Changes for the Better

MITSUBISHI ELECTRIC

CNC

MELDAS C6/C64

PLC PROGRAMMING MANUAL

(Ladder Section with MELSEC Tool)

BNP-B2309D(ENG)

Introduction

These specifications are the programming manual used when creating the sequence program with the PLC development software, or Mitsubishi Electric Co.'s integrated FA software MELSOFT series (GX Developer).
The PLC (Programmable Logic Controller) is largely divided into the basic commands, function commands and exclusive commands, and ample command types are available. The commands can be used according to the purpose and application such as the PLC support function used when supporting the user PLCs.
In addition to the explanation of commands and functions, the environment to develop the user PLC using GX Developer, especially the usage unique to MELDAS, is described.

⚠ CAUTION

- ⚠ For items described as "Restrictions" or "Usable State" in this manual, the instruction manual issued by the machine manufacturer takes precedence over this manual.
- ⚠ An effort has been made to describe special handling of this machine, but items that are not described must be interpreted as "not possible".
- ⚠ This manual is written on the assumption that all option functions are added. Refer to the specifications issued by the machine manufacturer before starting use.
- ⚠ Refer to the Instruction Manual issued by each machine manufacturer for details on each machine tool.
- ⚠ Some screens and functions may differ or some functions may not be usable depending on the NC version.

Refer to the related operation manuals for details of GX Developer and GX Converter usage.

[Documents relating to MELDAS C6/C64]

MELDAS C6/C64/C64T PLC Interface Manual BNP-B2261
MELDAS C6/C64 Network Manual.. BNP-B2373

Precautions for Safety

Always read the specifications issued by the machine manufacturer, this manual, related manuals and attached documents before installation, operation, programming, maintenance or inspection to ensure correct use.
Understand this numerical controller, safety items and cautions before using the unit.
This manual ranks the safety precautions into "DANGER", "WARNING" and "CAUTION".

⚠ **DANGER**	When there is a great risk that the user could be subject to fatalities or serious injuries if handling is mistaken.
⚠ **WARNING**	When the user could be subject to fatalities or serious injuries if handling is mistaken.
⚠ **CAUTION**	When the user could be subject to injuries or when physical damage could occur if handling is mistaken.

Note that even items ranked as " ⚠ **CAUTION**", may lead to major results depending on the situation. In any case, important information that must always be observed is described.

⚠ DANGER

Not applicable in this manual.

⚠ WARNING

Not applicable in this manual.

⚠ CAUTION

1. Items related to product and manual

- ⚠ For items described as "Restrictions" or "Usable State" in this manual, the instruction manual issued by the machine manufacturer takes precedence over this manual.
- ⚠ An effort has been made to describe special handling of this machine, but items that are not described must be interpreted as "not possible".
- ⚠ This manual is written on the assumption that all option functions are added. Refer to the specifications issued by the machine manufacturer before starting use.
- ⚠ Refer to the Instruction Manual issued by each machine manufacturer for details on each machine tool.
- ⚠ Some screens and functions may differ or some functions may not be usable depending on the NC version.

2. Items related to start up and maintenance

- ⚠ Read this manual carefully and confirm the safety enough before executing the operation of the program change, forced output, RUN, STOP, etc. during operation. Operation mistakes may cause damage of the machine and accidents.

⚠ CAUTION

3. Items related to program development

- ⚠ Always observe the cautions before development to develop a program.
- ⚠ If the data transferred does not follow the file name rule, the CNC will mistake it for another data, resulting in unexpected operation, e.g. PLC program erasure.
- ⚠ Do not read a sequence program on which a conversion error occurred into the GX Developer. The file may include unexpected contents to result an illegal operation.
- ⚠ When an error occurred at GX Developer On-line function, the error message may not explain exactly the state in the CNC side.
  Always refer to the error list.
- ⚠ When initializing PLC data storage area is performed, all sequence programs and messages currently stored in the CNC will be erased. Do not use this operation other than when the error cannot be solved.

Contents

1. PLC Development Environment Using GX Developer 1
1.1 Function 1
1.1.1 Development Environment Configuration 1
1.1.2 Software Configuration 1
1.1.3 GX Developer Functions Supported by C64 Series 3
1.1.3.1 Function Support Conditions (general section) 3
1.1.3.2 Function Support Conditions (on-line section) 7
1.2 Setup 10
1.2.1 Installing the Tools 10
1.2.2 Connecting the Serial Cable 10
1.3 Developing PLC Programs 11
1.3.1 Precautions before Development 11
1.3.2 Creating a New Program 13
1.3.3 Specifying the Connection Target 14
1.3.4 Starting/Stopping the PLC of the CNC 16
1.3.5 Writing the PLC Program to the CNC 17
1.3.6 Reading the PLC Program from the CNC 20
1.3.7 Verifying the PLC Programs 21
1.3.8 Monitoring the PLC Program 22
1.3.9 Diverting the PLC program that was developed using PLC4B 23
1.4 Creating PLC Message Data 27
1.4.1 Development Procedure 27
1.4.2 Message Data Description Method 29
1.4.3 Converting Data into GX Developer Format 34
1.4.4 Entering/Editing Data Using GX Developer 36
1.4.5 Writing to the CNC 39
1.4.6 Reading and Verifying from the CNC 39
1.5 Creating Device Comments 41
1.5.1 Development Procedure 41
1.5.2 Description Method for Indirect Entry 42
1.5.3 Converting Comment Data into GX Developer Data 43
1.5.4 Writing Comment Data to the CNC 45
1.6 PLC4B PLC Development Environment (M500) and Differences 46
1.6.1 Development Tools, etc. 46
1.6.2 PLC Commands 47
1.7 Error Status 49
1.8 Initializing for PLC Data Storage Area 51
1.8.1 Operation procedure 51

2. PLC Processing Program 52
2.1 PLC Processing Program Level and Operation 52
2.1.1 High-speed processing program and main processing program 52
2.1.2 Cautions on high-speed processing programming 53
2.2 Multi-Programming Function 54
2.2.1 Program Registration Numbers 54
2.2.2 Program Execution Order 54
2.2.3 Precautions 54
2.3 User Memory Area Configuration 54

3. Input/Output Signals 55
3.1 Input/Output Signal Types and Processing 55
3.2 Handling of Input Signals Designated for High-Speed Input 56

3.3 High-Speed Input/output Designation Method 57

4. Parameters 58
4.1 PLC Constants....58
4.2 Bit Selection Parameters 59

5. Explanation of Devices 63
5.1 Devices and Device Numbers 63
5.2 Device List....63
5.3 Detailed Explanation of Devices....64
5.3.1 Input/output X, Y 64
5.3.2 Internal Relays M and F, Latch Relay L....65
5.3.3 Special Relays SM....65
5.3.4 Link Relay B, Link Register W 66
5.3.5 Special Relay for Link SB, Special Register for Link SW....66
5.3.6 Timer T 67
5.3.7 Counter C....71
5.3.8 Data Register D....74
5.3.9 File Register R....75
5.3.10 Special Register SD....76
5.3.11 Index Register Z....76
5.3.12 Nesting N....77
5.3.13 Pointer P....77
5.3.14 Decimal Constant K 78
5.3.15 Hexadecimal Constant H 78
6. Explanation of Commands....79
6.1 Command List....79
6.1.1 Basic Commands....79
6.1.2 Function Commands....81
6.1.3 Exclusive Commands 1 94
6.1.4 Exclusive Commands 2 95
6.2 Command Formats....96
6.2.1 How to Read the Command Table 96
6.2.2 No. of Steps....97
6.2.3 END Command....98
6.2.4 Index Qualification....98
6.2.5 Digit Designation....100

7. Basic Commands (LD, LDI, AND, ANI, OR, ORI, ANB, ORB .....)....103

8. Function Commands (=, >, <, +, –, *, /, BCD, BIN, MOV .....)....145

9. Exclusive Commands 1 303

10. Exclusive Commands 2 334
10.1 ATC Exclusive Command....335
10.1.1 Outline of ATC Control....335
10.1.2 ATC Operation 335
10.1.3 Explanation of Terminology 335
10.1.4 Relationship between Tool Registration Screen and Magazines....336
10.1.5 Use of ATC and ROT Commands....337
10.1.6 Basic Format of ATC Exclusive Command 338
10.1.7 Command List....339
10.1.8 Control Data Buffer Contents....339
10.1.9 File Register (R Register) Assignment and Parameters 340
10.1.10 Details of Each Command....342
10.1.11 Precautions for Using ATC Exclusive Instructions 351

10.1.12 Examples of Tool Registration Screen 351
10.1.13 Display of Spindle Tool and Standby Tool....................353
10.2 S.ROT Commands 354
10.2.1 Command List....................354
10.3 Tool Life Management Exclusive Command....................360
10.3.1 Tool Life Management System....................360
10.3.2 Tool Command System 360
10.3.3 Spare Tool Selection System 361
10.3.4 Interface 361
10.3.5 User PLC Processing When the Tool Life Management Function Is Selected ...362
10.3.6 Examples of Tool Life Management Screen 370
10.4 DDB (Direct Data Bus) ... Asynchronous DDB....................371
10.4.1 Basic Format of Command....................371
10.4.2 Basic Format of Control Data....................371
10.5 External Search 374
10.5.1 Function....................374
10.5.2 Interface 374
10.5.3 Search Start Instruction 376
10.5.4 Timing Charts and Error Causes 376
10.5.5 Sequence Program Example....................378

11. PLC Help Function....................379
11.1 Alarm Message Display....................380
11.1.1 Interface 380
11.1.2 Message Creation....................381
11.1.3 F or R Type Selection Parameter 382
11.2 Operator Message Display....................383
11.2.1 Interface 383
11.2.2 Operator Message Preparation 384
11.2.3 Operator Message Display Validity Parameter 384
11.3 PLC Switches....................385
11.3.1 Explanation of Screen....................385
11.3.2 Explanation of Operation 386
11.3.3 Signal Processing 387
11.3.4 Switch Name Preparation 391
11.4 Key Operation by User PLC 392
11.4.1 Key Data Flow....................392
11.4.2 Key Operations That Can Be Performed....................392
11.4.3 Key Data Processing Timing 393
11.4.4 Layout of Keys on Communication Terminal....................394
11.4.5 List of Key Codes....................395
11.5 Load Meter Display....................396
11.5.1 Interface 396
11.6 External Machine Coordinate System Compensation 398
11.7 User PLC Version Display 399
11.7.1 Interface 399

12. PLC Axis Control 401
12.1 Outline....................401
12.2 Specifications....................401
12.2.1 Basic Specifications....................401
12.2.2 Other Restrictions 402
12.3 PLC Interface....................403
12.3.1 S.DDBS Function Command....................403
12.3.2 Control Information Data....................404
12.3.3 Control Information Data Details....................405
12.3.3.1 Commands 405
12.3.3.2 Status....................406

12.3.3.3 Alarm No. 414
12.3.3.4 Control Signals (PLC axis control information data) 415
12.3.3.5 Axis Designation 417
12.3.3.6 Operation Mode 417
12.3.3.7 Feedrate 418
12.3.3.8 Movement Data 418
12.3.3.9 Machine Position 419
12.3.3.10 Remaining Distance 419
12.3.4 Reference Point Return near Point Detection 420
12.3.5 Handle Feed Axis Selection 421

Appendix 1. Example of Faulty Circuit 422

Appendix 2. MELSEC QnA Series Command Lists 423
2.1 Sequence Commands 423
2.2 Basic Commands 424
2.3 Application Commands 429
2.4 Exclusive Commands 432

Appendix 3. PLC Development Environment using GPPQ 433
3.1 System Configuration at PLC Development 433
3.2 Development Tool Function Outline 433
3.2.1 CNVQ (data conversion software package) 433
3.2.2 LNKQ (sequence ladder generating connection function software package)........433
3.2.3 GPPQ (SW2IVD/NX-GPPQ type GPP Function Software Package) 434
3.3 GPPQ Function Outline and Functions Supported by the C64 Series 435
3.3.1 Function Support Conditions (general section) 435
3.3.2 Function Support Conditions (on-line section) 438
3.4 Setup Procedure 444
3.4.1 Tool Setup Procedure 444
3.4.2 Connection Procedure 444
3.5 PLC Program Development Procedure 445
3.5.1 Precautions before Development 445
3.5.2 Ladder Transfer to the C64 Controller 446
3.5.3 Ladder Read from the C64 Controller 448
3.5.4 Ladder Comparison with the C64 Controller 449
3.6 PLC-Related Data Development Procedure 450
3.6.1 PLC Related Data File Names 450
3.6.2 Development Procedure 451
3.6.3 Message Data Description Method 452
3.6.4 Conversion to GPPQ Data 457
3.6.5 Operation with the GPPQ 459
3.6.6 Transfer to the Controller 461
3.6.7 Reading and Comparing from the Controller 462
3.7 Differences From The M500 PLC Development Environment 464
3.7.1 PLC Commands 464
3.7.2 PLC Messages 466

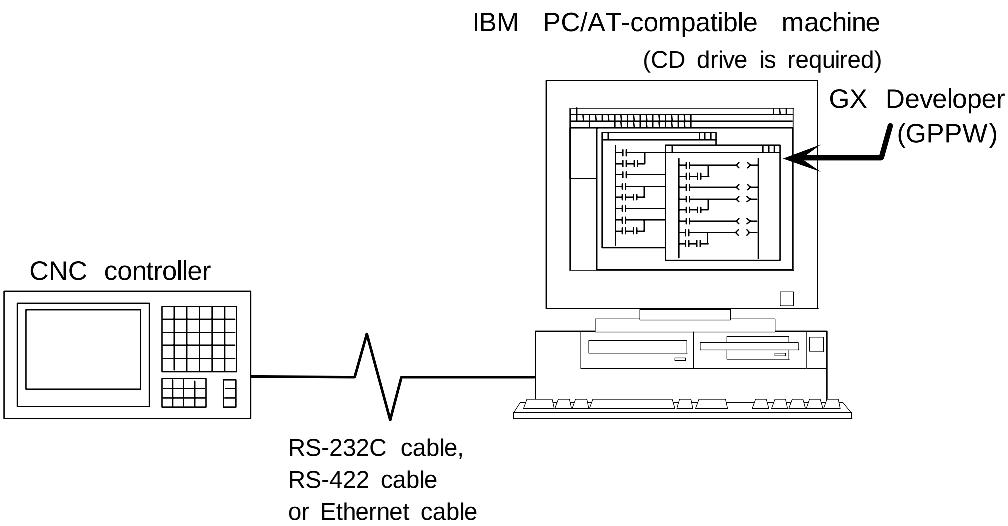
1. PLC Development Environment Using GX Developer

In the C64 Series, the user PLC development environment is supported using MELSEC PLC development tool, which is Mitsubishi integrated FA software MELSOFT series (GX Developer). This manual explains system configurations user PLC development environment using GX Developer, mainly usage specific to MELDAS.

1.1 Function

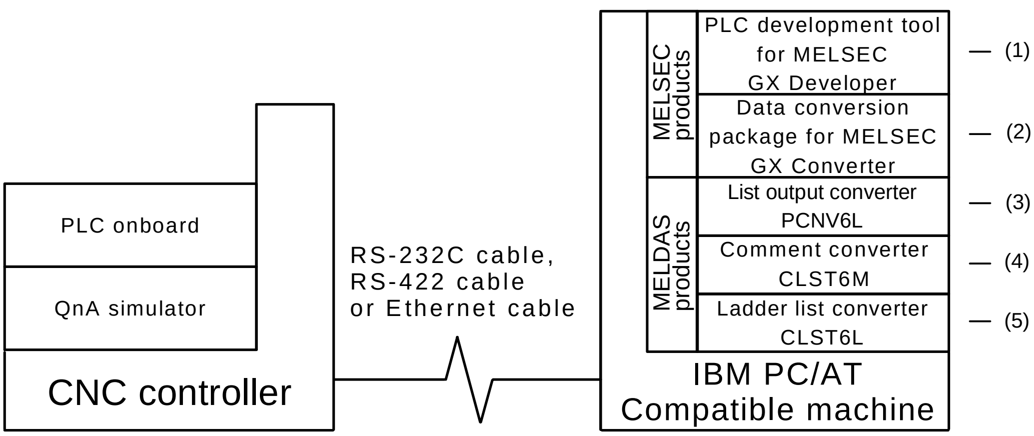
1.1.1 Development Environment Configuration

Most of the development works can be done by connecting the IBM PC/AT compatible machine and a CNC unit by an RS-232C cable or RS-422 cable and by executing the tools on the personal computer.



System configuration using GX Developer

1.1.2 Software Configuration



(1) GX Developer (PLC development software package for Windows)

GX Developer is a programming software package (model name: SW7D5C-GPPW) designed for Mitsubishi Electric's MELSEC series programmable logic controllers. The conventional function corresponding to MELDAS PLC development S/W (PLC4B) has been reinforced, and, furthermore, that is a strong tool added the monitoring function by way of RS-232C. Note that some functions specific to the "MELSEC series" may not be unavailable.
For MELDAS series ladder development, we recommend you to use GX Developer Version 4 (SW4D5C-GPPW) or later. For function details, refer to the operating manual supplied.
The DOS version "GPPQ" (SW2IVD/NX-GPPQ GPP function software package) of this package is also usable. Refer to "Appendix3. Operation Method Using GPPQ" for details.

(2) GX Converter (Data conversion software package for Windows)

The GX Converter is a tool that carries out file conversion of GX Developer data files and the following:

- Ladder list files and comment text files output by the CLIST6L
- Alarms and operator messages created by the text editor
- Data files of commercially available spreadsheet software, word processors and editors

This tool is add-on tool of the GX Developer, thus, start GX Converter from the GX Developer's menu.
This tool is a software package for MELSEC. GX Converter needs to be used with the versions following GX Developer Version 3 (SW3D5C-GPPW). Refer to the enclosed Operating Manual for function details.
The DOS version "CNVQ" (SW0IVD/NX-CNVQ data conversion software package) of this tool can also be used. Refer to "Appendix 3. Operation Methods Using GPPQ" for details.

(3) PCNV6L (List output converter)

This tool outputs a MELDAS specification ladder printing image with cross information into the text format from the GX Developer specification ladder list and comment data. Refer to the instruction manual for function details.
This tool works on the DOS of Windows.

(4) CLST6M (Device comment converter)

This tool outputs the contact/coil comment data of a user PLC ladder developed using PLC4B into the text format of the GX Developer specifications. The comment data developed using PLC4B can be used with GX Developer by using GX Converter to further convert the conversion results of this tool. Refer to the instruction manual for function details.
This tool works on the DOS of Windows.

(5) CLST6L (Ladder list converter)

This tool converts the user PLC ladder list data developed using PLC4B, and outputs the data in a ladder list format. The user PLC ladder developed using PLC4B can be used with the GX Developer by using the GX Converter to further convert the conversion results of this tool. Refer to the instruction manual for function details.
This tool works on the DOS of Windows.

1.1.3 GX Developer Functions Supported by C64 Series

The GX Developer functions explained here are those supported by the C64 Series in the "off-line functions" operated with the GX Developer independently and "on-line functions" carried out connected to the CNC controller.
Refer to the enclosed Operating Manual for details of respective functions.
Refer to "Appendix 3. Operation Methods Using GPPQ" for the GPPQ-specific functions and operations when the DOS version "GPPQ" (SW2IVD/NX-GPPQ GPP function software package) is used.

1.1.3.1 Function Support Conditions (general section)

The following shows a list of GX Developer outline functions supported by the C64 Series.
A ◎ mark indicates functions that can be used by the C64 Series. An ✕ mark indicates that the function cannot be used because it is related to "MELSEC Series" characteristic functions. The function details during on-line are described in the next section.

List of general section functions (1) ◎ : Possible, △ : Limitedly possible, ✕ : Not possible

Program type	Support	Remarks
Ladder	◎	
List	◎	
SFC	✕	
MELSAP-L	✕	
Function block	✕	

Function	Menu	Sub menu	Support	Remarks
Project	New project		◎	
	Open project		◎	
	Close project		◎	
	Save		◎	
	Save as		◎	
	Delete project		◎	
	Verify		◎	
	Copy		◎	
	Edit Data	New	◎	
		Copy	◎	
		Delete	◎	
		Rename	◎	
	Change PLC type		△	Fixed Q4A
	Import file	Import from GPPQ format file	◎	
		Import from GPPA format file	✕	
		Import from FXGP[WIN] format file	✕	
		Import from FXGP[DOS] format file	✕	
	Export file	Export to GPPQ format files	◎	
		Export to GPPA format files	✕	
		Export to FXGP[WIN] format file	✕	
		Export to FXGP[DOS] format file	✕	
		Export to TEXT ,CSV format file		

List of general section functions (2) ◎ : Possible, △ : Limitedly possible, ✕ : Not possible

Function	Menu	Sub menu	Support	Remarks
(Project)	Macro	Registration macros	◎	
		Macro utilize	◎	
		Delete macros	◎	
		Macro reference path	◎	
	Printer setup		◎	
	Print		◎	
	Start new GX Developer session		◎	
	Exit GX Developer		◎	
Edit	Undo		◎	
	Restore after ladder conversion		◎	
	Cut		◎	
	Copy		◎	
	Paste		◎	
	Insert line		◎	
	Delete line		◎	
	Insert row		◎	
	Delete row		◎	
	Insert NOP batch		◎	
	Delete NOP batch		◎	
	Draw line		◎	
	Delete line		◎	
	Change TC setting		◎	
	Read mode		◎	
	Write mode		◎	
	Ladder symbol	Open contact	◎	
		Close contact	◎	
		Open branch	◎	
		Close branch	◎	
		Coil	◎	
		Application instruction	◎	
		Vertical line	◎	
		Horizontal line	◎	
		Delete vertical line	◎	
		Delete horizontal line	◎	
		Rising pulse	◎	
		Falling pulse	◎	
		Rising pulse open branch	◎	
		Falling pulse close branch	◎	
		Invert operation results	◎	
		Convert operation results to rising pulse	◎	
		Convert operation results to falling pulse	◎	
	Documentation	Comment	◎	
		Statement	◎	
		Note	◎	
		Statement/Note block edit	◎	

List of general section functions (3) ◎ : Possible, △ : Limitedly possible, ✕ : Not possible

Function	Menu	Sub menu	Support	Remarks
Find/Replace	Find device		◎	
	Find instruction		◎	
	Find step no.		◎	
	Find character string		◎	
	Find contact or coil		◎	
	Replace device		◎	
	Replace instruction		◎	
	Change open/close contact		◎	
	Replace character string		◎	
	Change module start address		◎	
	Replace statement/note type		◎	
	Cross reference list		◎	
	List of used devices		◎	
Convert	Convert		◎	
	Convert (All programs being edited)		◎	
	Convert (Online change)		◎	
View	Comment		◎	
	Statement		◎	
	Note		◎	
	Alias		◎	
	Macro instruction format display		◎	
	Comment format	4*8 characters	◎	
		3*5 characters	◎	
	Alias format display	Replace device name and display	◎	
		Arrange with device and display	◎	
	Toolbar		◎	
	Status bar		◎	
	Zoom	50%	◎	
		75%	◎	
		100%	◎	
		150%	◎	
		Auto	◎	
	Project data list	Specify	◎	
	Instruction list		◎	
	Elapsed time		✕	
Online	Refer to "List of on-line section functions"	Refer to "List of on-line section functions"		
Diagnostics	PLC diagnostics		✕	
	Network diagnostics		✕	
	Ethernet diagnostics		✕	
	CC-Link diagnostics		✕	
	System monitor		✕	

List of general section functions (4) ◎ : Possible, △ : Limitedly possible, ✕ : Not possible

Function	Menu	Sub menu	Support	Remarks
Tools	Check program		◎	
	Merge data		◎	
	Check parameter		◎	
	Transfer ROM	Read	✕	
		Write	✕	
		Verify	✕	
		Write to file	✕	
	Delete unused comments		◎	
	Clear all parameters		◎	
	IC memory card	Read IC memory card	✕	
		Write IC memory card	✕	
		Read image data	✕	
		Write image data	✕	
	Start ladder logic test		✕	
	Set TEL data	Connection	✕	
		Disconnection	✕	
		TEL data	✕	
		AT command	✕	
		Call book	✕	
	Intelligent function utility	Utility list	✕	
	Customize keys		◎	
	Options		△	Partially impossible
	Create start-up setting file		◎	
Window	Cascade		◎	
	Tile vertically		◎	
	Tile horizontally		◎	
	Arrange icons		◎	
Help	PLC error		✕	
	Special relay/register		✕	
	Key operation list		◎	
	Product information		◎	
	Connect to MELFANSweb		◎	

1.1.3.2 Function Support Conditions (on-line section)

The following shows a list of GX Developer on-line functions supported by the C64 Series.
A ◎ mark indicates functions that can currently be used by the C64 Series. An ✕ mark indicates that the function cannot be used because it is related to "MELSEC Series" characteristic functions.

List of on-line section functions (1) ◎ : Possible, △ : Limitedly possible, ✕ : Not possible

Menu	Sub menu	Detailed function	Support	Remarks
Transfer setup		PC side I/F	△	
		PLC side I/F	△	Only for QnACPU
		Other station	△	
		Network route	△	
		Co-existence network route	△	
Read from PLC		Target memory	△	Only for internal memory
		Title	◎	
		File selection	◎	
		Device data	✕	
		Program	✕	
		Common	✕	
		Local	✕	
		Refresh view	◎	
		Free space volume	◎	
		Create title	✕	
Write to PLC		Target memory	△	Only for internal memory
		Title	◎	
		File selection	◎	
		Device data	✕	
		Program	✕	
		Common	✕	
		Local	✕	
		Free space volume	◎	
		Create title	✕	
Verify with PLC		Target memory	△	Only for internal memory
		Title	◎	
		File selection	◎	
		Program	✕	
		Refresh view	◎	
		Free space volume	◎	
		Create title	✕	
Write to PLC [Flash ROM]	Write the program memory to ROM		✕	
	Write to PLC [Flash ROM]		✕	
Delete PLC data		Target memory	△	Only for internal memory
		Title	◎	
		File selection	◎	
		Refresh view	◎	
		Free space volume	◎	
		Create title	✕	
Change PLC data attributes			✕	
PLC user data	Read PLC user data		✕	
	Write PLC user data		✕	
	Delete PLC user data		✕	

List of on-line section functions (2) ◎ : Possible, △ : Limitedly possible, ✕ : Not possible

Menu	Sub menu	Detailed function	Support	Remarks
Monitor	Monitor mode	ON/OFF state	◎	
		Scan time	◎	
		PLC status	◎	
	Monitor [Write mode]		◎	
	Start monitor [All windows]		◎	
	Stop monitor [All windows]		◎	
	Start monitor		◎	
	Stop monitor		◎	
	Change current value monitor [Decimal]		◎	
	Change current value monitor [Hexadecimal]		◎	
	Device batch	Device	◎	
		Connect	◎	
		Coil	◎	
		Setting value	◎	
		Current value	◎	
		Monitor format : Bit & word	◎	
		Monitor format : Bit	◎	
		Monitor format : word	◎	
		Display : 16bit integer	◎	
		Display : 32bit integer	◎	
		Display : Real number	✕	
		Display : ASCII character	✕	
		Value : DEC	◎	
		Value : HEX	◎	
		T/C set value Reference program	◎	
		Device test	◎	
	Entry data monitor	Device	◎	
		ON/OFF/Current	◎	
		Setting value	◎	
		Connect	◎	
		Coil	◎	
		Display : 16bit integer	◎	
		Display : 32bit integer	◎	
		Display : Real number	✕	
		Display : ASCII character	✕	
		Value : DEC	◎	
		Value : HEX	◎	
		T/C setting value, Local label Reference program	◎	
		Device test	◎	
	Buffer memory batch		✕	
	Monitor condition setup	Device	◎	
		Step No.	◎	
	Monitor stop condition setup	Device	◎	
		Step No.	◎	
	Program monitor list		✕	
	Interrupt program monitor list		✕	
	Scan time measurement		✕	
	Entry ladder monitor		◎	
	Delete all entry ladder		◎	

List of on-line section functions (3) ◎ : Possible, △ : Limitedly possible, ✕ : Not possible

Menu	Sub menu	Detailed function	Support	Remarks
Debug	Device test	FORCE ON	◎	
		FORCE OFF	◎	
		Toggle force	◎	
		Device	◎	
		Buffer memory	✕	
	Debug		✕	
	Skip execution		✕	
	Partial execution		✕	
	Step execution		✕	
Trace			✕	
Remote operation		PLC status	◎	
		RUN	◎	
		STOP	◎	
		PAUSE	✕	
		Latch clear	✕	
		STEP-RUN	✕	
		Reset	✕	
		Operation during RUN, STEP-RUN	✕	
		Specify execution destination	✕	
Keyword setup	Register		✕	
	Delete		✕	
	Disable		✕	
Clear PLC memory			✕	
Format PLC memory		Target memory	△	For only internal RAM
		Format	✕	
Arrange PLC memory			✕	
Set time		Date / time	◎	
		Day of week	✕	
		Specify execution destination	✕	

1.2 Setup

1.2.1 Installing the Tools

In the C64 Series PLC development environment, it is assumed that the various tools are used on an IBM PC/AT compatible machine. Prepare each tool so that it is IBM PC/AT compatible machine. Refer to the enclosed Operating Manual for the setup and start procedures of each tool.

1.2.2 Connecting the Serial Cable

As for the serial port connected with the CNC, refer to the MELDAS C6/C64/C64T Connection and Maintenance Manual (BNP-B2255).

(1) RS-232C connection

Between the IBM PC/AT compatible machine that uses GX Developer and the CNC controller, use an RS-232C serial cable equivalent to the one shown below in the RS-232C connection diagram.

(Note) The cables given in the connection diagrams of the GX Developer Operating Manual cannot be used.

As for the CNC side, setting for GPPW communication is not necessary.

(a) When connecting with C64 controller directly

NC side (TERMINAL) (20-pin half-pitch)		Cable connection and signal direction	Personal computer side (9-pin D-SUB)	
Signal name	Pin No.		Pin No.	Signal name
SD	6	→	2	RD
RD	16	←	3	SD
ER(DTR)	18	→	6	DR(DSR)
		→	8	CS(CTS)
GND	1	↔	1	GND
GND	11	↔	5	GND

(b) When connecting with C64 controller using the intermediate cable dedicated to C64

NC side (TERMINAL) (25-pin D-SUB)		Cable connection and signal direction	Personal computer side (9-pin D-SUB)	
Signal name	Pin No.		Pin No.	Signal name
SD	2	→	2	RD
RD	3	←	3	SD
ER(DTR)	20	→	6	DR(DSR)
		→	8	CS(CTS)
GND	1	↔	1	GND
GND	7	↔	5	GND

(2) RS-422 connection
MELSEC-dedicated cable can be used to connect with C64 controller.
Refer to the GX Developer Operating Manual for details.

(3) Ethernet connection
For the connection using Ethernet, the Ethernet card (FCU6-EX875) must be mounted to the extension slot on the control unit.
Connect the Ethernet cable to the modular jack on the Ethernet card.

1.3 Developing PLC Programs

1.3.1 Precautions before Development

Pay careful attention to the following items before developing PLC programs using the GX Developer.

(1) PC type selection

The PLC type must be set when newly creating programs, etc. Select the following CPU type when requested to select the PLC type with the GX Developer. An error will occur during transfer of the PLC program to the CNC if another PLC type is selected.

Select "Q4A" for CPU type.

(2) Device setting

Do not set the devices when developing the PLC program for the CNC. Develop the program with the device settings (No. of points, etc.) left at their default values applied when GX Developer was started. The PLC program cannot be transferred to the CNC normally when it is developed with settings other than the default values.

Do not set the devices.

(3) PLC commands

MELSEC-specific PLC commands cannot be used in the PLC program development for the CNC.
Refer to “6.1 Command List Table” and confirm the useable commands.
The format, etc., are changed with some commands. Refer to "1.5 PLC4B PLC Development Environment and Differences" for details.

MELSEC-specific PLC commands cannot be used.

(4) Label at the beginning of ladder program

In a MELDAS PLC program, a processing unit is differentiated by specifying a reserved label number at the beginning of processing. There are the following different processing units.

P251, P360 to 368 : PLC high-speed processing program starting label
P252, P370 to 378 : PLC main processing program starting label

Even if only the PLC main processing is to be performed, do not omit but describe the above label at the beginning of a PLC program. Unless the label is described, normal RUN cannot be performed.

Specify a label at the beginning of a PLC program.

(5) Statements and notes

GX Developer allows a PLC program to be commented (with interlinear statements and notes). They are available in two types: integrated and peripheral.

Integrated type : Can be downloaded together with a ladder program to the CNC controller.
Peripheral type : Cannot be downloaded.

The integrated type cannot be used with the C64 series. If it is used, a PLC program cannot be transferred to the CNC properly.

Do not use integrated type interlinear statements and notes.
Create the message data as the integrated type statement.

(6) File name

Inside the C64 series, PLC-related data are controlled and stored in the following categories. Therefore, they are also developed in the same categories.

⚠ CAUTION

⚠ If the data transferred does not follow the following file name rule, the NC will mistake it for another data, resulting in unexpected operation, e.g. PLC program erasure.

File name rule

M1 xxxx. WPG or WCD

- WPG or WCD: Extensions .... Automatically attached with the GX Developer (expresses file classification)
- xxxx.: Random file name .... User free designation
- M1: Data classification and language classification (M1: Message 1st language) .... user fixed designation

When data is transferred by GX Developer, its data type is distinguished by the file name.
An extension indicates a file type, and the first two characters denote a data type and a language type.
File name can be specified within eight characters including data classification and language classification freely with the exception of the extension.

(Note) File name that is not permitted.
The following characters at the head of file name are reserved for CNC side.
Do not use the file name of these characters combination.
Data classification : "M", "C", "H" (alphabet)
Language classification : "0" to "9" (number)

List of PLC-related data

	Related data classification	File name (GX Developer)	Remarks
1 2	PLC program (ladder) PLC program comment	zzzzzz.WPG zzzzzz.WCD	PLC ladder code Comment data for GX Developer
3	Message 1st language	M1xxxx.WPG	Message 1st language data such as alarm messages/ operator messages/ PLC switches/ tool registration comment/ load meter comment
4	Message 2nd language	M2xxxx.WPG	Same as above (2nd language data)

(a) PLC program (ladder)

- PLC program developed using GX Developer.
- Only one file can be stored in the NC.

(b) PLC program comment

- Program comment for GX Developer display
- Only one file can be stored in the NC with the same file name as the PLC program.
- A device comment (32 characters) and a device name (10 characters) can be defined for each device.
- Stored mainly when it is read to GX Developer and used as a comment.

(c) Message 1st language and (4) 2nd language

- Alarm message/operator message/PLC switch/comment message data.
- One 1st language file and one 2nd language file can be stored in the NC.
- The messages can be handled and edited as "integrated type interlinear statements" by GX Developer.
- The maximum message length and the number of messages can be specified for each message type.

1.3.2 Creating a New Program

Create a new program using GX Developer.
Refer to the GX Developer's operation manual for details of usage.

1.3.3 Specifying the Connection Target

You must specify the connection target before performing on-line operations from GX Developer to the CNC.

(1) Operation procedure

Perform the following operation from GX Developer to start the setting screen.

[Online] → [Transfer setup]

Set only the following items. Leave the other items unchanged from the initial values.

(a) Setting for serial connection (RS-232C or RS-422)

• Personal computer side I/F : Serial		
	Serial port name	COM1 or COM2
	Baud rate	19.2Kbps
• PLC side IF : CPU unit		

(b) Setting example for Ethernet connection

• Personal computer side I/F : Ethernet		
	Network No.	1
	Station No.	1
• PLC side I/F : Ethernet unit		
	Network No.	1
	Station No.	2
	IP address	(Address set by NC parameter)
	Routing method	Automatic conversion method

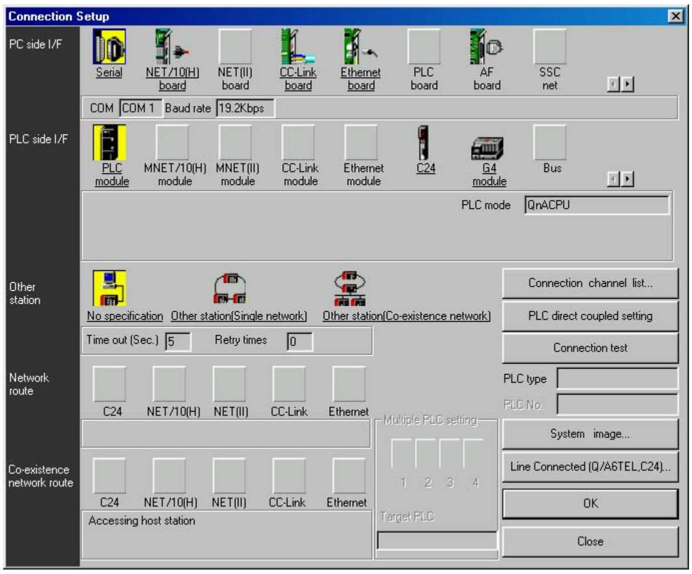
• CNC side IP address setting

# No.	Parameter name	Setting example
#1926	IP address	192.182.1.2
#1927	Subnet mask	255.255.255.0
#1928	Gateway address	192.182.1.254
#1929	Port number	64758

The local station’s IP address is set for the NC side.
The IP address set here is shared with the other Ethernet communication functions (GOT connection, etc.).

(Note) The setting example above is for when the GX Developer is connected to the CNC. To connect the GX Developer to the Ethernet or multiple hierarchical network of MELSECNET/10, setting values will differ.

• Setting the GX Developer Connection Destination
These parameters are used to set the GX Developer connection method. These parameters are included in the GX Developer project data.



1.3.4 Starting/Stopping the PLC of the CNC

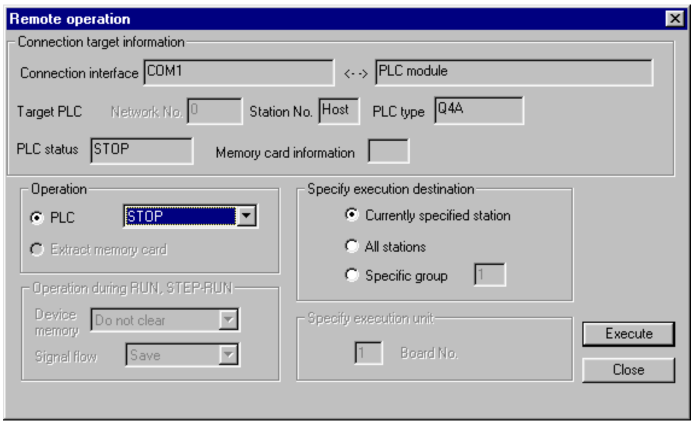
Before writing a ladder program, you must stop the PLC of the CNC.

(1) Operation procedure

Perform the following operation from GX Developer to start the operation screen.

[Online] → [Remote operation] or Alt + 6

On the following screen, set "STOP" or "RUN" in the [PLC] part under [Operation] and click [Execute]. The current status is displayed in [PLC status] under [Connection target information].



(Note) The operation other than "RUN" or "STOP" can not be performed.

The operation is completed when the following dialog appears. Click [OK]. The status after completion appears in [PLC status] on the remote operation screen displayed behind. If the status does not change, check whether an alarm is displayed or not on the CNC side.

1.3.5 Writing the PLC Program to the CNC

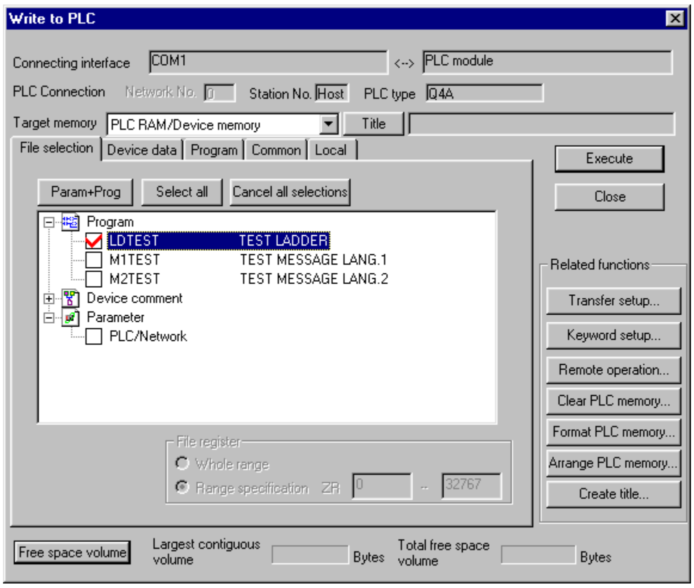
The following indicates how to write ladders from GX Developer to the CNC (especially the restrictions and C64 series-specific operations).

(1) Operation procedure

Perform the following operation from GX Developer to start the operation screen.

[Online] → [Write to PLC]

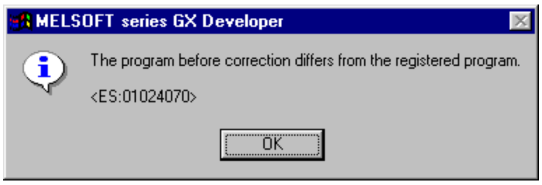
On the following screen, choose the ladder file to be written from the [File selection] tab and click [Execute].
You can command RUN/STOP of the PLC using [Remote operation] under [Related functions].



(Note) As [Target memory], only [PLC RAM/Device memory] is valid.
Do not set the other tabs ([Device data], [Program], [Common], [Local]) than [File selection].

(2) Operation to be performed at write error

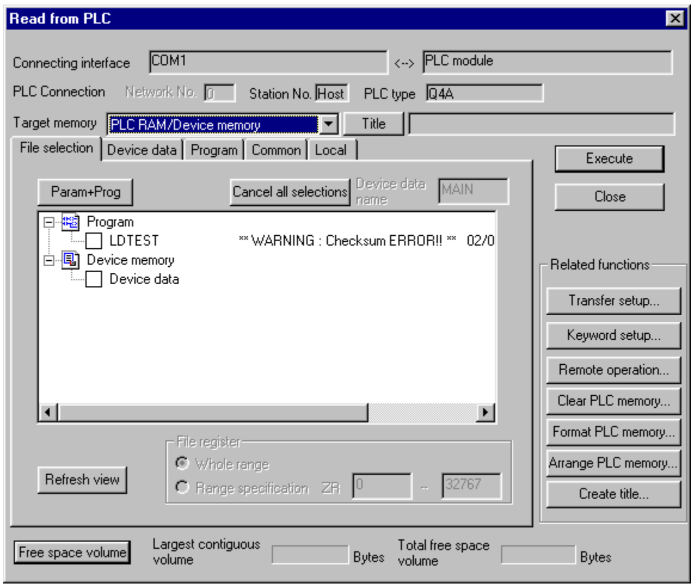
As soon as a ladder is written from GX Developer to the CNC, the CNC converts it into the CNC-specific ladder machine code. A conversion error occurs if any of the devices and command formats not supported by the C64 series is used.
At a conversion error, the CNC side ladder machine code is converted into the [NOP code], the ladder up to the last step is transferred, and the following dialog is then displayed on the GX Developer screen.



When the file that resulted in a conversion error is displayed with the [File selection] tab of the [Read from PLC] screen, the following warning appears in the title field.

"∗∗ WARNING Checksum ERROR!! ∗∗"

If you execute RUN the PLC as-is, an alarm occurs on the CNC side and the PLC does not RUN.



⚠ CAUTION

⚠ Do not read a sequence program on which a conversion error occurred into the GX Developer.
The file may include unexpected contents to result an illegal operation.

(3) How to confirm the error step number

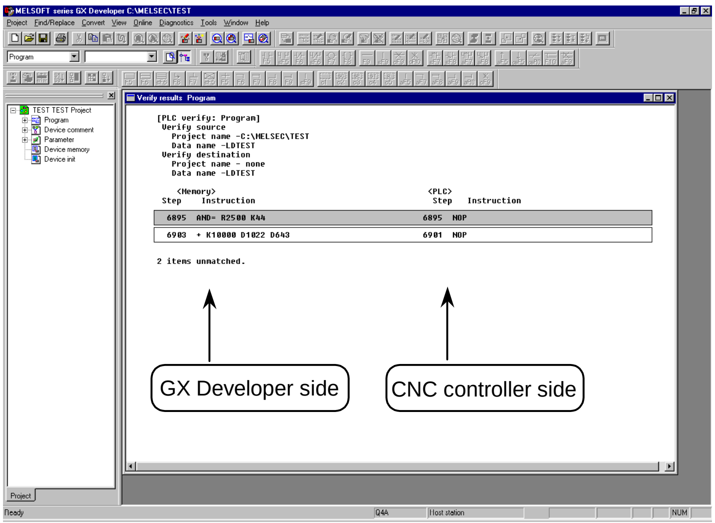
At a conversion error, error information is stored to the special registers as below. Device-monitoring these registers enables to find the error position.

SD30 : Error step No. where the error occurred.
SD31 : Cause of the error.

The error No. which occurred while writing from GX Developer to C64

SD31 value	Cause of error
1	Command format error.
2	File already exists.
3	File to be read is not found.
4	Object code error.
5	File can not be opened.
6	File name is too long.
7	File name is illegal.
9	This is not a ladder or a message file.
10	Ladder code has been already changed.
20	Device No. exceeds the specification range.
21	Search was failed.
22	Device No. is illegal.
23	There is an error in the conversion table data.
24	Conversion can not be performed. (Already converted with M500.)
25	Attribute code is illegal.
101	Warning : Device can not be converted.
102	Some of the designated digits can not be used in the ladder sub-routine.
103	Designated index can not be used in the ladder sub-routine.

The [Verify with PLC] function can be used to confirm the error step. Executing verification with PLC displays mismatches as in the following example. For details of the [Verify with PLC] function, refer to "1.3.7 Verifying the PLC Programs".
<Memory> indicates the GX Developer side, and <PLC> the CNC side.



1.3.6 Reading the PLC Program from the CNC

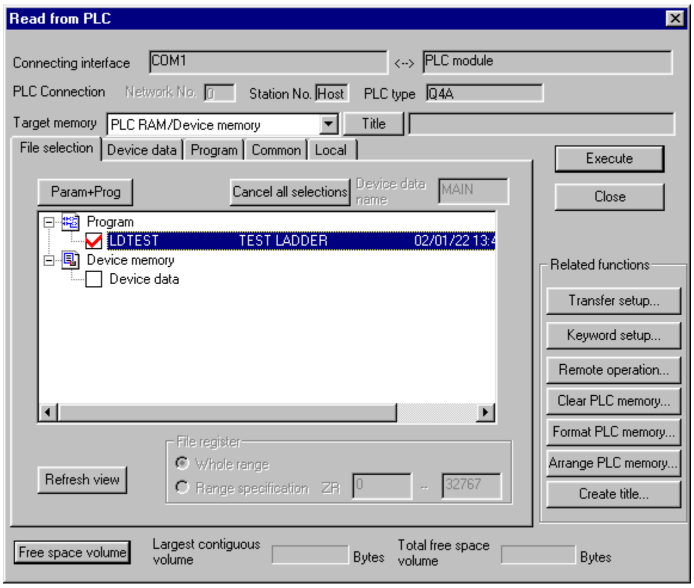
The following indicates how to read a ladder from the CNC to GX Developer.

(1) Operation procedure

Perform the following operation from GX Developer to start the operation screen.

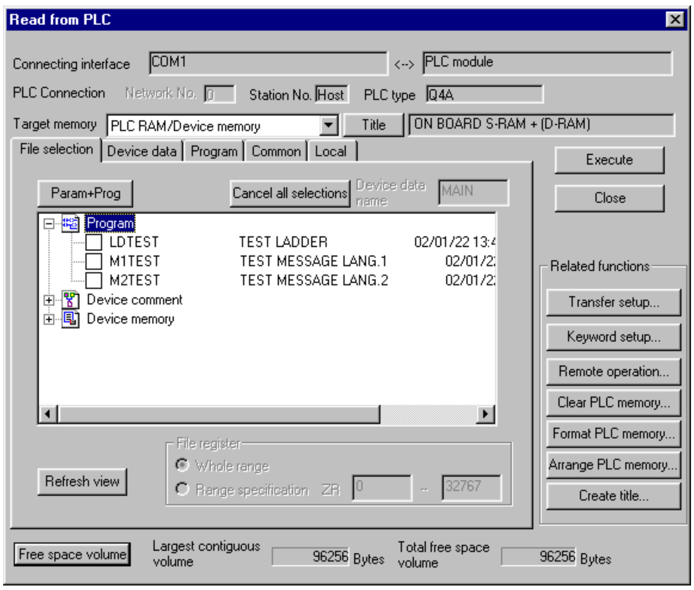
[Online] → [Read from PLC]

On the following screen, choose the ladder file to be read from the [File selection] tab, and click [Execute].



(Note) As [Target memory], the fitted memory is valid.
Do not set the other tabs ([Device data], [Program], [Common], [Local]) than [File selection].

The [Read from PLC] screen can also be used as a CNC side file listing function. Move the scroll bar of the [File selection] tab to the right to display the write date and size of each file. Click [Free space volume] to display the free area of the target memory.



1.3.7 Verifying the PLC Programs

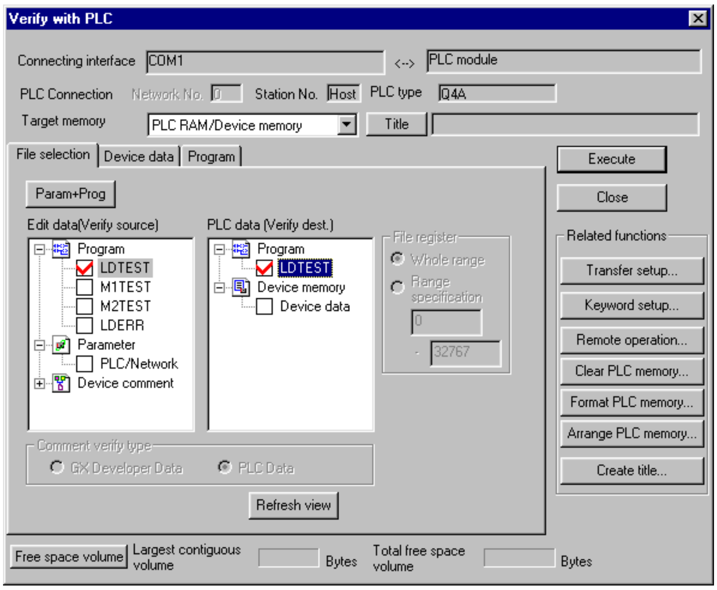
The following indicates how to verify ladders between the CNC and GX Developer.

(1) Operation procedure

Perform the following operation from GX Developer to start the operation screen.

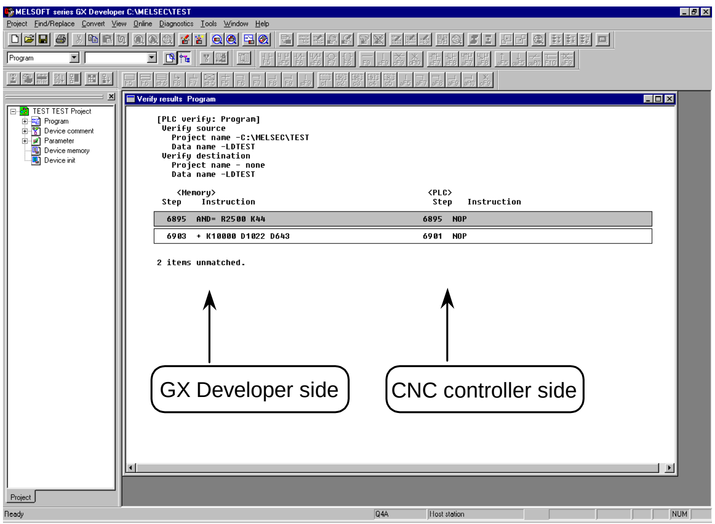
[Online] → [Verify with PLC]

On the following screen, choose the ladder files to be verified with the [File selection] tab, and click [Execute].
[Verify source] : GX Developer side [Verify dest] : CNC side



(Note) As [Target memory], the fitted memory is valid.
Do not set the other tab ([Program], [Device data]) than [File selection].

If verification mismatches occur, the following mismatch screen appears. Double-click the mismatch to display the corresponding part of the GX Developer side file.



1.3.8 Monitoring the PLC Program

There are no MELDAS-specific operations to monitor a PLC program. Refer to the GX Developer operating manual for the operation methods. For usable functions, refer to "1.1.3.2 Function Support Conditions (on-line section)". This section explains the operation procedure outline and precautions.

(1) Operation procedure

Perform the following operation from GX Developer to start monitoring.

(a) Display the ladder program to be monitored and move to the circuit part to be monitored.

(b) Perform the following operation to start monitoring.

[Online] → [Monitor] → [Monitor mode] or F3

(c) Perform the following operation to stop monitoring.

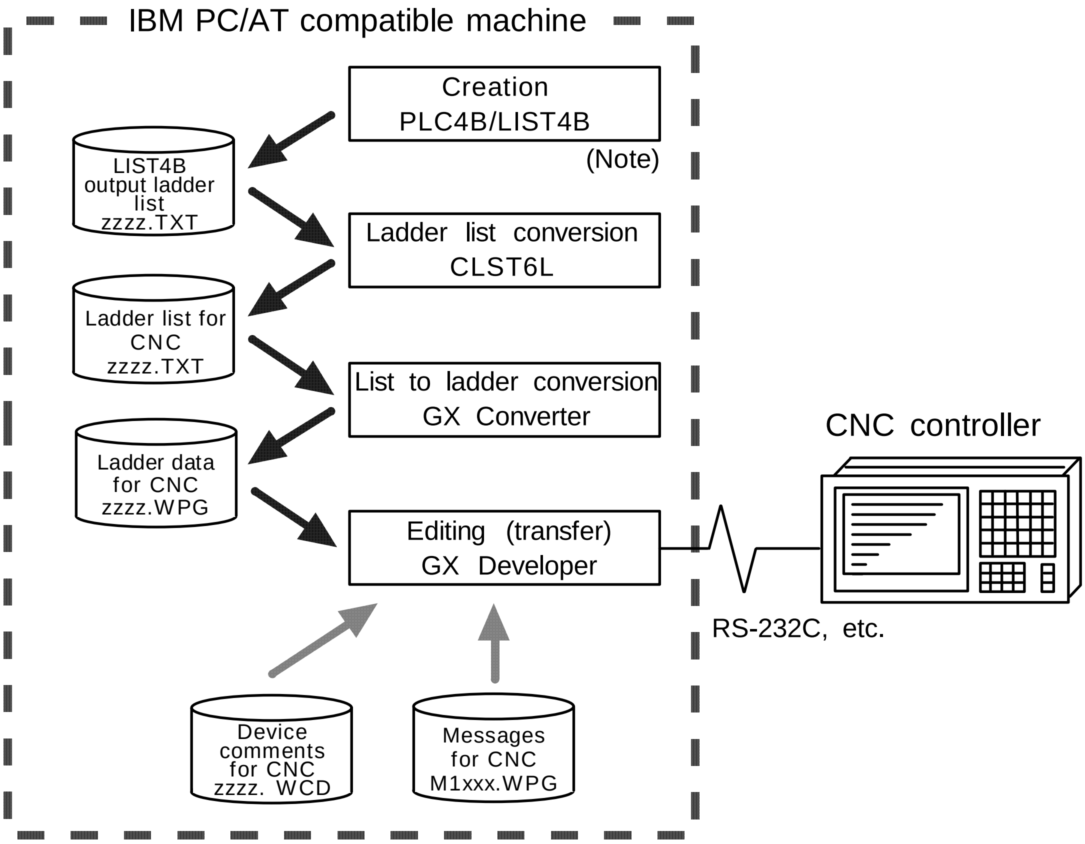
[Online] → [Monitor] → [Stop monitor] or Alt + F3

(Note) If the ladder program being run by the CNC differs from the one being displayed on GX Developer, monitoring will not result in an error but will continue.

1.3.9 Diverting the PLC program that was developed using PLC4B

(Note) PLC4B and LIST4B can not be used with C64. This section describes the method to use the ladder data on the C64 series system, however, the ladder data here must be created with PLC4B of the former-model CNC.

(1) Development procedure



(a) Creation
The PLC program created for the old model is output in a list format.

(b) Conversion
Using CLST6L (ladder list converter), the output program is converted into a PLC program (list format). Using GX Converter (data conversion software package), the list format program is converted into the GX Developer data.

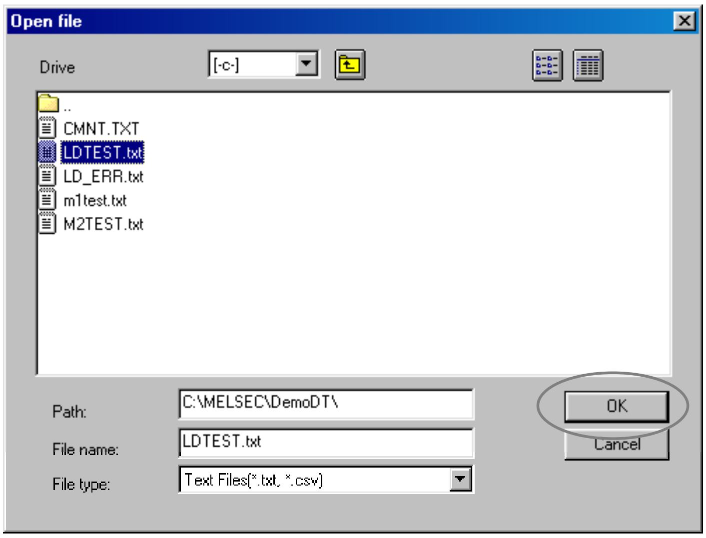
(c) Editing/transfer
The resultant program can be handled like a newly created PLC program.

(2) Starting GX Converter and specifying the file to be converted

Perform the following operation from GX Developer to start GX Converter (read).

[Project] → [Import file] → [Import to TEXT ,CSV format file]

On the following screen, specify the file to be converted (LDTEST.TXT) and click [OK].

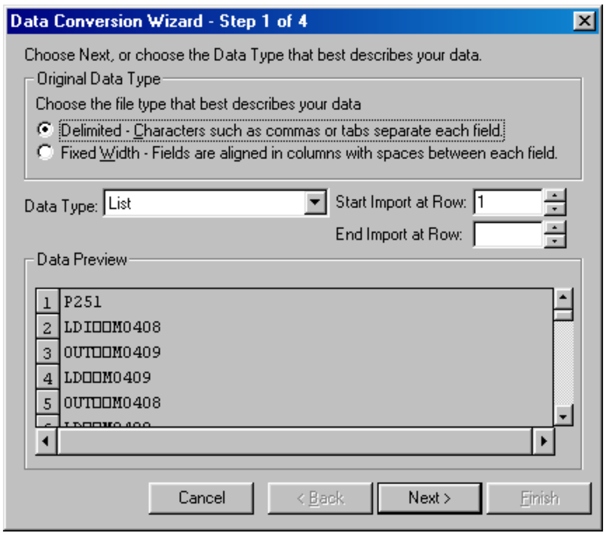


(3) Conversion format setting

Set the conversion format on the following data conversion wizard screen.

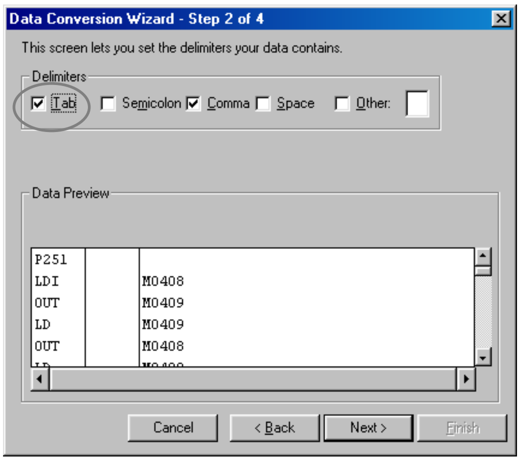
(a) Data conversion wizard 1/4

Choose [Original Data Type]-[Delimited] and [Data Type]-[List], and click [Next>].



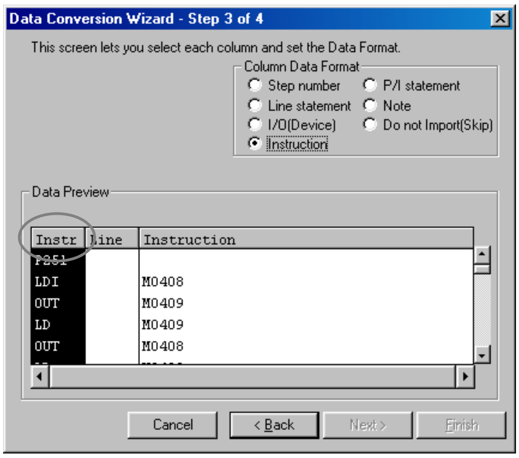
(b) Data conversion wizard 2/4

Choose [Delimiters]-[Tab] and click [Next>].



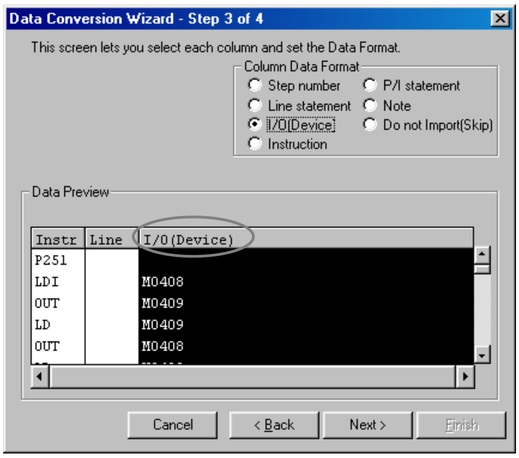
(c) Data conversion wizard 3/4

Choose to highlight the [Instr] column part in the [Data Preview] list and choose [Column Data Format]-[Instruction].



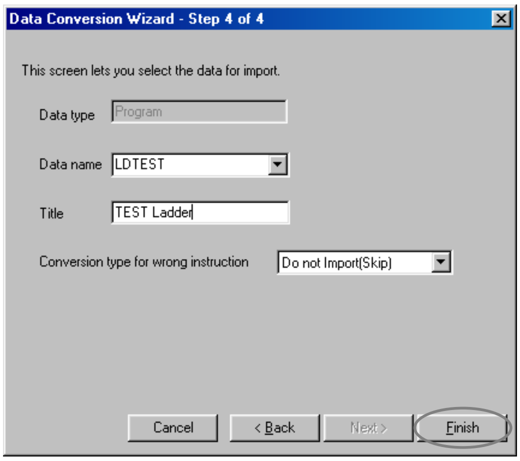
(d) Data conversion wizard 3/4

Further, choose to highlight the Argument column part in the [Data Preview] list and choose [Column Data Format]-[I/O(Device)]. Click [Next>].



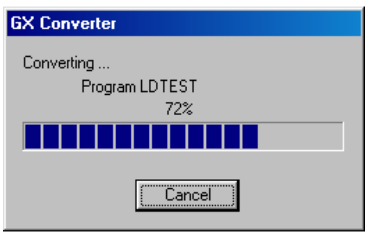
(e) Data conversion wizard 4/4

Set the program name used on GX Developer at [Data name] column and a ladder annotation at [Title] column, and click [Finish].



(f) Completion

The setting is complete when the following completed dialog appears after the converting dialog. Click [OK].



1.4 Creating PLC Message Data

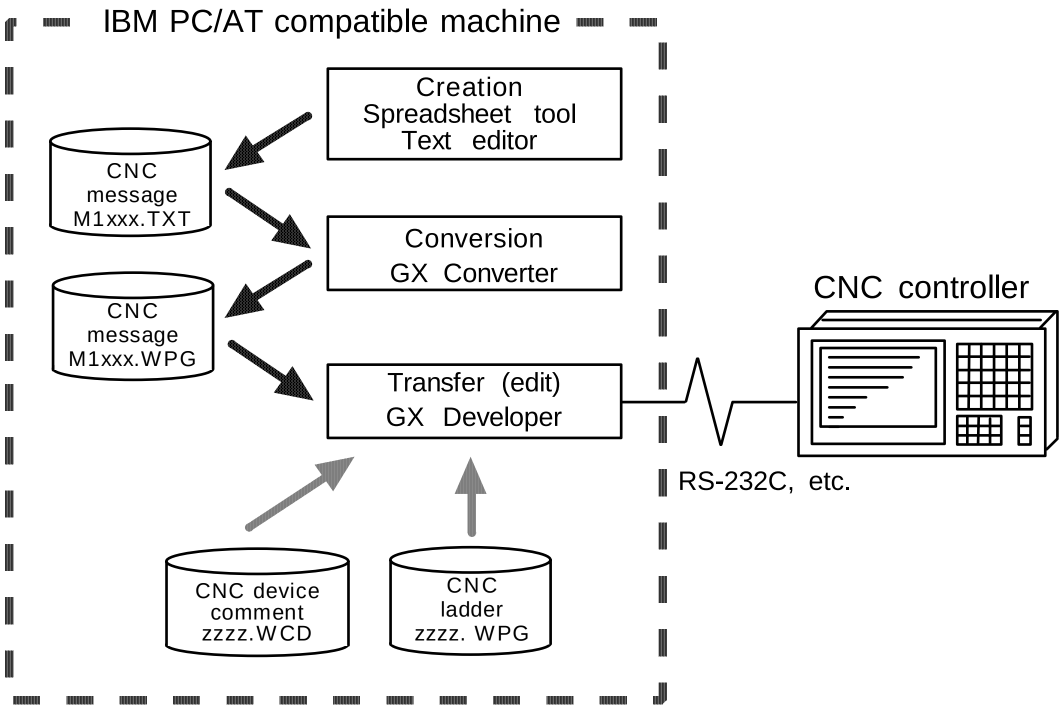
This chapter explains a procedure for developing PLC-related data such as alarm messages, operator messages, and PLC switches.

1.4.1 Development Procedure

There are the following two methods as a general development procedure of message data.

1) Making conversion into GX Developer data using a general text editor or spreadsheet tool and data conversion package.
(When there is a large volume of message data and you want to control them with a commercially available tool, for example)

2) Entering messages directly from GX Developer
(When there is a small volume of message data or when addition or correction is to be made, for example)



(1) Using a general text editor

(a) Creation

The message data is described using a general text editor. The description method and format will be described later.

(b) Conversion

The conversion from text data to GX Developer data is carried out using the "GX Converter (data conversion software package)".

(c) Transfer

With the GX Developer, the message data is handled as a PLC program interlinear comment, and can also be edited.
The message data is transferred to the CNC controller using the GX Developer, in the same manner as the ladder program.

(2) Entering messages directly from GX Developer

(a) Creation

The message data is described directly from GX Developer. The message data is handled as a PLC program interlinear comment by GX Developer. The description method and format will be described later.

(b) Transfer

The message data is transferred from GX Developer to the CNC in the same manner as the ladder program.

1.4.2 Message Data Description Method

The message data can be described as text data by a general text editor and also by commercially available spreadsheet software in addition to the direct input with GX Developer.

(1) Description Format

Message data is classified into setting areas to store the setting for each message and message areas to store message data. It is described in the following respective description format.

(a) Setting area

The message length and No. of messages are set for each message in the setting area. The message data region secured by the CNC can be adjusted to the most efficient status using these settings. The respective maximum values are set if nothing is set. (Refer to "(4) Precautions" for the maximum values.)

;$, *message classification code, maximum message length, No. of messages* [CR]

A:Alarm message	O:Operator message
P:PLC switch	M:Comment message

(b) Message area

The message area is described using the following description format.
The description format cannot be abbreviated. Comma(,) and [CR] must be described, even the message character string is blank.

Message classification	Description format
Alarm message	;A, *index No., data register No., message character string* [CR]
Operator message	;O, *index No., data register No., message character string* [CR]
PLC switch	;P, *switch No. message character string* [CR]
Comment message	;M, *device, device No., message character string* [CR]

Message classification code	: A one-byte alphabetic character expressing each message classification
Index No.	: One-byte number (0 to No. of messages in the setting area - 1)
Switch No.	: One-byte number (0 to No. of messages in the setting area - 1)
Data register No.	: One-byte number
Device	: One-byte number (1 or 2)
Device No.	: One-byte number (0 to 10)
Message character string	: One-byte alphanumeric character, shift JIS Code 1 character, No. of characters in the setting area message length. Semicolons, commas, spaces and tabs can also be used. Note that the tab at the head of the message character string is ignored.
Semicolon(;)	: Message data identification code
Comma(,)	: Separator between each description (a comma only is used to leave a message character string blank)
[CR]	: Line feed code, (CR/LF) or (LF).

(2) Description Method

The message data is described as text data by the following description format.

```
;# ladder ver1. '00.08.01                          ... Comment
;$, A,32, 200
;$, 0, 40, 200                                     ... Setting area
;$, P, 14, 32
;$, M, 60, 20
NOPLF
;A, 0, 0, Emergency stop
;A, 11, 1, Spindle alarm
            :
            :
NOPLF                                              ... Message area (alarm messages)
            :
            :
NOPLF
            :
            :
NOPLF                                              ... Page break code
;0, 1, 9000, MELDAS 600LADDER Ver1.0
;0, 20, 9000, BND-400W000-A0                       ... Message area (operator messages)
            :
            :
NOPLF
;P, 1, Program restart
;P, 2, Automatic power OFF                         ... Message area (PLC switches)
            :
            :
NOPLF
;M, 1, 0, [Spindle]
;M, 1, 0, [Standby 1]                              ... Message area (comments)
            :
            :
END                                                ... End code
```

(a) Comment

Statements having a semicolon (;) at the head of the line, in a different format than described in "(1) Description format", are regarded as comments. These comments are handled as comment data in the GX Developer also, but are erased during the transfer to the CNC controller. An error will occur if there is no semicolon at the head of the line.

(b) Setting area

Each message is set here. This area must be described before the message area of the relevant message. That setting will be ignored if it is described in the middle of or after the relevant message description.

(c) Message area

Collect similar messages in a group and describe them. There is no description order in the respective messages, but the latter description is validated if there are descriptions with the same factors (index No., etc.).

(d) Page break code

A page break code is described at one or more places approx. every 15 lines in the setting area and message area. The message data may skip if there is no page break code.

(e) End code

An end code is described at the end of the description. Description after the end code are ignored.
An error will occur if there is no end code.

(3) Details of comment message

The messages used for Tool registration screen and for load meter are defined as the comment messages
Comment messages is described using the following format.

;M, *device, device No., message character strings*[CR]

(a) Tool registration message

Maximum 8 characters for a step, and up to 5 steps of messages can be created. Even if more than 5 steps are created, the characters of first 5 steps are displayed.
[Description format]

;M, 1, 0, *message character strings*[CR]

(b) Load meter message

Maximum 40 characters for a step, and up to 7 steps of messages can be created.
- Message of 1st step is for the 1st part system.
- Message of 2nd step is for the 2nd part system.
- Message of 3rd step is for the 3rd part system.

  : :
- Message of 7th step is for the 7th part system.

[Description format]

;M, 2, 0, *message character strings*[CR]

[Example]

```
NOPLF
;M,1,0,[Spindle]
;M,1,0,[Index 1]
NOPLF
;M,2,0,Spindle 1
;M,2,0,0       50        100
;M,2,0,       %
;M,2,0,|_|_|_|_|_|_|_|_|_|_|
;M,2,0,Z- axis 1
;M,2,0,0       50        100
;M,2,0,       %
;M,2,0,|_|_|_|_|_|_|_|_|_|_|
NOPLF
;M,2,0, Spindle 2
;M,2,0,0       50        100
;M,2,0,       %
;M,2,0,|_|_|_|_|_|_|_|_|_|_|
;M,2,0, Z- axis 2
;M,2,0,0       50        100
;M,2,0,       %
;M,2,0,|_|_|_|_|_|_|_|_|_|_|
NOPLF
    :
```

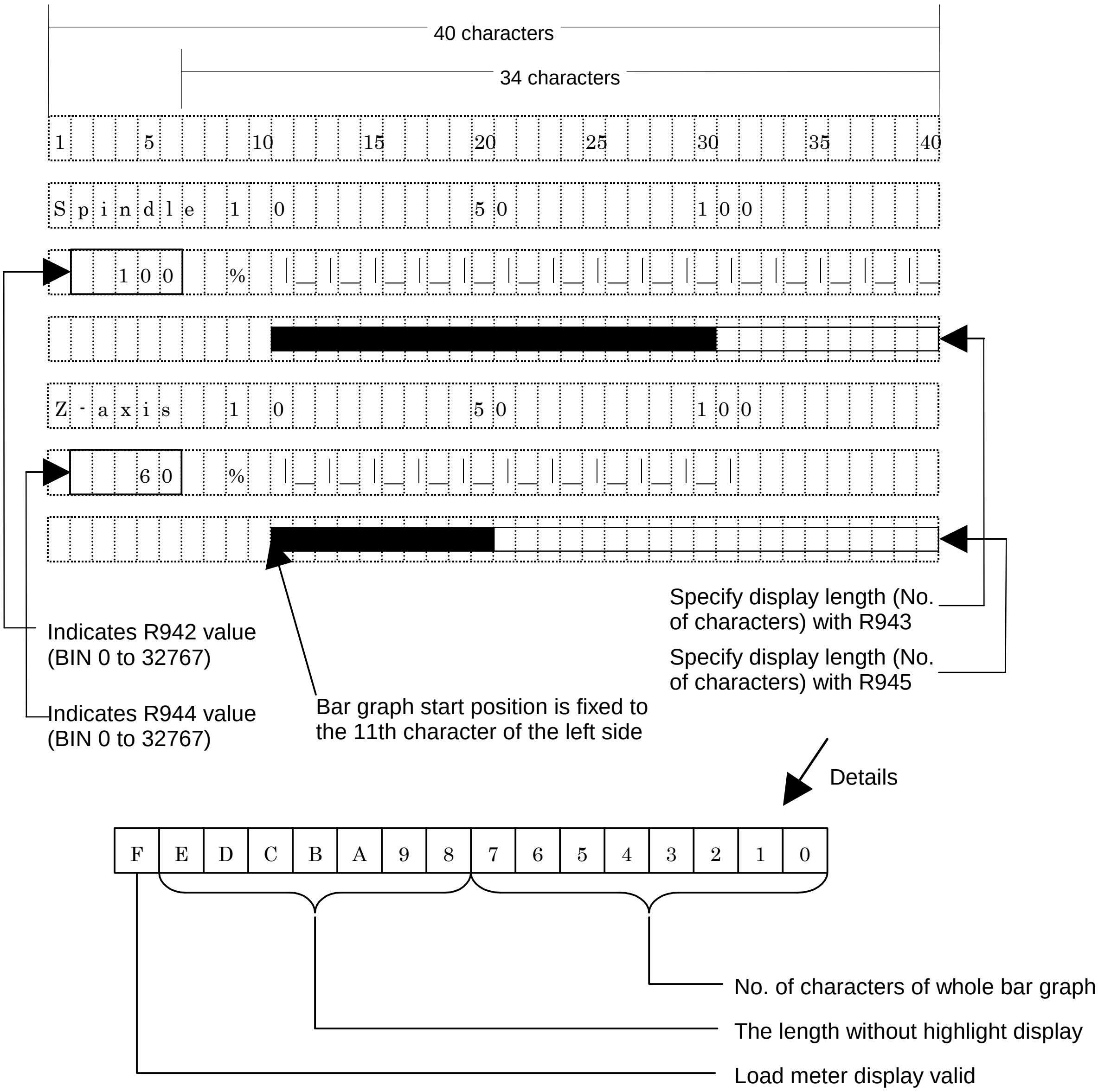
Tool registration message(Up to 5steps)

Load meter message (1st part system)
Line 1 Left 10 characters displayed
Line 1 Right 30 characters displayed
Line 2 Left 10 characters (Only 3 of left valid)
Line 2 Right 30 characters displayed
Line 4 Left 10 characters displayed
Line 4 Right 30 characters displayed
Line 5 Left 10 characters (Only 3 of left valid)
Line 5 Right 30 characters displayed

Load meter message (2nd part system)

: :

(c) Load meter display



List of file registers (R) used for load meter display

		For $1	For $2	For $3	For $4	For $5	For $6	For $7
Load meter 1	Numerical display	R942	R1042	R1142	R1242	R1342	R1442	R1542
	Bar graph display	R943	R1043	R1143	R1243	R1343	R1443	R1543
Load meter 2	Numerical display	R944	R1044	R1144	R1244	R1344	R1444	R1544
	Bar graph display	R945	R1045	R1145	R1245	R1345	R1445	R1545

(Note 1) Use $1 for models not having a part system.

(4) Precautions

No. of characters, quantity limitations, handling of information other than settings, handling of information other than format.

(a) Message data maximum value

Processing will be carried out with the following values considered as the maximum values if the setting is not carried out in the setting area, or if the description position in the setting area is illegal.

Message classification	Max. message length	Max. No. of messages
Alarm messages	32 byte	512
Operator messages	60 byte	512
PLC switches	14 byte	32
Comments	60 byte	100

[Note]
Two-byte data in the message character string is handled as two characters.
GX Developer accepts 64 characters as an interlinear comment. However, since that includes information other than a message character string (e.g. message classification code, index No. and data register No.), the message character string is actually up to 58 characters long.

(b) When the setting value and message data do not match

When the message data contents (such as index No, switch No. and message character string) overflows from the settings in the setting area, the data that overflowed is ignored.

1.4.3 Converting Data into GX Developer Format

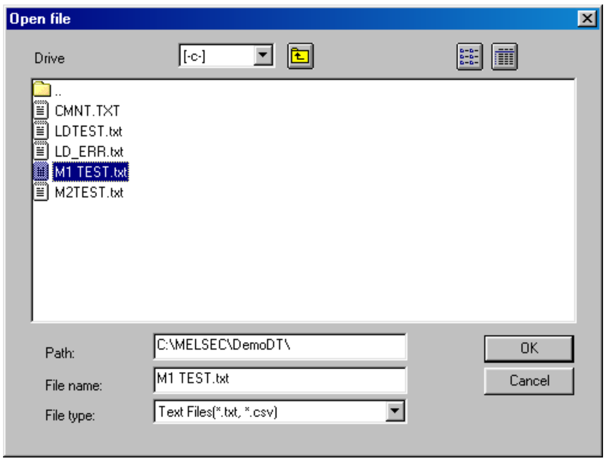
Convert the message data, which was described using a text editor or like, into GX Developer data in the following method. Use "GX Converter (data conversion software package)" for conversion. GX Converter can be started from the GX Developer menu.

(1) Starting GX Converter and specifying the file to be converted

Perform the following operation from GX Developer to start GX Converter (read).

[Project] → [Import file] → [Import from TEXT ,CSV format file]

On the following screen, specify the file to be converted (M1TEST.TXT) and click [OK].

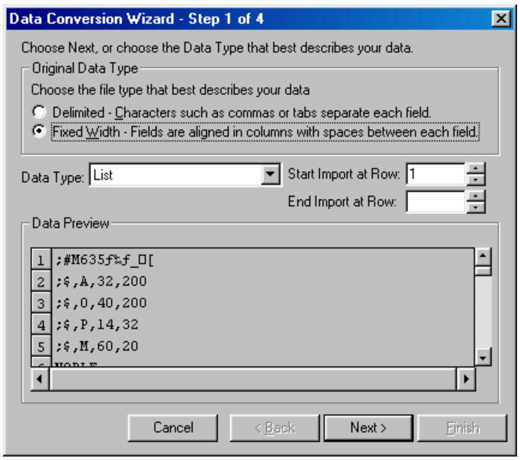


(2) Conversion format setting

Set the conversion format on the following data conversion wizard screen.

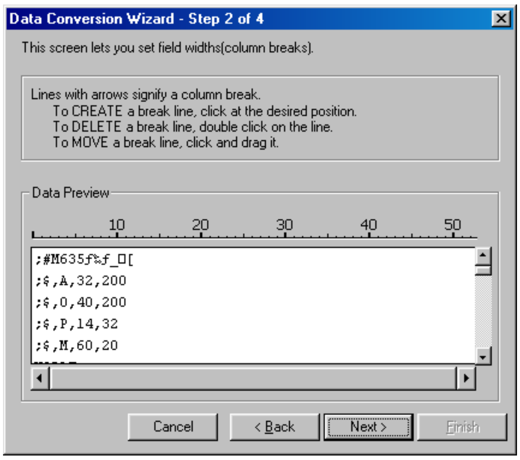
(a) Data conversion wizard 1/4

Choose [Original Data Type]-[Fixed Width] and [Data Type]-[List], and click [Next>].



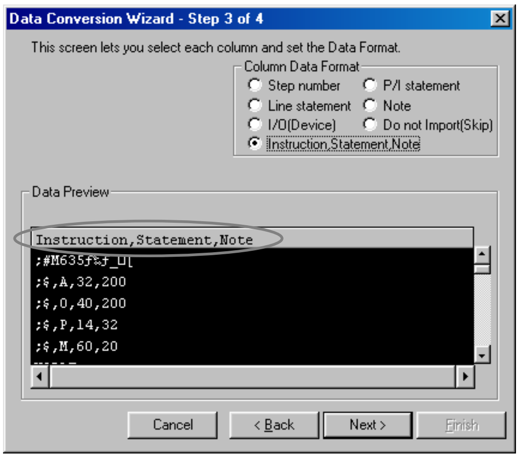
(b) Data conversion wizard 2/4

Just click [Next>].



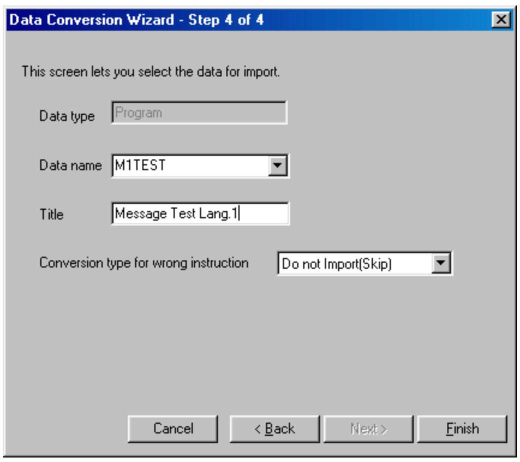
(c) Data conversion wizard 3/4

Choose to highlight the Command column part in the [Data Preview] list and choose [Column Data Format]-[Instruction ,Statement ,Note]. Click [Next>].



(d) Data conversion wizard 4/4

Set the program name used on GX Developer in [Data name] and a data annotation in [Title], and click [Finish]. The setting is complete when the completed dialog appears. Click [OK].



1.4.4 Entering/Editing Data Using GX Developer

The message data in GX Developer are handled as the "integrated type interlinear statements" of a PLC program. "Integrated type interlinear statements" are interlinear comments provided to assist the understanding of the PLC program, and those transferred to the controller together with the PLC program are called the "integrated type".
"Interlinear statements" can be displayed and edited using [Ladder] or [Instruction list].

(1) Interlinear statement display using circuit display

(a) Display of project data list

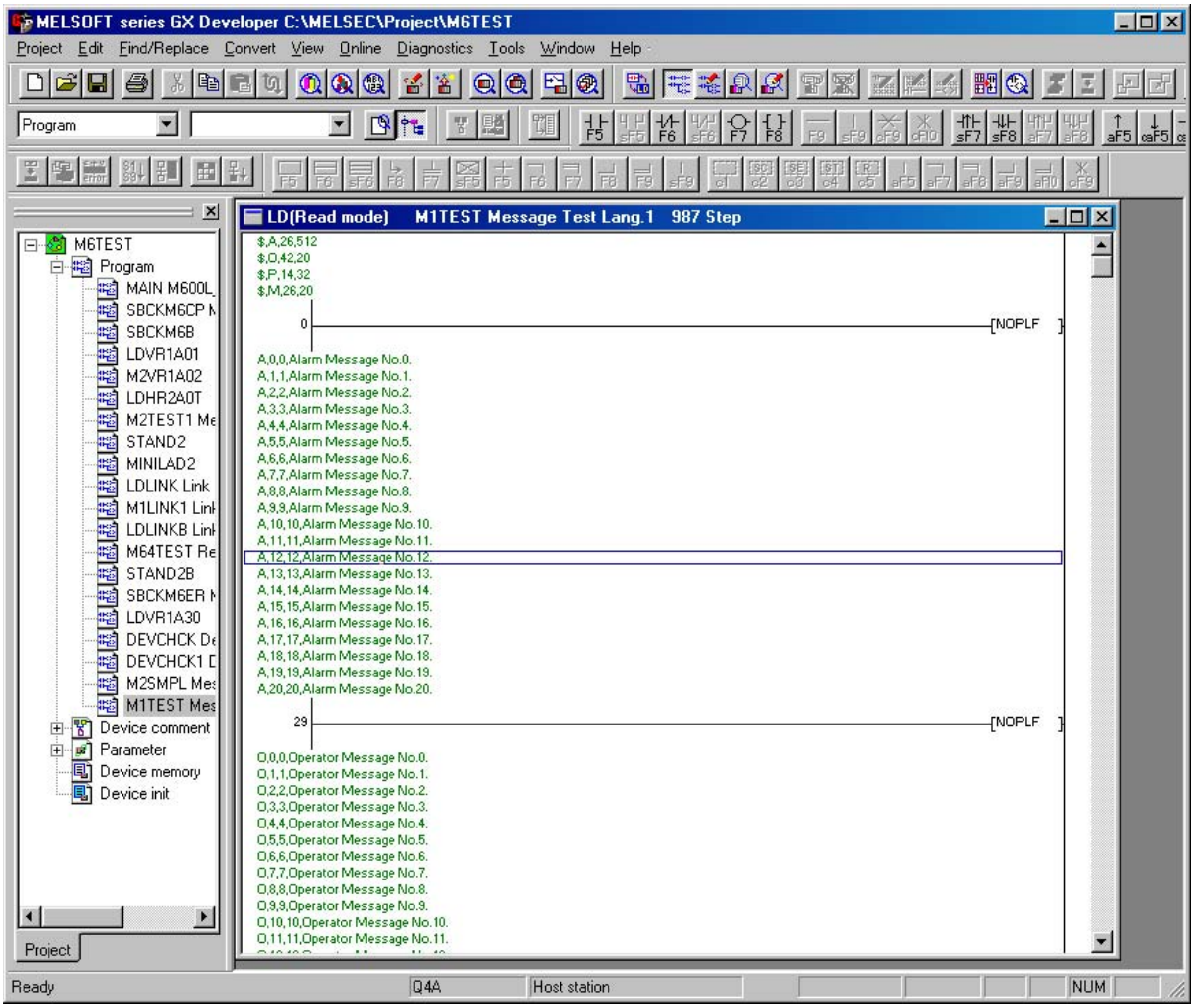
Perform the following operation to display the Project data list window and double-click the file name to display the edit screen. First, the normal ladder screen appears.

[View] → [Project data list], then double-click *File name you want to display*.

(b) Display of message data

Perform the following operation to display the message data that are integrated type interlinear statements.

[View] → [Statement]



(2) Interlinear Statement Display Using List Display

(a) Display of project data list

Perform the following operation to display the Project data list window and double-click the file name to display the edit screen. First, the normal ladder screen appears.

[View] → [Project data list], then double-click *File name you want to display*.

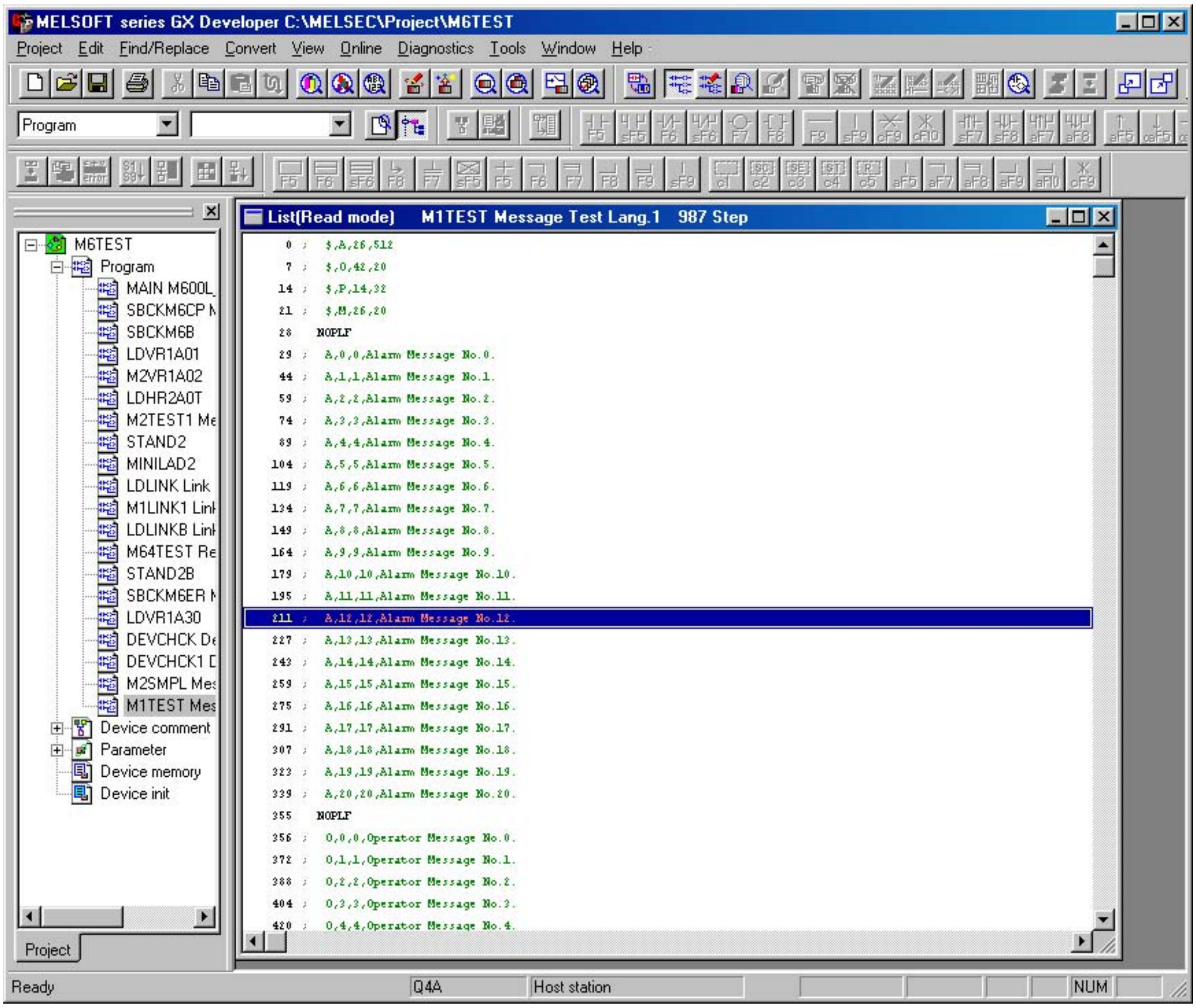
(b) Display of list data

Perform the following operation to display the list data. The list display also shows the message data that are integrated type interlinear statements.

[View] → [Instruction list]

Perform the following operation to return to the circuit display.

[View] → [Ladder]



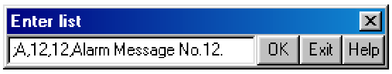
(3) Editing of integrated type interlinear statements

(a) Circuit display

On the circuit display screen that shows the integrated type interlinear statements, double-clicking the interlinear statement you want to edit displays the following dialog. Perform editing operation on the dialog and click [OK] or press [Enter].

(b) List display

On the list display screen, double-clicking the interlinear statement you want to edit displays the following dialog. Perform editing operation on the dialog and click [OK] or press [Enter].

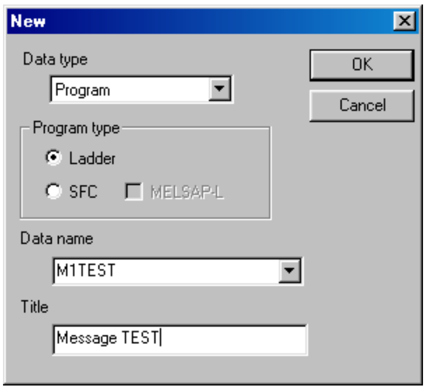


(c) Entering new message data

- Displaying new edit screen

Perform the following operation to display the [New] dialog, and set the [Data name] and [Title]. After setting, click [OK].

[Project] → [Edit Data] → [New]



- Changing to list display mode

Perform the following operation to display the list data.

[View] → [Instruction list]

- Entering message data

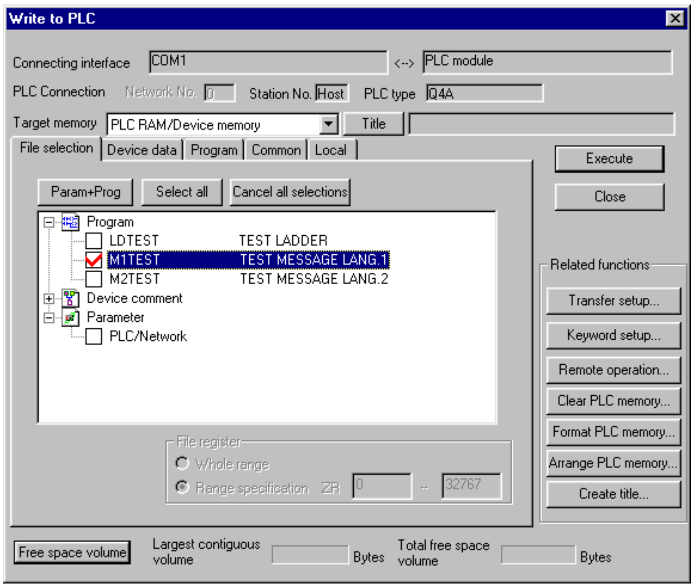
Press "Enter" on the "END" line, enter data as in “(b) List display”, and then press "Enter" on the next line and enter message data.

1.4.5 Writing to the CNC

The following shows the method of transferring a message from the GX Developer to the CNC. The transfer method is the same as the ladder code transfer method. Ladder codes and message data are distinguished by their file names only.
Perform the following operation to display the [Write to PLC] screen, and choose the file to be written.

[Online] → [Write to PLC]

The following example transfers a message first language file "M1TEST.GPG".



1.4.6 Reading and Verifying from the CNC

The following shows the method of reading and verifying a message from the CNC to the GX Developer. The method of reading and verifying is the same as that of ladder codes. Ladder codes and message data are distinguished by their file names only.

(1) Menu selection/screen operation

Refer to the following sections for operation methods.
For read : "1.3.6 Reading the PLC Program from the CNC"
For verification : "1.3.7 Verifying the PLC Programs"

(2) Message Read Format

The message description format was shown in "1.4.2 (1) Description format", but there are no special rules concerning provision of descriptions in the setting area or the order of message description in the message area. For that reason, the description format may differ between transfer and reading of the message data.
The following shows the format during reading as the "Standard description format".

Standard description format of message data

```
Alarm message setting           …(a)
Operator message setting
PLC switch setting
Comment message setting
Alarm messages                  …(b)
Operator messages               …(c)
PLC switches                    …(d)
Comment messages                …(e)
NOPLF
END
```

(a) Setting area

The settings are described in order of alarm, operator, PLC switch and comment.
The maximum value is described if the setting is abbreviated.

(b) Alarm messages

Each message data is described in order of the index Nos.

(c) Operator messages

The same as the alarm messages.

(d) PLC switches

Each message data is described in order of the switch Nos.

(e) Comment messages

These messages are described in the same order as described before transfer.

(f) Others

- Spaces and tabs are not included before and after the comma(,) separating the message data factors.
- The message character string is handled the same as normal data even when blank.
- The NOPLF code between messages is described to the position to which the message data following the NOPLF code during transfer moved.

1.5 Creating Device Comments

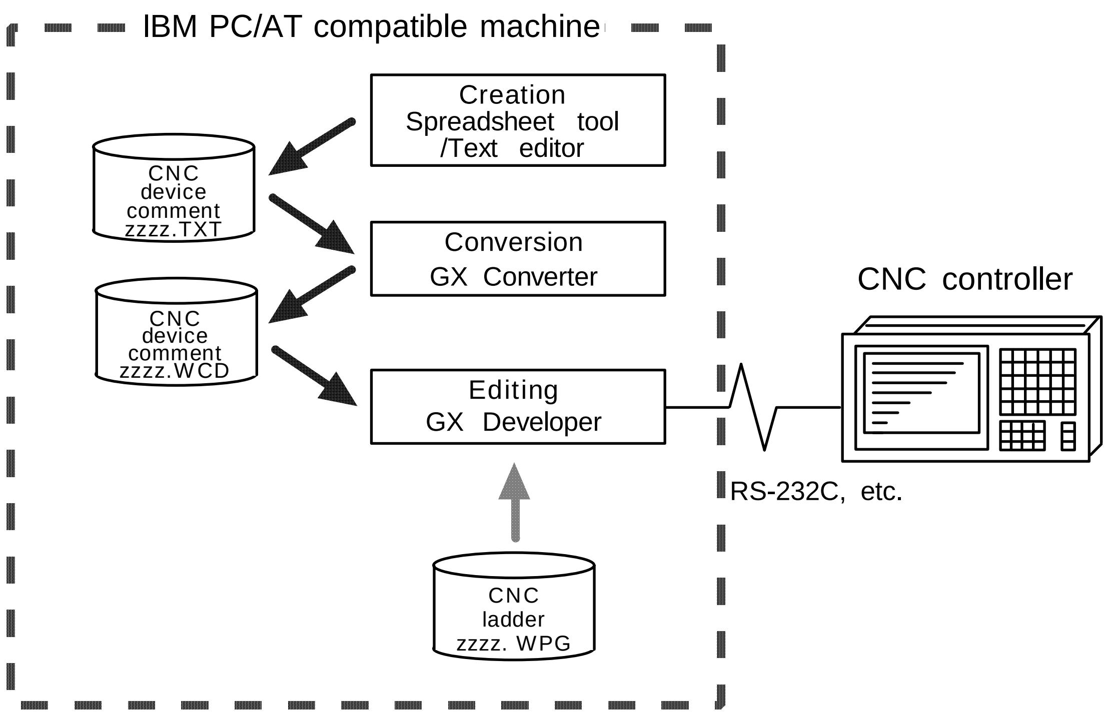
There are no MELDAS-specific operations for device comments. Therefore, refer to the GX Developer operating manual for the development method. This section describes the device comment development procedure outline and the development method using a general-purpose tool.

1.5.1 Development Procedure

There are the following two methods as a general development procedure of device comments.

(1) Indirect entry

In this method, device comments are converted into GX Developer data using a general text editor or spreadsheet tool and data conversion package. Use this method when you want to divert the device comments of the old model or when a device comment volume is large and you want to control them with a commercially available tool, for example.



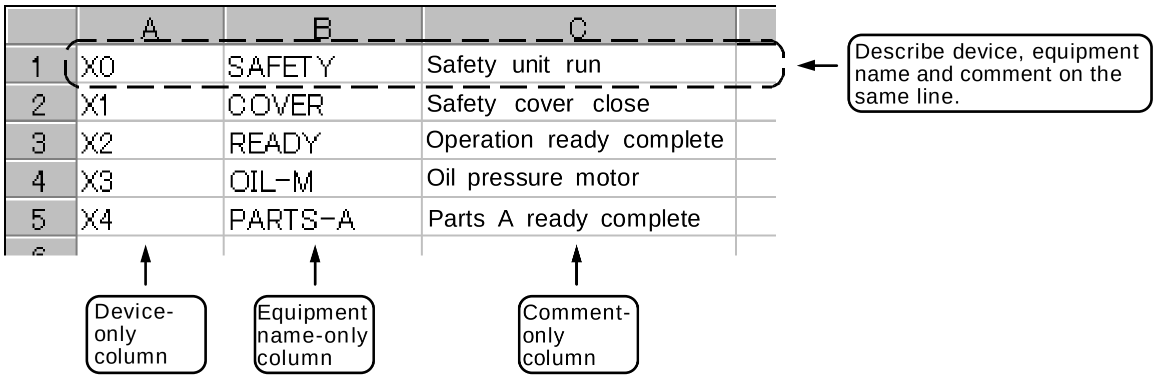
(2) Direct entry

In this method, device comments are entered directly from GX Developer. Use this method when a device comment volume is small or when addition or correction is to be made, for example.
There are the following three methods for direct entry from GX Developer. Refer to the operating manual for details.

- Creating comments on the device comment edit screen
- Creating device comments after circuit creation during ladder circuit creation
- Making addition/correction to device comments in the created ladder circuit

1.5.2 Description Method for Indirect Entry

The following explains the description method for creating device comments using a spreadsheet tool or like. The following example describes device comments using a spreadsheet tool.



Column data format	Explanation
Device	(1) Describe a device. • Conversion cannot be made if a device has not been described. Always describe a device. (2) A device is a required item. Describe it in one-byte code.
Equipment name	(1) Describe an equipment name. • It is not registered if the device part on the same row is blank or the device is illegal. (2) You can describe an equipment name of up to 8 characters.
Comment	(1) Describe a comment. • It is not registered if the device part on the same row is blank or the device is illegal. (2) You can describe a comment of up to 32 characters.

(Note) Describe data in any of the following combinations.
(1) Device, equipment name, comment
(2) Device, comment
(3) Device, equipment name

Save the above data in the CSV format. The following example shows the above data saved in the CSV format.

```
X0, SAFETY, Safety unit run
X1, COVER, Safety cover close
X2, READY, Operation ready complete
X3, OIL-M, Oil pressure motor
X4, PARTS-A, Parts A ready complete
```

1.5.3 Converting Comment Data into GX Developer Data

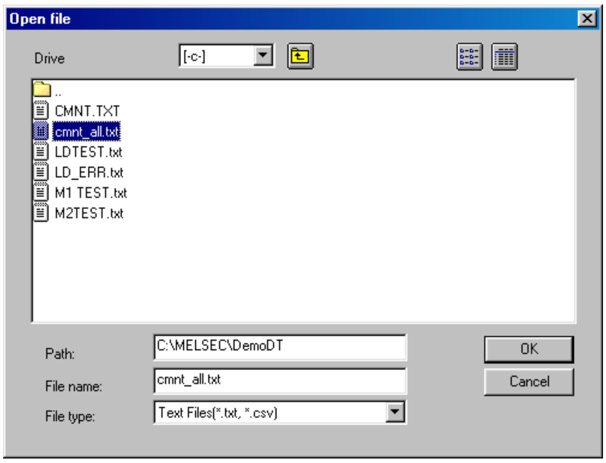
Convert the comment data (CSV format), which was created using a spreadsheet tool or like, into GX Developer data in the following method. Use "GX Converter (data conversion software package)" for conversion. GX Converter can be started from the GX Developer menu.

(1) Starting GX Converter and specifying the file to be converted

Perform the following operation from GX Developer to start GX Converter (read).

[Project] → [Import file] → [Import from TEXT ,CSV format file]

On the following screen, specify the file to be converted (cmnt_all.txt) and click [OK].

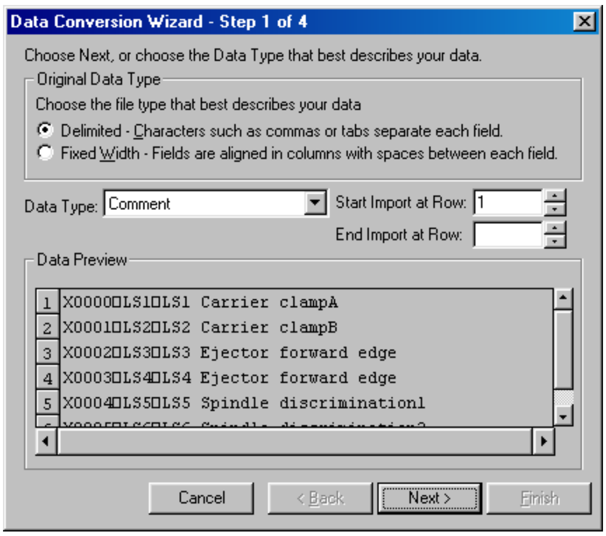


(2) Conversion format setting

Set the conversion format on the following data conversion wizard screen.

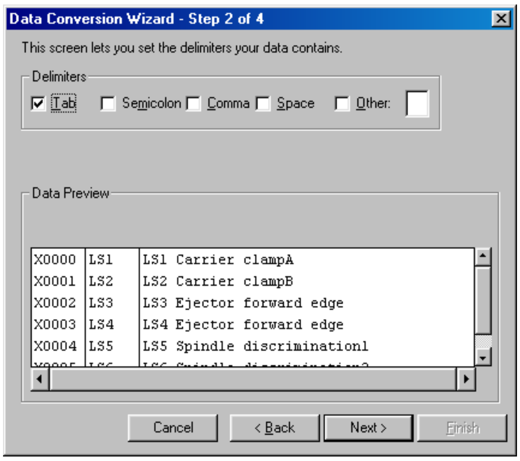
(a) Data conversion wizard 1/4

Choose [Original Data Type]-[Delimited] and [Data Type]-[Comment], and click [Next>].



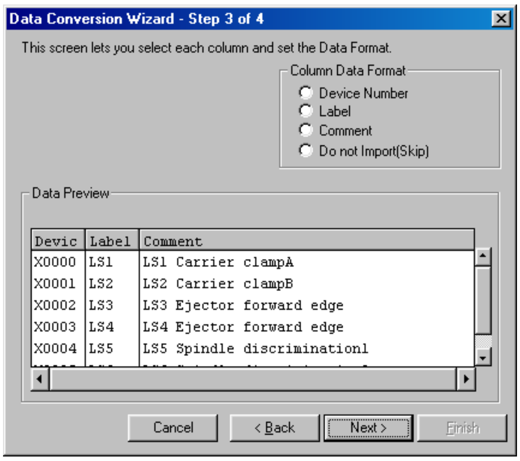
(b) Data conversion wizard 2/4

Choose [Delimiters]-[Tab] and click [Next>].



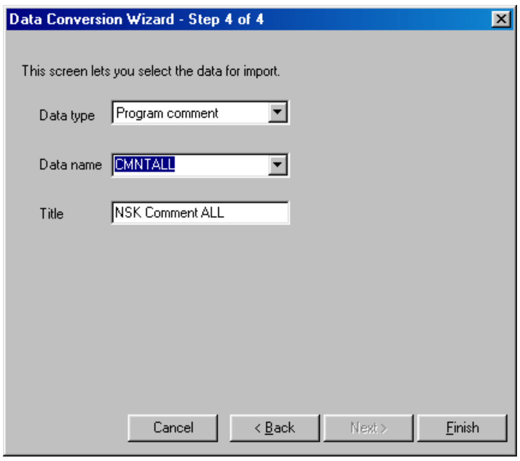
(c) Data conversion wizard 3/4

Make sure that the column parts in the [Data Preview] list are in order of [Device Number], [Label] and [Comment], and click [Next>].



(d) Data conversion wizard 4/4

Choose [Data type]-[Common comment] or [Program comment], set the comment file name used on GX Developer in [Data name] and a comment annotation in [Title], and click [Finish].

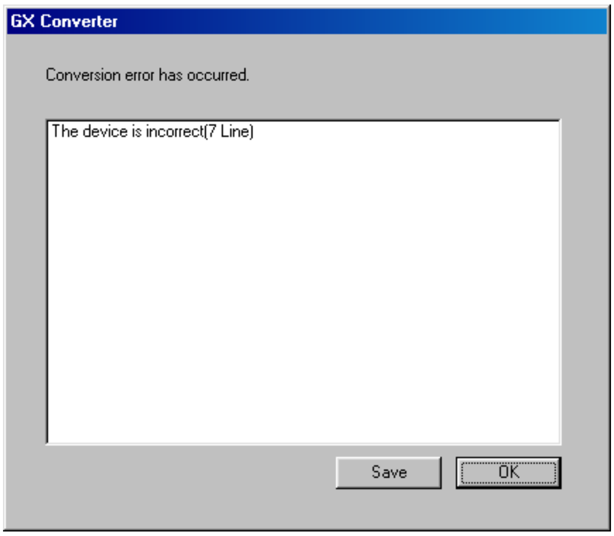


(e) Completion

The setting is complete when the following dialog appears. Click [OK].

(f) Error status

If an error occurred during conversion, its status and the line where it occurred are displayed.



1.5.4 Writing Comment Data to the CNC

The following shows the method of transferring a device comment data from the GX Developer to the CNC. The transfer method is the same as the ladder code transfer method. Ladder codes, message data and device comment data are distinguished by their file names only.
Perform the following operation to display the [Write to PLC] screen, and choose the file to be written.

[Online] → [Write to PLC]

1.6 PLC4B PLC Development Environment (M500) and Differences

This section explains differences between the PLC4B development environment and C64 series PLC development environment.

1.6.1 Development Tools, etc.

In the C64 series, a user PLC development environment that used the MELSEC PLC development tool was constructed. Consequently, the tools used at each development process differ. A comparison of each process is shown in the following table "List of development tool comparisons". Refer to the respective Instruction Manuals for details on each tool.

List of development tool comparisons

Development process		M500	C64
Application from the old model	Tool Hardware	Ladder and message conversion tool (CHG4PB) PC9801/PC-AT	Ladder list converter (CLST6L) ∗1 PC-AT
List -> ladder conversion	Tool Hardware	PLC development software (list section) (LIST4B) PC9801/PC-AT	GX Converter PC-AT
Ladder creation	Tool Hardware	PLC development software (ladder section) (PLC4B) PC9801/PC-AT	GX Developer PC-AT
Message creation	Tool Hardware	PLC development software (ladder section) (PLC4B) PC9801/PC-AT	Text editor -> GX Converter -> GX Developer PC-AT
Transfer to the CNC	Tool Hardware	PLC4B <-> FLD <-> M500 controller Via FLD	GX Developer <-> RS232C <-> CNC controller Via RS-232C
Monitor	Tool Hardware		GX Developer PC-AT <-> CNC controller
Print output	Tool (1) Hardware	PLC development software (ladder section) (PLC4B) PC9801/PC-AT	GX Developer PC-AT
	Tool (2) Hardware	PLC onboard (ONBD) M500 controller	

1.6.2 PLC Commands

Some commands have been changed because the user PLC development environment using the MELSEC PLC development tool has been supported in C64 series.
The command range that can be used in the MELSEC-QnA Series PLC program differs from the command range that can be used by the MELDAS Series. Because of this, some commands that can be used by the C64 series cannot be handled with the GX Developer. There are also commands that can be used by the GX Developer but cannot be used by the C64 series.
When these are arranged, they are classified into the three following types.

- C64 series commands that cannot be handled with the GX Developer
- C64 series commands that the format differs from that of the GX Developer
- Commands that can be used by the GX Developer, but cannot be used by the C64 series

(1) Commands that cannot be Handled with the GX Developer

Commands that cannot be handled with the GX Developer are substituted with alternate commands that can be handled with the GX Developer. Commands that can be alternated are shown in "Table of alternate command correspondence" as below.
When some commands described in "Table of alternate command correspondence" are created with the GX Developer with the C64 series command sign left as is, an error results and creation cannot be carried out. Create the commands using the GX Developer command sign described in the correspondence table.
When PLC programs containing alternate commands are written from the GX Developer to the C64 series, they are rewritten to the original MELDAS Series commands.

Table of alternate command correspondence

	Original MELDAS Series command		GX Developer command	
Classification	**Command sign**	**Symbol**	**Command sign**	**Symbol**
Bit	DEFR	─[DEFR D]─	ANDP	D ┤ ↑ ├
Average value	AVE	─[AVE S D n]┤	S.AVE	─[S.AVE S D n]┤
Carry flag set	STC	─[STC]┤	S.STC	─[S.STC]┤
Carry flag reset	CLC	─[CLC]┤	S.CLC	─[S.CLC]┤
ATC	ATC	─[ATC Kn Rn Rm]-< Mm >┤	S.ATC	─[S.ATC Kn Rn Rm Mm]┤
ROT	ROT	─[ROT Kn Rn Rm]-< Mm >┤	S.ROT	─[S.ROT Kn Rn Rm Mm]┤
TSRH	TSRH	─[TSRH Rm Rn]-< Mn >┤	S.TSRH	─[S.TSRH Rm Rn Mn]┤
DDBA	DDBA	─[DDBA Rn / Dn]┤	S.DDBA	─[S.DDBA Rn / Dn]┤
DDBS	DDBS	─[DDBS Rn]┤	S.DDBS	─[S.DDBS Rn]┤
BIT	LDBIT	├[BIT S1 n]─	LD<=	├[< = S1 n]─
	ANDBIT	─[BIT S1 n]─	AND<=	─[< = S1 n]─
	ORBIT	└[BIT S1 n]┘	OR<=	└[< = S1 n]┘
	LDBII	├[BII S1 n]─	LD<>	├[< > S1 n]─
	ANDBII	─[BII S1 n]─	AND<>	─[< > S1 n]─
	ORBII	└[BII S1 n]┘	OR<>	└[< > S1 n]┘
BCD decode	BDECO	─[BDECO S D n]─	S.BDECO	─[S.BDECO S D n]─

(2) Commands with the Format that Differs from that of the GX Developer

The device types and assignments have been reconsidered because the user PLC development environment using the MELSEC PLC development tool has been supported in C64 series.
The format of conventional "commands used by inputting or outputting an accumulator (A0, A1)" is changed so that the general word register can be designated with arguments without fixing by A0 or A1 input or output because of the discontinuance of accumulator (A0, A1).
Format-changed commands are shown in "Table of format-changed command correspondence" as below.
When some commands described in "Table of format-changed command correspondence" are created with the GX Developer with the conventional command sign left as is, an error results and creation cannot be carried out. Create the commands using the GX Developer command sign described in the correspondence table.

Table of format-changed command correspondence

	Original MELDAS Series command		Format-changed command		Remarks
Classification	Command sign	Symbol	Command sign	Symbol	
Right rotation	ROR	—[ROR n]┤	ROR	—[ROR D n]┤	∗1
	RCR	—[RCR n]┤	RCR	—[RCR D n]┤	
	DROR	—[DROR n]┤	DROR	—[DROR D n]┤	
	DRCR	—[DRCR n]┤	DRCR	—[DRCR D n]┤	
Left rotation	ROL	—[ROL n]┤	ROL	—[ROL D n]┤	
	RCL	—[RCL n]┤	RCL	—[RCL D n]┤	
	DROL	—[DROL n]┤	DROL	—[DROL D n]┤	
	DRCL	—[DRCL n]┤	DRCL	—[DRCL D n]┤	
Search	SER	—[SER S1 S2 n]┤	SER	—[SER S1 S2 D n]┤	∗2
Quantity of 1	SUM	—[SUM S]┤	SUM	—[SUM S D]┤	∗3

∗1 : D is the head No. of the rotation device (word 16-bit device)
∗2 : D is the head No. of the device that stores the search results (word bit device)
∗3 : D is the head No. of the device that stores the total No. of bits (word bit device)

(3) Commands that can be Used with the GX Developer, but cannot be Used by the C64 series

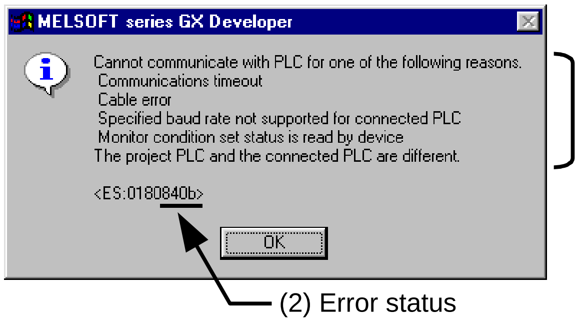
When commands that cannot be used by the C64 series are written from the GX Developer to the C64 series, they are rewritten to "NOP" commands, and if they are run, an alarm will occur. (Note that commands described in "Table of alternate command correspondence" are rewritten to the corresponding C64 series commands.)
"Commands that can be used by the GX Developer, but cannot be used in the C64 series " are defined by the following expression.

"**Commands that can be used by the GX Developer, but cannot be used in the C64 series"**
= "All commands described in the QnA Programming Instruction Manual"
- ("All commands described in the C64 series PLC Programming Instruction Manual"
+ "Table of alternate command correspondence"
+ " Table of format-changed command correspondence")

Refer to Appendix 2. MELSEC QnA Series command lists.

1.7 Error Status

If an error has occurred in GX Developer, the following dialog appears. The error message and error status are displayed in the dialog.



(1) Error message

(2) Error status

(Note) When an error occurred at GX Developer On-line function, the error message may not explain exactly the state in the CNC side.
Always refer to the error list.

The following table indicates the causes and remedies of the errors that can occur during online operation with the CNC. For other errors, refer to the GX Developer operating manual.

Status	Message	Cause	Remedy
2056	The executed function is not supported.	An attempt was made to execute a function not supported.	Check the specifications.
		The version of GX Developer used is not compatible to the CNC.	This error occurs when the GX Developer version is one of 7.10L to 7.14Q. Download from MELFANSweb to update the version to 7.17T or higher.
4002	Cannot communicate with the PLC.	An operation outside the specification was performed.	Check the storage capacity or the operation procedure.
4010	The PLC is in RUN mode, so writing cannot be done.	The PLC of the NC is running.	After stopping the PLC of the NC, start execution again.
4021	The applicable drive is not ready.	The specified target memory does not exist or is not in a usable status.	Change the target memory.
4029	Insufficient file capacity.	An attempt was made to write a file that exceeds the storage capacity.	Examine the file structure so that the data falls within the limited capacity.
402b	The file cannot be accessed.	An attempt was made to write the same type of file.	After deleting the same type of file from the NC side, start execution again.
4031	The specified device No. exceeds the permissible range.	The access request given is outside the accessible device range.	Check the number range of each device.
4052	The file is write protected.	The specified target memory is a write-disabled device (F-ROM).	Specify "internal RAM" as the target memory.
4053	Writing to the flash ROM failed.	An error occurred in the process to erase or to write data into the flash ROM.	The hardware may have inferiority or deterioration aspects. Contact our system department.

Status	Message	Cause	Remedy
4070	The program before correction differs from the registered program.	A ladder command outside the specification is included.	Perform verification to identify the command that is the cause of the problem.
4080	Incorrect data.	When the "Read from PLC" is executed: The data outside the specification is included in the specified file.	A ladder or message in the CNC side may be damaged. Contact our system department.
		When the "Write to PLC" is executed: Two or more END commands are included in the specified ladder file.	Edit the ladder program in the list mode to delete END commands except only one at the last line.
8201	Cannot communicate with the PLC.	The communication cable is faulty. • Not connected • DTR signal off	Check the serial port setting and cable connection.
840b	Cannot communicate with PLC for one of the following reasons.	There is no response from the NC. • The CNC has not started properly. • The connection channel of the CNC side serial port is different. • The serial cable outside the specifications is used for signal connection.	Check the following. • CNC side status • Cable connection • Bit selection : GPP communication valid : GPPW mode

(Note) For "PLC" in the message, read "CNC".

1.8 Initializing for PLC Data Storage Area

When an error has occurred during writing to the CNC, or when the normal sate is not recovered in spite of error handling, perform initialization for the PLC data storage area, and retry from the first.

⚠ CAUTION

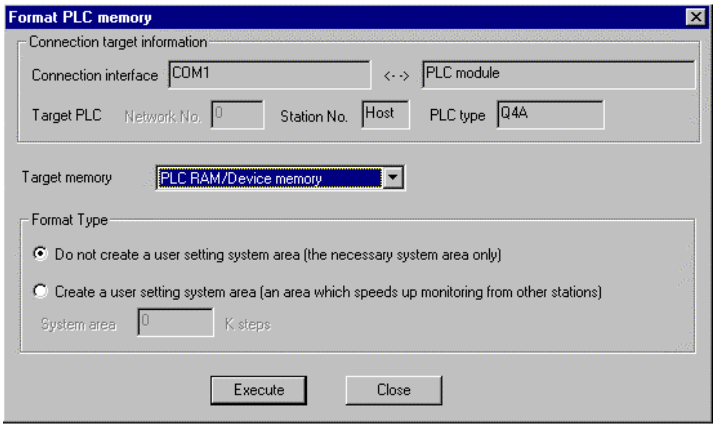
⚠ When initializing PLC data storage area is performed, all sequence programs and messages currently stored in the CNC will be erased. Do not use this operation other than when the error cannot be solved.

1.8.1 Operation procedure

Perform the following operation from GX Developer to start the operation screen.

[Online] → [Format PLC memory]

On the following screen, click [Execute].



(Note) As [Target memory], only "PLC RAM/Device memory" is valid.
The setting is not necessary for [Format Type].

The setting is completed when the following dialog appears. Click [OK]. All data stored in the F-ROM have been deleted and initialized.

2. PLC Processing Program

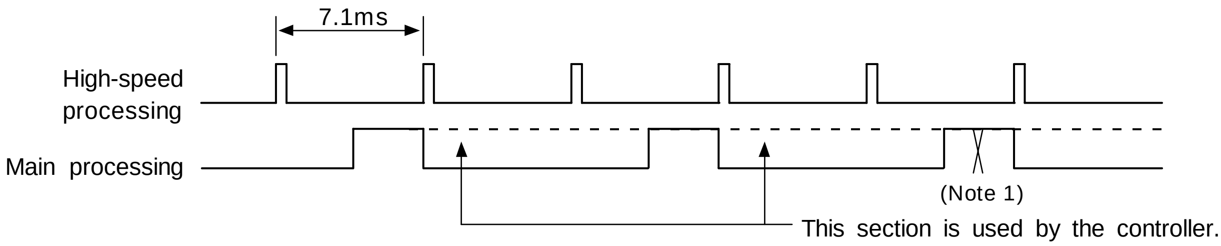
2.1 PLC Processing Program Level and Operation

2.1.1 High-speed processing program and main processing program

Table 2.1-1 explains the contents of users PLC processing level and Fig. 2.1-1 shows the timing chart.

Table 2.1-1 PLC processing level

Program name	Description (frequency, level, etc.)
High-speed processing program	This program starts periodically with a time interval of 7.1ms. This program has the highest level as a program that starts periodically. It is used in signal processing where high-speed processing is required. The steps for high-speed processing program should be up to 150 steps. Application example: Position count control of turret and ATC magazine
Main processing program (ladder)	This program runs constantly. When one ladder has been executed from the head to END, the cycle starts again at the head.



(Note 1) The section from the END command to the next scan is done immediately as shown with the X section. Note that the min. scan time will be 14.2ms.

Fig. 2.1-1 PLC processing program operation timing chart

2.1.2 Cautions on high-speed processing programming

The cautions on programming a high-speed processing program are explained. Pay careful attention to the following items before programming a high-speed processing program.

(1) Index resistor

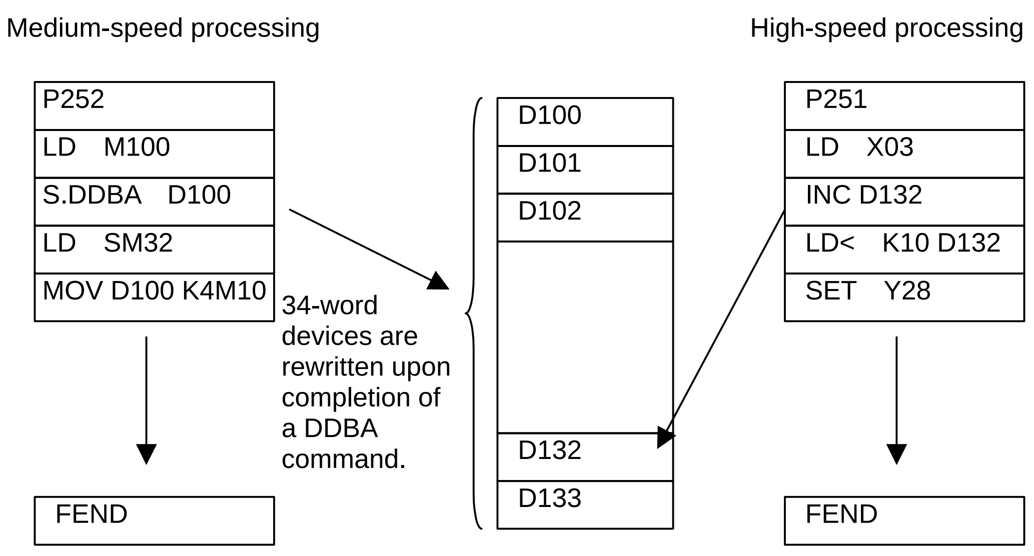
There are some function commands which use the index Z0 or Z1. When a value is changed by using Z0 or Z1 in high-speed processing, the function command on the main processing side may operate illegally. Do not use Z0 or Z1 for the high-speed processing.
Z2 to Z13 may be used on the high-speed processing, but cannot be used on the medium-speed side simultaneously.
If contents of the index resistor are rewritten by executing a high-speed processing while using on the medium-speed side, an illegal operation is caused.

(2) Each command of LDP, LDF, ORP, ORF, ANDP, ANDF

LDP, LDF, ORP, ORF, ANDP and ANDF are the commands that are turned ON by leading edge or trailing edge of device. Do not use these commands in high-speed processing since incorrect operation is caused.
When processing leading edge or trailing edge, use PLS/PLF commands or MEP/MEF commands.

(3) DDBA command

34-word devices from the head device used for a DDBA command are rewritten upon completion of a DDBA command. If using the medium-speed processing and high-speed processing overlapped in use this area, unexpected operation may be carried out. Please separate the area not to overlap.
The case where a device overlaps in the DDBA command executed by medium-speed processing and high-speed processing is shown below.



In medium-speed processing, the devices from D100 are used. Data is transmitted to D100-D133 (control data and 14 axes) at the completion of a DDBA command. On the other hand, in high-speed processing, D132 is used.
While processing a DDBA command with a medium-speed processing, high-speed processing may be started, which causes the data to be overwritten on D106-D133 at the completion of the DDBA command, even if D132 has been incremented. Thus, the contents of D132 will change.
Since such a phenomenon is not necessarily generated and it generates very rarely, studying a cause of the problem may be difficult. In order not to cause such a problem, do not use D100-D133 in high-speed processing.

2.2 Multi-Programming Function

Multiple PLC programs can be registered in C64 and executed in order.
Using this function, PLC program can also be developed by each process.

2.2.1 Program Registration Numbers

Max. registration numbers of PLC program are 9.
One program can contain programs for main and for high-speed together.

2.2.2 Program Execution Order

Multiple programs are executed in determined order, and not executed simultaneously.
The order is determined depending on the label No. in the head of program as following table, regardless of the registered order. If the same label No. has been used repeatedly, the program registered later will be valid, and the program made invalid will be not executed.

Label No.	Program	Execution order
P251	High-speed processing	1st
P360 to P368	High-speed processing	2nd to 10th (The smaller the number is, the higher the order is.)
P252	Main processing	1st
P370 to P378	Main processing	2nd to 10th (The smaller the number is, the higher the order is.)

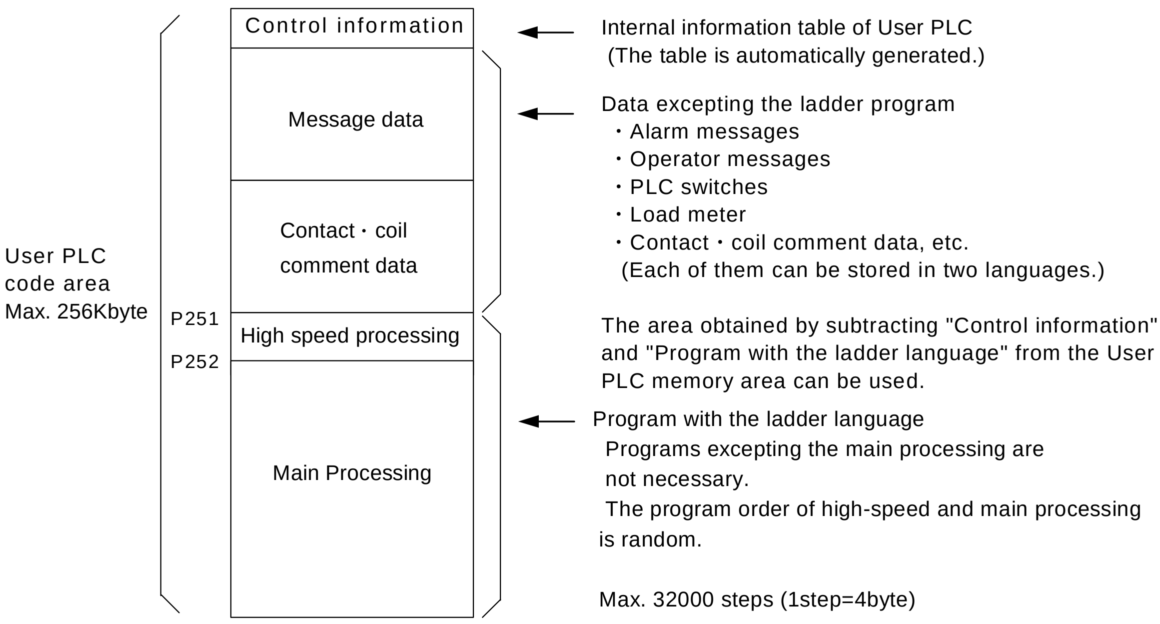
Each program ends by FEND command.

2.2.3 Precautions

(1) The label No. is common in all registered programs. Therefore, if the label No. is used repeatedly, the program registered later will be given priority.
(2) All devices used in the program are common. There is no local device. When one program end, the next program will take over each device state as it is.

2.3 User Memory Area Configuration

The user memory area approximate configuration and size are shown below.

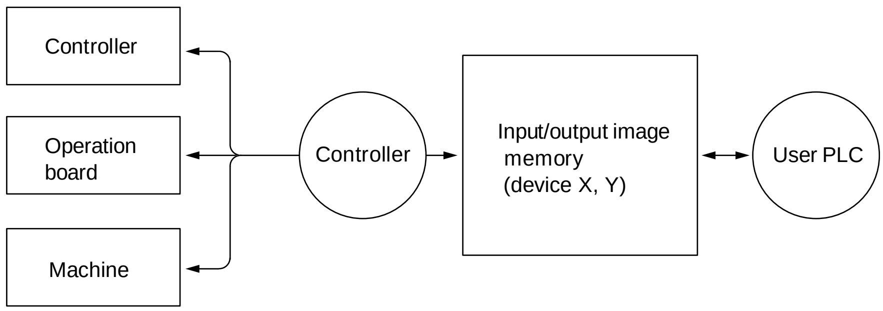


3. Input/Output Signals

3.1 Input/Output Signal Types and Processing

The input/output signals handled in user PLC are as follows:
(1) Input/output from/to controller
(2) Input/output from/to operation board (Note 1)
(3) Input/output from/to machine

The user PLC does not directly input or output these signals from or to hardware or controller; it inputs or outputs the signals from or to input/output image memory. For the reading and writing with the hardware or controller, the controller will perform the input/output according to the level of the main process or high-speed process.



(Note 1) The operation board here refers to when the remote I/O is installed on the communication terminal.

Fig. 3.1-1 Concept of input/output processing

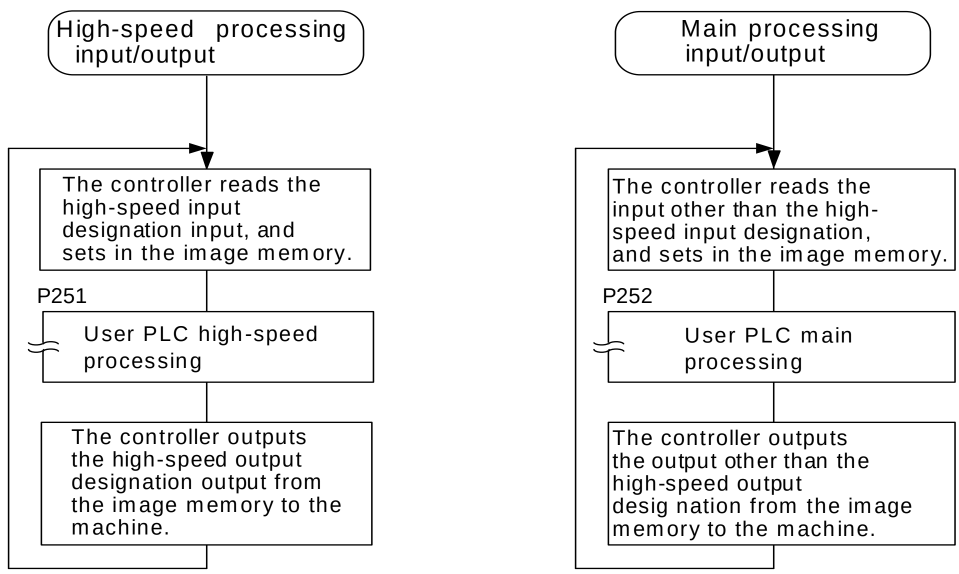


Fig. 3.1-2 Input/output processing conforming to program level

Table 3.1-1 lists whether or not high-speed input/output, interrupt input and initial processing can be performed.

Table 3.1-1 Whether or not high-speed input/output, interrupt input and initial can be performed

	High-speed input specification	**High-speed output specification**
Input signal from control unit	x	x
Output signal to control unit	x	x
Input signal from machine	◯ (2-byte units)	x
Output signal to machine	x	◯ (2-byte units)
Input signal from operation board	x	x
Output signal to operation board	x	x

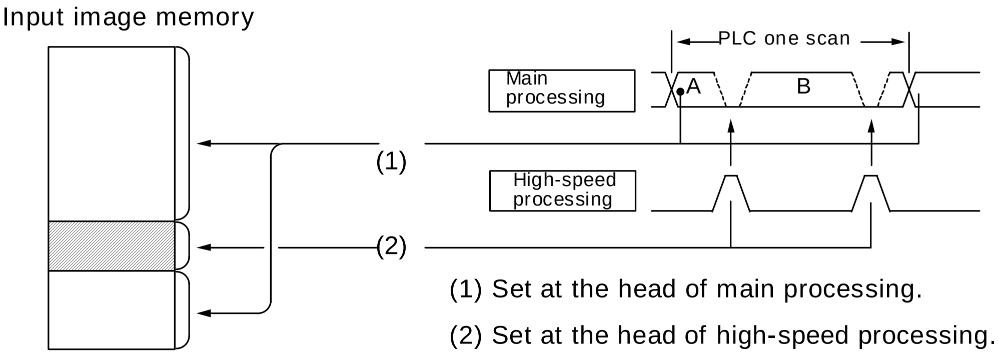
◯ : Possible x : Not possible

(Note 1) The operation board here refers to when the remote I/O is installed on the communication terminal.

3.2 Handling of Input Signals Designated for High-Speed Input

The input/output signals used in user PLC are input/output for each program level as shown in Fig. 3.1-2.
In high-speed processing, input/output signal for which high-speed input or output designation (parameter) is made is input or output each time the high-speed processing program runs. In main processing, signals other than the high-speed input/output designation are input/output.

When high-speed input designation signal is used in main processing, the input signal may change within one scan because high-speed processing whose level is higher than main processing interrupts. Input signal which must not change within one scan should be saved in temporary memory (M), etc., at the head of main processing and the temporary memory should be used in the main program, for example.



The hatched area is high-speed input designation part. Whenever the high-speed processing program runs, data is reset in the hatched area. Thus, the signal in the hatched area may change in main processing (A) and (B) because the high-speed process interrupts between (A) and (B) and re-reads the input signal in the hatched area.

3.3 High-Speed Input/output Designation Method

High-speed input/output is designated by setting the corresponding bit of the bit selection parameter as shown below.

