When comparing the current value of a counter

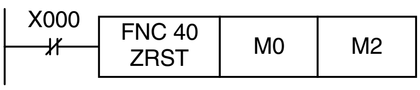


If it is necessary to clear the comparison result when the instruction is not executed, add the following contents under the above program.

1) RST instruction



2) ZRST instruction



M0 to M2 are reset.

9.2 FNC 11 – ZCP / Zone Compare

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction compares two values (zone) with the comparison source, and outputs the result (smaller, equal or larger) to bit devices (3 points).

→ **For the contact comparison instruction, refer to Chapter 28.**
→ **For floating point comparison, refer to Section 18.2.**

1. Instruction format

	FNC 11	
D	ZCP	P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	ZCP	Continuous Operation
	ZCPP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DZCP	Continuous Operation
	DZCPP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S1•	Data or device number handled as lower comparison value	16- or 32-bit binary
S2•	Data or device number handled as upper comparison value	16- or 32-bit binary
S•	Data or device number handled as comparison source	16- or 32-bit binary
D•	Head bit device number to which comparison result is output	Bit

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D•		✓	✓			✓	▲												✓					

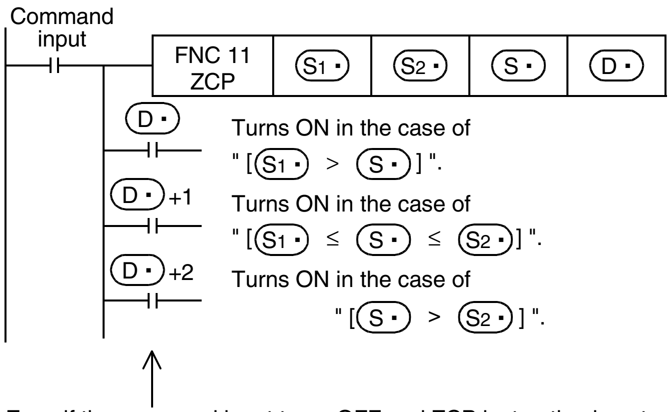
▲: "D□.b" cannot be indexing with index registers (V and Z).

Explanation of function and operation

1. 16-bit operation (ZCP and ZCPP)

The lower comparison value (S1·) and upper comparison value (S2·) are compared with the comparison source (S·). According to the result (smaller, within zone or larger), either one among (D·), (D·)+1 and (D·)+2 turns ON.

- Comparison is executed algebraically. Example: –10 < 2 <10

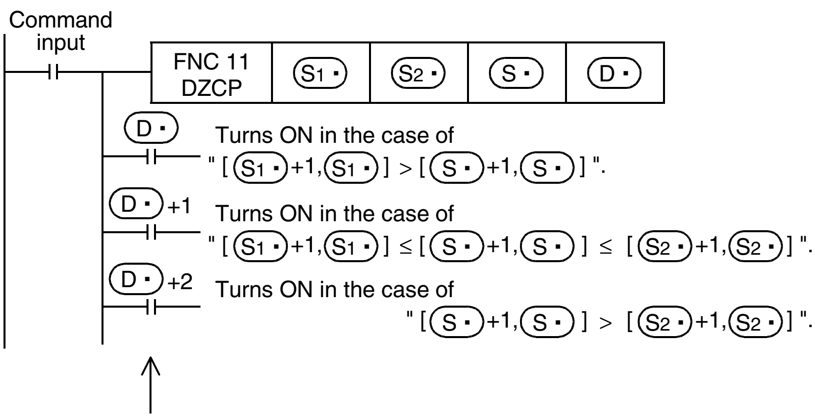


Even if the command input turns OFF and ZCP instruction is not executed, (D·), (D·)+1 and (D·)+2 latch the status just before the command input turns OFF from ON.

2. 32-bit operation (DZCP and DZCPP)

The lower comparison value [(S1·)+1, (S1·)] and upper comparison value [(S2·)+1, (S2·)] are compared with the comparison source [(S·)+1, (S·)]. According to the result (smaller, within zone or larger), either one among (D·), (D·)+1 and (D·)+2 turns ON.

- Comparison is executed algebraically. Example: –125400 < 22466 < 1015444



Even if the command input turns OFF and ZCP instruction is not executed, (D·), (D·)+1 and (D·)+2 latch the status just before the command input turns OFF from ON.

Cautions

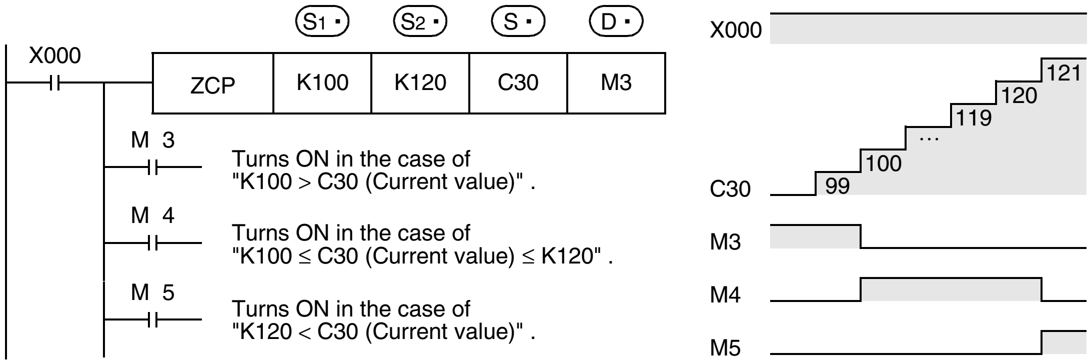
1. Number of occupied devices

From the device specified as (D·), three devices are occupied. Make sure not to use devices used in another control.

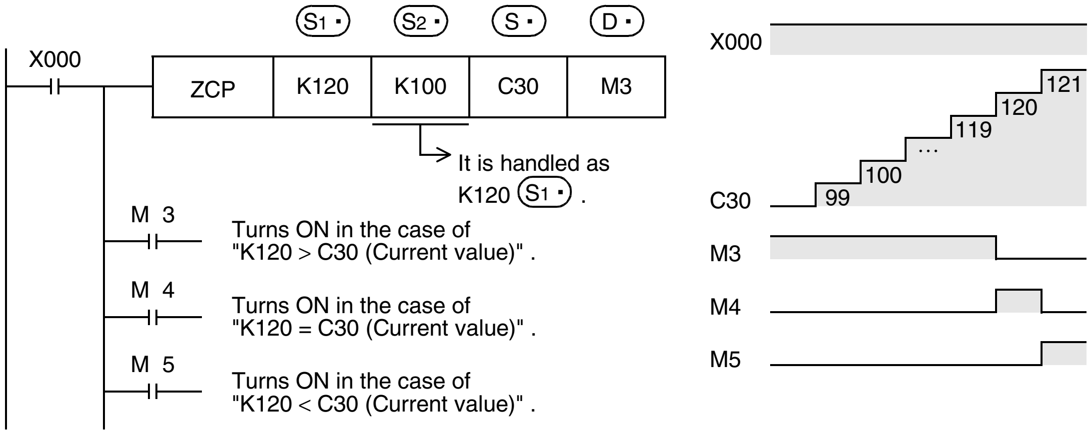
2. Upper comparison value and lower comparison value

The lower comparison value (S1·) should be smaller than the upper comparison value (S2·).

1) When the lower comparison value (S1·) is smaller than the upper comparison value (S2·)



2) When the lower comparison value (S1·) is larger than the upper comparison value (S2·)



9.3 FNC 12 – MOV / Move

Outline

This instruction transfers (copies) the contents of a device to another device.

1. Instruction format

D	FNC 12 MOV	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	MOV	Continuous Operation
	MOVP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DMOV	Continuous Operation
	DMOVP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S•	Transfer source data or device number storing data	16- or 32-bit binary
D•	Transfer destination device number	16- or 32-bit binary

3. Applicable devices

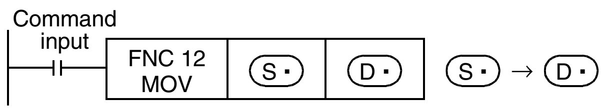
Operand Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D•									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

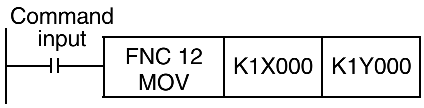
1. 16-bit operation (MOV and MOVP)

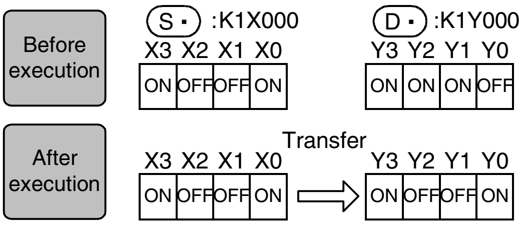
The contents of the transfer source S• are transferred to the transfer destination D•.

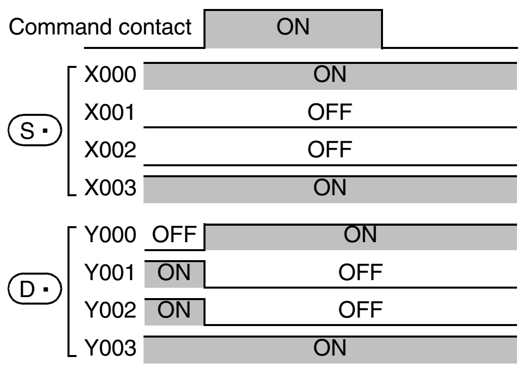
- While the command input is OFF, the transfer destination D• does not change.
- When a constant (K) is specified as the transfer source S•, it is automatically converted into binary.



When specifying digits of a bit device (K1X000 → K1Y000)
The bit device transfers a maximum of 16 points(multiple of 4).

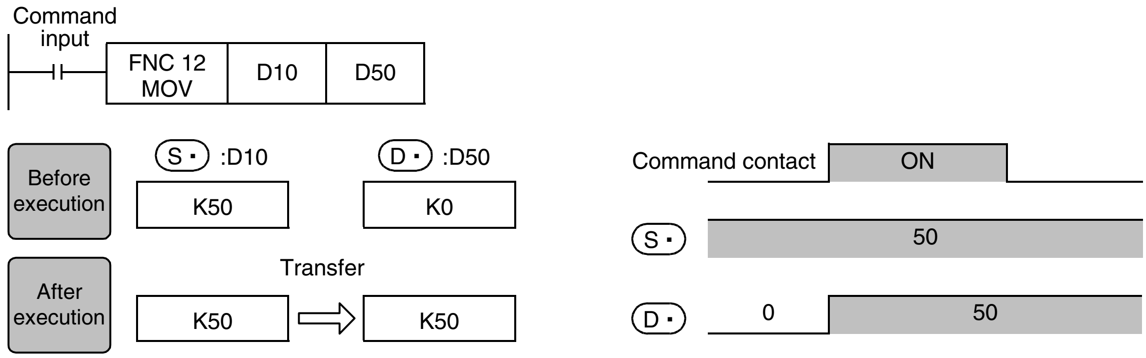






When a word device is specified

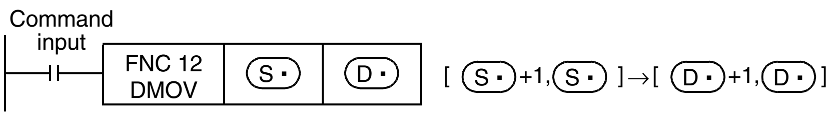
The word device transfers 1 point.



2. 32-bit operation (DMOV and DMOVP)

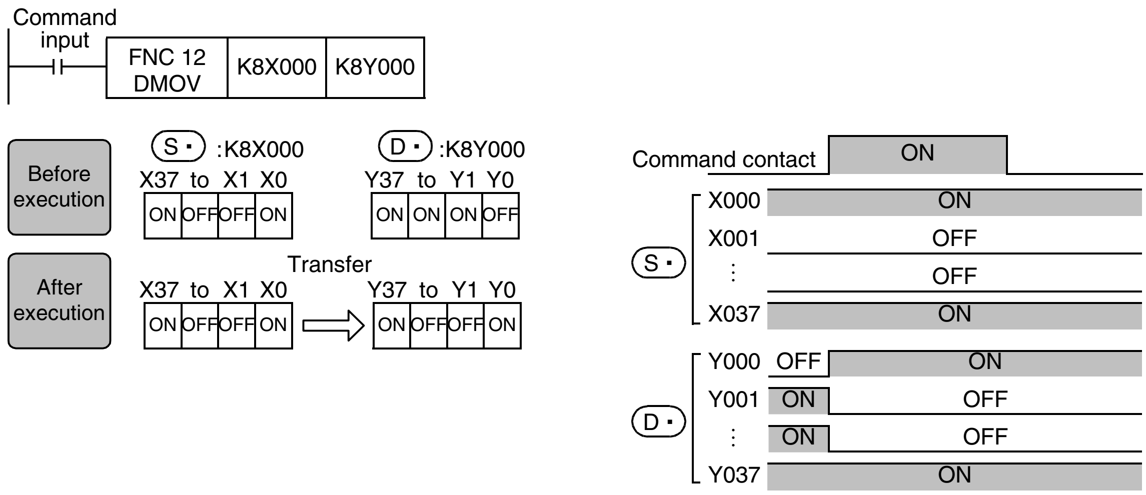
The contents of the transfer source [S·+1, S·] are transferred to the transfer destination [D·+1, D·].

- While the command input is OFF, the transfer destination D· does not change.
- When a constant (K) is specified as the transfer source [S·+1, S·], it is automatically converted into binary.



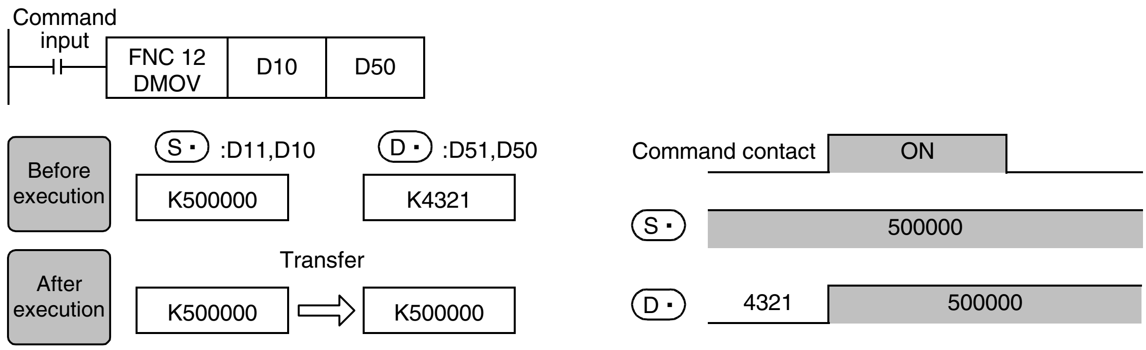
When specifying digits of a bit device (K8X000 → K8Y000)

The bit device transfers a maximum of 32 points (multiple of 4).



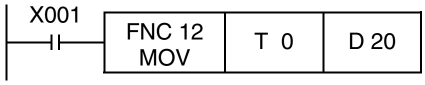
When a word device is specified

The word device transfers 1 point.



Program examples

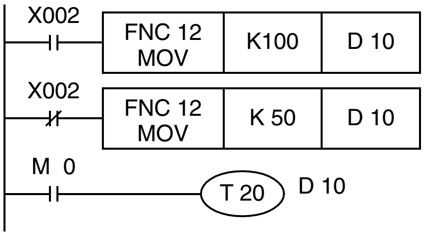
1. When reading the current value of a timer and counter



(Current value of T0) → (D20)
The operation is the same as a counter.

2. When indirectly specifying the set value of a timer or counter

As the set value of the timer T20, two values can be specified by turning ON or OFF the switch X002. For specifying more than two set values, more than one switch is required.



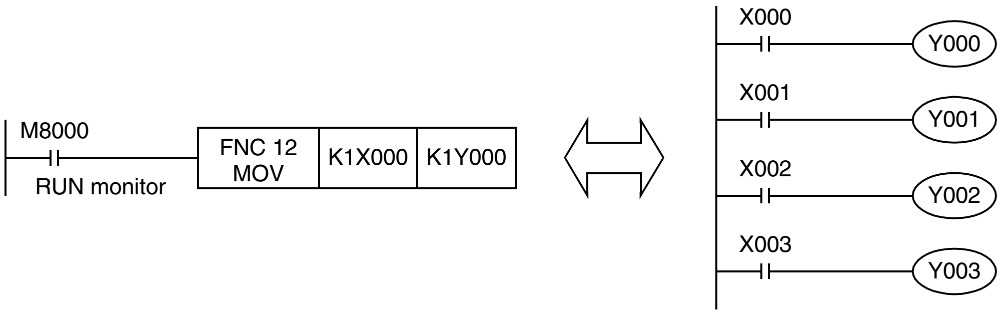
(K100) → (D 10)

(K50) → (D 10)

When X002 is ON, D10 = K100 (10-second timer).
When X002 is OFF, D10 = K50 (5-second timer).

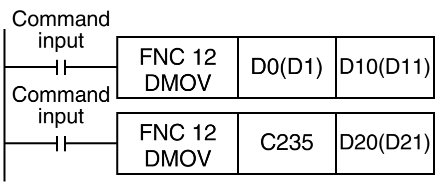
3. When transferring a bit device

The program written by basic instructions shown on the right can be expressed using MOV instruction as shown below.



4. When transferring 32-bit data

Make sure to use DMOV instruction for transferring the operation result of an applied instruction (such as MUL) whose operation result is output in 32 bits, and for transferring a 32-bit numeric value or transferring the current value of a high speed counter (C235 to C255) which is a 32-bit device.



(D 1,D 0) → (D 11,D 10)

(Current value of C235) → (D21, D20)

9.4 FNC 13 – SMOV / Shift Move

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction distributes and composes data in units of digit (4 bits).

1. Instruction format

FNC 13 SMOV P

16-bit Instruction	Mnemonic	Operation Condition
11 steps	SMOV	Continuous Operation
	SMOVP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand Type	Description	Data Type
(S•)	Word device number storing data whose digits will be moved	16-bit binary
m1	Head digit position to be moved	16-bit binary
m2	Number of digits to be moved	16-bit binary
(D•)	Word device number storing data whose digits are moved	16-bit binary
n	Head digit position of movement destination	16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S•)								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
m1																				✓	✓			
m2																				✓	✓			
(D•)									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
n																				✓	✓			

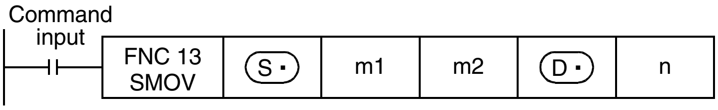
1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

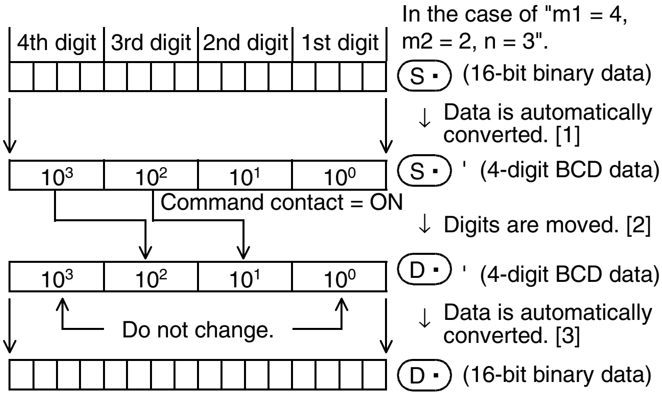
Explanation of function and operation

1. 16-bit operation (SMOV and SMOVP)

The contents of the transfer source (S·) and transfer destination (D·) are converted into 4-digit BCD (0000 to 9999) respectively. "m2" digits starting from "m1"th digit are transferred (composed) to the transfer destination (D·) starting from "n"th digit, converted into binary, and then stored to the transfer destination (D·).

- While the command input is OFF, the transfer destination (D·) does not change.
- When the command input turns ON, only the specified digits in the transfer destination (D·) are changed. The transfer source (S·) and unspecified digits in the transfer destination (D·) do not change.





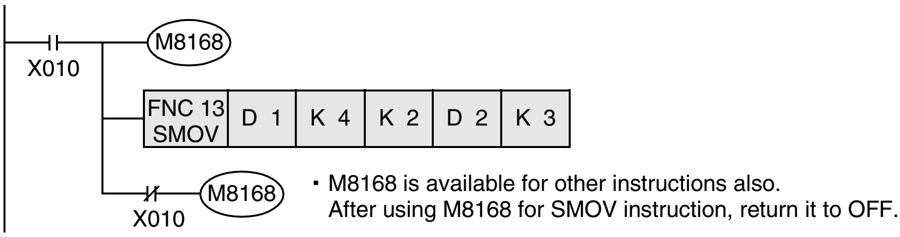
[1] (S·) is converted from binary into BCD.

[2] "m2" digits starting from "m1"th digit are transferred (composed) to (D·)' starting from "n"th digit.
The digits of 10^3 and 10^0 of (D·)' are not affected even if data is transferred from (S·)'.

[3] The composed data (BCD) is converted into binary, and stored to (D·).

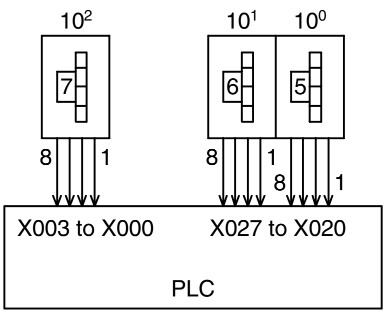
2. Extension function

When M8168 is set to ON first and then SMOV instruction is executed, conversion from binary to BCD is not executed.
Data is moved in units of 4 bits.

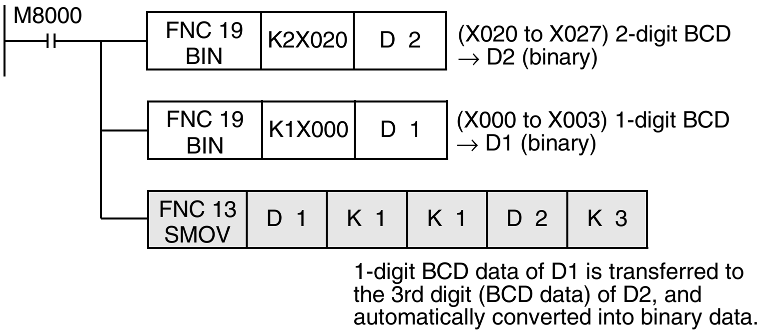


Program example

The data on three-digit digital switches are composed, and stored as binary data to D2.



Data on three digital switches connected to non-consecutive input terminals are composed.



9.5 FNC 14 – CML / Complement

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction inverts data in units of bit, and then transfers (copies) the inverted data.

1. Instruction format

D	FNC 14 CML	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	CML	Continuous Operation
	CMLP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DCML	Continuous Operation
	DCMLP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Data to be inverted or word device number storing data	16- or 32-bit binary
D·	Word device number storing inverted data	16- or 32-bit binary

3. Applicable devices

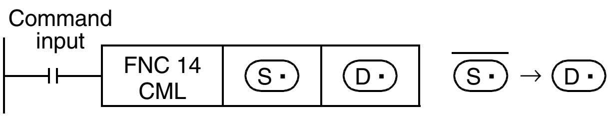
Oper-and Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D·									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

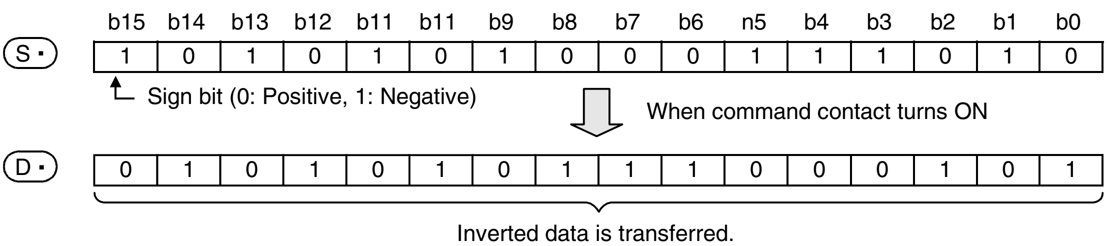
Explanation of function and operation

1. 16-bit operation (CML and CMLP)

Each bit of a device specified as S· is inverted (from 0 to 1 or from 1 to 0), and then transferred to D·.

- When a constant (K) is specified as S·, it is automatically converted into binary.
- This operation is useful when a logically inverted output is required as an output from a PLC.

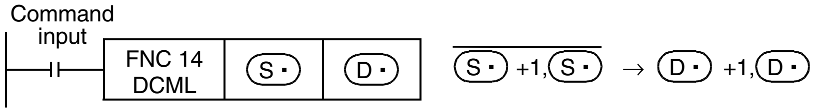




2. 32-bit operation (DCML and DCMLP)

Each bit of devices specified as [S· +1, S·] is inverted (from 0 to 1 or from 1 to 0), and then transferred to [D· +1, D·].

- When a constant (K) is specified as [S· +1, S·], it is automatically converted into binary.
- This operation is useful when a logically inverted output is required as an output from a PLC.



	b31	b30	b29	b28	b27	b26	b25	to	b7	b6	n5	b4	b3	b2	b1	b0
S·	1	0	1	0	1	0	1		0	0	1	1	1	0	1	0
D·	0	1	0	1	0	1	0		1	1	0	0	0	1	0	1

Sign bit (0: Positive, 1: Negative)

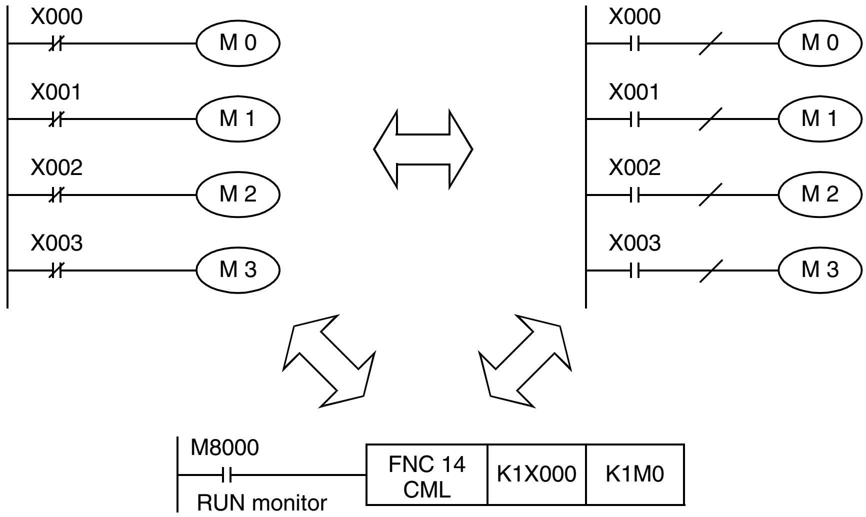
When command contact turns ON

Inverted data is transferred.

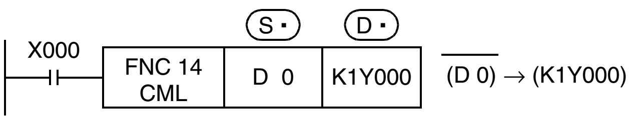
Program examples

1. When receiving an inverted input

The sequence program shown below can be written by CML instruction.



2. When four bits are specified for a device with digit specification



	Y015	b14	b13	b12	b11	b11	b9	b8	b7	b6	n5	b4	b3	b2	b1	b0
S· D 0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D·													0	1	0	1
K1Y000	Y017	Y016	Y015	Y014	Y013	Y012	Y011	Y010	Y007	Y006	Y005	Y004	Y003	Y002	Y001	Y000

Sign bit (0: Positive, 1: Negative)

X000 = ON

Do not change.

Inverted data is transferred.

9.6 FNC 15 – BMOV / Block Move

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction transfers (copies) a specified number of data at one time.

1. Instruction format

FNC 15 BMOV P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	BMOV	Continuous Operation
	BMOVP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand Type	Description	Data Type
S·	Transfer source data or device number storing data	16-bit binary
D·	Transfer destination device number	16-bit binary
n	Number of transferred points (including file registers) [n ≤ 512]	16-bit binary

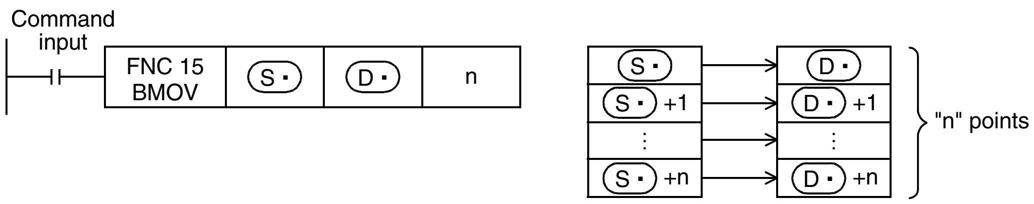
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
D·									✓	✓	✓	✓	✓	✓	✓	✓			✓					
n														✓						✓	✓			

Explanation of function and operation

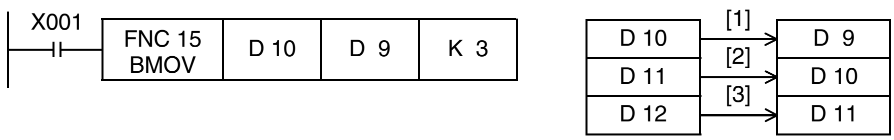
BMOV instruction transfers "n" points of data from S· to D· at one time.

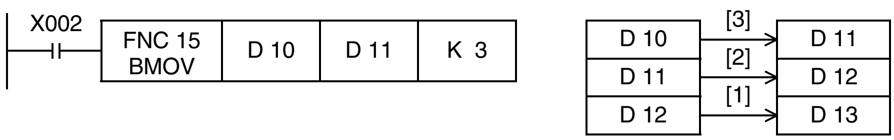
- If the device number range is exceeded, data is transferred within the possible range.



Transfer is enabled even if the transfer number range is overlapped.

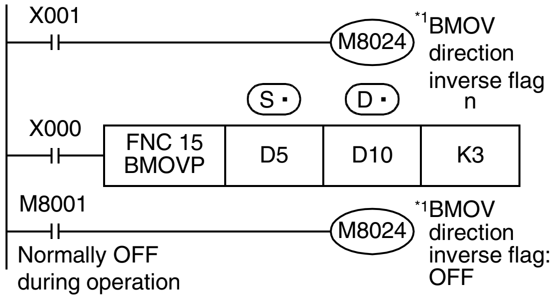
To prevent overwriting before transfer of source data, data is automatically transferred in the order "[1] → [2] → [3]" according to the number overlap status.





Extension function (bi-directional transfer function)

By controlling the direction inverse flag M8024*1 for BMOV (FNC 15) instruction, data can be transferred in two directions in one program.

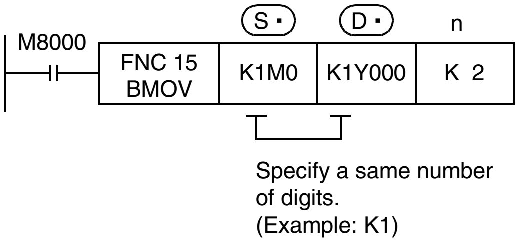


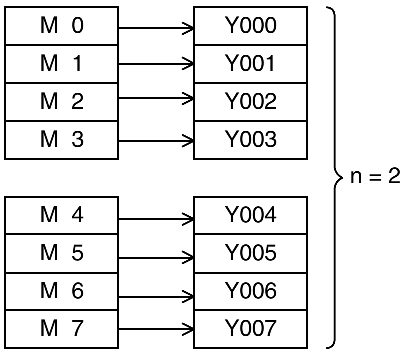
BMOV direction inverse flag	Transfer direction	S·, D·
M8024*1 : OFF	S· → D·	D5 → D10 D6 → D11 D7 → D12
M8024*1 : ON	S· ← D·	D5 ← D10 D6 ← D11 D7 ← D12

*1. M8024 is cleared when the PLC mode is changed from RUN to STOP.

Caution

When specifying digits of bit devices, specify a same number of digits for S· and D·.





9.6.1 Function of transfer between file registers and data registers

BMOV (FNC 15) instruction has a special function for file registers (D1000 and later).

→ **For details on file registers, refer to Section 4.8.**

1. What are file registers

By parameter setting, D1000 to D7999 can be handled as file registers, and written to and read from the program memory area.

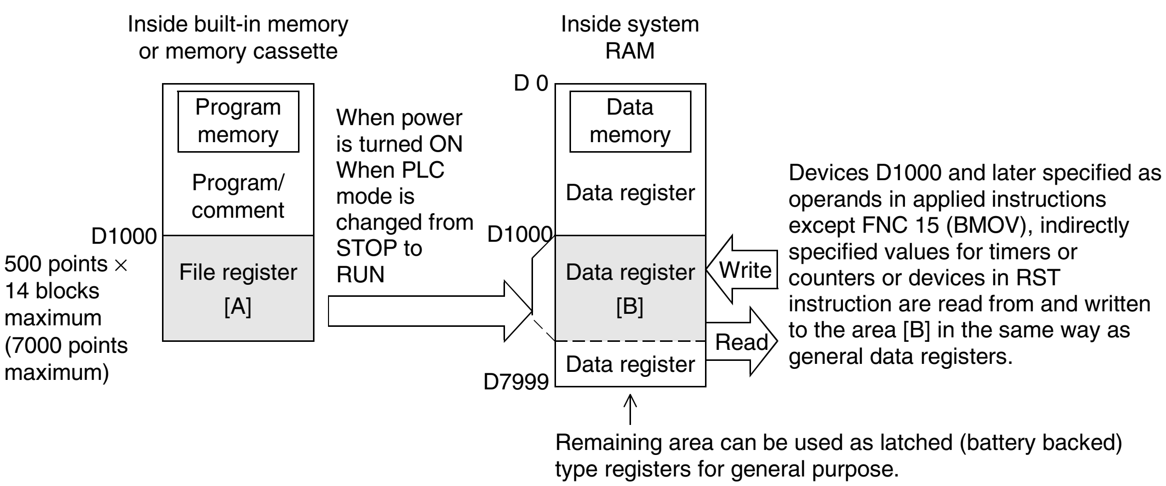
1) Outline of setting
File registers (D1000 to D7999) do not exist in the initial status. They are valid only when some number of file registers are secured by parameter setting in a programming tool.

2) Number of file registers
In parameter setting, set 500 file registers as 1 block.
1 to 14 blocks (each of which has 500 file registers) can be set.
1 block occupies 500 steps in the program memory area.

3) Difference between BMOV (FNC 15) instruction and other instructions
The table below shows the difference between BMOV (FNC 15) instruction and other instructions with regard to file registers (D1000 and later).

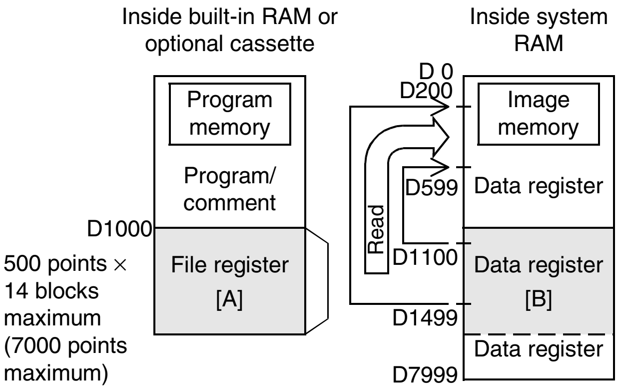
Instruction	Contents of transfer	Remarks
BMOV instruction	Can read from and write to the file register area [A] inside the program memory.	–
Other applied instructions	Can read from and write to the data register area [B] inside the program memory in the same way as general data registers.	Because the data register area [B] is provided inside the system RAM in PLCs, its contents can be arbitrarily changed without regard to the optional memory format.

When restoring the power, data registers set as file registers are automatically copied from the file register area [A] to the data register area [B].

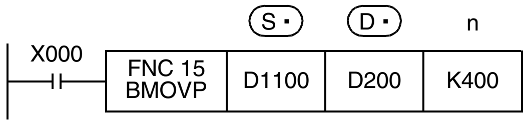


2. Cautions on use

1) When updating the contents of a file register with a same number (same-number update mode), make sure that the file register number is equivalent between S· and D·.
2) When using file registers in the same-number update mode, make sure that the number of transfer points specified by "n" does not exceed the file register area.
3) If the file register area is exceeded while file registers are used in the same-number update mode, an operation error (M8067) is caused and the instruction is not executed.
4) In the case of indexing (in the same-number update mode)
   When S· and D· are indexing with index, the instruction is executed if the actual device number is within the file register area and the number of transfer points does not exceed the file register area.
5) Handling of the flash memory
   (memory cassette FX3U-FLROM-16, FX3U-FLROM-64 and FX3U-FLROM-64L)
   When changing the contents of file registers secured inside the flash memory, observe the following condition:
   - Set to OFF the protect switch in the memory cassette.
   - When writing data using a continuous operation type instruction in a program, data is written to the flash memory in every operation cycle of the PLC.
     For preventing this, make sure to use a pulse operation type instruction (BMOVP) so that the number of times of writing is reduced.
   - It takes 66 to 132 ms to write data of one serial block (500 points) to the flash memory. Execution of the program is paused during this period. Because the watchdog timer is not refreshed at this time, it is necessary to take proper countermeasures such as insertion of WDT instruction in a user program.
6) File register operation
   File registers are secured inside the built-in memory or memory cassette.
   Different from general data registers, file registers can be read and written only by peripheral equipment or BMOV (FNC 15) instruction.
7) If a file register is not specified as the destination in BMOV (FNC 15) instruction, the file register is not accessed.
   a) Outline of memory operation



   b) Program example
   When X000 is set to ON, the data register area [B] is read.



   A file register can be specified as D·. But if a same number with S· is specified, the same-number register update mode is selected.
   However, even if a file register having different number is specified for S· and D· respectively, data cannot be transferred from the file register area to another file register area. In such a case, read the contents of a file register specified as S· in the same-number register update mode to the data register area [B] once, and then write the data.

→ **For the same-number register update mode of file registers, refer to Subsection 4.8.2.**

9.7 FNC 16 – FMOV / Fill Move

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction transfers same data to specified number of devices.

1. Instruction format

D	FNC 16 FMOV	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	FMOV	Continuous Operation
	FMOVP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DFMOV	Continuous Operation
	DFMOVP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(S·)	Transfer source data or device number storing data	16- or 32-bit binary
(D·)	Head word device number of transfer destination (Same data is transferred from the transfer source at one time.)	16- or 32-bit binary
n	Number of transfer points [K1 ≤ n ≤ K512, H1 ≤ n ≤ H1FF]	16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
(D·)									✓	✓	✓	✓	✓	✓	✓	✓			✓					
n																				✓	✓			

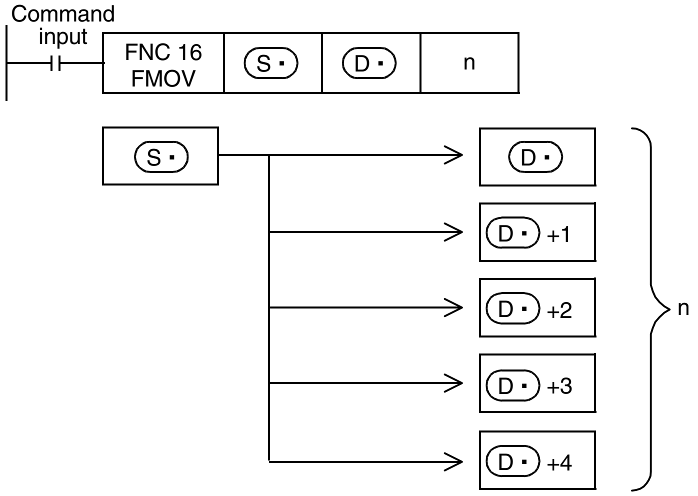
Explanation of function and operation

1. 16-bit operation (FMOV and FMOVP)

The contents of (S·) are transferred to "n" devices starting from (D·).

- The contents will be same among all of "n" devices.
- If the number of points specified by "n" exceeds the device number range, data is transferred within the possible range.
- While the command input is OFF, the transfer destination (D·) does not change.
- While the command input is ON, the data of the transfer source (S·) does not change.

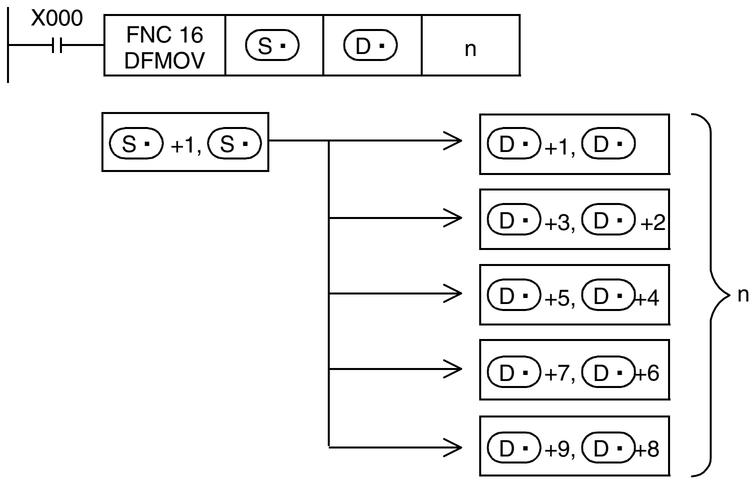
- When a constant (K) is specified as the transfer source (S·), it is automatically converted into binary.



2. 32-bit operation (DFMOV and DFMOVP)

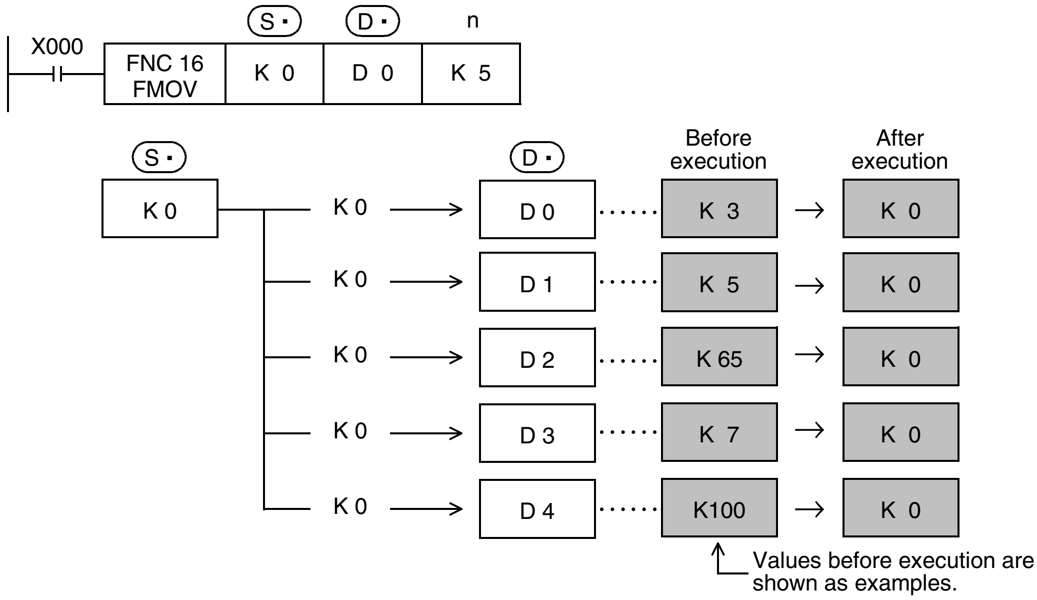
The contents of [(S·)+1, (S·)] are transferred to "n" 32-bit devices starting from [(D·)+1, (D·)].

- The contents will be the same among all of "n" 32-bit devices.
- If the number of points specified by "n" exceeds the device number range, data is transferred within the possible range.
- While the command input is OFF, the transfer destination [(D·)+1, (D·)] does not change.
- While the command input is ON, the data of the transfer source [(S·)+1, (S·)] does not change.
- When a constant (K) is specified as the transfer source [(S·)+1, (S·)], it is automatically converted into binary.



Program example

1. When writing specified data to two or more devices



9.8 FNC 17 – XCH / Exchange

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction exchanges data between two devices.

1. Instruction format

D FNC 17 XCH P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	XCH	Continuous Operation
	XCHP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DXCH	Continuous Operation
	DXCHP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
D1•	Device number storing data to be exchanged.	16- or 32-bit binary
D2•	Device number storing data to be exchanged.	16- or 32-bit binary

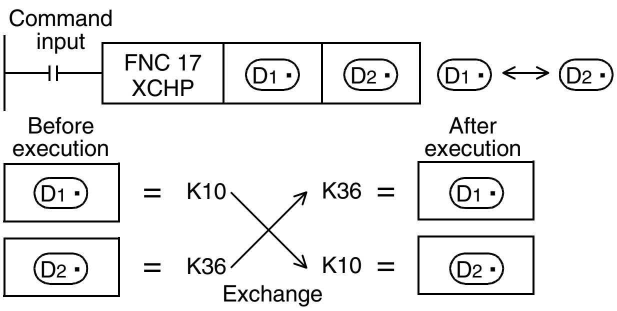
3. Applicable devices

Operand Type	Bit Devices: System User							Word Devices: Digit Specification				Word Devices: System User				Special Unit	Index			Others: Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
D1•									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
D2•									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

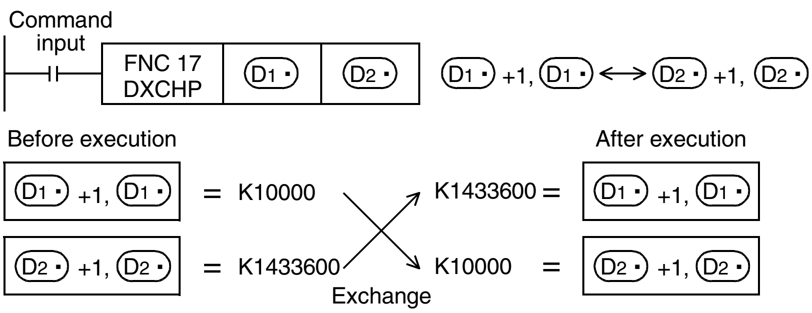
1. 16-bit operation (XCH and XCHP)

Data is exchanged between D1• and D2•.



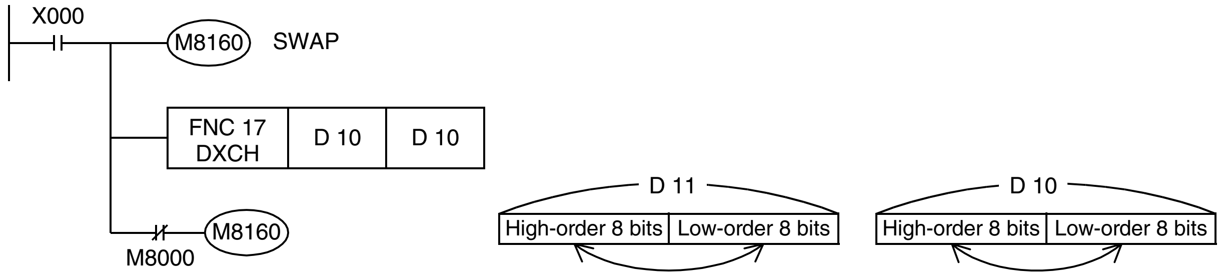
2. 32-bit operation (DXCH and DXCHP)

Data is exchanged between [D1• +1, D1•] and [D2• +1, D2•].



Extension function (function compatible between the FX2 Series and the FX2C Series)

When the instruction is executed while M8160 is ON, high-order 8 bits (byte) and low-order 8 bits (byte) of a word device are exchanged each other.
Because this instruction works in the same way as SWAP (FNC147) instruction, use SWAP instruction when programming new exchange.
In a 32-bit operation, high-order 8 bits (byte) and low-order 8 bits (byte) of each word device are exchanged for each other.



Error

An operation error occurs in the following case. The error flag M8067 turns ON, and the error code is stored in D8067.

- When M8160 is ON, and the device number is different between (D1·) and (D2·)

9.9 FNC 18 – BCD / Conversion to Binary Coded Decimal

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction converts binary (BIN) data into binary-coded decimal (BCD) data.
Binary data is used in operations in PLCs. Use this instruction to display numeric values on the seven-segment display unit equipped with BCD decoder.

1. Instruction format

D	FNC 18 BCD	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	BCD	Continuous Operation
	BCDP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DBCD	Continuous Operation
	DBCDP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Word device number storing the conversion source (binary) data	16- or 32-bit binary
D·	Word device number of the conversion destination (binary-coded decimal) data	16- or 32-bit binary

3. Applicable devices

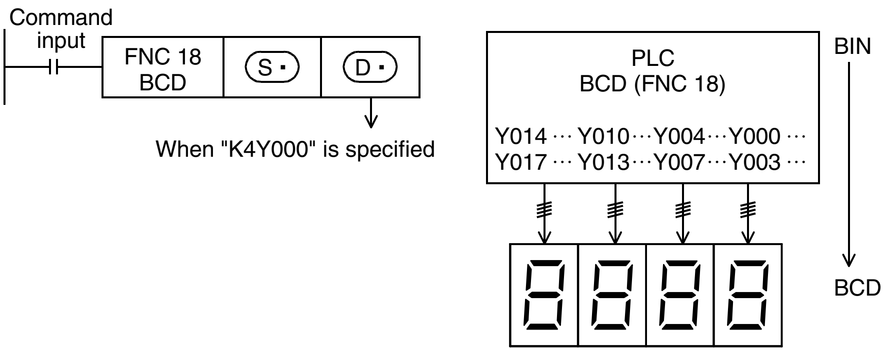
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
D·									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (BCD and BCDP)

This instruction converts the binary (BIN) data of S· into binary-coded decimal (BCD) data, and transfers the converted BCD data to D·.

- The data of S· can be converted if it is within the range from K0 to K9999 (BCD).
- The table below shows digit specification for S· and D·.

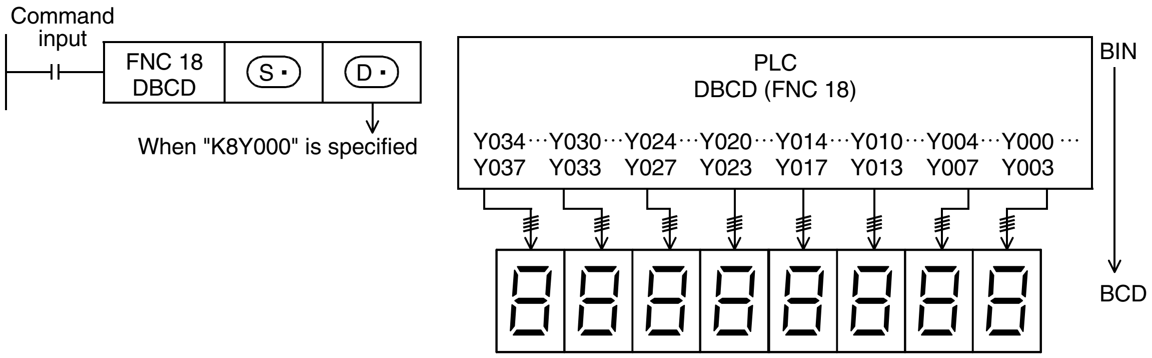


D·	Number of digits	Data range
K1Y000	1	0 to 9
K2Y000	2	00 to 99
K3Y000	3	000 to 999
K4Y000	4	0000 to 9999

2. 32-bit operation (DBCD and DBCDP)

This instruction converts the binary (BIN) data of [S·+1, S·] into binary-coded decimal (BCD) data, and transfers the converted BCD data to [D·+1, D·].

- The data of [S·+1, S·] can be converted if it is within the range from K0 to K99999999 (BCD).
- The table below shows digit specification for [S·+1, S·] and [D·+1, D·].



[D·+1, D·]	Number of digits	Data range
K1Y000	1	0 to 9
K2Y000	2	00 to 99
K3Y000	3	000 to 999
K4Y000	4	0000 to 9999
K5Y000	5	00000 to 99999
K6Y000	6	000000 to 999999
K7Y000	7	0000000 to 9999999
K8Y000	8	00000000 to 99999999

Related instruction

Instruction	Function
BIN (FNC 19)	Converts binary-coded decimal (BCD) data into binary (BIN) data.

Cautions

1. When using SEGL (FNC 74) or ARWS (FNC 75) instruction

Because conversion between binary-coded decimal data and binary data is automatically executed in SEGL (FNC 74) and ARWS (FNC 75) instructions, BCD instruction is not required.

2. Handling of BCD inputs and outputs

Binary data is used in all operations in PLCs including arithmetic operations (+, −, × and ÷), increment and decrement instructions.

- When receiving the digital switch information in the binary-coded decimal (BCD) format into a PLC, use BIN (FNC 19) instruction for converting BCD data into binary data.
- When outputting data to the seven-segment display unit handling binary-coded decimal (BCD) data, use BCD (FNC 18) instruction for converting binary data into BCD data.

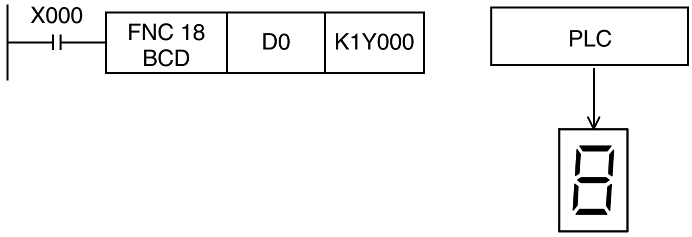
Errors

In BCD or BCDP (16-bit type) instructions, an operation error occurs when the S· value is outside the range from 0 to 9999.

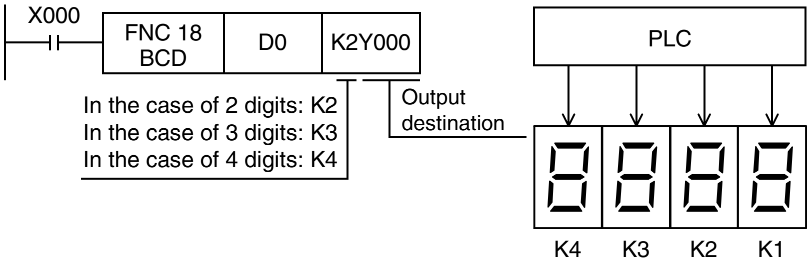
In DBCD or DBCDP (32-bit type) instructions, an operation error occurs when the S· value is outside the range from 0 to 99,999,999.

Program examples

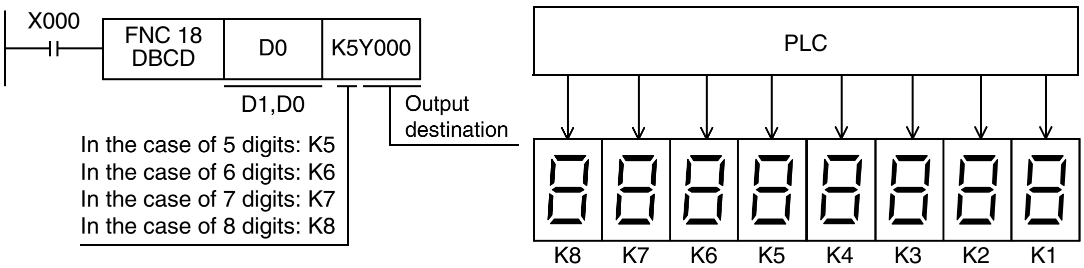
1. When the seven-segment display unit has 1 digit



2. When the seven-segment display unit has 2 to 4 digits



3. When the seven-segment display unit has 5 to 8 digits



9.10 FNC 19 – BIN Conversion to Binary

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction converts binary-coded decimal (BCD) data into binary (BIN) data.
Use this instruction to convert a binary-coded decimal (BCD) value such as a value set by a digital switch into binary (BIN) data and to receive the converted binary data so that the data can be handled in operations in PLCs.

1. Instruction format

D	FNC 19 BIN	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	BIN	Continuous Operation
	BINP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DBIN	Continuous Operation
	DBINP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Word device number storing the conversion source (binary-coded decimal) data	16- or 32-bit binary
D·	Word device number of the conversion destination (binary)	16- or 32-bit binary

3. Applicable devices

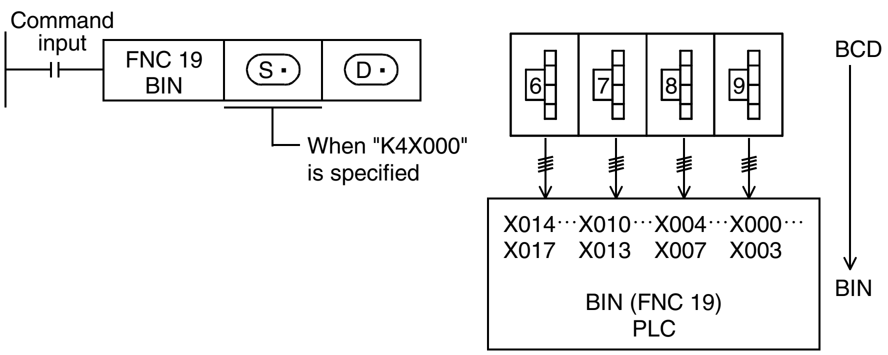
Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
D·									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (BIN and BINP)

This instruction converts the binary-coded decimal (BCD) data of S· into binary (BIN) data, and transfers the converted binary data to D·.

- The data of S· can be converted if it is within the range from K0 to K9999 (BCD).
- The table below shows digit specification for S· and D·.

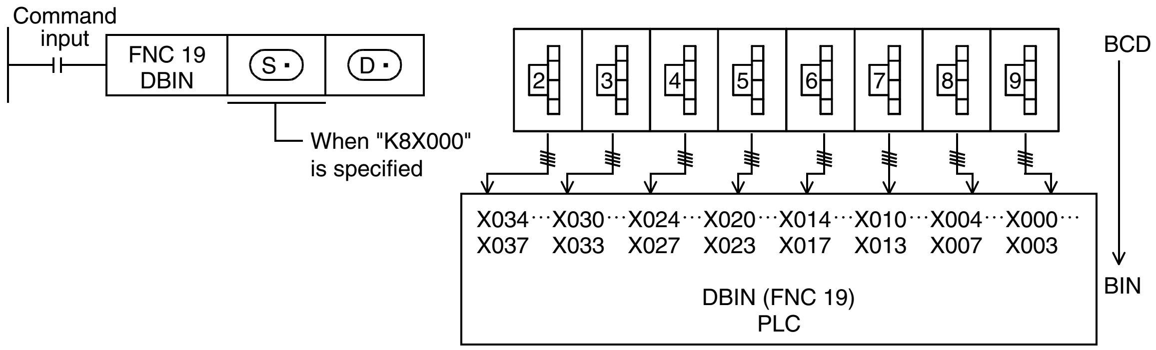


S·	Number of digits	Data range
K1X000	1	0 to 9
K2X000	2	00 to 99
K3X000	3	000 to 999
K4X000	4	0000 to 9999

2. 32-bit operation (DBIN and DBINP)

This instruction converts the binary-coded decimal (BCD) data of [S·+1, S·] into binary (BIN) data, and transfers the converted binary data to [D·+1, D·].

- The data of [S·+1, S·] can be converted if it is within the range from 0 to 99,999,999 (BCD).
- The table below shows digit specification for [S·+1, S·] and [D·+1, D·].



[S·+1, S·]	Number of digits	Data range
K1X000	1	0 to 9
K2X000	2	00 to 99
K3X000	3	000 to 999
K4X000	4	0000 to 9999
K5X000	5	00000 to 99999
K6X000	6	000000 to 999999
K7X000	7	0000000 to 9999999
K8X000	8	00000000 to 99999999

Related instruction

Instruction	Function
BCD (FNC 18)	Converts binary (BIN) data into binary-coded decimal (BCD) data.

Cautions

1. When using DSW (FNC 72) instruction

Because conversion between binary-coded decimal data and binary data is automatically executed in DSW (FNC 72) instruction, BIN instruction is not required.

2. Handling of BCD inputs and outputs

Binary data is used in all operations in PLCs including arithmetic operations (+, −, × and ÷), increment and decrement instructions.

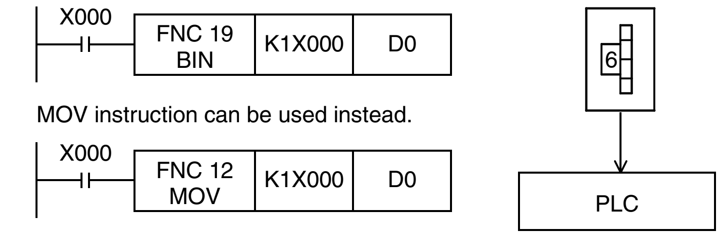
- When receiving the digital switch information in the binary-coded decimal (BCD) format into a PLC, use BIN (FNC 19) instruction for converting BCD data into binary data.
- When outputting data to the seven-segment display unit handling binary-coded decimal (BCD) data, use BCD (FNC 18) instruction for converting binary data into BCD data.

Error

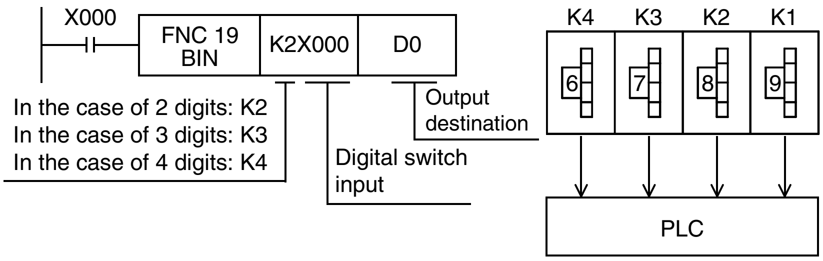
M8067 (operation error) turns ON when the source data is not binary-coded decimal (BCD).

Program examples

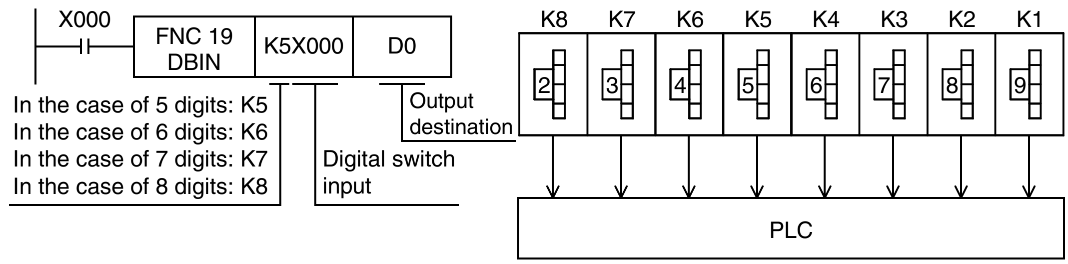
1. When the digital switch has 1 digit



2. When the digital switch has 2 to 4 digits



3. When the digital switch has 5 to 8 digits



10. Arithmetic and Logical Operation (+, −, ×, ÷) – FNC 20 to FNC 29

FNC 20 to FNC 29 provide instructions for arithmetic operations and logical operations of numeric data.

FNC No.	Mnemonic	Symbol	Function	Reference
20	ADD	ADD S1 S2 D	Addition	Section 10.1
21	SUB	SUB S1 S2 D	Subtraction	Section 10.2
22	MUL	MUL S1 S2 D	Multiplication	Section 10.3
23	DIV	DIV S1 S2 D	Division	Section 10.4
24	INC	INC D	Increment	Section 10.5
25	DEC	DEC D	Decrement	Section 10.6
26	WAND	WAND S1 S2 D	Logical Word AND	Section 10.7
27	WOR	WOR S1 S2 D	Logical Word OR	Section 10.8
28	WXOR	WXOR S1 S2 D	Logical Exclusive OR	Section 10.9
29	NEG	NEG D	Negation	Section 10.10

Floating point operation instructions
FX3U and FX3UC PLCs offer not only arithmetic operation instructions in the binary format but also arithmetic operation instructions in the floating point format.

FNC No.	Instruction mnemonic	Contents of processing
120	[D]EADD	Addition of binary floating point
121	[D]ESUB	Subtraction of binary floating point
122	[D]EMUL	Multiplication of binary floating point
123	[D]EDIV	Division of binary floating point

For details, refer to the explanation of each instruction.

→ **For the floating point operation, refer to Chapter 18.**

10.1 FNC 20 – ADD / Addition

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes addition by two values to obtain the result (A + B = C).

→ **For the floating point addition instruction EADD (FNC120), refer to Section 18.8.**

1. Instruction format

D	FNC 20 ADD	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	ADD	Continuous Operation
	ADDP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DADD	Continuous Operation
	DADDP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S1•	Data for addition or word device number storing data	16- or 32-bit binary
S2•	Data for addition or word device number storing data	16- or 32-bit binary
D•	Word device number storing the addition result	16- or 32-bit binary

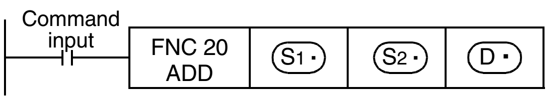
3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D•									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (ADD and ADDP)

The contents of S2• are added to S1• in the binary format, and the addition result is transferred to D•.

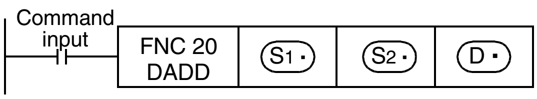


S1• + S2• → D•

- The most significant bit of each data indicates the sign (positive: 0 or negative: 1), and data are added algebraically.
  5 + (–8) = –3
- When a constant (K) is specified in S1• or S2•, it is automatically converted into the binary format.

2. 32-bit operation (DADD and DADDP)

The contents of [S2• +1, S2•] are added to [S1• +1, S1•] in the binary format, and the addition result is transferred to [D1• +1, D1•].



[S1• +1, S1•] + [S2• +1, S2•] → [D• +1, D•]

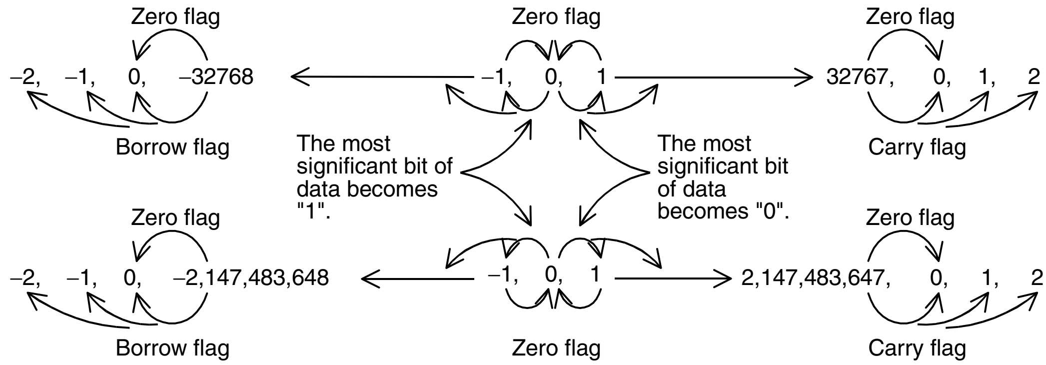
- The most significant bit of each data indicates the sign (positive: 0 or negative: 1), and data are added algebraically.
  5500 + (–8540) = –3040
- When a constant (K) is specified in [S1• +1, S1•] or [S2• +1, S2•], it is automatically converted into the binary format.

Related devices

1. Relationship between the flag operation and the sign (positive or negative) of a numeric value

→ For the flag operations, refer to Subsection 6.5.2.

Device	Name	Description
M8020	Zero	ON : When the operation result is 0 OFF: When the operation result is not 0
M8021	Borrow	ON : When the operation result is less than −32768 (in 16-bit operation) or −2,147,483,648 (in 32-bit operation) OFF: When the operation result is not less than −32768 (in 16-bit operation) or −2,147,483,648 (in 32-bit operation)
M8022	Carry	ON : When the operation result is more than 32767 (in 16-bit operation) or 2,147,483,647 (in 32-bit operation) OFF: When the operation result is not more than 32767 (in 16-bit operation) or 2,147,483,647 (in 32-bit operation)



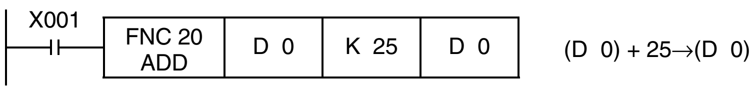
Cautions

1. When using a 32-bit operation instruction (DADD or DADDP)

When specifying word devices, a 16-bit word device on the low-order side is specified first, and a word device with the subsequent device number is automatically set for the high-order 16 bits.
To prevent number overlap, it is recommended to always specify an even number, for example.

2. When specifying the same device in the source and destination

The same device number can be specified for both the source and the destination.
In this case, note that the addition result changes in every operation cycle if a continuous operation type instruction (ADD or DADD) is used.

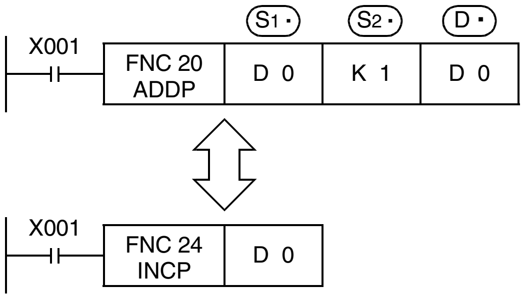


Program example

1. Difference between ADD instruction and INC instruction caused by a program for adding "+1"

When ADD[P] is executed, "1" is added to the contents of D0 every time X001 turns ON from OFF.
ADD[P] instruction is similar to INCP instruction described later except the contents shown in the table below:

			ADD, ADDP, DADD or DADDP instruction	INC, INCP, DINC, DINCP instruction
Flag (zero, borrow or carry)			Operates	Does not operate
Operation result	16-bit operation	(S·)+(+1)=(D·)	+32767→0→+1→+2→	+32767→–32768→–32767
		(S·)+(–1)=(D·)	←–2←–1←0←–32768	—
	32-bit operation	(S·)+(+1)=(D·)	+2,147,483,647→0→+1→+2→	+2,147,483,647→–2,147,483,648→–2,147,483,7
		(S·)+(–1)=(D·)	←–2←–1←0←–2,147,483,648	—



(D 0) + 1→(D 0)

(D 0) + 1→(D 0)

10.2 FNC 21 – SUB / Subtraction

FX3UC Ver.1.00 ⇒ FX3U Ver.2.20 ⇒

Outline

This instruction executes subtraction using two values to obtain the result (A – B = C).

→ For the floating point subtraction instruction ESUB (FNC121), refer to Section 18.9.

1. Instruction format

D	FNC 21 SUB	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	SUB	Continuous Operation
	SUBP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DSUB	Continuous Operation
	DSUBP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S1•	Data for subtraction or word device number storing data	16- or 32-bit binary
S2•	Data for subtraction or word device number storing data	16- or 32-bit binary
D•	Word device number storing the subtraction result	16- or 32-bit binary

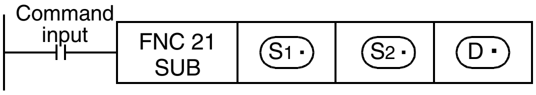
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D•									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (SUB and SUBP)

The contents of S2• are subtracted from S1• in the binary format, and the subtraction result is transferred to D•.

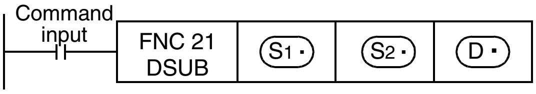


S1• – S2• → D•

- The most significant bit of each data indicates the sign (positive: 0 or negative: 1), and data are subtracted algebraically.
  5 – (–8) = 13
- When a constant (K) is specified in S1• or S2•, it is automatically converted into the binary format.

2. 32-bit operation (DSUB and DSUBP)

The contents of [S2•+1, S2•] are subtracted from [S1•+1, S1•] in the binary format, and the subtraction result is transferred to [D1•+1, D1•].



[S1•+1, S1•] – [S2•+1, S2•] → [D• +1, D•]

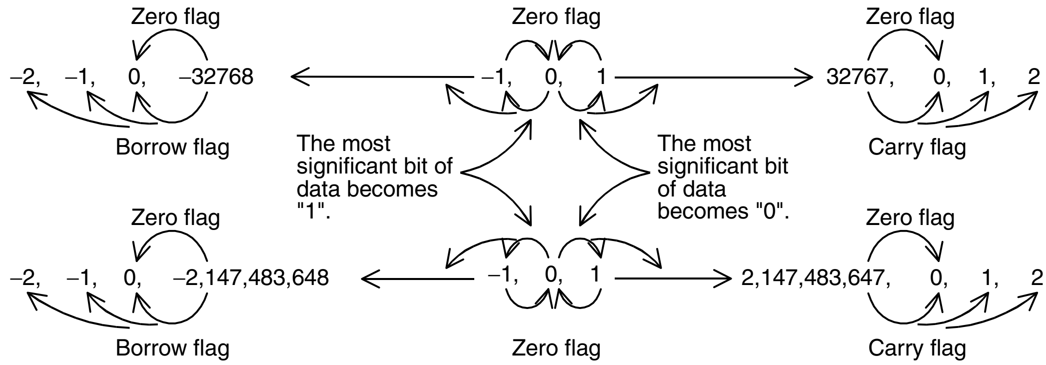
- The most significant bit of each data indicates the sign (positive: 0 or negative: 1), and data are subtracted algebraically.
  5500 – (–8540) = 14040
- When a constant (K) is specified in [S1•+1, S1•] or [S2•+1, S2•], it is automatically converted into the binary format.

Related devices

1. Relationship between the flag operation and the sign (positive or negative) of a numeric value

→ For the flag operations, refer to Subsection 6.5.2.

Device	Name	Description
M8020	Zero	ON : When the operation result is 0 OFF: When the operation result is other than 0
M8021	Borrow	ON : When the operation result is less than –32768 (in 16-bit operation) or –2,147,483,648 (in 32-bit operation) OFF: When the operation result is not less than –32768 (in 16-bit operation) or –2,147,483,648 (in 32-bit operation)
M8022	Carry	ON : When the operation result is more than 32767 (in 16-bit operation) or 2,147,483,647 (in 32-bit operation) OFF: When the operation result is not more than 32767 (in 16-bit operation) or 2,147,483,647 (in 32-bit operation)



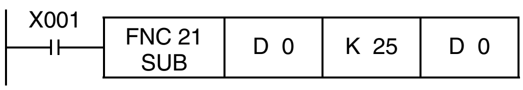
Cautions

1. When using a 32-bit operation instruction (DSUB or DSUBP)

When specifying word devices, a 16-bit word device on the low-order side is specified first, and then a word device with the subsequent device number is automatically set for the high-order 16 bits.
For preventing number overlap, it is recommended to always specify an even number, for example.

2. When specifying the same device in the source and destination

The same device number can be specified for both the source and the destination.
In this case, note that the addition result changes in every operation cycle if a continuous operation type instruction (SUB or DSUB) is used.



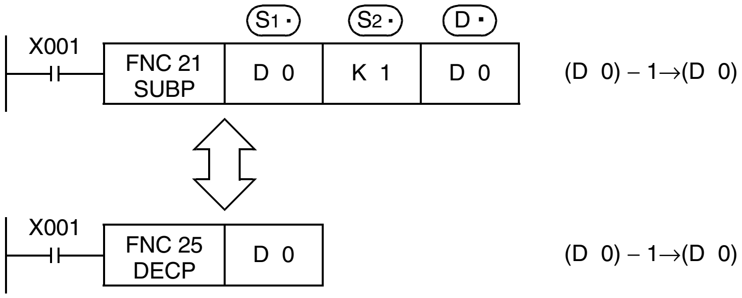
(D 0) – 25→(D 0)

Program example

1. Difference between SUB instruction and DEC instruction caused by a program for subtracting "1"

"1" is subtracted from the contents of D0 every time X001 turns ON from OFF.
SUB[P] instruction is similar to DECP instruction described later except the contents shown in the table below:

			[D] SUB instruction	[D] DEC instruction
Flag (zero, borrow or carry)			Operates	Does not operate
Operation result	16-bit operation	S· –(+1)= D·	←–2←–1←0←–32768	—
		S· –(–1)= D·	+32767→0→+1→+2→	+32767→–32768→–32767
	32-bit operation	S· –(+1)= D·	←–2←–1←0←–2,147,483,648	—
		S· –(–1)= D·	+2,147,483,647→0→+1→+2→	+2,147,483,647→–2,147,483,648→–2,147,483,7



10.3 FNC 22 – MUL / Multiplication

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes multiplication by two values to obtain the result (A x B = C).

→ For the floating point multiplication instruction EMUL (FNC122), refer to Section 18.10.

1. Instruction format

FNC 22	D MUL P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	MUL	Continuous Operation
	MULP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DMUL	Continuous Operation
	DMULP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S1•	Data for multiplication or word device number storing data	16- or 32-bit binary
S2•	Data for multiplication or word device number storing data	16- or 32-bit binary
D•	Head word device number storing the multiplication result	32- or 64-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
D•									✓	✓	✓	✓	✓	✓	✓	✓		▲	✓					

▲ : Available only in 16-bit operations (Not available in 32-bit operations)

Explanation of function and operation

1. 16-bit operation (MUL and MULP)

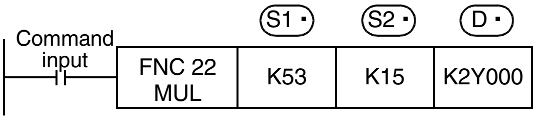
The contents of S1• are multiplied by S2• in the binary format, and the multiplication result is transferred to 32-bit [D• +1, D1•] .

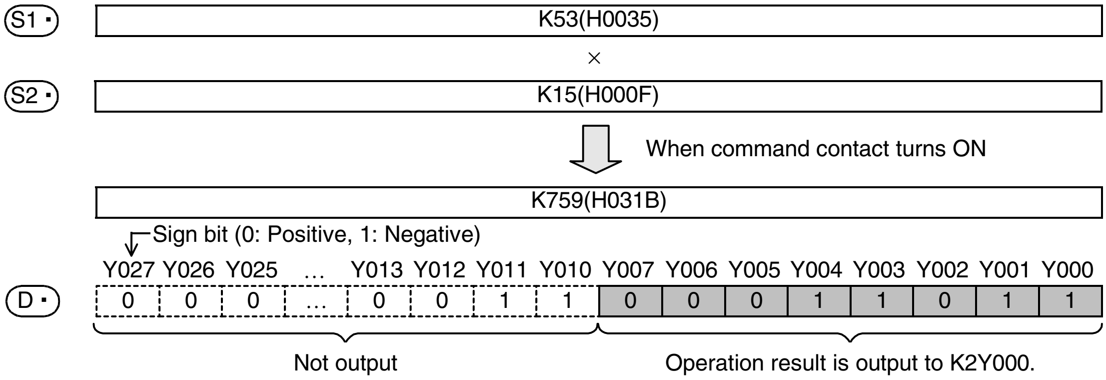
Command input ── FNC 22 MUL | S1• | S2• | D•

BIN (S1•) × BIN (S2•) → BIN (D• +1, D•)
16 bits × 16 bits → 32 bits

- The most significant bit of each data indicates the sign (positive: 0 or negative: 1), and data are multiplied algebraically.
  5 × (−8) = −40
- When a constant (K) is specified in S1• or S2• , it is automatically converted into the binary format.

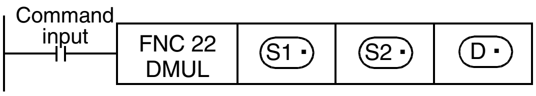
- When a digit (K1 to K8) is specified for [D· +1, D·]
  A digit can be specified in the range from K1 to K8.
  For example, when K2 is specified, only low-order 8 bits can be obtained out of the product (32 bits).





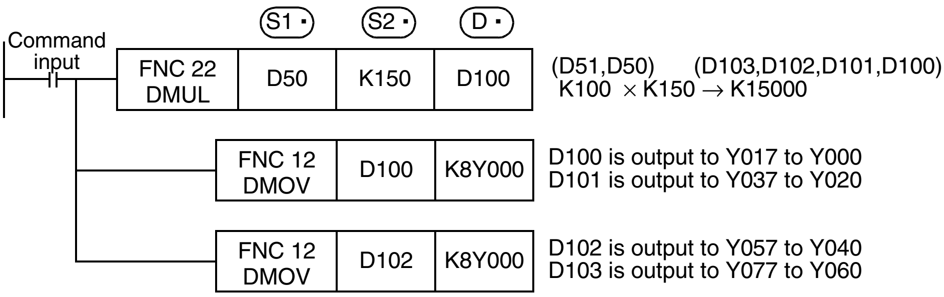
2. 32-bit operation (DMUL and DMULP)

The contents of [S1· +1, S1·] are multiplied by [S2· +1, S2·] in the binary format, and the multiplication result is transferred to 64-bit [D· +3, D· +2, D· +1, D·] (four word devices).



BIN [S1· +1, S1·] (32 bits) × BIN [S2· +1, S2·] (32 bits) → BIN [D· +3, D· +2, D· +1, D·] (64 bits)

- The most significant bit of each data indicates the sign (positive: 0 or negative: 1), and data are multiplied algebraically.
  5500 × (–8540) = –46,970,000
- When a constant (K) is specified in [S1· +1, S1·] or [S2· +1, S2·], it is automatically converted into the binary format.
- When a digit (K1 to K8) is specified for [D· +3, D· +2, D· +1, D·]
  The result is obtained only for low-order 32 bits, and is not obtained for high-order 32 bits.
  Transfer the data to word devices once, then execute the operation.



Related devices

1. Relationship between flag operation and numeric value

Devicee	Name	Description
M8304*1	Zero	ON: When the operation result is 0. OFF: When the operation result is a number other than 0.

*1. Available in Ver. 2.30

Cautions

1. Devices specified in (D·)

- In a 32-bit operation (by DMUL or DMULP), Z cannot be specified in (D·).

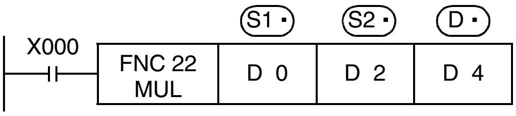
2. When monitoring the operation result in a programming tool

Even if word devices are used, the operation result (64 bits) cannot be monitored at one time.
In such a case, floating point operation is recommended.

→ **For the floating point operation, refer to Chapter 18.**

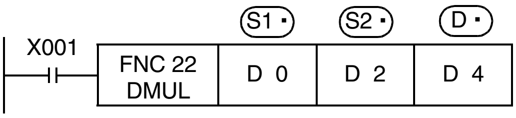
Program examples

1. 16-bit operation



(D 0) × (D 2) → (D 5,D 4)
8 9 72

2. 32-bit operation



(D 1,D 0) × (D 3,D 2) → (D 7,D 6,D 5,D 4)
1756 327 574,212

Function Changes According to Versions

Compatible Versions		Item	Function Summary
FX3U	FX3UC		
Ver. 2.30 or later	Ver. 2.30 or later	Zero Flag	Turns the special device M8304 ON when the operation result of MUL command is 0.

10.4 FNC 23 – DIV / Division

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes division by two values to obtain the result (A ÷ B = C ...).

→ **For the floating point division instruction EDIV (FNC123), refer to Section 18.11.**

1. Instruction format

	FNC 23	
D	DIV	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	DIV	Continuous Operation
	DIVP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DDIV	Continuous Operation
	DDIVP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S1•	Data for division or word device number storing data (dividend)	16- or 32-bit binary
S2•	Data for division or word device number storing data (divisor)	16- or 32-bit binary
D•	Head word device number storing the division result (quotient and remainder)	32- or 64-bit binary

3. Applicable devices

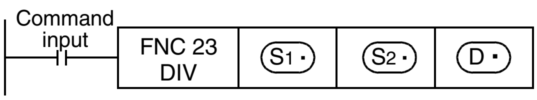
Operand Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index				Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
D•									✓	✓	✓	✓	✓	✓	✓	✓		▲	✓					

▲ : Available only in 16-bit operations (Not available in 32-bit operations)

Explanation of function and operation

1. 16-bit operation (DIV and DIVP)

S1• indicates the dividend, S2• indicates the divisor, the quotient is transferred to D•, and the remainder is transferred to D•+1.

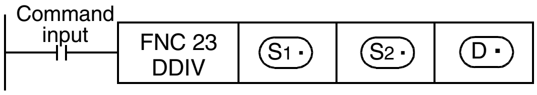


Dividend	Divisor	Quotient	Remainder
BIN	BIN	BIN	BIN
(S1•) ÷	(S2•) →	(D•) ...	(D•+1)
16 bits	16 bits	16 bits	16 bits

- The most significant bit of each data indicates the sign (positive: 0 or negative: 1), and data are divided algebraically.
  36 ÷ (–5) = –7 (quotient) ... 1 (remainder)
- Two devices in total starting from D• are occupied to store the operation result (quotient and remainder). Make sure that these two devices are not used for another control.
- When a constant (K) is specified as S1• or S2•, it is automatically converted into the binary format.

2. 32-bit operation (DDIV and DDIVP)

[S1·+1, S1·] indicates the dividend, [S2·+1, S2·] indicates the divisor, the quotient is transferred to [D·+1, D·]), and the remainder is transferred to [D·+3, D·+2].



Dividend BIN		Divisor BIN		Quotient BIN		Remainder BIN
[S1·+1, S1·]	÷	[S2·+1, S2·]	→	[D·+1, D·]	...	[D·+3, D·+2]
32 bits		32 bits		32 bits		32 bits

- Four devices in total starting from D· are occupied to store the operation result (quotient and remainder). Make sure that these four devices are not used for another control.
- The most significant bit of each data indicates the sign (positive: 0 or negative: 1), and data are divided algebraically.
  5500 ÷ (–540) = –10 (quotient) ... 100 (remainder)
- When a constant (K) is specified in [S1·+1, S1·] or [S2·+1, S2·], it is automatically converted into the binary format.

Related devices

1. Relationship between flag operation and numeric value

Devicee	Name	Description
M8304*1	Zero	ON : When the operation result is 0. OFF : When the operation result is a number other than 0.
M8306*1	Carry	ON : Carry flag operates when the operation result is over 32,767 (16-bit operation) or2,147,483,647 (32-bit operation). OFF : When the operation result is less than 32,767 (16-bit operation) or 2,147,483,647 (32-bit operation).

*1. Available in Ver. 2.30

Cautions

1. Operation result

- The most significant bit of the quotient and remainder indicates the sign (positive: 0, negative: 1) respectively.
- The quotient is negative when either the dividend or divisor is negative.
  The remainder is negative when the dividend is negative.

2. Device specified as D·

- The remainder is not obtained when a bit device is specified with digit specification.
- In a 32-bit operation (by DSUB or DSUBP), Z cannot be specified as D·.

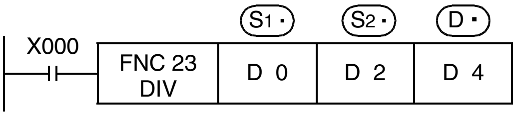
Error

- When the divisor S2· is "0", an operation error is caused and the instruction is not executed.
- A operation error results when the operation result is over 32,767 (16-bit operation) or 2,147,483,647 (32-bit operation). (Turns the carry flag ON.)

1 Introduction
2 Overview
3 Instruction List
4 Devices in Detail
5 Specified the Device & Constant
6 Before Programming
7 Basic Instruction
8 FNC00-FNC09 Program Flow
9 FNC10-FNC19 Move & Compare
10 FNC20-FNC29 Arith. & Logic Operation

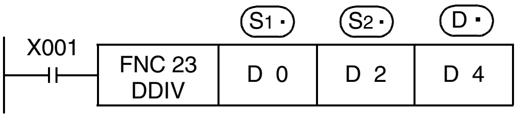
Program examples

1. 16-bit operation



Dividend		Divisor		Quotient		Remainder
(D 0)	÷	(D 2)	→	(D 4)	...	(D 5)
100		33		3		1

2. 32-bit operation



Dividend		Divisor		Quotient		Remainder
(D 1,D 0)	÷	(D 3,D 2)	→	(D 5,D 4)	...	(D 7,D 6)
100,000		3333		30		10

Function Changes According to Versions

Compatible Versions		Item	Function Summary
FX 3U	FX3UC		
Ver. 2.30 or later	Ver. 2.30 or later	Zero Flag	Turns M8304 ON when the operation result of DIV instruction is 0.
		Carry Flag	Turns M8306 ON when the operation result of DIV instruction overflows. 16-bit operation : Only when the maximum negative value(-32,768) is divided by -1. 32-bit operation : Only when the maximum negative value (-2,147,483,648) is divided by -1.

10.5 FNC 24 – INC / Increment

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction increments the data of a specified device by "1".

1. Instruction format

D	FNC 24 INC	P

16-bit Instruction	Mnemonic	Operation Condition
3 steps	INC	Continuous Operation
	INCP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
5 steps	DINC	Continuous Operation
	DINCP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
D·	Word device number storing data to be incremented by "1"	16- or 32-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
D·									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (INC and INCP)

The contents of D· are incremented by "1", and the increment result is transferred to D·.

Command input → FNC 24 INCP D·

D· + 1 → D·

2. 32-bit operation (DINC and DINCP)

The contents of [D·+1, D·] are incremented by "1", and the increment result is transferred to [D·+1, D·].

Command input → FNC 24 DINCP D·

[D·+1, D·] + 1 → [D·+ 1, D·]

Cautions

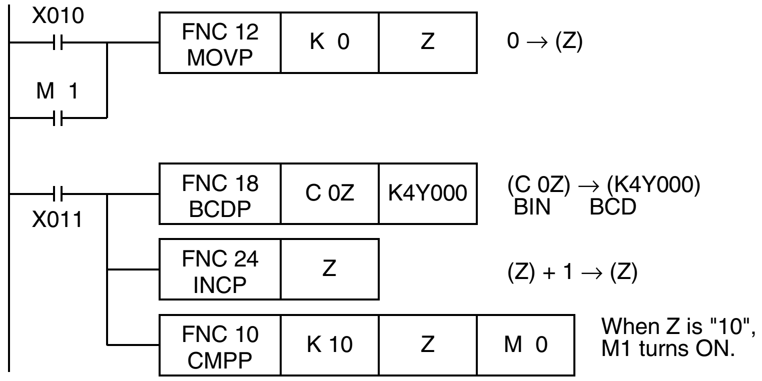
1. Note that data is incremented in every operation cycle in a continuous operation type instruction.

2. Flag operations

1) 16-bit operation
When "+32767" is incremented by "1", the result is "–32768". Flags (zero, carry and borrow) are not activated at this time.

2) 32-bit operation
When "+2,147,483,647" is incremented by "1", the result is "–2,147,483,648". Flags (zero, carry and borrow) are not activated at this time.

Program example



The current values of the counters C0 to C9 are converted into the BCD format, and output to K4Y000.

Z is cleared by the reset input X010.

Every time X011 is set to ON, the current values of C0, C1 ... C9 are output in turn.

10.6 FNC 25 – DEC / Decrement

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction decrements the data of a specified device by "1".

1. Instruction format

D	FNC 25 DEC	P

16-bit Instruction	Mnemonic	Operation Condition
3 steps	DEC	Continuous Operation
	DECP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
5 steps	DDEC	Continuous Operation
	DDECP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
(D·)	Word device number storing data to be decremented by "1"	16- or 32-bit binary

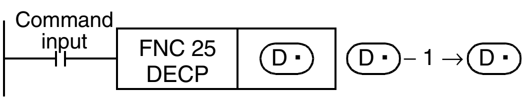
3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(D·)									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

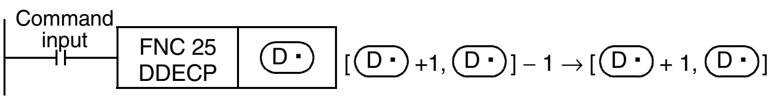
1. 16-bit operation (DEC and DECP)

The contents of (D·) are decremented by "1", and the decremented result is transferred to (D·).



2. 32-bit operation (DDEC and DDECP)

The contents of [(D·)+1, (D·)] are decremented by "1", and the decremented result is transferred to [(D·)+1, (D·)].



Cautions

1. Flag operations

1) 16-bit operation
When "–32768" is decremented by "1", the result is "+32767". Flags (zero, carry and borrow) are not activated at this time.

2) 32-bit operation
When "–2,147,483,648" is decremented by "1", the result is "+2,147,483,647". Flags (zero, carry and borrow) are not activated at this time.

10.7 FNC 26 – WAND / Logical Word AND

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes the logical product (AND) operation of two numeric values.

1. Instruction format

W / D	FNC 26 AND	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	WAND	Continuous Operation
	WANDP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DAND	Continuous Operation
	DANDP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S1•	Data used for logical product or word device number storing data	16- or 32-bit binary
S2•	Data used for logical product or word device number storing data	16- or 32-bit binary
D•	Word device number storing the logical product result	16- or 32-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D•									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (WAND and WANDP)

The logical product (AND) operation is executed to the contents of S1• and S2• in units of bit, and the result is transferred to D•.

Command input

FNC 26 WAND	S1•	S2•	D•

S1• ∧ S2• → D•

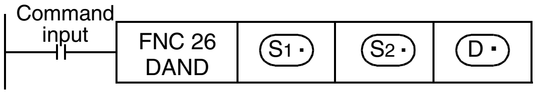
- While the command input is OFF, the data of the transfer destination D• does not change.
- While the command input is ON, the data of the transfer sources S1• and S2• do not change.
- When a constant (K) is specified in the transfer sources S1• and S2•, it is automatically converted into the binary format.
- The logical product operation is executed in units of bit as shown in the table below (1 ∧ 1 = 1, 0 ∧ 1 = 0, 1 ∧ 0 = 0, 0 ∧ 0 = 0).

1: ON, 0: OFF

	S1•	S2•	D• WAND (FNC 26) instruction
Logical operation (unit: bit)	0	0	0
	1	0	0
	0	1	0
	1	1	1

2. 32-bit operation (DAND and DANDP)

The logical product (AND) operation is executed to the contents of [S1·+1, S1·] and [S2·+1, S2·] in units of bit, and the result is transferred to [D·+1, D·].



S1· +1, S1· ∧ S2· +1, S2· → D· +1, D·

- While the command input is OFF, the data of the transfer destination [D·+1, D·] does not change.
- While the command input is ON, the data of the transfer source [S1·+1, S1·] [S2·+1, S2·] do not change.
- When a constant (K) is specified in the transfer source [S1·+1, S1·] [S2·+1, S2·], it is automatically converted into the binary format.
- The logical product operation is executed in units of bit as shown in the table below (1 ∧ 1 = 1, 0 ∧ 1 = 0, 1 ∧ 0 = 0, 0 ∧ 0 = 0).

1: ON, 0: OFF

	S1·+1, S1·	**S2·+1, S2·**	**D·+1, D·** **DAND (FNC 26) instruction**
Logical operation (unit: bit)	0	0	0
	1	0	0
	0	1	0
	1	1	1

10.8 FNC 27 – WOR / Logical Word OR

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction executes the logical sum (OR) operation of two numeric values.

1. Instruction format

W / D	FNC 27 OR	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	WOR	Continuous Operation
	WORP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DOR	Continuous Operation
	DORP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S1•	Data used for logical sum or word device number storing data	16- or 32-bit binary
S2•	Data used for logical sum or word device number storing data	16- or 32-bit binary
D•	Word device number storing the logical sum result	16- or 32-bit binary

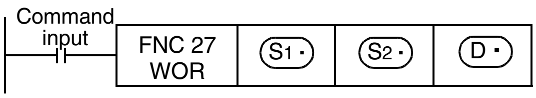
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index				Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D•									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (WOR and WORP)

The logical sum (OR) operation is executed to the contents of S1• and S2• in units of bit, and the result is transferred to D•.



S1• ∨ S2• → D•

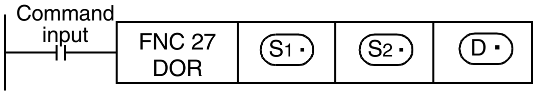
- While the command input is OFF, the data of the transfer destination D• does not change.
- While the command input is ON, the data of the transfer sources S1• and S2• do not change.
- When a constant (K) is specified in the transfer sources S1• and S2•, it is automatically converted into the binary format.
- The logical sum operation is executed in units of bit as shown in the table below (1 ∨ 1 = 1, 0 ∨ 1 = 1, 0 ∨ 0 = 0, 1 ∨ 0 = 1).

1: ON, 0: OFF

	S1•	S2•	D• WOR (FNC 27) instruction
Logical operation (unit: bit)	0	0	0
	1	0	1
	0	1	1
	1	1	1

2. 32-bit operation (DOR and DORP)

The logical sum (OR) operation is executed to the contents of [S1·+1, S1·] and [S2·+1, S2·] in units of bit, and the result is transferred to [D·+1, D·].



S1·+1, S1· ∨ S2·+1, S2· → D·+1, D·

- While the command input is OFF, the data of the transfer destination [D·+1, D·] does not change.
- While the command input is ON, the data of the transfer source [S1·+1, S1·] [S2·+1, S2·] do not change.
- When a constant (K) is specified in the transfer source [S1·+1, S1·] [S2·+1, S2·], it is automatically converted into the binary format.
- The logical sum operation is executed in units of bit as shown in the table below (1 ∨ 1 = 1, 0 ∨ 1 = 1, 0 ∨ 0 = 0, 1 ∨ 0 = 1).

1: ON, 0: OFF

	S1·+1, S1·	**S2·+1, S2·**	**D·+1, D·**
			DOR (FNC 27) instruction
Logical operation (unit: bit)	0	0	0
	1	0	1
	0	1	1
	1	1	1

10.9 FNC 28 – WXOR / Logical Exclusive OR

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes the exclusive logical sum (XOR) operation of two numeric values.

1. Instruction format

W / D	FNC 28 XOR	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	WXOR	Continuous Operation
	WXORP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DXOR	Continuous Operation
	DXORP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S1•	Data used for exclusive logical sum or word device number storing data	16- or 32-bit binary
S2•	Data used for exclusive logical sum or word device number storing data	16- or 32-bit binary
D•	Word device number storing the exclusive logical sum result	16- or 32-bit binary

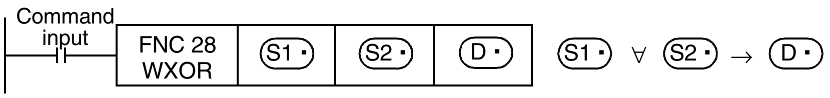
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D•									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (WXOR and WXORP)

The exclusive logical sum (XOR) operation is executed to the contents of S1• and S2• in units of bit, and the result is transferred to D•.



- While the command input is OFF, the data of the transfer destination D• does not change.
- While the command input is ON, the data of the transfer sources S1• and S2• do not change.
- When a constant (K) is specified in the transfer sources S1• and S2•, it is automatically converted into the binary format.
- The logical exclusive sum operation is executed in units of bit as shown in the table below (1 ∀ 1 = 0, 0 ∀ 0 = 0, 1 ∀ 0 = 1, 0 ∀ 1 = 1).

1: ON, 0: OFF

	S1•	S2•	D• WXOR (FNC 28) instruction
Logical operation (unit: bit)	0	0	0
	1	0	1
	0	1	1
	1	1	0

2. 32-bit operation (DXOR and DXORP)

The exclusive logical sum (XOR) operation is executed to the contents of [S1• +1, S1•] and [S2• +1, S2•] in units of bit, and the result is transferred to [D• +1, D•].

Command input

FNC 28 DXOR	S1•	S2•	D•

S1• +1, S1• ∀ S2• +1, S2• → D• +1, D•

- While the command input is OFF, the data of the transfer destination [D• +1, D•] does not change.
- While the command input is ON, the data of the transfer source [S1• +1, S1•] [S2• +1, S2•] do not change.
- When a constant (K) is specified in the transfer source [S1• +1, S1•] [S2• +1, S2•], it is automatically converted into the binary format.
- The exclusive logical sum operation is executed in units of bit as shown in the table below (1 ∀ 1 = 0, 0 ∀ 0 = 0, 1 ∀ 0 = 1, 0 ∀ 1 = 1).

1: ON, 0: OFF

	S1• +1, S1•	**S2• +1, S2•**	**D• +1, D•** DXOR (FNC 28) instruction
Logical operation (unit: bit)	0	0	0
	1	0	1
	0	1	1
	1	1	0

Program example

By combining WXOR and CML (FNC 14) instructions, the exclusive logical sum not (XORNOT) operation can be executed.

X000

FNC 28 WXOR	D 10	D 12	D 14
FNC 14 CML	D 14	D 14	

10.10 FNC 29 – NEG / Negation

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction obtains the complement of a numeric value (by inverting each bit and adding "1").
This instruction can be used to negate the sign of a numeric value.

→ For the Floating point negation ENEG (FNC128), refer to Section 18.16.

1. Instruction format

D	FNC 29 NEG	P

16-bit Instruction	Mnemonic	Operation Condition
3 steps	NEG	Continuous Operation
	NEGP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
5 steps	DNEG	Continuous Operation
	DNEGP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
(D·)	Word device number which stores data for obtaining complement and will store the operation result (The operation result will be stored in the same word device number.)	16- or 32-bit binary

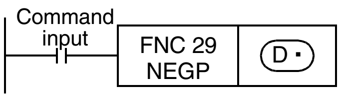
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(D·)									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (NEG and NEGP)

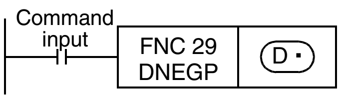
Each bit of (D·) is inverted (0 → 1, 1 → 0), "1" is added, and then the result is stored in the original device.



$\overline{(D\cdot)}+1 \rightarrow (D\cdot)$

2. 32-bit operation (DNEG and DNEGP)

Each bit of [(D·)+1, (D·)] is inverted (0 → 1, 1 → 0), "1" is added, and then the result is stored in the original device.



$\overline{((D\cdot)+1, (D\cdot))}+1 = ((D\cdot)+1, (D\cdot))$

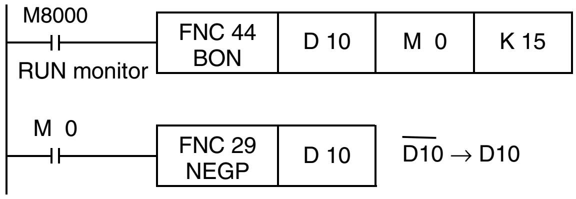
Caution

Note that the complement is obtained in every operation cycle in a continuous operation type instruction.

Program examples

The program examples below are provided to obtain the absolute value of a negative binary value.

1. Obtaining the absolute value of a negative value using NEG instruction

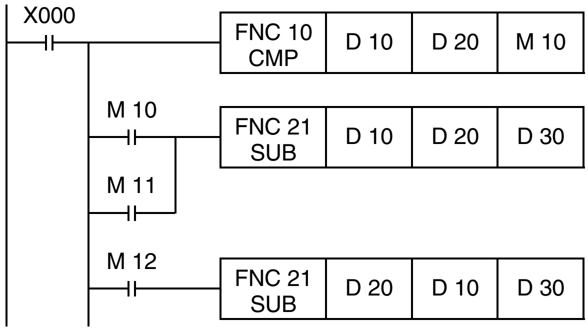


In BON (ON bit check) instruction, M0 turns ON when the bit 15 (b15 among b0 to b15) of D10 is "1".

NEGP instruction is executed for D10 only when M0 turns ON.

2. Obtaining the absolute value by SUB (subtraction) instruction

Even if NEG instruction is not used, D30 always stores the absolute value of the difference.



(D 10) > (D 20) M 10 = ON (D10) = (D 20) M 11 = ON (D 10) < (D 20) M 12 = ON

In the case of "D10 ≥ D20", D10 – D20 → D30.

In the case of "D10 < D20", D20 – D10 → D30.

Negative value expression and absolute value (reference)

In PLCs, a negative value is expressed in 2's complement.
When the most significant bit is "1", it is a negative value, and its absolute value can be obtained by NEG instruction.

(D 10) = 2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

(D 10) = 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

(D 10) = 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

(D 10) = –1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 → $\overline{\text{(D 10)}}$ + 1 = 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

(D 10) = –2
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 → $\overline{\text{(D 10)}}$ + 1 = 2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

⋮

(D 10) = –32767
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 → $\overline{\text{(D 10)}}$ + 1 = 32767
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

(D 10) = –32768
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 → $\overline{\text{(D 10)}}$ + 1 = –32768
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

The absolute value can be obtained up to 32767.

11. Rotation and Shift Operation – FNC 30 to FNC 39

FNC 30 to FNC 39 provide instructions for rotating and shifting bit data and word data in specified directions.

FNC No.	Mnemonic	Symbol	Function	Reference
30	ROR	ROR D n	Rotation Right	Section 11.1
31	ROL	ROL D n	Rotation Left	Section 11.2
32	RCR	RCR D n	Rotation Right with Carry	Section 11.3
33	RCL	RCL D n	Rotation Left with Carry	Section 11.4
34	SFTR	SFTR S D n1 n2	Bit Shift Right	Section 11.5
35	SFTL	SFTL S D n1 n2	Bit Shift Left	Section 11.6
36	WSFR	WSFR S D n1 n2	Word Shift Right	Section 11.7
37	WSFL	WSFL S D n1 n2	Word Shift Left	Section 11.8
38	SFWR	SFWR S D n	Shift write [FIFO/FILO control]	Section 11.9
39	SFRD	SFRD S D n	Shift read [FIFO control]	Section 11.10

11.1 FNC 30 – ROR / Rotation Right

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction shifts and rotates the bit information rightward by the specified number of bits without the carry flag.

1. Instruction format

D	FNC 30 ROR	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	ROR	Continuous Operation
	RORP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DROR	Continuous Operation
	DRORP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
D·	Word device number storing data to be rotated rightward	16- or 32-bit binary
n	Number of bits to be rotated [n ≤ 16 (16-bit instruction), n ≤ 32 (32-bit instruction)]	16- or 32-bit binary

3. Applicable devices

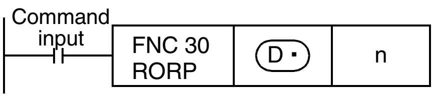
Operand Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
D·									▲	▲	▲	✓	✓	✓	✓	✓	✓	✓	✓					
n														✓	✓					✓	✓			

▲: In 16-bit operations, K4Y○○○, K4M○○○ and K4S○○○ are valid.
In 32-bit operations, K8Y○○○, K8M○○○ and K8S○○○ are valid.

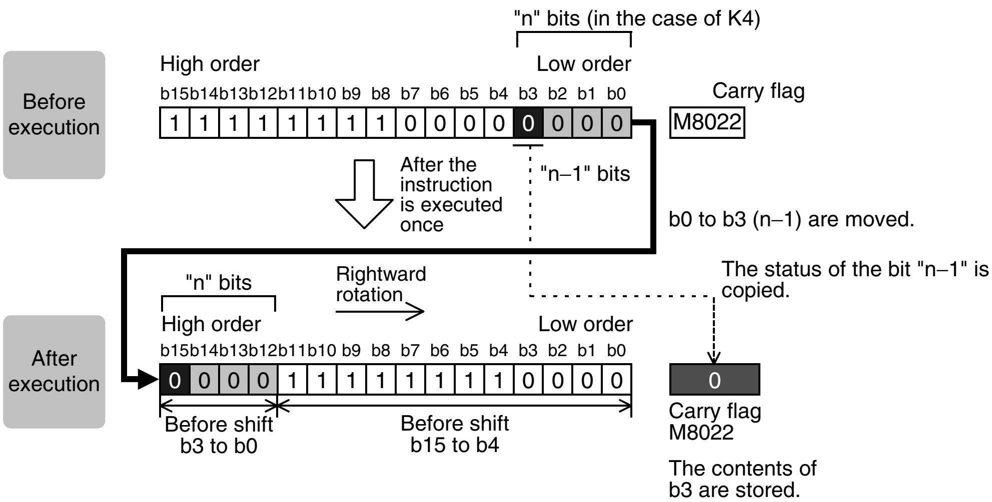
Explanation of function and operation

1. 16-bit operation (ROR and RORP)

"n" bits out of 16 bits of D· are rotated rightward.

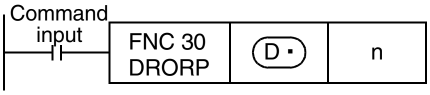


- The final bit is stored in the carry flag (M8022).
- In a device with digit specification, K4 (16-bit instruction) is valid.

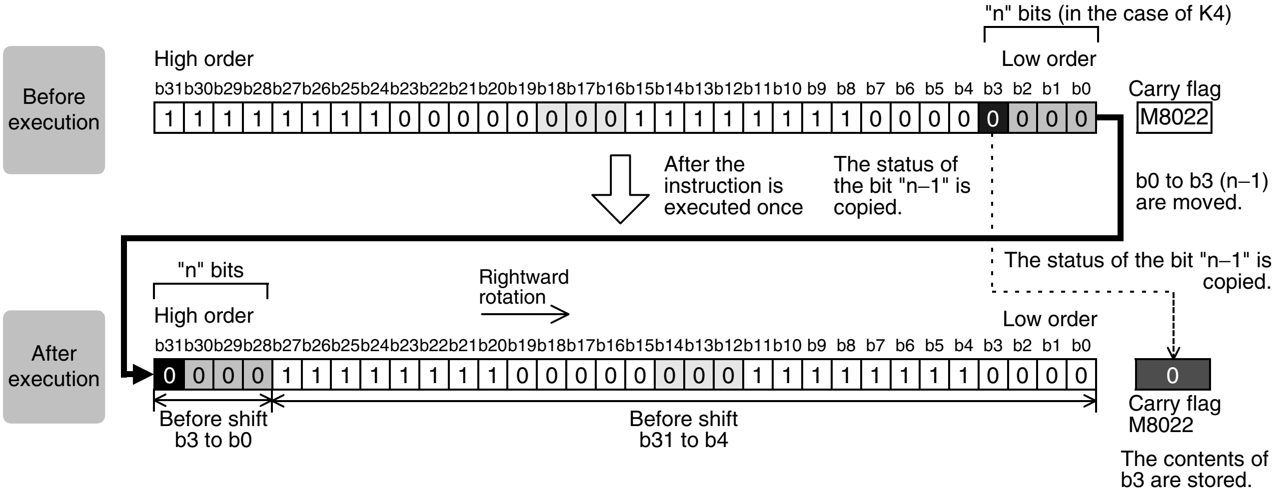


2. 32-bit operation (DROR and DRORP)

"n" bits out of 32 bits of [D· +1, D·] are rotated rightward.



- The final bit is stored in the carry flag (M8022).
- In a device with digit specification, K8 (32-bit instruction) is valid.



Related device

→ **For the carry flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8022	Carry	Turns ON when the bit shifted last from the lowest position is "1".

Cautions

1. In the case of continuous operation type instructions (ROR and DROR)

Note that shift and rotation are executed in every scan time (operation cycle).

2. When a device with digit specification is specified as D·

Only K4 (16-bit instruction) or K8 (32-bit instruction) is valid (examples: K4Y010 or K8M0).

11.2 FNC 31 – ROL / Rotation Left

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction shifts and rotates the bit information leftward by the specified number of bits without the carry flag.

1. Instruction format

D	FNC 31 ROL	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	ROL	Continuous Operation
	ROLP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DROL	Continuous Operation
	DROLP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(D·)	Word device number storing data to be rotated leftward	16- or 32-bit binary
n	Number of bits to be rotated [n ≤ 16 (16-bit instruction), n ≤ 32 (32-bit instruction)]	16- or 32-bit binary

3. Applicable devices

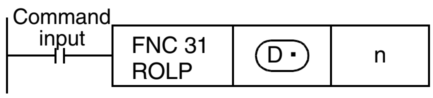
Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(D·)									▲	▲	▲	✓	✓	✓	✓	✓	✓	✓	✓					
n														✓	✓					✓	✓			

▲: In 16-bit operations, K4Y○○○, K4M○○○ and K4S○○○ are valid.
In 32-bit operations, K8Y○○○, K8M○○○ and K8S○○○ are valid.

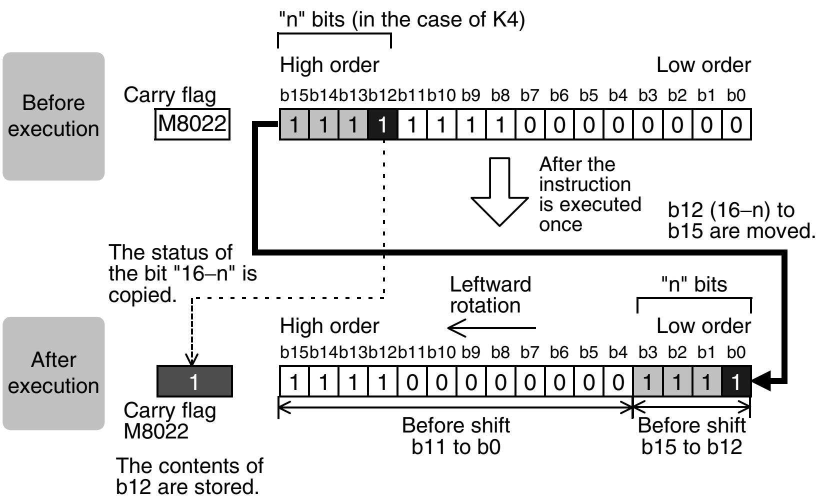
Explanation of function and operation

1. 16-bit operation (ROL and ROLP)

"n" bits out of 16 bits of (D·) are rotated leftward.

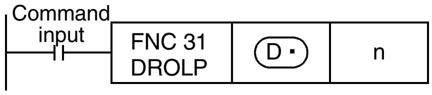


- The final bit is stored in the carry flag (M8022).
- In a device with digit specification, K4 (16-bit instruction) is valid.

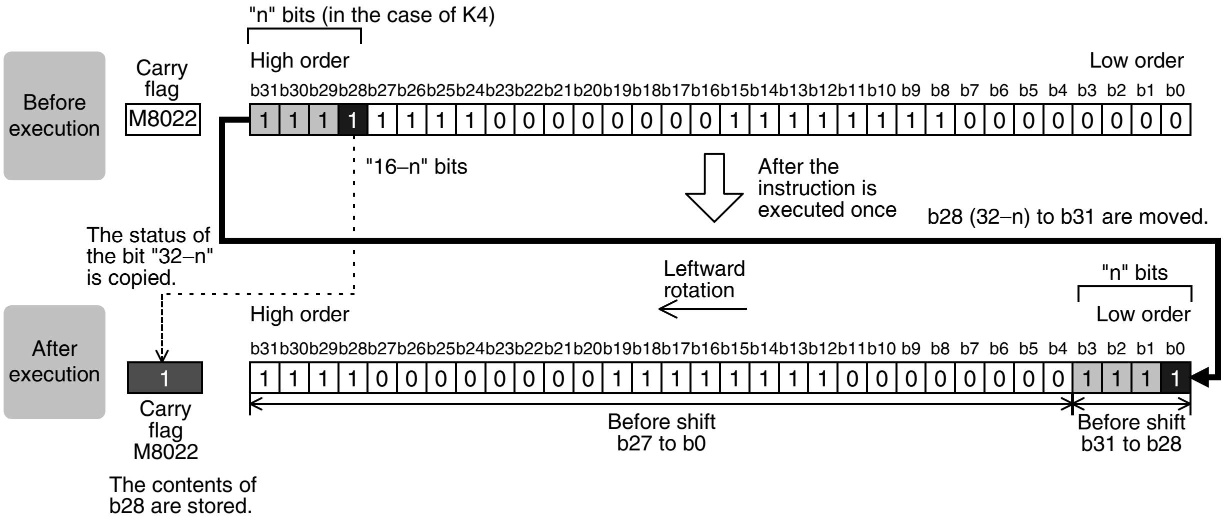


2. 32-bit operation (DROL and DROLP)

"n" bits out of 32 bits of [D· +1, D·] are rotated leftward.



- The final bit is stored in the carry flag (M8022).
- In a device with digit specification, K8 (32-bit instruction) is valid.



Related device

→ **For the carry flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8022	Carry	Turns ON when the bit shifted last from the highest position is "1".

Cautions

1. In the case of continuous operation type instructions (ROL and DROL)

Note that shift and rotation are executed in every scan time (operation cycle).

2. When a device with digit specification is specified as D·

Only K4 (16-bit instruction) or K8 (32-bit instruction) is valid (examples: K4Y010 or K8M0).

11.3 FNC 32 – RCR / Rotation Right with Carry

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction shifts and rotates the bit information rightward by the specified number of bits together with the carry flag.

1. Instruction format

D	FNC 32 RCR	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	RCR	Continuous Operation
	RCRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DRCR	Continuous Operation
	DRCRP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
D·	Word device number storing data to be rotated rightward	16- or 32-bit binary
n	Number of bits to be rotated [n ≤ 16 (16-bit instruction), n ≤ 32 (32-bit instruction)]	16- or 32-bit binary

3. Applicable devices

Operand Type	Bit Devices: System User							Word Devices: Digit Specification				Word Devices: System User				Special Unit	Index			Others: Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
D·									▲	▲	▲	✓	✓	✓	✓	✓	✓	✓	✓					
n														✓	✓					✓	✓			

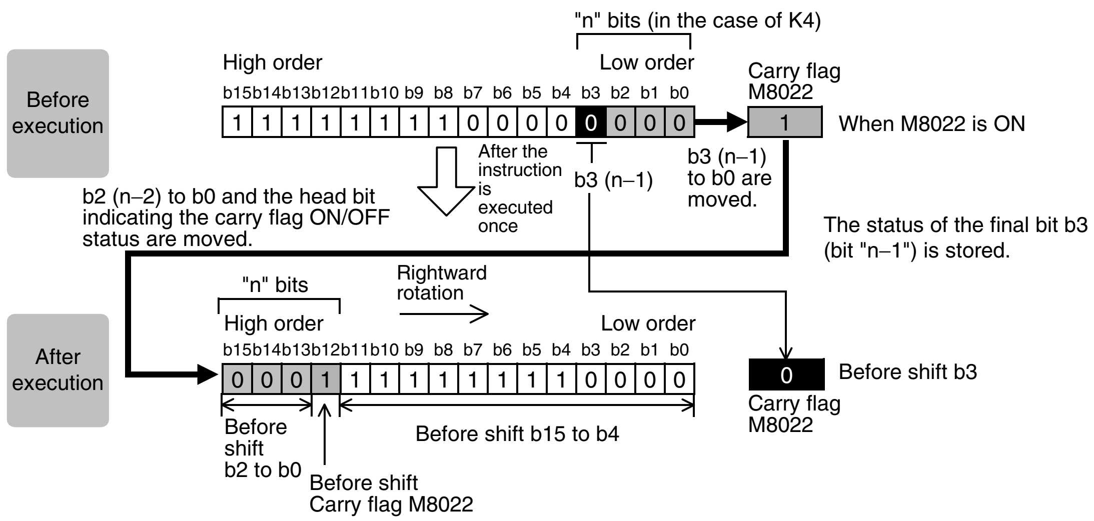
▲: In 16-bit operations, K4Y○○○, K4M○○○ and K4S○○○ are valid.
In 32-bit operations, K8Y○○○, K8M○○○ and K8S○○○ are valid.

Explanation of function and operation

1. 16-bit operation (RCR and RCRP)

"n" bits out of 16 bits of D· and 1 bit (carry flag M8022) are rotated rightward.

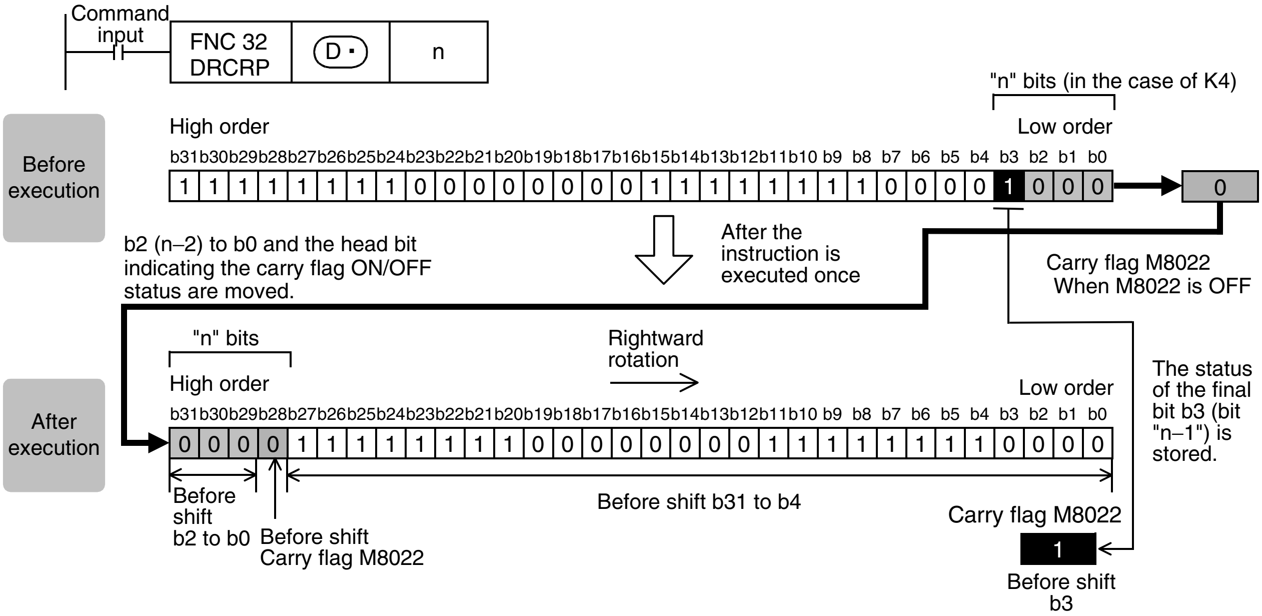
Command input	FNC 32 RCRP	D·	n



The carry flag is intervened in the rotation loop. If M8022 has been set to ON or OFF before the rotation instruction, the carry flag is transferred to the destination.

2. 32-bit operation (DRCR and DRCRP)

"n" bits out of 32 bits of [D• +1, D•] and 1 bit (carry flag M8022) are rotated rightward.



Related device

→ **For the carry flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8022	Carry	Turns ON when the bit shifted last from the lowest position is "1".

Cautions

1. In the case of continuous operation type instructions (RCR and DRCR)

Note that shift and rotation are executed in every scan time (operation cycle).

2. When a device with digit specification is specified as D•

Only K4 (16-bit instruction) or K8 (32-bit instruction) is valid (examples: K4Y010 or K8M0).

11.4 FNC 33 – RCL / Rotation Left with Carry

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction shifts and rotates the bit information leftward by the specified number of bits together with the carry flag.

1. Instruction format

D	FNC 33 RCL	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	RCL	Continuous Operation
	RCLP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DRCL	Continuous Operation
	DRCLP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(D·)	Word device number storing data to be rotated leftward	16- or 32-bit binary
n	Number of bits to be rotated [n ≤ 16 (16-bit instruction), n ≤ 32 (32-bit instruction)]	16- or 32-bit binary

3. Applicable devices

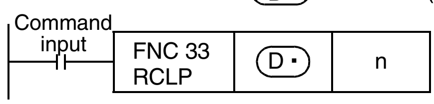
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(D·)									▲	▲	▲	✓	✓	✓	✓	✓	✓	✓	✓					
n														✓	✓					✓	✓			

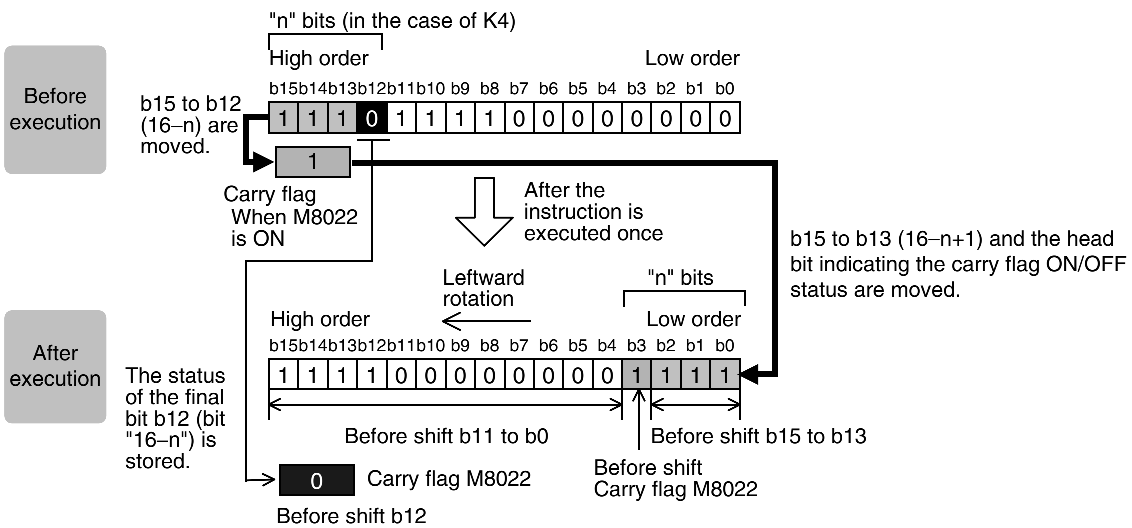
▲: In 16-bit operations, K4Y○○○, K4M○○○ and K4S○○○ are valid.
In 32-bit operations, K8Y○○○, K8M○○○ and K8S○○○ are valid.

Explanation of function and operation

1. 16-bit operation (RCL and RCLP)

"n" bits out of 16 bits of (D·) and 1 bit (carry flag M8022) are rotated leftward.

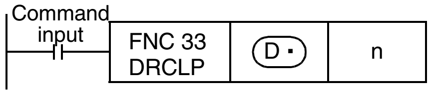


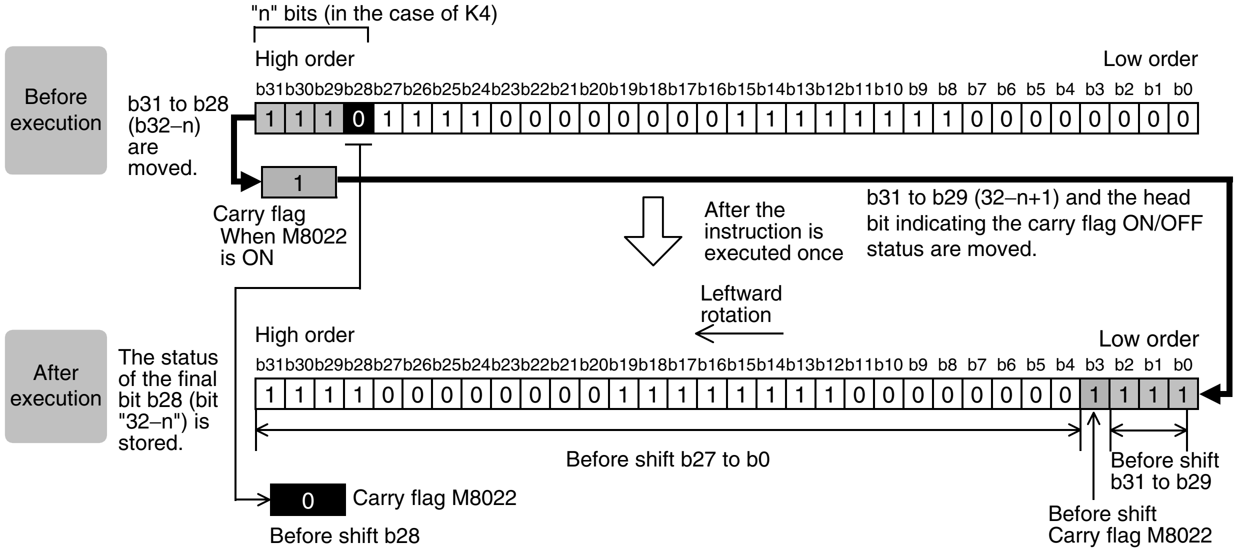


The carry flag is intervened in the rotation loop. If M8022 has been set to ON or OFF before the rotation instruction, the carry flag is transferred to the destination.

2. 32-bit operation (DRCL and DRCLP)

"n" bits out of 32 bits of [(D·)+1, (D·)] and 1 bit (carry flag M8022) are rotated leftward.





Related device

→ For the carry flag use method, refer to Subsection 6.5.2.

Device	Name	Description
M8022	Carry	Turns ON when the bit shifted last from the highest position is "1".

Cautions

1. In the case of continuous operation type instructions (RCL and DRCL)

Note that shift and rotation are executed in every scan time (operation cycle).

2. When a device with digit specification is specified as (D·)

Only K4 (16-bit instruction) or K8 (32-bit instruction) is valid (examples: K4Y010 or K8M0).

11.5 FNC 34 – SFTR / Bit Shift Right

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction shifts bit devices of the specified bit length rightward by the specified number of bits.
After shift, the bit device (S•) is transferred by "n2" bits from the most significant bit.

1. Instruction format

FNC 34 SFTR	P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	SFTR	Continuous Operation
	SFTRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand Type	Description	Data Type
(S•)	Head bit device number to be stored to the shift data after rightward shift	Bit
(D•)	Head bit device number to be shifted rightward	Bit
n1	Bit length of the shift data $n2 \leq n1 \leq 1024$	16-bit binary
n2	Number of bits to be shifted rightward $n2 \leq n1 \leq 1024$	16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S•)	✓	✓	✓			✓	▲												✓					
(D•)		✓	✓			✓													✓					
n1																				✓	✓			
n2														✓	✓					✓	✓			

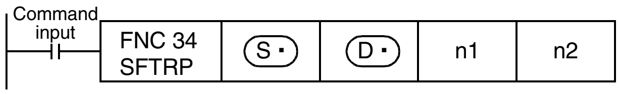
▲: "D□.b" cannot be indexed with index registers (V and Z).

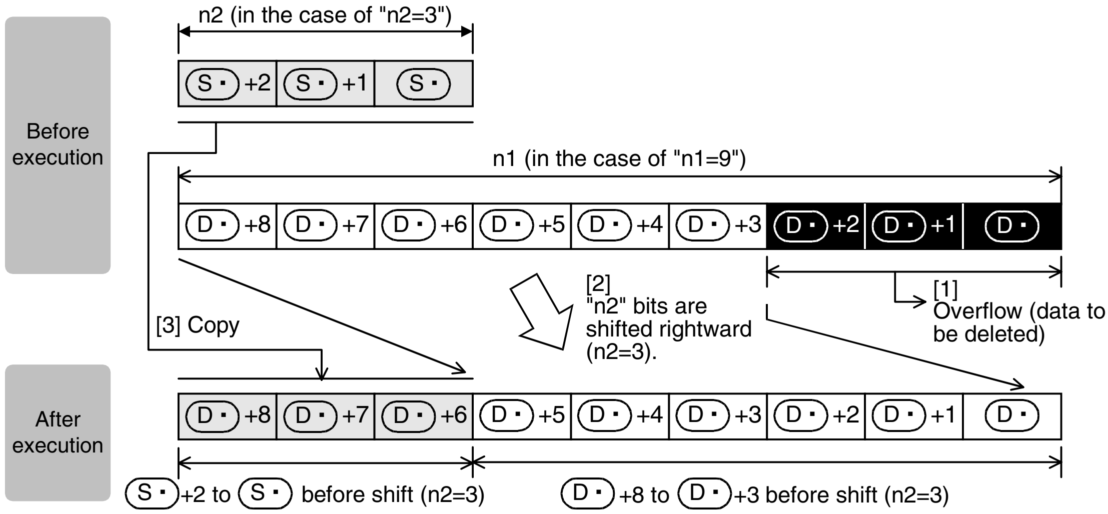
Explanation of function and operation

1. 16-bit operation (SFTR and SFTRP)

For "n1" bits (shift register length) starting from D·, "n2" bits are shifted rightward ([1] and [2] shown below).

After shift, "n2" bits from S· are transferred to "n2" bits from D·+n1-n2 ([3] shown below).





Caution

Note that "n2" bits are shifted every time the command input turns ON from OFF in SFTRP instruction, but that "n2" bits are shifted in each scan time (operation cycle) in SFTR instruction.

Error

If the transfer source S· is equivalent to the shifted device D·, an operation error occurs (error code: K6710).

11.6 FNC 35 – SFTL / Bit Shift Left

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction shifts bit devices of the specified bit length leftward by the specified number of bits.
After shift, the bit device (S•) is transferred by "n2" bits from the least significant bit.

1. Instruction format

FNC 35
SFTL P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	SFTL	Continuous Operation
	SFTLP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand Type	Description	Data Type
(S•)	Head bit device number to be stored to the shift data after leftward shift	Bit
(D•)	Head bit device number to be shifted leftward	Bit
n1	Bit length of the shift data n2 ≤ n1 ≤ 1024	16-bit binary
n2	Number of bits to be shifted leftward n2 ≤ n1 ≤ 1024	16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S•)	✓	✓	✓			✓	▲												✓					
(D•)		✓	✓			✓													✓					
n1																				✓	✓			
n2														✓	✓					✓	✓			

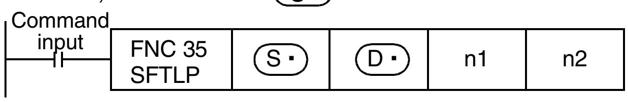
▲: "D□.b" cannot be indexed with index registers (V and Z).

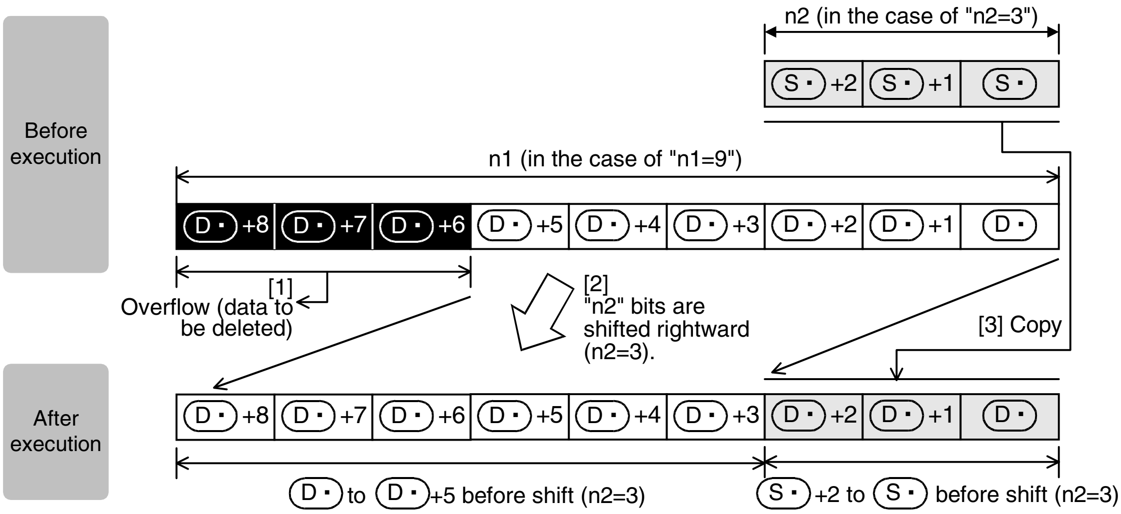
11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FMC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

Explanation of function and operation

1. 16-bit operation (SFTL and SFTLP)

For "n1" bits (shift register length) starting from (D·), "n2" bits are shifted leftward ([1] and [2] shown below). After shift, "n2" bits from (S·) are transferred to "n2" bits from (D·) ([3] shown below).





Caution

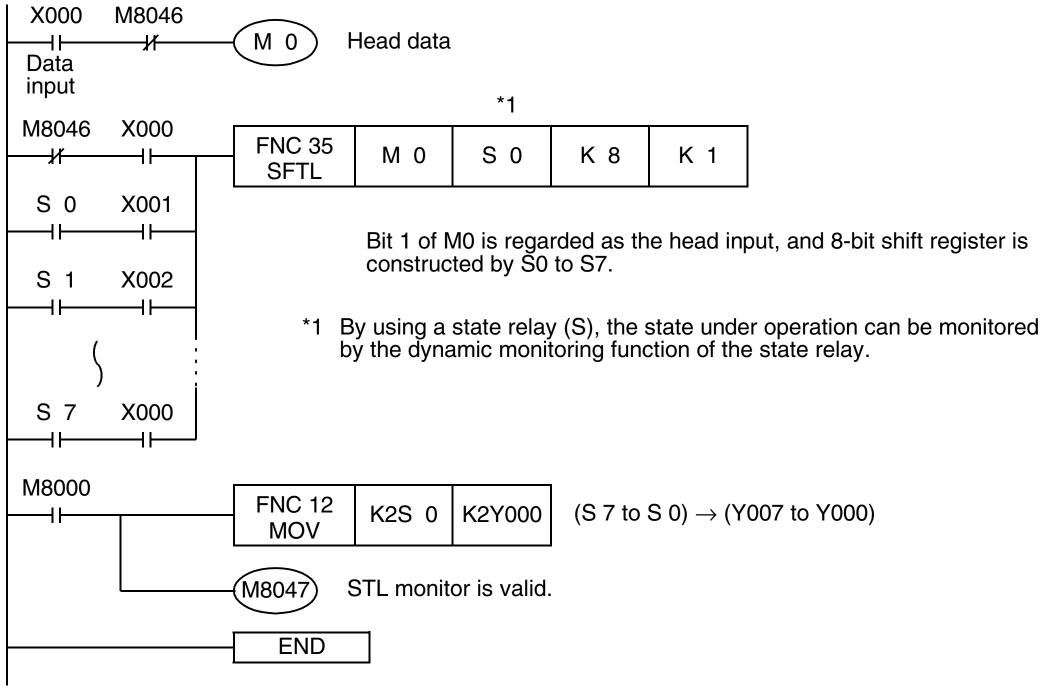
Note that "n2" bits are shifted every time the command input turns ON from OFF in SFTLP instruction, but that "n2" bits are shifted in each operation cycle in SFTL instruction.

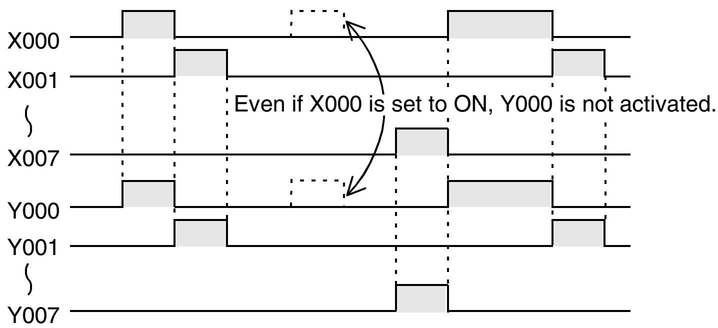
Error

If the transfer source (S·) is equivalent to the shifted device (D·), an operation error occurs (error code: K6710).

Program example (conditional stepping of 1-bit data)

By setting X000 to X007 to ON in turn, Y000 to Y007 are activated in turn.
If the order is wrong, activation is disabled.





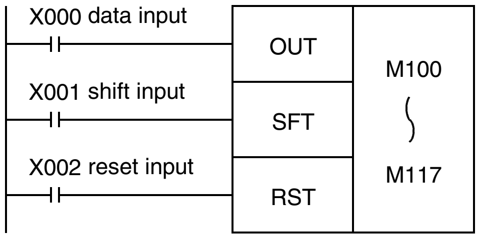
11.6.1 Replacement of SFT instruction in F1 and F2 Series

SFT instruction in F1/F2 PLCs corresponds to SFTL (FNC 35) instruction in FX3U/FX3UC PLCs as shown below:

1. F1 and F2 PLCs

M100: Input data
M101 to M117 (octal number): 15-step shift register

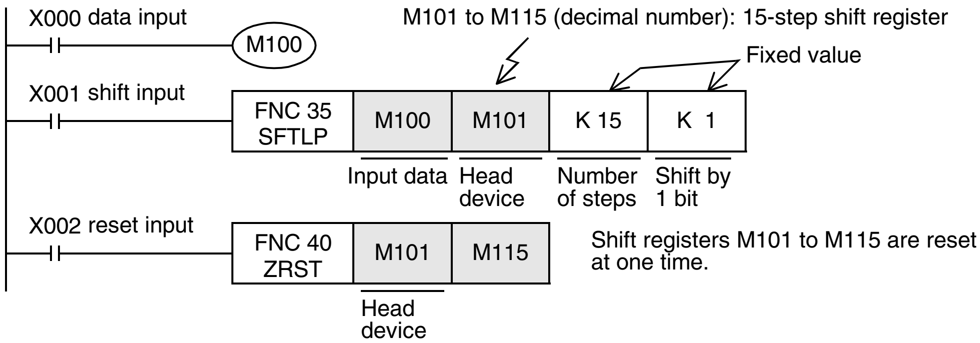
Circuit program



List program

```
0000  LD    X000
0001  OUT   M100
0002  LD    X001
0003  SFT   M100
0004  LD    X002
0005  RST   M100
```

2. FX3U and FX3UC PLCs



11.7 FNC 36 – WSFR / Word Shift Right

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction shifts word devices with "n1" data length rightward by "n2" words.

1. Instruction format

FNC 36 WSFR P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	WSFR	Continuous Operation
	WSFRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand Type	Description	Data Type
S·	Head device number to be stored to the shift data after rightward shift	16-bit binary
D·	Head word device number storing data to be shifted rightward	16-bit binary
n1	Word data length of the shift data $n2 \leq n1 \leq 512$	16-bit binary
n2	Number of words to be shifted rightward $n2 \leq n1 \leq 512$	16-bit binary

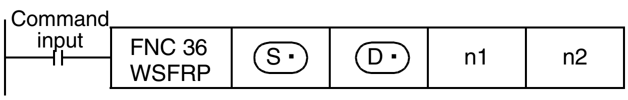
3. Applicable devices

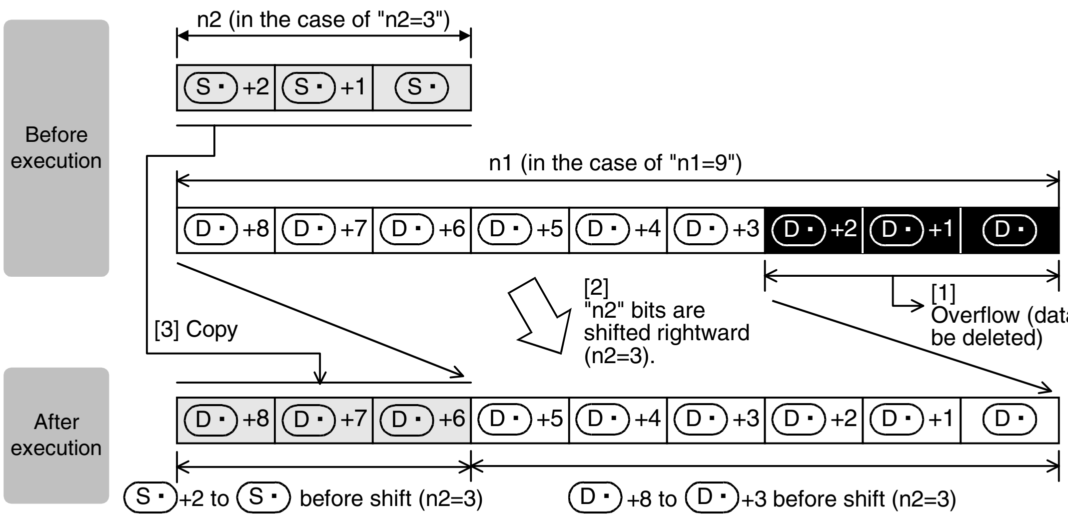
Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
D·									✓	✓	✓	✓	✓	✓	✓	✓			✓					
n1																				✓	✓			
n2														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (WSFR and WSFRP)

For "n1" word devices starting from D·, "n2" words are shifted rightward ([1] and [2] shown below).
After shift, "n2" words starting from S· are shifted to "n2" words starting from [D·+n1-n2] ([3] shown below).





Caution

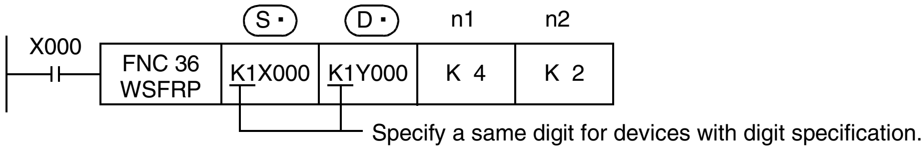
Note that "n2" words are shifted when the drive input turns ON in WSFRP instruction, but that "n2" words are shifted in each operation cycle in WSFR instruction.

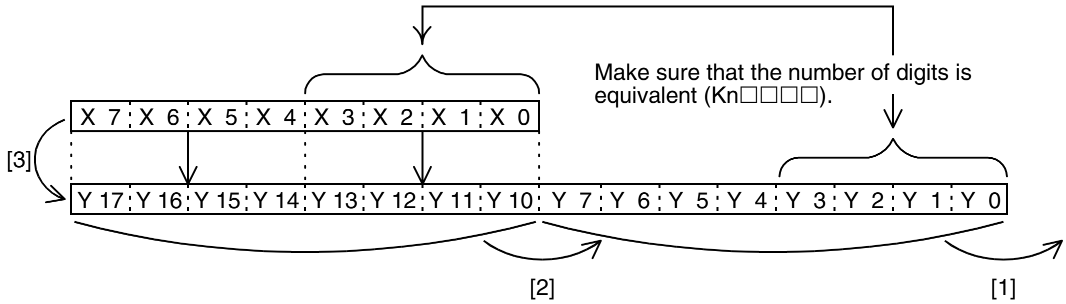
Error

If the transfer source (S·) is equivalent to the shifted device (D·), an operation error occurs (error code: K6710).

Program example

1. Shifting devices with digit specification





11.8 FNC 37 – WSFL / Word Shift Left

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction shifts the word data information leftward by the specified number of words.

1. Instruction format

FNC 37 WSFL P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	WSFL	Continuous Operation
	WSFLP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand Type	Description	Data Type
S•	Head device number to be stored to the shift data after leftward shift	16-bit binary
D•	Head word device number storing data to be shifted leftward	16-bit binary
n1	Word data length of the shift data $n2 \leq n1 \leq 512$	16-bit binary
n2	Number of words to be shifted leftward $n2 \leq n1 \leq 512$	16-bit binary

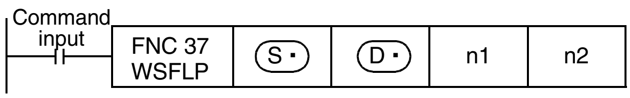
3. Applicable devices

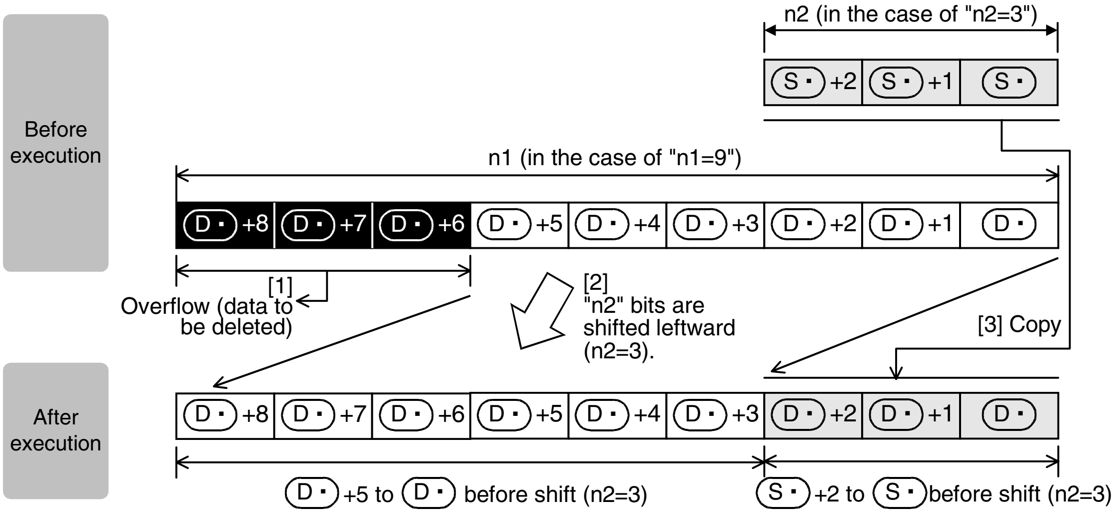
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
D•									✓	✓	✓	✓	✓	✓	✓	✓			✓					
n1																				✓	✓			
n2														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (WSFL and WSFLP)

For "n1" word devices starting from D•, "n2" words are shifted leftward ([1] and [2] shown below).
After shift, "n2" words starting from S• are shifted to "n2" words starting from D• ([3] shown below).





Caution

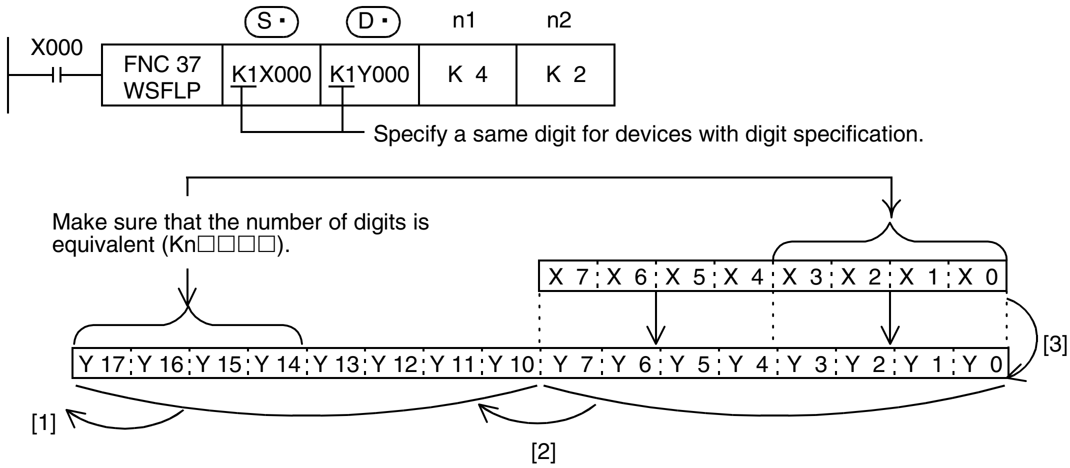
Note that "n2" words are shifted every time the drive input turns ON from OFF in WSFLP instruction, but that "n2" words are shifted in each operation cycle in WSFL instruction.

Error

If the transfer source (S·) is equivalent to the shifted device (D·), an operation error occurs (error code: K6710).

Program example

1. Shifting devices with digit specification



11.9 FNC 38 – SFWR / Shift Write [FIFO/FILO Control]

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction writes data for first-in first-out (FIFO) and last-in first-out (LIFO) control.

1. Instruction format

FNC 38 SFWR	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	SFWR	Continuous Operation
	SFWRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand Type	Description	Data Type
S·	Word device number storing data to be put in first	16-bit binary
D·	Head word device number storing data (The first word device works as the pointer, and data is stored in D· +1 and later)	16-bit binary
n	Number of store points plus "1"*1 $2 \leqq n \leqq 512$	16-bit binary

*1. "+1" is required for the pointer.

3. Applicable devices

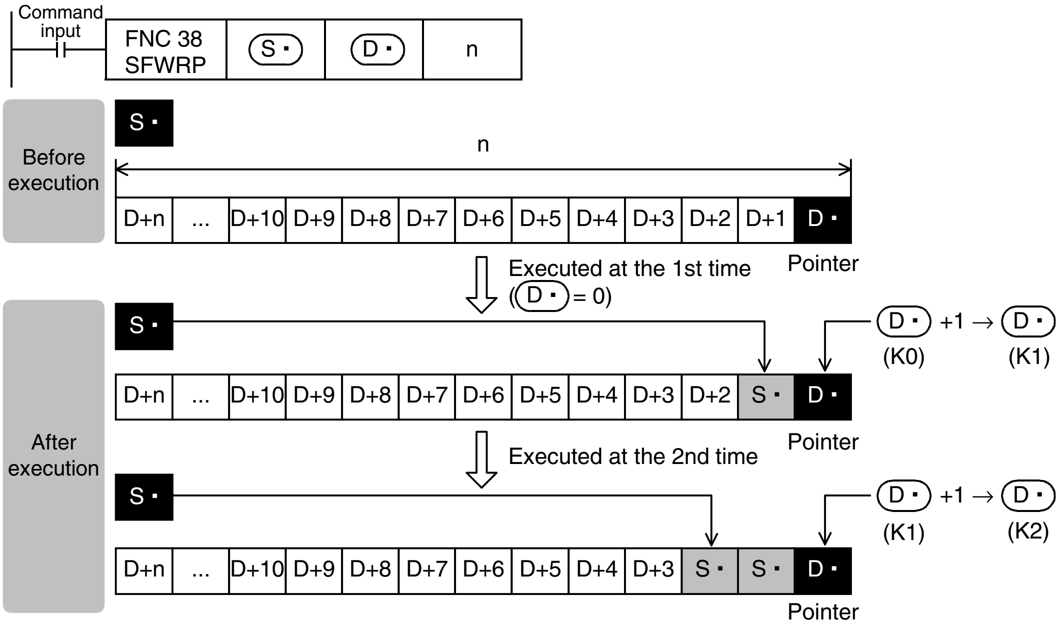
Oper-and Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D·									✓	✓	✓	✓	✓	✓	✓	✓			✓					
n																				✓	✓			

Explanation of function and operation

1. 16-bit operation (SFWR and SFWRP)

The contents of (S·) are written to "n-1" devices from (D·)+1, and "1" is added to the number of data stored in (D·).

For example, when (D·) = 0, the contents of (S·) are written to (D·)+1. When (D·) = 1, the contents of (S·) are written to (D·)+2.



1) When X000 turns ON from OFF, the contents of (S·) are stored to (D·)+1. So the contents of (D·)+1 become equivalent to (S·).
2) When the contents of (S·) are changed and then the command input is set to ON from OFF again, the new contents of (S·) are stored to (D·)+2. So the contents of (D·)+2 become equivalent to (S·). (When the continuous operation type SFWR instruction is used, the contents are stored in each operation cycle. Use the pulse operation type SFWRP instruction in programming.)
3) Data are stored from the right end in the same way, and the number of stored data is specified by the contents of the pointer (D·).

Related device

→ **For the carry flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8022	Carry	When the contents of the pointer (D·) exceeds "n-1", no operation is executed (so data is not written) and the carry flag M8022 turns ON.

Related instructions

Instruction	Description
SFRD (FNC 39)	Shift read (for FIFO control)
POP (FNC212)	Shift last data read (for FILO control)

Caution

1. In the case of continuous operation type (SFWR) instruction

Note that data are stored (overwritten) in each scan time (operation cycle).

Program example

1. Example of first-in first-out control

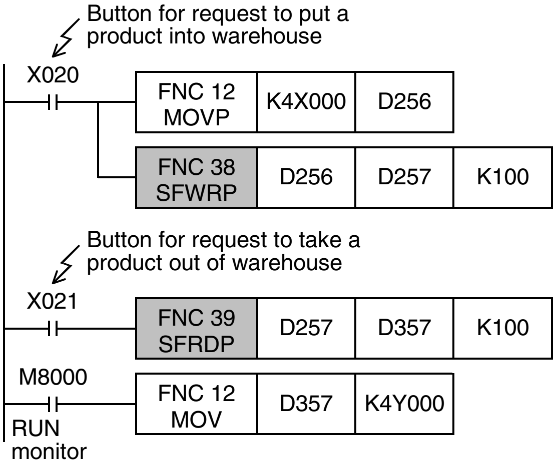
→ **For a program example of FILO, refer to Section 27.3.**

In the example below, the shift write (SFWR) and shift read (SFRD) instructions are used.

1) Contents of operation

- In this circuit example, a product number to be taken out now is output according to "first-in first-out" rule while products which were put into a warehouse with their product numbers registered are taken out of the warehouse.
- The product number is hexadecimal, and up to 4 digits. Up to 99 products can be stored in the warehouse.

2) Program

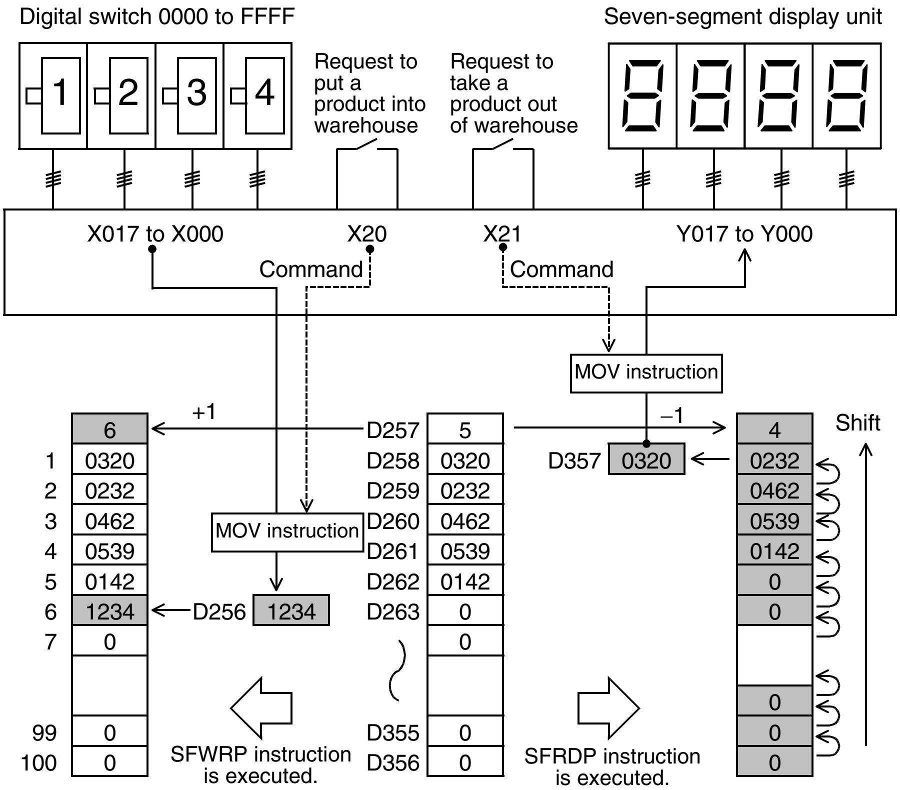


The product number is input from X000 to X017, and transferred to D256.

Pointer
D257: Data register for storing the product number
D258 to D356 (99 points)

The product number of a product put into first is output to D357 in response to the request to put a product out of the warehouse.

The product number to be taken out is output to Y000 to Y017 in a four-digit hexadecimal number.



11.10 FNC 39 – SFRD / Shift Read [FIFO Control]

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction reads data for first-in first-out control.

1. Instruction format

FNC 39 SFRD P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	SFRD	Continuous Operation
	SFRDP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand Type	Description	Data Type
S•	Head word device number storing data (The first word device works as the pointer, and data is stored in S• +1 and later.)	16-bit binary
D•	Word device number storing data taken out first $2 \le n \le 512$	16-bit binary
n	Number of store points plus "1"*1 $2 \le n \le 512$	16-bit binary

*1. "+1" is required for the pointer.

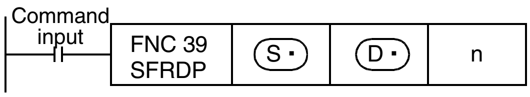
3. Applicable devices

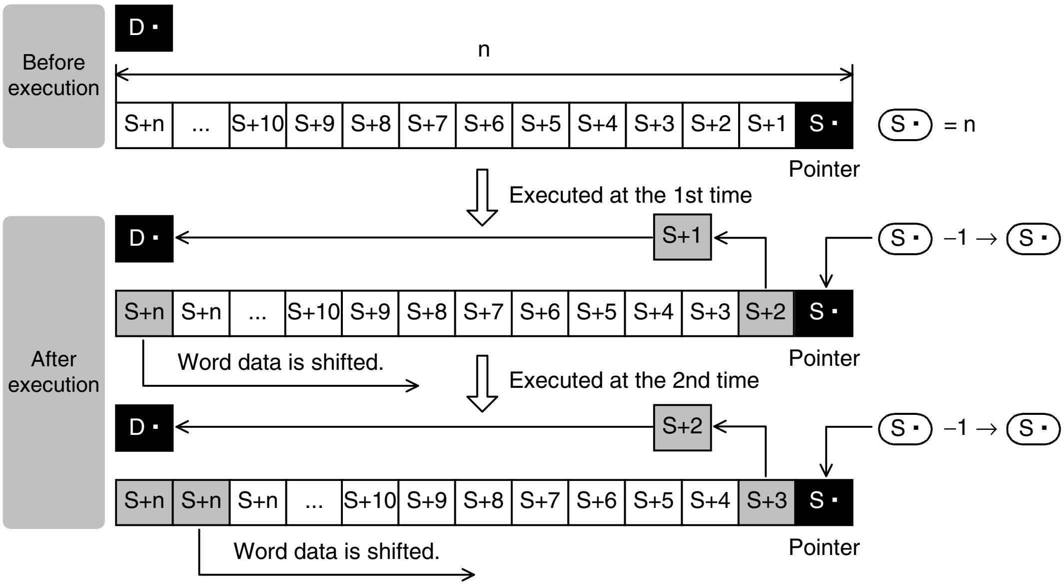
Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•									✓	✓	✓	✓	✓	✓	✓	✓			✓					
D•									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
n																				✓	✓			

Explanation of function and operation

1. 16-bit operation (SFRD and SFRDP)

S1• +1 written in turn by SFWR (FNC 38) instruction is transferred (read) to D•, and "n-1" words from S1• +1 are shifted rightward by 1 word. "1" is subtracted from the number of data, stored in S•.





1) When the command input turns ON, the contents of (S·)+1 are transferred (read) to (D·).
2) Accompanied by this transfer, the contents of the pointer (S·) decrease, and the data on the left side are shifted rightward by 1 word. (When the continuous operation type SFRD instruction is used, the contents are stored in turn in each operation cycle. Use the pulse operation type SFRDP instruction in programming.)

Related device

→ For the zero flag use method, refer to Subsection 6.5.2.

Device	Name	Description
M8020	Zero	Data is always read from (S·)+1. When the contents of the pointer (S·) become "0", the zero flag M8020 turns ON. (When (D·) is "0", no processing is executed and M8020 turns ON.)

Related instructions

Instruction	Description
SFWR (FNC 38)	Shift write (for FIFO/FILO control)
POP (FNC212)	Shift last data read (for FILO control)

Caution

1. Data after reading was executed

The contents of (S·)+n do not change by reading.

2. In the case of continuous operation type (SFRD) instruction

Data is read in turn in each scan time (operation cycle), but the contents of (S·)+n do not change.

Program example

Refer to the program example provided for SFWR (FNC 38) instruction.

→ For the program example, refer to Section 11.9.

12. Data Operation – FNC 40 to FNC 49

FNC 40 to FNC 49 provide instructions for executing complicated processing for fundamental applied instructions FNC 10 to FNC 39 and for executing special processing.

FNC No.	Mnemonic	Symbol	Function	Reference
40	ZRST	ZRST D1 D2	Zone Reset	Section 12.1
41	DECO	DECO S D n	Decode	Section 12.2
42	ENCO	ENCO S D n	Encode	Section 12.3
43	SUM	SUM S D	Sum of Active Bits	Section 12.4
44	BON	BON S D n	Check Specified Bit Status	Section 12.5
45	MEAN	MEAN S D n	Mean	Section 12.6
46	ANS	ANS S m D	Timed Annunciator Set	Section 12.7
47	ANR	ANR	Annunciator Reset	Section 12.8
48	SQR	SQR S D	Square Root	Section 12.9
49	FLT	FLT S D	Conversion to Floating Point	Section 12.10

12.1 FNC 40 – ZRST / Zone Reset

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction resets devices located in a zone between two specified devices at one time.
Use this instruction for restarting operation from the beginning after pause or after resetting control data.

1. Instruction format

FNC 40 ZRST P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	ZRST	Continuous Operation
	ZRSTP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description		Data type
D1·	Head bit or word device number to be reset at one time	D1· ≤ D2· Specify same type of devices.	16-bit binary
D2·	Last bit or word device number to be reset at one time		16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index				Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify		K	H	E	"□"	P
D1·		✓	✓			✓						✓	✓	✓	✓	✓			✓						
D2·		✓	✓			✓						✓	✓	✓	✓	✓			✓						

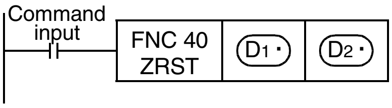
Explanation of function and operation

1. 16-bit operation (ZRST and ZRSTP)

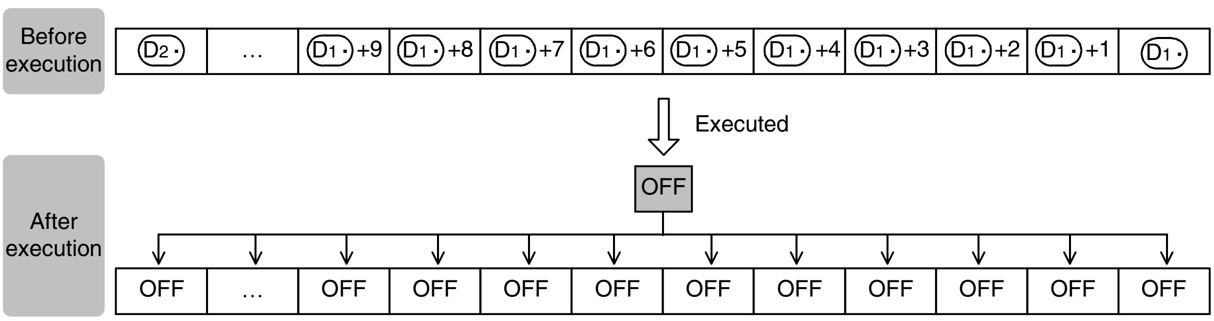
Same type of devices from D1· to D2· are reset at one time.

When D1· and D2· are bit devices

1) "OFF (reset)" is written to the entire range from D1· to D2· at one time.

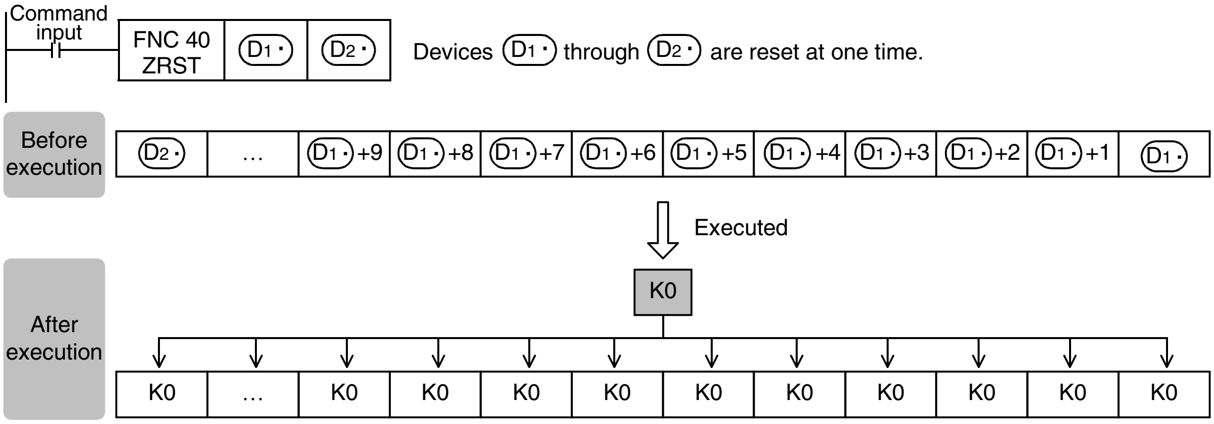


Devices D1· through D2· are reset at one time.



When (D1·) and (D2·) are word devices

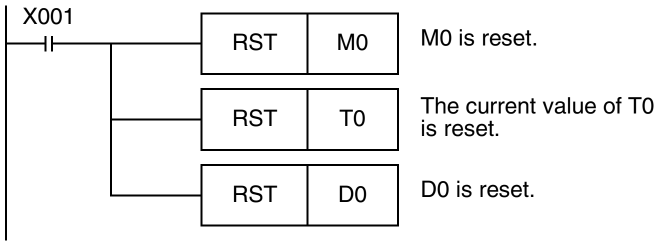
"K0" is written to the entire range from (D1·) to (D2·) at one time.



Related instructions

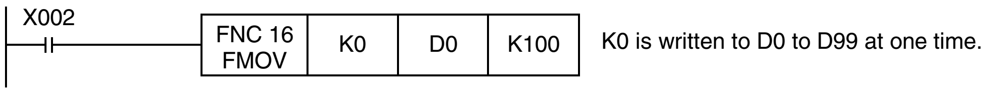
1. RST instruction

As an independent reset instruction for devices, RST instruction can be used for bit devices (Y, M and S) and word devices (T, C and D).



2. FMOV (FNC 16) instruction

FMOV (FNC 16) instruction is provided to write a constant (example: K0) at one time. By using this instruction, "0" can be written to word devices (KnY, KnM, KnS, T, C and D) at one time.



Cautions

1. Caution on specifying devices

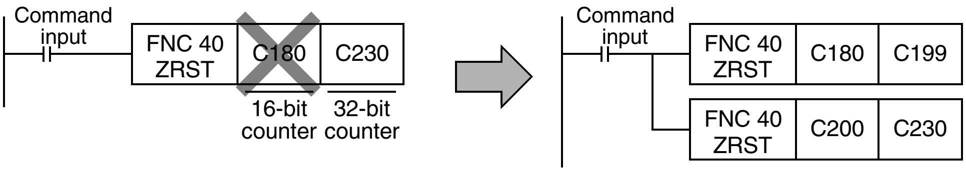
Specify same type of devices in (D1·) and (D2·). The device number of (D1·) should be smaller than or equal to the device number of (D2·).

If the device number of (D1·) is larger than the device number of (D2·), only one device specified in (D1·) is reset.

2. When specifying high speed counters (C235 to C255)

ZRST instruction is handled as the 16-bit type, but 32-bit counters can be specified in (D1·) and (D2·). However, it is not possible to specify a 16-bit counter in (D1·) and specify a 32-bit counter in (D2·); (D1·) and (D2·) should be a same type.

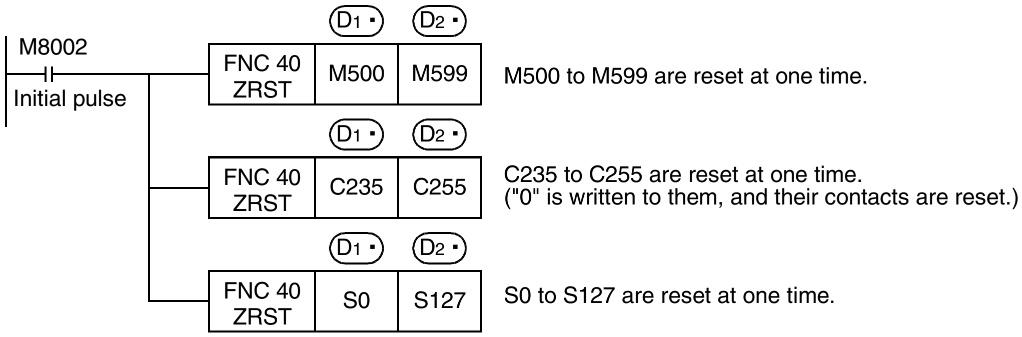
Example



Program example

1. When using devices in the latch area as non-latch type devices

In the program shown below, when the power of the PLC is turned ON or when the PLC mode is changed to RUN, the specified ranges of bit devices and word devices are reset at one time.



12.2 FNC 41 – DECO / Decode

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction converts numeric data into ON bit.
A bit number which is set to ON by this instruction indicates a numeric value.

1. Instruction format

FNC 41 DECO P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	DECO	Continuous Operation
	DECOP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
(S·)	Data to be decoded or word device number storing data	16-bit binary
(D·)	Bit or word device number storing the decoding result	16-bit binary
n	Number of bits of device storing the decoding result (n = 1 to 8) (No processing is executed in the case of "n = 0".)	16-bit binary

3. Applicable devices

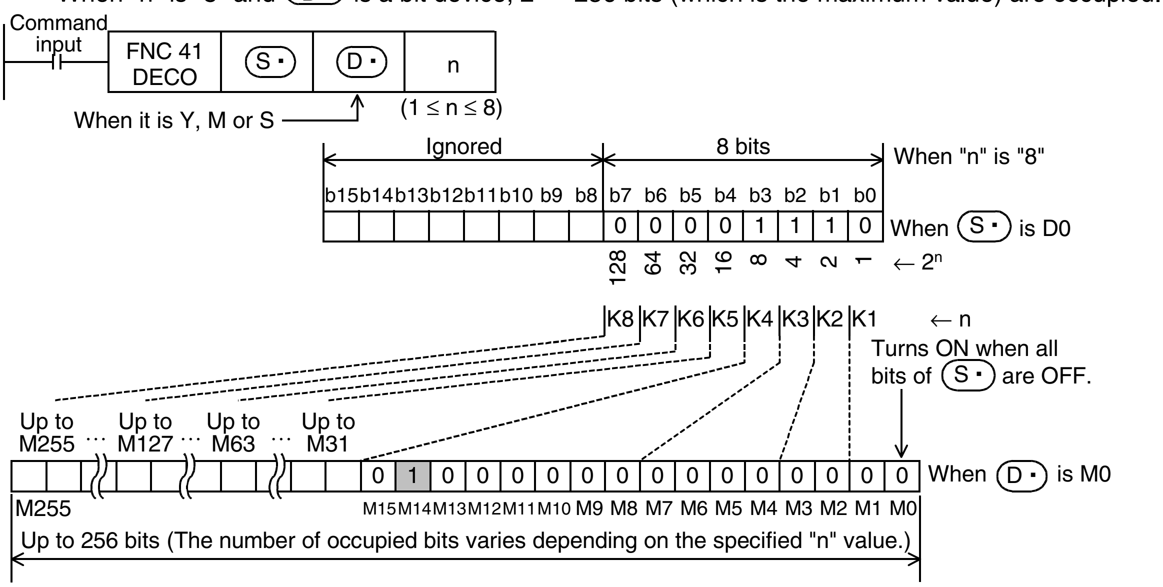
Oper-and Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)	✓	✓	✓			✓						✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
(D·)		✓	✓			✓						✓	✓	✓	✓	✓			✓					
n																				✓	✓			

Explanation of function and operation

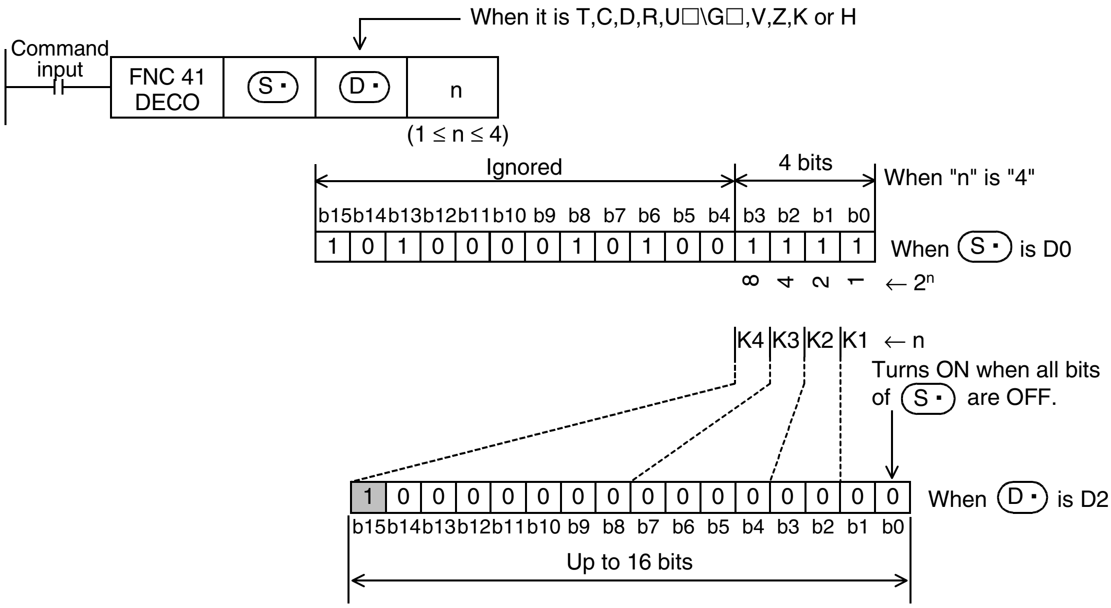
1. 16-bit operation (DECO and DECOP)

One bit among (D·) to (D·)+2^n – 1 is set to ON according to the (S·) value.

1) When (D·) is a bit device (1 ≤ n ≤ 8)
The numeric value (expressed in 2^n, 1 ≤ n ≤ 8) of a device specified by (S·) is decoded to (D·).
- When all bits of (S·) are "0", the bit device (D·) turns ON.
- When "n" is "8" and (D·) is a bit device, 2^8 = 256 bits (which is the maximum value) are occupied.



2) When (D·) is a word device ($1 \leq n \leq 4$)
The numeric value (expressed in 2^n on the low-order side) of (S·) is decoded to (D·).
- When all bits of (S·) are "0", b0 of the word device (D·) turns ON.
- In the case of "$n \leq 3$", all of high-order bits of (D·) become "0" (turn OFF).



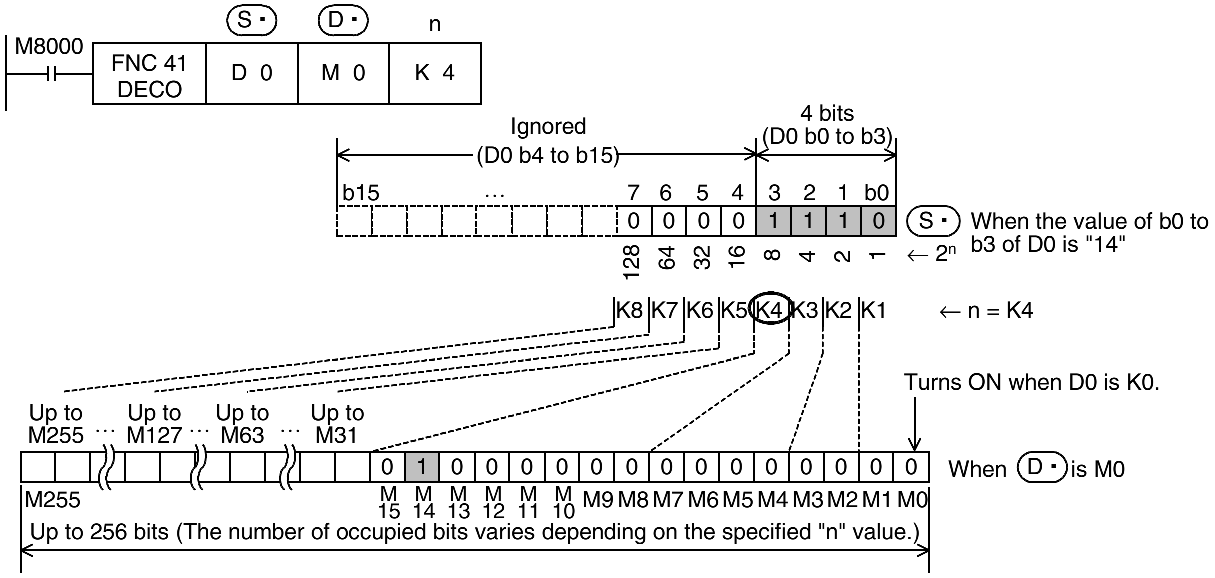
Caution

- While the command input is OFF, the instruction is not executed. The activated decode output is held in the previous ON/OFF status.
- When "n" is "0", the instruction executes no processing.

Program example

1. When setting bit devices to ON according to the value of a data register

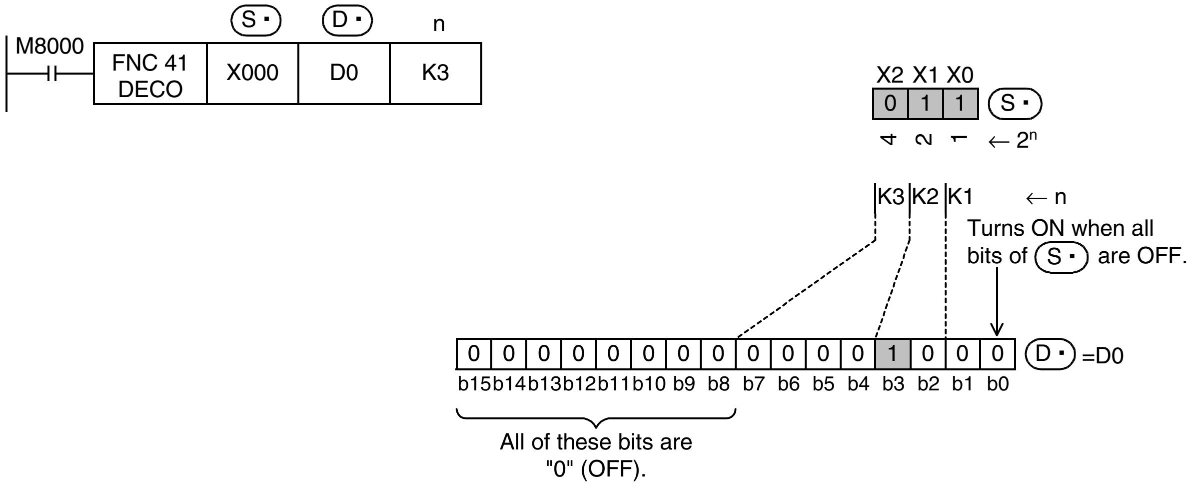
The value of D0 (whose current value is "14" in this example) is decoded to M0 to M15.



- When the value of b0 to b3 of D0 is "14 (= 0 + 2 + 4 + 8)", M14 (which is the 15th from M0) becomes "1" (turn ON).
- When the value of D0 is "0", M0 becomes "1" (turns ON).
- When "n" is set to "K4", either one point among M0 to M15 turns ON according to the value of D0 (0 to 15).
- By changing "n" from K1 to K8, D0 can correspond to numeric values from 0 to 255.
  However, because the device range of D· is occupied for decoding accordingly, such device range should not be used for another control.

2. Turning ON the bit out of word devices according to the contents of bit devices

The value expressed by X000 to X002 is decoded to D0 (X000 and X001 are ON, and X002 is OFF in this example.).



- When the values expressed by X000 to X002 are "3 (= 1 + 2 + 0)", b3 (which is the 4th from b0) becomes 1 (turns ON).
- When all of X000 to X002 are "0" (OFF), b0 becomes "1" (turns ON).

12.3 FNC 42 – ENCO / Encode

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction obtains positions in which bits are ON in data.

1. Instruction format

FNC 42 ENCO P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	ENCO	Continuous Operation
	ENCOP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
S·	Data to be encoded or word device number storing data	16-bit binary
D·	Word device number storing the encoding result	16-bit binary
n	Number of bits of device storing the encoding result (n = 1 to 8) (When "n" is "0", no processing is executed.)	16-bit binary

3. Applicable devices

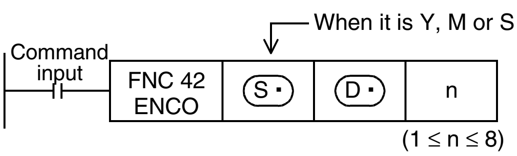
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·	✓	✓	✓			✓						✓	✓	✓	✓	✓	✓	✓	✓					
D·												✓	✓	✓	✓	✓	✓	✓	✓					
n																				✓	✓			

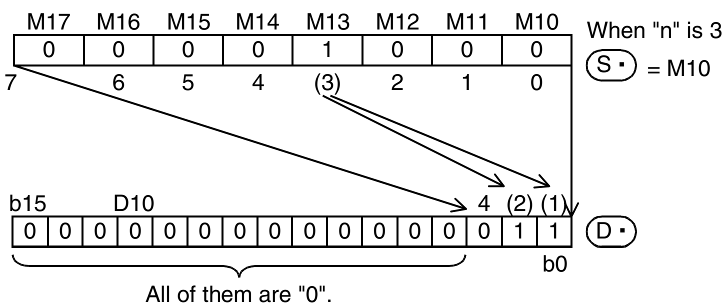
Explanation of function and operation

1. 16-bit operation (ENCO and ENCOP)

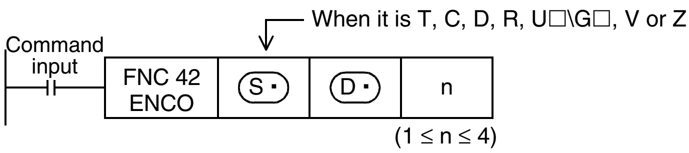
The 2^n bit of S· is encoded, and the result value is stored to D·.
This instruction converts data into binary data according to a bit position in the ON status.

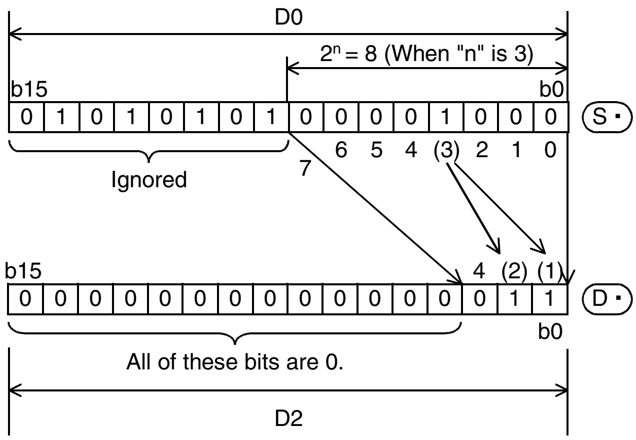
1) When S· is a bit device ($1 \le n \le 8$)
ON bit positions among "2^n" bits ($1 \le n \le 8$) from S· are encoded to D·.
- When "n" is "8", 2^8 = 256 bits (which is the maximum value) are occupied.
- The encoding result of D· is "0" (OFF) from the most significant bit to the low-order bit "n".





2) When (S·) is a word device ($1 \le n \le 4$)
ON bit positions among "2^n" bits ($1 \le n \le 4$) from a device specified in (S·) are encoded to (D·).
- The encoding result of (D·) is "0" (OFF) from the most significant bit to the low-order bit "n".





Cautions

1. When two or more bits are ON in the (S·) data
The low-order side is ignored, and only the ON position on the high-order side is encoded.

2. While the command input is OFF
The instruction is not executed. Activated encode outputs are latched in the previous ON/OFF status.

12.4 FNC 43 – SUM / Sum of Active Bits

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction counts the number of "1" (ON) bits in the data of a specified device.

1. Instruction format

D	FNC 43 SUM	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	SUM	Continuous Operation
	SUMP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DSUM	Continuous Operation
	DSUMP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S·	Word device number storing the source data	16- or 32-bit binary
D·	Word device number storing the result data	16- or 32-bit binary

3. Applicable devices

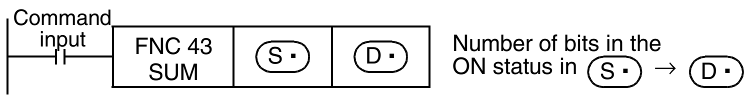
Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D·									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (SUM and SUMP)

The number of bits in the ON status in S· is counted, and stored to D·.

- When all bits are OFF in S·, the zero flag M8020 turns ON.



Word data

16-bit data

Before execution

	Device	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S·	K21847	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1
			9 ←		8 ←		7 ←		6 ←		5 ←		4 ←		3	2	1 (Number of "1" (ON))
D·	K0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

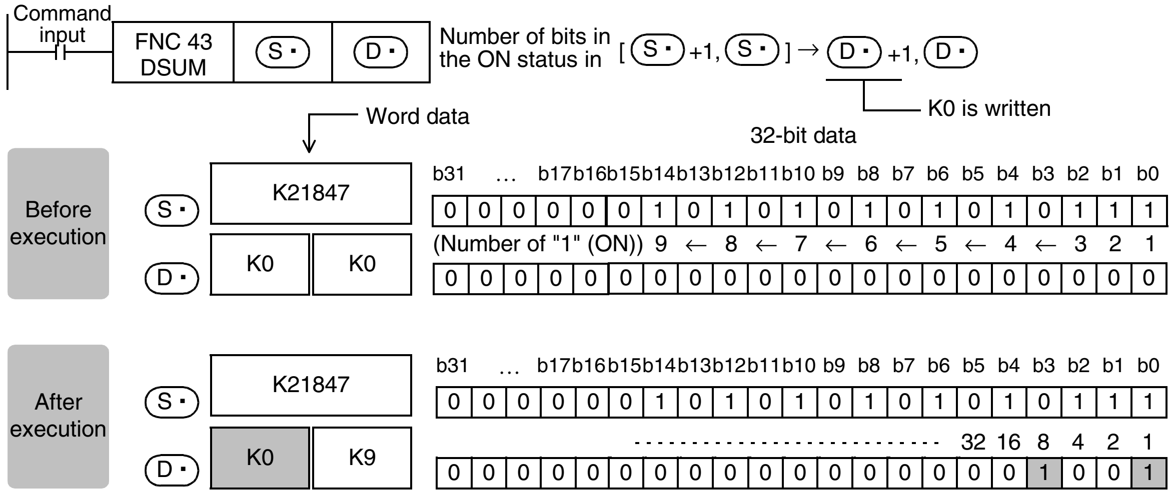
After execution

	Device	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S·	K21847	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1
		- - -											32	16	8	4	2	1
D·	K9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

2. 32-bit operation (DSUM and DSUMP)

The number of bits in the ON status in [S·+1, S·] is counted, and stored to D·.

- The number of bits in the ON status are stored in D·, and K0 is stored in D·+1.
- When all bits are OFF in [S·+1, S·], the zero flag M8020 turns ON.



3. Operation result D· according to the S· value (in 16-bit operation)

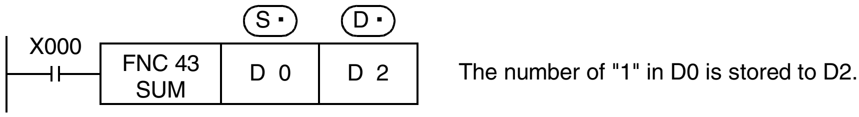
S·																		D·	M8020 (zero flag)
Bit device																Word device			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	Decimal	Hexadecimal		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	0	ON
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0001	1	OFF
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0002	1	OFF
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	3	0003	2	OFF
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4	0004	1	OFF
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	5	0005	2	OFF
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	6	0006	2	OFF
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	7	0007	3	OFF
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	8	0008	1	OFF
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	9	0009	2	OFF
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	10	000A	2	OFF
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	11	000B	3	OFF
⋮																⋮	⋮	⋮	OFF
1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	–5	FFFB	15	OFF
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	–4	FFFC	14	OFF
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	–3	FFFD	15	OFF
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	–2	FFFE	15	OFF
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	–1	FFFF	16	OFF

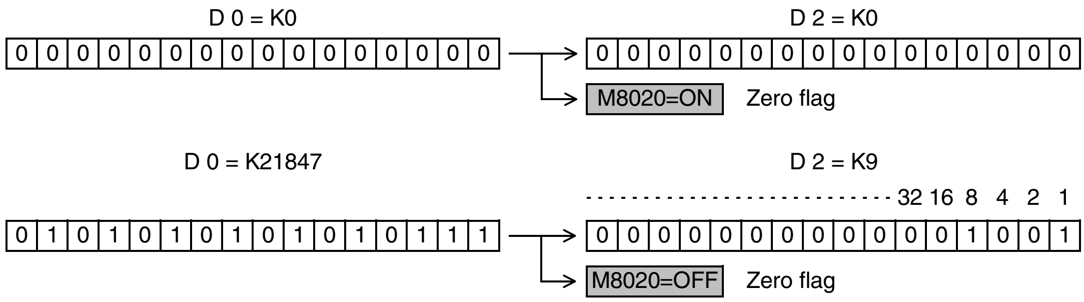
Caution

While the command input is OFF, the instruction is not executed.
The output of the number of bits in the ON status is latched in the previous status.

Program example

When X000 is ON, the number of bits in the ON status in D0 is counted, and stored to D2.





12.5 FNC 44 – BON / Check Specified Bit Status

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction checks whether a specified bit position in a specified device is ON or OFF.

1. Instruction format

	FNC 44	
D	BON	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	BON	Continuous Operation
	BONP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DBON	Continuous Operation
	DBONP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
(S·)	Word device number storing the source data	16- or 32-bit binary
(D·)	Bit device number to be driven	16- or 32-bit binary
n	Bit position to be checked [n: 0 to 15 (16-bit instruction), 0 to 31 (32-bit instruction)]	16- or 32-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
(D·)		✓	✓			✓	▲												✓					
n														✓	✓					✓	✓			

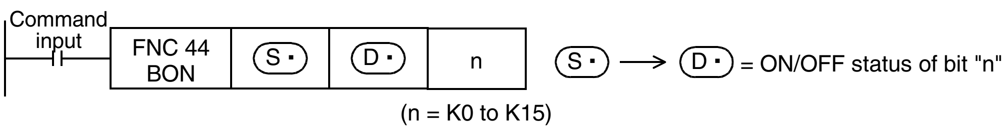
▲ : "D□.b" cannot be indexed with index registers (V and Z).

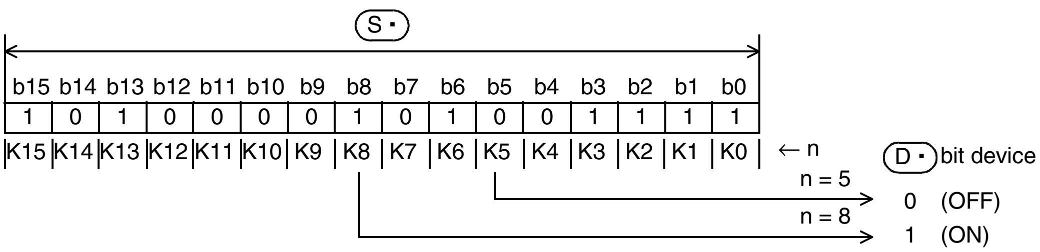
Explanation of function and operation

1. 16-bit operation (BON and BONP)

The status (ON or OFF) of the bit "n" in (S·) is output to (D·).
[When the bit "n" is ON, (D·) is set to ON. When the bit "n" is OFF, (D·) is set to OFF.]

- When a constant (K) is specified as the transfer source (S·), it is automatically converted into the binary format.

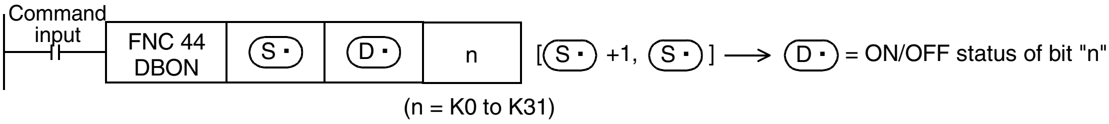


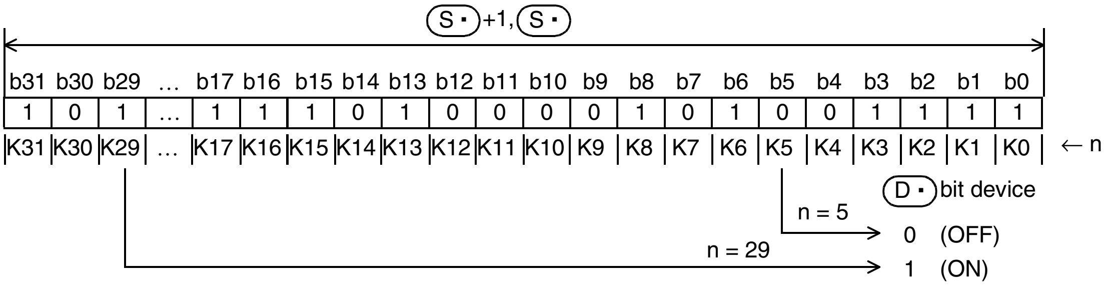


2. 32-bit operation (DBON and DBONP)

The status (ON or OFF) of the bit "n" in [S·+1, S·] is output to D·.
[When the bit "n" is ON, D· is set to ON. When the bit "n" is OFF, D· is set to OFF.]

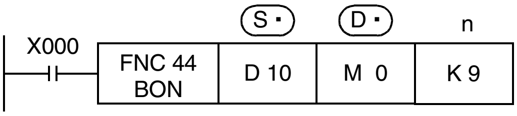
- When a constant (K) is specified as the transfer source [S·+1, S·], it is automatically converted into the binary format.

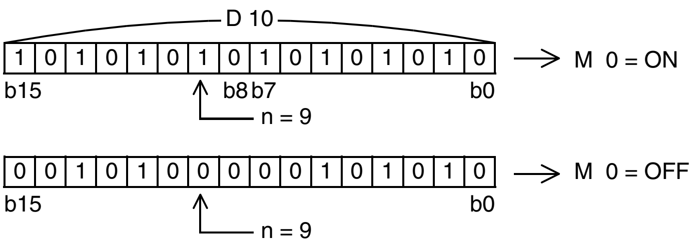




Program example

When the bit 9 (n = 9) in D10 is "1" (ON), M0 is set to "1" (ON).





12.6 FNC 45 – MEAN / Mean

FX3UC Ver.1.00 ⇒ FX3U Ver.2.20 ⇒

Outline

This instruction obtains the mean value of data.

