

CNC

MELDAS 60/60S Series

PLC PROGRAMMING MANUAL (LADDER SECTION WITH MELSEC TOOL)



MELSEC is registered trademark of Mitsubishi Electric Corporation.

Other company and product names that appear in this manual are trademarks or registered trademarks of the respective company.

Introduction

These specifications are the programming manual used when creating the sequence program for the MELDAS 60/60S Series with the onboard PLC development tool or PLC development software.

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.

*The "MELDAS60 Series" includes the M64A, M64, M65, M66 and M65V.

*The "MELDAS60S Series" includes the M64AS, M64S, M65S and M66S.

Details described in this manual

CAUTION

-  For items described in "Restrictions" or "Usable State", the instruction manual issued by the machine tool builder takes precedence over this manual.
-  Items not described in this manual 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 tool builder before starting use.
-  Refer to the Instruction Manual issued by each machine tool builder for details in each machine tool.
-  Some screens and functions may differ or may not be usable depending on the NC version.

General precautions

- (1) This Instruction Manual does not explain the operation procedures for programming the sequence program with onboard or personal computer. Refer to the related material listed below for details.

| | | | |
|--------------------------|---|-------|-----------|
| MELDAS 60/60S Series | PLC Onboard Instruction Manual | | BNP-B2213 |
| MELDAS 60/60S Series | PLC Program Development Manual (Personal Computer Section) | | BNP-B2215 |
| MELDAS 60/60S Series | PLC Interface Manual | | BNP-B2211 |
| MELDAS 600, 60/60 Series | PLC Development Software Manual (MELSEC Tool Section) | | BNP-B2252 |

Precautions for Safety

Always read the specifications issued by the machine tool builder, 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 tool builder 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 tool builder before starting use.
-  Refer to the Instruction Manual issued by each machine tool builder 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.

CONTENTS

| | |
|--|-----------|
| 1. System Configuration | 1 |
| 1.1 System Configuration for PLC Development | 1 |
| 1.2 User PLC (Ladder) Development Procedure | 2 |
| 2. PLC Processing Program | 3 |
| 2.1 PLC Processing Program Level and Operation | 3 |
| 2.2 User Memory Area Configuration | 3 |
| 3. Input/Output Signals | 4 |
| 3.1 Input/Output Signal Types and Processing | 4 |
| 3.2 Handling of Input Signals Designated for High-Speed Input..... | 5 |
| 3.3 High-Speed Input/output Designation Method..... | 6 |
| 4. Parameters 7 | |
| 4.1 PLC Constants | 7 |
| 4.2 Bit Selection Parameters | 8 |
| 5. Explanation of Devices | 12 |
| 5.1 Devices and Device Numbers | 12 |
| 5.2 Device List..... | 12 |
| 5.3 Detailed Explanation of Devices | 13 |
| 5.3.1 Input/output X, Y | 13 |
| 5.3.2 Internal Relays M and F, Latch Relay L..... | 14 |
| 5.3.3 Special Relays SM..... | 14 |
| 5.3.4 Timer T | 15 |
| 5.3.5 Counter C | 17 |
| 5.3.6 Data Register D | 17 |
| 5.3.7 File Register R | 18 |
| 5.3.8 Index Registers Z..... | 19 |
| 5.3.9 Nesting N..... | 20 |
| 5.3.10 Pointer P | 21 |
| 5.3.11 Decimal Constant K..... | 22 |
| 5.3.12 Hexadecimal Constant H | 22 |
| 6. Explanation of Commands | 23 |
| 6.1 Command List | 23 |
| 6.1.1 Basic Commands..... | 23 |
| 6.1.2 Function Commands..... | 24 |
| 6.1.3 Exclusive commands | 30 |
| 6.2 Command Formats..... | 31 |
| 6.2.1 How to Read the Command Table..... | 31 |
| 6.2.2 No. of Steps | 32 |
| 6.2.3 END Command..... | 33 |
| 6.2.4 Index Ornament | 33 |
| 6.2.5 Digit Designation..... | 34 |
| 7. Basic Commands | |
| (LD, LDI, AND, ANI, OR, ORI, ANB, ORB) | 37 |

| | |
|---|------------|
| 8. Function Commands | |
| (=, >, <, +, -, *, /, BCD, BIN, MOV) | 69 |
| 9. Exclusive Commands | 186 |
| 9.1 ATC Exclusive Command | 187 |
| 9.1.1 Outline of ATC Control | 187 |
| 9.1.2 ATC Operation | 187 |
| 9.1.3 Explanation of Terminology | 187 |
| 9.1.4 Relationship between Tool Registration Screen and Magazines | 188 |
| 9.1.5 Use of ATC and ROT Commands | 189 |
| 9.1.6 Basic Format of ATC Exclusive Command | 190 |
| 9.1.7 Command List | 191 |
| 9.1.8 Control Data Buffer Contents | 191 |
| 9.1.9 File Register (R Register) Assignment and Parameters | 192 |
| 9.1.10 Details of Each Command | 194 |
| 9.1.11 Precautions for Using ATC Exclusive Instructions | 203 |
| 9.1.12 Examples of Tool Registration Screen | 203 |
| 9.1.13 Display of Spindle Tool and Standby Tool | 205 |
| 9.2 S.ROT Commands | 206 |
| 9.2.1 Command List | 206 |
| 9.3 Tool Life Management Exclusive Command | 212 |
| 9.3.1 Tool Life Management System | 212 |
| 9.3.2 Tool Command System | 212 |
| 9.3.3 Spare Tool Selection System | 213 |
| 9.3.4 Interface | 213 |
| 9.3.5 User PLC Processing When the Tool Life Management Function Is Selected | 214 |
| 9.3.6 Examples of Tool Life Management Screen | 222 |
| 9.4 DDB (Direct Data Bus) ... Asynchronous DDB | 223 |
| 9.4.1 Basic Format of Command | 223 |
| 9.4.2 Basic Format of Control Data | 223 |
| 9.5 External Search | 226 |
| 9.5.1 Function | 226 |
| 9.5.2 Interface | 226 |
| 9.5.3 Search Start Instruction | 228 |
| 9.5.4 Timing Charts and Error Causes | 228 |
| 9.5.5 Sequence Program Example | 230 |
| 9.6 Chopping | 231 |
| 9.6.1 Chopping operation start | 232 |
| 9.6.2 Chopping operation stop | 234 |
| 9.6.3 Chopping compensation | 235 |
| 9.6.4 Chopping interface | 238 |
| 9.6.5 Parameters (DDB function instructions from PLC) | 239 |
| 9.6.6 Example of chopping control by program command | 245 |

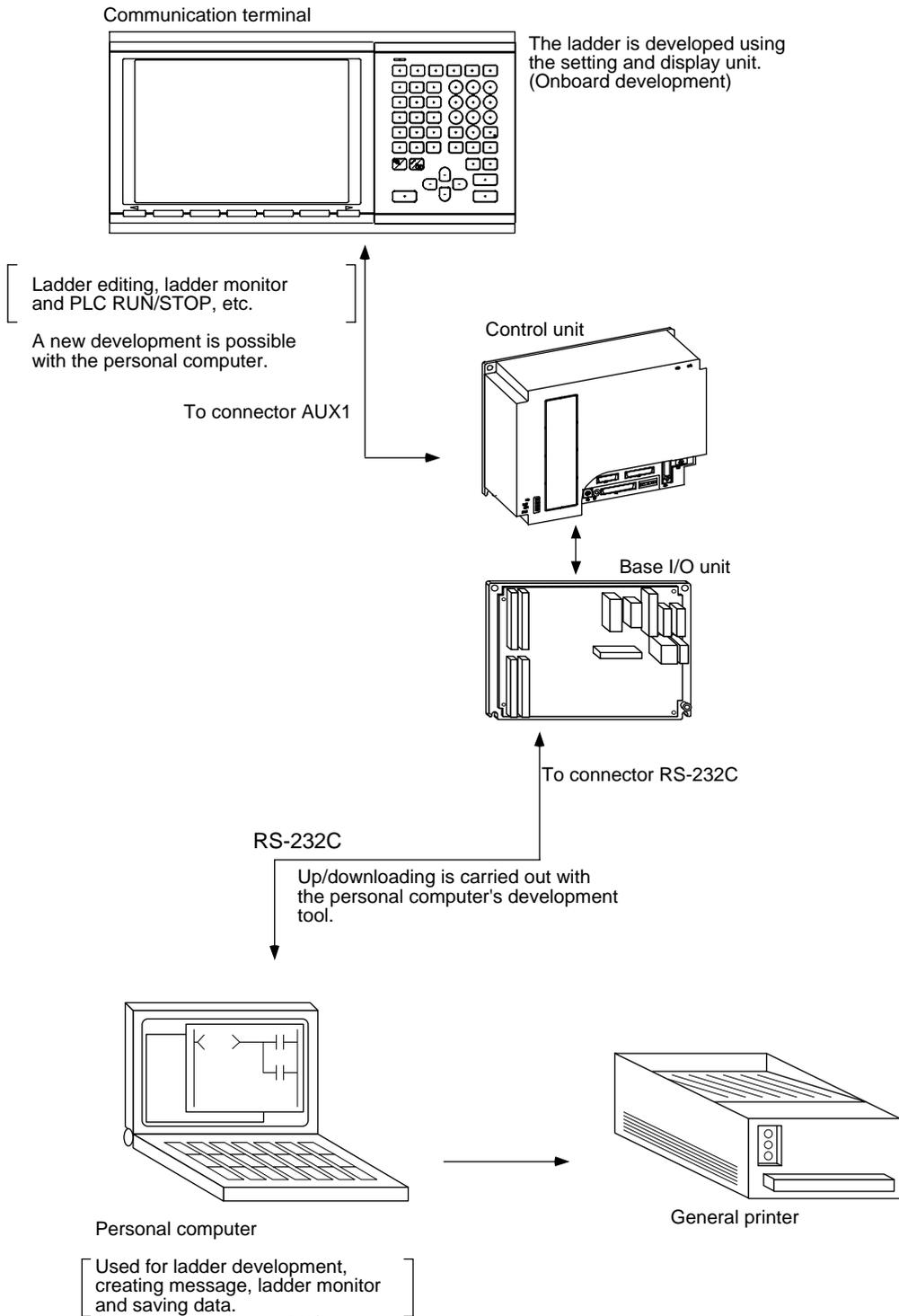
| | | |
|------------|---|------------|
| 9.7 | CC-Link | 248 |
| 9.7.1 | Input output signal | 249 |
| 9.7.2 | Communication data flow..... | 250 |
| 9.7.2.1 | Remote input and remote output (Master station ← local station/remote device station/remote I/O station) | 252 |
| 9.7.2.2 | Remote output and remote input (Master station → local station/remote device station/remote I/O station) | 253 |
| 9.7.3 | Automatic Refresh | 254 |
| 9.7.4 | Occupied number of stations of the system and settable range of the device | 256 |
| 9.7.5 | Transient function | 257 |
| 9.7.5.1 | Transient instruction (RIRD instruction) | 257 |
| 9.7.5.2 | Transient Instructions (RIWT instruction) | 259 |
| 9.7.5.3 | Transient instruction program example and error | 260 |
| 9.7.6 | Others..... | 261 |
| 10. | PLC Help Function..... | 262 |
| 10.1 | Alarm Message Display | 263 |
| 10.1.1 | Interface..... | 263 |
| 10.1.2 | Screen Display..... | 265 |
| 10.1.3 | Message Creation..... | 266 |
| 10.1.4 | Parameters | 269 |
| 10.2 | Operator Message Display | 271 |
| 10.2.1 | Interface..... | 271 |
| 10.2.2 | Operator Message Preparation..... | 272 |
| 10.2.3 | Operator Message Display Validity Parameter | 272 |
| 10.3 | PLC Switches | 273 |
| 10.3.1 | Explanation of Screen..... | 273 |
| 10.3.2 | Explanation of Operation | 274 |
| 10.3.3 | Signal Processing | 275 |
| 10.3.4 | Switch Name Preparation | 279 |
| 10.4 | Key Operation by User PLC | 280 |
| 10.4.1 | Key Data Flow | 280 |
| 10.4.2 | Key Operations That Can Be Performed | 280 |
| 10.4.3 | Key Data Processing Timing..... | 281 |
| 10.4.4 | Layout of Keys on Communication Terminal | 282 |
| 10.4.5 | List of Key Codes..... | 283 |
| 10.5 | Load Meter Display..... | 285 |
| 10.5.1 | Interface..... | 285 |
| 10.6 | External Machine Coordinate System Compensation | 287 |
| 10.7 | User PLC Version Display | 288 |
| 10.7.1 | Interface..... | 288 |
| 11. | PLC Axis Control | 290 |
| 11.1 | Outline | 290 |
| 11.2 | Specifications | 290 |
| 11.2.1 | Basic Specifications..... | 290 |
| 11.2.2 | Other Restrictions | 291 |

| | |
|--|------------|
| 11.3 PLC Interface | 292 |
| 11.3.1 S.DDBS Function Command | 292 |
| 11.3.2 Control Information Data..... | 293 |
| 11.3.3 Control Information Data Details..... | 294 |
| 11.3.3.1 Commands | 294 |
| 11.3.3.2 Status | 295 |
| 11.3.3.3 Alarm No..... | 302 |
| 11.3.3.4 Control Signals (PLC axis control information data) | 303 |
| 11.3.3.5 Axis Designation | 305 |
| 11.3.3.6 Operation Mode..... | 305 |
| 11.3.3.7 Feedrate | 306 |
| 11.3.3.8 Movement Data | 306 |
| 11.3.3.9 Machine Position | 307 |
| 11.3.3.10 Remaining Distance..... | 307 |
| 11.3.4 Reference Point Return Near Point Detection | 308 |
| 11.3.5 Handle Feed Axis Selection..... | 309 |
| 12. Appendix | 310 |
| 12.1 Example of Faulty Circuit..... | 310 |

1. System Configuration

1.1 System Configuration for PLC Development

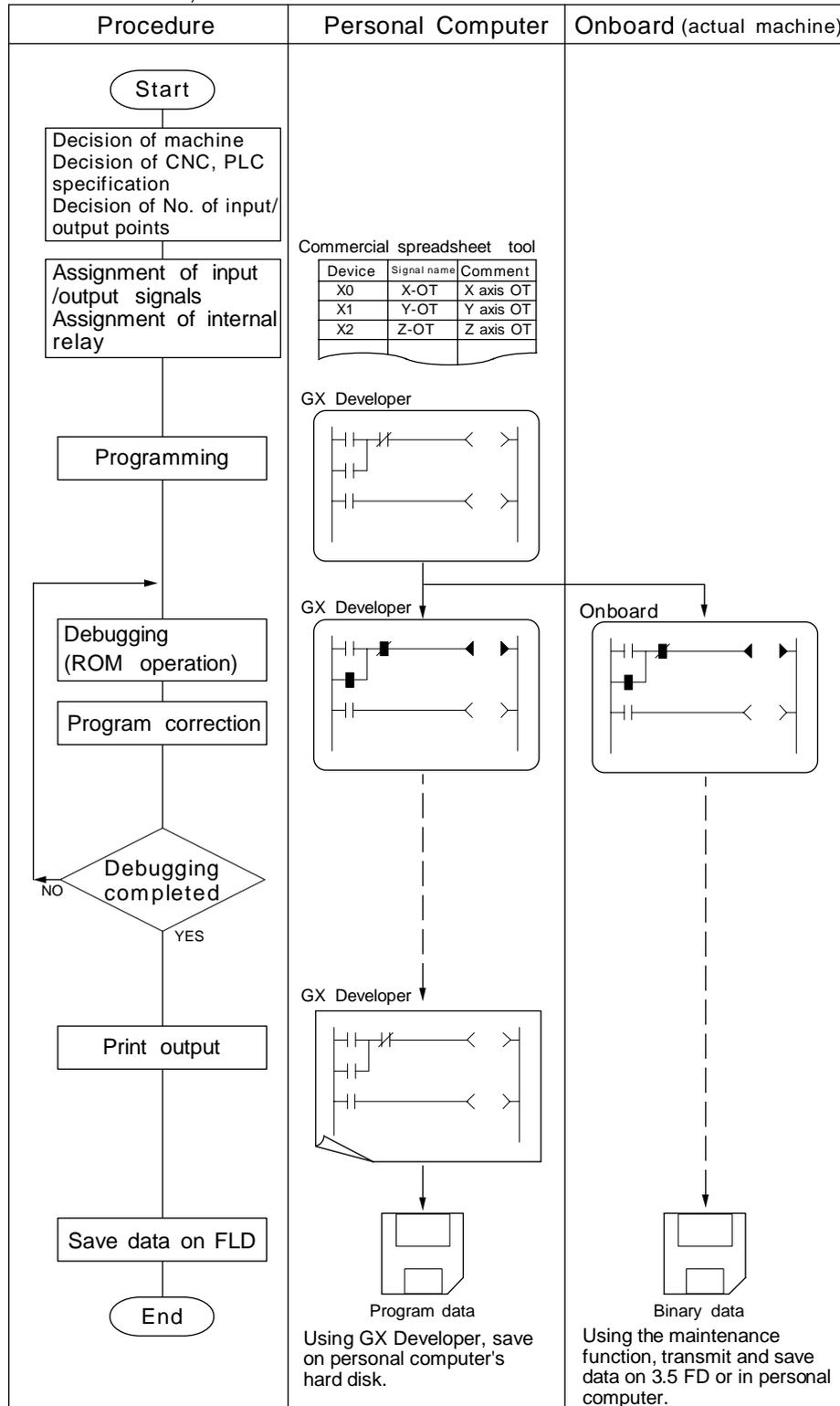
The system configuration for PLC development is shown below.



(Note) Refer to the "MELDAS 60/60S Series PLC Onboard Instruction Manual BNP-B2213" for edition using the communication terminal (onboard edition), and the "MELDAS 600/60/60S Series PLC Development Software Manual BNP-B2252" for development using the personal computer.

1.2 User PLC (Ladder) Development Procedure

The procedure for creating the user PLC, used to control the control target (machine) built into the control unit, is shown below.



The data created with the commercial spreadsheet tool can be used as the ladder comment data.

Using GX Developer, execute programming. After programming completed, download using RS-232C.

Using the online function of GX Developer or onboard, execute monitoring and correcting.

Printout to a commercial printer connected with the personal computer from GX Developer.

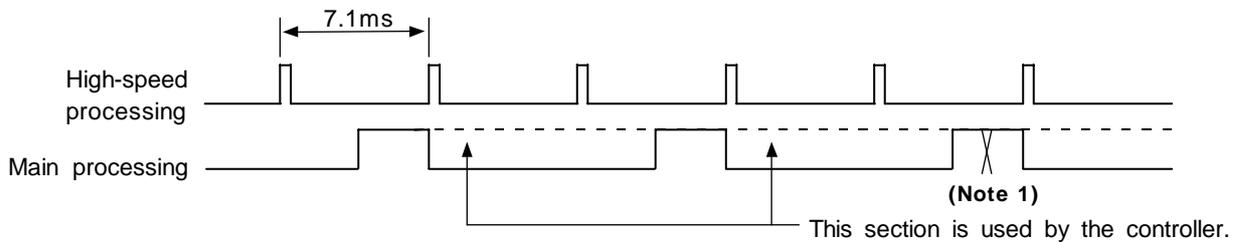
2. PLC Processing Program

2.1 PLC Processing Program Level and Operation

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. Processing time of this program shall not exceed 0.5ms. 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. |

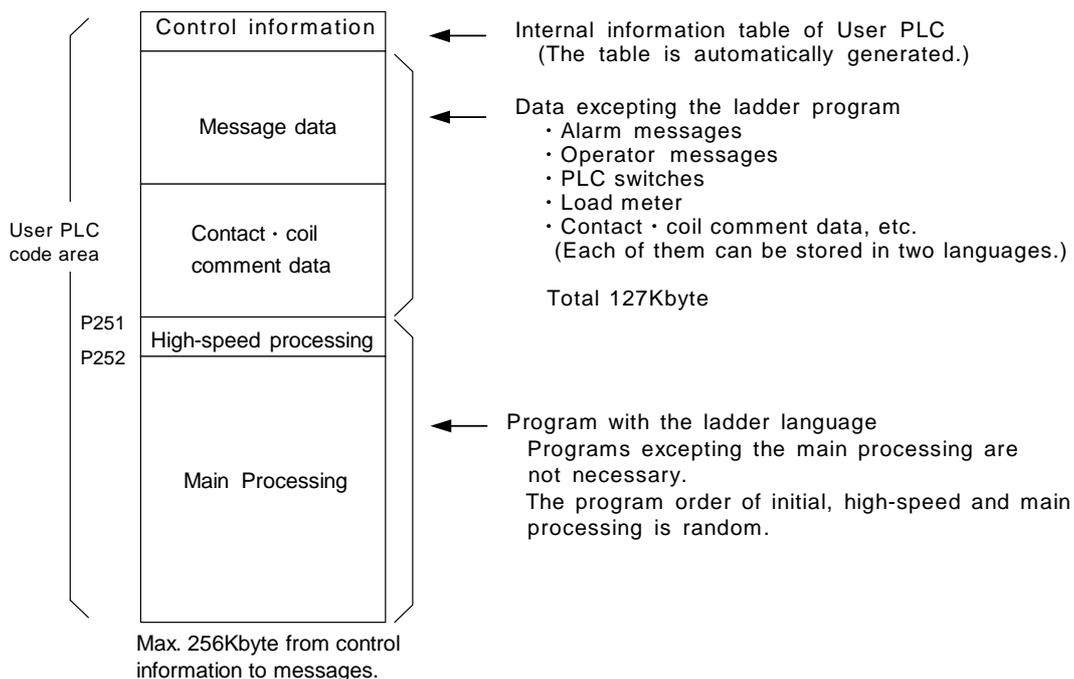


(Note 1) The section from the END command to the next scan is done immediately as shown with the X section. Note that the min. scan time will be 14.2ms.

Fig. 2.1-1 PLC processing program operation timing chart

2.2 User Memory Area Configuration

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



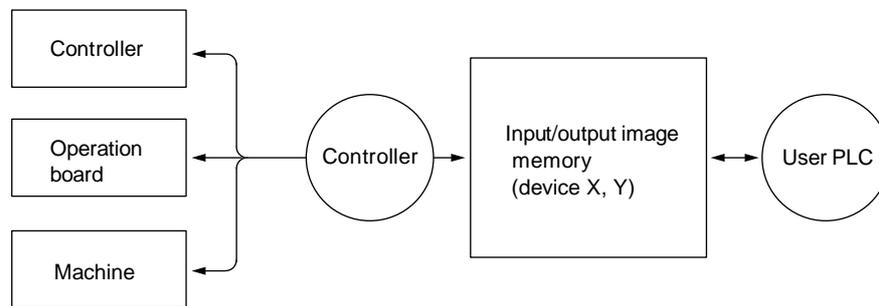
3. Input/Output Signals

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 unit is installed on the communication terminal.

Fig. 3.1-1 Concept of input/output processing

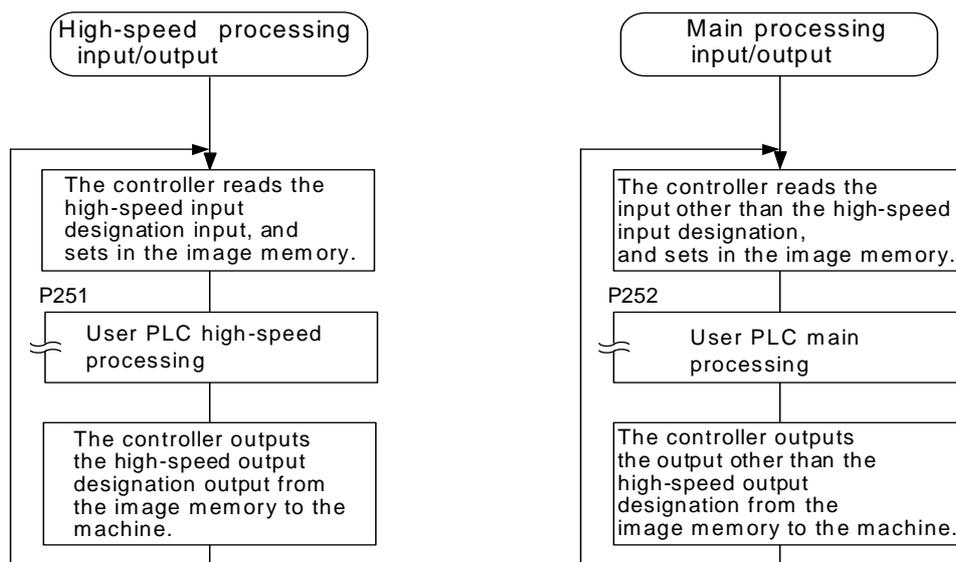


Fig. 3.1-2 Input/output processing conforming to program level

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 processing 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 |
| Input signal from MELSEC when connected to MELSEC | x | x |
| Output signal to MELSEC when connected to MELSEC | x | x |

○ : Possible x : Not possible

The operation board in Table 3.1-1 is applied when control is performed by operation board input/output card that can be added as NC option.

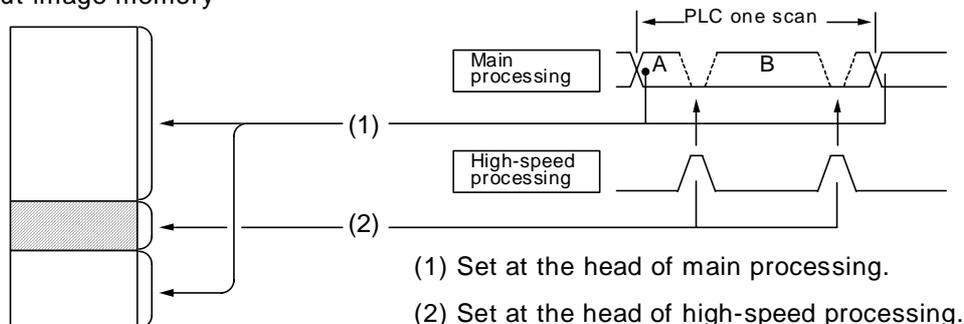
3.2 Handling of Input Signals Designated for High-Speed Input

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 interrupt input signals or 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.

Input image memory



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 |
|-------------------------|-------|-------|-------|-------|-------|-------|-------|-------|---|
| Bit selection parameter | X 7 0 | X 6 0 | X 5 0 | X 4 0 | X 3 0 | X 2 0 | X 1 0 | X 0 0 | These bits correspond to the low-order byte (bits 0~7) of file register R2928. |
| # 6 4 5 7 | S | S | S | S | S | S | S | S | |
| | X 7 F | X 6 F | X 5 F | X 4 F | X 3 F | X 2 F | X 1 F | X 0 F | |
| # 6 4 5 8 | S | S | S | S | S | S | S | S | These bits correspond to the high-order byte (bits 8~F) of file register R2928. |
| | X F 0 | X E 0 | X D 0 | X C 0 | X B 0 | X A 0 | X 9 0 | X 8 0 | |
| | X F F | X E F | X D F | X C F | X B F | X A F | X 9 F | X 8 F | |

(2) High-speed output designation

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | ← Bit |
|-------------------------|-------|-------|-------|-------|-------|-------|-------|-------|---|
| Bit selection parameter | Y 7 0 | Y 6 0 | Y 5 0 | Y 4 0 | Y 3 0 | Y 2 0 | Y 1 0 | Y 0 0 | These bits correspond to the low-order byte (bits 0~7) of file register R2930. |
| # 6 4 6 1 | S | S | S | S | S | S | S | S | |
| | Y 7 F | Y 6 F | Y 5 F | Y 4 F | Y 3 F | Y 2 F | Y 1 F | Y 0 F | |
| # 6 4 6 2 | S | S | S | S | S | S | S | S | These bits correspond to the high-order byte (bits 8~F) of file register R2930. |
| | Y F 0 | Y E 0 | Y D 0 | Y C 0 | Y B 0 | Y A 0 | Y 9 0 | Y 8 0 | |
| | Y F F | Y E F | Y D F | Y C F | Y B F | Y A F | Y 9 F | Y 8 F | |

- 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 R2928]— To designate X00~X0F, X10~X1F (bit 0 and 1 for H3)

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 | | # | Corresponding file registers | | # | Corresponding file registers | |
|------|------------------------------|-----------|------|------------------------------|-----------|------|------------------------------|-----------|
| | High order | Low order | | High order | Low order | | High order | Low order |
| 6301 | R2801 | R2800 | 6321 | R2841 | R2840 | 6341 | R2881 | R2880 |
| 6302 | R2803 | R2802 | 6322 | R2843 | R2842 | 6342 | R2883 | R2882 |
| 6303 | R2805 | R2804 | 6323 | R2845 | R2844 | 6343 | R2885 | R2884 |
| 6304 | R2807 | R2806 | 6324 | R2847 | R2846 | 6344 | R2887 | R2886 |
| 6305 | R2809 | R2808 | 6325 | R2849 | R2848 | 6345 | R2889 | R2888 |
| 6306 | R2811 | R2810 | 6326 | R2851 | R2850 | 6346 | R2891 | R2890 |
| 6307 | R2813 | R2812 | 6327 | R2853 | R2852 | 6347 | R2893 | R2892 |
| 6308 | R2815 | R1814 | 6328 | R2855 | R2854 | 6348 | R2895 | R2894 |
| 6309 | R2817 | R2816 | 6329 | R2857 | R2856 | | | |
| 6310 | R2819 | R2818 | 6330 | R2859 | R2858 | | | |
| 6311 | R2821 | R2820 | 6331 | R2861 | R2860 | | | |
| 6312 | R2823 | R2822 | 6332 | R2863 | R2862 | | | |
| 6313 | R2825 | R2824 | 6333 | R2865 | R2864 | | | |
| 6314 | R2827 | R2826 | 6334 | R2867 | R2866 | | | |
| 6315 | R2829 | R2828 | 6335 | R2869 | R2868 | | | |
| 6316 | R2831 | R2830 | 6336 | R2871 | R2870 | | | |
| 6317 | R2833 | R2832 | 6337 | R2873 | R2872 | | | |
| 6318 | R2835 | R2834 | 6338 | R2875 | R2874 | | | |
| 6319 | R2837 | R2836 | 6339 | R2877 | R2876 | | | |
| 6320 | R2839 | R2838 | 6340 | R2879 | R2878 | | | |

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 | # | Corresponding file register | # | Corresponding file register | # | Corresponding file register |
|------|-----------------------------|--|-----------------------------|------|-----------------------------|---|-----------------------------|
| 6401 | R2900-LOW | 6433 | R2916-LOW | 6449 | R2924-LOW | 6481 | R2940-LOW |
| 6402 | R2900-HIGH | 6434 | R2916-HIGH | 6450 | R2924-HIGH | 6482 | R2940-HIGH |
| 6403 | R2901-L | 6435 | R2917-L | 6451 | R2925-L | 6483 | R2941-L |
| 6404 | R2901-H | 6436 | R2917-H | 6452 | R2925-H | 6484 | R2941-H |
| 6405 | R2902-L | 6437 | R2918-L | 6453 | R2926-L | 6485 | R2942-L |
| 6406 | R2902-H | 6438 | R2918-H | 6454 | R2926-H | 6486 | R2942-H |
| 6407 | R2903-L | 6439 | R2919-L | 6455 | R2927-L | 6487 | R2943-L |
| 6408 | R2903-H | 6440 | R2919-H | 6456 | R2927-H | 6488 | R2943-H |
| 6409 | R2904-L | 6441 | R2920-L | 6457 | R2928-L | 6489 | R2944-L |
| 6410 | R2904-H | 6442 | R2920-H | 6458 | R2928-H | 6490 | R2944-H |
| 6411 | R2905-L | 6443 | R2921-L | 6459 | R2929-L | 6491 | R2945-L |
| 6412 | R2905-H | 6444 | R2921-H | 6460 | R2929-H | 6492 | R2945-H |
| 6413 | R2906-L | 6445 | R2922-L | 6461 | R2930-L | 6493 | R2946-L |
| 6414 | R2906-H | 6446 | R2922-H | 6462 | R2930-H | 6494 | R2946-H |
| 6415 | R2907-L | 6447 | R2923-L | 6463 | R2931-L | 6495 | R2947-L |
| 6416 | R2907-H | 6448 | R2923-H | 6464 | R2931-H | 6496 | R2947-H |
| 6417 | R2908-L | Use bit selection parameters #6401~#6448 freely. | | 6465 | R2932-L | Bit selection parameter #6449~#6496 are PLC operation selection parameters used by the machine manufacturer and MITSUBISHI. The contents are fixed. | |
| 6418 | R2908-H | | | 6466 | R2932-H | | |
| 6419 | R2909-L | | | 6467 | R2933-L | | |
| 6420 | R2909-H | | | 6468 | R2933-H | | |
| 6421 | R2910-L | | | 6469 | R2934-L | | |
| 6422 | R2910-H | | | 6470 | R2934-H | | |
| 6423 | R2911-L | | | 6471 | R2935-L | | |
| 6424 | R2911-H | | | 6472 | R2935-H | | |
| 6425 | R2912-L | | | 6473 | R2936-L | | |
| 6426 | R2912-H | | | 6474 | R2936-H | | |
| 6427 | R2913-L | | | 6475 | R2937-L | | |
| 6428 | R2913-H | | | 6476 | R2937-H | | |
| 6429 | R2914-L | | | 6477 | R2938-L | | |
| 6430 | R2914-H | | | 6478 | R2938-H | | |
| 6431 | R2915-L | | | 6479 | R2939-L | | |
| 6432 | R2915-H | | | 6480 | R2939-H | | |

Bit selection screen

[BIT SELECT] SETUP PARAM 6. 5/ 6

| | | | | |
|---------------|---------------|---------------|---------------|---------------|
| # 76543210 | # 76543210 | # 76543210 | # 76543210 | # 76543210 |
| 6401 00000000 | 6413 00000000 | 6425 00000000 | 6437 00000000 | 6449 00000000 |
| 6402 00000000 | 6414 00000000 | 6426 00000000 | 6438 00000000 | 6450 00000000 |
| 6403 00000000 | 6415 00000000 | 6427 00000000 | 6439 00000000 | 6451 00000000 |
| 6404 00000000 | 6416 00000000 | 6428 00000000 | 6440 00000000 | 6452 00000000 |
| 6405 00000000 | 6417 00000000 | 6429 00000000 | 6441 00000000 | 6453 00000000 |
| 6406 00000000 | 6418 00000000 | 6430 00000000 | 6442 00000000 | 6454 00000000 |
| 6407 00000000 | 6419 00000000 | 6431 00000000 | 6443 00000000 | 6455 00000000 |
| 6408 00000000 | 6420 00000000 | 6432 00000000 | 6444 00000000 | 6456 00000000 |
| 6409 00000000 | 6421 00000000 | 6433 00000000 | 6445 00000000 | 6457 00000000 |
| 6410 00000000 | 6422 00000000 | 6434 00000000 | 6446 00000000 | 6458 00000000 |
| 6411 00000000 | 6423 00000000 | 6435 00000000 | 6447 00000000 | 6459 00000000 |
| 6412 00000000 | 6424 00000000 | 6436 00000000 | 6448 00000000 | 6460 00000000 |

76543210

#() DATA()

MC-ERR **PLC** MACRO PSW MENU

Contents of bit selection parameters #6449~#6496

| | Symbol name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | | 0 |
|---|------------------|---|--|------------------------------|---------------------------------------|---------------------------|-------------------------------------|------------------------|----------|----------------------|
| 0 | #6449 R2924 L | Control unit thermal alarm on | Setting and display unit thermal mgmt on | - | | Counter C retention | Integrating timer T retention | PLC counter program on | | PLC timer program on |
| 1 | #6450 R2924 H | | External alarm message display | Alarm/operator change | Full screen display of message | - | Operator message on | R method | F method | Alarm message on |
| 2 | #6451 R2925 L | - | - | GX-Developer communication | PLC development environment selection | | Onboard editing not possible | APLC custom release | | Onboard on |
| 3 | #6452 R2925 H | - | | GOT communication connection | | Counter (fixed) retention | Integrating timer (fixed) retention | | | - |
| 4 | #6453 R2926 L | - | - | - | - | - | Message language change code | | | |
| 5 | #6454 R2926 H | | | | | | | | | |
| 6 | #6455 R2927 L | - | - | - | - | - | - | - | - | - |
| 7 | #6456 R2927 H | - | - | - | - | - | - | - | - | - |
| 8 | #6457 R2928 L | High-speed input specification 1 | | | | | | | | |
| 9 | #6458 R2928 H | High-speed input specification 2 | | | | | | | | |
| A | #6459 R2929 L | High-speed input specification 3 (Spare) | | | | | | | | |
| B | #6460 R2929 H | High-speed input specification 4 (Spare) | | | | | | | | |
| C | #6461 R2930 L | High-speed output specification 1 | | | | | | | | |
| D | #6462 R2930 H | High-speed output specification 2 | | | | | | | | |
| E | #6463 R2931 L | High-speed output specification 3 (Spare) | | | | | | | | |
| F | #6464 R2931 H | High-speed output specification 4 (Spare) | | | | | | | | |

4. Parameters

| | Symbol name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|------------------|---|---|------------------------|---|---|---|---|---|-----------------------|
| 0 | #6465 R2932 L | - | - | - | - | - | - | - | - | |
| 1 | #6466 R2932 H | - | - | - | - | - | - | - | - | |
| 2 | #6467 R2933 L | - | - | - | - | - | - | - | - | |
| 3 | #6468 R2933 H | | | | | | | | | |
| 4 | #6469 R2934 L | | | Standard PLC parameter | | | | | - | NC alarm 4 output off |
| 5 | #6470 R2934 H | | | | | | | | | |
| 6 | #6471 R2935 L | - | - | - | - | - | - | - | - | |
| 7 | #6472 R2935 H | - | - | - | - | - | - | - | - | |
| 8 | #6473 R2936 L | - | | | | | | | - | |
| 9 | #6474 R2936 H | | | | | | | | | |
| A | #6475 R2937 L | | | | | | | | | |
| B | #6476 R2937 H | | | | | | | | | |
| C | #6477 R2938 L | | | | | | | | | |
| D | #6478 R2938 H | | | | | | | | | |
| E | #6479 R2939 L | | | | | | | | | |
| F | #6480 R2939 H | | | | | | | | | |

(Note 1) Be sure to set the bits indicated - and blanks to 0.

(Note 2) Parameters #6481 to #6496 are reserved for debugging by Mitsubishi.

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 and H are represented in hexadecimal notation. The device numbers of other devices are represented in decimal notation.

5.2 Device List

| Device | Device No. | Unit | Details |
|--------|------------------------------|--------------------|--|
| X* | X0~XABF (2752 points) | 1 bit | Input signal to PLC. Machine input, etc. |
| Y* | Y0~YDEF (3584 points) | 1 bit | Output signal from PLC. Machine output, etc. |
| M | M0~M8191 (8192 points) | 1 bit | Temporary memory |
| F | F0~F127 (128 points) | 1 bit | Temporary memory, alarm message interface |
| L | L0~L255 (256 points) | 1 bit | Latch relay (backup memory) |
| SM* | SM0~SM127 (128 points) | 1 bit | Special relay |
| T | T0~T15 (16 points) | 1 bit or 16 bits | 10ms unit timer |
| | T16~T95 (80 points) | 1 bit or 16 bits | 100ms unit timer |
| | T96~T103 (8 points) | 1 bit or 16 bits | 100ms unit integrating timer |
| C | C0~C23 (24 points) | 1 bit or 16 bits | Counter |
| D | D0~D1023 (1024 points) | 16 bits or 32 bits | Data register for arithmetic operation |
| R* | R0~R8191 (8192 points) | 16 bits or 32 bits | File register. R500 to R549 and R1900 to R2799 are released to the user for interface between the PLC and controller. R1900 to R2799 are backed up by the battery. |
| Z | Z0~Z1 (2 points) | 16 bits | Index of D or R address ($\pm n$) |
| N | N0~N7 (8 points) | — | Master control nesting level |
| P* | P0~P255 (256 points) | — | Label for conditional jump and subroutine call |
| K | K-32768~K32767 | — | Decimal constant for 16-bit command |
| | K-2147483648~ K2147483647 | — | Decimal constant for 32-bit command |
| H | H0~HFFFF | — | Hexadecimal constant for 16-bit command |
| | H0~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) When using temporary memory such as M device, separate READ and WRITE every 8bits.

5.3 Detailed Explanation of Devices

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 CNC.

Input X

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

The diagram illustrates the connection between an external input circuit and a PLC program. On the left, the 'Input circuit' shows three devices: a push-button (PB1), a limit switch (LS2), and another push-button (PB16). PB1 and PB16 are connected to PLC input X10, while LS2 is connected to X11. A 'Hypothetical relay' is shown between the input devices and the PLC. On the right, the 'Program' section shows the PLC's internal logic. X10 is used as a normally closed contact (indicated by two parallel vertical lines), and X1F is used as a normally open contact (indicated by a single vertical line). The PLC is indicated by a dashed box and an arrow.

- The input No. is expressed with a hexadecimal.

Output Y

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

The diagram illustrates the connection between a PLC program and an external output circuit. On the left, the 'Program' section shows the PLC's internal logic for output Y10. It features a normally open contact (Y10) in parallel with a normally closed contact (Y10), and another set of normally open (Y10) and normally closed (Y10) contacts in series. On the right, the 'Output circuit' shows a 24V DC power source connected to a transistor. The transistor's base is connected to the PLC output Y10. A diode is connected in parallel with the transistor's collector-emitter junction. The collector is connected to a 'Load', and the emitter is connected to ground.

- 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 relays M |
|--|
| <ul style="list-style-type: none"> (a) These relays are 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 |
|--|
| <p>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.</p> |

| Latch relay L |
|--|
| <ul style="list-style-type: none"> (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 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 |
|---|
| <ul style="list-style-type: none"> (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. |

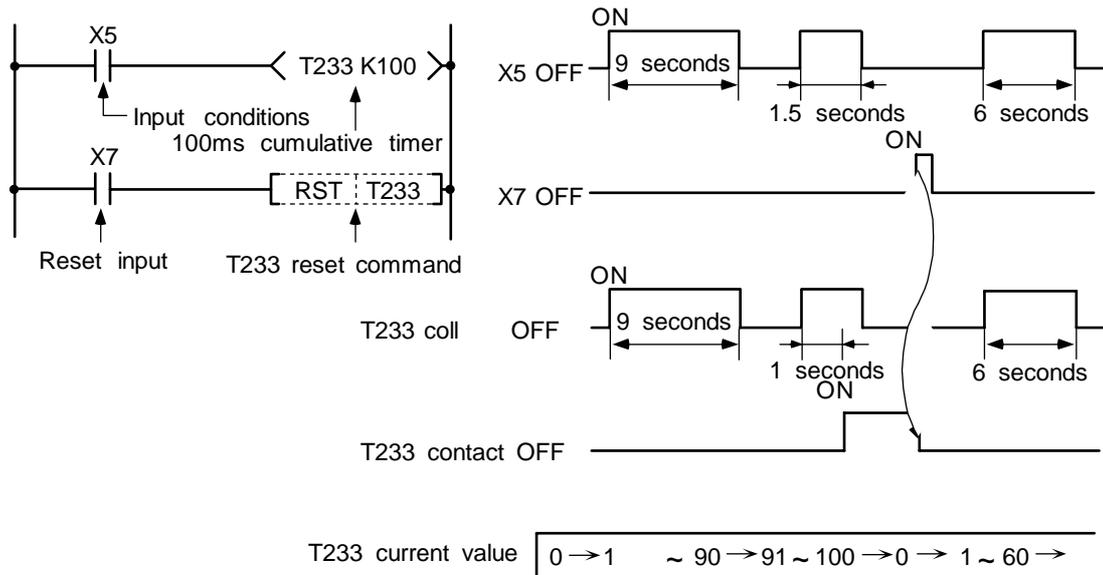
5.3.4 Timer T

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

| |
|---|
| <p>100ms Timer T</p> |
| <p>(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.</p> |
| |
| <p>(c) When #6449 bit0=1, the value is set with a decimal (Kn), 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.</p> |
| <p>10ms Timer T</p> |
| <p>(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.</p> |
| |
| <p>(c) When #6449 bit0=1, the value is set with a decimal (Kn), 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.</p> |

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) When #6449 bit0=1, the value is set with a decimal (Kn), 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 (#6449 bit2=1) is set, the 100ms integrated timer current value (count value) will be held even when the power is turned OFF.

- (2) With the device T, the contact • coil is handled as bit device, and the current value is handled as word device. In the function commands described after, the word device T indicates the current value even if there is no description about it.
- (3) When #6449 bit0=0 is set, timer value can be specified with the parameter set in the setting and display unit. At this time, the relationship between timer device and parameter is as shown below.

| Device | Parameter |
|--------------|----------------|
| T0 to T15 | #6000 to #6015 |
| T56 to T135 | #6016 to #6095 |
| T232 to T239 | #6096 to #6103 |

- (Note 1)** T16 to T55, T136 to T231, and T240 to T255 are specified with a program (Kn) regardless of #6449 bit0.
- (Note2)** Even when #6449 bit0=0, Kn is required for a sequence program. Note that, however, the Kn value is invalid.
- (Note 3)** When the data register (D) is used as setting value, the data register (D) details will be the setting value regardless of #6449 bit0.

5.3.5 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 |
|--|
| <p>(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.</p> <p>(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.</p> <p>(c) When the bit selection parameter is set, the counter current value (count value) will be held even when the power is turned OFF. Note that some can not be held depending on the version of CNC.</p> |

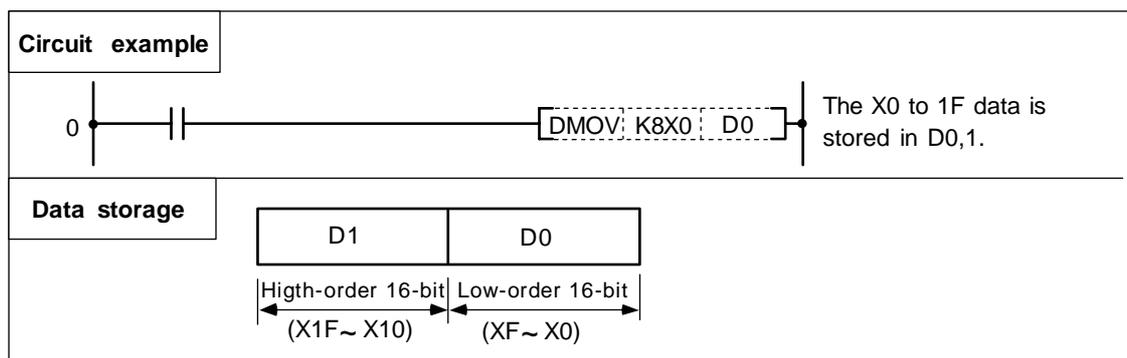
- (2) With the device C, the contact • coil is handled as bit device, and the current value (counter value) is handled as word device. In the function commands described after, the word device C indicates the current value (counter value) even if there is no description about it.
- (3) The counter setting value can be set with the setting and display unit using device C. (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.

(Note) The setting value for device C24 to C127 of counter C cannot be set from the setting and display unit.

5.3.6 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.



- (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:

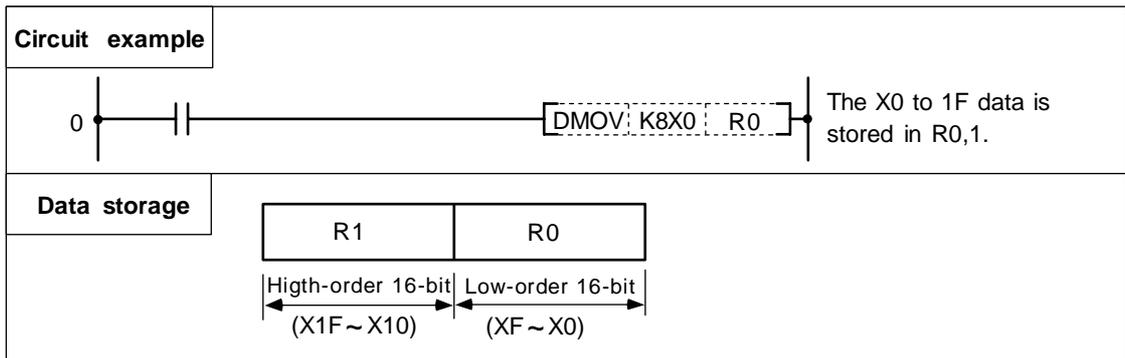
| | |
|-----------------------------------|----------------------|
| Decimal -32768 to 32767 | } For 16-bit command |
| Hexadecimal 0 to FFFF | |
| Decimal -2147483648 to 2147483647 | } For 32-bit command |
| Hexadecimal 0 to FFFFFFFF | |

- (6) Data registers D0 to D1023 are all user release data registers.

5.3.7 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.

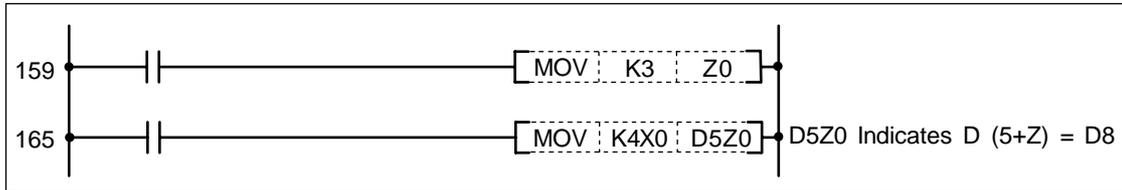


- (3) The data that is stored once in the sequence program is held until other data is stored.
- (4) With the file registers, the following registers are the user release.
 R500 to R549, R1900 to R2799
 The following registers of the registers above are not cleared when the power is turned OFF.
 R1900 to R2799
 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 | } | For 16-bit command (Using Dn) |
| Hexadecimal 0 to FFFF | | |
| Decimal -2147483648 to 2147483647 | } | For 32-bit command (Using Dn+1, Dn) |
| Hexadecimal 0 to FFFFFFFF | | |

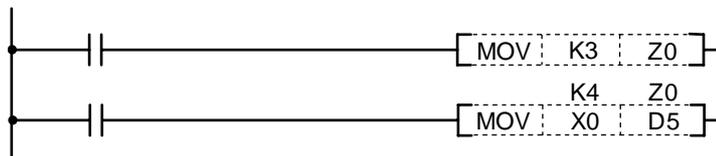
5.3.8 Index Registers Z

- (1) The index registers are used as ornaments for the device (T, C, D, R).



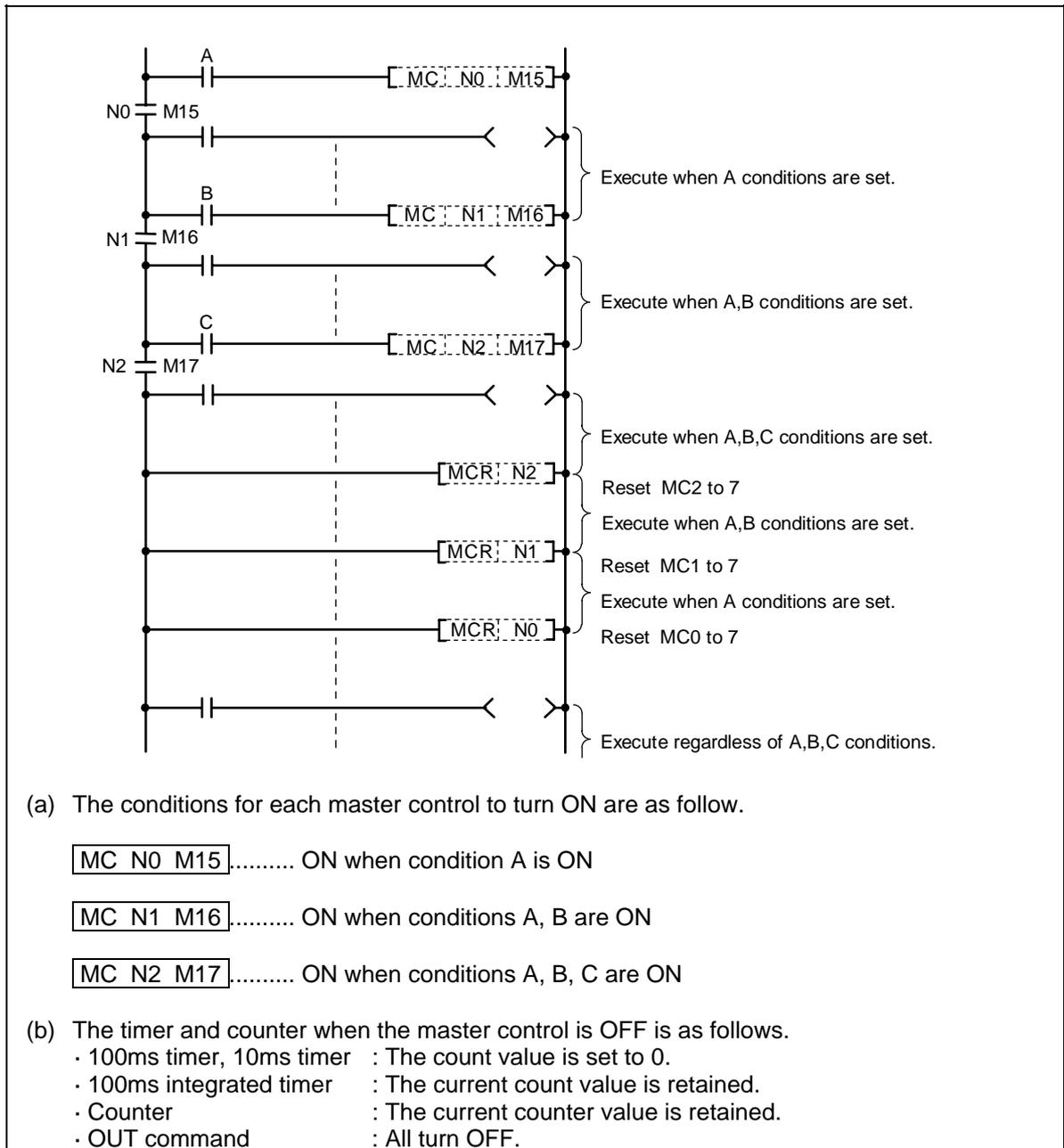
- (2) The index register has a 1-point 16-bit configuration, and can be read and written in 16-bit units.
 (3) The data stored in the index register is cleared when the power is turned OFF.
 (4) Values that can be stored: Decimal -32768 to 32767
 Hexadecimal 0 to FFFF

(Note) The CRT display of the index registers Z is as shown below.



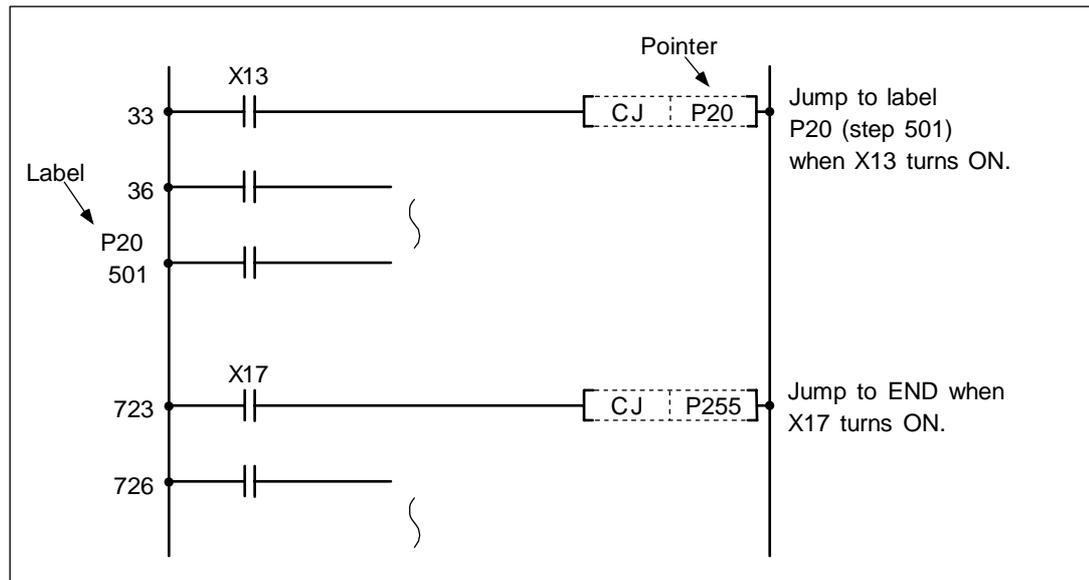
5.3.9 Nesting N

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



5.3.10 Pointer P

- (1) The pointer indicates the branch command (CJ, CALL) jump destination. The pointer No. assigned at the jump destination head is called the label.
- (2) Pointers P0 to P159, P251, P252, P255, P300 to P511 (for C language module call) 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: Label for starting PLC high-speed processing program.
P252: Label for starting PLC main (ladder) processing program.

CAUTION

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

- (Note 1)** Do not omit P252 label even when there is only a PLC main processing program.
- (Note 2)** P251 and P252 cannot be used as CJ 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 or CALL command device must also be programmed as a label.

5.3.11 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 binary value (binary) in the PLC.

5.3.12 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

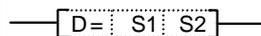
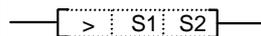
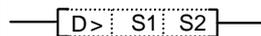
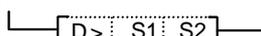
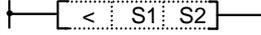
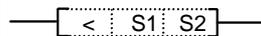
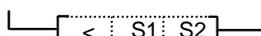
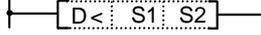
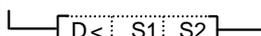
6.1.1 Basic Commands

| 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 | 38 |
| | | LDI | | Start of logic denial operation (B contact operation start) | 1 | 38 |
| | | AND | | Logical AND (A contact serial connection) | 1 | 40 |
| | | ANI | | Logical AND denial (B contact serial connection) | 1 | 40 |
| | | OR | | Logical OR (A contact parallel connection) | 1 | 42 |
| | | ORI | | Logical OR denial (B contact parallel connection) | 1 | 42 |
| | | ANB | | AND between logical blocks (Serial connection between blocks) | 1 | 44 |
| | | ORB | | OR between logical blocks (Parallel connection between blocks) | 1 | 46 |
| | | OUT | | Device output | 1~3 | 48 |
| | | SET | | Device set | 1 | 54 |
| | | RST | | Device reset | 1~2 | 56 |
| | | MC | | Master control start | 2 | 58 |
| | | MCR | | Master control release | 1 | 58 |
| | | PLS | | Generate one cycle worth of pulses at rising edge of input signal | 2 | 60 |
| | | PLF | | Generate one cycle worth of pulses at falling edge of input signal | 2 | 60 |
| | | SFT | | Device 1-bit shift | 4 | 62 |
| | | MPS | | Registration of logical operation result | 1 | 64 |
| | | MRD | | Read of operation results registered in MPS | 1 | 64 |
| | | MPP | | Reading and resetting of operation results registered in MPS | 1 | 64 |
| | | DEFR (ANDP) | | Generate one cycle worth of pulses to operation results at rising edge of input signal (Note) | 1 | 66 |

(Note) With the MELSEC PLC development tool (GX Developer), the "ANDP" command is alternatively used.

6.1.2 Function Commands

(1) Comparison commands

| Class | Process unit | Command sign | Symbol | Process details | No. of steps | Page |
|-------|--------------|--------------|---|--|--------------|------|
| = | 16-bit | LD= |  | Continuity state when (S1) = (S2) Non-continuity state when (S1) ≠ (S2) | 3 | 70 |
| | | AND= |  | | 3 | 70 |
| | | OR= |  | | 3 | 70 |
| | 32-bit | LDD= |  | Continuity state when (S1+1, S1)=(S2+1, S2) Non-continuity state when (S1+1, S1) ≠ (S2+1, S2) | 3~4 | 72 |
| | | ANDD= |  | | 3~4 | 72 |
| | | ORD= |  | | 3~4 | 72 |
| > | 16-bit | LD> |  | Continuity state when (S1) > (S2) Non-continuity state when (S1) ≤ (S2) | 3 | 74 |
| | | AND> |  | | 3 | 74 |
| | | OR> |  | | 3 | 74 |
| | 32-bit | LDD> |  | Continuity state when (S1+1, S1) > (S2+1, S2) Non-continuity state when (S1+1, S1) ≤ (S2+1, S2) | 3~4 | 76 |
| | | ANDD> |  | | 3~4 | 76 |
| | | ORD> |  | | 3~4 | 76 |
| < | 16-bit | LD< |  | Continuity state when (S1) < (S2) Non-continuity state when (S1) ≥ (S2) | 3 | 78 |
| | | AND< |  | | 3 | 78 |
| | | OR< |  | | 3 | 78 |
| | 32-bit | LDD< |  | Continuity state when (S1+1, S1) < (S2+1, S2) Non-continuity state when (S1+1, S1) ≥ (S2+1, S2) | 3~4 | 80 |
| | | ANDD< |  | | 3~4 | 80 |
| | | ORD< |  | | 3~4 | 80 |

(2) Arithmetic operation commands

| Class | Process unit | Command sign | Symbol | Process details | No. of steps | Page |
|-------|--------------|--------------|---|--|--------------|------|
| + | 16-bit | + | $\text{---} \left[\begin{array}{ c c c c } \hline + & S1 & S2 & D \\ \hline \end{array} \right] \text{---}$ | $(S1) + (S2) \rightarrow (D)$ | 4 | 82 |
| | 32-bit | D+ | $\text{---} \left[\begin{array}{ c c c c } \hline D+ & S1 & S2 & D \\ \hline \end{array} \right] \text{---}$ | $(S1+1, S1) + (S2+1, S2) \rightarrow (D+1, D)$ | 4~5 | 84 |
| - | 16-bit | - | $\text{---} \left[\begin{array}{ c c c c } \hline - & S1 & S2 & D \\ \hline \end{array} \right] \text{---}$ | $(S1) - (S2) \rightarrow (D)$ | 4 | 86 |
| | 32-bit | D- | $\text{---} \left[\begin{array}{ c c c c } \hline D- & S1 & S2 & D \\ \hline \end{array} \right] \text{---}$ | $(S1+1, S1) - (S2+1, S2) \rightarrow (D+1, D)$ | 4~5 | 88 |
| * | 16-bit | * | $\text{---} \left[\begin{array}{ c c c c } \hline * & S1 & S2 & D \\ \hline \end{array} \right] \text{---}$ | $(S1) \times (S2) \rightarrow (D+1, D)$ | 4 | 90 |
| | 32-bit | D* | $\text{---} \left[\begin{array}{ c c c c } \hline D* & S1 & S2 & D \\ \hline \end{array} \right] \text{---}$ | $(S1+1, S1) \times (S2+1, S2) \rightarrow (D+3, D+2, D+1, D)$ | 5~6 | 92 |
| / | 16-bit | / | $\text{---} \left[\begin{array}{ c c c c } \hline / & S1 & S2 & D \\ \hline \end{array} \right] \text{---}$ | $(S1) \dot{=} (S2) \rightarrow (D)$ Quotient (D) Remainder (D+1) | 5 | 94 |
| | 32-bit | D/ | $\text{---} \left[\begin{array}{ c c c c } \hline D/ & S1 & S2 & D \\ \hline \end{array} \right] \text{---}$ | $(S1+1, S1) \dot{=} (S2+1, S2) \rightarrow$ Quotient (D+1,D) Remainder (D+3, D+2) | 5~6 | 96 |
| +1 | 16-bit | INC | $\text{---} \left[\begin{array}{ c c } \hline \text{INC} & D \\ \hline \end{array} \right] \text{---}$ | $(D) + 1 \rightarrow (D)$ | 2 | 98 |
| | 32-bit | DINC | $\text{---} \left[\begin{array}{ c c } \hline \text{DINC} & D \\ \hline \end{array} \right] \text{---}$ | $(D+1, D) + 1 \rightarrow (D + 1, D)$ | 2 | 100 |
| -1 | 16-bit | DEC | $\text{---} \left[\begin{array}{ c c } \hline \text{DEC} & D \\ \hline \end{array} \right] \text{---}$ | $(D) - 1 \rightarrow (D)$ | 2 | 102 |
| | 32-bit | DDEC | $\text{---} \left[\begin{array}{ c c } \hline \text{DDEC} & D \\ \hline \end{array} \right] \text{---}$ | $(D + 1, D) - 1 \rightarrow (D + 1, D)$ | 2 | 104 |

(3) BCD ↔ BIN conversion commands

| Class | Process unit | Command sign | Symbol | Process details | No. of step | Page |
|-------|--------------|--------------|--|--|-------------|------|
| BCD | 16-bit | BCD | $\text{---} \left[\begin{array}{ c c c } \hline \text{BCD} & S & D \\ \hline \end{array} \right] \text{---}$ | BCD conversion $(S) \rightarrow (D)$ └ BIN (0 to 9999) | 3 | 106 |
| | 32-bit | DBCD | $\text{---} \left[\begin{array}{ c c c } \hline \text{DBCD} & S & D \\ \hline \end{array} \right] \text{---}$ | BCD conversion $(S1+1, S1) \rightarrow (D+1, D)$ └ BIN (0 to 99999999) | 4 | 108 |
| BIN | 16-bit | BIN | $\text{---} \left[\begin{array}{ c c c } \hline \text{BIN} & S & D \\ \hline \end{array} \right] \text{---}$ | BIN conversion $(S) \rightarrow (D)$ └ BIN (0 to 9999) | 3 | 110 |
| | 32-bit | DBIN | $\text{---} \left[\begin{array}{ c c c } \hline \text{DBIN} & S & D \\ \hline \end{array} \right] \text{---}$ | BIN conversion $(S1+1, S1) \rightarrow (D+1, D)$ └ BIN (0 to 99999999) | 4 | 112 |

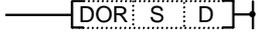
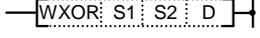
(4) Data transmission commands

| Class | Process unit | Command sign | Symbol | Process details | No. of step | Page |
|---------------------------------|--------------|--------------|--------|-----------------------|-------------|------|
| Transmission | 16-bit | MOV | | (S) → (D) | 3 | 114 |
| | 32-bit | DMOV | | (S+1,S) → (D+1,D) | 3-4 | 116 |
| Conversion | 16-bit | XCH | | (D1) ↔ (D2) | 4 | 118 |
| | 32-bit | DXCH | | (D1+1,D1) ↔ (D2+1,D2) | 4 | 120 |
| Batch transmission | 16-bit | BMOV | | | 5 | 122 |
| Batch transmission of same data | 16-bit | FMOV | | | 5 | 124 |

(5) Program branch commands

| Class | Process unit | Command sign | Symbol | Process details | No. of step | Page |
|-----------------|--------------|--------------|--------|--|-------------|------|
| Jump | — | CJ | | Jump to P** after input conditions are met | 2 | 126 |
| Program end | — | FEND | | End process during sequence program | 1 | 128 |
| Subroutine call | — | CALL | | Execute P** sub-routine program after input conditions are met | 2 | 130 |
| Return | — | RET | | Return to main program from subroutine program | 1 | 130 |

(6) Logical operation commands

| Class | Process unit | Command sign | Symbol | Process details | No. of step | Page |
|-----------------|--------------|--------------|---|---|-------------|------|
| Logical AND | 16-bit | WAND |  | $(S1) \wedge (S2) \rightarrow (D)$ | 4 | 132 |
| | 32-bit | DAND |  | $(D + 1, D) \wedge (S + 1, S) \rightarrow (D + 1, D)$ | 3~4 | 134 |
| Logical OR | 16-bit | WOR |  | $(S1) \vee (S2) \rightarrow (D)$ | 4 | 136 |
| | 32-bit | DOR |  | $(D + 1, D) \vee (S + 1, S) \rightarrow (D + 1, D)$ | 3~4 | 138 |
| Exclusive OR | 16-bit | WXOR |  | $(S1) \vee - (S2) \rightarrow (D)$ | 4 | 140 |
| | 32-bit | DXOR |  | $(D + 1, D) (S + 1, S) \rightarrow (D + 1, D)$ | 3~4 | 142 |
| Complement of 2 | 16-bit | NEG |  | $\overline{(D)} + 1 \rightarrow (D)$ | 2 | 144 |

(7) Rotation commands

| Class | Process unit | Command sign | Symbol | Process details | No. of step | Page |
|----------------|--------------|--------------|--------|--|-------------|------|
| Right rotation | 16-bit | ROR | | <p>b15 (D) b0 SM12 Rotate n bits right.</p> | 3 | 146 |
| | | RCR | | <p>b15 (D) b0 SM12 Rotate n bits right.</p> | 3 | 148 |
| | 32-bit | DROR | | <p>^(D+1) b31 ~ b16 b15 ~ b0 SM12 Rotate n bits right.</p> | 3 | 150 |
| | | DRCR | | <p>^(D+1) b31 ~ b16 b15 ~ b0 SM12 Rotate n bits right.</p> | 3 | 152 |
| Left rotation | 16-bit | ROL | | <p>SM12 b15 (D) b0 Rotate n bits left.</p> | 3 | 154 |
| | | RCL | | <p>SM12 b15 (D) b0 Rotate n bits left.</p> | 3 | 156 |
| | 32-bit | DROL | | <p>^(D+1) SM12 b31 ~ b16 b15 ~ b0 Rotate n bits left.</p> | 3 | 158 |
| | | DRCL | | <p>^(D+1) SM12 b31 ~ b16 b15 ~ b0 Rotate n bits left.</p> | 3 | 160 |
| Right shift | 16-bit | SFR | | <p>b15 bn b0 b15 b0 SM12 0~0</p> | 3 | 162 |
| | Device unit | DSFR | | <p>n (D) 0</p> | 4 | 164 |
| Left shift | 16-bit | SFL | | <p>b15 bn b0 SM12 b15 b0 0~0</p> | 3 | 166 |
| | Device unit | DSFL | | <p>n (D) 0</p> | 4 | 168 |

(8) Data processing commands

| Class | Process unit | Command sign | Symbol | Process details | No. of step | Page |
|-------------------------|---------------------|--------------|--------|---|-------------|------|
| Search | 16-bit | SER | | | 6 | 170 |
| Number of bits set to 1 | 16-bit | SUM | | | 4 | 172 |
| Decode | 2 ⁿ -bit | DECO | | | 5 | 174 |
| | 16-bit | SEG | | | 3 | 176 |
| Average value | 16-bit | S.AVE | | 16-bit data average value $\frac{1}{n} \sum_{i=1}^n (S+i) \rightarrow (D)$ | 5 | 178 |

(9) Other function commands

| Class | Process unit | Command sign | Symbol | Process details | No. of step | Page |
|------------------|--------------|--------------|--------|---|-------------|------|
| Carry flag set | — | S.STC | | Carry flag contact (SM12) is turned on. | 1 | 180 |
| Carry flag reset | — | S.CLC | | Carry flag contact (SM12) is turned off. | 1 | 180 |
| BIT | 1-bit | LDBIT (<=) | | Bit test (A contact operation start handling) (Note) | 2 | 182 |
| | | ANDBIT (<=) | | Bit test (A contact series connection handling) (Note) | 2 | 182 |
| | | ORBIT (<=) | | Bit test (A contact parallel connection handling) (Note) | 2 | 182 |
| | | LDBII (<>) | | Bit test (B contact operation start handling) (Note) | 2 | 184 |
| | | ANDBII (<>) | | Bit test (B contact series connection handling) (Note) | 2 | 184 |
| | | ORBII (<>) | | Bit test (B contact parallel connection handling) (Note) | 2 | 184 |

(Note) With the MELSEC PLC development tool (GX Developer), the comparison operation commands are alternatively used.

6.1.3 Exclusive commands

| 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 | 5 | 194 |
| | | | | K2: Tool number AND search | | 195 |
| | | | | K3: Tool change | | 196 |
| | | | | K4: Random position tool change | | 197 |
| | | | | K5: Forward rotation of pointer | | 198 |
| | | | | K6: Reverse rotation of pointer | | 198 |
| | | | | K7: Normal rotation of tool table | | 199 |
| | | | | K8: Reverse rotation of tool table | | 199 |
| | | | | K9: Tool data read | | 200 |
| | | | | K10: Tool data write | | 201 |
| | | | | K11: Automatic write of tool data | | 202 |
| ROT | — | S.ROT | — S.ROT: Kn: Rn: Rm: Mn — | K1: Rotary body index | 5 | 207 |
| | | | | K3: Ring counter | | 211 |
| TSRH | — | S.TSRH | — S.TSRH: Rn: Rm: Mn — | Spare tool selection in tool life management | 4 | 212 |
| DDB | — | S.DDBA (Asynchronous) | — S.DDBA: Rn/Dn — | Data designated after Rn/Dn is read/written. | 2 | 223 |
| | | S.DDBS (Synchronous) | — S.DDBS: Rn — | Data designated after Rn is read/written. | 2 | 226 |
| CC-Link | — | G.RIRD | — G.RIRD: Un: S1: D2 — | Device data is read from the designated station. | 5 | 254 |
| CC-Link | — | R.RIWT | — G.RIWT: Un: S1: S2: D — | Device data is written into the designated station. | 5 | 256 |

6.2 Command Formats

6.2.1 How to Read the Command Table

The basic command and function command explanations are shown below.

Example of D+ command

D +BIN 32-bit addition

The command signal is indicated.

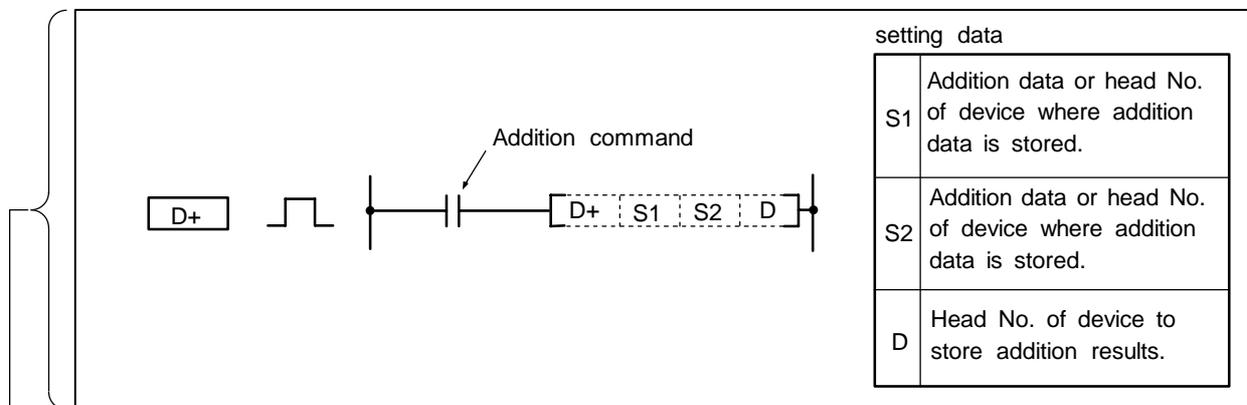
| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | | | | | | |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---|---------|-------|-------------------|--------------|-------|---|--|--|--|--|--|--|--|--|
| | Bit device | | | | | | Word device | | | | | | Constant | | Pointer | Level | | | | | | | | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | | | | | | | | |
| S1 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | | | | | |
| S2 | | | | | | | ○ | ○ | ○ | ○ | | | | | ○ | ○ | | | | | | | | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | | | | | |

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

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

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. 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 commands that can use an index (Z) are circled.
 Such a command is only the MOV command in this manual.



The D+ command circuit display format is indicated.

The functions first, then execution conditions, then program examples are described on the following pages.

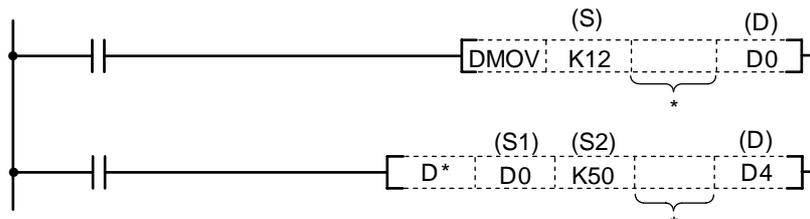
6.2.2 No. of Steps

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 some of the command codes and the 32-bit command constant K or H use two steps.

(Note) 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.)

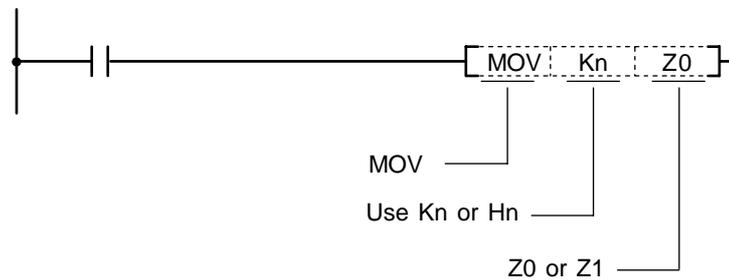


6.2.3 END Command

With the END command, both the circuit mode and the list mode are automatically created, so programming is not necessary.

6.2.4 Index Ornament

- (1) The index ornament is used to add an index (Z0, Z1) to a device, add the details of the directly designated device No. and index register, and designate the device No.
- (2) The index (Z0, Z1) can be set between -32768 to 32767 with a sign added.
- (3) The index ornament is used only for the MOV command. (It cannot be used for DMOV.)
- (4) The usable command format is shown below.
 - (a) Transmission of data to Z0, Z1

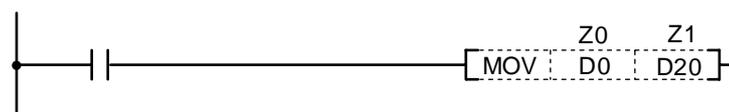


(b) Possible device combinations of MOV command with index ornament

| | S (source) | D (destination) | Program example |
|-----|------------------------------------|--|-----------------|
| MOV | Constant Kn or Hn | (Word device) · Z Example) D0Z0, R500Z1 | MOV K100 D0Z0 |
| | Word device Example) D0, R1900 | (Word device) · Z Example) D0Z0, R500Z1 | MOV D0 D100Z1 |
| | (Word device) · Z Example) D0Z0 | (Word device) · Z Example) D1Z0, D0Z1 | MOV D0Z0 D20Z0 |
| | (Word device) · Z Example) D0Z0 | Bit designation Example) K2Y20 | MOV D0Z0 K2M10 |
| | Bit designation Example) K2M00 | (Word device) · Z Example) D0Z0, R1900Z1 | MOV K2M10 D0Z0 |

(Note 1) The word device refers to T, C, D and R.

(Note 2) The display of the circuit with index ornament is as shown below.



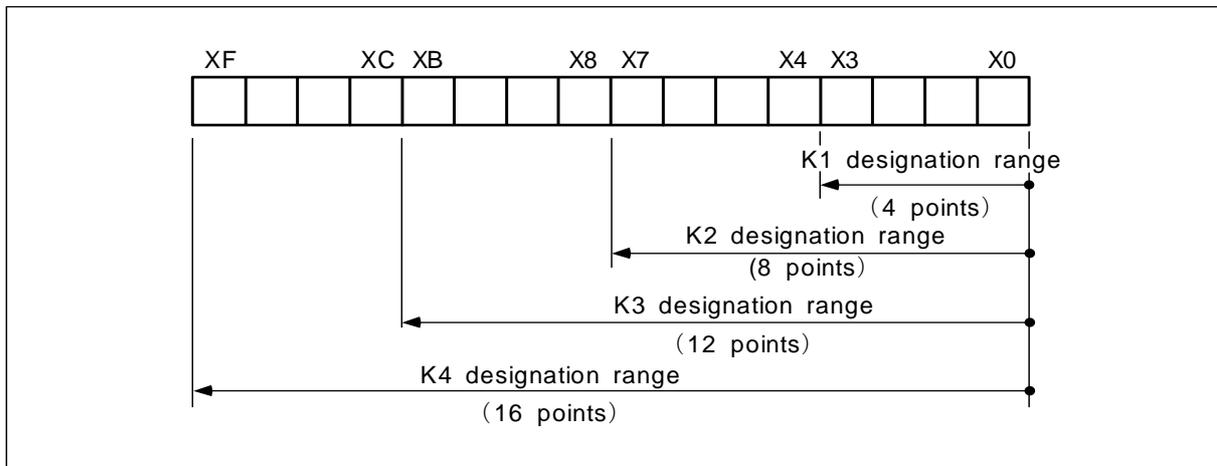
6.2.5 Digit Designation

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.

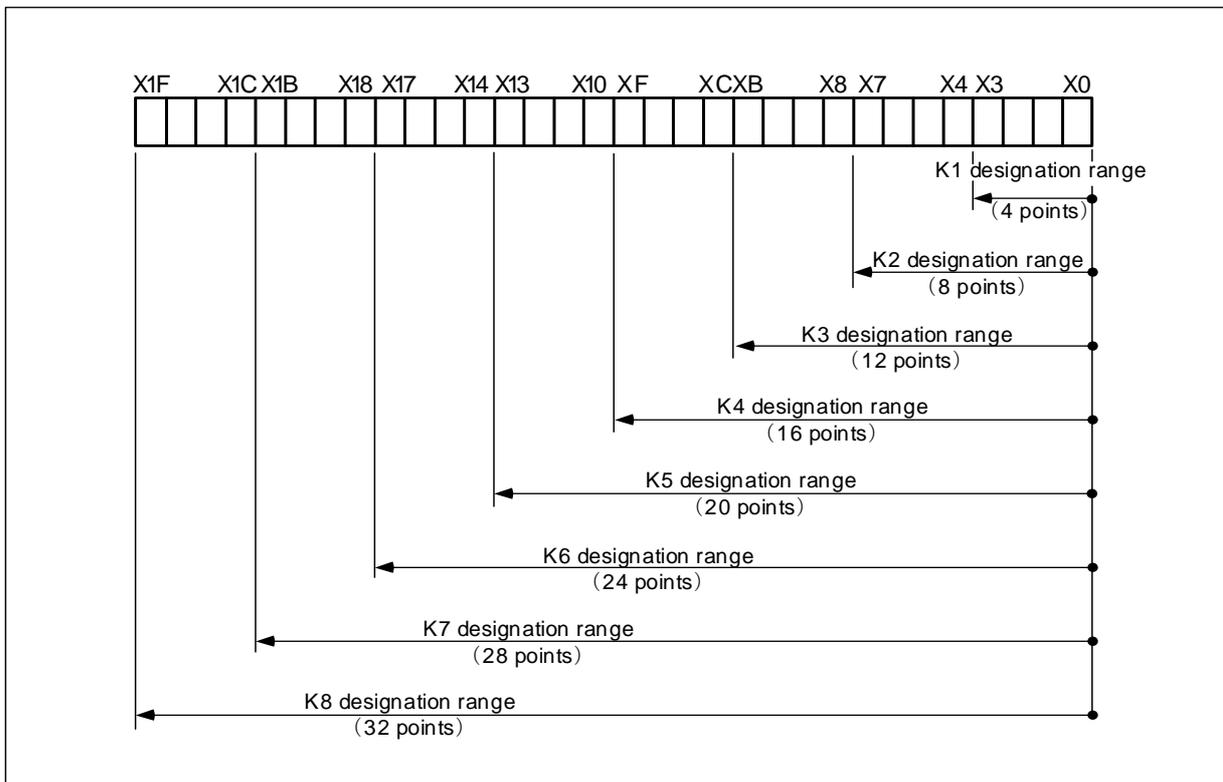
(a) 16-bit command: K1 to 4 (4 to 16 points)

(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.

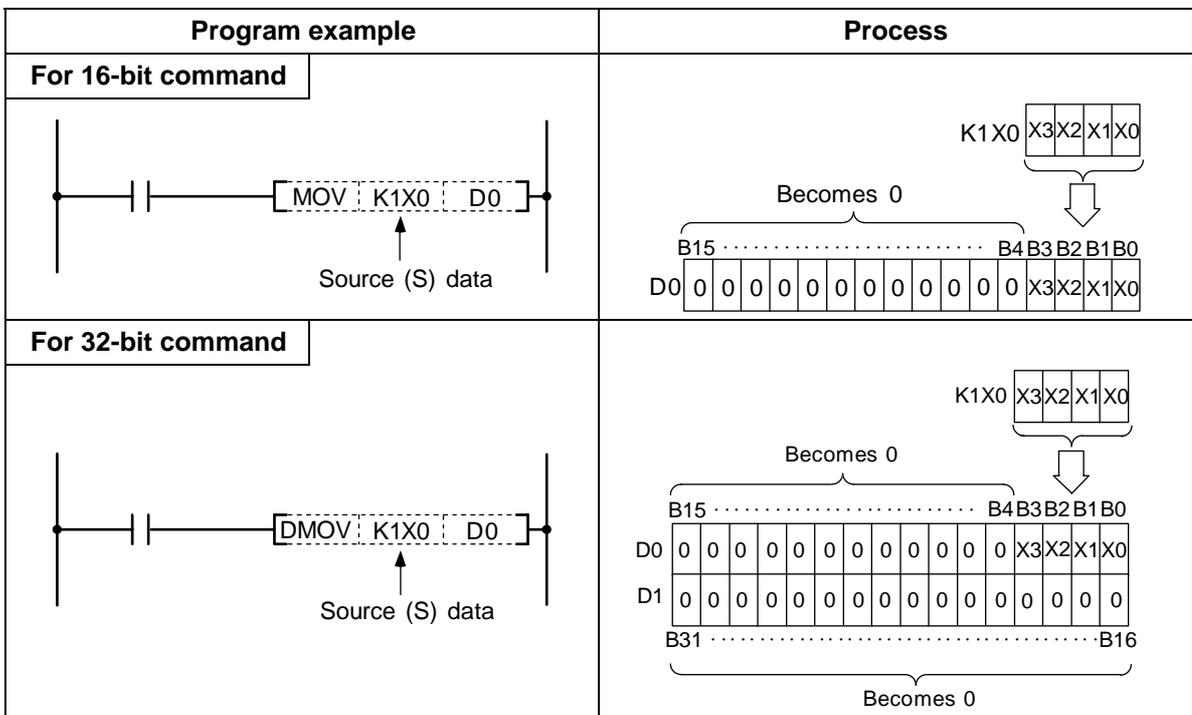


6. Explanation of Commands

- (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 |

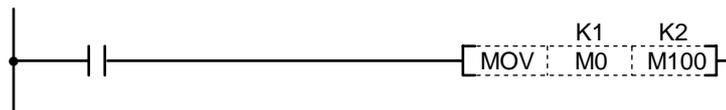


6. Explanation of Commands

- (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 |
|---|---------|
| <p style="text-align: center;">When source data (S) is a value</p> <p style="text-align: center;">Destination (D) side</p> | |
| <p style="text-align: center;">When source data (S) is a bit device</p> <p style="text-align: center;">(Note) Destination (D) side</p> | |
| <p style="text-align: center;">When source data (S) is a word device</p> <p style="text-align: center;">Destination (D) side</p> | |

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



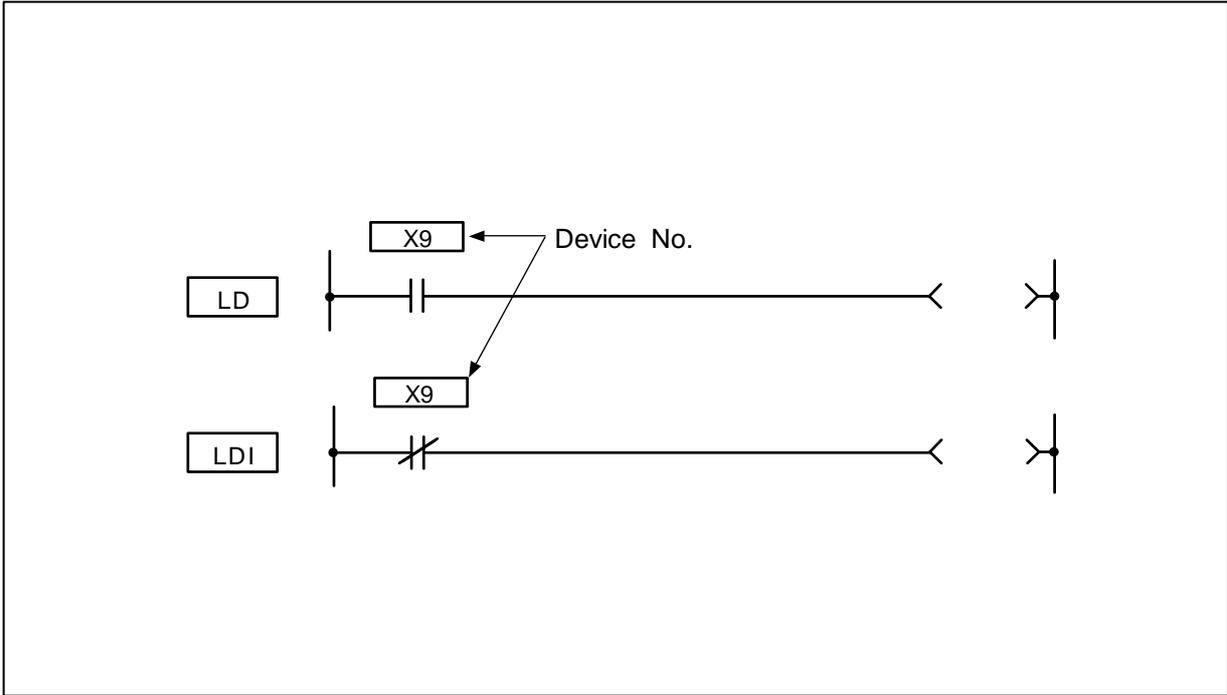
7. Basic Commands

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 | No. of steps | Index | | | | | |
|---------------|---|---|---|----|---|---|-------------|---|---|---|--|-----------|-------------------|--------------|-------|--|---------|-------|---|---|
| Bit device | | | | | | | Word device | | | | | Con-stant | | | | | Pointer | Level | | |
| X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | K | H | P | N |
| ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | | | 1 | | |



Function

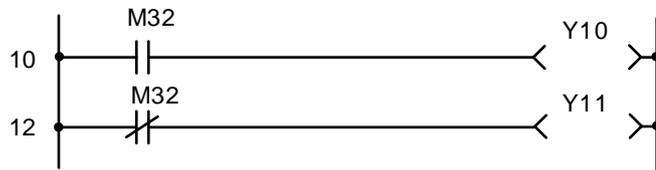
LD is the A contact operation start command and LDI is the B contact operation start command. The ON/OFF information of the designated device is read in as the operation results.