(1) High-speed input designation

	7	6	5	4	3	2	1	0	
Bit selection parameter #6457	X70 ~ X7F	X60 ~ X6F	X50 ~ X5F	X40 ~ X4F	X30 ~ X3F	X20 ~ X2F	X10 ~ X1F	X00 ~ X0F	These bits correspond to the low-order byte (bits 0 to 7) of file register R4628
#6458	XF0 ~ XFF	XE0 ~ XEF	XD0 ~ XDF	XC0 ~ XCF	XB0 ~ XBF	XA0 ~ XAF	X90 ~ X9F	X80 ~ X8F	These bits correspond to the high-order byte (bits 8 to F) of file register R4628

(2) High-speed output designation

	7	6	5	4	3	2	1	0	
Bit selection parameter #6461	Y70 ~ Y7F	Y60 ~ Y6F	Y50 ~ Y5F	Y40 ~ Y4F	Y30 ~ Y3F	Y20 ~ Y2F	Y10 ~ Y1F	Y00 ~ Y0F	These bits correspond to the low-order byte (bits 0 to 7) of file register R4630
#6462	YF0 ~ YFF	YE0 ~ YEF	YD0 ~ YDF	YC0 ~ YCF	YB0 ~ YBF	YA0 ~ YAF	Y90 ~ Y9F	Y80 ~ Y8F	These bits correspond to the high-order byte (bits 8 to F) of file register R4630

- As listed above, one bit corresponds to two bytes (16 points).
- Input or output in which 1 is set in the table is not performed at the main processing program level.
- Although the number of bits set to 1 is not limited, set only necessary ones from viewpoint of overhead.
- High-speed input/output designation corresponds to the bit selection parameter and can be set in the parameter. However, it is recommended to set in a sequence program to prevent a parameter setting error, etc.

Example: —[MOV H3 R4628]— ..... To designate X00~X0F, X10~X1F

Bits 0 and 1

4. Parameters

4.1 PLC Constants

The parameters that can be used in user PLC include PLC constants set in the data type.
Set up data is stored in a file register and is backed up. In contrast, if data is stored in the file register corresponding to PLC constant by using sequence program MOV instruction, etc., it is backed up. However, display remains unchanged. Display another screen once and then select the screen again.
48 PLC constants are set (the setting range is ±8 digits). (Signed 4-byte binary data)
The correspondence between the PLC constants and file registers is listed below. The setting and display screens are also shown.

#	Corresponding file registers	
	High order	Low order
6301	R4501	R4500
6302	R4503	R4502
6303	R4505	R4504
6304	R4507	R4506
6305	R4509	R4508
6306	R4511	R4510
6307	R4513	R4512
6308	R4515	R4514
6309	R4517	R4516
6310	R4519	R4518
6311	R4521	R4520
6312	R4523	R4522
6313	R4525	R4524
6314	R4527	R4526
6315	R4529	R4528
6316	R4531	R4530
6317	R4533	R4532
6318	R4535	R4534
6319	R4537	R4536
6320	R4539	R4538

#	Corresponding file registers	
	High order	Low order
6321	R4541	R4540
6322	R4543	R4542
6323	R4545	R4544
6324	R4547	R4546
6325	R4549	R4548
6326	R4551	R4550
6327	R4553	R4552
6328	R4555	R4554
6329	R4557	R4556
6330	R4559	R4558
6331	R4561	R4560
6322	R4563	R4562
6333	R4565	R4564
6334	R4567	R4566
6335	R4569	R4568
6336	R4571	R4570
6337	R4573	R4572
6338	R4575	R4574
6339	R4577	R4576
6340	R4579	R4578

#	Corresponding file registers	
	High order	Low order
6341	R4581	R4580
6342	R4583	R4582
6343	R4585	R4584
6344	R4587	R4586
6345	R4589	R4588
6346	R4591	R4590
6347	R4593	R4592
6348	R4595	R4594

PLC constant screen

```
[PLC DATA]                                          SETUP PARAM 6. 4/ 6
#                   #                   #                   #
6301          0     6313          0     6325          0     6337          0
6302          0     6314          0     6326          0     6338          0
6303          0     6315          0     6327          0     6339          0
6304          0     6316          0     6328          0     6340          0
6305          0     6317          0     6329          0     6341          0
6306          0     6318          0     6330          0     6342          0
6307          0     6319          0     6331          0     6343          0
6308          0     6320          0     6332          0     6344          0
6309          0     6321          0     6333          0     6345          0
6310          0     6322          0     6334          0     6346          0
6311          0     6323          0     6335          0     6347          0
6312          0     6324          0     6336          0     6348          0

#(    ) DATA(          )

      MC-ERR        PLC         MACRO         PSW          MENU
```

4.2 Bit Selection Parameters

The parameters that can be used in user PLC include bit selection parameters set in the bit type. Set up data is stored in a file register and is backed up.
For use in bit operation in a sequence program, the file register contents are transferred to temporary memory (M) using the MOV command. In contrast, if data is stored in the file register corresponding to bit selection by using the MOV command etc., it is backed up. However, display remains unchanged. Once display another screen and again select screen.
The corresponding between the bit selection parameters and file registers is listed below. The setting and display screens are also shown.

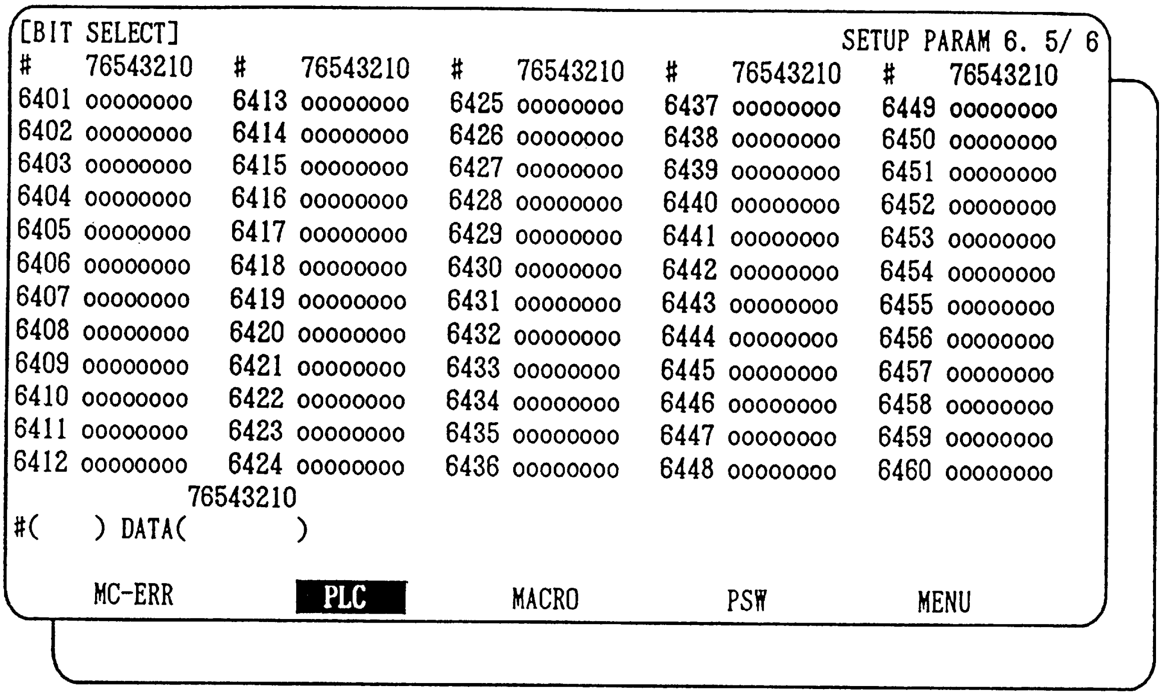
#	Corresponding file register
6401	R4600-LOW
6402	R4600-HIGH
6403	R4601-L
6404	R4601-H
6405	R4602-L
6406	R4602-H
6407	R4603-L
6408	R4603-H
6409	R4604-L
6410	R4604-H
6411	R4605-L
6412	R4605-H
6413	R4606-L
6414	R4606-H
6415	R4607-L
6416	R4607-H
6417	R4608-L
6418	R4608-H
6419	R4609-L
6420	R4609-H
6421	R4610-L
6422	R4610-H
6423	R4611-L
6424	R4611-H
6425	R4612-L
6426	R4612-H
6427	R4613-L
6428	R4613-H
6429	R4614-L
6430	R4614-H
6431	R4615-L
6432	R4615-H

#	Corresponding file register
6433	R4616-LOW
6434	R4616-HIGH
6435	R4617-L
6436	R4617-H
6437	R4618-L
6438	R4618-H
6439	R4619-L
6440	R4619-H
6441	R4620-L
6442	R4620-H
6443	R4621-L
6444	R4621-H
6445	R4622-L
6446	R4622-H
6447	R4623-L
6448	R4623-H
Use bit selection parameters #6401~#6448 freely.	

#	Corresponding file register
6449	R4624-LOW
6450	R4624-HIGH
6451	R4625-L
6452	R4625-H
6453	R4626-L
6454	R4626-H
6455	R4627-L
6456	R4627-H
6457	R4628-L
6458	R4628-H
6459	R4629-L
6460	R4629-H
6461	R4630-L
6462	R4630-H
6463	R4631-L
6464	R4631-H
6465	R4632-L
6466	R4632-H
6467	R4633-L
6468	R4633-H
6469	R4634-L
6470	R4634-H
6471	R4635-L
6472	R4635-H
6473	R4636-L
6474	R4636-H
6475	R4637-L
6476	R4637-H
6477	R4638-L
6478	R4638-H
6479	R4639-L
6480	R4639-H

#	Corresponding file register
6481	R4640-LOW
6482	R4640-HIGH
6483	R4641-L
6484	R4641-H
6485	R4642-L
6486	R4642-H
6487	R4643-L
6488	R4643-H
6489	R4644-L
6490	R4644-H
6491	R4645-L
6492	R4645-H
6493	R4646-L
6494	R4646-H
6495	R4647-L
6496	R4647-H
Bit selection parameter #6449~#6496 are PLC operation selection parameters used by the machine manufacturer and MITSUBISHI. The contents are fixed.	

Bit selection screen



Contents of bit selection parameters #6449~#6496

	Symbol name	7	6	5	4	3	2	1	0
0	Bit selection #6449 R4624L	Control unit thermal alarm invalid	Display unit thermal alarm invalid	—		Counter C hold	Integrating timer T hold	PLC counter program valid	PLC timer program valid
1	#6450 R4624H		—	Alarm/ operator changeover	Message full screen display	—	Operator message	1: R mode / 0: F mode	Alarm message valid
2	#6451 R4625L	—	—	—	—	—	—	—	1: F0 screen / 0: APLC release
3	#6452 R4625H	—				Counter (fixed) hold	Integrating timer (fixed) hold		—
4	#6453 R4626L	—	—	—	—	—		Message Language change code	
5	#6454 R4626H								Macro I/F per part system
6	#6455 R4627L	—	—	—	—	—	—	—	—
7	#6456 R4627H	—	—	—	—	—	—	—	—
8	#6457 R4628L			High-speed input designation 1					
9	#6458 R4628H			High-speed input designation 2					
A	#6459 R4629L	(Reserved)		High-speed input designation 3					
B	#6460 R4629H	(Reserved)		High-speed input designation 4					
C	#6461 R4630L			High-speed output designation 1					
D	#6462 R4630H			High-speed output designation 2					
E	#6463 R4631L	(Reserved)		High-speed output designation 3					
F	#6464 R4631H	(Reserved)		High-speed output designation 4					

	Symbol name	7	6	5	4	3	2	1	0
0	#6465 R4632L	—	—	—	—	—	—	—	—
1	#6466 R4632H	—	—	—	—	—	—	—	—
2	#6467 R4633L	—	—	—	—	—	—	—	—
3	#6468 R4633H	—	—	—	—	—	—	—	—
4	#6469 R4634L				Reserved for system			—	NC alarm output disabled
5	#6470 R4634H								
6	#6471 R4635L	—	—	—	—	—	—	—	—
7	#6472 R4635H	—	—	—	—	—	—	—	—
8	#6473 R4636L	—							—
9	#6474 R4636H								
A	#6475 R4637L								
B	#6476 R4637H								
C	#6477 R4638L								
D	#6478 R4638H								
E	#6479 R4639L								
F	#6480 R4639H								

(Note 1) The bits marked are used by the system. Be sure to set to 0.
(Note 2) Parameters #6481~#6496 are not used. They are for debugging at MITSUBISHI.
(Note 3) High speed input designation is valid for only the devices assigned to the remote I/O.
Note that as for the devices assigned to the input signals from the network such as HR863 Q bus bridge or HR865 CC-Link, high-speed input designation is invalid.

5. Explanation of Devices

5.1 Devices and Device Numbers

The devices are address symbols to identify signals handled in PLC. The device numbers are serial numbers assigned to the devices. The device numbers of devices X, Y, B, W, and H are represented in hexadecimal notation. The device numbers of other devices are represented in decimal notation.

5.2 Device List

Device	Device range		Units	Details
X*	X0 to XAFF	2816 points	1-bit	Input signals to the PLC. Machine input, etc.
Y*	Y0 to YE7F	3712 points	1-bit	Output signals from the PLC. Machine output, etc.
M	M0 to M8191	8192 points	1-bit	For temporary memory
L	L0 to L255	256 points	1-bit	Latch relay (Backup memory)
F	F0 to F127	128 points	1-bit	For temporary memory. Alarm message interface
SB	SB0 to SB1FF	512 points	1-bit	Special relay for links
B	B0 to B1FFF	8192 points	1-bit	Link relay
SM*	SM0 to SM127	128 points	1-bit	Special relay
V	V0 to V255	256 points	1-bit	Edge relay
SW	SW0 to SW1FF	512 points	16-bit	Special register for links
SD	SD0 to SD127	128 points	16-bit	Special register
T	T0 to T15	16 points	1-bit/16-bit	10ms unit timer
	T16 to T95	80 points	1-bit/16-bit	100ms unit timer
	T96 to T103	8 points	1-bit/16-bit	100ms incremented timer
	T104 to T143	40 points	1-bit/16-bit	10ms unit timer (Fixed timers)
	T144 to T239	96 points	1-bit/16-bit	100ms unit timer (Fixed timers)
	T240 to T255	16 points	1-bit/16-bit	100ms incremented timer (Fixed timers)
	T0000 to T0255	256 points	1-bit	T1: Timer coil
	T1000 to T1255	256 points	1-bit	T0: Timer contact
	T2000 to T2255	256 points	16-bit	TS: Timer setting value
	T3000 to T3255	256 points	16-bit	TA: Timer current value
C	C0 to C23	24 points	1-bit/16-bit	Counter
	C24 to C127	104 points	1-bit/16-bit	Counter (Fixed counters)
	C0000 to C0127	128 points	1-bit	C1: Counter coil
	C1000 to C1127	128 points	1-bit	C0: Counter contact
	C2000 to C2127	128 points	16-bit	CS: Counter setting value
	C3000 to C3127	128 points	16-bit	CA: Counter current value
D	D0 to D8191 (Note3)	8192 points	16-bit/32-bit	Data register
R*	R0 to R8191	8192 points	16-bit/32-bit	File register. CNC word I/F
W	W0 to W1FFF	8192 points	16-bit/32-bit	Link register
Z	Z0 to Z13	14 points	16-bit	Address index
N	N0 to N7			Master control's nesting level
P*	P0 to P255 P360 to P379			Conditional jump, subroutine call label
K	K-32768 to K32767			Decimal constant for 16-bit command
	K-2147483647 to K2147483647			Decimal constant for 32-bit command
H	H0 to HFFFF			Hexadecimal constant for 16-bit command
	H0 to HFFFFFFFF			Hexadecimal constant for 32-bit command

(Note 1) The applications of the devices having a * in the device column are separately determined. Do not use the undefined device Nos., even if they are open.

(Note 2) The fixed timer and fixed counter cannot be changed with the numerical setting. Note that those can be changed with the numerical setting when D or R device is specified.

(Note 3) The range of D0 to D8191 devices can be used on the software version D0 and higher.

5.3 Detailed Explanation of Devices

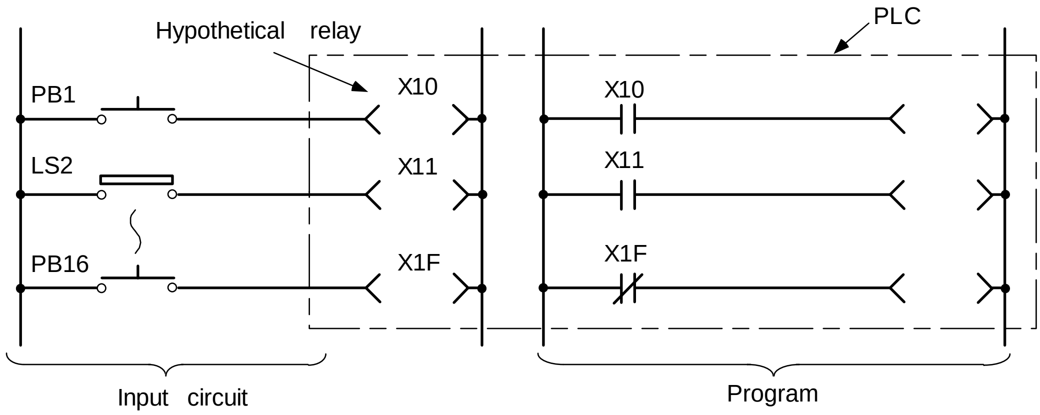
The devices used with the PLC are described below.

5.3.1 Input/output X, Y

Input/output X and Y are a window for executing communication with the PLC and external device or controller.

Input X

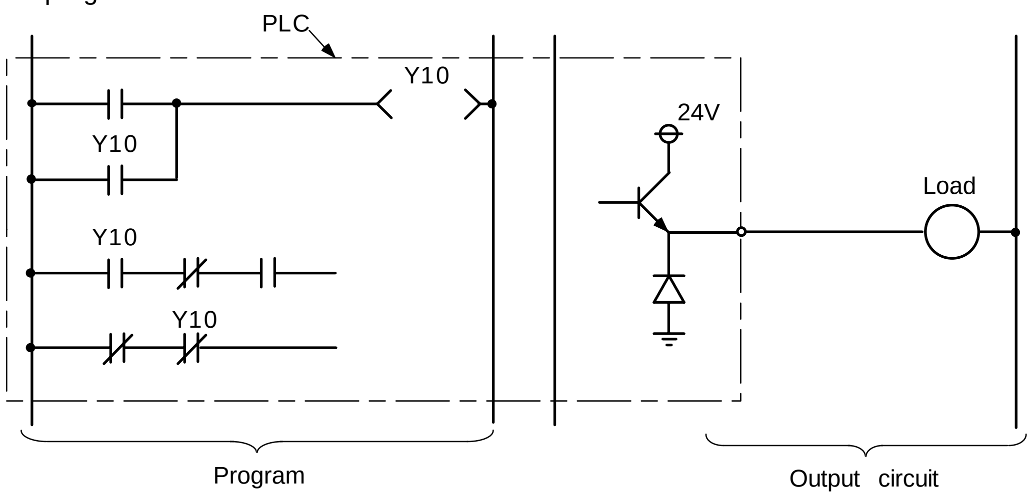
(1) This issued commands or data from an external device such as a push-button, changeover switch, limit switch or digital switch to the PLC.
(2) Assuming that there is a hypothetical relay Xn built-in the PLC per input point, the program uses the "A" contact and "B" contact of that Xn.
(3) There is no limit to the No. of "A" contacts and "B" contacts of the input Xn that can be used in the program.



(4) The input No. is expressed with a hexadecimal.

Output Y

(1) This outputs the results of the program control to the solenoid, magnetic switch, signal lamp or digital indicator, etc.
(2) The output (Y) can be retrieved with the equivalent of one "A" contact.
(3) There is no limit to the No. of "A" contacts and "B" contacts of the output Yn that can be used in the program.



(4) The output No. is expressed with a hexadecimal.

5.3.2 Internal Relays M and F, Latch Relay L

The internal relay and latch relay are auxiliary relays in the PLC that cannot directly output to an external source.

Internal relay M

(a) The relay is cleared when the power is turned OFF.
(b) There is no limit to the No. of "A" contacts and "B" contacts of the internal relays that can be used in the program.
(c) The internal relay No. is expressed with a decimal.

Internal relay F

Internal relay F is an interface for the alarm message display.
Use the bit selection parameter to determine whether to use this relay for the alarm message interface. The target will be F0 to F127. This internal relay can be used in the same manner as the internal relay M when not used as the alarm message interface.

Latch relay L

(a) The original state is held even when the power is turned OFF.
(b) There is no limit to the No. of "A" contacts and "B" contacts of the latch relay that can be used in the program.
(c) The latch No. is expressed with a decimal.

5.3.3 Special Relays SM

The special relays SM are relays having fixed applications such as the carrier flag for operation results and the display request signal to the setting and display unit. Even the relays of SM0 to SM127 that are not currently used must not be used as temporary memory.

Special relays SM

(a) This relay is cleared when the power is turned OFF.
(b) There is no limit to the No. of "A" contacts and "B" contacts of the special relays that can be used in the program.
(c) The special relay No. is expressed with a decimal.

5.3.4 Link Relay B, Link Register W

(1) Link relay B is a bit type device used for the data link in each link function. An unused part can be used as first memory, etc., however, the step No. will be increased.
(2) Link register W is a word type device used for the data link in each link function. An unused part can be used as first memory, etc., however, the step No. will be increased.

Link relay B, Link register W

(a) These are cleared when the power is turned OFF.
(b) There is no limit to the No. that can be used in the program.
(c) The register No. is expressed with a hexadecimal.

5.3.5 Special Relay for Link SB, Special Register for Link SW

Special relay for link SB

(a) Special relay for link SB is a relay used for the exchange of data between each network card and PLC program.
(b) SB is controlled to DN or OFF by various causes that happen during the data link. Thus, monitoring this relay helps you to find an error state of the data link.

Special register for link SW

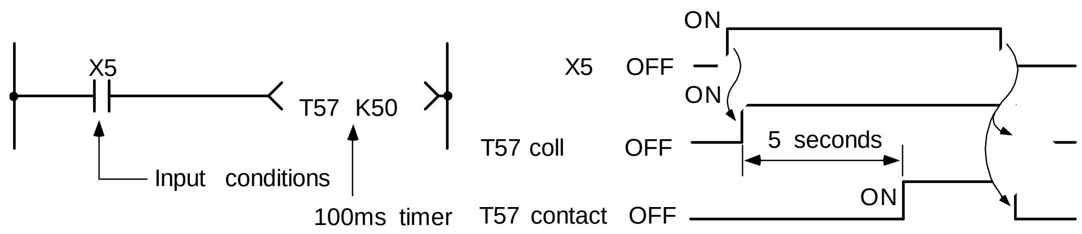
(a) Special register for link SW is a relay used for the exchange between each network card and PLC program.
(b) Information during data link is stored in this register. Thus, monitoring this register helps you to find an error occurrence point and the causes.

5.3.6 Timer T

(1) The 100ms timer, 10ms timer and 100ms integrated timer are available for this count-up type timer.

100ms Timer T

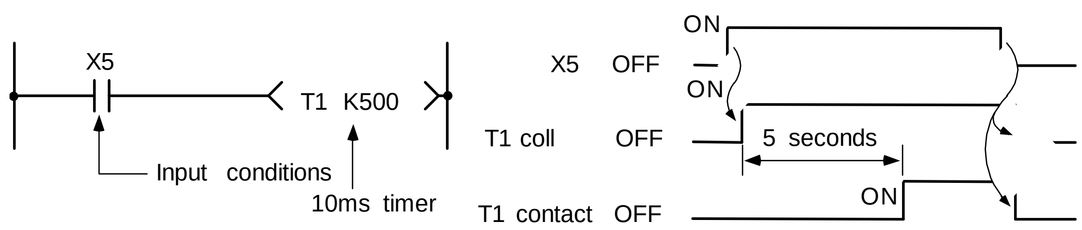
(a) When the input conditions are set, the count starts. When the set value is counted, that timer contact will turn ON.
(b) If the input conditions are turned OFF, the 100ms timer count value will be set to 0, and the contact will turn OFF.



(c) The value is set with a decimal, and can be designated from 1 to 32767 (0.1 to 3276.7 s). The data register (D) data can also be used as the setting value. File register (R) cannot be used.

10ms Timer T

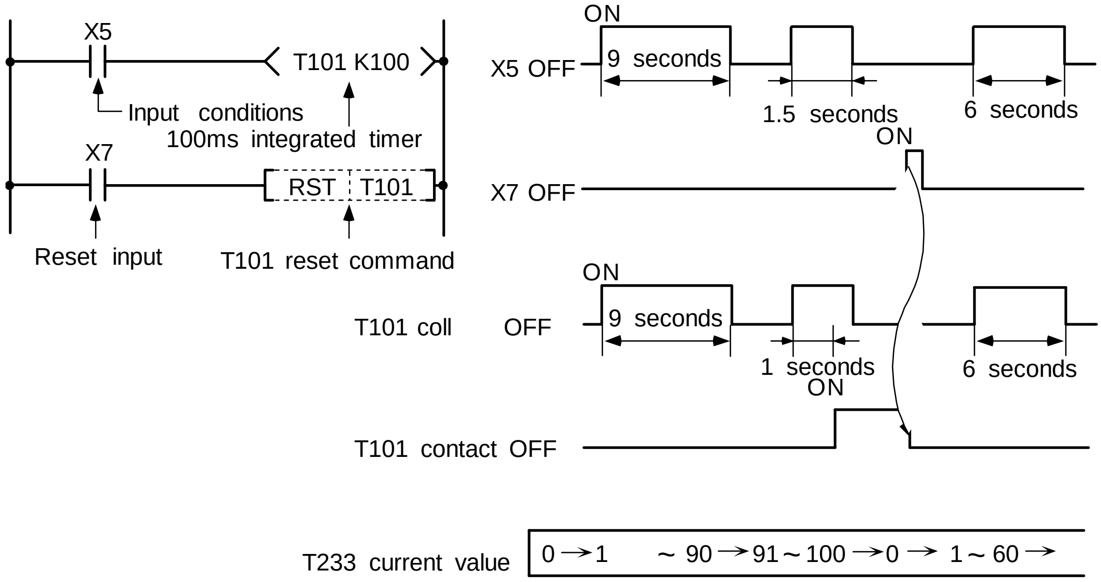
(a) When the input conditions are set, the count starts. When the set value is counted, that timer contact will turn ON.
(b) If the input conditions are turned OFF, the 10ms timer count value will be set to 0, and the contact will turn OFF.



(c) The value is set with a decimal, and can be designated from 1 to 32767 (0.01 to 327.67 s). The data register (D) data can also be used as the setting value. File register (R) cannot be used.

100ms Integrated timer T

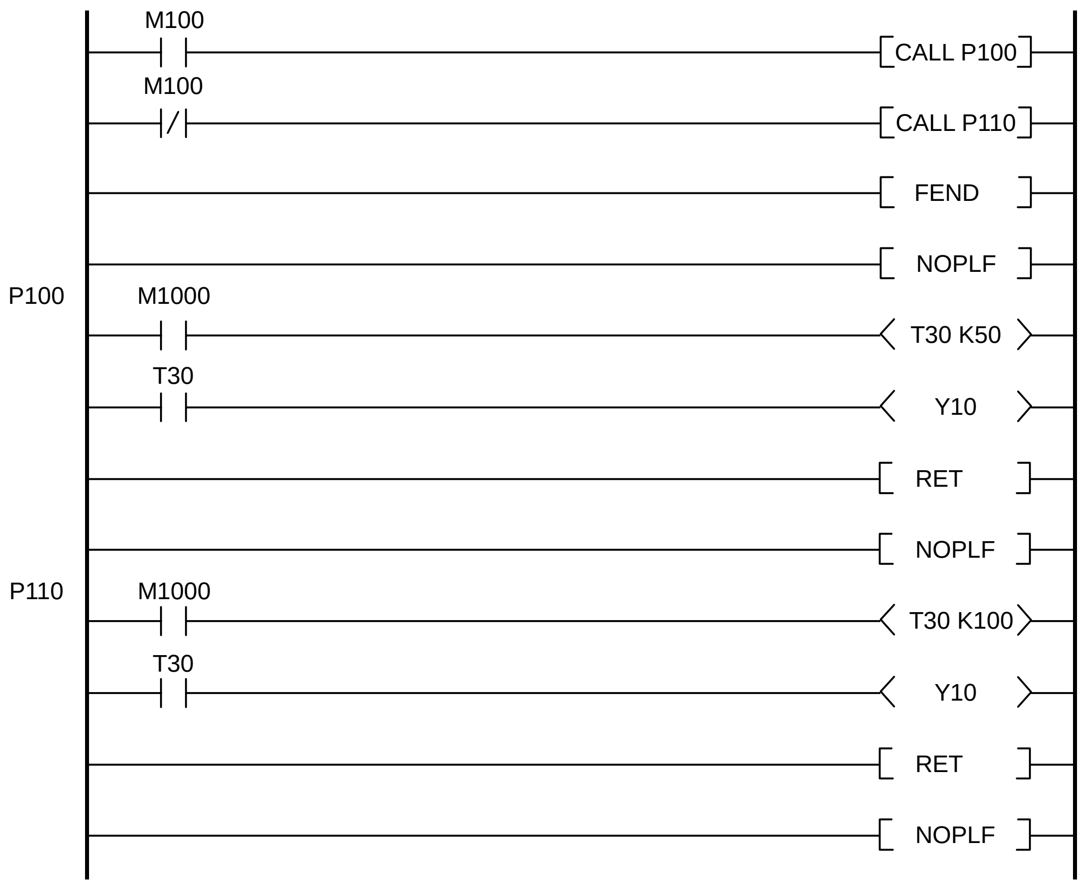
(a) When the input conditions are set, the count starts. When the set value is counted, that timer contact will turn ON.
(b) Even the input conditions are turned OFF, the 100ms integrated timer current value (count value) will be held, and the contact state will not change.
(c) The 100ms integrated timer count value will be set to 0 and the contact will turn OFF when the RST command is executed.



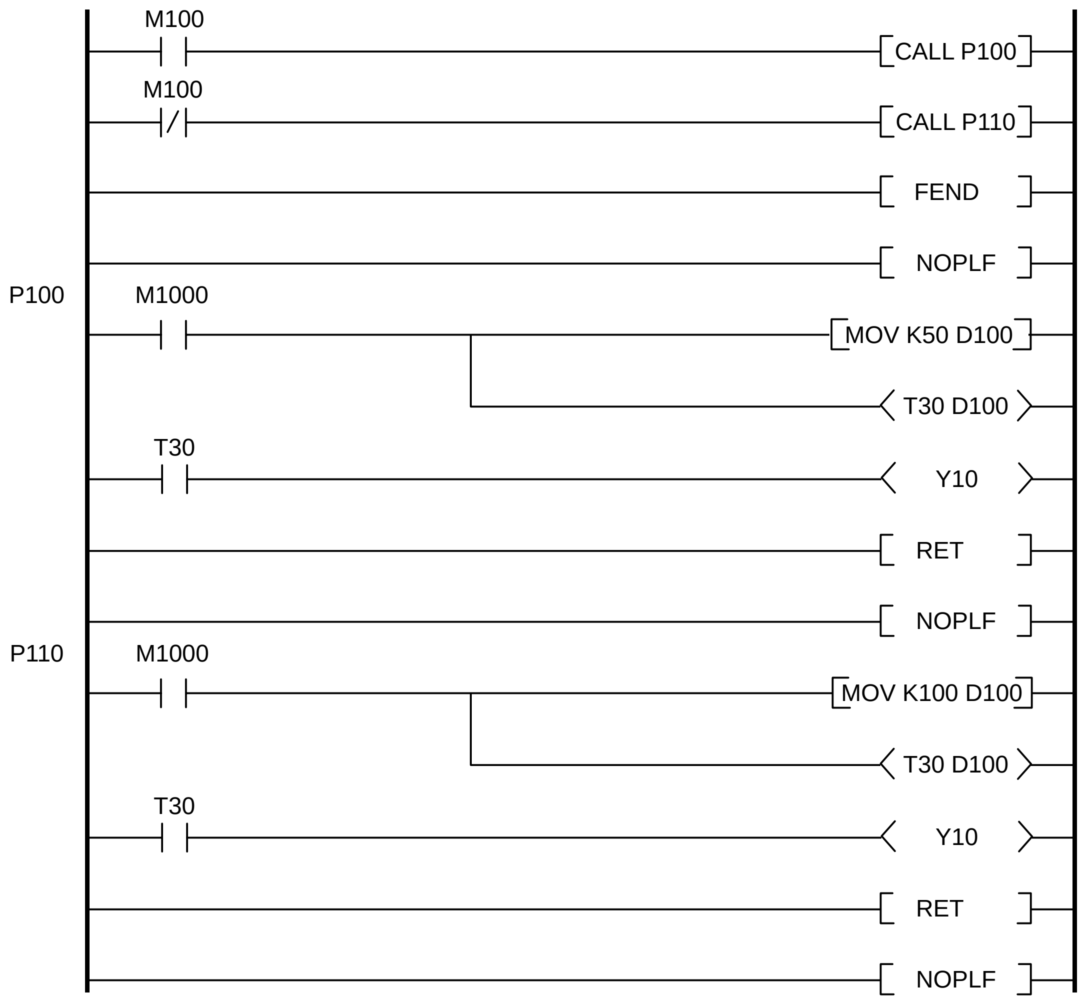
(d) The value is set with a decimal, and can be designated from 1 to 32767 (0.1 to 3267.7 s). The data register (D) data can also be used as the setting value. File register (R) cannot be used.
(e) When the bit selection parameter is set, the 100ms integrated timer current value (count value) will be held even when the power is turned OFF.

(2) Setting of timer setting value from setting and display unit
The timer setting value can be set with the setting and display unit using device T0 to T103. (Variable timer)
Whether the setting value (Kn) programmed with the sequence program or the setting value set from the setting and display unit is valid is selected with the bit selection parameters. The changeover is made in a group for T0 to T103. Even when set from the setting and display unit, the setting value (Kn) program will be required in the sequence program. However, the Kn value will be ignored. When the data register (D) is used for the setting value, the data register (D) details will be used as the setting value regardless of the parameter.

(3) Cautions for when using the same timer at two or more positions.
The timer programmed last will be valid even if the timer is set in the subprogram which is not ladder-processed according to the branch as shown in the following circuit.



In the circuit above, when M100 is ON, the subprogram of P110 will not be processed. However, the value of K100 programmed in the subprogram of P110 will be valid as the T30 timer value.
Therefore, when the circuit above is executed, the timer setting value will be the value of K100 (100(s) obtained by 100(ms)*100), in other words, Y10 device will be ON after 10 seconds although the subprogram of P100 is processed.
To make a correct operation, program the circuit as shown in the next page.



The circuit above enables that Y10 device will be ON after 5 seconds if M100 is ON and after 10 seconds if M100 is OFF.

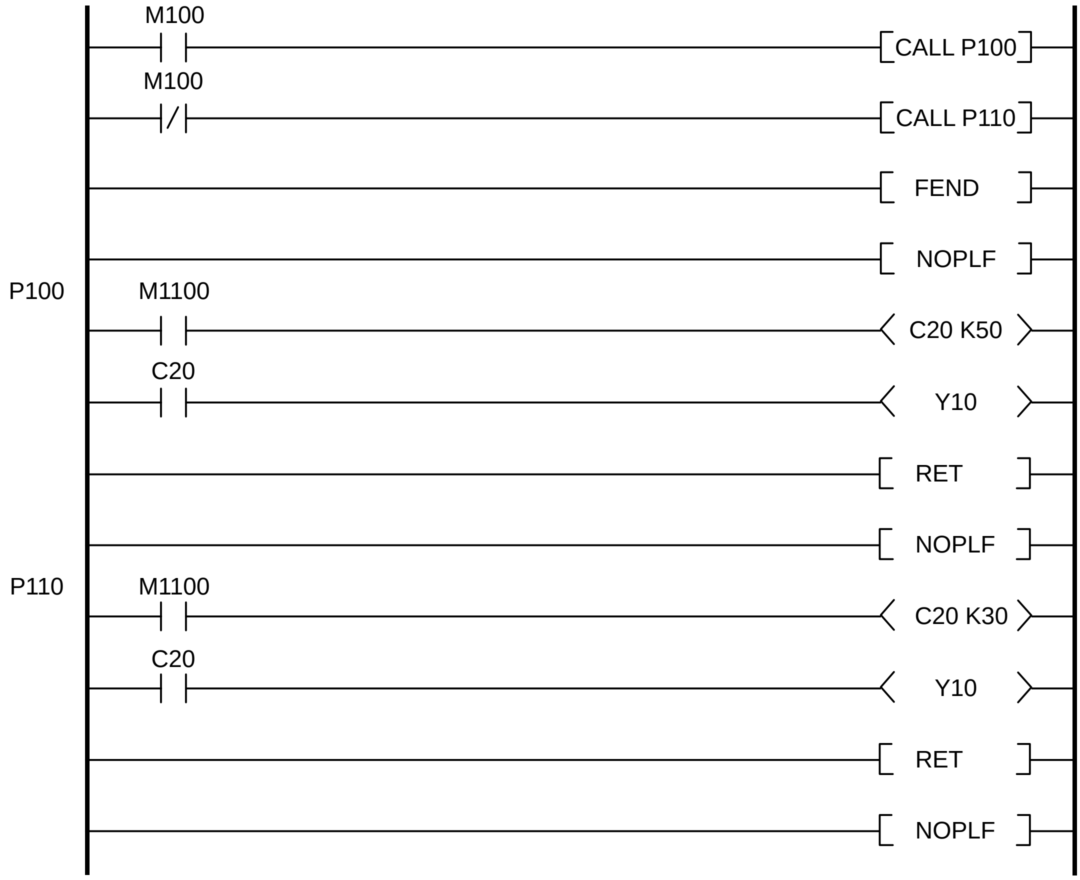
5.3.7 Counter C

(1) The counter counts up and detects the rising edge of the input conditions. Thus, the count will not take place when the input conditions are ON.

Counter C	
(a) The value is set with a decimal, and can be designated from 1 to 32767. The data register (D) data can also be used as the setting value. File register (R) cannot be used. (b) The counter count value will not be cleared even if the input conditions turn OFF. The counter count value must be cleared with the RST command. (c) When the bit selection parameter is set, the counter current value (count value) will be held even when the power is turned OFF.	

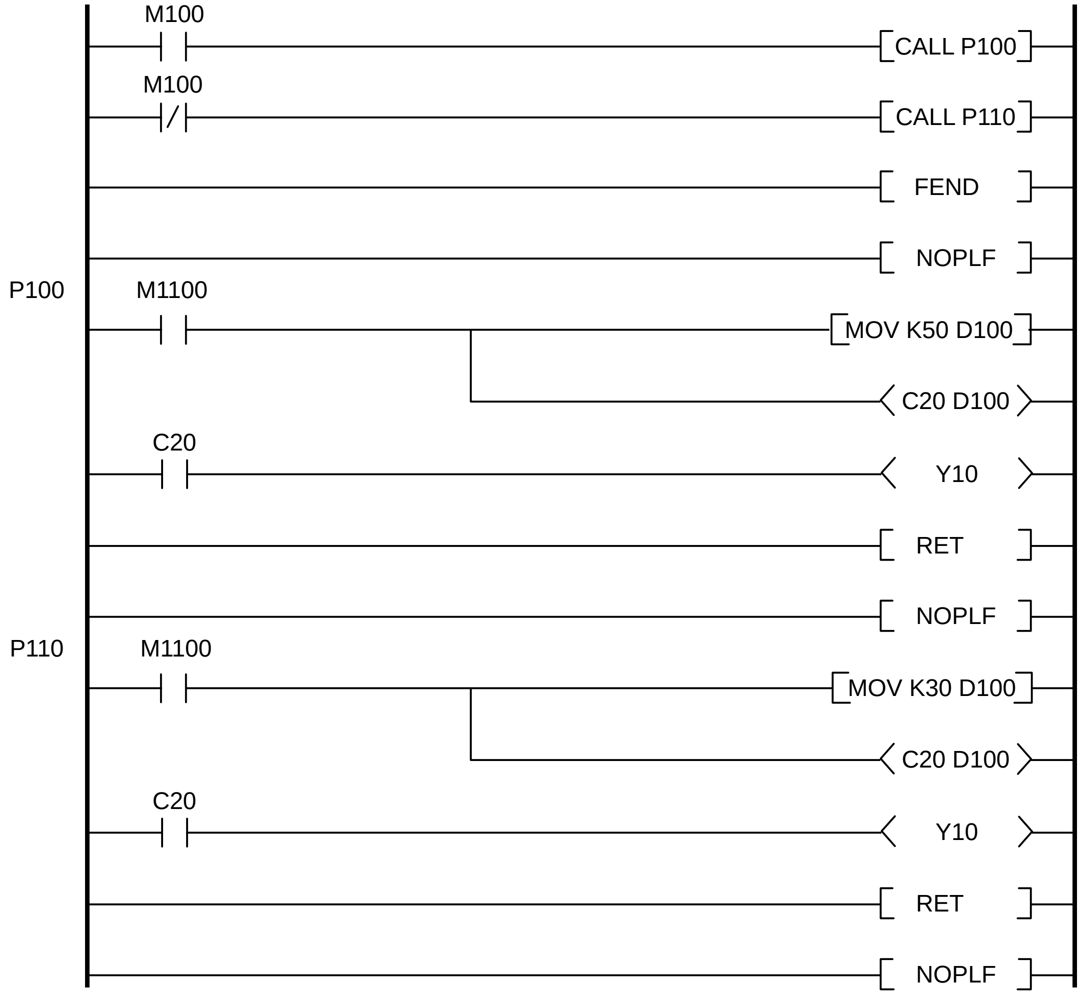
(2) Setting of counter setting value from setting and display unit
The counter setting value can be set with the setting and display unit using device Co to C23. (Variable counter)
Whether the setting value (Kn) programmed with the sequence program or the setting value set from the setting and display unit is valid is selected with the bit selection parameters. The changeover is made in a group for C0 to C23. Even when set from the setting and display unit, the setting value (Kn) program will be required in the sequence program. However, the Kn value will be ignored. When the data register (D) is used for the setting value, the data register (D) details will be used as the setting value regardless of the parameter.

(3) Cautions for when using the same counter at two or more positions.
The counter programmed last will be valid even if the counter is set in the subprogram which is not ladder-processed according to the branch as shown in the following circuit.



In the circuit above, when M100 is ON, the subprogram of P110 will not be processed. However, the value of K30 programmed in the subprogram of P110 will be valid as the C20 counter value. Therefore, when the circuit above is executed, the counter setting value will be the value of K30, in other words, Y10 device will be ON after 30 counts although the subprogram of P100 processed.

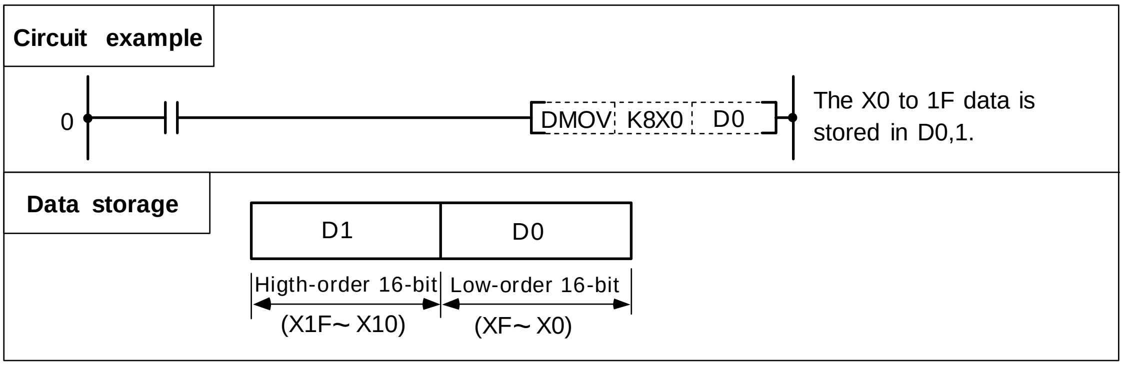
To make a correct operation, program the circuit as shown in the next page.



The circuit above enables that Y10 device will be ON after 50 counts if M100 is ON and after 30 counts if M100 is OFF.

5.3.8 Data Register D

(1) The data register is the memory that stores the data in the PLC.
(2) The data register has a 1-point 16-bit configuration, and can be read and written in 16-bit units. To handle 32-bit data, two points must be used. The data register No. designated with the 32-bit command will be the low-order 16-bit, and the designated data register No. +1 will be the high-order 16-bit.
(Example) Use of the DMOV command is shown below.



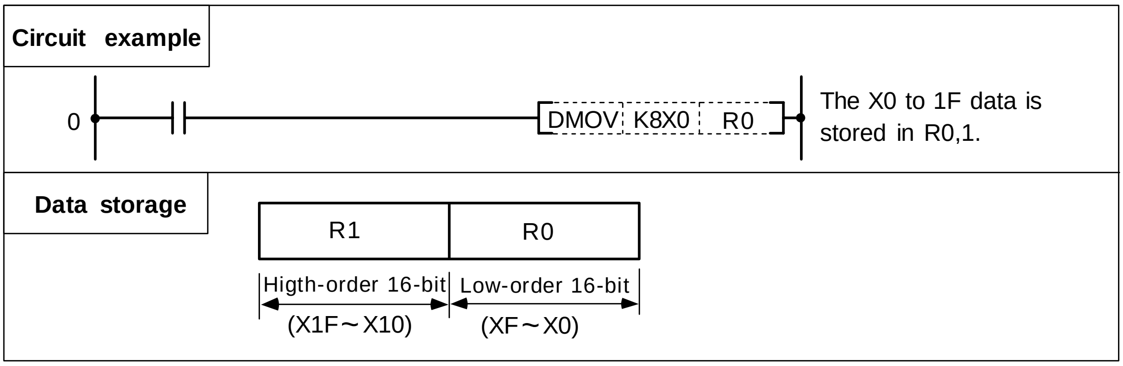
(3) The data that is stored once in the sequence program is held until other data is stored.
(4) The data stored in the data register is cleared when the power is turned OFF.
(5) Values that can be stored: Decimal -32768 to 32767, Hexadecimal 0 to FFFF } For 16-bit command (Using Dn)
Decimal -2147483648 to 2147483647, Hexadecimal 0 to FFFFFFFF } For 32-bit command (Using Dn+1, Dn)
(6) Data registers D0 to D8191 are all user release data registers.

(Note) The range of D0 to D8191 is valid on the software version D0 and higher.

5.3.9 File Register R

(1) As with the data registers, the file registers are memories used to store data. However, there are some that have fixed applications, and those that are released.
(2) The file register has a 1-point 16-bit configuration, and can be read and written in 16-bit units.
To handle 32-bit data, two points must be used. The file register No. designated with the 32-bit command will be the low-order 16-bit, and the designated file register No. +1 will be the high-order 16-bit.

(Example) Use of the DMOV command is shown below.



(3) The data that is stored once in the sequence program is held until other data is stored.
(4) The data stored in the file registers R4000 to R4499 and R6400 to R7199 and the user release registers R6400 to R7199 is not cleared when the power is turned OFF.
The other file registers have fixed applications such as interface of the PLC and CNC, parameter interface, etc.
(5) Values that can be stored: Decimal -32768 to 32767, Hexadecimal 0 to FFFF } For 16-bit command (Using Dn)
Decimal -2147483648 to 2147483647, Hexadecimal 0 to FFFFFFFF } For 32-bit command (Using Dn+1, Dn)

5.3.10 Special Register SD

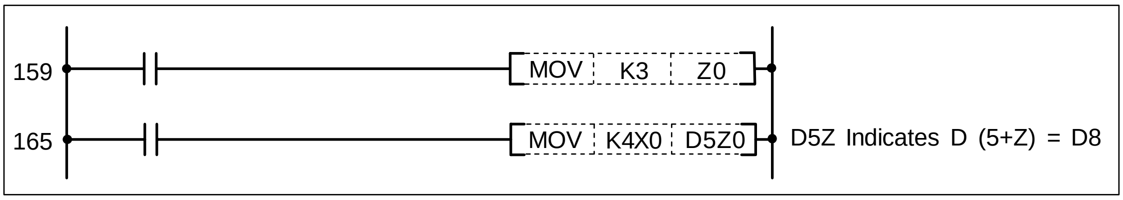
(1) Special register SD is a data register that the applications are fixed as 1-second counter, etc. Do not use even a part not used currently in SD0 to SD127 for other purpose such as temporary memory.

Special register SD

(a) The register is cleared when the power is turned OFF.
(b) There is no limit to the No. that can be used in the program.
(c) The special register No. is expressed with a hexadecimal.

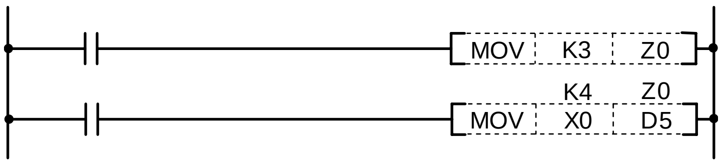
5.3.11 Index Register Z

(1) Z index register is available.
(2) The index register is used as ornaments for the device.



(3) The index register has a 1-point 16-bit configuration, and can be read and written in 16-bit units.
(4) The data stored in the index register is cleared when the power is turned OFF.
(5) Values that can be stored: Decimal -32768 to 32767
Hexadecimal 0 to FFFF

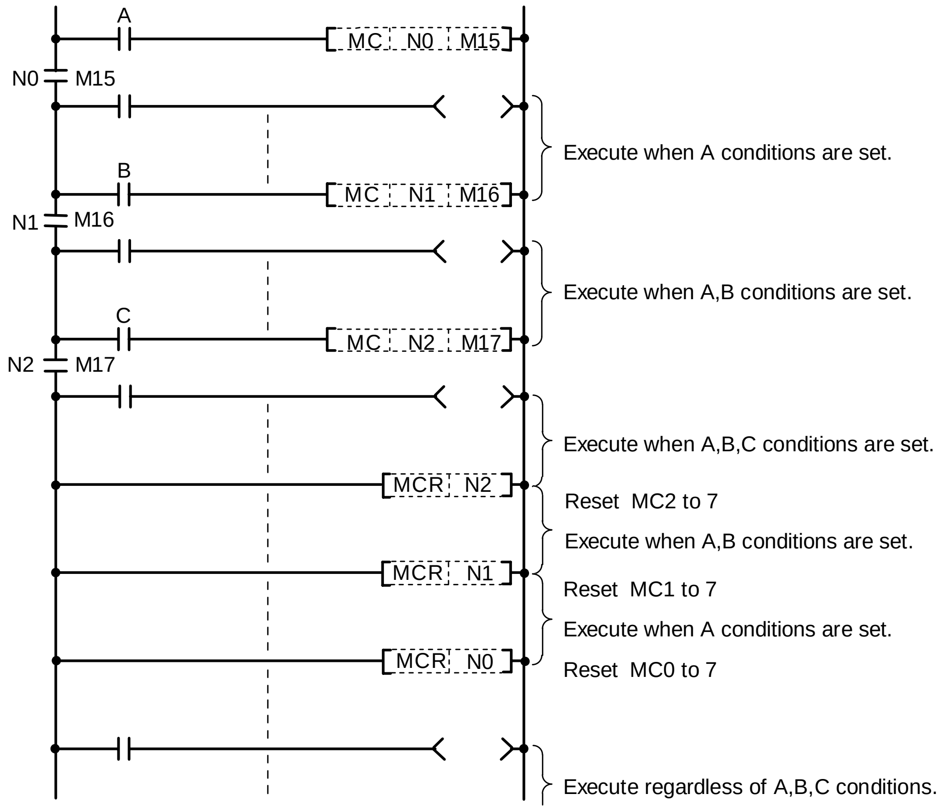
(Note1) The display of the index registers Z0 is as shown below.



(Note2) Don't use index resistor Z0, Z1 in high-speed program processing because the system use these resistor in some function commands. If the resistor is used, operation of main processing function commands may be unfixed.

5.3.12 Nesting N

(1) This indicates the master control nesting structure.
(2) The master control nesting (N) is used in order from smallest number.



(a) The conditions for each master control to turn ON are as follow.

MC N0 M15 .... ON when condition A is ON

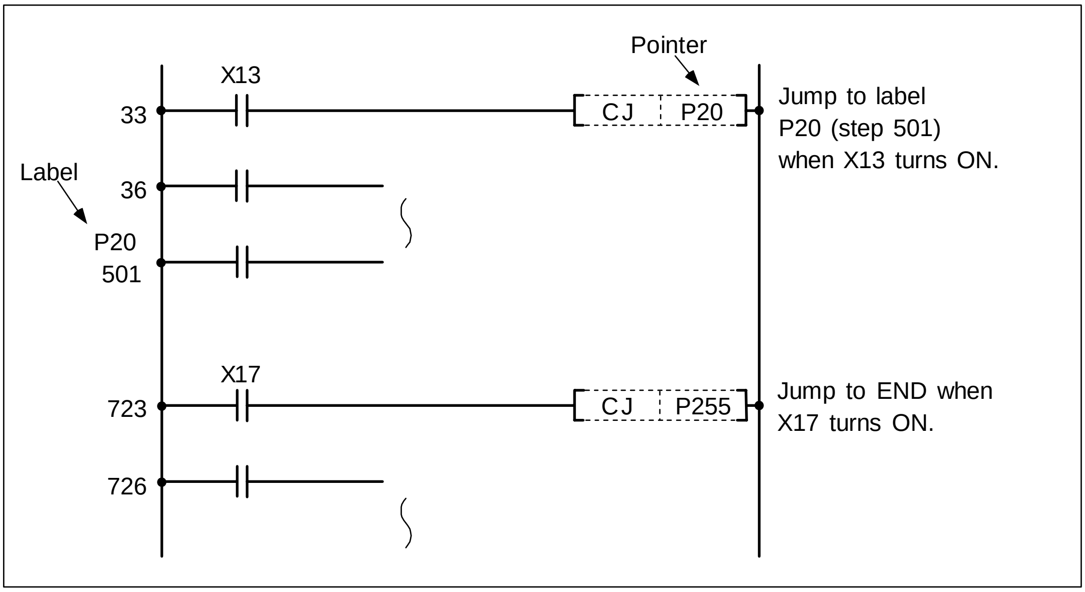
MC N0 M16 .... ON when conditions A, B are ON

MC N0 M17 .... ON when conditions A, B, C are ON

(b) The timer and counter when the master control is OFF is as follows.
· 100ms timer, 10ms timer: The count value is set to 0.
· 100ms integrated timer : The current count value is retained.
· Counter : The current counter value is retained.
· OUT command : All turn OFF.

5.3.13 Pointer P

(1) The pointer indicates the branch command (CJ, JMP, CALL) jump destination. The pointer No. assigned at the jump destination head is called the label.
(2) Pointers P0 to P249 are user release pointers.
(3) P255 always indicates END.
(P255 can be used as a device for CJ command, etc, but cannot be used as a label. This cannot be used for the CALL command device.)



(4) The special usages of the pointers other than P255 are shown below.
P251, P360 to P368: Label for starting PLC high-speed processing program.
P252, P370 to P378: Label for starting PLC main (ladder) processing program.

(Note 1) Do not omit the label of P252 and P370 to P378 even when there is only a PLC main processing program.
(Note 2) P251, P252 and P360 to P378 cannot be used as CJ, JMP or CALL command devices.
(Note 3) Do not create a program in which the P** in the PLC high speed processing program is jumped to from the PLC main processing program.
(Note 4) The P** used as a CJ, JMP or CALL command device must be programmed so that the **P will be sure to exist as a label in the same file as the command.

The PLC will not operate correctly if Notes 1 to 4 are not observed.

5.3.14 Decimal Constant K

(1) The decimal constant can be used in the following ways.
 (a) Timer counter setting value: Designate in the range of 1 to 32767.
 (b) Pointer No.: 0 to 159
 (c) Bit device digit designation: 1 to 8
 (d) Basic command, function command, exclusive command value setting
  · 16-bit command: -32768 to 32767
  · 32-bit command: -2147483648 to 2147483647
(2) The decimal constant is stored in the BIN value (binary) in the PLC.

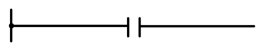
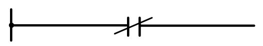
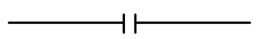
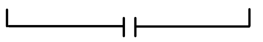
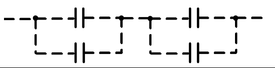
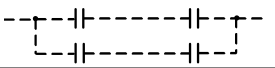
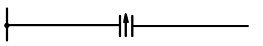
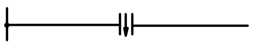
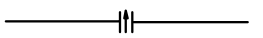
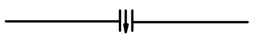
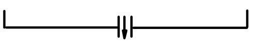
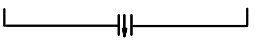
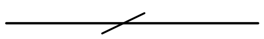
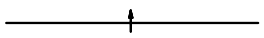
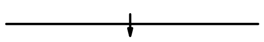
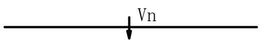
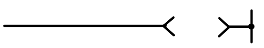
5.3.15 Hexadecimal Constant H

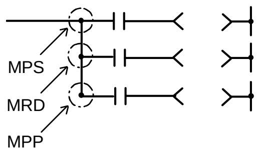
(1) The hexadecimal constant is used to designate the basic command, function command and exclusive command values.
 · 16-bit command: 0 to FFFF
 · 32-bit command: 0 to FFFFFFFF

6. Explanation of Commands

6.1 Command List

6.1.1 Basic Commands

Class	Process unit	Command sign	Symbol	Process details	No. of steps	Page
Basic command	Bit	LD		Start of logic operation (A contact operation start)	1/2	104
		LDI		Start of logic denial operation (B contact operation start)	1/2	104
		AND		Logical AND (A contact serial connection)	1/2	106
		ANI		Logical AND denial (B contact serial connection)	1/2	106
		OR		Logical OR (A contact parallel connection)	1/2	108
		ORI		Logical OR denial (B contact parallel connection)	1/2	108
		ANB		AND between logical blocks (Serial connection between blocks)	1	110
		ORB		OR between logical blocks (Parallel connection between blocks)	1	112
		LDP		Start of leading edge pulse operation	6	114
		LDF		Start of trailing edge pulse operation	6	114
		ANDP		Leading edge pulse serial connection	6	114
		ANDF		Trailing edge pulse serial connection	6	114
		ORP		Leading edge pulse parallel connection	6	114
		ORF		Trailing edge pulse parallel connection	6	114
		INV		Inversion of operation result	4	116
		MEP		Conversion of operation result to leading edge pulse	4	117
		MEF		Conversion of operation result to trailing edge pulse	4	117
		EGP		Conversion of operation result to leading edge pulse (registration with Vn)	6	118
		EGF		Conversion of operation result to trailing edge pulse (registration with Vn)	6	118
		OUT		Device output	1/2/3	120

Class	Process unit	Command sign	Symbol	Process details	No. of steps	Page
Basic command	Bit	SET	SET D	Device set	1/2	126
		RST	RST D	Device reset	1/2	128
		MC	MC n D	Master control start	2	130
		MCR	MCR n	Master control release	1	130
		PLS	PLS D	Generate one cycle worth of pulses at rising edge of input signal	2	132
		PLF	PLF D	Generate one cycle worth of pulses at falling edge of input signal	2	132
		FF	FF D	Reversal of device output	7	134
		SFT	SFT D	Device 1-bit shift	3	136
		SFTP	SFTP D		9	136
		MPS		Registration of logical operation result	1	138
		MRD		Read of operation results registered in MPS	1	138
		MPP		Reading and resetting of operation results registered in MPS	1	138
		NOP		Ignored (For program deletion or space)	1	140
		NOPLF	NOPLF	Ignored (For change pages during printing)	1	140
		PAGE	PAGE n	Ignored (Subsequent programs will be controlled from 0 step of page n)	2	140

6.1.2 Function Commands

(1) Comparison commands

Class	Process unit	Command sign	Symbol	Process details	No. of steps	Page
=	16-bit	LD=	[= S1 S2]	Continuity state when (S1) = (S2) Non-continuity state when (S1) =/ (S2)	3	146
		AND=	[= S1 S2]		3	146
		OR=	[= S1 S2]		3	146
	32-bit	LDD=	[D= S1 S2]	Continuity state when (S1+1, S1)=(S2+1, S2) Non-continuity state when (S1+1, S1) =/ (S2+1, S2)	3/4	148
		ANDD=	[D= S1 S2]		3/4	148
		ORD=	[D= S1 S2]		3/4	148
>	16-bit	LD>	[> S1 S2]	Continuity state when (S1) > (S2) Non-continuity state when (S1) <= (S2)	3	150
		AND>	[> S1 S2]		3	150
		OR>	[> S1 S2]		3	150
		LD>=	[>= S1 S2]	Continuity state when (S1) >= (S2) Non-continuity state when (S1) < (S2)	3	150
		AND>=	[>= S1 S2]		3	150
		OR>=	[>= S1 S2]		3	150
	32-bit	LDD>	[D> S1 S2]	Continuity state when (S1+1, S1) > (S2+1, S2) Non-continuity state when (S1+1, S1) <= (S2+1, S2)	3/4	152
		ANDD>	[D> S1 S2]		3/4	152
		ORD>	[D> S1 S2]		3/4	152
		LDD>=	[D>= S1 S2]	Continuity state when (S1+1, S1) >= (S2+1, S2) Non-continuity state when (S1+1, S1) < (S2+1, S2)	3/4	152
		ANDD>=	[D>= S1 S2]		3/4	152
		ORD>=	[D>= S1 S2]		3/4	152

Class	Process unit	Command sign	Symbol	Process details	No. of steps	Page
<	16-bit	LD<	[< S1 S2]	Continuity state when (S1) < (S2) Non-continuity state when (S1) >= (S2)	3	154
		AND<	[< S1 S2]		3	154
		OR<	[< S1 S2]		3	154
		LD<=	[<= S1 S2]	Continuity state when (S1) <= (S2) Non-continuity state when (S1) > (S2)	3	154
		AND<=	[<= S1 S2]		3	154
		OR<=	[<= S1 S2]		3	154
		LD<>	[<> S1 S2]	Continuity state when (S1) =/ (S2) Non-continuity state when (S1) = (S2)	3	158
		AND<>	[<> S1 S2]		3	158
		OR<>	[<> S1 S2]		3	158
	32-bit	LDD<	[D< S1 S2]	Continuity state when (S1+1, S1) < (S2+1, S2) Non-continuity state when (S1+1, S1) >= (S2+1, S2)	3/4	156
		ANDD<	[D< S1 S2]		3/4	156
		ORD<	[D< S1 S2]		3/4	156
		LDD<=	[D<= S1 S2]	Continuity state when (S1+1, S1) <= (S2+1, S2) Non-continuity state when (S1+1, S1) > (S2+1, S2)	3/4	156
		ANDD<=	[D<= S1 S2]		3/4	156
		ORD<=	[D<= S1 S2]		3/4	156
		LDD<>	[D<> S1 S2]	Continuity state when (S1+1, S1) =/ (S2+1, S2) Non-continuity state when (S1+1, S1) = (S2+1, S2)	3/4	160
		ANDD<>	[D<> S1 S2]		3/4	160
		ORD<>	[D<> S1 S2]		3/4	160

(2) Arithmetic operation commands

Class	Process unit	Command sign	Symbol	Process details	No. of steps	Page
+ (BIN)	16-bit	+	+ S D	(D) + (S)→(D) (BIN)	3	162
		+P	+P S D		9	162
		+	+ S1 S2 D	(S1) + (S2)→(D) (BIN)	4	162
		+P	+P S1 S2 D		10	162
	32-bit	D+	D+ S D	(D+1,D) + (S+1,S)→(D+1,D) (BIN)	3/4	164
		D+P	D+P S D		9/10	164
		D+	D+ S1 S2 D	(S1+1,S1)+(S2+1,S2)→(D+1,D) (BIN)	4/5	164
		D+P	D+P S1 S2 D		10/11	164
− (BIN)	16-bit	-	- S D	(D)−(S)→(D) (BIN)	3	166
		-P	-P S D		9	166
		-	- S1 S2 D	(S1)−(S2)→(D) (BIN)	4	166
		-P	-P S1 S2 D		10	166
	32-bit	D-	D- S D	(D+1,D)−(S+1,S)→(D+1,D) (BIN)	3/4	168
		D-P	D-P S D		9/10	168
		D-	D- S1 S2 D	(S1+1,S1)−(S2+1,S2)→(D+1,D) (BIN)	4/5	168
		D-P	D-P S1 S2 D		10/11	168
* (BIN)	16-bit	*	* S1 S2 D	(S1) x (S2)→(D+1,D) (BIN)	4	170
		*P	*P S1 S2 D		10	170
	32-bit	D*	D* S1 S2 D	(S1+1,S1) x (S2+1,S2) →(D+3,D+2,D+1,D) (BIN)	5/6	172
		D*P	D*P S1 S2 D		11/12	172

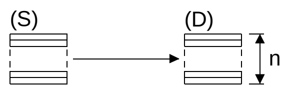
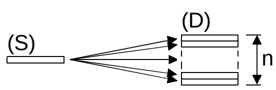
Class	Process unit	Command sign	Symbol	Process details	No. of steps	Page
/ (BIN)	16-bit	/	/ S1 S2 D	(S1) ÷ (S2) → Quotient (D) Remainder (D+1) (BIN)	5/6	174
		/P	/P S1 S2 D		11/12	174
	32-bit	D/	D/ S1 S2 D	(S1+1, S1) ÷ (S2+1, S2) → Quotient (D+1,D) Remainder (D+3, D+2) (BIN)	5/6	176
		D/P	D/P S1 S2 D		11/12	176
Four-rule operation (BCD)	16-bit	B+	B+ S1 S2 D	(S1)+(S2)→(D) (BCD)	5/6	178
		B+P	B+P S1 S2 D		11/12	178
		B-	B- S1 S2 D	(S1)−(S2)→(D) (BCD)	5/6	180
		B-P	B-P S1 S2 D		11/12	180
		B*	B* S1 S2 D	(S1) x (S2)→(D+1,D) (BCD)	5/6	182
		B*P	B*P S1 S2 D		11/12	182
		B/	B/ S1 S2 D	(S1) ÷ (S2) → Quotient (D) Remainder (D+1) (BCD)	5/6	184
		B/P	B/P S1 S2 D		11/12	184
+1	16-bit	INC	INC D	(D) + 1 → (D)	2	186
		INCP	INCP D		8	186
	32-bit	DINC	DINC D	(D+1, D) + 1 → (D + 1, D)	2	188
		DINCP	DINCP D		8	188
–1	16-bit	DEC	DEC D	(D) – 1 → (D)	2	190
		DECP	DECP D		8	190
	32-bit	DDEC	DDEC D	(D + 1, D) – 1 → (D + 1, D)	2	192
		DDECP	DDECP D		8	192

Class	Process unit	Command sign	Symbol	Process details	No. of steps	Page
Comple-ment of 2	16-bit	NEG	[NEG D]	· (D) → (D) BIN data	2	194
		NEGP	[NEGP D]		8	194
		DNEG	[DNEG D]	· (D+1, D) → (D + 1, D) BIN data	2	196
		DNEGP	[DNEGP D]		8	196

(3) BCD ↔ BIN conversion commands

Class	Process unit	Command sign	Symbol	Process details	No. of step	Page
BCD	16-bit	BCD	[BCD S D]	BCD conversion (S) → (D) BIN (0 to 9999)	3	198
		BCDP	[BCDP S D]		9	198
	32-bit	DBCD	[DBCD S D]	BCD conversion (S+1,S) → (D+1,D) BIN (0 to 99999999)	4/5	200
		DBCDP	[DBCDP S D]		10/11	200
BIN	16-bit	BIN	[BIN S D]	BIN conversion (S) → (D) BIN (0 to 9999)	3	202
		BINP	[BINP S D]		9	202
	32-bit	DBIN	[DBIN S D]	BIN conversion (S+1,S1) → (D+1,D) BIN (0 to 99999999)	4/5	204
		DBINP	[DBINP S D]		10/11	204

(4) Data transmission commands

Class	Process unit	Command sign	Symbol	Process details	No. of step	Page
Trans-mission	16-bit	MOV	MOV S D	· (S) → (D)	3	206
		MOVP	MOVP S D		9	206
	32-bit	DMOV	DMOV S D	· (S+1,S) → (D+1,D)	3/4	208
		DMOVP	DMOVP S D		9/10	208
	16-bit	CML	CML S D	· $\overline{\text{(S)}}$ → (D)	3	210
		CMLP	CMLP S D		9	210
	32-bit	DCML	DCML S D	· $\overline{\text{(S+1,S)}}$ → (D+1,D)	3/4	212
		DCMLP	DCMLP S D		9/10	212
Conversion	16-bit	XCH	XCH D1 D2	· (D1) ↔ (D2)	4	214
		XCHP	XCHP D1 D2		10	214
	32-bit	DXCH	DXCH D1 D2	· (D1+1,D1) ↔ (D2+1,D2)	4	216
		DXCHP	DXCHP D1 D		10	216
Batch trans-mission	16-bit	BMOV	BMOV S D n		5	218
		BMOVP	BMOVP S D n		11	218
Batch trans-mission of same data	16-bit	FMOV	FMOV S D n		5	220
		FMOVP	FMOVP S D n		11	220

(5) Program branch commands

Class	Process unit	Command sign	Symbol	Process details	No. of step	Page
Jump	—	CJ	[CJ P**]	Jump to Pn after input conditions are met	2	222
		JMP	[JMP P**]	Jump to Pn unconditionally	2	222
Program end	—	FEND	[FEND]	End process during sequence program	1	224
Subroutine call	—	CALL	[CALL P**]	Execute P** sub-routine program after input conditions are met	2	226
	—	CALLP	[CALLP P**]		8	226
Return	—	RET	[RET]	Return to main program from subroutine program	1	226
	—	FOR	[FOR n]	Execute n times between FOR to NEXT	5	230
	—	NEXT	[NEXT]		4	230
	—	BREAK	[BREAK D P**]	Forcibly end the execution of FOR to NEXT cycle and jump to pointer Pn	5	232
	—	BREAKP	[BREAKP D P**]		11	232

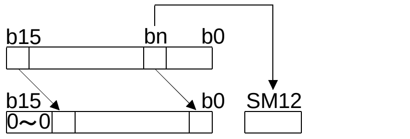
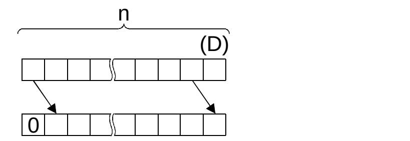
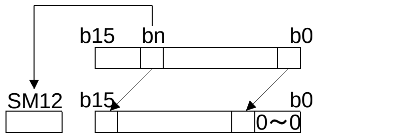
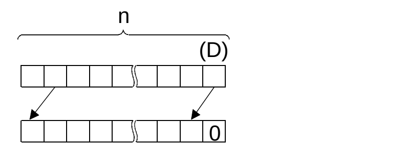
(6) Logical operation commands

Class	Process unit	Command sign	Symbol	Process details	No. of step	Page
Logical AND	16-bit	WAND	WAND S D	(D) ^ (S) → (D)	3	234
		WANDP	WANDP S D		9	234
		WAND	WAND S1 S2 D	(S1) ^ (S2) → (D)	4	234
		WANDP	WANDP S1 S2 D		10	234
	32-bit	DAND	DAND S D	(D + 1, D) ^ (S + 1, S) → (D + 1, D)	3/4	236
		DANDP	DANDP S D		9/10	236
		DAND	DAND S1 S2 D	(S1 + 1, S1) ^ (S2 + 1, S2) → (D + 1, D)	4/5	236
		DANDP	DANDP S1 S2 D		10/11	236
Logical OR	16-bit	WOR	WOR S D	(D) V (S) → (D)	3	238
		WORP	WORP S D		9	238
		WOR	WOR S1 S2 D	(S1) V (S2) → (D)	4	238
		WORP	WORP S1 S2 D		10	238
	32-bit	DOR	DOR S D	(D + 1, D) V (S + 1, S) → (D + 1, D)	3/4	240
		DORP	DORP S D		9/10	240
		DOR	DOR S1 S2 D	(S1 + 1, S1) V (S2 + 1, S2) → (D + 1, D)	4/5	240
		DORP	DORP S1 S2 D		10/11	240

Class	Process unit	Command sign	Symbol	Process details	No. of step	Page
Exclusive OR	16-bit	WXOR	[WXOR S D]	· $(D) \veebar (S) \rightarrow (D)$	3	242
		WXORP	[WXORP S D]		9	242
		WXOR	[WXOR S1 S2 D]	· $(S1) \veebar (S2) \rightarrow (D)$	4	242
		WXORP	[WXORP S1 S2 D]		10	242
	32-bit	DXOR	[DXOR S D]	· $(D+1,D) \veebar (S+1,S) \rightarrow (D+1,D)$	3/4	244
		DXORP	[DXORP S D]		9/10	244
		DXOR	[DXOR S1 S2 D]	· $(S1+1,S1) \veebar (S2+1,S2) \rightarrow (D+1,D)$	4/5	244
		DXORP	[DXORP S1 S2 D]		10/11	244
Non exclusive OR	16-bit	WXNR	[WNXR S D]	· $\overline{(D) \veebar (S)} \rightarrow (D)$	3	246
		WXNRP	[WNXRP S D]		9	246
		WXNR	[WNXR S1 S2 D]	· $\overline{(S1) \veebar (S2)} \rightarrow (D)$	4	246
		WXNRP	[WNXRP S1 S2 D]		10	246
	32-bit	DXNR	[DNXR S D]	· $\overline{(D+1,D) \veebar (S+1,S)} \rightarrow (D+1,D)$	3/4	248
		DXNRP	[DNXRP S D]		9/10	248
		DXNR	[DNXR S1 S2 D]	· $\overline{(S1+1,S1) \veebar (S2+1,S2)} \rightarrow (D+1,D)$	4/5	248
		DXNRP	[DNXRP S1 S2 D]		10/11	248

(7) Rotation commands

Class	Process unit	Command sign	Symbol	Process details	No. of step	Page
Right rotation	16-bit	ROR	[ROR D n]	b15 (D) b0 SM12 Rotate n bits right.	3	250
		RORP	[RORP D n]		9	250
		RCR	[RCR D n]	b15 (D) b0 SM12 Rotate n bits right.	3	252
		RCRP	[RCRP D n]		9	252
	32-bit	DROR	[DROR D n]	(D+1) b31～b16 (D) b15～b0 SM12 Rotate n bits right.	3	254
		DRORP	[DRORP D n]		9	254
		DRCR	[DRCR D n]	(D+1) b31～b16 (D) b15～b0 SM12 Rotate n bits right.	3	256
		DRCRP	[DRCRP D n]		9	256
Left rotation	16-bit	ROL	[ROL D n]	SM12 b15 (D) b0 Rotate n bits left.	3	258
		ROLP	[ROLP D n]		9	258
		RCL	[RCL D n]	SM12 b15 (D) b0 Rotate n bits left.	3	260
		RCLP	[RCLP D n]		9	260
	32-bit	DROL	[DROL D n]	SM12 (D+1) b31～b16 (D) b15～b0 Rotate n bits left.	3	262
		DROLP	[DROLP D n]		9	262
		DRCL	[DRCL D n]	SM12 (D+1) b31～b16 (D) b15～b0 Rotate n bits left.	3	264
		DRCLP	[DRCLP D n]		9	264

Class	Process unit	Command sign	Symbol	Process details	No. of step	Page
Right shift	16-bit	SFR	[SFR D n]		3	266
		SFRP	[SFRP D n]		9	266
	Device unit	DSFR	[DSFR D n]		4	268
		DSFRP	[DSFRP D n]		10	268
Left shift	16-bit	SFL	[SFL D n]		3	270
		SFLP	[SFLP D n]		9	270
	Device unit	DSFL	[DSFL D n]		4	272
		DSFLP	[DSFLP D n]		10	272

(8) Data processing commands

Class	Process unit	Command sign	Symbol	Process details	No. of step	Page
Search	16-bit	SER	SER S1 S2 D	(S1) (S2) n → (D) :Match No. (D+1) :Number of match data pieces	5	274
		SERP	SERP S1 S2 D		11	274
		DSER	DSER S1 S2 D	32 bit (S1) (S2) n → (D) :Match No. (D+1) :Number of match data pieces	5	276
		DSERP	DSERP S1 S2 D		11	276
Number of bits set to 1	16-bit	SUM	SUM S D	(S) b15 b0 → Number of bits set to 1.	4	278
		SUMP	SUMP S D		10	278
		DSUM	DSUM S D	(S+1) (S) → Number of bits set to 1.	4	280
		DSUMP	DSUMP S D		10	280
Decode	2^n-bit	DECO	DECO S D n	8→256 decode (S) n Decode → (D) 2^n bits	5	282
		DECOP	DECOP S D n		11	282
		S.BDECO	S.BDECO S D n	BCD 2 digits → 256 decode (S) 8 bits BCD BIN 8 → 256 bits	4	284
	16-bit	SEG	SEG S D	(S) b3～b0 7SEG → (D)	3	286
		SEGP	SEGP S D		9	266
Encode	2^n-bit	ENCO	ENCO S D n	256→8 encode (S) 2 bits Encode → (D) n	5	288
		ENCOP	ENCOP S D n		11	288
Average value	16-bit	S.AVE	S.AVE S D n	16-bit data average value $\frac{1}{n}\sum_{i=1}^{n}(S+i) \rightarrow (D)$	5	290

(9) Other function commands

Class	Process unit	Command sign	Symbol	Process details	No. of step	Page
Carry flag set	—	S.STC	[S.STC]	Carry flag contact (SM12) is turned on.	1	292
Carry flag reset	—	S.CLC	[S.CLC]	Carry flag contact (SM12) is turned off.	1	292
BIT	1-bit	LDBIT	[BIT S1 n]	Bit test (A contact operation start handling) (Note 1)	2	294
		ANDBIT	[BIT S1 n]	Bit test (A contact series connection handling) (Note 1)	2	294
		ORBIT	[BIT S1 n]	Bit test (A contact parallel connection handling) (Note 1)	2	294
		LDBII	[BII S1 n]	Bit test (B contact operation start handling) (Note 1)	2	296
		ANDBII	[BII S1 n]	Bit test (B contact series connection handling) (Note 1)	2	296
		ORBII	[BII S1 n]	Bit test (B contact parallel connection handling) (Note 1)	2	296
	1-bit	BSET	[BSET D n]	(D) b15 bn b0 → 1	2	298
		BSETP	[BSETP D n]		8	298
		BRST	[BRST D n]	(D) b15 bn b0 → 0	2	300
		BRSTP	[BRSTP D n]		8	300

(Note 1) LD<=, AND<=, OR<=, LD<>, AND<> and OR<> are used for input.

6.1.3 Exclusive Commands 1

Class	Process unit	Command sign	Symbol	Process details	No. of step	Page
Intelligent function	16-bit	FROM	FROM n1 n2 Dn n3	Reads the data of the buffer memory of MELSEC intelligent function unit out to the C6/C64 PLC.	5	304
		TO	TO n1 n2 Dn n3	Writes the data of C6/C64 PLC register into the buffer memory of MELSEC intelligent function unit.	5	304
Transient command	Bit	READ	G.READ Un S1 S2 D1 D2	Reads the word device data of the designated station out to the local station.	6	308
			GP.READ Un S1 S2 D1 D2		12	308
		WRITE	G.WRITE Un S1 S2 D1 D2	Writes the data of the local station to the word device data of the designated station.	6	308
			GP.WRITE Un S1 S2 D1 D2		12	308
	Bit	SREAD	G.SREAD Un S1 S2 D1 D2 D3	Reads the word device data of the designated station out to the local station. (With confirmation of the target station completion.)	7	308
			GP.SREAD Un S1 S2 D1 D2 D3		13	308
		SWRITE	G.SWRITE Un S1 S2 D1 D2 D3	Writes the data of the local station to the word device data of the designated station. (With confirmation of the target station completion.)	7	308
			GP.SWRITE Un S1 S2 D1 D2 D3		13	308
	Bit	RIRD	G.RIRD Un S1 D1 D2	Read the device data from the designated station CPU. (Valid on QnA series Ver.J and higher.)	5	318
			GP.RIRD Un S1 D1 D2		11	318
		RIWT	G.RIWT Un S1 D1 D2	Writes the device data to the designated station CPU. (Valid on QnA series Ver.J and higher.)	5	318
			GP.RIWT Un S1 D1 D2		11	318
Transient trans-mission	Bit	OPEN	ZP.OPEN "Un" S1 S2 D1	Open a connection with the client device to communicate with.	12	322
		CLOSE	ZP.CLOSE "Un" S1 S2 D1	Close the connection with the client device that is communicating with.	12	326
	Bit	BUFSND	ZP.BUFSND "Un" S1 S2 S3 D1	Sends data to the client device. (Fixed-buffer communication)	13	328
		BUFRCV	ZP.BUFRCV "Un" S1 S2 D1 D2	Reads the data received from the client device. (Fixed-buffer communication)	13	332

6.1.4 Exclusive Commands 2

Class	Process unit	Command sign	Symbol	Process details	No. of step	Page
ATC	—	S.ATC	S.ATC Kn Rn Rm Mn	K1: Tool number search	6	342
				K2: Tool number AND search		343
				K3: Tool change		344
				K4: Random position tool change		345
				K5: Forward rotation of pointer		346
				K6: Reverse rotation of pointer		346
				K7: Normal rotation of tool table		347
				K8: Reverse rotation of tool table		347
				K9: Tool data read		348
				K10: Tool data write		349
				K11: Automatic write of tool data		350
ROT	—	S.ROT	S.ROT Kn Rn Rm Mn	K1: Rotary body index	6	355
				K3: Ring counter		359
TSRH	—	S.TSRH	S.TSRH Rn Rm Mn	Spare tool selection in tool life management	5	360
DDB	—	S.DDBA (Asynchro-nous)	S.DDBA Rn/Dn	Data designated after Rn/Dn is read/written.	3	371
		S.DDBS (Synchro-nous)	S.DDBS Rn/Dn	Data designated after Rn/Dn is read/written.	3	376

6.2 Command Formats

6.2.1 How to Read the Command Table

The basic command and function command explanations are as follow.

Example of D+ command

The command signal is indicated.

○D＋, D＋P……BIN 32-bit addition

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Con-stant		Poin-ter			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3/4
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	4/5
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						

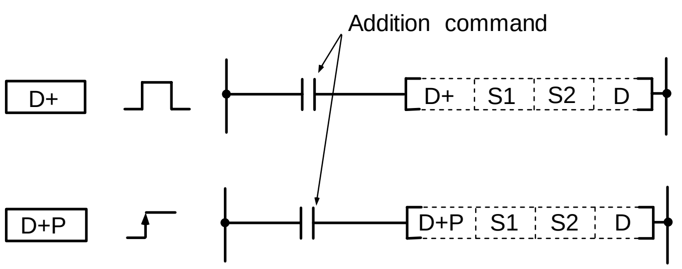
Expressed with T.
Same applies for C.

A circle is indicated if digit designation of the bit device is possible.

The devices that can be used with the D+ command are circled.

The commands that can use an index (Z0 to Z13) are circled.

The No. of steps of the D+ command is indicated.
This is a No. of steps required for the store in the controller.
In programming with MELSEC PLC development tool (GX Developer), the displayed No. of steps may be different from this No. of steps.
Description such as “4/5” indicates that the No. of steps is different depending on the designation device or the type of the command. For the 32-bit command, two steps are required for the constant. In the example for the D+ command, if S2 is the word device, the No. of steps will be 4 steps, and if S2 is the constant, the No. of steps will be 5 steps.
The command executed only at the leading edge (***P command) uses 6 steps bigger than the command executed when ON(***command).



Setting data

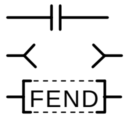
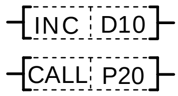
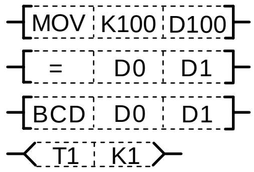
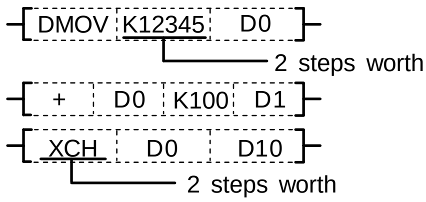
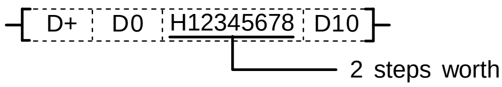
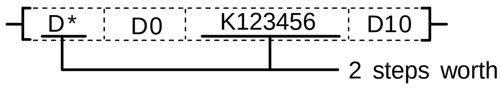
S1	Addition data or head No. of device where addition data is stored.
S2	Addition data or head No. of device where addition data is stored.
D	Head No. of device to store addition results.

The D+ command circuit display format is indicated.

The functions, execution conditions and program examples of each command are explained on the following pages.

6.2.2 No. of Steps

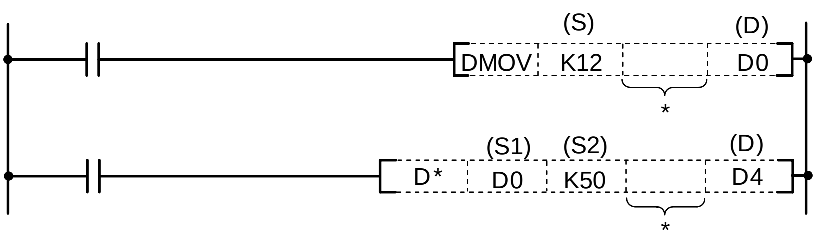
The basic No. of steps in the sequence command includes step 1 to step 6.
Main examples of each step are shown below.

Basic No. of steps	Command (mnemonic)	Circuit display
Step 1	LD, ANI, ANB, ORB, STC, CLC, FEND, RET, P**	
Step 2	INC, DEC, PLS, PLF, CJ, CALL	
Step 3	MOV, =, BCD, OUT, T	
Step 4	DMOV, +, -, XCH	
Step 5	D+, D-	
Step 6	D*, D/	

As shown above, the command code, source and destination in basic No. of steps for the command are equivalent to one step each. Only a part of commands, the 32-bit command constant K, H, a part of command code B, SB, V, T or C device uses two steps.

(Note1) The command executed only at the leading edge (***P command) uses 6 steps bigger than the command executed when ON (*** command).
For example, the MOV command uses 3 steps, and the MOVP command uses 9 steps.

(Note2) If the constant value in the DMOV or D* command, etc., is small, a display in which there is a space equivalent to one step will occur between the source (S) and destination (D) or between the source (S2) and destination (D). (Section marked with * in diagram.)

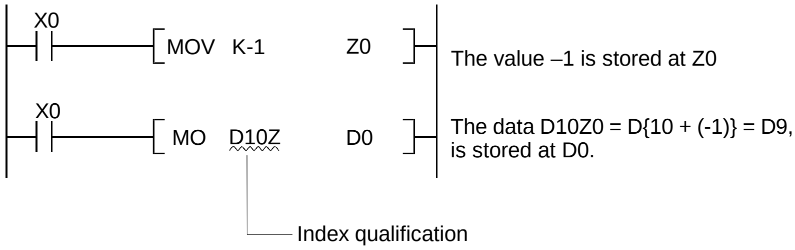


6.2.3 END Command

The END command is automatically created in both a circuit mode and a list mode.

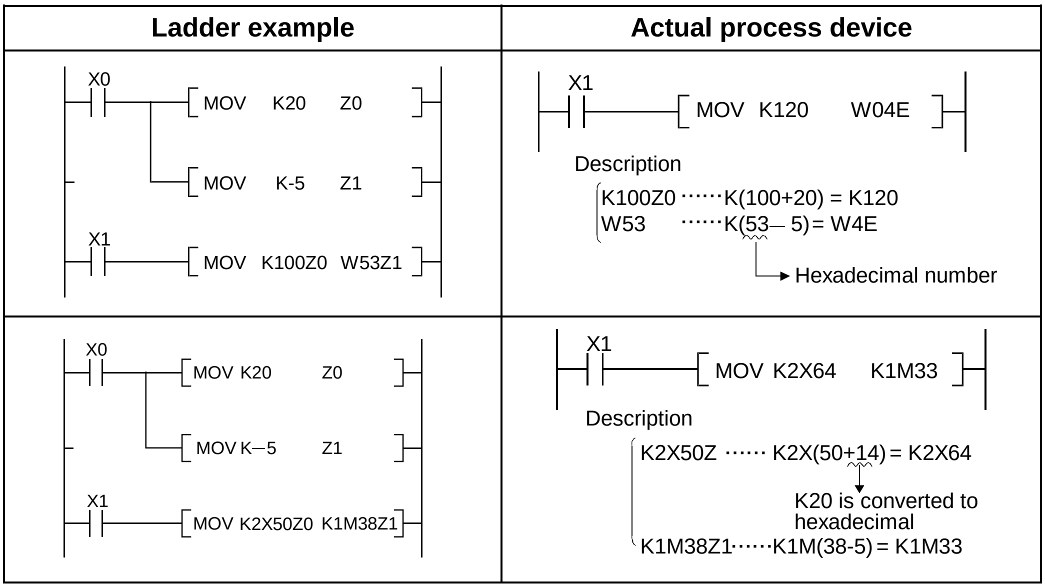
6.2.4 Index Qualification

(1) Index qualification
(a) Index qualification is an indirect setting made by using an index register.
When an index qualification is used in a sequence program, the device to be used will become the device number designated directly plus the contents of the index register.
For example, if D2Z2 has been designated, and the content of Z2 is 3, D(2+3), or D5, will become the designated device.
(b) There are 14 index registers, from Z0 to Z13.
Each index register can be set between –32768 and 32767.
Index qualification is as follows:



A case where index qualification has been performed, and the actual process device, would be as follows:

(Example) When Z0 = 20 and Z1 = -5



Ladder example and actual process device

(2) Devices which can be index-qualified
With the exception of the restrictions noted below, index qualification can be used with devices used with contacts, coils, basic commands, and application commands.

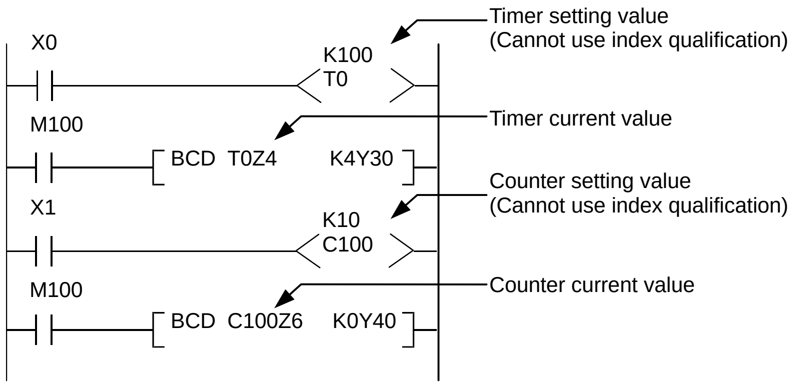
(a) Devices which cannot use index qualification

Device	Meaning
K, H	32-bits constant
P	Pointer used as labels
Z	Index register
T	Timer setting value, coil
C	Counter setting value, coil

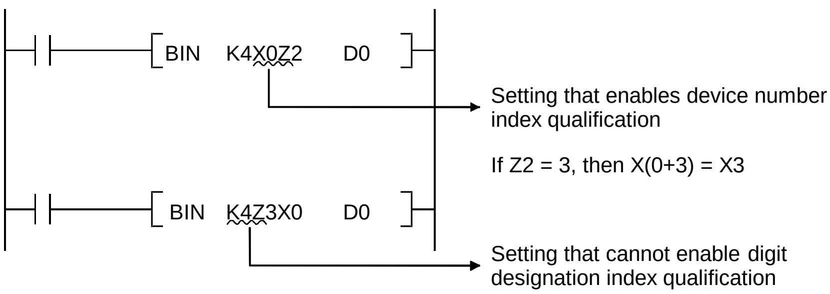
(b) Devices in which the index resistors are restricted

Device	Meaning	Example
T	Z0 or Z1 can use only to timer contact.	T0Z0 K100 T1
C	Z0 or Z1 can use only to counter contact.	C0Z1 K100 C1

For the current value of a timer and a counter, index resistor No. is not restricted.



(c) Others
1) Bit data
Device number can be index-qualified when performing digit designation.
However, index qualification is not possible for digit designation.

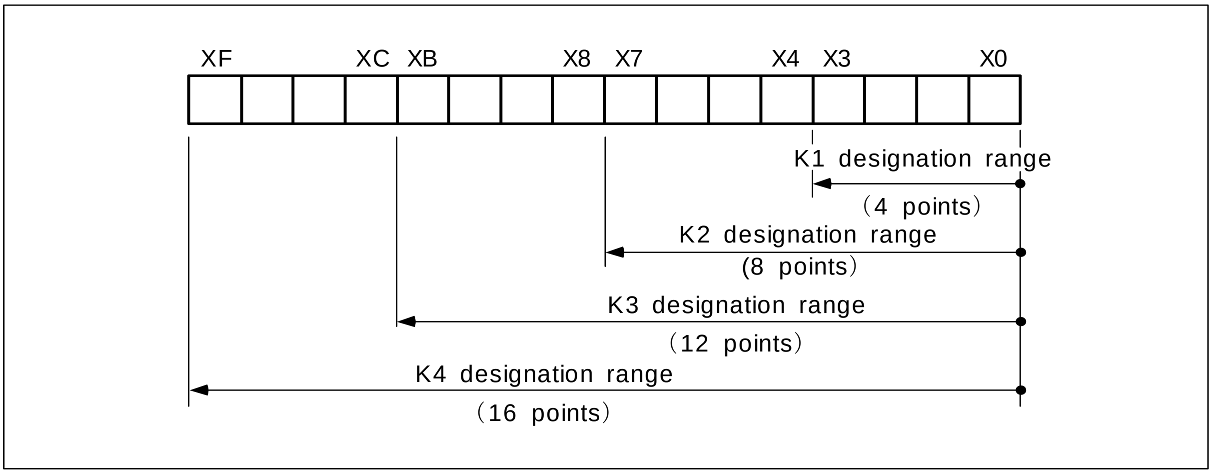


6.2.5 Digit Designation

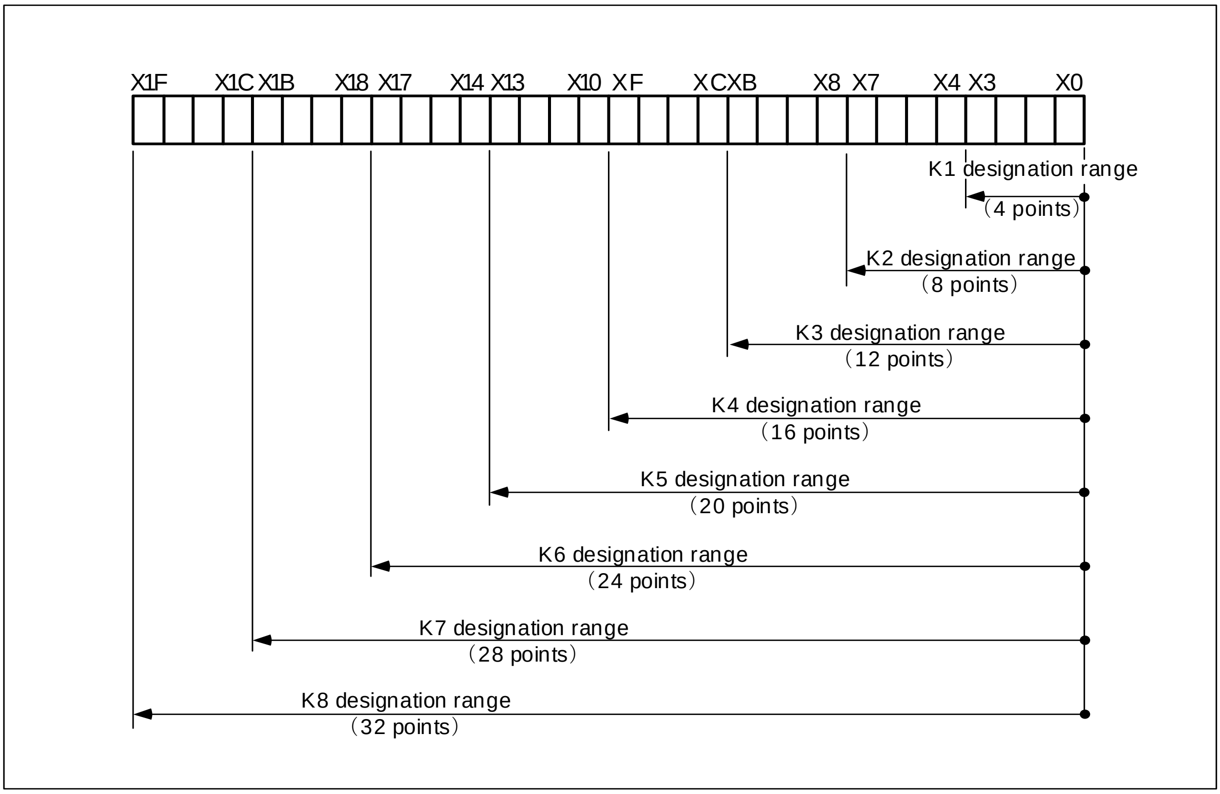
A digit may need to be designated for the bit device (X, Y, M, L, SM, F) when using the function command. How many points of 4-point unit bit devices are to be used with the 16-bit or 32-bit command is selected with this digit designation.
Use device K when designating the digit. The designation range is as shown below. A random bit device can be set for the bit device.

(a) 16-bit command: K1 to 4 (4 to 16 points)

(Example) Setting range with digit designation of X0 to F 16-bit data



(b) 32-bit command: K1 to 8 (4 to 32 points)
(Example) Setting range with digit designation of X0 to 1F 32-bit data.



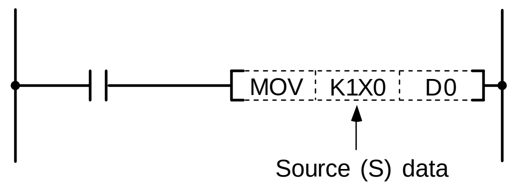
(1) When a digit is designated on the source (S) side, the values that can be handled as source data will be as shown below.

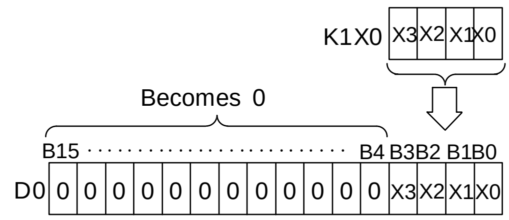
Table of digit designations and values that can be handled

	For 16-bit command	For 32-bit command
K1 (4 points)	0~15	0~15
K2 (8 points)	0~255	0~255
K3 (12 points)	0~4095	0~4095
K4 (16 points)	-32768~32767	0~65535
K5 (20 points)	—	0~1048575
K6 (24 points)	—	0~167772165
K7 (28 points)	—	0~268435455
K8 (32 points)	—	-2147483648~2147483647

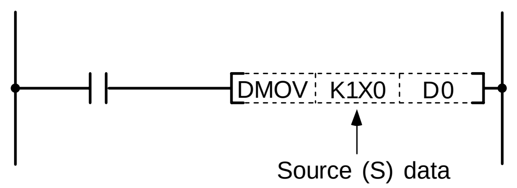
Program example	Process

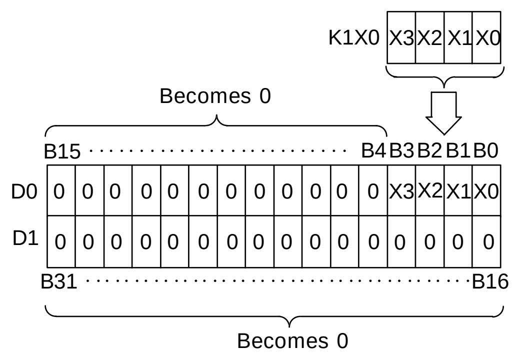
For 16-bit command





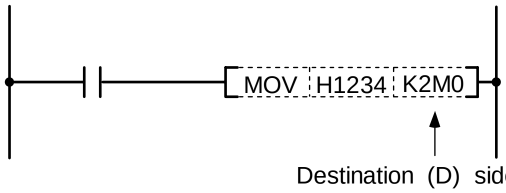
For 32-bit command

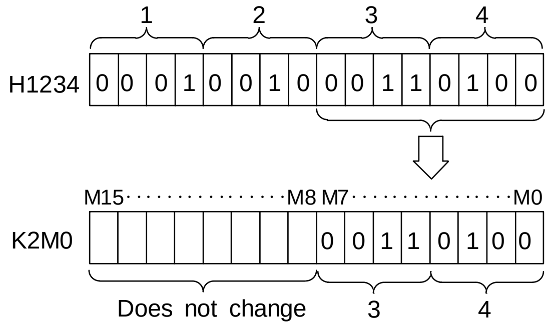




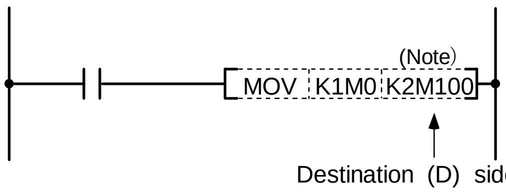
(2) When a digit is designated on the destination (D) side, the No. of points designated by the digit will be the target of the destination side.

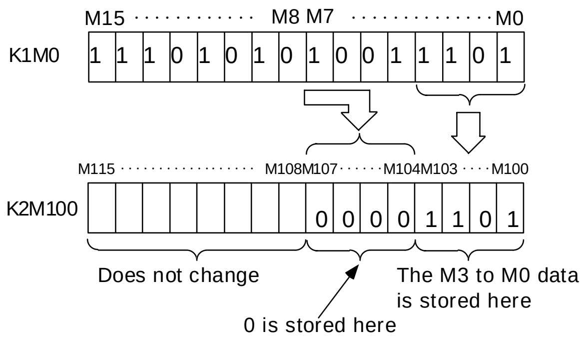
Circuit side	Process
When source data (S) is a value	



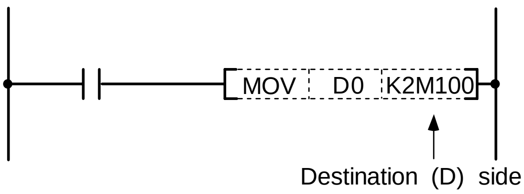


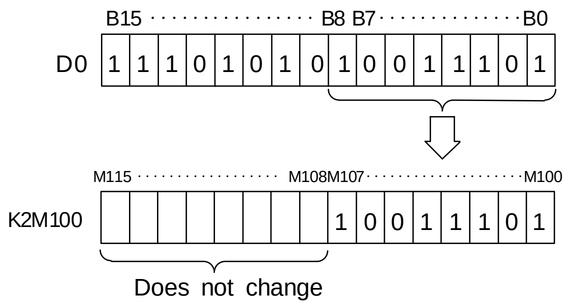
Circuit side	Process
When source (S) data is a bit device	



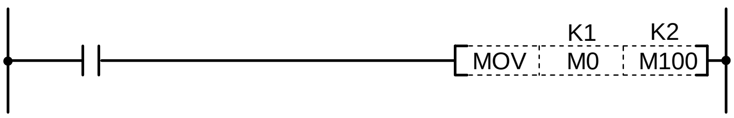


Circuit side	Process
When source (S) data is a word device	





(Note) The display of the circuit having a digit designation will be as follows.



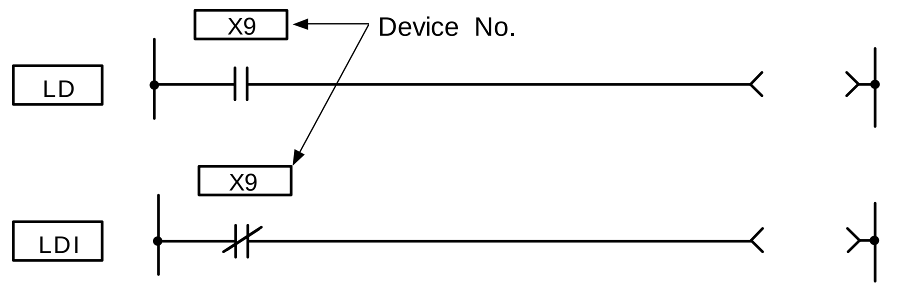
7. Basic Commands

These commands are the basis for the sequence programs. The sequence program cannot be created without these commands.
The circuit can be created (programmed) with the same image as creating a circuit by combining the actual relay A contacts and B contacts as done conventionally.

○ LD, LDI ... Operation start

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
	○	○	○	○	○	○	○	○	○														○	1/2

(Note) In the modification of index, 2 steps are used for B or SB device.



Function

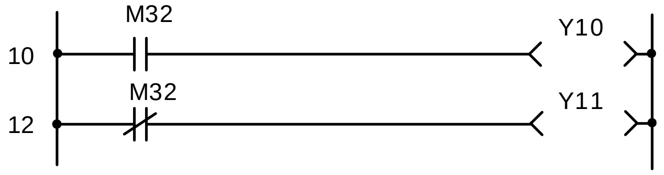
LD is the A contact operation start command and LDI is the B contact operation start command. The ON/OFF information of the designated device is read in as the operation results.

Execution conditions

This is executed per scan regardless of the device ON/OFF setting.

Program example

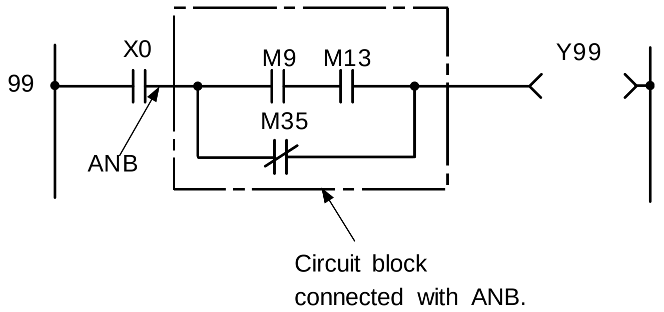
(1) Program used at head of circuit block.



Coding

No. of steps	Com-mand	Device		
10	LD	M32		
11	OUT	Y10		
12	LDI	M32		
13	OUT	Y11		
14				

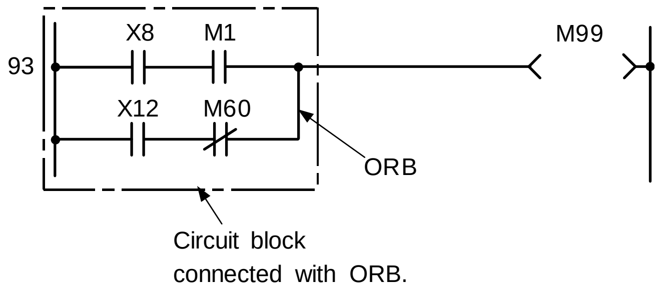
(2) Program used at head of circuit block connected with ANB.



Coding

No. of steps	Com-mand	Device		
99	LD	X0		
100	LD	M9		
101	AND	M13		
102	ORI	M35		
103	ANB			
104	OUT	Y99		
105				

(3) Program used at head of circuit block connected with ORB.



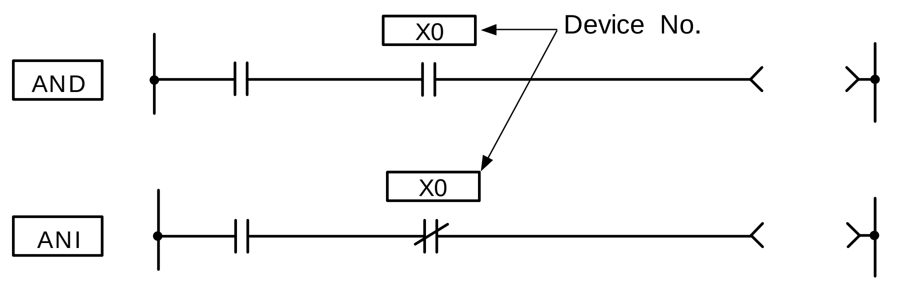
Coding

No. of steps	Com-mand	Device		
93	LD	X8		
94	AND	M1		
95	LD	X12		
96	ANI	M60		
97	ORB			
98	OUT	M99		
99				

○ AND, ANI ... Serial connection of contact

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
	○	○	○	○	○	○	○	○	○														○	1/2

(Note) In the modification of index, 2 steps are used for B or SB device.



Function

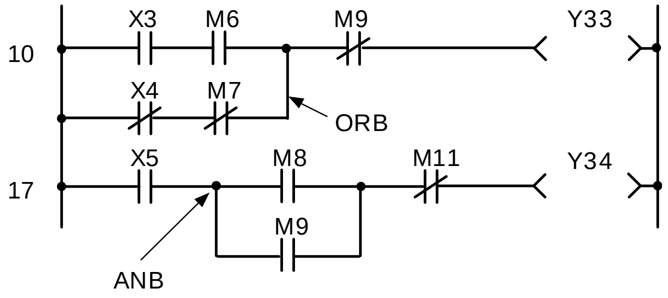
AND is the A contact serial connection command, and ANI is the B contact serial connection command. The ON/OFF information of the designated device is read in, and the AND operation with the operation results up to that point is executed. The result is the operation result.

Execution conditions

This is executed per scan regardless of the operation results before the AND, ANI commands.

Program example

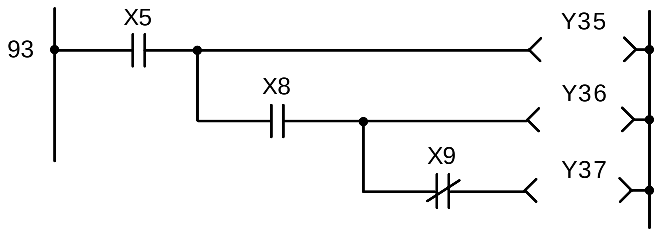
(1) Program used after LD, LDI, AND or ANI, etc.



Coding

No. of steps	Com-mand	Device		
10	LD	X3		
11	AND	M6		
12	LDI	X4		
13	ANI	M7		
14	ORB			
15	ANI	M9		
16	OUT	Y33		
17	LD	X5		
18	LD	M8		
19	OR	M9		
20	ANB			
21	ANI	M11		
22	OUT	Y34		
23				

(2) Program used to connect contact in parallel with coil.



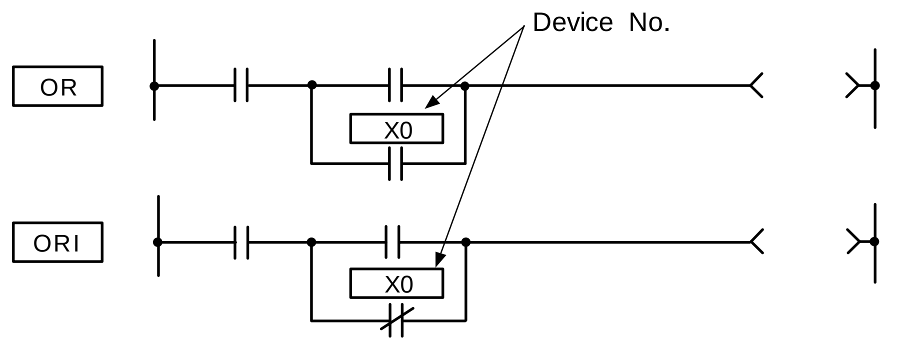
Coding

No. of steps	Com-mand	Device		
93	LD	X5		
94	OUT	Y35		
95	AND	X8		
96	OUT	Y36		
97	ANI	X9		
98	OUT	Y37		
99				

○ OR, ORI ... Parallel connection of one contact

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
	○	○	○	○	○	○	○	○	○														○	1/2

(Note) In the modification of index, 2 steps are used for B or SB device.



Function

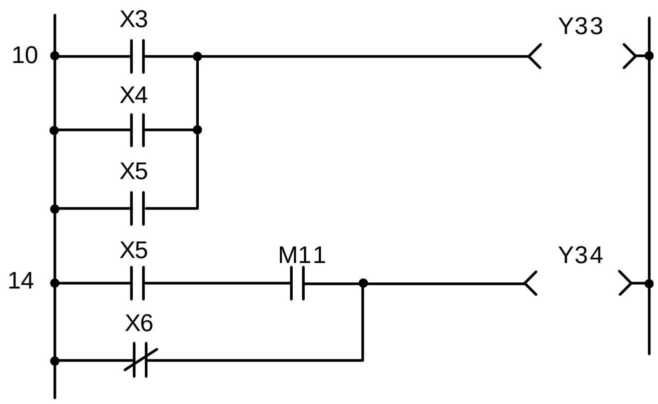
OR is the one A contact parallel connection command, and ORI is the one B contact parallel connection operation command. The ON/OFF information of the designated device is read in, and the OR operation with the operation results up to that point is executed. The result is the operation result.

Execution conditions

This is executed per scan regardless of the operation results before the OR, ORI commands.

Program example

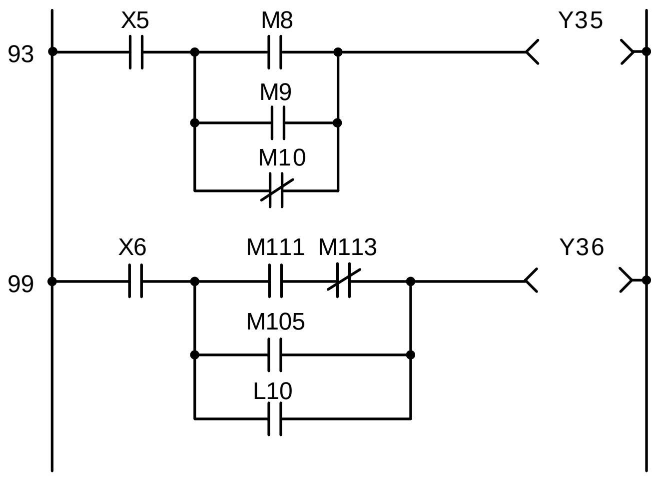
(1) Program used at head of circuit block.



Coding

No. of steps	Com-mand	Device		
10	LD	X3		
11	OR	X4		
12	OR	X5		
13	OUT	Y33		
14	LD	X5		
15	AND	M11		
16	ORI	X6		
17	OUT	Y34		
18				

(2) Program used in circuit.

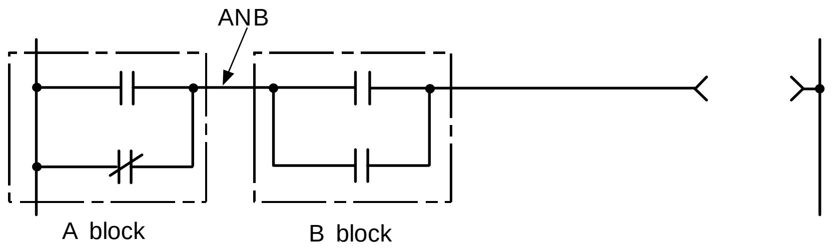


Coding

No. of steps	Com-mand	Device		
93	LD	X5		
94	LD	M8		
95	OR	M9		
96	ORI	M10		
97	ANB			
98	OUT	Y35		
99	LD	X6		
100	LD	M111		
101	ANI	M113		
102	OR	M105		
103	OR	L10		
104	ANB			
105	OUT	Y36		
106				

○ ANB ... Serial connection of circuit block

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
																								1

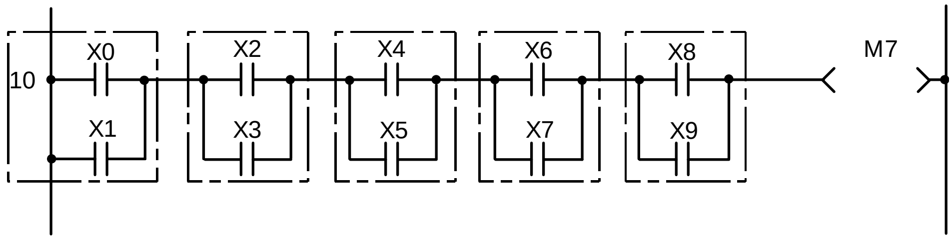


Function

(1) AND operation of the A block and B block is executed, and the operation results are obtained.
(2) The ANB symbol is a connection symbol instead of a contact symbol.
(3) When consecutively writing ANB, a max. of 7 commands (8 blocks) can be written. The PLC cannot execute a correct operation if 8 or more commands are written consecutively.

Program example

Program that serially connects continuous circuit blocks.

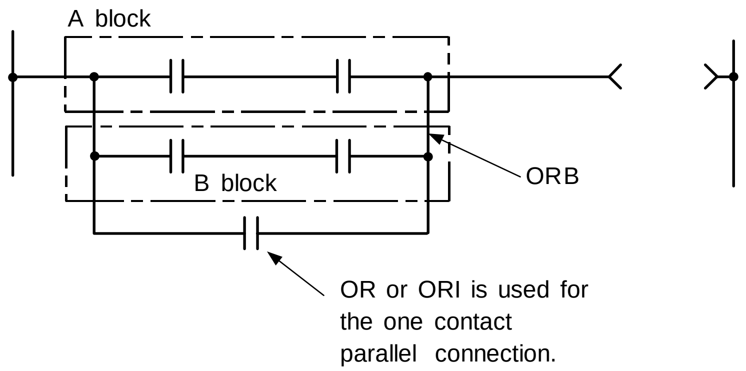


Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	OR	X1		
12	LD	X2		
13	OR	X3		
14	ANB			
15	LD	X4		
16	OR	X5		
17	ANB			
18	LD	X6		
19	OR	X7		
20	ANB			
21	LD	X8		
22	OR	X9		
23	ANB			
24	OUT	M7		
25				

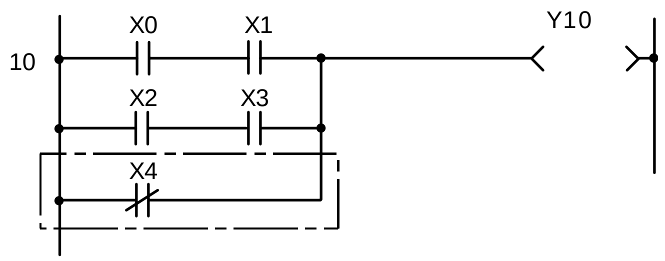
○ ORB ... Parallel connection of blocks

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
																								1



Function

(1) OR operation of the A block and B block is executed, and the operation results are obtained.
(2) ORB connects circuit blocks with two or more contacts in parallel. Use OR or ORI to connect circuit blocks with only one contact in parallel.



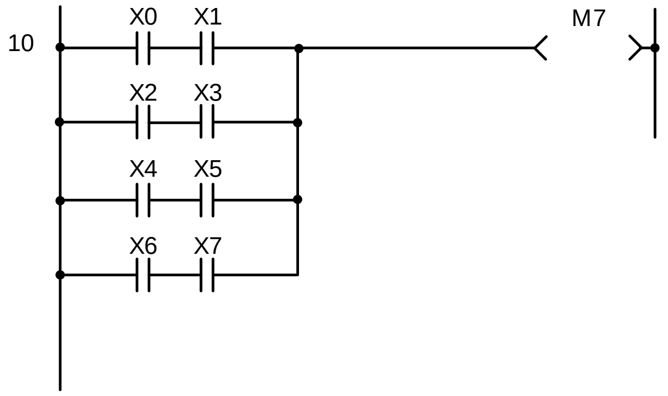
Coding

No. of steps	Command	Device		
10	LD	X0		
11	AND	X1		
12	LD	X2		
13	AND	X3		
14	ORB			
15	ORI	X4		
16	OUT	Y10		
17				

(3) The ORB symbol is a connection symbol instead of a contact symbol.
(4) When consecutively writing ORB, a max. of 7 commands (8 blocks) can be written. The PLC cannot execute a correct operation if 8 or more commands are written consecutively.

Program example

Program that connects continuous circuit blocks in parallel.

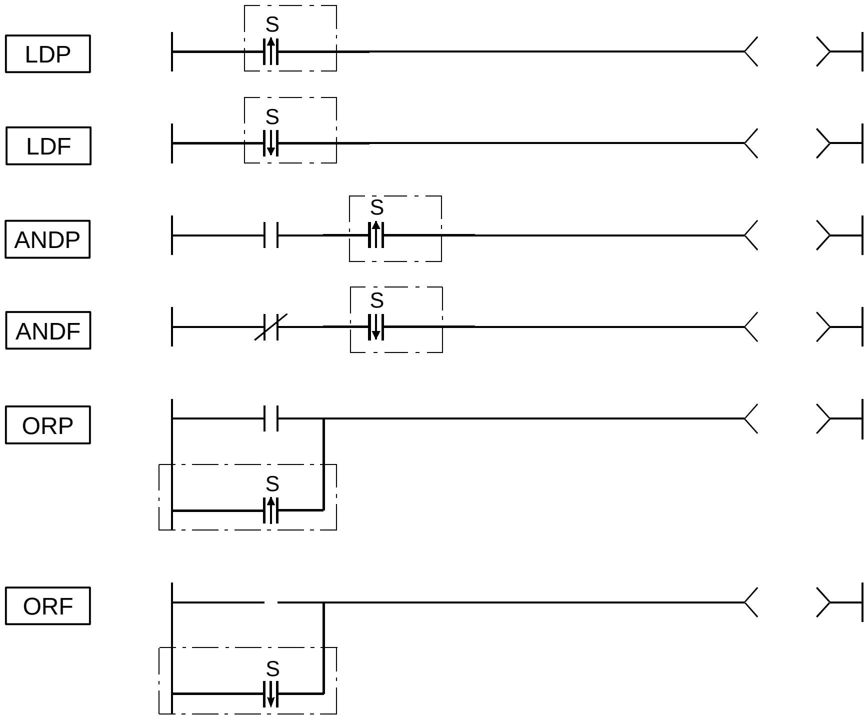


Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	AND	X1		
12	LD	X2		
13	AND	X3		
14	ORB			
15	LD	X4		
16	AND	X5		
17	ORB			
18	LD	X6		
19	AND	X7		
20	ORB			
21	OUT	M7		
22				

○ LDP, LDF, ANDP, ANDF, ORP, ORF ... Pulse operation start, pulse series connection, pulse parallel connection

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
	○	○	○	○	○			○	○														○	6



Setting data

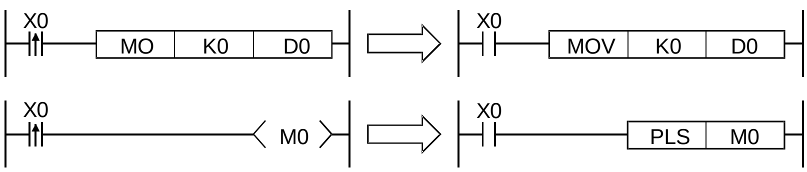
S	Device used as contacts

Function

LDP, LDF

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

A ladder using an LDP command　　A ladder not using an LDP command

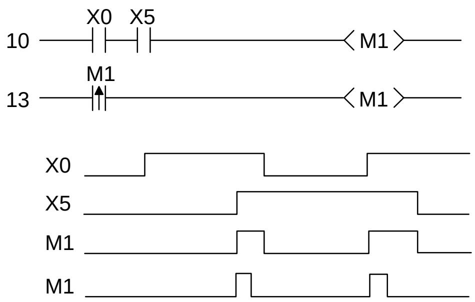


(2) LDF is the trailing edge pulse operation start command, and is ON only at the trailing edge of the designated bit device (when turns from ON to OFF).

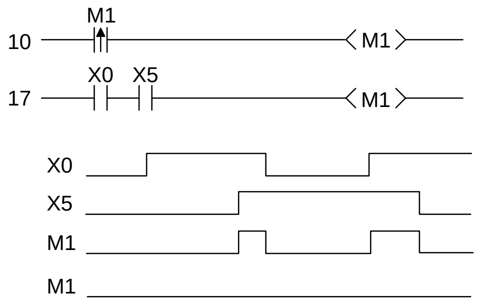
Cautions

(1) Devices used for LDP have to be limited to the ones already output with OUT command, etc. If the devices to be output later are used, the condition will not be achieved.

Correct example



Incorrect example



(2) The commands such as LDP, LDF, etc. cannot use in high-speed program processing. Use the commands such as PLS, MEP, etc.

ANDP, ANDF

(1) ANDP is a leading edge pulse series connection command, and ANDF is a trailing edge pulse series connection command. They perform an AND operation with the operation result up to that point, and take the resulting value as the operation result.
The ON/OFF data used by ANDP and ANDF are indicated in the table below:

Devices designated by ANDP	ANDP state	Devices designated by ANDF	ANDF state
OFF→ON	ON	OFF→ON	OFF
OFF	OFF	OFF	
ON		ON	
ON→OFF		ON→OFF	ON

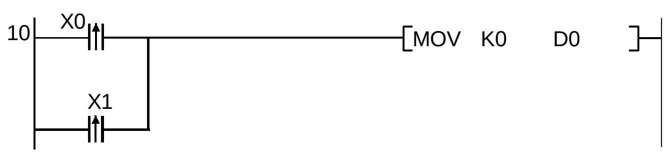
ORP, ORF

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

Devices designated by ORP	ORP state	Devices designated by ORF	ORF state
OFF→ON	ON	OFF→ON	OFF
OFF	OFF	OFF	
ON		ON	
ON→OFF		ON→OFF	ON

Program example

(1) Program to execute the MOV command at leading edge of X0 or X1:

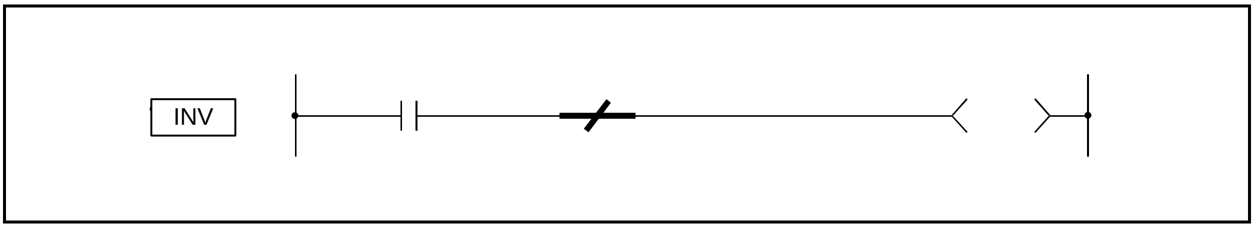


Coding

No. of steps	Command	Device		
10	LDP	X0		
16	ORP	X1		
22	MOV	K0	D0	
25				

○ INV ... Operation results inversion

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
																								4



Function

Inverts the operation result immediately prior to the INV command.

Operation result immediately prior to the INV command.	Operation result following the execution of the INV command.
OFF	ON
ON	OFF

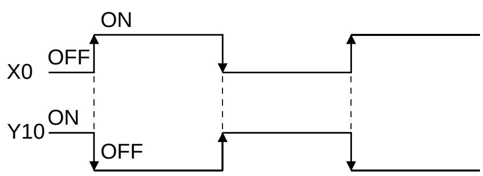
Program example

(1) Program to invert the X0 ON/OFF data, and to output to Y10.

Coding

No. of steps	Command	Device		
10	LD	X0		
11	INV			
15	OUT	Y10		
16				

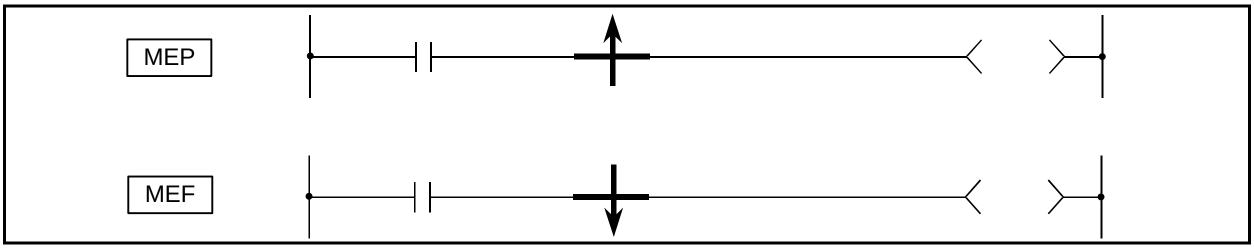
[Timing Chart]



Points
(1) Because the INV command operates with the operation results immediately prior to the INV command, the AND instruction should be used at the same position. The INV command cannot be used at the LD or OR position.

○ MEP, MEF ... Operation results pulse conversion

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
																								4



Function

MEP

(1) If operation results up to MEP command are leading edge (from OFF to ON), turns ON (continuity state).
If operation results up to MEP command are anything other than leading edge, turns OFF (non-continuity state).

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

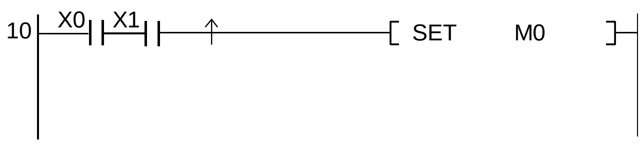
MEF

(1) If operation results up to MEF command are trailing edge (from ON to OFF), turns ON (continuity state).
If operation results up to MEF command are anything other than trailing edge, turns OFF (non-continuity state).

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

Program example

(1) Program to perform pulse conversion on the operation results of X0 and X1:



Coding

No. of steps	Command	Device		
10	LD	X0		
11	AND	X1		
12	MEP			
16	SET	M0		
17				

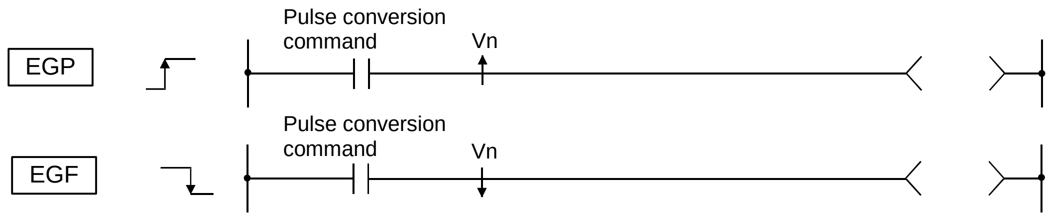
Points

(1) The MEP and MEF commands will occasionally not function properly when pulse conversion is conducted for a contact that has been indexed by a sub-routine program or by the FOR to NEXT loop.
If pulse conversion is to be conducted for a contact that has been indexed by a sub-routine program or by the FOR to NEXT loop, use the EGP/EGF commands.

(2) Because the MEP and MEF commands operate with the operation results immediately prior to the MEP and MEF commands, the AND command should be used at the same position.
The MEP and MEF commands cannot be used at the LD or OR position.

○ EGP, EGF ... Pulse conversion of edge relay operation results

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
										○													○	6



Setting data

Vn	Edge relay number where operation results are stored.

Function

EGP

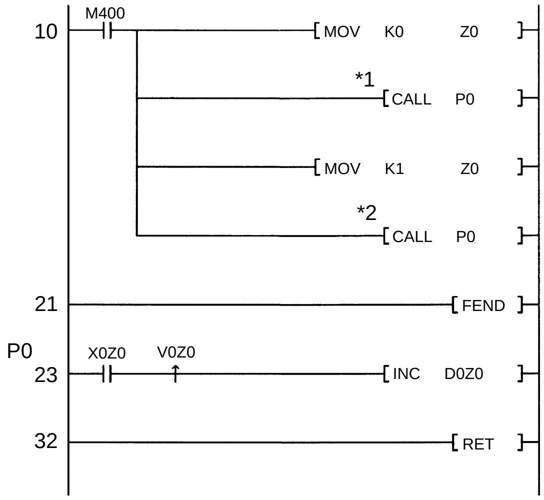
(1) Operation results up to the EGP command are stored in memory by the edge relay (V).
(2) EGP turns ON (continuity state) at the leading edge (OFF to ON) of the operation result up to the EGP command.
If the operation result up to the EGP command is other than a leading edge (i.e., from ON to ON, ON to OFF, or OFF to OFF), it turns OFF (non-continuity state).
(3) The EGP command is used for subroutine programs, and for conducting pulse operations for programs designated by index qualification between FOR to NEXT loop.
(4) The EGP command can be used like an AND command.

EGF

(1) Operation results up to the EGF command are stored in memory by the edge relay (V).
(2) EGF turns ON (continuity state) at the trailing edge (from ON to OFF) of the operation result up to the EGF command.
If the operation result up to the EGF command is other than a trailing edge (i.e., from OFF to ON, ON to ON, or OFF to OFF), it turns OFF (non-continuity state).
(3) The EGF command is used for subroutine programs, and for conducting pulse operations for programs designated by index qualification between FOR to NEXT loop.
(4) The EGF command can be used like an AND command.

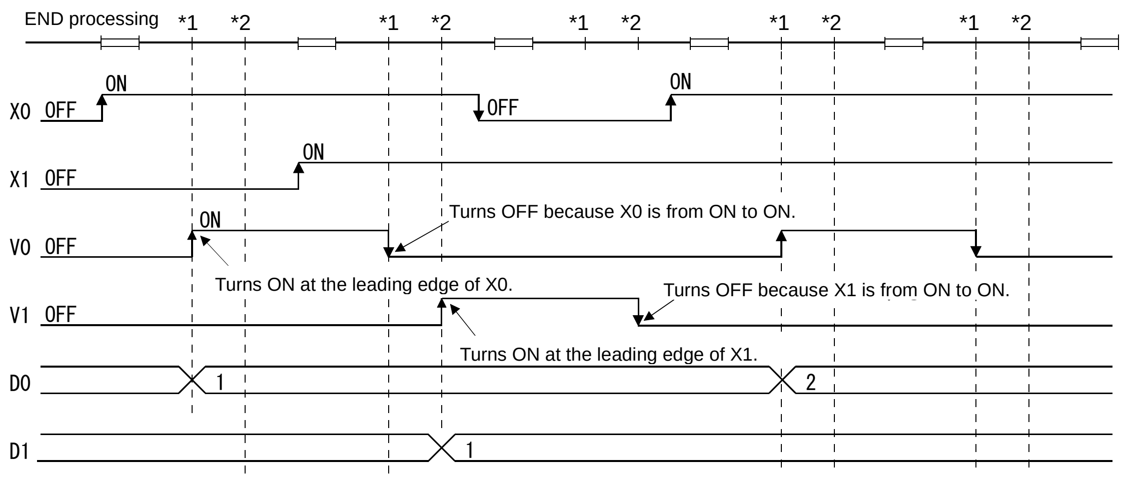
Program example

(1) Program containing a subroutine program using an EGP command.



Coding

No. of steps	Com-mand	Device		
10	LD	M400		
11	MOV	K0	Z0	
14	CALL	P0		
16	MOV	K1	Z0	
19	CALL	P0		
21	FEND			
22		P0		
23	LD	X0Z0		
24	EGP	V0Z0		
30	INC	D0Z0		
32	RET			
33				



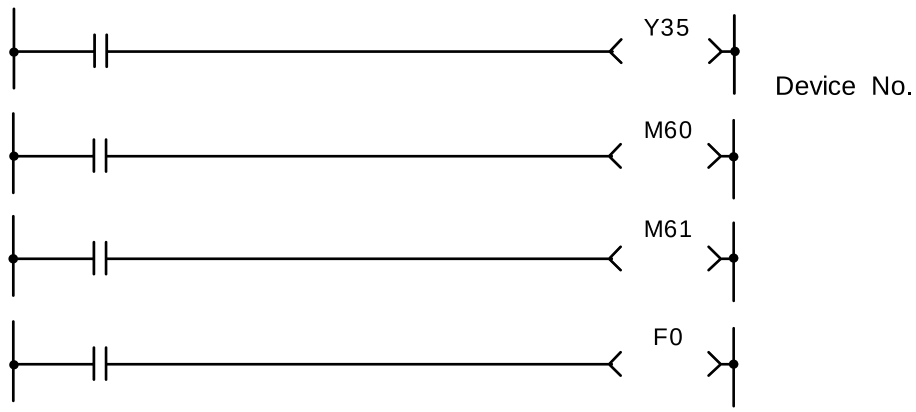
Operation of EGP and EGF commands

Points
(1) Because the EGP and EGF commands operate with the operation results immediately prior to the EGP and EGF commands, the AND instruction should be used at the same position. The EGP and EGF commands cannot be used at the LD or OR position.