1. Instruction format

D	FNC 45 MEAN	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	MEAN	Continuous Operation
	MEANP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DMEAN	Continuous Operation
	DMEANP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S•	Head word device number storing data to be averaged	16- or 32-bit binary
D•	Word device number storing the mean value result	16- or 32-bit binary
n	Number of data to be averaged (1 ≦ n ≦ 64)	16- or 32-bit binary

3. Applicable devices

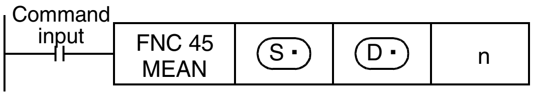
Operand Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
D•									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (MEAN and MEANP)

The mean value of "n" 16-bit data from S• is stored to D•.

- The sum is obtained as algebraic sum, and divided by "n".
- The remainder is ignored.

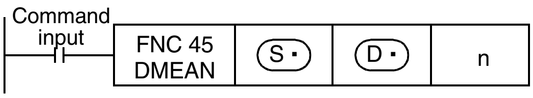


$$\frac{(S\cdot) + (S\cdot)+1 + \ldots\ldots + (S\cdot)+n-1}{n} \longrightarrow (D\cdot)$$

2. 32-bit operation (DMEAN and DMEANP)

The mean value of "n" 32-bit data from [S•+1, S•] is stored to [D•+1, D•].

- The sum is obtained as algebraic sum, and divided by "n".
- The remainder is ignored.



$$\frac{[(S\cdot)+1, (S\cdot)] + [(S\cdot)+3, (S\cdot)+2] + \ldots\ldots + [\{(S\cdot)+n\times 2-1\}, \{(S\cdot)+n\times 2-2\}]}{n} \longrightarrow [(D\cdot)+1, (D\cdot)]$$

Caution

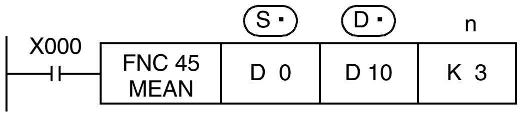
When a device number is exceeded, "n" is handled as a smaller value in the possible range.

Error

When "n" is any value outside the range from "1" to "64", an operation error (M8067) is caused.

Program example

The data of D0, D1 and D2 are summed, divided by "3", and then stored to D10.



$$\frac{(D\ 0) + (D\ 1) + (D\ 2)}{3} \rightarrow (D\ 10)$$

12.7 FNC 46 – ANS / Timed Annunciator Set

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction sets a state relay as an annunciator (S900 to S999).

1. Instruction format

FNC 46
ANS

16-bit Instruction	Mnemonic	Operation Condition
7 steps	ANS	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand type	Description	Data type
(S·)	Timer number for evaluation time	16-bit binary
m	Evaluation time data [m = 1 to 32767 (unit: 100 ms)]	16-bit binary
(D·)	Annunciator device to be set	16-bit binary

3. Applicable devices

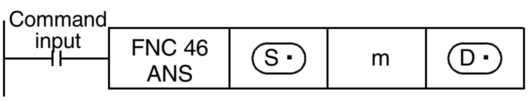
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)												▲1							✓					
m														✓	✓					✓	✓			
(D·)						▲2													✓					

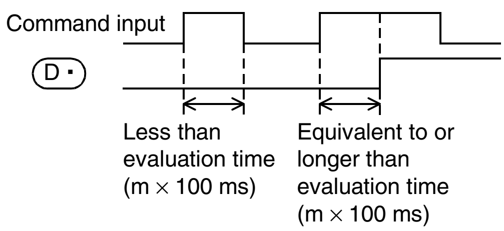
▲1 : T0 to T199
▲2 : S900 to S999

Explanation of function and operation

1. 16-bit operation

When the command input remains ON for equivalent to or longer than the evaluation time [m × 100 ms, timer (S·)], (D·) is set.
When the command input remains ON for less than the evaluation time [m × 100 ms] and then turns OFF, the current value of the timer for evaluation (S·) is reset and (D·) is not set.
When the command input turns OFF, the timer for evaluation is reset.





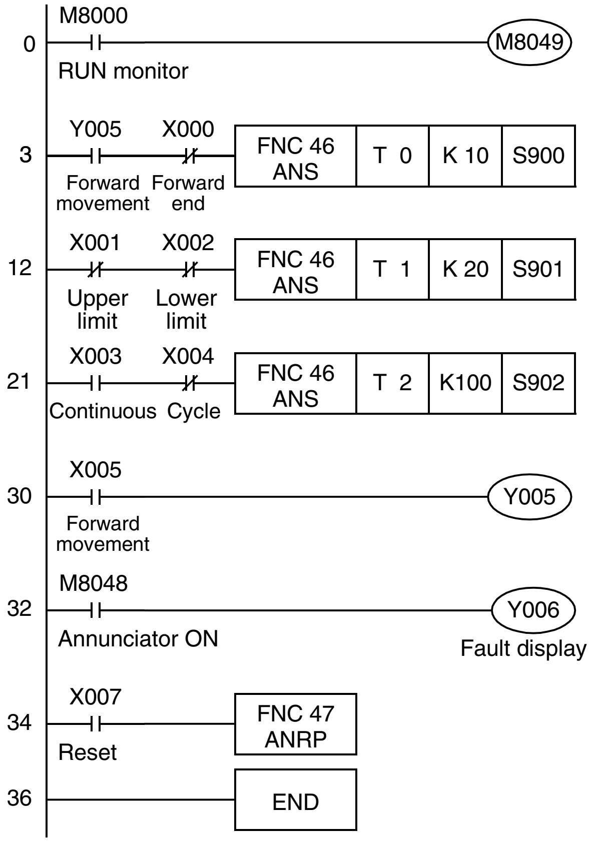
Related devices

Device	Name	Description
M8049	Enable annunciator	When M8049 is set to ON, M8048 and D8049 are valid.
M8048	Annunciator ON	When M8049 is ON and one of the state relays S900 to S999 is ON, M8048 turns ON.
D8049	Smallest state relay number in ON status	Among S900 to S999, the smallest state relay number in the ON status is stored.

Program example

1. Displaying a fault number using an annunciator

When the program for external fault diagnosis shown below is created and the content of D8049 (smallest state relay number in the ON status) is monitored, the smallest state relay number in the ON status from S900 to S999 is displayed.
If two or more faults are present at the same time, the next smallest fault number is displayed after the fault of the smallest fault number is cleared.



When M8049 turns ON, monitoring becomes valid.

If the forward end detection input X000 does not turn ON within 1 second after the forward movement output Y005 is driven, S900 turns ON.

If both the upper limit input X001 and the lower limit input X002 are OFF for 2 seconds or more due to a dog error, S901 turns ON.

The switch X004 is set to ON in one operation cycle of the machine. If the switch X004 is not set to ON while the continuous operation mode input X003 is ON in the machine whose tact time is less than 10 seconds, S902 turns ON.

When one among S900 to S999 turns ON, M8048 turns ON and the fault display output Y006 turns ON.

A state relay which was set to ON by the external fault diagnosis program is set to OFF by the reset button X007. Every time X007 is set to ON, an operation state relay in the ON status with the smallest device number is reset (set to OFF) in turn.

12.8 FNC 47 – ANR / Annunciator Reset

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction resets an annunciator (S900 to S999) in the ON status with the smallest number.

1. Instruction format

FNC 47 ANR P

16-bit Instruction	Mnemonic	Operation Condition
1 step	ANR	Continuous Operation
	ANRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
—	There is no set data.	—

3. Applicable devices

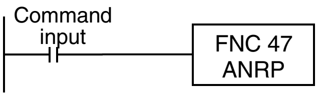
Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
—	There are no applicable devices.																							

Explanation of function and operation

1. 16-bit operation (ANR and ANRP)

When the command input turns ON, a state relay working as annunciator (S900 to S999) in the ON status is reset.

- If two or more state relays are ON, the state relay with the smallest number is reset.
  When the command input is set to ON again, the state relay with the next smallest number is reset among state relays working as annunciators (S900 to S999) in the ON status.



Related devices

Device	Name	Description
M8049	Enable annunciator	When M8049 is set to ON, M8048 and D8049 are valid.
M8048	Annunciator ON	When M8049 is ON and either one among the state relays S900 to S999 is ON, M8048 turns ON.
D8049	Minimum state relay number in ON status	Among S900 to S999, the minimum number in the ON status is stored.

Caution

1. Execution in each operation cycle

- When ANR instruction is used, annunciators in the ON status are reset in turn in each operation cycle.
- When ANRP instruction is used, an annunciator in the ON status is reset only in one operation cycle (only once).

Program example

Refer to ANS (FNC 46) instruction.

→ **For a program example, refer to Section 12.7.**

12.9 FNC 48 – SQR / Square Root

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction obtains the square root.
The ESQR (FNC127) instruction obtains the square root in floating point operation.

→ **For ESQR (FNC127) instruction, refer to Section 18.15.**

1. Instruction format

	FNC 48	
D	SQR	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	SQR	Continuous Operation
	SQRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DSQR	Continuous Operation
	DSQRP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
(S·)	Word device number storing data whose square root is obtained	16- or 32-bit binary
(D·)	Data register number storing the square root operation result	16- or 32-bit binary

(S·): K0 to K32767 in 16-bit operation, K0 to K2,147,483,647 in 32-bit operation

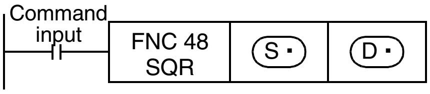
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)														✓	✓	✓			✓	✓	✓			
(D·)														✓	✓	✓			✓					

Explanation of function and operation

1. 16-bit operation (SQR and SQRP)

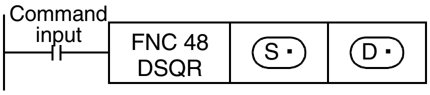
The square root of the data stored in (S·) is calculated, and stored to (D·).



$\sqrt{(S\cdot)} \rightarrow (D\cdot)$

2. 32-bit operation (DSQR and DSQRP)

The square root of the data stored in [(S·)+1, (S·)] is calculated, and stored to [(D·)+1, (D·)].



$\sqrt{(S\cdot)+1,(S\cdot)} \rightarrow (D\cdot)+1,(D\cdot)$

Caution

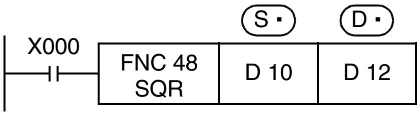
1. Operation result

1) The obtained square root is an integer because the decimal point is ignored.
   When the calculated value is ignored, M8021 (borrow flag) turns ON.
2) When the calculated value is true "0", M8020 (zero flag) turns ON.

11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FMC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

Program example

The square root of D10 is stored to D12.
The value of D10 is "100".



$$\sqrt{D\,10} \rightarrow D\,12$$

$$= \sqrt{100} \rightarrow 10$$

12.10 FNC 49 – FLT / Conversion to Floating Point

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction converts a binary integer into a binary floating point (real number).

1. Instruction format

D	FNC 49 FLT	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	FLT	Continuous Operation
	FLTP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DFLT	Continuous Operation
	DFLTP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S·	Data register number storing binary integer	16- or 32-bit binary
D·	Data register number storing binary floating point (real number)	16- or 32-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index				Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify		K	H	E	"□"	P
S·														✓	✓	✓			✓						
D·														✓	✓	✓			✓						

Explanation of function and operation

1. 16-bit operation (FLT and FLTP)

The binary integer data of S· is converted into binary floating point (real number), and stored to [D·+1, D·].

Command input —| |— [FNC 49 FLT | S· | D·]

S· → (D· + 1, D·)
Binary integer → Binary floating point (real number)

2. 32-bit operation (DFLT and DFLTP)

The binary integer data of [S·+1, S·] is converted into binary floating point (real number), and stored to [D·+1, D·].

Command input —| |— [FNC 49 DFLT | S· | D·]

(S·+ 1, S·) → (D· +1, D·)
Binary integer → Binary floating point (real number)

Related instruction

Instruction	Description
INT(FNC129)	It is inverse of FLT instruction, and converts binary floating point into binary integer.

Caution

1. It is not necessary to convert a constant (K or H) into floating point value.

The value of a K or H specified in each instruction for binary floating point (real number) operation is automatically converted into binary floating point (real number). It is not necessary to convert such a constant using by FLT instruction.
(K and H cannot be specified in RAD, DEG, EXP and LOGE instructions.)

11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FNC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

Related devices

→ For the method of the zero and borrow flags, refer to Subsection 6.5.2.

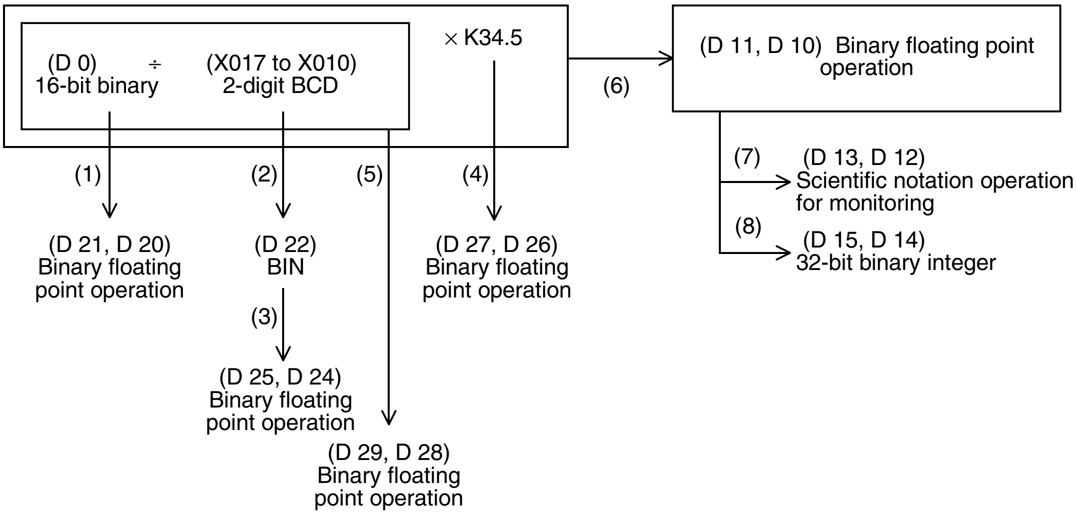
Device	Name	Description
M8020	Zero flag	Turns ON when the value is true "0".
M8021	Borrow flag	Turns ON when the floating point is rounded down.

Program example

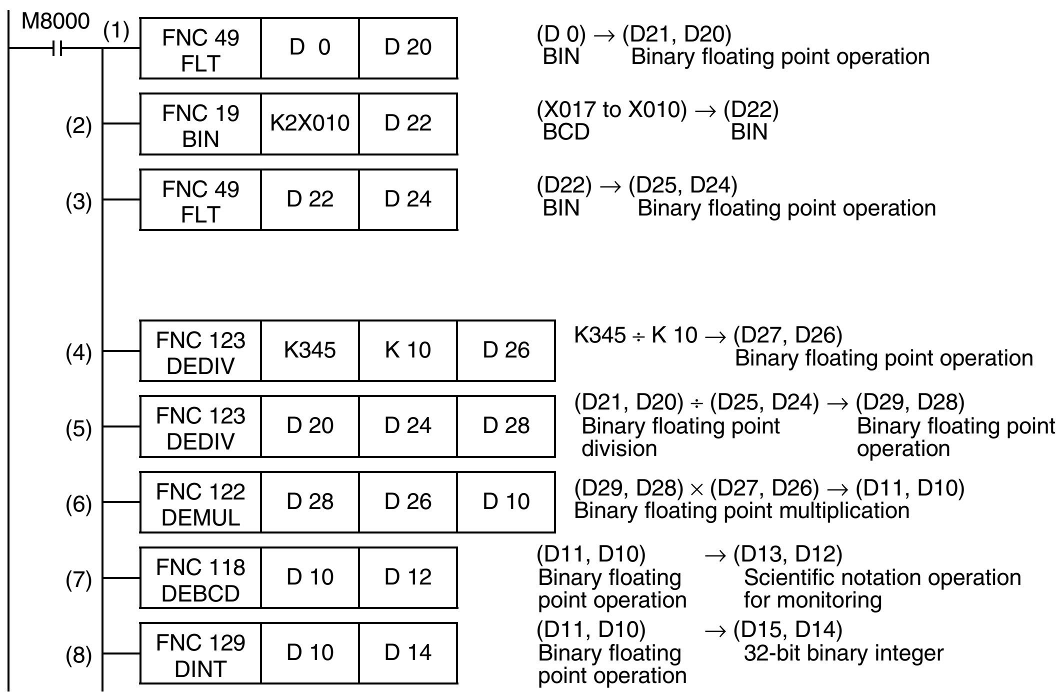
1. Arithmetic operations by binary floating point operations

The sequence program shown below is constructed as follows:

1) Calculation example



2) Sequence program



13. High Speed Processing – FNC 50 to FNC 59

FNC 50 to FNC 59 provide interrupt processing type high speed instructions that execute sequence control using the latest I/O information and utilize the high speed processing performance of the PLC.

FNC No.	Mnemonic	Symbol	Function	Reference
50	REF	REF D n	Refresh	Section 13.1
51	REFF	REFF n	Refresh and filter adjust	Section 13.2
52	MTR	MTR S D1 D2 n	Input Matrix	Section 13.3
53	HSCS	HSCS S1 S2 D	High speed counter set	Section 13.4
54	HSCR	HSCR S1 S2 D	High speed counter reset	Section 13.5
55	HSZ	HSZ S1 S2 S D	High speed counter zone compare	Section 13.6
56	SPD	SPD S1 S2 D	Speed Detection	Section 13.7
57	PLSY	PLSY S1 S2 D	Pulse Y Output	Section 13.8
58	PWM	PWM S1 S2 D	Pulse Width Modulation	Section 13.9
59	PLSR	PLSR S1 S2 S3 D	Acceleration/deceleration setup	Section 13.10

11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FMC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

13.1 FNC 50 – REF / Refresh

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction immediately outputs the latest input (X) information or the current output (Y) operation result in the middle of a sequence program.

1. Instruction format

FNC 50
REF P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	REF	Continuous Operation
	REFP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand Type	Description	Data Type
D	Head bit device (X or Y) number to be refreshed	Bit
n	Number of bit devices to be refreshed (multiple of 8 in the range from 8 to 256)	16-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
D	▲1	▲2																						
n																				▲3	▲3			

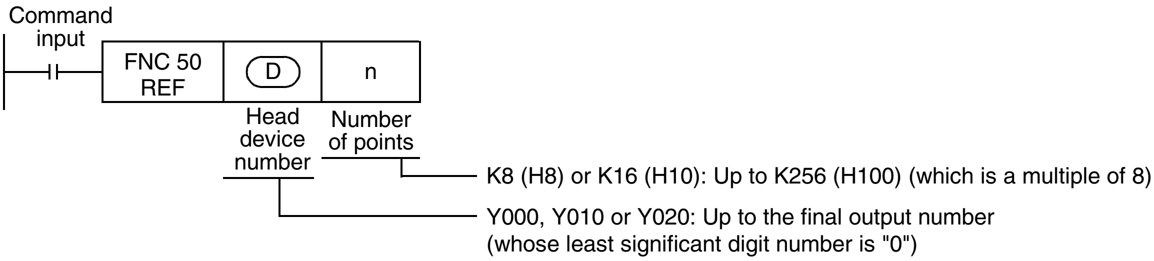
▲1: X000, X010 or X020: Up to the final input number (whose least significant digit number is "0")
▲2: Y000, Y010 or Y020: Up to the final output number (whose least significant digit number is "0")
▲3: K8 (H8) or K16 (H10): Up to K256 (H100) (which is a multiple of 8)

Explanation of function and operation

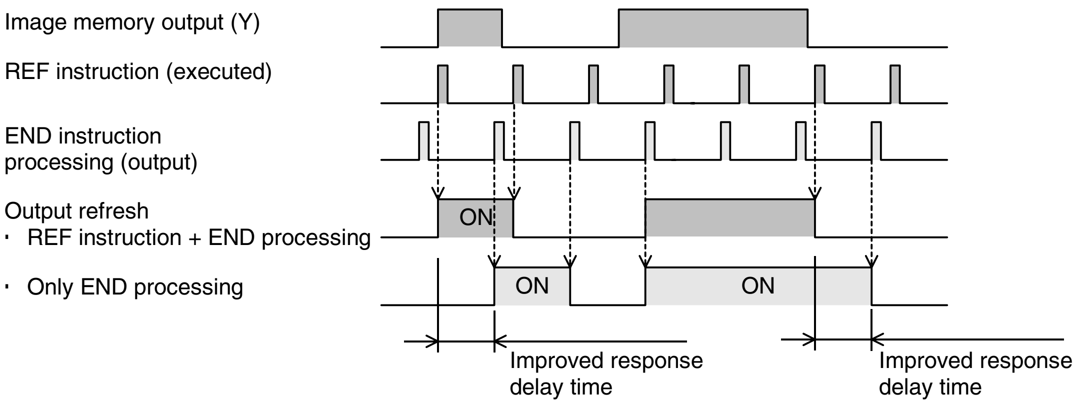
1. 16-bit operation (REF and REFP)

1) When refreshing outputs (Y)

"n" points are refreshed from the specified output device (D). ("n" is a multiple of 8.)

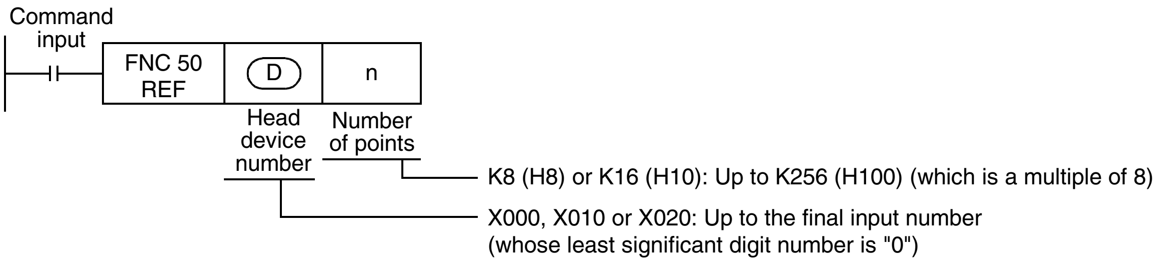


- When this instruction is executed, the output latch memory is refreshed to the output status in the specified range.

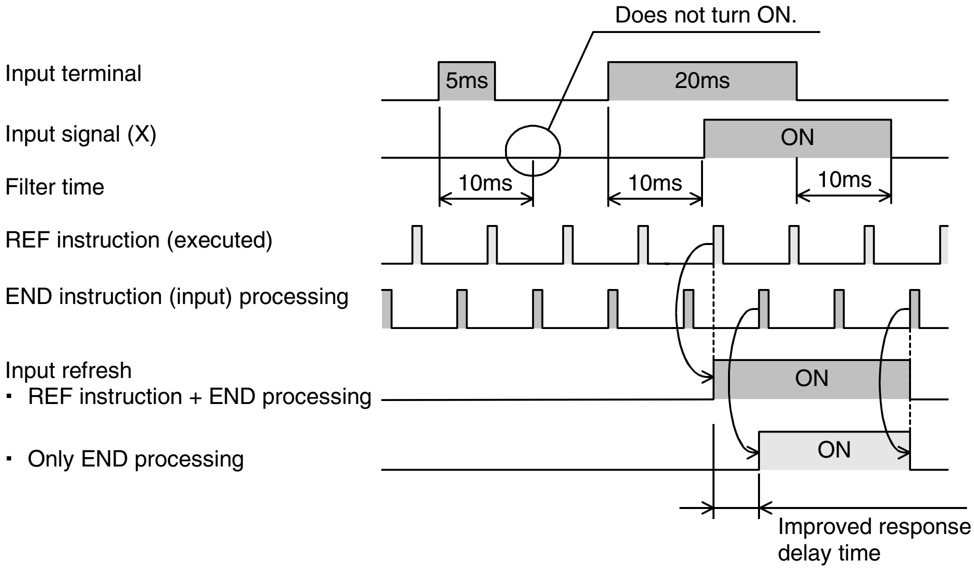


2) When refreshing inputs (X)

"n" points are refreshed from the specified input device (D). ("n" is a multiple of 8.)



- If the input information is turned ON approximately 10 ms (response delay time of the input filter) before the instruction is executed, the input image memory turns ON when the instruction is executed.
- In X000 to X017*1, the response delay time of the input filter can be changed.
  *1 X000 to X007 in the FX3U-16M□
  → **For details, refer to "13.1.1 What should be understood before using REF instruction" later.**



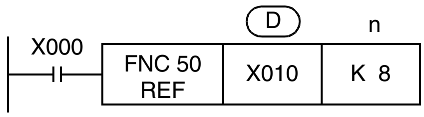
Cautions

1. Setting the number of refreshed points "n"
Set a multiple of 8 such as "K8 (H8), K16 (H10) … K256 (H100)". Any other numeric value causes an error.

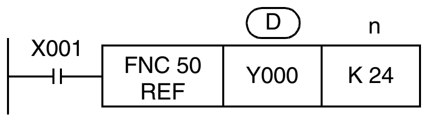
2. Setting the head device number (D)
Make sure that the least significant digit number is "0" such as "X000, X010, X020 …"or " Y000, Y010, Y020 …"

Program examples

1. When refreshing inputs
Only X010 to X017 (8 points in total) are refreshed.



2. When refreshing outputs
Y000 to Y007, Y010 to Y017 and Y020 to Y027 (24 points in total) are refreshed.



13.1.1 What should be understood before using REF instruction

1. Changing the input filter

The input filter value is determined by the contents of D8020 (initial value: 10 ms).
When setting a smaller value to the input filter, change the contents of D8020 by MOV instruction, etc.

Target range: X000 to X017 (In inputs X020 and later, the input filter value is fixed to 10 ms and cannot be changed.)
The above range changes according to the main unit and may be reduced depending on the number of built-in inputs.
(The target range is X000 to X007 in the FX3U-16M□.)

2. Output response time

After REF instruction is executed, the output (Y) sets the output signal to ON after the response time shown below.

→ **For details, refer to the Hardware Edition of the main unit.**

Target range: Y000 to final connected output number

1) In the case of relay output type
   The output contact is activated after the response time of the output relay.
   - Y000 and later: Approximately 10 ms
2) In the case of transistor output type
   - Y000, Y001, Y002 and Y003: 5 μs or less (load current = 10 mA or more, 5 to 24V DC)
   - Y004 and later: 0.2 ms or less (load current = 100 mA, 24V DC)

3. When using REF instruction between FOR and NEXT instructions or between a label (in a step of a smaller number) and CJ instruction (in a step of a larger number)

When the input information or immediate output is required even in the middle of a routine program during control, inputs or outputs can be refreshed inside the routine program.

4. When using the input interrupt (I) function

When executing the interrupt processing accompanied by I/O operations, I/O refresh can be executed in the interrupt routine to receive the latest input (X) information and give the immediate output (Y) of the operation result so that dispersion caused by the operation time is improved.

13.2 FNC 51 – REFF / Refresh and Filter Adjust

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

The input filter of the inputs X000 to X017*1 is the digital type, and its filter time can be changed using this instruction or D8020.

By this instruction, the information of inputs X000 to X017*1 can be received in an arbitrary step in the program at the specified input filter time, and then transferred to the image memory.

1. Instruction format

FNC 51 REFF P

16-bit Instruction	Mnemonic	Operation Condition
3 steps	REFF	Continuous Operation
	REFFP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand Type	Description	Data Type
n	Time data of digital filter [K0 to K60 (H0 to H3C) × 1 ms]	16-bit binary

3. Applicable devices

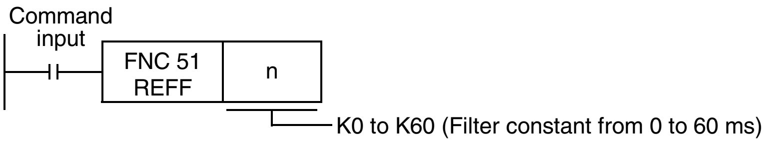
Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
n														✓	✓					▲	▲			

▲: K0 (H0) to K60 (H3C)

Explanation of function and operation

1. 16-bit operation (REFF and REFFP)

16 inputs from X000 to X017*1 in the image memory are refreshed at the input digital filter time [n × 1 ms].



- When the input turns ON "n × 1 ms" before the instruction is executed, the image memory is set to ON. When the input turns OFF "n × 1 ms" before the instruction is executed, the image memory is set to OFF.
- When the command input is ON, REFF instruction is executed in each operation cycle.
- When the command input is OFF, REFF instruction is not executed, and the input filter of X000 to X017*1 is set to the set value of D8020 (which is the value in the input processing).

*1. X000 to X007 in the FX3U-16M□

Cautions

1. Setting the filter time "n"

Set "n" within the range from K0 (H0) to K60 (H3C) [0 to 60 ms].

2. Function of the input filter

A digital filter is built in the inputs X000 to X017*1. The filter time can be changed in units of 1 ms within the range from 0 to 60 ms using an applied instruction. When the filter time is set to "0", the values shown in the table below result.

Input number	Input filter value when set to "0"
X000 to X005	5 μs*2
X006, X007	50 μs
X010 to X017*3	200 μs*3

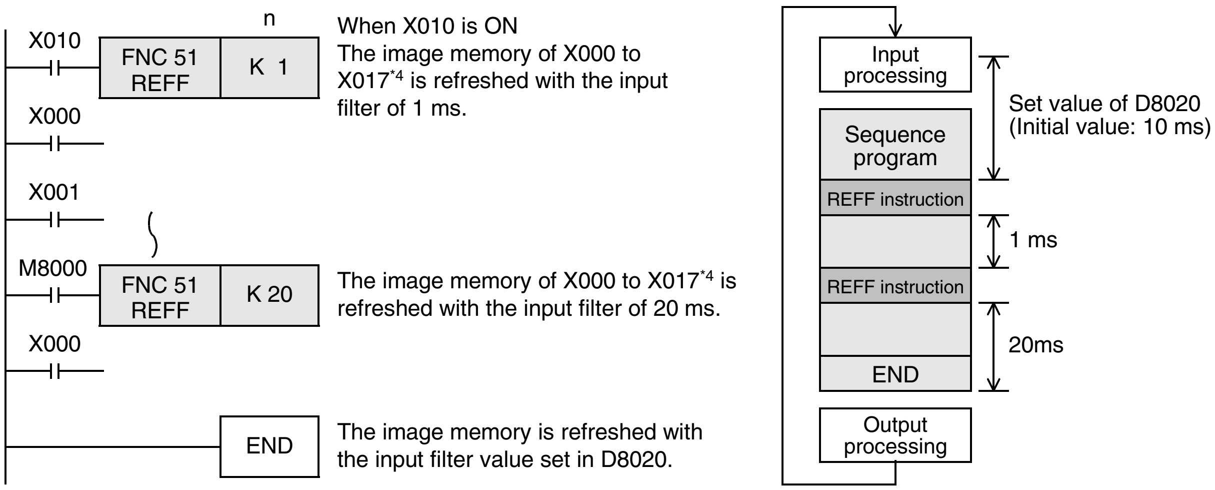
*1. X000 to X007 in the FX3U-16M□

*2. When setting the input filter time to "5 μs", perform the following:
- Make sure that the wiring length is 5 m or less.
- Connect a bleeder resistor of 1.5 kΩ (1 W or more) to the input terminal, and make sure that the load current in the open collector transistor output in the counterpart equipment is 20 mA or more including the input current in the main unit.

*3. The filter time is fixed to 10 ms in X010 to X017 when the FX3U-16M□ is used.

Program example

1. Relationship between the program and the filter time



*4. X000 to X007 in the FX3U-16M□

13.2.1 What should be understood before using REFF instruction

Generally, a C-R filter of approximately 10 ms is provided for inputs in PLCs as countermeasures against chattering and noise at the input contacts.

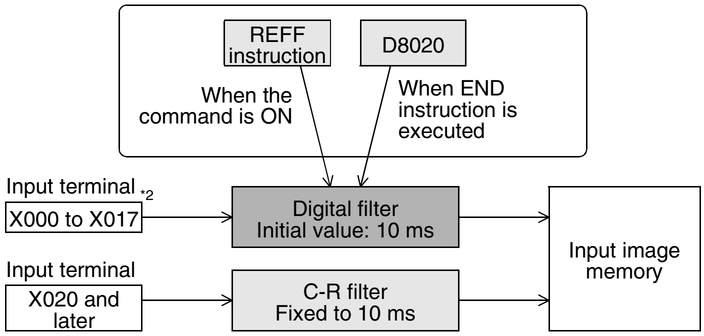
A digital filter is provided for the inputs X000 to X017*1 in FX3UC PLCs. The digital filter value can be changed within the range from 0 to 60 ms through using an instruction.

*1. X000 to X007 in the FX3U-16M□

1. How to change the digital filter (executing END instruction)

The input filter initial value (10 ms) of X000 to X017*2 is set in the special data register D8020.

Accordingly, by changing this value using MOV instruction, etc., the input filter value of X000 to X017*2 which is used during execution of END instruction can be changed.



*2. X000 to X007 in the FX3U-16M□

2. Instruction in which the digital filter is automatically changed

Without regard to the change of the filter time executed by the REFF instruction, when the following functions and instruction are executed, the input filter value is automatically changed (to 5 μs in X000 to X005 and 50 μs in X006 and X007).

However, if the digital filter is used for any purpose other than the following functions and instruction in a general program, the digital filter is set to the time set in D8020. As a result, the program is not executed correctly if the ON duration or OFF duration of the corresponding input signal is less than the input filter time.

- Input of interrupt pointer specified in the input interrupt function
- Input used in high speed counter
- Input used in SPD (FNC 56) instruction

13.3 FNC 52 – MTR / Input Matrix

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction reads matrix input as 8-point input × "n"-point output (transistor) in the time division method.

1. Instruction format

FNC 52 MTR

16-bit Instruction	Mnemonic	Operation Condition
9 steps	MTR	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	

2. Set data

Operand Type	Description	Data Type
S	Input device (X) number of matrix signal input X000, X010, X020 … final input device number (Only "0" is allowed in the least significant digit of device numbers.)	Bit
D1	Head device (Y) number of matrix signal output Y000, Y010, Y020 … final output device number (Only "0" is allowed in the least significant digit of device numbers.)	Bit
D2	Head bit device (Y, M or S) number of ON output destination Y000, Y010, Y020 … final Y number, M000, M010, M020 … final M number or S000, S010, S020 … final S number (Only "0" is allowed in the least significant digit of device numbers.)	Bit
n	Number of columns in matrix input (K2 to K8 or H2 to H8)	16-bit binary

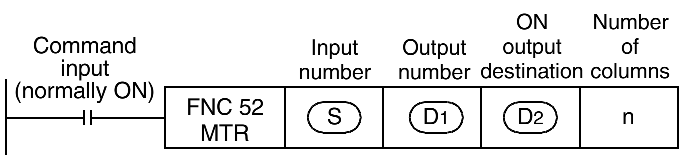
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S	✓																							
D1		✓																						
D2		✓	✓			✓																		
n																				✓	✓			

Explanation of function and operation

1. 16-bit operation (MTR)

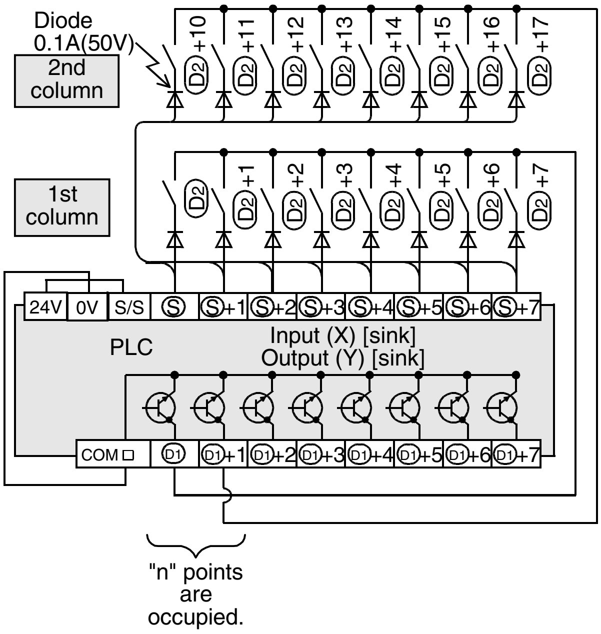
An input signal of 8 points × "n" columns is controlled in the time division method using 8 inputs (S) and "n" transistor outputs (D1). Each column is read in turn, and then output to (D2).

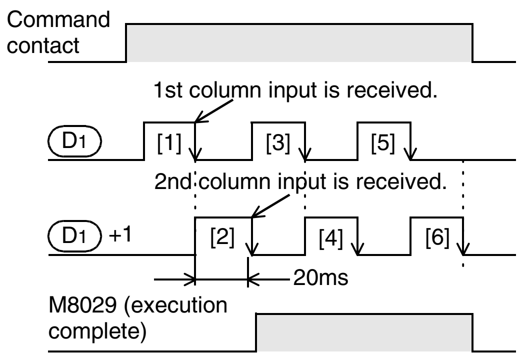


- For each output, the I/O processing is executed immediately in turn in interrupt at every 20 ms under consideration of the input filter response delay of 10 ms.

The figure below shows an example of the FX3U series main unit (sink input/sink output). For the wiring, refer to the following manual of the used PLC.

→ FX3U Hardware Edition





Related device

Device	Name	Description
M8029	Instruction execution complete	Turns ON after the first cycle operation.

Cautions

1. Number of occupied devices

1) Eight input points are occupied from the input device number specified in S.
2) "n" output points are occupied from the output device number specified in D1.
   When specifying the output in D2, make sure that "n" output numbers specified in D1 does not overlap the output specified in D2.

2. Wiring

One diode of 0.1 A/50 V is required for each switch.

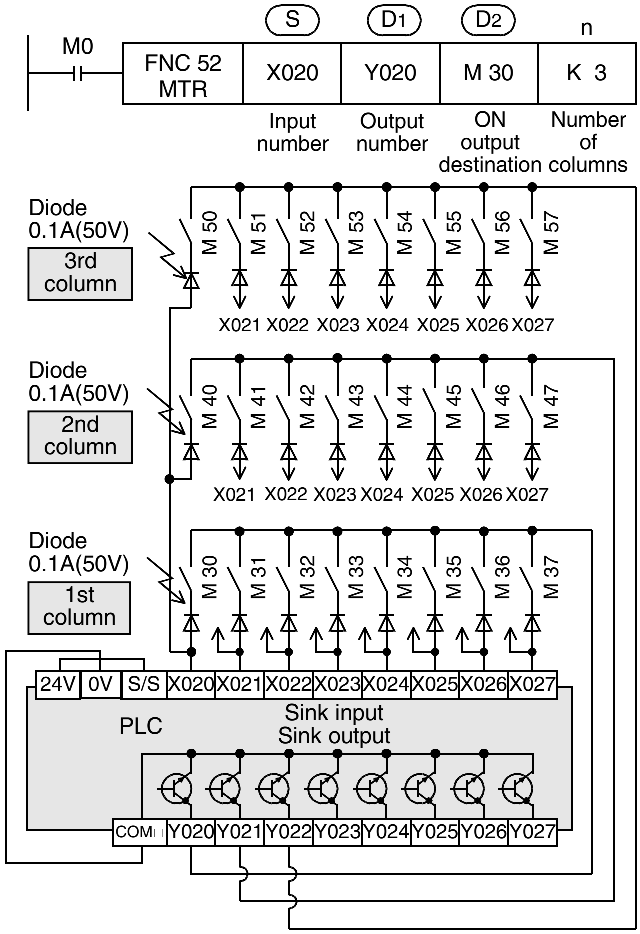
3. Output format

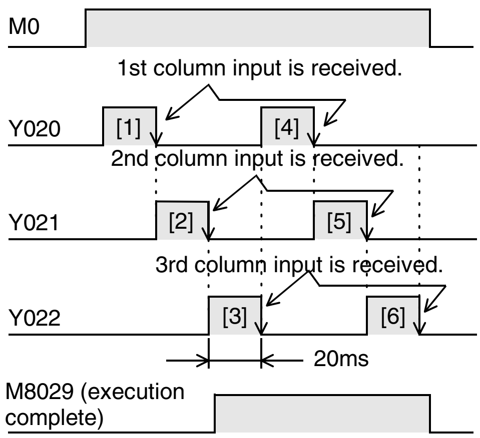
Use the transistor output format.

Program example

n=Three outputs (Y020, Y021 and Y022) are set to ON in turn repeatedly.
Every time an output is set to ON, eight inputs in the 1st, 2nd and 3rd columns are received in turn repeatedly, and stored to M30 to M37, M40 to M47 and M50 to M57 respectively.
In this program example, the FX3U series main unit (sink input/sink output) is used. For the wiring, refer to the following manual of the used PLC.

→ **FX3U Hardware Edition**

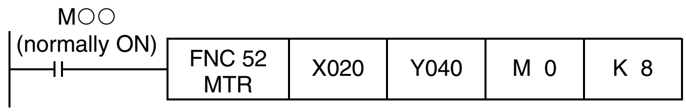




13.3.1 The method and cautions for MTR instruction

1. Command input

1) Setting the command input to normally ON
For MTR instruction, set the command input to normally ON.



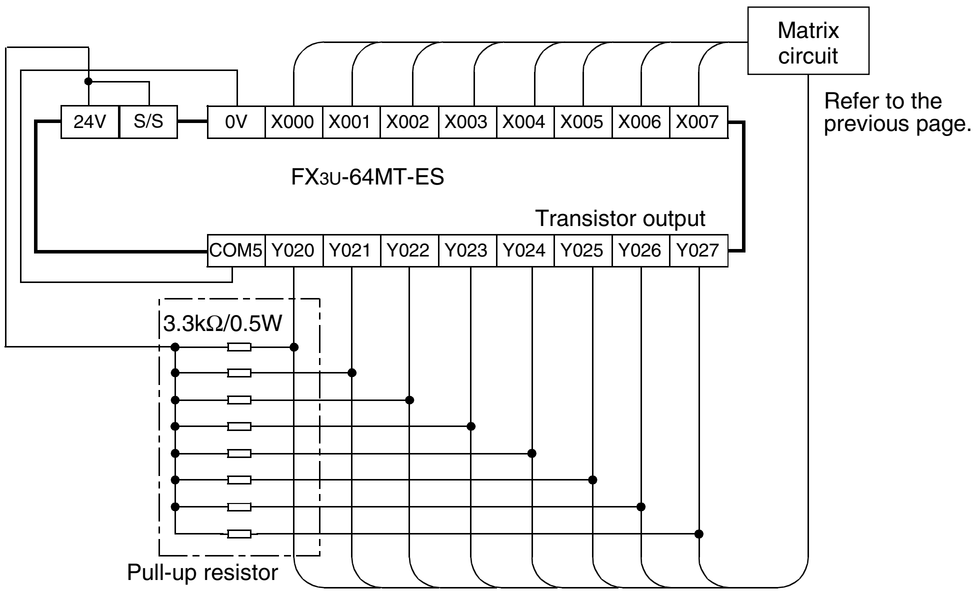
2. Input numbers used in MTR instruction

1) Inputs available in MTR instruction
Use the inputs X020 and later under normal conditions.

2) When using the inputs X000 to X017
The receiving speed is higher. Because the output transistor recovery time is long and the input sensitivity is high, however, erroneous inputs may occur.
To prevent erroneous inputs, connect pull-up resistors (3.3 kΩ/0.5 W) to transistor outputs used in MTR instruction.
For pull-up resistors, use the power supply shown in the table below.

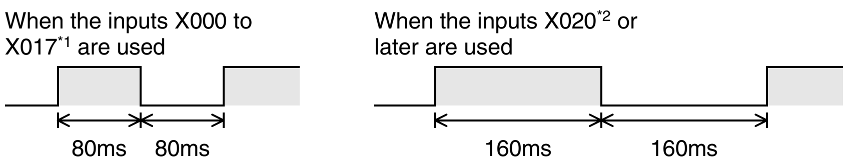
	Power supply used for pull-up resistors
AC power type PLC	Service power supply
DC power type PLC	Power supply for driving PLC

The figure below shows an example of the FX3U Series main unit (sink input/sink output).



3. ON/OFF duration of input signals

Because input of 64 points (8 rows × 8 columns) is received in a cycle of 80 or 160 ms, the ON/OFF duration of each input signal should be equivalent to or more than the value shown below:



*1. X000 to X007 in the FX3U-16M□

*2. X010 or later in the FX3U-16M□

13.4 FNC 53 – HSCS / High Speed Counter Set

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction compares a value counted by a high speed counter with a specified value, and immediately sets an external output (Y) if the two values are equivalent each other.

→ **For the counter interrupt using HSCS instruction, refer to Section 35.6.**

1. Instruction format

D FNC 53 HSCS	16-bit Instruction	Mnemonic	Operation Condition	–
	32-bit Instruction: 13 steps	Mnemonic: DHSCS	Operation Condition: Continuous Operation	

2. Set data

Operand Type	Description	Data Type
S1•	Data to be compared with the current data value of a high-speed counter or word device number.	32-bit binary
S2•	Device number of a high speed counter [C235 to C255]	32-bit binary
D•	Bit device number to be set to ON when the compared two values are equivalent to each other	Bit

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
S2•													✓						✓					
D•		✓	✓			✓	▲1												✓					▲2

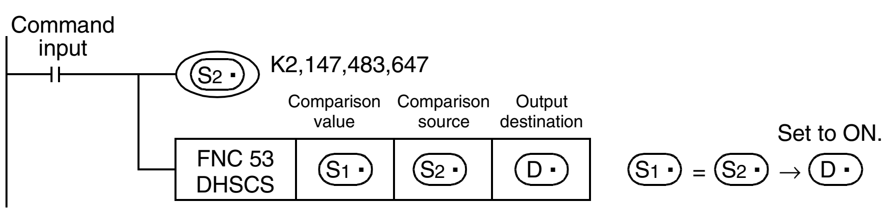
▲1: "D□.b" cannot be indexed with index registers (V and Z).
▲2: When using the counter interrupt function, specify an interrupt pointer.

→ **For counter interrupt using HSCS instruction, refer to Section 35.6.**

Explanation of function and operation

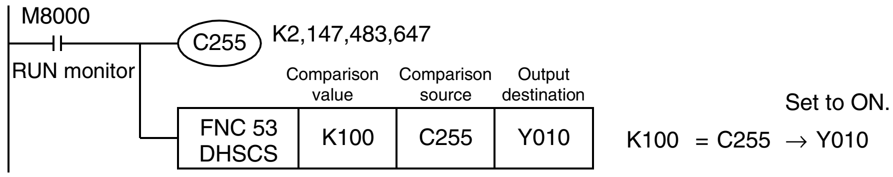
1. 32-bit operation (DHSCS)

When the current value of a high speed counter (C235 to C255) specified in S2• becomes the comparison value [S1•+1, S1•] (for example, when the current value changes from "199" to "200" or from "201" to "200" if the comparison value is K200), the bit device D• is set to ON without regard to the operation cycle. This instruction is executed after the counting processing in the high speed counter.



Operation

When the current value of the high speed counter C255 changes from "99" to "100" or from "101" to "100", Y010 is set to ON (output refresh).



Related instructions

The following instructions can be combined with high speed counters:

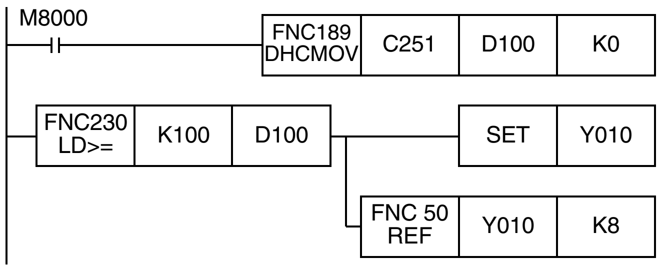
Instruction	FNC No.	Instruction name
DHSCS	FNC 53	High speed counter set
DHSCR	FNC 54	High speed counter reset
DHSZ	FNC 55	High speed counter zone compare
DHCMOV	FNC189	High speed counter move
DHSCT	FNC280	High speed counter compare with data table

Cautions

1. Selection of the count comparison method

When HSCS instruction is used, hardware counters (C235, C236, C237, C238, C239, C240, C244 (OP), C245 (OP), C246, C248 (OP), C251 and C253) are automatically switched to software counters, and the maximum frequency of each counter and the total frequency are affected.
Refer to the counting operation described below, and select according to the contents of control whether to use HSCS instruction or general-purpose comparison instruction.

1) Case to select HSCS instruction
   - When the output should be given when the counting result becomes equivalent to the comparison value without regard to the scan time of the PLC
2) Cases to select a general-purpose comparison instruction
   - When the required frequency is beyond the counting performance of the software counters
   - When counting is regarded as important, but the effect of the scan time can be ignored in operations according to the counting result
   - When the number of an instruction is more than 32



2. Device specification range

Only high speed counters (C235 to C255) can be specified as (S·).

3. Only 32-bit operation instructions are available.

Because instructions for high speed counters are dedicated to 32 bits, make sure to input "DHSCS (FNC 53)".

4. Priority order in operation among HSCS (FNC 53), HSCR (FNC 54), and HSZ (FNC 55) instructions for a same high speed counter

→ For details, refer to "6. Priority order in operations among HSCS (FNC 53), HSCR (FNC 54), and HSZ (FNC 55) instructions for the same high speed counter" in Subsection 13.4.1.

5. **Reset operation by an external terminal**

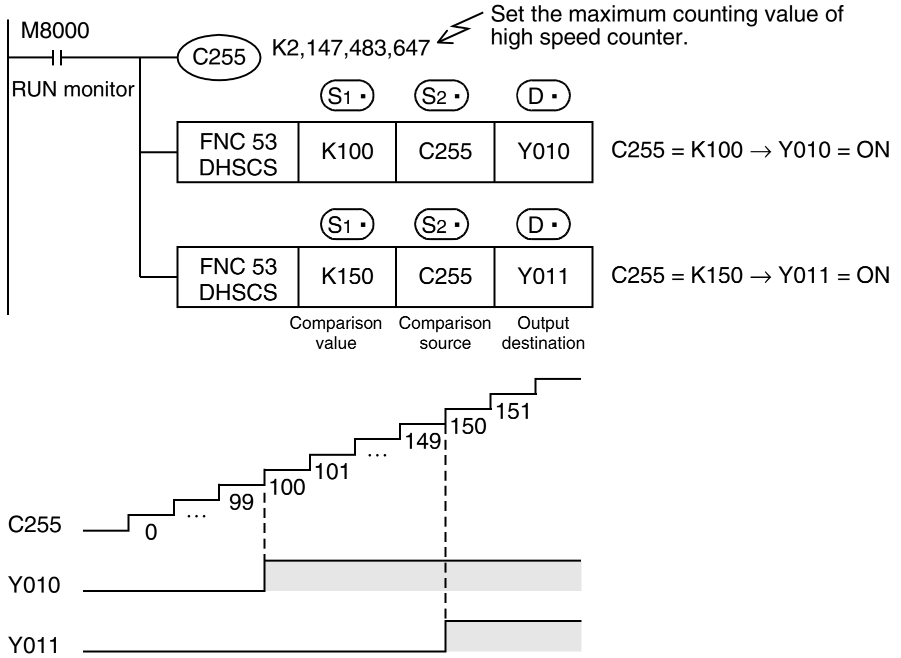
→ **For details, refer to "5. Reset operation by an external terminal" in Subsection 13.4.1.**

6. **For other cautions on using HSCS instruction, refer to the description later.**

→ **For details, refer to the next page.**

Program example

With regard to the current value of a counter, different outputs (Y) are arbitrary set to ON by two values.



13.4.1 Common cautions on using instructions for high speed counter

DHSCS (FNC 53), DHSCR (FNC 54), DHSZ (FNC 55) and DHSCT (FNC280) instructions are provided for high speed counters.
This section explains common cautions for these instructions.

1. **Limitation in the number of an instruction in a program**

DHSCS, DHSCR and DHSZ instructions can be used as many times as necessary in the same way as general instructions. However, the number of simultaneously driven instructions is limited. The DHSCT instruction can be used only once in any program.

Instruction	Limitation in number of instructions driven at same time
DHSCS	32 instructions including DHSCT instruction
DHSCR	
DHSZ*1	
DHSCT*1	Only 1 (This instruction can be used only once.)

*1. When DHSZ or DHSCT instruction is used, the maximum response frequency of every software counter and the total frequency are limited.

2. **Response frequency of high speed counters**

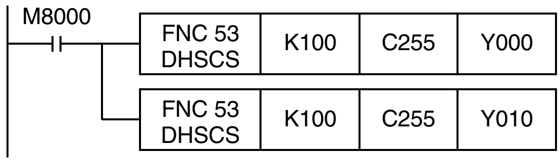
When DHSZ or DHSCT instruction is used, the maximum response frequency of every software counter and the total frequency are limited.

→ **For the maximum response frequency of each software counter and the total frequency, refer to Subsection 4.7.10.**

3. Specification of output numbers (Y)

When using the same instruction for high speed counter repeatedly or when driving two or more other instructions for high speed counter at the same time, specify such output devices (Y) whose high-order two digits are the same (in units of 8 devices).

1) When using devices of the same number (in units of 8 devices)
   Example: When using Y000, specify Y000 to Y007. When using Y010, specify Y010 to Y017.
2) When using two or more instructions for high speed counter and non-consecutive output (Y) numbers
   A program example is shown below:



When C255 reaches K100, the output Y000 is driven by interrupt. Y010 is driven when END processing is executed.
If interrupt drive is required, use an output number in the range from Y001 to Y007 whose high-order two digits are equivalent.

4. Caution on the counting operation when the current value is changed

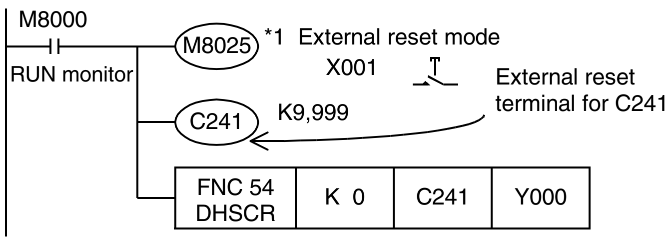
An instruction for the high speed counter gives the comparison result when a pulse is input to the input (X) of the high speed counter.
However, the comparison result is not given when the current value of the high speed counter is changed in the following method:

1) Change method (examples)
   a) Overwriting the contents of a word device used as the comparison value using DMOV instruction, etc.
   b) Resetting the current value of a high speed counter in a program
2) Operation
   Even if the condition for setting the output to ON or OFF is given as the comparison result, the comparison result does not change when an instruction is simply driven.

5. Reset operation by an external terminal [M8025[*1]: HSC (external reset) mode]

For a high speed counter equipped with an external reset terminal (R) such as C241, an instruction is executed and the comparison result is output at the rising edge of the reset input signal.

1) Program
   If an instruction for the high speed counter is used while M8025[*1] is driven, the instruction is executed again when the current value of the high speed counter C241 is cleared by an external reset terminal. And the comparison result is output even if a counting input is not given.

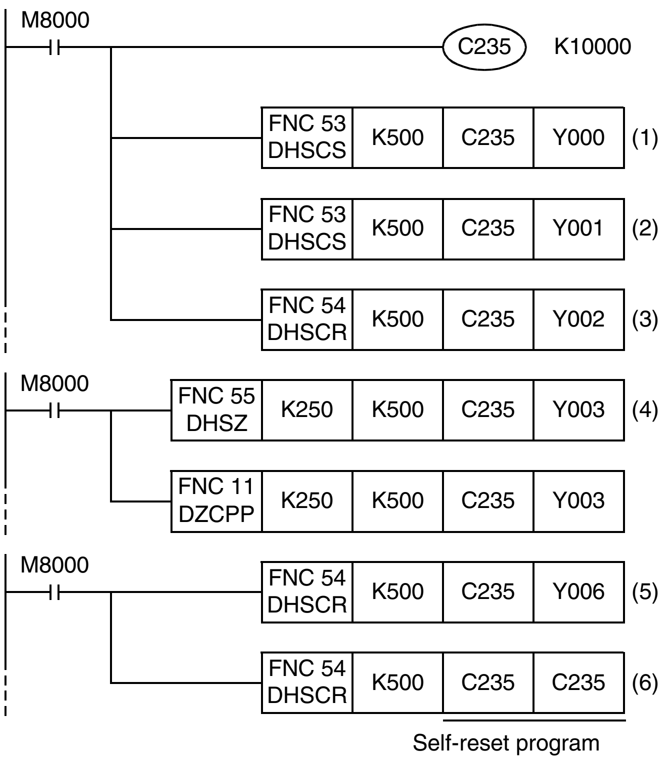


*1. M8025 is cleared when the PLC mode is changed from RUN to STOP.

2) Operation
   When the external reset input X001 turns ON while the current value of C241 is "100", for example, the current value of C241 is reset to "0". And Y000 is reset at this time even if a counting input is not given.

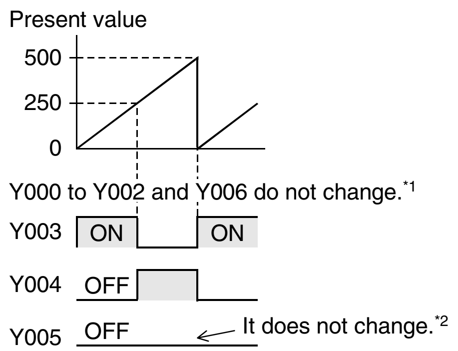
6. Priority order in operations among HSCS (FNC 53), HSCR (FNC 54), and HSZ (FNC 55) instructions for the same high speed counter

When the same comparison value is used for the same high speed counter in HSCS (FNC 53), HSCR (FNC 54), and HSZ (FNC 55) instructions, high speed counter reset (self-reset) by HSCR (FNC 54) instruction is executed with the highest priority (as shown in the table below).
In this case, the comparison results do not change in HSCS, HSCR, and HSZ instructions whose comparison value is programmed to be the same as the comparison value for self-reset by HSCR (FNC 54) instruction.
To change the comparison results, set the comparison value to "K0".



Program sequence	Processing sequence		
	FX3U/FX3UC	FX2N/FX2NC	FX1N/FX1S/FX1NC
DHSCS (1)	DHSCR (6) (self-reset)	DHSCS (1)	DHSCS (1)
DHSCS (2)	DHSZ (4)	DHSCS (2)	DHSCS (2)
DHSCR (3)	DHSCS (1)	DHSCR (3)	DHSCR (3)
DHSZ (4)	DHSCS (2)	DHSZ (4)	DHSZ (4)
DHSCR (5)	DHSCR (3)	DHSCR (5)	DHSCR (5)
DHSCR (6) (self-reset)	DHSCR (5)	DHSCR (6) (self-reset)	DHSCR (6) (self-reset)

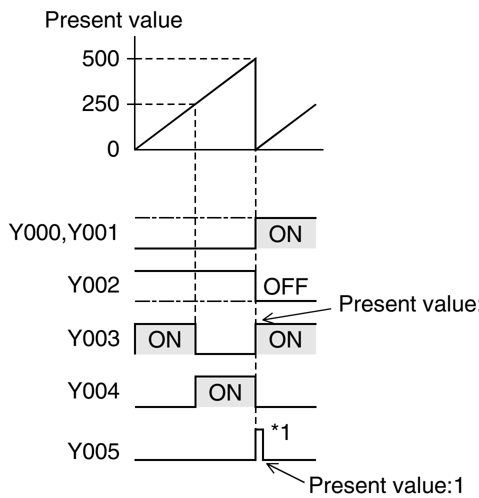
Operation of FX3U/FX3UC PLC



*1. To change the comparison results by the instructions (1) to (3) and (5), change the comparison value "K500" in the instructions (1) to (3) and (5) to "K0".

*2. To set Y005 to ON in the HSZ instruction (4), set a value smaller than the comparison value "K500". However, due to the response delay at the output, the output may not operate within the short time before the counter's present value is reset to "0".

Operation of FX2N/FX2NC/FX1N/FX1NC/FX1S PLC [reference]



*1. Due to the response delay at the output, the output may not operate within the short time before the counter's present value is reset to "0".

13.5 FNC 54 – HSCR / High Speed Counter Reset

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction compares the value counted by a high speed counter with a specified value at each count, and immediately resets an external output (Y) when both values become equivalent to each other.

1. Instruction format

D FNC 54 HSCR

16-bit Instruction	Mnemonic	Operation Condition
	–	

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DHSCR	Continuous Operation

2. Set data

Operand Type	Description	Data Type
(S1•)	Data to be compared with the current value of a high speed counter or word device number storing the data to be compared	32-bit binary
(S2•)	Device number of a high speed counter [C235 to C255]	32-bit binary
(D•)	Bit device number to be reset (set to OFF) when both values become equivalent each other.	Bit

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S1•)								✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
(S2•)													✓						✓					
(D•)		✓	✓			✓	▲1						▲2						✓					

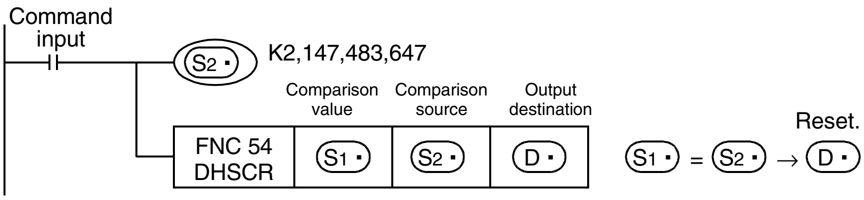
▲1: "D□.b" cannot be indexed with index registers (V and Z).

▲2: The same counter as (S2•) can be specified also. (Refer to the program example shown later.)

Explanation of function and operation

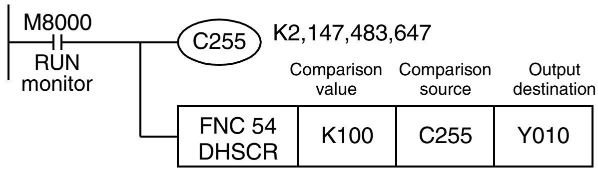
1. 32-bit operation (DHSCR)

When the current value of a high speed counter (C235 to C255) specified in (S2•) becomes the comparison value [(S1•)+1, (S1•)] (for example, when the current value changes from "199" to "200" or from "201" to "200" if the comparison value is K200), the bit device (D•) is reset (set to OFF) regardless of the operation cycle. In this instruction, the comparison processing is executed after the counting processing in the high speed counter.



Operation

When the present value of the high speed counter C255 changes (counts) from "99" to "100" or from "101" to "100", Y010 is reset (output refresh).



Reset.
K100 = C255 → Y010

Related instructions

The following instructions can be combined with high speed counters:

Instruction	FNC No.	Instruction name
DHSCS	FNC 53	High speed counter set
DHSCR	FNC 54	High speed counter reset
DHSZ	FNC 55	High speed counter zone compare
DHCMOV	FNC189	High speed counter move
DHSCT	FNC280	High speed counter compare with data table

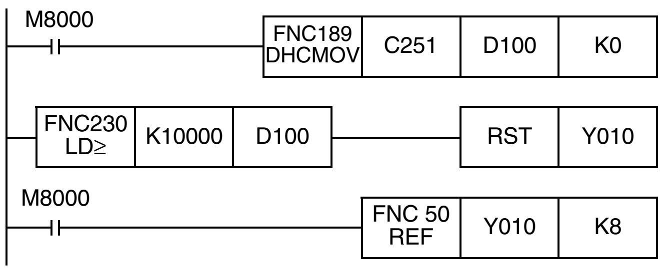
Cautions

1. Selection of the count comparison method

When HSCR instruction is used, hardware counters (C235, C236, C237, C238, C239, C240, C244 (OP), C245 (OP), C246, C248 (OP), C251 and C253) are automatically switched to software counters, and the maximum frequency of each counter and the total frequency are affected.
Refer to the counting operation described below, and select according to the contents of control whether to use HSCR instruction or general-purpose comparison instruction.

1) Case to select HSCR instruction
   - When the output should be given when the counting result becomes equivalent to the comparison value regardless of the scan time of the PLC

2) Cases to select a general-purpose comparison instruction
   - When the required frequency is beyond the counting performance of software counters
   - When counting is important, but the effect of the scan time can be ignored in operations depending on the counting result
   - When the number in an instruction is more than 32 bits



2. Only 32-bit operation instructions are available.

Because instructions for high speed counters are dedicated to 32 bits, make sure to input "DHSCR (FNC 54)".

3. Priority order in operation among HSCS (FNC 53), HSCR (FNC 54), and HSZ (FNC 55) instructions for the same high speed counter

→ For details, refer to "6. Priority order in operations among HSCS (FNC 53), HSCR (FNC 54), and HSZ (FNC 55) instructions for the same high speed counter" in Subsection 13.4.1.

4. Reset operation by an external terminal

→ For details, refer to "5. Reset operation by an external terminal [M8025*1: HSC (external reset) mode]" in Subsection 13.4.1.

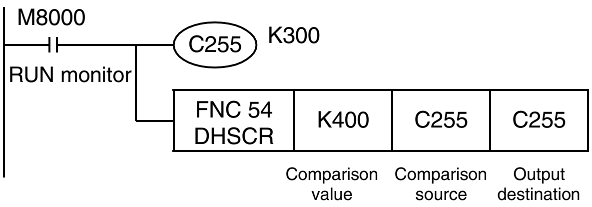
5. Other cautions on using HSCR instruction

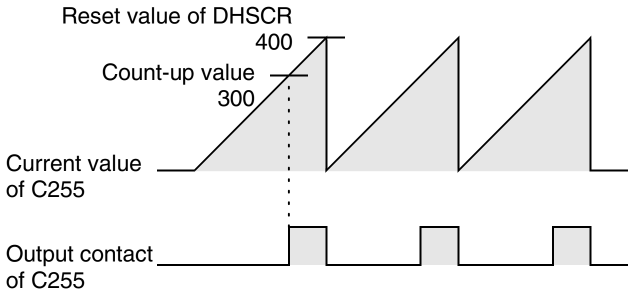
→ **For details, refer Subsection 13.4.1.**

Program example

1. Example of self-reset circuit

When the current value of C255 becomes "400", C255 is immediately reset. Its current value becomes "0", and the output contact is set to OFF.





13.6 FNC 55 – HSZ / High Speed Counter Zone Compare

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction compares the current value of a high speed counter with two values (one zone), and outputs the comparison result to three bit devices (refresh).

→ For the table high speed comparison mode, refer to Subsection 13.6.2.
→ For the frequency control mode, refer to Subsection 13.6.3.

1. Instruction format

D FNC 55 HSZ

16-bit Instruction	Mnemonic	Operation Condition
–		

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DHSZ	Continuous Operation

2. Set data

Operand Type	Description	Data Type
S1•	Data to be compared with the current value of a high speed counter or word device number storing data to be compared (comparison value 1)	32-bit binary
S2•	Data to be compared with the current value of a high speed counter or word device number storing data to be compared (comparison value 2)	32-bit binary
S•	Device number of a high speed counter [C235 to C255]	32-bit binary
D•	Head bit device number to which the comparison result is output based on upper and lower comparison values	Bit

3. Applicable devices

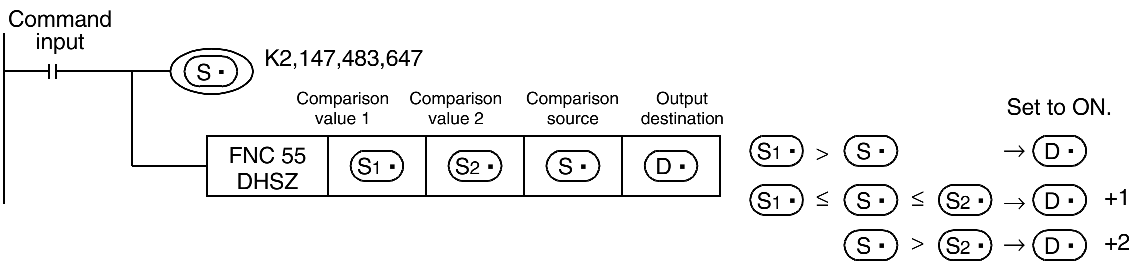
Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
S•													✓						✓					
D•		✓	✓			✓	▲												✓					

▲: "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

1. 32-bit operation (DHSZ)

The current value of a high speed counter (C235 to C255) specified in S• is compared with two comparison points (comparison value 1 and comparison value 2). Based on the comparison result, "smaller than the lower comparison value", "inside the comparison zone" or "larger than the upper comparison value", one among D•, D•+1 and D•+2 is set to ON regardless of the operation cycle.
In this instruction, the comparison processing is executed after the count processing in the high speed counter.



Comparison points

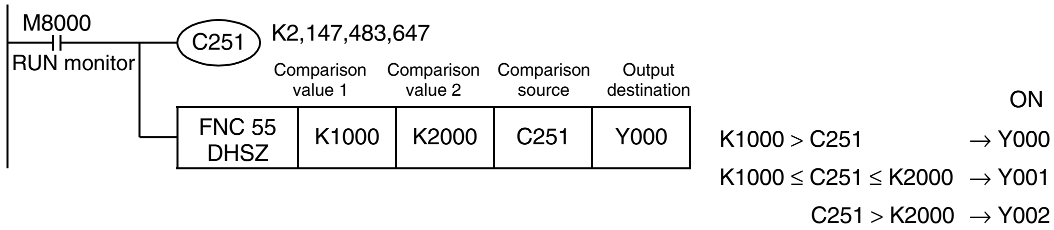
Make sure that the comparison value 1 and the comparison value 2 have the following relationship:

[S1• +1, S1•] ≤ [S2• +1, S2•]

Comparison point	Contents (32 bits)
Comparison value 1	S1• +1, S1•
Comparison value 2	S2• +1, S2•

Operation

When the current value of the high speed counter C251 changes (counts) as shown below, the comparison result is output to one of the outputs Y000, Y001 or Y002.



Comparison pattern	Current value of C251	Change of output contact (Y)		
		Y000	Y001	Y002
S1• > S•	1000 > S•	ON	OFF	OFF
	999 → 1000	ON → OFF	OFF → ON	OFF
	999 ← 1000	OFF → ON	ON → OFF	OFF
S1• ≤ S• ≤ S2•	999 → 1000	ON → OFF	OFF → ON	OFF
	999 ← 1000	OFF → ON	ON → OFF	OFF
	1000 ≤ S• ≤ 2000	OFF	ON	OFF
	2000 → 2001	OFF	ON → OFF	OFF → ON
	2000 ← 2001	OFF	OFF → ON	ON → OFF
S• < S2•	2000 → 2001	OFF	ON → OFF	OFF → ON
	2000 ← 2001	OFF	OFF → ON	ON → OFF
	S• > 2000	OFF	OFF	ON

Related instructions

The following instructions can be combined with high speed counters:

Instruction	FNC No.	Instruction name
DHSCS	FNC 53	High speed counter set
DHSCR	FNC 54	High speed counter reset
DHSZ	FNC 55	High speed counter zone compare
DHCMOV	FNC189	High speed counter move
DHSCT	FNC280	High speed counter compare with data table

Cautions

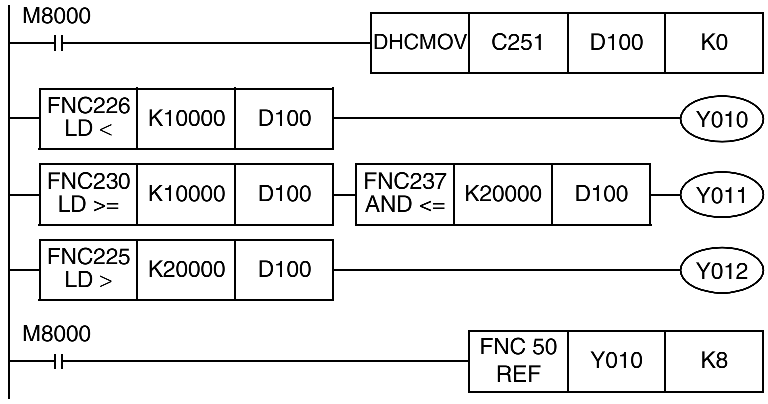
1. Selection of the count comparison method

When DHSZ instruction is used, hardware counters (C235, C236, C237, C238, C239, C240, C244 (OP), C245 (OP), C246, C248 (OP), C251 and C253) are automatically switched to software counters, and the maximum frequency of each counter and the total frequency are affected.

Refer to the counting operation described below, and select according to the contents of control whether to use DHSZ instruction or general-purpose comparison instruction.

1) Case to select DHSZ instruction
   - When the output should be given when the counting result becomes equivalent to the comparison value regardless of the scan time of the PLC

2) Cases to select a general-purpose comparison instruction
   - When the required frequency is beyond the counting performance of software counters
   - When counting is important, but the effect of the scan time can be ignored in operations depending on the counting result
   - When the number in an instruction is more than 32 bits



2. Device specification range

Only high speed counters (C235 to C255) can be specified as S·.

3. Only 32-bit operation instructions are available.

Because instructions for high speed counters are dedicated to 32 bits, make sure to input "DHSZ (FNC 55)".

4. Caution on values set in the comparison value 1 S1· and comparison value 2 S2·

Make sure that S1· is smaller than or equivalent to S2·.

5. Relationship between the comparison timing and the result output

1) DHSZ instruction executes comparison and outputs the result only when a counting pulse is input to a high speed counter.
   (When S1· is "1000" and S2· is "1999", the output D· is set to ON as soon as the current value of C235 changes from "999" to "1000" or from "1999" to "2000".)
2) Because the comparison result cannot be obtained when restoring the power or when the PLC mode switches from STOP to RUN, the result is not output even if the comparison condition is provided.

→ For details, refer to "13.6.1 Program in which comparison result is set to ON when power is turned ON [ZCP (FNC 11) instruction]".

6. Priority order in operation among HSCS (FNC 53), HSCR (FNC 54), and HSZ (FNC 55) instructions for a same high speed counter

→ For details, refer to "6. Priority order in operation among HSCS (FNC 53), HSCR (FNC 54), and HSZ (FNC 55) instructions for a same high speed counter" in Subsection 13.4.1.

7. Reset operation by an external terminal

→ For details, refer to "5. Reset operation by an external terminal [M8025*1: HSC (external reset) mode]" in Subsection 13.4.1.

8. Number of occupied devices

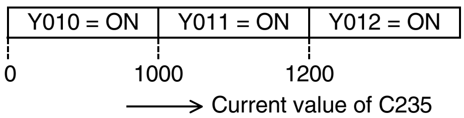
1) The comparison value occupies two devices from S1· or S2· respectively.
2) The output occupies three devices from D·.

13.6.1 Program in which comparison result is set to ON when power is turned ON [ZCP (FNC 11) instruction]

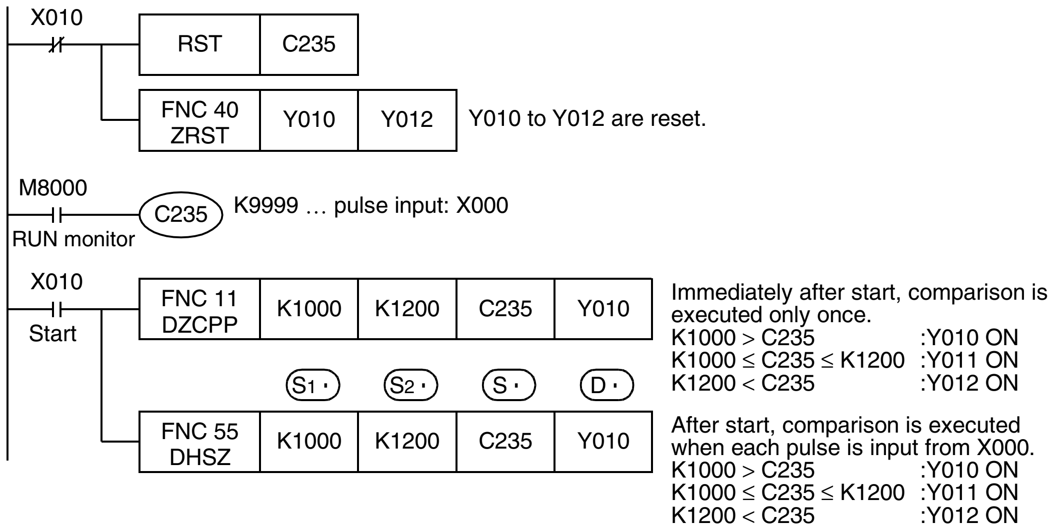
DHSZ instruction outputs the comparison result only when a counting pulse is input. Even if the current value of C235 is "0", Y010 remains OFF at the time of startup.
For initializing Y010, compare the current value of C235 with K1000 and K1200 and drive Y010 by DZCPP instruction (for general zone comparison) as pulse operation only at the time of startup.
Refer to the program example shown below.

Explanation of operation

The outputs Y010 to Y012 are as shown below:



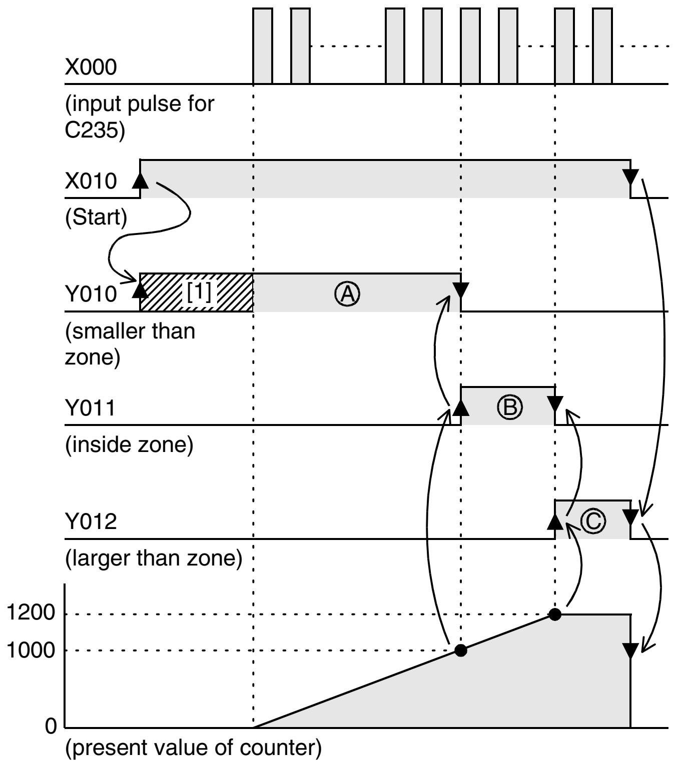
Program example



Timing chart

In the part [1] in the timing chart, Y010 remains OFF if the current value of a high speed counter (C235 in the example below) is "0" when restoring the power.

1) For initializing Y010, the current value of C235 is compared with K1000 and K1200, and Y010 is driven using the DZCPP instruction (for general zone comparison) as pulse operation only upon startup.
2) The comparison result in Y010 is latched until an input pulse is input and the comparison output is driven by the DHSZ instruction.
3) According to the current value of the counter, the DHSZ instruction drives the output (A), (B) or (C).


X000
(input pulse for C235)
X010
(Start)
Y010
[1]
Ⓐ
(smaller than zone)
Y011
Ⓑ
(inside zone)
Y012
Ⓒ
(larger than zone)
1200
1000
0
(present value of counter)

13.6.2 Table high speed comparison mode (M8130)

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This section explains the table high speed comparison mode (high speed pattern output) of the DHSZ instruction.
When two or more outputs should be activated at one time, use the HSCT instruction which can change up to 16 outputs.

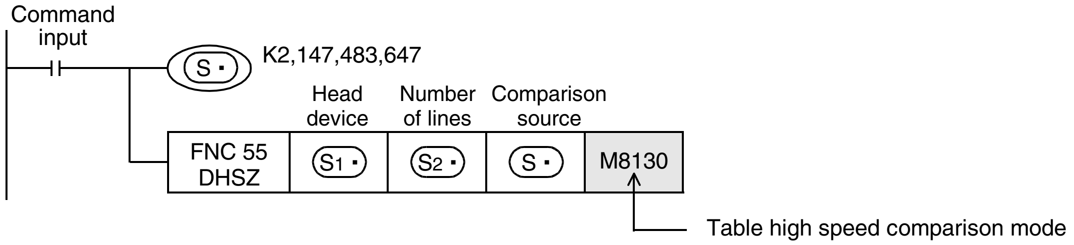
1. Set data

Operand Type	Description	Data Type
S1•	Head word device number storing the data table (only data register D)	32-bit binary
S2•	Number of lines in the table (only K or H) … K1 to K128 or H1 to H80	32-bit binary
S•	Device number of a high speed counter [C235 to C255]	32-bit binary
D•	M8130 (special auxiliary relay for declaring the table high speed comparison mode)	Bit

Explanation of function and operation

1. 32-bit operation (DHSZ)

When the special auxiliary relay M8130 for declaring the table high speed comparison mode is specified as D• in the DHSZ instruction, the special function shown below is provided.



Comparison table

Comparison data (32 bits)	Output (Y) number	SET/RST	Table counter (D8130)
S1• + 1, S1•	S1• + 2	S1• + 3	0 ↓
S1• + 5, S1• + 4	S1• + 6	S1• + 7	1 ↓
S1• + 9, S1• + 8	S1• + 10	S1• + 11	2 ↓
⋮	⋮	⋮	⋮
S1• + 5, S1• + 4	S1• + 6	S1• + 7	S2• – 1 ↓ Repeated from "0".

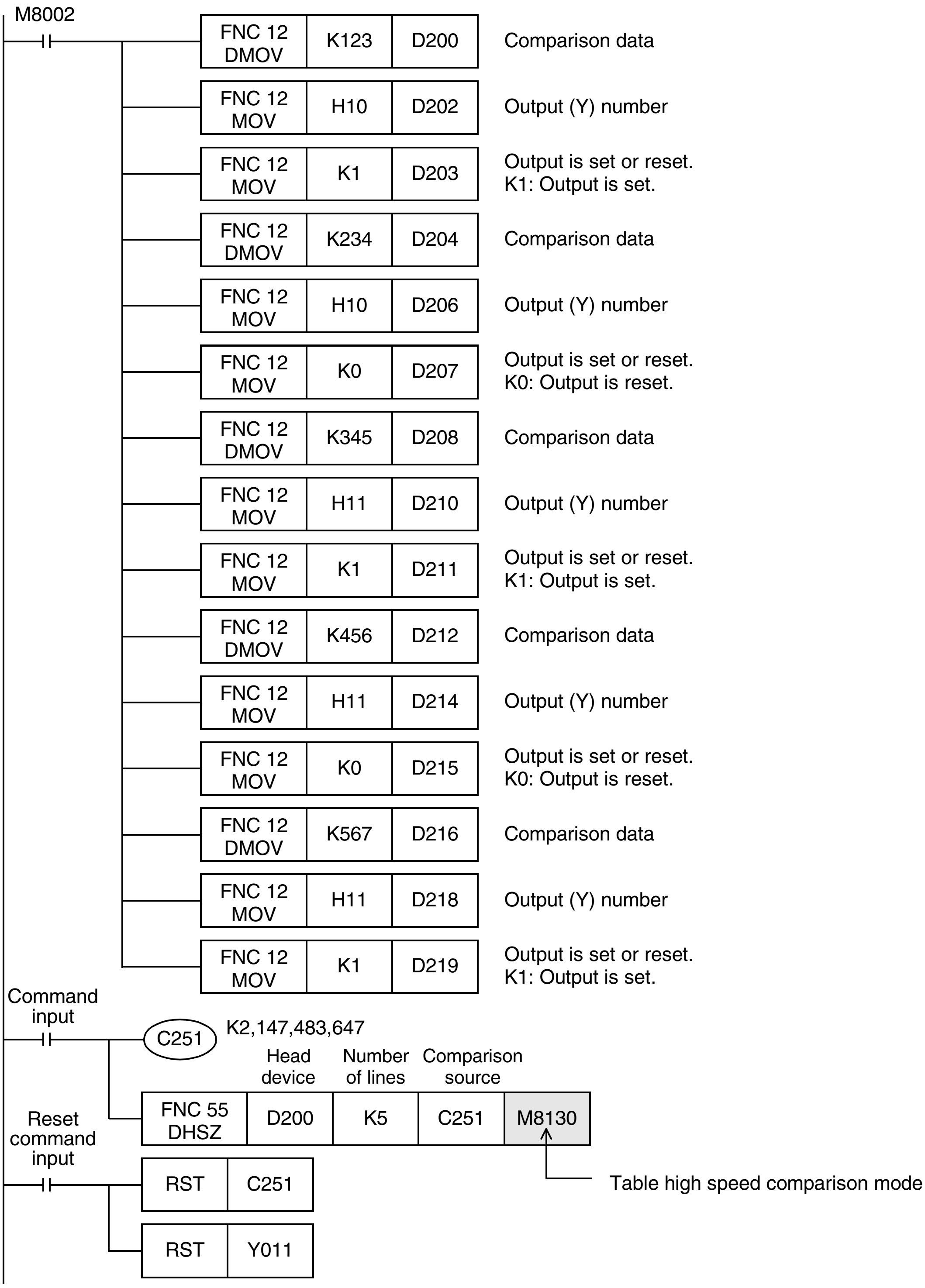
1) Specify the head device number for the comparison table as S1•.
   Because one line in the comparison table uses four devices, S2• × 4 devices are occupied from S1•.
2) Specify the number of lines in the comparison table as S2•.
   The created table starts from the head register S1•, and has the number of lines specified in S2•.
3) Comparison data
   Make sure that the comparison data is 32 bits.

4) Output (Y) number
   Specify each digit of the (Y) number in hexadecimal form.
   Example: When specifying Y010, specify "H10".
   When specifying Y020, specify "H20".

5) Specification of set and reset
   These set and reset are directly controlled as interrupt.

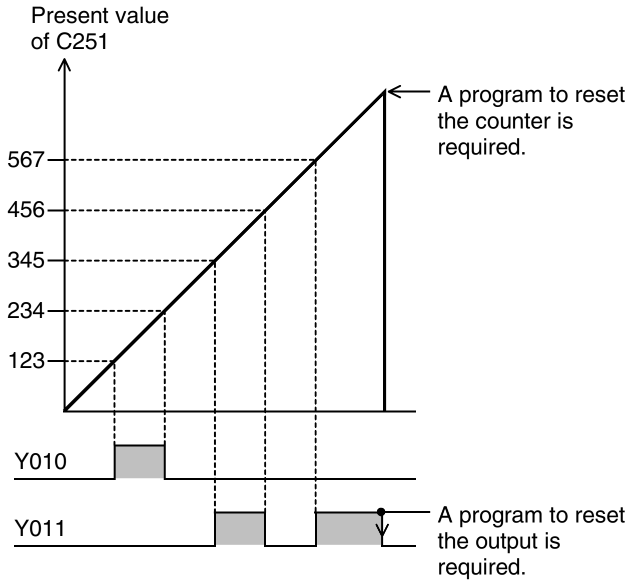
	Contents of setting
Set (ON)	K1/H1
Reset (OFF)	K0/H0

2. Operation



Comparison table

Comparison data	Output (Y) number	SET/RST	Table counter
D201,D200 K123	D 202 H10	D 203 K1	0 ↓
D205,D204 K234	D 206 H10	D 207 K0	1 ↓
D209,D208 K345	D 210 H11	D 211 K1	2 ↓
D213,D212 K456	D 214 H11	D 215 K0	3 ↓
D217,D216 K567	D 218 H11	D 219 K1	4 ↓ Repeated from "0".



1) When this instruction is executed, the top table in the data table is set as the comparison target data.
2) When the current value of the high speed counter C251 is equivalent to the comparison target data table, the output (Y) number specified in the table is set or reset.
   This output processing is directly executed without regard to completion of output refresh by END instruction.
3) "1" is added to the current value of the table counter D8130.
4) The comparison target data table is transferred to the next table.
5) The steps 2) and 3) are repeated until the current value of the table counter D8130 becomes "4".
   When the current value becomes "4", the program execution returns to the step 1), and the table counter D8130 is reset to "0".
   At this time, the complete flag M8131 turns ON.
6) When the command contact is set to OFF, execution of the instruction is stopped and the table counter D8130 is reset to "0".

Cautions

1. Limitation in the number of DHSZ instruction

This instruction can be programmed only once in a program.
With regard to the DHSCS, DHSCR, DHSZ and DHSCT instructions used for other purposes, up to 32 instructions including the DHSZ instruction can be driven at one time.

2. When the command input is set to OFF in the middle of execution

Execution of the instruction is aborted, and the table counter D8130 is reset to K0.
However, outputs which have been set or reset remain in the current status.

3. Output start timing

After the DHSZ instruction is first executed, creation of the table is completed by END instruction. After that, the DHSZ instruction becomes valid.
Accordingly, the output is activated from the second scan.

4. Current value of a high speed counter

Make sure to execute the DHSZ instruction from a point where the current value of the high speed counter (regarded as the operation target) is smaller than the value in the 1st line in the comparison table.

13.6.3 Frequency control mode (HSZ and PLSY instructions) (M8132)

When the special auxiliary relay M8132 for declaring the frequency control mode is specified as D· in the DHSZ instruction, the special function shown below is provided if DPLSY instruction is combined.

At this time, only a data register D can be specified as S1·, and a constant K or H can be specified as S2·. The available range is limited to "1 ≤ K, H ≤ 128".

A high speed counter C235 to C255 can be specified as S·.
This function is different from the zone comparison described above.
PLSY instruction is as shown on the next page, and only the pulse output can be changed by users.

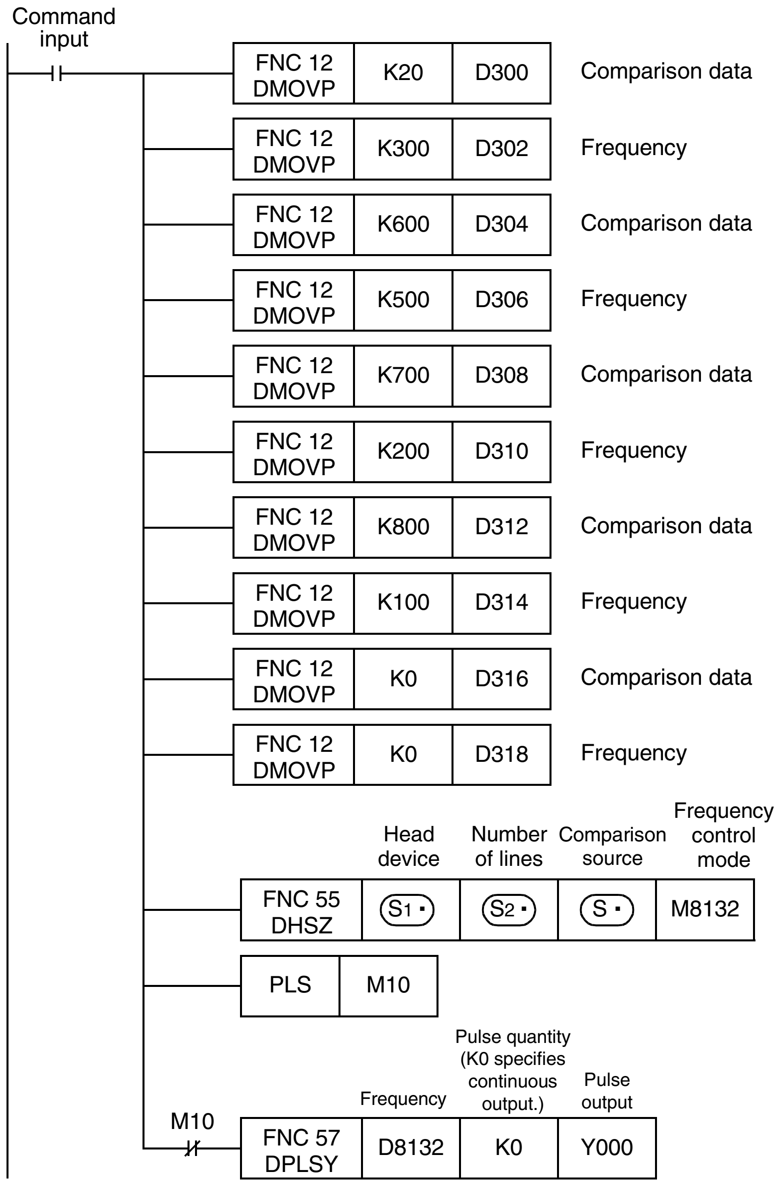
1. Control example

Example of table configuration and data setting

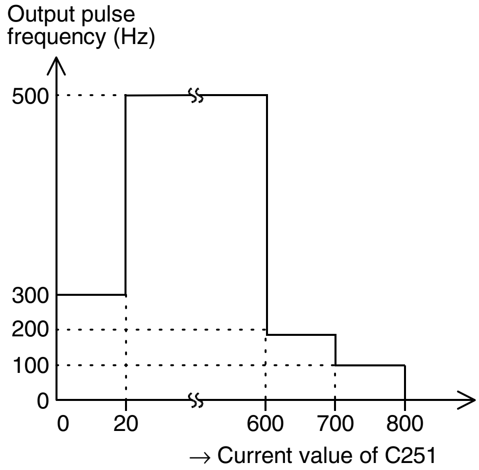
Comparison data	Frequency	Table counter D8131
D 301,D 300 K 20	D 302, D 303 K300	0 ↓
D 305,D 304 K600	D 306, D 307 K500	1 ↓
D 309,D 308 K700	D 310, D 311 K200	2 ↓
D 313,D 312 K800	D 314, D 315 K100	3 ↓
D 317,D 316 K 0	D 318, D 319 K 0	4 ↓

Head device (32 bits) specified as S1·

Number of lines specified as S2·


Command input
FNC 12 DMOVP K20 D300
Comparison data
FNC 12 DMOVP K300 D302
Frequency
FNC 12 DMOVP K600 D304
Comparison data
FNC 12 DMOVP K500 D306
Frequency
FNC 12 DMOVP K700 D308
Comparison data
FNC 12 DMOVP K200 D310
Frequency
FNC 12 DMOVP K800 D312
Comparison data
FNC 12 DMOVP K100 D314
Frequency
FNC 12 DMOVP K0 D316
Comparison data
FNC 12 DMOVP K0 D318
Frequency
Head device
Number of lines
Comparison source
Frequency control mode
FNC 55 DHSZ S1 S2 S M8132
PLS M10
M10
Frequency
Pulse quantity (K0 specifies continuous output.)
Pulse output
FNC 57 DPLSY D8132 K0 Y000

Output pulse characteristics



1) Write prescribed data in advance to data registers constructing the table as shown in this program example.
2) The output frequency of the PLSY instruction remains in the value (D303, D302) until the current value of a high speed counter specified in S· becomes equivalent to (D301, D300). (D302 specifies low-order 16 bits. D303 specifies high-order 16 bits, but is always "0".)
3) The operation in the 2nd line is started after that, and then the operation in each line is executed in turn.
4) When the operation in the last line is completed, the complete flag M8133 turns ON. The program execution returns to the 1st line, and the operation is repeated.
5) For stopping the operation in the last line, set the frequency in the last table to K0.
6) When the command input is set to OFF, the pulse output turns OFF and the table counter D8131 is reset.
7) After DHSZ instruction is first executed, creation of the table is completed at the END instruction. The DHSZ instruction becomes valid after that.
8) Accordingly, the contact of PLS M10 is used so that the PLSY instruction is executed from the second scan after the command input has been set to ON.

Data can be written to the table in a program as shown in this example or directly using keys in peripheral equipment.

1) M8132
   This is the special auxiliary relay for declaring the frequency control mode
2) D8132
   In the frequency control mode, the frequency set in the table is received by D8132 sequentially according to the table counter count D8131.
3) D8134 (low-order) and D8135 (high-order)
   In the frequency control mode, the comparison data in the table is received sequentially according to the table counter count.

Cautions

1) DHSZ instruction can be used only once.
2) With regard to the DHSCS (FNC 53), DHSCR (FNC 54), DHSZ (FNC 55) and DHSCT (FNC280) instructions used for other purposes, up to 32 instructions including the DHSZ instruction can be driven at one time.
3) Because the table is created when the END instruction is executed, it is necessary to delay execution of the PLSY (FNC 57) instruction until creation of the table is completed.
4) Do not change the data table while the DHSZ instruction is driven.
5) In the frequency control mode, simultaneous output to Y000 to Y001 is not permitted.

13.7 FNC 56 – SPD / Speed Detection

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction counts the input pulse for a specified period of time as interrupt input.
The function of this instruction varies depending on the version.

1. Instruction format

D FNC 56 SPD

16-bit Instruction	Mnemonic	Operation Condition
7 steps	SPD	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DSPD	Continuous Operation

2. Set data

Operand Type	Description	Data Type
S1•	Device number of pulse input (X)	Bit
S2•	Time data (ms) or word device number storing the data	16- or 32-bit binary
D•	Head word device number storing the pulse density data	16- or 32-bit binary

3. Applicable devices

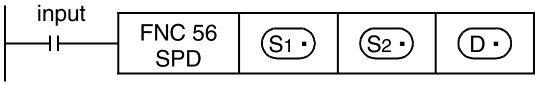
Oper-and Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•	▲																		✓					
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D•												✓	✓	✓	✓		✓	✓	✓					

▲: X000 to X007 can be specified.

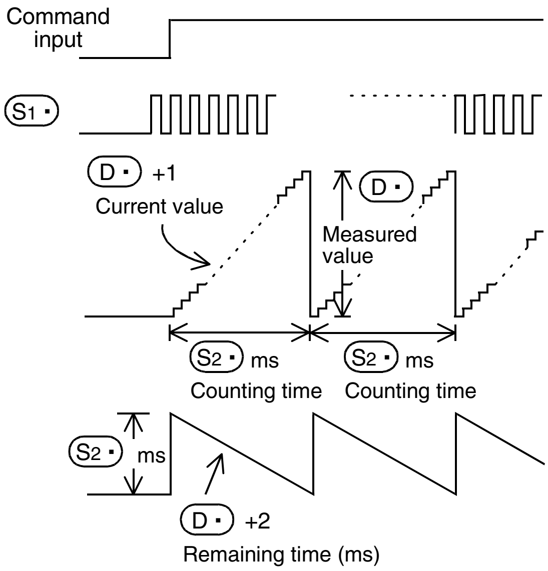
Explanation of function and operation

1. 16-bit operation (SPD)

The input pulse S1· is counted only for S2· × 1 ms. The measured value is stored in D·, the present value is stored in D·+1, and the remaining time is stored in D·+2 (ms).
By repeating this operation, the measured value D· will store the pulse density (which is proportional to the rotation speed).



1) Timing chart

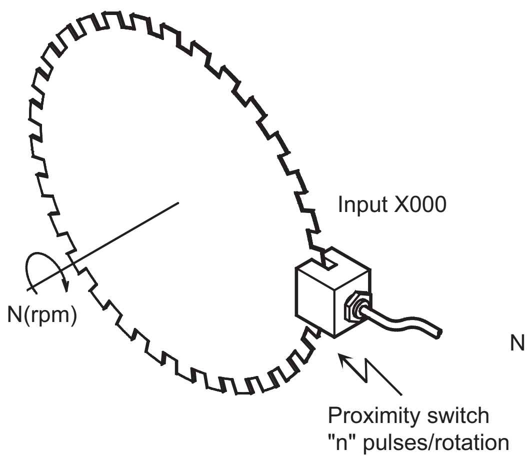


The command contact is set to ON.

D· + 1 counts "OFF → ON" operation of S1·.
S2· ms later, the counting result is stored to D·.
Accompanied by this operation, D·+1 is reset, and then counting of the "OFF → ON" operation of S1· is started again.

D· + 2 is used to measure the remaining time.

2) The measured value D· is in proportion to the number of rotations as shown below:



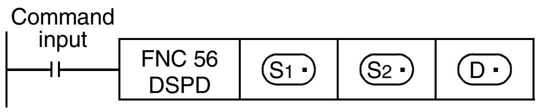
$$N = \frac{60\,(D\cdot)}{nt} \times 10^3 \text{ (r/min)}$$

t: Measurement time duration specified as S2· (ms)

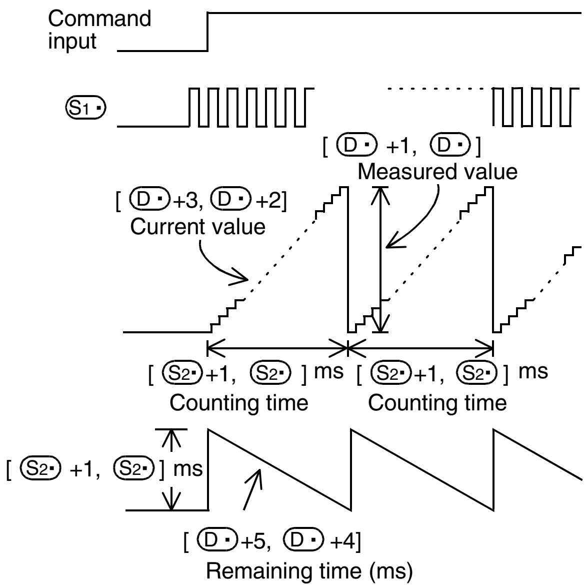
2. 32-bit operation (DSPD) [Ver.2.20 or later]

FX3UC Ver.2.20 → FX3U Ver.2.20 →

The input pulse S1• is counted only for [S2•+1, S2•] x 1 ms. The measured value is stored in [D•+1, D•], the present value is stored in [D•+3, D•+2], and the remaining time is stored in [D•+5, D•+4] (ms).
By repeating this operation, the measured value [D•+1, D•] will store the pulse density (which is proportional to the rotation speed).



1) Timing chart

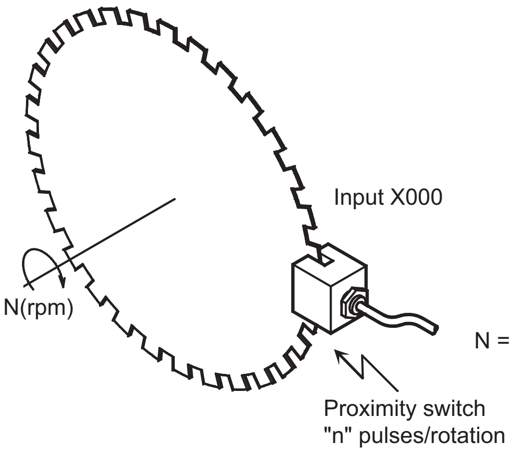


The command contact is set to ON.

[D•+3, D•+2] counts "OFF → ON" operation of S1•. [S2•+1, S2•] ms later, the counting result is stored to [D• +1, D•].
Accompanied by this operation, [D•+3, D•+2] is reset, and then counting of the "OFF → ON" operation of S1• is started again.

[D•+5, D•+4] is used to measure the remaining time.

2) The measured value [D• +1, D•] is in proportion to the number of rotations as shown below:



$$N = \frac{60\,[D\bullet +1, D\bullet]}{nt} \times 10^3 \text{ (r/min)}$$

t: Measurement time duration specified as [S2•+1, S2•] (ms)

Cautions

1. Input specifications of the input (S1·)

- An input device X000 to X007 specified as (S1·) cannot overlap the following usage:
  - High speed counter
  - Input interrupt
  - Pulse catch
  - DSZR instruction
  - DVIT instruction
  - ZRN instruction
- For one input, this instruction can be used at most only once.
- The maximum frequency of turning ON and OFF the inputs X000 to X007 is as shown below:

Used input number	Maximum input frequency		
	FX3UC PLC [FX3UC-32MT-LT]	FX3U PLC	
		Main unit	FX3U-4HSX-ADP
X000 to X005	100 kHz*1	100 kHz*1	200 kHz
X006,X007	10 kHz	10 kHz	

*1. When receiving pulses corresponding to the response frequency of 50 k to 100 kHz, perform the following:
- Make sure that the wiring length is 5 m or less.
- Connect a bleeder resistor of 1.5 kΩ (1 W or more) to the input terminal, and make sure that the load current in the open collector transistor output in the counterpart equipment is 20 mA or more.

2. Occupied devices

1) In the case of 16-bit operation
Three devices are occupied from a device specified in (D·).
2) In the case of 32-bit operation
Six devices are occupied from a device specified in (D·).

Function change depending on the version

The function of FNC 56 instruction is changed depending on the version as shown in the table below.

Applicable version		Item	Outline of function
FX3U	FX3UC		
Ver.2.20 or later	Ver.2.20 or later	Addition of 32-bit instruction	32-bit operations (DSPD) are enabled.

13.8 FNC 57 – PLSY / Pulse Y Output

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction generates a pulse signal.

→ **For the frequency control mode, refer to Subsection 13.6.3.**

1. Instruction format

D FNC 57 PLSY

16-bit Instruction	Mnemonic	Operation Condition
7 steps	PLSY	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DPLSY	Continuous Operation

2. Set data

Operand Type	Description	Data Type
S1•	Frequency (Hz) data or word device number storing data	16- or 32-bit binary
S2•	Pulse quantity data or word device number storing data	16- or 32-bit binary
D•	Device number (Y) from which pulses are output	Bit

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D•		▲																	✓					

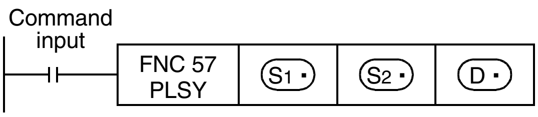
▲: Specify a transistor output in the main unit or Y000 or Y001 in a special high speed output adapter*1.

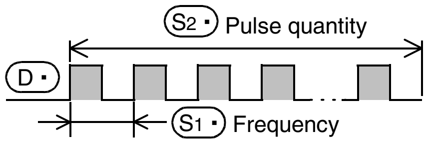
*1. Special high speed output adapters cannot be connected to the FX3UC-32MT-LT.

Explanation of function and operation

1. 16-bit operation (PLSY)

A pulse train at the frequency S1• is output by the quantity S2• from the output (Y) D•.

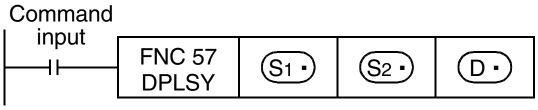


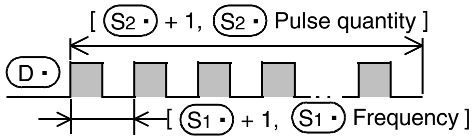


- Specify the frequency in S1•.
  Allowable setting range: 1 to 32767 (Hz)
- Specify the generated pulse quantity in S2•.
  Allowable setting range: 1 to 32767 (PLS)
- Specify the output (Y) number from which pulses are output in D•.
  Allowable setting range: Y000, Y001

2. 32-bit operation (DPLSY)

A pulse train at the frequency [S1·+1, S1·] is output by the quantity [S2·+1, S2·] from the output (Y) D·.





- Specify the frequency in [S1·+1, S1·].
  Allowable setting range: 1 to 200,000 (Hz)
- Specify the generated pulse quantity in [S2·+1, S2·].
  Allowable setting range: 1 to 2,147,483,647 (PLS)
- Specify the output (Y) number from which pulses are output in D·.
  Allowable setting range: Y000, Y001

→ **For the method to output pulses without any limitation, refer to the program example later.**

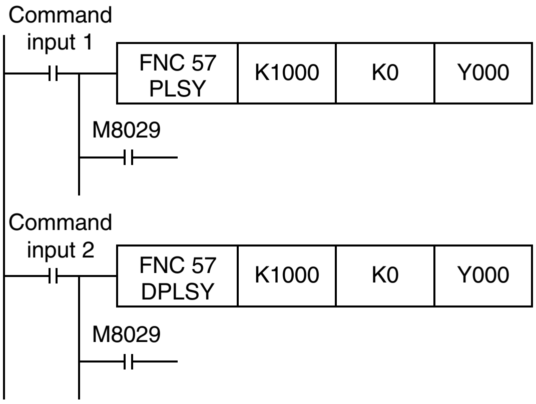
Related devices

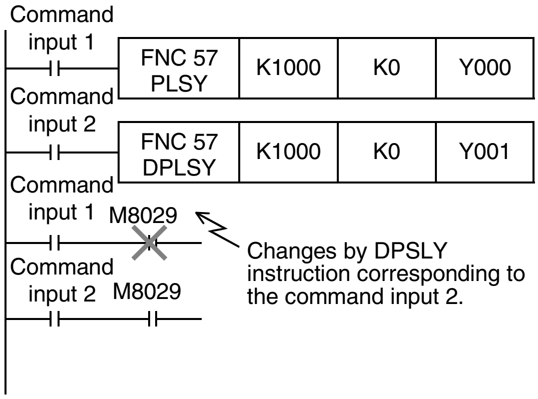
1. Instruction execution complete flag

The instruction execution complete flag M8029 used for PLSY instruction can be used also for other instructions. When using other instructions, setting the M8029 flag to ON or OFF, or using two or more PLSY instructions, make sure to use each M8029 flag just after an instruction to be monitored.

→ **For the instruction execution complete flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8029	Instruction execution complete	ON: Generation of specified number of pulses is completed. OFF: Generation of pulses is paused before the specified number of pulses is reached or the continuous pulse generation operation is stopped.





2. Monitoring of the current value of the number of generated pulses

The number of pulses output from Y000 or Y001 is stored in the following special data registers:

Device		Description	Contents of data
High order	Low order		
D8141	D8140	Accumulated number of pulses output from Y000	Accumulated number of pulses output from Y000 by PLSY and PLSR instructions
D8143	D8142	Accumulated number of pulses output from Y001	Accumulated number of pulses output from Y001 by PLSY and PLSR instructions
D8137	D8136	Total accumulated number of pulses output from Y000 and Y001	Total accumulated number of pulses output from Y000 and Y001 by PLSY and PLSR instructions

The contents of each data register can be cleared using the following program:

Command input

FNC 12 DMOV	K0	Low-order device shown in above table

3. How to stop the pulse output

- When the command input is set to OFF, the pulse generation is immediately stopped. When the command input is set to ON again, pulse generation operation is started from the beginning.
- When a special auxiliary relay (M) shown below is set to ON, the pulse output is stopped.

Device	Description
M8349	Immediately stops pulse output from Y000.
M8359	Immediately stops pulse output from Y001.

For outputting pulses again, set the device (M8349 or M8359) corresponding to the output signal to OFF, and then drive the pulse output instruction again.

Cautions

1. When a word device is specified as S1· or S2·

When the value of the word device is changed while the instruction is executed, the following operation results:

- When the data of S1· is changed, the output frequency changes accordingly.
- When the data of S2· is changed, the change (new value) becomes valid the next time the instruction is driven.

2. Frequency S1·

When using transistor outputs in the main unit, set the output frequency S1· to "100,000 Hz" or less.
If the load is operated using pulses at a frequency higher than 100,000 Hz, the PLC may be damaged.

3. Pulse output

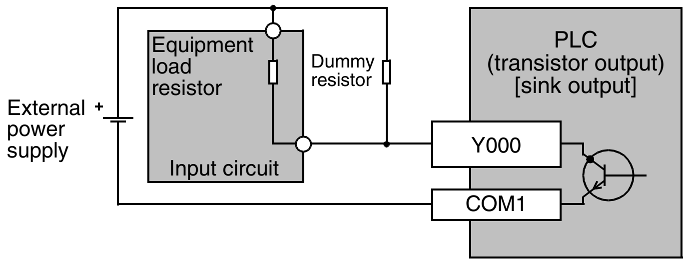
- Only a transistor output in the main unit or Y000 or Y001 in a special high speed output adapter[*1] can be specified in D·.
  When using PLSY (FNC 57) instruction in a relay output type FX3U PLC, a special high speed output adapter is required.
  *1. Special high speed output adapters cannot be connected to the FX3UC-32MT-LT.
- The duration of the ON/OFF pulses is 50% (ON = 50%, OFF = 50%).
- The pulse output is controlled by the dedicated hardware not affected by the sequence program (operation cycle).
- If the command input is set to OFF during continuous pulse output, the output from D· turns OFF.

4. Handling of pulse output terminals in the FX3U and FX3UC series main units

The outputs Y000 and Y001 are the high speed response type.
When using a pulse output instruction or positioning instruction, adjust the load current of the open collector transistor output to about 10 to 100 mA (5 to 24V DC).

Item	Description
Operating voltage range	5 to 24V DC
Operating current range	10 to 100 mA
Output frequency	100 kHz or less

When the load is smaller, connect a dummy resistor in parallel to the outside of a used output terminal (Y000 or Y001) as shown in the circuit diagram below so that the specified current shown above flows in the output transistor.



5. Cautions on using special high speed output adapters

1) Outputs of special high speed output adapters work as differential line drivers.
2) Set the pulse output type setting switch in a special high speed output adapter to the "pulse chain + direction" (PLS•DIR) side.
   If the switch is set to the "forward rotation pulse chain reverse rotation pulse chain" (FP•RP) side, normal operations are disabled. The pulse output destination changes depending on the PLC output status as shown in the table below.

Pulse output destination	Output affecting operation	Operation
D• = Y000	Y004	While Y004 is ON, pulses are output from Y000 in the high speed output adapter. While Y004 is OFF, pulses are output from Y004 in the high speed output adapter.
D• = Y001	Y005	While Y005 is ON, pulses are output from Y001 in the high speed output adapter. While Y005 is OFF, pulses are output from Y005 in the high speed output adapter.

3) Set the pulse output type setting switch while the PLC is stopped or while the power is OFF.
   Do not manipulate the pulse output form setting switch while pulses are being output.
4) When special high speed output adapters are connected, the same output numbers in the main unit are assigned as shown in the table below.
   Only wire the appropriate output terminals.
   Outputs in special high speed output adapters and the main unit operate as shown below.

Assignment of output numbers in special high speed output adapters

Status of output form setting switch	Signal name	Setting name in each positioning instruction	Output number			
			1st unit		2nd unit	
			1st axis	2nd axis	3rd axis	4th axis
"FP•RP" side	Forward rotation pulse chain (FP)	Pulse output destination	Y000	Y001	Y002	Y003
	Reverse rotation pulse chain (RP)	Rotation direction signal	Y004	Y005	Y006	Y007
"PLS•DIR" side	Pulse chain	Pulse output destination	Y000	Y001	Y002	Y003
	Direction	Rotation direction signal	Y004	Y005	Y006	Y007

Output operation

	Output operation
Relay output type main unit	Corresponding output remains ON (and the LED also remains ON) while the instruction is executed. Use a special high speed adapter
Special high speed output adapter	Outputs turn ON and OFF normally. Set the output frequency to "200kHz" or less.
Transistor output type main unit	Outputs turn ON and OFF normally. Set the output frequency to "100kHz" or less.

6. Others

1) Types of pulse output, positioning and other relevant instructions and their target output numbers

Classification	Instruction	Instruction name	Target output numbers
Pulse output	PLSY(FNC 57)	Pulse Y output	Y000,Y001
	PLSR(FNC 59)	Acceleration/deceleration setup	Y000,Y001
Positioning	DSZR(FNC150)	DOG search zero return	Y000,Y001,Y002,Y003
	DVIT(FNC151)	Interrupt positioning	Y000,Y001,Y002,Y003
	ZRN(FNC156)	Zero return	Y000,Y001,Y002,Y003
	PLSV(FNC157)	Variable speed pulse output	Y000,Y001,Y002,Y003
	DRVI(FNC158)	Drive to increment	Y000,Y001,Y002,Y003
	DRVA(FNC159)	Drive to absolute	Y000,Y001,Y002,Y003
High speed processing	PWM(FNC 58)	Pulse width modulation	Y000,Y001,Y002,Y003

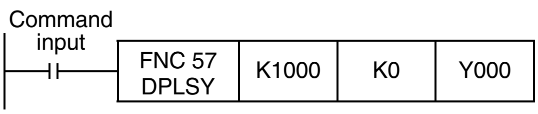
2) When using the same output relay (Y000 or Y001) in several instructions.
While a pulse output monitor (BUSY/READY) flag is ON a pulse output instruction and positioning instruction for the same output relay cannot be executed.
While a pulse output monitor flag is ON even after the instruction drive contact is set to OFF, a pulse output instruction or positioning instruction for the same output relay cannot be executed.
Before executing such an instruction, wait until the pulse output monitor flag turns OFF and one or more operation cycles pass.

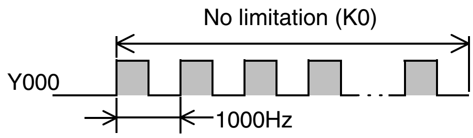
Pulse output destination device	Pulse output monitor flag
Y000	M8340
Y001	M8350

3) "Frequency control mode" in which DHSZ (FNC 55) and PLSY (FNC 57) instructions are combined can be used only once in a program.

Program example (when outputting pulses without any limitation)

When (S2•) is set to K0, pulses are output without any limitation.





13.9 FNC 58 – PWM / Pulse Width Modulation

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction outputs pulses with a specified period and ON duration.

1. Instruction format

FNC 58
PWM

16-bit Instruction	Mnemonic	Operation Condition
7 steps	PWM	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	

2. Set data

Operand Type	Description	Data Type
(S1•)	Pulse width data (ms) or word device number storing data	16-bit binary
(S2•)	Period data (ms) or word device number storing data	16-bit binary
(D•)	Device number (Y) from which pulses are output	Bit

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S1•)								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
(S2•)								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
(D•)		▲																	✓					

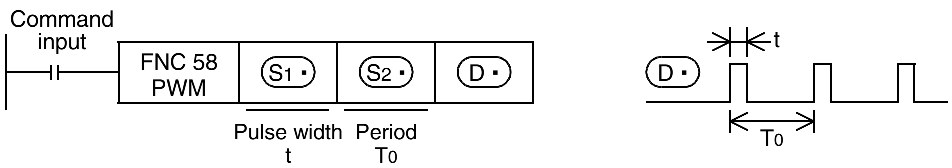
▲: Specify a transistor output Y000, Y001, or Y002 in the main unit or Y000, Y001, Y002, or Y003 in a special high speed output adapter*1.

*1. Special high speed output adapters cannot be connected to the FX3UC-32MT-LT.

Explanation of function and operation

1. 16-bit operation (PWM)

Pulses whose ON pulse width is (S1•) ms are output in periods of (S2•) ms.



- Specify the pulse width "t" in (S1•).
  Allowable setting range: 0 to 32767 ms
- Specify the period "T0" in (S2•).
  Allowable setting range: 1 to 32767 ms
- Specify the output (Y) number from which pulses are output in (D•).
  Allowable setting range: Y000, Y001, Y002, Y003

11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FMC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

Cautions

1. Setting the pulse width and period

Make sure that the pulse width (S1•) and period (S2•) satisfy the relationship "(S1•) ≤ (S2•)".

2. Pulse output

- Only the following outputs can be specified in (D•) according to the system configuration.
  - When using special high speed output adapters*1: Y000, Y001, Y002*2, or Y003*2
  - When using transistor outputs in the main unit (that is, when not using special high speed output adapters): Y000, Y001, or Y002

*1. Special high speed output adapters cannot be connected to the FX3UC-32MT-LT.
When using PWM (FNC 58) instruction in a relay output type FX3U PLC, a special high speed output adapter is required.

*2. For specifying Y002 or Y003 in a special high speed output adapter, the second special high speed output adapter is required.

- The pulse output is controlled by the interrupt processing not affected by the sequence program (operation cycle).
- If the command input is set to OFF, the output from (D•) turns OFF.
- While a pulse output monitor (BUSY/READY) flag is ON, a pulse output instruction and positioning instruction for the same output relay cannot be executed.
  While a pulse output monitor flag is ON even after the instruction drive contact is set to OFF, a pulse output instruction or positioning instruction for the same output relay cannot be executed.
  Before executing such an instruction, wait until the pulse output monitor flag turns OFF and one or more operation cycles pass.

Pulse output destination device	Pulse output monitor flag
Y000	M8340
Y001	M8350
Y002	M8360
Y003	M8370

3. Cautions on using special high speed output adapters

1) Outputs of special high speed output adapters work as differential line drivers.
2) Set the pulse output type setting switch in a special high speed output adapter to the "pulse chain + direction" (PLS•DIR) side.
   If the switch is set to the "forward rotation pulse chain reverse rotation pulse chain" (FP•RP) side, normal operations are disabled. The pulse output destination changes depending on the output status as shown in the table below.

Pulse output destination	Output affecting operation	Operation
(D•) = Y000	Y004	While Y004 is ON, pulses are output from Y000 in the high speed output adapter. While Y004 is OFF, pulses are output from Y004 in the high speed output adapter.
(D•) = Y001	Y005	While Y005 is ON, pulses are output from Y001 in the high speed output adapter. While Y005 is OFF, pulses are output from Y005 in the high speed output adapter.
(D•) = Y002	Y006	While Y006 is ON, pulses are output from Y002 in the high speed output adapter. While Y006 is OFF, pulses are output from Y006 in the high speed output adapter.
(D•) = Y003	Y007	While Y007 is ON, pulses are output from Y003 in the high speed output adapter. While Y007 is OFF, pulses are output from Y007 in the high speed output adapter.

3) Set the pulse output type setting switch while the PLC is stopped or while the power is OFF.
   Do not manipulate the pulse output type setting switch while pulses are being output.

4) When special high speed output adapters are connected, the same output numbers in the main unit are assigned as shown in the table below.
Only wire the appropriate output terminals.
Outputs in special high speed output adapters and the main unit operate as shown below.

Assignment of output numbers in special high speed output adapters

Setting status of output form setting switch	Signal name	Setting name in each positioning instruction	Output number			
			1st unit		2nd unit	
			1st axis	2nd axis	3rd axis	4th axis
"FP•RP" side	Forward rotation pulse chain (FP)	Pulse output destination	Y000	Y001	Y002	Y003
	Reverse rotation pulse chain (RP)	Rotation direction signal	Y004	Y005	Y006	Y007
"PLS•DIR" side	Pulse chain	Pulse output destination	Y000	Y001	Y002	Y003
	Direction	Rotation direction signal	Y004	Y005	Y006	Y007

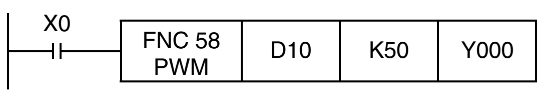
Output operation

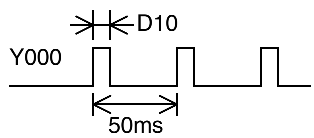
	Output operation
Relay output type main unit	Outputs turn ON and OFF normally, but do not use PWM (FNC 58) instruction because it does not support relay outputs. (Do not use PWM instruction because considerable output response delay may be generated, chattering may occur in contacts, or the contact life may be shortened.) Use a special high speed adapter.
Special high speed output adapter	Outputs turn ON and OFF normally. Set the output frequency to "200kHz" or less.
Transistor output type main unit	Outputs turn ON and OFF normally. Set the output frequency to "100kHz" or less.

Program example

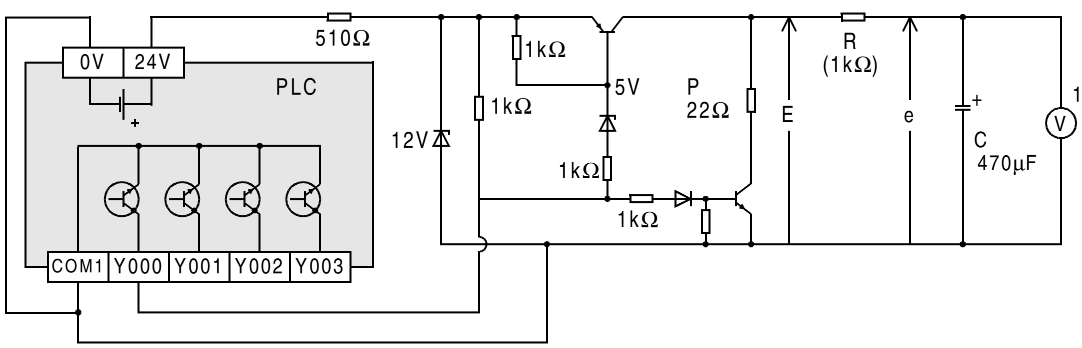
When the contents of D10 are changed in the range from "0" to "50" in the program example shown below, the average output from Y000 will be in the range from 0 to 100%.
In this program example the FX3U series main unit (sink output) is used. For wiring details, refer to the following manual.

→ **FX3U Hardware Edition**





Example of smoothing circuit



R > P
τ = P(kΩ) × C(μF) = 470ms >> T_0
The time constant of the filter should be considerably larger than the pulse cycle T_0.
The ripple value "Δe" in the mean output current "e" is approximately "$\Delta e/e \le T_0/\tau$"

13.10 FNC 59 – PLSR / Acceleration/Deceleration Setup

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This pulse output instruction has the acceleration/deceleration function.

1. Instruction format

	FNC 59
D	PLSR

16-bit Instruction	Mnemonic	Operation Condition
9 steps	PLSR	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DPLSR	Continuous Operation

2. Set data

Operand Type	Description	Data Type
S1•	Maximum frequency (Hz) data or word device number storing data	16- or 32-bit binary
S2•	Data of total number of output pulses (PLS) or word device number storing data	16- or 32-bit binary
S3	Acceleration/deceleration time (ms) data or word device number storing data	16-bit binary
D•	Device number (Y) from which pulses are output	Bit

3. Applicable devices

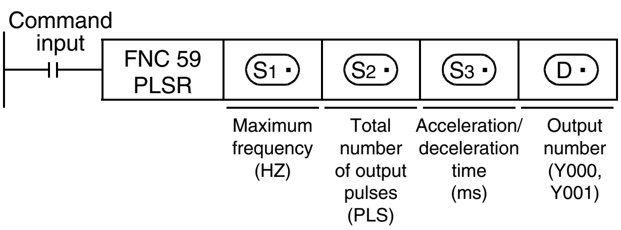
Oper-and Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S3								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D•		▲																	✓					

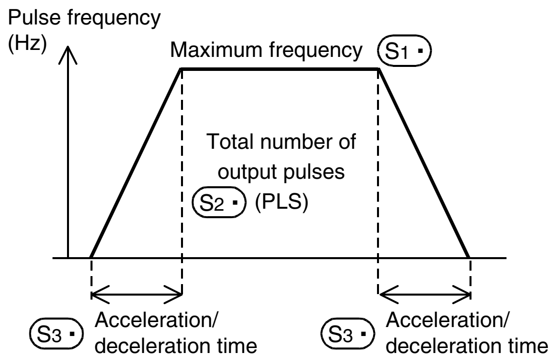
▲: Specify a transistor output in the main unit or Y000 or Y001 in a special high speed output adapter*1.
*1. Special high speed output adapters cannot be connected to the FX3UC-32MT-LT.

Explanation of function and operation

1. 16-bit operation (PLSR)

Pulses are output from the output (Y) D• by the specified number S2• with acceleration/deceleration to the maximum frequency S1• for the time S3• (ms).





S1• : Maximum frequency (Hz)
Allowable setting range: 10 to 32767 (Hz)

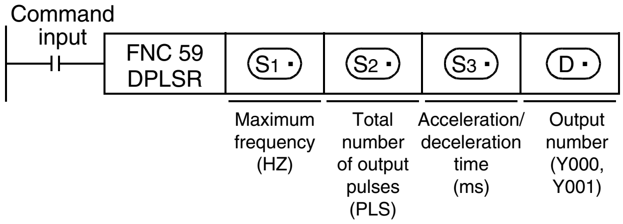
S2• : Total number of output pulses (PLS)
Allowable setting range: 1 to 32767 (PLS)

S3• : Acceleration/deceleration time (ms)
Allowable setting range: 50 to 5000 (ms)

D• : Pulse output number
Allowable setting range: Y000, Y001

2. 32-bit operation (DPLSR)

Pulses are output from the output (Y) D· by the specified number [S2·+1, S2·] with acceleration/deceleration to the maximum frequency [S1·+1, S1·] for the time [S3·+1, S3·](ms).



[S1·+1, S1·]: Maximum frequency (Hz)
Allowable setting range: 10 to 200,000 (Hz)

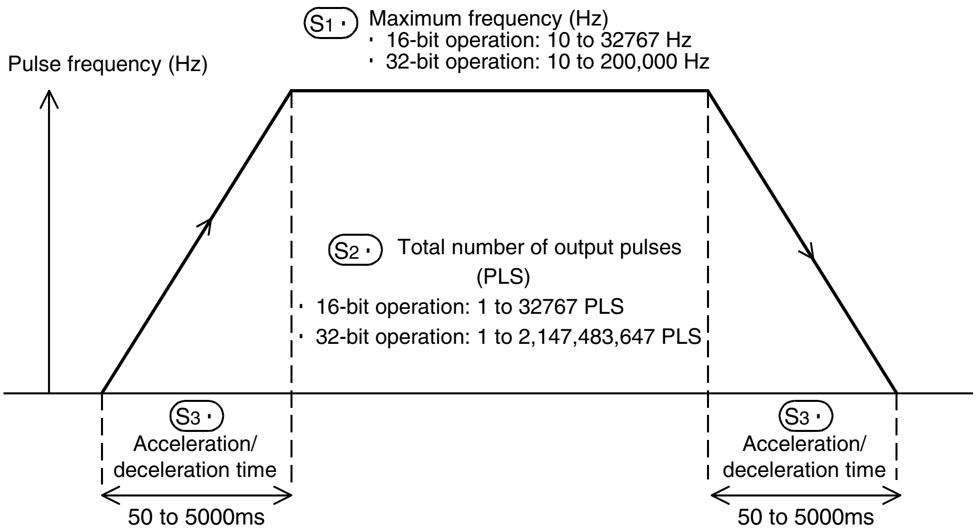
[S2·+1, S2·]: Total number of output pulses (PLS)
Allowable setting range: 1 to 2,147,483,647 (PLS)

[S3·+1, S3·]: Acceleration/deceleration time (ms)
Allowable setting range: 50 to 5000 (ms)

D·: Pulse output number
Allowable setting range: Y000, Y001

3. Pulse output specifications

- Simple positioning (with the acceleration/deceleration function)
  The operation pattern is as shown below:



- Output processing
  The pulse output is controlled by the dedicated hardware regardless of the operation cycle.
- Data change while the instruction is executed
  Even if operands are overwritten while the instruction is executed, such changes are not reflected immediately. The changes become valid the next time the instruction is driven.

Related devices

1. Instruction execution complete flag

→ For the instruction execution complete flag use method, refer to Subsection 6.5.2.

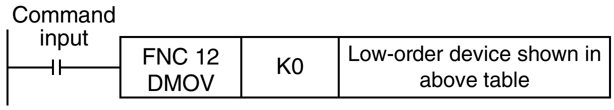
Device	Name	Description
M8029	Instruction execution complete	OFF: The input command is OFF, or pulses are being output. (This flag does not turn ON if the pulse output is interrupted in the middle of output.) ON: Output of the number of pulses set in S2• is completed.

2. Monitoring of the number of generated pulses

The number of pulses output from Y000 or Y001 is stored in the following special data registers:

Device		Description	Contents of data
High order	Low order		
D8141	D8140	Accumulated number of pulses output from Y000	Accumulated number of pulses output from Y000 by PLSY and PLSR instructions
D8143	D8142	Accumulated number of pulses output from Y001	Accumulated number of pulses output from Y001 by PLSY and PLSR instructions
D8137	D8136	Total accumulated number of pulses output from Y000 and Y001	Total accumulated number of pulses output from Y000 and Y001 by PLSY and PLSR instructions

The contents of each data register can be cleared using the following program:



3. How to stop the pulse output

- When the command input is set to OFF, the pulse generation is immediately stopped. When the command input is set to ON again, pulse generation operation is started from the beginning.
- When a special auxiliary relay (M) shown below is set to ON, the pulse output is stopped.

Device	Description
M8349	Immediately stops pulse output from Y000.
M8359	Immediately stops pulse output from Y001.

To output pulses again, set the device (M8349 or M8359) corresponding to the output signal to OFF, and then drive the pulse output instruction again.

Cautions

1. Frequency S1•

When using transistor outputs in the main unit, set the output frequency S1• to "100,000 Hz" or less.
If the load is operated using pulses at a frequency higher than 100,000 Hz from transistor outputs in the main unit, the PLC may be damaged.

2. Pulse output

- Only a transistor output in the main unit or Y000 or Y001 in a special high speed output adapter*1 can be specified in D•.

  *1. Special high speed output adapters cannot be connected to the FX3UC-32MT-LT.
  When using PLSR (FNC 59) instruction in a relay output type FX3U PLC, a special high speed output adapter is required.

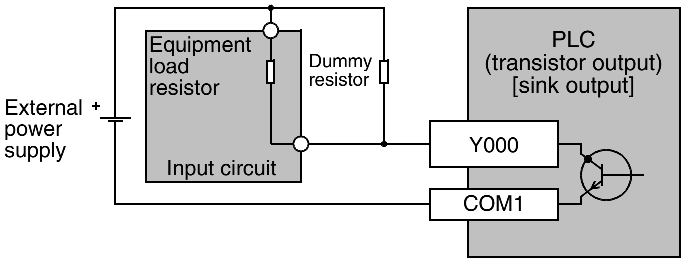
- The duration of the ON/OFF pulses is 50% (ON = 50%, OFF = 50%).
- The pulse output is controlled by the dedicated hardware not affected by the sequence program (operation cycle).
- If the command input is set to OFF during continuous pulse output, the output from D• turns OFF.

3. Handling of pulse output terminals in the FX3U and FX3UC series main units

The outputs Y000 and Y001 are the high speed response type.
When using a pulse output instruction or positioning instruction, adjust the load current of the open collector transistor output to about 10 to 100 mA (5 to 24V DC).

Item	Description
Operating voltage range	5 to 24V DC
Operating current range	10 to 100 mA
Output frequency	100 kHz or less

When the load is smaller, connect a dummy resistor in parallel to the outside of a used output terminal (Y000 or Y001) as shown in the circuit diagram below so that the specified current shown above flows in the output transistor.



4. Cautions on special high speed output adapters

1) Outputs of special high speed output adapters work as differential line drivers.
2) Set the pulse output type setting switch in a special high speed output adapter to the "pulse chain + direction" (PLS•DIR) side.
If the switch is set to the "forward rotation pulse chain reverse rotation pulse chain" (FP•RP) side, normal operations are disabled. The pulse output destination changes depending on the PLC output status as shown in the table below.

Pulse output destination	Output affecting operation	Operation
D• = Y000	Y004	While Y004 is ON, pulses are output from Y000 in the high speed output adapter. While Y004 is OFF, pulses are output from Y004 in the high speed output adapter.
D• = Y001	Y005	While Y005 is ON, pulses are output from Y001 in the high speed output adapter. While Y005 is OFF, pulses are output from Y005 in the high speed output adapter.

3) Set the pulse output type setting switch while the PLC is stopped or while the power is OFF.
Do not manipulate the pulse output type setting switch while pulses are being output.
4) When special high speed output adapters are connected, the same output numbers in the main unit are assigned as shown in the table below.
Only wire the appropriate output terminals.
Outputs in special high speed output adapters and the main unit operate as shown below.

Assignment of output numbers in special high speed output adapter

Setting status of output form setting switch	Signal name	Setting name in each positioning instruction	Output number			
			1st unit		2nd unit	
			1st axis	2nd axis	3rd axis	4th axis
"FP•RP" side	Forward rotation pulse chain (FP)	Pulse output destination	Y000	Y001	Y002	Y003
	Reverse rotation pulse chain (RP)	Rotation direction signal	Y004	Y005	Y006	Y007
"PLS•DIR" side	Pulse chain	Pulse output destination	Y000	Y001	Y002	Y003
	Direction	Rotation direction signal	Y004	Y005	Y006	Y007

Output operation

	Output operation
Relay output type main unit	Corresponding output remains ON (LED remains ON also) while the instruction is executed. Use a special high speed adapter.
Special high speed output adapter	Outputs turn ON and OFF normally. Set the output frequency to "200kHz" or less.
Transistor output type main unit	Outputs turn ON and OFF normally. Set the output frequency to "100kHz" or less.

5. Others

1) Types of pulse output, positioning and other relevant instructions and their target output numbers

Classification	Instruction	Instruction name	Target output numbers
Pulse output	PLSY(FNC 57)	Pulse Y output	Y000,Y001
	PLSR(FNC 59)	Acceleration/deceleration setup	Y000,Y001
Positioning	DSZR(FNC150)	DOG search zero return	Y000,Y001,Y002,Y003
	DVIT(FNC151)	Interrupt positioning	Y000,Y001,Y002,Y003
	ZRN(FNC156)	Zero return	Y000,Y001,Y002,Y003
	PLSV(FNC157)	Variable speed pulse output	Y000,Y001,Y002,Y003
	DRVI(FNC158)	Drive to increment	Y000,Y001,Y002,Y003
	DRVA(FNC159)	Drive to absolute	Y000,Y001,Y002,Y003
High speed processing	PWM(FNC 58)	Pulse width modulation	Y000,Y001,Y002,Y003

2) When using the same output relay (Y000 or Y001) in several instructions.
While a pulse output monitor (BUSY/READY) flag is ON, a pulse output instruction and positioning instruction for the same output relay cannot be executed.
While a pulse output monitor flag is ON, even after the instruction drive contact is set to OFF, a pulse output instruction or positioning instruction for the same output relay cannot be executed.
Before executing such an instruction, wait until the pulse output monitor flag turns OFF and one or more operation cycles pass.

Pulse output destination device	Pulse output monitor flag
Y000	M8340
Y001	M8350

14. Handy Instruction – FNC 60 to FNC 69

FNC 60 to FNC 69 provide handy instructions which achieve complicated control in a minimum sequence program.

FNC No.	Mnemonic	Symbol	Function	Reference
60	IST	IST S D1 D2	Initial State	Section 14.1
61	SER	SER S1 S2 D n	Search a Data Stack	Section 14.2
62	ABSD	ABSD S1 S2 D n	Absolute drum sequencer	Section 14.3
63	INCD	INCD S1 S2 D n	Incremental drum sequencer	Section 14.4
64	TTMR	TTMR D n	Teaching Timer	Section 14.5
65	STMR	STMR S m D	Special Timer	Section 14.6
66	ALT	ALT D	Alternate State	Section 14.7
67	RAMP	RAMP S1 S2 D n	Ramp Variable Value	Section 14.8
68	ROTC	ROTC S m1 m2 D	Rotary Table Control	Section 14.9
69	SORT	SORT S m1 m2 D n	SORT Tabulated Data	Section 14.10

14.1 FNC 60 – IST / Initial State

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction automatically controls the initial state and special auxiliary relays in a step ladder program.

→ **For SFC programs and step ladder, refer to Chapter 34.**

1. Instruction format

FNC 60 IST

16-bit Instruction	Mnemonic	Operation Condition
7 steps	IST	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand type	Description	Data type
(S•)	Head bit device number of the selector switch in the operation mode	Bit
(D1•)	Smallest state relay number of practical state relays in the automatic mode ((D1•) < (D2•))	Bit
(D2•)	Largest state relay number of practical state relays in the automatic mode ((D1•) < (D2•))	Bit

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index				Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify		K	H	E	"□"	P
(S•)	✓	✓	✓				▲1												✓						
(D1•)						▲2													✓						
(D2•)						▲2													✓						

▲1: "D□.b" cannot be indexed with index registers (V and Z).
▲2: S20 to S899 and S1000 to S4095

Explanation of function and operation

Command input

FNC 60 IST	(S•)	(D1•)	(D2•)

- Specify the head input in the operation mode in (S•).
  Selector switches in the operation mode occupy eight devices from the head device (S•), and the switch functions shown in the table below are assigned to each of them.
  When X020 is assigned as shown below, it is necessary to set X020 to X024 as rotary switches so that they do not turn ON at the same time.
  It is not necessary to wire unused switches, but they cannot be used for any other purpose because they are occupied by IST instruction.

Source	Device number (example)	Switch function
(S•)	X020	Individual operation
(S•) + 1	X021	Return to zero point
(S•) + 2	X022	Stepping
(S•) + 3	X023	Cycle operation
(S•) + 4	X024	Continuous operation
(S•) + 5	X025	Zero return start
(S•) + 6	X026	Automatic start
(S•) + 7	X027	Stop

- Specify the smallest device number of practical state relays in (D1•) (for the automatic mode).
- Specify the largest device number of practical state relays in (D2•) (for the automatic mode).

1. Control of devices by switch operations (occupied devices)

While the command input is ON, the following devices are automatically switched and controlled. While the command input is OFF, the devices are not switched.

Device number	Operation function
M8040	STL transfer disable
M8041*1	Transfer start
M8042	Start pulse
M8043*1	Zero return complete
M8045	All output reset disable
M8047*2	Enable STL monitoring

Device number	Operation function
S0	Individual operation initial state
S1	Zero return initial state
S2	Automatic operation initial state

*1.Cleared when the PLC mode is changed from RUN to STOP.

*2.Set to ON when END instruction is executed.

Do not program the following state relays as general state relays;

Device number	Operation function
S0 to S9	Occupied for the initial state • S0 to S2 are used for individual operation, zero return and automatic operation as shown above. • S3 to S9 can be used arbitrarily.
S10 to S19	Occupied for zero return

If the devices are switched among individual operation (X020), zero return (X021) and automatic operation (X022, X023 and X024) while the zero return complete device (M8043) is OFF, all outputs are set to OFF.
Automatic operation can be started again after zero return is completed.

→ **For introducing IST instruction, refer to "14.1.2 Example of IST instruction introduction (example of workpiece transfer mechanism)".**

Cautions

1. Device specified as (S·) and switches to be used

It is not necessary to use all switches for mode selection.
When some switches are not used, leave the corresponding numbers in the unused status. Such numbers cannot be used for any other purpose.

2. Programming order of IST instruction and STL instruction

- IST instruction should be programmed earlier than a series of STL circuit such as state relays S0 to S2.

3. State relays used for the zero return operation

Use the state relays S10 to S19 for the zero return operation.
In the final state in the zero return operation, set M8043 to ON, and then let it be reset to OFF by itself.

4. Limitation in the number of IST instruction

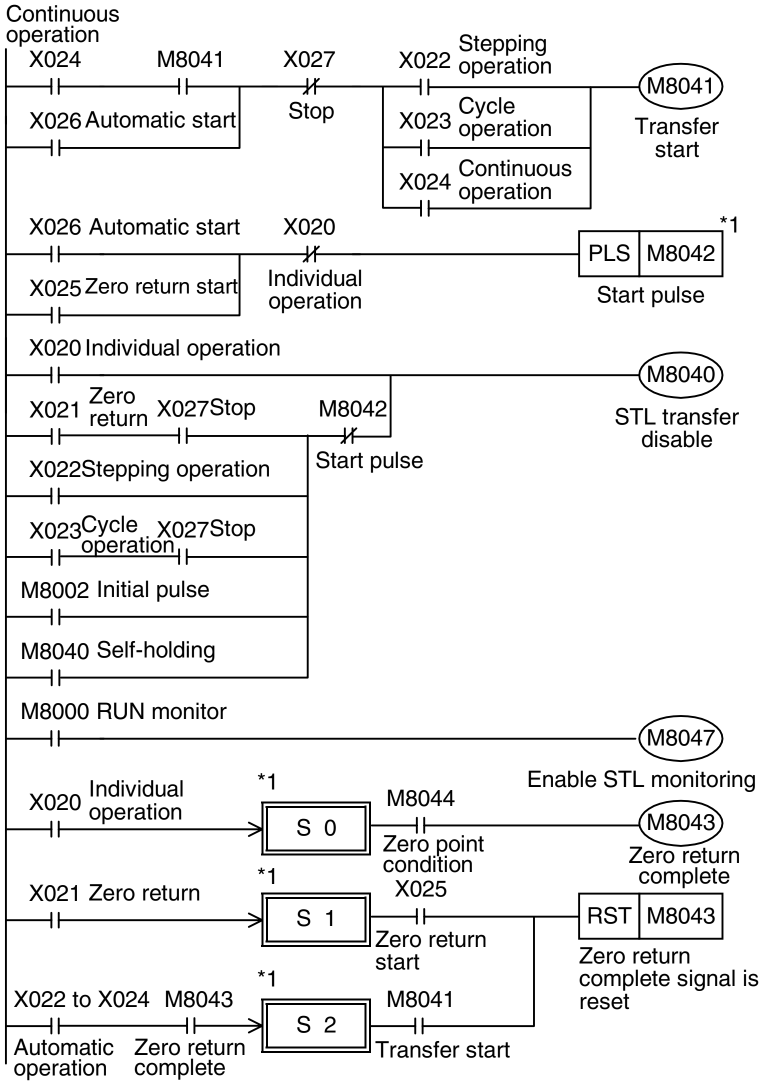
IST instruction can be used only once in a program.

11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FMC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

14.1.1 IST instruction equivalent circuit

The details of special auxiliary relays (M) and initial state relays (S0 to S9) which are automatically controlled by IST instruction are as shown in the equivalent circuit below. (Refer to the equivalent circuit below for reference.) **This equivalent circuit cannot be programmed.**

1. Equivalent circuit



M8041 is set to ON when the start button is pressed in the automatic mode. Especially in the continuous mode, M8041 holds its status by itself, and is reset when the stop button is pressed.

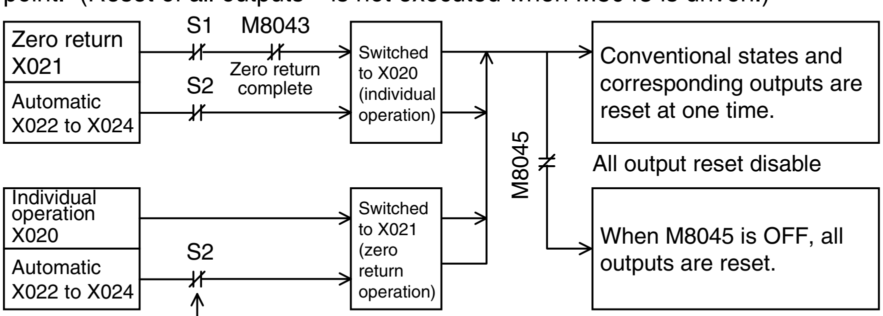
M8040 is set to ON in the stepping mode, and set to OFF every time the start button is pressed.
In the zero return operation or cycle operation, M8040 holds its status by itself when the stop button is pressed, and is reset when the start button is pressed.

The initial state is switched according to each mode input, and M8043 is controlled at the same time.
However, it is necessary to control M8044 and M8043 in user programs also.

*1. Because the above equivalent circuit is provided only for explanation, it cannot be actually programmed.

2. Switching of the operation mode

When the operation mode is switched among the individual operation, zero return operation and automatic operation, all outputs and conventional states are reset at one time unless the machine is located in the zero point. (Reset of all outputs*1 is not executed when M8045 is driven.)

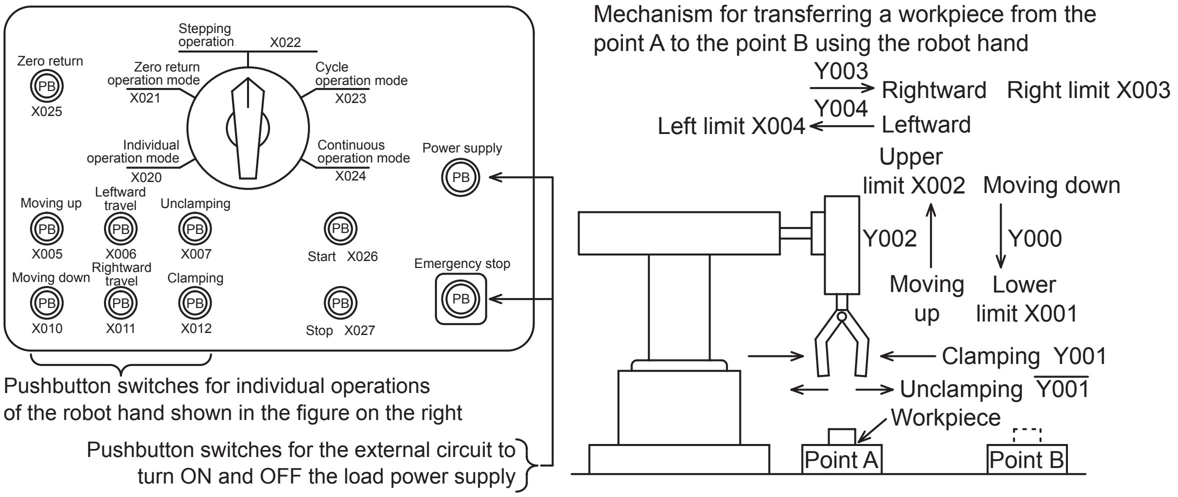


Even if the mode is switched from automatic operation to zero return operation while S2 is ON, state relays (except initial state relays) and outputs are not reset.

*1. All outputs: Outputs (Y) not driven by state relays (S) and outputs (Y) driven by state relays (S) in OUT and SET instructions

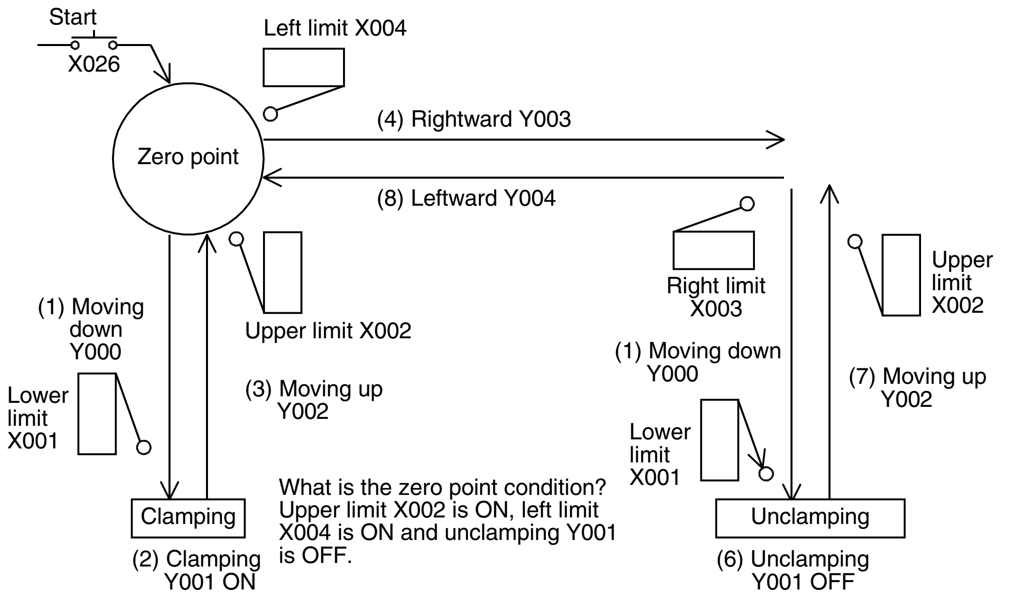
14.1.2 Example of IST instruction introduction (example of workpiece transfer mechanism)

1. Operation mode



Operation mode		Contents of operation
Manual mode	Individual operation mode	Each load is turned ON and OFF by an individual pushbutton switch.
	Zero return operation mode	When the pushbutton switch for zero return is pressed, the machine automatically returns to the zero point.
Automatic mode	Stepping operation mode	Every time the start button is pressed, the machine performs one process.
	Cycle operation mode	When the start button is pressed while the machine is located at the zero point, the machine performs one cycle of automatic operation and stops at the zero point. If the stop button is pressed in the middle of one cycle, the machine stops immediately. When the start button is pressed after that, the machine performs the continuous operation from the last position, and automatically stops at the zero point.
	Continuous operation mode	When the start button is pressed while the machine is located at the zero point, the machine starts continuous operation. When the stop button is pressed, the machine finishes the current cycle until the zero point, and then stops at the zero point.

2. Transfer mechanism



The upper left position is regarded as the zero point. The machine transfers a workpiece from the left to the right in the order "moving down → clamping → moving up → rightward travel → moving down → unclamping → moving up → leftward travel."

Double-solenoid type solenoid valves (with two inputs for driving and non-driving) are adopted for moving down, moving up, leftward travel and rightward travel. Single type solenoid valves (which operate only while the power is ON) are adopted for clamping.

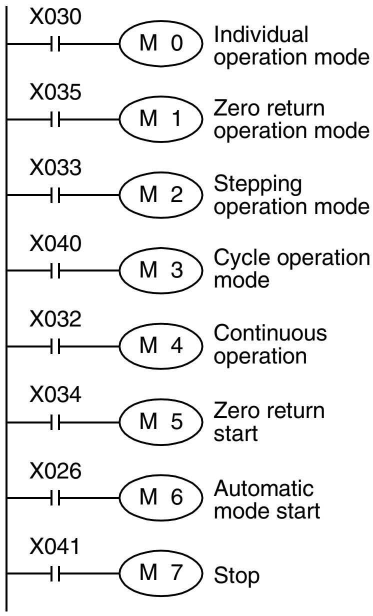
3. Assignment of mode selection inputs

For using IST instruction, it is necessary to assign inputs having consecutive device numbers as shown below for mode inputs.
When using non-consecutive inputs or omitting some modes, change the layout by using an auxiliary relay as the head input for mode specification as shown in the figure below.

- X020: Individual operation mode
- X021: Zero return operation mode
- X022: Stepping operation mode
- X023: Cycle operation mode
- X024: Continuous operation
- X025: Zero return start
- X026: Automatic mode start
- X027: Stop

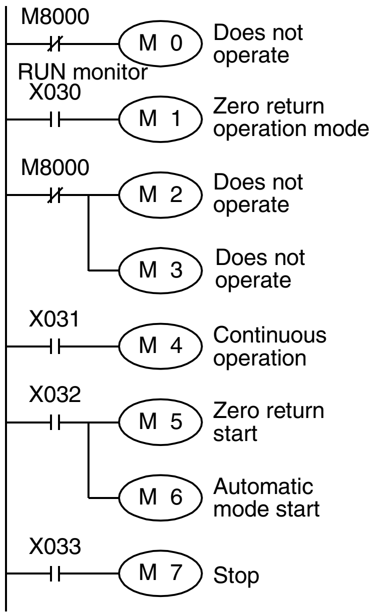
When inputs do not have consecutive device numbers

Example:
X030: Individual operation mode
X035: Zero return operation mode
X033: Stepping operation mode
X040: Cycle operation mode
X032: Continuous operation mode
X034: Zero return start
X026: Automatic mode start
X041: Stop



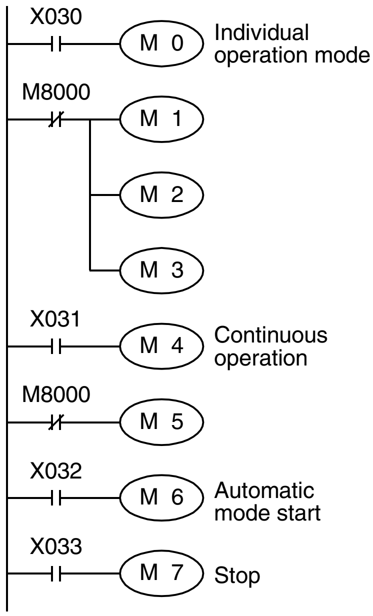
When only the continuous operation mode and zero return operation mode are used

Example:
X030: Zero return operation mode
X031: Continuous operation mode
X032: Automatic mode start zero return start
X033: Stop

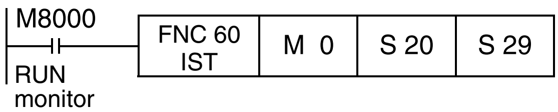


When only the continuous operation mode and individual operation mode are used

Example:
X030: Individual operation mode
X031: Continuous operation mode
X032: Automatic mode start
X033: Stop



In this example, M0 is used as the head input for mode specification.



4. Special auxiliary relay (M) for IST instruction

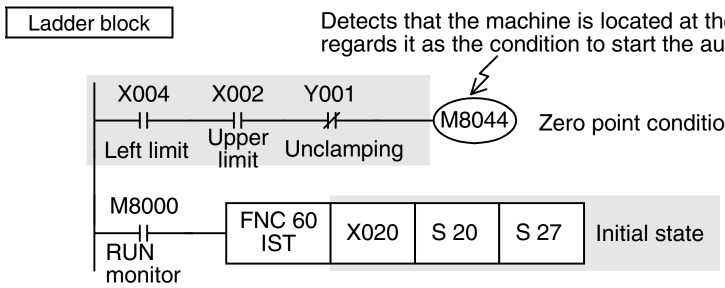
Auxiliary relays (M) used in IST instruction are classified into two types. Some auxiliary relays are automatically controlled by IST instruction itself according to the situation. Other auxiliary relays should be controlled by a program for preparation of operation or for purpose of control.

1) Special auxiliary relays automatically controlled by IST instruction
   a) M8040: STL transfer disable
      When this special auxiliary relay turns ON, transfer of every state is disabled.
      Individual operation mode:
      M8040 is always effective.
      Zero return operation mode and cycle operation mode:
      When the stop button is pressed, the operation is held until the start button is pressed.
      Stepping operation mode:
      M8040 is always effective except when the start button is pressed. When the start button is pressed, M8040 is not effective and transfer of states is allowed.
      Others: The operation is latched when the PLC mode switches from STOP to RUN, and reset when the start button is pressed.
      Even in the transfer disabled status, the operation is held for outputs in the states.
   b) M8041: Transfer start
      This special auxiliary relay allows transfer from the initial state S2 to the next state.
      Individual operation mode and zero return operation mode:
      M8041 is not effective.
      Stepping operation mode and cycle operation mode:
      M8041 is effective only while the start button is pressed and held.
      Continuous operation mode:
      The operation is latched when the start button is pressed, and cleared when the stop button is pressed.
   c) M8042: Start pulse
      M8042 is activated instantaneously only when the start button is pressed.
   d) M8047: Enable STL monitoring
      When IST instruction is executed, M8047 is set to ON.
      When the M8047 turns ON, STL monitoring becomes valid, and state relay numbers (S0 to S899) in the ON status are stored in turn in the ascending order of device number to the special auxiliary relays D8040 to D8047.
      Up to eight state relay numbers in the ON status can be monitored.
      If either state relay is ON, the special auxiliary relay M8046 is set to ON.
2) Auxiliary relays controlled by a sequence program
   → **For details of these controls, refer to the next page.**
   a) M8043: Zero return complete
      Set this special auxiliary relay (M) to ON by a user program when the machine returns to the zero point in the zero return operation mode.
   b) M8044: Zero point condition
      Detect the zero point condition of the machine, and drive this special auxiliary relay. This signal is effective in every mode.
   c) M8045: All output reset disable
      When the mode is switched among individual operation mode, zero return operation mode and automatic mode, all outputs and operation state relays are reset if the machine is not located at the zero point.
      If M8045 has been set to ON in advance, however, only operation state relays are reset.

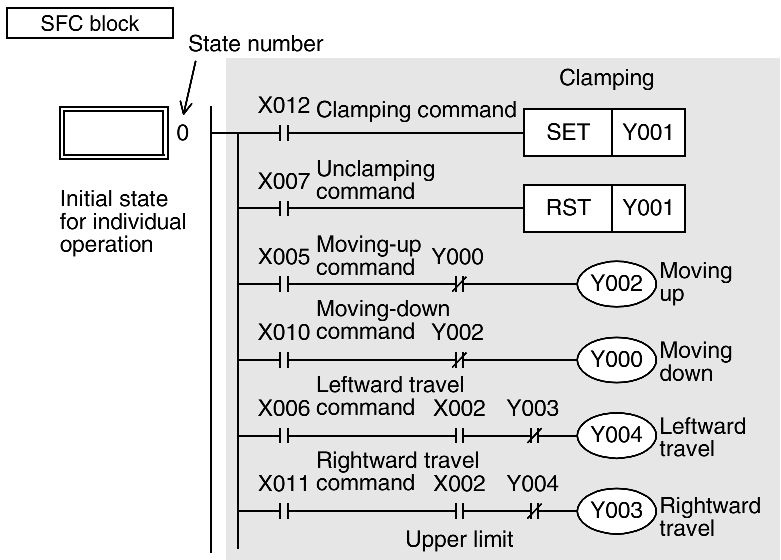
5. Program example

1) Circuit diagram
In the sequence circuit shown below, all areas except shaded areas are standard.
Program the shaded areas according to the contents of control.

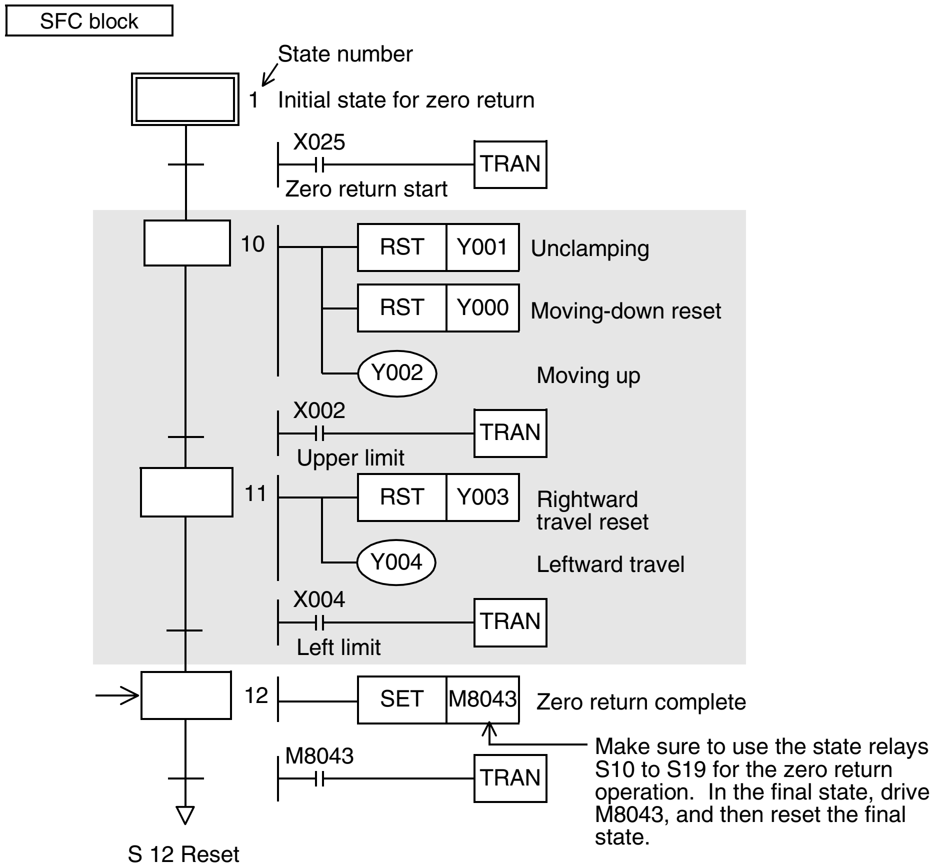
a) Initial circuit
While the machine is operating, the operation mode can be switched arbitrarily (among stepping operation, cycle operation and continuous operation) in the automatic mode.
When the operation mode is switched between the individual operation mode, zero return operation mode and automatic mode while the machine is operating, all outputs are reset once to assure safety, after which the following mode becomes valid.
(While M8045 (All output reset disable) is ON, outputs are not reset at all.)



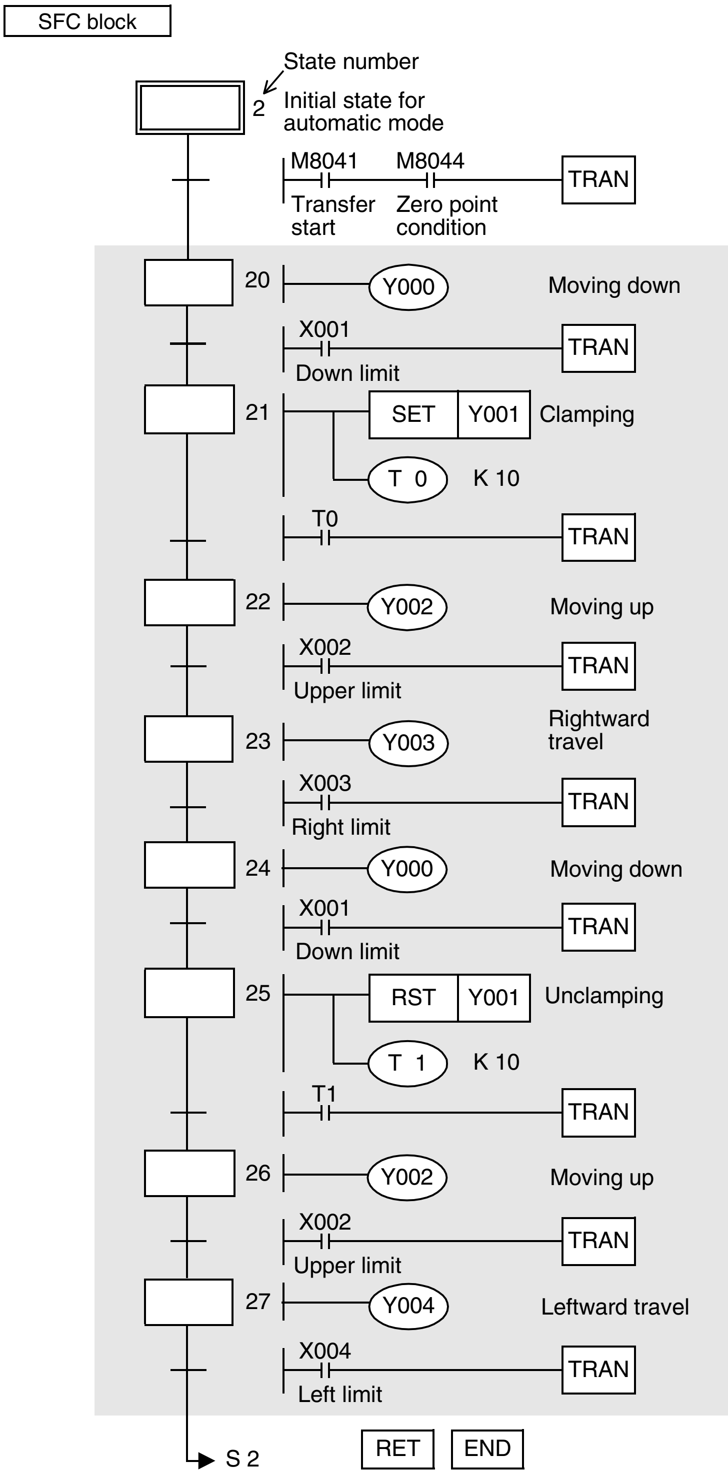
b) Individual operation mode
Programming is not required when the individual operation mode is not provided.



c) Zero return operation mode
Programming is not required when the zero return operation mode is not provided.
It is necessary to set M8043 (zero return complete) to ON before starting the automatic mode.



d) Automatic mode (stepping operation mode, cycle operation mode or continuous operation mode)



6. List program

The list program for the circuit diagram shown on the previous page is as shown below:

Initial circuit

```
0   LD   X  004
1   AND  X  002
2   ANI  Y  001
3   OUT  M8044
5   LD   M8000
6   IST     60
    (SP) X  020
    (SP) S  20
    (SP) S  27
13  STL  S  0
```

Individual operation

```
14  LD   X  012
15  SET  Y  001
16  LD   X  007
17  RST  Y  001
18  LD   X  005
19  ANI  Y  000
20  OUT  Y  002
21  LD   X  010
22  ANI  Y  002
23  OUT  Y  000
24  LD   X  006
25  AND  X  002
26  ANI  Y  003
27  OUT  Y  004
28  LD   X  011
29  AND  X  002
30  ANI  Y  004
31  OUT  Y  003
    (RET)
```

Programming is not required.

Zero return operation

```
32  STL  S  1
33  LD   X  025
34  SET  S  10
36  STL  S  10
37  RST  Y  001
38  RST  Y  000
39  OUT  Y  002
40  LD   X  002
41  SET  S  11
43  STL  S  11
44  RST  Y  003
45  OUT  Y  004
46  LD   X  004
47  SET  S  12
49  STL  S  12
50  SET  M8043
52  LD   M8043
53  RST  S  12
    (RET)
```

```
55  STL  S  2
56  LD   M8041
57  AND  M8044
58  SET  S  20
60  STL  S  20
61  OUT  Y  000
62  LD   X001
63  SET  S  21
6S  TL   S  21
66  SET  Y  001
67  OUT  T  0
    (SP) K  10
```

Automatic mode

```
70  LD   T  0
71  SET  S  22
73  STL  S  22
74  OUT  Y  002
75  LD   X  002
76  SET  S  23
78  STL  S  23
79  OUT  Y  003
80  LD   X  003
81  SET  S  24
83  STL  S  24
84  OUT  Y  000
85  LD   X  001
86  SET  S  25
88  STL  S  25
89  RST  Y  001
90  OUT  T  1
    (SP) K  10
93  LD   T  1
94  SET  S  26
96  STL  S  26
97  OUT  Y  002
98  LD   X  002
99  SET  S  27
101 STL  S  27
102 OUT  Y  004
103 LD   X  004
104 OUT  S  2
```

```
106 RET
107 END
```

14.2 FNC 61 – SER / Search a Data Stack

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction searches for the same data, maximum value and minimum value in a data table.

1. Instruction format

D	FNC 61 SER	P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	SER	Continuous Operation
	SERP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DSER	Continuous Operation
	DSERP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S1·	Head device number in which same data, maximum value and minimum value are searched	16- or 32-bit binary
S2·	Data to be searched for or device number storing data	16- or 32-bit binary
D·	Head device number storing number of same data, maximum value and minimum value detected by search	16- or 32-bit binary
n	Number of data in which same data, maximum value and minimum value are searched [16-bit instruction: 1 to 256, 32-bit instruction: 1 to 128]	16- or 32-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1·								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
S2·								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D·									✓	✓	✓	✓	✓	✓	✓	✓			✓					
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (SER and SERP)

In "n" data starting from S1·, same data as S2· is searched, and the search result is stored to D· to D· +4.

Command input	FNC 61 SER	S1·	S2·	D·	n

1) Contents of searched data and the search result
   a) When same data was detected
      Five devices starting from D· store the number of same data, first position, last position, maximum value position and minimum value position.
   b) When same data was not detected
      Five devices starting from D· store the number of same data, first position, last position, maximum value position and minimum value position.
      In this case, however, "0" is stored in three devices starting from D· (which store the number of same data, first position and last position).

11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FNC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

2) Operation example

a) Example of search result table configuration and data

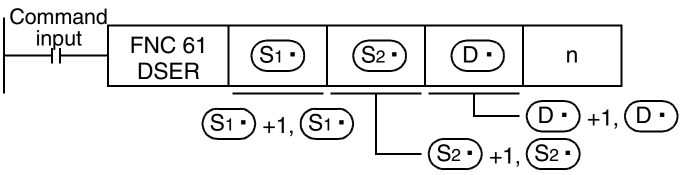
Searched device S1·	Searched data S1· value (example)	Comparison data S2· value (example)	Data position	Search result		
				Maximum value D· + 4	Same D·	Minimum value D· + 3
S1·	K100	K100	0		✓ (first position)	
S1· + 1	K111		1			
S1· + 2	K100		2		✓	
S1· + 3	K 98		3			
S1· + 4	K123		4			
S1· + 5	K 66		5			✓
S1· + 6	K100		6		✓ (last position)	
S1· + 7	K 95		7			
S1· + 8	K210		8	✓		
S1· + 9	K 88		9			

b) Search result table

Device number	Contents	Search result item
D·	3	Number of same data
D· + 1	0	Same data position (first position)
D· + 2	6	Same data position (last position)
D· + 3	5	Minimum value position (last position)
D· + 4	8	Maximum value position (last position)

2. 32-bit operation (DSER and DSERP)

In "n" data starting from [S1·+1, S1·], same data as [S2·+1, S2·] is searched, and the search result is stored to [D·+1, D·] to [D·+9, D·+8].



1) Contents of searched data and the search result

a) When same data was detected

Five 32-bit devices starting from [D·+1, D·] store the number of same data, first position, last position, maximum value position and minimum value position.

b) When same data was not detected

Five 32-bit devices starting from [D·+1, D·] store the number of same data, first position, last position, maximum value position and minimum value position.

In this case, however, "0" is stored in three devices starting from [D·+1, D·] (which store the number of same data, first position and last position).

2) Operation example

a) Example of search result table configuration and data

Searched device (S1•)+1	Searched data (S1•) value (example)	Comparison data (S2•)	Data position	Search result: Maximum value (D•)+4	Search result: Same (D•)	Search result: Minimum value (D•)+3
[(S1•)+1, (S1•)]	K100000	K100000	0		✓ (first position)	
[(S1•)+3, (S1•)+2]	K110100		1			
[(S1•)+5, (S1•)+4]	K100000		2		✓	
[(S1•)+7, (S1•)+6]	K 98000		3			
[(S1•)+9, (S1•)+8]	K123000		4			
[(S1•)+11, (S1•)+10]	K 66000		5			✓
[(S1•)+13, (S1•)+12]	K100000		6		✓ (last position)	
[(S1•)+15, (S1•)+14]	K 95000		7			
[(S1•)+17, (S1•)+16]	K910000		8	✓		
[(S1•)+19, (S1•)+18]	K910000		9	✓		

b) Search result table

Device number	Contents	Search result item
[(D•)+1, (D•)]	3	Number of same data
[(D•)+3, (D•)+2]	0	Same data position (first position)
[(D•)+5, (D•)+4]	6	Same data position (last position)
[(D•)+7, (D•)+6]	5	Minimum value position (last position)
[(D•)+9, (D•)+8]	9	Maximum value position (last position)

Cautions

1. Comparison of values

It is executed algebraically.
(example: –10 < 2)

2. When there are two or more maximum or minimum values

When there are two or more maximum or minimum values in the searched data, the last position of the max/min is stored respectively.

3. Number of occupied devices

When this instruction is driven, the following number of devices are occupied for storing the search result (D•).
Make sure that such devices are not used in other controls for the machine.

1) In the case of 16-bit operation
Five devices, (D•), (D•)+1, (D•)+2, (D•)+3 and (D•)+4, are occupied.

2) In the case of 32-bit operation
Ten devices, [(D•)+1, (D•)], [(D•)+3, (D•)+2], [(D•)+5, (D•)+4], [(D•)+7, (D•)+6] and [(D•)+9, (D•)+8], are occupied.

14.3 FNC 62 – ABSD / Absolute Drum Sequencer

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction creates many output patterns corresponding to the current value of a counter.

1. Instruction format

D	FNC 62 ABSD	

16-bit Instruction	Mnemonic	Operation Condition
9 steps	ABSD	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DABSD	Continuous Operation

2. Set data

Operand type	Description	Data type
S1•	Head device number storing the data table (with rising and faling point data)	16- or 32-bit binary
S2•	Counter number for monitoring the current value compared with the data table	16- or 32-bit binary
D•	Head bit device number to be output	Bit
n	Number of lines in the table and the number of output bit devices [1 ≤ n ≤ 64]	16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
S2•													✓						✓					
D•		✓	✓			✓	▲												✓					
n																				✓	✓			

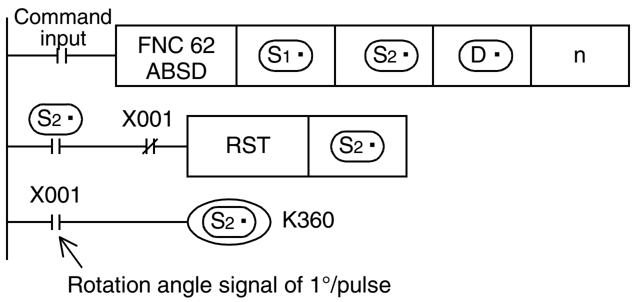
▲: "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

1. 16-bit operation (ABSD)

In this example, outputs are controlled to ON or OFF by one rotation (0 to 360° using the rotation angle signal of 1°/pulse).

The current value S2• of the counter is compared with the data table with "n" lines starting from S1• (which occupies "n" lines × 2 devices), and consecutive "n" outputs starting from D• are controlled to ON or OFF during one rotation.



1) Write the following data to S1· to S1·+2n+1 in advance by a transfer instruction:

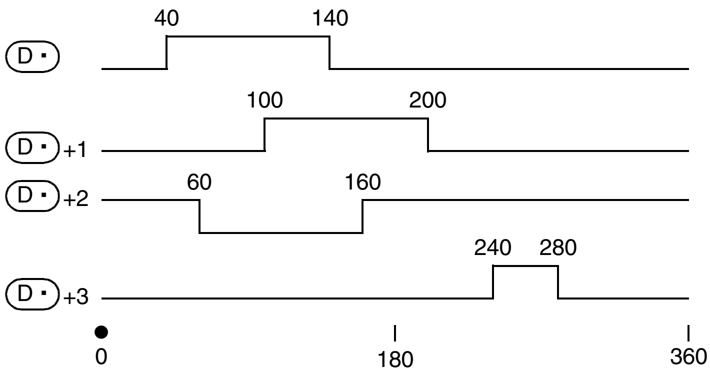
Rising point		Falling point		Target output
	Data value (example)		Data value (example)	
S1·	40	S1· + 1	140	D·
S1· + 2	100	S1· + 3	200	D· + 1
S1· + 4	160	S1· + 5	60	D· + 2
S1· + 6	240	S1· + 7	280	D· + 3
⋮	—	⋮	—	⋮
S1· + 2n		S1· + 2n + 1		D· + n – 1

For example, store the 16-bit rising point data to an even device number devices, and store the 16-bit falling data to an odd device number devices.

2) Output pattern

When the command input is set to ON, "n" points starting from D· change as shown below.

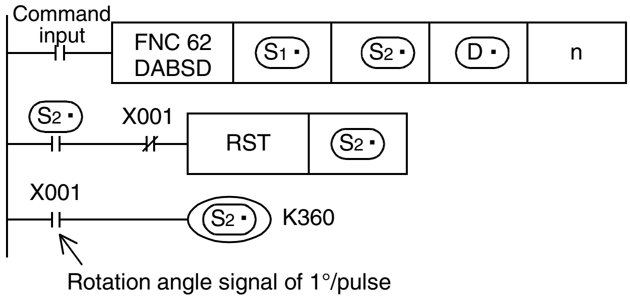
Each rising point/falling point can be changed respectively by overwriting the data in S1· to S1·+n × 2.



2. 32-bit operation (DABSD)

In this example, outputs are controlled to ON or OFF by one rotation (0 to 360° using the rotation angle signal of 1°/pulse).

The present value S2· of the counter is compared with the data table having "n" lines starting from [S1·+1, S1·] (which occupies "n" lines × 4 devices), and consecutive "n" outputs starting from D· are controlled to ON or OFF during one rotation.

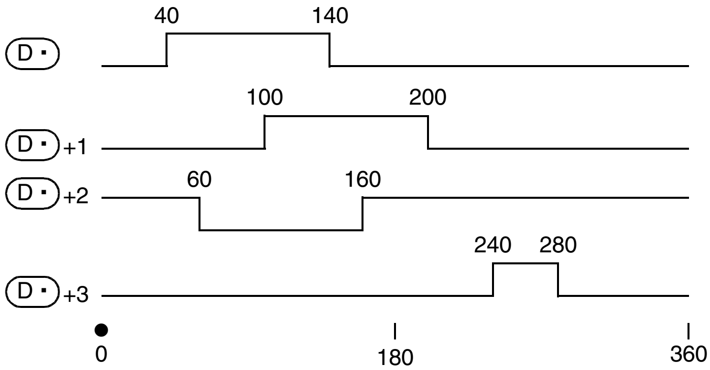


1) Write the following data to [S1•, S1•+1] to [S1•+4n+2, S1•+4n+3] in advance using a transfer instruction:

Rising point	Data value (example)	Falling point	Data value (example)	Target output
[S1• + 1, S1•]	40	[S1• + 3, S1• + 2]	140	D•
[S1• + 5, S1• + 4]	100	[S1• + 7, S1• + 6]	200	D• +1
[S1• + 9, S1• + 8]	160	[S1• + 11, S1• + 10]	60	D• +2
[S1• + 13, S1• + 12]	240	[S1• + 15, S1• + 14]	280	D• +3
⋮	—	⋮	—	⋮
[S1• + 4n + 1, S1• + 4n]		[S1• + 4n + 3, S1• + 4n + 2]		D• + n – 1

For example, store the 32-bit rising point data to devices having an even device number, and store the 32-bit falling data to devices having an odd device number.

When the command input is set to ON, "n" points starting from D• change as shown below.
Each rising point/falling point can be changed respectively by overwriting the data in [S1•+1, S1•] to [S1•+(n×2)+3, S1•+(n×2)+2].



Cautions

1. Specifying a high speed counter (C235 to C255)

In DABSD instruction, a high seed counter can be specified as S2•.
In this case, however, the output pattern contains response delay caused by the scan cycle with regard to the current value of a counter.
When high responsibility is required, use the table high speed comparison function offered by HSZ instruction, or use HSCT instruction.

2. When specifying digits of a bit device as S1•

1) Device number
Specify a multiple of 16 (0, 16, 32, 64 ...).
2) Number of digits
- In ABSD instruction (16-bit operation): Only K4 is available.
- In DABSD instruction (32-bit operation): Only K8 is available.

3. Other cautions

- The value "n" determines the number of target outputs (1 ≤ n ≤ 64).
- Even if the command input is set to OFF, the ON/OFF status of outputs does not change.

14.4 FNC 63 – INCD / Incremental Drum Sequencer

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction creates many output patterns using a pair of counters.

1. Instruction format

FNC 63 INCD

16-bit Instruction	Mnemonic	Operation Condition
9 steps	INCD	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand type	Description	Data type
S1•	Head word device number storing the set value	16-bit binary
S2•	Head number of counters whose current value is monitored	16-bit binary
D•	Head bit device number to be output	Bit
n	Number of output bit devices [1 ≤ n ≤ 64]	16-bit binary

3. Applicable devices

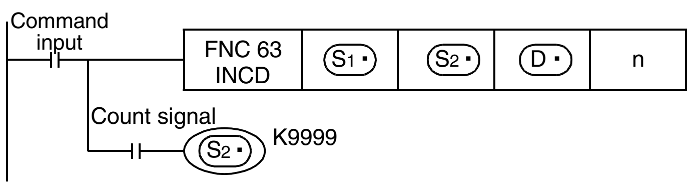
Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
S2•													✓						✓					
D•		✓	✓			✓	▲												✓					
n																				✓	✓			

▲: "D□.b" cannot be indexed with index registers (V and Z).

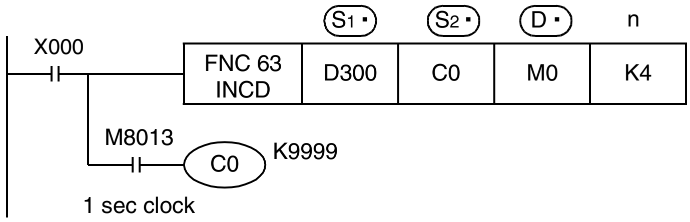
Explanation of function and operation

1. 16-bit operation (INCD)

The current value S2• of a counter is compared with the data table having "n" lines starting from S1• (which occupies "n" lines x 1 device). When S2• is equivalent to the table data, the current output is reset, and the next output is set to ON. In this way, the ON/OFF status of specified outputs is controlled in turn.

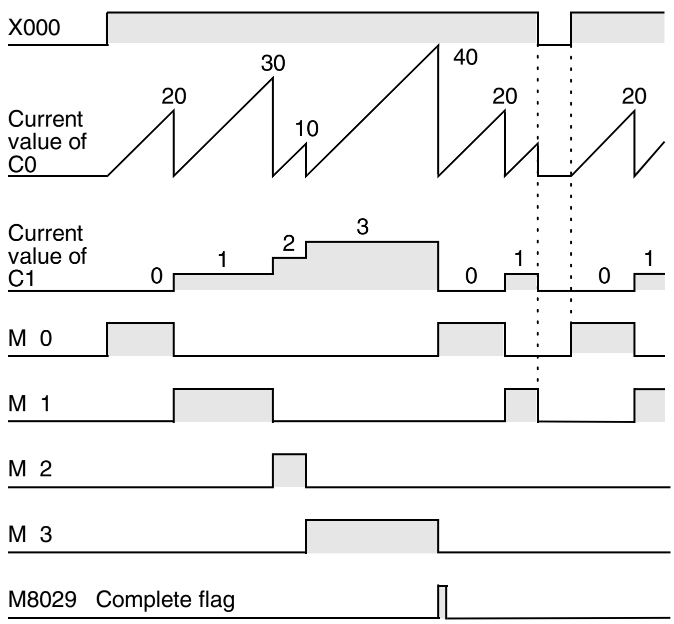


Operation



1) Timing chart
   Suppose that the following data is written in advance by a transfer instruction:

Device storing data		Output	
	Data value (example)		Example
S1•	D300 = 20	D•	M0
S1• + 1	D301 = 30	D• + 1	M1
S1• + 2	D302 = 10	D• + 2	M2
S1• + 3	D303 = 40	D• + 3	M3
⋮	⋮	⋮	⋮
S1• + n – 1	—	D• + n – 1	—



2) When the command contact turns ON, the output M0 turns ON.
3) When the current value of C0 reaches the comparison value D300, the output M0 is reset. "1" is added to the count value of the process counter C1, and the current value of the counter C0 is reset.
4) The next output M1 turns ON.
5) When the current value of C0 reaches the comparison value D301, the output M1 is reset. "1" is added to the count value of the process counter C1, and the current value of the counter C0 is reset.
6) The current value is compared for up to "n (K4)" outputs in the same way ($1 \leq n \leq 64$).
7) When the final process specified by "n" is finished, the execution complete flag M8029 turns ON and remains ON for one operation cycle.
   M8029 is used for many instructions as the instruction execution complete flag. Use M8029 as a contact just after a corresponding instruction.
8) The program execution returns to the beginning, and outputs are repeated.

Caution

1. When specifying digits of a bit device as S1•

As a device number, specify a multiple of 16 (0, 16, 32, 64 ...).

14.5 FNC 64 – TTMR / Teaching Timer

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction measures the period of time in which TTMR instruction is ON.
Use this instruction to adjust the set value of a timer by a pushbutton switch.

1. Instruction format

FNC 64 TTMR

16-bit Instruction	Mnemonic	Operation Condition
5 steps	TTMR	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand type	Description	Data type
D·	Device number storing the teaching data	16-bit binary
n	Magnification by which the teaching data is multiplied [K0 to K2/H0 to H2]	16-bit binary

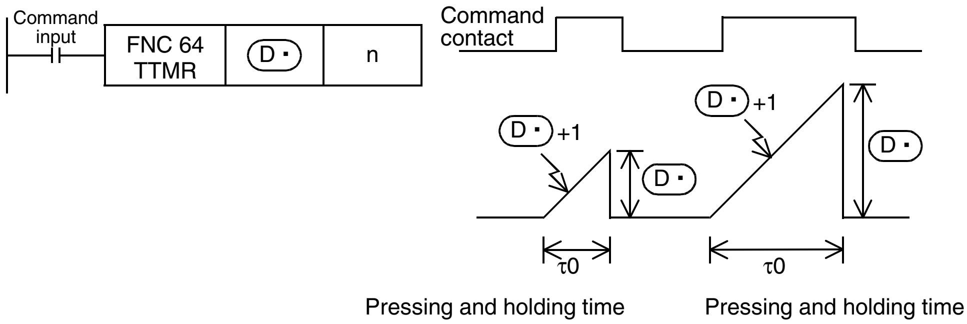
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
D·														✓	✓				✓					
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (TTMR)

The period of time to press and hold the command input (pushbutton switch) is measured in 1-second units, multiplied by the magnification (10^n), and then transferred to D·.



The table below shows the actual value indicated by D· depending on the magnification n and the pressing and holding time t0 (unit: 1 sec).

n	Magnification	D·
K0	τ0	D· × 1
K1	10τ0	D· × 10
K2	100τ0	D· × 100

Related instruction

There is a handy instruction as follows:

Instruction	Description
HOUR(FNC169)	Measures the input contact ON time in 1-hour units, and outputs alarm when the measurement result reaches a specified value.

Cautions

1. When the command contact turns OFF

The current value [D·+1] of the pressing and holding time is reset, and the teaching time D· will not change any more.

2. Number of occupied devices

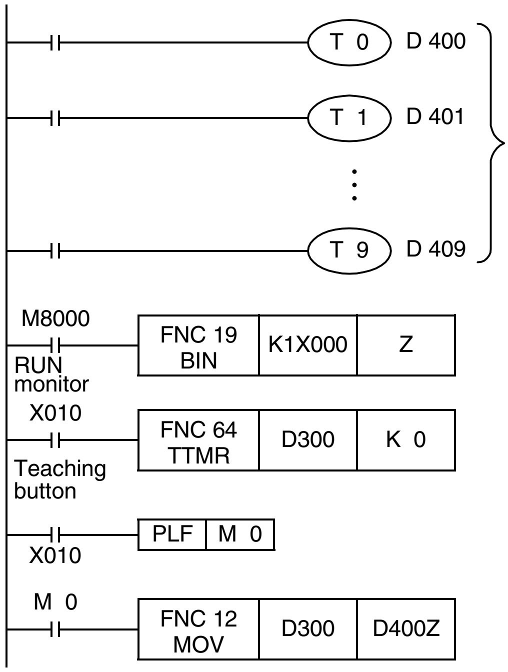
Two devices are occupied from a device specified as the teaching time D·.
Make sure that these devices are not used in other controls for the machine.

- D·: Teaching time
- D·+1: Current value of the pressing and holding time

Program example

1. Writing the teaching time to 10 types of data registers

Suppose that the set value is written to D400 to D409 in advance.



10 timers to be set
Because the timers T0 to T9 are the 100 ms type, the actual operating time (sec) is 1/10 of the teaching data.

Selecting a timer by a digital switch
An input to the 1-digit digital switch connected to X000 to X003 is converted into the binary format, and transferred to Z.

Measuring the teaching time
The time (sec) in which X010 is pressed and held is stored to D300.

Recovery of the teaching time is detected.

Writing the set value of a timer
The teaching time (D300) is transferred to the register D400Z for setting a timer selected by the digital switch.

14.6 FNC 65 – STMR / Special Timer

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction can easily make off-delay timers, one-shot timers and flicker timers.

1. Instruction format

FNC 65 STMR

16-bit Instruction	Mnemonic	Operation Condition
7 steps	STMR	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand type	Description	Data type
(S·)	Used timer number [T0 to T199 (100 ms timer)]	16-bit binary
m	Set value of the timer [1 to 32,767]	16-bit binary
(D·)	Head bit number to which the set value is output (Four devices are occupied.)	Bit

3. Applicable devices

Operand Type	Bit Devices: System User							Word Devices: Digit Specification				Word Devices: System User				Special Unit	Index			Others: Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)												✓							✓					
m														✓	✓					✓	✓			
(D·)		✓	✓			✓	▲												✓					

▲: "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

1. 16-bit operation (STMR)

The value specified in "m" is handled as the set value of a timer specified in (S·), and output to four devices starting from (D·).

Create a proper program according to the purpose while referring to the example shown below.

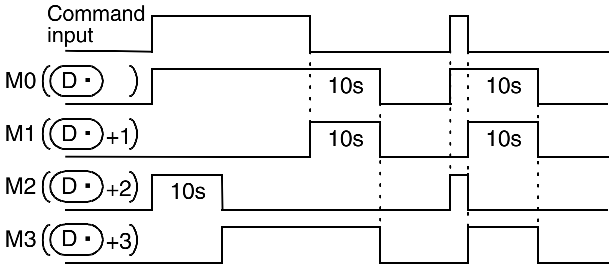
Command input	FNC 65 STMR	(S·)	m	(D·)

Off-delay timer and one-shot timer

When T10 is set to (S·), and M0 is set to (D·)

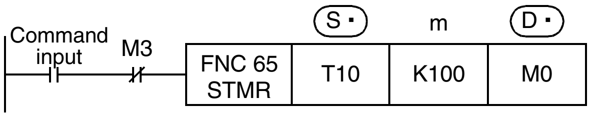
Command input	FNC 65 STMR	T10	K100	M0

- M0 [(D·)] : Off-delay timer which turns OFF with delay of the timer set value after the command contact turned OFF
- M1 [(D·)+1] : One-shot timer which turns ON after the command contact turned OFF from ON, and turns OFF after the timer set value
- M2 [(D·)+2] : Occupied, and can be used for flicker.
- M3 [(D·)+3] : Occupied.

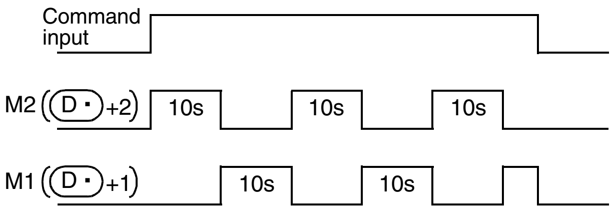


Flicker

In the program shown below which turns OFF STMR instruction at the NC contact of D·+3, flicker is output to D·+1 and D·+2.
D· and D·+3 are occupied.



- M0 [D·] : Occupied (, and can be used for off-delay timer). (Refer to the previous page.)
- M1 [D·+1] : Flicker (NO contact) which turns ON and OFF repeatedly at the interval of timer set value
- M2 [D·+2] : Flicker (NC contact) which turns ON and OFF repeatedly at the interval of timer set value
- M3 [D·+3] : Occupied.



Cautions

1. Handling of a specified timer

The timer number specified in this instruction cannot be used in other general circuits (such as OUT instruction).
If the timer number is used in other general circuits, the timer malfunctions.

2. Number of occupied devices

Four devices are occupied from a device specified in D·.
Make sure that these devices are not used in other controls for the machine.

Device	Function	
	Off-delay timer One-shot timer	Flicker
D·	Off-delay timer	Occupied
D· + 1	One-shot timer	Flicker (NO contact)
D· + 2	Occupied	Flicker (NC contact)
D· + 3	Occupied	Flicker (NC contact)

3. When the command contact is set to OFF

D·, D·+1 and D·+3 will turn OFF after the set time. D·+2 and the timer S· are immediately reset.

14.7 FNC 66 – ALT / Alternate State

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction alternates a bit device (from ON to OFF or from OFF to ON) when the input turns ON.

1. Instruction format

FNC 66 ALT P

16-bit Instruction	Mnemonic	Operation Condition
3 steps	ALT	Continuous Operation
	ALTP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
(D·)	Bit device number whose output is alternated	Bit

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index				Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify		K	H	E	"□"	P
(D·)		✓	✓			✓	▲												✓						

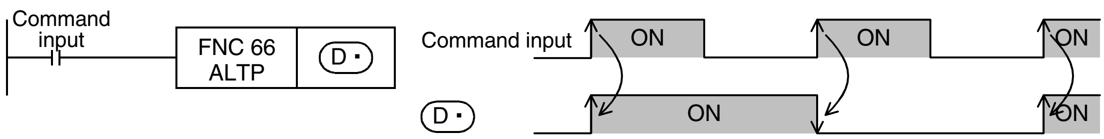
▲: "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

1. 16-bit operation (ALT and ALTP)

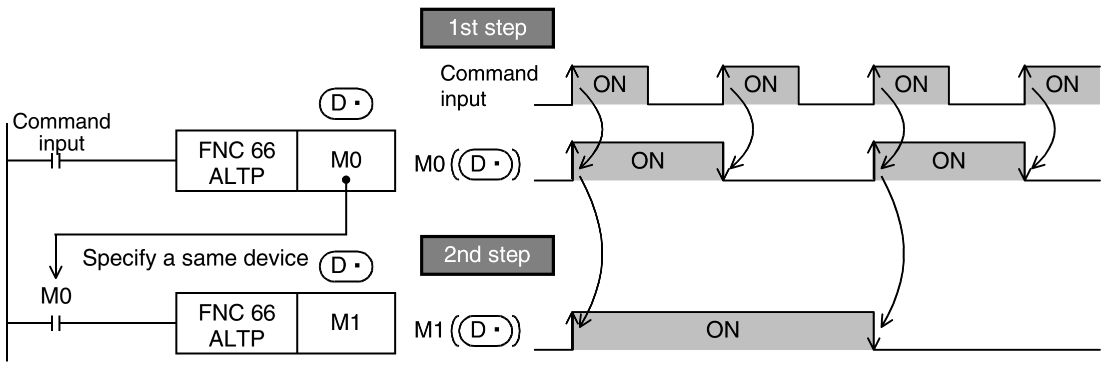
Alternating output (1-step)

Every time the command input turns ON from OFF, a bit device specified in (D·) is alternated (from ON to OFF or from OFF to ON).



Dividing output (by 2-step alternating output)

Multi-step dividing outputs are achieved by combination of two or more ALTP instructions.



Caution

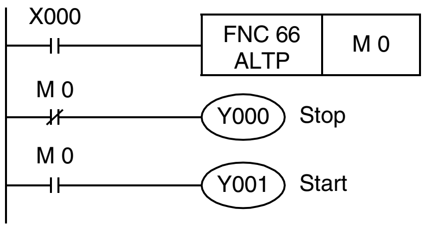
1. When using (continuous operation type) ALT instruction

- When ALT instruction is used, a specified bit device is alternated in every operation cycle.
  To alternate a specified device by turning the command ON or OFF, use the (pulse operation type) ALTP instruction, or use a pulse operation type command contact such as LDP.

Program examples

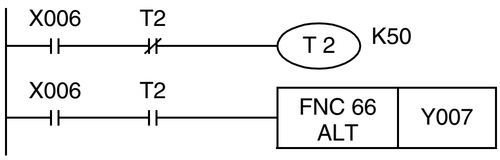
1. Start and stop by one input

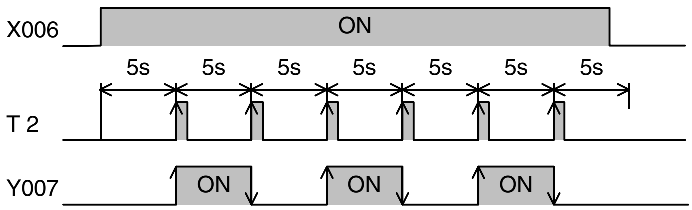
1) When the pushbutton switch X000 is pressed, the start output Y001 is set to ON.
2) When the pushbutton switch X000 is pressed again, the stop output Y000 is set to ON.