Execution conditions

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

Program example

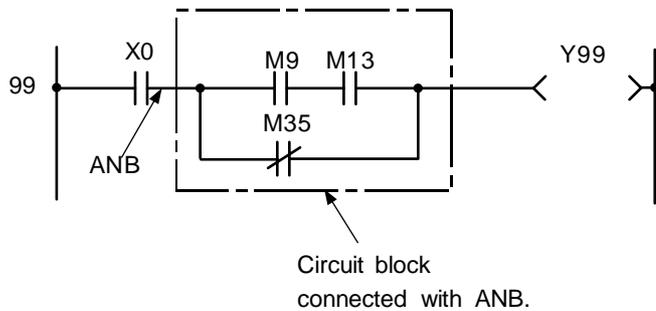
(1) Program used at head of circuit block.



Coding

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

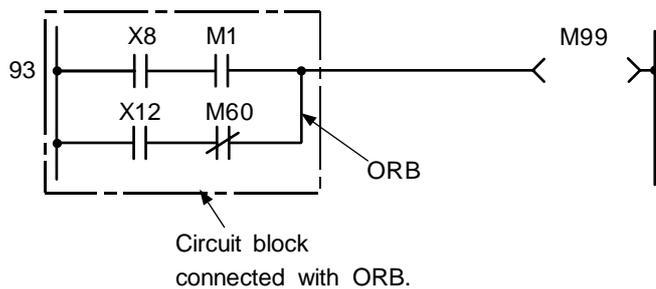
(2) Program used at head of circuit block connected with ANB.



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.

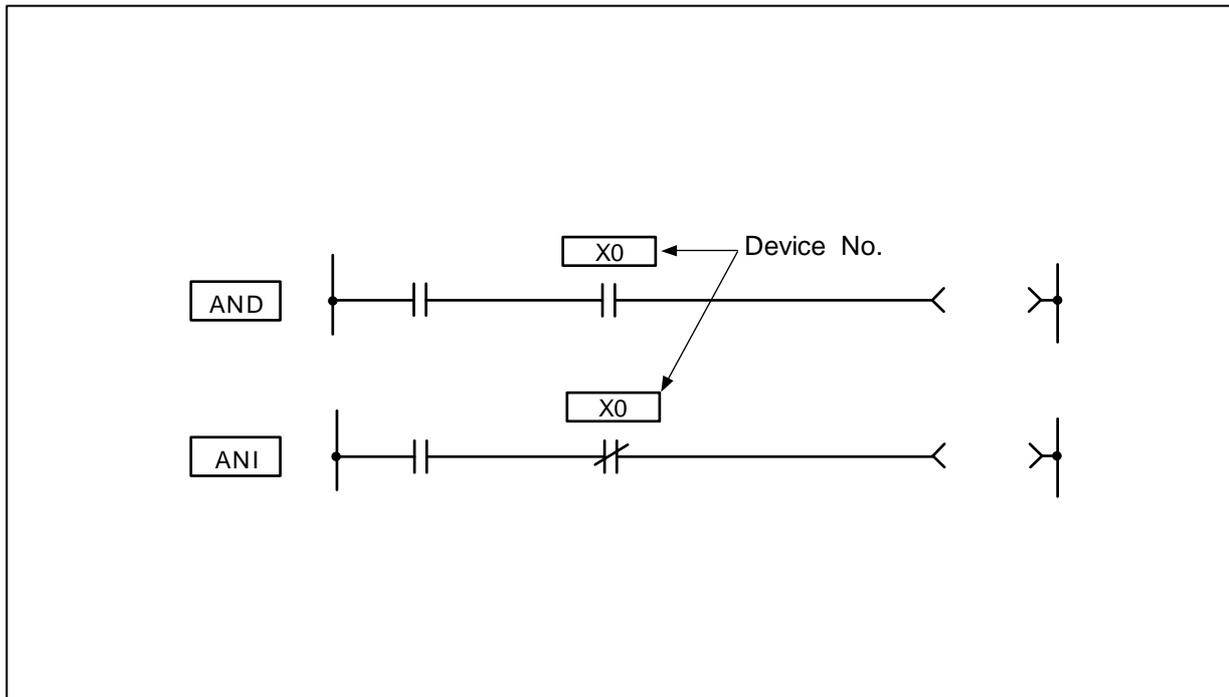


Coding

| 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 | No. of steps | Index | | | |
|---------------|---|---|---|----|---|---|-------------|---|---|---|-----------|--|---------|-------------------|--------------|-------|-------|---|--|
| Bit device | | | | | | | Word device | | | | Con-stant | | Pointer | | | | Level | | |
| X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | N | | | |
| ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | 1 | |



Function

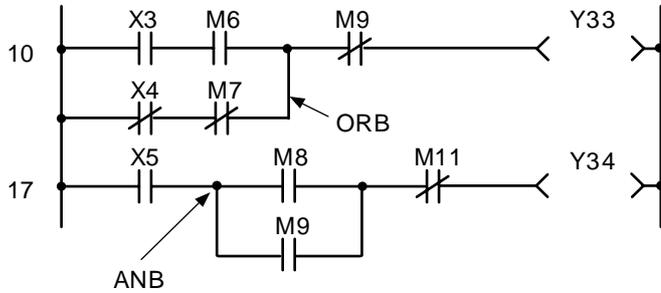
AND is the A contact serial connection command, and ANI is the B contact serial connection command. The ON/OFF information of the designated device is read in, and the AND operation with the operation results up to that point is executed. The result is the operation result.

Execution conditions

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

Program example

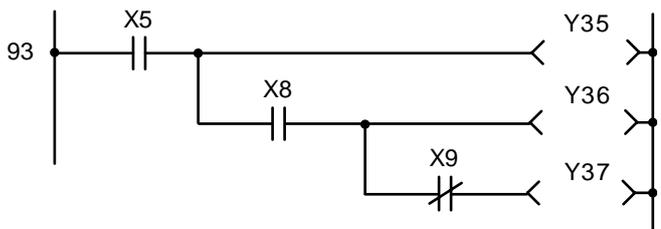
(1) Program used after LD, LDI, AND or ANI, etc.



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.

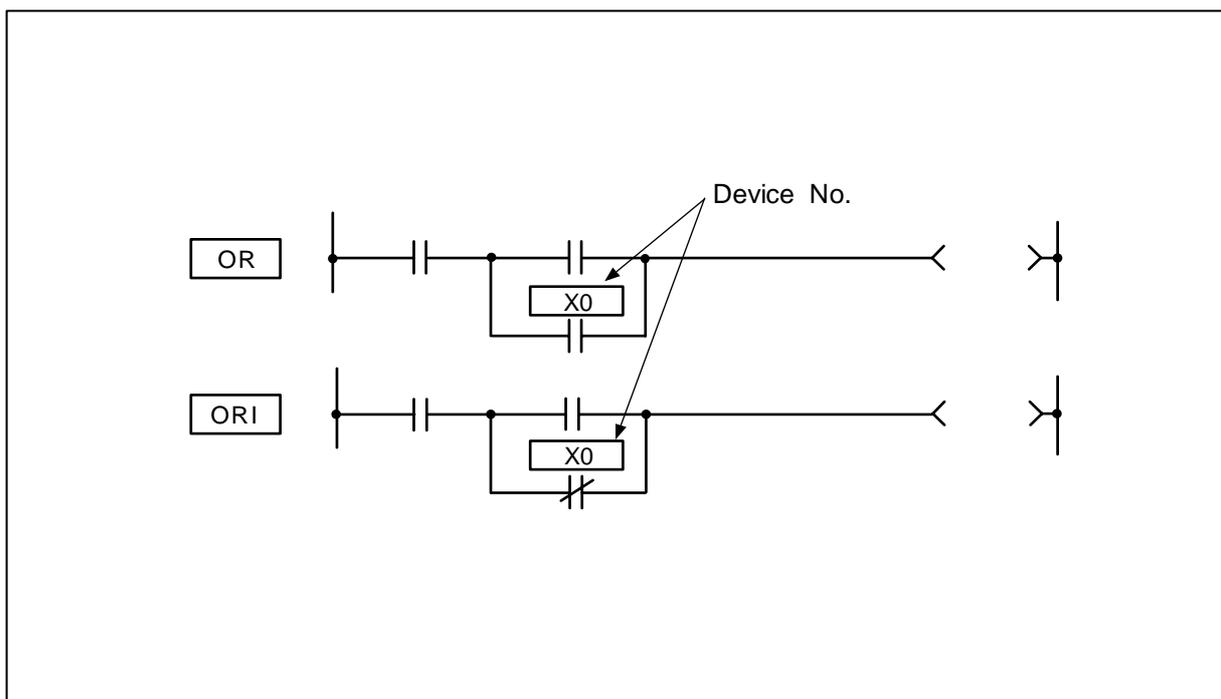


Coding

| 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 | No. of steps | Index | | | |
|---------------|---|---|---|----|---|---|---|-------------|---|---|--|-----------|---|-------------------|--------------|-------|---------|-------|---|
| Bit device | | | | | | | | Word device | | | | Con-stant | | | | | Pointer | Level | |
| X | Y | M | L | SM | F | T | C | D | R | Z | | | K | | | | H | P | N |
| ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | | | 1 | |



Function

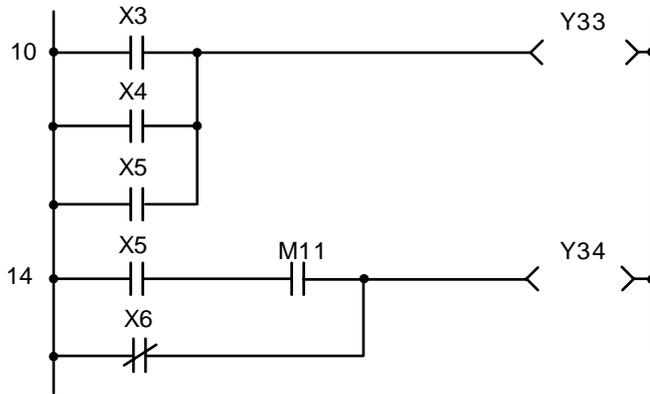
OR is the one A contact parallel connection command, and ORI is the one B contact parallel connection operation command. The ON/OFF information of the designated device is read in, and the OR operation with the operation results up to that point is executed. The result is the operation result.

Execution conditions

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

Program example

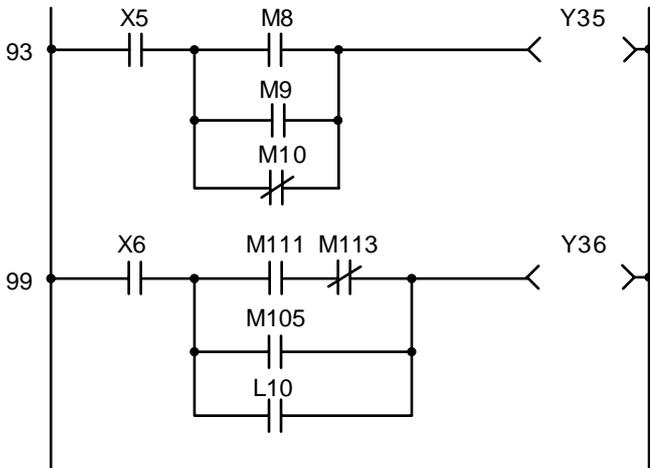
(1) Program used at head of circuit block.



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.

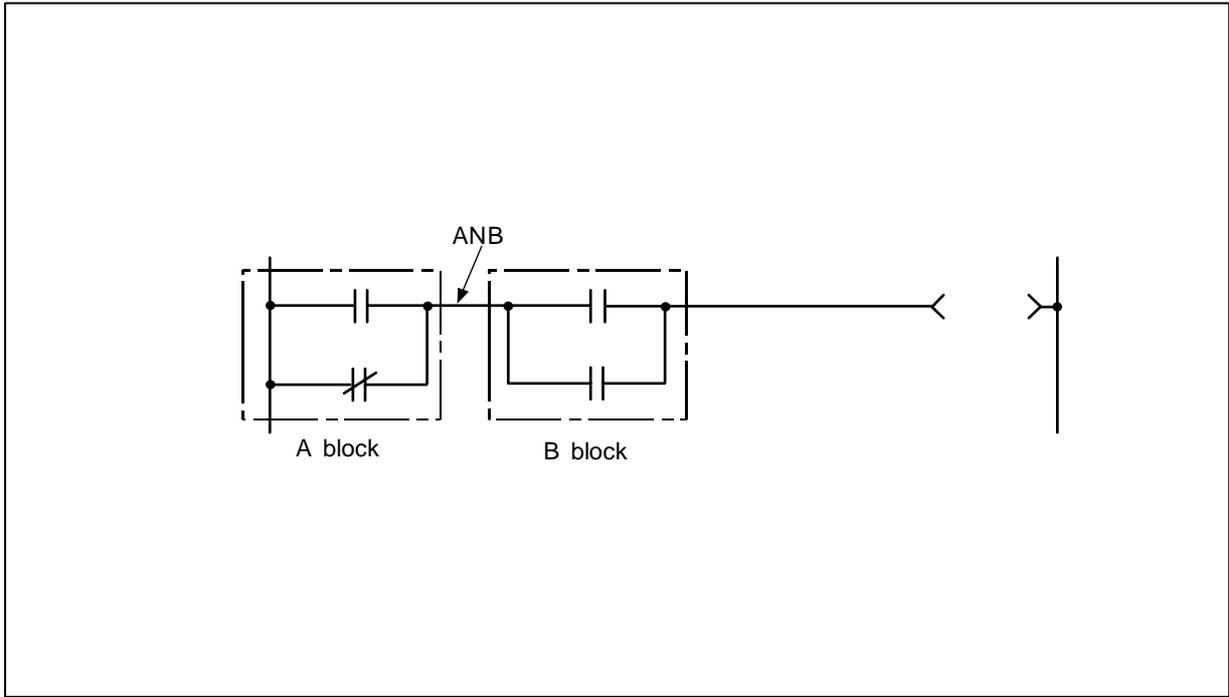


Coding

| 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 | No. of steps | Index | | | |
|---------------|---|---|---|----|---|---|-------------|---|---|---|--|--|---|-------------------|--------------|-------|----------|---------|-------|
| Bit device | | | | | | | Word device | | | | | | | | | | Constant | Pointer | Level |
| X | Y | M | L | SM | F | T | C | D | R | Z | | | K | | | | H | P | N |
| | | | | | | | | | | | | | | | | | 1 | | |

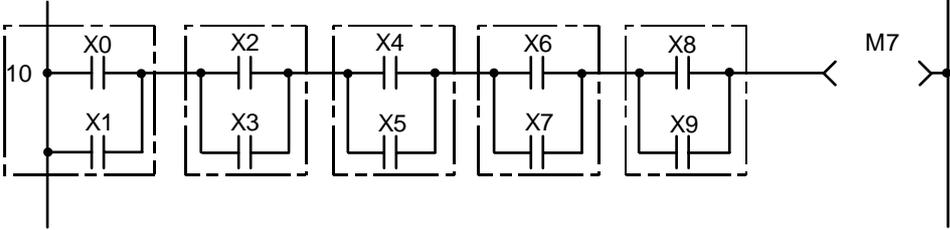


Function

- (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 PC cannot execute a correct operation if 8 or more commands are written consecutively.

Program example

Program that serially connects continuous circuit blocks.

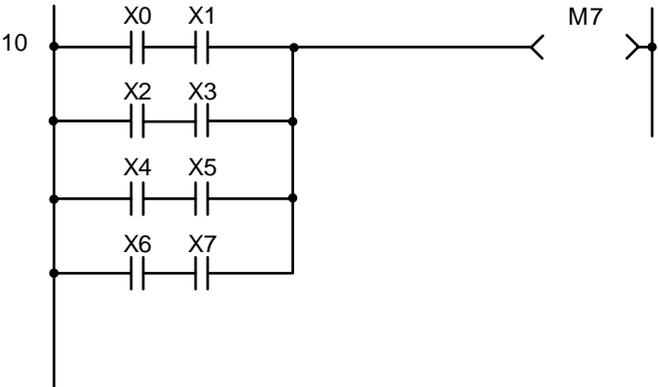


Coding

| 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 | | | | |

Program example

Program that connects continuous circuit blocks in parallel.

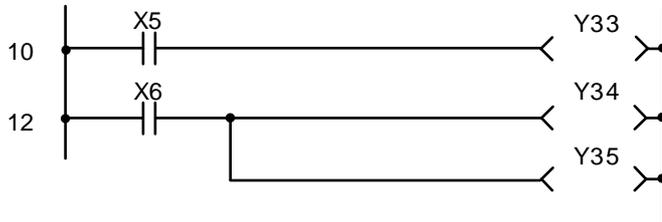


Coding

| 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 | | | | |

Program example

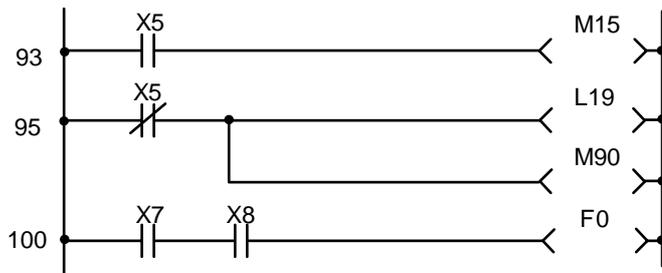
(1) Program output to output unit.



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.

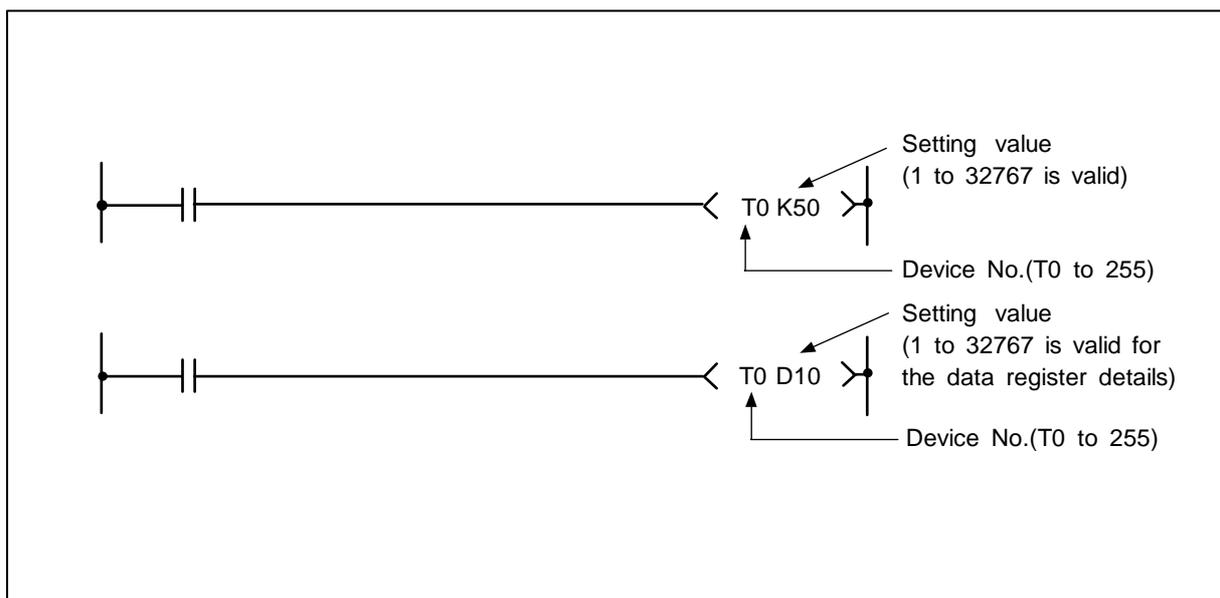


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|--|--|
| 93 | LD | X5 | | |
| 94 | OUT | M15 | | |
| 95 | LDI | X5 | | |
| 96 | OUT | L19 | | |
| 97 | OUT | M90 | | |
| 100 | LD | X7 | | |
| 101 | AND | X8 | | |
| 102 | OUT | F0 | | |
| 103 | | | | |

○ OUT T ... Timer output

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | |
|---------------|---------------|---|---|---|----|---|---|-------------|---|---|---|--|--|--|-------------------|--------------|-------|----------|---------|-------|
| | Bit device | | | | | | | Word device | | | | | | | | | | Constant | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | | K | H |
| Device | | | | | | ○ | | | | | | | | | | | | | | |
| 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 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 integrated timer | OFF | Hold current value | Non-continuity | Continuity | Continuity | Non-continuity |

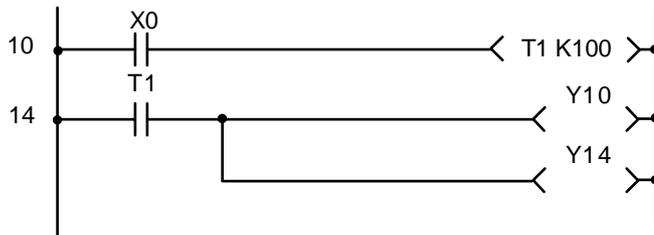
- (3) The state of the integrated timer contact after time up will not change until the RST command is executed.

Execution condition

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

Program example

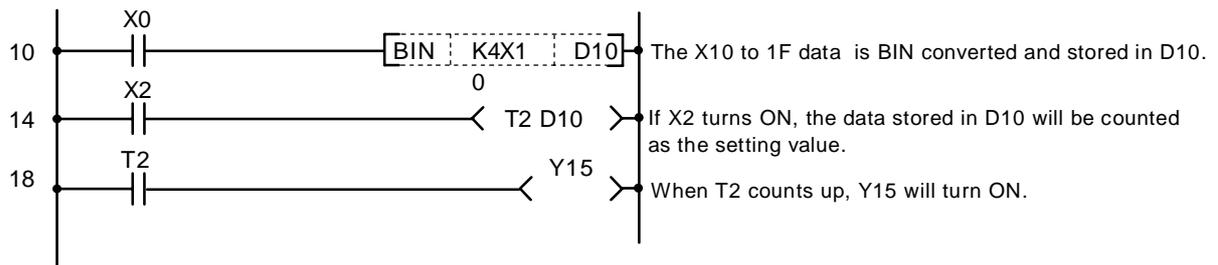
(1) Program to turn ON Y10 and Y14 ten seconds after X0 turns ON.



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.

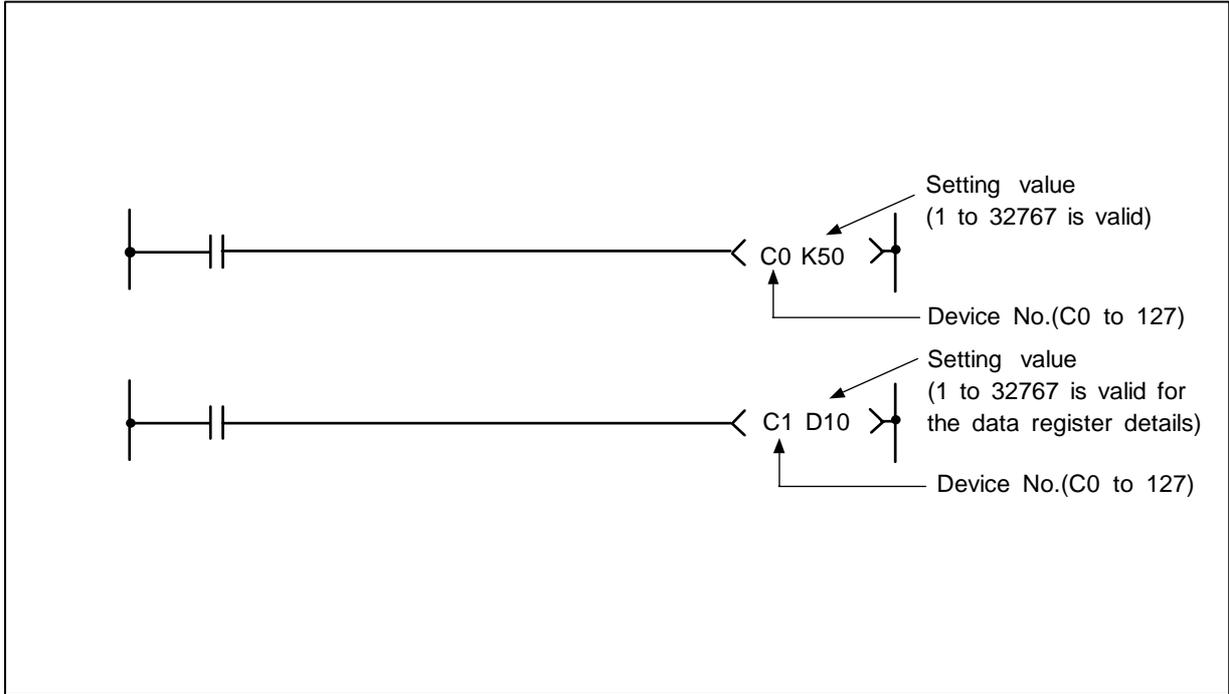


Coding

| 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 | No. of steps | Index | | | | |
|---------------|---------------|---|---|---|----|---|---|-------------|---|---|---|--|-----------|---------|-------------------|--------------|-------|-------|---|---|--|
| | Bit device | | | | | | | Word device | | | | | Con-stant | Pointer | | | | Level | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | | | | H | P | N | |
| Device | | | | | | | ○ | | | | | | | | | | | | | | |
| 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 |

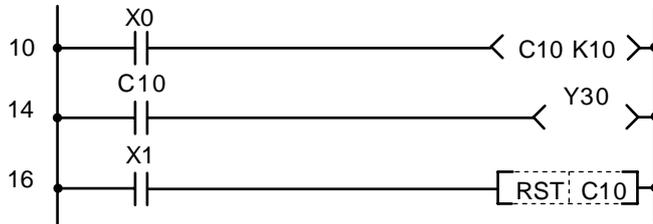
- (2) The value will not be counted when the operation results are ON. (A pulse change is not required to input the count.)
- (3) If the operation results change from OFF to ON after the "current value >= setting value" is established, the contact state will remain the same, however the current value will be incremented by 1.

Execution condition

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

Program example

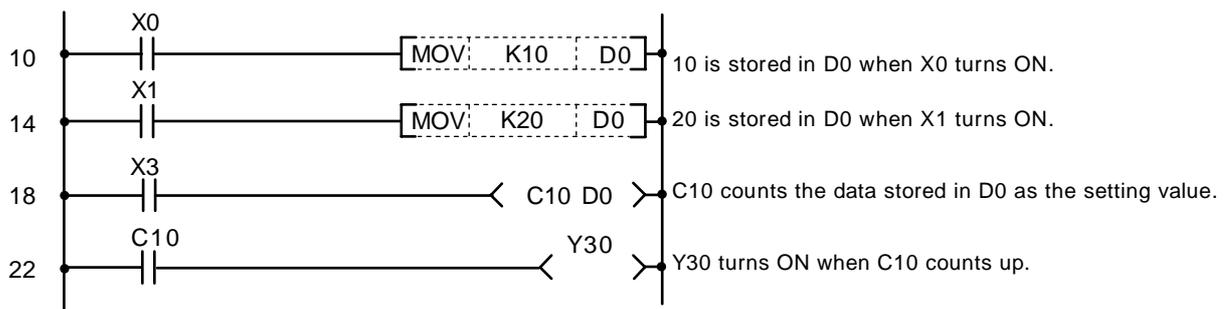
(1) Program to turn Y30 ON when X0 turns ON ten times, and to turn Y30 OFF when X1 turns ON.



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.

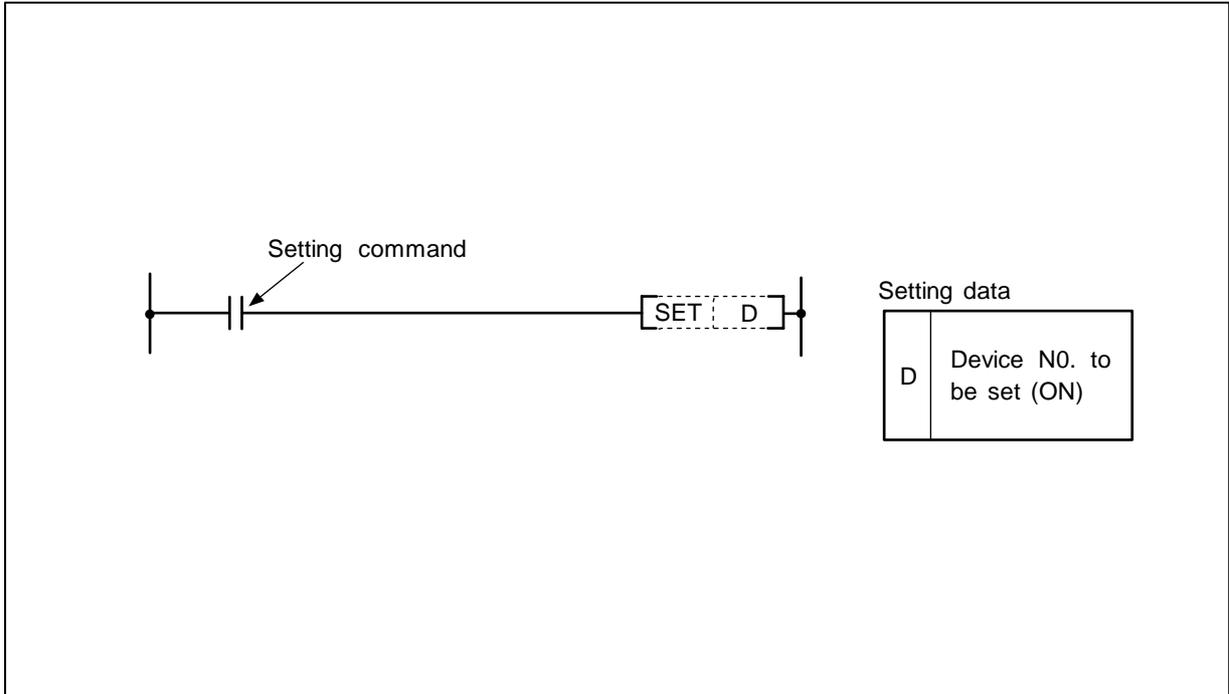


Coding

| 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 | | | | |

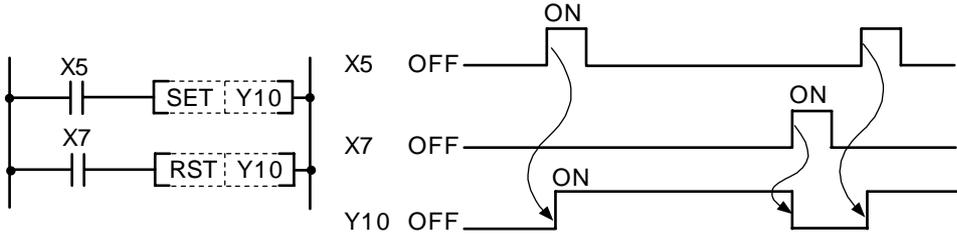
○ SET ... Device setting (ON)

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | |
|----------|---------------|---|---|---|----|---|---|-------------|---|---|---|--|--|---|-------------------|--------------|-------|----------|----------|-------|
| | Bit device | | | | | | | Word device | | | | | | | | | | Constant | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | | | | H | P | N |
| D | | ○ | ○ | ○ | ○ | ○ | | | | | | | | | | | | | 1 | |



Function

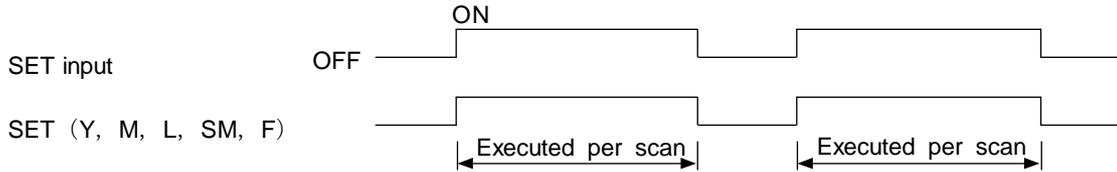
- (1) The designated device turns ON when the SET input turns ON.
- (2) The device turned ON remains ON even if the SET input turns OFF. The device can be turned OFF with the RST command.



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

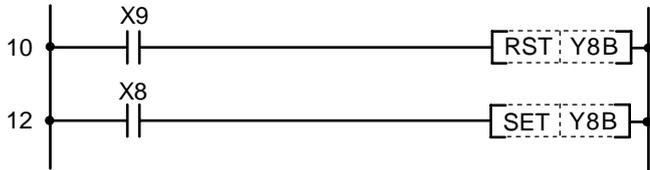
Execution condition

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



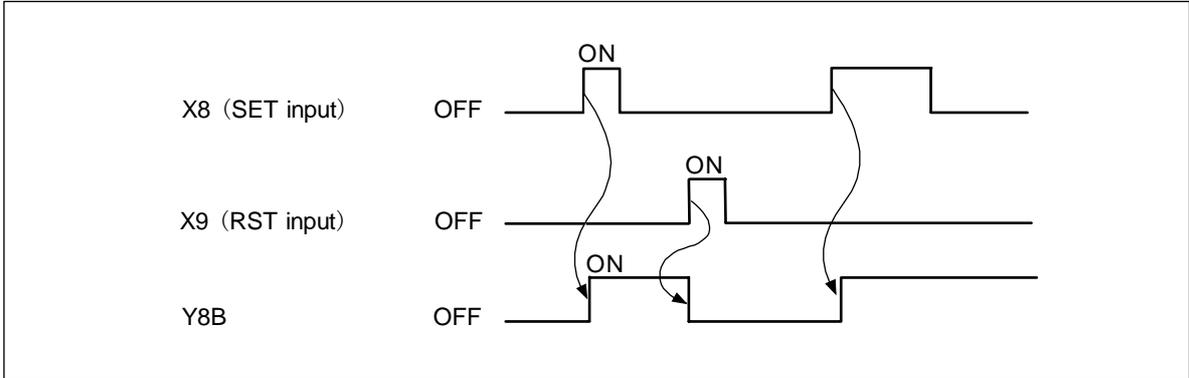
Program example

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



Coding

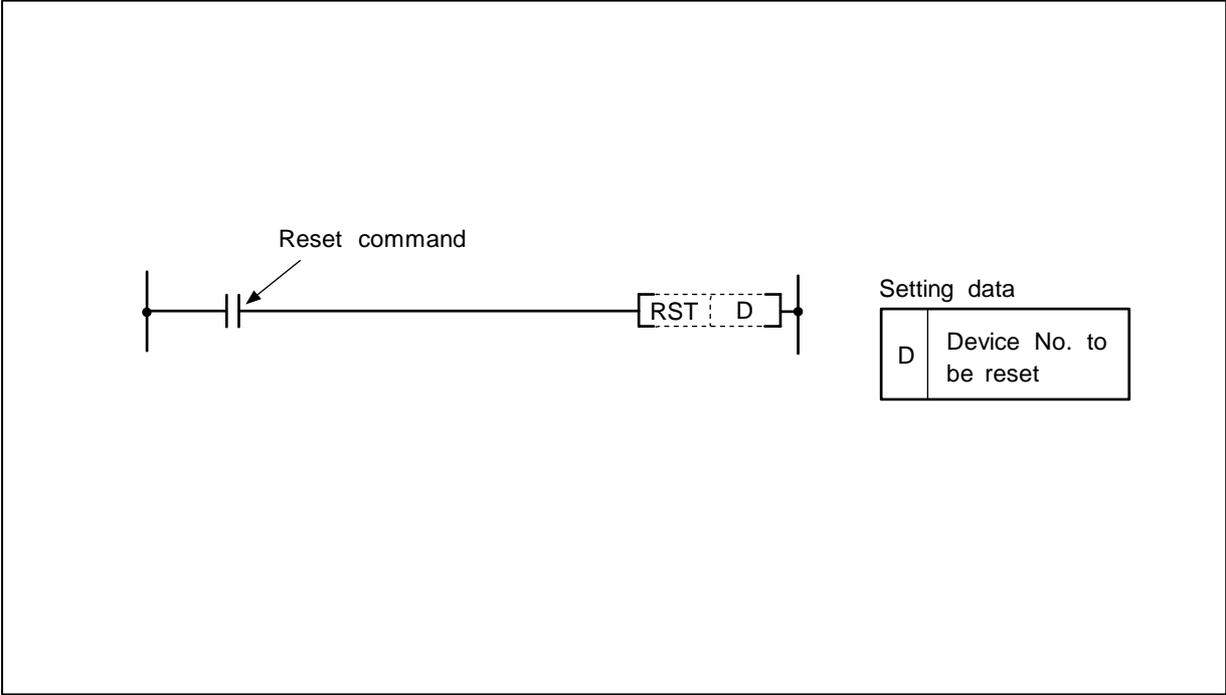
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|--|--|
| 10 | LD | X9 | | |
| 11 | RST | Y8B | | |
| 12 | LD | X8 | | |
| 13 | SET | Y8B | | |
| 14 | | | | |



Operation of SET and RST commands

○ RST ... Device resetting

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



Function

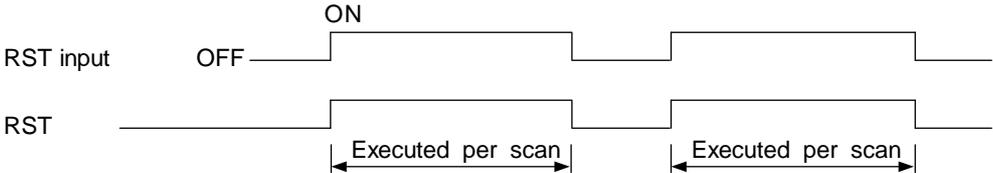
(1) The designated device will change as explained below when the RST input turns ON.

| 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.

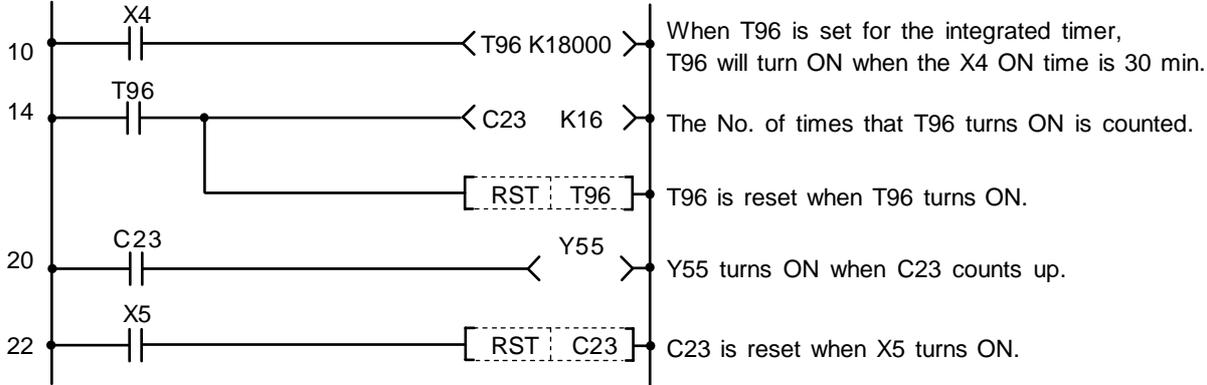
Execution condition

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



Program example

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



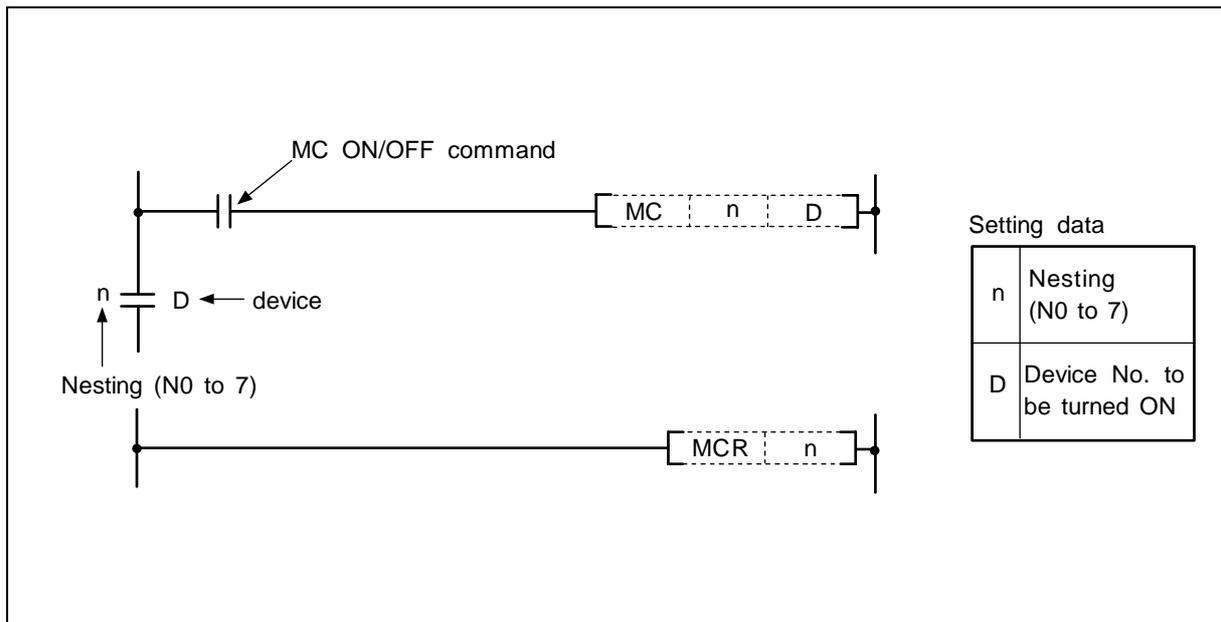
2 steps are used for T or C device.
1 step is used for the other devices.

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 | No. of steps | Index | | | | | |
|---|---------------|---|---|---|----|---|---|-------------|---|---|---|--|--|--|-------------------|--------------|-------|-----------|---------|-------|---|---|
| | Bit device | | | | | | | Word device | | | | | | | | | | Con-stant | Pointer | Level | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | | K | H | P | N |
| n | | | | | | | | | | | | | | | | | | ○ | | | | |
| D | | ○ | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | | | |



Function

MC

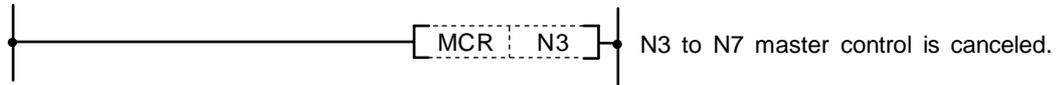
- (1) If the MC ON/OFF command is ON when the master control starts, the operation results between MC and MCR will remain the same.
- (2) If the MC ON/OFF command is OFF, the operation results between MC and MCR will be as follows.

| 100ms, 10ms timer | 100ms integrated 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 7) 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 for the destination will turn ON.

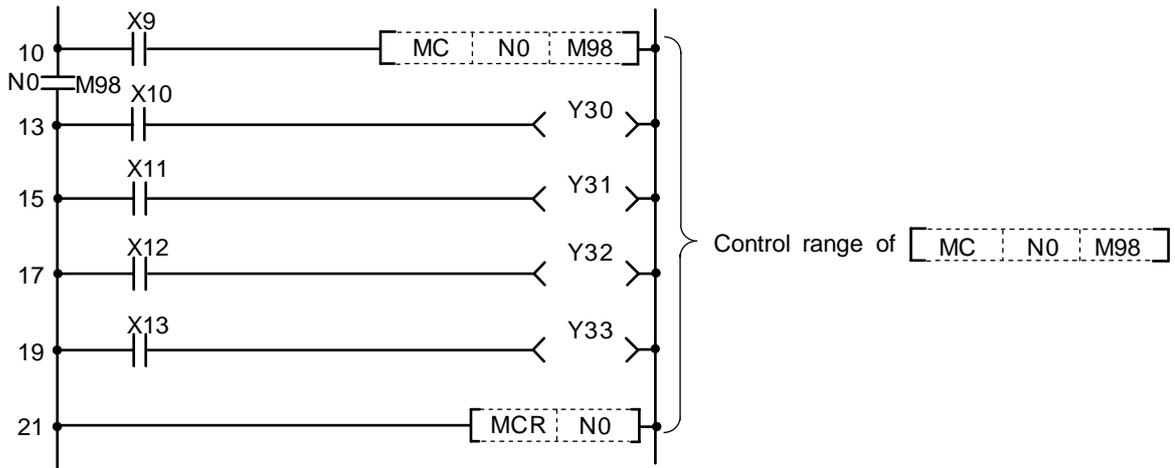
MCR

- (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.

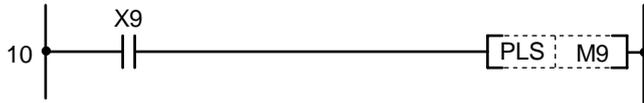


Coding

| 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 | | | | |

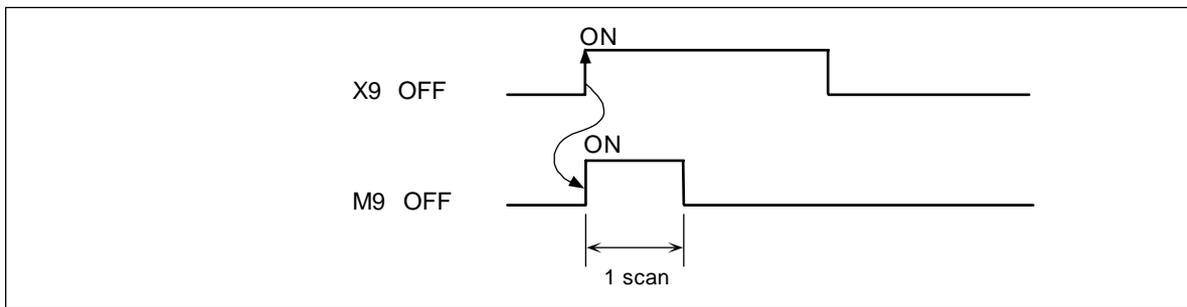
Program example

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



Coding

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

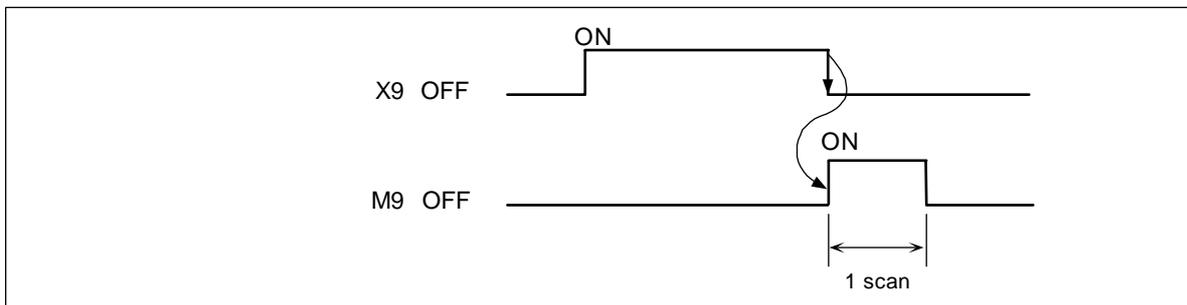


(2) Program to execute PLF command when X9 turns OFF.



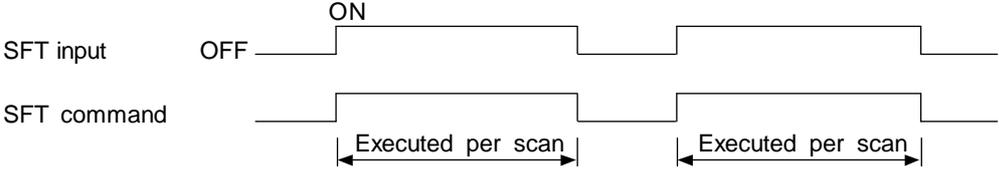
Coding

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



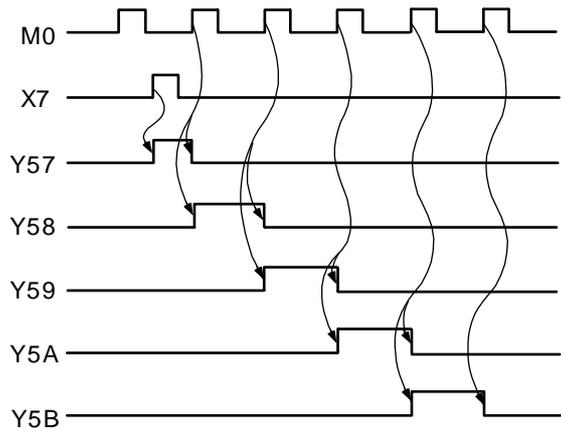
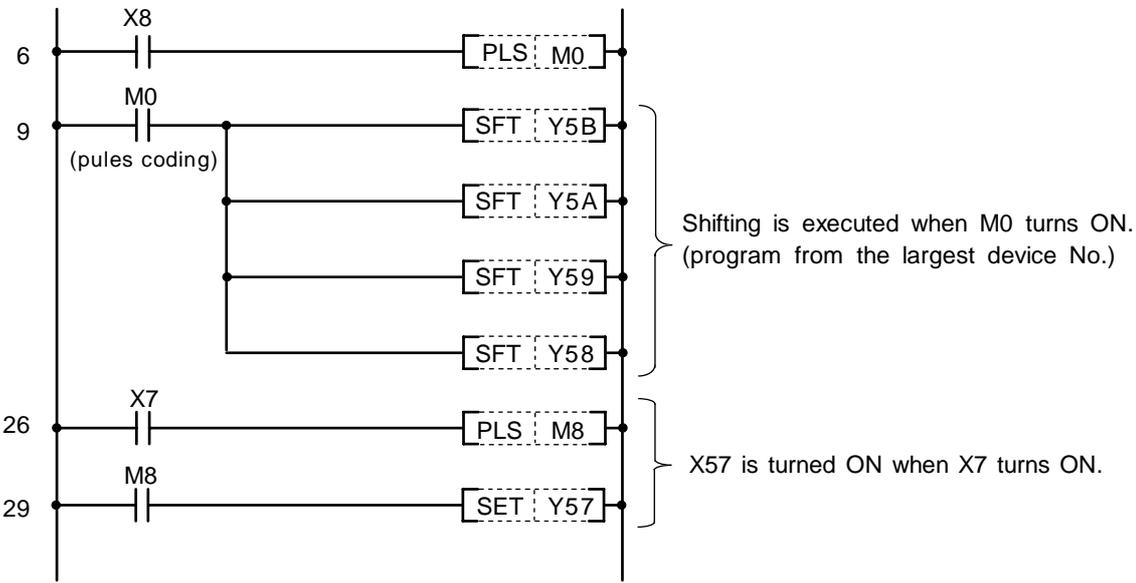
Execution condition

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



Program example

(1) Program to shift Y57 to 5B when X8 turns ON.

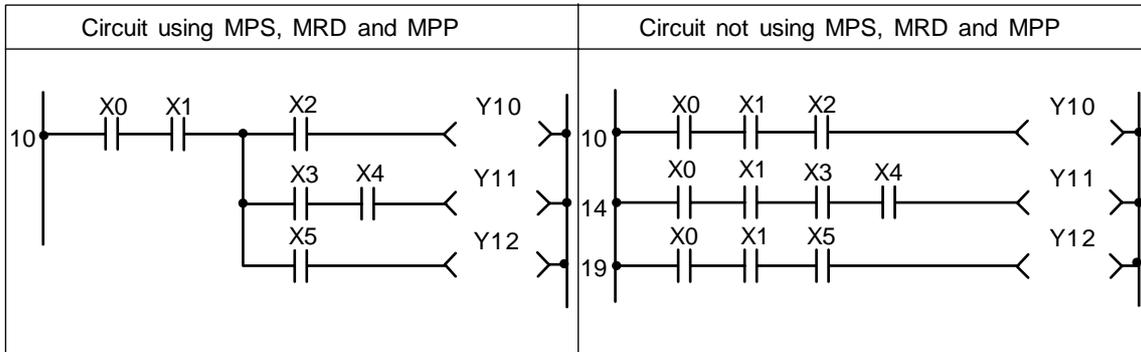


Coding

| No. of steps | Com- mand | Device | | |
|--------------|--------------|--------|--|--|
| 6 | LD | X8 | | |
| 7 | PLS | M0 | | |
| 9 | LD | M0 | | |
| 10 | SFT | Y5B | | |
| 14 | SFT | Y5A | | |
| 18 | SFT | Y59 | | |
| 22 | SFT | Y58 | | |
| 26 | LD | X7 | | |
| 27 | PLS | M8 | | |
| 29 | LD | M8 | | |
| 30 | SET | Y57 | | |
| 31 | | | | |

Point

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



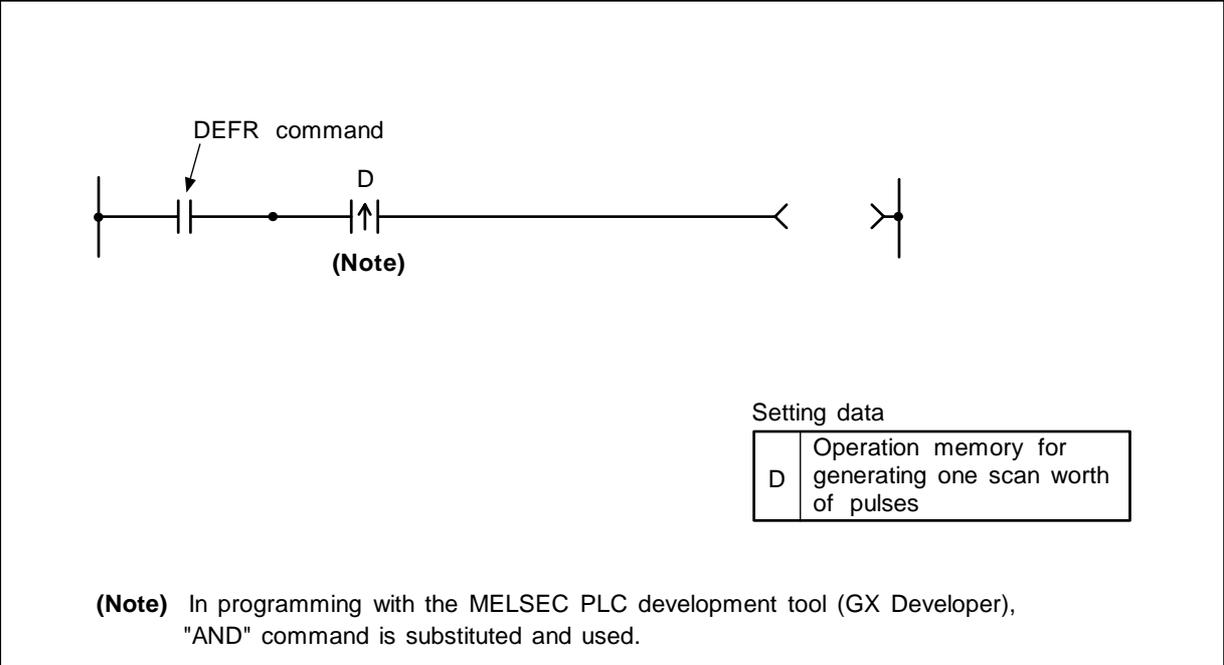
(1) Program using MPS, MRD and MPP.

Coding

| No. of steps | Command | 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 | | | | |

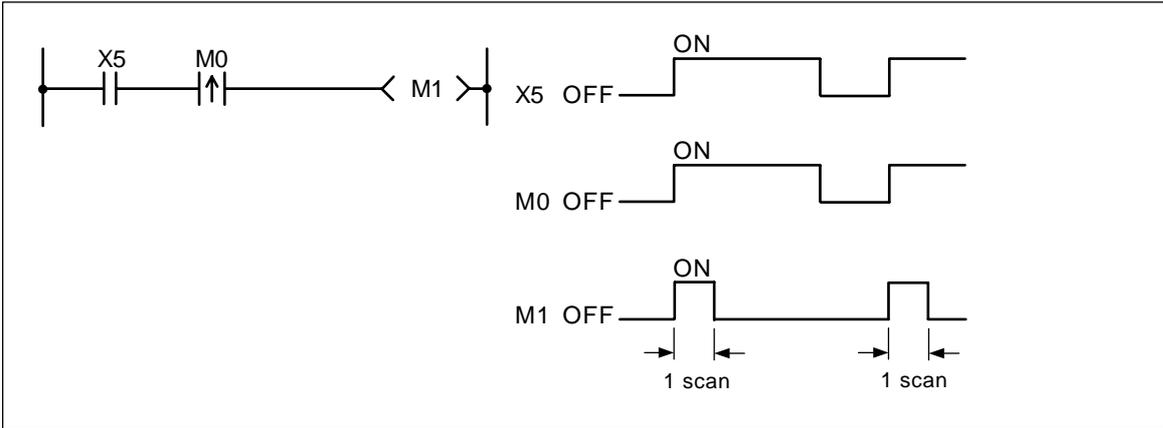
○ DEFR ... Pulses in regard to operation results

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | | |
|----------|---------------|---|---|---|----|---|---|-------------|---|---|---|--|--|--|-------------------|--------------|-------|-----------|---------|-------|---|---|
| | Bit device | | | | | | | Word device | | | | | | | | | | Con-stant | Pointer | Level | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | | K | H | P | N |
| D | | ○ | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | 1 | | |



Function

The operation results are turned ON for one scan when the DEFR command is turned from OFF to ON, and are turned OFF for all other cases.

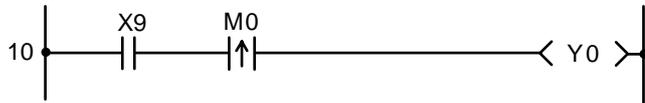


Execution conditions

This is executed per scan regardless of the operation results to the DEFR command.

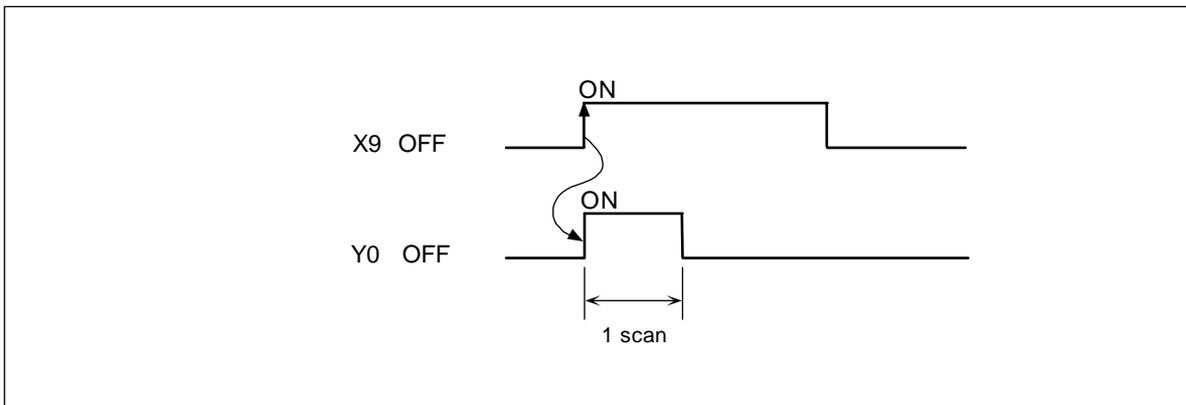
Program example

(1) Program to turn Y0 ON for one scan when X9 turns ON.

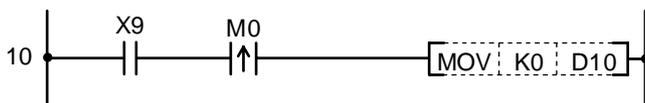


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|--|--|
| 10 | LD | X9 | | |
| 11 | ANDP | M0 | | |
| 12 | OUT | Y0 | | |
| 13 | | | | |



(2) Program to execute MOV command once when X9 turns ON.



Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-----|--|
| 10 | LD | X9 | | |
| 11 | ANDP | M0 | | |
| 12 | MOV | K0 | D10 | |
| 15 | | | | |

8. Function Commands

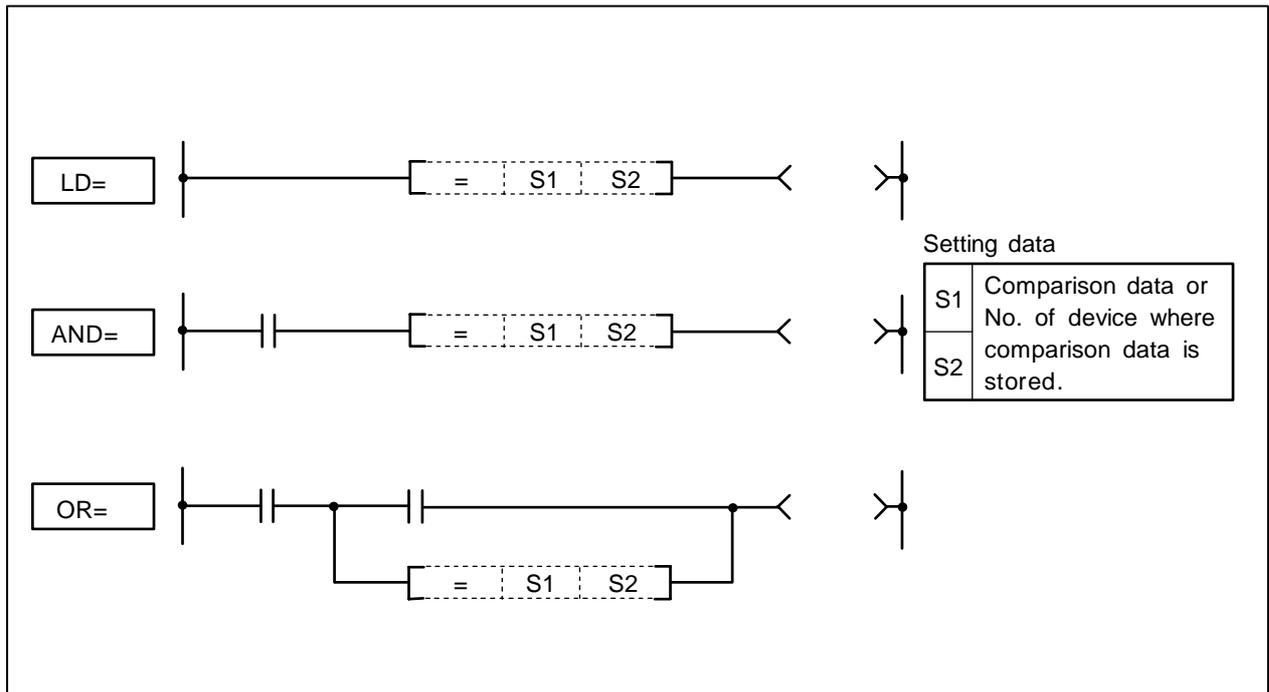
Recent sequence programs that require more advanced control cannot provide sufficient control only with basic commands and thus need four-rule operation and comparison, etc.

Many function commands have been prepared for this. There are approx. 76 types of function commands.

Each command is explained in the following section.

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

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|--|---------|-------|-------------------|--------------|-------|---|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | K | | | | H | P |
| S1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | ○ | ○ | | | ○ | 3 | |
| 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 |

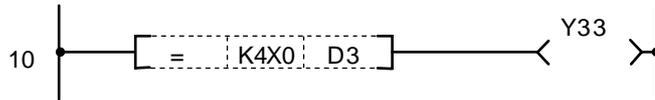
Execution conditions

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 |

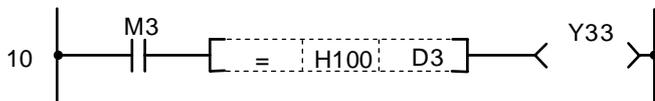
Program example

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

**Coding**

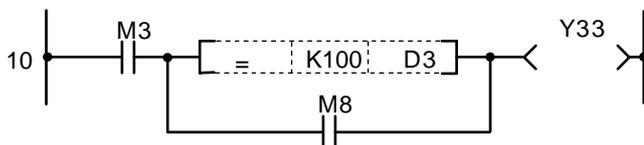
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD= | K4X0 | D3 | |
| 13 | OUT | Y33 | | |
| 14 | | | | |

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

**Coding**

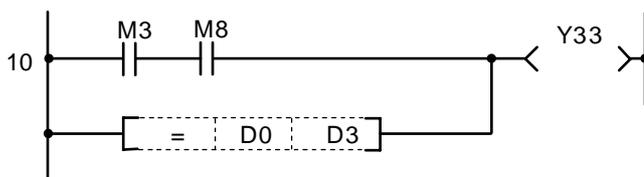
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M3 | | |
| 11 | AND= | H100 | D3 | |
| 14 | OUT | Y33 | | |
| 15 | | | | |

- (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 | | | | |

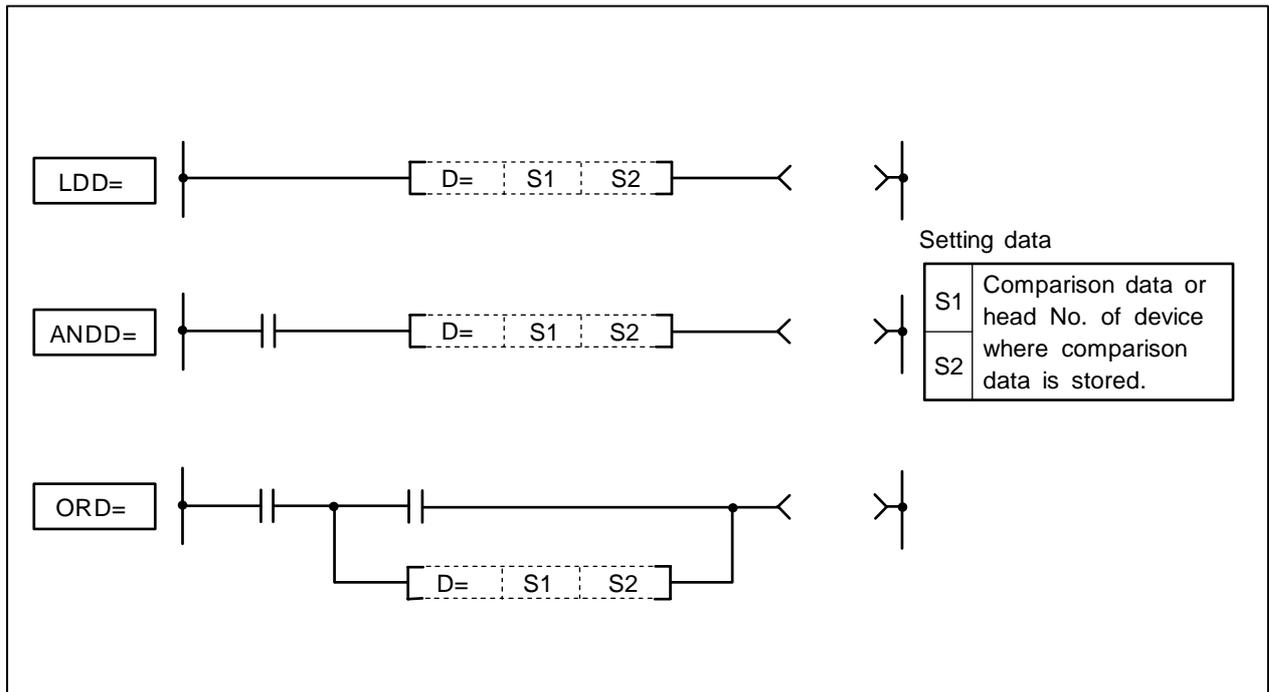
- (4) Program to compare the D0 and D3 data.

**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 | No. of steps | Index | |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|--|---------|-------|-------------------|--------------|-------|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | K | | | | H |
| S1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | ○ | ○ | | | | |
| 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 |

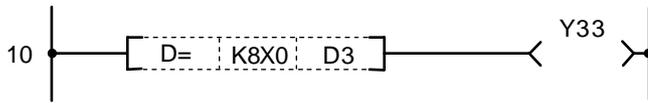
Execution conditions

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

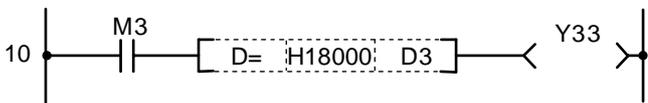
(1) Program to compare the X0 to 1F data, D3 and D4 data.



Coding

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

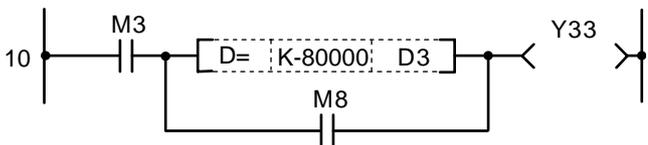
(2) Program to compare the BCD value 18000, D3 and D4 data.



Coding

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

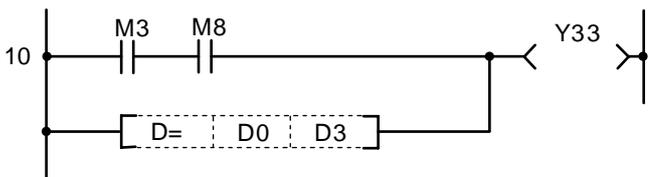
(3) Program to compare the BIN value -80000, D3 and D4 data.



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, D3 and D4 data.

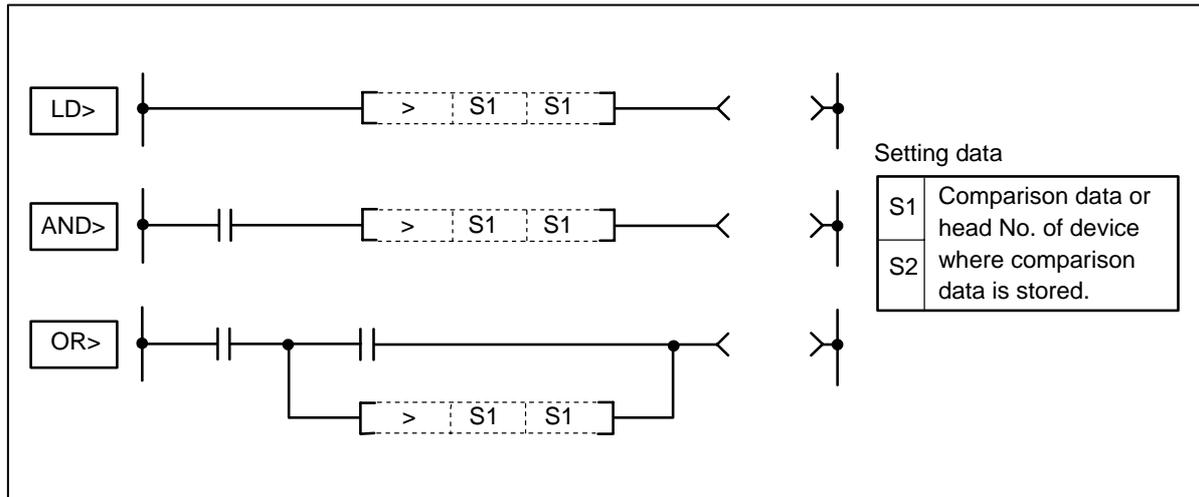


Coding

| 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 | No. of steps | Index | | |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|--|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | |
| S1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | | | | | ○ | 3 | |
| 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 |

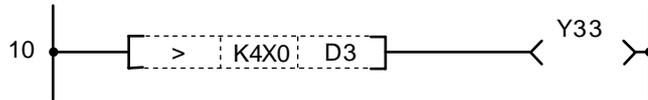
Execution conditions

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 |

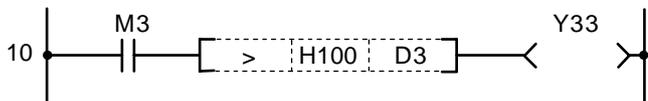
Program example

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

**Coding**

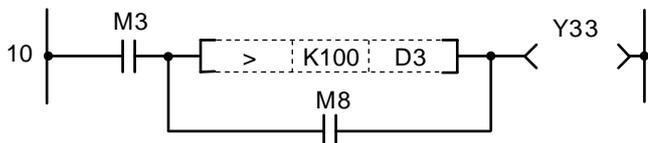
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD> | K4X0 | D3 | |
| 13 | OUT | Y33 | | |
| 14 | | | | |

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

**Coding**

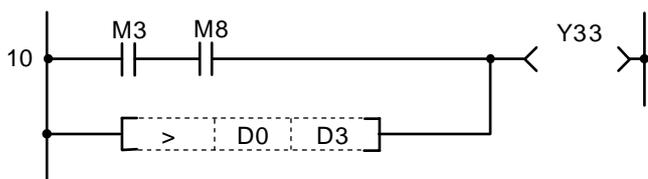
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M3 | | |
| 11 | AND> | H100 | D3 | |
| 14 | OUT | Y33 | | |
| 15 | | | | |

(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 | | | | |

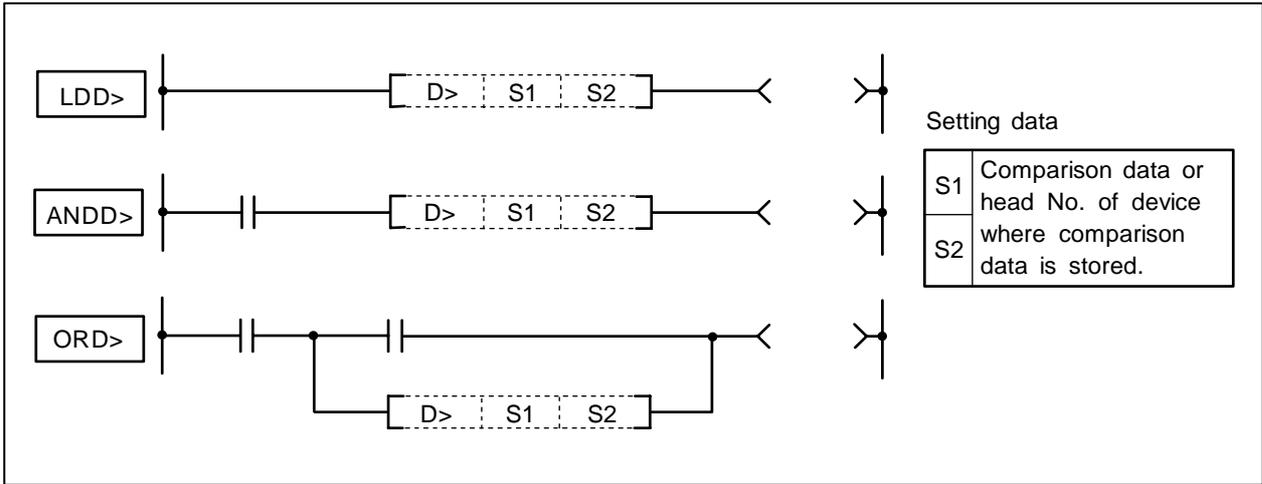
(4) Program to compare the D0 and D3 data.

**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 | No. of steps | Index | | |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|--|---------|-------|-------------------|--------------|-------|-----|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | K | | | | H | P |
| S1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | ○ | ○ | | | ○ | 3/4 | |
| 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 |

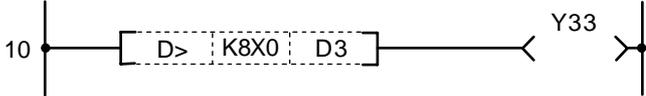
Execution conditions

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

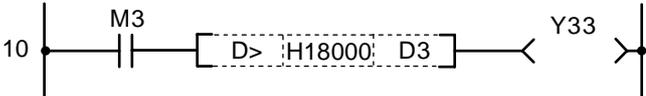
(1) Program to compare the X0 to 1F data, D3 and D4 data.



Coding

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

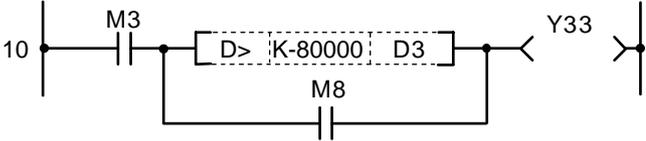
(2) Program to compare the BCD value 18000, D3 and D4 data.



Coding

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

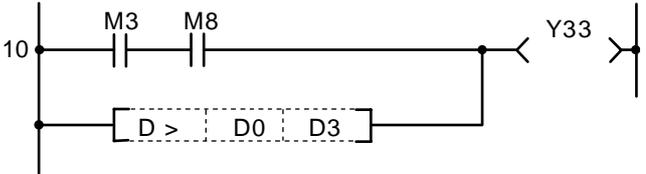
(3) Program to compare the BIN value -80000, D3 and D4 data.



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, D3 and D4 data.

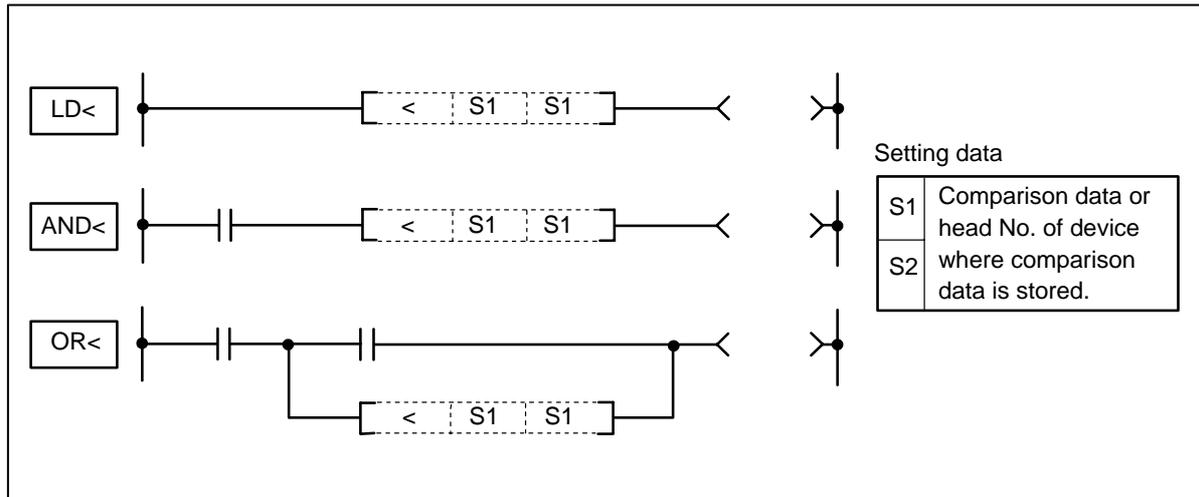


Coding

| 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 | No. of steps | Index | |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---------|-------|---|-------------------|--------------|-------|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | Pointer | Level | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N |
| S1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | ○ | ○ | | | ○ | 3 | |
| 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 |

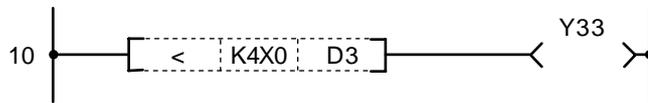
Execution conditions

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 |

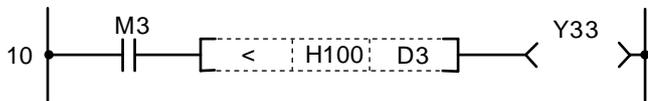
Program example

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

**Coding**

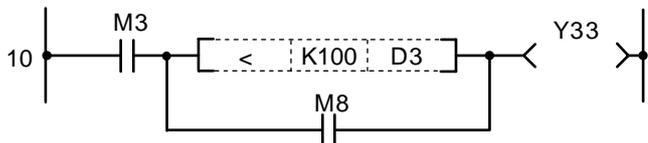
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| | | | | |
| 10 | LD< | K4X0 | D3 | |
| 13 | OUT | Y33 | | |
| 14 | | | | |

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

**Coding**

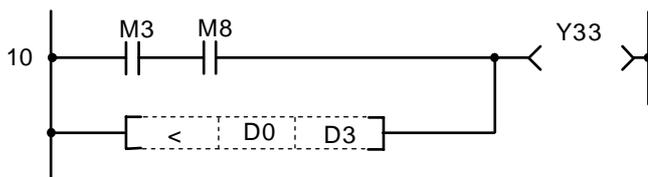
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| | | | | |
| 10 | LD | M3 | | |
| 11 | AND< | H100 | D3 | |
| 14 | OUT | Y33 | | |
| 15 | | | | |

- (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 | | | | |

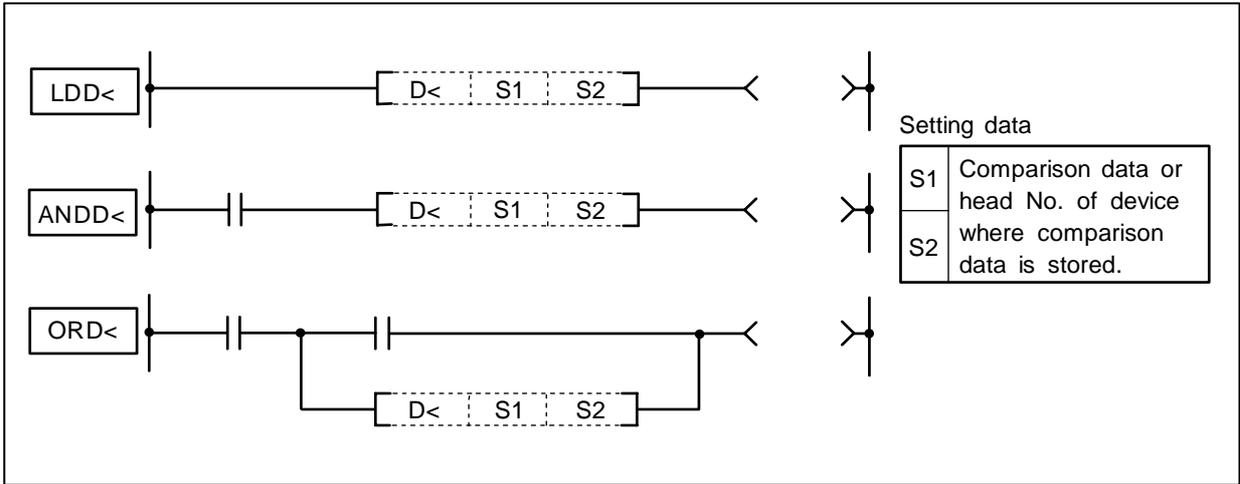
- (4) Program to compare the D0 and D3 data.

**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 | No. of steps | Index | |
|----|---------------|---|---|---|----|---|---|-------------|---|---|---|--|-----------|---|---------|-------|---|-------------------|--------------|-------|--|
| | Bit device | | | | | | | Word device | | | | | Con-stant | | Pointer | Level | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | N | | | | |
| S1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | ○ | ○ | | | ○ | 3/4 | |
| 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 |

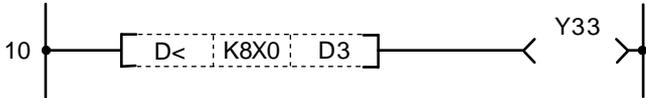
Execution conditions

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

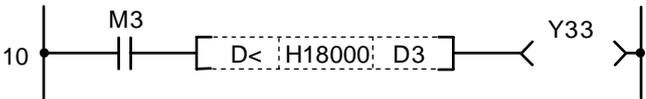
(1) Program to compare the X0 to 1F data, D3 and D4 data.



Coding

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

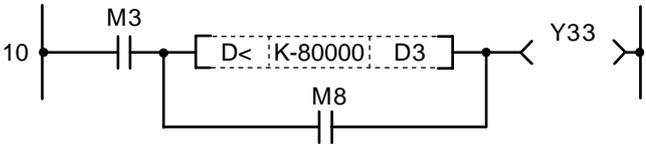
(2) Program to compare the BCD value 18000, D3 and D4 data.



Coding

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

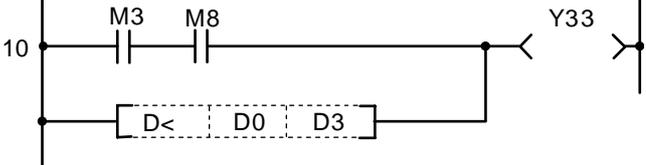
(3) Program to compare the BIN value -80000, D3 and D4 data.



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, D3 and D4 data.



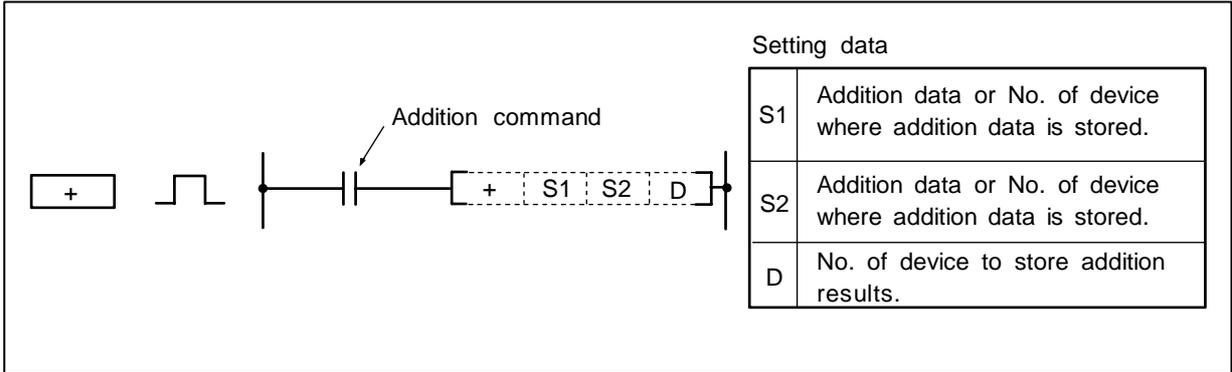
Coding

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

+

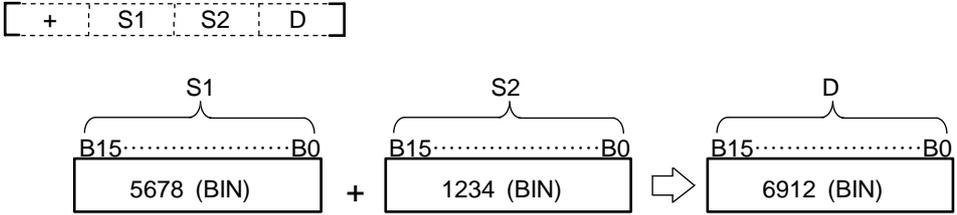
○+ ... BIN 16-bit addition

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index |
|-----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---|---------|-------|-------------------|--------------|-------|
| | Bit device | | | | | | Word device | | | | | | Constant | | Pointer | Level | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | |
| S1 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| S2 | | | | | | | ○ | ○ | ○ | ○ | | | | | ○ | ○ | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |



Function

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



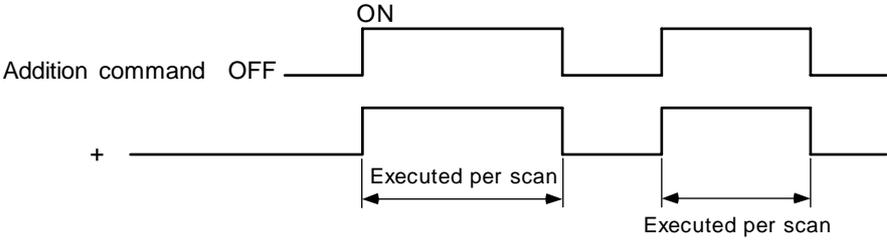
- (2) -32768 to 32767 (BIN 16-bit) can be designated in S1 and S2.
 (3) The positive/negative of the data in S1, S2 and D is determined with the highest-order bit (B15).

| B15 | Judgment of positive/negative |
|-----|-------------------------------|
| 0 | Positive |
| 1 | Negative |

- (4) The carry flag will not turn ON even if an overflow results.