○ OUT (Y, M, G, L, F, B, SB, SM) ... Output (Y, M, G, L, F, B, SB, SM)

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
		○	○	○	○	○	○		○														○	1/2

(Note) In the modification of index, 2 steps are used for B or SB device.



Function

The operation results before the OUT command are output to the designated device.

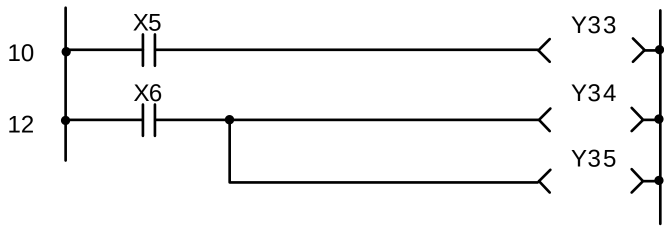
Operation results	OUT command		
	Coil	Contact	
		A contact	B contact
OFF	OFF	Non-continuity	Continuity
ON	ON	Continuity	Non-continuity

Execution condition

This is executed per scan regardless of the operation results before the OUT command.

Program example

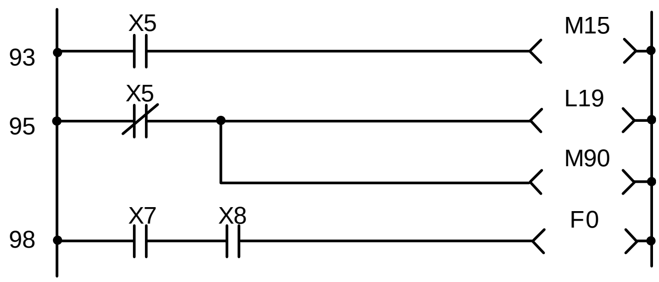
(1) Program output to output unit.



Coding

No. of steps	Com-mand	Device		
10	LD	X5		
11	OUT	Y33		
12	LD	X6		
13	OUT	Y34		
14	OUT	Y35		
15				

(2) Program that turns internal relay or latch relay ON/OFF.

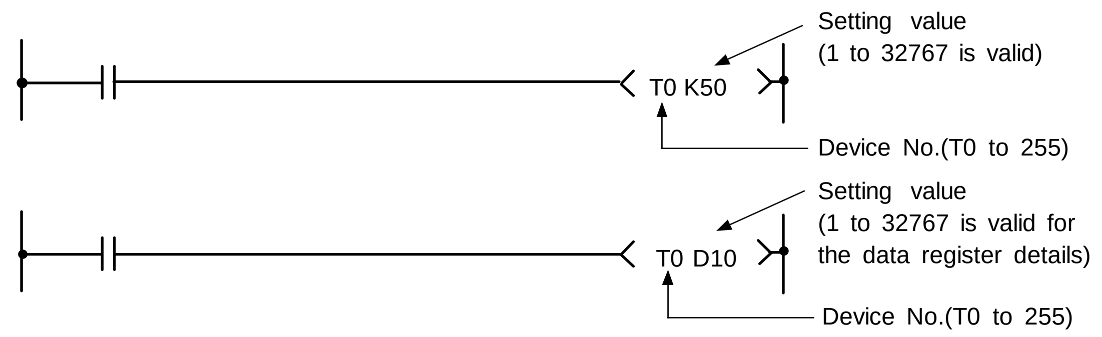


Coding

No. of steps	Com-mand	Device		
93	LD	X5		
94	OUT	M15		
95	LDI	X5		
96	OUT	L19		
97	OUT	M90		
98	LD	X7		
99	AND	X8		
100	OUT	F0		
101				

○ **OUT (T)... Timer output**

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
Device											○												○	3
Setting value													○						○	○				



Function

(1) When the operation results before the OUT command are ON, the timer coil will turn ON and count to the set value. When the time is counted up (count value >= set value), the contacts will change as shown below.

A contact	Continuity
B contact	Non-continuity

(2) If the operation results before the OUT command turn from ON to OFF, the following will occur.

Timer type	Timer coil	Timer current value	Before time up		After time up	
			A contact	B contact	A contact	B contact
100ms timer	OFF	0	Non-continuity	Continuity	Continuity	Non-continuity
10ms timer						
100ms cumulative timer	OFF	Hold current value	Non-continuity	Continuity	Continuity	Non-continuity

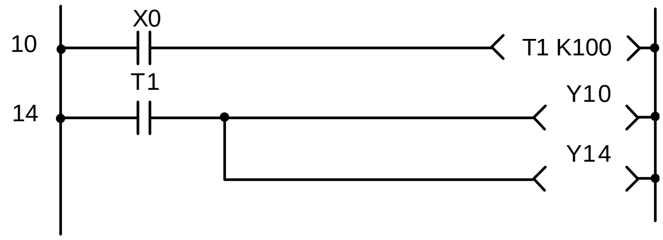
(3) The state of the cumulative timer contact after time up will not change until the RST command is executed.

Execution condition

This is executed per scan regardless of the operation results before the OUT command.

Program example

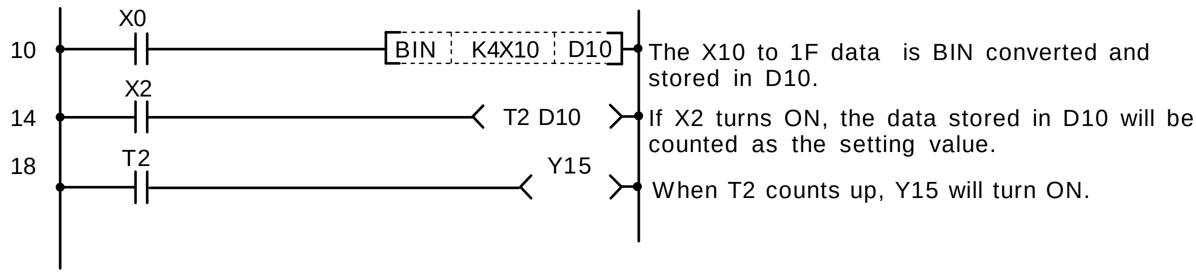
(1) Program to turn ON Y10 and Y14 ten seconds after X0 turns ON.



Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	OUT	T1	K100	
14	LD	T1		
15	OUT	Y10		
16	OUT	Y14		
17				

(2) Program to use X10 to 1F BCD data as timer setting value.

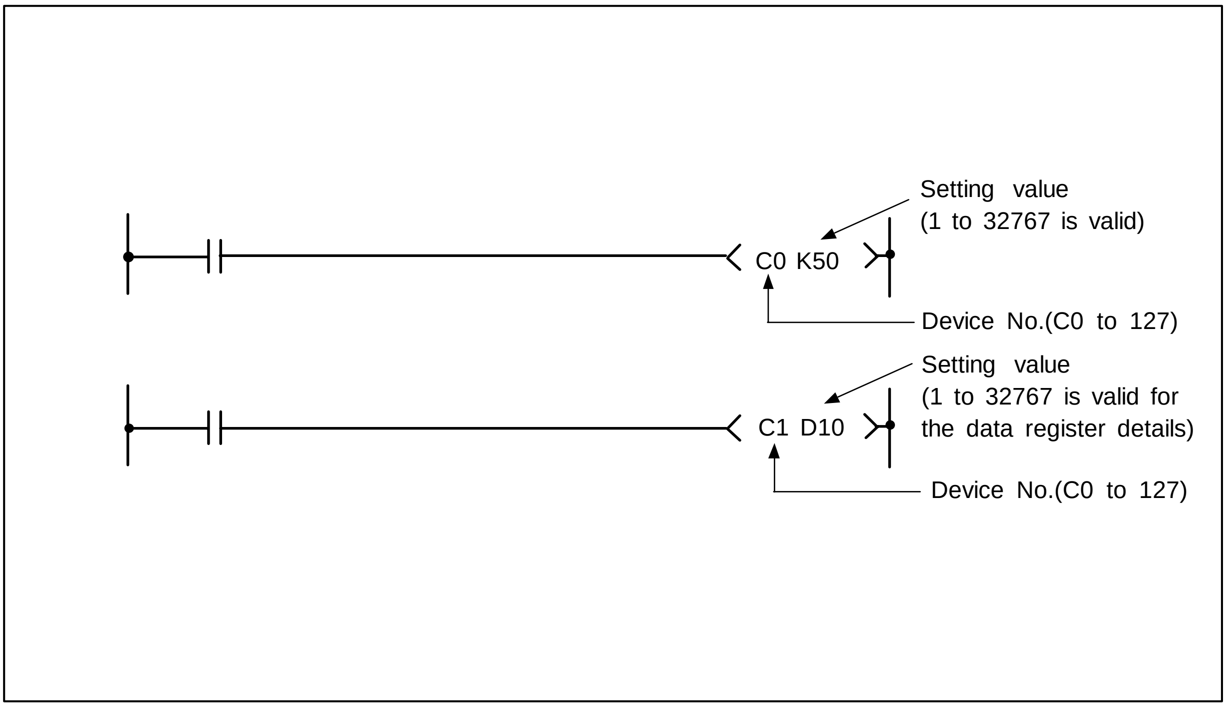


Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	BIN	K4X10	D10	
14	LD	X2		
15	OUT	T2	D10	
18	LD	T2		
19	OUT	Y15		
20				

○ OUT (C) ... Counter output

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
Device												○											○	3
Setting value													○						○	○				



Function

(1) If the operation results before the OUT command change from OFF to ON, the current value (count value) will be incremented by one. When the value is counted up (current value >= setting value), the contacts will change as shown below.

A contact	Continuity
B contact	Non-continuity

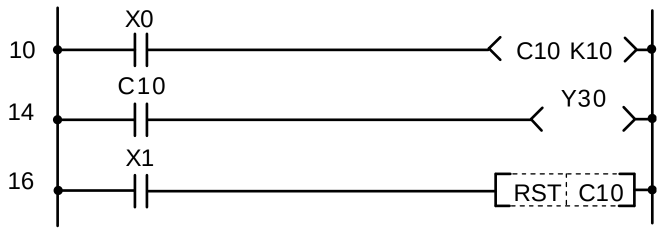
(2) The value will not be counted when the operation results are ON. (A pulse change is not required to input the count.)
(3) If the operation results change from OFF to ON after the current value >= setting value is established, the contact state will remain the same and the current value will not be counted up.

Execution condition

This is executed per scan regardless of the operation results before the OUT command.

Program example

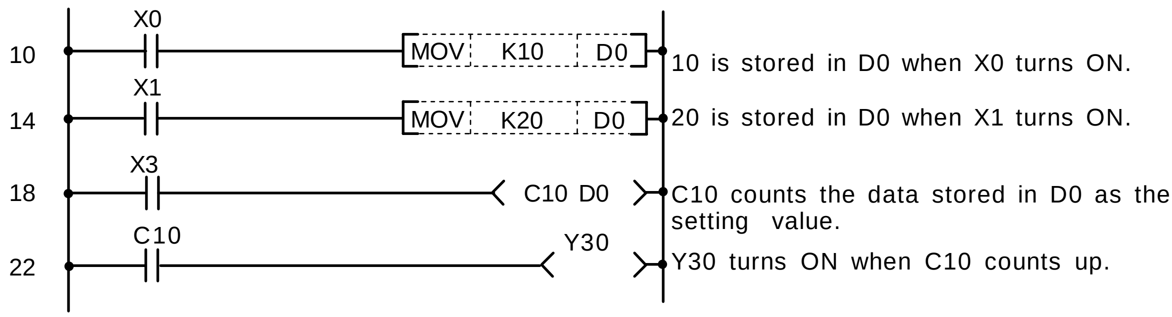
(1) Program to turn Y30 ON when X0 turns ON ten times, and to turn Y30 OFF when X1 turns ON.



Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	OUT	C10	K10	
14	LD	C10		
15	OUT	Y30		
16	LD	X1		
17	RST	C10		
19				

(2) Program to set C10 setting value to 10 when X0 turns ON, and to 20 when X1 turns ON.



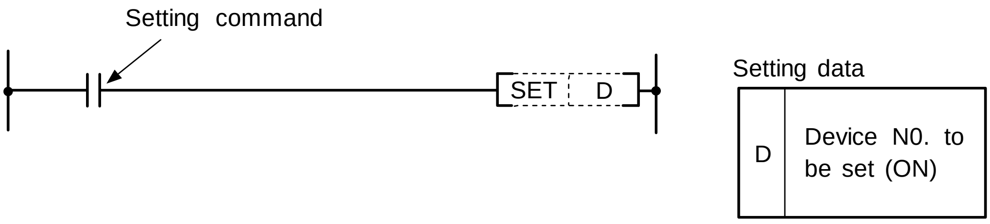
Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	MOV	K10	D0	
14	LD	X1		
15	MOV	K20	D0	
18	LD	X3		
19	OUT	C10	D0	
22	LD	C10		
23	OUT	Y30		
24				

○ SET ... Device setting (ON)

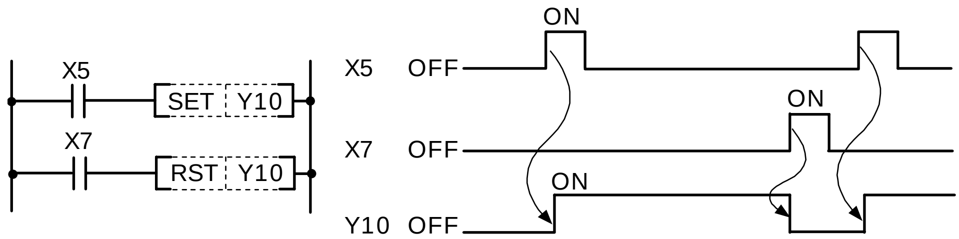
	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D	○	○	○	○	○	○	○		○														○	1/2

(Note) In the modification of index, 2 steps are used for B or SB device.



Function

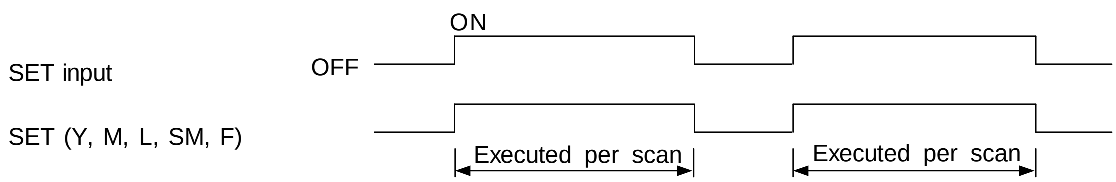
(1) The designated device turns ON when the SET input turns ON.
(2) The device turned ON remains ON even if the SET input turns OFF. The device can be turned OFF with the RST command.



(3) If the SET input is OFF, the state of the device will not change.

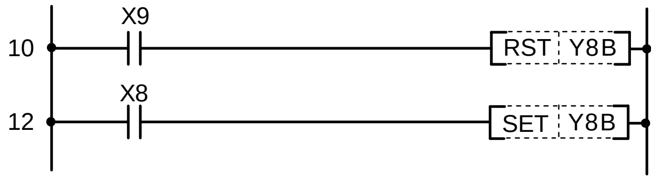
Execution condition

The execution conditions for the SET command are as shown below.



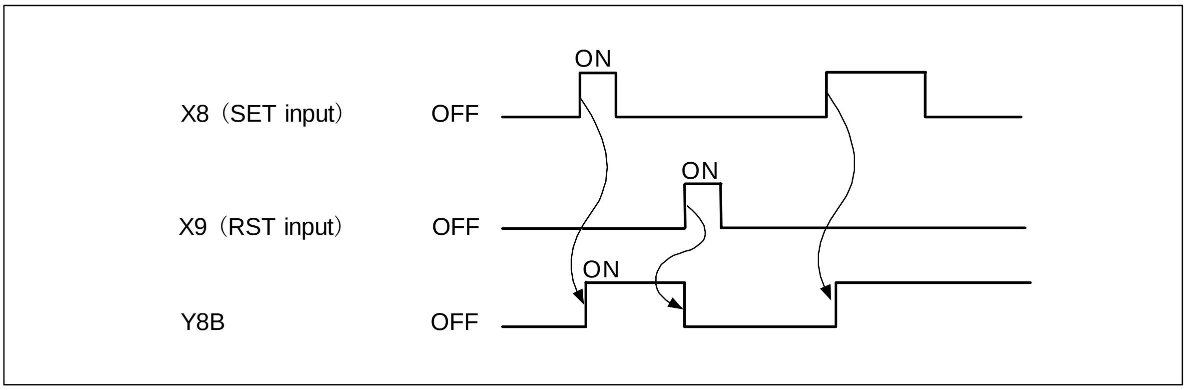
Program example

(1) Program to set Y8B (ON) when X8 turns ON, and reset Y8B (OFF) when X9 turns ON.



Coding

No. of steps	Com-mand	Device		
10	LD	X9		
11	RST	Y8B		
12	LD	X8		
13	SET	Y8B		
14				

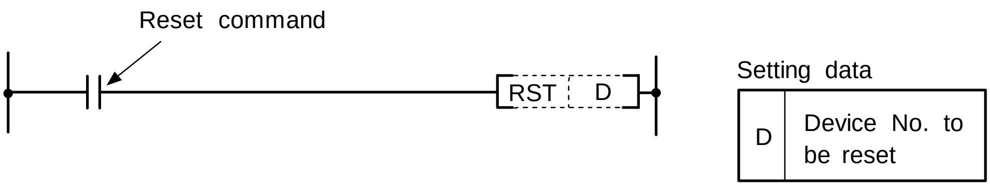


Operation of SET and RST commands

○ RST ... Device resetting

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D	○	○	○	○	○	○	○		○		○	○											○	1/2

(Note) In the modification of index, 2 steps are used for B or SB device.
2 steps are used for T or C device, also.



Function

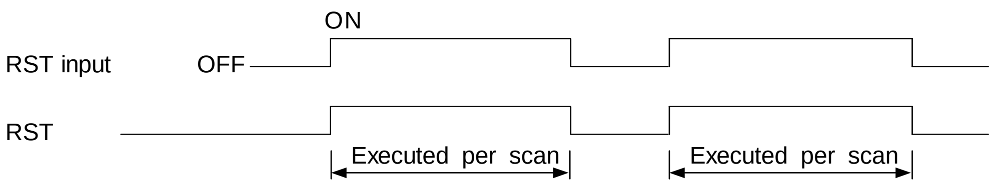
(1) The designated device will change as explained below when the RST input turns ON.

Device	Status
Y, M, L, SM, F	The coil and contact are turned OFF.
T, C	0 is set for the current value, and the coil and contact are turned OFF.

(2) If the RST input is OFF, the state of the device will not change.

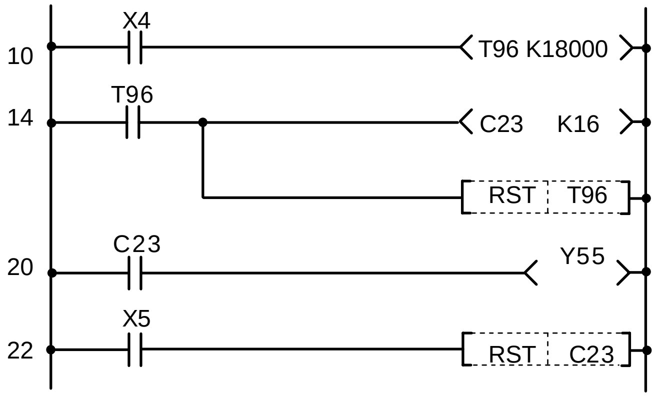
Execution condition

The execution conditions for the RST command are as shown below.



Program example

(1) Program to reset 100ms cumulative timer and counter.



When T96 is set for the cumulative timer, T96 will turn ON when the X4 ON time is 30 min.

The No. of times that T96 turns ON is counted.

T96 is reset when T96 turns ON.

Y55 turns ON when C23 counts up.

C23 is reset when X5 turns ON.

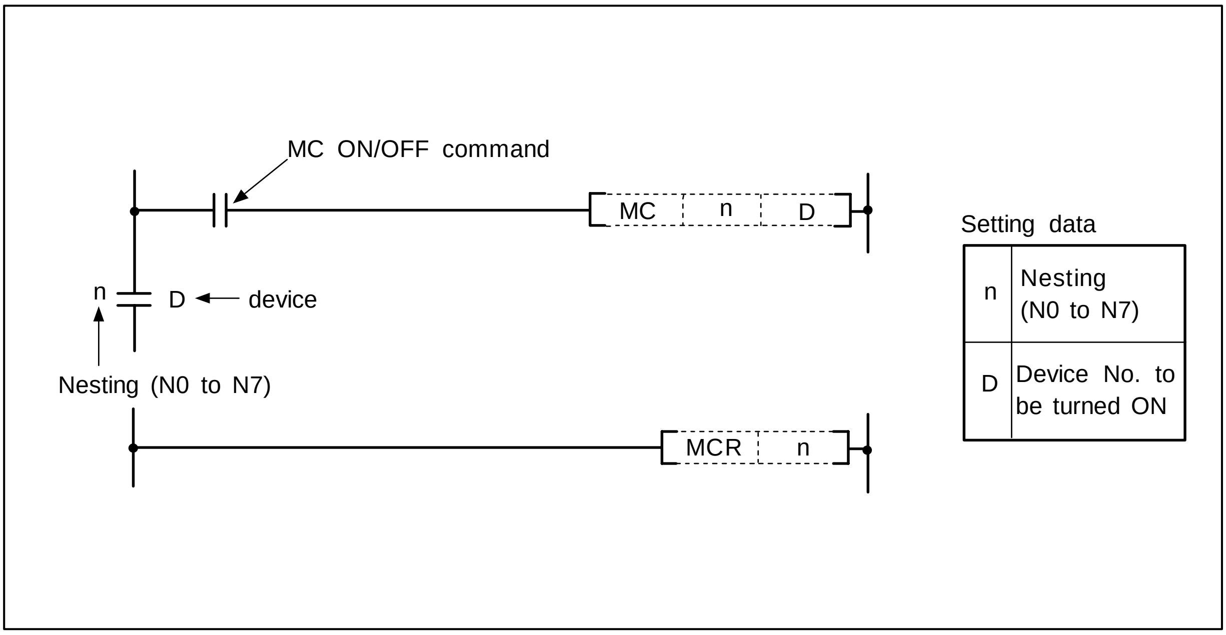
2 steps are used for T or C device.
1 step is used for the other devices.

Coding

No. of steps	Com-mand	Device		
10	LD	X4		
11	OUT	T96	K18000	
14	LD	T96		
15	OUT	C23	K16	
18	RST	T96		
20	LD	C23		
21	OUT	Y55		
22	LD	X5		
23	RST	C23		
25				

○ MC, MCR ... Master control set/reset

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
n																			○	○			○	2/1
D	○	○	○	○	○	○	○	○	○															



Function

MC

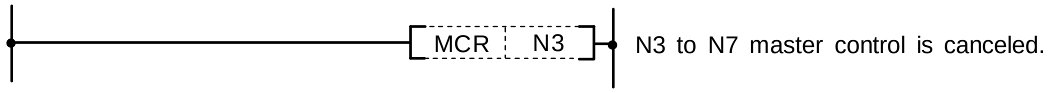
(1) If the MC ON/OFF command is ON when the master control starts, the operation results between MC and MCR will remain the same.
(2) If the MC ON/OFF command is OFF, the operation results between MC and MCR will be as follows.

100ms, 10ms timer	100ms cumulative timer counter	OUT command	SET/RST	SFT
Count value is set to 0	Current count value is held	All become OFF	The state is retained	

(3) Up to eight (N0 to N7) nests can be used. When using nests, the MC will use the nesting (N) from the smallest No., and MCR will use from the largest No.
(4) The program between the MC command and MCR command will be scanned regardless of the MC command ON/OFF state.
(5) By changing the destination D device, the MC command can be used as often as necessary in one scan.
(6) When the MC command is ON, the coil for the device designated as the destination will turn ON.

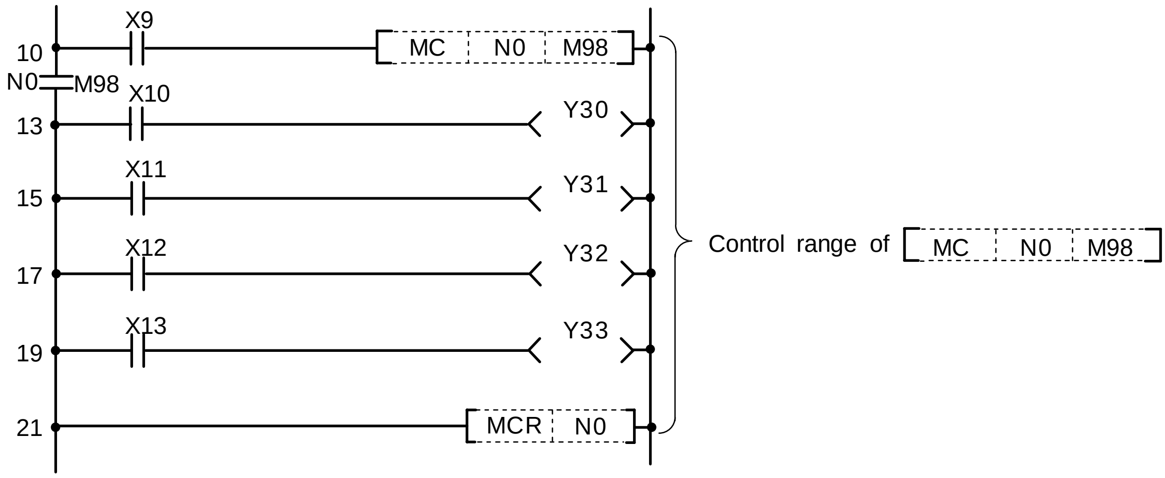
MCR

(1) This is the master control cancel command, and indicates the end of the master control range.
(2) The designated nesting (N) No. and following nests will be canceled.



Program example

(1) Program to turn MC ON when X9 is ON and turn MC OFF when OFF.

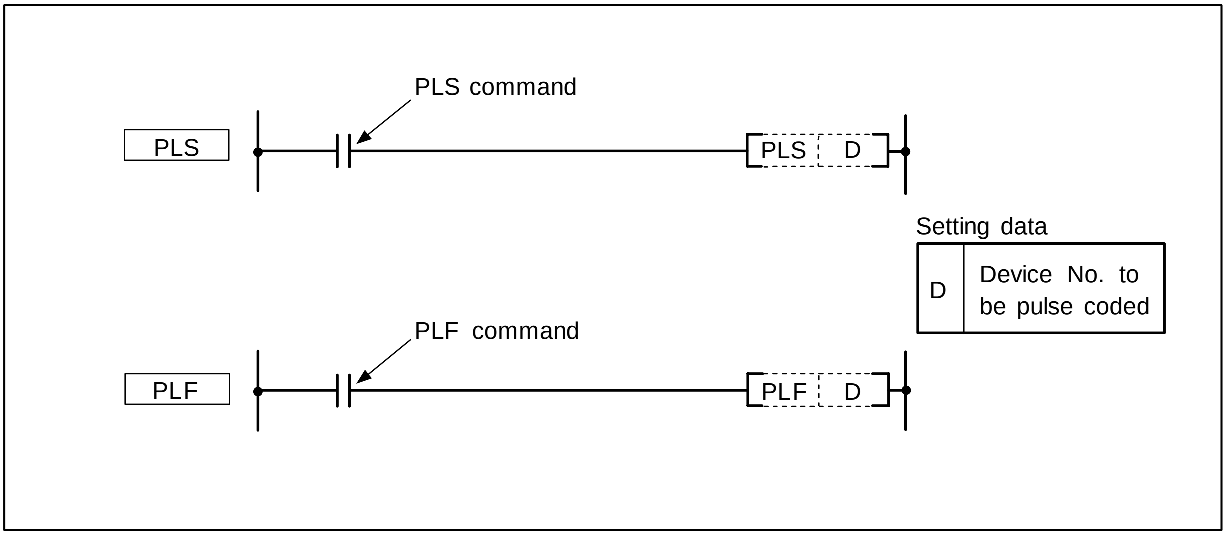


Coding

No. of steps	Com-mand	Device		
10	LD	X9		
11	MC	N0	M98	
13	LD	X10		
14	OUT	Y30		
15	LD	X11		
16	OUT	Y31		
17	LD	X12		
18	OUT	Y32		
19	LD	X13		
20	OUT	Y33		
21	MCR	N0		
22				

○ PLS, PLF ... Pulse (1 scan ON)

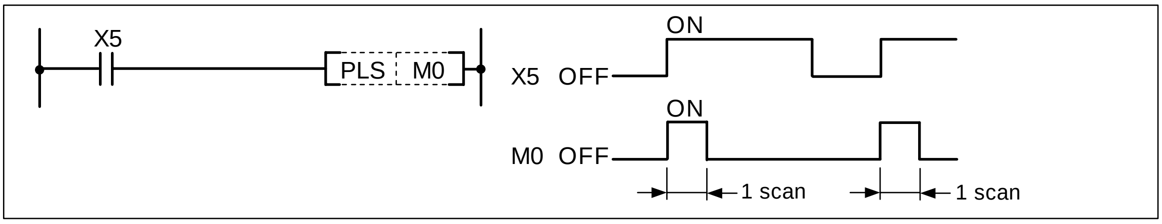
	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○			○														○	2



Function

PLS

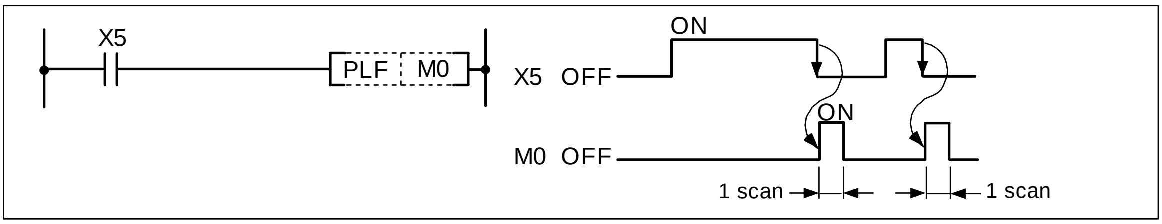
(1) The designated device is turned ON for one scan when the PLS command changes from OFF to ON and is turned OFF in all other cases.



(2) Even if the sequence program is changed from RUN to STOP and then RUN after the PLS command is executed, the PLS command will not be executed. If the PLS command is ON when the power is turned ON, the PLS command will be executed.

PLF

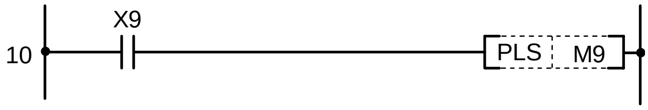
(1) The designated device is turned ON for one scan when the PLF command changes from ON to OFF and is turned OFF in all other cases.



(2) Even if the sequence program RUN switch is changed from RUN to STOP and then RUN after the PLF command is executed, the PLF command will not be executed.

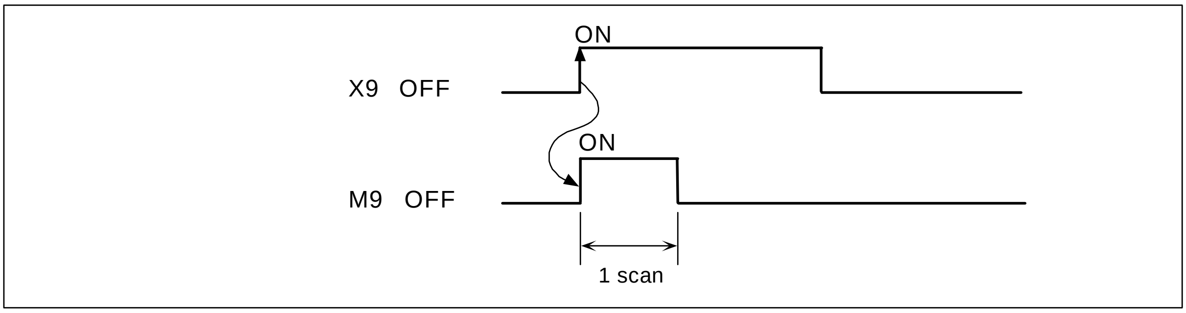
Program example

(1) Program to execute PLS command when X9 turns ON.

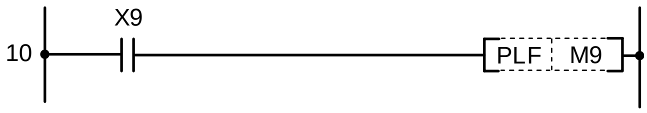


Coding

No. of steps	Com-mand	Device		
10	LD	X9		
11	PLS	M9		
13				

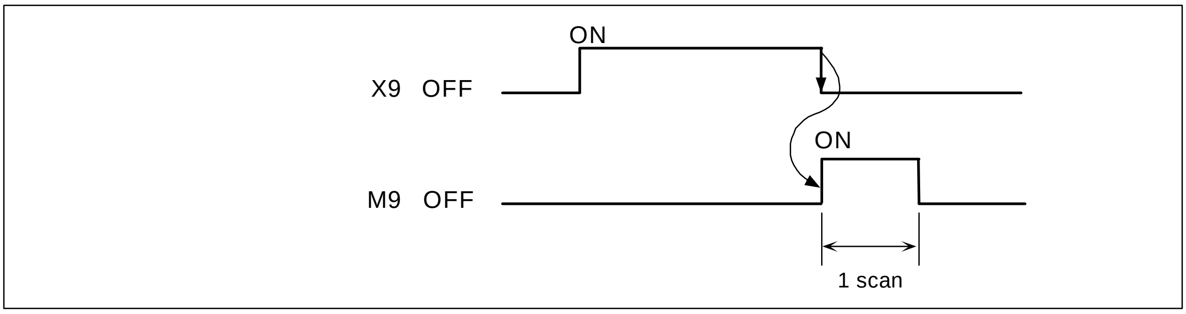


(2) Program to execute PLF command when X9 turns OFF.



Coding

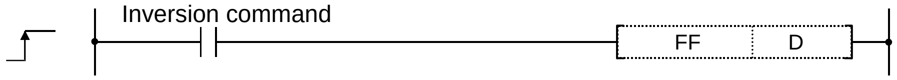
No. of steps	Com-mand	Device		
10	LD	X9		
11	PLF	M9		
13				



○ FF ... Bit device output inversion

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
		○	○	○	○	○	○		○														○	7

FF



Setting data

D	Device number to invert

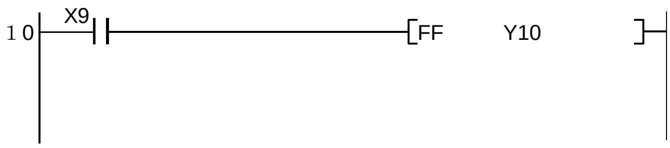
Function

(1) The status of the device designated by D is inverted when the inversion command turns from OFF to ON.

Device	Device status	
	Before FF execution	After FF execution
Bit device	OFF	ON
	ON	OFF
Designation of word device	0	1
	1	0

Program example

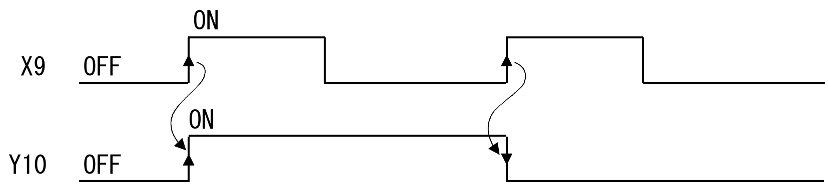
(1) Program to invert the output of Y10 when X9 turns ON.



Coding

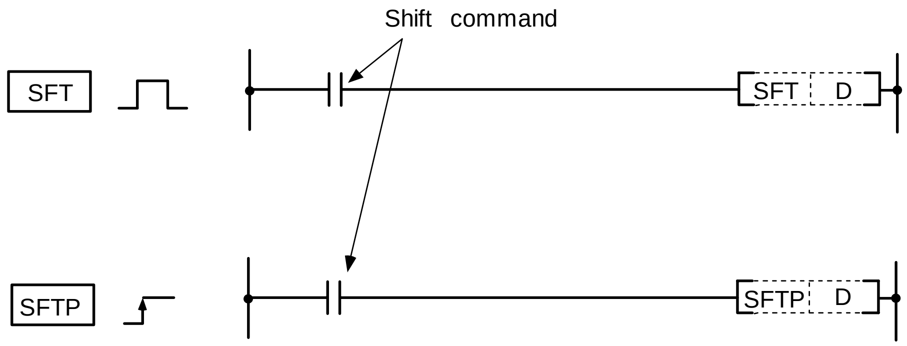
No. of steps	Com-Mand	Device		
10	LD	X9		
11	FF	Y10		
18				

[Timing Chart]



○ SFT, SFTP ... Device shift

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○				○															3

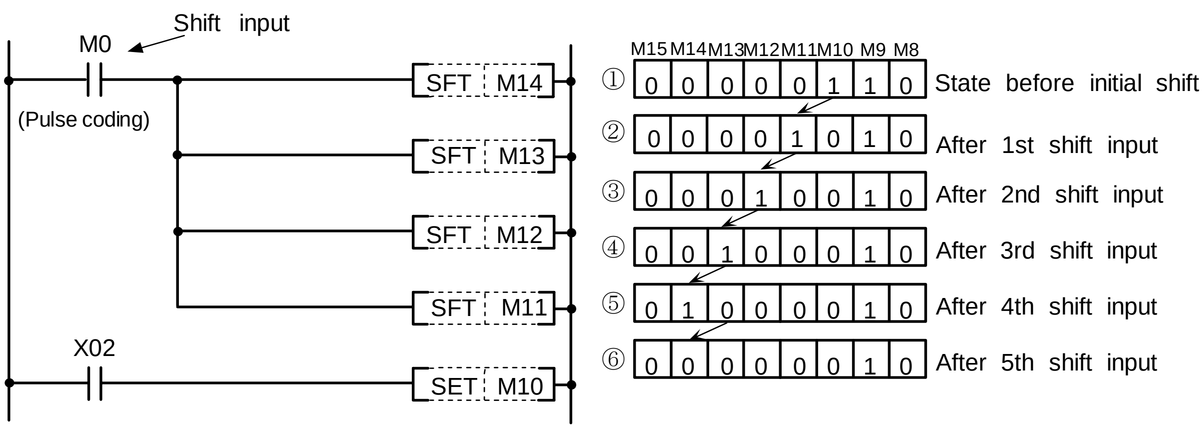


Setting data

D	Device No. to be shifted

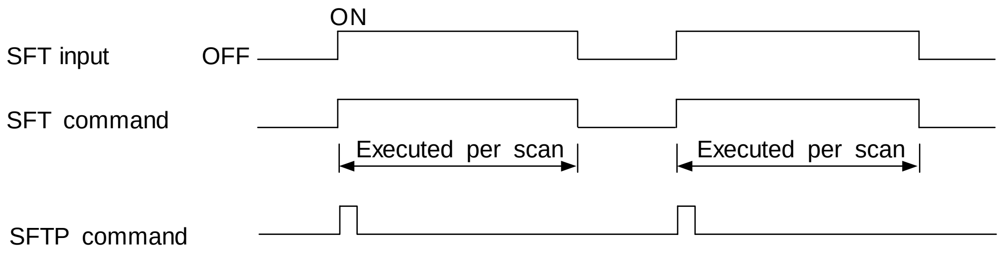
Function

(1) The device that designates the ON/OFF state of the device that is one number smaller than the device designated with D (destination) is shifted, and the device that is one number smaller is turned OFF.
(2) Turn the head device to be shifted ON with the SET command.
(3) When using SFT in succession, program from the largest device No.



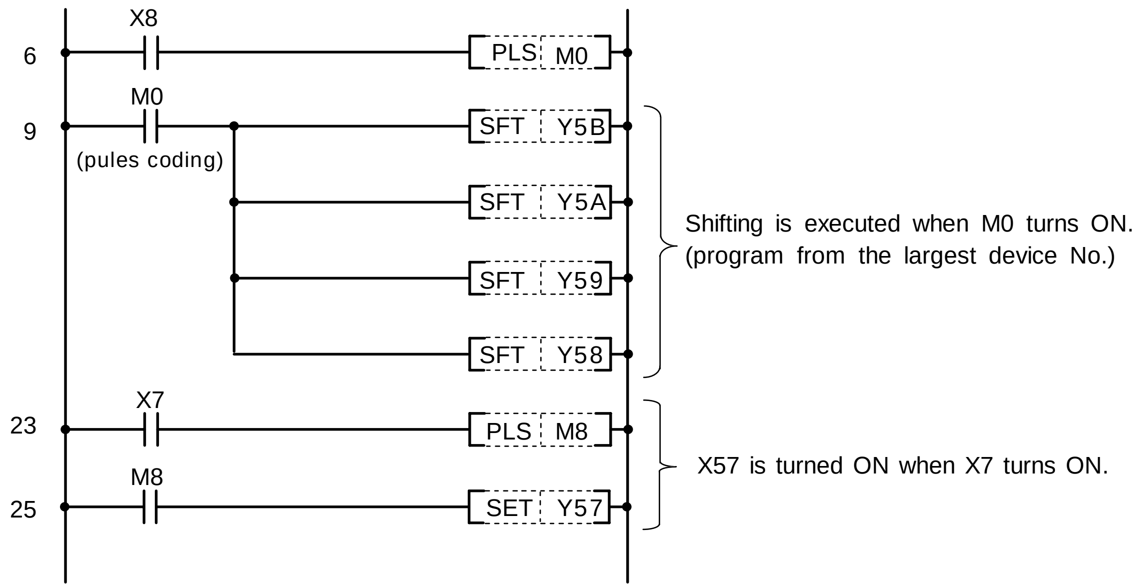
* In M8 to M15, "1" indicates ON and "0" indicates OFF.

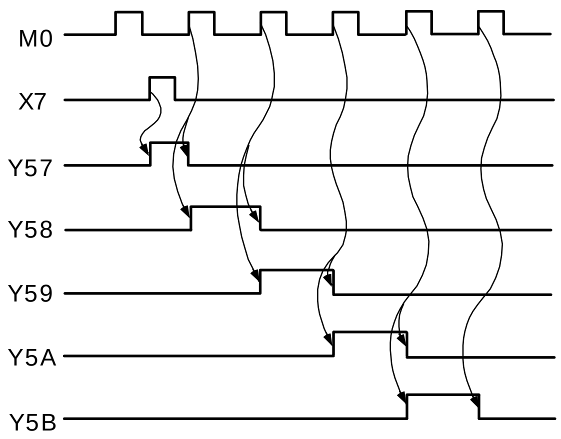
Operation of shift command



Program example

(1) Program to shift the data of Y57 to Y5B when X8 turns ON.



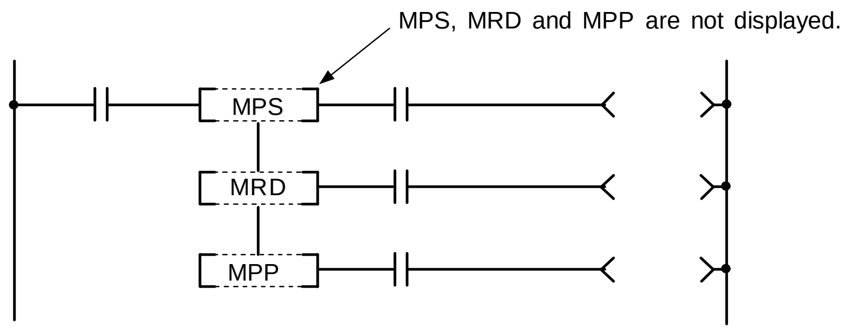


Coding

No. of steps	Com-mand	Device		
6	LD	X8		
7	PLS	M0		
9	LD	M0		
10	SFT	Y5B		
13	SFT	Y5A		
16	SFT	Y59		
19	SFT	Y58		
22	LD	X7		
23	PLS	M8		
25	LD	M8		
26	SET	Y57		
27				

○ MPS, MRD, MPP ... Registering, reading and clearing of operation results

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
																								1



Function

MPS

(1) The operation results (ON/OFF) just before the MPS command are registered.
(2) The MPS command can be used consecutively up to four times. If the MPP command is used in between, the No. of MPS usages will be decremented by one.

MRD

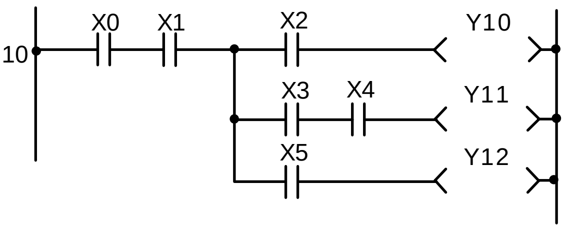
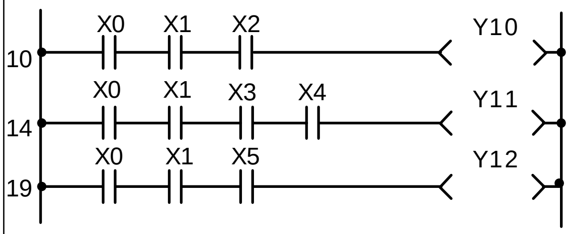
(1) The operation results registered with the MPS command are read, and the operation is continued from the next step using those operation results.

MPP

(1) The operation results registered with the MPS command are read, and the operation is continued from the next step using those operation results.
(2) The operation results registered with the MPS command are cleared.

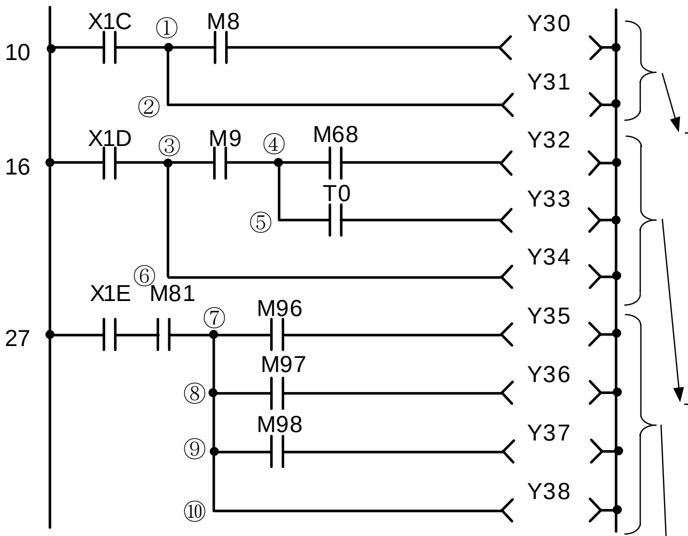
Points

(1) The circuits when MPS, MRD and MPP are used and not used are as follow.

Circuit using MPS, MRD and MPP	Circuit not using MPS, MRD and MPP
	

Program example

(1) Program using MPS, MRD and MPP.

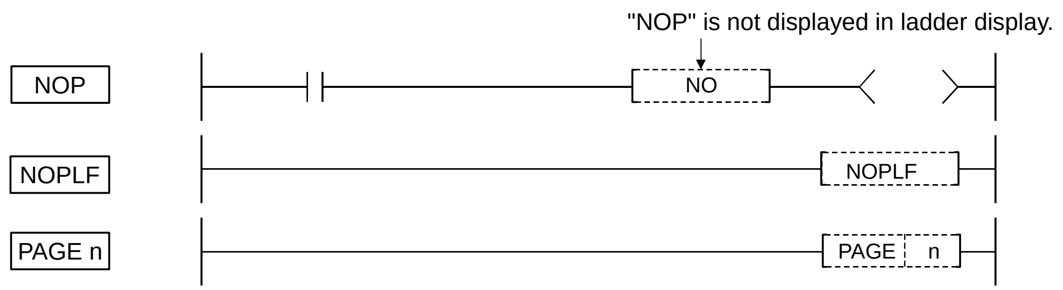


Coding

No. of steps	Com-mand	Device		
10	LD	X1C		
11	MPS			
12	AND	M8		
13	OUT	Y30		
14	MPP			
15	OUT	Y31		
16	LD	X1D		
17	MPS			
18	ANI	M9		
19	MPS			
20	AND	M68		
21	OUT	Y32		
22	MPP			
23	AND	T0		
24	OUT	Y33		
25	MPP			
26	OUT	Y34		
27	LD	X1E		
28	AND	M81		
29	MPS			
30	AND	M96		
31	OUT	Y35		
32	MRD			
33	AND	M97		
34	OUT	Y36		
35	MRD			
36	AND	M98		
37	OUT	Y37		
38	MPP			
39	OUT	Y38		
40				

○ NOP, NOPLF, PAGE n ... No operation

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
n																			○	○				1/1/2



Function

NOP

(1) This is a no-operation command that has no impact on any operation up to that point.
(2) The NOP command is used in the following cases:
    (a) To insert space for sequence program debugging.
    (b) To delete an command without having to change the number of steps. (Replace the command with NOP.)
    (c) To temporarily delete an command.

NOPLF

(1) This is a no-operation command that has no impact on any operation up to that point.
(2) The NOPLF command is used when printing from a peripheral device to insert a page break at any desired location.
    (a) When printing ladders
        • A page break will be inserted between ladder blocks with the presence of the NOPLF command.
        • The ladder cannot be displayed correctly if an NOPLF command is inserted in the midst of a ladder block.
          Do not insert an NOPLF command in the midst of a ladder block.

    (b) When printing command lists
        • A page break will be inserted after the printing of the NOPLF command.
(3) See the Operating Manual for the peripheral device in use for more information regarding printing from peripheral devices.

PAGE n

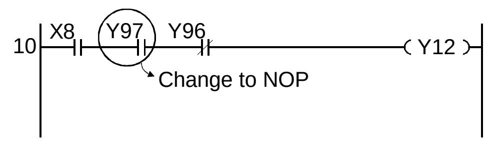
(1) This is a no-operation command that has no impact on any operation up to that point.
(2) Causes processing from step 0 of the designated nth page of the program following the PAGEn command. (Peripheral device display, printers, etc.)
(3) If there is no PAGEn command, processing begins from page 0.

Program example

NOP

(1) Contact closed……….Deletes AND or ANI command

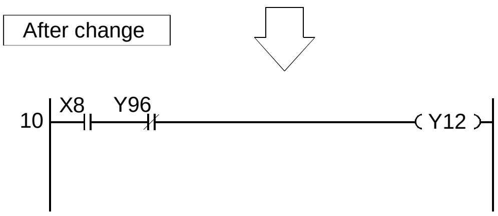
Before change



Coding

No. of steps	Com-mand	Device		
10	LD	X8		
11	AND	Y97		
12	ANI	X96		
13	OUT	Y12		
14				

After change

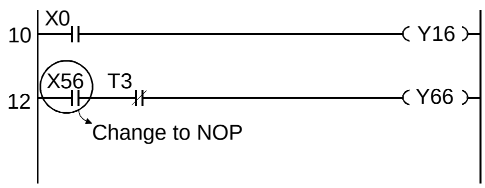


Coding

No. of steps	Com-mand	Device		
10	LD	X8		
11	NOP			
12	ANI	Y96		
13	OUT	Y12		
14				

(2) Contact closed……….Replaces LD or LDI command with NOP command.
(Note that when LD or LDI command is replaced with NOP command, the circuit will quite differ from the original one.)

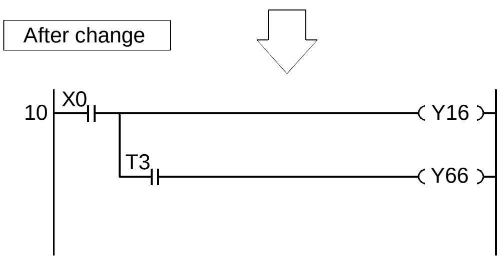
Before change



Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	OUT	Y16		
12	LD	Y56		
13	AND	T3		
14	OUT	Y66		
15				

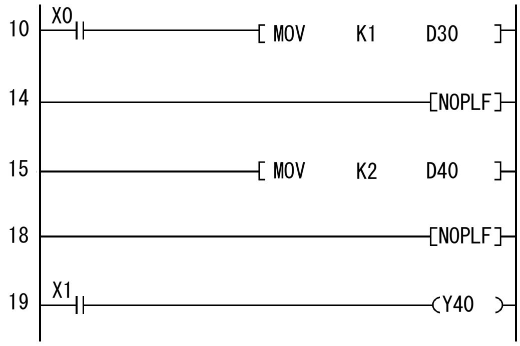
After change



Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	OUT	Y16		
12	NOP			
13	AND	T3		
14	OUT	Y66		
15				

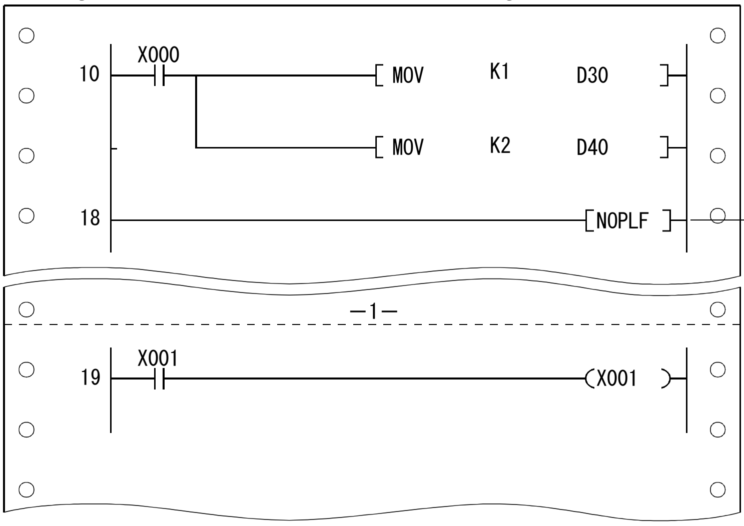
NOPLF



Coding

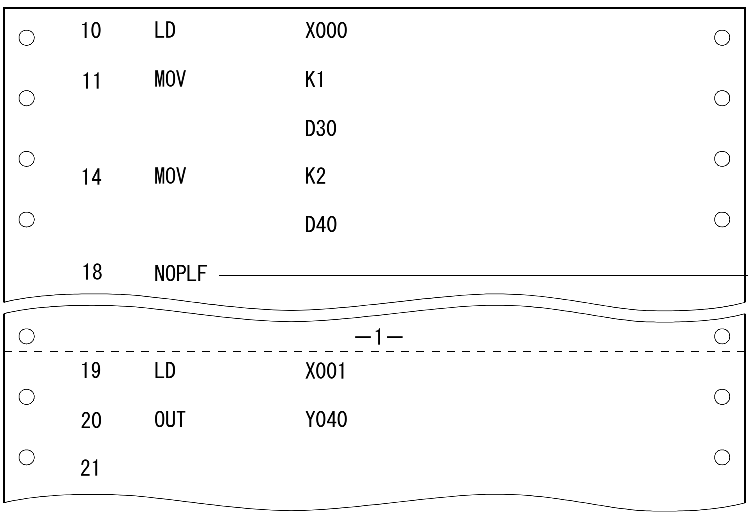
No. of steps	Com-mand	Device		
10	LD	X0		
11	MOV	K1	D30	
14	NOPLF			
15	MOV	K2	D40	
18	NOPLF			
19	LD	X1		
20	OUT	Y40		
21				

• Printing the ladder will result in the following:



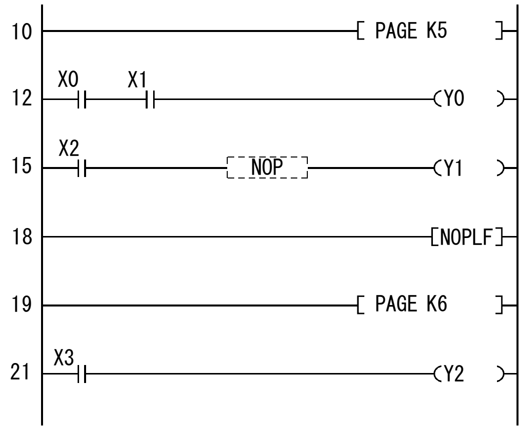
Page break inserted when NOPLF is inserted between ladder blocks.

• Printing an command list with the NOPLF command will result in the following:



Page break inserted after printing NOPLF.

PAGE n



Coding

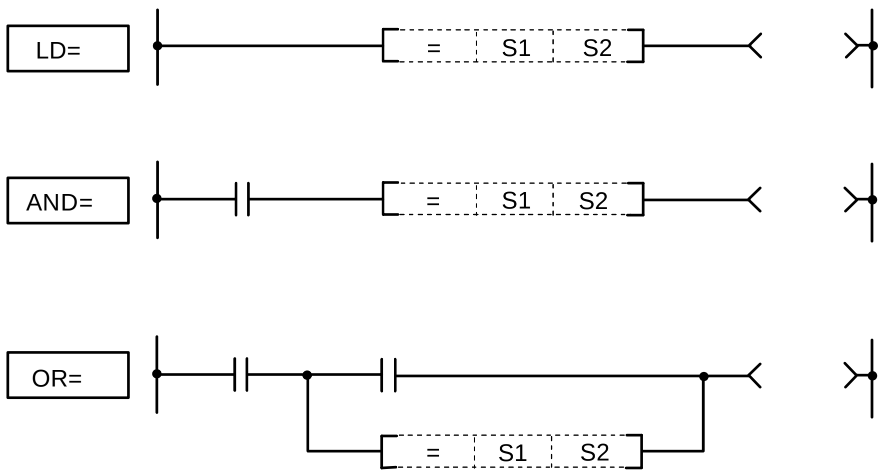
No. of steps	Com-mand	Device		
10	PAGE	K5		
12	LD	X0		
13	AND	X1		
14	OUT	Y0		
15	LD	X2		
16	NOP			
17	OUT	Y1		
18	NOPLF			
19	PAGE	K6		
21	LD	X3		
22	OUT	Y2		
23				

8. Function Commands

Recent sequence programs that require more advanced control cannot provide sufficient control only with basic commands and thus need four-rule operation and comparison, etc.
Many function commands have been prepared for this. There are approx. 118 types of function commands.
Each command is explained in the following section.

○ LD=, AND=, OR= .... Comparison of 16-bit data (=)

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				



Setting data

S1	Comparison data or No. of device where comparison data is stored.
S2	

Function

(1) 16-bit comparison operation is executed with "A" contact handling.
(2) The comparison operation results will be as follow.

Conditions	Comparison operation results
S1=S2	Continuity state
S1≠S2	Non-continuity state

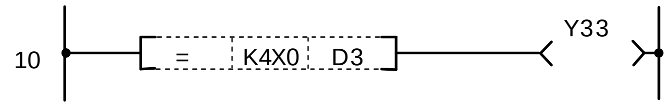
Execution conditions

The execution conditions for LD=, AND= and OR= are as follow.

Command	Execution conditions
LD=	Executed per scan
AND=	Executed only when previous contact command is ON
OR=	Executed per scan

Program example

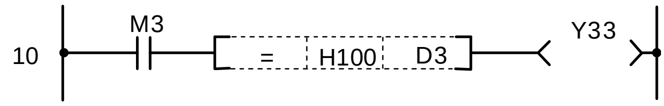
(1) Program to compare the X0 to F data and D3 data.



Coding

No. of steps	Com-mand	Device		
10	LD=	K4X0	D3	
13	OUT	Y33		
14				

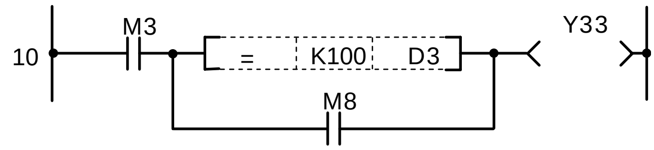
(2) Program to compare the BCD value 100 and D3 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND=	H100	D3	
14	OUT	Y33		
15				

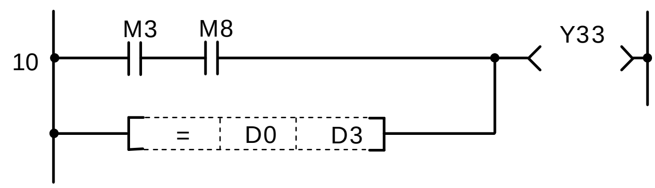
(3) Program to compare the BIN value 100 and D3 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	LD=	K100	D3	
14	OR	M8		
15	ANB			
16	OUT	Y33		
17				

(4) Program to compare the D0 and D3 data.

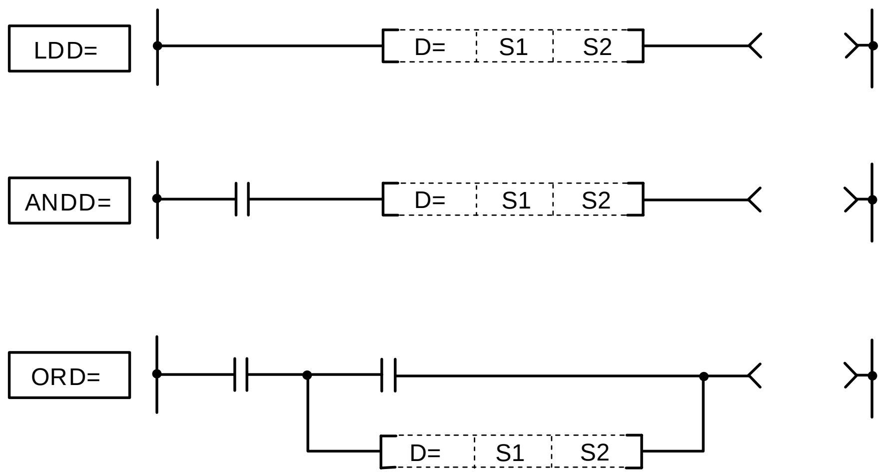


Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND	M8		
12	OR=	D0	D3	
15	OUT	Y33		
16				

○ LDD=, ANDD=, ORD= ... Comparison of 32-bit data (=)

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3/4
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				



Setting data

S1	Comparison data or head No. of device where comparison data is stored.
S2	

Function

(1) 32-bit comparison operation is executed with "A" contact handling.
(2) The comparison operation results will be as follow.

Conditions	Comparison operation results
S1=S2	Continuity state
S1≠S2	Non-continuity state

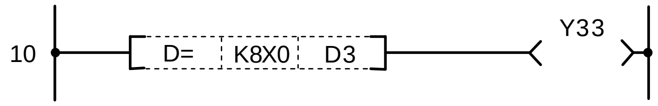
Execution conditions

The execution conditions for LDD=, ANDD= and ORD= are as follow.

Command	Execution conditions
LDD=	Executed per scan
ANDD=	Executed only when previous contact command is ON
ORD=	Executed per scan

Program example

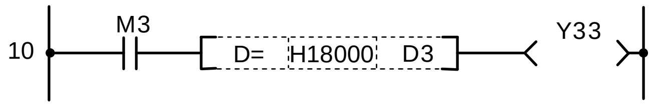
(1) Program to compare the X0 to 1F data and D3, 4 data.



Coding

No. of steps	Com-mand	Device		
10	LDD=	K8X0	D3	
13	OUT	Y33		
14				

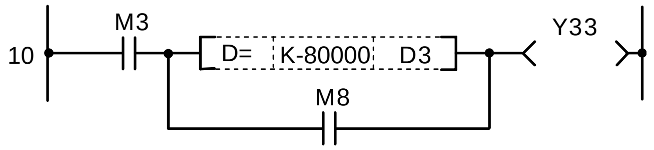
(2) Program to compare the BCD value 18000 and D3, 4 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	ANDD=	H18000	D3	
15	OUT	Y33		
16				

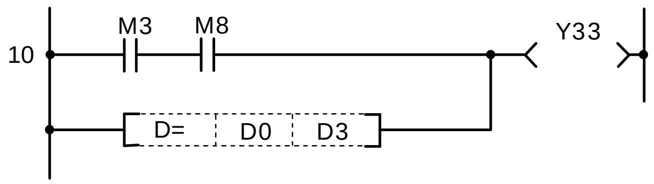
(3) Program to compare the BIN value -80000 and D3, 4 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	LDD=	K-80000	D3	
15	OR	M8		
16	ANB			
17	OUT	Y33		
18				

(4) Program to compare the D0, 1 data and D3, 4 data.

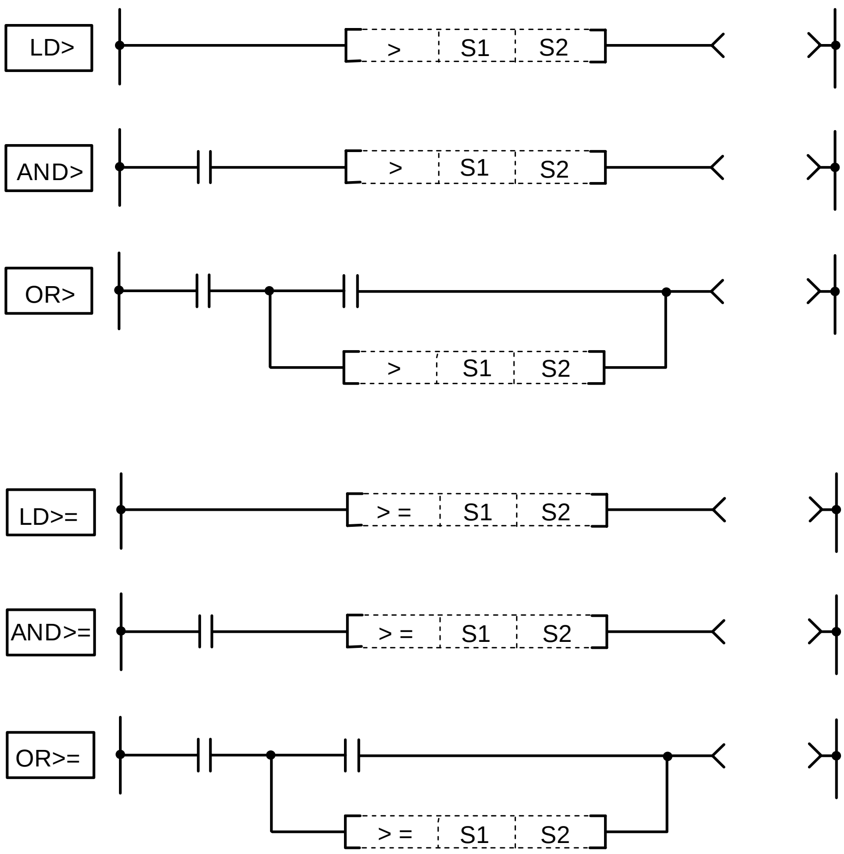


Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND	M8		
12	ORD=	D0	D3	
15	OUT	Y33		
16				

○ LD>, AND>, OR>, LD>=, AND>=, OR>= .... Comparison of 16-bit data (>, >=)

	Usable device																					Digit desig-nation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				



Setting data

S1	Comparison data or No. of device where comparison data is stored.
S2	

Function

(1) 16-bit comparison operation is executed with "A" contact handling.
(2) The comparison operation results will be as follow.

Command	Conditions	Comparison operation results
LD>, AND>, OR>	S1>S2	Continuity state
	S1<=S2	Non-continuity state
LD>=, AND>=, OR>=	S1>=S2	Continuity state
	S1<S2	Non-continuity state

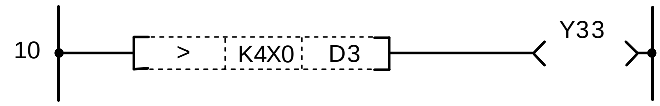
Execution conditions

The execution conditions for LD>, AND>, OR>, LD>=, AND>= and OR>= are as follow.

Command	Execution conditions
LD>, LD>=	Executed per scan
AND>, AND>=	Executed only when previous contact command is ON
OR>, OR>=	Executed per scan

Program example

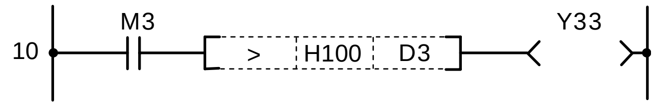
(1) Program to compare the X0 to F data and D3 data.



Coding

No. of steps	Com-mand	Device		
10	LD>	K4X0	D3	
13	OUT	Y33		
14				

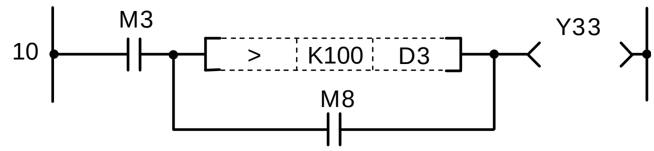
(2) Program to compare the BCD value 100 and D3 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND>	H100	D3	
14	OUT	Y33		
15				

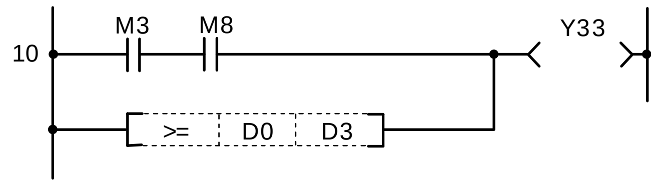
(3) Program to compare the BIN value 100 and D3 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	LD>	K100	D3	
14	OR	M8		
15	ANB			
16	OUT	Y33		
17				

(4) Program to compare the D0 and D3 data.

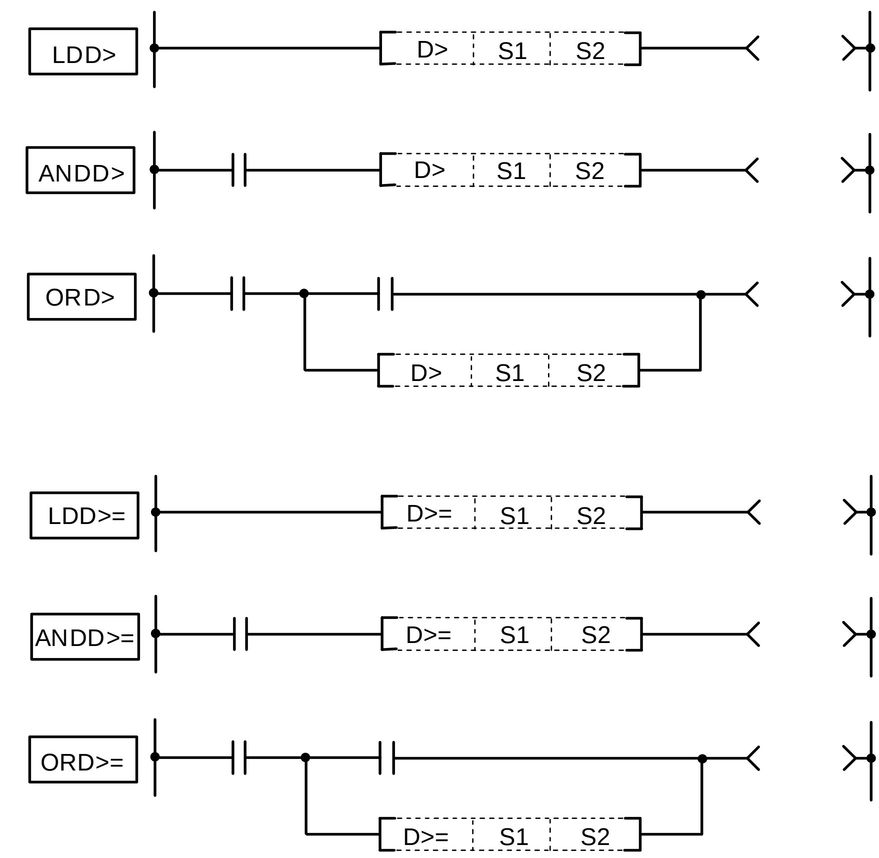


Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND	M8		
12	OR>=	D0	D3	
15	OUT	Y33		
16				

○ LDD>, ANDD>, ORD>, LDD>=, ANDD>=, ORD>= ... Comparison of 32-bit data (>, >=)

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3/4
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				



Setting data

S1	Comparison data or head No. of device where comparison data is stored.
S2	

Function

(1) 32-bit comparison operation is executed with "A" contact handling.
(2) The comparison operation results will be as follow.

Command	Conditions	Comparison operation results
LDD>, ANDD>, ORD>	S1>S2	Continuity state
	S1<=S2	Non-continuity state
LDD>=, ANDD>=, ORD>=	S1>=S2	Continuity state
	S1<S2	Non-continuity state

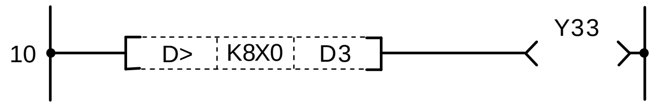
Execution conditions

The execution conditions for LDD>, ANDD>, ORD>, LDD>=, ANDD>= and ORD>= are as follow.

Command	Execution conditions
LDD>, LDD>=	Executed per scan
ANDD>, ANDD>=	Executed only when previous contact command is ON
ORD>, ORD>=	Executed per scan

Program example

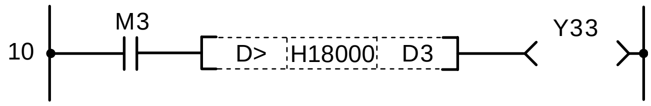
(1) Program to compare the X0 to 1F data and D3, 4 data.



Coding

No. of steps	Com-mand	Device		
10	LDD>	K8X0	D3	
13	OUT	Y33		
14				

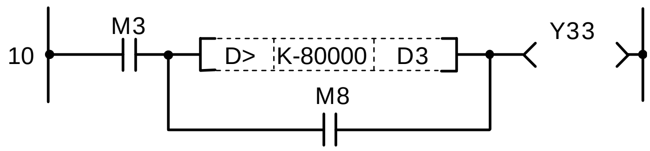
(2) Program to compare the BCD value 18000 and D3, 4 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	ANDD>	H18000	D3	
15	OUT	Y33		
16				

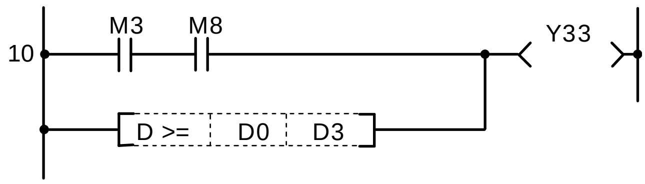
(3) Program to compare the BIN value -80000 and D3, 4 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	LDD>	K-80000	D3	
15	OR	M8		
16	ANB			
17	OUT	Y33		
18				

(4) Program to compare the D0, 1 data and D3, 4 data.

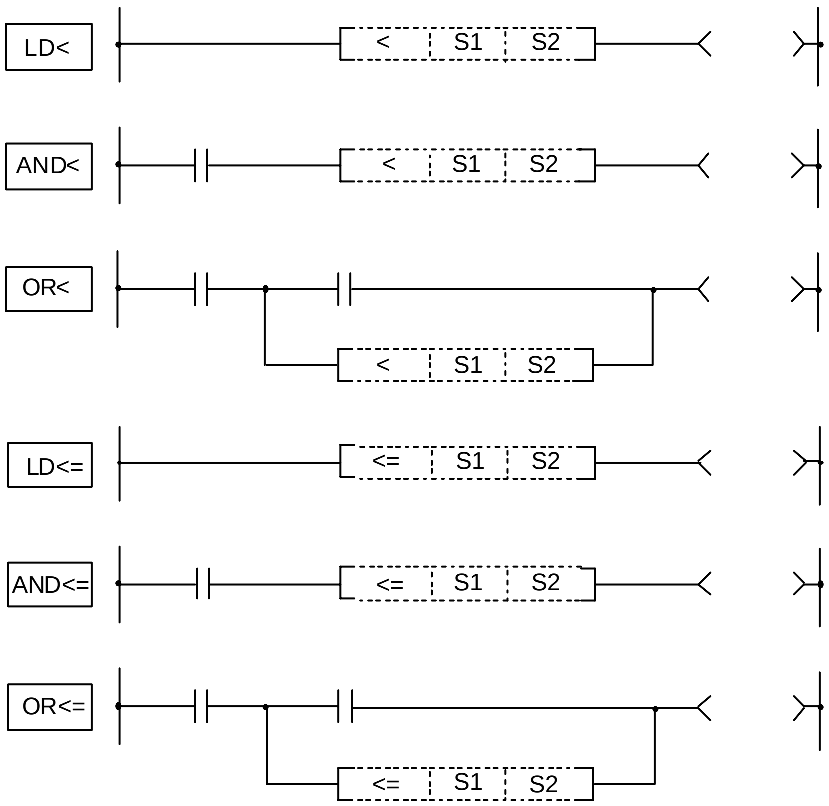


Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND	M8		
12	ORD>=	D0	D3	
15	OUT	Y33		
16				

○ LD<, AND<, OR<, LD<=, AND<=, OR<= .... Comparison of 16-bit data (<)

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○ (Note)	○ (Note)				



Setting data

S1	Comparison data or No. of device where comparison data is stored.
S2	

Setting data

S1	Comparison data or No. of device where comparison data is stored.
S2	

Function

(1) 16-bit comparison operation is executed with "A" contact handling.
(2) The comparison operation results will be as follow.

Command	Conditions	Comparison operation results
LD<, AND<, OR<	S1<S2	Continuity state
	S1>=S2	Non-continuity state
LD<=, AND<=, OR<=	S1<=S2	Non-continuity state
	S1>S2	Continuity state

Execution conditions

The execution conditions for LD<, AND<, OR<, LD<=, AND<= and OR<= are as follow.

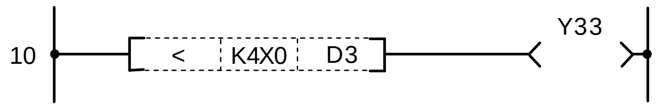
Command	Execution conditions
LD<, LD<=	Executed per scan
AND<, AND<=	Executed only when previous contact command is ON
OR<, OR<=	Executed per scan

Cautions

Each command of LD<=, AND<=, and OR<= cannot use a constant for S2. When S2 is a constant, it operates as LDBIT, ANDBIT, and an ORBIT command, respectively.

Program example

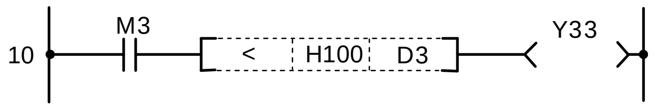
(1) Program to compare the X0 to F data and D3 data.



Coding

No. of steps	Com-mand	Device		
10	LD<	K4X0	D3	
13	OUT	Y33		
14				

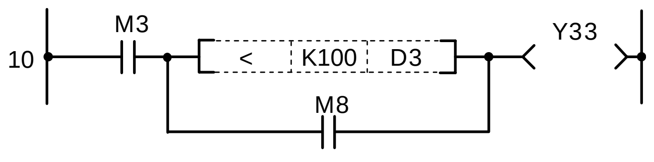
(2) Program to compare the BCD value 100 and D3 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND<	H100	D3	
14	OUT	Y33		
15				

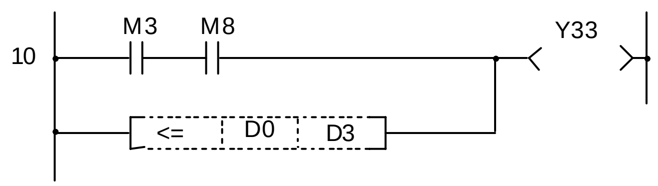
(3) Program to compare the BIN value 100 and D3 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	LD<	K100	D3	
14	OR	M8		
15	ANB			
16	OUT	Y33		
17				

(4) Program to compare the D0 and D3 data.

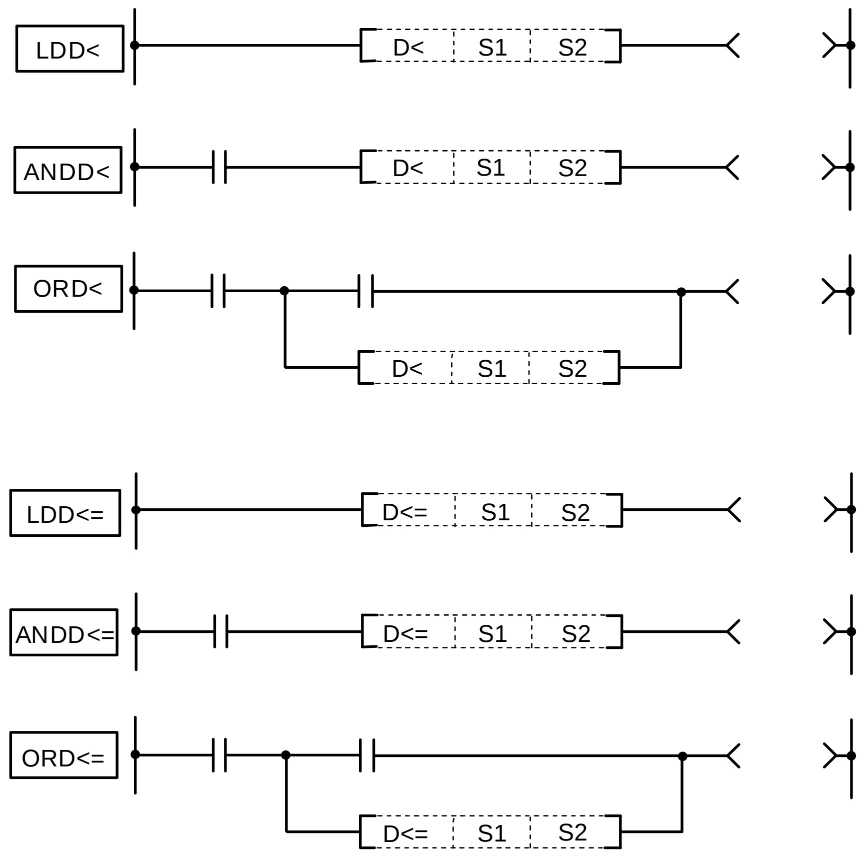


Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND	M8		
12	OR<	D0	D3	
15	OUT	Y33		
16				

○ LDD<, ANDD<, ORD<, LDD<=, ANDD<=, ORD<= ... Comparison of 32-bit data (<, <=)

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3/4
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				



Setting data

S1	Comparison data or head No. of device where comparison data is stored.
S2	

Function

(1) 32-bit comparison operation is executed with "A" contact handling.
(2) The comparison operation results will be as follows.

Command	Conditions	Comparison operation results
LDD<, ANDD<, ORD<	S1<S2	Continuity state
	S1>=S2	Non-continuity state
LDD<=, ANDD<=, ORD<=	S1<=S2	Continuity state
	S1>S2	Non-continuity state

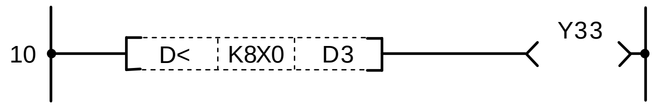
Execution conditions

The execution conditions for LDD<, ANDD<, ORD<, LDD<=, ANDD<= and ORD<= are as follows.

Command	Execution conditions
LDD<, LDD<=	Executed per scan
ANDD<, ANDD<=	Executed only when previous contact command is ON
ORD<, ORD<=	Executed per scan

Program example

(1) Program to compare the X0 to 1F data and D3, 4 data.



Coding

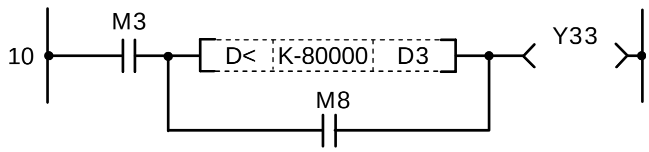
No. of steps	Com-mand	Device		
10	LDD<	K8X0	D3	
13	OUT	Y33		
14				

(2) Program to compare the BCD value 18000 and D3, 4 data.

Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	ANDD<	H18000	D3	
15	OUT	Y33		
16				

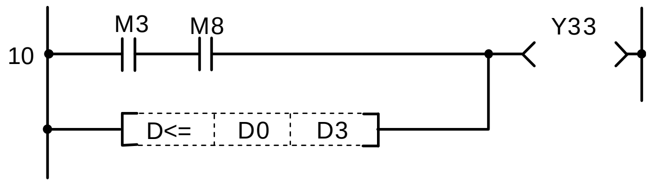
(3) Program to compare the BIN value -80000 and D3, 4 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	LDD<	K-80000	D3	
15	OR	M8		
16	ANB			
17	OUT	Y33		
18				

(4) Program to compare the D0, 1 data and D3, 4 data.

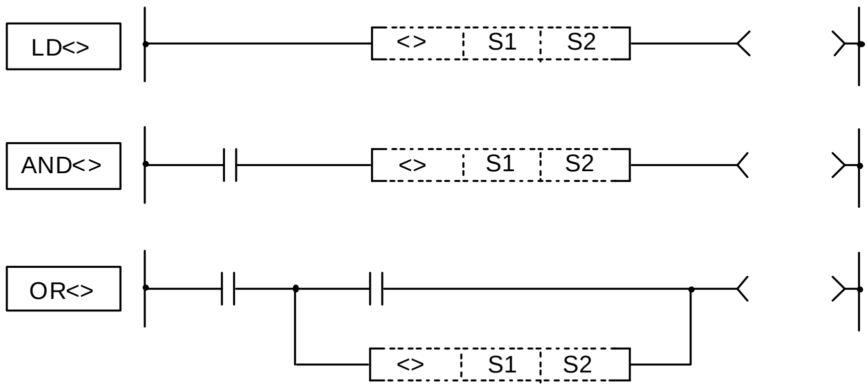


Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND	M8		
12	ORD<=	D0	D3	
15	OUT	Y33		
16				

○ LD<>, AND<>, OR<> .... Comparison of 16-bit data (<>)

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1	○	○	○	○	○	○	○		○	○	○	○	○	○	○	○	○	○	○	○		○	○	3/4
S2	○	○	○	○	○	○	○		○	○	○	○	○	○	○	○	○	○	○ (Note)	○ (Note)				



Setting data

S1 S2	Comparison data or head No. of device where comparison data is stored.

Function

(1) 16-bit comparison operation is executed with "A" contact handling.
(2) The comparison operation results will be as follow.

Conditions	Comparison operation results
S1<>S2	Continuity state
S1=S2	Non-continuity state

Execution conditions

The execution conditions for LD<>, AND<> and OR<> are as follow.

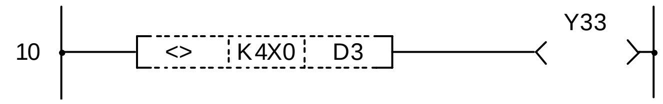
Command	Execution conditions
LD<>	Executed per scan
AND<>	Executed only when previous contact command is ON
OR<>	Executed per scan

Cautions

Each command of LD<>, AND<>, and OR<> cannot use a constant for S2. When S2 is a constant, it operates as LDBII, ANDBII, and an ORBII command, respectively.

Program example

(1) Program to compare the X0 to F data and D3 data.



Coding

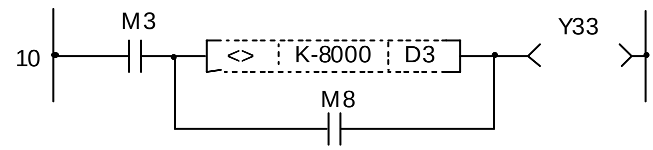
No. of steps	Com-mand	Device		
10	LD<>	K4X0	D3	
13	OUT	Y33		
14				

(2) Program to compare the BCD value 1800 and D3 data.

Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND<>	H1800	D3	
14	OUT	Y33		
15				

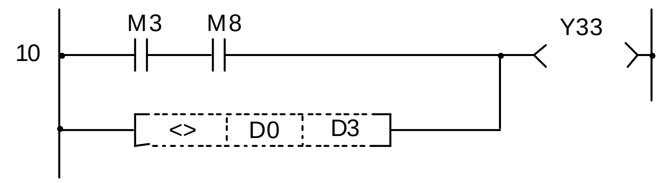
(3) Program to compare the BIN value -8000 and D3 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	LD<>	K-8000	D3	
15	OR	M8		
16	ANB			
17	OUT	Y33		
18				

(4) Program to compare the D0 data and D3 data.

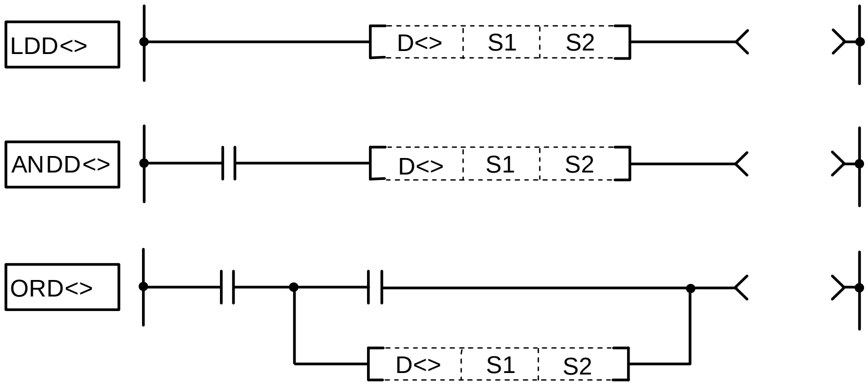


Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND	M8		
12	OR<>	D0	D3	
15	OUT	Y33		
16				

○ LDD<>, ANDD<>, ORD<> .... Comparison of 32-bit data (<>)

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3/4
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				



Setting data

S1	Comparison data or head No. of device where comparison data is stored.
S2	

Function

(1) 16-bit comparison operation is executed with "A" contact handling.
(2) The comparison operation results will be as follow.

Conditions	Comparison operation results
S1<>S2	Continuity state
S1=S2	Non-continuity state

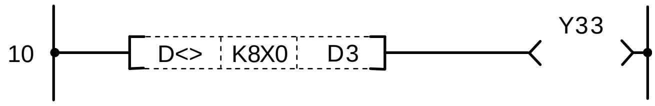
Execution conditions

The execution conditions for LDD<>, ANDD<> and ORD<> are as follow.

Command	Execution conditions
LDD<>	Executed per scan
ANDD<>	Executed only when previous contact command is ON
ORD<>	Executed per scan

Program example

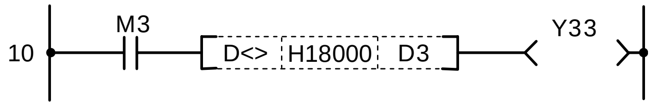
(1) Program to compare the X0 to F data and D3, D4 data.



Coding

No. of steps	Com-mand	Device		
10	LDD<>	K8X0	D3	
13	OUT	Y33		
14				

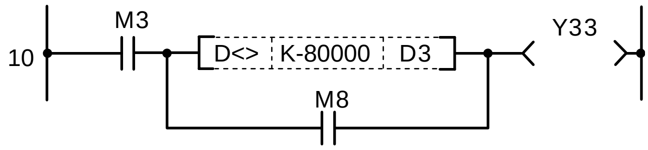
(2) Program to compare the BCD value 18000 and D3, D4 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	ANDD<>	H18000	D3	
14	OUT	Y33		
15				

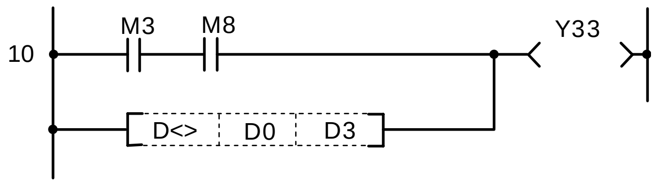
(3) Program to compare the BIN value -80000 and D3, D4 data.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	LDD<>	K-80000	D3	
15	OR	M8		
16	ANB			
17	OUT	Y33		
18				

(4) Program to compare the D0, D1 data and D3, D4 data.

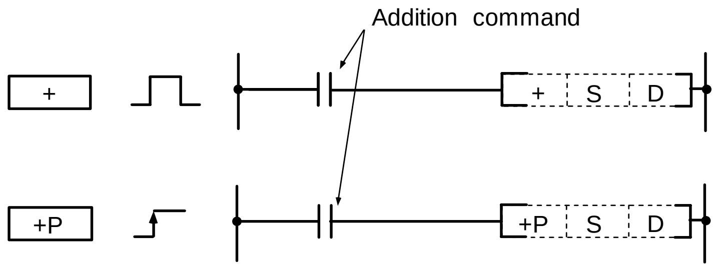


Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND	M8		
12	ORD<>	D0	D3	
15	OUT	Y33		
16				

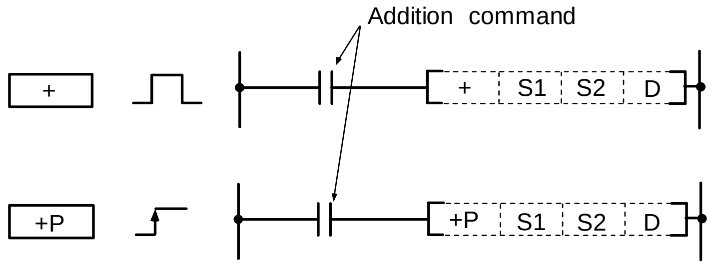
○+, +P ... BIN 16-bit addition

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	4
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Setting data (See *1)

S	Addition data or No. of device where addition data is stored.
D	Addition data or No. of device where addition data is stored. (The addition results are stored in this device.)

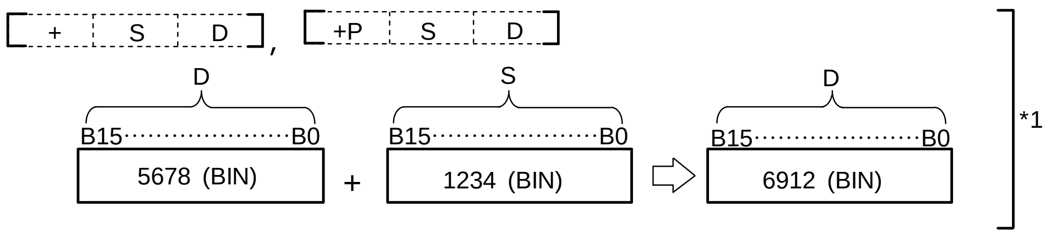


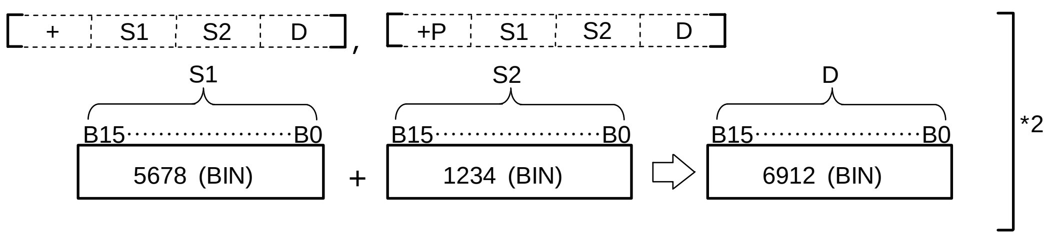
Setting data (See *2)

S1	Addition data or No. of device where addition data is stored.
S2	Addition data or No. of device where addition data is stored.
D	No. of device to store addition results.

Function

(1) The BIN data designated with S1 (D of *1) and the BIN data designated with S2 (S of *1) are added, and the addition results are stored in the device designated with D.





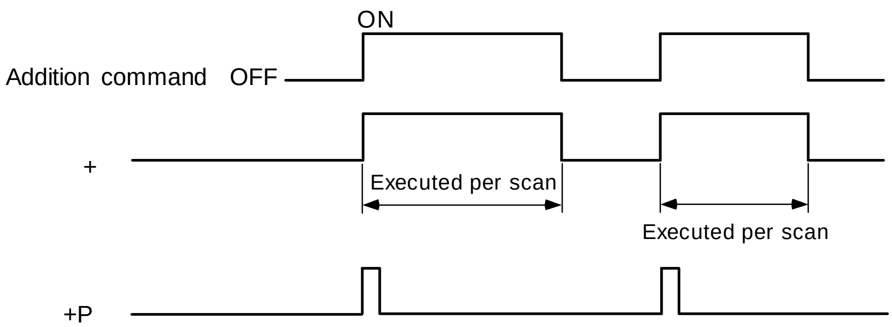
(2) -32768 to 32767 (BIN 16-bit) can be designated in S1, S2 and S.
(3) The positive/negative of the data in S1, S2, S and D is determined with the highest-order bit (B15).

B15	Judgment of positive/negative
0	Positive
1	Negative

(4) The carry flag will not turn ON if the 15th bit overflows.

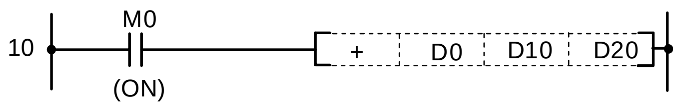
Execution conditions

The execution conditions for +, +P are as shown below.



Program example

(1) Program to add the D0 BIN data and D10 BIN data and output to D20.

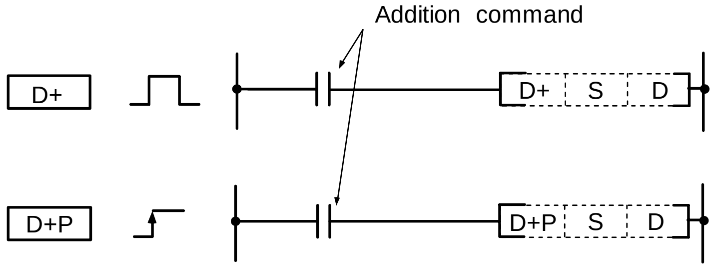


Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	+	D0	D10	D20
15				

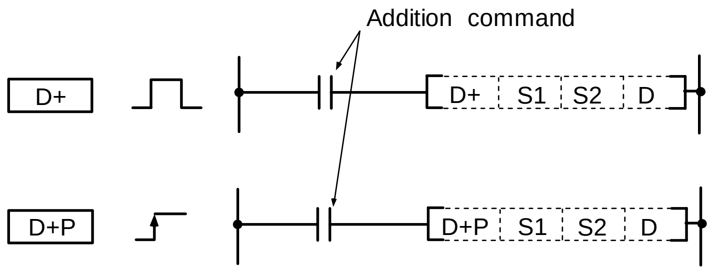
○ D+, D+P ... BIN 32-bit addition

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3/4
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	4/5
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Setting data (See *1)

S	Addition data or head No. of device where addition data is stored.
D	Addition data or head No. of device where addition data is stored. (The addition results are stored in this device.)

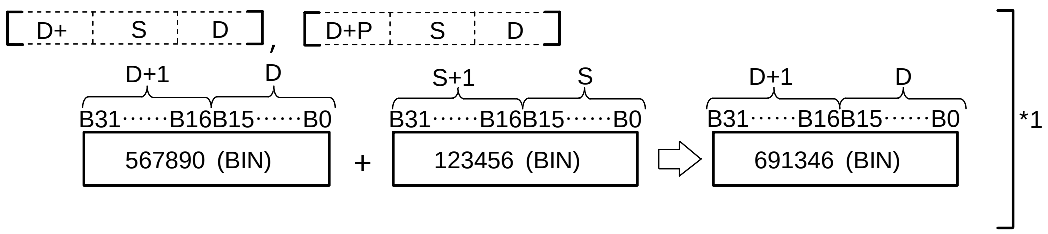


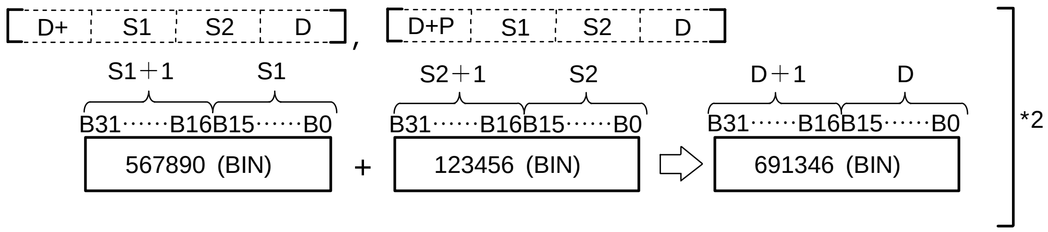
Setting data (See *2)

S1	Addition data or head No. of device where addition data is stored.
S2	Addition data or head No. of device where addition data is stored.
D	Head No. of device to store addition results.

Function

(1) The BIN data designated with S1 (D of *1) and the BIN data designated with S2 (S of *2) are added, and the addition results are stored in the device designated with D.





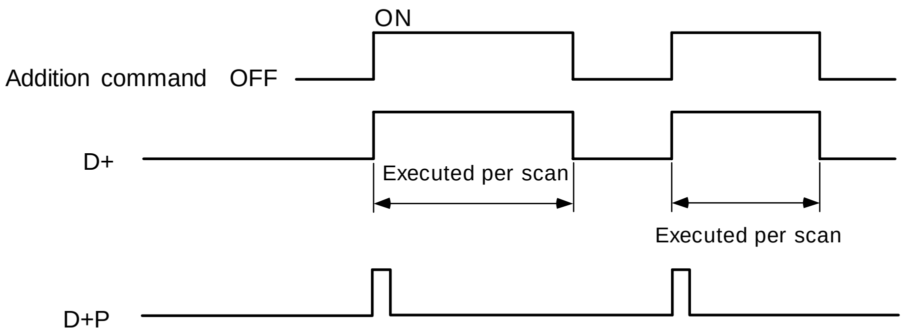
(2) -2147483648 to 2147483647 (BIN 32-bit) can be designated in S1, S2 and S.
(3) The positive/negative of the data in S1, S2, S and D is determined with the highest-order bit (B31).

B31	Judgment of positive/negative
0	Positive
1	Negative

(4) The carry flag will not turn ON if the 31st bit overflows.

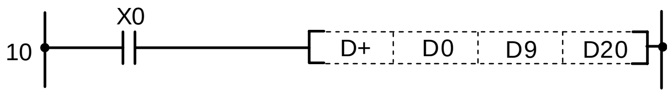
Execution conditions

The execution conditions for D+, D+P are as shown below.



Program example

(1) Program to add the D0, 1 data and D9, 10 data when X0 turns ON, and output the results to D20, 21.

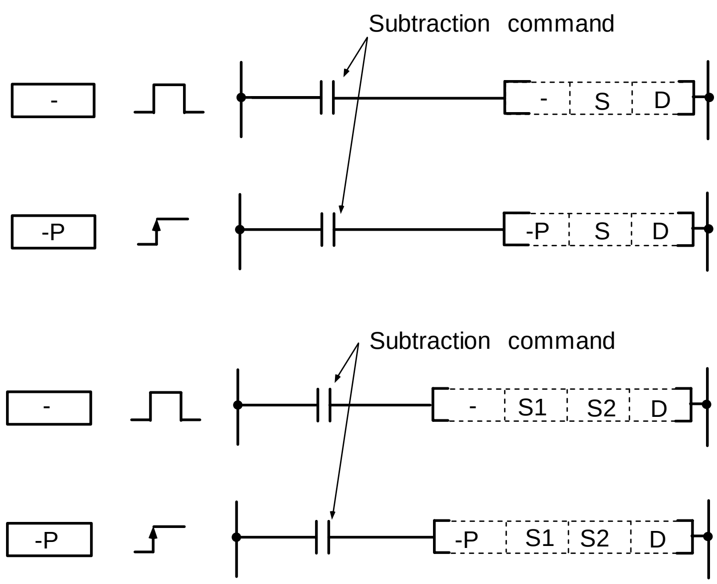


Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	D+	D0	D9	D20
15				

○ –, –P ... BIN 16-bit subtraction

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	4
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Setting data (See *1)

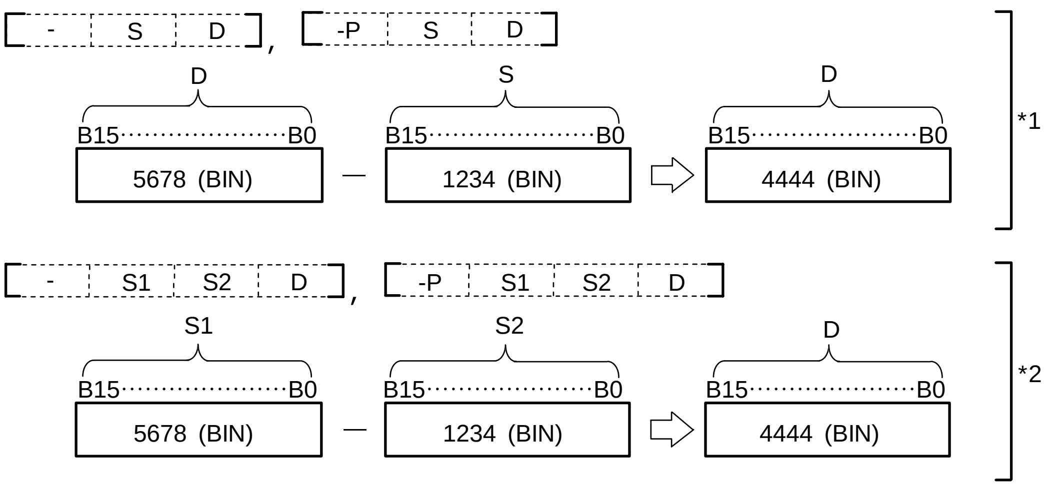
S	Subtraction data or No. of device where subtraction data is stored.
D	Subtraction data or No. of device where subtraction data is stored. (The subtracted results are stored in this device.)

Setting data (See *2)

S1	Subtraction data or No. of device where subtraction data is stored.
S2	Subtraction data or No. of device where subtraction data is stored.
D	No. of device to store subtraction results.

Function

(1) The device designated with S1 (D of *1) and the device designated with S2 (S of *2) are subtracted, and the subtracted results are stored in the device designated with D.



(2) -32768 to 32767 (BIN 16-bit) can be designated in S1, S2 and S.

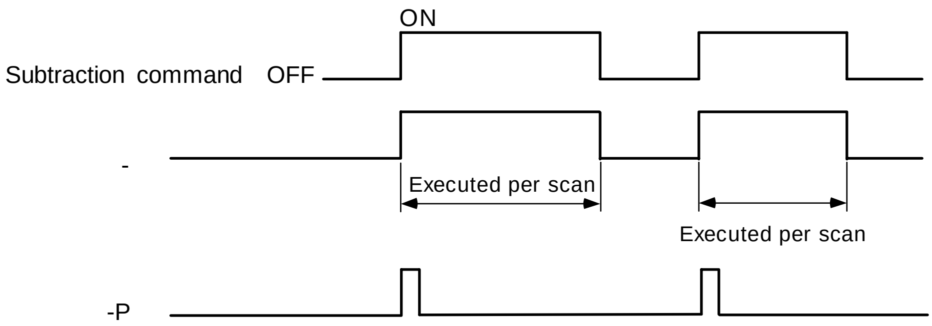
(3) The positive/negative of the data in S1, S2, S and D is determined with the highest-order bit (B15).

B15	Judgment of positive/negative
0	Positive
1	Negative

(4) The carry flag will not turn ON if the 0 bit underflows.

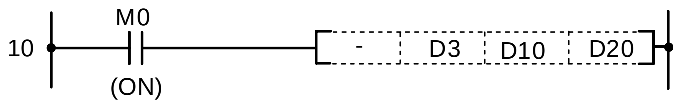
Execution conditions

The execution conditions for -, -P are as shown below.



Program example

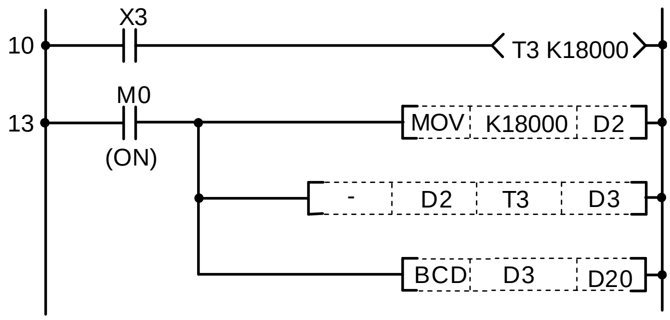
(1) Program to subtract the D10 BIN data from D3 and output to D20.



Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	-	D3	D10	D20
15				

(2) Program to output the difference of the timer T3 setting value and current value to D20 BCD data.

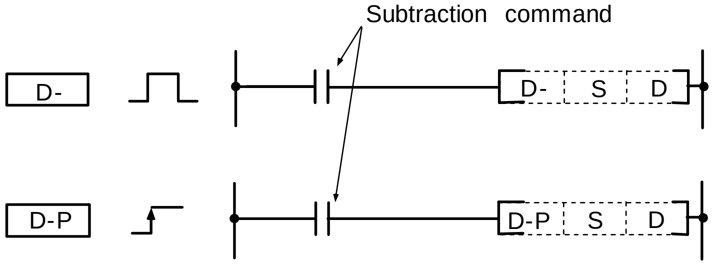


Coding

No. of steps	Com-mand	Device		
10	LD	X3		
11	OUT	T3	K18000	
13	LD	M0		
14	MOV	K18000	D2	
17	-	D2	T3	D3
21	BCD	D3	D20	
24				

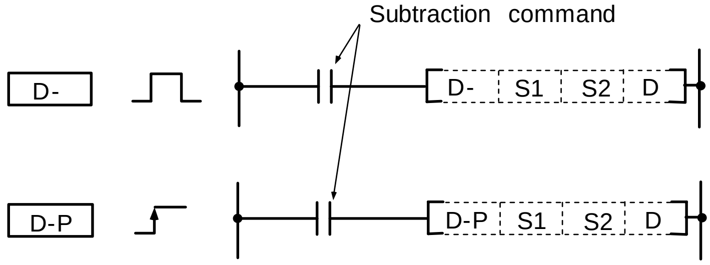
○ D–, D–P ... BIN 32-bit subtraction

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3/4
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	4/5
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Setting data (See *1)

S	Subtraction data or head No. of device where subtraction data is stored.
D	Subtraction data or head No. of device where subtraction data is stored. (The subtraction results are stored in this device.)

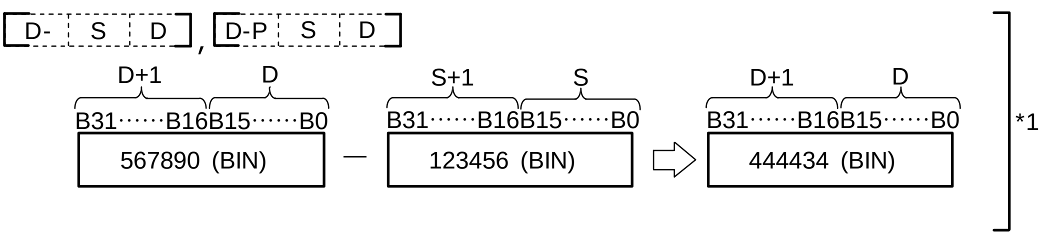


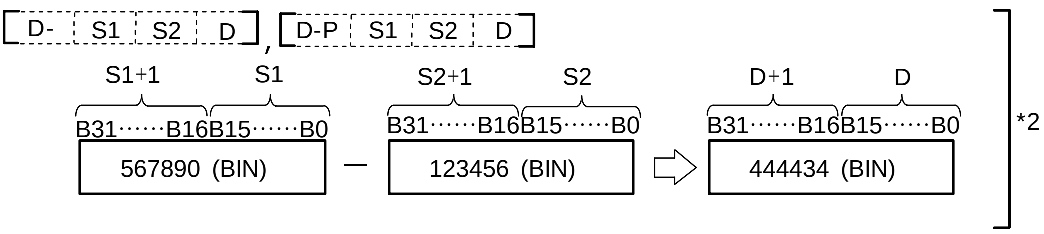
Setting data (See *2)

S1	Subtraction data or head No. of device where subtraction data is stored.
S2	Subtraction data or head No. of device where subtraction data is stored.
D	Head No. of device to store subtraction results.

Function

(1) The device designated with S1 (D of *1) and the device designated with S2 (S of *2) are subtracted, and the subtracted results are stored in the device designated with D.





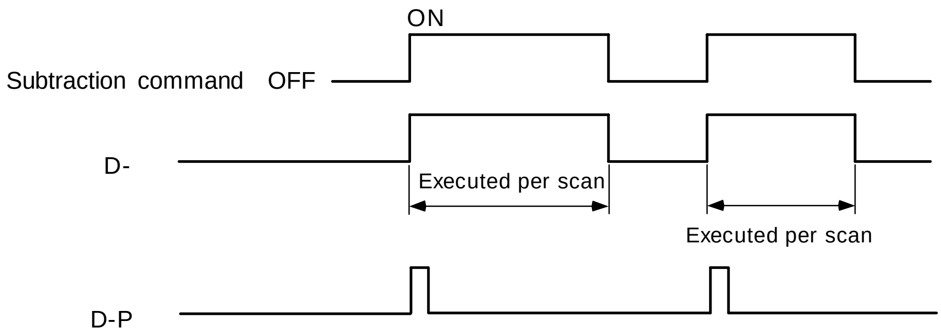
(2) -2147483648 to 2147483647 (BIN 32-bit) can be designated in S1, S2 and S.
(3) The positive/negative of the data in S1, S2, S and D is determined with the highest-order bit (B31).

B31	Judgment of positive/negative
0	Positive
1	Negative

(4) The carry flag will not turn ON if the 0th bit underflows.

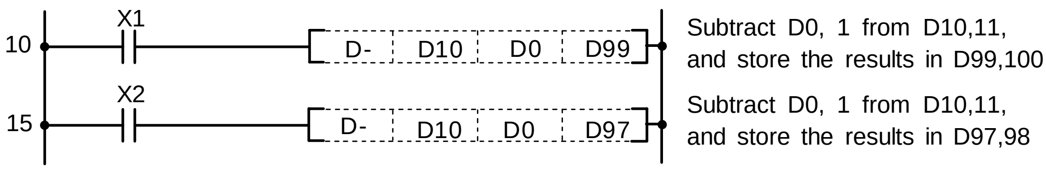
Execution conditions

The execution conditions for D-, D-P are as shown below.



Program example

(1) Program to subtract the D0, 1 data from the D10, 11 data when X1 turns ON, and output the results to D99, 100. Program to subtract the D0, 1 data from D10, 11 data when X2 turns ON, and output the results to D97, 98.

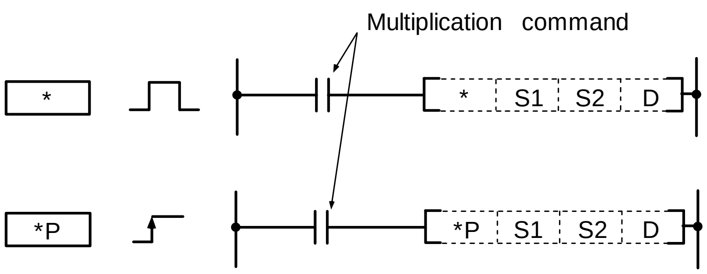


Coding

No. of steps	Com-mand	Device		
10	LD	X1		
11	D-	D10	D0	D99
15	LD	X2		
16	D-	D10	D0	D97
20				

○ *, *P ... BIN 16-bit multiplication

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1	○	○	○	○	○	○	○		○			○	○	○	○	○	○	○	○	○		○	○	4
S2	○	○	○	○	○	○	○		○			○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○			○	○	○	○	○	○	○						

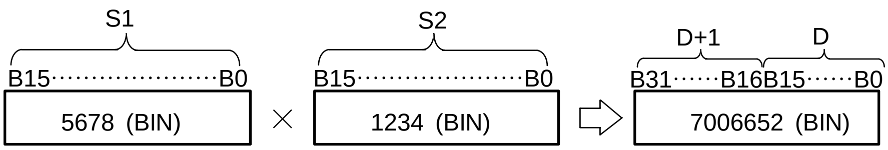


Setting data

S1	Multiplication data or No. of device where multiplication data is stored.
S2	Multiplication data or No. of device where multiplication data is stored.
D	Head No. of device to store multiplication results.

Function

(1) The BIN data designated with S1 and the BIN data designated with S2 are multiplied, and the multiplication results are stored in the device designated with D.



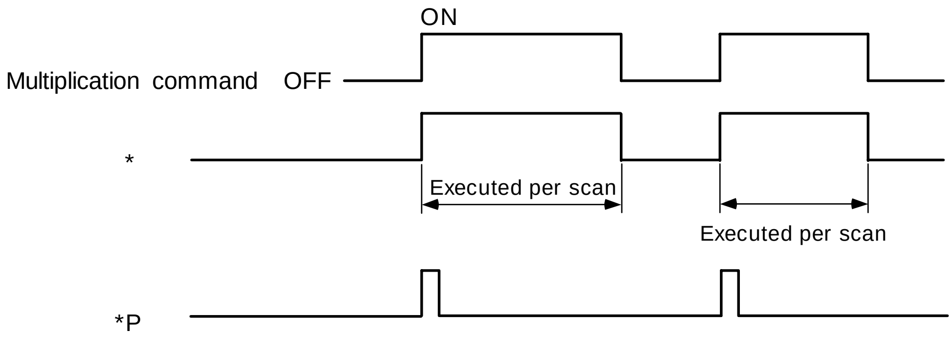
(2) -32768 to 32767 (BIN 16-bit) can be designated in S1 and S2.

(3) The positive/negative of the data in S1, S2 and D is determined with the highest-order bit (B15 for S1 and S2, B31 for D).

B15/B31	Judgment of positive/negative
0	Positive
1	Negative

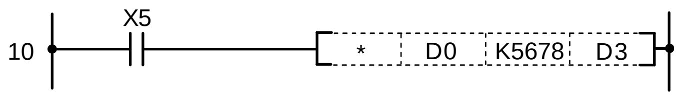
Execution conditions

The execution conditions for *, *P are as shown below.



Program example

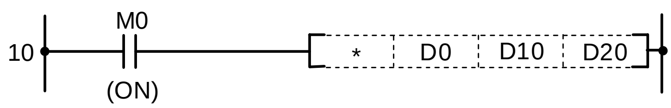
(1) Program to multiply the D0 data and BIN 5678 when X5 turns ON, and output the results to D3, 4.



Coding

No. of steps	Com-mand	Device		
10	LD	X5		
11	*	D0	K5678	D3
15				

(2) Program to multiple the D0 BIN data and D10 BIN data, and output the results to D20.

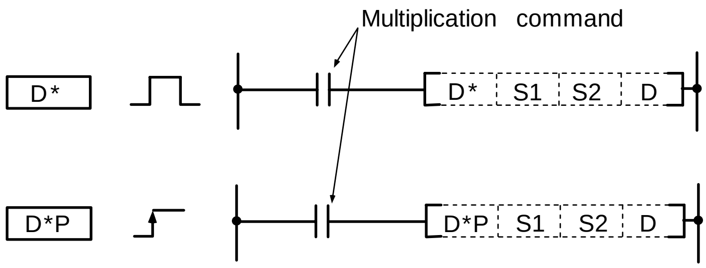


Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	*	D0	D10	D20
15				

○ D*, D*P ... BIN 32-bit multiplication

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1											○	○	○	○	○	○		○	○	○				5/6
S2											○	○	○	○	○	○		○	○	○				
D											○	○	○	○	○	○		○						

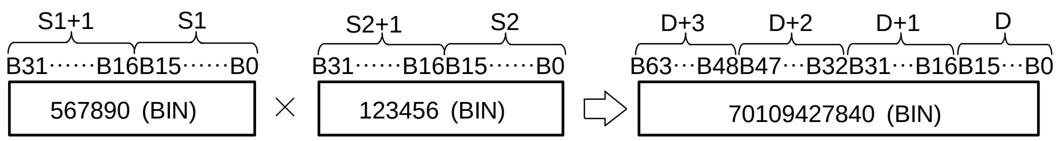


Setting data

S1	Multiplication data or head No. of device where multiplication data is stored.
S2	Multiplication data or head No. of device where multiplication data is stored.
D	Head No. of device to store multiplication results.

Function

(1) The BIN data designated with S1 and the BIN data designated with S2 are multiplied, and the multiplication results are stored in the device designated with D.



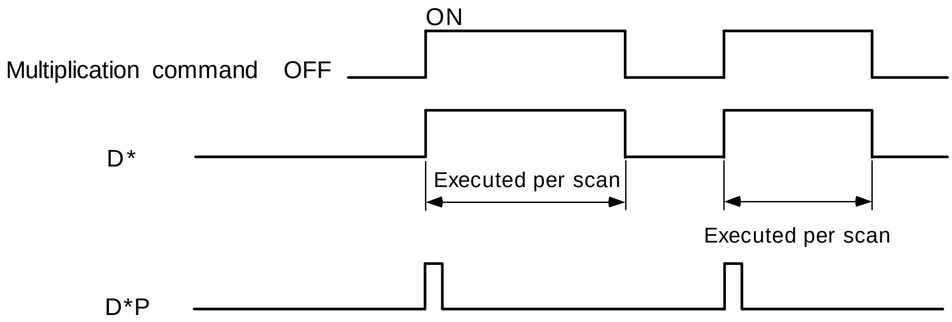
(2) -2147483648 to 2147483647 (BIN 32-bit) can be designated in S1 and S2.

(3) The positive/negative of the data in S1, S2 and D is determined with the highest-order bit (B31 for S1 and S2, B63 for D).

B31/B63	Judgment of positive/negative
0	Positive
1	Negative

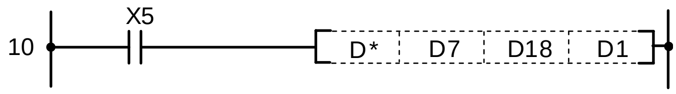
Execution conditions

The execution conditions for D*, D*P are as shown below.



Program example

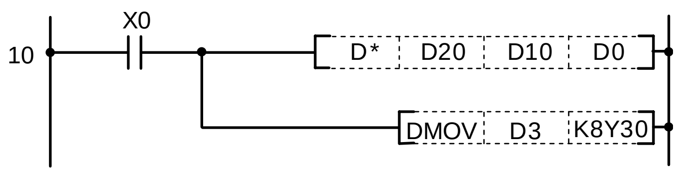
(1) Program to multiply the D7, 8 BIN data and D18, 19 BIN data when X5 turns ON, and output the results to D1 to 4.



Coding

No. of steps	Com-mand	Device		
10	LD	X5		
11	D*	D7	D18	D1
16				

(2) Program to multiply the D20 BIN data and D10 BIN data when X0 turns ON, and output the high-order 16-bit to Y30 to 4F.

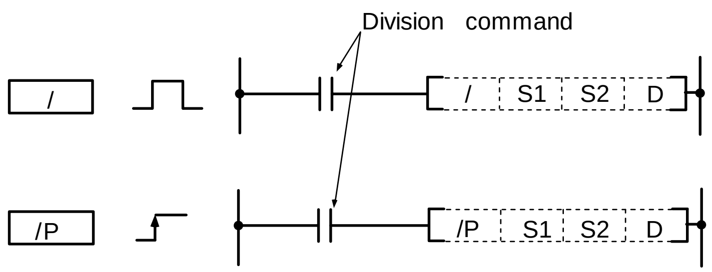


Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	D*	D20	D10	D0
16	DMOV	D3	K8Y30	
19				

○ /, /P ... BIN 16-bit division

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1												○	○	○	○	○		○	○	○				5/6
S2												○	○	○	○	○		○	○	○				
D												○	○	○	○	○		○						

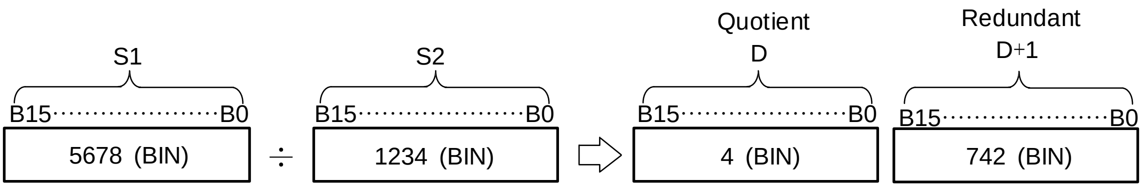


Setting data

S1	Division data or No. of device where division data is stored.
S2	Division data or No. of device where division data is stored.
D	Head No. of device to store division results.

Function

(1) The BIN data designated with S1 and the BIN data designated with S2 are divided, and the division results are stored in the device designated with D.



(2) -32768 to 32767 (BIN 16-bit) can be designated in S1 and S2.

(3) The positive/negative of the data in S1, S2 and D is determined with the highest-order bit (B15).

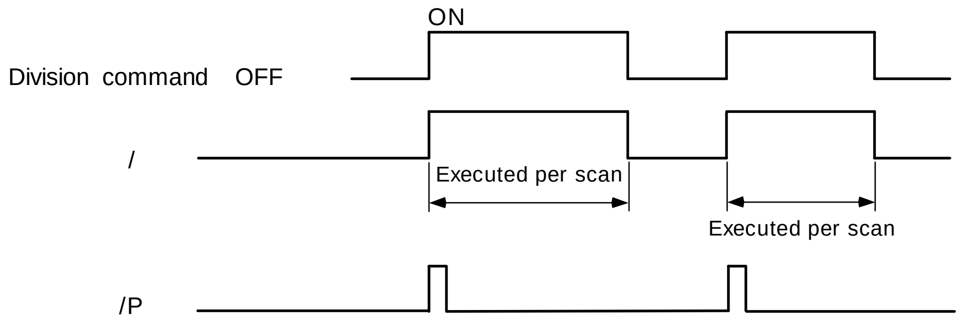
B15	Judgment of positive/negative
0	Positive
1	Negative

(4) For the word device, the operation results will be stored as quotient and redundant using the 32-bit.
Quotient ... Stored in low-order 16-bit.
Redundant... Stored in high-order 16-bit.

(5) The S1 and S2 data will not change even after operation is executed.

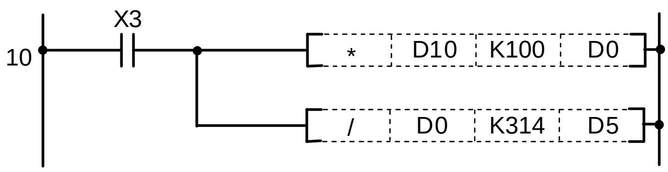
Execution conditions

The execution conditions for /, /P are as shown below.



Program example

(1) Program to divide the D10 data by 3.14 when X3 turns ON, and output the value (quotient) to D5.

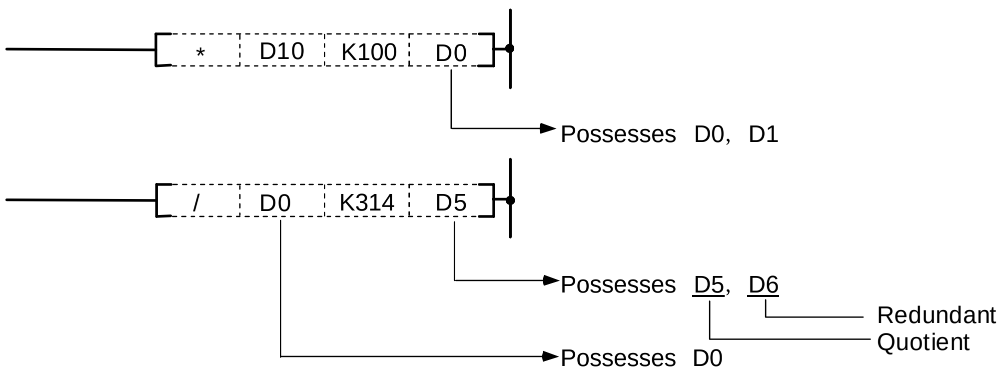


Coding

No. of steps	Com- mand	Device		
10	LD	X3		
11	*	D10	K100	D0
15	/	D0	K314	D5
20				

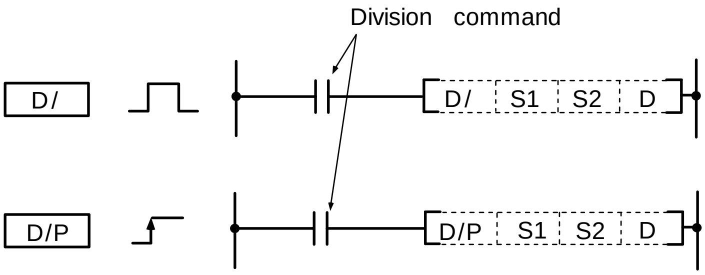
Point

The source and destination sides of the above program are as follow.



○ D/, D/P ... BIN 32-bit division

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1											○	○	○	○	○	○		○	○	○				5/6
S2											○	○	○	○	○	○		○	○	○				
D											○	○	○	○	○	○		○						

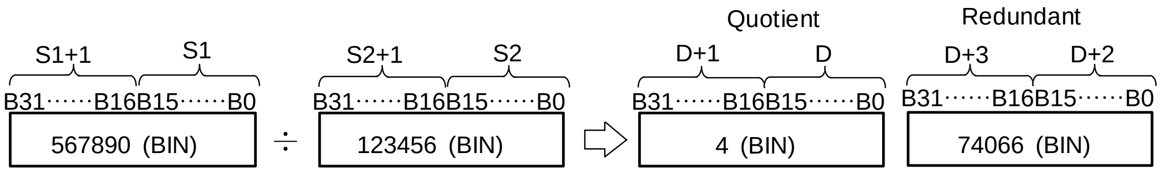


Setting data

S1	Division data or head No. of device where division data is stored.
S2	Division data or head No. of device where division data is stored.
D	Head No. of device to store division results.

Function

(1) The BIN data designated with S1 and the BIN data designated with S2 are divided, and the division results are stored in the device designated with D.



(2) -2147483648 to 2147483647 (BIN 32-bit) can be designated in S1 and S2.

(3) The positive/negative of the data in S1, S2 and D is determined with the highest-order bit (B31).

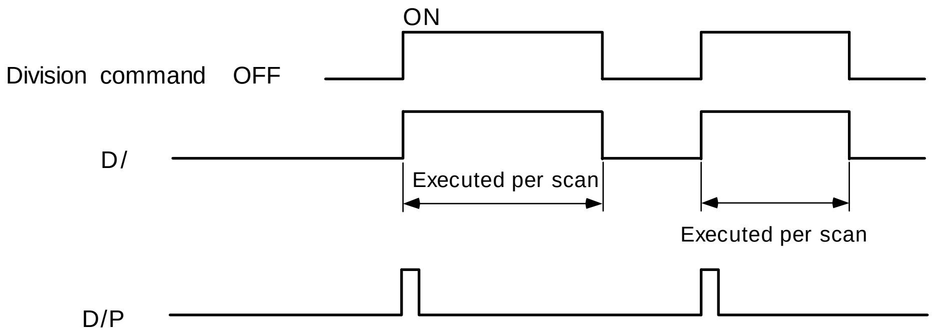
B31	Judgment of positive/negative
0	Positive
1	Negative

(4) For the word device, the operation results will be stored as quotient and redundant using the 64-bit.
Quotient ... Stored in low-order 32-bit.
Redundant... Stored in high-order 32-bit.

(5) The S1 and S2 data will not change even after operation is executed.

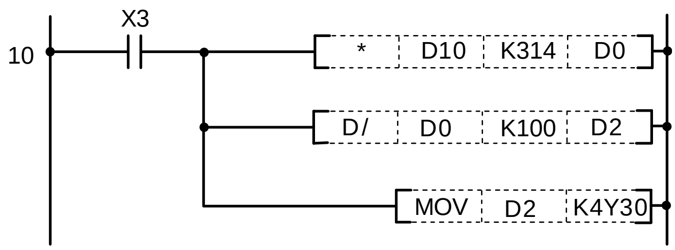
Execution conditions

The execution conditions for D/, D/P are as shown below.



Program example

(1) Program to multiply the D10 data by 3.14 when X3 turns ON, and output the low-order 16-bit of the results to Y30 to 3F.

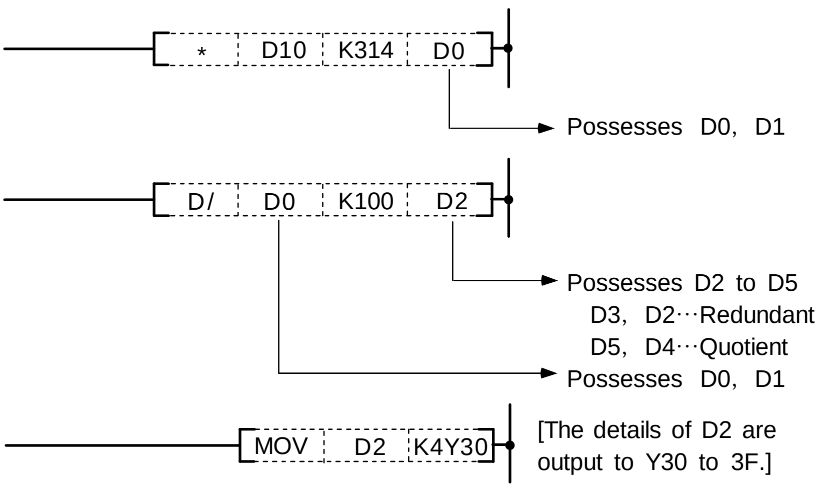


Coding

No. of steps	Com-mand	Device		
10	LD	X3		
11	*	D10	K314	D0
15	D/	D0	K100	D2
21	MOV	D2	K4Y30	
24				

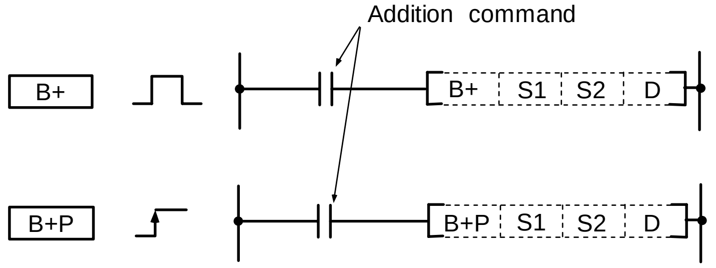
Point

The source and destination sides of the above program are as follow.



○ B+, B+P ... BCD 16-bit addition

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1											○	○	○	○	○	○		○	○	○			○	5/6
S2											○	○	○	○	○	○		○	○	○				
D											○	○	○	○	○	○		○						

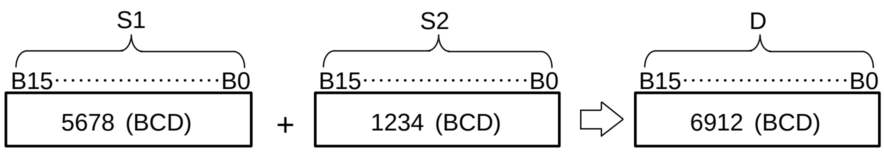


Setting data

S1	Addition data or No. of device where addition data is stored.
S2	Addition data or No. of device where addition data is stored.
D	No. of device to store addition results.

Function

(1) The BCD data designated with S1 and the BCD data designated with S2 are added, and the addition results are stored in the device designated with D.