2. Flicker operation

1) When the input X006 is set to ON, the contact of the timer T2 turns ON instantaneously every 5 seconds.
2) Every time the contact of T2 turns ON, the output Y007 is set to ON or OFF alternately.

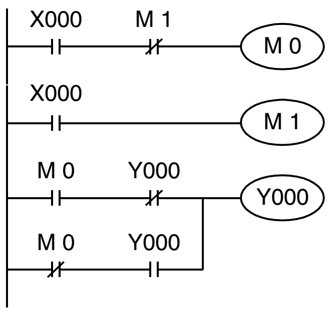




3. Alternating output operation using auxiliary relays (M) (operation equivalent to ALT instruction)

The circuit below is provided as an example of alternating operation using basic instructions and auxiliary relays (M) which is equivalent to ALT instruction.

1) When X000 is set to ON, M0 turns ON and remains ON for only one operation cycle.
2) When M0 turns ON for the first time, Y000 is latched. When M0 turns ON the second time, Y000 becomes unlatched.



14.8 FNC 67 – RAMP / Ramp Variable Value

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction obtains the data which changes between the start value (initial value) and the end value (target value) over the specified "n" times.

1. Instruction format

FNC 67 RAMP

16-bit Instruction	Mnemonic	Operation Condition
9 steps	RAMP	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	

2. Set data

Operand type	Description	Data type
S1•	Device number storing the initial value of ramp	16-bit binary
S2•	Device number storing the target value of ramp	16-bit binary
D•	Device number storing the current value of ramp	16-bit binary
n	Ramp transfer time (scan)	16-bit binary

3. Applicable devices

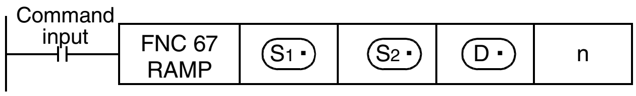
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•														✓	✓				✓					
S2•														✓	✓				✓					
D•														✓	✓				✓					
n														✓	✓					✓	✓			

Explanation of function and operation

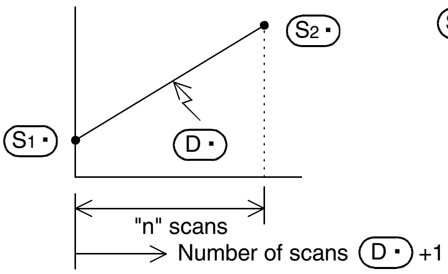
1. 16-bit operation (RAMP)

When the start value S1• and the end value S2• have been specified and the command input is set to ON, the value obtained by adding a value divided equally by "n" times to S1• in every operation cycle is stored to D•.

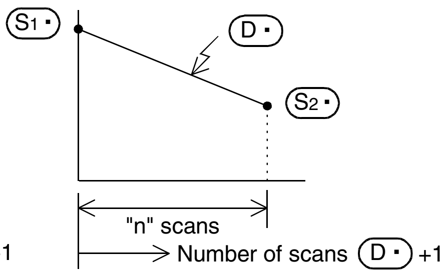
By combining this instruction and an analog output, the cushion start/stop command can be output.



1) In the case of " S1• < S2• "



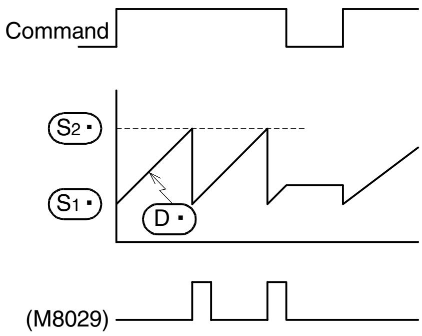
2) In the case of " S1• > S2• "



- The number of scans ("0" to "n") is stored in D•+1.
- The time from start to the end value is the operation cycle multiplied by "n" times.

11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FMC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

- If the command input is set to OFF in the middle of operation, execution is paused. (The present data value stored in D· is held, and the number of scans stored in D·+1 is cleared.) When the command input is set to ON again, D· is cleared, and the operation is started from S1·.
- After transfer is completed, the instruction execution complete flag M8029 turns ON, and the D· value is returned to the S1· value.

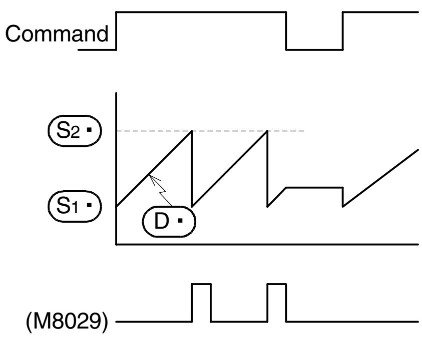


- When acquiring the operation result at a constant time interval (constant scan mode)
  Write a prescribed scan time (which is longer than the actual scan time) to D8039 and set M8039 to ON to select the constant scan mode in the PLC.
  For example, when "20 ms" is written to D8039 and "n" is set to 100, the D· value will change from S1· to S2· in 20 seconds.

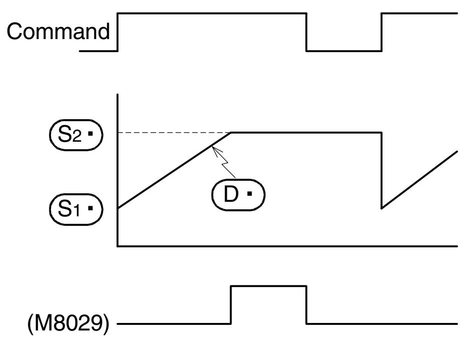
2. Operation of the mode flag (M8026)

Depending on the ON/OFF status of the mode flag M8026, the contents of D·+1 are changed as follows:

1) When M8026 is OFF



2) When M8026 is ON



Related devices

→ **For the instruction execution complete flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8029	Instruction execution complete	Turns ON when D· becomes equivalent to S2· after "n" operation cycles.
M8026*1	RAMP mode	Refer to the operation of the mode flag M8026 described above.

*1. Cleared when the RAMP mode is changed from RUN to STOP.

Caution

1. When specifying a latched (battery backed) type device as D·

When setting PLC to the RUN mode while the command input is ON, clear D· in advance.

14.9 FNC 68 – ROTC / Rotary Table Control

Outline

FX3UC Ver.1.00 | FX3U Ver.2.20

This instruction is suitable for efficient control of the rotary table for putting/taking a product into/out of the rotary table.

1. Instruction format

FNC 68
ROTC

16-bit Instruction	Mnemonic	Operation Condition
9 steps	ROTC	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand type	Description	Data type
S•	Data register for counting	16-bit binary
m1	Number of divisions	16-bit binary
m2	Number of low-speed sections	16-bit binary
D•	Head bit device number to be driven	16-bit binary

3. Applicable devices

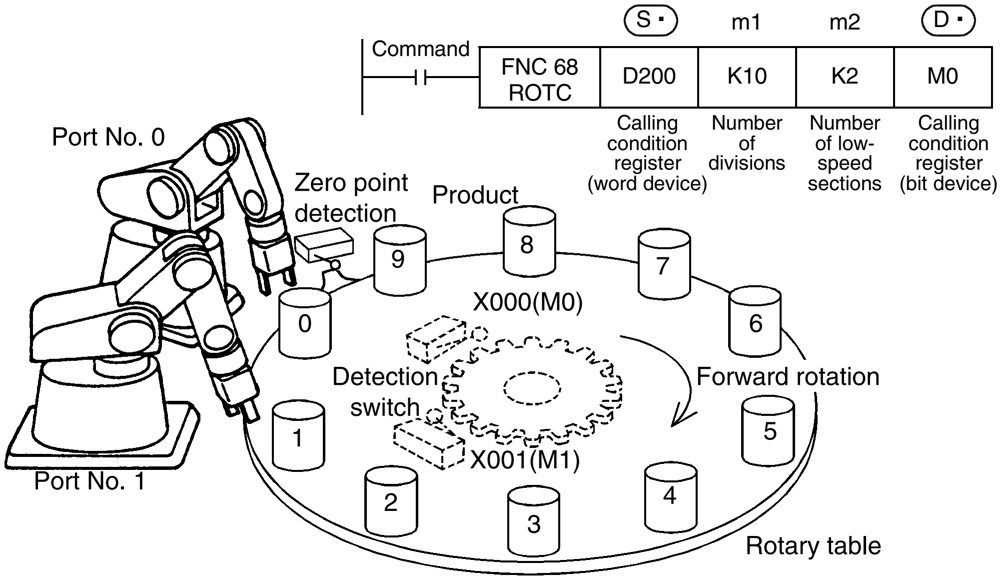
Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•														✓	✓				✓					
m1																				✓	✓			
m2																				✓	✓			
D•		✓	✓			✓	▲												✓					

▲: "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

1. 16-bit operation (ROTC)

The table rotation is controlled by "m2", (S·) and (D·) so that a product can be efficiently put into or taken out of the rotary table divided into "m1" (=10) sections as shown in the figure below.



1) Register (word device) specifying the calling condition (S·)

(S·)	Works as a register for counting.	Set them in advance using a transfer instruction.
(S·) + 1	Sets the port No. to be called.	
(S·) + 2	Sets the product No. to be called.	

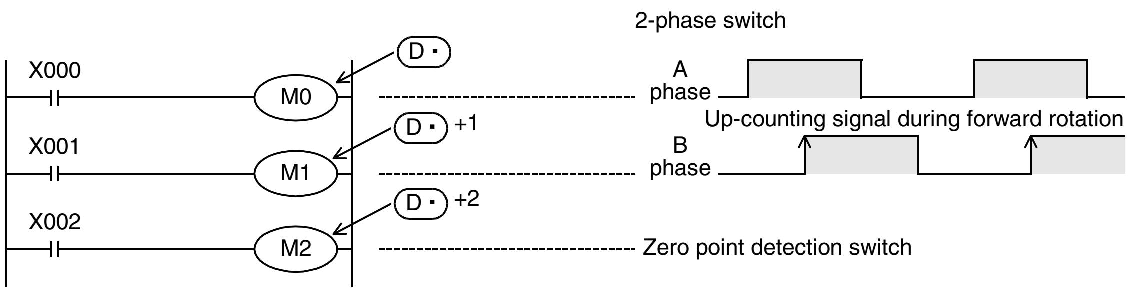
2) Register (bit device) specifying the calling condition (D·)

(D·) :	A phase signal	Construct an internal contact circuit in advance which is driven by the input signal (X)
(D·) + 1	B phase signal	
(D·) + 2	Zero point detection signal	
(D·) + 3	Forward rotation at high speed	
(D·) + 4	Forward rotation at low speed	
(D·) + 5	Stop	
(D·) + 6	Backward rotation at low speed	
(D·) + 7	Backward rotation at high speed	

Operation conditions

The conditions required to use this instruction are as shown in the example below.

1) Rotation detection signal: X → D·
   - Provide a 2-phase switch (X000 and X001) for detecting the rotation direction (forward or backward) of the table and the switch X002 which turns ON when the product No. 0 reaches the port No. 0.
   - Create the sequence program shown below.



X000 to X002 are replaced with internal contacts of D· to D·+2.
An arbitrary head device number can be specified by X or D·.

2) Specification of a register for counting: S·
   The counter S· detects which number of product is located at the port No. 0.
3) Registers specifying the calling condition: S·+1 and S·+2
   a) Set the port No. to be called in S·+1.
   b) Set the product No. to be called in S·+2.
4) Number of divisions m1 and number of low-speed sections m2
   Specify the number of divisions m1 of the table, and number of low-speed sections m2.

When the above conditions are specified, forward/backward rotation and high speed/low speed/stop are output to D·+3 to D·+7 specified by the head device D·.

Cautions

1. Operations caused by the command input ON/OFF status

- When the command input is set to ON and this instruction is executed, the result will be automatically output to D·+3 to D·+7.
- When the command input is set to OFF, D·+3 to D·+7 are set to OFF accordingly.

2. Multiple activation of the rotation detection signal (D· to D·+2) in one division

For example, when the rotation detection signal (D· to D·+2) is activated 10 times in one division, set a value multiplied by "10" to each division, port No. to be called and product No. to be called.
As a result, an intermediate value of the division number can be set to a low-speed section.

3. Zero point detection signal D·

When the zero point detection signal (M2) turns ON while the command input is ON, the contents of the register for counting S· are cleared to "0".
This clear operation should be executed before starting the operation.

11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FMC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

14.10 FNC 69 – SORT / SORT Tabulated Data

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction sorts a data table consisting of data (lines) and group data (columns) based on a specified group data (column) sorted by line in ascending order. This instruction stores the group data (columns) in serial devices.
On the other hand, SORT2 (FNC149) instruction stores the data (lines) in serial devices facilitating the addition of data (lines), and sorts a table in either ascending or descending order.

→ **For SORT2 (FNC149) instruction, refer to Section 19.7.**

1. Instruction format

FNC 69 SORT	16-bit Instruction	Mnemonic	Operation Condition	32-bit Instruction	Mnemonic	Operation Condition
	11 steps	SORT	Continuous Operation	–		

2. Set data

Operand type	Description	Data type
S	Head device number storing the data table [which occupies m1 × m2 points]	16-bit binary
m1	Number of data (lines) [1 to 32]	
m2	Number of group data (columns) [1 to 6]	
D	Head device number storing the operation result [which occupies m1 × m2 points]	
n	Column number of the group data (column) used as the basis of sorting [1 to m2]	

3. Applicable devices

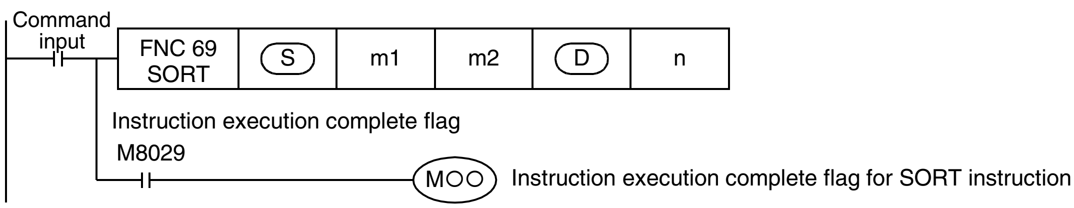
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S														✓	✓									
m1																				✓	✓			
m2																				✓	✓			
D														✓	✓									
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (SORT)

In the data table (sorting source) having (m1 × m2) points from (S), data lines are sorted in the ascending order based on the group data in the column No. "n", and the result is stored in the data table (sorting result) having (m1 × m2) points from (D).

→ **For operation examples, refer to the next page.**



- The data table configuration is explained in an example in which the sorting source data table has 3 lines and 4 columns (m1 = K3, m2 = K4). For the sorting result data table, understand (S) as (D).

Column No. / Line No.		Number of groups (m2 = K4)			
		1	2	3	4
		Control number	Height	Weight	Age
Number of data (m1 = 3)	1	(S)	(S)+3	(S)+6	(S)+9
	2	(S)+1	(S)+4	(S)+7	(S)+10
	3	(S)+2	(S)+5	(S)+8	(S)+11

- When the command input turns ON, data sorting is started. Data sorting is completed after "m1" scans, and the instruction execution complete flag M8029 is set to ON.

→ **For the instruction execution complete flag use method, refer to Subsection 6.5.2.**

2. Operation examples

When the instruction is executed with "n = K2 (column No. 2)" and "n = K3 (column No. 3) for the following sorting source data, the operations shown below are acquired.
It is recommended to put a serial number such as a control number in the first column so that the original line number can be estimated based on the contents.

Sorting source data

Column No. / Line No.		Number of groups (m2 = K4)			
		1	2	3	4
		Control number	Height	Weight	Age
Number of data (m1 = 5)	1	(S)	(S) + 5	(S) + 10	(S) + 15
		1	150	45	20
	2	(S) + 1	(S) + 6	(S) + 11	(S) + 16
		2	180	50	40
	3	(S) + 2	(S) + 7	(S) + 12	(S) + 17
		3	160	70	30
	4	(S) + 3	(S) + 8	(S) + 13	(S) + 18
		4	100	20	8
	5	(S) + 4	(S) + 9	(S) + 14	(S) + 19
		5	150	50	45

1) Sorting result when the instruction is executed with "n = K2 (column No. 2)"

Column No.	1	2	3	4
Line No.	Control number	Height	Weight	Age
1	(D)	(D) + 5	(D) + 10	(D) +15
	4	100	20	8
2	(D) + 1	(D) + 6	(D) + 11	(D) + 16
	1	150	45	20
3	(D) + 2	(D) + 7	(D) + 12	(D) + 17
	5	150	50	45
4	(D) + 3	(D) + 8	(D) + 13	(D) + 18
	3	160	70	30
5	(D) + 4	(D) + 9	(D) + 14	(D) + 19
	2	180	50	40

2) Sorting result when the instruction is executed with "n = K3 (column No. 3)"

Column No. / Line No.	1 Control number	2 Height	3 Weight	4 Age
1	D	D + 5	D + 10	D + 15
	4	100	20	8
2	D + 1	D + 6	D + 11	D + 16
	1	150	45	20
3	D + 2	D + 7	D + 12	D + 17
	2	180	50	40
4	D + 3	D + 8	D + 13	D + 18
	5	150	50	45
5	D + 4	D + 9	D + 14	D + 19
	3	160	70	30

Related device

→ **For the instruction execution complete flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8029	Instruction execution complete	Turns ON when sorting is completed.

Cautions

- Do not change the contents of operands and data while the instruction is executed.
- Before executing the instruction again, set the command input to OFF.
- Limitation in the number of instructions
  Only one instruction can be used in a program.
- When the same device is specified in S and D
  The source data is overwritten by the data acquired by sorting.
  Take special care so that the contents of S are not changed until execution is completed.

15. External FX I/O Device – FNC 70 to FNC 79

FNC 70 to FNC 79 provide instructions to receive data from and send data to external devices mainly using inputs and outputs in PLCs.
Because these instructions easily achieve complicated controls with a minimum required sequence program and external wiring, they are similar to handy instructions described in the preceding chapter.
FROM and TO instructions essential for controlling special units and special blocks are included in this group. (In FX3U and FX3UC PLCs, transfer can be executed also by MOV instruction.)

FNC No.	Mnemonic	Symbol	Function	Reference
70	TKY	TKY S D1 D2	Ten Key Input	Section 15.1
71	HKY	HKY S D1 D2 D3	Hexadecimal Input	Section 15.2
72	DSW	DSW S D1 D2 n	Digital switch (thumbwheel input)	Section 15.3
73	SEGD	SEGD S D	Seven Segment Decoder	Section 15.4
74	SEGL	SEGL S D n	Seven Segment With Latch	Section 15.5
75	ARWS	ARWS S D1 D2 n	Arrow Switch	Section 15.6
76	ASC	ASC S D	ASCII code data input	Section 15.7
77	PR	PR S D	Print (ASCII Code)	Section 15.8
78	FROM	FROM m1 m2 D n	Read From A Special Function Block	Section 15.9
79	TO	TO m1 m2 S n	Write To A Special Function Block	Section 15.10

15.1 FNC 70 – TKY / Ten Key Input

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction sets data to timers and counters through inputs of the ten keys from "0" to "9".

1. Instruction format

	FNC 70 TKY	
D		

16-bit Instruction	Mnemonic	Operation Condition
7 steps	TKY	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DTKY	Continuous Operation

2. Set data

Operand Type	Description	Data Type
(S·)	Head bit device number from which one of the ten keys is input [10 devices are occupied]	Bit
(D1·)	Word device number storing the data	16- or 32-bit binary
(D2·)	Head bit device number storing the key pressing information [11 devices are occupied]	Bit

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)	✓	✓	✓			✓	▲												✓					
(D1·)									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
(D2·)		✓	✓			✓	▲												✓					

▲: "D□.b" cannot be indexed with index registers (V and Z).

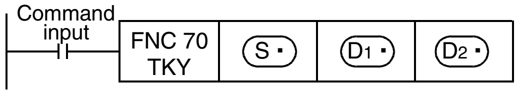
Explanation of function and operation

1. 16-bit operation (TKY)

(D1·) stores a numeric value input from (S·) to (S·)+9 connected to the ten keys. Output informations for key pressing and key sensing are output to (D2·) to (D2·)+10.

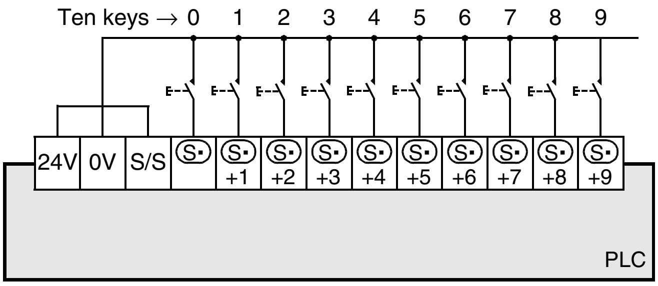
1) Input numeric value (D1·)
   - When an input value is larger than "9999", it overflows from the most significant digit.
   - An input numeric value is stored in the binary format.
   - When the ten keys are pressed in the order "[1] → [2] → [3] → [4]" in the figure shown on the next page, "2130" is stored in (D1·).

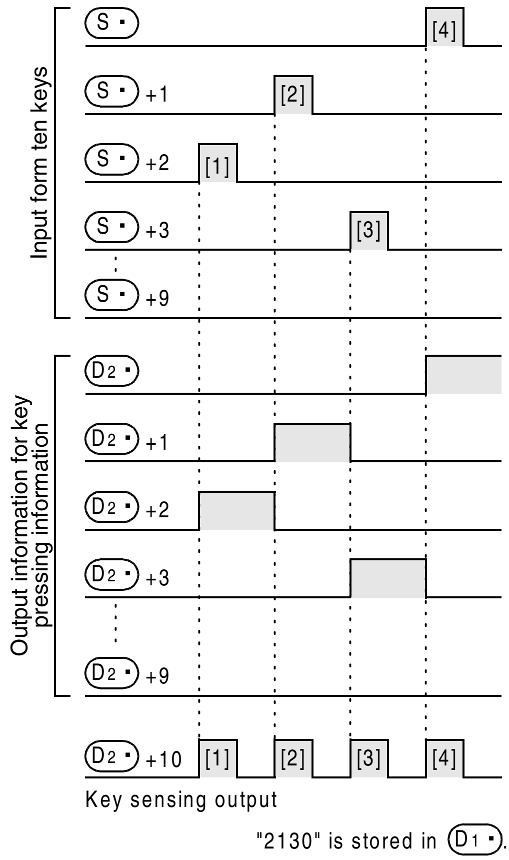
2) Key pressing information [D2· to D2·+10]
   - For the key pressing information, D2· to D2·+9 turn ON or OFF according to the pressed keys.
   - For the key sensing output, D2·+10 turns ON when any key is pressed.



The figure below shows an example of FX3U PLC (sink input).
For wiring details, refer to the following manual.

→ **FX3U Hardware Edition**



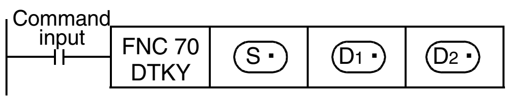


2. 32-bit operation (DTKY)

[D1·+1, D1·] store a numeric value input from S· to S·+9 connected to the ten keys. Output informations for key pressing and key sensing are output to D2· to D2·+10.

1) Input numeric value [D1·]
   - When an input value is larger than "99,999,999", it overflows from the most significant digit.
   - An input numeric value is stored in the binary format.

2) Key pressing information [D2· to D2·+10]
   - For the key pressing information, D2· to D2·+9 turn ON or OFF according to the pressed keys.
   - For the key sensing output, D2·+10 turns ON when any key is pressed.



For the ten-key connection example and key pressing information, refer to the 16-bit operation (TKY) shown above.

Cautions

1. When two or more keys are pressed at the same time

In such a case, only the first key pressed is valid.

2. When the command contact turns OFF

Though the contents of D1· do not change, all of D2· to D2·+10 turn OFF.

11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FNC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

3. Number of occupied device

1) Ten bit devices are occupied from (S•) for connecting the ten keys.
   Because these devices are occupied even if the ten keys are not connected, they cannot be used for any other purpose.
2) Eleven bit devices are occupied from (D2•) for outputting the key pressing information.
   Make sure that these devices are not used in other controls for the machine.
   - (D2•) to (D2•)+9: Turn ON or OFF according to input of the ten keys "0" to "9".
   - (D2•)+10: Is ON while either one among "0" to "9" keys is pressed (key sensing output).

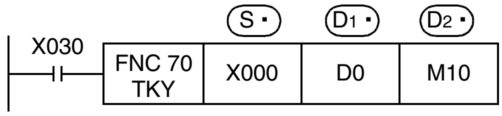
4. Limitation in the number the instruction

TKY or DTKY instruction can be used only once in a program.
When TKY and/or DTKY instruction should be used two or more times, use the indexing (V, Z) function.

Program example

In the program example shown below, the input X000 is set as the head bit device, and the ten keys "0" to "9" are connected.

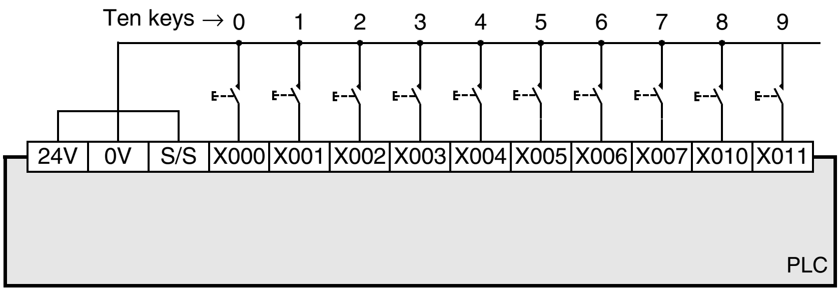
1. Program



2. Connection diagram

This connection diagram shows an example of FX3U PLC (sink input).
For wiring details, refer to the following manual.

→ **FX3U Hardware Edition**

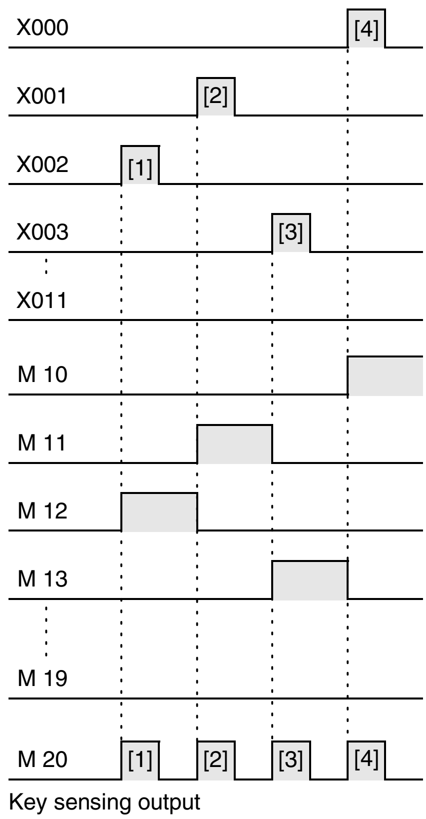


3. Timing chart

1) When the ten keys are pressed in the order "[1] → [2] → [3] → [4]" shown in the figure, "2130" is stored in (D0).
When an input value is larger than "9999", it overflows from the most significant digit.
(An input numeric value is stored in the binary format in D0).

2) When X002 is pressed, M12 turns ON and remains ON until another key is pressed. Other keys work in the same way.
In this way, M10 to M19 turn ON and OFF according to the inputs X000 to X011.

3) When pressing a key, the key sensing output M20 is ON only while it is pressed.



15.2 FNC 71 – HKY / Hexadecimal Input

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction multiplexes four X-devices and four Y-devices to allow for 16 key (0 to F) 4-digit (byte) input. Keys 0 to 9 stores numerical values, and keys A to F represent function keys.
When the extension function is set to ON, hexadecimal keys 0 to F all store their corresponding numerical values.

1. Instruction format

D	FNC 71 HKY

16-bit Instruction	Mnemonic	Operation Condition
9 steps	HKY	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DHKY	Continuous Operation

2. Set data

Operand Type	Description	Data Type
(S•)	Head X device number to be used (Four devices occupied.)	Bit
(D1•)	Head Y device number to be used (Four devices occupied.)	Bit
(D2•)	Device number storing the numerical input from the 16 keys	16- or 32-bit binary
(D3•)	Head bit device number storing the key pressing information (Eight devices are occupied.)	Bit

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S•)	✓																		✓					
(D1•)		✓																	✓					
(D2•)												✓	✓	✓	✓	✓	✓	✓	✓					
(D3•)		✓	✓			✓	▲												✓					

▲: "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

1. 16-bit operation (HKY)

signals [(S•) to (S•)+3] and [(D1•) to (D1•)+3] connected to the 16 key input (0 to F) are scanned.
When a key 0 to 9 is pressed, the corresponding numeric value is shifted into (D2•) from the least significant byte, and (D3•)+7 turns ON.
When a key A to F is pressed, the corresponding key press information bit [(D3•) to (D3•)+5] turns ON, and (D3•)+6 turns ON.

Command input

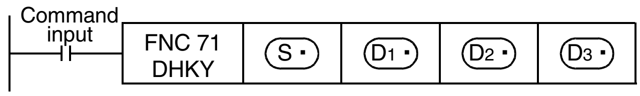
FNC 71 HKY	(S•)	(D1•)	(D2•)	(D3•)

1) Input of a numeric value through keys 0 to 9:
   - When an input value is larger than "9999", it overflows from the most significant digit.
   - The numeric value input is stored to D2• in binary.
   - The key sensing output D3• +7 turns ON when any key 0 to 9 is pressed.
2) Key pressing information for the keys A to F:
   - Six devices starting from D3• corresponding to keys A to F turn ON.
   - The key sensing output D3• +6 turns ON when any key A to F is pressed.

Key	Key pressing information	Key	Key pressing information
A	D3•	D	D3• +3
B	D3• +1	E	D3• +4
C	D3• +2	F	D3• +5

2. 32-bit operation (DHKY)

Signals [S• to S• +3] and [D1• to D1• +3] connected to the 16 key input (0 to F) are scanned.
When a key 0 to 9 is pressed, the corresponding numeric value is shifted into [D2• +1, D2•] from the least significant byte, and D3• +7 turns ON.
When a key A to F is pressed, the corresponding key press information bit [D3• to D3• +5] turns ON. and D3• +6 turns ON.



1) Input of a numeric value through keys 0 to 9:
   - When an input value is larger than "99,999,999", it overflows from the most significant digit.
   - The numeric value input is stored to [D2• +1, D2•] in binary.
   - The key sensing output D3• +7 turns ON when any key 0 to 9 is pressed.
2) Key pressing information for the keys A to F:
   Six devices starting from D3• corresponding to keys A to F turn ON.
   The key sensing output D3• +6 turns ON when any key A to F is pressed.

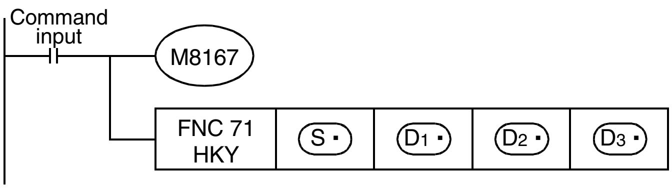
Extension function

When M8167 is set to ON making the extension function valid, the numerical values for keys 0 to F are stored in binary.
When the extension function is valid, the function and opearation are the same except for the following.

1. 16-bit operation (HKY)

Hexadecimal numerical value data input using keys 0 to F is shifted into D2• from the least significant byte.

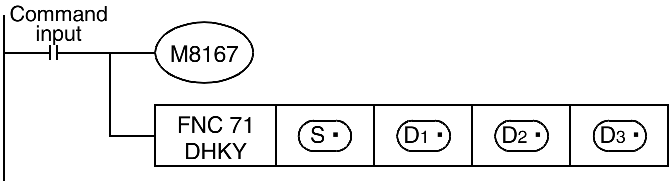
1) Input of a numeric value using keys 0 to F:
   - When the input value is larger than "FFFF", it overflows from the most significant digit.
   - Example:
     When "1 → 2 → 3 → B → F" is input, numerical value "23BF" is stored in D2• in binary.
     "1" overflows when "F" is input.



2. 32-bit operation (DHKY)

Hexadecimal numerical value data input using keys 0 to F is shifted into [D2•+1, D2•] from the least significant byte.

1) Input of a numeric value using keys 0 to F:
   - When the input value is larger than "FFFFFFFF", it overflows from the most significant digit.
   - Example:
     When "9 → 2 → 3 → B → F → A → F" is input, numerical value "923BFAF" is stored in [D2•+1, D2•] in binary.



Related devices

→ **For the instruction execution complete flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8167	Extension function flag	Turns ON/OFF the hexadecimal data handling function of HKY (FNC 71) instruction. OFF: Ten-keys and function keys ON: Hexadecimal keys
M8029	Instruction execution complete flag	OFF: Data is being output to D1• to D1•+3 or the instruction is not executed yet. ON: A cycle operation of outputting data to D1• to D1•+3 (scan of the keys 0 to F) is completed.

Cautions

1. Limitation in the number of instructions

HKY or DHKY instruction can be used only once in a program.
When TKY and/or DTKY instruction should be used two or more times, use the indexing (V, Z) function.

2. When two or more keys are pressed at the same time

In such a case, the first key pressed is valid.

3. When the command contact turns OFF

Though the contents of D2• do not change, D3• to D3•+7 turn OFF.

4. Number of devices occupied

1) Four devices are occupied from the head X device S• for connecting 16 keys.
2) Four devices are occupied from the head Y device D1• for connecting 16 keys.
3) Eight devices are occupied from the head device D3• for outputting the key pressing information.
   Make sure that these devices are not used by other machine controls.
   - D3• to D3•+5: Key pressing information for the keys A to F
   - D3•+6: Key sensing output for the keys A to F
   - D3•+7: Key sensing output for the keys 0 to 9

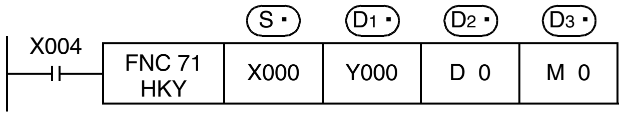
5. Key input receiving timing

HKY and DHKY instructions are executed in synchronization with the operation cycle of the PLC.
8 scan cycles are required to finish reading the keys.
To prevent key input receiving errors caused by the filter delay, utilize the "constant scan mode" and "timer interrupt" function.

6. Output format

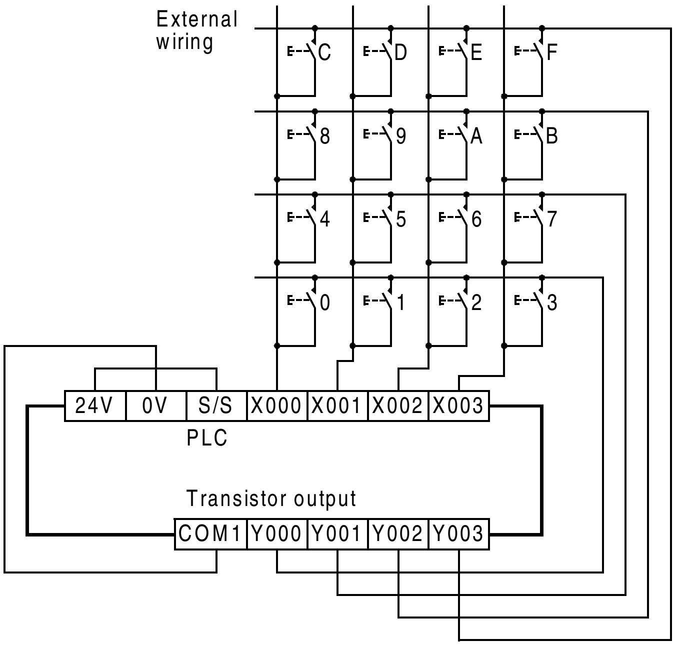
Use a transistor output type PLC.

Program example



The figure below shows an example of the FX3U series main unit (sink input/sink output).
For wiring details, refer to the following manual.

→ **FX3U Hardware Edition**



15.3 FNC 72 – DSW / Digital Switch (Thumbwheel Input)

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction reads the set value of digital switches.
This instruction can read a set of 4 digits (n = K1) or two sets of 4 digits (n = K2).

1. Instruction format

FNC 72
DSW

16-bit Instruction	Mnemonic	Operation Condition
9 steps	DSW	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	

2. Set data

Operand Type	Description	Data Type
(S•)	Head device (X) number connected to a digital switch (Four devices are occupied.)	Bit
(D1•)	Head device (Y) number to which the strobe signal is output (Four devices are occupied.)	Bit
(D2•)	Device number storing the numeric value of a digital switch ("n" devices are occupied.)	16-bit binary
n	Total number of 4-digit switch sets (4 digits/set) (n = 1 or 2)	16-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S•)	✓																		✓					
(D1•)		✓																	✓					
(D2•)												✓	✓	✓	✓	✓	✓	✓	✓					
n																				✓	✓			

Explanation of function and operation

1. 16-bit operation (DSW)

The value of each digital switch connected to (S•) is input in the time division method (in which the value is input from the 1st digit in turn by the output signal at the interval of 100 ms), and stored to (D2•).

Command input	FNC 72 DSW	(S•)	(D1•) Strobe signal	(D2•)	n

1) Data (D1•)

- A numeric value from 0 to 9999 (up to 4 digits) can be read.
- A numeric value is stored in the binary format.
- The first set is stored to (D2•), and the second set is stored to (D2•)+1.

2) Specification of the number of sets ("n")
   - When using one set of 4 digits [n = k1]
     A 4-digit BCD digital switch connected to S· to S·+3 is read in turn by the strobe signal D1· to D1·+3, and stored in the binary format to D2·.
   - When using two sets of 4 digits [n = k2]
     A 4-digit BCD digital switch connected to S· to S·+3 is read in turn by the strobe signal D1· to D1·+3, and stored in the binary format to D2·.
     A 4-digit BCD digital switch connected to S·+4 to S·+7 is read in turn by the strobe signal D1· to D1·+3, and stored in the binary format to D2·+1.

Related devices

→ **For the instruction execution complete flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8029	Instruction execution complete flag	OFF: Data is being output to D1· to D1·+3 or the instruction is not executed yet. ON: A cycle operation of outputting data to D1· to D1·+3 (scan of the 1st to 4th digits) is completed.

Cautions

1. When the command contact turns OFF

Though the contents of D2· do not change, all of D1· to D1·+3 turn OFF.

2. Number of occupied devices

1) When two sets of 4 digits (n = K2) are used, two devices are occupied starting from D2·.
2) When one set of 4 digits is used, four devices are occupied starting from S·. When two sets of 4 digits is used, eight devices are occupied starting from S·.

3. When connecting a digital switch of up to 3 digits

It is not necessary to wire the strobe signal (output for digit specification) D1· to unused digits. Because unused digits are occupied also by this instruction, however, they cannot be used for any other purpose. Make sure to leave unused outputs vacant.

4. Transistor output type is recommended

For continuously receiving digital switch values, make sure to use a transistor output type PLC.

→ **For a relay type PLC, refer to "How to use this instruction in a relay output type PLC" later.**

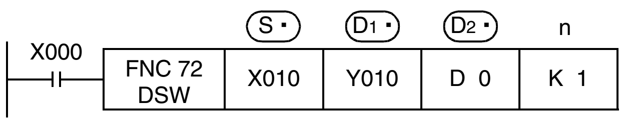
5. Digital switches

Use BCD output type digital switches.

Program example

In the program example shown below, digital switches are connected to inputs starting from X010 and outputs from Y010.

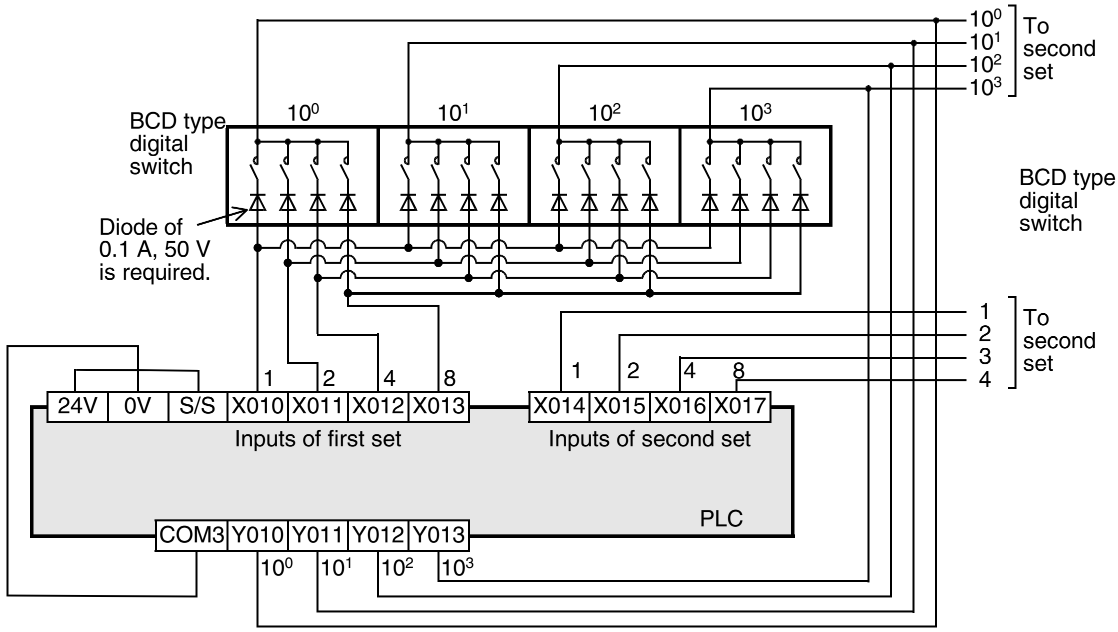
1. Program



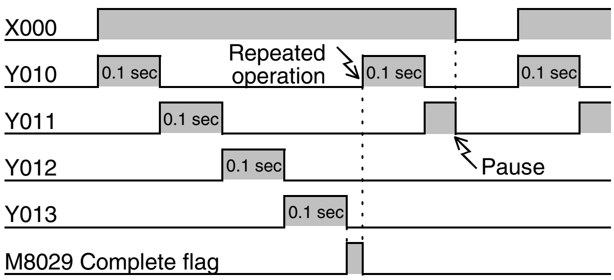
2. Connection diagram

The figure below shows an example of the FX3U series main unit (sink input/sink output).
For wiring details, refer to the following manual.

→ **FX3U Hardware Edition**



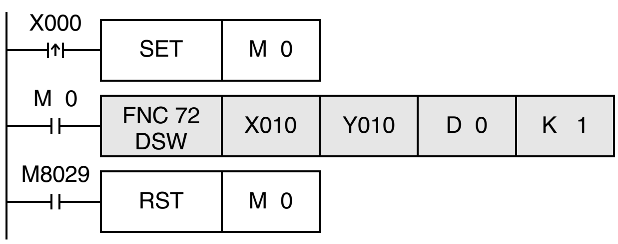
3. Timing chart



While X000 is ON, Y010 to Y013 turn ON in turn at every 100 ms. After one cycle is finished, the execution complete flag M8029 turns ON.

4. How to use this instruction in a relay output type PLC

By providing a "digital switch read input", this instruction can be used in a relay output type PLC.
When the push button switch (X000) is pressed, DSW (FNC 72) instruction executes a series of operations. Accordingly, with regard to this program, it is not necessary to consider the relay contact life even if Y010 to Y013 are relay outputs.



1) While M0 (digital switch read input) is ON, DSW (FNC 72) is driven.
2) DSW (FNC 72) completes one cycle of operation, and remains driven until the execution complete flag (M8029) turns ON.

15.4 FNC 73 – SEGD / Seven Segment Decoder

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction decodes data, and turns the seven-segment display unit (1 digit) ON.

1. Instruction format

FNC 73 SEGD P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	SEGD	Continuous Operation
	SEGDP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	

2. Set data

Operand Type	Description	Data Type
S·	Head word device to be decoded	16-bit binary
D·	Word device number storing the data to be displayed in the seven-segment display unit	16-bit binary

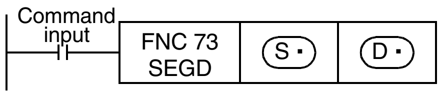
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D·									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

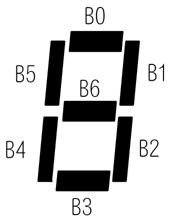
Explanation of function and operation

1. 16-bit operation (SEGD and SEGDP)

"0" to "F" (hexadecimal numbers) in low-order 4 bits (1 digit) of S· are decoded to data for the seven-segment display unit, and stored the low-order 8 bits of D·.



2. Seven-segment decoding table

S·					Seven-segment configuration	D·											Display data
Hexadecimal number	b3	b2	b1	b0		B15	...	B8	B7	B6	B5	B4	B3	B2	B1	B0	
0	0	0	0	0		–		–	0	0	1	1	1	1	1	1	0
1	0	0	0	1		–		–	0	0	0	0	0	1	1	0	1
2	0	0	1	0		–		–	0	1	0	1	1	0	1	1	2
3	0	0	1	1		–		–	0	1	0	0	1	1	1	1	3
4	0	1	0	0		–		–	0	1	1	0	0	1	1	0	4
5	0	1	0	1		–		–	0	1	1	0	1	1	0	1	5
6	0	1	1	0		–		–	0	1	1	1	1	1	0	1	6
7	0	1	1	1		–		–	0	0	1	0	0	1	1	1	7
8	1	0	0	0		–		–	0	1	1	1	1	1	1	1	8
9	1	0	0	1		–		–	0	1	1	0	1	1	1	1	9
A	1	0	1	0		–		–	0	1	1	1	0	1	1	1	A
B	1	0	1	1		–		–	0	1	1	1	1	1	0	0	b
C	1	1	0	0		–		–	0	0	1	1	1	0	0	1	C
D	1	1	0	1		–		–	0	1	0	1	1	1	1	0	d
E	1	1	1	0		–		–	0	1	1	1	1	0	0	1	E
F	1	1	1	1		–		–	0	1	1	1	0	0	0	1	F

↑
The head bit device or the least significant bit of a word device is handled as B0.

Caution

1. Number of occupied devices

Low-order 8 bits of D· are occupied, and high-order 8 bits do not change.

15.5 FNC 74 – SEGL / Seven Segment With Latch

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction controls one or two sets of 4-digit seven-segment display units having the latch function.

1. Instruction format

FNC 74
SEGL

16-bit Instruction	Mnemonic	Operation Condition
7 steps	SEGL	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	

2. Set data

Operand Type	Description	Data Type
S•	Head word device converted into the BCD format	16-bit binary
D•	Head Y number to be output	Bit
n	Parameter number [setting range: K0 (H0) to K7 (H7)]	16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D•		✓																	✓					
n																				✓	✓			

Explanation of function and operation

1. 16-bit operation (SEGL)

The 4-digit numeric value stored in S• is converted into BCD data, and each digit is output to the seven-segment display unit with the BCD decoder in the time division method.

Command input —| |— [FNC 74 SEGL | S• | D• | n]

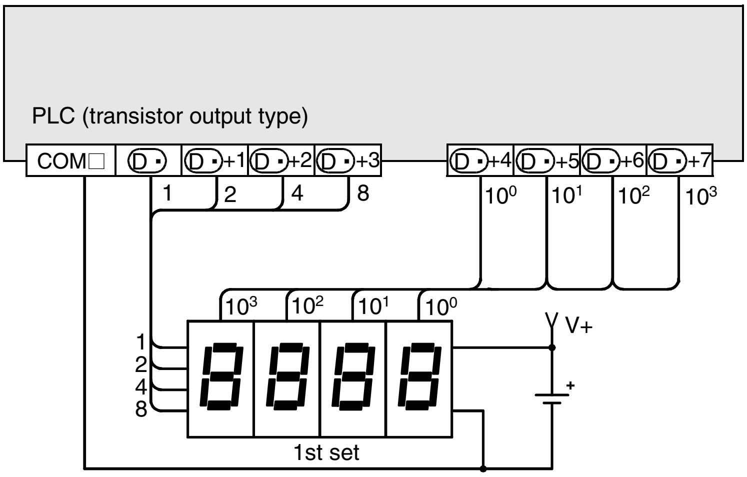
When using one set of 4 digits (n = K0 to K3)

→ **For selection of "n", refer to Subsection 15.5.2.**

1) Data and strobe signal
   A 4-digit numeric value stored in S• is converted from binary into BCD, and each digit is output in turn from D• to D• +3 in the time division method.
   The strobe signal is output in turn from D• +4 to D• +7 in the time division method also to latch one set of 4-digit seven-segment display unit.
2) For S•, binary data in the range from 0 to 9999 is valid.

3) Example of connecting one seven-segment display unit
The figure below shows an example of the FX3U series main unit (sink output).
For wiring details, refer to the following manual.

→ **FX3U Hardware Edition**

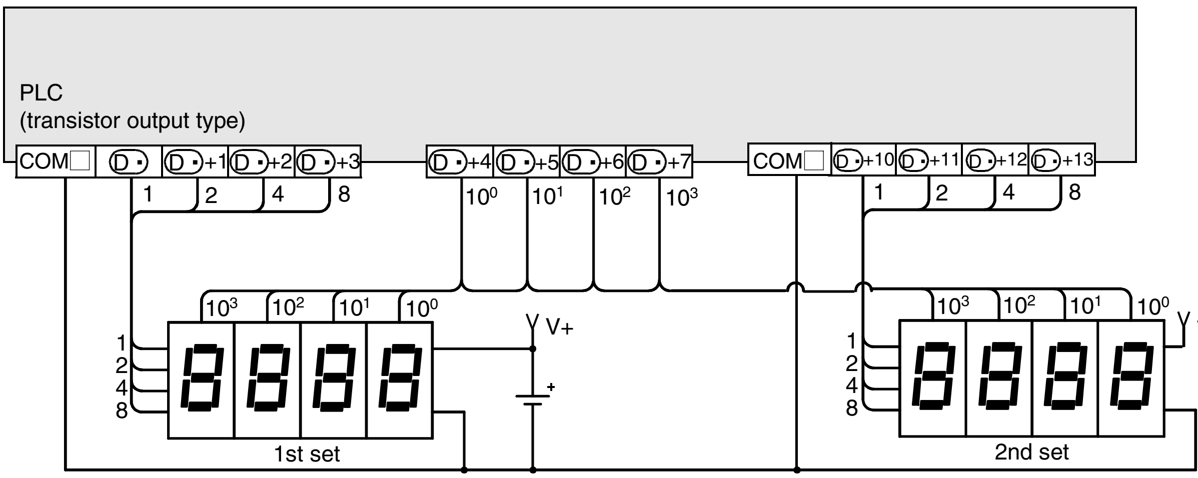


When using two sets of 4 digits (n = K4 to K7)

→ **For selection of "n", refer to Subsection 15.5.2.**

1) Data and strobe signal
   a) 1st set of 4 digits
      A 4-digit numeric value stored in S· is converted from binary into BCD, and its each digit is output in turn from D· to D·+3 in the time division method.
      The strobe signal is output in turn from D·+4 to D·+7 in the time division method also to latch the first set of 4-digit seven-segment display unit.
   b) 2nd set of 4 digits
      A 4-digit numeric value stored in S·+1 is converted from binary into BCD, and its each digit is output in turn from D·+10 to D·+13 in the time division method.
      The strobe signal is output in turn from D·+4 to D·+7 in the time division method also to latch the second set of 4-digit seven-segment display unit. (The strobe signal outputs D·+4 to D·+7 are shared by the 1st and 2nd sets.)
2) For S· and S·+1, binary data in the range from 0 to 9999 is valid.
3) Example of connecting two seven-segment display units
The figure below shows an example of the FX3U series main unit (sink output).
For wiring details, refer to the following manual.

→ **FX3U Hardware Edition**



11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FNC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

Related devices

→ **For the instruction execution complete flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8029	Instruction execution complete flag	Turns ON when output of 4 digits is finished.

Cautions

1. Time to update the 4-digit seven-segment display
The scan time (operation cycle) multiplied by 12 is required to update (one or two sets of) the 4-digit display.

2. Operation when the command input turns OFF
While the command input is ON, the operation is repeated.
When the command contact is set to OFF in the middle of an operation, the operation is paused. When the command contact is set to ON again, the operation is started from the beginning.

3. Number of occupied devices

When one set of 4 digits is used:	1 device is occupied from the head device specified in (S·).
	8 devices are occupied from the head device specified in (D·). Even if the number of digits is smaller than 4, unused devices cannot be used for any other purpose.
When two sets of 4 digits are used:	2 devices are occupied from the head device specified in (S·).
	Twelve devices are occupied from the head device specified in (D·). Even if the number of digits is smaller than 4, unused devices cannot be used for any other purpose.

4. Scan time (operation cycle) and the display timing
SEGL instruction is executed in synchronization with the scan time (operation cycle) of the PLC.
For achieving a series of display, the scan time of the PLC should be 10 ms or more.
If the scan time is less than 10 ms, use the constant scan mode so that the scan time exceeds 10 ms.

5. Output type of the PLC
Use a transistor output type PLC.

15.5.1 How to select a seven-segment display unit

When selecting a seven-segment display unit based on its electrical characteristics, refer to the manual below:

→ **For the wiring, refer to the Hardware Edition of the used PLC.**

1. Points to be checked for the seven-segment specifications

1) Whether the input voltage and current characteristics of the data input and strobe signal satisfy the output specifications of the PLC.
   - Whether the input signal voltage (Lo) is approximately 1.5 V or less
   - Whether the input voltage is from 5V DC to 30V DC
2) Whether the seven-segment display unit has the BCD decoding and latch functions

15.5.2 How to select parameter "n" based on seven-segment display specifications

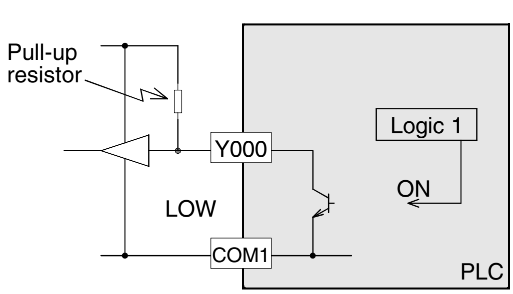
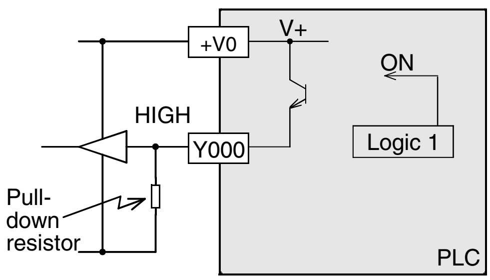
The value set to the parameter "n" varies depending on the signal logic of the seven-segment display. Select "n" as described below.
The check column is provided at the bottom of the table. Check a corresponding type of logic (positive or negative), and utilize it for parameter setting.

1. Role of the parameter "n"

The parameter "n" should be determined according to the data input logic (positive or negative) of the seven-segment display unit, the logic (positive or negative) of the strobe signal and the number of sets of 4 digits to be controlled (1 or 2).

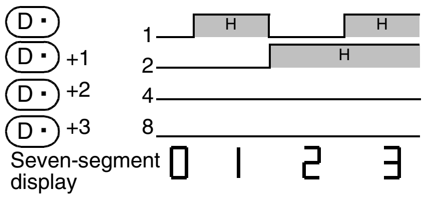
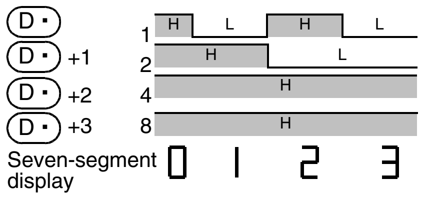
2. Checking the output logic of the PLC

Transistor outputs in PLCs are classified into the sink output type and source output type. The table below shows the specifications for each type.

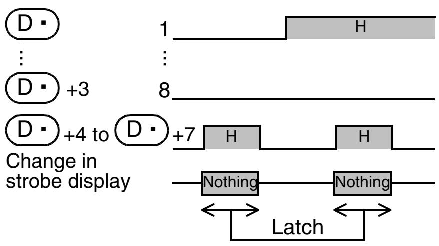
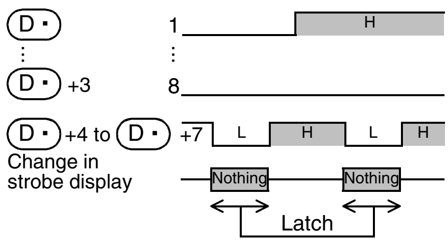
Logic	Negative logic	Positive logic
Output type	Sink output (– common)	Source output (+ common)
Output circuit		
Description	Because transistor output (sink) is provided, the output becomes low level (0 V) when the internal logic is "1 (ON output)". This is called "negative logic."	Because transistor output (source) is provided, the output becomes high level (V+) when the internal logic is "1 (ON output)". This is called "positive logic."
Check		

3. Confirming the logic of the seven-segment display unit

1) Data input

Logic	Negative logic	Positive logic
Timing chart		
Description	BCD data at low level	BCD data at high level
Check		

2) Strobe signal

Logic	Negative logic	Positive logic
Timing chart		
Description	Data latched at low level is held.	Data latched at high level is held.
Check		

4. Setting the parameter "n"

Set a proper value according to the logic (positive or negative) of the PLC and the logic (positive or negative) of the seven-segment display unit as shown in the table below:

PLC output logic	Data input	Strobe signal	Parameter "n"	
			4 digits × 1 set	**4 digits × 2 sets**
Negative logic	Negative logic (match)	Negative logic (match)	0	4
		Positive logic (mismatch)	1	5
	Positive logic (mismatch)	Positive logic (match)	2	6
		Negative logic (mismatch)	3	7
Positive logic	Positive logic (match)	Positive logic (match)	0	4
		Negative logic (mismatch)	1	5
	Negative logic (mismatch)	Positive logic (mismatch)	2	6
		Negative logic (match)	3	7

5. Explanation of the parameter "n" setting method according to an actual example

When the following seven-segment display unit is selected, "n" should be "1" when one display unit is connected (4 digits × 1 set) or "5" when two display units are connected (4 digits × 2 sets).

1) Transistor output of PLC
- Sink output = Negative logic
- Source output = Positive logic

2) Seven-segment display unit
- Data input = Negative logic
- Strobe signal = Positive logic

PLC output logic	Data input	Strobe signal	Parameter "n"	
			4 digits × 1 set	**4 digits × 2 sets**
Negative logic	Negative logic (match)	Negative logic (match)	0	4
		Positive logic (mismatch)	1	5
	Positive logic (mismatch)	Positive logic (match)	2	6
		Negative logic (mismatch)	3	7

15.6 FNC 75 – ARWS / Arrow Switch

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction inputs data through arrow switches used for shifting the digit and incrementing/decrementing the numeric value in each digit.

1. Instruction format

FNC 75
ARWS

16-bit Instruction	Mnemonic	Operation Condition
9 steps	ARWS	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand Type	Description	Data Type
S•	Head bit device number to be input	16- or 32-bit binary
D1•	Word device number storing data converted into BCD	16- or 32-bit binary
D2•	Head bit device (Y) number connected to seven-segment display unit	16- or 32-bit binary
n	Number of digits of seven-segment display unit [setting range: K0 to K3]	16- or 32-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•	✓	✓	✓			✓	▲												✓					
D1•												✓	✓	✓	✓	✓	✓	✓	✓					
D2•		✓																	✓					
n																				✓	✓			

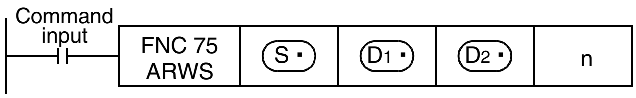
▲: "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

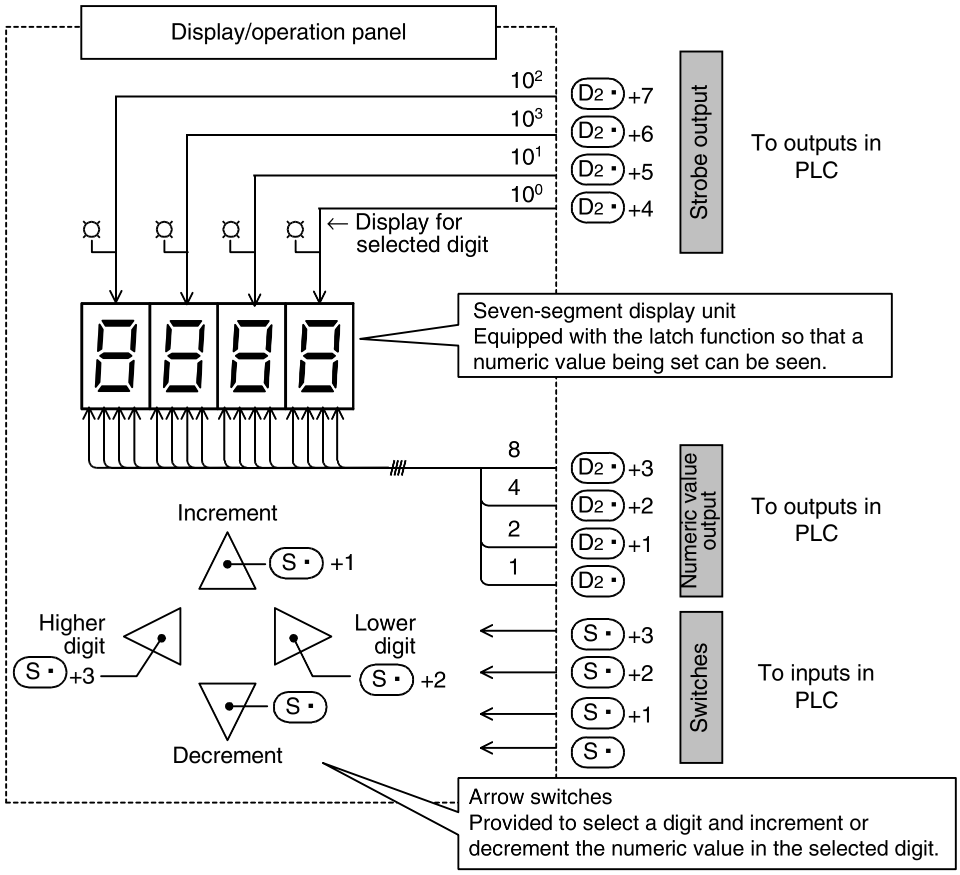
Four arrow switches are connected to the inputs S• to S•+3, a seven-segment display unit having the BCD decoder is connected to the outputs D2• to D2•+7, and a numeric value is input to D1•.

1. 16-bit operation (ARWS)

D1• actually stores a 16-bit binary value in the range from 0 to 9999, but the value is expressed in the BCD format in the explanation below for convenience.
When the command input is set to ON, ARWS instruction executes the following operation.



Contents of the display and operation part



1) Specifying the number of digits of the seven-segment display unit having the BCD decoder n
In the explanation below, "n" is set to "4" (up to the 10^3 digit).

2) Operation of the digit selection switches (S•+2 and S•+3)
- Operation when the lower digit input S•+2 turns ON
Every time the lower digit switch is pressed, the digit specification changes in the way "$10^3 \rightarrow 10^2 \rightarrow 10^1 \rightarrow 10^0 \rightarrow 10^3$".
- Operation when the higher digit input S•+3 turns ON
Every time the higher digit switch is pressed, the digit specification changes in the way "$10^3 \rightarrow 10^0 \rightarrow 10^1 \rightarrow 10^2 \rightarrow 10^3$".

3) Operation of the LED for displaying a selected digit (D2•+4 to D2•+7)
A specified digit can be displayed by the LED offered by the strobe signals D2•+4 to D2•+7.

4) Operation of the switches for changing data in each digit (S· and S·+1)
   In a digit specified by a digit selection switch described above, data is changed as follows:
   - When the increment input turns ON
     Every time the increment switch is pressed, the contents of D1· change in the way "0 → 1 → 2 → ... → 8 → 9 → 0 → 1".
   - When the decrement input turns ON
     Every time the decrement switch is pressed, the contents of D1· change in the way "0 → 9 → 8 → 7 ... 1 → 0 → 9".

   The contents can be displayed in the seven-segment display unit.
   As described above, a target numeric value can be written to D1· using a series of operation while looking at the seven-segment display unit.

Cautions

1. Setting of the parameter "n"

Refer to the explanation of parameter setting in SEGL (FNC 74) instruction. The setting range is from 0 to 3 for ARWS instruction.

→ For the parameter setting, refer to Subsection 15.5.2.

2. Output type of the PLC

Use a transistor output type PLC.

3. Scan time (operation cycle) and the display timing

ARWS instruction is executed in synchronization with the scan time (operation cycle) of the PLC.
For achieving a series of display, the scan time of the PLC should be 10 ms or more.
If the scan time is less than 10 ms, use the constant scan mode so that the scan time exceeds 10 ms.

4. Number of occupied devices

1) Four input devices are occupied starting from S·.
2) Eight output devices are occupied starting from D2·.

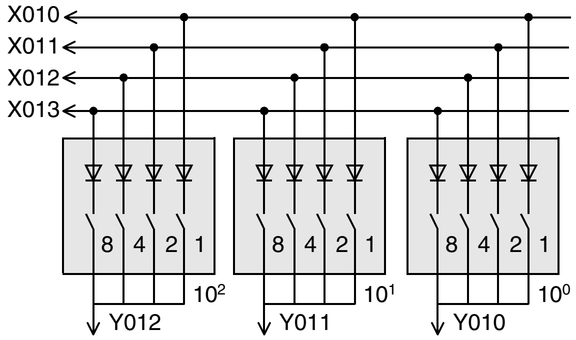
5. Limitation in the number of the instruction

ARWS instruction can be used only once in a program.
When ARWS instruction should be used two or more times, use the indexing (V, Z) function.

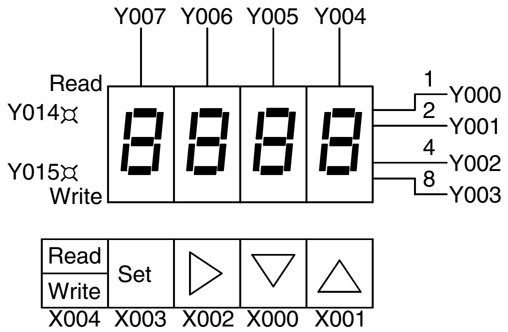
Program example

1. When changing the timer number and displaying the current value

1) Specifying the timer number using a 3-digit digital switch



2) Setting the constant of the timer using the arrow switches

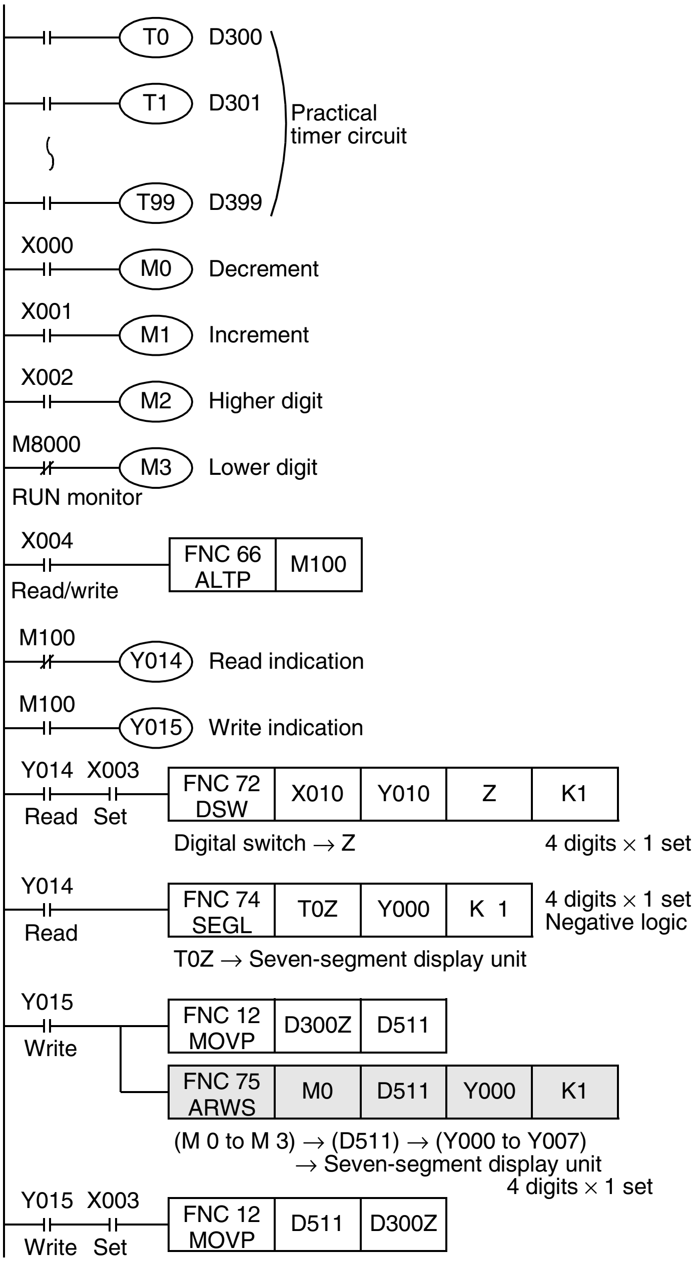


Explanation of operation

Every time the read/write key is pressed, the read/write LED lights alternately.

- In reading
  Set the timer number using the digital switch, and then press the set switch (X003).
- In writing
  Set a numeric value using the arrow switches while looking at the seven-segment display unit, and then press the switch X003.

Program


T0
D300
T1
D301
Practical timer circuit
T99
D399
X000
M0
Decrement
X001
M1
Increment
X002
M2
Higher digit
M8000
M3
Lower digit
RUN monitor
X004
FNC 66 ALTP
M100
Read/write
M100
Y014
Read indication
M100
Y015
Write indication
Y014 X003
Read Set
FNC 72 DSW
X010
Y010
Z
K1
Digital switch → Z
4 digits × 1 set
Y014
Read
FNC 74 SEGL
T0Z
Y000
K 1
4 digits × 1 set
Negative logic
T0Z → Seven-segment display unit
Y015
Write
FNC 12 MOVP
D300Z
D511
FNC 75 ARWS
M0
D511
Y000
K1
(M 0 to M 3) → (D511) → (Y000 to Y007)
→ Seven-segment display unit
4 digits × 1 set
Y015 X003
Write Set
FNC 12 MOVP
D511
D300Z

15.7 FNC 76 – ASC / ASCII Code Data Input

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction converts a half-width alphanumeric character string into ASCII codes.
Use this instruction for selecting one among two or more messages and displaying it on an external display unit.

1. Instruction format

FNC 76 ASC

16-bit Instruction	Mnemonic	Operation Condition
11 steps	ASC	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	

2. Set data

Operand Type	Description	Data Type
(S)	Eight half-width alphanumeric characters input from a personal computer	Character string (only ASCII codes)
(D·)	Head word device number storing ASCII data	16- or 32-bit binary

3. Applicable devices

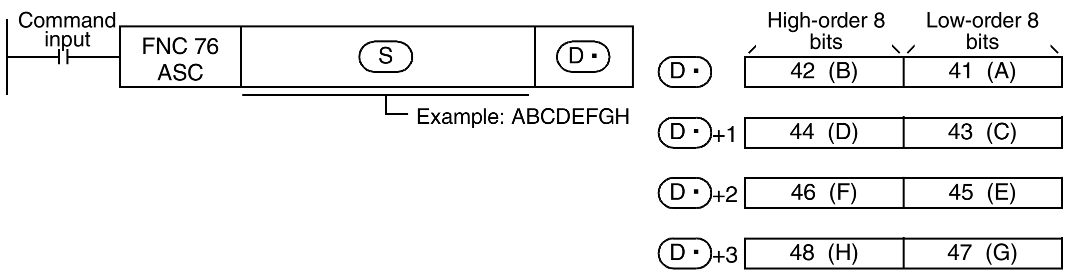
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S)																							✓	
(D·)												✓	✓	✓	✓	✓			✓					

Explanation of function and operation

1. 16-bit operation (ASC)

The half-width alphanumeric characters specified in (S) are converted into ASCII codes, and each ASCII code is transferred in turn to (D·).

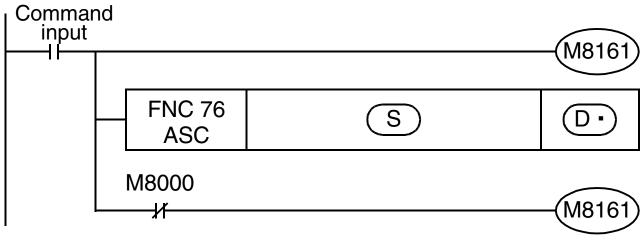
- (S) can handle half-width characters A to Z, 0 to 9 and symbols (, but cannot handle regular-width characters).
  A character string is entered when a program is created with a programming tool.
- (D·) stores converted ASCII codes in the order of low-order 8 bits and high-order 8 bits by 2 characters/byte at one time.



11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FNC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

Extension function

When M8161 is set to ON for making the extension function valid, a half-width alphanumeric character string specified in S is converted into ASCII codes, and transferred in turn only to low-order 8 bits (1 byte) of D•.



"H00" is stored in high-order 8 bits.

	D•		S
	High-order 8 bits	Low-order 8 bits	Character string
D•	00	41	A
D• +1	00	42	B
D• +2	00	43	C
D• +3	00	44	D
D• +4	00	45	E
D• +5	00	46	F
D• +6	00	47	G
D• +7	00	48	H

Related devices

Device	Name	Description
M8161	Extension function flag	8-bit processing mode for ASC (FNC 76), RS (FNC 80), ASCI (FNC 82), HEX (FNC 83) and CCD (FNC 84) instructions OFF: Two characters are stored to low-order 8 bits and high-order 8 bits in this order at one time (2 characters/word). ON: One character is stored to low-order 8 bits at one time (1 character/word).

Caution

1. Number of occupied devices

1) While the extension function is OFF
   - D• occupies as many devices as the number of characters divided by "2". (The decimal point is rounded up.)
2) While the extension function is ON
   - D• occupies as many devices as the number of characters in the character string.

2. When using RS (FNC 80), ASCI (FNC 82), HEX (FNC 83) and/or CCD (FNC 84) instructions

The extension function flag M8161 is also used for other instructions.
When using an instruction described above and the ASC instruction in the same program, make sure to set M8161 to ON or OFF just before the ASC instruction so that M8161 does not apply to another instruction.

15.8 FNC 77 – PR / Print (ASCII Code)

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction outputs ASCII code data to outputs (Y) in parallel.

1. Instruction format

FNC 77
PR

16-bit Instruction	Mnemonic	Operation Condition
5 steps	PR	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	

2. Set data

Operand Type	Description	Data Type
S·	Head device number storing ASCII code data	Character string (only ASCII codes)
D·	Head output (Y) number to which ASCII code data is output	Bit

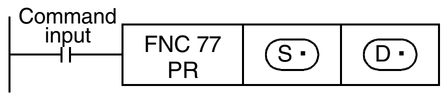
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·												✓	✓	✓	✓				✓					
D·		✓																	✓					

Explanation of function and operation

1. 16-bit operation (PR)

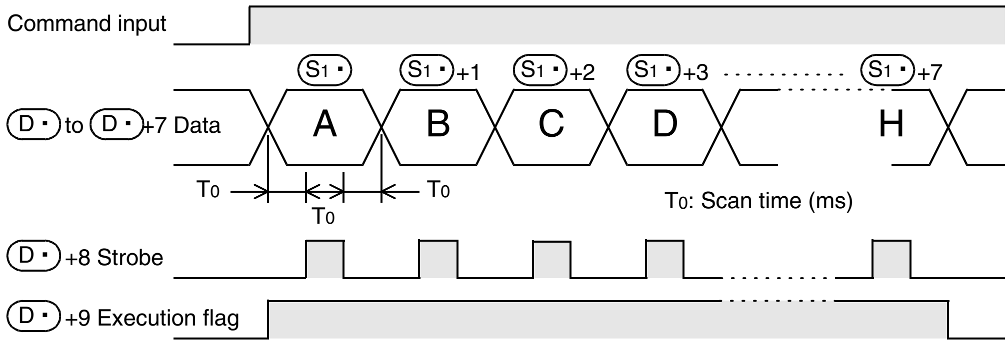
ASCII codes stored in low-order 8 bits (1 byte) of S· to S·+7 are output to D· to D·+7 in turn by one character at a time in the time division method.



The timing chart below shows a case in which the following ASCII codes are stored in S· to S·+7.
Eight bytes are sent from S· = "A" at first to "S·+7 = "H" at the end.

S·	S·+1	S·+2	S·+3	S·+4	S·+5	S·+6	S·+7
A (H41)	B (H42)	C (H43)	D (H44)	E (H45)	F (H46)	G (H47)	H (H48)

2. Timing chart



Types of output signals

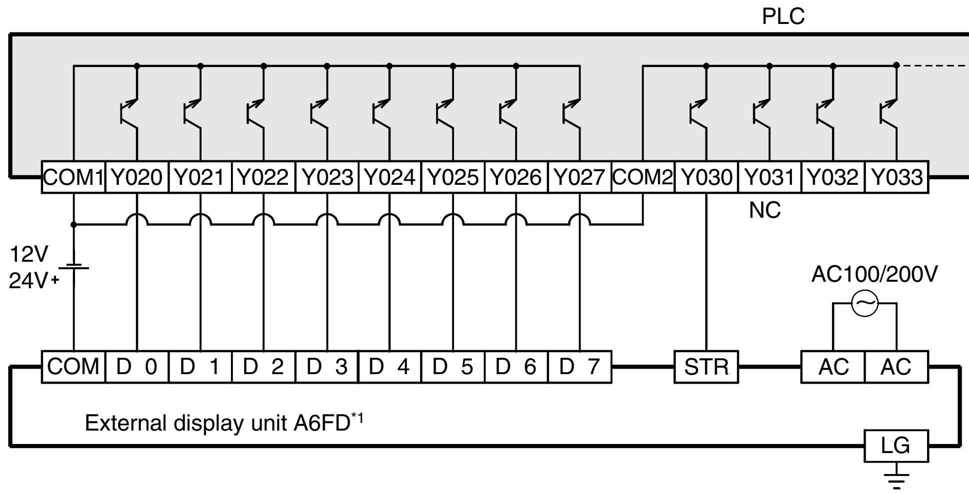
- D• to D•+7: Sending output (D• handles low-order bits, and D•+7 handles high-order bits.)
- D•+8: Strobe signal
- D•+9: Execution flag which operates as shown in the above timing chart

Extension function

1. 16-byte serial output

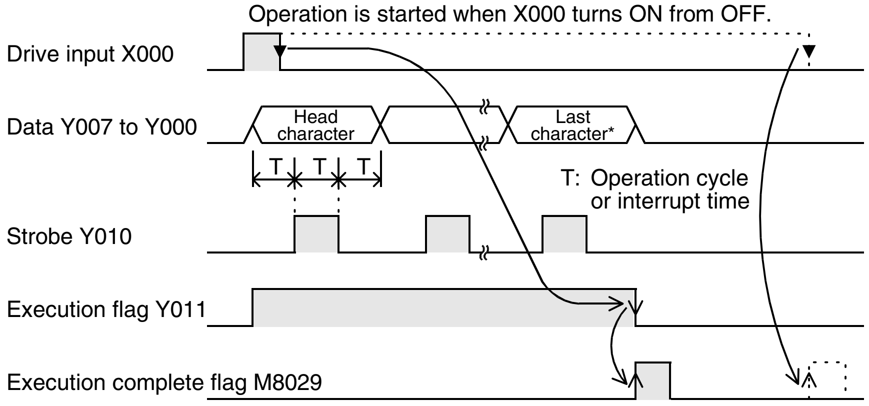
Depending on the ON/OFF status of the special auxiliary relay M8027, the number of characters output by one-time execution of the instruction varies.
While M8027 is OFF, 8-byte serial output (fixed to 8 characters) is executed. While M8027 is ON, 16-byte serial output (1 to 16 characters) is executed.
In the example shown below, up to 16 characters (1 character/byte) are output to the display unit (external display unit A6FD, for example).
It is supposed that data to be displayed is stored in hexadecimal codes in D300 to D307.

1) Connection example of the external display unit A6FD*1
The PLC shown in the example below is the FX2N-16EYT (sink input/sink output) connected to the FX3U-32M□.



*1. A6FD was distributed only inside Japan, however, production of the external display unit A6FD was terminated in November 2002.

2) Timing chart (while M8027 is ON)



* If "H00 (NUL code)" is contained in the data (16 characters), the character just before "H00 (NUL code)" is handled as the last character.

Related devices

Device	Name	Description
M8027[*1]	PR mode	OFF: 8-byte serial output (fixed to 8 characters) ON: 16-byte serial output (1 to 16 characters)

*1. Cleared when the PR mode is changed from RUN to STOP.

Cautions

1. Command input and instruction operation

While the command input is ON: Even if the command input is continuously ON or if the pulse operation type instruction is used, execution is completed after a series of outputs. M8029 turns ON only while M8027 is ON.

While the command input is OFF: All outputs are OFF.

2. Relationship with the scan time (operation cycle)

This instruction is executed in synchronization with the scan time.
If the scan time is short, the constant scan mode can be used. If the scan mode is too long, the timer interrupt function can be used.

3. Output type of the PLC

Use a transistor output type PLC.

4. When "00H (NUL code)" is contained in the data (while M8027 is ON)

The instruction is executed completely, and the data after "00H" is not output.
M8029 remains ON during one operation cycle.

15.9 FNC 78 – FROM / Read From A Special Function Block

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction reads the contents of buffer memories (BMF) in a special extension unit/block attached to a PLC.
When a large capacity of buffer memory (BFM) data is read by this instruction, a watchdog timer error may occur. When bad effect is not given to the control even if data to be read is divided, use RBFM (FNC278) instruction.

→ **For RBFM (FNC278) instruction, refer to Section 31.1.**

1. Instruction format

D	FNC 78 FROM	P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	FROM	Continuous Operation
	FROMP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DFROM	Continuous Operation
	DFROMP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
m1	Unit number of a special extension unit/block (K0 to K7 from the right side of the main unit)	16- or 32-bit binary
m2	Transfer source buffer memory (BFM) number	16- or 32-bit binary
(D·)	Transfer destination device number	16- or 32-bit binary
n	Number of transfer points	16- or 32-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
m1														✓	✓					✓	✓			
m2														✓	✓					✓	✓			
(D·)									✓	✓	✓	✓	✓	✓	✓		✓	✓	✓					
n														✓	✓					✓	✓			

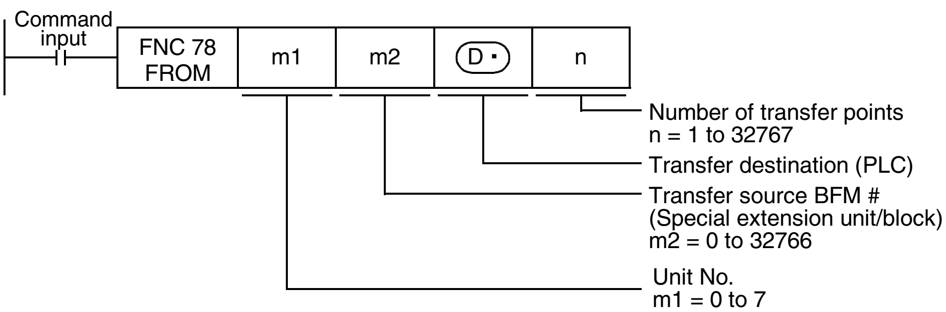
Explanation of function and operation

1. 16-bit operation (FROM and FROMP)

→ **For the common items between FROM instruction and TO instruction, refer to Subsection 15.9.1.**

Special extension unit/block (BFM) → PLC (word device)

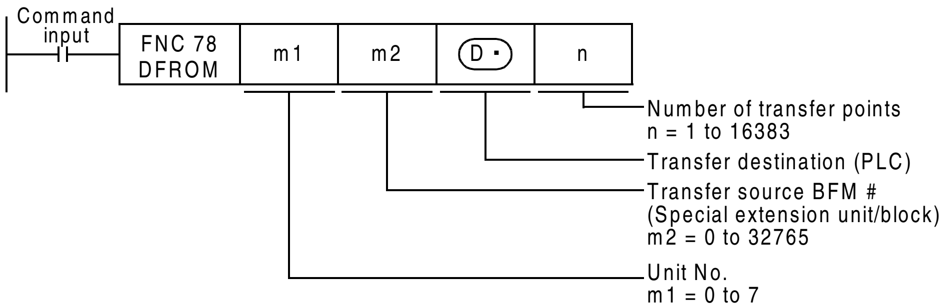
"n"-point 16-bit data starting from the buffer memory (BFM) # m2 inside a special extension unit/block No. m1 are transferred (read) to "n"-point 16-bit data starting from (D·) inside a PLC.



2. 32-bit operation (DFROM and DFROMP)

Special extension unit/block (BFM) → PLC (word device)

"n" 32-bit data starting from the buffer memory (BFM) # [m2+1, m2] inside a special extension unit/block No. m1 are transferred (read) to "n" devices starting from [D· +1, D·] inside a PLC.



Related devices

Device	Name	Description
M8028	Enable interrupt flag	Disables or enables interrupts while FROM/TO instruction is executed. → **For details, refer to "Acceptance of interrupts while FROM/TO instruction is executed (M8028)" on the next page.** OFF: Disables interrupts. (Interrupts are executed after FROM/TO instruction is executed.) ON: Enables interrupts.

Cautions

1. Digit specification in bit device D·

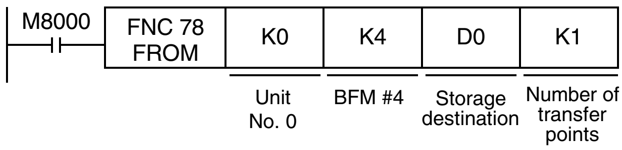
For the 16-bit operation instruction, specify K1 to K4. For the 32-bit operation instruction, specify K1 to K8.

Program examples

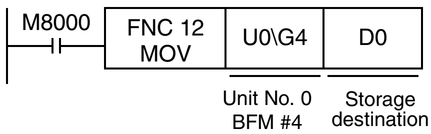
In programs, the contents of buffer memories (BFMs) in special extension unit/blocks are read (transferred) to data registers (D), extension registers (R) or auxiliary relays (M) with digit specification using an applied instruction such as FROM, MOV and BMOV.

Example: When the BFM #4 (abnormal station information) in the CC-Link/LT master unit (whose unit number is fixed to "0") built in the FX3UC-32MT-LT is read to D0

- In case of FROM instruction

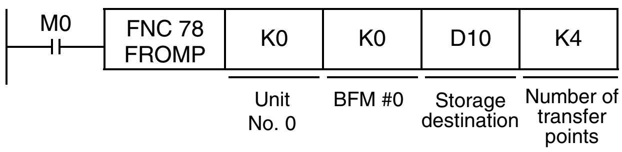


- In case of MOVE instruction

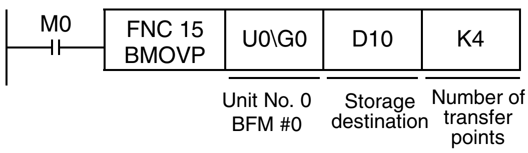


Example: When the BFMs #0 to 3 (remote station connection information) in the CC-Link/LT master unit (whose unit number is fixed to "0") built in the FX3UC-32MT-LT are read to D10 to D13

- In case of FROM instruction



- In case of BMOV instruction



15.9.1 Common items between FROM instruction and TO instruction (details)

Contents specified by operands

1. Unit number "m1" of a special extension unit/block

Use the unit number to specify which equipment FROM/TO instruction works for.
Setting range: K0 to K7

Unit No. 0 Built-in CC-Link/LT		Unit No.1	Unit No.2		Unit No.3
FX3UC-32MT-LT main unit	I/O extension block	Special extension block	Special extension block	I/O extension block	Special extension block

A unit number is automatically assigned to each special extension unit/block connected to a PLC.
The unit number is assigned in the way "No. 0 → No. 1 → No. 2 ..." starting from the equipment nearest to the main unit.

2. Buffer memory (BFM) number "m2"

Up to 32767 16-bit RAM memories are built into a special extension unit/block, and they are called buffer memories.
Buffer memory numbers range from "0" to "32766" and their contents vary depending on the function of the extension equipment.
Setting range: K0 to K32766

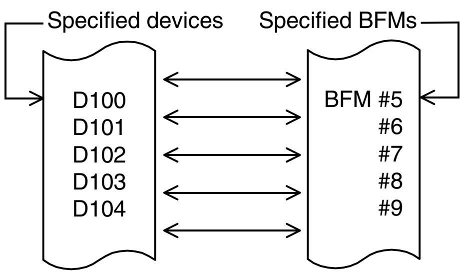
- When BFMs are handled in a 32-bit instruction, a specified BFM stores low-order 16 bits, and a consecutive BFM stores high-order 16-bits.

High-order 16 bits	Low-order 16 bits	
BFM #10	BFM #9	← Specified BFM number

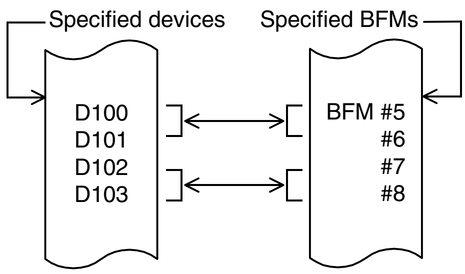
3. Number of transfer points "n"

Setting range: K1 to K32767

Specify the number of transferred word devices in "n".
"n =2" in a 16-bit instruction indicates the same meaning with "n = 1" in a 32-bit instruction.



In the case of "n = 5" in a 16-bit instruction



In the case of "n = 2" in a 32-bit instruction

Acceptance of interrupts while FROM/TO instruction is executed (M8028)

1. While M8028 is OFF

While a FROM/TO instruction is being executed, interrupts are automatically disabled. Input interrupts and timer interrupts are not executed.
Interrupts generated during the execution of FROM/TO instructions are immediately executed after the FROM/TO instruction completes.
FROM/TO instructions can be used in interrupt programs.

2. While M8028 is ON

When an interrupt is generated during the execution of a FROM/TO instruction, the FROM/TO operation is momentarily paused while the interrupt program executes. FROM/TO instructions cannot be used in interrupt programs.

Action against watchdog timer error

1. Cause of watchdog timer error

A watchdog timer error may occur in the following cases:

1) When many special extension equipment is connected
When many special extension equipment (such as positioning units, cam switches, link units and analog units) are connected, considerable time may be required to initialize buffer memories when the PLC mode is set to RUN, the operation time may be long, and a watchdog timer error may occur.

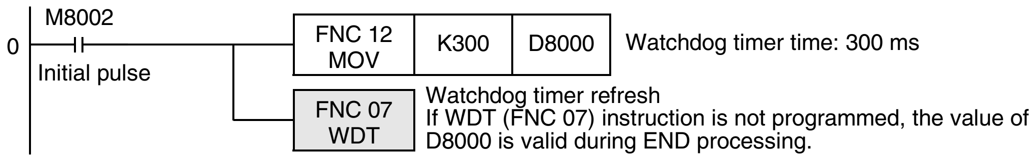
2) When many FROM/TO instructions are driven at the same time
When many FROM/TO instructions are driven at the same time or when many buffer memories are transferred, the operation time may be long, and a watchdog timer error may occur.

2. Countermeasures

1) Using RBFM (FNC278) or WBFM (FNC279) instruction [Ver.2.20 or later]

→ For divided BFM read [RBFM (FNC278) instruction], refer to Section 31.1.
→ For divided BFM write [WBFM (FNC279) instruction], refer to Section 31.2.

2) Changing the watchdog timer time
By overwriting the contents of D8000 (watchdog timer time), the watchdog timer detection time can be changed.
When the program shown below is input, the sequence program after the input will be monitored with the new watchdog timer time.



3) Changing FROM/TO instruction execution timing
Shift FROM/TO instruction execution timing to make the operation time shorter.

Handling of special extension units/blocks

For the special extension unit/block connection method, number of connectable special extension units/blocks and handling of I/O numbers, refer to the manuals of the PLC and each special extension unit/block.

15.10 FNC 79 – TO / Write To A Special Function Block

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction writes data from a PLC to buffer memories (BFM) in a special extension unit/block.
When a large capacity of data is written to buffer memories (BFM) by this instruction, a watchdog timer error may occur. When splitting the data to be written does not affect the control, use WBFM (FNC279) instruction.

→ For WBFM (FNC279) instruction, refer to Section 31.2.

1. Instruction format

D	FNC 79 TO	P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	TO	Continuous Operation
	TOP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DTO	Continuous Operation
	DTOP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
m1	Unit number of a special extension unit/block (K0 to K7 from the right side of the main unit)	16- or 32-bit binary
m2	Transfer destination buffer memory (BFM) number	16- or 32-bit binary
S•	Device number storing the transfer source data	16- or 32-bit binary
n	Number of transfer points	16- or 32-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
m1														✓	✓					✓	✓			
m2														✓	✓					✓	✓			
S•									✓	✓	✓	✓	✓	✓	✓		✓	✓	✓					
n														✓	✓					✓	✓			

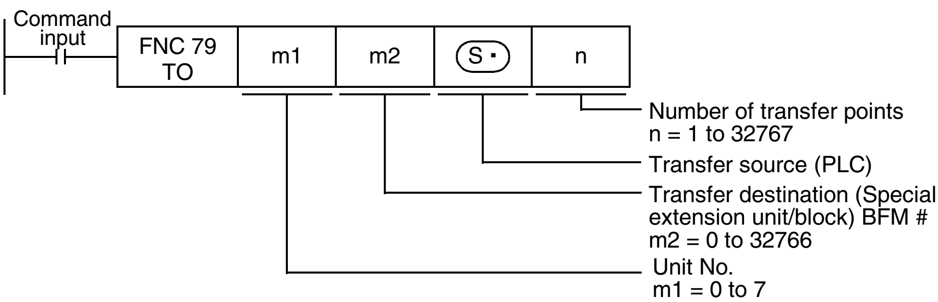
Explanation of function and operation

1. 16-bit operation (TO and TOP)

→ For the common items between FROM instruction and TO instruction, refer to Subsection 15.9.1.

PLC (word device) → Special extension unit/block (BFM)

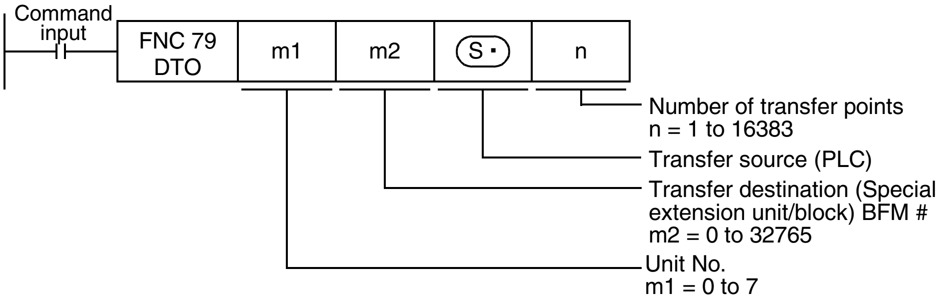
"n"-point 16-bit data starting from S• inside a PLC are transferred (written) to "n"-point buffer memories starting from the buffer memory (BFM) # m2 inside a special extension unit/block No. m1.



2. 32-bit operation (DTO and DTOP)

PLC (word device) → Special extension unit/block (BFM)

"n"-point 32-bit data starting from [S·, S·+1] inside a PLC are transferred (written) to "n"-point buffer memories starting from the buffer memory (BFM) # [m2+1, m2] inside a special extension unit/block No. m1.



Related devices

Device	Name	Description
M8028	Enable interrupt flag	Disables or enables interrupts while FROM/TO instruction is executed. → **For details, refer to "Acceptance of interrupt while FROM/TO instruction is executed (M8028)" in Subsection 15.9.1.** OFF: Disables interrupts. (Interrupts are executed after FROM/TO instruction is executed.) ON: Enables interrupts.

Cautions

1. Digit specification in bit device S·

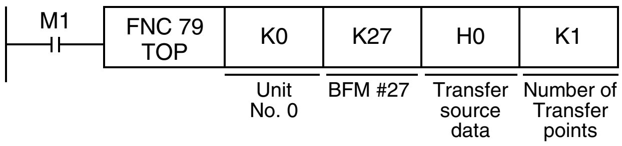
For the 16-bit operation instruction, specify K1 to K4. For the 32-bit operation instruction, specify K1 to K8.

Program examples

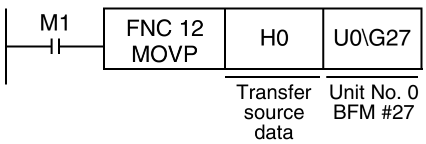
In programs, the contents of data registers (D), extension registers (R), auxiliary relays (M) with digit specification, or constants (K, H) are written (transferred) to buffer memories in a special extension unit/block using an applied instruction such as TO, MOV and BMOV.

Example: When writing "H0" to the BFM #27 (command) in the CC-Link/LT master unit (whose unit number is fixed to "0") built in the FX3UC-32MT-LT

- In case of TO instruction



- In case of MOV instruction



16. External FX Device – FNC 80 to FNC 89

FNC 80 to FNC 89 provide control instructions for special adapters mainly connected to serial ports. PID control loop instruction is included in this group.

FNC No.	Mnemonic	Format	Function	Reference
80	RS	RS S m D n	Serial Communication	Section 16.1
81	PRUN	PRUN S D	Parallel Run (Octal Mode)	Section 16.2
82	ASCI	ASCI S D n	Hexadecimal to ASCII Conversion	Section 16.3
83	HEX	HEX S D n	ASCII to Hexadecimal Conversion	Section 16.4
84	CCD	CCD S D n	Check Code	Section 16.5
85	–			–
86	–			–
87	RS2	RS2 S m D n n1	Serial Communication 2	Section 16.6
88	PID	PID S1 S2 S3 D	PID Control Loop	Section 16.7
89	–			–

16.1 FNC 80 – RS / Serial Communication

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction sends and receives data in no-protocol communication by way of a serial port (only the ch1) in accordance with RS-232C or RS-485 provided in the main unit.

1. Instruction format

FNC 80 RS	16-bit Instruction	Mnemonic	Operation Condition	32-bit Instruction	Mnemonic	Operation Condition
	9 steps	RS	Continuous Operation	–		

2. Set data

Operand type	Description	Data type
(S•)	Head device of data registers storing data to be sent	16-bit binary or character string
m	Number of bytes of data to be sent [setting range: 0 to 4096]*1	16-bit binary
(D•)	Head device of data registers storing received data when receiving is completed	16-bit binary or character string
n	Number of bytes to be received [setting range: 0 to 4096]*1	16-bit binary

*1. Make sure to observe "m + n ≤ 8000."

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S•)														✓	✓				✓					
m														✓	✓					✓	✓			
(D•)														✓	✓				✓					
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (RS)

This instruction sends and receives data in no-protocol communication by way of serial ports in accordance with RS-232C or RS-485 provided in the main unit.

→ **For detailed explanation, refer to the Communication Control Manual.**

Command input	FNC 80 RS	(S•)	m	(D•)	n

Related devices

→ For detailed explanation, refer to the Communication Control Manual.

Device	Name
M8063	Serial communication error 1
M8121*1	Sending wait flag
M8122*1	Sending request
M8123*1	Receiving complete flag
M8124	Carrier detection flag
M8129	Timeout evaluation flag
M8161*3	8-bit processing mode

Device	Name
D8120*2	Communication format setting
D8122*3	Remaining number of data to be sent
D8123*3	Monitor for number of received data
D8124	Header
D8125	Terminator
D8129*2	Timeout time setting
D8063	Error code number of serial communication error 1
D8405	Communication parameter display
D8419	Operation mode display

*1. Cleared in the following cases:
- When the PLC mode is changed from RUN to STOP
- When RS instruction is not driven

*2. Latched (battery backed).

*3. Cleared when the PLC mode is changed from RUN to STOP.

System configuration

For using this instruction, it is necessary to attach either product shown in the table below to the main unit.

→ For the system configuration, refer to the Hardware Edition of the used PLC.
→ For detailed explanation, refer to the Communication Control Manual.

Communication type	Option
RS-232C communication	FX3U-232-BD or FX3U-232ADP (with FX3U-CNV-BD)
RS-485 communication	FX3U-485-BD or FX3U-485ADP (with FX3U-CNV-BD)

Differences between RS (FNC 80) instruction and RS2 (FNC 87) instruction

Item	RS2 instruction	RS instruction	Remarks
Header size	1 to 4 characters (bytes)	Up to 1 character (byte)	In RS2 instruction, up to 4 characters (bytes) can be specified as a header or terminator.
Terminator size	1 to 4 characters (bytes)	Up to 1 character (byte)	
Attachment of check sum	The check sum can be automatically attached.	The check sum should be attached by a user program.	In RS2 instruction, the check sum can be automatically attached to the sent and received data. In this case, however, make sure to use a terminator in the communication frame to be sent and received.

Cautions

→ For other cautions, refer to the Communication Control Manual.

- RS (FNC 80) instruction can be used for ch1 only (cannot be used for ch2).
- Do not drive two or more RS (FNC 80) and/or RS2 (FNC 87) instructions for the same port at the same time.
- It is not permitted to use RS (FNC 80)/RS2 (FNC 87) instruction and IVCK (FNC270)/IVDR (FNC271)/ IVRD (FNC272)/IVWR (FNC273)/IVBWR (FNC274) instruction for the same port.

16.2 FNC 81 – PRUN / Parallel Run (Octal Mode)

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction handles the device number of (S•) with digit specification and the device number of (D•) as octal numbers, and transfers data.

1. Instruction format

D	FNC 81 PRUN	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	PRUN	Continuous Operation
	PRUNP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DPRUN	Continuous Operation
	DPRUNP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
(S•)	Digit specification*1	16- or 32-bit binary
(D•)	Device number of transfer destination*1	16- or 32-bit binary

*1. Make sure that the least significant digit of a specified device number is "0".

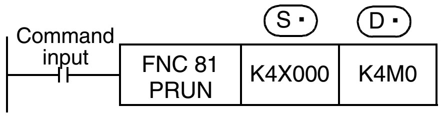
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices														Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S•)								✓		✓									✓					
(D•)									✓	✓									✓					

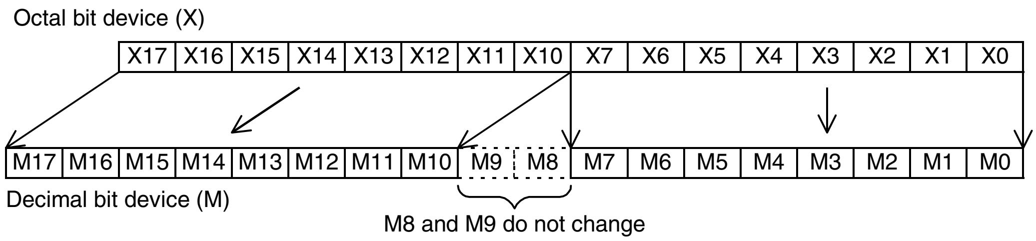
Explanation of function and operation

1. 16-bit operation (PRUN and PRUNP)

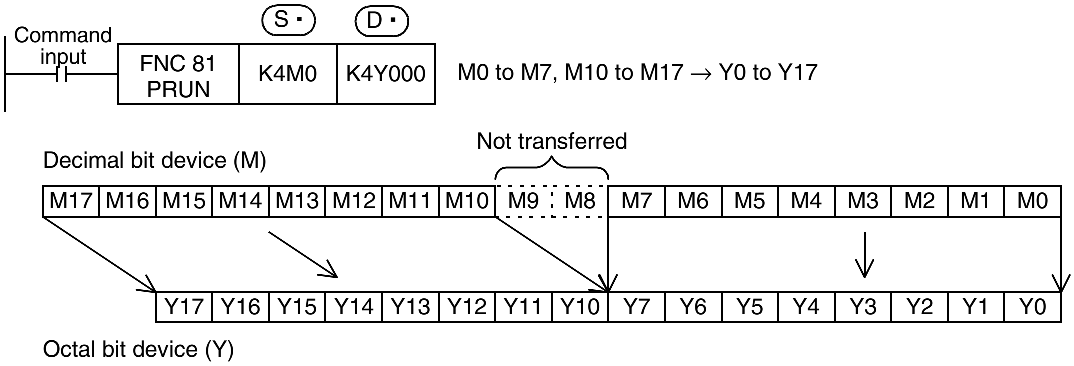
Octal bit device → Decimal bit device



X000 to X017 → M0 to M7, M10 to M17

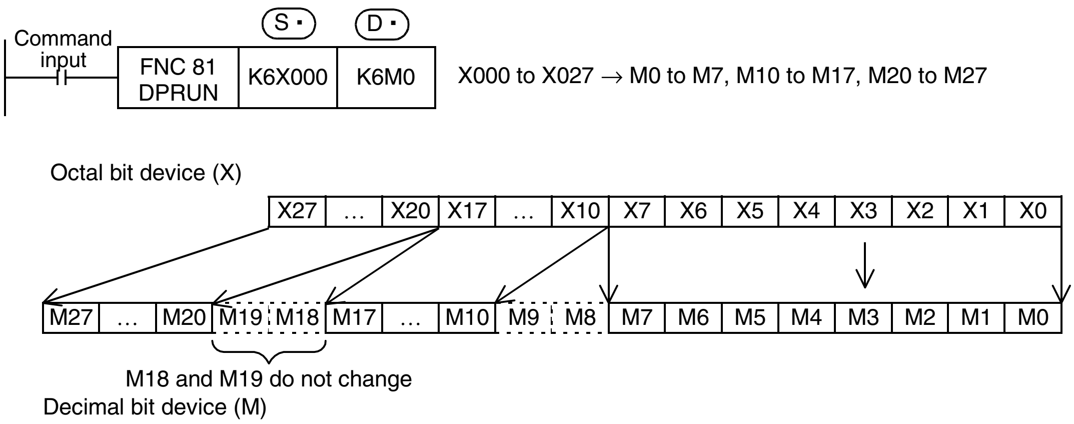


Decimal bit device → Octal bit device

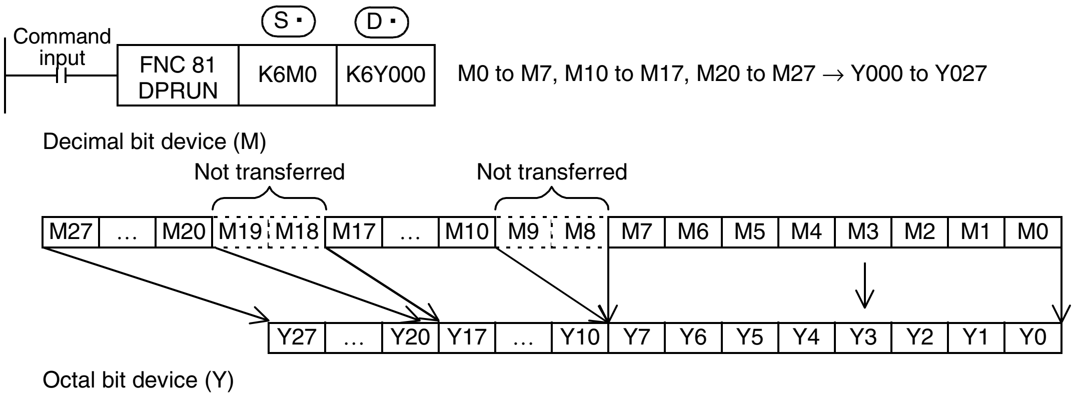


2. 32-bit operation (DPRUN and DPRUNP)

Octal bit device → Decimal bit device



Decimal bit device → Octal bit device



16.3 FNC 82 – ASCI / Hexadecimal to ASCII Conversion

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction converts hexadecimal code into ASCII code.
On the other hand, BINDA (FNC261) instruction converts binary data into ASCII code, and ESTR (FNC116) instruction converts binary floating point data into ASCII code.

→ For BINDA (FNC261) instruction, refer to Section 29.6.
→ For ESTR (FNC116) instruction, refer to Section 18.4.

1. Instruction format

FNC 82 ASCI P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	ASCI	Continuous Operation
	ASCIP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
S·	Head device number storing hexadecimal code to be converted	16-bit binary
D·	Head device number storing converted ASCII code	Character string (only ASCII code)
n	Number of characters (digits) of hexadecimal code to be converted [setting range: 1 to 256]	16-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D·									✓	✓	✓	✓	✓	✓	✓	✓			✓					
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (ASCI and ASCIP)

"n" hexadecimal code characters (digits) stored in S· and later are converted into ASCII code, and then stored to the devices D· and later.
The 16-bit mode and 8-bit mode options are available for this instruction. For operation in each mode, refer to the proceeding pages.

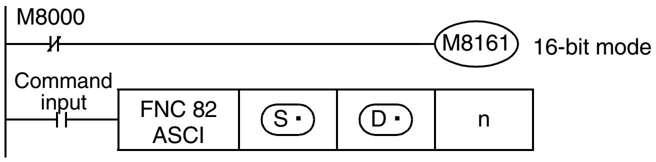
Command input

FNC 82 ASCI	S·	D·	n

2. 16-bit conversion mode (while M8161 is OFF) (M8161 is also used for the RS, HEX, CCD and CRC instructions.)

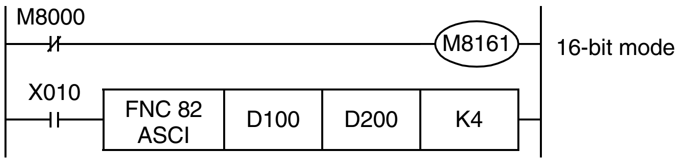
Each digit of hexadecimal data stored in (S·) and later is converted into ASCII code, and transferred to the high-order 8 bits and low-order 8 bits of each device (D·) and later. The number of digits (characters) to be converted is specified by "n".

Each ASCII code is stored in either the high-order 8 bits or low-order 8 bits of each device (D·) and later. M8161 is used also for RS, HEX, CCD and CRC instructions. When using the 16-bit mode, set M8161 to normally OFF. M8161 is cleared when the PLC mode is changed from RUN to STOP.



Operation

In the following program, conversion is executed as follows:



Devices after (S·)

D100 = 0ABCH
D101 = 1234H
D102 = 5678H

Number of specified digits (characters) and conversion result

n / (D·)	K1	K2	K3	K4	K5	K6	K7	K8	K9
Low-order 8 bits of D200	[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]	[8]
High-order 8 bits of D200		[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]
Low-order 8 bits of D201			[C]	[B]	[A]	[0]	[4]	[3]	[2]
High-order 8 bits of D201				[C]	[B]	[A]	[0]	[4]	[3]
Low-order 8 bits of D202					[C]	[B]	[A]	[0]	[4]
High-order 8 bits of D202						[C]	[B]	[A]	[0]
Low-order 8 bits of D203				Does not change			[C]	[B]	[A]
High-order 8 bits of D203								[C]	[B]
Low-order 8 bits of D204									[C]

Bit configuration in the case of "n = K4"

D 100 = 0ABCH

0	0	0	0	1	0	1	0	1	0	1	1	1	1	0	0
0				A				B				C			

D 200

0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0
[A] → 41H								[0] → 30H							

D 201

0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0
[C] → 43H								[B] → 42H							

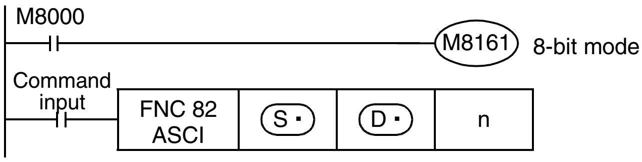
ASCII code

[0] = 30H [1] = 31H [5] = 35H
[A] = 41H [2] = 32H [6] = 36H
[B] = 42H [3] = 33H [7] = 37H
[C] = 43H [4] = 34H [8] = 38H

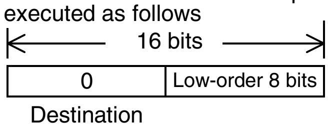
- When outputting data in the BCD format for a printer, for example, it is necessary to convert binary data into BCD data before executing this instruction.

3. 8-bit conversion mode (while M8161 is ON) (M8161 is used also for the RS, HEX, CCD and CRC instructions.)

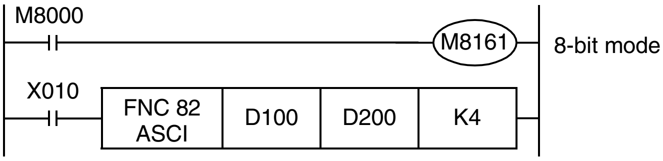
Each digit of hexadecimal data stored in (S·) and later is converted into an ASCII code, and transferred to low-order 8 bits of each device (D·) and later. The number of digits (characters) to be converted is specified by "n". "0" is stored in high-order 8 bits of each device (D·) and later.
M8161 is used also for the RS, HEX, CCD and CRC instructions. When using the 8-bit mode, set M8161 to normally ON. M8161 is cleared when the PLC mode is changed from RUN to STOP.



When M8161 is set to ON, the 8-bit mode is selected. The conversion processing is executed as follows



Operation
In the following program, conversion is executed as follows:



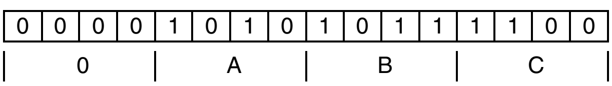
Devices after (S·)
D100 = 0ABCH
D101 = 1234H
D102 = 5678H

Number of specified digits (characters) and conversion result

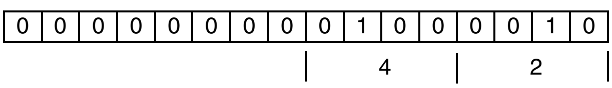
n / (D·)	K1	K2	K3	K4	K5	K6	K7	K8	K9
D 200	[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]	[8]
D 201		[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]
D 202			[C]	[B]	[A]	[0]	[4]	[3]	[2]
D 203				[C]	[B]	[A]	[0]	[4]	[3]
D 204					[C]	[B]	[A]	[0]	[4]
D 205						[C]	[B]	[A]	[0]
D 206		Does not change					[C]	[B]	[A]
D 207								[C]	[B]
D 208									[C]

Bit configuration in the case of "n = K2"

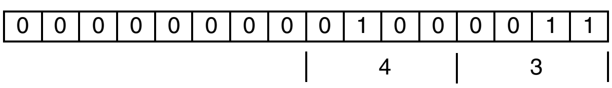
D 100 = 0ABCH



D200 = B → ASCII code = 42H



D201 = C → ASCII code = 43H



ASCII codes
[0] = 30H [1] = 31H [5] = 35H
[A] = 41H [2] = 32H [6] = 36H
[B] = 42H [3] = 33H [7] = 37H
[C] = 43H [4] = 34H [8] = 38H

- When outputting data in the BCD format for a printer, for example, it is necessary to convert binary data into BCD data before executing this instruction.

16.4 FNC 83 – HEX / ASCII to Hexadecimal Conversion

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction converts ASCII codes into hexadecimal codes.
On the other hand, DABIN (FNC260) instruction converts ASCII codes into binary data, and EVAL (FNC117) instruction converts ASCII codes into binary floating point data.

→ **For DABIN (FNC260) instruction, refer to Section 29.5.**
→ **For EVAL (FNC117) instruction, refer to Section 18.5.**

1. Instruction format

FNC 83
HEX P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	HEX	Continuous Operation
	HEXP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
S•	Head device number storing ASCII code to be converted	Character string (only ASCII code)*1
D•	Head device number storing converted hexadecimal code	16- or 32-bit binary
n	Number of ASCII codes (bytes) to be converted [setting range: 1 to 256]	16-bit binary

*1. Make sure to use only ASCII codes "0" to "9" and "A" to "F".

3. Applicable devices

Operand Type	Bit Devices: System User							Word Devices: Digit Specification				Word Devices: System User				Special Unit	Index			Others: Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓			
D•									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (HEX and HEXP)

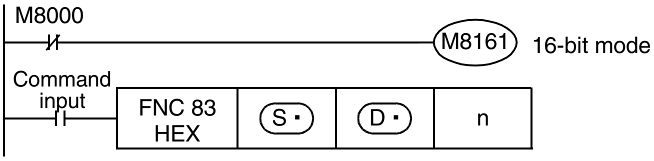
Among the ASCII codes stored in S• and later, "n" characters are converted into hexadecimal codes, and then stored to the devices D• and later.
The 16-bit mode and 8-bit mode are available for this instruction. For operation in each mode, refer to the proceeding pages.

Command input

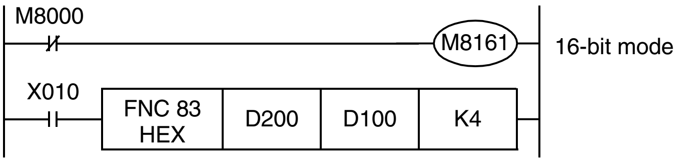
FNC 83 HEX	S•	D•	n

2. 16-bit conversion mode (while M8161 is OFF)
(M8161 is used also for the RS, ASCI, CCD, and CRC instructions.)

Each ASCII code stored in high-order 8 bits and low-order 8 bits of devices S· and later is converted into a hexadecimal code, and transferred to devices D· and later in units of 4 digits. The number of characters to be converted is specified by "n".
M8161 is used also for the RS, ASCI, CCD and CRC instructions. When using the 16-bit mode, set M8161 to normally OFF.
M8161 is cleared when the PLC mode is changed from RUN to STOP.



Operation
In the following program, conversion is executed as follows:

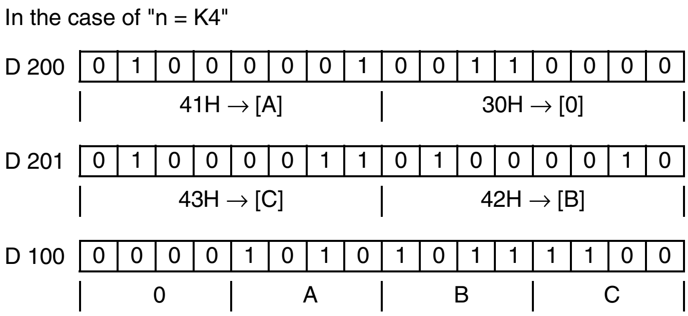


Conversion source data

S·	ASCII code	Hexadecimal code
Low-order 8 bits of D200	30H	0
High-order 8 bits of D200	41H	A
Low-order 8 bits of D201	42H	B
High-order 8 bits of D201	43H	C
Low-order 8 bits of D202	31H	1
High-order 8 bits of D202	32H	2
Low-order 8 bits of D203	33H	3
High-order 8 bits of D203	34H	4
Low-order 8 bits of D204	35H	5

Number of specified characters and conversion result
" • " indicates "0".

D· / n	D 102	D 101	D 100
1	Does not change		•••0H
2			••0AH
3			•0ABH
4			0ABCH
5		•••0H	ABC1H
6		••0AH	BC12H
7		•0ABH	C123H
8		0ABCH	1234H
9	•••0H	ABC1H	2345H

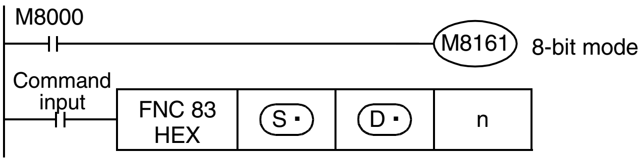


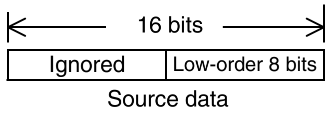
- When the input data is in BCD format, it is necessary to convert BCD data into binary data after executing this instruction.
- If ASCII code is not stored in S· in the HEX instruction, an operation error occurs and conversion into hexadecimal code is disabled. Especially, note that ASCII code should be stored in high-order 8 bits of S· also when M8161 is OFF.

3. 8-bit conversion mode (while M8161 is ON) (M8161 is used also for the RS, ASCI, CCD and CRC instructions.)

Each ASCII code stored in the low-order 8 bits of each device (S·) and later is converted into a hexadecimal code, and transferred to device (D·) and later in 4-digits units. The number of characters to be converted is specified by "n".

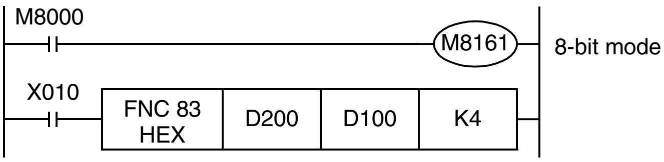
M8161 is also used for the RS, ASCI, CCD and CRC instructions. When using the 8-bit mode, set M8161 to normally ON. M8161 is cleared when the PLC mode is changed from RUN to STOP.





Operation

In the following program, conversion is executed as follows:



Conversion source data

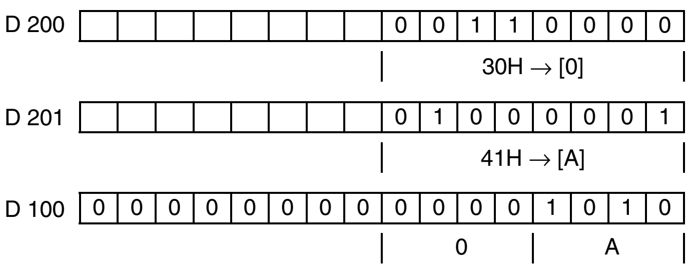
(S·)	ASCII code	Hexadecimal code
D 200	30H	0
D 201	41H	A
D 202	42H	B
D 203	43H	C
D 204	31H	1
D 205	32H	2
D 206	33H	3
D 207	34H	4
D 208	35H	5

Number of specified characters and conversion result

" • " indicates "0".

(D·) / n	D 102	D 101	D 100
1	Does not change		•••0H
2			••0AH
3			•0ABH
4			0ABCH
5		•••0H	ABC1H
6		••0AH	BC12H
7		•0ABH	C123H
8		0ABCH	1234H
9	•••0H	ABC1H	2345H

In the case of "n = K2"



- When the input data is in BCD format, it is necessary to convert BCD data into binary data after executing this instruction.

16.5 FNC 84 – CCD / Check Code

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction calculates the horizontal parity value and sum check value in the error check methods used in communication. There is another check method, CRC (cyclic redundancy check) also. For obtaining CRC value, use CRC instruction.

→ For CRC instruction, refer to Section 24.4.
→ For complement [NEG (FNC 29) instruction], refer to Section 10.10.

1. Instruction format

FNC 84 CCD P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	CCD	Continuous Operation
	CCDP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
(S•)	Head device number of applicable device	16-bit binary or character string
(D•)	Head device number storing the calculated data	16-bit binary or character string
n	Number of data [setting range: 1 to 256]	16-bit binary

3. Applicable devices

Operand Type	Bit Devices: System User X	Y	M	T	C	S	D□.b	Word Devices: Digit Specification KnX	KnY	KnM	KnS	System User T	C	D	R	Special Unit U□\G□	Index V	Z	Modify	Others: Constant K	H	Real Number E	Character String "□"	Pointer P
(S•)								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
(D•)									✓	✓	✓	✓	✓	✓	✓				✓					
n														✓	✓					✓	✓			

Explanation of function and operation

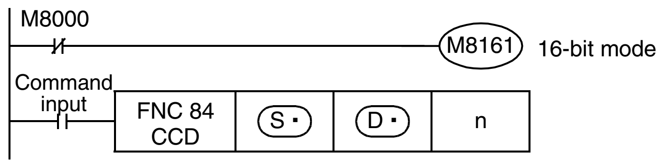
1. 16-bit operation (CCD and CCDP)

The addition data and horizontal parity value of data stored in (S•) to (S•)+n-1 are calculated. The addition data is stored to (D•), and the horizontal parity value is stored to (D•)+1.
The 16-bit mode and 8-bit mode are available in this instruction. For the operation in each mode, refer to the proceeding pages.

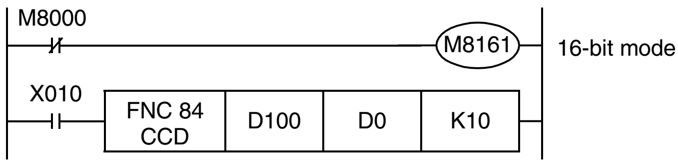
Command input	FNC 84 CCD	(S•)	(D•)	n

2. 16-bit conversion mode (while M8161 is OFF) (M8161 is also used for the RS, ASCI, HEX and CRC instructions.)

With regard to "n" data starting from (S·), the addition data and horizontal parity data of high-order 8 bits and low-order 8 bits are stored to (D·) and (D·)+1 respectively.
M8161 is used also for the RS, ASCI, HEX and CRC instructions. When using the 16-bit mode, set M8161 to normally OFF. M8161 is cleared when the PLC mode is changed from RUN to STOP.



Example of 16-bit conversion
In the following program, conversion is executed as follows:



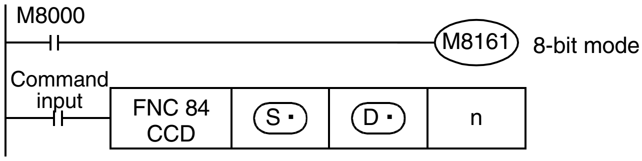
(S·)	Example of data contents	
Low-order 8 bits of D100	K100 = 01100100	
High-order 8 bits of D100	K111 = 0110111 (1)	←
Low-order 8 bits of D101	K100 = 01100100	
High-order 8 bits of D101	K 98 = 01100010	
Low-order 8 bits of D102	K123 = 0111101 (1)	←
High-order 8 bits of D102	K 66 = 01000010	
Low-order 8 bits of D103	K100 = 01100100	
High-order 8 bits of D103	K 95 = 0101111 (1)	←
Low-order 8 bits of D104	K210 = 11010010	
High-order 8 bits of D104	K 88 = 01011000	
Total	K1091	
Horizontal parity	1000010 (1)	←

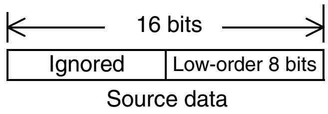
When the number of "1" is odd, the horizontal parity is "1".
When the number of "1" is even, the horizontal parity is "0".

D 0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	⟸ "1091" in BCD
D 1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	⟸ Horizontal parity

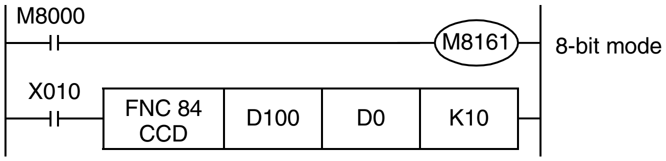
3. 8-bit conversion mode (while M8161 is ON)
(M8161 is used also for the RS, ASCI, HEX and CRC instructions.)

With regard to "n" data starting from (S·), the addition data and horizontal parity data of only low-order 8 bits are stored to (D·) and (D·)+1 respectively.
M8161 is also used for the RS, ASCI, HEX and CRC instructions. When using the 8-bit mode, set M8161 to normally ON. M8161 is cleared when the PLC mode is changed from RUN to STOP.





Example of 8-bit conversion
In the following program, conversion is executed as follows:



(S·)	Example of data contents	
D 100	K100 = 01100100	
D 101	K111 = 0110111 (1)	←
D 102	K100 = 01100100	
D 103	K 98 = 01100010	
D 104	K123 = 0111101 (1)	←
D 105	K 66 = 01000010	
D 106	K100 = 01100100	
D 107	K 95 = 0101111 (1)	←
D 108	K210 = 11010010	
D 109	K 88 = 01011000	
Total	K1091	
Horizontal parity	1000010 (1)	← When the number of "1" is odd, the horizontal parity is "1". When the number of "1" is even, the horizontal parity is "0".

D 0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	⟸ "1091" in BCD
D 1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	⟸ Horizontal parity

16.6 FNC 87 – RS2 / Serial Communication 2

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction sends and receives data in no-protocol communication by way of serial ports in accordance with RS-232C or RS-485 provided in the main unit.

1. Instruction format

FNC 87 RS2

16-bit Instruction	Mnemonic	Operation Condition
11 steps	RS2	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand type	Description	Data type
S•	Head device of data registers storing data to be sent	16-bit binary or character string
m	Number of bytes of data to be sent [setting range: 0 to 4,096]	16-bit binary
D•	Head device of data registers storing received data when receiving is completed	16-bit binary or character string
n	Number of bytes to be received [setting range: 0 to 4,096]	16-bit binary
n1	Used channel number [contents of setting: K1 = ch 1, K2 = ch 2]	16-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•														✓	✓				✓					
m														✓	✓					✓	✓			
D•														✓	✓				✓					
n														✓	✓					✓	✓			
n1																				✓	✓			

Explanation of function and operation

1. 16-bit operation (RS2)

This instruction sends and receives data in no-protocol communication by way of serial ports in accordance with RS-232C or RS-485 provided in the main unit.

→ **For detailed explanation, refer to the Data Communication Edition.**

Command input	FNC 87 RS2	S•	m	D•	n	n1

Related devices

→ For detailed explanation, refer to the Data Communication Edition.

Device ch1	Device ch2	Name
M8401	M8421	Sending wait flag*1
M8402	M8422	Sending request*1
M8403	M8423	Receiving complete flag*1
M8404	M8424	Carrier detection flag
M8405	M8425	Data Set Ready (DSR) Flag *2
–	–	–
M8409	M8429	Timeout evaluation flag
–	–	–
M8063	M8438	Serial communication error

Device ch1	Device ch2	Name
D8400	D8420	Communication format setting
–	–	
D8402	D8422	Remaining number of data to be sent*1
D8403	D8423	Monitor for number of received data*1
D8405	D8425	Communication parameter display
D8409	D8429	Timeout time setting
D8410	D8430	Header 1, 2
D8411	D8431	Header 3, 4
D8412	D8432	Terminator 1, 2
D8413	D8433	Terminator 3, 4
D8414	D8434	Receiving sum (received data)
D8415	D8435	Receiving sum (calculation result)
D8416	D8436	Sending sum
D8419	D8439	Operation mode display
D8063	D8438	Error code number of serial communication error

*1. Cleared when the PLC mode is changed from RUN to STOP. *2 Available in Ver. 2.30 or later.

System configuration

For using this instruction, it is necessary to attach either product shown in the table below to the main unit.

→ For the system configuration, refer to the Hardware Edition of the used PLC.
→ For detailed explanation, refer to the Data Communication Edition.

Communication type	Option
RS-232C communication	FX3U-232-BD or FX3U-232ADP
RS-485 communication	FX3U-485-BD or FX3U-485ADP

Differences between RS (FNC 80) instruction and RS2 (FNC 87) instruction

Item	RS2 instruction	RS instruction	Remarks
Header size	1 to 4 characters (bytes)	Up to 1 character (byte)	In RS2 instruction, up to 4 characters (bytes) can be specified as a header or terminator.
Terminator size	1 to 4 characters (bytes)	Up to 1 character (byte)	
Attachment of check sum	The check sum can be automatically attached.	The check sum should be attached by a user program.	In RS2 instruction, the check sum can be automatically attached to the sent and received data. In this case, however, make sure to use a terminator in the communication frame to be sent and received.

Cautions

→ For other cautions, refer to the Data Communication Edition.

- Do not drive two or more RS (FNC 80) and/or RS2 (FNC 87) instructions for the same port at the same time.
- It is not permitted to use RS (FNC 80)/RS2 (FNC 87) instruction and IVCK (FNC270)/IVDR (FNC271)/ IVRD (FNC272)/IVWR (FNC273)/IVBWzR (FNC274) instruction for the same port.

Function Changes According to Versions

Compatible Versions FX 3U	Compatible Versions FX3UC	Item	Function Summary
Ver. 2.30 or later	Ver. 2.30 or later	ch1 Data Set Ready (DSR) Flag	Turns the special device M8405 ON when the DR (DSR) signal of ch1 is ON.
		ch2 Data Set Ready (DSR) Flag	Turns the special device M8425 ON when the DR (DSR) signal of ch2 is ON.

16.7 FNC 88 – PID / PID Control Loop

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes PID control which changes the output value according to the input variation.

→ **For details, refer to the Analog Control Edition.**

1. Instruction format

FNC 88
PID

16-bit Instruction	Mnemonic	Operation Condition
9 steps	PID	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand type	Description	Data type
S1	Data register number storing the target value (SV)	16-bit binary
S2	Data register number storing the measured value (PV)	16-bit binary
S3	Data register number storing a parameter	16-bit binary
D	Data register number storing the output value (MV)	16-bit binary

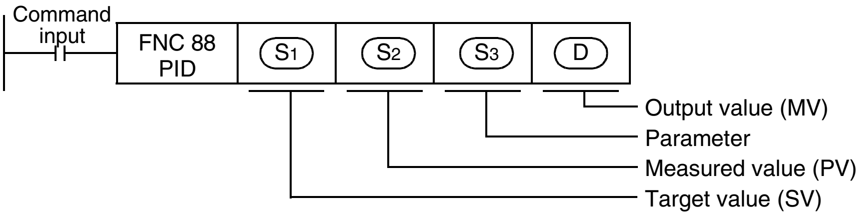
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1														✓	✓	✓								
S2														✓	✓	✓								
S3														✓	✓									
D														✓	✓	✓								

Explanation of function and operation

1. 16-bit operation (PID)

When the target value S1, measured value S2, and parameters S3 to S3+6 are set and a program is executed, the operation result (MV) is stored to the output value D at every sampling time S3.



2. Set items

Set item		Description	Number of occupied points
S1	Target value (SV)	• Set the target value (SV). • PID instruction does not change the contents of setting. • Caution on using the auto tuning (limit cycle method) If the target value for auto tuning is different from the target value for PID control, it is necessary to set a value including the bias value first, and then store the actual target value when the auto tuning flag turns OFF.	1
S2	Measured value (PV)	This is the input value in PID control loop.	1
S3	Parameter*1	1) Auto tuning: In the case of limit cycle method Twenty-nine devices are occupied from the head device specified in S3. 2) Auto tuning: In the case of step response method a) Operation setting (ACT): When bits 1, 2 and 5 are not all "0" Twenty-five devices are occupied from the head device specified in S3. b) Operation setting (ACT): When bits 1, 2 and 5 are all "0" Twenty devices are occupied from the head device specified in S3.	29 25 20
D	Output value (MV)	1) In case of PID control (normal processing) Before driving PID instruction, the user should set the initial output value. After that, the operation result is stored. 2) Auto tuning: In the case of limit cycle method During auto tuning, the ULV or LLV value is output automatically. When auto tuning is finished, the specified MV value is set. 3) Auto tuning: In the case of step response method Before driving PID instruction, the user should set the initial output value. During auto tuning, PID instruction does not change the MV output.	1

*1. When auto tuning is not used, the number of points is the same as the number in the step response method are occupied.

3. List of parameters S3 to S3+28

Set item			Setting Value	Remarks
S3	Sampling time (Ts)		1 to 32767 (ms)	It cannot be shorter than the operation cycle.
S3+1	Operation setting (ACT)	bit0	0: Forward operation, 1: Backward operation	Operation direction
		bit1	0: Input variation alarm is invalid. 1: Input variation alarm is valid.	
		bit2	0: Output variation alarm is invalid. 1: Output variation alarm is valid.	Do not set to ON bit 2 and bit 5 at the same time.
		bit3	Not available	
		bit4	0: Auto tuning is not executed. 1: Auto tuning is executed.	
		bit5	0: Upper and lower limits of output value are not valid. 1: Upper and lower limits of output value are valid.	Do not set to ON bit 2 and bit 5 at the same time.
		bit6	0: Step response method 1: Limit cycle method	Select the auto tuning mode.
		bit7 to bit15	Not available	
S3+2	Input filter constant (α)		0 to 99 (%)	When "0" is set, the input filter is not provided.
S3+3	Proportional gain (KP)		1 to 32767 (%)	
S3+4	Integral time (TI)		0 to 32767 ($\times$ 100 ms)	When "0" is set, it is handled as "\" (no integration).
S3+5	Derivative gain (KD)		0 to 100 (%)	When "0" is set, the derivative gain is not provided.
S3+6	Derivative time (TD)		0 to 32767 ($\times$ 10 ms)	When "0" is set, the derivative operation is not executed.

Set item			Setting Value	Remarks
S3 +7 : S3 +19	These devices are occupied for internal processing in PID control loop. Do not change the data.			
S3 +20*1	Input variation (incremental) alarm set value		0 to 32767	It is valid when bit 1 is set to "1" in S3 +1 for the operation setting (ACT).
S3 +21*1	Input variation (decremental) alarm set value		0 to 32767	It is valid when bit 1 is set to "1" in S3 +1 for the operation setting (ACT).
S3 +22*1	Output variation (incremental) alarm set value		0 to 32767	It is valid when bit 2 is set to "1" and bit 5 is set to "0" in S3 +1 for the operation setting (ACT).
	Output upper limit set value		–32768 to 32767	It is valid when bit 2 is set to "0" and bit 5 is set to "1" in S3 +1 for the operation setting (ACT).
S3 +23*1	Output variation (decremental) alarm set value		0 to 32767	It is valid when bit 2 is set to "1" and bit 5 is set to "0" in S3 +1 for the operation setting (ACT).
	Output lower limit set value		–32768 to 32767	It is valid when bit 2 is set to "0" and bit 5 is set to "1" in S3 +1 for the operation setting (ACT).
S3 +24*1	Alarm output	bit0	0: Input variation (incremental) is not exceeded. 1: Input variation (incremental) is exceeded.	It is valid when bit 1 is set to "1" or bit 2 is set to "1" in S3 +1 for the operation setting (ACT).
		bit1	0: Input variation (decremental) is not exceeded. 1: Input variation (decremental) is exceeded.	
		bit2	0: Output variation (incremental) is not exceeded. 1: Output variation (incremental) is exceeded.	
		bit3	0: Output variation (decremental) is not exceeded. 1: Output variation (decremental) is exceeded.	
The setting below is required when the limit cycle method is used (when bit 6 is set to "ON" in the operation setting (ACT)).				
S3 +25	PV value threshold (hysteresis) width (SHPV)		Set it according to the fluctuation of the measured value (MV).	They are occupied when bit 6 is set to "ON (limit cycle method)" in the operation setting (ACT).
S3 +26	Output value upper limit (ULV)		Set the maximum value (ULV) of the output value (MV).	
S3 +27	Output value lower limit (LLV)		Set the minimum value (LLV) of the output value (MV).	
S3 +28	Wait setting from end of tuning cycle to start of PID control (Kw)		–50 to 32717%	

*1. S3 +20 to S3 +24 are occupied when any bit 1, 2 or 5 is set to "1" in S3 +1 for operation setting (ACT).

Cautions

1. When using two or more PID instructions

Two or more PID instructions can be executed at the same time. (There is no limitation in the number of loops.) However, make sure that S3, D and other operands specified in each instruction are different to each other.

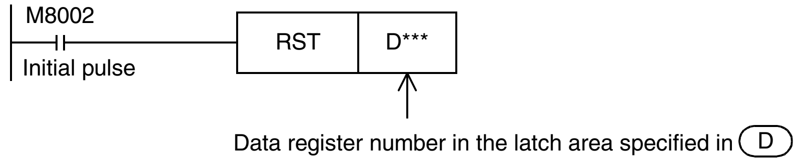
2. Number of devices occupied for parameters starting from S3

1) In the limit cycle method
   - Twenty-nine devices are occupied from the head device specified in S3.
2) In the step response method
   - Operation setting (ACT): When bits 1, 2 and 5 are not all "0"
     Twenty-five devices are occupied from the head device specified in S3.
   - Operation setting (ACT): When bits 1, 2 and 5 are all "0"
     Twenty devices are occupied from the head device specified in S3.

3. When specifying a device in the latch area backed up by the battery

For the output value (MV) in PID instruction, specify a data register (D) in the non-latch area.
(When specifying a data register in the latch area, make sure to clear the latched (battery backed) contents when the PLC mode is set to RUN using the following program.)

Program example



Error

When an operation error occurs, the special auxiliary relay M8067 turns ON, and the error code is stored in the special data register D8067.

→ **For the error code, refer to Section 37.4.**

17. Data Transfer 2 – FNC100 to FNC109

FNC100 to FNC109 provide an instruction for executing complicated processing for fundamental applied instructions and for executing special processing.

FNC No.	Mnemonic	Symbol	Function	Reference
100	–			
101	–			
102	ZPUSH	ZPUSH D	Batch Store of Index Register	Section 17.1
103	ZPOP	ZPOP D	Batch POP of Index Register	Section 17.2
104	–			
105	–			
160	–			
107	–			
108	–			
109	–			

17.1 FNC102 – ZPUSH/Batch Store of Index Register

Outline

This instruction temporarily batch-stores the present value of the index registers V0 to V7 and Z0 to Z7.
For restoring the present value of temporarily batch-stored index registers, use ZPOP (FNC103) instruction.

→ For ZPOP (FNC103) instruction, refer to Section 17.2.

1. Instruction format

FNC 102 ZPUSH P

16-bit Instruction	Mnemonic	Operation Condition
3 steps	ZPUSH	Continuous Operation
	ZPUSHP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

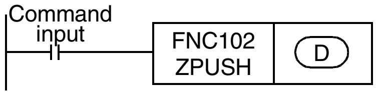
Operand Type	Description	Data Type
D	Head device number batch-storing the present value of the index registers V0 to V7 and Z0 to Z7 D: Number of times of batch-storage D+1 to D+16 × Number of times of batch-storage: Batch-stored data storage destination	16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
D														✓	✓									

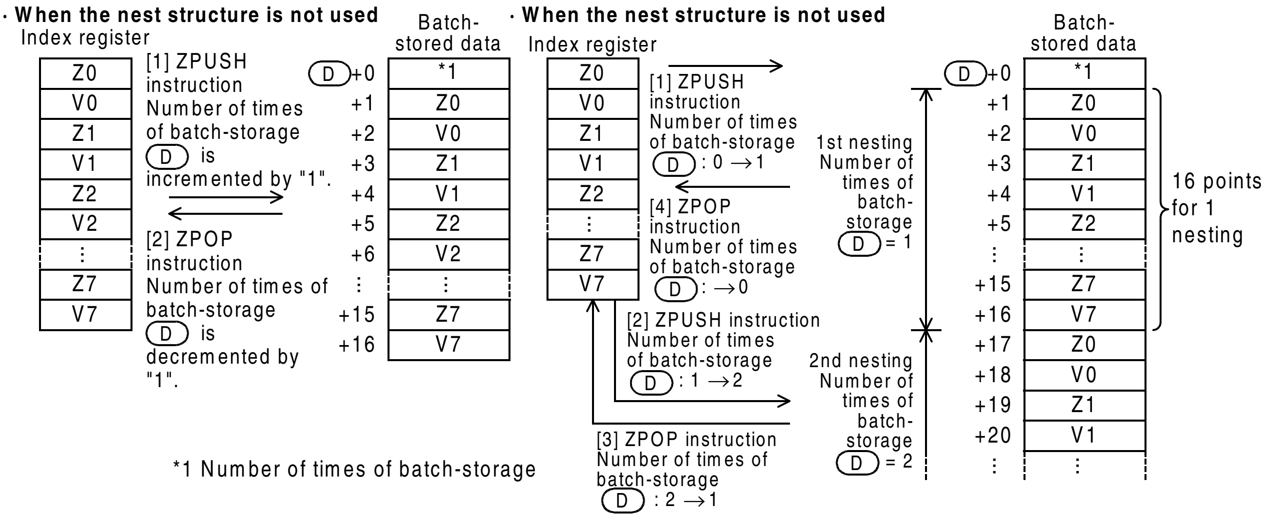
Explanation of function and operation

1. 16-bit operation (ZPUSH/ZPUSHP)



1) The contents of the index registers V0 to V7 and Z0 to Z7 are batch-stored temporarily to D and later.
   When the contents of index registers are batch-stored, the number of times of batch-storage D is incremented by "1".
2) For restoring the batch-stored data, use ZPOP (FNC103) instruction.
   Use ZPUSH (FNC102) and ZPOP (FNC103) instruction as a pair.
3) By specifying a same device to D, ZPUSH (FNC102) and ZPOP (FNC103) instructions can be used in the nest structure.
   In this case, the occupied points are added by "16" after D every time ZPUSH (FNC102) instruction is executed. Secure in advance sufficient area for the number of the next structure.

4) The figure below shows the data structure batch-stored in (D) and later.



Related instruction

Instruction	Description
ZPOP(FNC103)	Restores the index registers V0 to V7 and Z0 to Z7 which were batch-stored temporarily by ZPUSH (FNC102) instruction.

Cautions

- When not using the nest structure, clear the number of times of batch-storage (D) before executing ZPUSH (FNC102) instruction.
- When using the nest structure, clear the number of times of batch-storage (D) before executing ZPUSH (FNC102) instruction for the first time.

Errors

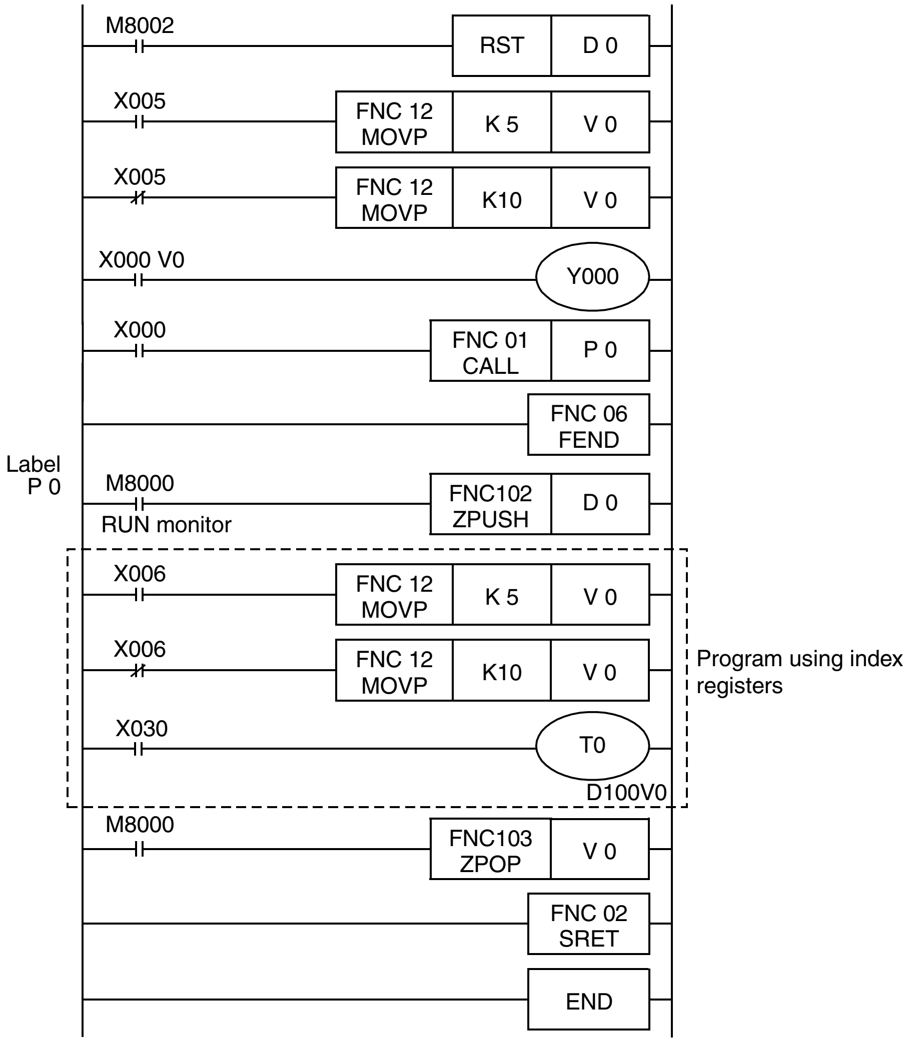
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the range of points used after (D) in ZPUSH (FNC102) instruction exceeds the corresponding device range (error code: K6706)
- When the number of times of batch-storage (D) stores a negative value while ZPUSH (FNC102) instruction is executed (error code: K6707)

11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FNC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

Program example

In the program shown below, the contents of the index registers Z0 to Z7 and V0 to V7 before execution of subroutine program are batch-stored in D0 and later when index registers are used in the subroutine after the pointer P0.



17.2 FNC103 – ZPOP/Batch POP of Index Register

Outline

FX3UC Ver.2.20 ⇒ FX3U Ver.2.20 ⇒

This instruction restores the contents of the index registers V0 to V7 and Z0 to Z8 which were batch-stored temporarily by ZPUSH (FNC102) instruction.

→ **For ZPUSH (FNC102) instruction, refer to Section 17.1**

1. Instruction format

FNC 103 ZPOP	P

16-bit Instruction	Mnemonic	Operation Condition
3 steps	ZPOP	Continuous Operation
	ZPOPP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand Type	Description	Data Type
(D)	Head device number temporarily batch-storing the contents of the index registers V0 to V7 and Z0 to Z7 (D): Number of times of batch-storage (D)+1 to (D)+16 × Number of times of batch-storage: Batch-stored data storage destination	16-bit binary

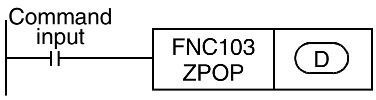
3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index				Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(D)														✓	✓									

Explanation of function and operation

1. 16-bit operation (ZPOP/ZPOPP)

→ **For the function and operation, refer also to Section 17.1.**



1) The contents of the index registers V0 to V7 and Z0 to Z7 which were batch-stored temporarily to (D) and later are restored to the original index registers. When the contents of the index registers are restored, the number of times of batch-storage (D) is decremented by "1".
2) For temporarily batch-storing the data, use ZPUSH (FNC102) instruction.
Use ZPUSH (FNC102) and ZPOP (FNC103) instruction as a pair.

Related instruction

Instruction	Description
ZPUSH(FNC102)	Temporarily batch-stores the present value of the index registers V0 to V7 and Z0 to Z7.

Errors

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the number of times of batch-storage (D) stores "0" or a negative value while ZPOP (FNC103) instruction is executed (error code: K6706)

Program example

→ **For a program example, refer to Section 17.1.**

11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FNC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

18. Floating Point – FNC110 to FNC139

FNC110 to FNC119, FNC120 to FNC129 and FNC130 to FNC139 provide instructions for conversion, comparison, arithmetic operations, square root operation, trigonometry, etc. for floating point operations.

FNC No.	Mnemonic	Symbol	Function	Reference
110	ECMP	ECMP S1 S2 D	Floating Point Compare	Section 18.1
111	EZCP	EZCP S1 S2 S D	Floating Point Zone Compare	Section 18.2
112	EMOV	EMOV S D	Floating Point Move	Section 18.3
113	–			–
114	–			–
115	–			–
116	ESTR	ESTR S1 S2 D	Floating Point to Character String Conversion	Section 18.4
117	EVAL	EVAL S D	Character String to Floating Point Conversion	Section 18.5
118	EBCD	EBCD S D	Floating Point to Scientific Notation Conversion	Section 18.6
119	EBIN	EBIN S D	Scientific Notation to Floating Point Conversion	Section 18.7
120	EADD	EADD S1 S2 D	Floating Point Addition	Section 18.8
121	ESUB	ESUB S1 S2 D	Floating Point Subtraction	Section 18.9
122	EMUL	EMUL S1 S2 D	Floating Point Multiplication	Section 18.10
123	EDIV	EDIV S1 S2 D	Floating Point Division	Section 18.11
124	EXP	EXP S D	Floating Point Exponent	Section 18.12
125	LOGE	LOGE S D	Floating Point Natural Logarithm	Section 18.13
126	LOG10	LOG10 S D	Floating Point Common Logarithm	Section 18.14
127	ESQR	ESQR S D	Floating Point Square Root	Section 18.15
128	ENEG	ENEG D	Floating Point Negation	Section 18.16

FNC No.	Mnemonic	Symbol	Function	Reference
129	INT	INT S D	Floating Point to Integer Conversion	Section 18.17
130	SIN	SIN S D	Floating Point Sine	Section 18.18
131	COS	COS S D	Floating Point Cosine	Section 18.19
132	TAN	TAN S D	Floating Point Tangent	Section 18.20
133	ASIN	ASIN S D	Floating Point Arc Sine	Section 18.21
134	ACOS	ACOS S D	Floating Point Arc Cosine	Section 18.22
135	ATAN	ATAN S D	Floating Point Arc Tangent	Section 18.23
136	RAD	RAD S D	Floating Point Degrees to Radians Conversion	Section 18.24
137	DEG	DEG S D	Floating Point Radians to Degrees Conversion	Section 18.25
138	–			–
139	–			–

18.1 FNC110 – ECMP / Floating Point Compare

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction compares two data (binary floating point), and outputs the result (larger, same or smaller) to three single bit devices.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D	FNC 110 ECMP	P

16-bit Instruction	Mnemonic	Operation Condition
	–	

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DECMP	Continuous Operation
	DECMPP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(S1•)	Device number storing binary floating point data to be compared	Real number (binary)*1
(S2•)	Device number storing binary floating point data to be compared	
(D•)	Head bit device number to which the comparison result is output (Three devices are occupied.)	Bit

*1. When a constant (K or H) is specified, it is automatically converted from binary into binary floating point (real number) when the instruction is executed.

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S1•)														✓	✓	✓			✓	✓	✓	✓		
(S2•)														✓	✓	✓			✓	✓	✓	✓		
(D•)		✓	✓			✓	▲												✓					

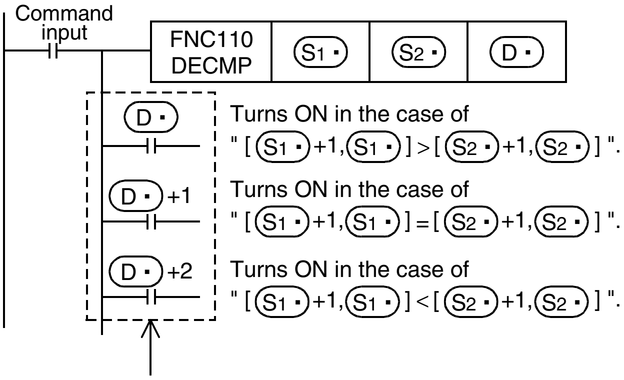
▲: "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

1. 32-bit operation (DECMP and DECMPP)

The comparison value [(S1•)+1, (S1•)] is compared with the comparison source [(S2•)+1, (S2•)] as floating point data, and either bit among (D•), (D•)+1 and (D•)+2 turns ON according to the result (smaller, same or larger).

- When a constant (K or H) is specified as [S1• +1, S1•] or [S2• +1, S2•], it is automatically converted from binary into binary floating point (real number) when the instruction is executed.



Even if the command input turns OFF and DECMP instruction is not executed, D• to D• +2 hold the status before the please rewrite this command input turned OFF.

Caution

1. Number of occupied devices

Three devices are occupied from D• (D•, D• +1 and D• +2).
Make sure that the devices are not used for any other purpose.

18.2 FNC111 – EZCP / Floating Point Zone Compare

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction compares data (binary floating point) with two values (one zone), and outputs the comparison result to three single bit devices.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

	FNC 111	
D	EZCP	P

16-bit Instruction	Mnemonic	Operation Condition
	–	

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DEZCP	Continuous Operation
	DEZCPP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S1•	Data register number storing binary floating point data to be compared	Real number (binary)*1
S2•	Data register number storing binary floating point data to be compared	
S•	Data register number storing binary floating point data to be compared	
D•	Head bit device number to which the comparison result is output (Three devices are occupied.)	Bit

*1. When a constant (K or H) is specified, it is automatically converted from binary into binary floating point (real number) when the instruction is executed.

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•														✓	✓	✓			✓	✓	✓	✓		
S2•														✓	✓	✓			✓	✓	✓	✓		
S•														✓	✓	✓			✓	✓	✓	✓		
D•		✓	✓			✓	▲												✓					

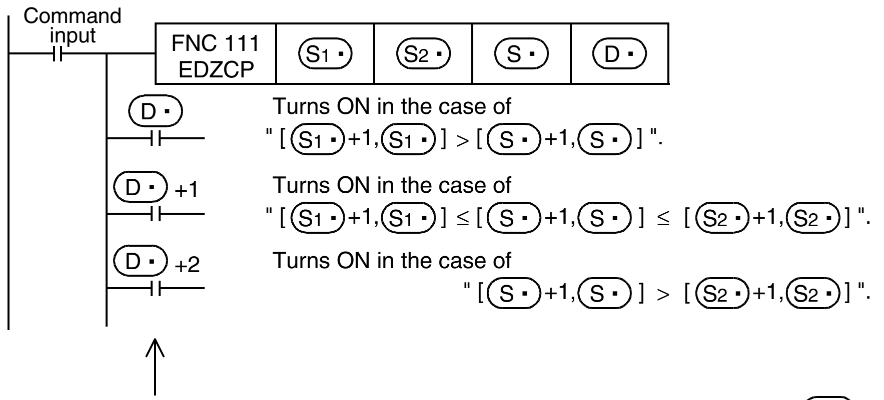
▲: "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

1. 32-bit operation (DEZCP and DEZCPP)

The comparison values [S1· +1, S1·], [S2· +1, S2·] are compared with the comparison source [S· +1, S·] as floating point data, and either bit among D·, D· +1, and D· +2 turns ON according to the result (smaller, same or larger).

- When a constant (K or H) is specified as [S1· +1, S1·], [S2· +1, S2·], or [S· +1, S·], it is automatically converted into binary floating point when the instruction is executed.



Even if the command input turns OFF and DEZCP instruction is not executed, D· to D· +2 hold the status before the command input turned OFF.

Cautions

1. Number of occupied devices

Three devices are occupied from D· (D·, D· +1 and D· +2).
Make sure that these devices are not used for any other purpose.

2. Comparison values [S1· +1, S1·] and [S2· +1, S2·]

Make sure that two comparison values have the following relationship:
[(S1· +1, S1·] ≤ [S2· +1, S2·]
In the case of "[S1· +1, S1·] > [S2· +1, S2·]", the value [S2· +1, S2·] is regarded as [S1· +1, S1·] value during comparison.

18.3 FNC112 – EMOV / Floating Point Move

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction transfers binary floating point data.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D	FNC 112 EMOV	P

16-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DEMOV	Continuous Operation
	DEMOVP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Binary floating point data (transfer source) or device number storing data	Real number (binary)
D·	Device number receiving floating point data	

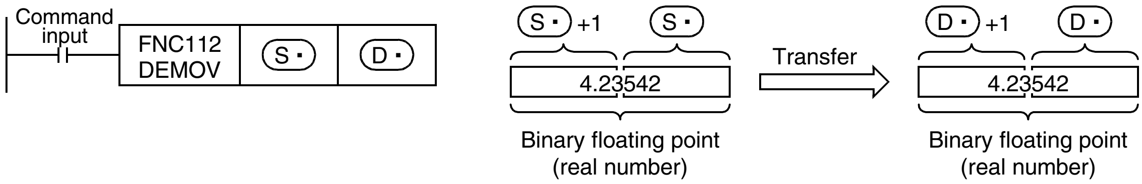
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·														✓	✓	✓			✓			✓		
D·														✓	✓	✓			✓					

Explanation of function and operation

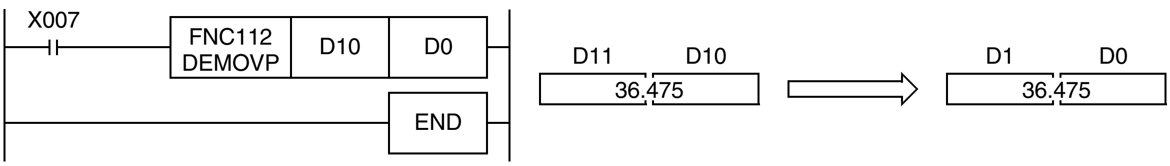
1. 32 bit operation (DEMOV and DEMOPV)

The contents (binary floating point data) of the transfer source [S· +1, S·] are transferred to [D· +1, D·]. A real number (E) can be directly specified as S·.

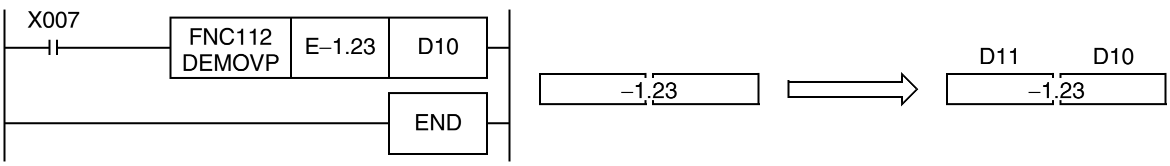


Program examples

1. In the program example shown below, a real number stored in D11 and D10 is transferred to D1 and D0 when X007 turns ON



2. In the program shown below, a real number "-1.23" is transferred to D11 and D10 when X007 turns ON



18.4 FNC116 – ESTR / Floating Point to Character String Conversion

Outline

FX3UC Ver.1.00 → | FX3U Ver.2.20 →

This instruction converts binary floating point data into a character string (ASCII codes) having a specified number of digits.
On the other hand, STR (FNC200) instruction converts binary data into a character string (ASCII codes).

→ **For a character string, refer to Section 5.3.**
→ **For handling of floating point, refer to Subsection 5.1.3.**
→ **For STR (FNC200) instruction, refer to Section 26.1.**

1. Instruction format

D	FNC 116 ESTR	P

16-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DESTR	Continuous Operation
	DESTRP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(S1•)	Binary floating point data to be converted or device storing data	Real number (binary)
(S2•)	Head device number storing the display specification of a numeric value to be converted	16-bit binary
(D•)	Head device number storing converted character string	Character string

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S1•)														✓	✓	✓			✓			✓		
(S2•)								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
(D•)									✓	✓	✓	✓	✓	✓	✓	✓			✓					

Explanation of function and operation

1. 32-bit operation (DESTR and DESTRP)

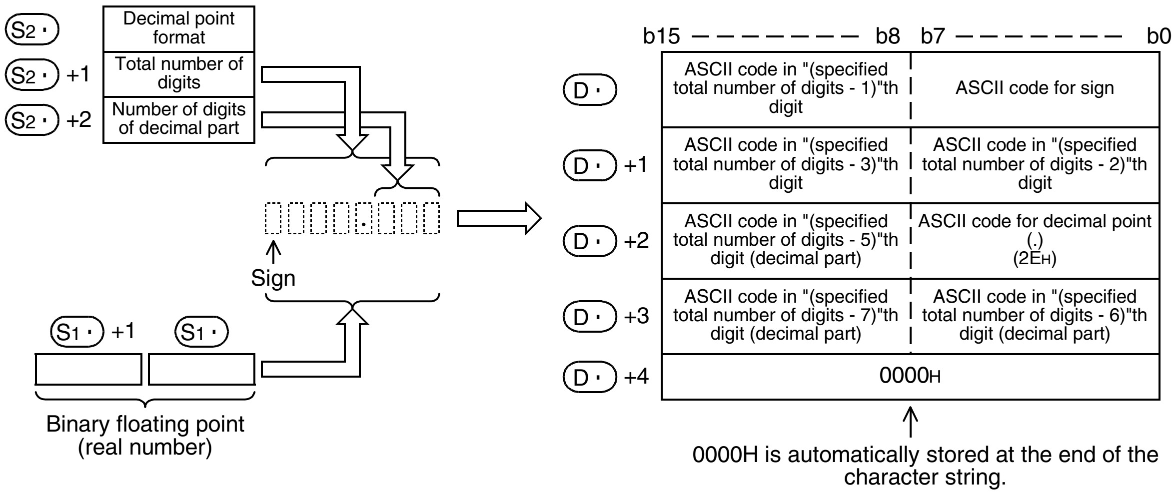
The contents (binary floating point data) of [(S1•)+1, (S1•)] are converted into a character string according to the contents specified by (S2•), (S2•)+1 and (S2•)+2, and then stored to devices (D•) and later. A real number can be directly specified as (S1•).

Command input —| |— [FNC116 DESTR | (S1•) | (S2•) | (D•)]

- The data after conversion varies depending on the display specification stored in (S2•).

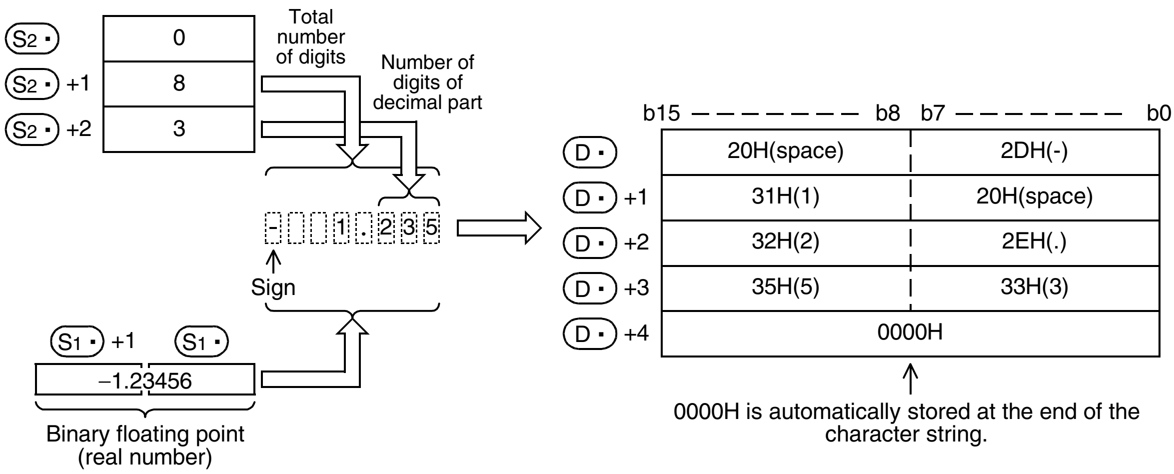
(S2•)	0: Decimal point format 1: Exponent format	The data after conversion varies depending on the display specification stored in (S2•).
(S2•)+1	Total number of digits	2 to 24 can be set.
(S2•)+2	Number of digits of decimal part	

2. In the case of decimal point format

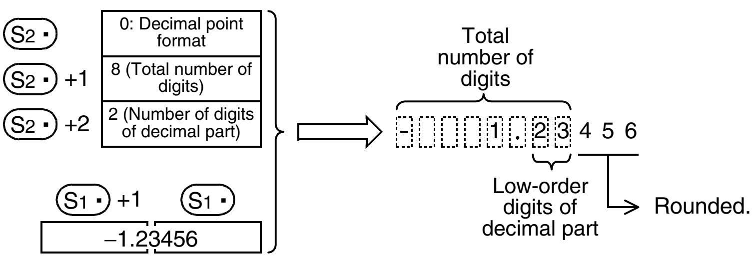


- The total number of digits which can be specified by S2· +1 is as follows (24 digits maximum):
  When the number of digits of the decimal part is "0", Total number of digits ≥ 2
  When the number of digits of the decimal part is any value other than "0", Total number of digits ≥ (Number of digits of decimal part + 3)
- The number of digits of the decimal part which can be specified by S2· +2 is from 0 to 7.
  However, the following must be satisfied, "Number of digits of decimal part ≤ (Total number of digits - 3)"

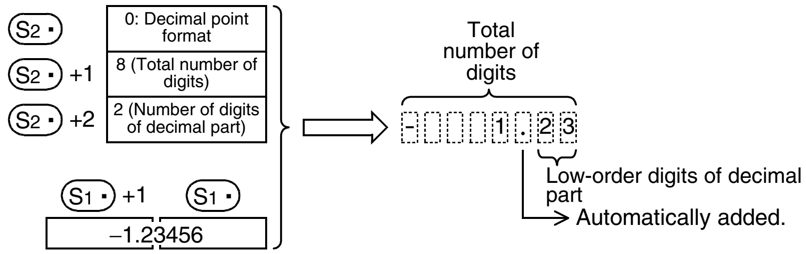
For example, when the total number of digits is "8", the number of digits of the decimal part is "3", and "-1.23456" is specified, data is stored in D· and later as shown below:



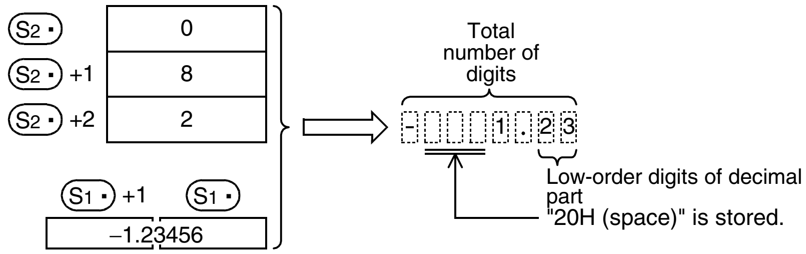
- The character string data after conversion is stored in the devices D· and later as shown below:
  - For the sign, "20H (space)" is stored when the binary floating point data is positive, and "2DH (-)" is stored when the data is negative.
  - If the decimal part of the binary floating point data cannot be accommodated in the number of digits of the decimal part, low-order digits of the decimal part are rounded.



- When the number of digits of the decimal part is set to any value other than "0", "2EH (.)" is automatically stored in "specified number of digits of decimal part + 1"th digit.
  When the number of digits of the decimal part is "0", "2EH (.)" is not stored.

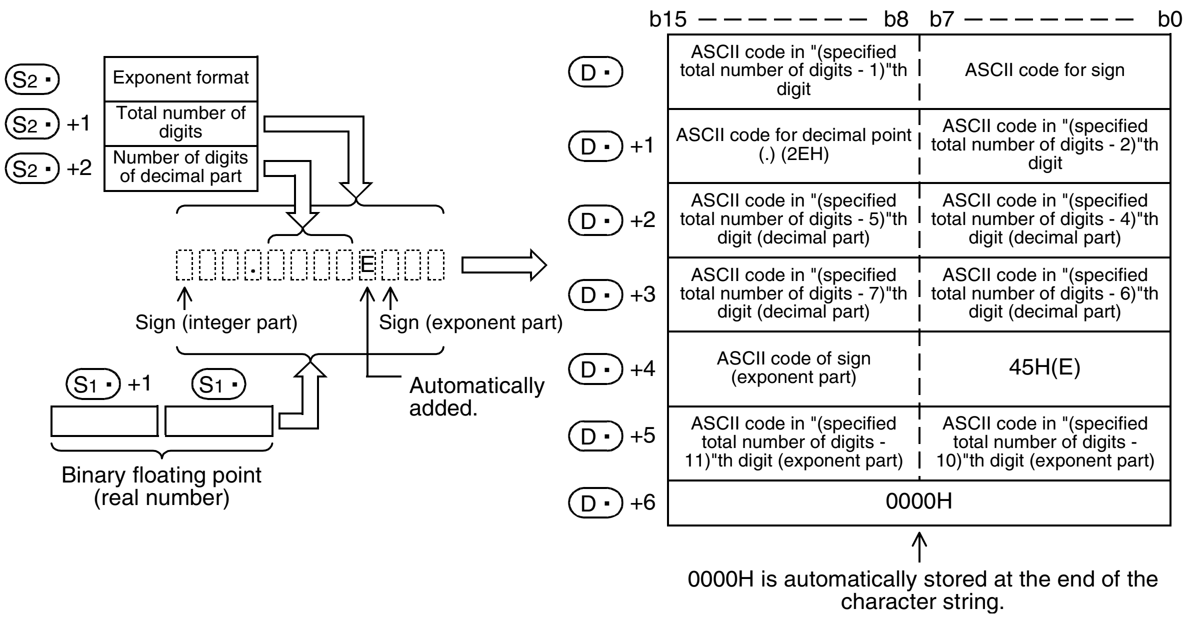


- When the total number of digits subtracted by the digits for sign, decimal point and decimal part is larger than the integer part of the binary floating point data, "20H (space)" is stored between the sign and the integer part.



- "00H" is automatically stored at the end of the converted character string.

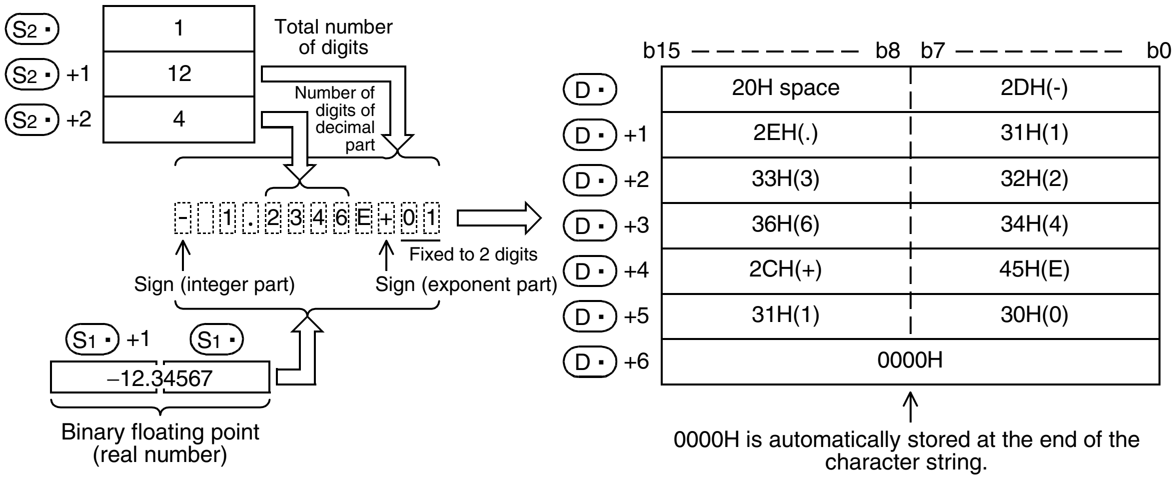
3. In the case of exponent format



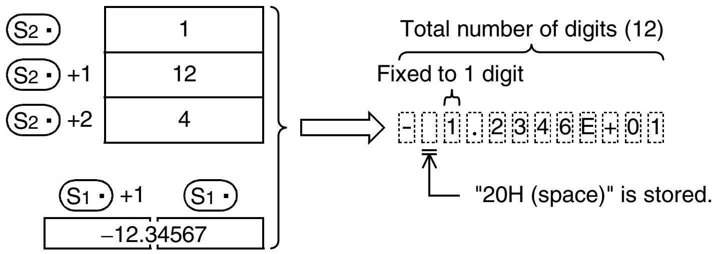
- The total number of digits which can be specified by S2· +1 is as follows (24 digits maximum):
  When the number of digits of the decimal part is "0" Total number of digits ≥ 6
  When the number of digits of the decimal part is any value other than "0" Total number of digits ≥ (Number of digits of decimal part + 7)

- The number of digits of the decimal part which can be specified by (S2•) +2 is from 0 to 7.
  However, the following must be satisfied, "Number of digits of decimal part ≤ (Total number of digits - 7)"

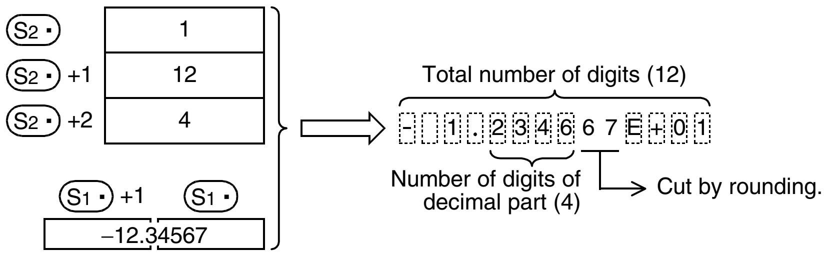
For example, when the total number of digits is "12", the number of digits of the decimal part is "4", and "-12.34567" is specified, data is stored in (D•) and later as shown below:



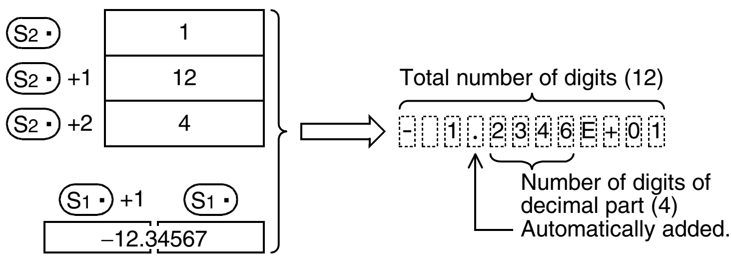
- The character string data after conversion is stored in the devices (D•) and later as shown below:
  - For the sign of the integer part, "20H (space)" is stored when the binary floating point data is positive, and "2DH (-)" is stored when the data is negative.
  - The integer part is fixed to 1 digit.
    "20H (space)" is stored between the integer part and the sign.



  - If the decimal part of the binary floating point data cannot be accommodated in the number of digits of the decimal part, low-order digits of the decimal part are rounded.

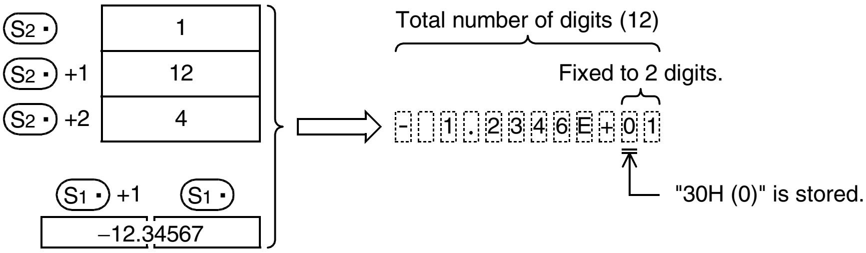


  - When the number of digits of the decimal part is set to any value other than "0", "2EH (.)" is automatically stored in "specified number of digits of decimal part + 1"th digit.
    When the number of digits of the decimal part is "0", "2EH (.)" is not stored.



  - For the sign of the exponent part, "2BH (+)" is stored when the exponent is positive, and "2DH (-)" is stored when the exponent is negative.

- The exponent part is fixed to 2 digits.
  When the exponent part is 1 digit, "30H (0)" is stored after the sign of the exponent part.



- "00H" is automatically stored at the end of the converted character string.

Related instructions

Instruction	Description
EVAL (FNC117)	Converts a character string (ASCII codes) into binary floating point data.
STR (FNC200)	Converts binary data into a character string (ASCII codes).
VAL (FNC201)	Converts a character string (ASCII codes) into binary data.

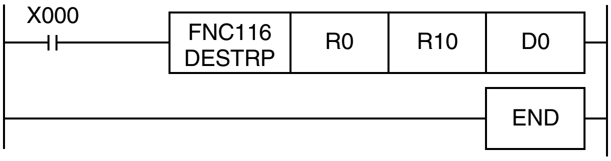
Errors

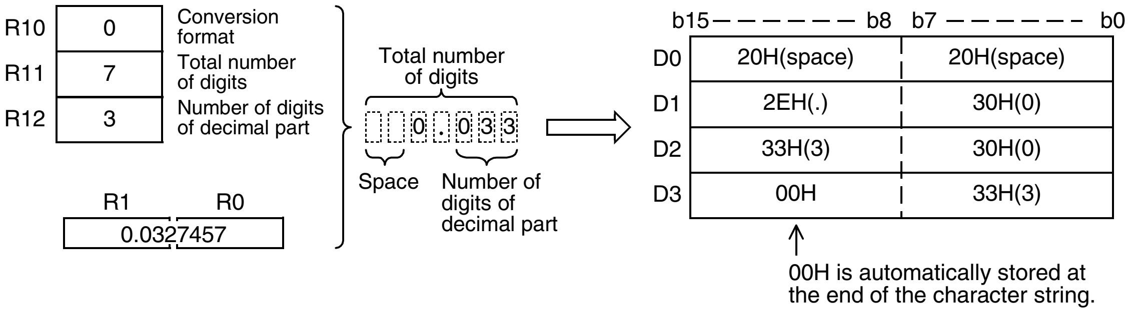
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When (S1•) is not located within the following range (error code: K6706)
  $0, \pm 2^{-126} \leq S1 < \pm 2^{128}$
- When the format specified by (S2•) is any value other than "0" or "1" (error code: K6706)
- When the total number of digits specified by (S2•) +1 is not located within the following range (error code: K6706)
  In the case of decimal point format:
  When the number of digits of the decimal part is "0", Total number of digits ≥ 2
  When the number of digits of the decimal part is any value other than "0", Total number of digits ≥ (Number of digits of decimal part + 3)
  In the case of exponent format:
  When the number of digits of the decimal part is "0", Total number of digits ≥ 6
  When the number of digits of the decimal part is any value other than "0", Total number of digits ≥ (Number of digits of decimal part + 7)
- When the number of digits of the decimal part specified by (S2•) +2 is not located within the following range (error code: K6706)
  In the case of decimal point format: Number of digits of decimal part ≤ (Total number of digits - 3)
  In the case of exponent format: Number of digits of decimal part ≤ (Total number of digits - 7)
- When the devices storing a character string specified by (D•) exceeds the allowable device range (error code: K6706)
- When the conversion result exceeds the specified total number of digits (error code: K6706)

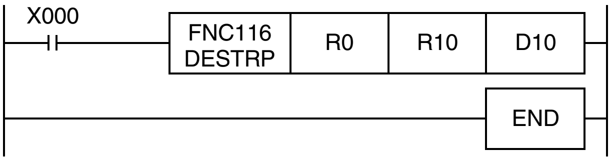
Program examples

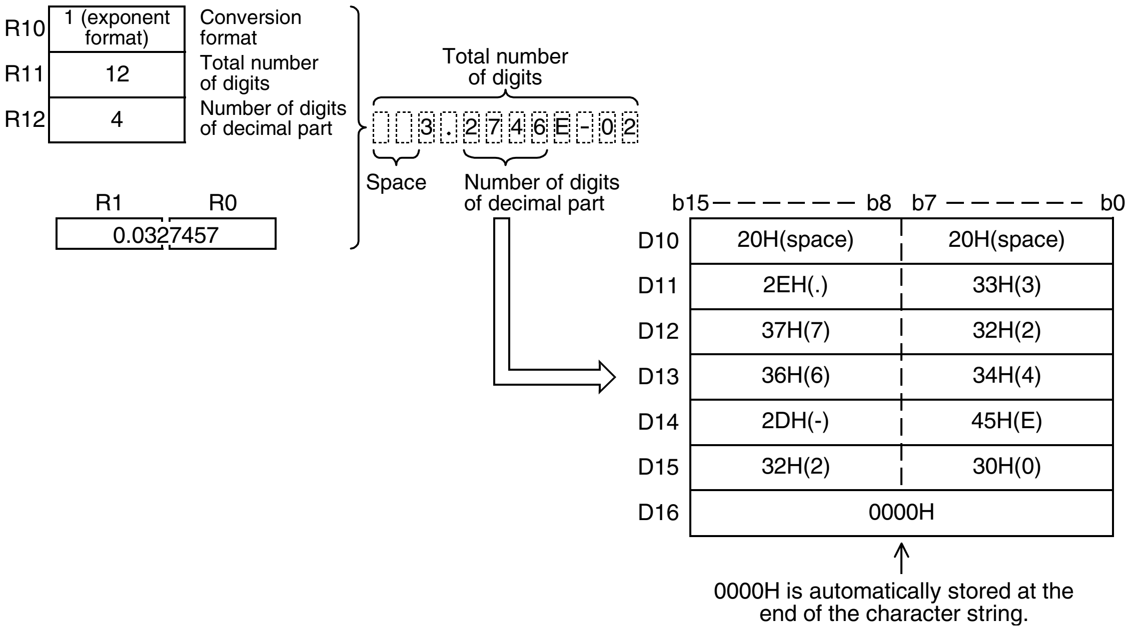
1) In the program example shown below, the contents (binary floating point data) of R0 and R1 are converted according to the contents specified by R10 to R12, and then stored to D0 and later when X000 turns ON





2) In the program shown below, the contents (binary floating point data) of R0 and R1 are converted according to the contents specified by R10 to R12, and then stored to D10 and later when X000 turns ON





18.5 FNC117 – EVAL / Character String to Floating Point Conversion

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction converts a character string (ASCII codes) into binary floating point data.
On the other hand, the VAL (FNC201) instruction converts a character string (ASCII codes) into binary data.

→ **For a character string, refer to Section 5.3.**
→ **For handling of floating point, refer to Subsection 5.1.3.**
→ **For VAL (FNC201) instruction, refer to Subsection 2.6.2.**

1. Instruction format

D | FNC 117 EVAL | P

16-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DEVAL	Continuous Operation
	DEVALP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Head device number storing character string data to be converted into binary floating point data	Character string
D·	Head device number storing converted binary floating point data	Real number (binary)

3. Applicable devices

Operand Type	Bit Devices: System User X	Y	M	T	C	S	D□.b	Word Devices: Digit Specification KnX	KnY	KnM	KnS	System User T	C	D	R	Special Unit U□\G□	Index V	Z	Modify	Others: Constant K	H	Real Number E	Character String "□"	Pointer P
S·								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
D·														✓	✓	✓			✓					

Explanation of function and operation

1. 32-bit operation (EVAL and EVALP)

A character string stored in S· and later is converted into binary floating point, and stored to [D·+1, D·].

Command input —| |— FNC117 DEVAL | S· | D·

A specified character string may be in the decimal point format or exponent format. A character string in either format can be converted into binary floating point data.

	b15 – b8	b7 – b0
S·	ASCII code for 1st character	ASCII code for sign
S·+1	ASCII code for 3rd character	ASCII code for 2nd character
S·+2	ASCII code for 5th character	ASCII code for 4th character
S·+3	ASCII code for 7th character	ASCII code for 6th character
S·+4		00H

↑ Indicates the end of the character string.

⇒ D·+1 D·
Binary floating point (real number)

11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FNC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

a) In the case of decimal point format

	b15 — b8	b7 — b0
S·	31H(1)	2DH(-)
S· +1	30H(0)	2EH(.)
S· +2	38H(8)	37H(7)
S· +3	32H(2)	31H(1)
S· +4		00H

- 1 . 0 7 8 1 2

⇒ D· +1 D·: −1.07812

Binary floating point (real number)

b) In the case of exponent format

	b15 — b8	b7 — b0
S·	20H(space)	2DH(-)
S· +1	2EH(.)	31H(1)
S· +2	32H(2)	33H(3)
S· +3	31H(1)	30H(0)
S· +4	2BH(+)	45H(E)
S· +5	30H(0)	31H(1)
S· +6		00H

-   1 . 3 2 0 1 E + 1 0

⇒ D· +1 D·: −1.3201E+10

Binary floating point (real number)

- When a character string to be converted into binary floating point specified by S· has 7 digits or more excluding the sign, decimal point and exponent part, the conversion result may contain rounding error.

  a) In the case of decimal point format

	b15 — b8	b7 — b0
S·	20H(space)	2DH(-)
S· +1	31H(1)	20H(space)
S· +2	33H(3)	2EH(.)
S· +3	31H(1)	30H(0)
S· +4	36H(6)	35H(5)
S· +5	31H(1)	38H(8)
S· +6	00H	32H(2)

-     1 . 3 0 1 5 6 8 1 2

Rounded.

⇒ D· +1 D·: −1.30156

Up to 6 digits are securely equivalent.

Binary floating point (real number)

b) In the case of exponent format

	b15 ------ b8	b7 ------ b0
S·	20H(.)	2DH(-)
S· +1	2EH(.)	31H(1)
S· +2	35H(5)	33H(3)
S· +3	33H(3)	30H(0)
S· +4	31H(1)	34H(4)
S· +5	45H(E)	32H(2)
S· +6	30H(0)	2DH(-)
	00H	32H(2)

- 1 . 3 5 0 3 4 1 2 E - 0 2

Rounded.

⇒ D· +1 D·: –1.35034E–2

Up to 6 digits are securely equivalent.

Binary floating point (real number)

- When "20H (space)" or "30H (0)" exists between numbers except the first "0" in a character string specified by S·, "20H" or "30H" is ignored during conversion.

	b15 ------ b8	b7 ------ b0
S·	20H(space)	2DH(-)
S· +1	31H(1)	30H(0)
S· +2	32H(2)	2EH(.)
S· +3	31H(1)	33H(3)
S· +4		00H

- 0 1 . 2 3 1

Ignored.

⇒ D· +1 D·: –1.231

Binary floating point (real number)

- When "30H (0)" exists between a number and "E" in a character string in the exponent format, "30H" is ignored during conversion.

b15 ------ b8	b7 ------ b0
20H(space)	2DH(-)
2EH(.)	31H(1)
34H(4)	30H(0)
33H(3)	35H(5)
2BH(+)	45H(E)
33H(3)	30H(0)
	00H

- 1 . 0 4 5 3 E + 0 3

Ignored.

⇒ D· +1 D·: –1.0453E+3

- A character string can consist of up to 24 characters.

Related devices

→ **For the use methods of the zero, borrow and carry flags, refer to Subsection 6.5.2.**

Device	Name	Description	
		Condition	Operation
M8020	Zero flag	The conversion result is true "0". (The mantissa part is "0".)	The zero flag M8020 turns ON.
M8021	Borrow flag	The absolute value of the conversion result is less than "2^{-126}".	The value of D· is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag M8021 turns ON.
M8022	Carry flag	The absolute value of the conversion result is not less than "2^{128}".	The value of D· is the maximum value (2^{128}) of 32-bit real numbers and the carry flag M8022 turns ON.

Related instructions

Instruction	Description
ESTR (FNC116)	Converts binary floating point data into a character string (ASCII codes).
STR (FNC200)	Converts binary data into a character string (ASCII codes).
VAL (FNC201)	Converts a character string (ASCII codes) into binary data.

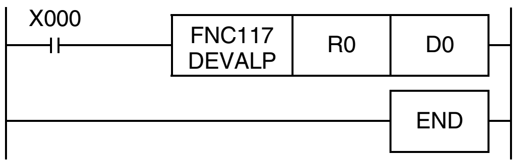
Errors

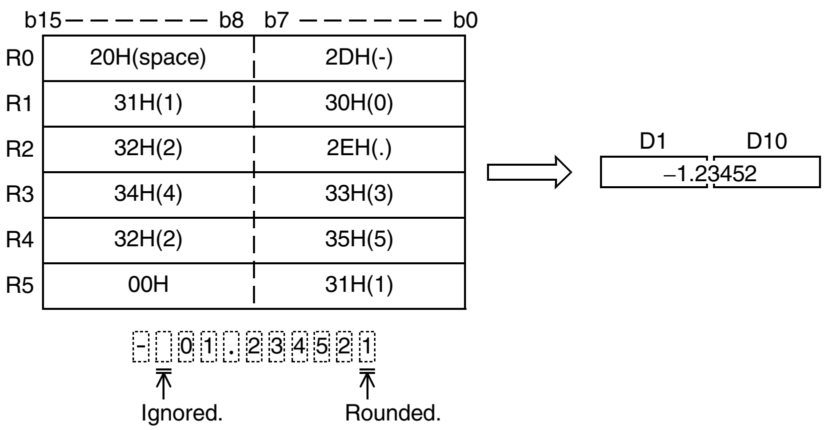
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When any character other than "30H (0)" to "39H (9)" exists in the integer part or decimal part (error code: K6706)
- When "2EH (.)" exists in two or more positions in a character string specified by D· (error code: K6706)
- When any character other than "45H (E)", "2CH (+)", "45H (E)" or "2DH (-)" exists in the exponent part, or when two or more exponent parts exist (error code: K6706)
- When "00H" does not exist in the corresponding device range starting from S· (error code: K6706)
- When the number of characters after S· is "0" or more than "24" (error code: K6706)

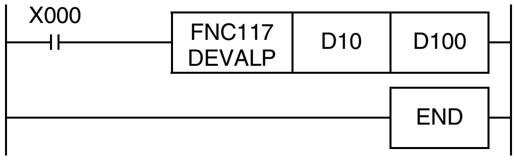
Program examples

1) In the program example shown below, a character string stored in R0 and later is converted into binary floating point, and stored to D0 and D1 when X000 turns ON

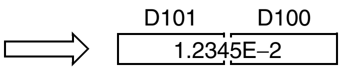


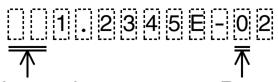


2) In the program shown below, a character string stored in D10 and later is converted into binary floating point, and stored to D100 and D101 when X000 turns ON



	b15 ——— b8	b7 ——— b0
D10	20H(space)	20H(space)
D11	2EH(.)	31H(1)
D12	33H(3)	32H(2)
D13	35H(5)	34H(4)
D14	2DH(-)	45H(E)
D15	32H(2)	30H(0)
D16		00H





Operations at overflow, underflow and zero

Condition	Operation
The absolute value of the conversion result is less than "2^{-126}".	The value of (D·) is the minimum value (2^{-126}) of 32-bit real numbers and the borrow flag M8021 turns ON.
The absolute value of the conversion result is not less than "2^{128}".	The value of (D·) is the maximum value (2^{128}) of 32-bit real numbers and the carry flag M8022 turns ON.
The conversion result is true "0". (The mantissa part is "0".)	The zero flag M8020 turns ON.

18.6 FNC118 – EBCD / Floating Point to Scientific Notation Conversion

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction converts binary floating point into scientific notation.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D FNC 118 EBCD P

16-bit Instruction	Mnemonic	Operation Condition
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DEBCD	Continuous Operation
	DEBCDP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(S·)	Data register number storing binary floating point	Real number (binary)
(D·)	Data register number storing converted scientific notation	Real number (decimal)

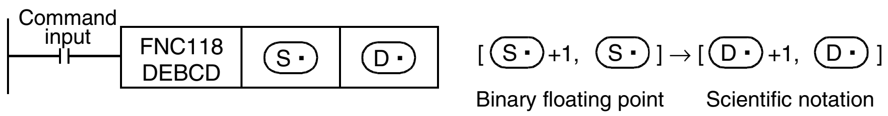
3. Applicable devices

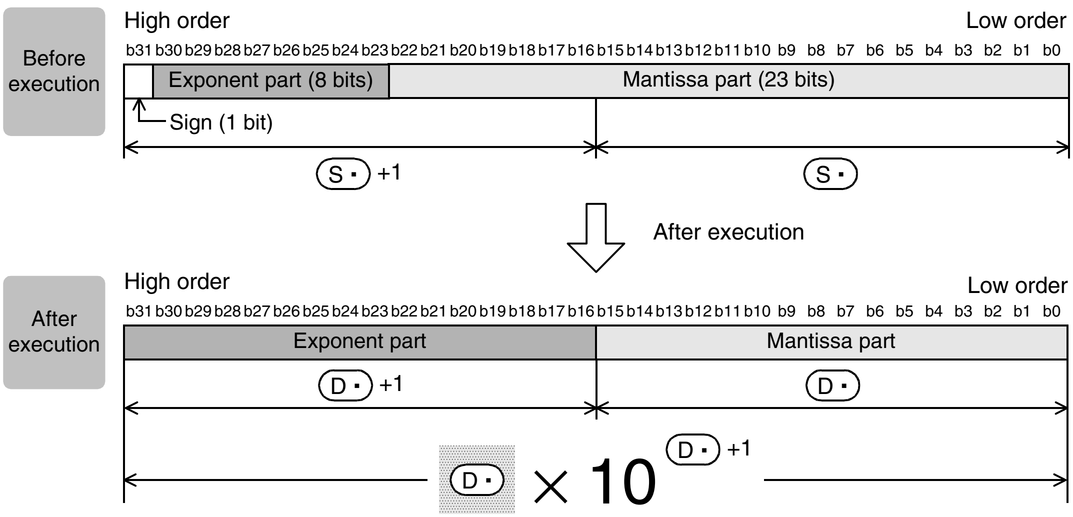
Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)														✓	✓	✓			✓					
(D·)														✓	✓	✓			✓					

Explanation of function and operation

1. 32-bit operation (DEBCD and DEBCDP)

Binary floating point stored in [(S·)+1, (S·)] is converted into scientific notation, and transferred to [(D·)+1, (D·)].





Caution

1. Handling of floating point

In floating point operations, all data is handled in binary floating point.
Because binary floating point is difficult to understand (requiring a dedicated monitoring method), it is converted into scientific notation so that monitoring can be easily executed by peripheral equipment.
GX Developer and GOT have the function to directly monitor and display binary floating point.