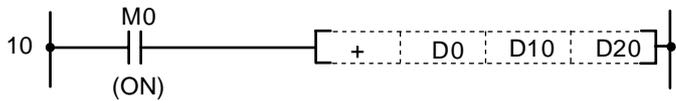
Execution conditions

The execution conditions for + are as shown below.



Program example

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

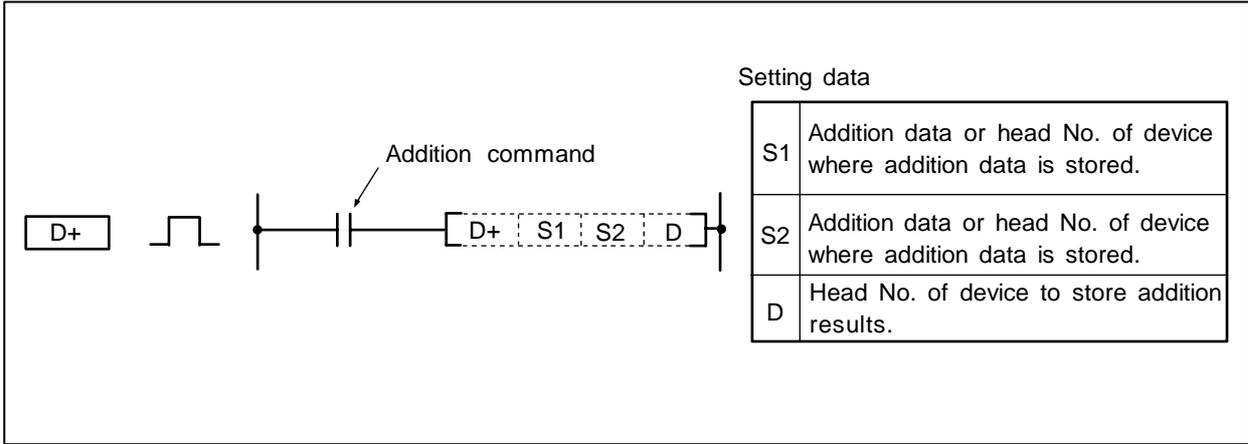


Coding

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

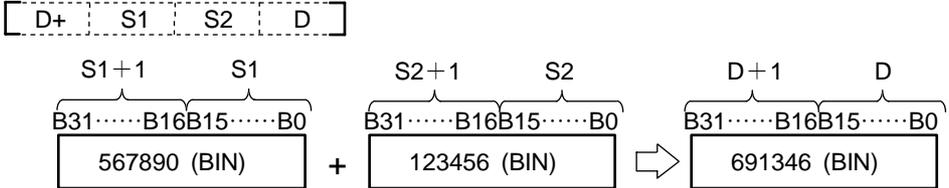
○ D+ ... BIN 32-bit addition

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|-----|--|--|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | | | |
| S1 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | 4/5 | | |
| S2 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | ○ | ○ | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | | |



Function

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



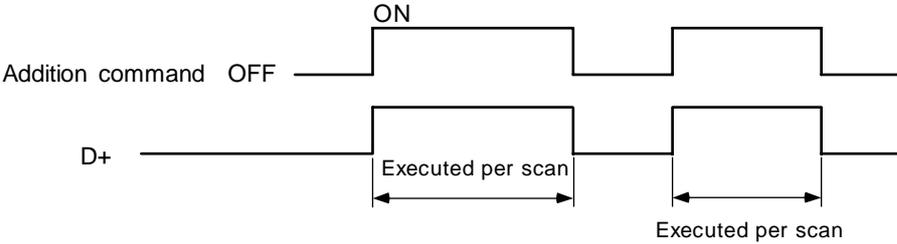
- (2) -2147483648 to 2147483647 (BIN 32-bit) can be designated in S1 and S2.
 (3) The positive/negative of the data in S1, S2 and D is determined with the highest-order bit (B31).

| B31 | Judgment of positive/negative |
|-----|-------------------------------|
| 0 | Positive |
| 1 | Negative |

- (4) The carry flag will not turn ON even if an overflow results.

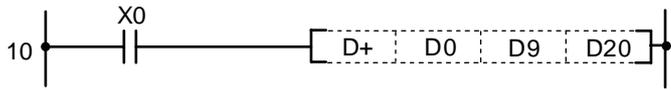
Execution conditions

The execution conditions for D+ 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.

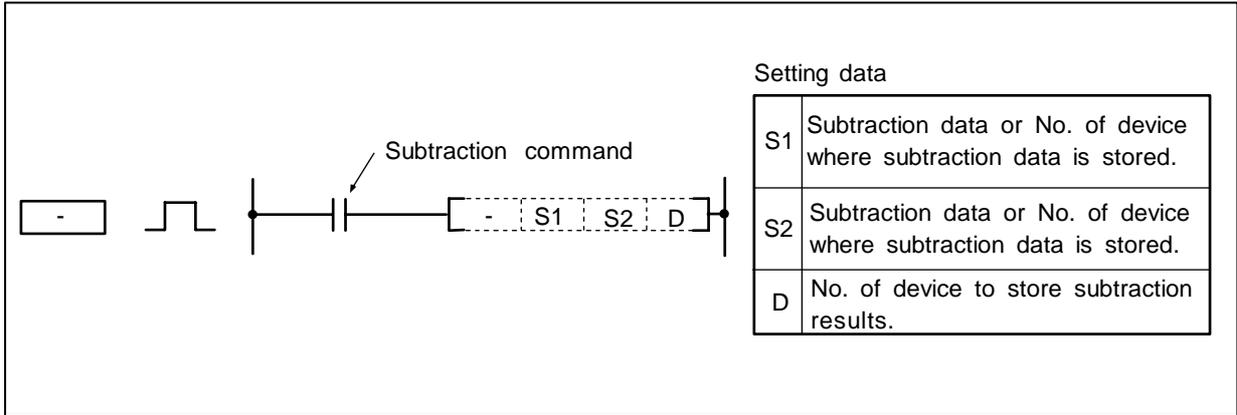


Coding

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

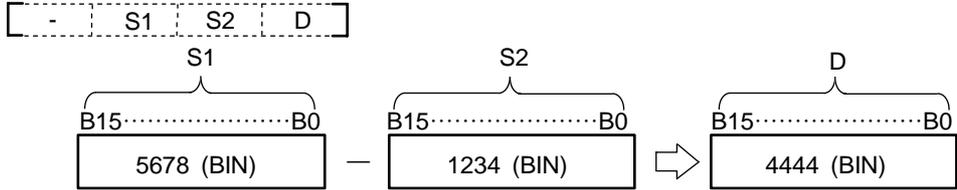
○ - ... BIN 16-bit subtraction

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | |
|-----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N |
| S1 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |
| S2 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | ○ | ○ | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |



Function

- (1) The device designated with S2 is subtracted From the device designated with S1, and the subtraction results are stored in the device designated with D.



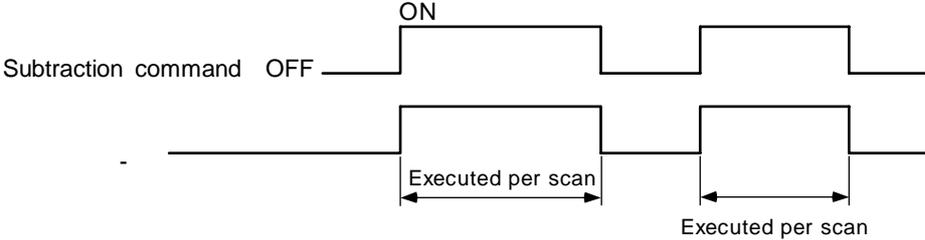
- (2) -32768 to 32767 (BIN 16-bit) can be designated in S1 and S2.
 (3) The positive/negative of the data in S1, S2 and D is determined with the highest-order bit (B15).

| B15 | Judgment of positive/negative |
|-----|-------------------------------|
| 0 | Positive |
| 1 | Negative |

- (4) The carry flag will not turn ON even if an underflow results.

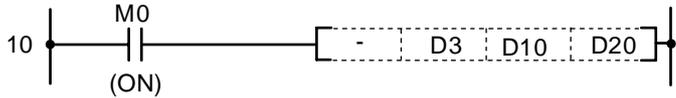
Execution conditions

The execution conditions for - are as shown below.



Program example

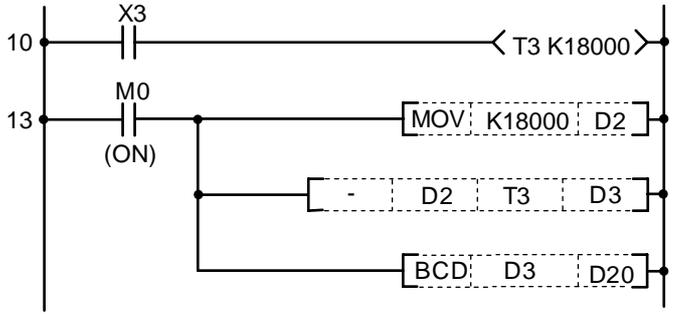
(1) Program to subtract the BIN data D10 from D3 and output to D20.



Coding

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

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

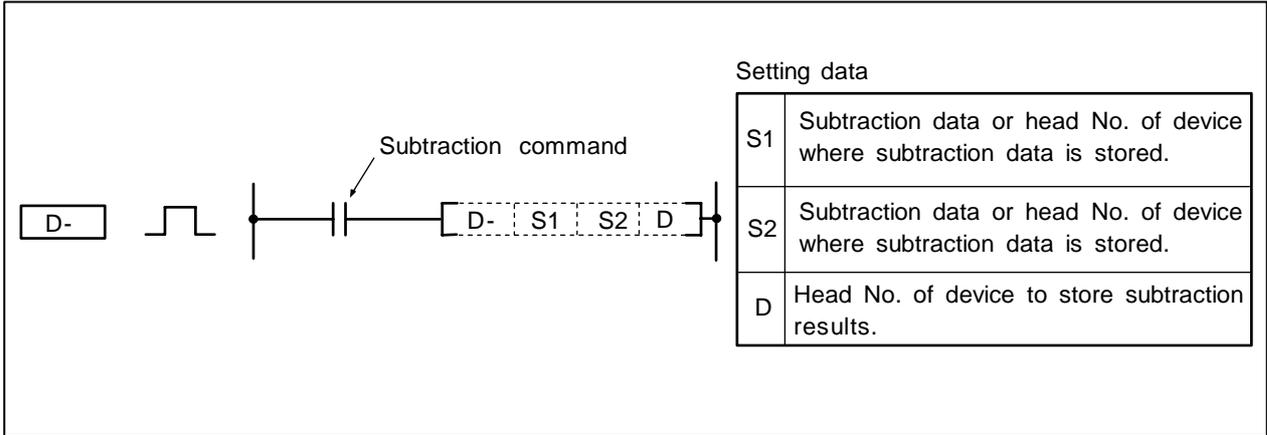


Coding

| 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 | | | | |

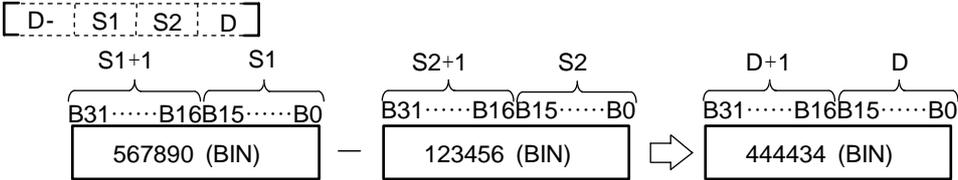
○ D- ... BIN 32-bit subtraction

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N |
| S1 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |
| S2 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | ○ | ○ | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |



Function

- (1) The device designated with S2 is subtracted from the device designated with S1, and the subtraction results are stored in the device designated with D.



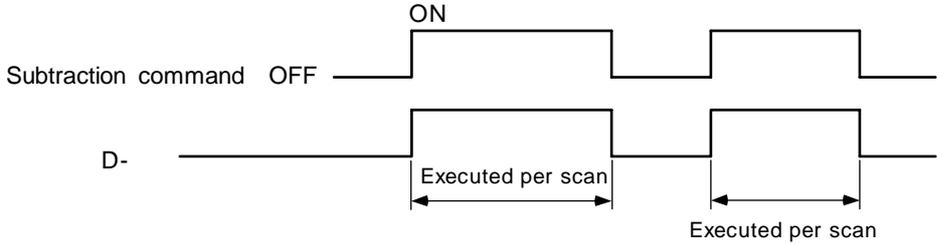
- (2) -2147483648 to 2147483647 (BIN 32-bit) can be designated in S1 and S2.
- (3) The positive/negative of the data in S1, S2 and D is determined with the highest-order bit (B31).

| B31 | Judgment of positive/negative |
|-----|-------------------------------|
| 0 | Positive |
| 1 | Negative |

- (4) The carry flag will not turn ON even if an underflow results.

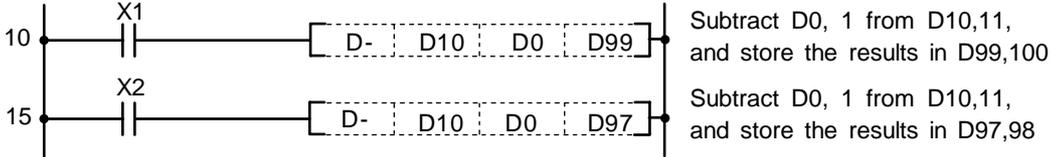
Execution conditions

The execution conditions for D- 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.

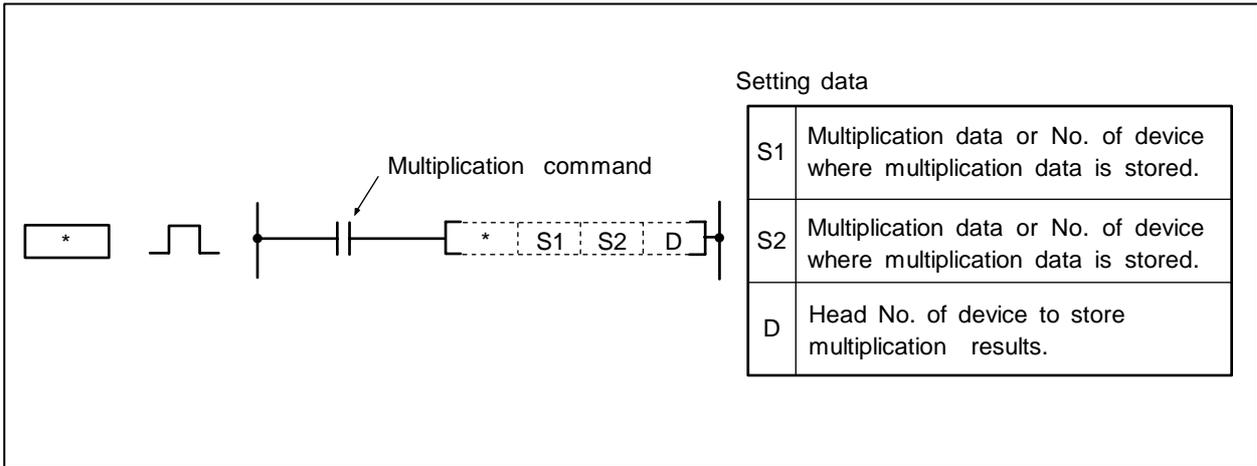


Coding

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

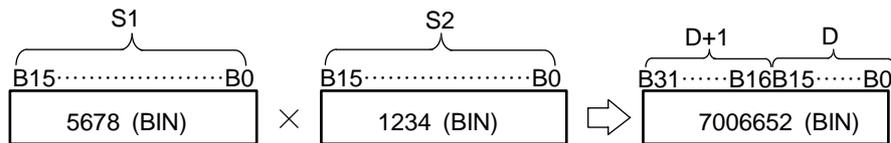
○ * ... BIN 16-bit multiplication

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N |
| S1 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |
| S2 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | ○ | ○ | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |



Function

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

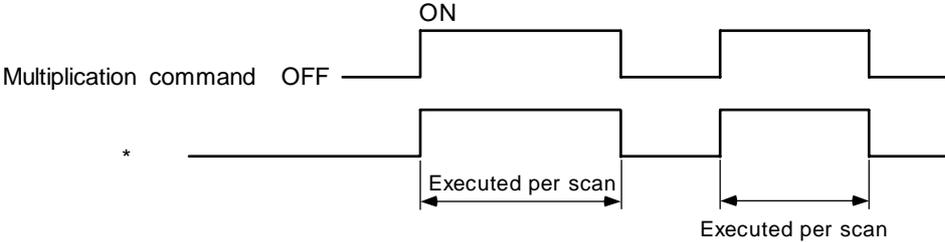


- (2) -32768 to 32767 (BIN 16-bit) can be designated in S1 and S2.
 (3) The positive/negative of the data in S1, S2 and D is determined with the highest-order bit (S1 and S2 is by B15, D is by B31).

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

Execution conditions

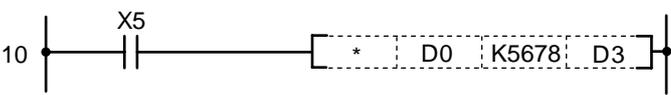
The execution conditions for * are as shown below.



Program example

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

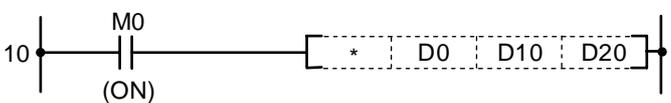
Coding



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

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

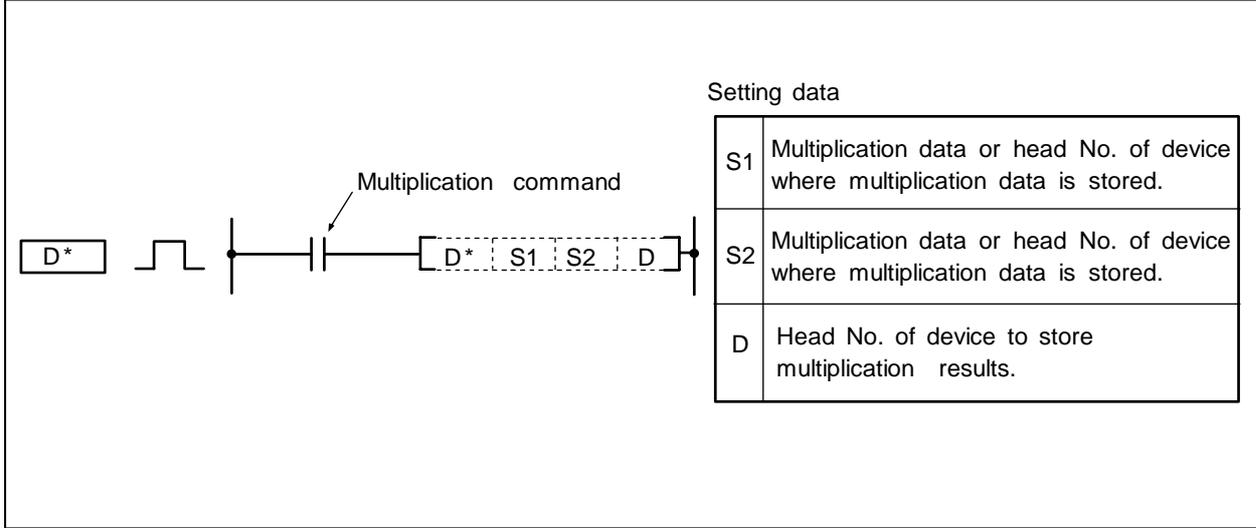
Coding



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

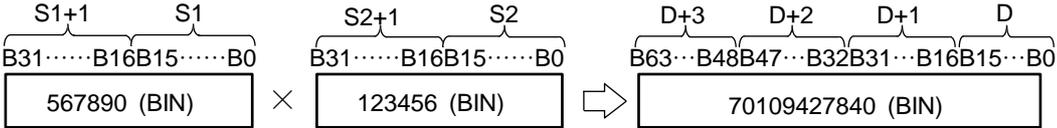
○ D* ... BIN 32-bit multiplication

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N |
| S1 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |
| S2 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | ○ | ○ | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |



Function

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

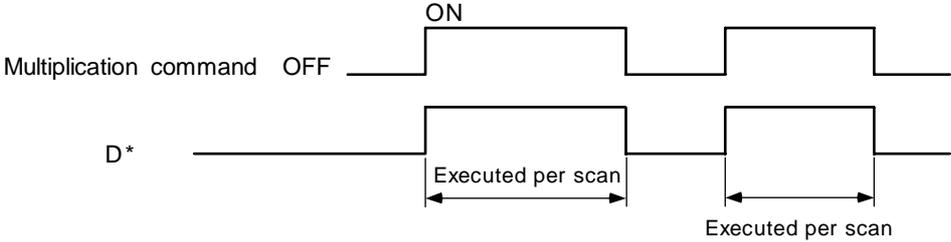


- (2) -2147483648 to 2147483647 (BIN 32-bit) can be designated in S1 and S2.
 (3) The positive/negative of the data in S1, S2 and D is determined with the highest-order bit (S1 and S2 is by B31, D is by B63).

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

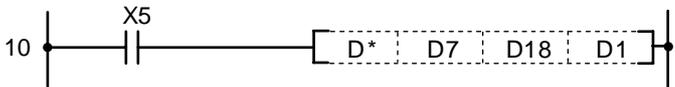
Execution conditions

The execution conditions for D* are as shown below.



Program example

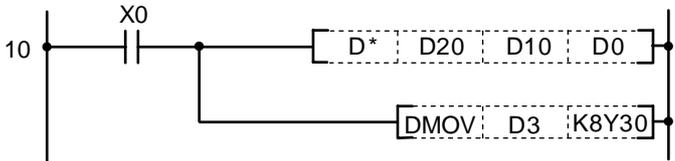
- (1) Program to multiply the D7, 8 BIN data and D18, 19 BIN data when X5 turns ON, and output the results to D1 to 4.



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.

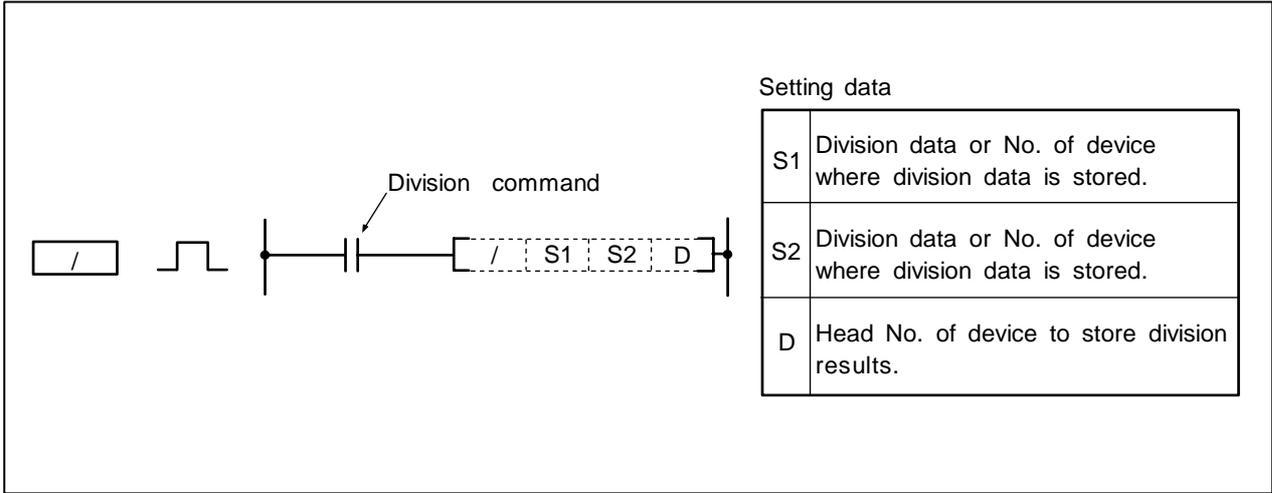


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-------|----|
| 10 | LD | X0 | | |
| 11 | D* | D20 | D10 | D0 |
| 16 | DMOV | D3 | K8Y30 | |
| 20 | | | | |

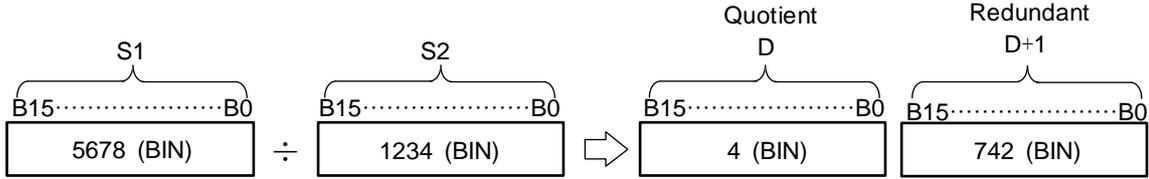
○ / ... BIN 16-bit division

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | |
|-----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---|---------|-------|-------------------|--------------|-------|---|
| | Bit device | | | | | | Word device | | | | | | Constant | | Pointer | Level | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N |
| S1 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |
| S2 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | ○ | ○ | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |



Function

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



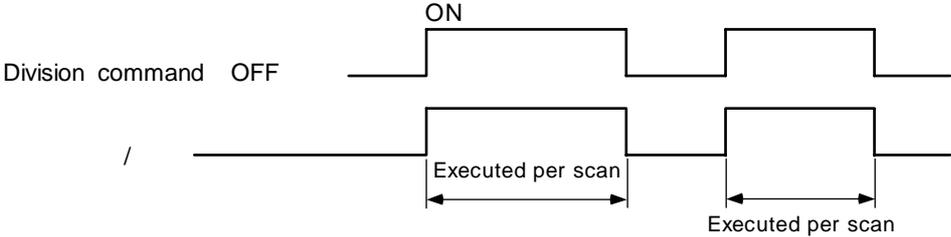
- (2) -32768 to 32767 (BIN 16-bit) can be designated in S1 and S2.
 (3) The positive/negative of the data in S1, S2 and D is determined with the highest-order bit (B15).

| 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

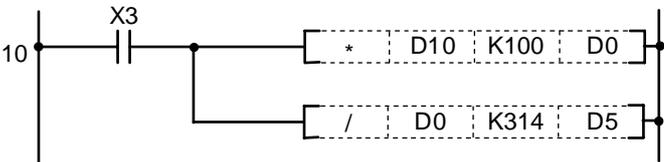
The execution conditions for / are as shown below.



Program example

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

Coding



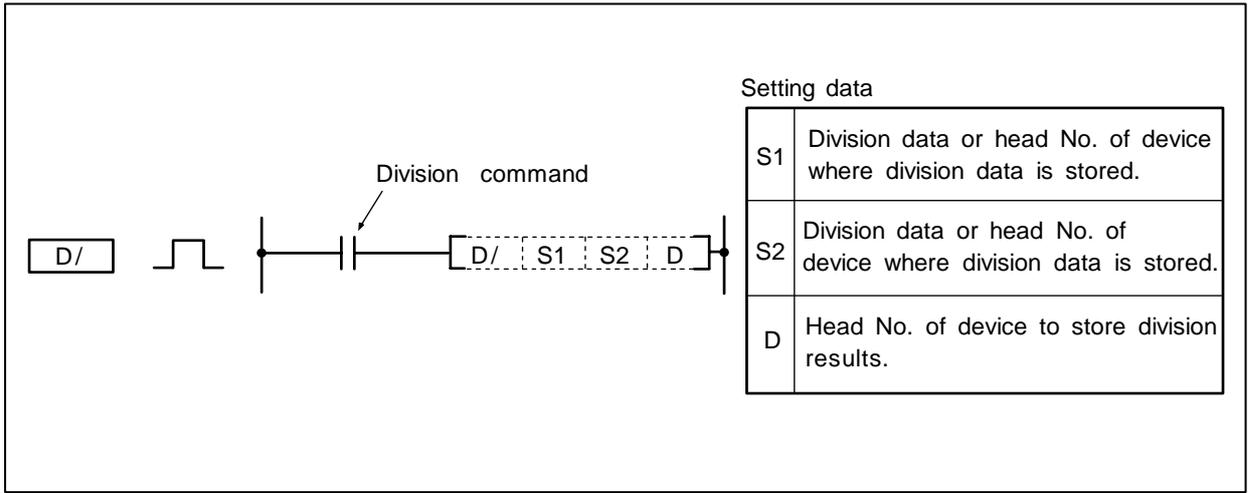
| No. of steps | Com- mand | Device | | |
|--------------|--------------|--------|------|----|
| 10 | LD | X3 | | |
| 11 | * | D10 | K100 | D0 |
| 15 | / | D0 | K314 | D5 |
| 20 | | | | |

Point

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

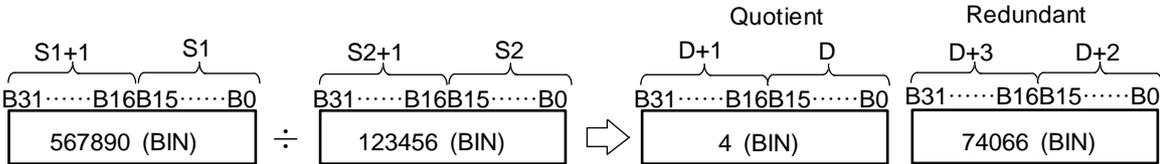
○ D/ ... BIN 32-bit division

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | |
|-----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|--|-----|--|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | | | |
| S1 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | 5/6 | |
| S2 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | ○ | ○ | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | | |



Function

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



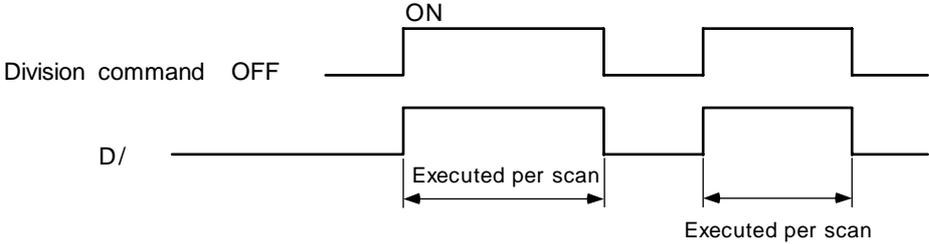
- (2) -2147483648 to 2147483647 (BIN 32-bit) can be designated in S1 and S2.
- (3) The positive/negative of the data in S1, S2 and D is determined with the highest-order bit (B31).

| 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

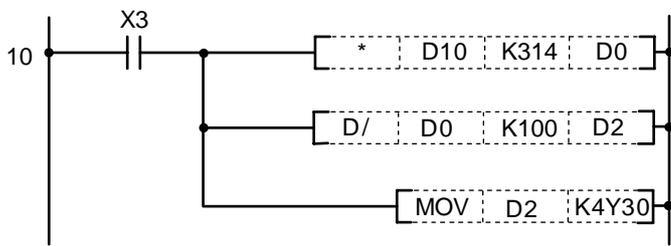
The execution conditions for D/ are as shown below.



Program example

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

Coding



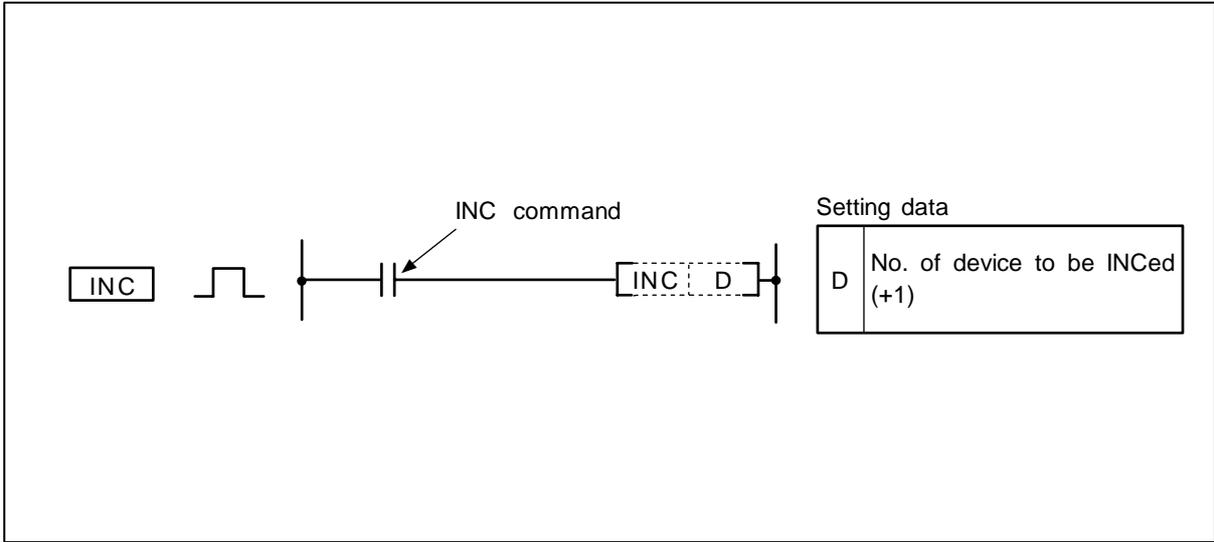
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-------|----|
| 10 | LD | X3 | | |
| 11 | * | D10 | K314 | D0 |
| 15 | D/ | D0 | K100 | D2 |
| 21 | MOV | D2 | K4Y30 | |
| 24 | | | | |

Point

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

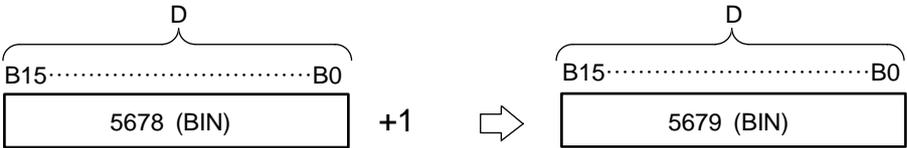
○ INC ... (16-bit BIN data) +1

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---------|-------------------|--------------|-------|-------|---|---|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | Pointer | | | | Level | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | K | H | P | N |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | 2 | |



Function

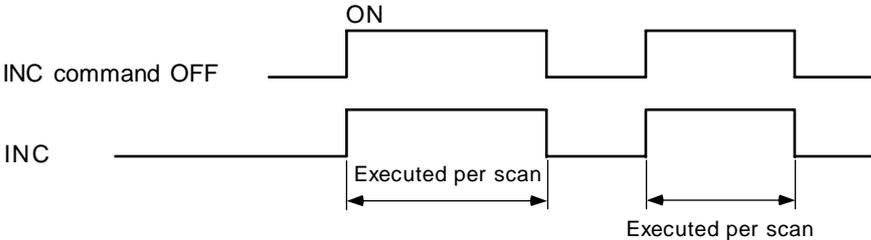
(1) The device (16-bit data) designated with D is incremented by one.



(2) If INC is executed when the details of the device designated with D are 32767, -32768 will be stored in the device designated with D.

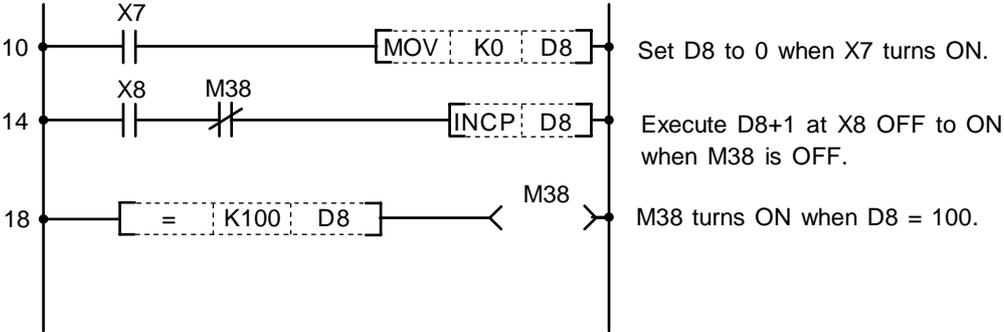
Execution conditions

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



Program example

(1) Example of addition counter program

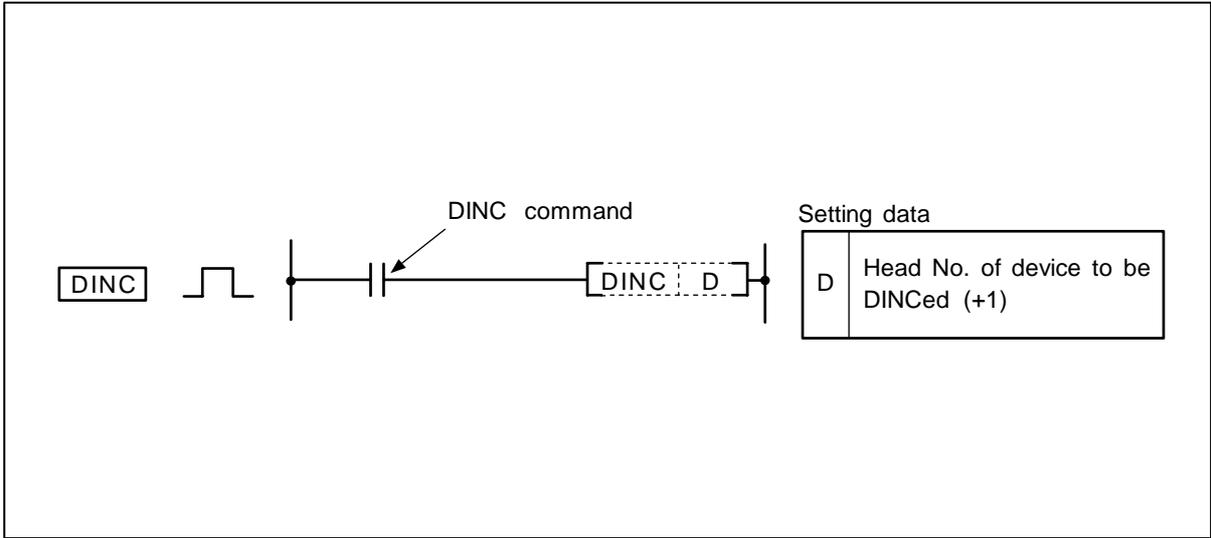


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| | | | | |
| 10 | LD | X7 | | |
| 11 | MOV | K0 | D8 | |
| 14 | LD | X8 | | |
| 15 | PLS | M5 | | |
| 17 | LD | M5 | | |
| 18 | ANI | M38 | | |
| 19 | INC | D8 | | |
| 21 | LD= | K100 | D8 | |
| 24 | OUT | M38 | | |
| 25 | | | | |

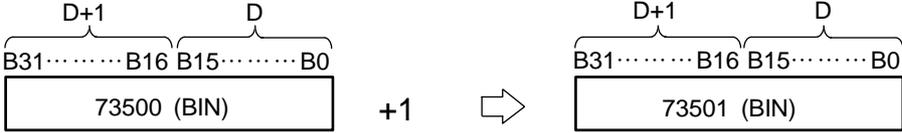
○ DINC ... (32-bit BIN data) +1

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---------|-------------------|--------------|-------|-------|---|---|---|
| | Bit device | | | | | | Word device | | | | | | Constant | Pointer | | | | Level | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | K | H | P | N |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | 2 | |



Function

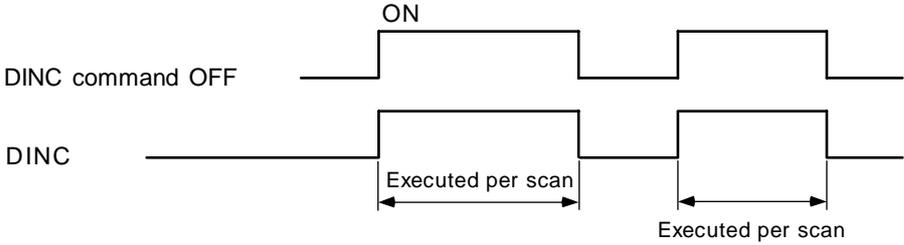
(1) The device (32-bit data) designated with D is incremented by one.



(2) If DINC is executed when the details of the device designated with D are 2147483647, -2147483648 will be stored in the device designated with D.

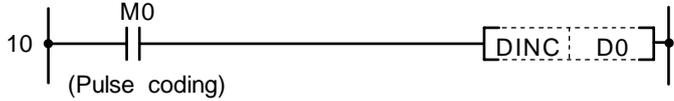
Execution conditions

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



Program example

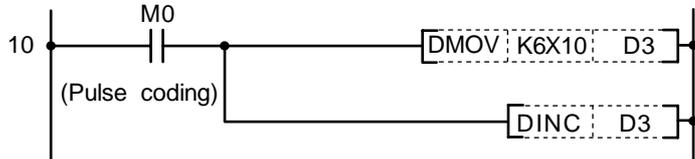
(1) Program to increment the D0, 1 data by one when M0 turns ON.



Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|--|--|
| 10 | LD | M0 | | |
| 11 | DINC | D0 | | |
| 13 | | | | |

(2) Program to increment X10 to 27 data by one when M0 turns ON, and to store the results in D3, 4.

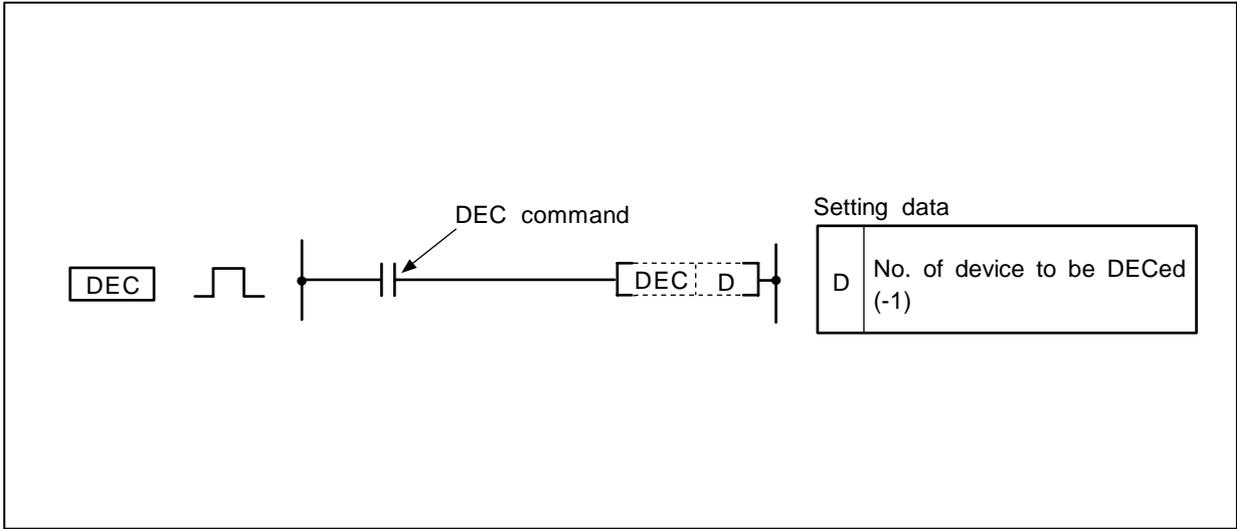


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M0 | | |
| 11 | DMOV | K6X10 | D3 | |
| 14 | DINC | D3 | | |
| 16 | | | | |

○ DEC ... (16-bit BIN data) -1

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---------|-------------------|--------------|-------|-------|---|---|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | Pointer | | | | Level | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | K | H | P | N |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | 2 | |



Function

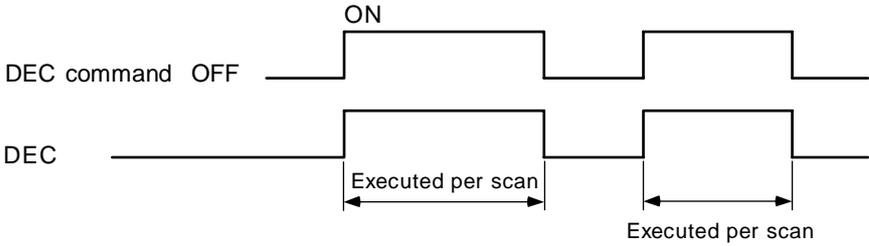
(1) The device (16-bit data) designated with D is decremented by one.



(2) If DEC is executed when the details of the device designated with D are 0, -1 will be stored in the device designated with D.

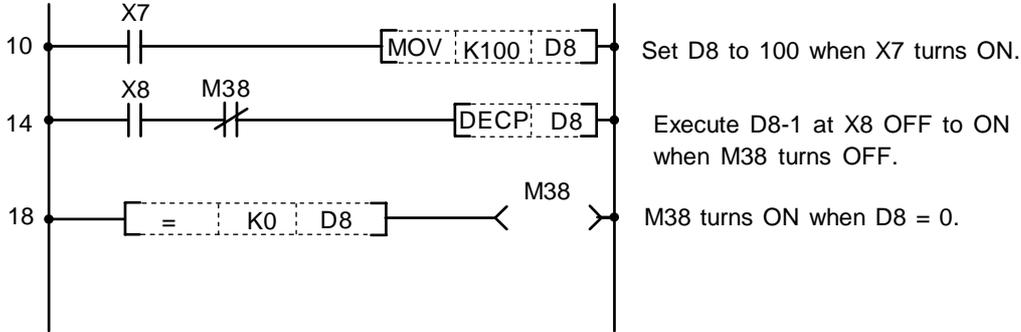
Execution conditions

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



Program example

(1) Example of subtraction counter program

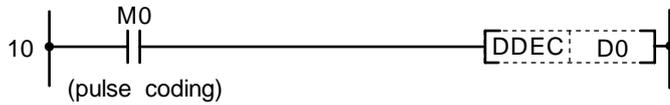


Coding

| No. of steps | Command | Device | | |
|--------------|---------|--------|----|--|
| 10 | LD | X7 | | |
| 11 | MOV | K100 | D8 | |
| 14 | LD | X8 | | |
| 15 | PLS | M5 | | |
| 17 | LD | M5 | | |
| 18 | ANI | M38 | | |
| 19 | DEC | D8 | | |
| 21 | LD= | K0 | D8 | |
| 24 | OUT | M38 | | |
| 25 | | | | |

Program example

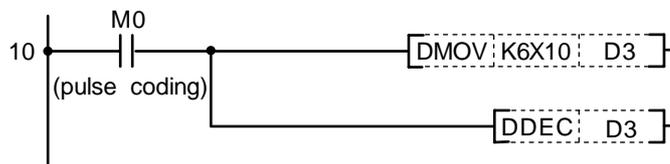
- (1) Program to decrement the D0, 1 data by one when M0 turns ON.



Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|--|--|
| 10 | LD | M0 | | |
| 11 | DDEC | D0 | | |
| 13 | | | | |

- (2) Program to decrement X10 to 27 data by one when M0 turns ON, and to store the results in D3, 4.

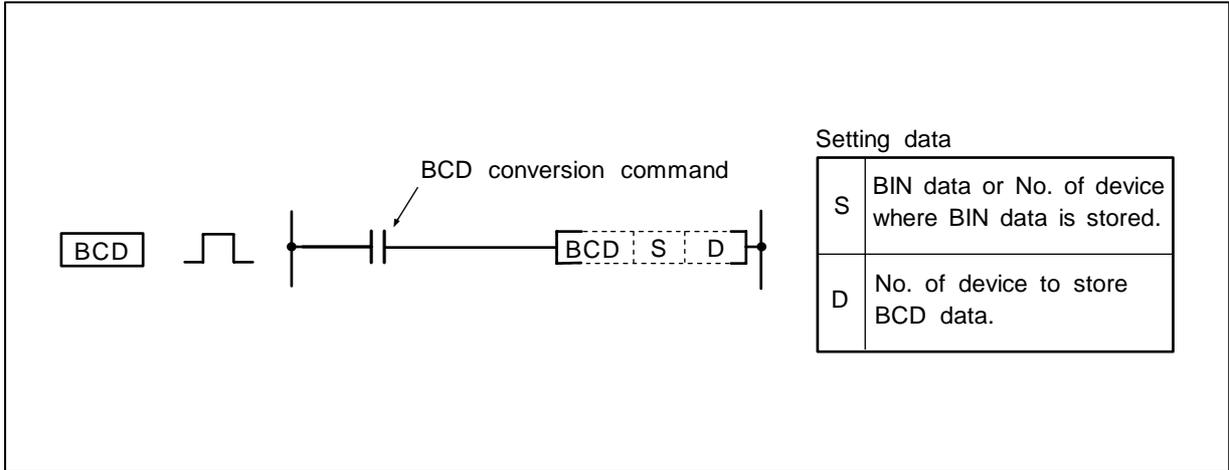


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M0 | | |
| 11 | DMOV | K6X10 | D3 | |
| 14 | DDEC | D3 | | |
| 16 | | | | |

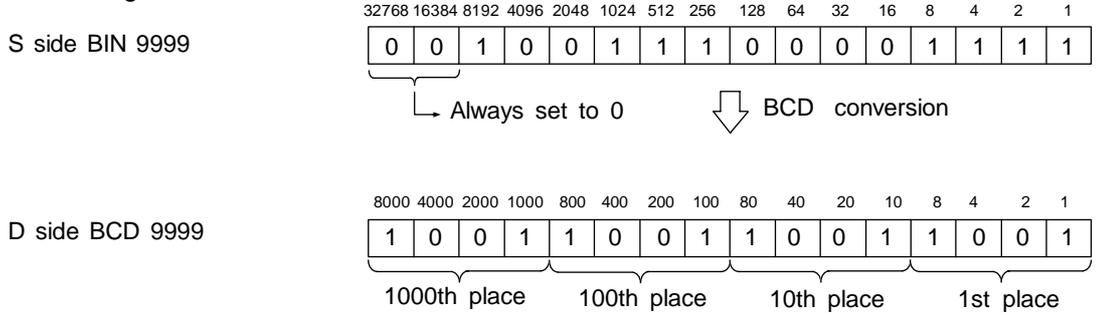
○ BCD ... BIN → BCD conversion (16-bit)

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|--|-------------------|--------------|-------|---------|-------|
| | Bit device | | | | | | Word device | | | | | | Constant | | | | | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | K | H |
| S | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |



Function

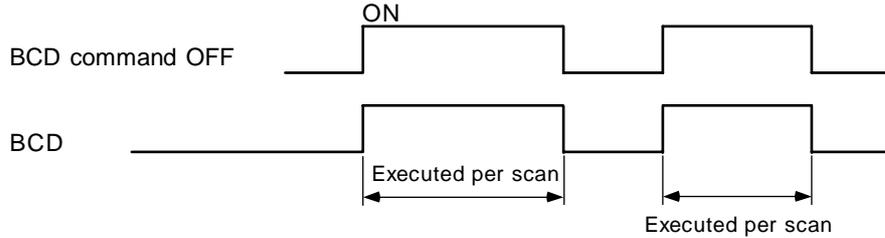
The BIN data (0 to 9999) of the device designated with S is BCD converted and transmitted to the device designated with D.



(Note 1) A minus value cannot be converted correctly.

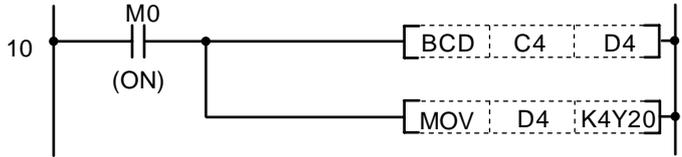
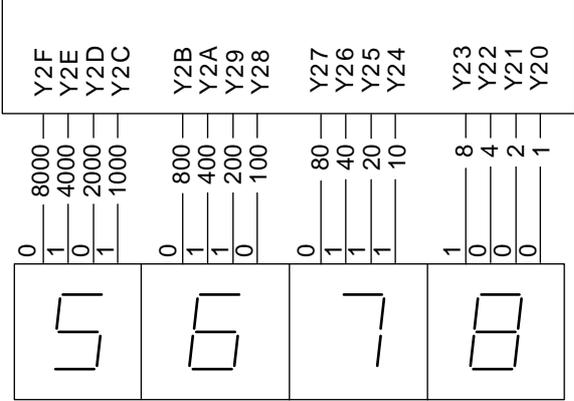
Execution conditions

The execution conditions for BCD are as follow.



Program example

(1) Program to output C4 current value from Y20 to 2F to BCD display.

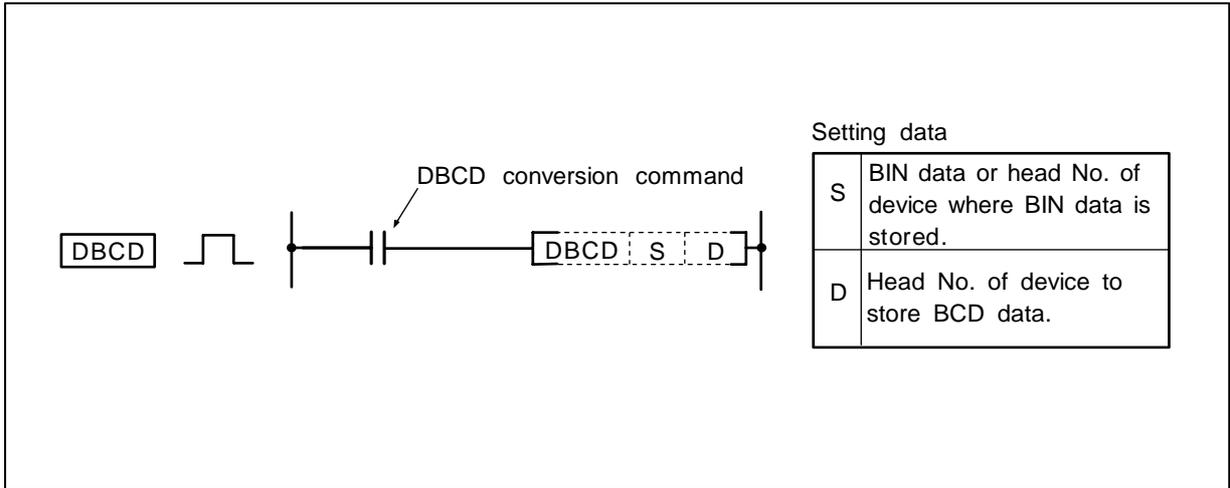


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-------|--|
| 10 | LD | M0 | | |
| 11 | BCD | C4 | D4 | |
| 14 | MOV | D4 | K4Y20 | |
| 17 | | | | |

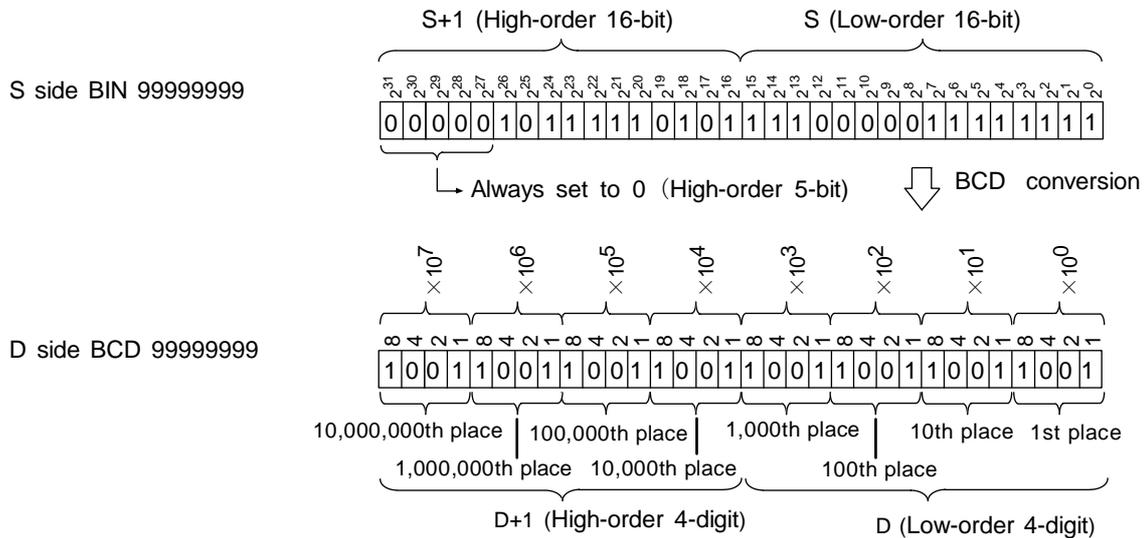
○ DBCD ... BIN → BCD conversion (32-bit)

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---|---------|-------|-------------------|--------------|-------|---|--|
| | Bit device | | | | | | Word device | | | | | | Constant | | Pointer | Level | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | |
| S | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | ○ | 4 | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | |



Function

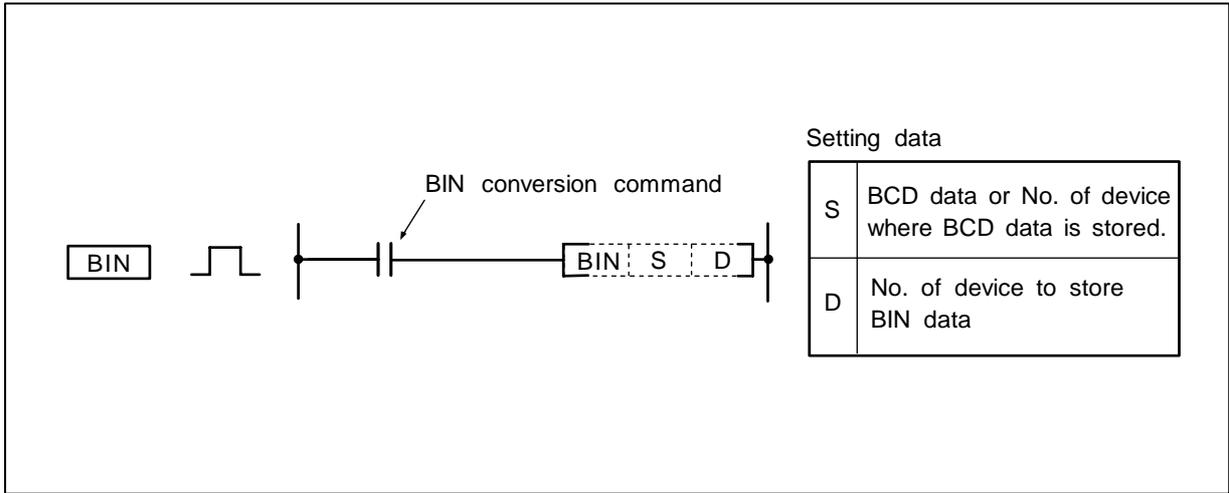
The BIN data (0 to 99999999) of the device designated with S is BCD converted and transmitted to the device designated with D.



(Note 1) A minus value cannot be converted correctly.

○ BIN ... BCD → BIN conversion (16-bit)

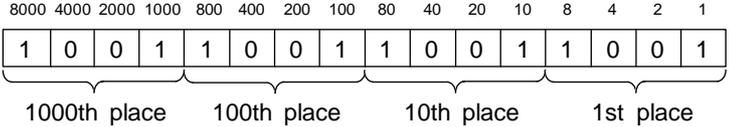
| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|--|-------------------|--------------|-------|---------|-------|
| | Bit device | | | | | | Word device | | | | | | Constant | | | | | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | K | H |
| S | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |



Function

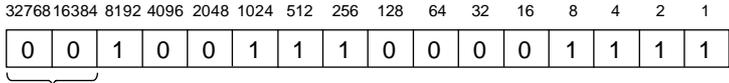
The BCD data (0 to 9999) of the device designated with S is BIN converted and transmitted to the device designated with D.

S side BCD 9999



↓ BIN conversion

D side BIN 9999

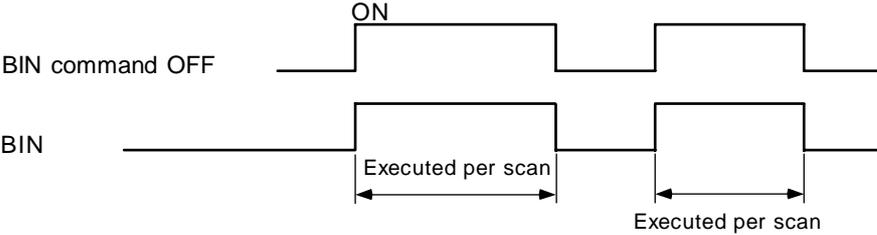


↳ Set to 0

(Note 1) A minus value cannot be converted correctly.

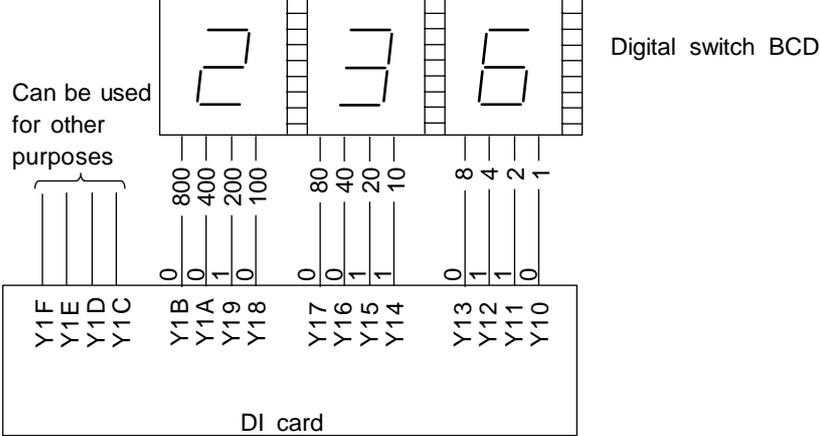
Execution conditions

The execution conditions for BIN are as follow.



Program example

(1) Program to BIN convert the X10 to 1B BCD data when X8 turns On, and store in D8.

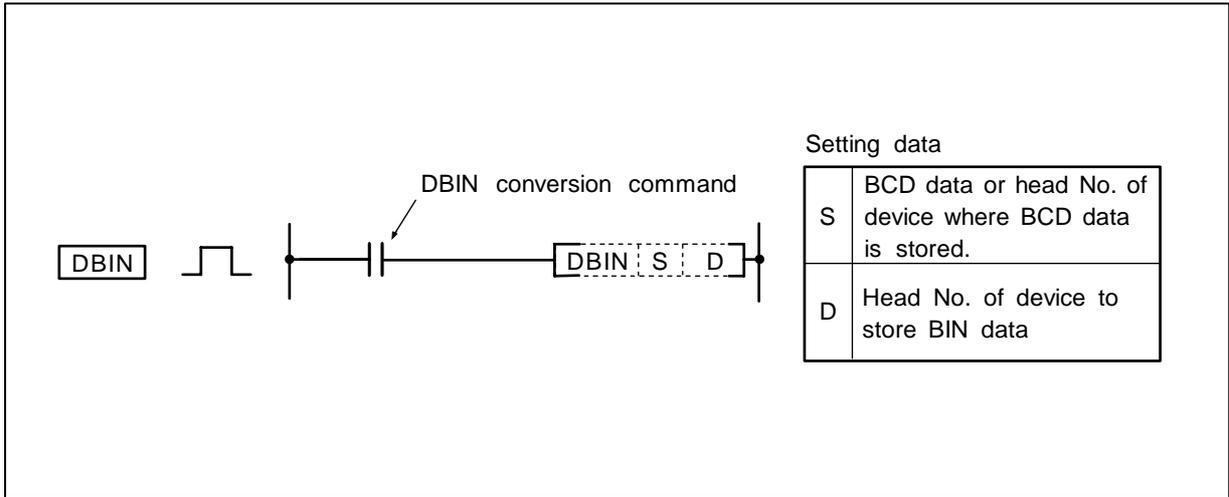


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| | | | | |
| 10 | LD | X8 | | |
| 11 | BIN | K3X10 | D8 | |
| 14 | | | | |

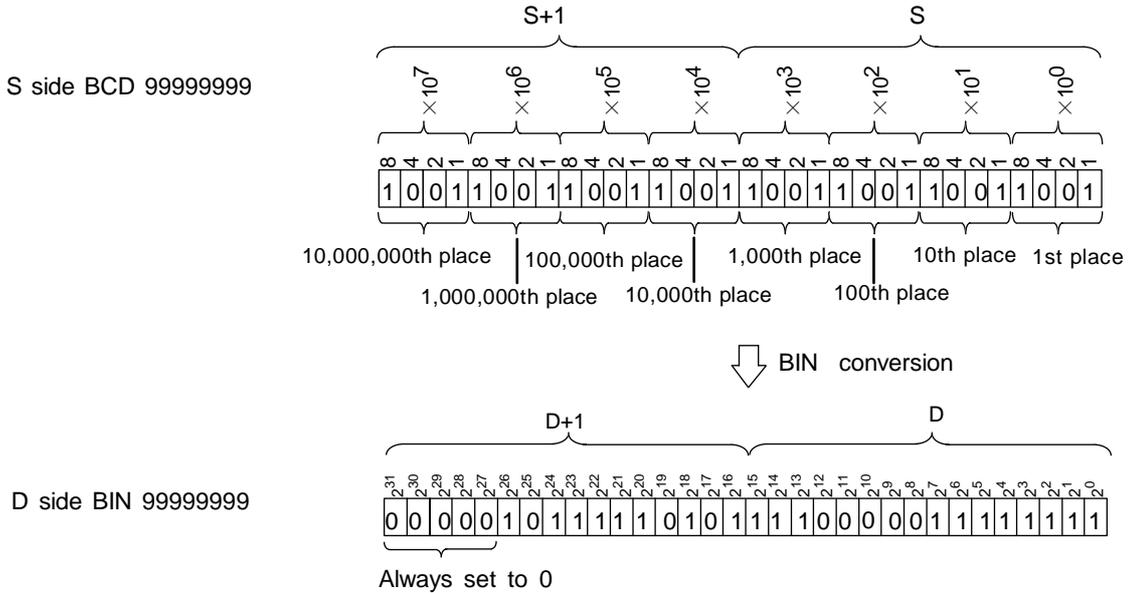
○ DBIN ... BCD → BIN conversion (32-bit)

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---|---------|---|-------------------|--------------|-------|-------|--|
| | Bit device | | | | | | Word device | | | | | | Constant | | Pointer | | | | | Level | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | |
| S | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | |



Function

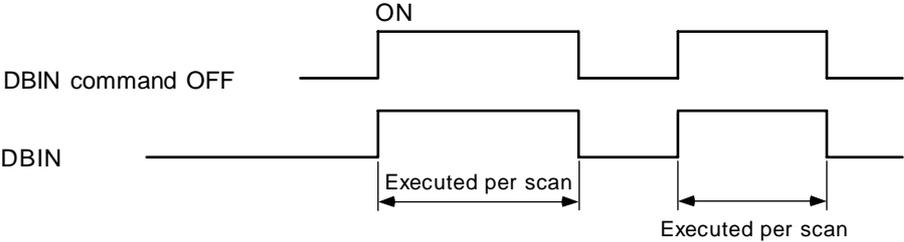
The BCD data (0 to 99999999) of the device designated with S is BIN converted and transmitted to the device designated with D.



(Note 1) A minus value cannot be converted correctly.

Execution conditions

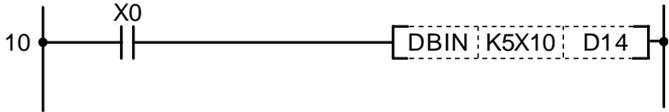
The execution conditions for DBIN are as follow.



Program example

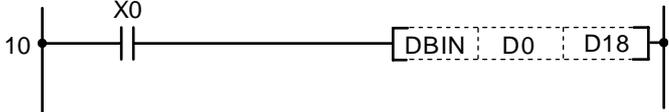
(1) Program to BIN convert the X10 to 23 BCD data when X0 turns ON, and to store in D14, 15.

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.



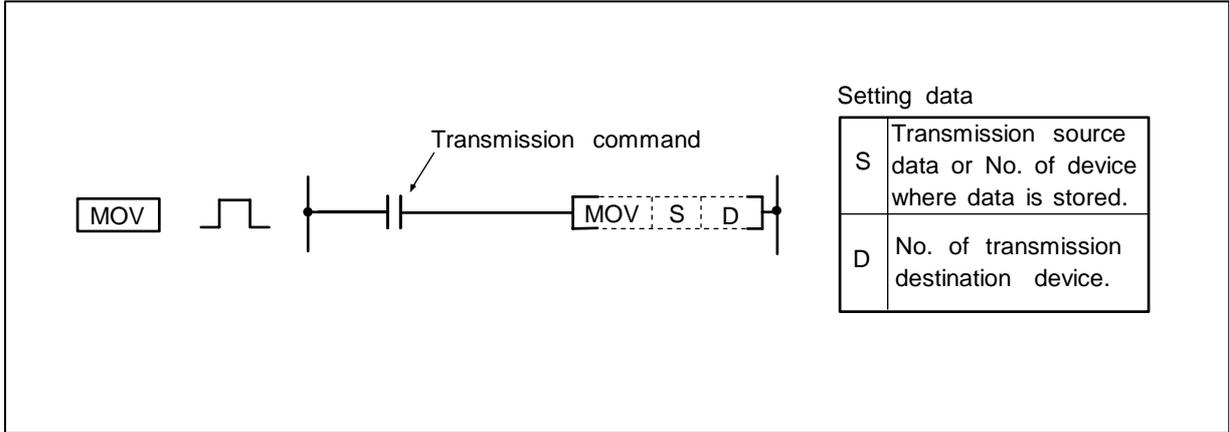
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-----|--|
| 10 | LD | X0 | | |
| 11 | DBIN | D0 | D18 | |
| 15 | | | | |

○ MOV ... 16-bit data transmission

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|---|----------|---|---------|-------|-------------------|--------------|-------|---|---|
| | Bit device | | | | | | Word device | | | | | | Constant | | Pointer | Level | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | |
| S | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | △ | | | ○ | ○ | | | ○ | 3 | ○ |
| D | *1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | △ | | | | | | | | | |

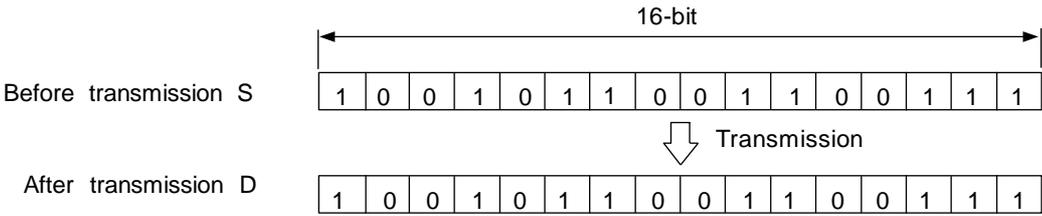
△ : MOV from a bit device (word device) to Z is not possible. (MOV from a constant to Z is possible.)
 Z cannot be independently placed on the source side, but can be used on the source side as ornaments for D and R.
 Refer to "Index Ornament" for details.

*1 MOV to device X can be programmed, but this is a command for testing by Mitsubishi. Do not use it.



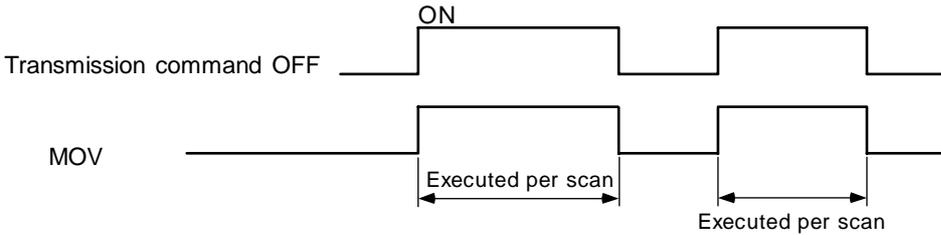
Function

The 16-bit data of the device designated with S is transmitted to the device designated with D.



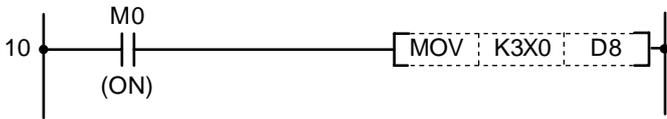
Execution conditions

The execution conditions for MOV are as shown below.



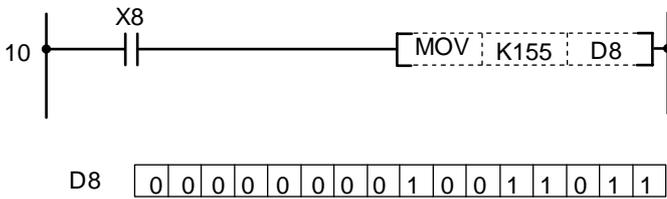
Program example

(1) Program to store input X0 to B data in D8.



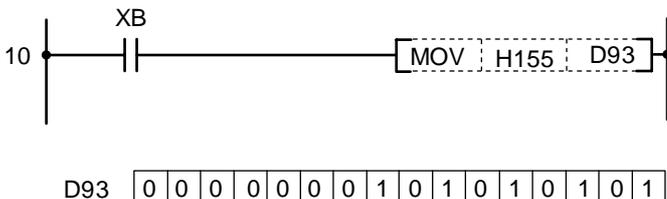
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M0 | | |
| 11 | MOV | K3X0 | D8 | |
| 14 | | | | |

(2) Program to store 155 in D8 as binary value when X8 turns ON.



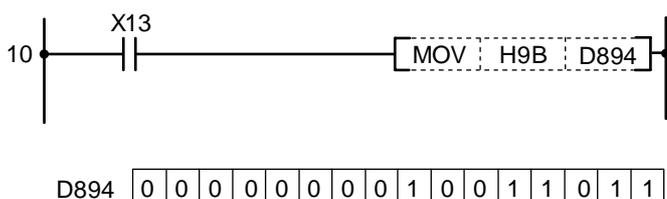
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | X8 | | |
| 11 | MOV | K155 | D8 | |
| 14 | | | | |

(3) Program to store 155 in D93 as BCD value in when XB turns ON.



| 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.

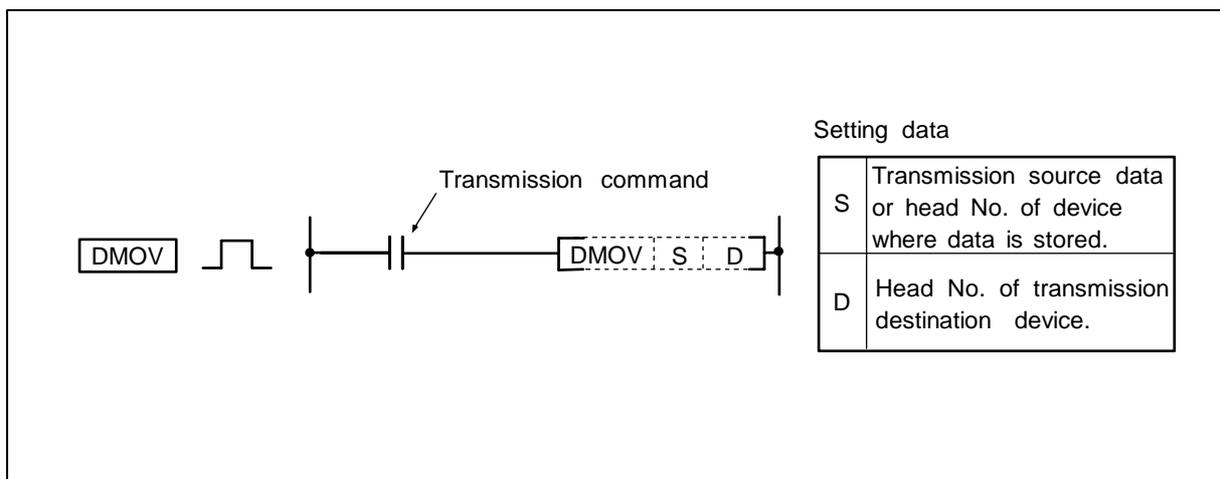


| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|------|--|
| 10 | LD | X13 | | |
| 11 | MOV | H9B | D894 | |
| 14 | | | | |

○ DMOV ... 32-bit data transmission

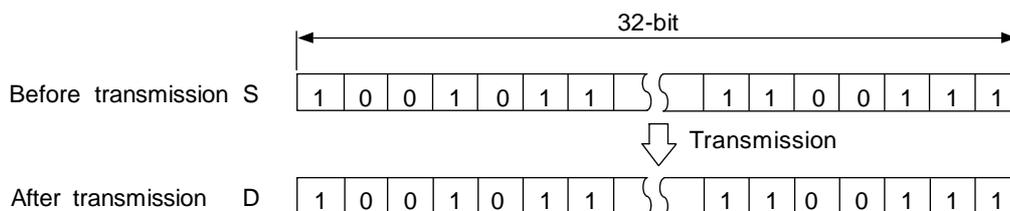
| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | |
| S | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | | | | | |
| D | *2 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | |

- *1 DMOV from a bit device to a bit device is not possible.
- *2 DMOV to device X can be programmed, but this is a command for testing by Mitsubishi. Do not use it.



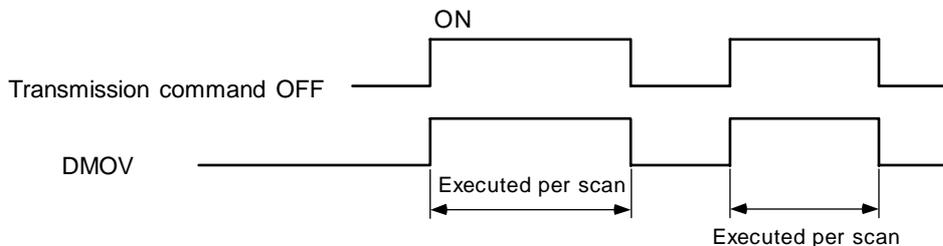
Function

The 32-bit data of the device designated with S is transmitted to the device designated with D.



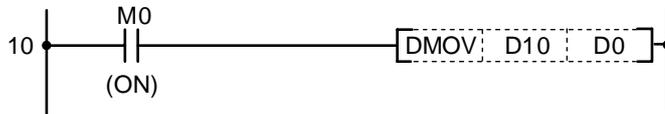
Execution conditions

The execution conditions for DMOV are as shown below.



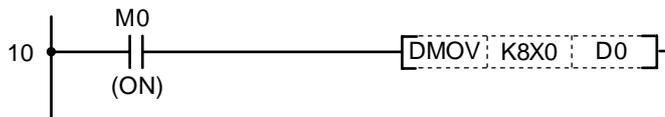
Program example

(1) Program to store D10, D11 data in D0, D1.



| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M0 | | |
| 11 | DMOV | D10 | D0 | |
| 14 | | | | |

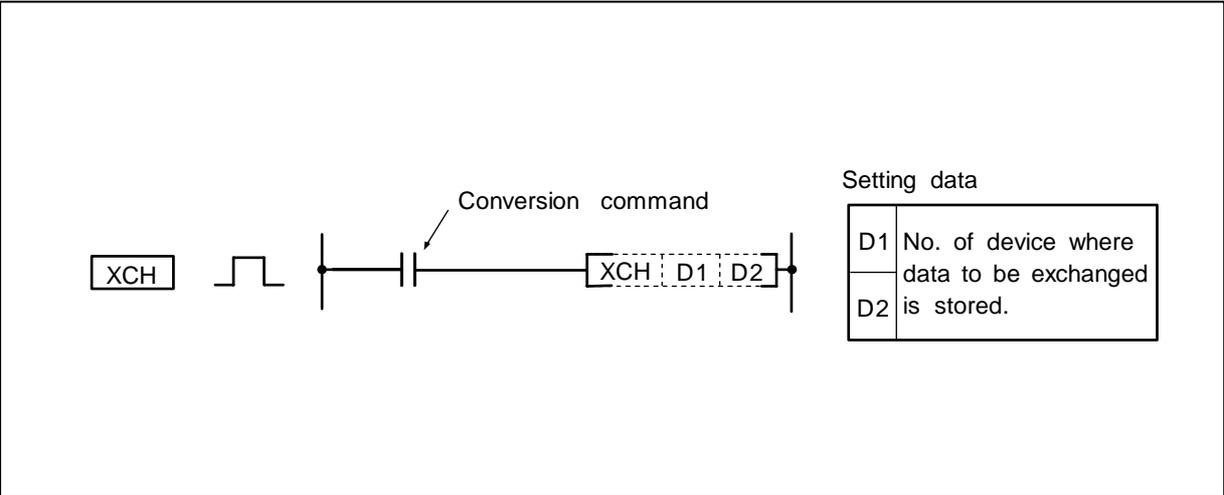
(2) Program to store X0 to 1F data in D0, D1.



| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M0 | | |
| 11 | DMOV | K8X0 | D0 | |
| 15 | | | | |

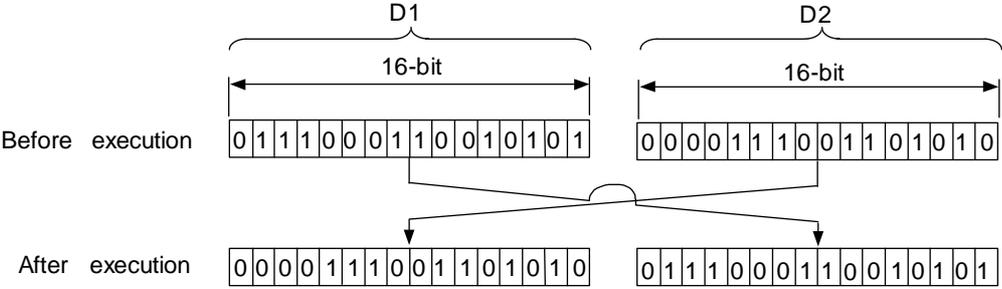
○ XCH ... 16-bit data exchange

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---------|-------------------|--------------|-------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | Pointer | | | | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | K |
| D1 | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | |
| D2 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | |



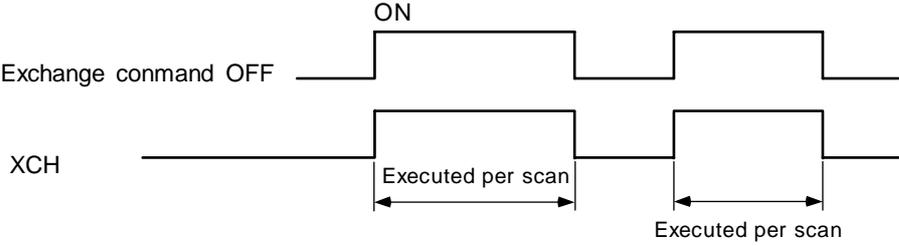
Function

The D1 and D2 16-bit data are exchanged.



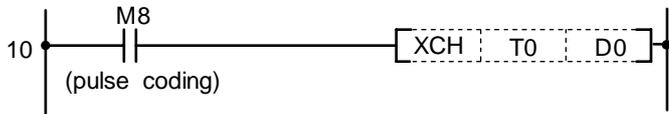
Execution conditions

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



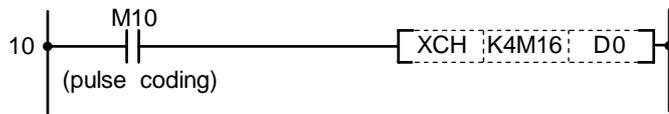
Program example

(1) Program to exchange T0 current value with D0 details when M8 turns ON.



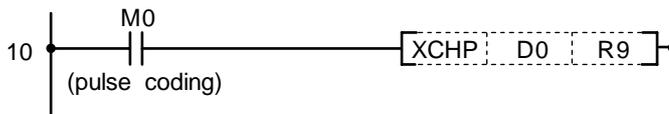
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M8 | | |
| 11 | XCH | T0 | D0 | |
| 15 | | | | |

(2) Program to exchange D0 details with M16 to M31 data when M10 turns ON.



| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M10 | | |
| 11 | XCH | K4M16 | D0 | |
| 15 | | | | |

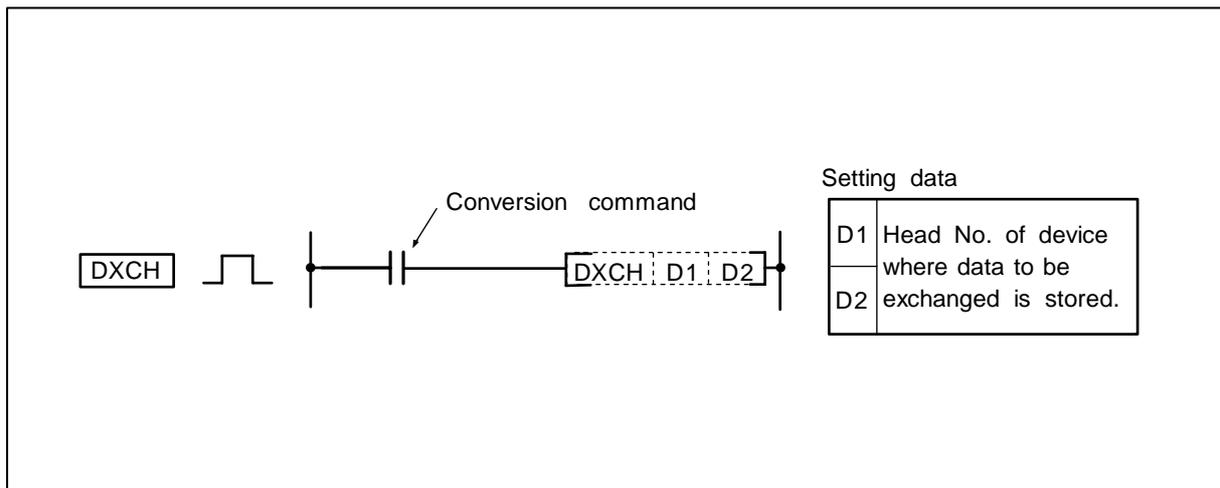
(3) Program to exchange D0 details with R9 details when M0 turns ON.



| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M0 | | |
| 11 | XCH | D0 | R9 | |
| 15 | | | | |

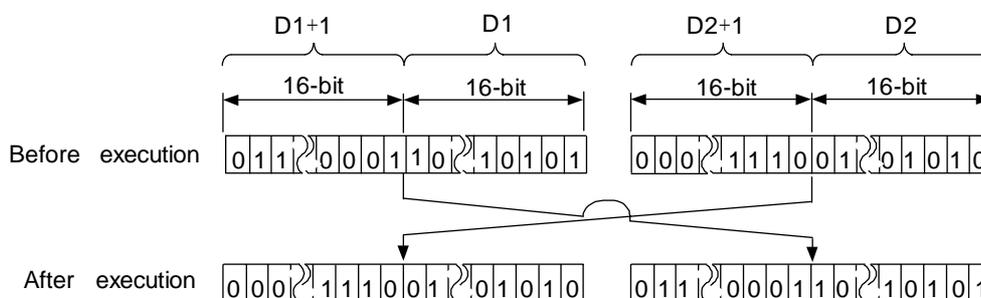
○ DXCH ... 32-bit data exchange

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|--|-------------------|--------------|-------|---------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | | | | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | | K |
| D1 | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | |
| D2 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |



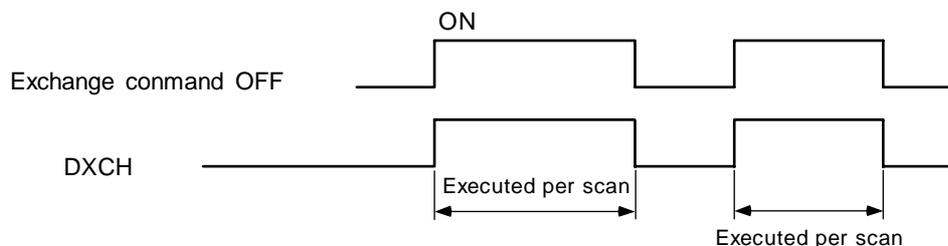
Function

The D1 and D2 32-bit data are exchanged.



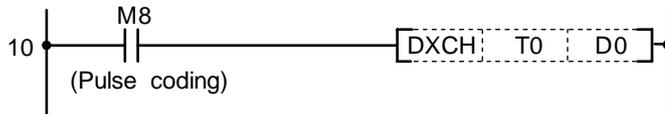
Execution conditions

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



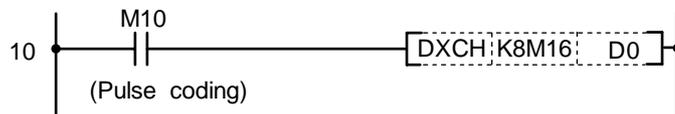
Program example

- (1) Program to exchange T0 and T1 current values with D0, 1 details when M8 turns ON.

**Coding**

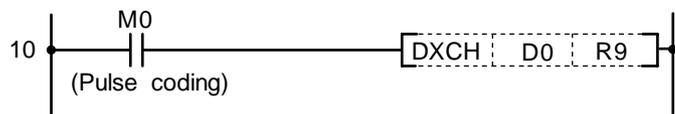
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M8 | | |
| 11 | DXCH | T0 | D0 | |
| 15 | | | | |

- (2) Program to exchange D0, 1 details with M16 to M47 data when M10 turns ON.