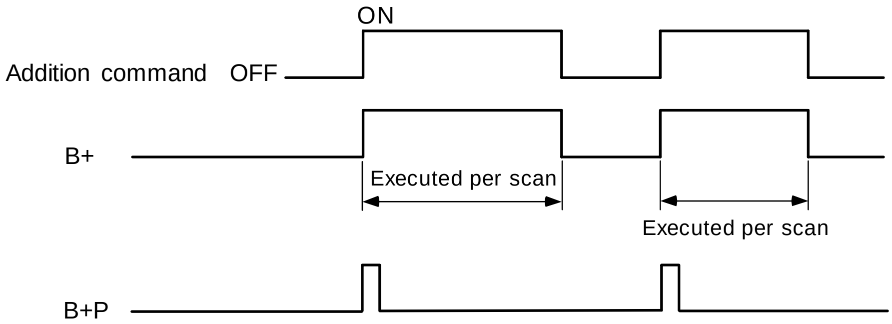


(2) 0 to 9999 (BCD 16-bit) can be designated in S1 and S2.
(3) If the result of the addition exceeds 9999, the higher bits are ignored. The carry flag in this case is not turned ON.

(Note 1) An error occurs if the S1, S2, or D BCD data is outside the range 0 to 9999 (BCD).

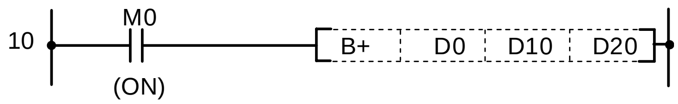
Execution conditions

The execution conditions for B+, B+P are as shown below.



Program example

(1) Program to add the D0 BCD data and D10 BCD data and output to D20.

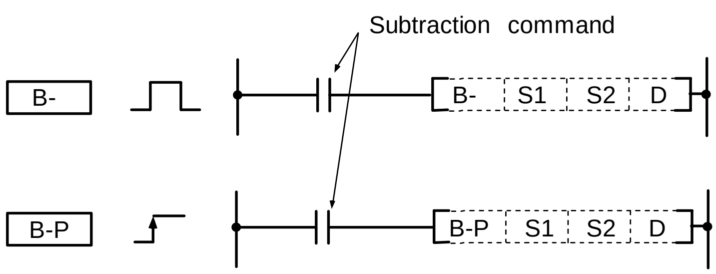


Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	B+	D0	D10	D20
16				

○ B–, B–P ... BCD 16-bit subtraction

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1											○	○	○	○	○	○		○	○	○			○	5/6
S2											○	○	○	○	○	○		○	○	○				
D											○	○	○	○	○	○		○						

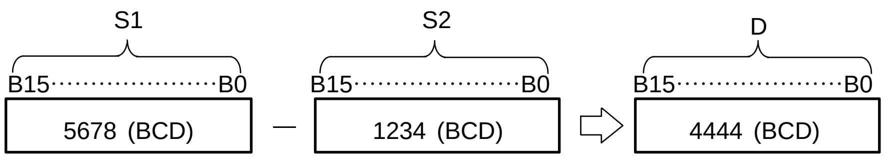


Setting data

S1	Subtraction data or No. of device where subtraction data is stored.
S2	Subtraction data or No. of device where subtraction data is stored.
D	No. of device to store subtraction results.

Function

(1) The device designated with S1 and the device designated with S2 are subtracted, and the subtracted results are stored in the device designated with D.



(2) 0 to 9999 (BCD 16-bit) can be designated in S1 and S2.

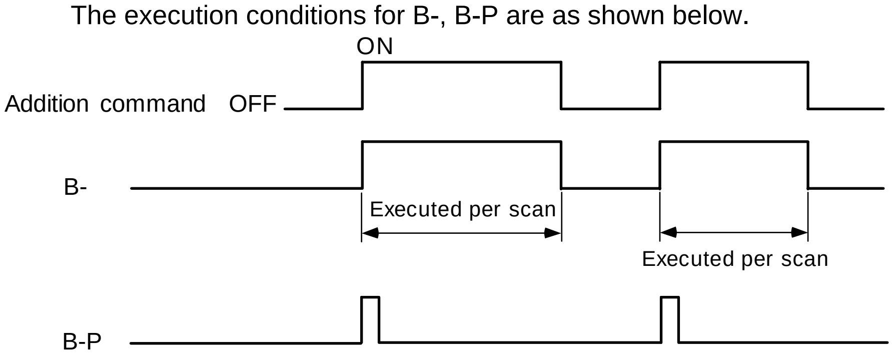
(3) The following will result if an underflow is generated by the subtraction. The carry flag in this case is not turned ON.

0	0	0	1	-	0	0	0	3	⇨	9	9	9	8

(Note 1) An error occurs if the S1, S2, or D BCD data is outside the range 0 to 9999 (BCD).

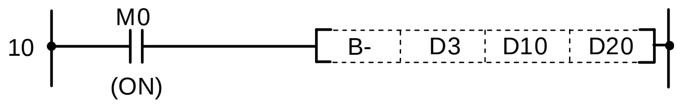
Execution conditions

The execution conditions for B-, B-P are as shown below.



Program example

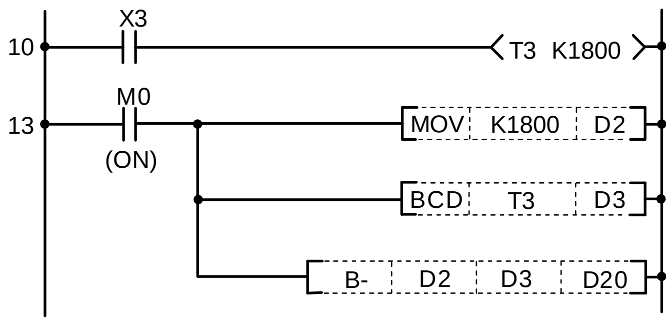
(1) Program to subtract the D10 BCD data from D3 and output to D20.



Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	B-	D3	D10	D20
16				

(2) Program to output the difference of the timer T3 setting value and current value to D20 BCD data.

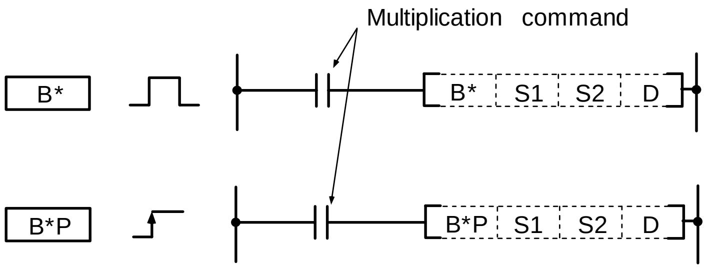


Coding

No. of steps	Com-mand	Device		
10	LD	X3		
11	OUT	T3	K1800	
13	LD	M0		
14	MOV	K1800	D2	
17	BCD	T3	D3	
20	B-	D2	D3	D20
25				

○ B*, B*P ... BCD 16-bit multiplication

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1											○	○	○	○	○	○		○	○	○			○	5/6
S2											○	○	○	○	○	○		○	○	○				
D											○	○	○	○	○	○		○						

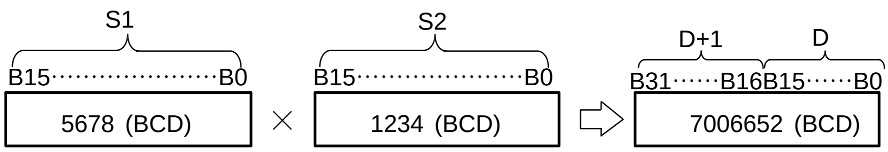


Setting data

S1	Multiplication data or No. of device where multiplication data is stored.
S2	Multiplication data or No. of device where multiplication data is stored.
D	Head No. of device to store multiplication results.

Function

(1) The BCD data designated with S1 and the BCD data designated with S2 are multiplied, and the multiplication results are stored in the device designated with D.

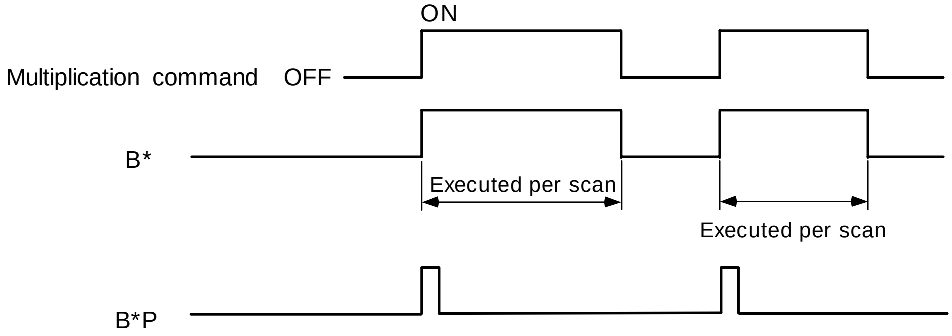


(2) 0 to 9999 (BCD 16-bit) can be designated in S1 and S2.

(Note 1) An error occurs if the S1, or S2 BCD data is outside the range 0 to 9999 (BCD).

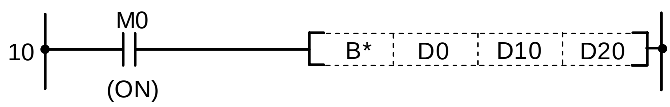
Execution conditions

The execution conditions for B*, B*P are as shown below.



Program example

(1) Program to multiple the D0 BCD data and D10 BCD data, and output the results to D20.

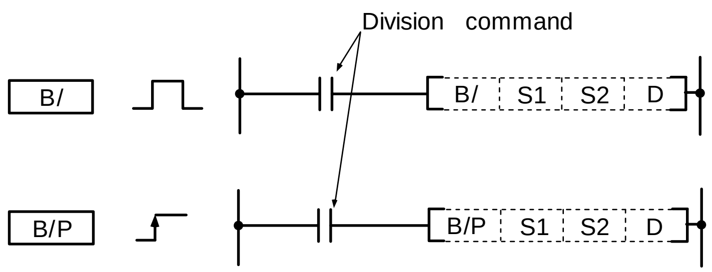


Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	B*	D0	D10	D20
16				

○ B/, B/P ... BCD 16-bit division

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1											○	○	○	○	○	○		○	○	○			○	5/6
S2											○	○	○	○	○	○		○	○	○				
D											○	○	○	○	○	○		○						

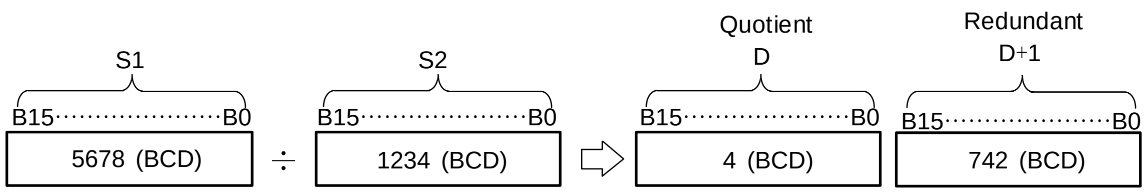


Setting data

S1	Division data or No. of device where division data is stored.
S2	Division data or No. of device where division data is stored.
D	Head No. of device to store division results.

Function

(1) The BCD data designated with S1 and the BCD data designated with S2 are divided, and the division results are stored in the device designated with D.



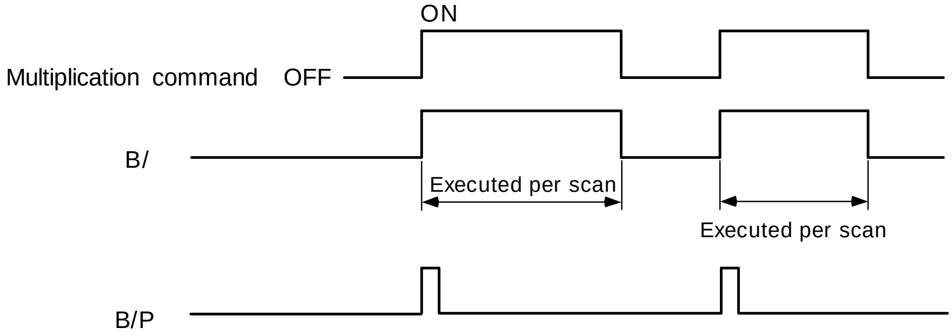
(2) 0 to 9999 (BCD 16-bit) can be designated in S1.
1 to 9999 (BCD 16-bit) can be designated in S2.

(3) For the word device, the operation results will be stored as quotient and redundant using the 32-bit.
Quotient ... Stored in low-order 16-bit.
Redundant... Stored in high-order 16-bit.

(Note 1) An error occurs if the S1, or S2 BCD data is outside the range 0 to 9999 (BCD).
(Note 2) An error occurs if S2 BCD data is "0".

Execution conditions

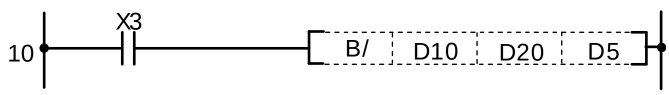
The execution conditions for B/, B/P are as shown below.



Program example

(1) Program to divide the D10 BCD data D20 when X3 turns ON, and output the value (quotient) to D5.

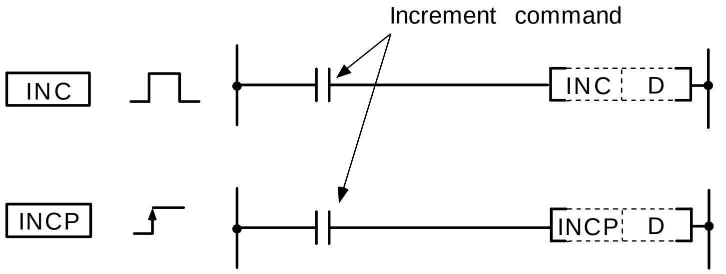
Coding



No. of steps	Com-mand	Device		
10	LD	X3		
11	B/	D10	D20	D5
16				

○ INC, INCP ... (16-bit BIN data) +1

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	2

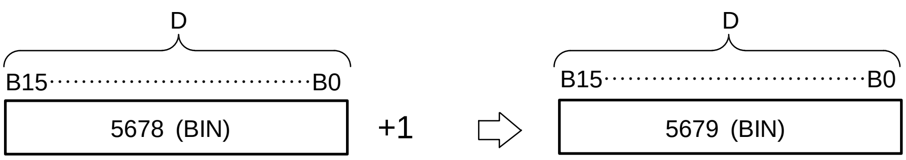


Setting data

D	No. of device to be incremented (+1)

Function

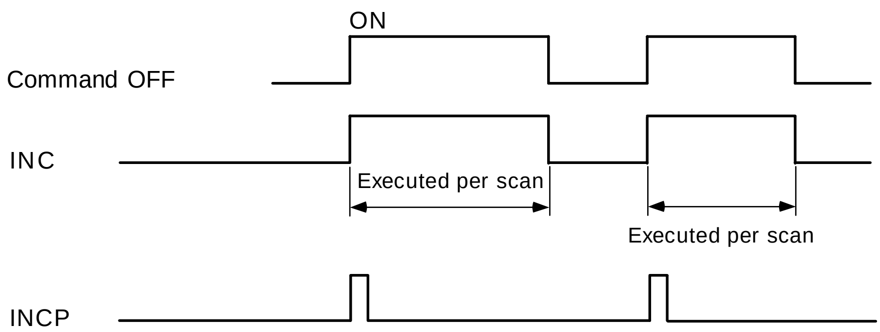
(1) The device (16-bit data) designated with D is incremented by one.



(2) If INC or INCP is executed when the data of the device designated with D is 32767, -32768 will be stored in the device designated with D.

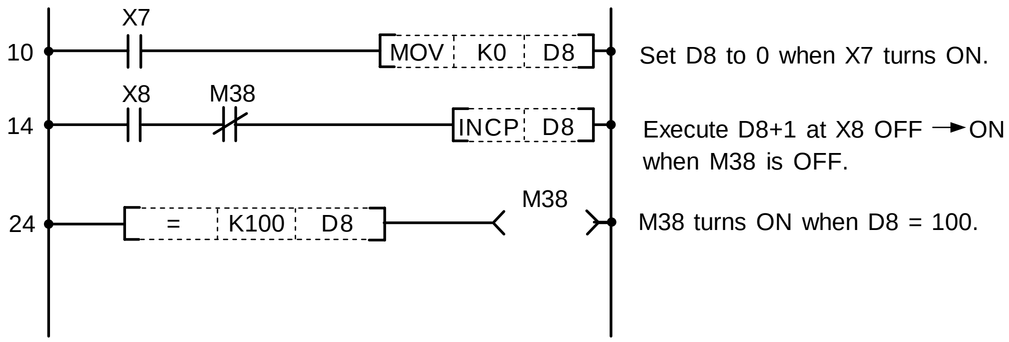
Execution conditions

The execution conditions for the INC, INCP commands are as shown below.



Program example

(1) Example of addition counter program

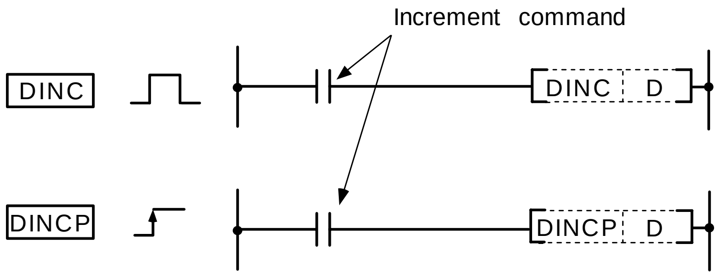


Coding

No. of steps	Com-mand	Device		
10	LD	X7		
11	MOV	K0	D8	
14	LD	X8		
15	ANI	M38		
16	INCP	D8		
24	LD=	K100	D8	
27	OUT	M38		
28				

○ DINC, DINCP ... (32-bit BIN data) +1

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	2

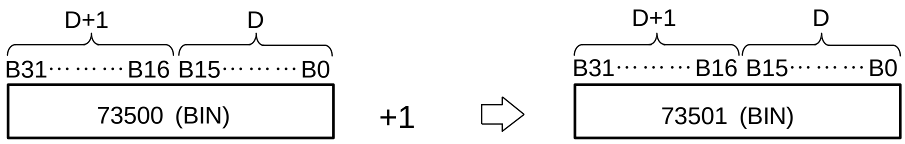


Setting data

D	Head No. of device to be incremented (+1)

Function

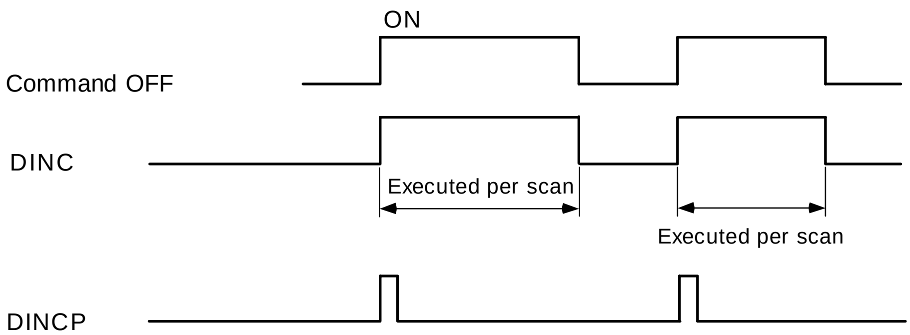
(1) The device (32-bit data) designated with D is incremented by one.



(2) If DINC or DINCP is executed when the data of the device designated with D is 2147483647, -2147483648 will be stored in the device designated with D.

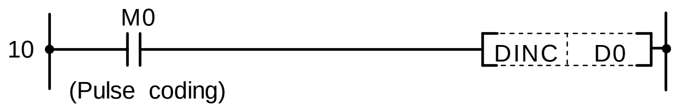
Execution conditions

The execution conditions for the DINC, DINCP commands are as shown below.



Program example

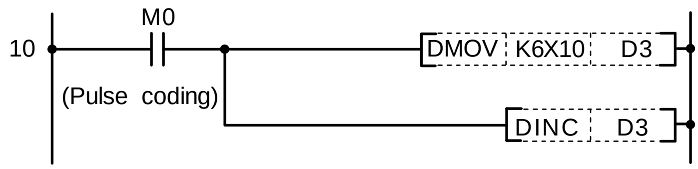
(1) Program to increment the D0, 1 data by one when M0 turns ON.



Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	DINC	D0		
13				

(2) Program to increment X10 to X27 data by one when M0 turns ON, and to store the results in D3, 4.

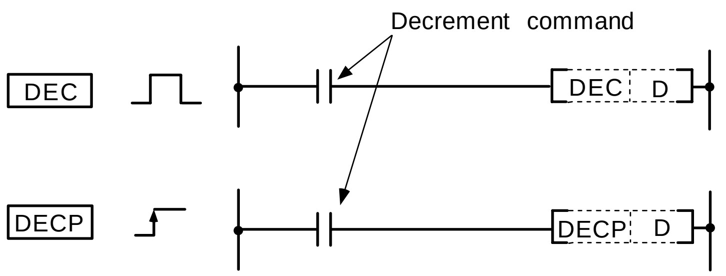


Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	DMOV	K6X10	D3	
14	DINC	D3		
16				

○ DEC, DECP ... (16-bit BIN data) –1

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	2

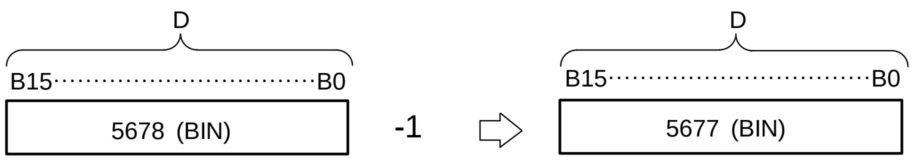


Setting data

D	No. of device to be decremented (-1)

Function

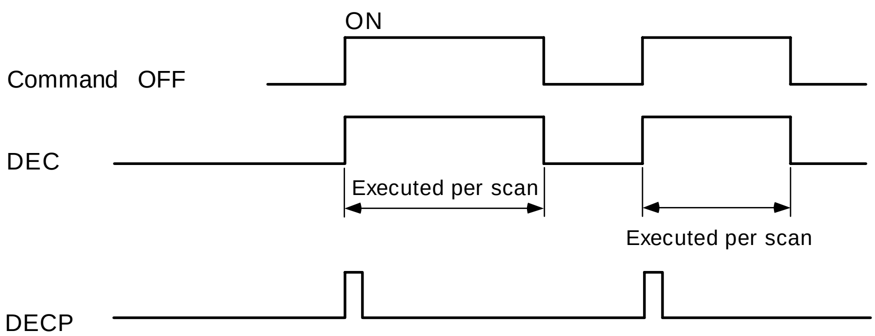
(1) The device (16-bit data) designated with D is decremented by one.



(2) If DEC or DECP is executed when the data of the device designated with D is 0, -1 will be stored in the device designated with D.

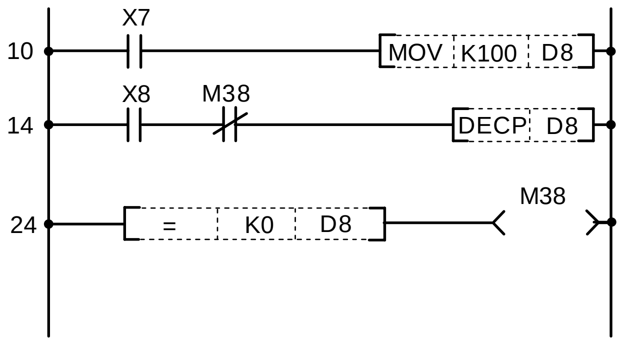
Execution conditions

The execution conditions for the DEC, DECP commands are as shown below.



Program example

(1) Example of subtraction counter program



Set D8 to 100 when X7 turns ON.

Execute D8-1 at X8 OFF→ ON when M38 turns OFF.

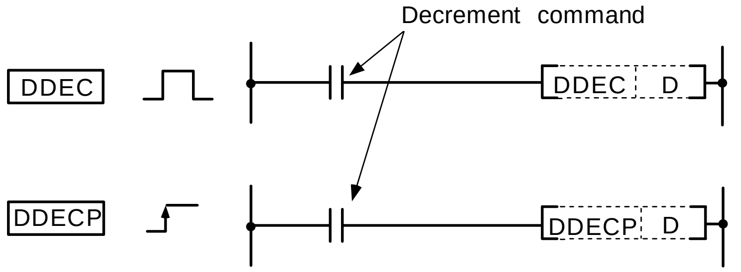
M38 turns ON when D8 = 0.

Coding

No. of steps	Com-mand	Device		
10	LD	X7		
11	MOV	K100	D8	
14	LD	X8		
15	ANI	M38		
16	DECP	D8		
24	LD=	K0	D8	
27	OUT	M38		
28				

○ **DDEC, DDECP ... (32-bit BIN data) –1**

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	2

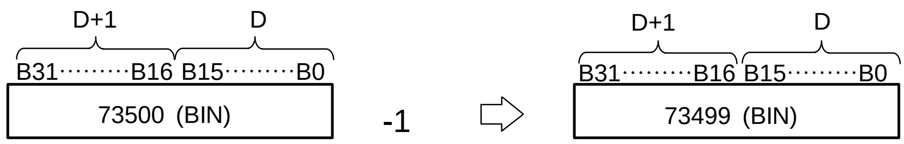


Setting data

D	Head No. of device to be decremented (-1)

Function

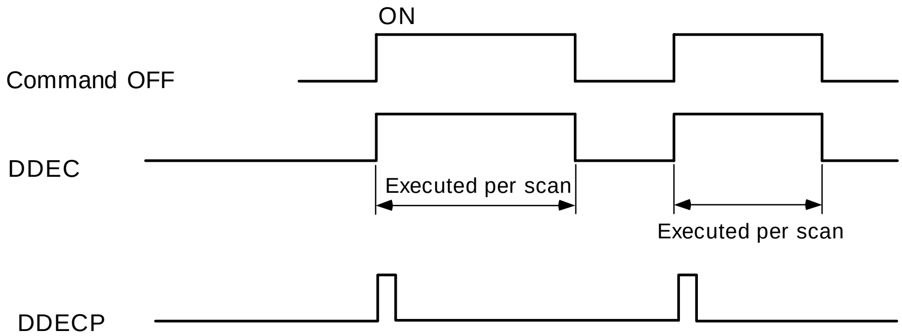
(1) The device (32-bit data) designated with D is decremented by one.



(2) If DDEC or DDECP is executed when the data of the device designated with D is 0, -1 will be stored in the device designated with D.

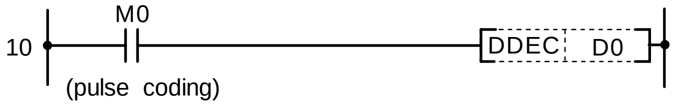
Execution conditions

The execution conditions for the DDEC, DDECP commands are as shown below.



Program example

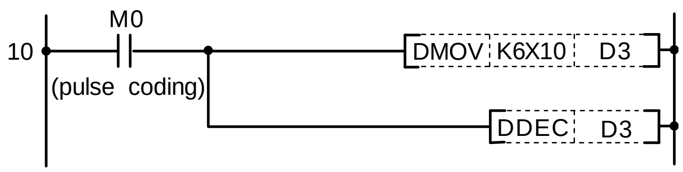
(1) Program to decrement the D0, 1 data by one when M0 turns ON.



Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	DDEC	D0		
13				

(2) Program to decrement X10 to X27 data by one when M0 turns ON, and to store the results in D3, 4.

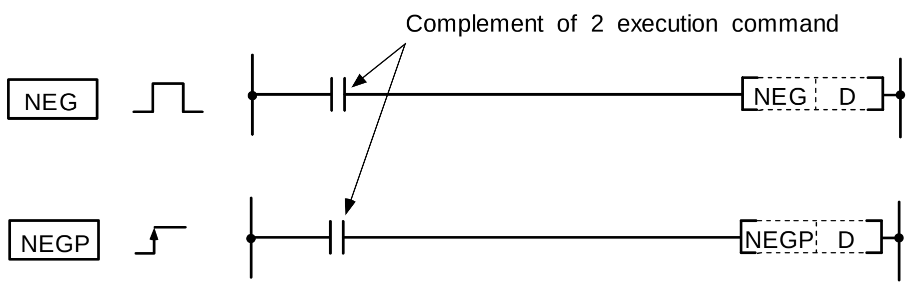


Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	DMOV	K6X10	D3	
14	DDEC	D3		
16				

○ NEG, NEGP ... Complement of 2 (BIN 16-bit data)

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○					○	2



Setting data

D	No. of device where data to be complemented by 2 is stored.

Function

(1) The 16-bit data of the device designated with D is reversed and incremented by one, and then stored in the device designated with D.

16 bit

Before execution D	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

↓ Reversal

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

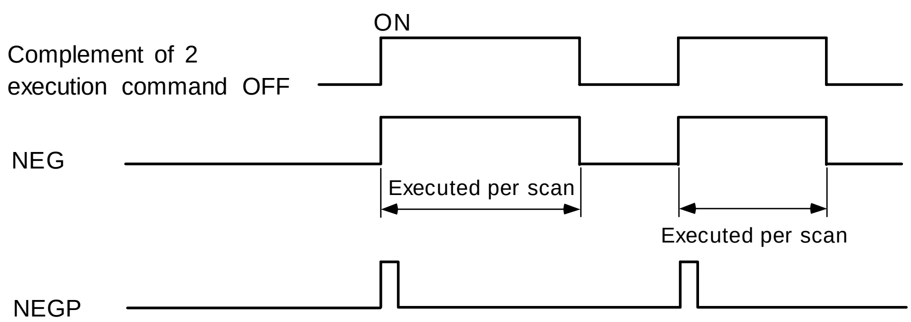
↓ +1

After execution D	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0

(2) This is used to use a negative BIN value as an absolute value.

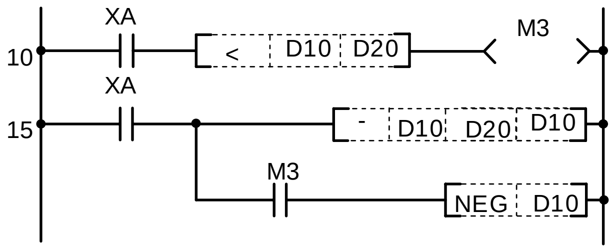
Execution conditions

The execution conditions for NEG, NEGD commands are as follow.



Program example

(1) Program to subtract D20 from D10 when XA turns ON and obtain an absolute value when the results are negative.



M3 turns ON when D10 < D20

D10-D20 is executed.

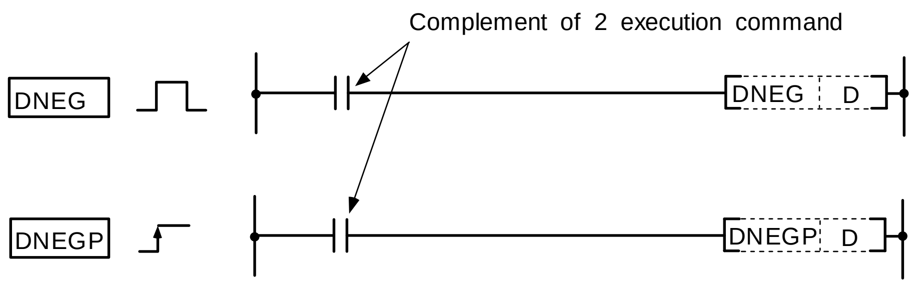
The absolute value (complement of 2) when M3 turns ON is obtained.

Coding

No. of steps	Com- mand	Device		
10	LD	XA		
11	AND<	D10	D20	
14	OUT	M3		
15	LD	XA		
16	-	D10	D20	D10
20	AND	M3		
21	NEG	D10		
23				

○ DNEG, DNEGP ... Complement of 2 (BIN 32-bit data)

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○					○	2



Setting data

D	Head No. of device where data to be complemented by 2 is stored.

Function

(1) The 32-bit data of the device designated with D is reversed and incremented by one, and then stored in the device designated with D.

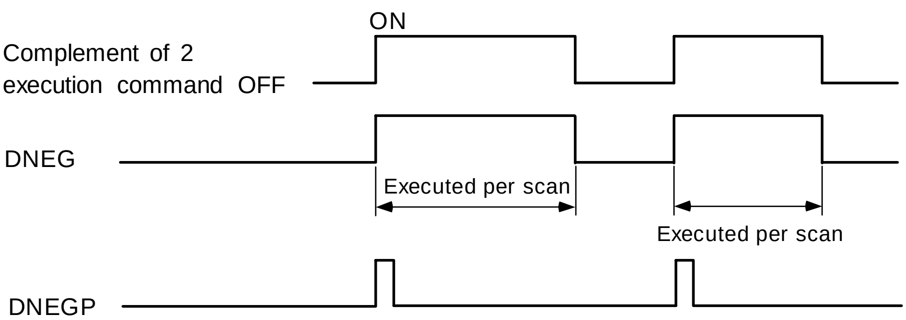
32 bit

Before execution D	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
↓ Reversal	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
↓ +1 After execution D	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

(2) This is used to use a negative BIN value as an absolute value.

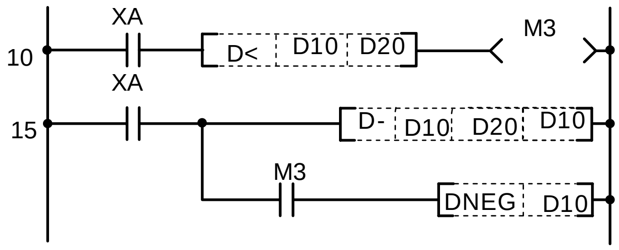
Execution conditions

The execution conditions for DNEG, DNEGP commands are as follow.



Program example

(1) Program to subtract D20 from D10 when XA turns ON and obtain an absolute value when the results are negative.



M3 turns ON when D10, 11 < D20, 21

Subtracts D20, 21 from D10, 11.

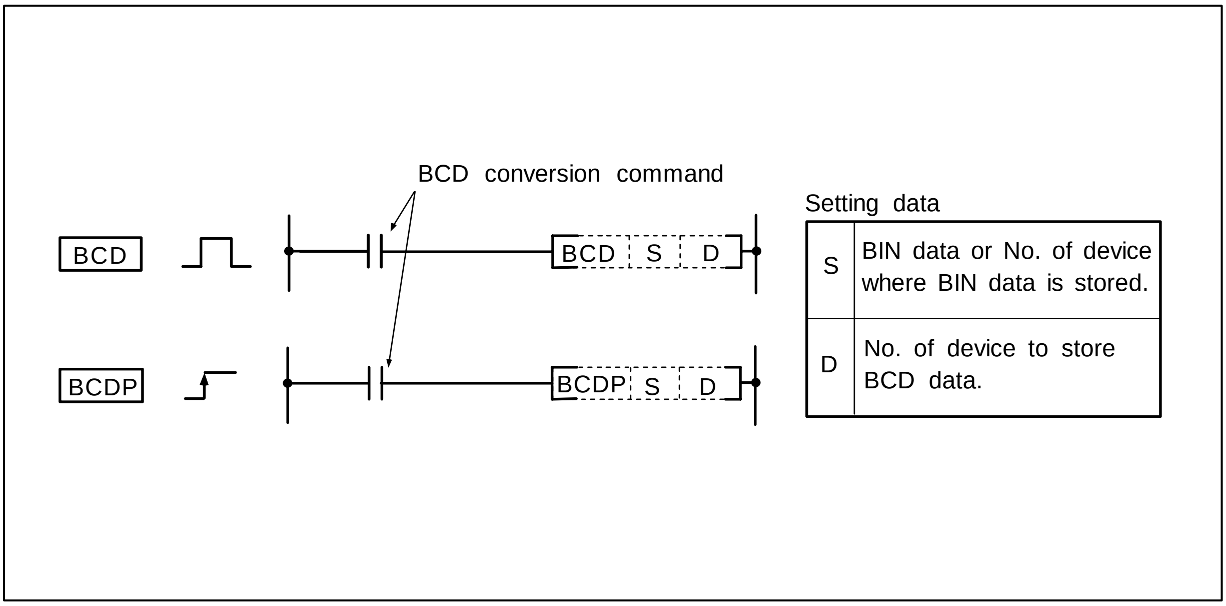
The absolute value (complement of 2) when M3 turns ON is obtained.

Coding

No. of steps	Com-mand	Device		
10	LD	XA		
11	ANDD<	D10	D20	
14	OUT	M3		
15	LD	XA		
16	D-	D10	D20	D10
20	AND	M3		
21	DNEG	D10		
23				

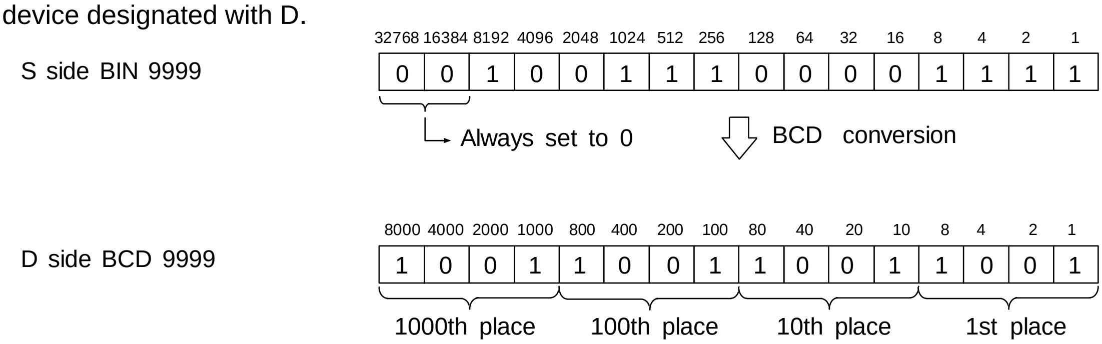
○ **BCD, BCDP ... BIN → BCD conversion (16-bit)**

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Function

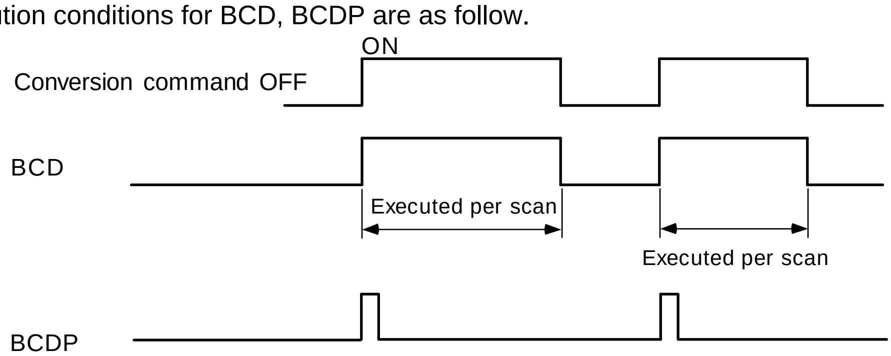
The BIN data (0 to 9999) of the device designated with S is BCD converted and transmitted to the device designated with D.



(Note 1) If the BIN data designated by S is not within 0 to 9999, it will not be converted correctly.

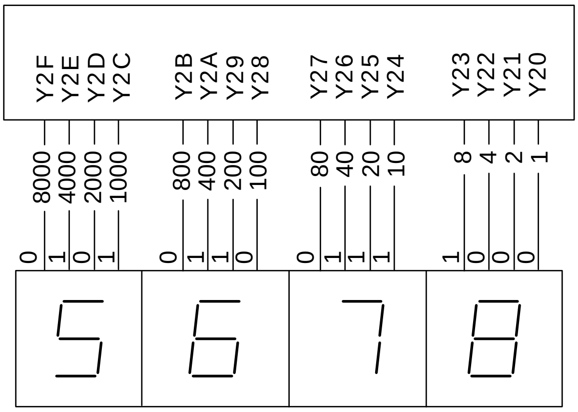
Execution conditions

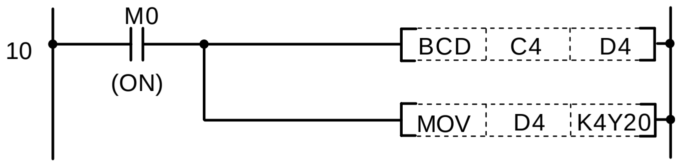
The execution conditions for BCD, BCDP are as follow.



Program example

(1) Program to output C4 current value from Y20 to Y2F to BCD display.



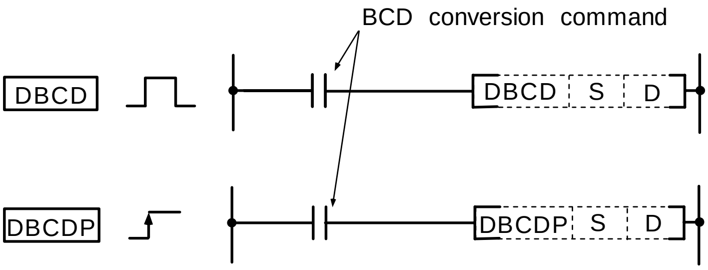


Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	BCD	C4	D4	
14	MOV	D4	K4Y20	
17				

○ DBCD, DBCDP ... BIN → BCD conversion (32-bit)

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○				○		○	○	○	○	○	○		○	○	○		○	○	4/5
D											○	○	○	○	○	○		○						

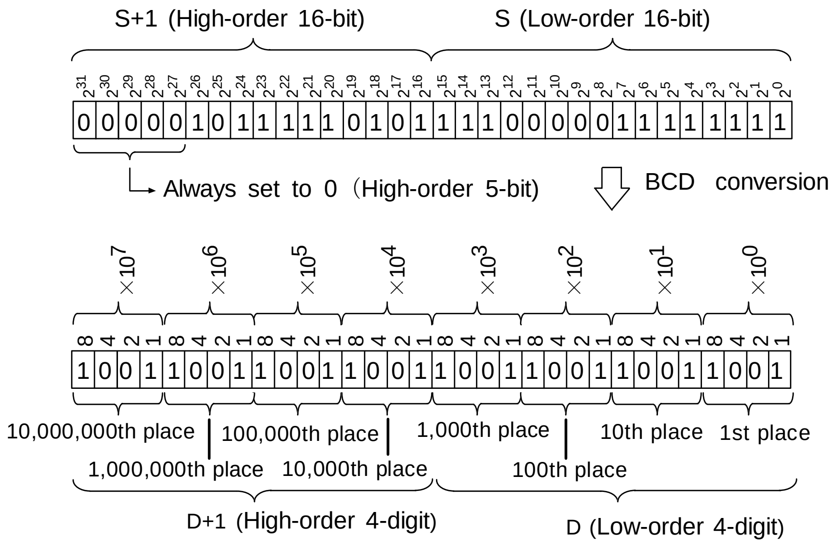


Setting data

S	BIN data or head No. of device where BIN data is stored.
D	Head No. of device to store BCD data.

Function

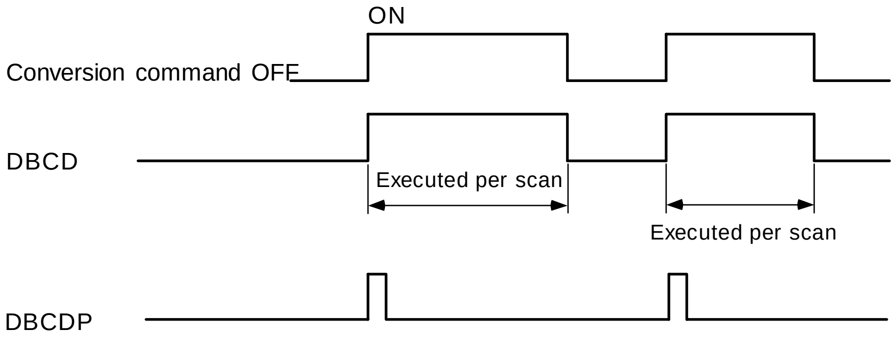
The BIN data (0 to 99999999) of the device designated with S is BCD converted and transmitted to the device designated with D.



(Note 1) If the BIN data designated by S is not within 0 to 99999999, it will not be converted correctly.

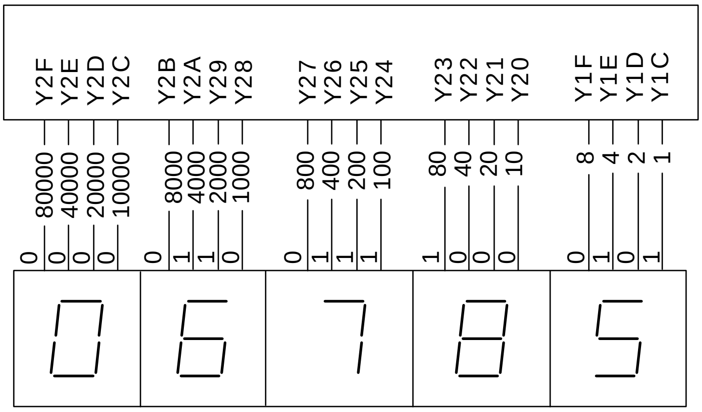
Execution conditions

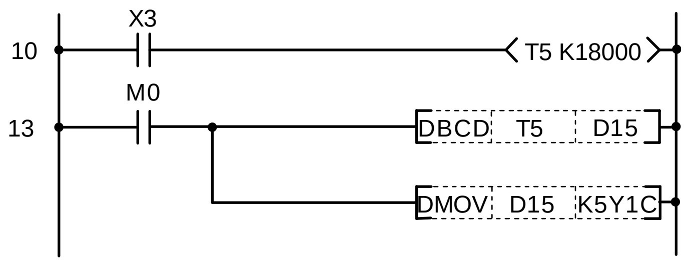
The execution conditions for DBCD, DBCDP are as follow.



Program example

(1) Program to output the current timer value of which the setting value exceeds 9999 to Y1C to 2F.



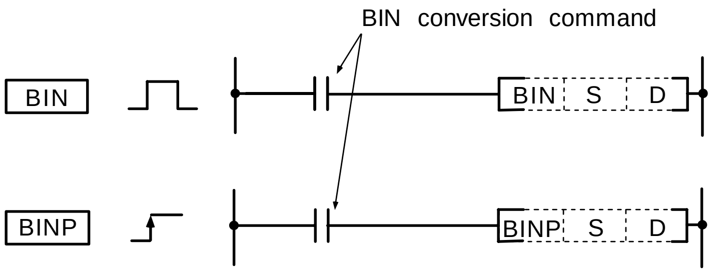


Coding

No. of steps	Com-mand	Device		
10	LD	X3		
11	OUT	T5	K18000	
13	LD	M0		
14	DBCD	T5	D15	
18	DMOV	D15	K5Y1C	
21				

○ BIN, BINP ... BCD → BIN conversion (16-bit)

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Setting data

S	BCD data or No. of device where BCD data is stored.
D	No. of device to store BIN data

Function

The BCD data (0 to 9999) of the device designated with S is BIN converted and transmitted to the device designated with D.

S side BCD 9999

8000	4000	2000	1000	800	400	200	100	80	40	20	10	8	4	2	1
1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1

1000th place　100th place　10th place　1st place

⇩ BIN conversion

D side BIN 9999

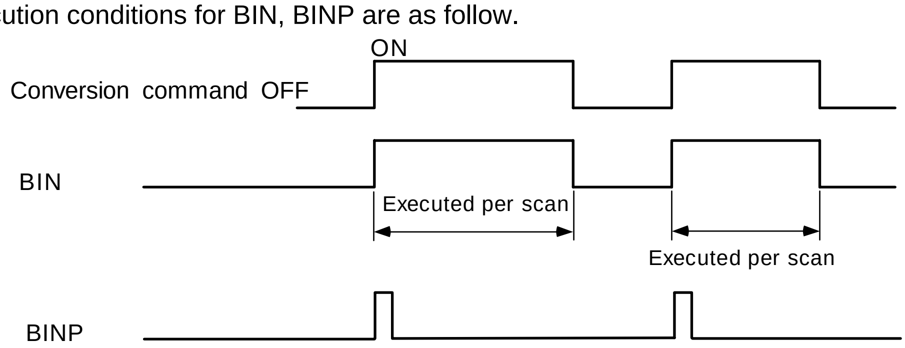
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1

└→ Set to 0

(Note 1) If the data designated by S is not within 0 to 9999, it will not be converted correctly.

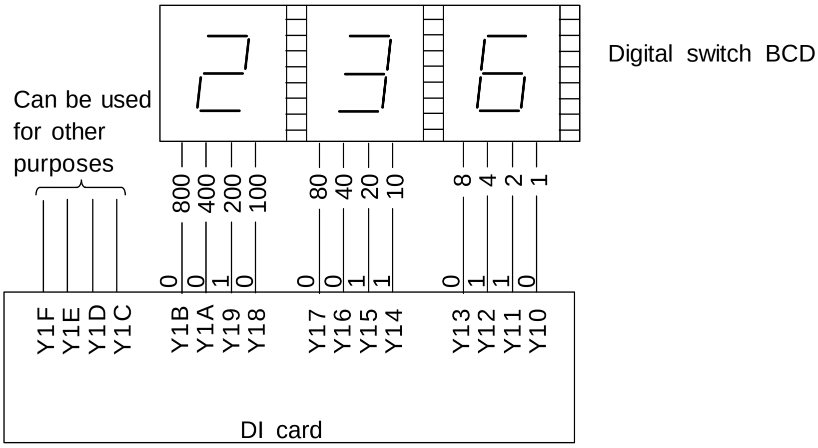
Execution conditions

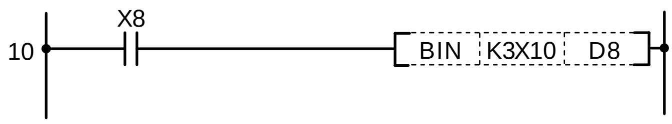
The execution conditions for BIN, BINP are as follow.



Program example

(1) Program to BIN convert the X10 to X1B BCD data when X8 turns On, and store in D8.



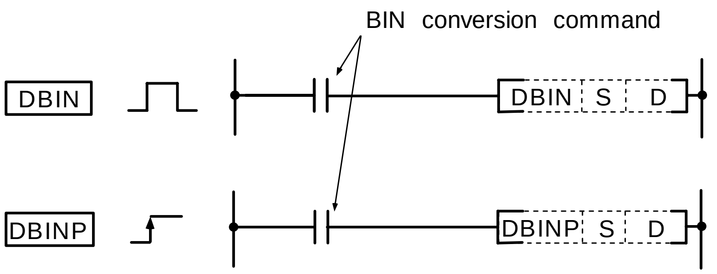


Coding

No. of steps	Com-mand	Device		
10	LD	X8		
11	BIN	K3X10	D8	
14				

○ DBIN, DBINP ... BCD → BIN conversion (32-bit)

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○				○		○	○	○	○	○	○		○	○	○		○	○	4/5
D											○	○	○	○	○	○		○						



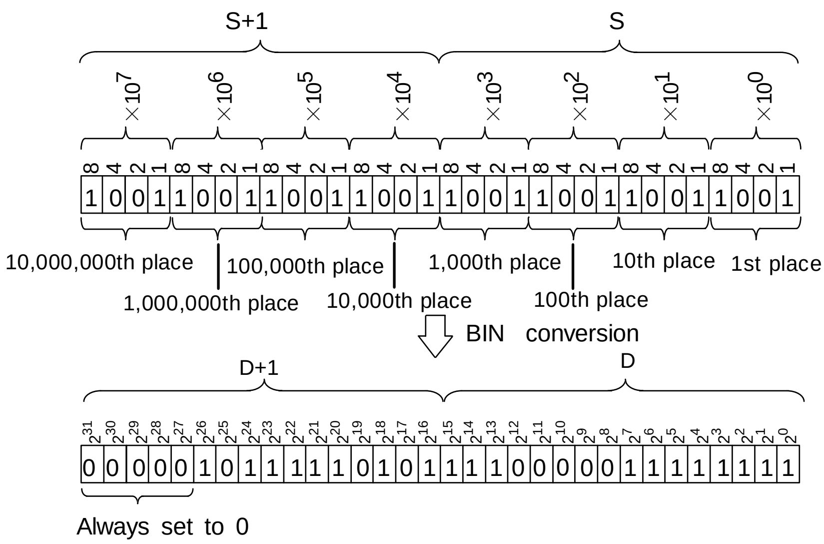
Setting data

S	BCD data or head No. of device where BCD data is stored.
D	Head No. of device to store BIN data

Function

The BCD data (0 to 99999999) of the device designated with S is BIN converted and transmitted to the device designated with D.

S side BCD 99999999

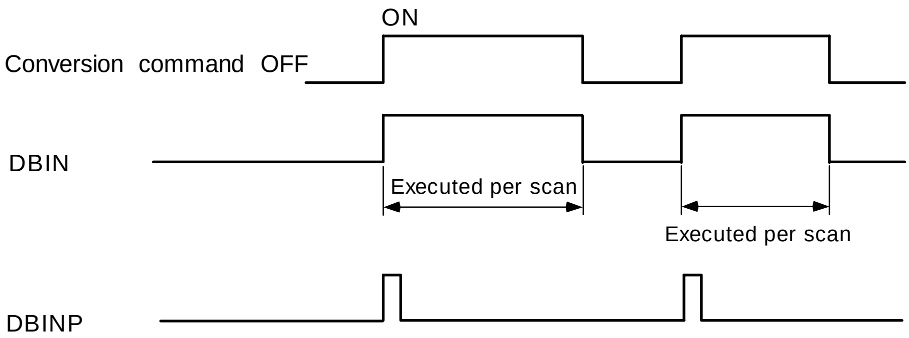


D side BIN 99999999

(Note 1) If the data designated by S is not within 0 to 99999999, it will not be converted correctly.

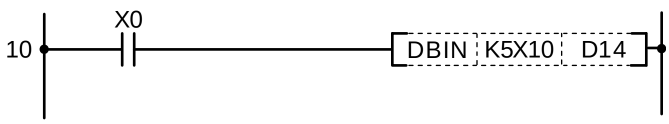
Execution conditions

The execution conditions for DBIN, DBINP are as follow.



Program example

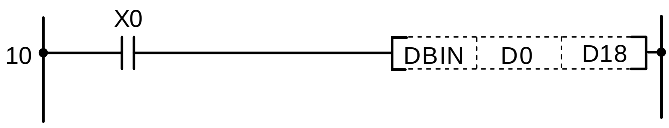
(1) Program to BIN convert the X10 to X23 BCD data when X0 turns ON, and to store in D14, 15.



Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	DBIN	K5X10	D14	
15				

(2) Program to BIN convert the D0, 1 data when X0 turns ON, and store in D18, 19.



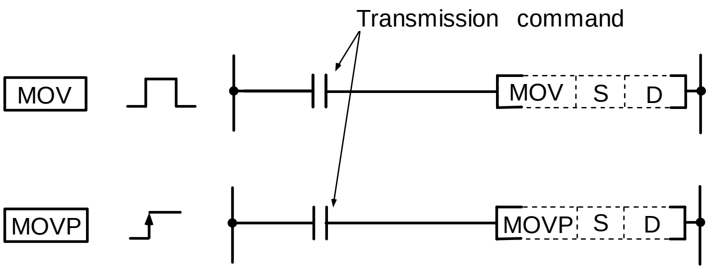
Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	DBIN	D0	D18	
15				

○ MOV, MOVP ... 16-bit data transmission

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
D	*1	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						

*1 Destination device X of MOV command is reserved for the test by Mitsubishi.
Be sure not to use this device with MOV even if it is possible to program.



Setting data

S	Transmission source data or No. of device where data is stored.
D	No. of transmission destination device.

Function

The 16-bit data of the device designated with S is transmitted to the device designated with D.

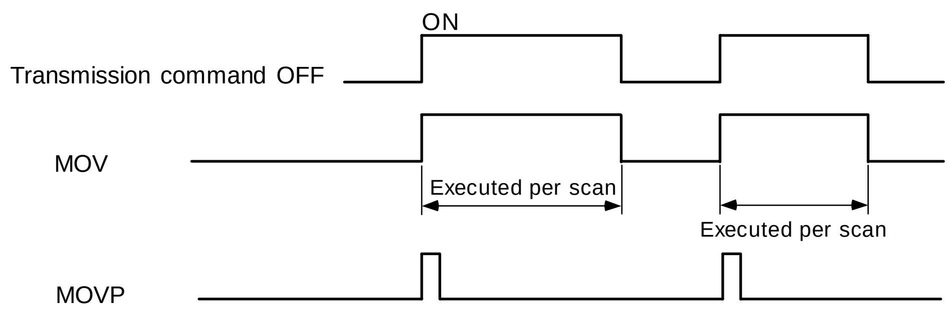
16-bit

Before transmission S	1	0	0	1	0	1	1	0	0	1	1	0	0	1	1	1
After transmission D	1	0	0	1	0	1	1	0	0	1	1	0	0	1	1	1

Transmission

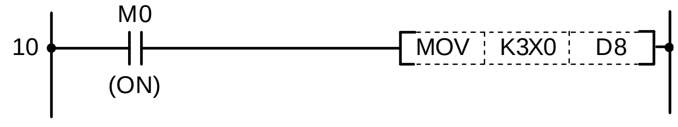
Execution conditions

The execution conditions for MOV, MOVP are as shown below.



Program example

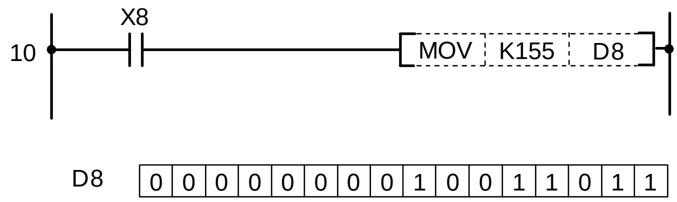
(1) Program to store input X0 to XB data in D8.



Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	MOV	K3X0	D8	
14				

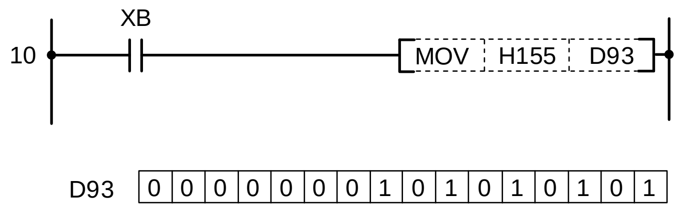
(2) Program to store "155" in D8 as binary value when X8 turns ON.



Coding

No. of steps	Com-mand	Device		
10	LD	X8		
11	MOV	K155	D8	
14				

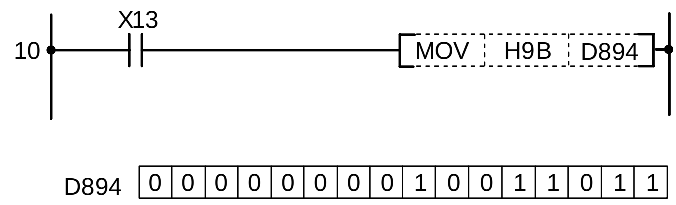
(3) Program to store "155" in D93 as BCD value in when XB turns ON.



Coding

No. of steps	Com-mand	Device		
10	LD	XB		
11	MOV	H155	D93	
14				

(4) Program to store "155" in D894 as hexadecimal (HEX) when X13 turns ON.



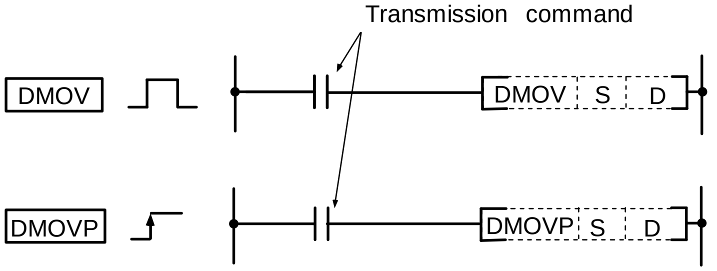
Coding

No. of steps	Com-mand	Device		
10	LD	X13		
11	MOV	H9B	D894	
14				

○ DMOV, DMOVP ... 32-bit data transmission

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3/4
D	*1	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						

*1 Destination device X of DMOV command is reserved for the test by Mitsubishi.
Be sure not to use this device with DMOV even if it is possible to program.

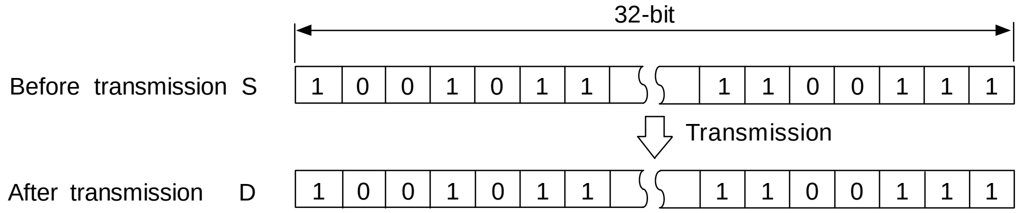


Setting data

S	Transmission source data or head No. of device where data is stored.
D	Head No. of transmission destination device.

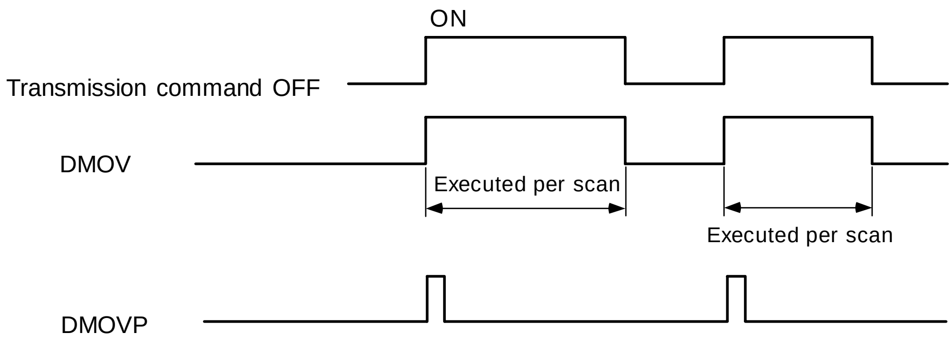
Function

The 32-bit data of the device designated with S is transmitted to the device designated with D.



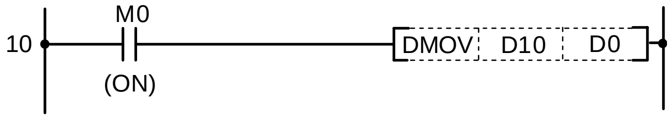
Execution conditions

The execution conditions for DMOV, DMOVP are as shown below.



Program example

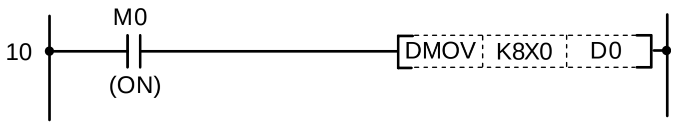
(1) Program to store D10, 11 data in D0, D1.



Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	DMOV	D10	D0	
14				

(2) Program to store X0 to X1F data in D0, D1.

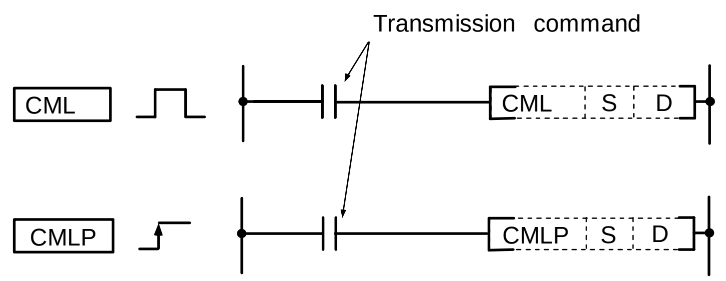


Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	DMOV	K8X0	D0	
14				

○ CML, CMLP ... 16-bit data negation transmission

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Setting data

S	Data to be inverted or No. of device where data is stored.
D	No. of device to store inversion results

Function

The 16-bit data designated with S is inverted bit by bit, and that results is transmitted to the device designated with D.

Before execution S

b15															b0
1	0	0	1	0	1	1	0	0	1	1	0	0	1	1	1

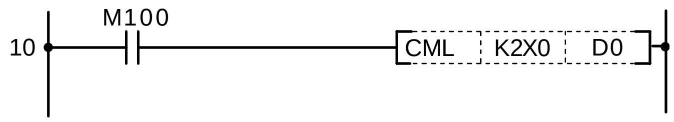
⇩ Inversion

After execution D

b15															b0
1	0	0	1	0	1	1	0	0	1	1	0	0	1	1	1

Program example

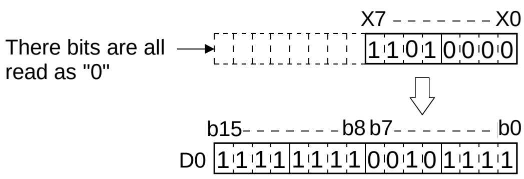
(1) Program to invert X0 to X7 data and to transmit the results to D0.



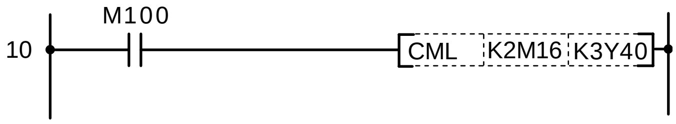
Coding

No. of steps	Com-mand	Device		
10	LD	M100		
11	CML	K2X0	D0	
14				

When the number of bits at S is less than the number of bits at D:



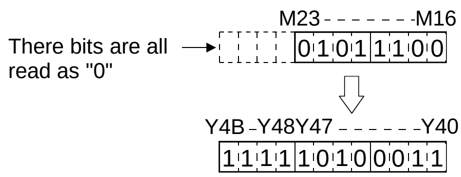
(2) Program to invert M16 to M23 data and to transmit the results to Y40 to Y47.



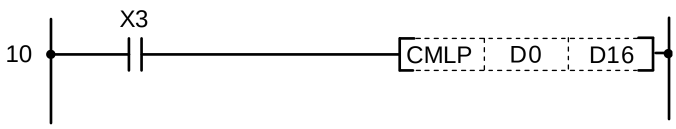
Coding

No. of steps	Com-mand	Device		
10	LD	M100		
11	CML	K2M16	K3Y40	
14				

When the number of bits at S is less than the number of bits at D:

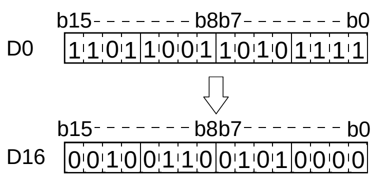


(3) Program to invert D0 data when X3 turns ON and to transmit the results to D16.



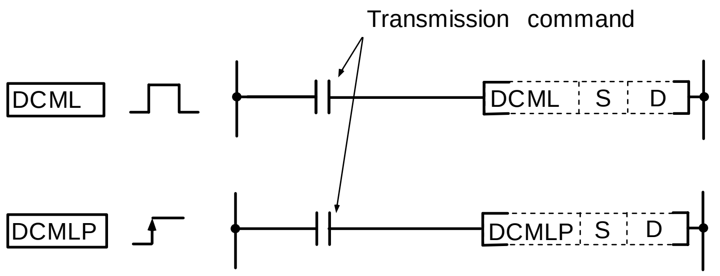
Coding

No. of steps	Com-mand	Device		
10	LD	X3		
11	CMLP	D0	D16	
20				



○ DCML, DCMLP ... 32-bit data negation transmission

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3/4
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						

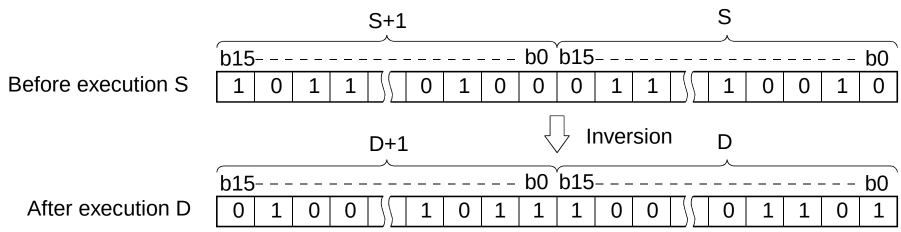


Setting data

S	Data to be inverted or head No. of device where data is stored.
D	Head No. of device to store inversion results

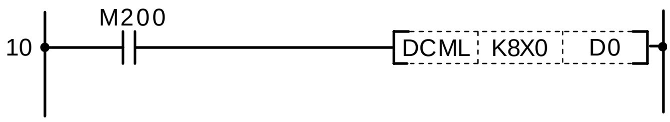
Function

The 32-bit data designated with S is inverted bit by bit, and that results is transmitted to the device designated with D.



Program example

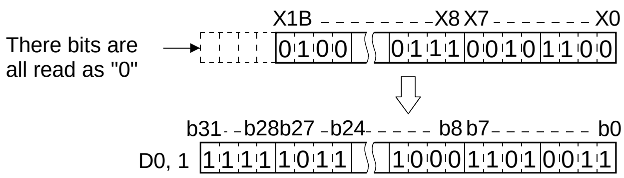
(1) Program to invert X0 to X1F data and to transmit the results to D0, 1.



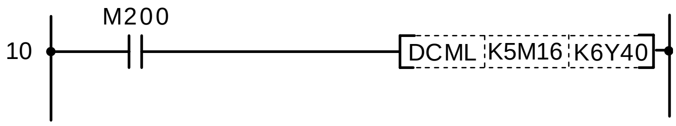
Coding

No. of steps	Com-mand	Device		
10	LD	M200		
11	DCML	K8X0	D0	
14				

When the number of bits at S is less than the number of bits at D:



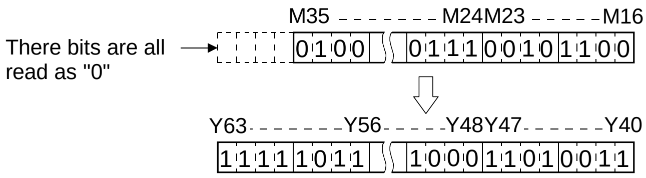
(2) Program to invert M16 to M35 data and to transmit the results to Y40 to Y63.



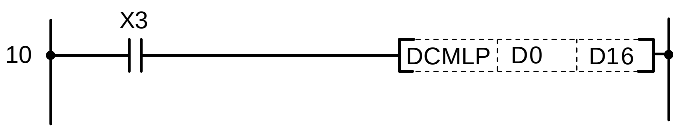
Coding

No. of steps	Com-mand	Device		
10	LD	M200		
11	DCML	K5M16	K6Y40	
14				

When the number of bits at S is less than the number of bits at D:

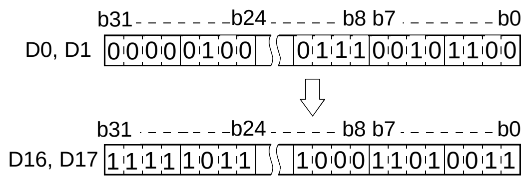


(3) Program to invert D0, 1 data when X3 turns ON and to transmit the results to D16, 17.



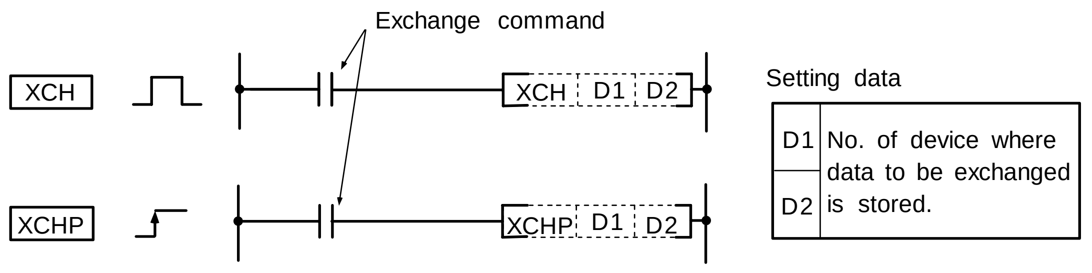
Coding

No. of steps	Com-mand	Device		
10	LD	X3		
11	DCMLP	D0	D16	
20				



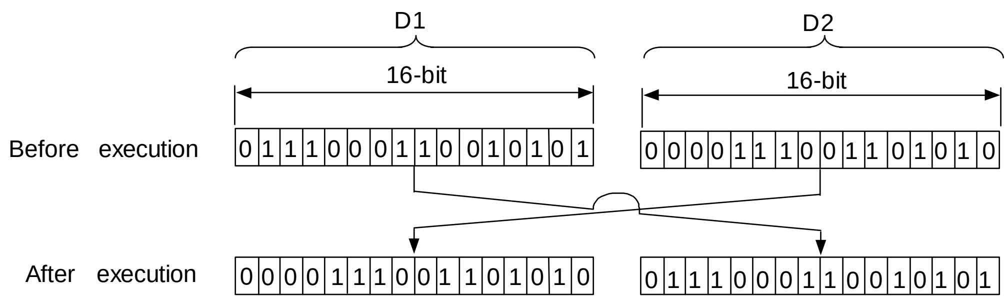
○ XCH, XCHP ... 16-bit data exchange

	Usable device: Bit device										Word device								Con-stant		Pointer	Digit desig-nation	Index	No. of steps
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D1		○	○	○	○	○	○		○		○	○	○	○	○	○		○				○	○	4
D2											○	○	○	○	○	○		○						



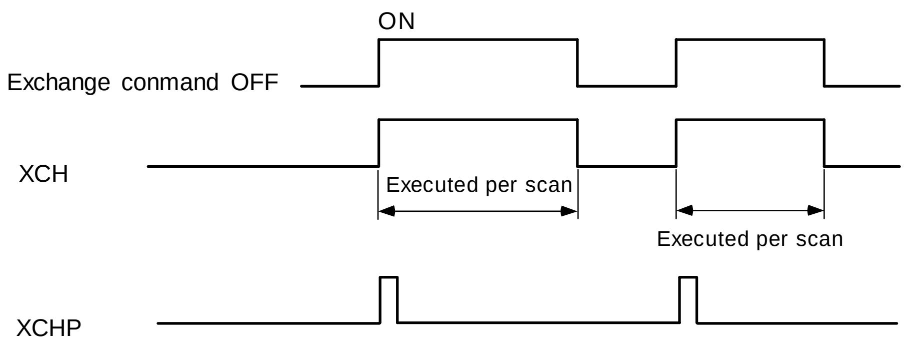
Function

The D1 and D2 16-bit data are exchanged.



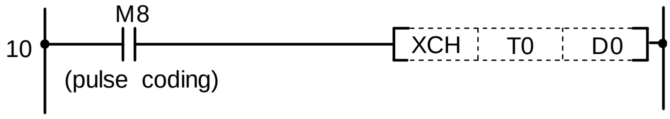
Execution conditions

The execution conditions for the XCH, XCHP are as shown below.



Program example

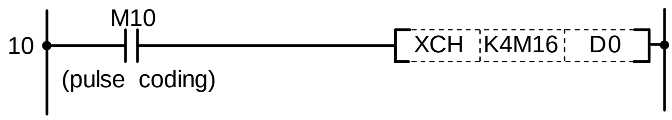
(1) Program to exchange T0 current value with D0 data when M8 turns ON.



Coding

No. of steps	Com-mand	Device		
10	LD	M8		
11	XCH	T0	D0	
15				

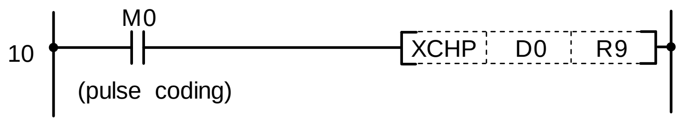
(2) Program to exchange D0 data with M16 to M31 data when M10 turns ON.



Coding

No. of steps	Com-mand	Device		
10	LD	M10		
11	XCH	K4M16	D0	
15				

(3) Program to exchange D0 data with R9 data when M0 turns ON.

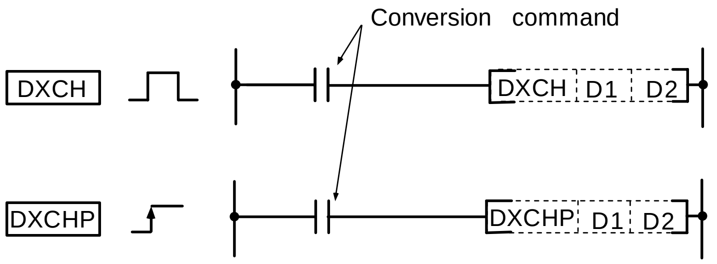


Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	XCHP	D0	R9	
21				

○ DXCH, DXCHP ... 32-bit data exchange

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D1		○	○	○	○	○	○		○		○	○	○	○	○	○		○				○	○	4
D2											○	○	○	○	○	○		○						

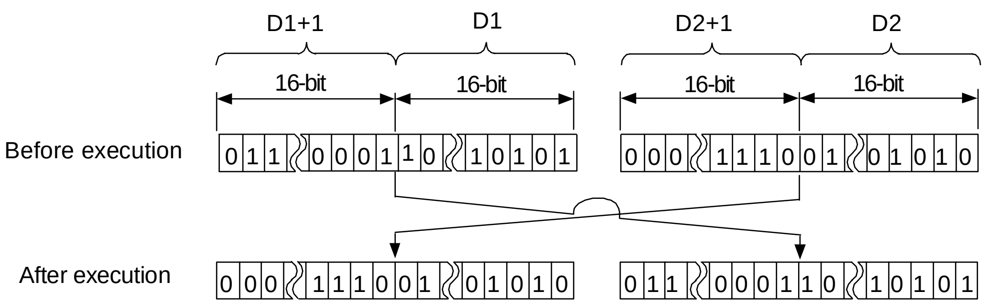


Setting data

D1	Head No. of device where data to be exchanged is stored.
D2	

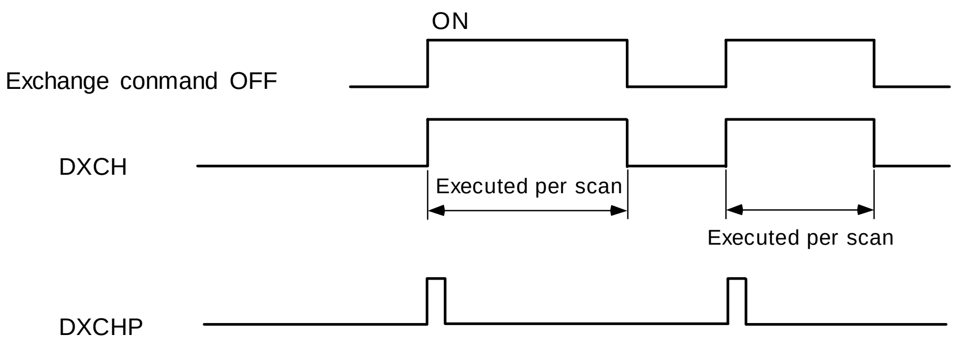
Function

The D1 and D2 32-bit data are exchanged.



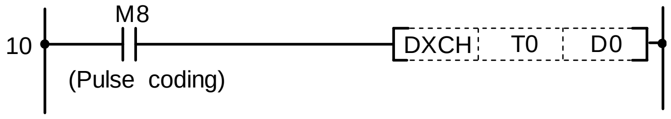
Execution conditions

The execution conditions for the DXCH, DXCHP are as shown below.



Program example

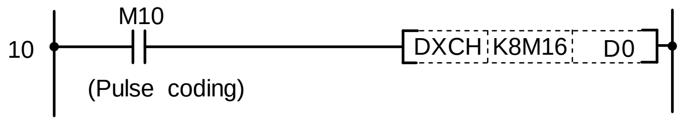
(1) Program to exchange T0, T1 current values with D0, 1 data when M8 turns ON.



Coding

No. of steps	Com-mand	Device		
10	LD	M8		
11	DXCH	T0	D0	
15				

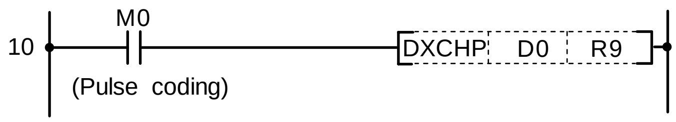
(2) Program to exchange D0, 1 data with M16 to M47 data when M10 turns ON.



Coding

No. of steps	Com-mand	Device		
10	LD	X10		
11	DXCH	K8M16	D0	
15				

(3) Program to exchange D0, 1 data with R9, 10 data when M0 turns ON.

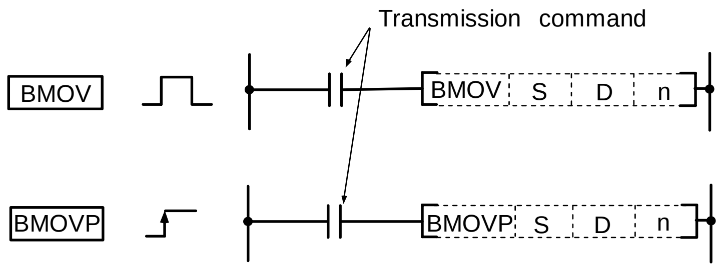


Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	DXCHP	D0	R9	
21				

○ BMOV, BMOVP ... Block transmission of 16-bit data

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S											○	○	○	○	○	○		○						5
D											○	○	○	○	○	○		○						
n											○	○	○	○	○	○		○	○	○			○	

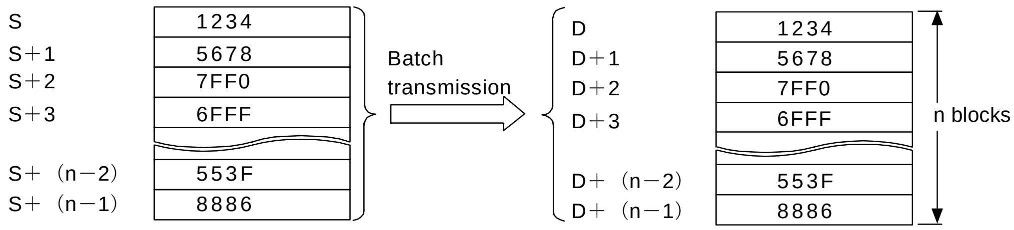


Setting data

S	Head No. of device where data to be transmitted is stored.
D	Head No. of device to store transmitted data
n	Number of blocks to be transmitted

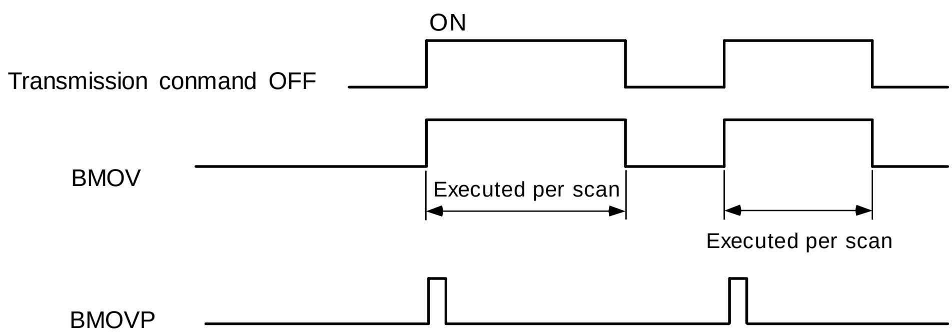
Function

The data of n blocks from the device designated with S are batch transmitted to the area of n blocks from the device designated with D.



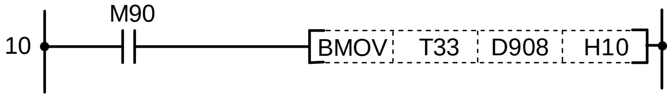
Execution conditions

The execution conditions of the BMOV, BMOVP are as shown below.



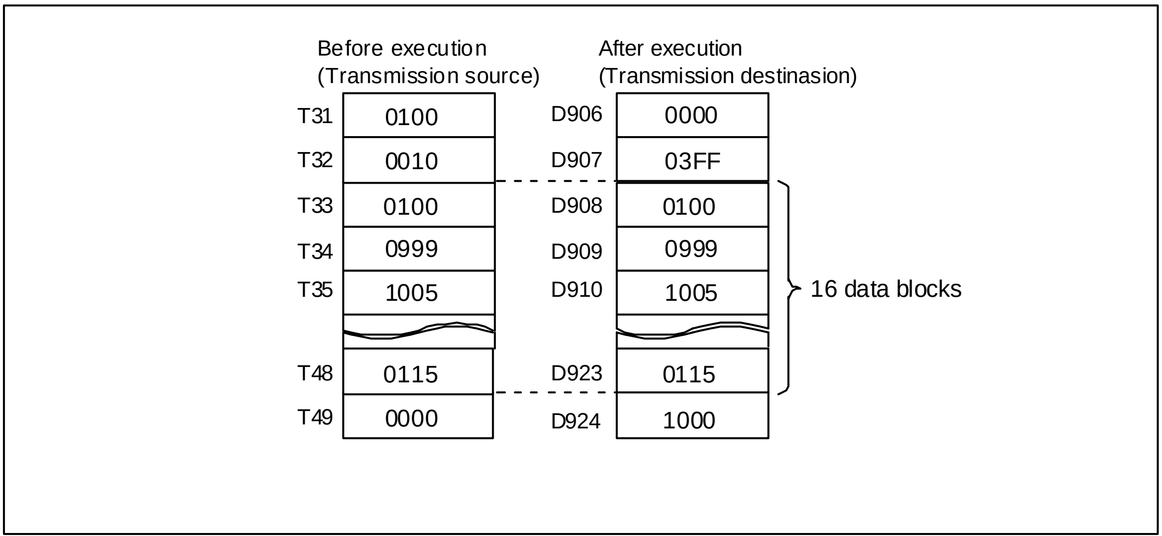
Program example

(1) Program to transmit the current values of T33 to T48 to D908 to D923.



Coding

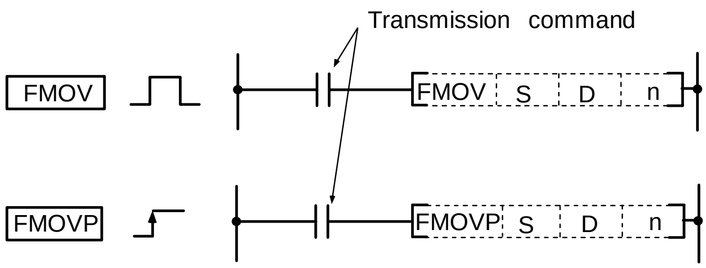
No. of steps	Com-mand	Device		
10	LD	M90		
11	BMOV	T33	D908	H10
16				



Block transmission with BMOV command

○ FMOV, FMOVP ... Batch transmission of same 16-bit data

	Usable device																						Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S											○	○	○	○	○	○		○	○	○			○	5
D											○	○	○	○	○	○		○						
n											○	○	○	○	○	○		○	○	○			○	

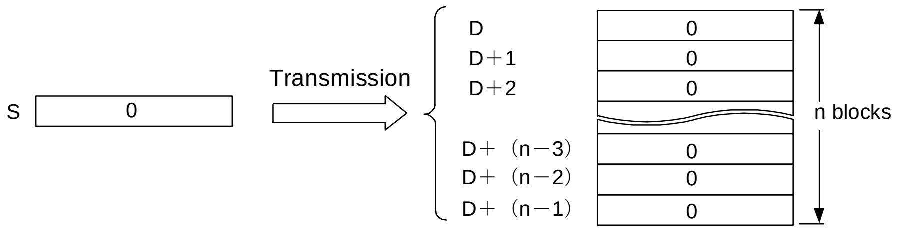


Setting data

S	Head No. of device where data to be transmitted is stored.
D	Head No. of device to store transmitted data
n	Number of blocks to be transmitted to

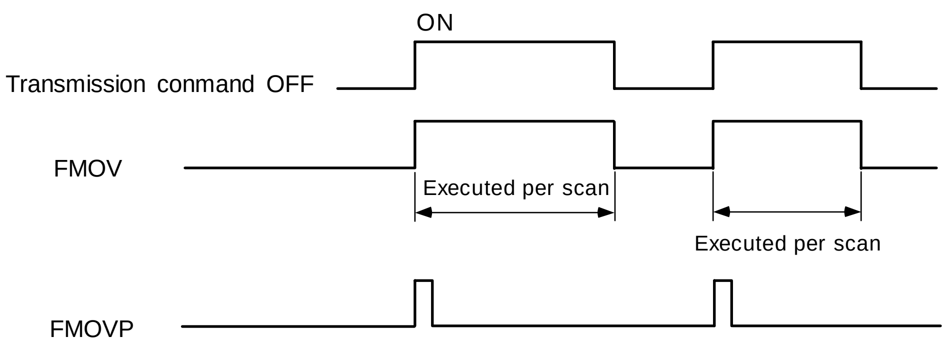
Function

The data of the device designated with S are transmitted to the area of n blocks from the device designated with D.



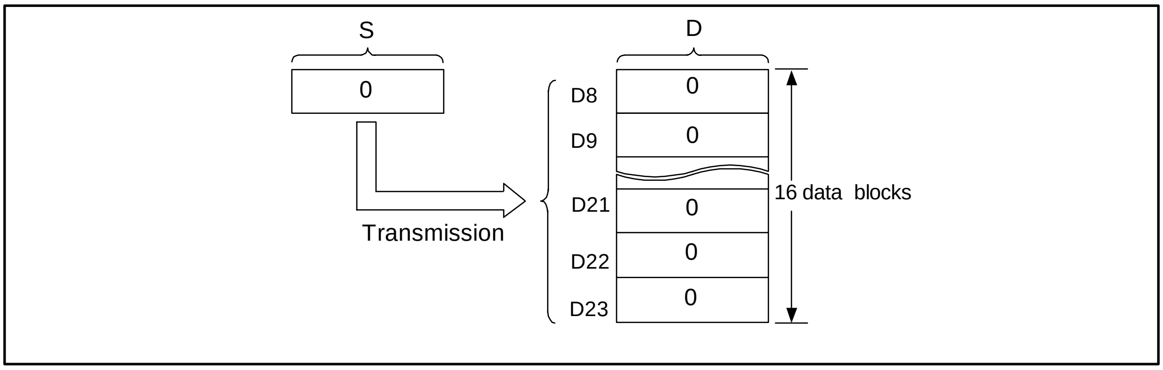
Execution conditions

The execution conditions of the FMOV, FMOVP are as shown below.

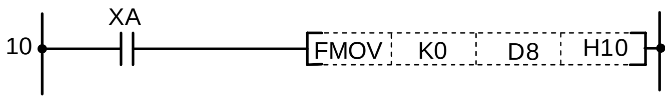


Program example

(1) Program to reset (clear) D8 to D23 when XA turns ON.



Resetting of data registers with FMOV command

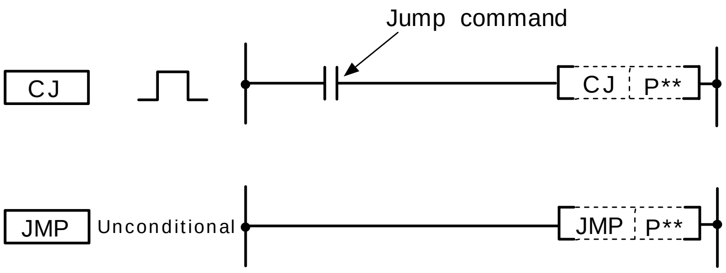


Coding

No. of steps	Com-mand	Device		
10	LD	XA		
11	FMOV	K0	D8	H10
16				

○ **CJ, JMP ... Conditional jump**

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
P																					○			2



Setting data

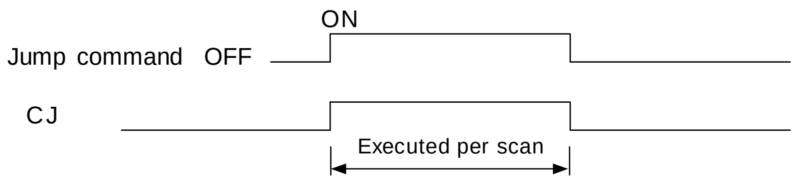
P**	Jump destination pointer No. (P0 to P249)

Function

CJ

(1) The block of the designated pointer No. within the same program file is executed when the jump command turns ON.

(2) The program of the next step is executed when the jump command is OFF.

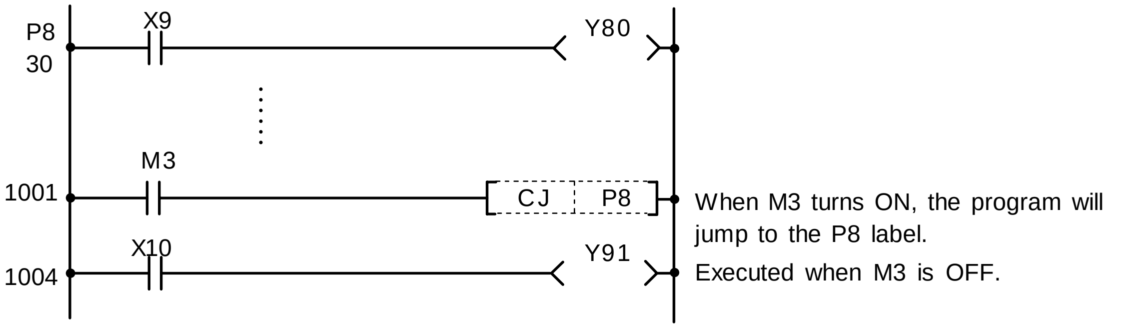


JMP

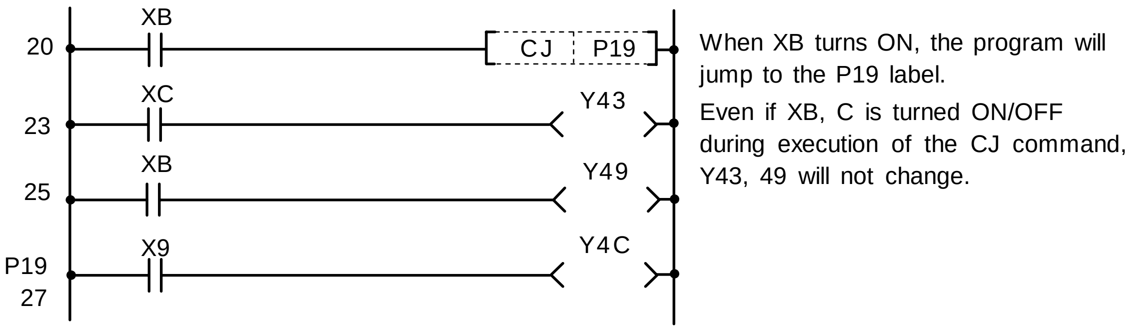
(1) The block of the designated pointer No. within the same program file is executed unconditionally.

Points

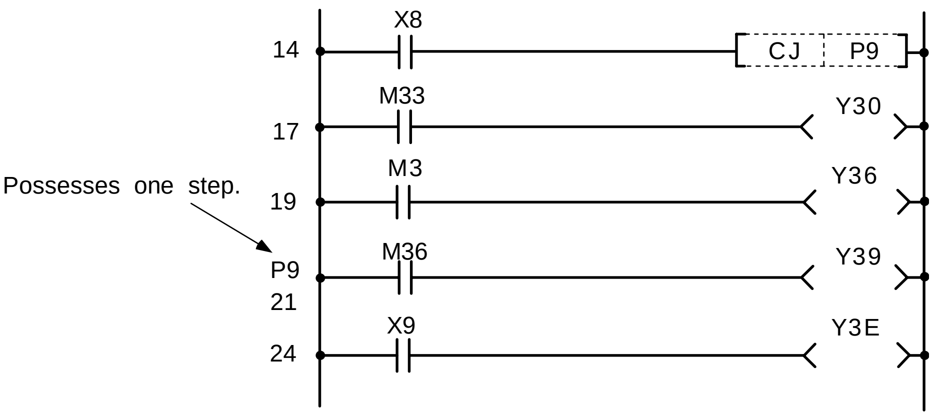
(1) After the timer coil is turned ON, if the timer that is turning the coil ON with the CJ or JMP command is jumped, the timer count will continue.
(2) The scan time will be shortened if jumping is done backward using the CJ or JMP command.
(3) The CJ and JMP commands can be used to jump to a smaller step.



(4) The devices skipped because CJ or JMP command will not change.



(5) Label (P**) possesses one step.

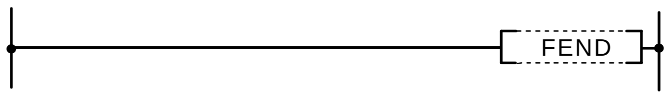


Caution

(1) Designate the pointer No. so that it comes prior to the END command.
(2) Designate the label No. which exists in the program file as the pointer No.
(3) Only the pointer No. which exists in the same program file as the jump command is available.

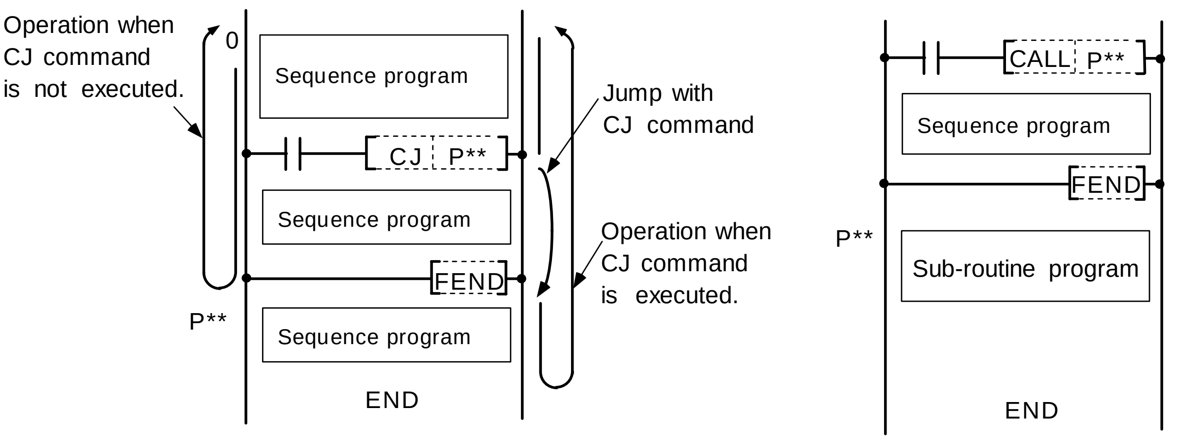
○ FEND ... Program end

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
																								1



Function

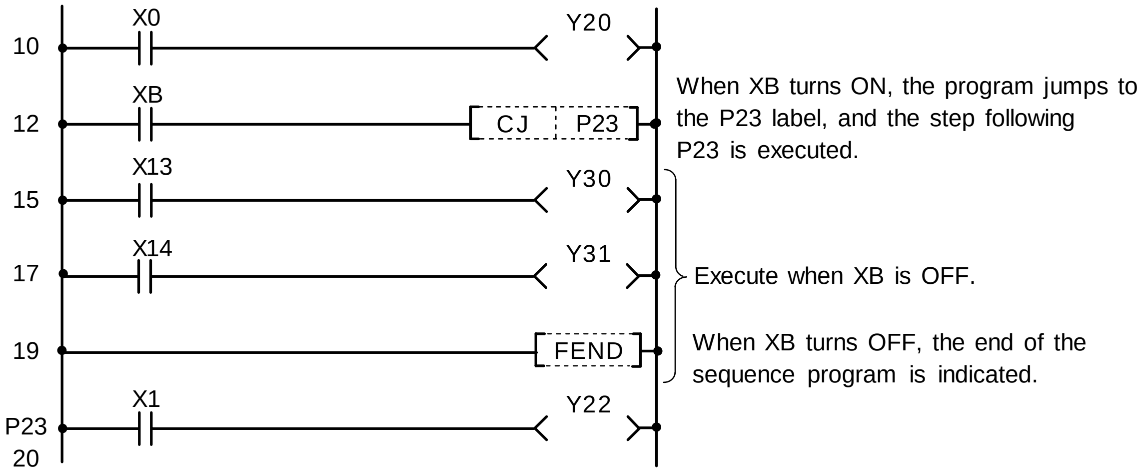
The sequence program is ended.



(a) When CJ command is used
(b) When a subroutine program is used

Program example

Program when using CJ command.

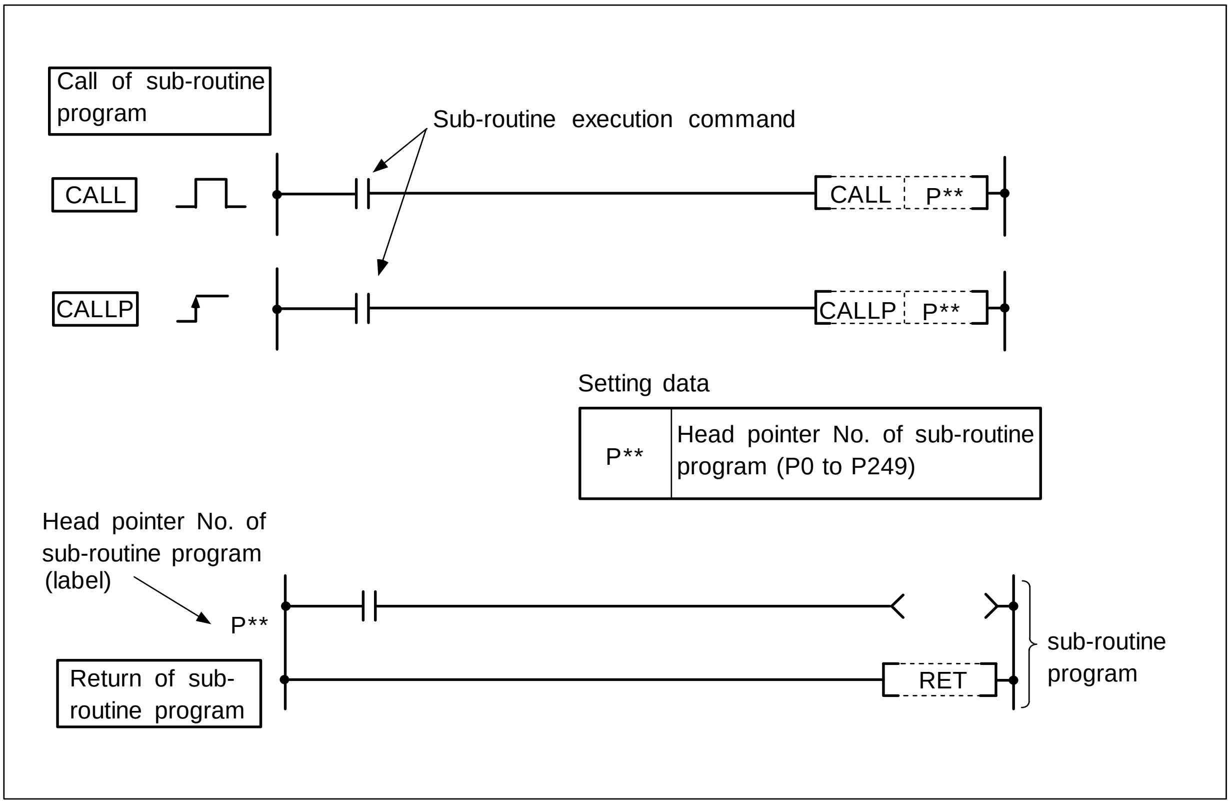


Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	OUT	Y20		
12	LD	XB		
13	CJ	P23		
15	LD	X13		
16	OUT	Y30		
17	LD	X14		
18	OUT	Y31		
19	FEND			
20		P23		
21	LD	X1		
22	OUT	Y22		
23				

○ **CALL, CALLP, RET ... Call/return of sub-routine program**

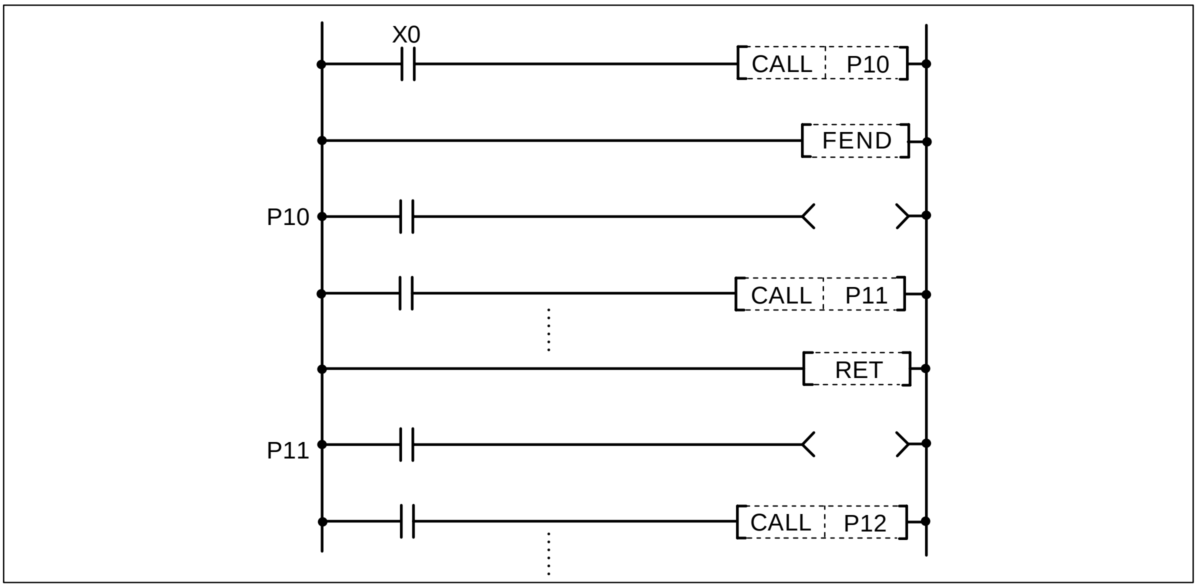
	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
P																					○			2/1



Function

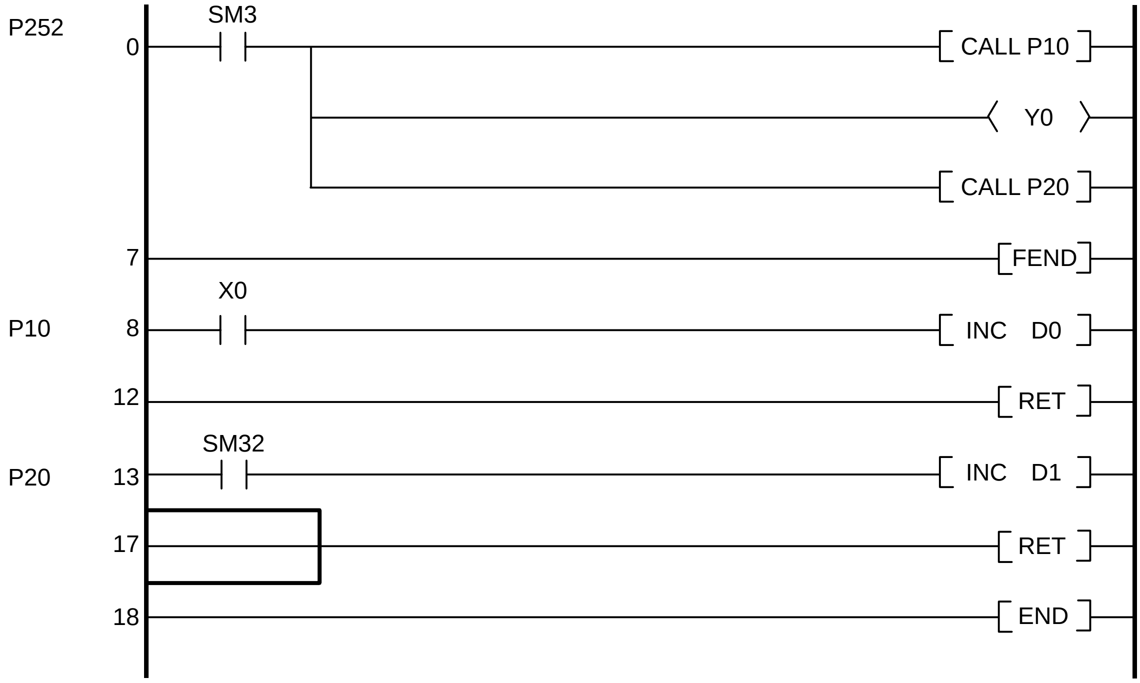
CALL CALLP

(1) The sub-routine program designated with the point (P**) is executed.



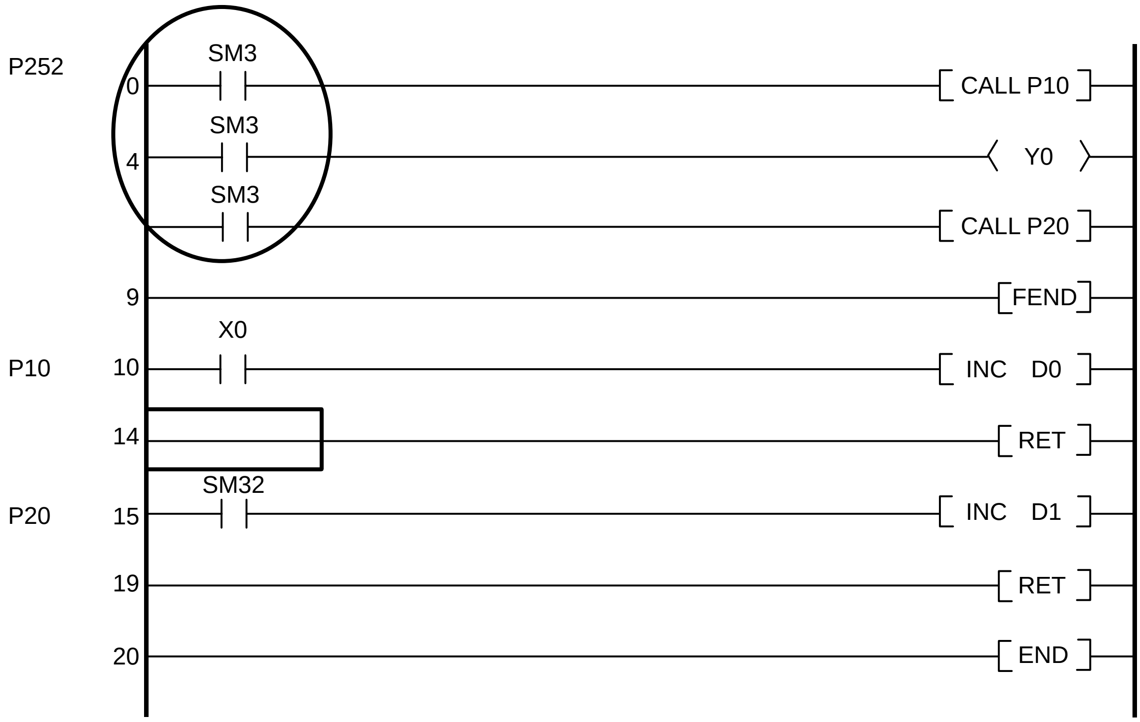
Cautions

When returning from CALL or CALLP command (that is, when executing RET command), the operation results obtained immediately before will be retained.
Thus, make sure not to program a command following to CALL or CALLP as it may cause an illegal operation. The example is shown below.



(1) The program of P10 label is always called.
(2) The operation results when returning from P10 is determined by the results of "LD X0".
(3) "OUT Y0" will be executed after returning from P10. In this time, the X0 state will affect the P10 results.
(4) In the same manner as (3), the X0 state will affect the P20 results.

Correct the above program as shown below.

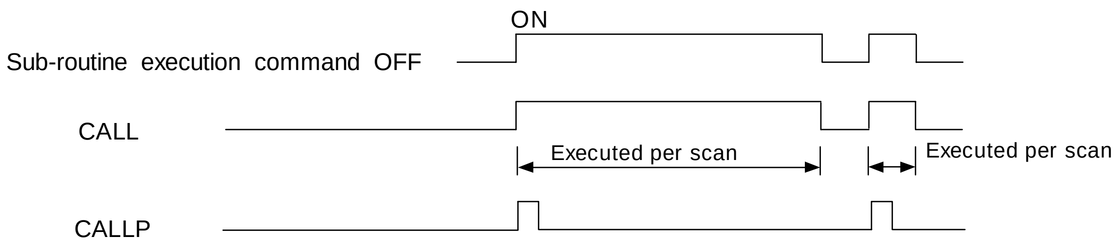


RET

(1) The end of the sub-routine program is indicated.
(2) When the RET command is executed, the sequence program in the step after the CALL or CALLP command will be executed.

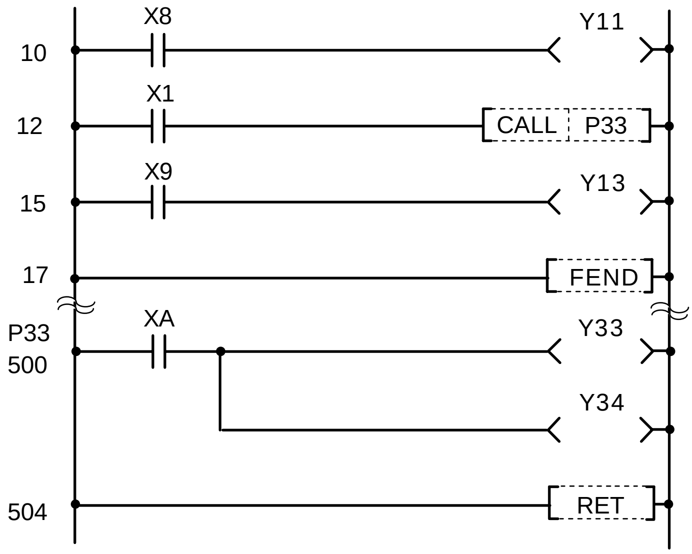
Execution conditions

The execution conditions of the CALL, CALLP are as shown below.



Program example

Program to execute sub-routine program when X1 changes from OFF to ON.



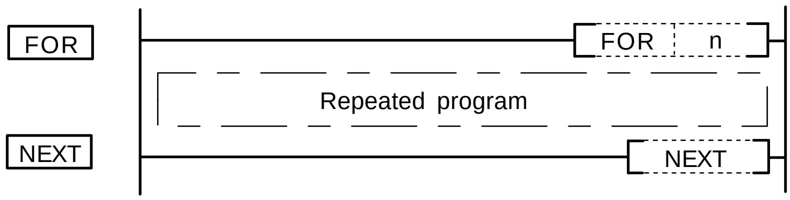
Coding

No. of steps	Com-mand	Device		
10	LD	X8		
11	OUT	Y11		
12	LD	X1		
13	CALL	P33		
15	LD	X9		
16	OUT	Y13		
17	FEND			
18				
:				
500		P33		
501	LD	XA		
502	OUT	Y33		
503	OUT	Y34		
504	RET			
505				

○ FOR, NEXT ... Command loop

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
n											○	○	○	○	○	○		○	○	○				5/4

(Note) "n" is used with FOR command.



Setting data

n	Number of repetitions of the FOR to NEXT loop (from 1 to 32767).

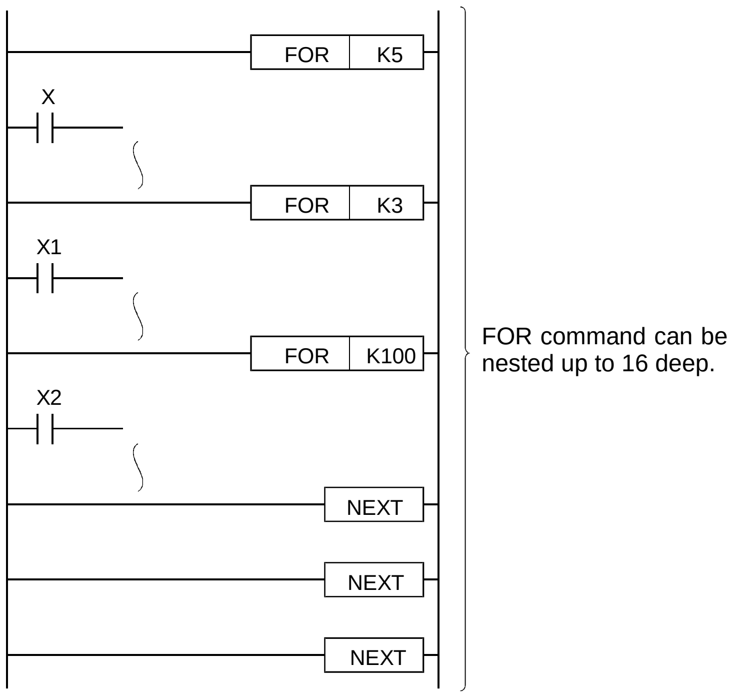
Function

(1) After the processing in the FOR to NEXT loop is executed n-times without conditions, the step following the NEXT command will be executed.

(2) The value of "n" can be designated at between 1 and 32767.
If it is designated at a value of from, -32768 to 0, it will be executed as though n=1.

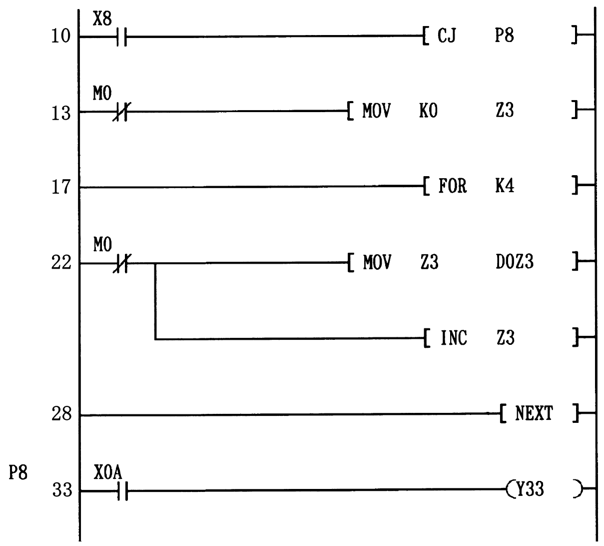
(3) If you do not desire to execute the processing called for within the FOR to NEXT loop, use the CJ command to jump past it.

(4) FOR command can be nested up to 16 deep.



Program example

(1) Program to execute FOR to NEXT loop when X8 is OFF, and not to execute it when X8 is ON.



Coding

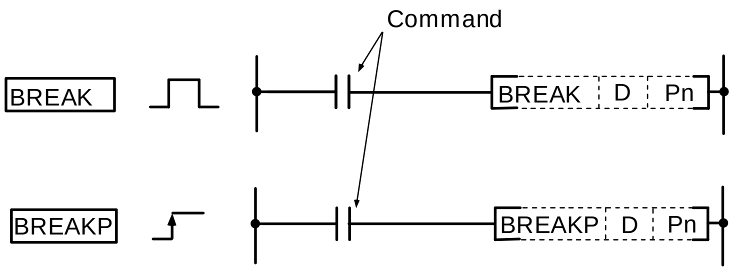
No. of steps	Com-mand	Device		
10	LD	X8		
11	CJ	P8		
13	LDI	M0		
14	MOV	K0	Z3	
17	FOR	K4		
22	LDI	M0		
23	MOV	Z3	D0Z3	
26	INC	Z3		
28	NEXT			
32		P8		
33	LD	X0A		
34	OUT	Y33		
35				

Caution

(1) To force an end to the repetitious execution of the FOR to NEXT loop during the execution of the loop, insert a BREAK command.
(2) Use the EGP, EGF commands to conduct index qualification for the FOR to NEXT contact.
(3) Be sure not to execute the END (FEND) command after the execution of the FOR command but before of the NEXT command.
(4) Always execute the FOR command prior to the NEXT command.
(5) Be sure not to insert the STOP command within the FOR to NEXT loop.

○ BREAK, BREAKP ... Forced end of FOR to NEXT loop

	Usable device																					Digit desig-nation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D											○	○	○	○	○	○		○						5
P																					○			

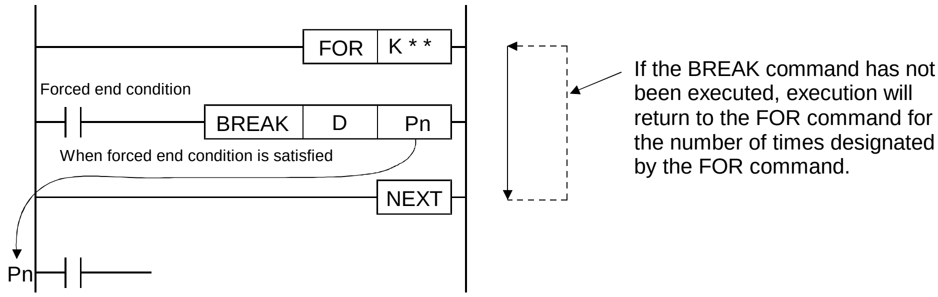


Setting data

D	Number of device to store the number of repetitions remaining
Pn	Number of branch destination pointer when the end of processing repetitions.

Function

(1) The repetition processing by the FOR to NEXT loop is forced to end, and shifts to the execution of the pointer designated by Pn.



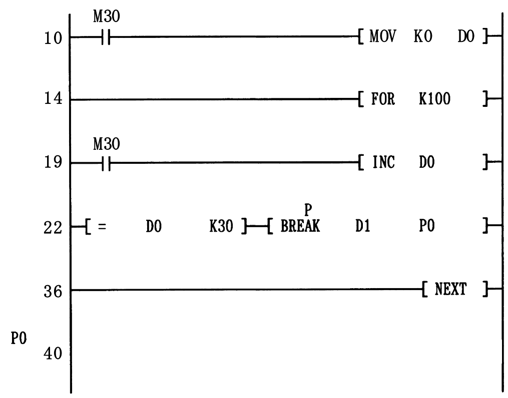
(2) The number of repetitions remaining at the time that the FOR to NEXT loop was forced to end is stored at D.
However, the number of times when the BREAK command was executed is also included in the number of repetitions remaining.

(3) The BREAK can be used only during the execution of a FOR to NEXT loop.

(4) A BREAK can be used only when there is only one level of nesting
When there are multiple nesting levels, execute the same number of BREAK commands as there are nesting levels.

Program example

(1) Program to force FOR to NEXT loop to end when D0 data reaches 30, that is, FOR to NEXT loop has been executed 30 times.



Coding

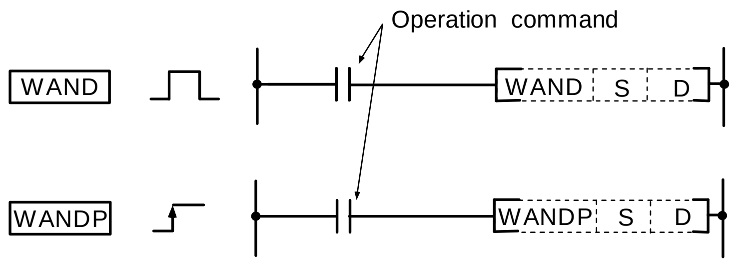
No. of steps	Com-mand	Device		
10	LD	M30		
11	MOV	K0	D0	
14	FOR	K100		
19	LD	M30		
20	INC	D0		
22	LD=	D0	K30	
25	BREAKP	D1	P0	
36	NEXT			
40		P0		

Caution

(1) The value 71 is stored in D1 when the BREAK command is executed in the program above.
(2) Be sure not to use the BREAK or BREAKP command in a case other than with the FOR to NEXT loop.
(3) Be sure that the pointer No. designated with Pn exists in the program file.

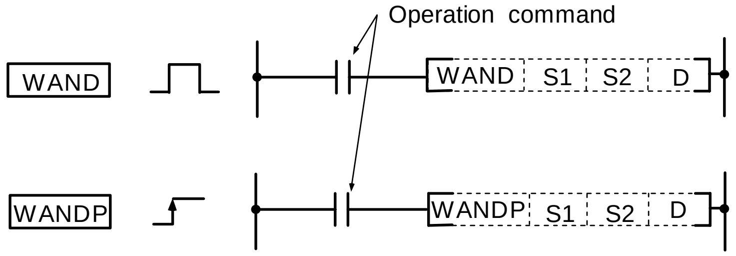
○ WAND, WANDP ... Logical AND of 16-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	4
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Setting data

S	Data to be logical ANDed or No. of device where data is stored.
D	The logical AND results are stored in D device.



Setting data

S1, S2	Data to be logical ANDed or No. of device where data is stored.
D	No. of device to store logical AND results.

Function

(1) WAND S D, WANDP S D

Logical AND is executed for each bit of the 16-bit data in the device designated with D and the device designated with S, and the results are stored in the device designated with D.

16-bit

Before execution	D	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
	WAND,WANDP																
	S	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
After execution	D	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0

WAND S1 S2 D, WANDP S1 S2 D

Logical AND is executed for each bit of the 16-bit data in the device designated with S1 and the device designated with S2, and the results are stored in the device designated with D.

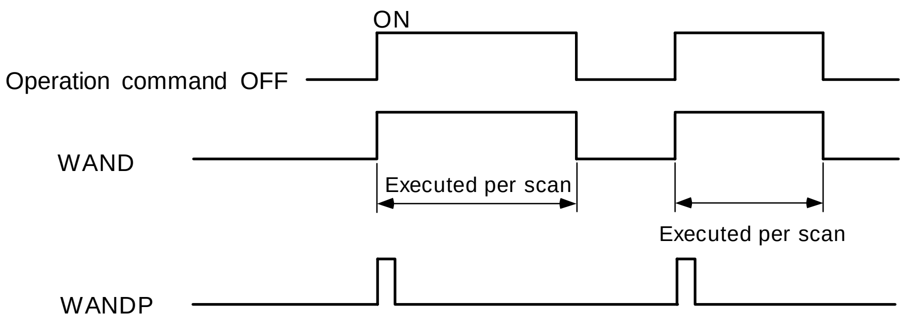
16-bit

Before execution	S1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
	WAND,WANDP																
	S2	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
After execution	D	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0

(2) The bits other than designated digit are operated as 0.
(Refer to program example (2).)

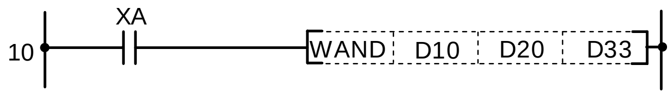
Execution conditions

The execution conditions for WAND, WANDP are as follows.



Program example

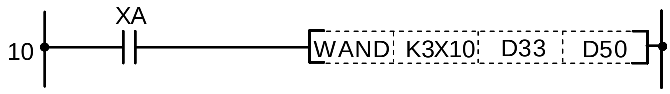
(1) Program to execute logical AND of the D10 data and D20 data when XA turns ON, and to store the results in D33.



Coding

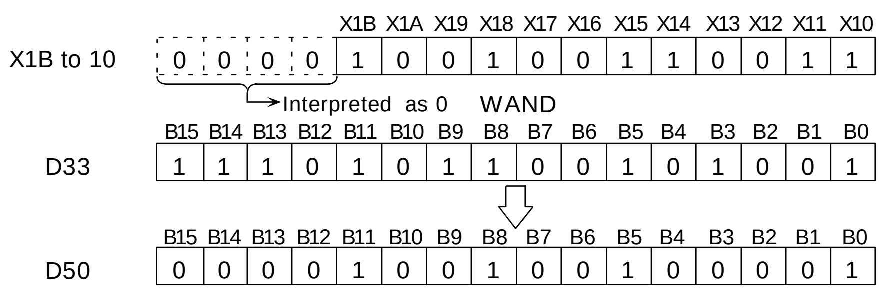
No. of steps	Com-mand	Device		
10	LD	XA		
11	WAND	D10	D20	D33
15				

(2) Program to execute logical AND of the X10 to 1B data and D33 data when XA turns ON, and to output the results to D50.



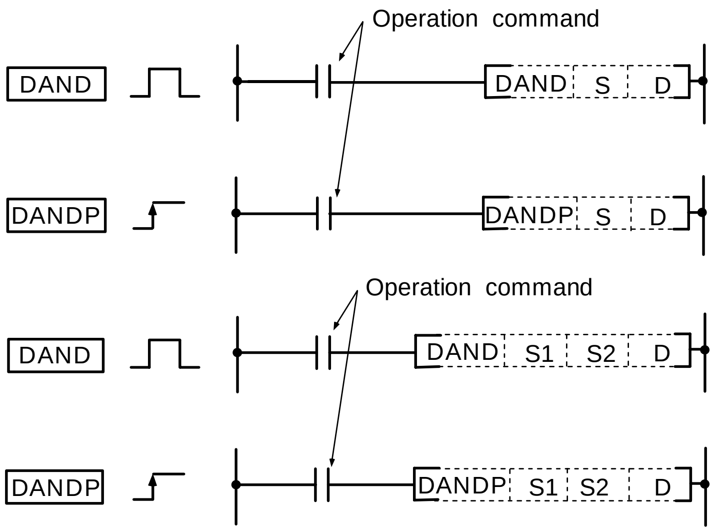
Coding

No. of steps	Com-mand	Device		
10	LD	XA		
11	WAND	K3X10	D33	D50
15				



○ DAND, DANDP ... Logical AND of 32-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3/4
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	4/5
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Setting data

S	Data to be logical ANDed or head No. of device where data is stored.
D	The logical AND results are stored in D device.

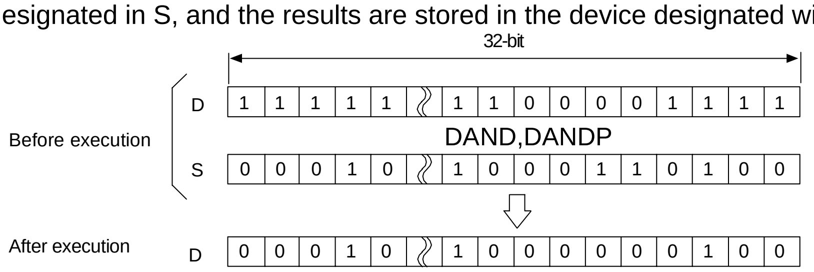
Setting data

S1 / S2	Data to be logical ANDed or head No. of device where data is stored.
D	Head No. of device to store logical AND results.

Function

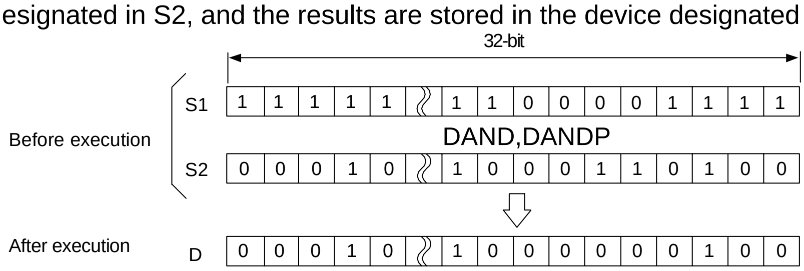
(1) [DAND S D], [DANDP S D]

Logical AND is executed for each bit of the 32-bit data in the device designated with D and the device designated in S, and the results are stored in the device designated with D.



[DAND S1 S2 D], [DANDP S1 S2 D]

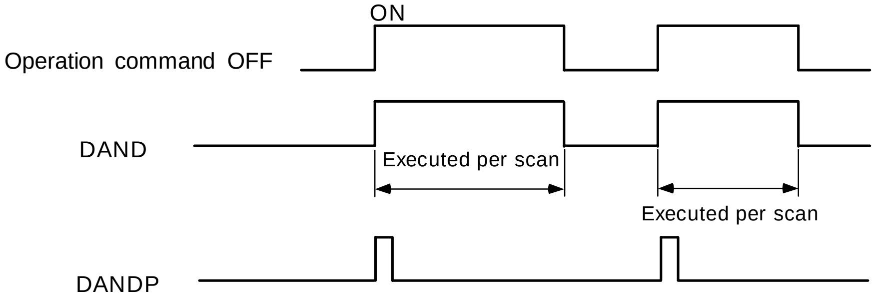
Logical AND is executed for each bit of the 32-bit data in the device designated with S1 and the device designated in S2, and the results are stored in the device designated with D.



(2) The bits other than designated digit are operated as 0.
(Refer to program example (1).)

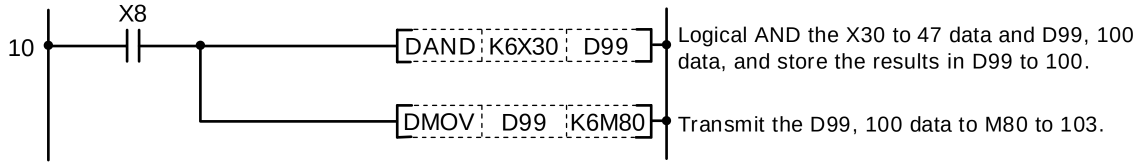
Execution conditions

The execution conditions for the DAND, DANDP are as follows.



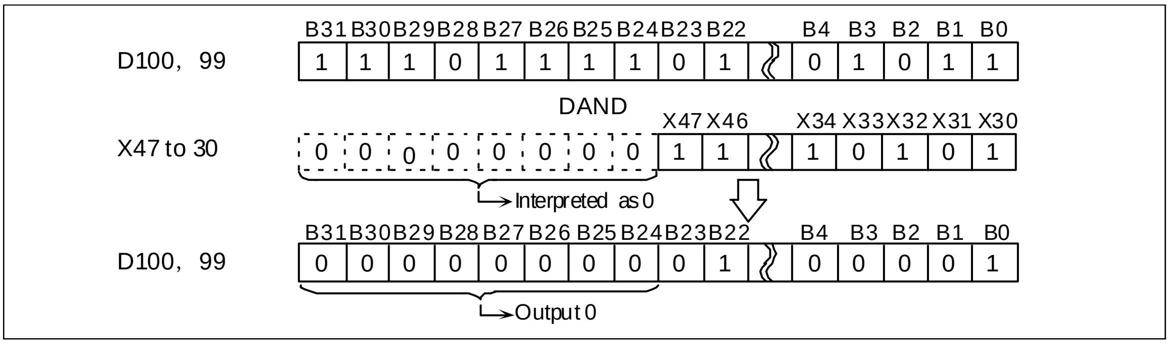
Program example

(1) Program to execute logical AND of the X30 to 47 24-bit data and D99, 100 data when X8 turns ON, and to transmit the results to M80 to 103.

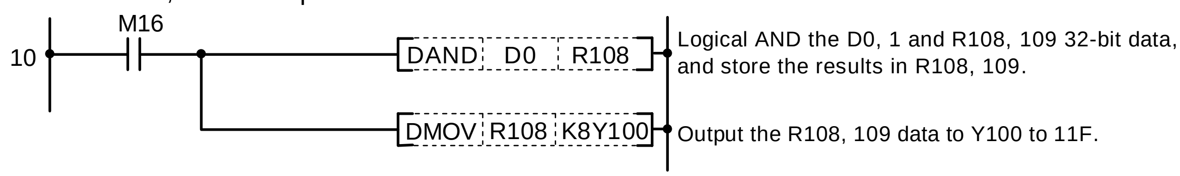


Coding

No. of steps	Command	Device		
10	LD	X8		
11	DAND	K6X30	D99	
14	DMOV	D99	K6M80	
17				



(2) Program to execute logical AND of the D0, 1 32-bit data and R108, 109 32-bit data when M16 turns ON, and to output the results to Y100 to 11F.