18.7 FNC119 – EBIN / Scientific Notation to Floating Point Conversion

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction converts scientific notation stored in devices into binary floating point.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D	FNC 119 EBIN	P

16-bit Instruction	Mnemonic	Operation Condition
–		

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DEBIN	Continuous Operation
	DEBINP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Data register number storing scientific notation data	Real number (decimal)
D·	Data register number storing converted binary floating point.	Real number (binary)

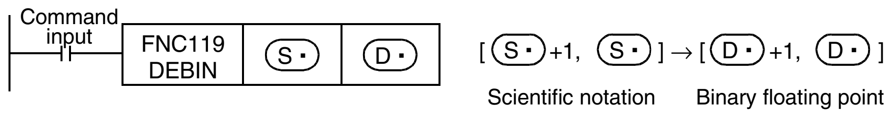
3. Applicable devices

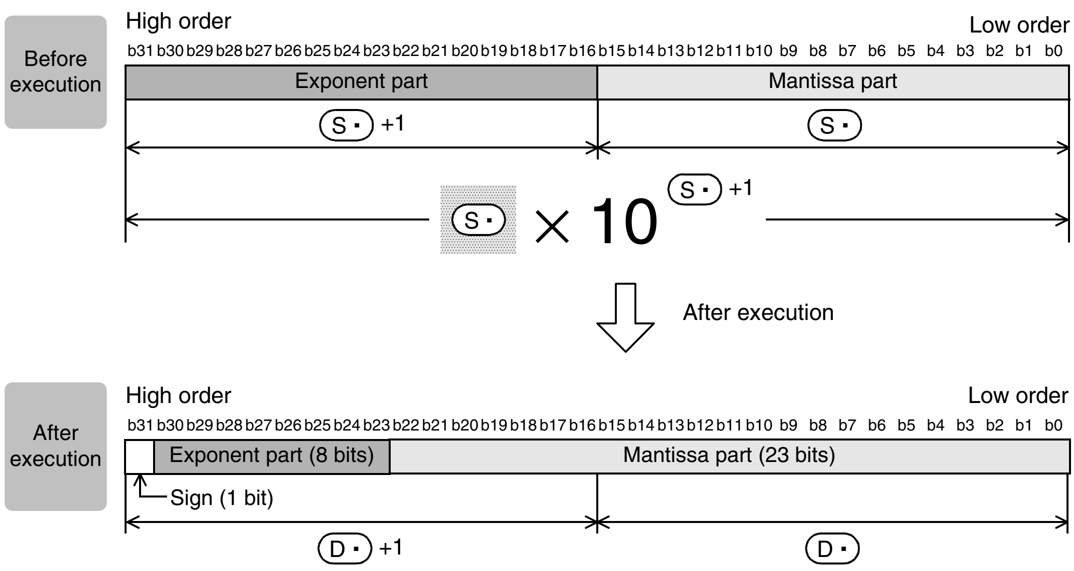
Oper-and Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·														✓	✓	✓			✓					
D·														✓	✓	✓			✓					

Explanation of function and operation

1. 32-bit operation (DEBIN and DEBINP)

Scientific notation stored in [S·+1, S·] is converted into binary floating point, and transferred to [D·+1, D·].



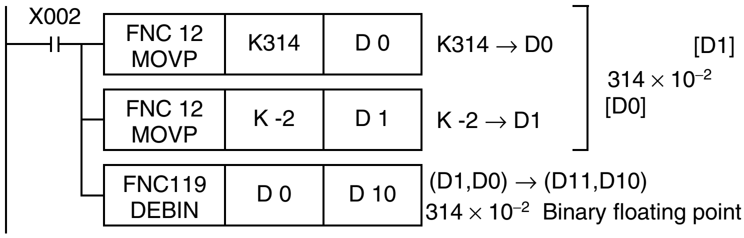


Program example

By DEBIN instruction, a numeric value containing the decimal point can be directly converted into binary floating point.

Example: Converting "3.14" into binary floating point

$3.14 = 314 \times 10^{-2}$ (scientific notation)



→ **For program examples of floating point operations, refer to Section 12.10.**

18.8 FNC120 – EADD / Floating Point Addition

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes addition of two binary floating point data.

→ **For program examples of floating point operations, refer to Section 12.10.**
→ **For handling of floating point, refer to Subsection 5.1.3.**
→ **For flag operations, refer to Subsection 6.5.2.**

1. Instruction format

D FNC 120 EADD P

16-bit Instruction	Mnemonic	Operation Condition
	–	

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DEADD	Continuous Operation
	DEADDP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S1•	Word device number storing binary floating point data used in addition	Real number (binary)*1
S2•	Word device number storing binary floating point data used in addition	
D•	Data register number storing the addition result	

*1. When a constant (K or H) is specified, it is automatically converted into binary floating point (real number) when the instruction is executed.

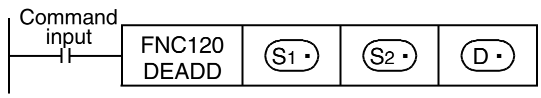
3. Applicable devices

Operand Type	Bit Devices: System User							Word Devices: Digit Specification				Word Devices: System User				Special Unit	Index			Others: Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•														✓	✓	✓			✓	✓	✓	✓		
S2•														✓	✓	✓			✓	✓	✓	✓		
D•														✓	✓	✓			✓					

Explanation of function and operation

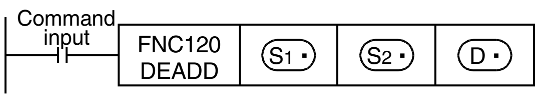
1. 32-bit operation (DEADD and DEADDP)

Binary floating point data [S2•+1, S2•] is added to binary floating point data [S1•+1, S1•], and the addition result in the binary floating point format is transferred to [D•+1, D•].



[S1•+1, S1•] + [S2•+1, S2•] → [D•+1, D•]
Binary floating point / Binary floating point / Binary floating point

When a constant (K or H) is specified as [S1•+1, S1•] or [S2•+1, S2•], it is automatically converted into binary floating point.



[S1•+1, S1•] + [K2346] → [D•+1, D•]
Binary floating point / Automatically converted into binary floating point / Binary floating point

Caution

1. When a same device is specified

The same device number can be specified in [S1•+1, S1•],[S2•+1, S2•] and [D•+1, D•].
In this case, note that the addition result changes in every operation cycle when the continuous operation type instruction (DEADD) is used.

18.9 FNC121 – ESUB / Floating Point Subtraction

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes subtraction of two binary floating point data.

→ **For program examples of floating point operations, refer to Section 12.10.**
→ **For handling of floating point, refer to Subsection 5.1.3.**
→ **For flag operations, refer to Subsection 6.5.2.**

1. Instruction format

D FNC 121 ESUB P

16-bit Instruction	Mnemonic	Operation Condition
–		

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DESUB	Continuous Operation
	DESUBP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(S1•)	Word device number storing binary floating point data used in subtraction	Real number (binary)*1
(S2•)	Word device number storing binary floating point data used in subtraction	
(D•)	Data register number storing the subtraction result	

*1. When a constant (K or H) is specified, it is automatically converted into binary floating point (real number) when the instruction is executed.

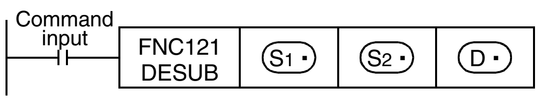
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S1•)														✓	✓	✓			✓	✓	✓	✓		
(S2•)														✓	✓	✓			✓	✓	✓	✓		
(D•)														✓	✓	✓			✓					

Explanation of function and operation

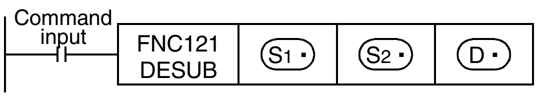
1. 32-bit operation (DESUB and DESUBP)

Binary floating point data [(S2•)+1, (S2•)] is subtracted from binary floating point data [(S1•)+1, (S1•)], and the subtraction result in the binary floating point format is transferred to [(D•)+1, (D•)].



[(S1•)+1, (S1•)] – [(S2•)+1, (S2•)] → [(D•)+1, (D•)]
Binary floating point | Binary floating point | Binary floating point

When a constant (K or H) is specified as [(S1•)+1, (S1•)] or [(S2•)+1, (S2•)], it is automatically converted into binary floating point.



[(S1•)+1, (S1•)] – [K2346] → [(D•)+1, (D•)]
Binary floating point | Automatically converted into binary floating point | Binary floating point

Caution

1. When a same device is specified

A same device number can be specified in [(S1•)+1, (S1•)],[(S2•)+1, (S2•)] and [(D•)+1, (D•)].
In this case, note that the subtraction result changes in every operation cycle when the continuous operation type instruction (DESUB) is used.

18.10 FNC122 – EMUL / Floating Point Multiplication

FX3UC Ver.1.00 ⇒ FX3U Ver.2.20 ⇒

Outline

This instruction executes multiplication of two binary floating point data.

→ For program examples of floating point operations, refer to Section 12.10.
→ For handling of floating point, refer to Subsection 5.1.3.

1. Instruction format

D	FNC 122 EMUL	P

16-bit Instruction	Mnemonic	Operation Condition
	–	

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DEMUL	Continuous Operation
	DEMULP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S1•	Word device number storing binary floating point data used in multiplication	Real number (binary)*1
S2•	Word device number storing binary floating point data used in multiplication	
D•	Data register number storing the multiplication result	

*1. When a constant (K or H) is specified, it is automatically converted into binary floating point (real number) when the instruction is executed.

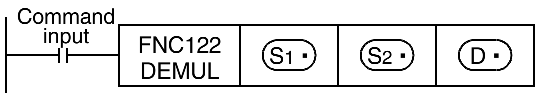
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•														✓	✓	✓			✓	✓	✓	✓		
S2•														✓	✓	✓			✓	✓	✓	✓		
D•														✓	✓	✓			✓					

Explanation of function and operation

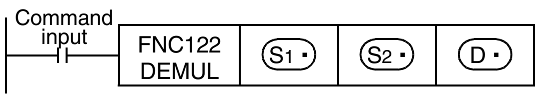
1. 32-bit operation (DEMUL and DEMULP)

Binary floating point data [S1• +1, S1•] is multiplied by binary floating point data [S2• +1, S2•], and the multiplication result in the binary floating point format is transferred to [D• +1, D•].



[S1• +1, S1•] × [S2• +1, S2•] → [D• +1, D•]
Binary floating point　Binary floating point　Binary floating point

When a constant (K or H) is specified as [S1• +1, S1•] or [S2• +1, S2•], it is automatically converted into binary floating point.



[S1• +1, S1•] × [K100] → [D• +1, D•]
Binary floating point　Automatically converted into binary floating point　Binary floating point

18.11 FNC123 – EDIV / Floating Point Division

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction executes division of two binary floating point.

→ For program examples of floating point operations, refer to Section 12.10.
→ For handling of floating point, refer to Subsection 5.1.3.
→ For flag operations, refer to Subsection 6.5.2.

1. Instruction format

	FNC 123	
D	EDIV	P

16-bit Instruction	Mnemonic	Operation Condition
	–	

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DEDIV	Continuous Operation
	DEDIVP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(S1·)	Word device number storing binary floating point data used in division	Real number (binary)*1
(S2·)	Word device number storing binary floating point data used in division	
(D·)	Data register number storing the division result	

*1. When a constant (K or H) is specified, it is automatically converted into binary floating point (real number) when the instruction is executed.

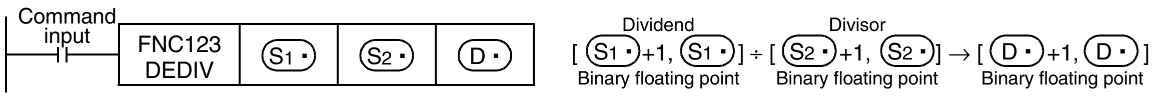
3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S1·)														✓	✓	✓			✓	✓	✓	✓		
(S2·)														✓	✓	✓			✓	✓	✓	✓		
(D·)														✓	✓	✓			✓					

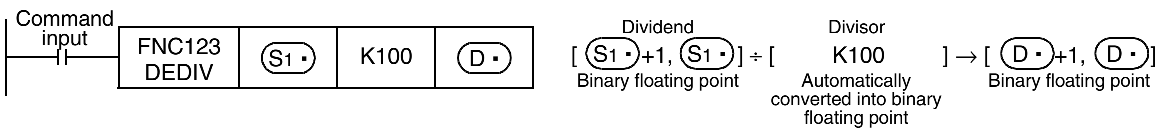
Explanation of function and operation

1. 32-bit operation (DEDIV and DESDIVP)

Binary floating point data [(S1·)+1, (S1·)] is divided by binary floating point data [(S2·)+1, (S2·)], and the division result in the binary floating point format is transferred to [(D·)+1, (D·)].



When a constant (K or H) is specified as [(S1·)+1, (S1·)] or [(S2·)+1, (S2·)], it is automatically converted into binary floating point.



18.12 FNC124 – EXP / Floating Point Exponent

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes exponential operation whose base is "e (2.71828)".

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D	FNC 124 EXP	P

16-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DEXP	Continuous Operation
	DEXPP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Head device number storing binary floating point data used in exponential operation.	Real number (binary)
D·	Head device number storing the operation result.	

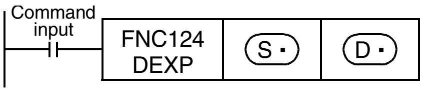
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·														✓	✓	✓			✓			✓		
D·														✓	✓	✓			✓					

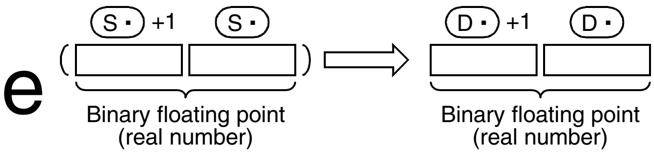
Explanation of function and operation

1. 32-bit operation (DEXP and DEXPP)

The exponent of [S·+1, S·] is calculated, and the operation result is stored to [D·+1, D·].
A real number can be directly specified as S·.



- In the exponential operation, the base (e) is set to "2.71828".



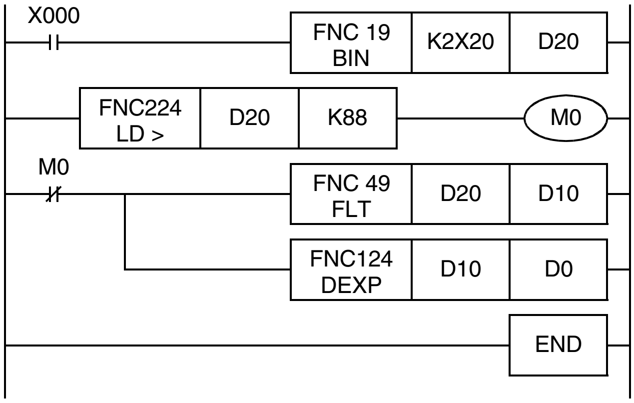
Error

An operation error occurs in the following case; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the operation result is outside the following range (error code: K6706)
$2^{-126} \leq |\text{ Operation result }| < 2^{128}$

Program example

In the program example shown below, the exponential operation is executed for a value set in the 2-digit BCD format in X020 to X027, and the operation result is stored in the binary floating point format to D0 and D1 when X000 turns ON.



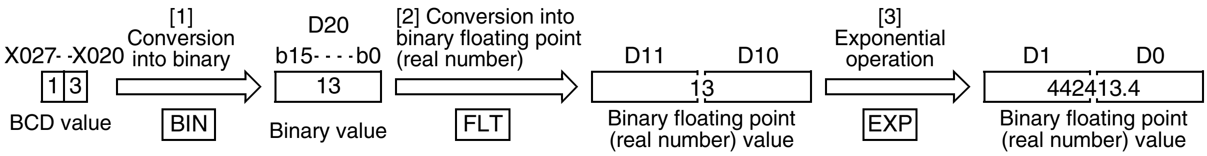
Data used in the exponential operation is input ([1]).

The range of the value to be operated is checked. (Refer to 1) in "Points" below.)

The input data is converted into binary floating point (real number) ([2]).

The exponential operation is executed ([3]).

Operation when "13" is specified in X020 to X027



Points

1) The operation result becomes less than "2^{128}" when the BCD value set in X020 to X027 is "88" or less because of "$\log e 2^{128}$ = 88.7".
If a value "89" or more is set, an operation error occurs. To prevent this operation error, when a value more than "89" is set, M0 is set to ON so that the exponential operation is not executed.

2) Conversion from natural logarithm into common logarithm
In the CPU, operations are executed in natural logarithm.
For obtaining a value in common logarithm, specify a common logarithm value divided by "0.4342945" in [(S·)+1, (S·)].

$$10^X = e^{\frac{X}{0.4342945}}$$

18.13 FNC125 – LOGE / Floating Point Natural Logarithm

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes the natural logarithm operation.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D FNC 125 LOGE P

16-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DLOGE	Continuous Operation
	DLOGEP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S•	Head device number storing binary floating point data used in the natural logarithm operation	Real number (binary)
D•	Head device number storing the operation result	

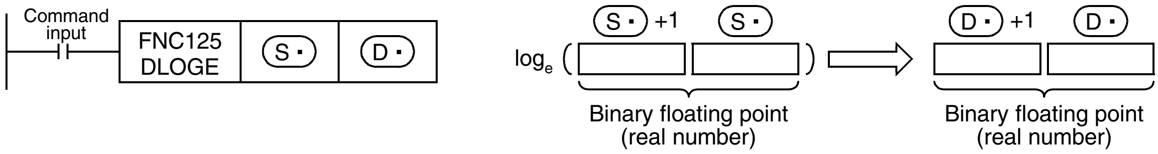
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•														✓	✓	✓			✓			✓		
D•														✓	✓	✓			✓					

Explanation of function and operation

1. 32-bit operation (DLOGE and DLOGEP)

Natural logarithm [logarithm whose base is "e (2.71828)"] of [S• +1, S•] is calculated, and the operation result is stored to [D• +1, D•]. A real number can be directly specified as S•.



- Only a positive value can be set in [S• +1, S•]. (The natural logarithm operation cannot be executed for a negative value.)

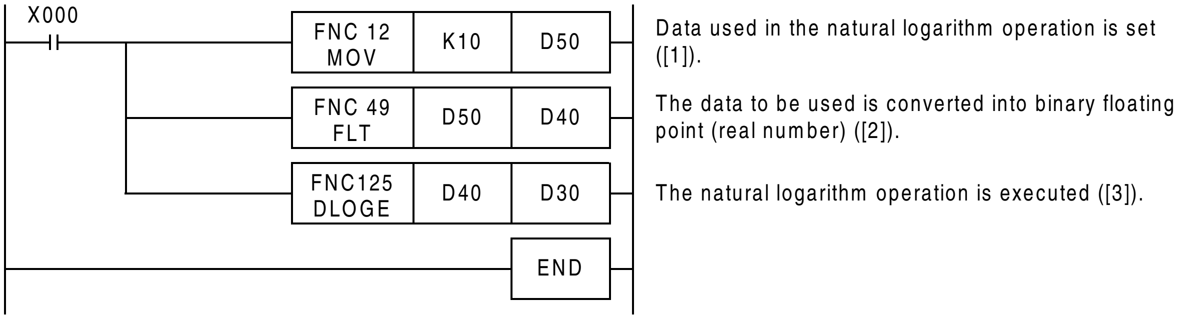
Errors

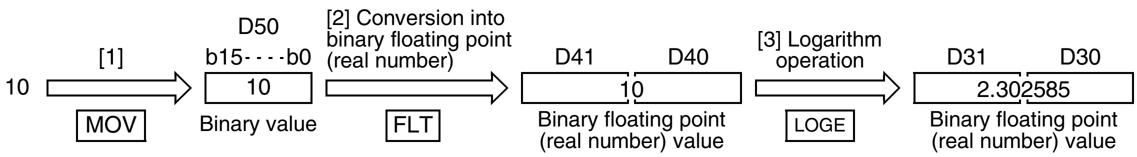
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When a negative value is specified in S• (error code: K6706)
- When "0" is specified in S• (error code: K6706)

Program example

In the program example shown below, natural logarithm of "10" set in D50 is calculated, and stored to D30 and D31 when X000 turns ON.





18.14 FNC126 – LOG10 / Floating Point Common Logarithm

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction executes the common logarithm operation.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D FNC 126 LOG10 P

16-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DLOG10	Continuous Operation
	DLOG10P	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(S·)	Head device number storing binary floating point data used in the common logarithm operation	Real number (binary)
(D·)	Head device number storing the operation result	

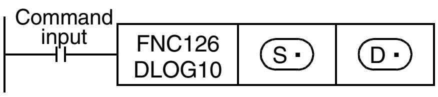
3. Applicable devices

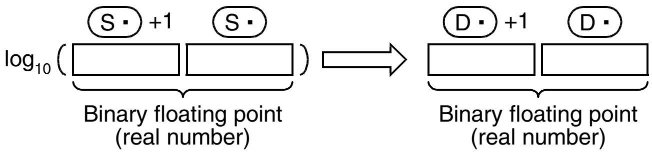
Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)														✓	✓	✓			✓			✓		
(D·)														✓	✓	✓			✓					

Explanation of function and operation

1. 32-bit operation (DLOG10 and DLOG10P)

Common logarithm [logarithm whose base is "10"] of [(S·)+1, (S·)] is calculated, and the operation result is stored to [(D·)+1, (D·)]. A real number can be directly specified as (S·).





- Only a positive value can be set in [(S·)+1, (S·)]. (The common logarithm operation cannot be executed for a negative value.)

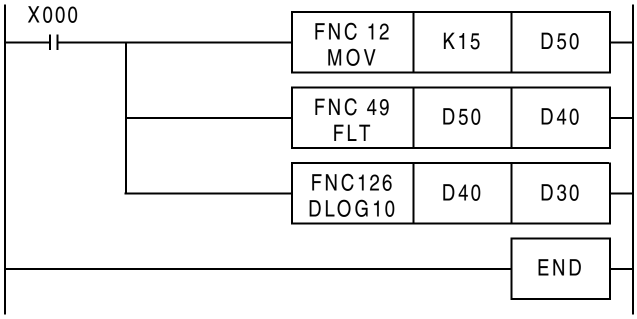
Errors

An operation error occurs in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When a negative value is specified in (S·) (error code: K6706)
- When "0" is specified in (S·) (error code: K6706)

Program example

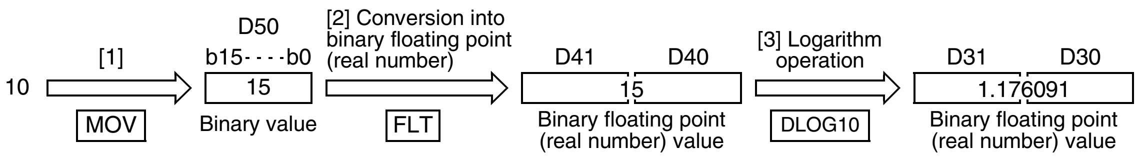
In the program example shown below, common logarithm of "15" set in D50 is calculated, and stored to D30 and D31 when X000 turns ON.



Data used in the common logarithm operation is set ([1]).

The data to be used is converted into binary floating point (real number) ([2]).

The common logarithm operation is executed ([3]).



18.15 FNC127 – ESQR / Floating Point Square Root

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction obtains the square root of binary floating point.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D	FNC 127 ESQR	P

16-bit Instruction	Mnemonic	Operation Condition
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DESQR	Continuous Operation
	DESQRP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(S·)	Word device number storing binary floating point data whose square root is calculated	Real number (binary)*1
(D·)	Data register number storing the square root of binary floating point data	

*1. When a constant (K or H) is specified, it is automatically converted into binary floating point (real number) when the instruction is executed.

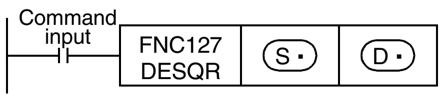
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)														✓	✓	✓			✓	✓	✓	✓		
(D·)														✓	✓	✓			✓					

Explanation of function and operation

1. 32-bit operation (DESQR and DESQRP)

The square root of [(S1·)+1, (S1·)] is calculated (in the binary floating point operation), and the result is transferred to [(D·)+1, (D·)].



$\sqrt{[(S\cdot)+1, (S\cdot)]}$ → [(D·)+1, (D·)]
Binary floating point → Binary floating point

Related device

→ **For the zero flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8020	Zero flag	Turns ON when the operation result is true "0".

Error

The contents of [(S1·)+1, (S1·)] are valid only when a positive value is set. When a negative value is set, the operation error flag M8067 turns ON, and the instruction is not executed.

18.16 FNC128 – ENEG / Floating Point Negation

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction inverts the sign of binary floating point (real number) data.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

FNC 128 ENEG	D	P

16-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

32-bit Instruction	Mnemonic	Operation Condition
5 steps	DENEG	Continuous Operation
	DENEGP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(D·)	Head device number storing binary floating data whose sign is to be inverted	Real number (binary)

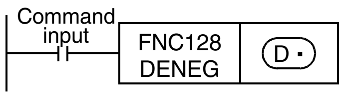
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(D·)														✓	✓	✓			✓					

Explanation of function and operation

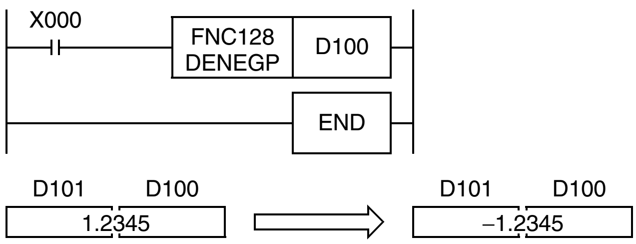
1. 32-bit operation (DENEG and DENEGP)

The sign of binary floating point stored in [(D·)+1, (D·)] is inverted, and the negation result is stored to [(D·)+1, (D·)].



Program example

In the program example shown below, the sign of floating point data stored in D100 and D101 is inverted, and the negation result is stored to D100 and D101 when X000 turns ON.



18.17 FNC129 – INT / Floating Point to Integer Conversion

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction converts binary floating point data into a binary integer which is a normal data format inside PLCs (binary floating point → binary integer).

→ **For program examples of floating point operations, refer to Section 12.10.**
→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D	FNC 129 INT	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	INT	Continuous Operation
	INTP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DINT	Continuous Operation
	DINTP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Data register number storing binary floating point data to be converted into a binary integer	Real number (binary)
D·	Data register number storing a converted binary integer	16- or 32-bit binary

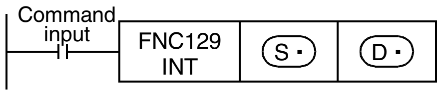
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·														✓	✓	✓			✓					
D·														✓	✓	✓			✓					

Explanation of function and operation

1. 16-bit operation (INT and INTP)

Binary floating point stored in [S·+1, S·] is converted into a binary integer, and transferred to D·.



S·+1, S· → D·
Binary floating point → 16-bit binary integer
The decimal part is cut.

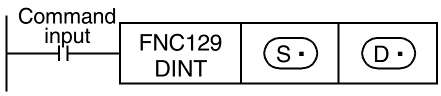
Instruction for inverse conversion

The inverse conversion is executed by FLT (FNC 49) instruction.

→ **For FLT (FNC 49) instruction, refer to Section 12.10.**

2. 32-bit operation (DINT and DINTP)

Binary floating point stored in [S·+1, S·] is converted into a binary integer, and transferred to [D·+1, D·].



S·+1, S· → D·+1, D·
Binary floating point → 32-bit binary integer
The decimal part is cut.

Instruction for inverse conversion

The inverse conversion is executed by DFLT (FNC 49) instruction.

→ **For FLT (FNC 49) instruction, refer to Section 12.10.**

Related devices

→ For the methods of zero, borrow and carry flags, refer to Subsection 6.5.2.

Device	Name	Description
M8020	Zero flag	Turns ON when the operation result is 0
M8021	Borrow flag	Turns ON when the conversion result is cut in the decimal part.
M8022	Carry flag	Turns ON when the operation result is outside the range from –32768 to 32767 (in 16-bit operation) or from –2,147,483,648 to 2,147,483,647 (in 32-bit operation) and overflow occurs. (The operation result is not reflected.)

Caution

1. Caution in the operation

- Values after the decimal point are rounded.

18.18 FNC130 – SIN / Floating Point Sine

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction obtains the sine value of an angle (in radians).

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D	FNC 130 SIN	P

16-bit Instruction	Mnemonic	Operation Condition
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DSIN	Continuous Operation
	DSINP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Device number storing an angle (in radians) in binary floating point	Real number (binary)
D·	Device number storing the sine value in binary floating point	

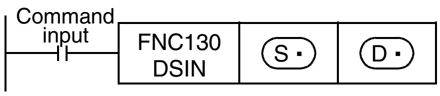
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·														✓	✓	✓			✓			✓		
D·														✓	✓	✓			✓					

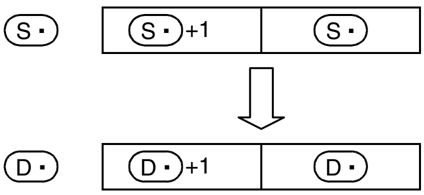
Explanation of function and operation

1. 32-bit operation (DSIN and DSINP)

A value of angle (binary floating point) specified in [S·+1, S·] is converted into the sine value, and transferred to [D·+1, D·].



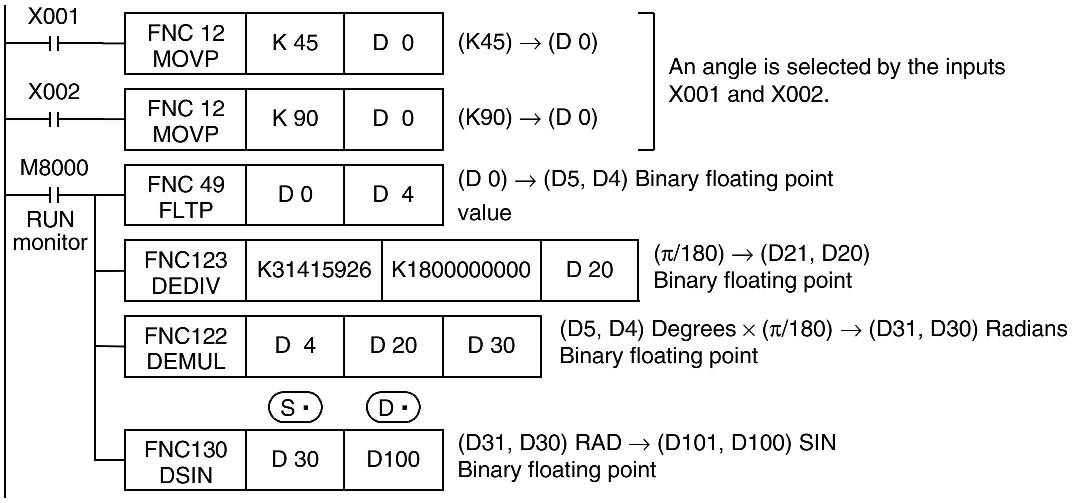
[S·+1, S·] RAD → [D·+1, D·] SIN
Binary floating point   Binary floating point



Value in radians (Value in degrees × π /180)
Binary floating point

Sine value
Binary floating point

Program example



18.19 FNC131 – COS / Floating Point Cosine

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction obtains the cosine value of an angle (in radians).

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D	FNC 131 COS	P

16-bit Instruction	Mnemonic	Operation Condition
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DCOS	Continuous Operation
	DCOSP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(S•)	Device number storing an angle (in radians) in binary floating point	Real number (binary)
(D•)	Device number storing the cosine value in binary floating point	

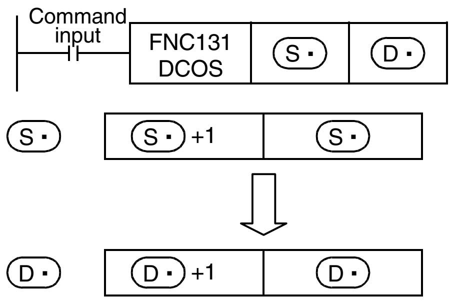
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S•)														✓	✓	✓			✓			✓		
(D•)														✓	✓	✓			✓					

Explanation of function and operation

1. 32-bit operation (DCOS and DCOSP)

A value of angle (binary floating point) specified in [(S•)+1, (S•)] is converted into the cosine value, and transferred to [(D•)+1, (D•)].



[(S•)+1, (S•)] RAD → [(D•)+1, (D•)] COS
Binary floating point   Binary floating point

Value in radians (Value in degrees × π /180)
Binary floating point

Cosine value
Binary floating point

18.20 FNC132 – TAN / Floating Point Tangent

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction obtains the tangent value of an angle (in radians).

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D	FNC 132 TAN	P

16-bit Instruction	Mnemonic	Operation Condition
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DTAN	Continuous Operation
	DTANP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(S·)	Device number storing an angle (in radians) in binary floating point	Real number (binary)
(D·)	Device number storing the tangent value in binary floating point	

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)														✓	✓	✓			✓			✓		
(D·)														✓	✓	✓			✓					

Explanation of function and operation

1. 32-bit operation (DTAN and DTANP)

A value of angle (binary floating point) specified in [(S·)+1, (S·)] is converted into the tangent value, and transferred to [(D·)+1, (D·)].

Command input	FNC132 DTAN	(S·)	(D·)

[(S·)+1, (S·)] RAD → [(D·)+1, (D·)] TAN
Binary floating point → Binary floating point

(S·)	(S·)+1	(S·)
(D·)	(D·)+1	(D·)

(S·): Value in radians (Value in degrees × π /180)
Binary floating point

⇩

(D·): Tangent value
Binary floating point

18.21 FNC133 – ASIN / Floating Point Arc Sine

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes the SIN^{-1} (arc sine) operation.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D	FNC 133 ASIN	P

16-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DASIN	Continuous Operation
	DASINP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Head device number storing a sine value used in the SIN^{-1} (arc sine) operation.	Real number (binary)
D·	Head device number storing the operation result	

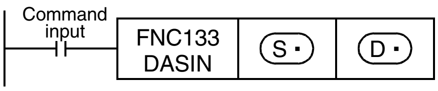
3. Applicable devices

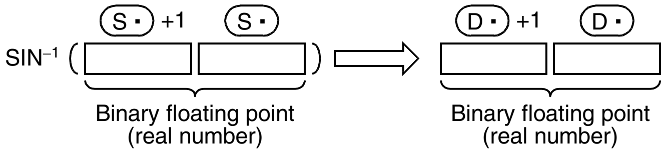
Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index				Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify		K	H	E	"□"	P
S·														✓	✓	✓			✓				✓		
D·														✓	✓	✓			✓						

Explanation of function and operation

1. 32-bit operation (DASIN and DASINP)

An angle is obtained from the sine value stored in [S·+1, S·], and stored to [D·+1, D·].
A real number can be directly specified as S·.





- The sine value stored in [S·+1, S·] can be set within the range from –1.0 to +1.0.
- The angle (operation result) stored in [D·+1, D·] is expressed in radians (from $-\pi/2$ to $\pi/2$).
  For conversion between radians and degrees, refer to RAD (FNC136) and DEG (FNC137) instructions.

→ **For RAD (FNC136) instruction, refer to Section 18.24.**
→ **For DEG (FNC137) instruction, refer to Section 18.25.**

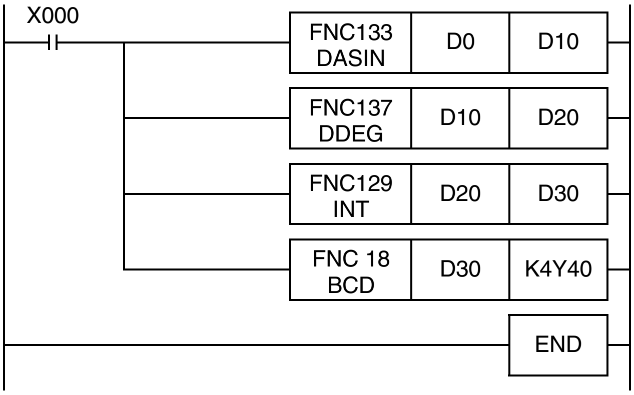
Error

An operation error is caused in the following case; The error flag M8067 turns ON, and the error code is stored in D8067.

- When a value specified in S· is outside the range from –1.0 to +1.0 (error code: K6706)

Program example

In the program example shown below, the SIN^{-1} value of data (binary floating point) stored in D0 and D1 is calculated, and the angle is output in 4-digit BCD to Y040 to Y057 when X000 turns ON.



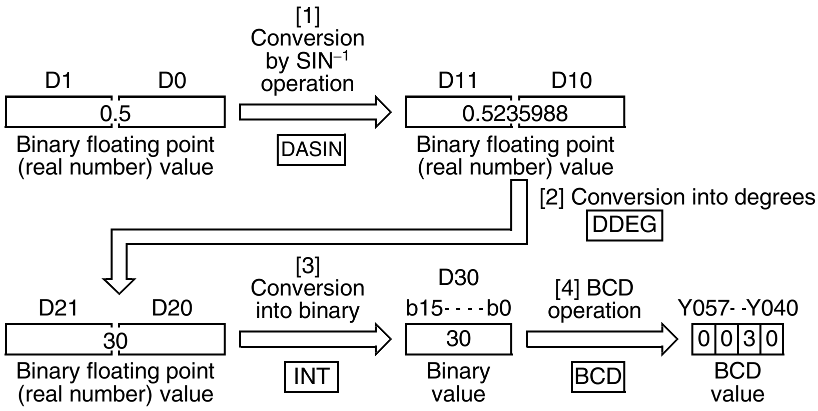
The angle (in radians) is calculated by the SIN^{-1} operation ([1]).

The value in radians is converted into the value in degrees ([2]).

The angle expressed in binary floating point (real number) is converted into an integer (binary) ([3]).

The angle expressed in integer (binary) is output to the display unit ([4]).

Operation when "0.5" is stored in D0 and D1



18.22 FNC134 – ACOS / Floating Point Arc Cosine

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes the COS^{-1} (arc cosine) operation.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D	FNC 134 ACOS	P

16-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DACOS	Continuous Operation
	DACOSP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(S·)	Head device number storing a cosine value used in the COS^{-1} (arc cosine) operation	Real number (binary)
(D·)	Head device number storing the operation result	

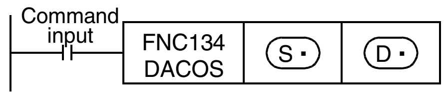
3. Applicable devices

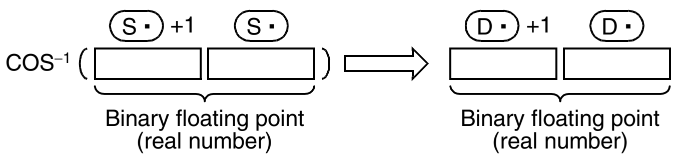
Operand Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)														✓	✓	✓			✓			✓		
(D·)														✓	✓	✓			✓					

Explanation of function and operation

1. 32-bit operation (DACOS and DACOSP)

An angle is obtained from the cosine value stored in [(S·)+1, (S·)], and stored to [(D·)+1, (D·)].
A real number can be directly specified as (S·).





- The cosine value stored in [(S·)+1, (S·)] can be set within the range from –1.0 to +1.0.
- The angle (operation result) stored in [(D·)+1, (D·)] is expressed in radians (from 0 to π).
  For conversion between radians and degrees, refer to RAD (FNC136) and DEG (FNC137) instructions.

→ **For RAD (FNC136) instruction, refer to Section 18.24.**
→ **For DEG (FNC137) instruction, refer to Section 18.25.**

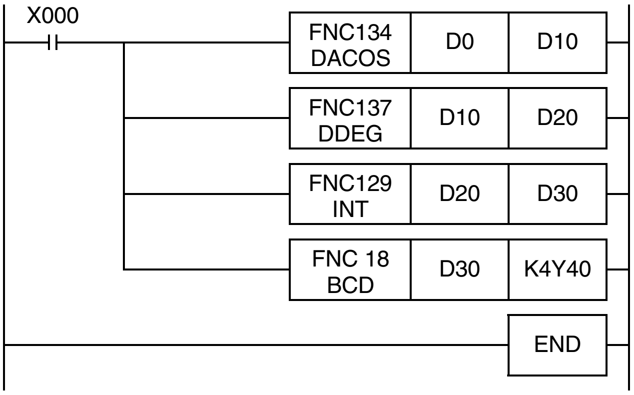
Error

An operation error is caused in the following case; The error flag M8067 turns ON, and the error code is stored in D8067.

- When a value specified in (S·) is outside the range from –1.0 to +1.0 (error code: K6706)

Program example

In the program example shown below, the COS^{-1} value of data (binary floating point) stored in D0 and D1 is calculated, and the angle is output in 4-digit BCD to Y040 to Y057 when X000 turns ON.



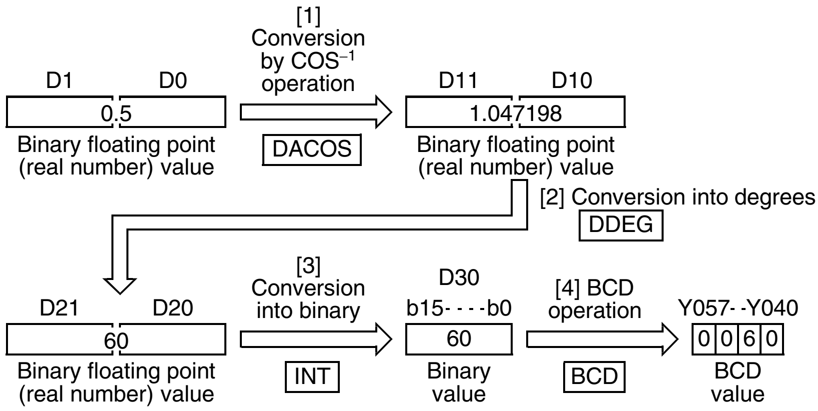
The angle (in radians) is calculated by the COS^{-1} operation ([1]).

The value in radians is converted into the value in degrees ([2]).

The angle expressed in the binary floating point (real number) is converted into an integer (binary) ([3]).

The angle expressed in integer (binary) is output to the display unit ([4]).

Operation when "0.5" is stored in D0 and D1



11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FMC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

18.23 FNC135 – ATAN / Floating Point Arc Tangent

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction executes the TAN^{-1} (arc tangent) operation.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D	FNC 135 ATAN	P

16-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DATAN	Continuous Operation
	DATANP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(S·)	Head device number storing a tangent value used in the TAN^{-1} (arc tangent) operation	Real number (binary)
(D·)	Head device number storing the operation result	

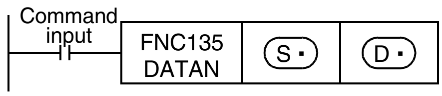
3. Applicable devices

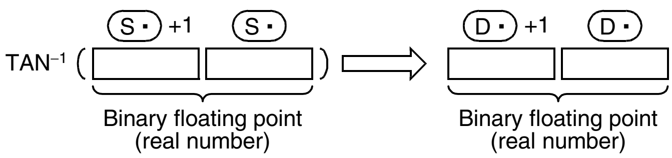
Operand Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index				Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify		K	H	E	"□"	P
(S·)														✓	✓	✓			✓				✓		
(D·)														✓	✓	✓			✓						

Explanation of function and operation

1. 32-bit operation (DATAN and DATANP)

An angle is obtained from the tangent value stored in [(S·)+1, (S·)], and stored to [(D·)+1, (D·)].
A real number can be directly specified as (S·).



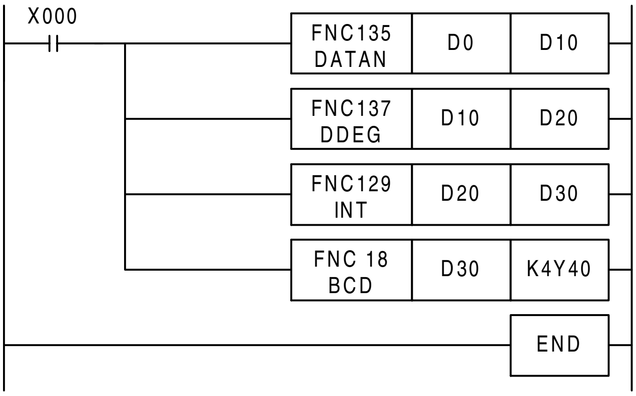


- The angle (operation result) stored in [(D·)+1, (D·)] is expressed in radians (from $-\pi/2$ to $+\pi/2$).
  For conversion between radians and degrees, refer to RAD (FNC136) and DEG (FNC137) instructions.

→ **For RAD (FNC136) instruction, refer to Section 18.24.**
→ **For DEG (FNC137) instruction, refer to Section 18.25.**

Program example

In the program example shown below, the TAN^{-1} value of data (binary floating point) stored in D0 and D1 is calculated, and the angle is output in 4-digit BCD to Y040 to Y057 when X000 turns ON.



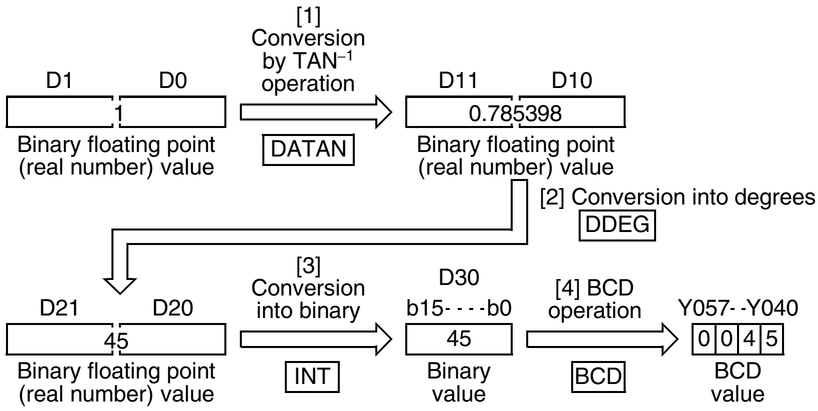
The angle (in radians) is calculated by the TAN^{-1} operation ([1]).

The value in radians is converted into the value in degrees ([2]).

The angle expressed in binary floating point (real number) is converted into an integer (binary) ([3]).

The angle expressed in integer (binary) is output to the display unit ([4]).

Operation when "1" is stored in D0 and D1



18.24 FNC136 – RAD / Floating Point Degrees to Radians Conversion

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction converts a value in degrees into a value in radians.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D FNC 136 RAD P

16-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DRAD	Continuous Operation
	DRADP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Head device number storing a value in degrees to be converted into a value in radians	Real number (binary)
D·	Head device number storing a value in radians acquired by conversion	

3. Applicable devices

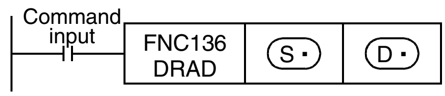
Oper-and Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·														✓	✓	✓			✓			✓		
D·														✓	✓	✓			✓					

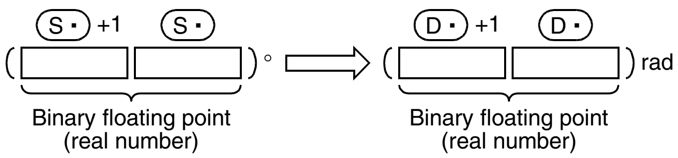
Explanation of function and operation

1. 32-bit operation (DRAD and DRADP)

The unit of [S·+1, S·] is converted from degrees into radians, and the operation result is stored to [D·+1, D·].

A real number can be directly specified as S·.



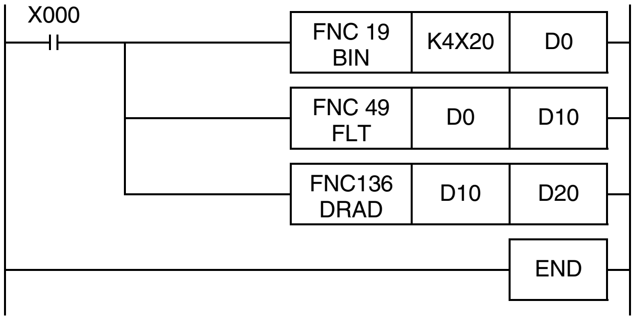


- The conversion from degrees into radians is executed as follows:

$$\text{Radians} = \text{Degrees} \times \frac{\pi}{180}$$

Program example

In the program example shown below, a 4-digit BCD value set in degrees in X020 to X037 is converted into a binary floating point value in radians, and stored to D20 and D21 when X000 turns ON.

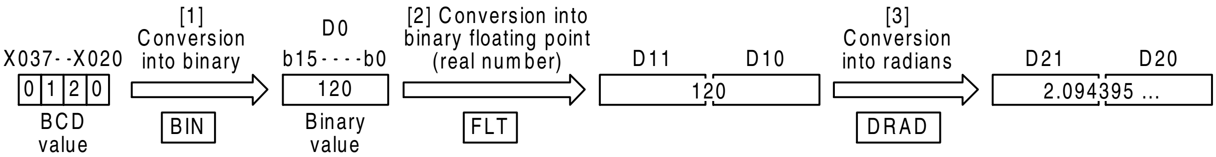


Angle to be converted into radians is input ([1]).

The input angle is converted into binary floating point (real number) ([2]).

The angle is converted from degrees into radians ([3]).

Operation when "120" is specified in X020 to X037



11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FMC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

18.25 FNC137 – DEG / Floating Point Radians to Degrees Conversion

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction converts a value in radians into a value in degrees.

→ **For handling of floating point, refer to Subsection 5.1.3.**

1. Instruction format

D FNC 137 DEG P

16-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DDEG	Continuous Operation
	DDEGP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Head device number storing a value in radians to be converted into a value in degrees	Real number (binary)
D·	Head device number storing a value in degrees acquired by conversion	

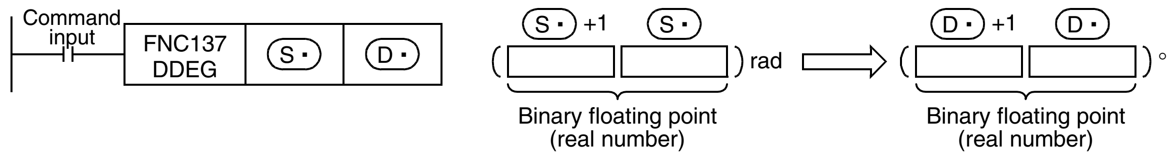
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·														✓	✓	✓			✓			✓		
D·														✓	✓	✓			✓					

Explanation of function and operation

1. 32-bit operation (DDEG and DDEGP)

The unit of [S·+1, S·] is converted from radians into degrees, and the operation result is stored to [D·+1, D·].

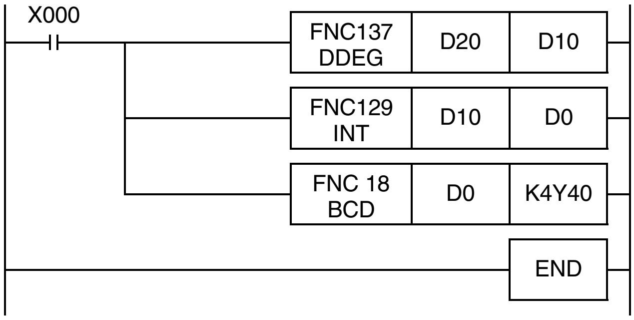


- The conversion from radians into degrees is executed as follows:

$$\text{Degrees} = \text{Radians} \times \frac{180}{\pi}$$

Program example

In the program example shown below, a binary floating point value set in radians in D20 and D21 is converted into a BCD value in degrees, and stored to Y040 and Y057 when X000 turns ON.

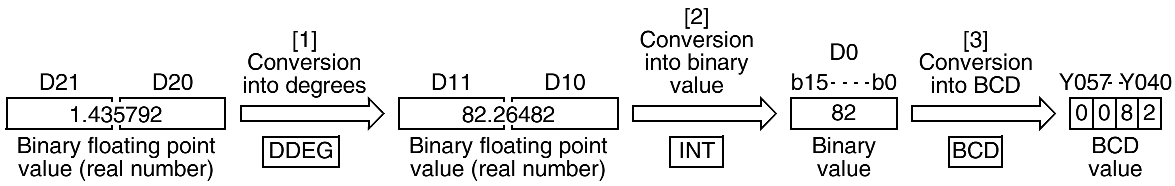


A value in radians is converted into a value in degrees ([1]).

The angle in binary floating point (real number) is converted into an integer ([2]).

The converted integer is output to the display unit ([3]).

Operation when "1.435792" is specified in D20 and D21



19. Data Operation 2 – FNC140 to FNC149

FNC140 to FNC149 provide instructions for executing complicated processing for fundamental applied instructions and for executing special processing.

FNC No.	Mnemonic	Symbol	Function	Reference
140	WSUM	WSUM S D n	Sum of Word Data	Section 19.1
141	WTOB	WTOB S D n	WORD to BYTE	Section 19.2
142	BTOW	BTOW S D n	BYTE to WORD	Section 19.3
143	UNI	UNI S D n	4-bit Linking of Word Data	Section 19.4
144	DIS	DIS S D n	4-bit Grouping of Word Data	Section 19.5
145	–			
146	–			
147	SWAP	SWAP S	Byte Swap	Section 19.6
148	–			
149	SORT2	SORT2 S m1 m2 D n	Sort Tabulated Data 2	Section 19.7

19.1 FNC140 – WSUM / Sum of Word Data

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

This instruction calculates the sum of consecutive 16-bit or 32-bit data.
When calculating the addition data (sum value) in units of byte (8 bits), use the CCD (FNC 84) instruction.

→ **For CCD (FNC 84) instruction, refer to Section 16.5.**

1. Instruction format

D	FNC 140 WSUM	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	WSUM	Continuous Operation
	WSUMP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DWSUM	Continuous Operation
	DWSUMP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S•	Head device number storing data whose sum is calculated	16- or 32-bit binary
D•	Head device number storing sum	32- or 64-bit binary
n	Number of data (0 < n)	16- or 32-bit binary

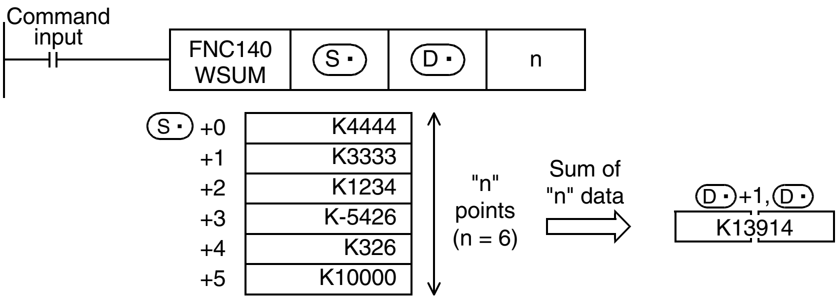
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•												✓	✓	✓	✓	✓			✓					
D•												✓	✓	✓	✓	✓			✓					
n														✓	✓					✓	✓			

Explanation of function and operation

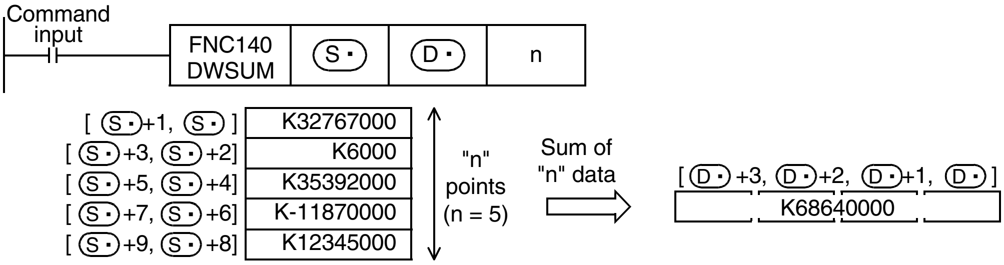
1. 16-bit operation (WSUM and WSUMP)

The sum of "n" 16-bit data starting from S• is stored as 32-bit data in [D•+1, D•].



2. 32-bit operation (DWSUM and DWSUMP)

The sum of "n" 32-bit data starting from [S·+1, S·] is stored as 64-bit data in [D·+3, D·+2, D·+1, D·].



Related instruction

Instruction	Description
CCD (FNC 84)	Check code Calculates the sum of 16-bit data in units of byte (8 bits) and the horizontal parity.

Caution

In the 32-bit operation, the acquired sum is 64-bit data. FX3U and FX3UC PLCs cannot handle 64-bit data. When the sum is within the numeric range of 32-bit data (K-2,147,483,648 to K2,147,483,647), however, FX3U and FX3UC PLCs can handle the low-order 32 bits of 32-bit data as the sum while ignoring the high-order 32 bits.

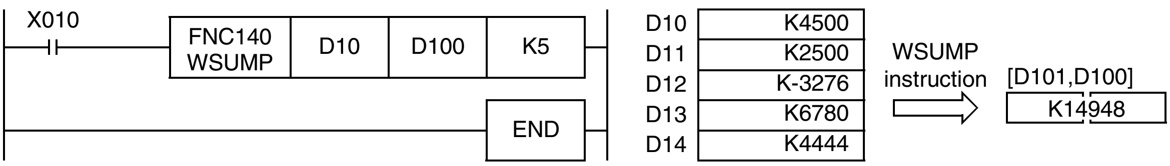
Errors

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When "n" points starting from S· are outside the specified device range (error code: K6706)
- When "n" is smaller than or equivalent to "0" (error code: K6706)
- When D· are outside the specified device range. (error code: K6706)

Program example

In the program shown below, the sum of 16-bit data stored in D10 to D14 is stored in [D101, D100].



19.2 FNC141 – WTOB / WORD to BYTE

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

This instruction separates consecutive 16-bit data in byte units (8 bits).

1. Instruction format

FNC 141
WTOB P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	WTOB	Continuous Operation
	WTOBP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
S·	Head device number storing data to be separated in byte units	16-bit binary
D·	Head device number storing result of separation in byte units	
n	Number of byte data to be separated (0 ≤ n)	

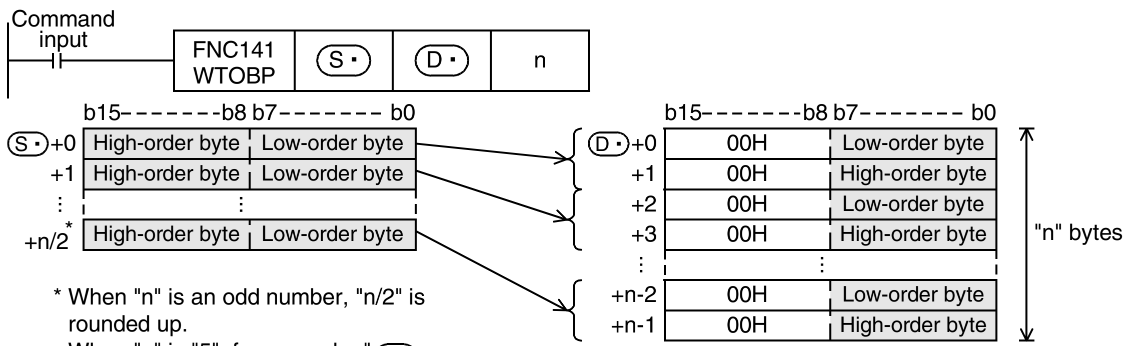
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·												✓	✓	✓	✓				✓					
D·												✓	✓	✓	✓				✓					
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (WTOB and WTOBP)

1) "n/2" 16-bit data stored in S· and later is separated into "n" bytes, and stored in "n" devices starting from D· as shown below.

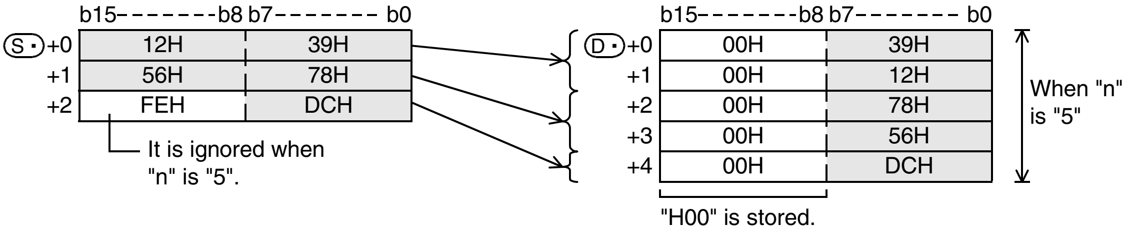


* When "n" is an odd number, "n/2" is rounded up.
When "n" is "5", for example, "S· +3" is used.

2) "H00" is stored in the high-order byte (8 bits) of each device (D· and later) storing the separated byte data.

11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FMC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

3) When "n" is an odd number, only the low-order byte (8 bits) of the final separation source device is regarded as the target data as shown in the figure below.
   For example, when "n" is "5", the data from (S•) to the low-order byte (8 bits) of (S•)+2 is stored in (D•) to (D•)+4.



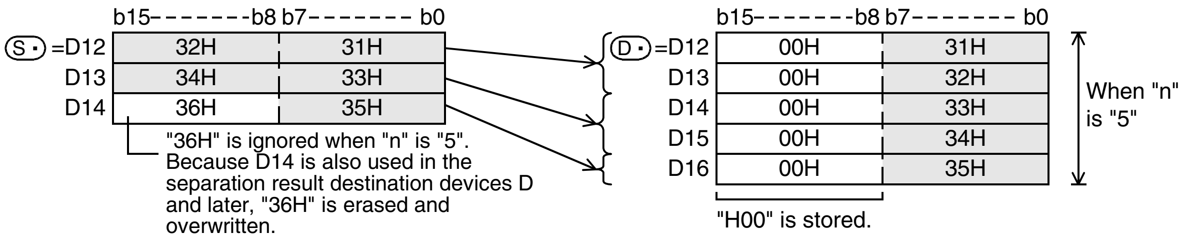
4) When "n" is "0", WTOB instruction is not executed.

Related instruction

Instruction	Description
BTOW (FNC142)	Combines the low-order 8 bits (low-order byte) of consecutive 16-bit data.

Caution

Devices storing the separation source data can overlap devices storing the separated data.
When "n" is an odd number, however, the high-order byte (8 bits) of the final separation source device is overwritten and erased.



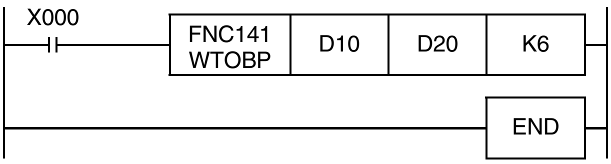
Errors

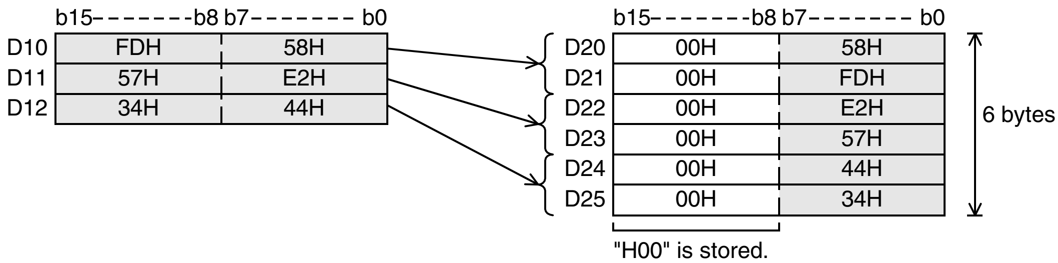
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the separation source devices (S•) to (S•)+n/2 are outside the specified device range (error code: K6706)
  When "n" is an odd number, the number of a rounded up value decides the number of devices. (error code: K6706)
- When the separated data destination devices (D•) to (D•)+n-1 are outside the specified device range (error code: K6706)

Program example

In the program shown below, the data stored in D10 to D12 is separated in byte units, and stored in D20 to D25.





19.3 FNC142 – BTOW / BYTE to WORD

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

This instruction combines the low-order 8 bits (low-order byte) of consecutive 16-bit data.

1. Instruction format

FNC 142 BTOW P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	BTOW	Continuous Operation
	BTOWP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
(S·)	Head device number storing data to be combined in byte units	16-bit binary
(D·)	Head device number storing data acquired by combination in byte units	16-bit binary
n	Number of byte data to be combined (0 ≤ n)	16-bit binary

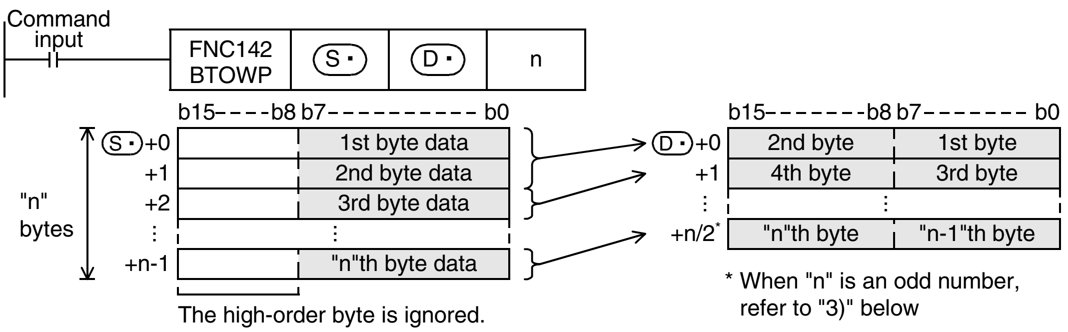
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)												✓	✓	✓	✓				✓					
(D·)												✓	✓	✓	✓				✓					
n														✓	✓					✓	✓			

Explanation of function and operation

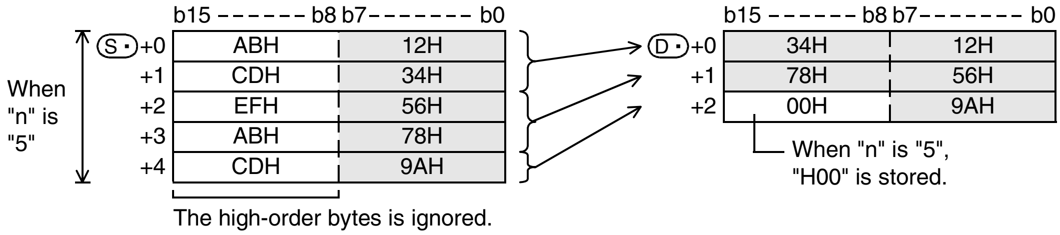
1. 16-bit operation (BTOW and BTOWP)

1) The low-order byte (8 bits) of "n" 16-bit data starting from (S·) is combined, and stored in "n/2" devices starting from (D·) as shown below.



2) The high-order byte (8 bits) of each combination source 16-bit data ((S·) and later) is ignored.

3) When "n" is an odd number, "H00" is stored in the high-order byte (8 bits) of the final one among the combination result destination devices as shown below.
   For example, when "n" is "5", the low-order byte (8 bits) of S· to S·+4 is stored in D· to D·+2, and "H00" is stored in the high-order byte (8 bits) of D·+2.



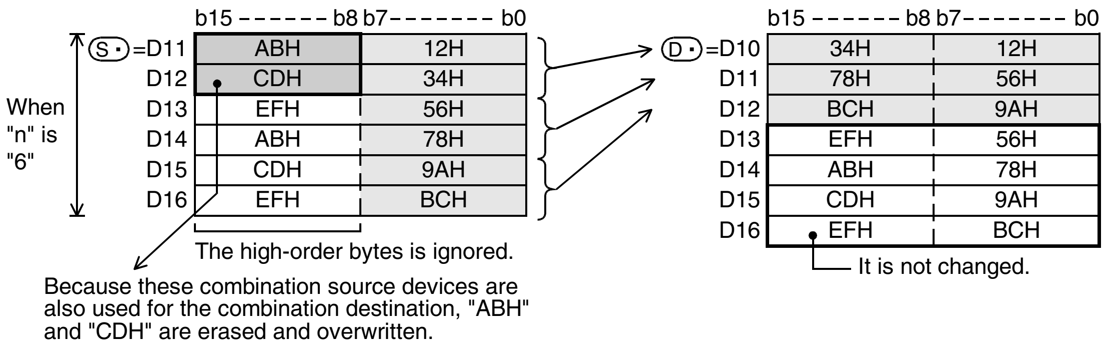
4) When "n" is "0", the BTOW instruction is not executed.

Related instruction

Instruction	Description
WTOB (FNC141)	Separates consecutive 16-bit data in byte units (8 bits).

Caution

Devices storing the combination source data may be equivalent to devices storing the combined data.
After combination, however, the high-order byte (8 bits) of the combination source data stored in the devices used for the combination destination data is erased and overwritten with the data acquired by combining the high-order byte (8 bits).



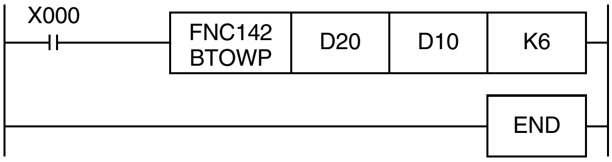
Errors

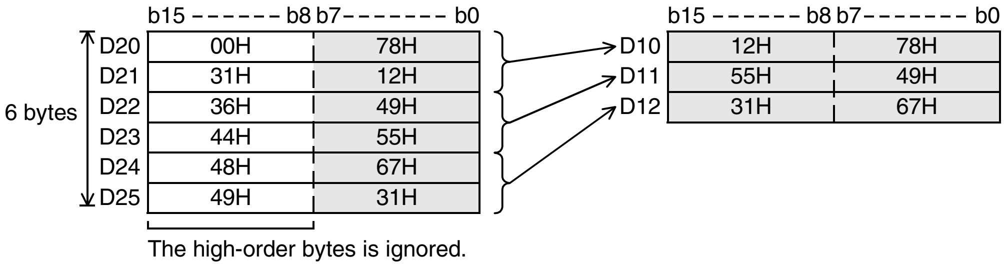
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the combination source devices S· to S·+n-1 are outside the specified device range (error code: K6706)
- When the combined data destination devices D· to D·+n/2 are outside the specified device range (error code: K6706)
  When "n" is an odd number, the number of a rounded up value decides the numer of devices. (error code: K6706)

Program example

In the program shown below, the low-order byte (8 bits) data stored in D20 to D25 is combined, and stored in D10 to D12.





19.4 FNC143 – UNI / 4-bit Linking of Word Data

FX3UC Ver.2.20 → FX3U Ver.2.20 →

Outline

This instruction combines the low-order 4 bits of consecutive 16-bit data.

1. Instruction format

FNC 143 UNI P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	UNI	Continuous Operation
	UNIP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
S·	Head device number storing data to be combined	16-bit binary
D·	Head device number storing combined data	
n	Number of data to be combined (0 to 4, When "n" is "0", UNI instruction is not executed.)	

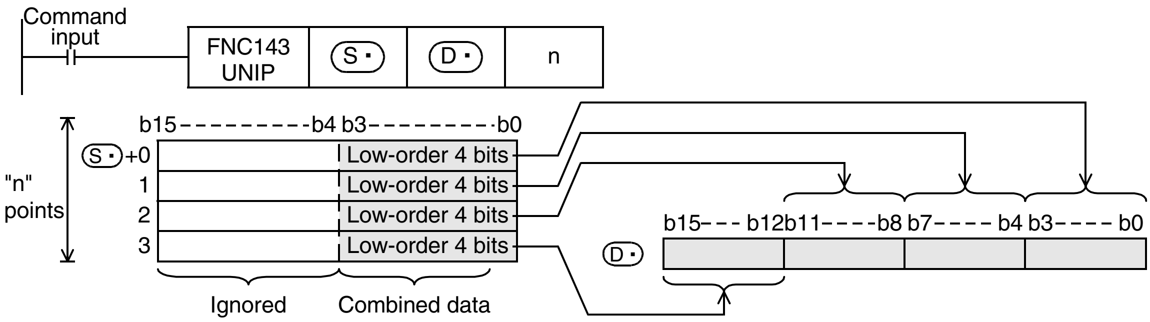
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·												✓	✓	✓	✓				✓					
D·												✓	✓	✓	✓				✓					
n														✓	✓					✓	✓			

Explanation of function and operation

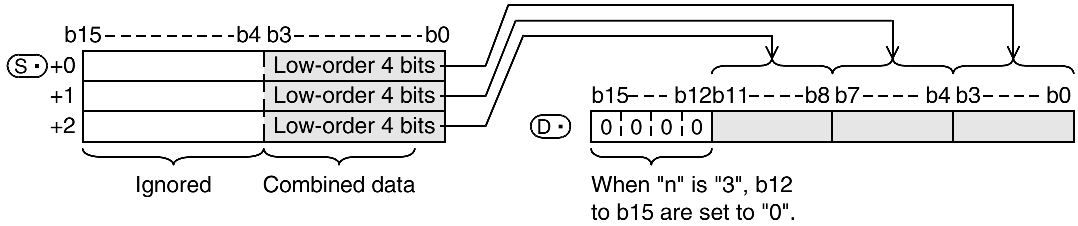
1. 16-bit operation (UNI/UNIP)

1) The low-order 4 bits of "n" 16-bit data starting from S· are combined, and stored in D· as shown below.



2) Specify a number 1 to 4 in "n".
In the case of "n = 0", UNI instruction is not executed.

3) In the case of "1 ≤ n ≤ 3", the high-order {4 × (4-n)} bits of (D•) are set to "0".
For example, when "n" is "3", the low-order 4 bits of (S•) to (S•)+2 are stored in b0 to b11 of (D•), and the high-order 4 bits of (D•) are set to "0".



Related instruction

Instruction	Description
DIS (FNC144)	Separates 16-bit data in 4-bit units.

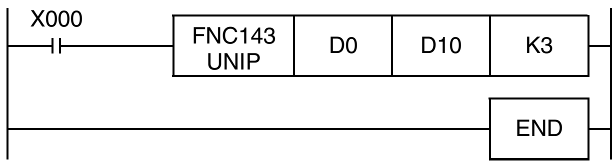
Errors

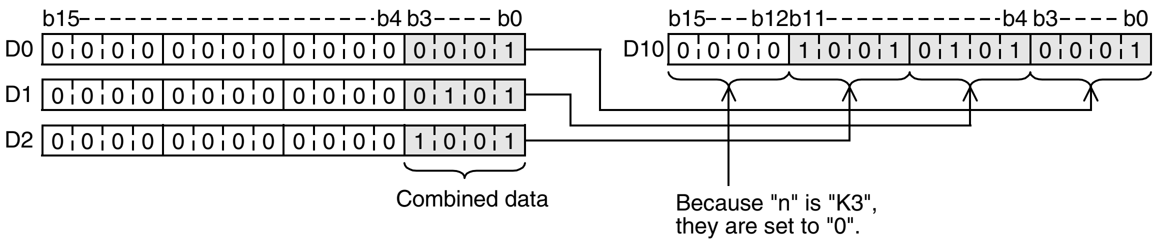
An operation error occurs in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When (S•) to (S•)+n are outside the specified device range (error code: K6706)
- When "n" is outside the range from "0 to 4" (error code: K6706)

Program example

In the program below, the low-order 4 bits of D0 to D2 are combined and stored in D10 when X000 turns ON.





19.5 FNC144 – DIS / 4-bit Grouping of Word Data

FX3UC Ver.2.20 → FX3U Ver.2.20 →

Outline

This instruction separates 16-bit data into 4 bit units.

1. Instruction format

FNC 144 DIS P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	DIS	Continuous Operation
	DISP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
S·	Device number storing data to be separated	16-bit binary
D·	Head device number storing separated data	
n	Number of data to be separated (0 to 4) (When "n" is "0", DIS instruction is not executed.)	

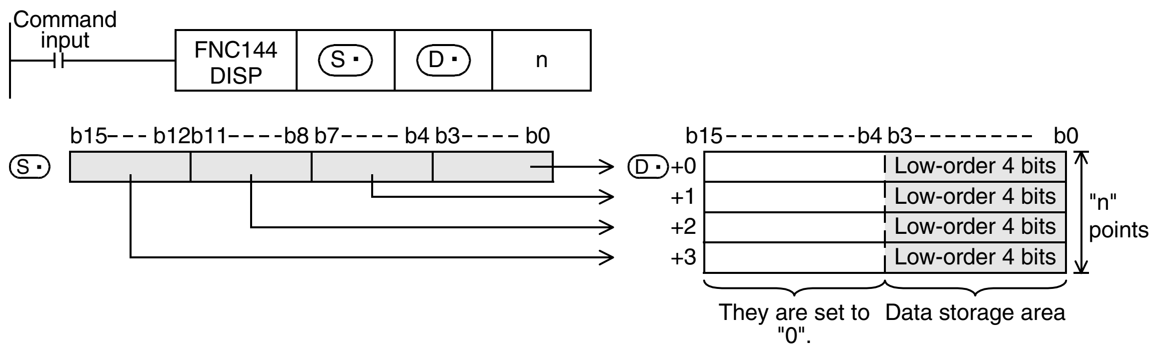
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·												✓	✓	✓	✓				✓					
D·												✓	✓	✓	✓				✓					
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (DIS and DISP)

1) 16-bit data stored in S· is separated in 4-bit units, and stored in D· as shown below.



2) Specify a number 1 to 4 in "n".
In the case of "n = 0", DIS instruction is not executed.

3) High-order 12 bits of "n" devices starting from D· are set to "0".

Related instruction

Instruction	Description
UNI (FNC143)	Combines low-order 4 bits of 16-bit data.

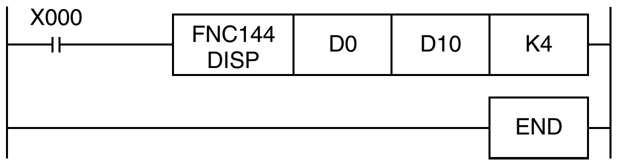
Errors

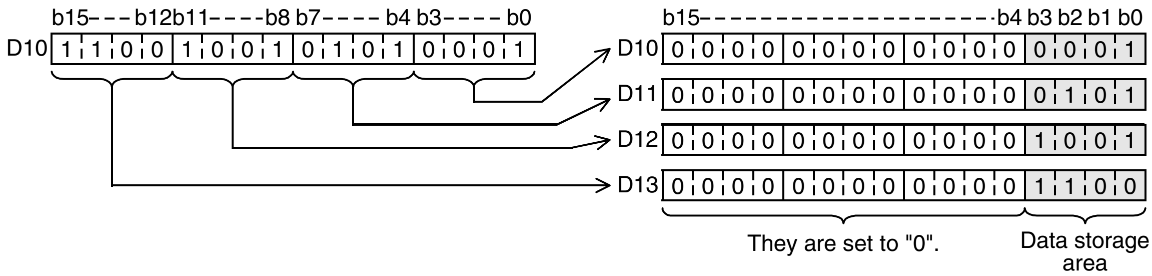
An operation error occurs in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When (D·) to (D·)+n are outside the specified device range (error code: K6706)
- When "n" is outside the range from "0 to 4" (error code: K6706)

Program example

In the program below, D0 is separated into 4 bit units and stored in D10 to D13 when X000 turns ON.





19.6 FNC147 – SWAP / Byte Swap

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction swaps the high-order 8 bits and low-order 8 bits of a word device.

1. Instruction format

D	FNC 147 SWAP	P

16-bit Instruction	Mnemonic	Operation Condition
3 steps	SWAP	Continuous Operation
	SWAPP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
5 steps	DSWAP	Continuous Operation
	DSWAPP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S·	Word device whose high-order 8 bits and low-order 8 bits are swapped for each other	16- or 32-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

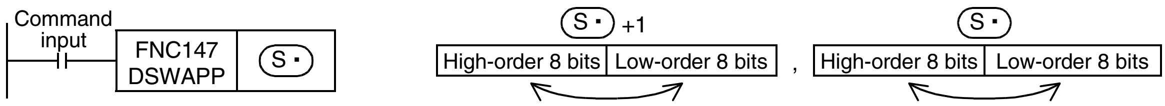
Explanation of function and operation

1. 16-bit operation (SWAP and SWAPP)

High-order 8 bits and low-order 8 bits are swapped for each other.

2. 32-bit operation (DSWAP and DSWAPP)

High-order 8 bits and low-order 8 bits are swapped for each other in each word device.



Caution

- When the continuous operation type instruction is used, swapping is executed in each operation cycle. This instruction works in the same way as the extension function of the XCH (FNC 17) instruction.

19.7 FNC149 – SORT2 / Sort Tabulated Data 2

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

This instruction sorts a data table consisting of data (lines) and group data (columns) based on a specified group data (column) sorted by line in either ascending or descending order. This instruction stores the data (lines) in serial devices facilitating the addition of data (lines).
On the other hand, the SORT (FNC 69) instruction stores the group data (columns) in serial devices, and sorts a table in ascending order only.

→ **For SORT (FNC 69) instruction, refer to Section 14.10.**

1. Instruction format

FNC 149
D SORT2

16-bit Instruction	Mnemonic	Operation Condition
11 steps	SORT	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
21 steps	DSORT2	Continuous Operation

2. Set data

Operand type	Description	Data type
(S)	Head device number storing the data table [which occupies m1 × m2 points]	16- or 32-bit binary
m1	Number of data (lines) [1 to 32]	
m2	Number of group data (columns) [1 to 6]	
(D)	Head device number storing the operation result [which occupies m1 × m2 points]	
n	Column number of group data (column) used as the basis of sorting [1 to m2]	

3. Applicable devices

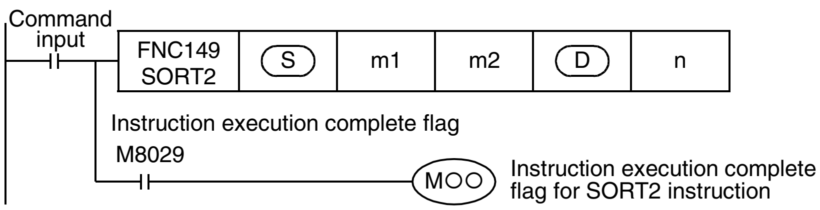
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S)														✓	✓									
m1														✓	✓					✓	✓			
m2																				✓	✓			
(D)														✓	✓									
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (SORT2)

In the data table (sorting source) having (m1 x m2) points from (S), data lines are sorted in the ascending or descending order based on the group data in column No. "n", and the result is stored in the data table (occupying m1 x m2 points) from (D).

→ **For operation examples, refer to Page 523.**



The data table configuration is explained in an example in which the sorting source data table has 3 lines and 4 columns (m1 = K3, m2 = K4). For the sorting result data table, understand S as D.

Column No. / Line No.		Number of groups (m2 = K4)			
		1	2	3	4
		Control number	Height	Weight	Age
Number of data (m1 = 3)	1	S	S+1	S+2	S+3
	2	S+4	S+5	S+6	S+7
	3	S+8	S+9	S+10	S+11

- Set the sorting order by setting M8165 to ON or OFF.

	Sorting order
M8165=ON	Descending order
M8165=OFF	Ascending order

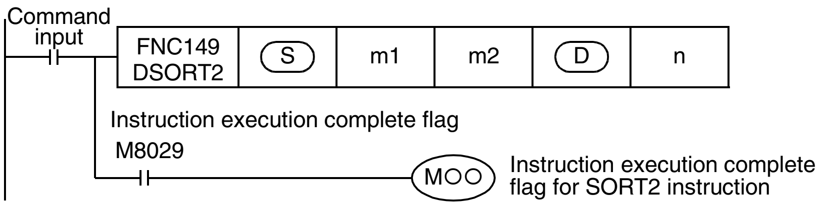
- When the command input turns ON, data sorting is started. Data sorting is completed after "m1" scans, and the instruction execution complete flag M8029 is set to ON.

→ **For the instruction execution complete flag use method, refer to Subsection 6.5.2.**

2. 32-bit operation (DSORT2)

In the data table (sorting source) having (m1 × m2) points from [S+1, S], data lines are sorted in the ascending or descending order based on the group data in the column No. "n", and the result is stored in the data table (sorting result) having (m1 × m2) points from [D+1, D].

→ **For operation examples, refer to Page 523.**



The data table configuration is explained in an example in which the sorting source data table has 3 lines and 4 columns (m1 = K3, m2 = K4). For the sorting result data table, understand S as D.

Column No. / Line No.		Number of groups (m2 = K4)			
		1	2	3	4
		Control number	Height	Weight	Age
Number of data (m1 = 3)	1	[S+1, S]	[S+3, S+2]	[S+5, S+4]	[S+7, S+6]
	2	[S+9, S+8]	[S+11, S+10]	[S+13, S+12]	[S+15, S+14]
	3	[S+17, S+16]	[S+19, S+18]	[S+21, S+20]	[S+23, S+22]

- Set the sorting order by setting M8165 to ON or OFF.

	Sorting order
M8165=ON	Descending order
M8165=OFF	Ascending order

- When a data register D or extension register (R) is used for "m1", the data length is 32 bits.
  For example, when "m1" is specified in D0, "m1" is 32-bit data stored in [D1, D0].
- When the command input turns ON, data sorting is started. Data sorting is completed after "m1" scans, and the instruction execution complete flag M8029 is set to ON.

→ **For the instruction execution complete flag use method, refer to Subsection 6.5.2.**

3. Operation examples

When the instruction is executed with "n = K2 (column No. 2)" and "n = K3 (column No. 3)" for the following sorting source data, the operations shown below result.
The operation examples below indicate 16-bit operations. In the case of 32-bit operation, construct the data table with 32-bit binary data.
It is recommended to put a serial number such as a control number in the first column so that the original line number can be estimated based on the contents.

Sorting source data

Column No. / Line No.		Number of groups (m2 = K4)			
	Column No.	1	2	3	4
Line No.		Control number	Height	Weight	Age
Number of data (m1 = 5)	1	S	S +1	S +2	S +3
		1	150	45	20
	2	S +4	S +5	S +6	S +7
		2	180	50	40
	3	S +8	S +9	S +10	S +11
		3	160	70	30
	4	S +12	S +13	S +14	S +15
		4	100	20	8
	5	S +16	S +17	S +18	S +19
		5	150	50	45

1) Sorting result when the instruction is executed with "n = K2 (column No. 2)" (in the case of ascending order)

Column No. / Line No.	1	2	3	4
	Control number	Height	Weight	Age
1	D	D +1	D +2	D +3
	4	100	20	8
2	D +4	D +5	D +6	D +7
	1	150	45	20
3	D +8	D +9	D +10	D +11
	5	150	50	45
4	D +12	D +13	D +14	D +15
	3	160	70	30
5	D +16	D +17	D +18	D +19
	2	180	50	40

2) Sorting result when the instruction is executed with "n = K3 (column No. 3)" (in the case of descending order)

Column No. / Line No.	1	2	3	4
	Control number	Height	Weight	Age
1	D	D +1	D +2	D +3
	3	160	70	30
2	D +4	D +5	D +6	D +7
	2	180	50	40
3	D +8	D +9	D +10	D +11
	5	150	50	45
4	D +12	D +13	D +14	D +15
	1	150	45	20
5	D +16	D +17	D +18	D +19
	4	100	20	8

Related devices

→ For the instruction execution complete flag use method, refer to Subsection 6.5.2.

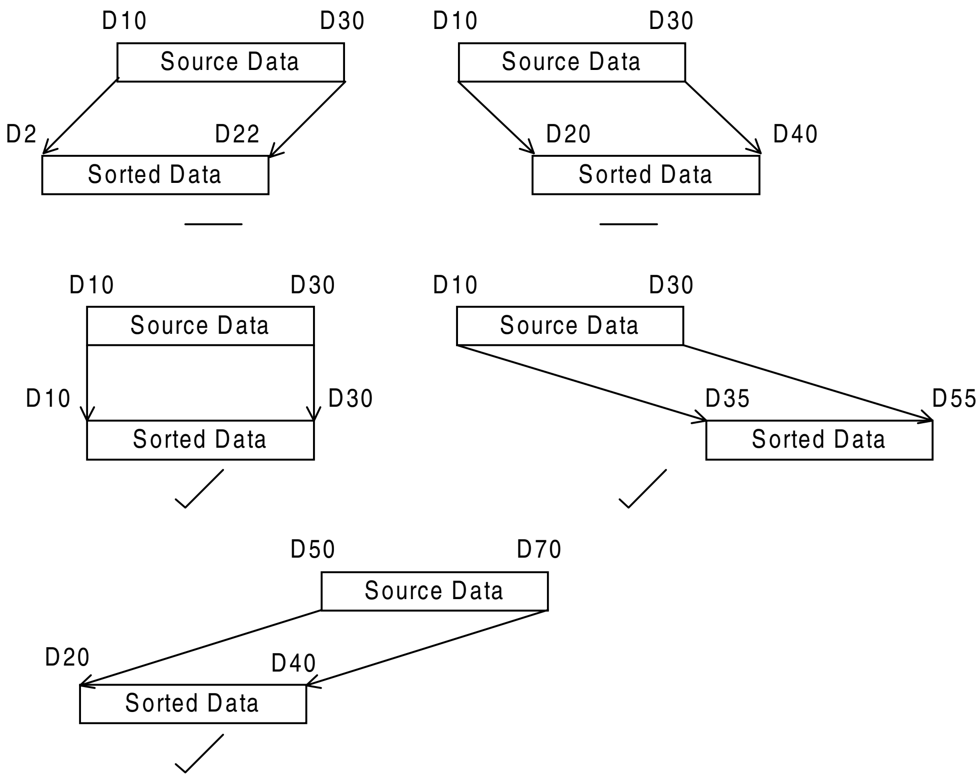
Device	Name	Description
M8029	Instruction execution complete	Turns ON when data sorting is completed.
M8165	Descending order	Sorts data in the descending order when set to ON. Sorts data in the ascending order when set to OFF.

Related instruction

Instruction	Description
SORT (FNC 69)	Sort tabulated data This instruction sorts a data table consisting of data (lines) and group data (columns) based on a specified group data (column) sorted by line in ascending order. This instruction stores the group data (columns) in serial devices.

Cautions

- Do not change the contents of operands and data during operation.
- To execute SORT2 instruction again, set the command input to OFF once, then ON again.
- Limitation in number of SORT2 instructions
  Up to two SORT2 instructions can be simultaneously driven in a program.
- Writing during RUN is disabled for a circuit block including SORT2 instruction.
- When the same device is specified in S and D
  The source data is overwritten with the data acquired by sorting.
  Pay close attention not to change the contents of S until execution of SORT2 instruction is completed.
- Ensure that the sorted data does not overlap with the source data.



20. Positioning Control – FNC150 to FNC159

FNC150 to FNC159 provide positioning instructions using the pulse output function built in PLCs.

→ **For details, refer to the Positioning Control Edition.**

FNC No.	Mnemonic	Symbol	Function	Reference
150	DSZR	DSZR S1 S2 D1 D2	DOG Search Zero Return	Section 20.1
151	DVIT	DVIT S1 S2 D1 D2	Interrupt Positioning	Section 20.2
152	TBL	TBL D n	Batch Data Positioning Mode	Section 20.3
153	–			–
154	–			–
155	ABS	ABS S D1 D2	Absolute Current Value Read	Section 20.4
156	ZRN	ZRN S1 S2 S3 D	Zero Return	Section 20.5
157	PLSV	PLSV S D1 D2	Variable Speed Pulse Output	Section 20.6
158	DRVI	DRVI S1 S2 D1 D2	Drive to Increment	Section 20.7
159	DRVA	DRVA S1 S2 D1 D2	Drive to Absolute	Section 20.8

Caution on writing during RUN

During RUN, avoid writing while any positioning control instruction (FNC150, FNC151, or FNC156 to FNC159) is executed (that is, while pulses are output).
If writing is excuted during RUN to a circuit block including a target instruction below while pulses are output, the PLC executes the operation shown below.

Target instruction		PLC operation when writing excuted during RUN while instruction is executed
DSZR (FNC150)		Decelerates and stops pulse output.
DVIT (FNC151)		
TBL (FNC152)		Disables writing during RUN.
ZRN (FNC156)		Decelerates and stops pulse output.
PLSV (FNC157)	During operation with acceleration/deceleration*1	Decelerates and stops pulse output.
	During operation without acceleration/deceleration	Immediately stops pulse output.
DRVI (FNC158) DRVA (FNC159)		Decelerates and stops pulse output.

*1. Supported in Ver.2.20 or later

20.1 FNC150 – DSZR / Dog Search Zero Return

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes zero return, and aligns the mechanical position with a present value register inside the PLC.
In addition, this instruction enables the following functions not supported by the ZRN (FNC156) instruction:

- DOG search function
- Zero return by the near-point (dog) signal and zero-phase signal
  It is not possible, however, to count the zero-phase signal and then determine the zero point.

→ **For explanation of the instruction, refer to the Positioning Control Edition.**
→ **For cautions on using special high speed output adapters, refer to the Positioning Control Edition.**

1. Instruction format

FNC 150
DSZR

16-bit Instruction	Mnemonic	Operation Condition
9 steps	DSZR	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand type	Description	Data type
S1•	Device number inputting near-point signal (dog)	Bit
S2•	Input number inputting zero-phase signal	
D1•	Device number outputting pulse	
D2•	Device number to which rotation direction signal is output	

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•	✓	✓	✓	✓			▲1												✓					
S2•	▲2																		✓					
D1•		▲3																	✓					
D2•		▲4	✓	✓			▲1												✓					

▲1 : "D□.b" cannot be indexed with index registers (V and Z).
▲2 : Specify either one among X000 to X007.
▲3 : Specify a transistor output Y000, Y001, or Y002 in the main unit or Y000, Y001, Y002*2, or Y003*2 in a special high speed output adapter*1.

*1. Special high speed output adapters cannot be connected to the FX3UC-32MT-LT.

*2. For specifying Y002 or Y003 in a special high speed output adapter, the second special high speed output adapter is required.

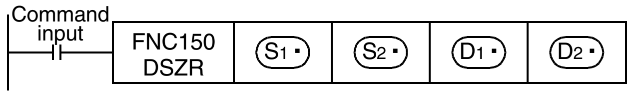
Points

- When using a relay output type FX3U PLC, a special high speed output adapter is required.
- Outputs of special high speed output adapters work as differential line drivers.

▲4 : When using a special high speed output adapter for the pulse output destination in an FX3U PLC, the rotation direction signal must be used by the following table output.
When using a built-in transistor output for the pulse output destination in an FX3U/FX3UC PLC, the rotation direction signal must use transistor output.

Special high speed output adapter No.	Pulse output	Rotation direction output
No. 1 (1st unit)	D1• =Y000	D2• =Y004
	D1• =Y001	D2• =Y005
No. 2 (2nd unit)	D1• =Y002	D2• =Y006
	D1• =Y003	D2• =Y007

Explanation of function and operation



Caution on writing during RUN

During RUN, avoid writing while the DSZR (FNC150) instruction is executed (that is, while a pulse is output).
Note that if writing is executed during RUN to a circuit block including the FNC150 instruction while pulses are output, the PLC decelerates and stops pulse output.

Function change depending on the version

The function of FNC150 instruction is changed depending on the version as shown in the table below.

→ **For explanation of the instruction and the contents of function change, refer to the Positioning Control Edition.**

Applicable version		Item	Outline of function
FX3U	FX3UC		
Ver.2.20 or later	Ver.2.20 or later	Clear signal output destination specification function	When a special auxiliary relay corresponding to D1• is set to ON, the clear signal output destination is changed to an output number specified by a special data register corresponding to D• .

20.2 FNC151 – DVIT / Interrupt Positioning

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes one-speed interrupt constant quantity feed.

→ **For explanation of the instruction, refer to the Positioning Control Edition.**
→ **For cautions on using special high speed output adapters, refer to the Positioning Control Edition.**

1. Instruction format

	FNC 151	
D	DVIT	

16-bit Instruction	Mnemonic	Operation Condition
9 steps	DVIT	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DDVIT	Continuous Operation

2. Set data

Operand type	Description	Data type
S1•	Output pulse number (incremental address) after interrupt*1	16- or 32-bit binary
S2•	Output pulse frequency*2	
D1•	Device number outputting pulse	Bit
D2•	Device number to which rotation direction signal is output	

*1. Setting range: –32768 to +32767 (except 0) in 16-bit operation
–999,999 to +999,999 (except 0) in 32-bit operation

*2. Setting range: 10 to 32767 Hz in 16-bit operation
Following range in 32-bit operation

Pulse output destination		Setting range
FX3U PLC	Special high speed output adapter	10 to 200,000 (Hz)
FX3U/FX3UC PLC	Main unit (transistor output)	10 to 100,000 (Hz)

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓			
D1•		▲1																	✓					
D2•		▲2	✓			✓	▲3												✓					

▲1 : Specify a transistor output Y000, Y001, or Y002 in the main unit or Y000, Y001, Y002*2, or Y003*2 in a special high speed output adapter*1.

*1. Special high speed output adapters cannot be connected to the FX3UC-32MT-LT.

*2. For specifying Y002 or Y003 in a special high speed output adapter, the second special high speed output adapter is required.

Points

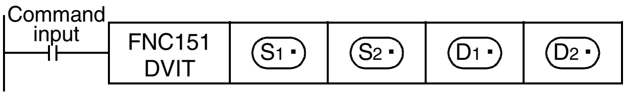
- When using a relay output type FX3U PLC, a special high speed output adapter is required.
- Outputs of special high speed output adapters work as differential line drivers.

▲2 : When using a special high speed output adapter for the pulse output destination in an FX3U PLC, the rotation direction signal must be used by the following table output.
When using a built-in transistor output for the pulse output destination in an FX3U/FX3UC PLC, the rotation direction signal must use transistor output.

Special high speed output adapter No.	Pulse output	Rotation direction output
No. 1 (1st unit)	D1• =Y000	D2• =Y004
	D1• =Y001	D2• =Y005
No. 2 (2nd unit)	D1• =Y002	D2• =Y006
	D1• =Y003	D2• =Y007

▲3 : "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation



Caution on writing during RUN

During RUN, avoid writing while the DVIT (FNC151) instruction is executed (that is, while a pulse is output).
Note that if writing is executed during RUN to a circuit block including the FNC151 instruction while pulses are output, the PLC decelerates and stops pulse output.

Function change depending on the version

The functions of FNC151 instruction are changed depending on the version as shown in the table below.

→ For explanation of the instruction and the contents of function change, refer to the Positioning Control Edition.

Applicable version		Item	Outline of function
FX3U	FX3UC		
Ver.2.20 or later	Ver.1.30 or later	Interrupt input signal specification function	When M8336 is set to ON, the interrupt input signal corresponding to Y000 to Y003 is changed to an input number (X000 to X007) specified by D8336. When using a transistor output in the main unit, Y003 cannot be specified.
Ver.2.20 or later	Ver.2.20 or later	User interrupt mode	When "8" is specified by D8336 to the interrupt input signal corresponding to Y000 to Y003 and M8336 is set to ON, the interrupt input signal is changed to a special auxiliary relay. When this changed special auxiliary relay is set to ON from OFF in an input interrupt program, the PLC starts the interrupt operation. When this function is used, however, the logic of the interrupt input cannot be inverted. In addition, when using a transistor output in the main unit, Y003 cannot be specified.

20.3 FNC152 – TBL / Batch Data Positioning Mode

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

This instruction executes one specified table operation set in the data table in advance in GX Developer (Ver.8.23Z or later).

→ **For explanation of the instruction, refer to the Positioning Control Edition.**
→ **For cautions on using special high speed output adapters, refer to the Positioning Control Edition.**

Instruction	Description
DVIT (FNC151)	Interrupt positioning
PLSV (FNC157)	Variable speed pulse output
DRVI (FNC158)	Drive to increment
DRVA (FNC159)	Drive to absolute

1. Instruction format

	FNC 152
D	TBL

16-bit Instruction	Mnemonic	Operation Condition
–		

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DTBL	Continuous Operation

2. Set data

Operand type	Description	Data type
(D)	Device number outputting pulse	Bit
n	Table number [1 to 100] to be executed	32-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(D)		▲1																						
n																				✓	✓			

▲1 : Specify a transistor output Y000, Y001, or Y002 in the main unit or Y000, Y001, Y002*2, or Y003*2 in a special high speed output adapter*1.

*1. Special high speed output adapters cannot be connected to the FX3UC-32MT-LT.

*2. For specifying Y002 or Y003 in a special high speed output adapter, the second special high speed output adapter is required.

Points

- When using a relay output type FX3U PLC, a special high speed output adapter is required.
- Outputs of special high speed output adapters work as differential line drivers.

Explanation of function and operation

Command input	FNC152 DTBL	(D·)	n

Caution on writing during RUN

Writing is disabled to a circuit block including TBL (FNC152) instruction during RUN.

20.4 FNC155 – ABS / Absolute Current Value Read

Outline

FX3UC Ver.1.00 ⇒ FX3U Ver.2.20 ⇒

This instruction reads the absolute position (ABS) data when the Mitsubishi servo amplifier (equipped with the absolute position detection function) MR-H, MR-J2(S), or MR-J3 is connected. The data is converted into a pulse when being read.

→ **For explanation of the instruction, refer to the Positioning Control Edition.**

1. Instruction format

D	FNC 155 ABS

16-bit Instruction	Mnemonic	Operation Condition
–		

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DABS	Continuous Operation

2. Set data

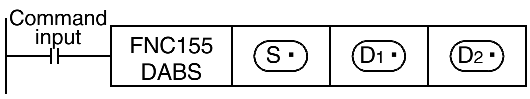
Operand type	Description	Data type
(S•)	Head device number inputting absolute (ABS) data output signal sent from servo amplifier Three points are occupied from (S•).	Bit
(D1•)	Head device number outputting absolute (ABS) data control signal to servo amplifier Three points are occupied from (D1•).	Bit
(D2•)	Device number storing absolute (ABS) data (32-bit value)	32-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S•)	✓	✓	✓			✓	▲2												✓					
(D1•)		▲1	✓			✓	▲2												✓					
(D2•)									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

▲1: Specify a transistor output.
▲2: "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation



11 FNC30-FNC39 Rotation and Shift
12 FNC40-FNC49 Data Operation
13 FNC50-FNC59 High Speed Processing
14 FMC60-FNC69 Handy Instruction
15 FNC70-FNC79 External FX I/O Device
16 FNC80-FNC89 External FX Device
17 FNC100-FNC109 Data Transfer 2
18 FNC110-FNC139 Floating Point
19 FNC140-FNC149 Data Operation 2
20 FNC150-FNC159 Positioning Control

20.5 FNC156 – ZRN / Zero Return

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes zero return, and aligns the mechanical position with a present value register inside the PLC.
When the dog search function is required, use DSZR (FNC150) instruction.

→ **For explanation of the instruction, refer to the Positioning Control Edition.**
→ **For cautions on using special high speed output adapters, refer to the Positioning Control Edition.**

1. Instruction format

D FNC 156 ZRN

16-bit Instruction	Mnemonic	Operation Condition
9 steps	ZRN	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DZRN	Continuous Operation

2. Set data

Operand type	Description	Data type
S1•	Speed when zero return is started*1	16- or 32-bit binary
S2•	Creep speed [10 to 32767 Hz]	
S3•	Device number inputting near-point signal (dog)	Bit
D•	Device number outputting pulse	

*1. Setting range: 10 to 32767 Hz in 16-bit operation
Following range in 32-bit operation

Pulse output destination		Setting range
FX3U PLC	Special high speed output adapter	10 to 200,000 (Hz)
FX3U/FX3UC PLC	Main unit (transistor output)	10 to 100,000 (Hz)

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S3•	✓	✓	✓			✓	▲1												✓					
D•		▲2																	✓					

▲1 : "D□.b" cannot be indexed with index registers (V and Z).
▲2 : Specify a transistor output Y000, Y001, or Y002 in the main unit or Y000, Y001, Y002*2, or Y003*2 in a special high speed output adapter*1.

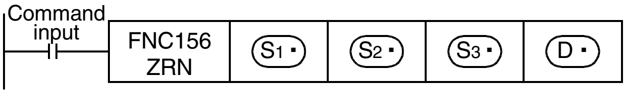
*1. Special high speed output adapters cannot be connected to the FX3UC-32MT-LT.

*2. For specifying Y002 or Y003 in a special high speed output adapter, the second special high speed output adapter is required.

Points

- When using a relay output type FX3U PLC, a special high speed output adapter is required.
- Outputs of special high speed output adapters work as differential line drivers.

Explanation of function and operation



Caution on writing during RUN

During RUN, avoid writing while the ZRN (FNC156) instruction is executed (that is, while pulses are output). Note that if writing is executed during RUN to a circuit block including the FNC156 instruction while pulses are output, the PLC decelerates and stops pulse output.

Function change depending on the version

The function of FNC156 instruction is changed depending on the version as shown in the table below.

→ **For explanation of the instruction and the contents of function change, refer to the Positioning Control Edition.**

Applicable version		Item	Outline of function
FX3U	**FX3UC**		
Ver.2.20 or later	Ver.2.20 or later	Clear signal output destination specification function	When a special auxiliary relay corresponding to D・ is set to ON, the clear signal output destination is changed to an output number specified by a special data register corresponding to D・.

20.6 FNC157 – PLSV / Variable Speed Pulse Output

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction outputs variable speed pulses with an assigned rotation direction.

→ For explanation of the instruction, refer to the Positioning Control Edition.

→ For cautions on using special high speed output adapters, refer to the Positioning Control Edition.

1. Instruction format

D FNC 157 PLSV

16-bit Instruction	Mnemonic	Operation Condition
9 steps	PLSV	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DPLSV	Continuous Operation

2. Set data

Operand type	Description	Data type
S1•	Device number specifying output pulse frequency*1	16- or 32-bit binary
D1•	Device number outputting pulse	Bit
D2•	Device number to which rotation direction signal is output	

*1. Setting range: –32768 to –1 , +1 to +32767 (except 0) Hz in 16-bit operation
Following range in 32-bit operation

Pulse output destination		Setting range
FX3U PLC	Special high speed output adapter	–200,000 to –1, +1 to 200,000 (Hz)
FX3U/FX3UC PLC	Main unit (transistor output)	–100,000 to –1, +1 to 100,000 (Hz)

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D1•		▲1																	✓					
D2•		▲2	✓			✓	▲3												✓					

▲1 : Specify a transistor output Y000, Y001, or Y002 in the main unit or Y000, Y001, Y002*2, or Y003*2 in a special high speed output adapter*1.

*1. Special high speed output adapters cannot be connected to the FX3UC-32MT-LT.

*2. For specifying Y002 or Y003 in a special high speed output adapter, the second special high speed output adapter is required.

Points

- When using a relay output type FX3U PLC, a special high speed output adapter is required.
- Outputs of special high speed output adapters work as differential line drivers.

▲2 : When using a special high speed output adapter for the pulse output destination in an FX3U PLC, the rotation direction signal must be used by the following table output.
When using a built-in transistor output for the pulse output destination in an FX3U/FX3UC PLC, the rotation direction signal must use transistor output.

Special high speed output adapter No.	Pulse output	Rotation direction output
No. 1 (1st unit)	D1• =Y000	D2• =Y004
	D1• =Y001	D2• =Y005
No. 2 (2nd unit)	D1• =Y002	D2• =Y006
	D1• =Y003	D2• =Y007

▲3 : "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

Command input	FNC157 PLSV	S•	D1•	D2•

Caution on writing during RUN

During RUN, avoid writing while PLSV (FNC157) instruction is executed (that is, while pulses are output). Note that if writing is executed during RUN to a circuit block including FNC157 instruction while pulses are output, the PLC executes the operation shown below.

	PLC operation when writing is excuted during RUN while instruction is executed
During operation with acceleration/deceleration*1	Decelerates and stops pulse output.
During operation without acceleration/deceleration	Immediately stops pulse output.

*1. Supported in Ver.2.20 or later

Function change depending on the version

The function of the FNC157 instruction is changed depending on the version as shown in the table below.

→ **For explanation of the instruction and the contents of function change, refer to the Positioning Control Edition.**

Applicable version		Item	Outline of function
FX3U	FX3UC		
Ver.2.20 or later	Ver.2.20 or later	Acceleration/ deceleration operation function	When M8338 is set to ON, the PLC accelerates or decelerates up to S1• in the acceleration or deceleration time corresponding to D1• if S1• changes.

20.7 FNC158 – DRVI / Drive to Increment

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes one-speed positioning by incremental drive. The movement distance from the present position can be specified with the positive or negative sign.

→ For explanation of the instruction, refer to the Positioning Control Edition.
→ For cautions on using special high speed output adapters, refer to the Positioning Control Edition.

1. Instruction format

	FNC 158
D	DRVI

16-bit Instruction	Mnemonic	Operation Condition
9 steps	DRVI	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DDRVI	Continuous Operation

2. Set data

Operand type	Description	Data type
S1•	Output pulse number (relative address)*1	16- or 32-bit binary
S2•	Output pulse frequency*2	
D1•	Device number outputting pulse	Bit
D2•	Device number to which rotation direction signal is output	

*1. Setting range: –32768 to +32767 (except 0) in 16-bit operation
–999,999 to +999,999 (except 0) in 32-bit operation

*2. Setting range: 10 to 32767 Hz in 16-bit operation
Following range in 32-bit operation

Pulse output destination		Setting range
FX3U PLC	Special high speed output adapter	10 to 200,000 (Hz)
FX3U/FX3UC PLC	Main unit (transistor output)	10 to 100,000 (Hz)

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D1•		▲1																	✓					
D2•		▲2	✓			✓	▲3												✓					

▲1 : Specify a transistor output Y000, Y001, or Y002 in the main unit or Y000, Y001, Y002*2, or Y003*2 in a special high speed output adapter*1.

*1. Special high speed output adapters cannot be connected to the FX3UC-32MT-LT.

*2. For specifying Y002 or Y003 in a special high speed output adapter, the second special high speed output adapter is required.

Points

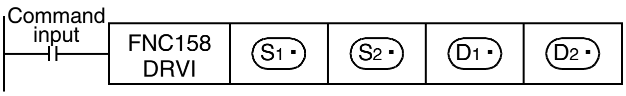
- When using a relay output type FX3U PLC, a special high speed output adapter is required.
- Outputs of special high speed output adapters work as differential line drivers.

▲2 :When using a special high speed output adapter for the pulse output destination in an FX3U PLC, the rotation direction signal must be used by the following table output.
When using a built-in transistor output for the pulse output destination in an FX3U/FX3UC PLC, the rotation direction signal must use transistor output.

Special high speed output adapter No.	Pulse output	Rotation direction output
No. 1 (1st unit)	D1• =Y000	D2• =Y004
	D1• =Y001	D2• =Y005
No. 2 (2nd unit)	D1• =Y002	D2• =Y006
	D1• =Y003	D2• =Y007

▲3 : "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation



Caution on writing during RUN

During RUN, avoid writing while DRVI (FNC158) instruction is executed (that is, while pulses are output). Note that if writing is executed during RUN to a circuit block including FNC158 instruction while pulses are output, the PLC decelerates and stops pulse output.

20.8 FNC159 – DRVA / Drive to Absolute

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes one-speed positioning by absolute drive. The movement distance from the zero point can be specified.

→ **For explanation of the instruction, refer to the Positioning Control Edition.**
→ **For cautions on using special high speed output adapters, refer to the Positioning Control Edition.**

1. Instruction format

FNC 159 D DRVA

16-bit Instruction	Mnemonic	Operation Condition
9 steps	DRVA	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DDRVA	Continuous Operation

2. Set data

Operand type	Description	Data type
S1•	Output pulse number (absolute address)*1	16- or 32-bit binary
S2•	Output pulse frequency*2	
D1•	Device number outputting pulse	Bit
D2•	Device number to which rotation direction signal is output	

*1. Setting range: –32768 to +32767 in 16-bit operation
–999,999 to +999,999 in 32-bit operation

*2. Setting range: 10 to 32767 Hz in 16-bit operation
Following range in 32-bit operation

Pulse output destination		Setting range
FX3U PLC	Special high speed output adapter	10 to 200,000 (Hz)
FX3U/FX3UC PLC	Main unit (transistor output)	10 to 100,000 (Hz)

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D1•		▲1																	✓					
D2•		▲2	✓			✓	▲3												✓					

▲1 : Specify a transistor output Y000, Y001, or Y002 in the main unit or Y000, Y001, Y002*2, or Y003*2 in a special high speed output adapter*1.

*1. Special high speed output adapters cannot be connected to the FX3UC-32MT-LT.

*2. For specifying Y002 or Y003 in a special high speed output adapter, the second special high speed output adapter is required.

Points

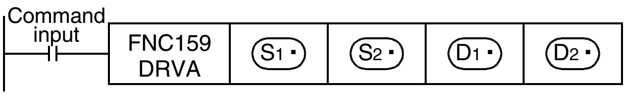
- When using a relay output type FX3U PLC, a special high speed output adapter is required.
- Outputs of special high speed output adapters work as differential line drivers.

▲2 : When using a special high speed output adapter for the pulse output destination in an FX3U PLC, the rotation direction signal must be used by the following table output.
When using a built-in transistor output for the pulse output destination in an FX3U/FX3UC PLC, the rotation direction signal must use transistor output.

Special high speed output adapter No.	Pulse output	Rotation direction output
No. 1 (1st unit)	D1• =Y000	D2• =Y004
	D1• =Y001	D2• =Y005
No. 2 (2nd unit)	D1• =Y002	D2• =Y006
	D1• =Y003	D2• =Y007

▲3 : "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation



Caution on writing during RUN

During RUN, avoid writing while DRVA (FNC159) instruction is executed (that is, while pulses are output).
Note that if writing is executed during RUN to a circuit block including FNC159 instruction while pulses are output, the PLC decelerates and stops pulse output.

21. Real Time Clock Control – FNC160 to FNC169

FNC160 to FNC169 provide operation and comparison instructions for the time data.
These instructions can set the time of the real time clock built in a PLC, and converts the format of the time data.

FNC No.	Mnemonic	Symbol	Function	Reference
160	TCMP	TCMP S1 S2 S3 S D	RTC data compare	Section 21.1
161	TZCP	TZCP S1 S2 S D	RTC data zone compare	Section 21.2
162	TADD	TADD S1 S2 D	RTC data addition	Section 21.3
163	TSUB	TSUB S1 S2 D	RTC data subtraction	Section 21.4
164	HTOS	HTOS S D	Hour to second conversion	Section 21.5
165	STOH	STOH S D	Second to hour conversion	Section 21.6
166	TRD	TRD D	Read RTC data	Section 21.7
167	TWR	TWR S	Set RTC data	Section 21.8
168	–			–
169	HOUR	HOUR S D1 D2	Hour Meter	Section 21.9

21.1 FNC160 – TCMP / RTC Data Compare

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction compares the comparison time with the time data, and turns ON or OFF bit devices according to the comparison result.

1. Instruction format

FNC 160
TCMP P

16-bit Instruction	Mnemonic	Operation Condition
11 steps	TCMP	Continuous Operation
	TCMPP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
S1•	Specifies "hour" of the comparison time [setting range: 0 to 23].	16-bit binary
S2•	Specifies "minute" of the comparison time [setting range: 0 to 59].	16-bit binary
S3•	Specifies "second" of the comparison time [setting range: 0 to 59].	16-bit binary
S•	Specifies "hour" of the time data (hour, minute, and second). (Three devices are occupied.)	16-bit binary
D•	Turns ON or OFF according to the comparison result. (Three devices are occupied.)	Bit

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others						
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S3•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S•												✓	✓	✓	✓	✓			✓					
D•		✓	✓			✓	▲												✓					

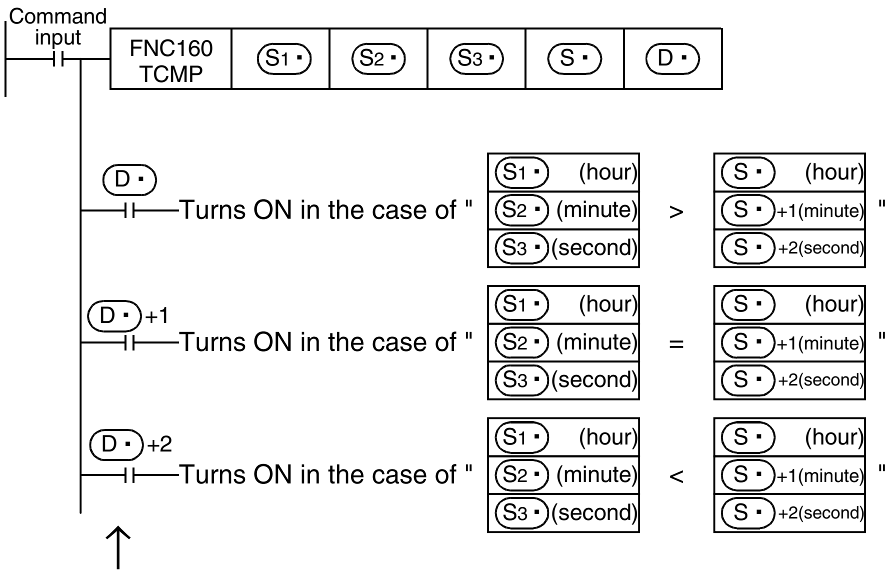
▲: "D□.b" cannot be indexed with index registers (V and Z).

21 FNC160-FNC169 Real Time Clock Control
22 FNC170-FNC179 External Device
23 FNC180 Alternate Instructions
24 FNC181-FNC189 Others
25 FNC190-FNC199 Block Data Operation
26 FNC200-FNC209 Character String Control
27 FNC210-FNC219 Data Operation 3
28 FNC220-FNC249 Data Comparison
29 FNC250-FNC269 Data Table Operation
30 FNC270-FNC274 Ex-Device Inverter Comms

Explanation of function and operation

1. 16-bit operation (TCMP)

The comparison time (hour, minute, and second) stored in S1·, S2·, and S3· is compared with the time data (hour, minute, and second) stored in S·, S·+1, and S·+2. Three devices starting from D· turn ON or OFF according to the comparison result.

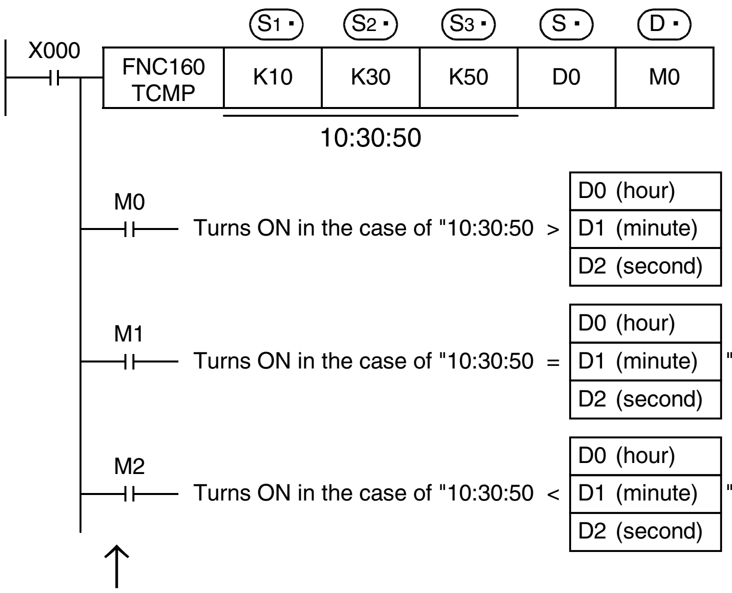


Even if the command contact turns OFF from ON and TCMP instruction is not executed, D·, D·+1 and D·+2 hold the status before the command contact turned OFF.

Cautions

1) Number of occupied devices
   Three devices are occupied respectively by S· and D·.
   Make sure that these devices are not used in other controls for the machine.
2) When utilizing the time (hour, minute, and second) of the real time clock built in a PLC
   Read the values of special data registers by TRD (FNC166) instruction, and then specify those word devices as the operands.

Program example



Even if X000 is OFF and TCMP instruction is not executed, M0 to M2 hold the status before X000 turned OFF.

S1· : Specifies "hour" of the comparison time.
S2· : Specifies "minute" of the comparison time.
S3· : Specifies "second" of the comparison time.
S· : Specifies "hour" of the time data.
S·+1 : Specifies "minute" of the time data.
S·+2 : Specifies "second" of the time data.
D·, D·+1, D·+2
: Three bit devices turn ON or OFF according to the comparison result.

21.2 FNC161 – TZCP / RTC Data Zone Compare

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction compares two comparison time (comparison time zone) with the time data, and turns ON or OFF the specified bit devices according to the comparison results.

1. Instruction format

FNC 161
TZCP P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	TZCP	Continuous Operation
	TZCPP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
S1•	Specifies "hour" of the lower limit comparison time (hour, minute, and second). (Three devices are occupied.)	16-bit binary
S2•	Specifies "hour" of the upper limit comparison time (hour, minute, and second). (Three devices are occupied.)	16-bit binary
S•	Specifies "hour" of the time data (hour, minute, and second). (Three devices are occupied.)	16-bit binary
D•	Turns ON or OFF according to the comparison result. (Three devices are occupied.)	Bit

3. Applicable devices

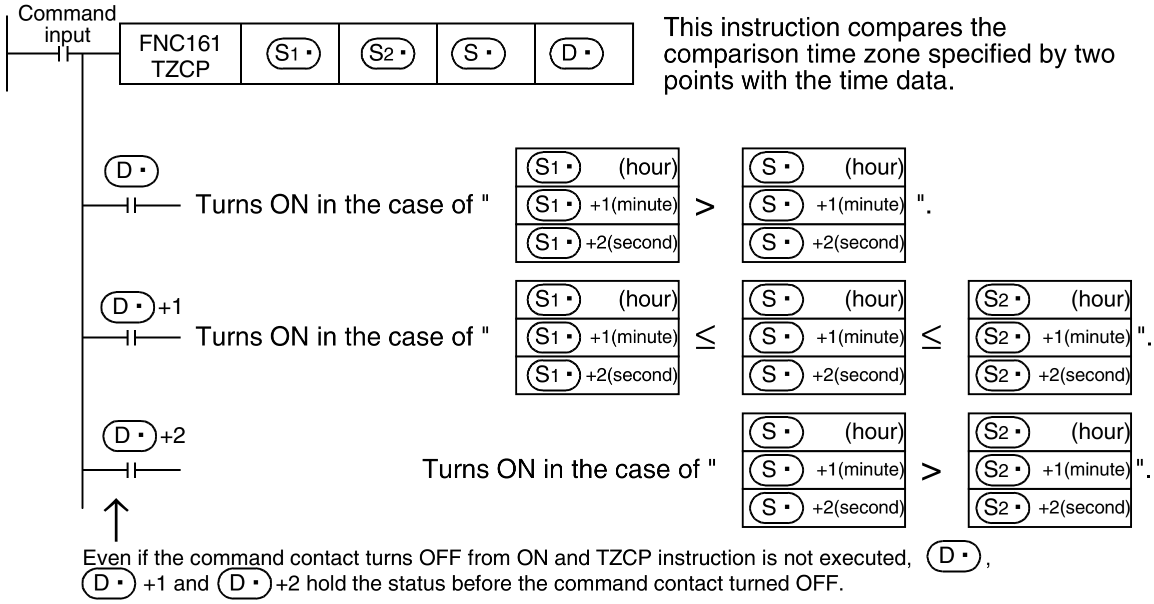
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•												✓	✓	✓	✓	✓			✓					
S2•												✓	✓	✓	✓	✓			✓					
S•												✓	✓	✓	✓	✓			✓					
D•		✓	✓			✓	▲												✓					

▲: "D□.b" cannot be indexed with index registers (V and Z).

Explaination of function and operation

1. 16-bit operation (TZCP)

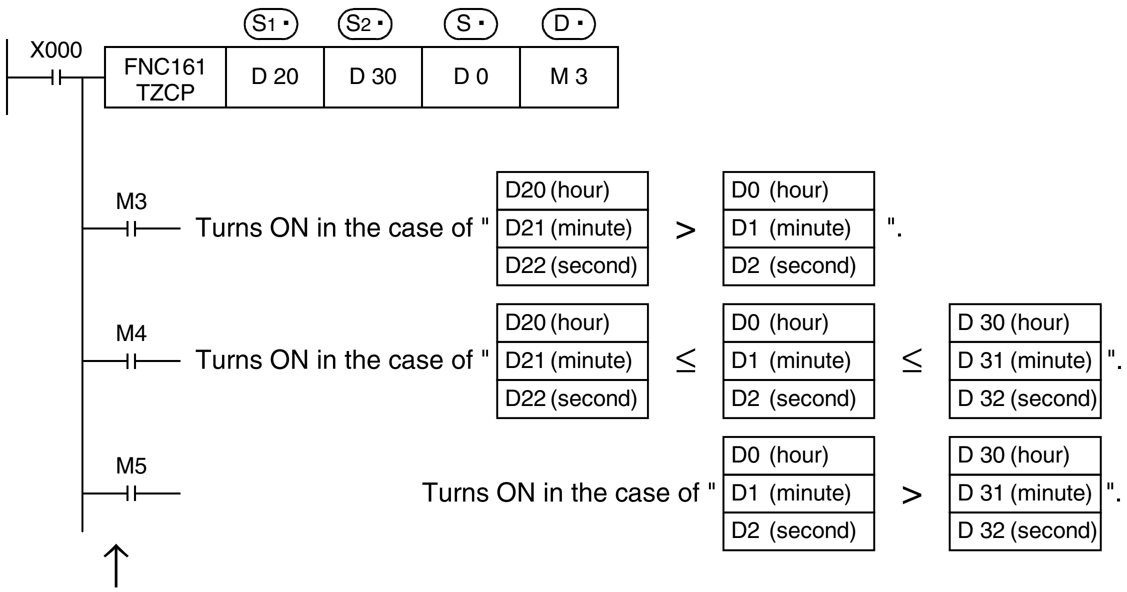
The lower limit and upper limit comparison time (hour, minute, and second) are compared with the time data (hour, minute, and second) stored in three devices S·, S·+1, and S·+2. Three devices starting from D· turn ON or OFF according to the comparison result.



Cautions

1) Number of occupied devices
   Three devices are occupied respectively by S1·, S2·, S3·, and D·.
   Make sure that these devices are not used in other controls for the machine.
2) When utilizing the time (hour, minute, and second) of the real time clock built in a PLC
   Read the values of special data registers by TRD (FNC166) instruction, and then specify those word devices as the operands.

Program example



Even if X000 is OFF and TZCP instruction is not executed, M0 to M2 hold the status before X000 turned OFF.

S1· , S1· +1 and S1· +2 : Specify the lower limit of the comparison time zone in "hour", "minute" and "second.

S2· , S2· +1 and S2· +2 : Specify the upper limit of the comparison time zone in "hour", "minute" and "second.

S· , S· +1 and S· +2 : Specify the time data in "hour", "minute" and "second.

D· , D· +1 and D· +2 : Turn ON or OFF according to the comparison result.

The setting range of "hour" is from 0 to 23.

The setting range of "minute" is from 0 to 59.

The setting range of "second" is from 0 to 59.

21.3 FNC162 – TADD / RTC Data Addition

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes addition of two time data, and stores the addition result to word devices.

1. Instruction format

FNC 162 TADD P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	TADD	Continuous Operation
	TADDP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
S1•	Specifies "hour" of the time data (hour, minute, and second) used in addition. (Three devices are occupied.)	16-bit binary
S2•	Specifies "hour" of the time data (hour, minute, and second) used in addition. (Three devices are occupied.)	16-bit binary
D•	Stores the addition result (hour, minute, and second) of two time data. (Three devices are occupied.)	16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•												✓	✓	✓	✓	✓			✓					
S2•												✓	✓	✓	✓	✓			✓					
D•												✓	✓	✓	✓	✓			✓					

Explanation of function and operation

1. 16-bit operation (TADD)

The time data (hour, minute, and second) stored in S2•, S2•+1, and S2•+2 is added to the time data (hour, minute, and second) stored in S1•, S1•+1, and S1•+2, and the addition result (hour, minute, and second) is stored in D•, D•+1, and D•+2.

Command input — FNC162 TADD | S1• | S2• | D•

(S1•, S1•+1, S1•+2)+(S2•, S2•+1, S2•+2)
→(D•, D•+1, D•+2)

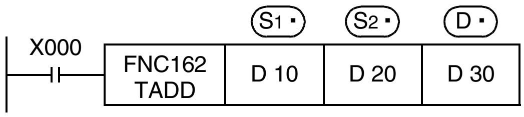
S1• (hour)	+	S2• (hour)	→	D• (hour)	The setting range of "hour" is from 0 to 23.
S1• +1(minute)		S2• +1(minute)		D• +1(minute)	The setting range of "minute" is from 0 to 59.
S1• +2(second)		S2• +2(second)		D• +2(second)	The setting range of "second" is from 0 to 59.

- When the operation result exceeds 24 hours, the carry flag turns ON, and the value simply acquired by addition subtracted by 24 hours is stored as the operation result.
- When the operation result becomes "0" (0:0:0), the zero flag turns ON.

Cautions

1) Number of occupied devices
   Three devices are occupied by (S1·), (S2·) and (D·) respectively.
   Make sure that these devices are not used in other controls for the machine.
2) When utilizing the time (hour, minute, and second) of the real time clock built in a PLC
   Read the values of special data registers using the TRD (FNC166) instruction, and then specify those word devices as the operands.

Program example



(D10 ,D11 ,D12)+(D20 ,D21 ,D22)
→ (D30 ,D31,D32)

(S1·)		+	(S2·)		→	(D·)	
(S1·)	10 (hour)		(S2·)	3 (hour)		(D·)	13 (hour)
(S1·) +1	30 (minute)		(S2·) +1	10 (minute)		(D·) +1	40 (minute)
(S1·) +2	10 (second)		(S2·) +2	5 (second)		(D·) +2	15 (second)

10:30:10 3:10:5 13:40:15

When the operation result exceeds 24 hours

(S1·)	+	(S2·)	→	(D·)	
18 (hour)		10 (hour)		4 (hour)	← 18+10=28(≥24)
10 (minute)		20 (minute)		30 (minute)	
30 (second)		5 (second)		35 (second)	

18:10:30 10:20:5 4:30:35

21.4 FNC163 – TSUB / RTC Data Subtraction

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes subtraction of two time data, and stores the subtraction result to word devices.

1. Instruction format

FNC 163 TSUB P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	TSUB	Continuous Operation
	TSUBP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	
	–	

2. Set data

Operand type	Description	Data type
S1•	Specifies "hour" of the time data (hour, minute, and second) used in subtraction. (Three devices are occupied.)	16-bit binary
S2•	Specifies "hour" of the time data (hour, minute, and second) used in subtraction. (Three devices are occupied.)	16-bit binary
D•	Stores the subtraction result (hour, minute, and second) of two time data. (Three devices are occupied.)	16-bit binary

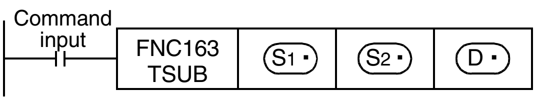
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•												✓	✓	✓	✓	✓			✓					
S2•												✓	✓	✓	✓	✓			✓					
D•												✓	✓	✓	✓	✓			✓					

Explanation of function and operation

1. 16-bit operation (TSUB)

The time data (hour, minute, and second) stored in S2•, S2•+1, and S2•+2 is subtracted from the time data (hour, minute, and second) stored in S1•, S1•+1, and S1•+2, and the subtraction result (hour, minute, and second) is stored in D•, D•+1, and D•+2.



(S1•, S1•+1, S1•+2)–(S2•, S2•+1, S2•+2)
→ (D•, D•+1, D•+2)

S1•		S2•		D•	
S1• (hour)	–	S2• (hour)	→	D• (hour)	The setting range of "hour" is from 0 to 23.
S1• +1 (minute)		S2• +1 (minute)		D• +1 (minute)	The setting range of "minute" is from 0 to 59.
S1• +2 (second)		S2• +2 (second)		D• +2 (second)	The setting range of "second" is from 0 to 59.

When the operation result is smaller than 0 hour, the borrow flag turns ON, and the value simply acquired by subtraction added by 24 hours is stored as the operation result.
When the operation result becomes "0" (0:0:0), the zero flag turns ON.

Cautions

1) Number of occupied devices
   Three devices are occupied by (S1·), (S2·) and (D·) respectively.
   Make sure that these devices are not used in other controls for the machine.
2) When utilizing the time (hour, minute, and second) of the real time clock built in a PLC
   Read the values of special data registers using TRD (FNC166) instruction, and then specify those word devices as the operands.

Program example

X000	FNC163 TSUB	(S1·) D 10	(S2·) D 20	(D·) D 30

(D10, D11, D12) – (D20, D21, D22)
→ (D30, D31, D32)

(S1·)		(S2·)		(D·)
D 10 10 (hour)		D 20 3 (hour)		D 30 7 (hour)
D 11 30 (minute)	–	D 21 10 (minute)	→	D 31 20 (minute)
D 12 10 (second)		D 22 5 (second)		D 32 5 (second)
10:30:10		3:10:5		7:20:5

When the operation result is smaller than "00:00:00"

(S1·)		(S2·)		(D·)	
5 (hour)		18 (hour)		11 (hour)	← 5–18=–13 (<0)
20 (minute)	–	10 (minute)	→	10 (minute)	
40 (second)		5 (second)		35 (second)	
5:20:40		18:10:5		11:10:35	

21.5 FNC164 – HTOS / Hour to Second Conversion

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction converts the time data in units of "hour, minute, and second" into data in units of "second".

1. Instruction format

D	FNC 164 HTOS	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	HTOS	Continuous Operation
	HTOSP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DHTOS	Continuous Operation
	DHTOSP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S·	Head device number storing the time data (hour, minute and second) before conversion	16-bit binary
D·	Device number storing the time data (second) after conversion	16- or 32-bit binary

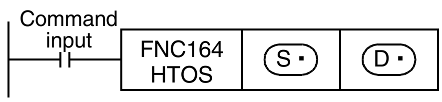
3. Applicable devices

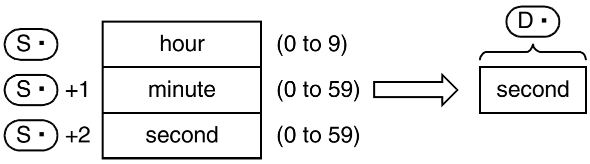
Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index				Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify		K	H	E	"□"	P
S·								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓						
D·									✓	✓	✓	✓	✓	✓	✓	✓			✓						

Explanation of function and operation

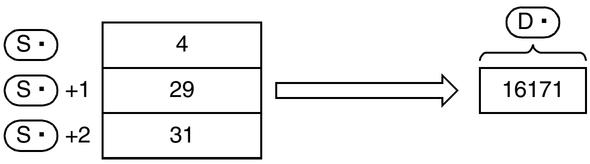
1. 16-bit operation (HTOS and HTOSP)

The time data (hour, minute, and second) stored in S·, S·+1, and S·+2 is converted into data in units of "second", and stored to D·.



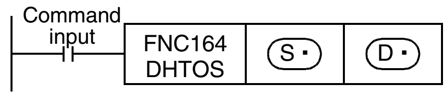


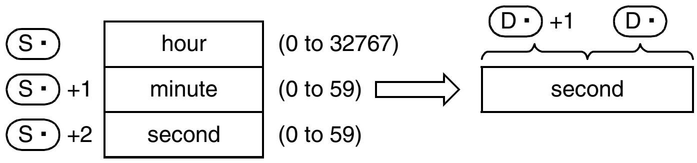
For example, when "4 hours 29 minutes 31 seconds" is specified, the operation is as follows:



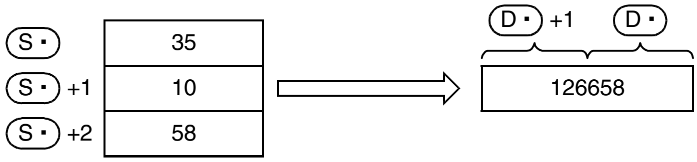
2. 32-bit operation (DHTOS and DHTOSP)

The time data (hour, minute, and second) stored in (S·), (S·)+1, and (S·)+2 is converted into data in units of "second", and stored to (D·)+1, (D·).





For example, when "35 hours 10 minutes 58 seconds" is specified, the operation is as follows:



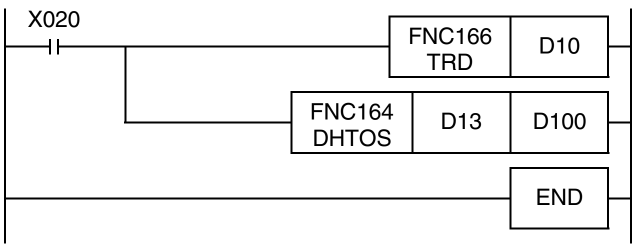
Error

An operation error is caused in the following case; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the data of (S·), (S·)+ 1 or (S·)+ 2 is outside the allowable range (error code: K6706)

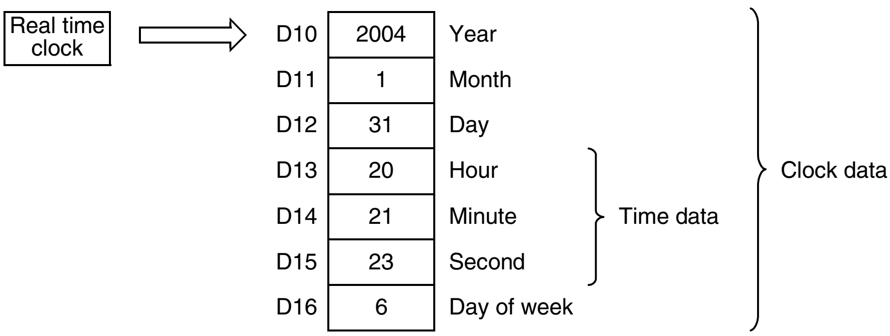
Program example

In the program shown below, the time data read from the real time clock built in a PLC is converted into data in units of "second", and stored to D100 and D101 when X020 turns ON.

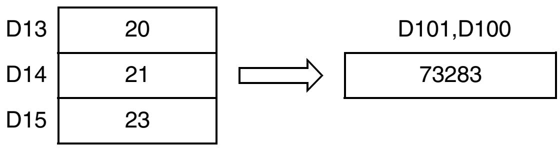


Operation

- Clock data reading operation by TRD (FNC166) instruction



- Conversion operation into "second" by DHTOS (FNC164) instruction



21.6 FNC165 – STOH / Second to Hour Conversion

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction converts the time data in units of "second" into data in units of "hour, minute, and second".

1. Instruction format

D	FNC 165 STOH	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	STOH	Continuous Operation
	STOHP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DSTOH	Continuous Operation
	DSTOHP	Pulse (Single) Operation

2. Set data

Operand type	Description	Data type
S·	Device number storing the time data (second) before conversion	16- or 32-bit binary
D·	Head device number storing the time data (hour, minute and second) after conversion	16-bit binary

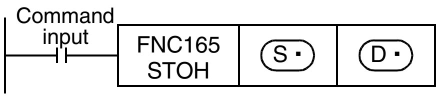
3. Applicable devices

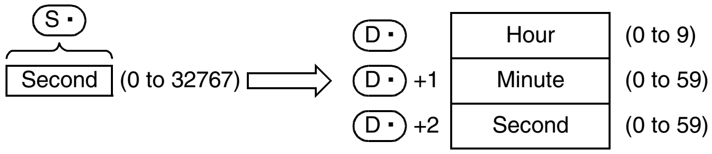
Operand Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index				Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify		K	H	E	"□"	P
S·								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓						
D·									✓	✓	✓	✓	✓	✓	✓	✓			✓						

Explanation of function and operation

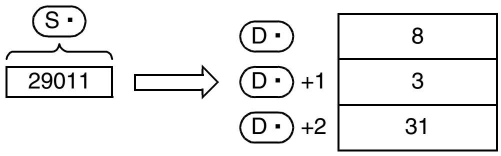
1. 16-bit operation (STOH and STOHP)

The time data in units of "second" stored in S· is converted into data in units of "hour, minute, and second", and stored to D·, D·+1, and D·+2 (hour, minute, and second).



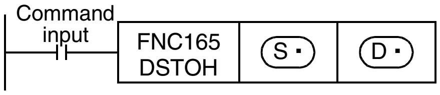


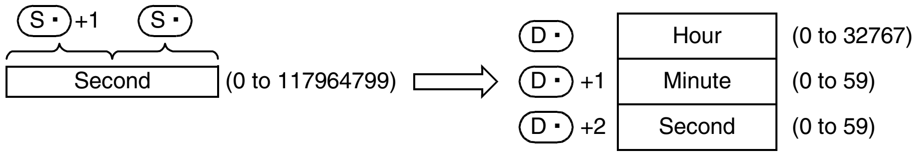
For example, when "29,011 seconds" is specified, the operation is as follows:



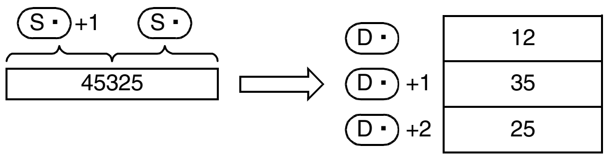
2. 32-bit operation (DSTOH and DSTOHP)

The time data in units of "second" stored in (S·)+1 and (S·) is converted into data in units of "hour, minute, and second", and stored to three devices (D·), (D·)+1, and (D·)+2 (hour, minute, and second).





For example, when "45,325 seconds" is specified, the operation is as follows:



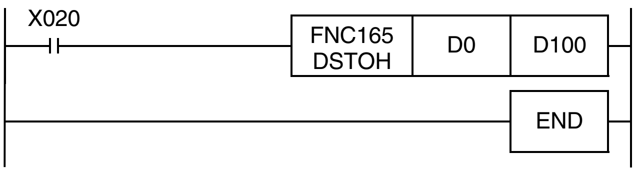
Error

An operation error is caused in the following case; The error flag M8067 turns ON, and the error code is store in D8067.

- When the data of (S·) is outside the allowable range (error code: K6706)

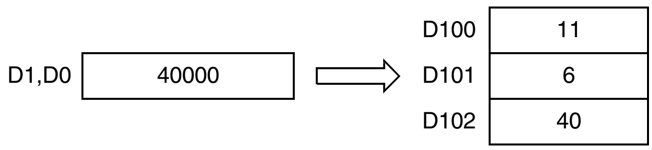
Program example

In the program shown below, the time data in units of "second" stored in D0 and D1 is converted into data in units of "hour, minute, and second", and stored to D100, D101, and D102 when X020 turns ON.



Operation

- Converting the data in second into the data in hour, minute and second using STOHP instruction (when "40,000 seconds" is specified by D1 and D0)



21.7 FNC166 – TRD / Read RTC data

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction reads the clock data of the real time clock built in a PLC.

1. Instruction format

FNC 166 TRD P

16-bit Instruction	Mnemonic	Operation Condition
3 steps	TRD	Continuous Operation
	TRDP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand type	Description	Data type
(D·)	Specifies the head device number storing the clock data. (Seven devices are occupied.)	16-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(D·)												✓	✓	✓	✓	✓			✓					

Explanation of function and operation

1. 16-bit operation (TRD)

The clock data stored in D8013 to D8019 of the real time clock built in a PLC is read in the following format, and stored to (D·) to (D·)+6.

Command input — FNC166 TRD (D·)

This instruction reads the real time clock data in a PLC, and transfers it to seven data registers.

	Device	Item	Clock data		Device	Item
Special data register	D8018	Year	0 to 99 (lower two digits)	→	D 0	Year
	D8017	Month	1 to 12	→	D 1	Month
	D8016	Day	1 to 31	→	D 2	Day
	D8015	Hour	0 to 23	→	D 3	Hour
	D8014	Minute	0 to 59	→	D 4	Minute
	D8013	Second	0 to 59	→	D 5	Second
	D8019	Day of week	0 (Sunday) to 6 (Saturday)	→	D 6	Day of week

Caution

1. Number of occupied devices

Seven devices are occupied by (D·).
Make sure that these devices are not used in other controls for the machine.

21.8 FNC167 – TWR / Set RTC data

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction writes the clock data to the real time clock built in a PLC.

1. Instruction format

FNC 167 TWR	P

16-bit Instruction	Mnemonic	Operation Condition
3 steps	TWR	Continuous Operation
	TWRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	

2. Set data

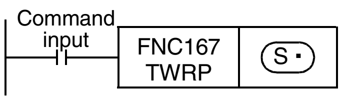
Operand type	Description	Data type
D·	Specifies the head device number to which the clock data is written. (Seven devices are occupied.)	16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
D·												✓	✓	✓	✓	✓			✓					

Explanation of function and operation

The clock data stored in S· to S· +6 is written to D8013 to D8019 for the real time clock built in a PLC.



21 FNC160-FNC169 Real Time Clock Control
22 FNC170-FNC179 External Device
23 FNC180 Alternate Instructions
24 FNC181-FNC189 Others
25 FNC190-FNC199 Block Data Operation
26 FNC200-FNC209 Character String Control
27 FNC210-FNC219 Data Operation 3
28 FNC220-FNC249 Data Comparison
29 FNC250-FNC269 Data Table Operation
30 FNC270-FNC274 Ex-Device Inverter Comms

- D8018 (year data) can be converted into the 4-digit mode. (Refer to the program example shown later.)

	Device	Item	Clock data		Device	Item	
Time data to be set	D 10	Year	0 to 99 (lower two digits)	→	D8018	Year	Special data register
	D 11	Month	1 to 12	→	D8017	Month	
	D 12	Day	1 to 31	→	D8016	Day	
	D 13	Hour	0 to 23	→	D8015	Hour	
	D 14	Minute	0 to 59	→	D8014	Minute	
	D 15	Second	o to 59	→	D8013	Second	
	D 16	Day of week	0 (Sunday) to 6 (Saturday)	→	D8019	Day of week	

- When TWR (FNC167) instruction is executed, the clock data of the real time clock is immediately changed. Accordingly, transfer the clock data several minutes ahead to S· to S·+6 in advance, and then execute FNC167 instruction when the accurate time has come.
- When setting the clock data (time) using this instruction, it is not necessary to control the special auxiliary relay M8015 (time stop and time setting).
- If a numeric value indicating impossible date/time is set, the clock data is not changed.
  Set the correct clock data, and then write it.

Caution

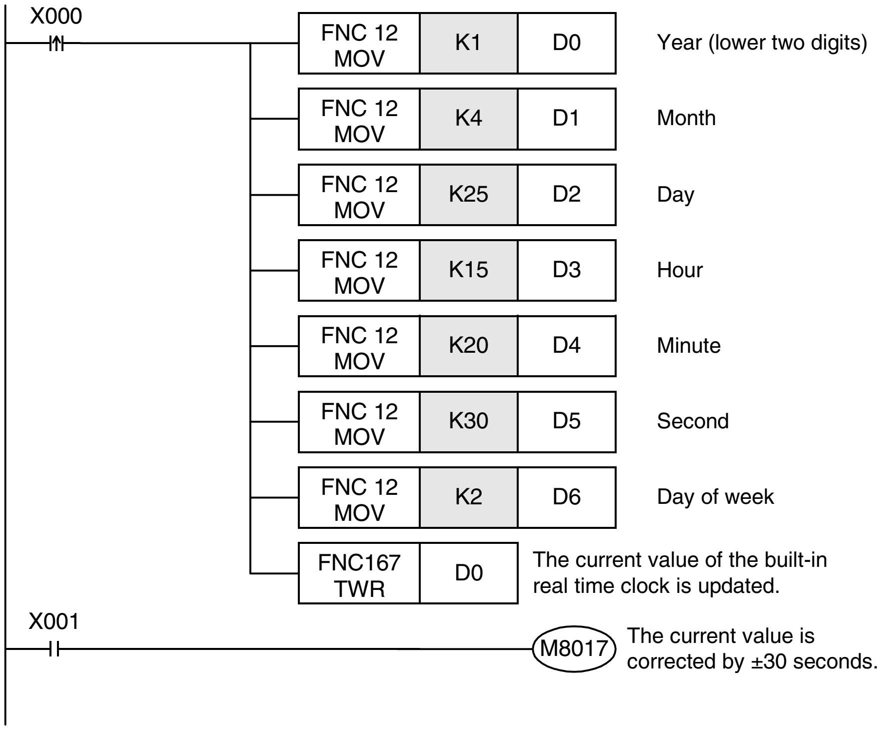
1. Number of occupied devices

Seven devices are occupied by S·.
Make sure that these devices are not used in other controls for the machine.

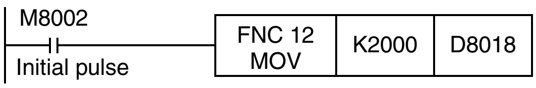
Program example

1. Example of setting the clock data (time)

In the program example shown below, the real time clock is set (to 15:20:30 on Tuesday, April 25, 2001).



- The shaded area indicates the set value of each item.
- When setting the time, it is recommended to set the time to several minutes ahead in advance, and then set X000 to ON when the accurate time is reached. The set time is then immediately written to the real time clock, and the clock data is updated.
- Every time X001 is set to ON, the current time can be corrected by ±30 seconds.
- When handling the year in the 4-digit mode, add the following program.
  D8018 will specify the 4-digit year mode in the second scan and later after the PLC mode is changed to RUN.



  - A PLC is normally operating in the 2-digit year mode. When the above instruction is executed and "K2000 (fixed value)" is transferred to D8018 (year) in only one operation cycle after the PLC mode was changed to RUN, the year mode is switched to the 4-digit mode.
  - Execute this program every time the PLC mode is changed to RUN. Even if "K2000" is transferred, only the display format is changed to the 4-digit year mode. The current date and time are not affected.
  - In the 4-digit year mode, the set values "80 to 99" correspond to "1980 to 1999", and "00 to 79" correspond to "2000 to 2079".
    Examples:
    "80" indicates 1980. "99" indicates 1999. "00" indicates 2000. "79" indicates 2079.
- When the data access unit FX-10DU-E is connected, select the 2-digit year mode. If the 4-digit year mode is selected, the year is not correctly displayed in the current version of the FX-10DU-E.

21.9 FNC169 – HOUR / Hour Meter

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction measures the ON time of the input contact in units of hour.

1. Instruction format

	FNC 169 HOUR
D	

16-bit Instruction	Mnemonic	Operation Condition
7 steps	HOUR	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DHOUR	Continuous Operation

2. Set data

Operand type	Description	Data type
(S•)	Time after which (D2•) is set to ON (unit: hour)	16- or 32-bit binary
(D1•)	Current value (unit: hour) (latched (battery backed) type data register latched (battery backed))	16- or 32-bit binary
(D2•)	Head device number to which alarm is output	16- or 32-bit binary

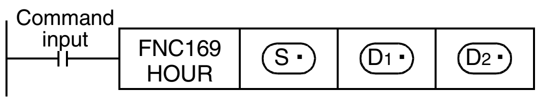
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S•)								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
(D1•)														✓	✓				✓					
(D2•)		✓	✓			✓	▲												✓					

▲: "D□.b" cannot be indexed with index registers (V and Z).

Explanation of function and operation

1. 16-bit operation



When the accumulated ON time of the command input exceeds the time stored in (S•), (D2•) is set to ON.
The current value less than one hour is stored in (D1•)+1 (unit: second).

(S•) : Time after which (D2•) is set to ON
Specify a value in units of hour.
(D1•) : Current value in units of hour
(D1•) + 1 : Current value less than one hour (unit: second)
(D2•) : Alarm output destination
It turns ON when the current value (D1•) exceeds the time specified in (S•).

- Specify a latched (battery backed) type data register as (D1•) so that the current value data can be continuously used even after the PLC turns OFF.
  If a general type data register is used, the current value data is cleared when the power of the PLC is turned OFF or when the PLC mode switches from STOP to RUN.
- Even after the alarm output (D2•) turns ON, the measurement is continued.
- When the current value (D1•) reaches the maximum value of 16-bit data, the measurement is stopped.
  For continuing the measurement, clear the current value stored in (D1•) and (D1•)+1.

2. 32-bit operation

Command input

FNC169 DHOUR	S·	D1·	D2·

[S· +1, S·]: Time after which D2· is set to ON
Specify the high-order side in S1· +1, and the low-order side in S1·.
[D1· +1, D1·]: Current value in units of hour
The high-order side is stored in D1· +1, and the low-order side is stored in D1·.
D1· +2 : Current value less than one hour (unit: second)
D2· : Alarm output destination
It turns ON when the current value [D1· +1, D1·] exceeds the time specified in S·.

- Specify a latched (battery backed) type data register as D1· so that the current value data can be continuously used even after the PLC turns OFF.
  If a general data type register is used, the current value data is cleared when the power of the PLC is turned OFF or when the PLC mode switches from STOP to RUN.
- Even after the alarm output D2· turns ON, the measurement is continued.
- When the current value [D1· +1, D1·] reaches the maximum value of 32-bit data, the measurement is stopped.
  For continuing the measurement, clear the current value stored in D1· to D1· +2.

Caution

Number of occupied devices

Two (16-bit operation) or three (32-bit operation) devices are occupied by D1·.
Make sure that these devices are not used in other controls for the machine.

Program example

In the program example shown below, when the accumulated X000 ON time exceeds 300 hours, Y005 turns ON.
The current value less than one hour is stored in D201 in units of second.

X000

	S·	D1·	D2·
FNC169 HOUR	K300	D 200	Y005

S· : Time after which D2· is set to ON
Specify a value in units of hour.
D1· : Current value in units of hour
D1· +1 : Current value less than one hour (unit: second)
D2· : Alarm output destination
It turns ON when the current value D1· exceeds the time specified in S·.
(In this example, it turns ON when the current value becomes 300 hours 1 second.)

22. External Device – FNC170 to FNC179

FNC170 to FNC179 provide conversion instructions for gray codes used in absolute type rotary encoders and instructions dedicated to analog blocks.

FNC No.	Mnemonic	Symbol	Function	Reference
170	GRY	GRY S D	Decimal to Gray Code Conversion	Section 22.1
171	GBIN	GBIN S D	Gray Code to Decimal Conversion	Section 22.2
172	–			–
173	–			–
174	–			–
175	–			–
176	RD3A	RD3A m1 m2 D	Read form Dedicated Analog Block	Section 22.3
177	WR3A	WR3A m1 m2 S	Write to Dedicated Analog Block	Section 22.4
178	–			–
179	–			–

22.1 FNC170 – GRY / Decimal to Gray Code Conversion

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction converts a binary value into a gray code, and transfers it.

1. Instruction format

D	FNC 170 GRY	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	GRY	Continuous Operation
	GRYP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DGRY	Continuous Operation
	DGRYP	Pulse (Single) Operation

2. Set data

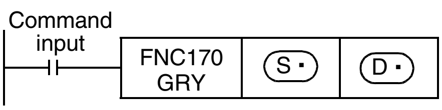
Operand Type	Description	Data Type
(S·)	Conversion source data or word device storing conversion source data	16- or 32-bit binary
(D·)	Word device storing data after conversion	16- or 32-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
(D·)									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (GRY and GRYP)



This instruction converts and transfers data from the source (binary) to the destination (gray code).

When (S·) is K1234 and (D·) is K3Y10

	b15															b0
BIN 1234	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0

⇩

	Y23			Y20	Y17							Y10
GRY 1234	0	1	1	0	1	0	1	1	1	0	1	1

- (S·) can store a value from 0 to 32767.

2. 32-bit operation (DGRY and DGRYP)

- A binary value can be converted into a gray code of up to 32 bits.
- (S·) can store a value from 0 to 2,147,483,647.

Caution

The data conversion speed depends on the scan time of the PLC.

21 FNC160-FNC169 Real Time Clock Control
22 FNC170-FNC179 External Device
23 FNC180 Alternate Instructions
24 FNC181-FNC189 Others
25 FNC190-FNC199 Block Data Operation
26 FNC200-FNC209 Character String Control
27 FNC210-FNC219 Data Operation 3
28 FNC220-FNC249 Data Comparison
29 FNC250-FNC269 Data Table Operation
30 FNC270-FNC274 Ex-Device Inverter Comms

22.2 FNC171 – GBIN / Gray Code to Decimal Conversion

FX3UC Ver.1.00 ⇒ FX3U Ver.2.20 ⇒

Outline

This instruction converts a gray code into a binary value, and transfers it.

1. Instruction format

	FNC 171	
D	GBIN	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	GBIN	Continuous Operation
	GBINP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DGBIN	Continuous Operation
	DGBINP	Pulse (Single) Operation

2. Set data

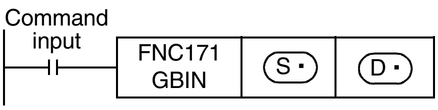
Operand Type	Description	Data Type
(S·)	Conversion source data or word device storing conversion source data	16- or 32-bit binary
(D·)	Word device storing data after conversion	16- or 32-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
(D·)									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (GBIN and GBINP)



This instruction converts and transfers data from the source (gray code) to the destination (binary).

When (S·) is K3X000 and (D·) is D10

GRY 1234

X13			X10	X7							X0
0	1	1	0	1	0	1	1	1	0	1	1

⇩

D10 BIN 1234

b15															b0
0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0

- This instruction can be used for detecting an absolute position by a gray code type encoder.
- (S·) can store a value from 0 to 32,767.

2. 32-bit operation (DGBIN and DGBINP)

- A gray code can be converted into a binary value of up to 32 bits.
- (S·) can store a value from 0 to 2,147,483,647.

Caution

When an input relay (X) is specified as (S·), the response relay will be "Scan time of PLC + Input filter constant".
The filter constant can be changed in X000 to X017 using REFF (FNC 51) instruction or D8020 (filter adjustment) so that the filter constant delay is eliminated.

22.3 FNC176 – RD3A / Read form Dedicated Analog Block

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction reads an analog input value from the analog block FX0N-3A or FX2N-2AD.

1. Instruction format

FNC 176 RD3A	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	RD3A	Continuous Operation
	RD3AP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	

2. Set data

Operand Type	Description	Data Type
m1 •	Special block number - FX3U PLC : K0 to K7 - FX3UC PLC : K1 to K7	16-bit binary
m2 •	Analog input channel number	16-bit binary
D•	Word device storing the read data	16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
m1 •								✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓			
m2 •								✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓			
D•									✓	✓	✓	✓	✓	✓	✓		✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (RD3A)

Command input

FNC176 RD3A	m1•	m2•	D•

m1 • : Special block number
FX3U PLC : K0 to K7
FX3UC PLC: K1 to K7 (K0 indicates the built-in CC-Link/LT master.)

m2 • : Analog input channel number
FX0N-3A : K1 (ch 1) or K2 (ch 2)
FX2N-2AD : K21 (ch 1) or K22 (ch 2)

D• : Read data
A value read from the analog block is stored.
FX0N-3A : 0 to 255 (8 bits)
FX2N-2AD : 0 to 4095 (12 bits)

22.4 FNC177 – WR3A / Write to Dedicated Analog Block

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction writes a digital value to the analog block FX0N-3A or FX2N-2DA.

1. Instruction format

FNC 177 WR3A P	16-bit Instruction	Mnemonic	Operation Condition	32-bit Instruction	Mnemonic	Operation Condition
	7 steps	WR3A	Continuous Operation	–		
		WR3AP	Pulse (Single) Operation			

2. Set data

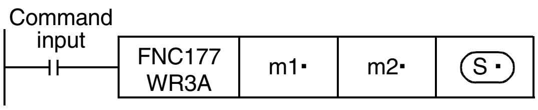
Operand Type	Description	Data Type
m1 •	Special block number - FX3U PLC : K0 to K7 - FX3UC PLC : K1 to K7	16- or 32-bit binary
m2 •	Analog output channel number	16- or 32-bit binary
S•	Data to be written or word device storing data to be written	16- or 32-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
m1 •								✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓			
m2 •								✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓			
S•									✓	✓	✓	✓	✓	✓	✓		✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (WR3A)



m1 • : Special block number
FX3U PLC : K0 to K7
FX3UC PLC: K1 to K7 (K0 indicates the built-in CC-Link/LT master.)

m2 • : Analog input channel number
FX0N-3A : K1 (ch 1)
FX2N-2DA : K21 (ch 1) or K22 (ch 2)

S• : Data to be written
Specify a value output to the analog block.
FX0N-3A : 0 to 255 (8 bits)
FX2N-2DA : 0 to 4095 (12 bits)

21 FNC160-FNC169 Real Time Clock Control
22 FNC170-FNC179 External Device
23 FNC180 Alternate Instructions
24 FNC181-FNC189 Others
25 FNC190-FNC199 Block Data Operation
26 FNC200-FNC209 Character String Control
27 FNC210-FNC219 Data Operation 3
28 FNC220-FNC249 Data Comparison
29 FNC250-FNC269 Data Table Operation
30 FNC270-FNC274 Ex-Device Inverter Comms

23. Introduction of Alternate Instructions – FNC180

23.1 Instruction correspondence table

Outline

EXTR instruction is provided for FX2N and FX2NC PLCs.
For FX3U and FX3UC PLCs with a built-in inverter communication function, dedicated instructions shown below are provided. (EXTR instruction is not provided.)

Instruction correspondence table

FX2N/FX2NC		FX3U/FX3UC		Description
EXTR K10	→	FNC270	IVCK	Inverter status check
EXTR K11	→	FNC271	IVDR	Inverter drive
EXTR K12	→	FNC272	IVRD	Inverter parameter read
EXTR K13	→	FNC273	IVWR	Inverter parameter write
–		FNC274	IVBWR	Inverter parameter block write

→ **For details, refer to the Communication Control Manual.**

MEMO

24. Others – FNC181 to FNC189

FNC181 to FNC189 provide instructions for generating random numbers, executing CRC data operations, and processing data in high speed counter operations.

FNC No.	Mnemonic	Symbol	Function	Reference
181	–			–
182	COMRD	COMRD S D	Read device comment data	Section 24.1
183	–			–
184	RND	RND D	Random Number Generation	Section 24.2
185	–			–
186	DUTY	DUTY n1 n2 D	Timing pulse generation	Section 24.3
187	–			–
188	CRC	CRC S D n	Cyclic Redundancy Check	Section 24.4
189	HCMOV	HCMOV S D n	High speed counter move	Section 24.5

24.1 FNC182 – COMRD / Read Device Comment Data

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

This instruction reads the comment data for devices registered (written) by programming software such as GX Developer.

1. Instruction format

FNC 182 COMRD P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	COMRD	Continuous Operation
	COMRDP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	–

2. Set data

Operand Type	Description	Data Type
S·	Device number in which comment to be read is registered	Device name
D·	Head device number storing read comment	Character string

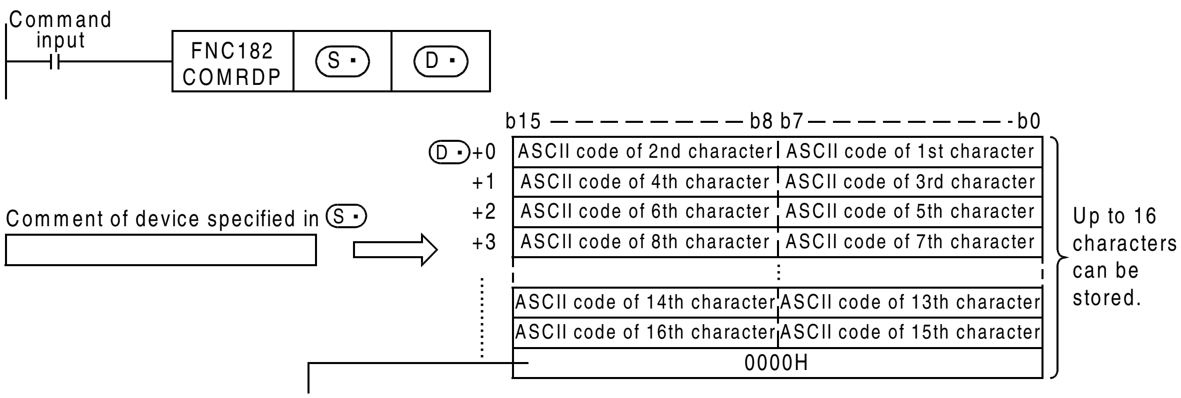
3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·	✓	✓	✓			✓						✓	✓	✓	✓				✓					
D·												✓	✓	✓	✓				✓					

Explanation of function and operation

1. 16-bit operation (COMRD and COMRDP)

1) The comment registered in the device S· is read, and stored in ASCII codes in D· and later.



When the number of characters in the comment is even
· When M8091 is OFF, "0000H" is stored in the next device of the device storing the final character.
· When M8091 is ON, the next device of the device storing the final character does not change.

For example, when the comment of (S·) is "LineNo.1Start", it is stored in (D·) and later as shown below.

Comment of (S·)

LineNo.1Start

	b15 – b8	b7 – b0
(D·)+0	69H(i)	4CH(L)
+1	65H(e)	6EH(n)
+2	6FH(o)	4EH(N)
+3	31H(1)	2EH(.)
+4	74H(t)	53H(S)
+5	72H(r)	61H(a)
+6	00H	74H(t)

When the number of characters in the comment is odd
· When M8091 is OFF, "00H" is stored in the high-order byte of the device storing the final character.
· When M8091 is ON, the high-order byte of the device storing the final character does not change.

2) The final data of (D·) is as follows depending on the ON/OFF status of M8091.

ON/OFF status	Contents of processing
M8091 = OFF	• When the number of characters in the comment is odd, "00H" is stored in the high-order byte (8 bits) of the device storing the final character. • When the number of characters in the comment is even, "00H" is stored in the next device of the device storing the final character.
M8091 = ON	• When the number of characters in the comment is odd, the high-order byte (8 bits) of the device storing the final character does not change. • When the number of characters in the comment is even, the next device of the device storing the final character does not change.

Related device

Device	Name	Description
M8091	Output character number selector signal	Refer to the above explanation.

Caution

- To the device (S·), specify a device number for which comment is registered (written) in the PLC.
  If comment is not registered (written) for the device (S·), "20H" (space) is stored in (D·) and later for the number of characters in the comment (16 half-width characters).

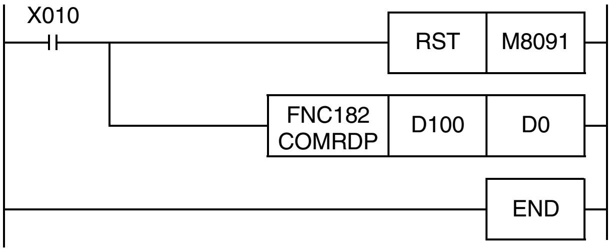
Errors

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When any comment is not registered (written) in the device (S·) (error code: K6706)
- When the range of points used after (D·) for comment exceeds the corresponding device range (error code: K6706)

Program example

In the program shown below, the comment "Target Line A" set in D100 is stored in ASCII codes in D0 and later (when M8091 is OFF) when X010 is set to ON.



Comment of D100

Target Line A

	b15 – – – – – – – b8	b7 – – – – – – – – b0
D0	61H(a)	54H(T)
D1	67H(g)	72H(r)
D2	74H(f)	65H(e)
D3	4CH(L)	20H(space)
D4	6EH(n)	69H(i)
D5	20H(space)	65H(e)
D6	00H	41H(A)

24.2 FNC184 – RND / Random Number Generation

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction generates random numbers.

1. Instruction format

FNC 184 RND P

16-bit Instruction	Mnemonic	Operation Condition
3 steps	RND	Continuous Operation
	RNDP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
D·	Head device number storing a random number	16-bit binary

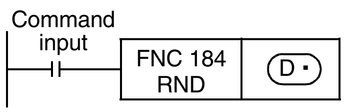
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
D·									✓	✓	✓	✓	✓	✓	✓	✓			✓					

Explanation of function and operation

1. 16-bit operation (RND and RNDP)

This instruction generates a pseudo-random number within the range from 0 to 32767, and stores it as a random number to D·.
In the pseudo-random number sequence, the source value of a random number is calculated at every time, and this instruction calculates a pseudo-random number using the source value.



Pseudo-random number calculation equation:

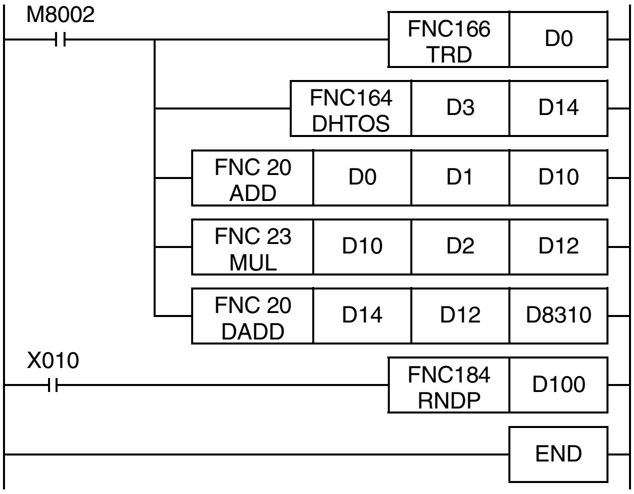
(D8311, D8310) = (D8311, D8310) *1 × 1103515245 + 12345.....(1)

D· = "([D8311, D8310]>>16)&<logical product>00007FFFh"

*1. To (D8311, D8310), write a non-negative value (0 to 2,147,483,647) only once when the PLC mode switches from STOP to RUN.
[K1 is written to (D8311, D8310) as the initial value when the power is restored.]

Program example

In the program example shown below, a random number is stored to D100 every time X010 turns ON.
When the PLC mode switches from STOP to RUN, the time data converted into seconds and added by the value "(Year + Month) × Day" is written to D8311 and D8310.



The clock data is read.

Data in hour, minute and second → Data in second

The data in second is added by the value "(Year + Month) × Day", and written to D8311 and D8310.

24.3 FNC186 – DUTY / Timing Pulse Generation

Outline

FX3UC Ver.2.20 ➡ FX3U Ver.2.20 ➡

This instruction generates the timing signal whose one cycle corresponds to the specified number of operation cycles.

1. Instruction format

FNC 186 DUTY

16-bit Instruction	Mnemonic	Operation Condition
7 steps	DUTY	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand Type	Description	Data Type
n1	Number of scans (operation cycles) to remain ON [n1 > 0]	16-bit binary
n2	Number of scans (operation cycles) to remain OFF [n2 > 0]	16-bit binary
(D·)	Timing clock output destination	Bit

3. Applicable devices

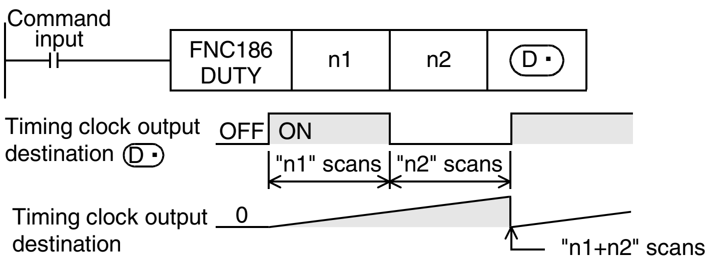
Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
n1												✓	✓	✓	✓					✓	✓			
n2												✓	✓	✓	✓					✓	✓			
(D·)			▲																✓					

▲: Specify either one among M8330 to M8334.

Explanation of function and operation

1. 16-bit operation (DUTY)

1) The timing clock output destination (D·) is set to ON and OFF with the ON duration for "n1" scans and OFF duration for "n2" scans.



2) Specify either one among M8330 to M8334 as the timing clock output destination device (D·).

21 FNC160-FNC169 Real Time Clock Control
22 FNC170-FNC179 External Device
23 FNC180 Alternate Instructions
24 FNC181-FNC189 Others
25 FNC190-FNC199 Block Data Operation
26 FNC200-FNC209 Character String Control
27 FNC210-FNC219 Data Operation 3
28 FNC220-FNC249 Data Comparison
29 FNC250-FNC269 Data Table Operation
30 FNC270-FNC274 Ex-Device Inverter Comms

3) The counted number of scans is stored in either one among D8330 to D8334 in accordance with the timing clock output destination device D·.
The counted number of scans stored in either one among D8330 to D8334 is reset when the counted value reaches "n1+n2" or when the command input (instruction) is set to ON.

Timing clock output destination device D·	Scan counting device
M8330	D8330
M8331	D8331
M8332	D8332
M8333	D8333
M8334	D8334

4) When the command input is set to ON, the operation is started. The timing clock output destination device D· is set to ON or OFF by END instruction.
Even if the command input is set to OFF, the operation is not stopped.
In the STOP mode, the operation is suspended. When the power of the PLC is turned OFF, the operation is stopped.

5) When "n1" and "n2" are set to "0", the device D· is set to the following status:

n1/n2 status	D· ON/OFF status
n1 = 0, n2 ≥ 0	D· Fixed to OFF
n1 > 0, n2 = 0	D· Fixed to ON

Related devices

Device	Name	Description
M8330	Timing clock output 1	Timing clock output in DUTY (FNC186) instruction
M8331	Timing clock output 2	
M8332	Timing clock output 3	
M8333	Timing clock output 4	
M8334	Timing clock output 5	
D8330	Counted number of scans for timing clock output 1	Counted number of scans for timing clock output 1 in DUTY (FNC186) instruction
D8331	Counted number of scans for timing clock output 2	Counted number of scans for timing clock output 2 in DUTY (FNC186) instruction
D8332	Counted number of scans for timing clock output 3	Counted number of scans for timing clock output 3 in DUTY (FNC186) instruction
D8333	Counted number of scans for timing clock output 4	Counted number of scans for timing clock output 4 in DUTY (FNC186) instruction
D8334	Counted number of scans for timing clock output 5	Counted number of scans for timing clock output 5 in DUTY (FNC186) instruction

Caution

- DUTY (FNC186) instruction can be used up to 5 times (points).
  It is not permitted, however, to use the same timing clock output destination device D· for two or more DUTY (FNC186) instructions.

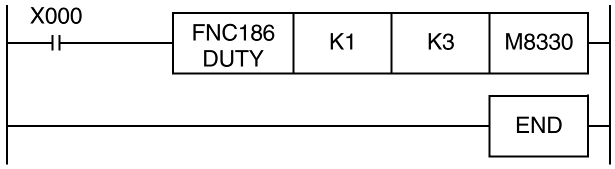
Errors

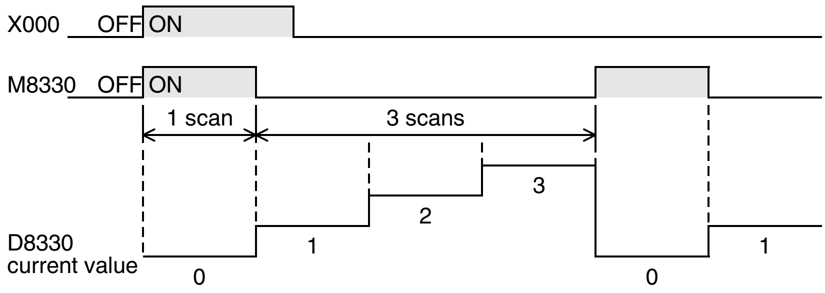
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When "n1" and/or "n2" is less than "0" (error code: K6706)
- When any device other than M8330 to M834 is set to D· (error code: K6705)

Program example

In the program shown below, when X000 is set to ON, M8330 is set to ON for 1 scan and OFF for 3 scans.





24.4 FNC188 – CRC / Cyclic Redundancy Check

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This CRC instruction calculates the CRC (cyclic redundancy check) value which is an error check method used in communication.
In addition to CRC value, there are other error check methods such as parity check and sum check. For obtaining the horizontal parity value and sum check value, CCD (FNC 84) instruction is available.
CRC instruction uses “$X^{16} + X^{15} + X^{2} + 1$” as a polynomial for generating the CRC value (CRC-16).

→ **For CCD instruction (check code), refer to Section 16.5.**

1. Instruction format

FNC 188 CRC P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	CRC	Continuous Operation
	CRCP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
S·	Head device number storing data for which the CRC value is generated	16-bit binary
D·	Device number storing the generated CRC value	
n	Number of 8-bit (1-byte) data for which the CRC value is generated or the device number storing the number of data	

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·								▲	▲	▲	▲	✓	✓	✓	✓	✓			✓					
D·									▲	▲	▲	✓	✓	✓	✓	✓			✓					
n														✓	✓					✓	✓			

▲: Make sure to specify four digits (K4□○○○) when specifying the digits of a bit device.

Explanation of function and operation

1. 16-bit operation

CRC value is generated for “n” 8-bit data (unit: byte) starting from a device specified in S·, and stored to D·.
The 8-bit conversion mode and 16-bit conversion mode are available in this instruction, and the mode can be switched by turning ON or OFF M8161. For the operation in each mode, refer to the following pages.
“$X^{16} + X^{15} + X^{2} + 1$” is used as a polynomial for generating the CRC value (CRC-16).

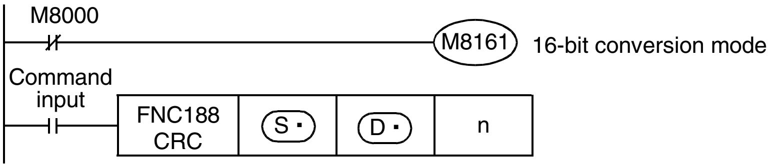
Command input

FNC188 CRC	S·	D·	n

16-bit conversion mode (while M8161 is OFF)

In this mode, the operation is executed for high-order 8 bits (1 byte) and low-order 8 bits (1 byte) of a device specified in S•.

The operation result is stored to one 16-bit device specified in D•.

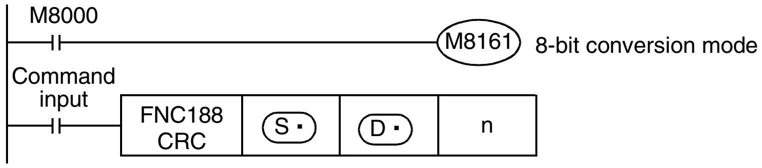


			Example: S• = D100, D• = D0, n = 6		
			Device	Contents of target data	
				8 bits	16 bits
Device storing data for which the CRC value is generated	S•	Low-order byte	Low-order bits of D100	01H	0301H
		High-order byte	High-order bits of D100	03H	
	S• +1	Low-order byte	Low-order bits of D101	03H	0203H
		High-order byte	High-order bits of D101	02H	
	S• +2	Low-order byte	Low-order bits of D102	00H	1400H
		High-order byte	High-order bits of D102	14H	
	⋮	⋮	–		
	S• +n/2-1	Low-order byte	–		
		High-order byte			
Device storing the generated CRC value	D•	Low-order byte	Low-order bits of D0	E4H	41E4H
		High-order byte	High-order bits of D0	41H	

8-bit conversion mode (while M8161 is ON)

In this mode, the operation is executed only for low-order 8 bits (low-order 1 byte) of a device specified by (S•).

With regard to the operation result, low-order 8 bits (1 byte) are stored to a device specified by (D•), and high-order 8 bits (1 byte) are stored to a device specified by (D•)+1.



			Example: (S•) = D100, (D•) = D0, n = 6	
			Device	**Contents of target data**
Device storing data for which the CRC value is generated	(S•)	Low-order byte	Low-order bits of D100	01H
	(S•)+1	Low-order byte	Low-order bits of D101	03H
	(S•)+2	Low-order byte	Low-order bits of D102	03H
	(S•)+3	Low-order byte	Low-order bits of D103	02H
	(S•)+4	Low-order byte	Low-order bits of D104	00H
	(S•)+5	Low-order byte	Low-order bits of D105	14H
	⋮		–	
	(S•)+n-1	Low-order byte	–	
Device storing the generated CRC value	(D•)	Low-order byte	Low-order bits of D0	E4H
	(D•)+1	Low-order byte	Low-order bits of D1	41H

2. Related device

Related device	Description	
M8161*1	ON	CRC instruction operates in the 8-bit mode.
	OFF	CRC instruction operates in the 16-bit mode.

*1. Cleared when the PLC mode is changed from RUN to STOP.

Caution

In this instruction, "$X^{16} + X^{15} + X^2 + 1$" is used as a polynomial for generating the CRC value (CRC-16). There are many other standard polynomials for generating the CRC value. Note that the CRC value completely differs if an adopted polynomial is different.

Reference: Major polynomials for generating the CRC value

Name	Polynomial
CRC-12	$X^{12} + x^{11} + X^3 + X^2 + X + 1$
CRC-16	$X^{16} + X^{15} + X^2 + 1$
CRC-32	$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
CRC-CCITT	$X^{16} + X^{12} + X^5 + 1$

Errors

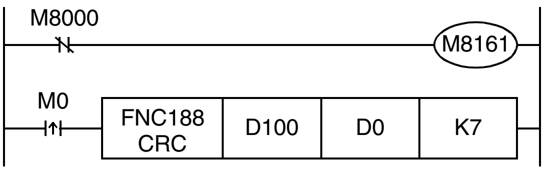
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When any digits other than 4 digits are specified as (S·) or (D·) in digit specification of bit device (error code: K6706)
- When n is outside the allowable range (1 to 256) (error code: K6706)
- When a device specified by (S·)+n-1 or (D·)+1 is outside the allowable range (error code: K6706)

Program example

In the program example shown below, the CRC value of the ASCII code “0123456” stored in D100 to D106 is generated and stored to D0 when M0 turns ON.

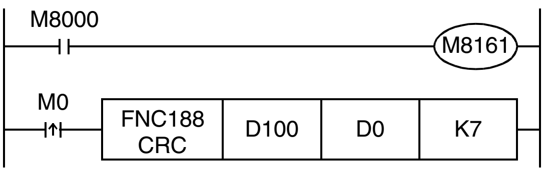
1. In the case of 16-bit mode



16-bit conversion mode

	Contents of data			
			Target data	
Device storing data for which CRC value is generated	D100	3130H	Low-order byte	30H
			High-order byte	31H
	D101	3332H	Low-order byte	32H
			High-order byte	33H
	D102	3534H	Low-order byte	34H
			High-order byte	35H
	D103	3736H	Low-order byte	36H
			–	–
Device storing generated CRC value	D0	2ACFH	Low-order byte	CFH
			High-order byte	2AH

2. In the case of 8-bit mode



8-bit conversion mode

	Contents of target data		
Device storing data for which the CRC value is generated	D100	Low-order byte	30H
	D101	Low-order byte	31H
	D102	Low-order byte	32H
	D103	Low-order byte	33H
	D104	Low-order byte	34H
	D105	Low-order byte	35H
	D106	Low-order byte	36H
Device storing the generated CRC value	D0	Low-order byte	CFH
	D1	Low-order byte	2AH

24.5 FNC189 – HCMOV / High Speed Counter Move

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction updates the current value of a specified high speed counter or ring counter.
The function of this instruction varies depending on the PLC version.

1. Instruction format

D FNC 189 HCMOV

16-bit Instruction	Mnemonic	Operation Condition
	—	—

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DHCMOV	Continuous Operation
	—	

2. Set data

Operand Type	Description	Data Type
S	Device number of high speed counter or ring counter*1 handled as transfer source	32-bit binary
D	Device number handled as transfer destination	
n	Specification to clear the current value of high speed counter or ring counter*1 (transfer source) after transfer [clear (K1), no clear (K0)]	16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index				Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify		K	H	E	"□"	P
S													▲	▲											
D														✓	✓										
n																					✓	✓			

▲: Only high speed counters (C235 to C255) and ring counters (D8099 and D8398)*1 can be specified.

Explanation of function and operation

1. 32-bit operation (DHCMOV)

Command input

FNC189 DHCMOV	S	D	n

- The current value of a high speed counter or ring counter specified in S is transferred to [D+1, D].

Device S		[D+1, D] after instruction is executed
High speed counter	C235 to C255	Current value of high speed counter S → [D+1, D]
Ring counter*1	D8099	D8099 → D "0" is stored in D+1.
	D8398	Current value of [D8399, D8398] → [D+1, D]

- After transfer, the current value of the high speed counter or ring counter is processed as shown in the table below depending on the set value of "n":

"n" set value	Operation
K0 (H0)	Does not clear the current value (no processing).
K1 (H1)	Clears the current value to "0".

*1. Ring counters (D8099 and D8398) cannot be specified in FX3UC PLCs before Ver.2.20.

2. High speed counter current value update timing and the effect of DHCMOV instruction

1) High speed counter current value update timing
When a pulse is input to an input terminal for a high speed counter (C235 to C255), the high speed counter executes up-counting or down-counting.
If the current value of a high speed counter is handled in an applied instruction such as the normal MOV instruction, the current value is updated at the timing shown in the table below. As a result, it is affected by the program scan time.

	Current value update timing
Hardware counter	When OUT instruction for the counter is executed
Software counter	Every time a pulse is input

By using DHCMOV instruction, the current value can be updated and transferred when it is executed.

2) Effect of DHCMOV instruction

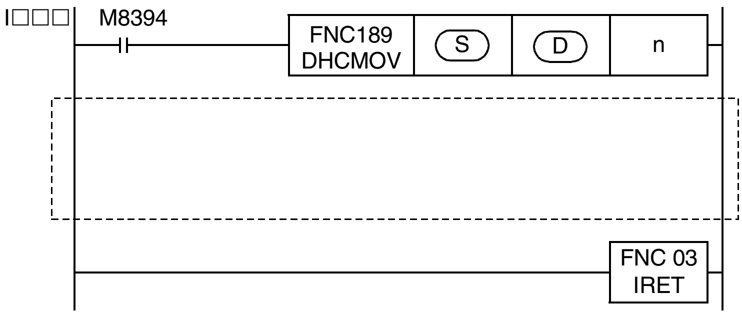
- By using both input interrupt and DHCMOV instruction, the current value of a high speed counter can be received at the rising edge or falling edge of an external input (at reception of input interrupt).
→ **Refer to the Program example 2.**
- When DHCMOV instruction is used just before a comparison instruction (CMP, ZCP or comparison contact instruction), the latest value of a high speed counter is used in comparison.The following points must be kept in mind when using the DHCMOV command.
  - When the current value of a high speed counter is compared using CMP, ZCP or comparison contact instruction (not using a designated high speed counter comparison instruction), a hardware counter does not change into a software counter.
→ **For the condition in which a hardware counter is handled as a software counter, refer to Subsection 4.7.9.**
  - When the number of high speed software counter comparison instructions is reduced, the total frequency limitation is decreased.
→ **For the limitation in software counters by the total frequency, refer to Subsection 4.7.10.**
  - When it is necessary to execute comparison and change an output contact (Y) as soon as the current value of a high speed counter changes, use a designhated high speed counter comparison instruction (HSCS, HSCR or HSZ).
  - DHCMOV instruction can be used as many times as necessary.

Cautions

When programming DHCMOV instruction in an input interrupt program, the following points should be observed.
For assignment of pointers for input interrupt and inputs, refer to the table shown in 5) below.

1) Program EI (FNC 04) and FEND (FNC 06) instructions in the main program. They are necessary to execute an input interrupt program.
→ For EI (FNC 04) and FEND (FNC 06) instructions, refer to Section 8.5 and Section 8.6.

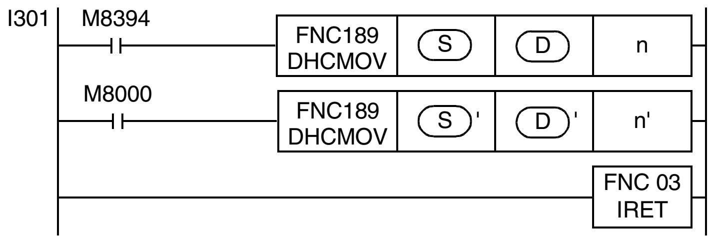
2) When programming DHCMOV instruction in the 1st line in an input interrupt program, make sure to use the pattern program shown below.
Make sure to use the command contact M8394.



Pattern program for using the DHCMOV instruction in an interrupt program

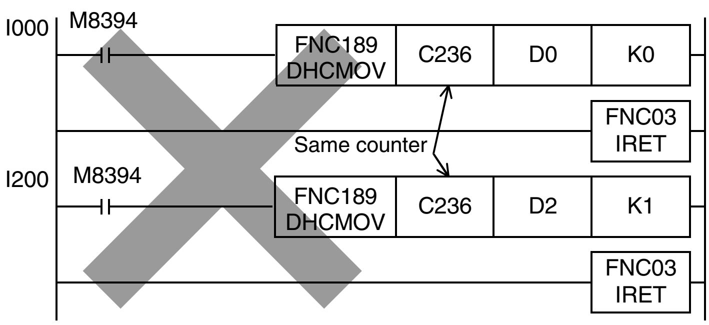
Program to be processed as input interrupt

3) If two or more DHCMOV instructions are used in one input interrupt program, only the first instruction (just after the interrupt pointer) is executed when the interrupt is generated.
The rest of the interrupt, including additional DHCMOV instructions, is executed according to normal interrupt processing.
Do not use M8394 as the command contact for the DHCMOV instructions following the first.



When the input X003 turns from OFF to ON (that is, when input interrupt is accepted):
S → [D+1, D]
When this instruction is executed in interrupt program:
S' → [D' +1, D']

4) It is not permitted to use DHCMOV instruction for the same counter in two or more input interrupt programs.



5) While input interrupts are disabled by the interrupt disable flags (shown in the table below), DHCMOV instructions are not executed when they are placed inside a corresponding interrupt.

Interrupt disable flag	Corresponding interrupt pointer	Input number corresponding to interrupt pointer
M8050*1	I000,I001	X000
M8051*1	I100,I101	X001
M8052*1	I200,I201	X002
M8053*1	I300,I301	X003
M8054*1	I400,I401	X004
M8055*1	I500,I501	X005

*1. Cleared when the PLC mode is changed from RUN to STOP.

6) If an input interrupt is generated while input interrupts are disabled by something other than the interrupt disable flags M8050 to M8055 (after execution of DI instruction and before execution of EI instruction), DHCMOV instruction is immediately executed, but execution of the interrupt program is held. The interrupt program will be executed after EI instruction is executed and interrupts are enabled.

Function change depending on the version

The function of FNC189 instruction changes depending on the version as shown in the table below.

Applicable version		Item	Outline of function
FX3U	FX3UC		
Ver.2.20 or later	Ver.2.20 or later	Target device	Ring counter (D8099 and D8398) can be specified in S.

Error

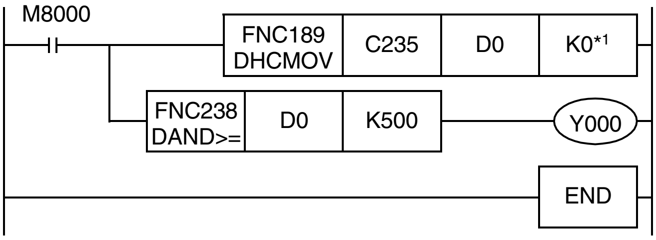
An operation error occurs in the following case; The error flag M8067 turns ON, and the error code is stored in D8067.

- When a device specified in S· or [D·+1, D·] is outside the allowable range (error code: K6705)

Program examples

1. Program example 1

In the program example below, the current value of the high speed counter C235 is compared in each operation cycle, and then the output Y000 is set to ON if the current value is "K500" or more (when the current value of C235 is not cleared).



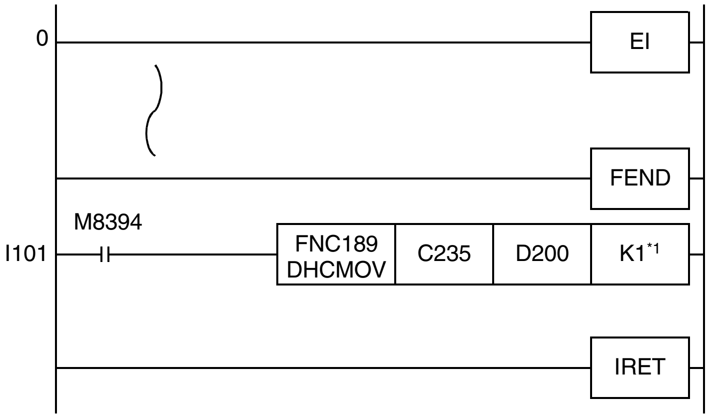
The current value of C235 is transferred to D1 and D0.
(The current value of C235 is not cleared.)

In the case of "(D1, D0) ≥ K500", Y000 is set to ON.

*1. K0: The current value of the high speed counter is not cleared when DHCMOV instruction is executed.
K1: The current value of the high speed counter is cleared when DHCMOV instruction is executed.

2. Program example 2

In the program example shown below, the current value of C235 is transferred to D201 and D200, and the current value of C235 is cleared when X001 turns from OFF to ON.



- When X001 turns from OFF to ON, the interrupt program from I101 to IRET is executed.
- The current value of C235 is transferred to D201 and D200. (The current value of C235 is cleared to “0”.)

*1. K0: The current value of the high speed counter is not cleared when DHCMOV instruction is executed.
K1: The current value of the high speed counter is cleared when DHCMOV instruction is executed.

25. Block Data Operation – FNC190 to FNC199

FNC190 to FNC199 provide instructions for adding, subtracting and comparing block data.

FNC No.	Mnemonic	Symbol	Function	Reference
190	–			–
191	–			–
192	BK+	BK+ S1 S2 D n	Block Data Addition	Section 25.1
193	BK–	BK- S1 S2 D n	Block Data Subtraction	Section 25.2
194	BKCMP=	BKCMP= S1 S2 D n	Block Data Compare S1 = S2	Section 25.3
195	BKCMP>	BKCMP> S1 S2 D n	Block Data Compare S1 > S2	Section 25.3
196	BKCMP<	BKCMP< S1 S2 D n	Block Data Compare S1 < S2	Section 25.3
197	BKCMP<>	BKCMP<> S1 S2 D n	Block Data Compare S1 ≠ S2	Section 25.3
198	BKCMP<=	BKCMP<= S1 S2 D n	Block Data Compare S1 ≤ S2	Section 25.3
199	BKCMP>=	BKCMP>= S1 S2 D n	Block Data Compare S1 ≥ S2	Section 25.3

21 FNC160-FNC169 Real Time Clock Control
22 FNC170-FNC179 External Device
23 FNC180 Alternate Instructions
24 FNC181-FNC189 Others
25 FNC190-FNC199 Block Data Operation
26 FNC200-FNC209 Character String Control
27 FNC210-FNC219 Data Operation 3
28 FNC220-FNC249 Data Comparison
29 FNC250-FNC269 Data Table Operation
30 FNC270-FNC274 Ex-Device Inverter Comms

25.1 FNC192 – BK+ / Block Data Addition

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

This instruction adds binary block data.

1. Instruction format

D	FNC 192 BK+	P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	BK+	Continuous Operation
	BK+P	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DBK+	Continuous Operation
	DBK+P	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S1•	Head device number storing addition data	16- or 32-bit binary
S2•	Added constant or head device number storing addition data	
D•	Head device number storing operation result	
n	Number of data	

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index				Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•												✓	✓	✓	✓				✓					
S2•												✓	✓	✓	✓				✓	✓	✓			
D•												✓	✓	✓	✓				✓					
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (BK+ and BK+P)

Command input	FNC192 BK+	S1•	S2•	D•	n

1) "n" 16-bit binary data starting from S2• are added to "n" 16-bit binary data starting from S1•, and the operation result is stored in "n" points starting from D•.

S1• (b15 ---- b0)		+	S2• (b15 ---- b0)		⇨	D• (b15 ---- b0)	
+0	K1234		+0	K4000		+0	K5234
+1	K4567		+1	K1234		+1	K5801
+2	K-2000		+2	K-1234		+2	K-3234
+(n-2)	K-1234		+(n-2)	K5000		+(n-2)	K3766
+(n-1)	K4000		+(n-1)	K4321		+(n-1)	K8321

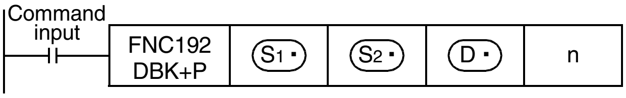
"n" points

2) A (16-bit) constant from -32768 to +32767 can be directly specified in S2•.

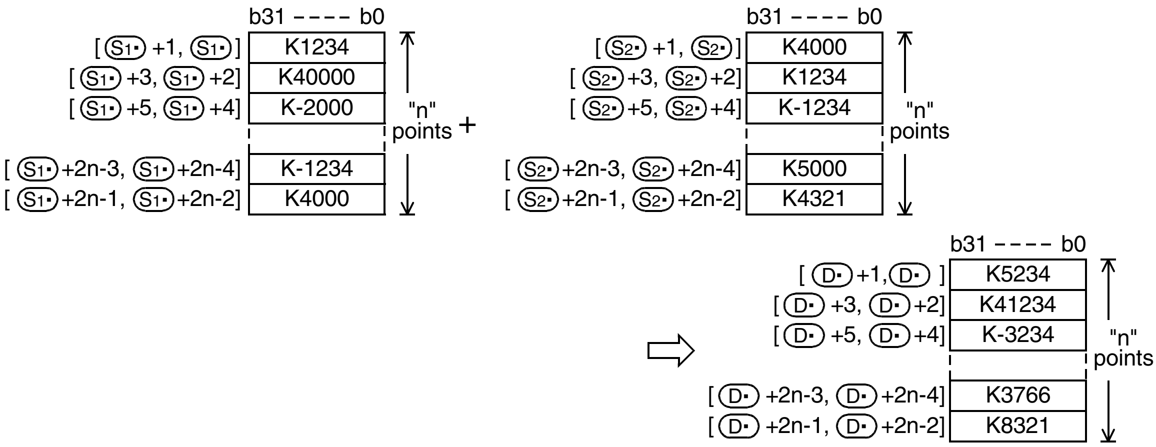
S1• (b15 ---- b0)		+	S2•	⇨	D• (b15 ---- b0)	
+0	K1234				+0	K5555
+1	K4567				+1	K8888
+2	K-2000		K4321		+2	K2321
+(n-2)	K-1234				+(n-2)	K3087
+(n-1)	K4000				+(n-1)	K8321

"n" points

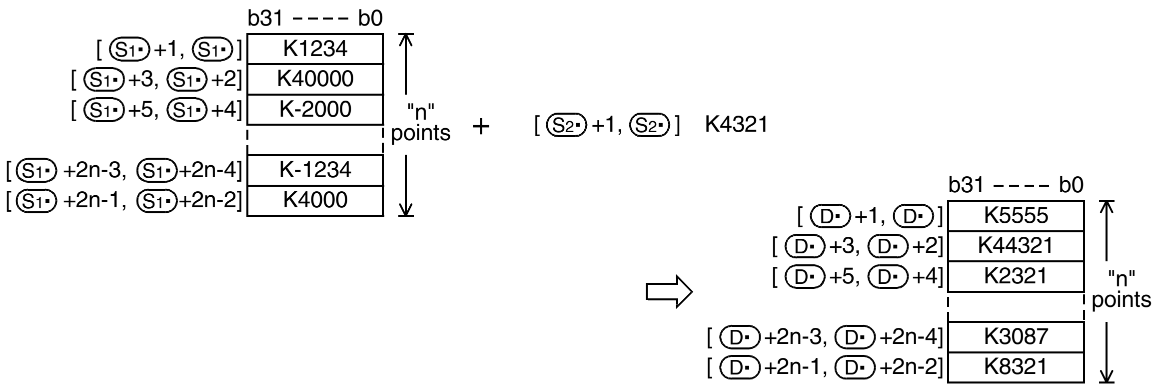
2. 32-bit operation (DBK+ and DBK+P)



1) "2n" 32-bit binary data starting from [S2• +1, S2•] are added to "2n" 32-bit binary data starting from [S1• +1, S1•], and the operation result is stored in "2n" points starting from [D• +1, D•].