**Coding**

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | X10 | | |
| 11 | DXCH | K8M16 | D0 | |
| 15 | | | | |

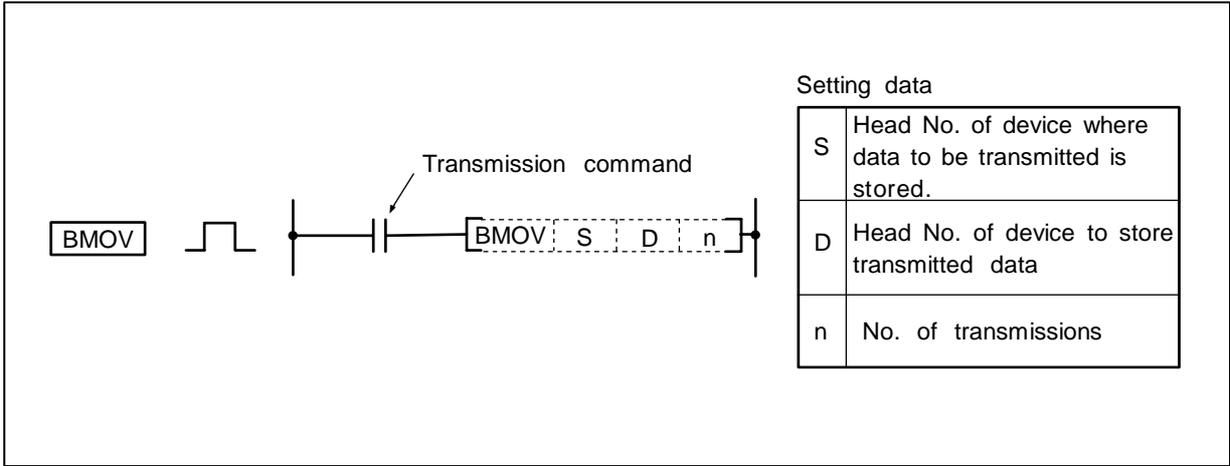
- (3) Program to exchange D0, 1 details with R9, 10 details when M0 turns ON.

**Coding**

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M0 | | |
| 11 | DXCH | D0 | R9 | |
| 15 | | | | |

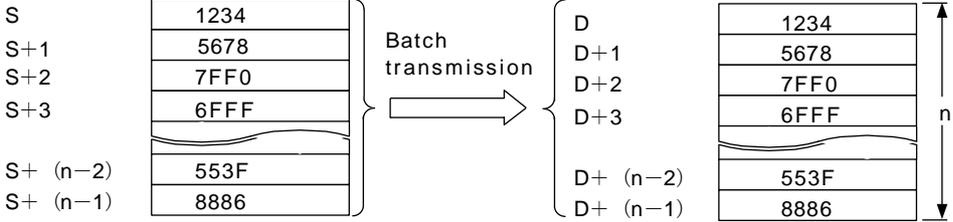
○ BMOV ... Block transmission of 16-bit data

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|--|---|--|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | | | |
| S | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | 5 | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | | |
| n | | | | | | | | | | | | | | | | | ○ | ○ | | | | | |



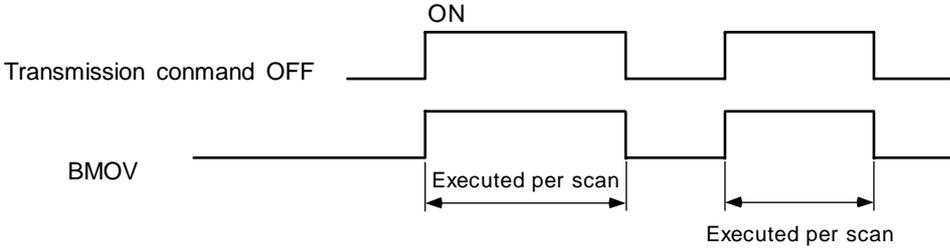
Function

The details of n points from the device designated with S are batch transmitted to the n point designated with D.



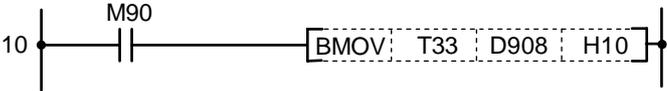
Execution conditions

The execution conditions of the BMOV command are as shown below.



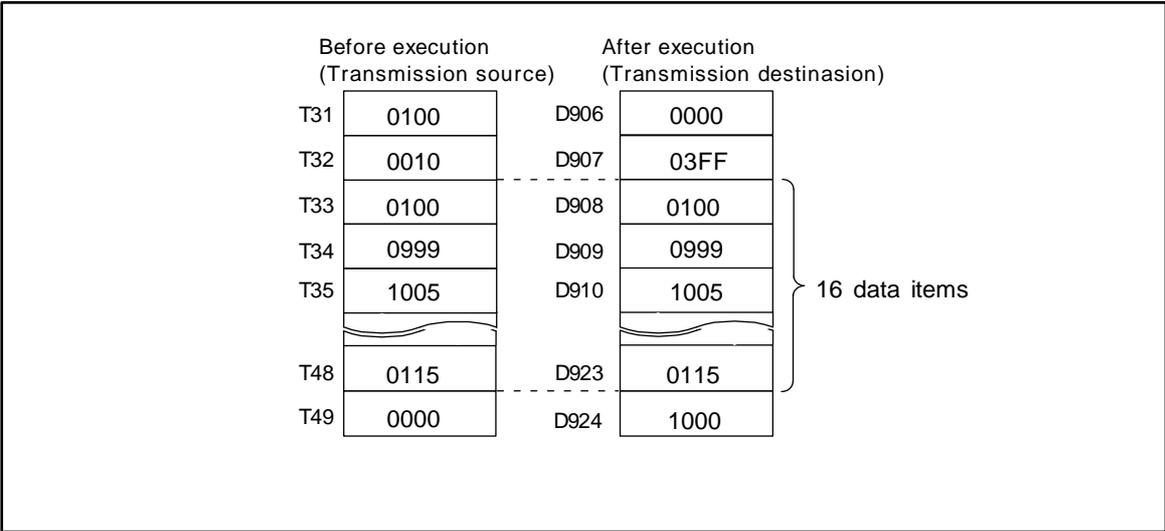
Program example

(1) Program to transmit the current values of T33 to 48 to D908 to 923.



Coding

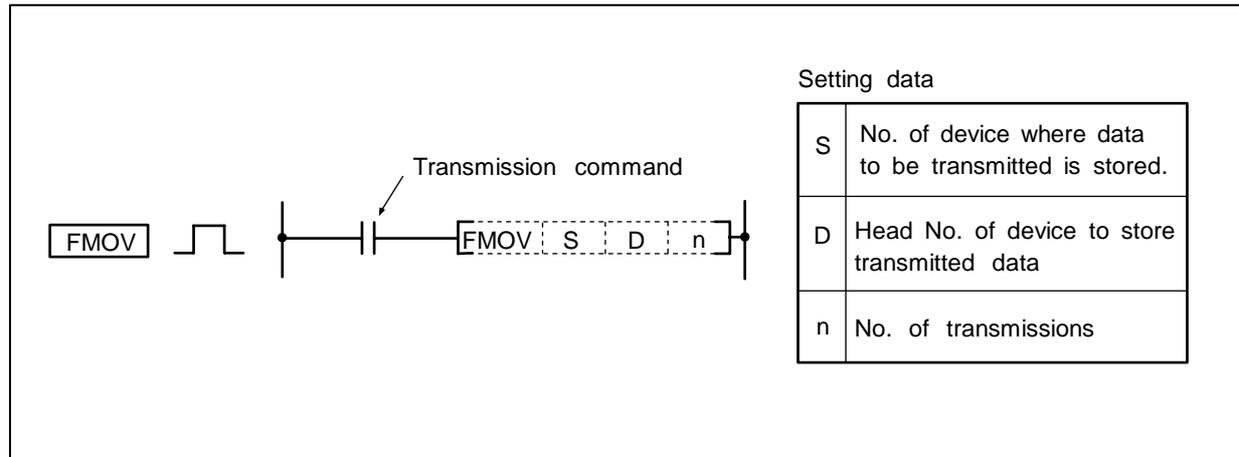
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|------|-----|
| 10 | LD | M90 | | |
| 11 | BMOV | T33 | D908 | H10 |
| 16 | | | | |



Block transmission with BMOV command

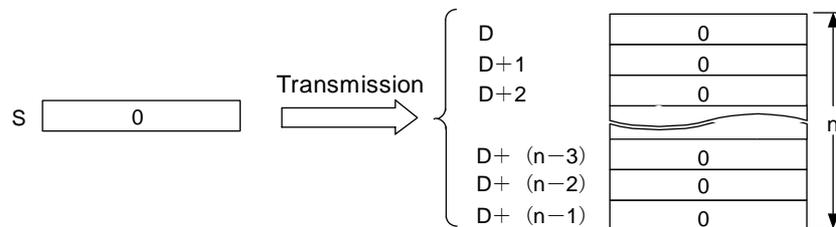
○ FMOV ... Batch transmission of same 16-bit data

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | |
|---|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---|-------------------|--------------|-------|---------|-------|---|--|
| | Bit device | | | | | | Word device | | | | | | Constant | | | | | Pointer | Level | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | | | | H | P | N | |
| S | | | | | | | ○ | ○ | ○ | ○ | | | | | ○ | ○ | | | | 5 | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | |
| n | | | | | | | | | | | | | | | ○ | ○ | | | | | |



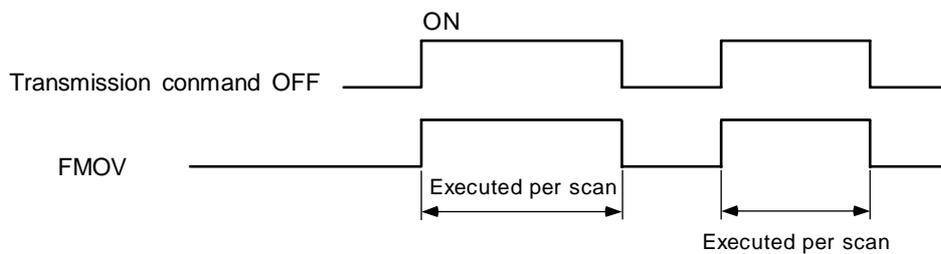
Function

The details of the device designated with S are transmitted to the area by n points designated with D in batch.



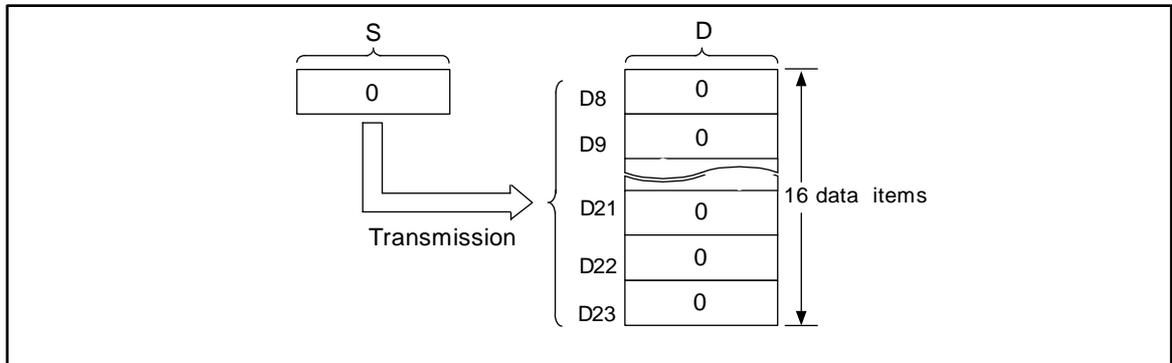
Execution conditions

The execution conditions of the FMOV command are as shown below.

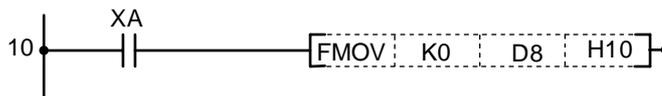


Program example

(1) Program to reset (clear) D8 to 23 when XA turns ON.



Resetting of data registers with FMOV command

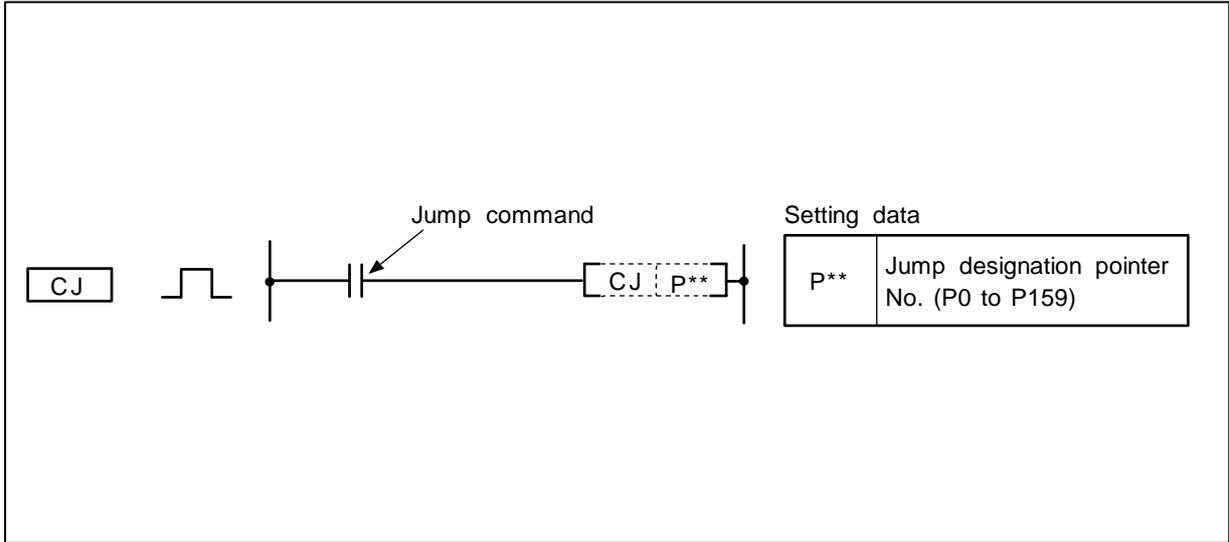


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|-----|
| | | | | |
| 10 | LD | XA | | |
| 11 | FMOV | K0 | D8 | H10 |
| 16 | | | | |

○ CJ ... Conditional jump

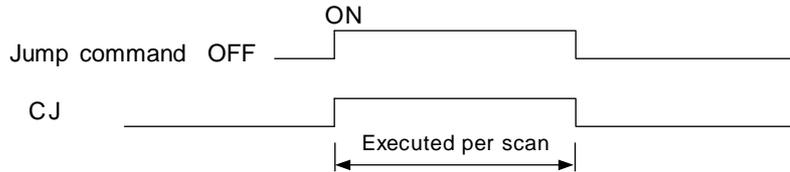
| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---------|-------------------|--------------|-------|-------|---|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | Pointer | | | | Level | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | | | | H | P | N |
| P | | | | | | | | | | | | | | | | ○ | | | 2 | |

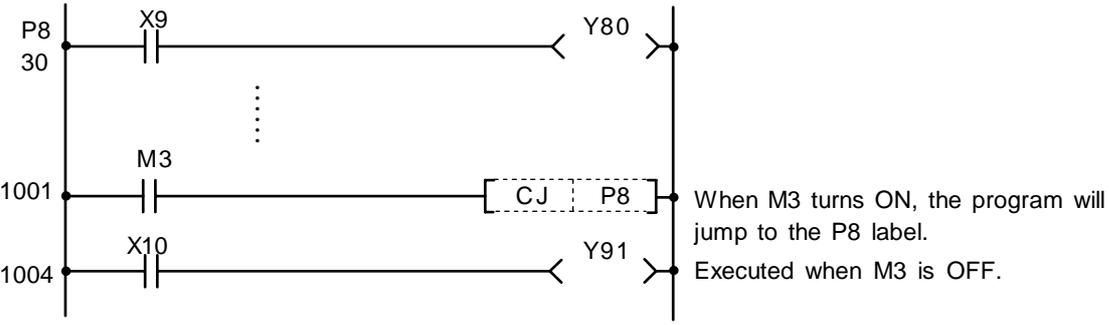
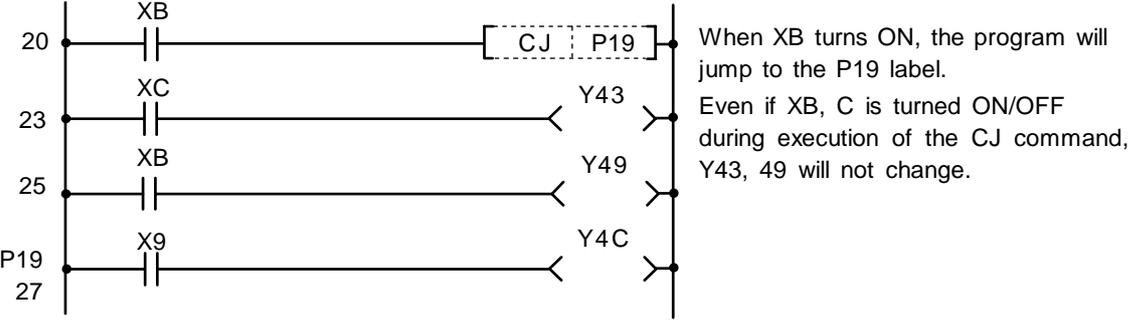
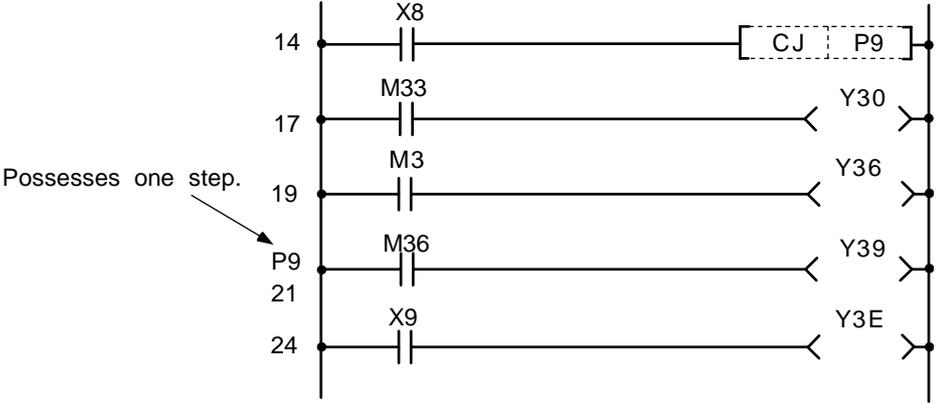


Function

CJ

- (1) The program of the designated pointer No. is executed when the jump command turns ON.
- (2) The program of the next step is executed when the jump command is OFF.

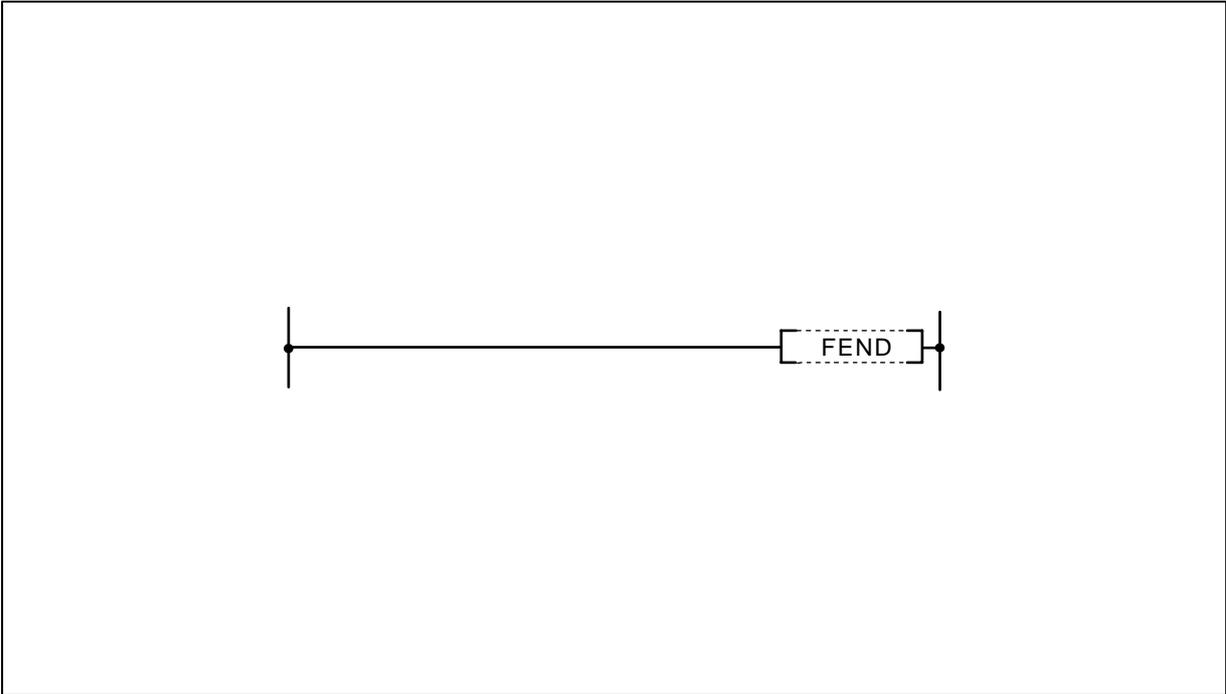


| Point | |
|---|--|
| <p>(a) After the timer coil is turned ON, even if the timer that is turning the coil ON with the CJ command is jumped, the timer count will continue.</p> <p>(b) The scan time will be shortened if jumping is done after the CJ command.</p> <p>(c) The CJ command can be used to jump to a previous step.</p> |  |
| <p>(d) The devices skipped with CJ will not change.</p> |  |
| <p>(e) Label (P**) possesses one step.</p> |  |

- (Notes)**
- (1) Designate the pointer No. so that it comes prior to the END command.
 - (2) Designate the level No. which exists in the program file as the pointer No.

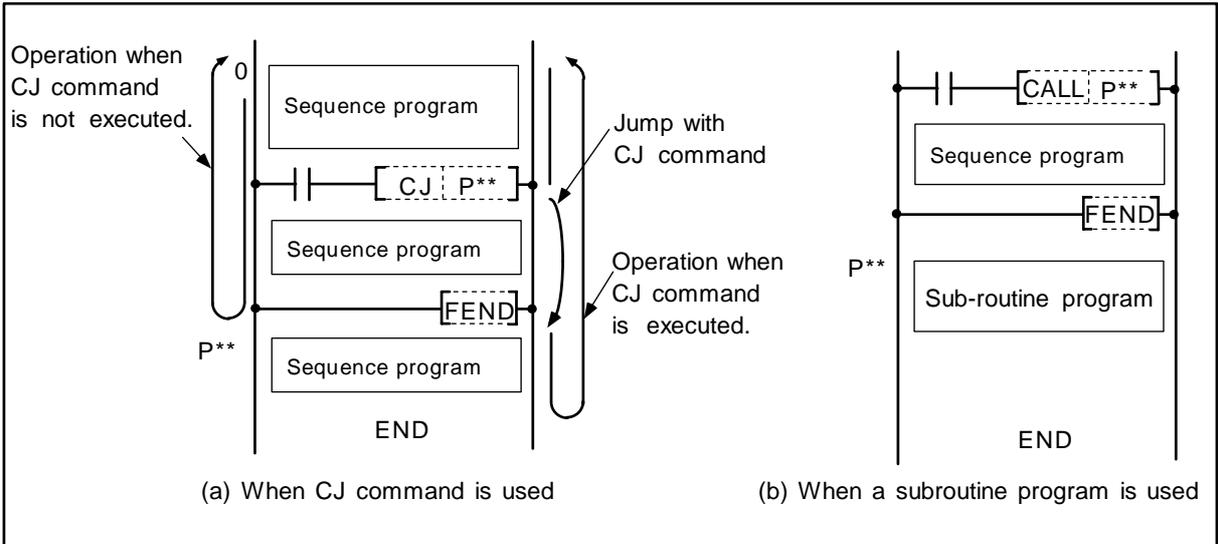
○ FEND ... Program end

| Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | |
|---------------|---|---|---|----|-------------|---|---|---|---|----------|--|---------|-------|-------------------|--------------|-------|---|---|--|
| Bit device | | | | | Word device | | | | | Constant | | Pointer | Level | | | | | | |
| X | Y | M | L | SM | F | T | C | D | R | Z | | K | H | | | | P | N | |
| | | | | | | | | | | | | | | | | | | 1 | |



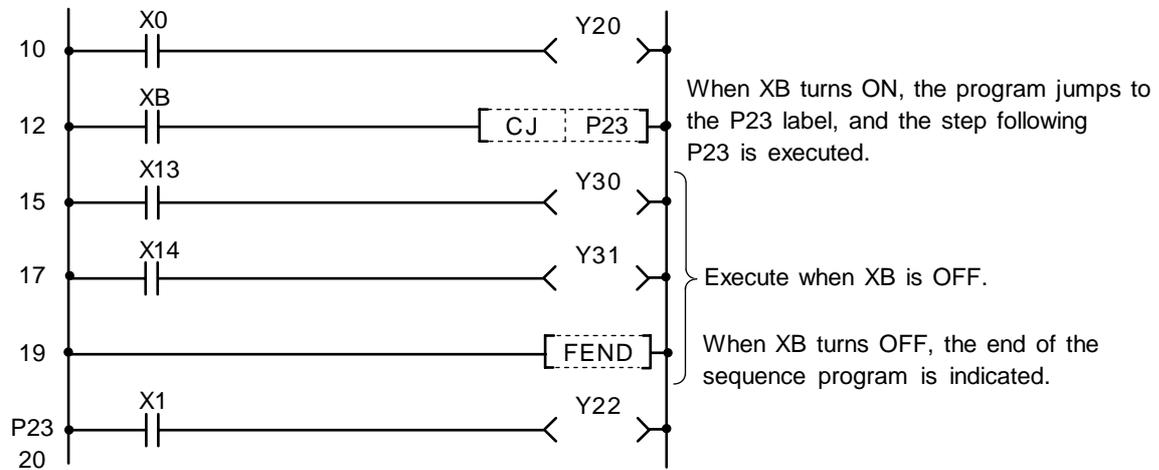
Function

The sequence program is ended.



Program example

Program when using CJ command

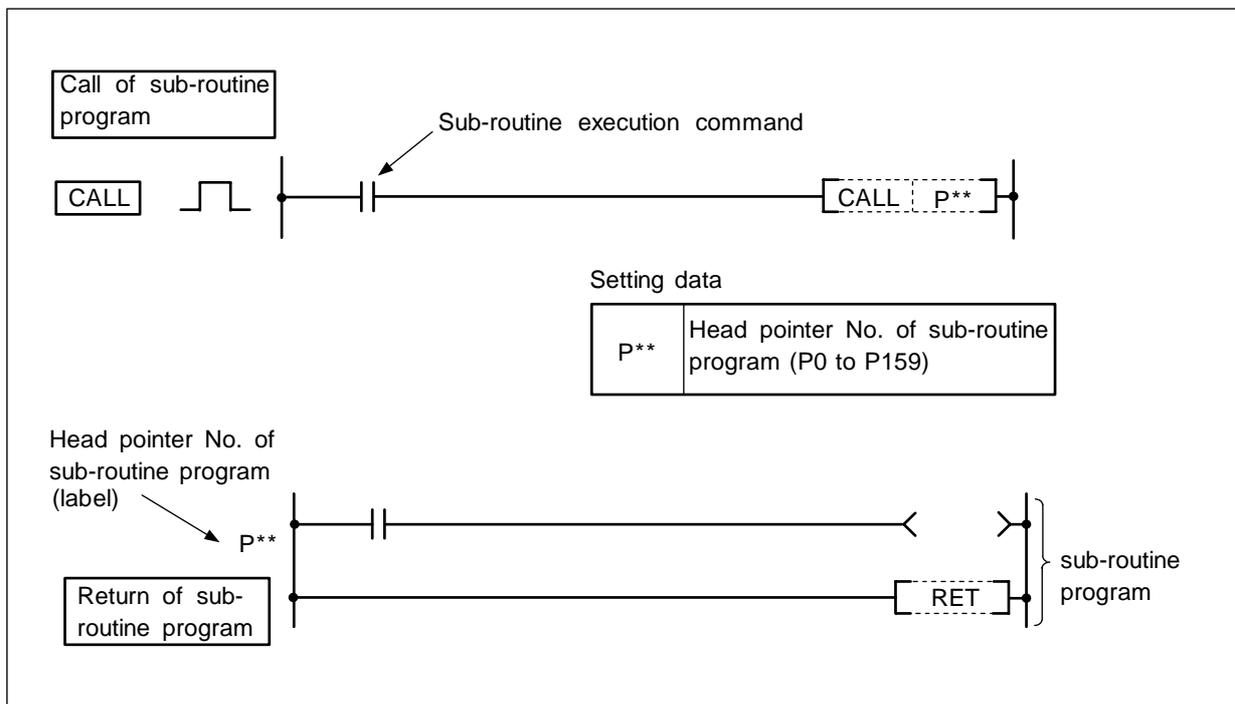


Coding

| 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, RET ... Call/return of sub-routine program

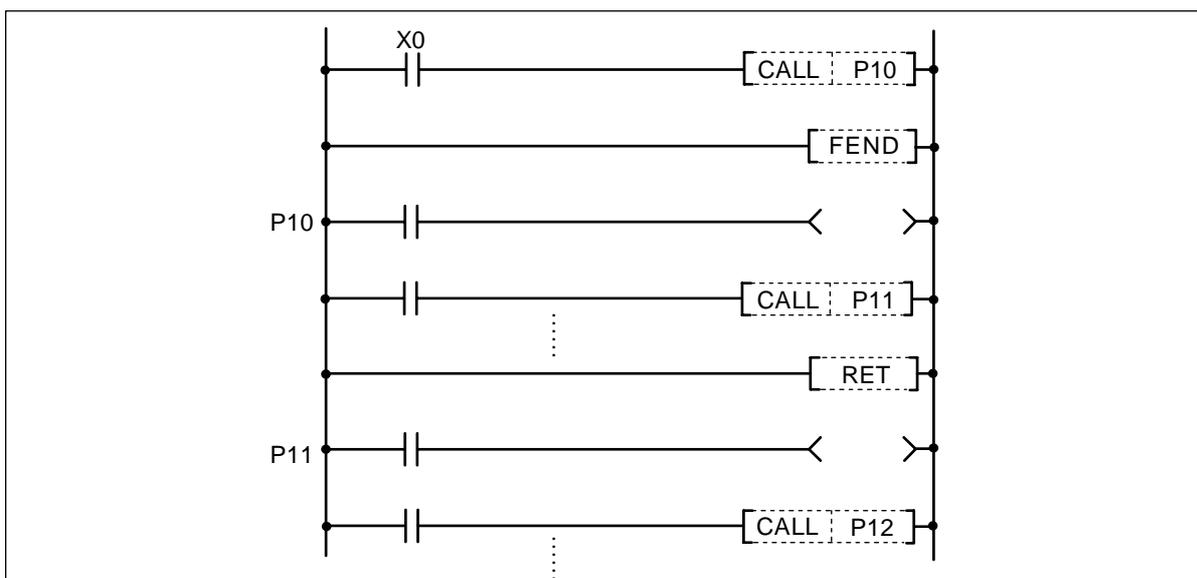
| P | Usable device | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | |
|---|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|-------------------|--------------|-------|---------|-------|-----|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | | | Pointer | Level | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | K | H | P | N |
| | | | | | | | | | | | | | | | | ○ | | | 2/1 | |



Function

CALL

(1) The sub-routine program designated with the point (P**) is executed.

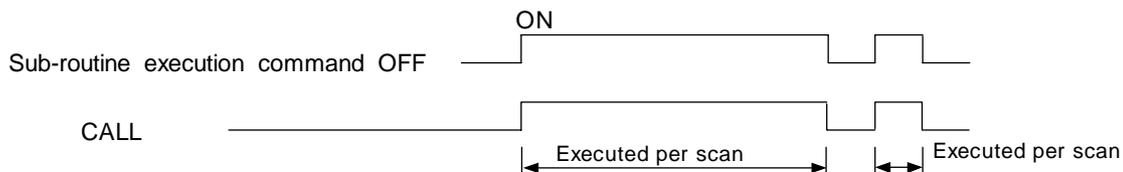


RET

- (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 command will be executed.

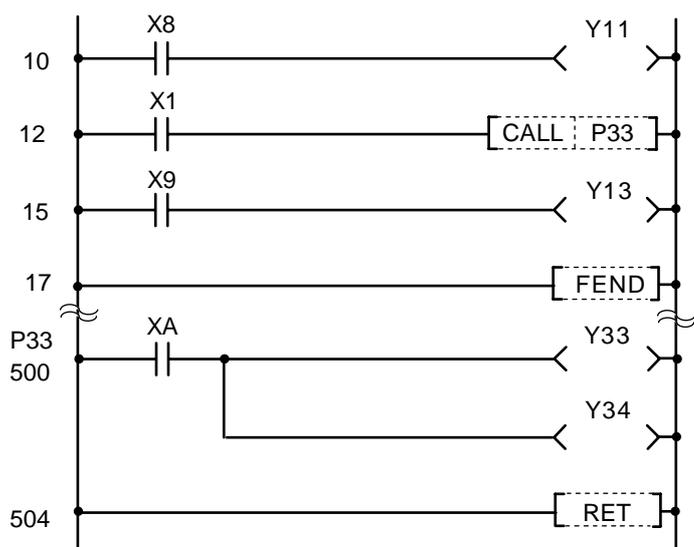
Execution conditions

The execution conditions of the CALL command are as shown below.



Program example

Program to execute sub-routine program when X1 changes from OFF to ON.

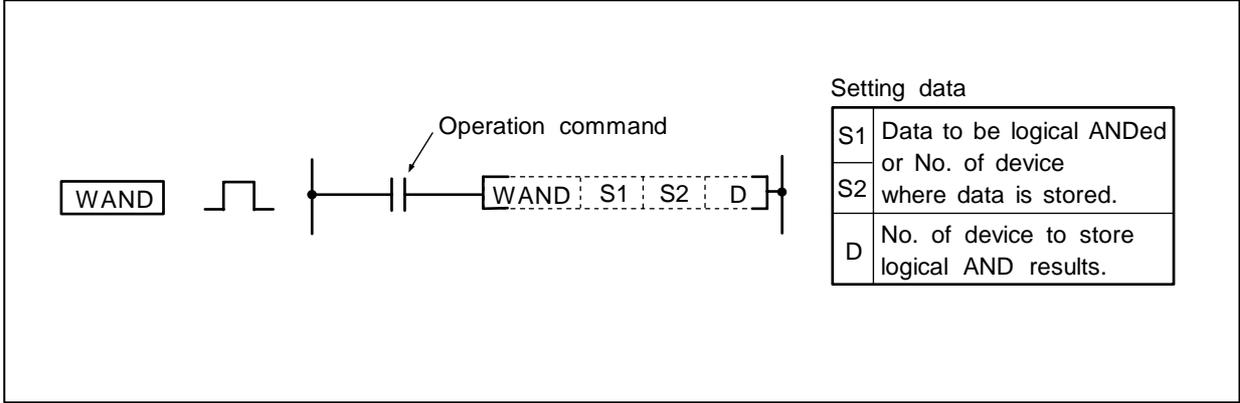


Coding

| 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 | | | | |

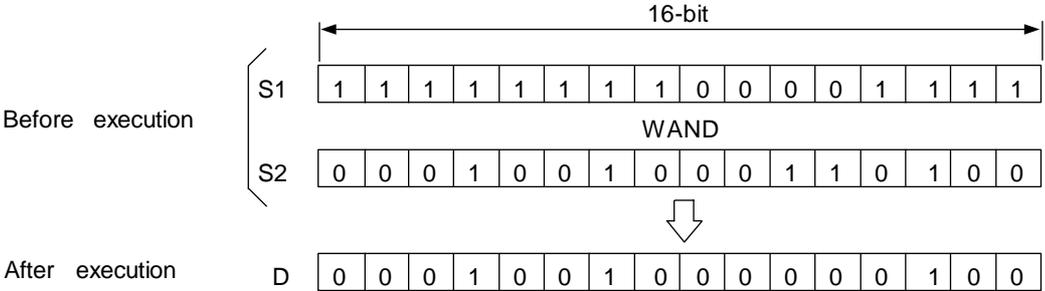
○ WAND ... Logical AND of 16-bit data

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | | |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|---|---|--|--|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | | | | |
| S1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | | | ○ | 4 | | |
| S2 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | ○ | ○ | | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | |



Function

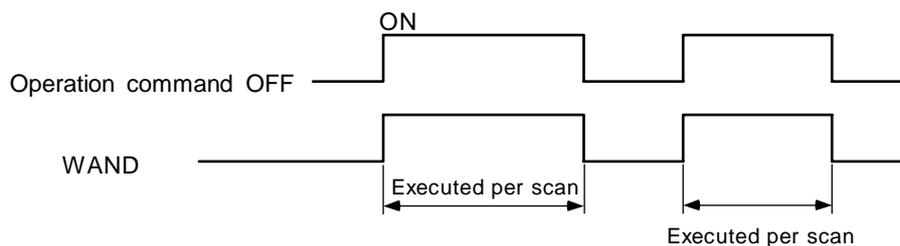
- (1) Logical AND is executed for each bit of the 16-bit data in the device designated with S1 and the device designated with S2, and the results are stored in the device designated with D.



- (2) The bit device other than the designated digits are operated as 0. (Refer to program example (2).)

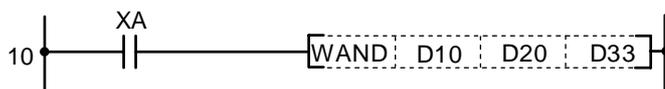
Execution conditions

The execution conditions for WAND are as follow.



Program example

- (1) Program that executes logical AND of the D10 data and D20 data when XA turns ON, and stores the results in D33.



Coding

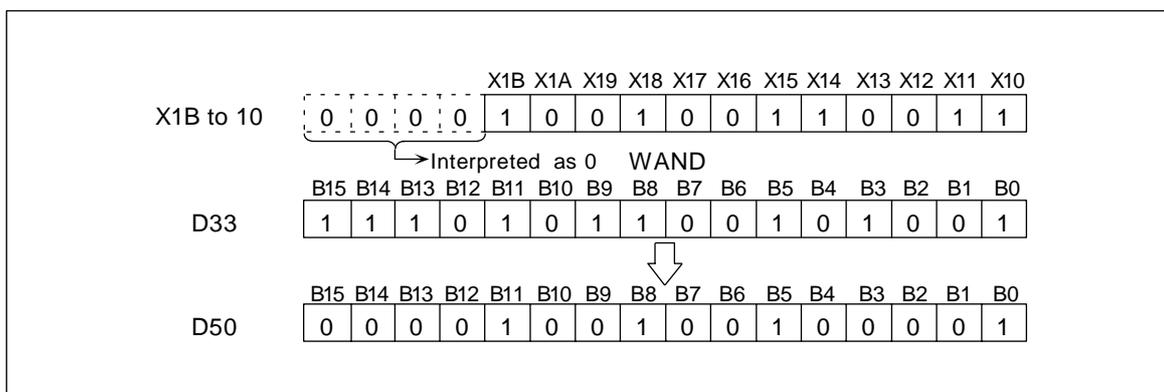
| No. of steps | Com- mand | Device | | |
|--------------|--------------|--------|-----|-----|
| 10 | LD | XA | | |
| 11 | WAND | D10 | D20 | D33 |
| 15 | | | | |

- (2) Program that executes logical AND of the X10 to 1B data and D33 data when XA turns ON, and outputs the results to D50.



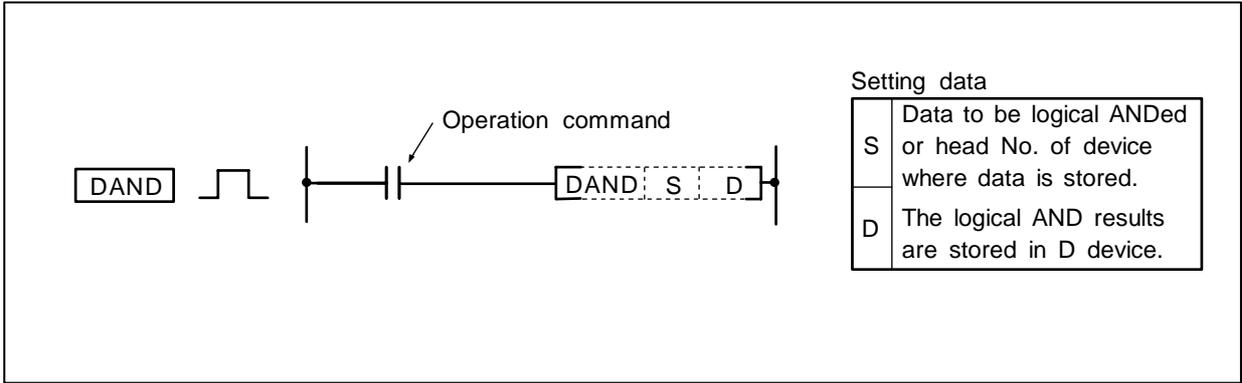
Coding

| No. of steps | Com- mand | Device | | |
|--------------|--------------|--------|-----|-----|
| 10 | LD | XA | | |
| 11 | WAND | K3X10 | D33 | D50 |
| 15 | | | | |



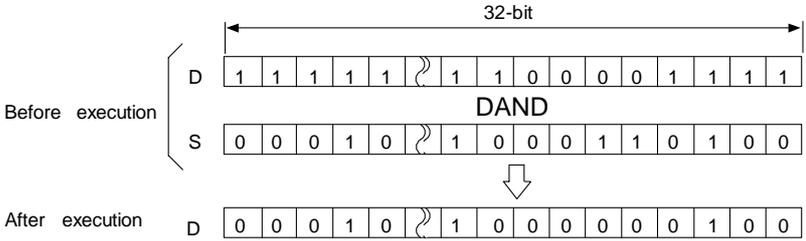
○ DAND ... Logical AND of 32-bit data

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|--|---------|-------|-------------------|--------------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | K | | | |
| S | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | ○ | ○ | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |



Function

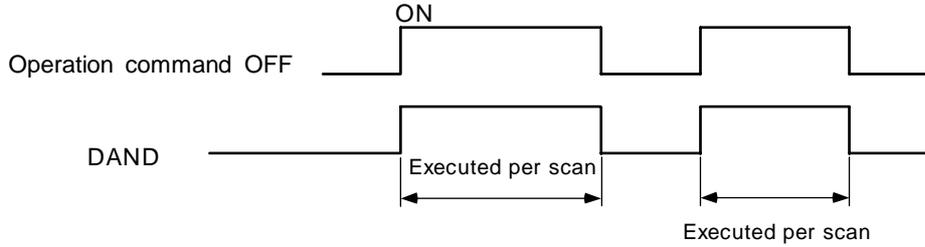
- (1) Logical AND is executed for each bit of the 32-bit data in the device designated with D and the device designated in S, and the results are stored in the device designated with D.



- (2) The bit device other than the designated digits are operated as 0.
(Refer to program example (1).)

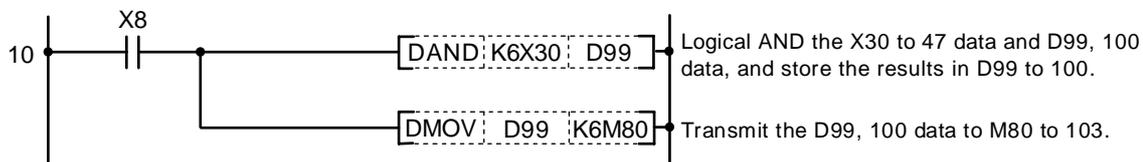
Execution conditions

The execution conditions for the DAND command are as follow.



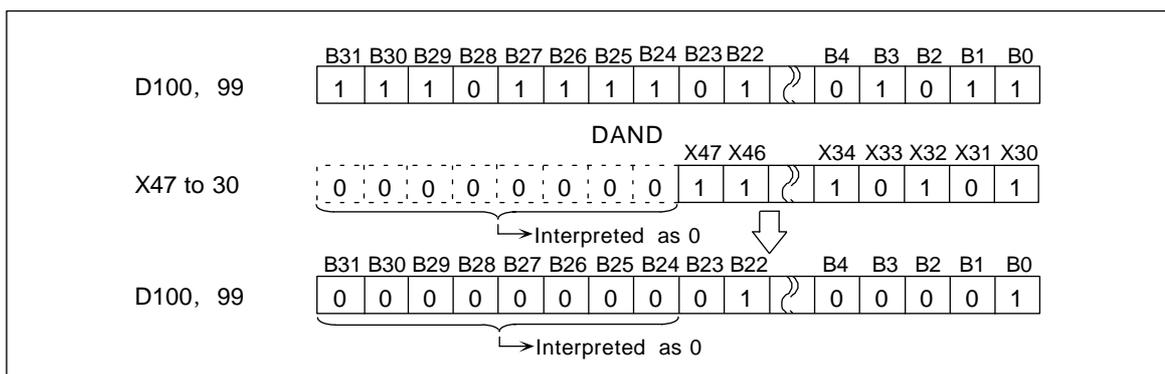
Program example

- (1) Program that executes logical AND of the X30 to 47 24-bit data and D99, 100 data when X8 turns ON, and transmit the results to M80 to 103.

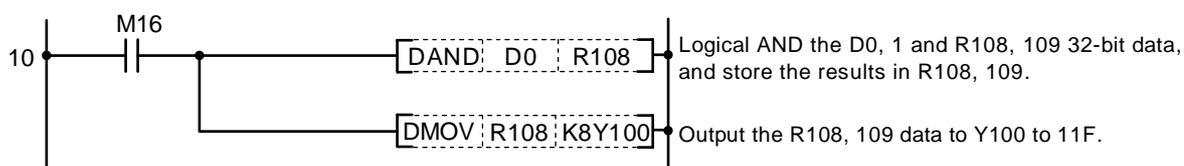


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-------|--|
| 10 | LD | X8 | | |
| 11 | DAND | K6X30 | D99 | |
| 14 | DMOV | D99 | K6M80 | |
| 18 | | | | |



- (2) Program that executes logical AND of the D0, 1 32-bit data and R108, 109 when M16 turns ON, and outputs the results to Y100 to 11F.

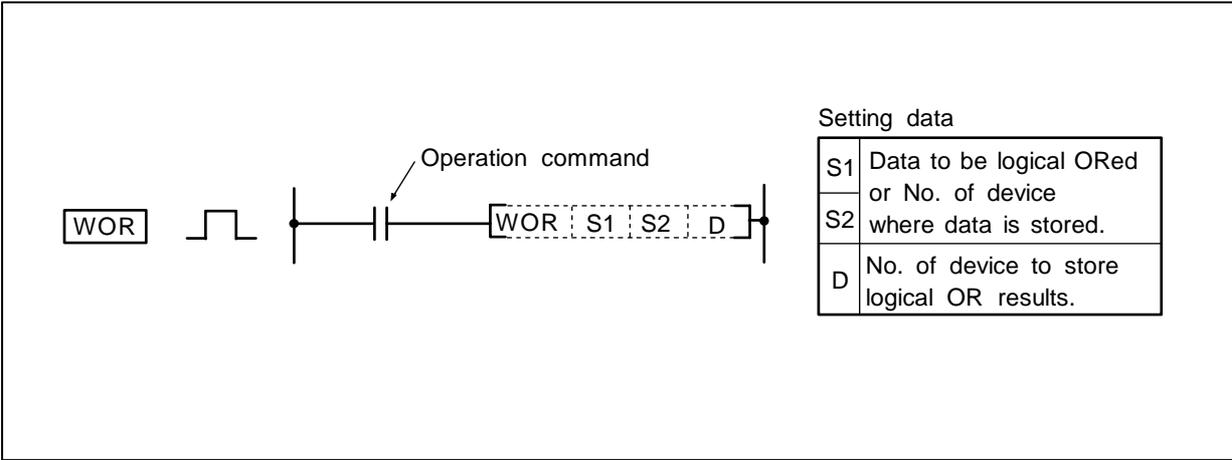


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|--------|--|
| 10 | LD | M16 | | |
| 11 | DAND | D0 | R108 | |
| 14 | DMOV | R108 | K8Y100 | |
| 18 | | | | |

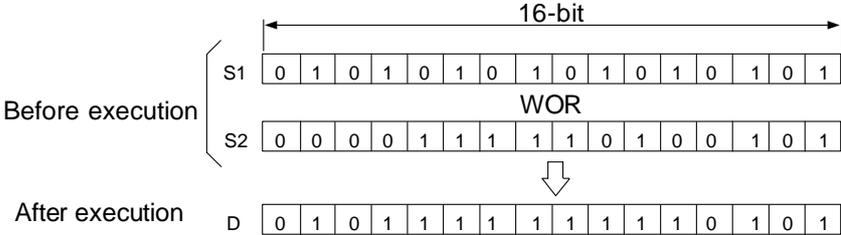
○ WOR ... Logical OR of 16-bit data

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | |
|-----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|---|---|--|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | | | |
| S1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | | | ○ | 4 | |
| S2 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | ○ | ○ | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | | |



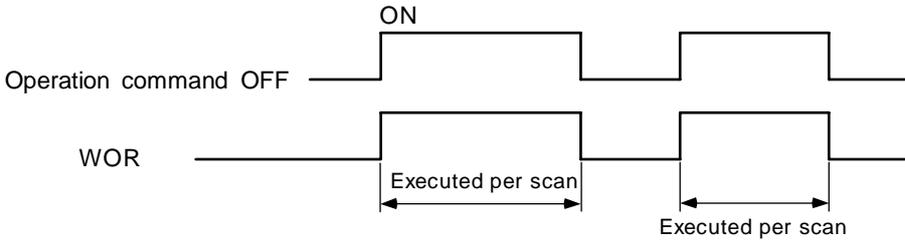
Function

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.



Execution conditions

The execution conditions for WOR are as follow.



Program example

- (1) Program that executes logical OR of the D10 data and D20 data when XA turns ON, and stores the results in D33.



Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-----|-----|
| 10 | LD | XA | | |
| 11 | WOR | D10 | D20 | D33 |
| 15 | | | | |

- (2) Program that executes logical OR of the X10 to 1B data and D33 data when XA turns ON, and outputs the results in D100.

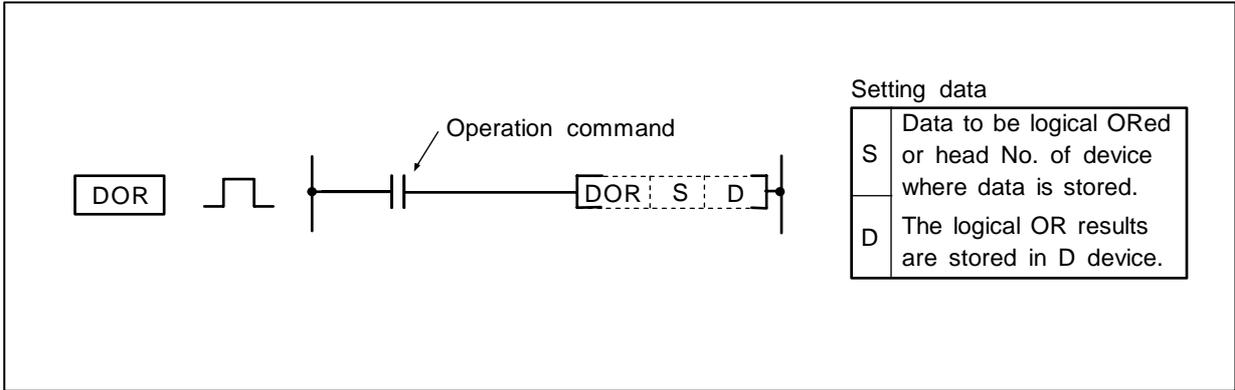


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-----|------|
| 10 | LD | XA | | |
| 11 | WOR | K3X10 | D33 | D100 |
| 15 | | | | |

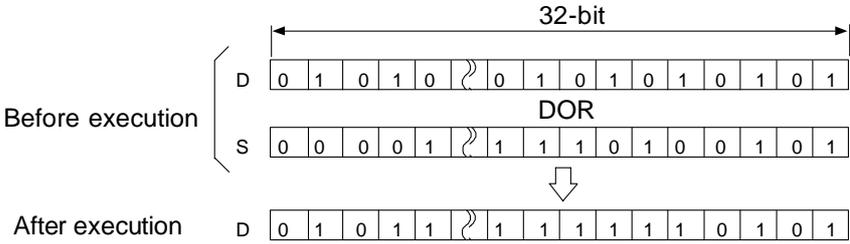
○ DOR ... Logical OR of 32-bit data

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---|---------|-------|-------------------|--------------|-------|---|-----|--|--|
| | Bit device | | | | | | Word device | | | | | | Constant | | Pointer | Level | | | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | | | |
| S | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | ○ | ○ | | | | | ○ | 3/4 | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | | |



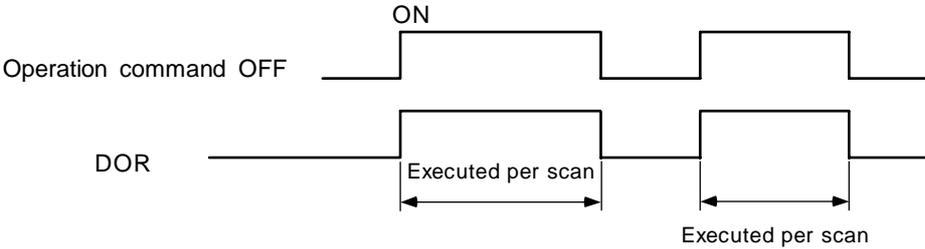
Function

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.



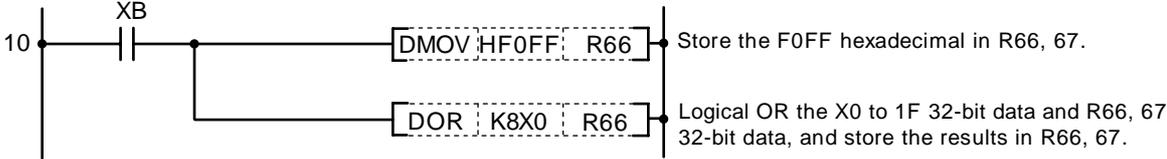
Execution conditions

The execution conditions for DOR are as follow.



Program example

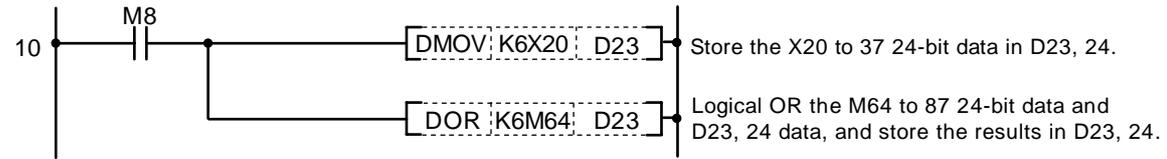
(1) Program that executes logical OR of the X0 to 1F 32-bit data and the F0FF hexadecimal when XB turns ON, and stores the results in R66, 67.



Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-----|--|
| 10 | LD | XB | | |
| 11 | DMOV | HFOFF | R66 | |
| 14 | DOR | K8X0 | R66 | |
| 18 | | | | |

(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.

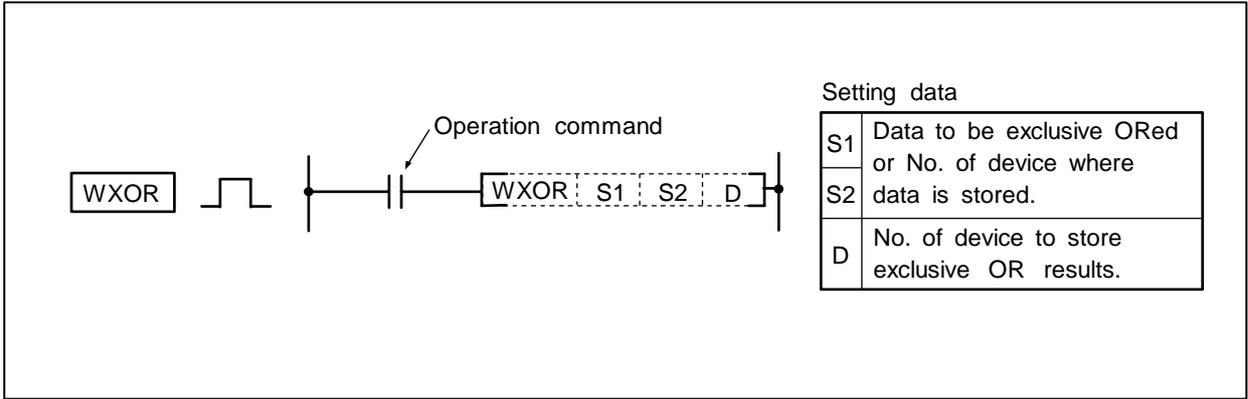


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-----|--|
| 10 | LD | M8 | | |
| 11 | DMOV | K6X20 | D23 | |
| 14 | DOR | K6M64 | D23 | |
| 18 | | | | |

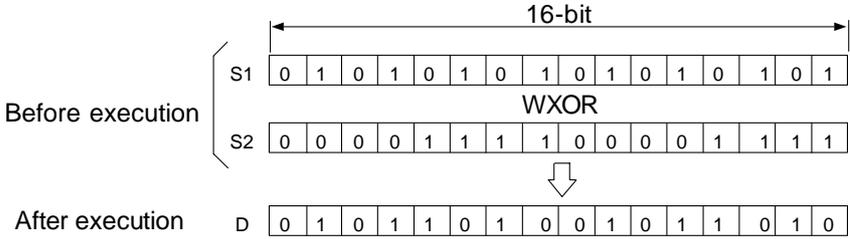
○ WXOR ... Exclusive OR of 16-bit data

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | |
|-----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---|---------|-------|-------------------|--------------|-------|---|
| | Bit device | | | | | | Word device | | | | | | Constant | | Pointer | Level | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N |
| S1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | | |
| S2 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | ○ | ○ | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |



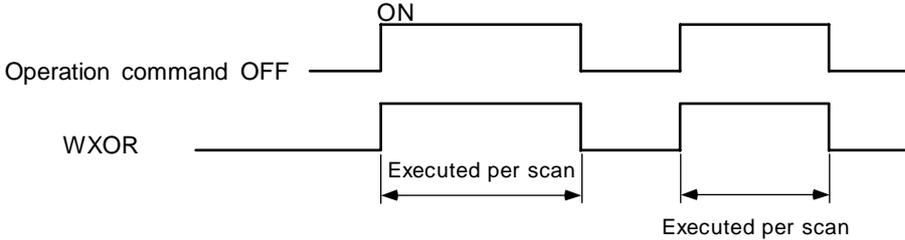
Function

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.



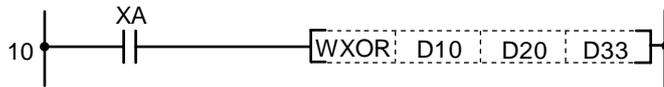
Execution conditions

The execution conditions for WXOR are as follow.



Program example

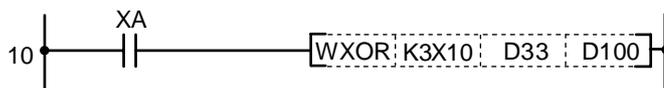
- (1) Program that executes exclusive OR of the D10 data and D20 data when XA turns ON, and stores the results in D33.



Coding

| No. of steps | Command | Device | | |
|--------------|---------|--------|-----|-----|
| 10 | LD | XA | | |
| 11 | WXOR | D10 | D20 | D33 |
| 15 | | | | |

- (2) Program that executes exclusive OR of the X10 to 1B data and D33 data when XA turns ON, and outputs the results to D100.

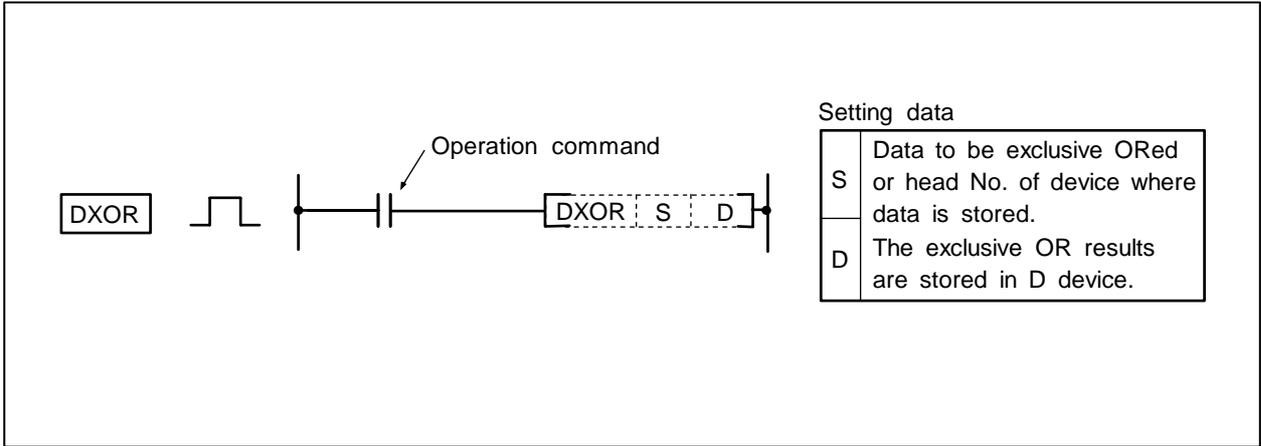


Coding

| No. of steps | Command | Device | | |
|--------------|---------|--------|-----|------|
| 10 | LD | XA | | |
| 11 | WXOR | K3X10 | D33 | D100 |
| 15 | | | | |

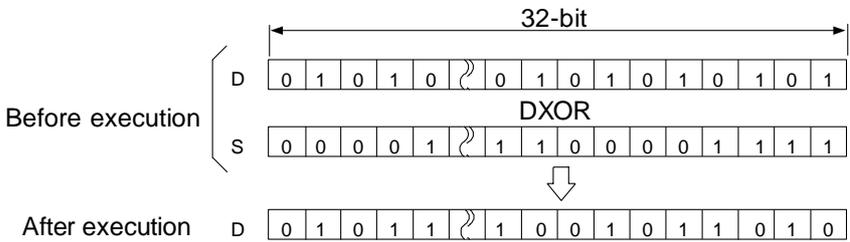
○ DXOR ... Exclusive OR of 32-bit data

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---|---------|-------|-------------------|--------------|-------|---|-----|--|--|
| | Bit device | | | | | | Word device | | | | | | Constant | | Pointer | Level | | | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | | | |
| S | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | ○ | ○ | | | | | ○ | 3/4 | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | | |



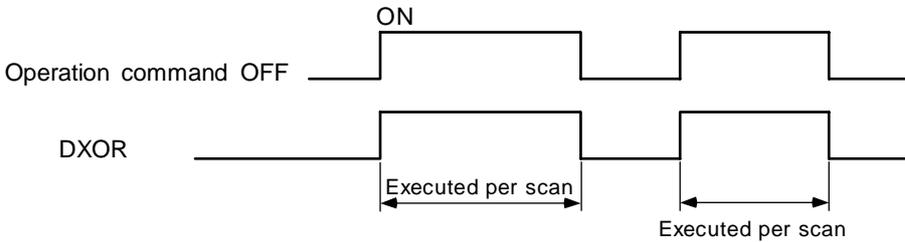
Function

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.



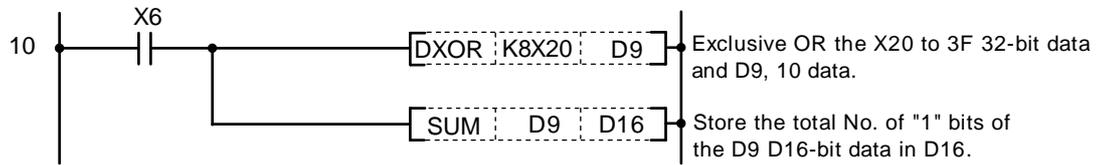
Execution conditions

The execution conditions for DXOR are as follow.



Program example

- (1) Program that compares the X20 to 3F 32-bit data and the D9, 10 data when X6 turns ON, and stores the differing No. of bits in D16.

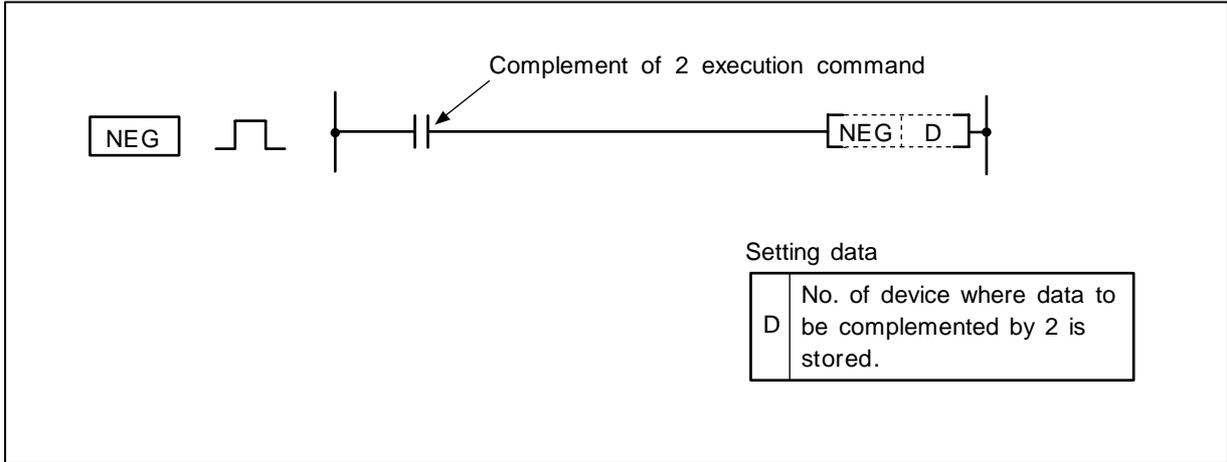


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-----|--|
| 10 | LD | X6 | | |
| 11 | DXOR | K8X20 | D9 | |
| 14 | SUM | D9 | D16 | |
| 18 | | | | |

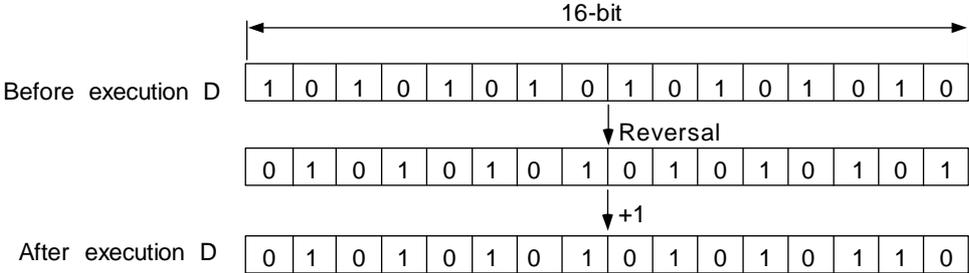
○ NEG ... Complement of 2 (BIN 16-bit data)

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---------|-------------------|--------------|-------|-------|---|---|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | Pointer | | | | Level | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | K | H | P | N |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | 2 | |



Function

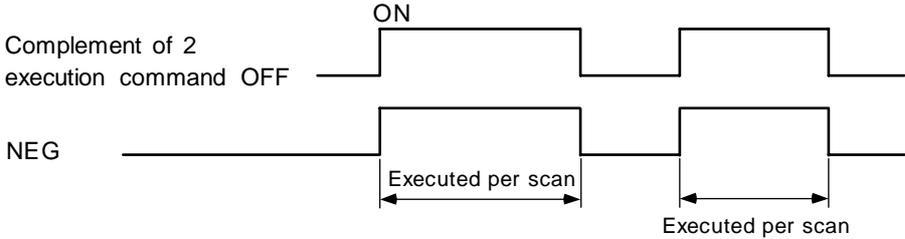
- (1) The 16-bit data of the device designated with D is reversed and incremented by one, and then stored in the device designated with D.



- (2) This is used to use a negative BIN value as an absolute value.

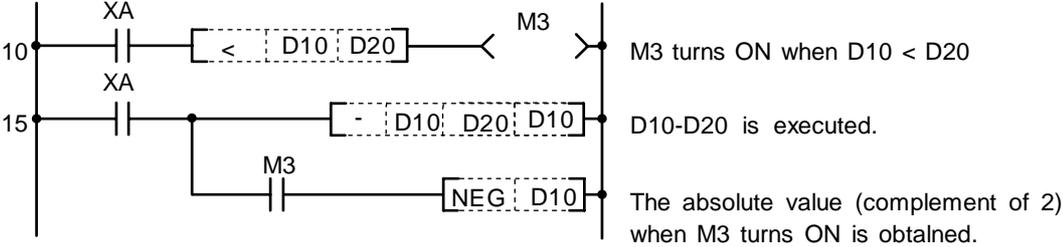
Execution conditions

The execution conditions for NEG are as follow.



Program example

(1) Program to calculate D10 - D20 when XA turns ON and obtain an absolute value when the results are negative.

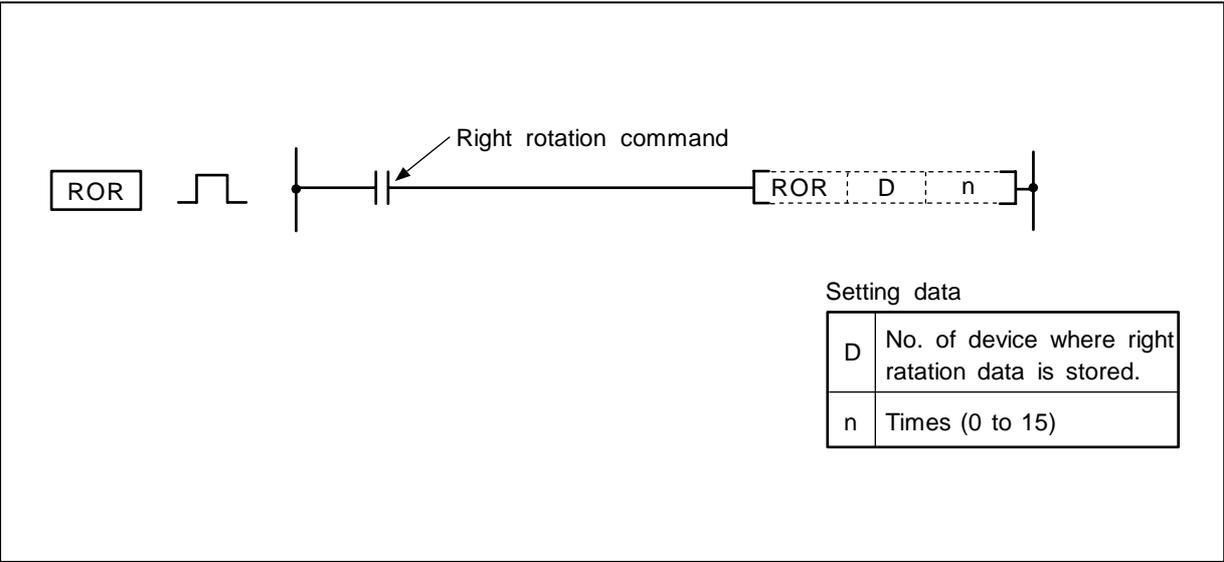


Coding

| 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 | | | | |

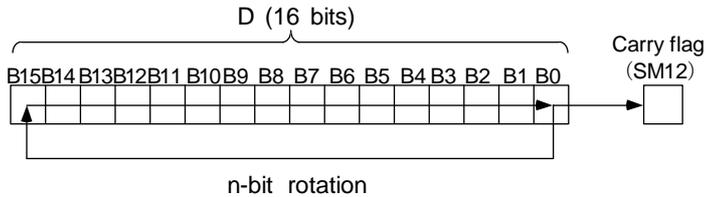
○ ROR ... Right rotation of 16-bit data

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|--|-------------------|--------------|-------|---------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | | | | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | K | H |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| n | | | | | | | | | | | | | | | ○ | ○ | | | |



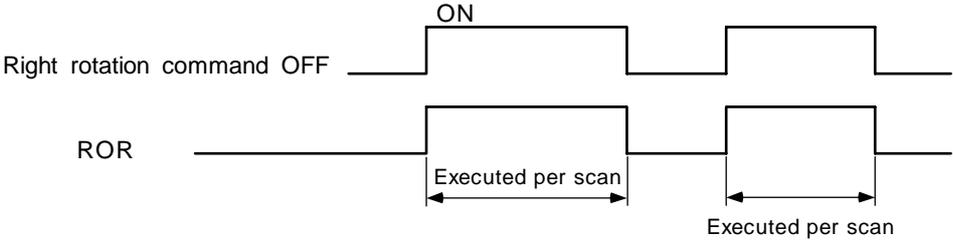
Function

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 command are as shown below.



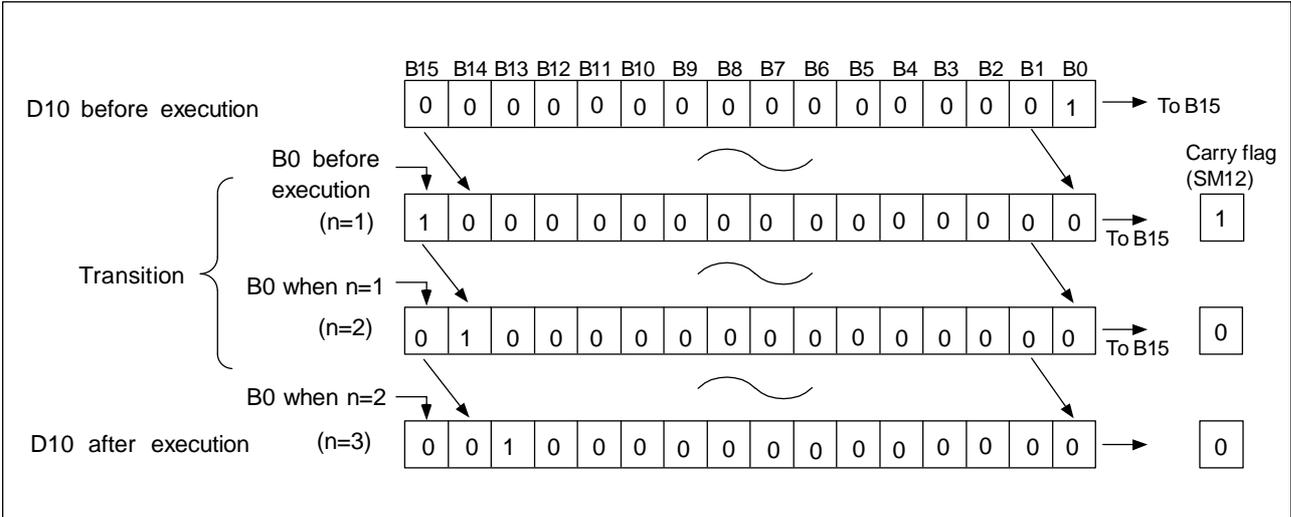
Program example

Program to rotate the D10 details 3 bits to the right when M0 turns ON.



Coding

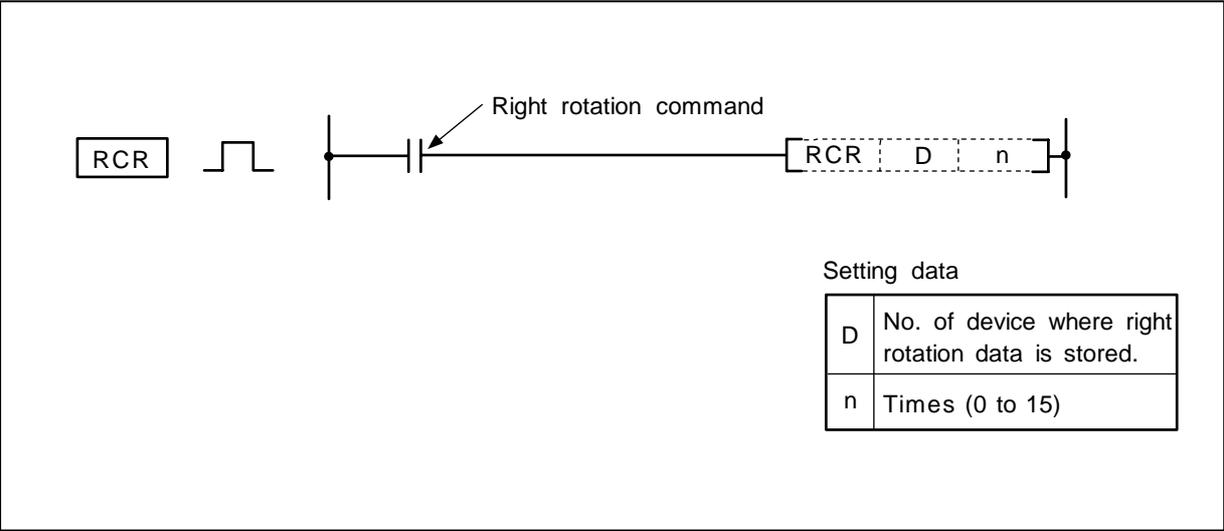
| No. of steps | Command | Device | | |
|--------------|---------|--------|----|--|
| 10 | LD | M0 | | |
| 11 | ROR | D10 | K3 | |
| 14 | | | | |



Right rotation of data using ROR command

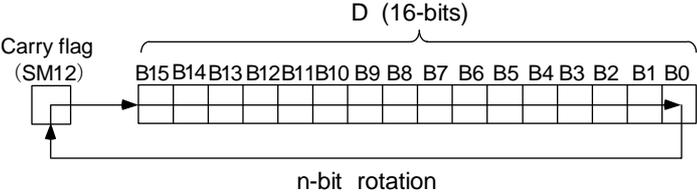
○ RCR ... Right rotation of 16-bit data

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|-------------------|--------------|-------|---------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | | | | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | | | | H | P |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| n | | | | | | | | | | | | | | | ○ | ○ | | | |



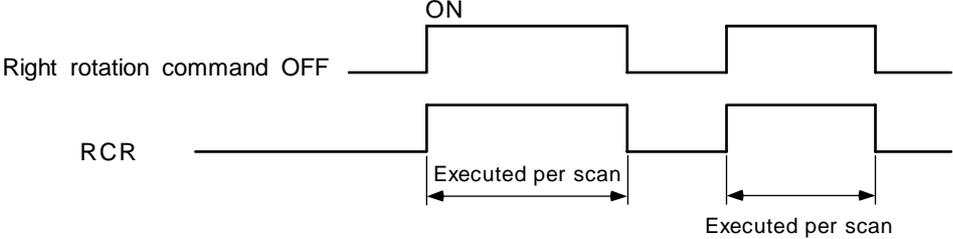
Function

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.



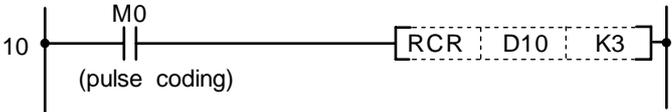
Execution conditions

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



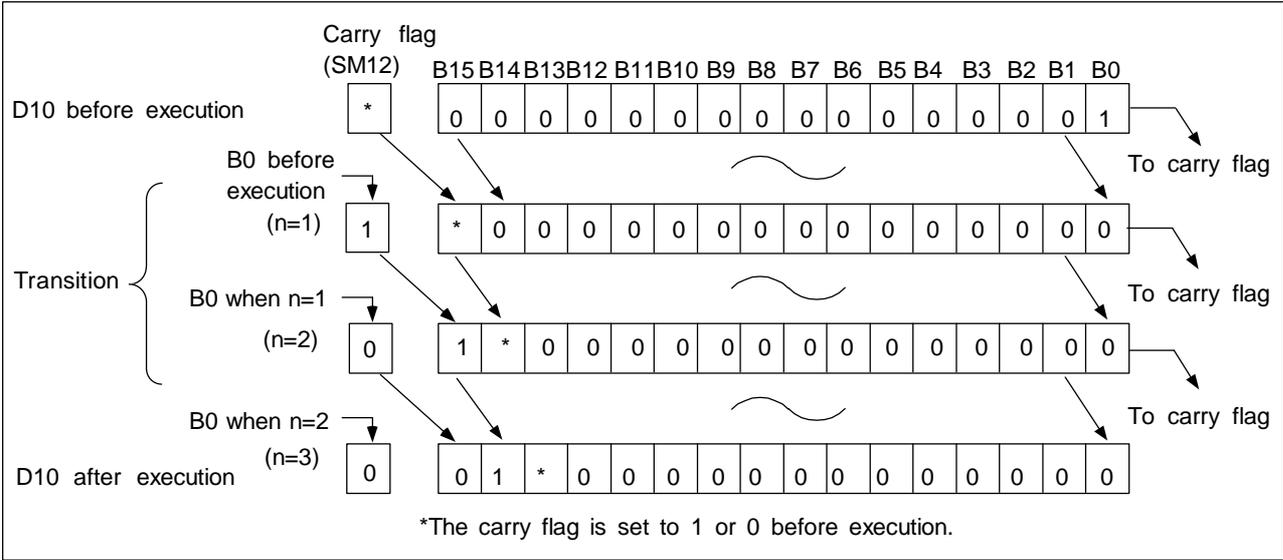
Program example

Program to rotate the D10 details 3 bits to the right when M0 turns ON.



Coding

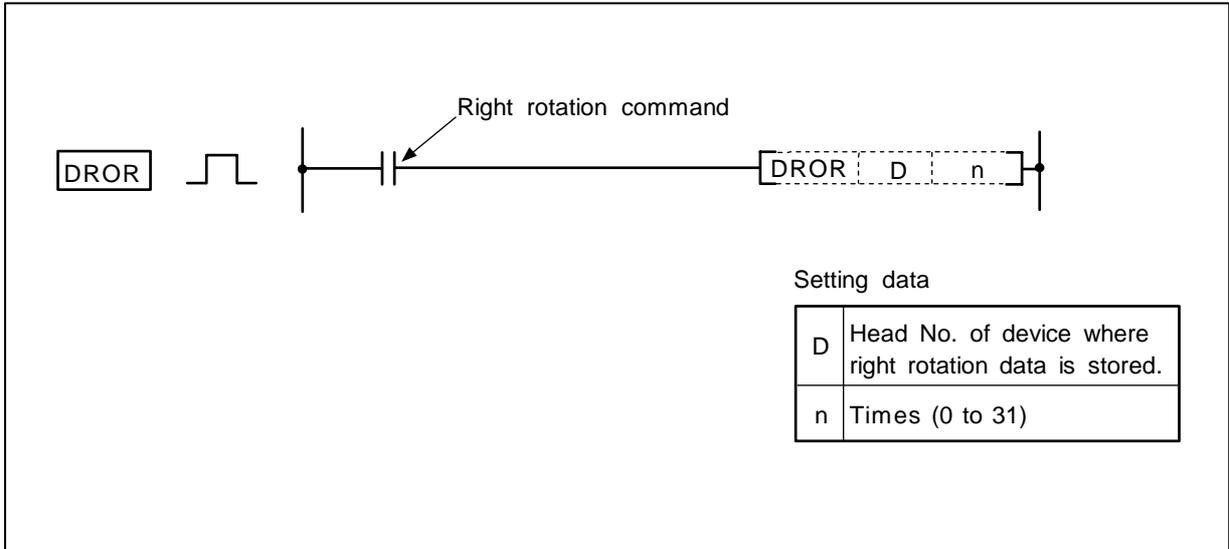
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M0 | | |
| 11 | RCR | D10 | K3 | |
| 14 | | | | |



Right rotation of data using RCR command

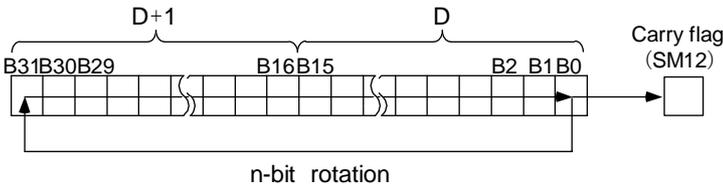
○ DROR ... Right rotation of 32-bit data

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|--|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | |
| n | | | | | | | | | | | | | | ○ | ○ | | | | | | |



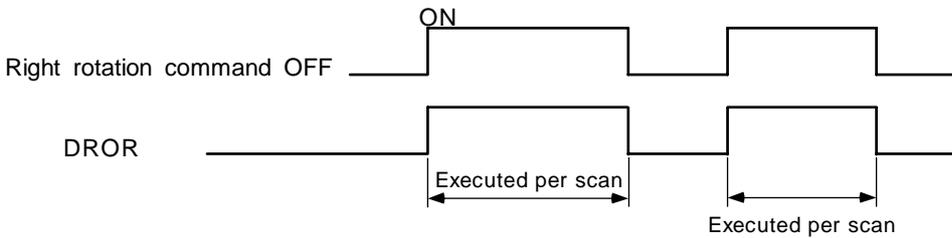
Function

The 32-bit data designated with D is rotated n bits to the right excluding the carry flag.



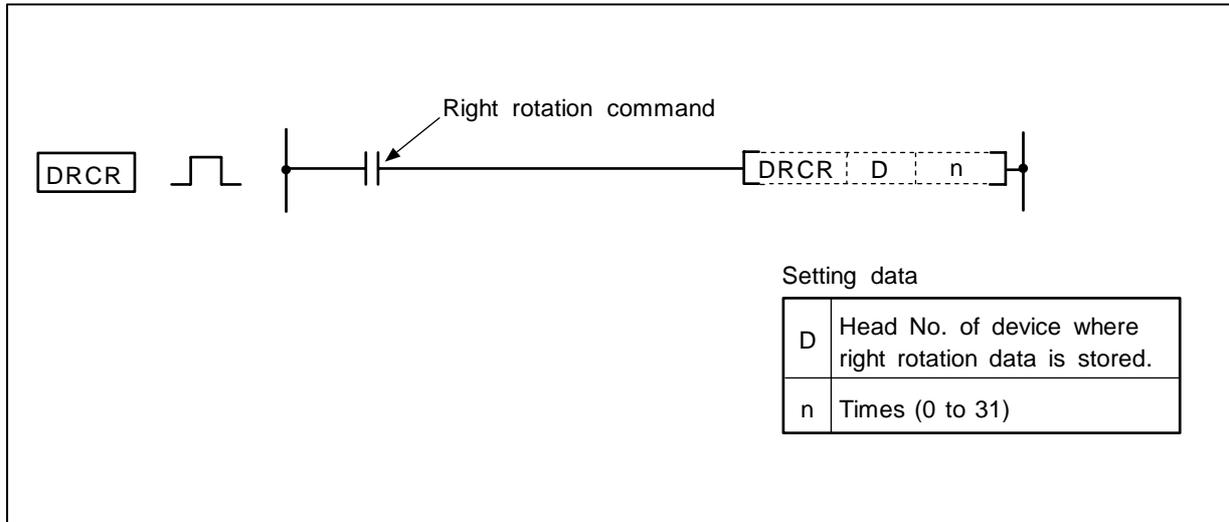
Execution conditions

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



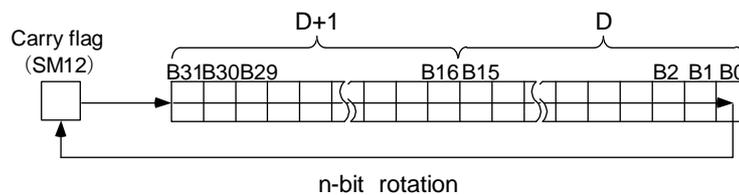
○ DRCR ... Right rotation of 32-bit data

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|--|--|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | | |
| n | | | | | | | | | | | | | ○ | ○ | | | | | | | | | 3 |



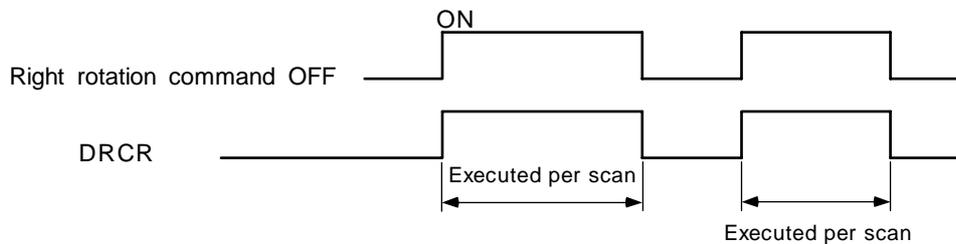
Function

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.