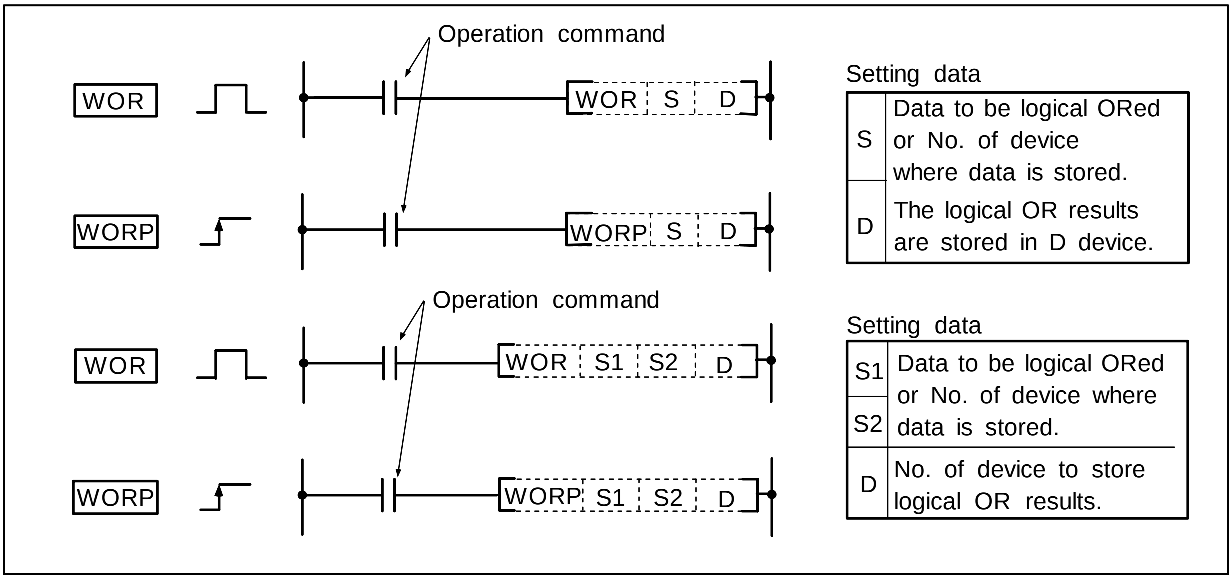


Coding

No. of steps	Command	Device		
10	LD	M16		
11	DAND	D0	R108	
14	DMOV	R108	K8Y100	
17				

○ WOR, WORP ... Logical OR of 16-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	4
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Function

WOR S D, WORP S D

Logical OR is executed for each bit of the 16-bit data in the device designated with D and the device designated with S, and the results are stored in the device designated with D.

16-bit

Before execution	D	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	WOR,WORP																
	S2	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0	1
After execution	D	0	1	0	1	1	1	1	1	1	1	1	1	0	1	0	1

WOR S1 S2 D, WORP S1 S2 D

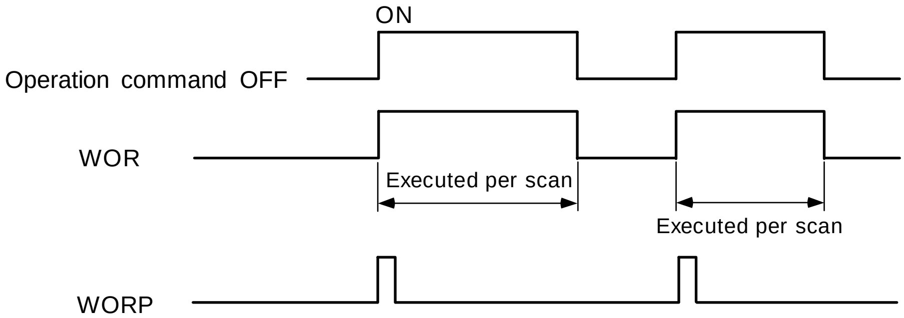
Logical OR is executed for each bit of the 16-bit data in the device designated with S1 and the device designated with S2, and the results are stored in the device designated with D.

16-bit

Before execution	S1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	WOR,WORP																
	S2	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0	1
After execution	D	0	1	0	1	1	1	1	1	1	1	1	1	0	1	0	1

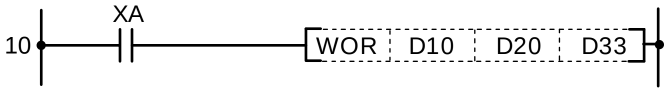
Execution conditions

The execution conditions for WOR, WORP are as follows.



Program example

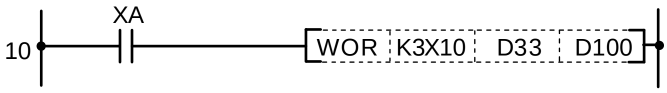
(1) Program to execute logical OR of the D10 data and D20 data when XA turns ON, and to store the result in D33.



Coding

No. of steps	Com-mand	Device		
10	LD	XA		
11	WOR	D10	D20	D33
15				

(2) Program to execute logical OR of the X10 to 1B data and D33 data when XA turns ON, and to output the results in D100.

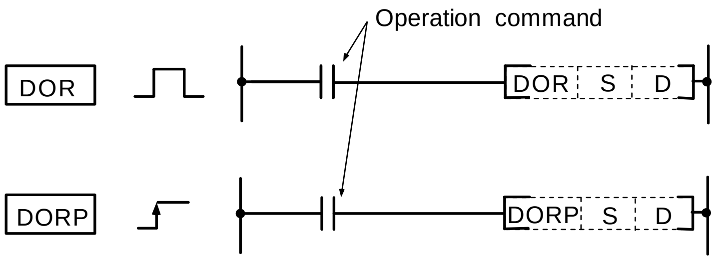


Coding

No. of steps	Com-mand	Device		
10	LD	XA		
11	WOR	K3X10	D33	D100
15				

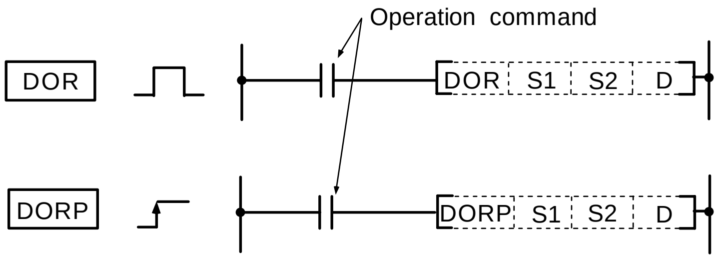
○ DOR, DORP ... Logical OR of 32-bit data

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3/4
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	4/5
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Setting data

S	Data to be logical ORed or head No. of device where data is stored.
D	The logical OR results are stored in D device.



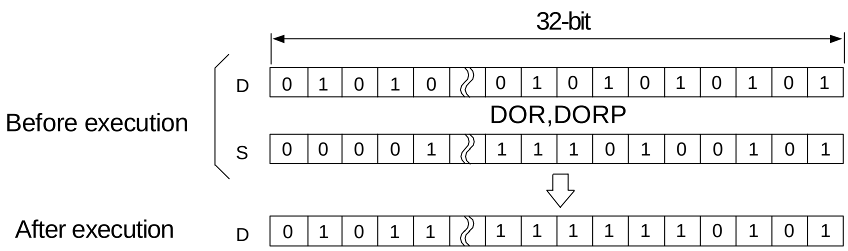
Setting data

S1 / S2	Data to be logical ORed or head No. of device where data is stored.
D	Head No. of device to store logical OR results.

Function

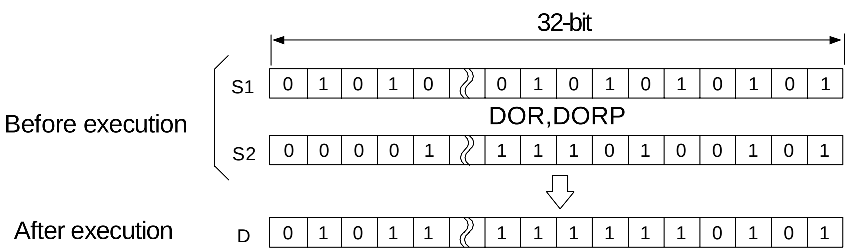
[DOR S D], [DORP S D]

Logical OR is executed for each bit of the 32-bit data in the device designated with D and the device designated with S, and the results are stored in the device designated with D.



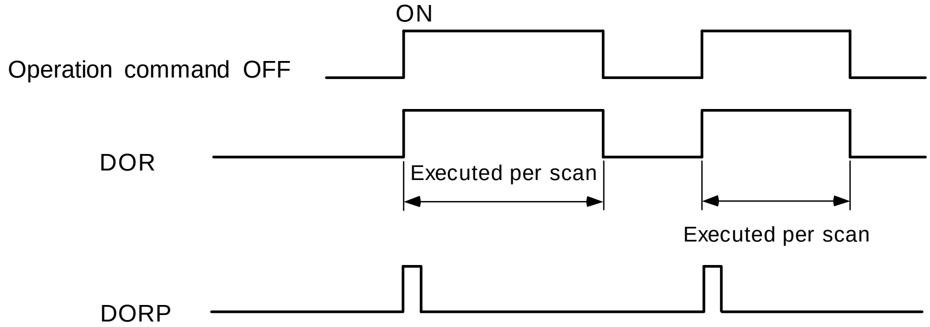
[DOR S1 S2 D], [DORP S1 S2 D]

Logical OR is executed for each bit of the 32-bit data in the device designated with S1 and the device designated with S2, and the results are stored in the device designated with D.



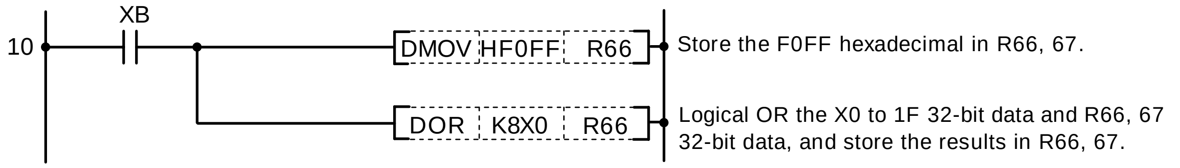
Execution conditions

The execution conditions for DOR, DORP are as follows.



Program example

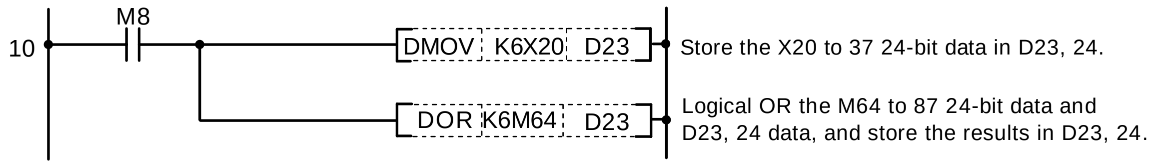
(1) Program to execute logical OR of the X0 to 1F 32-bit data and the F0FF hexadecimal when XB turns ON, and to store the results in R66, 67.



Coding

No. of steps	Com-mand	Device		
10	LD	XB		
11	DMOV	HFOFF	R66	
14	DOR	X8X0	R66	
17				

(2) Program that executes logical OR of the M64 to 87 24-bit data and X20 to 37 24-bit data when M8 turns ON, and stores the results in D23, 24.

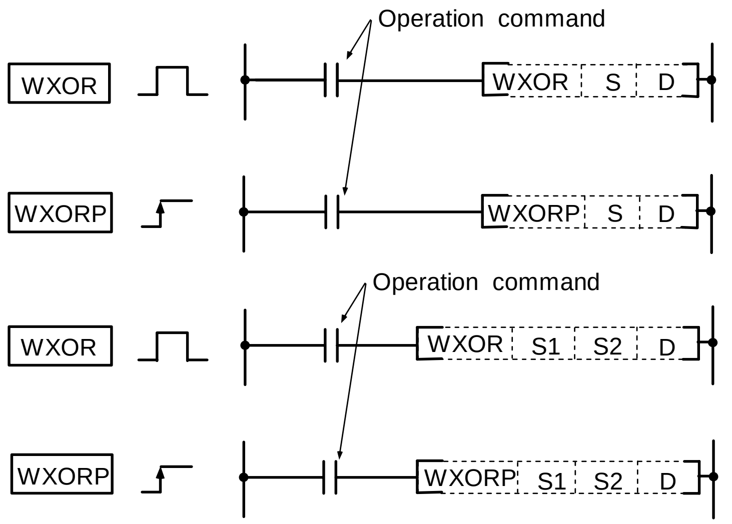


Coding

No. of steps	Com-mand	Device			
10	LD	M8			
11	DMOV	K6X20	D23		
14	DOR	K6M64	D23		
17					

○ WXOR, WXORP ... Exclusive OR of 16-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	4
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Setting data

S	Data to be exclusive ORed or No. of device where data is stored.
D	The exclusive OR results are stored in D device.

Setting data

S1	Data to be exclusive ORed or No. of device where data is stored.
S2	Data to be exclusive ORed or No. of device where data is stored.
D	No. of device to store exclusive OR results.

Function

[WXOR S D], [WXORP S D]

Exclusive OR is executed for each bit of the 16-bit data designated with D and designated with S, and the results are stored in the device designated with D.

16-bit

Before execution	D	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		WXOR,WXORP															
	S	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
After execution	D	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0

[WXOR S1 S2 D], [WXORP S1 S2 D]

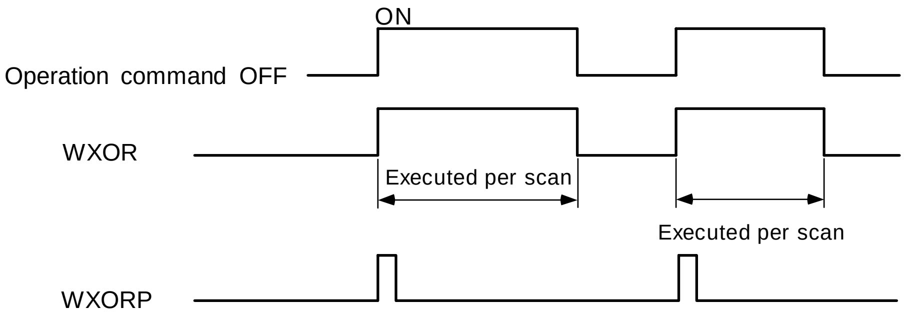
Exclusive OR is executed for each bit of the 16-bit data designated with S1 and designated with S2, and the results are stored in the device designated with D.

16-bit

Before execution	S1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		WXOR,WXORP															
	S2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
After execution	D	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0

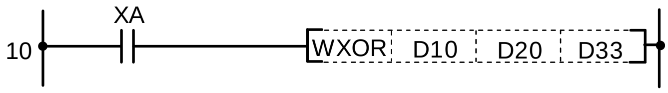
Execution conditions

The execution conditions for WXOR, WXORP are as follows.



Program example

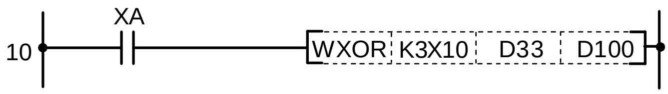
(1) Program to execute exclusive OR of the D10 data and D20 data when XA turns ON, and to store the results in D33.



Coding

No. of steps	Com-mand	Device		
10	LD	XA		
11	WXOR	D10	D20	D33
15				

(2) Program to execute exclusive OR of the X10 to 1B data and D33 data when XA turns ON, and to output the results to D100.

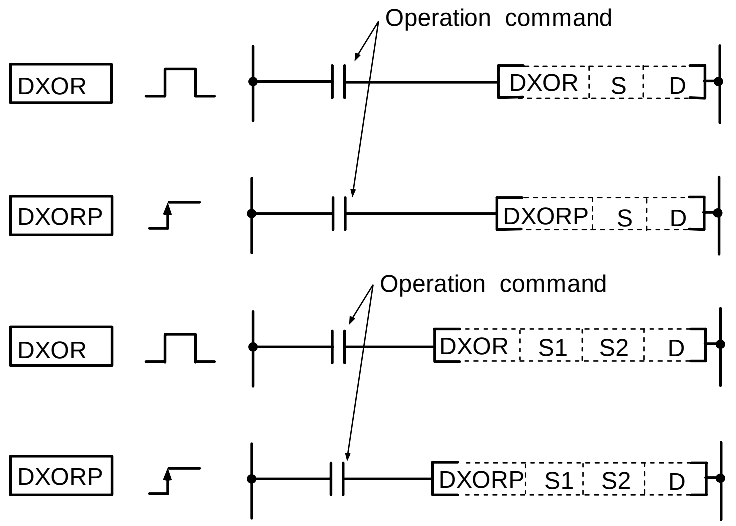


Coding

No. of steps	Com-mand	Device		
10	LD	XA		
11	WXOR	K3X10	D33	D100
15				

○ DXOR, DXORP ... Exclusive OR of 32-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3/4
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	4/5
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Setting data

S	Data to be exclusive ORed or head No. of device where data is stored.
D	The exclusive OR results are stored in D device.

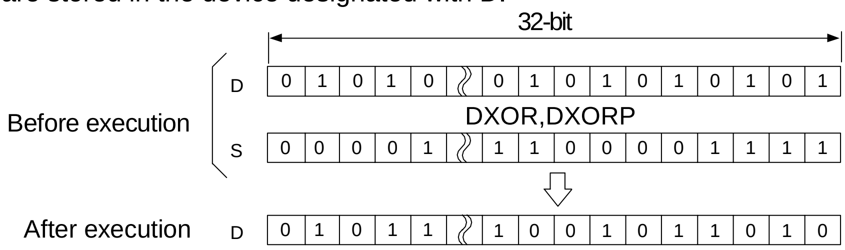
Setting data

S1	Data to be exclusive ORed or head No. of device where data is stored.
S2	
D	Head No. of device to store exclusive OR results.

Function

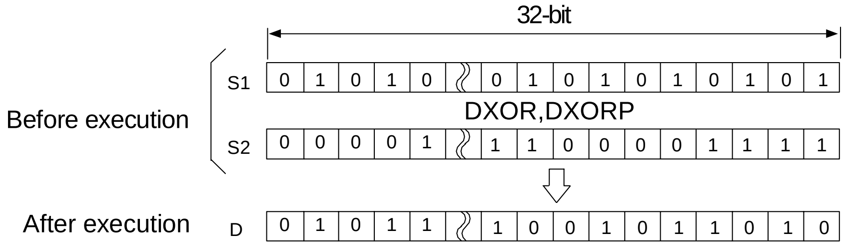
[DXOR S D], [DXORP S D]

Exclusive OR is executed for each bit of the 32-bit data designated with D and designated with S, and the results are stored in the device designated with D.



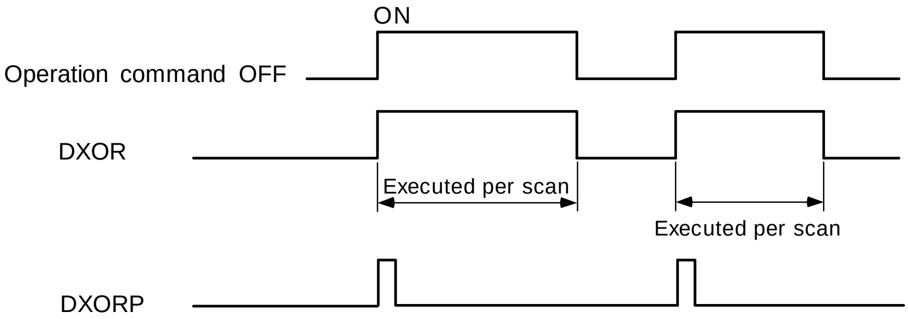
[DXOR S1 S2 D], [DXORP S1 S2 D]

Exclusive OR is executed for each bit of the 32-bit data designated with S1 and designated with S2, and the results are stored in the device designated with D.



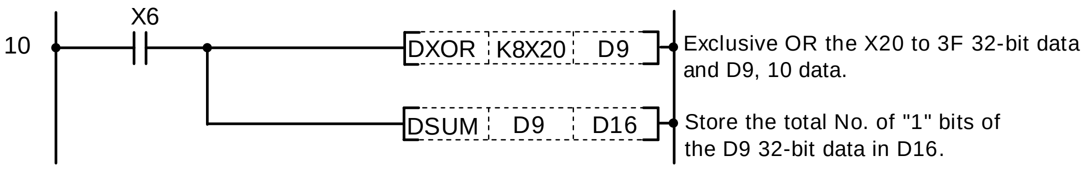
Execution conditions

The execution conditions for DXOR, DXORP are as follows.



Program example

(1) Program to compare the X20 to 3F 32-bit data and the D9, 10 data when X6 turns ON, and to store the No. of unmatched bits in D16.

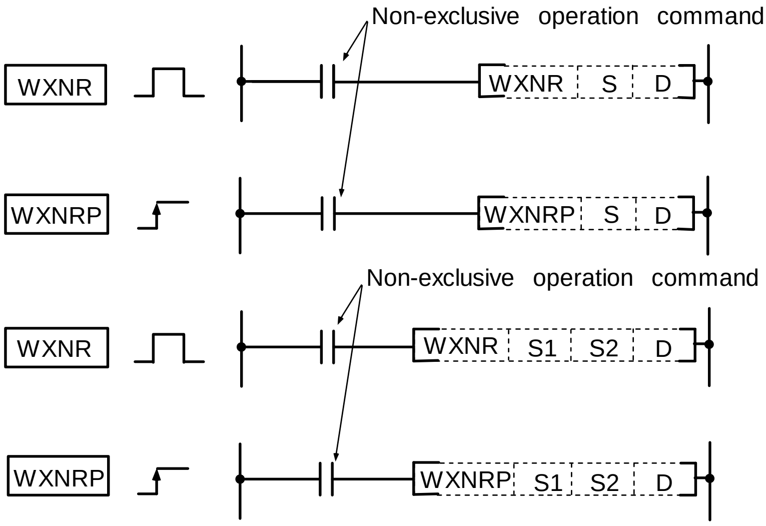


Coding

No. of steps	Com-mand	Device		
10	LD	X6		
11	DXOR	K8X20	D9	
14	DSUM	D9	D16	
18				

○ WXNR, WXNRP ... Non-exclusive OR of 16-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	4
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Setting data

S	Data to be non-exclusive ORed or No. of device where data is stored.
D	The non-exclusive OR results are stored in D device.

Setting data

S1	Data to be non-exclusive ORed or No. of device where data is stored.
S2	
D	No. of device to store non-exclusive OR results.

Function

[WXNR S D], [WXNRP S D]

Non-exclusive OR is executed for each bit of the 16-bit data designated with D and designated with S, and the results are stored in the device designated with D.

16-bit

Before execution	D	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		WXOR,WXORP															
	S	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
After execution	D	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0

[WXNR S1 S2 D], [WXNRP S1 S2 D]

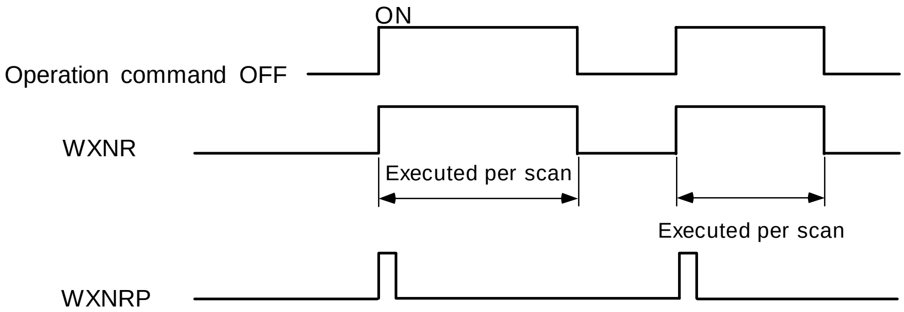
Non-Exclusive OR is executed for each bit of the 16-bit data designated with S1 and designated with S2, and the results are stored in the device designated with D.

16-bit

Before execution	S1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		WXOR,WXORP															
	S2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
After execution	D	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0

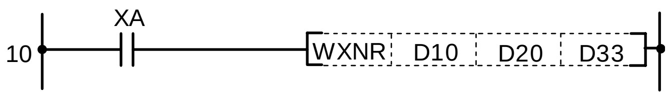
Execution conditions

The execution conditions for WXNR, WXNRP are as follows.



Program example

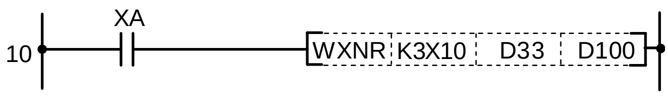
(1) Program to execute non-exclusive OR of the D10 data and D20 data when XA turns ON, and to store the results in D33.



Coding

No. of steps	Com-mand	Device		
10	LD	XA		
11	WXNR	D10	D20	D33
15				

(2) Program to execute non-exclusive OR of the X10 to 1B data and D33 data when XA turns ON, and to output the results to D100.

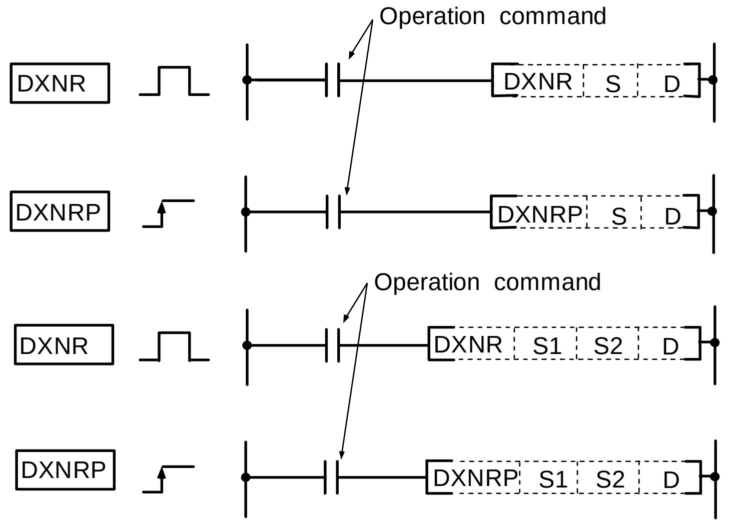


Coding

No. of steps	Com-mand	Device		
10	LD	XA		
11	WXNR	K3X10	D33	D100
15				

○ DXNR, DXNRP ... Non-exclusive OR of 32-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3/4
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						
S1	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	4/5
S2	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						



Setting data

S	Data to be non-exclusive ORed or head No. of device where data is stored.
D	The non-exclusive OR results are stored in D device.

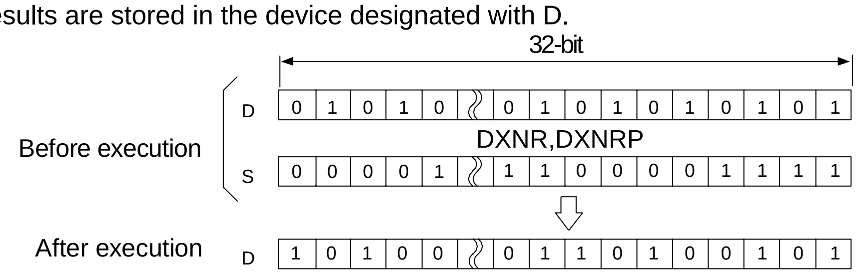
Setting data

S1 / S2	Data to be non-exclusive ORed or head No. of device where data is stored.
D	Head No. of device to store non-exclusive OR results.

Function

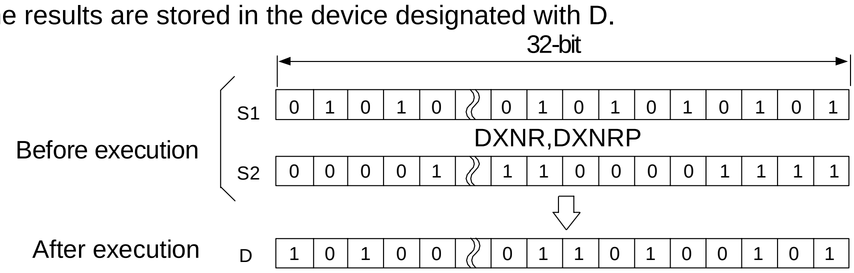
[DXNR S D], [DXNRP S D]

Non-exclusive OR is executed for each bit of the 32-bit data designated with D and designated with S, and the results are stored in the device designated with D.



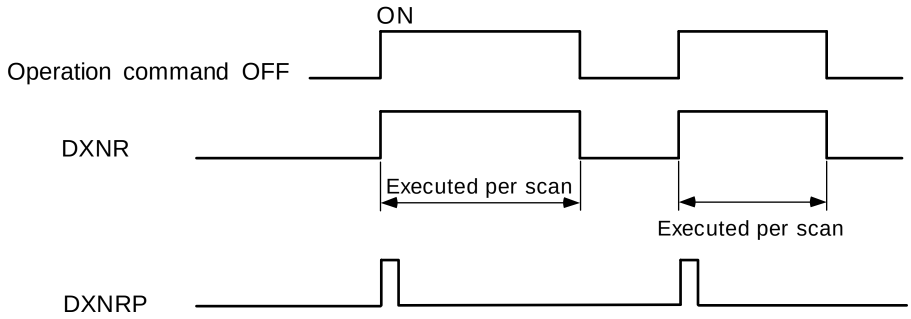
[DXNR S1 S2 D], [DXNRP S1 S2 D]

Non-exclusive OR is executed for each bit of the 32-bit data designated with S1 and designated with S2, and the results are stored in the device designated with D.



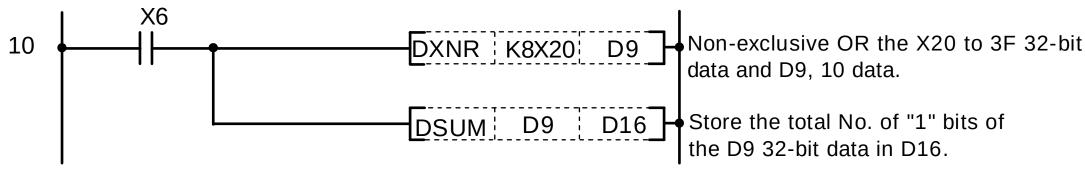
Execution conditions

The execution conditions for DXNR, DXNRP are as follows.



Program example

(1) Program to compare the X20 to 3F 32-bit data and the D9, 10 data when X6 turns ON, and to store the No. of matched bits in D16.

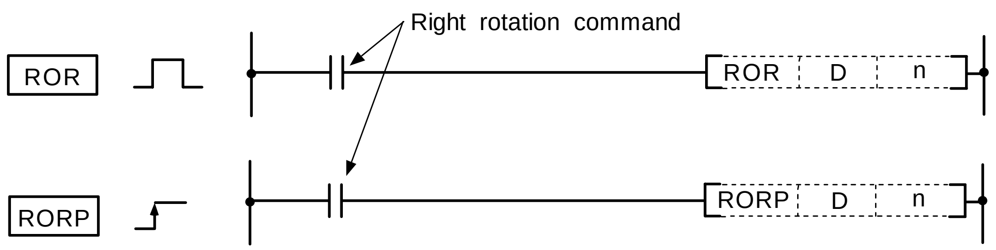


Coding

No. of steps	Com-mand	Device		
10	LD	X6		
11	DXNR	K8X20	D9	
14	DSUM	D9	D16	
18				

○ ROR, RORP ... Right rotation of 16-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	3
n	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				

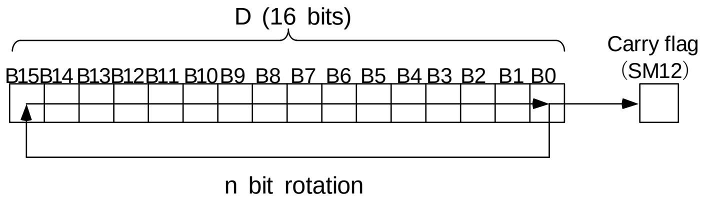


Setting data

D	No. of device where right rotation data is stored.
n	Times (0 to 15)

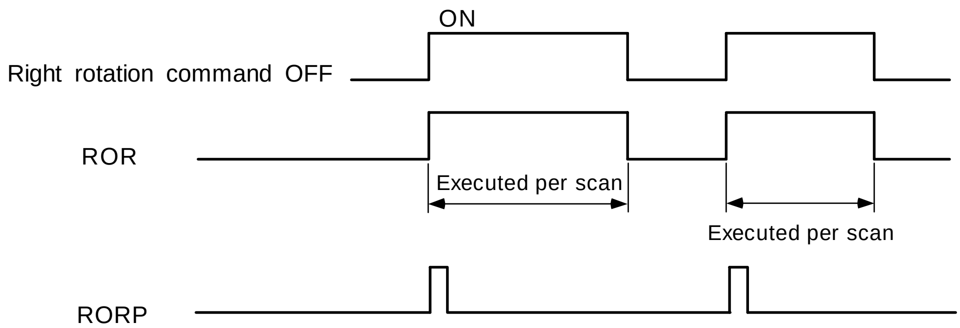
Function

The 16-bit data designated with D is rotated n bits to the right excluding the carry flag.



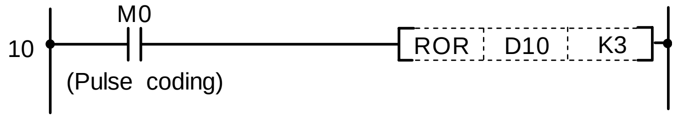
Execution conditions

The execution conditions for the ROR, RORP are as shown below.



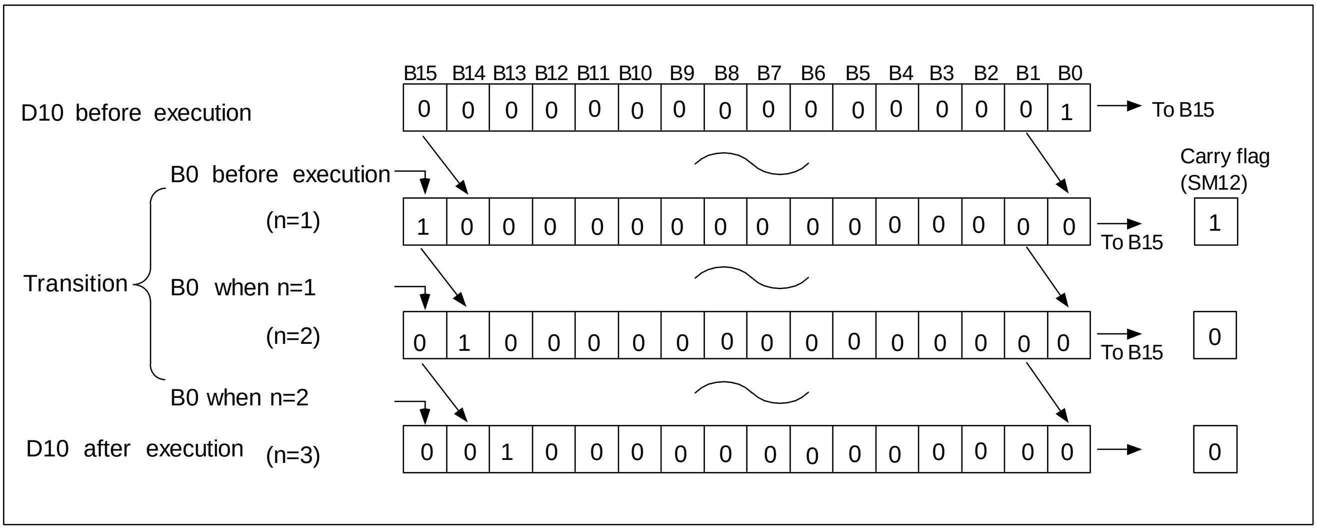
Program example

(1) Program to rotate the D10 data 3 bits to the right when M0 turns ON.



Coding

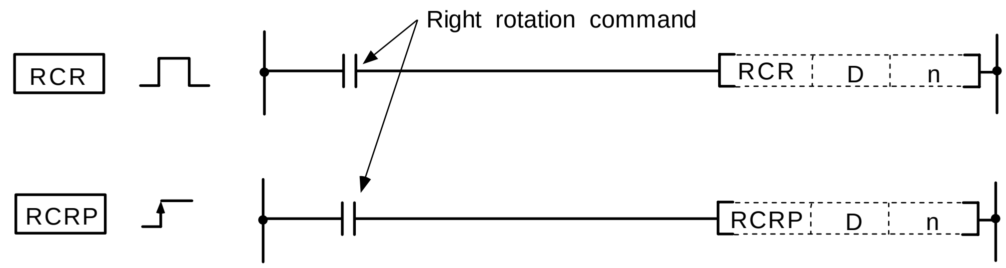
No. of steps	Com-mand	Device		
10	LD	M0		
11	ROR	D10	K3	
14				



Right rotation of data using ROR command

○ RCR, RCRP ... Right rotation of 16-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	3
n	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				

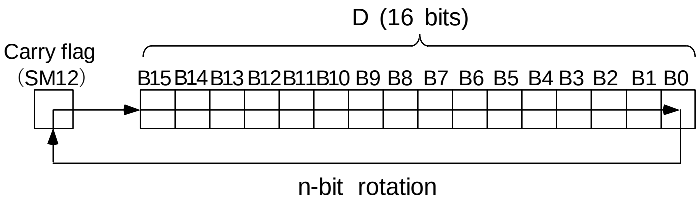


Setting data

D	No. of device where right rotation data is stored.
n	Times (0 to 15)

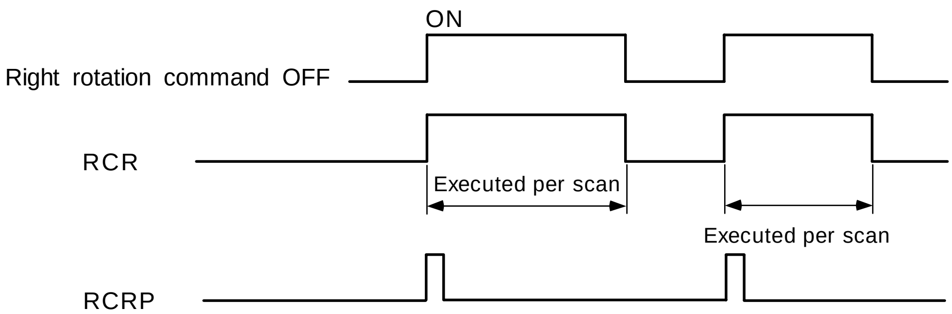
Function

The 16-bit data designated with D is rotated n bits to the right including the carry flag.
The carry flag must be set to 1 or 0 before executing RCR, RCRP.



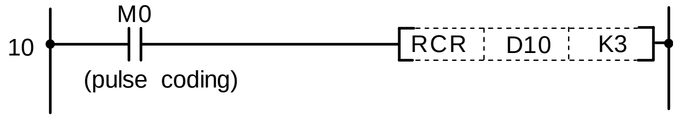
Execution conditions

The execution conditions for the RCR, RCRP are as shown below.



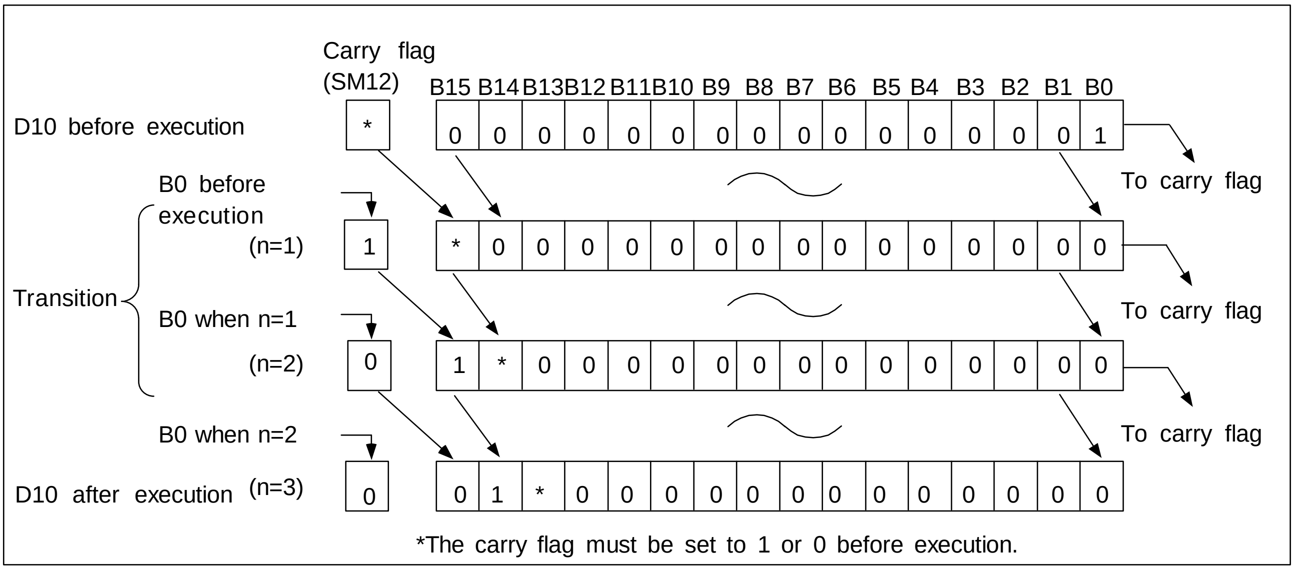
Program example

(1) Program to rotate the D10 data 3 bits to the right when M0 turns ON.



Coding

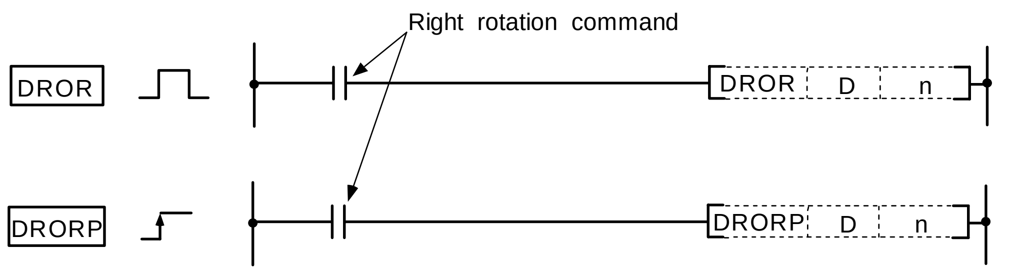
No. of steps	Com-mand	Device		
10	LD	M0		
11	RCR	D10	K3	
14				



Right rotation of data using RCR command

○ DROR, DRORP ... Right rotation of 32-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	3
n	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				

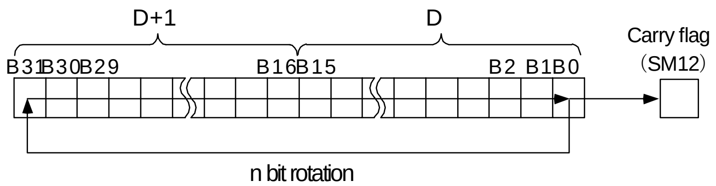


Setting data

D	Head No. of device where right rotation data is stored.
n	Times (0 to 31)

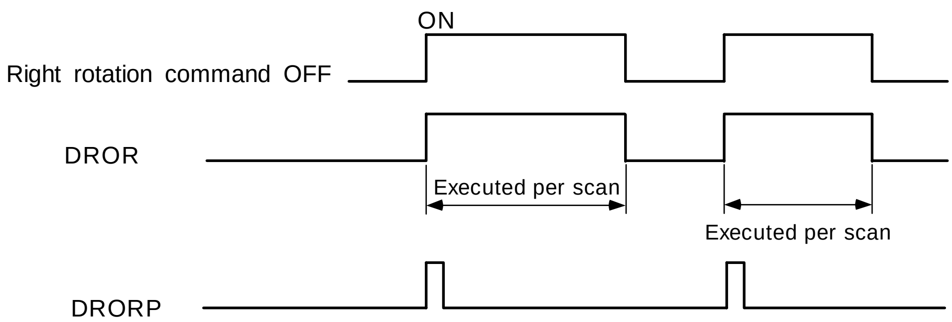
Function

The 32-bit data designated with D is rotated n bits to the right excluding the carry flag.



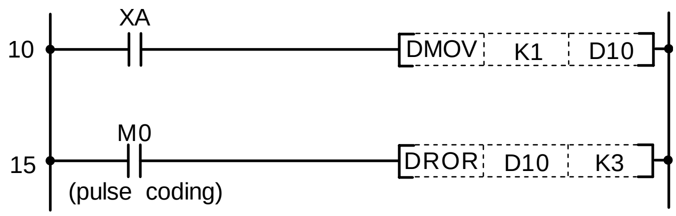
Execution conditions

The execution conditions for the DROR, DRORP are as shown below.



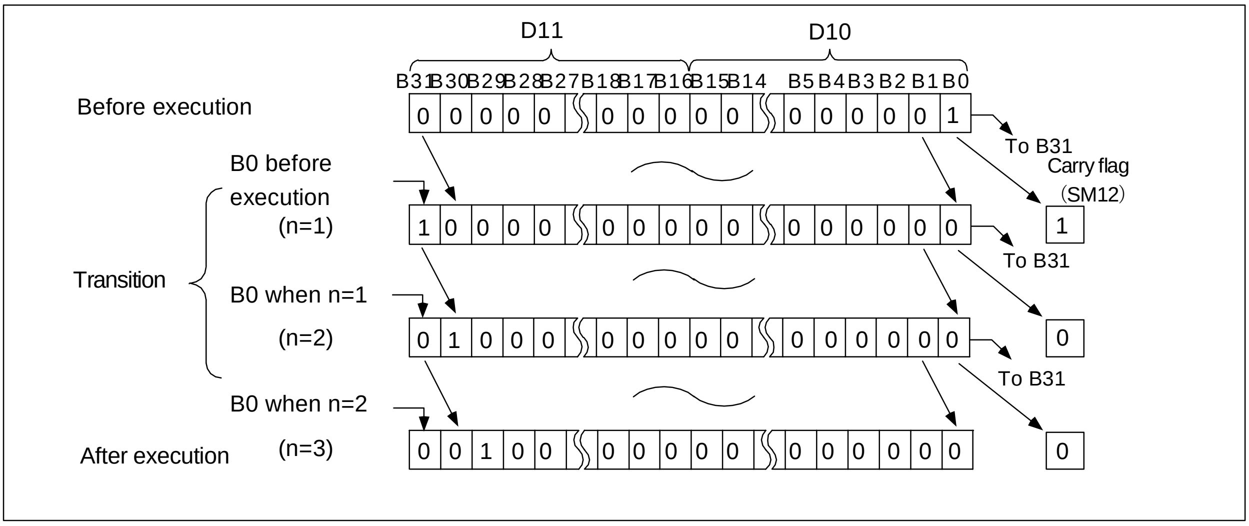
Program example

(1) Program to rotate the D10, 11 data 3 bits to the right when M0 turns ON.



Coding

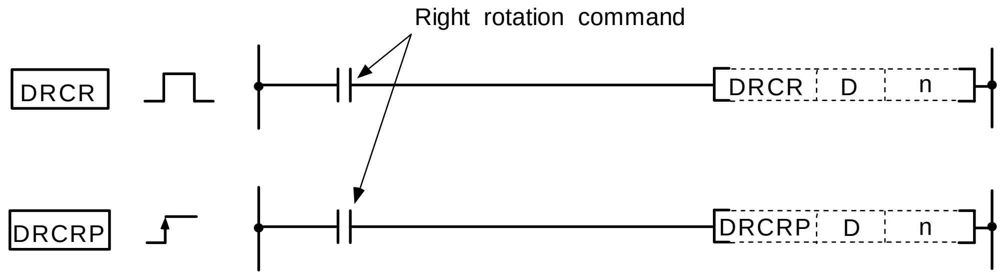
No. of steps	Com-mand	Device		
10	LD	XA		
11	DMOV	K1	D10	
15	LD	M0		
16	DROR	D10	K3	
19				



Right rotation of data using DROR command

○ DRCR, DRCRP ... Right rotation of 32-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	3
n	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				

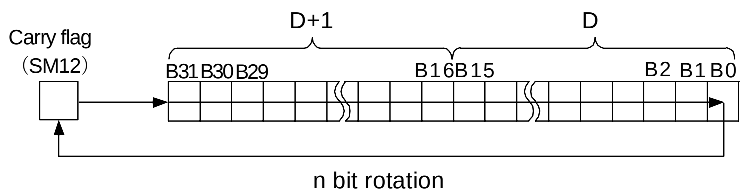


Setting data

D	Head No. of device where right rotation data is stored.
n	Times (0 to 31)

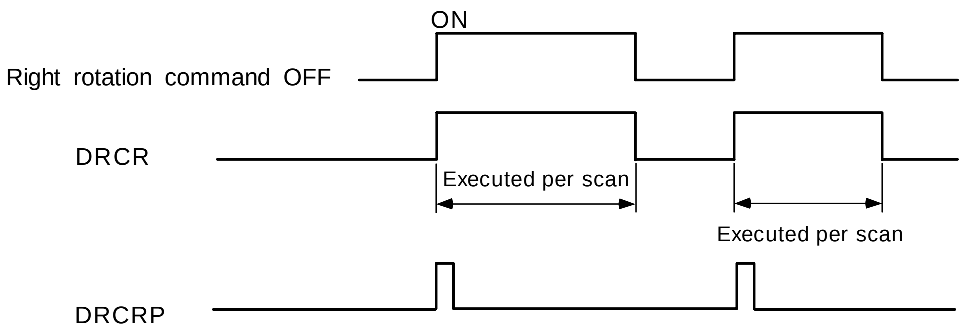
Function

The 32-bit data designated with D is rotated n bits to the right including the carry flag.
The carry flag must be set to 1 or 0 before executing DRCR, DRCRP.



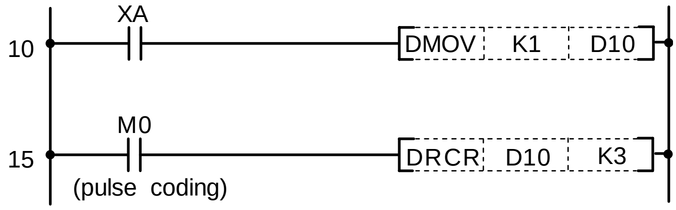
Execution conditions

The execution conditions for the DRCR, DRCRP are as shown below.



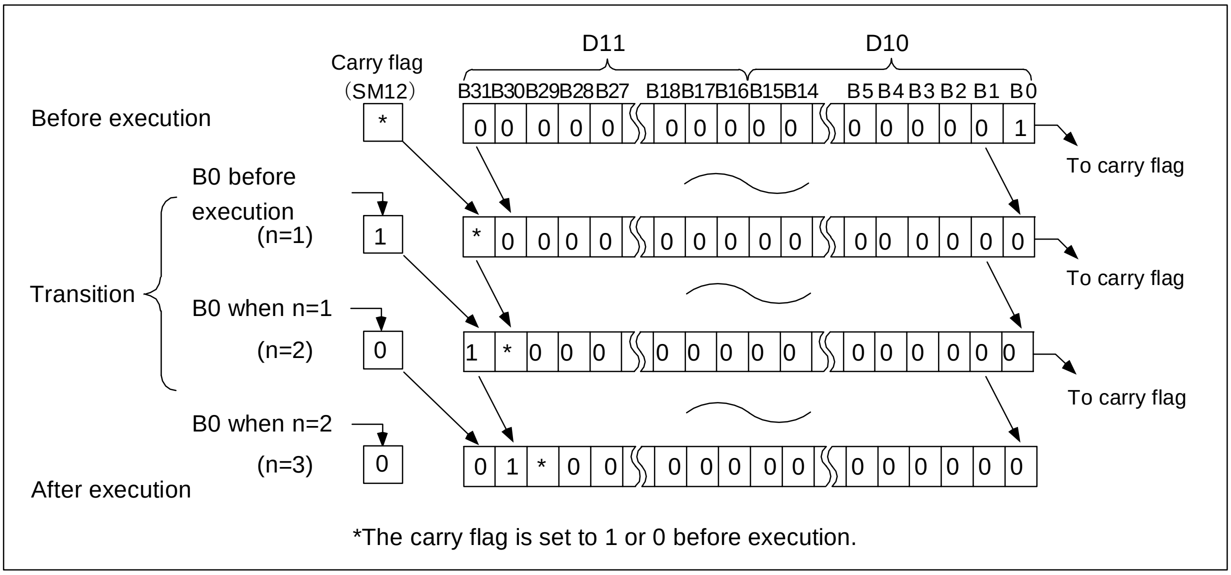
Program example

(1) Program to rotate the D10, 11 data 3 bits to the right when M0 turns ON.



Coding

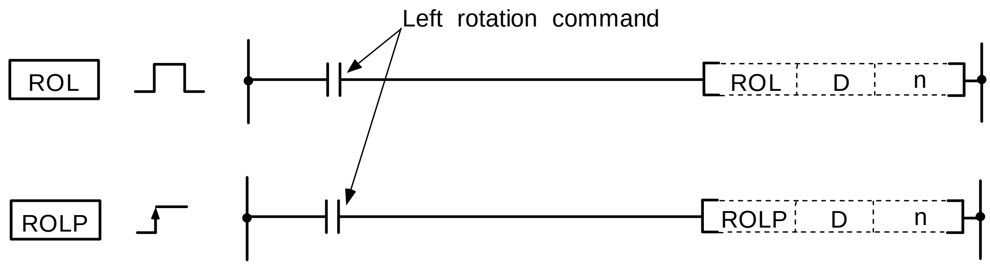
No. of steps	Com-mand	Device		
10	LD	XA		
11	DMOV	K1	D10	
15	LD	M0		
16	DRCR	D10	K3	
19				



Right rotation of data using DRCR command

○ ROL, ROLP ... Left rotation of 16-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	3
n	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				

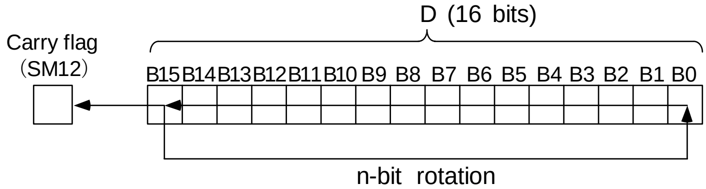


Setting data

D	No.of device where left rotation data is stored.
n	Times (0 to 15)

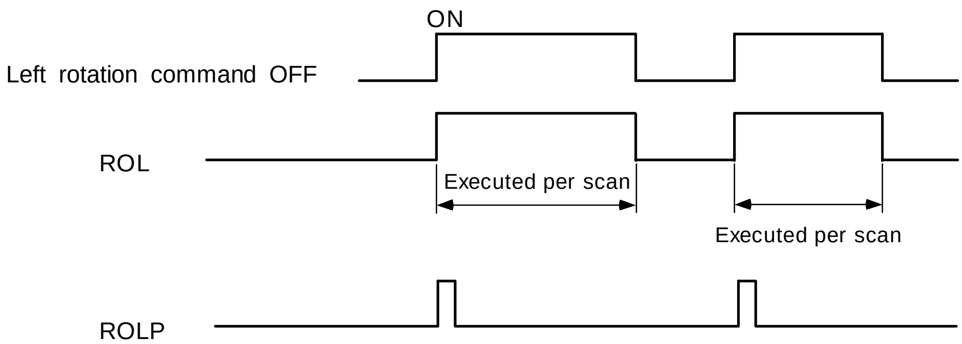
Function

The 16-bit data designated with D is rotated n bits to the left excluding the carry flag.
The carry flag is set to 1 or 0 after executing ROL, ROLP.



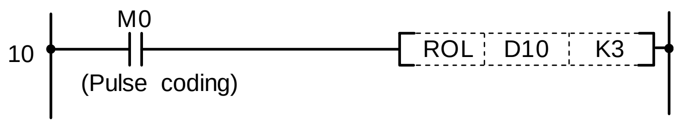
Execution conditions

The execution conditions for the ROL, ROLP are as shown below.



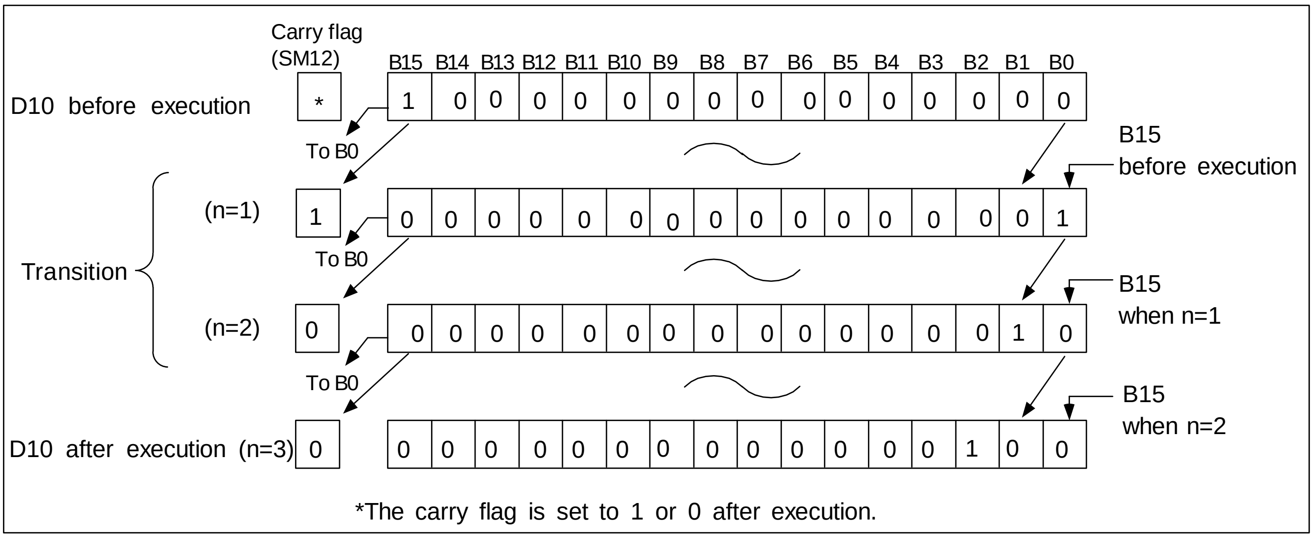
Program example

Program to rotate the D10 data 3 bits to the left when M0 turns ON.



Coding

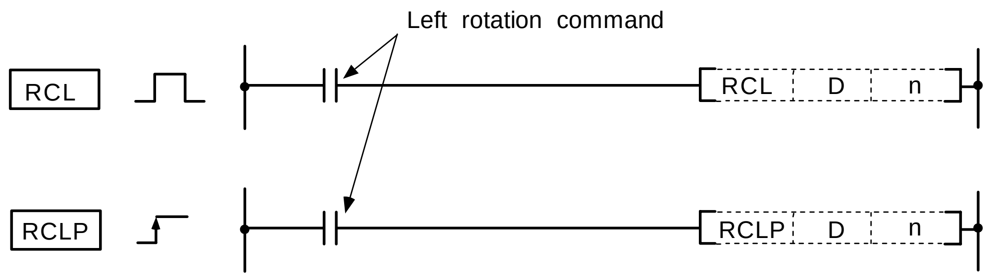
No. of steps	Com-mand	Device		
10	LD	M0		
11	ROL	D10	K3	
14				



Left rotation of data using ROL command

○ RCL, RCLP ... Left rotation of 16-bit data

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	3
n	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				

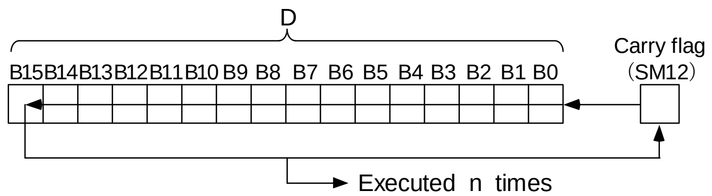


Setting data

D	No. of device where left rotation data is stored.
n	Times (0 to 15)

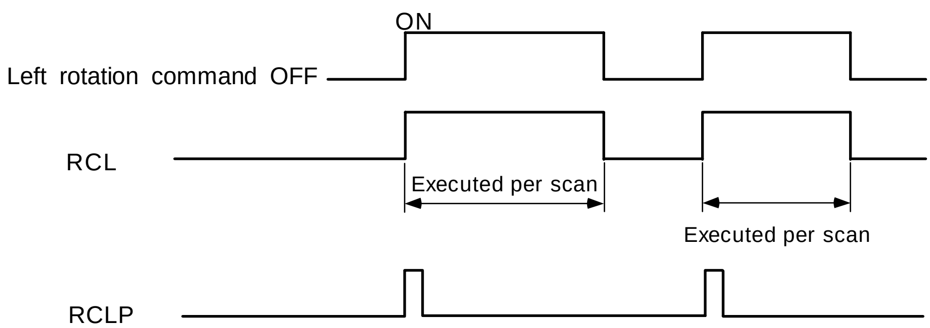
Function

The 16-bit data designated with D is rotated n bits to the left including the carry flag.
The carry flag must be set to 1 or 0 before executing RCL, RCLP.



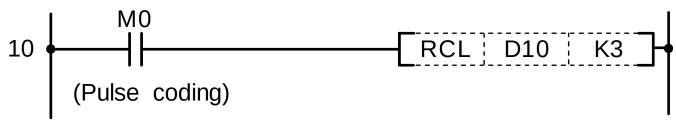
Execution conditions

The execution conditions for the RCL, RCLP are as shown below.



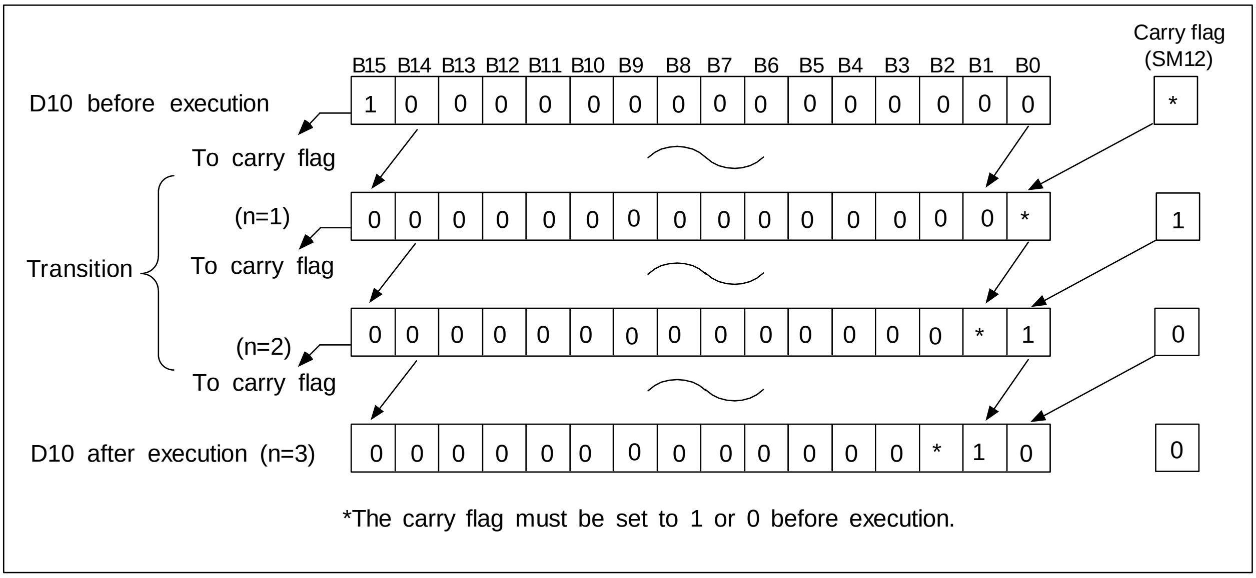
Program example

Program to rotate the D10 data 3 bits to the left when M0 turns ON.



Coding

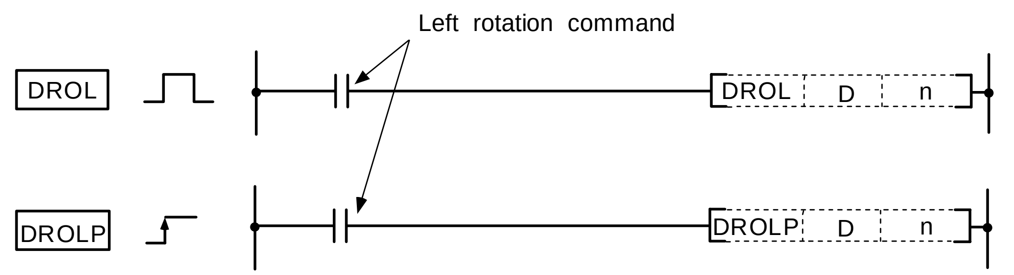
No. of steps	Com-mand	Device		
10	LD	M0		
11	RCL	D10	K3	
14				



Left rotation of data using RCL command

○ DROL, DROLP ... Left rotation of 32-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	3
n	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				

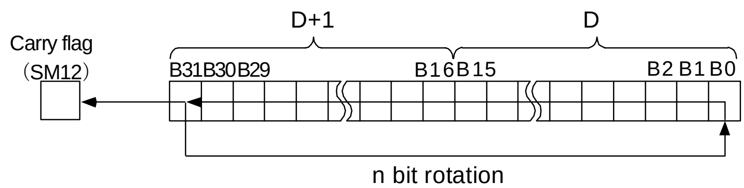


Setting data

D	Head No. of device where left rotation data is stored.
n	Times (0 to 31)

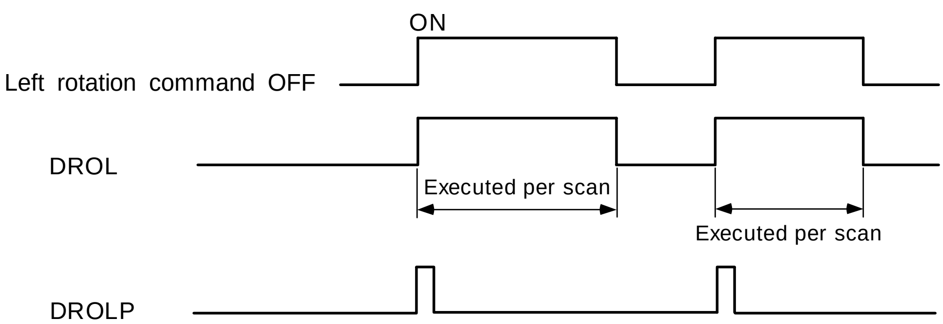
Function

The 32-bit data designated with D is rotated n bits to the left excluding the carry flag.



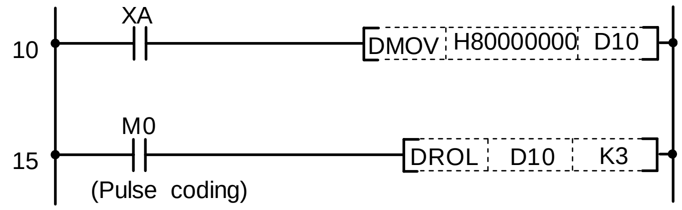
Execution conditions

The execution conditions for the DROL, DROLP are as shown below.



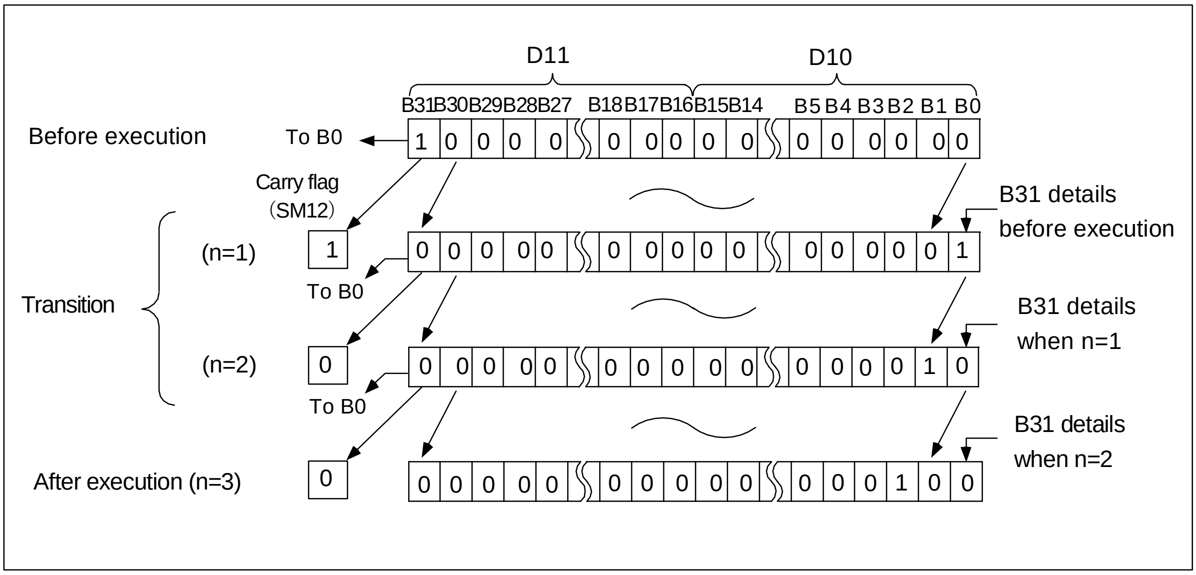
Program example

Program to rotate the D10, 11 data 3 bits to the left when M0 turns ON.



Coding

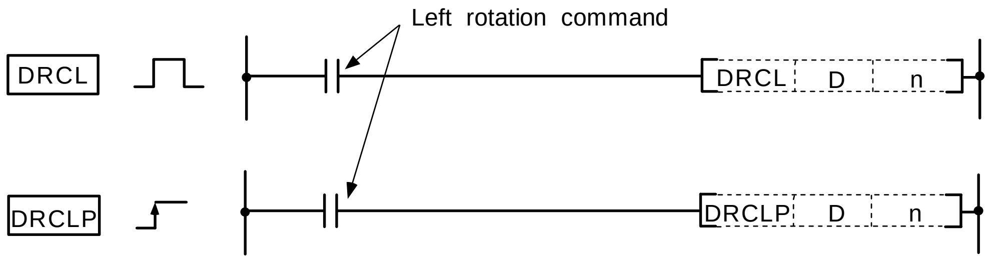
No. of steps	Com-mand	Device		
10	LD	XA		
11	DMOV	H80000000	D0	
15	LD	M0		
16	DROL	D10	K3	
19				



Left rotation of data using DROL command

○ **DRCL, DRCLP ... Left rotation of 32-bit data**

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	3
n	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				

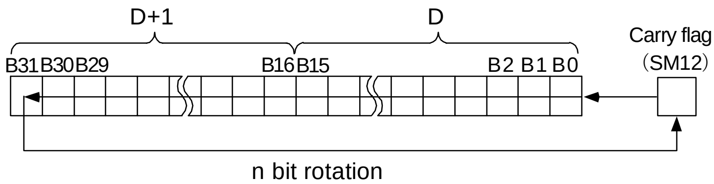


Setting data

D	Head No. of device where left rotation data is stored.
n	Times (0 to 31)

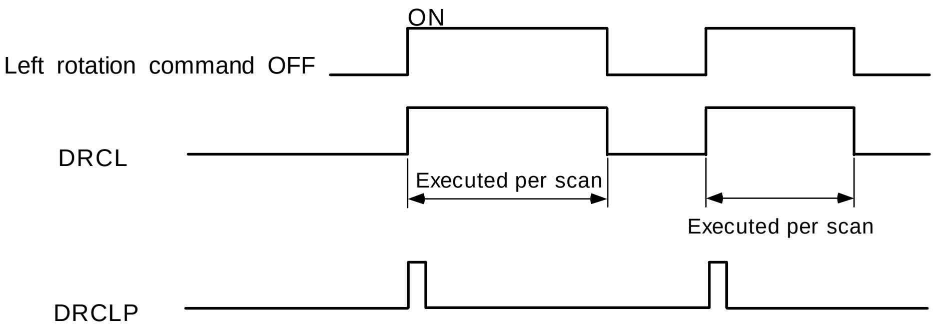
Function

The 32-bit data designated with D is rotated n bits to the left including the carry flag.
The carry flag must be set to 1 or 0 before executing DRCL.



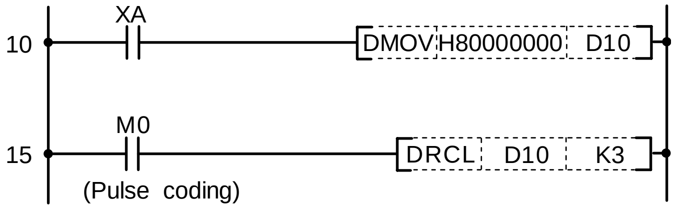
Execution conditions

The execution conditions for the DRCL, DRCLP are as shown below.



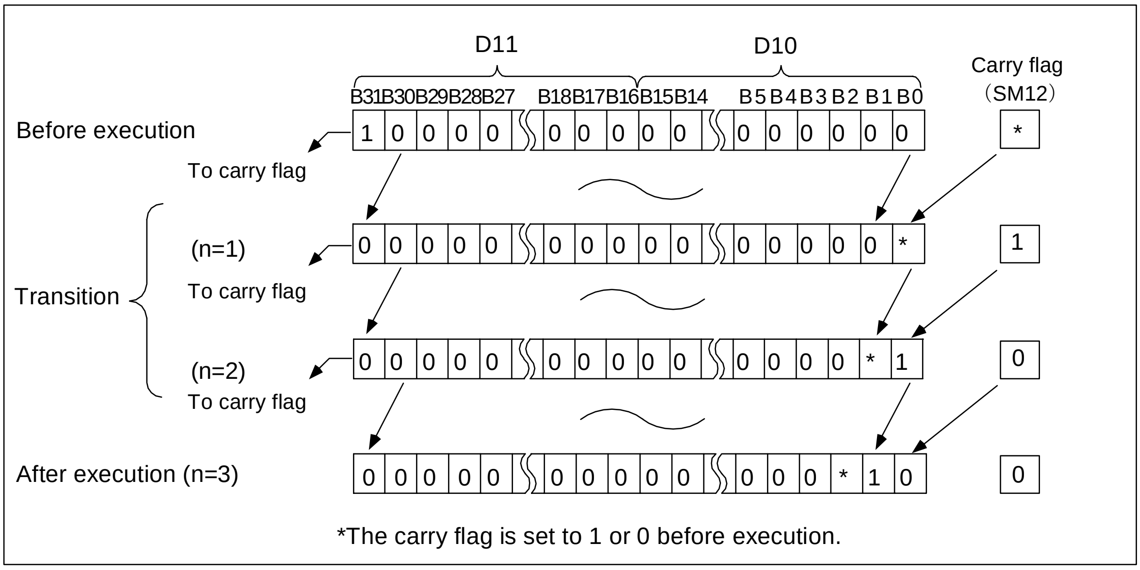
Program example

Program to rotate the D10, 11 data 3 bits to the left when M0 turns ON.



Coding

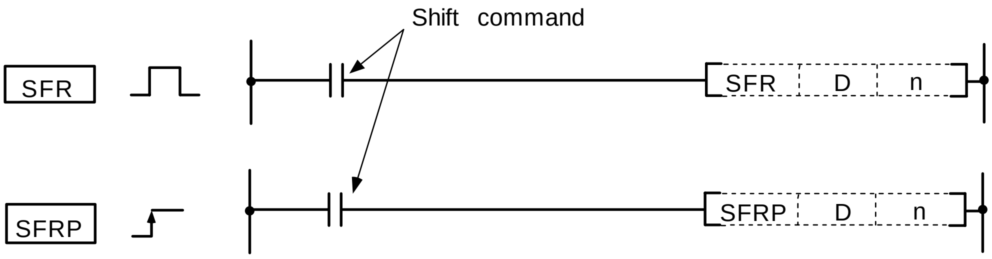
No. of steps	Command	Device		
10	LD	XA		
11	DMOV	H80000000	D10	
15	LD	M0		
16	DRCL	D10	K3	
19				



Left rotation of data using DRCL command

○ **SFR, SFRP ... Right shift of 16-bit data**

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	3
n	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				

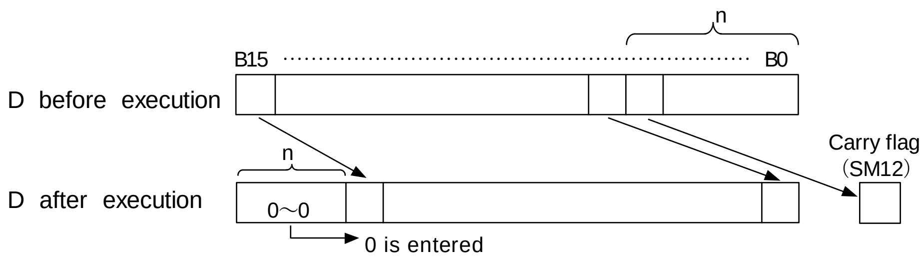


Setting data

D	No. of device where shift data is stored.
n	No. of shifts

Function

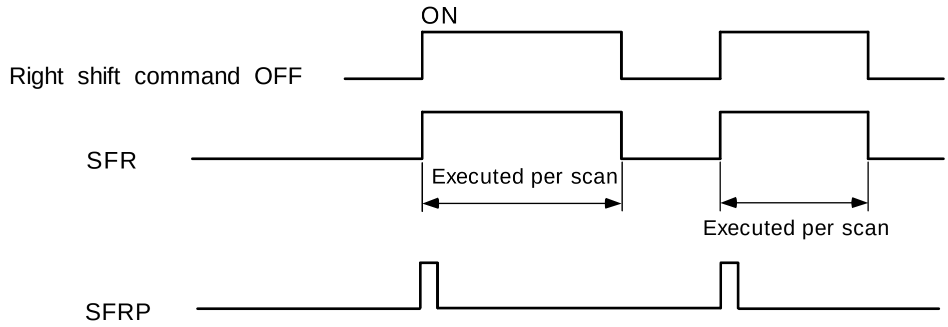
(1) The 16-bit data of the device designated with D is shifted n bits to the right.



(2) n bits from the highest order are set to 0.
(3) The T, C shift will be a current value (attribute value or count value) shift. (Shifting with the setting value is not possible.)

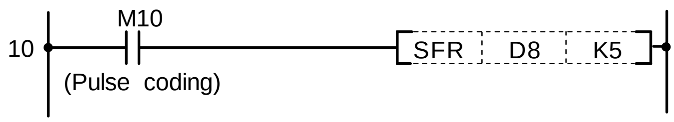
Execution conditions

The execution conditions for SFR, SFRP are as shown below.



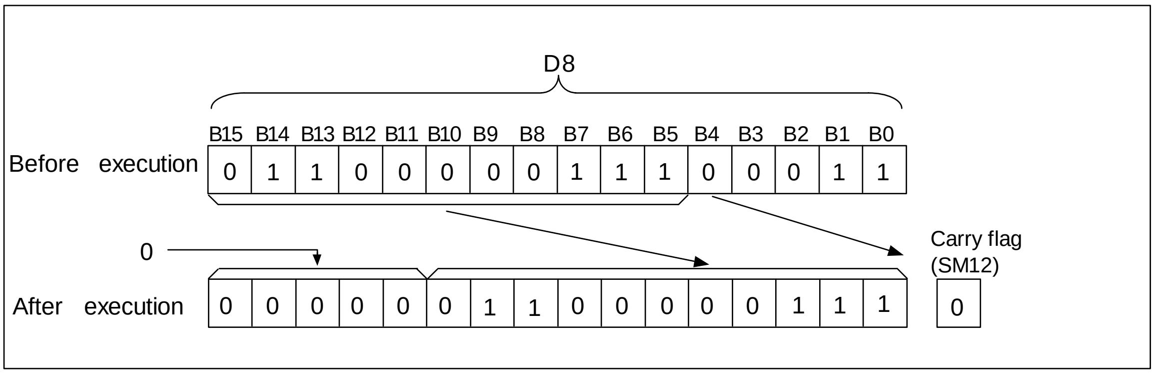
Program example

Program that shifts the D8 data 5 bits to the right when M10 turns ON.



Coding

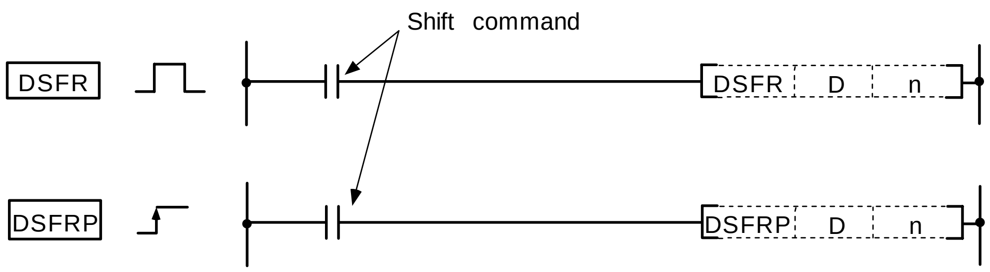
No. of steps	Com-mand	Device		
10	LD	M10		
11	SFR	D8	K5	
14				



Right shift of data with SFR command (word device)

○ DSFR, DSFRP ... Right shift of word device in batch

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D											○	○	○	○	○	○		○						4
n											○	○	○	○	○	○		○	○	○			○	

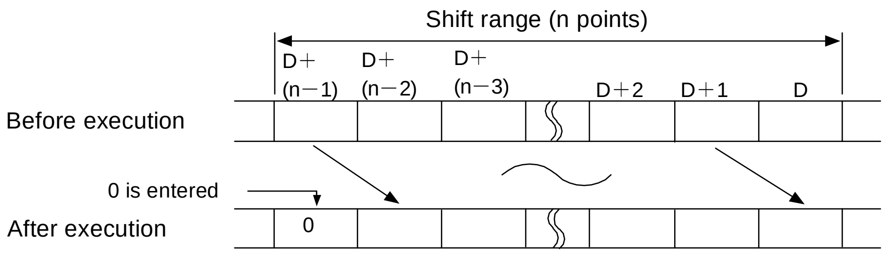


Setting data

D	Head No. of device to be shifted
n	Shift range

Function

(1) n points starting at the head of the device designated with D are shifted one point to the right.

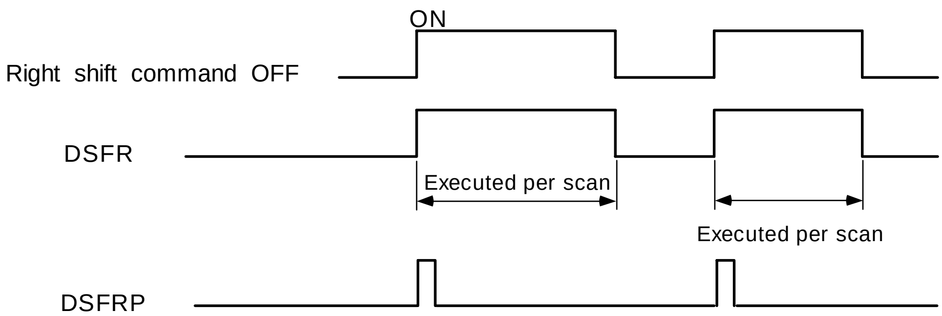


(2) The highest order device is set to 0.

(3) The T, C shift will be a current value (attribute value or count value) shift. (Shifting with the setting value is not possible.)

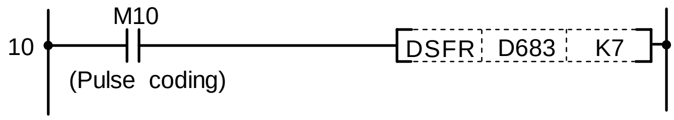
Execution conditions

The execution conditions of DSFR, DSFRP are as shown below.



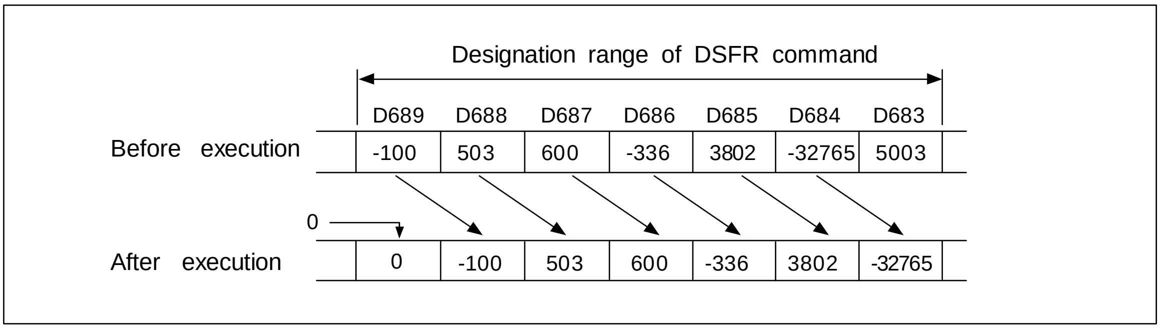
Program example

(1) Program to shift the D668 to 689 data to the right when M10 turns ON.



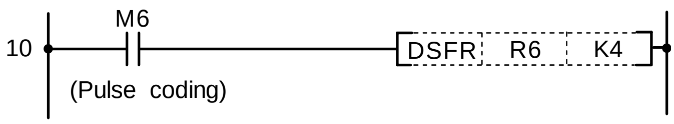
Coding

No. of steps	Com-mand	Device		
10	LD	M10		
11	DSFR	D683	K7	
15				



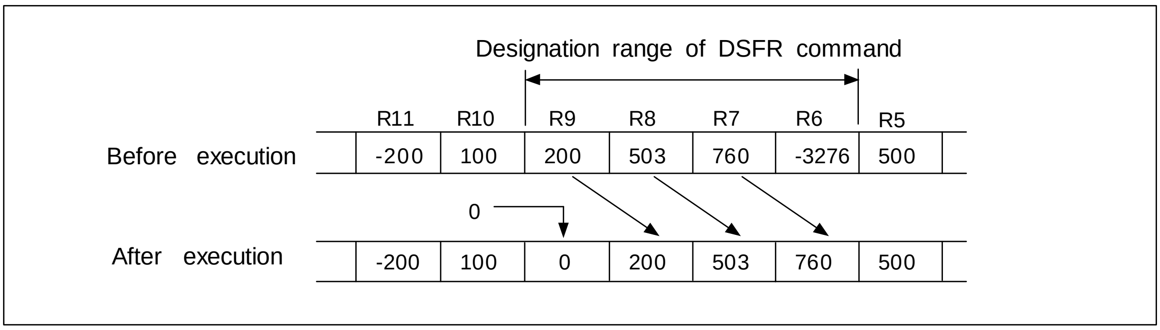
Right shift of data with DSFR command

(2) Program to shift the R6 to 9 data to the right when M6 turns ON.



Coding

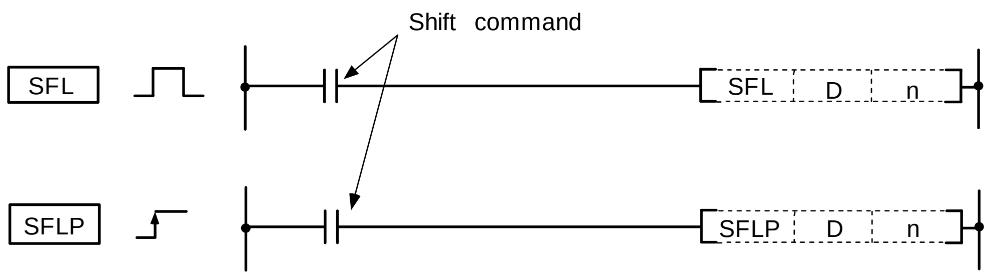
No. of steps	Com-mand	Device		
10	LD	M6		
11	DSFR	R6	K4	
15				



Right shift of data with DSFR command

○ SFL, SFLP ... Left shift of 16-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○				○	○	3
n		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○				

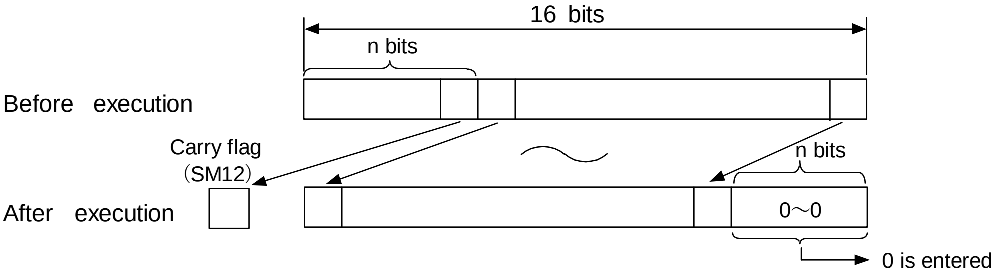


Setting data

D	No. of device where shift data is stored.
n	No. of shifts

Function

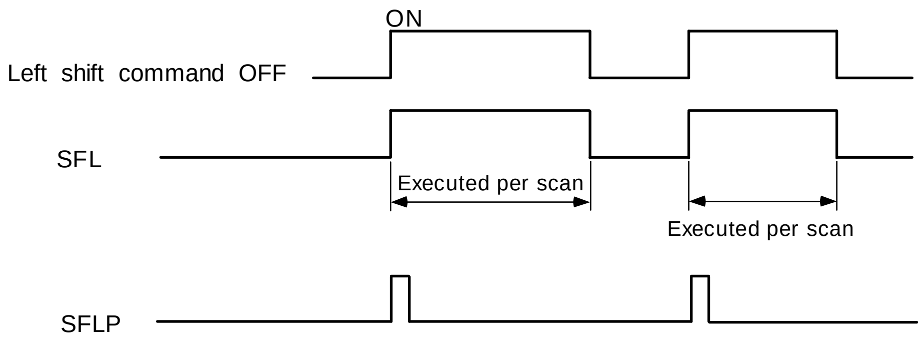
(1) The 16-bit data of the device designated with D is shifted n bits to the left.
(2) n bits from the lowest order are set to 0.



(3) The T, C shift will be a current value (attribute value or count value) shift. (Shifting with the setting value is not possible.)

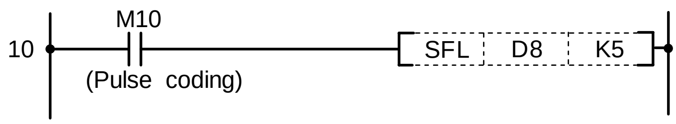
Execution conditions

The execution conditions for SFL, SFLP are as shown below.



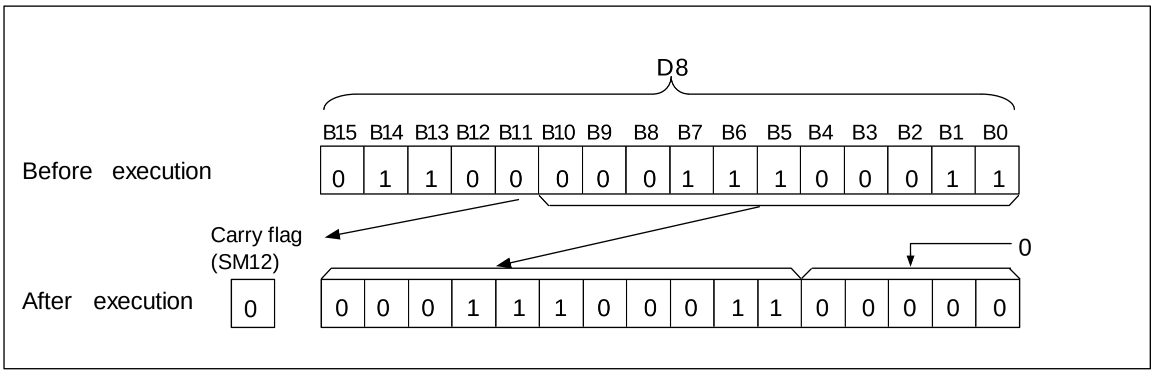
Program example

(1) Program that shifts the D8 data 5 bits to the left when M10 turns ON.



Coding

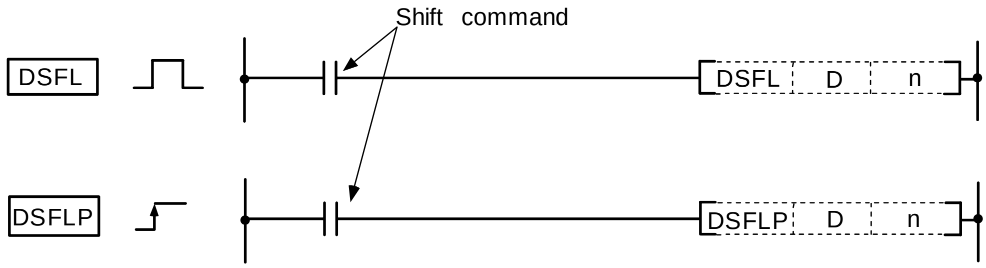
No. of steps	Com-mand	Device		
10	LD	M10		
11	SFL	D8	K5	
14				



Left shift of data with SFL command (word device)

○ DSFL, DSFLP ... Left shift of word device in batch

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D											○	○	○	○	○	○		○						4
n											○	○	○	○	○	○		○	○	○			○	

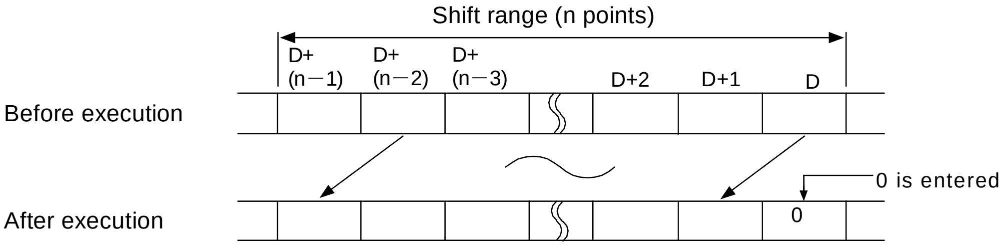


Setting data

D	Head No. of device to be shifted.
n	Shift range

Function

(1) n points starting at the head of the device designated with D are shifted one point to the left.

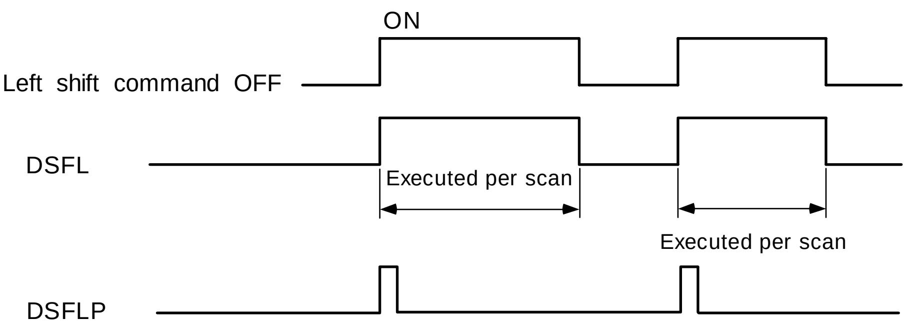


(2) The lowest order device is set to 0.

(3) The T, C shift will be a current value (attribute value or count value) shift. (Shifting with the setting value is not possible.)

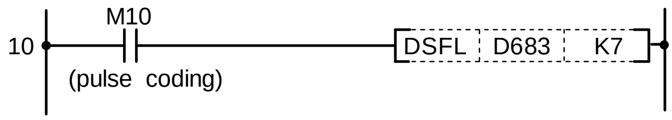
Execution conditions

The execution conditions of DSFL, DSFLP are as shown below.



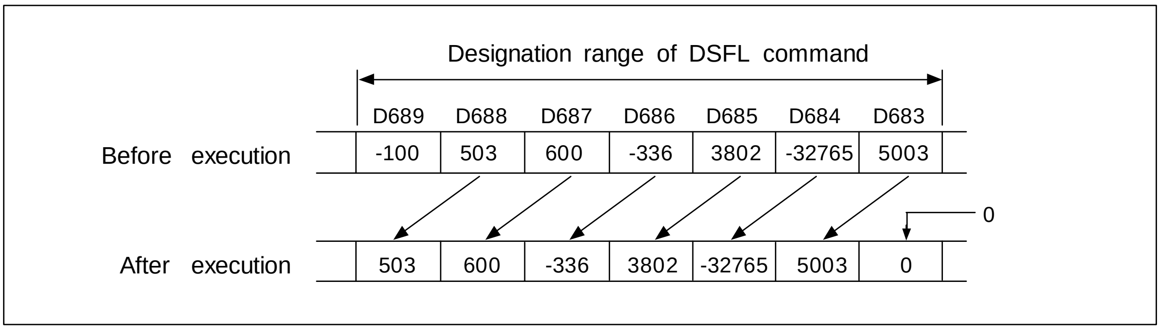
Program example

(1) Program to shift the D683 to 689 data to the left when M10 turns ON.



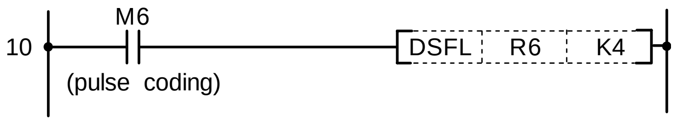
Coding

No. of steps	Com-mand	Device		
10	LD	M10		
11	DSFL	D683	K7	
15				



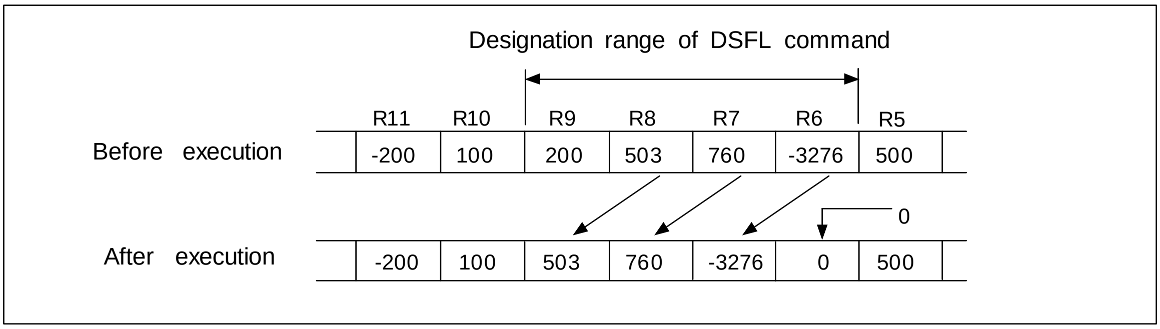
Left shift of data with DSFL command

(2) Program to shift the R6 to 9 data to the left when M6 turns ON.



Coding

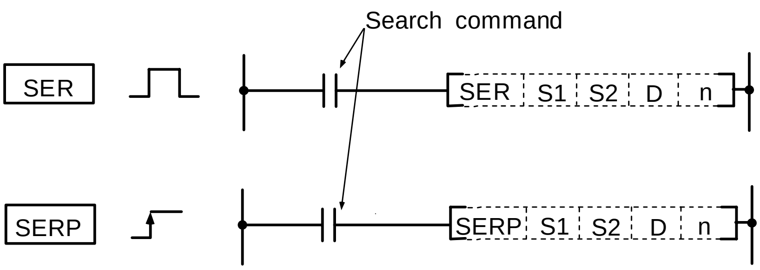
No. of steps	Com-mand	Device		
10	LD	M6		
11	DSFL	R6	K4	
15				



Left shift of data with DSFL command

○ SER, SERP ... Search of 16-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1											○	○	○	○	○	○		○					○	6
S2											○	○	○	○	○	○		○						
D											○	○	○	○	○	○		○					○	
n											○	○	○	○	○	○		○	○	○			○	



Setting data

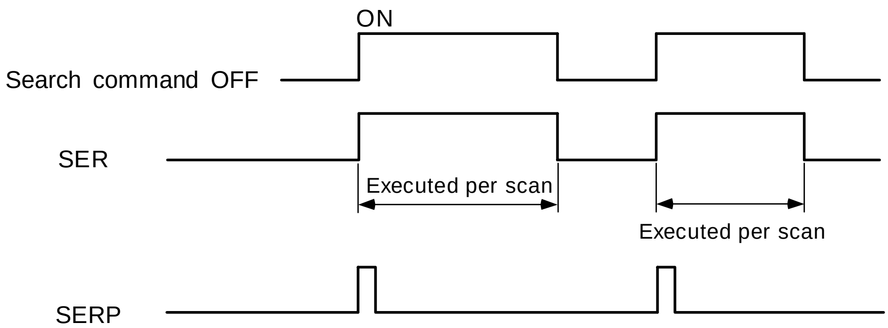
S1	No. of device where search data is stored.
S2	Head No. of device to be searched.
D	Head No. of device where search results are stored.
n	No. of devices to be searched.

Function

(1) Using the 16-bit data of the device designated with S1 as the keyword, the n points from the 16-bit data of the device designated with S2 are searched.
(2) The number of data matching the keyword is stored in D+1. The relative position of the device containing the first matched data counted from S2 is stored in D.
(3) When n is a negative value, it is interpreted as 0.
(4) No process is executed when n = 0.

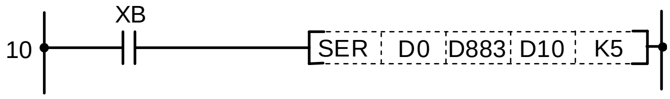
Execution conditions

The execution conditions for SER, SERP are as shown below.



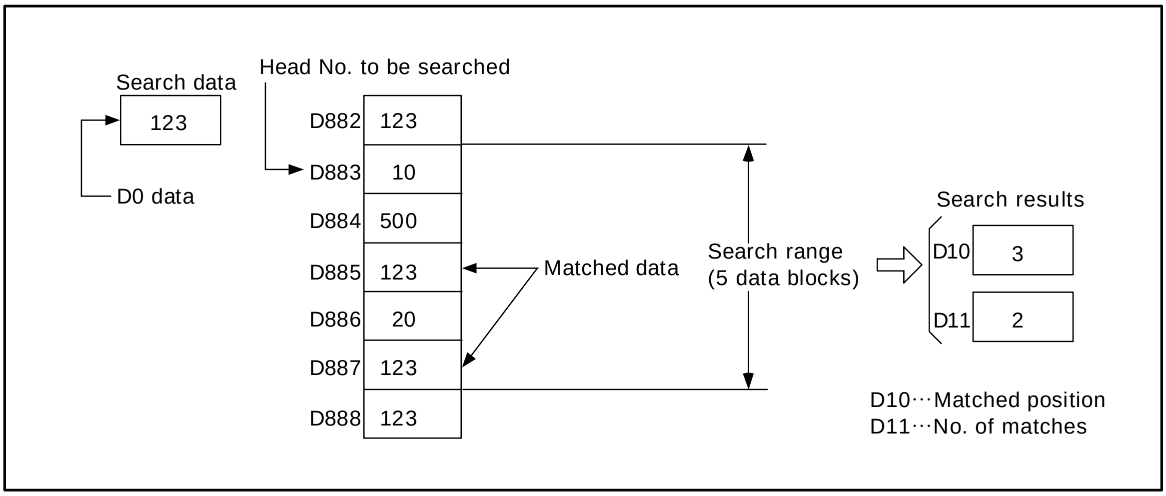
Program example

Program to compare the D883 to D887 data with "123" when XB turns ON.



Coding

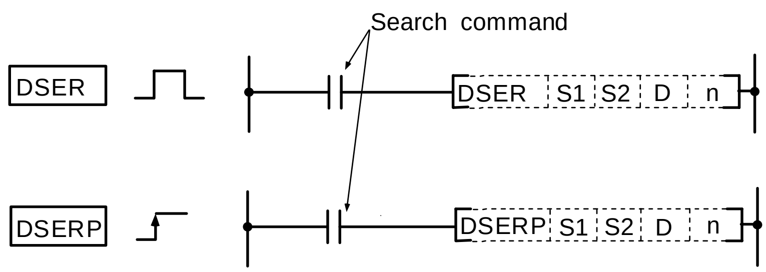
No. of steps	Com-mand	Device			
10	LD	XB			
11	SER	D0	D883	D10	K5
17					



Search of data using SER command

○ DSER, DSERP ... Search of 32-bit data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1											○	○	○	○	○	○		○					○	6
S2											○	○	○	○	○	○		○						
D											○	○	○	○	○	○		○					○	
n											○	○	○	○	○	○		○	○	○			○	



Setting data

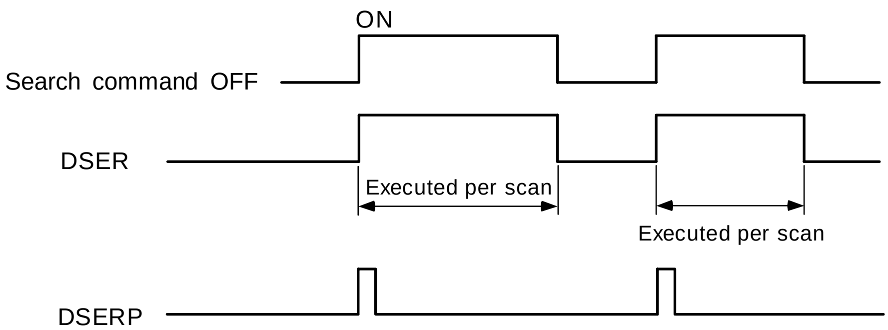
S1	No. of device where search data is stored.
S2	Head No. of device to be searched.
D	Head No. of device where search results are stored.
n	No. of devices to be searched.

Function

(1) Using the 32-bit data of the device designated with S1 as the keyword, the n points from the 32-bit data of the device designated with S2 are searched.
(2) The number of data matching the keyword is stored in D+1. The relative position of the device containing the first matched data counted from S2 is stored in D.
(3) When n is a negative value, it is interpreted as 0.
(4) No process is executed when n = 0.

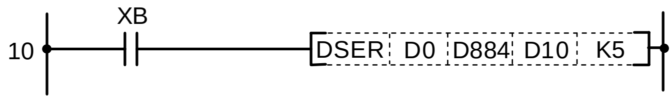
Execution conditions

The execution conditions for DSER, DSERP are as shown below.



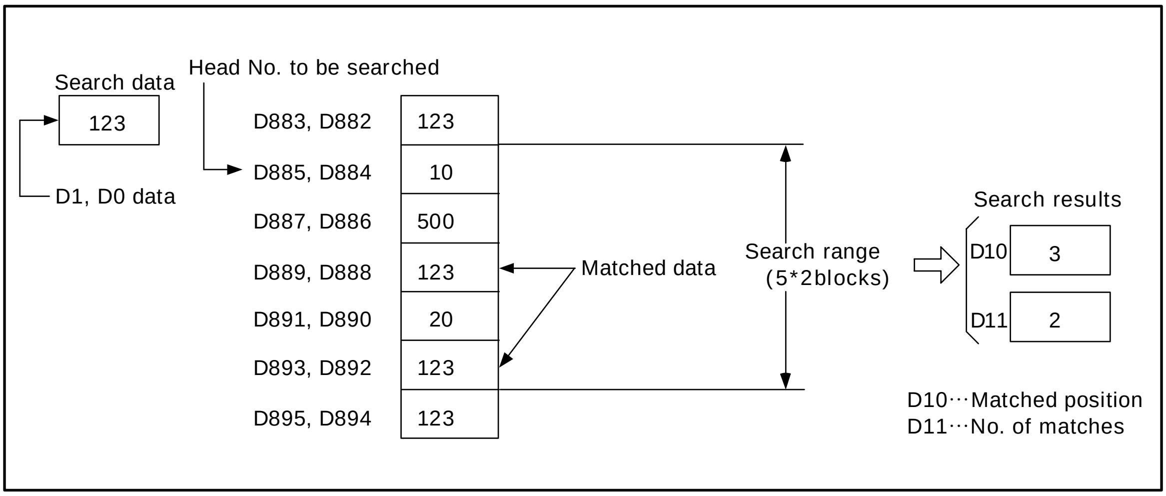
Program example

Program to compare the D884 to D893 data with "123" when XB turns ON.



Coding

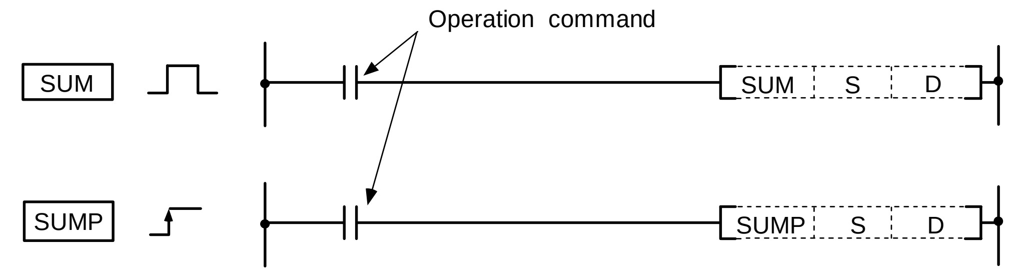
No. of steps	Com-mand	Device			
10	LD	XB			
11	DSER	D0	D884	D10	K5
17					



Search of data using DSER command

○ SUM, SUMP ... Count of No. of 16-bit data set to 1

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S											○	○	○	○	○	○		○					○	4
D											○	○	○	○	○	○		○						

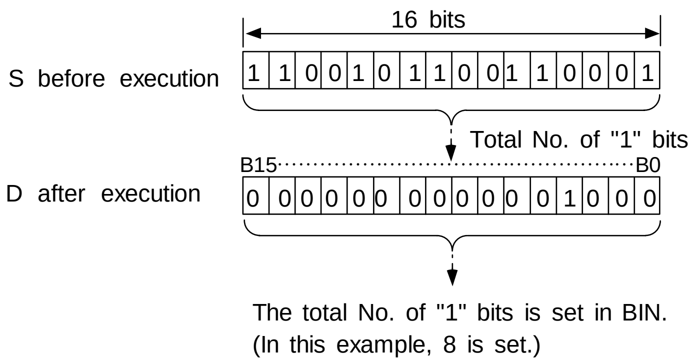


Setting data

S	Head No. of device to count the total No. of bits set to 1.
D	Head No. of device where the total No. of bits is stored.

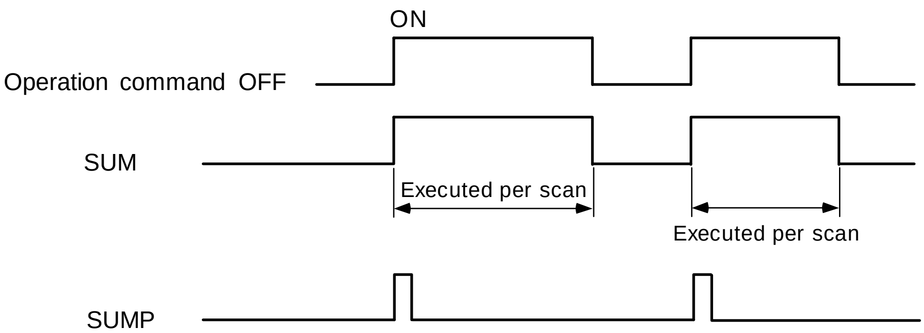
Function

The total No. of bits that are set to 1 in the 16-bit data of the device designated with S is stored in D.



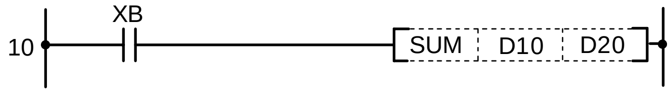
Execution conditions

The execution conditions for SUM, SUMP are as shown below.



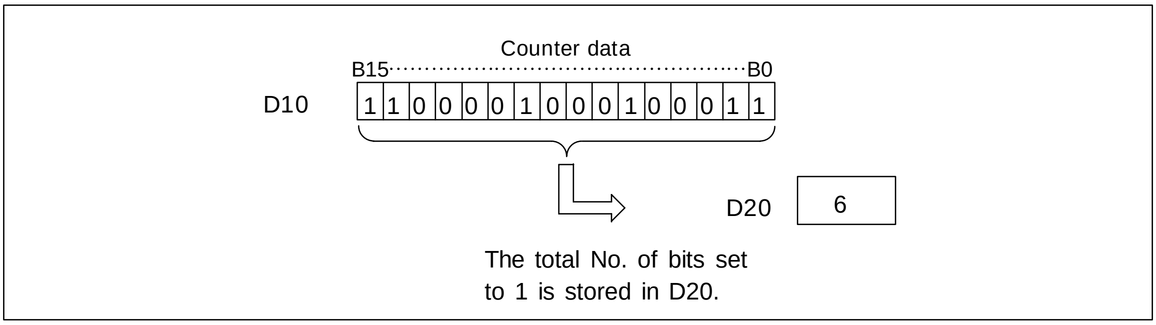
Program example

Program to obtain the No. of D10 data bits that are set to ON (1) when XB turns ON.



Coding

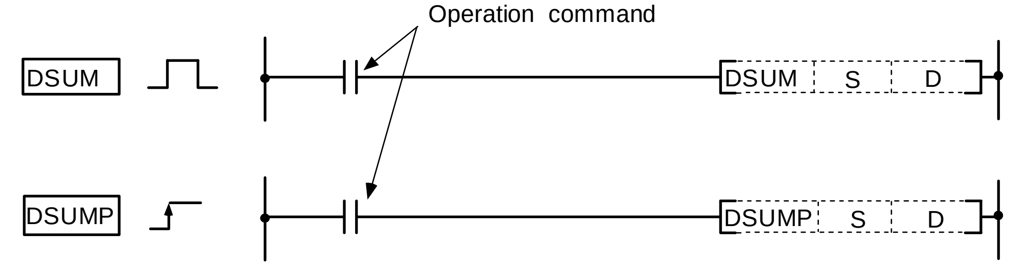
No. of steps	Com-mand	Device		
10	LD	XB		
11	SUM	D10	D20	
15				



Counting with SUM command

○ DSUM, DSUMP ... Count of No. of 32-bit data set to "1"

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S											○	○	○	○	○	○		○					○	4
D											○	○	○	○	○	○		○						

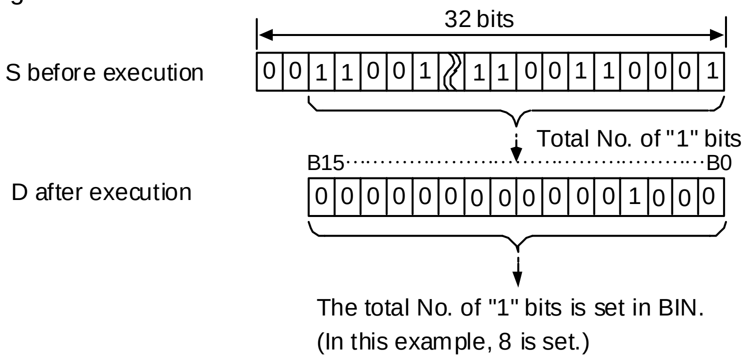


Setting data

S	Head No. of device to count the total No. of bits set to 1.
D	Head No. of device where the total No. of bits is stored.

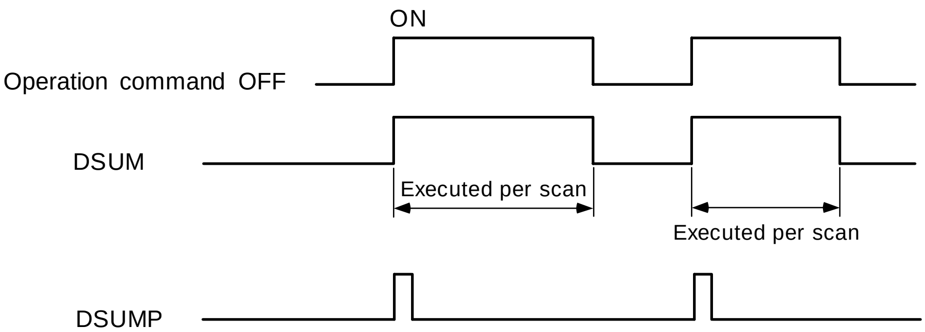
Function

The total No. of bits that are set to 1 in the 32-bit data of the device designated with S is stored in the 16-bit data designated with D.



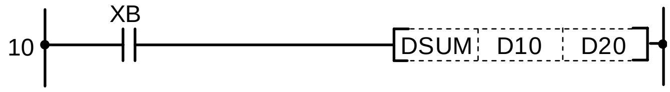
Execution conditions

The execution conditions for DSUM, DSUMP are as shown below.



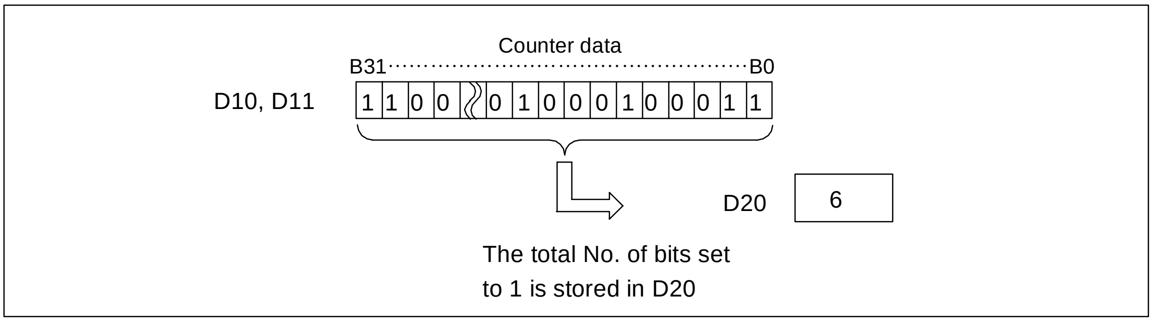
Program example

Program to obtain the No. of D10, D11 data bits that are set to ON (1) when XB turns ON.



Coding

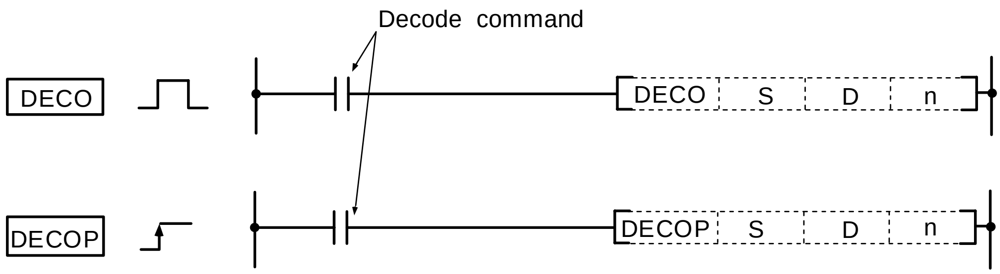
No. of steps	Com- mand	Device		
10	LD	XB		
11	DSUM	D10	D20	
15				



Counting with DSUM command

○ **DECO, DECOP ... 8 → 256 bit decoding**

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S											○	○	○	○	○	○		○					○	5
D											○	○	○	○	○	○		○						
n											○	○	○	○	○	○		○	○	○			○	



Setting data

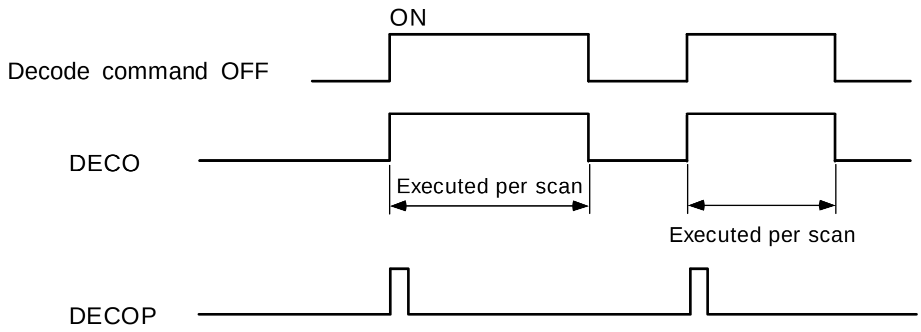
S	No. of device where data to be decoded is set.
D	Head No. of device to store the decoding results.
n	Valid bit length (1 to 8)

Function

(1) The low-order n bits of the device designated with S are decoded, and the results are stored in the 2^n bits from the device designated with D.
(2) 1 to 8 can be designated for n.
(3) No process is executed when n = 0, and the data of the device designated with D will not change.
(4) The word device is handled as 16 bits.

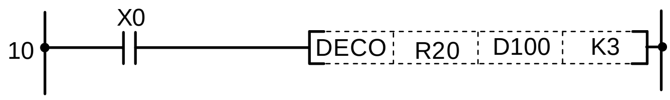
Execution conditions

The execution conditions for DECO, DECOP are as shown below.



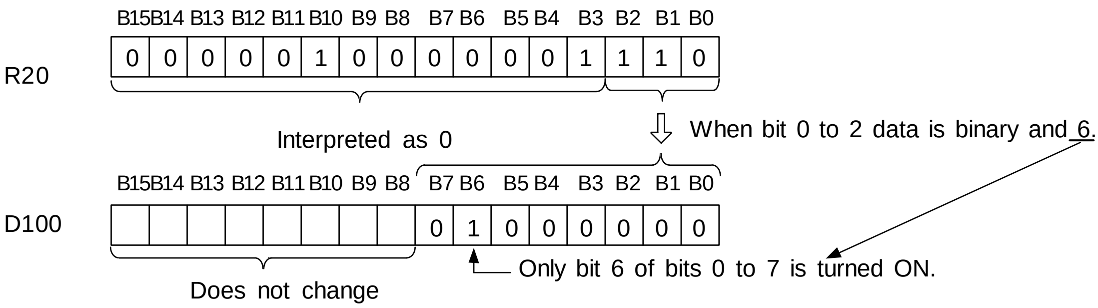
Program example

(1) Program to decode the three bits 0 to 2 of R20, and turn the bits corresponding in D100 ON.



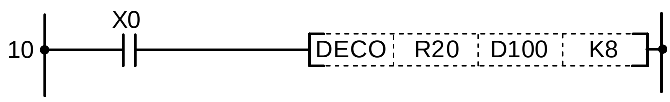
Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	DECO	R20	D100	K3
16				



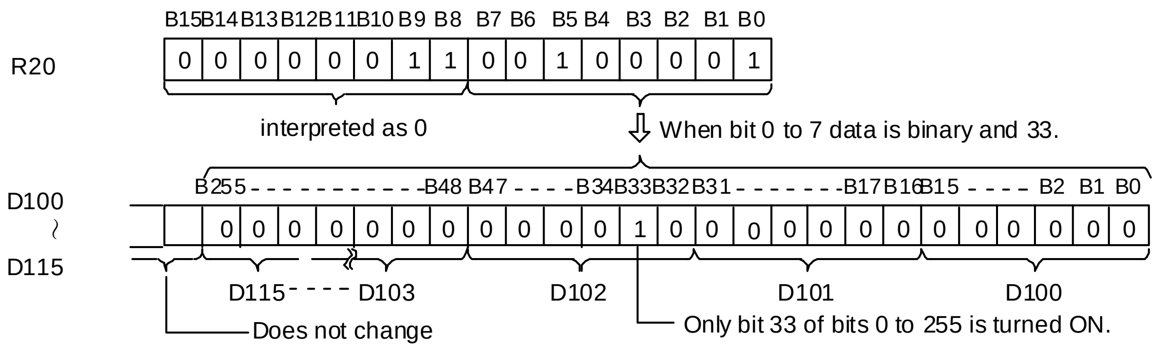
(Note 1) The D100 bit 0 turns ON when the R20 B0 to B2 is 0.
(Note 2) The D100 data remain the same even if X0 turns OFF.

(2) Program to decode the eight bits 0 to 7 of R20, and turn the bits corresponding in D100 to D115 (2^8 = 256 bits) ON.



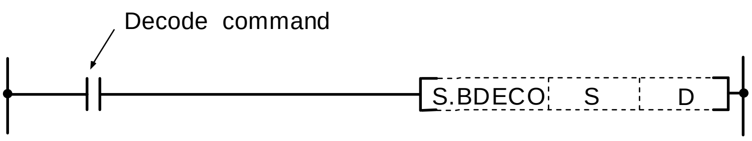
Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	DECO	R20	D100	K8
16				



○ S.BDECO ... BCD data decoding

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S											○	○	○	○	○	○		○					○	4
D											○	○	○	○	○	○		○						



Setting data

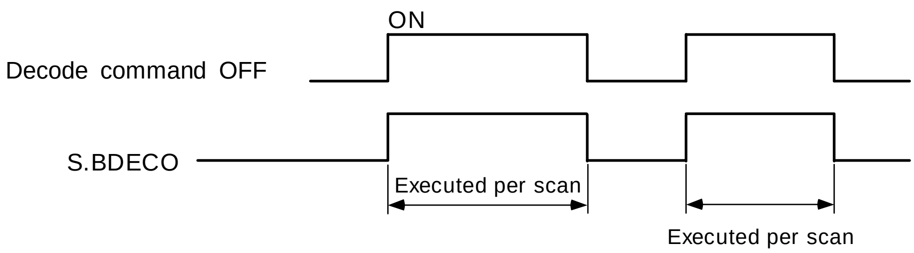
S	No. of device where data to be decoded is set.
D	Head No. of device to store the decoding results.

Function

(1) The low-order 8 bits of the device designated with S, handled as BCD data, are converted to BIN data.
The low-order 8 bits of the BIN data is decoded, and the results are stored as decoded data in 2^8 bits (=256 bits =16 words) data from the device designated with D.

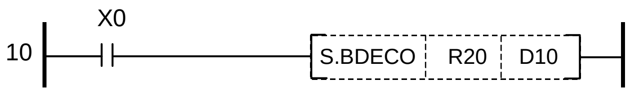
Execution conditions

The execution conditions for S.BDECO are as shown below.



Program example

(1) Program to convert the low-order 8 bits (BCD) of R20 to the BIN data, and to store the results obtained by decoding in D10 to D25 data



Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	S.BDECO	R22	D10	
15				

	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
R20	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1

Invalid

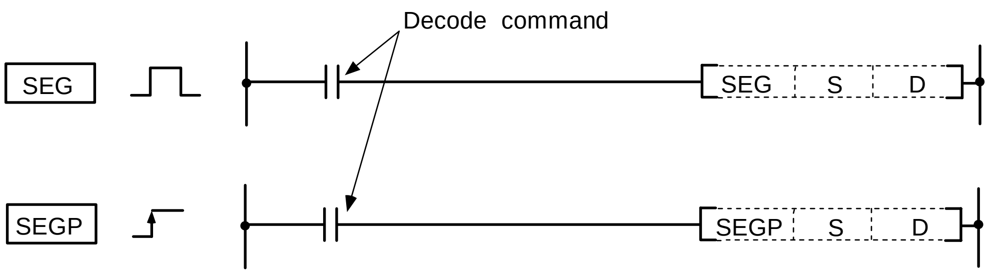
	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
D10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
D11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	B47	B46	B45	B44	B43	B42	B41	B40	B39	B38	B37	B36	B35	B34	B33	B32
D12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

	B255	B254	B253	B252	B251	B250	B249	B248	B247	B246	B245	B244	B243	B242	B241	B240
D25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(Note 1) Converting to BIN data is executed for the low-order 8 bits designated with S.
(Note 2) Decoding is executed for the BIN data after conversion.
(Note 3) D10 to D25 data of above example remains the same even if X0 turns OFF.

○ SEG, SEGP ... Decoding to 7-segment display data

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○		○	○	3
D		○	○	○	○	○	○		○		○	○	○	○	○	○	○	○						

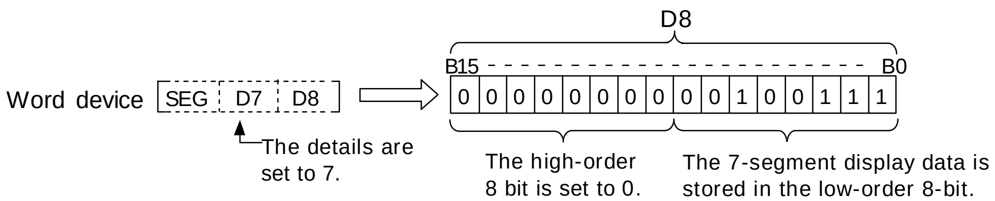


Setting data

S	Decode data or No. of device where decode data is stored.
D	No. of device to store the decoding results.

Function

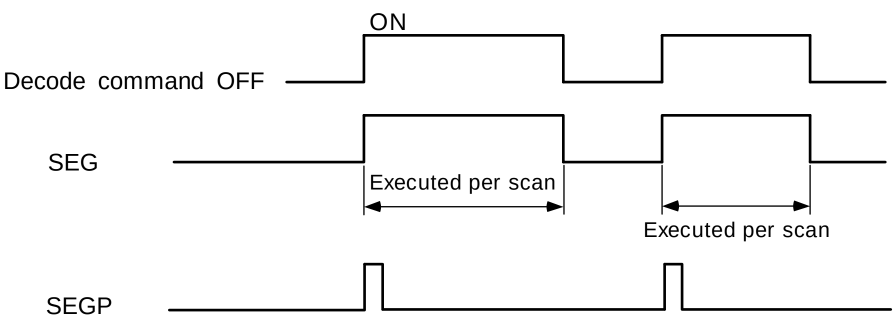
(1) The 0 to F data designated with the low-order 4-bit in S is decoded in the 7-segment display data and stored in D.



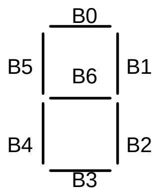
(2) Refer to the following page for the 7-segment display.

Execution conditions

The execution conditions for SEG, SEGP are as follow.



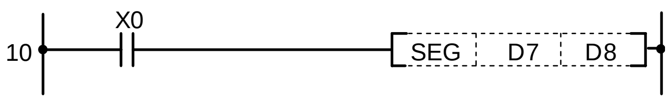
7-segment decode table

S		Configuration of 7-segment	D								Display data
Hexa-decimal	Bit pattern		B7	B6	B5	B4	B3	B2	B1	B0	
0	0000		0	0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		0	0	0	0	0	1	1	1	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		0	1	1	0	0	1	1	1	9
A	1010		0	1	1	1	0	1	1	1	A
B	1011		0	1	1	1	1	1	0	0	b
C	1100		0	0	1	1	1	0	0	1	C
D	1101		0	1	0	1	1	1	1	0	d
E	1110		0	1	1	1	1	0	0	1	E
F	1111		0	1	1	1	0	0	0	1	F

Lowest-order bit of word device

Program example

Program to convert D7 data into 7-segment display data when X0 turns ON, and output to D8.

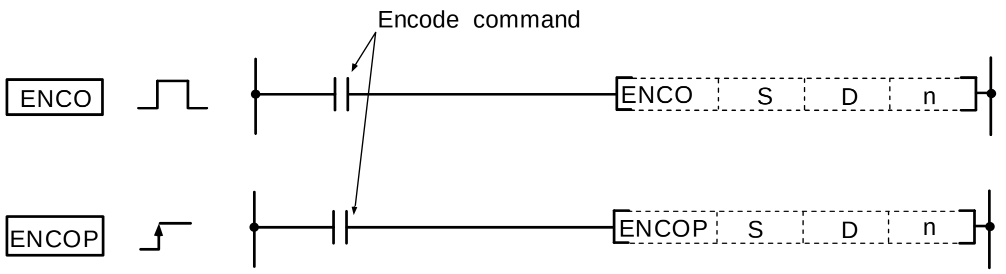


Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	SEG	D7	D8	
14				

○ ENCO, ENCOP ... 256 → 8 bit encoding

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S											○	○	○	○	○	○		○						5
D											○	○	○	○	○	○		○						
n											○	○	○	○	○	○		○	○	○			○	



Setting data

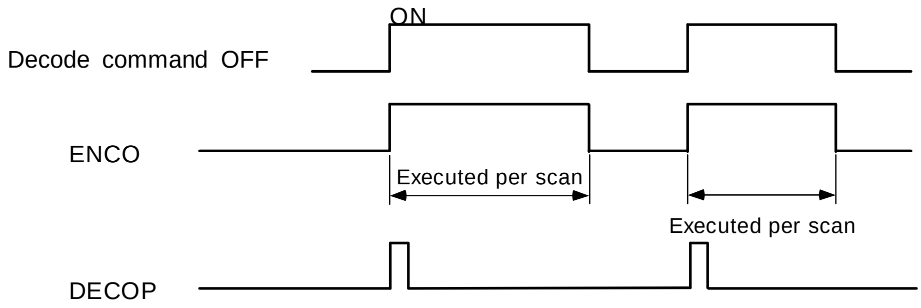
S	No. of device where data to be encoded is set.
D	Head No. of device to store the encoding results.
n	Valid bit length (1 to 8)

Function

(1) The 2^n bits of the device designated with S are encoded, and the results are stored in the low-order n bits from the device designated with D as the encoded data.
(2) 1 to 8 can be designated for n.
(3) No process is executed when n = 0, and the details of the device designated with D will not change.
(4) The word device is handled as 16 bits.
(Note 1) An error will occur if n is other than the range of 0 to 8.
(Note 2) An error will occur when 2^n bits from S exceeds the corresponding device range.
(Note 3) An error will occur when all data of 2^n bits from S is 0.

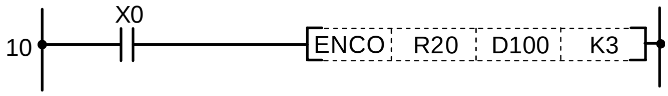
Execution conditions

The execution conditions for ENCO, ENCOP are as shown below.



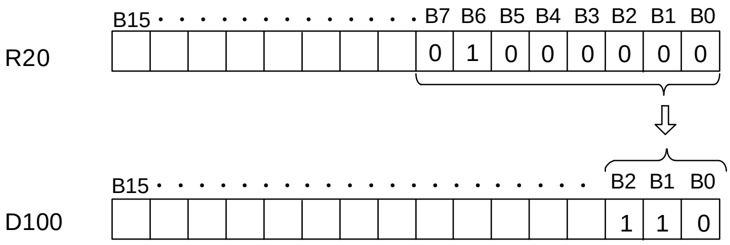
Program example

(1) Program to encode the eight bits 0 to 7 of R20, and turn the bits corresponding in D100 ON.



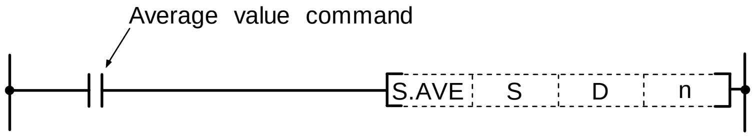
Coding

No. of steps	Com-mand	Device		
10	LD	X0		
11	ENCO	R20	D100	K3
16				



○ S.AVE ... Calculation of average value

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S											○	○	○	○	○	○		○						
D											○	○	○	○	○	○		○					○	5
n																			○	○				

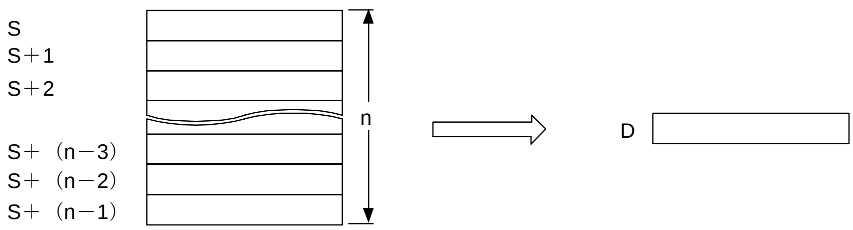


Setting data

S	Head No. of device where data to be averaged is stored.
D	Device No. of output destination
n	No. of averages

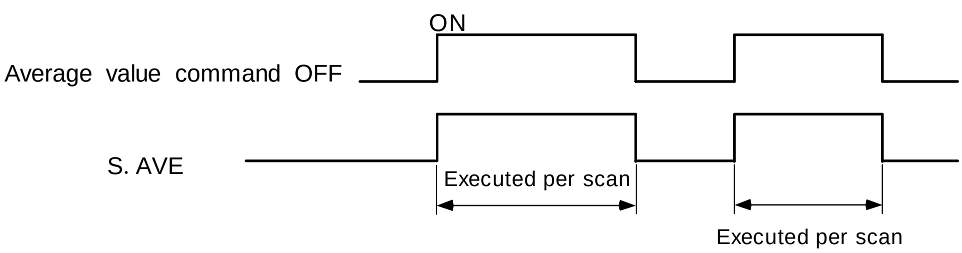
Function

The details of the n point devices from the device designated with S are averaged, and the results are output to the device designated with D.



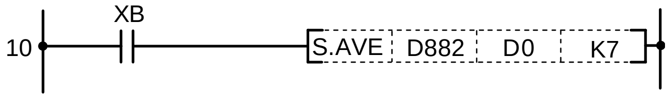
Execution conditions

The execution conditions for S.AVE are as shown below.