2) A (32-bit) constant from -2,147,483,648 to +2,147,483,647 can be directly specified in [S2• +1, S2•].



Related instruction

Instruction	Description
BK- (FNC193)	Subtracts binary block data.

Caution

1) When underflow or overflow occurs in the operation result, the following processing is executed. At this time, the carry flag does not turn ON.
   - In the case of 16-bit operation

     K32767(H7FFF) + K2(H0002) → K-32767(H8001)
     K-32767(H8000) + K-2(HFFFE) → K32766(H7FFE)

   - In the case of 32-bit operation

     K2,147,483,647(H7FFFFFFF) + K2(H00000002) → K-2,147,483,647(H80000001)
     K-2,147,483,648(H80000000) + K-2(HFFFFFFFE) → K2,147,483,646(H7FFFFFFE)

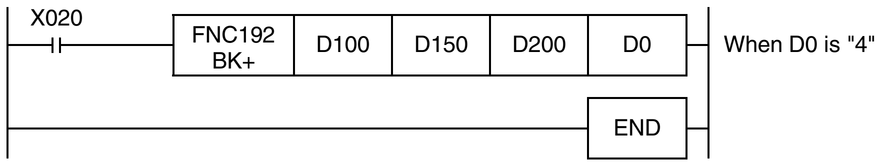
Errors

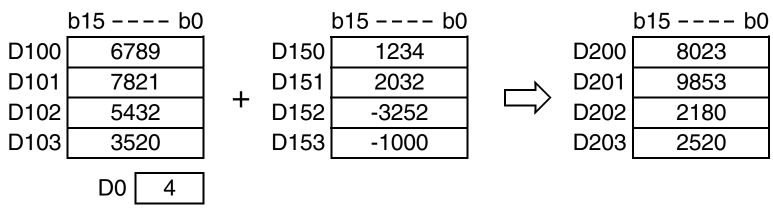
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When "n" ("2n" in 32-bit operation) devices starting from S1•, S2•, and/or D• exceed the corresponding device range (error code: K6706)
- When "n" ("2n" in 32-bit operation) devices starting from S1• overlap "n" ("2n" in 32-bit operation) devices starting from D• (error code: K6706)
- When "n" ("2n" in 32-bit operation) devices starting from S2• overlap "n" ("2n" in 32-bit operation) devices starting from D• (error code: K6706)

Program example

In the program shown below, the specified number of data stored in D150 to D0 are added to the specified number of data stored in D100 to D0 when X020 is set to ON, and the operation result is stored in D200 and later.





25.2 NFC193 – BK– / Block Data Subtraction

FX3UC Ver.2.20 → FX3U Ver.2.20 →

Outline

This instruction subtracts binary block data.

1. Instruction format

FNC 193	
D	BK–	P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	BK– BK–P	Continuous Operation Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DBK– DBK–P	Continuous Operation Pulse (Single) Operation

2. Set data

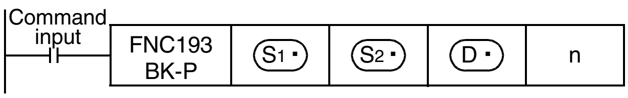
Operand Type	Description	Data Type
S1•	Head device number storing subtraction data	16- or 32-bit binary
S2•	Subtracted constant or head device number storing subtraction data	
D•	Head device number storing operation result	
n	Number of data	

3. Applicable devices

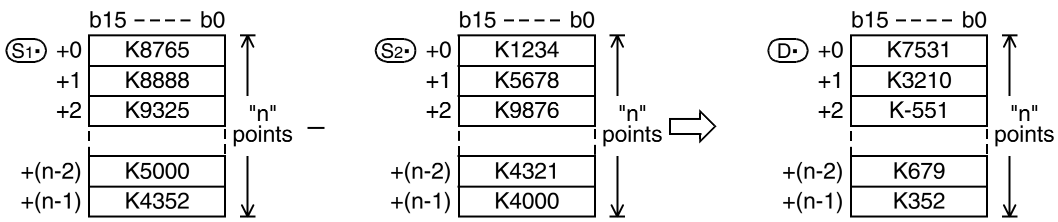
Operand Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•												✓	✓	✓	✓				✓					
S2•												✓	✓	✓	✓				✓	✓	✓			
D•												✓	✓	✓	✓				✓					
n														✓	✓					✓	✓			

Explanation of function and operation

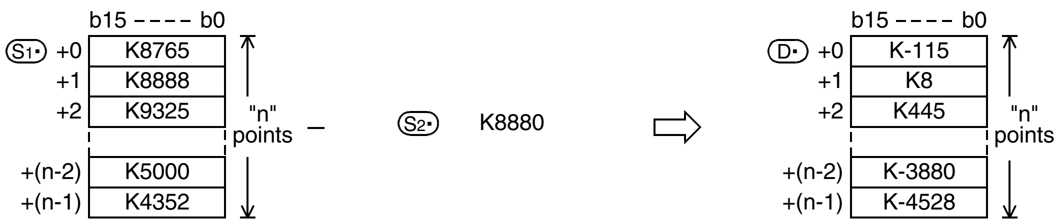
1. 16-bit operation (BK- and BK-P)



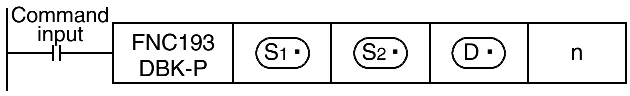
1) "n" 16-bit binary data starting from S2• are subtracted from "n" 16-bit binary data starting from S1•, and the operation result is stored in "n" points starting from D•.



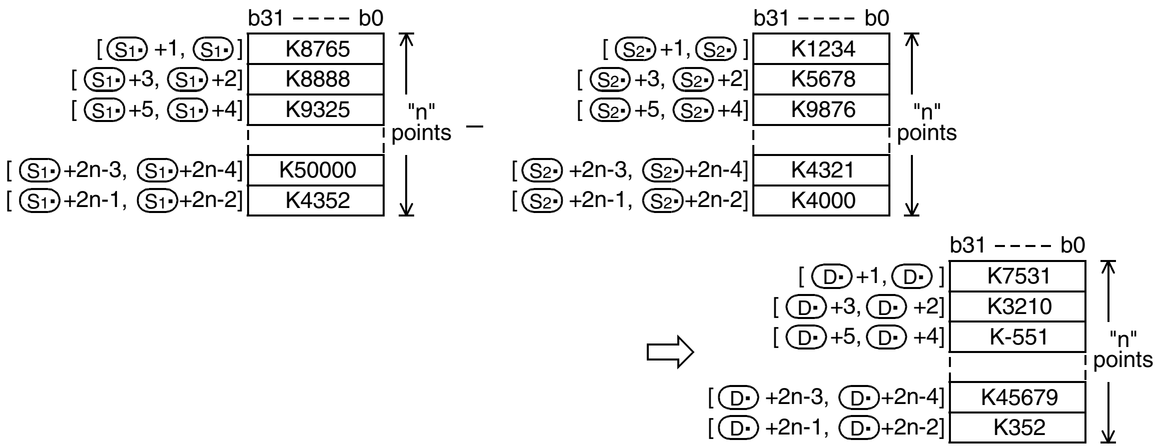
2) A (16-bit) constant from -32768 to +32767 can be directly specified in S2•.



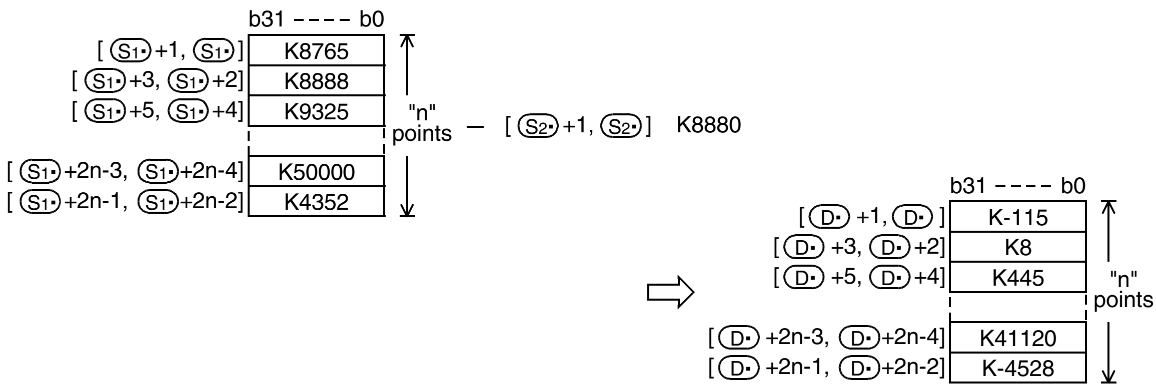
2. 32-bit operation (DBK- and DBK-P)



1) "2n" 32-bit binary data starting from [S2•+1, S2•] are subtracted from "2n" 32-bit binary data starting from [S1•+1, S1•], and the operation result is stored in "2n" points starting from [D•+1, D•].



2) A (32-bit) constant from -2,147,483,648 to +2,147,483,647 can be directly specified in [S2•+1, S2•].



Related instruction

Instruction	Description
BK+ (FNC192)	Adds binary block data.

Caution

1) When underflow or overflow occurs in the operation result, the following processing is executed. At this time, the carry flag does not turn ON.
   - In the case of 16-bit operation

K-32767(H8000)	–	K2(H0002)	→	K32766(H7FFE)
K32767(H7FFF)	–	K-2(HFFFE)	→	K-32766(H8001)

   - In the case of 32-bit operation

K-2,147,483,648(H80000000)	–	K2(H00000002)	→	K2,147,483,646(H7FFFFFFE)
K2,147,483,647(H7FFFFFFF)	–	K-2(HFFFFFFFE)	→	K-2,147,483,647(H80000001)

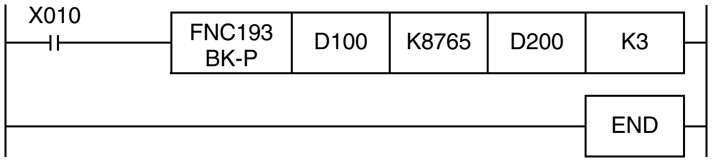
Errors

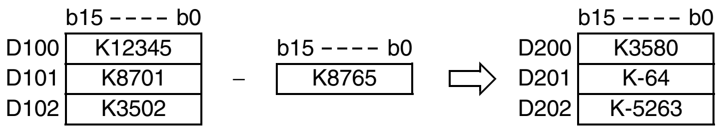
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When "n" ("2n" in 32-bit operation) devices starting from (S1·), (S2·), and/or (D·) exceed the corresponding device range (error code: K6706)
- When "n" ("2n" in 32-bit operation) devices starting from (S1·) overlap "n" ("2n" in 32-bit operation) devices starting from (D·) (error code: K6706)
- When "n" ("2n" in 32-bit operation) devices starting from (S2·) overlap "n" ("2n" in 32-bit operation) devices starting from (D·) (error code: K6706)

Program example

In the program shown below, the constant "8765" is subtracted from the data stored in D100 to D102 when X010 is set to ON, and the operation result is stored in D200 and later.





25.3 FNC194~199 – BKCMP=, >, <, < >, <=, >= / Block Data Compare

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

These instructions compare block data in the comparison condition set in each instruction.

1. Instruction format

FNC	16-bit Instruction	Mnemonic	Operation Condition	32-bit Instruction	Mnemonic	Operation Condition
FNC 194 D BKCMP= P	9 steps	BKCMP= BKCMP=P	Continuous Operation Pulse (Single) Operation	17 steps	DBKCMP= DBKCMP=P	Continuous Operation Pulse (Single) Operation
FNC 195 D BKCMP> P	9 steps	BKCMP> BKCMP>P	Continuous Operation Pulse (Single) Operation	17 steps	DBKCMP> DBKCMP>P	Continuous Operation Pulse (Single) Operation
FNC 196 D BKCMP< P	9 steps	BKCMP< BKCMP<P	Continuous Operation Pulse (Single) Operation	17 steps	DBKCMP< DBKCMP<P	Continuous Operation Pulse (Single) Operation
FNC 197 D BKCMP<> P	9 steps	BKCMP<> BKCMP<>P	Continuous Operation Pulse (Single) Operation	17 steps	DBKCMP<> DBKCMP<>P	Continuous Operation Pulse (Single) Operation
FNC 198 D BKCMP<= P	9 steps	BKCMP<= BKCMP<=P	Continuous Operation Pulse (Single) Operation	17 steps	DBKCMP<= DBKCMP<=P	Continuous Operation Pulse (Single) Operation
FNC 199 D BKCMP>= P	9 steps	BKCMP>= BKCMP>=P	Continuous Operation Pulse (Single) Operation	17 steps	DBKCMP>= DBKCMP>=P	Continuous Operation Pulse (Single) Operation

2. Set data (common among FNC194 to FNC199)

Operand Type	Description	Data Type
S1•	Comparison value of device number storing comparison value	16- or 32-bit binary
S2•	Head device number storing comparison source data	16- or 32-bit binary
D•	Head device number storing comparison result	Bit
n	Number of compared data	16- or 32-bit binary

3. Applicable devices (common among FNC194 to FNC199)

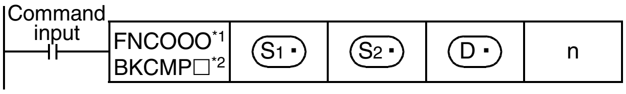
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•												✓	✓	✓	✓				✓	✓	✓			
S2•												✓	✓	✓	✓				✓					
D•		✓	✓			✓	▲												✓					
n														✓	✓					✓	✓			

▲: "D□.b" cannot be indexed with index registers (V and Z).

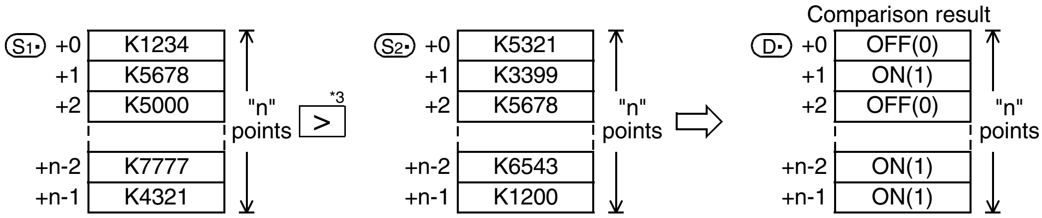
Explanation of function and operation

1. 16-bit operation (BKCMP=, >, <, <>, <=, >= / BKCMP=P, >P, <P, <>P, <=P, and >=P)

1) "n" 16-bit binary data starting from (S1•) are compared with "n" 16-bit binary data starting from (S2•), and the comparison result is stored in "n" points starting from (D•).

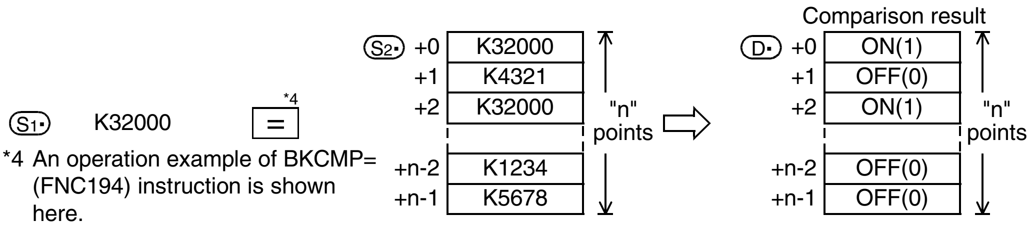


*1 The number out of 194 to 199 is put in "OOO".
*2 The symbol out of "=", ">", "<", "<>", "<=", or ">=" corresponding to the FNC No. is put in □.



*3 An operation example of BKCMP> (FNC195) instruction is shown here.

2) A constant can be directly specified in (S1•).



*4 An operation example of BKCMP= (FNC194) instruction is shown here.

3) The table below shows the comparison result in each instruction:

Instruction	Comparison result ON (1) condition	Comparison result OFF (0) condition
BKCMP=(FNC194)	(S1•) = (S2•)	(S1•) <> (S2•)
BKCMP>(FNC195)	(S1•) > (S2•)	(S1•) <= (S2•)
BKCMP<(FNC196)	(S1•) < (S2•)	(S1•) >= (S2•)
BKCMP<>(FNC197)	(S1•) <> (S2•)	(S1•) = (S2•)
BKCMP<=(FNC198)	(S1•) <= (S2•)	(S1•) > (S2•)
BKCMP>=(FNC199)	(S1•) >= (S2•)	(S1•) < (S2•)

4) When the comparison result is ON (1) in all of "n" points starting from (D•), M8090 (block comparison signal) turns ON.

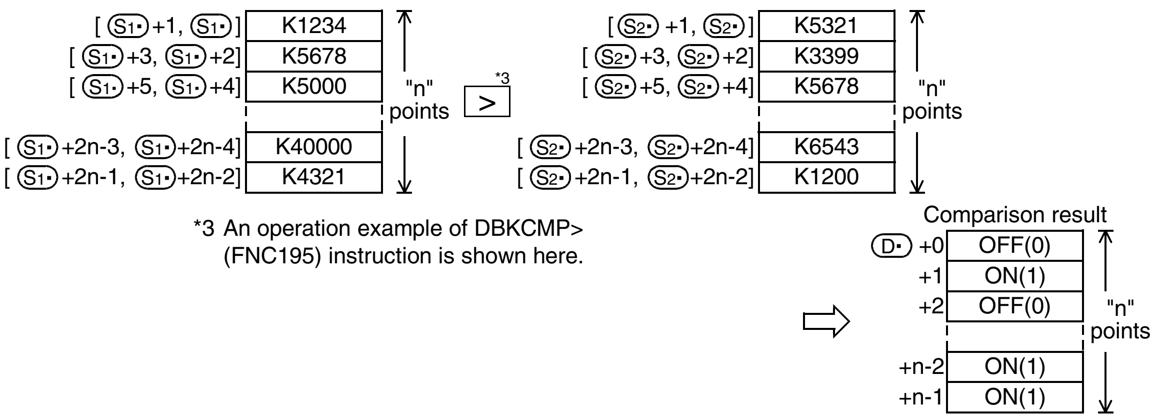
2. 32-bit operation (DBKCMP=, >, <, <>, <=, >= / DBKCMP=P, >P, <P, <>P, <=P, and >=P)

1) "n" 32-bit binary data starting from [S1• +1, S1•] are compared with "n" 32-bit binary data starting from [S2• +1, S2•], and the comparison result is stored in "n" points starting from [D• +1, D•].

Command input

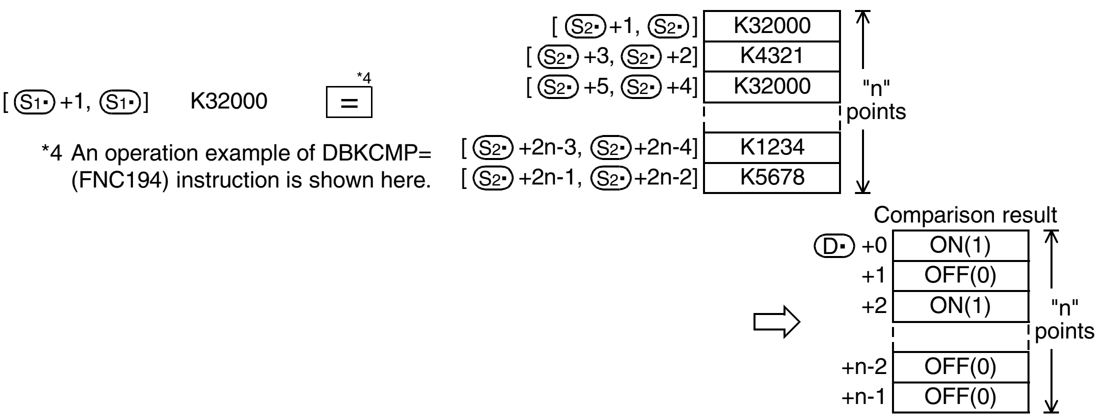
FNCOOO*1 DBKCMP□*2	S1•	S2•	D•	n

*1 The number out of 194 to 199 is put in "OOO".
*2 The symbol out of "=", ">", "<", "<>", "<=", or ">=" corresponding to the FNC No. is put in □.



*3 An operation example of DBKCMP> (FNC195) instruction is shown here.

2) A constant can be directly specified in [S1• +1, S1•].



*4 An operation example of DBKCMP= (FNC194) instruction is shown here.

3) The table below shows the comparison result for each instruction:

Instruction	Comparison result ON (1) condition	Comparison result OFF (0) condition
DBKCMP=(FNC194)	[S1• +1, S1•] = [S2• +1, S2•]	[S1• +1, S1•] ≠ [S2• +1, S2•]
DBKCMP>(FNC195)	[S1• +1, S1•] > [S2• +1, S2•]	[S1• +1, S1•] <= [S2• +1, S2•]
DBKCMP<(FNC196)	[S1• +1, S1•]< [S2• +1, S2•]	[S1• +1, S1•] >= [S2• +1, S2•]
DBKCMP<>(FNC197)	[S1• +1, S1•] ≠ [S2• +1, S2•]	[S1• +1, S1•] = [S2• +1, S2•]
DBKCMP<=(FNC198)	[S1• +1, S1•] <= [S2• +1, S2•]	[S1• +1, S1•] > [S2• +1, S2•]
DBKCMP>=(FNC199)	[S1• +1, S1•] >= [S2• +1, S2•]	[S1• +1, S1•] < [S2• +1, S2•]

4) When the comparison result is ON (1) in all of "n" points starting from [D• +1, D•], the M8090 (block comparison signal) turns ON.

Related device

→ **For the block comparison signal use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8090	Block comparison signal	Turns ON when all comparison results are "ON (1)" in a block data instruction. DBKCMP= (FNC194), DBKCMP> (FNC195), DBKCMP< (FNC196), DBKCMP<> (FNC197), DBKCMP<= (FNC198), and DBKCMP>= (FNC199)

Caution

- When using 32-bit counters and 32-bit high speed counters
  For comparing 32-bit counters and 32-bit high speed counters (C200 to C255), make sure to use an instruction for 32-bit operation (DBKCMP=, DBKCMP>, DBKCMP<, DBKCMP<>, DBKCMP<=, or DBKCMP>=).
  If an instruction for 16-bit operation (BKCMP=, BKCMP>, BKCMP<, BKCMP<>, BKCMP<=, or BKCMP>=) is used, an operation error is caused (error code: K6705).

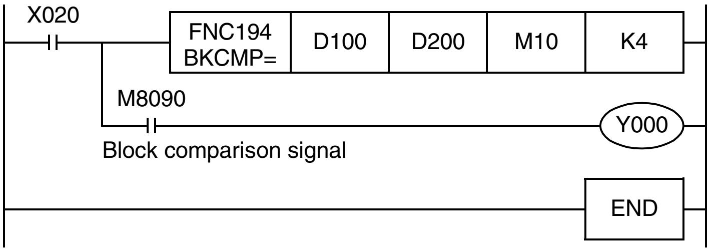
Errors

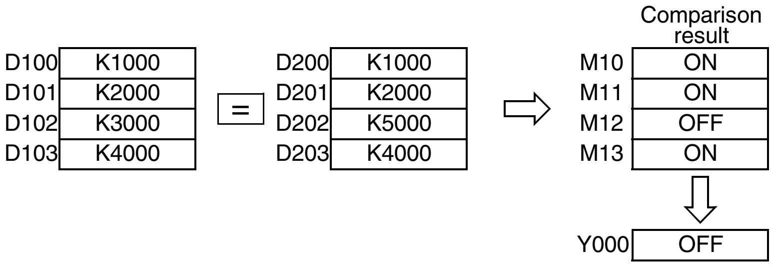
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the range of "n" ("2n" in 32-bit operation) points starting from S1· and/or S2· exceeds the corresponding device range (error code: K6706)
- When the range of "n" points starting from D· exceeds the corresponding device range (error code: K6706)
- When data registers starting from D· specified as "D□.b" overlap "n" ("2n" in 32-bit operation) points starting from S1· (error code: K6706)
- When data registers starting from D· specified as "D□.b" overlap "n" ("2n" in 32-bit operation) points starting from S2· (error code: K6706)
- When a 32-bit counter (C200 to C255) is specified in S1· and/or S2· in 16-bit operation (error code: K6705)
  For comparing 32-bit counters, make sure to use an instruction for 32-bit operation (DBKCMP=, DBKCMP>, DBKCMP<, DBKCMP<>, DBKCMP<=, or DBKCMP>=).

Program example

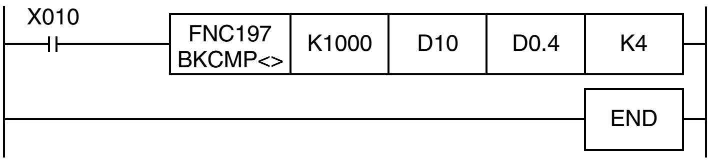
1) In the program shown below, four 16-bit binary data starting from D100 are compared with four 16-bit binary data starting from D200 by BKCMP= (FNC194) instruction when X020 is set to ON, and the comparison result is stored in four points starting from M10.
   When the comparison result is "ON (1)" in all of the four points starting from M10, Y000 is set to ON.

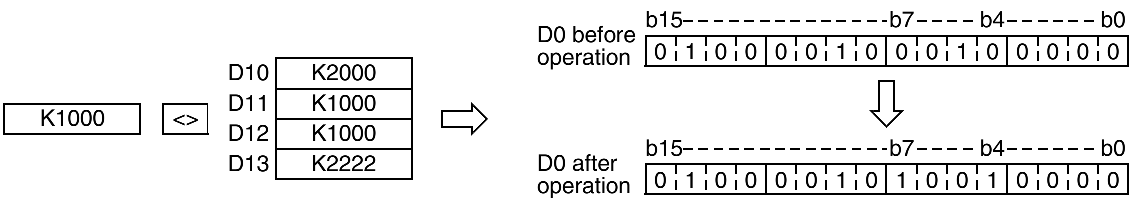




(When all of M10 to M13 are ON, Y000 is set to ON.)

2) In the program shown below, the constant K1000 is compared with four data starting from D10 when X010 is set to ON, and the comparison result is stored in b4 to b7 of D0.





26. Character String Control – FNC200 to FNC209

FNC200 to FNC209 provide instructions for controlling character strings such as linking character string data, replacing some characters and extracting character string data.

FNC No.	Mnemonic	Symbol	Function	Reference
200	STR	STR S1 S2 D	BIN to Character String Conversion	Section 26.1
201	VAL	VAL S D1 D2	Character String to BIN Conversion	Section 26.2
202	$+	$+ S1 S2 D	Link Character Strings	Section 26.3
203	LEN	LEN S D	Character String Length Detection	Section 26.4
204	RIGHT	RIGHT S D n	Extracting Character String Data from the Right	Section 26.5
205	LEFT	LEFT S D n	Extracting Character String Data from the Left	Section 26.6
206	MIDR	MIDR S1 D S2	Random Selection of Character Strings	Section 26.7
207	MIDW	MIDW S1 D S2	Random Replacement of Character Strings	Section 26.8
208	INSTR	INSTR S1 S2 D n	Character string search	Section 26.9
209	$MOV	$MOV S D	Character String Transfer	Section 26.10

26.1 FNC200 – STR / BIN to Character String Conversion

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

This instruction converts binary data into character strings (ASCII codes).
On the other hand, the ESTR (FNC116) instruction converts floating point data into character strings.

→ **For character strings, refer to Section 5.3.**
→ **For ESTR (FNC116) instruction, refer to Section 18.4.**

1. Instruction format

	FNC 200	
D	STR	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	STR	Continuous Operation
	STRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DSTR	Continuous Operation
	DSTRP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S1•	Head device number storing the number of digits of a numeric value to be converted	16-bit binary
S2•	Device number storing binary data to be converted	16- or 32-bit binary
D•	Head device number storing converted character string	Character string

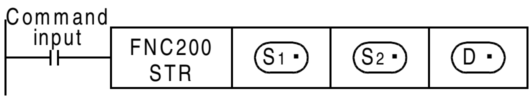
3. Applicable devices

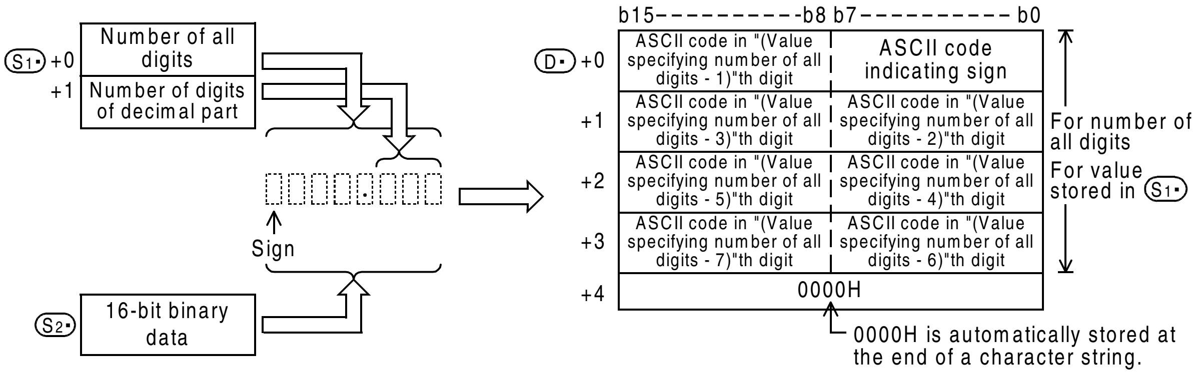
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•												✓	✓	✓	✓				✓					
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D•												✓	✓	✓	✓				✓					

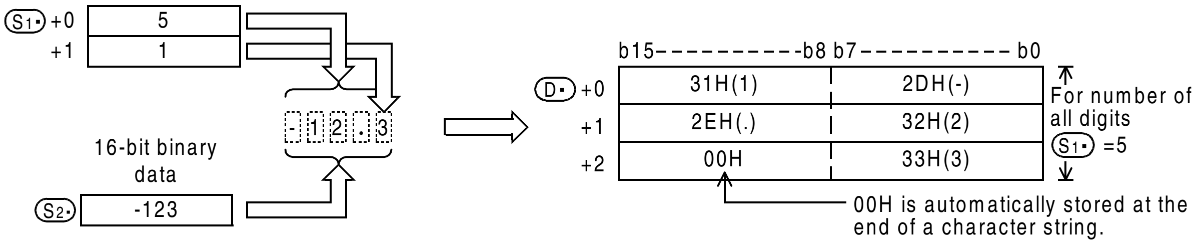
Explanation of function and operation

1. 16-bit operation (STR and STRP)

1) All digits (specified by S1•) of 16-bit binary data stored in S2• are converted into ASCII codes while the decimal point is added to the position specified by the device storing the number of digits of the decimal part (S1•+1), and stored in D• and later.







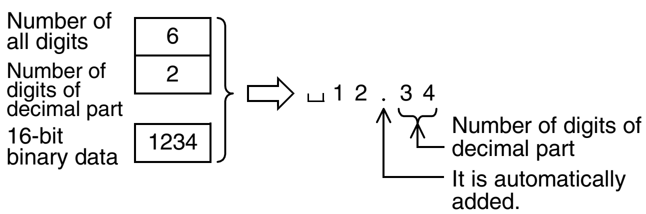
2) Set the number of all digits S1• in the range from 2 to 8.

3) Set the number of digits of the decimal part S1•+1 in the range from 0 to 5.
Make sure to satisfy "Number of digits of decimal part <= (Number of all digits -3)".

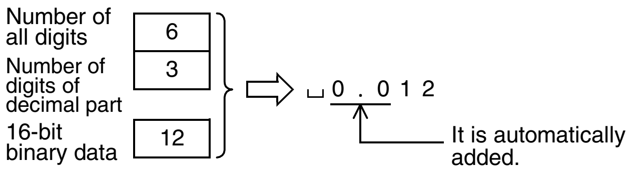
4) 16-bit binary data to be converted stored in S2• should be within the range from -32768 to +32767.

5) Converted character string data is stored in D• and later as shown below.

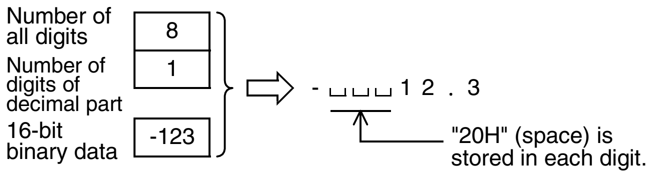
- As the sign, "space" (20H) is stored when the 16-bit binary data stored in S2• is positive, and "-" (2DH) is stored when the 16-bit binary data stored in S2• is negative.
- When the number of digits of the decimal part S1•+1 is set to any value other than "0", the decimal point "." (2EH) is automatically added in "number of digits of decimal part + 1"th digit.
  When the number of digits of the decimal part S1•+1 is set to "0", the decimal point is not added.



- When the number of digits of the decimal part S1•+1 is larger than the number of digits of 16-bit binary data stored in S2•, "0" (30H) is automatically added, and the data is shifted to the right end during conversion.



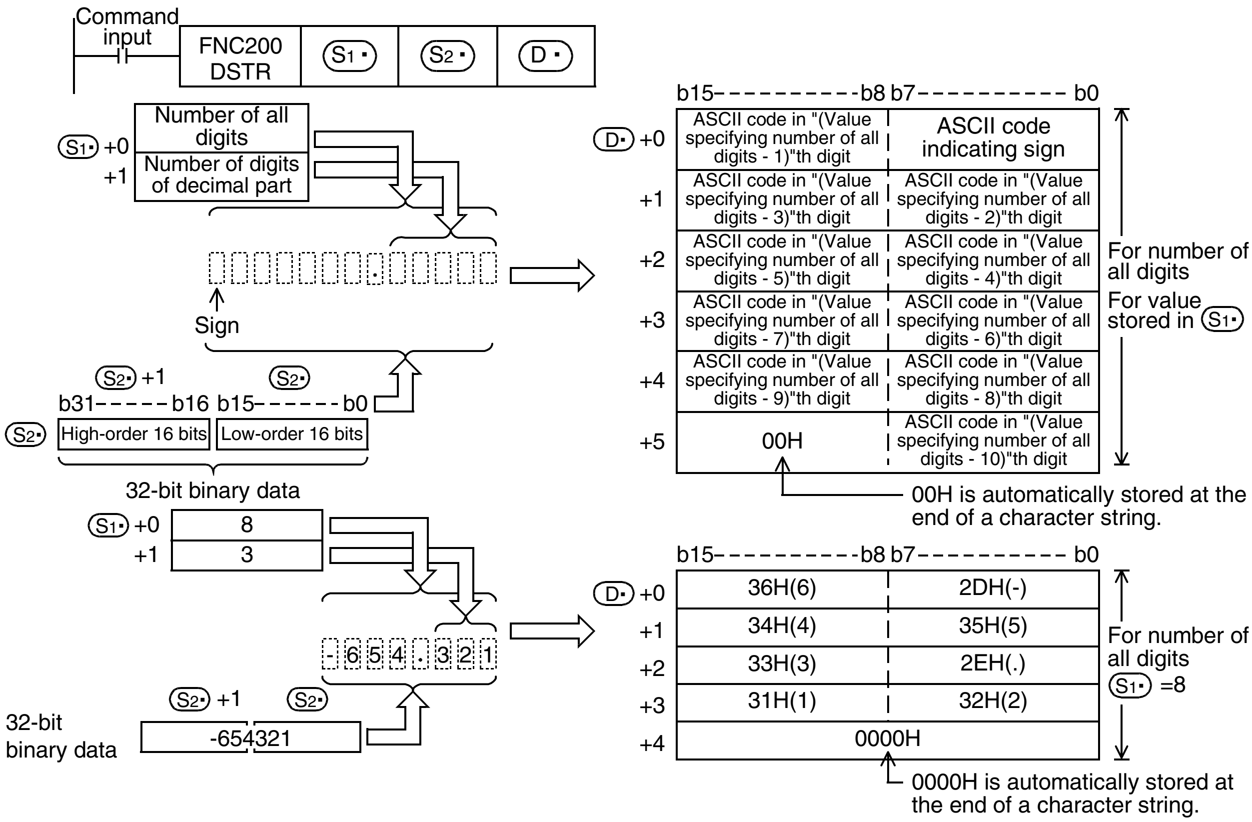
- When the number of all digits stored in S1• excluding the sign and decimal point is larger than the number of digits of 16-bit binary data stored in S2•, "space" (20H) is stored in each digit between the sign and the numeric value.



  When the number of all digits stored in S1• excluding the sign and decimal point is smaller than the number of digits of 16-bit binary data stored in S2•, an error is caused.
- "00H" indicating the end of a character string is automatically stored at the end of a converted character string.
  When the number of all digits is even, "0000H" is stored in the device after the last character.
  When the number of all digits is odd, "00H" is stored in the high-order byte (8 bits) of the device storing the final character.

2. 32-bit operation (DSTR and DSTRP)

1) All digits (specified by S1•) of 32-bit binary data stored in [S2•+1, S2•] are converted into ASCII codes while the decimal point is added to the position specified by the device storing the number of digits of the decimal part (S1•+1), and stored in D• and later.



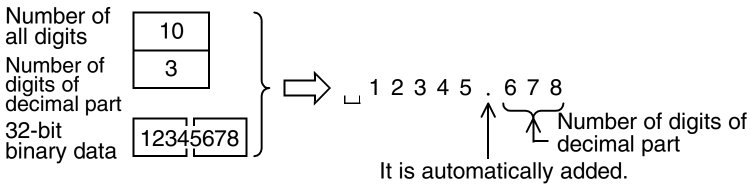
2) Set the number of all digits S1• in the range from 2 to 13.

3) Set the number of digits of the decimal part S1•+1 in the range from 0 to 10.
   Make sure to satisfy "Number of digits of decimal part <= (Number of all digits -3)".

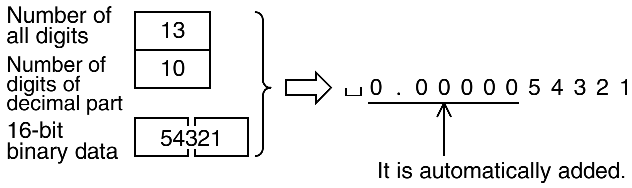
4) 32-bit binary data to be converted stored in [S2•+1, S2•] should be within the range from -2,147,483,648 to +2,147,483,647.

5) Converted character string data is stored in D· and later as shown below.

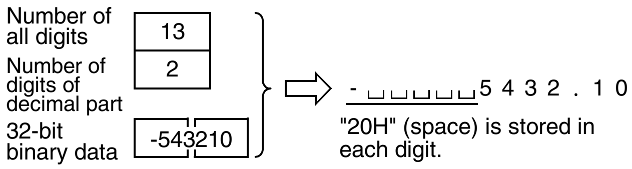
- For the sign, "space" (20H) is stored when the 32-bit binary data stored in S2· is positive, and "-(2DH)" is stored when the 32-bit binary data stored in S2· is negative.
- When the number of digits of the decimal part S1· +1 is set to any value other than "0", the decimal point "." (2EH) is automatically added in "number of digits of decimal part + 1"th digit.
  When the number of digits of the decimal part S1· +1 is set to "0", the decimal point is not added.



- When the number of digits of the decimal part S1· +1 is larger than the number of digits of 32-bit binary data stored in [S2· +1, S2·], "0" (30H) is automatically added, and the data is shifted to the right end during conversion.



- When the number of all digits stored in S1· excluding the sign and decimal point is larger than the number of digits of 32-bit binary data stored in [S2· +1, S2·], "space" (20H) is stored in each digit between the sign and the numeric value.



  When the number of all digits stored in S1· excluding the sign and decimal point is smaller than the number of digits of 32-bit binary data stored in [S2· +1, S2·], an error is caused.
- "00H" indicating the end of a character string is automatically stored at the end of a converted character string.
  When the number of all digits is even, "0000H" is stored in the device after the last character.
  When the number of all digits is odd, "00H" is stored in the high-order byte (8 bits) of the device storing the final character.

Related instructions

Instruction	Description
ESTR(FNC116)	Converts binary floating point data into a character string (ASCII codes) with a specified number of digits.
EVAL(FNC117)	Converts a character string (ASCII codes) into binary floating point data.
VAL(FNC201)	Converts a character string (ASCII codes) into binary data.

Errors

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the number of all digits stored in S1· is outside the following range (error code: K6706)

	Setting range
16-bit operation	2 to 8
32-bit operation	2 to 13

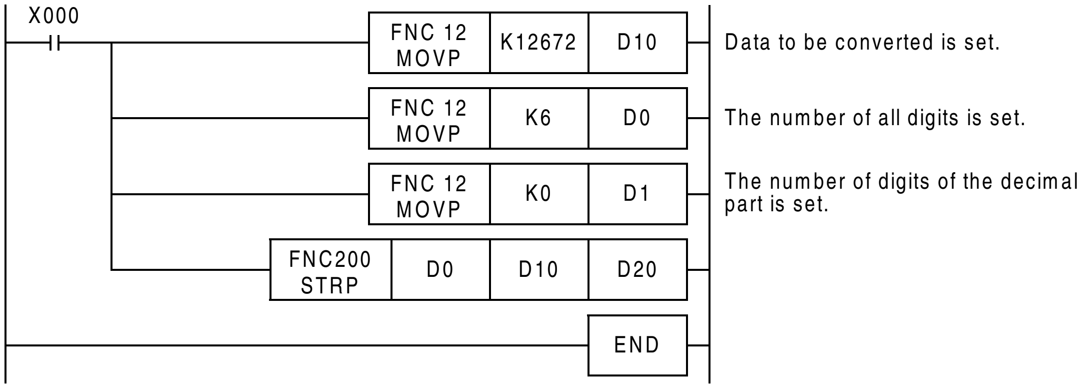
- When the number of digits of the decimal part stored in S1· +1 is outside the following range (error code: K6706)

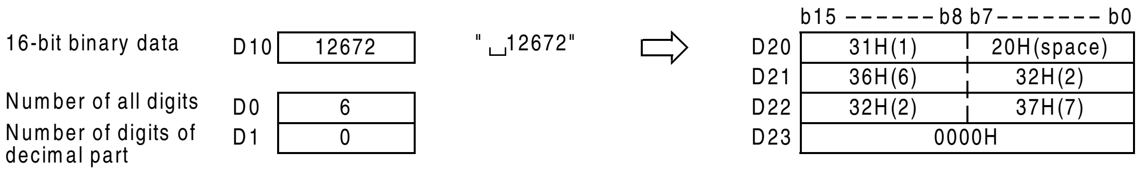
	Setting range
16-bit operation	0 to 5
32-bit operation	0 to 10

- When the relationship between the number of all digits stored in (S1·) and the number of digits of the decimal part stored in (S1·)+1 does not satisfy the following (error code: K6706)
  (Number of all digits -3) ≥ Number of digits of decimal part
- When the number of all digits stored in (S1·) including the digit for sign and the digit for decimal point is smaller than the number of digits of the binary data stored in [(S2·)+1, (S2·)] (error code: K6706)
- When the devices (D·) and later storing a character string exceeds the corresponding device range (error code: K6706)

Program example

In the program below, the 16-bit binary data stored in D10 is converted into a character string in accordance with the digit specification by D0 and D1 when X000 is set to ON, and then stored in D20 to D23.





26.2 FNC201 – VAL / Character String to BIN Conversion

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

This instruction converts a character string (ASCII codes) into binary data.
On the other hand, EVAL (FNC117) instruction converts a character string (ASCII codes) into floating point data.

→ **For character strings, refer to Section 5.3.**
→ **For EVAL (FNC117) instruction, refer to Section 18.5.**

1. Instruction format

	FNC 201	
D	VAL	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	VAL	Continuous Operation
	VALP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DVAL	Continuous Operation
	DVALP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Head device number storing a character string to be converted into binary data	Character string
D1·	Head device number storing the number of all digits of the binary data acquired by conversion	16-bit binary
D2·	Head device number storing the binary data acquired by conversion	16- or 32-bit binary

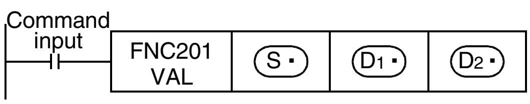
3. Applicable devices

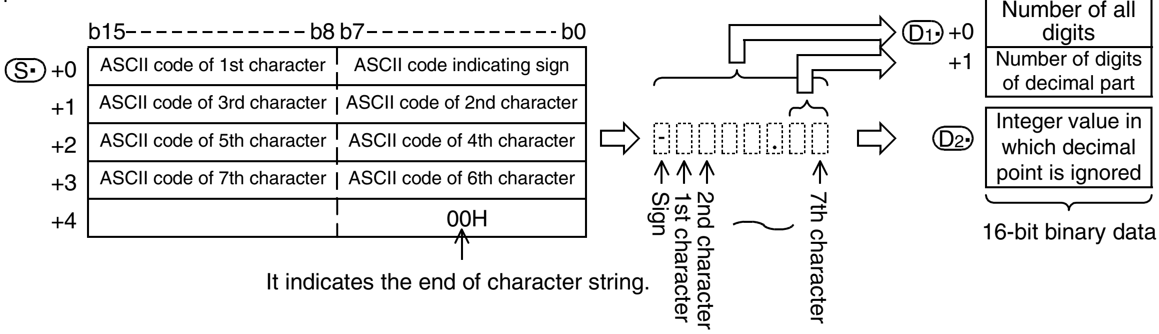
Oper-and Type	Bit Devices System User							Word Devices Digit Specification				System User				Special Unit	Index			Others Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·												✓	✓	✓	✓				✓					
D1·												✓	✓	✓	✓				✓					
D2·									✓	✓	✓	✓	✓	✓	✓	✓			✓					

Explanation of function and operation

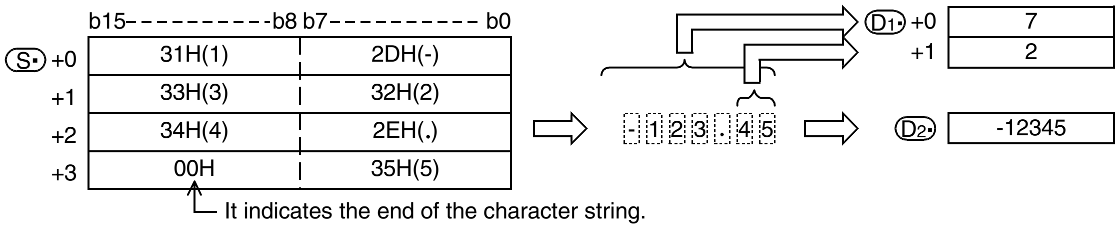
1. 16-bit operation (VAL and VALP)

1) A character string stored in S· and later is converted into 16-bit binary data. The number of all digits of the binary data acquired for conversion is stored in D1·, the number of digits of the decimal part is stored in D1·+1, and the converted binary data is stored in D2·.
In converting a character string into binary data, the data from S· to a device number storing "00H" is handled as a character string in byte units.





For example, when a character string "-123.45" is specified in (S·) and later, the conversion result is stored in (D1·) and (D2·) as shown below.



2) Character string to be converted

a) Number of characters of character string and the numeric range when the decimal point is ignored

	Description
Number of all characters (digits)	2 to 8
Number of characters (digits) of decimal part	0 to 5 and smaller than "number of all digits -3"
Numeric range when decimal point is ignored	–32768 to +32767 Example: 123.45 → 12345

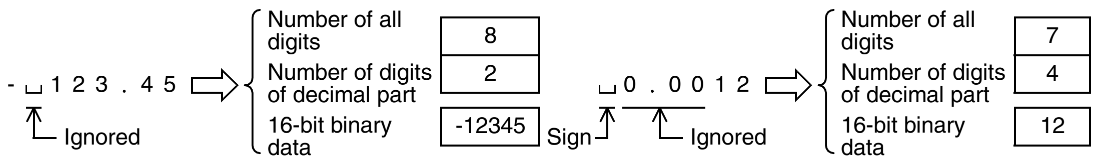
b) Character types used in characters to be converted

		Character type
Sign	Positive numeric value	"Space" (20H)
	Negative numeric value	"–" (2DH)
Decimal point		"." (2EH)
Number		"0" (30H) to "9" (39H)

3) (D1·) stores the number of all digits. The number of all digits indicates the number of all characters (including the number, sign and decimal point).

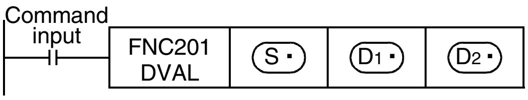
4) (D1·)+1 stores the number of digits of the decimal part. The number of digits of the decimal part indicates the number of all characters after the decimal point "." (2EH).

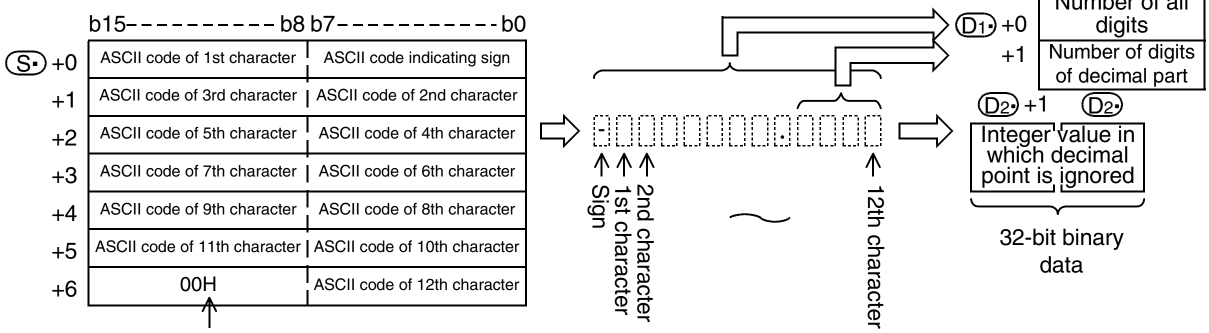
5) (D2·) stores 16-bit data (bin) converted from a character string with the decimalpoint ignored.
In the character string located in (S·) and later, "space" (20H) and "0" (30H) characters between the sign and the first number other than "0" are ignored in the conversion to 16-bit binary data.



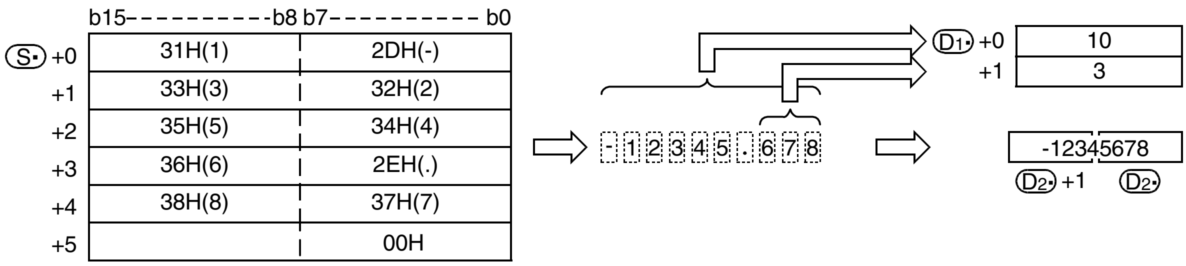
2. 32-bit operation (DVAL and DVALP)

1) A character string stored in (S·) and later is converted into 32-bit binary data. The number of all digits of the binary data acquired for conversion is stored in (D1·), the number of digits of the decimal part is stored in (D1·)+1, and the binary data is stored in [(D2·)+1, (D2·)].
In conversion from a character string into binary data, the data from (S·) to a device number storing "00H" is handled as a character string in byte units.





For example, when a character string "-12345.678" is specified in (S·) and later, the conversion result is stored in (D1·) and (D2·) as shown below.



2) Character string to be converted

a) Number of characters of character string and the numeric range when the decimal point is ignored

	Description
Number of all characters (digits)	2 to 13
Number of characters (digits) of decimal part	0 to 10 and smaller than "number of all digits –3"
Numeric range when decimal point is ignored	–2,147,483,648 to +2,147,483,647 Example: 12345.678 → "12345678"

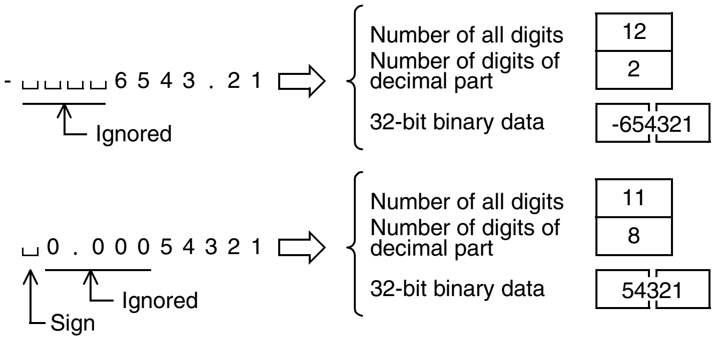
b) Character types used in characters to be converted

		Character type
Sign	Positive numeric value	"Space" (20H)
	Negative numeric value	"–" (2DH)
Decimal point		"." (2EH)
Number		"0" (30H) to "9" (39H)

3) (D1·) stores the number of all digits. The number of all digits indicates the number of all characters (including the number, sign and decimal point).

4) (D1·)+1 stores the number of digits of the decimal part. The number of digits of the decimal part indicates the number of all characters after the decimal point "." (2EH).

5) [D2• +1, D2•] stores 16-bit data (bin) converted from a character string with the decimalpoint ignored. In the character string located in S• and later, "space" (20H) and "0" (30H) characters between the sign and the first number other than "0" are ignored in the conversion to 32-bit binary data.



Related instructions

Instruction	Description
ESTR(FNC116)	Converts binary floating point data into a character string (ASCII codes) with a specified number of digits.
EVAL(FNC117)	Converts a character string (ASCII codes) into binary floating point data.
STR(FNC200)	Converts binary data into a character string (ASCII codes).

Errors

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the number of characters of a character string to be converted (S• and later) is outside the following range (error code: K6706)

	Setting range
16-bit operation	2 to 8
32-bit operation	2 to 13

- When the number of characters of the decimal part of a character string to be converted (S• and later) is outside the following range (error code: K6706)

	Setting range
16-bit operation	0 to 5
32-bit operation	0 to 10

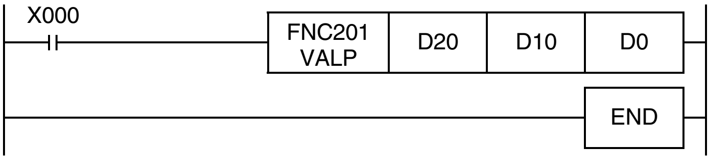
- When the relationship between the number of all characters of a character string to be converted (S• and later) and the number of characters of the decimal part does not satisfy the following (error code: K6706)
  (Number of all characters -3) ≥ Number of characters of decimal part
- When the sign is set to any ASCII code other than "space" (20H) and "-" (2DH) (error code: K6706)
- When a digit of a number is set to any ASCII code other than "0" (30H) to "9" (39H) and decimal point "." (2EH) (error code: K6706)
- When the decimal point "." (2EH) is set two or more times in a character string to be converted (S• and later) (error code: K6706)
- When the binary data acquired by conversion is outside the following range (error code: K6706)

	Setting range
16-bit operation	–32768 to 32767
32-bit operation	–2,147,483,648 to 2,147,483,647

- When "00H" is not present in the location from S• to the final device number (error code: K6706)

Program example

1) In the program below, the character string data stored in D20 to D22 is regarded as an integer value, converted into a binary value, and stored in D0 when X000 is set to ON.

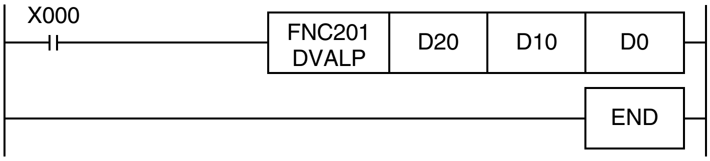


	b15 ------ b8	b7 ------- b0
D20	31H(1)	2DH(-)
D21	2EH(.)	36H(6)
D22	34H(4)	35H(5)
D23		00H

"00H" is stored here.

D0	-1654	
D10	6	Number of all digits
D11	2	Number of digits of decimal part

2) In the program below, the character string data stored in D20 to D24 is regarded as an integer value, converted into a binary value, and stored in D0 when X000 is set to ON.



	b15 ------ b8	b7 ------- b0
D20	37H(7)	20H(space)
D21	31H(1)	39H(9)
D22	30H(0)	30H(0)
D23	36H(6)	2EH(.)
D24	31H(1)	31H(1)
D25		00H

"00H" is stored here.

	D1	D0
D0	7910	0611

D10	10	Number of all digits
D11	3	Number of digits of decimal part

26.3 FNC202 – $+ / Link Character Strings

FX3UC Ver.1.00→ FX3U Ver.2.20→

Outline

This instruction links a character string to another character string.

→ For handling of character strings, refer to Section 5.3.

1. Instruction format

FNC 202 $+ P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	$+	Continuous Operation
	$+P	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
S1•	Head device number storing the link source data (character string) or directly specified character string	Character string
S2•	Head device number storing the link data (character string) or directly specified character string	
D•	Head device number storing the linked data (character string)	

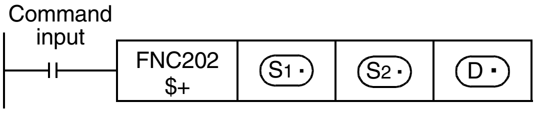
3. Applicable devices

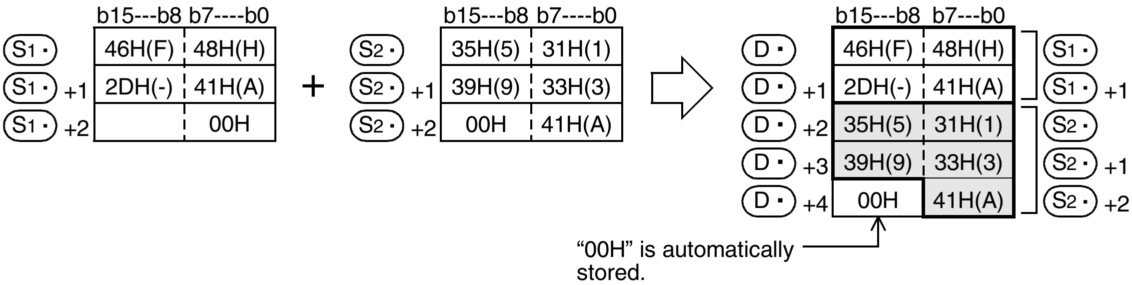
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓				✓	
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓				✓	
D•									✓	✓	✓	✓	✓	✓	✓	✓			✓					

Explanation of function and operation

1. 16-bit operation ($+ and $+P)

The character string data stored in S2• and later is linked to the end of the character string data stored in S1• and later, and the linked data is stored to devices starting from D•.
A character string stored in S1• or S2• or later indicates the data from the specified device to the first "00H" in units of byte.





- In linking, "00H" indicating the end of a character string specified in S1• is ignored, and a character string specified in S2• is linked to the last character specified in S1•.
  When a character string is linked, "00H" is automatically added at the end.
  - When the number of characters after linking is odd, "00H" is stored in the high-order byte of the device storing the last character.
  - When the number of characters after linking is even, "0000H" is stored in the device after the last character.

Cautions

- When directly specifying a character string, up to 32 characters can be specified (input).
  However, this limitation in the number of characters is not applied when a word device is specified in S1• or S2•.
- When the values in both S1• and S2• start from "00H" (that is, when the number of characters is "0"), "0000H" is stored in D•.

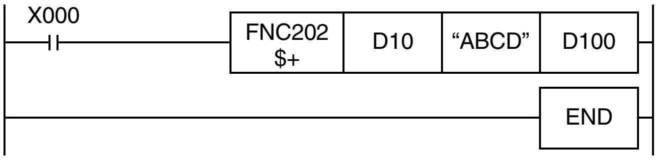
Errors

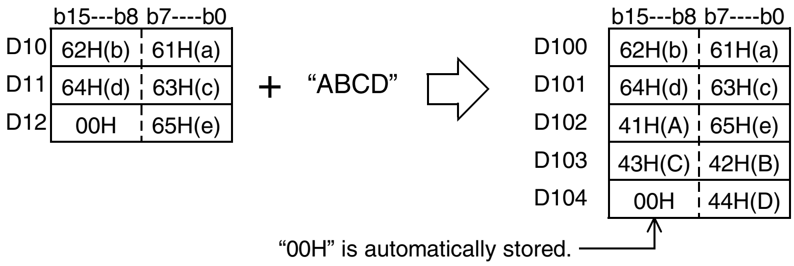
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the number of devices after a device number specified by D• is smaller than the number of devices required to store all linked character strings (that is, when "00H" cannot be stored after all character strings and the last character) (error code: K6706)
- When the same device is specified in S1•, S2• and D• as a device for storing a character string (error code: K6706)
- When "00H" is not set within the corresponding device range after the device specified by S1• or S2• (error code: K6706)

Program example

In the program example shown below, a character string stored in D10 to D12 (abcde) is linked to the character string "ABCD", and the result is stored to D100 and later when X000 turns ON.





26.4 FNC203 – LEN / Character String Length Detection

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction detects the number of characters (bytes) of a specified character string.

→ **For handling of character strings, refer to Section 5.3.**

1. Instruction format

FNC 203 LEN	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	LEN	Continuous Operation
	LENP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
(S·)	Head device number storing a character string whose length is to be detected	Character string
(D·)	Device number storing the detected character string length (number of bytes)	16-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index				Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
(D·)									✓	✓	✓	✓	✓	✓	✓	✓			✓					

Explanation of function and operation

1. 16-bit operation (LEN and LENP)

The length of a character string stored in (S·) and later is detected, and stored to (D·). Data starting from (S·) until the first device storing "00H" is handled as a character string in units of byte.

Command input —| |— [FNC203 LEN | (S·) | (D·)]

	b15----------b8	b7-------------b0
(S·)	2nd character	1st character
(S·) +1	4th character	3rd character
(S·) +2	6th character	5th character
	⁓	
(S·) +n	00H	"n"th character

↑ It indicates the end of character string.

⇒ (D·) b15---------------b0: Character string length

For example, when "ABCDEFGHI" is stored in (S·) and later as shown below, K9 is stored to (D·).

	b15----------b8	b7-------------b0
(S·)	42H(B)	41H(A)
(S·) +1	44H(D)	43H(C)
(S·) +2	46H(F)	45H(E)
(S·) +3	48H(H)	47H(G)
(S·) +4	00H	49H(I)

"ABCDEFGHI" ⇒ (D·) b15---------------b0: 9

Caution

- This instruction can handle character codes other than ASCII codes, but the character string length is handled in byte units (8 bits). Accordingly, in the case of character codes in which 2 bytes express 1 character such as shift JIS codes, the length of 1 character is detected as "2".

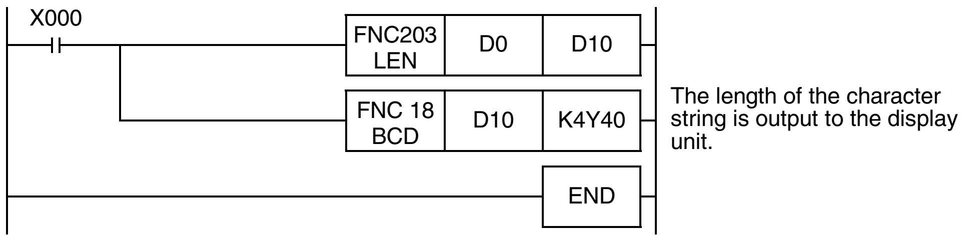
Errors

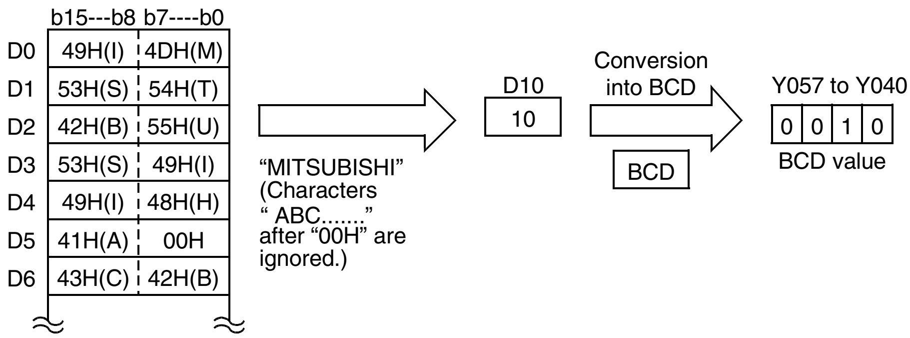
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When "00H" is not set within the corresponding device range after a device specified by S· (error code: K6706)
- When the detected number of characters is "32768" or more (error code: K6706)

Program example

In the program example shown below, the length of a character string stored in D0 and later is output in 4-digit BCD to Y040 to Y057 when X000 turns ON.





26.5 FNC204 – RIGHT / Extracting Character String Data from the Right

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction extracts a specified number of characters from the right end of a specified character string.

→ **For handling of character strings, refer to Section 5.3.**

1. Instruction format

FNC 204 RIGHT P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	RIGHT	Continuous Operation
	RIGHTP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
(S·)	Head device number storing a character string	Character string
(D·)	Head device number storing extracted character string	Character string
n	Number of characters to be extracted	16-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
(D·)									✓	✓	✓	✓	✓	✓	✓	✓			✓					
n														✓	✓					✓	✓			

Explanation of function and operation

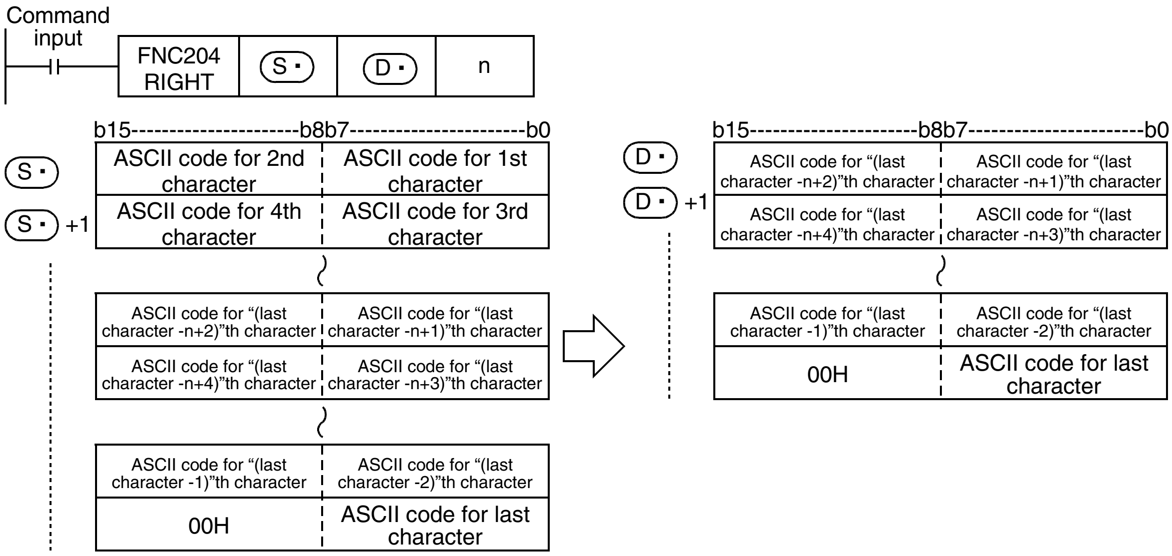
1. 16-bit operation (RIGHT and RIGHTP)

"n" characters are extracted from the right end (that is, from the end) of the character string data stored in (S·) and later, and stored to (D·) and later.

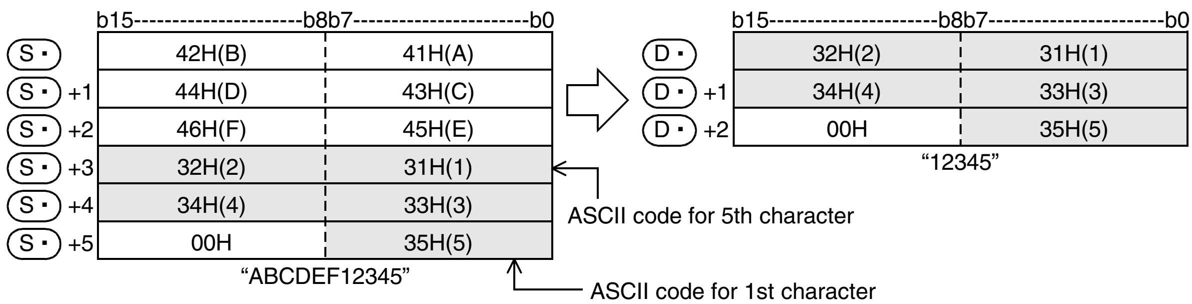
If the number of characters specified by "n" is "0", the NULL code (0000H) is stored to (D·).

When characters are extracted from a character string, "00H" is automatically added at the end of the extracted characters.

- When the number of extracted characters is odd, "00H" is stored in the high-order byte of a device storing the last character.
- When the number of extracted characters is even, "0000H" is stored in the device after the last character.



In the case of "n = 5"



- A character string stored in (S·) and later indicates data stored in devices from the specified device until "00H" is first detected in byte units.

Cautions

When handling character codes other than ASCII codes, note the following contents:

- The number of characters is handled in byte units (8 bits). Accordingly, in the case of character codes in which 2 bytes express 1 character such as shift JIS codes, the length of 1 character is detected as "2".
- When extracting characters from a character string including character codes in which 2 bytes express 1 character such as shift JIS codes, consider the number of characters to be extracted in units of character codes for 1 character.
  Note that the expected character code is not given if only 1 byte is executed out of a 2-byte character code.

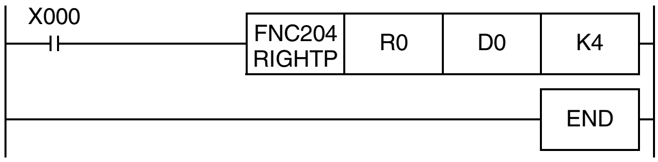
Errors

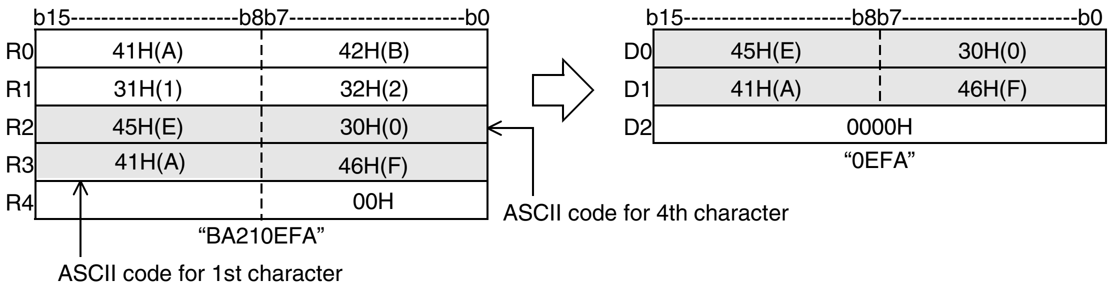
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When "00H" is not set within the corresponding device range after a device specified by (S·) (error code: K6706)
- When "n" exceeds the number of characters specified by (S·) (error code: K6706)
- When the number of devices after a device number specified by (D·) is smaller than the number of devices required to store extracted "n" characters (that is, when "00H" cannot be stored after all character strings and the last character) (error code: K6706)
- When "n" is a negative value (error code: K6706)

Program example

In the program example shown below, 4 characters are extracted from the right end of the character string data stored in R0 and later, and stored to D0 and later when X000 turns ON.





26.6 FNC205 – LEFT / Extracting Character String Data from the Left

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction extracts a specified number of characters from the left end of a specified character string.

→ **For handling of character strings, refer to Section 5.3.**

1. Instruction format

FNC 205 LEFT	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	LEFT	Continuous Operation
	LEFTP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
(S·)	Head device number storing a character string	Character string
(D·)	Head device number storing extracted character string	Character string
n	Number of characters to be extracted	16-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
(D·)									✓	✓	✓	✓	✓	✓	✓	✓			✓					
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (LEFT and LEFTP)

"n" characters are extracted from the left end (that is, from the head) of the character string data stored in S· and later and stored to D· and later.

If the number of characters specified by "n" is "0", the NULL code (0000H) is stored to D·.

When characters are extracted from a character string, "00H" is automatically added at the end of the extracted characters.

- When the number of extracted characters is odd, "00H" is stored in the high-order byte of a device storing the last character.
- When the number of extracted characters is even, "0000H" is stored in the device after the last character.

Command input

FNC205 LEFT	S·	D·	n

Source (S·):

	b15--------b8	b7--------b0
S·	ASCII code for 2nd character	ASCII code for 1st character
S· +1	ASCII code for 4th character	ASCII code for 3rd character
⋮	⋮	⋮
	ASCII code for n-1 character	ASCII code for n-2 character
	ASCII code for n+1 character	ASCII code for n character
⋮	⋮	⋮
	00H	ASCII code for last character

Destination (D·):

	b15--------b8	b7--------b0
D·	ASCII code for 2nd character	ASCII code for 1st character
D· +1	ASCII code for 4th character	ASCII code for 3rd character
⋮	⋮	⋮
	ASCII code for n-1 character	ASCII code for n-2 character
	00H	ASCII code for n character

In the case of "n = 7"

	b15--------b8	b7--------b0
S·	42H(B)	41H(A)
S· +1	44H(D)	43H(C)
S· +2	46H(F)	45H(E)
S· +3	32H(2)	31H(1)
S· +4	34H(4)	33H(3)
S· +5	00H	35H(5)

"ABCDEF12345"

ASCII code for 1st character

ASCII code for 7th character

	b15--------b8	b7--------b0
D·	42H(B)	41H(A)
D· +1	44H(D)	43H(C)
D· +2	46H(F)	45H(E)
D· +3	00H	31H(1)

"ABCDEF1"

- A character string stored in S· and later indicates data stored in devices from the specified device until "00H" is first detected in byte units.

Cautions

When handling character codes other than ASCII codes, note the following contents:

- The number of characters is handled in byte units (8 bits). Accordingly, in the case of character codes in which 2 bytes express 1 character such as shift JIS codes, the length of 1 character is detected as "2".
- When extracting characters from a character string including character codes in which 2 bytes express 1 character such as shift JIS codes, consider the number of characters to be extracted in units of character codes for 1 character.
  Note that the expected character code is not given if only 1 byte is executed out of a 2-byte character code.

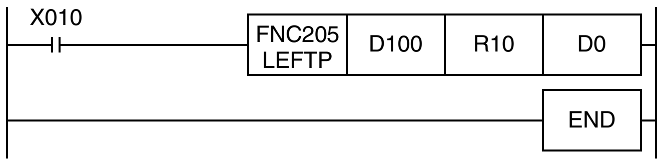
Errors

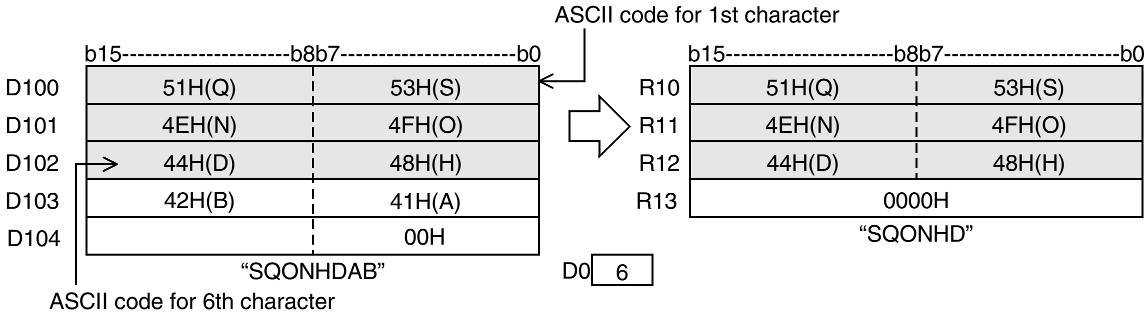
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When “00H” is not set within the corresponding device range after a device specified by (S·) (error code: K6706)
- When “n” exceeds the number of characters specified by (S·) (error code: K6706)
- When the number of devices after a device number specified by (D·) is smaller than the number of devices required to store extracted “n” characters (that is, when “00H” cannot be stored after all character strings and the last character) (error code: K6706)
- When “n” is a negative value (error code: K6706)

Program example

In the program example shown below, the number of characters which is equivalent to the number stored in D0 are extracted from the left end of the character string data stored in D100 and later, and stored to R10 and later when X010 turns ON.





26.7 FNC206 – MIDR / Random Selection of Character Strings

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction extracts a specified number of characters from arbitrary positions of a specified character string.

→ **For handling of character strings, refer to Section 5.3.**

1. Instruction format

FNC 206 MIDR P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	MIDR	Continuous Operation
	MIDRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
S1•	Head device number storing a character string	Character string
D•	Head device number storing extracted character string	Character string
S2•	Head device number specifying the head position and number of characters to be extracted S2• : Head character position S2• +1 : Number of characters	16-bit binary

3. Applicable devices

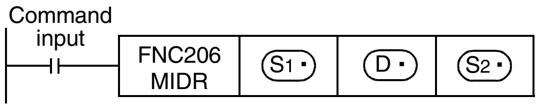
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
D•									✓	✓	✓	✓	✓	✓	✓	✓			✓					
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					

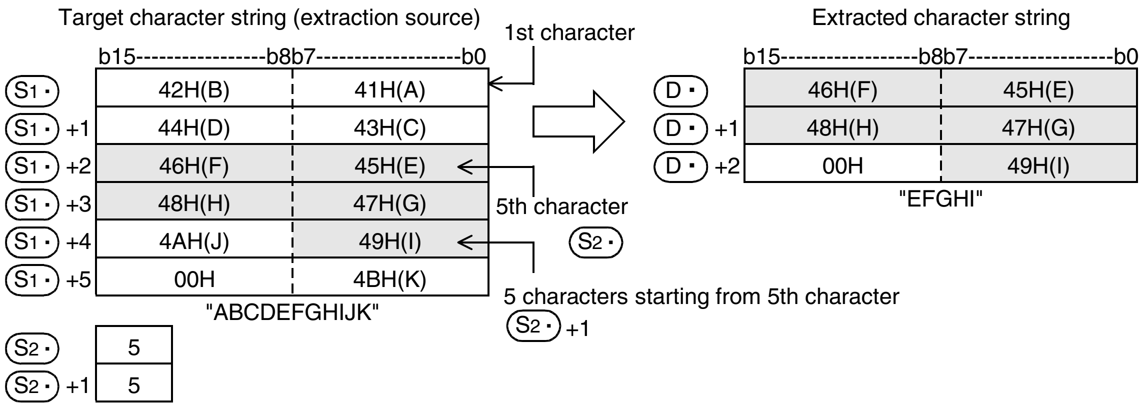
Explanation of function and operation

1. 16-bit operation (MIDR and MIDRP)

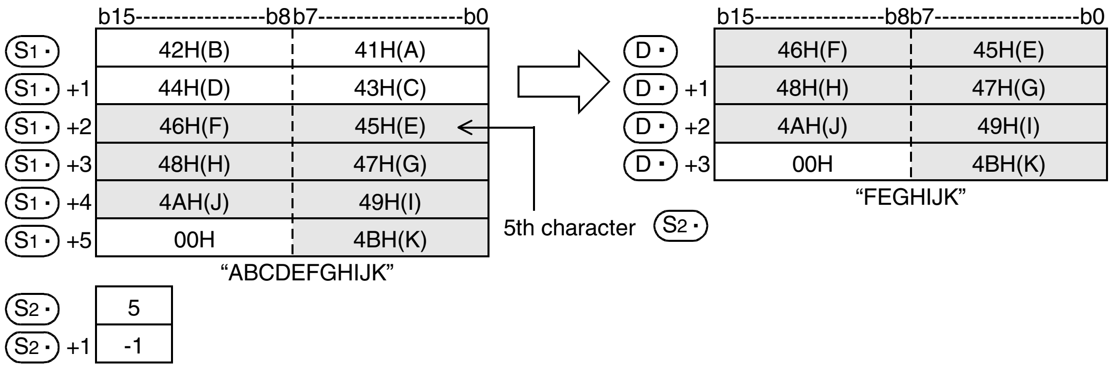
"S2• +1" characters are extracted leftward from the position specified by S2• of the character string data stored in S1• and later, and stored to D• and later.
When characters are extracted from a character string, "00H" is automatically added at the end of the extracted characters.

- When the number of extracted characters specified by S2• +1 is odd, "00H" is stored in the high-order byte of a device storing the last character.
- When the number of extracted characters specified by S2• +1 is even, "0000H" is stored in the device after the last character.





- A character string stored in S1• and later indicates data stored in devices from the specified device until "00H" is first detected in units of byte.
- When the number of characters to be extracted specified by S2• +1 is "0", the extraction processing is not executed.
- When the number of characters to be extracted specified by S2• +1 is "-1", the entire character string stored in S1• and later is stored to D• and later.



Cautions

When handling character codes other than ASCII codes, note the following contents:

- The number of characters is handled in byte units (8 bits). Accordingly, in the case of character codes in which 2 bytes express 1 character such as shift JIS codes, the length of 1 character is detected as “2”.
- When extracting characters from a character string including character codes in which 2 bytes express 1 character such as shift JIS codes, consider the number of characters to be extracted in units of character codes for 1 character.
  Note that the expected character code is not given if only 1 byte is executed out of a 2-byte character code.

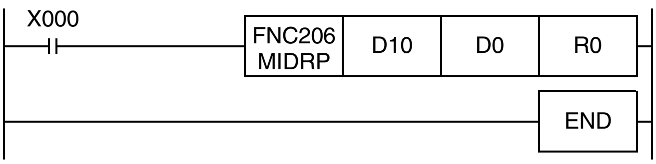
Errors

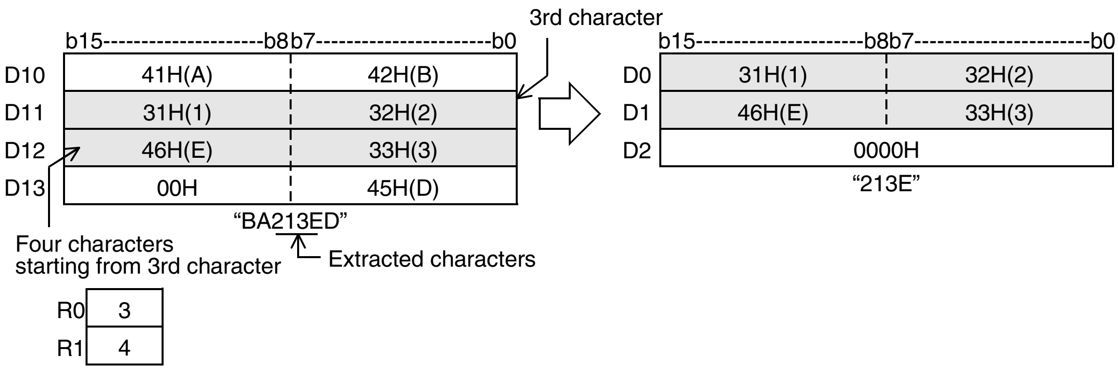
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When “00H” is not set within the corresponding device range after a device specified by (S1·) (error code: K6706)
- When the value specified by (S2·)+1 exceeds the number of characters specified by (S1·) (error code: K6706)
- When the number of characters specified by (S2·)+1 from the position specified by (D·) exceeds the device range specified by (D·) (error code: K6706)
- When the number of devices after a device number specified by (D·) is smaller than the number of devices required to store extracted characters as many as the number specified by (S2·)+1 (that is, when “00H” cannot be stored after all character strings and the last character) (error code: K6706)
- When (S2·) specifies a negative value (error code: K6706)
- When (S2·)+1 specifies “-2” or less (error code: K6706)
- When (S2·)+1 specifies a number larger than the number of characters specified by (S1·) (error code: K6706)

Program example

In the program example shown below, four characters are extracted from the 3rd character from the left end of the character string data stored in D10 and later, and then stored to D0 and later when X000 turns ON.





26.8 FNC207 – MIDW / Random Replacement of Character Strings

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction replaces the characters in arbitrary positions inside designated character string with a specified character string.

→ **For handling of character strings, refer to Section 5.3.**

1. Instruction format

FNC 207
MIDW P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	MIDW	Continuous Operation
	MIDWP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
S1•	Head device number storing a character string used in overwriting	Character string
D•	Head device number storing character string to be overwritten	Character string
S2•	Head device number specifying the head position and number of characters to be overwritten S2• : Head character position to be overwritten S2• +1 : Number of characters to be overwritten	16-bit binary

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
D•									✓	✓	✓	✓	✓	✓	✓	✓			✓					
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					

Explanation of function and operation

1. 16-bit operation (MIDW and MIDWP)

"S2· +1" characters are extracted from the left end (that is, the head) of the character string data stored in S1· and later, and stored to the position specified by S2· and later of the character string data stored in D· and later.

Command input

FNC207 MIDW	S1·	D·	S2·

	b15--------------b8	b7----------------b0	
S1·	31H(1)	30H(0)	← 1st character
S1· +1	33H(3)	32H(2)	
S1· +2	35H(5)	34H(4)	
S1· +3	37H(7)	36H(6)	
S1· +4	00H	38H(8)	

"012345678"

6th character

S2·	3	Position from the left end in the character string stored in D· and later
S2· +1	6	Number of characters from the left end in the character string stored in S1· and later

Before execution

	b15--------------b8	b7----------------b0	
D·	42H(B)	41H(A)	← 1st character
D· +1	44H(D)	43H(C)	← 3rd character
D· +2	46H(F)	45H(E)	
D· +3	48H(H)	47H(G)	
D· +4	00H	49H(I)	

"ABCDEFGHI"

After execution

	b15--------------b8	b7----------------b0	
D·	42H(B)	41H(A)	
D· +1	31H(1)	30H(0)	← 3rd character
D· +2	33H(3)	32H(2)	
D· +3	35H(5)	34H(4)	
D· +4	00H	49H(I)	

"AB012345I"

Six characters starting from 3rd character (overwritten character string)

- The character string stored in S1· and later or D· and later indicates data stored in devices from the specified device until "00H" is first detected in byte units.
- When the number of characters to be overwritten specified by S2· +1 is "0", the overwriting processing is not executed.
- When the number of characters to be overwritten specified by S2· +1 exceeds the last character of the character string stored in D· and later, data is stored up to the last character.

	b15--------------b8	b7----------------b0	
S1·	31H(1)	30H(0)	← 1st character
S1· +1	33H(3)	32H(2)	
S1· +2	35H(5)	34H(4)	
S1· +3	37H(7)	36H(6)	
S1· +4	00H	38H(8)	

"012345678"

8th character

S2·	5	Position from the left end in the character string stored in D· and later
S2· +1	8	Number of characters from the left end in the character string stored in S1· and later

Before execution

	b15--------------b8	b7----------------b0	
D·	42H(B)	41H(A)	
D· +1	44H(D)	43H(C)	
D· +2	46H(F)	45H(E)	← 5th character
D· +3	48H(H)	47H(G)	
D· +4	00H	49H(I)	

"ABCDEFGHI"

After execution

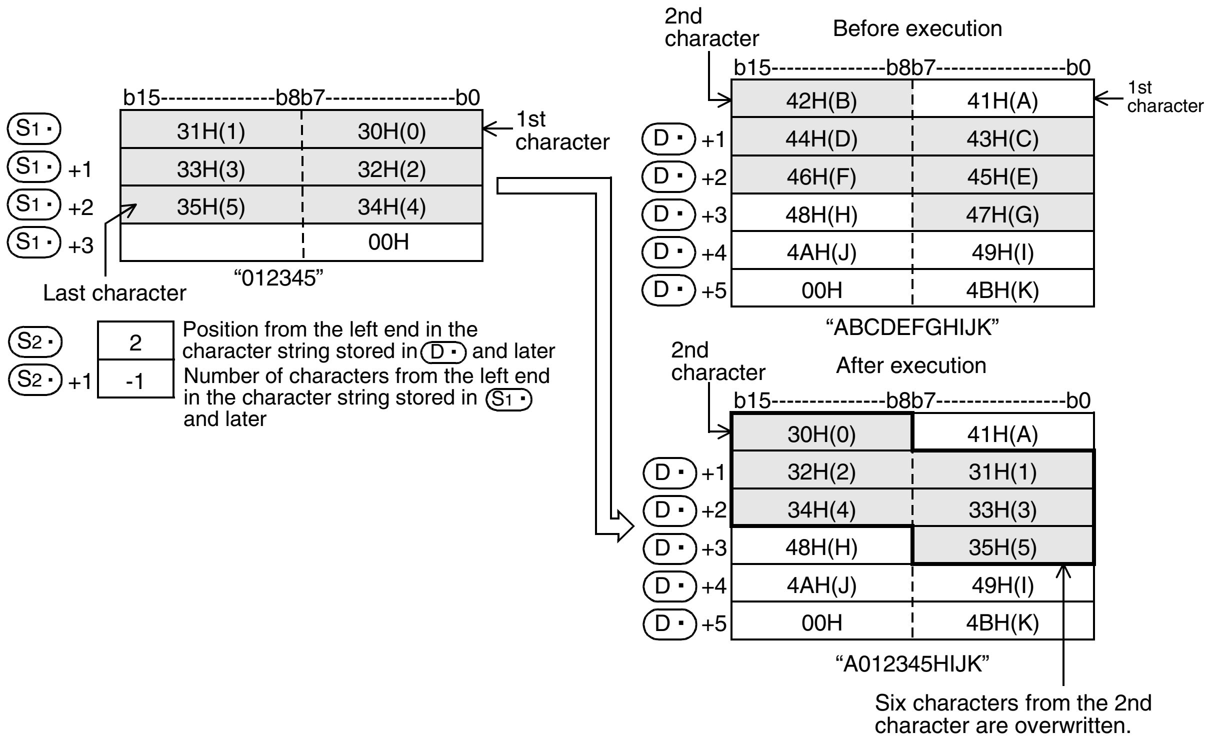
	b15--------------b8	b7----------------b0	
D·	42H(B)	41H(A)	
D· +1	44H(D)	43H(C)	
D· +2	31H(1)	30H(0)	← 5th character
D· +3	33H(3)	32H(2)	
D· +4	00H	34H(4)	

"ABCD01234"

Characters from the 5th character to the last character are overwritten. Excessive characters, "35H (5)" to "37H (7)", are not stored.

21 FNC160-FNC169 Real Time Clock Control
22 FNC170-FNC179 External Device
23 FNC180 Alternate Instructions
24 FNC181-FNC189 Others
25 FNC190-FNC199 Block Data Operation
26 FNC200-FNC209 Character String Control
27 FNC210-FNC219 Data Operation 3
28 FNC220-FNC249 Data Comparison
29 FNC250-FNC269 Data Table Operation
30 FNC270-FNC274 Ex-Device Inverter Comms

- When (S2•)+1 (the number of characters to be extracted) is “-1”, the entire character string stored in (S1•) and later is stored to (D•) and later.



Cautions

This instruction can handle character codes other than ASCII codes, but please note the following:

- The number of characters is handled in byte units (8 bits). Accordingly, in the case of character codes in which 2 bytes express 1 character such as shift JIS codes, the length of 1 character is detected as “2”.
- When overwriting a character string including character codes in which 2 bytes express 1 character such as shift JIS codes, consider the number of characters to be extracted in units of character codes for 1 character.
  Note that the expected character code is not given if only 1 byte is overwritten out of a 2-byte character code.

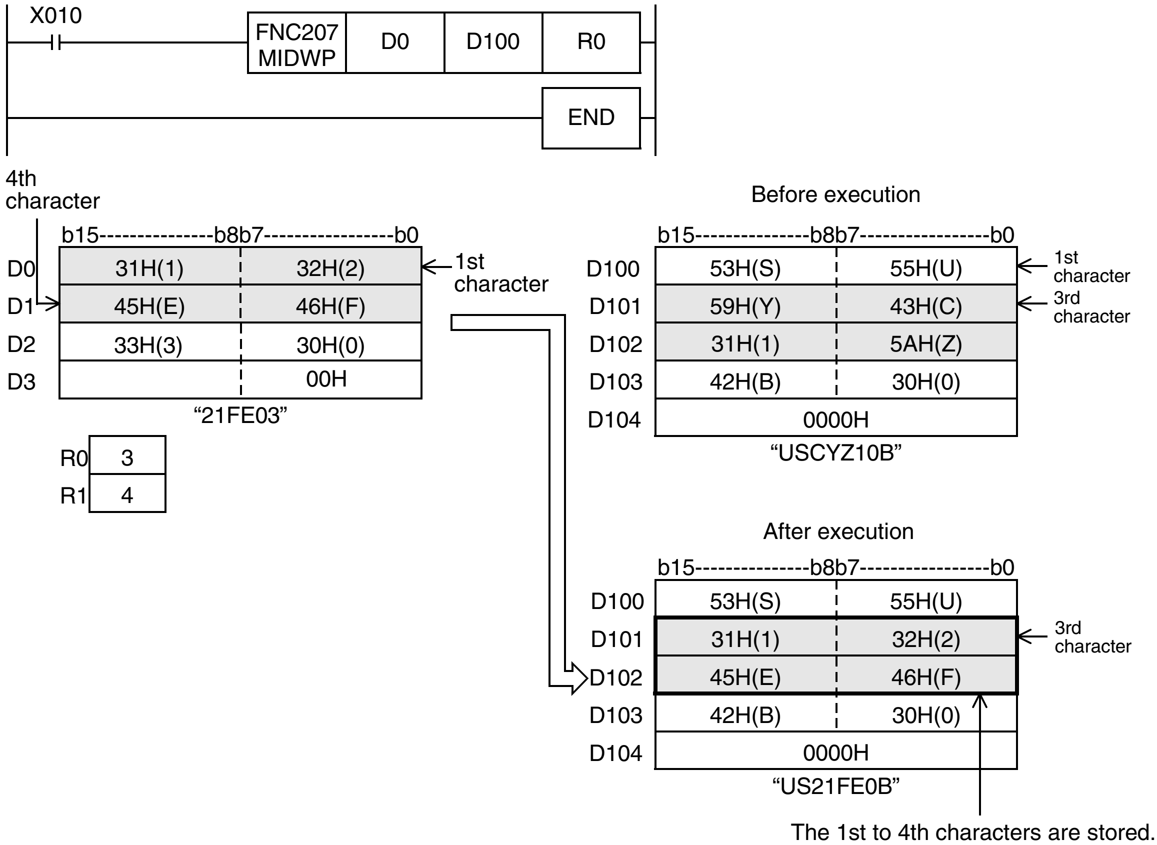
Errors

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When “00H” is not set within the corresponding device range after a device specified by (S1•) or (D•) (error code: K6706)
- When the value specified by (S2•) exceeds the number of characters of the character string stored in (D•) and later (error code: K6706)
- When the number of characters specified by (S2•)+1 exceeds the number of characters specified by (S1•) (error code: K6706)
- When (S2•) specifies a negative value (error code: K6706)
- When (S2•)+1 specifies “-2” or less (error code: K6706)

Program example

In the program example shown below, 4 characters are extracted from the character string data stored in D0 and later, and stored to the 3rd character (from the left end) and later for the character string data stored in D100 and later when X010 turns ON.



26.9 FNC208 – INSTR / Character string search

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

This instruction searches a specified character string within another character string.

1. Instruction format

FNC 208 INSTR P	16-bit Instruction	Mnemonic	Operation Condition	32-bit Instruction	Mnemonic	Operation Condition
	9 steps	INSTR	Continuous Operation		—	
		INSTRP	Pulse (Single) Operation		—	

2. Set data

Operand Type	Description	Data Type
(S1•)	Head device number storing a character string	Character string
(S2•)	Head device number storing a character string to be searched	Character string
(D•)	Head device number storing search result	16-bit binary
n	Search start position	16-bit binary

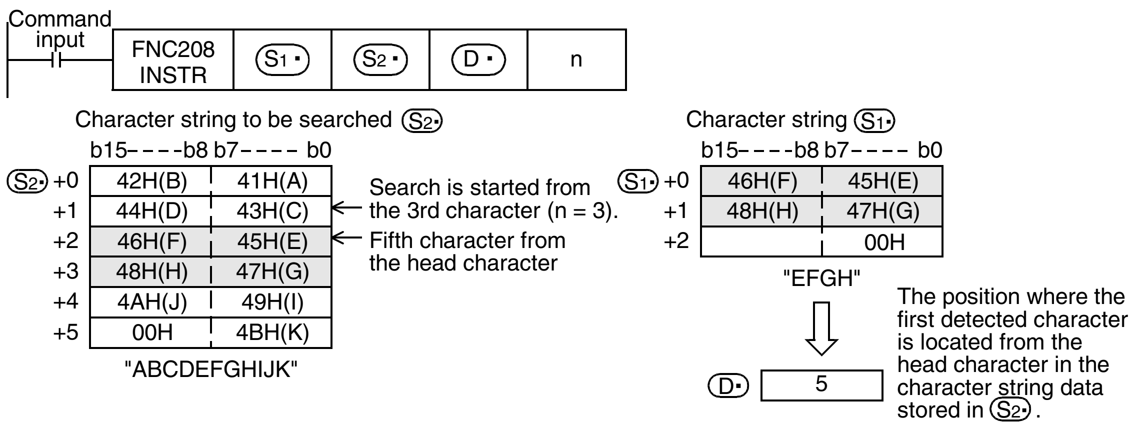
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index				Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify		K	H	E	"□"	P
(S1•)												✓	✓	✓	✓				✓					✓	
(S2•)												✓	✓	✓	✓				✓						
(D•)												✓	✓	✓	✓				✓						
n														✓	✓						✓	✓			

Explanation of function and operation

1. 16-bit operation (INSTR and INSTRP)

1) The character string stored in (S1•) and higher is searched for within the character string (S2•) and higher. The search begins at the "n"th character from the left end (head character) of (S2•) and the search result is stored in (D•). The search result provides the first matching character (located from the left end (head character)) in (S2•).



2) When the searched character string is not detected, "0" is stored in (D•).

3) When the search start position "n" is a negative number or "0", search processing is not executed.

4) A character string can be directly specified in the character string (S1·).

Character string to be searched (S2·)

	b15----b8	b7---- b0	
(S2·) +0	32H(2)	31H(1)	
+1	34H(4)	33H(3)	← Search is started from the 3rd character (n = 3).
+2	42H(B)	41H(A)	← Fifth character from the head character
+3	36H(6)	35H(5)	
+4	42H(B)	41H(A)	
+5		00H	

"1234AB56AB"

Character string (S1·)
"AB"
⇩
D100 | 5

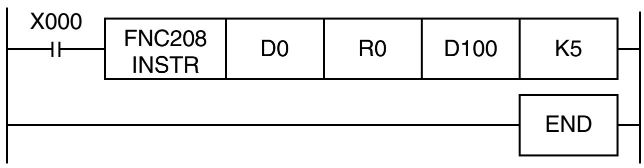
Errors

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the search start position "n" exceeds the number of characters stored in (S2·) (error code: K6706)
- When "00H (NULL)" is not located within the corresponding device range starting from (S1·) (error code: K6706)
- When "00H (NULL)" is not located within the corresponding device range starting from (S2·) (error code: K6706)

Program example

1) In the program example below, the character string "CI23" (D0 and later) is searched from the 5th character from the left end (head character) of the character string "CI2312CIM" (R0 and later) when X000 is set to ON. The search result is stored in D100



Character string to be searched R0

	b15----b8	b7---- b0	
R0	49H(I)	43H(C)	These characters are not searched because the search start position is "5".
R1	33H(3)	32H(2)	
R2	32H(2)	31H(1)	← The search source character string is searched from the 5th character.
R3	49H(I)	43H(C)	
R4	00H	4DH(M)	

"CI2312CIM"

Character string D0

	b15----b8	b7---- b0
D0	49H(I)	43H(C)
D1	33H(3)	32H(2)
D2		00H

"CI23"
⇩
D100 | 0

Because the searched character string is not detected, "0" is stored.

26.10 FNC209 – $MOV / Character String Transfer

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction transfers character string data.

→ **For handling of character strings, refer to Section 5.3.**

1. Instruction format

FNC 209 $MOV P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	$MOV	Continuous Operation
	$MOVP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
(S•)	Directly specified character string (up to 32 characters) or head device number storing character string which is handled as the transfer source	Character string
(D•)	Head device number storing transferred character string	Character string

3. Applicable devices

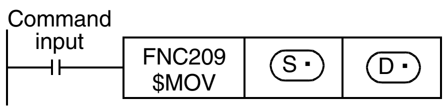
Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S•)								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓				✓	
(D•)									✓	✓	✓	✓	✓	✓	✓	✓			✓					

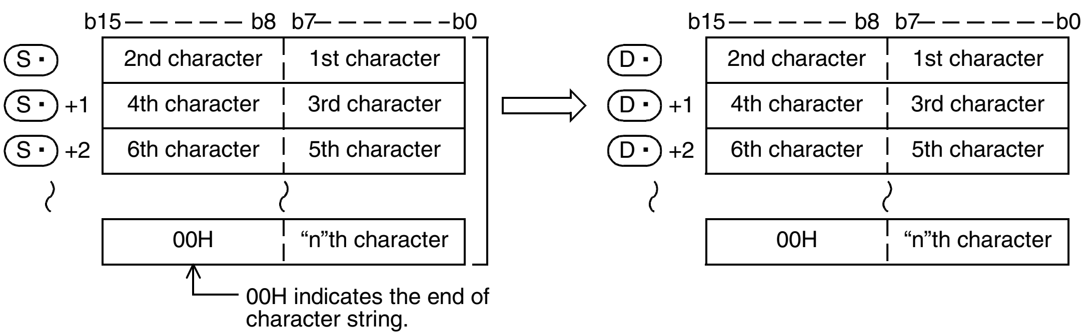
Explanation of function and operation

1. 16-bit operation ($MOV and $MOVP)

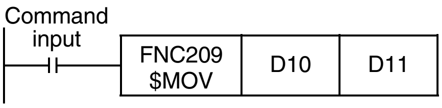
The character string data stored in the device specified by (S•) and later is transferred to the device specified by (D•) and later.

From the device number specified by (S•) to a device after that which stores "00H" in its high-order or low-order byte are transferred at one time.





Even if the device range "S· to S·+n" storing the transfer source character string data overlaps the device range "D· to D·+n/2" storing the transferred character string data, transfer is executed.
For example, when a character string stored in D10 to D13 is transferred to D11 to D14, the transfer is executed as shown below:



Before transfer:

	b15 – b8	b7 – b0
D10	32H(2)	31H(1)
D11	34H(4)	33H(3)
D12	36H(6)	35H(5)
D13	0000H	
D14		

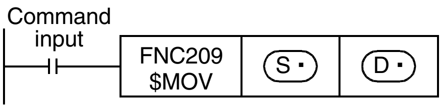
After transfer:

	b15 – b8	b7 – b0
D10	32H(2)	31H(1)
D11	32H(2)	31H(1)
D12	34H(4)	33H(3)
D13	36H(6)	35H(5)
D14	0000H	

··· It is same as the character string before transfer.

Caution

When "00H" is stored in the low-order byte of S·+n, "00H" is stored to both the high-order byte and low-order byte of D·+n.



	b15 – b8	b7 – b0
S·	42H(B)	41H(A)
S· +1	44H(D)	43H(C)
S· +2	45H(E)	00H

The high-order byte is not transferred.

	b15 – b8	b7 – b0
D·	42H(B)	41H(A)
D· +1	44H(D)	43H(C)
D· +2	00H	00H

"00H" is automatically stored in the high-order byte.

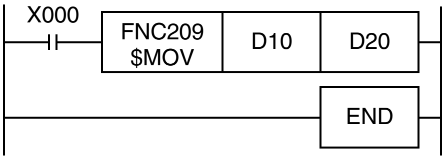
Errors

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When "00H" does not exist in the range specified from device S· (error code: K6706)
- When the specified character string cannot be stored in devices from the device specified by D· to the last device (error code: K6706)

Program example

In the program example shown below, character string data stored in D10 to D12 is transferred to D20 through D22.



	b15 – b8	b7 – b0
D10	4DH(M)	2AH(*)
D11	45H(E)	45H(E)
D12		00H

⇒

	b15 – b8	b7 – b0
D20	4DH(M)	2AH(*)
D21	45H(E)	45H(E)
D22	0000H	

27. Data Operation 3 – FNC210 to FNC219

FNC210 to FNC219 provide instructions for reading last-in data and controlling leftward/rightward shift instructions with carry.

FNC No.	Mnemonic	Symbol	Function	Reference
210	FDEL	FDEL S D n	Deleting Data from Tables	Section 27.1
211	FINS	FINS S D n	Inserting Data to Tables	Section 27.2
212	POP	POP S D n	Shift Last Data Read [FILO Control]	Section 27.3
213	SFR	SFR D n	16-bit data n Bit Shift Right with Carry	Section 27.4
214	SFL	SFL D n	16-bit data n Bit Shift Left with Carry	Section 27.5
215	–			–
216	–			–
217	–			–
218	–			–
219	–			–

27.1 FNC210 – FDEL / Deleting Data from Tables

FX3UC Ver.2.20 → FX3U Ver.2.20 →

Outline

This instruction deletes an arbitrary data from a data table.

1. Instruction format

FNC 210 FDEL P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	FDEL	Continuous Operation
	FDELP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
S•	Device number storing deleted data	16-bit binary
D•	Head device number in data table	
n	Position of deleted data in table	

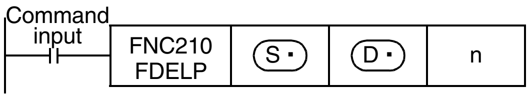
3. Applicable devices

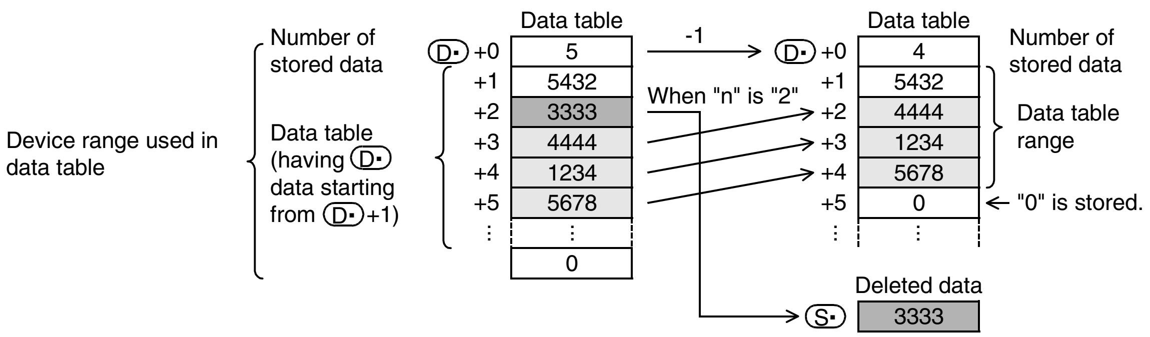
Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•												✓	✓	✓	✓				✓					
D•												✓	✓	✓	✓				✓					
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (FDEL and FDELP)

"n"th data is deleted from a data table (stored in D• and later), and the deleted data is stored in S•. "n+1"th data and later in the data table are shifted forward one by one, and the number of stored data is subtracted by "–1".





Caution

- The device range used in a data table should be controlled by the user.
  The data table has D• data starting from the next device (D•+1) after D• indicating the number of stored data.

→ **Refer to the program example.**

Related instruction

Instruction	Description
FINS(FNC211)	Inserts data into an arbitrary position in a data table.

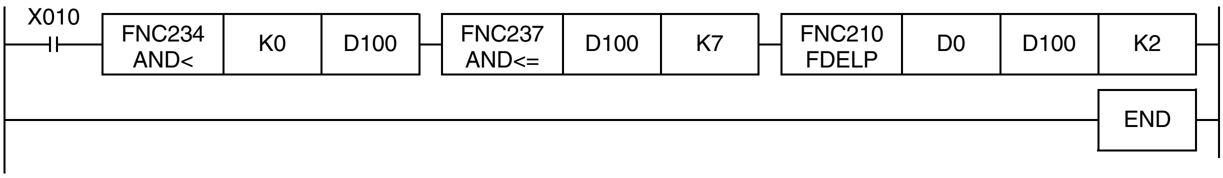
Errors

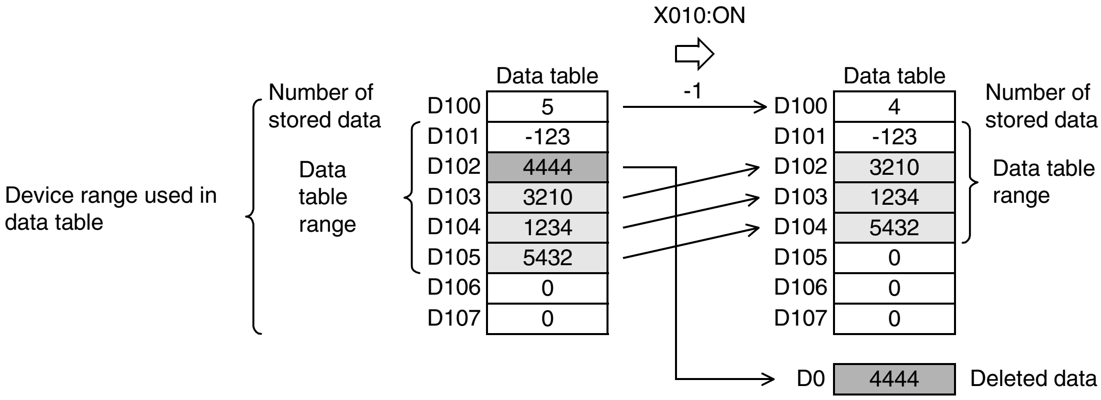
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When "n"th position from D· is larger than the number of data added by "1" (error code: K6706)
- When the value "n" exceeds the device range of the data table D· (error code: K6706)
- When FNC210 instruction is executed in the condition "n <= 0" (error code: K6706)
- When the number of stored data specified in D· is "0" (error code: K6706)
- When the data table range exceeds the corresponding device range (error code: K6706)

Program example

In the program shown below, the 2nd data is deleted from the data table stored in D100 to D105, and the deleted data is stored in D0.
When the number of stored data is "0", however, FDEL (FNC210) instruction is not executed. (The device range used in the data table is D100 to D107).





27.2 FNC211 – FINS / Inserting Data to Tables

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

This instruction inserts data into an arbitrary position in a data table.

1. Instruction format

FNC 211 FINS	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	FINS	Continuous Operation
	FINSP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
S·	Device number storing inserted data	16-bit binary
D·	Head device number in data table	
n	Data insertion position in table	

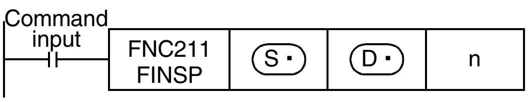
3. Applicable devices

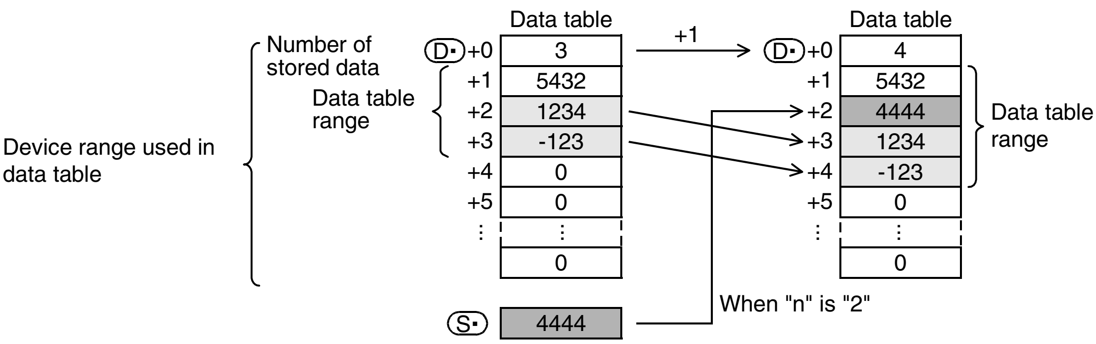
Operand Type	Bit Devices: System User X	Y	M	T	C	S	D□.b	Word Devices: Digit Specification KnX	KnY	KnM	KnS	System User T	C	D	R	Special Unit U□\G□	Index V	Z	Modify	Others: Constant K	H	Real Number E	Character String "□"	Pointer P
S·												✓	✓	✓	✓				✓	✓	✓			
D·												✓	✓	✓	✓				✓					
n														✓	✓					✓	✓			

Explanation of function and operation

1. 16-bit operation (FINS and FINSP)

16-bit data S· is inserted in "n"th position in a data table (stored in D· and later).
"n"th data and later in the data table are shifted backward one by one, and the number of stored data is added by "1".





Caution

- The device range used in a data table should be controlled by the user.
  The data table has D· data starting from the next device (D·+1) after D· indicating the number of stored data.

→ **Refer to the program example.**

Related instruction

Instruction	Description
FDEL(FNC210)	Deletes an arbitrary data from a data table.

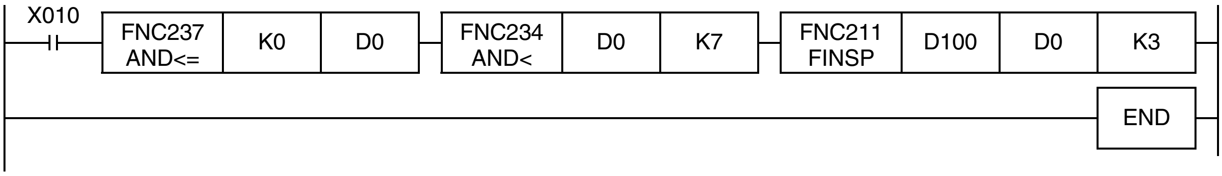
Errors

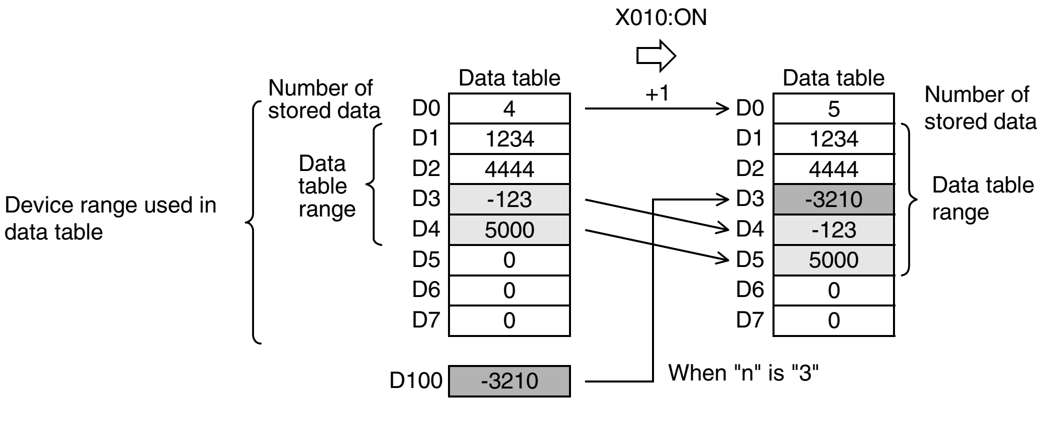
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When "n"th position from (D·) is larger than the number of data added by "1" (error code: K6706)
- When the value "n" exceeds the device range of the data table (D·) (error code: K6706)
- When FNC211 instruction is executed in the condition "n <= 0" (error code: K6706)
- When the data table range exceeds the corresponding device range (error code: K6706)

Program example

In the program shown below, data stored in D100 is inserted into the 3rd position in the data table stored in D0 to D4.
When the number of stored data exceeds "7", however, FINS (FNC211) instruction is not executed. (The device range used in the data table is D0 to D7).





27.3 FNC212 – POP / Shift Last Data Read [FILO Control]

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction reads the last data written by shift write (SFWR) instruction for FILO control.

→ For SFWR (FNC 38) instruction, refer to Section 11.9.

1. Instruction format

FNC 212 POP	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	POP	Continuous Operation
	POPP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
(S·)	Head device number storing first-in data (including pointer data) (Head word device number storing data)	16-bit binary
(D·)	Device number storing last-out data	
n	Number of stored data (Add "1" because pointer data is also included.) 2 <= n <= 512	

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)									✓	✓	✓	✓	✓	✓	✓	✓			✓					
(D·)									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
n																				✓	✓			

Explanation of function and operation

1. 16-bit operation (POP and POPP)

Command input

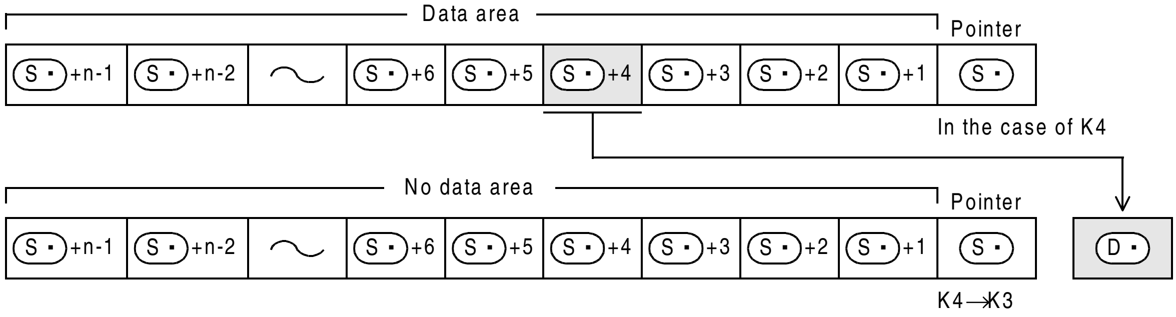
FNC212 POPP	(S·)	(D·)	n

Data for FILO controll

	Description
(S·)	Pointer data (number of stored data)
(S·)+1	Data area (First-in data written by shift write (SFWR) instruction)
(S·)+2	
(S·)+3	
⋮	
(S·)+n-3	
(S·)+n-2	
(S·)+n-1	

- Every time the instruction is executed for the word devices (S·) to (S·)+n-1, a device "(S·) + Pointer data (S·)" is read to (D·). (The last data written by the shift write (SFWR) instruction for first-in first-out control is read to (D·).) Specify "n" in the range from "2" to "512".

- Subtract “1” from the value of the pointer data S· .



Related device

→ **For the zero flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8020	Zero flag	Turns ON when the instruction is executed while the pointer S· is “0”.

Related instructions

Instruction	Description
SFWR(FNC 38)	Shift write [for FIFO/FILO control]
SFRD(FNC 39)	Shift read [for FIFO control]

Cautions

- When this instruction is programmed in the continuous operation type, the instruction is executed in every operation cycle. As a result, an expected operation may not be achieved.
  Usually, program this instruction in the “pulse operation type”, or let this instruction be executed by a “pulsed command contact”.
- When the current value of the pointer S· is “0”, the zero flag M8020 turns ON and the instruction is not executed.
  Check in advance using a comparison instruction whether the current value of S· satisfies “1 ≤ S· ≤ n-1”, and then execute this instruction.
- When the current value of the pointer S· is “1”, “0” is written to S· and the zero flag M8020 turns ON.

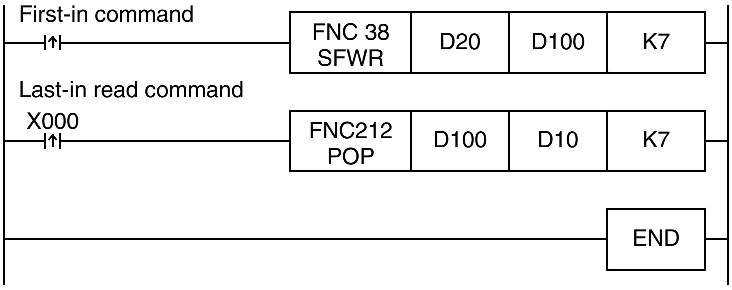
Error

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When S· is larger than “n-1” (error code: K6706)
- When S· is smaller than “0” (error code: K6706)

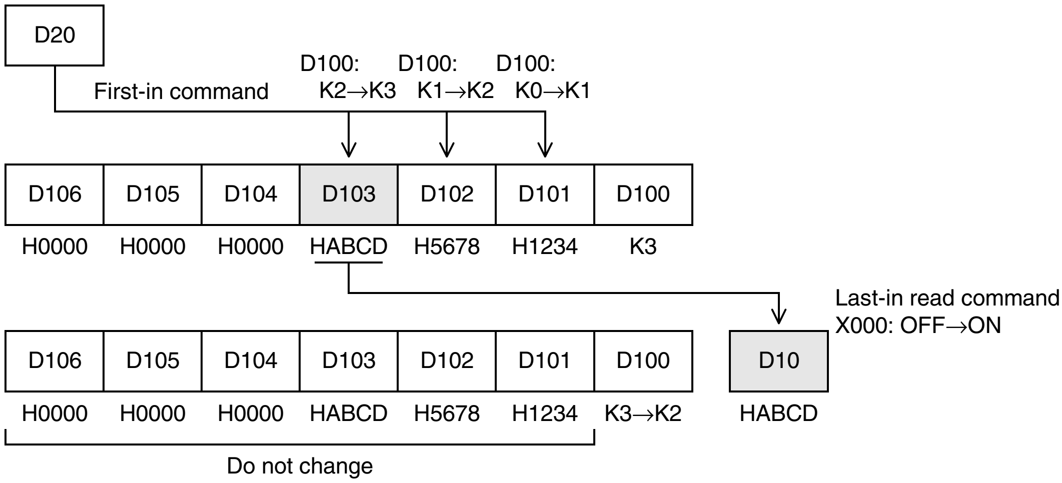
Program example

Among values stored in D20 input first to D101 to D106, the last value input is stored to D10, and "1" is subtracted from the number of stored data (pointer D100) every time X000 turns ON.



When the first-in data are as shown in the table below

Pointer	D100	K3
Data	D101	H1234
	D102	H5678
	D103	HABCD
	D104	H0000
	D105	H0000
	D106	H0000



27.4 FNC213 – SFR / Bit Shift Right with Carry

FX3UC Ver.1.00 → FX3U Ver.2.20 →

Outline

This instruction shifts 16 bits stored in a word device rightward by "n" bits.

1. Instruction format

FNC 213 SFR P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	SFR	Continuous Operation
	SFRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

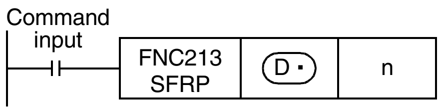
Operand Type	Description	Data Type
D·	Device number storing data to be shifted	16-bit binary
n	Number of times of shift (0 ≤ n ≤ 15)	

3. Applicable devices

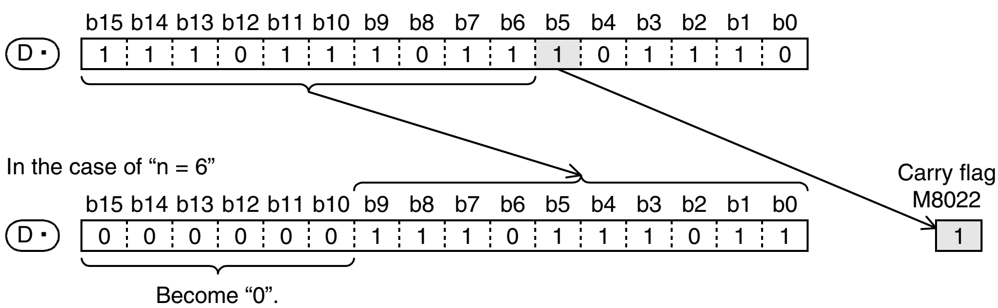
Operand Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
D·									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
n								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓			

Explanation of function and operation

1. 16-bit operation (SFR and SFRP)

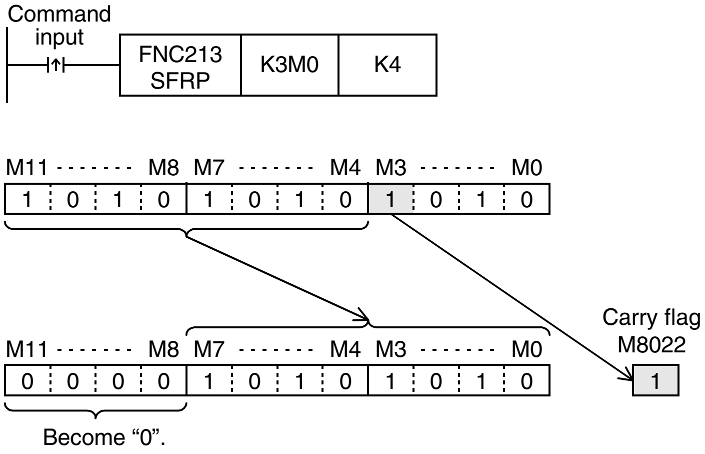


1) 16 bits stored in a word device D· are shifted rightward by "n" bits.
   Specify a value in the range from "0" to "15" as "n".
   If "16" or larger value is specified as "n", 16 bits are shifted rightward by the remainder of "n/16".
   For example, when "n" is set to "18", 16 bits are shifted rightward by 2 bits (18/16 = 1 ... 2).
2) The ON (1)/OFF (0) status of the "n"th bit (bit "n-1") in the word device D· is transferred to the carry flag M8022.
3) "0" is set to "n" bits from the most significant bit.



When a bit device is specified by digit specification

4×K□ bits are shifted according to the data bit specification.



Related device

→ **For the carry flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8022	Carry flag	Shifts the ON/OFF status of bit "n-1".

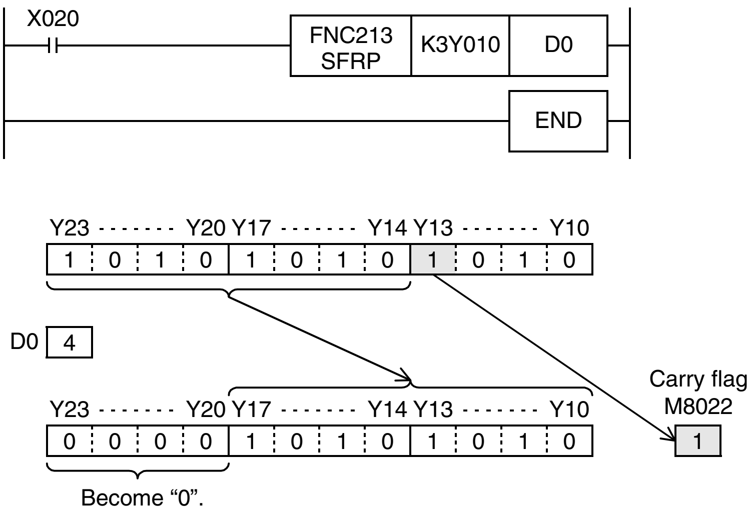
Error

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When a negative value is set to "n" (error code: K6706)

Program example

In the program example shown below, the contents of Y010 to Y023 are shifted rightward by the number of bits specified by D0 when X020 turns ON.



27.5 FNC214 – SFL / Bit Shift Left with Carry

FX3UC Ver.1.00 ⇒ FX3U Ver.2.20 ⇒

Outline

This instruction shifts 16 bits stored in a word device leftward by "n" bits.

1. Instruction format

FNC 214 SFL	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	SFL	Continuous Operation
	SFLP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

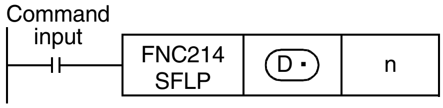
Operand Type	Description	Data Type
(D·)	Device number storing data to be shifted	16-bit binary
n	Number of times of shift (0 ≤ n ≤ 15)	

3. Applicable devices

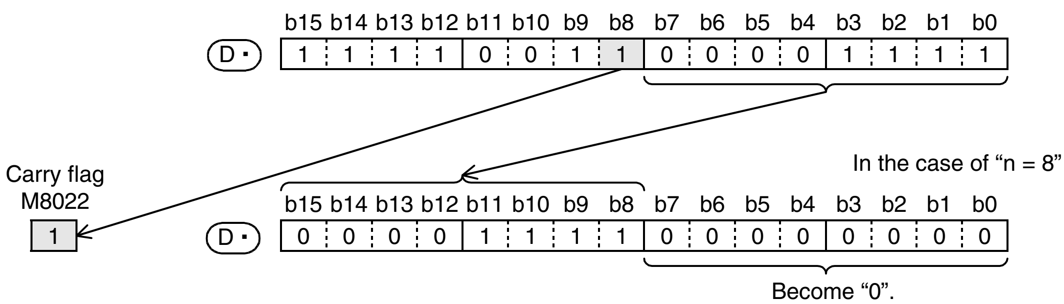
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(D·)									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
n								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓			

Explanation of function and operation

1. 16-bit operation (SFL and SFLP)

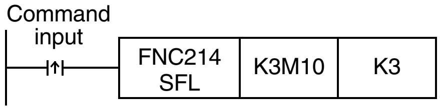


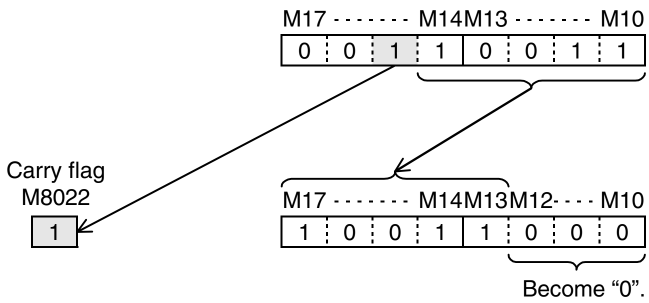
1) 16 bits stored in a word device (D·) are shifted leftward by "n" bits.
   Specify a value in the range from "0" to "15" as "n".
   If "16" or larger value is specified as "n", 16 bits are shifted leftward by the remainder of "n/16".
   For example, when "n" is set to "18", 16 bits are shifted leftward by 2 bits (18/16 = 1 ... 2).
2) The ON (1)/OFF (0) status of the "n+1"th bit (bit "n") in the word device (D·) is transferred to the carry flag M8022.
3) "0" is set to "n" bits from the least significant bit.



When a bit device is specified by digit specification

4×K□ bits are shifted according to the data of bit specification.





Related device

→ **For the carry flag use method, refer to Subsection 6.5.2.**

Device	Name	Description
M8022	Carry flag	Shifts the ON/OFF status of bit “n”.

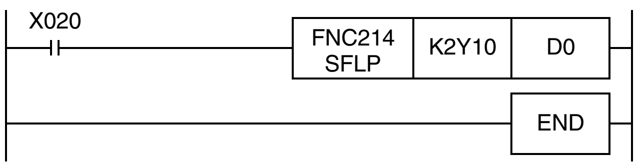
Error

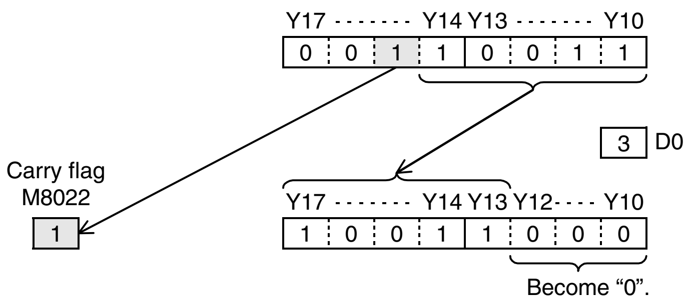
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When a negative value is set to “n” (error code: K6706)

Program example

In the program example shown below, the contents of Y010 to Y017 are shifted leftward by the number of bits specified by D0 when X020 turns ON.

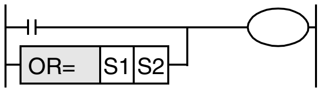
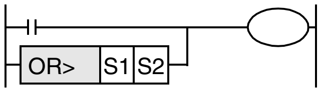
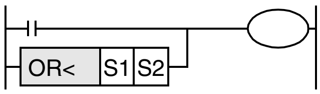
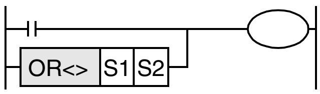
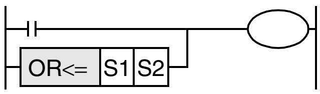
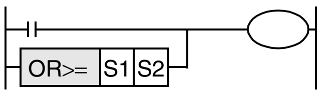




28. Data Comparison – FNC220 to FNC249

FNC220 to FNC249 provide data comparison instructions which can be handled as contact symbols in programming such as LD, AND and OR.

FNC No.	Mnemonic	Symbol	Function	Reference
220	–			–
221	–			–
222	–			–
223	–			–
224	LD=	LD= S1 S2	Load Compare S1 = S2	Section 28.1
225	LD>	LD> S1 S2	Load Compare S1 > S2	Section 28.1
226	LD<	LD< S1 S2	Load Compare S1 < S2	Section 28.1
227	–			–
228	LD<>	LD<> S1 S2	Load Compare S1 ≠ S2	Section 28.1
229	LD<=	LD<= S1 S2	Load Compare S1 <= S2	Section 28.1
230	LD>=	LD>= S1 S2	Load Compare S1 >= S2	Section 28.1
231	–			–
232	AND=	AND= S1 S2	AND Compare S1 = S2	Section 28.2
233	AND>	AND> S1 S2	AND Compare S1 > S2	Section 28.2
234	AND<	AND< S1 S2	AND Compare S1 < S2	Section 28.2
235	–			–
236	AND<>	AND<> S1 S2	AND Compare S1 ≠ S2	Section 28.2
237	AND<=	AND<= S1 S2	AND Compare S1 <= S2	Section 28.2
238	AND>=	AND>= S1 S2	AND Compare S1 >= S2	Section 28.2

FNC No.	Mnemonic	Symbol	Function	Reference
239	–			–
240	OR=		OR Compare $S_1 = S_2$	Section 28.3
241	OR>		OR Compare $S_1 > S_2$	Section 28.3
242	OR<		OR Compare $S_1 < S_2$	Section 28.3
243	–			–
244	OR<>		OR Compare $S_1 \neq S_2$	Section 28.3
245	OR<=		OR Compare $S_1 <= S_2$	Section 28.3
246	OR>=		OR Compare $S_1 >= S_2$	Section 28.3
247	–			–
248	–			–
249	–			–

28.1 FNC224~230 – LD =, >, <, <>, <=, >= / Data Comparison

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

These instructions compare numeric values, and set a contact to ON when the condition agrees so that an operation is started.

1. Instruction format

Instruction	16-bit Instruction	Mnemonic	Operation Condition	32-bit Instruction	Mnemonic	Operation Condition
FNC 224 D LD=	5 steps	LD=	Continuous Operation	9 steps	LDD=	Continuous Operation
FNC 225 D LD>	5 steps	LD>	Continuous Operation	9 steps	LDD>	Continuous Operation
FNC 226 D LD<	5 steps	LD<	Continuous Operation	9 steps	LDD<	Continuous Operation
FNC 228 D LD<>	5 steps	LD<>	Continuous Operation	9 steps	LDD<>	Continuous Operation
FNC 229 D LD<=	5 steps	LD<=	Continuous Operation	9 steps	LDD<=	Continuous Operation
FNC 230 D LD>=	5 steps	LD>=	Continuous Operation	9 steps	LDD>=	Continuous Operation

2. Set data (common among FNC224 to FNC230)

Operand Type	Description	Data Type
S1•	Device number storing comparison data	16- or 32-bit binary
S2•	Device number storing comparison data	16- or 32-bit binary

3. Applicable devices (common among FNC224 to FNC230)

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			

Explanation of function and operation

FNC224 to FNC230 are data comparison instructions connected to bus lines.

The contents of (S1•) are compared with the contents of (S2•) in the binary format, and a contact becomes conductive (ON) or non-conductive (OFF) depending on the comparison result.

FNC No.	16-bit instruction	32-bit instruction	ON condition	OFF condition
224	LD=	LDD=	(S1•) = (S2•)	(S1•) ≠ (S2•)
225	LD>	LDD>	(S1•) > (S2•)	(S1•) <= (S2•)
226	LD<	LDD<	(S1•) < (S2•)	(S1•) >= (S2•)
228	LD<>	LDD<>	(S1•) ≠ (S2•)	(S1•) = (S2•)
229	LD<=	LDD<=	(S1•) <= (S2•)	(S1•) > (S2•)
230	LD>=	LDD>=	(S1•) >= (S2•)	(S1•) < (S2•)

Cautions

1. Negative value

When the most significant bit is "1" in the data stored in (S1•) or (S2•), it is regarded as a negative value in comparison.

- In the 16-bit operation: bit 15
- In the 32-bit operation: bit 31

2. When using 32-bit high speed counters

Make sure to execute the 32-bit operation (such as "LDD=", "LDD>" and "LDD<") when comparing 32-bit high speed counters (C200 and later).
If a 32-bit high speed counter is specified in the 16-bit operation (such as "LD=", "LD>" and "LD<"), a program error or operation error is caused.

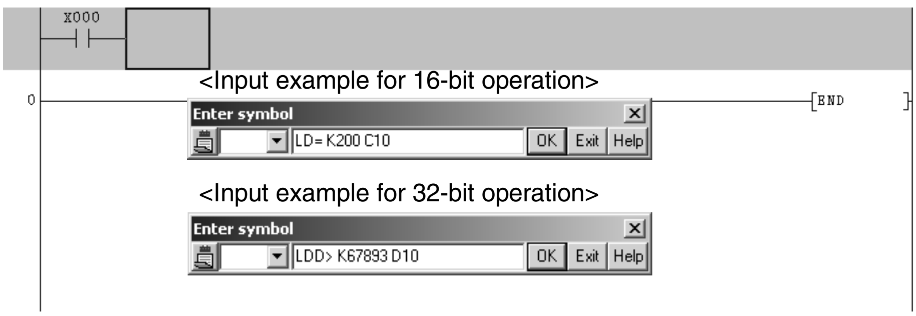
3. Programming of data comparison instructions

In programming in GX Developer, symbols "≤" and "≥" cannot be input.
Separate "≤" into "<" and "=", and separate "≥" into ">" and "=" in input.
When inputting a data comparison instruction in a circuit diagram, use the symbol "-[ ]-".
The input procedure is described below:

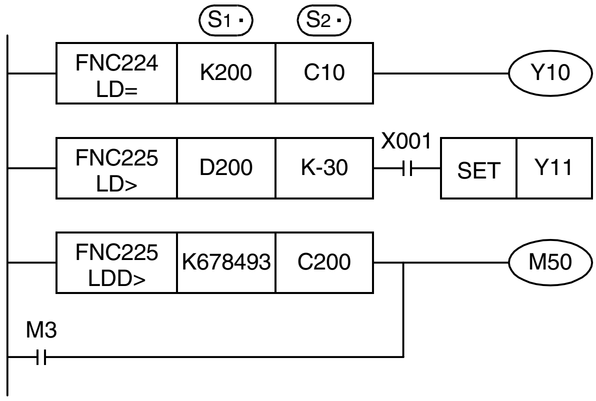
Operating procedure

a) Display the circuit program edit window, and put the cursor in a position where a data comparison instruction is to be used.
b) Select the coil symbol "-[ ]-".
c) Input "Instruction" → "space" → "value or device" → "space" → "value or device".
For an input example, refer to "Instruction input window in GX Developer" shown below.
d) Click the [OK] button.
e) Input other contacts and coil drive units consecutively.

Instruction input window in GX Developer



Program example



When the current value of the counter C10 is "200", Y010 is driven.

When the contents of D200 are "-29" or more and X001 is ON, Y011 is set.

When the contents of the counter C200 are less than "K678,493" or when M3 turns ON, M50 is driven.

28.2 FNC232~238 – AND=, >, <, < >, <=, >= / Data Comparison

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

These instructions compare numeric values, and set a contact to ON when the condition agrees.

1. Instruction format

Instruction	16-bit Instruction	Mnemonic	Operation Condition	32-bit Instruction	Mnemonic	Operation Condition
FNC 232 D AND=	5 steps	AND=	Continuous Operation	9 steps	ANDD=	Continuous Operation
FNC 233 D AND>	5 steps	AND>	Continuous Operation	9 steps	ANDD>	Continuous Operation
FNC 234 D AND<	5 steps	AND<	Continuous Operation	9 steps	ANDD<	Continuous Operation
FNC 236 D AND<>	5 steps	AND<>	Continuous Operation	9 steps	ANDD<>	Continuous Operation
FNC 237 D AND<=	5 steps	AND<=	Continuous Operation	9 steps	ANDD<=	Continuous Operation
FNC 238 D AND>=	5 steps	AND>=	Continuous Operation	9 steps	ANDD>=	Continuous Operation

2. Set data (common among FNC232 to FNC238)

Operand Type	Description	Data Type
S1•	Device number storing comparison data	16- or 32-bit binary
S2•	Device number storing comparison data	16- or 32-bit binary

3. Applicable devices (common among FNC232 to FNC238)

Oper-and Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			

Explanation of function and operation

FNC232 to FNC237 are data comparison instructions connected to other contacts in series.

The contents of S1• are compared with the contents of S2• in binary format, and a contact becomes conductive (ON) or non-conductive (OFF) depending on the comparison result.

FNC No.	16-bit instruction	32-bit instruction	ON condition	OFF condition
232	AND=	ANDD=	S1• = S2•	S1• ≠ S2•
233	AND>	ANDD>	S1• > S2•	S1• <= S2•
234	AND<	ANDD<	S1• < S2•	S1• >= S2•
235	AND<>	ANDD<>	S1• ≠ S2•	S1• = S2•
236	AND<=	ANDD<=	S1• <= S2•	S1• > S2•
237	AND>=	ANDD>=	S1• >= S2•	S1• < S2•

Cautions

1. Negative value

When the most significant bit is "1" in the data stored in S1• or S2•, it is regarded as a negative value in comparison.

- In the 16-bit operation: bit 15
- In the 32-bit operation: bit 31

2. When using 32-bit high speed counters

Make sure to execute the 32-bit operation (such as "ANDD=", "ANDD>" and "ANDD<") when comparing 32-bit high speed counters (C200 and later).
If a 32-bit high speed counter is specified in the 16-bit operation (such as "AND=", "AND>" and "AND<"), a program error or operation error is caused.

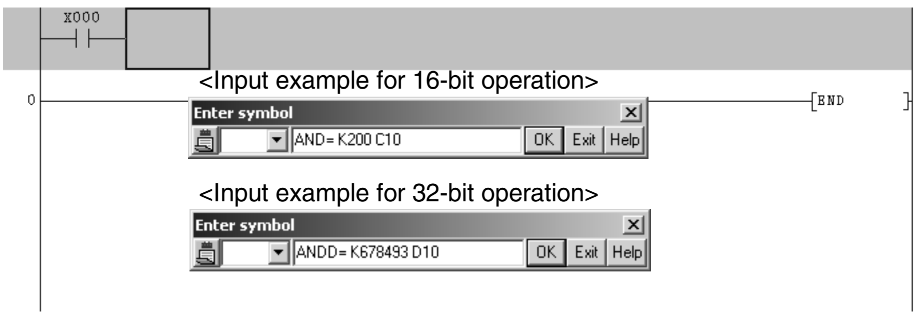
3. Programming of data comparison instructions

When programming in GX Developer, symbols "≤" and "≥" cannot be input.
Separate "≤" into "<" and "=", and separate "≥" into ">" and "=".
When inputting a data comparison instruction in a circuit diagram, use the symbol "-[ ]-".
The input procedure is described below:

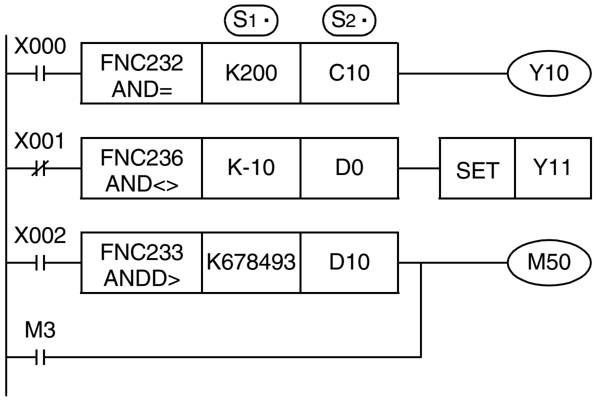
Operating procedure

a) Display the circuit program edit window, and put the cursor in a position where a data comparison instruction is to be used.
b) Select the coil symbol “-[ ]-”.
c) Input "Instruction" → "space" → "value or device" → "space" → "value or device".
For an input example, refer to "Instruction input window in GX Developer" shown below.
d) Click the [OK] button.
e) Input other contacts and coil drive units consecutively.

Instruction input window in GX Developer



Program example



When X000 is ON and the current value of the counter C10 is "200" Y010 is driven.

When X001 is OFF and the contents of the data register D0 are not "-10", Y011 is set.

When X002 is ON, when the contents of the data registers D11 and D10 are less than "K678,493", or when M3 turns ON, M50 is driven.

28.3 FNC240~246 – OR=, >, <, < >, <=, >= / Data Comparison

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

These instructions compare numeric values, and set a contact to ON when the condition agrees.

1. Instruction format

FNC No.	16-bit Instruction	Mnemonic	Operation Condition	32-bit Instruction	Mnemonic	Operation Condition
FNC 240 D OR=	5 steps	OR=	Continuous Operation	9 steps	ORD=	Continuous Operation
FNC 241 D OR>	5 steps	OR>	Continuous Operation	9 steps	ORD>	Continuous Operation
FNC 242 D OR<	5 steps	OR<	Continuous Operation	9 steps	ORD<	Continuous Operation
FNC 244 D OR<>	5 steps	OR<>	Continuous Operation	9 steps	ORD<>	Continuous Operation
FNC 245 D OR<=	5 steps	OR<=	Continuous Operation	9 steps	ORD<=	Continuous Operation
FNC 246 D OR>=	5 steps	OR>=	Continuous Operation	9 steps	ORD>=	Continuous Operation

2. Set data (common among FNC240 to FNC246)

Operand Type	Description	Data Type
S1•	Device number storing comparison data	16- or 32-bit binary
S2•	Device number storing comparison data	16- or 32-bit binary

3. Applicable devices (common among FNC240 to FNC246)

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			

21 FNC160-FNC169 Real Time Clock Control
22 FNC170-FNC179 External Device
23 FNC180 Alternate Instructions
24 FNC181-FNC189 Others
25 FNC190-FNC199 Block Data Operation
26 FNC200-FNC209 Character String Control
27 FNC210-FNC219 Data Operation 3
28 FNC220-FNC249 Data Comparison
29 FNC250-FNC269 Data Table Operation
30 FNC270-FNC274 Ex-Device Inverter Comms

Explanation of function and operation

FNC240 to 246 are data comparison instructions connected to other contacts in parallel.

The contents of S1• are compared with the contents of S2• in binary format, and a contact becomes conductive (ON) or non-conductive (OFF) depending on the comparison result.

FNC No.	16-bit instruction	32-bit instruction	ON condition	OFF condition
240	OR=	ORD=	S1• = S2•	S1• ≠ S2•
241	OR>	ORD>	S1• > S2•	S1• <= S2•
242	OR<	ORD<	S1• < S2•	S1• >= S2•
244	OR<>	ORD<>	S1• ≠ S2•	S1• = S2•
245	OR<=	ORD<=	S1• <= S2•	S1• > S2•
246	OR>=	ORD>=	S1• >= S2•	S1• < S2•

Cautions

1. Negative value

When the most significant bit is "1" in the data stored in S1• or S2•, it is regarded as a negative value in comparison.

- In the 16-bit operation: bit 15
- In the 32-bit operation: bit 31

2. When using 32-bit counters and 32-bit high speed counters

Make sure to execute the 32-bit operation (such as "ORD=", "ORD>" and "ORD<") when comparing 32-bit counters and 32-bit high speed counters (C200 and later).
If a 32-bit high speed counter is specified in the 16-bit operation (such as "ORD=", "OR>" and "OR<"), a program error or operation error is caused.

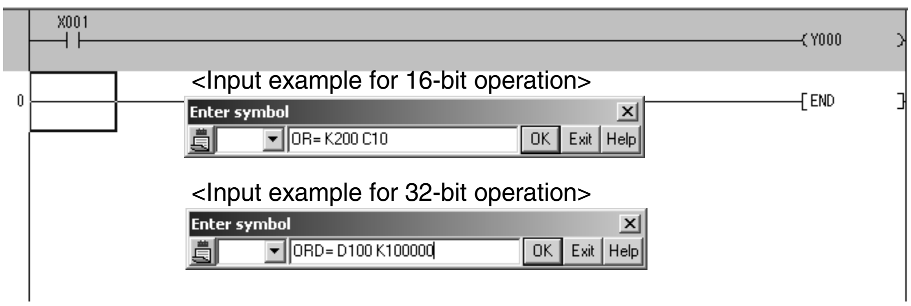
3. Programming of data comparison instructions

When programming in GX Developer, symbols "≤" and "≥" cannot be input.
Separate "≤" into "<" and "=", and separate "≥" into ">" and "=".
When inputting a data comparison instruction in a circuit diagram, use the symbol "-[ ]-".
The input procedure is described below:

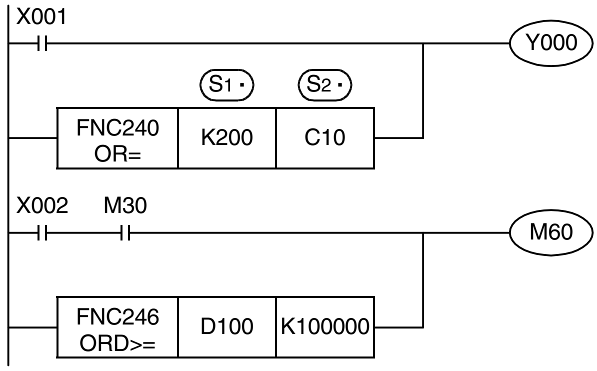
Operating procedure

a) Display the circuit program edit window, and put the cursor in a position where a data comparison instruction is to be used.
b) Select the coil symbol "-[ ]-".
c) Input "Instruction" → "space" → "value or device" → "space" → "value or device".
For an input example, refer to "Instruction input window in GX Developer" shown below.
d) Click the [OK] button.
e) Input other contacts and coil drive units consecutively.

Instruction input window in GX Developer



Program example



When X001 turns ON or when the current value of the counter C10 is "200", Y000 is driven.

When X002 and M30 turn ON or when the contents of the data registers D101 and D100 are more than "K100,000" M60 is driven.

29. Data Table Operation – FNC250 to FNC269

FNC No.	Mnemonic	Symbol	Function	Reference
250	–			–
251	–			–
252	–			–
253	–			–
254	–			–
255	–			–
256	LIMIT	LIMIT S1 S2 S3 D	Limit Control	Section 29.1
257	BAND	BAND S1 S2 S3 D	Dead Band Control	Section 29.2
258	ZONE	ZONE S1 S2 S3 D	Zone Control	Section 29.3
259	SCL	SCL S1 S2 D	Scaling (Coordinate by Point Data)	Section 29.4
260	DABIN	DABIN S D	Decimal ASCII to BIN Conversion	Section 29.5
261	BINDA	BINDA S D	BIN to Decimal ASCII Conversion	Section 29.6
262	–			–
263	–			–
264	–			–
265	–			–
266	–			–
267	–			–
268	–			–
269	SCL2	SCL2 S1 S2 D	Scaling 2 (Coordinate by X/Y Data)	Section 29.7

29.1 FNC256 – LIMIT / Limit Control

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction provides the upper limit value and lower limit value for an input numeric value, and controls the output value using these limit values.

1. Instruction format

D	FNC 256 LIMIT	P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	LIMIT	Continuous Operation
	LIMITP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DLIMIT	Continuous Operation
	DLIMITP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S1•	Lower limit value (minimum output value)	16- or 32-bit binary
S2•	Upper limit value (maximum output value)	
S3•	Input value controlled by the upper and lower limit values	
D•	Head device number storing the output value controlled by the upper and lower limit values	

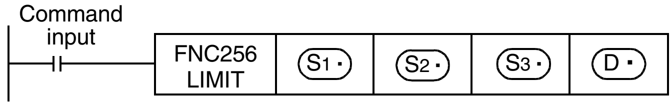
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓			
S3•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
D•									✓	✓	✓	✓	✓	✓	✓	✓			✓					

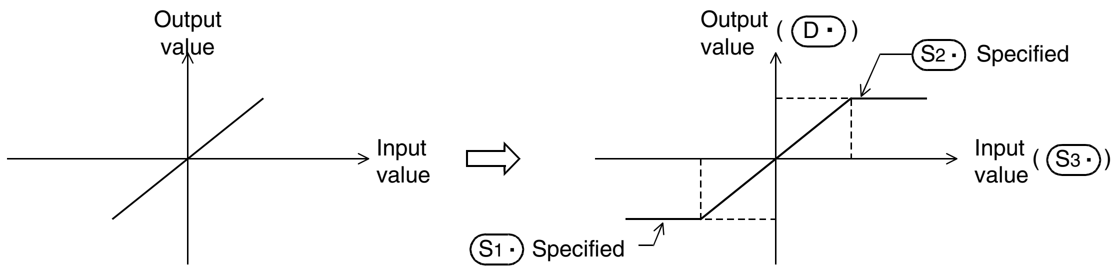
Explanation of function and operation

1. 16-bit operation (LIMIT and LIMITP)

Depending on how the input value (16-bit binary value) specified by S3• compares to the range between S1• and S2•, the output value D• is controlled.
The output value is controlled as shown below:



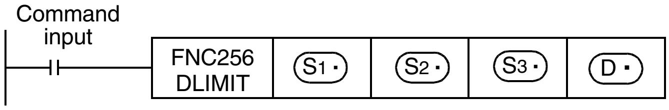
•In the case of " S1• Lower limit value > S3• Input value" .... S1• Lower limit value → D• Output value
•In the case of " S2• Upper limit value < S3• Input value" .... S2• Upper limit value → D• Output value
•In the case of " S1• Lower limit value ≤ S3• Input value ≤ S2• Upper limit value" .... S3• Input value → D• Output value



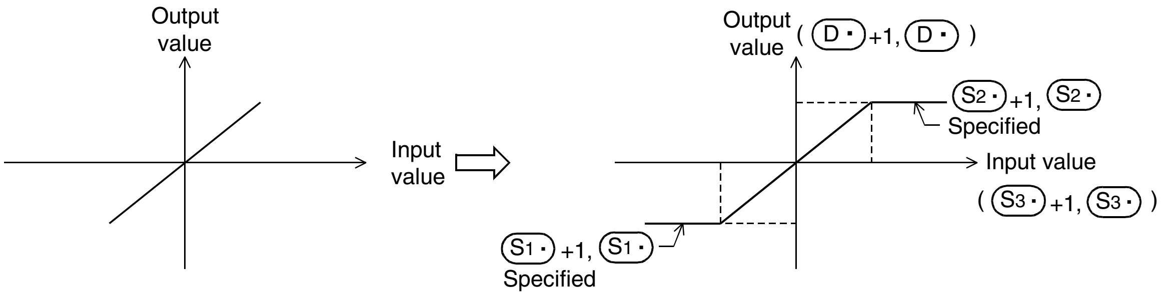
- When controlling the output value using only the upper limit value, set "-32768" to the lower limit value specified in S1•.
- When controlling the output value using only the lower limit value, set "32767" to the upper limit value specified in S2•.

2. 32-bit operation (DLIMIT and DLIMITP)

Depending on how the input value (32-bit binary value) specified by [S3·+1, S3·] compares to the range between [S1·+1, S1·] and [S2·+1, S2·], the output value [D·+1, D·] is controlled.



- In the case of "Lower limit value (S1·+1, S1·) > Input value (S3·+1, S3·)" .... Lower limit value (S1·+1, S1·) → Output value (D·+1, D·)
- In the case of "Upper limit value (S2·+1, S2·) < Input value (S3·+1, S3·)" .... Upper limit value (S2·+1, S2·) → Output value (D·+1, D·)
- In the case of "Lower limit value (S1·+1, S1·) ≤ Input value (S3·+1, S3·) ≤ Upper limit value (S2·+1, S2·)" .... Input value (S3·+1, S3·) → Output value (D·+1, D·)



- When controlling the output value using only the upper limit value, set "-2,147,483,648" to the lower limit value specified in [S1·+1, S1·].
- When controlling the output value using only the lower limit value, set "2,147,483,647" to the upper limit value specified in [S2·+1, S2·].

Error

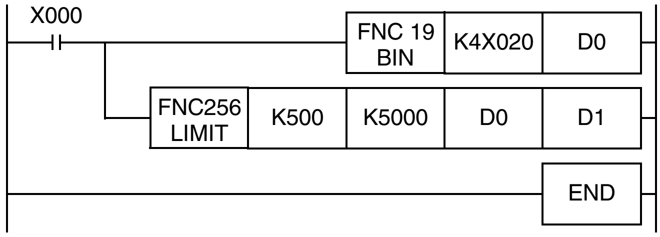
An operation error is caused when the instruction is executed in the setting status shown below; The error flag M8067 turns ON, and the error code (K6706) is stored in D8067.

	Relationship
16-bit operation	S1· ≤ S2·
32-bit operation	[S1·+1, S1·] ≤ [S2·+1, S2·]

Program examples

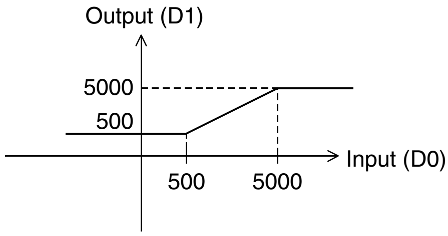
1. Program example 1

In the program example shown below, the BCD data set in X020 to X037 is controlled by the limit values “500” to “5000”, and the controlled value is output to D1 when X000 turns ON.



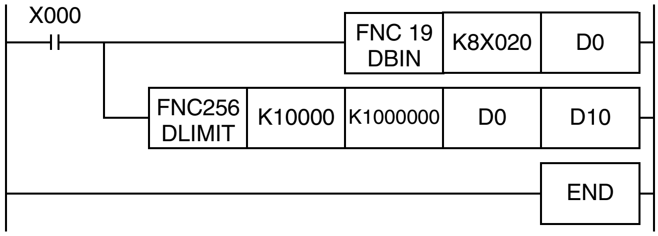
Operation

- In the case of “D0 < 500”, “500” is output to D1.
- In the case of “500 ≤ D0 ≤ 5000”, the value of D0 is output to D1.
- In the case of “D0 > 5000”, “5000” is output to D1.



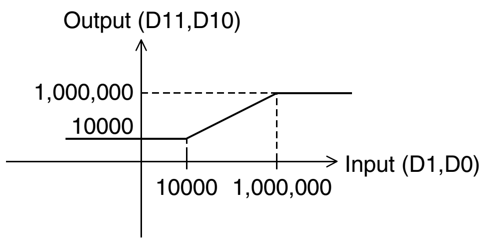
2. Program example 2

In the program example shown below, the BCD data set in X020 to X057 is controlled by the limit values “10000” and “1,000,000”, and the controlled value is output to D11 and D10 when X000 turns ON.



Operation

- In the case of “(D1, D0) < 10000”, “10000” is output to (D11, D10).
- In the case of “10000 ≤ (D1, D0) ≤ 1,000,000”, the value of (D1, D0) is output to (D11, D10).
- In the case of “(D1, D0) > 1,000,000”, “1,000,000” is output to (D11, D10).



29.2 FNC257 – BAND / Dead Band Control

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction provides the upper limit value and lower limit value of the dead band for an input numeric value, and controls the output value using these limit values.

1. Instruction format

D	FNC 257 BAND	P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	BAND	Continuous Operation
	BANDP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DBAND	Continuous Operation
	DBANDP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S1•	Lower limit value of the dead band (no-output band)	16- or 32-bit binary
S2•	Upper limit value of the dead band (no-output band)	
S3•	Input value controlled by the dead band	
D•	Device number storing the output value controlled by the dead band	

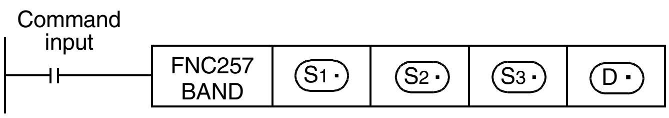
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index				Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓			
S3•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
D•									✓	✓	✓	✓	✓	✓	✓	✓			✓					

Explanation of function and operation

1. 16-bit operation (BAND and BANDP)

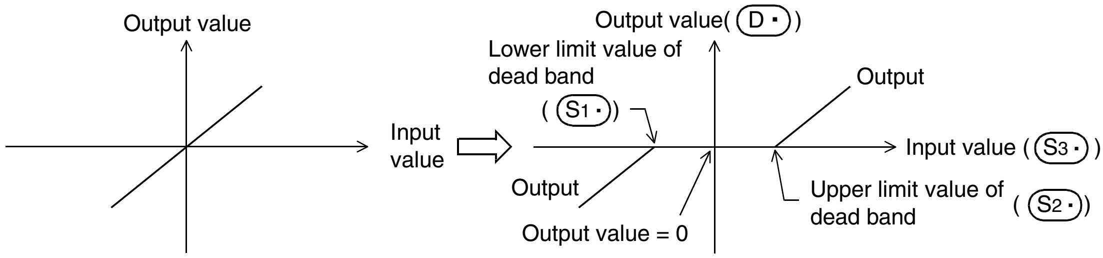
Depending on how the input value (16-bit binary value) specified by S3• compares to the dead band range between S1• and S2•, the output value D• is controlled.



•In the case of " S1• Lower limit value > S3• Input value" .... S3• Input value – S1• Lower limit value → D• Output value

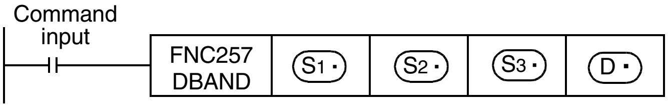
•In the case of " S2• Upper limit value < S3• Input value" .... S3• Input value – S2• Upper limit value → D• Output value

•In the case of " S1• Lower limit value ≤ S3• Input value ≤ S2• Upper limit value" ....0 → D• Output value

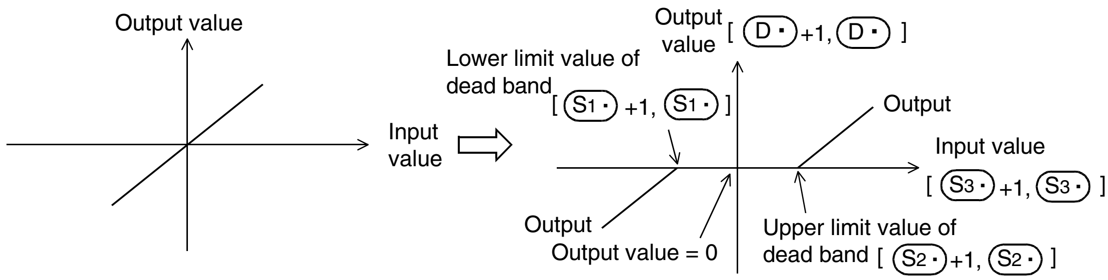


2. 32-bit operation (DBAND and DBANDP)

Depending on how the input value (32-bit binary value) specified by [S3· +1, S3·] compares to the dead band range between [S1· +1, S1·] and [S2· +1, S2·], the output value [D· +1, D·] is controlled. The output value is controlled as shown below:



- In the case of "Lower limit value (S1· +1, S1·) > Input value (S3· +1, S3·)" ....Input value (S3· +1, S3·) – Lower limit value (S1· +1, S1·) → Output value (D· +1, D·)
- In the case of "Upper limit value (S2· +1, S2·) < Input value (S3· +1, S3·)" ....Input value (S3· +1, S3·) – Upper limit value (S2· +1, S2·) → Output value (D· +1, D·)
- In the case of "Lower limit value (S1· +1, S1·) ≤ Input value (S3· +1, S3·) ≤ Upper limit value (S2· +1, S2·)" .... 0 → Output value (D· +1, D·)



Caution

- When the output value overflows, it is handled as follows:
  - In the 16-bit operation
    The output value is a 16-bit binary value with sign. Accordingly, if the operation result is outside the range from -32768 to +32767, it is handled as follows:

    Lower limit value of dead band S1· = 10
    Input value S3· = -32768
    ⇒ Output value = -32768-10
    = 8000H-AH
    = 7FF6H
    = 32758

  - In the 32-bit operation
    The output value is a 32-bit binary value with sign. Accordingly, if the operation result is outside the range from -2,147,483,648 to +2,147,483,647, it is handled as follows:

    Lower limit value of dead band (S1· +1, S1·) = 1000
    Input value (S3· +1, S3·) = -2,147,483,648
    ⇒ Output value = -2,147,483,648-1000
    = 8000000H-000003E8H
    = 7FFFFC18H
    = 2,147,482,648

Error

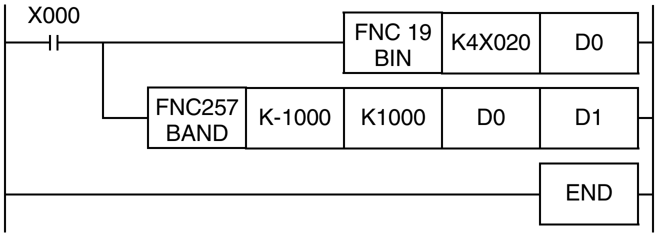
An operation error is caused when the instruction is executed in the setting status shown below; The error flag M8067 turns ON, and the error code (K6706) is stored in D8067.

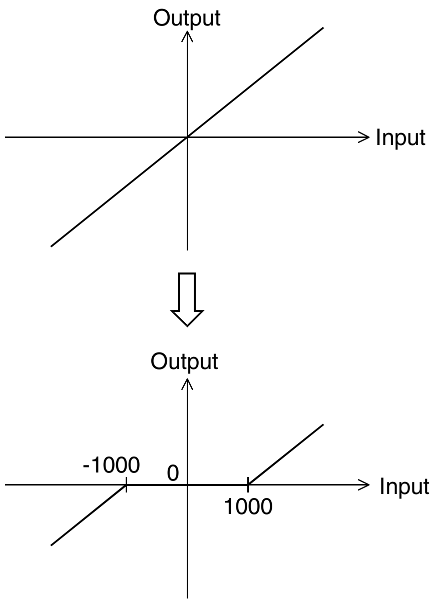
	Relationship
16-bit operation	S1· > S2·
32-bit operation	[S1· +1, S1·] > [S2· +1, S2·]

Program examples

1. Program example 1

In the program example shown below, the BCD data set in X020 to X037 is controlled by the dead band from “-1000” to “+1000”, and a controlled value is output to D1 when X000 turns ON.



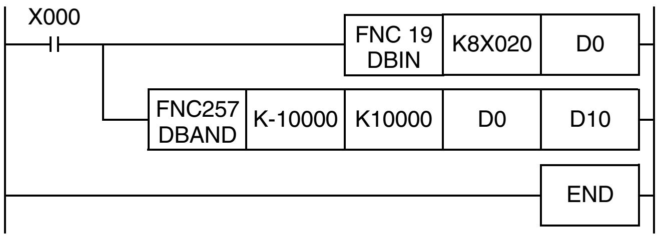


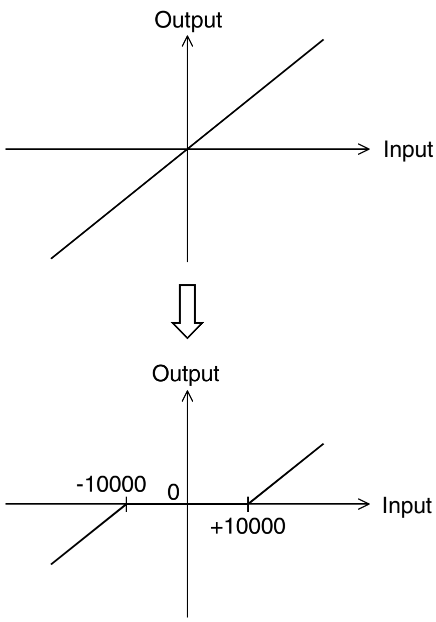
Operation

- In the case of “D0 < -1000”, “D0 - (-1000)” is output to D1.
- In the case of “-1000 ≤ D0 ≤ +1000”, “0” is output to D1.
- In the case of “D0 > +1000”, “D0 - 1000” is output to D1.

2. Program example 2

In the program example shown below, the BCD data set in X020 to X057 is controlled by the dead band from “-10000” to “+10000”, and a controlled value is output to D11 and D10 when X000 turns ON.





Operation

- In the case of “(D1, D0) < -10000”, “(D1, D0) - (-10000)” is output to (D11, D10).
- In the case of “-10000 ≤ (D1, D0) ≤ +10000”, “0” is output to (D11, D10).
- In the case of “(D1, D0) > +10000”, “(D1, D0) - 10000” is output to (D11, D10).

29.3 FNC258 – ZONE / Zone Control

Outline

Depending on how the input value compares to positive or negative, the output value is controlled by the bias value specified.

1. Instruction format

D	FNC 258 ZONE	P

16-bit Instruction	Mnemonic	Operation Condition
9 steps	ZONE	Continuous Operation
	ZONEP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
17 steps	DZONE	Continuous Operation
	DZONEP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S1•	Negative bias value to be added to the input value	16- or 32-bit binary
S2•	Positive bias value to be added to the input value	
S3•	Input value controlled by the zone	
D•	Head device number storing the output value controlled by the zone	

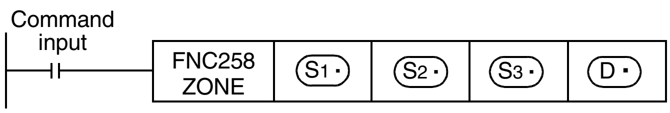
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓			
S2•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓			
S3•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓					
D•									✓	✓	✓	✓	✓	✓	✓	✓			✓					

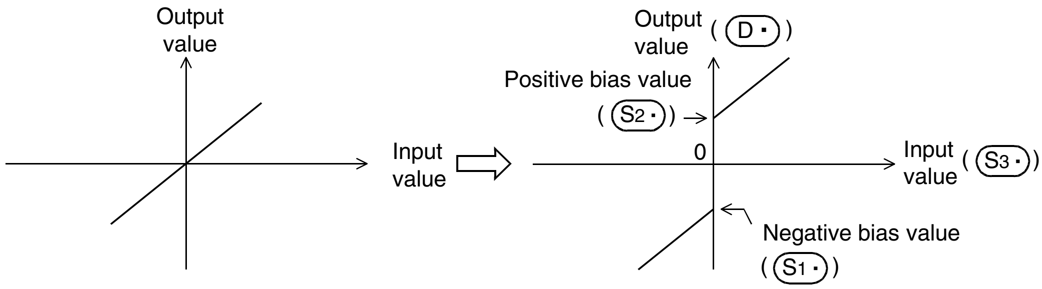
Explanation of function and operation

1. 16-bit operation (ZONE and ZONEP)

The bias value specified by S1• or S2• is added to the input value specified by S3•, and output to the device specified by D•.
The bias value is added as shown below:

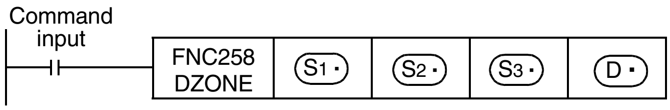


- In the case of "S3• Input value < 0" ..... S3• Input value + S1• Negative bias value → D• Output value
- In the case of "S3• Input value = 0" ....0 → D• Output value
- In the case of "S3• Input value > 0" ..... S3• Input value + S2• Positive bias value → D• Output value



2. 32-bit operation (DZONE and DZONEP)

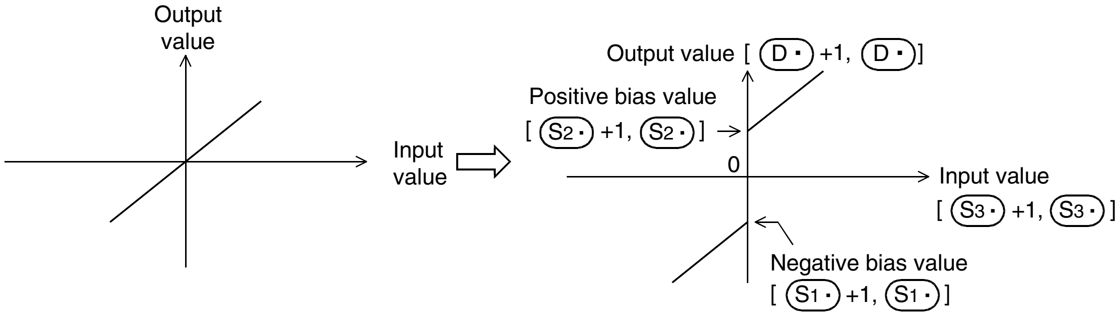
The bias value specified by [S1·+1, S1·] or [S2·+1, S2·] is added to the input value specified by [S3·+1, S3·], and output to the device specified by [D·+1, D·].
The bias value is added as shown below:



•In the case of "Input value < 0" (S3·+1, S3·) .... Input value (S3·+1, S3·) + Negative bias value (S1·+1, S1·) → Output value (D·+1, D·)

•In the case of "Input value = 0" (S3·+1, S3·) .... 0 → Output value (D·+1, D·)

•In the case of "Input value > 0" (S3·+1, S3·) .... Input value (S3·+1, S3·) + Positive bias value (S2·+1, S2·) → Output value (D·+1, D·)



Caution

- When the output value overflows, it is handled as follows:
  - In the 16-bit operation
    The operation result is a 16-bit binary value with sign. Accordingly, if the output value is outside the range from -32768 to +32767, it is handled as follows:

    Negative bias value S1· = -100
    Input value S3· = -32768
    ⇨ Output value = -32768+(-100)
    = 8000H+FF9CH
    = 7F9CH
    = 32668

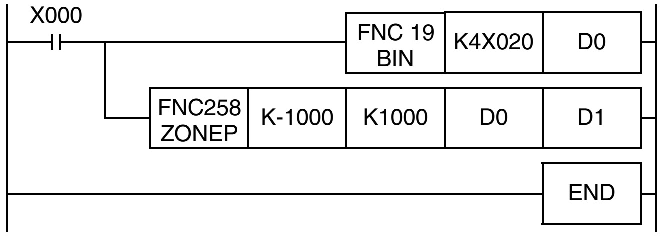
  - In the 32-bit operation
    The output value is a 32-bit binary value with sign. Accordingly, if the operation result is outside the range from -2,147,483,648 to +2,147,483,647, it is handled as follows:

    Negative bias value (S1·+1, S1·) = -1000
    Input value (S3·+1, S3·) = -2,147,483,648
    ⇨ Output value = -2,147,483,648+(-1000)
    = 80000000H+FFFF
    = 7FFFFC18
    = 2,147,482,648

Program examples

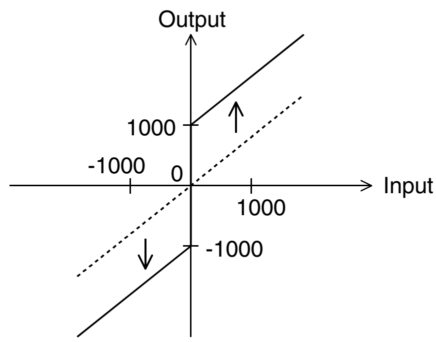
1. Program example 1

In the program example shown below, the BCD data set in X020 to X037 is controlled by the zone from "-1000" to "+1000", and the controlled value is output to D1 when X000 turns ON.



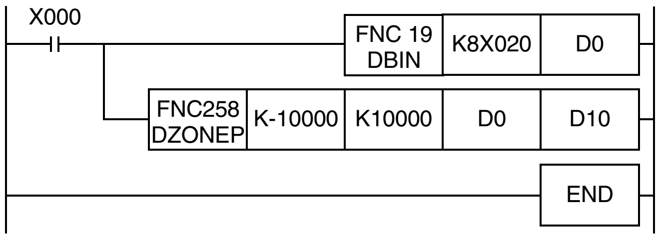
Operation

- In the case of "D0 < 0", "D0 + (-1000)" is output to D1.
- In the case of "D = 0", "0" is output to D1.
- In the case of "D0 > 0", "D0 + 1000" is output to D1.



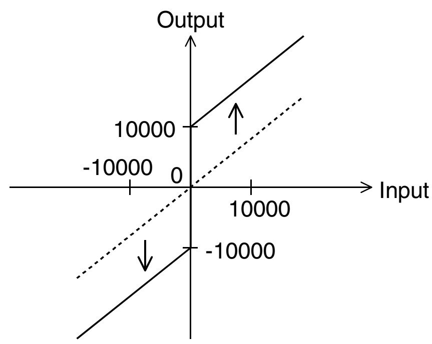
2. Program example 2

In the program example shown below, the BCD data set in X020 to X057 is controlled by the zone from "-10000" to "+10000", and the controlled value is output to D11 and D10 when X000 turns ON.



Operation

- In the case of "(D1, D0) < 0", "(D1, D0) + (-10000)" is output to (D11, D10).
- In the case of "(D1, D0) = 0", the "0" is output to (D11, D10).
- In the case of "(D1, D0) > 0", "(D1, D0) + 10000" is output to (D11, D10).



29.4 FNC259 – SCL / Scaling (Coordinate by Point Data)

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction executes scaling of the input value using a specified data table, and outputs the result.
SCL2 (FNC269) is also available with a different data table configuration for scaling.

→ **For SCL2 (FNC269) instruction, refer to Section 29.7.**

1. Instruction format

D	FNC 259 SCL	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	SCL	Continuous Operation
	SCLP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DSCL	Continuous Operation
	DSCLP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
(S1•)	Input value used in scaling or device number storing the input value	16- or 32-bit binary
(S2•)	Head device number storing the conversion table used in scaling	
(D•)	Device number storing the output value controlled by scaling	

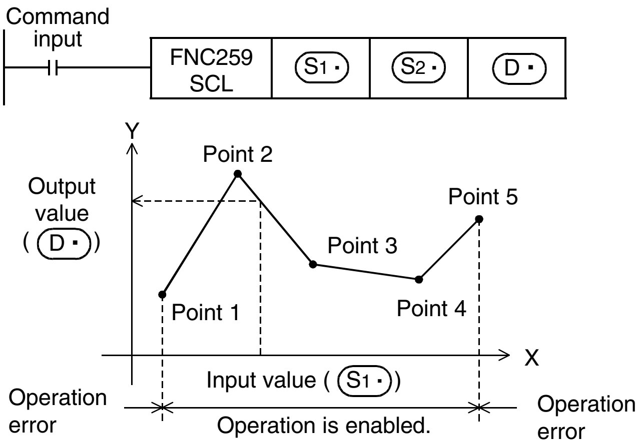
3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S1•)								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓			
(S2•)														✓	✓				✓					
(D•)									✓	✓	✓	✓	✓	✓	✓	✓			✓					

Explanation of function and operation

1. 16-bit operation (SCL and SCLP)

The input value specified in (S1•) is processed by scaling for the specified conversion characteristics, and stored to a device number specified in (D•). Conversion for scaling is executed based on the data table stored in a device specified in (S2•) and later.
If the output data is not an integer, however, the number in the first decimal place is rounded.
→ **For the method to set the conversion table for scaling, refer to the next page.**

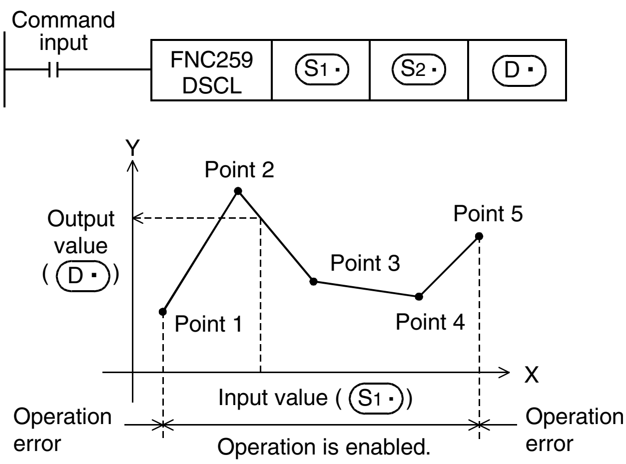


Conversion setting data table for scaling

Set item		Device assignment in setting data table
Number of coordinate points ("5" in the case shown in the left figure)		S2•
Point 1	X coordinate	S2• +1
	Y coordinate	S2• +2
Point 2	X coordinate	S2• +3
	Y coordinate	S2• +4
Point 3	X coordinate	S2• +5
	Y coordinate	S2• +6
Point 4	X coordinate	S2• +7
	Y coordinate	S2• +8
Point 5	X coordinate	S2• +9
	Y coordinate	S2• +10

2. 32-bit operation (DSCL and DSCLP)

The input value specified in [S1• +1, S1•] is processed by scaling for the specified conversion characteristics, and stored to a device number specified in [D• +1, D•]. Conversion for scaling is executed based on the data table stored in a device specified in [S2• +1, S2•] and later.
If the output data is not an integer, however, the number in the first decimal place is rounded.



Conversion setting data table for scaling

Set item		Device assignment in setting data table
Number of coordinate points ("5" in the case shown in the left figure)		[S2• +1, S2•]
Point 1	X coordinate	[S2• +3, S2• +2]
	Y coordinate	[S2• +5, S2• +4]
Point 2	X coordinate	[S2• +7, S2• +6]
	Y coordinate	[S2• +9, S2• +8]
Point 3	X coordinate	[S2• +11, S2• +10]
	Y coordinate	[S2• +13, S2• +12]
Point 4	X coordinate	[S2• +15, S2• +14]
	Y coordinate	[S2• +17, S2• +16]
Point 5	X coordinate	[S2• +19, S2• +18]
	Y coordinate	[S2• +21, S2• +20]

3. Setting the conversion table for scaling

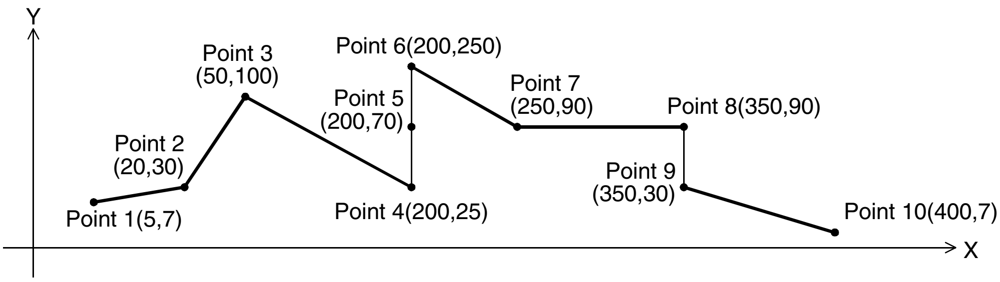
The conversion table for scaling is set based on the data table stored in a device specified in [S2• +1, S2•] and later.
The data table has the following configuration:

→ **For a setting example, refer to the next page.**

Set item		Device assignment in setting data table	
		16-bit operation	32-bit operation
Number of coordinate points		S2•	[S2• +1, S2•]
Point 1	X coordinate	S2• +1	[S2• +3, S2• +2]
	Y coordinate	S2• +2	[S2• +5, S2• +4]
Point 2	X coordinate	S2• +3	[S2• +7, S2• +6]
	Y coordinate	S2• +4	[S2• +9, S2• +8]
⋮	⋮	⋮	⋮
Point n (last)	X coordinate	S2• +2n-1	[S2• +4n-1, S2• +4n-2]
	Y coordinate	S2• +2n	[S2• +4n+1, S2• +4n]

Setting example of the conversion table for scaling

A setting example for the 16-bit operation is shown below.
For the 32-bit operation, set each item using a 32-bit binary value.
In the case of the conversion characteristics for scaling shown in the figure below, set the following data table.



Setting the conversion setting data table for scaling

Set item		Setting device and setting contents			Remarks
		When R0 is specified in (S2•)		Setting contents	
Number of coordinate points		(S2•)	R0	K10	
Point 1	X coordinate	(S2•)+1	R1	K5	
	Y coordinate	(S2•)+2	R2	K7	
Point 2	X coordinate	(S2•)+3	R3	K20	
	Y coordinate	(S2•)+4	R4	K30	
Point 3	X coordinate	(S2•)+5	R5	K50	
	Y coordinate	(S2•)+6	R6	K100	
Point 4	X coordinate	(S2•)+7	R7	K200	When coordinates are specified using three points in this way, the output value can be set to an intermediate value. In this example, the output value (intermediate value) is specified by the Y coordinate of the point 5. If the X coordinate is the same at three points or more, the value at the second point is also output.
	Y coordinate	(S2•)+8	R8	K25	
Point 5	X coordinate	(S2•)+9	R9	K200	
	Y coordinate	(S2•)+10	R10	K70	
Point 6	X coordinate	(S2•)+11	R11	K200	
	Y coordinate	(S2•)+12	R12	K250	
Point 7	X coordinate	(S2•)+13	R13	K250	
	Y coordinate	(S2•)+14	R14	K90	
Point 8	X coordinate	(S2•)+15	R15	K350	When coordinates are specified using two points in this way, the output value is the Y coordinate at the next point. In this example, the output value is specified by the Y coordinate of the point 9.
	Y coordinate	(S2•)+16	R16	K90	
Point 9	X coordinate	(S2•)+17	R17	K350	
	Y coordinate	(S2•)+18	R18	K30	
Point 10	X coordinate	(S2•)+19	R19	K400	
	Y coordinate	(S2•)+20	R20	K7	

Errors

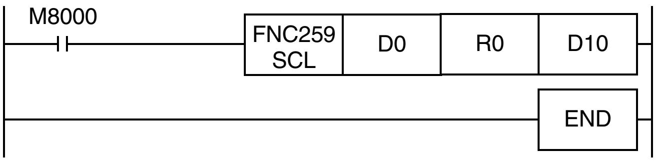
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the Xn data is not set in the ascending order in the data table (error code: K6706)
  The data table is searched from the low-order side of device numbers in the data table in the operation. Accordingly, even if only some Xn data is set in the ascending order in the data table, the instruction is executed without operation error up to the area of the data table in which the Xn data is set in the ascending order.
- When (S1•) is outside the data table (error code: K6706)
- When the value exceeds the 32-bit data range in the middle of operation (error code: K6706)
  In this case, check whether the distance between points is not “65535” or more.
  If the distance is “65535” or more, reduce the distance between points.

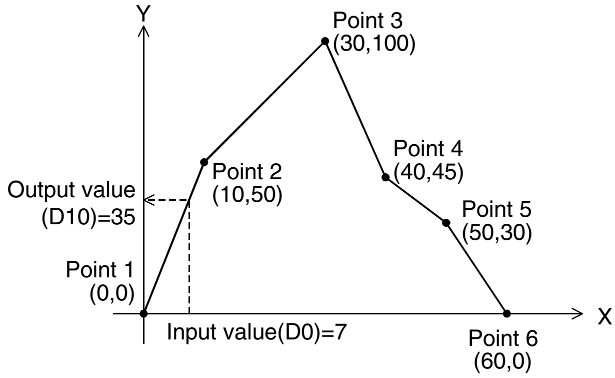
Program example

In the program example shown below, the value input to D0 is processed by scaling based on the conversion table for scaling set in R0 and later, and output to D10.

Program



Operation



Conversion setting data table for scaling

Set item		Device	Setting contents
Number of coordinate points		R0	K6
Point 1	X coordinate	R1	K0
	Y coordinate	R2	K0
Point 2	X coordinate	R3	K10
	Y coordinate	R4	K50
Point 3	X coordinate	R5	K30
	Y coordinate	R6	K100
Point 4	X coordinate	R7	K40
	Y coordinate	R8	K45
Point 5	X coordinate	R9	K50
	Y coordinate	R10	K30
Point 6	X coordinate	R11	K60
	Y coordinate	R12	K0

29.5 FNC260 – DABIN / Decimal ASCII to BIN Conversion

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

This instruction converts numeric data expressed in decimal ASCII codes (30H to 39H) into binary data.

1. Instruction format

D	FNC 260 DABIN	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	DABIN	Continuous Operation
	DABINP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DDABIN	Continuous Operation
	DDABINP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Head device number storing data (ASCII codes) to be converted into binary data	Character string
D·	Device number storing conversion result	16- or 32-bit binary

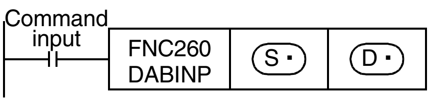
3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·												✓	✓	✓	✓				✓					
D·									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					

Explanation of function and operation

1. 16-bit operation (DABIN and DABINP)

1) Data stored in S· to S·+2 expressed in decimal ASCII codes (30H to 39H) is converted into 16-bit binary data, and stored in D·.



	b15 ------ b8	b7 ------ b0
S· +0	ASCII code for 10000's digit	Sign data
+1	ASCII code for 100's digit	ASCII code for 1000's digit
+2	ASCII code for 1's digit	ASCII code for 10's digit

⇒ D· (b15 ------- b0): 16-bit binary data

For example, when S· to S·+2 store ASCII codes expressing "-25108", 16-bit binary data is stored in D· as follows:

	b15 ------ b8	b7 ------ b0
S· +0	32H(2)	2DH(-)
+1	31H(1)	35H(5)
+2	38H(8)	30H(0)

⇒ D· (b15 ------- b0): -25108

2) The numeric range of data stored in S· to S·+2 is from -32768 to +32767.
3) As "sign data" (low-order byte of S·), "20H (space)" is set when the data to be converted is positive, and "2DH (-)" is set when the data to be converted is negative.
4) An ASCII code for each digit is within the range from 30H to 39H.
5) When an ASCII code for each digit is "20H (space)" or "00H (null)", it is handled as "30H".

2. 32-bit operation (DDABIN and DDABINP)

1) Data stored in (S·) to (S·)+5 expressed in decimal ASCII codes (30H to 39H) is converted into 32-bit binary data, and stored in [(D·)+1, (D·)].

Command input

FNC260 DDABINP	(S·)	(D·)

	b15–––––––––b8	b7–––––––––b0
(S·)+0	ASCII code for 1,000,000,000's digit	Sign data
+1	ASCII code for 10,000,000's digit	ASCII code for 100,000,000's digit
+2	ASCII code for 100,000's digit	ASCII code for 1,000,000's digit
+3	ASCII code for 1000's digit	ASCII code for 10000's digit
+4	ASCII code for 10's digit	ASCII code for 100's digit
+5	(Ignored)	ASCII code for 1's digit

⇨

(D·)+1 b31 –––– b16	(D·) b15 –––– b0
High-order 16 bits	Low-order 16 bits

32-bit data

For example, when (S·) to (S·)+5 store ASCII codes expressing "-1,234,543,210", 32-bit binary data is stored in [(D·)+1, (D·)] as follows:

	b15–––––b8	b7–––––b0
(S·)+0	31H(1)	2DH(-)
+1	33H(3)	32H(2)
+2	35H(5)	34H(4)
+3	33H(3)	34H(4)
+4	31H(1)	32H(2)
+5	(Ignored)	30H(0)

⇨

(D·)+1	(D·)
-1234543210	

2) The numeric range of data stored in (S·) to (S·)+5 is from -2,147,483,648 to +2,147,483,647. The high-order byte of (S·)+5 is ignored.
3) As "sign data" (low-order byte of (S·)), "20H (space)" is set when the data to be converted is positive, and "2DH (-)" is set when the data to be converted is negative.
4) An ASCII code for each digit is within the range from 30H to 39H.
5) When an ASCII code for each digit is "20H (space)" or "00H (NULL)", it is handled as "30H".

Related instructions

Instruction	Description
ASCI(FNC 82)	Converts hexadecimal codes into ASCII codes.
HEX(FNC 83)	Converts ASCII codes into hexadecimal codes.
ESTR(FNC116)	Converts binary floating point data into a character string (ASCII codes) of specified number of digits.
EVAL(FNC117)	Converts a character string (ASCII codes) into binary floating point data.
BINDA(FNC261)	Converts binary data into decimal ASCII codes (30H to 39H).

Errors

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

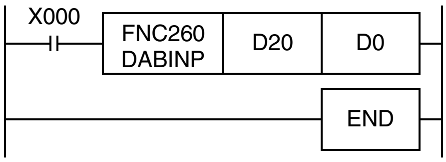
- When the sign data stored in (S·) is any value other than "20H (space)" or "2DH (-)" (error code: K6706)
- When an ASCII code for each digit stored in (S·) to (S·)+2(5) is any value other than "30H" to "39H", "20H (space)", or "00H (NULL)" (error code: K6706)
- When the numeric range of (S·) to (S·)+2(5) is outside the following range (error code: K6706)

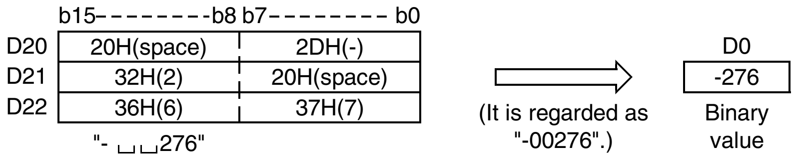
	Setting range
16-bit operation	–32768 to 32767
32-bit operation	–2,147,483,648 to 2,147,483,647

- When (S·) to (S·)+2(5) exceeds the device range (error code: K6706)

Program example

In the program below, the sign and decimal ASCII codes in five digits stored in D20 to D22 are converted into a binary value and stored in D0 when X000 is set to ON.





29.6 FNC261 – BINDA / BIN to Decimal ASCII Conversion

FX3UC Ver.2.20 ⇒ FX3U Ver.2.20 ⇒

Outline

This instruction converts binary data into decimal ASCII codes (30H to 39H).

1. Instruction format

	FNC 261	
D	BINDA	P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	BINDA	Continuous Operation
	BINDAP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
9 steps	DBINDA	Continuous Operation
	DBINDAP	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S·	Device number storing binary data to be converted into ASCII codes	16- or 32-bit binary
D·	Head device number storing conversion result	Character string

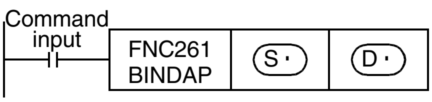
3. Applicable devices

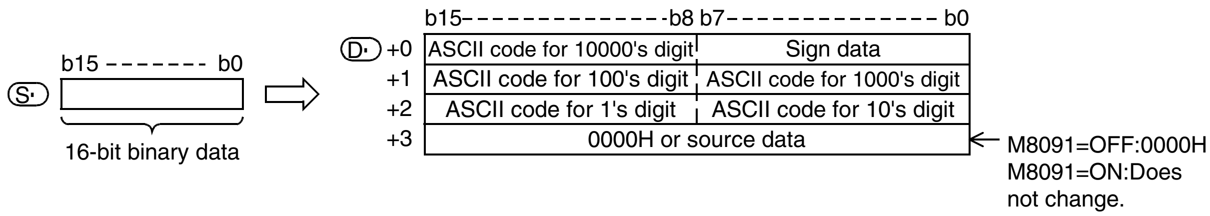
Operand Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
D·												✓	✓	✓	✓				✓					

Explanation of function and operation

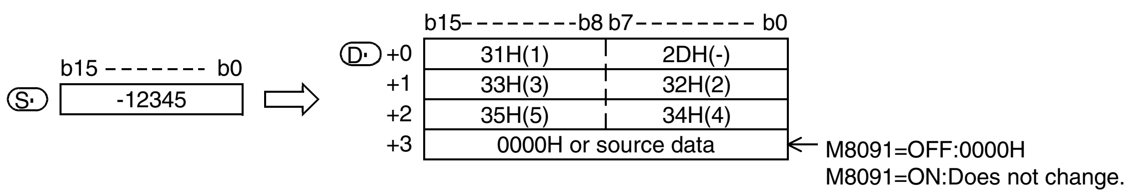
1. 16-bit operation (BINDA and BINDAP)

1) Each digit of 16-bit binary data stored in S· is converted into an ASCII code (30H to 39H), and stored in D· and later.





For example, when S· stores "-12345", the conversion result is stored in D· and later as follows:

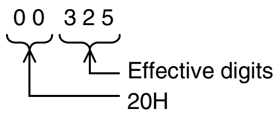


2) The numeric range of 16-bit binary data stored in S· is from -32768 to +32767.

3) The conversion result stored in D· is as follows:

a) As "sign data" (low-order byte of D·), "20H (space)" is set when the 16-bit binary data stored in S· is positive, and "2DH (-)" is set when 16-bit binary data stored in S· is negative.

b) "20H (space)" is stored for "0" on the left side of the effective digits (zero suppression).

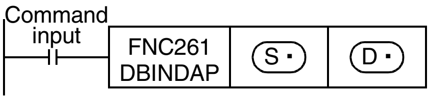


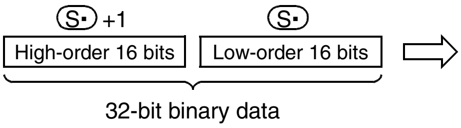
c) (D•) +3 is set as follows depending on the ON/OFF status of M8091.

ON/OFF status	Contents of processing
M8091=OFF	(D•) +3 is set to "00H (NULL)".
M8091=ON	(D•) +3 does not change.

2. 32-bit operation (DBINDA and DBINDAP)

1) Each digit of 32-bit binary data stored in [(S•)+1, (S•)] is converted into an ASCII code (30H to 39H), and stored in (D•) and later.