Execution conditions

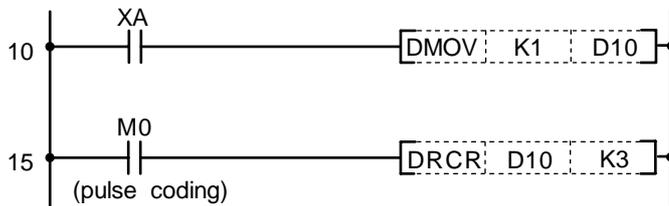
The execution conditions for the DRCR command are as shown below.



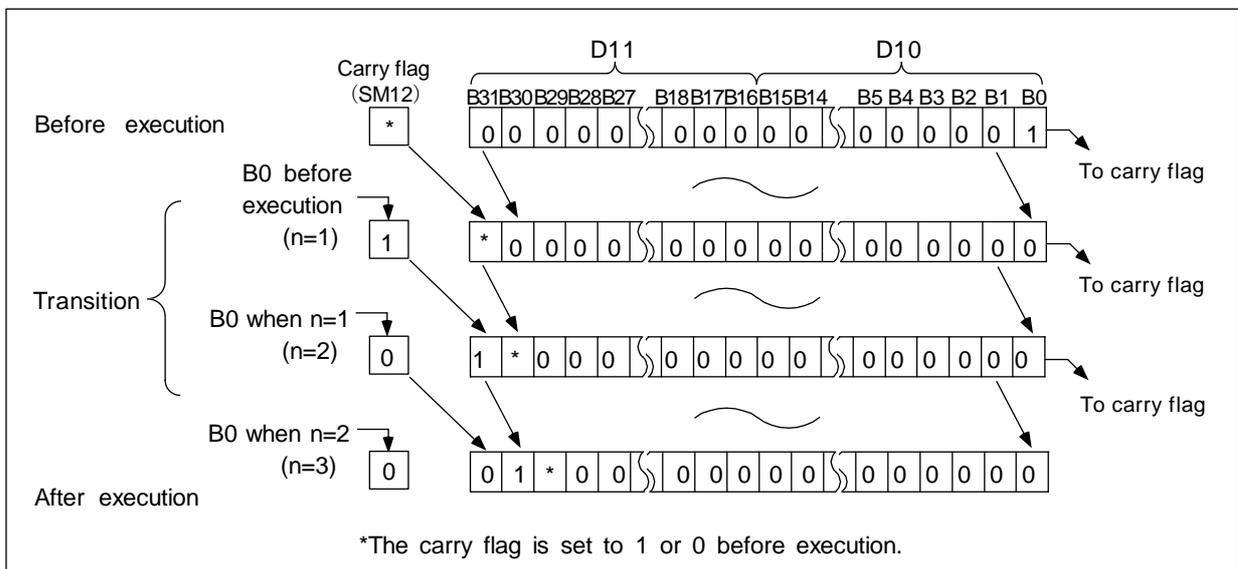
Program example

Program to rotate the D10, 11 details 3 bits to the right when M0 turns ON.

Coding



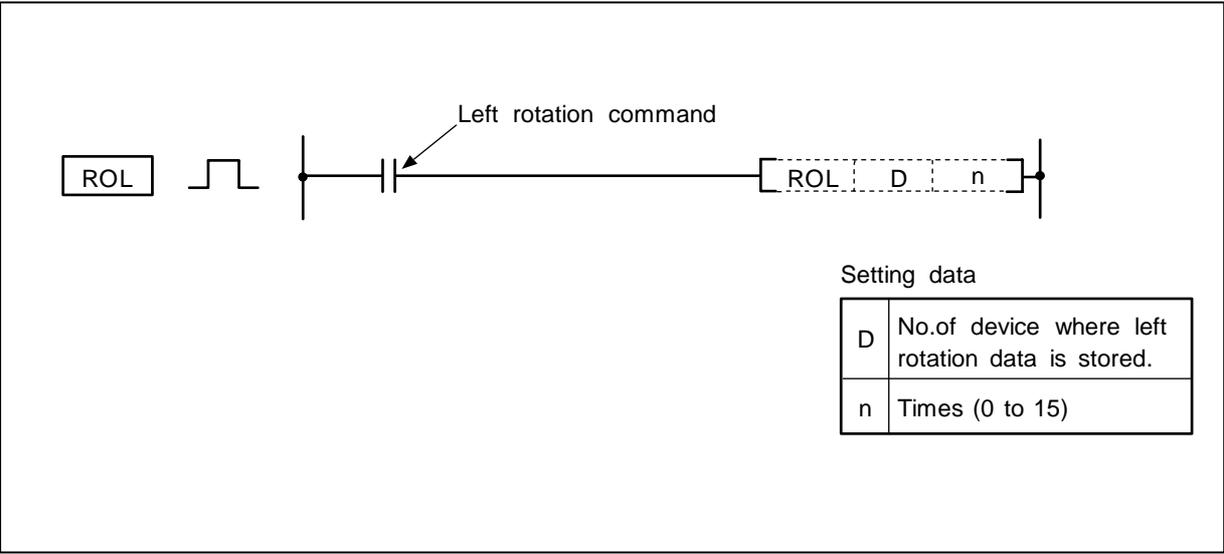
| No. of steps | Command | Device | | |
|--------------|---------|--------|-----|--|
| 10 | LD | XA | | |
| 11 | DMOV | K1 | D10 | |
| 15 | LD | M0 | | |
| 16 | DRCR | D10 | K3 | |
| 19 | | | | |



Right rotation of data using DRCR command

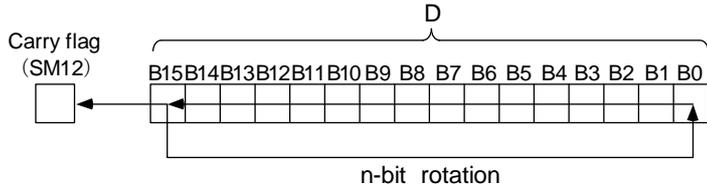
○ ROL ... Left rotation of 16-bit data

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|--|-------------------|--------------|-------|---------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | | | | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | K | H |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| n | | | | | | | | | | | | | | | ○ | ○ | | | |



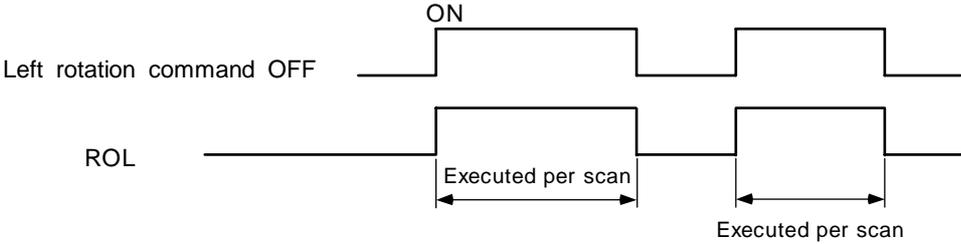
Function

The 16-bit data designated with D is rotated n bits to the left excluding the carry flag. The carry flag must be set to 1 or 0 after executing ROL.



Execution conditions

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



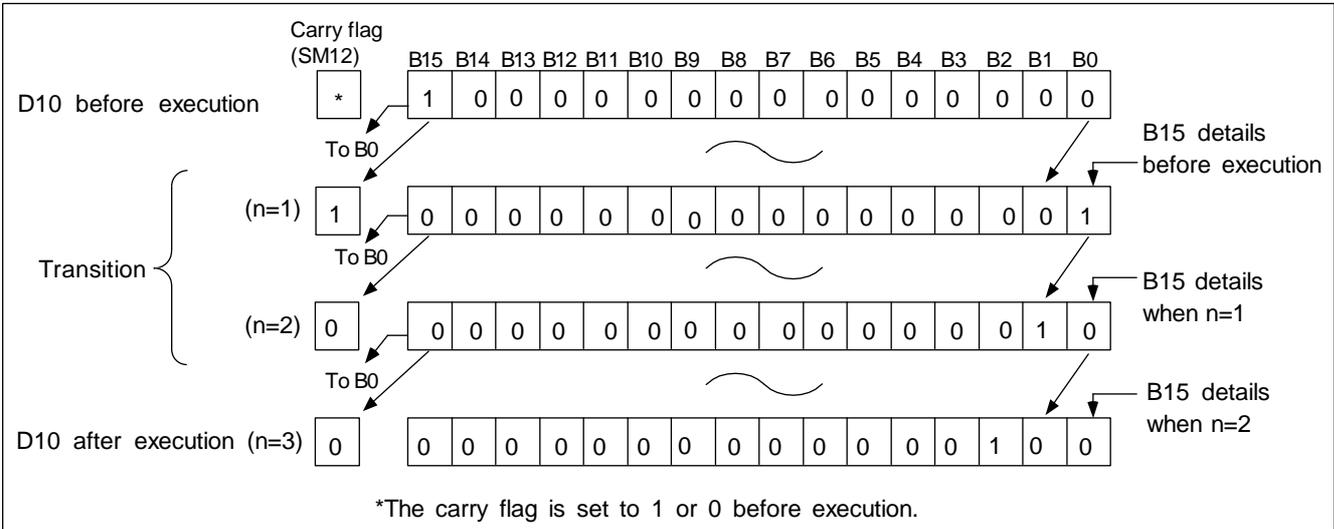
Program example

Program to rotate the D10 details 3 bits to the left when M0 turns ON.



Coding

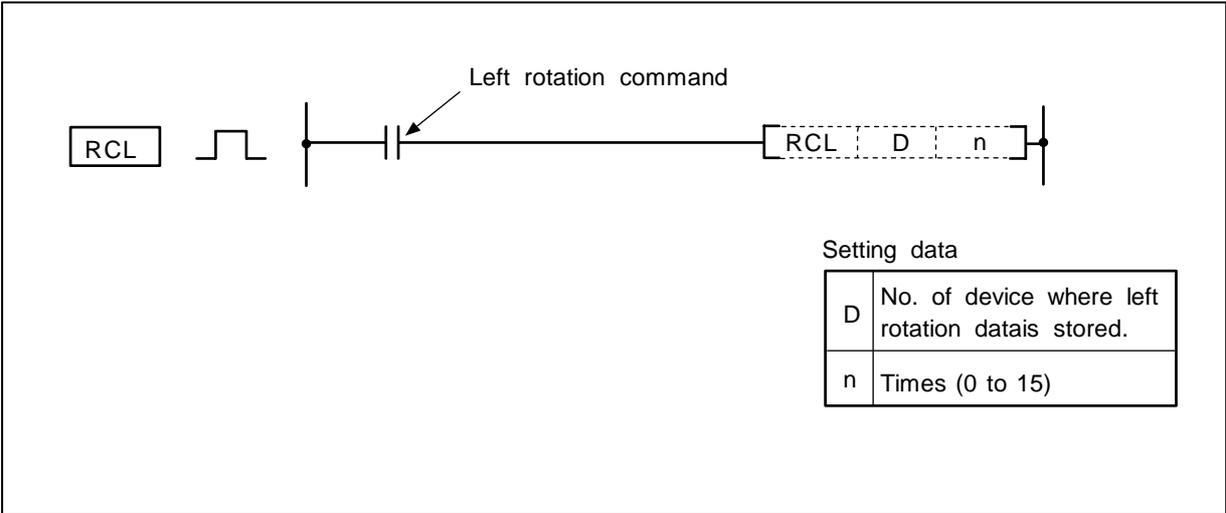
| No. of steps | Command | Device | | |
|--------------|---------|--------|----|--|
| 10 | LD | M0 | | |
| 11 | ROL | D10 | K3 | |
| 14 | | | | |



Left rotation of data using ROL command

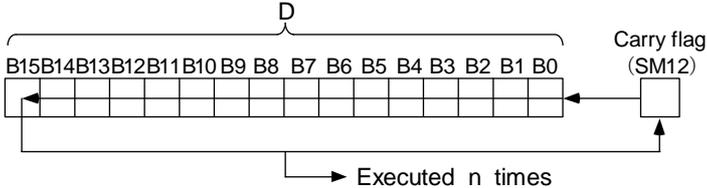
○ RCL ... Left rotation of 16-bit data

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|-------------------|--------------|-------|---------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | | | | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | | | | H | P |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| n | | | | | | | | | | | | | | | ○ | ○ | | | |



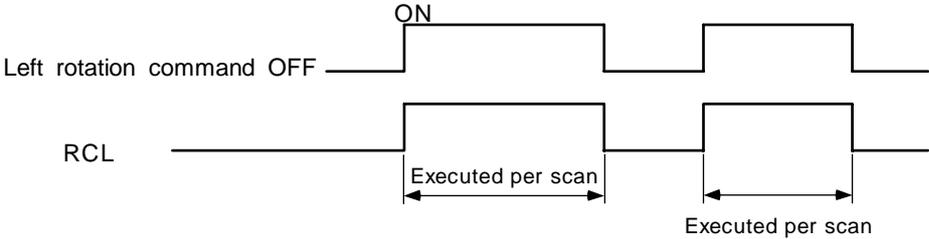
Function

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.



Execution conditions

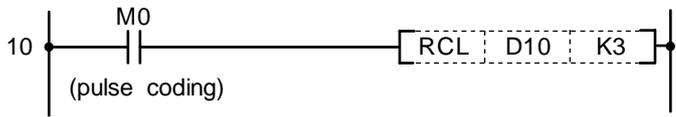
The execution conditions for the RCL command are as shown below.



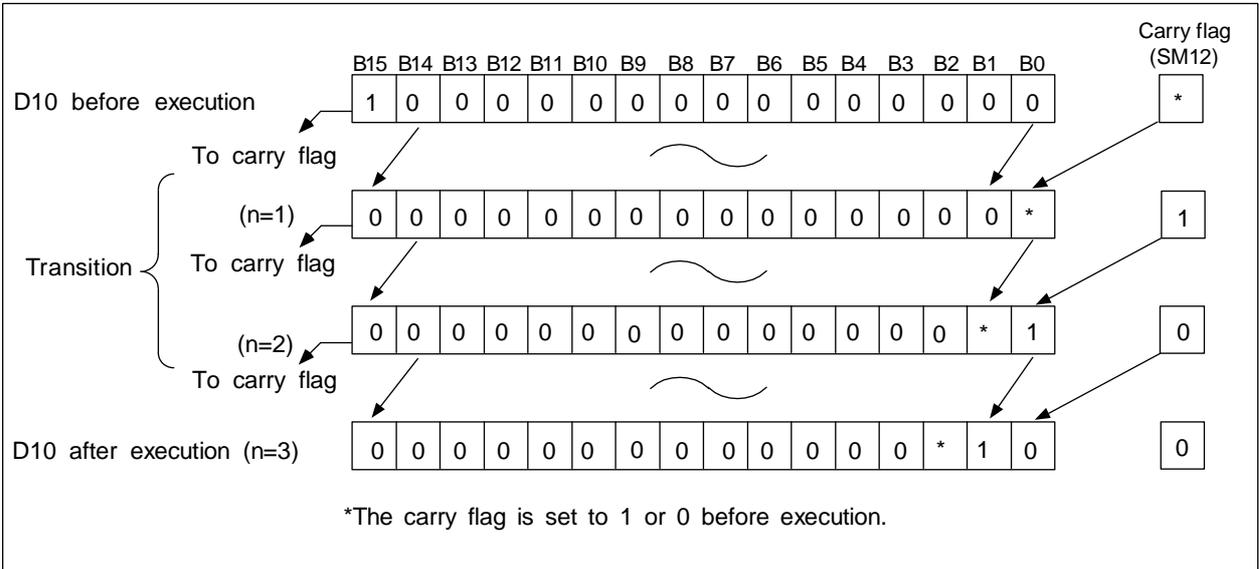
Program example

Program to rotate the D10 details 3 bits to the left when M0 turns ON.

Coding



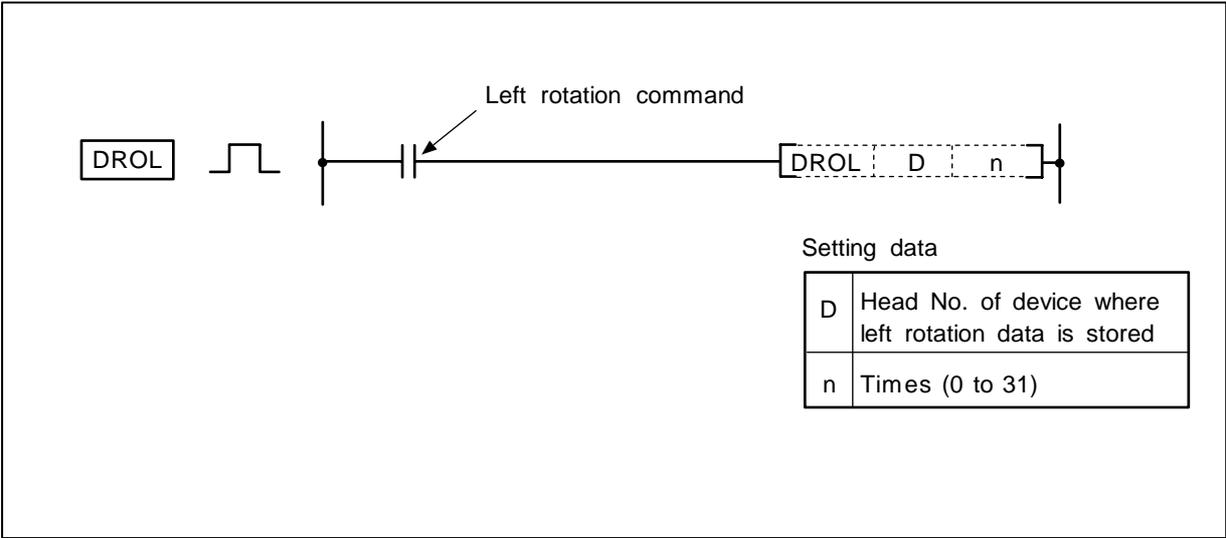
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M0 | | |
| 11 | RCL | D10 | K3 | |
| 14 | | | | |



Left rotation of data using RCL command

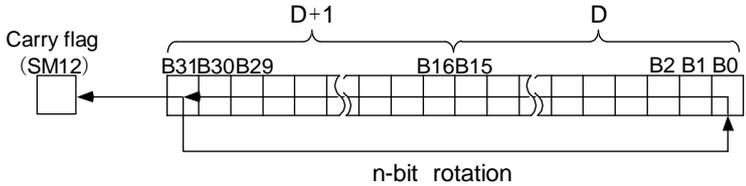
○ DROL ... Left rotation of 32-bit data

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | | | | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---|---------|-------|-------------------|--------------|-------|---|--|--|--|--|--|---|--|
| | Bit device | | | | | | Word device | | | | | | Constant | | Pointer | Level | | | | | | | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | | | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | | | | |
| n | | | | | | | | | | | | | | | ○ | ○ | | | | | | | | | | 3 | |



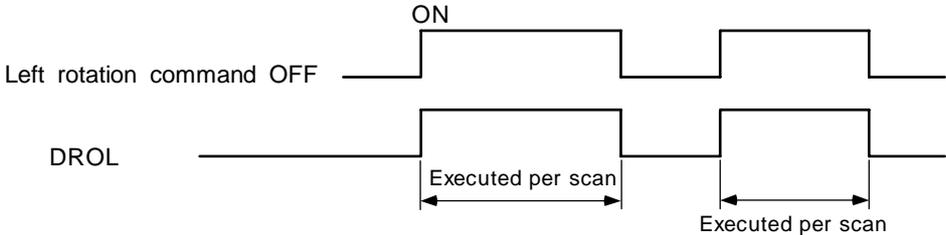
Function

The 32-bit data designated with D is rotated n bits to the left excluding the carry flag.



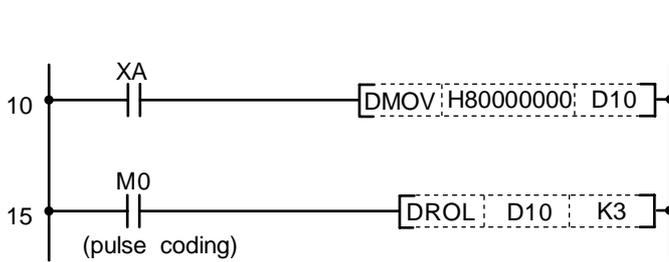
Execution conditions

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



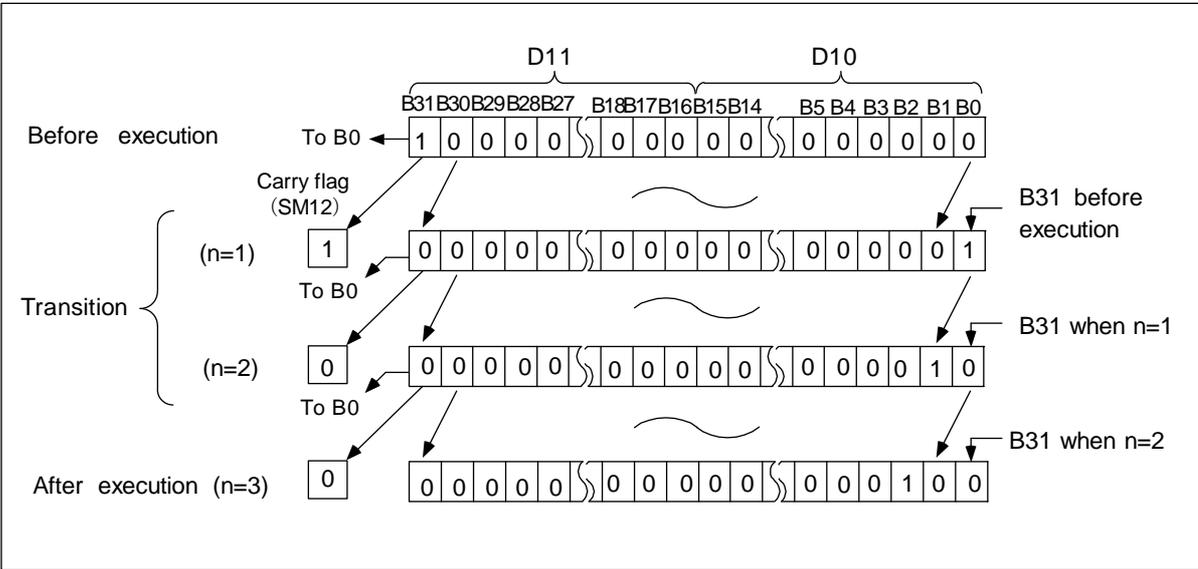
Program example

Program to rotate the D10, 11 details 3 bits to the left when M0 turns ON.



Coding

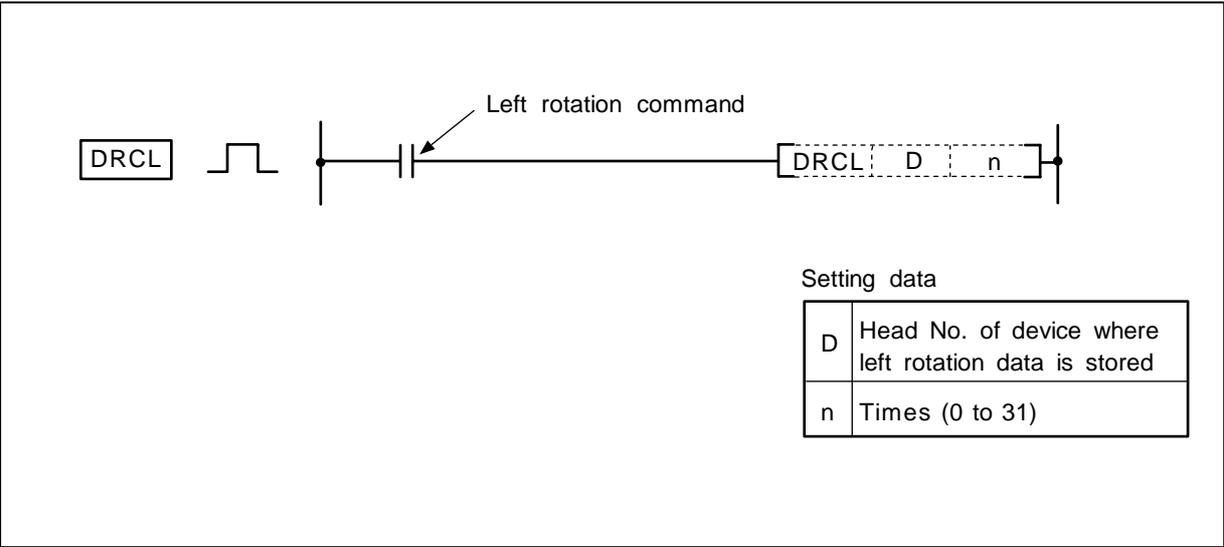
| No. of steps | Com-mand | Device | | |
|--------------|----------|-----------|-----|--|
| 10 | LD | XA | | |
| 11 | DMOV | H80000000 | D10 | |
| 15 | LD | M0 | | |
| 16 | DROL | D10 | K3 | |
| 19 | | | | |



Left rotation of data using DROL command

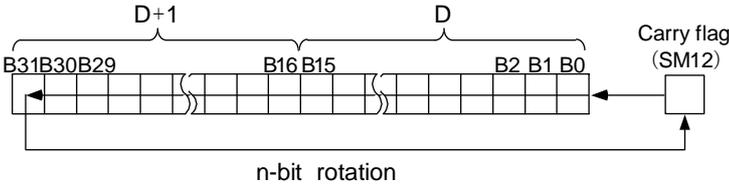
○ DRCL ... Left rotation of 32-bit data

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | | | | | | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---|---------|-------|-------------------|--------------|-------|---|--|--|--|--|--|---|--|
| | Bit device | | | | | | Word device | | | | | | Constant | | Pointer | Level | | | | | | | | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N | | | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | | | | |
| n | | | | | | | | | | | | | | | ○ | ○ | | | | | | | | | | 3 | |



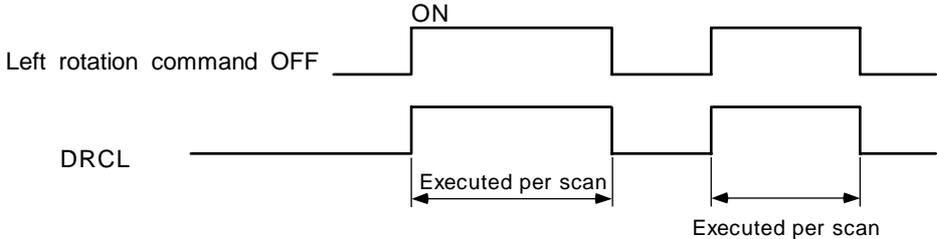
Function

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

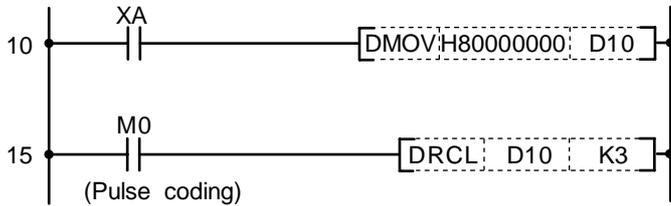
The execution conditions for the DRCL command are as shown below.



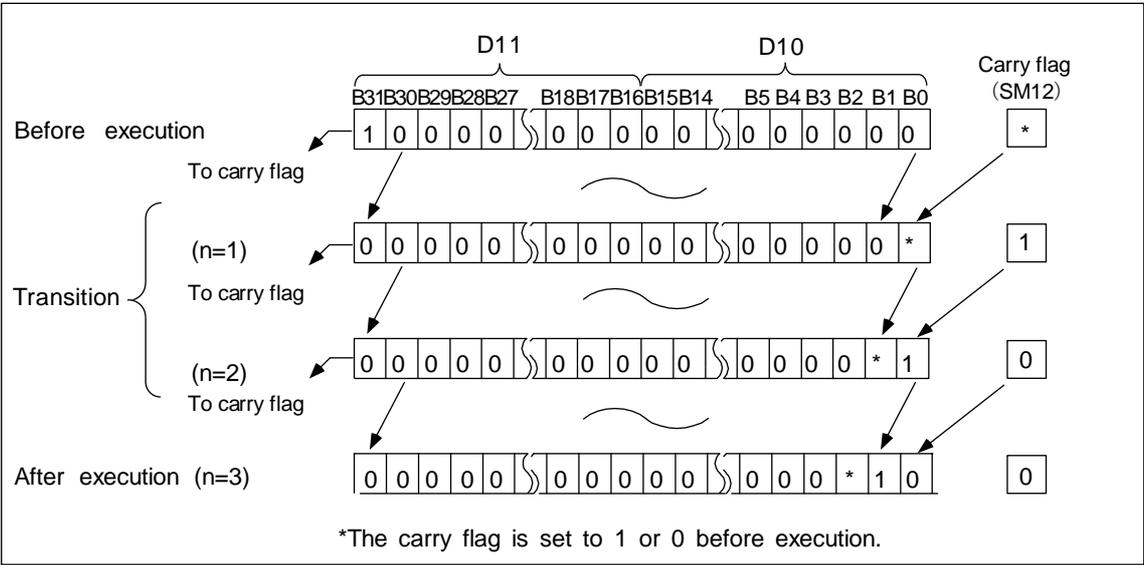
Program example

Program to rotate the D10, 11 details 3 bits to the left when M0 turns ON.

Coding



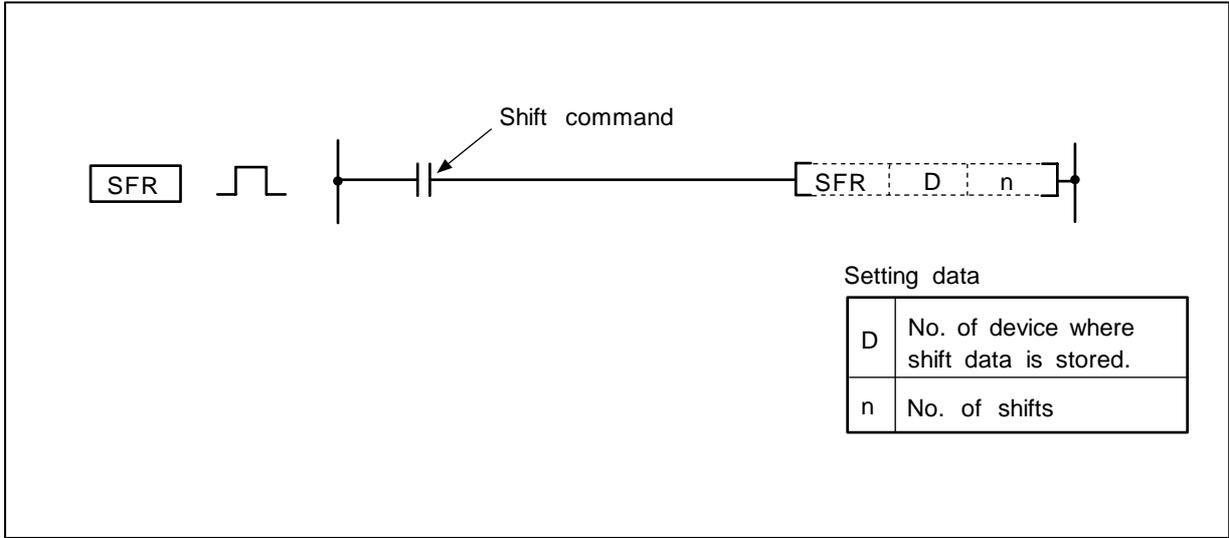
| No. of steps | Com-mand | Device | | |
|--------------|----------|-----------|-----|--|
| 10 | LD | XA | | |
| 11 | DMOV | H80000000 | D10 | |
| 15 | LD | M0 | | |
| 16 | DRCL | D10 | K3 | |
| 19 | | | | |



Left rotation of data using DRCL command

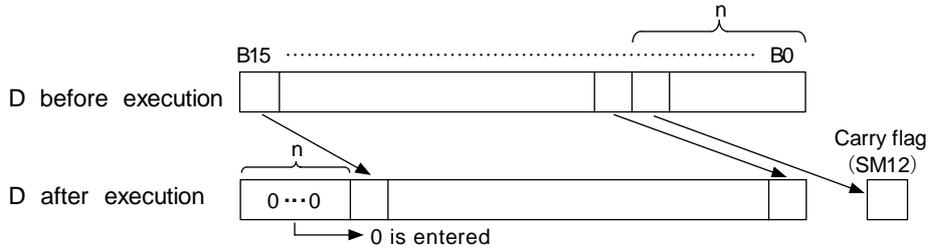
○ SFR ... Right shift of 16-bit data

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|--|-------------------|--------------|-------|---------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | | | | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | | K |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| n | | | | | | | | | | | | | | | ○ | ○ | | | |



Function

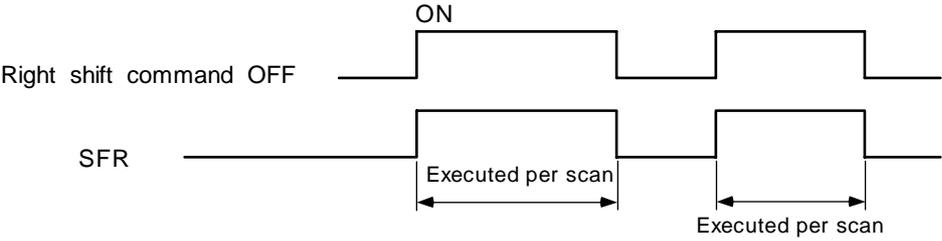
(1) The 16-bit data of the device designated with D is shifted n bits to the right.



- (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.)

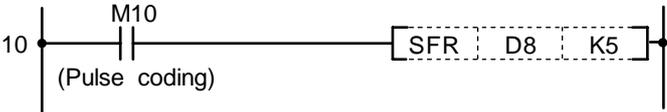
Execution conditions

The execution conditions for SFR are as shown below.



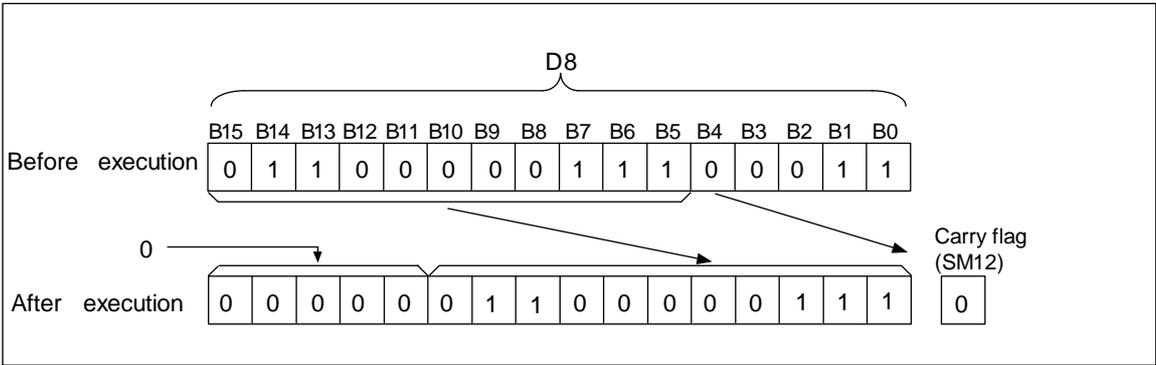
Program example

Program that shifts the details of D8 5 bits to the right when M10 turns ON.



Coding

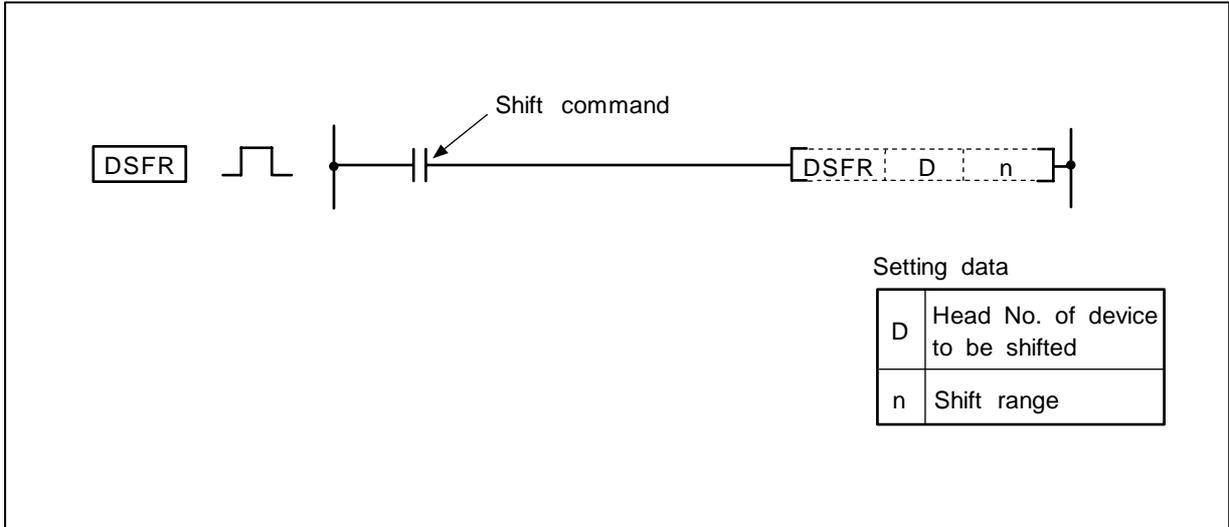
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M10 | | |
| 11 | SFR | D8 | K5 | |
| 14 | | | | |



Right shift of data with SFR command (word device)

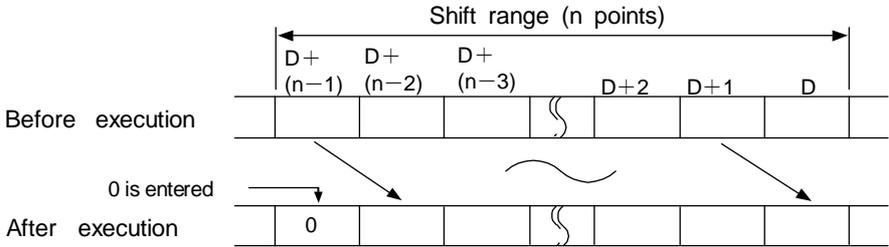
○ DSFR ... Right shift of word device in batch

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---|-------------------|--------------|-------|---------|-------|
| | Bit device | | | | | | Word device | | | | | | Constant | | | | | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | | | | H | P |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| n | | | | | | | | | | | | | | | ○ | ○ | | | |



Function

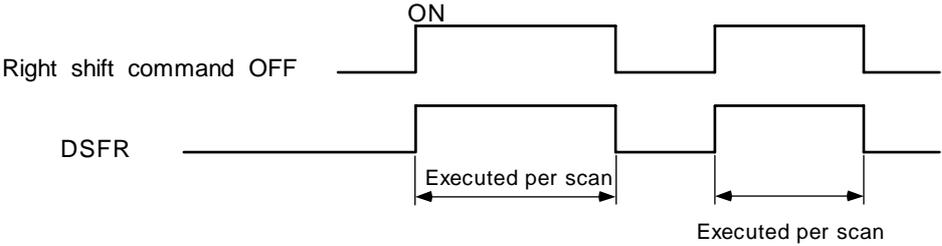
(1) n points starting at the head of the device designated with D are shifted one point to the right.



- (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.)

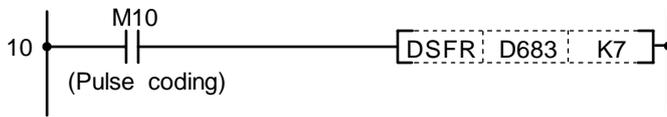
Execution conditions

The execution conditions of DSFR are as shown below.



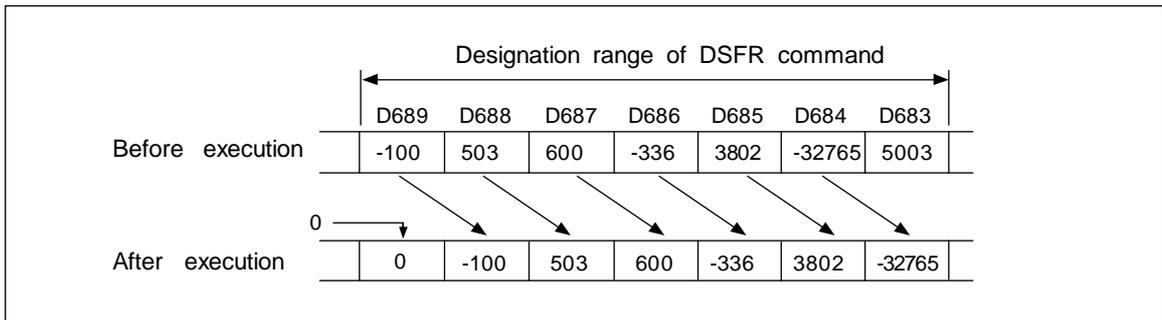
Program example

- (1) Program to shift the details of D683 to 689 to the right when M10 turns ON.



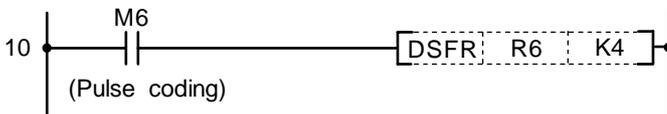
Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M10 | | |
| 11 | DSFR | D683 | K7 | |
| 15 | | | | |



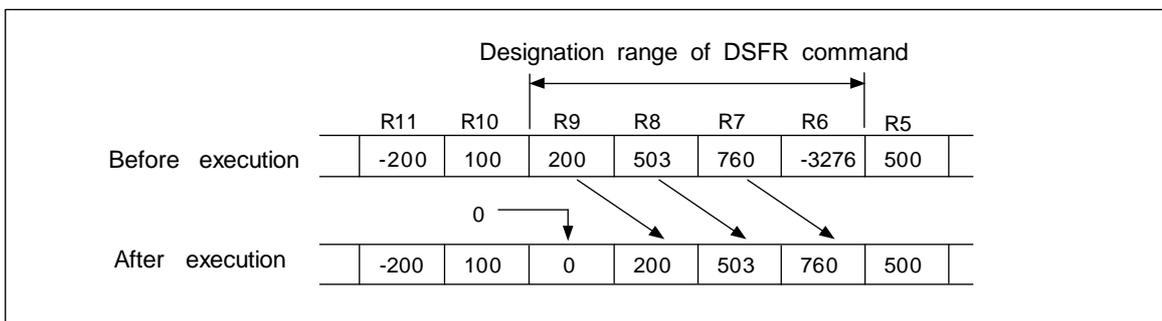
Right shift of data with DSFR command

- (2) Program to shift the details of R6 to 9 to the right when M6 turns ON.



Coding

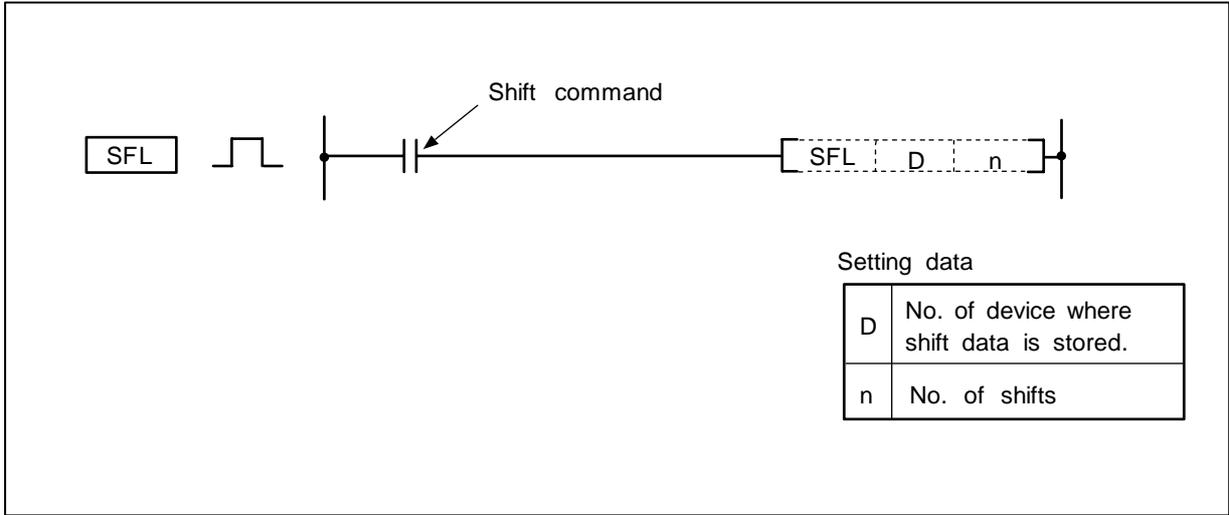
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M6 | | |
| 11 | DSFR | R6 | K4 | |
| 15 | | | | |



Right shift of data with DSFR command

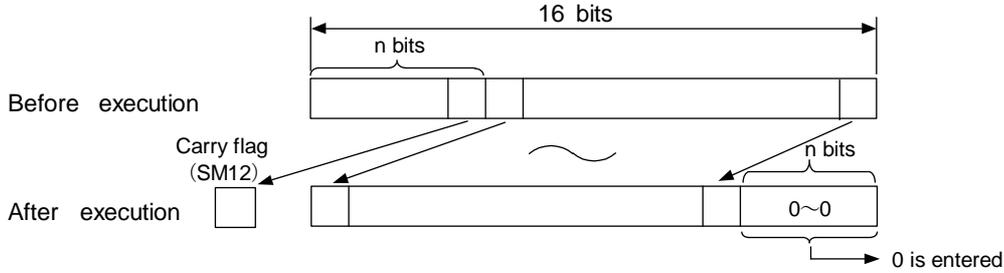
○ SFL ... Left shift of 16-bit data

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---|-------------------|--------------|-------|---------|-------|
| | Bit device | | | | | | Word device | | | | | | Constant | | | | | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | | | | H | P |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| n | | | | | | | | | | | | | | | ○ | ○ | | | |



Function

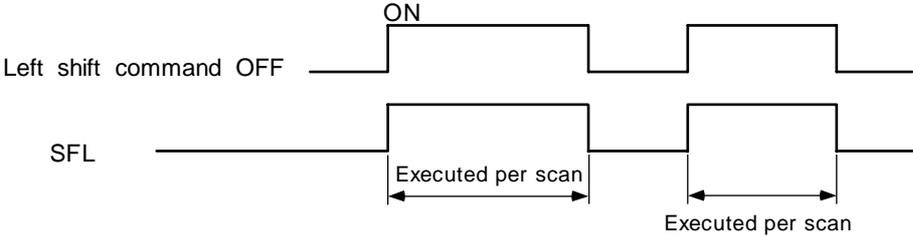
- (1) The 16-bit data of the device designated with D is shifted n bits to the left.
- (2) n bits from the lowest order are set to 0.



- (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 are as shown below.



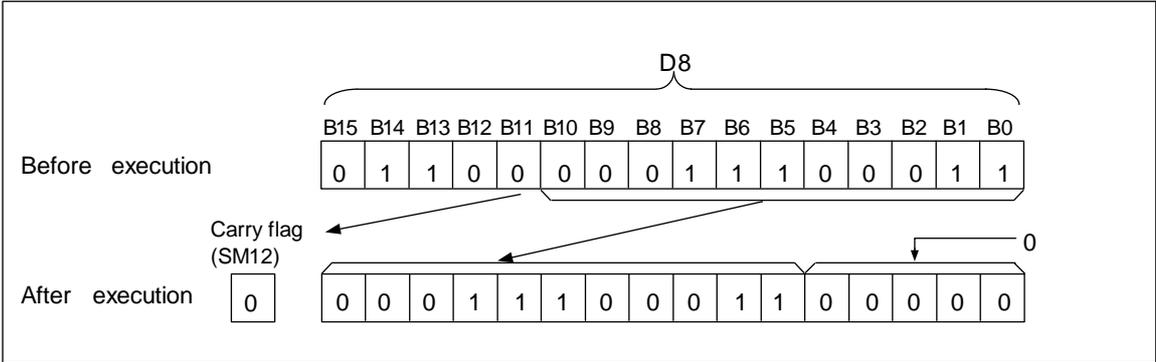
Program example

(1) Program that shifts the details of D8 5 bits to the left when M10 turns ON.



Coding

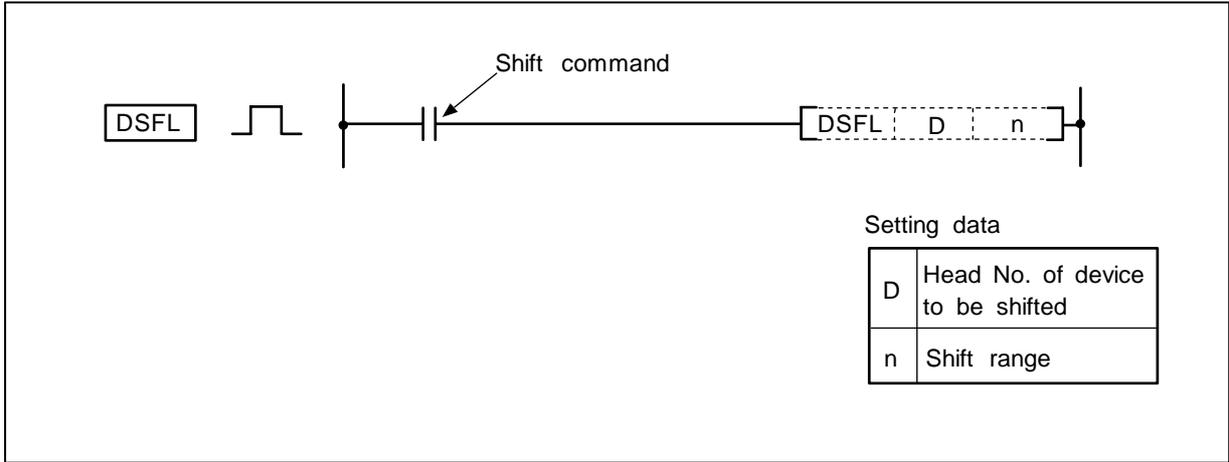
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| | | | | |
| 10 | LD | M10 | | |
| 11 | SFL | D8 | K5 | |
| 14 | | | | |



Left shift of data with SFL command (word device)

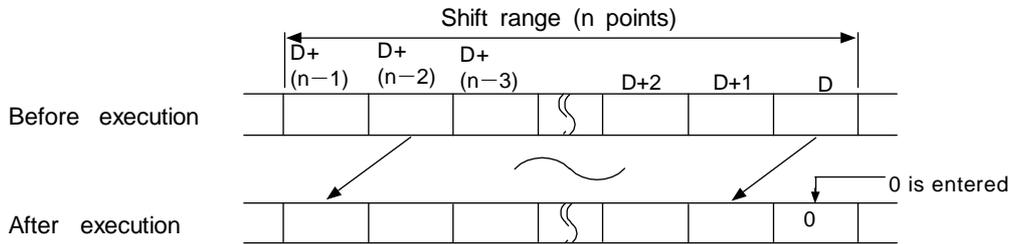
○ DSFL ... Left shift of word device in batch

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|--|-------------------|--------------|-------|---------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | | | | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | | K |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| n | | | | | | | | | | | | | | | ○ | ○ | | | |



Function

(1) n points starting at the head of the device designated with D are shifted one point to the left.

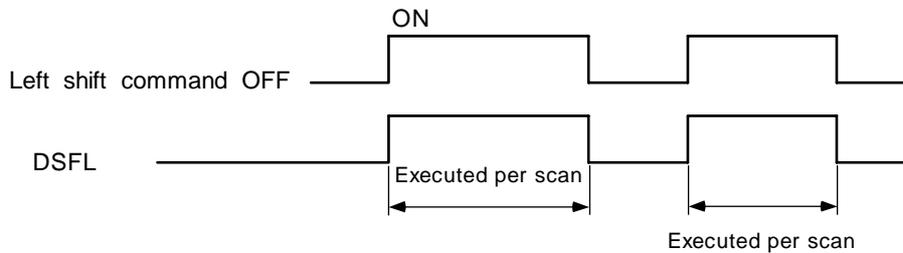


(2) The lowest order device is set to 0.

(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

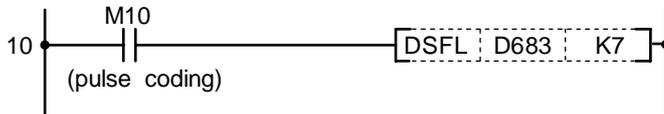
The execution conditions of DSFL are as shown below.



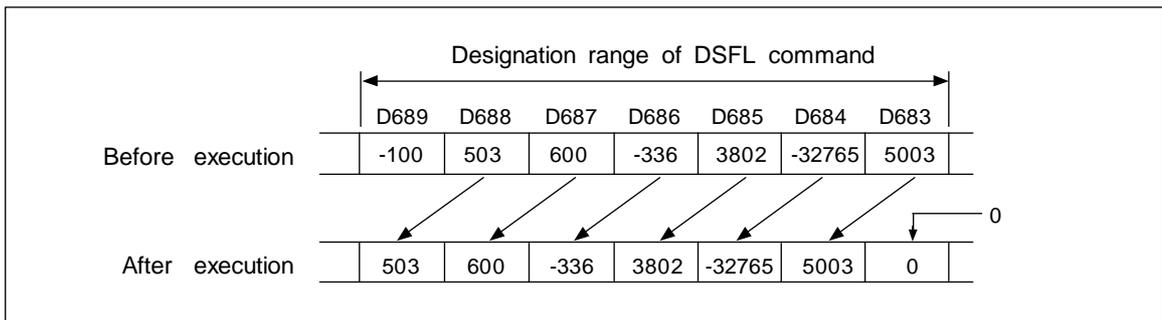
Program example

- (1) Program to shift the details of D683 to 689 to the left when M10 turns ON.

Coding



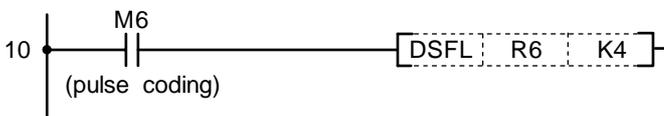
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M10 | | |
| 11 | DSFL | D683 | K7 | |
| 15 | | | | |



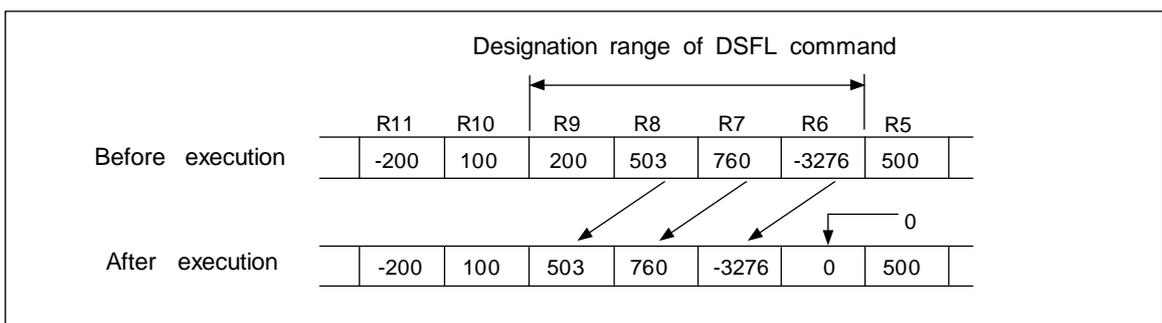
Left shift of data with DSFL command

- (2) Program to shift the details of R6 to 9 to the left when M6 turns ON.

Coding



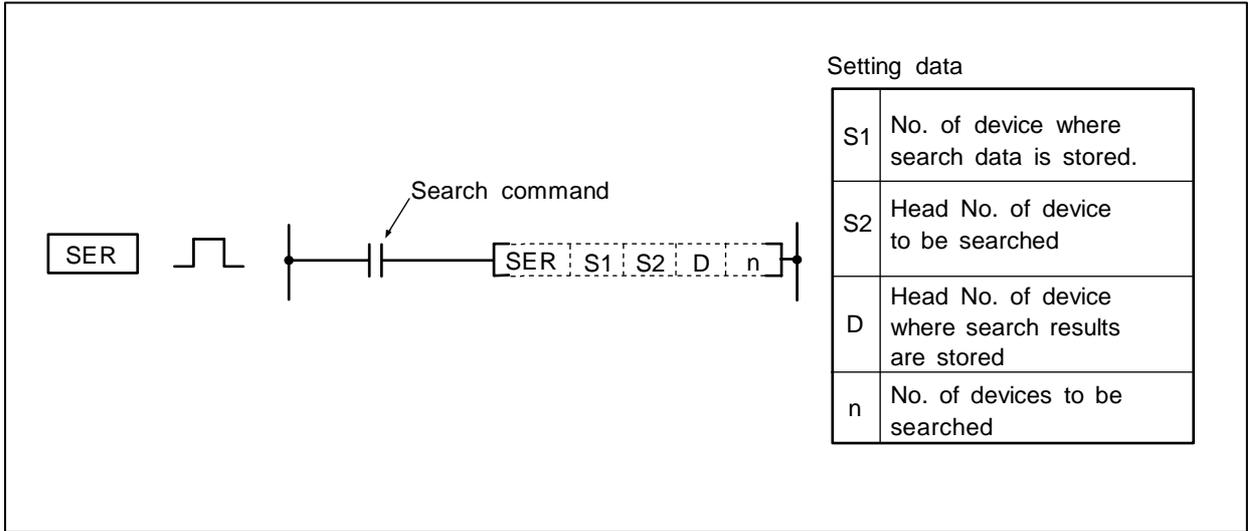
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD | M6 | | |
| 11 | DSFL | R6 | K4 | |
| 15 | | | | |



Left shift of data with DSFL command

○ SER ... Search of 16-bit data

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|--|---------|-------|-------------------|--------------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | K | | | |
| S1 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| S2 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| n | | | | | | | | | | | | | | | ○ | ○ | | | |

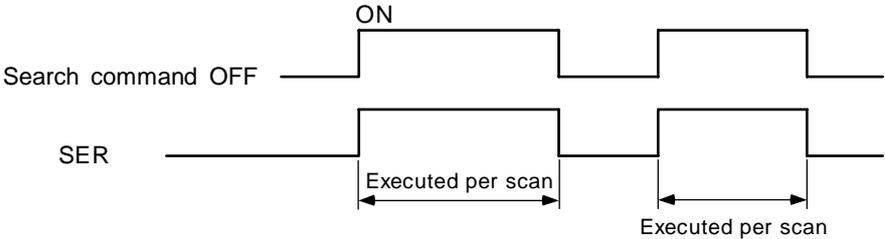


Function

- (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 items 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 are as shown below.



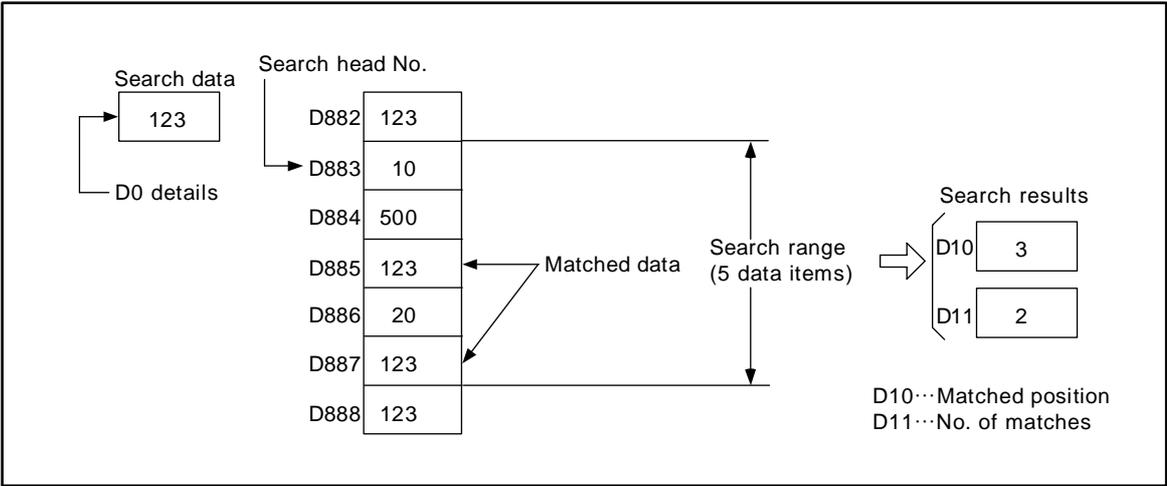
Program example

Program to compare the data in D883 to D887 with 123 when XB turns ON.

Coding



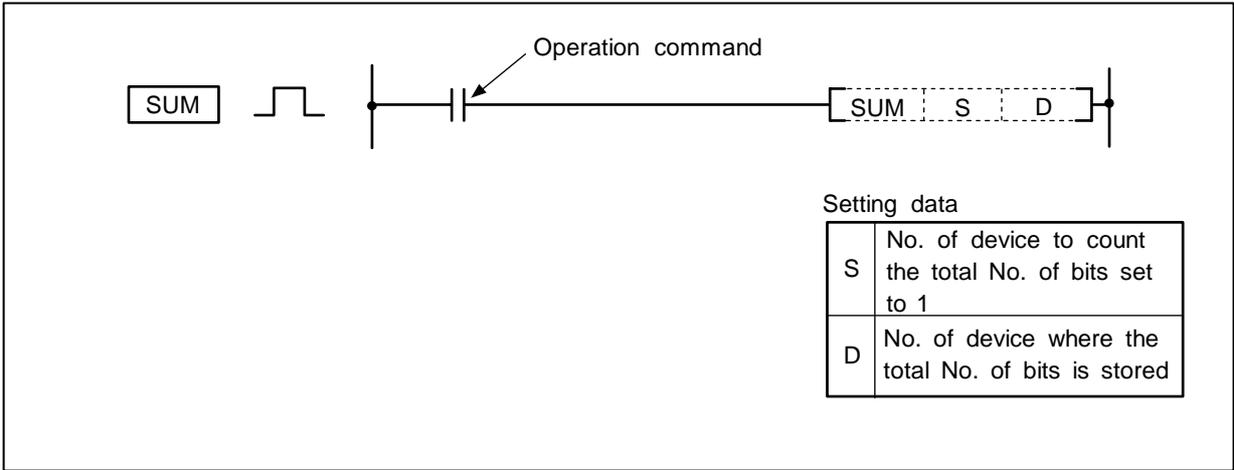
| No. of steps | Com-mand | Device | | | |
|--------------|----------|--------|------|-----|----|
| 10 | LD | XB | | | |
| 11 | SER | D0 | D883 | D10 | K5 |
| 17 | | | | | |



Search of data using SER command

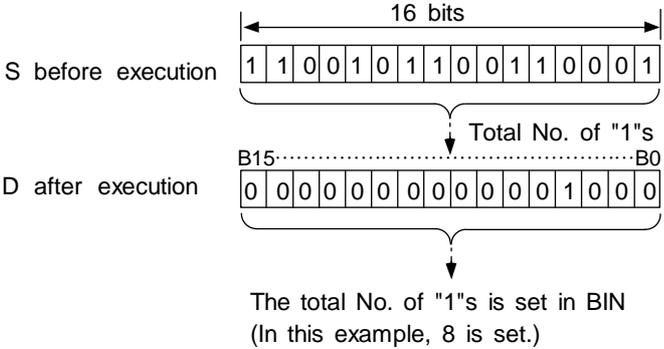
○ SUM ... Count of No. of 16-bit data items set to 1

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---------|-------------------|--------------|-------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | Pointer | | | | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | K |
| S | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | |



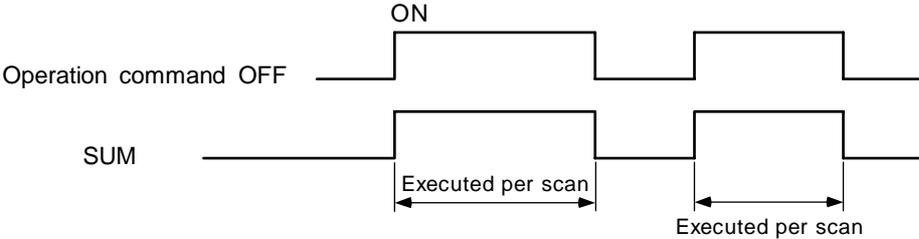
Function

The total No. of bits in the 16-bit data of the device designated with S that are set to "1" is stored in D.



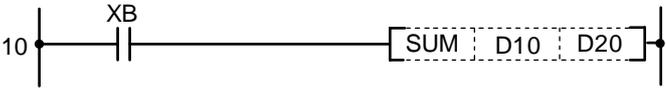
Execution conditions

The execution conditions for SUM are as shown below.



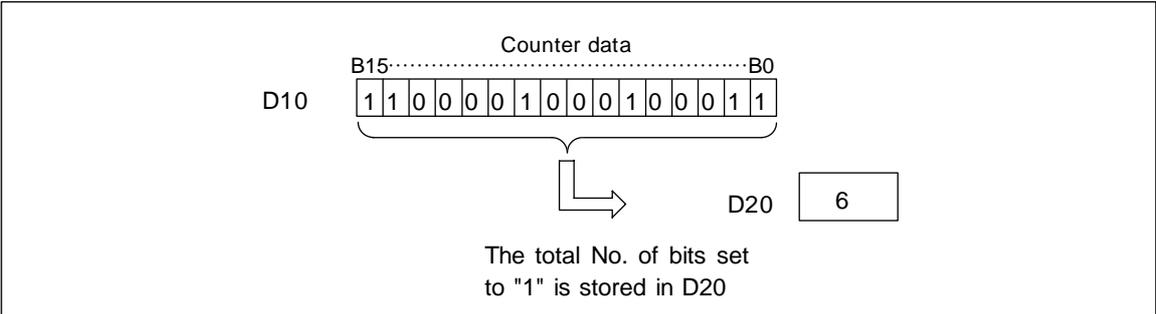
Program example

Program to obtain the No. of D10 data bits that are set to ON (1) when XB turns ON.



Coding

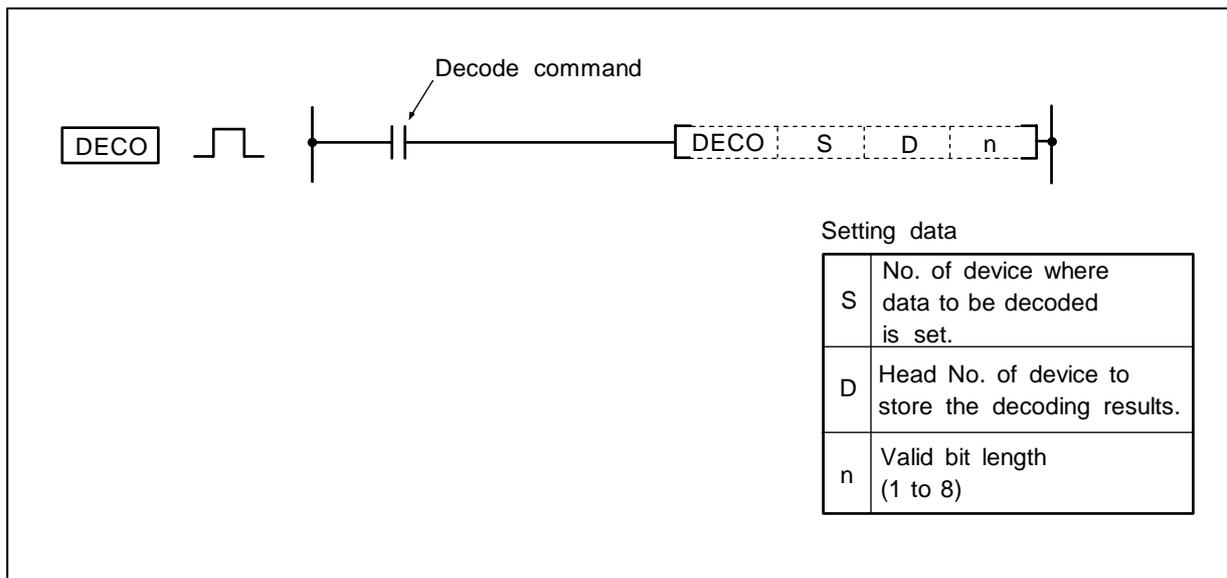
| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-----|--|
| 10 | LD | XB | | |
| 11 | SUM | D10 | D20 | |
| 15 | | | | |



Counting with SUM command

○ DECO ... 8 → 256 bit decoding

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | |
|---|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N |
| S | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |
| n | | | | | | | | | | | | | | | | | ○ | ○ | | |

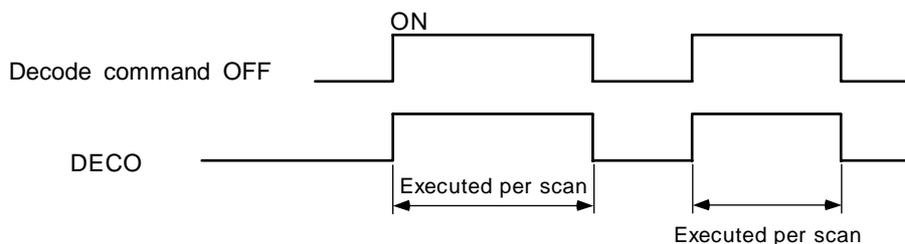


Function

- (1) The low-order n bits of the device designated with S are decoded, and the results are stored in the 2ⁿ bit 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 details of the device designated with D will not change.
- (4) The word device is handled as 16 bits.

Execution conditions

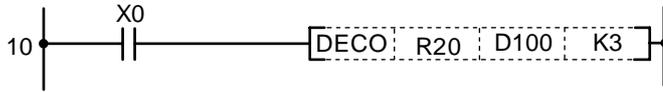
The execution conditions for DECO are as shown below.



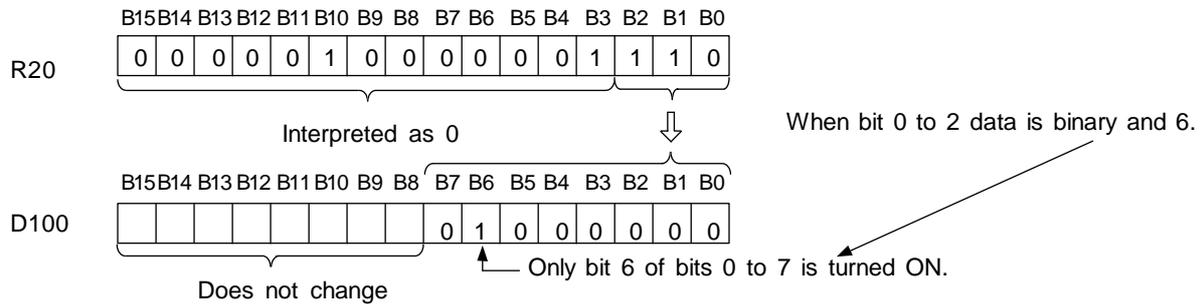
Program example

(1) Program to decode the three bits 0 to 2 of R20, and turn the bits corresponding in D100 ON.

Coding



| No. of steps | Command | Device | | |
|--------------|---------|--------|------|----|
| 10 | LD | X0 | | |
| 11 | DECO | R20 | D100 | K3 |
| 16 | | | | |

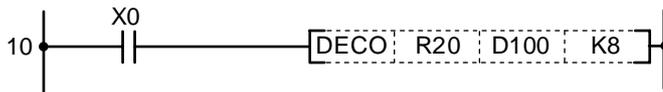


(Note 1) The D100 bit 0 turns ON when the B0 to B2 of R20 are 0.

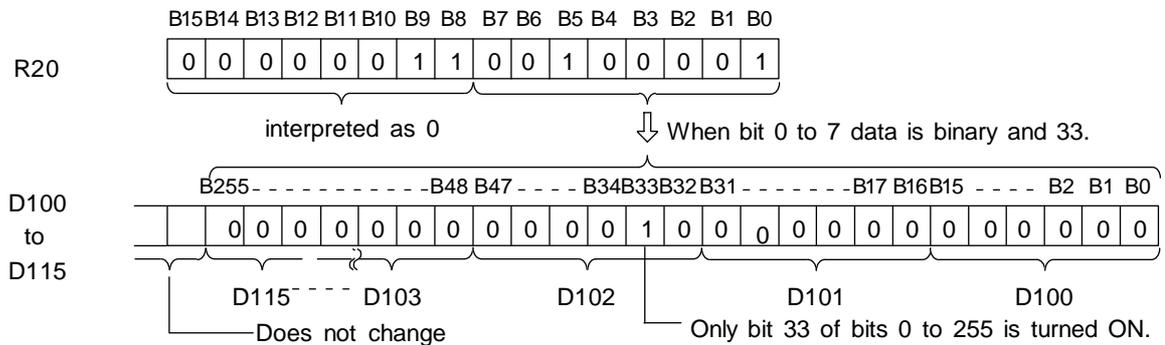
(Note 2) The D100 details remain the same even if X0 turns OFF.

(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

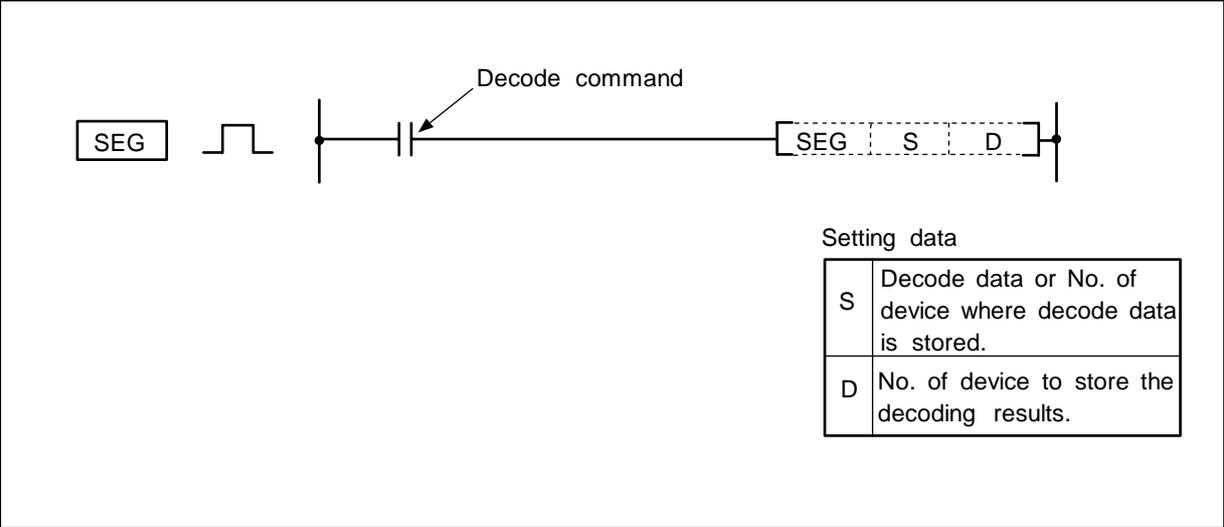


| No. of steps | Command | Device | | |
|--------------|---------|--------|------|----|
| 10 | LD | X0 | | |
| 11 | DECO | R20 | D100 | K8 |
| 16 | | | | |



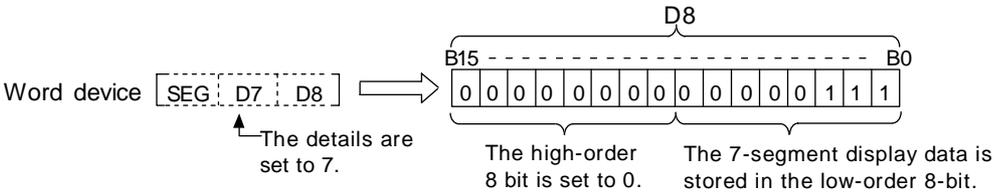
○ SEG ... Decoding to 7-segment display data

| | Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|--|-------------------|--------------|-------|---------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | | | | Pointer | Level |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | | | | K |
| S | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |



Function

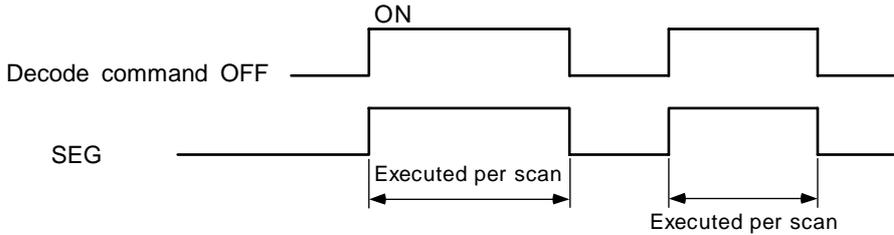
- (1) The 0 to F data designated with the low-order 4-bit in S is decoded in the 7-segment display data and stored in D.



- (2) Refer to the following page for the 7-segment display.

Execution conditions

The execution conditions for SEG are as follow.



7-segment decode table

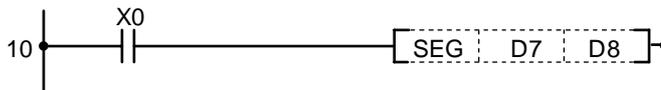
| S | | Configuration of 7-segment | D | | | | | | | Display data | |
|--------------|-------------|----------------------------|----|----|----|----|----|----|----|--------------|----|
| Hexa-decimal | Bit pattern | | B7 | B6 | B5 | B4 | B3 | B2 | B1 | | B0 |
| 0 | 0000 | | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 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.

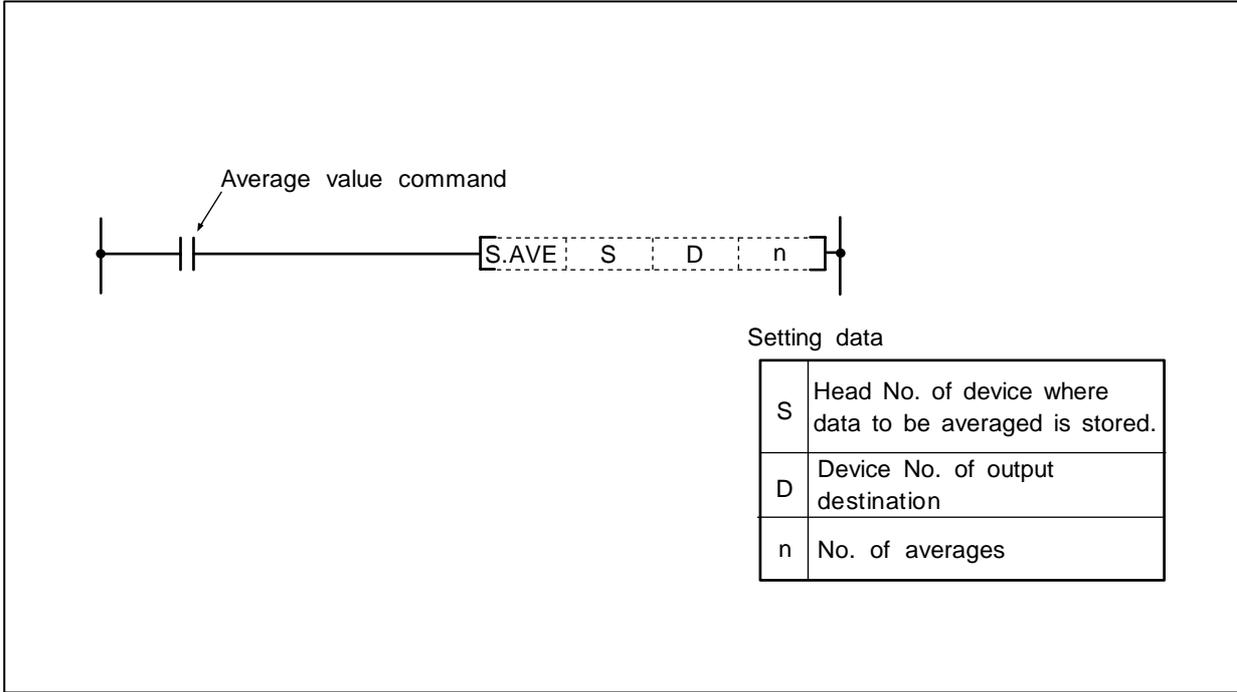
Coding



| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| | | | | |
| 10 | LD | X0 | | |
| 11 | SEG | D7 | D8 | |
| 14 | | | | |

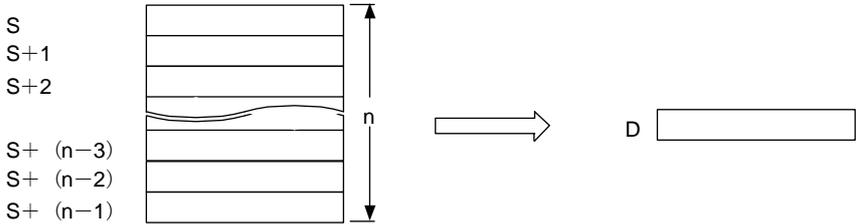
○ S.AVE ... Calculation of average value

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index | |
|----------|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|---|---------|-------|-------------------|--------------|-------|---|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | K | H | P | | | | N |
| S | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |
| D | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | | |
| n | | | | | | | | | | | | | | ○ | ○ | | | | | |



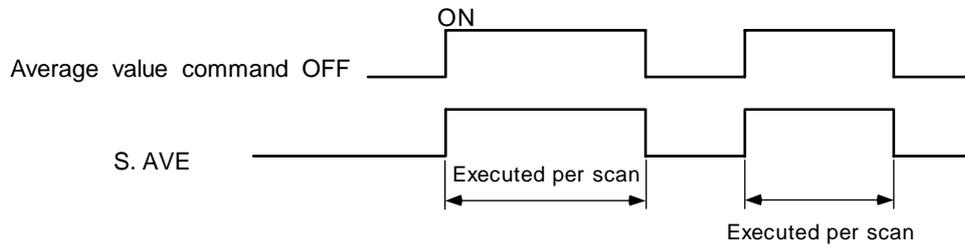
Function

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.



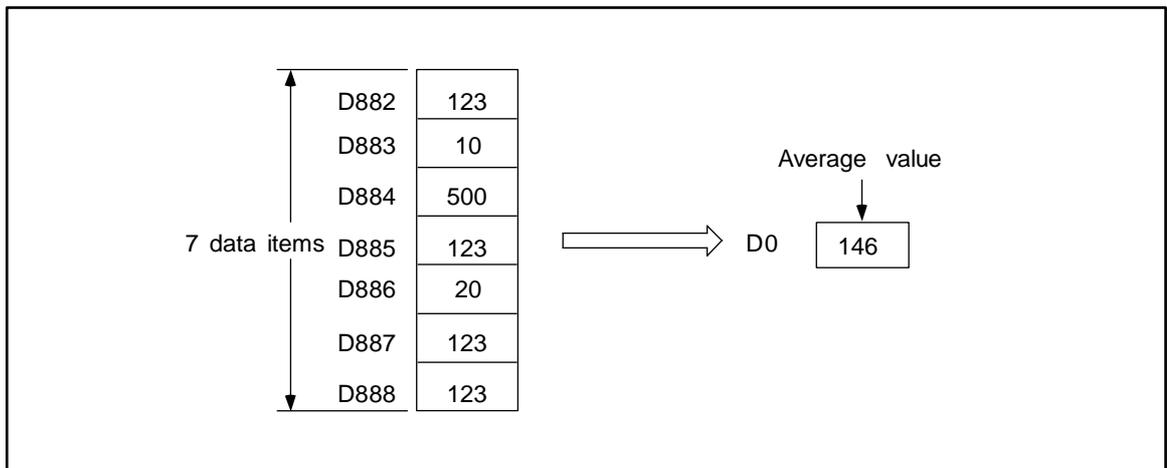
Program example

- (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 | | | | |

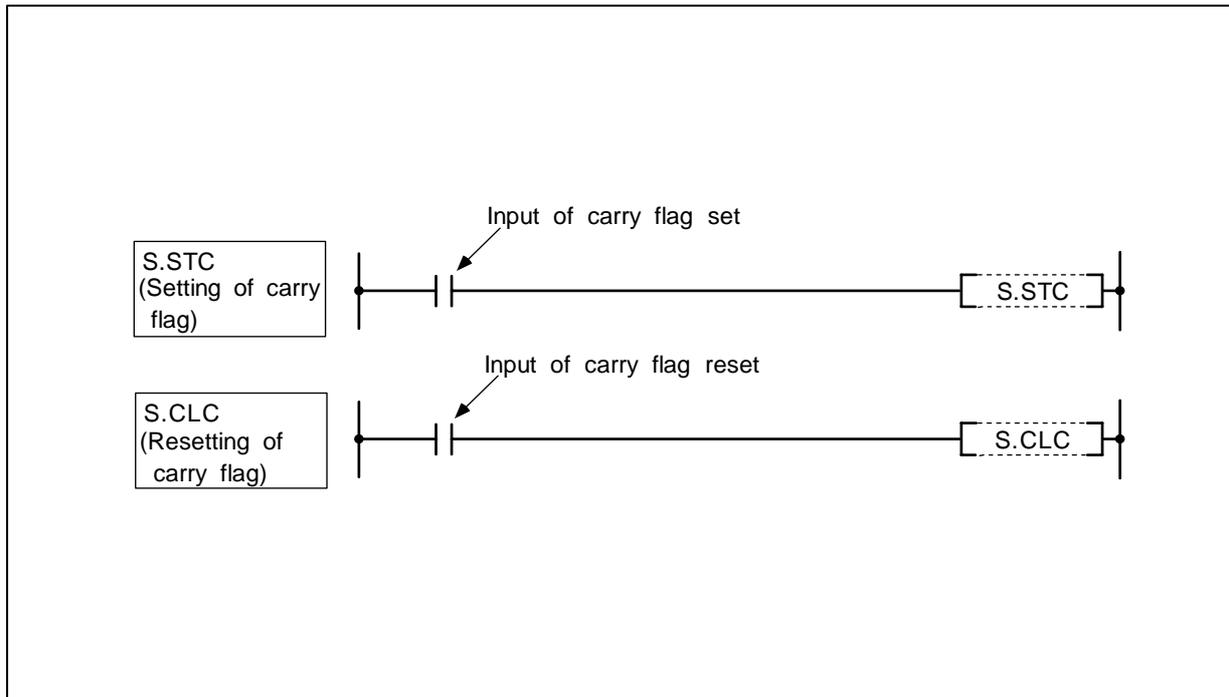


Averaging of data with S.AVE command

(Note) Fractional values are omitted.

○ S.STC, S.CLC ... Setting/resetting of carry flag

| Usable device | | | | | | | | | | | | | | Digit designation | No. of steps | Index | | |
|---------------|---|---|---|----|---|-------------|---|---|---|---|--|----------|---------|-------------------|--------------|-------|-------|--|
| Bit device | | | | | | Word device | | | | | | Constant | Pointer | | | | Level | |
| X | Y | M | L | SM | F | T | C | D | R | Z | | K | H | P | N | | | |
| | | | | | | | | | | | | | | | | | 1 | |



Function

S.STC

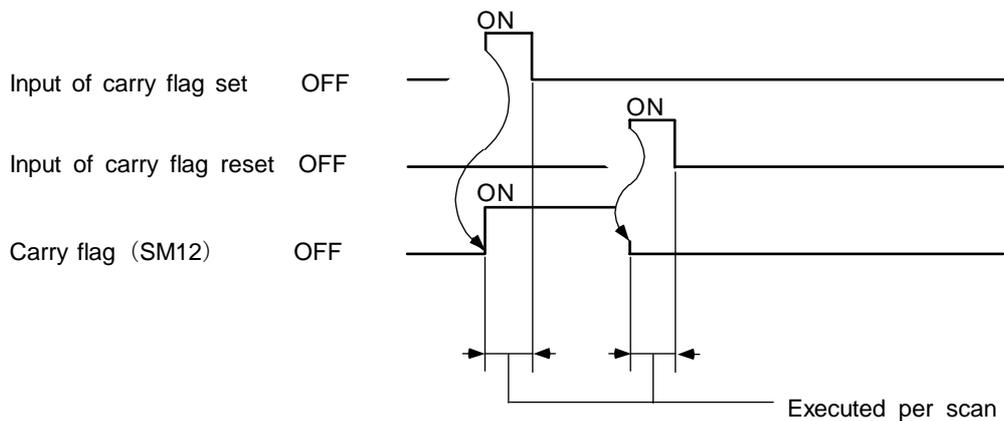
(1) The carry flag contact (SM12) is set (ON).

S.CLC

(1) The carry flag contact (SM12) is reset (OFF).

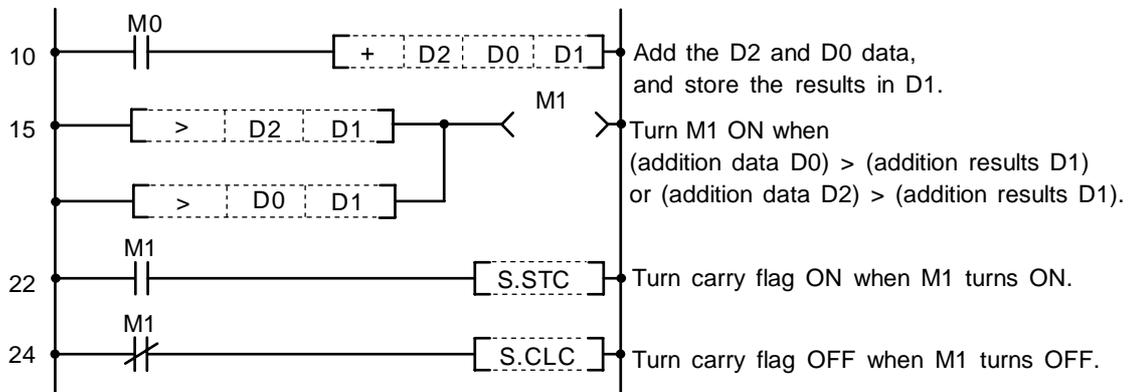
Execution conditions

The execution conditions for S.STC and S.CLC are as shown below.