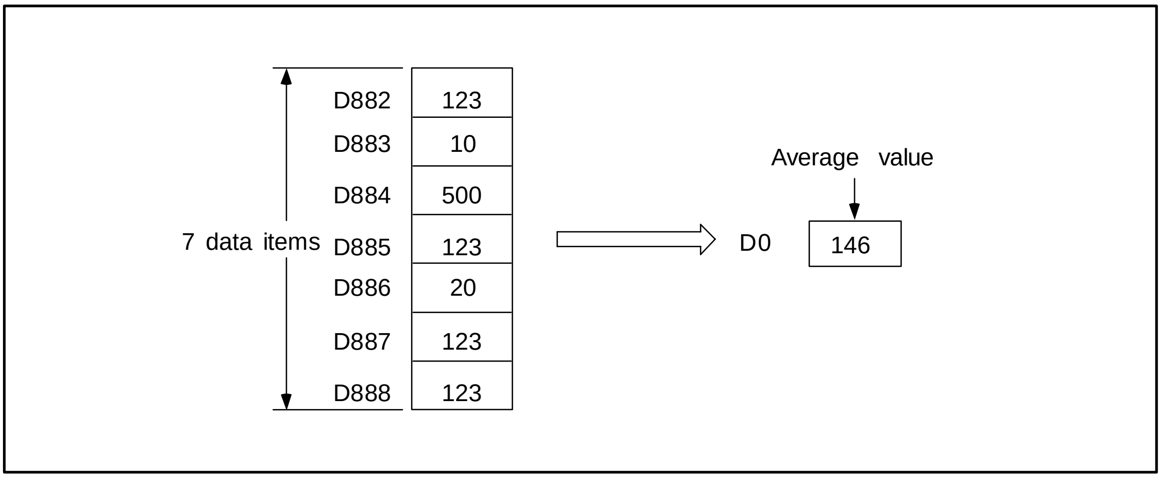
Program example

(1) Program to average the details of D882 to D888 when XB turns ON, and to output the results to D0.



Coding

No. of steps	Com-mand	Device		
10	LD	XB		
11	S.AVE	D882	D0	K7
16				

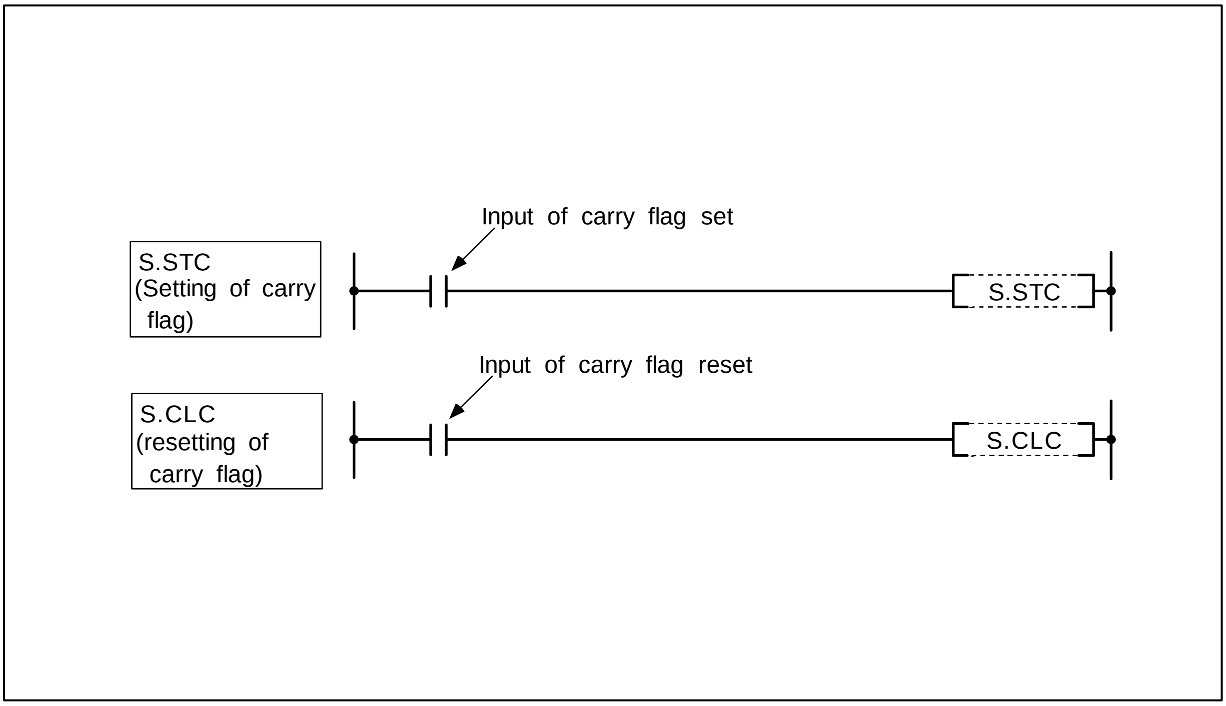


Averaging of data with S.AVE command

(Note) Fractional values are omitted.

○ S.STC, S.CLC ... Setting/resetting of carry flag

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
																								1



Function

S.STC

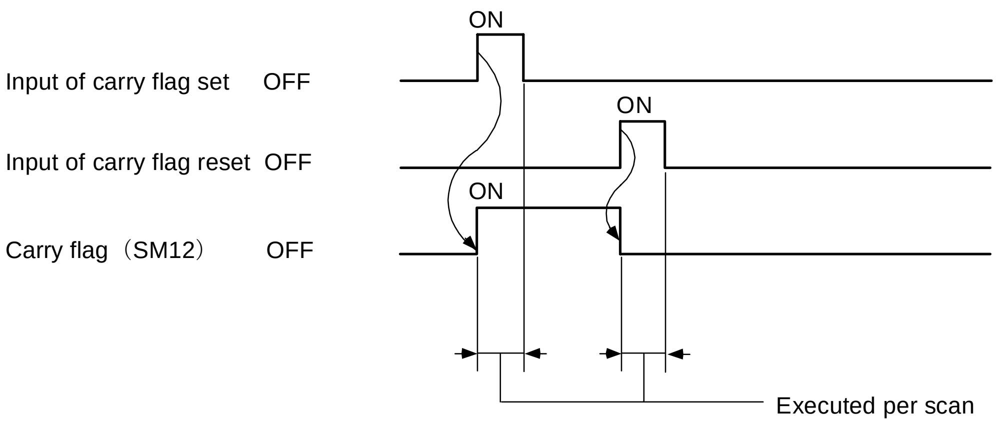
(1) The carry flag contact (SM12) is set (ON).

S.CLC

(1) The carry flag contact (SM12) is reset (OFF).

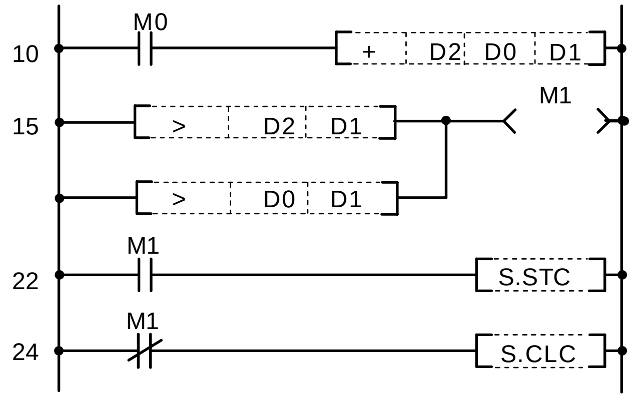
Execution conditions

The execution conditions for S.STC and S.CLC are as shown below.



Program example

Program to add the D2 data and D0 data when M0 turns ON and to turn the carry flag (SM12) ON if the results exceed 32767. If the results are 32767 or less, the carry flag is turned OFF.



Add the D2 and D0 data, and store the results in D1. (D2 + D0 -> D1)

Turn M1 ON when added data (D0) exceeds the results (D1) or the data to be added to exceeds the results

Turn carry flag ON when M1 turns ON.

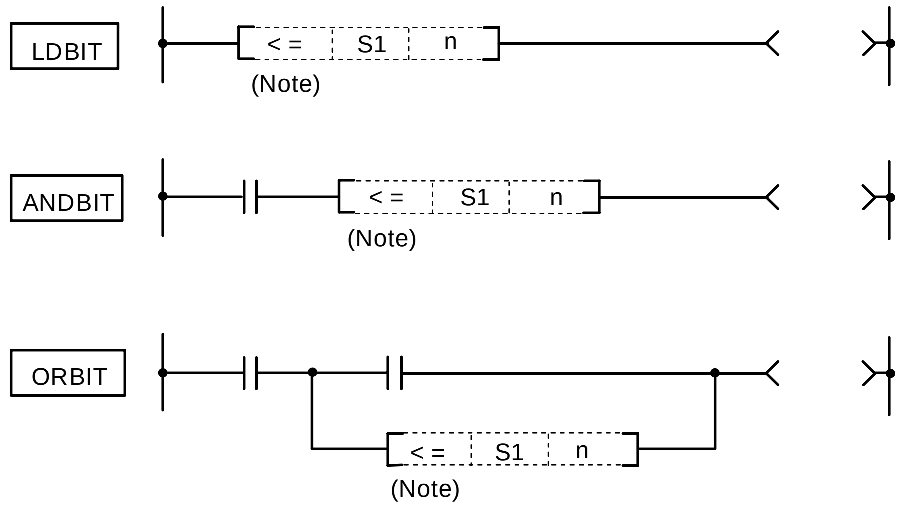
Turn carry flag OFF when M1 turns OFF.

Coding

No. of steps	Com-mand	Device		
10	LD	M0		
11	+	D2	D0	D1
15	LD>	D2	D1	
18	OR>	D0	D1	
21	OUT	M1		
22	LD	M1		
23	S.STC			
24	LDI	M1		
25	S.CLC			
26				

○ LDBIT, ANDBIT, ORBIT ... Bit test of "A" contact handling

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S											○	○	○	○	○	○	○	○					○	2
n																			○ (Note)	○ (Note)				



Setting data

S1	No. of device to execute bit test.
n	Bit to execute bit test.

(Note) In programming with the MELSEC PLC development tool (GPPQ or GX Developer), the comparison operation command mentioned above is subsutituted and used.

Function

(1) A bit test of the 16-bit device is executed with "A" contact handling.
(2) The bit test results are as shown below.

Condition	Bit test results
When test bit is 1	Continuity
When test bit is 0	Non-continuity

Execution conditions

The execution conditions for LDBIT, ANDBIT and ORBIT are as shown below.

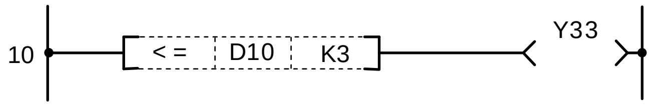
Condition	Execution conditions
LDBIT	Executed per scan
ANDBIT	Executed only when previous contact command is ON
ORBIT	Executed per scan

Cautions

Other than a constant cannot use for n. When other than a constant is used for n, it operates as LD<=, AND<=, and an OR<= command.

Program example

(1) Program to test bit 3 of D10.



Coding

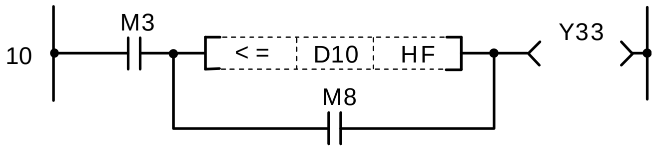
No. of steps	Com-mand	Device		
10	LD<=	D10	K3	
12	OUT	Y33		
13				

(2) Program to test bit 15 of D10.

Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND<=	D10	K15	
13	OUT	Y33		
14				

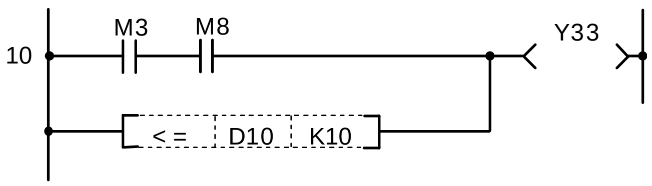
(3) Program to test bit 15 of D10.



Coding

No. of steps	Com-Mand	Device		
10	LD	M3		
11	LD<=	D10	HF	
13	OR	M8		
14	ANB			
15	OUT	Y33		
16				

(4) Program to test bit 10 of D10.

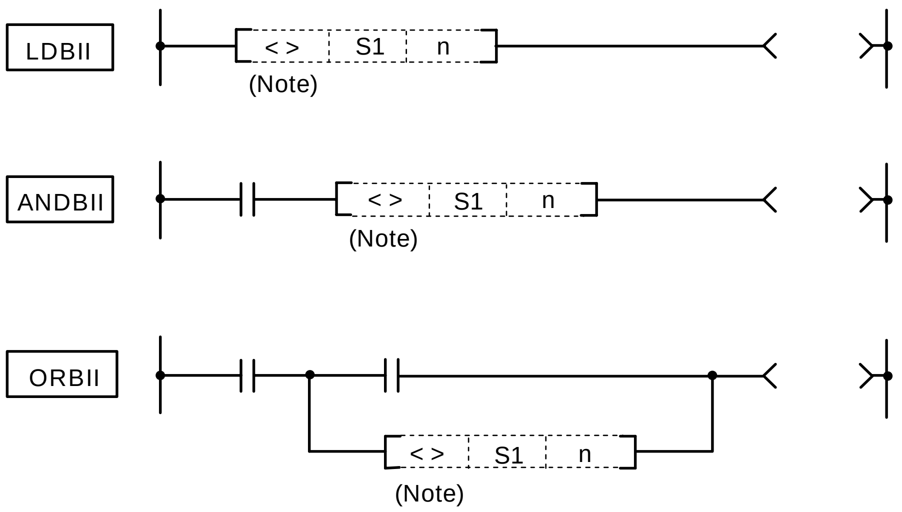


Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND	M8		
12	OR<=	D10	K10	
14	OUT	Y33		
15				

○ LDBII, ANDBII, ORBII ... Bit test of "B" contact handling

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S											○	○	○	○	○	○	○	○					○	2
n																			○ (Note)	○ (Note)				



Setting data

S1	No. of device to execute bit test.
n	Bit to execute bit test.

(Note) In programming with the MELSEC PLC development tool (GPPQ or GX Developer), the comparison operation command mentioned above is subsutituted and used.

Function

(1) A bit test of the 16-bit device is executed with "B" contact handling.
(2) The bit test results are as shown below.

Condition	Bit test results
When test bit is 0	Continuity
When test bit is 1	Non-continuity

Execution conditions

The execution conditions for LDBII, ANDBII and ORBII are as shown below.

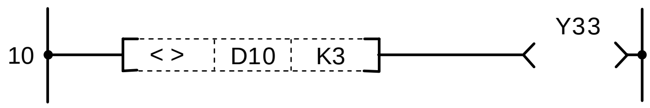
Condition	Execution conditions
LDBII	Executed per scan
ANDBII	Executed only when previous contact command is ON
ORBII	Executed per scan

Cautions

Other than a constant cannot use for n. When other than a constant is used for n, it operates as LD<>, AND<>, and an OR<> command.

Program example

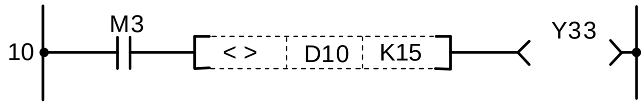
(1) Program to test bit 3 of D10.



Coding

No. of steps	Com-mand	Device		
10	LD<>	D10	K3	
12	OUT	Y33		
13				

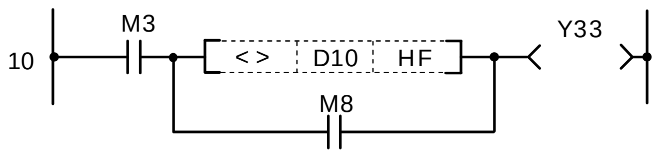
(2) Program to test bit 15 of D10.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND<>	D10	K15	
13	OUT	Y33		
14				

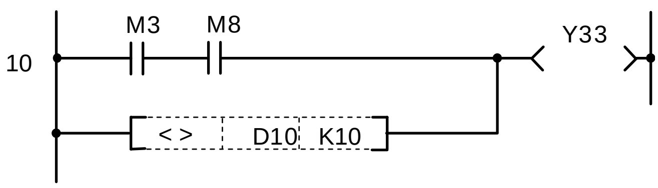
(3) Program to test bit 15 of D10.



Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	LD<>	D10	HF	
13	OR	M8		
14	ANB			
15	OUT	Y33		
16				

(4) Program to test bit 10 of D10.

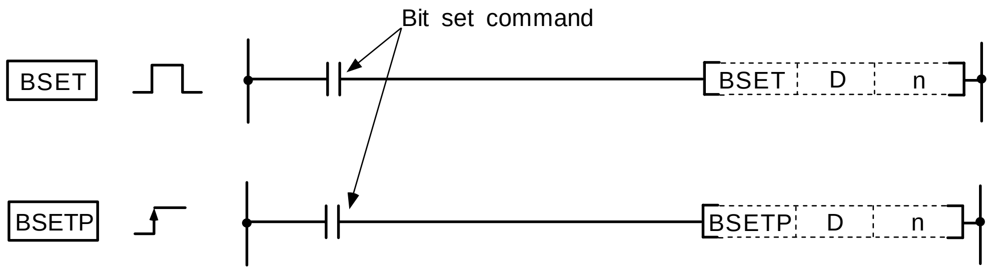


Coding

No. of steps	Com-mand	Device		
10	LD	M3		
11	AND	M8		
12	OR<>	D10	K10	
14	OUT	Y33		
15				

○ BSET, BSETP ... Bit setting of word device

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D											○	○	○	○	○	○	○	○					○	2
n																			○	○				

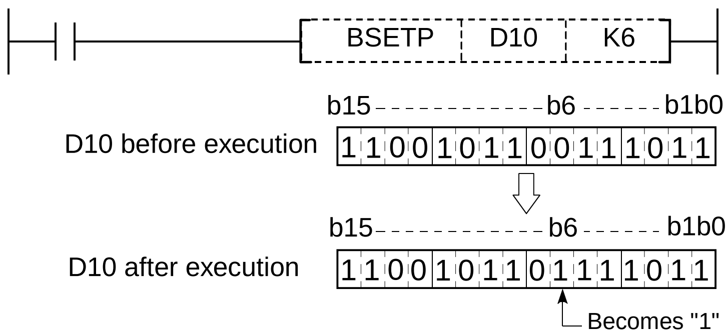


Setting data

D	No. of device to set the bit.
n	Bit No. to be set.

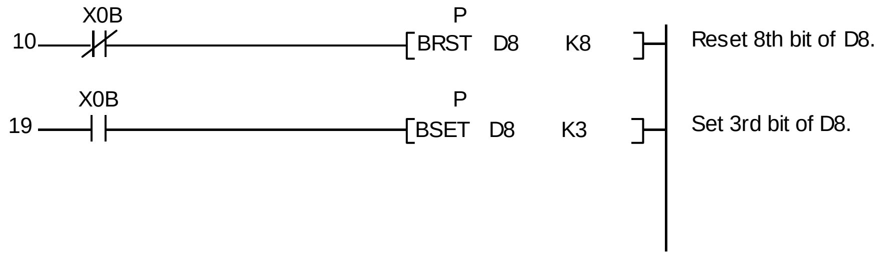
Function

(1) The bit n of the word device designated with D is set to 1.
(2) The value 0 to 15 can be designated for n.
If n exceeds 15, the command is executed is executed with the low-order 4 bits data.



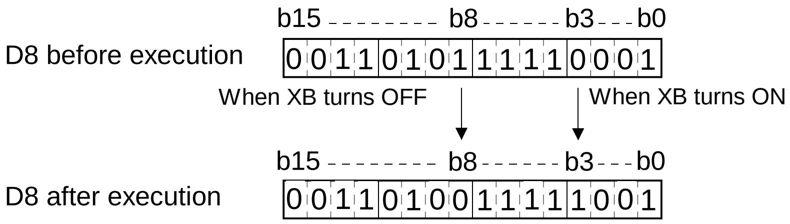
Program example

(1) Program to reset 8th bit (to 0) of D8 when XB turns OFF, and to set 3rd bit (to 1) of D8 when XB turns ON.



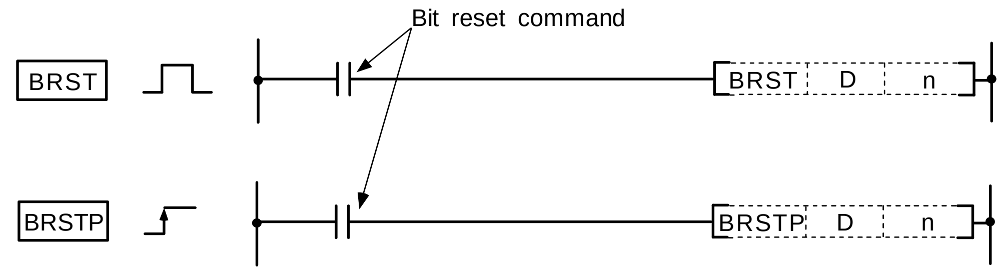
Coding

No. of steps	Com-mand	Device		
10	LDI	X0B		
11	BRSTP	D8	K8	
19	LD	X0B		
20	BSETP	D8	K3	
28				



○ BRST, BRSTP ... Bit resetting of word device

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
D											○	○	○	○	○	○	○	○					○	2
n																			○	○				

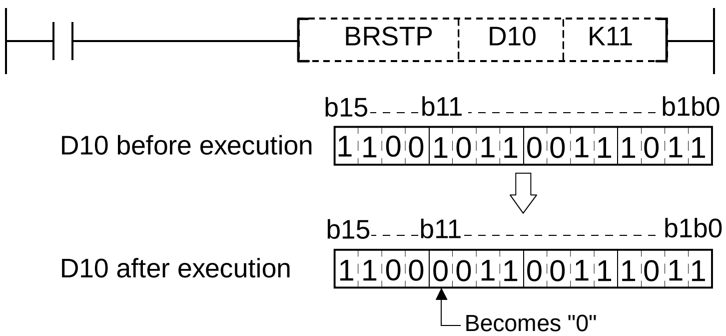


Setting data

D	No. of device to reset the bit.
n	Bit No. to be reset.

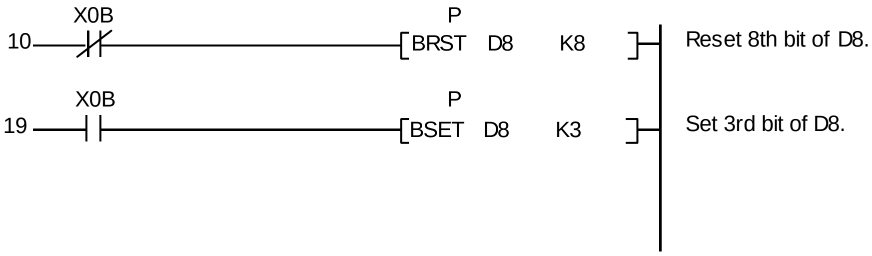
Function

(1) The bit n of the word device designated with D is reset to 0.
(2) The value 0 to 15 can be designated for n.
If n exceeds 15, the command is executed is executed with the low-order 4 bits data.



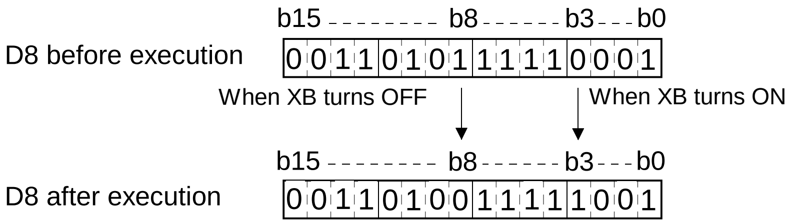
Program example

(1) Program to reset 8th bit (to 0) of D8 when XB turns OFF, and to set 3rd bit (to 1) of D8 when XB turns ON.



Coding

No. of steps	Com- mand	Device		
10	LDI	X0B		
11	BRSTP	D8	K8	
19	LD	X0B		
20	BSETP	D8	K3	
28				



9. Exclusive Commands 1

Although the basic and functional commands are not used only for specific purposes, some commands may be efficient if command applications such as data transfer between under PLC and controller and controller display screen are limited.
Then, we provide a number of exclusive commands which are explained below.

Examples of exclusive commands:

· Intelligent function command (FROM, TO)
· Transient command (OPEN, CLOSE, BUFSND, BUFRCV)
· Transient transmission command
(READ, SREAD, WRITE, SWRITE, RIRD, RIWT)

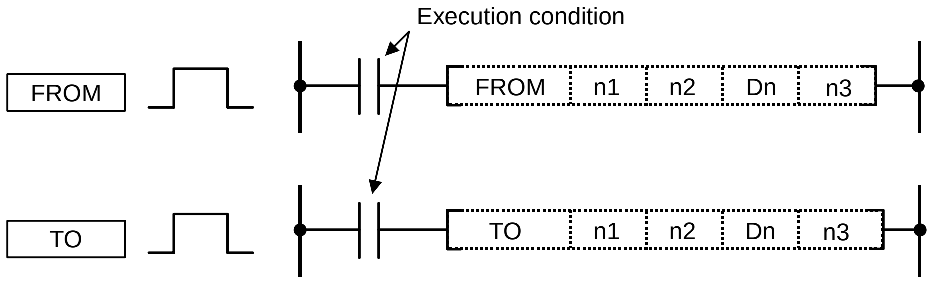
Refer to "9. Exclusive Commands 1".

· ATC dedicated command (ATC)
· Rotary body control command (ROT)
· Tool life management exclusive command (TSRH)
· DDB (direct data bus) ..... asynchronous
· External search .... synchronous

Refer to "10. Exclusive Commands 2".

○ FROM, TO ... Reading from buffer memory / Writing to buffer memory

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
n1													○	○	○	○		○	○	○				5
n2													○	○	○	○		○	○	○				
Dn	○	○	○	○	○	○	○		○				○	○	○	○		○						
n3													○	○	○	○		○	○	○				



Setting data (FROM)

n1	Head I/O No. (Designate the first two digits of the 3-digit I/O No. allocated to the intelligent function unit.)
n2	Head address of the buffer memory in which the data to read is stored.
Dn	Head No. of the PLC register of C6/C64 which will store the read data.
n3	The number of words of the data to read.

Setting data (TO)

n1	Head I/O No. (Designate the first two digits of the 3-digit I/O No. allocated to the intelligent function unit.)
n2	Head address of the buffer memory in which the data to write is stored.
Dn	Head No. of the PLC register of C6/C64 which will store the write data.
n3	The number of words of the data to write.

(Note) As for bit device, the only available unit for designation is 16 bits.

Function

FROM

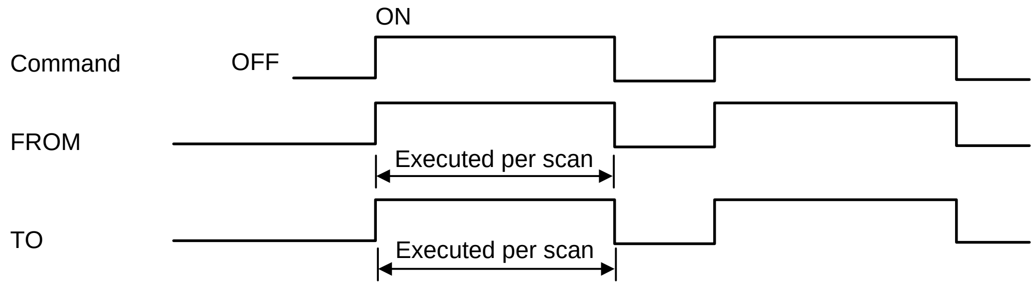
Reads the data of the buffer memory of MELSEC intelligent function unit out to the C6/C64 PLC.

TO

Writes the data of C6/C64 PLC register into the buffer memory of MELSEC intelligent function unit.

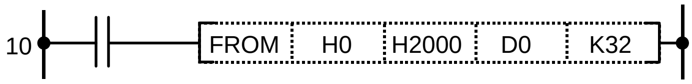
Execution conditions

The execution conditions for FROM, TO are as shown below.



Program example

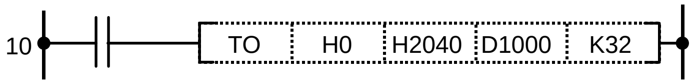
(1) To read 32 words of data in Area 2 of the FL-net unit (buffer memory address 2000H) and write them into D0 to D31 of data register. (When the I/O number of FL-net unit is 000.)



Coding

No. of steps	Com-mand	Device			
10	LD				
11	FROM	H0	H2000	D0	K32
16					

(2) To write 32 words of data in data register D1000 to D1031 into the buffer memory addresses 2040H to 205FH of FL-net unit's Area 2. (When the I/O number of FL-net unit is 000.)



Coding

No. of steps	Com-mand	Device			
10	LD				
11	TO	H0	H2040	D1000	K32
16					

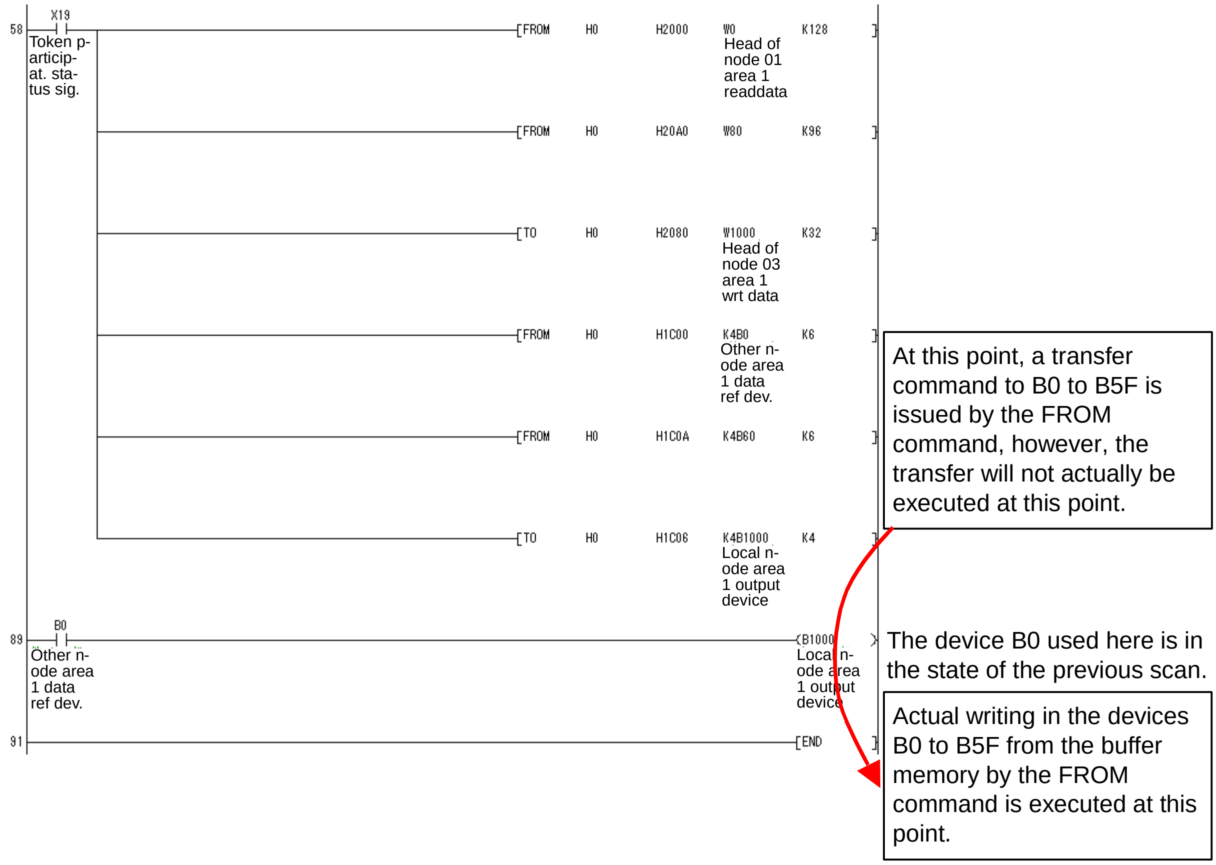
Restrictions in Using FROM/TO Command

Using FROM/TO command by the built-in PLC in C6/C64 has the restrictions below.

(1) The number of FROM and TO commands that can be used in one scan (including multiple program) is 50 each. Using more than 50 will cause the alarm "Q01 EMERGENCY STOP LAD 0005", and will stop the built-in PLC.

(2) The accessible size of buffer memory in one scan (including multiple program) by FROM/TO command is up to 12k words. Exceeding 12k words will cause the alarm "Q01 EMERGENCY STOP LAD 0006", and will stop the built-in PLC.

(3) Actual transfer from the buffer memory to read devices by FROM command, and transfer from write devices to the buffer memory by TO command are done at the execution of END command. (In multiple programs, at the execution of END command at the end of all the programs.) Thus, device data renewal will be delayed by 1 scan.



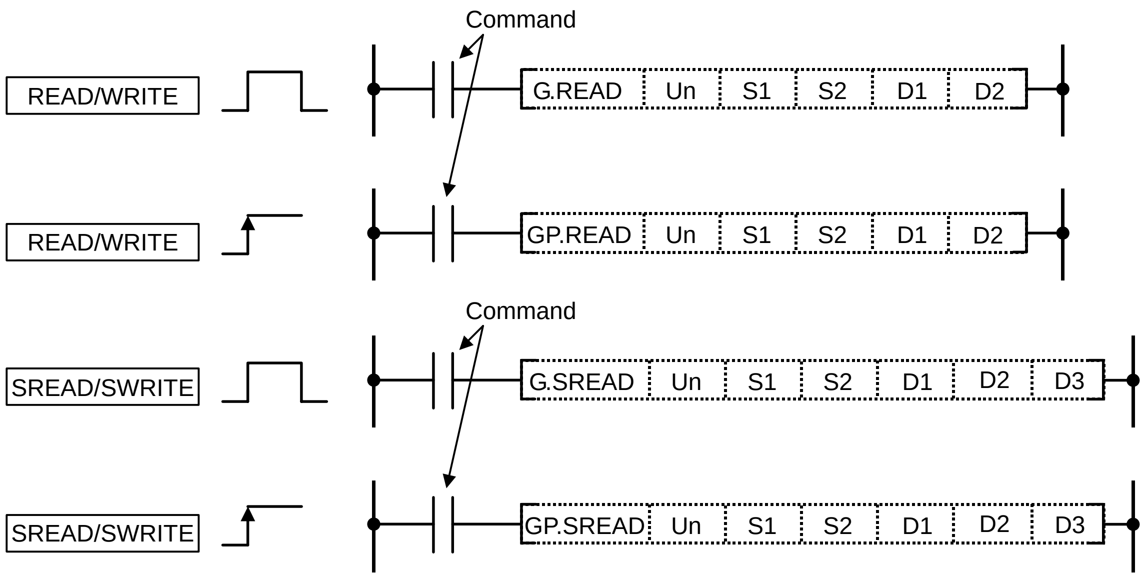
(4) FROM/TO commands cannot be used in high-speed program processing.
If FROM/TO command is executed in high-speed program processing, the alarm "Q01 EMERGENCY STOP LAD 0007" will occur and the built-in PLC will stop.

(5) If bit device is used for FROM/TO command, the only available unit for designation is 16 bits.
If bit device is not designated in the unit of 16 bits, the alarm "Q01 EMERGENCY STOP LAD 0008" will occur and the built-in PLC will stop.

Correct example :	FROM	H0 H1C00 K4B20 K32	
	TO	H0 H1C80 K4M64 K32	
Incorrect example :	FROM	H0 H1C00 K4B28 K32	→ "Q01 EMERGENCY STOP LAD 0008"
	TO	H0 H1C80 K4M100 K32	→ "Q01 EMERGENCY STOP LAD 0008"

○ READ, SREAD, WRITE, SWRITE ... Transient command

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Con-stant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1											○	○	○	○	○	○		○						6/12
S2											○	○	○	○	○	○		○						
D1											○	○	○	○	○	○		○						
D2	○	○	○	○	○	○	○		○															
S1											○	○	○	○	○	○		○						7/13
S2											○	○	○	○	○	○		○						
D1											○	○	○	○	○	○		○						
D2	○	○	○	○	○	○	○		○															
D3	○	○	○	○	○	○	○		○															



(Note1) The stations targeted for this command are the QnCPU, QnACPU or MELDAS master/local stations.

Setting data (READ/SREAD)

Un **(Note2)**	Local station head input/output No.
S1	Head device of local station storing the control data
S2	Head device of target station storing the data to be read
D1	Head device of local station where the read data is to be stored
D2	Local station device that turns 1 scan ON at completion of command
D3	Target station device that turns 1 scan ON at completion of command

Setting data (WRITE/SWRITE)

Un **(Note2)**	Local station head input/output No.
S1	Head device of local station storing the control data
S2	Head device of local station storing the data to be written
D1	Head device of target station where the written data is to be stored
D2	Local station device that turns 1 scan ON at completion of command
D3	Target station device that turns 1 scan ON at completion of command

(Note2) Designate the Un value as U20 for the EXT1 card, as U28 for the EXT2 card, and as U30 for the EXT3 card.

Function

READ/SREAD

Reads the word device data of the designated station out to the local station.
SREAD turns the target station device ON at read completion, that enables the target station to confirm the data has been read out.

(1) Control data

[Control data composition (S1)]

Device	Item	Data set by User (at execution)*1	Data set by System (at completion)*2	
(S1)	Execution/Error complete type	○		
(S1)+ 1	Completion status		○	
(S1)+ 2	Channel used by the local station	○		
(S1)+ 3	(Not used)	—	—	
(S1)+ 4	Target station network No.	○		
(S1)+ 5	Target station No.	○		
(S1)+ 6	(Not used)			
(S1)+ 7	Number of resends	○	○	
(S1)+ 8	Arrival monitoring time	○		
(S1)+ 9	Read data length	○		
(S1)+10	(Not used)	—	—	
(S1)+11	Clock set flag (Set only when an error is detected)		○	Used when "Error complete type" is set as "Clock data setting required".
(S1)+12	Year/Month of error occurrence		○	
(S1)+13	Date/Hour of error occurrence		○	
(S1)+14	Minute/Second of error occurrence		○	
(S1)+15	Day of the week of error occurrence		○	
(S1)+16	Error detection network No.		○	
(S1)+17	Error detection station No.		○	

*1:Items set in a sequence program.
*2:Items to be stored automatically at command completion.

Control data

Device	Item	Set data	Setting range	Set by
(S1)+0	Execution type	With arrival confirmation (bit0=1 fixed)	0001H 0081H	User
	Error complete type	Error complete type (bit 7) Set whether or not the clock data setting is required at error completion. 0:Clock data setting is not required Clock data is not stored in (S1)+11 to (S1)+17 when errors occur. 1:Clock data setting is required Clock data is stored in (S1)+11 to (S1)+17 when errors occur.		

Device	Item	Set data	Setting range	Set by
(S1)+1	Completion status	Store the status at completion of a command. 0 : Normal Other than 0 : Error (error code)	—	System
(S1)+2	Channel used by the local station	Designate the channel used by the local station.	1 to 8	User
(S1)+3	(Not used)	—	—	—
(S1)+4	Target station network No.	Designate the network No. of the target station. 1 to 239:Network No. 254:When 254 is designated in "Un"	1 to 239, 254	User
(S1)+5	Target station No.	Designate the target station. 1 to 64:The station with the corresponding station No.	1 to 64	User
(S1)+6	(Not used)	(Fixed value)	0	User
(S1)+7	Number of resends	(1) At command execution Set the number of resends when the command is not completed within the monitoring time designated by (S1)+8. 0 to 15 (times)	0 to 15	User
		(2) At command completion Store the number of resends executed (result). 0 to 15 (times)		System
(S1)+8	Arrival monitoring time	When the command fails to complete within the monitoring time, it is resent for the number of resends designated in (S1)+7.	1 to 32767 0: 10 seconds	User
(S1)+9	Read data length	Designate the number of data to be read. 1 to 480 (Words)	1 to 480	User
(S1)+10	(Not used)	—	—	—
(S1)+11	Clock set flag (Set only when an error is detected)	Store the valid/invalid status of the data in ((S1)+12 to (S1)+17). 0:Invalid 1:Valid	—	System
(S1)+12	Year/Month of error occurrence	Higher 8 bits : month (01H to 12H), lower 8 bits : last 2 digits of year (00H to 99H) b15 to b8 | b7 to b0 Month (01H to 12H) | Year (00H to 99H)	—	System
(S1)+13	Date/Hour of error occurrence	Higher 8 bits : hour (00H to 23H), lower 8 bits : date (01H to 31H) b15 to b8 | b7 to b0 Hour (00H to 23H) | Date (01H to 31H)	—	System
(S1)+14	Minute/Second of error occurrence	Higher 8 bits : second (00H to 59H), lower 8 bits : minute (00H to 59H) b15 to b8 | b7 to b0 Second (00H to 59H) | Minute (00H to 59H)	—	System
(S1)+15	Day of the week of error occurrence	Higher 8 bits : 00H, lower 8 bits : day of the week (00H(Sunday) to 06H(Saturday)) b15 to b8 | b7 to b0 00H | Day of week (00H to 06H)	—	System

Device	Item	Set data	Setting range	Set by
(S1)+16	Error detection network No.	Stores the network number of the station that detected an error. However, it is not stored when the completion status of (S1)+1 is "Channel in use (C085H)." 1 to 239 (Network No.)	1 to 239	System
(S1)+17	Error detection station No.	Stores the station number of the station that detected an error. However, it is not stored when the completion status of (S1)+1 is "Channel in use (C085H)." 1 to 64 (Station No.)	1 to 64	System

WRITE/SWRITE

Writes the data of the local station to the word device data area of the designated station.
SWRITE turns the target station device ON at write completion, that enables the target station to confirm the data has been written.

(1) Control data

[Control data composition (S1)]

Refer to the next page for details.

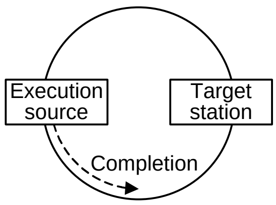
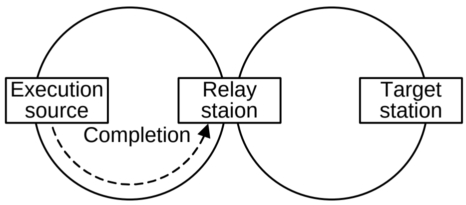
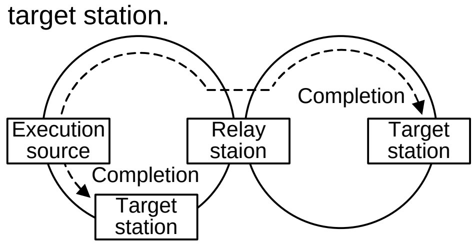
Device	Item	Data set by User (at execution)*1	Data set by System (at completion)*2
(S1)	Execution/Error complete type	○	
(S1)+ 1	Completion status		○
(S1)+ 2	Channel used by the local station	○	
(S1)+ 3	(Not used)	—	—
(S1)+ 4	Target station network No.	○	
(S1)+ 5	Target station No.	○	
(S1)+ 6	(Not used)		
(S1)+ 7	Number of resends	○	○
(S1)+ 8	Arrival monitoring time	○	
(S1)+ 9	Write data length	○	
(S1)+10	(Not used)	—	—
(S1)+11	Clock set flag (Set only when an error is detected)		○
(S1)+12	Year/Month of error occurrence		○
(S1)+13	Date/Hour of error occurrence		○
(S1)+14	Minute/Second of error occurrence		○
(S1)+15	Day of the week of error occurrence		○
(S1)+16	Error detection network No.		○
(S1)+17	Error detection station No.		○

Used when "Error complete type" is set as "Clock data setting required".

*1:Items set in a sequence program.

*2:Items to be stored automatically at command completion.

Control data

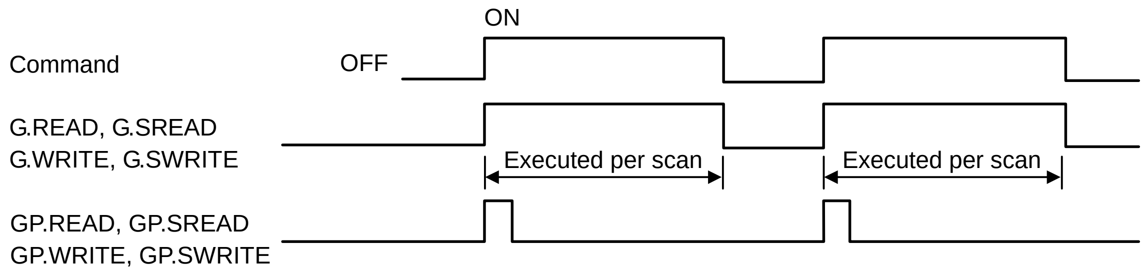
Device	Item	Set data	Setting range	Set by
(S1)+0	Execution type	b15 ~ b7 ~ b0 | 0 | (2) | 0 | (1) | (1) Execution type (bit 0) 0:No arrival confirmation When the target station is on the same network: Completed when the data is sent from the local station.  When the target station is on the other network: Completed when the data arrives at the relay station on the network of the local station.  1:With arrival confirmation Completed when data is written to the target station.  	0001H 0081H	User
	Error complete type	(2) Error complete type (bit 7) Designate whether the clock data setting is required or not at error completion. 0:Clock data setting not required: Clock data at error occurrence is not stored in (S)+11 to (S)+17. 1:Clock data setting required: Clock data at error occurrence is stored in (S)+11 to (S)+17.		
(S1)+1	Completion status	Store the status at completion of a command. 0:Normal Other than 0:Error (error code)	—	System
(S1)+2	Channel used by the local station	Set the channel used by the local station.	1 to 8	User
(S1)+3	(Not used)	—	—	—

Device	Item	Set data	Setting range	Set by
(S1)+4	Target station network No.	Designate the network No. of the target station. 1 to 239:Network No. 254:When 254 is designated in "Un"	1 to 239, 254	User
(S1)+5	Target station No.	Designate the target station. 1 to 64 : The station with the corresponding station number 81H to 89H : All the station with the indicted group number (Can be set when the execution type designated in (S1) is "0:No arrival confirmation".) FFH : All stations having the target network number (simultaneous broadcast): Excluding the local station (Can be set when the execution type designated in (S1) is "0:No arrival confirmation".)	1 to 64 81H to 89H FFH	User
(S1)+6	(Not used)	(Fixed value)	—	—
(S1)+7	Number of resends	Valid when the execution type designated in (S1) is "1:With arrival confirmation." (1) At command execution Set the number of resends when the command fails to complete within the monitoring time designated by (S1)+8. 0 to 15 (times)	0 to 15	User
		(2) At command completion Store the number of resends executed (result). 0 to 15 (times)		System
(S1)+8	Arrival monitoring time	Valid when the execution type designated in (S1) is "1:With arrival confirmation." Set the monitoring time until the command completion. When the command fails to complete within the monitoring time, it is resent for the number of resends designated in (S1)+7.	1 to 32767 0: 10 seconds	User
(S1)+9	Write data length	Designate the number of write data in (S2) to (S2)+n. 1 to 480 (Words)	1 to 480	User
(S1)+10	(Not used)	—	—	—
(S1)+11	Clock set flag (Set only when an error is detected)	Store the valid/invalid status of the data in ((S1)+12 to (S1)+17). 0:Invalid 1:Valid	—	System
(S1)+12	Year/Month of error occurrence	Higher 8 bits : month (01H to 12H), lower 8 bits : last 2 digits of year (00H to 99H) b15 to b8 | b7 to b0 Month (01H to 12H) | Year (00H to 99H)	—	System

Device	Item	Set data	Setting range	Set by
(S1)+13	Date/Hour of error occurrence	Higher 8 bits : hour (00H to 23H), lower 8 bits : date (01H to 31H) b15 to b8 | b7 to b0 Hour (00H to 23H) | Date (01H to 31H)	—	System
(S1)+14	Minute/Second of error occurrence	Higher 8 bits : second (00H to 59H), lower 8 bits : minute (00H to 59H) b15 to b8 | b7 to b0 Second (00H to 59H) | Minute (00H to 59H)	—	System
(S1)+15	Day of the week of error occurrence	Higher 8 bits : 00H, lower 8 bits : day of the week (00H(Sunday) to 06H(Saturday)) b15 to b8 | b7 to b0 00H | Day of week (00H to 06H)	—	System
(S1)+16	Error detection network No.	Stores the network number of the station that detected an error. However, it is not stored when the completion status of (S1)+1 is "Channel in use (C085H)." 1 to 239 (Network No.)	1 to 239	System
(S1)+17	Error detection station No.	Stores the station number of the station that detected an error. However, it is not stored when the completion status of (S1)+1 is "Channel in use (C085H)." 1 to 64 (Station No.)	1 to 64	System

Execution conditions

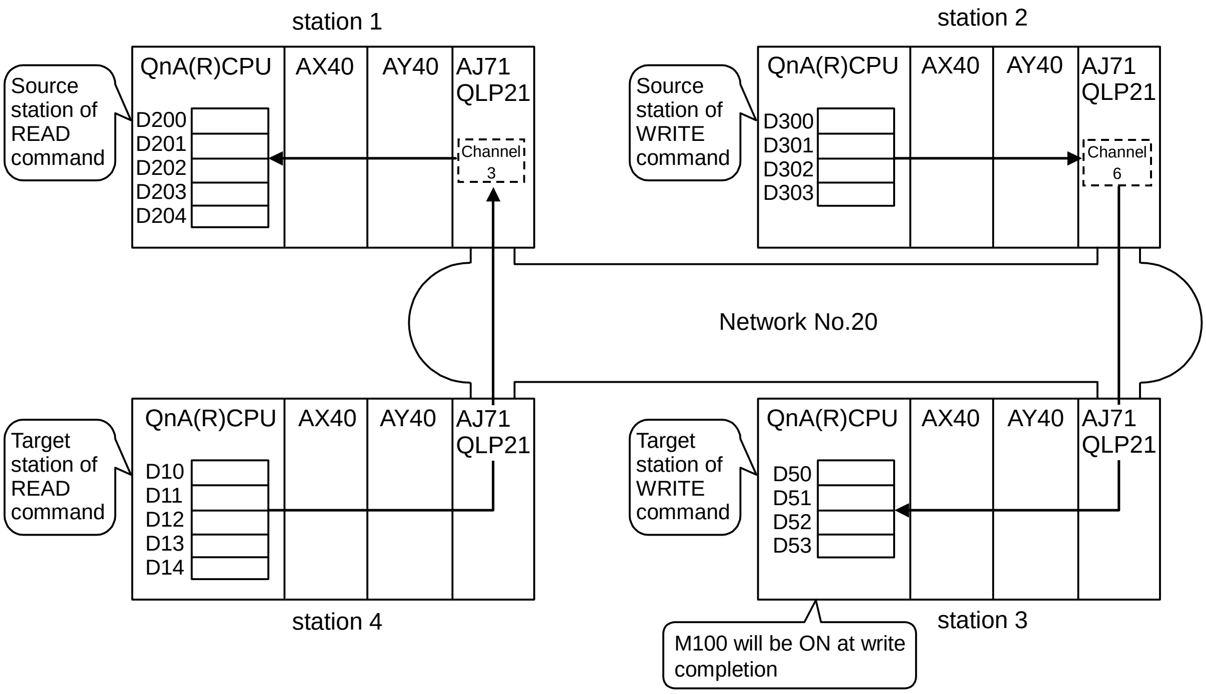
The execution conditions for READ, SREAD, WRITE, SWRITE are as shown below.



Program example

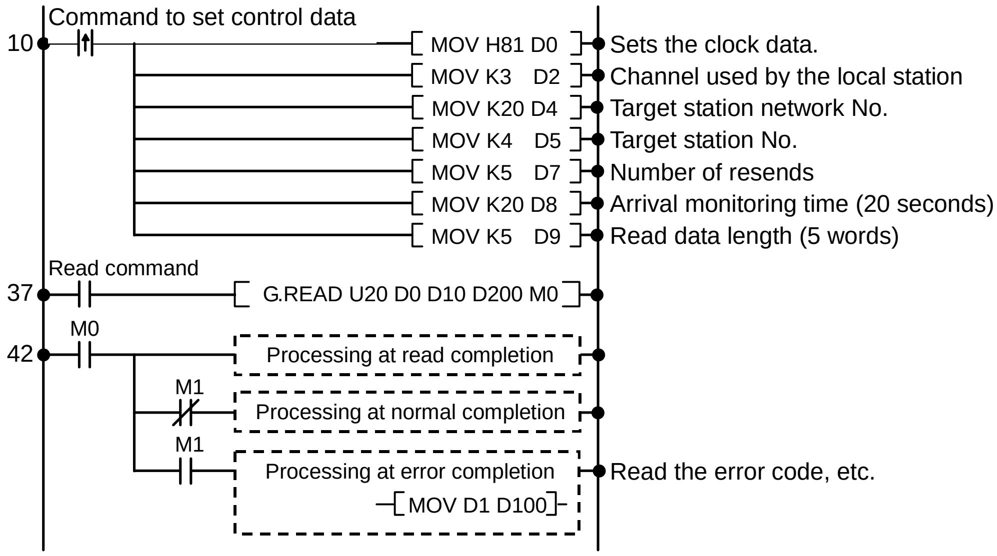
Reading the data stored in D10 to D14 of the station 4 into D200 to D204 of the station 1.
(→Refer to (a).)

Writing the data stored in D300 to D304 of the station 2 into D50 to D53 of the station 3.
(→Refer to (b).)



(a) Program of Ch.1 (READ command)

To execute the program as below, apply the interlock.

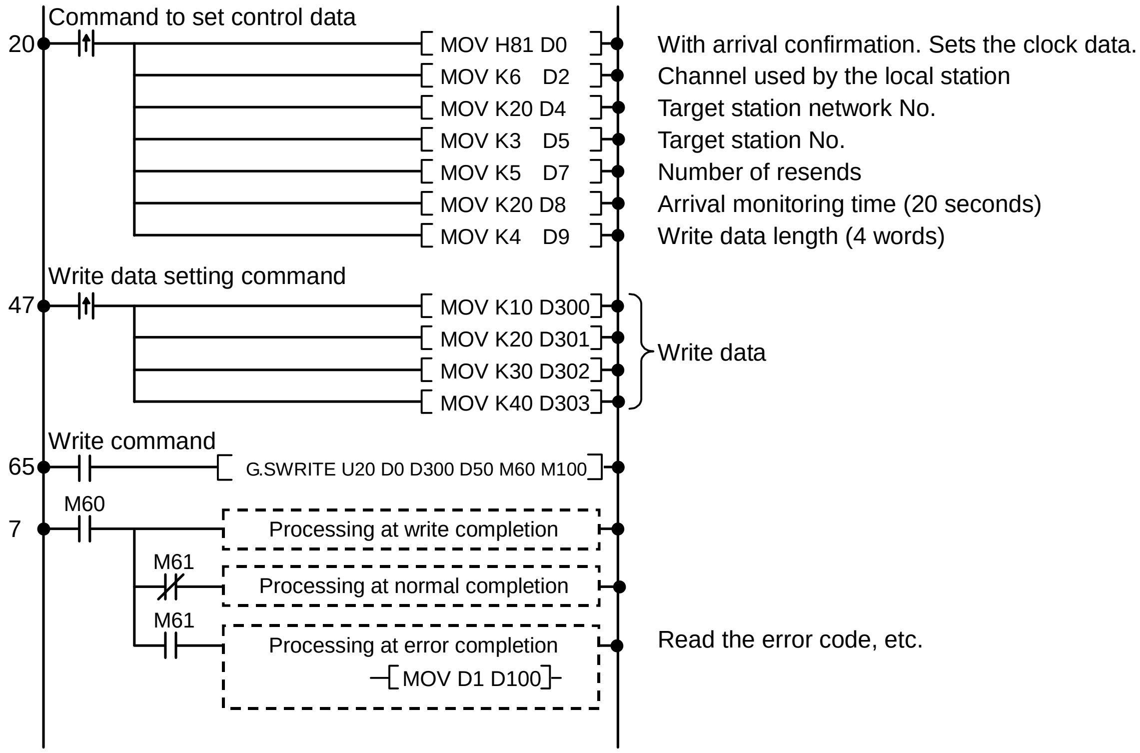


Coding

No. of steps	Com-mand	Device				
10	LDP					
16	MOV	H81	D0			
19	MOV	K3	D2			
22	MOV	K20	D4			
25	MOV	K4	D5			
28	MOV	K5	D7			
31	MOV	K20	D8			
34	MOV	K5	D9			
37	LD					
38	G.READ	U20	D0	D10	D200	M0
44						

(b) Program of Ch.2 (WRITE command)

To execute the program as below, apply the interlock.

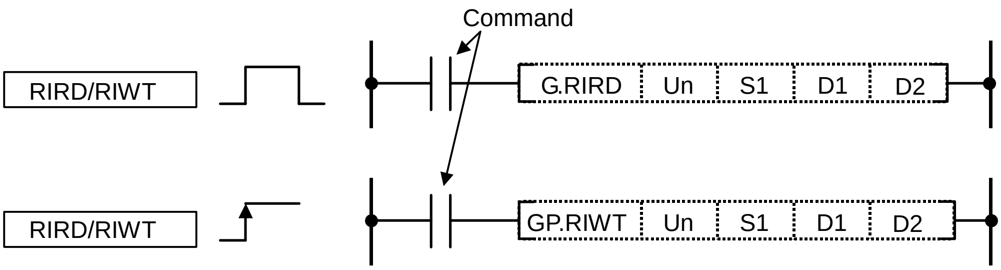


Coding

No. of steps	Com-mand	Device					
20	LDP						
26	MOV	H81	D0				
29	MOV	K6	D2				
32	MOV	K20	D4				
35	MOV	K3	D5				
38	MOV	K5	D7				
41	MOV	K20	D8				
44	MOV	K4	D9				
47	LDP						
53	MOV	K10	D300				
56	MOV	K20	D301				
59	MOV	K30	D302				
62	MOV	K40	D303				
65	LD						
66	G.SWRITE	U20	D0	D300	D50	M60	M100
73							

○ RIRD, RIWT ... Read/Write the device data

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S											○	○	○	○	○	○		○						5/11
D1											○	○	○	○	○	○		○						
D2	○	○	○	○	○	○	○		○															



(Note1) The stations targeted for this command are the QnCPU, QnACPU, ACPU or MELDAS master/local stations.

Setting data (RIRD)

Un **(Note2)**	Local station head input/output No.
S1	Head device of local station storing the control data
D1	Head No. of device storing read data
D2	Device that turns 1 scan ON at completion of read (D2)+1 device also turns ON at error completion.

Setting data (RIWT)

Un **(Note2)**	Local station head input/output No.
S1	Head device of local station storing the control data
D1	Head No. of device storing write data
D2	Device that turns 1 scan ON at completion of write (D2)+1 device also turns ON at error completion.

(Note2) Designate the Un value as U20 for the EXT1 card, as U28 for the EXT2 card, and as U30 for the EXT3 card.

Function

RIRD

Reads the device data from the designated station CPU.

(1) Control data

Device	Item	Set data	Setting range	Set by (*1)
(S) + 0	Completion status	Stores the status when the command is complete. 0 : No error (normal completion) Other than 0 : Error code	—	System
(S) + 1	Station number	Designate the station numbers of the local station and intelligent device station.	0 to 64	User
(S) + 2	Access code Attribute code	b15 b8 b7 b0 b15–b8: Access code, b7–b0: Attribute code	See (1) and (2).	User
(S) + 3	Device number	Designate device start number.	*2	User
(S) + 4	Number of points to read	Designate the read data count (in word units).	1 to 480 *3 1 to 32 *4	User

*1: User : Data to be set by user before executing exclusive command.
System : The results of the exclusive command executed is stored by the sequencer CPU.

*2: See the manual for the local station or the intelligent device station from which data will be read.

*3: Indicates the maximum number of data items that can be read.

*4: When the counterpart PLC CPU is other than QCPU (Q mode)/ QCPU (A mode)/ QnACPU/ AnUCPU and reads the PLC CPU device, the setting range will be 1 to 32 words.

(2) Device memory in the PLC CPU

Device contents	Name	Device type		Unit	Access code	Attribute code
		Bit	Word			
Input relay	X	○		Hexadecimal	01H	
Output relay	Y	○		Hexadecimal	02H	
Internal relay	M	○		Decimal	03H	
Latch relay	L	○		Decimal	83H	
Link relay	B	○		Hexadecimal	23H	
Timer (contact)	T	○		Decimal	09H	
Timer (coil)	T	○		Decimal	0AH	
Timer (present value)	T		○	Decimal	0CH	
Retentive timer (contact)	T	○		Decimal	89H	
Retentive timer (coil)	T	○		Decimal	8AH	
Retentive timer (present value)	T		○	Decimal	8CH	05H
Counter (contact)	C	○		Decimal	11H	
Counter (coil)	C	○		Decimal	12H	
Counter (present value)	C		○	Decimal	14H	
Data register	D		○	Decimal	04H	
Link register	W		○	Hexadecimal	24H	
File register	R		○	Decimal	84H	
Special link relay	SB	○		Hexadecimal	63H	
Special link register	SW		○	Hexadecimal	64H	
Special relay	SM	○		Decimal	43H	
Special register	SD		○	Decimal	44H	

(Note) Devices other than shown above cannot be accessed.
When accessing a bit device, specify it with 0 or a multiple of 16.

RIWT

Writes the device data to the designated station CPU.

(1) Control data

Device	Item	Set data	Setting range	Set by (*1)
(S) + 0	Completion status	Stores the status when the command is complete. 0 : No error (normal completion) Other than 0 : Error code	—	System
(S) + 1	Station number	Designate the station numbers of the local station and intelligent device station.	0 to 64	User
(S) + 2	Access code Attribute code	b15 to b8: Access code; b7 to b0: Attribute code	See (1) and (2).	User
(S) + 3	Device number	Designate device start number.	*2	User
(S) + 4	Number of points to read	Designate the read data count (in word units).	1 to 480 *3 1 to 32 *4	User

*1: User : Data to be set by user before executing exclusive command.
System : The results of the exclusive command executed is stored by the sequencer CPU.

*2: See the manual for the local station or the intelligent device station to which data will be written

*3: Indicates the maximum number of data items that can be written.

*4: When the counterpart PLC CPU is other than QCPU (Q mode)/ QCPU (A mode)/ QnACPU/ AnUCPU and writes the PLC CPU device, the setting range will be 1 to 10 words.

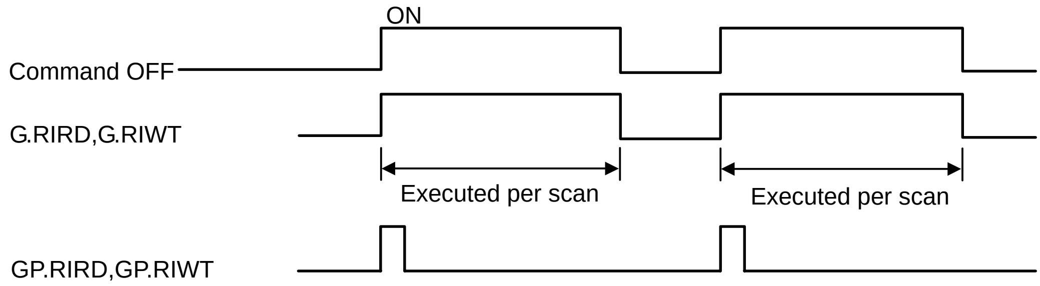
(2) Device memory in the PLC CPU

Device contents	Name	Device type: Bit	Device type: Word	Unit	Access code	Attribute code
Input relay	X	○		Hexadecimal	01H	05H
Output relay	Y	○		Hexadecimal	02H	
Internal relay	M	○		Decimal	03H	
Latch relay	L	○		Decimal	83H	
Link relay	B	○		Hexadecimal	23H	
Timer (contact)	T	○		Decimal	09H	
Timer (coil)	T	○		Decimal	0AH	
Timer (present value)	T		○	Decimal	0CH	
Retentive timer (contact)	T	○		Decimal	89H	
Retentive timer (coil)	T	○		Decimal	8AH	
Retentive timer (present value)	T		○	Decimal	8CH	
Counter (contact)	C	○		Decimal	11H	
Counter (coil)	C	○		Decimal	12H	
Counter (present value)	C		○	Decimal	14H	
Data register	D		○	Decimal	04H	
Link register	W		○	Hexadecimal	24H	
File register	R		○	Decimal	84H	
Special link relay	SB	○		Hexadecimal	63H	
Special link register	SW		○	Hexadecimal	64H	
Special relay	SM	○		Decimal	43H	
Special register	SD		○	Decimal	44H	

(Note) Devices other than shown above cannot be accessed.
When accessing a bit device, specify it with 0 or a multiple of 16.

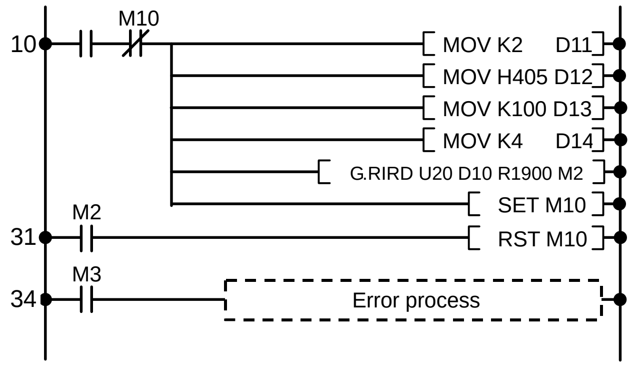
Execution conditions

The execution conditions for RIRD, RIWT are as shown below.



Program example

Program to read out the D100 to D103 data of the local station 2nd channel.



Coding

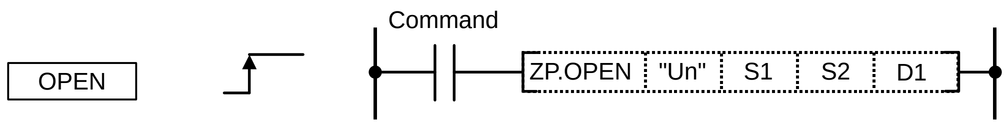
No. of steps	Com-mand	Device			
10	LD				
11	ANI	M10			
12	MOV	K2	D11		
15	MOV	H405	D12		
18	MOV	K100	D13		
21	MOV	K4	D14		
24	G.RIRD	U20	D10	R1900	M2
29	SET	M10			
31	LD	M2			
32	RST	M10			
34	LD	M3			
35					

Program example

(1) If the target station is MELDAS, only the device memory can be designated in the access code. The buffer memory in the CC-Link unit cannot be designated.
(2) This command is usable only with the FCU6-HR865 unit card version B and above. A timeout error will occur with earlier card versions.

○ **OPEN ... Opens a connection**

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S												○	○	○	○	○		○	○	○				12
D1												○	○	○	○	○		○						
D2		○	○	○	○	○	○		○															



Setting data	Details	Data type
"Un"	Head input/output No. for Ethernet I/F card	BIN16 bit
S1	Connection No. (1 to 8)	BIN16 bit
S2	Head device of local station storing control data	BIN16 bit
D1	Local device that turns 1 scan ON at completion of command. (D1)+1 device also turns ON at error completion.	Bit

Function

Opens a connection with the client device to communicate with.

(1) Control data

Device	Item	Setting data	Setting range	Setting side (*1)
(S2) + 0	Execution type/ completion type	Fix to a setting not used by GX Developer.	8000_H	User
(S2) + 1	Completion status	The status at completion is stored. 0000_H : Normal completion Other than 0000_H : Error completion (error code) Refer to "(2) Details of Error codes" for the contents of error.	—	System
(S2) + 2	Application setting area	Designate the connection application. b15 … b9 b8 … b1 b0 [| | (5) | (4) | | | (1)] (1) Buffer application (bit 0) 0 : Transmission 1 : Reception (4) Communication method (bit 8) 0 : TCP/IP 1 : UDP/IP (5) Communication procedure (bit 9) 0 : Procedural 1 : Non-procedural	(Indicated on left)	User

*1: User : Data to be set by user before executing exclusive command.
System : The results of the exclusive command executed is stored by the sequencer CPU.

Device	Item	Setting data	Setting range	Setting side
(S2) + 3	Blank		0	
(S2) + 4 to (S2) + 5	Client device IP address	Designate the client device's IP address. (Set 0×A8C00101 for IP = 168.192.1.1)	1_H to $FFFFFFFF_H$	User
(S2) + 6	Client device port No.	Designate the port No. for the client device.	1025 to 65535	User
(S2) + 7 to (S2) + 9	Blank		0	

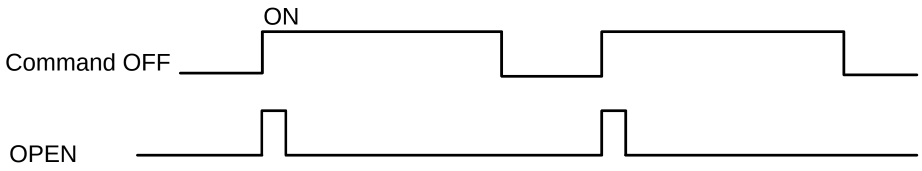
(2) Error codes

The errors that occur when each dedicated command is executed, and the error codes are show below.

Error code	Details	OPEN	BUFSND	BUFRCV	CLOSE
2111_H	The input/output No. is not 20, 28, or 30. The head character of the input/output No. is not "U". The Ethernet I/F card is not mounted in the designated slot.	○	○	○	○
4100_H	The device designation in the command is illegal.	○	○	○	○
$C014_H$	The initialization process or open process is not completed.		○	○	
$C017_H$	An error occurred when opening the TCP connection.	○			
$C020_H$	The data length exceeds the tolerable range.		○	○	
$C021_H$	An error completion response was received.		○		
$C022_H$	The response was not received within the response monitor timer value.		○		
$C030_H$	A transmission error occurred.		○		
$C033_H$	A client device with the set IP address does not exist.	○			
$C1A6_H$	BUFSND was executed in a connection opened for reception. BUFRCV was executed in a connection opened for transmission. An opened connection was opened again. The connection No. is not within the specified range.	○	○	○	○

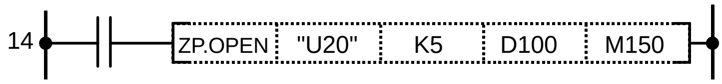
Execution conditions

The execution condition for OPEN is as shown below.



Program example

Program to open a connection to send data with port No. :6000 and connection No.:5.



No. of steps	Command	Device			
1	MOPV	H8000	D100		
4	MOV	H100	D102		
7	DMOV	H0C0A80103	D104		
11	MOV	K6000	D106		
14	ZP.OPEN	"U20"	K5	D100	M150
26					

Cautions

(1) Even if the dedicated commands (OPEN, CLOSE, BUFSND, BUFRCV) are executed with the high-speed PLC, the actual operation will take place at the same timing as the medium-speed PLC. Thus, the dedicated command should be executed with the medium-speed PLC.

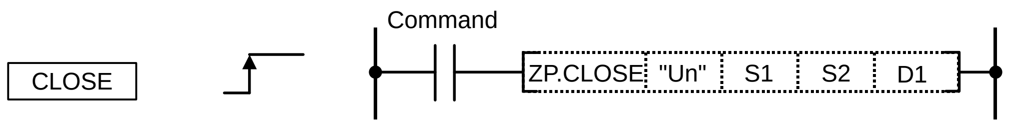
(2) Open transmission using TCP after the client device's reception has been opened. If transmission is opened before reception is opened, an error completion (completion status C033) will occur.
If transmission is opened for a client device with a nonexistent IP address, an error completion (completion status C033) will occur after one minute.

(3) If a connection opened with TCP is closed, and is tried to be opened again within a minute, an error completion (completion status C017) will occur. Wait a while before opening the connection again.

(4) If the connection No. is not within the specified range, an error completion (completion status C1A6) will occur, but the completion device and completion device +1 will not turn ON.

○ CLOSE ... Close the connection

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S												○	○	○	○	○		○	○	○				12
D1												○	○	○	○	○		○						
D2		○	○	○	○	○	○		○															



Setting data	Details	Data type
"Un"	Head input/output No. for Ethernet I/F card	BIN16 bit
S1	Connection No. (1 to 8)	BIN16 bit
S2	Head device of local station storing control data	BIN16 bit
D1	Local device that turns 1 scan ON at completion of command. (D1)+1 device also turns ON at error completion.	Bit

Function

Closes the connection with the client device that is communicating with.

(1) Control data

Device	Item	Setting data	Setting range	Setting side (*1)
(S2) + 0	System area	—	—	—
(S2) + 1	Completion status	The status at completion is stored. 0000_H : Normal completion Other than 0000_H : Error completion (error code)	—	System

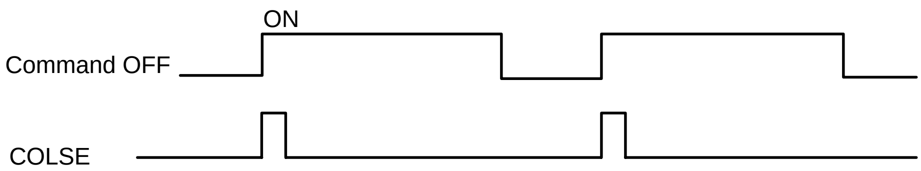
*1: User : Data to be set by user before executing exclusive command.
System : The results of the exclusive command executed is stored by the sequencer CPU.

(2) Error code

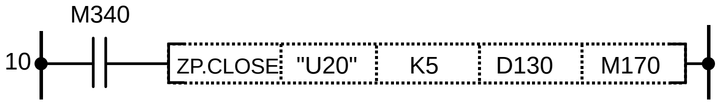
Refer to OPEN command for details of error code.

Execution conditions

The execution condition for CLOSE is as shown below.



Program example



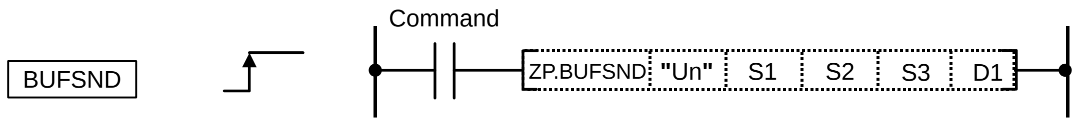
No. of steps	Command	Device			
10	LD	M340			
11	ZP.CLOSE	"U20"	K5	D130	M170
23					

Cautions

(1) Even if the dedicated commands (OPEN, CLOSE, BUFSND, BUFRCV) are executed with the high-speed PLC, the actual operation will take place at the same timing as the medium-speed PLC. Thus, the dedicated command should be executed with the medium-speed PLC.
(2) If a connection opened with TCP is closed, and is tried to be opened again within a minute, an error completion (completion status C017) will occur. Wait a while before opening the connection again.
(3) If the connection No. is not within the specified range, an error completion (completion status C1A6) will occur, but the completion device and completion device +1 will not turn ON.

○ BUFSND ... Data transmission (for fixed–buffer communication)

	Usable device																					Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1												○	○	○	○	○		○	○	○				13
S2												○	○	○	○	○		○						
S3												○	○	○	○	○		○						
D1		○	○	○	○	○	○		○															



Setting data	Details	Data type
"Un"	Head input/output No. for Ethernet I/F card	BIN16 bit
S1	Connection No. (1 to 8)	BIN16 bit
S2	Head device of local station storing control data	BIN16 bit
S3	Head No. of device to store the data to be sent	BIN16 bit
D1	Local device that turns 1 scan ON at completion of command. (D1)+1 device also turns ON at error completion.	Bit

Function

Sends data to client device. (Fixed-buffer communication.)

(1) Control data

Device	Item	Setting data	Setting range	Setting side (*1)
(S2) + 0	System area	—	—	—
(S2) + 1	Completion status	The status at completion is stored. 0000_H : Normal completion Other than 0000_H : Error completion (error code)	—	System

*1: User : Data to be set by user before executing exclusive command.
System : The results of the exclusive command executed is stored by the sequencer CPU.

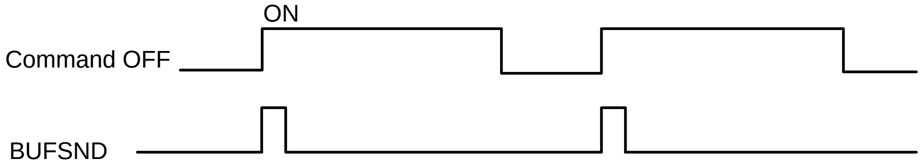
(2) Send data

Device	Item	Setting data	Setting range	Setting side
(S3) + 0	Send data length	Designate the send data length. Designate the data length according to the communication procedure.		User
		Procedural : No. of words	1 to 1017	
		Non-procedural : No. of bytes	1 to 2046	
(S3) + 1 to (S3) + n	Send data	Designate the send data.	—	User

(3) Error code
Refer to OPEN command for details of error code.

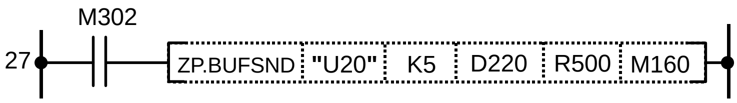
Execution conditions

The execution condition for BUFSND is as shown below.



Program example

The program to send a connection to send in R500 and above with the port No. :6000 and the connection No.:5.



No. of steps	Command	Device				
27	LD	M302				
28	ZP.BUFSND	"U20"	K5	D220	R500	M160
41						

Cautions

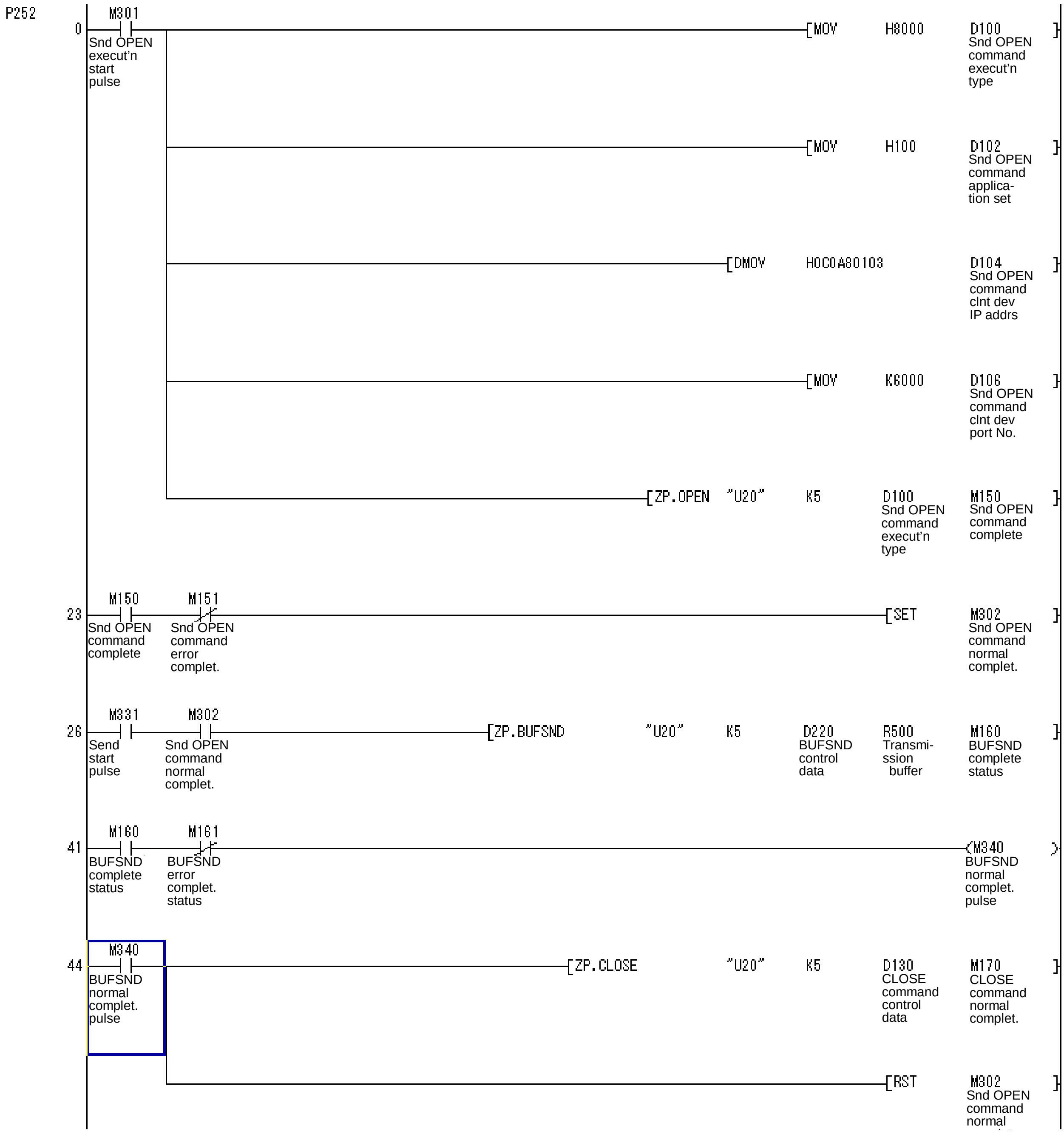
(1) Even if the dedicated commands (OPEN, CLOSE, BUFSND, BUFRCV) are executed with the high-speed PLC, the actual operation will take place at the same timing as the medium-speed PLC. Thus, the dedicated command should be executed with the medium-speed PLC.

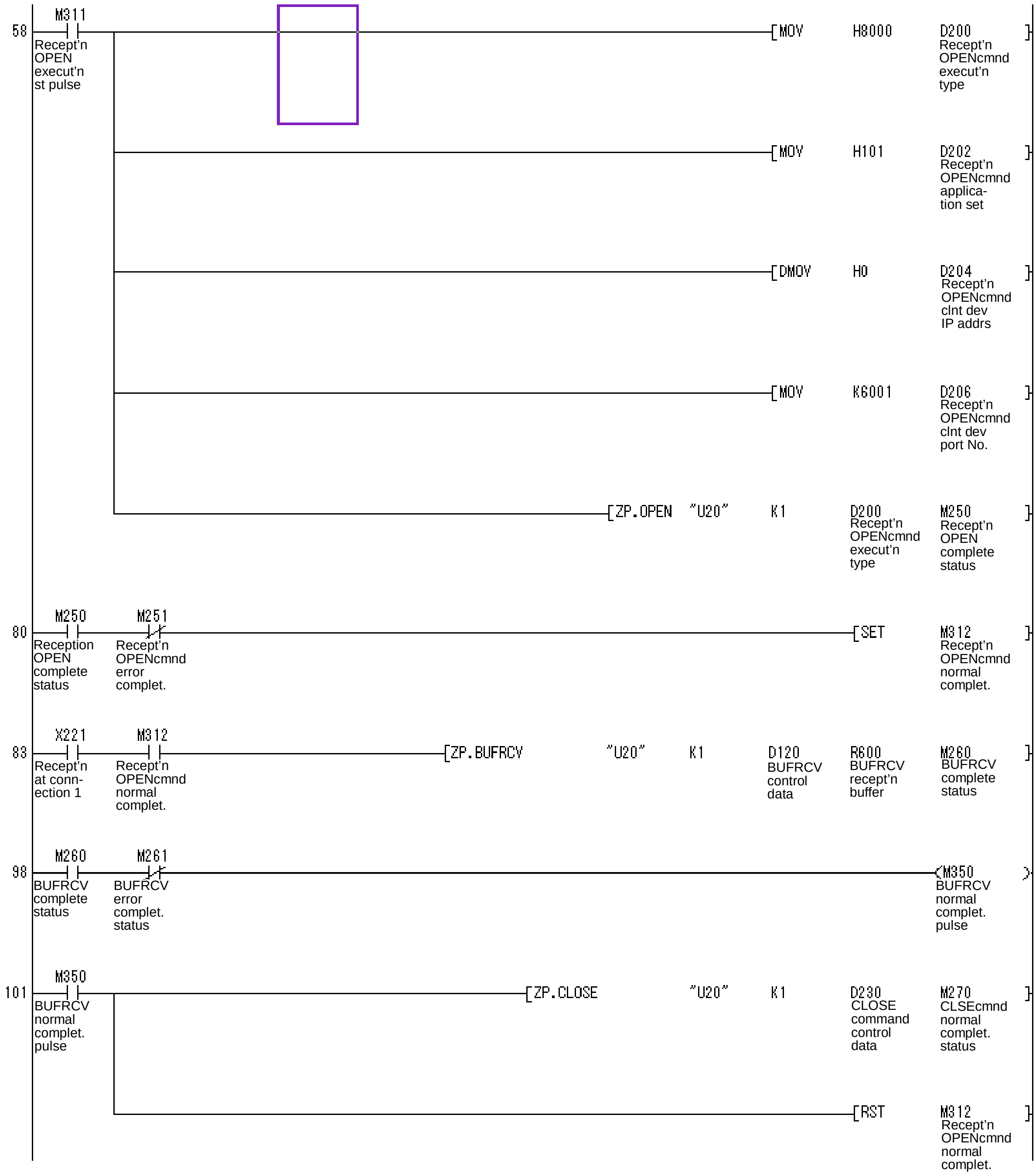
(2) Do not update the BUFSND command transmission buffer until the command is completed. Failure to observe this could result in malfunctions.

(3) If the connection No. is not within the specified range, an error completion (completion status C1A6) will occur, but the completion device and completion device +1 will not turn ON.

Example of data communication program

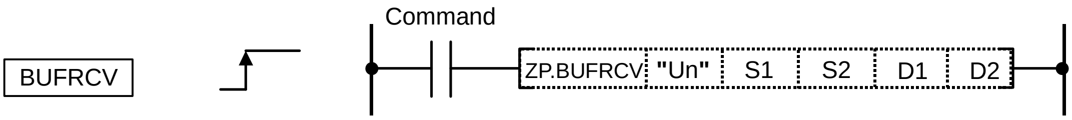
The following is an example of the program that executes sending (BUFSND) and receiving (BUFRCV). UDP/IP, non-procedural type are used for both commands.
The send destination IP address is 192.168.1.3, and the applicable port No. is 6000. Connection 5 is used. In the actual program, the send data is set in R500 and above.




58
M311
Recept'n OPEN execut'n st pulse
[MOV H8000 D200
Recept'n OPENcmnd execut'n type
[MOV H101 D202
Recept'n OPENcmnd applica-tion set
[DMOV H0 D204
Recept'n OPENcmnd clnt dev IP addrs
[MOV K6001 D206
Recept'n OPENcmnd clnt dev port No.
[ZP.OPEN "U20" K1 D200 M250
Recept'n OPENcmnd execut'n type
Recept'n OPEN complete status
80
M250
Reception OPEN complete status
M251
Recept'n OPENcmnd error complet.
[SET M312
Recept'n OPENcmnd normal complet.
83
X221
Recept'n at conn-ection 1
M312
Recept'n OPENcmnd normal complet.
[ZP.BUFRCV "U20" K1 D120 R600 M260
BUFRCV control data
BUFRCV recept'n buffer
BUFRCV complete status
98
M260
BUFRCV complete status
M261
BUFRCV error complet. status
(M350
BUFRCV normal complet. pulse
101
M350
BUFRCV normal complet. pulse
[ZP.CLOSE "U20" K1 D230 M270
CLOSE command control data
CLSEcmnd normal complet. status
[RST M312
Recept'n OPENcmnd normal complet.

○ BUFRCV ... Data reception (for fixed–buffer communication)

	Usable device																				Digit designation	Index	No. of steps
	Bit device										Word device								Constant		Pointer			
	X	Y	M	L	F	B	SB	T	SM	V	T	C	D	R	W	SW	Z	SD	K	H	P			
S1												○	○	○	○	○		○	○	○				13
S2												○	○	○	○	○		○						
D1												○	○	○	○	○		○						
D2		○	○	○	○	○	○		○															



Setting data	Details	Data type
"Un"	Head input/output No. for Ethernet I/F card	BIN16 bit
S1	Connection No. (1 to 8)	BIN16 bit
S2	Head device of local station storing control data	BIN16 bit
D1	Head No. of device to store the received data	BIN16 bit
D2	Local device that turns 1 scan ON at completion of command. (D1)+1 device also turns ON at error completion.	Bit

Function

Reads the data received from the client device. (Fixed-buffer communication.)
Used by the main program.

(1) Control data

Device	Item	Setting data	Setting range	Setting side (*1)
(S2) + 0	System area	—	—	—
(S2) + 1	Completion status	The status at completion is stored. 0000_H : Normal completion Other than 0000_H : Error completion (error code)	—	System

*1: User : Data to be set by user before executing exclusive command.
System : The results of the exclusive command executed is stored by the sequencer CPU.

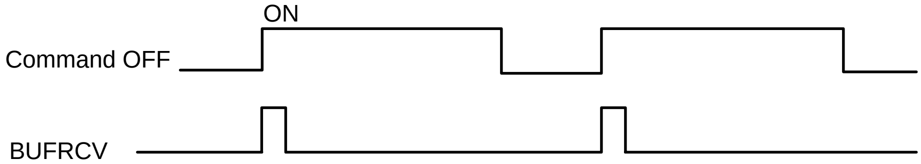
(2) Received data

Device	Item	Setting data	Setting range	Setting side (*1)
(D1) + 0	Received data length	Designate the received data length. Designate the data length according to the communication procedure.		System
		Procedural : No. of words	1 to 1017	
		Non-procedural : No. of bytes	1 to 2046	
(D1) + 1 to (D1) + n	Received data	Designate the received data.	—	System

(3) Error code
Refer to OPEN command for details of error code.

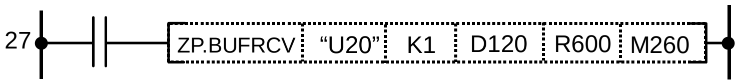
Execution conditions

The execution condition for BUFRCV is as shown below.



Program example

The program to receive data stored in R600 and above with the port No. :6001 and the connection No.:1.



No. of steps	Command	Device				
27	LD	M302				
28	ZP.BUFRCV	"U20"	K1	D120	R600	M260
41						

Cautions

(1) Even if the dedicated commands (OPEN, CLOSE, BUFSND, BUFRCV) are executed with the high-speed PLC, the actual operation will take place at the same timing as the medium-speed PLC. Thus, the dedicated command should be executed with the medium-speed PLC.

(2) Refer to the BUFRCV command reception buffer after the command is completed. Failure to observe this could result in malfunctions.

(3) If the connection No. is not within the specified range, an error completion (completion status C1A6) will occur, but the completion device and completion device +1 will not turn ON.

10. Exclusive Commands 2

Although the basic and functional commands are not used only for specific purposes, some commands may be efficient if command applications such as data transfer between under PLC and controller and controller display screen are limited.
Then, we provide a number of exclusive commands which are explained below.

Examples of exclusive commands:

- Intelligent function command (FROM, TO)
- Transient command (OPEN, CLOSE, BUFSND, BUFRCV)
- Transient transmission command (READ, SREAD, WRITE, SWRITE, RIRD, RIWT)

Refer to "9. Exclusive Command 1".

- ATC dedicated command (ATC)
- Rotary body control command (ROT)
- Tool life management exclusive command (TSRH)
- DDB (direct data bus) ..... asynchronous
- External search .... synchronous

Refer to "10. Exclusive Command 2".

10.1 ATC Exclusive Command

10.1.1 Outline of ATC Control

The ATC (Automatic Tool Change) can be controlled in the following two ways:

(1) Mechanical random control

With the information of magazine position from the machine, and T command, the control system determines the direction of magazine rotation, number of steps required, etc. for index of the magazine, according to the given command.
Each tool and magazine tool pot (socket) have a one-on-one corresponding relation.
Usually, the "intermediate pot" that supports the transfer of the tool is provided between the spindle and the magazine.
This control is possible by not using ATC command, but ROT command only.

(2) Memory random control

With the information of magazine rotation, or magazine position from the machine, the control system refers to tool No. stored in the memory. For index of the magazine, the direction of magazine rotation and number of steps are determined by the given T command and tool No. stored in the memory.
Each tool and magazine tool pot (socket) does not always have a one-on-one corresponding relation.
Usually, the "intermediate pot" is not provided.

10.1.2 ATC Operation

The motions related to ATC operation can be largely divided into the following four motions:

(1) Index of magazine ……………………...………. (ATC-K1, K2, K5, K6, K7, K8)
(2) Tool change (arm, or the like is used) ……...... (ATC-K3, K4)
(3) Transfer of tool to intermediate pot or arm ..... (Normal function commands such as MOV, XCH are used.)
(4) Others ..………………………………………….. (ATC-K9, K10, K11)

10.1.3 Explanation of Terminology

(1) Pointer

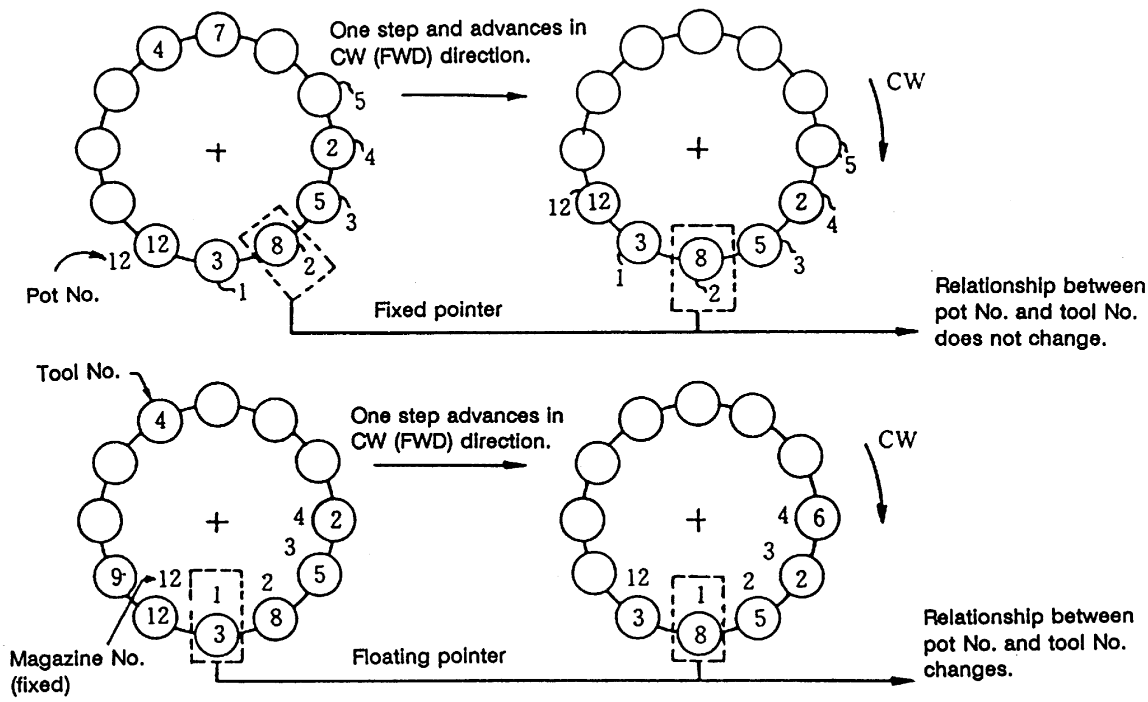
This points out the position where the magazine is indexed. When a tool table in which tool No. are previously recorded is used, the tool table does not rotate with rotation of the magazine and the pointer serves as "ring counter" for control of magazine position.

(2) Fixed pointer

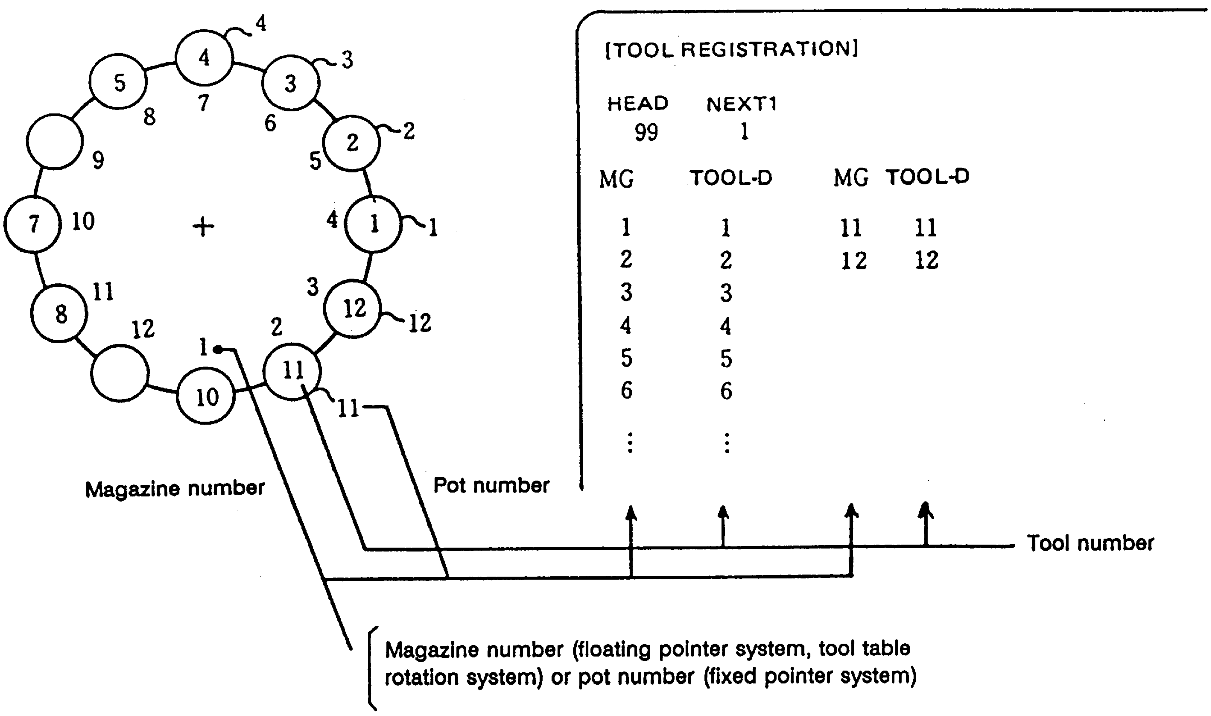
This is the type with tool pots numbered and the relationship between tool pot and tool No. is fixed if the magazine is rotated. When the tool table is rotated, fixed pointer does not functionally differ from "floating pointer".

(3) Floating pointer

This is the type with numbered fixed position on magazine and the relationship between magazine No. and tool No. changes when the magazine rotates.



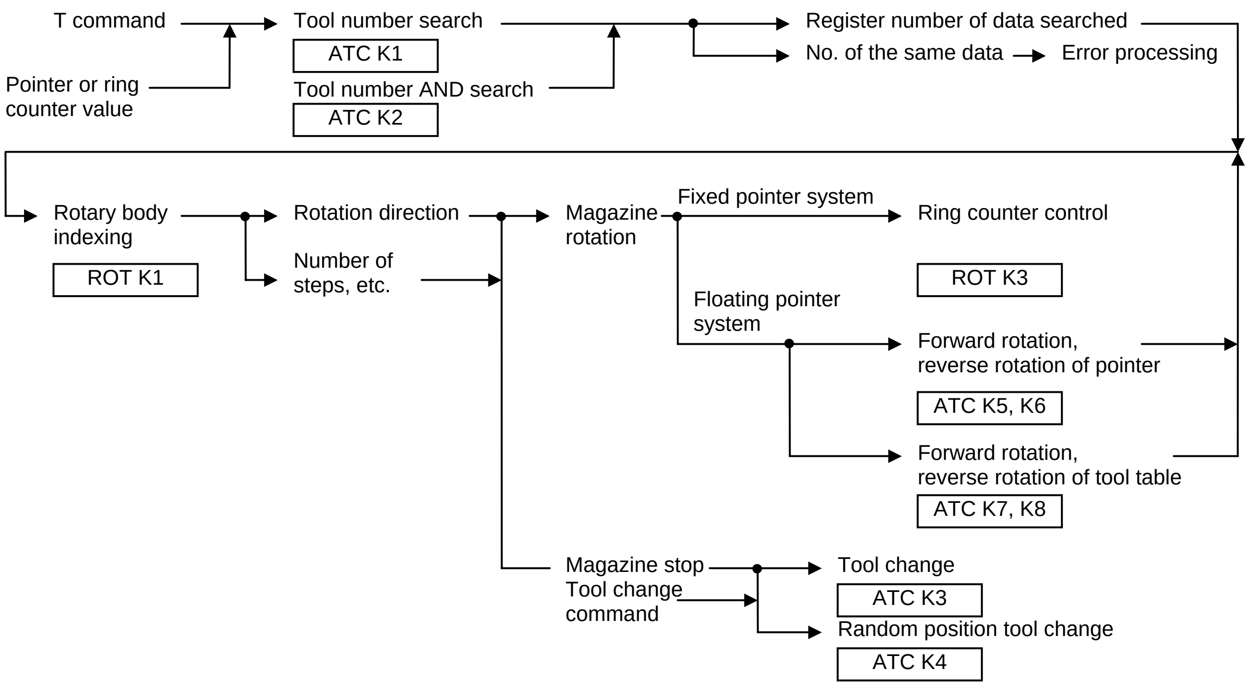
10.1.4 Relationship between Tool Registration Screen and Magazines



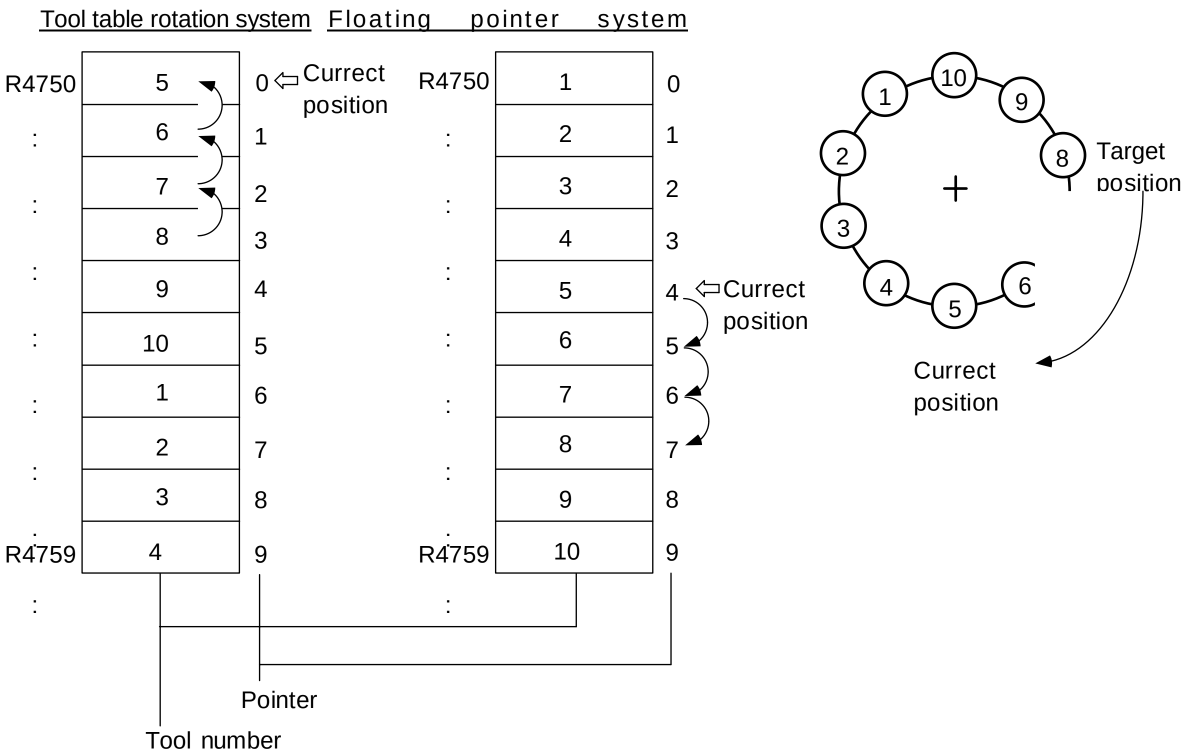
When the floating pointer system or tool table rotation system is selected on the tool registration screen, correspondence display between the magazines and tools changes each time the magazine rotates; when the fixed pointer system is selected, it does not change.

10.1.5 Use of ATC and ROT Commands

The use order of the ATC and ROT commands during the T command or tool change command is shown below:



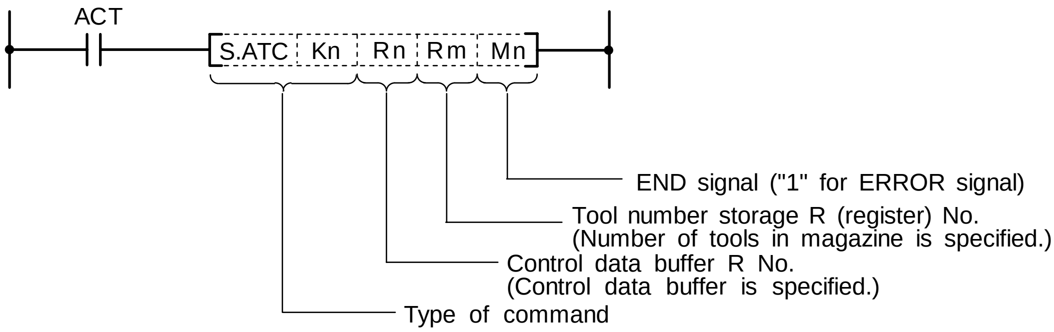
The relationship between the tool number search command and rotary body indexing command when the tool table rotation system or floating pointer system is used is explained below.

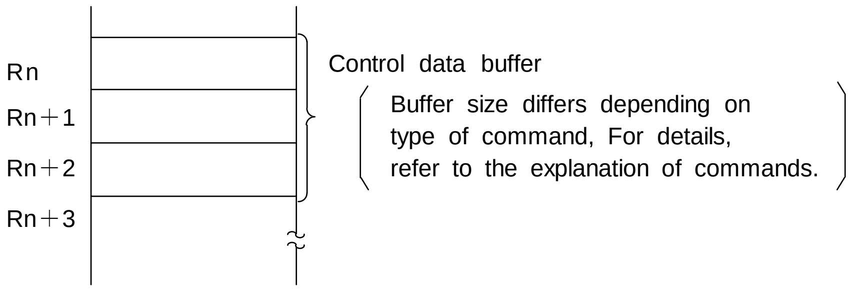


(1) Index tool number 8 in the situation shown above
    (a) In the tool table rotation system, the tool number search command outputs 3.
    (b) In the floating pointer system, the tool number search command outputs 7.
(2) The tool number search command output result is used by the rotary body indexing command to find the rotation direction, the number of steps, etc.
    (a) In the tool table rotation system, rotation direction CW and number of steps 3 are found from the relationship between current value 0 (pointer 0) and tool number search output result 3.
    (b) In the floating pointer system, rotation direction CW and number of steps 3 are found from the relationship between current value 4 (pointer 4) and tool number search output result 7, as in (a) above.

In the fixed pointer system, the pointer is fixed to 0 and the ring counter of 0 to n-1 (n is the number of magazines) separate from the pointer is controlled. The counter value is used as the current position.

10.1.6 Basic Format of ATC Exclusive Command





10.1.7 Command List

Command	Description
S.ATC K1 Rn Rm Mn	Tool No. search
S.ATC K2 Rn Rm Mn	Tool No. logical product search
S.ATC K3 Rn Rm Mn	Tool change
S.ATC K4 Rn Rm Mn	Random position tool change
S.ATC K5 Rn Rm Mn	Pointer forward rotation
S.ATC K6 Rn Rm Mn	Pointer reverse rotation
S.ATC K7 Rn Rm Mn	Tool table forward rotation
S.ATC K8 Rn Rm Mn	Tool table reverse rotation
S.ATC K9 Rn Rm Mn	Tool data read
S.ATC K10 Rn Rm Mn	Tool data write
S.ATC K11 Rn Rm Mn	Automatic tool data write

10.1.8 Control Data Buffer Contents

	Command	Rn	Rn+1	Rn+2
1	Tool No. search	R No. to store search data	R No. to which data output	—
2	Tool No. logical product search	R No. to store search data	R No. to which data output	Logical product data position R No.
3	Tool change (Ex.: Spindle↔Index position)	R No. to specify the position of tool change	—	—
4	Random position tool change	R No. to specify the position of tool change	R No. to specify the tool to be changed	—
5	Pointer forward rotation	—	—	—
6	Pointer reverse rotation	—	—	—
7	Tool table forward rotation	—	—	—
8	Tool table reverse rotation	—	—	—
9	Tool data read	R No. for magazine position (to be read)	R No. to which data read	—
10	Tool data write	R No. for magazine position (to be written)	R No. to which data written	—
11	Automatic tool data write	R No. to store Initial data	—	—

10.1.9 File Register (R Register) Assignment and Parameters

(1) File registers for ATC control

The file registers used with the ATC are as shown below.

		Corresponding file (R) register						Remarks (data type)
Magazine		No. 1 magazine		No. 2 magazine		No. 3 magazine		
T4-digit/T8-digit specifications		T4-digit	T8-digit	T4-digit	T8-digit	T4-digit	T8-digit	
ATC control parameters		R4700	←	←	←	←	←	—
No. of magazine designation		R4710	←	R4711	←	R4712	←	Binary
Pointer designation		R4715	←	R4716	←	R4717	←	Binary
Spindle tool		R4720	R4720 R4721	R4730	R4730 R4731	—	—	BCD
Standby 1 tool		R4721	R4722 R4723	R4731	R4732 R4733	—	—	BCD
Standby 2 tool		R4722	R4724 R4725	R4732	R4734 R4735	—	—	BCD
Standby 3 tool		R4723	R4726 R4727	R4733	R4736 R4737	—	—	BCD
Standby 4 tool		R4724	R4728 R4729	R4734	R4738 R4739	—	—	BCD
AUX data		R4748	←	←	←	←	←	Binary (0~99)
Magazine tool data	MG1	R4750	R4750 R4751	R4990	R4990 R4991	R5230	R5230 R5231	BCD
	MG2	R4751	R4752 R4753	R4991	R4992 R4993	R5231	R5232 R5233	BCD
	MG3	R4752	R4754 R4755	R4992	R4994 R4995	R5232	R5234 R5235	BCD
	≈	≈	≈	≈	≈	≈	≈	≈
	MG79	R4828	R4906 R4907	R5068	R5146 R5147	R5308	R5386 R5387	BCD
	MG80	R4829	R4908 R4909	R5069	R5148 R5149	R5309	R5388 R5389	BCD

(Note 1) A maximum of 80 tools per magazine can be used.
(Note 2) The tool registration screen has been prepared only for the No. 1 magazine.

(2) Control parameter contents

R4700	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

$\underbrace{\text{F E D C B A 9 8}}$ Max. number of standby displayed: 4

Bit 0:
0 : T 4-digit
1 : T 8-digit

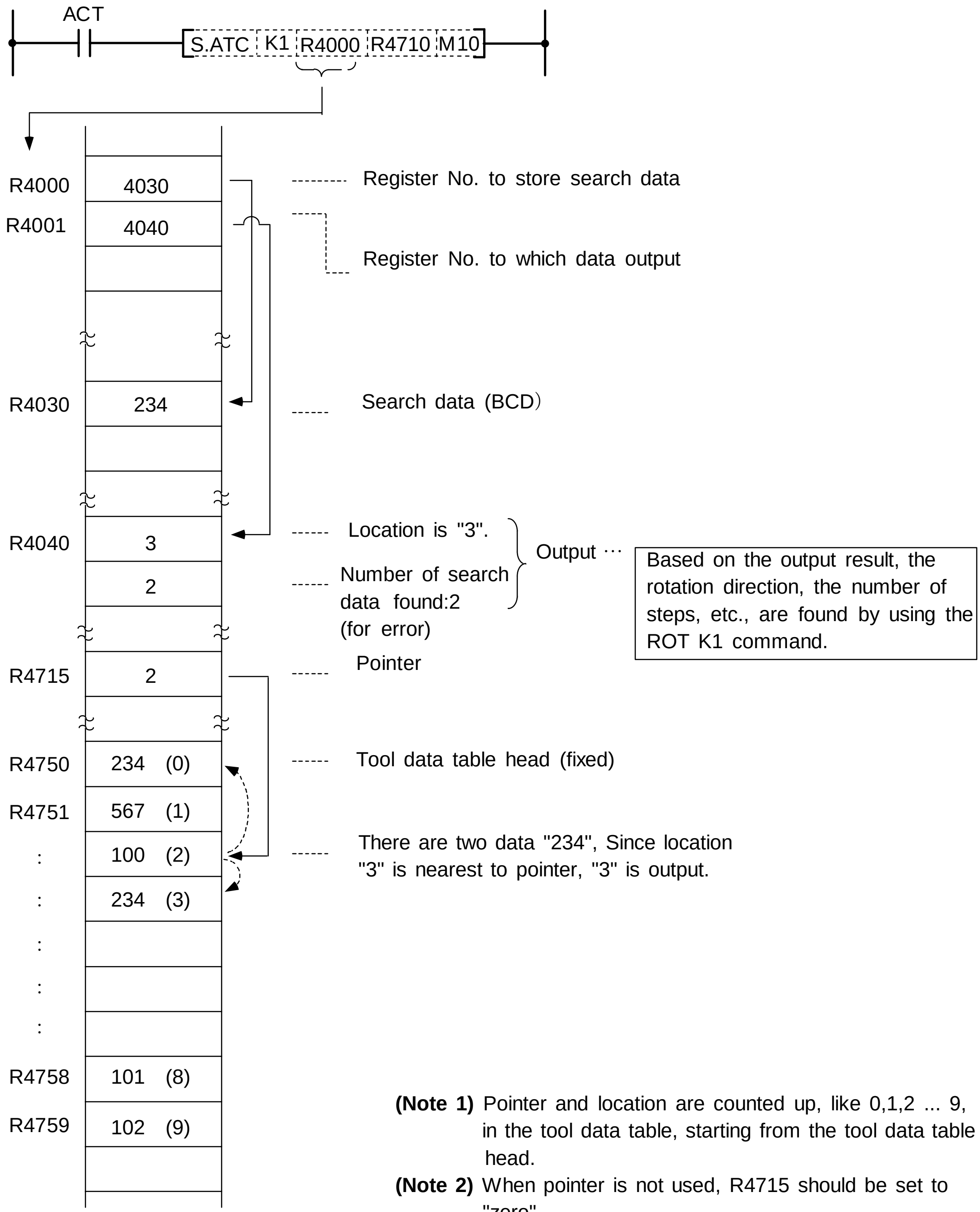
Bit 1:
0 : Magazine starts from "1".
1 : Magazine starts from "0".

For details on the control parameters, refer to "10.1.12 Examples of Tool Registration Screen".

10.1.10 Details of Each Command

(1) Tool No. search

This command is used to search for tool No. stored in the tool data table.
When the command tool No. is found, number of searched data and its location are output. If two or more tool No. are found, the location of tool No. nearest to the pointer is output.



(Example for 10 magazines)

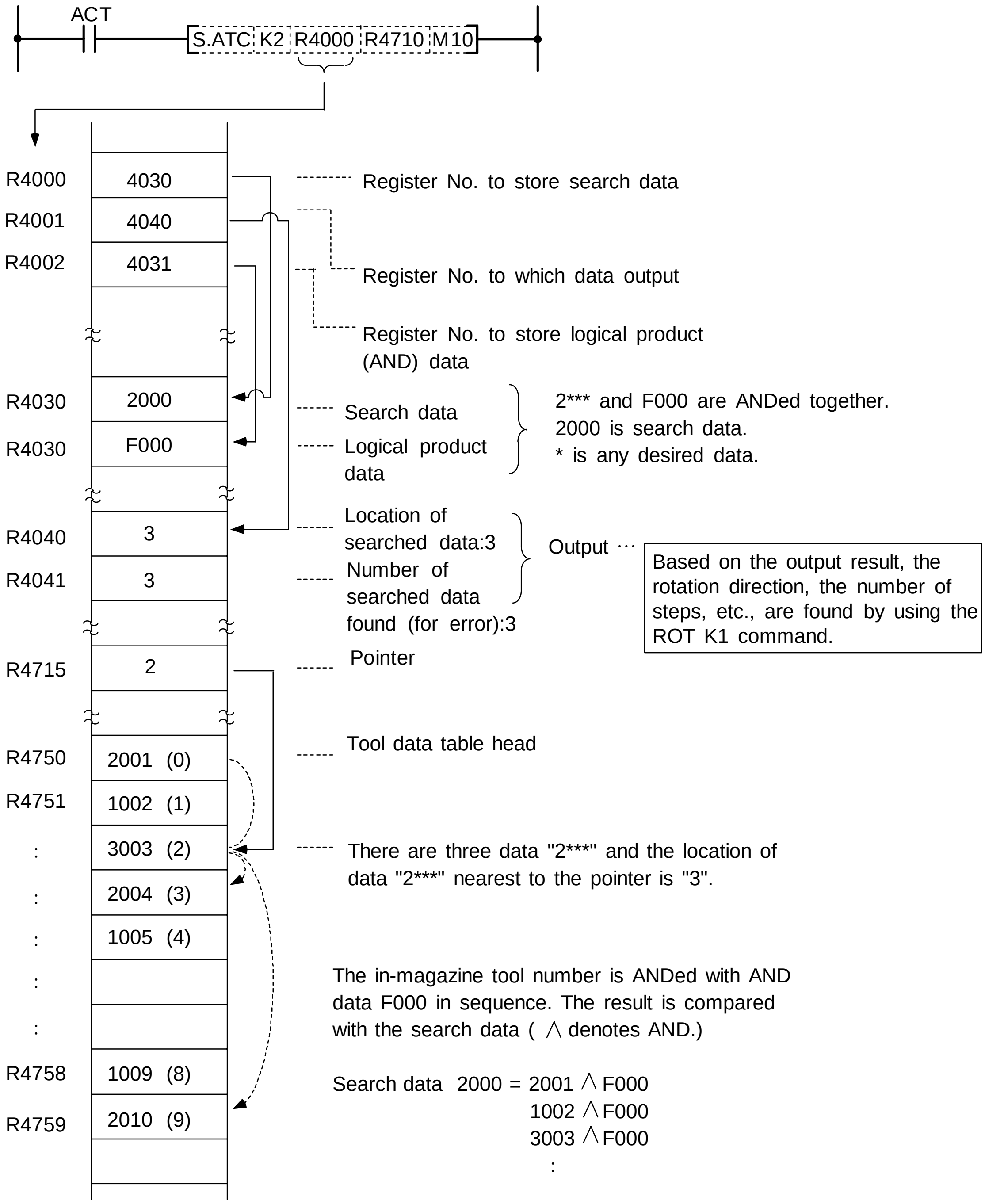
(Note 1) Pointer and location are counted up, like 0,1,2 ... 9, in the tool data table, starting from the tool data table head.

(Note 2) When pointer is not used, R4715 should be set to "zero".

(Ex.) MOV K0 R4715

(2) Tool No. logical product (AND) search

Tool number AND search is the same as the tool number search command (ATC K1) in function: search data and in-magazine tool number and AND data are ANDed together for a search.



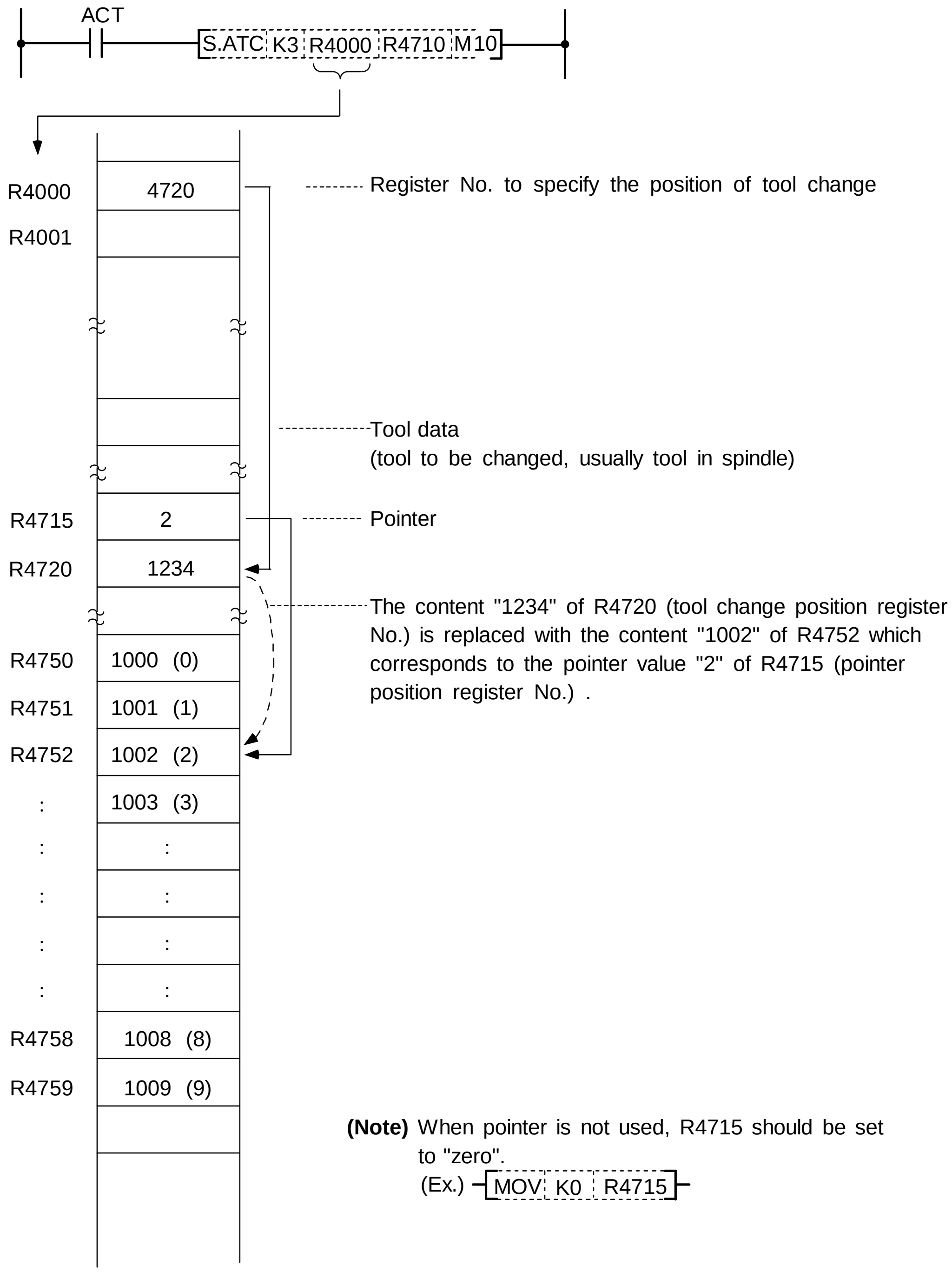
(Note 1) Pointer and location are counted up, like 0,1,2 ... 9, in the tool data table, starting from the tool data table head.

(Note 2) When pointer is not used, R4751 should be set to "zero".

(Ex.) MOV K0 R4715

(3) Tool change

When a spindle tool and a magazine index tool are exchanged by the ATC arm, etc., the contents in the memory (R register) must be updated correspondingly.

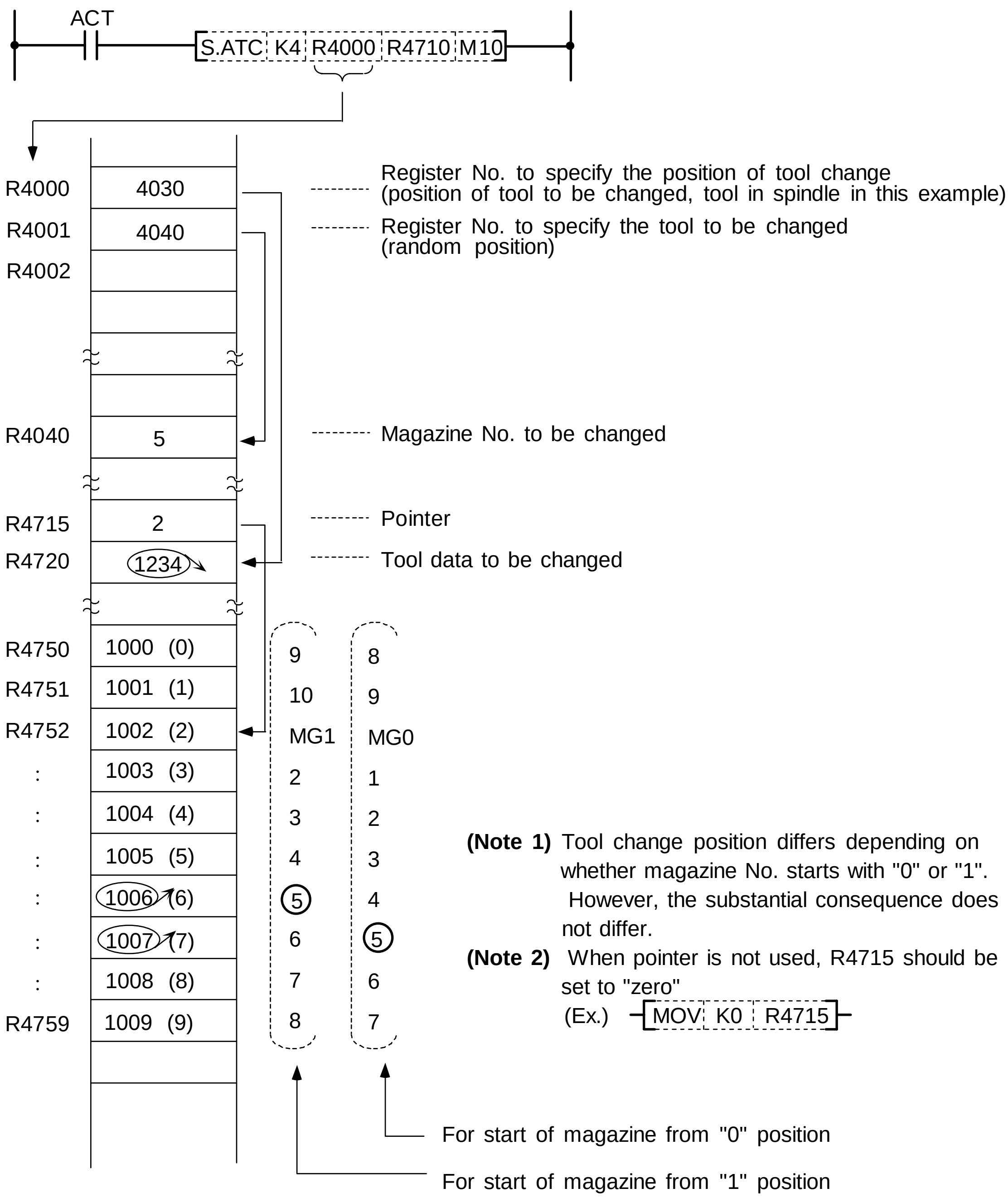


(Note) When pointer is not used, R4715 should be set to "zero".

(Ex.) [MOV K0 R4715]

(4) Random position tool change

In tool change, a spindle tool is usually exchanged with a magazine index tool. It may often occur, however, that tool change must be performed at a station other than the usual tool change position (tool change at auxiliary tool change position, for example). This command is used in such cases.



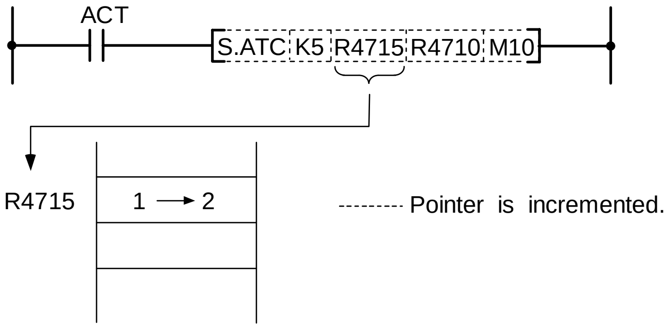
(Note 1) Tool change position differs depending on whether magazine No. starts with "0" or "1". However, the substantial consequence does not differ.

(Note 2) When pointer is not used, R4715 should be set to "zero"

(Ex.) MOV K0 R4715

(5) Pointer "FWD" rotation

In the ATC control with floating pointer, pointer count is controlled so that it coincides with the actually indexed magazine position when the magazine rotates in "FWD" direction for index.

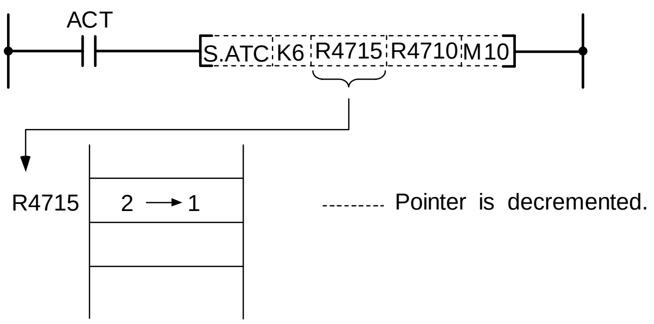


When a magazine with 10 tools is used, the control sequence is as follows:
0, 1, 2, 3 .... 9, 0, 1, 2, ... 8, 9, 0, 1 ...

(Note 1) When this command is executed, the relationship between magazine No. and tool No., appearing on the tool entry display, changes accordingly.

(6) Pointer "REV" rotation

In the ATC control with floating pointer, pointer count is controlled so that it coincides with actually indexed magazine position when the magazine rotates in "REV" direction for index.

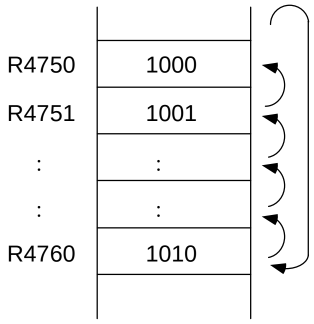


When a magazine with 10 tools is used, for example, the control sequence is as follows:
2, 1, 0, 9, 8 .... 2, 1, 0, 9, 8 .... 1, 0, 9, 8 ...

(Note 1) When this command is executed, the relationship between magazine No. and tool No., appearing on the tool entry display, changes accordingly.

(7) Tool table "FWD" rotation

The tool table rotates in "FWD" direction in accordance with the magazine rotation.

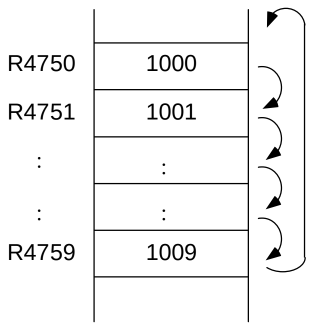


(Note 1) In this control mode, pointer always indicates "0" (tool table head).

(Note 2) When this command is executed, the relationship between magazine No. and tool No., appearing on the tool entry display, changes accordingly.

(8) Tool table "REV" rotation

The tool table rotates in "REV" direction in accordance with the magazine rotation.

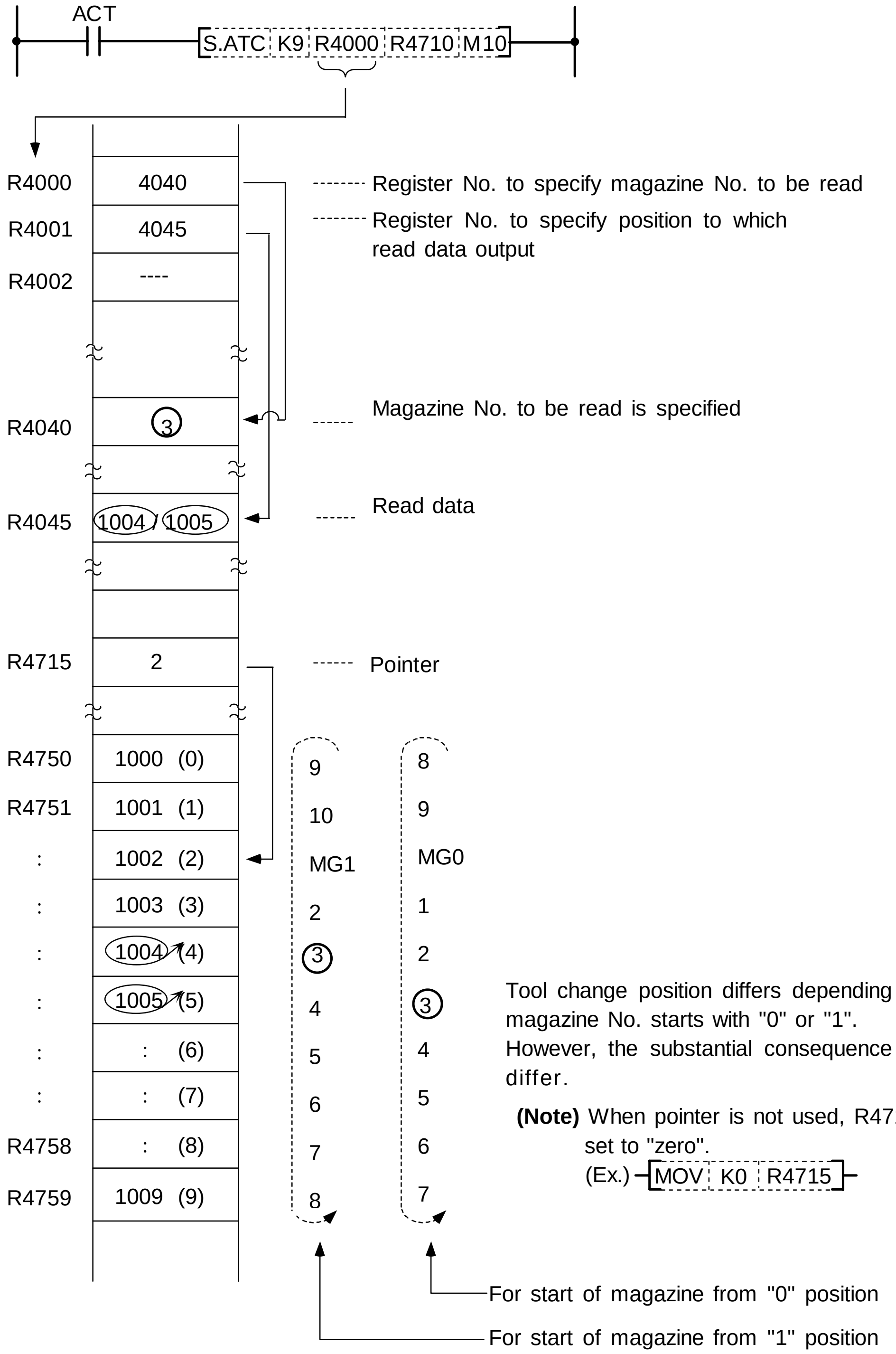


(Note 1) In this control mode, pointer always indicates "0" (tool table head).

(Note 2) When this command is executed, the relationship between magazine No. and tool No., appearing on the tool entry display, changes accordingly.

(9) Tool data read

This command is used to call a specific tool No. in the magazine.