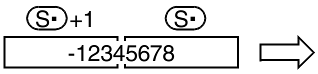




	b15 – b8	b7 – b0
(D•) +0	ASCII code for 1,000,000,000's digit	Sign data
+1	ASCII code for 10,000,000's digit	ASCII code for 100,000,000's digit
+2	ASCII code for 100,000's digit	ASCII code for 1,000,000's digit
+3	ASCII code for 1000's digit	ASCII code for 10000's digit
+4	ASCII code for 10's digit	ASCII code for 100's digit
+5	00H or 20H	ASCII code for 1's digit

M8091=OFF:00H
M8091=ON:20H

For example, when [(S•)+1, (S•)] stores "-12,345,678", the conversion result is stored in (D•) and later as follows:



	b15 – b8	b7 – b0
(D•) +0	20H (space)	2DH(-)
+1	31H(1)	20H (space)
+2	33H(3)	32H(2)
+3	35H(5)	34H(4)
+4	37H(7)	36H(6)
+5	00H or 20H	38H(8)

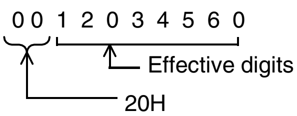
M8091=OFF:00H
M8091=ON:20H

2) The numeric range of 32-bit binary data stored in [(S•)+1, (S•)] is from -2,147,483,648 to +2,147,483,647.

3) The conversion result stored in (D•) is as follows:

a) "sign data" (low-order byte of (D•)) "20H (space)" is set when the 32-bit binary data stored in [(S•)+1, (S•)] is positive, and "2DH (-)" is set when 32-bit binary data stored in [(S•)+1, (S•)] is negative.

b) "20H (space)" is stored for "0" on the left side of the effective digits (zero suppression).



c) The high-order byte of (D•) +5 is set as follows depending on the ON/OFF status of M8091.

ON/OFF status	Contents of processing
M8091=OFF	The high-order byte of (D•) +5 is set to "00H (NULL)."
M8091=ON	The high-order byte of (D•) +5 is set to "20H (space)."

Related devices

Device	Name	Description
M8091	Output character quantity selector signal	• In the case of 16-bit operation - When M8091 is OFF, D· +3 is set to "00H (NULL)." - When M8091 is ON, D· +3 does not change. • In the case of 32-bit operation - When M8091 is OFF, the high-order byte of D· +5 is set to "00H (NULL)." - When M8091 is ON, the high-order byte of D· +5 is set to "20H (space)."

Related instructions

Instruction	Description
ASCI(FNC 82)	Converts hexadecimal codes into ASCII codes.
HEX(FNC 83)	Converts ASCII codes into hexadecimal codes.
ESTR(FNC116)	Converts binary floating point data into a character string (ASCII codes) of specified number of digits.
EVAL(FNC117)	Converts a character string (ASCII codes) into binary floating point data.
DABIN(FNC260)	Converts numeric data expressed in decimal ASCII codes (30H to 39H) into binary data.

Cautions

1. Occupied points of the device

The table below shows the occupied points of D· for 16-bit operation(BINDA/BINDAP) at M8091 ON/OFF and 32-bit operation (DBINDA/DBINDAP).

		Occupied Points of D·
16-bit operation	M8091=ON	3
	M8091=OFF	4
32-bit operation		6

Errors

An operation error is caused in the following case; The error flag M8067 turns ON, and the error code is stored in D8067.

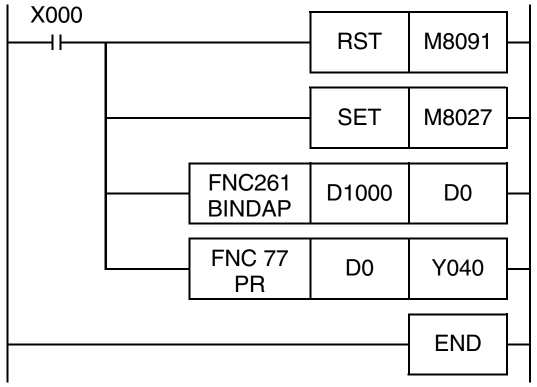
- When the occupied point of D· in storage location of ASCII code character string exceeds, the corresponding device rang (error code: K6706).

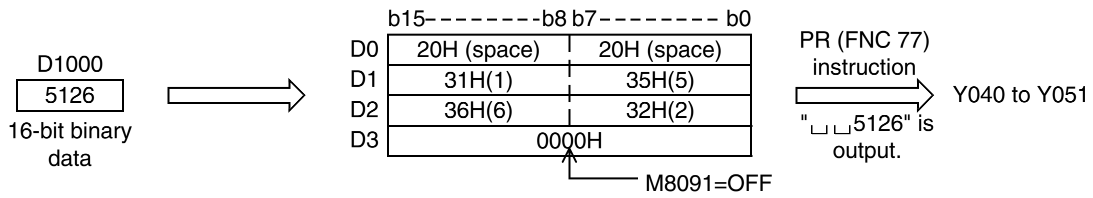
Program example

In the program below, 16-bit binary data stored in D1000 is converted into decimal ASCII codes when X000 is set to ON, and the ASCII codes converted by PR (FNC 77) instruction are output one by one in the time division method to Y040 to Y051.

By setting to OFF the output character selector signal M8091 and setting to ON PR mode flag M8027, ASCII codes up to "00H" are output.

→ **For PR mode flag and PR (FNC 77) instruction, refer to Section 15.8.**





29.7 FNC269 – SCL2 / Scaling 2 (Coordinate by X/Y Data)

Outline

FX3UC Ver.1.30 → FX3U Ver.2.20 →

This instruction executes scaling of the input value using a specified data table, and outputs the result.
SCL (FNC259) is also available with a different data table configuration for scaling.
SCL2 instruction is supported in the FX3UC Series Ver. 1.30 or later.

→ **For SCL (FNC259) instruction, refer to Section 29.4.**

1. Instruction format

D	FNC 269 SCL2	P

16-bit Instruction	Mnemonic	Operation Condition
7 steps	SCL2	Continuous Operation
	SCL2P	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
13 steps	DSCL2	Continuous Operation
	DSCL2P	Pulse (Single) Operation

2. Set data

Operand Type	Description	Data Type
S1•	Input value used in scaling or device number storing the input value	16- or 32-bit binary
S2•	Head device number storing the conversion table used in scaling	
D•	Device number storing the output value controlled by scaling	

3. Applicable devices

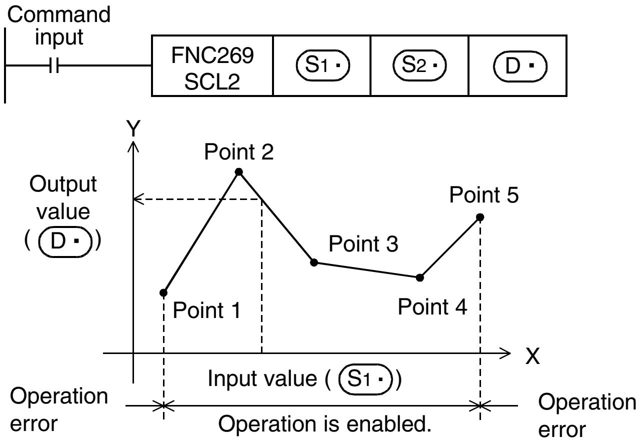
Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•								✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓			
S2•														✓	✓				✓					
D•									✓	✓	✓	✓	✓	✓	✓	✓			✓					

Explanation of function and operation

1. 16-bit operation (SCL2 and SCL2P)

The input value specified in S1• is processed by scaling for the specified conversion characteristics, and stored to a device number specified in D•. Conversion for scaling is executed based on the data table stored in a device specified in S2• and later.
If the output data is not an integer, however, the number in the first decimal place is rounded.

→ **For the method to set the conversion table for scaling, refer to the next page.**

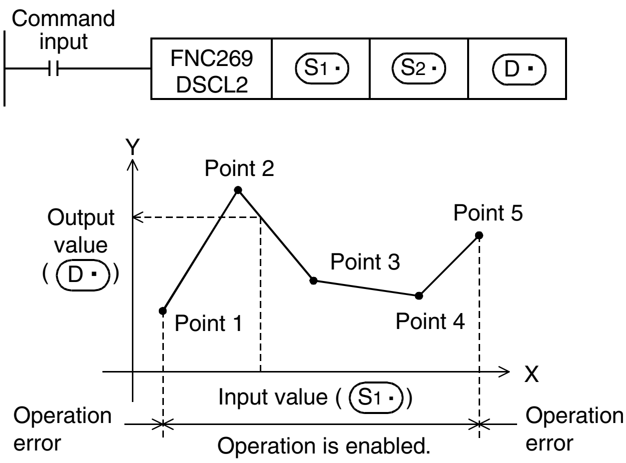


Conversion setting data table for scaling

Set item		Device assignment in setting data table
Number of coordinate points ("5" in the case shown in the left figure)		S2•
X coordinate	Point 1	S2• +1
	Point 2	S2• +2
	Point 3	S2• +3
	Point 4	S2• +4
	Point 5	S2• +5
Y coordinate	Point 1	S2• +6
	Point 2	S2• +7
	Point 3	S2• +8
	Point 4	S2• +9
	Point 5	S2• +10

2. 32-bit operation (DSCL2 and DSCL2P)

The input value specified in [S1• +1, S1•] is processed by scaling for the specified conversion characteristics, and stored to a device number specified in [D• +1, D•]. Conversion for scaling is executed based on the data table stored in a device specified in [S2• +1, S2•] and later.
If the output data is not an integer, however, the number in the first decimal place is rounded.



Conversion setting data table for scaling

Set item		Device assignment in setting data table
Number of coordinate points ("5" in the case shown in the left figure)		[S2• +1, S2•]
X coordinate	Point 1	[S2• +3, S2• +2]
	Point 2	[S2• +5, S2• +4]
	Point 3	[S2• +7, S2• +6]
	Point 4	[S2• +9, S2• +8]
	Point 5	[S2• +11, S2• +10]
Y coordinate	Point 1	[S2• +13, S2• +12]
	Point 2	[S2• +15, S2• +14]
	Point 3	[S2• +17, S2• +16]
	Point 4	[S2• +19, S2• +18]
	Point 5	[S2• +21, S2• +20]

3. Setting the conversion table for scaling

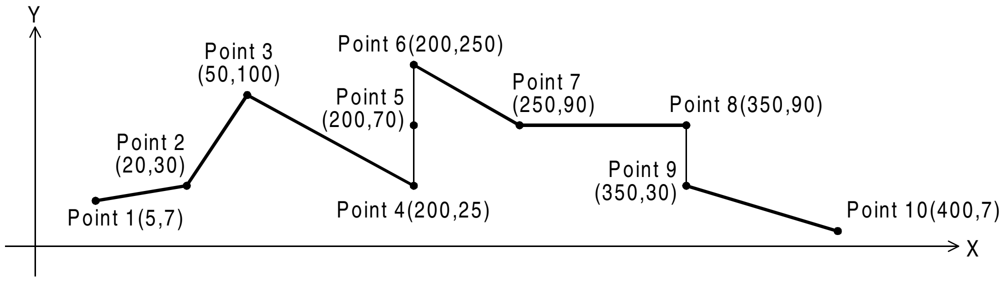
The conversion table for scaling is set based on the data table stored in a device specified in [S2• +1, S2•] and later.
The data table has the following configuration:

→ **For a setting example, refer to the next page.**

Set item		Device assignment in setting data table	
		16-bit operation	32-bit operation
Number of coordinate points		S2•	[S2• +1, S2•]
X coordinate	Point 1	S2• +1	[S2• +3, S2• +2]
	Point 2	S2• +2	[S2• +5, S2• +4]
	⋮	⋮	⋮
	Point n (last)	S2• +n	[S2• +2n+1, S2• +2n]
Y coordinate	Point 1	S2• +n+1	[S2• +2n+3, S2• +2n+2]
	Point 2	S2• +n+2	[S2• +2n+5, S2• +2n+4]
	⋮	⋮	⋮
	Point n (last)	S2• +2n	[S2• +4n+1, S2• +4n]

Setting example of the conversion table for scaling

A setting example for the 16-bit operation is shown below.
For the 32-bit operation, set each item using 32-bit binary value.
In the case of the conversion characteristics for scaling shown in the figure below, set the following data table.



Setting the conversion setting data table for scaling

Set item		Setting device and setting contents: When R0 is specified in (S2•)		Setting device and setting contents: Setting contents	Remarks
Number of coordinate points		(S2•)	R0	K10	
X coordinate	Point 1	(S2•) +1	R1	K5	
	Point 2	(S2•) +2	R2	K20	
	Point 3	(S2•) +3	R3	K50	
	Point 4	(S2•) +4	R4	K200	Refer to *1.
	Point 5	(S2•) +5	R5	K200	
	Point 6	(S2•) +6	R6	K200	
	Point 7	(S2•) +7	R7	K250	
	Point 8	(S2•) +8	R8	K350	Refer to *2.
	Point 9	(S2•) +9	R9	K350	
	Point 10	(S2•) +10	R10	K400	
Y coordinate	Point 1	(S2•) +11	R11	K7	
	Point 2	(S2•) +12	R12	K30	
	Point 3	(S2•) +13	R13	K100	
	Point 4	(S2•) +14	R14	K25	Refer to *1.
	Point 5	(S2•) +15	R15	K70	
	Point 6	(S2•) +16	R16	K250	
	Point 7	(S2•) +17	R17	K90	
	Point 8	(S2•) +18	R18	K90	Refer to *2.
	Point 9	(S2•) +19	R19	K30	
	Point 10	(S2•) +20	R20	K7	

*1. When coordinates are specified using three points as shown in the points 4, 5 and 6, the output value can be set to an intermediate value.
In this example, the output value (intermediate value) is specified by the Y coordinate of the point 5.
If the X coordinate is same at three points or more, the value at the second point is output also.

*2. When coordinates are specified using two points as shown in the points 8 and 9, the output value is the Y coordinate at the next point.
In this example, the output value is specified by the Y coordinate of the point 9.

Errors

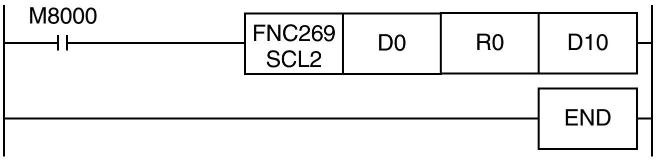
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the Xn data is not set in the ascending order in the data table (error code: K6706)
  The data table is searched from the low-order side of the device numbers in the data table in the operation. Accordingly, even if only some Xn data is set in the ascending order in the data table, the instruction is executed without operation error up to the area of the data table in which the Xn data is set in the ascending order.
- When S1· is outside the data table (error code: K6706)
- When the value exceeds the 32-bit data range in the middle of operation (error code: K6706)
  In this case, check whether the distance between points is not “65535” or more.
  If the distance is “65535” or more, reduce the distance between points.

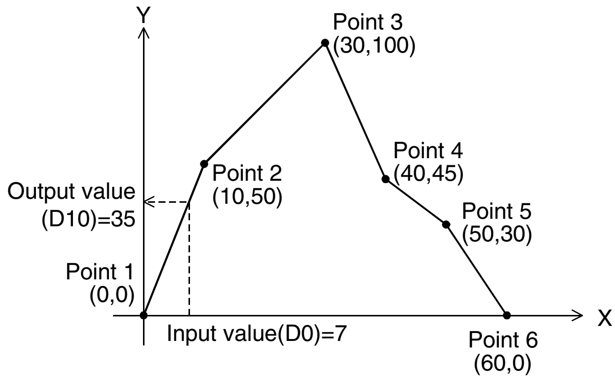
Program example

In the program example shown below, the value input to D0 is processed by scaling based on the conversion table for scaling set in R0 and later, and output to D10.

Program



Operation



Conversion setting data table for scaling

Set item		Device	Setting contents
Number of coordinate points		R0	K6
X coordinate	Point 1	R1	K0
	Point 2	R2	K10
	Point 3	R3	K30
	Point 4	R4	K40
	Point 5	R5	K50
	Point 6	R6	K60
Y coordinate	Point 1	R7	K0
	Point 2	R8	K50
	Point 3	R9	K100
	Point 4	R10	K45
	Point 5	R11	K30
	Point 6	R12	K0

30. External Device Communication (Inverter Communication) – FNC270 to FNC274

FNC270 to FNC274 provide instructions for controlling operations and reading/writing parameters while two or more FREQROL inverters are connected.

FNC No.	Mnemonic	Symbol	Function	Reference
270	IVCK	IVCK S1 S2 D n	Inverter Status Check	Section 30.1
271	IVDR	IVDR S1 S2 S3 n	Inverter Drive	Section 30.2
272	IVRD	IVRD S1 S2 D n	Inverter Parameter Read	Section 30.3
273	IVWR	IVWR S1 S2 S3 n	Inverter Parameter Write	Section 30.4
274	IVBWR	IVBWR S1 S2 S3 n	Inverter Parameter Block Write	Section 30.5

30.1 FNC270 – IVCK / Inverter Status Check

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction reads the operation status of an inverter to a PLC using the computer link operation function of the inverter. Applicable inverters vary depending on the version.
This instruction corresponds to the EXTR (K10) instruction in the FX2N/FX2NC Series.

→ **For detailed explanation of the instruction, refer to the Communication Control Manual.**

1. Instruction format

FNC 270 IVCK	16-bit Instruction	Mnemonic	Operation Condition	32-bit Instruction	Mnemonic	Operation Condition
	9 steps	IVCK —	Continuous Operation		—	—

2. Set data

Operand Type	Description	Data Type
S1·	Station number of an inverter (K0 to K31)	16-bit binary
S2·	Instruction code of an inverter (shown on the next page)	
D·	Device number storing the read value	
n	Used channel (K1: ch 1, K2: ch 2)	

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1·														✓	✓	✓			✓	✓	✓			
S2·														✓	✓	✓			✓	✓	✓			
D·									✓	✓	✓			✓	✓	✓			✓	✓	✓			
n																				✓	✓			

Explanation of function and operation

→ **For detailed explanation of the instruction, refer to the Communication Control Manual.**

1. 16-bit operation (IVCK)

The operation status corresponding to the instruction code*2 specified in S2· of an inverter*1 connected to a communication port n whose station number is specified in S1· is read and transferred to D·.

Command input

FNC270 IVCK	S1·	S2·	D·	n

*1. Mitsubishi FREQROL-F700*3/A700*3/V500/F500/A500/E500/S500 (having the communication function) Series general-purpose inverters

*2. Refer to the instruction code list shown on the next page.
Refer to pages in the inverter manual on which the computer link function is explained in detail.

*3. Mitsubishi FREQROL-F700/A700 Series general-purpose inverters are supported in Ver.2.20 and later.

2. Instruction codes of inverters

The table below shows the inverter instruction codes, (S2•), along with their functions.
For instruction codes, refer to pages in the inverter manual where the computer link function is explained in detail.

Instruction code of inverter (S2•)	Read contents	Corresponding inverter						
		F700	A700	V500	F500	A500	E500	S500
H7B	Operation mode	✓	✓	✓	✓	✓	✓	✓
H6F	Output frequency (number of rotations)	✓	✓	✓	✓	✓	✓	✓
H70	Output current	✓	✓	✓	✓	✓	✓	✓
H71	Output voltage	✓	✓	✓	✓	✓	✓	–
H72	Special monitor	✓	✓	✓	✓	✓	–	–
H73	Special monitor selection number	✓	✓	✓	✓	✓	–	–
H74	Abnormal contents	✓	✓	✓	✓	✓	✓	✓
H75	Abnormal contents	✓	✓	✓	✓	✓	✓	✓
H76	Abnormal contents	✓	✓	✓	✓	✓	✓	–
H77	Abnormal contents	✓	✓	✓	✓	✓	✓	–
H79	Inverter status monitor (extension)	✓	✓	–	–	–	–	–
H7A	Inverter status monitor	✓	✓	✓	✓	✓	✓	✓
H6E	Set frequency (read from EEPROM)	✓	✓	✓	✓	✓	✓	✓
H6D	Set frequency (read from RAM)	✓	✓	✓	✓	✓	✓	✓

3. Related devices

→ **For the instruction execution complete flag use method, refer to Subsection 6.5.2.**

Number		Description
ch1	ch2	
M8029		Instruction execution complete
M8063	M8438	Serial communication error
M8151	M8156	Inverter communicating*1
M8152	M8157	Inverter communication error*1
M8153	M8158	Inverter communication error latch*1
M8154	M8159	IVBWR instruction error*1

Number		Description
ch1	ch2	
D8063	D8438	Error code of serial communication error
D8150	D8155	Response wait time in inverter communication
D8151	D8156	Step number in inverter communication*2
D8152	D8157	Error code of inverter communication error*1
D8153	D8158	Latch of inverter communication error occurrence step*2
D8154	D8159	IVBWR instruction error parameter number*2

*1. Cleared when the PLC mode switches from STOP to RUN.

*2. Initial value: -1

Cautions

→ **For other cautions, refer to the Communication Control Manual.**

- It is not permitted to use the RS (FNC 80)/RS2 (FNC 87) instruction and an inverter communication instruction (FNC270 to FNC274) for the same port.
- Two or more inverter communication instructions (FNC270 to FNC274) can be driven for the same port at the same time.

Function change depending on the version

Inverter models available in this instruction are added depending on the version as shown in the table below.

Applicable version		Item	Outline of function
FX3U	FX3UC		
Ver.2.20 or later	Ver.2.20 or later	Addition of applicable models	Mitsubishi FREQROL-F700/A700 Series general-purpose inverters are supported.

30.2 FNC271 – IVDR / Inverter Drive

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction writes a control value required in the operation of an inverter using the computer link operation function of the inverter.
This instruction corresponds to the EXTR (K11) instruction in the FX2N/FX2NC Series.

→ **For detailed explanation of the instruction, refer to the Communication Control Manual.**

1. Instruction format

FNC 271 IVDR

16-bit Instruction	Mnemonic	Operation Condition
9 steps	IVDR	Continuous Operation
	—	

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
S1•	Station number of an inverter (K0 to K31)	16-bit binary
S2•	Instruction code of an inverter (shown on the next page)	
S3•	Set value to be written to a parameter of an inverter or device number storing the data to be set	
n	Used channel (K1: ch 1, K2: ch 2)	

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•														✓	✓	✓			✓	✓	✓			
S2•														✓	✓	✓			✓	✓	✓			
S3•								✓	✓	✓	✓			✓	✓	✓			✓					
n																				✓	✓			

Explanation of function and operation

→ **For detailed explanation of the instruction, refer to the Communication Control Manual.**

1. 16-bit operation (IVDR)

The control value specified in S3• is written to the instruction code*2 specified in S2• of an inverter*1 connected to a communication port n whose station number is specified in S1•.

Command input

FNC271 IVDR	S1•	S2•	S3•	n

*1. Mitsubishi FREQROL-F700*3/A700*3/V500/F500/A500/E500/S500 (having the communication function) Series general-purpose inverters

*2. Refer to the instruction code list shown on the next page.
Refer to pages in the inverter manual on which the computer link function is explained in detail.

*3. Mitsubishi FREQROL-F700/A700 Series general-purpose inverters are supported in Ver.2.20 and later.

2. Instruction codes of inverters

The table below shows the inverter instruction codes, (S2•), along with their functions.
For instruction codes, refer to pages in the inverter manual where the computer link function is explained in detail.

(Hexadecimal instruction code of inverter specified in (S2•)	Written contents	Corresponding inverter						
		F700	A700	V500	F500	A500	E500	S500
HFB	Operation mode	✓	✓	✓	✓	✓	✓	✓
HF3	Special monitor selection number	✓	✓	✓	✓	✓	–	–
HF9	Operation command (extension)	✓	✓	–	–	–	–	–
HFA	Operation command	✓	✓	✓	✓	✓	✓	✓
HEE	Set frequency (written to EEPROM)	✓	✓	✓	✓	✓	✓	✓
HED	Set frequency (written to RAM)	✓	✓	✓	✓	✓	✓	✓
HFD	Inverter reset	✓	✓	✓	✓	✓	✓	✓
HF4	Abnormal contents all clear	✓	✓	–	✓	✓	✓	✓
HFC	Parameter all clear	✓	✓	✓	✓	✓	✓	✓
HFC	User clear	✓	✓	–	✓	✓	–	–

3. Related devices

→ **For the instruction execution complete flag use method, refer to Subsection 6.5.2.**

Number		Description
ch1	ch2	
M8029		Instruction execution complete
M8063	M8438	Serial communication error
M8151	M8156	Inverter communicating*1
M8152	M8157	Inverter communication error*1
M8153	M8158	Inverter communication error latch*1
M8154	M8159	IVBWR instruction error*1

Number		Description
ch1	ch2	
D8063	D8438	Error code of serial communication error
D8150	D8155	Response wait time in inverter communication
D8151	D8156	Step number in inverter communication*2
D8152	D8157	Error code of inverter communication error*1
D8153	D8158	Latch of inverter communication error occurrence step*2
D8154	D8159	IVBWR instruction error parameter number*2

*1. Cleared when the PLC mode switches from STOP to RUN.

*2. Initial value: -1

Cautions

→ **For other cautions, refer to the Communication Control Manual.**

- It is not permitted to use the RS (FNC 80)/RS2 (FNC 87) instruction and an inverter communication instruction (FNC270 to FNC274) for the same port.
- Two or more inverter communication instructions (FNC270 to FNC274) can be driven for the same port at the same time.

Function change depending on the version

Inverter models available in this instruction are added depending on the version as shown in the table below.

Applicable version		Item	Outline of function
FX3U	FX3UC		
Ver.2.20 or later	Ver.2.20 or later	Addition of applicable models	Mitsubishi FREQROL-F700/A700 Series general-purpose inverters are supported.

30.3 FNC272 – IVRD / Inverter Parameter Read

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction reads a parameter of an inverter to a PLC using the computer link operation function of the inverter.
This instruction corresponds to the EXTR (K12) instruction in the FX2N/FX2NC Series.

→ **For detailed explanation of the instruction, refer to the Communication Control Manual.**

1. Instruction format

FNC 272 IVRD

16-bit Instruction	Mnemonic	Operation Condition
9 steps	IVRD —	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	—

2. Set data

Operand Type	Description	Data Type
S1•	Station number of an inverter (K0 to K31)	16-bit binary
S2•	Parameter number in an inverter	
D•	Device number storing the read value	
n	Used channel (K1: ch 1, K2: ch 2)	

3. Applicable devices

Operand Type	Bit Devices: System User							Word Devices: Digit Specification				Word Devices: System User				Special Unit	Index			Others: Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•														✓	✓	✓			✓	✓	✓			
S2•														✓	✓	✓			✓	✓	✓			
D•														✓	✓	✓			✓					
n																				✓	✓			

Explanation of function and operation

→ **For detailed explanation of the instruction, refer to the Communication Control Manual.**

1. 16-bit operation (IVRD)

The value of the parameter S2• is read from an inverter*1 connected to a communication port n whose station number is S1•, and output to D•.

Command input

FNC272 IVRD	S1•	S2•	D•	n

*1. Mitsubishi FREQROL-F700*3/A700*3/V500/F500/A500/E500/S500 (having the communication function) Series general-purpose inverters

*2. Mitsubishi FREQROL-F700/A700 Series general-purpose inverters are supported in Ver.2.20 and later.

2. Related devices

→ For the instruction execution complete flag use method, refer to Subsection 6.5.2.

Number ch1	Number ch2	Description
M8029		Instruction execution complete
M8063	M8438	Serial communication error
M8151	M8156	Inverter communicating*1
M8152	M8157	Inverter communication error*1
M8153	M8158	Inverter communication error latch*1
M8154	M8159	IVBWR instruction error*1

Number ch1	Number ch2	Description
D8063	D8438	Error code of serial communication error
D8150	D8155	Response wait time in inverter communication
D8151	D8156	Step number in inverter communication*2
D8152	D8157	Error code of inverter communication error*1
D8153	D8158	Latch of inverter communication error occurrence step*2
D8154	D8159	IVBWR instruction error parameter number*2

*1. Cleared when the PLC mode switches from STOP to RUN.

*2. Initial value: -1

Cautions

→ For other cautions, refer to the Communication Control Manual.

- It is not permitted to use the RS (FNC 80)/RS2 (FNC 87) instruction and an inverter communication instruction (FNC270 to FNC274) for the same port.
- Two or more inverter communication instructions (FNC270 to FNC274) can be driven for the same port at the same time.

Function change depending on the version

Inverter models available in this instruction are added depending on the version as shown in the table below.

Applicable version FX3U	Applicable version FX3UC	Item	Outline of function
Ver.2.20 or later	Ver.2.20 or later	Addition of applicable models	Mitsubishi FREQROL-F700/A700 Series general-purpose inverters are supported.

30.4 FNC273 – IVWR / Inverter Parameter Write

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction writes a parameter of an inverter using the computer link operation function of the inverter.
This instruction corresponds to the EXTR (K13) instruction in the FX2N/FX2NC Series.

→ **For detailed explanation of the instruction, refer to the Communication Control Manual.**

1. Instruction format

FNC 273
IVWR

16-bit Instruction	Mnemonic	Operation Condition
9 steps	IVWR	Continuous Operation
	—	

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
(S1•)	Station number of an inverter (K0 to K31)	16-bit binary
(S2•)	Parameter number in an inverter	
(S3•)	Set value to be written to a parameter of an inverter or device number storing the data to be set	
n	Used channel (K1: ch 1, K2: ch 2)	

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S1•)														✓	✓	✓			✓	✓	✓			
(S2•)														✓	✓	✓			✓	✓	✓			
(S3•)														✓	✓	✓			✓	✓	✓			
n																				✓	✓			

Explanation of function and operation

→ **For detailed explanation of the instruction, refer to the Communication Control Manual.**

1. 16-bit operation (IVWR)

A value specified in (S3•) is written to a parameter (S2•) in an inverter*1 connected to a communication port n whose station number is (S1•).

Command input

FNC273 IVWR	(S1•)	(S2•)	(S3•)	n

*1. Mitsubishi FREQROL-F700*3/A700*3/V500/F500/A500/E500/S500 (having the communication function) Series general-purpose inverters

*2. Mitsubishi FREQROL-F700/A700 Series general-purpose inverters are supported in Ver.2.20 and later.

2. Related devices

→ **For the instruction execution complete flag use method, refer to Subsection 6.5.2.**

Number		Description
ch1	ch2	
M8029		Instruction execution complete
M8063	M8438	Serial communication error
M8151	M8156	Inverter communicating*1
M8152	M8157	Inverter communication error*1
M8153	M8158	Inverter communication error latch*1
M8154	M8159	IVBWR instruction error*1

Number		Description
ch1	ch2	
D8063	D8438	Error code of serial communication error
D8150	D8155	Response wait time in inverter communication
D8151	D8156	Step number in inverter communication*2
D8152	D8157	Error code of inverter communication error*1
D8153	D8158	Latch of inverter communication error occurrence step*2
D8154	D8159	IVBWR instruction error parameter number*2

*1. Cleared when the PLC mode switches from STOP to RUN.

*2. Initial value: -1

Cautions

→ **For other cautions, refer to the Communication Control Manual.**

- It is not permitted to use the RS (FNC 80)/RS2 (FNC 87) instruction and an inverter communication instruction (FNC270 to FNC274) for the same port.
- Two or more inverter communication instructions (FNC270 to FNC274) can be driven for the same port at the same time.

Function change depending on the version

Inverter models available in this instruction are added depending on the version as shown in the table below.

Applicable version		Item	Outline of function
FX3U	FX3UC		
Ver.2.20 or later	Ver.2.20 or later	Addition of applicable models	Mitsubishi FREQROL-F700/A700 Series general-purpose inverters are supported.

30.5 FNC274 – IVBWR / Inverter Parameter Block Write

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction writes parameters of an inverter at one time using the computer link operation function of the inverter.

→ **For detailed explanation of the instruction, refer to the Communication Control Manual.**

1. Instruction format

FNC 274
IVBWR

16-bit Instruction	Mnemonic	Operation Condition
9 steps	IVBWR —	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
	— —	

2. Set data

Operand Type	Description	Data Type
S1•	Station number of an inverter (K0 to K31)	16-bit binary
S2•	Number of parameters in an inverter to be written at one time	
S3•	Head device number of a parameter table to be written to an inverter	
n	Used channel (K1: ch 1, K2: ch 2)	

3. Applicable devices

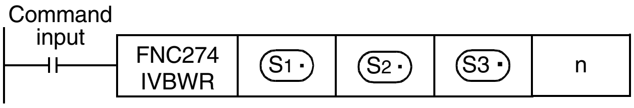
Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•														✓	✓	✓			✓	✓	✓			
S2•														✓	✓	✓			✓	✓	✓			
S3•														✓	✓	✓			✓					
n																				✓	✓			

Explanation of function and operation

→ For detailed explanation of the instruction, refer to the Data Communication Edition.

1. 16-bit operation (IVBWR)

A data table (parameter numbers and set values) specified in S2· and S3· is written at one time to an inverter[*1] connected to a communication port n whose station number is S1·.



*1. Mitsubishi FREQROL-F700[*3]/A700[*3]/V500/F500/A500/E500/S500 (having the communication function) Series general-purpose inverters

*2. The table below shows the data table format.
S2·: Number of parameters to be written
S3·: Head device number of data table

Device	Parameter numbers to be written and set values	
S3·	1st parameter	Parameter number
S3· +1		Set value
S3· +2	2nd parameter	Parameter number
S3· +3		Set value
⋮	⋮	⋮
S3· +2 S2· -4	"S2· -1"th parameter	Parameter number
S3· +2 S2· -3		Set value
S3· +2 S2· -2	"S2·"th parameter	Parameter number
S3· +2 S2· -1		Set value

*3. Mitsubishi FREQROL-F700/A700 Series general-purpose inverters are supported in Ver.2.20 and later.

2. Related devices

→ For the instruction execution complete flag use method, refer to Subsection 6.5.2.

Number ch1	Number ch2	Description
M8029		Instruction execution complete
M8063	M8438	Serial communication error
M8151	M8156	Inverter communicating[*1]
M8152	M8157	Inverter communication error[*1]
M8153	M8158	Inverter communication error latch[*1]
M8154	M8159	IVBWR instruction error[*1]

Number ch1	Number ch2	Description
D8063	D8438	Error code of serial communication error
D8150	D8155	Response wait time in inverter communication
D8151	D8156	Step number in inverter communication[*2]
D8152	D8157	Error code of inverter communication error[*1]
D8153	D8158	Latch of inverter communication error occurrence step[*2]
D8154	D8159	IVBWR instruction error parameter number[*2]

*1. Cleared when the PLC mode switches from STOP to RUN.

*2. Initial value: -1

Cautions

→ For other cautions, refer to the Communication Control Manual.

- It is not permitted to use the RS (FNC 80)/RS2 (FNC 87) instruction and an inverter communication instruction (FNC270 to FNC274) for the same port.
- Two or more inverter communication instructions (FNC270 to FNC274) can be driven for the same port at the same time.

Function change depending on the version

Inverter models available in this instruction are added depending on the version as shown in the table below.

Applicable version		Item	Outline of function
FX3U	FX3UC		
Ver.2.20 or later	Ver.2.20 or later	Addition of applicable models	Mitsubishi FREQROL-F700/A700 Series general-purpose inverters are supported.

31. Data Transfer 3 – FNC275 to FNC279

FNC275 to FNC279 provide instructions for executing more complicated processing for fundamental applied instructions and for special processing.

FNC No.	Mnemonic	Symbol	Function	Reference
275	–			
276	–			
277	–			
278	RBFM	RBFM m1 m2 D n1 n2	Divided BFM Read	Section 31.1
279	WBFM	WBFM m1 m2 S n1 n2	Divided BFM Write	Section 31.2

31.1 FNC278 – RBFM / Divided BFM Read

Outline

FX3UC Ver.2.20 → FX3U Ver.2.20 →

This instruction reads data from continuous buffer memories (BFM) in a special function block/unit over several operation cycles by the time division method. This instruction is convenient for reading receive data, etc. stored in buffer memories in a special function block/unit for communication by the time division method.
FROM (FNC 78) instruction is also available to read the buffer memory (BFM) data.

→ **For FROM (FNC 78) instruction, refer to Section 15.9.**

1. Instruction format

FNC 278
RBFM

16-bit Instruction	Mnemonic	Operation Condition
11 steps	RBFM	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
	–	

2. Set data

Operand Type	Description	Data Type
m1	Unit number [0 to 7]	16-bit binary data
m2	Head buffer memory (BFM) number [0 to 32766]	
D•	Head device number storing data to be read from buffer memory (BFM)	
n1	Number of all buffer memories (BFM) to be read [1 to 32767]	
n2	Number of points transferred in one operation cycle [1 to 32767]	

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
m1														✓	✓					✓	✓			
m2														✓	✓					✓	✓			
D•														▲	✓				✓					
n1														✓	✓					✓	✓			
n2														✓	✓					✓	✓			

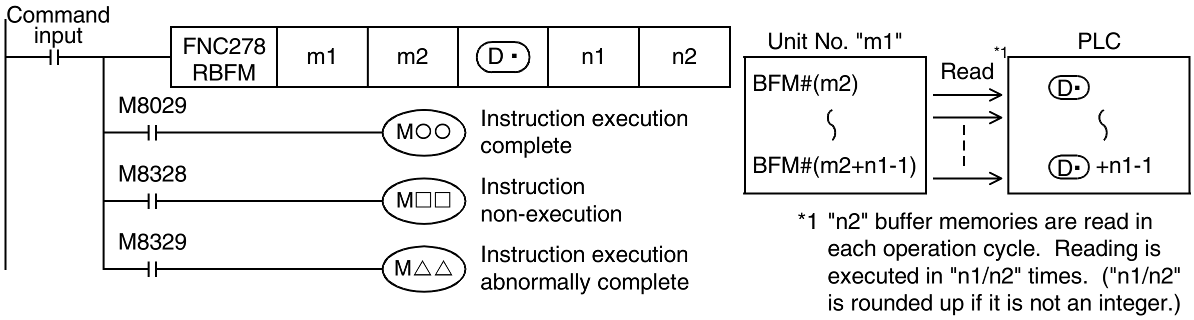
▲: Except special data register (D)

Explanation of function and operation

1. 16-bit operation (RBFM)

"n1" buffer memory (BFM) units at location # "m2" in special function unit/block No. "m1" are read to (D) in the PLC. While transfering, "n1" is divided by "n2" so n1/n2 buffer memories (rounded up when there is a remainder) are transferred per scan time

→ **For the unit No., buffer memory (BFM) #, cautions, and program example, refer to Subsection 31.1.1.**



*1 "n2" buffer memories are read in each operation cycle. Reading is executed in "n1/n2" times. ("n1/n2" is rounded up if it is not an integer.)

- When the instruction is finished normally, the instruction execution complete flag M8029 turns ON. When the instruction is finished abnormally, the instruction execution abnormally complete flag M8329 turns ON.
- When RBFM (FNC278) or WBFM (FNC279) instruction is executed in another step for the same unit number, the instruction non-execution flag M8328 is set to ON, and execution of such an instruction is paused.
  When execution of the other target instruction is complete, the paused instruction resumes.

Related devices

→ **For the flag use methods for instruction execution complete and instruction execution abnormally complete, refer to Subsection 6.5.2.**

Device	Name	Description
M8029	Instruction execution complete	Turns ON when an instruction is finished normally.
M8328	Instruction non-execution	Turns ON when RBFM (FNC278) or WBFM (FNC279) instruction in another step is executed for the same unit number.
M8329	Instruction execution abnormally complete	Turns ON when an instruction is finished abnormally.

Related instructions

Instruction	Description
FROM(FNC 78)	Read from a special function block
TO(FNC 79)	Write to a special function block
WBFM(FNC279)	Divided BFM write

Errors

An operation error is caused in the following case; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the unit number "m1" does not exist (error code: K6708)

31.1.1 Common items between RBFM (FNC278) instruction and WBFM (FNC279) instruction

Specification of unit number of special function block/unit and buffer memory

→ For the connection method of special extension units/blocks, number of connectable units/blocks, and handling of I/O numbers, refer to the manual of the PLC used and special function block/unit.

1. Unit number "m1" of a special extension unit/block

Use the unit number to specify to which equipment the RBFM/WBFM instruction works.
Setting range: K0 to K7

Unit No. 0 Built-in CC-Link/LT		Unit No. 1	Unit No. 2		Unit No. 3
FX3UC-32MT-LT main unit	I/O extension block	Special extension block	Special extension block	I/O extension block	Special extension block

A unit number is automatically assigned to each special extension unit/block connected to the PLC.
The unit number is assigned in the way "No. 0 → No. 1 → No. 2 ... starting from the equipment nearest the main unit.

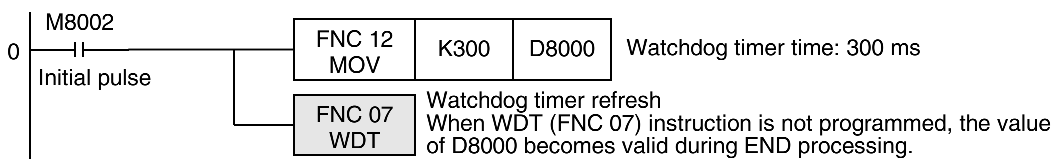
2. Buffer memory (BFM) number "m2"

Up to 32767 16-bit RAM memories are built in a special extension unit/block, and they are called buffer memories.
The buffer memory number is from "0" to "32766", and the contents are determined according to each special function unit/block.
Setting range: K0 to K32766

→ For the contents of buffer memories, refer to the manual of the special function block/unit used.

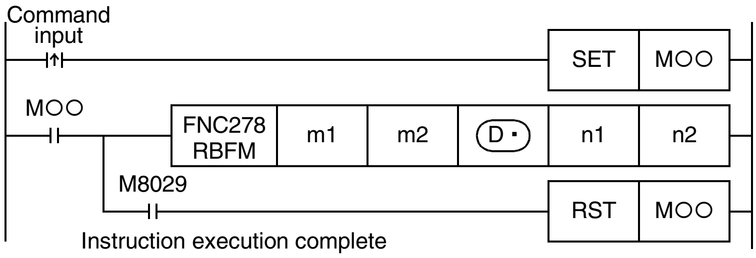
Cautions

- A watchdog timer error may occur when many numbers of points are transferred in one operation cycle. In such a case, take either of the following countermeasures:
  - Change the watchdog timer time
    By overwriting the contents of D8000 (watchdog timer time), the watchdog timer detection time is changed (initial value: K200).
    When the program shown below is input, the sequence program will be monitored with the new watchdog timer time.



  - Change the number of transferred points "n2" in each operation cycle
    Change the number of transferred points "n2" in each operation cycle to a smaller value.

- Do not stop the driving of the instruction while it is being executed. If driving is stopped, the buffer memory (BFM) reading/writing processing is suspended, but the data acquired in the middle of reading/writing processing is stored in (D·) and later and buffer memories (BFM).

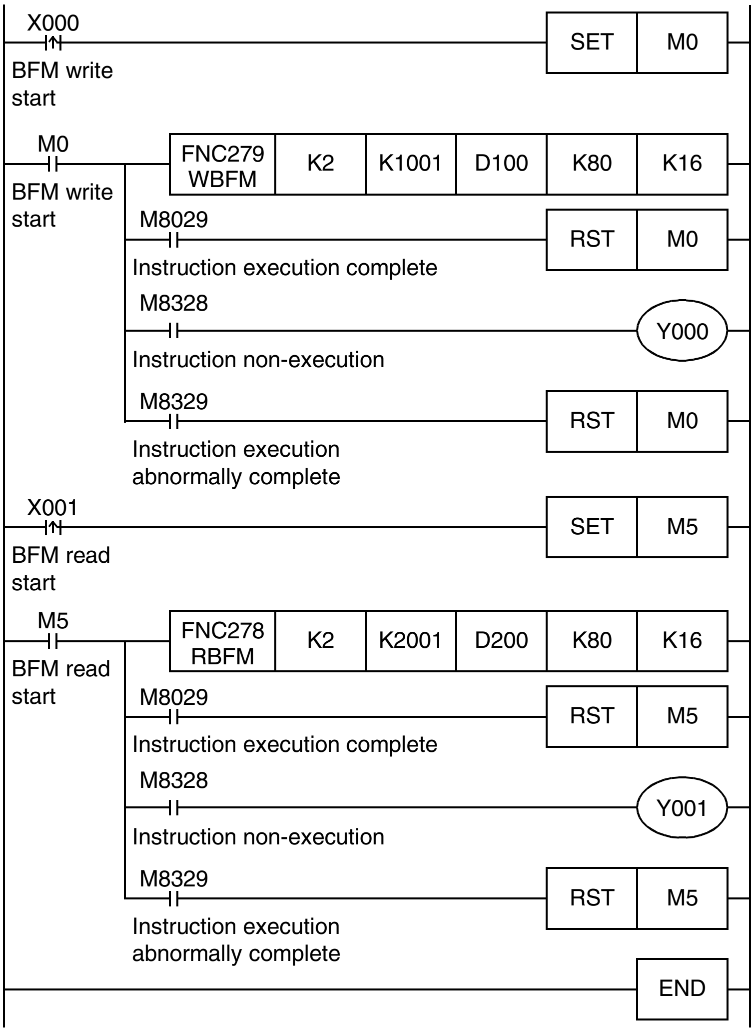


- When indexing is executed, the contents of index registers at the begining of execution are used. Even if the contents of index registers are changed after the instruction, such changes do not affect the process of the instruction.
- The contents of "n1" devices starting from (D·) change while RBFM (FNC278) instruction is executed. After execution of the instruction is completed, execute another instruction for "n1" devices starting from (D·).
- Do not update (change) the contents of "n1" devices starting from (S·) while WBFM (FNC279) instruction is executed. If the contents are updated, the intended data may not be written to the buffer memories (BFM).
- Do not update (change) the contents of "n1" buffer memories (BFM) starting from the buffer memory No. "m2" while RBFM (FNC278) instruction is executed. If the contents are updated, the intended data may not be read.

Program example

In the example shown below, data is read from and written to the buffer memories (BFM) in the unit No. 2 as follows:

- When X000 is set to ON, data stored in D100 to D179 (80 points) are written to the buffer memories (BFM) #1001 to 1080 in the special function block/unit whose unit number is No. 2 by 16 points in each operation cycle.
- When X001 is set to ON, the buffer memories (BFM) #2001 to 2080 (80 points) in the special function block/unit whose unit number is No. 2 are written to D200 to D279 by 16 points in each operation cycle.



The BFM write start flag is set to ON.

D100 to D179 (80 points) are written to the buffer memories #1001 to 1080 in the unit No. 2 (in 5 operation cycles).

The BFM write start flag is set to OFF.

Execution of the WBFM instruction is waited.

The BFM write start flag is set to OFF.

The BFM read start flag is set to ON.

The buffer memories #2001 to 2080 (80 points) in the unit No. 2 are read to D200 to D279 (in 5 operation cycles).

The BFM read start flag is set to OFF.

Execution of the RBFM instruction is waited.

The BFM read start flag is set to OFF.

31.2 FNC279 – WBFM / Divided BFM Write

FX3UC Ver.2.20 → FX3U Ver.2.20 →

Outline

This instruction writes data to continuous buffer memories (BFM) in a special function block/unit over several operation cycles by the time division method. This instruction is convenient for writing send data, etc. to buffer memories in a special function block/unit for communication by the time division method.
TO (FNC 79) instruction is also available for writing data to the buffer memory (BFM).

→ **For TO (FNC 79) instruction, refer to Section 15.10.**

1. Instruction format

FNC 279
WBFM

16-bit Instruction	Mnemonic	Operation Condition
11 steps	WBFM	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
–		

2. Set data

Operand Type	Description	Data Type
m1	Unit number [0 to 7]	16-bit binary data
m2	Head buffer memory (BFM) number [0 to 32766]	
S•	Head device number storing data to be written to buffer memory (BFM)	
n1	Number of all buffer memories (BFM) to be written [1 to 32767]	
n2	Number of points transferred in one operation cycle [1 to 32767]	

3. Applicable devices

Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
m1														✓	✓					✓	✓			
m2														✓	✓					✓	✓			
S•														▲	✓				✓					
n1														✓	✓					✓	✓			
n2														✓	✓					✓	✓			

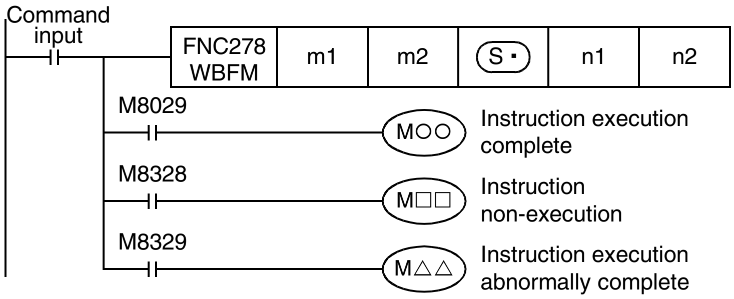
▲: Except special data register (D)

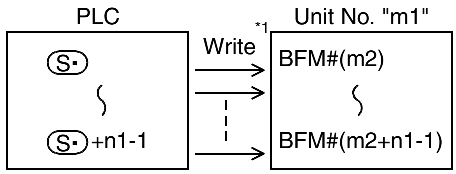
Explanation of function and operation

1. 16-bit operation (WBFM)

"n1" word units from (S) in the PLC are written to buffer memory (BFM) location # "m2" in special function unit/block No. "m1". While transfering, "n1" is divided by "n2" so n1/n2 words (rounded up when there is a remainder) are transferred per scan time.

→ **For the unit No., buffer memory (BFM) No., cautions, and program example, refer to Subsection 31.1.1.**





*1 "n2" points are written in each operation cycle. Writing is executed in "n1/n2" times. ("n1/n2" is rounded up if it is not an integer.)

- When the instruction is finished normally, the instruction execution complete flag M8029 turns ON. When the instruction is finished abnormally, the instruction execution abnormally complete flag M8329 turns ON.
- When RBFM (FNC278) or WBFM (FNC279) instruction is executed in another step for the same unit number, the instruction non-execution flag M8328 is set to ON, and execution of such an instruction is paused.
  When execution of the other target instruction is complete, the paused instruction resumes.

Related devices

→ **For the flag use methods for instruction execution complete and instruction execution abnormally complete, refer to Subsection 6.5.2.**

Device	Name	Description
M8029	Instruction execution complete	Turns ON when an instruction is finished normally.
M8328	Instruction non-execution	Turns ON when RBFM (FNC278) or WBFM (FNC279) instruction in another step is executed for a same unit number.
M8329	Instruction execution abnormally complete	Turns ON when an instruction is finished abnormally.

Related instructions

Instruction	Description
FROM(FNC 78)	Read from a special function block
TO(FNC 79)	Write to a special function block
RBFM(FNC278)	Divided BFM read

Errors

An operation error is caused in the following case; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the unit number "m1" does not exist (error code: K6708)

31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC·STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List

32. High Speed Processing 2 – FNC280 to FNC289

FNC No.	Mnemonic	Symbol	Function	Reference
280	HSCT	HSCT S1 m S2 D n	High Speed Counter Compare With Data Table	Section 32.1
281	–			–
282	–			–
283	–			–
284	–			–
285	–			–
286	–			–
287	–			–
288	–			–
289	–			–

32.1 FNC280 – HSCT / High Speed Counter Compare With Data Table

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction compares the current value of a high speed counter with preliminarily prepared data tables, and then sets or resets up to 16 output devices.

1. Instruction format

FNC 280
D HSCT

16-bit Instruction	Mnemonic	Operation Condition
	—	—

32-bit Instruction	Mnemonic	Operation Condition
21 steps	DHSCT	Continuous Operation
	—	

2. Set data

Operand Type	Description	Data Type
S1•	Head device number storing a data table	16- or 32-bit binary
m	Number of data tables (number of comparison points) [1 ≤ m ≤ 128]	16-bit binary
S2•	High speed counter number (C235 to C255)	32-bit binary
D•	Head device number to which the operation status is output	Bit
n	Number of devices to which the operation status is output [1 ≤ n ≤ 16]	16-bit binary

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S1•														✓	✓				✓					
m																				✓	✓			
S2•													▲						✓					
D•		✓	✓			✓													✓					
n																				✓	✓			

Only a high speed counter C235 to C255 can be specified in "▲".

Explanation of function and operation

1. 32-bit operation (DHSCT)

The current value of a high speed counter specified in S2• is compared with the data table shown below which has (3×m) points is stored in S1• and later, and the output status (ON or OFF) specified in the data table is output to D• to D•+n-1.

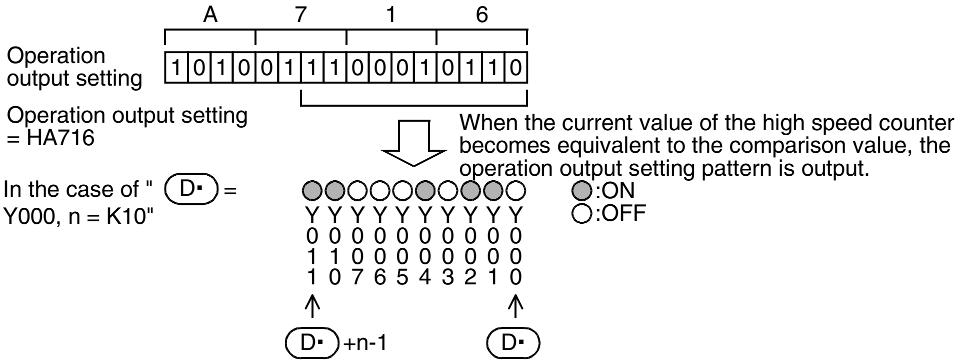
Command input

FNC280 DHSCT	S1•	m	S2•	D•	n

Data table used for comparison

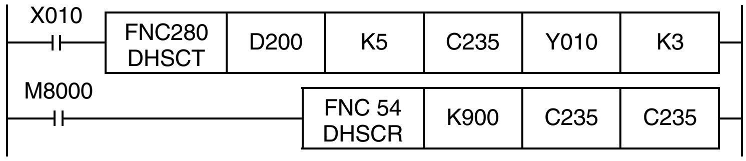
Comparison point number	Comparison data	Operation output setting (SET [1] or RESET [0])	Operation output destination
0	(S1•) +1, (S1•)	(S1•) +2	(D•) to (D•) +n-1
1	(S1•) +4, (S1•) +3	(S1•) +5	
2	(S1•) +7 (S1•) +6	(S1•) +8	
⋮	⋮	⋮	
m-2	(S1•) +3m-5, (S1•) +3m-6	(S1•) +3m-4	
m-1	(S1•) +3m-2, (S1•) +3m-3	(S1•) +3m-1	

Operation output setting (SET [1] or RESET [0]) [Up to 16 points]

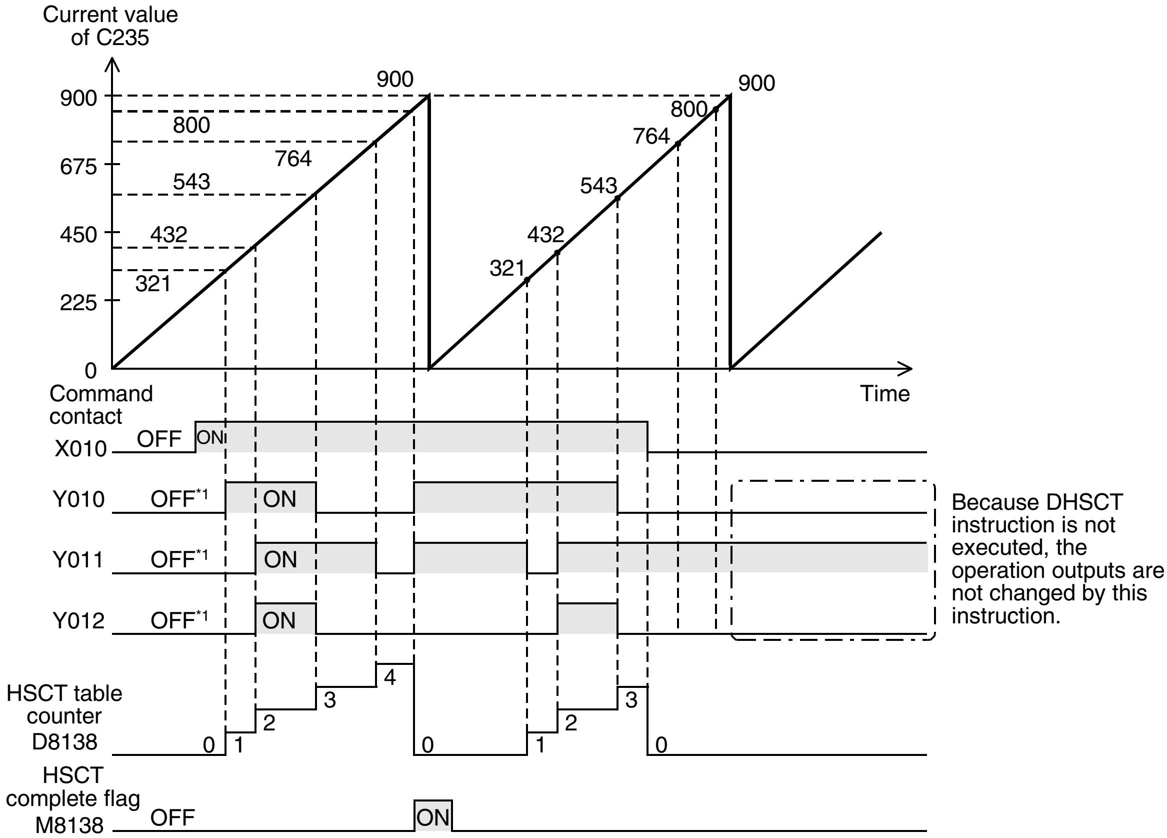


1) When this instruction is executed, the data table is set as the comparison target.
2) When the current value of a high speed counter, specified in (S2•), becomes equivalent to the comparison target data table, the operation output specified by the comparison target data table is output to (D•) to (D•) +n-1.
   If an output (Y) is specified in (D•), the output processing is executed immediately without waiting for the output refresh executed by the END instruction.
   When specifying an output (Y), make sure that the least significant digit of the device number is “0”.
   Examples: Y000, Y010 and Y020
3) Immediately after step 2), “1” is added to the current table counter value D8138.
4) The next comparison point is set as the comparison target data.
5) Steps 2) and 3) are repeated until the current value of the table counter D8138 becomes "m".
   When the current value becomes "m", the complete flag M8138 turns ON, and the execution returns to step 1). At this time, the table counter D8138 is reset to "0".
6) When the command contact is set to OFF, execution of the instruction is stopped and the table counter D8138 is reset to “0”.

Operation example



Comparison point number	Comparison data		SET/RESET pattern		Table counter D8138
	Device	Value	Device	Value	
0	D201,D200	K321	D202	H0001	0↓
1	D204,D203	K432	D205	H0007	1↓
2	D207,D206	K543	D208	H0002	2↓
3	D210,D209	K764	D211	H0000	3↓
4	D213,D212	K800	D214	H0003	4↓ (Repeated from “0↓”)



*1. If this instruction is not executed, no processing is executed for outputs.
In the operation example shown above, the command contact is “OFF”.

2. Related device

Device	Name	Description
M8138	HSCT(FNC280) complete flag	Turns ON when the operation for the final table No. "m-1" is completed.
D8138	HSCT(FNC280) table counter	Stores the table number handled as the comparison target.

Cautions

- This instruction can be executed only once in a program.
  If this instruction is programmed two or more times, an operation error is caused by the second instruction and later, and the instruction will not be executed. (error code: K6765)
- This instruction constructs the data table at the END instruction of the first execution of the instruction. Accordingly, the operation output works after the second scan and later.
- With regard to DHSCT (FNC280), DHSCS (FNC 53), DHSCR (FNC 54) and DHSZ (FNC 55) instructions, up to 32 instructions can be executed in one operation cycle. An operation error is caused by the 33rd instruction and later, and the instruction will not be executed. (error code: K6705)
- If an output (Y) is specified in D·, the output processing is executed immediately without waiting for the output refresh executed by END instruction.
  When specifying an output (Y), make sure that the least significant digit of the device number is "0".
  Examples: Y000, Y010 and Y020
- When a high speed counter specified in S2· is indexed with index, all high speed counters are handled as software counters.
- For this instruction, only one comparison point (one line) is handled as the comparison target at one time. Comparison will not move to the next comparison point until the current counter value becomes equivalent to the comparison point currently selected as the comparison target.
  If the current value of a high speed counter executes up counting using the comparison data table shown in the operation example on the previous page, make sure to execute the instruction while the current value of the high speed counter is smaller than the comparison value in comparison point No. 1.

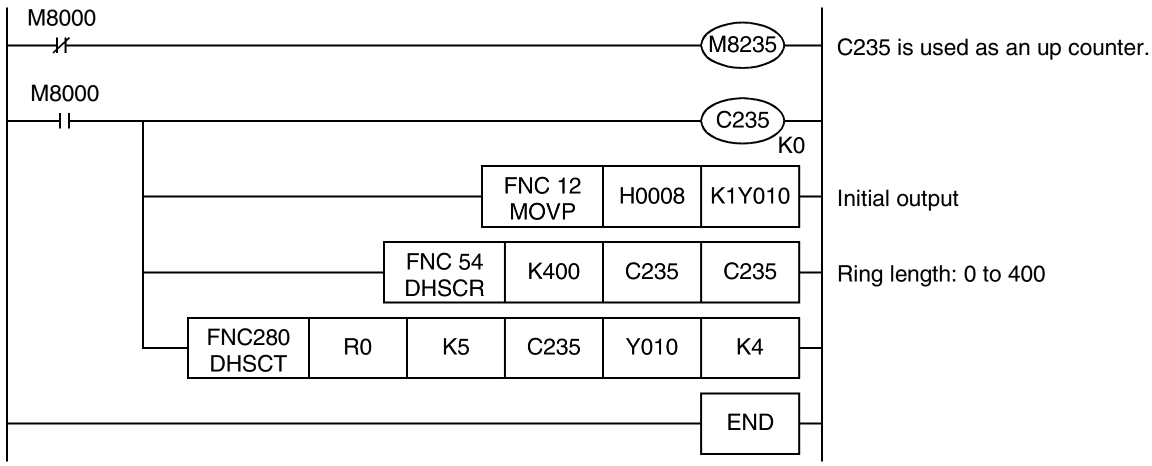
Errors

An operation error occurs in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When any devices other than high speed counters C235 to C255 are specified in S2· (error code: K6706)
- When the "3m-1"th device from a device specified in S1· exceeds the last number of the device (error code: K6706)
- When the "n"th device from a device specified in D· exceeds the last number of the device (error code: K6706)
- When this instruction is used two or more times in a program (error code: K6765)
- With regard to DHSCT (FNC280), DHSCS (FNC 53), DHSCR (FNC 54) and DHSZ (FNC 55) instructions, up to 32 instructions can be executed in one operation cycle. An operation error is caused by the 33rd instruction and later, and the instruction will not be executed. (error code: K6705)

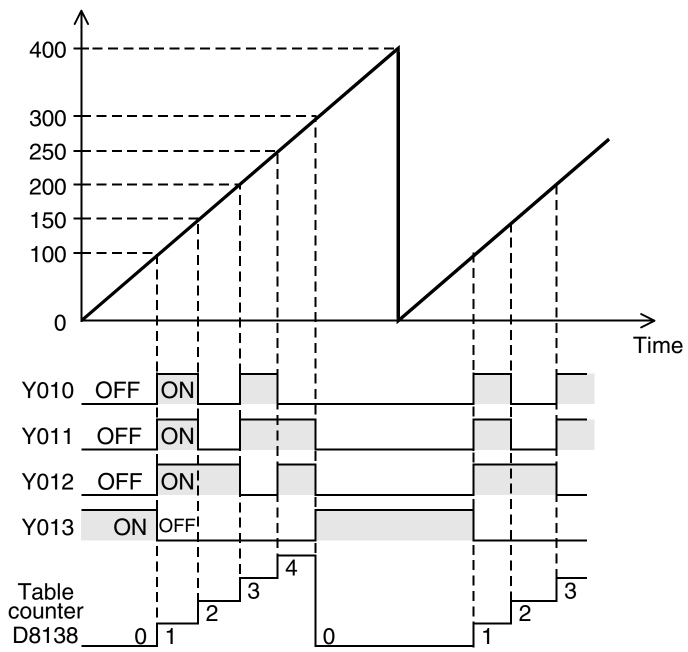
Program example

In the program shown below, the current value of C235 (counting X000) is compared with the comparison data table set in R0 and later, and a specified pattern is output to Y010 to Y013.



Operation example

Comparison point	Comparison data		SET/RESET pattern		Table counter D8138
	Device	Current value	Device	Current value	
0	R1,R0	K100	R2	H0007	0↓
1	R4,R3	K150	R5	H0004	1↓
2	R7,R6	K200	R8	H0003	2↓
3	R10,R9	K250	R11	H0006	3↓
4	R13,R12	K300	R14	H0008	4↓ (Repeated from “0↓”)



33. Extension File Register Control – FNC290 to FNC299

FNC No.	Mnemonic	Symbol	Function	Reference
290	LOADR	LOADR S n	Load From ER	Section 33.1
291	SAVER	SAVER S m D	Save to ER	Section 33.2
292	INITR	INITR S n	Initialize R and ER	Section 33.3
293	LOGR	LOGR S m D1 n D2	Logging R and ER	Section 33.4
294	RWER	RWER S n	Rewrite to ER	Section 33.5
295	INITER	INITER S n	Initialize ER	Section 33.6
296	–			–
297	–			–
298	–			–
299	–			–

33.1 FNC290 – LOADR / Load From ER

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction reads the current values of extension file registers (ER) stored in a memory cassette (flash memory), and transfers them to extension registers (R) stored in the PLC's built-in RAM.

1. Instruction format

FNC 290
LOADR

16-bit Instruction	Mnemonic	Operation Condition
5 steps	LOADR	Continuous Operation
	LOADRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
(S·)	Device number of extension register (transfer destination) to which data is to be transferred (The extension file register having the same number is handled as the data transfer source.)	16-bit binary
n	Number of points to be read (transferred) [0 ≤ n ≤ 32767]	

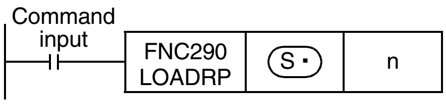
3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)															✓				✓					
n														✓						✓	✓			

Explanation of function and operation

1. 16-bit operation (LOADR and LOADRP)

The contents (current values) of extension file registers (ER) stored in a memory cassette (flash memory) having the same numbers with the extension registers specified by (S·) to (S·)+n-1 are read, and transferred to the extension registers specified by (S·) to (S·)+n-1 stored in the PLC's built-in RAM.



Extension file registers (ER) inside memory cassette	Read (transfer)	Extension registers (R) inside built-in RAM
E (S·)	→	(S·)
E (S·)+1	→	(S·)+1
E (S·)+2	→	(S·)+2
E (S·)+3	→	(S·)+3
≀	≀	≀
E (S·)+n-2	→	(S·)+n-2
E (S·)+n-1	→	(S·)+n-1

- Reading and transfer are executed in units of device. Up to 32768 devices can be read and transferred.
- Different from SAVER (FNC291), INITR (FNC292) and LOGR (FNC293) instructions, it is not necessary to execute this instruction in units of sector.
- If "n" is set to "0", it is handled as "32768" when the instruction is executed.

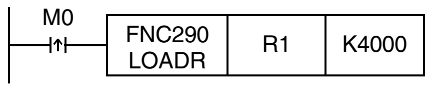
Errors

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the last device number to be transferred exceeds "32767" (error code: K6706)
  At this time, devices up to the last one (R32767) are read and transferred.
- When a memory cassette is not connected (error code: K6771)

Program example

In the program example shown below, the contents (current value) of 4000 extension file registers ER1 to ER4000 inside the memory cassette are read, and transferred to 4000 extension registers R1 to R4000 inside the built-in RAM.



Extension file registers (ER) inside memory cassette → Read (transfer) → Extension registers (R) inside built-in RAM

Device number	Current value	Read (transfer)	Device number	Current value
ER1	K100	→	R1	K100
ER2	K50	→	R2	K50
ER3	H0003	→	R3	H0003
ER4	H0101	→	R4	H0101
≀	≀	≀	≀	≀
ER3999	K55	→	R3999	K55
ER4000	K59	→	R4000	K59

33.2 FNC291 – SAVER / Save to ER

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction writes the current values of extension registers (R) stored in the PLC's built-in RAM to extension file registers (ER) stored in a memory cassette (flash memory) in units of sector (2048 points).
RWER (FNC294) instruction provided in FX3UC PLCs Ver.1.30 or later and FX3U PLCs writes (transfers) only arbitrary number of points. It is not necessary to execute INITR (FNC292) or INITER (FNC295) instruction every time when RWER instruction is used.

→ **For RWER instruction, refer to Section 33.5.**

1. Instruction format

FNC 291 SAVER

16-bit Instruction	Mnemonic	Operation Condition
7 steps	SAVER —	Continuous Operation

32-bit Instruction	Mnemonic	Operation Condition
	— —	

2. Set data

Operand Type	Description	Data Type
(S·)	Device number of extension register to which data is to be written (Only the head device number of a sector of extension registers can be specified.)	16-bit binary
n	Number of points written (transferred) in one operation cycle [0 ≤ n ≤ 2048]	
(D·)	Device number storing the number of already written points	

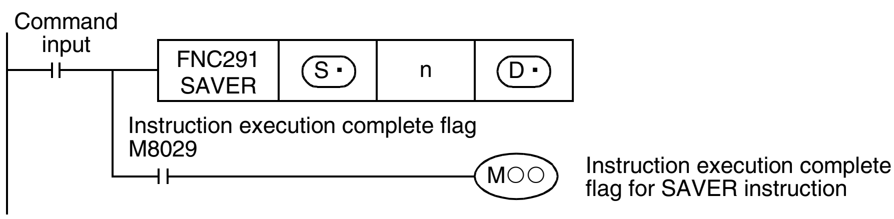
3. Applicable devices

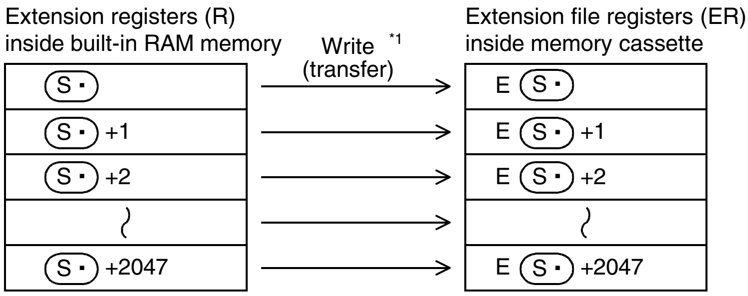
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S·)															✓				✓					
n																				✓	✓			
(D·)														✓					✓					

Explanation of function and operation

1. 16-bit operation (SAVER)

The contents (current values) of extension registers (R) specified by (S·) to (S·)+2047 are written (transferred) to extension file registers (ER) inside a memory cassette (flash memory) having the same device numbers in "2048/n" operation cycles ("2048/n+1" cycles if there is the remainder).
While the instruction is being executed, the number of already written points is stored in (D·).





*1 “n” points are written (transferred) in each operation cycle.

- Extension file registers are written in units of sector (2048 points).
  The table below shows the head device number in each sector:

Sector number	Head device number	Written device range	Sector number	Head device number	Written device range
Sector 0	R0	ER0 to ER2047	Sector 8	R16384	ER16384 to ER18431
Sector 1	R2048	ER2048 to ER4095	Sector 9	R18432	ER18432 to ER20479
Sector 2	R4096	ER4096 to ER6143	Sector 10	R20480	ER20480 to ER22527
Sector 3	R6144	ER6144 to ER8191	Sector 11	R22528	ER22528 to ER24575
Sector 4	R8192	ER8192 to ER10239	Sector 12	R24576	ER24576 to ER26623
Sector 5	R10240	ER10240 to ER12287	Sector 13	R26624	ER26624 to ER28671
Sector 6	R12288	ER12288 to ER14335	Sector 14	R28672	ER28672 to ER30719
Sector 7	R14336	ER14336 to ER16383	Sector 15	R30720	ER30720 to ER32767

- If "n" is set to "0", it is handled as "2048" when the instruction is executed.
- When writing (transfer) of 2048 points is finished, execution of the instruction is completed and the instruction execution complete flag M8029 turns ON.
- The number of already written points is stored in D·.

2. Related device

→ **For the instruction execution complete flag use method, refer to Subsection 6.5.2.**

Device number	Name	Description
M8029	Instruction execution complete flag	When execution of the target instruction is completed, the instruction execution complete flag M8029 turns ON. In a program, however, there may be two or more instructions which can use the flag M8029. To avoid confusion, make sure to use the NO contact of this flag immediately under SAVER instruction so that this flag works only for SAVER instruction.

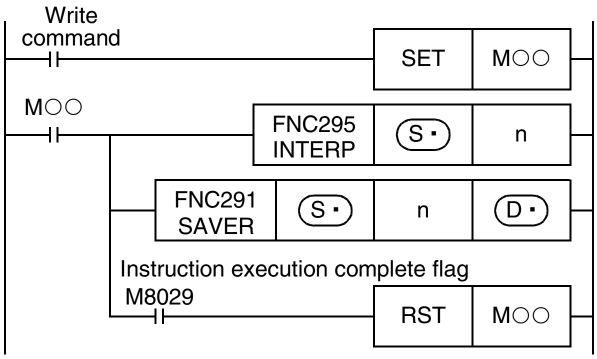
Cautions

1. Cautions on writing data to a memory cassette

Memory cassettes adopt flash memory. Note the following contents when writing data to extension file registers in a memory cassette with the FNC291 instruction.

- It takes about 340ms to write 2048 points. If “n” is set to K0 or K2048, the operation cycle for executing this instruction becomes longer than about 340ms.
  If the operation cycle is severely affected, write data in two or more operation cycles.
  When writing data in two or more operation cycles, set “n” in the range from K1 to K1024.

- Do not abort execution of this instruction in the middle of operation. If execution is aborted, unexpected data may be written to extension file registers.
  If execution of this instruction is aborted by turning OFF the power, execute the instruction again using step 2 described below after turning ON the power again.



- Execute INITER (FNC295) or INITR (FNC292) instruction to target extension file registers (ER) before executing SAVER instruction. If SAVER instruction is driven before INITER (FNC295) or INITR (FNC292) instruction is executed, an operation error (error code: K6770) may be caused.
  To avoid such an operation error, make a program for executing SAVER instruction in the following sequence:

When the FX3U/FX3UC PLC is Ver.1.30 or later

FX3UC Ver.1.30 FX3U Ver.2.20

[1] When storing data of 2048 extension registers (R) in one sector to extension file registers (ER)

1) Execute INITER (FNC295) instruction to extension file registers (ER) specified as targets in SAVER instruction.
2) Execute SAVER instruction.

[2] When storing the contents of an arbitrary number of extension registers (R) to extension file registers (ER)

Use RWER instruction.

→ **For RWER (FNC294) instruction, refer to Section 33.5.**

When the FX3UC PLC is former than Ver.1.30

FX3UC Ver.1.30

[1] When storing data of 2048 extension registers (R) in one sector to extension file registers (ER)

If the extension registers (R) have data to be stored in extension file registers (ER), use the procedure [2].

1) Execute INITR (FNC292) instruction to extension registers (R) and extension file registers (ER) specified as targets in SAVER instruction.
2) Store data to extension registers (R) specified as targets.
3) Execute SAVER instruction.

[2] When storing data of 2048 extension registers (R) in one sector to extension file registers (ER)

1) Temporarily withdraw the data of extension registers (R) specified as targets in SAVER instruction to data registers or unused 2048 extension registers (R) by using BMOV (FNC 15) instruction.
2) Execute INITR (FNC292) instruction to extension registers (R) and extension file registers (ER) specified as targets in SAVER instruction.
3) Return the data of 1024 points temporarily withdrawn in step 1) to extension registers (R) specified as targets by using BMOV (FNC 15) instruction.
4) Execute SAVER instruction.

Errors

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

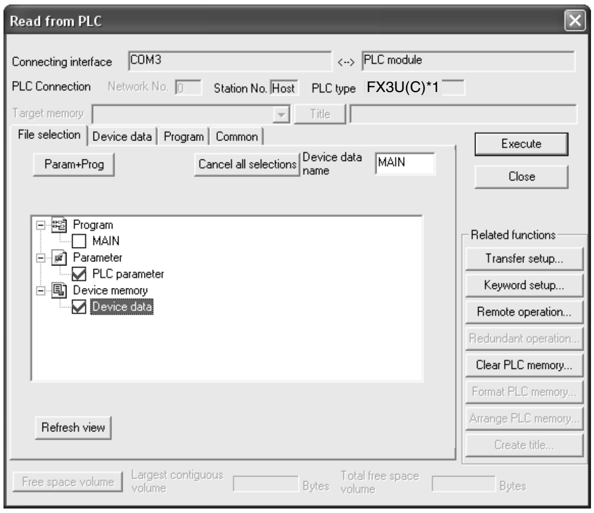
- When any device number other than the head device number of a sector of extension file registers is set to (S·) (error code: K6706)
- When a memory cassette is not connected (error code: K6771)
- When the protect switch of the memory cassette is set to ON (error code: K6770)
- When the collation result after data writing is "mismatch" due to omission of initialization or for another reason (error code: K6770)
  When this error occurs, the current values (data) of extension registers (R) may be lost. To avoid the data loss, back up the data of extension registers (R) in advance using the following procedure:

  1)Set the PLC mode to STOP.

  2)Create a new project in GX Developer.
  This step is not necessary if it is alright to overwrite the current project.

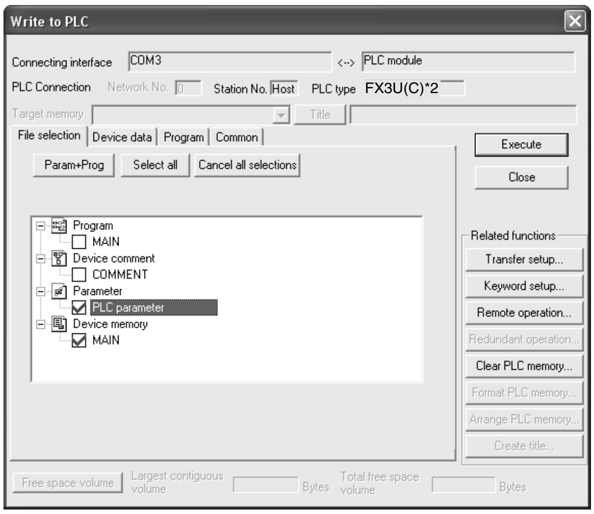
  3)Read the contents of extension registers (R) to GX Developer.
  [1]Select “Online” → “Read from PLC..." to display the “Read from PLC” window.
  [2]Click “Parameter” and “Device data” to put a check mark to each of them.
  [3]Click [Execute] button to execute reading.
  [4]When reading is completed, save the project.

  4)Change the current program inside the PLC to the program shown in “1. Cautions on writing data to a memory cassette” in “Cautions” on the previous page.



*1.For Ver. 8.13P to 8.24A of GX Developer, the PLC type is FX3UC.

  5)To the PLC, write the data which was temporarily withdrawn to GX Developer.
  [1]Select “Online” → “Write to PLC...” to display the “Write to PLC” window.
  [2]Click “Parameter” and “MAIN” to put a check mark to each of them.
  [3]Click [Execute] button to execute writing.



*2.For Ver. 8.13P to 8.24A of GX Developer, the PLC type is FX3UC.

  6)Change the PLC mode from STOP to RUN, execute the program, and store the data to extension file registers inside the memory cassette.

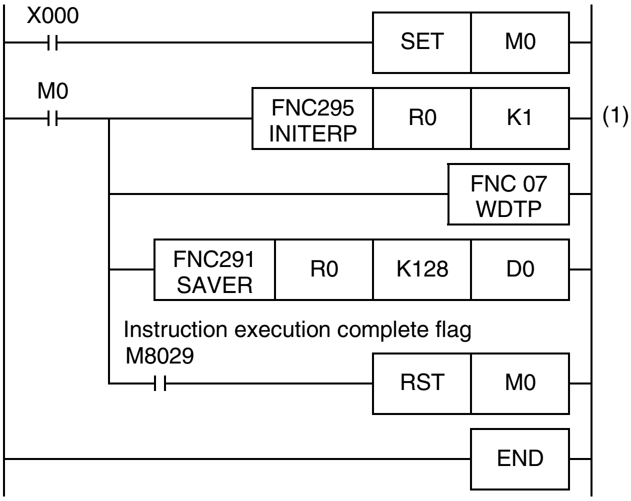
Program examples

1) In the case of FX3UC PLCs Ver. 1.30 or later and FX3U PLCs Ver. 2.20 or later

FX3UC Ver.1.30 → FX3U Ver.2.20 →

In the example shown below, only extension registers R10 to R19 (in sector 0) need to be updated in the extension file registers (ER). When X000 is set to ON, sector 0 (head device R0) is written to the extension file registers 128 points at a time. (128 points are written in one operation cycle)

Program



Operation

Setting data (A): R0 to R99; Changed data: R10 to R19; Unused: R100 to R2047

Setting data: Extension registers (R) Device number	Current value	Setting backup data: Extension file registers (ER) Device number	Current value
R0	K100	ER0	K100
R1	K105	ER1	K105
:	:	:	:
R10	K200	ER10	K300
R11	K215	ER11	K330
R12	K400	ER12	K350
:	:	:	:
R19	K350	ER19	K400
:	:	:	:
R99	K1000	ER99	K1000
R100	HFFFF	ER100	HFFFF
:	:	:	:
R2047	HFFFF	ER2047	HFFFF

(1) ⇨ INITER instruction

Setting backup data: Extension file registers (ER) Device number	Current value
ER0	HFFFF
ER1	HFFFF
:	:
ER10	HFFFF
ER11	HFFFF
ER12	HFFFF
:	:
ER19	HFFFF
:	:
ER99	HFFFF
ER100	HFFFF
:	:
ER2047	HFFFF

(2) ⇨ SAVER instruction

To the next page

(2) SAVER instruction

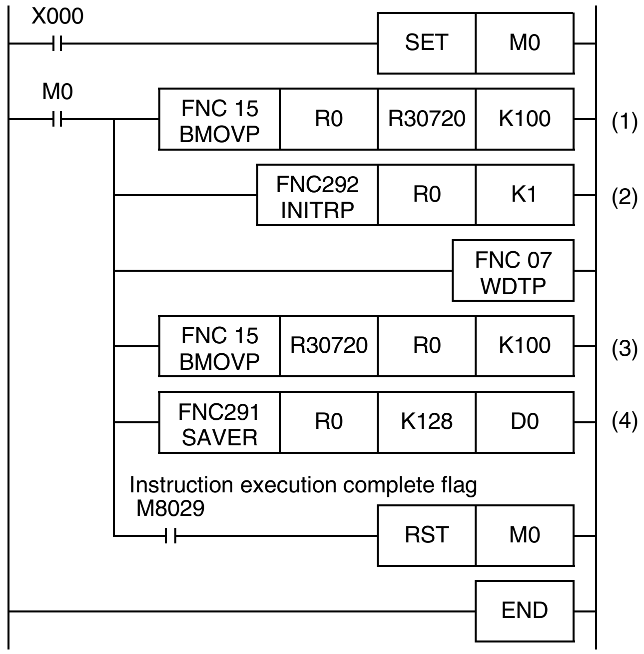
	Setting data: Extension registers (R) Device number	Current value		Setting backup data: Extension file registers (ER) Device number	Current value	Number of already written points (D0)
Setting data	R0	K100	1st operation cycle	ER0	K100	
Setting data	R1	K105		ER1	K105	
Setting data	:	:		:	:	
Setting data / Changed data	R10	K200		ER10	K200	
Setting data / Changed data	R11	K215		ER11	K215	
Setting data / Changed data	R12	K400		ER12	K400	
Setting data / Changed data	:	:		:	:	
Setting data / Changed data	R19	K350		ER19	K350	
Setting data	:	:		:	:	
Setting data	R99	K1000		ER1000	K1000	
Unused	R100	HFFFF		ER100	HFFFF	
Unused	:	:		:	:	K128
Unused	R127	HFFFF		ER127	HFFFF	
Unused	R128	HFFFF	2nd operation cycle	ER128	HFFFF	
Unused	:	:		:	:	
Unused	R255	HFFFF		ER255	HFFFF	K256
Unused	R256	HFFFF	3rd to 15th operation cycle	ER256	HFFFF	:
Unused	:	:		:	:	:
Unused	R1919	HFFFF		ER1919	HFFFF	K1920
Unused	R1920	HFFFF	16th operation cycle	ER1920	HFFFF	
Unused	:	:		:	:	
Unused	R2047	HFFFF		ER2047	HFFFF	K2048

2) In the case of FX3UC PLCs former than Ver.1.30

FX3UC Ver.1.30

In the program example shown below, the changed content settings of the extension registers R10 to R19 (sector 0) are reflected on extension file registers (ER) when X000 is set to ON. (128 points are written in one operation cycle.)

Program



Operation

Setting data: Extension registers (R) / Setting backup data: Extension file registers (ER)

	Device number (R)	Current value	Device number (ER)	Current value
Setting data (A)	R0	K100	ER0	K100
	R1	K105	ER1	K105
	:	:	:	:
Changed data	R10	K200	ER10	K300
	R11	K215	ER11	K330
	R12	K400	ER12	K350
	:	:	:	:
	R19	K350	ER19	K400
	:	:	:	:
	R99	K1000	ER99	K1000
Unused	R100	HFFFF	ER100	HFFFF
	:	:	:	:
	R2047	HFFFF	ER2047	HFFFF

(1) ⇒ BMOV instruction

Temporarily withdrawn data*1: Unused extension registers

Device number	Current value
R30720	K100
R30721	K105
:	:
R30730	K200
R30731	K215
R30732	K400
:	:
R30739	K350
:	:
R30819	K1000

(2) ⇒ INITR instruction

*1 Use unused devices as an area to which data are temporarily withdrawn.

To the next page

(2) INITR instruction

Setting data		Setting backup data	
Extension registers (R)		Extension file registers (ER)	
Device number	Current value	Device number	Current value
R0	HFFFF	ER0	HFFFF
R1	HFFFF	ER1	HFFFF
:	:	:	:
R10	HFFFF	ER10	HFFFF
R11	HFFFF	ER11	HFFFF
R12	HFFFF	ER12	HFFFF
:	:	:	:
R19	HFFFF	ER19	HFFFF
:	:	:	:
R99	HFFFF	ER99	HFFFF
R100	HFFFF	ER100	HFFFF
:	:	:	:
R2047	HFFFF	ER2047	HFFFF

(3) BMOV instruction

Setting data

Unused extension registers		
	Device number	Current value
Setting data (A)	R0	K100
	R1	K105
	:	:
	R10	K200
	R11	K215
	R12	K400
	:	:
	R19	K350
	:	:
	R99	K1000
	R100	HFFFF
	:	:
	R2047	HFFFF

(4) SAVER instruction

Setting data

		Extension registers (R)		
		Device number	Current value	
Setting data		R0	K100	1st operation cycle
		R1	K105	
		:	:	
	Changed data	R10	K200	
		R11	K215	
		R12	K400	
		:	:	
		R19	K350	
		:	:	
		R99	K1000	
Unused		R100	HFFFF	
		:	:	
		R127	HFFFF	
		R128	HFFFF	2nd operation cycle
		:	:	
		R255	HFFFF	
		R256	HFFFF	3rd to 15th operation cycle
		:	:	
		R1919	HFFFF	
		R1920	HFFFF	16th operation cycle
		:	:	
		R2047	HFFFF	

Setting backup data

Extension file registers (ER)		Number of already written points (D0)
Device number	Current value	
ER0	K100	
ER1	K105	
:	:	
ER10	K200	
ER11	K215	
ER12	K400	
:	:	
ER19	K350	
:	:	
ER1000	K1000	
ER100	HFFFF	
:	:	K128
ER127	HFFFF	
ER128	HFFFF	
:	:	
ER255	HFFFF	K256
ER256	HFFFF	:
:	:	:
ER1919	HFFFF	K1920
ER1920	HFFFF	
:	:	
ER2047	HFFFF	K2048

33.3 FNC292 – INITR / Initialize R and ER

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction initializes (to “HFFFF” <K-1>) extension registers (R) in the RAM built in a PLC and extension file registers in a memory cassette (flash memory) before data logging by LOGR (FNC293) instruction.
In FX3UC PLCs former than Ver.1.30, use this instruction to initialize extension file registers (ER) before writing data to them using SAVER (FNC291) instruction.
In FX3UC PLCs Ver.1.30 or later and FX3U PLCs, INITER (FNC295) instruction is also provided to initialize (to "HFFFF" <K-1>) only extension file registers (ER) in a memory cassette (flash memory) in units of sector.

→ For SAVER (FNC291) instruction, refer to Section 33.2.
→ For LOGR (FNC293) instruction, refer to Section 33.4.
→ For INITER (FNC295) instruction, refer to Section 33.6.

1. Instruction format

FNC 292 INITR P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	INITR	Continuous Operation
	INITRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
S·	Device number of extension register and extension file register*1 to be initialized It is possible to specify only the head device number in a sector of extension registers.	16-bit binary
n	Number of sectors of extension registers and extension file registers to be initialized	16-bit binary

*1. When a memory cassette is not used, extension file registers (ER) are not initialized.

3. Applicable devices

Operand Type	Bit Devices							Word Devices													Others					
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·															✓				✓					
n																				✓	✓			

Explanation of function and operation

1. 16-bit operation (INITR and INITRP)

“n” sectors of extension registers in the PLC’s built-in RAM starting from the one specified by S· and “n” sectors of extension file registers in a memory cassette (flash memory) having the same device numbers are initialized (to “HFFFF” <K-1>).
Initialization is executed in units of sector.

Command input —| |— [FNC292 INITRP | S· | n]

The table below shows the head device number in each sector:

Sector number	Head device number	Initialized device range
Sector 0	R0	R0 to R2047, ER0 to ER2047
Sector 1	R2048	R2048 to R4095, ER2048 to ER4095
Sector 2	R4096	R4096 to R6143, ER4096 to ER6143
Sector 3	R6144	R6144 to R8191, ER6144 to ER8191
Sector 4	R8192	R8192 to R10239, ER8192 to ER10239
Sector 5	R10240	R10240 to R12287, ER10240 to ER12287
Sector 6	R12288	R12288 to R14335, ER12288 to ER14335
Sector 7	R14336	R14336 to R16383, ER14336 to ER16383

Sector number	Head device number	Initialized device range
Sector 8	R16384	R16384 to R18431, ER16384 to ER18431
Sector 9	R18432	R18432 to R20479, ER18432 to ER20479
Sector 10	R20480	R20480 to R22527, ER20480 to ER22527
Sector 11	R22528	R22528 to R24575, ER22528 to ER24575
Sector 12	R24576	R24576 to R26623, ER24576 to ER26623
Sector 13	R26624	R26624 to R28671, ER26624 to ER28671
Sector 14	R28672	R28672 to R30719, ER28672 to ER30719
Sector 15	R30720	R30720 to R32767, ER30720 to ER32767

Operation (when a memory cassette is used)

- Extension registers (R)
  [inside the built-in RAM memory]

Device number	Current value	
	Before execution	After execution
S•	H0010	HFFFF
S• +1	H0020	HFFFF
S• +2	H0011	HFFFF
⋮	⋮	⋮
S• +(2048×n)-1	HABCD	HFFFF

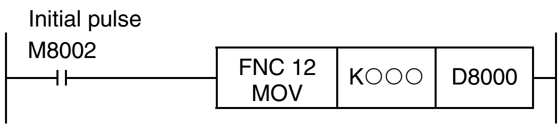
- Extension file registers (ER)
  [inside the memory cassette]

Device number	Current value	
	Before execution	After execution
S•	H1234	HFFFF
S• +1	H5678	HFFFF
S• +2	H90AB	HFFFF
⋮	⋮	⋮
S• +(2048×n)-1	HCDEF	HFFFF

Caution

When a memory cassette is attached, 18 ms is required to initialize one sector.
(When a memory cassette is not attached, only 1 ms or less is required to initialize one sector.)
When initializing two or more sectors, take either measures shown below.

- Set a large value to the watchdog timer D8000 using the following program

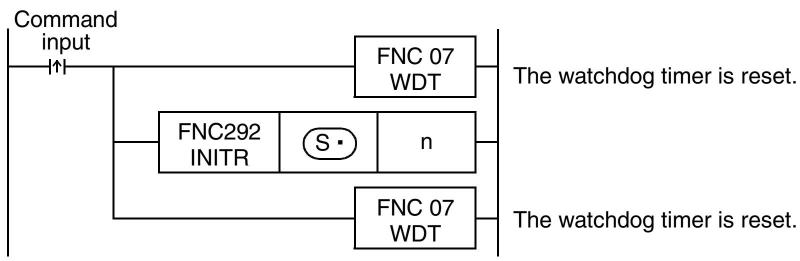


Guideline of the watchdog timer set value

A value acquired by the following procedure can be regarded as the guideline of the watchdog timer set value.
If an acquired value is 200 ms or less, however, it is not necessary to change the watchdog timer set value.

1) Write a program to be executed from GX Developer to the PLC.
[Online]→[Write to PLC...]
2) Set the current value of D8000 (unit: ms) to "1000" using the device test function in GX Developer.
[Online]→[Debug]→[Device test...]→ "Word device/buffer memory" in Device test dialogbox
3) Set the PLC mode to RUN, and execute the program. (Execute this instruction also.)
4) Monitor the maximum scan time D8012 (unit: 0.1ms) using the device batch monitoring function in GX Developer.
5) The watchdog timer (ms) should be set more than the maximum scan time value (D8012).
Note that the maximum scan time value in D8012 is stored in units of 0.1ms.
The set value D8000 (unit: ms) of the watchdog timer should be added 50 to 100 to one-tenth of D8012.

- Setting WDT (FNC 07) instruction just before and after INITR instruction as shown below:



If the processing time of the INITR command exceeds 200ms, set the watchdog timer value D8000 (unit: ms) to the proceessing time or more.

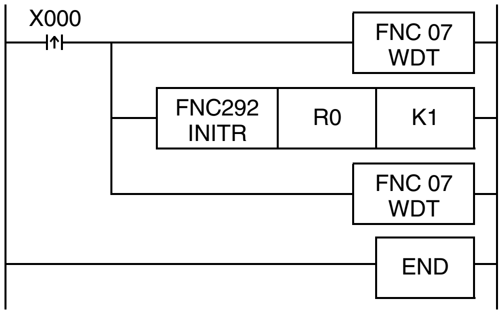
Errors

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When any device number other than the head device number of a sector of extension file registers is set to S· (error code: K6706)
- When a device number to be initialized exceeds "32767" (error code: K6706)
In this case, devices up to R32767 (ER32767) are initialized.
- When the protect switch of the memory cassette is set to ON (error code: K6770)

Program example

In the program example shown below, the extension registers R0 to R2047 in the sector 0 are initialized. Note that the extension file registers ER0 to ER2047 are also initialized if a memory cassette is attached.



- Extension registers (R) [inside the built-in RAM memory]

Device number	Current value	
	Before execution	After execution
R0	H1234	HFFFF
R1	H5678	HFFFF
R2	H90AB	HFFFF
⋮	⋮	⋮
R2047	HCDEF	HFFFF

33.4 FNC293 – LOGR / Logging R and ER

Outline

FX3UC Ver.1.00 → FX3U Ver.2.20 →

This instruction logs specified devices, and stores the logged data to extension registers in the RAM and extension file registers in a memory cassette.

1. Instruction format

FNC 293 LOGR	P

16-bit Instruction	Mnemonic	Operation Condition
11 steps	LOGR	Continuous Operation
	LOGRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
S•	Head device number to be logged*1	16-bit binary
m	Number of devices to be logged [1 ≤ m ≤ 8000]	
D1	Head device number used in logging	
n	Number of sectors of devices used in logging [1 ≤ n ≤ 16]	
D2•	Number of logged data	

*1. C200 to C255 cannot be set.

3. Applicable devices

Operand Type	Bit Devices: System User							Word Devices: Digit Specification				Word Devices: System User				Special Unit	Index			Others: Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S•												✓	✓	✓					✓					
m														✓						✓	✓			
D1															✓									
n																				✓	✓			
D2•														✓					✓					

Explanation of function and operation

1. 16-bit operation (LOGR and LOGRP)

While the instruction is driven, "m" devices starting from S• are logged until "n" sectors of extension registers (R) starting from D1 and extension file registers (ER) in a memory cassette are filled.

The number of logged data is stored to D2•.

If a memory cassette is not used, data is not written to extension file registers (ER).

Command input

FNC293 LOGRP	S•	m	D1	n	D2•

Logging data format

Device	Contents	Number of stored data	Area
D1 to D1 +m-1	Destination storing 1st logging data S• to S• +m-1	→ D1 = m	Data writing area 1926 × n (within Logging data storage area 2048 × n)
D1 +m to D1 +2m-1	Destination storing 2nd logging data S• to S• +m-1	→ D1 = 2m	
D1 +2m to D1 +3m-1	Destination storing 3rd logging data	→ D1 = 3m	
⋮	⋮	⋮	
D1 +(1926 × n)-1 ⋮ D1 +(2048 × n)-1	Writing position control area (Every time one word is used in the data writing area, bits are set to OFF (0) from ON (1) in turn from bit 0 of [D1 + (1926 × n) -1]. After bit 15 of [D1 + (1926 × n) -1] is set to OFF (0), bit 0 of the next device [D1 + (1926 × n)] is set to OFF (0) in the next logging.		Data writing position control area 122 × n

Logging data storage area 2048 × n

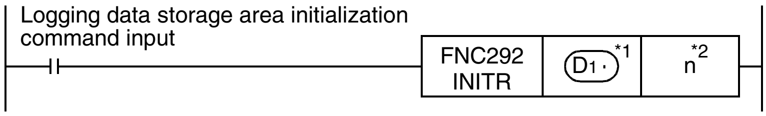
The table below shows the head device number in each sector:

Sector number	Head device number	Written device range
Sector 0	R0	R0 to R2047, ER0 to ER2047
Sector 1	R2048	R2048 to R4095, ER2048 to ER4095
Sector 2	R4096	R4096 to R6143, ER4096 to ER6143
Sector 3	R6144	R6144 to R8191, ER6144 to ER8191
Sector 4	R8192	R8192 to R10239, ER8192 to ER10239
Sector 5	R10240	R10240 to R12287, ER10240 to ER12287
Sector 6	R12288	R12288 to R14335, ER12288 to ER14335
Sector 7	R14336	R14336 to R16383, ER14336 to ER16383

Sector number	Head device number	Written device range
Sector 8	R16384	R16384 to R18431, ER16384 to ER18431
Sector 9	R18432	R18432 to R20479, ER18432 to ER20479
Sector 10	R20480	R20480 to R22527, ER20480 to ER22527
Sector 11	R22528	R22528 to R24575, ER22528 to ER24575
Sector 12	R24576	R24576 to R26623, ER24576 to ER26623
Sector 13	R26624	R26624 to R28671, ER26624 to ER28671
Sector 14	R28672	R28672 to R30719, ER28672 to ER30719
Sector 15	R30720	R30720 to R32767, ER30720 to ER32767

Cautions

- LOGR instruction executes logging in each operation in the continuous operation type.
  When logging should be executed only once by one input, use the pulse operation type.
- Caution on using a memory cassette
  Flash memory is adopted in a memory cassette. Make sure to initialize the data storage area in units of sector before starting logging.
  If LOGR instruction is executed without initializing the data storage area, an operation error (error code: K6770) may be caused.



*1 Specify the same device as (D1 ·) in LOGR instruction.
*2 Specify the same number as (n) in LOGR instruction.

Errors

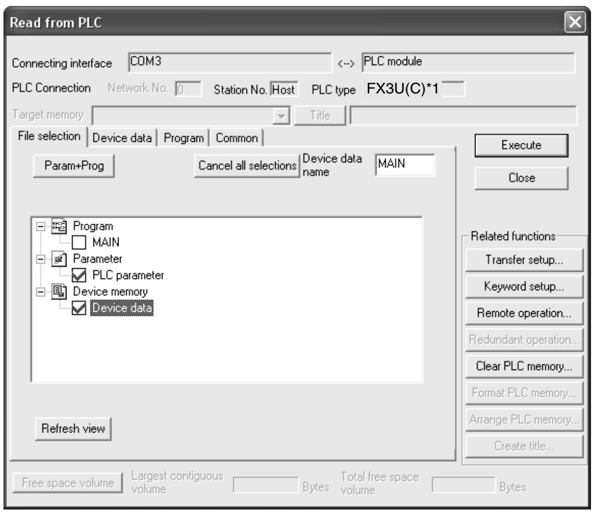
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When any device number other than the head device number of a sector of extension file registers is set to (S ·) (error code: K6706)
- While data is written, the remaining area and the data quantity to be written are compared with each other. If the remaining storage area is insufficient, only a limited amount of data is written. (error code: K6706)
- When the protect switch of the memory cassette is set to ON (error code: K6770)
- When the collation result after data writing is "mismatch" due to omission of initialization or for another reason (error code: K6770)
  When this error occurs, the current values (data) of extension registers (R) may be lost. To avoid the data loss, back up the data of extension registers (R) in advance using the following procedure:

  1)Set the PLC mode to STOP.

  2)Create a new project in GX Developer.
  This step is not necessary if it is alright to overwrite the current project.

  3)Read the contents of extension registers (R) to GX Developer.
  [1]Select "Online" → "Read from PLC..." to display "Read from PLC" window.
  [2]Click "Parameter" and "Device data" to put a check mark next to each of them.
  [3]Click [Execute] button to execute reading.
  [4]When reading is completed, save the project.



*1.For Ver. 8.13P to 8.24A of GX Developer, the PLC type is FX3UC.

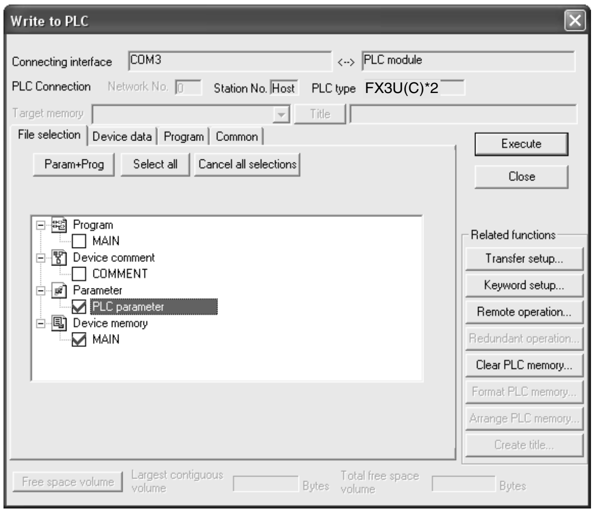
4)Change the current program inside the PLC to the program shown in “1. Cautions on writing data to a memory cassette” in “Cautions” on the previous page.

5)To the PLC, write the data which was temporarily withdrawn to GX Developer.

[1]Select “Online” → “Write to PLC...” to display the “Write to PLC” window.

[2]Click “Parameter” and “MAIN” to put a check mark to each of them.

[3]Click [Execute] button to execute writing.

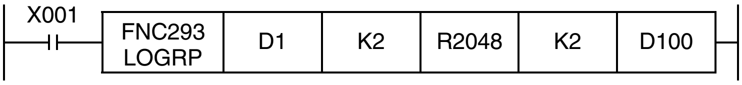


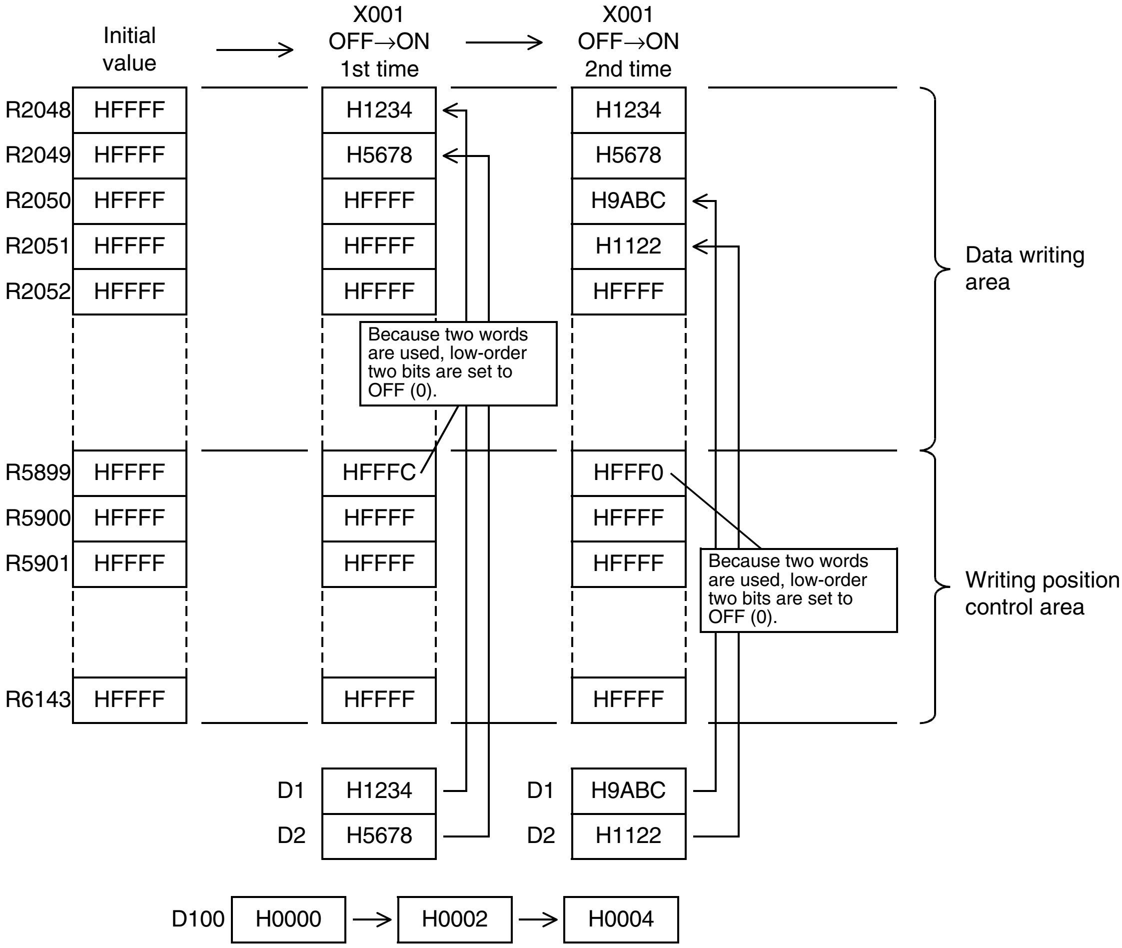
*2.For Ver. 8.13P to 8.24A of GX Developer, the PLC type is FX3UC.

6)Change the PLC mode from STOP to RUN, execute the program, and store the data to the extension file registers inside the memory cassette.

Program example

In the program example shown below, D1 and D2 are logged to the area from R2048 to R4095 every time X001 turns ON.





31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC•STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List

33.5 FNC294 – RWER / Rewrite to ER

Outline

FX3UC Ver.1.30 → FX3U Ver.2.20 →

This instruction writes the current values of an arbitrary number of extension registers (R) in the PLC's built-in RAM to extension file registers (ER) in a memory cassette (flash memory).
Because RWER (FNC294) instruction is not supported in FX3UC PLCs former than Ver.1.30, use SAVER (FNC291) instruction instead.

→ **For SAVER (FNC291) instruction, refer to Section 33.2.**

1. Instruction format

FNC 294 RWER P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	RWER	Continuous Operation
	RWERP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
S·	Device number of extension register storing data	16-bit binary
n	Number of written (transferred) devices [0 ≤ n ≤ 32767]	

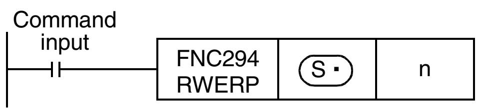
3. Applicable devices

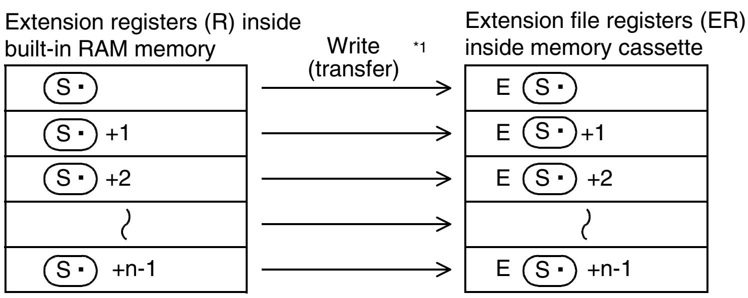
Operand Type	Bit Devices							Word Devices												Others				
	System User							Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
S·															✓				✓					
n														✓						✓	✓			

Explanation of function and operation

1. 16-bit operation (RWER)

The contents (current values) of "n" extension registers (R) starting from S· are written (transferred) to extension file registers having the same device numbers in a memory cassette (flash memory).





*1 All points specified by the instruction are written (transferred).

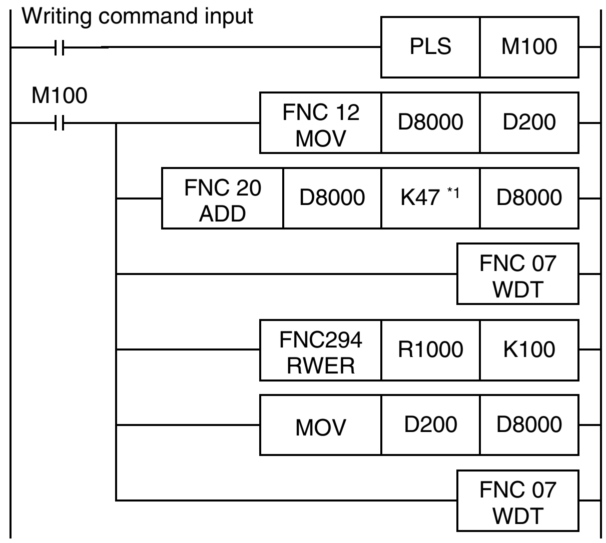
- When "n" is set to "0", it is handled as "32768" when the instruction is executed.

Cautions

1. Cautions on writing data to a memory cassette

Memory cassettes adopt flash memory. Note the following contents when writing data to extension file registers in a memory cassette with the FNC294 instruction.

- Though extension file registers to be written can be specified arbitrarily, writing is executed in units of sector.
  It takes about 47 ms to write one sector. If the extension file registers to be written are located in two sectors, the instruction execution time will be about 94 ms.
  Make sure to change the set value of the watchdog timer D8000 before executing this instruction.



*1 Number of sectors to be written × 47 ms

The contents of R1000 to R1099 in the sector 0 are written to ER1000 to ER1099.

The table below shows the head device number in each sector:

Sector number	Device range	Sector number	Device range
Sector 0	ER0 to ER2047	Sector 8	ER16384 to ER18431
Sector 1	ER2048 to ER4095	Sector 9	ER18432 to ER20479
Sector 2	ER4096 to ER6143	Sector 10	ER20480 to ER22527
Sector 3	ER6144 to ER8191	Sector 11	ER22528 to ER24575
Sector 4	ER8192 to ER10239	Sector 12	ER24576 to ER26623
Sector 5	ER10240 to ER12287	Sector 13	ER26624 to ER28671
Sector 6	ER12288 to ER14335	Sector 14	ER28672 to ER30719
Sector 7	ER14336 to ER16383	Sector 15	ER30720 to ER32767

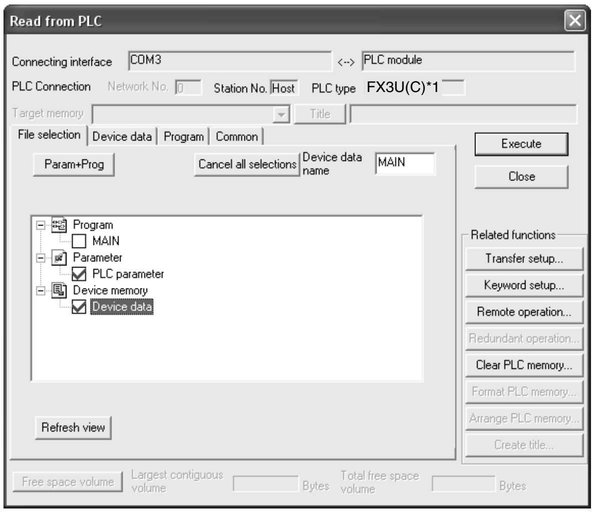
- Do not turn OFF the power while this instruction is being executed. If the power is turned OFF, execution of this instruction may be aborted. If execution is aborted, the data may be lost. Make sure to back up the data before executing this instruction.

→ **For the backup method, refer to the next page.**

Data backup method

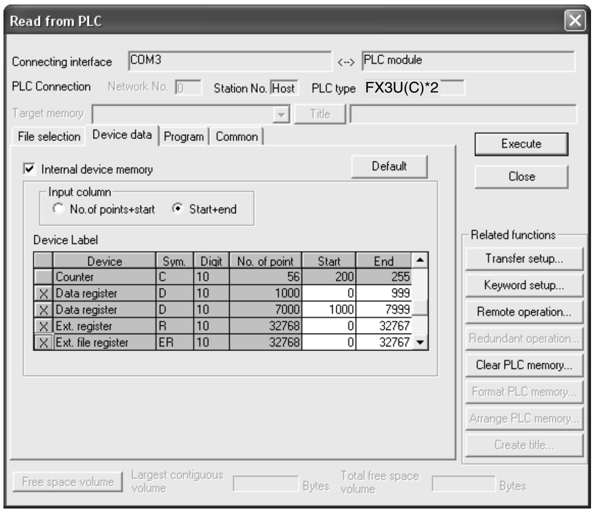
When the contents of extension file registers (ER) and extension registers (R) should not be lost, back up the current values (data) of extension file registers (ER) and extension registers (R) in advance using the following procedure:

1) Set the PLC mode to STOP.
2) Create a new project in GX Developer.
   This step is not necessary if it is alright to overwrite the current project.
3) Read the contents of extension file registers (ER) and extension registers (R) to GX Developer.
   [1]Select “Online” → “Read from PLC...” to display the “Read from PLC” window.
   [2]Click “Parameter” and “Device data” to put a check mark next to each of them.



*1.For Ver. 8.13P to 8.24A of GX Developer, the PLC type is FX3UC.

   [3]Select “Ext. file register” and “Ext. register” on the “Device data” tab. In GX Developer former than Ver.8.18U, the extension file register range cannot be set.
   [4]Click [Execute] button to execute reading.
   [5]When reading is complete, save the project.



*2.For Ver. 8.13P to 8.24A of GX Developer, the PLC type is FX3UC.

Errors

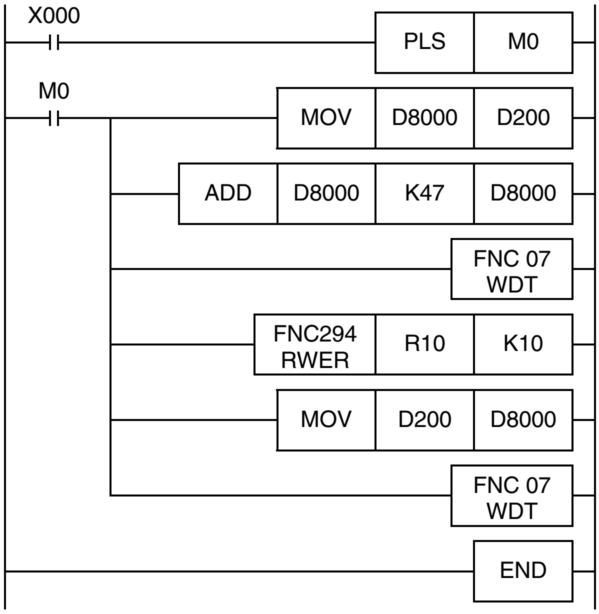
An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When the last device number to be transferred exceeds “32767” (error code: K6706)
- When a memory cassette is not connected (error code: K6771)
- When the protect switch of the memory cassette is set to ON (error code: K6770)

Program example

In the program example shown below, the contents of extension registers R10 to R19 (sector 0) used for setting data are reflected on extension file registers (ER) when X000 turns ON.

Program



Operation

Setting data

Extension registers (R)	
Device number	Current value
R0	K100
R1	K105
:	:
R10	K200
R11	K215
R12	K400
:	:
R19	K350
:	:
R99	K1000
R100	HFFFF
:	:
R2047	HFFFF

Setting data (A): R0 to R99
Changed data: R10 to R19
Unused: R100 to R2047

Write
(when X000
turns ON)

Setting backup data

Extension file registers (ER)	
Device number	Current value
ER0	K100
ER1	K105
:	:
ER10	K200
ER11	K215
ER12	K400
:	:
ER19	K350
:	:
ER99	K1000
ER100	HFFFF
:	:
ER2047	HFFFF

33.6 FNC295 – INITER / Initialize ER

Outline

FX3UC Ver.1.30 → FX3U Ver.2.20 →

This instruction initializes extension file registers (ER) to "HFFFF" (<K-1>) in a memory cassette (flash memory) before executing SAVER (FNC291) instruction.
Because INITER (FNC295) instruction is not supported in FX3UC PLCs former than Ver.1.30, use INITR (FNC292) instruction instead.

→ **For SAVER (FNC291) instruction, refer to Section 33.2.**
→ **For INITR (FNC292) instruction, refer to Section 33.3.**

1. Instruction format

FNC 295 INITER P

16-bit Instruction	Mnemonic	Operation Condition
5 steps	INITR	Continuous Operation
	INITRP	Pulse (Single) Operation

32-bit Instruction	Mnemonic	Operation Condition
	—	
	—	

2. Set data

Operand Type	Description	Data Type
(S•)	Head device number of extension register sector with the same device number of extension file register to be initialized	16-bit binary
n	Number of sectors of extension registers and extension file registers to be initialized	

3. Applicable devices

Oper-and Type	Bit Devices							Word Devices													Others				
	System User							Digit Specification				System User				Special Unit	Index			Con-stant		Real Number	Charac-ter String	Pointer
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P
(S•)															✓				✓					
n																				✓	✓			

Explanation of function and operation

1. 16-bit operation (INITER and INITERP)

"n" sectors of extension file registers in a memory cassette (flash memory) with the same device number as (S•) are initialized to "HFFFF" (<K-1>).
Initialization is executed in units of sector.

Command input ─┤├─ | FNC295 INITERP | (S•) | n |

The table below shows the head device number in each sector:

Sector number	Head device number	Initialized device range
Sector 0	R0	ER0 to ER2047
Sector 1	R2048	ER2048 to ER4095
Sector 2	R4096	ER4096 to ER6143
Sector 3	R6144	ER6144 to ER8191
Sector 4	R8192	ER8192 to ER10239
Sector 5	R10240	ER10240 to ER12287
Sector 6	R12288	ER12288 to ER14335
Sector 7	R14336	ER14336 to ER16383
Sector 8	R16384	ER16384 to ER18431
Sector 9	R18432	ER18432 to ER20479
Sector 10	R20480	ER20480 to ER22527
Sector 11	R22528	ER22528 to ER24575
Sector 12	R24576	ER24576 to ER26623
Sector 13	R26624	ER26624 to ER28671
Sector 14	R28672	ER28672 to ER30719
Sector 15	R30720	ER30720 to ER32767

Operation

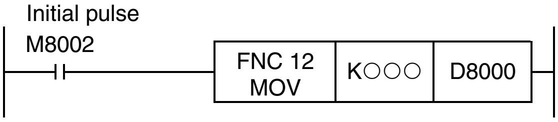
- Extension file registers (ER) [inside the memory cassette]

Device number	Current value	
	Before execution	**After execution**
S•	H1234	HFFFF
S• +1	H5678	HFFFF
S• +2	H90AB	HFFFF
⋮	⋮	⋮
S• +(2048×n)-1	HCDEF	HFFFF

Caution

About 25 ms is required to initialize one sector.
When initializing two or more sectors, take either measure shown below.

- Set a large value to the watchdog timer D8000 using the following program

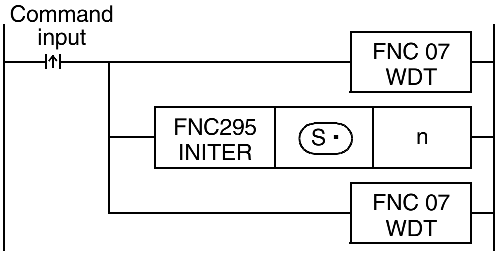


Guideline of the watchdog timer set value

A value acquired by the following procedure can be regarded as the guideline of the watchdog timer set value.
If an acquired value is 200 ms or less, however, it is not necessary to change the watchdog timer set value.

1) Write a program to be executed from GX Developer to the PLC.
   [Online]→[Write to PLC...]
2) Set the current value of D8000 (unit: ms) to "1000" using the device test function in GX Developer.
   [Online]→[Debug]→[Device test...]→ "Word device/buffer memory" in Device test dialogbox
3) Set the PLC mode to RUN, and execute the program. (Execute this instruction also.)
4) Monitor the maximum scan time D8012 (unit: 0.1ms) using the device batch monitoring function in GX Developer.
5) The watchdog timer (ms) should be set more than the maximum scan time value (D8012).
   Note that the maximum scan time value in D8012 is stored in units of 0.1ms.
   The set value D8000 (unit: ms) of the watchdog timer should be added 50 to100 to one-tenth of D8012.

- Setting WDT (FNC 07) instruction just before and after INITER instruction as shown below:



If the processing time of the INITR command exceeds 200ms, set the watchdog timer value D8000 (unit: ms) to the processing time or more.

31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC•STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List

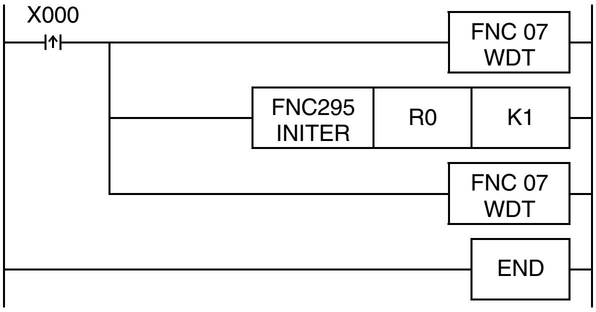
Errors

An operation error is caused in the following cases; The error flag M8067 turns ON, and the error code is stored in D8067.

- When any device number other than the head device number of a sector of extension file registers (ER) is set to S· (error code: K6706)
- When a device number to be initialized exceeds “32767” (error code: K6706)
  In this case, devices up to R32767 (ER32767) are initialized.
- When the protect switch of the memory cassette is set to ON (error code: K6770)
- When a memory cassette is not connected (error code: K6771)

Program example

In the program example shown below, the extension file registers ER0 to ER2047 in sector 0 are initialized.



- Extension file registers (ER) [inside the memory cassette]

Device number	Current value	
	Before execution	After execution
ER0	H1234	HFFFF
ER1	H5678	HFFFF
ER2	H90AB	HFFFF
⋮	⋮	⋮
ER2047	HCDEF	HFFFF

34. SFC Program and Step Ladder

This chapter explains the programming procedures and sequence operations for the “SFC” and “step ladder” programming methods in GX Developer.

34.1 SFC Program

34.1.1 Outline

Sequence control using the SFC (sequential function chart) is available in FX PLCs.
In SFC programs, the role of each process and the overall control flow can be expressed easily based on machine operations, so sequence design is easy. Accordingly, machine operations can be easily transmitted to any person, and created programs are efficient in maintenance, specifications changes and actions against problems.
When SFC programs and step ladder instructions are programmed conforming to the same rules, they are compatible with each other.
As a result, the same contents can be handled in relay ladder charts which are familiar and easy to understand.

34.1.2 Explanation of function and operation

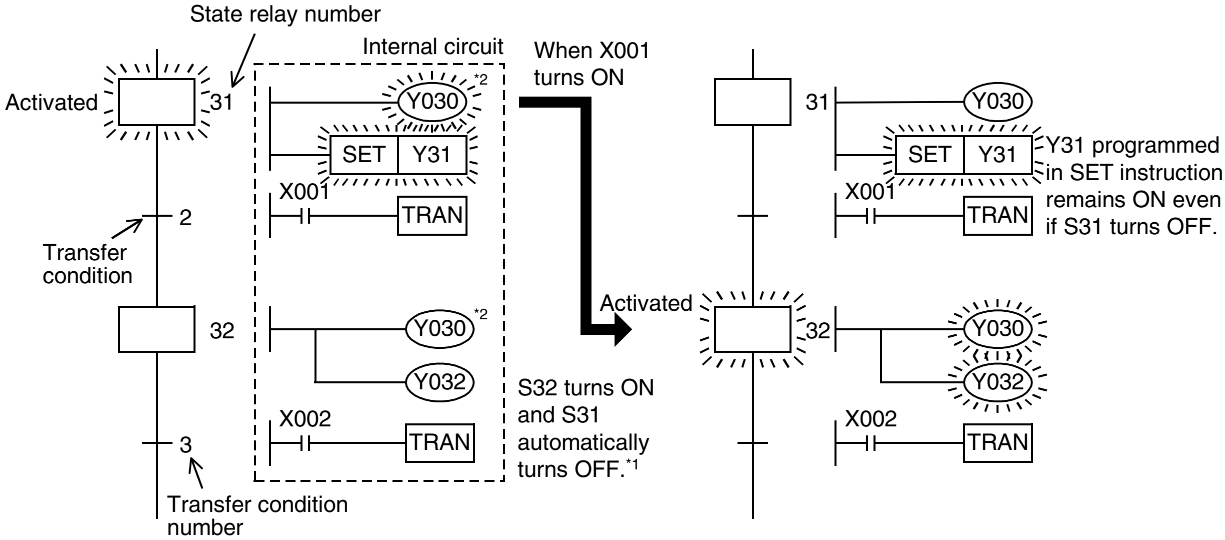
In SFC programs, a state relay State S is regarded as one control process, and the input conditions and output control sequence are programmed in each process.
Because the preceding process is stopped when the program execution proceeds to the next process, a machine can be controlled using simple sequences for each process.

Operation of state relay State S and driven instruction

In SFC programs, each process performed by the machine is expressed by a state relay.

- When a state relay turns ON, a connected circuit (internal circuit) is activated.
  When a state relay turns OFF, a connected internal circuit is deactivated.
  After one operation cycle, non-driving of an instruction (jump status) is not available.
- When a condition (transfer condition) provided between state relays is satisfied, the next state relay turns ON, and the state relay which has been ON so far turns OFF (transfer operation).
  In the state relay ON status transfer process, both state relays are ON only momentarily (for one operation cycle).
  In the next operation cycle after the ON status is transferred to the next state relay, the former state is reset to OFF.
- A state relay number can be used only once.

31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC•STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List



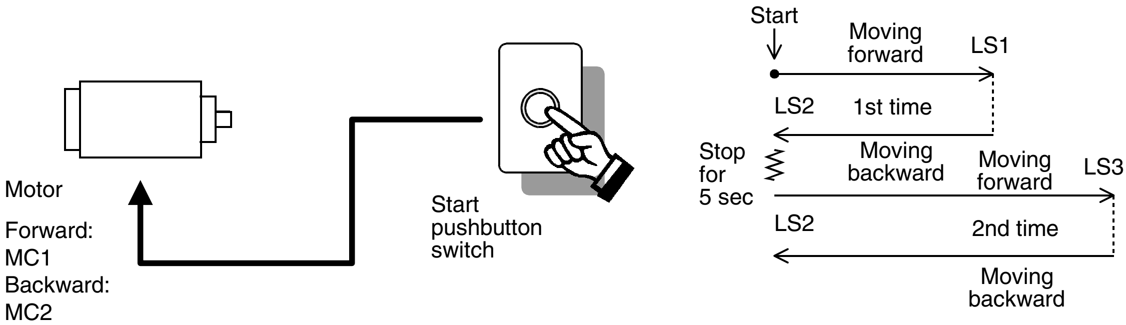
*1. In the next operation cycle after the ON status is transferred from a former state relay to the next state relay, the former state relay turns OFF.

*2. An output coil can be used again in different state relays.

34.1.3 SFC program creating procedure

Create an SFC program using the following procedure:

1. Operation example

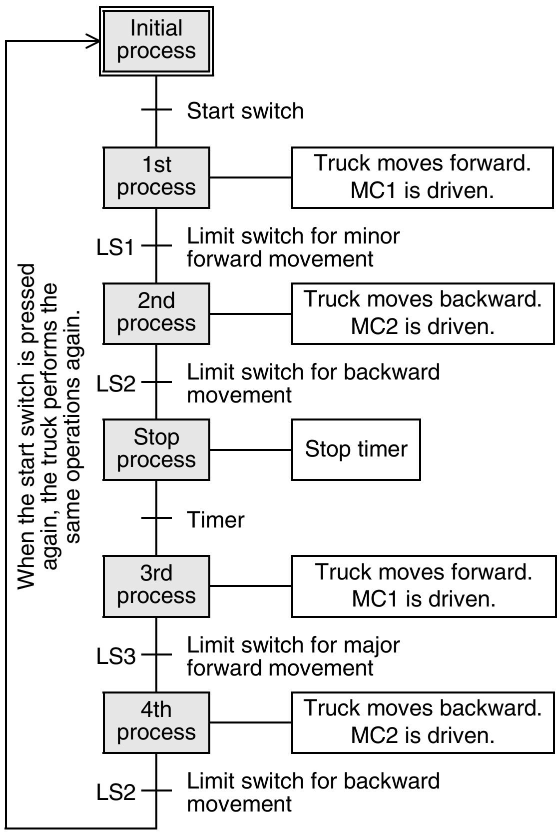


1) When the start pushbutton switch is pressed, the truck moves forward. When the limit switch LS1 turns ON, the truck immediately starts to move backward.
(The limit switch LS1 is normally OFF, and turns ON when the truck reaches the forward limit. Other limit switches function in the same way.)
2) When the truck moves backward and the limit switch LS2 turns ON, the truck stops for 5 seconds, and then starts to move forward again. When the limit switch LS3 turns ON, the truck immediately starts to move backward.
3) When LS2 turns ON after that, the truck driving motor stops.
4) When the start pushbutton switch is pressed again after a series of operations finish, the above operation is repeated.

2. Creating a process drawing

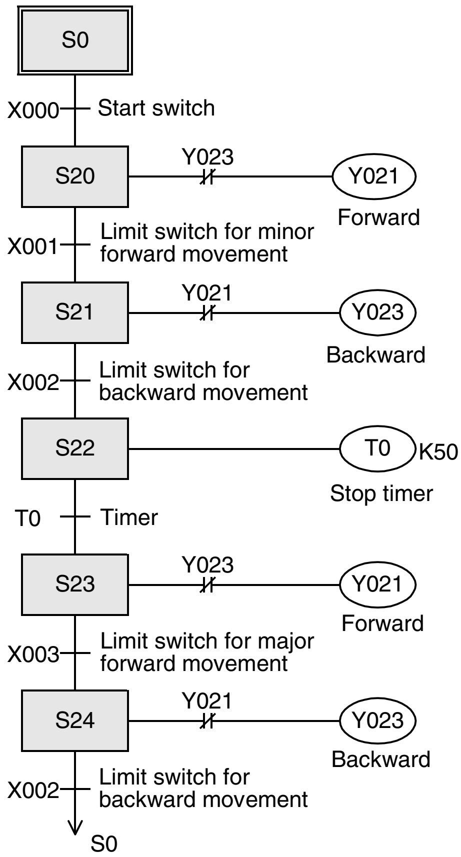
Create the process drawing shown on the right using the following procedure:

1) Divide the operation described in the above example into individual processes, and express each process in a rectangle in the order of operation from top to bottom.
2) Connect each process with vertical lines, and write the condition for each proceeding process. When performing repeated operations, indicate with an arrow the process the truck will return to after a series of operations finish.
3) Write the operation performed in each process on the right side of each rectangle indicating a process.



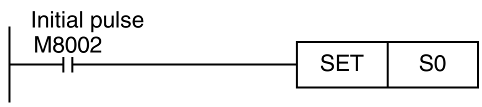
3. Assigning devices

Assign devices of a PLC in the created process drawing.



1) Assign a state relay S· to a rectangle indicating a process.
   At this time, assign a state relay (S0 to S9) to the initial process.
   After the first process, arbitrarily assign state relay numbers (S20 to S899) except the initial state relays.
   (There is no relationship between state relay numbers and order of processes.)
   There are latched (battery backed) type state relays whose ON/OFF status is stored against power failure.
   The state relays S10 to S19 are used for special purposes when the IST (FNC 60) instruction is used.
2) Assign a device (input terminal number connected to a pushbutton switch or limit switch, timer number, etc.) to each transfer condition.
   NO contact and NC contact are available for a transfer condition.
   If there are two or more transfer conditions, AND circuit or OR circuit is available.
3) Assign a device (output terminal number connected to external equipment, timer number, etc.) used for an operation performed in each process.
   Many devices such as timers, counters and auxiliary relays are provided in a PLC, and can be used arbitrarily.
   The timer T0 is used here. Because T0 works by the 0.1 sec clock, the output contact turns ON five seconds after a coil is driven when the set value is K50.
   If there are two or more loads such as timers and counters which are driven at the same time, two or more circuits can be assigned to one state relay.
4) When performing repeated operations or skipping some processes (jump operation), use " ↳" and specify the jump destination state relay number.

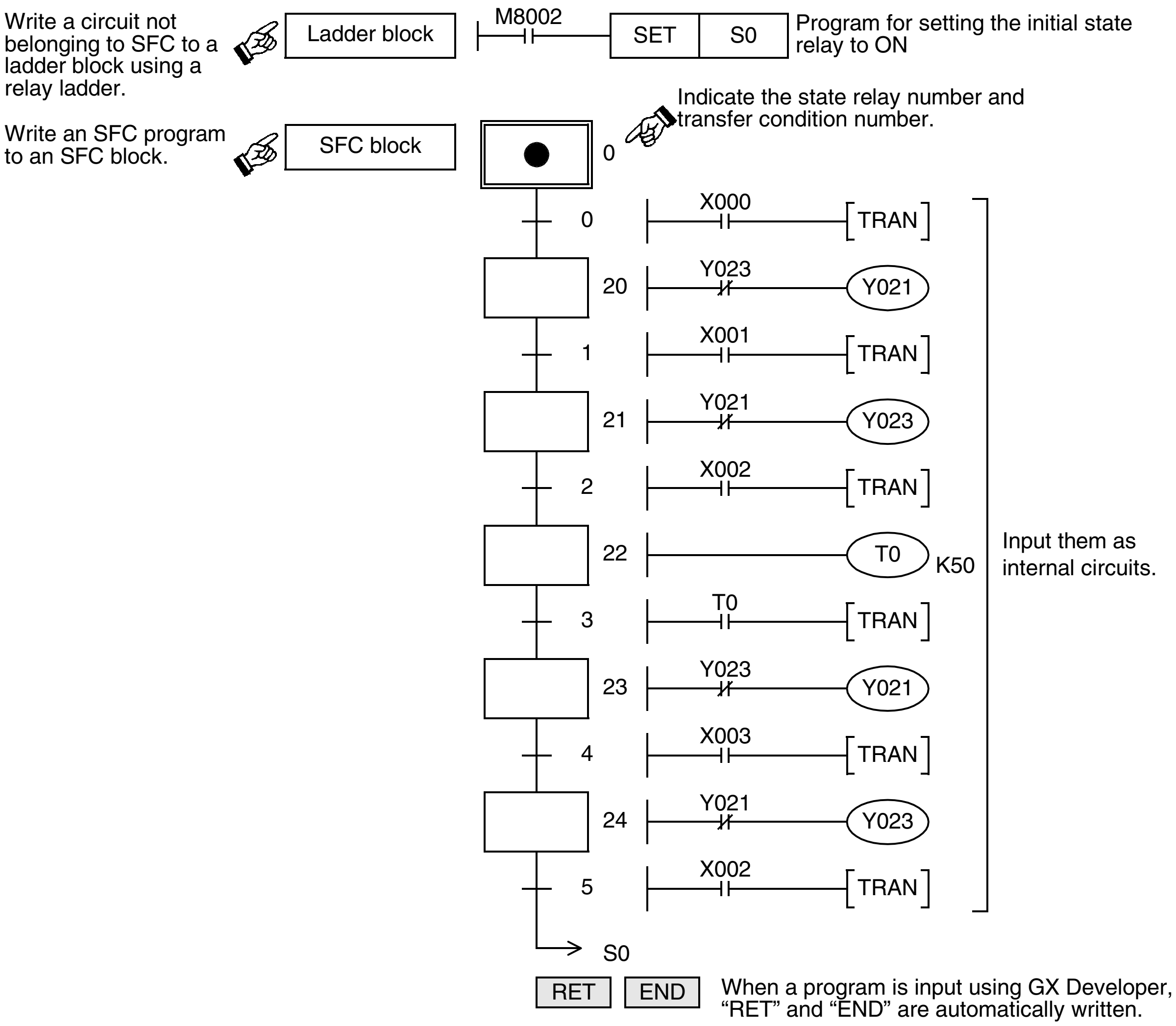
In this example, only the SFC program creating procedure is explained. In practical cases, a circuit for setting the initial state relay to ON is required to execute the SFC program.
Create a circuit for setting the initial state relay to ON using the relay ladder.
At this time, use SET instruction to set the initial state relay to ON.



4. Inputting and indicating a program using GX Developer

- Input a circuit for setting the initial state relay to ON using the relay ladder.
  In this example, the initial state relay S0 is set to ON in a ladder block using the special auxiliary relay M8002 which turns ON momentarily when the PLC mode is changed from STOP to RUN.
- When inputting a program using GX Developer, write a relay ladder program to a ladder block, and write an SFC program to an SFC block.
- Programs expressing operations in state relays and transfer conditions are handled as internal circuits of the state relays and transfer conditions.
  Create each one using a relay ladder.

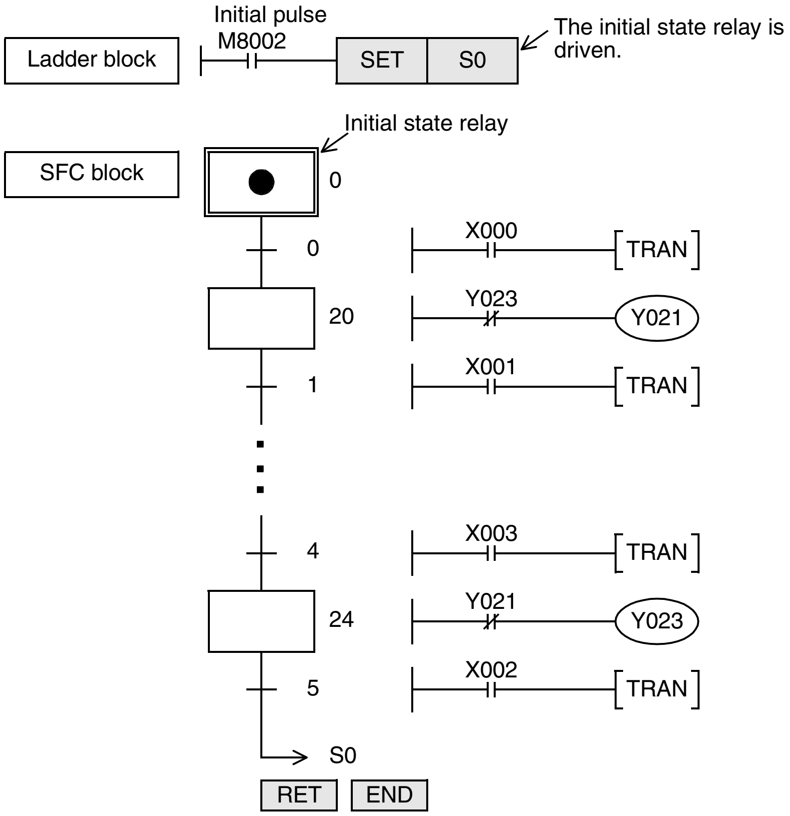
For details of programming procedure in GX Developer, refer to GX Developer Operating Manual.



34.1.4 Handling and role of initial state relay

Handling of the initial state relay

- A state relay located at the head of an SFC program is the initial state relay. Only state relays S0 to S9 are available.
- The initial state relay is driven by way of another state relay (S24 in the example shown below). But it is necessary to drive the initial state relay in advance by another measure at the start of operation.
- In the example shown below, the initial state relay is driven by the special auxiliary relay M8002 which turns ON and remains ON only momentarily when the PLC mode is changed from STOP to RUN.
- General state relays other than initial state relays should be driven by another state relay. They cannot be driven by any other device.
- The state relay which may be driven by a contact other than the STL instruction is there by defined as the initial state relay, and should be described at the top of the flow.



Role of the initial state relay

1. Used as an identification device for inverse conversion

- In inverse conversion from an instruction list into an SFC program, it is necessary to identify the top of the flow. For this purpose, use the initial state relay S0 to S9.
  If any other state relay number is used, inverse conversion is disabled.
- Program the STL instruction for the initial state relay before the STL instructions for subsequment state relays. Program the RET instruction at the end.
  By this programming method, if there are two or more independent flows, they the separated from each other.

2. Used to prevent double start

- In the above example, even if the start button is pressed while the state relay S24 is ON, the command is invalid (S0 does not turn ON).
  As a result, double start is prevented.

34.1.5 Latched (battery backed) type state relays

In the latched (battery backed) type state relays S·, the ON/OFF status is backed up by the battery against power failure.
Use this type of state relays if the operation should be restarted from the last point at power recovery after power failure occurred in the middle of machine operations.

34.1.6 Role of RET instruction

- Use RET instruction at the end of an SFC program.
  When inputting an SFC program using GX Developer, however, it is not necessary to input RET instruction (because RET instruction is automatically written).
- In a PLC, two or more SFC blocks can be put between step 0 and the END instruction.
  When there are ladder blocks and SFC blocks, put RET instruction at the end of each SFC program.

34.1.7 Preliminary knowledge for creating SFC program

List of sequence instructions available in states

State relay		Instruction		
		LD/LDI/LDP/LDF, AND/ANI/ANDP/ANDF, OR/ORI/ORP/ORF, INV,OUT,SET/RST, PLS/PLF	**ANB/ORB/MPS/MRD/ MPP**	**MC/MCR**
Initial/general state relay		Available	Available*1	Not available
Branch/ recombination state relay	Drive processing	Available	Available*1	Not available
	Transfer processing	Available	Not available	Not available

- STL instruction cannot be used in interrupt programs and subroutine programs.
- It is not prohibited to use jump instructions in state relays. However, it is not recommended to use jump instructions because complicated movements will result.

*1. The MPS instruction cannot be used immediately after a state relay (STL instruction), even in a drive processing circuit.

Special auxiliary relays

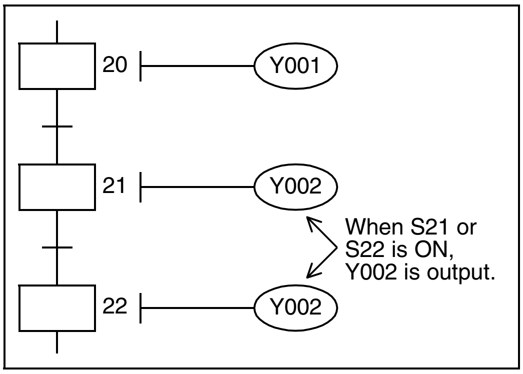
For efficiently creating SFC programs, it is necessary to use some special auxiliary relays. The table below shows major ones.

Device number	Name	Function and application
M8000	RUN monitor	This relay is normally ON while the PLC is in the RUN mode. Use this relay as the program input condition requiring the normally driven status or for indicating the PLC operation status.
M8002	Initial pulse	This relay turns ON and remains ON only instantaneously when the PLC mode is changed from STOP to RUN. Use this relay for the initial setting of a program or for setting the initial state relay.
M8040	STL transfer disable	When this relay is set to ON, transfer to the ON status is disabled among all state relays. Because programs in state relays are operating even in the transfer disabled status, output coils do not turn OFF automatically.
M8046*1	STL state ON	When any state relay turns ON, this relay automatically turns ON. Use this relay to prevent simultaneous start up of another float or as process ON/OFF flag.
M8047*1	Enable STL monitoring	When this relay is driven, the device number of a state relay in the ON status having the smallest device number among S0 to S899 and S1000 to S4095 is stored to D8040, and the state relay number in the ON status having the next smallest device number is stored to D8041. In this way, up to eight state relays in the ON status are stored up to D8047. • In the FX-PCS/WIN(-E), FX-20P(-E), and FX-10P(-E), when this relay is driven, the state relays in the ON status are automatically read and displayed. For details, refer to the manual of each peripheral equipment. • In the SFC monitor in GX Developer, the automatic scroll monitoring function is valid even if this relay is not driven.

*1. Processed when END instruction is executed.

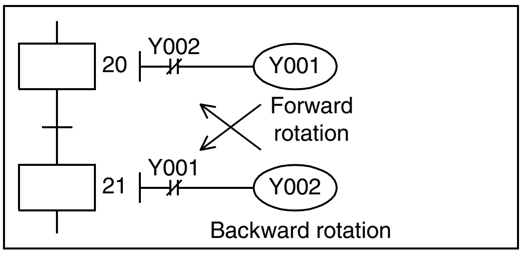
Operation of state relays and use of an output two or more times

- In different state relays, a same output device (Y002 in this example) can be programmed as shown in the right figure.
  In this case, when S21 or S22 is ON, Y002 is output.
  However, if the same device as an output coil (Y002) in a state relay is programmed in a ladder block program or if a same output coil is programmed twice in one state relay, it is handled in the same way as general double coil.



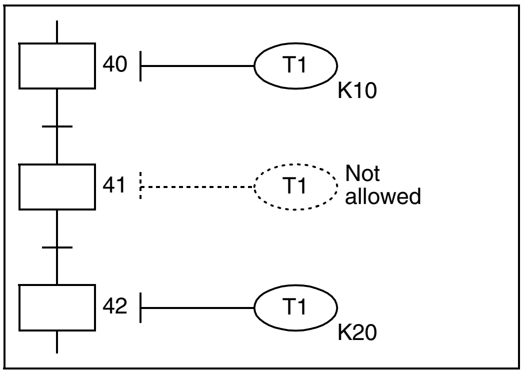
Interlock of outputs

- In the state relay ON status transfer process, both states turn ON only instantaneously (during one operation cycle) at the same time.
  Accordingly, between a pair of outputs which should not be set to ON at the same time, provide an interlock outside the PLC in conformance to the handy manual of the PLC so that simultaneous ON can be prevented.
  In addition, provide interlock in the program as shown in the right figure.



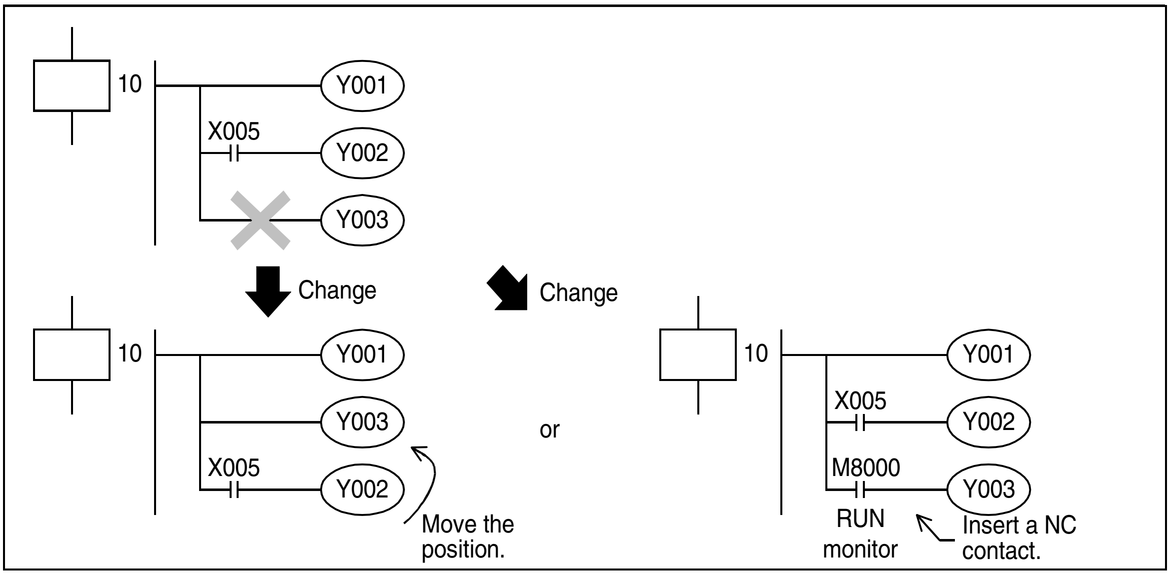
Use of a timer two or more times

- In the same way as an output coil, a timer coil can be programmed in different state relays. However, it is not permitted to program the same timer coil in adjacent state relays. If the same timer coil is programmed in adjacent state relays, the timer coil is not set to OFF at process transfer, so the current value is not reset.



Output driving method

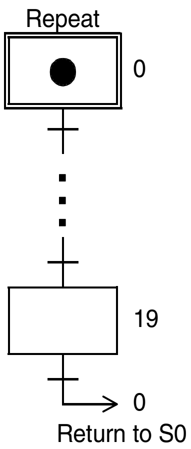
- It is not permitted to write program an instruction not requiring a contact after LD or LDI instruction from a bus line in a state relay.
  Change such a circuit as shown below.

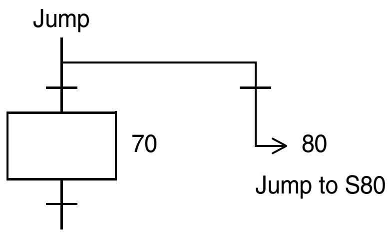


Operations of " ↳" and "▽"

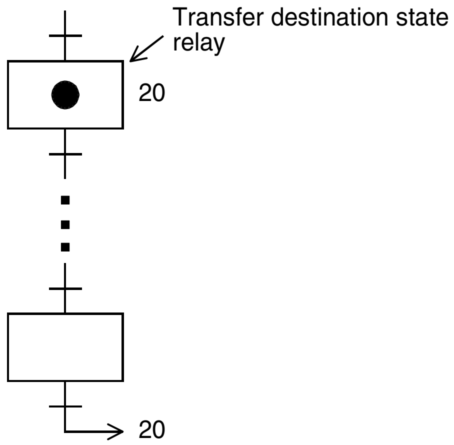
Use " ↳" to express transfer to a state relay in an upper position (repeat), transfer to a state relay in a lower position (jump), or transfer to a state relay in another separate flow.
Use "▽" to express reset of a state relay.

1) Transfer source program

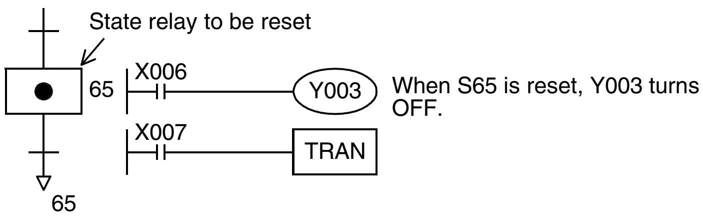




2) Transfer destination program
In GX Developer, "●" is automatically displayed in the transfer destination state relay.



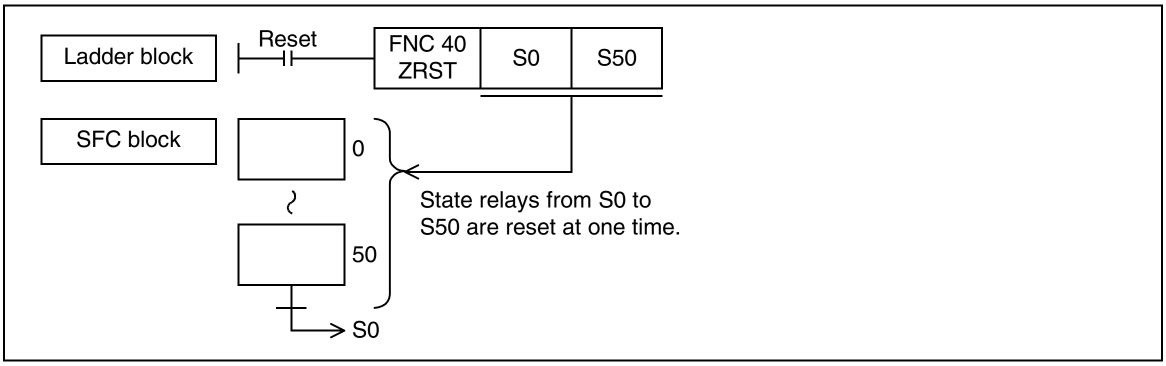
3) Reset circuit program
In the program example shown below, S65 is reset from S65 by way of X007.
Reset of another state relay (S70, for example) from S65 is executed in the same way, but in this case S65 is not reset because this is not transfer.
In GX Developer, "●" is automatically displayed in a state relay to be reset.



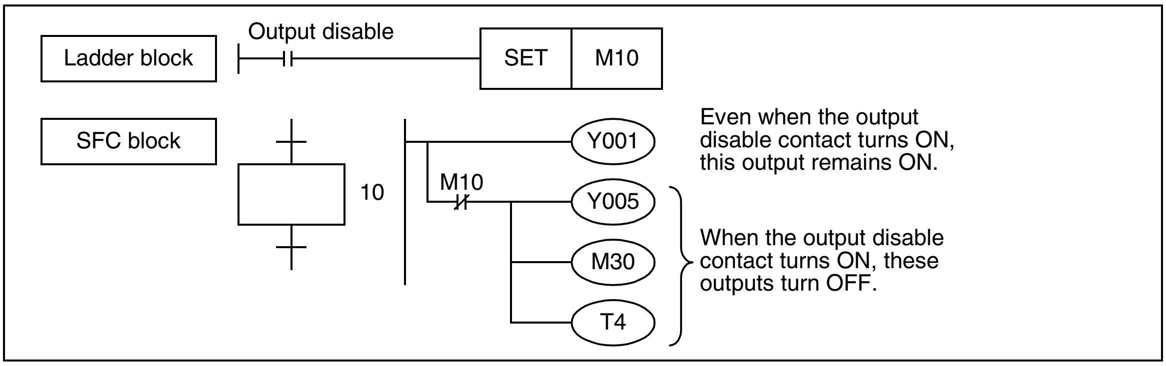
Reset of state relays at one time and output disability

For output disability corresponding to emergency stop, follow “Cautions on safety” described in the PLC manual.

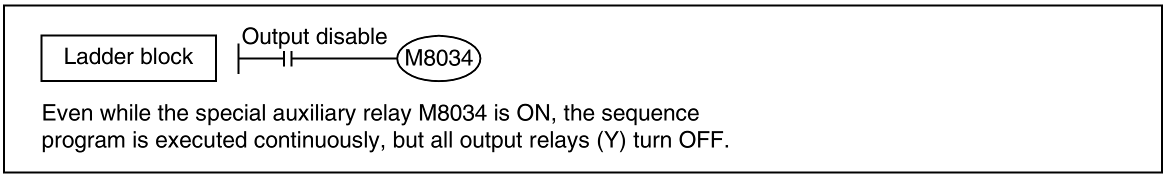
1) Resetting many state relays at one time by specifying a range
Fifty-one state relays from S0 to S50 are reset at one time.



2) Disabling arbitrary output of state relays in the ON status

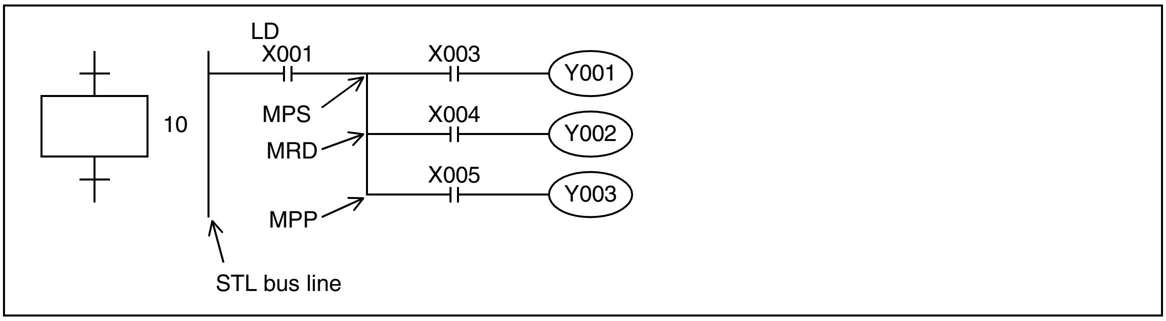


3) Setting to OFF all output relays (Y) in a PLC
Even while the special auxiliary relay M8034 is ON, the sequence program is executed continuously, but all output relays (Y) turn OFF. (These output relays are in the ON status in the monitor.)



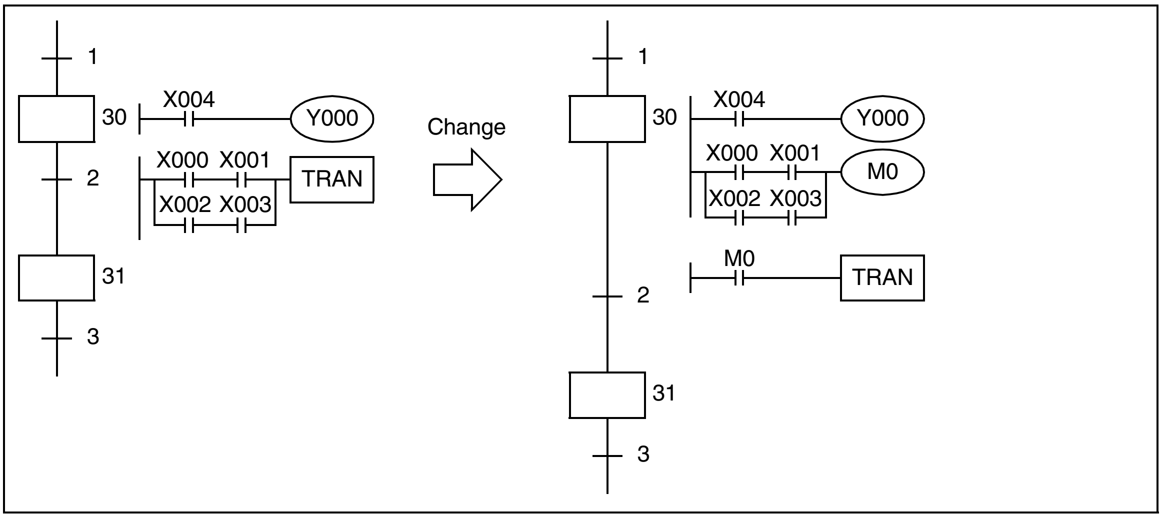
Position of MPS, MRD and MPP instructions

MPS, MRD or MPP instruction cannot be used directly from a bus line in a state relay inside the STL.
Program MPS, MRD or MPP instruction after LD or LDI instruction as shown below.



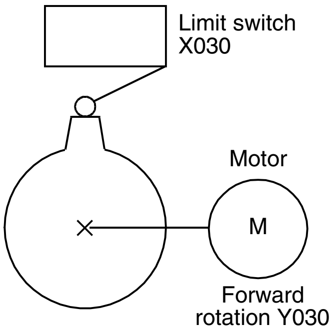
Programming complicated transfer conditions

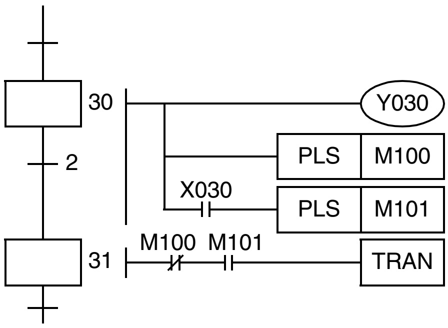
In a transfer condition circuit, ANB, ORB, MPS, MRD and MPP instructions are not available.
Program the circuit as shown below:



Processing of state relay whose transfer condition is already satisfied

In some cases, it is necessary to execute the next transfer after the limit switch X030 (working as the transfer condition) in the ON status is set to OFF once, and then set to ON again.
In such a case, make the transfer condition into pulses as shown below so that transfer is not executed by M100 when S30 turns ON for the first time.



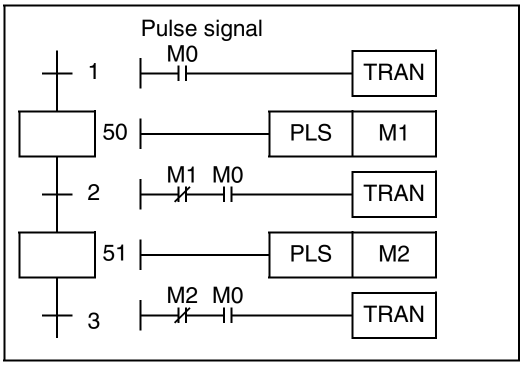


Transfer of state relay ON status by a same signal

In some cases, it is necessary to transfer the state relay ON status by the ON/OFF operation of one pushbutton switch.
To achieve such a transfer, it is necessary to convert the transfer signal into pulses in programming.
The following two methods are available to convert the transfer condition into pulses:

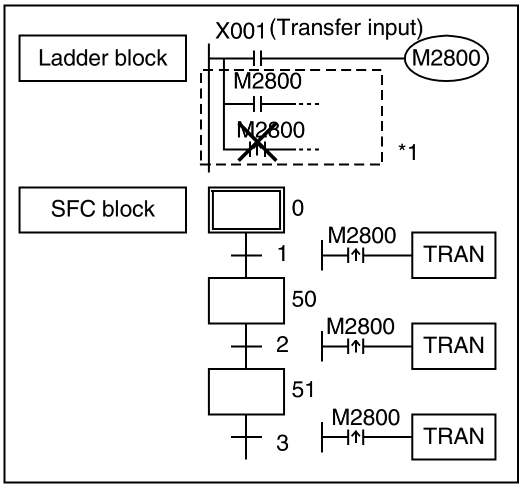
1. Procedure using PLS instruction

Immediately after M0 turns ON and then S50 turns ON, the transfer condition M1 (NC contact) is open. As a result, it is not possible to transfer the ON status to S51 at the same time when S50 turns ON.
When M0 turns ON again, the ON status is transferred to S51.



2. Procedure using a pulse contact instruction (M2800 to M3071)

By using an auxiliary relay M2800 to M3071 in a rising/falling edge detection instruction (LDP, LDF, ANDP, ANDF, ORP or ORF), the ON status can be efficiently transferred by the same signal.
When M2800 or later is specified as a device in a rising/falling edge detection instruction, only the first rising/falling edge detection instruction after a coil instruction is executed. Accordingly, when X001 is set to ON, only the transfer condition in a state relay currently in the ON status is ON during one operation cycle, and then the ON status is transferred to the next state relay.

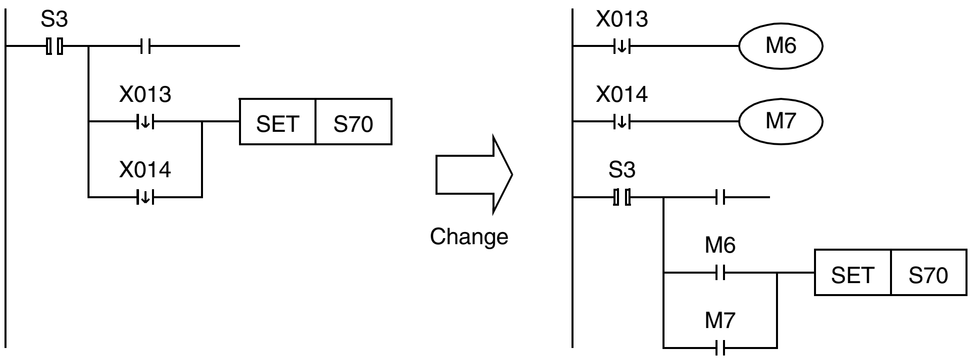


*1. It is alright to program a device number used in LD, LDI, OR or ORI instructions after a corresponding coil in the ladder block. However, if a same device number is programmed in LDP, LDF, ANDP, ANDF, ORP or ORF instruction, the priority is given to such an instruction and the transfer condition is not effective.

Caution on using a rising/falling edge detection contact

When a rising/falling detection contact in LDP, LDF, ANDP, ANDF, ORP or ORF instruction is used in a state relay, the contact whose status was changed while the state relay was OFF is detected when the state relay turns ON the next time.

When it is necessary to immediately detect the rising edge or falling edge for a condition which may change while a state relay is OFF, change the program as shown below:



When the ON status is transferred to S70 at the falling edge of X013 and then X014 turns OFF after that, the falling edge of X014 is not detected at this point because S3 is OFF. When S3 turns ON the next time, the falling edge of X014 is detected.

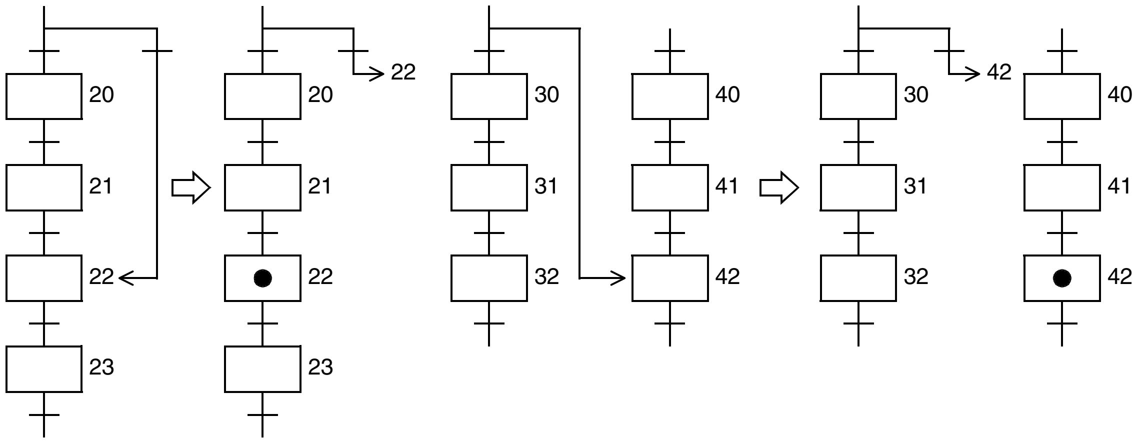
Accordingly, when S3 turns ON the next time, the ON status is immediately transferred to S70.

34.1.8 SFC flow formats

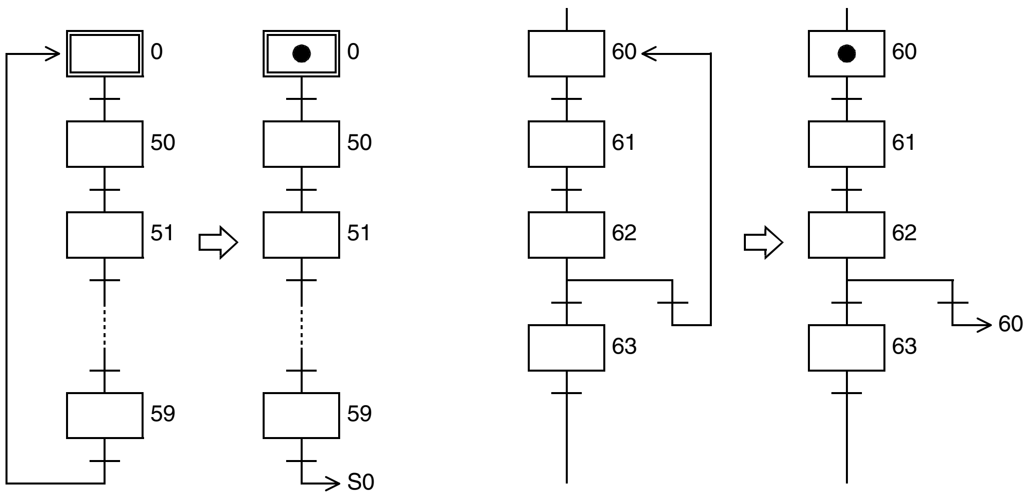
This section shows operation patterns of single flows and operation patterns when selective branches and parallel branches are combined in SFC programs.

1. Jump and repeat flows

1) Jump
Direct transfer to a state relay in a lower position or transfer to a state relay in a different flow is called jump, and the transfer destination state relay is indicated by " ↳".



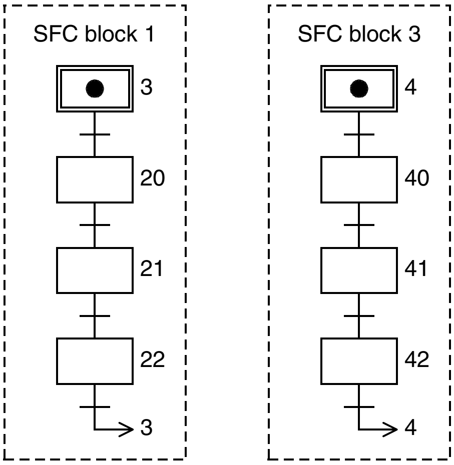
2) Repeat
Transfer to a state relay in an upper position is called repeat, and the transfer destination state relay is indicated by " ↳" in the same way as "jump."



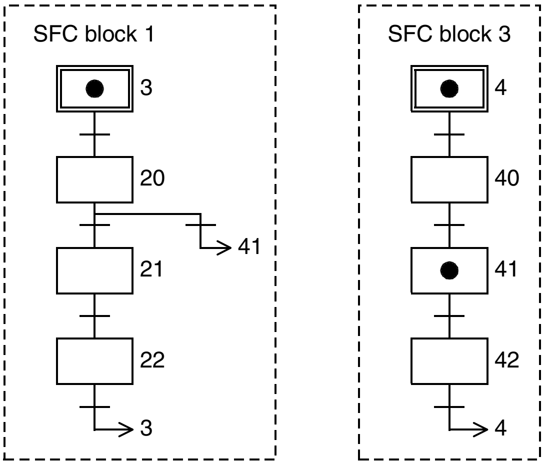
Separation of flow

When creating an SFC program having two or more initial state relays, separate the blocks for each initial state relay.
The ON status can be transferred among SFC programs created by block separation (jump to a different flow).
A state relay in a program created in a different block can be used as a contact for the internal circuit or transfer condition of another state relay.

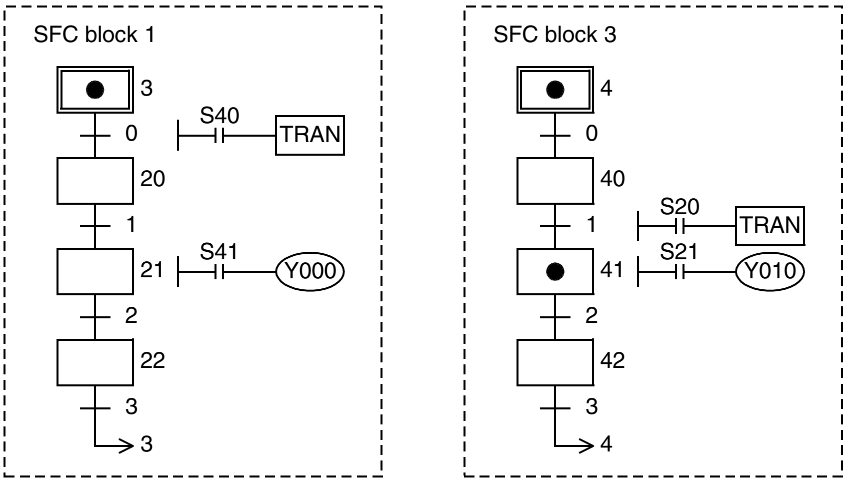
1. Separation of flow



2. Jump to another flow



3. Using a state relay in a program created in a different block

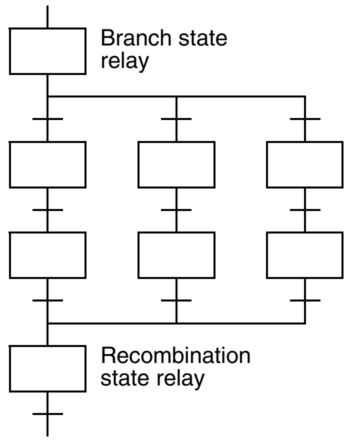


Composite flows with branches and recombination

The single flow format is the fundamental style in process transfer control. Only single flow is sufficient in sequence control for simple operations. When various input conditions and operator manipulations intervene, however, complicated conditions can be easily handled by using selective branches and parallel branches.
A branch for selectively processing many processes depending on a condition is called selective branch. A branch for processing many processes at the same time is called translational branch.

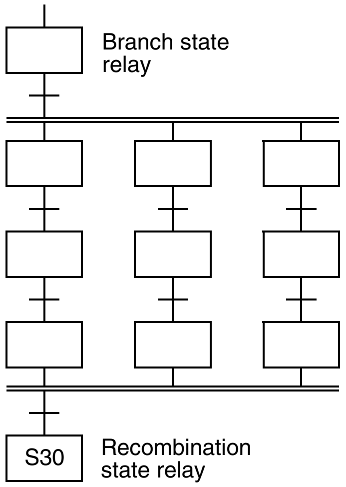
1. **Selective branch**

Either one among many flows is selected and executed.



2. **Parallel branch**

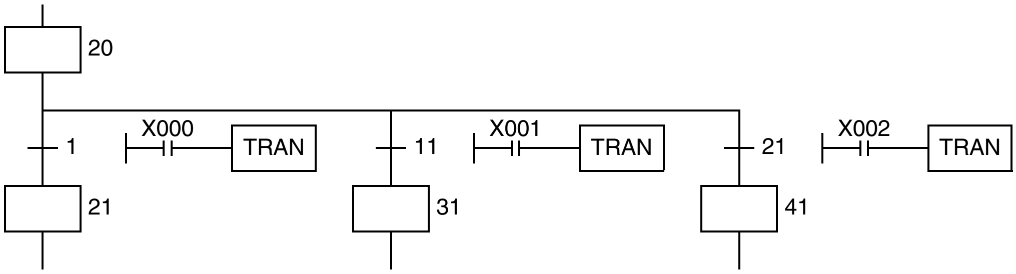
All of many flows are executed at the same time.



34.1.9 Program of branch/recombination state relays

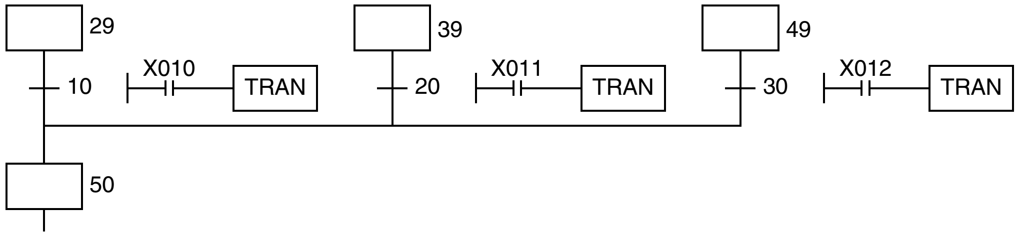
Selective branch

After making a branch, create a transfer condition.



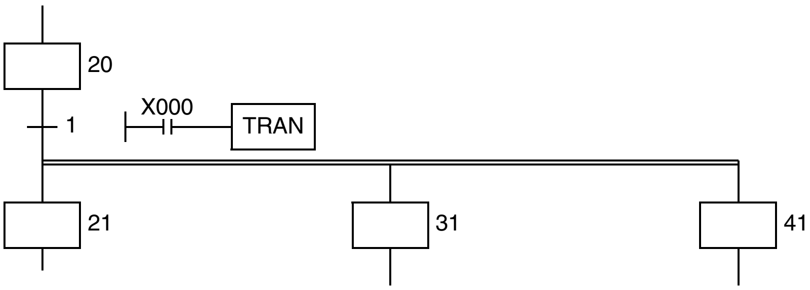
Selective recombination

After creating a transfer condition, recombine.



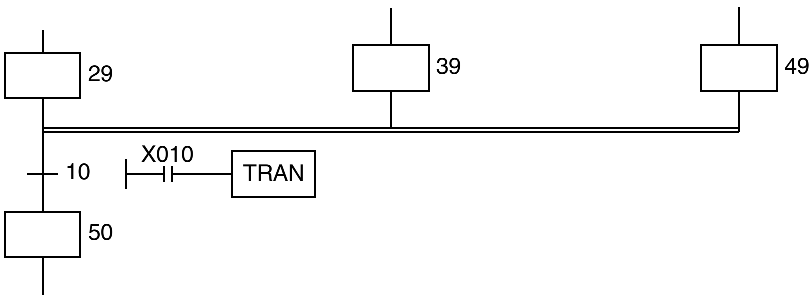
Parallel branch

After creating a transfer condition, make a branch.



Parallel recombination

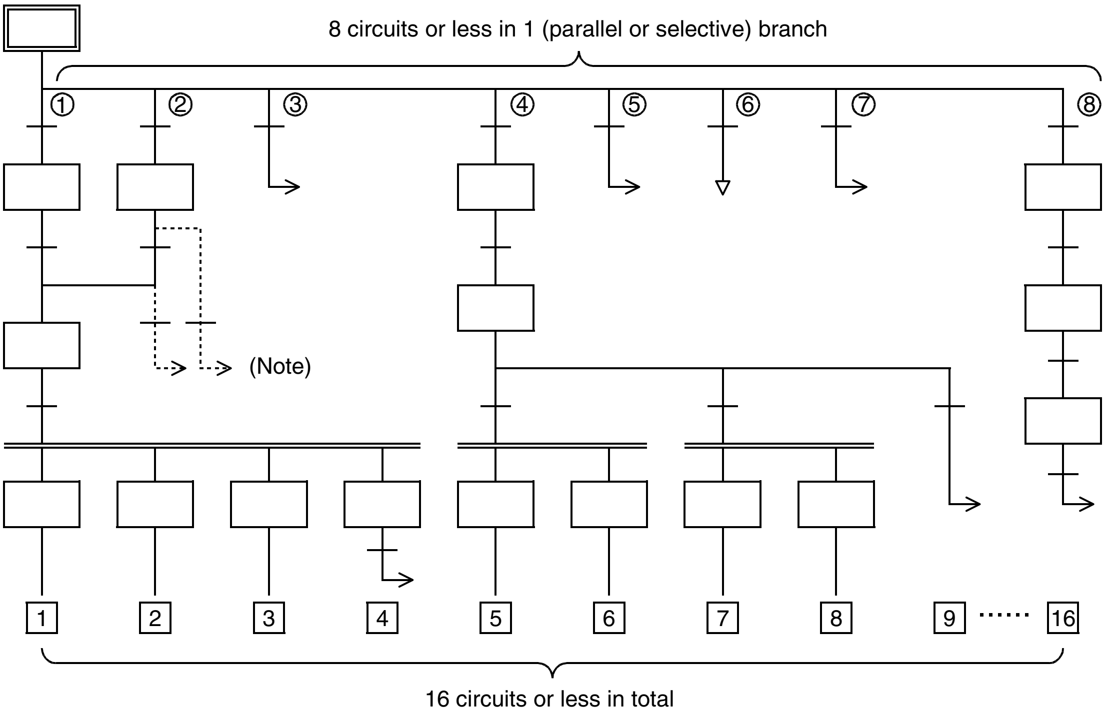
After recombining, create a transfer condition.



34.1.10 Rule for creating branch circuit

Limitation in the number of branch circuits

In one parallel branch or selective branch, up to eight circuits can be provided.
When there are many parallel branches and selective branches, however, the total number of circuits per initial state is limited to 16 or less.



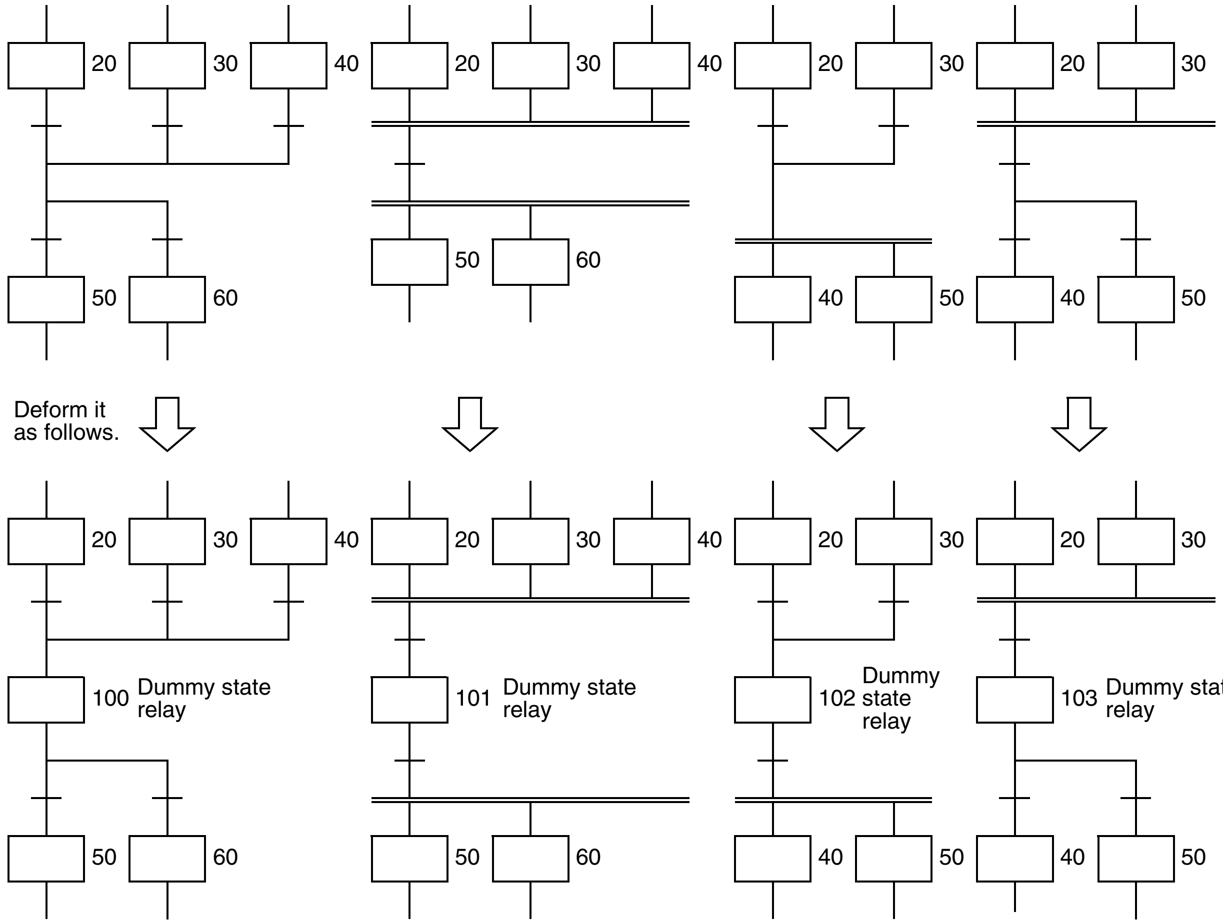
It is not permitted to execute transfer or reset from a recombination line or state relay before recombination to a branch state relay.
Make sure to provide a dummy state, then execute transfer or reset from a branch line to a separate state relay.

31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC•STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List

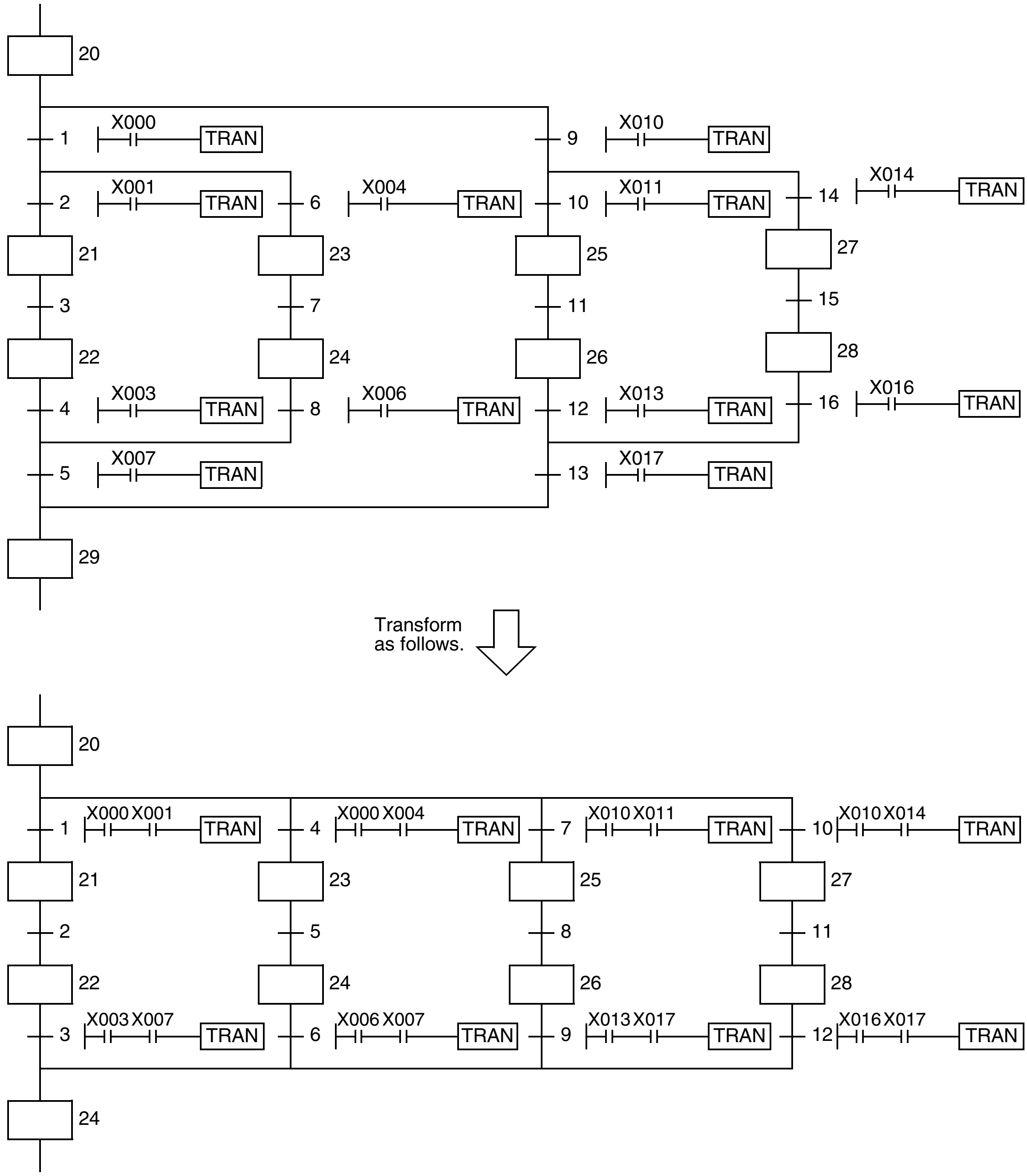
Composition of branches/recombination and dummy state

1. When a recombination line is directly connected to a branch line without a state relay

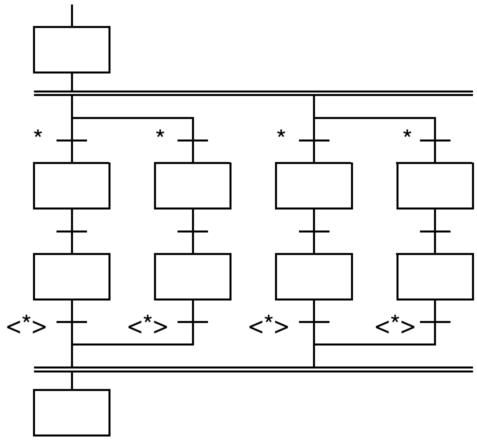
When a recombination line is directly connected to a branch line without a state relay as shown below, it is recommended to provide a dummy state relay between the lines.
There are no dedicated numbers for dummy state relays.
Use a state relay number not used in a program as a dummy state relay.



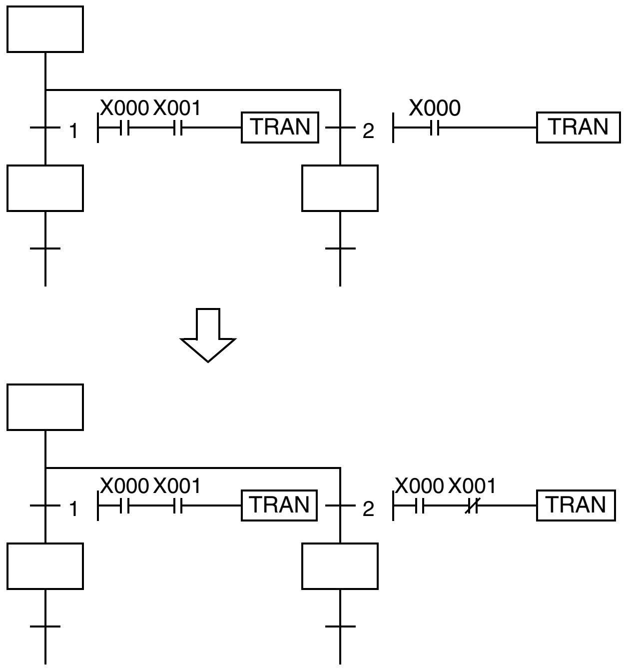
2. When there are selective branches continuously, reduce the number of branches.



3. It is not permitted to provide a selective transfer condition * after parallel branches or to recombine parallel branches after a transfer condition < * >.

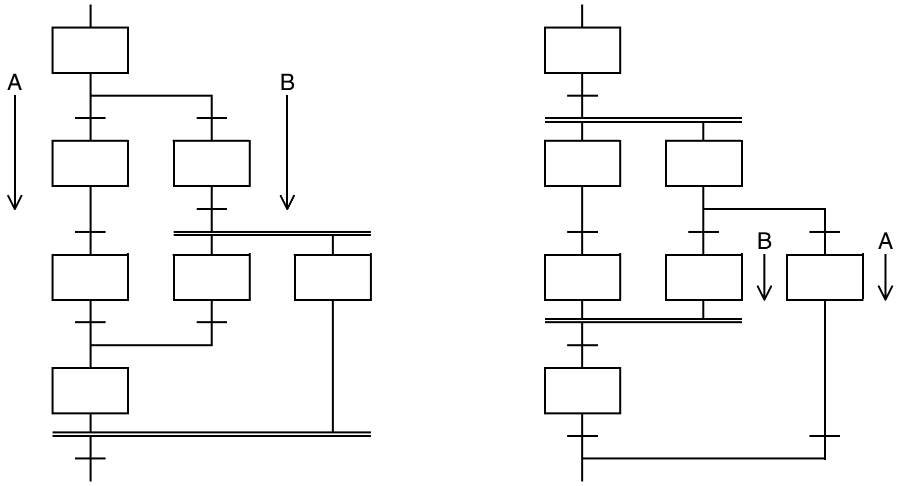


4. In the flow shown below, it is not determined whether a selective or parallel branch is provided. Change it as shown below.



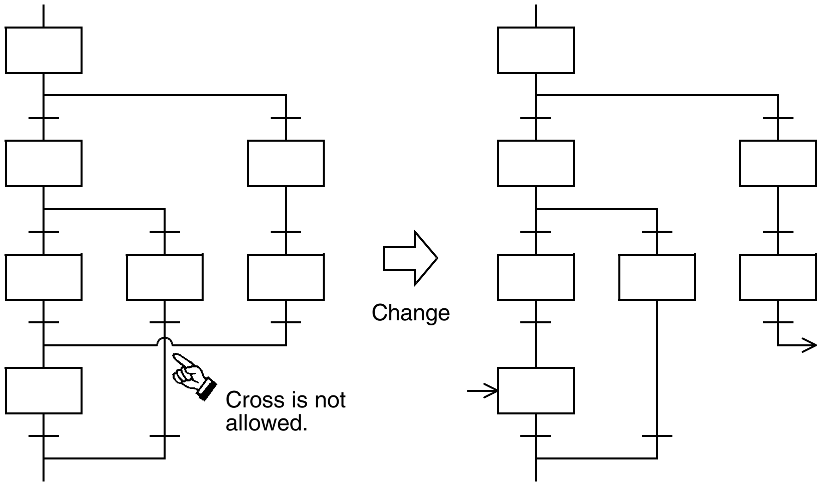
5. The following flows are allowed.

Flow B is alright. In flow A, however, note that an operation is paused at a point where parallel branches are recombined.



6. It is not permitted to cross flows in SFC programs.

Change a flow on the left to a flow on the right. This change enables inverse conversion from a program on the instruction word basis into an SFC program. (The flow on the left cannot be converted into an SFC program.)

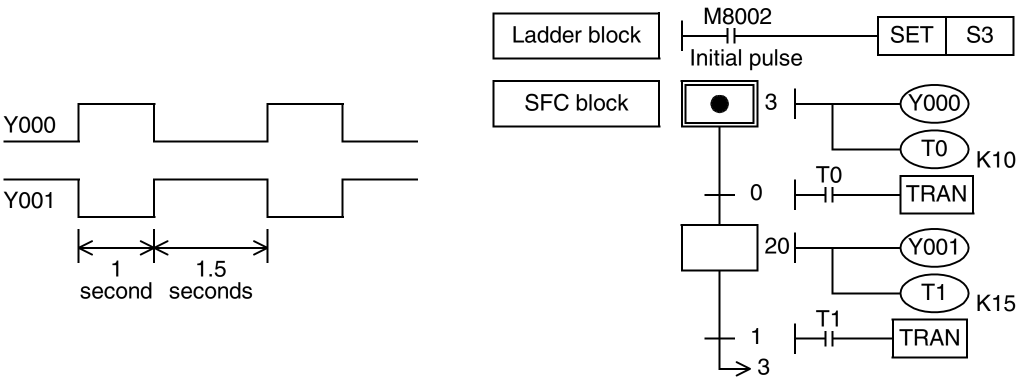


34.1.11 Program examples

Examples of single flows

1. Example of flicker circuit

1) When the PLC mode is changed from STOP to RUN, the state relay S3 is driven by the initial pulse (M8002).
2) The state relay S3 outputs Y000. One second later, the ON status transfers to the state relay S20.
3) The state relay S20 outputs Y001. 1.5 seconds later, the ON status returns to the state relay S3.

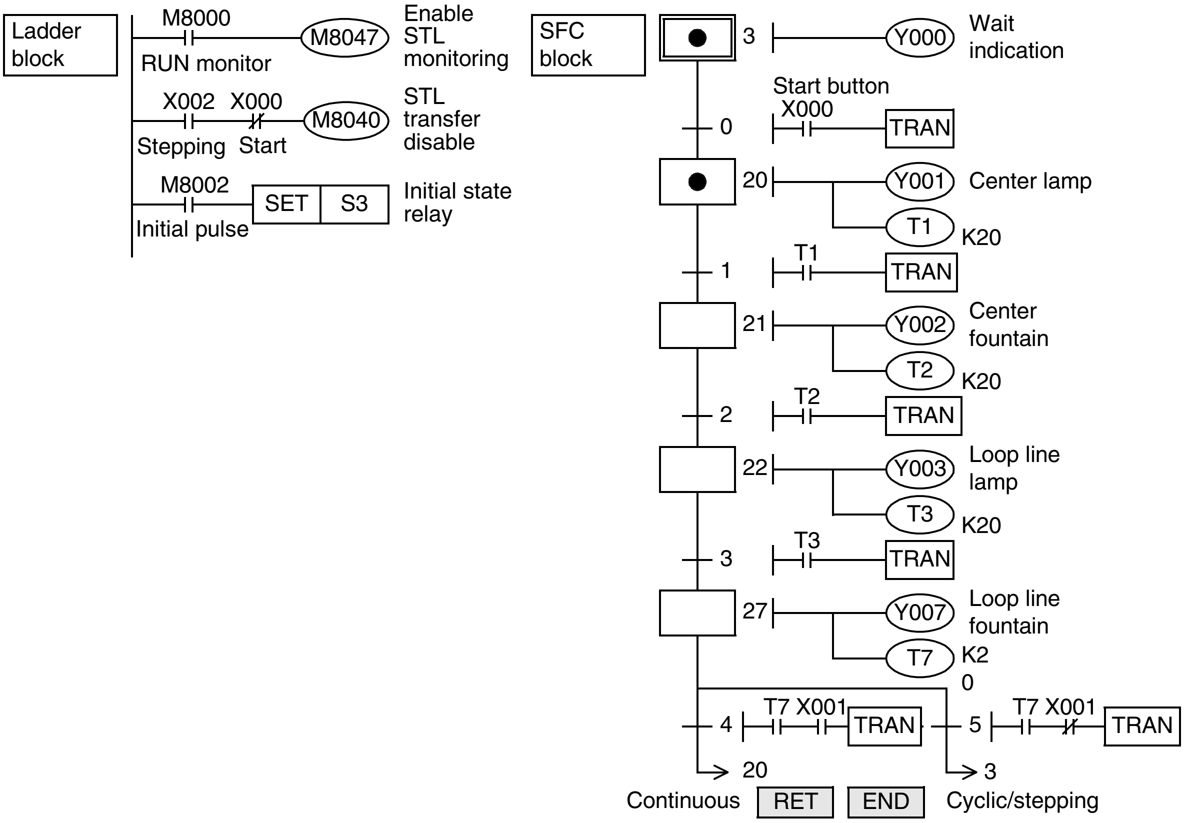


2. Example of fountain control

1) Cyclic operation (X001 = OFF, X002 = OFF)
   When the start button X000 is pressed, the outputs turn ON in the order "Y000 (wait indication) → Y001 (center lamp) → Y002 (center fountain) → Y003 (loop line lamp) → Y007 (loop line fountain) → Y000 (wait indication)", and then the outputs return to the wait status.
   Each output is switched in turn every 2 seconds by a timer.