Program example

In this program, the positive value data D2 and D0 are added upon M0's turning ON, and the carry flag (SM12) is turned ON if the results exceed 32767. The carry flag is turned OFF if the results are 32767 or less.

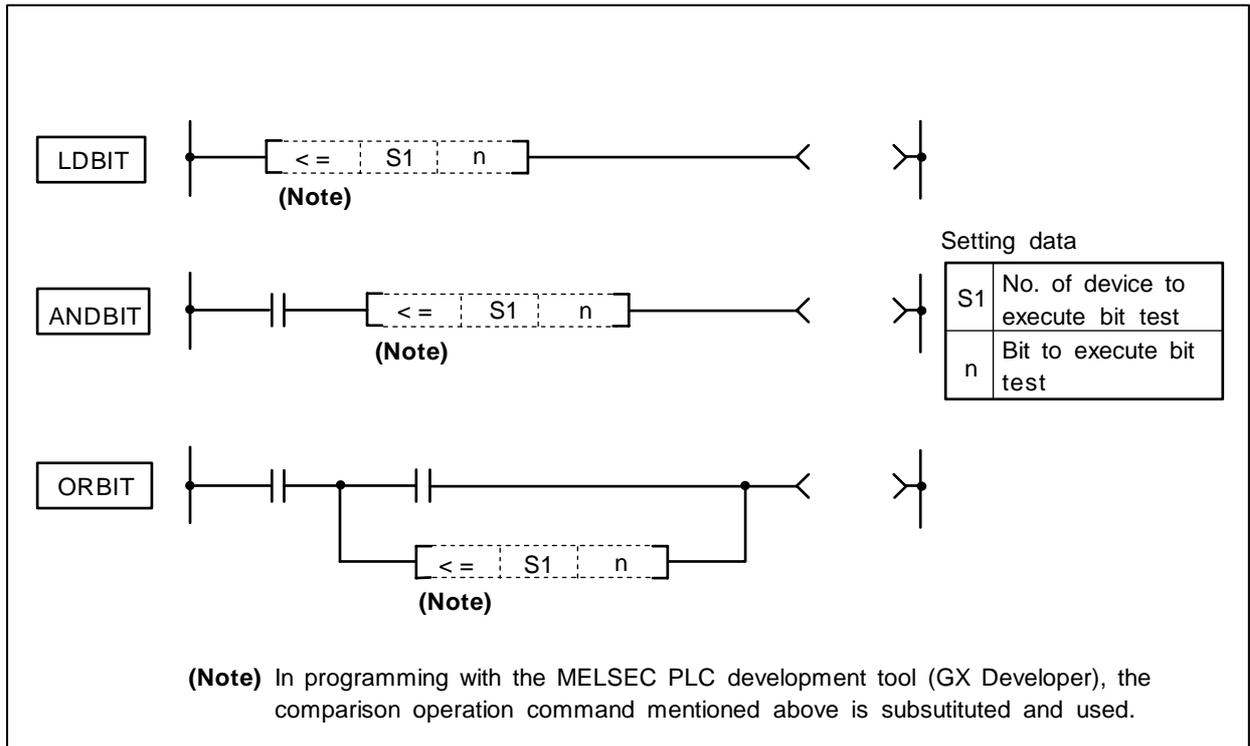


Coding

| 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 | LD1 | M1 | | |
| 25 | S.CLC | | | |
| 26 | | | | |

○ LDBIT, ANDBIT, ORBIT ... Bit test of "A" contact handling

| | Usable device | | | | | | | | | | | | | | | | Digit designation | No. of steps | Index |
|----|---------------|---|---|---|----|---|-------------|---|---|---|---|--|-----------|--|---------|-------|-------------------|--------------|-------|
| | Bit device | | | | | | Word device | | | | | | Con-stant | | Pointer | Level | | | |
| | X | Y | M | L | SM | F | T | C | D | R | Z | | | | | K | | | |
| S1 | | | | | | | ○ | ○ | ○ | ○ | | | | | | | | | |
| n | | | | | | | | | | | | | | | ○ | ○ | | | |



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 |

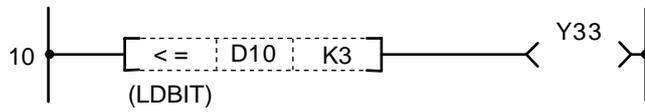
Execution conditions

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 |

Program example

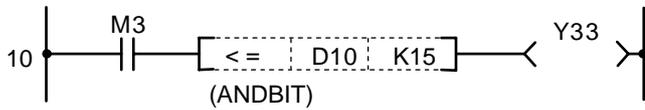
(1) Program to test bit 3 of D10.



Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|----|--|
| 10 | LD<= | D10 | K3 | |
| 12 | OUT | Y33 | | |
| 13 | | | | |

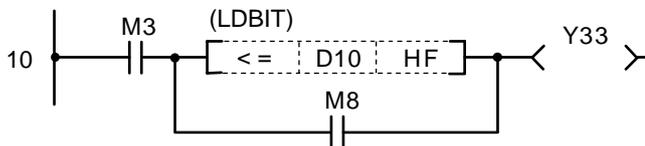
(2) Program to test bit 15 of D10.



Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-----|--|
| 10 | LD | M3 | | |
| 11 | AND<= | D10 | K15 | |
| 13 | OUT | Y33 | | |
| 14 | | | | |

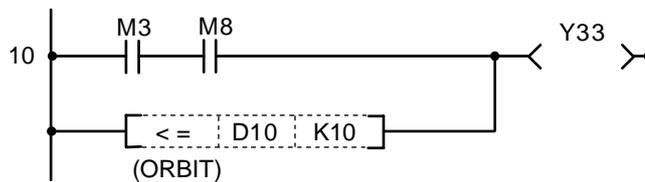
(3) Program to test bit 15 of D10.



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.

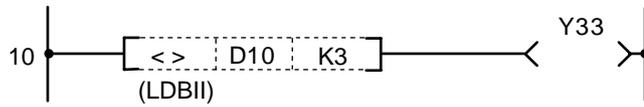


Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-----|--|
| 10 | LD | M3 | | |
| 11 | AND | M8 | | |
| 12 | OR<= | D10 | K10 | |
| 14 | OUT | Y33 | | |
| 15 | | | | |

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 | | | | |

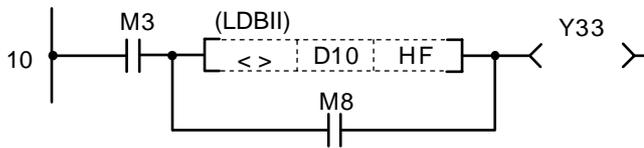
(2) Program to test bit 15 of D10.



Coding

| No. of steps | Com-mand | Device | | |
|--------------|----------|--------|-----|--|
| 10 | LD | M3 | | |
| 11 | AND<> | D10 | K15 | |
| 13 | OUT | Y33 | | |
| 14 | | | | |

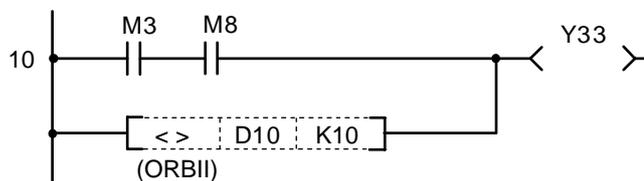
(3) Program to test bit 15 of D10.



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 | | | | |

9. Exclusive Commands

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, the M300 series provides a number of exclusive commands which are explained below.

Examples of exclusive commands:

- ATC dedicated command (ATC)
- Rotary body control command (ROT)
- Tool life management exclusive command (TSRH)
- DDB (direct data bus) asynchronous
- External search synchronous
- Chopping
- CC-Link

9.1 ATC Exclusive Command

9.1.1 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.

9.1.2 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)

9.1.3 Explanation of Terminology

(1) Pointer

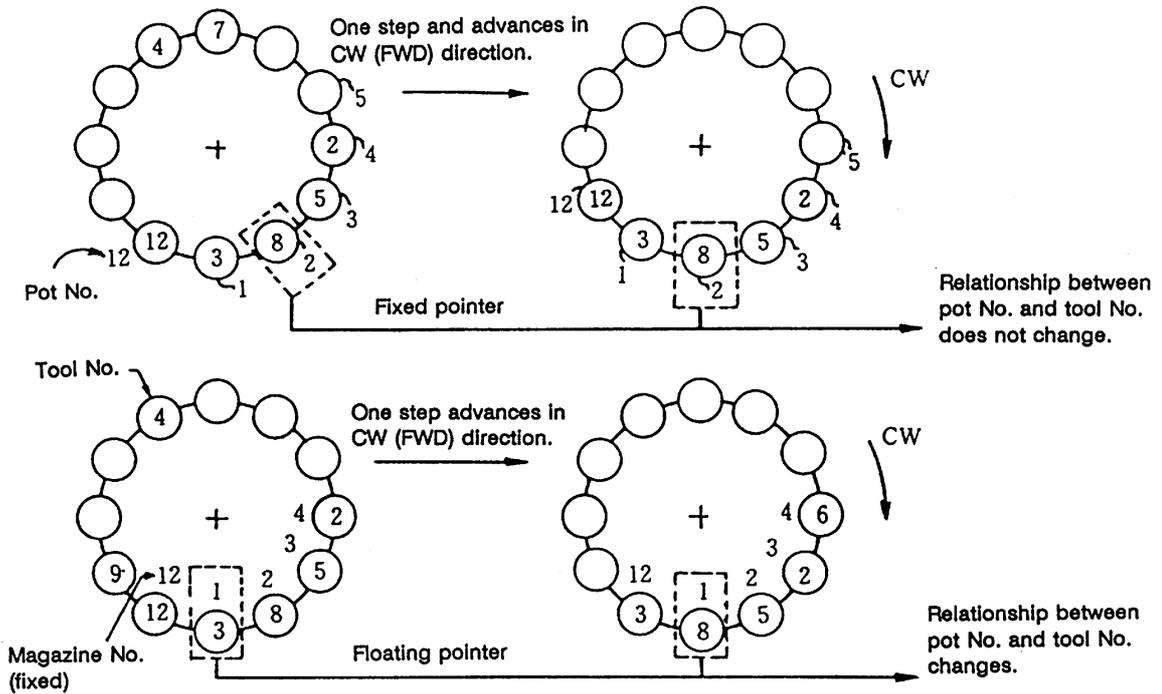
This points out the position where the magazine is indexed. When a tool table in which tool No. are previously recorded is used, the tool table does not rotate with rotation of the magazine and the pointer serves as "ring counter" for control of magazine position.

(2) Fixed pointer

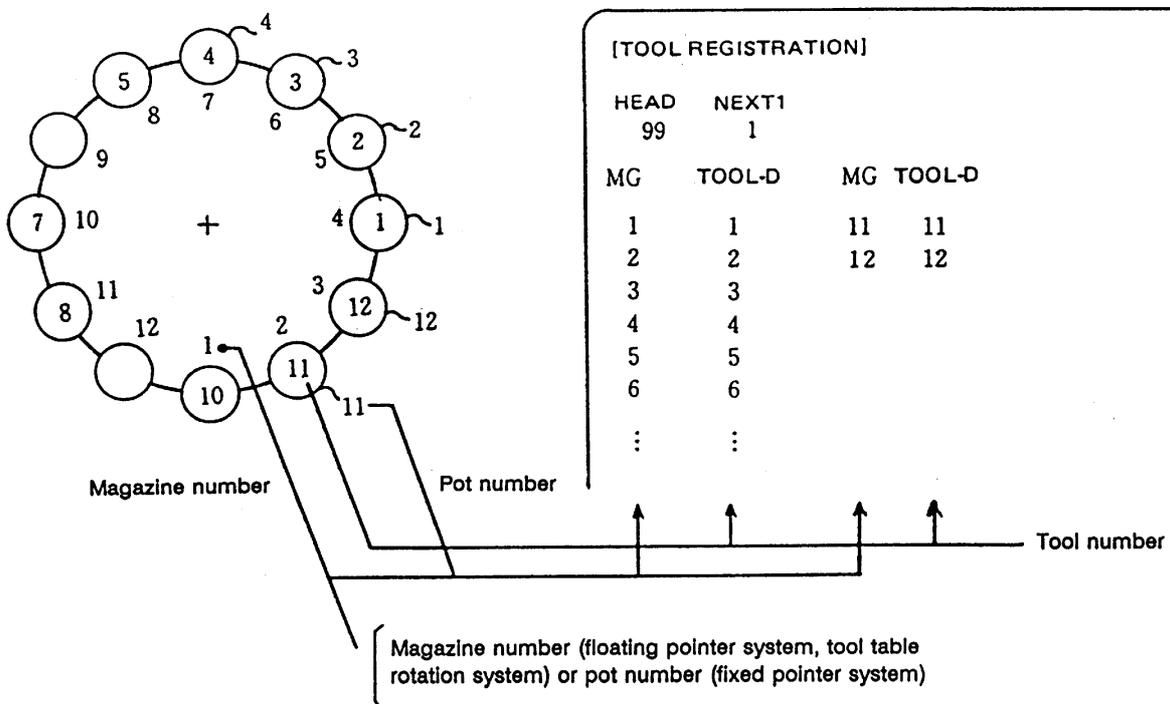
This is the type with tool pots numbered and the relationship between tool pot and tool No. is fixed if the magazine is rotated. When the tool table is rotated, fixed pointer does not functionally differ from "floating pointer".

(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.



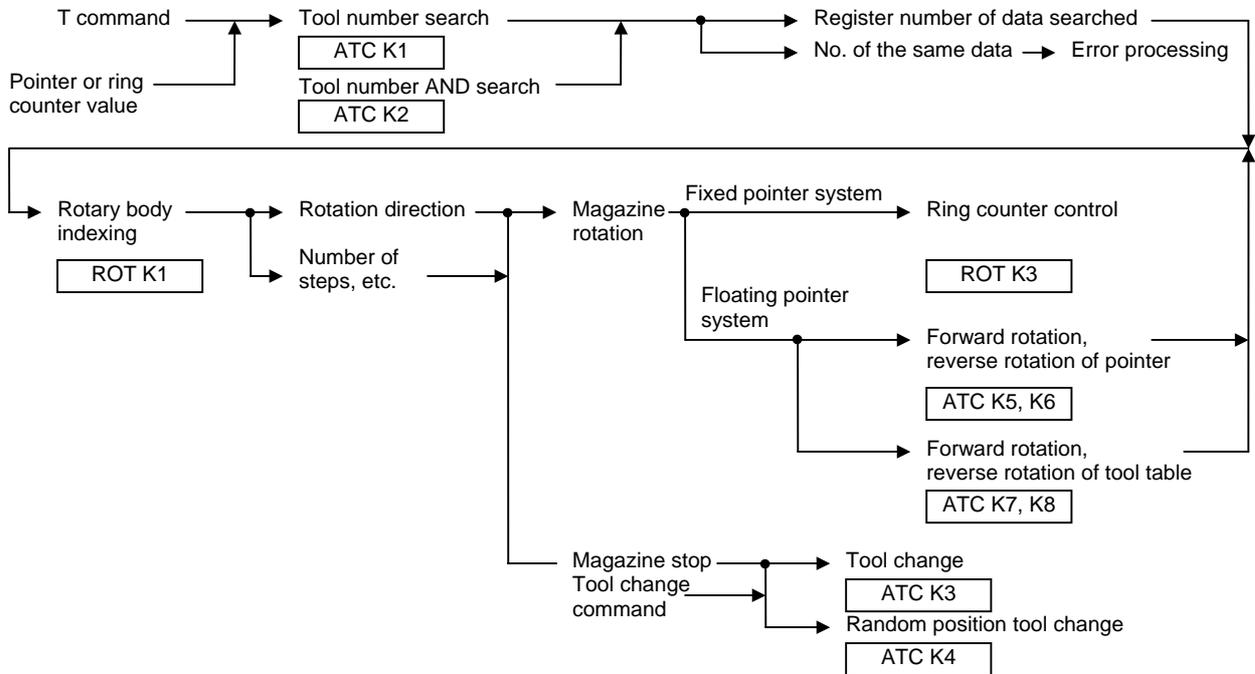
9.1.4 Relationship between Tool Registration Screen and Magazines



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.

9.1.5 Use of ATC and ROT Commands

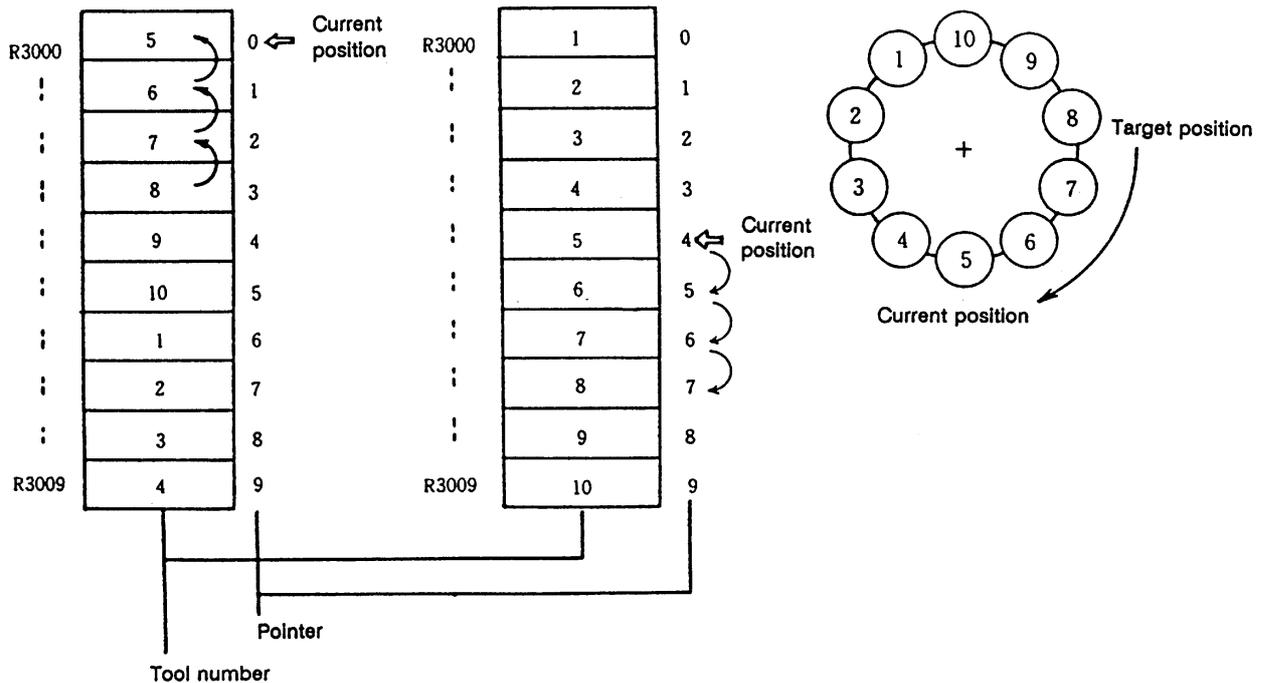
The use order of the ATC and ROT commands during the T command or tool change command is shown below:



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.

Tool table rotation system

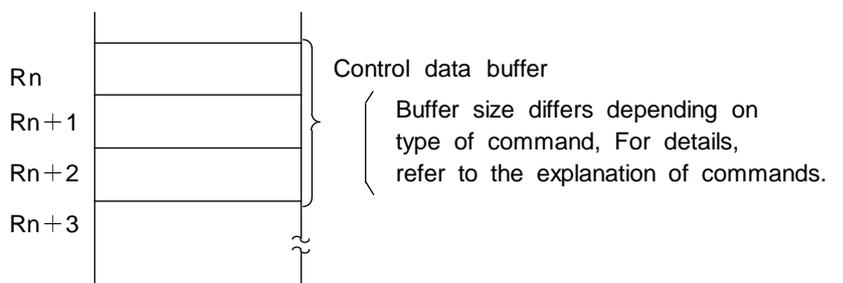
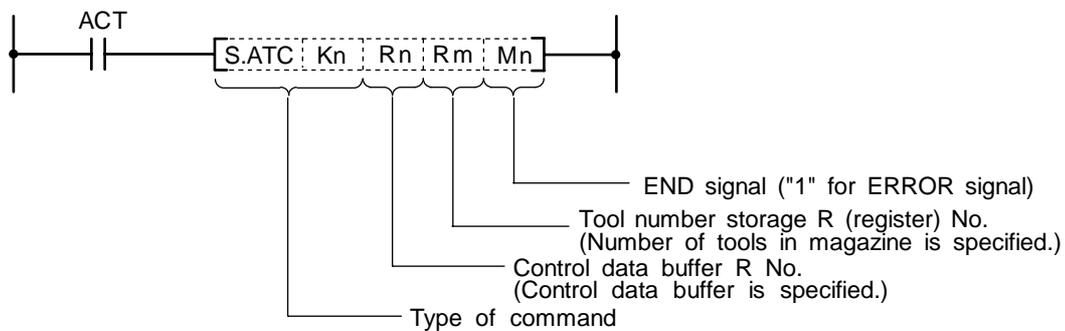
Floating pointer system



- (1) Index tool number 8 in the situation shown in the drawing.
 - (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.

9.1.6 Basic Format of ATC Exclusive Command



9.1.7 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 |
| | |

9.1.8 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 | — | — |

9.1.9 File Register (R Register) Assignment and Parameters

(1) File registers for ATC control

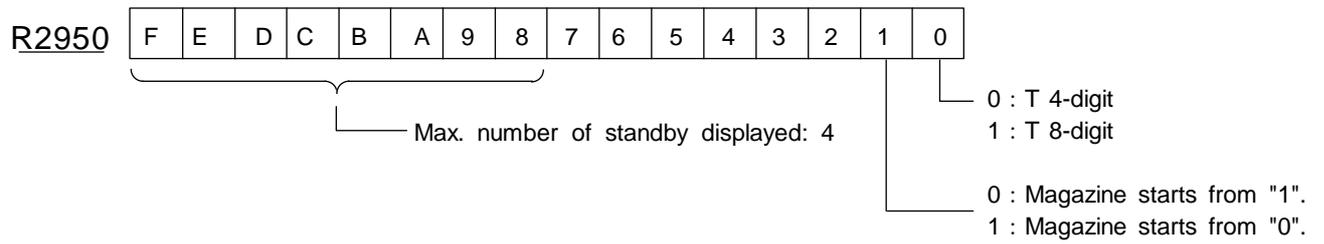
The file registers used with the ATC are as shown below.

| Magazine | Corresponding file (R) register | | | | | | Remarks (data type) | |
|-------------------------------------|---------------------------------|----------------|-------------------|----------------|-------------------|--------------|------------------------|-----|
| | 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 | R2950 | ← | ← | ← | ← | ← | — | |
| No. of magazine designation | R2960 | ← | R2961 | ← | R2962 | ← | Binary | |
| Pointer designation | R2965 | ← | R2966 | ← | R2967 | | Binary | |
| Spindle tool | R2970 | R2970 R2971 | R2980 | R2980 R2981 | — | — | BCD | |
| Standby 1 tool | R2971 | R2972 R2973 | R2981 | R2982 R2983 | — | — | BCD | |
| Standby 2 tool | R2972 | R2974 R2975 | R2982 | R2984 R2985 | — | — | BCD | |
| Standby 3 tool | R2973 | R2976 R2977 | R2983 | R2986 R2987 | — | — | BCD | |
| Standby 4 tool | R2974 | R2978 R2979 | R2984 | R2988 R2989 | — | — | BCD | |
| AUX data | R2998 | ← | ← | ← | ← | ← | Binary (0~99) | |
| Magazine tool data | MG1 | R3000 | R3000 R3001 | R3240 | R3240 R3241 | R3480 | R3480 R3481 | BCD |
| | MG2 | R3001 | R3002 R3003 | R3241 | R3242 R3243 | R3481 | R3482 R3483 | BCD |
| | MG3 | R3002 | R3004 R3005 | R3242 | R3244 R3245 | R3482 | R3484 R3485 | BCD |
| | ≈ | ≈ | ≈ | ≈ | ≈ | ≈ | ≈ | ≈ |
| | MG79 | R3078 | R3156 R3157 | R3318 | R3396 R3397 | R3558 | R3636 R3637 | BCD |
| | MG80 | R3079 | R3158 R3159 | R3319 | R3398 R3399 | R3559 | R3638 R3639 | 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



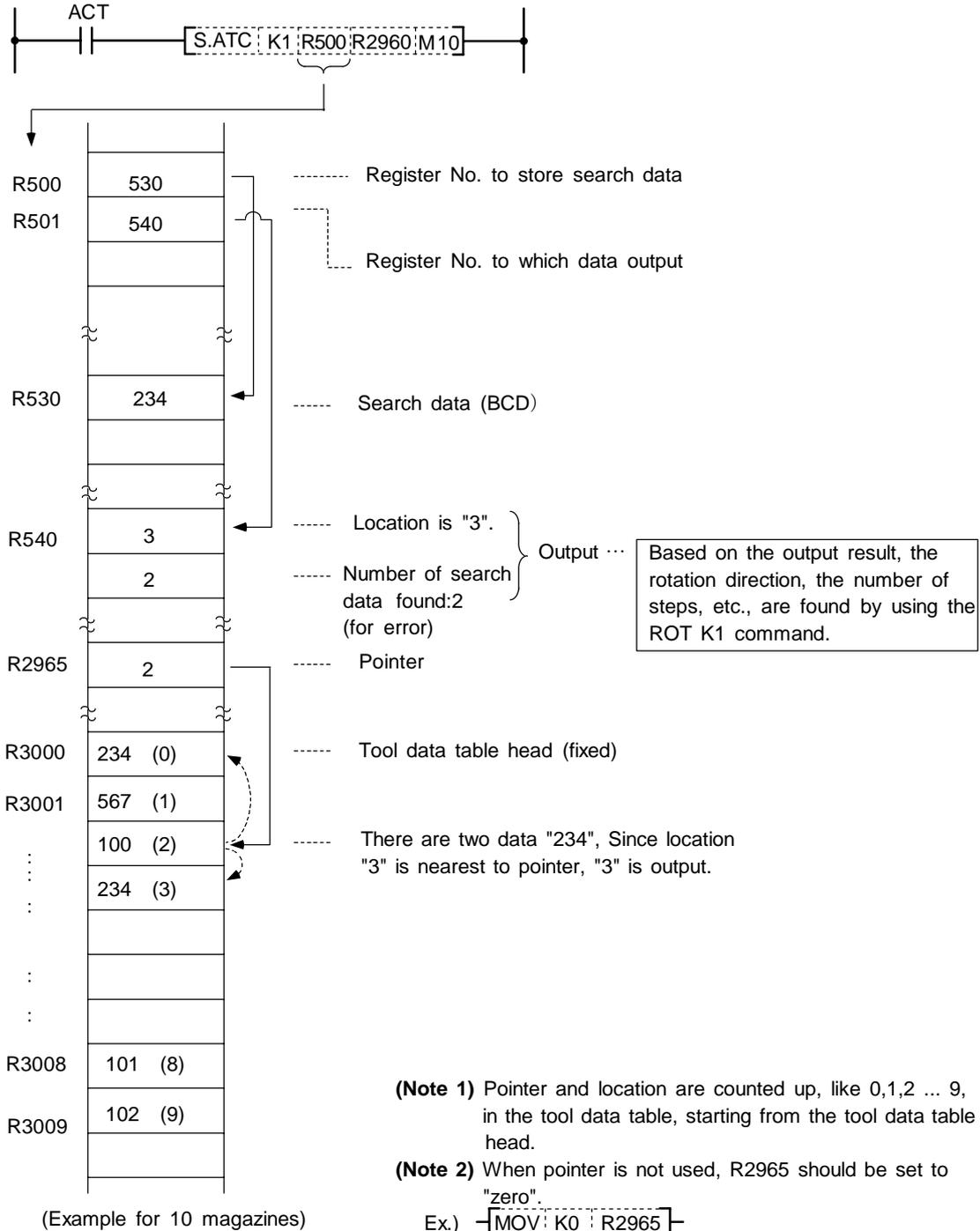
For details on the control parameters, refer to 9.1.12 Examples of Tool Registration Screen.

9.1.10 Details of Each Command

(1) Tool No. search

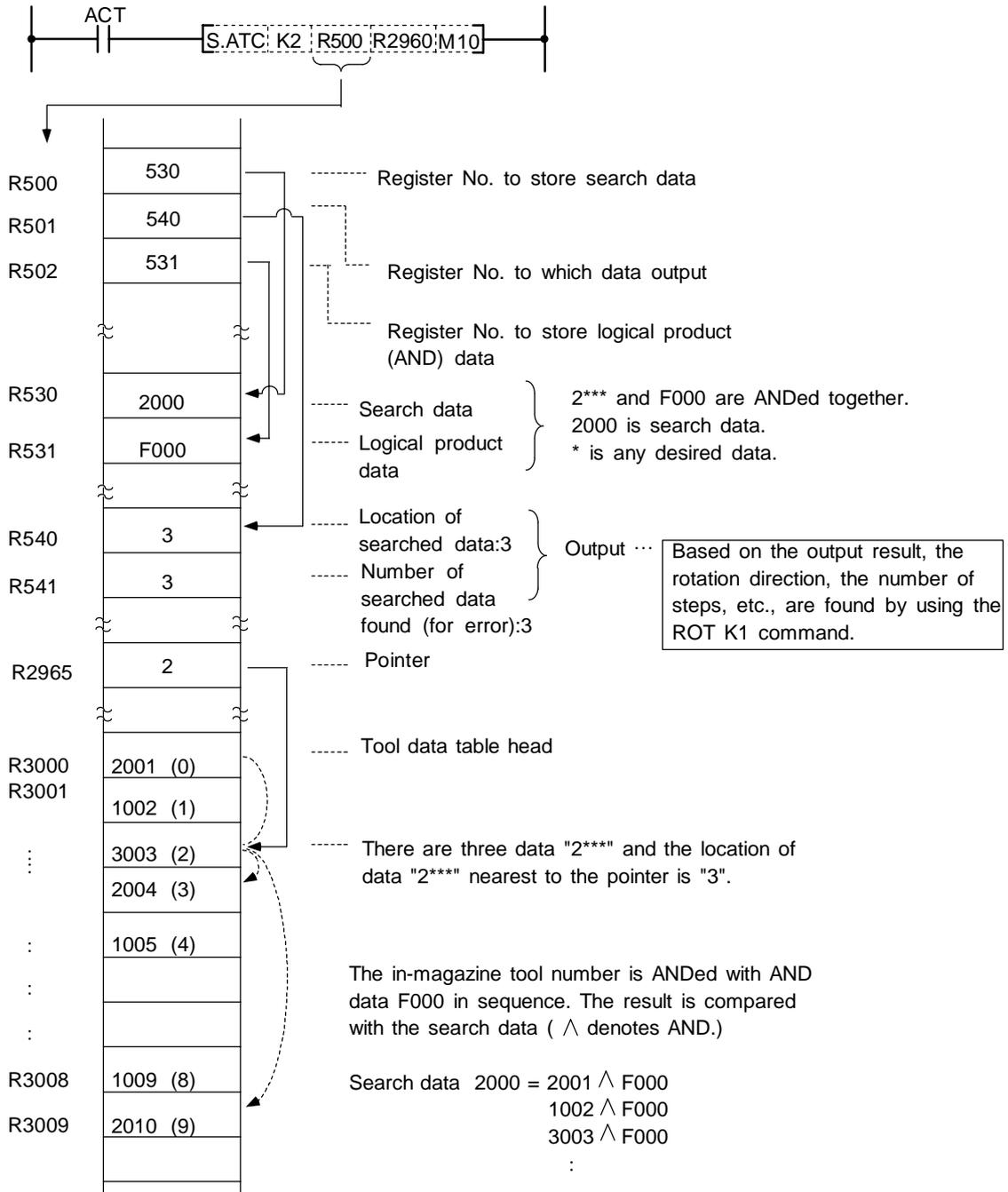
This command is used to search for tool No. stored in the tool data table.

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.



(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.



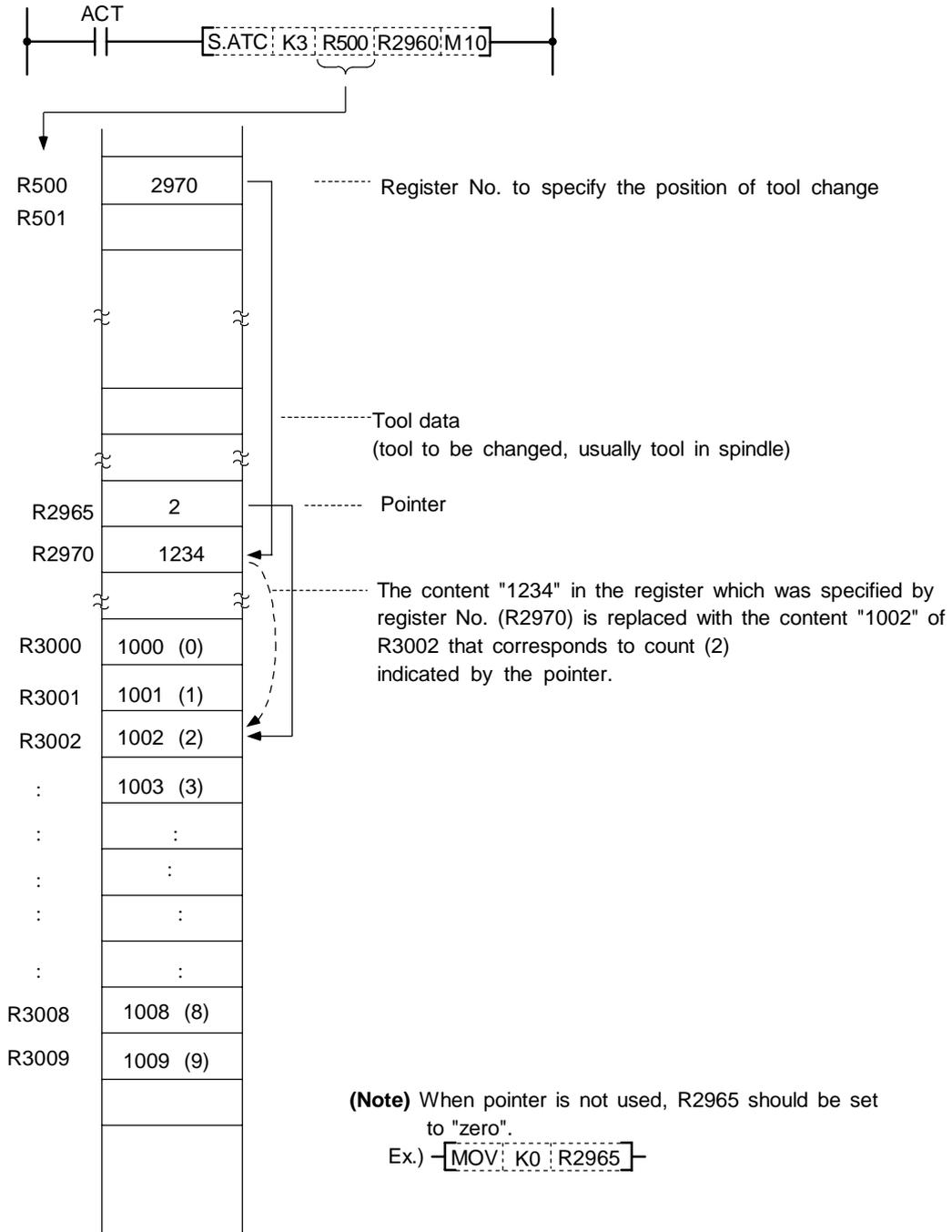
(Note 1) Pointer and location are counted up, like 0,1,2 ... 9, in the tool data table, starting from the tool data table head.

(Note 2) When pointer is not used, R2965 should be set to "zero".

Ex.) -[MOV K0 R2965]-

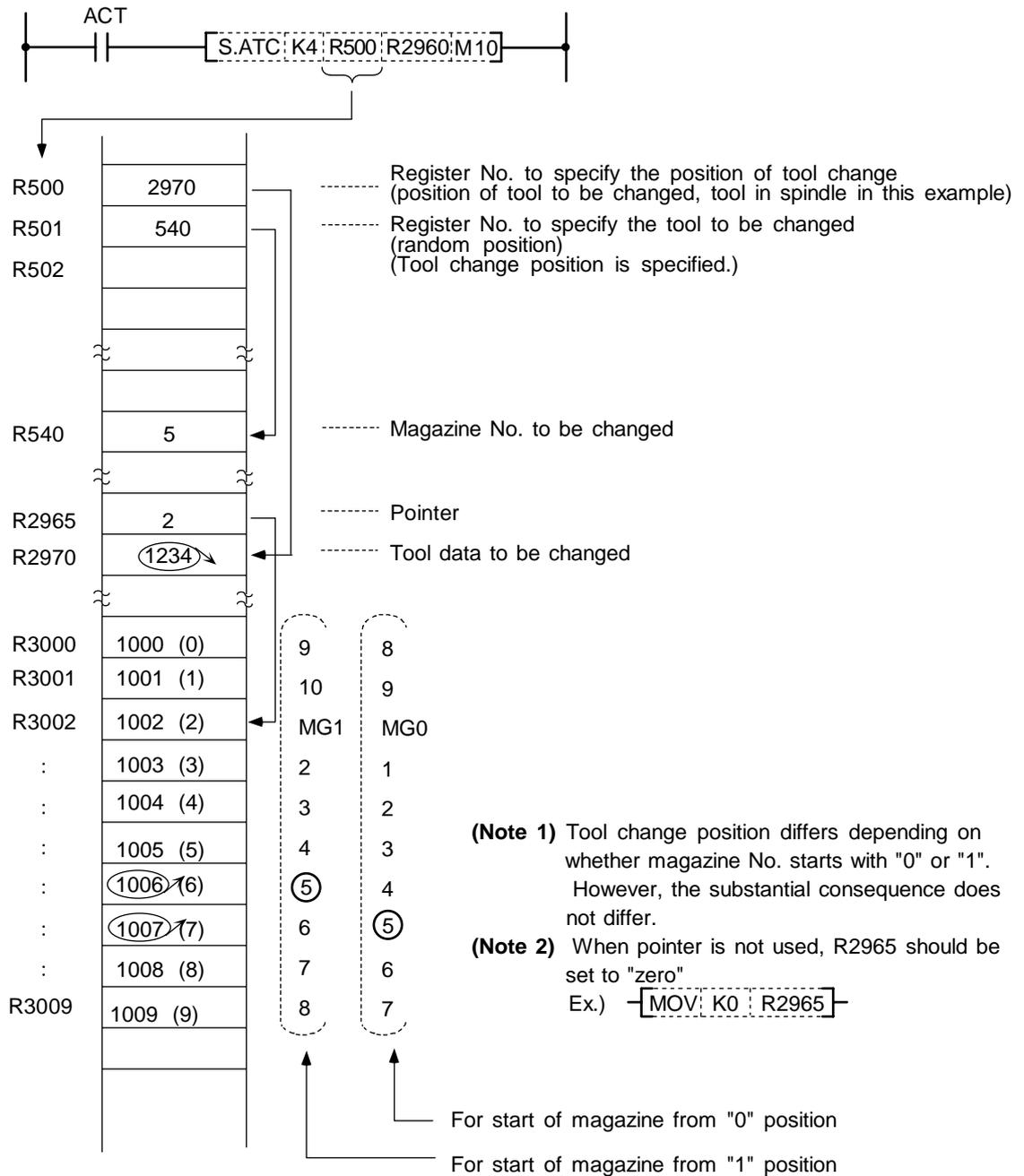
(3) Tool change

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.



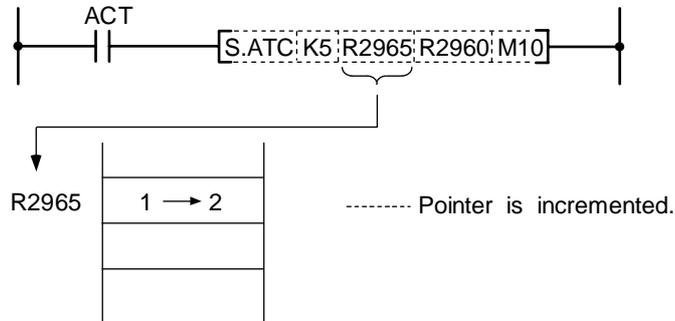
(4) Random position tool change

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.



(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.

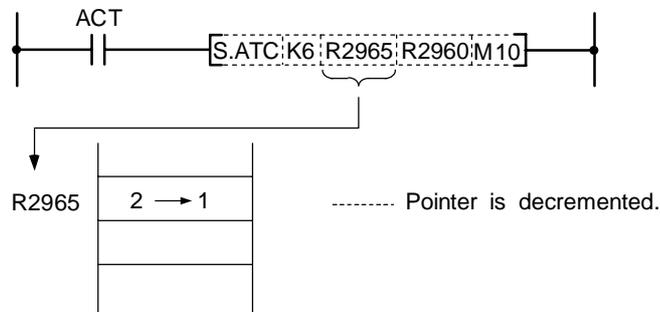


When a magazine with 10 tools is used, the control sequence is as follows:
0, 1, 2, 3 9, 0, 1, 2, 8, 9, 0, 1 ...

(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.

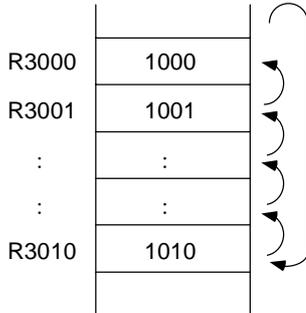


When a magazine with 10 tools is used, for example, the control sequence is as follows:
2, 1, 0, 9, 8 2, 1, 0, 9, 8 1, 0, 9, 8 ...

(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.

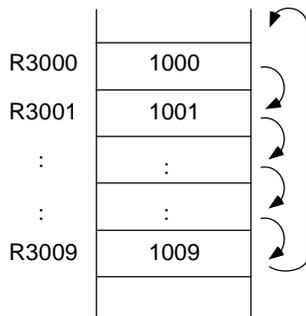


(Note 1) In this control mode, pointer always indicates "0" (tool table head).

(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.

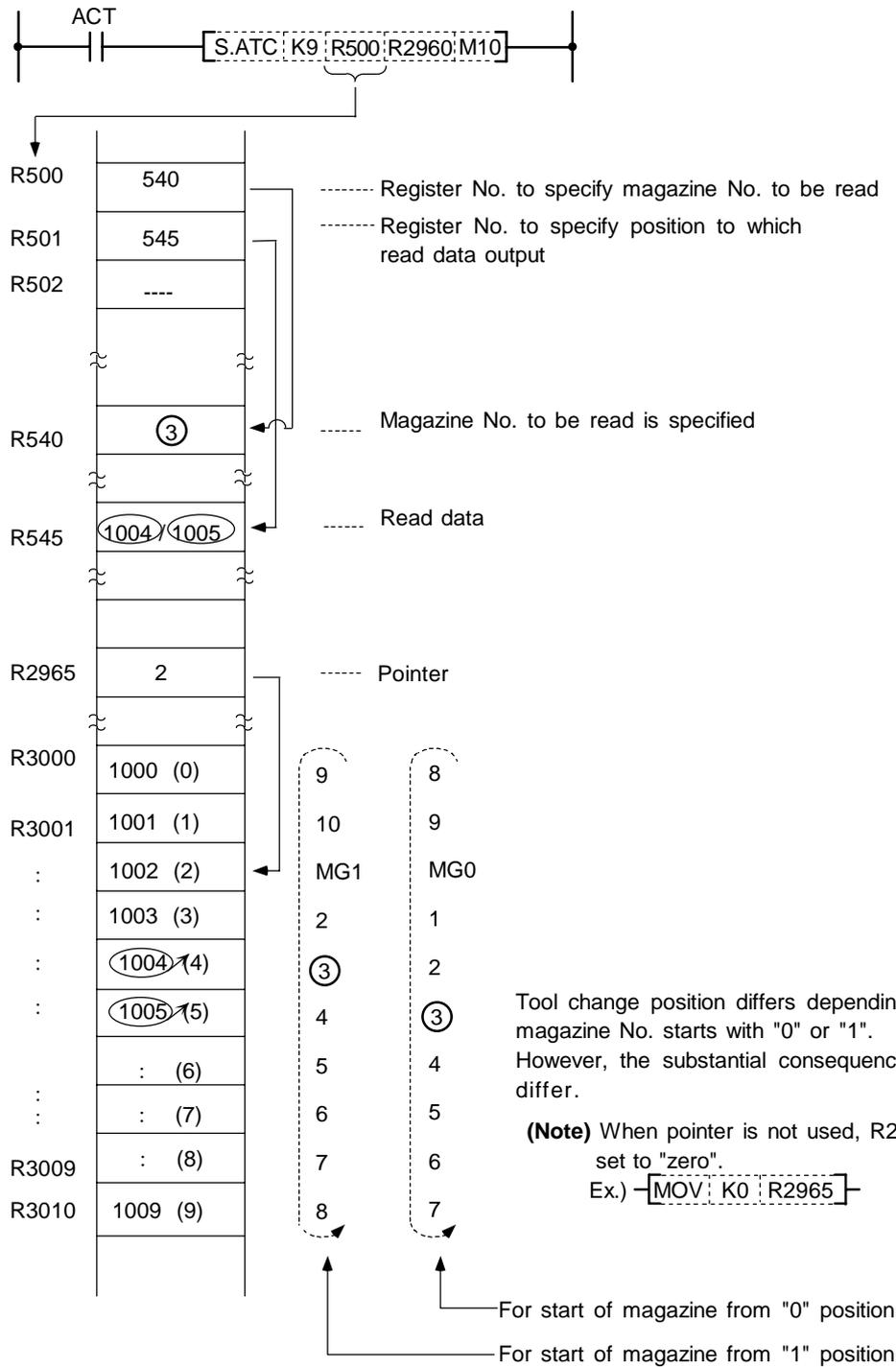


(Note 1) In this control mode, pointer always indicates "0" (tool table head).

(Note 2) When this command is executed, the relationship between magazine No. and tool No., appearing on the tool entry display, changes accordingly.

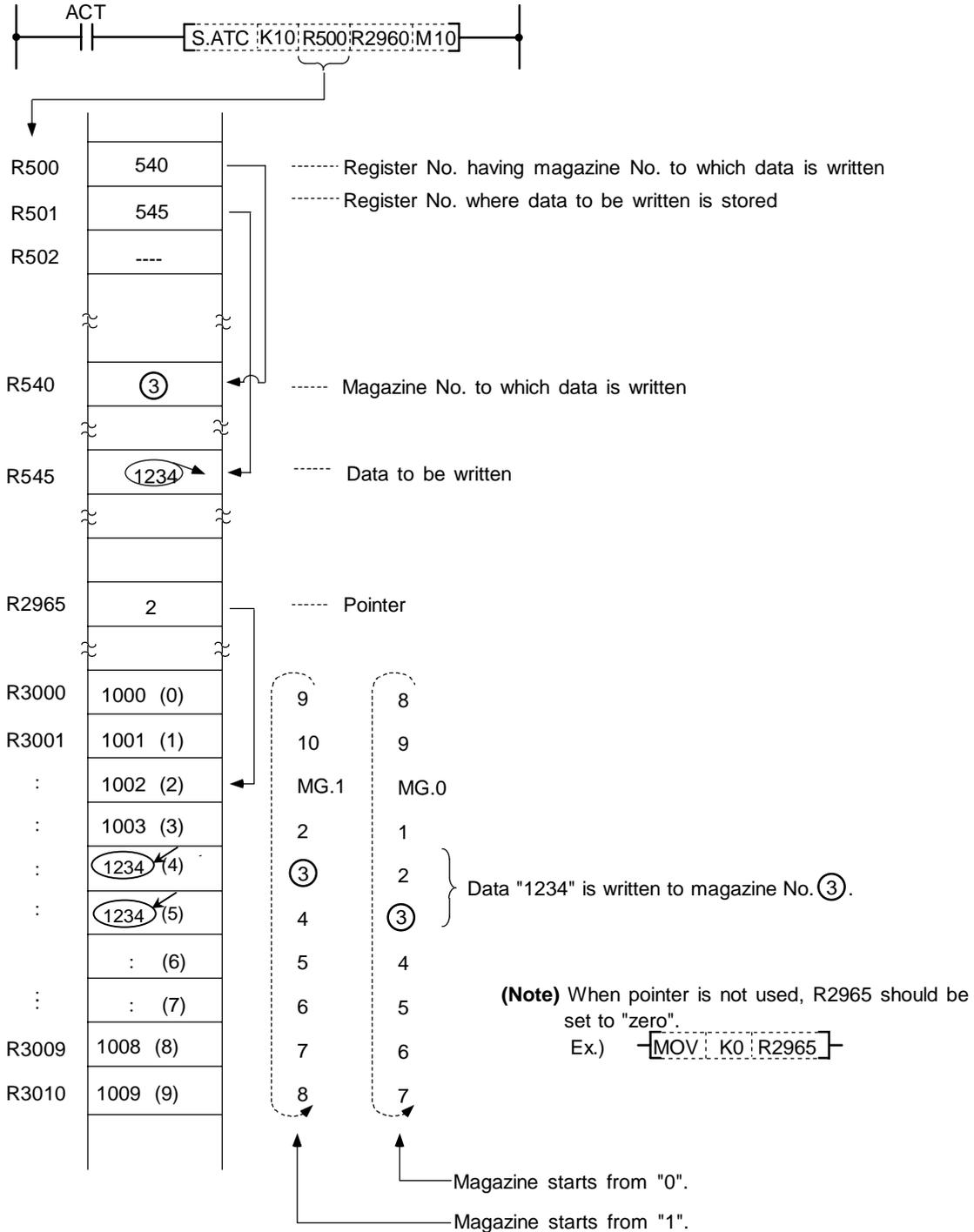
(9) Tool data read

This command is used to call a specific tool No. in the magazine.



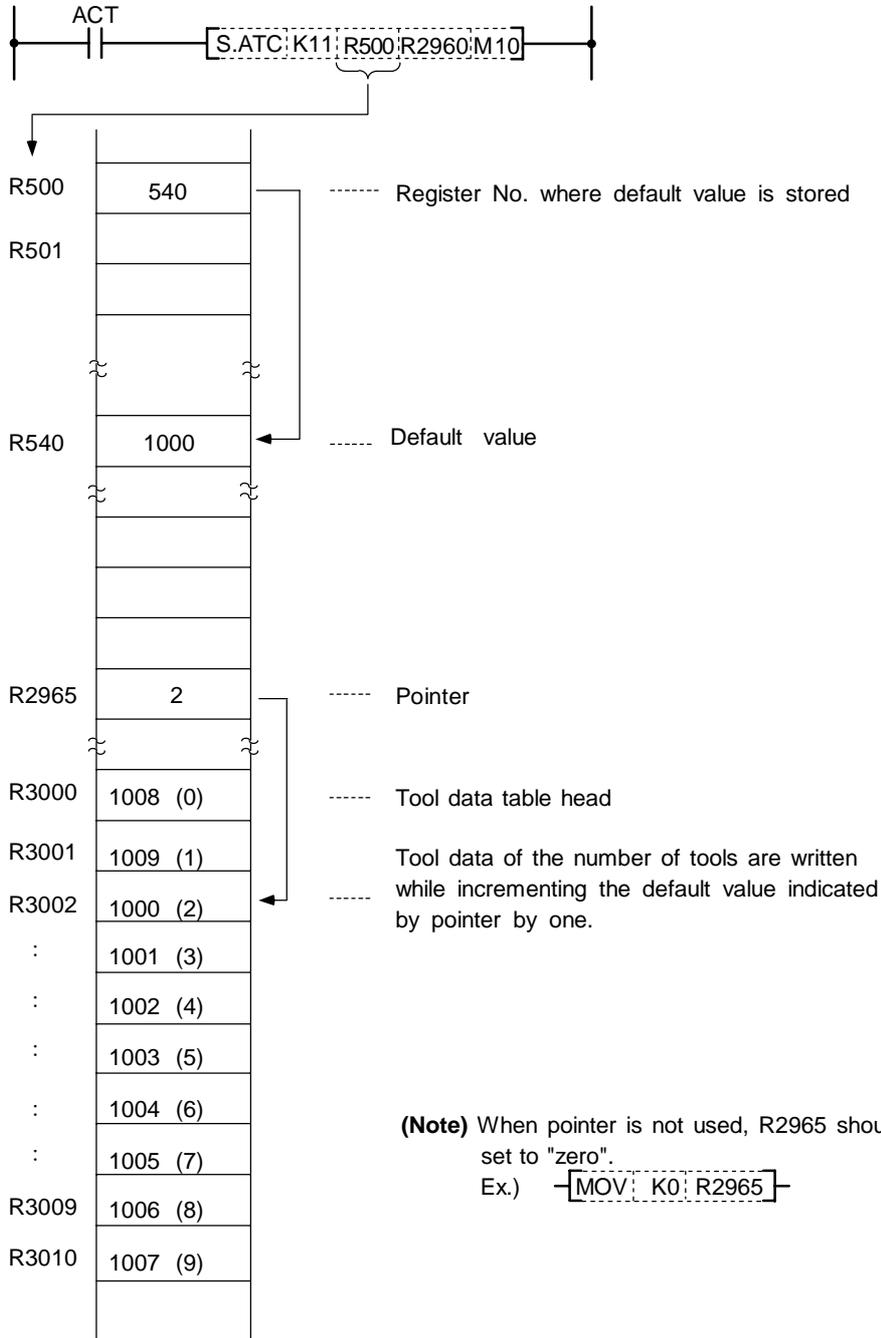
(10) Tool data write

Instead of setting tool No. through the setting and display unit, the tool No. is entered to each magazine No. set through PLC program.



(11) Automatic tool data write

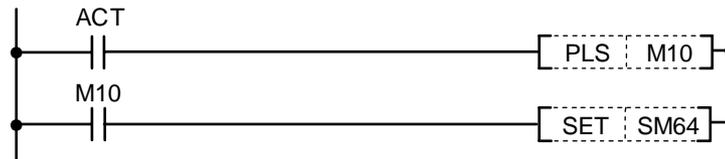
All tool Nos. are written (entered) in batch. This command is used for initialization, etc. The data are written one after another for each tool, starting from the default value.



9.1.11 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.

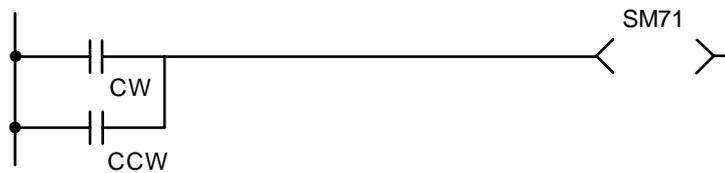
Program example)



- 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 (R2998) is valid while SM71 is being ON.

9.1.12 Examples of Tool Registration Screen

Tool registration screen examples are given below.
 For operation, refer to the Operation Manual.

| [TOOL REGISTRATION] | | | | TOOL2.1/3 | |
|-------------------------------|--------|--------|--------|-----------|--------|
| HEAD | NEXT 1 | NEXT 2 | NEXT 3 | INDEX | |
| MG | TOOL-D | MG | TOOL-D | MG | TOOL-D |
| 1 | | 11 | | 21 | |
| 2 | | 12 | | 22 | |
| 3 | | 13 | | 23 | |
| 4 | | 14 | | 24 | |
| 5 | | 15 | | 25 | |
| 6 | | 16 | | 26 | |
| 7 | | 17 | | 27 | |
| 8 | | 18 | | 28 | |
| 9 | | 19 | | 29 | |
| 10 | | 20 | | 30 | |
| T | M | | | | |
| MG () TOOL () D () AUX () | | | | | |

Manual numeric value command part
 Setting part

Comment display part (prepared by the user)
 Spindle, standby tool number display part
 Magazine number, magazine tool number guide display (fixed)
 Magazine tool number display part

OFFSET REGIST LIFE MENU

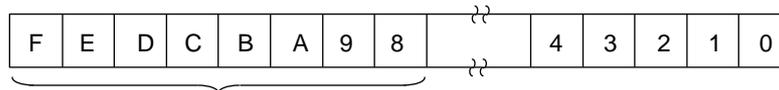
(1) Comment display part

Comment in the comment display part is prepared by the user who uses the comment display function described in the PLC Development Software Manual (BNP-B2252).

(2) Spindle tool, standby tool display part

The number of display items can be changed according to the control parameter value.

Control parameter (R2950)



- 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.

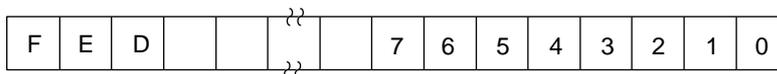
(a) Number of magazines

Number-of-magazine parameter (R2960): 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 (R2950)



- └ 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 | 5 |
| 2 | 11 |
| 5 | 12 |

The magazine number starts at 1.

| MG TOOL-D | MG TOOL-D |
|-----------|-----------|
| 0 | 5 |
| 1 | 10 |
| 5 | 11 |

The magazine number starts at 0.

9.1.13 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 for setting and display unit type 9

| | | | |
|---------------------|----------|--------------|------|
| 012345678 N12345-12 | | MONITOR 1 | |
| <SUB> | 0 5678 M | 45-12 [HEAD] | 1234 |
| [POSITION] | | [NEXT1] | 5678 |
| X | -345.678 | S | 2345 |
| Y | 345.678 | T | 1234 |
| Z | 0.000 #1 | M | 56 |
| C | 0.000 | FC | 0.00 |

} Spindle and standby display

```

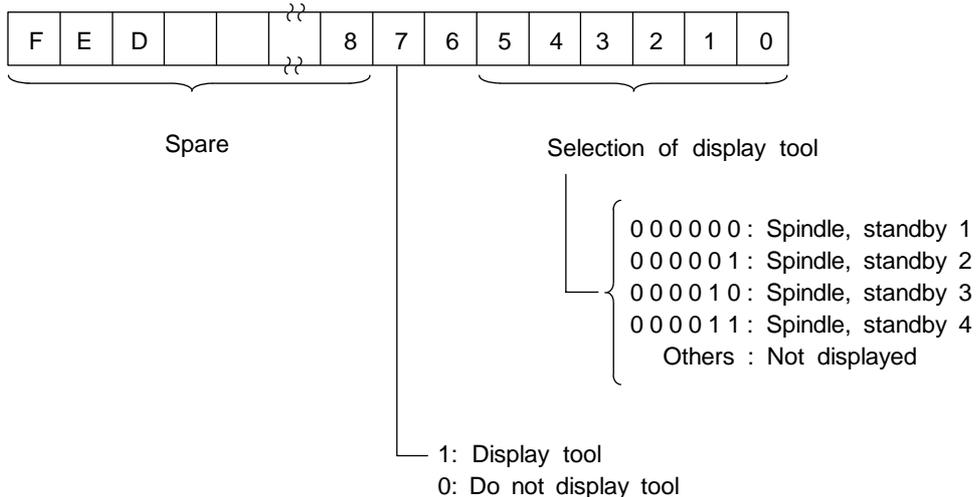
G00 X-345.678 Y345.678;
T1234;
N100 S5000 M3;
N200 G00 Z-100.;
    
```

POSI COORDI COMMAND SEARCH MENU

(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 (R2953)



9.2 S.ROT Commands

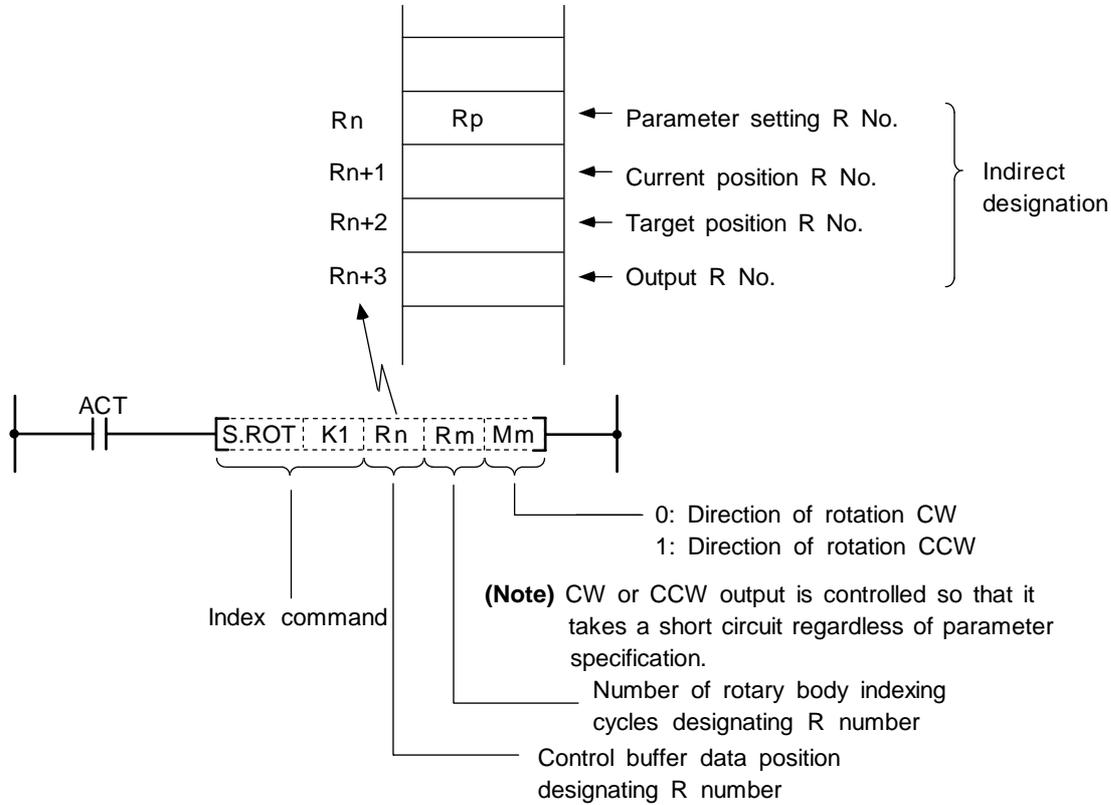
ROT commands are prepared as functions such as rotary body target position, rotation direction and ring counter. The commands can be used to determine the direction of rotation and number of steps with the data resulting from ATC exclusive command tool No. search processing.

9.2.1 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



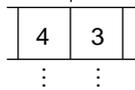
F

E

Error output
0: Normal completion
1: Error completion

(Spare)

0: Rotary body starts from 1
1: Rotary body starts from 0



0: Direction of rotation is determined for shorter reach.
1: Direction of rotation is not determined for shorter reach.

0,0 : (Step No. +1) is counted.
0,1 : Step No. is counted.
1,0 : (Step No. -1) is counted.

(When the current value is equal to the target value, -1 will appear.)

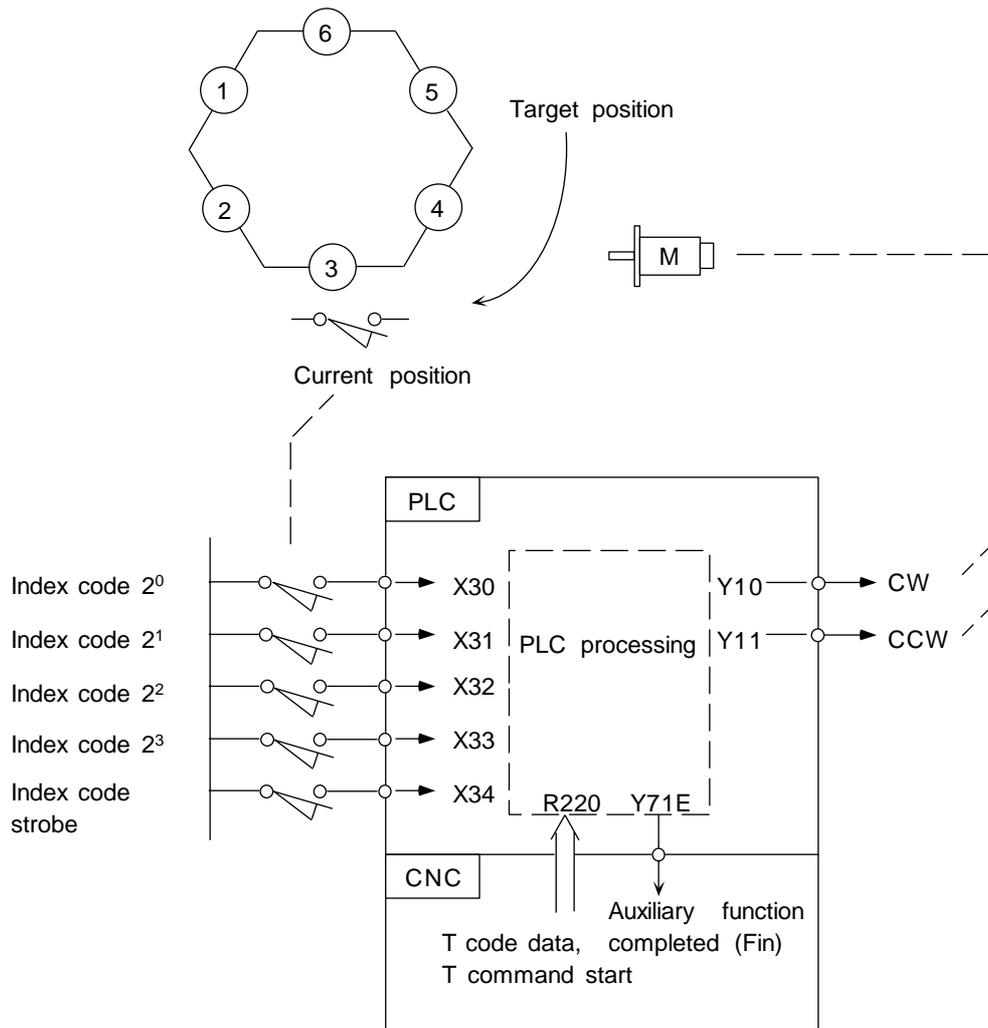
(Note 1) The Index command is executed after setting R numbers to Rn to Rn+3 and writing data in the file registers (R) each corresponding to the R numbers. However, data setting to the parameter (Rp) is done once before execution of the Index command; this is to prevent the error code from being cleared.

(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

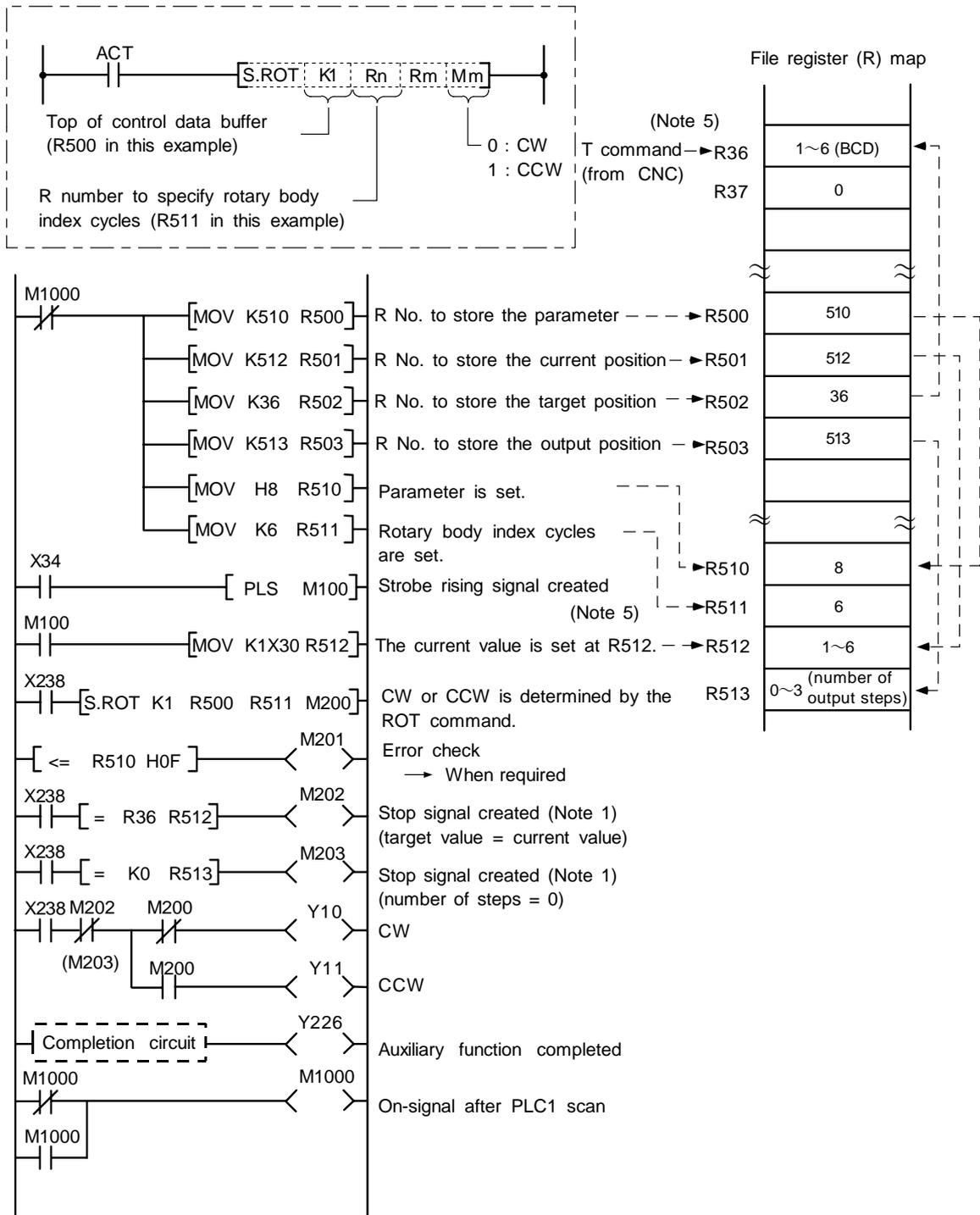
- Conditions:** (i) The number of rotary body index cycles is 6.
(ii) The target position is specified by a T command.

(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 R36 (B C D) 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.

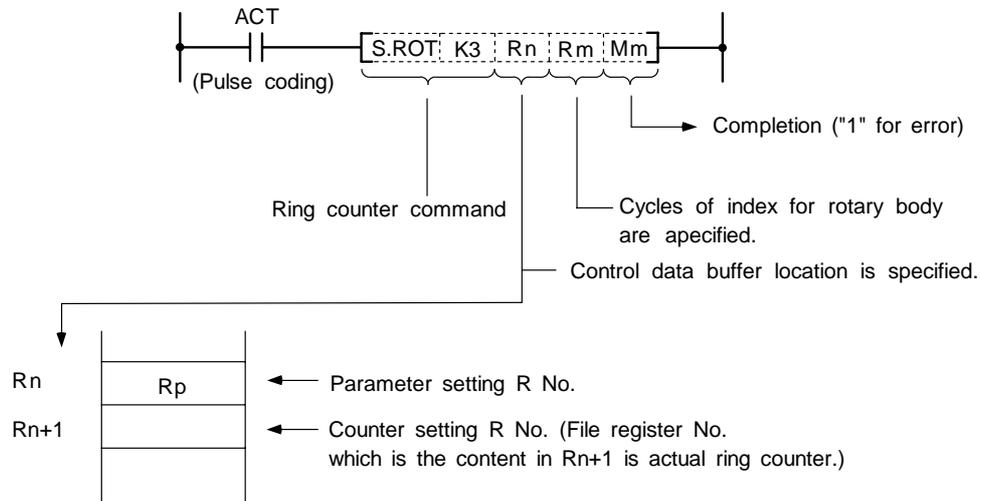
9. Exclusive Commands



- (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 (R512) before the ROT command is activated, an error results.
- (Note 4)** The control parameters (R510) 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 (R36) 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 R36 are used as they are. The target position and current value (R36 and R512 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 R36 are binary converted.)

(2) Ring counter (Up/down counter)

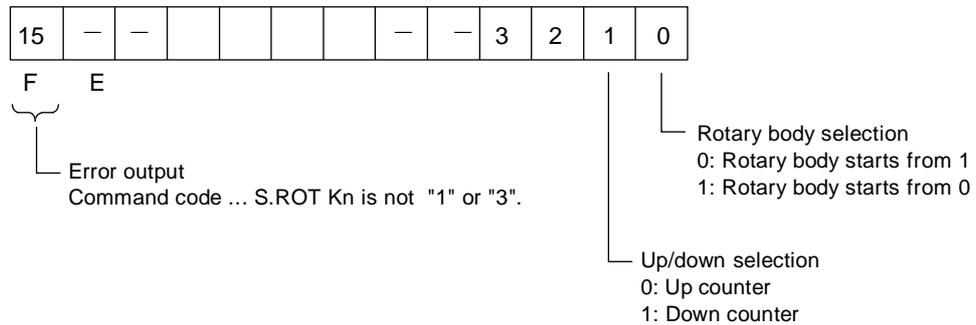
This command is used to control position of rotary body (or turret).



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



(Note 1) The ring counter command is executed after setting R numbers to Rn to Rn+1 and specifying data for the parameter.

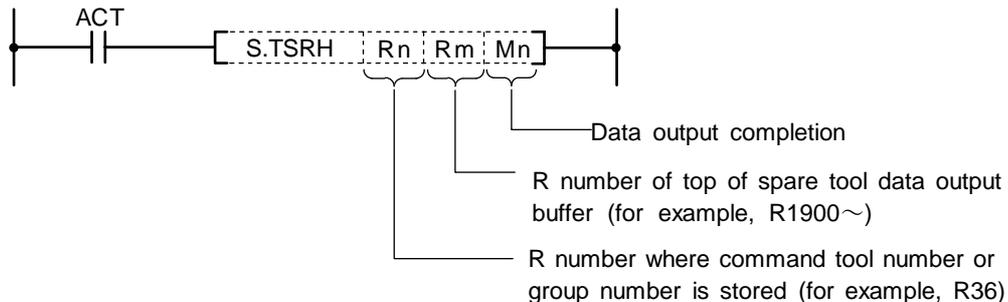
(Note 2) The error code (Mm) of the ring counter command and the error code in bit F of the parameter (Rp) are cleared when the activating signal (ACT) goes off. The activating signal (ACT) of the ring counter command is generally pulsed. This makes it hard for the interface diagnostic and ladder monitor programs to detect an error signal. For debugging, therefore, an error hold circuit is provided after the ring count command to ease error detection.

9.3 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



9.3.1 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 (R3720, R3721) is integrated and the tool use state is monitored. Tool data corresponding to the spindle tool is also output. (R3724~R3735)

- (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 (R3724~R3735) and tool offset corresponding to the spindle tool is made.

9.3.2 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.

9.3.3 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.

9.3.4 Interface

(1) User PLC → Controller

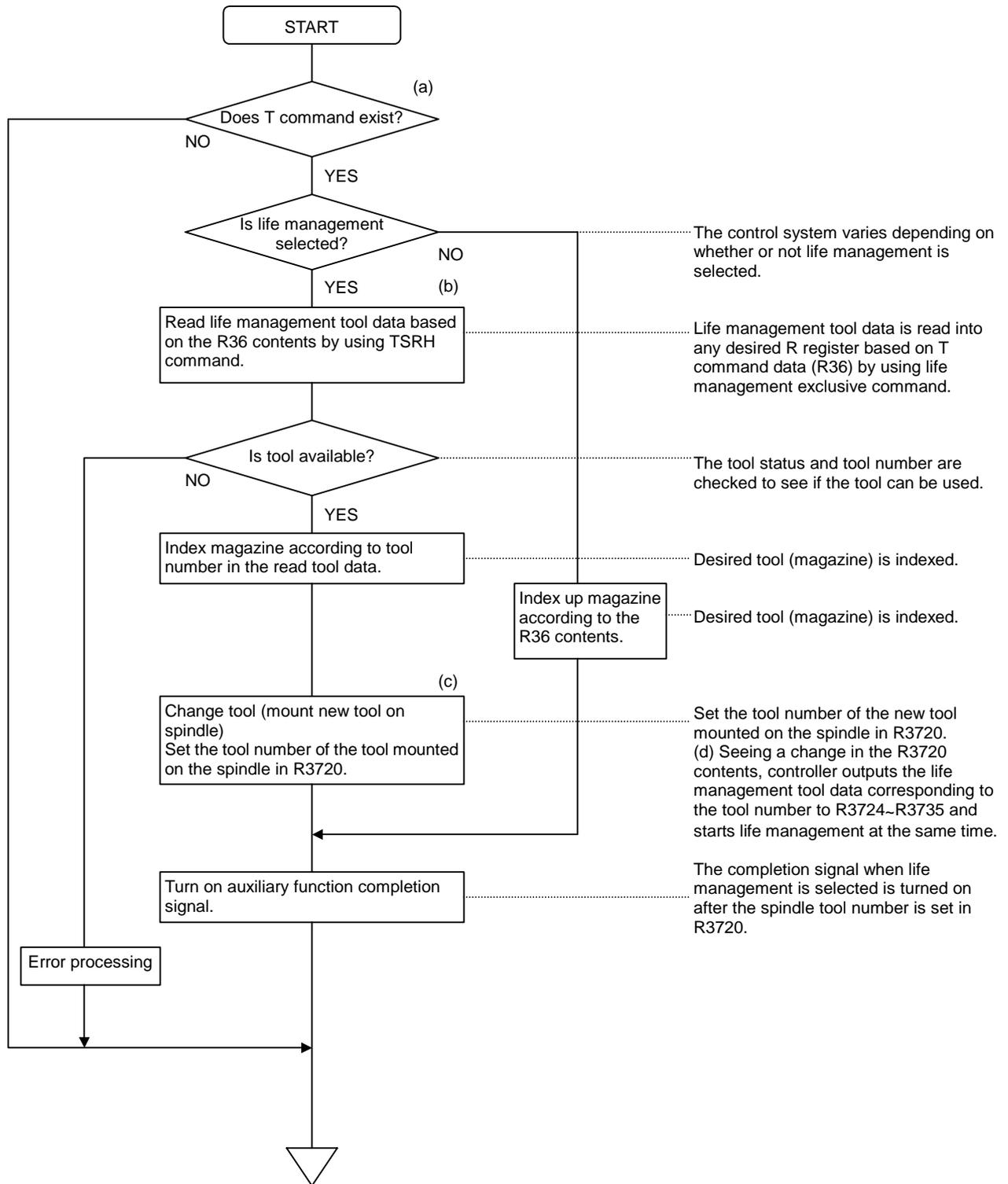
| Device name | Signal name | Explanation |
|-------------|------------------------------------|---|
| Y29A | Auxiliary function locking signal | While this signal is input, tool life management is not made. |
| Y2C8 | Tool error 1 signal | This signal indicates tool error state 1. When controller inputs the signal it changes the status in spindle tool data to 3. (Unused tools or used tools are changed to toll error state 1.) |
| Y2C9 | 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.) |
| Y2CA | Usage data counter validity signal | If this signal is not input, the usage data is not counted. |
| Y2CB | 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 |
|-------------|------------------------------------|---|
| X20B | 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.) |

9.3.5 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.

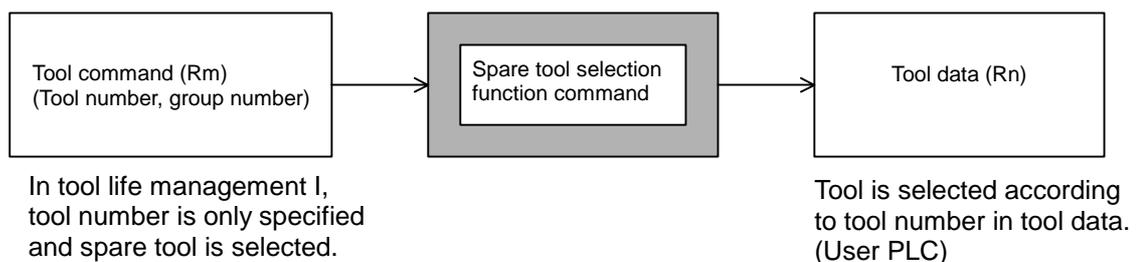
(Note) If -1 is set in the group number in the output tool data, the tool data is invalid. At the time, the specified tool number is output to the tool number in the output tool data as it is.

(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 (R3720~R3721). The controller outputs the spindle tool data corresponding to the tool number of the spindle tool every user PLC main cycle (R3724~R3735).
- 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.

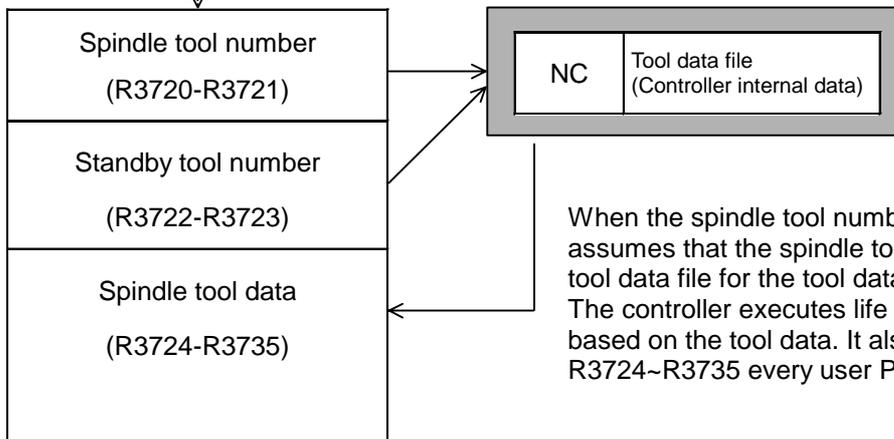
In tool life management II, it also executes tool offset corresponding to the spindle tool.

(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 (R3720~R3721) 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>

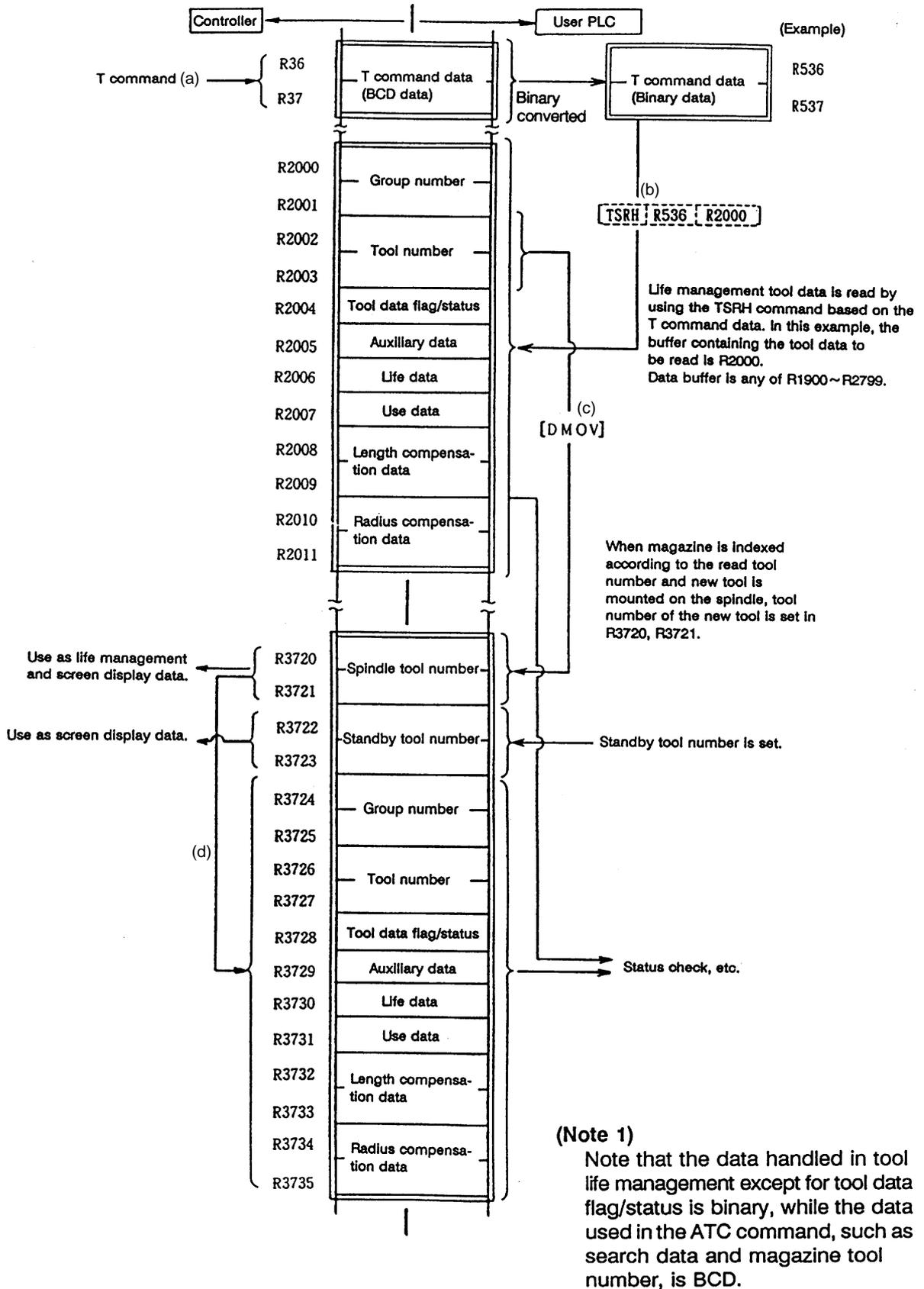
<When tool is changed>

When tool is changed, the spindle tool number is set in R3720, R3721. (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 R3724~R3735 every user PLC main cycle.

(3) Tool data flow



(4) Tool data

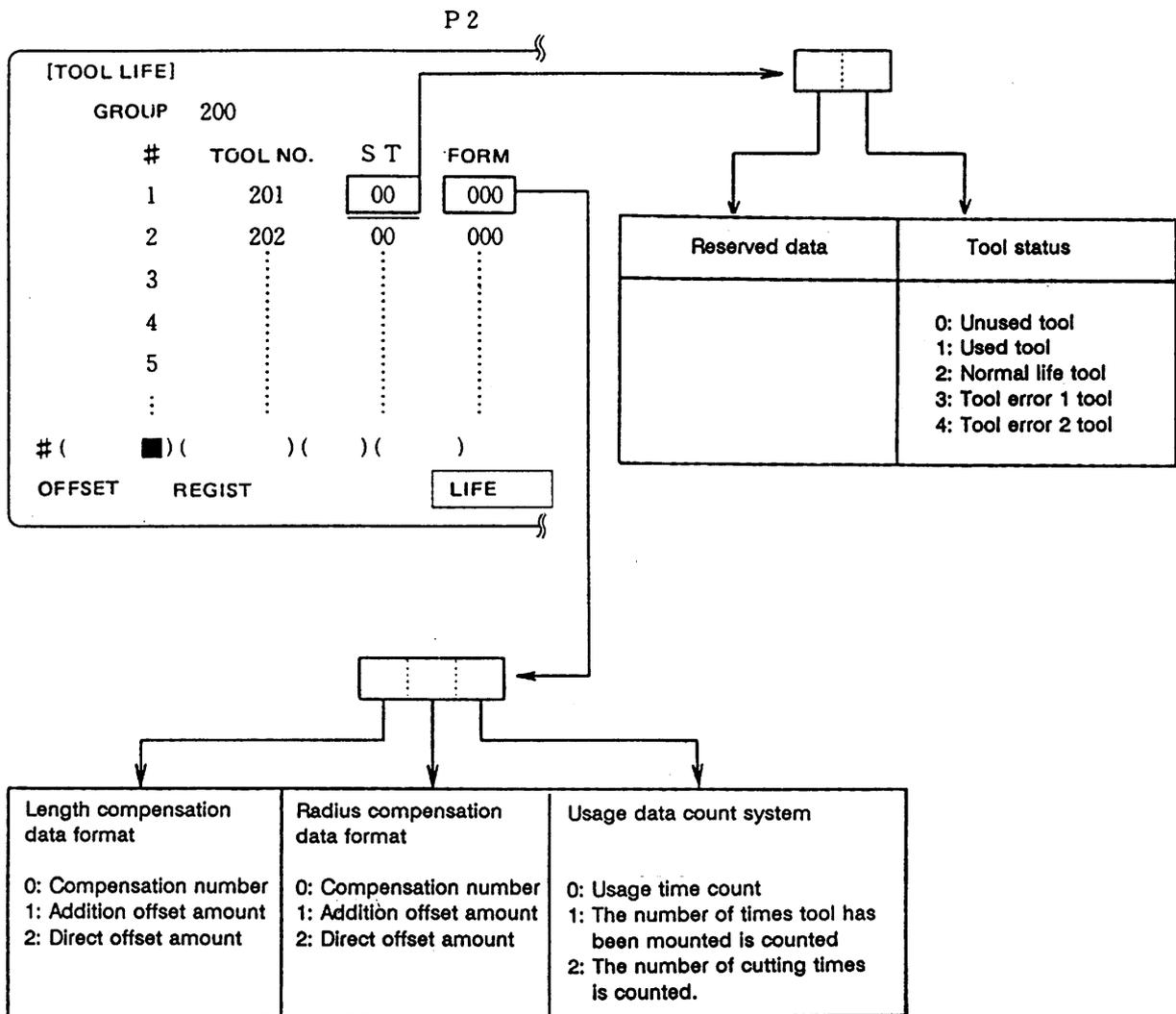
The tool data is tool management data such as the group number, tool number, and tool status. The details are given below:

| 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 use data | Use time or use 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 R3728)

| bit | Explanation |
|-------|--|
| bit 0 | Length compensation data format (spare tool compensation system) |
| bit 1 | Radius compensation data format (spare tool compensation system) |
| bit 2 | Usage data count system |
| bit 3 | Usage data count system |
| bit 4 | Usage data count system |
| bit 5 | Usage data count system |
| bit 6 | Usage data count system |
| bit 7 | Usage data count system |

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 number 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.