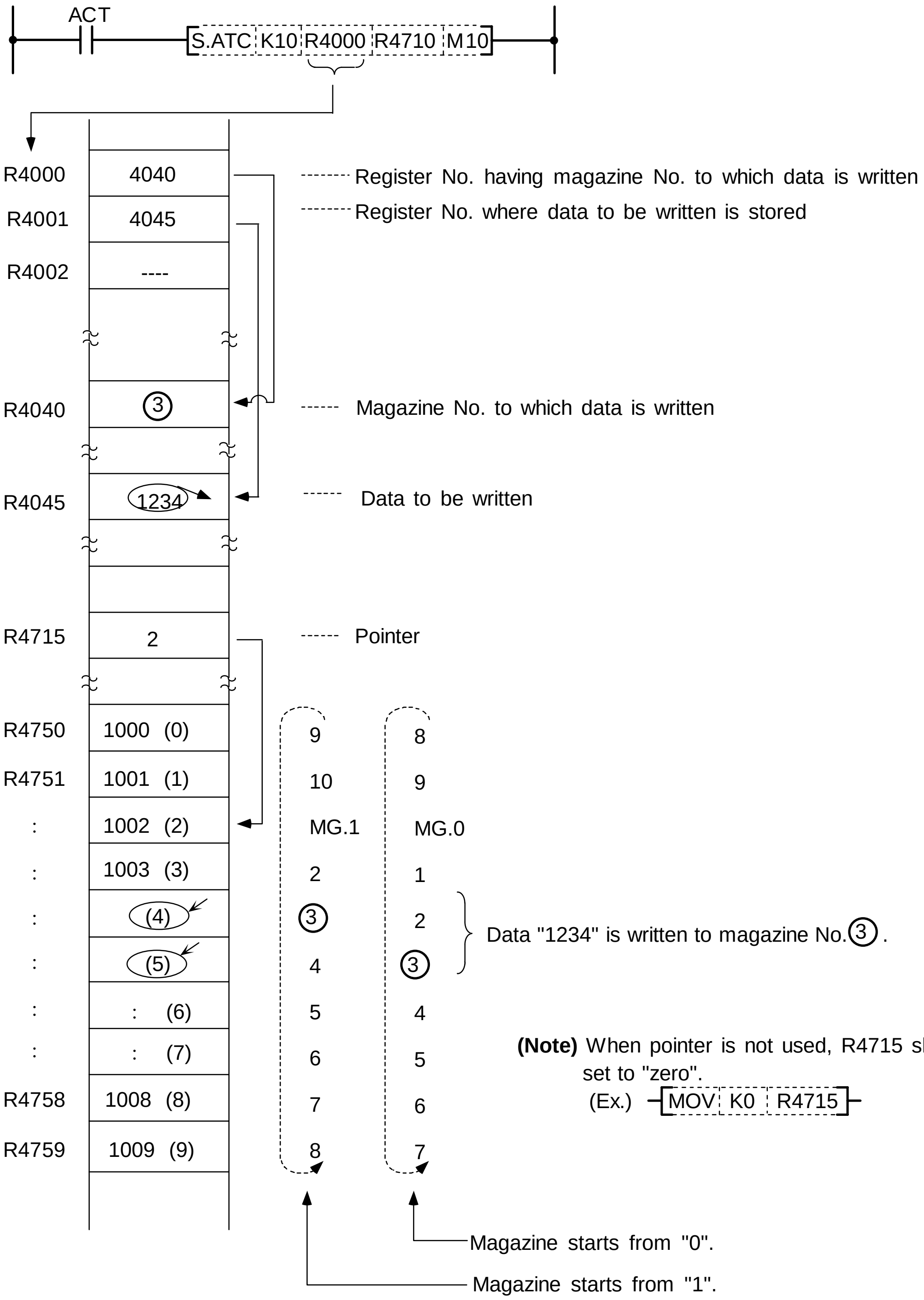


Tool change position differs depending on whether magazine No. starts with "0" or "1".
However, the substantial consequence does not differ.

(Note) When pointer is not used, R4715 should be set to "zero".
(Ex.) MOV K0 R4715

(10) Tool data write

Instead of setting tool No. through the setting and display unit, the tool No. is entered to each magazine No. set through PLC program.

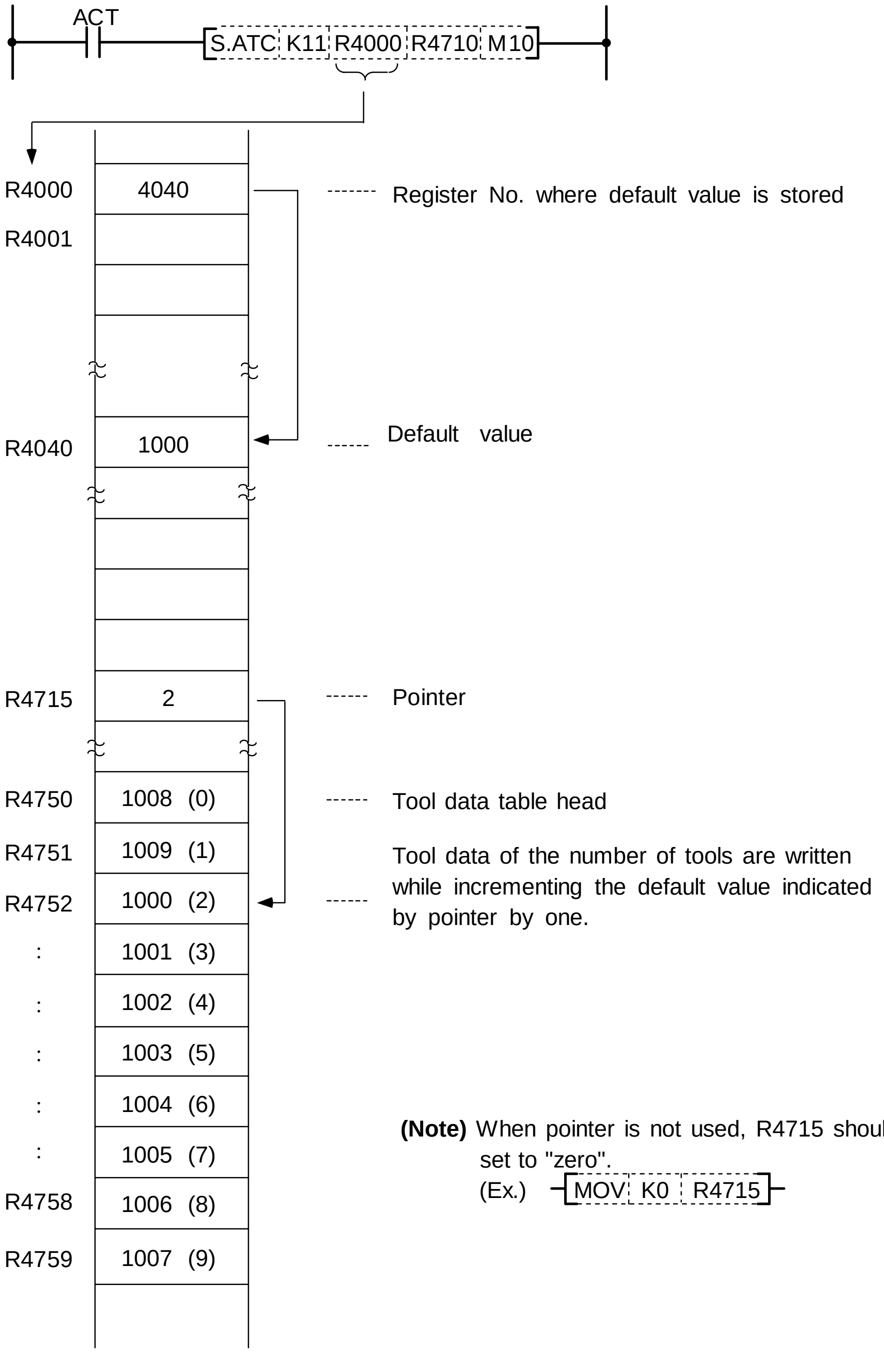


(Note) When pointer is not used, R4715 should be set to "zero".

(Ex.) MOV K0 R4715

(11) Automatic tool data write

All tool Nos. are written (entered) in batch. This command is used for initialization, etc.
The data are written one after another for each tool, starting from the default value.

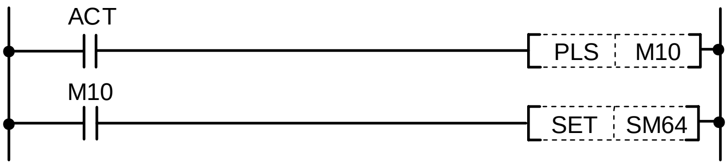


(Note) When pointer is not used, R4715 should be set to "zero".
(Ex.) MOV K0 R4715

10.1.11 Precautions for Using ATC Exclusive Instructions

(1) When tool data is rewritten by ATC or other than ATC command, tool registration screen display is not updated. The following processing is required:
- Turn on special relay SM64 by using the SET command.

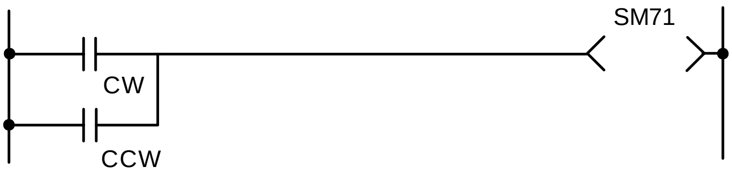
Program example)



- SM64 processing is not required for ATC commands ATC K5, K6 (forward rotation, reverse rotation of pointer), ATC K7, K8 (forward rotation, reverse rotation of tool table).
- SM64 is set through the use of the user PLC and reset by controller.

(2) Method of tool registration prohibiting during magazine rotation
If tool data is set on the tool registration screen during magazine rotation, data may be set in erroneous position. To prevent this error, a signal called special relay SM71 is provided.
- Turn on SM71 during magazine rotation.

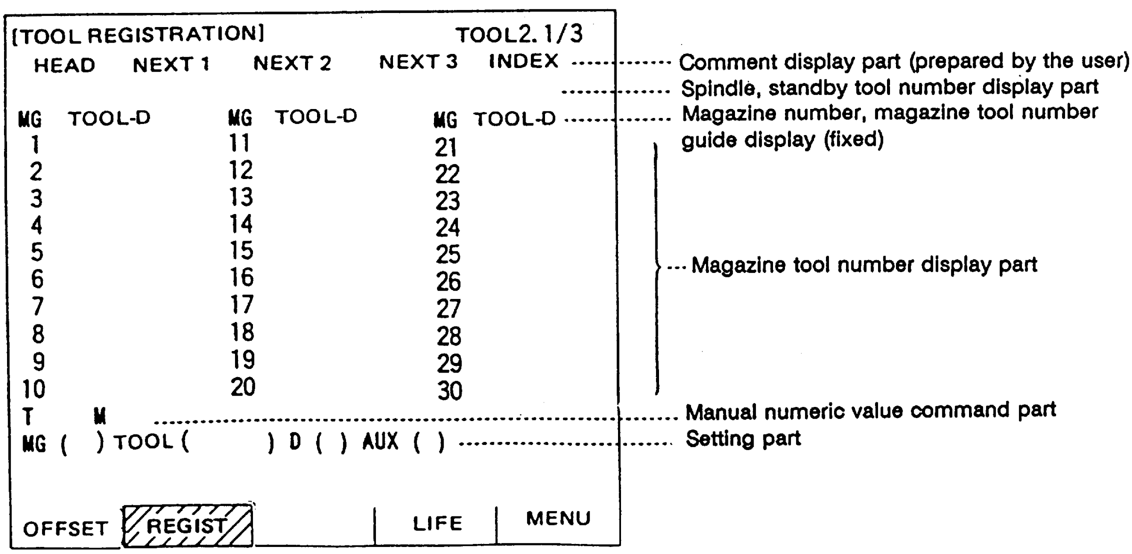
Program example)



- Setting of AUX data (R4748) is valid while SM71 is being ON.

10.1.12 Examples of Tool Registration Screen

Tool registration screen examples are given below.
For operation, refer to the Operation Manual.



(1) Comment display part

Comment in the comment display part is prepared by the user with the comment display function. Refer to the section "1.4 Creating PLC-related Data".

(2) Spindle tool, standby tool display part

The number of display items can be changed according to the control parameter value.

Control parameter (R4700)

F	E	D	C	B	A	9	8	~	4	3	2	1	0

00: Only spindle tool is displayed.
01: Spindle tool and standby 1 are displayed.
02: Spindle tool and standby 1 and 2 are displayed.
03: Spindle tool and standby 1~3 are displayed.
04: Spindle tool and standby 1~4 are displayed.
05 or more: No spindle tool or standby tool is displayed.

Hexadecimal expression

(3) Magazine tool number display part

The number of displayed magazine tools and the magazine number start value can be changed according to the number-of-magazine parameter and control parameter values.

(a) Number of magazines

Number-of-magazine parameter (R4710): The value can be set in the range of 0 to 80.

(Note) If 0 is set, the magazine number is not displayed. However, the magazine number and magazine tool number guide part is displayed.

(b) Magazine number start value

Control parameter (R4700)

F	E	D			~		7	6	5	4	3	2	1	0

0: The magazine number starts at 1.
1: The magazine number starts at 0.

Example) Magazine number display when the number of magazines is 12.

```
MG TOOL-D MG TOOL-D
1          ~
2          11
~          12
```

The magazine number starts at 1.

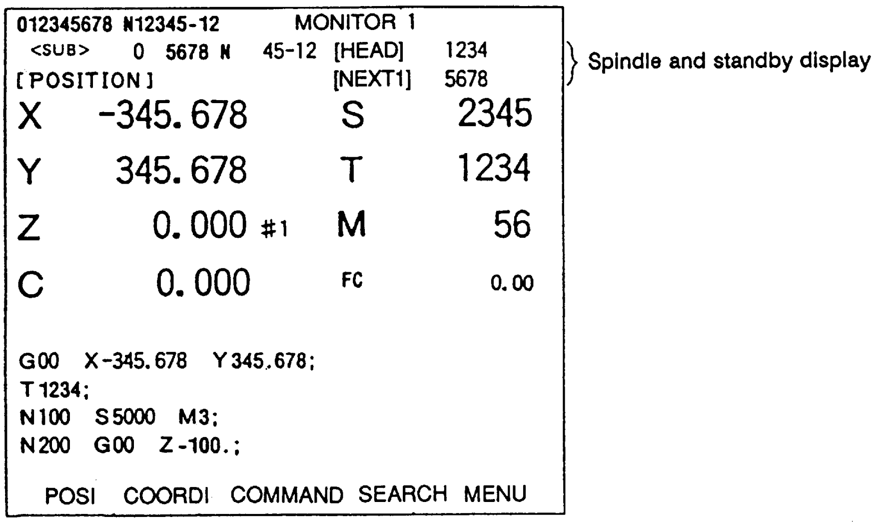
```
MG TOOL-D MG TOOL-D
0          ~
1          10
~          11
```

The magazine number starts at 0.

10.1.13 Display of Spindle Tool and Standby Tool

The tool mounted on the spindle or the tool to be mounted next on the spindle (standby tool) and tool No. in the magazine are set and displayed on the tool registration screen. However, the spindle and standby tool Nos. can also be displayed on the position display screen and tool length measurement screen that are often used. With this, the changes in the magazine pot and spindle tool No. according to the tool selection command or tool change command can be confirmed.

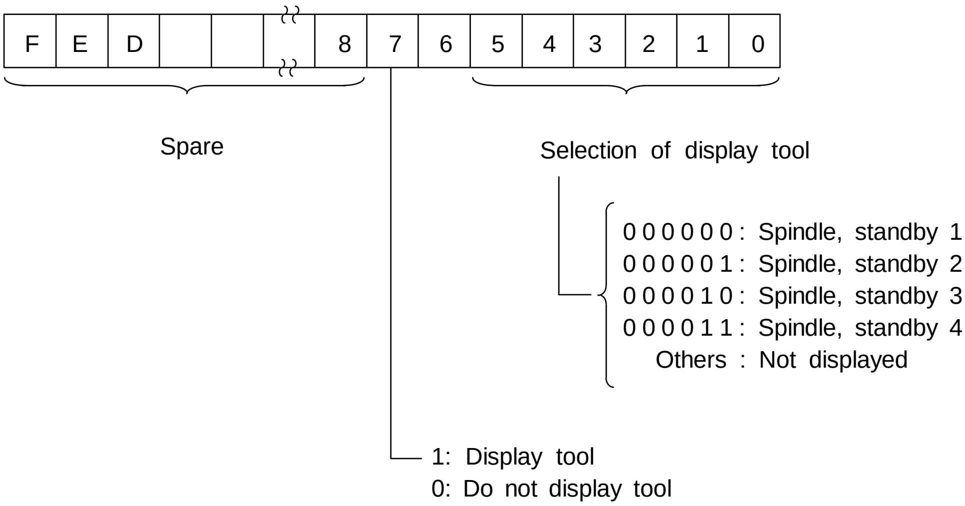
(1) Position display screen



(2) Display tool selection parameter

A maximum of four standby tools can be displayed on the tool registration screen. The No. of the standby tool and the title to be displayed on the POSITION screen and TOOL DATA screen, etc., are selected.

Display tool selection parameter (R4703)



10.2 S.ROT Commands

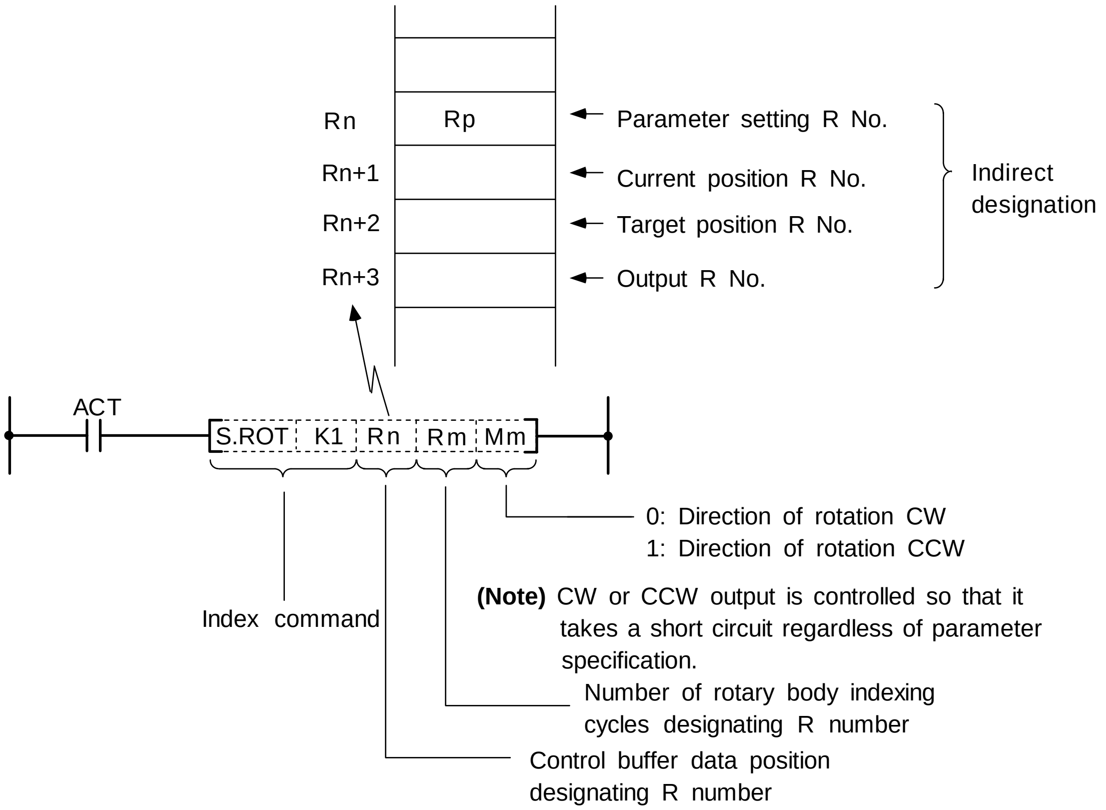
ROT commands are prepared as functions such as rotary body target position, rotation direction and ring counter. The commands can be used to determine the direction of rotation and number of steps with the data resulting from ATC exclusive command tool No. search processing.

10.2.1 Command List

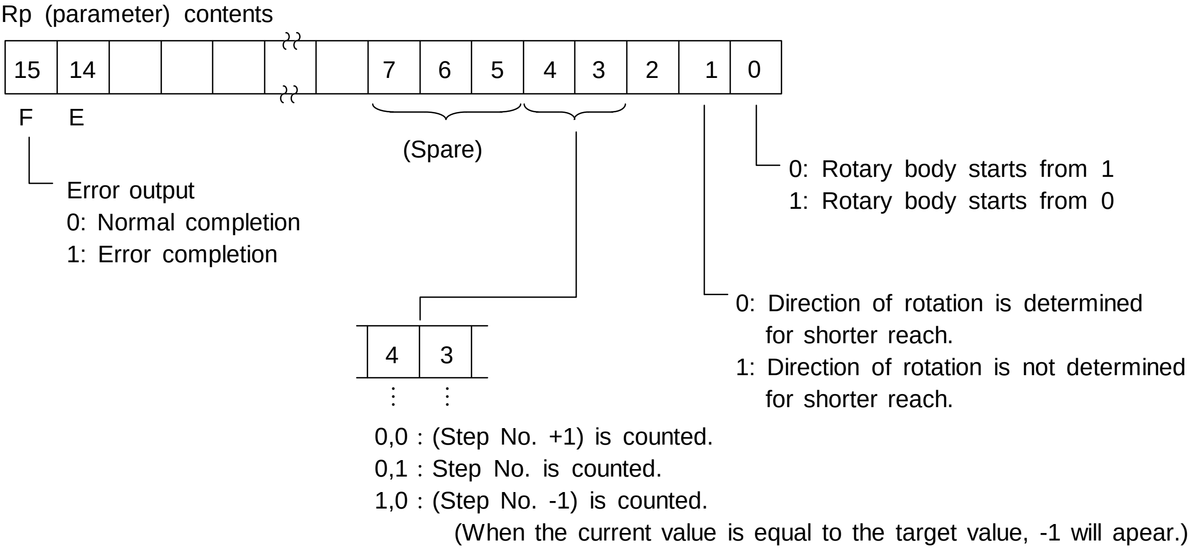
Command	Description
S.ROT K1 Rn Rm Mn	Rotary body indexing
S.ROT K3 Rn Rm Mn	Ring counter

(1) Rotary body indexing

Direction of rotation and number of steps of ATC magazine (or turret) are determined automatically.



Rp (parameter) contents



(Note 1) The Index command is executed after setting R numbers to Rn to Rn+3 and writing data in the file registers (Rs) each corresponding to the R numbers. However, data setting to the parameter (Rp) is done once before execution of the Index command; this is to prevent the error code from being cleared.

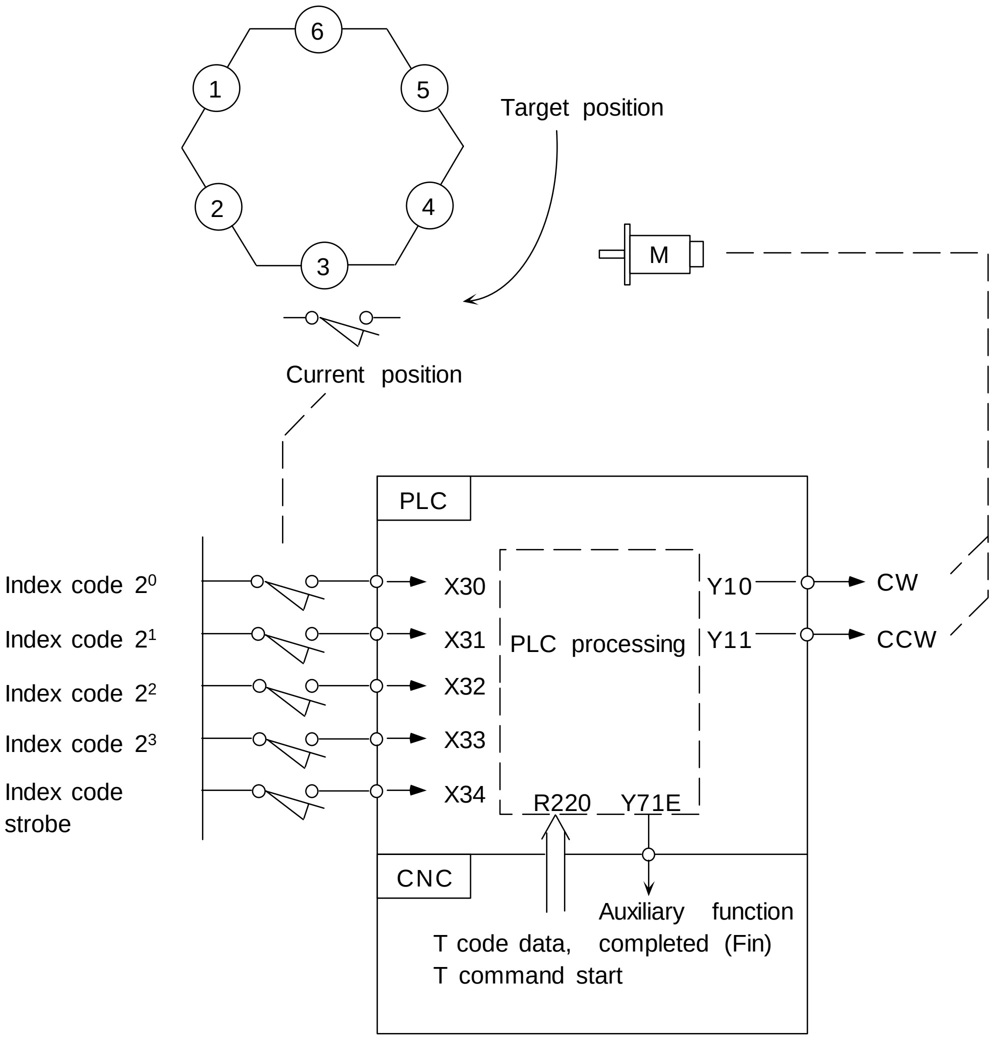
(Note 2) The error code stored in bit F of the parameter (Rp) is not cleared even if the Index command activating signal (ACT) goes off.

(a) Example of rotary body index by ROT K1 instruction

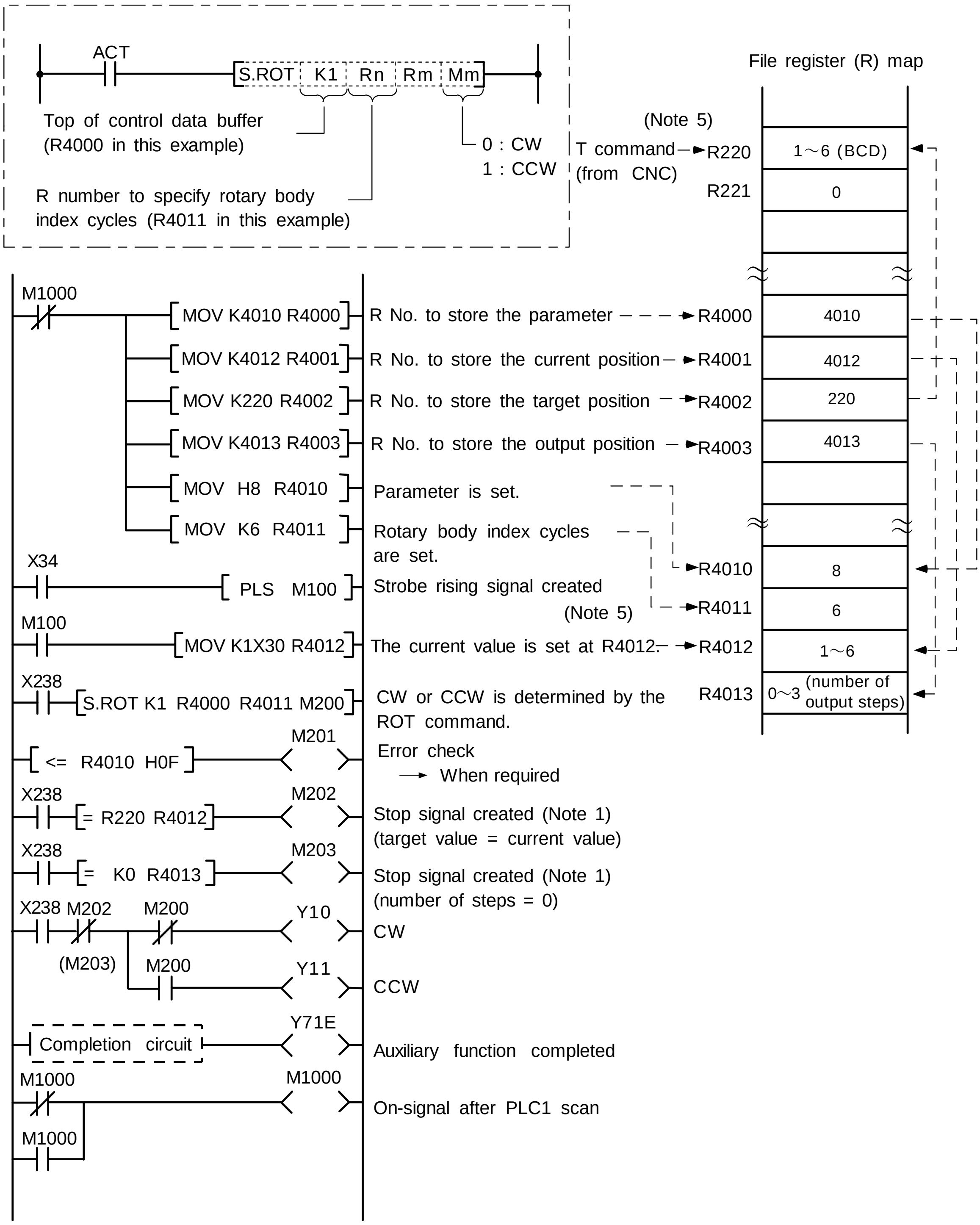
Conditions: (i) The number of rotary body index cycles is 6.

(ii) The target position is specified by a T command.

(Note) Normally the target position must be a binary, but in this example, the number of rotary body index cycles is 1 to 6, and there is no difference between the binary and BCD. Thus, the direct T command output file register R220 (BCD) is used.



In the example of ladder circuit shown below, the rotation direction is determined by the T command and current position data given by the machine, and the rotary body is rotated in that direction until the target position reaches the current position. When indexing is completed, the auxiliary command completion signal is turned on.


ACT
S.ROT K1 Rn Rm Mm
Top of control data buffer (R4000 in this example)
R number to specify rotary body index cycles (R4011 in this example)
0 : CW
1 : CCW
File register (R) map
(Note 5)
T command
(from CNC)
R220
1～6 (BCD)
R221
0
M1000
MOV K4010 R4000
R No. to store the parameter
R4000
4010
MOV K4012 R4001
R No. to store the current position
R4001
4012
MOV K220 R4002
R No. to store the target position
R4002
220
MOV K4013 R4003
R No. to store the output position
R4003
4013
MOV H8 R4010
Parameter is set.
MOV K6 R4011
Rotary body index cycles are set.
X34
PLS M100
Strobe rising signal created
(Note 5)
R4010
8
R4011
6
M100
MOV K1X30 R4012
The current value is set at R4012.
R4012
1～6
X238
S.ROT K1 R4000 R4011 M200
CW or CCW is determined by the ROT command.
R4013
0～3 (number of output steps)
M201
<= R4010 H0F
Error check
When required
X238
M202
= R220 R4012
Stop signal created (Note 1)
(target value = current value)
X238
M203
= K0 R4013
Stop signal created (Note 1)
(number of steps = 0)
X238 M202
M200
Y10
CW
(M203)
M200
Y11
CCW
Y71E
Completion circuit
Auxiliary function completed
M1000
M1000
On-signal after PLC1 scan
M1000

(Note 1) Either M202 or M203 can be used for a stop signal.

(Note 2) The devices (X, Y, and R) are used in this example for no special purpose. Use any device within the available range.

(Note 3) If a number from 1 to 6 has not been specified for current position data (R4012) before the ROT command is activated, an error results.

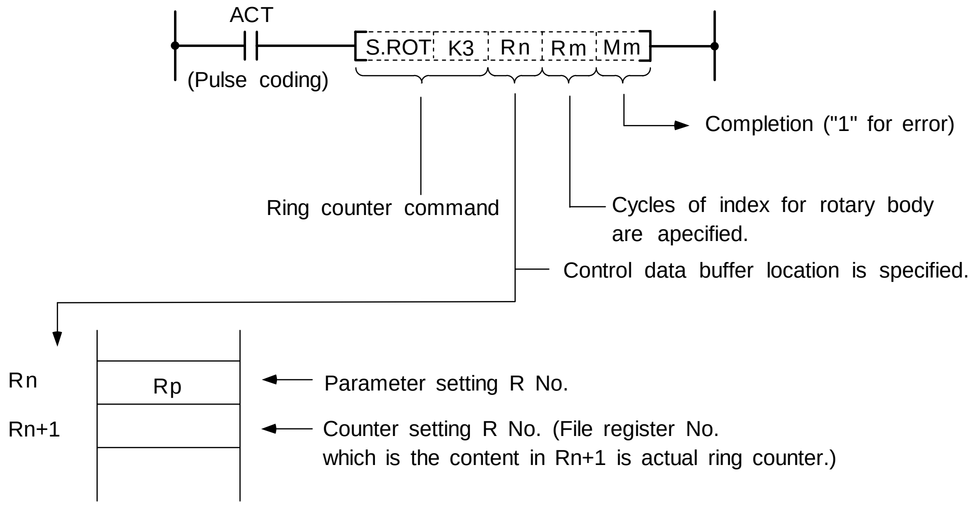
(Note 4) The control parameters (R4010) are specified as follows:
(1) Rotary body starts from 1
(2) Take a short cut.
(3) Calculate the number of steps.

(Note 5) The T command (R220) is output with a BCD code. In this example, the number of rotary body index cycles is 1 to 6, and there is no difference between the binary and BCD. Thus, the contents of R220 are used as they are.
The target position and current value (R220 and R4012 in this example), which are the data to be compared in the S.ROT K1 command must be binaries. (In actual use, the contents of R220 are binary converted.)

(Note 6) The command “<=” in the ladder shown above corresponds to LDBIT command of On-board function.

(2) Ring counter (Up/down counter)

This command is used to control position of rotary body (or turret).



The ring counter is a binary counter; it is used as an up/down counter of "start from 0" or "start from 1" according to the parameter rotary body command.

Rp (parameter) contents

Rp (parameter) contents

15	–	–					–	–	3	2	1	0
F	E											

F: Error output
Command code ... S.ROT Kn is not "1" or "3".

1: Up/down selection
0: Up counter
1: Down counter

0: Rotary body selection
0: Rotary body starts from 1
1: Rotary body starts from 0

(Note 1) The ring counter command is executed after setting R numbers to Rn to Rn+1 and specifying data for the parameter.

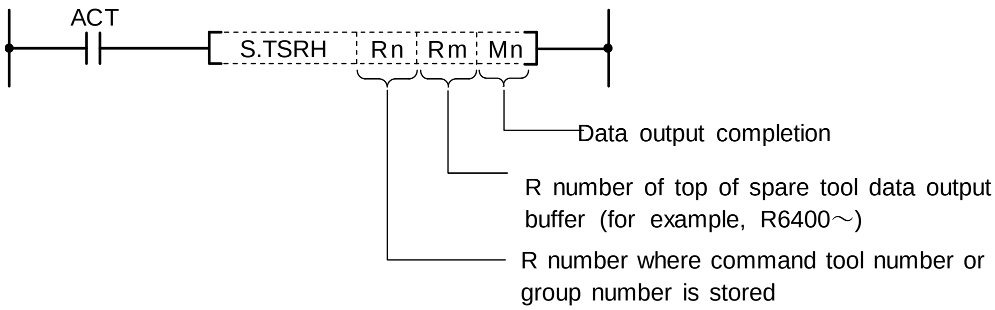
(Note 2) The error code (Mm) of the ring counter command and the error code in bit F of the parameter (Rp) are cleared when the activating signal (ACT) goes off. The activating signal (ACT) of the ring counter command is generally pulsed. This makes it hard for the interface diagnostic and ladder monitor programs to detect an error signal. For debugging, therefore, an error hold circuit is provided after the ring count command to ease error detection.

10.3 Tool Life Management Exclusive Command

(When BASE SPEC parameter #1037 cmdtype is set to 1 or 2.)

The following command is provided only for tool life management. (It is used for the machining centers.)

1. Spare tool selection ... TSRH



10.3.1 Tool Life Management System

(1) Tool life management I (When BASE SPEC parameter #1096 T-Ltyp is set to 1.)
The use time or use count of the spindle tool specified from user PLC (R6350, R6351) is integrated and the tool use state is monitored. Tool data corresponding to the spindle tool is also output. (R6354~R6365)

(2) Tool life management II (When BASE SPEC parameter #1096 T-Ltyp is set to 2.)
Tool life management II is provided by adding the spare tool selection function to tool life management I. Spare tool is selected among group by the spare tool selection command executed by user PLC during tool command, etc., and the tool data of the spare tool is output.
Tool data corresponding to the spindle tool specified from user PLC is output (R6354~R6365) and tool offset corresponding to the spindle tool is made.

10.3.2 Tool Command System

One of the following two can be selected by using a parameter for command tool number (Rm contents) input to the spare tool selection command in tool life management II:

(1) Group number command system (When BASE SPEC parameter #1104 T-Com2 is set to 0.)
The command tool number (Rm contents) input to the spare tool selection command is handled as group number. Spare tool is selected among the tools corresponding to the group number in tool data.

(2) Tool number command system (When BASE SPEC parameter #1104 T-Com2 is set to 1.)
The command tool number (Rm contents) input to the spare tool selection command is handled as a tool number. The group number containing the command tool number is found and spare tool is selected among the group.

10.3.3 Spare Tool Selection System

One of the following two can be selected by using a parameter for the spare tool selection system of the spare tool selection command in tool life management II:

(1) Selection in tool registration order (When BASE SPEC parameter #1105 T-Sel2 is set to 0.)
Spare tool is selected among the used tools of a single group in the registration number order. If used tools do not exist, spare tool is selected among unused tools in the registration number order. If none of used and unused tools exist, spare tool is selected among normal life tools and abnormal tools (the former is assigned higher priority) in the registration number order.

(2) Life equality selection (When BASE SPEC parameter #1105 T-Sel2 is set to 1.)
Tool whose remaining life is the longest is selected among the used and unused tools of a single group. If more than one tool has the same remaining life, it is selected in the registration number order. If none of used and unused tools exist, spare tool is selected among normal life tools and abnormal tools (the former is assigned higher priority) in the registration number order.

10.3.4 Interface

(1) User PLC → Controller

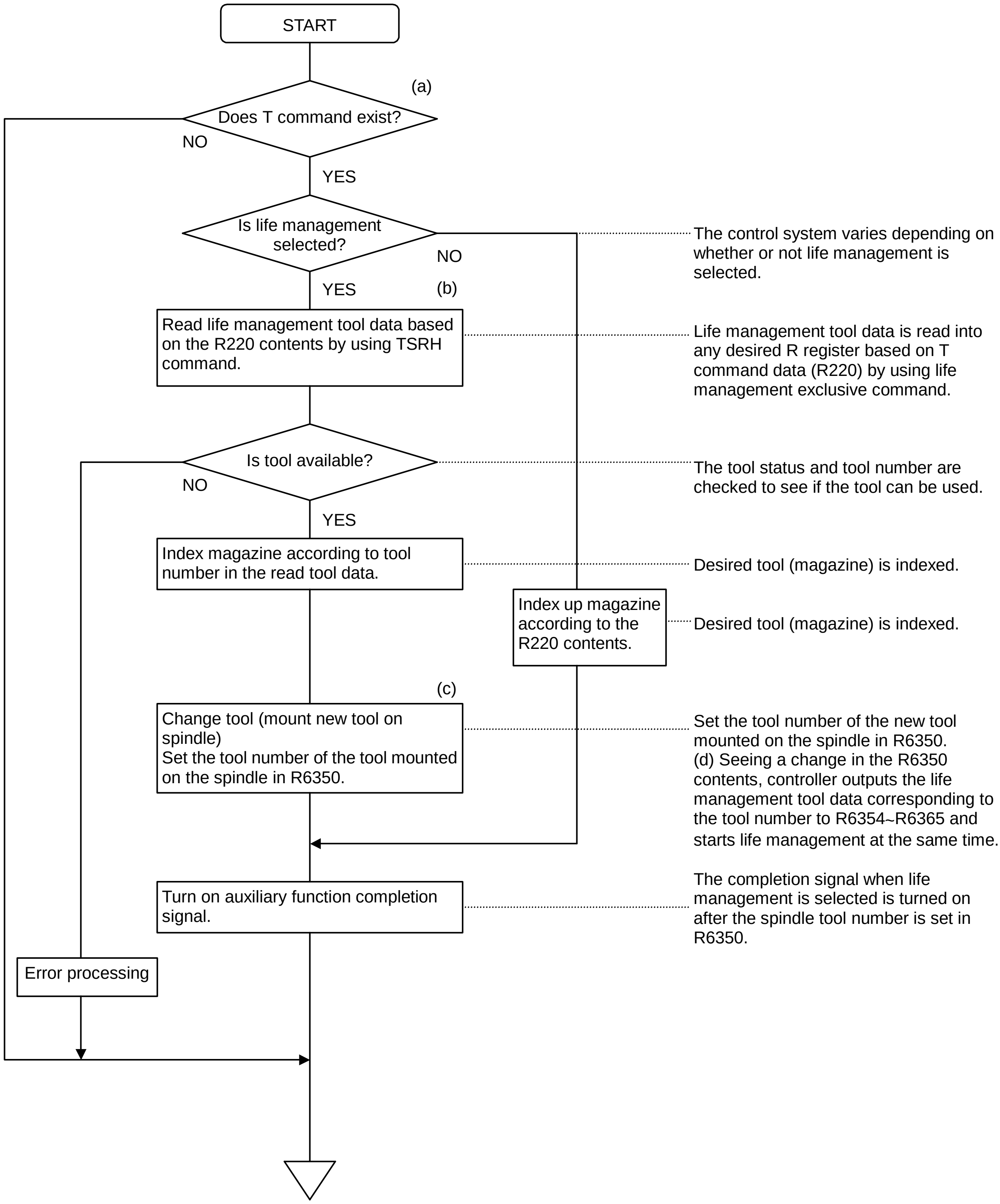
Device name	Signal name	Explanation
Y75A	Auxiliary function locking signal	While this signal is input, tool life management is not made.
Y788	Tool error 1 signal	This signal indicates tool error state 1. When controller inputs the signal it changes the status in spindle tool data to 3. (Unused tools or used tools are changed to toll error state 1.)
Y789	Tool error 2 signal	This signal indicates tool error state 2. When controller inputs the signal, it changes the status in spindle tool data to 4. (Unused tools or used tools are changed to toll error state 2.)
Y78A	Usage data counter validity signal	If this signal is not input, the usage data is not counted.
Y78B	Tool life management input signal	If this signal is input to controller and the tool life management output signal is output to PLC, tool life management is made.

(2) Controller → User PLC

Device name	Signal name	Explanation
X62B	Tool life management output signal	The controller outputs this signal to PLC while the tool life management function is selected. (When BASE SPEC parameter #1103 T-Life is set to 1.)

10.3.5 User PLC Processing When the Tool Life Management Function Is Selected

A PLC processing example when tool change is made by the T command is given below:



(1) Procedure when tool command is executed

(a) Tool life management I

1) When tool command (T command) is given, the controller outputs T code data and start signal (TF). **(Note)** The T code data (BCD) is binary converted and then used.
2) The user PLC checks the tool command. If life management is required, the user PLC executes the spare tool selection command.
3) The spare tool selection command outputs the tool data of the tool corresponding to the specified tool number.
4) The user PLC decides whether or not the tool can be used according to the status in the output tool data, and selects command tool or performs alarm processing.

(Note) If -1 is set in the group number in the output tool data, the tool data is invalid. At the time, the specified tool number is output to the tool number in the output tool data as it is.

(b) Tool life management II

1) When tool command (T command) is given, the controller outputs T code data and start signal (TF). **(Note)** The T code data (BCD) is binary converted and then used.
2) The user PLC checks the tool command. If life management is required, the user PLC executes the spare tool selection command.
3) The spare tool selection command selects the spare tool corresponding to the specified number (group number, tool number) and outputs the tool data of the spare tool.
4) The user PLC decides whether or not the tool can be used according to the status in the output tool data, and selects command tool or performs alarm processing.

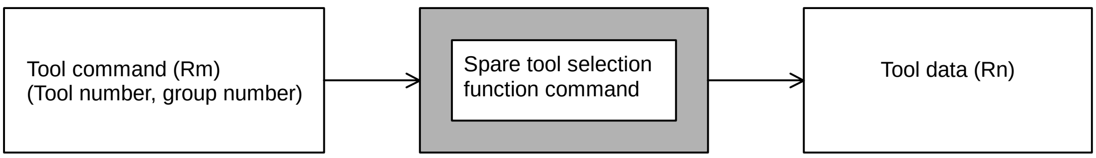
(Note) If -1 is set in the group number in the output tool data, the tool data is invalid. At the time, the specified tool number is output to the tool number in the output tool data as it is.

(2) Procedure when spindle tool is changed

1) When spindle tool is changed during the spindle tool change command (M06), etc., the user PLC specifies the tool number of the spindle tool (R6350~R6351).
The controller outputs the spindle tool data corresponding to the tool number of the spindle tool every user PLC main cycle (R6354~R6365).
2) The controller integrates the use time or use count of the spindle tool based on the spindle tool data in the tool data file.
In tool life management II, it also executes tool offset corresponding to the spindle tool.

(Note) If -1 is set in the group number in the output spindle tool data, the spindle tool data is invalid. At the time, the specified tool number (R6350~R6351) is output to the tool number in the output spindle tool data as it is. The controller does not integrate the usage time or usage count of the spindle tool or make tool offset.

<When tool command is executed>

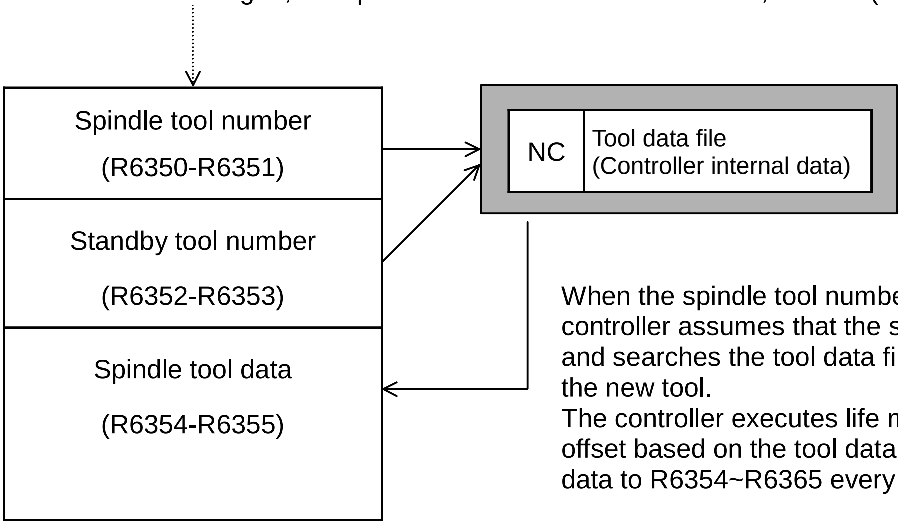


In tool life management I, tool number is only specified and spare tool is selected.

Tool is selected according to tool number in tool data. (User PLC)

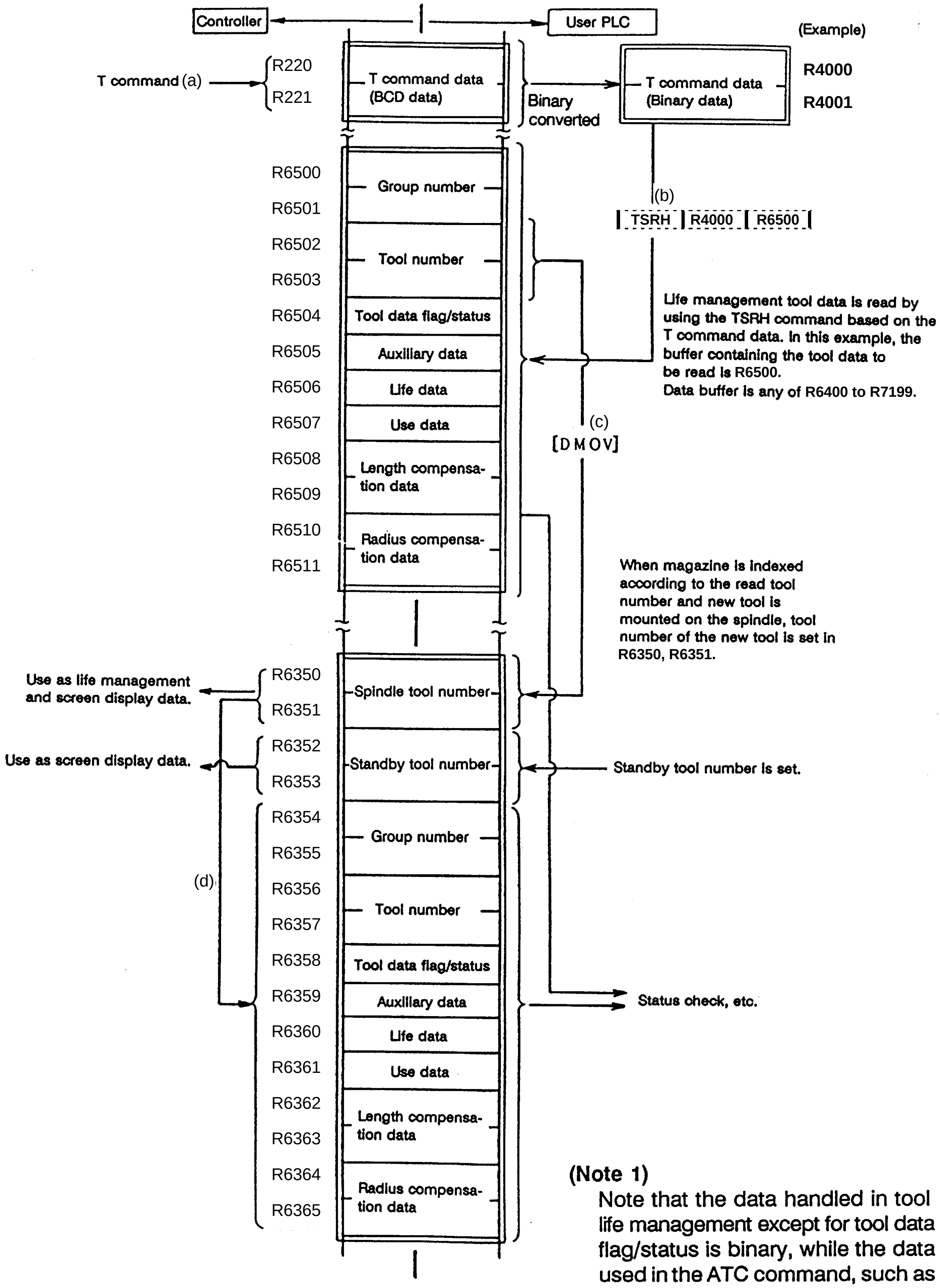
<When tool is changed>

When tool is changed, the spindle tool number is set in R6350, R6351. (User PLC)



When the spindle tool number changes, the controller assumes that the spindle tool is changed, and searches the tool data file for the tool data of the new tool.
The controller executes life management and tool offset based on the tool data. It also outputs the tool data to R6354~R6365 every user PLC main cycle.

(3) Tool data flow



(Note 1)
Note that the data handled in tool life management except for tool data flag/status is binary, while the data used in the ATC command, such as search data and magazine tool number, is BCD.

(4) Tool data

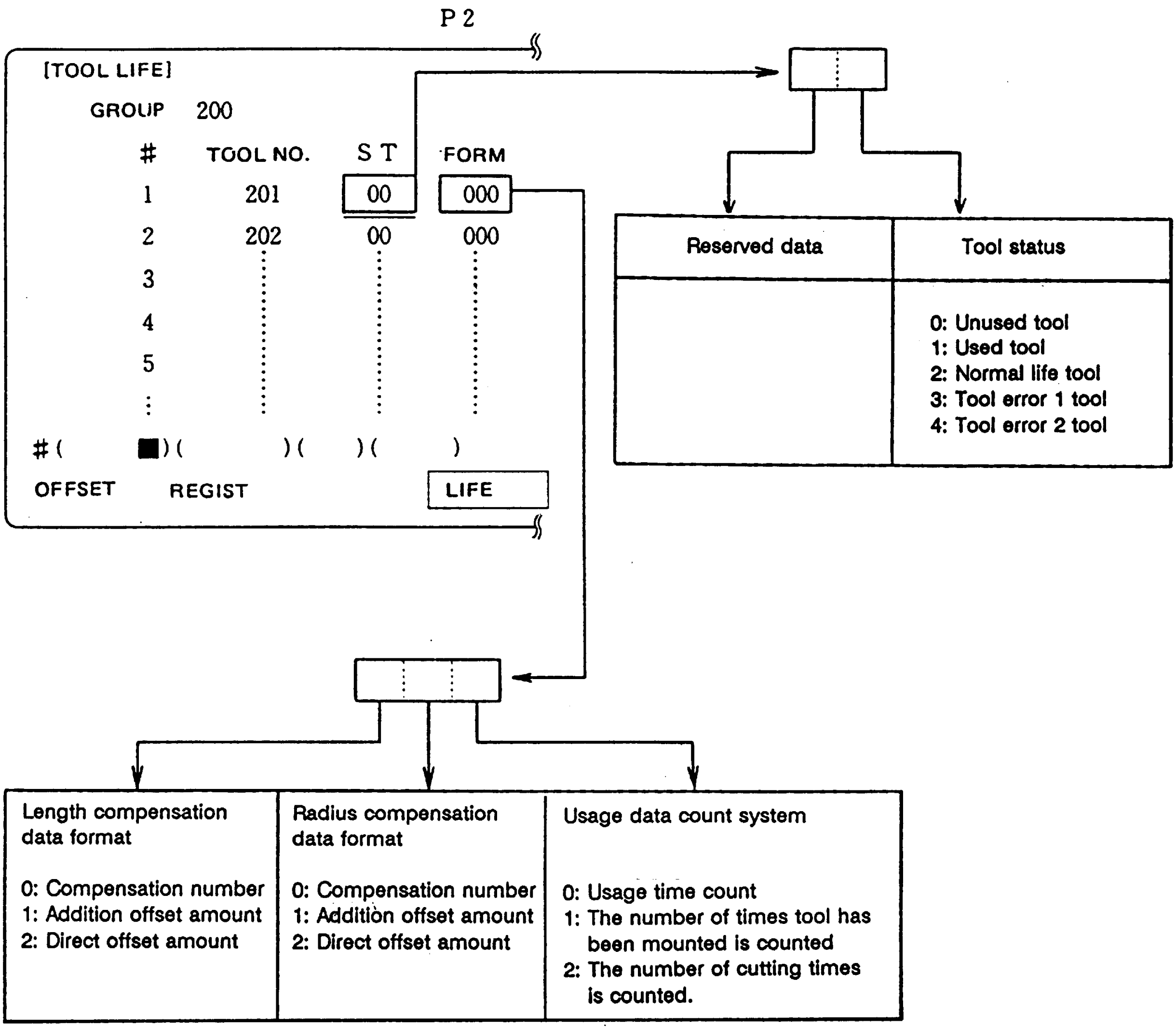
The tool data is tool management data such as the group number, tool number, and tool status. The details are given below:

Tool data name	Explanation	Data range
Group number	Number to manage tools of the same type (form and dimensions) in a group. The tools assigned the same group number are assumed to be spare tools.	1 - 99999999
Tool number	Number unique to each tool actually output during tool command execution	1 - 99999999
Tool data flag	Parameter of use data count system, length compensation system, radius compensation system, etc.	
Tool status	The tool state is indicated.	0 - FF (H)
Auxiliary data	Reserved data	0 - 65535
Tool life data	Life time or life count for each tool. (If 0 is set, infinity is assumed to be specified.)	0 - 4000 (minutes) 0 - 9999 (times)
Tool usage data	Usage time or usage count for each tool.	0 - 4000 (minutes) 0 - 9999 (times)
Tool length compensation data	Length compensation data set in any format of compensation number, direct offset amount, and addition offset amount.	Compensation numbers 1 - 400 Direct offset amount ±99999.999 Addition offset amount ±99999.999
Tool radius compensation data	Radius compensation data set in any format of compensation number, direct offset amount, and addition offset amount.	Compensation numbers 1 - 400 Direct offset amount ±99999.999 Addition offset amount ±99999.999

(5) Tool data flag and tool status

The tool data flag and tool status contents are shown below:

(a) Correspondence with tool life management data screen



(b) Tool data flag ..... Bits 0~7 of file register Rn (such as R6358)

bit	Explanation	
bit 0 bit 1	Length compensation data format (spare tool compensation system)	0: Compensation number 1: Addition offset amount 2: Direct offset amount
bit 2 bit 3	Radius compensation data format (spare tool compensation system)	0: Compensation number 1: Addition offset amount 2: Direct offset amount
bit 4 bit 5	Usage data count system	0: Usage time (minutes) 1: Number of times tool has been mounted 2: Number of cutting times
bit 6		
bit 7		

1) Spare tool compensation system

Tool compensation corresponding to the spindle tool can be made in tool life management II.

One of the following three types of length and compensation can be selected by setting tool data:

i) Compensation umber system (0 is set on the tool data registration screen.)
Compensation data in tool data is handled as the compensation number. It is replaced with the compensation number given in a work program and compensation is executed.

ii) Addition compensation system (1 is set on the tool data registration screen.)
Compensation data in tool data is handled as addition offset amount. It is added to the offset amount indicated by the compensation number given in a work program and compensation is executed.

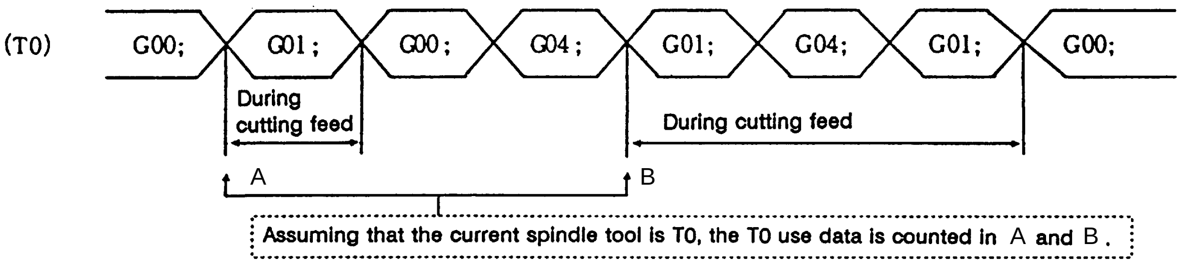
iii) Direct compensation system (2 is set on the tool data registration screen.)
Compensation data in tool data is handled as direct offset amount. It is replaced with the offset amount indicated by the compensation number given in a work program and compensation is executed.

2) Usage data count system

i) Usage time count
For usage data, the execution time of cutting feed (such as G01, G02, or G03) is counted in 3.75-s units. However, the life data and usage data are displayed in minute units on the tool data registration screen.

ii) Number of times tool has been mounted is counted
When tool is used as spindle tool in tool change, etc., usage data is counted. However, if cutting feed (G01, G02, or G03) is not executed after tool is used as spindle tool, usage data is not counted.

iii) Number of cutting times is counted
Usage data is counted when a change is made from rapid traverse feed (such as G00) command to cutting feed (such as G01, G02, or G03) command as shown below. However rapid traverse or cutting feed command with no movement becomes invalid.
Even if a command other than the rapid traverse command appears between cutting feed commands, usage data is not counted.



Caution:

When none of the tool life management input signal and use data count validity signal are input or during machine lock, auxiliary function lock, dry run, or single block, usage data is not counted.

- The usage data is not counted when the life data is 0.
- Life management is executed even in the MDI operation mode.
- The usage data is not counted even when the status is 2 or more (normal life, error tool 1, error tool 2).

(c) Tool status ..... Bits 8~F of file register Rn (such as R6358)

bit	Explanation
bit 8	Tool status (numeric data 0~4) 0: Unused tool 1: Used tool 2: Normal life tool 3: Tool error 1 tool 4: Tool error 2 tool
bit 9	
bit A	
bit B	
bit C	(Reserved)
bit D	
bit E	
bit F	

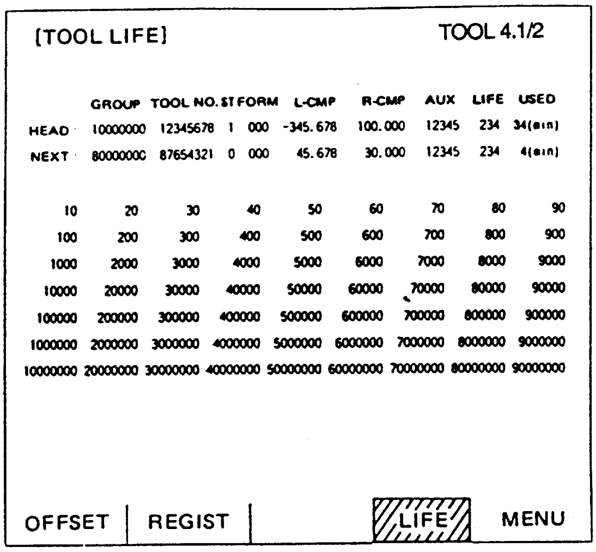
(d) Tool status contents

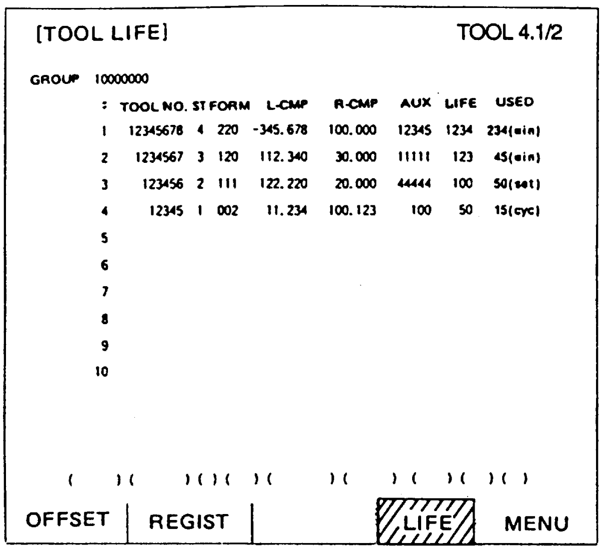
When the tool status number is 0 or 1, NC assumes the tool to be available.

Tool status number	Explanation
0	Indicates unused tool. Normally, this state is set when tool is replaced with a new tool.
1	Indicates used tool. When actual cutting is started, this state is set.
2	Indicates normal life tool. When use data exceeds life data, this state is set.
3	Indicates tool error 1 tool. When controller inputs the tool error 1 signal, this state is set.
4	Indicates tool error 2 tool. When controller inputs the tool error 2 signal, this state is set.

10.3.6 Examples of Tool Life Management Screen

Tool life management screen examples are given below.
For operation, refer to the Operation Manual.



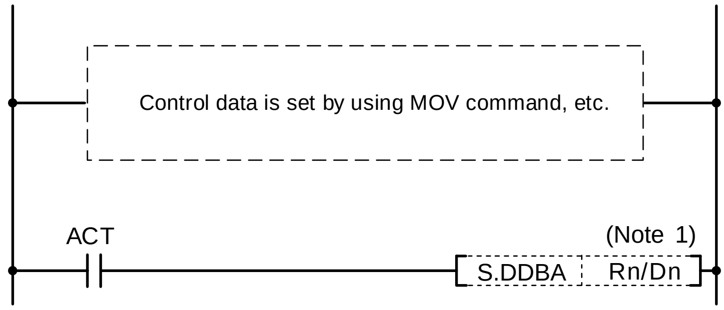


Screen example

10.4 DDB (Direct Data Bus) ... Asynchronous DDB

The DDB function is used for PLC to directly read/write various pieces of data that controller has. PLC can read specified data into buffer or write specified data into controller by storing necessary information for read/write and calling the DDB function. Generally, data is read or written for each data piece; data concerning the control axes is processed in batch as many as the specified number of axes.

10.4.1 Basic Format of Command



(Note 1) File registers (Rn) and data registers (Dn) to which the user is accessible can be used as the asynchronous DDB control data buffer. The file registers (R) to which the user is accessible are R4000 through R4499 (not backed up) and R6400 through R7199 (backed up).

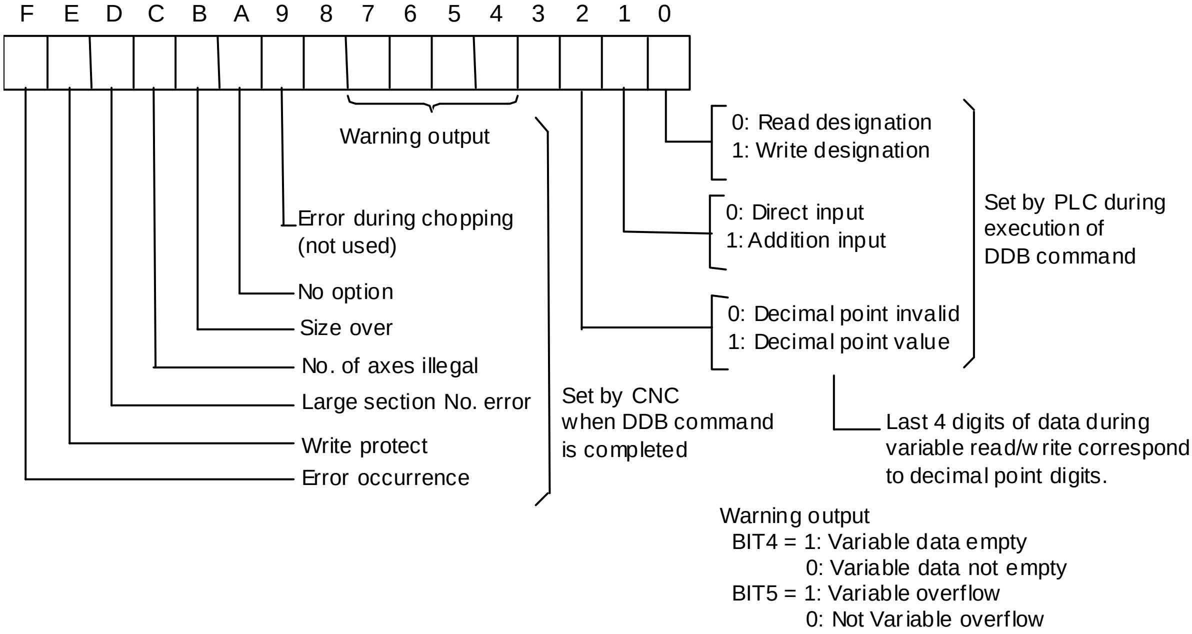
10.4.2 Basic Format of Control Data

Rn(Dn)	Control signal
Rn+1(Dn+1)	Large section No.
Rn+2(Dn+2)	Sub section No.
Rn+4(Dn+4)	Data size
Rn+5(Dn+5)	Read/write designated axis, part system designation
Rn+6(Dn+6)	Read/write data (For 1st axis)
Rn+8(Dn+8)	(For 2nd axis)
Rn+10(Dn+10)	(For 3rd axis)
Rn+12(Dn+12)	(For 4th axis)

(Note 1) Part system designation is used for multi-part system specifications.

(Note 2) If only the 3rd axis is designated for example, the section for the 3rd axis will be read/write data.

(1) Control signals (Rn), (Dn)

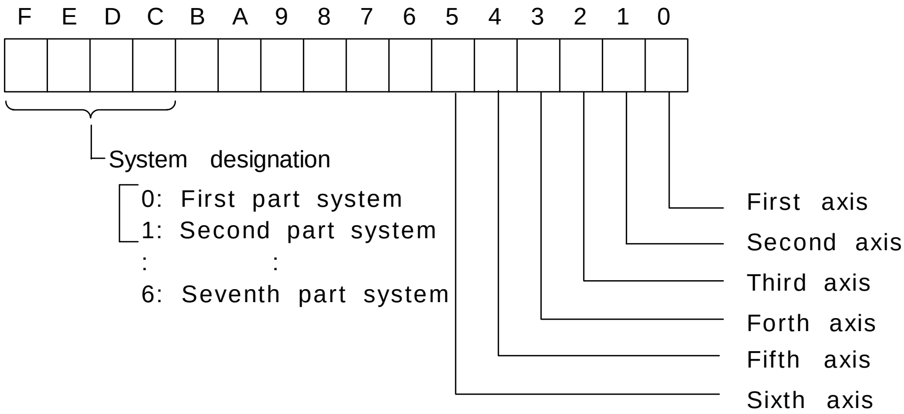


(2) Large section number (Rn+1), (Dn+1)
Specify the large section number of the data to be read/written in binary form.

(3) Sub-section number (Rn+2, Rn+3), (Dn+2, Dn+3)
(LOW) (HIGH) (LOW) (HIGH)
Specify the sub-section number of the data to be read/written in binary form.

(4) Data size (Rn+4), (Dn+4)
Specify the size of the data to be read/written in binary form.
1: One byte
2: Two bytes
4: Four bytes
If any value other than 1, 2, or 4 is specified, the invalid data size alarm will occur.

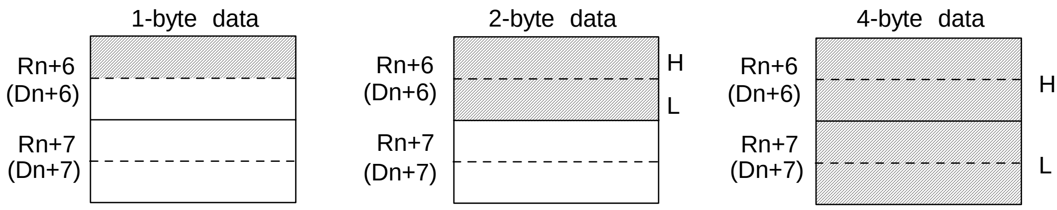
(5) Read/write specifications axis (Rn+5), (Dn+5)
Specify the axis to read or write data for each axis classified by major classification numbers.



If axis specification is not made or exceeds the maximum control axis when axis data is read or written, the invalid axis number alarm will occur.

(6) Read/write data (Rn+6, Rn+7), (Dn+6, Dn+7)
(LOW) (HIGH) (LOW) (HIGH)

When data is read, the controller outputs data specified by PLC. When data is written, PLC sets the data to be written.



The effective portion of data varies depending on the data size. (Hatched portion)
When read is specified the sign of 1-byte or 2-byte is extended to four bytes.

The main data that can be referenced by using the DDB function is listed below.

Specification item	Contents	Read	Write	Remarks
Asynchronous	Current position in workpiece coordinate, machine coordinate system, length, radius offset amount	○ ○	— ○	
	Parameters (Maximum rotation speed of spindle, second, third, and fourth reference position coordinates, stored stroke limit, coordinate system offset, etc.)	○	○	
	User macro variables	○	○	
	Modal data of G code, etc.	○	—	
	Controller alarm number	○	—	
	Compensation function (External workpiece coordinate system input, external tool compensation input)	—	○	
Synchronous	External search	—	—	
	PLC axis control, etc.	—	—	

Caution:

The DDBA command is issued after setting necessary data such as control signal and large and sub-classification numbers to the buffer (Rn or Dn). A read or write of the control signal is specified only once before execution of the DDBA command to prevent error codes stored in high-order bits by the CNC from being erased.

10.5 External Search

10.5.1 Function

When PLC specifies the program number, sequence number, and block number of a work program for the controller, the external search function searches memory for the program number, sequence number, and block number.
Note that the external search from PLC high-speed processing is not possible, however.

10.5.2 Interface

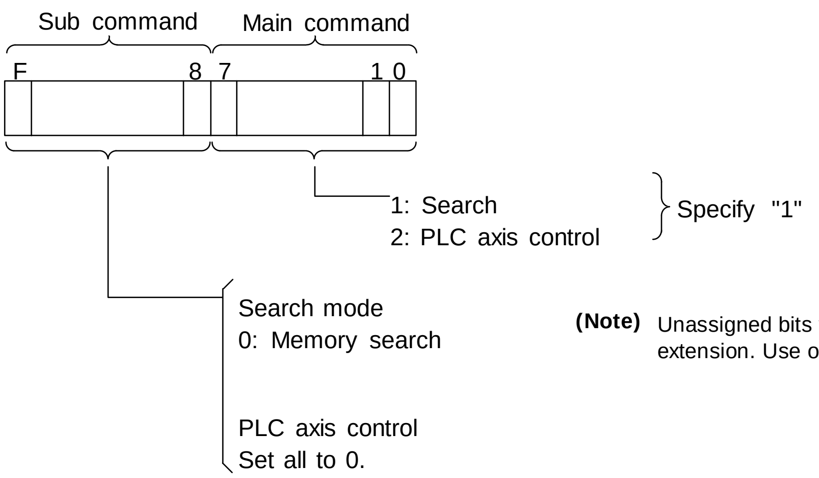
PLC sets data except the status.

Rn+0	Two bytes	Command
1	Two bytes	Status
2, 3	Four bytes	Program No.
4, 5	Four bytes	Sequence No.
6, 7	Four bytes	Block No.
8	Two bytes	Part system designation

(Note 1) File register (Rn) that can be used by the user is used for the control data buffer. Data register (Dn) cannot be used.

(Note 2) Part system designation is used for multi-system specifications.

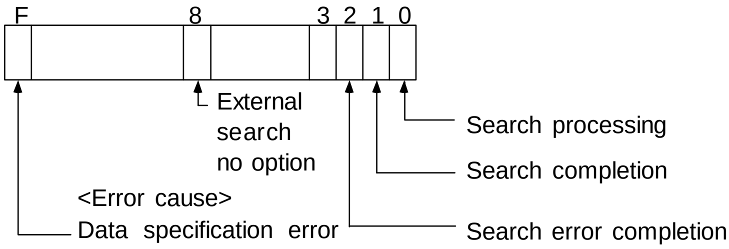
(1) Command



(Note) Unassigned bits will be used for later function extension. Use only bits shown here.

(2) Status

The search state is indicated.
The status is set by the controller and is used by PLC for completion check, etc.



The status is cleared by the controller when the search start instruction execution condition is off.

(3) Program number

Specify the program number to be searched in binary form in the range of 1 to 99999999 (eight digits).
Specify 0 to search for the sequence number of the current program selected.
If a number other than 0~99999999 is specified, a data specification error will occur.

(4) Sequence number

Specify the sequence number to be searched in binary form in the range of 1 to 99999 (five digits).
Specify 0 to search for the head of the specified program number.
If a number other than 0~99999 is specified, a data specification error will occur.

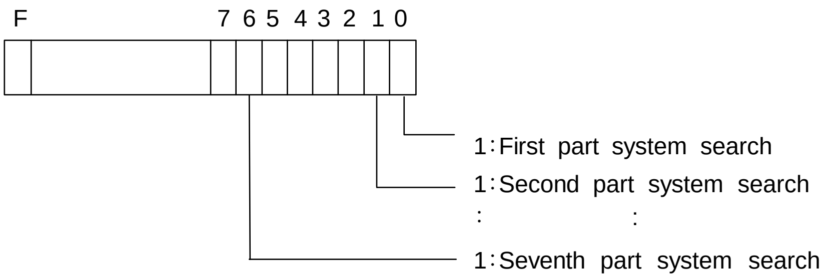
(5) Block number

Specify the block number to be searched in binary form in the range of 0 to 99 (two digits).
If a number other than 0~99 is specified, a data specified error will occur.

Program No.	Sequence No.	Search
Specified	Specified	Memory or tape is searched for the specified sequence number of the specified program.
Specified	Not specified (=0)	Memory or tape is searched for the top of the specified program.
Not specified (=0)	Specified	Memory or tape is searched for the specified sequence number of the current program selected.
Not specified (=0)	Not specified (=0)	Error (no specification)

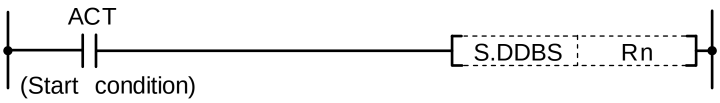
(6) Part system specification

Specify the part system to be searched. If no part system specification is made, only the first part system is searched.



10.5.3 Search Start Instruction

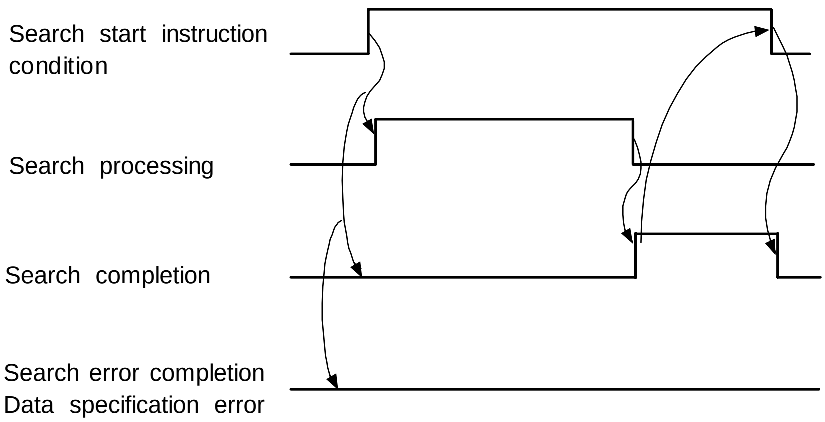
After interface data between the controller and PLC is prepared, search is started by using the following instruction:



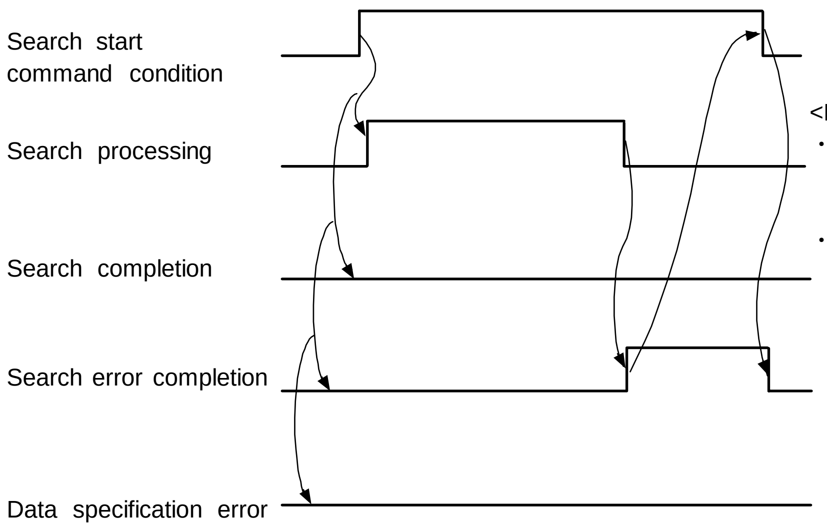
(Rn is any file register that can be used by the user.)

10.5.4 Timing Charts and Error Causes

(1) Normal completion

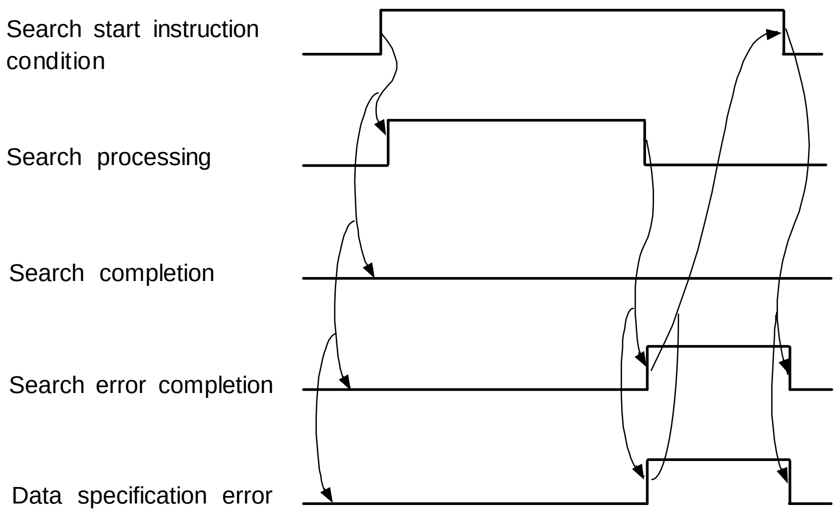


(2) Search error completion



<Error cause>
- The specified program number or sequence number is not found.
- The NC operation state is not reset state any other search-impossible state.

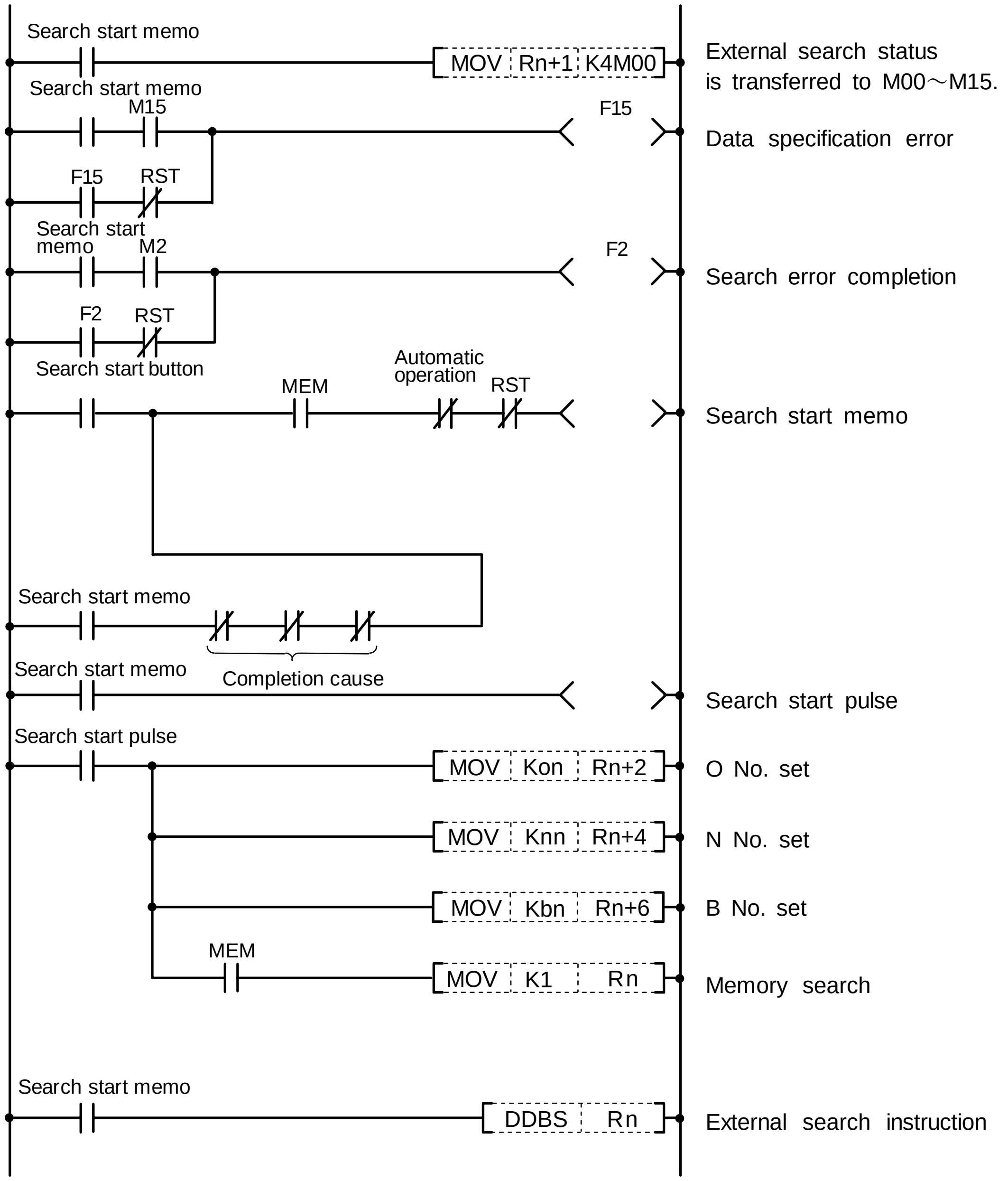
(3) Search error completion (Data specification error)



<Error cause>

- Program number and sequence number are not specified.
- Program number or sequence number is specified beyond the range.

10.5.5 Sequence Program Example



RST: Reset signal (reset button, output during reset, etc.)

11. PLC Help Function

To help the user PLC, an exclusive interface is provided between the user PLC and controller or PLC basic. The function and interface are explained below.

PLC help function examples:
· Alarm message display
· Operator message display
· PLC switches
· Key operation by user PLC
· Load meter display
· External machine coordinate system compensation
· User PLC version display

11.1 Alarm Message Display

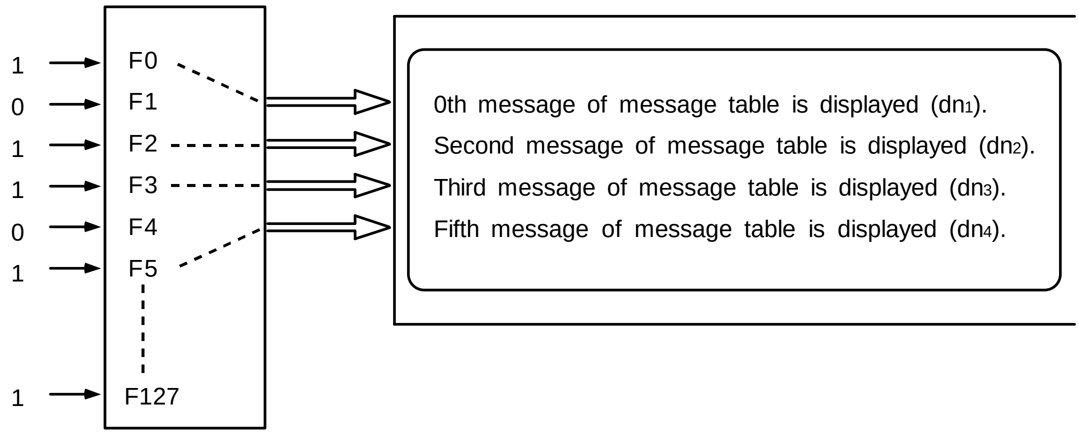
The contents of an alarm that occurred during sequence (user PLC) processing can be displayed on the setting and display unit.
Up to four alarm messages can be displayed at a time on the alarm diagnosis screen. The maximum length of a message is 32 characters.

11.1.1 Interface

The alarm message display interface is available in the two types: F type in which temporary memory F is used for message display request and R type in which file register (R) is used for message display request. Either type is selected by using a parameter.

(1) F type interface

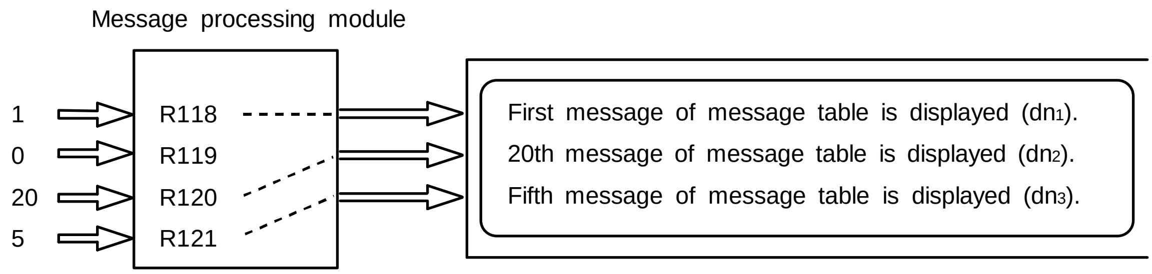
This interface applies to 128 points of temporary memory F0~F127.
If temporary memory F is used as the alarm interface, do not use it for another purpose.



The highest priority is assigned to the F0 signal. The message corresponding to Fn set to 1 is fetched from the message table and displayed in order starting at F0. If no messages are prepared or Fm greater than the number of prepared messages is set to 1, the message "USER PC ERROR m" is displayed.

(2) R type interface

This interface applies to file registers R118~R121. The numeric value (binary) contained in each of the R registers indicates the position of the message to be displayed in the message table.
The message is cleared by setting the R register to 0.

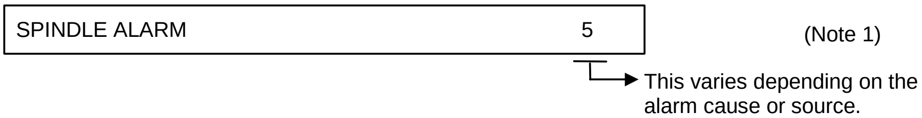


The messages are displayed starting at the message corresponding to R118 from top to bottom.
Since message display is cleared by setting the R register to 0, number 0 in the table message cannot be used in the R mode.
If greater value than the number of prepared messages, m is set in the R register, the message "USER PC ERROR m" is displayed.

(3) Alarm classification display

Classification No. can be displayed following the message to be displayed regardless of the F or R type. (Dn1~Dn4 in the figure)
For example, one typical alarm message is prepared and classification No. can be used to indicate the alarm source or cause.

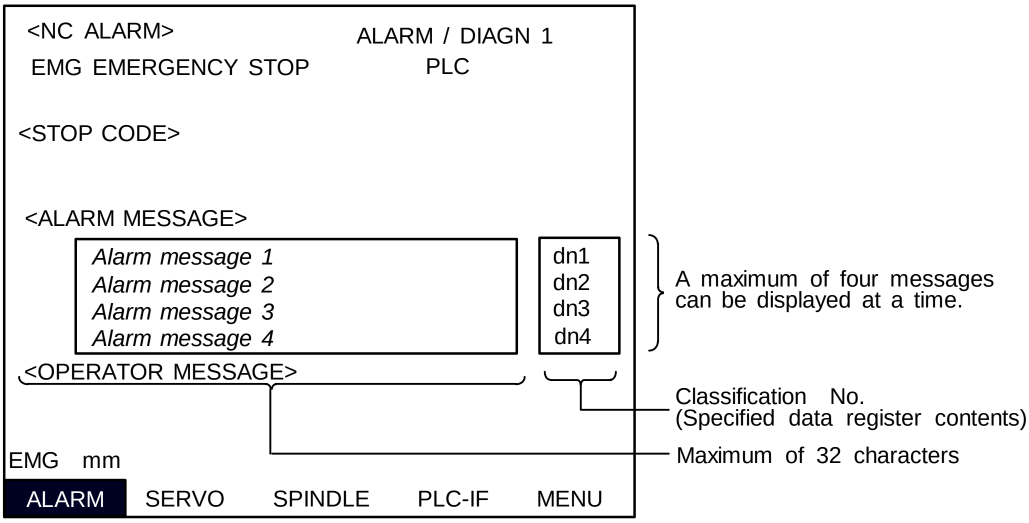
Example) When spindle alarm occurs, the message "SPINDLE ALARM" is displayed and the alarm source or cause is indicated by the classification No.



For the classification No., the contents of each data register specified in alarm message preparation are displayed. Data register D0 cannot be specified.

(Note 1) The display of the classification No. by cause is updated when an alarm message display changes. It is not updated if only the contents of the specified data register (Dn1 to Dn4) change. If the contents of the specified data register are 0, no classification Nos. are displayed.

Display example



11.1.2 Message Creation

(1) Alarm message type

Create messages by using PLC development software (GX-Developer).

Set the number of characters for one message and the number of messages to be prepared, then enter message data through the keyboard.
The maximum length of an alarm message is 32 characters.
A maximum of 512 alarm messages can be prepared. For details, refer to the section "1.4 Creating PLC Message Data".

11.1.3 F or R Type Selection Parameter

Set the parameter on the bit selection screen of PLC parameter (setup parameter).

[Bit selection parameter screen]

```
                         7   6   5   4   3   2   1   0   <— Bit
# (6450)   Data   ( 0   0   0   0   0   0   *   1 )
```

Use number 6450.

Bit 0:
0: Alarm message invalid.
1: Alarm message valid.

Bit 1:
0: F type interface
1: R type interface

[Reference] #6450 corresponds to the high-order byte of the file register R4624.

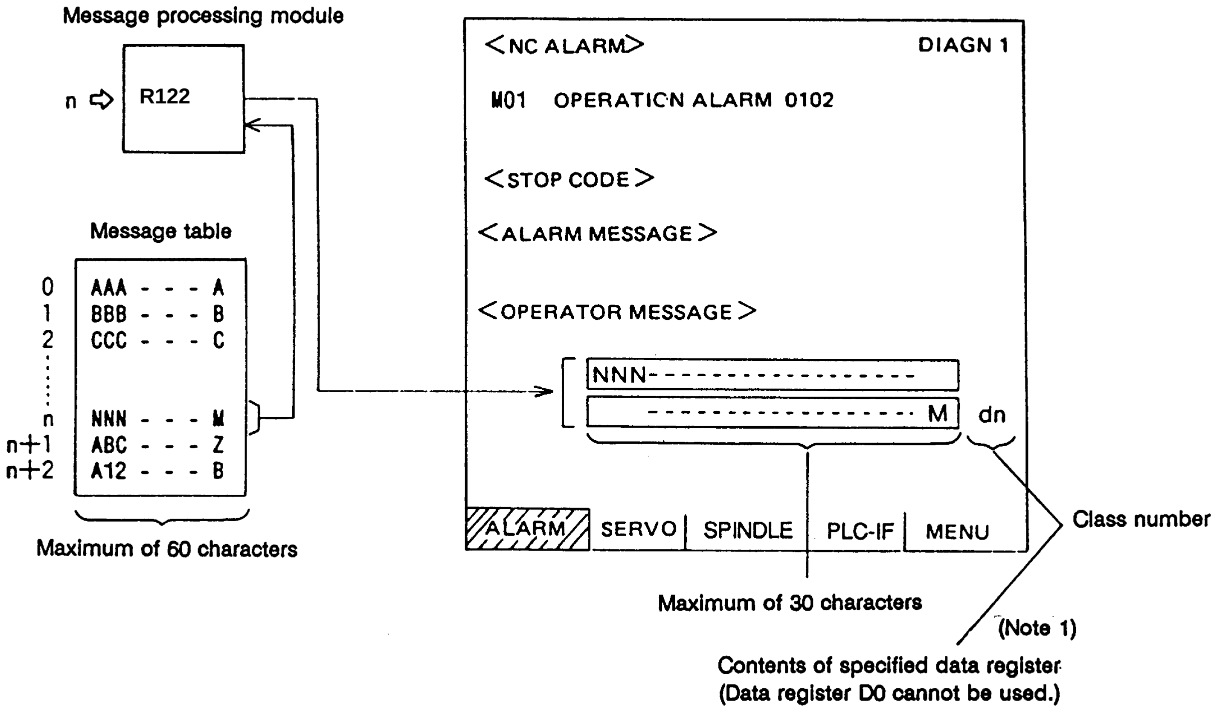
11.2 Operator Message Display

When a condition to inform the operator of a message occurs, an operator message can be displayed independently of an alarm message.
A maximum of 60 characters can be displayed for the operator message on the alarm diagnosis screen. One operator message can be displayed at a time.

11.2.1 Interface

An operator message is displayed by setting the number of the operator message table to be displayed in file register R122. It is cleared by setting R122 to 0. Thus, number 0 of the operator message table cannot be displayed.

Display example



As with alarm messages, the contents of the data register specified for the class number display in operator message preparation are also displayed.

(Note 1) The class number display is updated when the contents of file register R122 change. It is not updated if only the contents of the specified data register (Dn) change.
To change the class number display only, the contents of R122 must be cleared to 0. If the contents of the specified data register are 0, no class numbers are displayed.

11.2.2 Operator Message Preparation

Create messages by using PLC development software (GX Developer).

According to the description format, set the number of characters for one message and the number of messages to be prepared, then prepare message data.
The maximum length of an operator message is 60 characters. A maximum of 512 operator messages can be prepared. For details, refer to the section "1.4 Creating PLC-related Data".
However, the number of operator messages may be limited depending on the available memory capacity. For details, refer to the section "1.4 Creating PLC Message Data".

11.2.3 Operator Message Display Validity Parameter

The parameter is set on the machine manufacturer parameter bit selection screen.

```
                  7  6  5  4  3  2  1  0  ←bit
#(6450) Data（ 0  0  0  0  0  0  0  0 ）
    |                          └── 0: Operator message display invalid.
    |                              1: Operator message display valid.
    └ Use number 6450.
```

(Reference) #6450 corresponds to the high-order byte of file register R4624.

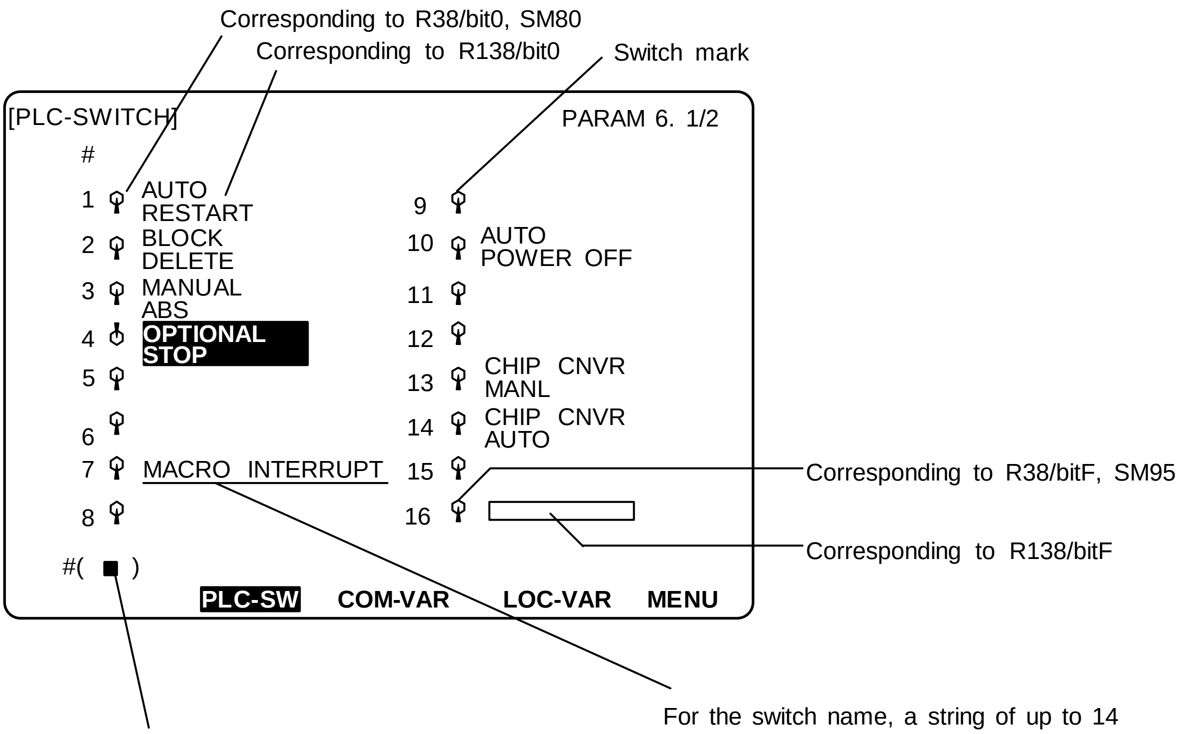
11.3 PLC Switches

Similar function to machine operation switches can be provided by using the controller setting and display unit. The number of switch points is 32. The switch names can be given as desired.

11.3.1 Explanation of Screen

The screen is explained below.

PARAMETER SCREEN PLC SWITCH (MENU)



Setting part switch on, off is indicated.
For example, when 1 is set and INPUT is pressed, #1 switch is turned on.
When the same operation is again performed, the switch is turned off.
However, on/off control of certain switches may trigger other switches to turn on or off (depending on the user PLC SM80 to SM111 operations).

For the switch name, a string of up to 14 alphanumeric and kana characters (kanji requires 2-character space) can be displayed.

Switch on state display part :
Switch off state display :

11.3.2 Explanation of Operation

To turn on or off a switch, set the number of the switch to be turned on or off in the parentheses of setting part # ( ) and press the INPUT CALC key.

Depending on the state of the switch, its input device R38, 39 is turned on (off) and accordingly the switch mark indicates the on (off) state.

When the optional stop is to be turned on:

Set 4 in # ( ).

Press the INPUT key.

[PLC-SWITCH]

#

1	AUTO RESTART	9	
2	BLOCK DELETE	10	AUTO POWER OFF
3	MANUAL ABS	11	
4	**OPTIONAL STOP**	12	
5	HANDLEIT	13	CHIP CNVR MANL
6	PROGRAM RESTART	14	CHIP CNVR AUTO

The switch can be turned off (on) the same way.
Special relay SM can reverse the switch on/off states. When special relay SM is activated, the on/off state of the corresponding switch and device R is reversed.
To display the switch validity state, etc., the switch name can be highlighted. To do this, turn on or off output device R corresponding to the switch name.
The corresponding table of the switch numbers, input device R, output device R, and special relay SM is listed below:

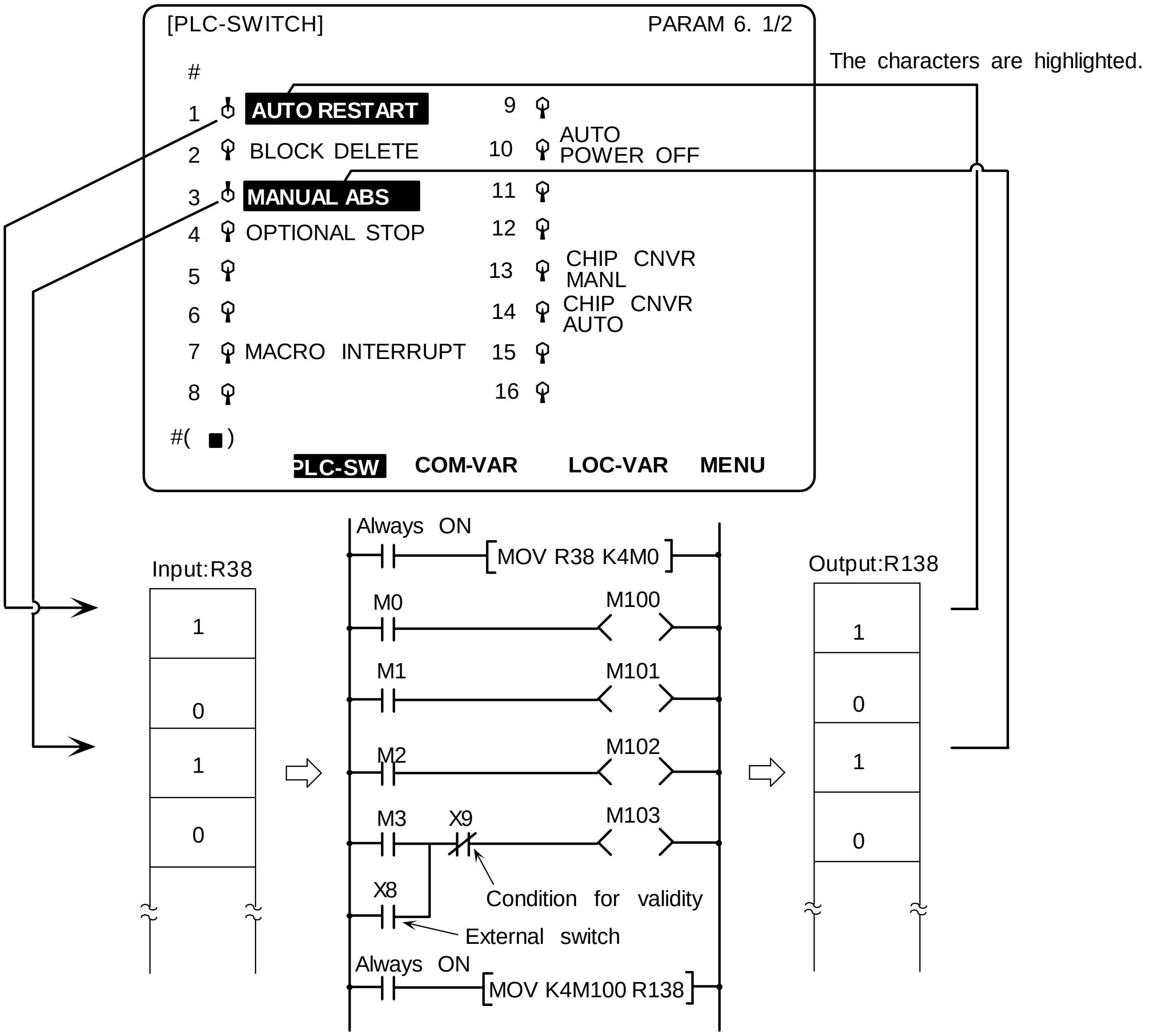
Switch No.	Corresponding device			Switch No.	Corresponding device		
	Input	Output	Highlight		Input	Output	Highlight
#1	R38 / bit0	R138 / bit0	SM80	#17	R39 / bit0	R139 / bit0	SM96
#2	R38 / bit1	R138 / bit1	SM81	#18	R39 / bit1	R139 / bit1	SM97
#3	R38 / bit2	R138 / bit2	SM82	#19	R39 / bit2	R139 / bit2	SM98
#4	R38 / bit3	R138 / bit3	SM83	#20	R39 / bit3	R139 / bit3	SM99
#5	R38 / bit4	R138 / bit4	SM84	#21	R39 / bit4	R139 / bit4	SM100
#6	R38 / bit5	R138 / bit5	SM85	#22	R39 / bit5	R139 / bit5	SM101
#7	R38 / bit6	R138 / bit6	SM86	#23	R39 / bit6	R139 / bit6	SM102
#8	R38 / bit7	R138 / bit7	SM87	#24	R39 / bit7	R139 / bit7	SM103
#9	R38 / bit8	R138 / bit8	SM88	#25	R39 / bit8	R139 / bit8	SM104
#10	R38 / bit9	R138 / bit9	SM89	#26	R39 / bit9	R139 / bit9	SM105
#11	R38 / bitA	R138 / bitA	SM90	#27	R39 / bitA	R139 / bitA	SM106
#12	R38 / bitB	R138 / bitB	SM91	#28	R39 / bitB	R139 / bitB	SM107
#13	R38 / bitC	R138 / bitC	SM92	#29	R39 / bitC	R139 / bitC	SM108
#14	R38 / bitD	R138 / bitD	SM93	#30	R39 / bitD	R139 / bitD	SM109
#15	R38 / bitE	R138 / bitE	SM94	#31	R39 / bitE	R139 / bitE	SM110
#16	R38 / bitF	R138 / bitF	SM95	#32	R39 / bitF	R139 / bitF	SM111

(Note 1) Input devices R38, 39 hold the state even if power is turned OFF.

○ The table below shows the message displayed during operation on the PLC switch screen.

No.	Message	Explanation	Remedy
E01	SETTING ERROR	A number outside the allowable setting range from 1 to 32 is specified in # ( ).	Specify a valid number within the range.

11.3.3 Signal Processing

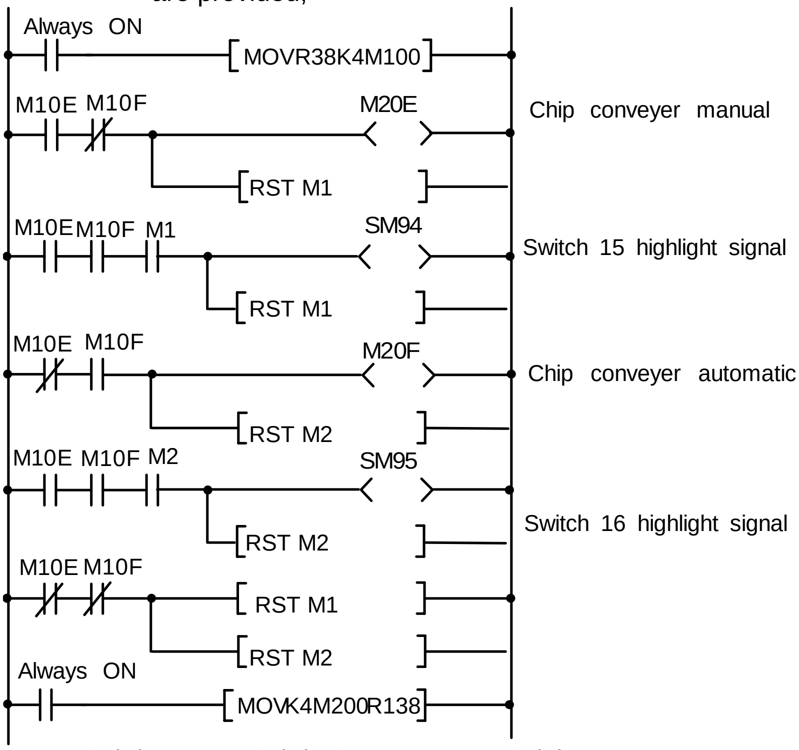


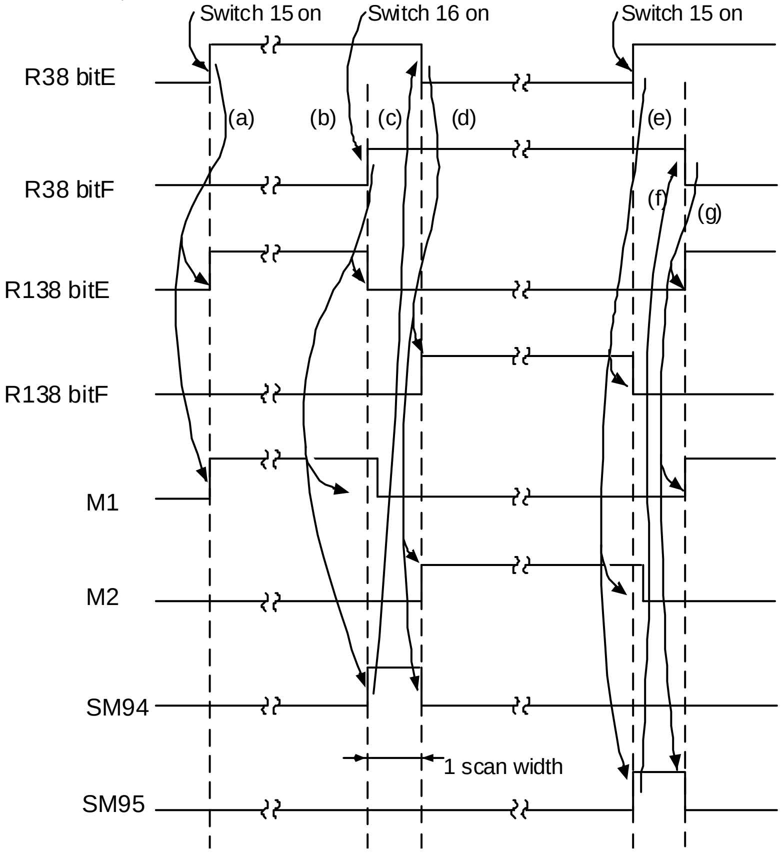
- When setting is done on the PLC switch screen, the input device R38, R39 corresponding to the specified switch number is turned on or off to switch over the switch state.
- When special relay SM is turned on from the user PLC, its corresponding input device R38, R39 and the switch state are reversed. Special relay SM is reset immediately after the CNC reverses the input device R38, R39 and the switch state. It is turned on by one pulse (scan) only also in the user PLC. In either case, when output device R138, R139 is set to on based on the input device R38, R39 state, the corresponding switch name is highlighted.

The following shows an example of operation of special relay SM from the user PLC.

(1) Two-point switch

(Example) When two opposite switches, chip conveyer manual and chip conveyer automatic, are provided;





(a) When switch 15 (R38 bitE) is on and switch 16 (R38 bitF) is off, R138 bitE and M1 turn on. [Initial state]

(b) When switch 16 (R38 bitF) turns on while being in state (a), R138 bitE turns off, SM94 turns on, and M1 turns off.

(c) Turning SM94 on reverses R38 bitE (to off).

(d) When R38 bitE is off and R38 bitF is on,SM94 turns off and R138 bitF and M2 turn on.

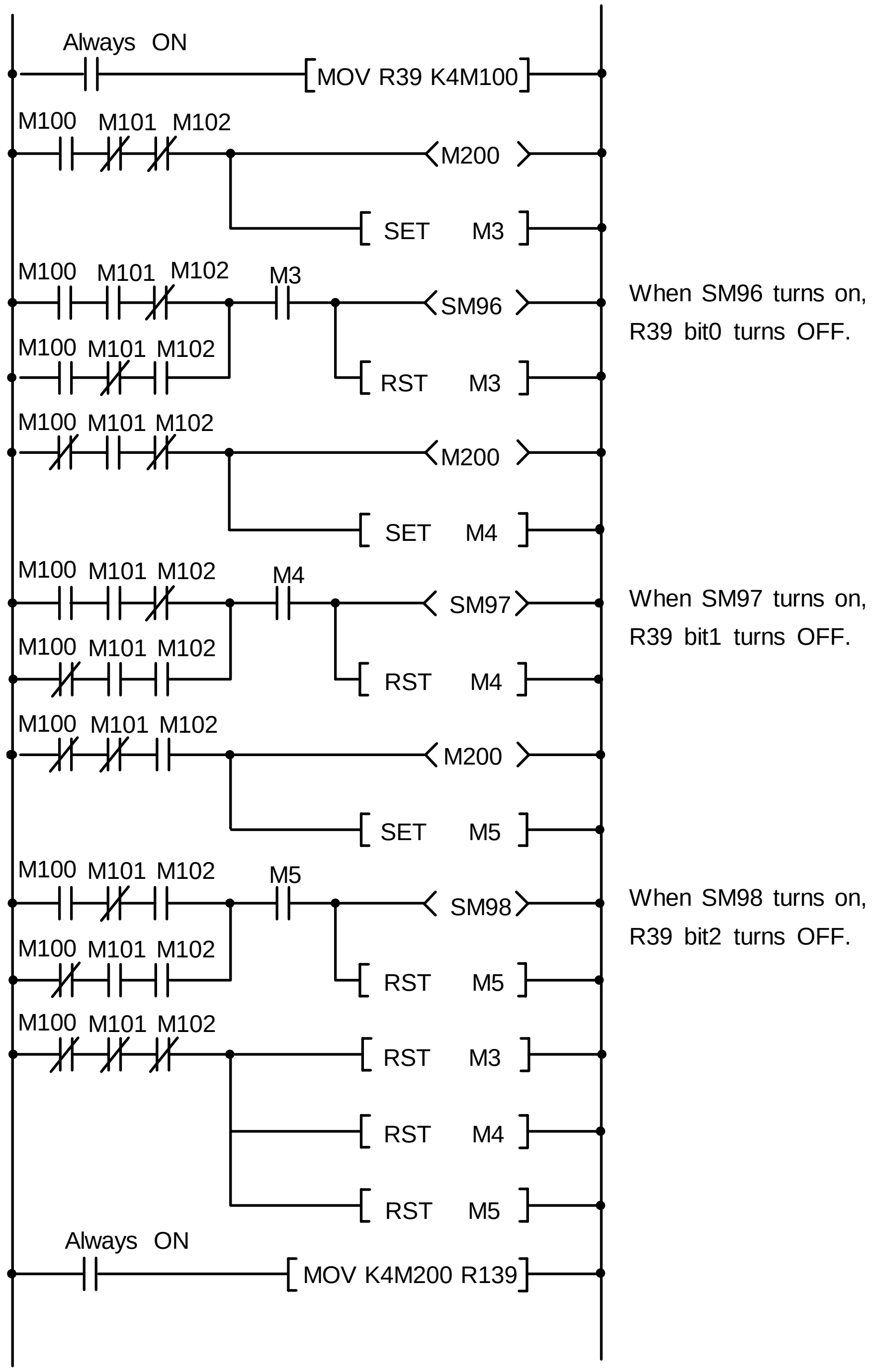
(e) When switch 15 (R38 bitE) turns on while being in state (d), R138 bitF turns off, SM95 turns on, and M2 turns off.

(f) Turning SM95 on reverses R38 bitF (to off).

(g) When R38 bitF is off and R38 bitE is on, SM95 turns off and R138 bitE and M1 turn on again.

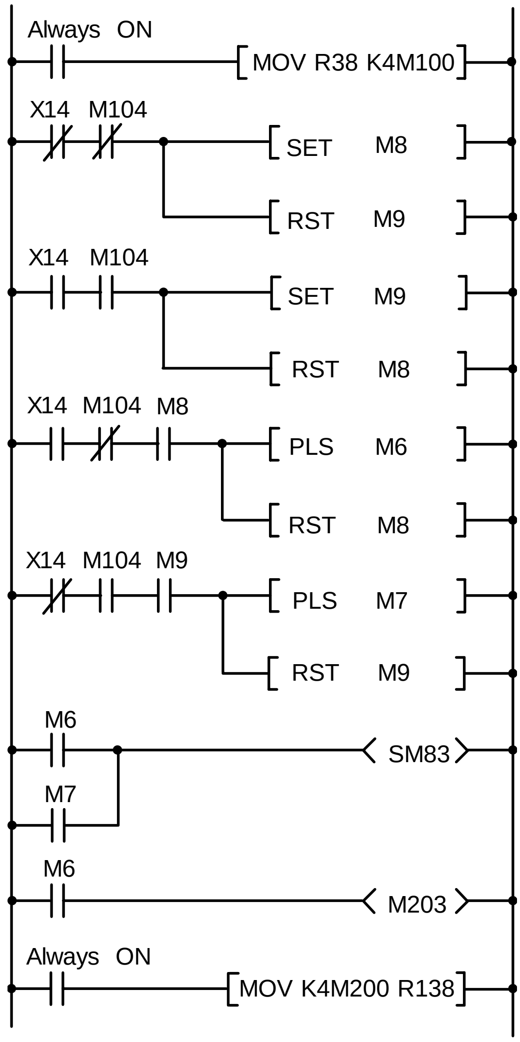
(2) Three-point switch

(Example) When three opposite switches 17, 18, and 19 are provided;



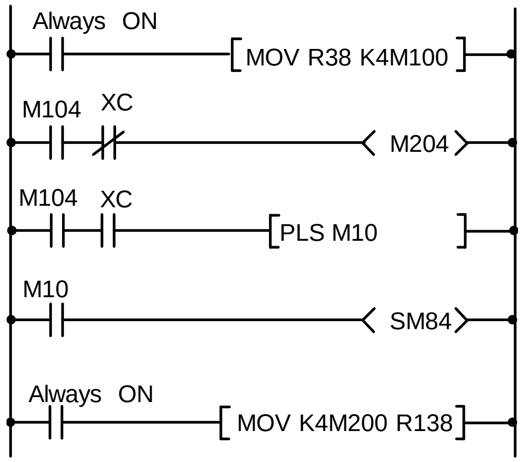
(3) External switch and PLC switch

(Example 1) When an external optional stop switch (X14) is provided;



Under sequence control in the above example, the switch marks on the PLC switch screen can be operated from both external and PLC switches.

(Example 2) When an external switch (XC) that inhibits a PLC switch handle interrupt is provided;



Under sequence control in the above example, when the external switch (XC) is on, the PLC switch for a handle interrupt cannot be turned on.

11.3.4 Switch Name Preparation

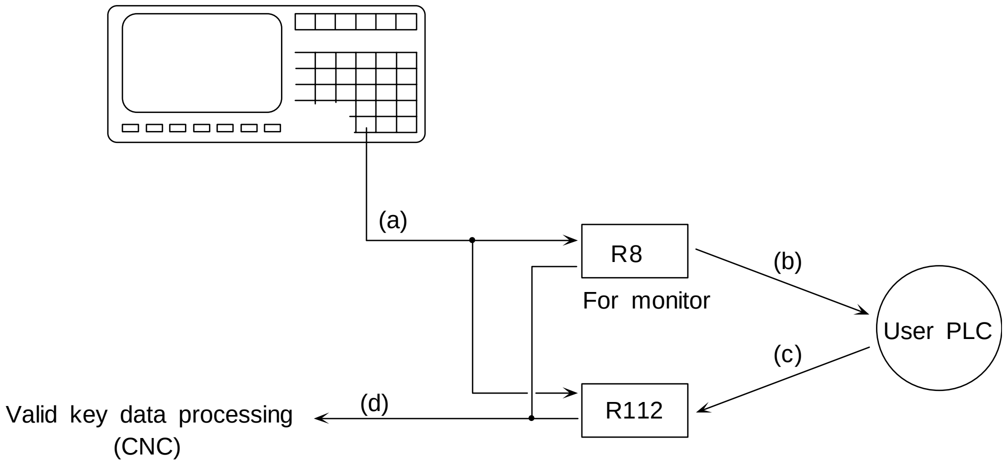
Prepare PLC switch names by using PLC development software (GX Developer).

According to the description format, set the number of characters for one switch name and the number of switch names to be prepared, then prepare switch name data. The maximum length of a switch name is 14 characters. A maximum of 32 switch names can be prepared.
For details, refer to the section "1.4 PLC Creating PLC Message Data".

11.4 Key Operation by User PLC

The same operation as if the operator performed key operation can be performed by operating key data by user PLC.

11.4.1 Key Data Flow



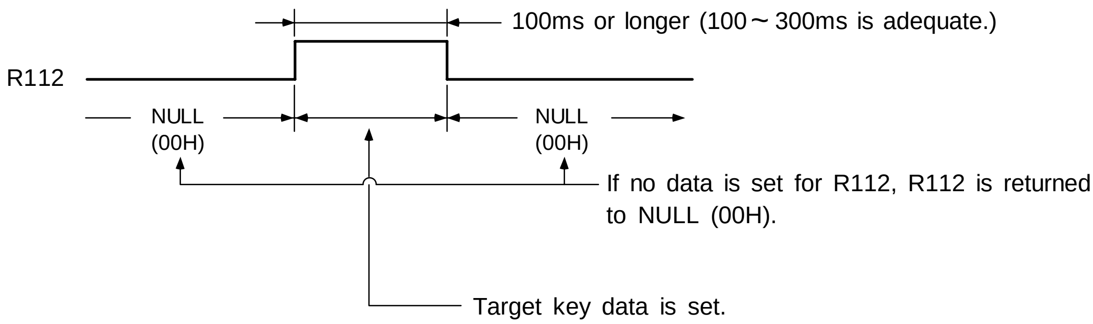
(a) Key data is set in file registers R8 and R112 at the top of the user PLC main.
(b) The user PLC refers to the key data and performs necessary processing.
(c)The user PLC sets the key data matching the operation board being used in R112.
(d) After user PLC main processing is performed, controller performs valid key data processing according to the R8 and R112 contents.

11.4.2 Key Operations That Can Be Performed

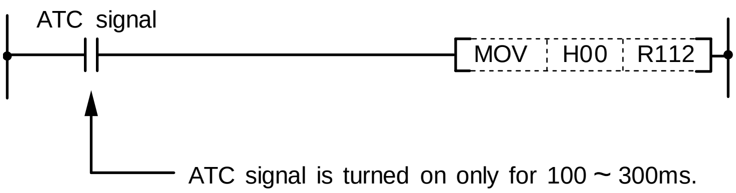
(1) When a key is pressed, it is ignored.
· The R8 contents are judged and NULL (00H) code is set in R112.
(2) When R8 is NULL, that is, key operation is not performed, user PLC performs key operation conforming to the operator.
· Key data matching the target operation is set in R112.

11.4.3 Key Data Processing Timing

Key data is processed at the timing shown below.
Set data in R112 only when it is necessary. Normal key operation by the operator is made impossible.



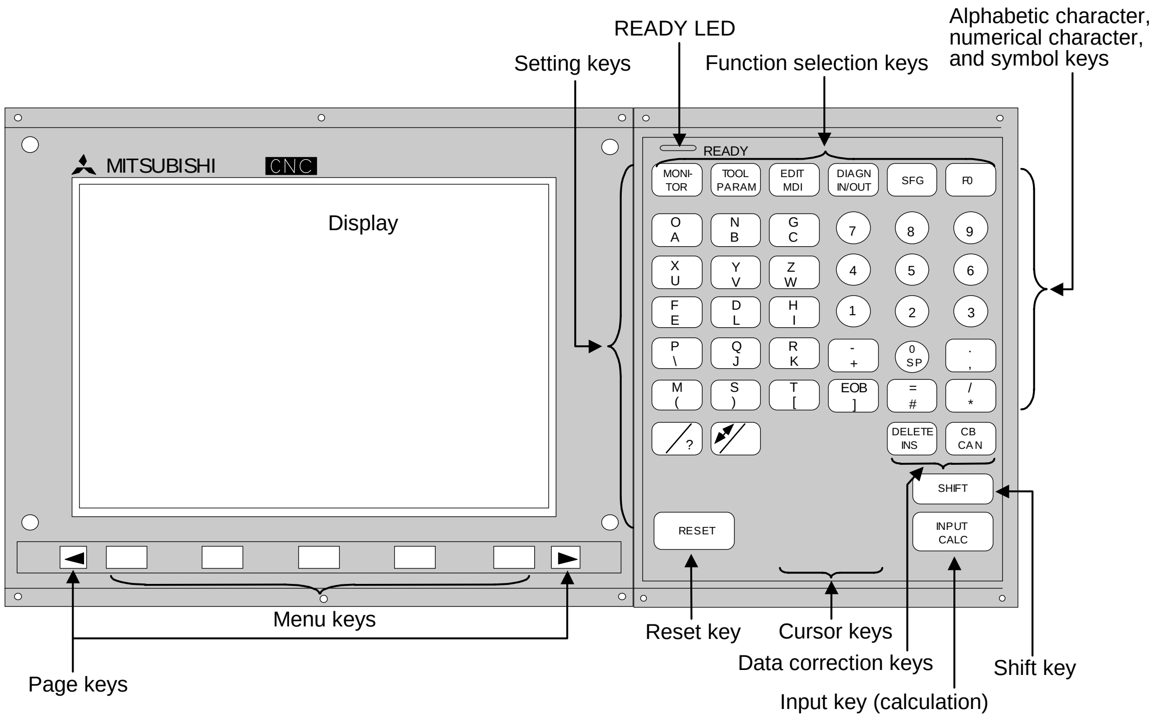
(Example)



11.4.4 Layout of Keys on Communication Terminal

The layouts for the keys on the communication terminal used with this controller is as shown below.

(1) Key layout for setting and display unit (NC-dedicated display unit)



(Note 1) When inputting an alphabet or symbol on the lower right of the alphabet or symbol keys, press SHIFT, and then press the corresponding key.

(Example) When SHIFT O/A are pressed, "A" will be input.

11.4.5 List of Key Codes

Key symbol	Code (HEX)	Key symbol	Code (HEX)	Key symbol	Code (HEX)	Key symbol	Code (HEX)
MONITOR	80	(↑) ⇈	0B(F8)	– (+)	2D(2B)	O (A)	4F(41)
TOOL/PARAM	81	(↓) ⇊	0A(F7)	• (,)	2E(2C)	N (B)	4E(42)
EDIT/MDI	83	(←) ⇇	08 (F5)	EOB ( ])	3B (5D)	G (C)	47 (43)
DIAGN IN/OUT	85	(→) ⇉	09(F6)	= (#)	3D(23)	X (U)	58(55)
SFG	86	DELETE (INS)	7F(8C)	/ (*)	2F(2A)	Y (V)	59(56)
F0	87	C.B.(CAN)	8E(18)			Z (W)	5A(57)
		SHIFT	88	0 (SP)	30(20)	F (E)	46(45)
		INPUT(CALC)	0D(F4)	1	31	D (L)	44(4C)
				2	32	H (!)	48(21)
				3	33	P (I)	50 (49)
Previous page	90	Window key (?HELP)	89(F9)	4	34	Q (J)	51(4A)
Next page	9A	Activ Wind (CTRL)	8A(8B)	5	35	R (K)	52(4B)
Menu 1	91			6	36	M (( )	4D(28)
Menu 2	92			7	37	S ( ))	53(29)
Menu 3	93			8	38	T ([ )	54(5B)
Menu 4	94			9 ($)	39(24)		
Menu 5	95						

* The key signals and codes shown in parentheses are the shift IN side key signals.

Shift is canceled by pressing another key after pressing the shift key, or by pressing the shift key again.

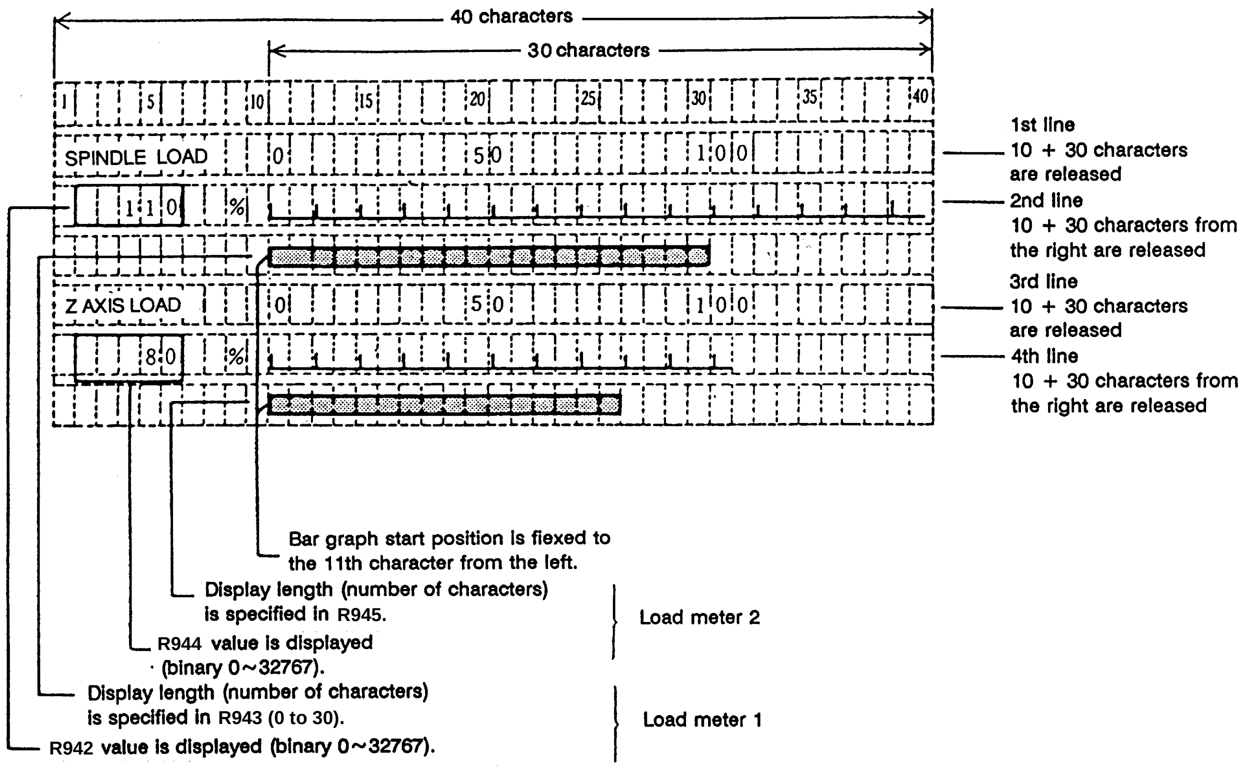
(Example 1)	SHIFT	N(B)	0(SP)	←	Key pressed
	88	42	30	←	Code generated

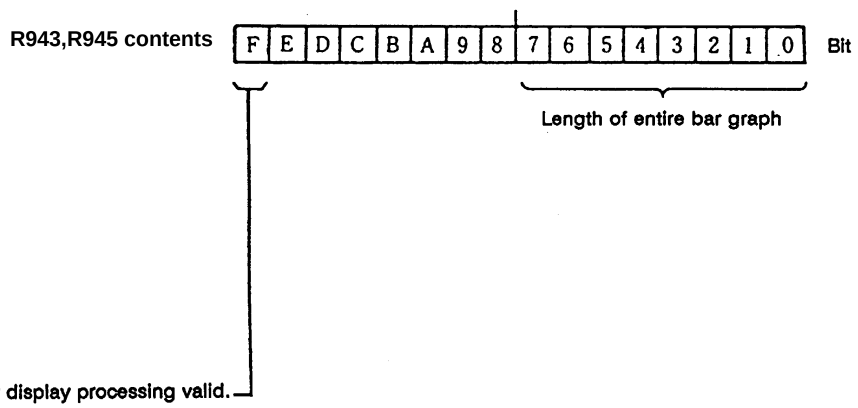
(Example 2)	SHIFT	SHIFT	0(SP)	←	Key pressed
	88	88	30	←	Code generated

11.5 Load Meter Display

The load meter can be displayed by setting a value in the designated file register (R) with the ladder program. The spindle load, Z axis load, etc. characters and scale are created with comments in the PLC development software (GX Developer) message function.
For details, refer to the section "1.4 Creating PLC Message Data".

11.5.1 Interface



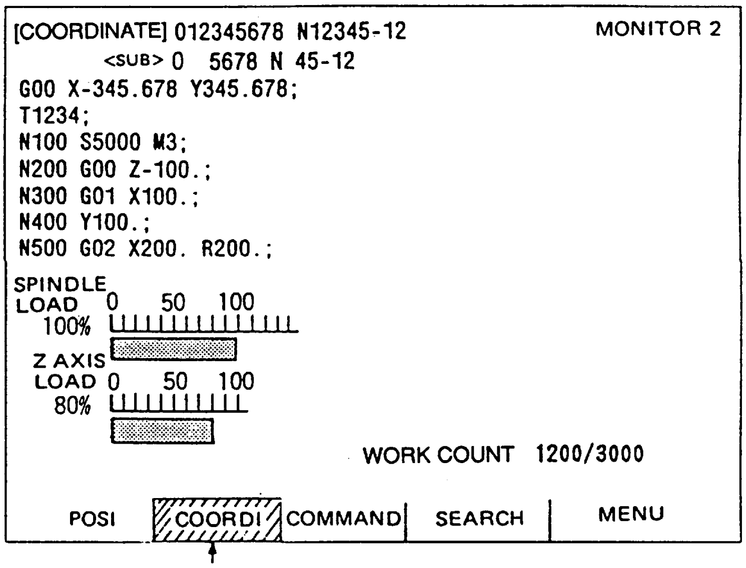


○ **File register (R) for load meter display**

		For $1	For $2	For $3	For $4	For $5	For $6	For $7
Load meter 1	Numerical display	R942	R1042	R1142	R1242	R1342	R1442	R1542
	Bar graph display	R943	R1043	R1143	R1243	R1343	R1443	R1543
Load meter 2	Numerical display	R944	R1044	R1144	R1244	R1344	R1444	R1544
	Bar graph display	R945	R1045	R1145	R1245	R1345	R1445	R1545

(Note 1) Use $1 for models not having a part system.

Display example of type 9 setting and display unit
(**Note:** This screen consists of 80 characters wide x 18 lines long.)



Display on coordinate screen of position display second menu.

11.6 External Machine Coordinate System Compensation

External machine coordinate system compensation is executed by setting compensation data (absolute amount) in the PLC file register (R) for each axis.
Thus, the compensation timing is when PLC rewrites file register (R) compensation data. Necessary condition, timing, etc., are set by user PLC.
The interface between user PLC and CNC is shown below.