2) Continuous operation (X001 = ON)
   Y001 to Y007 turn ON in turn repeatedly.

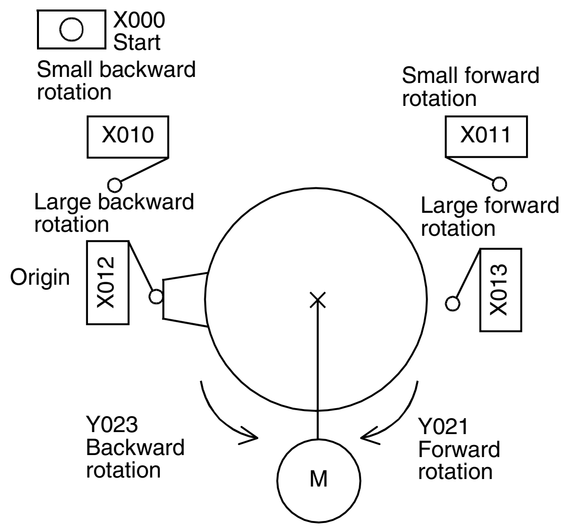
3) Stepping operation (X002 = ON)
   Every time the start button is pressed, each output turns ON in turn.



31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC•STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List

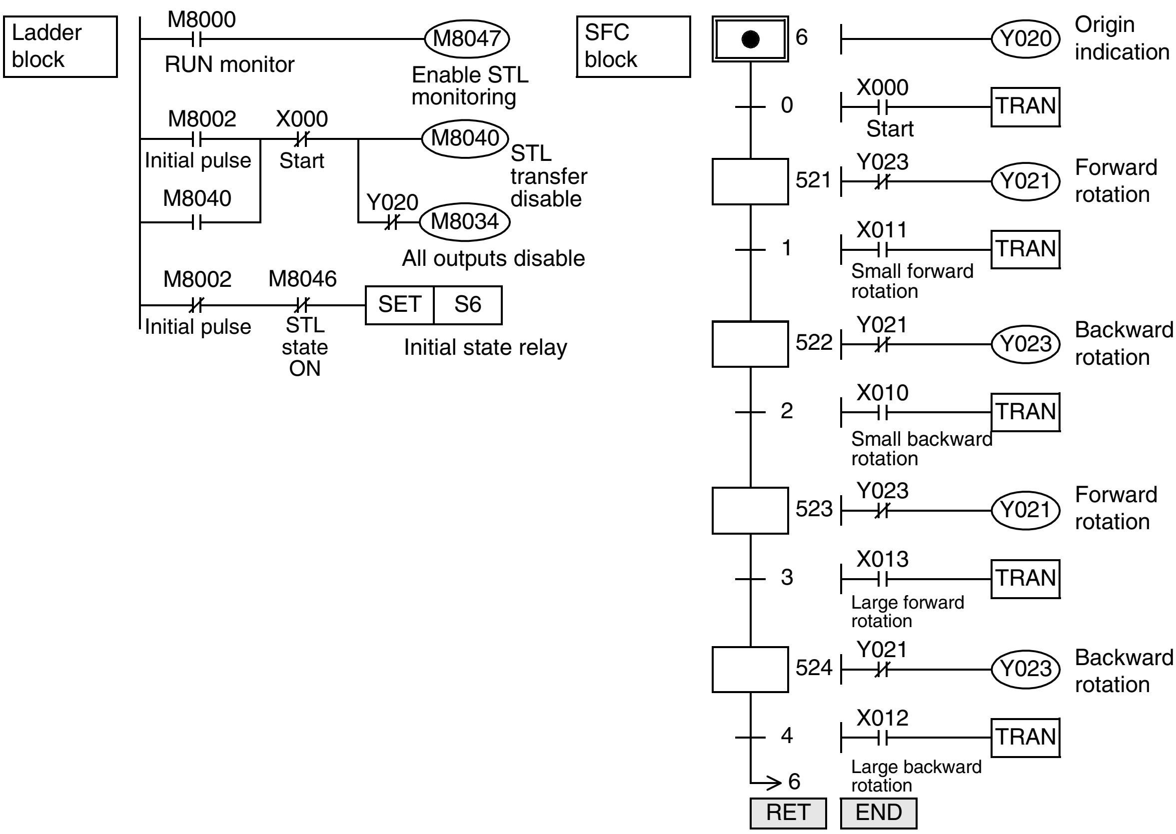
3. Example of cam shaft turning control

The limit switches X013 and X011 are provided in two positions, large forward rotation angle and small forward rotation angle.
The limit switches X012 and X010 are provided in two positions, large backward rotation angle and small backward rotation angle.
When the start button is pressed, the cam shaft performs the operation "small forward rotation → small backward rotation → large forward rotation → large backward rotation", and then stops.



The limit switches X010 to X013 are normally OFF. When the cam shaft reaches a specified angle, a corresponding limit switch turns ON.

- When M8047 turns ON, the operation state monitoring becomes valid. If either one among S0 to S899 and S1000 to S4095 is ON, M8046 turns ON after the END instruction is executed.
- This SFC program adopts latched (battery backed) type state relays so that the operation is restarted from this process when the start button is pressed even after the power is interrupted in the middle of operation. However, all outputs except Y020 are disabled until the start button is pressed.

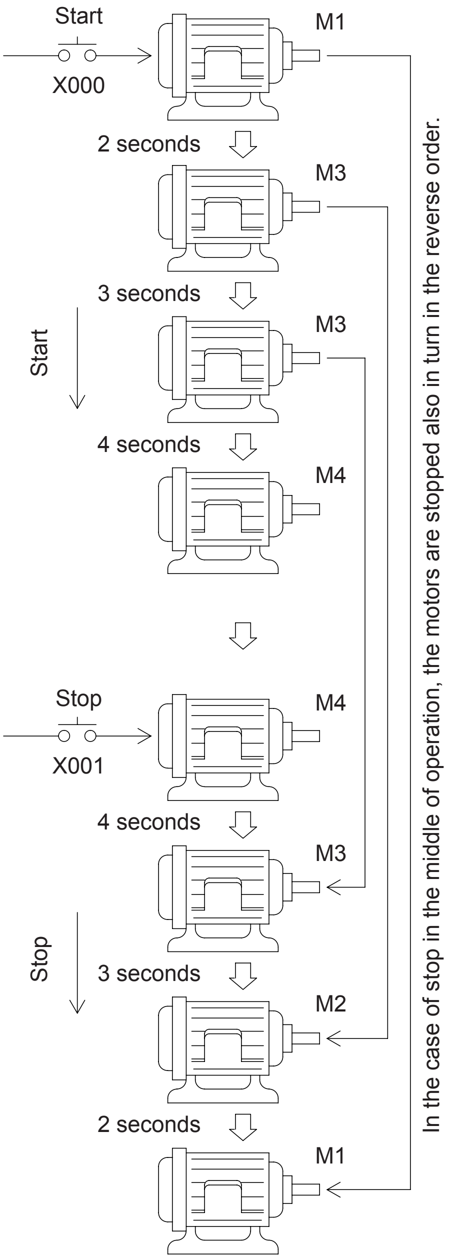


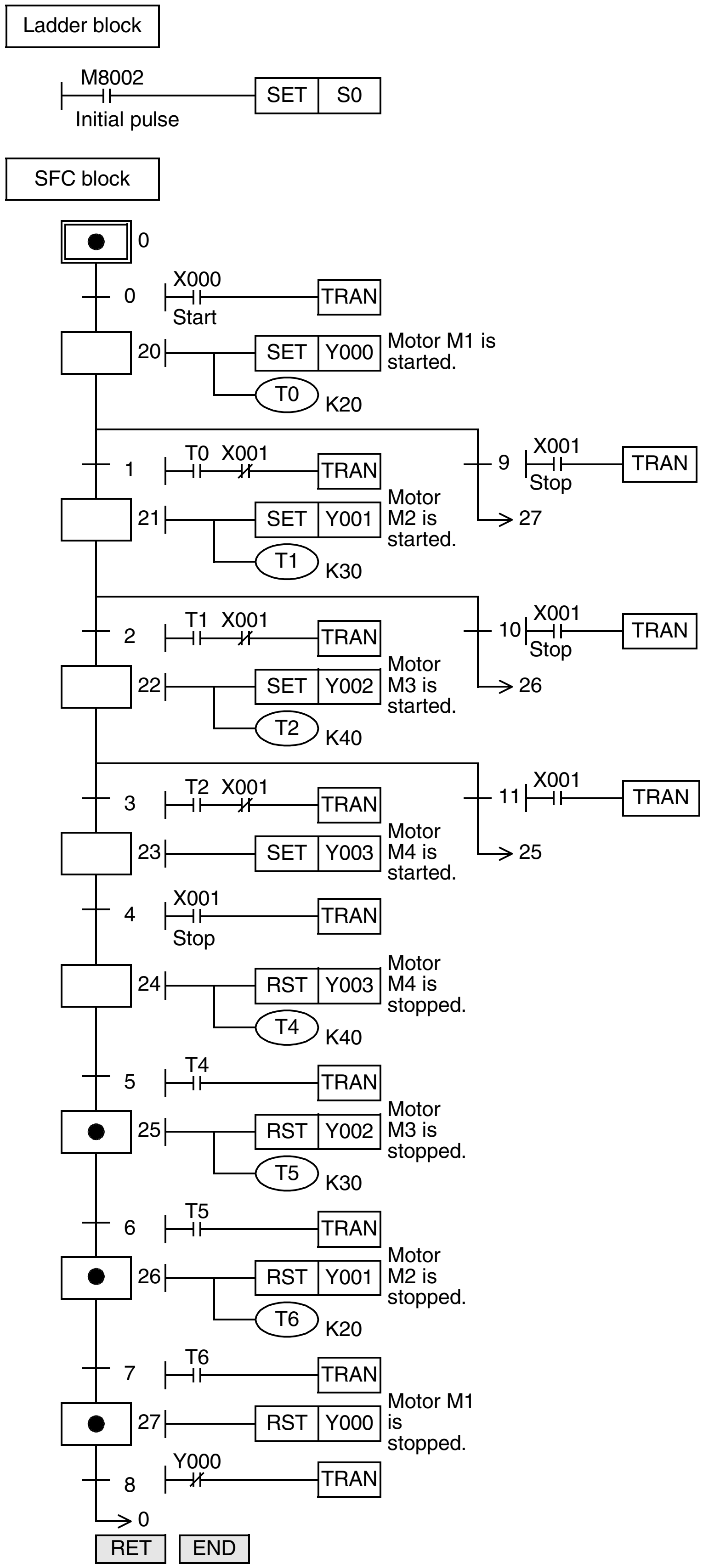
<M8034: All outputs disable>

When M8034 is set to ON, all outputs to the outside turn OFF even though the PLC is executing each program in RUN mode.

4. Example of sequential start and stop

The motors M1 to M4 are started in turn by a timer, and stopped in turn in the reverse order.
This SFC flow is based on a single flow, and has jumps of state relays.

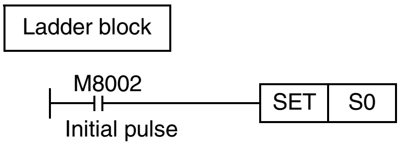


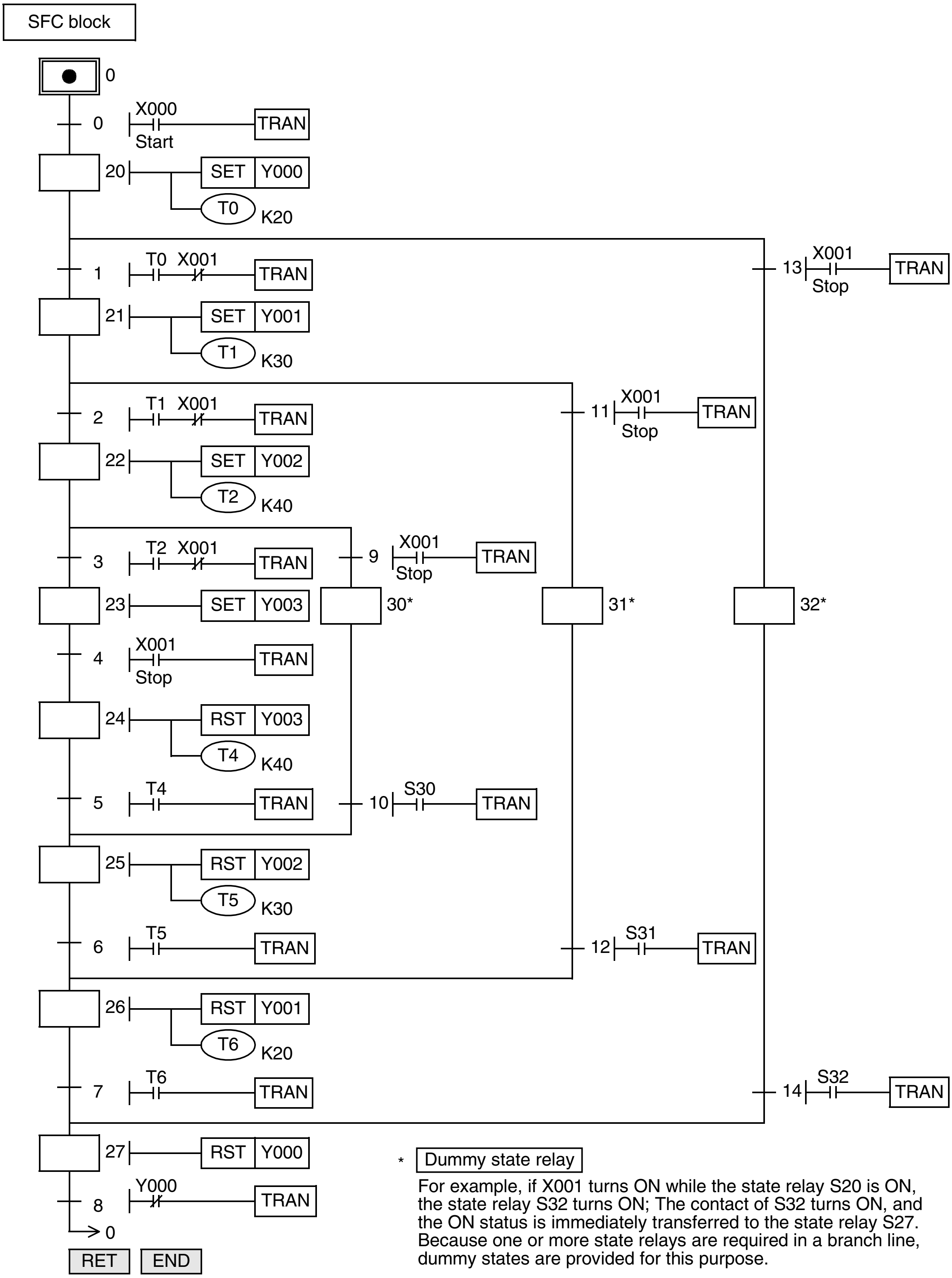


This SFC program shows an example in which a part of the flow is skipped according to a condition, and the execution is transferred to a state in a lower position.
The execution can be transferred to a state in an upper position.

31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC•STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List

The partial skip flow shown on the previous page can be expressed in a flow of selective branches and recombination as shown below.
Make sure that a flow proceeds from top to bottom, and that a flow does not cross except branch lines and recombination lines.



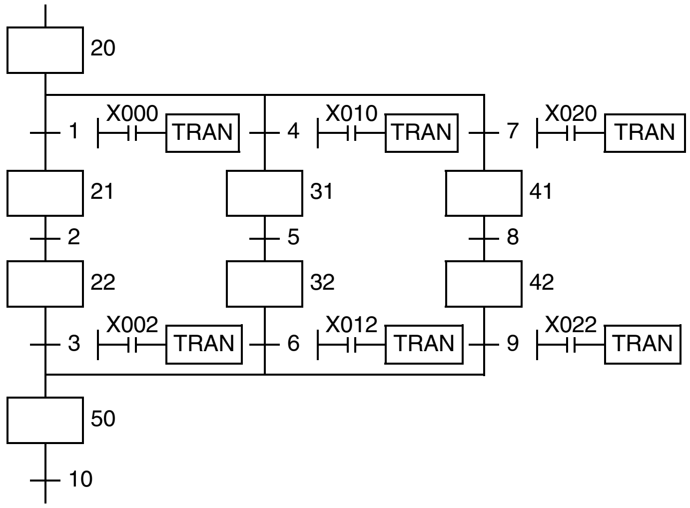


* Dummy state relay

For example, if X001 turns ON while the state relay S20 is ON, the state relay S32 turns ON; The contact of S32 turns ON, and the ON status is immediately transferred to the state relay S27. Because one or more state relays are required in a branch line, dummy states are provided for this purpose.

Examples of flows having selective branches and recombination

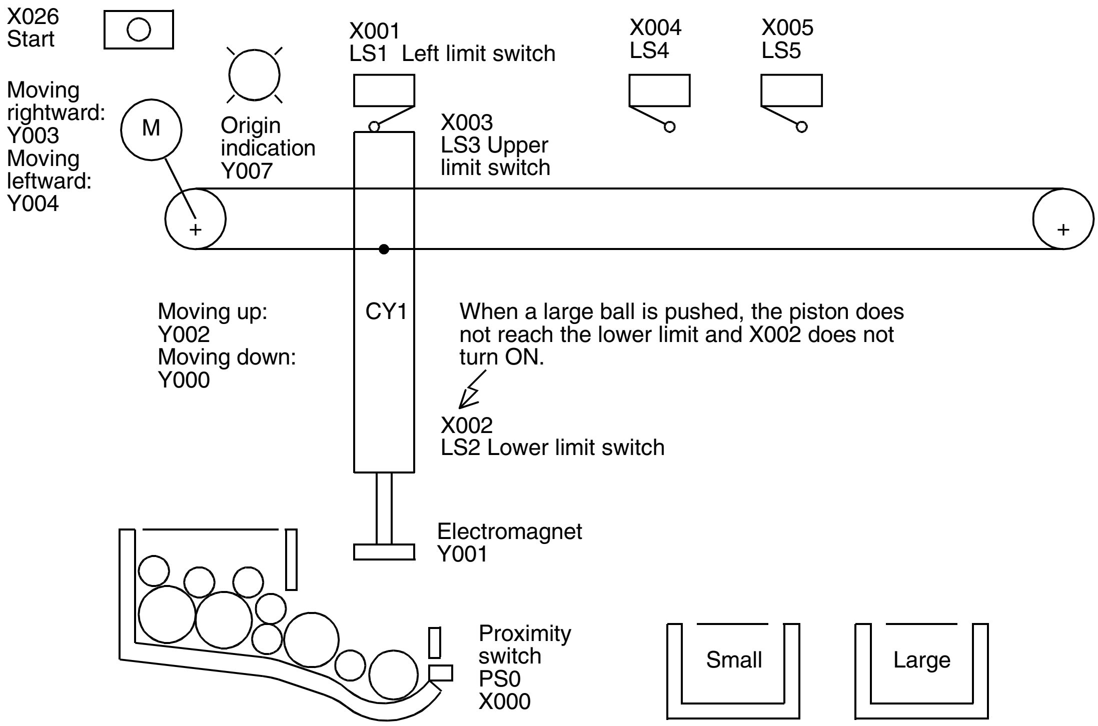
1. Operation of selective branch



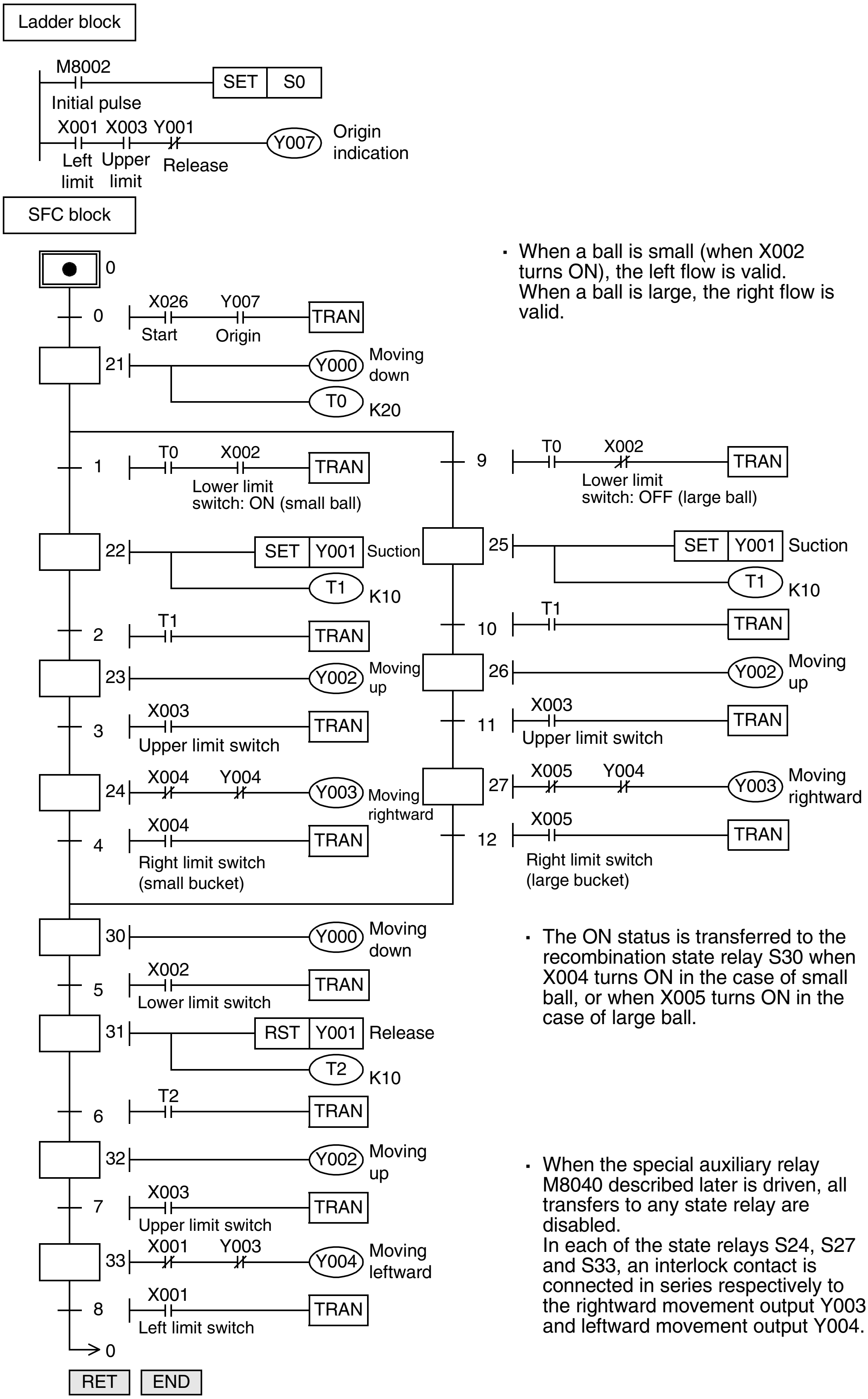
- When two or more flows are provided and either one is selected and executed, it is called a selective branch.
- In the example shown on the right, X000, X010 and X020 should not turn ON at the same time.
- For example, when X000 turns ON while S20 is ON, the ON status is transferred to S21; S20 turns OFF, and S21 turns ON.
  Accordingly, even if X010 or X020 turns ON after that, S31 and S41 do not turn ON.
- The recombination state relay S50 is driven by either one among S22, S32 and S42.

2. Example of selecting and carrying large and small balls

The figure below shows a mechanism which selects and carries large and small balls using conveyors.
The upper left position is regarded as the origin, and the mechanism performs in the order "moving down → suction → moving up → moving rightward → moving down → release → moving up → moving leftward".
When the arm moves down and the electromagnet pushes a large ball, the lower limit switch LS2 turns OFF.
When the electromagnet pushes a small ball, LS2 turns ON.



In an SFC program for selecting large and small products or judging products as accepted or rejected, selective branches and recombination are adopted as shown in the figure below.



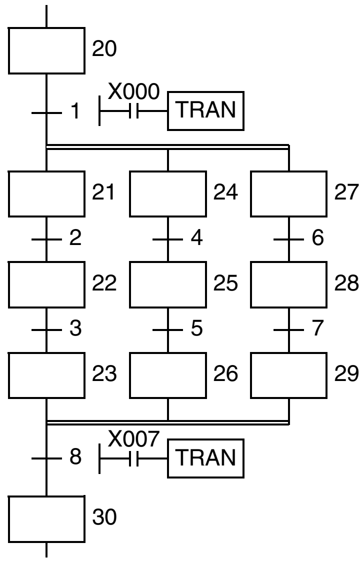
- When a ball is small (when X002 turns ON), the left flow is valid. When a ball is large, the right flow is valid.
- The ON status is transferred to the recombination state relay S30 when X004 turns ON in the case of small ball, or when X005 turns ON in the case of large ball.
- When the special auxiliary relay M8040 described later is driven, all transfers to any state relay are disabled. In each of the state relays S24, S27 and S33, an interlock contact is connected in series respectively to the rightward movement output Y003 and leftward movement output Y004.

Example of flows having parallel branches and recombination

1. Operation of parallel branch

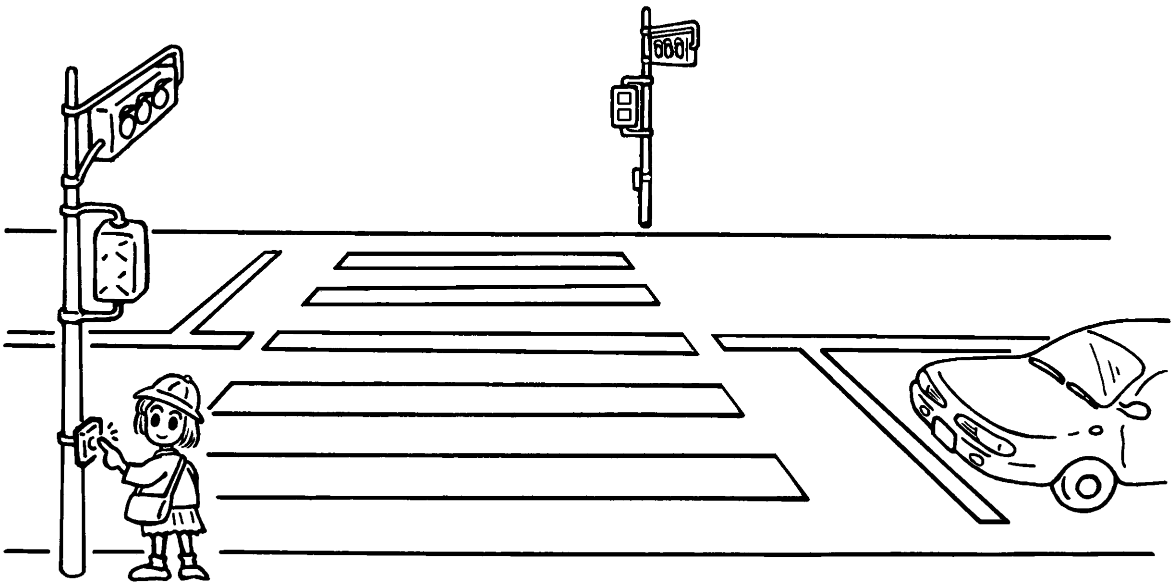
- Branches in which all flows proceed at the same time are called parallel branches.
- In the example shown on the left, when X000 turns ON while S20 is ON, S21, S24 and S27 turn ON at the same time and the operation is started in each flow.
- When the operation is finished in each flow and X007 turns ON, the recombination state relay S30 turns ON. S23, S26 and S29 turn OFF.
- Such recombination is sometimes called wait recombination. (The original flow continues its operation until all flows finish their operations and join the original flow.)

When the parts A, B and C are processed in parallel and then assembled afterward, flows having parallel branches and recombination are used.



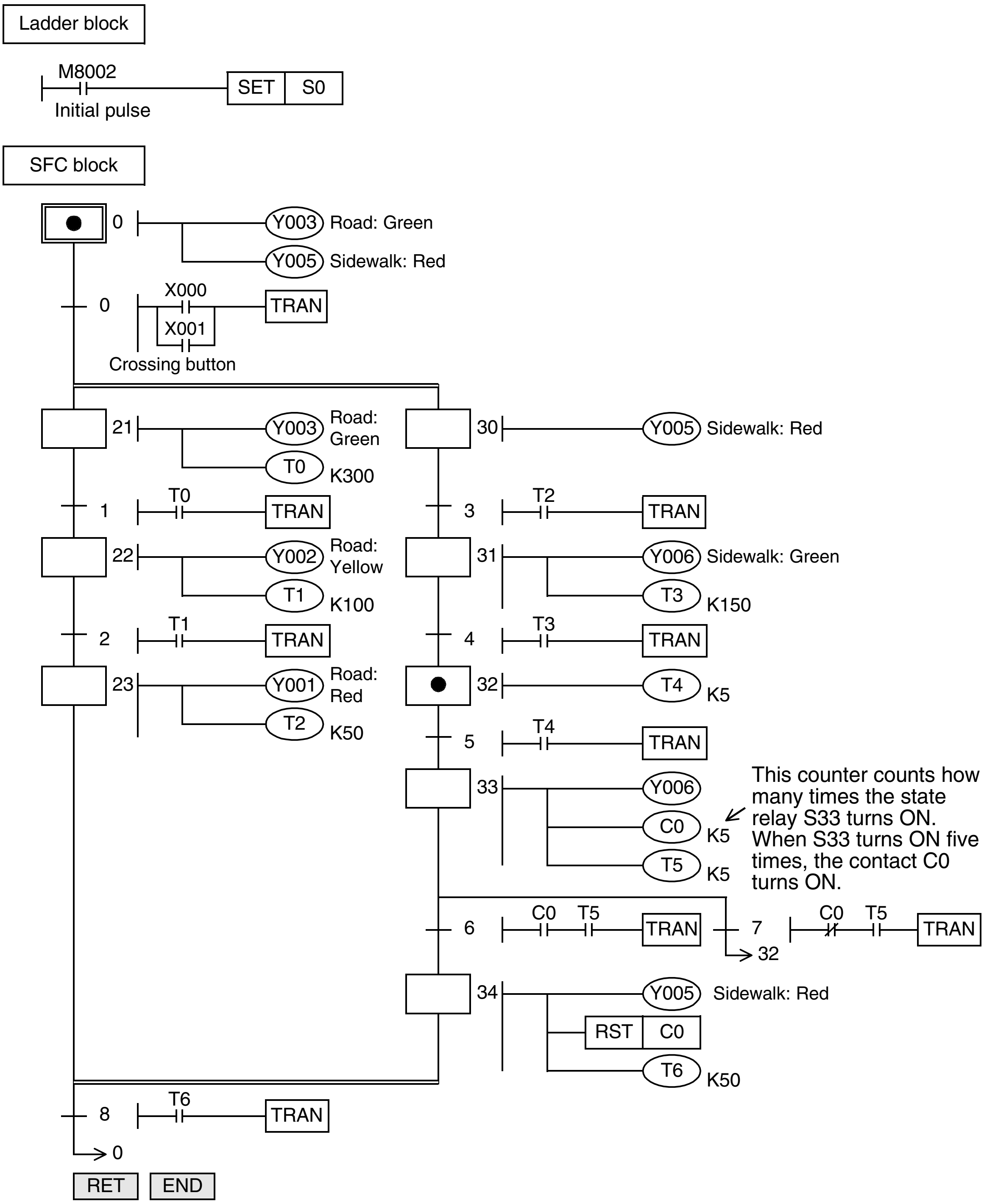
2. Example of pushbutton type crosswalk

A pushbutton type crosswalk shown in the figure below can be expressed in flows having parallel branches and recombination.
Y003: Green Y002: Yellow Y001: Red



The SFC program for a pushbutton type crosswalk is as shown below. In this example, a partial flow (jump to a state relay located in an upper position) is repeated for blinking the green lamp on the crosswalk.

- When the PLC mode is changed from STOP to RUN, the initial state relay S0 turns ON. Normally, the green lamp is ON for the road and the red lamp is ON for the sidewalk.
- When the crossing button X000 or X001 is pressed, the state relay S21 specifies “road: green” and the state relay S30 specifies “sidewalk: red”. The signal lamp status is not changed.
- 30 seconds later, the yellow lamp turns ON for the road. 10 seconds later after that, the red lamp turns ON for the road.
- When the timer T2 (5 seconds) reaches timeout after that, the green lamp turns ON for the sidewalk.
- 15 seconds later, the green lamp starts to blink for the sidewalk. (S32 turns OFF the green lamp, and S33 turns ON the green lamp.)
- While the green lamp is blinking, S32 and S33 turn ON and OFF repeatedly. When the counter C0 (set value: 5) turns ON, S34 turns ON. 5 seconds after the red lamp turns ON for the sidewalk, the signal lamps return to the initial state.
- Even if the crossing button X000 or X001 is pressed in the middle of operation, the pressing is ignored.


Ladder block
M8002
Initial pulse
SET S0
SFC block
0
Y003 Road: Green
Y005 Sidewalk: Red
0
X000
X001
Crossing button
TRAN
21
Y003 Road: Green
T0 K300
1
T0
TRAN
22
Y002 Road: Yellow
T1 K100
2
T1
TRAN
23
Y001 Road: Red
T2 K50
30
Y005 Sidewalk: Red
3
T2
TRAN
31
Y006 Sidewalk: Green
T3 K150
4
T3
TRAN
32
T4 K5
5
T4
TRAN
33
Y006
C0 K5
T5 K5
This counter counts how many times the state relay S33 turns ON. When S33 turns ON five times, the contact C0 turns ON.
6
C0 T5
TRAN
7
C0 T5
TRAN
32
34
Y005 Sidewalk: Red
RST C0
T6 K50
8
T6
TRAN
0
RET
END

34.2 Step Ladder

34.2.1 Outline

In programs using step ladder instructions, a state relay State S is assigned to each process based on machine operations, and sequences of input condition and output control are programmed as circuits connected to contacts (STL contacts) of state relays in the same way as SFC programs.
The concept of program creation and the types and operations of state relays are the same as for SFC programs. However, because the contents can be expressed in the ladder format, step ladder programs can be handled as familiar relay ladder charts even though the actual contents are the same as those of SFC programs.
In step ladder programs, the list format is also available.
SFC programs and step ladder programs can be converted each other if they are programmed in the same rules respectively.

This section explains the expressions and cautions of step ladder programs in comparison with SFC programs, and the input order in the list format.

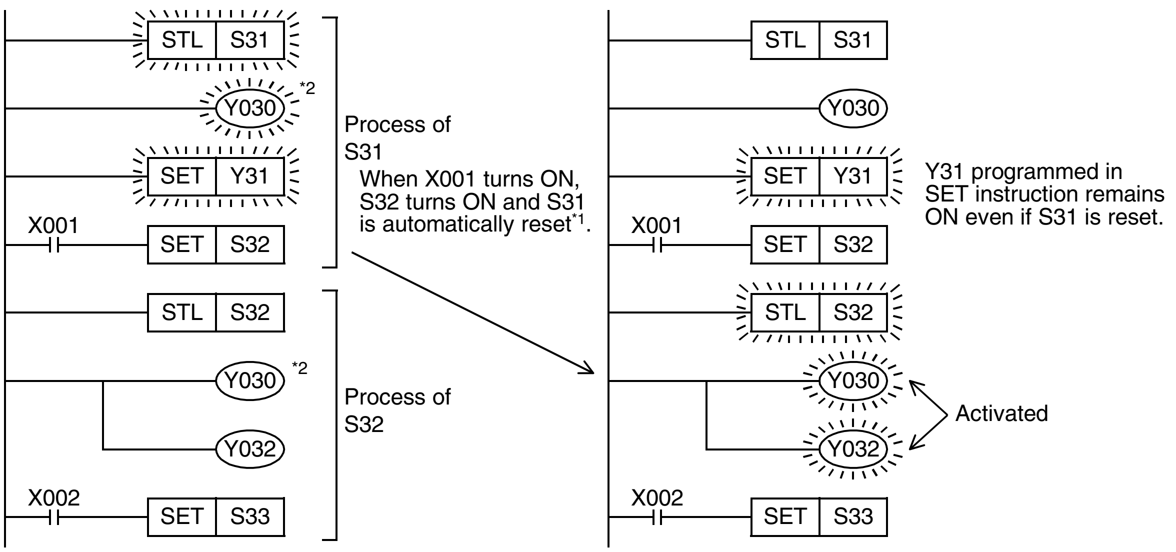
34.2.2 Explanation of function and operation

In a step ladder program, a state relay State S is regarded as one control process, and a sequence of input condition and output control are programmed in a state relay.
Because the preceding process is not performed any more when the program execution proceeds to the next process, a machine can be controlled using simple sequences for each process.

Operation of step ladder instructions

In a step ladder program, each process performed by the machine is expressed by a state relay.
A state relay consists of a drive coil and contact (STL contact) in the same way as other relays.
Use SET or OUT instruction to drive a coil, and use STL instruction for a contact.

- When a state relay turns ON, a connected circuit (internal circuit) is activated by way of an STL contact.
  When a state relay turns OFF, a connected internal circuit is deactivated by way of an STL contact.
  After one operation cycle, non-driving of an instruction (jump status) is not available.
- When a condition (transfer condition) provided between state relays is satisfied, the next state relay turns ON, and the state relay which has been ON so far turns OFF (transfer operation).
  In the state relay ON status transfer process, the both state relays are ON only instantaneously (during one operation cycle).
  In the next operation cycle after the ON status was transferred, the former state is reset to OFF.
- One state relay number can be used only once.

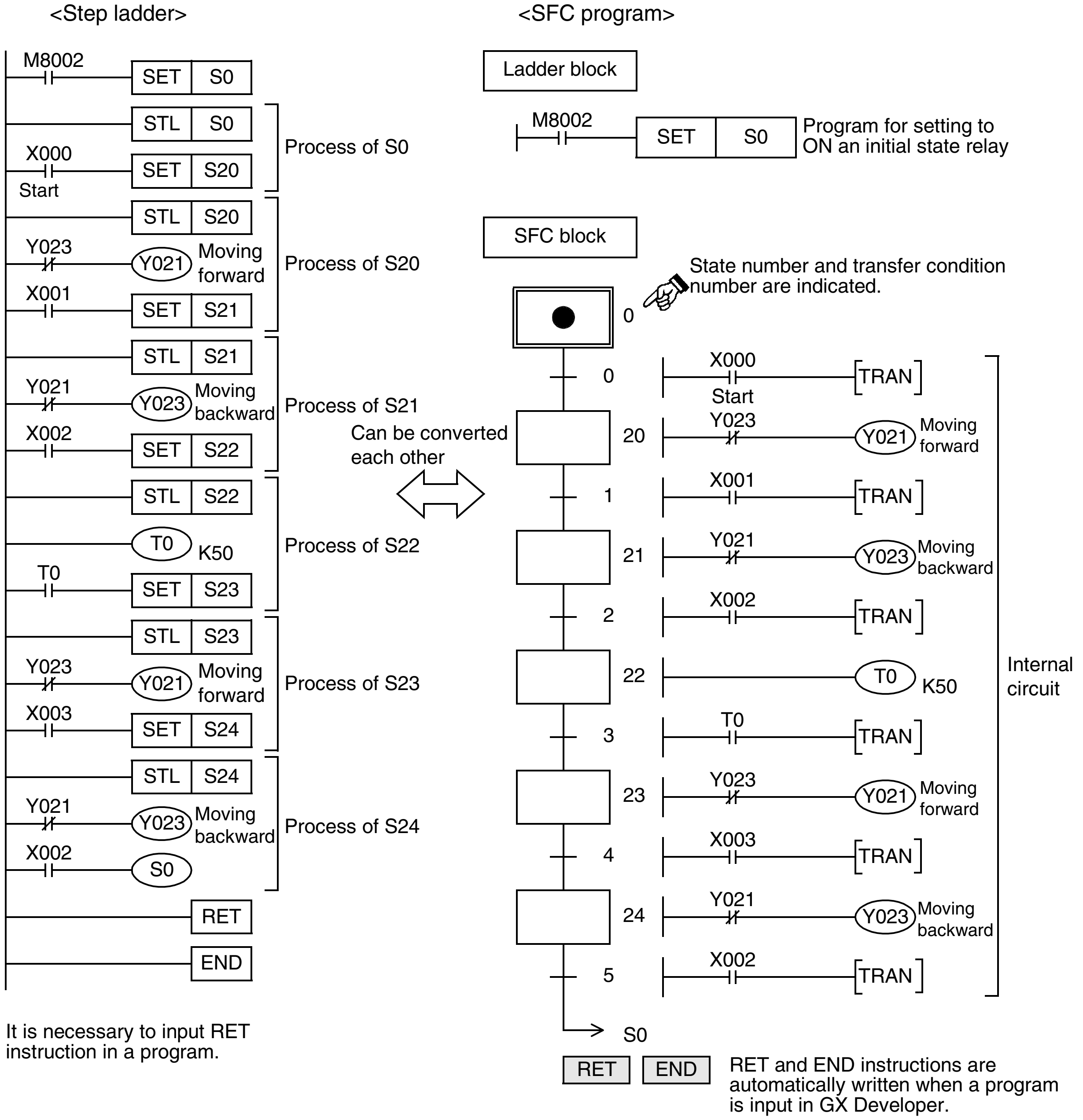


*1. In the next operation cycle after the ON status was transferred from a former state relay to a next state relay, the former state relay turns OFF.

*2. An output coil can be used again in different state relays.

34.2.3 Expression of step ladder

Step ladder programs and SFC programs are substantially the same as described above, but actual programs are expressed as shown below.
A step ladder program is expressed as relay ladder, but it can be created according to the machine control flow using state relays.

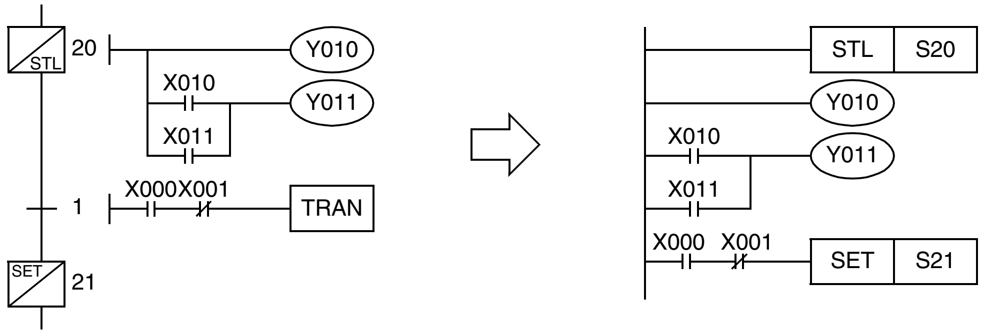


34.2.4 Creation of step ladder program (SFC program → STL program)

The figure on the left shows one state relay extracted as an example from an SFC program.
Each state relay has three functions, driving a load, specifying a transfer destination and specifying a transfer condition.
The step ladder shown on the right expresses this SFC program as a relay sequence.
In this program, a load is driven, and then the ON status is transferred.
In a state relay without any load, the drive processing is not required.
For the program creation procedure, refer to the description on SFC programs.

→ For the program creation procedure, refer to Subsection 34.1.3.
→ For the handling and role of initial state relays, refer to Subsection 34.1.4.
→ For latched (battery backed) type state relays, refer to Subsection 34.1.5.
→ For RET instruction, refer to Subsection 34.1.6.

<SFC program> <Step ladder>



<List program>

0	STL	S20
1	OUT	Y010
2	LD	X010
3	OR	X011
4	OUT	Y011
5	LD	X000
6	ANI	X001
7	SET	S21 *1

The above program can be expressed in the list format (list program) shown on the left.
The segment from the STL instruction to the RET instruction is handled as a step ladder program.

*1. SET and RST instructions for a state relay are two-step instructions.

- When every state relay used in an SFC program is defined, programming is complete.
- Program a step ladder program starting from the initial state relay in the order of state relay ON status transfer.
  Make sure to put the RET instruction at the end of a step ladder program.

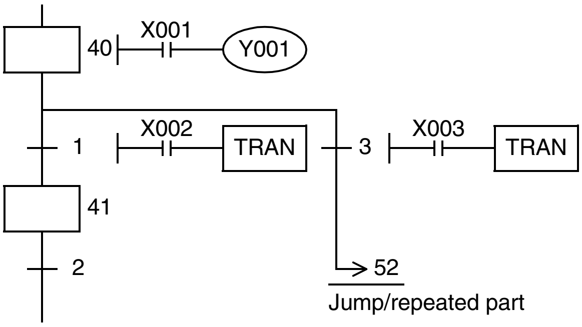
31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC•STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List

Program with jump/repeated flows

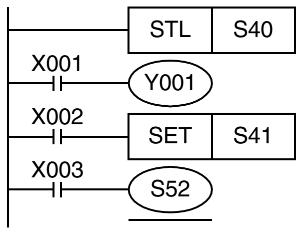
1. Program for the transfer source

Use OUT instruction in the jump/repeated part.

<SFC program>



<Step ladder>



<List program>

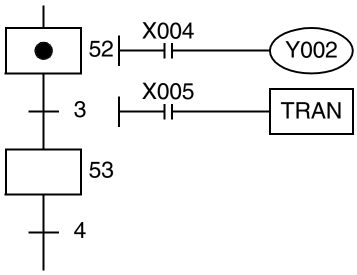
```
STL  S40
LD   X001
OUT  Y001
LD   X002
SET  S41
LD   X003
OUT  S52
```

Use OUT instruction in the jump/repeated part.

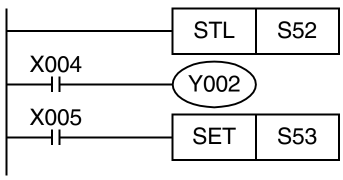
2. Program for the transfer destination

There is no change in programming especially for the transfer destination.

<SFC program>



<Step ladder>



<List program>

```
STL  S52
LD   X004
OUT  Y002
LD   X005
SET  S53
```

34.2.5 Preliminary knowledge for creating step ladder programs

Refer to the preliminary knowledge for creating SFC programs also.

→ **For the preliminary knowledge for creating SFC programs, refer to Subsection 34.1.7.**

List of sequence instructions available between STL instruction and RET instruction

State relay		Instruction		
		LD/LDI/LDP/LDF, AND/ANI/ANDP/ANDF, OR/ORI/ORP/ORF, INV,OUT,SET/RST, PLS/PLF	**ANB/ORB/MPS/MRD/ MPP**	**MC/MCR**
Initial/general state relay		Available	Available*1	Not available
Branch/ recombination state relay	Drive processing	Available	Available*1	Not available
	Transfer processing	Available	Not available	Not available

- STL instruction cannot be used in interrupt programs and subroutine programs.
- It is not prohibited to use jump instructions in state relays. But it is not recommended to use jump instructions because complicated movements will be resulted.

*1. MPS instruction cannot be used immediately after an STL instruction, even in a drive processing circuit.

Special auxiliary relays

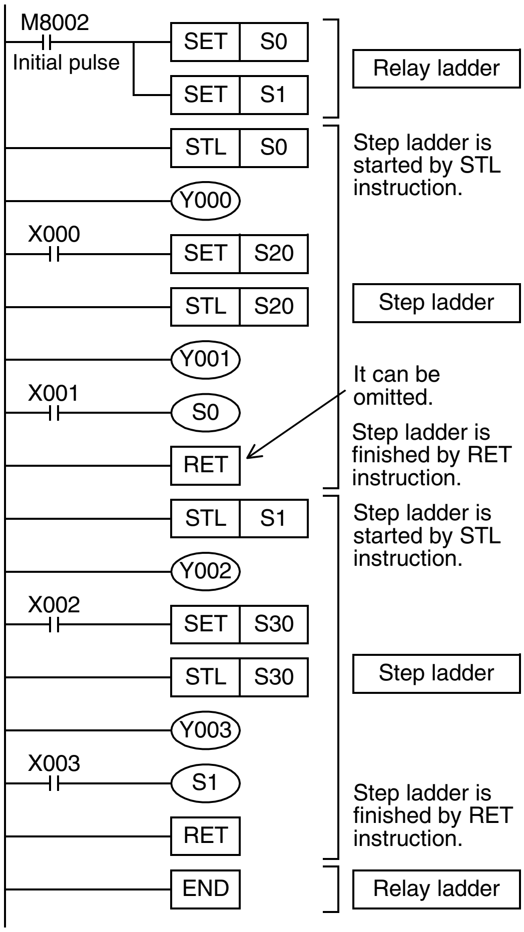
For efficiently creating step ladder programs, it is necessary to use some special auxiliary relays. The table below shows major ones.
The special auxiliary relays shown below are the same as those available in SFC programs.

Device number	Name	Function and application
M8000	RUN monitor	This relay is normally ON while the PLC is in the RUN mode. Use this relay as the program input condition requiring the normally driven status or for indicating the PLC operation status.
M8002	Initial pulse	This relay turns ON and remains ON only instantaneously when the PLC mode is changed from STOP to RUN. Use this relay for the initial setting of a program or for setting the initial state relay.
M8040	STL transfer disable	When this relay is set to ON, transfer of the ON status is disabled among all state relays. Because programs in state relays are operating even in the transfer disabled status, output coils do not turn OFF automatically.
M8046*1	STL state ON	When any state relay turns ON, this relay automatically turns ON. Use this relay to prevent simultaneous start up of another float or as a process ON/OFF flag.
M8047*1	Enable STL monitoring	When this relay is driven, the device number of a state relay in the ON status having the smallest device number among S0 to S899 and S1000 to S4095 is stored to D8040, and the state relay number in the ON status having the next smallest device number is stored to D8041. In this way, up to eight state relays in the ON status are stored up to D8047. • In the FX-PCS/WIN(-E), FX-20P(-E) and FX-10P(-E), when this relay is driven, the state relays in the ON status are automatically read and displayed. For details, refer to the manual of each peripheral equipment. • In the SFC monitor in GX Developer, the automatic scroll monitoring function is valid even if this relay is not driven.

*1. Processed when END instruction is executed.

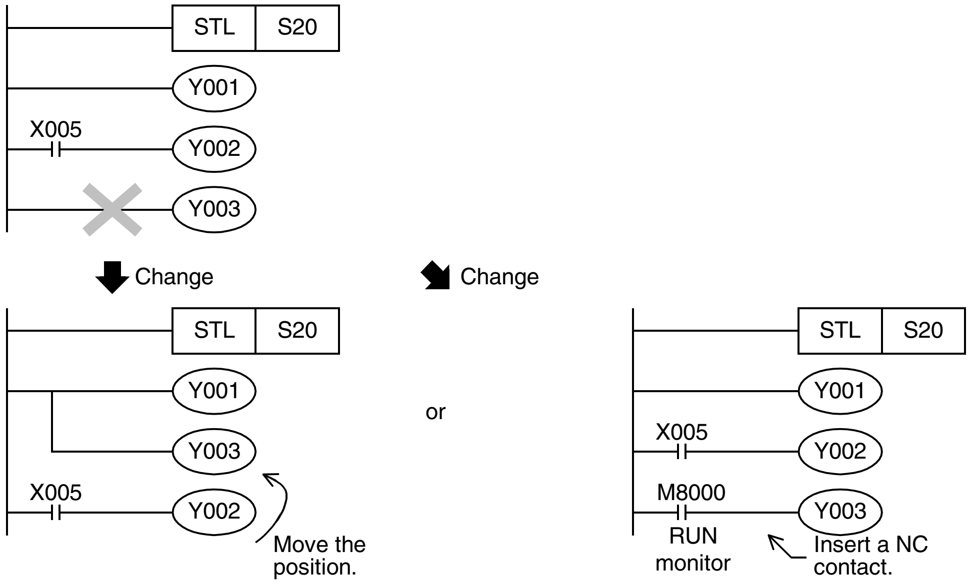
Block

A step ladder program is created as ladder circuits in the same way as relay ladder. Accordingly, different from SFC programs, it is not necessary to divide blocks for relay ladder parts and SFC parts.
When there are ladder blocks and SFC blocks, put RET instruction at the end of each step ladder program.
A PLC starts the step ladder processing by STL instruction, and returns to the relay ladder processing from the step ladder processing by RET instruction.
However, when consecutively programming a step ladder in a different flow (when there is no relay ladder before the step ladder in the different flow), RET instruction between flows can be omitted, and RET instruction can be programmed only at the end of the last flow.



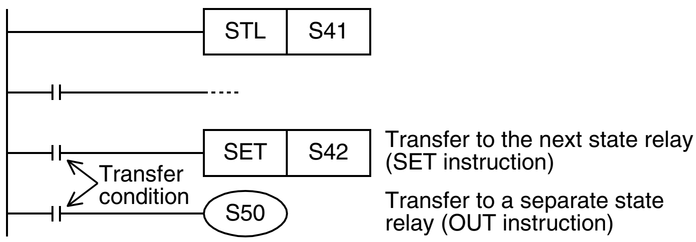
Output driving method

It is required to include a LD or LDI instruction before the last OUT instruction in a state relay.
Change such a circuit as shown below.



State relay transfer method

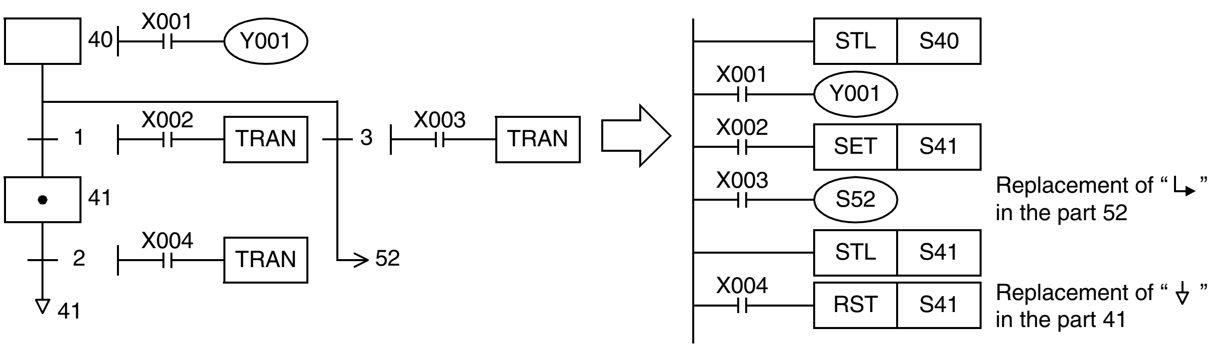
Each OUT and SET instructions in state relays automatically resets the transfer source, and has the self-holding function.
OUT instructions can be used only for transfer to a separate state relay in an SFC program.



Replacing " ↳" and "▽"

Replace the symbol " ↳" used in SFC programs to express repeat, jump or transfer to a state relay in another separate flow with the OUT instruction.
Replace the symbol "▽" (used to express reset of a state relay) with the RST instruction.

<SFC program> <Step ladder>

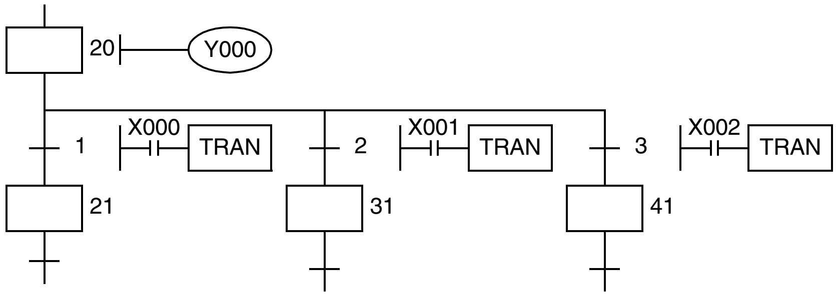


34.2.6 Program with state relays in branches and recombination

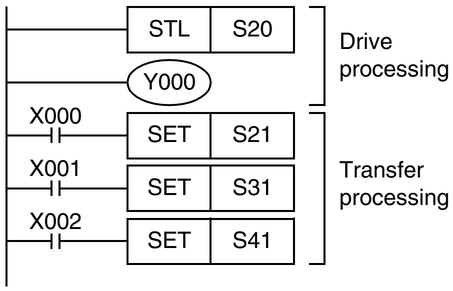
1. Example of selective branch

Do not use MPS, MRD, MPP, ANB and ORB instructions in a transfer processing program with branches and recombination.
Even in a load driving circuit, MPS instructions cannot be used immediately after STL instructions.
In the same way as programs for general state relays, program the drive processing first, and then program the transfer processing.
Continuously program all transfer processing.

<SFC program>



<Step ladder>



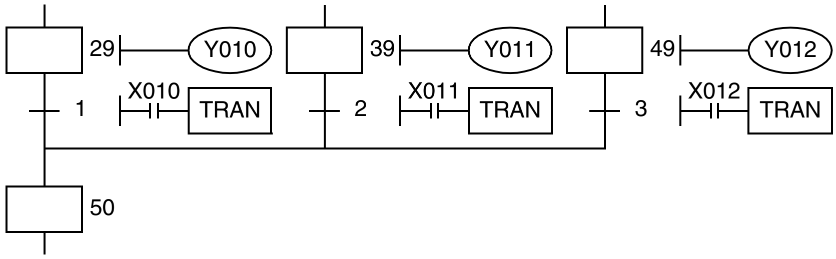
<List program>

```
STL   S20
OUT   Y000
LD    X000
SET   S21
LD    X001
SET   S31
LD    X002
SET   S41
```

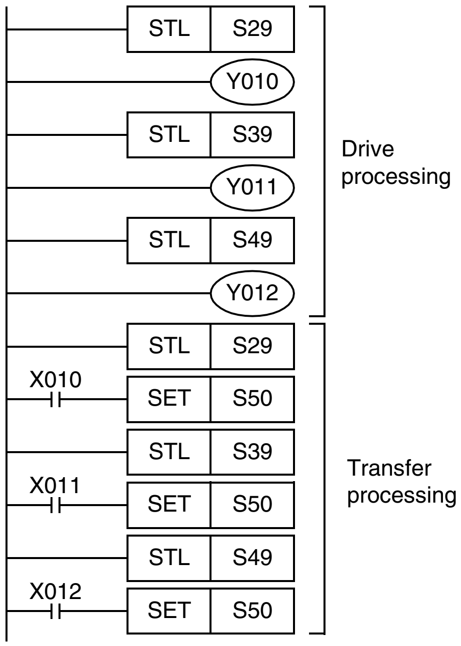
2. Example of selective recombination

Do not use MPS, MRD, MPP, ANB and ORB instructions in a transfer processing program with branches and recombination.
Even in a load driving circuit, MPS instructions cannot be used immediately after STL instructions.
Pay attention to the programming order so that a branch line does not cross a recombination line.

<SFC program>



<Step ladder>



<List program>

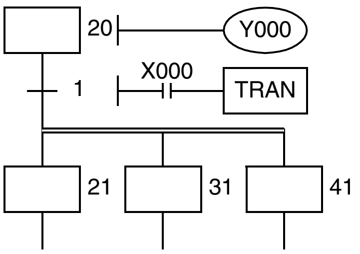
```
STL  S29
OUT  Y010
STL  S39
OUT  Y011
STL  S49
OUT  Y012
STL  S29
LD   X010
SET  S50
STL  S39
LD   X011
SET  S50
STL  S49
LD   X012
SET  S50
```

Before recombination, first program the drive processing of state relays.
After that, program only the transfer processing to recombination state relays.
This rule should be observed to enable inverse conversion into an SFC program.

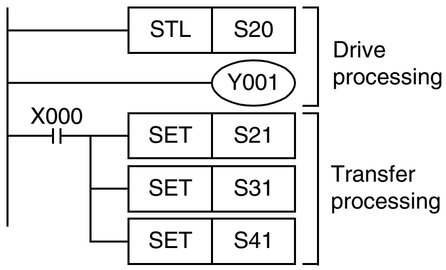
3. Example of parallel branch

Do not use MPS, MRD, MPP, ANB and ORB instructions in a program with branches and recombination.
Even in a load driving circuit, MPS instructions cannot be used immediately after STL instructions.
In the same way as programs for general state relays, program the drive processing first, and then program the transfer processing.
Continuously program all transfer processing.

<SFC program>



<Step ladder>



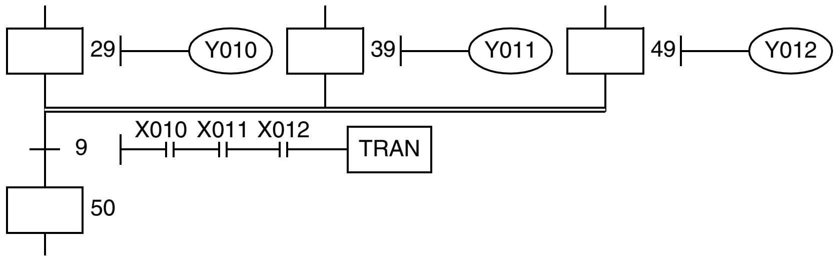
<List program>

```
STL  S20
OUT  Y000
LD   X000
SET  S21
SET  S31
SET  S41
```

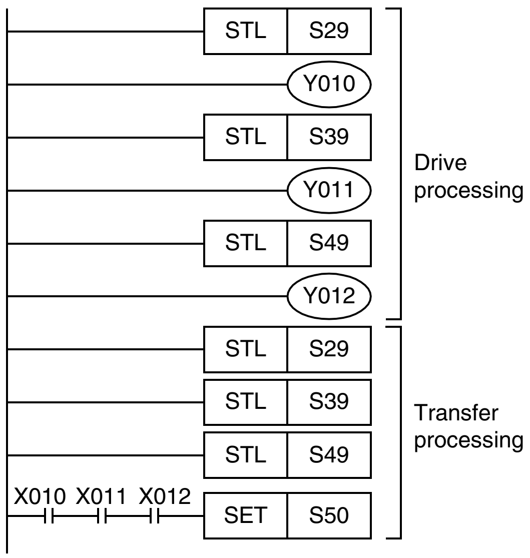
4. Example of parallel recombination

Do not use MPS, MRD, MPP, ANB and ORB instructions in a program with branches and recombination.
Even in a load driving circuit, MPS instructions cannot be used immediately after STL instructions.
Pay attention to the programming order so that a branch line does not cross a recombination line.

<SFC program>



<Step ladder>



<List program>

```
STL  S29
OUT  Y010
STL  S39
OUT  Y011
STL  S49
OUT  Y012
STL  S29
STL  S39
STL  S49
LD   X010
AND  X011
AND  X012
SET  S50
```

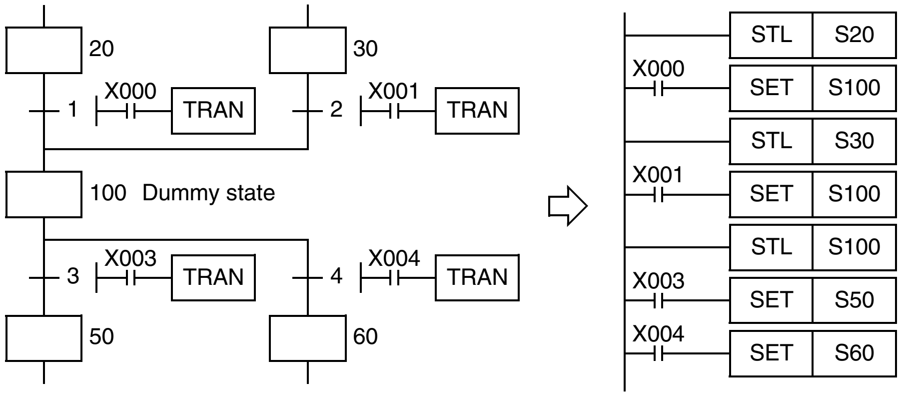
Before recombination, first program the drive processing of state relays.
After that, program only the transfer processing to recombination state relays.

5. Composition of branches and recombination

When a recombination line is directly connected to a branch line (not by way of a state relay as shown below), it is recommended to provide a dummy state relay between the lines.
Create step ladder programs as shown below.

1) Selective recombination and selective branch

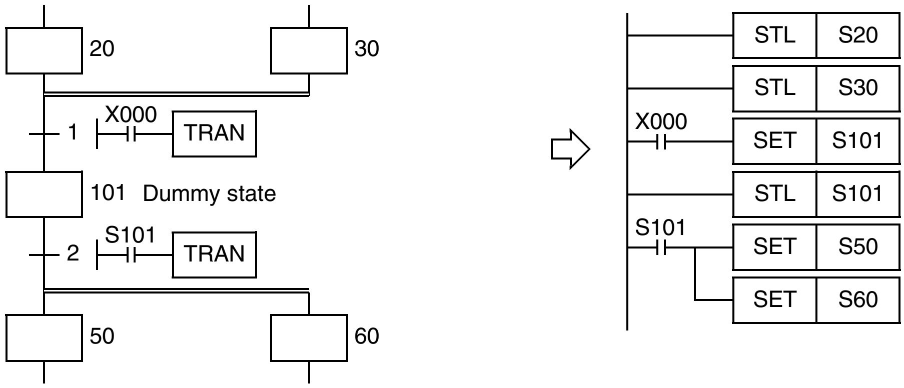
<SFC program> <Step ladder> <List program>




```
STL  S20
LD   X000
SET  S100
STL  S30
LD   X001
SET  S100
STL  S100
LD   X003
SET  S50
LD   X004
SET  S60
```

2) Parallel recombination and parallel branch

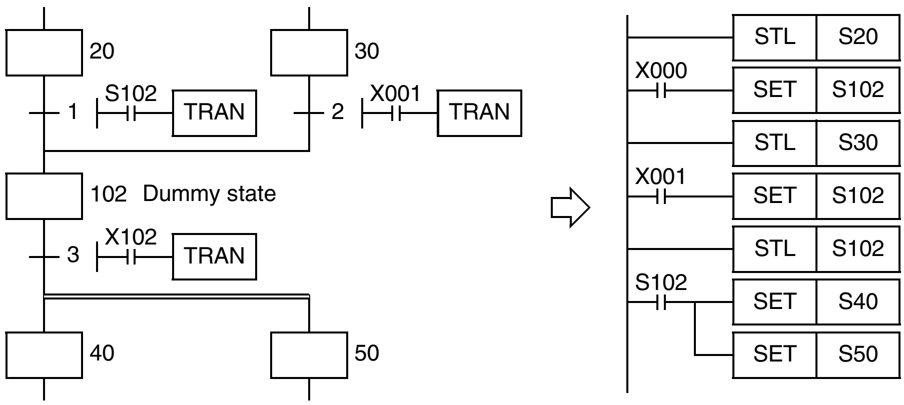
<SFC program> <Step ladder> <List program>




```
STL  S20
STL  S30
LD   X000
SET  S101
STL  S101
LD   S101
SET  S50
SET  S60
```

3) Selective recombination and parallel branch

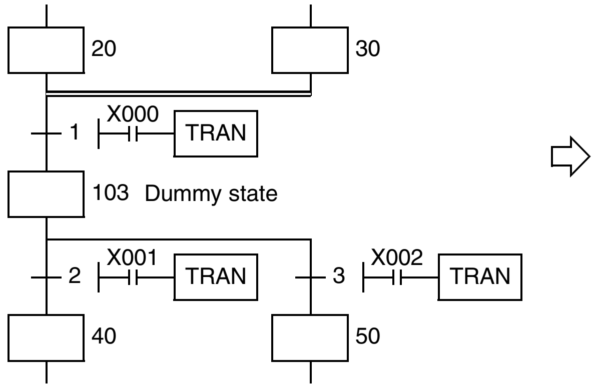
<SFC program> <Step ladder> <List program>



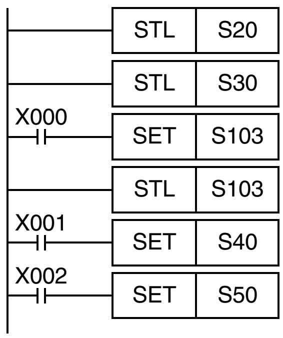

```
STL  S20
LD   X000
SET  S102
STL  S30
LD   X001
SET  S102
STL  S102
LD   S102
SET  S40
SET  S50
```

4) Parallel recombination and selective branch

<SFC program>



<Step ladder>



<List program>

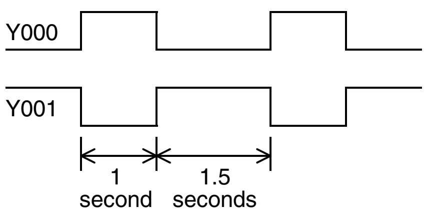
STL	S20
STL	S30
LD	X000
SET	S103
STL	S103
LD	X001
SET	S40
LD	X002
SET	S50

34.2.7 Program examples

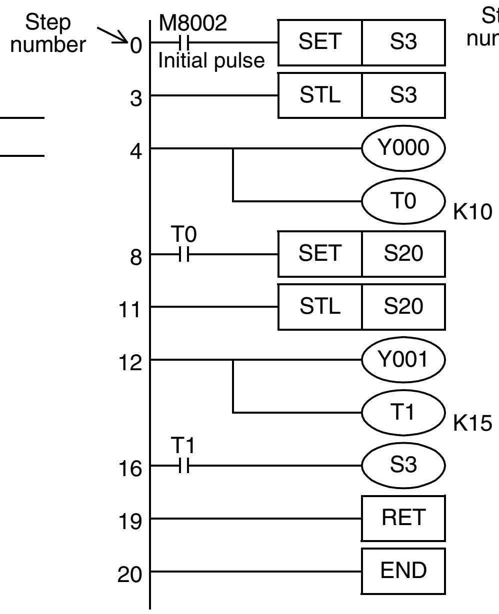
Examples of single flows

1. Example of flicker circuit

- When the PLC mode is changed from STOP to RUN, the state relay S3 is driven by the initial pulse (M8002).
- The state relay S3 outputs Y000. 1 second later, the ON status transfers to the state relay S20.
- The state relay S20 outputs Y001. 1.5 seconds later, the ON status returns to the state relay S3.



<Step ladder>



<List program>

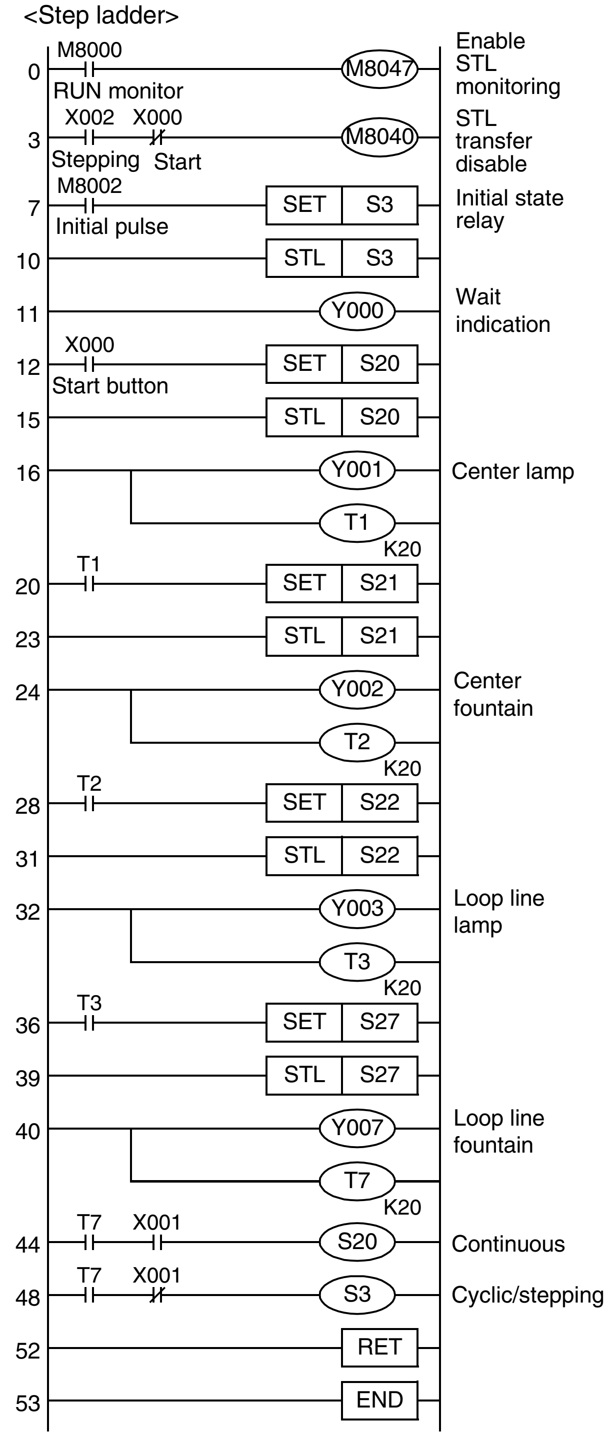
Step number		
0	LD	M8002
1	SET	S3
3	STL	S3
4	OUT	Y000
5	OUT	T0 K10
8	LD	T0
9	SET	S20
11	STL	S20
12	OUT	Y001
13	OUT	T1 K15
16	LD	T1
17	OUT	S3
19	RET	
20	END	

31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC•STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List

2. Example of fountain control

1) Cyclic operation (X001 = OFF, X002 = OFF)
   When the start button X000 is pressed, the outputs turn ON in the order "Y000 (wait indication) → Y001 (center lamp) → Y002 (center fountain) → Y003 (loop line lamp) → Y007 (loop line fountain) → Y000 (wait indication)", and then the outputs return to the wait status.
   Each output is switched in turn every 2 seconds by a timer.
2) Continuous operation (X001 = ON)
   Y001 to Y007 turn ON in turn repeatedly.
3) Stepping operation (X002 = ON)
   Every time the start button is pressed, each output turns ON in turn.

<Step ladder>



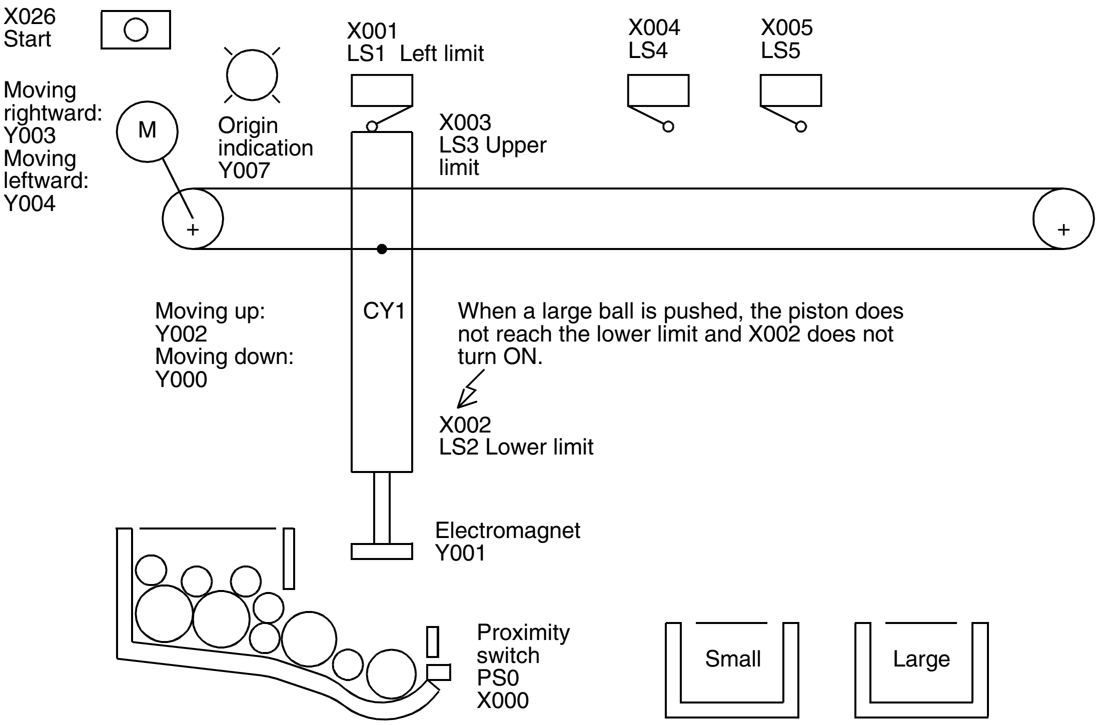
<List program>

0	LD	M8000
1	OUT	M8047
3	LD	X002
4	ANI	X000
5	OUT	M8040
7	LD	M8002
8	SET	S3
10	STL	S3
11	OUT	Y000
12	LD	X000
13	SET	S20
15	STL	S20
16	OUT	Y001
17	OUT	T1 K20
20	LD	T1
21	SET	S21
23	STL	S21
24	OUT	Y002
25	OUT	T2 K20
28	LD	T2
29	SET	S22
31	STL	S22
32	OUT	Y003
33	OUT	T3 K20
36	LD	T3
37	SET	S27
39	STL	S27
40	OUT	Y007
41	OUT	T7 K20
44	LD	T7
45	AND	X001
46	OUT	S20
48	LD	T7
49	ANI	X001
50	OUT	S3
52	RET	
53	END	

Examples of flows with selective branches and recombination

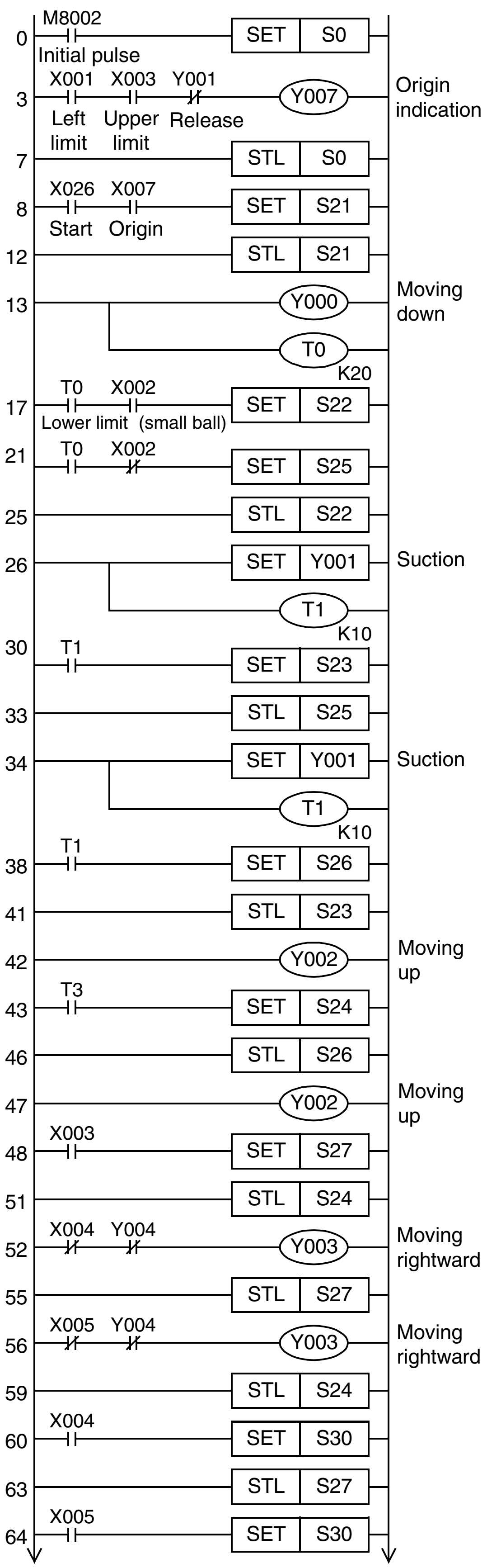
1. Example of selecting and carrying large and small balls

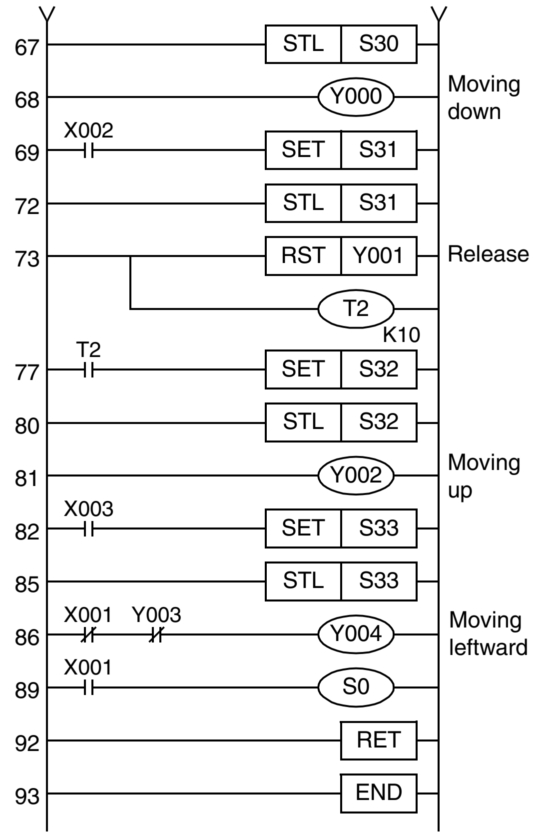
The figure below shows a mechanism which selects and carries large and small balls using conveyors.
The upper left position is regarded as the origin, and the mechanism performs in the order "moving down → suction → moving up → moving rightward → moving down → release → moving up → moving leftward."
When the arm moves down and the electromagnet pushes a large ball, the lower limit switch LS2 turns OFF.
When the electromagnet pushes a small ball, LS2 turns ON.



The figure below shows a step ladder program for selecting the ball size and judging balls as accepted or rejected.

<Step ladder>





- The ON status is transferred to the recombination state relay S30 when X004 turns ON in the case of small ball, or when X005 turns ON in the case of large ball.
- When the special auxiliary relay M8040 is driven, all transfers to any state relay are disabled.
  In each of the state relays S24, S27 and S33, an interlock contact is connected in series respectively to the rightward movement output Y003 and leftward movement output Y004.

<List program>

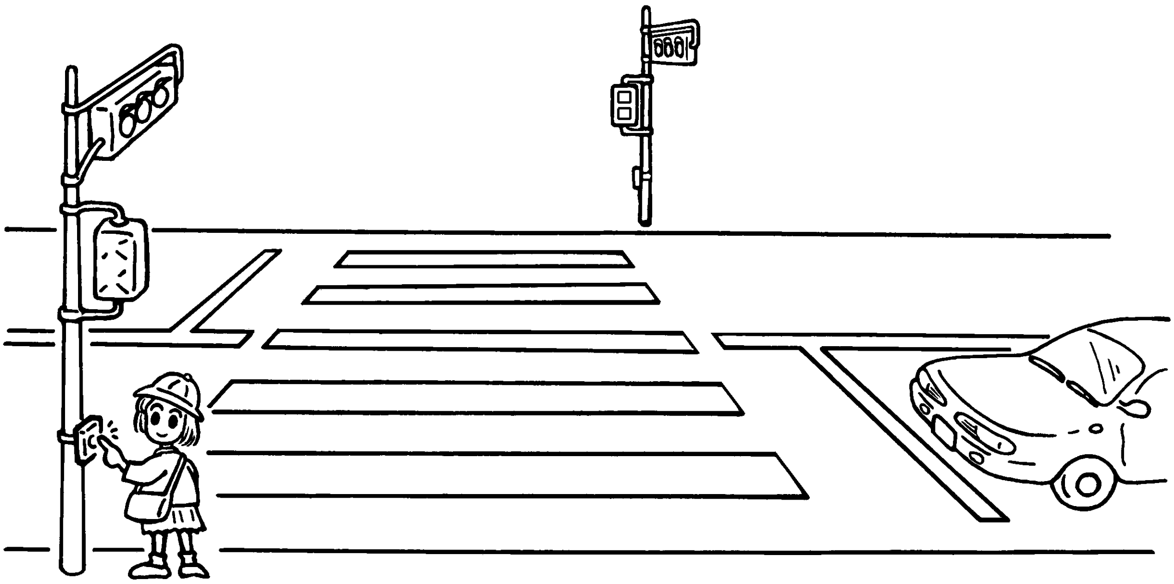
0	LD	M8002	34	SET	Y001	65	SET	S30
1	SET	S0	35	OUT	T1 K10	67	STL	S30
3	LD	X001	38	LD	T1	68	OUT	Y000
4	AND	X003	39	SET	S26	69	LD	X002
5	ANI	Y001	41	STL	S233	70	SET	S31
6	OUT	Y007	42	OUT	Y002	72	STL	S31
7	STL	S0	43	LD	T3	73	RST	Y001
8	LD	X026	44	SET	S24	74	OUT	T2 K10
9	AND	Y007	46	STL	S26	77	LD	T2
10	SET	S21	47	OUT	Y002	78	SET	S32
12	STL	S21	48	LD	X003	80	STL	S32
13	OUT	Y000	49	SET	S27	81	OUT	Y002
14	OUT	T0 K20	51	STL	S24	82	LD	X003
17	LD	T0	52	LDI	X004	83	SET	S33
18	AND	X002	53	ANI	Y004	85	STL	S33
19	SET	S22	54	OUT	Y003	86	LDI	X001
21	LD	T0	55	STL	S27	87	ANI	Y003
22	ANI	X002	56	LDI	X005	88	OUT	Y004
23	SET	S25	57	ANI	Y004	89	LD	X001
25	STL	S22	58	OUT	Y003	90	OUT	S0
26	SET	Y001	59	STL	S24	92	RET	
27	OUT	T1 K10	60	LD	X004	93	END	
30	LD	T1	61	SET	S30			
31	SET	S23	63	STL	S27			
33	STL	S25	64	LD	X005			

Example of flows with parallel branches and recombination

When the parts A, B and C are processed in parallel and then assembled, flows having parallel branches and recombination are used.

1. Example of pushbutton type crosswalk

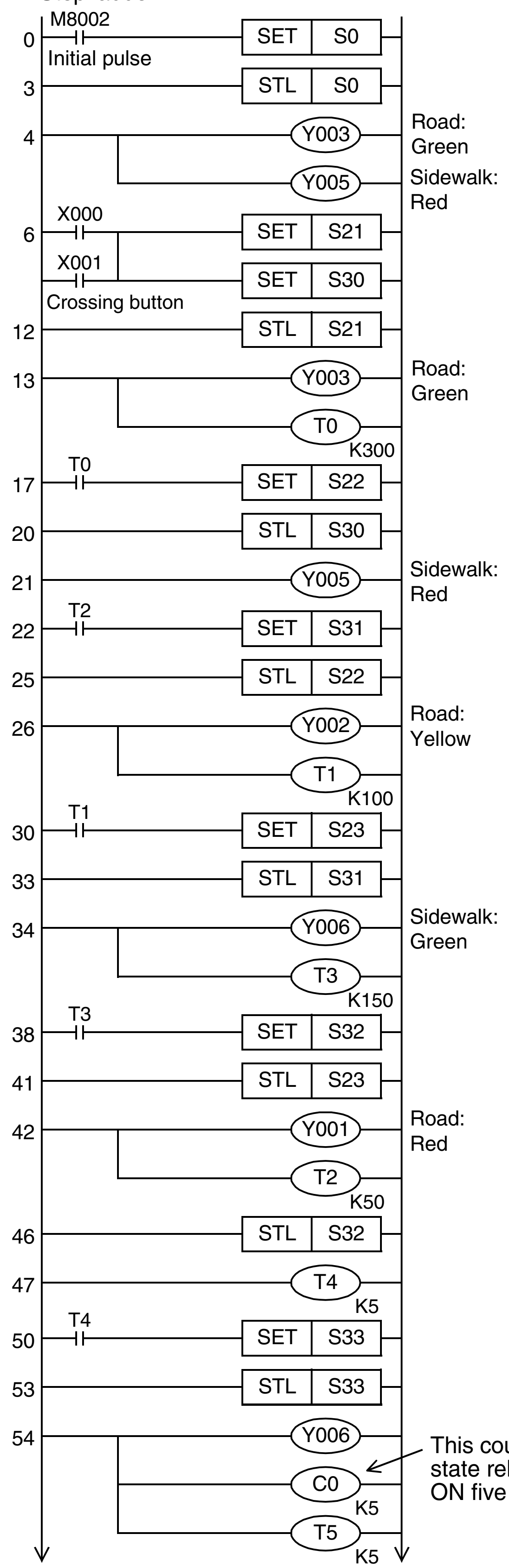
A pushbutton type crosswalk shown in the figure below can be expressed in flows having parallel branches and recombination.
Y003: Green Y002: Yellow Y001: Red



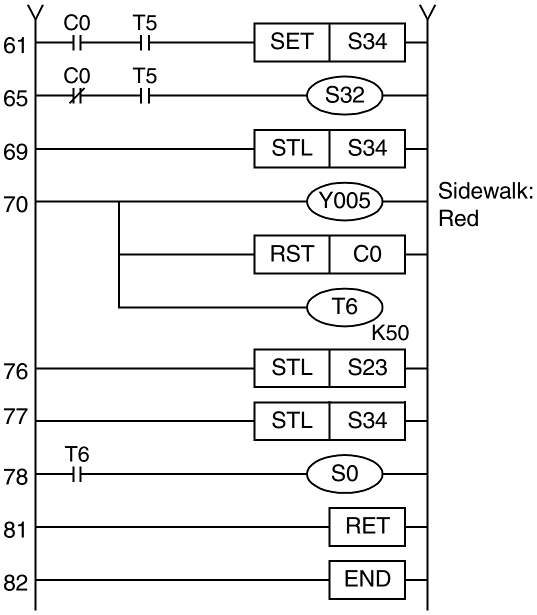
The SFC program for a pushbutton type crosswalk is as shown below. In this example, a partial flow (jump to a state relay located in an upper position) is repeated for blinking the green lamp on the crosswalk.

- When the PLC mode is changed from STOP to RUN, the initial state relay S0 turns ON. Normally, the green lamp is ON for the road and the red lamp is ON for the sidewalk.
- When the crossing button X000 or X001 is pressed, the state relay S21 specifies “road: green” and the state relay S30 specifies “sidewalk: red”. The signal lamp status is not changed.
- Thirty seconds later, the yellow lamp turns ON for the road. Ten seconds later after that, the red lamp turns ON for the road.
- When the timer T2 (5 seconds) reaches timeout after that, the green lamp turns ON for the sidewalk.
- Fifteen seconds later, the green lamp starts to blink for the sidewalk. (S32 turns OFF the green lamp, and S33 turns ON the green lamp.)
- While the green lamp is blinking, S32 and S33 turn ON and OFF repeatedly. When the counter C0 (set value: 5) turns ON, S34 turns ON. Five seconds after the red lamp turns ON for the sidewalk, the signal lamps return to the initial state.
- Even if the crossing button X000 or X001 is pressed in the middle of operation, the pressing is ignored.

<Step ladder>



This counter counts how many times the state relay S33 turns ON. When S33 turns ON five times, the contact C0 turns ON.



<List program>

0	LD	M8002	26	OUT	Y002	58	OUT	T5 K5
1	SET	S0	27	OUT	T1 K100	61	LD	C0
3	STL	S0	30	LD	T1	62	AND	T5
4	OUT	Y003	31	SET	S23	63	SET	S34
5	OUT	Y005	33	STL	S31	65	LDI	C0
6	LD	X000	34	OUT	Y006	66	AND	T5
7	OR	X001	35	OUT	T3 K150	67	OUT	S32
8	SET	S21	38	LD	T3	69	STL	S34
10	SET	S30	39	SET	S32	70	OUT	Y005
12	STL	S21	41	STL	S23	71	RST	C0
13	OUT	Y003	42	OUT	Y001	73	OUT	T6 K50
14	OUT	T0 K300	43	OUT	T2 K50	76	STL	S23
17	LD	T0	46	STL	S32	77	STL	S34
18	SET	S22	47	OUT	T4 K5	78	LD	T6
20	STL	S30	50	LD	T4	79	OUT	S0
21	OUT	Y005	51	SET	S33	81	RET	
22	LD	T2	53	STL	S33	82	END	
23	SET	S31	54	OUT	Y006			
25	STL	S22	55	OUT	C0 K5			

35. Interrupt Function and Pulse Catch Function

This chapter explains the built-in interrupt function and pulse catch function in FX PLCs.

35.1 Outline

This section explains the function to immediately execute an interrupt program (interrupt routine) without affecting the operation cycle of the sequence program (main) while using a interrupt function as a trigger. The delay by operation cycle and machine operation affected by uneven time intervals in normal sequence program process can be improved.

1. **Input interrupt function (interrupt of external signal input (X))**
   By the input signal from an input (X000 to X005), the main sequence program is paused, and an interrupt routine program is executed with priority.
   The input interrupt execution timing can be specified on the rising edge or falling edge of the signal by the pointer number.
   → **For details, refer to Section 35.3.**

2. **Input interrupt delay function (interrupt of external signal input (X))**
   By the input signal from an input (X000 to X005), the main sequence program is paused, and an interrupt routine program is executed with priority after the delay time (set in units of 1 ms).
   The input interrupt execution timing can be specified on the rising edge or falling edge of the signal by the pointer number.
   → **For details, refer to Section 35.4.**

3. **Timer interrupt function (timer interrupt activated in a constant cycle)**
   The main sequence program is paused in a constant cycle of 10 to 99 ms, and an interrupt routine program is executed with priority.
   → **For details, refer to Section 35.5.**

4. **High speed counter interrupt function (interrupt function given at counting up)**
   When the current value of a high speed counter reaches a specified value, the main sequence program is paused and an interrupt routine program is executed with priority.
   → **For details, refer to Section 35.6.**

5. **Pulse catch function**
   When the input signal from an input (X000 to X007) turns ON from OFF, a special auxiliary relay M8170 to M8177 is set in the interrupt processing. By a relay M8170 to M8177 in a normal sequence program, a signal that remains ON longer than the receivable range with regular input processing can be easily received.
   When processing such a signal that turns ON and OFF several times in one operation cycle, however, use the input interrupt function.
   → **For details, refer to Section 35.7.**

35.2 Common Items

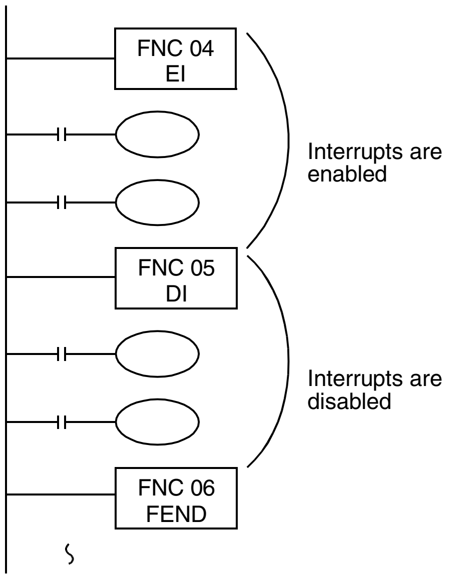
35.2.1 How to disable interrupt function and pulse catch function

This section describes how to disable the interrupt function and pulse catch function.

1. Limiting the program interrupt range [interrupt function and pulse catch function]

1) Programming method
   Program the FNC 05 (DI) instruction to set the interrupt disabled zone.
   Even if an interrupt is generated between the DI instruction and EI instruction (interrupt disabled zone), the interrupt is executed after the EI instruction.

2) Program example



3) Cautions
   a) The interrupt inputs with special auxiliary relay for interrupt disable (M8050 to M8059) turned ON are excluded.
      These special auxiliary relays are not valid for the pulse catch function.
   b) When the disabled zone is long, interrupts are accepted, but the interrupt processing is started after considerable time.
      When the interrupt disabling setting is not required, program only EI instruction. It is not always necessary to program DI instruction.

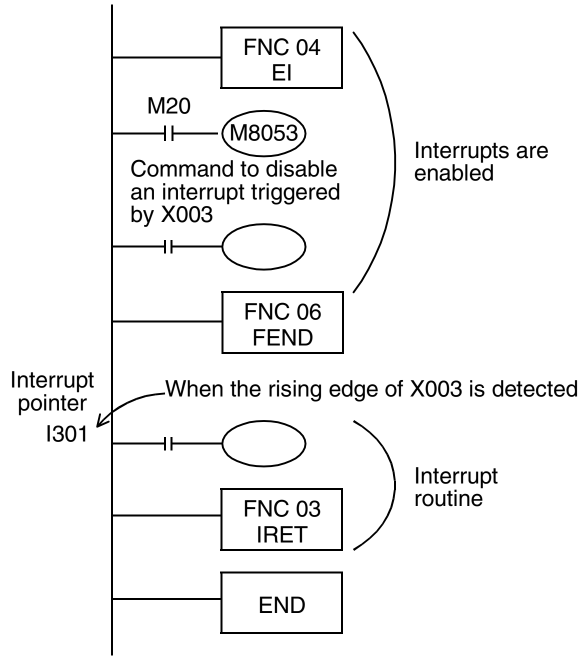
2. Disabling interrupt pointers (for each interrupt routine) [interrupt function]

1) Programming method
The special auxiliary relays M8050 to M8059 for disabling interrupt are provided.
While an interrupt disable flag (M8050 to M8059) is ON, a corresponding interrupt program is not executed even if the interrupt disable flag is set to OFF after a corresponding interrupt is generated.

Input interrupt	The input interrupts X000 to X005 correspond to M8050 to M8055[*1] respectively. When a relay M8050 to M8055 turns ON, a corresponding input interrupt is disabled.
Timer interrupt	The timer interrupts I6□□ to I8□□ correspond to M8056 to M8058[*1] respectively. When a relay M8056 to M8058 turns ON, a corresponding timer interrupt is disabled.
High speed counter interrupt	When M8059[*1] turns ON, all of the high speed counter interrupts I010 to I060 are disabled.

*1. Cleared when the PLC mode is changed from RUN to STOP.

2) Program example
In the program example shown below, when M8053 is set to ON by M20, the interrupt input I301 triggered by X003 is disabled.



35.2.2 Related items

1. Using the I/O refresh function (REFF instruction)

When controlling an input relay or output relay in an interrupt program, the I/O refresh instruction REFF (FNC 50) can be used to acquire the latest input information and immediately output the operation result. As a result, high speed control is achieved without being affected by the operation cycle of the PLC.

2. Interrupt operation while FROM/TO instruction is executed

The interrupt operation is executed as follows depending on the ON/OFF status of the special auxiliary relay M8028.

1) While M8028 is OFF
While FROM/TO instructions are being executed, interrupts are automatically disabled. Input interrupts and timer interrupts are not executed.
Interrupts generated during this period are immediately executed when the execution of FROM/TO instructions are completed.
FROM/TO instruction can be used in an interrupt program when M8028 is OFF.

2) While M8028 is ON
When an interrupt is generated while a FROM/TO instruction is being executed, execution of the FROM/TO instruction is paused and the interrupt is immediately executed.
FROM/TO instructions cannot be used in an interrupt routine program when M8028 is ON.

35.2.3 Cautions on use (common)

This section explains common cautions on using the interrupt function or pulse catch function.
Specific cautions on each interrupt function are explained in the description of each interrupt function.

1. Processing when many interrupts are generated

When many interrupts are generated in turn, priority is given to the first one. When many interrupts are generated at the same time, priority is given to the one having the smallest pointer number.
While an interrupt routine is being executed, other interrupts are disabled.

2. When double interrupt (interrupt during another interrupt) is required [interrupt function]

Usually, interrupts are disabled in an interrupt routine (program).
When EI (FNC 04) and DI (FNC 05) instructions are programmed in an interrupt routine, up to two interrupts can be accepted.

3. Operation when a timer is used [interrupt function]

More sure that counting using a general timer is disabled, even a 1ms retentive type timer.
In an interrupt routine, use timers for routine program T192 to T199.

4. Non-overlap of input [input interrupt (with/without delay function) and pulse catch function]

The inputs X000 to X007 can be used for high speed counters, input interrupts, pulse catch, SPD, ZRN, DSZR and DVIT instructions and for general-purpose inputs.
Make sure inputs do not overlap with each other.

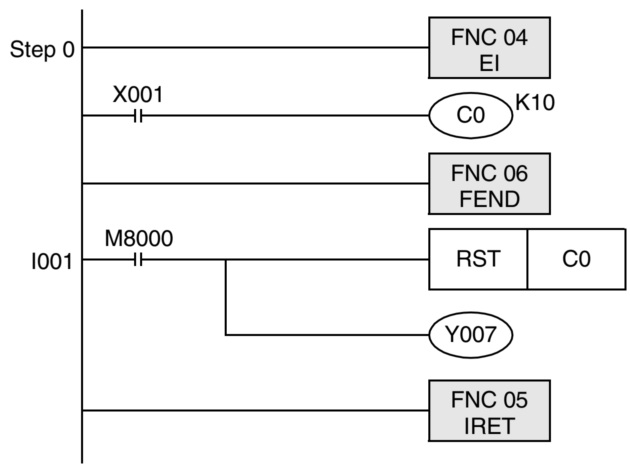
5. Operation of devices latched in the ON status [interrupt function]

Devices which were set to ON in an interrupt routine are held in the ON status even after the interrupt routine is finished.
When RST instruction for a timer or counter is executed, the reset status of the timer or counter is also held.
To turn OFF a device held in the ON status or for canceling such a timer or counter held in the reset status, reset such a device or deactivate RST instruction respectively inside or outsite the routine.

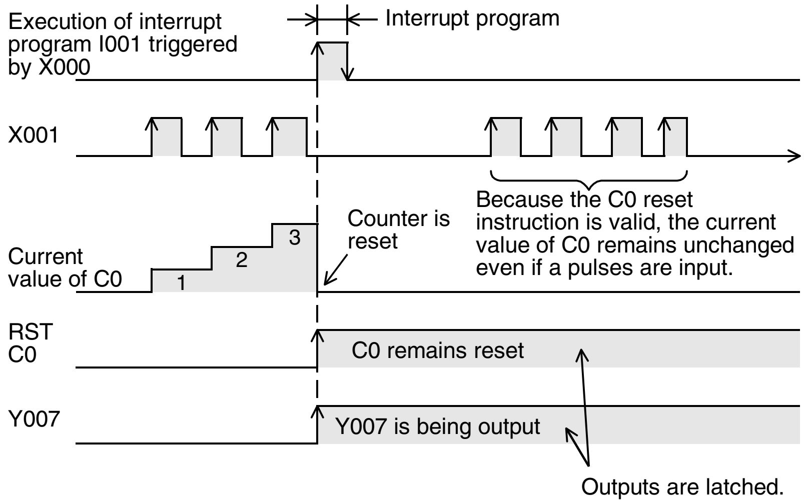
Example in which outputs are latched

In the program example shown below, the counter C0 is provided to count X001. When X001 turns ON from OFF, the interrupt program I001 is executed only in one scan, and then the counter C0 is reset and Y007 is output.

1) Program example

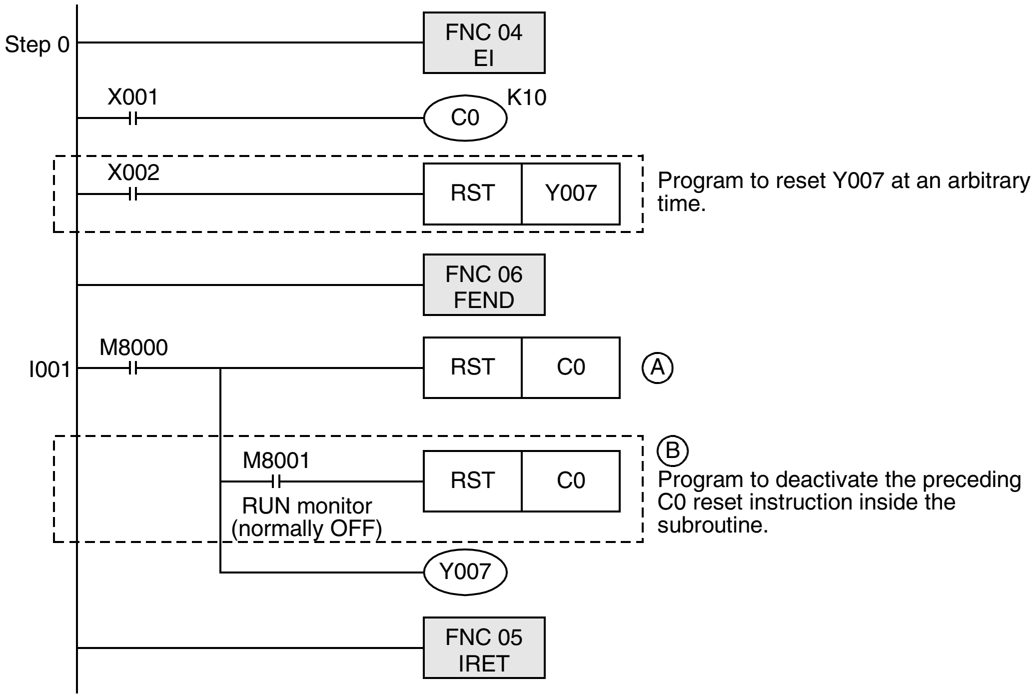


2) Timing chart

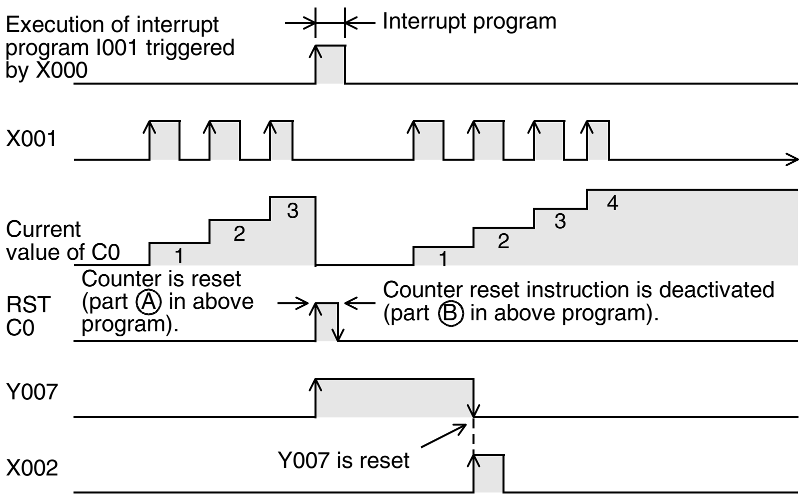


Example in which latched outputs are reset (countermeasures)

1) Program example



2) Timing chart



35.3 Input Interrupt (Interrupt Triggered by External Signal) [Without Delay Function]

35.3.1 Input interrupt (interrupt triggered by external signal) [without delay function]

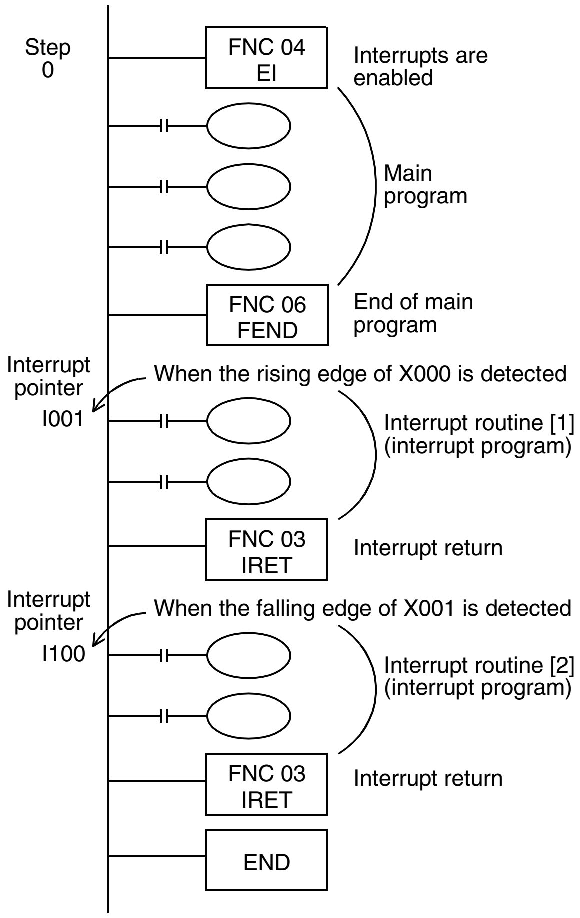
1. Outline

An interrupt routine is executed by the input signal from an input X000 to X005.

2. Application

Because the external input signal can be processed without being affected by the operation cycle of the PLC, this interrupt is suitable to high speed control and receiving of short pulses.

3. Basic program (programming procedure)



<u>Main program</u>

Interrupt inputs are accepted after EI instruction.

It is not necessary to program DI (disable interrupt) instruction if there is no zone where input interrupts should be disabled.

FEND instruction finishes the main program. Make sure to describe an interrupt routine after FEND instruction.

When X000 turns ON, its rising edge is detected, and the interrupt routine [1] is executed. IRET instruction returns the program execution to the main program.

When X001 turns OFF, its falling edge is detected, and the interrupt routine [2] is executed. IRET instruction returns the program execution to the main program.

"END" indicates the end of program.

4. Number and operation of (six) interrupt pointers

I □ 0□

- 0: Interrupt at falling edge, 1: Interrupt at rising edge
- 0 to 5 according to the inputs X000 to X005

Input number	Pointer number		Interrupt disable command
	Interrupt at rising edge	Interrupt at falling edge	
X000	I001	I000	M8050*1
X001	I101	I100	M8051*1
X002	I201	I200	M8052*1
X003	I301	I300	M8053*1
X004	I401	I400	M8054*1
X005	I501	I500	M8055*1

*1. Cleared when the PLC mode is changed from RUN to STOP.

5. How to disable each interrupt input

When either one among M8050 to M8055 is set to ON in a program, interrupts from the corresponding input number are disabled.
(Refer to the above table for the correspondence.)

6. Cautions

1) Do not use an input two or more times
Make sure that an input relay number used as an interrupt pointer is not used in high speed counters, pulse catch functions and applied instructions such as FNC 56 (speed detection) which use the same input range.

2) Automatic adjustment of the input filter
When an input interrupt pointer I□0□ is specified, the input filter of the input relay is automatically changed to the input filter for high speed receiving.
Accordingly, it is not necessary to change the filter value using REFF (FNC 51) instruction or special data register D8020 (input filter adjustment).
The input filter of an input relay not being used as an input interrupt pointer operates at 10 ms (initial value).

3) Pulse width of input interrupt
For executing input interrupt by an external signal, it is necessary to input the ON or OFF signal having the duration shown in the table below or more.

Input number	Input filter value when "0" is set
X000 to X005	5μs*1

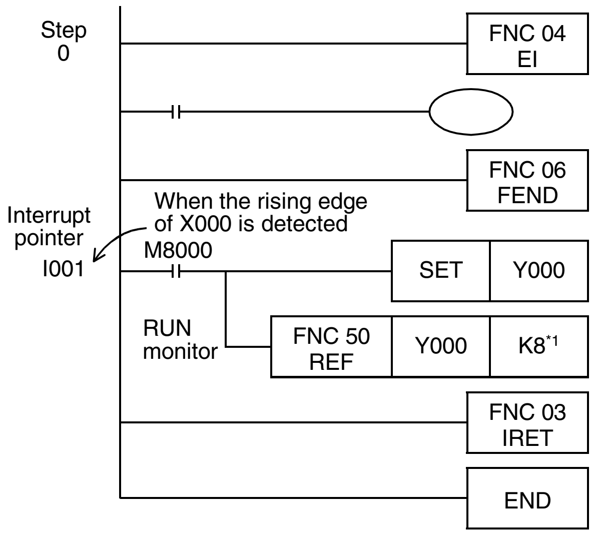
*1. When using the input filter at the filter value of 5 μs or when receiving a pulse whose response frequency is 50 k to 100 kHz using a high speed counter, perform the following:
-Make sure that the wiring length is 5 m or less.
-Connect a bleeder resistor of 1.5 kΩ (1 W or more) to an input terminal, and make sure that the load current of the open collector transistor output in the counterpart equipment is 20 mA or more including the input current in the main unit.

4) Using a pointer number two or more times
It is not possible to program an interrupt at the rising edge and an interrupt at the falling edge for an input such as I001 or I000.

31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC•STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List

7. Program examples

1) When using both an external input interrupt at the rising edge and the output refresh (REF instruction)
In the program example shown below, the output Y000 immediately turns ON when the rising edge of the external input X000 is detected.



Interrupts are enabled by EI instruction.
The main program is described.

The main program is finished by FEND instruction.

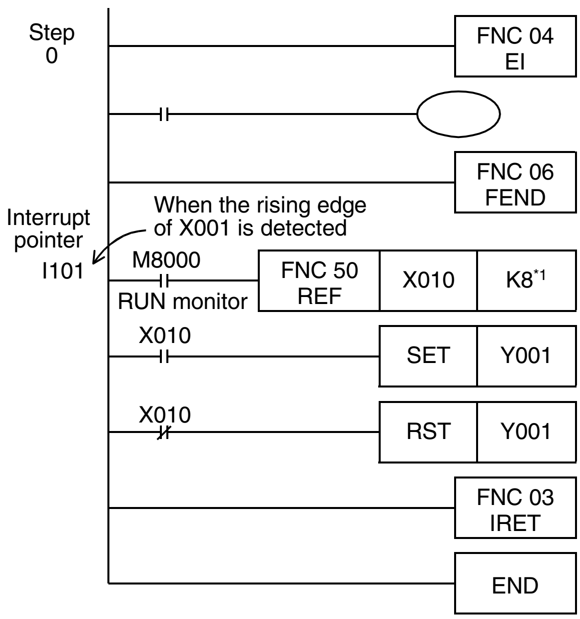
When an interrupt routine is executed by turning ON of X000, Y000 is set to ON unconditionally.

The outputs Y000 to Y007 are overwritten with the latest information by the output refresh instruction.
If the output refresh instruction is not provided, Y000 turns ON after END instruction after the program execution returned to the main routine.

If "SET Y000" is changed to "RST Y000", Y000 is immediately set to OFF by turning ON of X000.

*1. Make sure to specify a multiple of "8" for the number of inputs/outputs to be refreshed by REF (FNC 50) instruction.
If any value other than a multiple of "8" is specified, an operation error occurs and REF (FNC 50) instruction is not executed.

2) When using both an input interrupt and the input refresh (REF instruction)
In the program example shown below, an interrupt is processed using the latest input information.



Interrupts are enabled by EI instruction.
The main program is described.

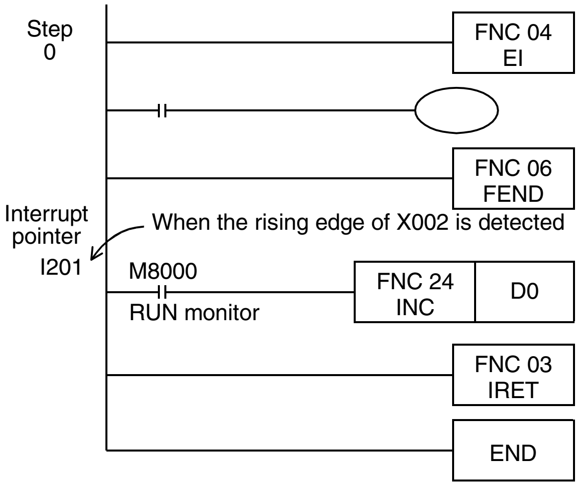
The main program is finished by FEND instruction.

When an interrupt routine is executed by turning X001 to ON, the input refresh is executed unconditionally, and the ON/OFF information of X010 to X017 at the current time is received.

Y001 is set to ON or OFF according to the ON/OFF status of X010.

*1. Make sure to specify a multiple of "8" as the number of inputs/outputs to be refreshed by REF (FNC 50) instruction.
If any value other than a multiple of "8" is specified, an operation error occurs and REF (FNC 50) instruction is not executed.

3) When counting the number of times of input generation (in the same way as 1-phase high speed counter)
In the program example shown below, external inputs are counted.

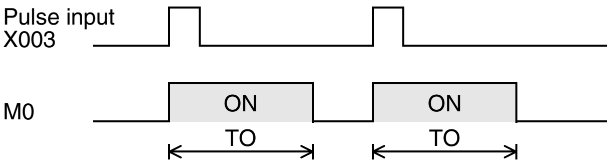


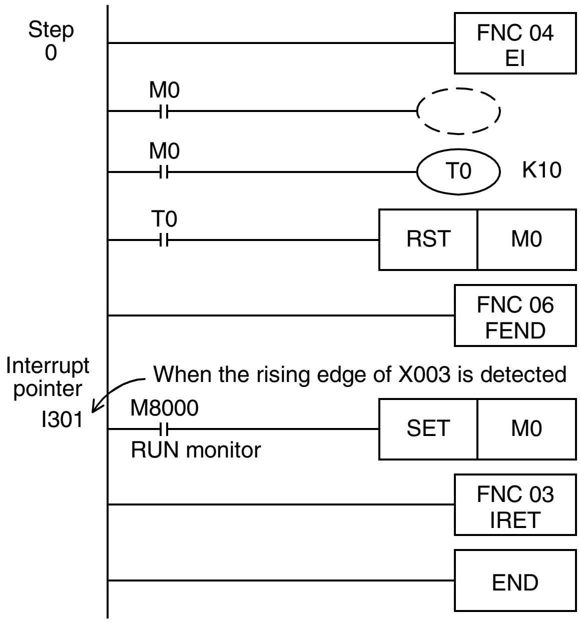
Interrupts are enabled by EI instruction.
The main program is described.

The main program is finished by FEND instruction.

When X002 turns ON, "1" is added to the value of D0.
INC instruction executes increment in every operation cycle, but the interrupt routine is executed only once by an input signal. Accordingly, it is not necessary to use INCP (pulse operation type) instruction.

4) When catching a short pulse
In the program example shown below, the ON status is held for a certain period of time after a short pulse turns ON.





Interrupts are enabled by EI instruction.

The ON/OFF signal in M0 is utilized in this program.

The period of time to hold M0 is specified.

After the timer time, M0 is reset.

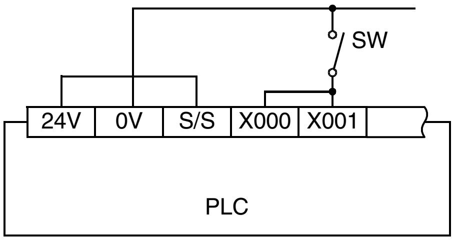
The main program is finished by FEND instruction.

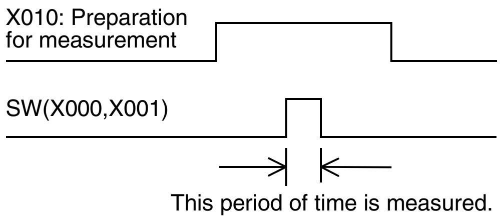
When X003 turns ON and the interrupt routine is executed, M0 is set to ON unconditionally.

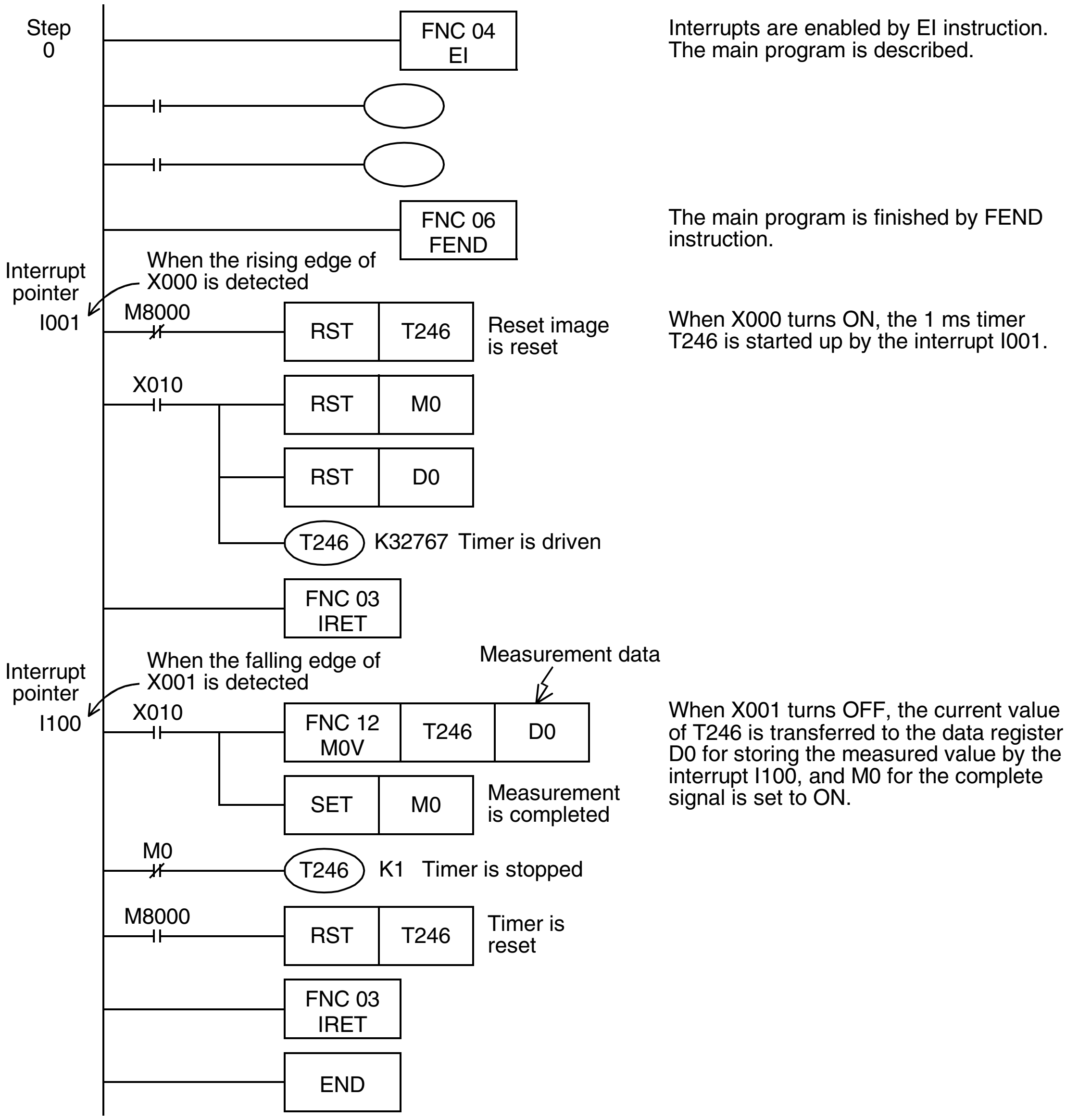
35.3.2 Examples of practical programs (programs to measure short pulse width)

1. Example of program to measure the short pulse width using input interrupt

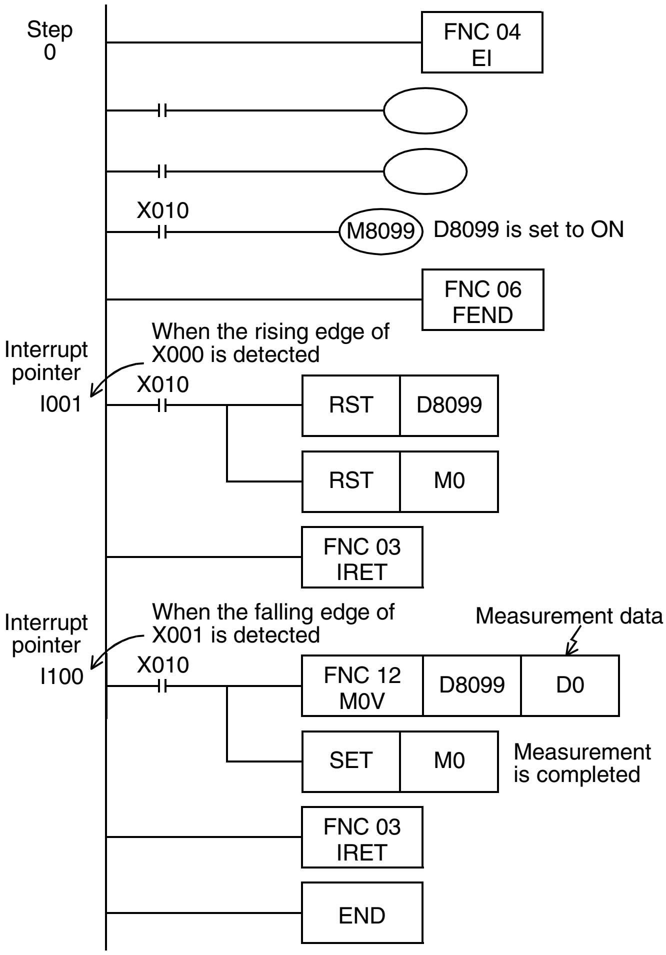
By using a 1 ms retentive type timer or the special data register D8099 (high speed ring counter), the short pulse width can be measured in units of 1 ms or 0.1 ms.
The figure below shows an example of FX3U PLC (sink input).







2. Example of program to measure the short pulse width using a high speed ring counter



Interrupts are enabled by EI instruction.
The main program is described.

The ring counter is set to ON.

When X000 turns ON: The ring counter is reset to OFF, and measurement is started.

When X001 turns OFF: The ring counter value is transferred to D0, and measurement is completed.

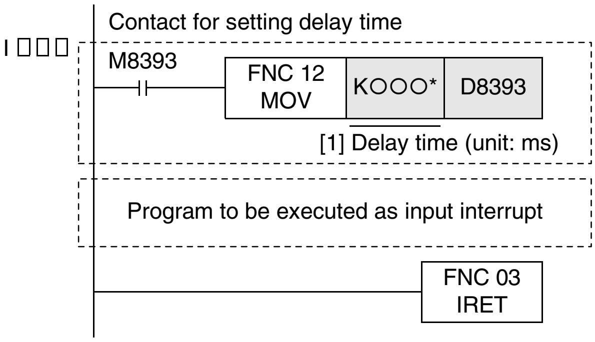
The special data register M8099 up-counts the 0.1 ms clock from the next operation cycle after being driven.
When the count value exceeds "32767", it is returned to "0".

35.4 Input interrupt (Interrupt by External Signal) [With Delay Function]

1. Outline

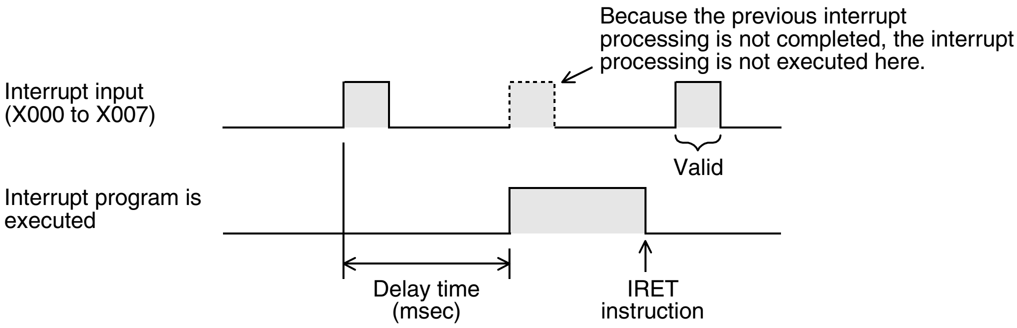
An input interrupt has the function to delay execution of an interrupt routine in units of 1 ms.
The delay time can be specified using the pattern program shown below.
By using the delay function, the mounting position of a sensor used for input interrupts can be adjusted electrically without changing the actual position.

2. Programming procedure



- Delay time specification program
  Make sure to describe the delay time specification program shown on the left at the head of an interrupt routine program.
  Because this program is a pattern program, change only the delay time ([1]).
  Only a constant (K) or data register (D) is available for specifying the time here*.
- End of interrupt program.

3. Timing chart



35.5 Timer Interrupt (Interrupt in Constant Cycle)

35.5.1 Timer interrupt (interrupt in constant cycle)

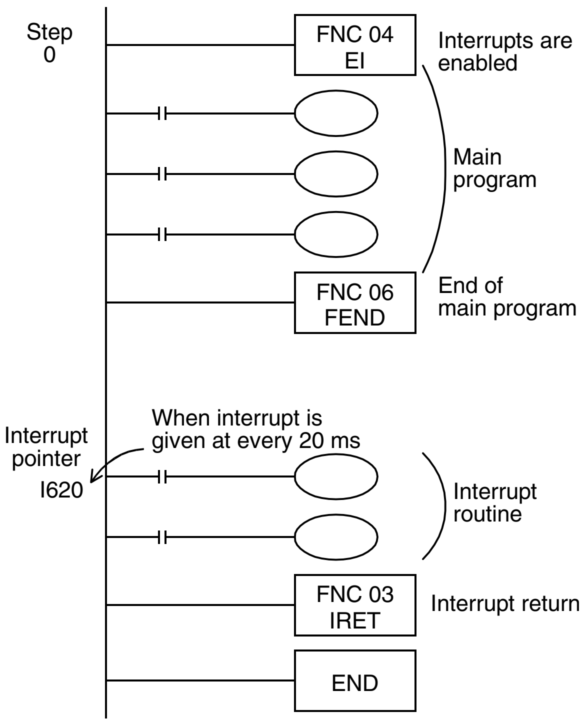
1. Outline

An interrupt routine is executed at every 1 to 99 ms without being affected by the operation cycle of a PLC.

2. Application

This type of interrupt is suitable when a certain program should be executed at high speed while the main program operation time is long or when a program should be executed at a constant time interval in sequence operations.

3. Basic program (programming procedure)



Main program

Timer interrupts are enabled after EI instruction.
It is not necessary to program DI (disable interrupt) if there is no zone where input interrupts should be disabled.

FEND instruction indicates the end of the main program.
Make sure to describe an interrupt routine after FEND instruction.

Interrupt subroutine

The interrupt routine is executed at every 20 ms.
Create a program to be executed as interrupt.
IRET instruction returns the program execution to the main program.

4. Number and operation of (three) timer interrupt pointers

I □ □ □
- Timer time: 1 to 99 (ms)
- Timer interrupt pointer (6, 7 or 8)

An interrupt routine program is executed at every specified interrupt cycle time (1 to 99 ms). Use the type of interrupt in control requiring cyclic interrupt processing regardless of the operating cycle of a PLC.

Input number	Interrupt cycle (ms)	Interrupt disable Flag
I6□□	An integer in the range from 1 to 99 is put in "□□" in the pointer name. Example: "I610" indicates a timer interrupt at every 10 ms.	M8056[*1]
I7□□		M8057[*1]
I8□□		M8058[*1]

*1. Cleared when the PLC mode is changed from RUN to STOP.

Caution

If the timer interrupt time is set to 9ms or less, the timer interrupt processing may not be executed in an accurate cycle in the following cases. Therefore, using a time that is over 10 ms is recommended.

- When the interrupt program processing time is long
- When the main program contains an applied instruction which processing time is long

31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC•STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List

5. Cautions

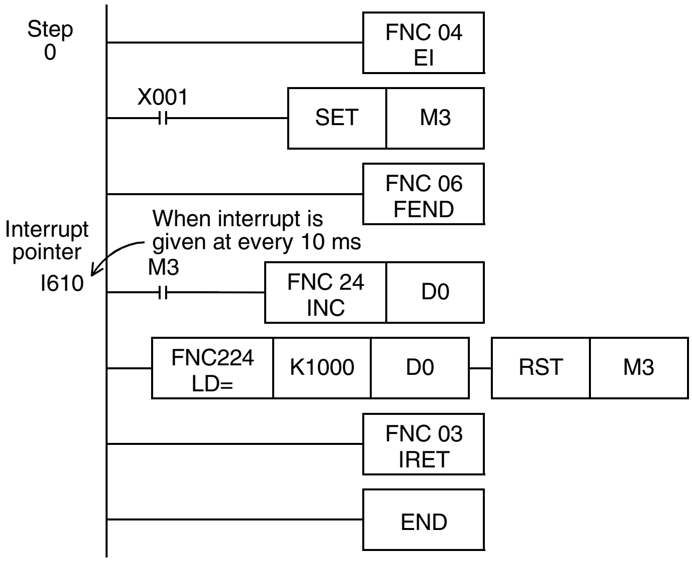
- Each pointer number (I6, I7 or I8) can be used only once.
- When M8056 to M8058 is set to ON in a program, a corresponding timer interrupt is disabled.

6. Program example

→ **For program examples in which RAMP (FNC 67) or HKY (FNC 71) instructions are combined, refer to Subsection 35.5.2.**

In the program example shown below, data is added and the addition result is compared with the set value every 10 ms.

1) Program example



Interrupts are enabled by EI instruction. The main program is described.

When M3 is set to ON, FNC 24 (INC) instruction becomes valid.

The main program is finished by FEND instruction.

"1" is added to the current value of D0 at every 10 ms.

When the current value of D0 reaches "1000", M3 is reset.

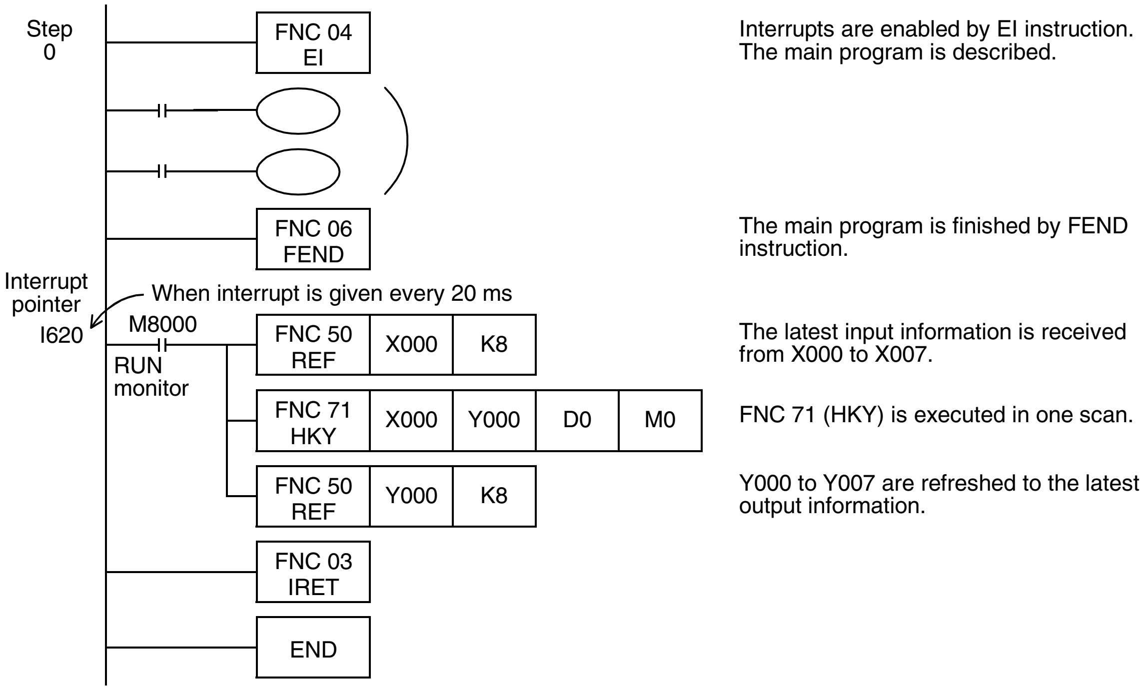
The current value of D0 is ramp data which changes from "0" to "1000" in 10 seconds.
In the program example using FNC 67 (RAMP) instruction shown later, the ramp data is made using a dedicated applied instruction.

35.5.2 Examples of practical program (timer interrupt programs using applied instruction)

RAMP (FNC 67), HKY (FNC 71), SEGL (FNC 74), ARWS (FNC 75) and PR (FNC 77) instructions execute a series of operations in synchronization with the scan time.
Because the total time may be too long or time fluctuation may cause a problem in these instructions, it is recommended to execute these instructions at a constant time interval using the timer interrupt function.
When not using the timer interrupt function, use the constant scan mode.

1. Timer interrupt processing of HKY (FNC 71) instruction

→ For HKY (FNC 71) instruction, refer to Section 15.2.

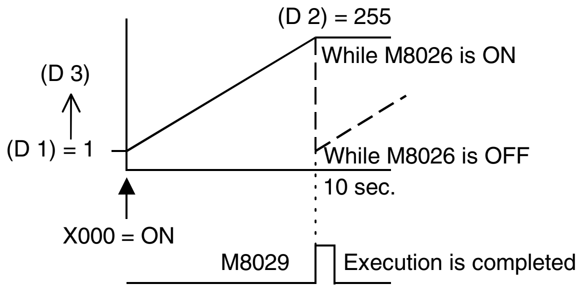


2. Timer interrupt processing of RAMP (FNC 67) instruction

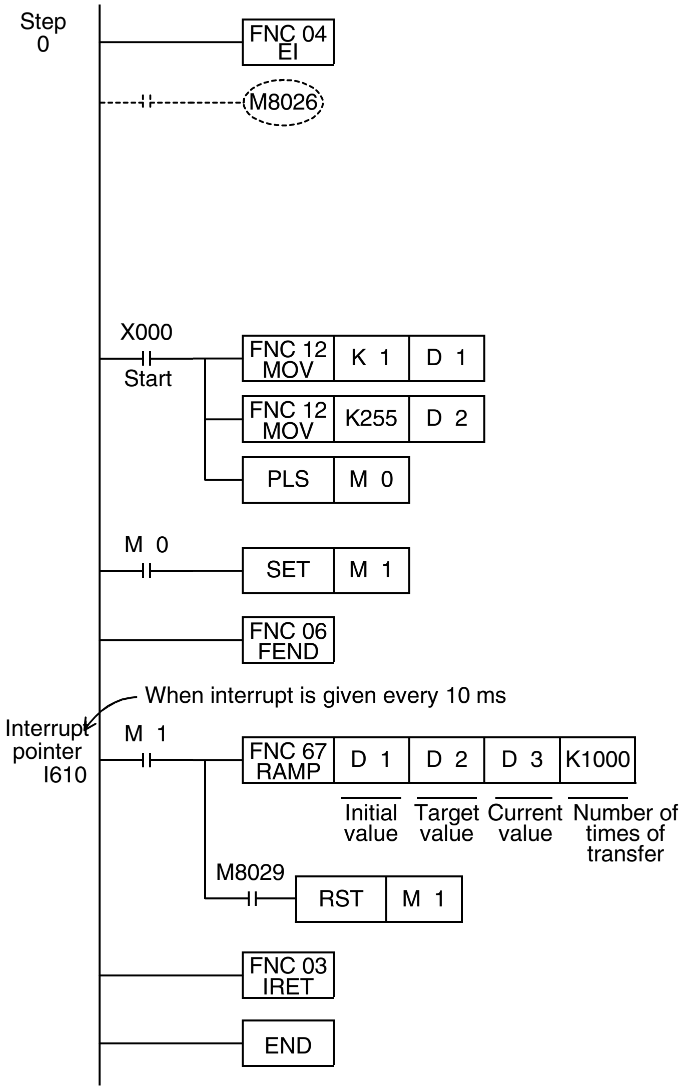
The ramp signal output circuit shown below is programmed using the timer interrupt function executed every 10 ms.

→ **For the use method of the instruction execution complete flag M8029, refer to Subsection 6.5.2.**
→ **For RAMP (FNC 67) instruction, refer to Section 14.8.**

1) Ramp output pattern
D4 is occupied as a register for counting the number of times of execution.



2) Program



Interrupts are enabled by EI instruction.
The main program is described.

With M8026 turned ON, when the value of (D3) reaches the final value (D2), the final value is latched.

As soon as the start command is given, the initial value (D1) and target value (D2) are transferred.

The main program is finished by FEND instruction.

While the instruction is executed 1000 times (in 10 seconds), the contents of D3 are changed from the value of D1 to the value of D2.

When the instruction execution complete flag M8029 turns ON, RAMP instruction drive input is set to OFF.
If FNC 67 (RAMP) instruction is continuously executed while M8026 is OFF, the value of D3 returns to the initial value (D1) immediately after it reaches the final value (D2), and then the same operation is repeated.
This program is not necessary when M8026 is ON.

35.6 Counter Interrupt - Interrupt Triggered by Counting Up of High Speed Counter

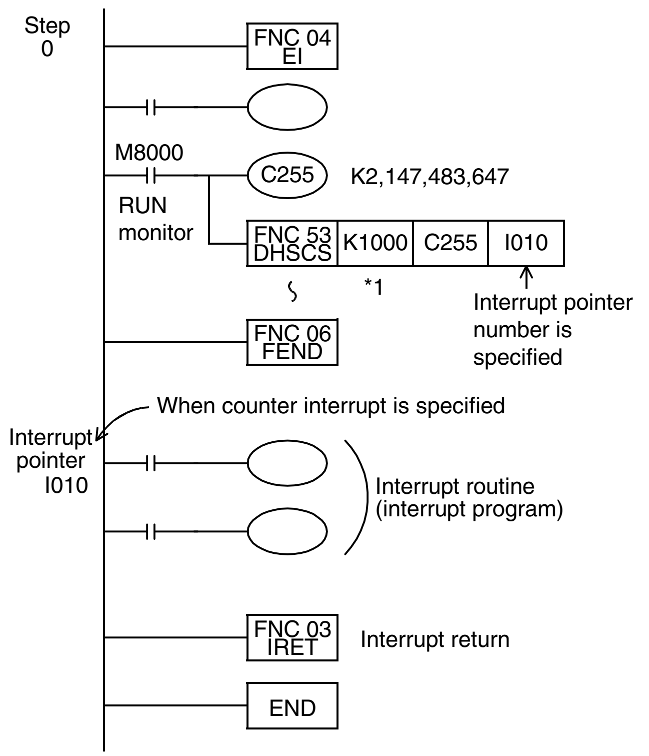
1. Outline

This type of interrupt utilizes the current value of a high speed counter.

2. Application

This type of interrupt is used together with the comparison set instruction DHSCS (FNC 53). When the current value of a high speed counter reaches the specified value, an interrupt routine is executed.

3. Basic program (programming procedure)



Main program
Interrupts are enabled after EI (FNC 04) instruction.
The main program is described.

The coil of a high speed counter is driven, and an interrupt pointer is specified in DHSCS (FNC 53) instruction.

When the current value of C255 changes from "999" to "1000", the interrupt routine is executed. For an interrupt routine use example, refer to the input interrupt function described in the preceding section.

*1. When the comparison value specified by a data register, etc. is changed, the current value is actually changed to the specified value when END instruction is executed.

4. Number and operation of (six) counter interrupt pointers

I 0 □ 0
Counter interrupt pointer (1 to 6)

Pointer No.	Interrupt disable flag
I010	M8059*1
I020	
I030	
I040	
I050	
I060	

*1. Cleared when the PLC mode is changed from RUN to STOP.

5. When setting an interrupt output (Y or M) to ON or OFF using a high speed counter

When only controlling the ON/OFF status of an output relay (Y) or auxiliary relay (M) according to the current value of a high speed counter, a required program can be easily created using DHSCS (FNC 53), DHSCR (FNC 54) or DHSZ (FNC 55) instruction.

6. Cautions

1) Pointer number
   Pointer numbers cannot overlap with each other.
2) Disabling interrupts
   When the special auxiliary relay M8059 is set to ON in a program, all counter interrupts are disabled.

35.7 Pulse Catch Function [M8170 to M8177]

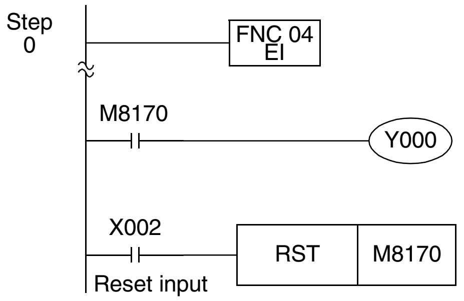
When the input relay X000 to X007 turns ON from OFF after the FNC 04 (EI) instruction is executed, the special auxiliary relay M8170 to M8177 is set for interrupt processing.

1. Assignment of input numbers and special auxiliary relays

Pulse catch input	Pulse catch relay
X000	M8170*1
X001	M8171*1
X002	M8172*1
X003	M8173*1
X004	M8174*1
X005	M8175*1
X006	M8176*1
X007	M8177*1

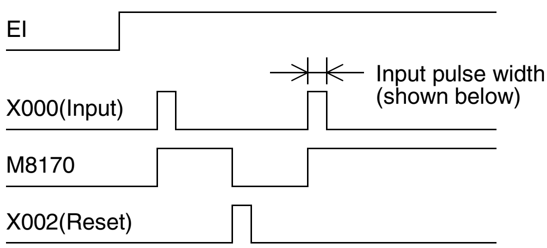
*1. Cleared when the PLC mode is changed from STOP to RUN.

2. Program example



When the rising edge of X000 is detected, M8170 is set as interrupt.

The pulse catch result is reset.



[X000 to X005]: 5 μs or more*1
[X006 and X007]: 50 μs or more

*1. When using the pulse catch function at 5 μs or when receiving a pulse whose response frequency is 50 k to 100 kHz using a high speed counter, perform the following:
   -Make sure that the wiring length is 5 m or less.
   -Connect a bleeder resistor of 1.5 kΩ (1 W or more) to the input terminal, and make sure that the load current of the open collector transistor output in the counterpart equipment is 20 mA or more including the input current in the main unit.

3. Cautions on use

1) When receiving an input again, it is necessary to reset the device which was once set using a program. Accordingly, until a device is reset, a new input cannot be received.
2) When it is necessary to receive continuous short pulses (input signals), use the external input interrupt function or high speed counter function.
3) A filter adjustment program is not required.
4) The pulse catch function is executed regardless of the operations of the special auxiliary relays M8050 to M8055 for respectively disabling interrupts.

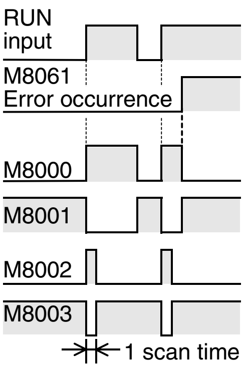
36. Operation of Special Devices (M8000 -, D8000 -)

36.1 Special Device List (M8000 -, D8000 -)

The device numbers and functions of the special auxiliary relays (indicated as "special M" in tables) and special data registers (indicated as "special D" in tables) are shown below.
Note that functions of certain devices vary depending on the series of the PLC.
Do not use the undefined / blank special auxiliary relays and special data registers in the sequence program since they are occupied by the CPU.
In addition, do not activate or write to the devices with brackets on the first letter such as [M]8000 or [D]8001 in the program.

→ **Refer to Section 36.2 for supplementary information on analog special adapter devices.**

36.1.1 Special Auxiliary Relay (M8000 to M8511)

Number and name	Operation and function	Applicable model FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
PLC Status									
[M]8000 RUN monitor NO contact	 →**Refer to Subsection 36.2.1.**	✓	✓	–	✓	✓	✓	✓	✓
[M]8001 RUN monitor NC contact		✓	✓	–	✓	✓	✓	✓	✓
[M]8002 Initial pulse NO contact		✓	✓	–	✓	✓	✓	✓	✓
[M]8003 Initial pulse NC contact		✓	✓	–	✓	✓	✓	✓	✓
[M]8004 Error occurrence	• FX3U, FX3UC ON when either M8060, M8061, M8064, M8065, M8066, or M8067 is ON. • FX1S,FX1N,FX2N,FX1NC,FX2NC ON when either M8060, M8061, M8063, M8064, M8065, M8066, or M8067 is ON.	✓	✓	D8004	✓	✓	✓	✓	✓
[M]8005 Battery voltage low	ON when battery voltage is below the value set in D8006. →**Refer to Subsection 36.2.3.**	✓	✓	D8005	–	–	✓	–	✓
[M]8006 Battery error latch	It is set when battery voltage low is detected. →**Refer to Subsection 36.2.3.**	✓	✓	D8006	–	–	✓	–	✓
[M]8007 Momentary power failure	ON for 1 scan, when momentary power failure is detected Even if M8007 turns ON, PLC continues to RUN as long as the duration of power loss is within the time period specified in D8008. →**Refer to Subsection 36.2.4.**	✓	✓	D8007 D8008	–	–	✓	–	✓
[M]8008 Power failure detected	Set when momentary power failure is detected. If power loss time is longer than the time period specified in D8008, M8008 is reset and PLC is switched to STOP mode.(M8000=OFF). →**Refer to Subsection 36.2.4.**	✓	✓	D8008	–	–	✓	–	✓
[M]8009 24V DC down	ON when 24V DC power fails in either powered extension unit or extension power supply unit*1.	✓	✓	D8009	–	–	✓	–	✓

*1. Powered extension units are applicable only to FX1N/FX2N/FX3U PLCs.
Extension power supply units are applicable only to FX3UC PLCs.

Number and name	Operation and function	Applicable model: FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Clock									
[M]8010	Not used	–	–	–	–	–	–	–	–
[M]8011 10 ms clock pulse	ON and OFF in 10 ms cycles (ON: 5 ms, OFF: 5 ms) →**Refer to Subsection 36.2.6.**	✓	✓	–	✓	✓	✓	✓	✓
[M]8012 100 ms clock pulse	ON and OFF in 100 ms cycles (ON: 50 ms, OFF: 50 ms) →**Refer to Subsection 36.2.6.**	✓	✓	–	✓	✓	✓	✓	✓
[M]8013 1 sec clock pulse	ON and OFF in 1 sec cycles (ON: 500 ms, OFF: 500 ms) →**Refer to Subsection 36.2.6.**	✓	✓	–	✓	✓	✓	✓	✓
[M]8014 1 min clock pulse	ON and OFF in 1 min cycles (ON: 30 sec, OFF: 30 sec) →**Refer to Subsection 36.2.6.**	✓	✓	–	✓	✓	✓	✓	✓
M 8015	Clock stop and preset For real time clock →**Refer to Subsection 36.2.7.**	✓	✓	–	✓	✓	✓	✓	✓*3
M 8016	Time read display is stopped For real time clock →**Refer to Subsection 36.2.7.**	✓	✓	–	✓	✓	✓	✓	✓*3
M 8017	±30 seconds correction For real time clock →**Refer to Subsection 36.2.7.**	✓	✓	–	✓	✓	✓	✓	✓*3
[M]8018	Installation detection (Always ON) For real time clock →**Refer to Subsection 36.2.7.**	✓	✓	–	✓ (Always ON)*3				
M 8019	Real time clock (RTC) error For real time clock →**Refer to Subsection 36.2.7.**	✓	✓	–	✓	✓	✓	✓	✓*3
Flag									
[M]8020 Zero	ON when the result of addition/subtraction is 0. →**Refer to Subsection 6.5.2 for usage.**	✓	✓	–	✓	✓	✓	✓	✓
[M]8021 Borrow	ON when the result of subtraction is less than the min. negative number. →**Refer to Subsection 6.5.2 for usage.**	✓	✓	–	✓	✓	✓	✓	✓
M 8022 Carry	ON when 'carry' occurs in result of addition or when an overflow occurs in result of shift operation. →**Refer to Subsection 6.5.2 for usage.**	✓	✓	–	✓	✓	✓	✓	✓
[M]8023	Not used	–	–	–	–	–	–	–	–
M 8024*1	BMOV direction specification (FNC 15)	✓	✓	–	–	✓	✓	–	✓
M 8025*2	HSC mode (FNC 53 to 55)	✓	✓	–	–	–	✓	–	✓
M 8026*2	RAMP mode (FNC 67)	✓	✓	–	–	–	✓	–	✓
M 8027*2	PR mode (FNC 77)	✓	✓	–	–	–	✓	–	✓
M 8028	100 ms/10 ms timer changeover	–	–	–	✓	–	–	–	–
	Interrupt permission during FROM/TO (FNC 78 and 79) instruction execution	✓	✓	–	–	–	✓	–	✓
[M]8029 Instruction execution complete	ON when operation such as DSW (FNC 72) is completed. →**Refer to Subsection 6.5.2 for usage.**	✓	✓	–	✓	✓	✓	✓	✓

*1. The operation varies according to PLC:
- Not cleared in an FX1N, FX2N, or FX2NC PLC.
- Cleared in an FX3U or FX3UC when PLC switches from RUN to STOP.

*2. The operation varies according to PLC:
- Not cleared in an FX2N or FX2NC PLC.
- Cleared in an FX3U or FX3UC when PLC switches from RUN to STOP.

*3. The FX2NC PLC required the optional memory board (with the real time clock).

Number and name	Operation and function	Applicable model: FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
PLC Mode									
M 8030*1 Battery LED OFF	When M8030 set to ON, LED on PLC is not lit even if battery voltage low is detected. →**Refer to Subsection 36.2.10.**	✓	✓	–	–	–	✓	–	✓
M 8031*1 Non-latch memory all clear	If this special auxiliary relay is activated, the ON/OFF image memory of Y, M, S, T, and C, and present values of T, C, D, special data registers*3, and R*2 are cleared to zero. However, file registers (D) in program memory and extension file registers (ER)*2 in the memory cassette are not cleared. →**Refer to Subsection 36.2.11.**	✓	✓	–	✓	✓	✓	✓	✓
M 8032*1 Latch memory all clear		✓	✓	–	✓	✓	✓	✓	✓
M 8033 Memory hold STOP	When PLC is switched from RUN to STOP, image memory and data memory are retained. →**Refer to Subsection 36.2.12.**	✓	✓	–	✓	✓	✓	✓	✓
M 8034*1 All outputs disable	All external output contacts of PLC are turned OFF. →**Refer to Subsection 36.2.13.**	✓	✓	–	✓	✓	✓	✓	✓
M 8035 Forced RUN mode	→**Refer to Subsection 36.2.14 for details.**	✓	✓	–	✓	✓	✓	✓	✓
M 8036 Forced RUN signal		✓	✓	–	✓	✓	✓	✓	✓
M 8037 Forced STOP signal		✓	✓	–	✓	✓	✓	✓	✓
[M]8038 Parameter setting	Communication parameter setting flag (for N:N network setting) →**Refer to Data Communication Edition.**	✓	✓	D8176 to D8180	✓	✓	✓*4	✓	✓
M 8039 Constant scan mode	When M8039 is ON, PLC waits until scan time specified in D8039 and then executes cyclic operation. →**Refer to Subsection 36.2.15.**	✓	✓	D8039	✓	✓	✓	✓	✓

*1. It is executed when END instruction is executed.

*2. R and ER are available only in FX3U/FX3UC PLCs.

*3. Special data registers are not cleared in FX1S/FX1N/FX2N/FX1NC/FX2NC PLCs.

*4. It is available in Ver. 2.00 or later.

Number and name	Operation and function	Applicable model							
		FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Step Ladder and Annunciator (Refer to ANS (FNC 46), ANR (FNC 47), IST (FNC 60), and Chapter 34 for details.)									
M 8040 Transfer disable	While M8040 is turned ON, transfer between states is disabled.	✓	✓	–	✓	✓	✓	✓	✓
[M]8041[*1] Transfer start	Transfer from initial state is enabled in automatic operation mode.	✓	✓	–	✓	✓	✓	✓	✓
[M]8042 Start pulse	Pulse output is given in response to a start input.	✓	✓	–	✓	✓	✓	✓	✓
M 8043[*1] Zero return complete	Set this in the last state of zero return mode.	✓	✓	–	✓	✓	✓	✓	✓
M 8044[*1] Zero point condition	Set this when machine zero return is detected.	✓	✓	–	✓	✓	✓	✓	✓
M 8045 All output reset disable	Disables the 'all output reset' function when the operation mode is changed.	✓	✓	–	✓	✓	✓	✓	✓
[M]8046[*2] STL state ON	ON when M8047 is ON and any state S0 to S899 or S1000 to S4095[*3] is active.	✓	✓	M8047	✓	✓	✓	✓	✓
M 8047[*2] STL monitoring enable	D8040 to D8047 are enabled when M8047 is ON.	✓	✓	D8040 to D8047	✓	✓	✓	✓	✓
[M]8048[*2] Annunciator operate	ON when M8049 is ON and any annunciator S900 to S999 is ON.	✓	✓	–	–	–	✓	–	✓
M 8049[*1] Annunciator enable	D8049 is enabled when M8049 is ON.	✓	✓	D8049 M8048	–	–	✓	–	✓

*1. Cleared when PLC switches from RUN to STOP.

*2. Executed at END instruction.

*3. S1000 to S4095 are available only in the FX3U/FX3UC Series PLC.

Number and name	Operation and function	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Interrupt Disable (Refer to Subsection 35.2.1.) for details.									
M8050 (input interrupt) I00□ disable[*4]	• If an input interrupt or timer interrupt occurs while a special auxiliary relay for that interrupt (M8050 - M8058) is ON, the interrupt will not operate. For example, turning M8050 ON disables the I00□ interrupt; hence, the interrupt routine is not processed even in an allowable program area. • If an input interrupt or timer interrupt occurs while a special auxiliary relay for that interrupt (M8050 - M8058) is OFF, a) The interrupt will be accepted. b) The interrupt routine will be processed promptly if it is permitted by the EI (FNC 04) instruction. However, if the DI (FNC 05) instruction disables interrupts, the interrupt program will not be processed until EI (FNC 04) permits the interrupts.	✓	✓	–	✓	✓	✓	✓	✓
M8051 (input interrupt) I10□ disable[*4]		✓	✓	–	✓	✓	✓	✓	✓
M8052 (input interrupt) I20□ disable[*4]		✓	✓	–	✓	✓	✓	✓	✓
M8053 (input interrupt) I30□ disable[*4]		✓	✓	–	✓	✓	✓	✓	✓
M8054 (input interrupt) I40□ disable[*4]		✓	✓	–	✓	✓	✓	✓	✓
M8055 (input interrupt) I50□ disable[*4]		✓	✓	–	✓	✓	✓	✓	✓
M8056 (Timer interrupt) I6□□ disable[*4]		✓	✓	–	–	–	✓	–	✓
M8057 (Timer interrupt) I7□□ disable[*4]		✓	✓	–	–	–	✓	–	✓
M8058 (Timer interrupt) I8□□ disable[*4]		✓	✓	–	–	–	✓	–	✓
M8059 Counter interrupt disable[*4]	Interrupt of I010 to I060 disabled	✓	✓	–	–	–	✓	–	✓

*4. Cleared when PLC switches from RUN to STOP.

Number and name	Operation and function	Applicable model							
		FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Error Detection (Refer to Chapter 37 for details.)									
[M]8060	I/O configuration error	✓	✓	D8060	–	–	✓	–	✓
[M]8061	PLC hardware error	✓	✓	D8061	✓	✓	✓	✓	✓
[M]8062	PLC/PP communication error	–	–	D8062	–	–	✓	–	✓
[M]8063*1*2	Serial communication error 1 [ch1]	✓	✓	D8063	✓	✓	✓	✓	✓
[M]8064	Parameter error	✓	✓	D8064	✓	✓	✓	✓	✓
[M]8065	Syntax error	✓	✓	D8065 D8069 D8314*5 D8315*5	✓	✓	✓	✓	✓
[M]8066	Ladder error	✓	✓	D8066 D8069 D8314*5 D8315*5	✓	✓	✓	✓	✓
[M]8067*3	Operation error	✓	✓	D8067 D8069 D8314*5 D8315*5	✓	✓	✓	✓	✓
M 8068	Operation error latch	✓	✓	D8068 D8312*5 D8313*5	✓	✓	✓	✓	✓
M 8069*4	I/O bus check	✓	✓	–	–	–	✓	–	✓

*1. The operation varies according to PLC:
- Cleared in an FX1S, FX1N, FX2N, FX1NC, or FX2NC when PLC switches from STOP to RUN.
- Not cleared in an FX3U or FX3UC PLC.

*2. Serial communication error 2 [ch2] in FX3U and FX3UC PLCs is detected by M8438.

*3. Cleared when PLC switches from STOP to RUN.

*4. When M8069 is ON, I/O bus check is executed. <Refer to Subsection 37.3.1 for details.>

*5. Only supported in FX3U/FX3UC

Number and name	Operation and function	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Parallel Link									
M 8070*6	Parallel link Set M8070 when using master station.	✓	✓	–	✓	✓	✓	✓	✓
M 8071*6	Parallel link Set M8071 when using slave station.	✓	✓	–	✓	✓	✓	✓	✓
[M]8072	Parallel link ON when operating	✓	✓	–	✓	✓	✓	✓	✓
[M]8073	Parallel link ON when M8070 or M8071 setting is incorrect	✓	✓	–	✓	✓	✓	✓	✓

*6. Cleared when PLC switches from STOP to RUN.

Number and name	Operation and function	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Sampling Trace									
[M]8074	Not used	–	–	–	–	–	–	–	–
[M]8075	Ready request for sampling trace	✓	✓	D8075 to D8098	–	–	✓	–	✓
[M]8076	Start request for sampling trace	✓	✓		–	–	✓	–	✓
[M]8077	ON during sampling trace	✓	✓		–	–	✓	–	✓
[M]8078	ON when sampling trace is completed	✓	✓		–	–	✓	–	✓
[M]8079	Sampling trace system area	✓	✓		–	–	✓	–	✓
[M]8080	Not used	–	–	–	–	–	–	–	–
[M]8081		–	–	–	–	–	–	–	–
[M]8082		–	–	–	–	–	–	–	–
[M]8083		–	–	–	–	–	–	–	–
[M]8084		–	–	–	–	–	–	–	–
[M]8085		–	–	–	–	–	–	–	–
[M]8086		–	–	–	–	–	–	–	–
[M]8087		–	–	–	–	–	–	–	–
[M]8088		–	–	–	–	–	–	–	–
[M]8089		–	–	–	–	–	–	–	–

Number and name	Operation and function	Applicable model							
		FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Flag									
[M]8090	BKCMP (FNC194 to FNC199) instructions - Block comparison signal	✓	✓*1	–	–	–	–	–	–
M 8091	COMRD (FNC182) and BINDA (FNC261) instructions - Output character quantity selector signal	✓	✓*1	–	–	–	–	–	–
[M]8092	Not used	–	–	–	–	–	–	–	–
[M]8093	Not used	–	–	–	–	–	–	–	–
[M]8094	Not used	–	–	–	–	–	–	–	–
[M]8095	Not used	–	–	–	–	–	–	–	–
[M]8096	Not used	–	–	–	–	–	–	–	–
[M]8097	Not used	–	–	–	–	–	–	–	–
[M]8098	Not used	–	–	–	–	–	–	–	–
High Speed Ring Counter									
M 8099*2	High speed ring counter operation (in units of 0.1ms, 16 bits)	✓	✓	D8099	–	–	✓	–	✓
[M]8100	Not used	–	–	–	–	–	–	–	–

*1. Available in Ver. 2.20 or later.

*2. For FX2N/FX2NC, 0.1ms high speed ring counter D8099 will operate after END instruction is executed with after M8099 is driven.
For FX3U/FX3UC, 0.1ms high speed ring counter D8099 will operate after M8099 is driven.

Number and name	Operation and function	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Memory Information									
[M]8101	Not used	–	–	–	–	–	–	–	–
[M]8102	Not used	–	–	–	–	–	–	–	–
[M]8103	Not used	–	–	–	–	–	–	–	–
[M]8104	ON when function extension memory is installed.	–	–	D8104 D8105	–	–	✓*3	–	✓*3
[M]8105	ON when writing to flash memory	✓	✓	–	–	–	–	–	–
[M]8106	Not used	–	–	–	–	–	–	–	–
[M]8107	Device comment registration check	✓	✓	D8107	–	–	–	–	–
[M]8108	Not used	–	–	–	–	–	–	–	–

*3. Available in Ver. 3.00 or later.

Number and name	Operation and function	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Output Refresh Error (Refer to Chapter 37 for details.)									
[M]8109	Output refresh error	✓	✓	D8109	–	–	✓	–	✓
[M]8110	Not used	–	–	–	–	–	–	–	–
[M]8111	Not used	–	–	–	–	–	–	–	–
Expansion Board [Dedicated to FX1S and FX1N]									
M 8112	FX1N-4EX-BD: BX0 input	–	–	–	✓	✓	–	–	–
	FX1N-2AD-BD: ch1 input mode change	–	–	D8112	✓	✓	–	–	–
	FX1N-1DA-BD: output mode change	–	–	D8114	✓	✓	–	–	–
M 8113	FX1N-4EX-BD: BX1 input	–	–	–	✓	✓	–	–	–
	FX1N-2AD-BD: ch2 input mode change	–	–	D8113	✓	✓	–	–	–
M 8114	FX1N-4EX-BD: BX2 input	–	–	–	✓	✓	–	–	–
M 8115	FX1N-4EX-BD: BX3 input	–	–	–	✓	✓	–	–	–
M 8116	FX1N-2EYT-BD: BY0 output	–	–	–	✓	✓	–	–	–
M 8117	FX1N-2EYT-BD: BY1 output	–	–	–	✓	✓	–	–	–
[M]8118	Not used	–	–	–	–	–	–	–	–
[M]8119	Not used	–	–	–	–	–	–	–	–

Number and name	Operation and function	Applicable model							
		FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
RS (FNC 80) and Computer Link [ch1] (Refer to Data Communication Edition for details.)									
[M]8120	Not used	–	–	–	–	–	–	–	–
[M]8121*1	RS (FNC 80) instruction: Send wait flag	✓	✓	–	✓	✓	✓	✓	✓
M 8122*1	RS (FNC 80) instruction: Send request	✓	✓	D8122	✓	✓	✓	✓	✓
M 8123*1	RS (FNC 80) instruction: Receive complete flag	✓	✓	D8123	✓	✓	✓	✓	✓
[M]8124	RS (FNC 80) instruction: Carrier detection flag	✓	✓	–	✓	✓	✓	✓	✓
[M]8125	Not used	–	–	–	–	–	–	–	–
[M]8126	Computer link [ch1] Global ON	✓	✓	D8127 D8128 D8129	✓	✓	✓	✓	✓
[M]8127	Computer link [ch1] On-demand send processing	✓	✓		✓	✓	✓	✓	✓
M 8128	Computer link [ch1] On-demand error flag	✓	✓		✓	✓	✓	✓	✓
M 8129	Computer link [ch1] On-demand Word / Byte changeover RS (FNC 80) instruction: Time-out check flag	✓	✓		✓	✓	✓	✓	✓

*1. Cleared when PLC switches from RUN to STOP or RS instruction is OFF.

Number and name	Operation and function	FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
High Speed Counter Comparison, High Speed Table, and Positioning [Positioning is supported in FX1S, FX1N, and FX1NC PLCs.]									
M 8130	HSZ (FNC 55)instruction: Table comparison mode	✓	✓	D8130	–	–	✓	–	✓
[M]8131	HSZ (FNC 55)instruction: Table comparison mode completion flag	✓	✓		–	–	✓	–	✓
M 8132	HSZ (FNC 55) and PLSY (FNC 57) instructions: Speed pattern mode	✓	✓	D8131 to D8134	–	–	✓	–	✓
[M]8133	HSZ (FNC 55) and PLSY (FNC 57) instructions: Speed pattern mode completion flag	✓	✓		–	–	✓	–	✓
[M]8134	Not used	–	–	–	–	–	–	–	–
[M]8135		–	–	–	–	–	–	–	–
[M]8136		–	–	–	–	–	–	–	–
[M]8137		–	–	–	–	–	–	–	–
[M]8138	HSCT (FNC280) instruction: Instruction execution complete flag	✓	✓	D8138	–	–	–	–	–
[M]8139	HSCS(FNC 53), HSCR(FNC 54), HS2(FNC 55), HSCT(FNC280) instructions: High speed counter comparison instruction executing	✓	✓	D8139	–	–	–	–	–
M 8140	ZRN (FNC156) instruction: CLR signal output function enable	–	–	–	✓	✓	–	✓	–
[M]8141	Not used	–	–	–	–	–	–	–	–
[M]8142		–	–	–	–	–	–	–	–
[M]8143		–	–	–	–	–	–	–	–
[M]8144		–	–	–	–	–	–	–	–
M 8145	[Y000] Pulse output stop command	–	–	–	✓	✓	–	✓	–
M 8146	[Y001] Pulse output stop command	–	–	–	✓	✓	–	✓	–
[M]8147	[Y000] Pulse output monitor (BUSY/READY)	–	–	–	✓	✓	–	✓	–
[M]8148	[Y001] Pulse output monitor (BUSY/READY)	–	–	–	✓	✓	–	✓	–
[M]8149	Not used	–	–	–	–	–	–	–	–

Number and name	Operation and function	Applicable model							
		FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Inverter Communication Function (Refer to Data Communication Edition for details.)									
[M]8150	Not used	–	–	–	–	–	–	–	–
[M]8151*1	Inverter communication in execution [ch1]	✓	✓	D8151	–	–	–	–	–
[M]8152*1	Inverter communication error [ch1]	✓	✓	D8152	–	–	–	–	–
[M]8153*1	Inverter communication error latch [ch1]	✓	✓	D8153	–	–	–	–	–
[M]8154*1	IVBWR (FNC274) instruction error [ch1]	✓	✓	D8154	–	–	–	–	–
[M]8154	Defined per use of EXTR (FNC180)	–	–	–	–	–	✓*2	–	✓*2
[M]8155	EXTR (FNC180) is using communicating port	–	–	D8155	–	–	✓*2	–	✓*2
[M]8156*1	Inverter communication in execution [ch2]	✓	✓	D8156					
[M]8156	Communication error or parameter error in EXTR (FNC180) instruction	–	–	D8156	–	–	✓*2	–	✓*2
[M]8157*1	Inverter communication error [ch2]	✓	✓	D8157					
	Communication error in EXTR (FNC180) instruction is latched.	–	–	D8157	–	–	✓*2	–	✓*2
[M]8158*1	Inverter communication error latch [ch2]	✓	✓	D8158	–	–	–	–	–
[M]8159*1	IVBWR (FNC274) instruction error [ch2]	✓	✓	D8159	–	–	–	–	–

*1. Cleared when PLC switches from STOP to RUN.

*2. Available in Ver. 3.00 or later.

Number and name	Operation and function	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Advanced Function									
M 8160*3	SWAP function of XCH (FNC 17)	✓	✓	–	–	–	✓	–	✓
M 8161*3*4	8-bit process mode	✓	✓	–	✓	✓	✓	✓	✓
M 8162	High speed parallel link mode	✓	✓	–	✓	✓	✓	✓	✓
[M]8163	Not used	–	–	–	–	–	–	–	–
M 8164*3	FROM (FNC 78), TO (FNC 79) instructions: Transfer points variable mode	–	–	D8164	–	–	✓*5	–	✓
M 8165*3	SORT2 (FNC149) instruction: Sorting in descending order	✓	✓*6	–	–	–	–	–	–
[M]8166	Not used	–	–	–	–	–	–	–	–
M 8167*3	HKY (FNC 71) instruction: HEX data handling function	✓	✓	–	–	–	✓	–	✓
M 8168*3	SMOV (FNC 13) instruction: HEX data handling function	✓	✓	–	–	–	✓	–	✓
[M]8169	Not used	–	–	–	–	–	–	–	–

*3. Cleared when PLC switches from RUN to STOP.

*4. Applicable to ASC (FNC 76), RS (FNC 80), ASCI (FNC 82), HEX (FNC 83), CCD (FNC 84), and CRC (FNC188) instructions*7.

*5. Available in Ver. 2.00 or later.

*6. Available in Ver. 2.20 or later.

*7. CRC (FNC188) instruction is available only in FX3U and FX3UC PLCs.

31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC•STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List

Number and name	Operation and function	Applicable model							
		FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Pulse Catch (Refer to Section 35.7 for details.)									
M 8170*1	Input X000 pulse catch	✓	✓	–	✓	✓	✓	✓	✓
M 8171*1	Input X001 pulse catch	✓	✓	–	✓	✓	✓	✓	✓
M 8172*1	Input X002 pulse catch	✓	✓	–	✓	✓	✓	✓	✓
M 8173*1	Input X003 pulse catch	✓	✓	–	✓	✓	✓	✓	✓
M 8174*1	Input X004 pulse catch	✓	✓	–	✓	✓	✓	✓	✓
M 8175*1	Input X005 pulse catch	✓	✓	–	✓	✓	✓	✓	✓
M 8176*1	Input X006 pulse catch	✓	✓	–	–	–	–	–	–
M 8177*1	Input X007 pulse catch	✓	✓	–	–	–	–	–	–

*1. Cleared when PLC switches from STOP to RUN.
FX2N, FX2NC, FX3U, or FX3UC Series PLC: EI (FNC 04) instruction is necessary.
FX1S, FX1N, or FX1NC Series PLC: EI (FNC 04) instruction is unnecessary.

Communication Port Channel Setting (Refer to Data Communication Edition for details.)									
M 8178	Parallel link channel switch (OFF: ch1/ON: ch2)	✓	✓	–	–	–	–	–	–
M 8179	N:N network channel switch*2	✓	✓	–	–	–	–	–	–

*2. The channel is specified by either setting or not setting M8179 in the setting program.
→ **For setting program, Refer to Data Communication Edition**
• ch1: M8179 is not set in program
• ch2: M8179 is set in program

N:N Network (Refer to FX Series User's Manual - Data Communication Edition for details.)									
[M]8180	Not used	–	–	–	–	–	–	–	–
[M]8181		–	–	–	–	–	–	–	–
[M]8182		–	–	–	–	–	–	–	–
[M]8183*3	Data communication error (Master station)	✓	✓	D8201 to D8218	(M504)	✓	✓*4	✓	✓
[M]8184*3	Data communication error (Slave station No.1)	✓	✓		(M505)	✓	✓*4	✓	✓
[M]8185*3	Data communication error (Slave station No.2)	✓	✓		(M506)	✓	✓*4	✓	✓
[M]8186*3	Data communication error (Slave station No.3)	✓	✓		(M507)	✓	✓*4	✓	✓
[M]8187*3	Data communication error (Slave station No.4)	✓	✓		(M508)	✓	✓*4	✓	✓
[M]8188*3	Data communication error (Slave station No.5)	✓	✓		(M509)	✓	✓*4	✓	✓
[M]8189*3	Data communication error (Slave station No.6)	✓	✓		(M510)	✓	✓*4	✓	✓
[M]8190*3	Data communication error (Slave station No.7)	✓	✓		(M511)	✓	✓*4	✓	✓
[M]8191*3	Data communication in execution	✓	✓		(M503)	✓	✓*4	✓	✓
[M]8192	Not used	–	–	–	–	–	–	–	–
[M]8193		–	–	–	–	–	–	–	–
[M]8194		–	–	–	–	–	–	–	–
[M]8195		–	–	–	–	–	–	–	–
[M]8196		–	–	–	–	–	–	–	–
[M]8197		–	–	–	–	–	–	–	–

*3. In FX1S PLCs, use numbers shown inside parentheses.
*4. Available in Ver. 2.00 or later.

High Speed Counter Edge Count Specification (Refer to Subsection 4.7.8 for details.)									
M 8198*5*6	C251, C252, C254: 1/4 edge count selector	✓	✓	–	–	–	–	–	–
M 8199*5*6	C253, C255, or C253 (OP): 1/4 edge count selector	✓	✓	–	–	–	–	–	–

*5. OFF: 1 edge count
ON: 4 edge count

*6. Cleared when PLC switches from RUN to STOP.

Number and name		Operation and function	Applicable model: FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Counter Up/down Counter Counting Direction (Refer to Section 4.6 for details.)										
M 8200	C200		✓	✓	–	–	✓	✓	✓	✓
M 8201	C201		✓	✓	–	–	✓	✓	✓	✓
M 8202	C202		✓	✓	–	–	✓	✓	✓	✓
M 8203	C203		✓	✓	–	–	✓	✓	✓	✓
M 8204	C204		✓	✓	–	–	✓	✓	✓	✓
M 8205	C205		✓	✓	–	–	✓	✓	✓	✓
M 8206	C206		✓	✓	–	–	✓	✓	✓	✓
M 8207	C207		✓	✓	–	–	✓	✓	✓	✓
M 8208	C208		✓	✓	–	–	✓	✓	✓	✓
M 8209	C209		✓	✓	–	–	✓	✓	✓	✓
M 8210	C210		✓	✓	–	–	✓	✓	✓	✓
M 8211	C211		✓	✓	–	–	✓	✓	✓	✓
M 8212	C212		✓	✓	–	–	✓	✓	✓	✓
M 8213	C213		✓	✓	–	–	✓	✓	✓	✓
M 8214	C214		✓	✓	–	–	✓	✓	✓	✓
M 8215	C215		✓	✓	–	–	✓	✓	✓	✓
M 8216	C216	When M8□□□ is ON, the corresponding C□□□ is changed to down mode. • ON: Down count operation • OFF: Up count operation	✓	✓	–	–	✓	✓	✓	✓
M 8217	C217		✓	✓	–	–	✓	✓	✓	✓
M 8218	C218		✓	✓	–	–	✓	✓	✓	✓
M 8219	C219		✓	✓	–	–	✓	✓	✓	✓
M 8220	C220		✓	✓	–	–	✓	✓	✓	✓
M 8221	C221		✓	✓	–	–	✓	✓	✓	✓
M 8222	C222		✓	✓	–	–	✓	✓	✓	✓
M 8223	C223		✓	✓	–	–	✓	✓	✓	✓
M 8224	C224		✓	✓	–	–	✓	✓	✓	✓
M 8225	C225		✓	✓	–	–	✓	✓	✓	✓
M 8226	C226		✓	✓	–	–	✓	✓	✓	✓
M 8227	C227		✓	✓	–	–	✓	✓	✓	✓
M 8228	C228		✓	✓	–	–	✓	✓	✓	✓
M 8229	C229		✓	✓	–	–	✓	✓	✓	✓
M 8230	C230		✓	✓	–	–	✓	✓	✓	✓
M 8231	C231		✓	✓	–	–	✓	✓	✓	✓
M 8232	C232		✓	✓	–	–	✓	✓	✓	✓
M 8233	C233		✓	✓	–	–	✓	✓	✓	✓
M 8234	C234		✓	✓	–	–	✓	✓	✓	✓

Number and name		Operation and function	Applicable model: FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
High Speed Counter	Up/down Counter Counting Direction (Refer to Section 4.7 for details.)									
M 8235	C235	When M8□□□ is ON, the corresponding C□□□ is changed to down mode. • ON: Down count operation • OFF: Up count operation	✓	✓	–	✓	✓	✓	✓	✓
M 8236	C236		✓	✓	–	✓	✓	✓	✓	✓
M 8237	C237		✓	✓	–	✓	✓	✓	✓	✓
M 8238	C238		✓	✓	–	✓	✓	✓	✓	✓
M 8239	C239		✓	✓	–	✓	✓	✓	✓	✓
M 8240	C240		✓	✓	–	✓	✓	✓	✓	✓
M 8241	C241		✓	✓	–	✓	✓	✓	✓	✓
M 8242	C242		✓	✓	–	✓	✓	✓	✓	✓
M 8243	C243		✓	✓	–	✓	✓	✓	✓	✓
M 8244	C244		✓	✓	–	✓	✓	✓	✓	✓
M 8245	C245		✓	✓	–	✓	✓	✓	✓	✓
High Speed Counter	Up/down Counter Monitoring (Refer to Section 4.7 for details.)									
[M]8246	C246	When C□□□ of 1-phase 2-input or 2-phase 2-input counter is in down mode, the corresponding M8□□□ turns ON. • ON: Down count operation • OFF: Up count operation	✓	✓	–	✓	✓	✓	✓	✓
[M]8247	C247		✓	✓	–	✓	✓	✓	✓	✓
[M]8248	C248		✓	✓	–	✓	✓	✓	✓	✓
[M]8249	C249		✓	✓	–	✓	✓	✓	✓	✓
[M]8250	C250		✓	✓	–	✓	✓	✓	✓	✓
[M]8251	C251		✓	✓	–	✓	✓	✓	✓	✓
[M]8252	C252		✓	✓	–	✓	✓	✓	✓	✓
[M]8253	C253		✓	✓	–	✓	✓	✓	✓	✓
[M]8254	C254		✓	✓	–	✓	✓	✓	✓	✓
[M]8255	C255		✓	✓	–	✓	✓	✓	✓	✓
[M]8256 to [M]8259	Not used		–	–	–	–	–	–	–	–
Analog Special Adapter (Refer to Subsection 36.2.17 for applicability of each analog special adapter.)										
M 8260 to M 8269	1st special adapter*1		✓	✓*2	–	–	–	–	–	–
M 8270 to M 8279	2nd special adapter*1		✓	✓*2	–	–	–	–	–	–
M 8280 to M 8289	3rd special adapter*1		✓	✓*2	–	–	–	–	–	–
M 8290 to M 8299	4th special adapter*1		✓	✓*2	–	–	–	–	–	–
Flags										
[M]8300 to [M]8303	Not used		–	–	–	–	–	–	–	–
[M]8304 Zero	ON when the multiplication and division calculated result is 0		✓*3	✓*3	–	–	–	–	–	–
[M]8305	Not used		–	–	–	–	–	–	–	–
[M]8306 Carry	ON when the division calculated result overflows		✓*3	✓*3	–	–	–	–	–	–
[M]8307 to [M]8315	Not used		–	–	–	–	–	–	–	–

*1. Count the number of connected FX3U-4AD-ADP, FX3U-4DA-ADP, FX3U-4AD-TC-ADP, and FX3U-4AD-PT-ADP units from the main unit.

*2. Available in Ver. 1.20 or later.

*3. Available in Ver. 2.30 or later.

Number and name	Operation and function	Applicable model							
		FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Unconnected I/O Designation Error (Refer to Chapter 37 for details.) and flag									
[M]8316[*4]	Unconnected I/O designation error	✓	✓	D8316 D8317	–	–	–	–	–
[M]8317	Not used	–	–	–	–	–	–	–	–
[M]8318	BFM initialization failure ON when a FROM/TO error has occurred in a special function block/unit as specified in the BFM initialization function at changing PLC from STOP to RUN. When M8318 turns ON, the unit number in which the error has occurred is stored in D8318, and the BFM number is stored in D8319.	✓	✓[*5]	D8318 D8319	–	–	–	–	–
[M]8319 to [M]8327	Not used	–	–	–	–	–	–	–	–
[M]8328	Instruction non-execution	✓	✓[*5]	–	–	–	–	–	–
[M]8329	Instruction execution abnormal end	✓	✓	–	–	–	–	–	–

*4. If the I/O device numbers are unavailable, M8316 turns ON when its directly designated to device numbers including LD, AND, OR, and OUT instructions or indirectly designated by index.

*5. Available in Ver. 2.20 or later.

Number and name	Operation and function	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Timing Clock (Refer to Section 24.3 for details.) and Positioning [FX3U and FX3UC PLCs] (Refer to Positioning Control Edition for details.)									
[M]8330	DUTY (FNC186) instruction: Timing clock output 1	✓	✓[*1]	D8330	–	–	–	–	–
[M]8331	DUTY (FNC186) instruction: Timing clock output 2	✓	✓[*1]	D8331	–	–	–	–	–
[M]8332	DUTY (FNC186) instruction: Timing clock output 3	✓	✓[*1]	D8332	–	–	–	–	–
[M]8333	DUTY (FNC186) instruction: Timing clock output 4	✓	✓[*1]	D8333	–	–	–	–	–
[M]8334	DUTY (FNC186) instruction: Timing clock output 5	✓	✓[*1]	D8334	–	–	–	–	–
[M]8335	Not used	–	–	–	–	–	–	–	–
M 8336[*2]	DVIT (FNC151) instruction: Interrupt input specification function enabled	✓	✓[*3]	D8336	–	–	–	–	–
[M]8337	Not used	–	–	–	–	–	–	–	–
M 8338	PLSV (FNC157) instruction: Acceleration/deceleration operation	✓	✓[*1]	–	–	–	–	–	–
[M]8339	Not used	–	–	–	–	–	–	–	–
[M]8340	[Y000] Pulse output monitor (ON: BUSY/ OFF: READY)	✓	✓	–	–	–	–	–	–
M 8341[*2]	[Y000] Clear signal output function enable	✓	✓	–	–	–	–	–	–
M 8342[*2]	[Y000] Zero return direction specification	✓	✓	–	–	–	–	–	–
M 8343	[Y000] Forward limit	✓	✓	–	–	–	–	–	–
M 8344	[Y000] Reverse limit	✓	✓	–	–	–	–	–	–
M 8345[*2]	[Y000] DOG signal logic reverse	✓	✓	–	–	–	–	–	–
M 8346[*2]	[Y000] Zero point signal logic reverse	✓	✓	–	–	–	–	–	–
M 8347[*2]	[Y000] Interrupt signal logic reverse	✓	✓	–	–	–	–	–	–
[M]8348	[Y000] Positioning instruction activation	✓	✓	–	–	–	–	–	–
M 8349[*2]	[Y000] Pulse output stop command	✓	✓	–	–	–	–	–	–
[M]8350	[Y001] Pulse output monitor (ON: BUSY/ OFF: READY)	✓	✓	–	–	–	–	–	–
M 8351[*2]	[Y001] Clear signal output function enable	✓	✓	–	–	–	–	–	–
M 8352[*2]	[Y001] Zero return direction specification	✓	✓	–	–	–	–	–	–
M 8353	[Y001] Forward limit	✓	✓	–	–	–	–	–	–
M 8354	[Y001] Reverse limit	✓	✓	–	–	–	–	–	–
M 8355[*2]	[Y001] DOG signal logic reverse	✓	✓	–	–	–	–	–	–
M 8356[*2]	[Y001] Zero point signal logic reverse	✓	✓	–	–	–	–	–	–
M 8357[*2]	[Y001] Interrupt signal logic reverse	✓	✓	–	–	–	–	–	–
[M]8358	[Y001] Positioning instruction activation	✓	✓	–	–	–	–	–	–
M 8359[*2]	[Y001] Pulse output stop command	✓	✓	–	–	–	–	–	–

Number and name	Operation and function	Applicable model							
		FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Timing Clock (Refer to Section 24.3 for details.) and Positioning [FX3U and FX3UC PLCs] (Refer to Positioning Control Edition for details.)									
[M]8360	[Y002] Pulse output monitor (ON: BUSY/ OFF: READY)	✓	✓	–	–	–	–	–	–
M 8361*2	[Y002] Clear signal output function enable	✓	✓	–	–	–	–	–	–
M 8362*2	[Y002] Zero return direction specification	✓	✓	–	–	–	–	–	–
M 8363	[Y002] Forward limit	✓	✓	–	–	–	–	–	–
M 8364	[Y002] Reverse limit	✓	✓	–	–	–	–	–	–
M 8365*2	[Y002] DOG signal logic reverse	✓	✓	–	–	–	–	–	–
M 8366*2	[Y002] Zero point signal logic reverse	✓	✓	–	–	–	–	–	–
M 8367*2	[Y002] Interrupt signal logic reverse	✓	✓	–	–	–	–	–	–
[M]8368	[Y002] Positioning instruction activation	✓	✓	–	–	–	–	–	–
M 8369*2	[Y002] Pulse output stop command	✓	✓	–	–	–	–	–	–

*1. Available in Ver. 2.20 or later.

*2. Cleared when PLC switches from RUN to STOP.

*3. Available in the Ver. 1.30 or later.

Positioning [FX3U PLCs] (Refer to Positioning Control Manual for details.)									
[M]8370	[Y003] Pulse output monitor (ON: BUSY/ OFF: READY)	✓*2	–	–	–	–	–	–	–
M 8371*1	[Y003] Clear signal output function enable	✓*2	–	–	–	–	–	–	–
M 8372*1	[Y003] Zero return direction specification	✓*2	–	–	–	–	–	–	–
M 8373	[Y003] Forward limit	✓*2	–	–	–	–	–	–	–
M 8374	[Y003] Reverse limit	✓*2	–	–	–	–	–	–	–
M 8375*1	[Y003] DOG signal logic reverse	✓*2	–	–	–	–	–	–	–
M 8376*1	[Y003] Zero point signal logic reverse	✓*2	–	–	–	–	–	–	–
M 8377*1	[Y003] Interrupt signal logic reverse	✓*2	–	–	–	–	–	–	–
[M]8378	[Y003] Positioning instruction activation	✓*2	–	–	–	–	–	–	–
M 8379*1	[Y003] Pulse output stop command	✓*2	–	–	–	–	–	–	–

*1. Cleared when PLC switches from RUN to STOP.

*2. Available only when two FX3U-2HSY-ADP units are connected to an FX3U PLC.

High Speed Counter Function (Refer to Subsection 4.7.5 for details.)									
[M]8380*3	Operation status of C235, C241, C244, C246, C247, C249, C251, C252, and C254	✓	✓	–	–	–	–	–	–
[M]8381*3	Operation status of C236	✓	✓	–	–	–	–	–	–
[M]8382*3	Operation status of C237, C242, and C245	✓	✓	–	–	–	–	–	–
[M]8383*3	Operation status of C238, C248, C248 (OP), C250, C253, and C255	✓	✓	–	–	–	–	–	–
[M]8384*3	Operation status of C239 and C243	✓	✓	–	–	–	–	–	–
[M]8385*3	Operation status of C240	✓	✓	–	–	–	–	–	–
[M]8386*3	Operation status of C244 (OP)	✓	✓	–	–	–	–	–	–
[M]8387*3	Operation status of C245 (OP)	✓	✓	–	–	–	–	–	–
[M]8388	Contact for high speed counter function change	✓	✓	–	–	–	–	–	–
M 8389	External reset input logic reverse	✓	✓	–	–	–	–	–	–
M 8390	Function changeover device for C244	✓	✓	–	–	–	–	–	–
M 8391	Function changeover device for C245	✓	✓	–	–	–	–	–	–
M 8392	Function changeover device for C248 and C253	✓	✓	–	–	–	–	–	–

*3. Cleared when PLC switches from STOP to RUN.

Number and name	Operation and function	Applicable model							
		FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Interrupt Program									
[M]8393	Contact for delay time setting	✓	✓	D8393	–	–	–	–	–
[M]8394	HCMOV (FNC189): Drive contact for interrupt program	✓	✓	–	–	–	–	–	–
[M]8395	Not used	–	–	–	–	–	–	–	–
[M]8396		–	–	–	–	–	–	–	–
[M]8397		–	–	–	–	–	–	–	–
Ring Counter									
M 8398	Ring counter operation (in units of 1ms, 32 bits)*4	✓	✓	D8398, D8399	–	–	–	–	–
[M]8399	Not used	–	–	–	–	–	–	–	–

*4. 1ms ring counter (D8399, D8398) will operate after M8398 turns ON.

Number and name	Operation and function	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
RS2 (FNC 87) [ch1] (Refer to Data Communication Edition for details.)									
[M]8400	Not used	–	–	–	–	–	–	–	–
[M]8401	RS2 (FNC 87) [ch1] Send wait flag	✓	✓	–	–	–	–	–	–
M 8402	RS2 (FNC 87) [ch1] Send request	✓	✓	D8402	–	–	–	–	–
M 8403	RS2 (FNC 87) [ch1] Receive complete flag	✓	✓	D8403	–	–	–	–	–
[M]8404	RS2 (FNC 87) [ch1] Carrier detection flag	✓	✓	–	–	–	–	–	–
[M]8405	RS2 (FNC87) [ch1] Data set ready (DSR) flag	✓*1	✓*1	–	–	–	–	–	–
[M]8406	Not used	–	–	–	–	–	–	–	–
[M]8407		–	–	–	–	–	–	–	–
[M]8408		–	–	–	–	–	–	–	–
M 8409	RS2 (FNC 87) [ch1] Time-out check flag	✓	✓	–	–	–	–	–	–

*1. Available in Ver. 2.30 or later.

Number and name	Operation and function	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
RS2 (FNC 87) [ch2] and Computer Link [ch2] (Refer to Data Communication Edition for details.)									
[M]8410 to [M]8420	Not used	–	–	–	–	–	–	–	–
[M]8421	RS2 (FNC 87) [ch2] Send wait flag	✓	✓	–	–	–	–	–	–
M 8422	RS2 (FNC 87) [ch2] Send request	✓	✓	D8422	–	–	–	–	–
M 8423	RS2 (FNC 87) [ch2] Receive complete flag	✓	✓	D8423	–	–	–	–	–
[M]8424	RS2 (FNC 87) [ch2] Carrier detection flag	✓	✓	–	–	–	–	–	–
[M]8425	RS2 (FNC87) [ch2] Data set ready (DSR) flag	✓*2	✓*2	–	–	–	–	–	–
[M]8426	Computer link [ch2] Global ON	✓	✓		–	–	–	–	–
[M]8427	Computer link [ch2] On-demand send processing	✓	✓	D8427	–	–	–	–	–
M 8428	Computer link [ch2] On-demand error flag	✓	✓	D8428	–	–	–	–	–
M 8429	Computer link [ch2] On-demand Word/Byte changeover RS2 (FNC 87) [ch2] Time-out check flag	✓	✓	D8429	–	–	–	–	–

*2. Available in Ver. 2.30 or later.

Number and name	Operation and function	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Error Detection (Refer to Chapter 37 for details.)									
[M]8430 to [M]8437	Not used	–	–	–	–	–	–	–	–
M 8438	Serial communication error 2 [ch2]	✓	✓	D8438	–	–	–	–	–
[M]8439 to [M]8448	Not used	–	–	–					
[M]8449	Special block error flag	✓	✓*3	D8449					
[M]8450 to [M]8459	Not used	–	–	–	–	–	–	–	–

*3. Available in Ver. 2.20 or later.

31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC•STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List

Number and name	Operation and function	Applicable model							
		FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Positioning [FX3U and FX3UC PLCs] (Refer to Positioning Control Edition for details.)									
M 8460	DVIT (FNC151) instruction [Y000] User interrupt input command	✓	✓*4	D8336	–	–	–	–	–
M 8461	DVIT (FNC151) instruction [Y001] User interrupt input command	✓	✓*4	D8336	–	–	–	–	–
M 8462	DVIT (FNC151) instruction [Y002] User interrupt input command	✓	✓*4	D8336	–	–	–	–	–
M 8463	DVIT (FNC151) instruction [Y003] User interrupt input command	✓*5	–	D8336	–	–	–	–	–
M 8464	DSZR (FNC150), ZRN (FNC156) instructions [Y000] Clear signal device specification function enabled	✓	✓*4	D8464	–	–	–	–	–
M 8465	DSZR (FNC150), ZRN (FNC156) instructions [Y001] Clear signal device specification function enabled	✓	✓*4	D8465	–	–	–	–	–
M 8466	DSZR (FNC150), ZRN (FNC156) instructions [Y002] Clear signal device specification function enabled	✓	✓*4	D8466	–	–	–	–	–
M 8467	DSZR (FNC150), ZRN (FNC156) instructions [Y003] Clear signal device specification function enabled	✓*5	–	D8467	–	–	–	–	–
[M]8468 to [M]8511	Not used	–	–	–	–	–	–	–	–

*4. Available in Ver. 2.20 or later.

*5. Available only when two FX3U-2HSY-ADP adapters are connected to an FX3U PLC.

36.1.2 Special Data Register (D8000 to D8511)

Number and name	Content of register	Applicable model							
		FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
PLC Status									
D 8000 Watchdog timer	Default value is shown on the right (in 1ms steps) (Writes from system ROM at power ON) Value overwritten by program is valid after END or WDT instruction execution. →**Refer to Subsection 36.2.2.**	200	200	–	200	200	200	200	200
[D]8001 PLC type and system version	2 4 1 0 0 BCD converted value (2 4: Indicated on the right; 1 0 0: Version 1.00)	24	24	D8101[*1]	22	26	24	26	24
[D]8002 Memory capacity	• 2...2K steps • 4...4K steps • 8...8K steps • If 16K steps or more "8" is written to D8002 and "16" or "64" is written to D8102.	✓ 8	✓ 8	D8102	✓ 2	✓ 8	✓ 4 8	✓ 8	✓ 4 8
[D]8003 Memory type	Type of internal memory/RAM/EEPROM/EPROM cassette or ON/OFF status of protect switch is stored Present value / Type of memory / Protect switch: 00H / RAM memory cassette / – 01H / EPROM memory cassette / – 02H / EEPROM memory cassette or flash memory cassette / OFF 0AH / EEPROM memory cassette or flash memory cassette / ON 10H / Built-in memory in PLC / –	✓	✓	–	✓	✓	✓	✓	✓
[D]8004 Error number M	8 0 6 0 BCD converted value 8060 to 8068 (when M8004 is ON)	✓	✓	M8004	✓	✓	✓	✓	✓
[D]8005 Battery voltage	3 0 BCD converted value (in 0.1-V units) Battery voltage present value (Example: 3.0V)	✓	✓	M8005	–	–	✓	–	✓
[D]8006 Low battery voltage detection level	Default: • FX2N and FX2NC PLCs: 3.0V (in 0.1-V units) • FX3U and FX3UC PLCs: 2.7V (in 0.1-V units) (Writes from system ROM at power ON)	✓	✓	M8006	–	–	✓	–	✓

*1. The corresponding special device D8101 is available only in FX3U and FX3UC PLCs.
No corresponding device is available in FX1S, FX1N, FX2N, FX1NC, and FX2NC PLCs.

Number and name	Content of register	Applicable model							
		FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
PLC Status									
[D]8007 Momentary power failure count	Operation frequency of M8007 is stored. Cleared at power-off.	✓	✓	M8007	–	–	✓	–	✓
D 8008 Power failure detection	Default:*1 • FX3U, FX2N PLCs: 10 ms (AC power supply type) • FX2NC, FX3UC PLCs: 5 ms (DC power supply type)	✓	✓	M8008	–	–	✓	–	✓
[D]8009 24V DC failed device	Minimum input device number of extension units and extension power units in which 24V DC has failed.	✓	✓	M8009	–	–	✓	–	✓

*1. The power failure detection time in FX2N and FX2NC PLCs is as follows.
- 10ms for FX2N PLCs used with a 100V AC power supply system. D8008 is set to 10ms by default.
- 100ms maximum for FX2N PLCs used with a 200V AC power supply system. D8008 can be set in the range from 10 to 100ms.
- 5ms for FX2N PLCs used with a DC power supply type. Write "K-1" to D8008 for correction.
- 5ms for FX2NC PLCs used with a DC power supply type. System writes "K-1" to D8008 for correction. Do not make any change in a sequence program.
- For FX3U and FX3UC PLCs, refer to Subsection 36.2.4.