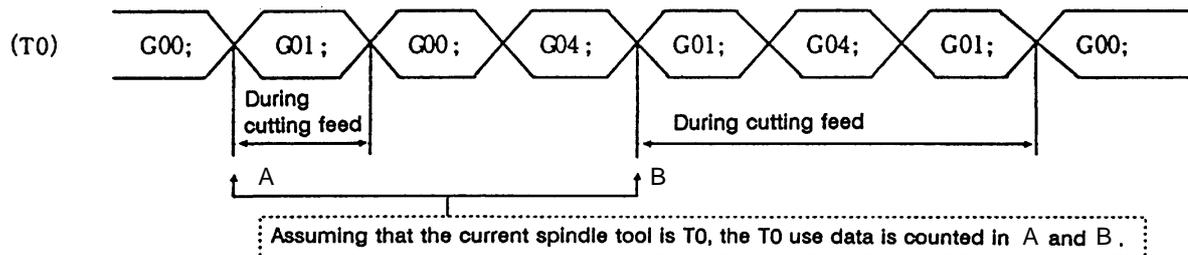
- ii) Number of times tool has been mounted is counted

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 R3728)

| 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 | |
| bit D | (Reserved) |
| 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. |

9.3.6 Examples of Tool Life Management Screen

Tool life management screen examples are given below.
For operation, refer to the Operation Manual.

| [TOOL LIFE] | | TOOL 4.1/2 | | | | | | | |
|-------------|----------|------------|----------|----------|----------|----------|----------|----------|----------|
| | GROUP | TOOL NO. | ST FORM | L-CMP | R-CMP | AUX | LIFE | USED | |
| HEAD | 10000000 | 12345678 | 1 | 000 | -345.678 | 100.000 | 12345 | 234 | 34(min) |
| NEXT | 80000000 | 87654321 | 0 | 000 | 45.678 | 30.000 | 12345 | 234 | 4(min) |
| | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 |
| | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 |
| | 1000 | 2000 | 3000 | 4000 | 5000 | 6000 | 7000 | 8000 | 9000 |
| | 10000 | 20000 | 30000 | 40000 | 50000 | 60000 | 70000 | 80000 | 90000 |
| | 100000 | 200000 | 300000 | 400000 | 500000 | 600000 | 700000 | 800000 | 900000 |
| | 1000000 | 2000000 | 3000000 | 4000000 | 5000000 | 6000000 | 7000000 | 8000000 | 9000000 |
| | 10000000 | 20000000 | 30000000 | 40000000 | 50000000 | 60000000 | 70000000 | 80000000 | 90000000 |

OFFSET | REGIST | **LIFE** | MENU

| [TOOL LIFE] | | TOOL 4.1/2 | | | | | | | |
|-------------|----------|------------|---------|-------|----------|---------|-------|------|----------|
| | GROUP | TOOL NO. | ST FORM | L-CMP | R-CMP | AUX | LIFE | USED | |
| | 10000000 | | | | | | | | |
| | 1 | 12345678 | 4 | 220 | -345.678 | 100.000 | 12345 | 1234 | 234(min) |
| | 2 | 1234567 | 3 | 120 | 112.340 | 30.000 | 11111 | 123 | 45(min) |
| | 3 | 123456 | 2 | 111 | 122.220 | 20.000 | 44444 | 100 | 50(set) |
| | 4 | 12345 | 1 | 002 | 11.234 | 100.123 | 100 | 50 | 15(cyc) |
| | 5 | | | | | | | | |
| | 6 | | | | | | | | |
| | 7 | | | | | | | | |
| | 8 | | | | | | | | |
| | 9 | | | | | | | | |
| | 10 | | | | | | | | |

() () () () () () () () () ()

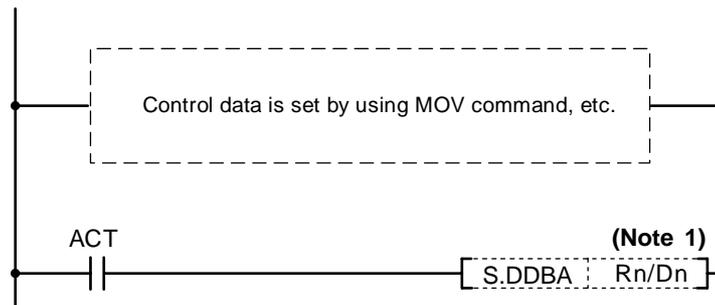
OFFSET | REGIST | **LIFE** | MENU

Tool life management screen example on type 9 setting and display unit

9.4 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.

9.4.1 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 R500 through R549 (not backed up) and R1900 through R2799 (backed up).

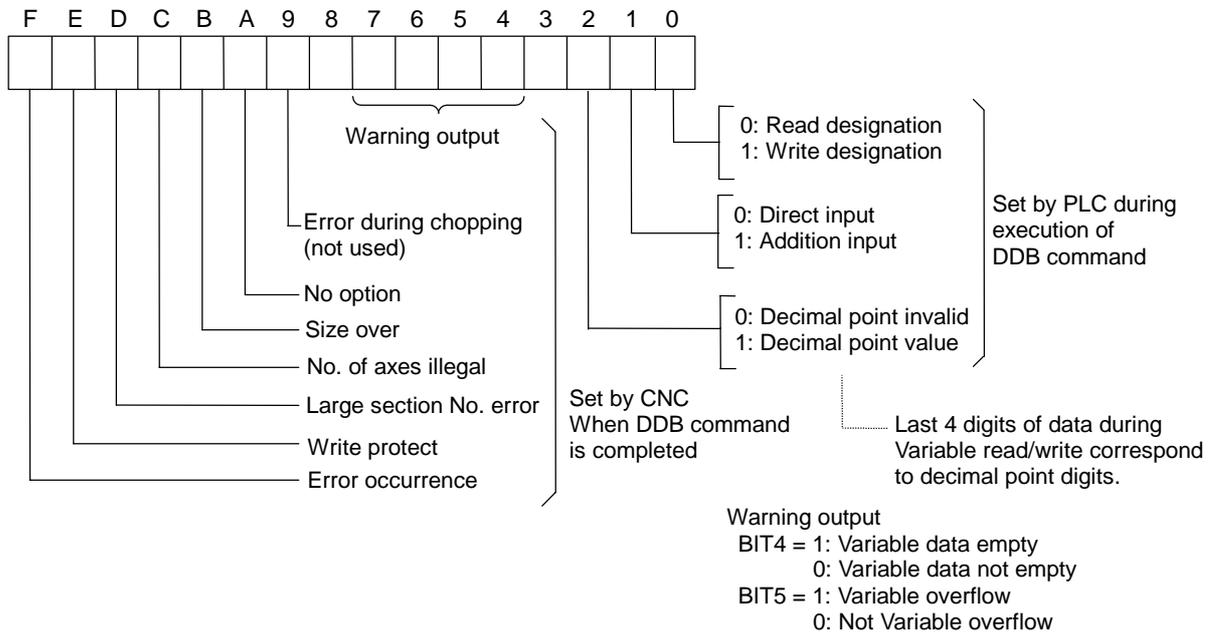
9.4.2 Basic Format of Control Data

| | |
|--------------|--|
| 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

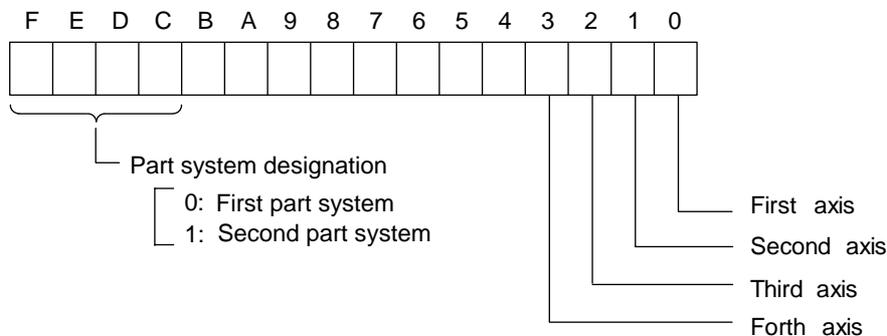
2: Two bytes

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)

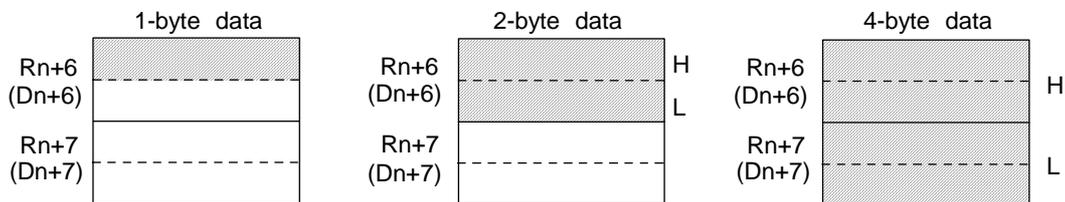
Specify the axis to read or write data for each axis classified by major classification numbers.



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 work 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 work 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.

9.5 External Search

9.5.1 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 or tape for the program number, sequence number, and block number.

9.5.2 Interface

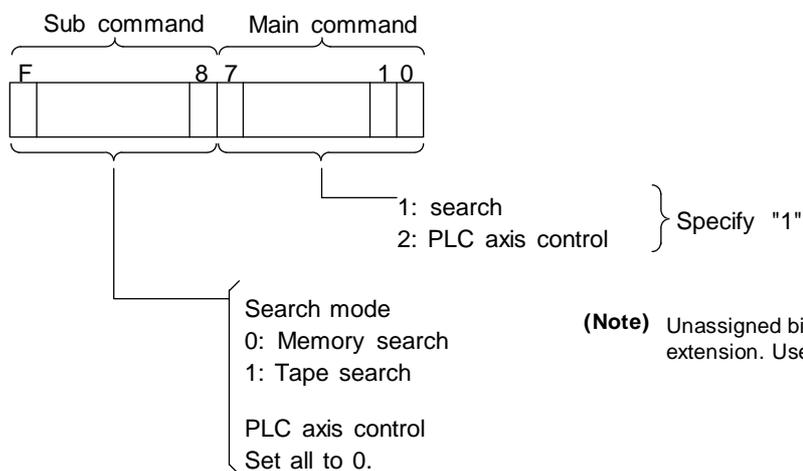
PLC sets data except the status.

| | | |
|------|------------|-------------------------|
| Rn+0 | Two bytes | command |
| 1 | Two bytes | Status |
| 2 | Four bytes | Program No. |
| 3 | | |
| 4 | Four bytes | Sequence No. |
| 5 | | |
| 6 | Four bytes | Block No. |
| 7 | | |
| 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-part system specifications.

(1) Command

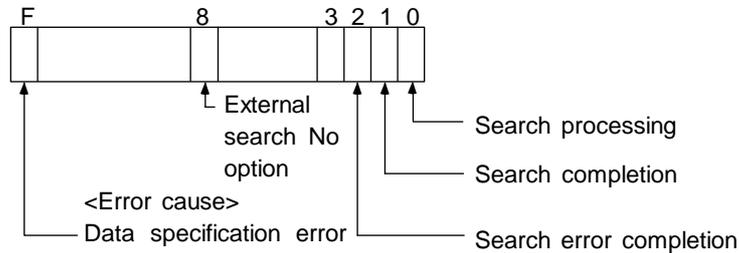


(Note) Unassigned bits will be used for later function extension. Use only bits shown here.

(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).

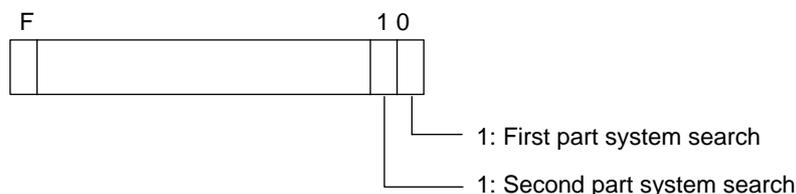
If a number other than 0~99 is specified, a data specified error will occur.

| 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.

If multiple part systems are specified, all the specified part systems are searched simultaneously. Note that the program No., sequence No. and block No. to be searched are common for all the part systems.



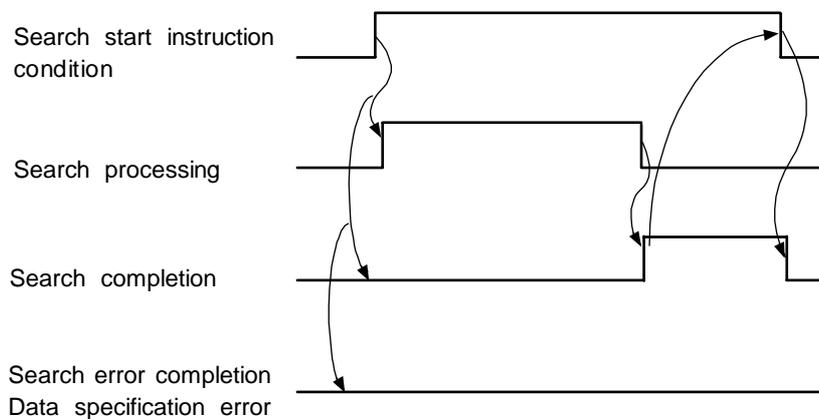
9.5.3 Search Start Instruction

After interface data between the controller and PLC is prepared, search is started by using the following instruction:

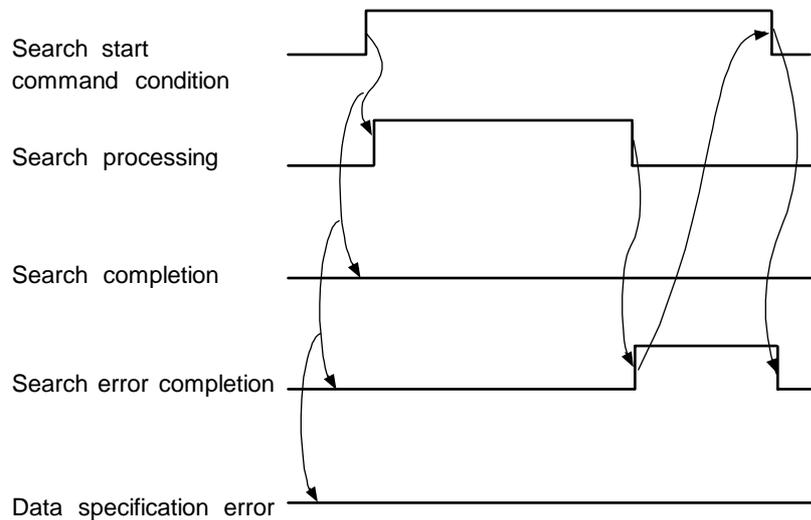


9.5.4 Timing Charts and Error Causes

(1) Normal completion



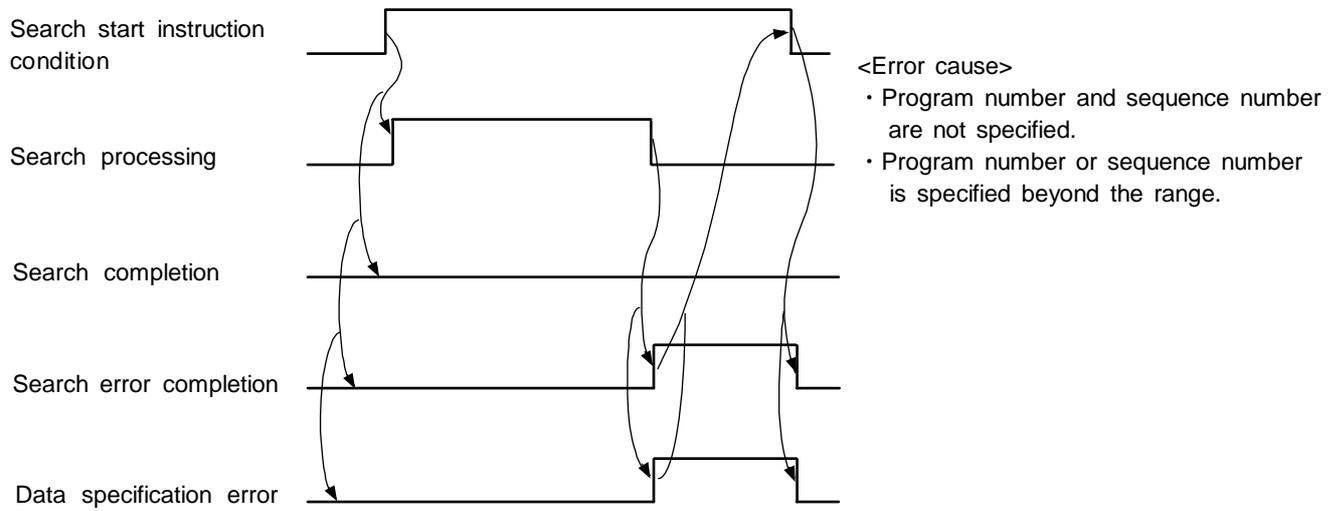
(2) Search error completion



<Error cause>

- The specified program number or sequence number is not found.
- In tape search, tape or I/O device does not exist.
- In tape search, an I/O error occurred.
- The NC operation state is not reset state
Any other search-impossible state.

(3) Search error completion (Data specification error)



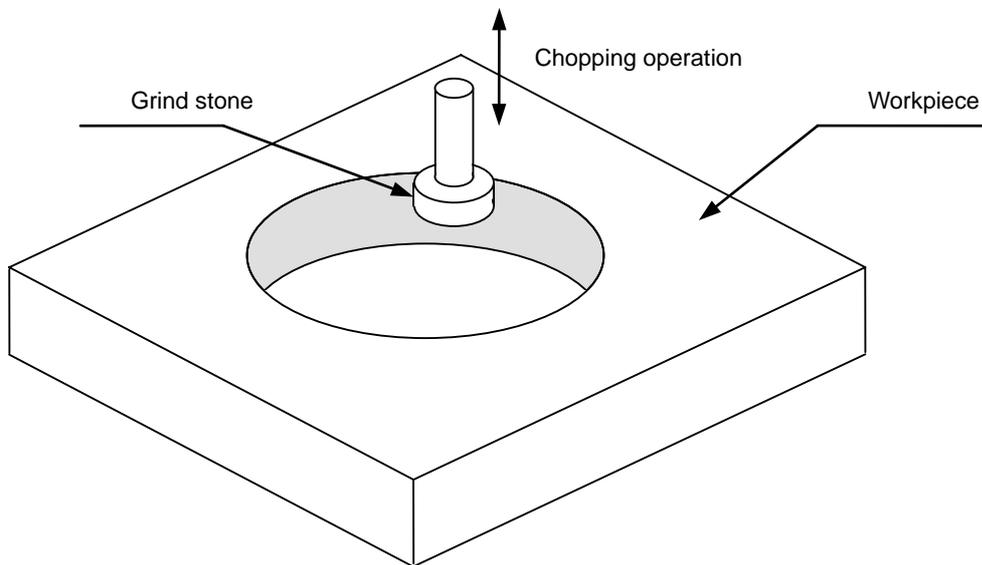
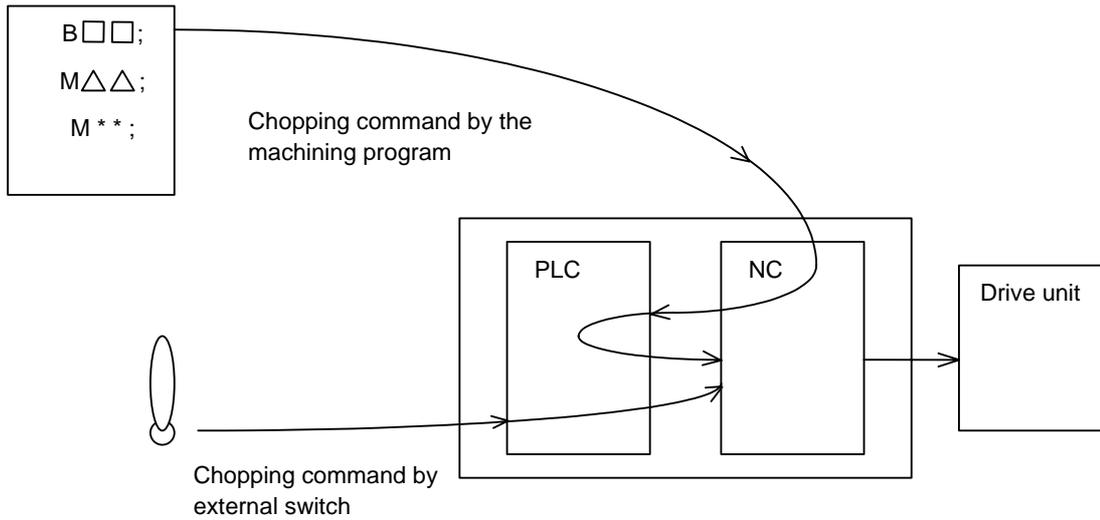
9.6 Chopping

With this function, the chopping axis constantly moves back and forth independently of the program operation during executing the program.

By applying chopping, higher surface accuracy can be achieved than that of abrasive grain.

The chopping operation is started/stopped by the "Chopping" signal from the PLC.

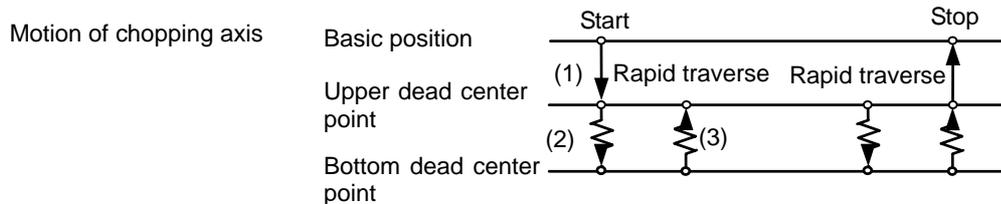
When the chopping operation is commanded from the machining program, use the auxiliary instruction (M or B) codes.



9.6.1 Chopping operation start

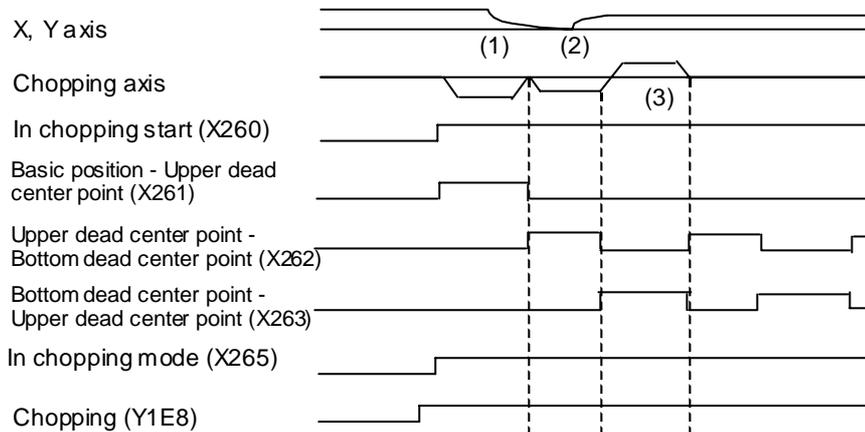
The chopping mode is entered at the rising edge of the "Chopping" signal (Y1E8), and the chopping operation is started based on the position determined with the program, etc. The chopping control sequence is the following.

- When the chopping axis is not moving, chopping is started immediately.
- When the chopping axis is moving, chopping is valid from the next block in the automatic mode, and an operation alarm will occur in the manual mode.

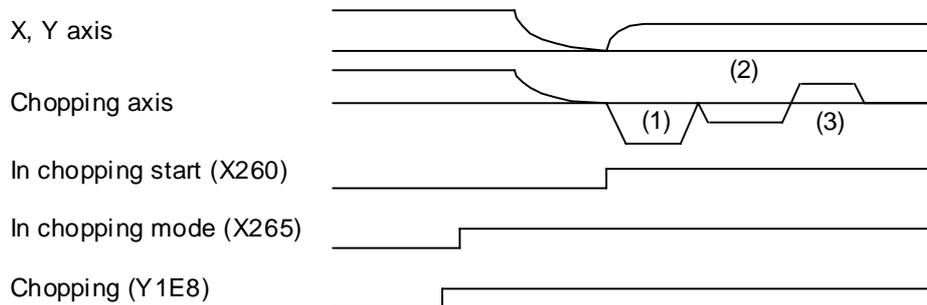


(1) In automatic mode

(a) When the chopping axis is not moving:



(b) When the chopping axis is moving:



The "In chopping start" is entered after the chopping axis movement has been finished.

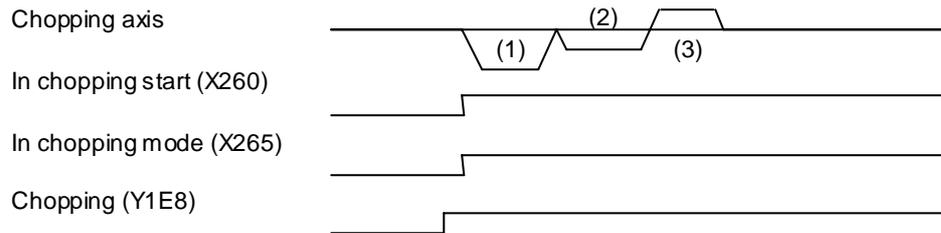
(2) In manual mode

In the jog and step mode, when the chopping axis is not moving, the chopping operation is started at the rising edge of the "Chopping" signal.

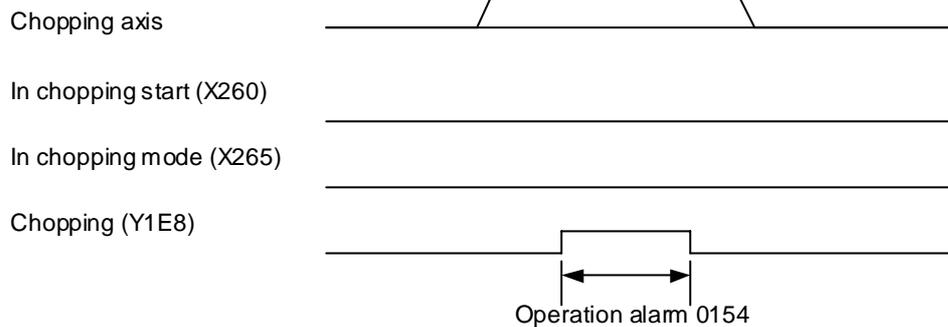
If the "Chopping" signal is turned ON when the chopping axis is moving, the OPERATION ALARM 0154 will occur, and the chopping will not be started.

(Rising edge of the "Chopping" signal is ignored.)

(a) When the chopping axis is not moving:



(b) When the chopping axis is moving:



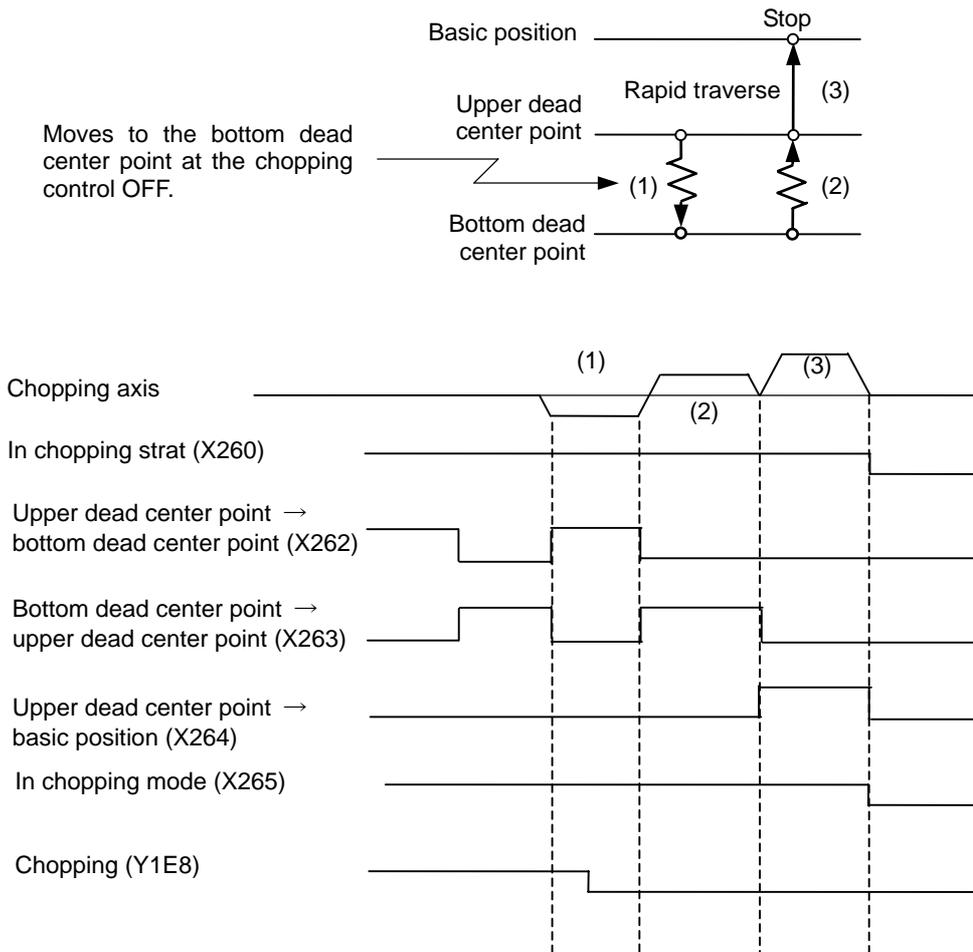
In the handle mode, when the chopping axis is not selected to the handle axis, the chopping operation is started at the rising edge of the "Chopping" signal.

If the "Chopping" signal is turned ON when the chopping axis is selected as a handle axis, the OPERATION ALARM 0154 will occur, and the chopping is not started.

9.6.2 Chopping operation stop

The chopping operation is stopped at the falling edge of the "Chopping" signal from the PLC. The chopping axis moves to the basic position with the rapid traverse after executing the chopping operation to the upper dead center point. The chopping axis once moves to the bottom point even while moving from the upper dead center point to the bottom dead center point.

Stop operation of the chopping axis



The "In chopping start" and the "In chopping mode" signals are turned OFF upon completion of the basic position return.

9.6.3 Chopping compensation

Because this function involves high-speed repetitive motions, the positioning method allowing compensation based on the calculation from the machinery operation (feedback position of the motor end) is adopted, rather than the method using in-position check. Compensation amount used for positioning is calculated every 4 cycles from the start of chopping operation, based on the difference between the commanded position and feedback position. Then the compensation amount is added to the positioning command for the next cycle so that the difference between the commanded position and feedback position will disappear. (Compensation value sequential update method: Refer to Fig.1) However, with this method, if the grindstone contacts with the workpiece, the chopping width before and after compensation may be differed, and which may affect the machining surface. In this case, the compensation value fixed method is appropriate.

With the compensation amount fixed method, compensation amount based on a dry run operation is recorded in advance so that, in the real operation, compensation is carried out from the first positioning to the bottom dead center point using the compensation amount recorded earlier. (Compensation value fixed method: Refer to Fig.2)

Fig.1 Chopping operation in compensation value sequential update method

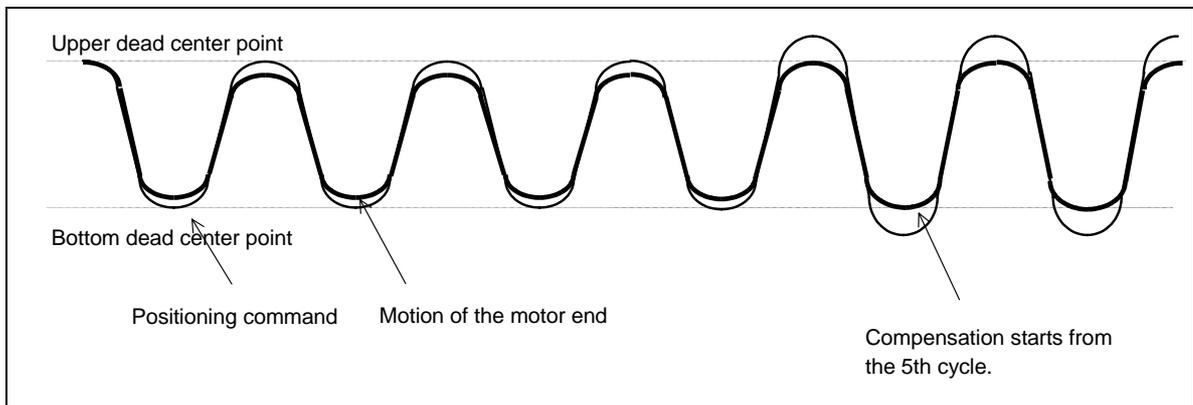
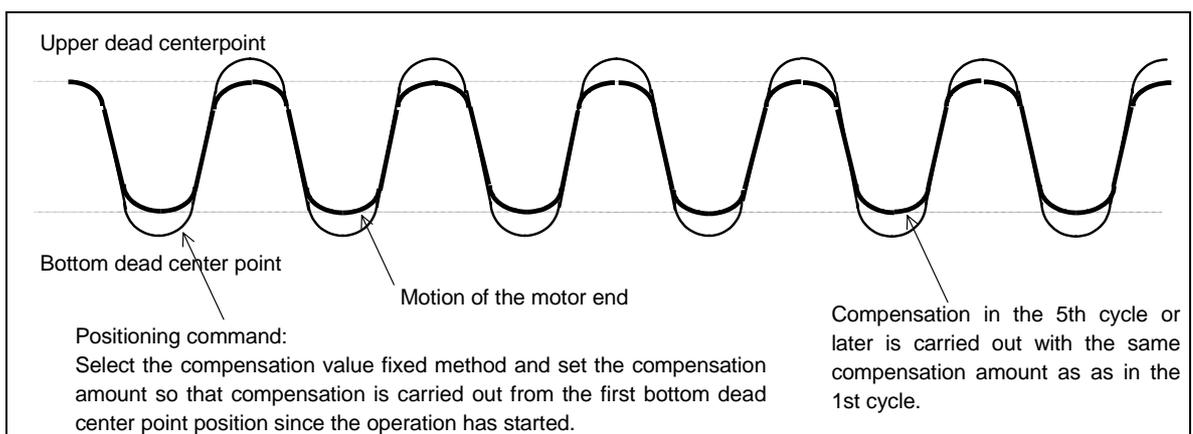


Fig.2 Chopping operation in compensation value fixed method



(1) Compensation value sequential update method

Every chopping command starts with "0" compensation amount. Compensation amount is calculated every 4-cycle chopping operation, and the compensation is carried out.

(2) Compensation value fixed method

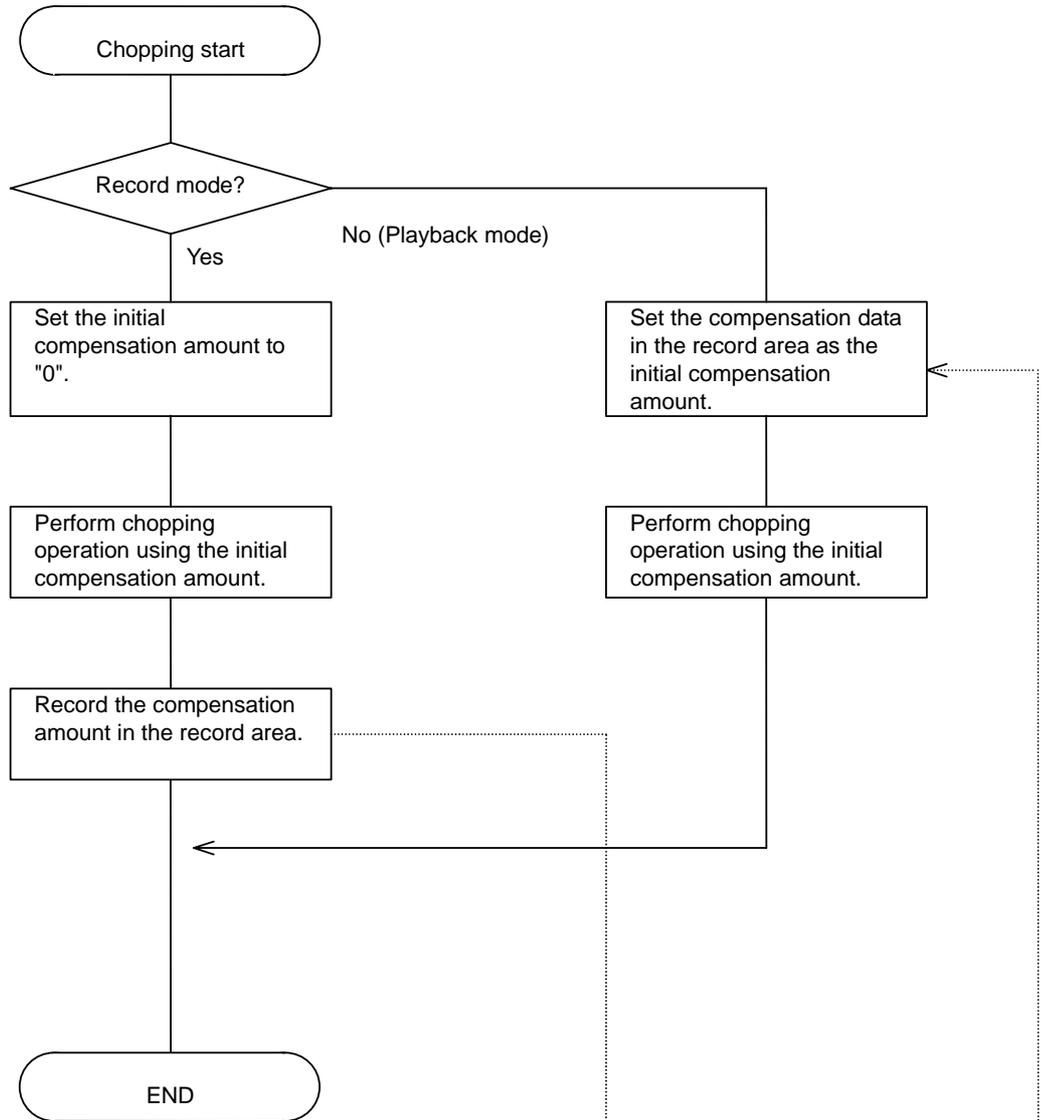
Compensation value fixed method includes the record mode and the playback mode.

<Record mode>

- Override, command axis, upper/bottom dead center point position, number of cycles, and compensation amount are recorded as the chopping control data.
- Compensation amount record area is specified with R register.
- Number of sets for compensation amount record area is determined by the number of R registers to be secured.
14 consecutive R registers are required for 1 set of record.
- Compensation amount is always updated in the record mode.

<Playback mode>

- Chopping operation is started using the data (override, command axis, upper/bottom dead center point position, number of cycles, compensation amount) recorded in the record mode. Compensation amount is not calculated in the playback mode.



Compensation amount record area (R register: For N sets)

| | Error status | Completion status | Command - Feedback | |
|-----|--------------|-------------------|--------------------|--|
| 0 | Override | Axis | Up dead pt. | Bottom dead pt. No. of cycles Compen. amnt (Width) Compen. amnt (Ctr) (Open to user) |
| 1 | Override | Axis | Up dead pt. | Bottom dead pt. No. of cycles Compen. amnt (Width) Compen. amnt (Ctr) (Open to user) |
| N-2 | Override | Axis | Up dead pt. | Bottom dead pt. No. of cycles Compen. amnt (Width) Compen. amnt (Ctr) (Open to user) |
| N-1 | Override | Axis | Up dead pt. | Bottom dead pt. No. of cycles Compen. amnt (Width) Compen. amnt (Ctr) (Open to user) |

14 R registers are required per one set.

For N sets, the number of R registers required is 14*N+4.

9.6.4 Chopping interface

(1) PLC→NC

| Device No. | | Abbreviation | Signal name |
|------------|------|--------------|-------------|
| \$1 | \$2 | | |
| Y1E8 | W068 | CHPS | Chopping |

(2) NC→PLC

In chopping start, intervals during chopping and chopping mode are output.

| Device No. | | Abbreviation | Signal name |
|------------|------|--------------|------------------------------------|
| \$1 | \$2 | | |
| X260 | U0E0 | CHOP | In chopping start |
| X261 | U0E1 | CHP1 | Basic position→upper dead point |
| X262 | U0E2 | CHP2 | Upper dead point→bottom dead point |
| X263 | U0E3 | CHP3 | Bottom dead point→upper dead point |
| X264 | U0E4 | CHP4 | Upper dead point→basic position |
| X265 | U0E5 | CHPMD | In chopping mode |

(3) Chopping override (PLC→NC)

Set within the range of 0% to 100% by 1% increments.

| Device No. | | Abbreviation | Signal name |
|------------|------|--------------|-------------------|
| \$1 | \$2 | | |
| R135 | R335 | CHPOV | Chopping override |

9.6.5 Parameters (DDB function instructions from PLC)

Parameters for chopping functions (DDB function instructions from PLC) are as follows.

<Compensation value sequential update method>

- Rapid traverse override valid/invalid selection
- Chopping axis
- Upper dead center point position L1 (increment from the basic position)
- Bottom dead center point position L2 (increment from the upper dead center point)
- Number of cycles/min

<Compensation value fixed method>

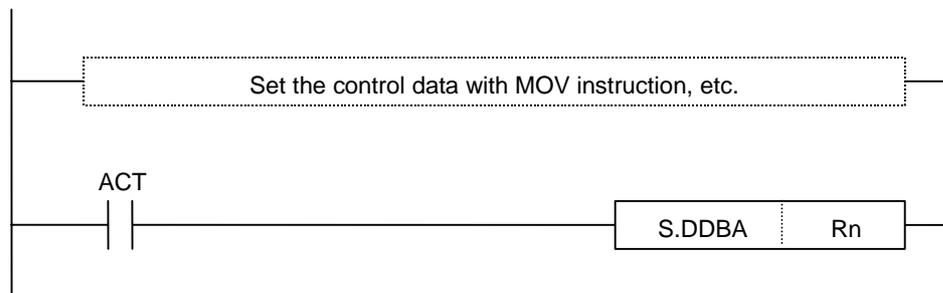
- Mode for the compensation value fixed method
- Data No.

Each parameter can be set from PLC using DDB function.

The master parameter is kept in R register, and when changing parameters, it is read into the current parameter area in the NC by the DDB function instruction.

Parameters can be changed during chopping.

(1) Program example



(Note) Writing parameters from PLC using DDB at every scan execution may cause a longer cycle time because the chopping axis stops once at the bottom dead center point and the upper dead center point even if the value is not to be changed. Thus, change the parameter (turn "ACT" ON) only when necessity of parameter change arose.)

(2) Control data

Data to be used differs depending on whether the compensation value sequential update method is applied or compensation amount fixed method is applied.

Update : Specify with the compensation value sequential update method

Fixed : Specify with the compensation value fixed method

| | | | | | |
|-------|---|----|--|--------|-------|
| Rn | a | a: | Control status (Rn) | Update | Fixed |
| Rn+1 | b | | bit0 : Set to "1". bit1 : Set to "0". bit2 to bit8 : Not used | | |
| Rn+2 | c | | bitF : Error occurred This turns ON if an alarm occurs when the chopping parameter valid signal is turned ON. The details of error is notified with bit9 to C. (Note) | | |
| Rn+4 | d | | bit9 : Chopping error bitA : Chopping specifications is not available | | |
| Rn+5 | e | | bitB : Compensation method is set to other than 0/1 bitC : Multiple chopping axes are specified | | |
| Rn+6 | f | b: | Section No. (Rn+1) | Update | Fixed |
| | | | This sets 0100(HEX). | | |
| | | c: | Sub-section No. (Rn+2[low], Rn+3[high]) | Update | Fixed |
| Rn+8 | g | | 0000(HEX) : Compensation value sequential update method 0001(HEX) : Compensation value fixed method | | |
| | | d: | Rapid traverse override valid/invalid (Rn+4) | Update | |
| Rn+10 | h | | This sets the rapid traverse override valid/invalid in respect to the movement speed between the basic position and the upper dead center point. 0 : Invalid 1 : Valid | | |
| Rn+12 | i | e: | Chopping axis designation (Rn+5) | Update | |
| Rn+13 | j | | bit0 : 1st axis bit1 : 2nd axis : : bit7 : 8th axis bitC : This sets the part system. 0: 1st part system 1: 2nd part system bit8 to B, bitD to bitF : Not used (Set to "0".) | | |
| | | f: | Upper dead center point (Rn+6[low], Rn+7[high]) | Update | |
| | | | This sets the movement amount of basic position → upper dead center point with the code. Use the setting and display unit (#1003 iunit) for setting. | | |
| | | g: | Bottom dead center point (Rn+8[low], Rn+9[high]) | Update | |
| | | | This sets the distance of upper dead center point → bottom dead center point with the code. Use the setting and display unit for setting. | | |
| | | h: | Number of cycles (Rn+10[low], Rn+11[high]) | Update | |
| | | | This sets the number of cycles for chopping cycle. (Unit: Number of cycles/min) | | |
| | | l: | Operation mode with the compensation value fixed method (Rn+12) | Fixed | |
| | | | 0000(HEX) : Playback mode 0001(HEX) : Record mode | | |
| | | j: | Data No. (Rn+13) | Fixed | |
| | | | This specifies what number data (n-th data) from the head of the record area (specified by the parameter) to be used. (Both the record mode and playback mode must be specified. 1st data area is specified with 0.) | | |

(Note) If an alarm occurs when the chopping parameter valid signal is turned ON, Rn bit is turned ON. Alarm details is output to the chopping error No. (R554), as well.

| Rn bit | Error | Cause |
|--|------------------------------|--|
| BITA BITF | Option error | There is no specification for chopping. |
| BITB BITF | Compensation method error | Compensation method is set to other than 0(Compensation value sequential update type) or 1(Compensation value fixed type). |
| BITC BITF | Illegal number of axes error | Multiple chopping axes are specified by the PLC interface. |
| BIT9 BITF | Chopping error | Part system commanded by PLC interface is not valid. |
| | | Chopping axis is not specified by either PLC interface or parameter. |
| | | Rotary axis is specified as the chopping axis. |
| | | Rapid traverse override valid/invalid is set to other than 0(invalid) or 1(valid). |
| | | Data No. of the control data is a negative value. |
| | | Compensation amount record area exceeds R register backup area (R1900 to R2800). ((Rm+14xN sets+4) > 2800.) |
| | | The mode for the compensation value fixed method is set to other than 0(playback mode) or 1(record mode). |
| | | Number of cycles is 0 or less, or over 1056. (If 0 or less, 1 is applied. If exceeds 1056, 1056 is applied.) |
| | | Acceleration determined by the parameter exceeds clamp/chtL. (The number of cycles is reduced.) |
| | | The chopping axis is changed during chopping operation. (Chopping axis is not changed during chopping.) |
| | | F(feedrate) exceeds the clamp speed. (The speed is clamped to the clamp speed (#2081 chclsp).) |
| Chopping axis's #2081 chclsp (chopping clamp speed) and #2002 clamp (cutting clamp speed) are both set to "0". | | |

The error bit shown above is not turned ON in the following cases. However, chopping error No. is output.

- Control data area exceeds the R register area designated for the control data.
- Control data area and compensation amount record area are overlapped.

(3) Compensation amount record area (Dedicated for compensation value fixed method)
 Rm is specified with the parameter (#1324 chop_R).

| | | | |
|-------|--|----|--|
| Rm | a | a: | Error status (in playback mode) (Rm) bit0 : This is turned ON when the difference between the commanded stroke and the actual stroke has exceeded the tolerance set with the parameter (#2080 chwid). |
| Rm+1 | b | b: | Chopping compensation amount record completion status (Rm+1) (in record mode) bit0 : "1" at completion of recording bit1 : "1" when recording is not completed |
| Rm+2 | c | c: | Command - Feedback (Rm+2[low], Rm+3[high]) [In playback mode] Difference between the commanded stroke and the actual stroke is stored when the difference has exceeded the tolerance set with the parameter (#2080 chwid). [In record mode] Difference between command and feedback is stored every time the compensation amount is calculated. |
| Rm+4 | d | d: | Rapid traverse override valid/invalid (Rm+4) Set the rapid traverse override valid/invalid in respect to the movement speed between the basic position and upper dead center point. 0 : Invalid 1 : Valid |
| Rm+5 | e | e: | Chopping axis designation (Rm+5) bit0 : 1st axis Select any one of the existing axes using bit. bit1 : 2nd axis When no axis is specified, the axis whose base specification parameter "chop_ax" is "1" (the smallest No. of axis) within the same part system is selected. : : bit7 : 8th axis bitC : This sets the part system. 0 : 1st part system 1 : 2nd part system bit8 to B, bitD to bitF : Not used (Set to "0".) |
| Rm+6 | f | f: | Upper dead center point (Rm+6[low], Rm+7[high]) Set the movement amount of basic position → upper dead center point with the code. Use the setting and display unit (#1003 iunit) for setting. |
| Rm+8 | g | g: | Bottom dead center point (Rm+8[low], Rm+9[high]) Set the distance of upper dead center point → bottom dead center point with the code. Use the setting and display unit for setting. |
| Rm+10 | h | h: | Number of cycles (Rm+10[low], Rm+11[high]) Set the number of cycles for chopping cycle. (Unit: Number of cycles/min) |
| Rm+12 | i | i: | Compensation amount (Width) (Rm+12[low], Rm+13[high]) Compensation amount to be added to the chopping upper/bottom dead center point command. In the playback mode, this is used for amplitude compensation. When started with the record mode, this is automatically stored. |
| Rm+14 | j | j: | Compensation amount (Center) (Rm+14[low], Rm+15[high]) Compensation amount to be added to the chopping upper/bottom dead center point command. In the playback mode, this is used for compensating the center of amplitude. When started with the record mode, this is automatically stored. |
| Rm+16 | k | k | Data to be opened (Rm+16[low], Rm+17[high]) Use this for managing the compensation amount record area, etc. by using ladder of the user. |
| Rm+18 | Repeat the same setting as in Rm+4 to Rm+17 hereafter. | | |
| Rm+19 | | | |
| Rm+20 | | | |
| Rm+22 | | | |
| | | | |
| | | | |

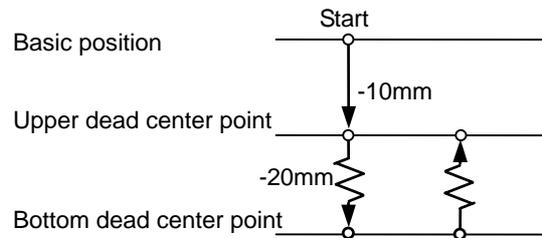
(4) Setting example for the compensation value sequential update method

The following parameters are set using R2000 to R2011 as DDB buffer.

| Parameter | Decimal | HEX | Setting details |
|--|---------|----------|-----------------------------------|
| Rapid traverse override valid/invalid | 1 | 0001 | Valid |
| Chopping axis designation | 4 | 0004 | 1st part system Z axis (3rd axis) |
| Upper dead center point (increment amount from the basic position) | -10000 | FFFFD8F0 | -10000 (Output unit) |
| Bottom dead center point (increment amount from the upper dead center point) | -20000 | FFFFB1E0 | -20000 (Output unit) |
| Number of cycles | 50 | 00000032 | 50/min |

| | | |
|-------|--------------|-------------------------------|
| R2000 | 0001 | Control signal |
| R2001 | 0000 | Section No. |
| R2002 | 0000 0000 | Sub-section No. |
| R2004 | 0001 | Rapid traverse override valid |
| R2005 | 0004 | Chopping axis designation |
| R2006 | D8F0 FFFF | Upper dead center point |
| R2008 | B1E0 FFFF | Bottom dead center point |
| R2010 | 0032 0000 | Number of cycles |

Chopping axis operation

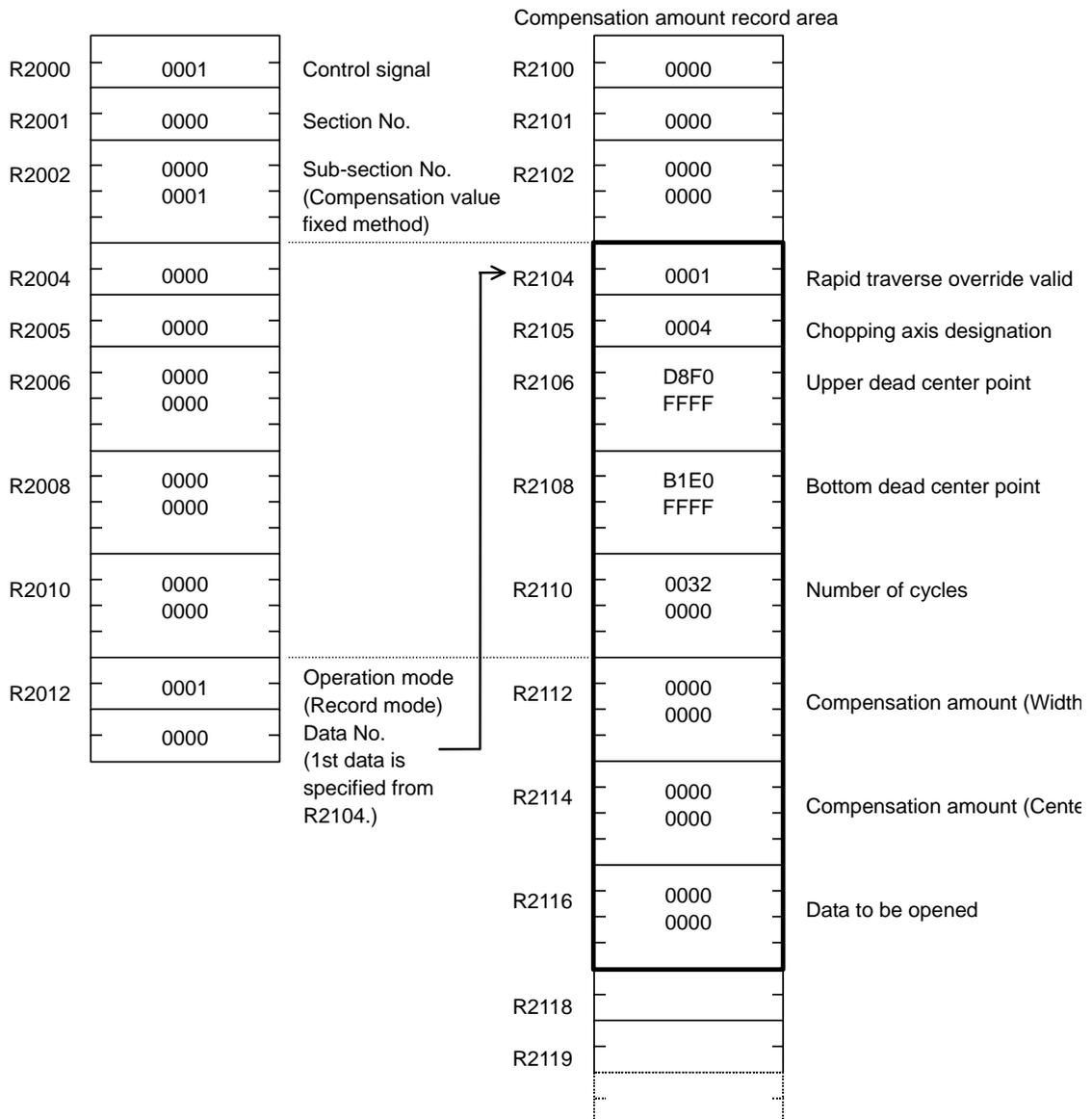


(5) Setting example for the compensation value fixed method

The following parameters are set using R2000 to R2011 as DDB buffer.

R2100 (#1324 chop_R = 2100) is used for the compensation amount record area.

| Parameter | Decimal | HEX | Setting details |
|--|---------|----------|-----------------------------------|
| Rapid traverse override valid/invalid | 1 | 0001 | Valid |
| Chopping axis designation | 4 | 0004 | 1st part system Z axis (3rd axis) |
| Upper dead center point (increment amount from the basic position) | -10000 | FFFFD8F0 | -10000 (Output unit) |
| Bottom dead center point (increment amount from the upper dead center point) | -20000 | FFFFB1E0 | -20000 (Output unit) |
| Number of cycles | 50 | 00000032 | 50/min |



9.6.6 Example of chopping control by program command

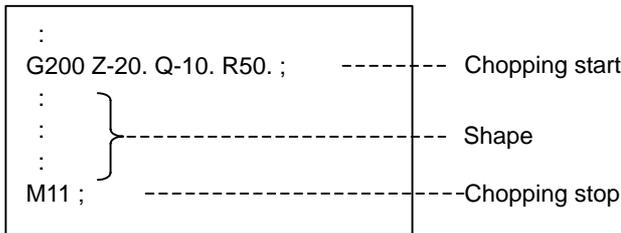
In the example given below, the upper dead center point (increment from the basic position), bottom dead center point (increment from the upper dead center point), and number of cycles (times/min) are set using G code macro.

The above data is set to the local variables by G code macro. The local variable data is read by the ladder upon execution of M code (M10). Then, chopping is started upon DDB function instruction. The chopping is stopped by the ladder upon execution of M code (M11).

(1) G code macro execution

The following is an example in which O9000 is defined as the sub-program of G200 (G65 macro type).

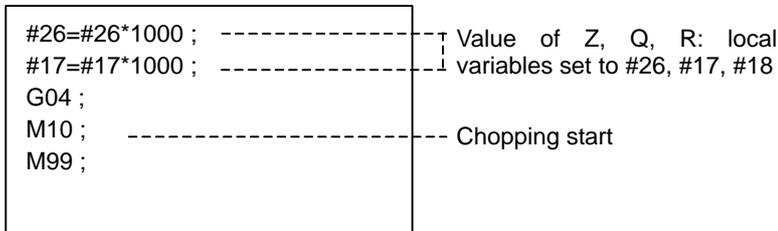
Main program



Argument of G200

- Z : Upper dead center point (Increment from the basic position)
- Q : Bottom dead center point (Increment from the upper dead center point)
- R : Number of cycles/min.

O9000

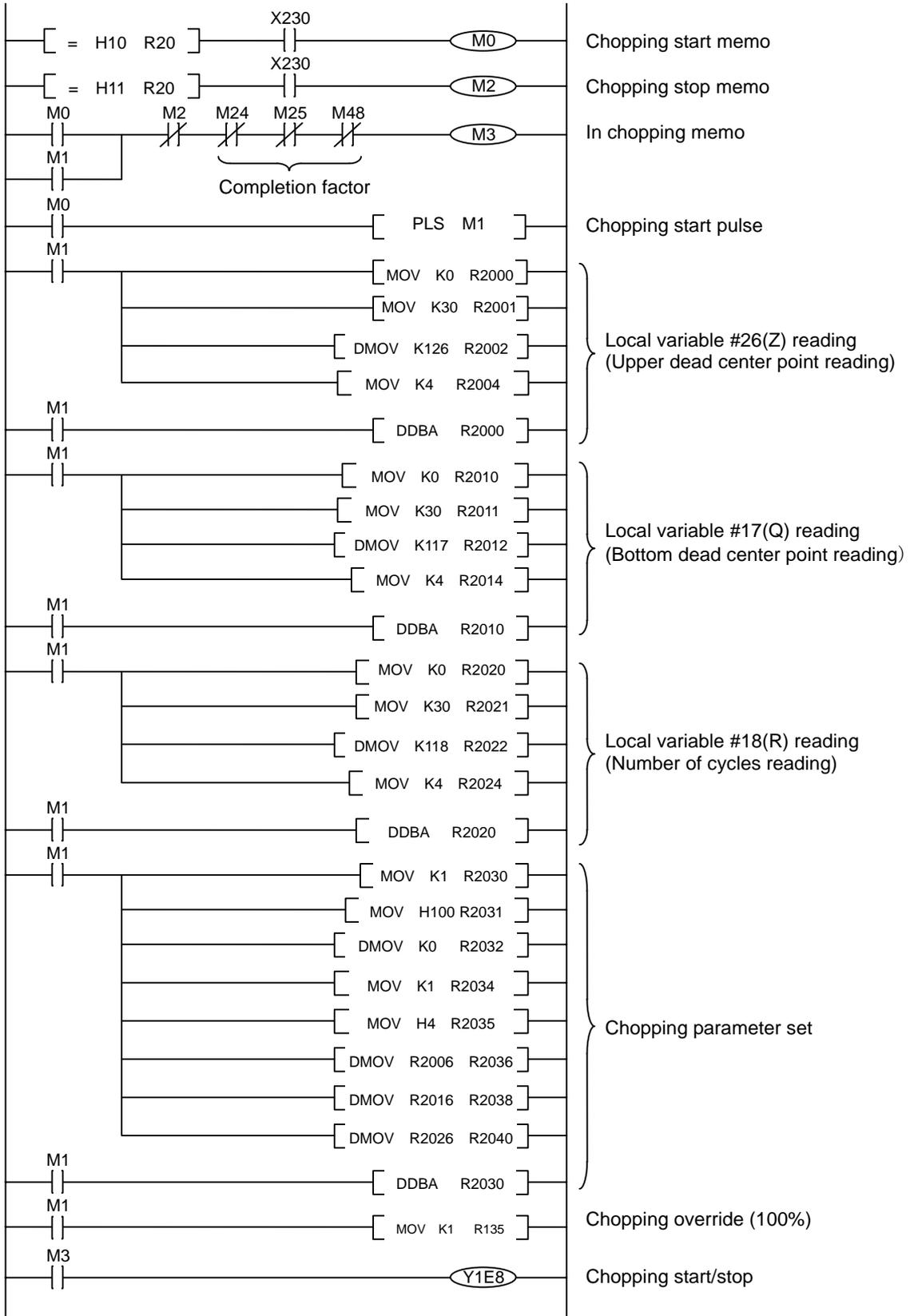


(Note 1) As for Z, Q commands, even if a decimal place is omitted (Ex. Z-20. → Z-20), the unit is remained mm.

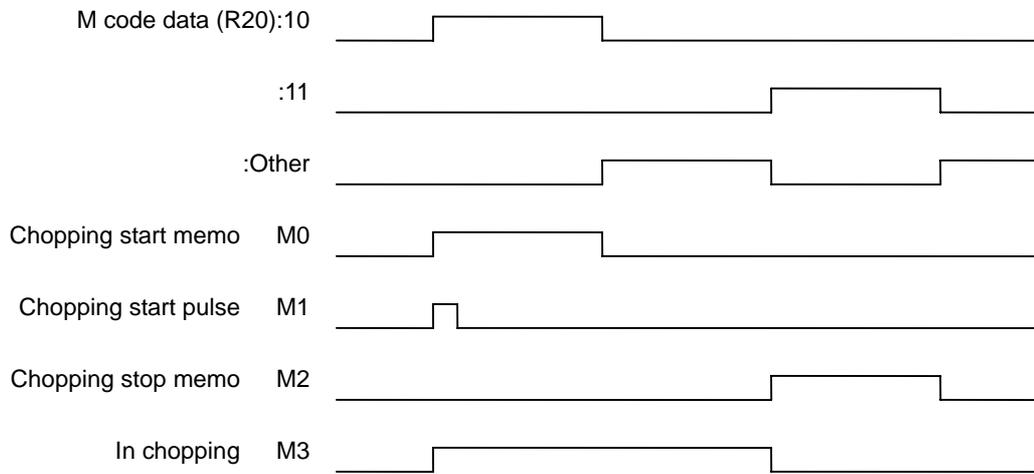
(Note 2) With the submicron system, change the constant for macro operation from 1000-fold to 10000-fold.

(Note 3) When a macro call is executed, the nesting level of local variable will be 1, and the level of local variable will also be 1. So, the number of layers of nesting has to be kept to 4.

(2) Set the local variables of (1) for chopping parameters by using DDB function, and start the chopping operation. The following is its sequence example. (Compensation value sequential update method)



Sequence example timing chart



- (Note 1) Chopping axis cannot be specified as a synchronous control axis.
- (Note 2) Chopping function can be applied to only one axis per part system.

9.7 CC-Link

NC unit can be directly connected to the network to serve as the master/local station of the MELSEC CC-Link.

To enable this connection, the CC-Link master/local units (HR576) must be installed in the expansion slot.

When using this function, the user PLC ladder type must be the DX Developer type.

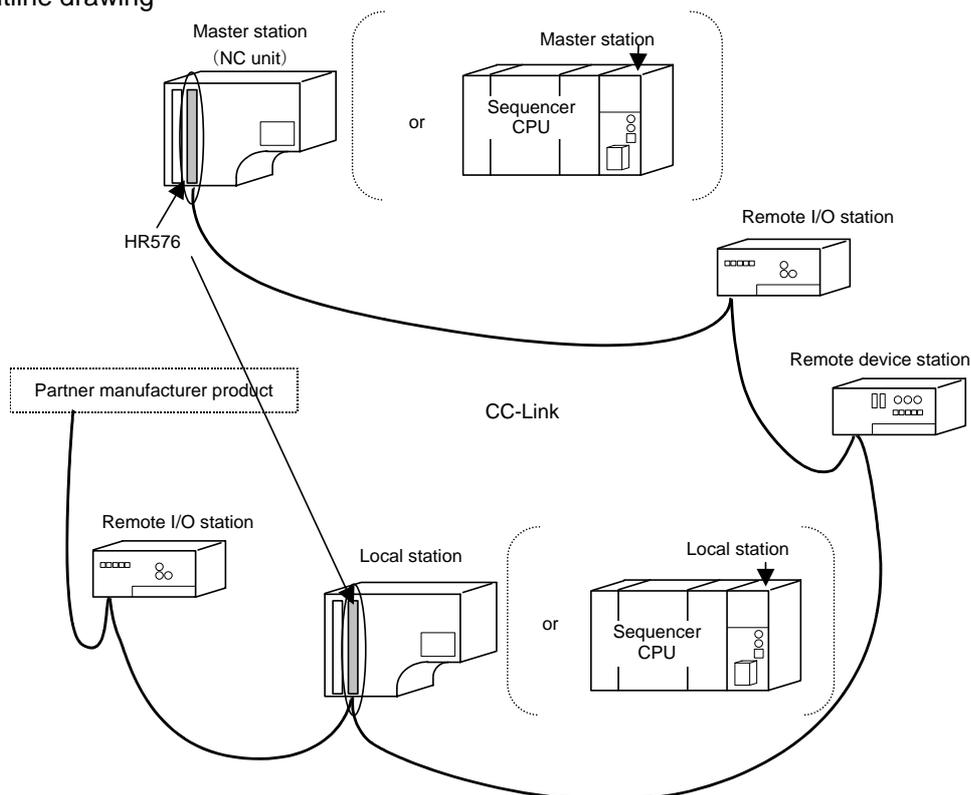
With this function, the transient instruction with the MELSEC A series cannot be used.

When connected with GOT, set so that GOT serves as a remote device station. (Cannot be set as an intelligent device station.)

(1) Outline of CC-Link

- Distributing and installing each unit to the equipments such as conveyor line and mechanical device can simplify the wiring of the whole system.
- The ON/OFF data and numerical data such as input/output treated by each unit can be communicated easily and at high speed.
- The simple distribution system can be established by connecting several sequencer CPUs or NCs.
- Connecting the device equipments made by the partner manufacturer can flexibly support various systems.

(2) Outline drawing



| | |
|----------------------------|---|
| Master station | This station controls the remote station and local station. One master station is required for one system. |
| Local station | This station contains the CPU and can communicate with the master and the other local stations. |
| Remote I/O station | Remote station that handles only bit information. |
| Remote device station | Remote station that handles bit information and word information. |
| Intelligent device station | This station allows the transient transmission. (Including local station) |

9.7.1 Input output signal

Details of input/output signals are shown below.

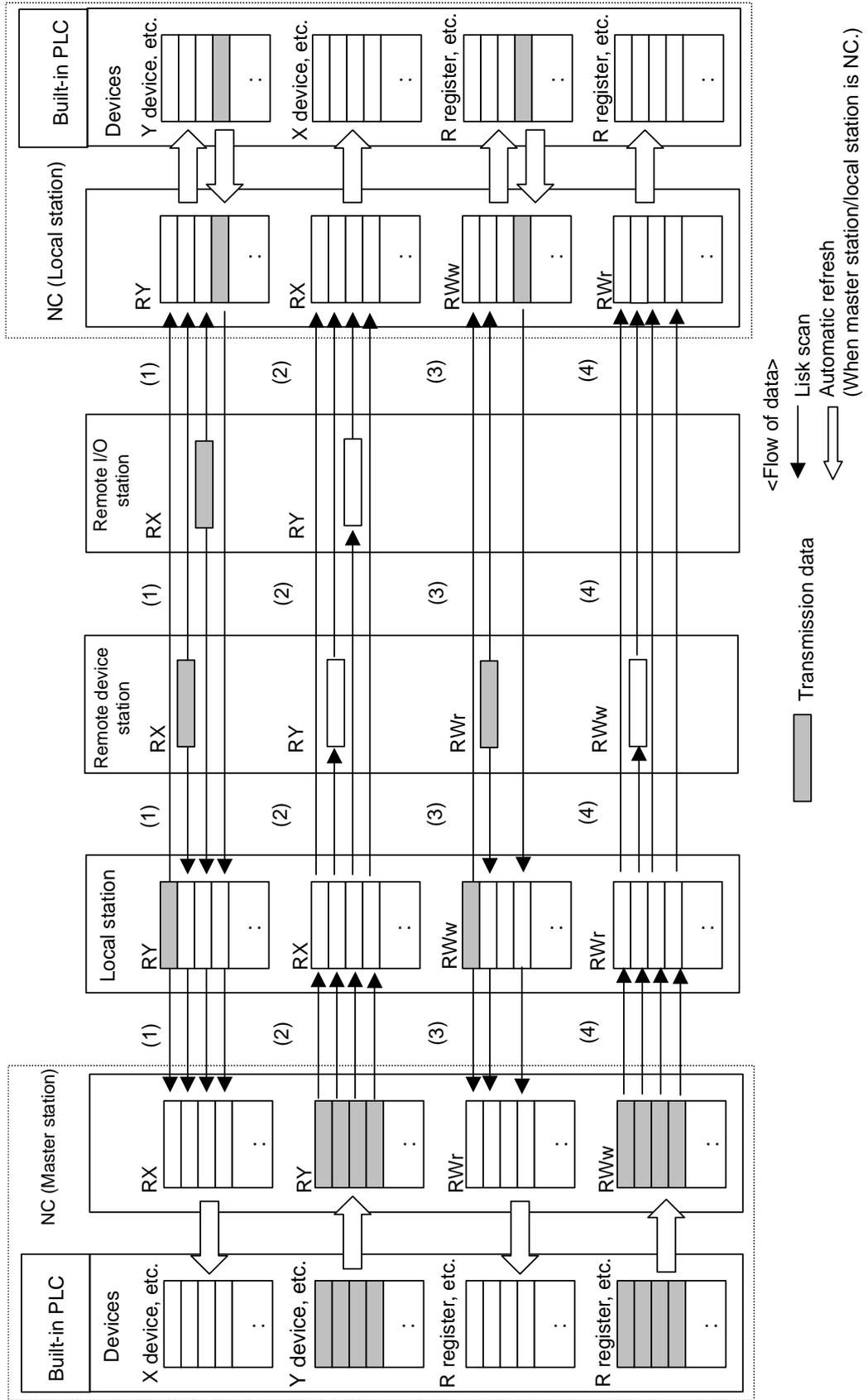
| NC ← Master/local unit (HR576) | | | | NC → Master/local unit (HR576) | | | |
|--------------------------------|---|----------------|---------------|--------------------------------|---|----------------|---------------|
| Input No. | Signal name | Usability | | Output No. | Signal name | Usability | |
| | | Master station | Local station | | | Master station | Local station |
| X480 | Unit error | ○ | ○ | Y500 | Refresh instruction | ○ | ○ |
| X481 | Data link state at host station | ○ | ○ | Y501 | (Prohibited to use) | - | - |
| X482 | Parameter setting status | ○ | × | Y502 | | | |
| X483 | Data link status at other station | ○ | ○ | Y503 | | | |
| X484 | Unit reset acceptance completed | ○ | ○ | Y504 | Unit reset request | ○ | ○ |
| X485 | (Prohibited to use) | - | - | Y505 | (Prohibited to use) | - | - |
| X486 | Data link startup normal completion | ○ | × | Y506 | Data link start request | ○ | × |
| X487 | Data link startup error completion | ○ | × | Y507 | (Prohibited to use) | - | - |
| X488 | Data link startup by E ² ROM parameter normal completion | × | × | Y508 | Data link startup request from E ² ROM parameter | × | × |
| X489 | Data link startup by E ² ROM parameter error completion | × | × | Y509 | (Prohibited to use) | - | - |
| X48A | Parameter registration to E ² ROM normal completion | × | × | Y50A | Parameter registration request to E ² ROM | × | × |
| X48B | Parameter registration to E ² ROM error completion | × | × | Y50B | (Prohibited to use) | - | - |
| X48C | (Prohibited to use) | - | - | Y50C | | | |
| X48D | | | | Y50D | | | |
| X48E | | | | Y50E | | | |
| X48F | Unit ready | ○ | ○ | Y50F | (Prohibited to use) | - | - |
| X490 | (Prohibited to use) | - | - | Y510 | | | |
| X491 | | | | Y511 | | | |
| X492 | | | | Y512 | | | |
| X493 | | | | Y513 | | | |
| X494 | | | | Y514 | | | |
| X495 | | | | Y515 | | | |
| X496 | | | | Y516 | | | |
| X497 | | | | Y517 | | | |
| X498 | | | | Y518 | | | |
| X499 | | | | Y519 | | | |
| X49A | | | | Y51A | | | |
| X49B | | | | Y51B | | | |
| X49C | | | | Y51C | | | |
| X49D | | | | Y51D | | | |
| X49E | | | | Y51E | | | |
| X49F | Y51F | | | | | | |

○ : Usable × : Not usable

To use the CC-Link function, turn the "refresh instruction" command (Y500) ON after starting up the NC. When the NC is the master station, turn the "data link start" command (Y506) ON.

9.7.2 Communication data flow

The flow of data communicated by the CC-Link's link scan is as follows.
 (The master station and local station of MELSEC CPU can be also mixed.)



- (1) By executing a link scan, data in the remote device station's remote input (RX) and in the local station's remote output (RY) is transmitted to the master station's remote input (RX) and the local station's remote output (RY).
- (2) By executing a link scan, data in the master station's remote output (RY) is transmitted to the remote I/O station and remote device station's remote output (RY) and the local station's remote input (RX).
- (3) By executing a link scan, data in the remote device station's remote register (RW_r) and the local station's remote register (RW_w) is transmitted to the master station's remote register (RW_r) and the local station's remote register (RW_w).
- (4) By executing a link scan, data in the master station's remote register (RW_w) is transmitted to the remote device station's remote register (RW_w) and the local station's remote register (RW_r).

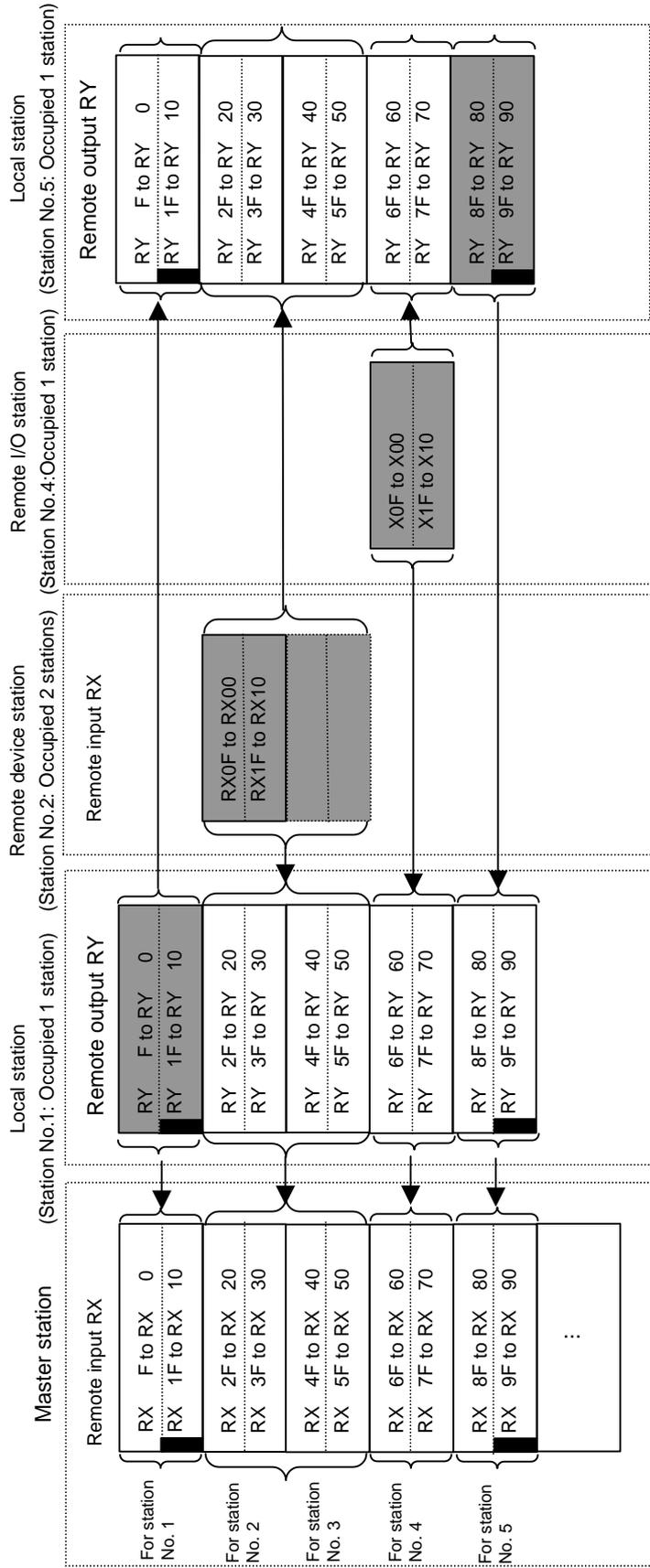
9.7.2.1 Remote input and remote output (Master station ← local station/remote device station/remote I/O station)

(1) Master station

- Status of input from the local station(RY), remote device station and remote I/O station (RX) is stored.
- Two words are used per station.

(2) Local station

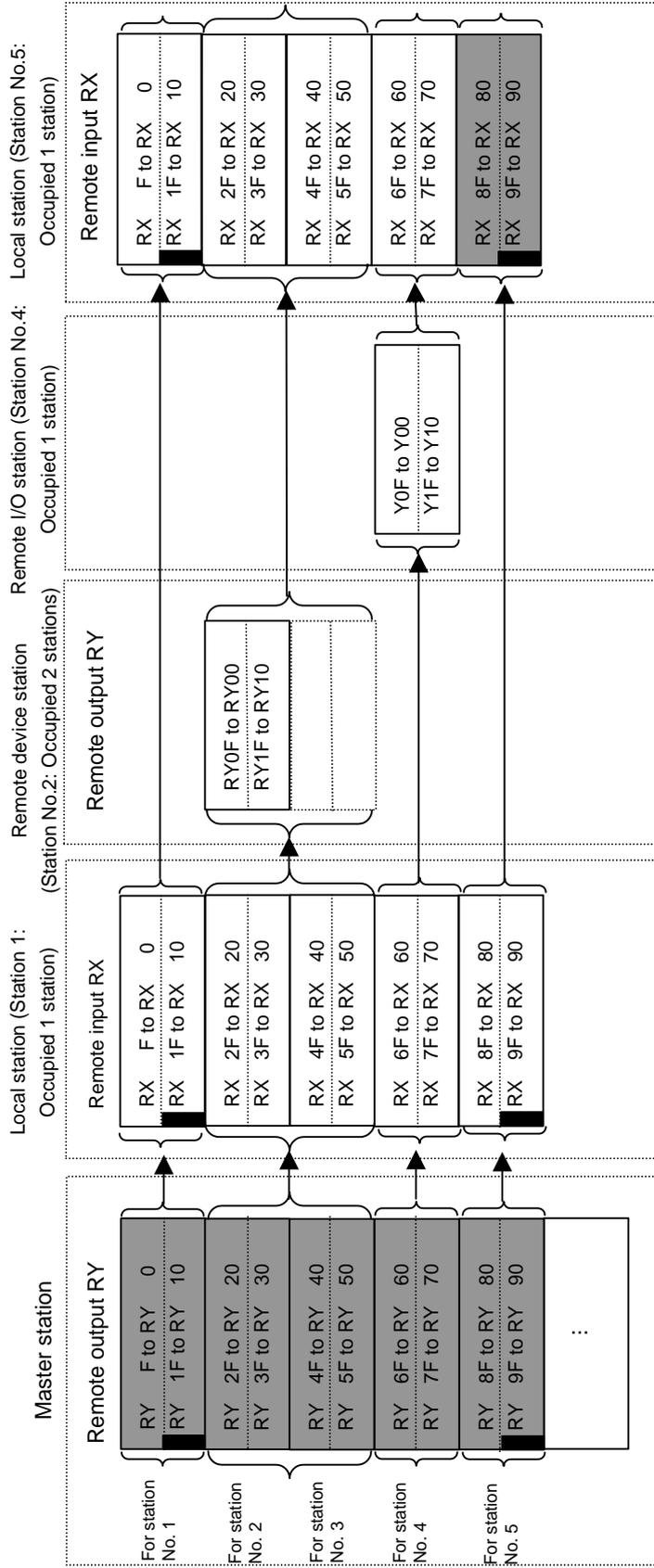
- The data to be transmitted to the master station is stored in the remote output (RY) that is corresponding to the self-station.
- Status of input from the remote device station, remote I/O station (RX) and the other local stations is stored.
- Two words are used per station.



••• The last two bits of the local station cannot be used.

9.7.2.2 Remote output and remote input (Master station → local station/remote device station/remote I/O station)

- (1) Master station
 - Status of output to the remote device station, remote I/O station (RY) and all the local station (RX) is stored.
 - Two words are used per station.
- (2) Local station
 - Data received from the remote device station, remote I/O station (RY), master station (RY) is stored.
 - Two words are used per station.



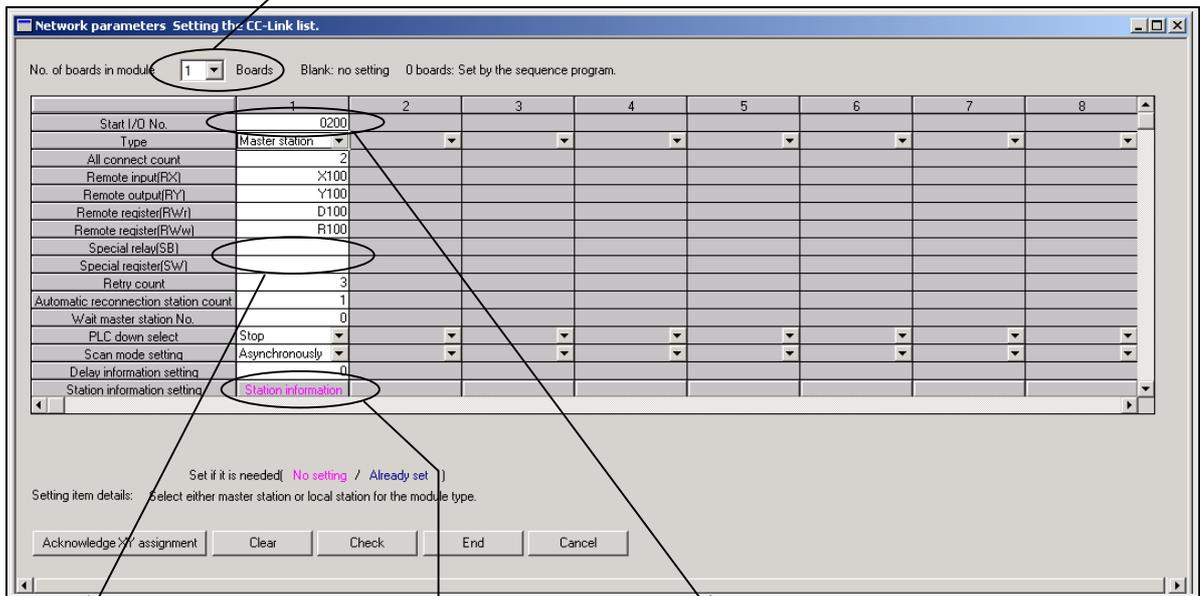
••• The last two bits of the local station cannot be used.

9.7.3 Automatic Refresh

Data is automatically transmitted between the CC-Link master/local unit (HR576) and NC built-in PLC device. The transmission size and destination device are set with the parameters using GX Developer. For the master station, parameters for station information must be set. Parameters cannot be set with the PLC program.

(1) GX Developer setting example

Set "1" for No. of board in module.
(Maximum number of CC-Link cards that can be mounted on the NC is one.)



When there is no setting, SB and SW devices are not automatically refreshed.

Set station information. Refer to (2) for the setting example.

Always set 200 to the head I/O No. (Same setting applied for any connection slot.)

The devices that can be set as the transmission destination for automatic refresh are as follow.

| Device name | RX,RY,SB | RWr,RWw,SW |
|-------------|------------|------------|
| X | ○(RX only) | × |
| Y | ○(RY only) | × |
| M | ○ | ○ |
| L | ○ | ○ |
| D | ○ | ○ |
| R | ○ | ○ |

○: Possible ×: Not possible

(2) Example of station information designation

CC-Link station information. Module 1

| StationNo. | Station type | Exclusive station count | Reserve/invalid station select | Intelligent buffer select(word) | | |
|------------|----------------------------|-------------------------|--------------------------------|---------------------------------|---------|-----------|
| | | | | Send | Receive | Automatic |
| 1/ 1 | Remote I/O station | Exclusive station 1 | No setting | | | |
| 2/ 2 | Intelligent device station | Exclusive station 4 | No setting | 64 | 64 | 128 |

Default Check End Cancel

| Item | Details |
|----------------------------------|--|
| Station type | When connecting NC or MELSEC sequencer as a local station, connect with the intelligent device station. |
| Exclusive station count | Sets the number of occupied stations specified with the condition setting switch. (SW5). (1 to 4 stations) |
| Reserve/invalid station select | This sets when specifying either reserved station or invalid station. To make it function as CC-Link, specify "no setting". Reserved station: Set the unconnected station (station to be connected) as a reserved station so that it is not handled as an error station. If the connected remote station/local station is set as reserved station, the designated remote station/local station will be disabled for data link. Invalid station: Set the station which is incapable of data link due to power OFF, etc. as an invalid station so that it is not handled as "data link error station" at the master station and local station. Note that however, error detection will be disabled at this time. |
| Intelligent buffer select (word) | Specifies the assignment of buffer memory size for when executing transient transmission to the intelligent device station. Specify the size where 7 words were added to the size of data to be sent/received for the transmission/reception buffer. In the example above, up to 57 words can be sent/received by the transient transmission. Automatic update buffer has to be assigned in an appropriate size per intelligent device. |

9.7.4 Occupied number of stations of the system and settable range of the device

The device range allocated for CC-Link remote I/O (RX, RY) and remote register (RWw, RWr) varies depending on the number of occupied stations (actual number of link points) set per system. In order to operation NC and CC-Link, set within the designated range.

| RX,RY,SB | RWr,RWw,SW | Settable device name | Settable device range |
|----------|------------|----------------------|--|
| RX only | — | X | <Area 1> Min. : X0 Max. : X140 -actual number of link points <Area 2> Min. : X640 Max. : X740 - actual number of link points |
| RY only | — | Y | <Area 1> Min.: Y0 Max.: Y140 - actual number of link points <Area 2> Min.: Y740 Max.: Y840 - actual number of link points |
| ○ | ○ | M | Min.: M0 Max.: M8192 - actual number of link points |
| ○ | ○ | L | Min.: L0 Max.: L256 - actual number of link points |
| ○ | ○ | D | Min.: D0 Max.: D1024 - actual number of link points |
| ○ | ○ | R | <Area 1> Min.: R500 Max.: R550 - actual number of link points <Area 2> Min.: R1900 Max.: R2800 – actual number of link points |

<Example> Setting range when the occupied number of stations is 30 per one CC-Link system:

Actual number of remote input/output points (actual number of link points): Occupied number of stations(30)*32=960 points

Setting range of RX and RY when they are set (in the order of RX, and then RY) with M devices:

RY device: M0 to M7232 (M8192 – actual number of link points(960)=M7232)

RX device: M0 to M6272 (M7232 – actual number of link points(960)=M6272)

(Note 1) NC or PLC ladder does not operate normally when the device area secured for CC-Link is duplicated with the actual machine input/output signal used by the NC or machine side or when it is outside the range indicated above. When the NC does not start normally, reduce the number of connections or set the station No. to a smaller one so that the actual number of link points is reduced and the device area falls in the range indicated in the table above. Then, restart the NC and set the CC-Link parameters again.