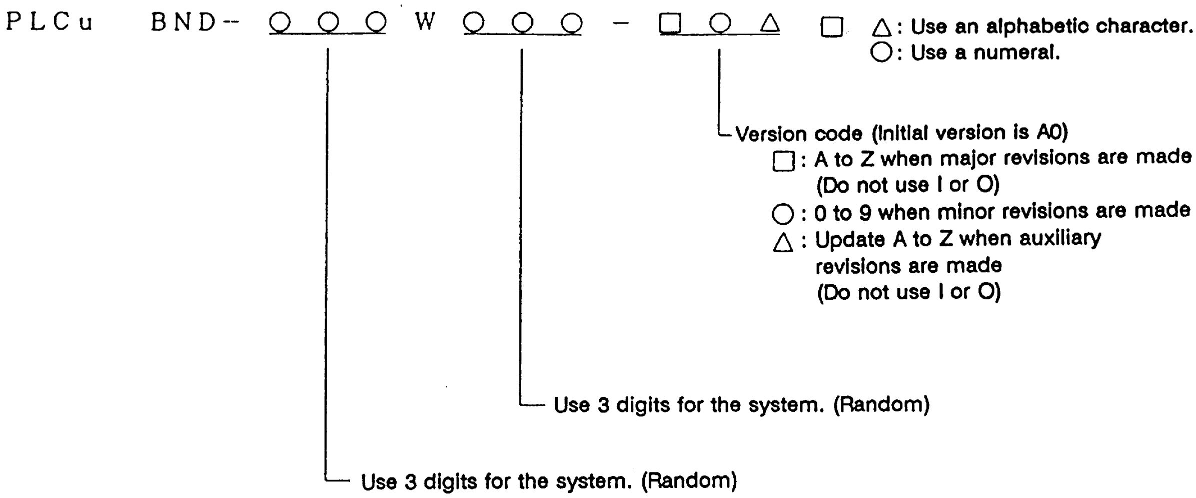
File register	Contents	File register	Contents
R2300	Compensation data for the 1st axis	R2650	Compensation data for the 8 th axis
R2350	Compensation data for the 2nd axis	R2700	Compensation data for the 9 th axis
R2400	Compensation data for the 3rd axis	R2750	Compensation data for the 9 th axis
R2450	Compensation data for the 4th axis	R2800	Compensation data for the 11th axis
R2500	Compensation data for the 5th axis	R2850	Compensation data for the 12th axis
R2550	Compensation data for the 6th axis	R2900	Compensation data for the 13th axis
R2600	Compensation data for the 7th axis	R2950	Compensation data for the 14th axis

Data in file registers (R2300 to R2950) is not backed up. If it must be backed up, use back-up file registers (R6400 to R7199).

(Note 1) The maximum delay to compensation is (one user PLC scan + 15ms). However, smoothing time constant and servo follow delay are not contained.

11.7 User PLC Version Display

The user PLC version can be displayed together with the controller software version on the DIAGN/IN/OUT → menu changeover → configuration (menu) screen of the setting and display unit.

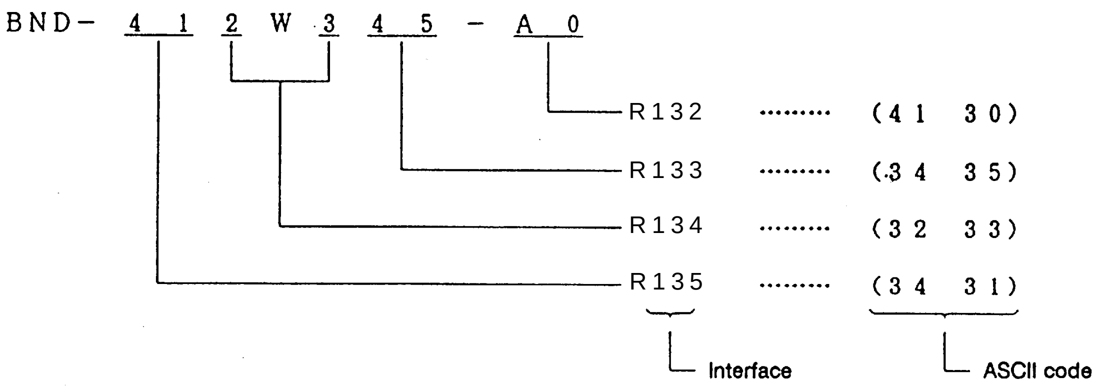


(Note) The user PLC must be controlled by the user.

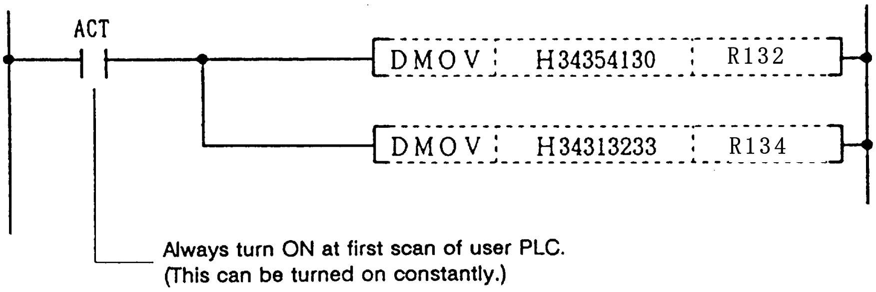
11.7.1 Interface

Data corresponding to the characters to be displayed on the corresponding file register (R) is set.

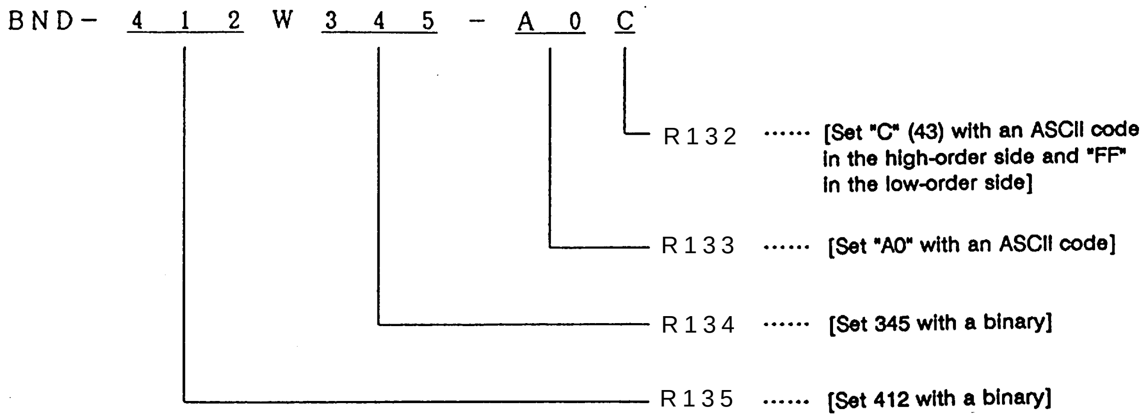
(1) To display a 2-digit version code



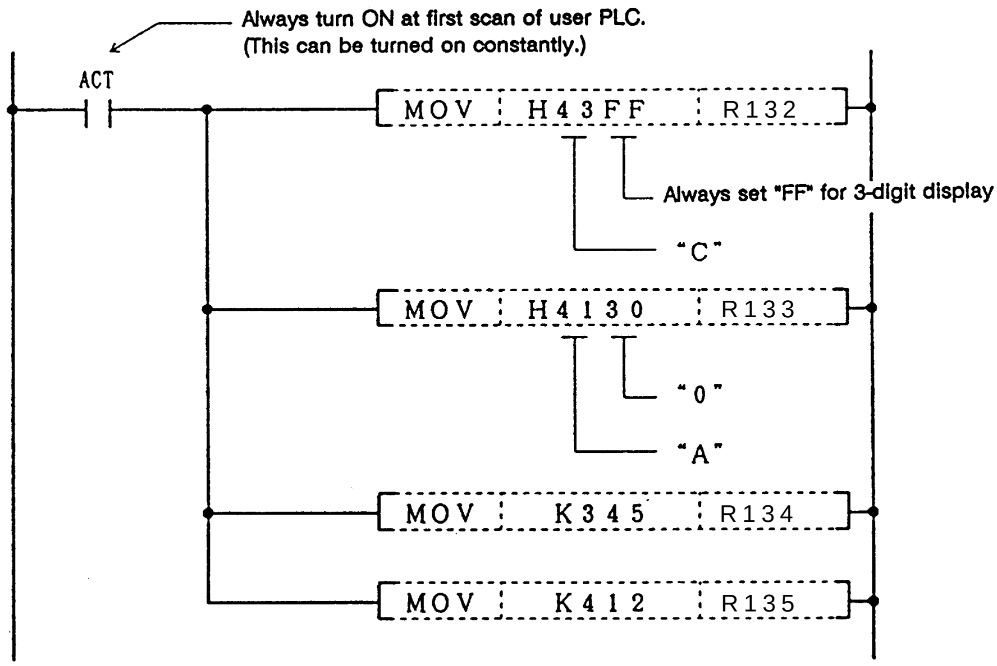
Program example)



(2) To display a 3-digit version code



Program example)



12. PLC Axis Control

12.1 Outline

This function allows an independent axis to be controlled with commands from the PLC, separately from the NC control axis.

12.2 Specifications

12.2.1 Basic Specifications

Item	Details
No. of control axes	Max. 7 axes
Simultaneous control axes	The PLC control axis is controlled independently of the NC control axis. Simultaneous start of multiple PLC axes is possible.
Command unit	Min. command unit 0.001mm (0.0001 inch) 0.0001mm (0.00001 inch) (Same command unit as the NC control axis.)
Feedrate	(Min. command unit 0.001mm) Rapid traverse 0 to 1000000 mm/min. (0 to 100000 inch/min.) Cutting feed 0 to 1000000 mm/min. (0 to 100000 inch/min.) (Min. command unit 0.0001mm) Rapid traverse 0 to 100000 mm/min. (0 to 10000 inch/min.) Cutting feed 0 to 100000 mm/min. (0 to 10000 inch/min.)
Movement commands	Incremental value commands from the current position. Absolute value commands of the machine coordinate system. 0~±99999999 (0.001mm/0.0001inch)
Operation modes	Rapid traverse, cutting feed Jog feed (+), (-) Reference point return feed (+), (-) Handle feed
Acceleration/ deceleration	Rapid traverse, Jog feed Reference point return feed } Linear acceleration/linear deceleration Cutting feed } Exponential function acceleration/ exponential function deceleration Handle feed } Step
Backlash compensation	Provided
Stroke end	Not provided
Soft limit	Provided
Rotation axis commands	Provided Absolute value commands ···· Rotation amount within one rotation. (Rotates the remainder divided by 360°.) Incremental commands ······· Rotates the commanded rotation amount.
Inch/mm changeover	Not provided Command to match the feedback unit.
Position detector	Encoder (absolute position detection also possible)

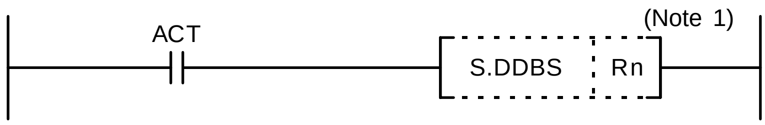
12.2.2 Other Restrictions

(1) There is no mirror image, external deceleration or machine lock function.
(2) Rapid feed override, cutting override and dry run control are not possible.
(3) Automatic operation start, automatic operation stop, reset and interlock NC controls are invalid for PLC control axes.
The same control can be realized using an interface dedicated for PLC control axes.
(4) There is no dedicated emergency switch. The emergency stop is valid in the same manner as the NC control axis.

12.3 PLC Interface

The interface between the PLC and NC is carried out by setting the control information data in the R-register (Note 1) with the PLC, and calling the S.DDBS function.

12.3.1 S.DDBS Function Command



When ACT is set to 1, the PLC axis control process is carried out with the control information data contents. Thus, ACT should be set to 1 during PLC axis control.
Setting ACT to 0 causes a reset status.

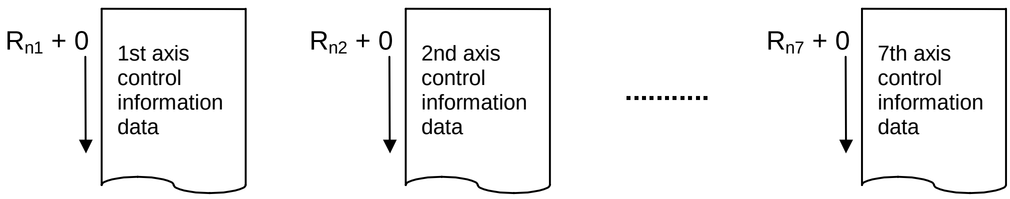
(Note 1) The following R-registers can be used.
R4000 to R4499 (No battery backup)
R6400 to R7199 (Battery backup)

12.3.2 Control Information Data

Set the control information data in the R-register before calling the S.DDBS function command. The following is a list of control information data.

R_n + 0	2 bytes	Command	PLC → CNC
1	2 bytes	Status	CNC → PLC
2	2 bytes	Alarm details	CNC → PLC
3	2 bytes	Control signal	PLC → CNC
4	2 bytes	Axis designation	PLC → CNC
5	2 bytes	Operation mode	PLC → CNC
6, 7	4 bytes	Feedrate	PLC → CNC
8, 9	4 bytes	Movement data	PLC → CNC
10, 11	4 bytes	Machine position	CNC → PLC
12, 13	4 bytes	Remaining distance	CNC → PLC

A max. of 7 axes can be controlled by the PLC. Each axis should have its own control information data.



12.3.3 Control Information Data Details

12.3.3.1 Commands

Commands consist of main commands and sub-commands.

	F 8	7 0
$R_n + 0$	Sub-commands	Main commands

Main commands: The types of DBBS main commands are as follows.
1: Search
2: PLC axis control

Sub-commands: The PLC axis control sub-command is as follows.
0: Movement data output and control signal output

(Note 1) "Input" and "output" are the input/output looking from the PLC side.

12.3.3.2 Status

The status is set by the NC to indicate the execution status of this function command and the status of the axis being controlled.

	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
R_n + 1																

bit 0: busy	Command processing	bit 8 : oper	Option error
1: den	Axis movement completed	9:	
2: move	Axis moving	A:	
3: SA	Servo ready	B:	
4: svon	Servo ON	C:	
5: ZP	Reference point reached	D:	
6:		E: ALM2	Axis in control alarm
7: WAIT	Axis movement wait	F: ALM1	Control information data designation alarm

bit 0: busy   Command processing

This turns ON when the command is being processed.
The next command is not received while this bit is ON. The next command to be issued is received while this bit is OFF.
Even if movement distance is microscopically small, at least 1 scan is turned ON.

bit 1: den   Axis movement completed

This bit turns ON when the initialization and commanded movement are completed. This bit stays OFF during movement, even when an interlock is applied. This bit turns ON at reset or servo OFF, or when ACT = 0.

bit 2: move   Axis moving

This bit turns ON when the machine is moving, and turns OFF when the machine is stopped.
When movement distance is microscopically small, PLC may be unable to detect having turned ON.

bit 3: SA   Servo ready

This bit turns ON when the servo is ready. It turns OFF during emergency stops and servo alarms.

bit 4: svon   Servo ON

This bit turns OFF when a servo OFF signal is output. It also turns OFF during emergency stops and servo alarms.
Machine movement is possible when this signal is ON.

bit5: ZP   Reference point reached

This bit turns ON when the reference point is reached after completion of a reference point return.
It turns OFF when the machine moves.

bit7: WAIT   Axis movement wait

This bit turns ON in the buffering mode when the axis movement of the previous block has been completed, and the machine is in a WAIT 5 status. It turns OFF when the previous block movement is completed and the movement of the next block begins.

bit 8: oper   Option error

This bit turns ON when an attempt is made to execute PLC axis control when there is no PLC axis control option.

bit E: ALM2   Axis in control alarm

This bit turns ON when an alarm occurs (such as a servo alarm) during execution of axis control. Axis control cannot be executed while this bit is ON.
After the cause of the alarm has been removed, turn the bit OFF by outputting a reset signal, setting ACT to 0, or turning the power OFF then ON again.

(Note) When alarms occur during axis control, the same alarms appear in the screen as for NC control axes. Set the PLC 1st axis to "1", and the PLC 2nd axis to "2".

Example: When a servo alarm occurs for the PLC 1st axis

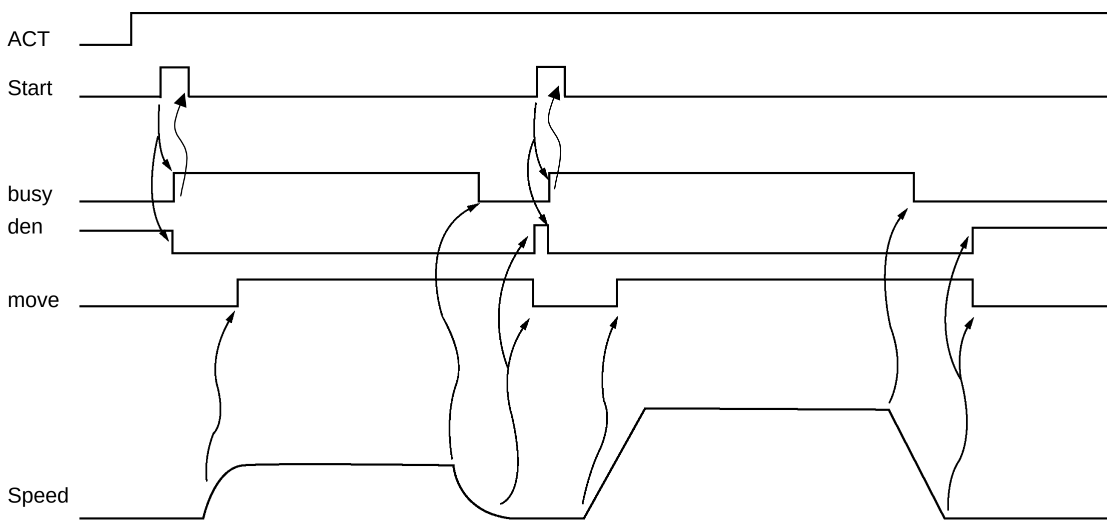
```
S03 Servo alarm        52        1
                                 └─ PLC axis
```

bit F: ALM:1   Control information data designation alarm

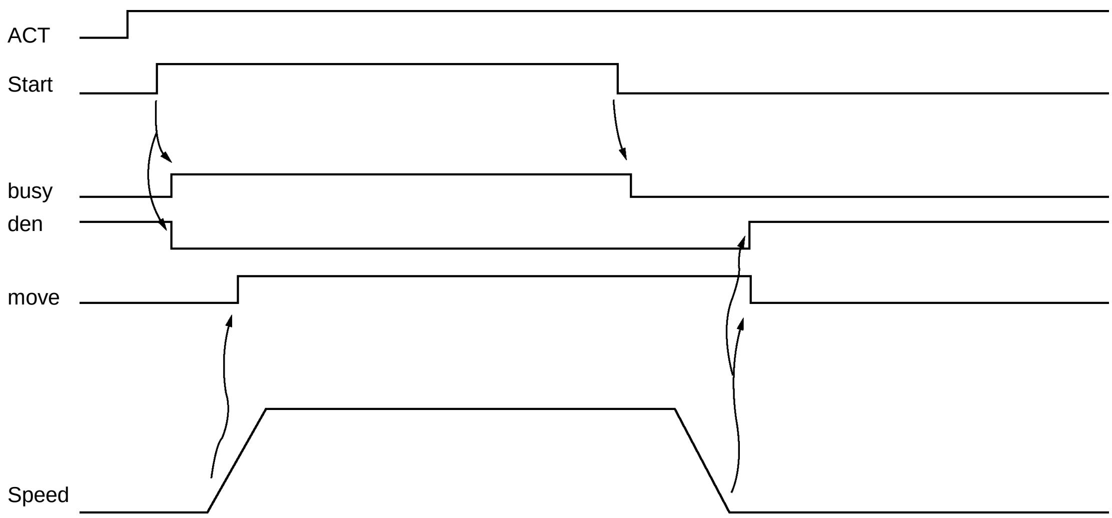
This bit turns ON when the designated details of the control information data are illegal. Thus, the PLC axis control process is not executed. Turn the bit OFF by correcting the data, outputting a reset signal, or setting ACT to 0.

Timing chart

(1) For rapid traverse and cutting feed mode



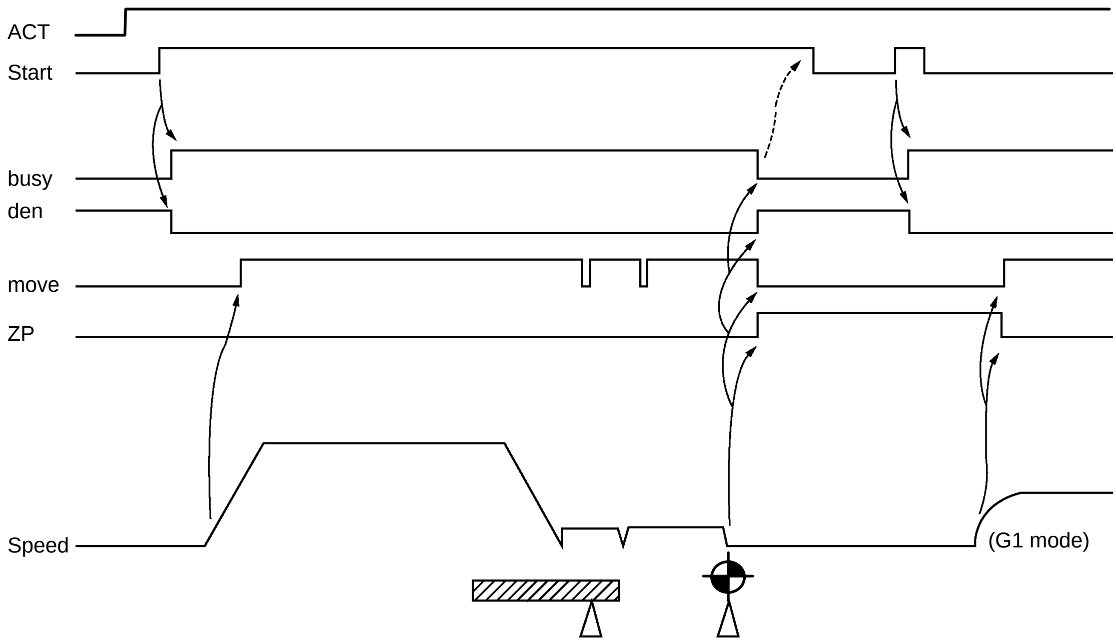
(2) For jog feed mode



(Note) The axis moves by jog feed only during start ON.

(3) For reference point return feed mode

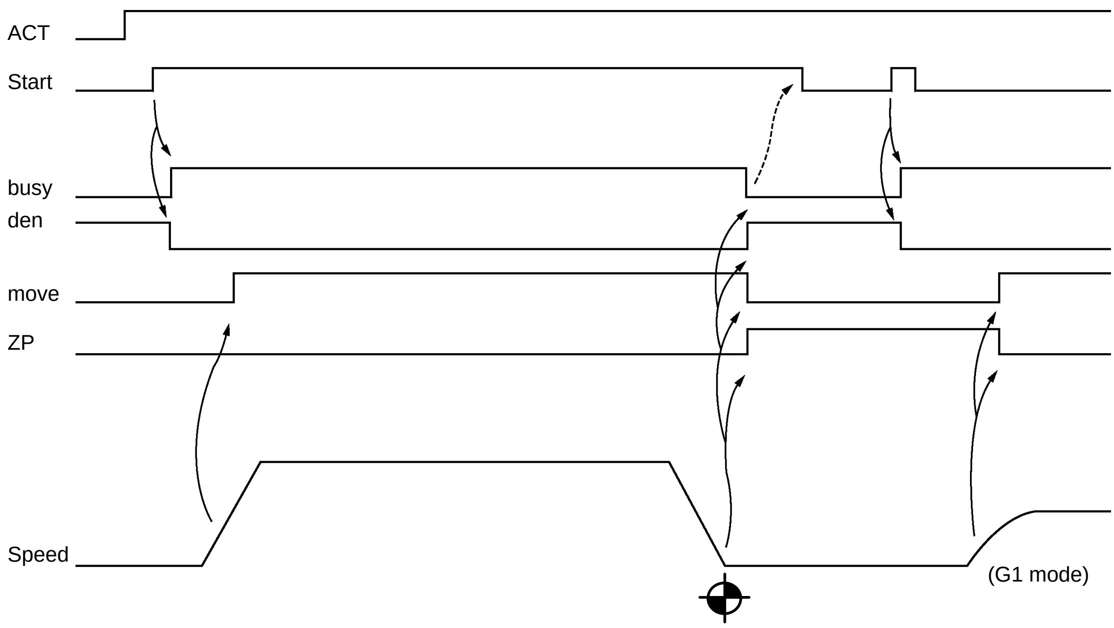
(3-1) Dog-type reference point return



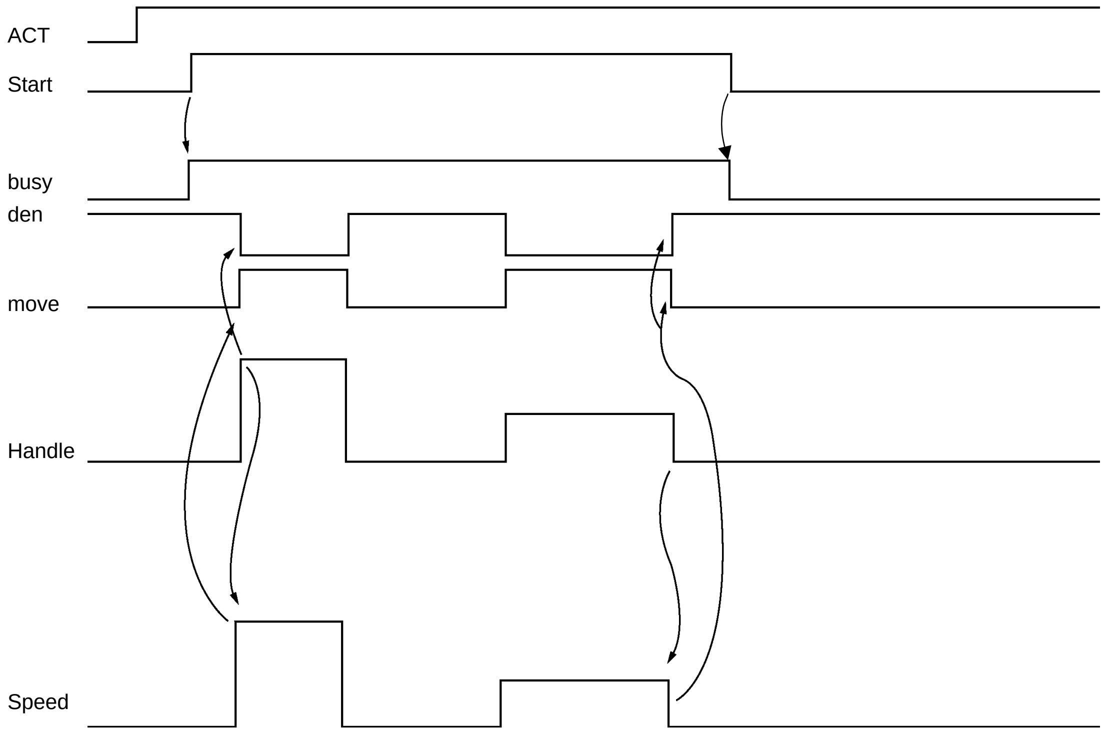
(Note 1) The axis moves by reference point return feed only during start ON. Turn the start OFF after confirming that the reference point has been reached.

(Note 2) The first reference point return after the power is turned ON is always dog-type. All returns after that are high-speed reference point returns.

(3-2) High-speed reference point return

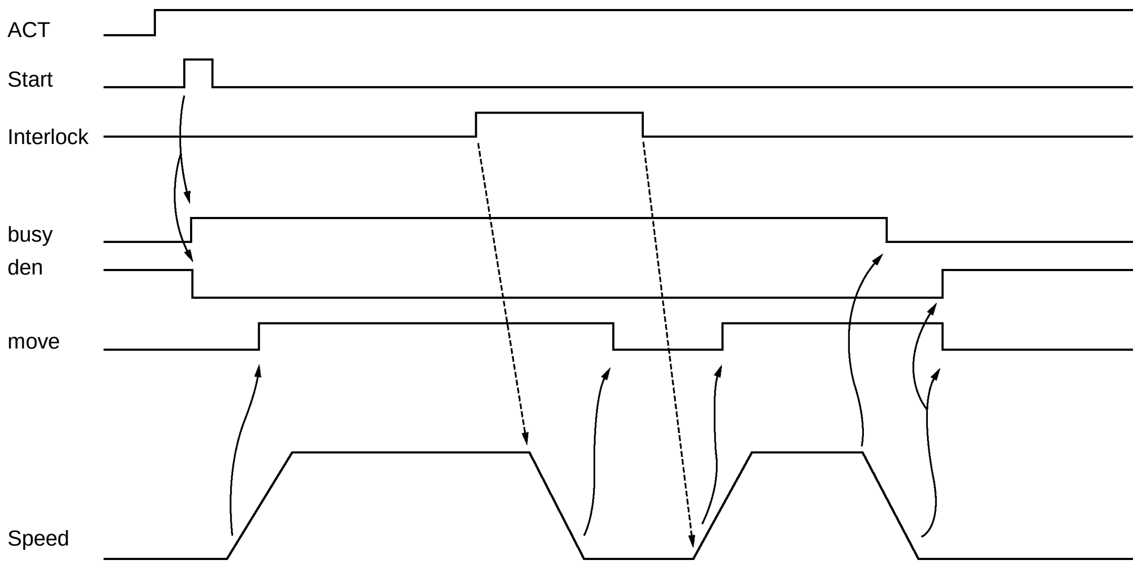


(4) **For handle feed mode**

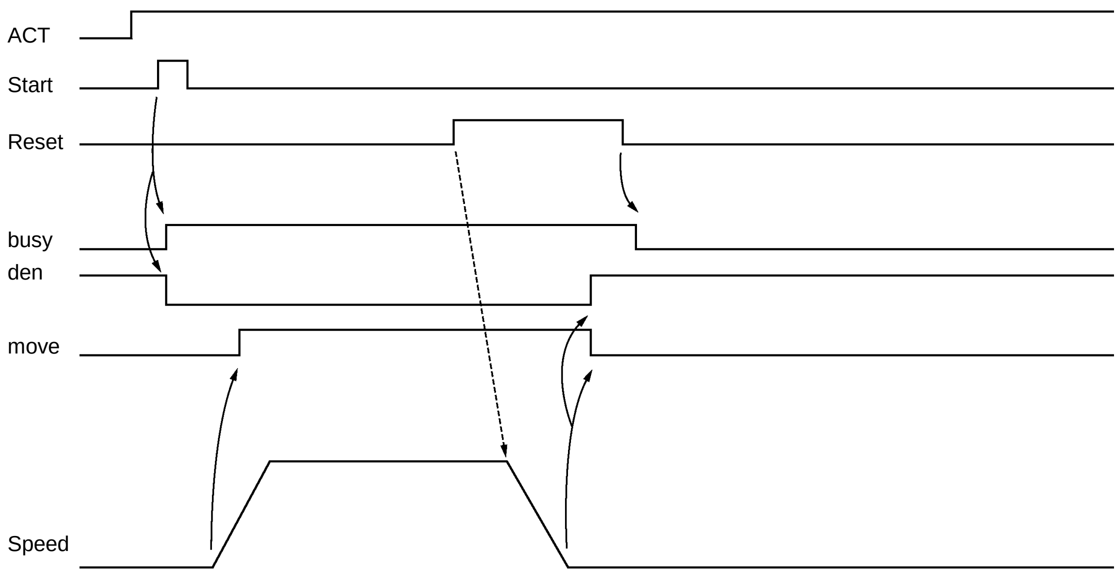


(Note) Handle feed is possible only during start ON.

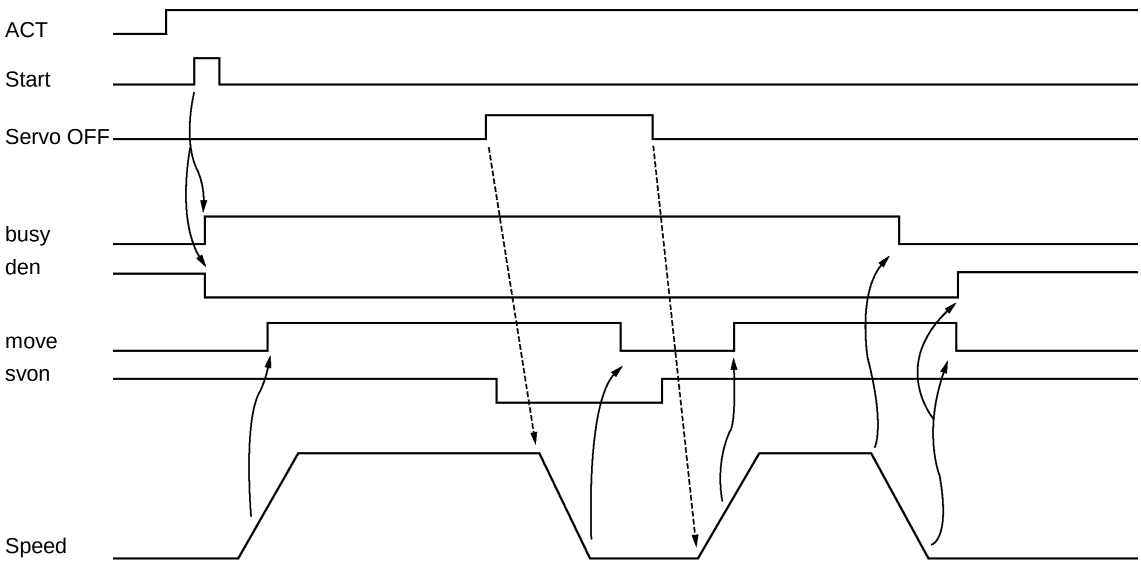
(5) When the interlock signal is ON (= 1)



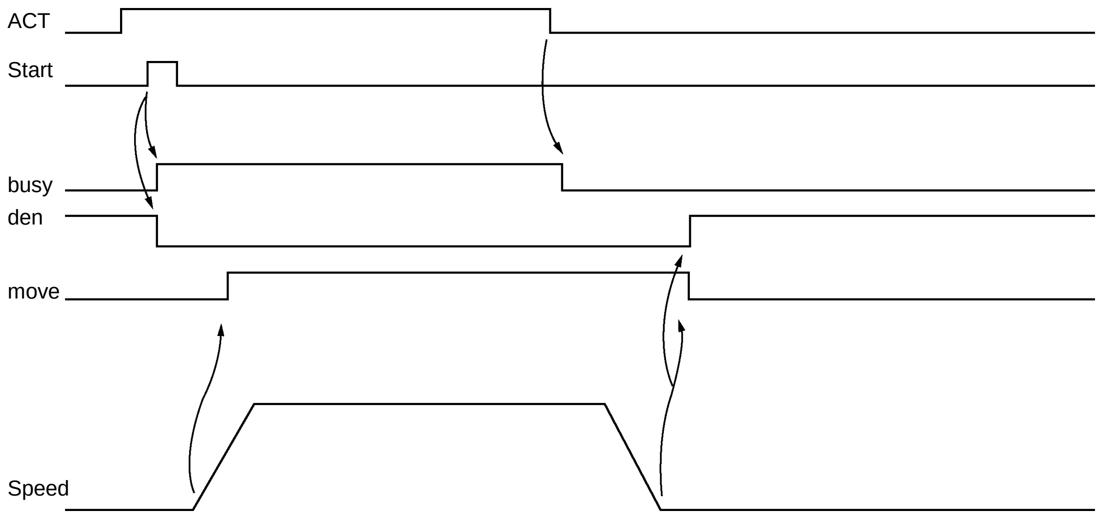
(6) When the reset signal is ON (= 1)



(7) When the servo OFF signal is ON (= 1)

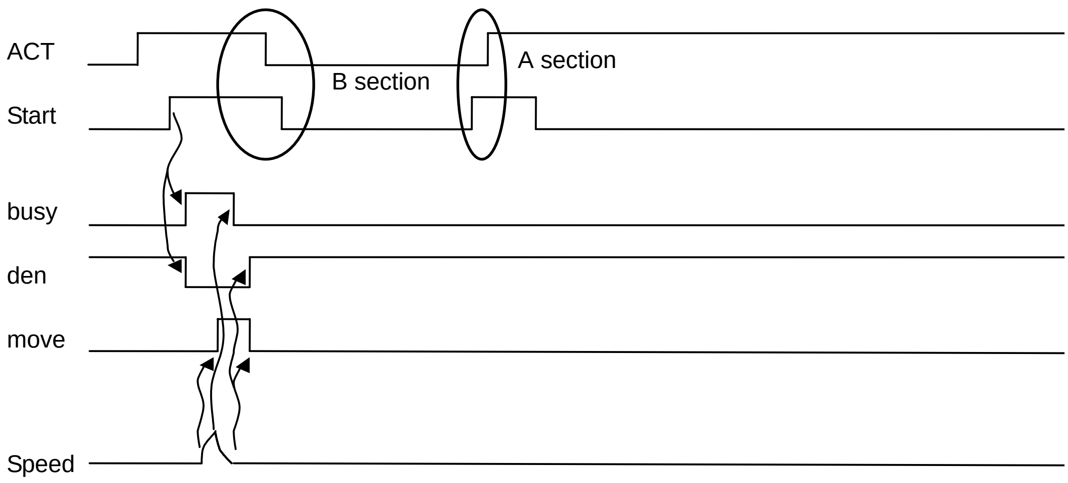


(8) When the ACT signal is OFF (= 0)



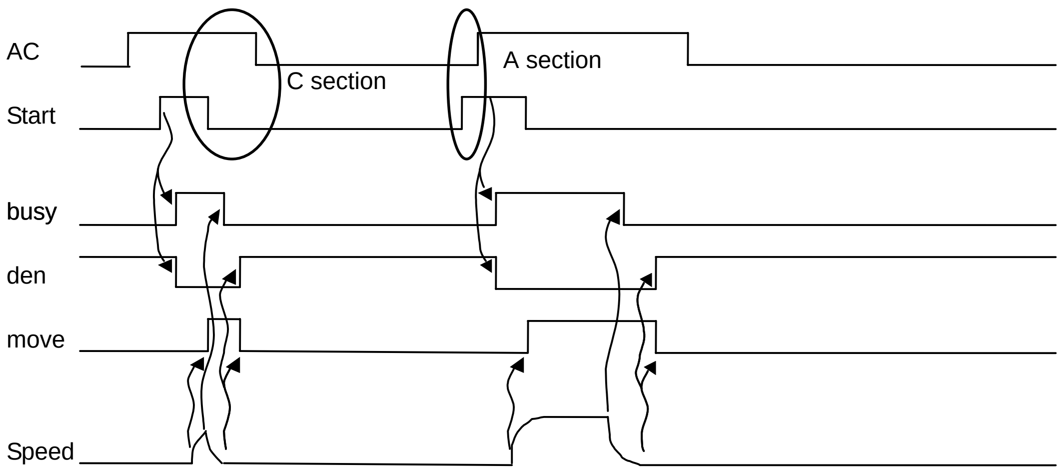
(9) When the start signal ON (=1) is earlier than the ACT signal ON (=1) (A section)

(9-1) When OFF the last ACT signal OFF is earlier than a start signal OFF (B section)



When ACT is turned OFF at the 1st time, a start signal continues being ON, and since the start signal is turned ON previously when ACT is turned ON at the 2nd time, NC cannot detect the standup of a start signal. Therefore, a PLC axis does not operate by the 2nd start signal.

(9-2) When the ACT signal OFF is later than the start signal OFF (C section)



When ACT is turned OFF at the 1st time, a start signal is turned OFF, NC can detect the standup of a start signal. Therefore, a PLC axis operates by the 2nd start signal.

When the start signal ON (=1) is earlier than the ACT signal ON (=1), a PLC axis does not operate according to the state of the last start signal. In order to avoid such a situation, please turn ON ACT after a start signal or turn OFF ACT after a start signal.

12.3.3.3 Alarm No.

The alarm Nos. of status ALM1 and ALM2 are set.

F 8	7 0
ALM1 Alarm No.	ALM2 Alarm No.

The details of each alarm No. are shown below.

(1) ALM1 (Control information data designation alarm)

Alarm No.	Details
01	Control signal illegal (A signal other than a registered control signal has been commanded.)
02	Axis No. illegal
03	Operation mode illegal (0 to 6)
04	Movement data range exceeded -99999999 to +99999999
05	
06	
⋮	
10	Reference point return not complete (absolute value command not possible)
11	
12	

(1) ALM2 (Axis in control alarm)

Alarm No.	Details
0	Servo alarm (Alarm No. is displayed in the PLC axis monitor screen. Refer to the Drive Unit Maintenance Manual for details.)
1	Z-phase not passed
2	Soft limit (+)
3	Soft limit (-)

12.3.3.4 Control Signals (PLC axis control information data)

Control signals such as start, interlock, reset, axis removal and axis removal 2 are designated for the PLC axis.

	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
$R_n + 3$																

bit 0: Start	bit 8: Absolute value command
1: Interlock	9:
2: Reset	A:
3: Servo OFF	B:
4: Axis removal	C:
5: Axis removal 2	D:
6:	E:
7:	F:

bit 0: Start

Starting begins at the at the rising edge (OFF -> ON) of the start signal, based on the control information data.
Turn ON a start signal after turning ON the ACT of DDBS starting conditions.
The axis does not move during interlock, servo OFF, axis removal and axis removal 2. Movement starts after interlock, servo OFF, axis removal and axis removal 2 are canceled.
Start is invalid during resetting.

bit 1: Interlock

The moving PLC axis executes a deceleration stop when the interlock signal turns ON. The stopped PLC axis will resume movement when the interlock signal turns OFF (is canceled).

bit 2: Reset

The PLC axis is reset when the reset signal turns ON. Moving PLC axes will execute a deceleration stop. Commands and controls are invalid during resetting.
If the reset signal turns ON during an alarm occurrence, the alarm will be cleared.

bit 3: Servo OFF

The PLC axis will execute a deceleration stop and its servo will turn OFF when the servo OFF signal turns ON. Whether the PLC axis movement is compensated during servo OFF can be selected in the basic specification parameter "#1064 svof".
A servo ON status will result when the power is turned ON.

bit4: Axis removal

The axis will execute a deceleration stop, and a servo OFF status will result, when the axis removal signal turns ON. A servo ON status will result and the stopped PLC axis will resume movement when the axis removal signal turns OFF (is canceled).
Axis removal is validated when either this signal or machining parameter and axis parameter "#8201 Axis Removal" is validated.
The zero point return will become incomplete when the axis is removed. Therefore, a dog-type reference point return must be completed again when starting with an absolute value command.

bit 5: Axis removal 2

The axis will execute a deceleration stop, and a servo OFF/ready OFF status will result, when the axis removal 2 signal turns ON. A servo ON/ready ON status will result for the stopped PLC axis when the axis removal 2 signal turns OFF (is canceled).
A restart must be executed to start the movement again.
Position control cannot be carried out while the axis removal 2 signal is ON. However, position detection is possible so the position will not be lost.

bit 8: Absolute value command

Turn this bit ON when the movement data is commanded in absolute values.
When this bit is OFF, the commands will be processed as incremental value commands.

12.3.3.5 Axis Designation

The axis No. of the PLC axis is designated.

$R_n + 4$	Axis designation

0: 1st axis
1: 2nd axis
:   :
6: 7th axis

12.3.3.6 Operation Mode

The operation mode for the PLC axis is designated.

For example, in the handle mode, Rn+5=6 (DATA) is set.

$R_n + 5$	Operation mode

0: Rapid traverse (G0)
1: Cutting feed (G1)
2: Jog feed (+)
3: Jog feed (-)
4: Reference point return (+)
5: Reference point return (-)
6: Handle feed

The axis movement will not be affected by changing the operation mode, even while the axis is moving. The new operation mode is validated at the next start.

12.3.3.7 Feedrate

When the operation mode is cutting feed or jog feed (Rn + 5 = 1 to 3), the PLC axis feedrate is designated with a binary code.

R_n + 6 7	Feedrate

Designation value
1 to 1000000 mm/min. (0.1 inch/min.)

(Note 1) The feedrate designated in the parameters is used for the rapid traverse mode and reference point return mode.
(Note 2) The feedrate can be changed during axis movement. In that case, change using a direct feedrate data (Rn + 6, 7) is possible.

12.3.3.8 Movement Data

When the operation mode is rapid traverse or cutting feed, the movement data is designated with a binary code.

R_n + 8 9	Movement data

Designation value
0 to ±99999999 (0.001mm/0.0001inch)

(Note 1) The movement data is classified as follows by the absolute value command flag (bit 8) of the command signal.
Absolute value command flag = 0: Incremental value from the current position
Absolute value command flag = 1: Absolute value of the machine coordinate system
(Note 2) If the movement amount is changed during axis movement, the new movement amount will be validated at the next start.

12.3.3.9 Machine Position

The machine position output to the machine system is expressed. The machine position becomes the rfp (reference point) when the reference point is reached.

R_n + 10	Machine position (input unit)
11	

12.3.3.10 Remaining Distance

The remaining distance of the movement data output to the machine system is expressed.

R_n + 12	Remaining distance (input unit)
13	

12.3.4 Reference Point Return near Point Detection

Set the near point dog signal of the PLC axis reference point return for the following devices in the PLC.

Device No.		Signal name	
Y438	*PCD1	PLC axis	Reference point return near point detection 1
Y439	*PCD2	PLC axis	Reference point return near point detection 2
Y43A	*PCD3	PLC axis	Reference point return near point detection 3
Y43B	*PCD4	PLC axis	Reference point return near point detection 4
Y43C	*PCD5	PLC axis	Reference point return near point detection 5
Y43D	*PCD6	PLC axis	Reference point return near point detection 6
Y43E	*PCD7	PLC axis	Reference point return near point detection 7

(Note) The responsiveness when the dog signal is set in PLC middle-speed processing is worse than when set in PLC high-speed processing.

12.3.5 Handle Feed Axis Selection

The axis is designated for the following devices when handle feed is carried out with a PLC axis.

Device No.		Signal name
Y42D	HS1P	1st handle PLC axis valid
Y42E	HS2P	2nd handle PLC axis valid
Y42F	HS3P	3rd handle PLC axis valid

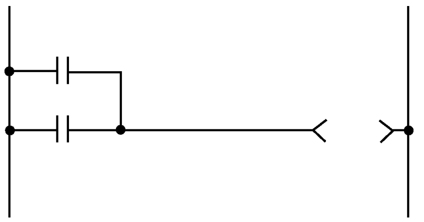
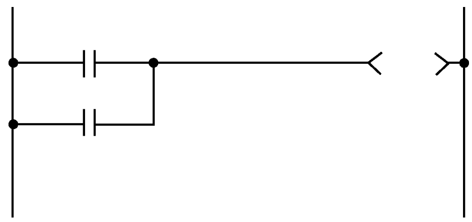
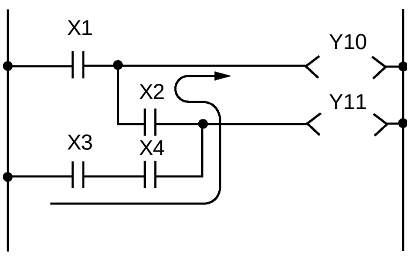
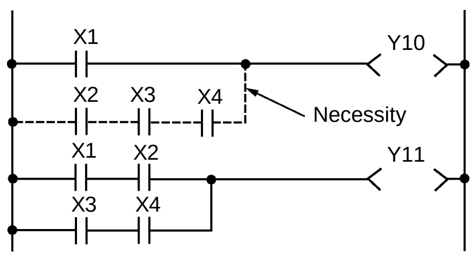
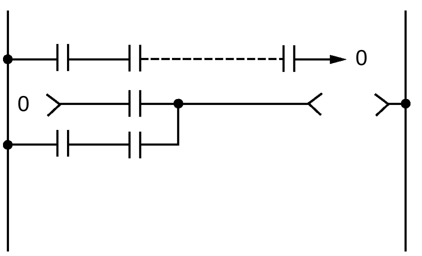
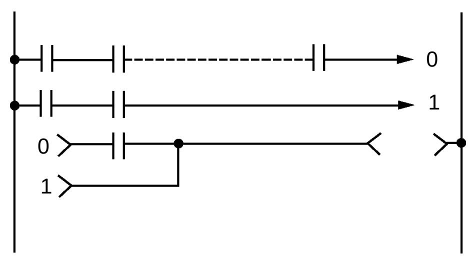
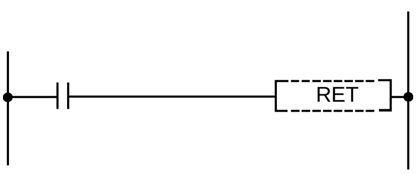
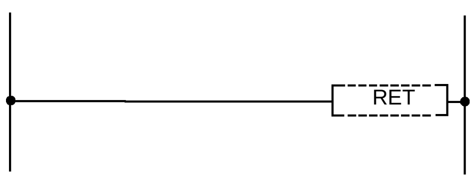
When Y42D to Y42F are ON, each handle changes to PLC axis dedication.
"Y740 to 744 and 747" for the 1st handle usually used in the control device are used for the axis selection of each handle.
("Y748 to 74C and 7AF" for the 2nd handle, "Y750 to 754 and 757" for the 3rd handle)

PLC axes are counted as PLC such as first axis and second axis. Therefore, if you will operate the first handle in the first axis of PLC, turn ON Y42D, Y740 to Y744 and Y747.

(Note) The handle feed magnification is also used for NC control axes.

Appendix 1. Example of Faulty Circuit

Wrong configurations of circuits are shown below. Correct the circuitry, if any.

Faulty circuit producing errors →	**Correct circuit**
(1) Circuit containing OR 	
(2) Rounding circuit  Whether or not the Y10 condition includes X3, X4 and X2 is unknown.	
(3) Modification of loopback circuit 	
(4) Presence of a contact before RET, FEND, or MCR circuit 	

Appendix 2. MELSEC QnA Series Command Lists

The following command lists are excerpts from the "QnACPU Programming Manual (Common Commands)" (model name: QNACPU-P (KYOUTU) 13J522).

In these lists, the commands "marked ✕" are unavailable for the C64 series. (When written from GX Developer to the C64 series, they are replaced by "NOP" commands.)

2.1 Sequence Commands

Contact commands

Classification	Command sign	Symbol
Contact	LD	
	LDI	
	AND	
	ANI	
	OR	
	ORI	
	LDP	
	LDF	
	ANDP	
	ANDF	
	ORP	
	ORF	

Coupling commands

Classification	Command sign	Symbol
Coupling	ANB	ANB
	ORB	ORB
	MPS	MPS
	MRD	MRD
	MPP	MPP
	INV	
	MEP	
	MEF	
	EGP	Vn
	EGF	Vn

Output commands

Classification	Command sign	Symbol
Output	OUT	
	SET	SET D
	RST	RST D
	PLS	PLS D
	PLF	PLF D
	FF	FF D
	~~DELTA~~	DELTA D
	~~DELTAP~~	DELTAP D

Shift commands

Classification	Command sign	Symbol
Shift	SFT	SFT D
	SFTP	SFTP D

Master control commands

Classification	Command sign	Symbol
Master control	MC	MC n D
	MCR	MCR n

End commands

Classification	Command sign	Symbol
Program end	FEND	FEND
	END	END

Other commands

Classification	Command sign	Symbol
Stop	STOP	STOP
No operation	NOP	
	NOPLF	NOPLF
	PAGE	PAGE n

2.2 Basic Commands

Comparison operation commands

Classification	Command sign	Symbol
16-bit data comparison	LD=	= S1 S2
	AND=	= S1 S2
	OR=	= S1 S2
	~~LD<>~~	<> S1 S2
	~~AND<>~~	<> S1 S2
	~~OR<>~~	<> S1 S2
	LD>	> S1 S2
	AND>	> S1 S2
	OR>	> S1 S2
	~~LD<=~~	<= S1 S2
	~~AND<=~~	<= S1 S2
	~~OR<=~~	<= S1 S2
	LD<	< S1 S2
	AND<	< S1 S2
	OR<	< S1 S2
	LD>=	>= S1 S2
	AND>=	>= S1 S2
	OR>=	>= S1 S2

Comparison operation commands (continued)

Classification	Command sign	Symbol
32-bit data comparison	LDD=	D= S1 S2
	ANDD=	D= S1 S2
	ORD=	D = S1 S2
	LDD<>	D<> S1 S2
	ANDD<>	D <> S1 S2
	ORD<>	D<> S1 S2
	LDD>	D> S1 S2
	ANDD>	D> S1 S2
	ORD>	D> S1 S2
	LDD<=	D<= S1 S2
	ANDD<=	D <= S1 S2
	ORD<=	D<= S1 S2
	LDD<	D< S1 S2
	ANDD<	D < S1 S2
	ORD<	D< S1 S2
	LDD>=	D >= S1 S2
	ANDD>=	D >= S1 S2
	ORD>=	D>= S1 S2

Comparison operation commands (continued)

Classification	Command sign	Symbol
Real number data comparison	LDE=	E= S1 S2
	ANDE=	E= S1 S2
	ORE=	E= S1 S2
	LDE<>	E<> S1 S2
	ANDE<>	E<> S1 S2
	ORE<>	E<> S1 S2
	LDE>	E> S1 S2
	ANDE>	E> S1 S2
	ORE>	E> S1 S2
	LDE<=	E<= S1 S2
	ANDE<=	E<= S1 S2
	ORE<=	E<= S1 S2
	LDE<	E< S1 S2
	ANDE<	E< S1 S2
	ORE<	E< S1 S2
	LDE>=	E>= S1 S2
	ANDE>=	E>= S1 S2
	ORE>=	E>= S1 S2

Comparison operation commands (continued)

Classification	Command sign	Symbol
Character string data comparison	LD$=	$= S1 S2
	AND$=	$= S1 S2
	OR$=	$= S1 S2
	LD$<>	$<> S1 S2
	AND$<>	$<> S1 S2
	OR$<>	$<> S1 S2
	LD$>	$> S1 S2
	AND$>	$> S1 S2
	OR$>	$> S1 S2
	LD$<=	$<= S1 S2
	AND$<=	$<= S1 S2
	OR$<=	$<= S1 S2
	LD$<	$< S1 S2
	AND$<	$< S1 S2
	OR$<	$< S1 S2
	LD$>=	$>= S1 S2
	AND$>=	$>= S1 S2
	OR$>=	$>= S1 S2
Block data comparison	BKCMP=	BKCMP= S1 S2 D n
	BKCMP<>	BKCMP<> S1 S2 D n
	BKCMP>	BKCMP> S1 S2 D n
	BKCMP<=	BKCMP<= S1 S2 D n
	BKCMP<	BKCMP< S1 S2 D n
	BKCMP>=	BKCMP>= S1 S2 D n
	BKCMP=P	BKCMP=P S1 S2 D n
	BKCMP<>P	BKCMP<>P S1 S2 D n
	BKCMP>P	BKCMP>P S1 S2 D n
	BKCMP<=P	BKCMP<=P S1 S2 D n
	BKCMP<P	BKCMP<P S1 S2 D n
	BKCMP>=P	BKCMP>=P S1 S2 D n

Arithmetic operation commands

Classification	Command sign	Symbol
BIN 16-bit addition/subtraction	+	+ S D
	+P	+P S D
	+	+ S1 S2 D
	+P	+P S1 S2 D
	−	− S D
	−P	−P S D
	−	− S1 S2 D
	−P	−P S1 S2 D
BIN 32-bit addition/subtraction	D+	D+ S D
	D+P	D+P S D
	D+	D+ S1 S2 D
	D+P	D+P S1 S2 D
	D−	D− S D
	D−P	D−P S D
	D−	D− S1 S2 D
	D−P	D−P S1 S2 D
BIN 16-bit multiplication/division	∗	∗ S1 S2 D
	∗P	∗P S1 S2 D
	/	/ S1 S2 D
	/P	/P S1 S2 D
BIN 32-bit multiplication/division	D∗	D∗ S1 S2 D
	D∗P	D∗P S1 S2 D
	D/	D/ S1 S2 D
	D/P	D/P S1 S2 D

Arithmetic operation commands (continued)

Classification	Command sign	Symbol
BCD 4-digit addition/subtraction	~~B+~~	B+ S D
	~~B+P~~	B+P S D
	B+	B+ S1 S2 D
	B+P	B+P S1 S2 D
	~~B−~~	B− S D
	~~B−P~~	B−P S D
	B−	B− S1 S2 D
	B−P	B−P S1 S2 D
BCD 8-digit addition/subtraction	~~DB+~~	DB+ S D
	~~DB+P~~	DB+P S D
	~~DB+~~	DB+ S1 S2 D
	~~DB+P~~	DB+P S1 S2 D
	~~DB−~~	DB− S D
	~~DB−P~~	DB−P S D
	~~DB−~~	DB− S1 S2 D
	~~DB−P~~	DB−P S1 S2 D
BCD 4-digit multiplication/division	B∗	B∗ S1 S2 D
	B∗P	B∗P S1 S2 D
	B/	B/ S1 S2 D
	B/P	B/P S1 S2 D
BCD 8-digit multiplication/division	~~DB∗~~	DB∗ S1 S2 D
	~~DB∗P~~	DB∗P S1 S2 D
	~~DB/~~	DB/ S1 S2 D
	~~DB/P~~	DB/P S1 S2 D

Arithmetic operation commands (continued)

Classification	Command sign	Symbol
Floating-point data addition/ subtraction	E+	E+ S D
	E+P	E+P S D
	E+	E+ S1 S2 D
	E+P	E+P S1 S2 D
	E−	E− S D
	E−P	E−P S D
	E−	E− S1 S2 D
	E−P	E−P S1 S2 D
Floating-point data multiplication/ division	E∗	E∗ S1 S2 D
	E∗P	E∗P S1 S2 D
	E/	E/ S1 S2 D
	E/P	E/P S1 S2 D
BIN block addition/ subtraction	BK+	BK+ S1 S2 D n
	BK+P	BK+P S1 S2 D n
	BK−	BK− S1 S2 D n
	BK−P	BK−P S1 S2 D n
Character string data coupling	$+	$+ S D
	$+P	$+P S D
	$+	$+ S1 S2 D
	$+P	$+P S1 S2 D
BIN data increment	INC	INC D
	INCP	INCP D
	DINC	DINC D
	DINCP	DINCP D
	DEC	DEC D
	DECP	DECP D
	DDEC	DDEC D
	DDECP	DDECP D

Data conversion commands

Classification	Command sign	Symbol
BCD conversion	BCD	BCD S D
	BCDP	BCDP S D
	DBCD	DBCD S D
	DBCDP	DBCDP S D

Data conversion commands (continued)

Classification	Command sign	Symbol
BIN conversion	BIN	BIN S D
	BINP	BINP S D
	DBIN	DBIN S D
	DBINP	DBINP S D
BIN -> floating-point conversion	FLT	FLT S D
	FLTP	FLTP S D
	DFLT	DFLT S D
	DFLTP	DFLTP S D
Floating-point -> BIN conversion	INT	INT S D
	INTP	INTP S D
	DINT	DINT S D
	DINTP	DINTP S D
BIN 16-bit <-> 32-bit conversion	DBL	DBL S D
	DBLP	DBLP S D
	WORD	WORD S D
	WORDP	WORDP S D
BIN -> gray code conversion	GRY	GRY S D
	GRYP	GRYP S D
	DGRY	DGRY S D
	DGRYP	DGRYP S D
Gray code -> BIN conversion	GBIN	GBIN S D
	GBINP	GBINP S D
	DGBIN	DGBIN S D
	DGBINP	DGBINP S D
2's complement	NEG	NEG D
	NEGP	NEGP D
	DNEG	DNEG D
	DNEGP	DNEGP D
	ENEG	ENEG D
	ENEGP	ENEGP D
Block conversion	BKBCD	BKBCD S D n
	BKBCDP	BKBCDP S D n
	BKBIN	BKBIN S D n
	BKBINP	BKBINP S D n

Data transfer commands

Classification	Command sign	Symbol
16-bit data transfer	MOV	MOV S D
	MOVP	MOVP S D
32-bit data transfer	DMOV	DMOV S D
	DMOVP	DMOVP S D
Floating-point data transfer	~~EMOV~~	EMOV S D
	~~EMOVP~~	EMOVP S D
Character string data transfer	~~$MOV~~	$MOV S D
	~~$MOVP~~	$MOVP S D
16-bit data NOT transfer	CML	CML S D
	CMLP	CMLP S D
32-bit data NOT transfer	DCML	DCML S D
	DCMLP	DCMLP S D
Block transfer	BMOV	BMOV S D n
	BMOVP	BMOVP S D n
Same data block transfer	FMOV	FMOV S D n
	FMOVP	FMOVP S D n
16-bit data change	XCH	XCH S D
	XCHP	XCHP S D
32-bit data change	DXCH	DXCH S D
	DXCHP	DXCHP S D
Block data change	~~BXCH~~	BXCH S D n
	~~BXCHP~~	BXCHP S D n
Upper/lower byte change	~~SWAP~~	SWAP D
	~~SWAPP~~	SWAPP D

Program branch commands

Classification	Command sign	Symbol
Jump	CJ	CJ Pn
	~~SCJ~~	SCJ Pn
	JMP	JMP Pn
	~~GOEND~~	GOEND

Program execution control commands

Classification	Command sign	Symbol
Interrupt disable	~~DI~~	DI
Interrupt enable	~~EI~~	EI
Interrupt disable/enable setting	~~IMASK~~	IMASK S
Return	~~IRET~~	IRET

I/O refresh commands

Classification	Command sign	Symbol
I/O refresh	~~RFS~~	RFS D n

Other useful commands

Classification	Command sign	Symbol
Up/down counter	~~UDCNT1~~	UDCNT1 S D n
	~~UDCNT2~~	UDCNT2 S D n
Teaching timer	~~TTMR~~	TTMR D n
Special timer	~~STMR~~	STMR S n D
Nearest access control	~~ROTC~~	ROTC S n1 n2 D
Ramp signal	~~RAMP~~	RAMP n1 n2 D1 n3 D2
Pulse density	~~SPD~~	SPD S n D
Pulse output	~~PLSY~~	PLSY n1 n2 D
Pulse width modulation	~~PWM~~	PWM n1 n2 D
Matrix input	~~MTR~~	MTR S D1 D2 n

2.3 Application Commands

Logical operation commands

Classification	Command sign	Symbol
AND	WAND	WAND S D
	WANDP	WANDP S D
	WAND	WAND S1 S2 D
	WANDP	WANDP S1 S2 D
	DAND	DAND S D
	DANDP	DANDP S D
	DAND	DAND S1 S2 D
	DANDP	DANDP S1 S2 D
	~~BKAND~~	BKAND S1 S2 D n
	~~BKANDP~~	BKANDP S1 S2 D n
OR	WOR	WOR S D
	WORP	WORP S D
	WOR	WOR S1 S2 D
	WORP	WORP S1 S2 D
	DOR	DOR S D
	DORP	DORP S D
	DOR	DOR S1 S2 D
	DORP	DORP S1 S2 D
	~~BKOR~~	BKOR S1 S2 D n
	~~BKORP~~	BKORP S1 S2 D n
Exclusive OR	WXOR	WXOR S D
	WXORP	WXORP S D
	WXOR	WXOR S1 S2 D
	WXORP	WXORP S1 S2 D
	DXOR	DXOR S D
	DXORP	DXORP S D
	DXOR	DXOR S1 S2 D
	DXORP	DXORP S1 S2 D
	~~BKXOR~~	BKXOR S1 S2 D n
	~~BKXORP~~	BKXORP S1 S2 D n

Logical operation commands (continued)

Classification	Command sign	Symbol
NOT-exclusive OR	WXNR	WXNR S D
	WXNRP	WXNRP S D
	WXNR	WXNR S1 S2 D
	WXNRP	WXNRP S1 S2 D
	DXNR	DXNR S D
	DXNRP	DXNRP S D
	DXNR	DXNR S1 S2 D
	DXNRP	DXNRP S1 S2 D
	~~BKXNOR~~	BKXNOR S1 S2 D n
	~~BKXNORP~~	BKXNORP S1 S2 D n

Rotation commands

Classification	Command sign	Symbol
Right rotation	ROR	ROR D n
	RORP	RORP D n
	RCR	RCR D n
	RCRP	RCRP D n
Left rotation	ROL	ROL D n
	ROLP	ROLP D n
	RCL	RCL D n
	RCLP	RCLP D n
Right rotation	DROR	DROR D n
	DRORP	DRORP D n
	DRCR	DRCR D n
	DRCRP	DRCRP D n
Left rotation	DROL	DROL D n
	DROLP	DROLP D n
	DRCL	DRCL D n
	DRCLP	DRCLP D n

Shift commands

Classification	Command sign	Symbol
n-bit shift	SFR	SFR | D | n
	SFRP	SFRP | D | n
	SFL	SFL | D | n
	SFLP	SFLP | D | n
1-bit shift	BSFR	BSFR | D | n
	BSFRP	BSFRP | D | n
	BSFL	BSFL | D | n
	BSFLP	BSFLP | D | n
1-word shift	DSFR	DSFR | D | n
	DSFRP	DSFRP | D | n
	DSFL	DSFL | D | n
	DSFLP	DSFLP | D | n

Bit processing commands

Classification	Command sign	Symbol
Bit set/reset	BSET	BSET | D | n
	BSETP	BSETP | D | n
	BRST	BRST | D | n
	BRSTP	BRSTP | D | n
Bit test	TEST	TEST | S1 | S2 | D
	TESTP	TESTP | S1 | S2 | D
	DTEST	DTEST | S1 | S2 | D
	DTESTP	DTESTP | S1 | S2 | D
Bit device Batch reset	BKRST	BKRST | S | n
	BKRSTP	BKRSTP | S | n

Data processing commands

Classification	Command sign	Symbol
Data search	SER	SER | S1 | S2 | D | n
	SERP	SERP | S1 | S2 | D | n
	DSER	DSER | S1 | S2 | D | n
	DSERP	DSERP | S1 | S2 | D | n
Bit check	SUM	SUM | S | D
	SUMP	SUMP | S | D
	DSUM	DSUM | S | D
	DSUMP	DSUMP | S | D
Decode	DECO	DECO | S | D | n
	DECOP	DECOP | S | D | n
Encode	ENCO	ENCO | S | D | n
	ENCOP	ENCOP | S | D | n
7-segment decode	SEG	SEG | S | D
	SEGP	SEGP | S | D
Dissociation • Association	DIS	DIS | S | D | n
	DISP	DISP | S | D | n
	UNI	UNI | S | D | n
	UNIP	UNIP | S | D | n
	NDIS	NDIS | S1 | D | S2
	NDISP	NDISP | S1 | D | S2
	NUNI	NUNI | S1 | D | S2
	NUNIP	NUNIP | S1 | D | S2
	WTOB	WTOB | S | D | n
	WTOBP	WTOBP | S | D | n
	BTOW	BTOW | S | D | n
	BTOWP	BTOWP | S | D | n

Data processing commands (continued)

Classification	Command sign	Symbol
Retrieval	MAX	MAX S D n
	MAXP	MAXP S D n
	MIN	MIN S D n
	MINP	MINP S D n
	DMAX	DMAX S D n
	DMAXP	DMAXP S D n
	DMIN	DMIN S D n
	DMINP	DMINP S D n
Sort	SORT	SORT S1 n S2 D1 D2 S2:Number of data blocks to be compared at a time. D1:Device to be forced ON at sort completion D2:Used by system
	DSORT	DSORT S1 n S2 D1 D2 S2:Number of data blocks to be compared at a time. D1:Device to be forced ON at sort completion D2:Used by system
Total value calculation	WSUM	WSUM S D n
	WSUMP	WSUMP S D n
	DWSUM	DWSUM S D n
	DWSUMP	DWSUMP S D n

Table operation commands

Classification	Command sign	Symbol
Table processing	FIFW	FIFW S D
	FIFWP	FIFWP S D
	FIFR	FIFR S D
	FIFRP	FIFRP S D
	FPOP	FPOP S D
	FPOPP	FPOPP S D
	FINS	FINS S D n
	FINSP	FINS S D n

Structuring commands

Classification	Command sign	Symbol
Repeat	FOR	FOR n
	NEXT	NEXT
	BREAK	BREAK D Pn
	BREAKP	BREAKP D Pn
Sub-routine program call	CALL	CALL Pn S1～Sn
	CALLP	CALLP Pn S1～Sn
	RET	RET
	FCALL	FCALL Pn S1～Sn
	FCALLP	FCALLP Pn S1～Sn
	ECALL	SORT * Pn S1~Sn * : Program name
	ECALLP	ECALLP * Pn S1~Sn * : Program name
	EFCALL	EFCALL * Pn S1~Sn * : Program name
	EFCALLP	EFCALLP * Pn S1~Sn * : Program name
	COM	COM
Fixed index qualification	IX	IX S Device qualification circuit IXEND
	IXEND	
	IXDEV	IXDEV
	IXSET	IXSET Pn D Designation of qualification value

2.4 Exclusive Commands

Exclusive commands (communication)

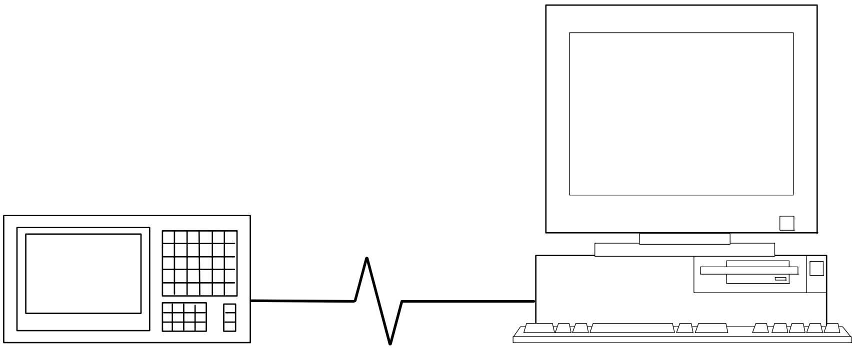
Classification	Command sign	Symbol
Transient transmission	OPEN	ZP.OPEN "Un" S1 S2 D1
	CLOSE	ZP.CLOSE "Un" S1 S2 D1
	BUFSND	ZP.BUFSND "Un" S1 S2 S3 D1
	BUFRCV	ZP.BUFRCV "Un" S1 S2 S3 D1 D2
	~~ERRCLR~~	
	~~ERRRD~~	
	~~UINI~~	
Intelligent function	FROM	FROM n1 n2 Dn D3
	TO	TO n1 n2 Dn D3
Transient command	READ	G.READ Un S1 S2 D1 D2
		GP.READ Un S1 S2 D1 D2
	SREAD	G.SREAD Un S1 S2 D1 D2 D3
		GP.SREAD Un S1 S2 D1 D2 D3
	WRITE	G.WRIT Un S1 S2 D1 D2
		GP.WRITE Un S1 S2 D1 D2
	SWRITE	G.SWRITE Un S1 S2 D1 D2 D3
		GP.SWRITE Un S1 S2 D1 D2 D3
	RIRD	G.RIRD Un S1 D1 D2
		GP.RIRD Un S1 D1 D2
	RIWT	G.RIWT Un S1 D1 D2
		GP.RIWT Un S1 D1 D2

Appendix 3. PLC Development Environment using GPPQ

The system configuration and respective tool function outlines of the PLC development environment are described here, especially the function of the "MELSEC QnA Series GPP Function Software Package (GPPQ)" that is the main tool.

3.1 System Configuration at PLC Development

PLC development is carried out with the CNC controller connected to an IBM PC-compatible personal computer, using an RS-232C or RS-422 cable. Each development tool is then started on the personal computer.



3.2 Development Tool Function Outline

3.2.1 CNVQ (data conversion software package)

The "CNVQ" is a tool that carries out file conversion of "GPPQ" data files and the following:

- Ladder list files output by the CLIST6L
- Alarms and operator messages created by the text editor
- Data files of commercially available spreadsheet software, word processors and editors

This tool is a software package (model name: SW0IVD-CNVQ) for various MELSEC support. Refer to the enclosed Operating Manual for function details.

3.2.2 LNKQ (sequence ladder generating connection function software package)

This tool supports more efficient sequence program development such as the use of modules for ladder programs.
This tool is a software package (model name: SW0IVD-LNKQ) for various MELSEC support. Refer to the enclosed Operating Manual for function details.

3.2.3 GPPQ (SW2IVD/NX-GPPQ type GPP Function Software Package)

The GPPQ is a programming software package (model name: SW2IVD-GPPQ) for the mainstream model "QnA Series" of the Mitsubishi programmable logic controller MELSEC Series.
The GPPQ is a powerful tool with the same functions as conventional MELDAS PLC development software (PLC4B), but strengthened. Furthermore, a monitoring function has been added via an RS-232C cable. Note that some functions related to the "QnA Series" characteristic functions cannot be used.

[Main Strengthened Function Points]

1) Operating environment

Operates on Windows 95 DOS window

2) Circuit editing functions

Overwrite/insert mode
Range-designated copy/paste
Device/command batch conversion
Multiple file simultaneous editing, copy/paste between files with windows
Timer/counter setting value batch change
Tag registration/jump

3) Comment functions

Three types of comments can be simultaneously edited or batch edited when editing the circuit.
• Device comment (explanatory statement for device unit)
• Note (note statement expressing meaning or application at head of each circuit block)
• Statement (explanatory statement making program easier to understand)

4) File control functions

Tree display of directory
Display of directory and file header statement

5) Printing functions

Commercial printer (ESC/P, PC-PR201H) compatible
Various information (circuit, list, device list, timer/counter setting value, header, footer, title) support

6) Help functions

Help screen display (at error occurrence, during operation, list) support
Menu guidance display

3.3 GPPQ Function Outline and Functions Supported by the C64 Series

The GPPQ is a programming software package (model name: SW1IVD-GPPQ) for the mainstream model "QnA Series" of the Mitsubishi programmable logic controller MELSEC series.
The GPPQ is a powerful tool with the same functions as conventional MELDAS PLC development software (PLC4B), but strengthened. Furthermore, a monitoring function has been added via an RS-232C cable. Note that some functions related to the "QnA Series" characteristic functions cannot be used.
The GPPQ functions explained here are those supported by the C64 Series in the "off-line functions" operated with the GPPQ independently and "on-line functions" carried out connected to the CNC controller.
Refer to the enclosed Operating Manual (off-line section and on-line section) for function details.

3.3.1 Function Support Conditions (general section)

The following shows a list of GPPQ outline functions supported by the C64 Series.
A ◎ mark indicates functions that can be used by the C64 Series. An × mark indicates that the function cannot be used because it is related to "QnA Series" characteristic functions. The function details during on-line are described in the next section.

List of general section functions (1)

◎ : Possible, × : Support not possible/not determined

Mode	Function	Support	Remarks
Initialization	New creation	◎	
	File new read	◎	
	PC new read	→	Refer to the List of on-line section functions
	File quit	◎	
	PC type change	×	Q4ACPU only
Option	Environment setting	◎	
	Display and operation option	◎	
	Startup setting	◎	
Circuit	Write	→	Refer to the List of on-line section functions
	Read	→	Refer to the List of on-line section functions
	Monitor	→	Refer to the List of on-line section functions
	Test	→	Refer to the List of on-line section functions
	Debug	→	Refer to the List of on-line section functions
	File access	→	Refer to the List of on-line section functions
	PC access	→	Refer to the List of on-line section functions
	Program search	→	Refer to the List of on-line section functions
	Comment display	→	Refer to the List of on-line section functions
	Program edit	→	Refer to the List of on-line section functions
	Monitor and test	→	Refer to the List of on-line section functions
	Window changeover	→	Refer to the List of on-line section functions
	Option	→	Refer to the List of on-line section functions
List	Write	◎	
	Read	◎	
	File access	◎	
	PC access	→	Refer to the List of on-line section functions
	Program search	◎	
	Comment display	◎	
	Program edit	◎	
	Window changeover	◎	
	Option	◎	

List of general section functions (2)

◎ : Possible, ✕ : Support not possible/not determined

Mode	Function	Support	Remarks
Parameter	PC name setting	✕	
	PC system setting	✕	
	PC file setting	✕	
	Device setting	✕	
	PC RAS setting	✕	
	I/O assignment	✕	
	MELSECNET (II,10)/Ethernet setting	✕	
	MELSECNET/MINI setting	✕	
	MELSECNET/J setting	✕	
	Miscellaneous setting	✕	
	Duplex PC setting	✕	
	SFC	✕	
	X/Y assignment confirmation	✕	
	File access	✕	
	PC access	✕	
	Parameter search	✕	
	Parameter edit	✕	
	Miscellaneous setting	✕	
	Window changeover	✕	
	Option	✕	
Device	Device value input	→	Refer to the List of on-line section functions
	Device initialization	→	Refer to the List of on-line section functions
	Buffer memory simulation	→	Refer to the List of on-line section functions
	File access	◎	
	PC access	→	Refer to the List of on-line section functions
	Device search	◎	
	Device display	◎	
	Device edit	→	Refer to the List of on-line section functions
	Window changeover	◎	
	Option	◎	
On-line	Drive name selection	→	Refer to the List of on-line section functions
	File name selection	→	Refer to the List of on-line section functions
	File access	→	Refer to the List of on-line section functions
	PC access	→	Refer to the List of on-line section functions
	Data search	→	Refer to the List of on-line section functions
	Trace	→	Refer to the List of on-line section functions
	Trace device edit	→	Refer to the List of on-line section functions
	Device test	→	Refer to the List of on-line section functions
	Window changeover	→	Refer to the List of on-line section functions
PC diagnosis	Diagnosis target selection	✕	
	Current error display	✕	
	Fault history display	✕	
	CPU message	✕	
	Unit detailed display	✕	
	File access	✕	
	PC access	→	Refer to the List of on-line section functions
	Data search	✕	
	Error display	✕	
	Network monitor and test	✕	
	Window changeover	✕	
	Option	✕	

List of general section functions (3)

◎ : Possible, ✕ : Support not possible/not determined

Mode	Function	Support	Remarks
Text creation	Device comment	◎	
	Pointer statement	◎	
	Interlinear statement	◎	
	Note	◎	
	Print statement	◎	
	File access	◎	
	PC access	→	Refer to the List of on-line section functions
	Comment search	◎	
	Comment edit	◎	
	Window changeover	◎	
	Option	◎	
Printer	Printer data setting	◎	
	Print execution	◎	
	Printer setting	◎	
	Paper size change	◎	
	Edit	◎	
	Window changeover	◎	
	Option	◎	
File maintenance	Machine name/File name selection	◎	
	File access	◎	
	PC access	→	Refer to the List of on-line section functions
	File search	◎	
	Display	◎	
	IC memory card	✕	
	Window changeover	◎	
	Option	◎	
Program generation	Label name definition	◎	
	File access	◎	
	Program generation	◎	
	Label display	◎	
	Display	◎	
	Label edit	◎	
	Window changeover	◎	
	Option	◎	
SFC		✕	
Quit		◎	

3.3.2 Function Support Conditions (on-line section)

The following shows a list of GPPQ on-line functions supported by the C64 Series.
A ◎ mark indicates functions that can currently be used by the C64 Series. An × mark indicates that the function cannot be used because it is related to "QnA Series" characteristic functions.

List of on-line section functions (1)

◎ : Possible, × : Support not possible/not determined

Mode	Function	Menu	Detailed function	Support	Remarks
Circuit mode	6.2 Circuit write	6.2.1 Write during run		◎	
	6.3 Circuit read	6.3.1 Step No. read		◎	
		6.3.2 Command read		◎	
		6.3.3 Device read		◎	
		6.3.4 Contact and coil designation read		◎	
		6.3.5 Statement and note designation read		◎	
		6.3.6 Last circuit designation read		◎	
	6.4 Monitor	6.4.1 Monitor screen common display			
			Monitor destination CPU name	◎	
			No. of Monitor program steps	◎	
			Monitor program name	◎	
			Scan time	◎	
			Monitor interval	◎	
			CPU operation status	◎	
			Monitor destination	◎	
			Flicker during communication	◎	
		6.4.2 Circuit monitor			
			Bit device monitor display	◎	
			Word device monitor display	◎	
			Digit designation monitor display	◎	
			Timer/counter monitor display	◎	
			Double word monitor display	◎	
			Index modification monitor display	◎	
			Monitor step changeover	◎	
			Changeover to next circuit block	◎	
			Changeover to previous circuit block	◎	
			Monitor stop	◎	
			Monitor column numeric value/No. of display levels changeover	◎	
		6.4.3 Device registration monitor			
			Bit device monitor display	◎	
			Word device monitor display	◎	
			Digit designation monitor display	◎	
			Timer/counter monitor display	◎	
			Double word monitor display	◎	
			Index modification monitor display	◎	
			16-bit integer designation	◎	
			32-bit integer designation	◎	
			Real number designation	×	
			Character designation	×	
			Registration device erase	◎	
		6.4.4 ON/OFF cause automatic search		◎	

List of on-line section functions (2)

◎ : Possible, × : Support not possible/not determined

Mode	Function	Menu	Detailed function	Support	Remarks
Circuit mode	6.5 Test	6.5.1 Monitor trigger stop			
			Stop at bit device rising edge	×	
			Stop at bit device falling edge	×	
			Stop at word device designation value	×	
			Monitor restart	×	
		6.5.2 Forced ON/OFF			
			Forced ON	×	
			Forced OFF	×	
		6.5.3 Current value change			
			16-bit change	×	
			32-bit change	×	
		6.5.4 Setting value change			
			Timer	×	
			Counter	×	
	6.6 Debug	6.6.1 Step execution		×	
		6.6.2 Partial execution		×	
	6.7 File menu	6.7.1 Read		◎	
		6.7.2 Write		◎	
		6.7.3 Compare		◎	
	6.8 PC menu	6.8.1 Read		◎	
		6.8.2 Write			
			Parameter	◎	
			Sequence program	◎	
			Entire range	◎	
			Range designation	×	
			Device comment	◎	
			Device default value	×	
			Simulation data	×	
			File register	×	
			Device memory	×	
			Entire range	×	
			Range designation	×	
		6.8.3 Compare		◎	
		6.8.4 New edit target read		×	
		6.8.5 List of files			
			File name	◎	
			Classification	◎	
			Size	◎	
			Date	◎	
			Time	◎	
			Header statement	◎	
			No. of files	◎	
			Continuous max. open capacity	◎	
			Entire open capacity	◎	
		6.8.6 Connection designation			
			Peripheral side interface	◎	
			PC side interface	×	Fixed
			Target CPU	×	Fixed
			Target memory	×	Fixed

List of on-line section functions (3)

◎ : Possible, ✕ : Support not possible/not determined

Mode	Function	Menu	Detailed function	Support	Remarks
Circuit mode	6.8 PC menu	6.8.7 Remote operation	CPU status display	◎	
			RUN	◎	
			STOP	◎	
			PAUSE	✕	
			STEP-RUN	✕	
			Latch clear	✕	
			Reset	✕	
			Operation during RUN, STEP-RUN	✕	
			Execution destination designation	✕	
			Execution unit designation	✕	
		6.8.8 Write option		◎	For write during run
	6.9 Display menu	6.9.1 Circuit display changeover			
			Device name display	◎	
			Comment display	◎	
			Statement display	◎	
			Display with note	◎	
	6.10 Monitor and test menu	6.10.1 Device batch monitor			
			Bit device monitor display	◎	
			Word device monitor display	◎	
			Bit and word monitor	◎	
			Bit multipoint monitor	◎	
			Word multipoint monitor	◎	
			Timer/counter mulipoint monitor	◎	
			Target device change	◎	
			Decimal/hexadecimal display changeover	◎	
			Integer/real number display changeover	◎	
			0 → F/F → 0 display changeover	◎	
			Forced ON	◎	
			Forced OFF	◎	
		6.10.2 Multiple device batch monitor			
			Bit device monitor display	◎	
			Word device monitor display	◎	
			Decimal/hexadecimal display changeover	◎	
			Integer/real number display changeover	◎	
			Registration device erase	◎	
			Forced ON	◎	
			Forced OFF	◎	
		6.10.3 ON/OFF cause automatic search		◎	
		6.10.4 Scan time measure		✕	
		6.10.5 Device registration monitor	Bit device monitor display	◎	
			Word device monitor display	◎	

List of on-line section functions (4)

◎ : Possible, × : Support not possible/not determined

Mode	Function	Menu	Detailed function	Support	Remarks
Circuit mode	6.10 Monitor and test menu	6.10.5 Device registration monitor	Digit designation monitor display	◎	
			Timer/counter monitor display	◎	
			Double word monitor display	◎	
			Index modification monitor display	◎	
			16-bit integer designation	◎	
			32-bit integer designation	◎	
			Real number designation	◎	
			Character designation	◎	
			Registration device erase	◎	
		6.10.6 Monitor and test menu			
			Constant monitoring	×	
			Word device condition	◎	
			Bit device condition	◎	
			Step No.	×	
		6.10.7 Monitor stop condition setting			
			Monitoring does not stop	◎	
			Word device condition	◎	
			Bit device condition	◎	
			Operation status	×	
		6.10.8 Monitor data registration/application		◎	
		6.10.9 Device test			
			Forced ON	◎	
			Forced OFF	◎	
			Forced ON/OFF reverse	◎	
			Current value change	◎	
		6.10.10 Sampling/monitoring trace		×	
		6.10.11 Step execution		×	
		6.10.12 Partial execution		×	
		6.10.13 Skip execution		×	
		6.10.14 Simulation		×	
		6.10.15 Monitor column numeric value display changeover			
			Numeric value display format	◎	
			No. of display level	◎	
		6.10.16 Program execution status monitor		×	
	6.11 Option menu	6.11.1 Monitor destination setting		×	
Para-meter mode	8.2 PC menu				
				→	Refer to 6.8 PC menu
Device mode	9.2 Device value input			×	
	9.3 Default value range setting			×	
	9.4 Buffer memory simulation			×	
	9.5 PC menu			→	Refer to 6.8 PC menu
	9.6 Edit menu	9.6.1 Simulation range		×	

List of on-line section functions (5)

◎ : Possible, × : Support not possible/not determined

Mode	Function	Menu	Detailed function	Support	Remarks
On-line mode	10.2 Drive name selection		Internal RAM	◎	Fixed
			IC memory card	×	Fixed
	10.3 File selection			◎	
	10.4 File menu	10.4.1 Read		◎	
		10.4.2 Write		◎	
		10.4.3 Compare		◎	
		10.4.4 New edit target read		◎	
		10.4.5 List of files		◎	
		10.4.6 Delete		◎	
		10.4.7 File quit		◎	
	10.5 PC menu	10.5.1 Read		◎	
		10.5.2 Write			
			Parameter	×	
			Sequence program	◎	
			Entire range	◎	
			Range designation	×	
			Device comment	◎	
			Device default value	×	
			Simulation data	×	
			File register	×	
			Device memory	×	
			Entire range	×	
			Range designation	×	
		10.5.3 Compare		◎	
		10.5.4 New edit target read		◎	
		10.5.5 List of files			
			File name	◎	
			Classification	◎	
			Size	◎	
			Date	◎	
			Time	◎	
			Header statement	◎	
			No. of files	◎	
			Continuous max. open capacity	◎	
			Entire open capacity	◎	
		10.5.6 Connection designation			
			Peripheral side interface	◎	
			PC side interface	×	Fixed
			Target CPU	×	Fixed
			Target memory	×	Fixed
		10.5.7 Remote operation	CPU status display	◎	
			RUN	◎	
			STOP	◎	
			PAUSE	×	
			STEP-RUN	×	
			Latch clear	×	
			Reset	×	
			Operation during RUN, STEP-RUN	×	
			Execution destination designation	×	
			Execution unit designation	×	

List of on-line section functions (6)

◎ : Possible, × : Support not possible/not determined

Mode	Function	Menu	Detailed function	Support	Remarks
On-line mode	10.5 PC menu	10.5.8 Write option		◎	For write during run
		10.5.9 Key word registration		×	
		10.5.10 Device memory operation		×	
		10.5.11 PC memory batch operation		×	
		10.5.12 Delete		◎	
		10.5.13 Header statement creation		×	
	10.6 Search menu	10.6.1 Head/end search		×	
		10.6.2 File search		×	
		10.6.3 Frequency search		×	
		10.6.4 Time search		×	
		10.6.5 Data search		×	
	10.7 Trace menu	10.7.1 Sampling trace		×	
		10.7.2 Monitoring trance		×	
		10.7.3 Status latch		×	
		10.7.4 Program trace		×	
	10.8 Edit menu	10.8.1 Cut/copy/paste		×	
	10.9 Test menu	10.9.1 Device test		×	
			Forced ON	◎	
			Forced OFF	◎	
			Forced ON/OFF reverse	◎	
			Current value change	◎	
	10.10 Window menu	10.10.1 Display window change		◎	
PC diagno-sis mode	11.2 Diagnosis target selection			×	
	11.3 Current error display			×	
	11.4 Fault history display			×	
	11.5 CPU message			×	
	11.6 Unit detailed display			×	
	11.7 File menu			×	
	11.8 PC menu			×	
	11.9 Search menu			×	
	11.10 Display menu			×	
	11.11 Network menu			×	
	11.12 Window menu			×	
	11.13 Option menu			×	
File mainte-nance mode					
	13.2 PC menu			→	Refer to 6.8 PC menu
	13.3 IC card menu			×	

3.4 Setup Procedure

3.4.1 Tool Setup Procedure

In the C64 Series PLC development environment, it is assumed that the various tools are used with an IBM PC-compatible personal computer. Prepare each tool so that it is IBM PC-compatible.
Refer to the enclosed Operating Manual for the setup and start procedures of each tool.

3.4.2 Connection Procedure

For the serial port connected with the CNC controller side, refer to "C64 Series Connection manual".
Connect the CNC controller and the IBM PC-compatible personal computer that will use the GPPQ, using an RS-232C serial cable equivalent to the RS-232C connector shown in the following connection diagram.
Note that the cables described in the connection diagrams in the GPPQ Operating Manual cannot be used.

(1) RS-232C connection

(a) When connecting with C64 directly

NC side (20-pin HALF PITCH)		Cable connection and signal direction	Personal computer side (9-pin D-SUB)	
Signal name	**Pin No.**		**Pin No.**	**Signal name**
SD	6	→	2	RD
RD	16	←	3	SD
ER (DTR)	18	→	6	DR (DSR)
		→	8	CS (CTS)
GND	1	↔	1	GND
GND	11	↔	5	GND

(b) Connection when using the intermediate cable for C64 only

NC side (25-pin D-SUB)		Cable connection and signal direction	Personal computer side (9-pin D-SUB)	
Signal name	**Pin No.**		**Pin No.**	**Signal name**
SD	2	→	2	RD
RD	3	←	3	SD
ER (DTR)	20	→	6	DR (DSR)
		→	8	CS (CTS)
GND	1	↔	1	GND
GND	7	↔	5	GND

(2) RS-422 connection

The cable for MELSEC only is used for the connection. Refer to the GPPQ operation manual for the details.

3.5 PLC Program Development Procedure

3.5.1 Precautions before Development

Pay careful attention to the following items before developing ladder programs using the GPPQ.

(1) PC Type Selection

The PC type must be set when newly creating programs, etc. Select the following CPU type when requesting PC type selection with the GPPQ. An error will occur during transfer of the ladder program to the C64 controller if another PC type is selected.

CPU type: Q4A

(2) Device Setting

Do not set the devices when developing in the ladder program for the C64 controller. Develop the program with the device settings (No. of points, etc.) left at their default values applied when GPPQ was started. The ladder program cannot be transferred to the C64 controller normally when it is developed with settings other than the default values.

Do not set the devices.

(3) PLC Commands

MELSEC-specific PLC commands cannot be used in the ladder program development for the C64 controller. Only M500 equivalent PLC commands and formats can be used.
The format, etc., changes with some commands. Refer to "3.7 Differences from the M500 PLC development environment" for details.

MELSEC-specific PLC commands cannot be used.

3.5.2 Ladder Transfer to the C64 Controller

The method of transferring a ladder to the C64 controller with the GPPQ (especially the restrictions and C64 characteristic operations) is explained below.

(1) Menu Selection

The screen is selected from the menu using the following GPPQ basic operation.

Mode screen → Alt (GRPH) [Menu] → 2 [PC] → PC menu window → 2 [Write] → PC write screen

(2) Screen Operation

The required options are selected and executed on the following screen.

```
[Write to PC]
Interface     RS232C    <--->    QnACPU
Target PC     Network : 0       Station : FF      PC Type : Q4A
Target Mem    Internal RAM        Title  [                                    ]
1. File   1. File Name [          ]     Title  [                                    ]
          2. Tgt   1. [ * ] Parameter
                   2. [ * ] Seq/SFC Prog      1. ( * ) Whole Range
                                              2. (   ) Step Range    [       ] – [       ]
                   3. [   ] Device Comment    3. (   ) Step Range    P [     ]  –  [     ]
                   4. [   ] Dev Init Value    4. (   ) Block Range   [       ] – [       ]
                   5. [   ] Simulation Data
                   6. [   ] File Register     1. ( * ) Whole Range
                                              2. (   ) Specify     ZR[      ] – [      ]
2. Device Mem      1. [   ] Internal          1. ( * ) Whole Range
                                              2. (   ) Specify Detail Range

                        Execute(Y)      Cancel(N)

                 Ctrl+L : filelist   Ctrl+D : directo   Space : sele Esc : close
```

[Restrictions]

The following options of [2. Tgt] under [1. File] can be selected.

[1. Parameter]
[2. Seq/SPC Prog] and [1. Whole Range]
[3. Device Comment]

Note that nothing will be transferred even if [1. Parameter] is selected.

(3) Operation during a Transfer Error

The ladder data is conversed to the ladder machine code characteristic to the C64 controller simultaneously with the ladder transfer from the GPPQ to the C64 controller. A conversion error will occur if devices or command formats are used that are not supported by the C64 Series.
During a conversion error, the following message appears at the bottom of the GPPQ screen after the C64 ladder machine code is converted to an "NOP code" and transferred to the last step.

Communication cannot be carried out with the PC. Error No.=4070

The following warning statement appears in the header statement item when the file having the conversion error is displayed with the PC file list function.

[List]					
:					
File	Type	Size	Data	Time	Title
PLCTEST	QnA Seq	12345	97-11-18	10:34	[∗∗WARNING: Checksum ERROR!!∗∗]
:					

If an attempt is made to run the PLC in this status, the following alarm message will appear on the NC side, and the PLC will not run.

Q01 Emergency stop PLC

(4) Error Step No. Confirmation Method

The error step can be confirmed using the PC compare function. The mismatched contents appear as in the following example when the PC comparison is executed. Refer to "Appendix 3.5.4 Ladder comparison with the C64 controller" for details.

[Sequence Program Diff List]					
	<Memory>			<PC>	
Step	Ins	I/O	Step	Ins	I/O
15	AND=	R4918	15	NOP	
		K106			
18	+	R4916	16	NOP	
		K10000			
		D87			
22	END		17	NOP	
	No Data exists.		18	+	R4916
					K10000
					D87
	No Data exists.		22	END	
PgDn : Next page					Esc: Close

3.5.3 Ladder Read from the C64 Controller

The method of reading a ladder from the M600 controller to the GPPQ is explained below.

(1) Menu Selection

The screen is selected from the menu using the following GPPQ basic operation.

Mode screen → Alt (GRPH) [Menu] → 2 [PC] → PC menu window → 1 [Read] → PC read screen

(2) Screen Operation

The file names are designated and executed on the following screen.

File name designation method

Input the file name stored in the read target memory.

Display the read target memory list screen by pressing Ctrl + D .

Display the file list screen by pressing Ctrl + L .

After selecting the file using ↑ and ↓ , display the file name selected in the file name designation column by pressing Enter .

```
[Read from PC]
------------------------------------------------------------------------------------------------
Interface      RS232C     <--->    QnACPU
Target PC      Network : 0        Station : FF       PC Type : Q4A
Target Mem     Internal RAM         Title   [                                       ]
1. File   1. File Name [          ]        Title   [                                       ]
          2. Tgt   1. [ * ] Parameter
                   2. [ * ] Seq/SFC Prog       1. ( * ) Whole Range
                                               2. (   ) Step Range     [        ] – [        ]
                   3. [   ] Device Comment     3. (   ) Step Range     P [      ]  –  [      ]
                   4. [   ] Dev Init Value     4. (   ) Block Range    [        ] – [        ]
                   5. [   ] Simulation Data
                   6. [   ] File Register      1. ( * ) Whole Range
                                               2. (   ) Specify      ZR[      ] – [      ]
2. Device Mem      1. [   ] Internal           1. ( * ) Whole Range
                                               2. (   ) Specify Detail Range

                        Execute(Y)      Cancel(N)
------------------------------------------------------------------------------------------------
               Ctrl+L : filelist   Ctrl+D : directo   Space : sele   Esc : close
```

[Restrictions]

The following options of [2. Tgt] under [1. File] can be selected.

[2. Seq/SPC Prog] and [1. Whole Range], [3. Device Comment]

3.5.4 Ladder Comparison with the C64 Controller

The method of comparing the ladders between the C64 controller and GPPQ is explained below.

(1) Menu Selection

The screen is selected from the menu using the following GPPQ basic operation.

Mode screen → [Alt] ([GRPH]) (Menu) → [2] (PC) → PC menu window → [3] (Compare) → PC compare screen

(2) Screen Operation

The file names are designated and executed on the following screen. The options that can be selected are the same as those of the ladder read.

File name designation method

Input the file name stored in the read target memory.

Display the read target memory list screen by pressing [Ctrl] + [D].

Display the file list screen by pressing [Ctrl] + [L].

After selecting the file using [↑] and [↓], display the file name selected in the file name designation column by pressing [Enter].

```
[PC compare]
----------------------------------------------------------------------------------
Interface      RS232C     <--->    QnACPU
Target PC      Network : 0        Station : FF      PC Type : Q4A
Target Mem     Internal RAM        Title  [                                        ]
1. File   1. File Name [          ]   Title  [                                     ]
          2. Tgt   1. [ * ] Parameter
      :
      :
      :

                          Execute(Y)     Cancel(N)
----------------------------------------------------------------------------------
                  Ctrl+L : filelist   Ctrl+D : directo   Space : sele   Esc : close
```

When the comparison contents do not match, a screen with the mismatched contents appears.

```
[Sequence Program Diff List]
----------------------------------------------------------------
             <Memory>                           <PC>
   Step    Ins      I/O               Step    Ins      I/O
    15     AND=     R4918              15     NOP
                    K106
    18     +        R4916              16     NOP
                    K10000
                    D87
    22     END                         17     NOP
           No Data exists.             18     +        R4916
                                                       K10000
                                                       D87
           No Data exists.             22     END
----------------------------------------------------------------
PgDn : Next page                                   Esc : Close
```

3.6 PLC-Related Data Development Procedure

The development procedure is explained here for alarm messages, operator messages, PLC switches, etc., as PLC related data.

3.6.1 PLC Related Data File Names

The PLC related data is controlled and stored in the categories shown below inside the C64 Series. Consequently, that data development is also carried out in each of the same categories.

[Important]

Be careful when transferring the message data. If message data that does not follow the file name regulations below is transferred, it will be mistaken for the PLC program, and the PLC program may be erased.

M1 xxxx. GPG

Extensions .... Automatically attached with the GPPQ (expresses file classification)

Random file name ..... User free designation

Data classification and language classification ..... user fixed designation
(M1: Message 1st language)

The data classification is distinguished by file name when transferring with the GPPQ.
The extensions express the file classification, and the two characters at the head express the data classification and language classification.
xxxx can be freely designated within eight characters, excluding extensions.

List of PLC related data

	Related data classification	File name	Remarks
1	PLC program (ladder)	zzzz.GPG	PLC program code
2	PLC program (comment)		Comment data for GPPQ
3	Program comment 1st language	C1xxxx.GCD zzzz.GCD	English contact/coil comment data (when the file name is the same as the PLC program code)
4	Program comment 2nd language	C2xxxx.GCD	Same as above (language data other than English)
5	Message 1st language	M1xxxx.GPG	Alarm messages/operator messages/PLC switches Other English message data
6	Message 2nd language	M2xxxx.GPG	Same as above (language data other than English)

∗ The PLC program and program comment are explained in "1.4 Creating PLC-Related Data".

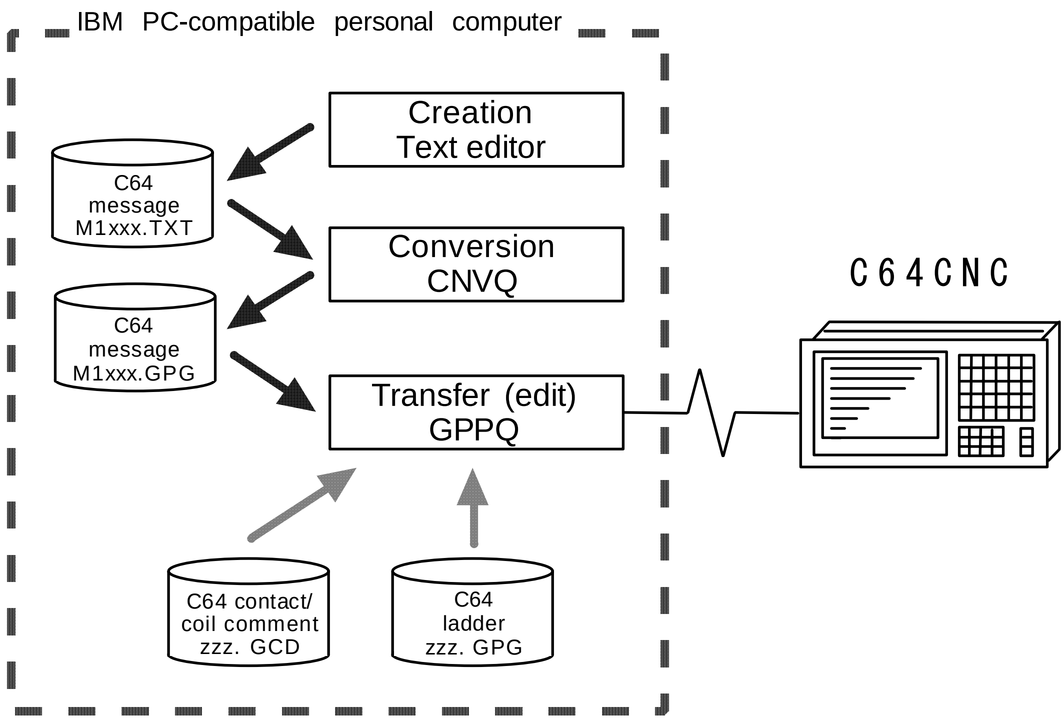
3.6.2 Development Procedure

The general development procedure for message date is explained below.

(1) Creation
The message data is described using a general text editor. The description method and format will be described later.

(2) Conversion
The conversion from text data to GPPQ data is carried out using the "CNVQ(data conversion software package)".

(3) Transfer
The message data is transferred to the C64 controller using the GPPQ, in the same manner as the ladder program.
The message data is handled as a PLC program interlinear comment, and can also be edited.



3.6.3 Message Data Description Method

The message data can be described as text data by a general text editor. Commercially available spreadsheet software can also be used if it conforms to the following description method.

(1) Description Format

Message data is classified into setting areas for setting each message and message areas for the respective message main unit. It is described in the following respective description format.

(a) Setting area

The message length and No. of messages is set for each message in the setting area. The message data region secured by the CNC controller can be adjusted to the most efficient status using these settings. The respective maximum values are set if nothing is set. (Refer to "(4) Precautions" for the maximum values.)

;$, message classification code, maximum message length, No. of messages [CR]

(b) Message area

The message area is described using the following description format.
The description format cannot be abbreviated. Comma(,) and [CR] must be described, even the message character string is blank.

Message classification	Description format
Alarm message	;A, index No., data register No., message character string [CR]
Operator message	;O, index No., data register No., message character string [CR]
PLC switch	;P, switch No. message character string[CR]
Comment message	;M, device, device No., message character string [CR]

Message classification code : One half-byte alphabetic character expressing each message classification
Index No. : Half-byte number (0 to No. of messages in the setting area - 1)
Switch No. : Half-byte number (0 to No. of messages in the setting area - 1)
Data register No. : Half-byte number
Device : Half-byte number (1 or 2)
Device No. : Half-byte number (0 to 10)
Message character string : Half-byte alphanumeric character, shift JIS Code 1 character, No. of characters in the setting area message length. Semicolons, commas, spaces and tabs can also be used. Note that the tab at the head of the message character string is ignored.
Semicolon(;) : Message data identification code
Comma(,) : Separator between each description (a comma only is used to leave a message character string blank)
[CR] : Line feed code, (CR/LF) or (LF).

(2) Description Method

The message data is described as text data by the following description format.

```
;# ladder ver1.0 '97.08.01              ··· Comment
;$, A, 32, 200
;$, 0, 40, 200                          ··· Setting area
;$, P, 14, 32
;$, M, 60, 20
NOPLF
;A, 0, 0, Emergency stop
;A, 11, 1, Spindle alarm
          :
          :
NOPLF                                   ··· Message area (alarm messages)
          :
          :
NOPLF
          :
          :
NOPLF                                   ··· Page break code
;0, 1, 9000, MELDAS 600LADDER Ver1.0
;0, 20, 9000, BND-400W000-A0            ··· Message area (operator messages)
          :
          :
NOPLF
;P, 1, Program restart
;P, 2, Automatic power OFF              ··· Message area (PLC switches)
          :
          :
NOPLF
;M, 1, 0, [Spindle]
;M, 1, 0, [Standby 1]                   ··· Message area (comments)
          :
          :
END                                     ··· End code
```

(a) Comment

Comments having a semicolon (;) at the head of the line, in a different format than described in "(1) Description format", are regarded as comments. These comments are handled as comment data in the GPPQ also, but are erased during the transfer to the C64 controller. An error will occur when descriptions not having a semicolon at the head of the line are converted into GPPQ data.

(b) Setting area

Each message is set here. This area must be described before the message area of the relevant message. That setting will be ignored if it is described during the relevant message or after description.

(c) Message area

Each message is grouped into similar messages and described. There is no description order in the respective messages, but the latter description is validated if there are descriptions with the same factors (index No., etc.).

(d) Page break code

A page break code is described at one or more places approx. every 20 lines in the setting area and message area. The message data may skip if there is no page break code.

(e) End code

An end code is described at the end of the description. Description after the end code are ignored.
An error will result during data conversion to the GPPQ if there is no end code.

(3) Details of comment message

The message of tool registration screen and the massage for load meter are defined in the comment message. The description format of comment message is shown as below.

;M, Device, Device No., Message character string

(a) Tool registration screen message

The message can be made up to 5 steps with maximum 8 characters. Even if the message is made above 5 steps, The characters of the first 5 steps are displayed in CNC Controller.
[Description format]

;M, 1, 0, Massage character string

(b) Load meter message

The message can be made up to 7 steps with maximum 40 characters.
• Message of 1st step is for the 1st system.
• Message of 2nd step is for the 2nd system.
• Message of 3rd step is for the 3rd system.
: :
• Message of 7th step is for the 7th system.
[Description format]

;M, 2, 0, Load meter character string

[Example of description]

```
NOPLF
;M,1,0,[Spindle]
;M,1,0,[Standby 1]
NOPLF
;M,2,0,Spindle load 1
;M,2,0,0        50          100
;M,2,0,         %
;M,2,0,|_|_|_|_|_|_|_|_|_|_|_|
;M,2,0,Z axis load 1
;M,2,0,0        50          100
;M,2,0,         %
;M,2,0,|_|_|_|_|_|_|_|_|_|_|_|
NOPLF
;M,2,0,Spindle load 2
;M,2,0,0        50          100
;M,2,0,         %
;M,2,0,|_|_|_|_|_|_|_|_|_|_|_|
;M,2,0,Z axis load 2
;M,2,0,0        50          100
;M,2,0,         %
;M,2,0,|_|_|_|_|_|_|_|_|_|_|_|
NOPLF
   :
```

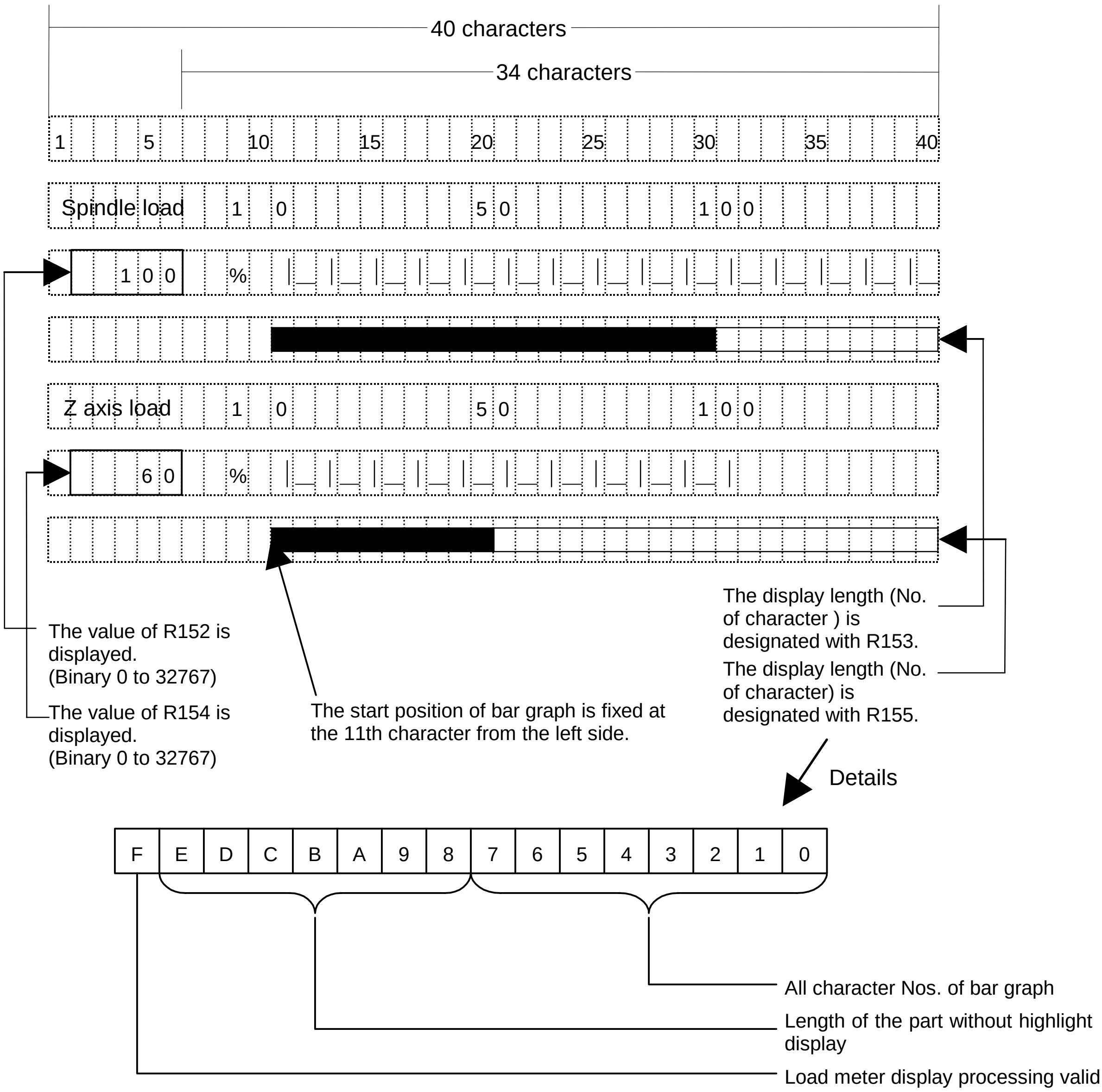
Tool registration screen message (max. 5 steps)

Load meter message (1 system)
The 10 characters of left side at first line are displayed.
The 30 characters of light side at first line are displayed.
The 10 characters of left side at second line.
(Only 3 characters of left side are valid.)
The 30 characters of light side at second line are displayed.
The 10 characters of left side at fourth line are displayed.
The 30 characters of light side at fourth line are displayed.
The 10 characters of left side at fifth line.
(Only 3 characters of left side are valid.)
The 30 characters of light side at fifth line are displayed.

Load meter message (2 system)

: :

(c) Load meter display



○ List of file register (R) for load meter display

		For $ 1	For $ 2
Load meter 1	For numerical value display	R152	R352
	For bar graph display	R153	R353
Load meter 2	For numerical value display	R154	R354
	For bar graph display	R155	R355

(4) Precautions

NO. of characters quantity limitations, handling of information other than settings, handling of information other than format.

(a) Message data maximum value
Processing will be carried out with the values at the following maximum values if the setting is not carried out in the setting area, or if the description position in the setting area is illegal.

Message classification	Max. message length	Max. No. of messages
Alarm messages	32 byte	512
Operator messages	60 byte	512
PLC switches	14 byte	32
Comments	60 byte	100

Full-byte data in the message character string is handled as two characters.

(b) When the setting value and message data do not match
When the message data contents (index/switch No. and message character string) overflows from the settings in the setting area, the data that overflowed is ignored.

3.6.4 Conversion to GPPQ Data

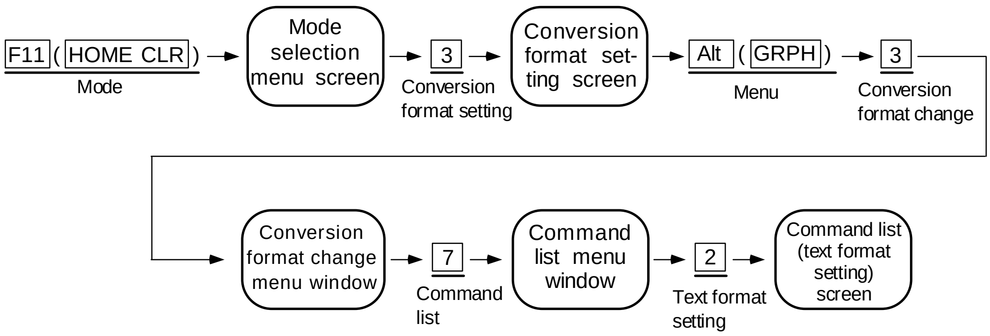
"CNVQ (data conversion software package)" is used in the conversion from text data to GPPQ data. It is also possible to convert to GPPQ data from spreadsheet software instead of text data. Refer to the Operating Manual for details.

(1) Initialization

After CNVQ is started, the project file (file for setting the drive/path name, system name, machine name, file name, etc.) is read or newly created.
Refer to the Operating Manual for the operation method.

(2) Conversion Format Setting

The screen is selected from the menu using the following CNVQ basic operation.

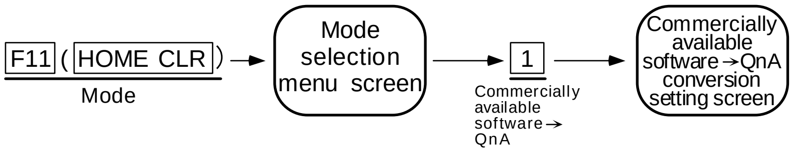


The conversion format is set on the following screen. Select the "Statement" item shown by the arrow (1) when converting message data.
Refer to the Operating Manual for details.

```
[Command list (text format setting)]

 1. Delimter
    1. ( * ) Tab       1. ( * ) Auto Adjust (1Tab [   8]Column)
                       2. (   ) Left Align
                       3. (   ) Specify      [   1]Tab
    2. ( * ) Space     1. ( * ) Auto Adjust
                       2. (   ) Left Align
                       3. (   ) Specify      [   1]Column
 2. Input Conditions
    1. (   ) Step #
    2. (   ) Device Comment
    3. ( * ) Statement                                          (1)
    4. (   ) Note
    5. (   ) Device Label
    6. (   ) Without NOP

                  Execute(Y)             Cancel(N)

                               Space : Select    Esc : Close
```

(3) Commercially Available Software QnA Conversion Setting

The screen is selected from the menu using the following CNVQ basic operation.

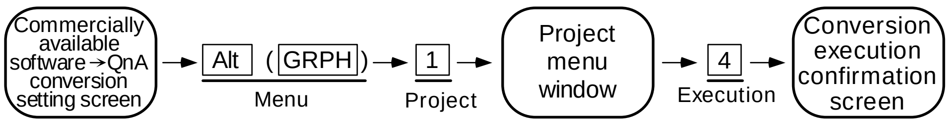


The conversion source/destination directory path and conversion file name, etc., are set on the following screen. Refer to the Operating Manual for details.

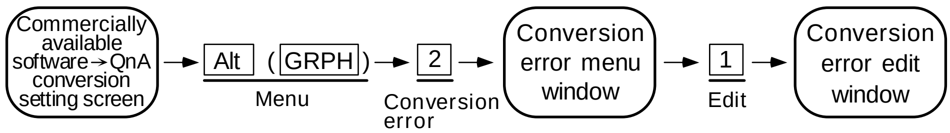
```
Other  --> QnA Convert            C:M600\TEST\SETUPM1                 <Ins>  F11:Mode
 Conversion Destination File
  Drive/Path[C:\GPPQ\USR
 Conversion Source File
  Drive/Path[C:\CNVQ\USR\M600\TEST

 #   Conv  SourceFile   DestFile                      Data #  FileType ConvType  Another
000   [  ] [M1TEST  ]  [M600\TEST\M1TEST          ]  [    1] <Text>  <   List   >   [  ]
001   [  ] [        ]  [                          ]  [     ] <     > <          >   [  ]
002   [  ] [        ]  [                          ]  [     ] <     > <          >   [  ]
003   [  ] [        ]  [                          ]  [     ] <     > <          >   [  ]
004   [  ] [        ]  [                          ]  [     ] <     > <          >   [  ]
005   [  ] [        ]  [                          ]  [     ] <     > <          >   [  ]
006   [  ] [        ]  [                          ]  [     ] <     > <          >   [  ]
007   [  ] [        ]  [                          ]  [     ] <     > <          >   [  ]
008   [  ] [        ]  [                          ]  [     ] <     > <          >   [  ]
009   [  ] [        ]  [                          ]  [     ] <     > <          >   [  ]
010   [  ] [        ]  [                          ]  [     ] <     > <          >   [  ]

 PgUp : Prev  Page   PgDn : Next Page           Ctrl+P :Drive/Path   Enter : Decide
```

(4) Conversion Execution

The screen is selected from the menu using the following CNVQ basic operation.



The conversion status appears at the bottom of the screen. If "There were n conversion errors" appears when the conversion is finished, the conversion error edit screen is displayed from the menu using the following CNVQ basic operation, and the error contents confirmed.

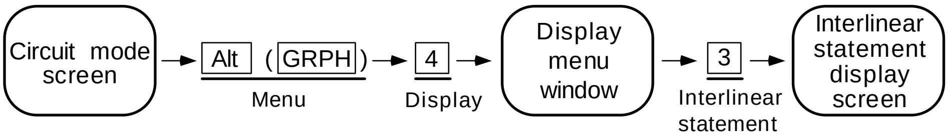


3.6.5 Operation with the GPPQ

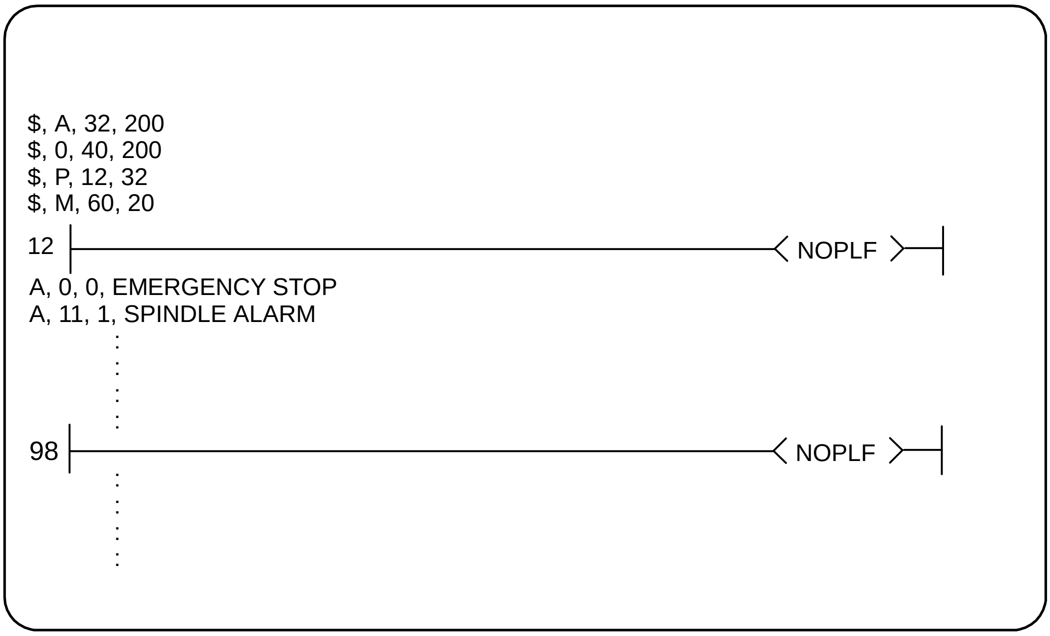
Message data in the GPPQ is handled as "Integrated interlinear statements" in the PLC program. "Integrated interlinear statements" are interlinear comments to help in understanding the PLC program. Interlinear comments transferred with the PLC program to the controller are called "integrated".
"Interlinear statements" can be displayed in the "Circuit mode", and edited on the "Interlinear statement edit screen".

(1) Interlinear Statement Display in the Circuit Mode

The screen is selected from the menu using the following GPPQ basic operation.

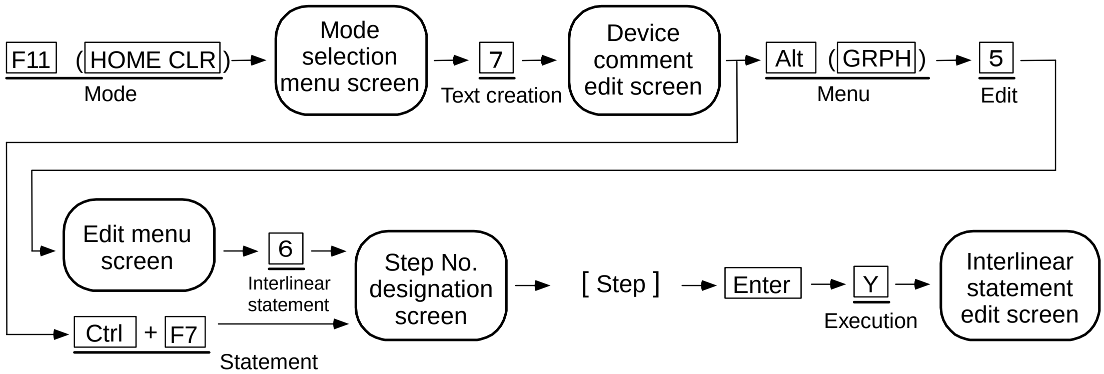


The following appears. At that time, the page break code and end code appear as normal ladder codes.



(2) Interlinear Statement Editing

The screen is selected from the menu using the following GPPQ basic operation.



The message data can be edited on the following screen. The GPPQ data can be converted to text data using the CNVQ to save the edited data as text data. Refer to the CNVQ Operating Manual for details.

```
Document Statmnt CPU: Q4A
  Step
  Level     1234567890123456789012345678901234567890123456789012345678901234
[     0]    [$, A, 32, 200                                                   ]
[     3]    [$, 0, 40, 200                                                   ]
[     6]    [$, P, 12, 32                                                    ]
[     9]    [$, M, 60, 20                                                    ]
[    12]    [A, 0, 0, EMERGENCY STOP                                         ]
[    15]    [A, 11, 1, SPINDLE ALARM                                         ]
[      ]    [       :                                                        ]
[      ]    [       :                                                        ]
[      ]    [       :                                                        ]
[      ]    [       :                                                        ]
[      ]    [       :                                                        ]
[      ]    [       :                                                        ]
[      ]    [       :                                                        ]
[      ]    [       :                                                        ]
[      ]    [       :                                                        ]
```

3.6.6 Transfer to the Controller

The following shows the method of transferring a message from the GPPQ to the CNC controller. The transfer method is the same as that of the ladder code. Ladder codes and message data are distinguished by their file names only.

(1) Menu Selection

The screen is selected from the menu using the following basic operation.

Mode screen → $\underbrace{\boxed{\text{Alt}}\ (\boxed{\text{GRPH}})}_{\text{Menu}}$ → $\underbrace{\boxed{2}}_{\text{PC}}$ → PC menu window → $\underbrace{\boxed{2}}_{\text{Write}}$ → PC menu screen

(2) Screen Operation

Transfer is executed in the following screen.
An example of transferring a message 1st language file "M1TEST.GPG" is shown here.

```
[Write to PC]
--------------------------------------------------------------------------------
Interface      RS232C      <--->     QnACPU
Target PC      Network : 0         Station : FF        PC Type : Q4A
Target Mem     Internal RAM          Title   [ ON BOARD S-RAM + (D-RAM)          ]
1. File    1. File Name [             ]       Title   [                           ]
           2. Tgt   1. [ * ] Parameter
                    2. [ * ] Seq/SFC Prog        1. ( * ) Whole Range
                                                 2. (   ) Step Range     [        ] – [        ]
                    3. [   ] Device Comment      3. (   ) Step Range     P [      ]  –  [      ]
                    4. [   ] Dev Init Value      4. (   ) Block Range    [        ] – [        ]
                    5. [   ] Simulation Data
                    6. [   ] File Register       1. ( * ) Whole Range
                                                 2. (   ) Specify       ZR[       ] – [       ]
2. Device Mem       1. [   ] Internal            1. ( * ) Whole Range
                                                 2. (   ) Specify Detail Range

                              Execute(Y)       Cancel(N)
--------------------------------------------------------------------------------
                         Ctrl+L : filelist   Ctrl+D : directo   Space : sele Esc:close
```

3.6.7 Reading and Comparing from the Controller

The following describes the method of reading and comparing a message from the CNC controller to the GPPQ. The method of reading and comparing is the same as that of ladder codes. Ladder codes and message data are distinguished by their file names only.

(1) Menu Selection/Screen Operation

Refer to the following sections for operation methods.
For reading : "3.5.3 Ladder read from the C64 controller"
For comparing : "3.5.4 Ladder comparison with the C64 controller"

(2) Message Read Format

The message description format was shown in "3.6.3 (1) Description Format", but there are no special rules concerning provision of descriptions in the setting area or the order of message description in the message area. For that reason, the description format may differ between transfer and reading of the message data.
The following shows the format during reading as the "Standard description format".

Standard description format of message data

```
Alarm message setting          … (1)
Operator message setting
PLC switch setting
Comment message setting
Alarm messages                 … (2)
Operator messages              … (3)
PLC switches                   … (4)
Comment messages               … (5)
NOPLF
END
```

(a) Setting area
The settings are described in order of alarm, operator, PLC switch and comment.
The maximum value is described if the setting is abbreviated.

(b) Alarm messages
Each message data is described in order of the index Nos.

(c) Operator messages
The same as the alarm messages.

(d) PLC switches
Each message data is described in order of the switch Nos.

(e) Comment messages
These messages are described in the same order as described before transfer.

(f) Others
- Spaces and tabs are not included before and after the comma(,) separating the message data factors.
- The message character string is handled the same as normal data even when blank.
- The NOPLF code between messages is described to the position to which the message data following the NOPLF code during transfer moved.

(3) Message Data Comparison

Message data comparison can be executed in the same manner as the ladder program, however, note that the target compared is the ladder code (NOPLF, END) only. Therefore, message data described as "Integrated interlinear statements" are not compared.
When comparing, read the data to the GPPQ side using the read function, then compare using the master file and file comparison function.

3.7 Differences From The M500 PLC Development Environment

3.7.1 PLC Commands

The command range that can be used in the MELSEC-QnA Series PLC program differs from the command range that can be used by the C64. Because of this, some commands that can be used by the C64 cannot be handled with the GPPQ. There are also commands that can be created with the GPPQ but cannot be used by the C64. When these are arranged, they are classified into the three following types.

- C64 commands that cannot be handled with the GPPQ
- C64 commands that the format differs from that of the GPPQ
- Commands that can be used with the GPPQ, but cannot be used by the C64

(1) C64 Commands that cannot be Handled with the GPPQ

Commands that cannot be handled with the GPPQ are substituted with alternate commands that can be handled with the GPPQ. Commands that can be alternated are shown in "Table of alternate command correspondence".
When some commands described in "Table of alternate command correspondence" as below are created with the GPPQ with the C64 command sign left as is, an error results and creation cannot be carried out. Create the commands using the GPPQ command sign described in the correspondence table.
When PLC programs containing alternate commands are written from the GPPQ to the C64, they are rewritten to the original MELDAS commands. Consequently, these commands appear in a different format in the GPPQ and C64 onboard ladder monitor when PLC programs containing alternate commands are monitored.

Table of alternate command correspondence

	C64 command		GPPQ command	
Classifi-cation	Command sign	Symbol	Command sign	Symbol
Bit	DEFR	─[DEFR D]─	ANDP	D ─┤↑├─
Average value	AVE	─[AVE S D n]┤	S.AVE	─[S.AVE S D n]┤
Carry flag set	STC	─[STC]┤	S.STC	─[S.STC]┤
Carry flag reset	CLC	─[CLC]┤	S.CLC	─[S.CLC]┤
ATC	ATC	─[ATC Kn Rn Rm]-< Mm >┤	S.ATC	─[S.ATC Kn Rn Rm Mm]┤
ROT	ROT	─[ROT Kn Rn Rm]-< Mm >┤	S.ROT	─[S.ROT Kn Rn Rm Mm]┤
TSRH	TSRH	─[TSRH Rm Rn]-< Mm >┤	S.TSRH	─[S.TSRH Rm Rn Mn]┤
DDBA	DDBA	─[DDBA Rn/Dn]-< Mm >┤	S.DDBA	─[S.DDBA Rn/Dn]┤
DDBS	DDBS	─[DDBS Rn]┤	S.DDBS	─[S.DDBS Rn]┤
BIT	LDBIT	├[BIT S1 n]─	LD<=	├[< = S1 n]─
	ANDBIT	─[BIT S1 n]─	AND<=	─[< = S1 n]─
	ORBIT	└[BIT S1 n]┘	OR<=	└[< = S1 n]┘
	LDBII	├[BII S1 n]─	LD<>	├[< > S1 n]─
	ANDBII	─[BII S1 n]─	AND<>	─[< > S1 n]─
	ORBII	└[BII S1 n]┘	OR<>	└[< > S1 n]┘
BCD decode	BDECO	─[BDECO S D n]─	S.BDECO	─[S.BDECO S D n]─

(2) Commands that the Format Differs from that of the GPPQ

"Commands used by inputting or outputting an accumulator (A0, A1)" in the conventional M500 Series did not have A0 or A1 designated in the command. However, in the C64 or GPPQ, the format is changed so that the general word register can be designated without fixing by A0 or A1 input or output.
Format-changed commands are shown in "Table of format-changed command correspondence".
When some commands described in "Table of format-changed command correspondence" are created with the GPPQ with the conventional command sign left as is, an error results and creation cannot be carried out. Create the commands using the command sign after format-changed described in the correspondence table.

Table of format-changed command correspondence

	M500 command		C64/GPPQ command		
Classification	Command sign	Symbol	Command sign	Symbol	Remarks
Right rotation	ROR	—[ROR n]┤	ROR	—[ROR D n]┤	∗1
	RCR	—[RCR n]┤	RCR	—[RCR D n]┤	
	DROR	—[DROR n]┤	DROR	—[DROR D n]┤	
	DRCR	—[DRCR n]┤	DRCR	—[DRCR D n]┤	
Left rotation	ROL	—[ROL n]┤	ROL	—[ROL D n]┤	
	RCL	—[RCL n]┤	RCL	—[RCL D n]┤	
	DROL	—[DROL n]┤	DROL	—[DROL D n]┤	
	DRCL	—[DRCL n]┤	DRCL	—[DRCL D n]┤	
Search	SER	—[SER S1 S2 n]┤	SER	—[SER S1 S2 D n]┤	∗2
Quantity of 1	SUM	—[SUM S]┤	SUM	—[SUM S D]┤	∗3

∗1 : D is the head No. of the rotation device (word 16-bit device)
∗2 : D is the head No. of the device that stores the search results (word bit device)
∗3 : D is the head No. of the device that stores the total No. of bits (word bit device)

(3) Commands that can be Used with the GPPQ, but cannot be Used by the C64

When commands that cannot be used by the M600 are written from the GPPQ to the C64, they are rewritten to "NOP" commands. (Note that commands described in "Table of alternate command correspondence" are rewritten to the corresponding MELDAS commands.)
The following may occur with some commands that cannot be used by the C64. A "verify error" will result when comparison is carried out after transmitting PLC programs containing commands not described in the alternate command correspondence table are transferred from the GPPQ to the C64.
Be careful when creating PLC programs with the GPPQ.
With some commands that cannot be used by the CNC, an error will not occur even when carrying out comparison of commands described in the alternate command correspondence table.
"Commands that can be created with the GPPQ, but cannot be used in the C64" are defined by the following expression.

"Commands that can be created with the GPPQ, but cannot be used in the C64"
= "All commands described in the QnA Programming Instruction Manual"
- ("All commands described in the C64 PLC Programming Instruction Manual"
+ "Table of alternate command correspondence"
+ "Table of format-changed command correspondence")

Refer to Appendix 2. List of MELSEC QnA Series Commands

3.7.2 PLC Messages

The contact/coil comments and other messages (alarm/operator messages, PLC switches, comments) are developed separately with their respective tools. The development tool for each message is shown in "List of message development tool comparisons".

List of message development tool comparisons

Message classification	M500	C64
Contact comments Coil comments	PLC development software (ladder section) (PLC4B) or PLC onboard (ONBD)	MELSEC QnA Series GPP function software package (GPPQ)
Alarm messages Operator messages PLC switches Comments		Text editor, etc. -> MELSEC QnA Series Data conversion software package (CNVQ)

(1) Contact/Coil Comments

Contact/coil comments created by the "Text creation mode" and "Device comment edit" of the "MELSEC QnA Series GPP function software package (GPPQ)" are saved as a file or transferred to the C64 by the RS-232C cable. The No. of character and quantity limitations are the same specifications as those of the M500.

(2) Other Messages (alarm/operator messages, PLC switches, comments)

With other messages, the described text file is converted with the designated format by the "MELSEC QnA Series data conversion software package (CNVQ)", and then that is transferred with the GPPQ to the C64 via the RS-232C cable.

Revision History

Date of revision	Manual No.	Revision details
Feb. 2002	BNP-B2309A	First edition created.
May 2004	BNP-B2309C	• The section "1. GX Developer" is revised wholly. • The following commands were added in according to the software version D; FROM, TO, OPEN, CLOSE, BUFSND, BUFRCV, READ, SREAD, WRITE, SWRITE, RIRD, RIWT • The explanations of following commands were revised. CALL/CALLP, BMOV/BMOVP, S.DECO, ENCO/ENCOP, FF • Number of steps for some commands such as "FF" or "××p" (with leading edge) were changed. • Mistakes were corrected.
Dec. 2004	BNP-B2309D	• The following commands were added. LD<=,AND<=,OR<=,LD<>,AND<>,OR<> • The section "2.1.2 Cautions on high-speed processing programming" was added in "2.1 PLC Processing Program Level and Operation". • The explanation was added in "12. PLC Axis Control". • Mistakes were corrected.

Notice

Every effort has been made to keep up with software and hardware revisions in the contents described in this manual. However, please understand that in some unavoidable cases simultaneous revision is not possible.
Please contact your Mitsubishi Electric dealer with any questions or comments regarding the use of this product.

Duplication Prohibited

This instruction manual may not be reproduced in any form, in part or in whole, without written permission from Mitsubishi Electric Corporation.

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE : MITSUBISHI DENKI BLDG., 2-2-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN

MODEL	MC6/C64
MODEL CODE	008-185
Manual No.	BNP-B2309D(ENG)

Specifications subject to change without notice.
(0412) MEE
Printed in Japan on recycled paper.