Number and name	Content of register	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Clock									
[D]8010 Present scan time	Accumulated instruction-execution time from step 0 (in 0.1-ms units) →**Refer to Subsection 36.2.5**	✓ Same as on the right	✓ Same as on the right	–	✓ Indicated value includes waiting time of constant scan operation (when M8039 is activated).				
[D]8011 Minimum scan time	Minimum value of scan time (in 0.1-ms units) →**Refer to Subsection 36.2.5.**			–					
[D]8012 Maximum scan time	Maximum value of scan time (in 0.1-ms units) →**Refer to Subsection 36.2.5.**			–					
D 8013 Second data	0 to 59 seconds (for real time clock) →**Refer to Subsection 36.2.7.**	✓	✓	–	✓	✓	✓	✓	✓*2
D 8014 Minute data	0 to 59 minutes (for real time clock) →**Refer to Subsection 36.2.7.**	✓	✓	–	✓	✓	✓	✓	✓*2
D 8015 Hour data	0 to 23 hours (for real time clock) →**Refer to Subsection 36.2.7.**	✓	✓	–	✓	✓	✓	✓	✓*2
D 8016 Day data	1 to 31 days (for real time clock) →**Refer to Subsection 36.2.7.**	✓	✓	–	✓	✓	✓	✓	✓*2
D 8017 Month data	1 to 12 months (for real time clock) →**Refer to Subsection 36.2.7.**	✓	✓	–	✓	✓	✓	✓	✓*2
D 8018 Year data	2 digits of year data (0 to 99) (for real time clock) →**Refer to Subsection 36.2.7.**	✓	✓	–	✓	✓	✓	✓	✓*2
D 8019 Day-of-the-week data	0 (Sunday) to 6 (Saturday) (for real time clock) →**Refer to Subsection 36.2.7.**	✓	✓	–	✓	✓	✓	✓	✓*2

*2. A memory board having the real time clock function is required in FX2NC PLCs.

Number and name	Content of register	Applicable model							
		FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Input Filter									
D 8020 Input filter adjustment	Input filter value of X000 to X017 (X000 to X007 in FX3U-16M□) (Default: 10 ms) →**Refer to Subsection 36.2.9.**	✓	✓	–	✓	✓*1	✓	✓	✓
[D]8021	Not used	–	–	–	–	–	–	–	–
[D]8022		–	–	–	–	–	–	–	–
[D]8023		–	–	–	–	–	–	–	–
[D]8024		–	–	–	–	–	–	–	–
[D]8025		–	–	–	–	–	–	–	–
[D]8026		–	–	–	–	–	–	–	–
[D]8027		–	–	–	–	–	–	–	–
Index Register Z0 and V0									
[D]8028	Value of Z0 (Z) register*2	✓	✓	–	✓	✓	✓	✓	✓
[D]8029	Value of V0 (V) register*2	✓	✓	–	✓	✓	✓	✓	✓

*1. In FX1N PLCs, the input filter can be adjusted in X000 to X007.

*2. The values of Z1 to Z7 and V1 to V7 are stored in D8182 to D8195.

Number and name	Content of register	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Analog Volume [FX1S, FX1N]									
[D]8030	Value of analog volume VR1 (Integer from 0 to 255)	–	–	–	✓	✓	–	–	–
[D]8031	Value of analog volume VR2 (Integer from 0 to 255)	–	–	–	✓	✓	–	–	–
Constant Scan									
[D]8032	Not used	–	–	–	–	–	–	–	–
[D]8033		–	–	–	–	–	–	–	–
[D]8034		–	–	–	–	–	–	–	–
[D]8035		–	–	–	–	–	–	–	–
[D]8036		–	–	–	–	–	–	–	–
[D]8037		–	–	–	–	–	–	–	–
[D]8038		–	–	–	–	–	–	–	–
D 8039 Constant scan duration	Default: 0 ms (in 1 ms steps) (Writes from system ROM at power ON) Can be overwritten by program →**Refer to Subsection 36.2.15.**	✓	✓	M8039	✓	✓	✓	✓	✓

Number and name	Content of register	Applicable model							
		FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Stepladder and Annunciator									
[D]8040*1 ON state number 1	The smallest number out of active state ranging from S0 to S899 and S1000 to S4095*2 is stored in D8040 and the second-smallest state number is stored in D8041. Active state numbers are then sequentially stored in registers up to D8047 (Max. 8 points).	✓	✓	M8047	✓	✓	✓	✓	✓
[D]8041*1 ON state number 2		✓	✓		✓	✓	✓	✓	✓
[D]8042*1 ON state number 3		✓	✓		✓	✓	✓	✓	✓
[D]8043*1 ON state number 4		✓	✓		✓	✓	✓	✓	✓
[D]8044*1 ON state number 5		✓	✓		✓	✓	✓	✓	✓
[D]8045*1 ON state number 6		✓	✓		✓	✓	✓	✓	✓
[D]8046*1 ON state number 7		✓	✓		✓	✓	✓	✓	✓
[D]8047*1 ON state number 8		✓	✓		✓	✓	✓	✓	✓
[D]8048	Not used	–	–	–	–	–	–	–	–
[D]8049*1 On state minimum number	When M8049 is ON, the smallest number out of active annunciator relay ranging from S900 to S999 is stored in D8049.	✓	✓	M8049	–	–	✓	–	✓
[D]8050 to [D]8059	Not used	–	–	–	–	–	–	–	–

*1. Executed at END instruction.

*2. S1000 to S4095 are available only in FX3U/FX3UC PLCs.

Error Detection (Refer to Chapter 37 for details.)									
[D]8060	If the unit or block corresponding to a programmed I/O number is not actually loaded, M8060 is set to ON and the first device number of the erroneous block is written to D8060. Example: If X020 is unconnected. 1 0 2 0 BCD converted value Device number 10 to 337 1: Input X 0: Output Y	✓	✓	M8060	–	–	✓	–	✓
[D]8061	Error code for PLC hardware error	✓	✓	M8061	✓	✓	✓	✓	✓
[D]8062	Error code for PLC/PP communication error	✓	✓	M8062	–	–	✓	–	✓
[D]8063	Error code for serial communication error 1 [ch1]	✓	✓	M8063	✓	✓	✓	✓	✓
[D]8064	Error code for parameter error	✓	✓	M8064	✓	✓	✓	✓	✓
[D]8065	Error code for syntax error	✓	✓	M8065	✓	✓	✓	✓	✓
[D]8066	Error code for ladder error	✓	✓	M8066	✓	✓	✓	✓	✓
[D]8067	Error code for operation error	✓	✓	M8067	✓	✓	✓	✓	✓
D 8068*3	Operation error step number latched	✓*4	✓*4	M8068	✓	✓	✓	✓	✓
[D]8069*3	Error step number of M8065 to M8067	✓*5	✓*5	M8065 to M8067	✓	✓	✓	✓	✓

*3. Cleared when PLC switches from STOP to RUN.

*4. In case of 32K steps or more, step number is stored in [D8313, D8312].

*5. In case of 32K steps or more, step number is stored in [D8315, D8314].

Number and name	Content of register	Applicable model							
		FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Parallel Link (Refer to Data Communication Edition for details.)									
[D]8070	Parallel link error time-out check time: 500 ms	✓	✓	–	✓	✓	✓	✓	✓
[D]8071	Not used	–	–	–	–	–	–	–	–
[D]8072		–	–	–	–	–	–	–	–
[D]8073		–	–	–	–	–	–	–	–
Sampling Trace[*1]									
[D]8074	These devices are occupied by the PLC system when the sampling trace function is used in the A6GPP, A6PHP, A7PHP, or personal computer[*1].	✓	✓	M8075 to M8079	–	–	✓	–	✓
[D]8075		✓	✓		–	–	✓	–	✓
[D]8076		✓	✓		–	–	✓	–	✓
[D]8077		✓	✓		–	–	✓	–	✓
[D]8078		✓	✓		–	–	✓	–	✓
[D]8079		✓	✓		–	–	✓	–	✓
[D]8080		✓	✓		–	–	✓	–	✓
[D]8081		✓	✓		–	–	✓	–	✓
[D]8082		✓	✓		–	–	✓	–	✓
[D]8083		✓	✓		–	–	✓	–	✓
[D]8084		✓	✓		–	–	✓	–	✓
[D]8085		✓	✓		–	–	✓	–	✓
[D]8086		✓	✓		–	–	✓	–	✓
[D]8087		✓	✓		–	–	✓	–	✓
[D]8088		✓	✓		–	–	✓	–	✓
[D]8089		✓	✓		–	–	✓	–	✓
[D]8090		✓	✓		–	–	✓	–	✓
[D]8091		✓	✓		–	–	✓	–	✓
[D]8092		✓	✓		–	–	✓	–	✓
[D]8093		✓	✓		–	–	✓	–	✓
[D]8094		✓	✓		–	–	✓	–	✓
[D]8095		✓	✓		–	–	✓	–	✓
[D]8096		✓	✓		–	–	✓	–	✓
[D]8097		✓	✓		–	–	✓	–	✓
[D]8098		✓	✓		–	–	✓	–	✓

*1. The sampling trace devices are used by peripheral equipment.

Number and name	Content of register	FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
High Speed Ring Counter									
D 8099	Up-operation high speed ring counter of 0 to 32,767 (in units of 0.1ms, 16-bit)[*2]	✓	✓	M8099	–	–	✓	–	✓
[D]8100	Not used	–	–	–	–	–	–	–	–

*2. For FX2N/FX2NC, 0.1ms high speed ring counter D8099 will operate after END instruction is executed while M8099 is ON.
For FX3U/FX3UC, 0.1ms high speed ring counter D8099 will operate after M8099 turns ON.

Number and name	Content of register	Applicable model: FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Memory Information									
[D]8101 PLC type and system version	1 6 1 0 0 BCD converted value 16: FX3U/FX3UC 100: Version 1.00	✓	✓	–	–	–	–	–	–
[D]8102	2 ...... 2K steps 4 ...... 4K steps 8 ...... 8K steps 16 .... 16K steps 64 .... 64K steps	✓ 16[*1] 64	✓ 16[*1] 64	–	✓ 2	✓ 8	✓ 4 8 16	✓ 8	✓ 4 8 16
[D]8103	Not used	–	–	–	–	–	–	–	–
[D]8104	Identity code for function extension memory	–	–	M8104	–	–	✓[*2]	–	✓[*2]
[D]8105	Version of function extension memory (Version 1.00 = 100)	–	–		–	–	✓[*2]	–	✓[*2]
[D]8106	Not used	–	–	–	–	–	–	–	–
[D]8107	Number of registered device comments	✓	✓	M8107	–	–	–	–	–
[D]8108	Number of special function units/blocks connected	✓	✓	–	–	–	–	–	–

*1. When loading FX3U-FLROM-16

*2. Available in Ver. 3.00 or later

Number and name	Content of register	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Output Refresh Error (Refer to Chapter 37 for details.)									
[D]8109	Y number where output refresh error occurs	✓	✓	M8109	–	–	✓	–	✓
[D]8110	Not used	–	–	–	–	–	–	–	–
[D]8111		–	–	–	–	–	–	–	–
Expansion Board Dedicated to FX1S and FX1N									
[D]8112	FX1N-2AD-BD: Digital value of ch1	–	–	M8112	✓	✓	–	–	–
[D]8113	FX1N-2AD-BD: Digital value of ch2	–	–	M8113	✓	✓	–	–	–
D 8114	FX1N-1DA-BD: Digital value to be output	–	–	M8112	✓	✓	–	–	–
[D]8115 to [D]8119	Not used	–	–	–	–	–	–	–	–
RS (FNC 80) and Computer Link [ch1] (Refer to Data Communication Edition for details.)									
D 8120[*2]	RS (FNC 80) instruction and computer link [ch1] Communication format setting	✓	✓	–	✓	✓	✓	✓	✓
D 8121[*2]	Computer link [ch1] Station number setting	✓	✓	–	✓	✓	✓	✓	✓
[D]8122[*3]	RS (FNC 80) instruction: Remaining points of transmit data	✓	✓	M8122	✓	✓	✓	✓	✓
[D]8123[*3]	RS (FNC 80) instruction: Monitoring receive data points	✓	✓	M8123	✓	✓	✓	✓	✓
D 8124	RS (FNC 80) instruction: Header <Default: STX>	✓	✓	–	✓	✓	✓	✓	✓
D 8125	RS (FNC 80) instruction: Terminator <Default: ETX>	✓	✓	–	✓	✓	✓	✓	✓
[D]8126	Not used	–	–	–	–	–	–	–	–
D 8127	Computer link [ch1] Specification of on-demand head device register	✓	✓	M8126 to M8129	✓	✓	✓	✓	✓
D 8128	Computer link [ch1] Specification of on-demand data length register	✓	✓		✓	✓	✓	✓	✓
D 8129[*2]	RS (FNC 80) instruction, computer link [ch1] Time-out time setting	✓	✓		✓	✓	✓	✓	✓

*3. Latch (battery backed) device

*4. Cleared when PLC switches from RUN to STOP.

Number and name		Content of register	Applicable model							
			FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
High Speed Counter Comparison, High Speed Table, and Positioning [Positioning is supported in FX1S, FX1N, and FX1NC PLCs.]										
[D]8130		HSZ (FNC 55) instruction: High speed comparison table counter	✓	✓	M8130	–	–	✓	–	✓
[D]8131		HSZ (FNC 55) and PLSY (FNC 57) instructions: Speed pattern table counter	✓	✓	M8132	–	–	✓	–	✓
[D]8132	Lower	HSZ (FNC 55) and PLSY (FNC 57) instructions: Speed pattern frequency	✓	✓	M8132	–	–	✓	–	✓
[D]8133	Upper									
[D]8134	Lower	HSZ (FNC 55) and PLSY (FNC 57) instructions: Number of target pulses for speed pattern	✓	✓	M8132	–	–	✓	–	✓
[D]8135	Upper									
D 8136	Lower	PLSY (FNC 57), PLSR (FNC 59) instructions: Accumulated total number of pulses output to Y000 and Y001	✓	✓	–	✓	✓	✓	✓	✓
D 8137	Upper									
[D]8138		HSCT (FNC280) instruction: Table count	✓	✓	M8138	–	–	–	–	–
[D]8139		HSCS (FNC 53), HSCR (FNC 54), HSZ (FNC 55), and HSCT (FNC280) instructions: Number of instructions being executed	✓	✓	M8139	–	–	–	–	–
D 8140	Lower	Accumulated number of pulses output to Y000 for PLSY (FNC 57) and PLSR (FNC 59) instructions, or current address of Y000 for positioning instruction	✓	✓	–	✓	✓	✓	✓	✓
D 8141	Upper									
D 8142	Lower	Accumulated number of pulses output to Y001 for PLSY (FNC 57) and PLSR (FNC 59) instructions, or current address of Y001 for positioning instruction	✓	✓	–	✓	✓	✓	✓	✓
D 8143	Upper									
[D]8144		Not used	–	–	–	–	–	–	–	–
D 8145		ZRN (FNC156), DRVI (FNC158), and DRVA (FNC159) instructions: Bias speed Default: 0	–	–	–	✓	✓	–	✓	–
D 8146	Lower	ZRN (FNC156), DRVI (FNC158), and DRVA (FNC159) instructions: Maximum speed • Default in FX1S and FX1N: 100000 • Default in FX1NC: 100000*1	–	–	–	✓	✓	–	✓*1	–
D 8147	Upper									
D 8148		ZRN (FNC156), DRVI (FNC158), and DRVA (FNC159) instructions: Acceleration/deceleration time Default: 100	–	–	–	✓	✓	–	✓	–
[D]8149		Not used	–	–	–	–	–	–	–	–

*1. Must be changed to 10000 or less by PLC program.

Number and name	Content of register	Applicable model							
		FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Inverter Communication Function (Refer to Data Communication Edition for details.)									
D 8150	Response wait time of inverter communication [ch1]	✓	✓	–	–	–	–	–	–
[D]8151	Step number of instruction during inverter communication [ch1] Default: -1	✓	✓	M8151	–	–	–	–	–
[D]8152*1	Error code for inverter communication [ch1]	✓	✓	M8152	–	–	–	–	–
[D]8153	Inverter communication error step number latched [ch1] Default: -1	✓	✓	M8153	–	–	–	–	–
[D]8154	Parameter number when error occurs during IVBWR (FNC274) instruction [ch1] Default: -1	✓	✓	M8154	–	–	–	–	–
	Response waiting time of EXTR (FNC180) instruction	–	–	–	–	–	✓*2	–	✓*2
D 8155	Response wait time of inverter communication [ch2]	✓	✓	–	–	–	–	–	–
[D]8155	Step number during communication of EXTR (FNC180) instruction	–	–	M8155	–	–	✓*2	–	✓*2
[D]8156	Step number of instruction during inverter communication [ch2] Default: -1	✓	✓	M8156	–	–	–	–	–
	Error code for EXTR (FNC180) instruction	–	–	M8156	–	–	✓*2	–	✓*2
[D]8157*1	Error code for inverter communication [ch2]	✓	✓	M8157	–	–	–	–	–
[D]8157	Error step (latched) for EXTR (FNC180) instruction Default: -1	–	–	M8157	–	–	✓*2	–	✓*2
[D]8158	Inverter communication error step number latched [ch2] Default: -1	✓	✓	M8158	–	–	–	–	–
[D]8159	Parameter number when error occurs during IVBWR (FNC274) instruction [ch2] Default: -1	✓	✓	M8159	–	–	–	–	–

*1. Cleared when PLC switches from STOP to RUN.

*2. Available in Ver. 3.00 or later.

Display Module Function [FX1S and FX1N]									
D 8158	FX1N-5DM: Control device (D) Default: -1	–	–	–	✓	✓	–	–	–
D 8159	FX1N-5DM: Control device (M) Default: -1	–	–	–	✓	✓	–	–	–

<table>
<tr><th rowspan="2">Number and name</th><th rowspan="2">Content of register</th><th colspan="8">Applicable model</th></tr>
<tr><th>FX3U</th><th>FX3UC</th><th>Corresponding special device</th><th>FX1S</th><th>FX1N</th><th>FX2N</th><th>FX1NC</th><th>FX2NC</th></tr>
<tr><td colspan="10">Advanced Function</td></tr>
<tr><td>[D]8160</td><td rowspan="4">Not used</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>[D]8161</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>[D]8162</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>[D]8163</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>D 8164</td><td>FROM (FNC 78), TO (FNC 79) instructions:
Specification of transfer points</td><td>–</td><td>–</td><td>M8164</td><td>–</td><td>–</td><td>✓*1</td><td>–</td><td>✓</td></tr>
<tr><td>[D]8165</td><td rowspan="4">Not used</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>[D]8166</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>[D]8167</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>[D]8168</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>[D]8169</td><td>Access restriction status by 2nd keyword
<table>
<tr><th rowspan="2">Present value</th><th rowspan="2">Access restriction status</th><th colspan="2">Program</th><th rowspan="2">Monitoring</th><th rowspan="2">Present value change</th></tr>
<tr><th>Read</th><th>Write</th></tr>
<tr><td>H0000</td><td>2nd keyword is not set.</td><td>✓*2</td><td>✓*2</td><td>✓*2</td><td>✓*2</td></tr>
<tr><td>H0010</td><td>Write protection</td><td>✓</td><td>–</td><td>✓</td><td>✓</td></tr>
<tr><td>H0011</td><td>Read / write protection</td><td>–</td><td>–</td><td>✓</td><td>✓</td></tr>
<tr><td>H0012</td><td>All online operation protection</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>H0020</td><td>Keyword cancel</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td></tr>
</table></td><td>✓</td><td>✓*1</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
</table>

*1. Available in Ver. 2.00 or later.

*2. The accessibility is restricted depending on the keyword setting status.

<table>
<tr><td colspan="10">N:N Network (setting) (Refer to Data Communication Edition for details.)</td></tr>
<tr><td>[D]8170</td><td rowspan="3">Not used</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>[D]8171</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>[D]8172</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>[D]8173</td><td>Station number</td><td>✓</td><td>✓</td><td>–</td><td>✓</td><td>✓</td><td>✓*3</td><td>✓</td><td>✓</td></tr>
<tr><td>[D]8174</td><td>Total number of slave stations</td><td>✓</td><td>✓</td><td>–</td><td>✓</td><td>✓</td><td>✓*3</td><td>✓</td><td>✓</td></tr>
<tr><td>[D]8175</td><td>Refresh range</td><td>✓</td><td>✓</td><td>–</td><td>✓</td><td>✓</td><td>✓*3</td><td>✓</td><td>✓</td></tr>
<tr><td>D 8176</td><td>Station number setting</td><td>✓</td><td>✓</td><td rowspan="5">M8038</td><td>✓</td><td>✓</td><td>✓*3</td><td>✓</td><td>✓</td></tr>
<tr><td>D 8177</td><td>Total slave station number setting</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓*3</td><td>✓</td><td>✓</td></tr>
<tr><td>D 8178</td><td>Refresh range setting</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓*3</td><td>✓</td><td>✓</td></tr>
<tr><td>D 8179</td><td>Retry count setting</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓*3</td><td>✓</td><td>✓</td></tr>
<tr><td>D 8180</td><td>Comms time-out setting</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓*3</td><td>✓</td><td>✓</td></tr>
<tr><td>[D]8181</td><td>Not used</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
</table>

*3. Available in Ver. 2.00 or later.

Number and name	Content of register	Applicable model							
		FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Index Register Z1 to Z7 and V1 to V7									
[D]8182	Value of Z1 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8183	Value of V1 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8184	Value of Z2 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8185	Value of V2 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8186	Value of Z3 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8187	Value of V3 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8188	Value of Z4 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8189	Value of V4 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8190	Value of Z5 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8191	Value of V5 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8192	Value of Z6 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8193	Value of V6 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8194	Value of Z7 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8195	Value of V7 register	✓	✓	–	✓	✓	✓	✓	✓
[D]8196	Not used	–	–	–	–	–	–	–	–
[D]8197		–	–	–	–	–	–	–	–
[D]8198		–	–	–	–	–	–	–	–
[D]8199		–	–	–	–	–	–	–	–
N:N Network (monitoring) (Refer to Data Communication Edition for details.)									
[D]8200	Not used	–	–	–	–	–	–	–	–
[D]8201*1	Current link scan time	✓	✓	–	(D201)	✓	✓*2	✓	✓
[D]8202*1	Maximum link scan time	✓	✓	–	(D202)	✓	✓*2	✓	✓
[D]8203*1	Number of communication error at master station	✓	✓	M8183 to M8191	(D203)	✓	✓*2	✓	✓
[D]8204*1	Number of communication error at slave station No.1	✓	✓		(D204)	✓	✓*2	✓	✓
[D]8205*1	Number of communication error at slave station No.2	✓	✓		(D205)	✓	✓*2	✓	✓
[D]8206*1	Number of communication error at slave station No.3	✓	✓		(D206)	✓	✓*2	✓	✓
[D]8207*1	Number of communication error at slave station No.4	✓	✓		(D207)	✓	✓*2	✓	✓
[D]8208*1	Number of communication error at slave station No.5	✓	✓		(D208)	✓	✓*2	✓	✓
[D]8209*1	Number of communication error at slave station No.6	✓	✓		(D209)	✓	✓*2	✓	✓
[D]8210*1	Number of communication error at slave station No.7	✓	✓		(D210)	✓	✓*2	✓	✓
[D]8211*1	Code of communication error at master station	✓	✓		(D211)	✓	✓*2	✓	✓
[D]8212*1	Code of communication error at slave station No.1	✓	✓		(D212)	✓	✓*2	✓	✓
[D]8213*1	Code of communication error at slave station No.2	✓	✓		(D213)	✓	✓*2	✓	✓
[D]8214*1	Code of communication error at slave station No.3	✓	✓		(D214)	✓	✓*2	✓	✓
[D]8215*1	Code of communication error at slave station No.4	✓	✓		(D215)	✓	✓*2	✓	✓
[D]8216*1	Code of communication error at slave station No.5	✓	✓		(D216)	✓	✓*2	✓	✓
[D]8217*1	Code of communication error at slave station No.6	✓	✓		(D217)	✓	✓*2	✓	✓
[D]8218*1	Code of communication error at slave station No.7	✓	✓		(D218)	✓	✓*2	✓	✓
[D]8219 to [D]8259	Not used	–	–	–	–	–	–	–	–

*1. In FX1S PLCs, use numbers shown inside parentheses.

*2. Available in Ver. 2.00 or later.

Number and name	Content of register	Applicable model: FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Analog Special Adapter (Refer to Subsection 36.2.17 for applicability of each analog special adapter.)									
D 8260 to D 8269	1st special adapter*1	✓	✓*2		–	–	–	–	–
D 8270 to D 8279	2nd special adapter*1	✓	✓*2		–	–	–	–	–
D 8280 to D 8289	3rd special adapter*1	✓	✓*2		–	–	–	–	–
D 8290 to D 8299	4th special adapter*1	✓	✓*2		–	–	–	–	–

*1. Count the number of connected FX3U-4AD-ADP, FX3U-4DA-ADP, FX3U-4AD-TC-ADP, and FX3U-4AD-PT-ADP units from the main unit..

*2. Available in Ver. 1.20 or later.

Number and name	Content of register	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Display Module Function FX3U-7DM (Refer to Hardware Edition Manual of PLC main unit used for details.)									
D 8300	Control device (D) for display module Default: K-1	✓	✓	–	–	–	–	–	–
D 8301	Control device (M) for display module Default: K-1	✓	✓	–	–	–	–	–	–
[D]8302*3	Language display setting Japanese: K0 English: Other than K0	✓	✓	–	–	–	–	–	–
[D]8303	LCD contrast setting value Default: K0	✓	✓	–	–	–	–	–	–
[D]8304	Not used	–	–	–	–	–	–	–	–
[D]8305	Not used	–	–	–	–	–	–	–	–
[D]8306	Not used	–	–	–	–	–	–	–	–
[D]8307	Not used	–	–	–	–	–	–	–	–
[D]8308	Not used	–	–	–	–	–	–	–	–
[D]8309	Not used	–	–	–	–	–	–	–	–

*3. Latch (battery backed) device

Number and name	Content of register	FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
RND (FNC184)									
[D]8310 / [D]8311	Lower / Upper: RND (FNC184) instruction: Data for generating random number Default: K1	✓	✓	–	–	–	–	–	–
Syntax, Circuit, Operation, or Unconnected I/O Designation Error Step Number (Refer to Chapter 37 for details.)									
D 8312 / D 8313	Lower / Upper: Operation error step number latched (32-bit)	✓	✓	M8068	–	–	–	–	–
[D]8314*4 / [D]8315*4	Lower / Upper: Error step number of M8065 to M8067 (32-bit)	✓	✓	M8065 to M8067	–	–	–	–	–
[D]8316 / [D]8317	Lower / Upper: Step number of instruction specifying an unconnected I/O number (directly or indirectly using index register)	✓	✓	M8316	–	–	–	–	–
[D]8318	BFM initialization function: Error unit number	✓	✓*5	M8318	–	–	–	–	–
[D]8319	BFM initialization function: Error BFM number	✓	✓*5	M8318	–	–	–	–	–
[D]8320 to [D]8328	Not used	–	–	–	–	–	–	–	–

*4. Cleared when PLC switches from STOP to RUN.

*5. Available in Ver. 2.20 or later.

Number and name	Content of register		Applicable model: FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Timing Clock (Refer to Section 24.3 for details.) and Positioning [FX3U and FX3UC PLCs] (Refer to Positioning Control Edition for details.)										
[D]8329	Not used		–	–	–	–	–	–	–	–
[D]8330	DUTY (FNC186) instruction: Scan counting for timing clock output 1		✓	✓*1	–	–	–	–	–	–
[D]8331	DUTY (FNC186) instruction: Scan counting for timing clock output 2		✓	✓*1	–	–	–	–	–	–
[D]8332	DUTY (FNC186) instruction: Scan counting for timing clock output 3		✓	✓*1	–	–	–	–	–	–
[D]8333	DUTY (FNC186) instruction: Scan counting for timing clock output 4		✓	✓*1	–	–	–	–	–	–
[D]8334	DUTY (FNC186) instruction: Scan counting for timing clock output 5		✓	✓*1	–	–	–	–	–	–
D 8336	DVIT (FNC151) instruction: Specification of interrupt input		✓	✓*2	M8336	–	–	–	–	–
[D]8337 to [D]8339	Not used		–	–	–	–	–	–	–	–
D 8340	Lower	[Y000] Current value register Default: 0	✓	✓	–	–	–	–	–	–
D 8341	Upper		✓	✓						
D 8342	[Y000] Bias speed Default: 0		✓	✓	–	–	–	–	–	–
D 8343	Lower	[Y000] Maximum speed Default: 100000	✓	✓	–	–	–	–	–	–
D 8344	Upper		✓	✓						
D 8345	[Y000] Creep speed Default: 1000		✓	✓	–	–	–	–	–	–
D 8346	Lower	[Y000] Zero return speed Default: 50000	✓	✓	–	–	–	–	–	–
D 8347	Upper		✓	✓						
D 8348	[Y000] Acceleration time Default: 100		✓	✓	–	–	–	–	–	–
D 8349	[Y000] Deceleration time Default: 100		✓	✓	–	–	–	–	–	–
D 8350	Lower	[Y001] Current value register Default: 0	✓	✓	–	–	–	–	–	–
D 8351	Upper		✓	✓						
D 8352	[Y001] Bias speed Default: 0		✓	✓	–	–	–	–	–	–
D 8353	Lower	[Y001] Maximum speed Default: 100000	✓	✓	–	–	–	–	–	–
D 8354	Upper		✓	✓						
D 8355	[Y001] Creep speed Default: 1000		✓	✓	–	–	–	–	–	–
D 8356	Lower	[Y001] Zero return speed Default: 50000	✓	✓	–	–	–	–	–	–
D 8357	Upper		✓	✓						
D 8358	[Y001] Acceleration time Default: 100		✓	✓	–	–	–	–	–	–
D 8359	[Y001] Deceleration time Default: 100		✓	✓	–	–	–	–	–	–

*1. Available in Ver. 2.20 or later.

*2. Available in Ver. 1.30 or later.

Number and name	Content of register		Applicable model							
			FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Timing Clock (Refer to Section 24.3 for details.) and Positioning [FX3U and FX3UC PLCs] (Refer to Positioning Control Edition for details.)										
D 8360	Lower	[Y002] Current value register Default: 0	✓	✓	–	–	–	–	–	–
D 8361	Upper		✓	✓						
D 8362	[Y002] Bias speed Default: 0		✓	✓	–	–	–	–	–	–
D 8363	Lower	[Y002] Maximum speed Default: 100000	✓	✓	–	–	–	–	–	–
D 8364	Upper		✓	✓						
D 8365	[Y002] Creep speed Default: 1000		✓	✓	–	–	–	–	–	–
D 8366	Lower	[Y002] Zero return speed Default: 50000	✓	✓	–	–	–	–	–	–
D 8367	Upper		✓	✓						
D 8368	[Y002] Acceleration time Default: 100		✓	✓	–	–	–	–	–	–
D 8369	[Y002] Deceleration time Default: 100		✓	✓	–	–	–	–	–	–
D 8370	Lower	[Y003] Current value register Default: 0	✓*1	–	–	–	–	–	–	–
D 8371	Upper			–	–	–	–	–	–	–
D 8372	[Y003] Bias speed Default: 0		✓*1	–	–	–	–	–	–	–
D 8373	Lower	[Y003] Maximum speed Default: 100000	✓*1	–	–	–	–	–	–	–
D 8374	Upper			–	–	–	–	–	–	–
D 8375	[Y003] Creep speed Default: 1000		✓*1	–	–	–	–	–	–	–
D 8376	Lower	[Y003] Zero return speed Default: 50000	✓*1	–	–	–	–	–	–	–
D 8377	Upper			–	–	–	–	–	–	–
D 8378	[Y003] Acceleration time Default: 100		✓*1	–	–	–	–	–	–	–
D 8379	[Y003] Deceleration time Default: 100		✓*1	–	–	–	–	–	–	–
[D]8380 to [D]8392	Not used			–	–	–	–	–	–	–

*1. Available only when two FX3U-2HSY-ADP adapters are connected to an FX3U PLC.

Number and name	Content of register		FX3U	FX3UC	Corresponding special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Interrupt Program (Refer to Chapter 35 for details.)										
D 8393	Delay time →**Refer to Section 35.4.**		✓	✓	M8393	–	–	–	–	–
[D]8394	Not used		–	–	–	–	–	–	–	–
[D]8395			–	–	–	–	–	–	–	–
[D]8396			–	–	–	–	–	–	–	–
[D]8397			–	–	–	–	–	–	–	–
Ring Counter										
D 8398	Lower	Up-operation ring counter of 0 to 2,147,483,647 (in 1-ms units, 32-bit)*2	✓	✓	M8398	–	–	–	–	–
D 8399	Upper									

*2. 1ms ring counter (D8399, D8398) will operate after M8398 turns ON.

Number and name	Content of register	Applicable model							
		FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
RS2 (FNC 87) [ch1] (Refer to Data Communication Edition for details.)									
D 8400	RS2 (FNC 87) [ch1] Communication format setting	✓	✓	–	–	–	–	–	–
[D]8401	Not used	–	–	–	–	–	–	–	–
[D]8402[*1]	RS2 (FNC 87) [ch1] Remaining points of transmit data	✓	✓	M8402	–	–	–	–	–
[D]8403[*1]	RS2 (FNC 87) [ch1] Monitoring receive data points	✓	✓	M8403	–	–	–	–	–
[D]8404	Not used	–	–	–	–	–	–	–	–
[D]8405	Communication parameter display [ch1]	✓	✓	–	–	–	–	–	–
[D]8406	Not used	–	–	–	–	–	–	–	–
[D]8407		–	–	–	–	–	–	–	–
[D]8408		–	–	–	–	–	–	–	–
D 8409	RS2 (FNC 87) [ch1] Time-out time setting	✓	✓	–	–	–	–	–	–
D 8410	RS2 (FNC 87) [ch1] Header 1 and 2 <Default: STX>	✓	✓	–	–	–	–	–	–
D 8411	RS2 (FNC 87) [ch1] Header 3 and 4	✓	✓	–	–	–	–	–	–
D 8412	RS2 (FNC 87) [ch1] Terminator 1 and 2 <Default: ETX>	✓	✓	–	–	–	–	–	–
D 8413	RS2 (FNC 87) [ch1] Terminator 3 and 4	✓	✓	–	–	–	–	–	–
[D]8414	RS2 (FNC 87) [ch1] Receive sum (received data)	✓	✓	–	–	–	–	–	–
[D]8415	RS2 (FNC 87) [ch1] Receive sum (calculated result)	✓	✓	–	–	–	–	–	–
[D]8416	RS2 (FNC 87) [ch1] Send sum	✓	✓	–	–	–	–	–	–
[D]8417	Not used	–	–	–	–	–	–	–	–
[D]8418		–	–	–	–	–	–	–	–
[D]8419	Operation mode display [ch1]	✓	✓	–	–	–	–	–	–
RS2 (FNC 87) [ch2] and Computer Link [ch2] (Refer to Data Communication Edition for details.)									
D 8420	RS2 (FNC 87) [ch2] Communication format setting	✓	✓	–	–	–	–	–	–
D 8421	Computer link [ch2] Station number setting	✓	✓	–	–	–	–	–	–
[D]8422[*1]	RS2 (FNC 87) [ch2] Remaining points of transmit data	✓	✓	M8422	–	–	–	–	–
[D]8423[*1]	RS2 (FNC 87) [ch2] Monitoring receive data points	✓	✓	M8423	–	–	–	–	–
[D]8424	Not used	–	–	–	–	–	–	–	–
[D]8425	Communication parameter display [ch2]	✓	✓	–	–	–	–	–	–
[D]8426	Not used	–	–	–	–	–	–	–	–
D 8427	Computer link [ch2] Specification of on-demand head device register	✓	✓	M8426 to M8429	–	–	–	–	–
D 8428	Computer link [ch2] Specification of on-demand data length register	✓	✓		–	–	–	–	–
D 8429	RS2 (FNC 87) [ch2], computer link [ch2] Time-out time setting	✓	✓		–	–	–	–	–
D 8430	RS2 (FNC 87) [ch2] Header 1 and 2 <Default: STX>	✓	✓	–	–	–	–	–	–
D 8431	RS2 (FNC 87) [ch2] Header 3 and 4	✓	✓	–	–	–	–	–	–
D 8432	RS2 (FNC 87) [ch2] Terminator 1 and 2 <Default: ETX>	✓	✓	–	–	–	–	–	–
D 8433	RS2 (FNC 87) [ch2] Terminator 3 and 4	✓	✓	–	–	–	–	–	–
[D]8434	RS2 (FNC 87) [ch2] Receive sum (received data)	✓	✓	–	–	–	–	–	–
[D]8435	RS2 (FNC 87) [ch2] Receive sum (calculated result)	✓	✓	–	–	–	–	–	–
[D]8436	RS2 (FNC 87) [ch2] Send sum	✓	✓	–	–	–	–	–	–
[D]8437	Not used	–	–	–	–	–	–	–	–

*1. Cleared when PLC switches from RUN to STOP.

Number and name	Content of register	Applicable model							
		FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Error Detection (Refer to Chapter 37 for details.)									
[D]8438	Error code for serial communication error 2 [ch2]	✓	✓	M8438	–	–	–	–	–
RS2 (FNC 87) [ch2] and Computer Link [ch2] (Refer to Data Communication Edition for details.)									
[D]8439	Operation mode display [ch2]	✓	✓	–	–	–	–	–	–
Error Detection (Refer to Chapter 37 for details.)									
[D]8440 to [D]8448	Not used	–	–	–	–	–	–	–	–
[D]8449	Special block error code	✓	✓*1	M8449	–	–	–	–	–
[D]8450 to [D]8459	Not used	–	–	–	–	–	–	–	–

*1. Available in Ver. 2.20 or later.

Number and name	Content of register	FX3U	FX3UC	Correspond-ing special device	FX1S	FX1N	FX2N	FX1NC	FX2NC
Positioning [FX3U and FX3UC PLCs] (Refer to Positioning Control Edition for details.)									
[D]8460 to [D]8463	Not used	✓	✓	–	–	–	–	–	–
D 8464	DSZR (FNC150) and ZRN (FNC156) instructions: [Y000] Clear signal device specification	✓	✓*2	M8464	–	–	–	–	–
D 8465	DSZR (FNC150) and ZRN (FNC156) instructions: [Y001] Clear signal device specification	✓	✓*2	M8465	–	–	–	–	–
D 8466	DSZR (FNC150) and ZRN (FNC156) instructions: [Y002] Clear signal device specification	✓	✓*2	M8466	–	–	–	–	–
D 8467	DSZR (FNC150) and ZRN (FNC156) instructions: [Y003] Clear signal device specification	✓*3	–	M8467	–	–	–	–	–
[D]8468 to [D]8511	Not used	–	–	–	–	–	–	–	–

*2. Available in Ver. 2.20 or later.

*3. Available only when two FX3U-2HSY-ADP adapters are connected to an FX3U PLC.

36.2 Supplement of Special Devices (M8000 - and D8000 -)

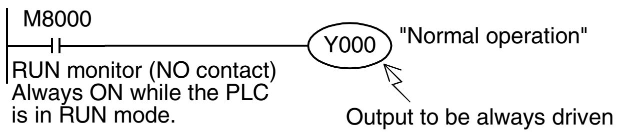
This section explains how to use the provided special devices to activate built-in PLC functions for additional program control.

36.2.1 RUN monitor and initial pulse [M8000 to M8003]

1. RUN monitor (M8000 and M8001)

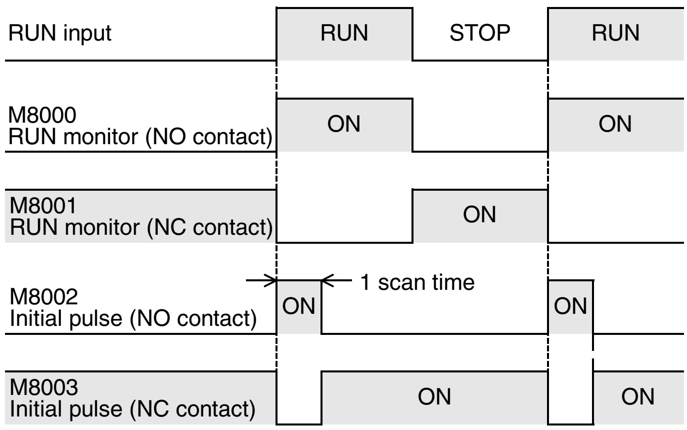
The RUN monitor (M8000 and M8001) may be used to continually drive an output during PLC "normal operation."

1) Example program



M8001 is always OFF while the PLC is in the RUN mode.

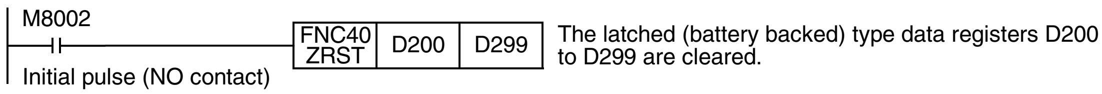
2) Flag operation timing



2. Initial pulse (M8002 and M8003)

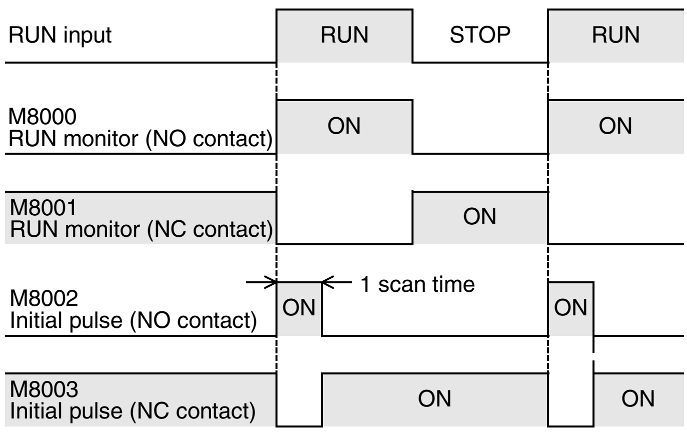
The initial pulse M8002 & M8003 is turned to ON or OFF respectively during the 1st scan of the PLC program. It can be utilized as an initial setting signal in a program for initializing the program, for writing a specified value, or for another purpose.

1) Example program



M8003 turns OFF momentarily (for only 1 scan time) when the PLC enters the RUN mode.

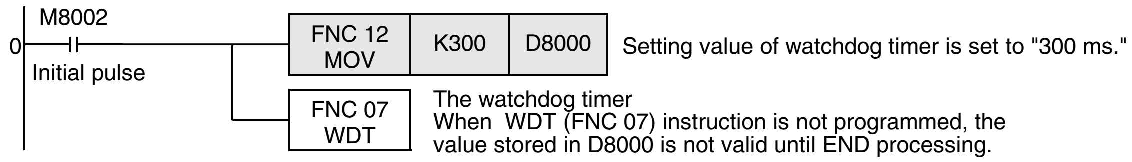
2) Flag operation timing



36.2.2 Watchdog timer [D8000]

The watchdog timer monitors the operation (scan) time of the PLC. When the operation is not completed within the specified time, ERROR (ERR) LED light turns on and all outputs are turned OFF.
When the power is initially turned ON, "200 ms" is transferred from the system to D8000 as the default value. For executing a program beyond 200 ms, the contents of D8000 must be changed by the user program.

1. Example program



2. When a watchdog timer error occurs

A watchdog timer error may occur in the following cases. Add the above program to somewhere near the first step or adjust the number of execution FROM/TO instructions at the same scan.

1) When using many special function units/blocks
   When many special function units/blocks (such as positioning, cam switches, link and analog) are used, buffer memory initial setting time becomes long at turning on the PLC, thus extending the operation time and allowing the possibility for a watchdog timer error to occur.
2) When executing many FROM/TO instructions at the same time
   When many FROM/TO instructions are executed or when many buffer memories are transferred, it extends the scan time, and a watchdog timer error may occur.
3) When using many high speed counters (software counters)
   When many high speed counters are programmed and high frequency is counted at the same time, it extends the scan time, and a watchdog timer error may occur.

3. How to reset the watchdog timer

The watchdog timer can be reset in the middle of a sequence program using WDT (FNC 07) instruction.
It is recommended to reset the watchdog timer by WDT (FNC 07) instruction when the scan time of a particular sequence program is extended or when many special function units/blocks are connected.

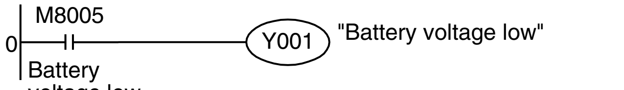
→ **For WDT (FNC 07) instruction, refer to Section 8.8.**

4. Cautions on changing the watchdog timer

The watchdog timer time can be set to a maximum of 32767 ms. However, CPU error detection is delayed when the watchdog timer time is extended.
It is recommended to use the default value (200 ms) when no problems are to be expected in operation.

36.2.3 Battery voltage low detection [M8005 and M8006]

This special device detects low voltage in the lithium battery for memory backup.
When the PLC detects low battery voltage, BATT (BAT) LED light turns on. The following program demonstrates it's use.



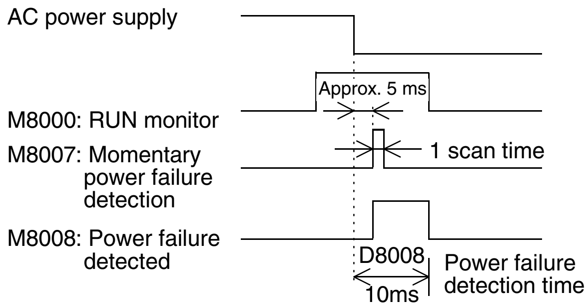
M8006 latches the battery voltage low.

36.2.4 Power failure detection time [D8008, M8008 and M8007]

1. FX3U PLC (AC power supply type)

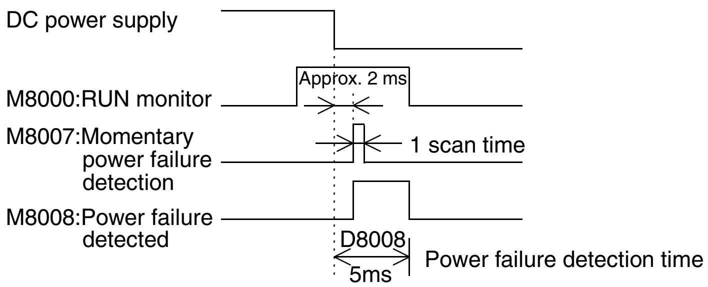
The table below shows the allowable momentary power failure time in FX3U PLCs (AC power supply type).

Supply voltage	Allowable momentary power failure time
100V AC system	10 ms
200V AC system	Setting range: 10 to 100 ms Set a value to D8008. Default: 10 ms



2. FX3UC PLC (DC power supply type)

The allowable momentary power failure time in the FX3UC PLC (DC power supply type) is 5 ms. Do not overwrite the power failure detection time in device D8008.



36.2.5 Scan time (monitor) [D8010 to D8012]

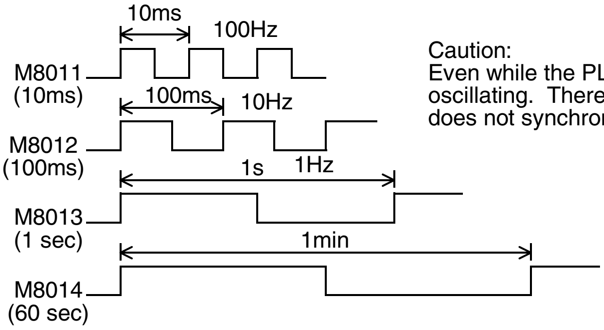
The present value, minimum value and maximum value of the PLC scan time (operation time) are stored in D8010, D8011 and D8012 respectively.
When using the constant scan mode, the values stored in these devices include the waiting time for the constant scan time.

D8010: Present value
D8011: Minimum value
D8012: Maximum value

The values stored in these devices can be monitored from peripheral equipment.

36.2.6 Internal clock [M8011 to M8014]

The PLC has the following four types of internal clocks which are always oscillating while the PLC power is ON.



Caution:
Even while the PLC is in the STOP mode, these clocks are always oscillating. Therefore, the rising edge of the RUN monitor (M8000) does not synchronize with the clock start timing.

36.2.7 Real time clock [M8015 to M8019 and D8013 to D8019]

1. Assignment of special auxiliary relays (M8015 to M8019) and special data registers (D8013 to D8019)

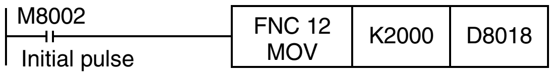
Number	Name	Operation and function
M8015	Clock stop and preset	When M8015 turns ON, the real time clock is stopped. At the edge from ON to OFF, the time from D8013 to D8019 is written to the PLC and the clock is started again.
M8016	Time read display is stopped	When M8016 turns ON, the time display is stopped (but RTC is continued).
M8017	±30 seconds correction	At the edge from OFF to ON, the RTC is set to the nearest minute. (When the second data is from 0 to 29, it is set to 0. When the second data is from 30 to 59, it is set to 0 and the minute data is incriminated by "1".)
M8018	Installation detection	This device is always ON.
M8019	Real time clock (RTC) error	When the data stored in special data registers is outside the allowable time, setting range this device turns ON.

Number	Name	Set value range	Operation and function
D8013	Second data	0 to 59	Use these devices for writing the initial value in time setting or read the present time. • D8018 (year data) can be changed over to the four-digit year mode. In the four-digit year mode, 1980 to 2079 can be displayed. • Clock accuracy: ±45 sec/month (at 25°C) • Leap year correction: Provided
D8014	Minute data	0 to 59	
D8015	Hour data	0 to 23	
D8016	Day data	1 to 31	
D8017	Month data	1 to 12	
D8018	Year data	00 to 99 (last two digits of year)	
D8019	Day-of-the-week data	0 to 6 (which corresponds to Sunday to Saturday)	

→ **For the real time clock setting method, refer to Subsection 36.2.8.**

2. Changeover of the year display (to the four-digit mode)

When changing the year data to the four-digit mode, add the following program.
D8018 is set to the four-digit year mode on the second ladder scan in RUN mode.



1) The PLC is usually operating in the two-digit year mode.
   When the above instruction is executed during RUN and "K2000 (fixed value)" is transferred to D8018 (year data), D8018 switches to the four-digit year mode.
2) Execute this program every time the PLC enters RUN.
   Only the year data display switches to four-digit mode when "K2000" is transferred. The actual time date is not affected.
3) In the four-digit year mode, the values "80" to "99" correspond to "1980" to "1999" and "00" to "79" correspond to "2000" to "2079".
   Examples: 80 = 1980, 99 = 1999, 00 = 2000, 79 = 2079
4) When connecting the data access unit FX-10DU-E, FX-20DU-E or FX-25DU-E, use the two-digit year mode.
   If the four-digit year mode is selected, the year data will not be displayed correctly.

36.2.8 How to set real time clock

The real time clock is set by the following method.

1. Method using the display module FX3U-7DM (FX3UC PLC built-in)

1) Scroll to "ClockMenu" by pressing [+] or [-] key on the MENU screen, and press [OK] key. The selection screen on the right is displayed.
Press [ESC] key to cancel the operation and return to "TOP screen".

```
> Current time
  Clock setting
```

2) Scroll to "Clock setting" by pressing [+] or [-] key.
Press [ESC] key to cancel the operation and return to "MENU screen."

```
  Current time
> Clock setting
```

3) Press [OK] key, and "Clock setting" screen on the right is displayed.
Press [ESC] key to cancel the operation and return to "selection screen."

```
Clock setting
    19. 8.1993 *
   7:11:19[Thu]
```

4) Change the flickering value by pressing [+] or [-] key, and press [OK] key to set the value. The items are set in the order "year → month → day → hour → minute → second."
When the last "second" value is decided by pressing [OK] key, the message "Current time is set" appears, and setting of the present time is completed.

```
Clock setting
   31. 1.2004 *
  23:59:59[Sat]
```

The default "Year" display is a 2-digit value indicating the Western calendar year.

Operation key	Contents of operation
ESC	Return to the previous setting item. When pressed while "year" data is flickering, "selection" screen is displayed.
–	Decreases a numeric value. A numeric value decreases at high speed when pressed for 1 second or more.
+	Increases a numeric value. A numeric value increases at high speed when pressed for 1 second or more.
OK	Shifts to the next set item. When pressed while "second" data is flickering, the message "Current time is set" appears.

5) Press [OK] or [ESC] key to return to "selection screen."
6) Press [ESC] key to return to "MENU screen."

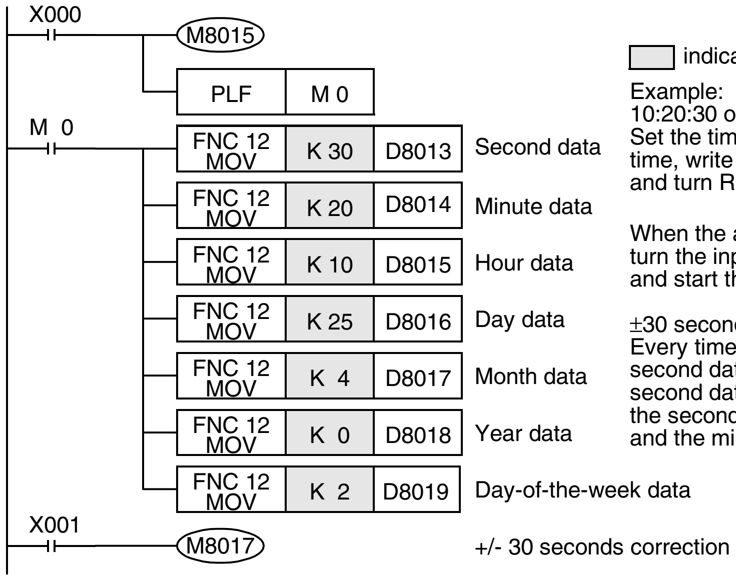
```
  Current time
> Clock setting
```

2. Method by program

1) Method using TWR (FNC167) instruction dedicated to time setting.

→ **For details, refer to Section 21.8.**

2) Method using M8015 and D8013 to D8019.
When not using TWR (FNC167) instruction dedicated to time setting, the following program can be used for time setting.



▭ indicates a set item.

Example:
10:20:30 on April 25 Tuesday in 2000
Set the time 2 to 3 minutes ahead of the actual time, write the program on the left to the PLC, and turn RUN X000 ON.

When the actual time reaches the input set time, turn the input switch X000 OFF to set the time and start the clock operation.

±30 seconds correction
Every time X001 is set from OFF to ON, the second data of the clock is corrected. (When the second data is from 0 to 29, it is set to 0. When the second data is from 30 to 59, it is set to 0 and the minute data is incremented by "1".)

Point

a) While M8015 is OFF, the registers for date and time cannot be changed.
Make sure to set M8015 to ON, and then input date and time.
b) When inputting the time, set the time several minutes ahead of the current time.
When the actual time reaches the set time, set M8015 to OFF from ON to make the new time valid and to begin the real time clock again.
c) If values indicating infeasible date and time are input (Example: February 30, 2006), the time cannot be set. Input the correct date and time.

3. Method by programming tool

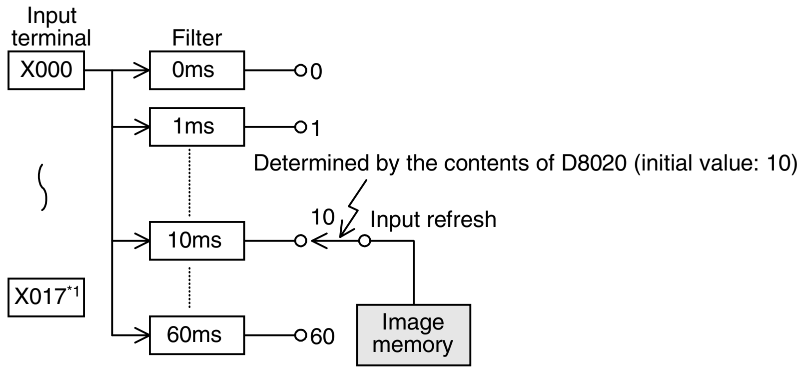
1) Starting the time change
Set M8015 to ON by forced ON/OFF.
2) By using the data register value change function in the device monitor of the programming tool, input the date and time (several minutes ahead) to each data register.
3) Finishing the time change
When the actual time reaches the input data, reset M8015 by forced ON/OFF.

Point

a) While M8015 is OFF, the registers for date and time cannot be changed.
Make sure to set M8015 to ON, and then input date and time.
b) When inputting the time, set the time several minutes ahead of the current time.
When the actual time reaches the set time, set M8015 to OFF from ON to make the new time valid and to begin the real time clock again.
c) If values indicating infeasible date and time are input (Example: February 30, 2006), the time cannot be set. Input the correct date and time.

36.2.9 Input filter adjustment [D8020]

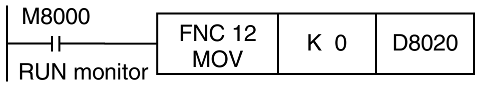
The inputs X000 to X017*1 have a digital filter circuit with a setting range of 0 to 60 ms. The digital filter setting value is set between 0 to 60 ms (in 1 ms steps) in special data register D8020. After PLC powers ON, D8020 is automatically set to K10 (10 ms).



*1. X000 to X007 in the FX3U-16M□

1. Program example for adjusting the input filter

When the program shown below is executed, the filter constant is changed to 0 ms.
Because the C-R filter is provided in the hardware, however, the filter constant is shown in the table below when "0" is specified.



Input number	Input filter value when "0" is set
X000 to X005	5μs*1
X006, X007	50μs
X010 to X017*2	200μs

*1. When setting the input filter to "5 μs" or when receiving pulses whose response frequency is 50 k to 100 kHz using high speed counters, perform the following:
- Input wiring length should be 5 m (16' 4") or less.
- Connect a bleeder resistor (1.5 kΩ, 1/2 W) to an input terminal. The load current of the open collector transistor output in the device on the other end should be 20 mA or more including the input current of the PLC.

*2. X000 to X007 in the FX3U-16M□

- The input filter constant can be changed as many times as need in the user program.
- This input filter adjustment described here is not required when using high speed counter, input interrupt, or pulse catch (M8170 to M8175) functions.

36.2.10 Battery [BATT (BAT)] LED OFF command [M8030]

1. Batteryless operation

When M8030 is set to ON, the battery LED does not turn ON even if the voltage in the battery for memory backup becomes low.
When the indication of "battery voltage low error" is not required or when the battery is removed, set M8030 to ON.
When the batteryless operation is required, however, do not use M8030 but refer to "2. Parameter setting in peripheral equipment" below.

2. Parameter setting in peripheral equipment

Specify "batteryless operation" mode in the parameter settings.

1) When the batteryless operation option is specified
   When "batteryless mode" is specified, the control to turn OFF the BATT (BAT) LED and initialization of the latch area for the devices shown below are automatically executed by the PLC system.
   - Auxiliary relay (M) - Counter (C) - State relay (S)
   - Data register (D) - Timer (T) - Extension data register (R)
2) Applicable programming tool
   Some programming tool versions do not support "batteryless mode." In such versions, input a sequence program to enable the batteryless operation as explained below.

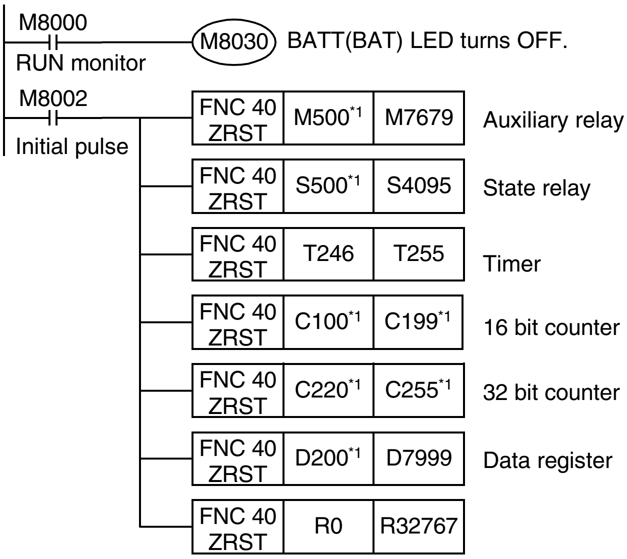
3. Conditions for batteryless operation

1) An FLROM (optional memory cassette) for program memory is installed so that programs are not erased.
2) The latch (battery backed) devices such as auxiliary relays and data registers are not used for control.
3) The sampling trace function is not used.
4) The real time clock function is not used.

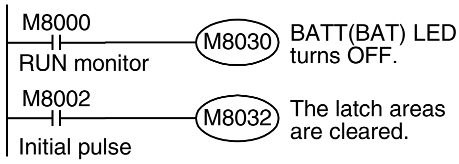
4. Example program for batteryless mode

When a parameter setting for "batteryless mode" is not available, create the sequence program shown below.

- Example program for clearing the memory backup area
  {when the latch (battery backed) ranges in the parameter settings are set to their initial values.}



or



*1. The device number is the default number of the latch (battery backed) device range limits in the parameter settings.
If the latch (battery backed) device numbers in the parameter settings are changed to other values, change the device numbers here in accordance with the changed latch ranges.

5. Cautions for communication setting devices (D8120, D8121 and D8129)

The special data register D8120 (communication format setting), D8121 (station number setting), and D8129 (time-out check time) are backed up by the battery.
When using the batteryless function, reset these special data registers once, and then transfer proper set values to them by program.
The communication conditions can be set in the parameter settings.
When the communication conditions are set in the parameter settings, the PLC transfers the parameter values to the above special data registers before operation. Thus it is recommended to set the communication conditions via the parameter settings.

36.2.11 Clear command [M8031 and M8032]

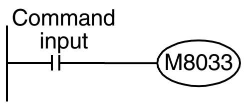
For all devices (image memory) in the PLC, the latch (battery backed) areas and non-latch areas can be cleared.
M8031 (non-latch memory all clear), M8032 (latch memory all clear)

Device number	Cleared devices
M8031 (non-latch memory all clear)	• Contact image of output relay (Y), general type auxiliary relay (M), and general type state relay (S) • Contact and coil of timer (T) • Contact, counting coil, and reset coil of general type counter (C) • Present value of general type data register (D) • Present value register of timer (T) • Present value register of general type counter (C)
M8032 (latch memory all clear)	• Contact image of latch (battery backed) type auxiliary relay (M) and latch (battery backed) type state relay (S) • Contact and coil of retentive type timer (T) • Contact, counting coil, and reset coil of latch (battery backed) type counter and high speed counter • Present value of latch (battery backed) type data register (D) • Present value register of retentive type timer (T) and 1-ms timer (T) • Present value register of latch (battery backed) type counter and high speed counter

36.2.12 Memory hold stop [M8033] (output hold in STOP mode)

When the special auxiliary relay M8033 is turned ON, the output status in the RUN mode is retained even if the PLC status switches from RUN to STOP.

1. Example program



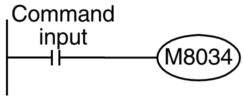
Outputs are retained even during STOP mode.

For example, when a heater is driven by the PLC, the PLC can be stopped while the heater and other equipment are kept driven, and then the PLC can be started again after program changes.

36.2.13 All outputs disable [M8034]

When M8034 is turned ON, the output memory is cleared. As a result, all actual output relay contacts are turned OFF and the PLC is operated in the image memory.

1. Program example

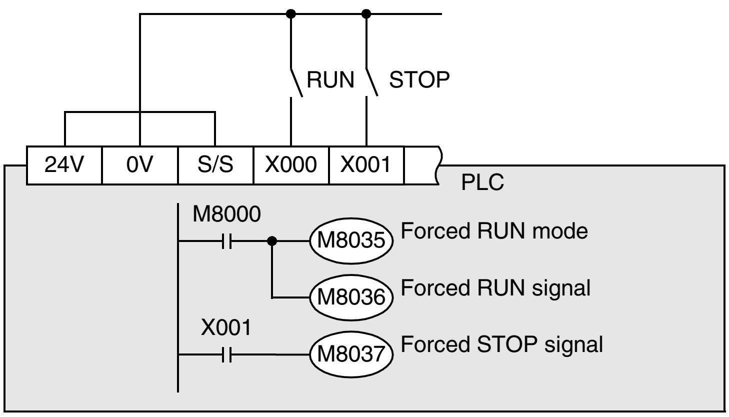


All outputs are disabled (image RUN status).

36.2.14 Individual operation for RUN/STOP input [M8035 to M8037]

When using external push button switches to control the PLC's RUN/STOP mode, operate the switches using the following procedure.
The PLC enters RUN mode by one-shot input of the RUN switch, while one-shot input of the STOP switch drives the STOP mode.

1. Program example



The figure on the left shows an example of FX3U PLC (sink input).

Input the above program in the PLC.

2. Setting method

1) Turn the built-in RUN/STOP switch to STOP.
2) Specify the RUN input switch, input (X) (X000 is specified in the above circuit diagram example.)
Make the external RUN/STOP input valid by specifying an input between X000 and X017*1 for the RUN input signal.
*1.X000 to X007 in the FX3U-16M□
a) Display the parameter setting in the programming tool
In GX Developer case, double-click [Parameter] - [PC parameter] in the project tree to display the dialog box.
Click "PLC system (1)" tab, and set "RUN terminal input."
b) Specify the input (X) number to switch from STOP mode to RUN mode.
3) Specify the STOP switch input (X)
Specify an arbitrary input terminal (actual I/O on the PLC) in the sequence program.
Refer to the above program.
4) Transfer the program and parameters to the PLC.
5) For the parameter settings to become valid, the PLC power must be turned from OFF to ON.

3. Cautions

1) When both RUN and STOP switches are pressed at the same time, priority is given to the STOP switch.
2) When the built-in RUN/STOP switch is turned to RUN, the PLC can be set to RUN mode.
However, when the STOP switch assigned to an arbitrary input is activated, the PLC will enter STOP mode. (Even if the built-in switch is turned to RUN, priority is given to the STOP command.)

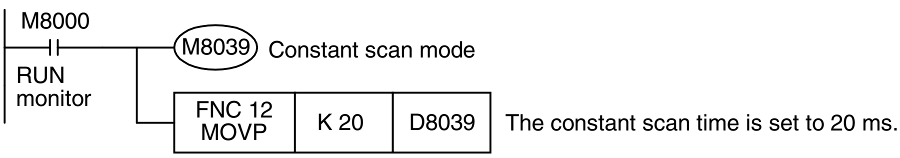
4. RUN/STOP command via the programming tool

1) Using the programming software for personal computer
   There is a remote RUN/STOP function in the programming software.
   By using the programming software, the PLC can be set to the RUN or STOP mode by giving a command from the personal computer.
2) Using any other programming tool
   When M8035 (forced RUN mode) and M8036 (forced RUN signal) are set to ON in the forced ON/OFF procedure, the PLC begins RUN mode.
   When M8037 (forced STOP signal) is set to ON, the PLC changes to STOP mode.
3) Even when the built-in RUN/STOP switch is on the RUN side of the PLC.
   The remote STOP command via the programming tool or M8037 (forced STOP signal) are valid.

36.2.15 Constant scan mode [M8039 and D8039]

When the special auxiliary relay M8039 is set to ON and a setting value for the constant scan time (in 1-ms units) is stored in special data register D8039, the scan time in the PLC does not become shorter than the value stored in D8039.
The PLC pauses for the remaining time when the operation ends earlier, and then returns to step 0.

1. Example program



2. Cautions

1) When using an instruction executed in synchronization with a scan
   a) When using an instruction executed in synchronization with a scan such as RAMP (FNC 67), HKY (FNC 71), SEGL (FNC 74), ARWS (FNC 75) and PR (FNC 77), it is recommended to use the constant scan mode or to use the instruction in a timer interrupt program.
   b) When using HKY (FNC 71) instruction
      It is necessary to use a scan time of 20 ms or more due to the response delays of the key input filter.
2) Scan time display (D8010 to D8012)
   The scan time specified in the constant scan time is included in the scan time display stored in D8010 to D8012.

36.2.16 State control in program with STL instruction/SFC chart [M8040]

When M8040 is ON, the state relay ON status is not transferred even if the transfer condition is satisfied.
The transfer between states is disabled and the output status remains in stopped State.

→ **For resetting outputs in a state, refer to Subsection 34.1.7.**

36.2.17 Analog special adapters [M8260 to M8299 and D8260 to D8299]

When analog special adapters are connected, operations and functions are assigned to the devices shown in the tables below in accordance with the number of connected analog special adapters.
Devices which cannot be written to are shaded in the "Operation and function" column.

→ **For details, refer to the manual of each product.**

1. Special auxiliary relays (M8260 to M8299)

Number	Operation and function			
	FX3U-4AD-ADP	FX3U-4DA-ADP	FX3U-4AD-PT-ADP	FX3U-4AD-TC-ADP
Applicable version	Ver.1.20 or later*1	Ver.1.20 or later*1	Ver.1.30 or later*1	Ver.1.30 or later*1
1st analog special adapter				
M 8260	Input mode switching Ch1	Output mode switching Ch1	Temperature unit selection	Temperature unit selection
M 8261	Input mode switching Ch2	Output mode switching Ch2	Not used	Type-K/-J switching
M 8262	Input mode switching Ch3	Output mode switching Ch3	Not used	Not used
M 8263	Input mode switching Ch4	Output mode switching Ch4	Not used	Not used
M 8264	Not used	Output hold mode cancel Ch1	Not used	Not used
M 8265	Not used	Output hold mode cancel Ch2	Not used	Not used
M 8266	Not used	Output hold mode cancel Ch3	Not used	Not used
M 8267	Not used	Output hold mode cancel Ch4	Not used	Not used
M 8268	Not used	Not used	Not used	Not used
M 8269	Not used	Not used	Not used	Not used
2nd analog special adapter				
M 8270	Input mode switching Ch1	Output mode switching Ch1	Temperature unit selection	Temperature unit selection
M 8271	Input mode switching Ch2	Output mode switching Ch2	Not used	Type-K/-J switching
M 8272	Input mode switching Ch3	Output mode switching Ch3	Not used	Not used
M 8273	Input mode switching Ch4	Output mode switching Ch4	Not used	Not used
M 8274	Not used	Output hold mode cancel Ch1	Not used	Not used
M 8275	Not used	Output hold mode cancel Ch2	Not used	Not used
M 8276	Not used	Output hold mode cancel Ch3	Not used	Not used
M 8277	Not used	Output hold mode cancel Ch4	Not used	Not used
M 8278	Not used	Not used	Not used	Not used
M 8279	Not used	Not used	Not used	Not used
3rd analog special adapter				
M 8280	Input mode switching Ch1	Output mode switching Ch1	Temperature unit selection	Temperature unit selection
M 8281	Input mode switching Ch2	Output mode switching Ch2	Not used	Type-K/-J switching
M 8282	Input mode switching Ch3	Output mode switching Ch3	Not used	Not used
M 8283	Input mode switching Ch4	Output mode switching Ch4	Not used	Not used
M 8284	Not used	Output hold mode cancel Ch1	Not used	Not used
M 8285	Not used	Output hold mode cancel Ch2	Not used	Not used
M 8286	Not used	Output hold mode cancel Ch3	Not used	Not used
M 8287	Not used	Output hold mode cancel Ch4	Not used	Not used
M 8288	Not used	Not used	Not used	Not used
M 8289	Not used	Not used	Not used	Not used
4th analog special adapter				
M 8290	Input mode switching Ch1	Output mode switching Ch1	Temperature unit selection	Temperature unit selection
M 8291	Input mode switching Ch2	Output mode switching Ch2	Not used	Type-K/-J switching
M 8292	Input mode switching Ch3	Output mode switching Ch3	Not used	Not used
M 8293	Input mode switching Ch4	Output mode switching Ch4	Not used	Not used
M 8294	Not used	Output hold mode cancel Ch1	Not used	Not used
M 8295	Not used	Output hold mode cancel Ch2	Not used	Not used
M 8296	Not used	Output hold mode cancel Ch3	Not used	Not used
M 8297	Not used	Output hold mode cancel Ch4	Not used	Not used
M 8298	Not used	Not used	Not used	Not used
M 8299	Not used	Not used	Not used	Not used

*1. FX3U PLCs support these operations and functions.

2. Special data registers (D8260 to D8299)

Number	Operation and function			
	FX3U-4AD-ADP	FX3U-4DA-ADP	FX3U-4AD-PT-ADP	FX3U-4AD-TC-ADP
Applicable version	Ver.1.20 or later*1	Ver.1.20 or later*1	Ver.1.30 or later*1	Ver.1.30 or later*1
1st analog special adapter				
D 8260	Input data Ch1	Output data Ch1	Measured temperature Ch1	Measured temperature Ch1
D 8261	Input data Ch2	Output data Ch2	Measured temperature Ch2	Measured temperature Ch2
D 8262	Input data Ch3	Output data Ch3	Measured temperature Ch3	Measured temperature Ch3
D 8263	Input data Ch4	Output data Ch4	Measured temperature Ch4	Measured temperature Ch4
D 8264	Number of averaging times for Ch1 (1 to 4095)	Not used	Number of averaging times for Ch1 (1 to 4095)	Number of averaging times for Ch1 (1 to 4095)
D 8265	Number of averaging times for Ch2 (1 to 4095)	Not used	Number of averaging times for Ch2 (1 to 4095)	Number of averaging times for Ch2 (1 to 4095)
D 8266	Number of averaging times for Ch3 (1 to 4095)	Not used	Number of averaging times for Ch3 (1 to 4095)	Number of averaging times for Ch3 (1 to 4095)
D 8267	Number of averaging times for Ch4 (1 to 4095)	Not used	Number of averaging times for Ch4 (1 to 4095)	Number of averaging times for Ch4 (1 to 4095)
D 8268	Error status	Error status	Error status	Error status
D 8269	Model code: K1	Model code: K2	Model code: K20	Model code: K10
2nd analog special adapter				
D 8270	Input data Ch1	Output data Ch1	Measured temperature Ch1	Measured temperature Ch1
D 8271	Input data Ch2	Output data Ch2	Measured temperature Ch2	Measured temperature Ch2
D 8272	Input data Ch3	Output data Ch3	Measured temperature Ch3	Measured temperature Ch3
D 8273	Input data Ch4	Output data Ch4	Measured temperature Ch4	Measured temperature Ch4
D 8274	Number of averaging times for Ch1 (1 to 4095)	Not used	Number of averaging times for Ch1 (1 to 4095)	Number of averaging times for Ch1 (1 to 4095)
D 8275	Number of averaging times for Ch2 (1 to 4095)	Not used	Number of averaging times for Ch2 (1 to 4095)	Number of averaging times for Ch2 (1 to 4095)
D 8276	Number of averaging times for Ch3 (1 to 4095)	Not used	Number of averaging times for Ch3 (1 to 4095)	Number of averaging times for Ch3 (1 to 4095)
D 8277	Number of averaging times for Ch4 (1 to 4095)	Not used	Number of averaging times for Ch4 (1 to 4095)	Number of averaging times for Ch4 (1 to 4095)
D 8278	Error status	Error status	Error status	Error status
D 8279	Model code: K1	Model code: K2	Model code: K20	Model code: K10
3rd analog special adapter				
D 8280	Input data Ch1	Output data Ch1	Measured temperature Ch1	Measured temperature Ch1
D 8281	Input data Ch2	Output data Ch2	Measured temperature Ch2	Measured temperature Ch2
D 8282	Input data Ch3	Output data Ch3	Measured temperature Ch3	Measured temperature Ch3
D 8283	Input data Ch4	Output data Ch4	Measured temperature Ch4	Measured temperature Ch4
D 8284	Number of averaging times for Ch1 (1 to 4095)	Not used	Number of averaging times for Ch1 (1 to 4095)	Number of averaging times for Ch1 (1 to 4095)
D 8285	Number of averaging times for Ch2 (1 to 4095)	Not used	Number of averaging times for Ch2 (1 to 4095)	Number of averaging times for Ch2 (1 to 4095)
D 8286	Number of averaging times for Ch3 (1 to 4095)	Not used	Number of averaging times for Ch3 (1 to 4095)	Number of averaging times for Ch3 (1 to 4095)
D 8287	Number of averaging times for Ch4 (1 to 4095)	Not used	Number of averaging times for Ch4 (1 to 4095)	Number of averaging times for Ch4 (1 to 4095)
D 8288	Error status	Error status	Error status	Error status
D 8289	Model code: K1	Model code: K2	Model code: K20	Model code: K10
4th analog special adapter				
D 8290	Input data Ch1	Output data Ch1	Measured temperature Ch1	Measured temperature Ch1
D 8291	Input data Ch2	Output data Ch2	Measured temperature Ch2	Measured temperature Ch2
D 8292	Input data Ch3	Output data Ch3	Measured temperature Ch3	Measured temperature Ch3
D 8293	Input data Ch4	Output data Ch4	Measured temperature Ch4	Measured temperature Ch4
D 8294	Number of averaging times for Ch1 (1 to 4095)	Not used	Number of averaging times for Ch1 (1 to 4095)	Number of averaging times for Ch1 (1 to 4095)
D 8295	Number of averaging times for Ch2 (1 to 4095)	Not used	Number of averaging times for Ch2 (1 to 4095)	Number of averaging times for Ch2 (1 to 4095)
D 8296	Number of averaging times for Ch3 (1 to 4095)	Not used	Number of averaging times for Ch3 (1 to 4095)	Number of averaging times for Ch3 (1 to 4095)
D 8297	Number of averaging times for Ch4 (1 to 4095)	Not used	Number of averaging times for Ch4 (1 to 4095)	Number of averaging times for Ch4 (1 to 4095)
D 8298	Error status	Error status	Error status	Error status
D 8299	Model code: K1	Model code: K2	Model code: K20	Model code: K10

*1. FX3U PLCs support these operations and functions.

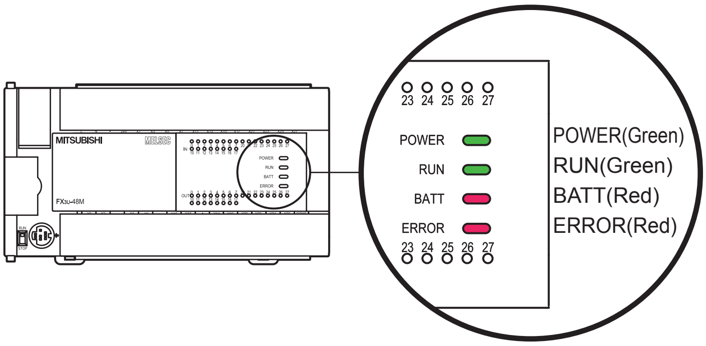
37. Error Check Method and Error Code List

When an error occurs while the program is being executed, troubleshoot the cause of the error in accordance with this chapter.
For details of errors, refer to the Data Communication Edition and the Hardware Edition of the PLC main unit.

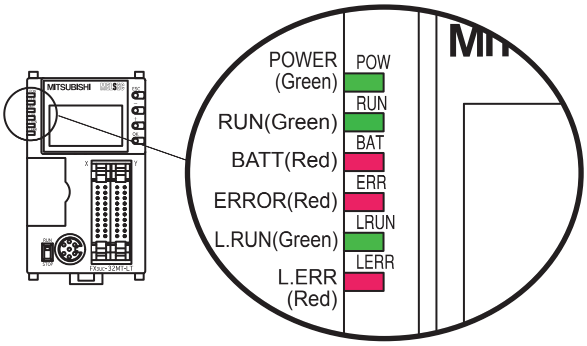
37.1 States and Colors of LEDs PLC Operation Status

When an error has occurred, the PLC state can be checked by the LED status lights on the PLC.

FX3U PLC



FX3UC PLC



37.1.1 POWER LED [lit, flickering or unlit] [FX3U/FX3UC]

LED status	PLC status	Action
Lit	The voltage is correctly supplied to the power terminal.	The power supply is normal.
Flickering	The following status occurred: • The voltage or current is incorrectly supplied to the power terminal. • The wiring is incorrect. • There is fault inside the PLC.	• Check the power supply voltage. • Disconnect cables except the power cable, turn the PLC power ON again, and check whether the status is changed. If the status is not improved, consult a Mitsubishi Electric Distributor.
Unlit	The following status occurred: • The power is OFF. • The voltage is incorrectly supplied to the power terminal. • There is wire breakage in the power cable.	When the power is not OFF, check the power supply and power line route. When the power is correctly supplied, consult a Mitsubishi Electric Distributor.

37.1.2 RUN LED [lit or unlit] [FX3U/FX3UC]

LED status	PLC status	Action
Lit	Sequence program is executing.	The PLC operation status is indicated. This LED is not lit depending on the ERROR LED status. (Refer to Subsection 37.1.4.)
Unlit	Sequence program is stopped.	

37.1.3 BATT LED [lit or unlit] [FX3U/FX3UC]

LED status	PLC status	Action
Lit	The battery voltage is low.	Replace the battery as soon as possible. (Refer to FX3U/FX3UC Hardware edition.)
Unlit	The battery voltage exceeds the value set in D8006.	The battery is normal.

37.1.4 ERROR LED [lit, flickering or unlit] [FX3U/FX3UC]

LED status	PLC status	Action
Lit	A watchdog timer error has occurred, or the hardware of the PLC may be damaged.	1) Change the PLC mode to STOP, and turn ON the PLC power again. When the ERROR LED is off, a watchdog timer error occurred. Take one of the following actions: - Review the program, and make sure that the maximum value (D8012) of the scan time is not larger than the set value (D8000) of the watchdog timer. - Make sure that an input used for input interrupt or pulse catch does not abnormally turn ON and OFF several times in one scan time. - Make sure that the frequency of the pulse (duty: 50%) input to a high speed counter is within the specifications range. - Adding WDT instruction Use two or more WDT instructions in a program so that the watchdog timer is reset several times in one scan time. - Change the set value of the watchdog timer Change the set value (D8000) of the watchdog timer in a program so that it is larger than the maximum value (D8012) of the scan time. 2) Remove the PLC, and connect another power supply to the PLC. If the ERROR LED is off, the cause of the error may be noise. Examine the following action: - Check the wiring for grounding, and then review the wiring route and installation location. - Add a noise filter in the power line. 3) If the ERROR LED is not off even after the step 1) or 2), consult a Mitsubishi Electric Distributor.
Flickering	Either of the following errors occur in PLC: • Parameter error • Syntax error • Circuit error	Execute PLC diagnostics and program check by programming tool. For instructions, refer to Section 37.4.
Unlit	Error which stops PLC has not occurred.	If PLC operation is a failure, execute the PLC diagnostics or program check by programming tool. An I/O configuration error, serial communication error, or operation error may occur.

31 FNC275-FNC279 Data Transfer 3
32 FNC280-FNC289 High Speed Processing 2
33 FNC290-FNC299 Extension File Register
34 SFC•STL Programming
35 Interrupt Function
36 Special Device
37 Error Code
A Version Up Information
B Execution Times
C Applicable Instruction List

37.1.5 L RUN LED [FX3UC]

Mode	LED status	PLC status	Action
ONLINE	Lit	Data link is executing	—
	Unlit	Data link is stopped	• Take action according to the L ERR LED status.
CONFIG	Lit	Data link is executing	—
	Unlit	Data link is stopped	• Take action according to the L ERR LED status.
TEST	Lit	The self-loopback test is normally finished.	—
	Unlit	The self-loopback test is abnormally finished. (This LED is off while the self-loopback test is executing.)	• Make sure that the power is correctly supplied to the PLC. • If the L RUN LED is not off even after the above check, consult a Mitsubishi Electric Distributor.

37.1.6 L ERR LED [FX3UC]

Mode	LED status	PLC status	Action
ONLINE	Lit	• Unit disconnected • Outside-control-range station error	• Securely connect the built-in master and a remote I/O unit. • Make sure that the connected remote I/O units are consistent with the detailed information on remote stations.
	Flickering	All stations are abnormal	• Securely connect the built-in master and a remote I/O unit. • Make sure that the connected remote I/O units are consistent with the detailed information on remote stations.
	Unlit	Data link is being normally executed	—
CONFIG	Lit	Used station numbers mismatch. (Remote stations are checked while the remote station information is edited.)	• Securely connect the built-in master and a remote I/O unit. • Make sure that the connected remote I/O units are consistent with the detailed information on remote stations.
	Flickering	All stations are abnormal. (Remote stations are checked while the remote station information is edited.)	
	Unlit	Data link is being normally executed	—
TEST	Lit	The self-loopback test is abnormally finished.	• Make sure that the power is correctly supplied to the PLC. • If the L RUN LED is on even after the above check, consult a Mitsubishi Electric Distributor.
	Unlit	The self-loopback test is normally finished. (This LED is off while the self-loopback test is executing.)	—

→ **For details, refer to the Hardware Edition of the PLC main unit.**

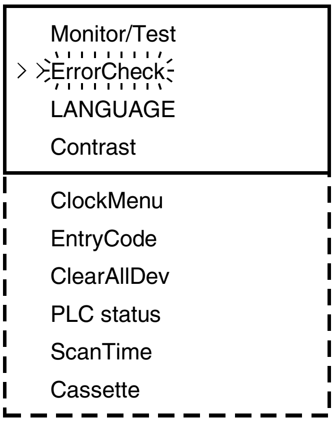
37.2 Error Code Check Method and Indication

37.2.1 Error code check method by display module (FX3U-7DM)

The error code can be checked by programming tool and FX3U-7DM (hereinafter called display module).
For the check method by programming tool, refer to the program check or PLC diagnostics described in the programming tool manual.

Error code check method by display module

1) Scroll to "ErrorCheck" by pressing [+] or [-] key on "MENU screen" (shown on the right figure).
For the menu system, refer to FX3U/FX3UC Hardware Edition.
On this menu screen, the operation keys are as shown below:



Operation key	Contents of operation
ESC	Return to "TOP screen".
-	Moves the cursor up. Moves the cursor at high speed when pressed for 1 second or more. When the cursor is located at the top, [-] key operation is invalid.
+	Moves the cursor down. Moves the cursor at high speed when pressed for 1 second or more. When the cursor is located at the bottom, [+] key operation is invalid.
OK	Selects a flickering item with the cursor.

2) Pressing [OK] key executes the error check and displays the result on "error display screen" (shown in the right figure).
Press [ESC] key to cancel the operation and return to "Top screen".

3) If two or more errors occur, press [+] or [-] key to changeover the page.

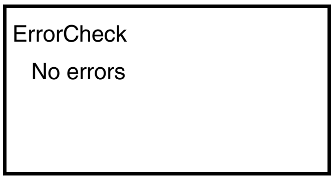
Operation key	Contents of operation	
ESC	Returns to "menu screen."	
-	When one or no error	Is invalid.
	When two or more errors	Displays the previous error display screen.
+	When one or no error	Is invalid.
	When two or more errors	Displays the next error display screen.
OK	Returns to "menu screen."	

Displayed contents

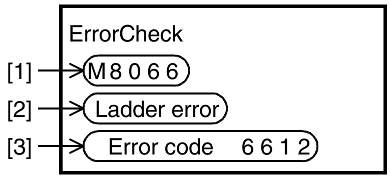
	Displayed contents
[1]	Error flag
[2]	Error name
[3]	Error code
[4]	Number of errors at same time (When two or more errors occur, this information displays.)

4) Press [ESC] key to cancel the operation and return to "menu screen."

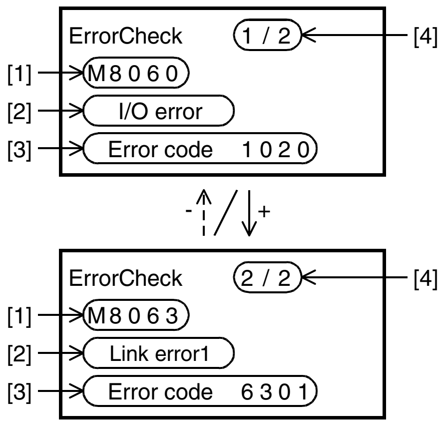
When error does not occur



When one error occurs



2 or more errors occur

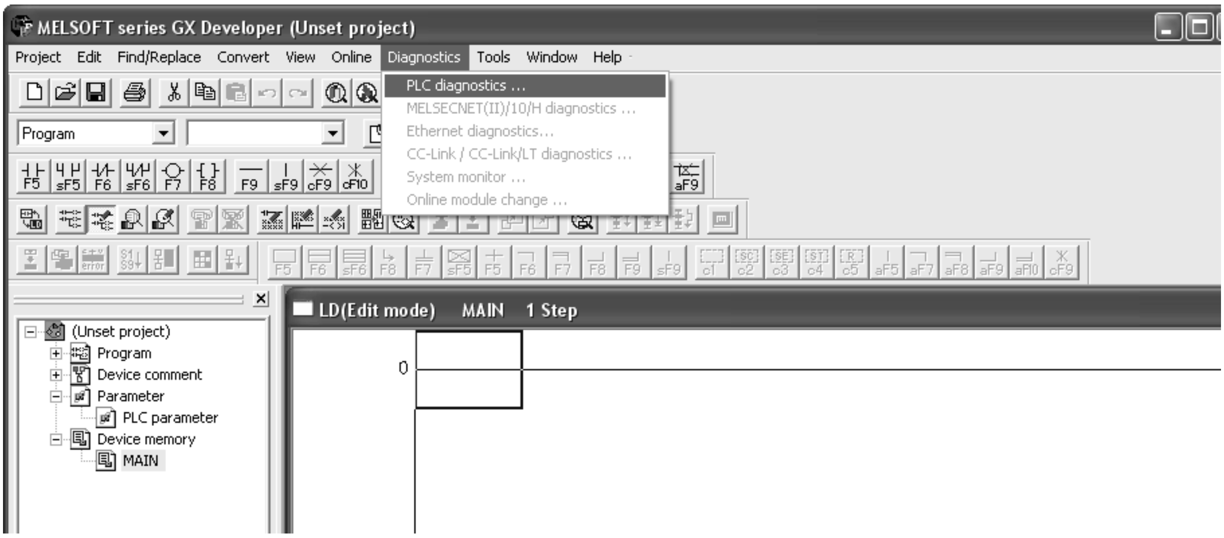


37.2.2 Error code check method by GX Developer

1 Connect a personal computer to PLC.

2 Execute PLC diagnostics.

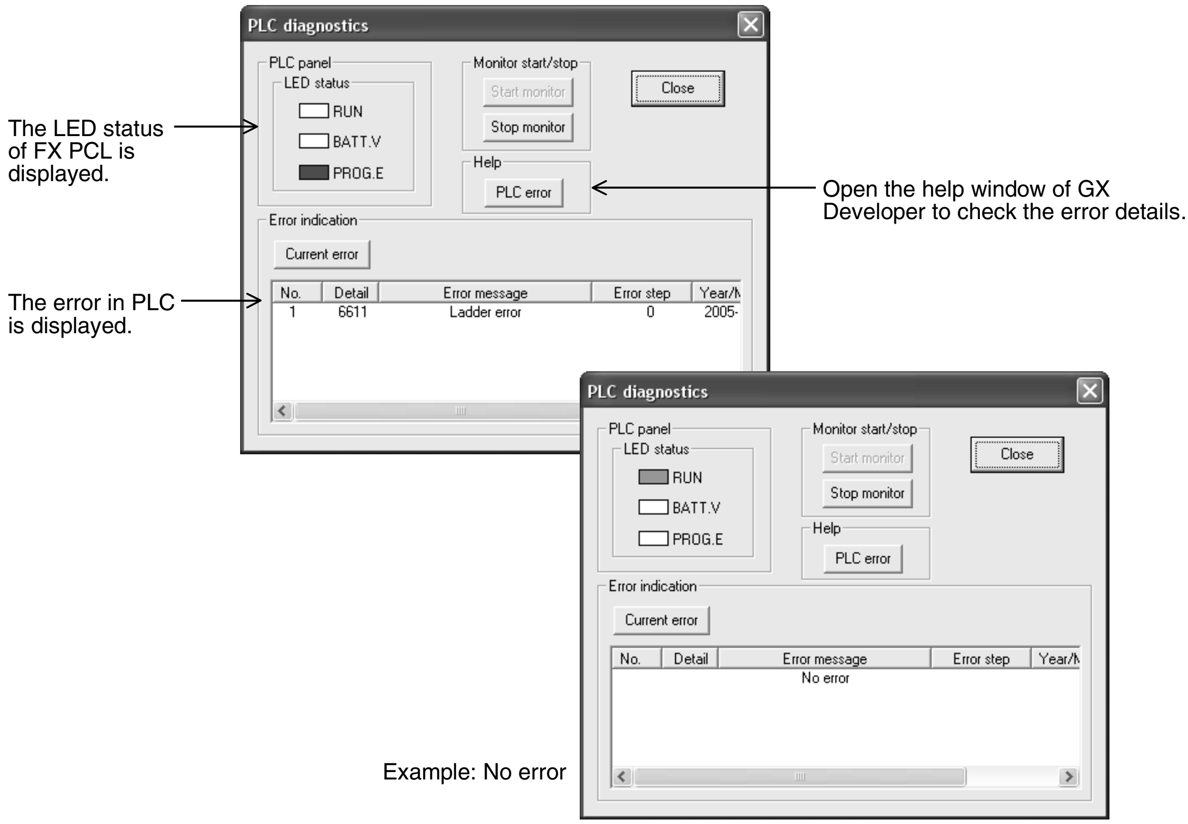
Click [Diagnostics] - [PLC diagnostics] on the tool menu, and execute the PLC diagnostics.



3 Check the diagnostics result.

The error check result displays in the following windows.

Example: one error occurs



Example: No error

37.2.3 Error indication

The table below shows the error expression in this manual, GX Developer, and display modules.

This manual	GX Developer	Display modules
	English version	Display in English
I/O configuration error	I/O config err	I/O error
PLC hardware error	PLC H/W error	PLC H/W error
PLC/PP communication error	PLC/PP comm err	Comms.error
Serial communication error 1 [ch1]	Link error	Link error1
Serial communication error 2 [ch2]	Link error2	Link error2
Parameter error	Param error	Parameter error
Syntax error	Syntax error	Grammar error
Circuit error	Ladder error	Ladder error
Operation error	Operation err	Runtime error
BFM initialization failure	–	–
Special block error	–	SFB error

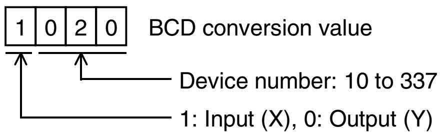
37.3 Supplementary Explanation of Devices for Error Detection

37.3.1 Error detection (M8060 to/D8060 to)

When the M8060, M8061, M8064 to M8067 turn ON, the smallest ON device number is stored in D8004, and M8004 turns ON.

1) M8060,M8061,M8064 to M8067 are cleared when the PLC mode switches from STOP to RUN.
   Note that M8068 and D8068 do not clear.
2) When turning M8069 ON in advance, PLC will enter STOP mode (as M8061 PLC hardware error occurs) if a failure occurs in an I/O extension unit, an extension power supply module, or an extension unit/block.
   When turning M8069 ON, PLC executes I/O bus check. If an error is found, error code 6103 or 6104 is stored to D8061, and M8061 turns ON.
   When error code 6104 is stored, M8009 turns ON, and the PLC stores the I/O numbers following the extension power supply module or the powered extension unit with DC 24V output failure to D8009.
3) If the unit or block corresponding to a programmed I/O number is not actually loaded, M8060 is set to ON and the first device number of the erroneous block is written to D8060.

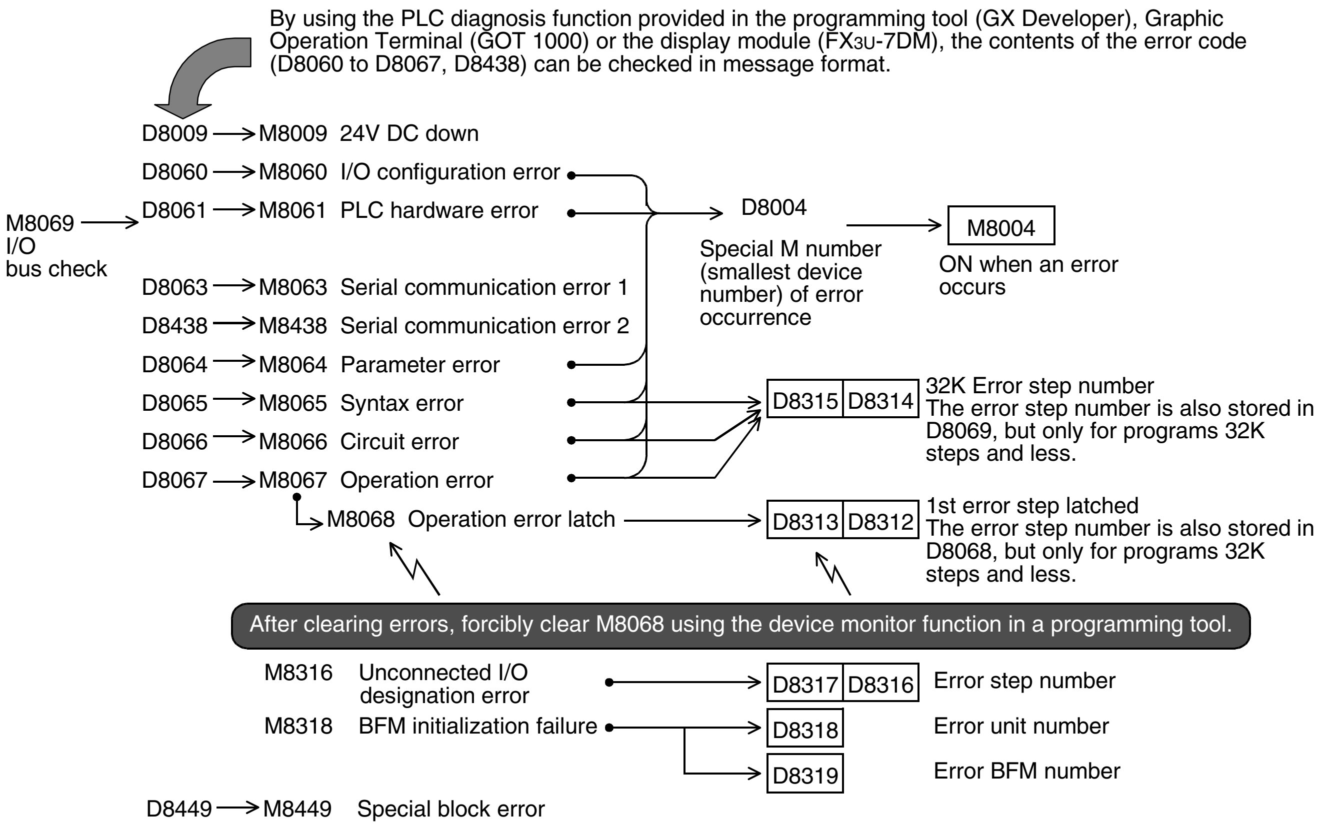
   Example: When X020 is unconnected



4) When a device number is specified directly or indirectly with an index by the LD, AND, OR or OUT instruction, and if the device numbers specified in those instructions are not actually loaded, M8316 will turn ON and the error step number in the instruction will be written to D8317 (high-order bits) and D8316 (low-order bits).

37.3.2 Operations of special devices for error detection

Special auxiliary relays for error detection and special data registers for error detection operate in the relationship shown below.
The state of error occurrence can be checked by monitoring the contents of auxiliary relays and data registers or by using the PLC diagnosis function programming tool.



37.3.3 Error detection timing

Error item	ERROR LED status	PLC status	Error detection timing		
			When power is turned from OFF to ON	When PLC mode switches from STOP to RUN	Other timing
M8060 I/O configuration error	Unlit	RUN	Check	Check	—
M8061 PLC hardware error	Lit	STOP	Check	—	Always
M8063 Serial communication error 1 [ch 1]	Unlit	RUN	—	—	When receiving signal from counterpart station
M8438 Serial communication error 2 [ch 2]	Unlit	RUN	—	—	When receiving signal from counterpart station
M8064 Parameter error	Flickering	STOP	Check	Check	When program is changed (STOP) When program is transferred (STOP)
M8065 Syntax error	Flickering	STOP			
M8066 Circuit error	Flickering	STOP			
M8067 Operation error	Unlit	RUN	—	—	RUN mode
M8068 Operation error latch	Unlit	RUN			
M8109 I/O refresh error	Unlit	RUN	—	—	Always
M8316 Unconnected I/O designation error	Unlit	RUN	—	—	RUN mode
M8318 BFM initialization failure	Unlit	RUN	—	Check	—
M8449 Special block error	Unlit	RUN	—	—	Always

37.4 Error Code List and Action

When a program error occurs in the PLC, the error code is stored in special data registers D8060 - D8067 and D8438. The following actions should be followed for diagnostic errors.

Error codes in shaded columns are added in FX3U and FX3UC PLCs.

Error code	PLC operation at error occurrence	Contents of error	Action
I/O configuration error [M8060(D8060)]			
Ex-ample: 1020	Continues operation	The head number of unconnected I/O device Example: When X020 is unconnected 1 0 2 0 BCD conversion value Device number: 10 to 337 1: Input (X), 0: Output (Y) • 1st to 3rd digits: Device number • 4th digit: I/O type (1 = input (X), 0 = output (Y)) Example: When 1020 is stored in D8060 Inputs X020 and later are unconnected.	Unconnected I/O relay numbers are programmed. The PLC continues its operation. Modify the program, check wiring connection, or add the appropriate unit/block.
Serial communication error 2 [M8438 (D8438)]			
0000	—	No error	
3801	Continues operation	Parity, overrun or framing error	• Inverter communication, computer link and programming: Ensure the communication parameters are correctly set according to their applications. • N:N network, parallel link, etc.: Check programs according to applications. • Remote maintenance: Ensure modem power is ON and check the settings of the AT commands. • Wiring: Check the communication cables for correct wiring.
3802		Communication character error	
3803		Communication data sum check error	
3804		Communication data format error	
3805		Command error	
3806		Communication time-out detected	
3807		Modem initialization error	
3808		N:N network parameter error	
3812		Parallel link character error	
3813		Parallel link sum error	
3814		Parallel link format error	
3820		Inverter communication error	
PLC hardware error [M8061(D8061)]			
0000	—	No error	
6101	Stops operation	RAM error	
6102		Operation circuit error	
6103		I/O bus error (M8069 = ON)	Confirm for the correct connection of extension cables.
6104		Powered extension unit 24 V failure (M8069 = ON)	
6105		Watchdog timer error	Confirm user program. The scan time exceeds the value stored in D8000.
6106		I/O table creation error (CPU error)	• When turning the power ON to the main unit, a 24V power failure occurs in a powered extension unit. (The error occurs if the 24V power is not supplied for 10 seconds or more after main power turns ON.) • When turning main power ON, I/O assignment to CC-Link/LT (built into the FX3UC PLC) is disabled.
6107		System configuration error	Check the number of connected special function units/blocks. Some special function units/blocks have a connection number limit.

Error codes in shaded columns are added in FX3U and FX3UC PLCs.

Error code	PLC operation at error occurrence	Contents of error	Action
PLC/PP communication error (D8062)			
0000	—	No error	
6201	Continues operation	Parity, overrun or framing error	Confirm the cable connection between the programming panel (PP)/programming device and the PLC. This error may occur when a cable is disconnected an reconnected during PLC monitoring.
6202		Communication character error	
6203		Communication data sum check error	
6204		Data format error	
6205		Command error	
Serial communication error 1 [M8063 (D8063)]			
0000	—	No error	
6301	Continues operation	Parity, overrun or framing error	• Inverter communication, computer link and programming: Ensure the communication parameters are correctly set according to their applications. • N:N network, parallel link, etc.: Check programs according to applications. • Remote maintenance: Ensure modem power is ON and check the settings of the AT commands. • Wiring: Check the communication cables for correct wiring.
6302		Communication character error	
6303		Communication data sum check error	
6304		Communication data format error	
6305		Command error	
6306		Communication time-out detected	
6307		Modem initialization error	
6308		N:N network parameter error	
6312		Parallel link character error	
6313		Parallel link sum error	
6314		Parallel link format error	
6320		Inverter communication error	
Parameter error [M8064(D8064)]			
0000	—	No error	
6401	Stops operation	Program sum check error	STOP the PLC, and correctly set the parameters.
6402		Memory capacity setting error	
6403		Latched device area setting error	
6404		Comment area setting error	
6405		File register area setting error	
6406		Special unit (BFM) initial value setting, positioning instruction setting sum check error	
6407		Special unit (BFM) initial value setting, positioning instruction setting error	
6409		Other setting error	
Syntax error [M8065(D8065)]			
0000	—	No error	
6501	Stops operation	Incorrect combination of instruction, device symbol and device number	During programming, each instruction is checked. If a syntax error is detected, modify the instruction correctly.
6502		No OUT T or OUT C before setting value	
6503		• No setting value after OUT T or OUT C • Insufficient number of operands for an applied instruction	
6504		• Same label number is used more than once. • Same interrupt input or high speed counter input is used more than once.	
6505		Device number is out of allowable range.	
6506		Invalid instruction	
6507		Invalid label number [P]	
6508		Invalid interrupt input [I]	
6509		Other error	
6510		MC nesting number error	

Error codes in shaded columns are added in FX3U and FX3UC PLCs.

Error code	PLC operation at error occurrence	Contents of error	Action
Circuit error [M8066(D8066)]			
0000	—	No error	
6610	Stops operation	LD, LDI is continuously used 9 times or more.	This error occurs when a combination of instructions is incorrect in the entire circuit block or when the relationship between a pair of instructions is incorrect. Modify the instructions in the program mode so that their mutual relationship becomes correct.
6611		More ANB/ORB instructions than LD/LDI instructions	
6612		Less ANB/ORB instructions than LD/LDI instructions	
6613		MPS is continuously used 12 times or more.	
6614		No MPS instruction	
6615		No MPP instruction	
6616		No coil between MPS, MRD and MPP, or incorrect combination	
6617		Instruction below is not connected to bus line: STL, RET, MCR, P, I, DI, EI, FOR, NEXT, SRET, IRET, FEND or END	
6618		STL, MC or MCR can be used only in main program, but it is used elsewhere (e.g. in interrupt routine or subroutine).	
6619		Invalid instruction is used in FOR-NEXT loop: STL, RET, MC, MCR, I (interrupt pointer) or IRET.	
6620		FOR-NEXT instruction nesting level exceeded	
6621		Numbers of FOR and NEXT instructions do not match.	
6622		No NEXT instruction	
6623		No MC instruction	
6624		No MCR instruction	
6625		STL instruction is continuously used 9 times or more.	
6626		Invalid instruction is programmed within STL-RET loop: MC, MCR, I (interrupt pointer), SRET or IRET.	
6627		No STL instruction	
6628		Invalid instruction is used in main program: I (interrupt pointer), SRET or IRET	
6629		No P or I (interrupt pointer)	
6630		No SRET or IRET instruction STL-RET / MC-MCR instructions programmed in the subroutine.	
6631		SRET programmed in invalid location	
6632		FEND programmed in invalid location	

Error codes in shaded columns are added in FX3U and FX3UC PLCs.

Error code	PLC operation at error occurrence	Contents of error	Action
Operation error [M8067(D8067)]			
0000		No error	
6701	Continues operation	• No jump destination (pointer) for CJ or CALL instruction • Label is undefined or out of P0 to P4095 due to indexing • Label P63 is executed in CALL instruction; cannot be used in CALL instruction as P63 is for jumping to END instruction.	This error occurs in the execution of operation. Review the program, or check the contents of the operands used in the applied instructions. Even if the syntax or circuit design is correct, an operation error may still occur. For example: "T200Z" itself is not an error. But if Z had a value of 400, the timer T600 would be attempted to be accessed. This would cause an operation error since there is no T600 device available.
6702		CALL instruction nesting level is 6 or more	
6703		Interrupt nesting level is 3 or more	
6704		FOR-NEXT instruction nesting level is 6 or more.	
6705		Operand of applied instruction is inapplicable device.	
6706		Device number range or data value for operand of applied instruction exceeds limit.	
6707		File register is accessed without parameter setting of file register.	
6708		FROM/TO instruction error	This error occurs in the execution of operation. • Review the program, or check the contents of the operands used in the applied instructions. • Verify that the specified buffer memories exist in the equipment. • Verify that the extension cables are correctly connected.
6709		Other (e.g. improper branching)	This error occurs in the execution of operation. Review the program, or check the contents of the operands used in the applied instructions. Even if the syntax or circuit design is correct, an operation error may still occur. For example: "T200Z" itself is not an error. But if Z had a value of 400, the timer T600 would be attempted to be accessed. This would cause an operation error since there is no T600 device available.
6710		Mismatch among parameters	This error occurs when the same device is used within the source and destination in a shift instruction, etc.
6730		Incorrect sampling time (T_S) ($T_S \leq 0$)	<PID instruction is stopped.> This error occurs in the parameter setting value or operation data executing PID instruction. Check the contents of the parameters.
6732		Incompatible input filter constant (α) ($\alpha < 0$ or $100 \leq \alpha$)	
6733		Incompatible proportional gain (K_P) ($K_P < 0$)	
6734		Incompatible integral time (T_I) ($T_I < 0$)	
6735		Incompatible derivative gain (K_D) ($K_D < 0$ or $201 \leq K_D$)	
6736		Incompatible derivative time (T_D) ($T_D < 0$)	
6740		Sampling time (T_S) $\leq$ Scan time	<Auto tuning is continued.> The operation is continued in the condition "sampling time (T_S) = cyclic time (scan time)".

Error codes in shaded columns are added in FX3U and FX3UC PLCs.

Error code	PLC operation at error occurrence	Contents of error	Action
Operation error [M8067(D8067)]			
6742	Continues operation	Variation of measured value exceeds limit. (△PV < –32768 or +32767 < △PV)	<PID operation is continued.> The operation is continued with each parameter set to the maximum and minimum value.
6743		Deviation exceeds limit. (EV < –32768 or +32767 < EV)	
6744		Integral result exceeds limit. (Outside range from –32768 to +32767)	
6745		Derivative value exceeds limit due to derivative gain (KD).	
6746		Derivative result exceeds limit. (Outside range from –32768 to +32767)	
6747		PID operation result exceeds limit. (Outside range from –32768 to +32767)	
6748		PID output upper limit set value < PID output lower limit set value.	<Transpose of output upper limit value and output lower limit value. → PID operation is continued.> Verify that the target setting contents are correct.
6749		Abnormal PID input variation alarm set value or output variation alarm set value (Set value < 0)	<Alarm output is not given. → PID operation is continued.> Verify that the target setting contents are correct.
6750		<Step response method> Improper auto tuning result	<Auto tuning is finished. → PID operation is started.> • When auto tuning was started, the difference between the measured value and the target value was 150 or less. (SV–PV≤150) • When auto tuning was started, the difference between the measured value and the target value was 1/3 or more. • Check the measured value and target value, and then execute auto tuning again.
6751		<Step response method> Auto tuning operation direction mismatch	<Auto tuning is forcibly finished. → PID operation is not started.> The operation direction estimated from the measured value at the start of auto tuning was different from the actual operation direction of the output during auto tuning. Correct the relationship among the target value, output value for auto tuning, and the measured value, and then execute auto tuning again.
6752		<Step response method> Improper auto tuning operation	<Auto tuning is finished. → PID operation is not started.> Because the set value was fluctuated during auto tuning, auto tuning was not executed correctly. Set the sampling time to a value larger than the output change cycle, or set a larger value for the input filter constant. After changing the setting, execute auto tuning again.
6753		<Limit cycle method> Abnormal output set value for auto tuning [ULV (upper limit) ≤ LLV (lower limit)]	<Auto tuning is forcibly finished. → PID operation is not started.> Check whether the target setting contents are correct.
6754		<Limit cycle method> Abnormal PV threshold (hysteresis) set value for auto tuning (SHPV < 0)	
6755		<Limit cycle method> Abnormal auto tuning transfer status (Data of device controlling transfer status is abnormally overwritten.)	<Auto tuning is forcibly finished. → PID operation is not started.> Ensure that devices occupied by PID instruction are not overwritten in the program.

Error codes in shaded columns are added in FX3U and FX3UC PLCs.

Error code	PLC operation at error occurrence	Contents of error	Action
Operation error [M8067(D8067)]			
6756	Continues operation	<Limit cycle method> Abnormal result due to excessive auto tuning measurement time (τon > τ, τon < 0, τ < 0)	<Auto tuning is forcibly finished. → PID operation is not started.> The auto tuning time is longer than necessary. Increase the difference (ULV - LLV) between the upper limit and lower limit of the output value for auto tuning, set a smaller value to the input filter constant (α), or set a smaller value to the PV threshold (SH_{PV}) for auto tuning, and then check the result for improvement.
6757		<Limit cycle method> Auto tuning result exceeds proportional gain. (KP= outside range from 0 to 32767)	<Auto tuning is finished (K_P = 32767). → PID operation is started.> The variation of the measured value (PV) is small compared with the output value. Multiply the measured value (PV) by "10" so that the variation of the measured value will increase during auto tuning.
6758		<Limit cycle method> Auto tuning result exceeds integral time. (T_I = outside range from 0 to 32767)	<Auto tuning is finished (K_P = 32767). → PID operation is started.> The auto tuning time is longer than necessary. Increase the difference (ULV - LLV) between the upper limit and lower limit of the output value for auto tuning, set a smaller value to the input filter constant (α), or set a smaller value to the PV threshold (SH_{PV}) for auto tuning, and then check the result for improvement.
6759		<Limit cycle method> Auto tuning result exceeds derivative time. (T_D = outside range from 0 to 32767)	
6760		ABS data read from servo sum check error	Check servo wiring and parameter setting. Also check the ABS instruction.
6762		Port specified by inverter communication instruction is already used in another communication.	Check to make sure the port is not specified by another instruction.
6763		1) Input (X) specified by DSZR, DVIT or ZRN instruction is already used in another instruction. 2) The interrupt signal device for DVIT instruction is outside the allowable setting range.	1) Check to make sure the input (X), as specified by DSZR, DVIT or ZRN instruction, is not being used for the following purposes: - Input interrupt (including the delay function) - High speed counter C235 to C255 - Pulse catch M8050 to M8057 - SPD instruction 2) Check the contents of D8336 for the correct interrupt signal specification for DVIT instruction.
6764		Pulse output number is already used in a positioning instruction or pulse output instruction (PLSY, PWM, etc.).	Check to make sure the pulse output destination is not being driven by another positioning instruction.
6765		Number of applied instruction exceeds limit.	Confirm that the number of times an applied instruction is used in the program does not exceed the specified limit.
6770		Writing error to flash memory cassette	
6771		Flash memory cassette is not connected.	Check for the correct attachment of the memory cassette.
6772		Flash memory cassette is protected against writing.	The write-protect switch of the flash memory cassette was set to ON when data was transferred to the flash memory. Set the protect switch to OFF.
6773		Access error to flash memory during writing in RUN mode	While data was written in the RUN mode, data was transferred to (read from or written to) the flash memory (Set the protect switch to OFF).

Error codes in shaded columns are added in FX3U and FX3UC PLCs.

Error code	PLC operation at error occurrence	Contents of error	Action
Special block error [M8449 (D8449)]			
☐020[*1]	Continues operation	General data sum error	Check for the correct connection of extension cables.
☐021[*1]		General data message error	
☐080[*1]		FROM/TO error	This error occurs in the execution of operation. • Review the program, or check the contents of the operands used in the applied instructions. • Check whether the specified buffer memories exist in the counterpart equipment. • Check for the correct connection of extension cables.
☐090[*1]		Peripheral equipment access error	• Check the cable connection between the programming panel (PP) / programming device and the PLC. • Check for the correct connection of extension cables.

*1. The unit number 0 to 7 of the special function unit/block error is put in ☐.

Appendix A: Programming Tool Applicability and Version Upgrade History

Appendix A-1 Programming Tool Applicability

Appendix A-1-1 Programming tool applicability

1. Applicable versions of programming tool

GX Developer is applicable to FX3U and FX3UC PLCs starting with the following version:

1) FX3U PLCs

FX3U PLC version	Model name (Media model name is shown below.)	Applicable GX Developer version	Remarks
Ver. 2.20 or later	GX Developer SW□D5C(F)-GPPW-J SW□D5C(F)-GPPW-E	Ver.8.23Z or later	• Supports FX3U PLCs (Ver.2.20 or later). Model selection: FX3U(C)*1
Ver. 2.30 or later		Ver.8.29F or later	• Supports FX3U PLCs (Ver.2.30 or later). Model selection: FX3U(C)

*1. For Ver. 8.23Z or 8.24A of GX Developer, the PLC type is FX3UC.

2) FX3UC PLCs

FX3UC PLC version	Model name (Media model name is shown below.)	Applicable GX Developer version	Remarks
Ver.1.00 or later	GX Developer SW□D5C(F)-GPPW-J SW□D5C(F)-GPPW-E	Ver.8.13P or later	• Supports FX3UC PLCs (Ver.1.00 or later). Model selection: FX3UC
Ver.1.30 or later		Ver.8.18U or later	• Supports FX3UC PLCs (Ver.1.30 or later). Model selection: FX3UC
Ver. 2.20 or later		Ver.8.23Z or later	• Supports FX3UC PLCs (Ver.2.20 or later). Model selection: FX3U(C)*2
Ver. 2.30 or later		Ver.8.29F or later	• Supports FX3UC PLCs (Ver.2.30 or later). Model selection: FX3U(C)

*2. For Ver. 8.23Z or 8.24A of GX Developer, the PLC type is FX3UC.

2. In the case of programming tool (version) not applicable to FX3U and FX3UC PLCs (using an alternative model)

Even using a programming tool not applicable to the FX3U and FX3UC PLCs, programming is enabled when an alternative model is set.

In this case, however, programming is enabled only in the function ranges such as instructions, device ranges and program sizes available in a PLC selected as the alternative model. In addition, use a programming tool that can select either FX3U(C) or FX3UC to change parameters, i.e. memory capacity, file register capacity, etc.

Model to be programmed	Model to be set	Priority High → Low					
FX3U PLC	FX3U(C)*3	→	FX3UC	→	FX2N	→	FX2
FX3UC PLC	FX3U(C)*3	→	FX3UC	→	FX2N	→	FX2

*3. The model selection is "FX3UC" in some versions of GX Developer Ver.8.23Z or later.

3. Program transfer speed and programming tool

When either of the following interfaces is used for GX Developer (Ver.8.13P or later), writing and reading of programs and monitoring of devices can be executed at high speed (115.2 kbps) in FX3U and FX3UC PLCs.

- Standard built-in port or expansion board FX3UC-422-BD for RS-422
  When the RS-232C/RS-422 converter FX-232AWC-H or USB/RS-422 converter FX-USB-AW is connected
- Expansion board FX3U-232-BD for RS-232C
- Special adapter FX3U-232ADP for RS-232C
- Expansion board FX3U-USB-BD for USB

In programming software (GX Developer versions earlier then 8.13P),the communication speed is 9,600 or 19,200 bps.
In GX Developer, the communication speed can be set in the following position:
Select "Online" → "Transfer setup..." → "PC side I/F", and double-click "Serial" icon.

Appendix A-1-2 Cautions on writing during RUN

In FX3U and FX3UC PLCs, writing is enabled during RUN (program changes in the RUN mode) using the following programming tools.

→ For the operating procedure of and cautions on writing during RUN, refer to the manual of the programming tool used.

Programming tools which support writing during RUN

Programming tool	Version	Remarks
GX Developer	Ver.2.00A or later	Writing in the instruction and device ranges during RUN is supported in FX2N PLCs Ver.1.00 or later.
	Ver.7.00A or later	Writing in the instruction and device ranges during RUN is supported in FX2N PLCs Ver.3.00 or later.
	Ver.8.13P or later	Writing in the instruction and device ranges during RUN is supported in FX3UC PLCs Ver.1.00 or later.
	Ver.8.18U or later	Writing in the instruction and device ranges during RUN is supported in FX3UC PLCs Ver.1.30 or later.
	Ver.8.23Z or later	Writing in the instruction and device ranges during RUN is supported in FX3U and FX3UC PLCs Ver.2.20 or later.
	Ver. 8.29F or later	Writing in the instruction and device ranges during RUN is supported in FX3U and FX3UC PLCs Ver. 2.30 or later
FX-PCS/WIN(-E)	Ver.1.00 or later	Writing in the instruction and device ranges during RUN is supported in FX2 PLCs Ver.3.30 or later.
	Ver.2.00 or later	Writing in the instruction and device ranges during RUN is supported in FX2N PLCs Ver.1.00 or later.
	Ver.4.20 or later	Writing in the instruction and device ranges during RUN is supported in FX2N PLCs Ver.3.00 or later.

Cautions on writing during RUN

Item		Caution
Program memories which can be written in RUN mode		Built-in RAM and optional memory cassette (whose write protect switch is set to OFF)
Number of program steps which can be written for circuit change in RUN mode	GX Developer Ver.8.23Z or later	• In case of FX3U/FX3UC PLCs Ver.2.00 or later 256 steps or less after edit (addition/deletion) (including NOP immediately after circuit blocks except final circuit) • In case of FX3UC PLCs earlier than Ver.2.00 127 steps or less after edit (addition/deletion) (including NOP immediately after circuit blocks except final circuit)
	• GX Developer Ver.8.22Y and earlier • FX-PCS/WIN(-E)	127 steps or less after edit (addition/deletion) (including NOP immediately after circuit blocks except final circuit)

<table>
<tr><th>Item</th><th>Caution</th></tr>
<tr><td rowspan="3">Circuit blocks which cannot be written during RUN</td><td>Circuit blocks in which labels P and I are added, deleted or changed in edited circuits</td></tr>
<tr><td>Circuit blocks in which 1-ms timers (T246 to T249 and T255 to T511) are added in edited circuits</td></tr>
<tr><td>Circuit blocks in which the following instructions are included in edited circuits
• Instruction to output high speed counters C235 to C255 (OUT instruction)
• SORT2 (FNC149), TBL (FNC152), RBFM (FNC278) and/or WBFM (FNC279) instruction</td></tr>
<tr><td rowspan="3">Circuit blocks which require attention on operation after writing during RUN</td><td>During RUN ,avoid writing to a circuit block including the following instructions under execution. If writing is executed during RUN to such a circuit block, the PLC decelerates and stops pulse output.
• DSZR (FNC150), DVIT (FNC151), ZRN (FNC156), PLSV (FNC157) instruction [with acceleration/deceleration operation], DRVI (FNC158) and/or DRVA (FNC159) instruction</td></tr>
<tr><td>During RUN ,avoid writing to a circuit block including the following instruction under execution. If writing is executed during RUN to such a circuit block, the PLC immediately stops pulse output.
• PLSV (FNC157) instruction [without acceleration/deceleration operation]</td></tr>
<tr><td>During RUN ,avoid writing to a circuit block including the following instructions under execution of communication. If write during RUN is executed to such a circuit block, the PLC may stop communication after that.
If the PLC stops communication, set the PLC to the STOP mode once, and then set it to the RUN mode again.
• IVCK (FNC270), IVDR (FNC271), IVRD (FNC272), IVWR (FNC273) and/or IVBWR (FNC274) instruction
• When writing is completed during RUN for a circuit including an instruction for falling edge pulse (LDF, ANDF, or ORF instruction), the instruction for falling edge pulse is not executed without regard to the ON/OFF status of the target device of the instruction for falling edge pulse.
When writing is completed during RUN for a circuit including an instruction for falling edge pulse (PLF instruction), the instruction for falling edge pulse is not executed without regard to the ON/OFF status of the operation condition device.
It is necessary to set to ON the target device or operation condition device once and then set it to OFF to execute the instruction for falling edge pulse.
• When writing is completed during RUN for a circuit including an instruction for rising edge pulse, the instruction for rising edge pulse is executed if a target device of the instruction for rising edge pulse or the operation condition device is ON.
Target instructions for rising edge pulse: LDP, ANDP, ORP, and pulse operation type applied instructions (such as MOVP)
<table>
<tr><th>While write during RUN is executed</th><th>Instruction for rising edge pulse</th><th>Instruction for falling edge pulse</th></tr>
<tr><td>OFF</td><td>Not executed</td><td>Not executed</td></tr>
<tr><td>ON</td><td>Executed*1</td><td>Not executed</td></tr>
</table>
*1. The PLS instruction is not executed.</td></tr>
</table>

<table>
<tr><th>Item</th><th>Caution Description</th></tr>
<tr><td>Circuit blooks which require attention on operation after writing during RUN</td><td>When writing to a circuit block during RUN, which includes the following instructions, the following results.
• Pulse instruction during rising edge of operation results (MEP instruction)
After writing to the circuit with MEP instructions during RUN, the MEP instruction result turns ON (conductive) while the operation results leading up to the MEP instruction are ON.
• Pulse instruction during falling edge of operation results (MEF instruction)
After writing to the circuit with MEF instructions during RUN, the MEF instruction result turns OFF (non-conductive), regardless of the operation results up to the MEF instruction. The operation results of the MEF instruction turns ON (conductive) when the operation results leading up to the MEF instruction are turned OFF.
<table>
<tr><th>Operation results leading up to MEP/MEF instruction</th><th>MEP instruction</th><th>MEF instruction</th></tr>
<tr><td>OFF</td><td>OFF
(non-conductive)</td><td>OFF
(non-conductive)</td></tr>
<tr><td>ON</td><td>ON
(conductive)</td><td>OFF
(non-conductive)</td></tr>
</table></td></tr>
<tr><td>Other</td><td>• When GX Developer Ver. 8.13 or later is used to write during RUN, the following results.
If the number of program steps is reduced by deleting contact or coil application instructions, etc., the program becomes shorter by the number of reduced steps.</td></tr>
</table>

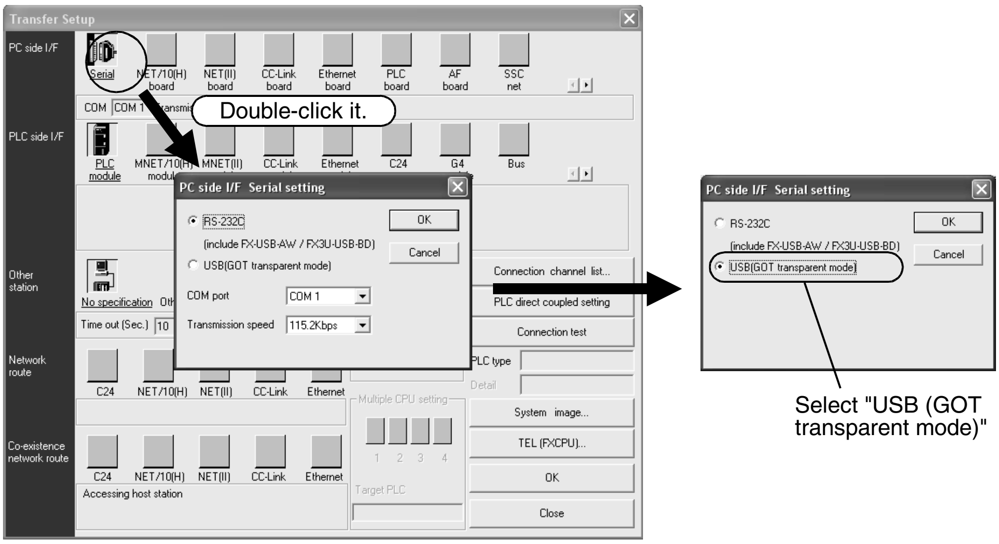
Appendix A-1-3 Cautions on using transparent function by way of USB in GOT1000 Series

When monitoring circuits, device registration, etc. or reading/writing programs in an FX3U or FX3UC PLC from GX Developer Ver.8.22Y or later using the transparent function by way of USB in the GOT1000 Series, make sure to execute the following settings.
If the following setting is not provided, a communication error occurs.

	GX Developer Ver.8.21X or former	GX Developer Ver8.22Y or later
When using transparent function by way of USB in GOT1000 Series	Not supported (not available)	Setting shown below is required.
When using transparent function by way of RS-232 in GOT1000 Series	Set "COM port" and "Transmission speed" on "PC side I/F Serial setting" dialog box.	Select "RS-232C" in setting shown below, and set "COM port" and "Transmission speed".
When directly connecting GX Developer to PLC		

Setting in GX Developer (Ver. 8.22Y or later)

1. **Select [Online] → [Transfer setup...] to open "Transfer setup" dialog box.**
2. **Double-click [Serial] in [PC side I/F] to open "PC side I/F Serial setting" dialog box.**
3. **Select "USB (GOT Transparent mode)".**



4. **Click [OK] button to finish the setting.**

Appendix A-1-4 Cautions on using transparent (2-port) function of GOT-F900 Series

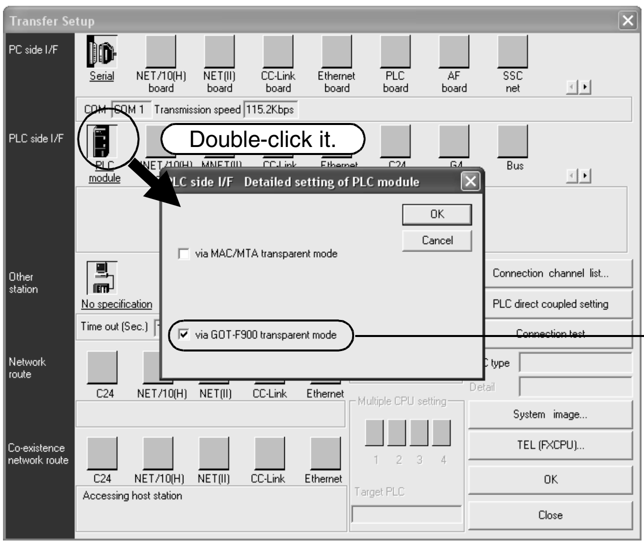
When monitoring circuits, device registration, etc. in an FX3U or FX3UC PLC from GX Developer Ver. 8.13P or later using the transparent (2-port) function in the GOT-F900 Series, make sure to execute the following settings.
If the following setting is not provided, monitoring cannot be normally executed .

	GX Developer Ver.8.12N or earlier*1	GX Developer Ver.8.13P or later	GX Developer Ver.8.22Y or later
When using transparent function in GOT-F900 Series	Setting shown below is not required.	Setting shown below is required.	Select "RS-232C" on "PC side I/F Serial setting" dialog box, and execute settings shown below.
When directly connecting GX Developer to PLC	Set "COM port" and "Transmission speed" on "PC side I/F Serial setting" dialog box.		Select "RS-232C" on "PC side I/F Serial setting" dialog box, and set "COM port" and "Transmission speed".

*1. GX Developer Ver.8.13P or later supports the FX3UC Series.

Setting in GX Developer (Ver.8.13P or later)

1. **Select [Online] → [Transfer setup...] to open "Transfer Setup" dialog box.**
2. **Double-click [PLC module] in [PLC side I/F] to open [PLC side I/F Detailed setting of PLC module] dialog box.**
3. **Put a check mark in the check box [via GOT-F900 transparent mode] as shown below.**



Put a check mark in the check box "via GOT-F900 transparent mode".

4. **Click [OK] button to finish the setting.**

Appendix A-2 Peripheral product applicability (except programming tools)

Appendix A-2-1 Applicable products and versions

Product Name	Applicable Versions
GOT1000 Series	From the first version

Cautions

This product is compatible with the FX_{3U} device range, however confirm with the GOT manual regarding other compatible items.

Appendix A-2-2 Incompatible peripheral products

Product Name	Compatible Versions
F940WGOT (built-in 2 port interface)	Ver. 1.00 or later (from the first version)
F940GOT (built-in 2 port interface)	Ver. 1.00 or later (from the first version)*1
F930GOT (-K) (built-in 2 port interface)	Ver. 1.00 or later (from the first version)
F920GOT (-K) (built-in 2 port interface)	Ver. 1.00 or later (from the first version)
ET-940 (built-in 2 port interface)	Ver. 1.00 or later (from the first version)*1
FX-100M (-SET0)	From the first version
FX-100U	Ver. 3.00 or later

Limitations

The compatibility range includes up to the FX_{2N}/FX_{2NC} PLC and is limited to the FX_{2N}/FX_{2NC} PLC function range of commands, device ranges, program sizes, etc.

*1. The transparent (2 ports) function of GX Developer is not compatible with Ver. 1.10 or earlier of F940GOT and ET-940.

Appendix A-3 Version Upgrade History

Appendix A-3-1 Version check

The D8001 (decimal) special data register contains information for determining the PLC version. Or in FX3U and FX3UC PLCs, the PLC version can be verified in "PLC Status" within the display module.

→ **For the operating procedure of the display module, refer to the Hardware Edition Manual of the PLC to be checked.**

D8001
PLC type and version

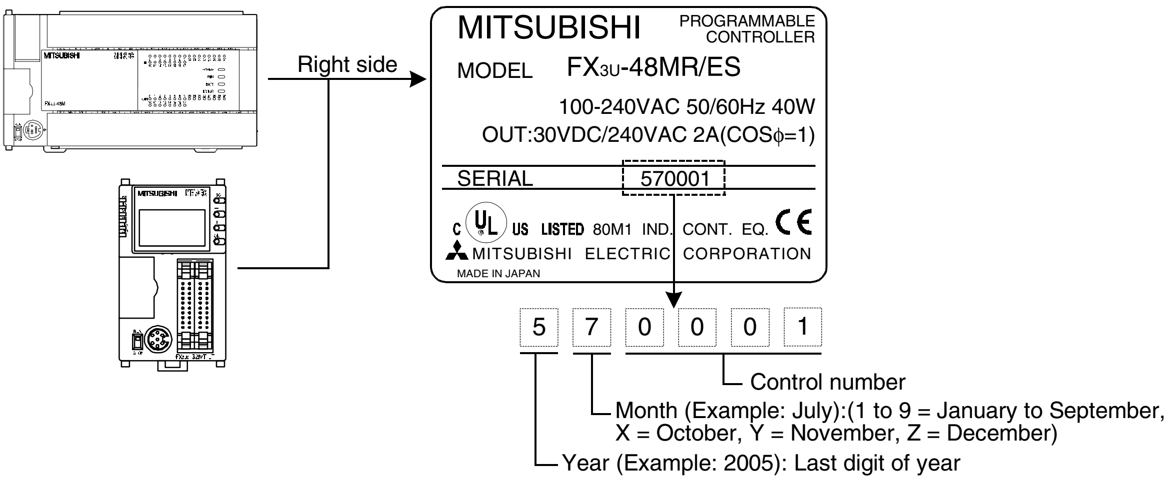
2 4 1 0 0

- 1 0 0: Version information (Example: Ver. 1.00)
- 2 4: PLC type (Example: 24 = FX2N/FX2NC/FX3U/FX3UC PLC)

Appendix A-3-2 How to look at manufacturer's serial number

The year and month of production of the product can be seen from the manufacturer's serial number "SERIAL" indicated on the label adhered to the right side of the product.

Example: FX3U-48MR/ES (manufacturer's serial number: 570001)



Appendix A-3-3 Version upgrade history [FX3U]

Version	Manufacturer's serial number	Contents of version upgrade
Ver.2.20	55**** (May, 2005)	First product (supporting the functions described as "Ver.2.20 or later" in this manual) Corresponds to FX3UC PLC Ver. 2.20.
Ver. 2.30	5Y**** (November, 2005)	Adding the following instructions and function up. Adding MEP and MEF instructions Function up of MUL (FNC 22), DIV (FNC 23), and RS2 (FNC 87) instructions

→ **For the method to look at the manufacturer's serial number, refer to Appendix A-3-2.**

Appendix A-3-4 Version upgrade history [FX3UC]

Version	Manufacturer's serial number	Contents of version upgrade
Ver.1.00	41**** (January, 2004)	First product
Ver.1.20	44**** (April, 2004)	Supports connection of following special analog adapters: - FX3U-4AD-ADP - FX3U-4DA-ADP
Ver.1.30	48**** (August, 2004)	• Supports connection of following special analog adapters: - FX3U-4AD-PT-ADP - FX3U-4AD-TC-ADP • Supports connection of following special function block: - FX3UC-4AD • Supports following 3 instructions: - SCL2 (FNC269), RWER (FNC 294), and INITER (FNC 295) • Adds function of following instruction: - DVIT (FNC151)
Ver.2.20	55**** (May, 2005)	Supports the functions described as "Ver.2.20 or later" in this manual. • Supports following 28 instructions: ZPUSH(FNC102), ZPOP(FNC103), WSUM(FNC140),WTOB(FNC141), BTOW(FNC142),UNI(FNC143), DIS(FNC144), SORT2(FNC149), TBL(FNC152), COMRD(FNC182), DUTY(FNC186), BK+(FNC192), BK-(FNC193), BKCMP=(FNC194), BKCMP>(FNC195), BKCMP<(FNC196), BKCMP<>(FNC197), BKCMP<=(FNC198), BKCMP>=(FNC199), STR(FNC200), VAL(FNC201), INSTR(FNC208), FDEL(FNC210), FINS(FNC211), DABIN(FNC260), BINDA(FNC261), RBFM(FNC278), WBFM(FNC279) • Adds function of following 5 instructions: SPD(FNC 56), DSZR(FNC150), DVIT(FNC151), ZRN(FNC156), PLSV(FNC157), HCMOV(FNC189) • Supports connection of FREQROL-F700/A700 inverters supporting following 5 instructions: IVCK (FNC270), IVDR (FNC271), IVRD (FNC272), IVWR (FNC273), and IVBWR (FNC274) • Adds second entry code (when GX Developer SW8.23Z (Ver. 8.23Z or later) is used). • Supports BFM initial value setting function (when GX Developer SW8.23Z (Ver. 8.23Z or later) is used). • Mitigates restriction in writing during RUN (when GX Developer SW8.23Z (Ver. 8.23Z or later) is used). - Number of steps which can be changed by one-time write during RUN is changed. 127 steps → 256 steps - Handling of circuit blocks which can be changed by one-time write during RUN is changed. Program of continuous circuit blocks having 127 steps or less → Program of circuit blocks having 256 steps or less in total
Ver. 2.30	5Y**** (November, 2005)	Adding the following instructions and function upgrade. Adding MEP and MEF instructions Function upgrade of MUL (FNC 22), DIV (FNC 23), and RS2 (FNC 87) instructions.

→ For the method to look at the manufacturer's serial number, refer to Appendix A-3-2

Appendix B: Instruction Execution Time

The instruction execution time in FX3U and FX3UC PLCs is as shown below:

Measurement condition

- High speed counters and interrupt instructions (I) are not used together.
- In operands, data registers are used as target devices.
- Indexing (V or Z) is not provided.

Appendix B-1 Basic Instruction Execution Time

Instruction	Execution time in ON status (μs)		Execution time in OFF status (μs)		Remarks
	16-bit instruction	32-bit instruction	16-bit instruction	32-bit instruction	
Contact instructions					
LD	0.065				0.129 μs in 2-step instruction*1 0.193 μs in 3-step instruction*3
LDI	0.065				0.129 μs in 2-step instruction*1 0.193 μs in 3-step instruction*3
LDP	7.8		—		
LDF	7.8		—		
AND	0.065				0.129 μs in 2-step instruction*1 0.193 μs in 3-step instruction*3
ANI	0.065				0.129 μs in 2-step instruction*1 0.193 μs in 3-step instruction*3
ANP	7.5		—		
ANF	7.5		—		
OR	0.065				0.129 μs in 2-step instruction*1 0.193 μs in 3-step instruction*3
ORI	0.065				0.129 μs in 2-step instruction*1 0.193 μs in 3-step instruction*3
ORP	7.4		—		
ORF	7.4		—		
Connection instructions					
ANB	0.065				
ORB	0.065				
MPS	0.065				
MRD	0.065				
MPP	0.065				
INV	0.065				
MEP	3.4				
MEF	3.4				
Output instructions					
OUT Y,M	0.065				0.129 μs in 2-step instruction*2 0.193 μs in 3-step instruction*3
OUT S	4.8		4.8		
OUT T K	0.71		0.71		T192 to T199, T246 to T511 11.6 μs in execution in ON status 8.2 μs in execution in OFF status

Instruction	Execution time in ON status (μs)		Execution time in OFF status (μs)		Remarks
	16-bit instruction	32-bit instruction	16-bit instruction	32-bit instruction	
Output instructions					
OUT T D	0.71		0.71		T192 to T199, T246 to T511 11.6 μs in execution in ON status 8.2 μs in execution in OFF status
OUT C K	0.71	6.1	0.71	6.1	C235 to C255 9.5 μs in execution in ON status 9.0 μs in execution in OFF status
OUT C D	0.71	6.1	0.71	6.1	C235 to C255 9.5 μs in execution in ON status 9.0 μs in execution in OFF status
SET Y,M	0.065				0.129 μs in 2-step instruction*2 0.193 μs in 3-step instruction*3
SET S	4.7 or 6.6 + 0.9n		0.13		n: Number of recombination 4.7 μs when there is no recombination
RST Y,M	0.065				0.129 μs in 2-step instruction*2 0.193 μs in 3-step instruction*3
RST S	4.6		0.13		
RST T	0.45		0.45		
RST C	0.45	5.8	0.45	4.8	
RST D	5.4	—	0.195	—	
PLS Y,M	0.257				0.321 μs in M3584 to M7679
PLF Y,M	0.257				0.321 μs in M3584 to M7679
Master control instructions					
MC	4.3		4.7		
MCR	3.9		—		
Other instruction					
NOP	0.065				
End instruction					
END	113.9 + 2.13X + 3.25Y		—		Even if FEND and END are used together, execution time of only END is required. X: Number of input points, Y: Number of output points

*1. M1536 to M3583, M8256 to M8511, S1024 to S4095

*2. M1536 to M3583, M8000 to M8511

*3. M3584 to M7679

Appendix B-2 Step Ladder Instruction Execution Time

Instruction	Execution time in ON status (μs)		Execution time in OFF status (μs)		Remarks
	16-bit instruction	32-bit instruction	16-bit instruction	32-bit instruction	
STL	5.1 + 1.6n		—		n: Number of recombination n = 0: When there is no recombination
RET	2.9		—		

Appendix B-3 Label (P/I) Execution Time

Instruction	Execution time in ON status (μs)		Execution time in OFF status (μs)		Remarks
	16-bit instruction	32-bit instruction	16-bit instruction	32-bit instruction	
P***	0.065				0.129 μs in P256 to P4095
I***	0.065				

Appendix B-4 Applied Instruction Execution Time

FNC No.	Instruction	Execution time in ON status (μs)		Execution time in OFF status (μs)		Remarks
		16-bit instruction	32-bit instruction	16-bit instruction	32-bit instruction	
Program flow						
00	CJ	8.0	—	0.195	—	
01	CALL	13.5 (CALL+SRET)	—	0.195	—	
02	SRET		—	—	—	
03	IRET	4.4	—	—	—	
04	EI	3.8	—	—	—	
05	DI	3.7	—	—	—	
06	FEND	113.9 + 2.13X + 3.25Y		—	—	Even if FEND and END are used together, execution time of only END is required. X: Number of input points, Y: Number of output points
07	WDT	5.4	—	0.065	—	
08	FOR	11.6 (FOR + NEXT)	—	—	—	
09	NEXT		—	—	—	
Move and compare						
10	CMP	15.5	16.0	0.455	0.845	
11	ZCP	18.9	19.7	0.585	1.105	
12	MOV	0.64	1.48	0.32	1.48	For details, refer to Appendix B-6-2.
13	SMOV	22.9	—	0.715	—	
14	CML	10.6	10.2	0.325	0.585	
15	BMOV	13.9 + 0.44n	—	0.455	—	
16	FMOV	14.2 + 0.19n	14.0 + 0.38n	0.455	0.845	
17	XCH	10.7	11.4	0.325	0.585	
18	BCD	11.0	15.5	0.325	0.585	
19	BIN	7.5	8.4	0.325	0.585	
Arithmetic and logical operation						
20	ADD	8.1	8.5	0.455	0.845	
21	SUB	8.1	8.5	0.455	0.845	
22	MUL	7.0	7.3	0.455	0.845	
23	DIV	8.2	8.6	0.455	0.845	
24	INC	6.2	6.4	0.195	0.325	
25	DEC	6.2	6.4	0.195	0.325	
26	WAND	7.0	7.3	0.455	0.845	
27	WOR	7.0	7.3	0.455	0.845	
28	WXOR	7.0	7.3	0.455	0.845	
29	NEG	7.6	8.0	0.195	0.325	

FNC No.	Instruction	Execution time in ON status (μs)		Execution time in OFF status (μs)		Remarks
		16-bit instruction	32-bit instruction	16-bit instruction	32-bit instruction	
Rotation and shift						
30	ROR	10.5	11.5	0.325	0.585	
31	ROL	10.5	11.5	0.325	0.585	
32	RCR	10.9	11.8	0.325	0.585	
33	RCL	10.9	11.8	0.325	0.585	
34	SFTR	23.2 + 0.08n	—	0.585	—	
35	SFTL	23.2 + 0.08n	—	0.585	—	
36	WSFR	7.5 + 0.44n	—	0.585	—	
37	WSFL	7.5 + 0.44n	—	0.585	—	
38	SFWR	8.1	—	0.455	—	
39	SFRD	7.7	—	0.455	—	
Data operation						
40	ZRST(D)	11.1 + 0.19n	—	0.325	—	n: Number of reset points
	ZRST(T)	17.1 + 0.23n	—	0.325	—	n: Number of reset points
	ZRST(M)	20.7 + 0.02n	—	0.325	—	n: Number of reset points
41	DECO	13.5	—	0.455	—	
42	ENCO	18.0	—	0.455	—	
43	SUM	12.7	16.9	0.325	0.585	
44	BON	14.4	15.1	0.455	0.845	
45	MEAN	11.8 + 0.41n	17.8 + 2.13n	0.455	0.845	
46	ANS	20.4	—	19.7	—	
47	ANR	7.0	—	0.065	—	
48	SQR	9.7	12.1	0.325	0.585	
49	FLT	9.8	9.5	0.325	0.585	
High speed processing						
50	REF	4.5 + 1.39n	—	0.325	—	
51	REFF	14.4 + 0.24n	—	0.195	—	
52	MTR	5.9	—	5.5	—	
53	HSCS	—	20.0	—	0.845	
54	HSCR	—	20.0	—	0.845	
55	HSZ	—	22.0	—	1.105	
56	SPD	16.0	16.0	12.6	12.6	
57	PLSY	20.0	13.6	6.9	6.9	
58	PWM	10.6	—	6.2	—	
59	PLSR	11.2	11.2	7.0	7.0	
Handy instructions						
60	IST	28.5	—	0.455	—	
61	SER	16.4 + 1.4n	18.5 + 2.13n	0.585	1.105	
62	ABSD	19 + 0.85n	20.0 + 1.23n	0.585	1.105	
63	INCD	23.7	—	6.5	—	
64	TTMR	10.4	—	9.2	—	
65	STMR	19.0	—	21.0	—	
66	ALT	11.6	—	0.2	—	
67	RAMP	15.0	—	7.5	—	
68	ROTC	25.8	—	24.8	—	
69	SORT	18.4	—	6.6	—	

FNC No.	Instruction	Execution time in ON status (μs)		Execution time in OFF status (μs)		Remarks
		16-bit instruction	32-bit instruction	16-bit instruction	32-bit instruction	
External FX I/O device						
70	TKY	21.5	21.8	5.2	5.2	
71	HKY	32.0	32.3	5.7	5.7	
72	DSW	26.8	—	22.1	—	
73	SEGD	10.8	—	0.325	—	
74	SEGL	22.3	—	7.5	—	
75	ARWS	28.8	—	5.2	—	
76	ASC	19.8	—	0.715	—	
77	PR	24.0	—	13.6	—	
78	FROM*1	141 + 419n	119 + 841n	0.585	1.105	n: Number of transfer points
	FROM*2	107 + 903n	119 + 1791n	0.585	1.105	n: Number of transfer points
	FROM*3	27.9 + 108n	17.6 + 187.4n	0.585	1.105	n: Number of transfer points
79	TO*1	87 + 483n	73 + 967n	0.585	1.105	n: Number of transfer points
	TO*2	73 + 967n	67 + 1923n	0.585	1.105	n: Number of transfer points
	TO*3	96.7 + 119.2n	17.3 + 297.7n	0.585	1.105	n: Number of transfer points

*1. When the instruction is executed to BFM #0 to BFM #31 in a special function block/unit for the FX0N/FX2N/FX2NC Series or in the CC-Link/LT built in the FX3UC-32MT-LT

*2. When the instruction is executed to BFM #32 or later in a special function block/unit for the FX0N/FX2N/FX2NC Series or in the CC-Link/LT built in the FX3UC-32MT-LT

*3. When the instruction is executed to a BFM in a special function block/unit for the FX3U/FX3UC Series

External FX device						
80	RS	15.6	—	5.7	—	
81	PRUN	17.1 + 1.67n	18.2 + 2.9n	0.325	0.585	
82	ASCI	13.5 + 1.45n	—	0.455	—	
83	HEX	13.6 + 1.89n	—	0.455	—	
84	CCD	13.6 + 1.63n	—	0.455	—	
85	VRRD	—	—	—	—	
86	VRSC	—	—	—	—	
87	RS2	18.1	—	5.3	—	
88	PID	20.0	—	8.9	—	
89	—	—	—	—	—	
Data move 2						
100	—	—	—	—	—	
101	—	—	—	—	—	
102	ZPUSH	16.0	—	0.195	—	
103	ZPOP	16.0	—	0.195	—	
104	—	—	—	—	—	
105	—	—	—	—	—	
106	—	—	—	—	—	
107	—	—	—	—	—	
108	—	—	—	—	—	
109	—	—	—	—	—	

FNC No.	Instruction	Execution time in ON status (μs)		Execution time in OFF status (μs)		Remarks
		16-bit instruction	32-bit instruction	16-bit instruction	32-bit instruction	
Floating point						
110	ECMP	—	18.2	—	0.845	
111	EZCP	—	21.6	—	1.105	
112	EMOV	—	10.0	—	0.585	
113	—	—	—	—	—	
114	—	—	—	—	—	
115	—	—	—	—	—	
116	ESTR	—	27 + 1.7n + 1.26m	—	0.845	n: Number of character digits m: Number of digits in decimal part
117	EVAL	—	26 + 3.8n	—	0.585	
118	EBCD	—	10.0	—	0.585	
119	EBIN	—	11.9	—	0.585	
120	EADD	—	14.2	—	0.845	
121	ESUB	—	14.2	—	0.845	
122	EMUL	—	14.1	—	0.845	
123	EDIV	—	17.7	—	0.845	
124	EXP	—	11.9	—	0.585	
125	LOGE	—	24.0	—	0.585	
126	LOG10	—	24.3	—	0.585	
127	ESQR	—	10.6	—	0.585	
128	ENEG	—	8.9	—	0.325	
129	INT	13.2	13.0	0.325	0.585	
130	SIN	—	12.0	—	0.585	
131	COS	—	23.2	—	0.585	
132	TAN	—	12.0	—	0.585	
133	ASIN	—	13.5	—	0.585	
134	ACOS	—	13.5	—	0.585	
135	ATAN	—	12.0	—	0.585	
136	RAD	—	14.9	—	0.585	
137	DEG	—	14.9	—	0.585	
138	—	—	—	—	—	
139	—	—	—	—	—	
Data operation 2						
140	WSUM	11.7 + 0.38n	14.1 + 1.94n	0.455	0.845	
141	WTOB	12.6 + 1.43n	—	0.455	—	
142	BTOW	12.6 + 0.92n	—	0.455	—	
143	UNI	11.6 + 0.4n	—	0.455	—	
144	DIS	10.6 + 0.2n	—	0.455	—	
145	—	—	—	—	—	
146	—	—	—	—	—	
147	SWAP	7.7	8.0	0.195	0.325	
148	—	—	—	—	—	
149	SORT2	13.2	15.2	6.5	7.7	

FNC No.	Instruction	Execution time in ON status (µs)		Execution time in OFF status (µs)		Remarks
		16-bit instruction	32-bit instruction	16-bit instruction	32-bit instruction	
Positioning control						
150	DSZR	170.0	—	7.0	—	
151	DVIT	178.0	178.0	7.1	7.1	
152	TBL	—	*1	—	7.1	
153	—	—	—	—	—	
154	—	—	—	—	—	
155	ABS	—	25.4	—	22.2	
156	ZRN	58.0	62.0	7.1	7.1	
157	PLSV	144.0	144.0	7.1	7.1	
158	DRVI	178.0	178.0	7.1	7.1	
159	DRVA	178.0	178.0	7.1	7.1	

*1. The instruction execution time is as shown below depending on the executed positioning type (instruction)

Positioning type	Operation instruction	Instruction execution time in ON status (µs)
DDVIT (interrupt positioning)	DDVIT instruction	178.0 µs
DPLSV (variable speed pulse output)	DPLSV instruction	144.0 µs
DDRVI (drive to increment)	DDRVI instruction	178.0 µs
DRRVA (drive to absolute)	DRRVA instruction	178.0 µs

FNC No.	Instruction	Execution time in ON status (µs) 16-bit instruction	Execution time in ON status (µs) 32-bit instruction	Execution time in OFF status (µs) 16-bit instruction	Execution time in OFF status (µs) 32-bit instruction	Remarks
Real time clock control						
160	TCMP	21.3	—	0.715	—	
161	TZCP	22.6	—	0.585	—	
162	TADD	13.4	—	0.455	—	
163	TSUB	13.4	—	0.455	—	
164	HTOS	10.8	11.0	0.325	0.585	
165	STOH	11.4	11.6	0.325	0.585	
166	TRD	10.0	—	0.195	—	
167	TWR	344.4	—	0.195	—	
168	—	—	—	—	—	
169	HOUR	15.5	16.1	15.2	15.9	
External device						
170	GRY	10.2	10.7	0.325	0.585	
171	GBIN	15.4	16.0	0.325	0.585	
176	RD3A(3A)	1404	—	0.455	—	FX0N-3A
	RD3A(2AD)	1828	—	0.455	—	FX2N-2AD
177	WR3A(3A)	1466	—	0.455	—	FX0N-3A
	WR3A(2DA)	2919	—	0.455	—	FX2N-2DA
Extension function						
180	—	—	—	—	—	
Other instructions						
181	—	—	—	—	—	
182	COMRD	33.7	—	0.325	—	
183	—	—	—	—	—	
184	RND	8.5	—	0.195	—	
185	—	—	—	—	—	
186	DUTY	6.0	—	6.0	—	
187	—	—	—	—	—	
188	CRC	12.6 + 0.82n	—	0.455	—	
189	HCMOV	—	14.8	—	0.845	

FNC No.	Instruction	Execution time in ON status (μs)		Execution time in OFF status (μs)		Remarks
		16-bit instruction	32-bit instruction	16-bit instruction	32-bit instruction	
Block data operation						
190	—	—	—	—	—	
191	—	—	—	—	—	
192	BK+	13.1 + 0.66n	13.9 + 1.23n	0.585	1.105	
193	BK-	13.1 + 0.66n	13.9 + 1.23n	0.585	1.105	
194	BKCMP=	19.6 + 1.88n	20.3 + 2.26n	0.585	1.105	
195	BKCMP>	19.6 + 1.88n	20.3 + 2.26n	0.585	1.105	
196	BKCMP<	19.6 + 1.88n	20.3 + 2.26n	0.585	1.105	
197	BKCMP<>	19.6 + 1.88n	20.3 + 2.26n	0.585	1.105	
198	BKCMP<=	19.6 + 1.88n	20.3 + 2.26n	0.585	1.105	
199	BKCMP>=	19.6 + 1.88n	20.3 + 2.26n	0.585	1.105	
Character string control						
200	STR	34.6	47.0	0.455	0.845	
201	VAL	20.7	29.2	0.455	0.845	
202	$+	24.8 + 1.5m	—	0.455	—	m: Number of character strings
203	LEN	12 + 0.44m	—	0.325	—	m: Number of character strings
204	RIGHT	18.1 + 1.06n + 0.47m	—	0.455	—	n: Number of character strings m: Number of character data
205	LEFT	18.1 + 0.74n + 0.44m	—	0.455	—	n: Number of character strings m: Number of character data
206	MIDR	25 + 0.59n + 0.68m	—	0.455	—	n: Character position m: Number of characters
207	MIDW	25.8 + 0.3m + 0.44n	—	0.455	—	n: Character position m: Number of stored characters
208	INSTR	20.6 + 2.98m	—	0.585	—	m: Number of searched characters[*1]
209	$MOV	16 + 1.52n	—	0.325	—	

*1. Number of characters from the head of the searched character string to detected character

FNC No.	Instruction	16-bit instruction (ON)	32-bit instruction (ON)	16-bit instruction (OFF)	32-bit instruction (OFF)	Remarks
Data operation 3						
210	FDEL	43 + 0.95m	—	—	—	m: Number of data shifted forward[*2]
211	FINS	63 + 0.98m	—	—	—	m: Number of data shifted backward[*3]
212	POP	7.8	—	0.455	—	
213	SFR	9.3	—	0.325	—	
214	SFL	9.3	—	0.325	—	
215	—	—	—	—	—	
216	—	—	—	—	—	
217	—	—	—	—	—	
218	—	—	—	—	—	
219	—	—	—	—	—	

*2. m = (Number of data tables) - (Table position of deleted data)
Number of data tables: Present value of D·, table position of deleted data: n

*3. m = (Number of data tables) - (Table position of data insertion)
Number of data tables: Present value of D·, table position data insertion: n

FNC No.	Instruction	Execution time in ON status (μs)		Execution time in OFF status (μs)		Remarks
		16-bit instruction	32-bit instruction	16-bit instruction	32-bit instruction	
Data comparison						
220	—	—	—	—	—	
221	—	—	—	—	—	
222	—	—	—	—	—	
223	—	—	—	—	—	
224	LD=	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
225	LD>	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
226	LD<	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
227	—	—	—	—	—	
228	LD<>	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
229	LD<=	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
230	LD>=	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
231	—	—	—	—	—	
232	AND=	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
233	ADN>	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
234	AND<	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
235	—	—	—	—	—	
236	AND<>	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
237	AND<=	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
238	AND>=	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
239	—	—	—	—	—	
240	OR=	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
241	OR>	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
242	OR<	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
243	—	—	—	—	—	
244	OR<>	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
245	OR<=	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
246	OR>=	1.22	1.48	1.22	1.48	For details, refer to Appendix B-6-2.
247	—	—	—	—	—	
248	—	—	—	—	—	
249	—	—	—	—	—	

FNC No.	Instruction	Execution time in ON status (μs)		Execution time in OFF status (μs)		Remarks
		16-bit instruction	32-bit instruction	16-bit instruction	32-bit instruction	
Data table processing						
250	—	—	—	—	—	
251	—	—	—	—	—	
252	—	—	—	—	—	
253	—	—	—	—	—	
254	—	—	—	—	—	
255	—	—	—	—	—	
256	LIMIT	8.1	8.6	0.585	1.105	
257	BAND	8.1	8.6	0.585	1.105	
258	ZONE	7.9	8.5	0.585	1.105	
259	SCL	15.9	16.8	0.455	0.845	
260	DABIN	13.7	19.5	0.325	0.585	
261	BINDA	16.7	23.1	0.325	0.585	
262	—	—	—	—	—	
263	—	—	—	—	—	
264	—	—	—	—	—	
265	—	—	—	—	—	
266	—	—	—	—	—	
267	—	—	—	—	—	
268	—	—	—	—	—	
269	SCL2	2.79 + 5.21n	29.06 + 7.94n	0.455	0.845	n: Number of data
External device communication (inverter communication)						
270	IVCK	14.1	—	6.5	—	
271	IVDR	14.1	—	6.5	—	
272	IVRD	16.2	—	6.5	—	
273	IVWR	16.2	—	6.5	—	
274	IVBWR	20.0	—	6.5	—	
Data move 3						
275	—	—	—	—	—	
276	—	—	—	—	—	
277	—	—	—	—	—	
278	RBFM*1	50+900n	—	0.715	—	n: Number of points transferred in one operation cycle
	RBFM*2	244 + 103n	—	0.715	—	
279	WBFM*1	50+900n	—	0.715	—	n: Number of points transferred in one operation cycle
	WBFM*2	292 + 116n	—	0.715	—	

*1. When the instruction is executed to a BFM in a special function block/unit for the FX0N/FX2N/FX2NC Series or in the CC-Link/LT built in the FX3UC-32MT-LT

*2. When the instruction is executed to a BFM in a special function block/unit for the FX3U/FX3UC Series

FNC No.	Instruction	16-bit ON	32-bit ON	16-bit OFF	32-bit OFF	Remarks
High speed processing 2						
280	HSCT	—	30.0	—	1.365	
281	—	—	—	—	—	
282	—	—	—	—	—	
283	—	—	—	—	—	
284	—	—	—	—	—	
285	—	—	—	—	—	
286	—	—	—	—	—	
287	—	—	—	—	—	
288	—	—	—	—	—	
289	—	—	—	—	—	

FNC No.	Instruction	Execution time in ON status (μs)		Execution time in OFF status (μs)		Remarks
		16-bit instruction	32-bit instruction	16-bit instruction	32-bit instruction	
Extension file register control						
290	LOADR	13.2 + 0.44n	—	0.325	—	
291	SAVER	166n	—	6.4	—	
292	INITR	17600n	—	0.325	—	
293	LOGR	244 + 17.9n	—	0.715	—	
294	RWER	46700n	—	0.325	—	n: Number of write target sectors
295	INITER	17300n	—	0.325	—	
296 to 299	—	—	—	—	—	

Appendix B-5 Execution Time of Pulse Generation Instruction P in Each Applied Instruction

Applied instruction	Execution time	Remarks
MOV instruction (FNC 12)	Execution time at rising edge of input: Execution time in ON status	1.22 μs in MOVP instruction
	Non-execution time: Execution time in OFF status	
Other applied instructions	Execution time at rising edge of input: Execution time in ON status + 0.45 μs	
	Non-execution time: Execution time in OFF status + 0.45 μs	

Appendix B-6 Execution Time on Combination of Applicable Devices and Indexing

In examples shown below for basic instructions, MOV instruction and data comparison instructions, the instruction execution time varies depending on the combination of target devices and absence/presence of indexing.
The combinations of target devices marked with "*1" and "*2" are handled as exceptions because high speed processing is adopted in the technique.

Appendix B-6-1 Basic instruction (LD/LDI/AND/ANI/OR/ORI) execution time

Specified device type	Condition	Instruction execution time
Bit device	Without indexing	0.065
	With indexing	11.9
Word device	Bit specification	8.8

Appendix B-6-2 Applied instruction execution time

1. MOV (FNC 12) instruction execution time

MOV instruction (16-bit operation)

Command contact	S (source)	D (destination)							
		Without indexing (µs)				With indexing (µs)			
		KnY,KnM,KnS	T,C,D	R	U□\G□	KnY,KnM,KnS	T,C,D	R	U□\G□
ON	KnX,KnY,KnM,KnS	15.5	12.1	13.8	250.7	18.6	16.6	16.9	253.4
	T,C,D	12.1	0.64*1	10.4	247.3	16.6	14.5	14.8	251.3
	R	13.8	10.4	12.1	248.9	16.9	14.8	15.1	251.6
	U□\G□	227.3	223.9	225.6	462.0	230.0	227.9	228.2	464.3
	K,H	12.5	0.64*1	10.8	247.7	15.4	13.3	13.6	250.2
OFF	KnX,KnY,KnM,KnS								
	T,C,D		0.32*2		0.325 (0.325 in any combination other than *2)				
	R								
	U□\G□								
	K,H		0.32*2						

*1. *2. These combinations are handled as exceptions because high speed processing is adopted in the technique.

DMOV instruction (32-bit instruction)

Command contact	S (source)	D (destination)							
		Without indexing (µs)				With indexing (µs)			
		KnY,KnM,KnS	T,C,D	R	U□\G□	KnY,KnM,KnS	T,C,D	R	U□\G□
ON	KnX,KnY,KnM,KnS	15.5	12.4	13.8	483.9	18.7	17.0	17.1	486.8
	T,C,D	12.4	1.48*1	10.7	480.7	17.0	15.2	15.3	485.0
	R	13.8	10.7	12.2	482.2	17.1	15.3	15.4	485.2
	U□\G□	439.1	435.9	437.4	907.0	442.1	422.4	440.4	909.6
	K,H	13.0	1.48*1	11.4	481.5	16.2	14.4	14.5	484.2
OFF	KnX,KnY,KnM,KnS								
	T,C,D		1.48*2		0.585 (0.585 in any combination except *2)				
	R								
	U□\G□								
	K,H		1.48*2						

*1. *2. These combinations are handled as exceptions because high speed processing is adopted in the technique.

2. Data comparison instruction execution time

Data comparison instruction (16-bit operation)

S (source)	D (destination)							
	Without indexing (μs)				With indexing (μs)			
	KnY,KnM, KnS	T,C,D	R	U□\G□	KnY,KnM, KnS	T,C,D	R	U□\G□
KnX,KnY,KnM,KnS	16.2	13.0	14.7	228.2	19.4	17.4	17.6	230.9
T,C,D	13.0	1.22*1	11.3	224.8	17.4	15.4	15.7	229.0
R	14.7	11.3	12.9	226.5	17.6	15.7	16.0	229.3
U□\G□	228.2	224.8	226.5	439.5	230.9	229.0	229.3	442.0
K,H	13.4	1.22*1	11.7	225.2	16.3	14.2	14.5	227.9

*1. These combinations are handled as exceptions because high speed processing is adopted in the technique.

Data comparison instruction (32-bit operation)

S (source)	D (destination)							
	Without indexing (μs)				With indexing (μs)			
	KnY,KnM, KnS	T,C,D	R	U□\G□	KnY,KnM, KnS	T,C,D	R	U□\G□
KnX,KnY,KnM,KnS	16.4	13.2	14.7	440.2	19.6	17.9	18.0	442.9
T,C,D	13.2	1.48*1	11.6	436.8	17.9	16.1	16.2	441.3
R	14.7	11.6	13.0	438.3	18.0	16.2	16.3	441.4
U□\G□	440.2	436.8	438.3	863.1	442.9	441.3	441.4	865.8
K,H	13.9	1.48*1	12.3	437.5	17.1	15.3	15.4	440.5

*1. These combinations are handled as exceptions because high speed processing is adopted in the technique.

Appendix C: Applied Instruction List [by Instruction Type/in Alphabetic Order]

Appendix C-1 Applied instructions [by instruction type]

Applied instructions are classified into the following eighteen types:

1	Data transfer instructions	10	Character string operation instructions
2	Data conversion instructions	11	Program flow control instructions
3	Comparison instructions	12	I/O refresh instructions
4	Arithmetic operation instructions	13	Real time clock control instructions
5	Logical operation instructions	14	Pulse output/positioning control instructions
6	Special function instructions	15	Serial communication instructions
7	Rotation instructions	16	Special block/unit control instructions
8	Shift instructions	17	Extension register/extension file register control instructions
9	Data operation instructions	18	Other handy instructions

1. Data move instructions

Mnemonic	FNC No.	Function	Ref. Page
MOV	FNC 12	Move	226
SMOV	FNC 13	Shift Move	229
CML	FNC 14	Complement	231
BMOV	FNC 15	Block Move	233
FMOV	FNC 16	Fill Move	237
PRUN	FNC 81	Parallel Run (Octal Mode)	440
XCH	FNC 17	Exchange	240
SWAP	FNC147	Byte Swap	524
EMOV	FNC112	Floating Point Move	468
HCMOV	FNC189	High Speed Counter Move	584

2. Data conversion instructions

Mnemonic	FNC No.	Function	Ref. Page
BCD	FNC 18	Conversion to Binary Coded Decimal	242
BIN	FNC 19	Conversion to Binary	245
GRY	FNC170	Decimal to Gray Code Conversion	565
GBIN	FNC171	Gray Code to Decimal Conversion	566
FLT	FNC 49	Conversion to Floating Point	317
INT	FNC129	Floating Point to Integer Conversion	496
EBCD	FNC118	Floating Point to Scientific Notation Conversion	480
EBIN	FNC119	Scientific Notation to Floating Point Conversion	482
RAD	FNC136	Floating Point Degree to Radian Conversion	508

2. Data conversion instructions

Mnemonic	FNC No.	Function	Ref. Page
DEG	FNC137	Floating Point Radian to Degree Conversion	510

3. Comparison instructions

Mnemonic	FNC No.	Function	Ref. Page
LD=	FNC224	Load Compare S1 = S2	649
LD>	FNC225	Load Compare S1 > S2	649
LD<	FNC226	Load Compare S1 < S2	649
LD<>	FNC228	Load Compare S1 ≠ S2	649
LD<=	FNC229	Load Compare S1 ≤ S2	649
LD>=	FNC230	Load Compare S1 ≥ S2	649
AND=	FNC232	AND Compare S1 = S2	652
AND>	FNC233	AND Compare S1 > S2	652
AND<	FNC234	AND Compare S1 < S2	652
AND<>	FNC236	AND Compare S1 ≠ S2	652
AND<=	FNC237	AND Compare S1 ≤ S2	652
AND>=	FNC238	AND Compare S1 ≥ S2	652

3. Comparison instructions

Mnemonic	FNC No.	Function	Ref. Page
OR=	FNC240	OR Compare S1 = S2	655
OR>	FNC241	OR Compare S1 > S2	655
OR<	FNC242	OR Compare S1 < S2	655
OR<>	FNC244	OR Compare S1 ≠ S2	655
OR<=	FNC245	OR Compare S1 ≤ S2	655
OR>=	FNC246	OR Compare S1 ≥ S2	655
CMP	FNC 10	Compare	221
ZCP	FNC 11	Zone Compare	223
ECMP	FNC110	Floating Point Compare	464
EZCP	FNC111	Floating Point Zone Compare	466
HSCS	FNC 53	High Speed Counter Set	331
HSCR	FNC 54	High Speed Counter Reset	336
HSZ	FNC 55	High Speed Counter Zone Compare	339
HSCT	FNC280	High Speed Counter Compare With Data Table	707
BKCMP=	FNC194	Block Data Compare S1 = S2	596
BKCMP>	FNC195	Block Data Compare S1 > S2	596
BKCMP<	FNC196	Block Data Compare S1 < S2	596
BKCMP<>	FNC197	Block Data Compare S1 ≠ S2	596
BKCMP<=	FNC198	Block Data Compare S1 ≤ S2	596
BKCMP>=	FNC199	Block Data Compare S1 ≥ S2	596

4. Arithmetic operation instructions

Mnemonic	FNC No.	Function	Ref. Page
ADD	FNC 20	Addition	249
SUB	FNC 21	Subtraction	252
MUL	FNC 22	Multiplication	255
DIV	FNC 23	Division	258
EADD	FNC120	Floating Point Addition	484
ESUB	FNC121	Floating Point Subtraction	485
EMUL	FNC122	Floating Point M Multiplication	486
EDIV	FNC123	Floating Point Division	487
BK+	FNC192	Block Data Addition	590
BK-	FNC193	Block Data Subtraction	593
INC	FNC 24	Increment	261

4. Arithmetic operation instructions

Mnemonic	FNC No.	Function	Ref. Page
DEC	FNC 25	Decrement	263

5. Logical operation instructions

Mnemonic	FNC No.	Function	Ref. Page
WAND	FNC 26	Logical Word AND	264
WOR	FNC 27	Logical Word OR	266
WXOR	FNC 28	Logical Exclusive OR	268

6. Special function instructions

Mnemonic	FNC No.	Function	Ref. Page
SQR	FNC 48	Square Root	315
ESQR	FNC127	Floating Point Square Root	494
EXP	FNC124	Floating Point Exponent	488
LOGE	FNC125	Floating Point Natural Logarithm	490
LOG10	FNC126	Floating Point Common Logarithm	492
SIN	FNC130	Floating Point Sine	498
COS	FNC131	Floating Point Cosine	500
TAN	FNC132	Floating Point Tangent	501
ASIN	FNC133	Floating Point Arc Sine	502
ACOS	FNC134	Floating Point Arc Cosine	504
ATAN	FNC135	Floating Point Arc Tangent	506
RND	FNC184	Random Number Generation	575

7. Rotation instructions

Mnemonic	FNC No.	Function	Ref. Page
ROR	FNC 30	Rotation Right	273
ROL	FNC 31	Rotation Left	275
RCR	FNC 32	Rotation Right with Carry	277
RCL	FNC 33	Rotation Left with Carry	279

8. Shift instructions

Mnemonic	FNC No.	Function	Ref. Page
SFTR	FNC 34	Bit Shift Right	281
SFTL	FNC 35	Bit Shift Left	283
SFR	FNC213	Bit Shift Right with Carry	643
SFL	FNC214	Bit Shift Left with Carry	645
WSFR	FNC 36	Word Shift Right	287
WSFL	FNC 37	Word Shift Left	289
SFWR	FNC 38	Shift Write [FIFO/FILO Control]	291
SFRD	FNC 39	Shift Read [FIFO Control]	294
POP	FNC212	Shift Last Data Read [FILO Control]	640

9. Data operation instructions

Mnemonic	FNC No.	Function	Ref. Page
ZRST	FNC 40	Zone Reset	297
DECO	FNC 41	Decode	300
ENCO	FNC 42	Encode	303
MEAN	FNC 45	Mean	310
WSUM	FNC140	Sum of Word Data	513
SUM	FNC 43	Sum of Active Bits	305
BON	FNC 44	Check Specified Bit Status	308
NEG	FNC 29	Negation	270
ENEG	FNC128	Floating Point Negation	495
WTOB	FNC141	WORD to BYTE	515
BTOW	FNC142	BYTE to WORD	517
UNI	FNC143	4-bit Linking of Word Data	520
DIS	FNC144	4-bit Grouping of Word Data	522
CCD	FNC 84	Check Code	448
CRC	FNC188	Cyclic Redundancy Check	580
LIMIT	FNC256	Limit Control	659
BAND	FNC257	Dead Band Control	663
ZONE	FNC258	Zone Control	666
SCL	FNC259	Scaling (Coordinate by Point Data)	669
SCL2	FNC269	Scaling 2 (Coordinate by X/Y Data)	681
SORT	FNC 69	Sort Tabulated Data	398
SORT2	FNC149	Sort Tabulated Data 2	525
SER	FNC 61	Search a Data Stack	379
FDEL	FNC210	Deleting Data from Tables	636
FINS	FNC211	Inserting Data to Tables	638

10.Character string operation instructions

Mnemonic	FNC No.	Function	Ref. Page
ESTR	FNC116	Floating Point to Character String Conversion	469
EVAL	FNC117	Character String to Floating Point Conversion	475
STR	FNC200	BIN to Character String Conversion	602
VAL	FNC201	Character String to BIN Conversion	607
DABIN	FNC260	Decimal ASCII to BIN Conversion	674
BINDA	FNC261	BIN to Decimal ASCII Conversion	677
ASCI	FNC 82	Hexadecimal to ASCII Conversion	442
HEX	FNC 83	ASCII to Hexadecimal Conversion	445
$MOV	FNC209	Character String Transfer	632
$+	FNC202	Link Character Strings	612
LEN	FNC203	Character String Length Detection	615

10.Character string operation instructions

Mnemonic	FNC No.	Function	Ref. Page
RIGHT	FNC204	Extracting Character String Data From the Right	617
LEFT	FNC205	Extracting Character String Data from the Left	620
MIDR	FNC206	Random Selection of Character Strings	623
MIDW	FNC207	Random Replacement of Character Strings	626
INSTR	FNC208	Character string search	630
COMRD	FNC182	Read Device Comment Data	572

11.Program flow control instructions

Mnemonic	FNC No.	Function	Ref. Page
CJ	FNC 00	Conditional Jump	195
CALL	FNC 01	Call Subroutine	202
SRET	FNC 02	Subroutine Return	206
IRET	FNC 03	Interrupt Return	207
EI	FNC 04	Enable Interrupt	209
DI	FNC 05	Disable Interrupt	210
FEND	FNC 06	Main Routine Program End	211
FOR	FNC 08	Start a FOR/NEXT Loop	216
NEXT	FNC 09	End a FOR/NEXT Loop	217

12.I/O refresh instructions

Mnemonic	FNC No.	Function	Ref. Page
REF	FNC 50	Refresh	320
REFF	FNC 51	Refresh and Filter Adjust	324

13.Real time clock control instructions

Mnemonic	FNC No.	Function	Ref. Page
TCMP	FNC160	RTC Data Compare	545
TZCP	FNC161	RTC Data Zone Compare	547
TADD	FNC162	RTC Data Addition	550
TSUB	FNC163	RTC Data Subtraction	552
TRD	FNC166	Read RTC data	558
TWR	FNC167	Set RTC data	559
HTOS	FNC164	Hour to Second Conversion	554
STOH	FNC165	Second to Hour Conversion	556

14.Pulse output/positioning control instructions

Mnemonic	FNC No.	Function	Ref. Page
ABS	FNC155	Absolute Current Value Read	535
DSZR	FNC150	DOG Search Zero Return	530
ZRN	FNC156	Zero Return	536

14.Pulse output/positioning control instructions

Mnemonic	FNC No.	Function	Ref. Page
TBL	FNC152	Batch Data Positioning Mode	534
DVIT	FNC151	Interrupt Positioning	532
DRVI	FNC158	Drive to Increment	540
DRVA	FNC159	Drive to Absolute	542
PLSV	FNC157	Variable Speed Pulse Output	538
PLSY	FNC 57	Pulse Y Output	354
PLSR	FNC 59	Acceleration/Deceleration Setup	362

15.Serial communication instructions

Mnemonic	FNC No.	Function	Ref. Page
RS	FNC 80	Serial Communication	438
RS2	FNC 87	Serial Communication 2	451
IVCK	FNC270	Inverter Status Check	687
IVDR	FNC271	Inverter Drive	689
IVRD	FNC272	Inverter Parameter Read	691
IVWR	FNC273	Inverter Parameter Write	693
IVBWR	FNC274	Inverter Parameter Block Write	695

16.Special block/unit control instructions

Mnemonic	FNC No.	Function	Ref. Page
FROM	FNC 78	Read From a Special Function Block	430
TO	FNC 79	Write To a Special Function Block	435
RD3A	FNC176	Read form Dedicated Analog Block	567
WR3A	FNC177	Write to Dedicated Analog Block	568
RBFM	FNC278	Divided BFM Read	699
WBFM	FNC279	Divided BFM Write	704

17.Extension register/extension file register control instructions

Mnemonic	FNC No.	Function	Ref. Page
LOADR	FNC290	Load From ER	713
SAVER	FNC291	Save to ER	715
RWER	FNC294	Rewrite to ER	732
INITR	FNC292	Initialize R and ER	723
INITER	FNC295	Initialize ER	736
LOGR	FNC293	Logging R and ER	727

18.Other handy instructions

Mnemonic	FNC No.	Function	Ref. Page
WDT	FNC 07	Watchdog Timer Refresh	213
ALT	FNC 66	Alternate State	391

18.Other handy instructions

Mnemonic	FNC No.	Function	Ref. Page
ANS	FNC 46	Timed Annunciator Set	312
ANR	FNC 47	Annunciator Reset	314
HOUR	FNC169	Hour Meter	562
RAMP	FNC 67	Ramp Variable Value	393
SPD	FNC 56	Speed Detection	350
PWM	FNC 58	Pulse Width Modulation	359
DUTY	FNC186	Timing Pulse Generation	577
PID	FNC 88	PID Control Loop	453
ZPUSH	FNC102	Batch Store of Index Register	458
ZPOP	FNC103	Batch POP of Index Register	461
TTMR	FNC 64	Teaching Timer	387
STMR	FNC 65	Special Timer	389
ABSD	FNC 62	Absolute Drum Sequencer	382
INCD	FNC 63	Incremental Drum Sequencer	385
ROTC	FNC 68	Rotary Table Control	395
IST	FNC 60	Initial State	368
MTR	FNC 52	Input Matrix	327
TKY	FNC 70	Ten Key Input	402
HKY	FNC 71	Hexadecimal Input	406
DSW	FNC 72	Digital Switch (Thumbwheel Input)	410
SEGD	FNC 73	Seven Segment Decoder	413
SEGL	FNC 74	Seven Segment With Latch	415
ARWS	FNC 75	Arrow Switch	420
ASC	FNC 76	ASCII Code Data Input	425
PR	FNC 77	Print (ASCII Code)	427

Appendix C-2 Applied instructions [in alphabetical order]

Mnemonic	FNC No.	Function	Ref. Page
Symbol			
$+	FNC202	Link Character Strings	612
$MOV	FNC209	Character String Transfer	632
A			
ABS	FNC155	Absolute Current Value Read	535
ABSD	FNC 62	Absolute Drum Sequencer	382
ACOS	FNC134	Floating Point Arc Cosine	504
ADD	FNC 20	Addition	249
ALT	FNC 66	Alternate State	391
AND<	FNC234	AND Compare $S_1 < S_2$	652
AND<>	FNC236	AND Compare $S_1 \neq S_2$	652
AND=	FNC232	AND Compare $S_1 = S_2$	652
AND>	FNC233	AND Compare $S_1 > S_2$	652
AND<=	FNC237	AND Compare $S_1 \leq S_2$	652
AND>=	FNC238	AND Compare $S_1 \geq S_2$	652
ANR	FNC 47	Annunciator Reset	314
ANS	FNC 46	Timed Annunciator Set	312
ARWS	FNC 75	Arrow Switch	420
ASC	FNC 76	ASCII Code Data Input	425
ASCI	FNC 82	Hexadecimal to ASCII Conversion	442
ASIN	FNC133	Floating Point Arc Sine	502
ATAN	FNC135	Floating Point Arc Tangent	506
B			
BAND	FNC257	Dead Band Control	663
BCD	FNC 18	Conversion to Binary Coded Decimal	242
BIN	FNC 19	Conversion to Binary	245
BINDA	FNC261	BIN to Decimal ASCII Conversion	677
BK-	FNC193	Block Data Subtraction	593
BK+	FNC192	Block Data Addition	590
BKCMP<	FNC196	Block Data Compare $S_1 < S_2$	596
BKCMP<=	FNC198	Block Data Compare $S_1 \leq S_2$	596
BKCMP<>	FNC197	Block Data Compare $S_1 \neq S_2$	596
BKCMP=	FNC194	Block Data Compare $S_1 = S_2$	596
BKCMP>	FNC195	Block Data Compare $S_1 > S_2$	596
BKCMP>=	FNC199	Block Data Compare $S_1 \geq S_2$	596
BMOV	FNC 15	Block Move	233
BON	FNC 44	Check Specified Bit Status	308
BTOW	FNC142	BYTE to WORD	517
C			
CALL	FNC 01	Call Subroutine	202
CCD	FNC 84	Check Code	448
CJ	FNC 00	Conditional Jump	195
CML	FNC 14	Complement	231
CMP	FNC 10	Compare	221
COMRD	FNC182	Read Device Comment Data	572
COS	FNC131	Floating Point Cosine	500
CRC	FNC188	Cyclic Redundancy Check	580
D			
DABIN	FNC260	Decimal ASCII to BIN Conversion	674
DEC	FNC 25	Decrement	263
DECO	FNC 41	Decode	300
DEG	FNC137	Floating Point Radian to Degree Conversion	510
DI	FNC 05	Disable Interrupt	210
DIS	FNC144	4-bit Grouping of Word Data	522
DIV	FNC 23	Division	258
DRVA	FNC159	Drive to Absolute	542
DRVI	FNC158	Drive to Increment	540
DSW	FNC 72	Digital Switch (Thumbwheel Input)	410
DSZR	FNC150	DOG Search Zero Return	530
DUTY	FNC186	Timing Pulse Generation	577
DVIT	FNC151	Interrupt Positioning	532

Mnemonic	FNC No.	Function	Ref. Page
E			
EADD	FNC120	Floating Point Addition	484
EBCD	FNC118	Floating Point to Scientific Notation Conversion	480
EBIN	FNC119	Scientific Notation to Floating Point Conversion	482
ECMP	FNC110	Floating Point Compare	464
EDIV	FNC123	Floating Point Division	487
EI	FNC 04	Enable Interrupt	209
EMOV	FNC112	Floating Point Move	468
EMUL	FNC122	Floating Point Multiplication	486
ENCO	FNC 42	Encode	303
ENEG	FNC128	Floating Point Negation	495
ESQR	FNC127	Floating Point Square Root	494
ESTR	FNC116	Floating Point to Character String Conversion	469
ESUB	FNC121	Floating Point Subtraction	485
EVAL	FNC117	Character String to Floating Point Conversion	475
EXP	FNC124	Floating Point Exponent	488
EZCP	FNC111	Floating Point Zone Compare	466
F			
FDEL	FNC210	Deleting Data from Tables	636
FEND	FNC 06	Main Routine Program End	211
FINS	FNC211	Inserting Data to Tables	638
FLT	FNC 49	Conversion to Floating Point	317
FMOV	FNC 16	Fill Move	237
FOR	FNC 08	Start a FOR/NEXT Loop	216
FROM	FNC 78	Read From a Special Function Block	430
G			
GBIN	FNC171	Gray Code to Decimal Conversion	566
GRY	FNC170	Decimal to Gray Code Conversion	565
H			
HCMOV	FNC189	High Speed Counter Move	584
HEX	FNC 83	ASCII to Hexadecimal Conversion	445
HKY	FNC 71	Hexadecimal Input	406
HOUR	FNC169	Hour Meter	562
HSCR	FNC 54	High Speed Counter Reset	336
HSCS	FNC 53	High Speed Counter Set	331
HSCT	FNC280	High Speed Counter Compare With Data Table	707
HSZ	FNC 55	High Speed Counter Zone Compare	339
HTOS	FNC164	Hour to Second Conversion	554

Mnemonic	FNC No.	Function	Ref. Page
I			
INC	FNC 24	Increment	261
INCD	FNC 63	Incremental Drum Sequencer	385
INITER	FNC295	Initialize ER	736
INITR	FNC292	Initialize R and ER	723
INSTR	FNC208	Character string search	630
INT	FNC129	Floating Point to Integer Conversion	496
IRET	FNC 03	Interrupt Return	207
IST	FNC 60	Initial State	368
IVBWR	FNC274	Inverter Parameter Block Write	695
IVCK	FNC270	Inverter Status Check	687
IVDR	FNC271	Inverter Drive	689
IVRD	FNC272	Inverter Parameter Read	691
IVWR	FNC273	Inverter Parameter Write	693
L			
LD<	FNC226	Load Compare $S_1 < S_2$	649
LD<>	FNC228	Load Compare $S_1 \neq S_2$	649
LD=	FNC224	Load Compare $S_1 = S_2$	649
LD>	FNC225	Load Compare $S_1 > S_2$	649
LD<=	FNC229	Load Compare $S_1 \leq S_2$	649
LD>=	FNC230	Load Compare $S_1 \geq S_2$	649
LEFT	FNC205	Extracting Character String Data from the Left	620
LEN	FNC203	Character String Length Detection	615
LIMIT	FNC256	Limit Control	659
LOADR	FNC290	Load From ER	713
LOG10	FNC126	Floating Point Common Logarithm	492
LOGE	FNC125	Floating Point Natural Logarithm	490
LOGR	FNC293	Logging R and ER	727
M			
MEAN	FNC 45	Mean	310
MIDR	FNC206	Random Selection of Character Strings	623
MIDW	FNC207	Random Replacement of Character Strings	626
MOV	FNC 12	Move	226
MTR	FNC 52	Input Matrix	327
MUL	FNC 22	Multiplication	255
N			
NEG	FNC 29	Negation	270
NEXT	FNC 09	End a FOR/NEXT Loop	217

Mnemonic	FNC No.	Function	Ref. Page
O			
OR<	FNC242	OR Compare $S_1 < S_2$	655
OR<>	FNC244	OR Compare $S_1 \neq S_2$	655
OR=	FNC240	OR Compare $S_1 = S_2$	655
OR>	FNC241	OR Compare $S_1 > S_2$	655
OR<=	FNC245	OR Compare $S_1 \leq S_2$	655
OR>=	FNC246	OR Compare $S_1 \geq S_2$	655
P			
PID	FNC 88	PID Control Loop	453
PLSR	FNC 59	Acceleration/Deceleration Setup	362
PLSV	FNC157	Variable Speed Pulse Output	538
PLSY	FNC 57	Pulse Y Output	354
POP	FNC212	Shift Last Data Read [FILO Control]	640
PR	FNC 77	Print (ASCII Code)	427
PRUN	FNC 81	Parallel Run (Octal Mode)	440
PWM	FNC 58	Pulse Width Modulation	359
R			
RAD	FNC136	Floating Point Degree to Radian Conversion	508
RAMP	FNC 67	Ramp Variable Value	393
RBFM	FNC278	Divided BFM Read	699
RCL	FNC 33	Rotation Left with Carry	279
RCR	FNC 32	Rotation Right with Carry	277
RD3A	FNC176	Read form Dedicated Analog Block	567
REF	FNC 50	Refresh	320
REFF	FNC 51	Refresh and Filter Adjust	324
RIGHT	FNC204	Extracting Character String Data From the Right	617
RND	FNC184	Random Number Generation	575
ROL	FNC 31	Rotation Left	275
ROR	FNC 30	Rotation Right	273
ROTC	FNC 68	Rotary Table Control	395
RS	FNC 80	Serial Communication	438
RS2	FNC 87	Serial Communication 2	451
RWER	FNC294	Rewrite to ER	732

Mnemonic	FNC No.	Function	Ref. Page
S			
SAVER	FNC291	Save to ER	715
SCL	FNC259	Scaling (Coordinate by Point Data)	669
SCL2	FNC269	Scaling 2 (Coordinate by X/Y Data)	681
SEGD	FNC 73	Seven Segment Decoder	413
SEGL	FNC 74	Seven Segment With Latch	415
SER	FNC 61	Search a Data Stack	379
SFL	FNC214	Bit Shift Left with Carry	645
SFR	FNC213	Bit Shift Right with Carry	643
SFRD	FNC 39	Shift Read [FIFO Control]	294
SFTL	FNC 35	Bit Shift Left	283
SFTR	FNC 34	Bit Shift Right	281
SFWR	FNC 38	Shift Write [FIFO/FILO Control]	291
SIN	FNC130	Floating Point Sine	498
SMOV	FNC 13	Shift Move	229
SORT	FNC 69	Sort Tabulated Data	398
SORT2	FNC149	Sort Tabulated Data 2	525
SPD	FNC 56	Speed Detection	350
SQR	FNC 48	Square Root	315
SRET	FNC 02	Subroutine Return	206
STMR	FNC 65	Special Timer	389
STOH	FNC165	Second to Hour Conversion	556
STR	FNC200	BIN to Character String Conversion	602
SUB	FNC 21	Subtraction	252
SUM	FNC 43	Sum of Active Bits	305
SWAP	FNC147	Byte Swap	524
T			
TADD	FNC162	RTC Data Addition	550
TAN	FNC132	Floating Point Tangent	501
TBL	FNC152	Batch Data Positioning Mode	534
TCMP	FNC160	RTC Data Compare	545
TKY	FNC 70	Ten Key Input	402
TO	FNC 79	Write To a Special Function Block	435
TRD	FNC166	Read RTC data	558
TSUB	FNC163	RTC Data Subtraction	552
TTMR	FNC 64	Teaching Timer	387
TWR	FNC167	Set RTC data	559
TZCP	FNC161	RTC Data Zone Compare	547
U			
UNI	FNC143	4-bit Linking of Word Data	520
V			
VAL	FNC201	Character String to BIN Conversion	607

Mnemonic	FNC No.	Function	Ref. Page
W			
WAND	FNC 26	Logical Word AND	264
WBFM	FNC279	Divided BFM Write	704
WDT	FNC 07	Watchdog Timer Refresh	213
WOR	FNC 27	Logical Word OR	266
WR3A	FNC177	Write to Dedicated Analog Block	568
WSFL	FNC 37	Word Shift Left	289
WSFR	FNC 36	Word Shift Right	287
WSUM	FNC140	Sum of Word Data	513
WTOB	FNC141	WORD to BYTE	515
WXOR	FNC 28	Logical Exclusive OR	268
X			
XCH	FNC 17	Exchange	240
Z			
ZCP	FNC 11	Zone Compare	223
ZONE	FNC258	Zone Control	666
ZPOP	FNC103	Batch POP of Index Register	461
ZPUSH	FNC102	Batch Store of Index Register	458
ZRN	FNC156	Zero Return	536
ZRST	FNC 40	Zone Reset	297

MEMO

Warranty

Please confirm the following product warranty details before using this product.

1. **Gratis Warranty Term and Gratis Warranty Range**
If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company. However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]
The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]
1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
   a) Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
   b) Failure caused by unapproved modifications, etc., to the product by the user.
   c) When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
   d) Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
   e) Relay failure or output contact failure caused by usage beyond the specified Life of contact (cycles).
   f) Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
   g) Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
   h) Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. **Onerous repair term after discontinuation of production**
   1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued.
   Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.
   2) Product supply (including repair parts) is not available after production is discontinued.

3. **Overseas service**
Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. **Exclusion of loss in opportunity and secondary loss from warranty liability**
Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation of damages caused by any cause found not to be the responsibility of Mitsubishi, loss in opportunity, lost profits incurred to the user or third person by Failures of Mitsubishi products, special damages and secondary damages whether foreseeable or not , compensation for accidents, and compensation for damages to products other than Mitsubishi products, replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. **Changes in product specifications**
The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

6. **Product application**
   1) In using the Mitsubishi MELSEC programmable logic controller, the usage conditions shall be that the application will not lead to a major accident even if any problem or fault should occur in the programmable logic controller device, and that backup and fail-safe functions are systematically provided outside of the device for any problem or fault.
   2) The Mitsubishi programmable logic controller has been designed and manufactured for applications in general industries, etc. Thus, applications in which the public could be affected such as in nuclear power plants and other power plants operated by respective power companies, and applications in which a special quality assurance system is required, such as for Railway companies or Public service purposes shall be excluded from the programmable logic controller applications.
   In addition, applications in which human life or property that could be greatly affected, such as in aircraft, medical applications, incineration and fuel devices, manned transportation, equipment for recreation and amusement, and safety devices, shall also be excluded from the programmable logic controller range of applications.
   However, in certain cases, some applications may be possible, providing the user consults their local Mitsubishi representative outlining the special requirements of the project, and providing that all parties concerned agree to the special circumstances, solely at the users discretion.

Revised History

<table>
<tr><th>Date Created</th><th>Revision</th><th>Discription</th></tr>
<tr><td>7/2005</td><td>A</td><td>First Edition</td></tr>
<tr><td>2/2006</td><td>B</td><td>
FX3U and FX3UC series version 2.30 compatible
• Two instructions are added.
- MEP, MEF [Section 3.1, Chapter 7, Appendix A-1-2, Appendix 8-1]
• Functions of instructions are added
- MUL (FNC 22) [Section 6.5.2, Section 10.3, Section 36.1.1]
- DIV (FNC 23) [Section 6.5.2, Section 10.4, Section 36.1.1]
- RS2 (FNC 87) [Section 36.1.1]
• Example of index modification for instructions having limitation on number of executions [Section 5.7.3]
• Other
- Configuration
<table>
<tr><th>Revision A</th><th>Revision B</th></tr>
<tr><td>7.11 PLS, PLF</td><td>7.11 MEP, MEF</td></tr>
<tr><td>7.12 SET, RST</td><td>7.12 PLS, PLF</td></tr>
<tr><td>7.13 NOP</td><td>7.13 SET, RST</td></tr>
<tr><td>7.14 END</td><td>7.14 NOP</td></tr>
<tr><td>7.15 Number of Instruction Steps and Specified Devices</td><td>7.15 END</td></tr>
<tr><td>-</td><td>7.16 Number of Instruction Steps and Specified Devices</td></tr>
<tr><td>Appendix A-2 Version Upgrade History</td><td>Appendix A-2 Peripheral Products applicability (except programming tools)</td></tr>
<tr><td>Appendix A-2-1 Version check method</td><td>Appendix A-2-1 Applicable products and versions</td></tr>
<tr><td>Appendix A-2-2 How to look at manufacturer's serial number</td><td>Appendix A-2-2 Incompatible peripheral products</td></tr>
<tr><td>-</td><td>Appendix A-3 Version Upgrade History</td></tr>
<tr><td>-</td><td>Appendix A-3-1 Version check method</td></tr>
<tr><td>-</td><td>Appendix A-3-2 How to look at manufacturer's serial number</td></tr>
<tr><td>-</td><td>Appendix A-3-3 Version Upgrade history [FX3U]</td></tr>
<tr><td>-</td><td>Appendix A-3-4 Version Upgrade history [FX3UC]</td></tr>
</table>
- FX3U transistor output compatibility (change the [illegible] from FX3UC to FX3U)[Section 13.3, Item 13.3.1, Section 13.8, Section 13.9, Section 13.10, Section 15.2, Section 15.3, Section 15.5]
</td></tr>
</table>

PROGRAMMING MANUAL - Basic & Applied Instructions Edition

FX3U/FX3UC SERIES PROGRAMMABLE CONTROLLERS

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
HIMEJI WORKS: 840, CHIYODA CHO, HIMEJI, JAPAN

MODEL	FX-P3-E
MODEL CODE	09R517

JY997D16601B
(MEE)

Effective Feb. 2006
Specifications are subject to change without notice.