(Note 2) When changing CC-Link system configuration, always confirm that the parameters of all the NCs connected with CC-Link is within the range.

9.7.5 Transient function

With transient function, data is not transmitted constantly but is only written and read as required among arbitrary stations. The other stations must be compatible with the transient function.

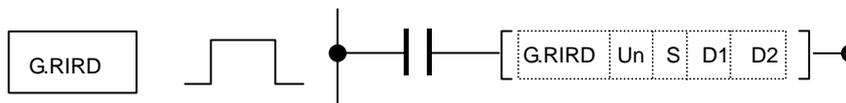
With NC, only RIRD instruction and RIWT instruction are compatible.

Enter the following ladder from the GX Developer only when the ladder program area corresponds to CC-Link. In this case, ladder cannot be edited with the NC ladder edit screen. ("Ladder" menu is hidden.)

Transient function cannot be used with MELSEC A series.

9.7.5.1 Transient instruction (RIRD instruction)

| | Usable device | | | | | | | | | | | | | | |
|----|---------------|---|---|---|---|----|---|---|-------------|---|-----------|---|---------|-------|-------------------|
| | Bit device | | | | | | | | Word device | | Con-stant | | Pointer | Index | Digit designation |
| | X | Y | M | F | L | SM | T | C | D | R | K | H | P | Z | K |
| S | | | | | | | | | ○ | ○ | | | | | |
| D1 | | | | | | | | | ○ | ○ | | | | | |
| D2 | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | |



(1) Setting data

G.RIRD

| Device | Details |
|-------------|--|
| Un (Note 1) | Head input/output No. of the self-station. |
| S | Head No. of the device where control data is stored. |
| D1 | Head No. of the device to store the read data. |
| D2 | Device that is turned ON for one scan upon completion of instruction. (In the case of error completion, D2+1 is turned ON, as well.) |

(Note 1) When executing an instruction from the NC unit, specify U20.

(2) Control data

| Device | Item | Setting data | Setting range | Setting side |
|--------|--------------------------|---|-----------------------------|--------------|
| S + 0 | Completion status | This stores the status at the completion of instruction. 0: No error (Normal completion) Other than 0: Error code | - | System |
| S + 1 | Station No. | This specifies the reading source station No. | 0 to 64 | User |
| S + 2 | Access code | This specifies the reading source device using the access code. | Refer to "(3) Access code". | User |
| S + 3 | Device No. | This specifies the device No. | (Note 2) | User |
| S + 4 | Number of reading points | This specifies the reading size (word unit). | 1 to 480 (Note 3) | User |

(Note 2) Setting range varies depending on the size of device to be read.

(Note 3) Refer to "(2) Example of station information designation" in "9.7.3 Automatic Refresh".

(3) Access code

Specify the reading source (or writing destination) device with the following access codes when executing the transient instruction (RIRD instruction/RIWT instruction).

Low-order 8-bit: Always specify 05H. (If other than 05H is specified, an error occurs.)

High-order 8-bit: Specify a device corresponding to purpose as shown in the table below.

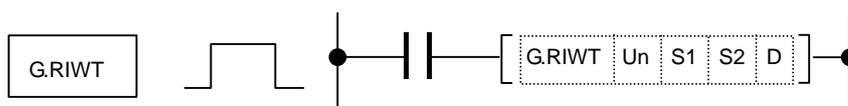
| Device contents | Name | Device type | | Access code (High-order 8-bit) |
|---------------------------------|------|-------------|------|--------------------------------|
| | | Bit | Word | |
| Input relay | X | ○ | | 01H |
| Output relay | Y | ○ | | 02H |
| Inside relay | M | ○ | | 03H |
| Latch relay | L | ○ | | 83H |
| Link relay | B | ○ | | (Not available) |
| Timer (contact) | T | ○ | | 09H |
| Timer (coil) | T | ○ | | 0AH |
| Timer (current value) | T | | ○ | 0CH |
| Retentive timer (contact) | ST | ○ | | (Not available) |
| Retentive timer (coil) | ST | ○ | | (Not available) |
| Retentive timer (current value) | ST | | ○ | (Not available) |
| Counter (contact) | C | ○ | | 11H |
| Counter (coil) | C | ○ | | 12H |
| Counter (current value) | C | | ○ | 14H |
| Data register | D | | ○ | 04H |
| Link register | W | | ○ | (Not available) |
| File register | R | | ○ | 84H |
| Special link relay | SB | ○ | | (Not available) |
| Special link register | SW | | ○ | (Not available) |
| Special relay | SM | ○ | | (Not available) |
| Special register | SD | | ○ | (Not available) |

(Note 1) Devices other than shown above cannot be accessed.

(Note 2) When accessing a bit device, specify it with 0 or a multiple of 16.

9.7.5.2 Transient Instructions (RIWT instruction)

| | Usable device | | | | | | | | | | | | | | |
|----|---------------|---|---|---|---|----|---|---|-------------|---|-----------|---|---------|-------|-------------------|
| | Bit device | | | | | | | | Word device | | Con-stant | | Pointer | Index | Digit designation |
| | X | Y | M | F | L | SM | T | C | D | R | K | H | P | Z | K |
| S1 | | | | | | | | | ○ | ○ | | | | | |
| S2 | | | | | | | | | ○ | ○ | | | | | |
| D | ○ | ○ | ○ | ○ | ○ | ○ | | | | | | | | | |



(1) Setting data
G. RIWT

| Device | Details |
|-------------|--|
| Un (Note 1) | Head input/output No. of the self-station. |
| S1 | Head No. of the device where control data is stored. |
| S2 | Head No. of the device to store the data to be written. |
| D | Device that is turned ON for one scan upon completion of instruction. (In the case of error completion, D+1 is turned ON, as well.) |

(Note 1) When executing an instruction from the NC unit, specify U20.

(2) Control data

| Device | Item | Setting data | Setting range | Setting side |
|--------|--------------------------|--|---|--------------|
| S + 0 | Completion status | This stores the status at the time of instruction completion. 0: No error (Normal completion) Other than 0: Error code | - | System |
| S + 1 | Station No. | This specifies the the writing destination station No. | 0 to 64 | User |
| S + 2 | Access code | This specifies the writing destination device using the access code. | Refer to 9.7.5.1 Transient function (RIRD instruction) "(2) Access code". | User |
| S + 3 | Device No. | This specifies device No. | (Note 2) | User |
| S + 4 | Number of writing points | This specifies the writing size (word unit). | 1 to 480 (Note 3) | User |

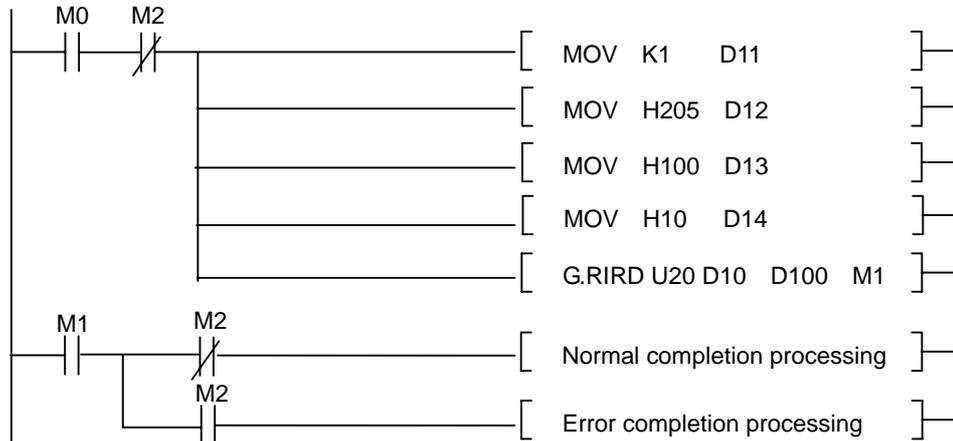
(Note 2) Setting range varies depending on the size of device to be written.

(Note 3) Refer to "(2) Example of station information designation" in "9.7.3 Automatic refresh".

9.7.5.3 Transient instruction program example and error

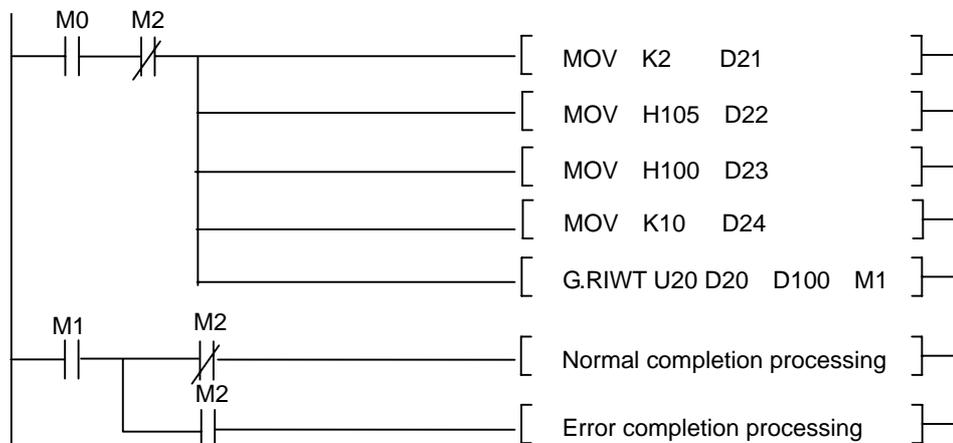
(1) Program example
(Example 1)

This is the program where 16 points of data from Y100 device of the station No.1 intelligent device station connected to the master station (NC) is stored in D100 and after of the master station when M0 is turned ON.



(Example 2)

This is the program to store 10 points of data from D100 of the master station in the station No.2 intelligent device station connected to the master station (NC) when M0 is turned ON.



(2) Error

The followings are the example of error status to be stored in the control data completion status (S+0).

| Error status | Error details |
|--------------|--|
| F114 | Instruction is in execution. (Busy) |
| F110 | Time out |
| 2111 | Invalid card was detected. |
| 4100 | Illegal device was specified. |

9.7.6 Others

(1) Backup of CC-Link related parameters

CC-Link related parameters are the network parameters to be written from GX Developer. These parameters are stored in the ladder program area within NC, which is a different area than that of regular NC parameters.

In order to store these network parameters externally, the data must be output and saved with the following methods.

| Data output operation | Usable device |
|---|----------------|
| With GX Developer: (1) Select "PC" menu. (2) Select "read" menu. (3) Read the selected parameter and store them as a file on PC. (Note) Select "write" menu and follow the same procedures to write parameters. | GX Developer |
| With NC: (1) Select the data output screen. (2) Output the PLC program area. (Output by setting #(99)DATA(ALL3).) (Note) Select the data input screen to write parameters. | RS-232C device |

(2) Replacing the CC-Link card

The CC-Link parameters are stored in the NC. Therefore, if the CC-Link card mounted on the NC must be replaced due to malfunction, etc., the parameters are not necessary to be written again with GX Developer, etc.

(3) Precautions when inputting CC-Link related parameters

When inputting CC-Link related parameters with GX Developer, NC ladder program area must correspond to the CC-Link related parameters.

In order to correspond to the CC-Link related parameters, NC ladder program area must be formatted with the NC system incorporating CC-Link function before writing the ladder into the NC (only once at the initial time). Select the DX Developer "PC memory format" menu when formatting the NC ladder program area. Make sure that the CC-Link card is mounted when formatting.

(Note 1) When outputting CC-Link related parameters or making a PLC program cassette, make sure that the NC ladder program area corresponds to the CC-Link related parameters.

(Note 2) Having formatted the ladder program area, make sure to write the ladder before turning OFF the power. If the power was turned OFF without writing the ladder, carry out formatting again.

10. 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

10.1 Alarm Message Display

There are two types of alarm message, which can be selected with a parameter (described later)

| Format | Alarm message | External alarm message |
|----------------------------|--|---|
| Max. No. of messages | 256 messages | 256 messages |
| Max. data length | 32 bytes per message | 128 bytes per message |
| Number of Display messages | 4 messages | 1 to 4 messages (according to data length) |
| Interface | F type / R type (classification No. designated) | F type / R type (without classification No.) |
| Available language | 2 languages | 8 languages |
| Store method | User PLC attached data | Independent data (other area) |

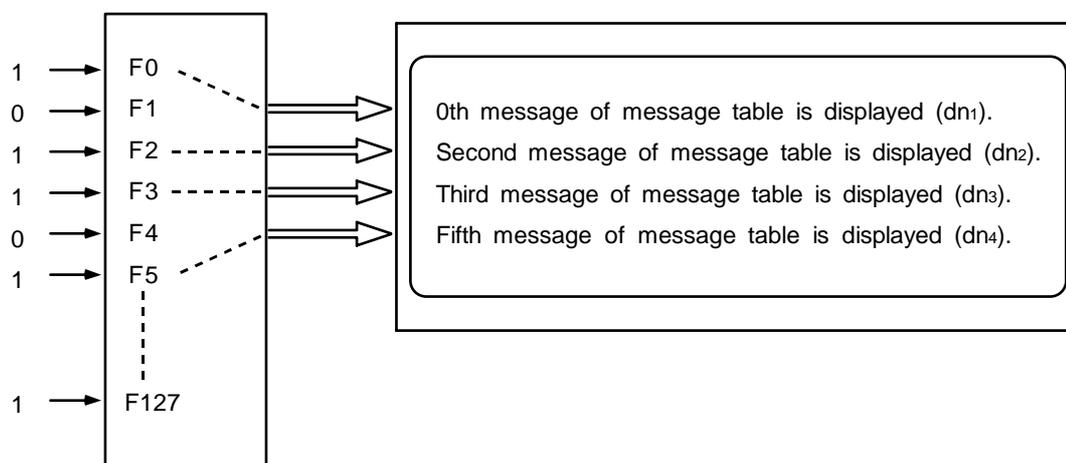
10.1.1 Interface

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 method 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.

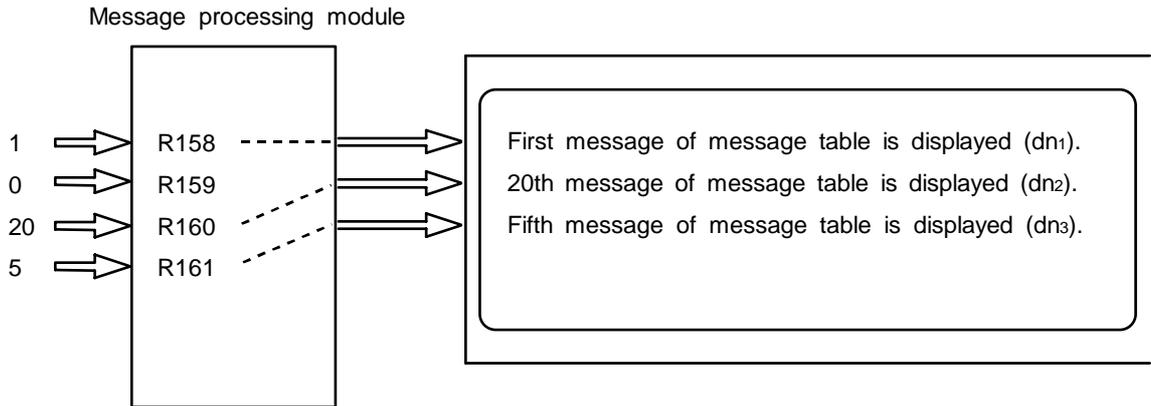


The highest priority is assigned to the F0 signal. The message corresponding to Fn set to 1 is fetched from the message table and displayed in order starting at F0. If no messages are prepared or Fm greater than the number of prepared messages is set to 1, the message "USER PC ERROR m" is displayed.

(2) R type interface

This interface applies to file registers R158~R161. 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 R158 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 (Only for Alarm message type)

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.



(Note 1)

→ This varies depending on the alarm cause or source.

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.

10.1.2 Screen Display

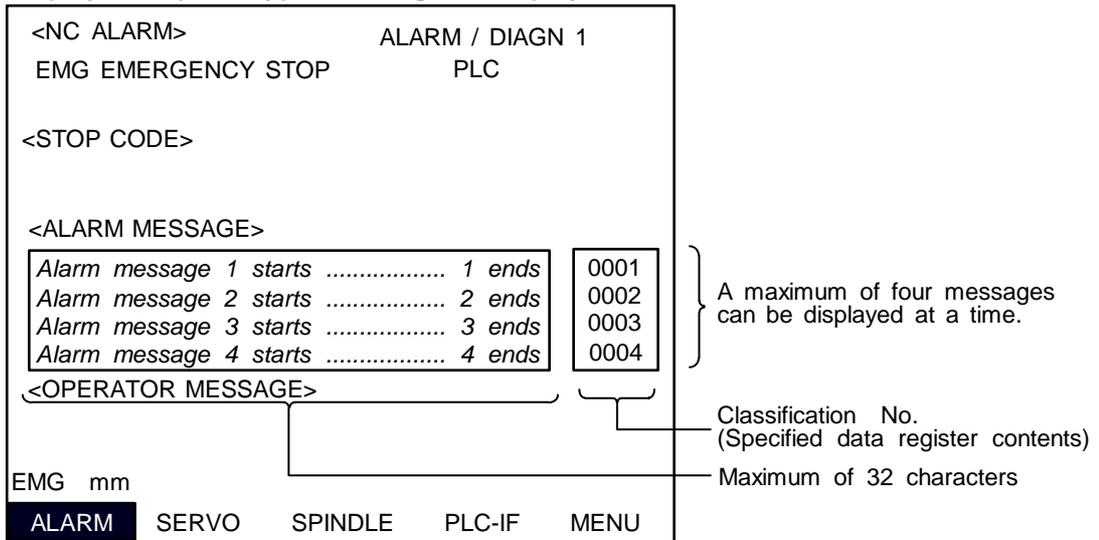
Screen Display depends on the message type as described below.

(1) Alarm message type

Message length is up to 32 characters.

Alarm messages corresponding to four classification Nos. can be displayed.

Display example of type 9 setting and display unit



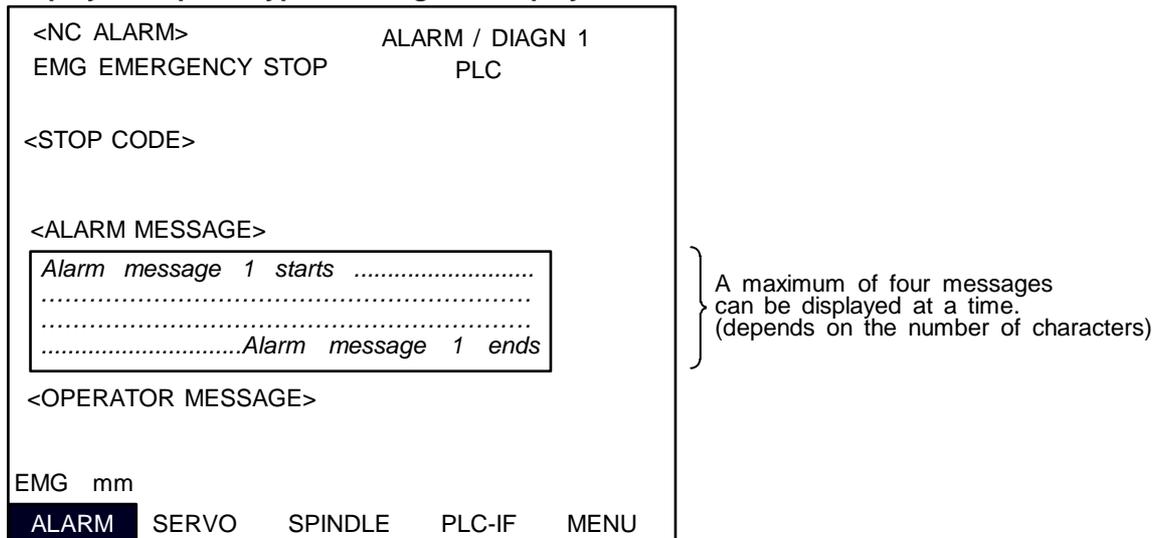
(2) External alarm message type

The contents of data register is not displayed for this type.

Display area is 32-character width and has 4 lines. (Total: 128 characters)

Up to 4 messages can be displayed in the area.

Display example of type 9 setting and display unit



Note that the number of displayed external alarm messages depends on their number of characters.

| Number of External alarm message characters | |
|---|---|
| 0 to 32 characters | 33 to 64 characters |
| 4 messages displayed | 2 messages displayed |
| <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Alarm message 1 starts.....1 ends</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Alarm message 2 starts.....2 ends</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Alarm message 3 starts.....3 ends</div> <div style="border: 1px solid black; padding: 2px;">Alarm message 4 starts.....4 ends</div> | <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Alarm message 1 starts..... Alarm message 1 ends</div> <div style="border: 1px solid black; padding: 2px;">Alarm message 2 starts..... Alarm message 2 ends</div> |
| 65 to 96 characters | 97 to 128 characters |
| 2 messages displayed (Note that the second message has up to 32 characters from the head) | 1 message displayed |
| <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Alarm message 1 starts..... Alarm message 1 ends</div> <div style="border: 1px solid black; padding: 2px;">Alarm message 2 starts.....</div> | <div style="border: 1px solid black; padding: 2px;">Alarm message 1 starts..... Alarm message 1 ends</div> |

10.1.3 Message Creation

(1) Alarm message type

Create messages by using PLC development software (GX-Developer). **(Note 1)**

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 "MELDAS 600, 60/60S Series PLC Development Software Manual (BNP-B2252)".

(Note 1) PLC Onboard does not include the message creation function.

(2) External alarm message type

Text-form PLC alarm message can be input as the External alarm message.

Moreover, PLC alarm message can be input or output with a maintenance data format.

Details of the external alarm messages creation method are described below.

(a) Input with text data format

1) Format of the text file

Format of the message text is shown below.

| | | |
|--|---|--------------------|
| JPN01 96-12-01 ↓ | (i) Version data | } Language No. : 0 |
| 256 * 16,0 ↓ | (ii) Number of characters / messages and language designation | |
| alarm_message001alarm_message002alarm_message003..... alarm_message256 ↓ | (iii) Character string of PLC alarm message | |
| ENG01 96-12-01 ↓ | (i) Version data | } Language No. : 1 |
| 256 * 16,1 ↓ | (ii) Number of characters / messages and language designation | |
| alarm_message001alarm_message002alarm_message003..... alarm_message256 ↓ | (iii) Character string of PLC alarm message | |
| • • • • • | (i) ~ (iii) Up to 8 sets | |
| % | (iv) End code | |

(i) Version data

Up to 15 alphabetical / numerical characters are available.

(Version data of selected language is displayed on construction screen.)

(ii) Number of characters / messages and language designation

Designate the “number of message” and the “number of characters for one message” in decimal. Add “*” (0x2a) code between the numbers.

Always designate even number for the number of characters.

Maximum of messages is 256, and maximum of characters for one message is 128.

These numbers can be designated for each alarm message.

In other words, message size may vary from message to message.

To designate a language, add a comma “,” and parameter data.

When “,0” (number: 0) or nothing is designated, the language for number 0 is selected.

(iii) Character string of PLC alarm message

Set the message text.

It is not necessary to add some code to separate messages.

(The messages are recognized following to (ii) conditions.)

Maximum number of message character strings is 32768 (128 characters × 256 strings) for each language.

(iv) End code

Set “%” (0x25) code.

(Note 1) Always add a return code (CR + LF) in each
 Even if version data is not necessary, return code is needed.
 When a message text file without a return code is used, "E86 INPUT DATA ERR" error will occur.

(Note 2) Make sure so that the number of all characters designated in (ii) conditions (number of characters for one message × number of messages = number of all characters) is the same as the total number of message characters set in (iii).

- 2) Input PLC alarm messages
 Select DATA IN screen and input PLC alarm message with text data format.

#(98) DATA() INPUT

(b) Input/output with maintenance data format

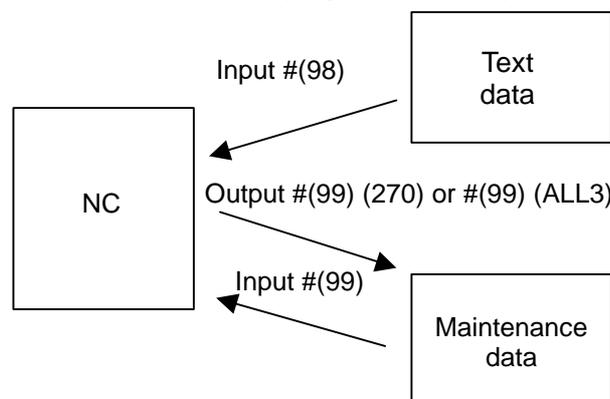
- 1) Input PLC alarm message with maintenance data format
 Select DATA IN screen and input PLC alarm message with maintenance data format .

#(99) DATA() INPUT

- 2) Output PLC alarm message with maintenance data format
 Select DATA OUT screen and input PLC alarm message with maintenance data format .

#(99) DATA(270) INPUT

PLC alarm messages are also output when APLC program batch output (#(99) (ALL3)) is performed.



- 3) N-number assignment of maintenance data
 The head N-number is assigned for each language.
 (The last N-number depends on the data size of alarm message.)
 If there is no message data for some language, the N number assigned for the language will be ignored when maintenance data is output.

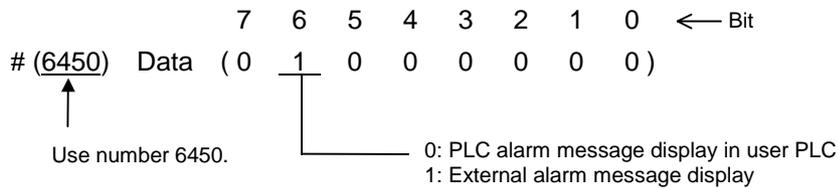
| Language | N number |
|---------------|----------------|
| Language No.1 | 0 to 2499 |
| Language No.2 | 2500 to 4999 |
| Language No.3 | 5000 to 7499 |
| Language No.4 | 7500 to 9999 |
| Language No.5 | 10000 to 12499 |
| Language No.6 | 12500 to 14999 |
| Language No.7 | 15000 to 17499 |
| Language No.8 | 17500 to 19999 |

- (c) Precautions at external alarm creation
 2-byte character (kanji (Chinese character), kana, etc.) can be available for PLC alarm message.
 However, make sure that 2-byte character starts from uneven byte position.
 If 2-byte character is at an even byte position, it may cause overflow to the next line and illegal display.

10.1.4 Parameters

(1) PLC alarm message selection parameter

[Bit selection parameter screen]



The operation is as the following depending on the bit state of the bit selection #6450.

Bit 6 = 0

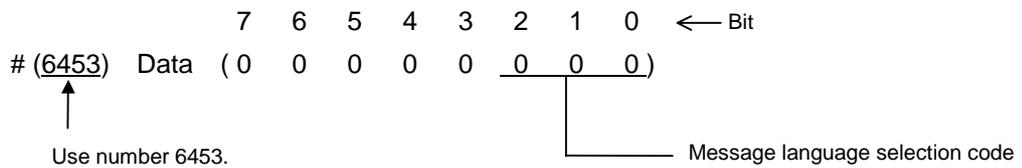
The PLC alarm message in the user PLC is displayed as usual.

Bit 6 = 1

The external alarm message input with the text format is displayed.

(2) Language selection parameter

[Bit selection parameter screen]

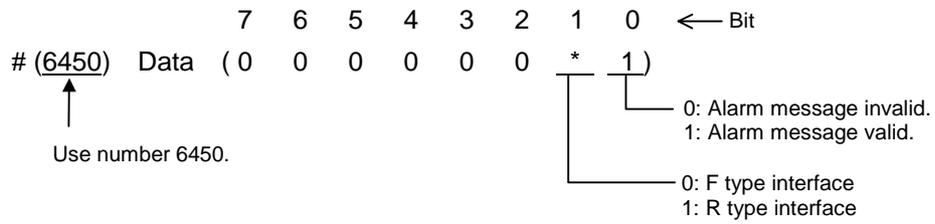


| No. | Bit | | | Notes |
|-------|-----|---|---|------------------------------|
| | 2 | 1 | 0 | |
| #6453 | 0 | 0 | 0 | The language 1 is displayed. |
| | 0 | 0 | 1 | The language 2 is displayed. |
| | 0 | 1 | 0 | The language 3 is displayed. |
| | 0 | 1 | 1 | The language 4 is displayed. |
| | 1 | 0 | 0 | The language 5 is displayed. |
| | 1 | 0 | 1 | The language 6 is displayed. |
| | 1 | 1 | 0 | The language 7 is displayed. |
| | 1 | 1 | 1 | The language 8 is displayed. |

(3) F or R Type Selection Parameter

Set the parameter on the bit selection screen of PLC parameter (setup para).

[Bit selection parameter screen]



[Reference] #6450 corresponds to the high-order byte of the file register R2924.

10.2 Operator Message Display

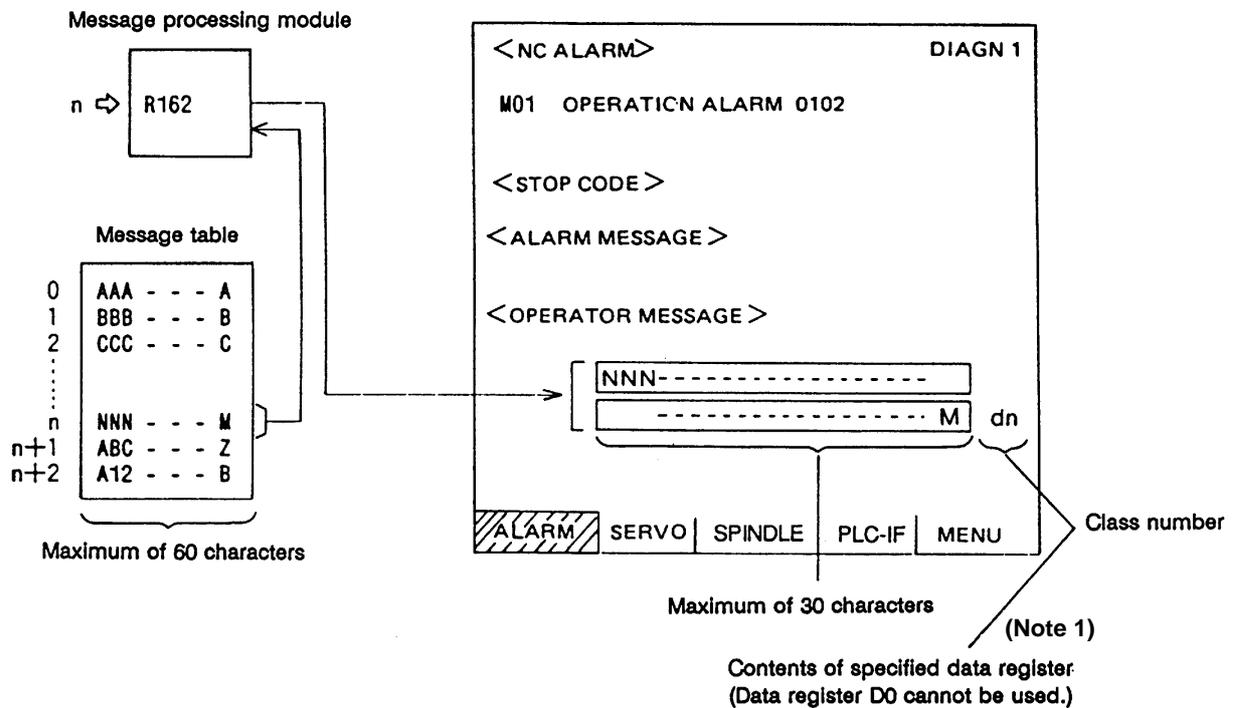
When a condition to inform the operator of a message occurs, an operator message can be displayed independently of an alarm message.

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.

10.2.1 Interface

An operator message is displayed by setting the number of the operator message table to be displayed in file register R162. It is cleared by setting R162 to 0. Thus, number 0 of the operator message table cannot be displayed.

Display example of type 9 setting and display unit



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 R162 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 R162 must be cleared to 0. If the contents of the specified data register are 0, no class numbers are displayed.

10.2.2 Operator Message Preparation

Create messages by using PLC development software (GX Developer). **(Note1)**

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.

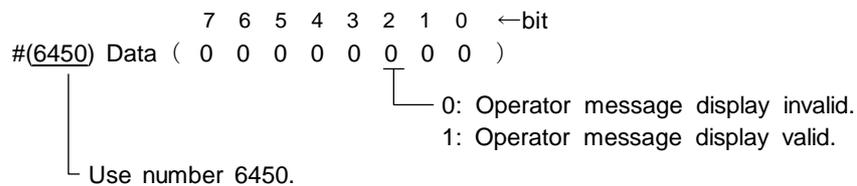
The maximum length of an operator message is 60 characters. A maximum of 512 operator messages can be prepared. For details, refer to "MELDAS 600, 60/60S Series PLC Development Software Manual (BNP-B2252)".

However, the number of operator messages may be limited depending on the available memory capacity. For details, refer to the PLC Development Software Manual.

(Note 1) PLC Onboard does not include the message creation function.

10.2.3 Operator Message Display Validity Parameter

The parameter is set on the machine manufacturer parameter bit selection screen.



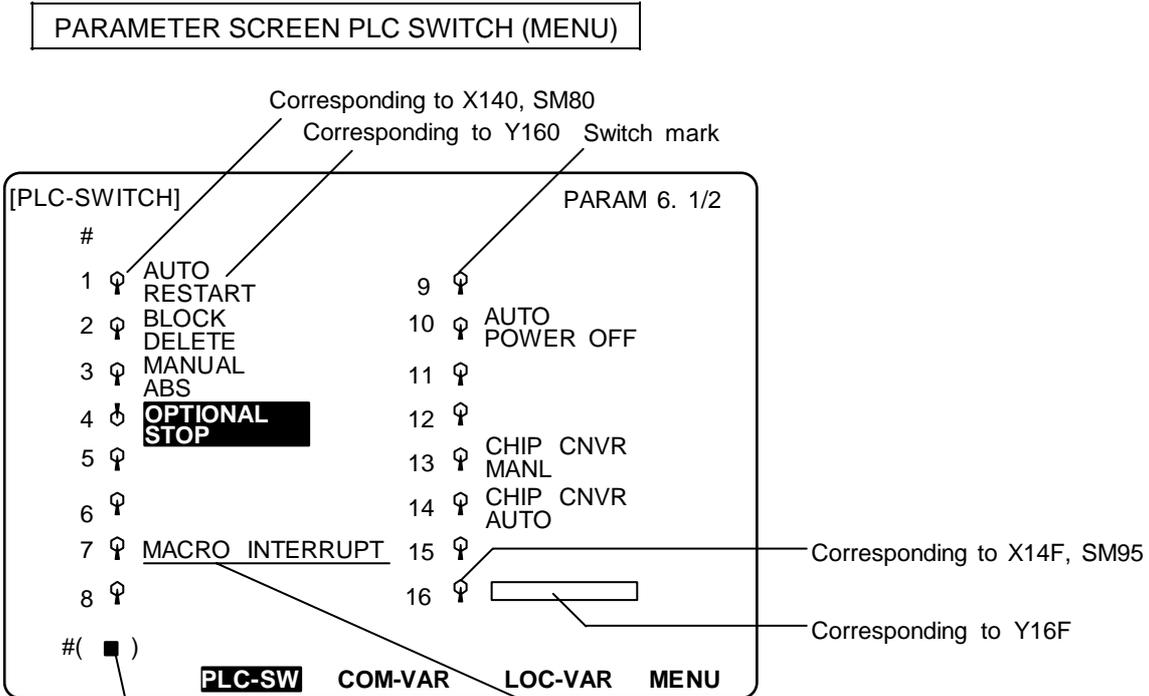
(Reference) #6450 corresponds to the high-order byte of file register R2924.

10.3 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.

10.3.1 Explanation of Screen

The screen is explained below.



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).

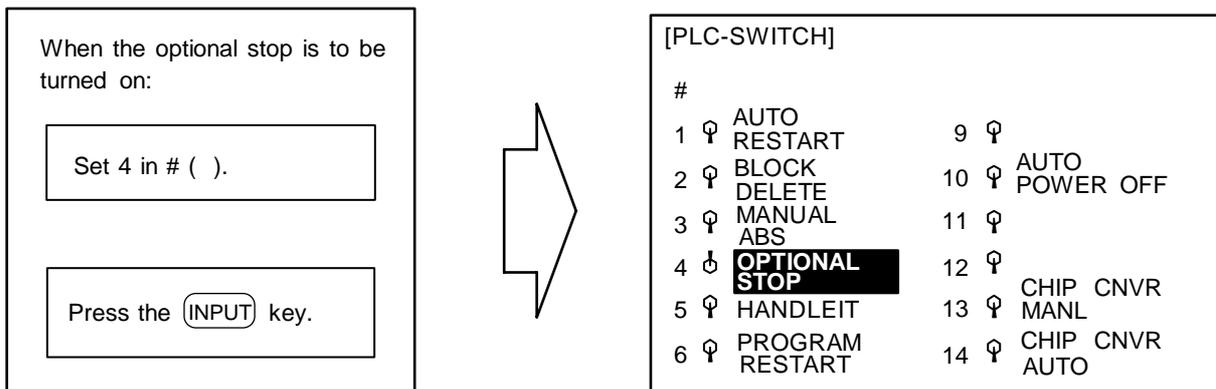
For the switch name, a string of up to 14 alphanumeric and kana characters (kanji requires 2-character space) can be displayed.

Switch on state display part : 
 Switch off state display : 

10.3.2 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 X is turned on (off) and accordingly the switch mark indicates the on (off) state.



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 X is reversed.

To display the switch validity state, etc., the switch name can be highlighted. To do this, turn on or off output device Y corresponding to the switch name.

The corresponding table of the switch numbers, input device X, output device Y, and special relay SM is listed below:

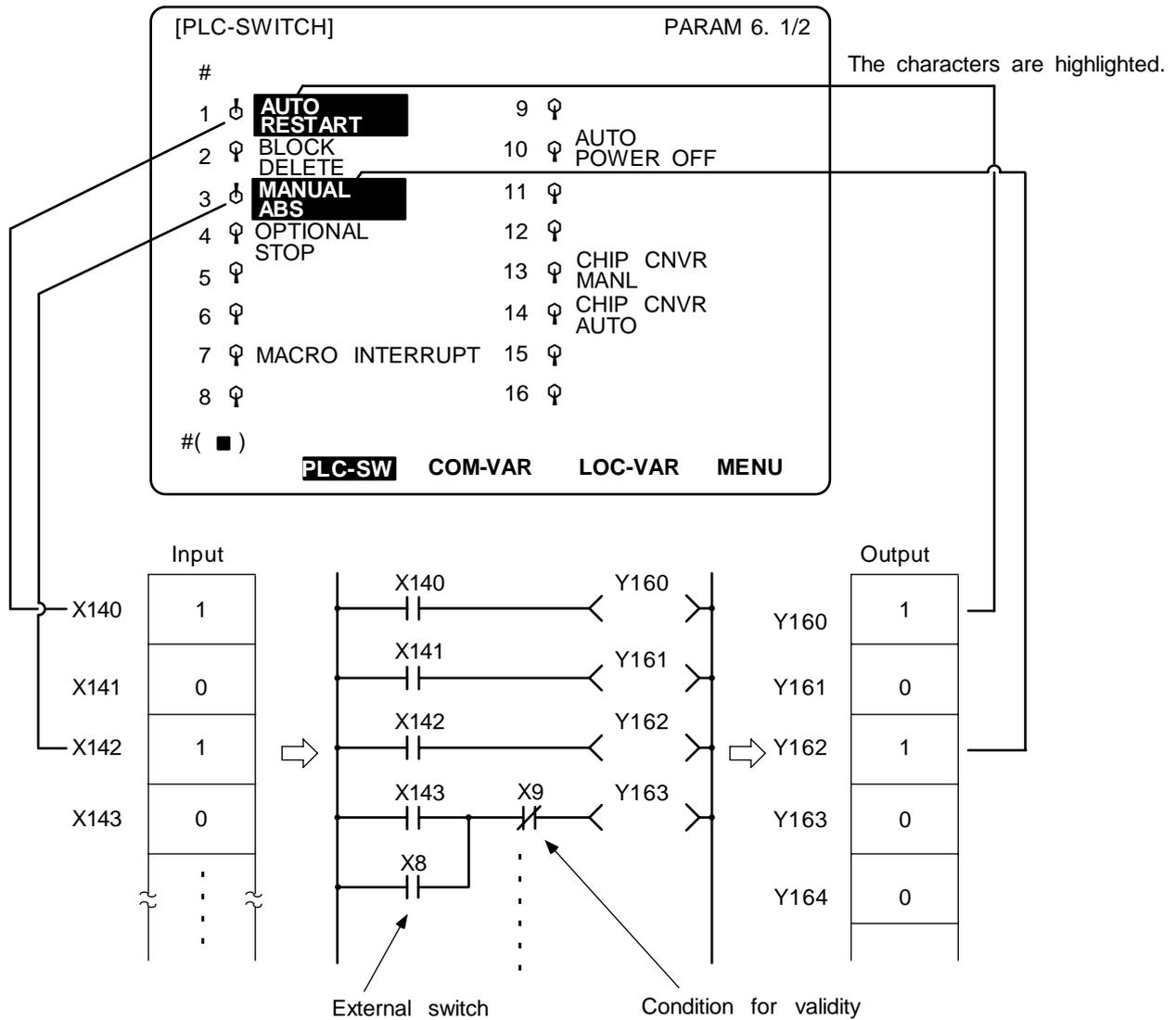
| Switch No. | Corresponding device | | | Switch No. | Corresponding device | | |
|------------|----------------------|------|------|------------|----------------------|------|-------|
| | X | Y | E | | E | Y | E |
| #1 | X140 | Y160 | SM80 | #17 | X150 | Y170 | SM96 |
| #2 | X141 | Y161 | SM81 | #18 | X151 | Y171 | SM97 |
| #3 | X142 | Y162 | SM82 | #19 | X152 | Y172 | SM98 |
| #4 | X143 | Y163 | SM83 | #20 | X153 | Y173 | SM99 |
| #5 | X144 | Y164 | SM84 | #21 | X154 | Y174 | SM100 |
| #6 | X145 | Y165 | SM85 | #22 | X155 | Y175 | SM101 |
| #7 | X146 | Y166 | SM86 | #23 | X156 | Y176 | SM102 |
| #8 | X147 | Y167 | SM87 | #24 | X157 | Y177 | SM103 |
| #9 | X148 | Y168 | SM88 | #25 | X158 | Y178 | SM104 |
| #10 | X149 | Y169 | SM89 | #26 | X159 | Y179 | SM105 |
| #11 | X14A | Y16A | SM90 | #27 | X15A | Y17A | SM106 |
| #12 | X14B | Y16B | SM91 | #28 | X15B | Y17B | SM107 |
| #13 | X14C | Y16C | SM92 | #29 | X15C | Y17C | SM108 |
| #14 | X14D | Y16D | SM93 | #30 | X15D | Y17D | SM109 |
| #15 | X14E | Y16E | SM94 | #31 | X15E | Y17E | SM110 |
| #16 | X14F | Y16F | SM95 | #32 | X15F | Y17F | SM111 |

(Note 1) Input device X also holds the state 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. |

10.3.3 Signal Processing

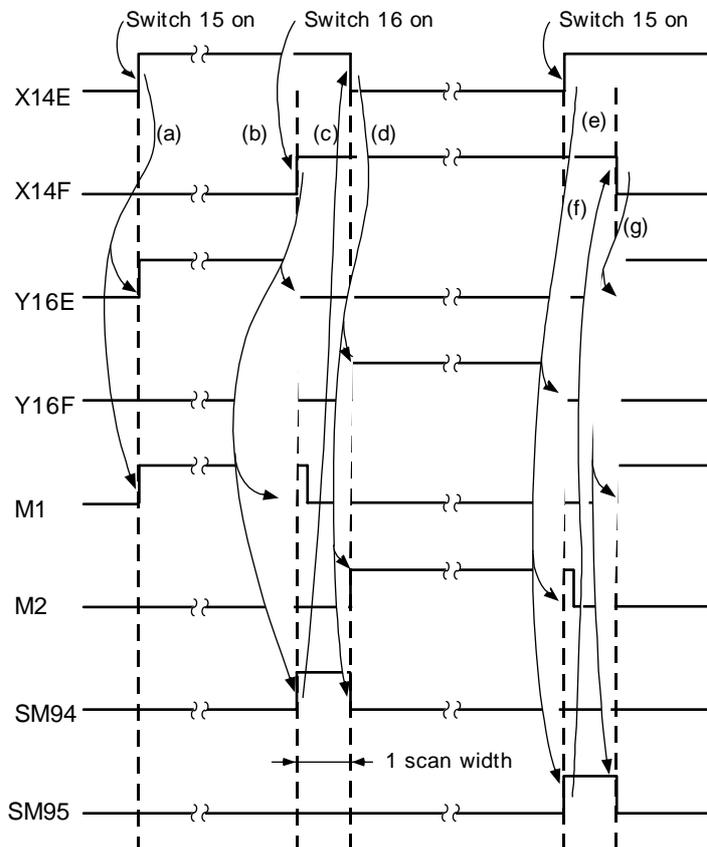
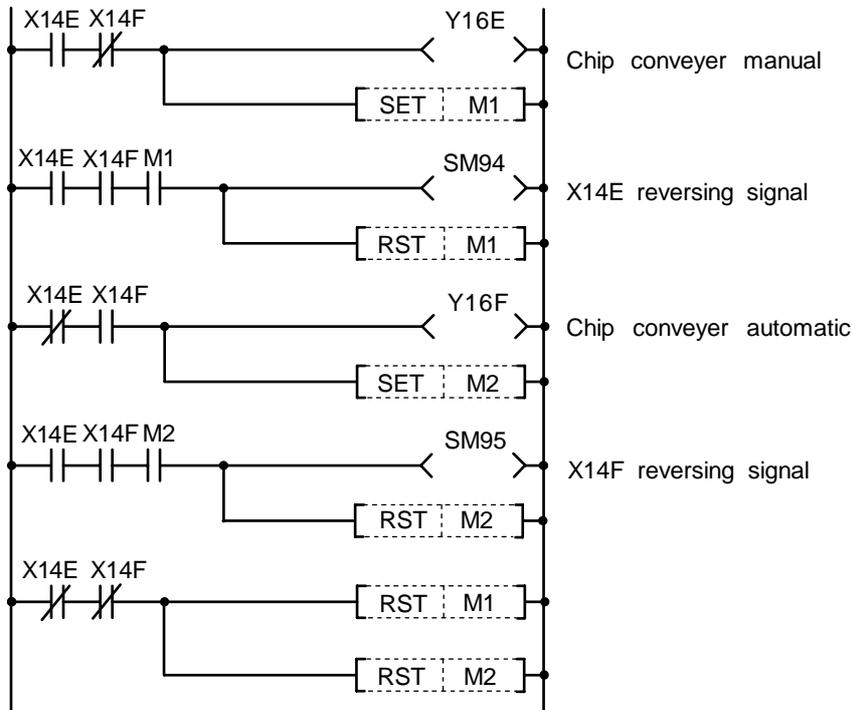


- When setting is done on the PLC switch screen, the input device X corresponding to the specified switch number is turned on or off to switch over the switch state.
- When special relay SM is turned on from the user PLC, its corresponding input device X and the switch state are reversed. Special relay SM is reset immediately after the CNC reverses the input device X and the switch state. It is turned on by one pulse (scan) only also in the user PLC. In either case, when output device Y is set to on based on the input device X 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

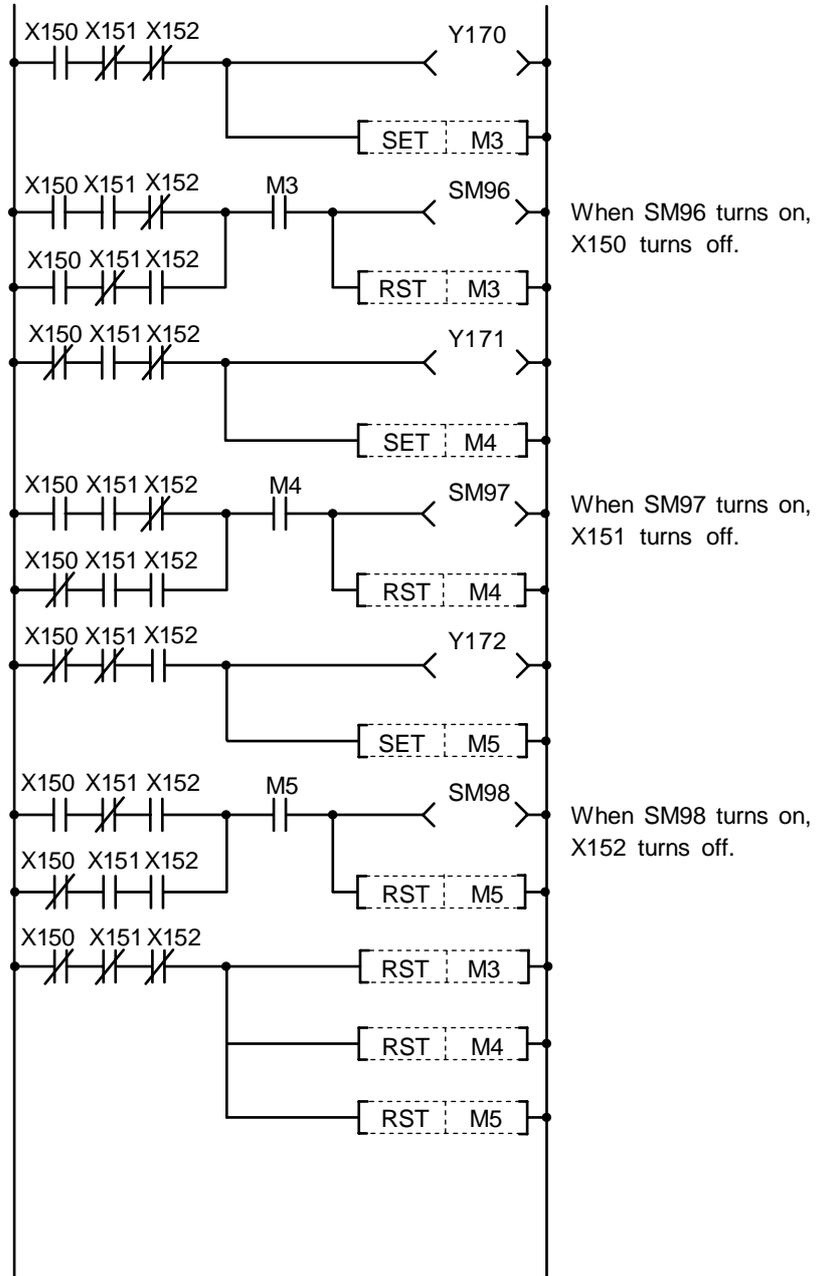
(Example) When two opposite switches, chip conveyer manual and chip conveyer automatic, are provided;



- (a) When switch 15 (X14E) is on and switch 16 (X14F) is off, Y16E and M1 turn on. [Initial state]
- (b) When switch 16 (X14F) turns on while being in state (a), Y16E turns off, SM94 turns on, and M1 turns off.
- (c) Turning SM94 on reverses X14E (to off).
- (d) When X14E is off and X14F is on, SM94 turns off and Y16F and M2 turn on.
- (e) When switch 15 (X14E) turns on while being in state (d), Y16F turns off, SM95 turns on, and M2 turns off.
- (f) Turning SM95 on reverses X14F (to off).
- (g) When X14F is off and X14E is on, SM95 turns off and Y16E and M1 turn on again.

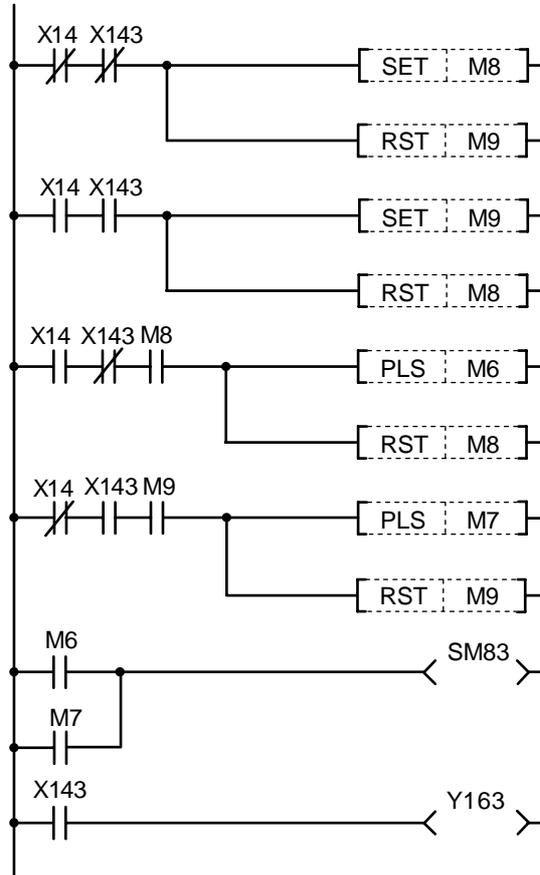
(2) Three-point switch

(Example) When three opposite switches 17, 18, and 19 are provided;



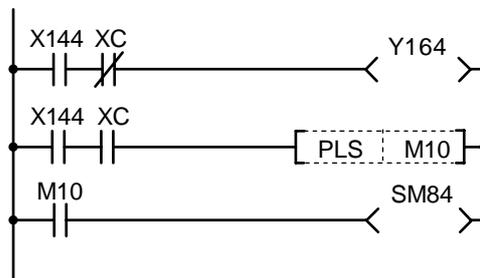
(3) External switch and PLC switch

(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.

10.3.4 Switch Name Preparation

Prepare PLC switch names by using PLC development software (GX Developer). **(Note1)**

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.

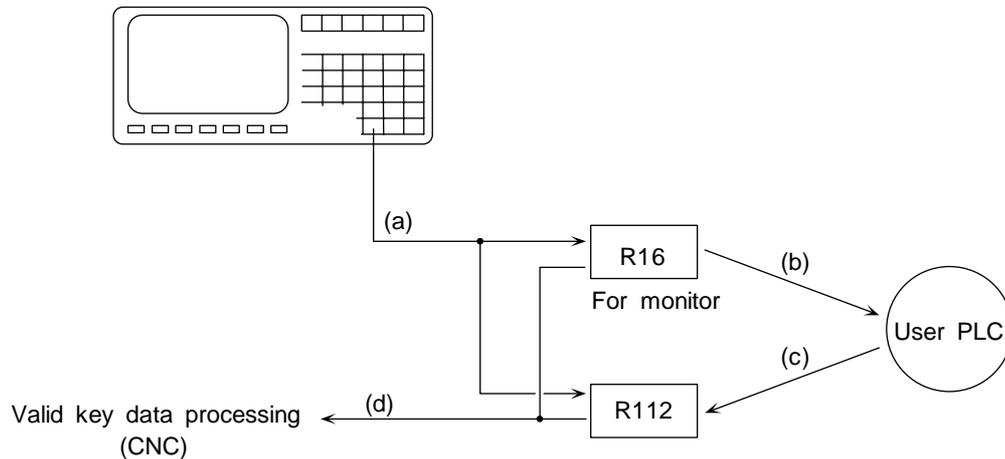
For details, refer to "MELDAS 600, 60/60S Series PLC Development Software Manual (BNP-B2252)".

(Note 1) PLC Onboard does not include the switch name creation function.

10.4 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.

10.4.1 Key Data Flow



- (a) Key data is set in file registers R16 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 R16 and R112 contents.

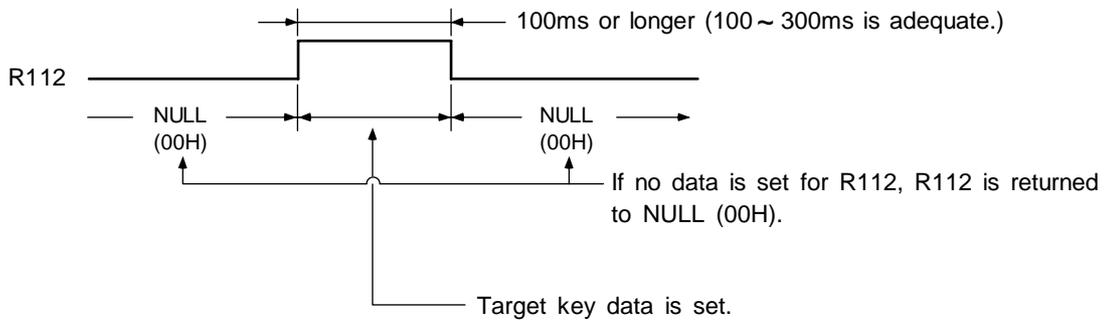
10.4.2 Key Operations That Can Be Performed

- (1) When a key is pressed, it is ignored.
 - The R16 contents are judged and NULL (00H) code is set in R112.
- (2) When R16 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.

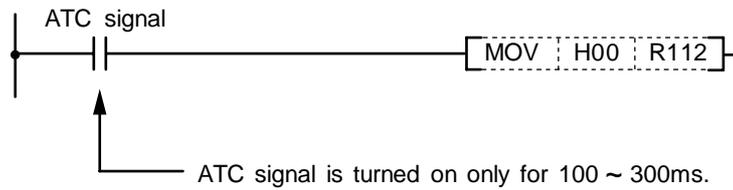
10.4.3 Key Data Processing Timing

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)



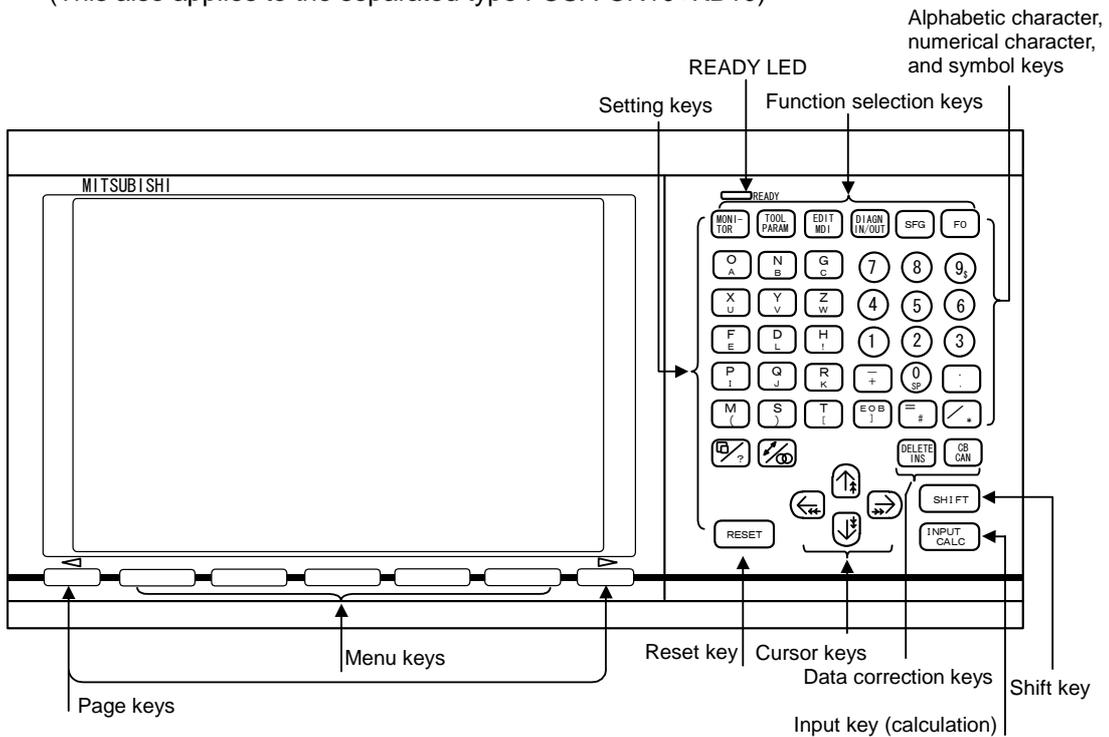
10.4.4 Layout of Keys on Communication Terminal

There are two types of layouts for the keys on the communication terminal used with this controller as shown below.

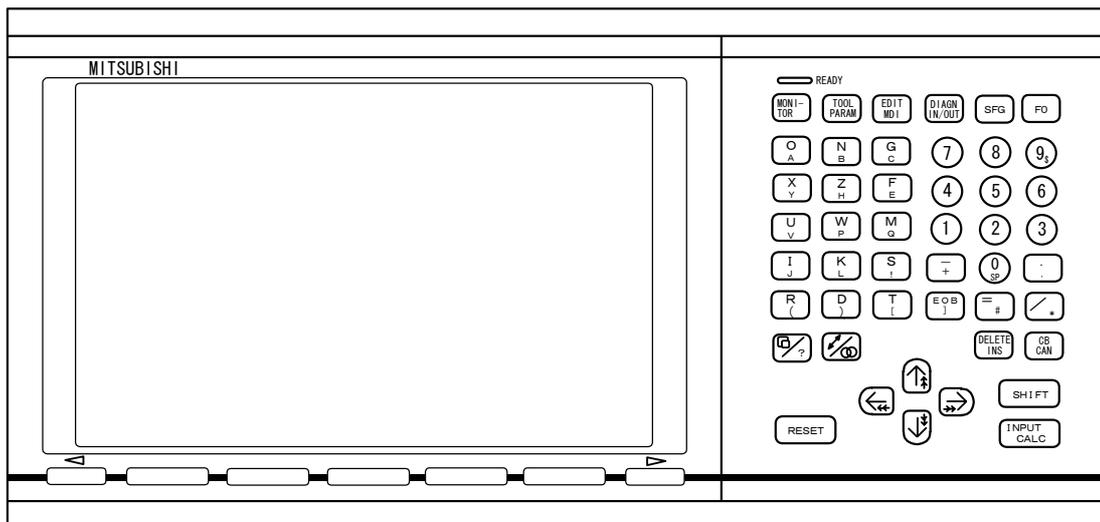
The layouts of the alphabetic keys differ.

(1) Key layout for communication terminal CT100

(This also applies to the separated type FCUA-CR10+KB10)



(2) Key layout for communication terminal CT120



(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.

10.4.5 List of Key Codes

(1) For communication terminal CT100, KB10 (M series)

| 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

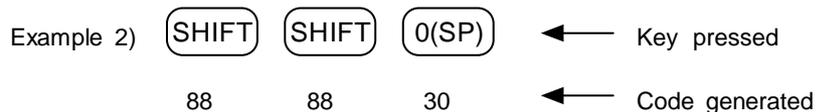
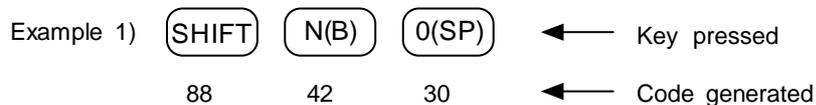
Example 2) **SHIFT** **SHIFT** **0(SP)** ← Key pressed
 88 88 30 ← Code generated

(2) For communication terminal CT120 (L series)

| 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(59) |
| SFG | 86 | DELETE (INS) | 7F(8C) | / (*) | 2F(2A) | Z (H) | 5A(48) |
| F0 | 87 | C.B.(CAN) | 8E(18) | | | F (E) | 46(45) |
| | | SHIFT | 88 | 0 (SP) | 30(20) | U (V) | 55(56) |
| | | INPUT(CALC) | 0D(F4) | 1 | 31 | W (P) | 57(50) |
| | | | | 2 | 32 | M (Q) | 4D(51) |
| | | | | 3 | 33 | I (J) | 49 (4A) |
| Previous page | 90 | Window key (?HELP) | 89(F9) | 4 | 34 | K (L) | 4B(4C) |
| Next page | 9A | Activ Wind (CTRL) | 8A(8B) | 5 | 35 | S (!) | 53(21) |
| Menu 1 | 91 | | | 6 | 36 | R (() | 52(28) |
| Menu 2 | 92 | | | 7 | 37 | D ()) | 44(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.



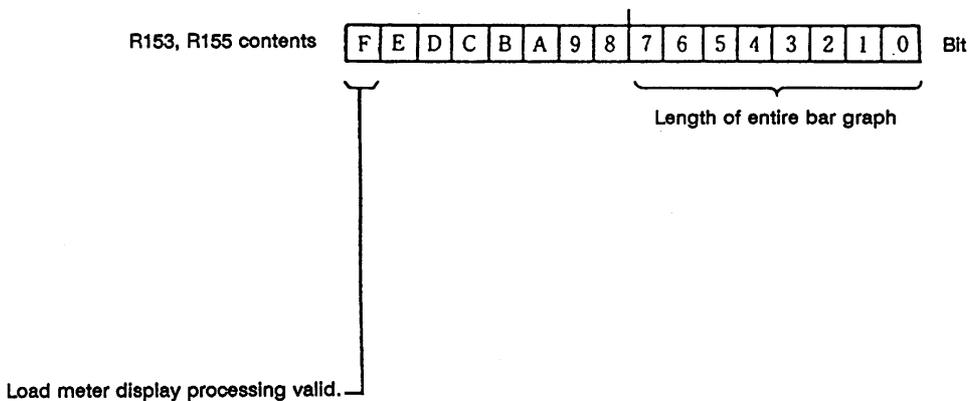
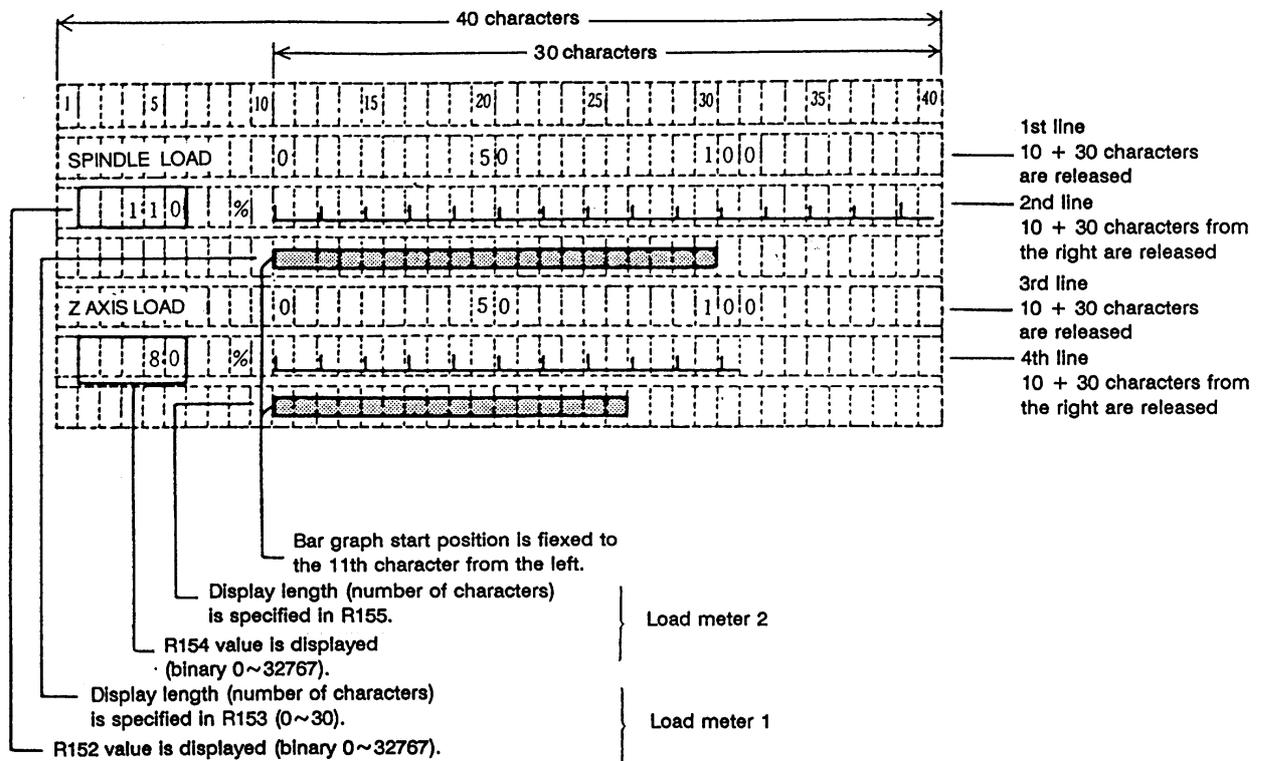
10.5 Load Meter Display

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 "MELDAS 600, 60/60S Series PLC Development Software Manual (BNP-B2252)".

(Note 1) PLC Onboard does not include the switch name creation function.

10.5.1 Interface

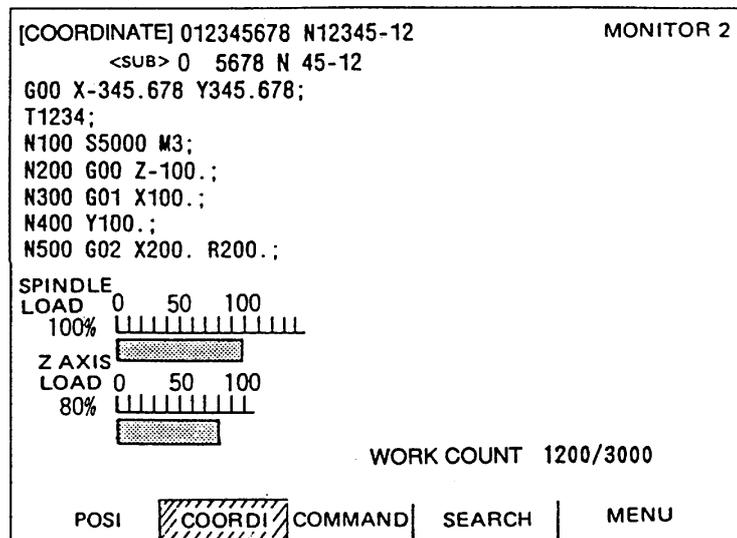


○ File register (R) for load meter display

| | | For \$1 | For \$2 |
|--------------|-------------------|---------|---------|
| Load meter 1 | Numerical display | R152 | R352 |
| | Bar graph display | R153 | R353 |
| Load meter 2 | Numerical display | R154 | R354 |
| | Bar graph display | R155 | R355 |

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

Display example of type 9 setting and display unit



Display on coordinate screen of position display second menu.

(Note) This screen consists of 80 characters wide x 18 lines long.

10.6 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.

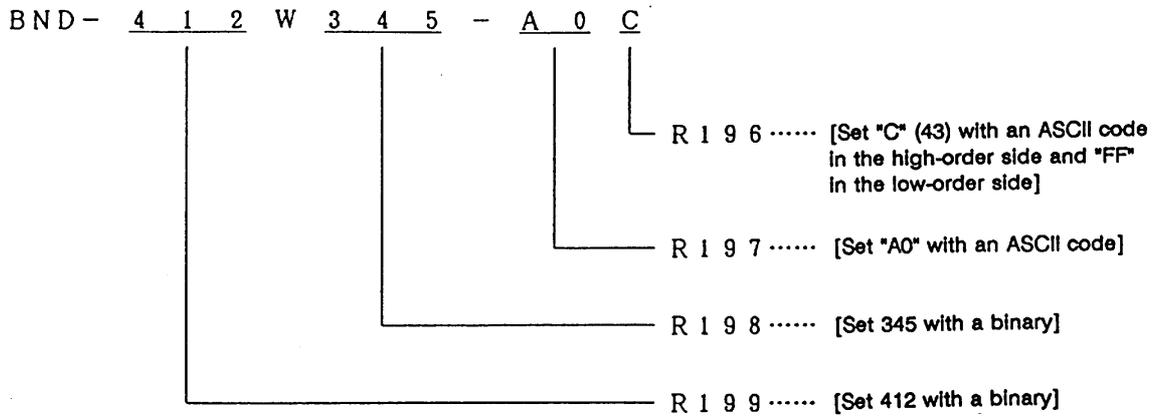
| File register | Contents | File register | Contents |
|---------------|---|---------------|---|
| R560 | \$1 Compensation data for the first axis | R568 | \$2 Compensation data for the first axis |
| R561 | \$1 Compensation data for the second axis | R569 | \$2 Compensation data for the second axis |
| R562 | \$1 Compensation data for the third axis | R570 | — |
| R563 | \$1 Compensation data for the fourth axis | R571 | — |
| R564 | — | R572 | — |
| R565 | — | R573 | — |
| R566 | — | R574 | — |
| R567 | — | R575 | — |

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

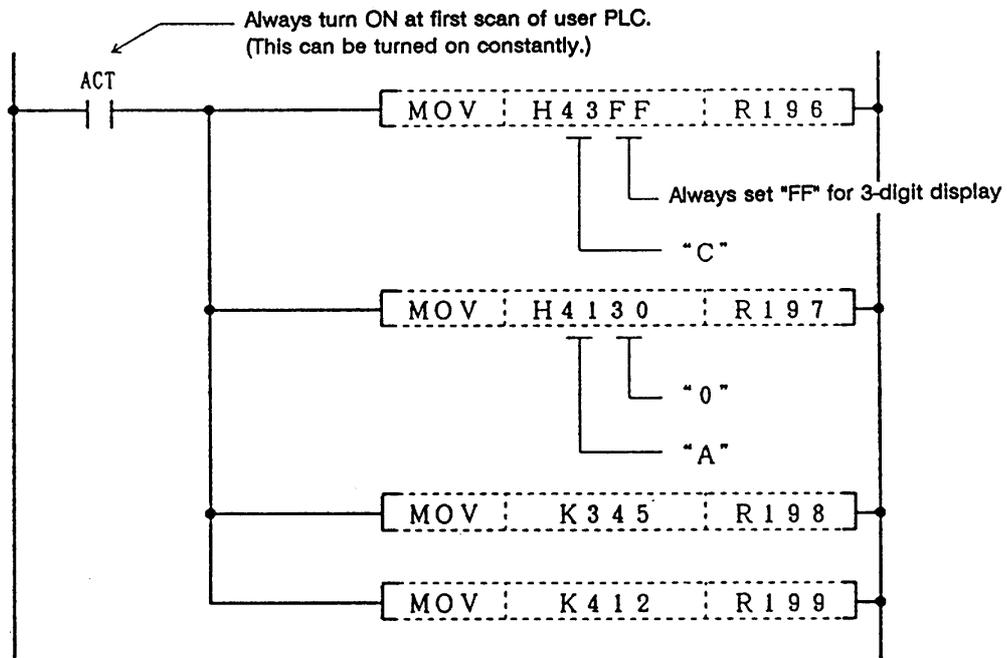
Data in file registers R560~R575 is not backed up. If it must be backed up, use back-up file registers (R1900~R2799).

(Note 1) The maximum delay to compensation is (one user PLC scan + 15ms). However, smoothing time constant and servo follow delay are not contained.

(2) To display a 3-digit version code



Program example)



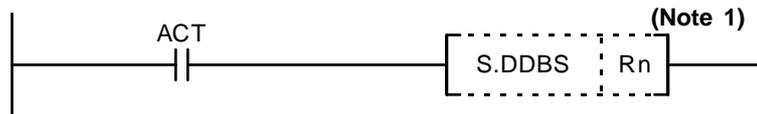
11.2.2 Other Restrictions

- (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.

11.3 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 DDBS function.

11.3.1 S.DDBS Function Command

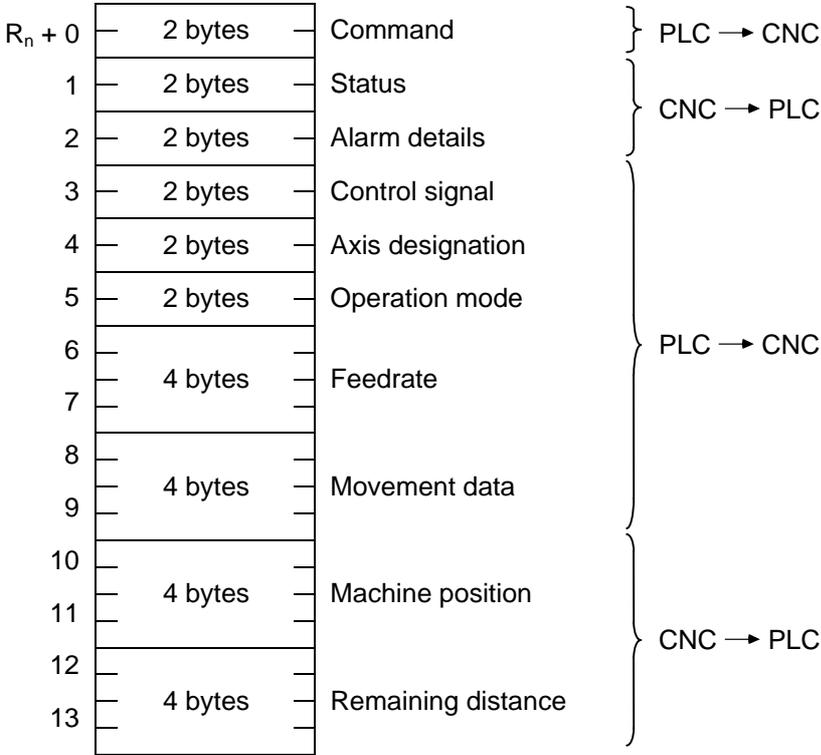


When ACT is set to 1, the PLC axis control process is carried out with the control information data contents. Thus, ACT should be set to 1 during PLC axis control. Setting ACT to 0 causes a reset status.

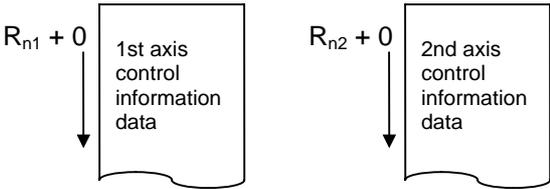
(Note 1) The following R-registers can be used.
R500 to R549 (No battery backup)
R1900 to R2799 (Battery backup)

11.3.2 Control Information Data

Set the control information data in the R-register before calling the DDBS function command. The following is a list of control information data.



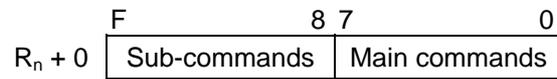
A max. of 2 axes can be controlled by the PLC. Each axis should have its own control information data.



11.3.3 Control Information Data Details

11.3.3.1 Commands

Commands consist of main commands and sub-commands.



Main commands: The types of DBBS main commands are as follows.

- 1: Search
- 2: PLC axis control

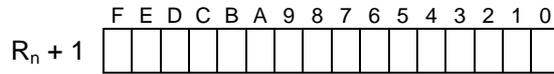
Sub-commands: The PLC axis control sub-command is as follows.

- 0: Movement data output and control signal output

(Note 1) "Input" and "output" are the input/output looking from the PLC side.

11.3.3.2 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.



| | | | |
|-------------|-------------------------|--------------|--|
| 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: | | 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.

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.

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.

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 turning the reset signal ON, 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

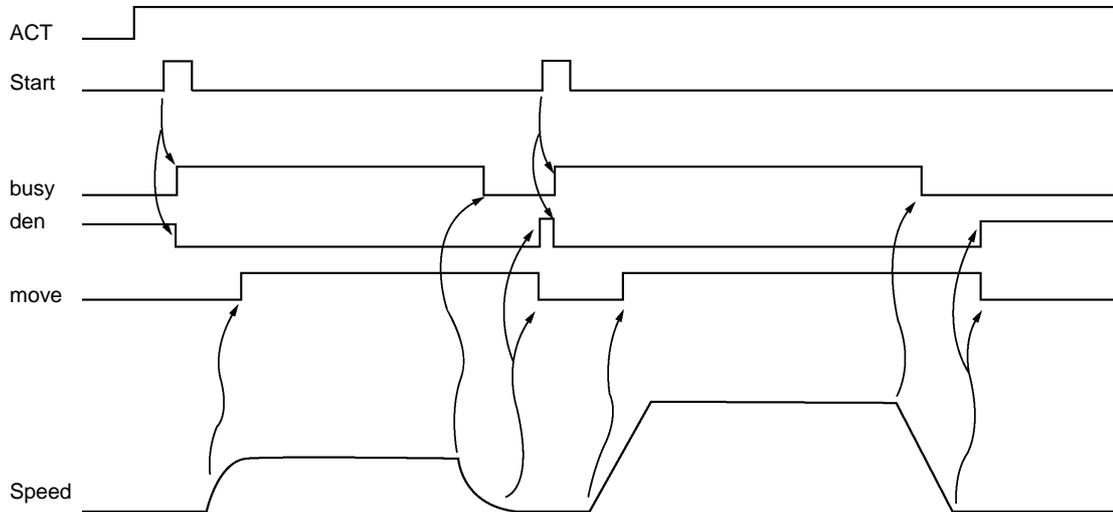
| | | |
|-----------------|----|-------------|
| S03 Servo alarm | 52 | 1 |
| | | └─ PLC axis |

bit F: ALM:1 Control information data designation alarm

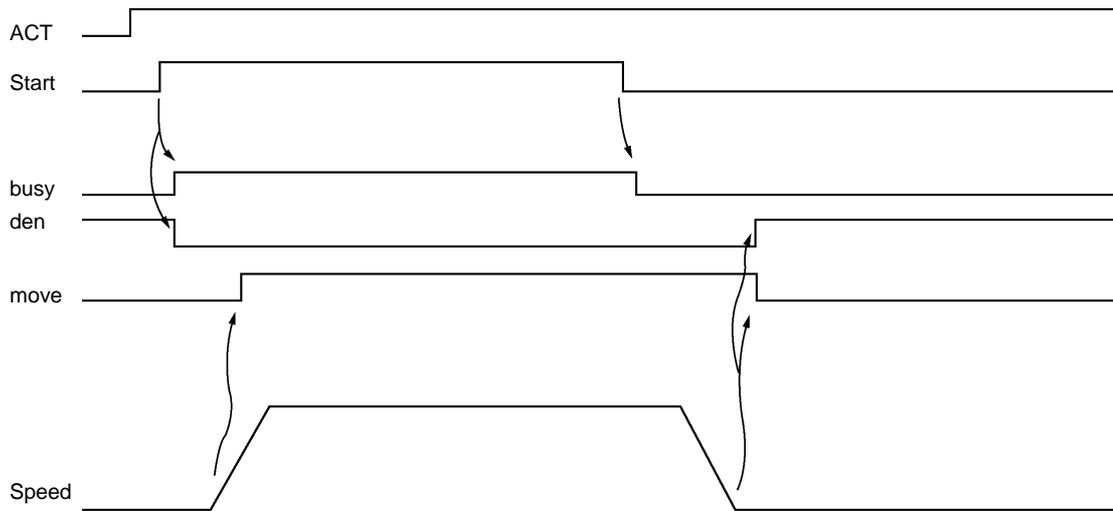
This bit turns ON when the designated details of the control information data are illegal. Thus, the PLC axis control process is not executed. Turn the bit OFF by correcting the data, turning the reset signal ON, or setting ACT to 0.

Timing chart

(1) For rapid traverse and cutting feed mode



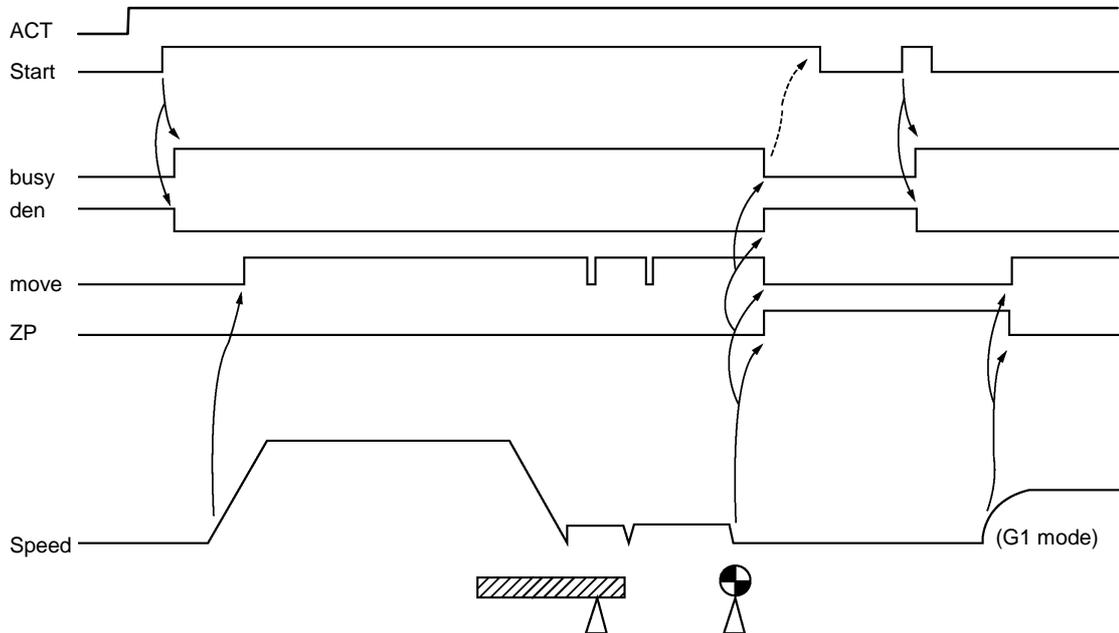
(2) For jog feed mode



(Note) The axis moves by jog feed only during start ON.

(3) For reference point return feed mode

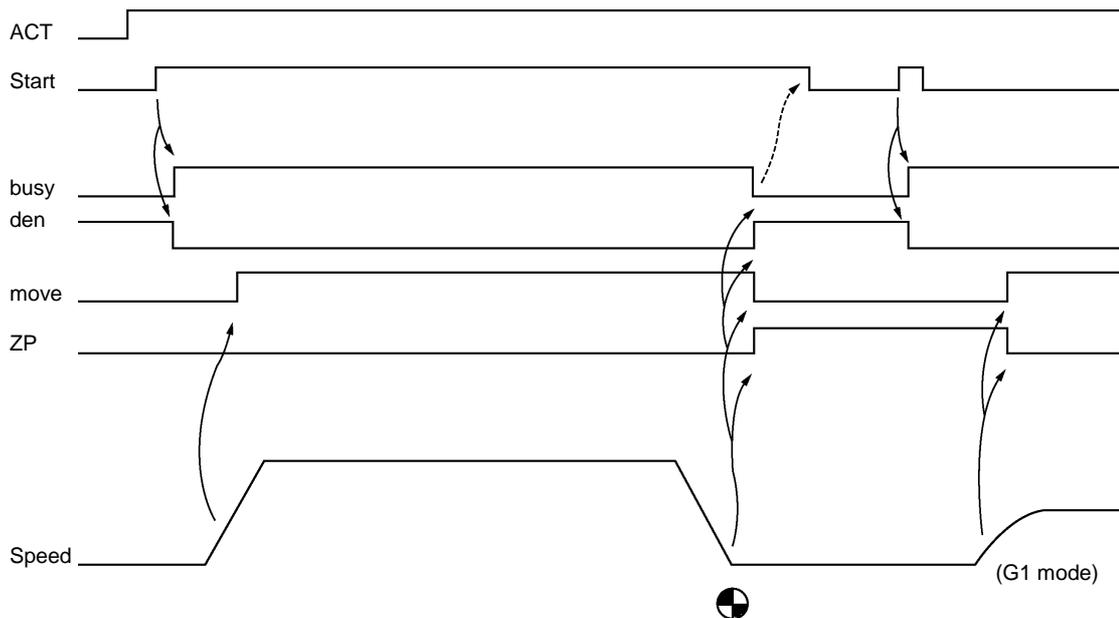
(3-1) Dog-type reference point return



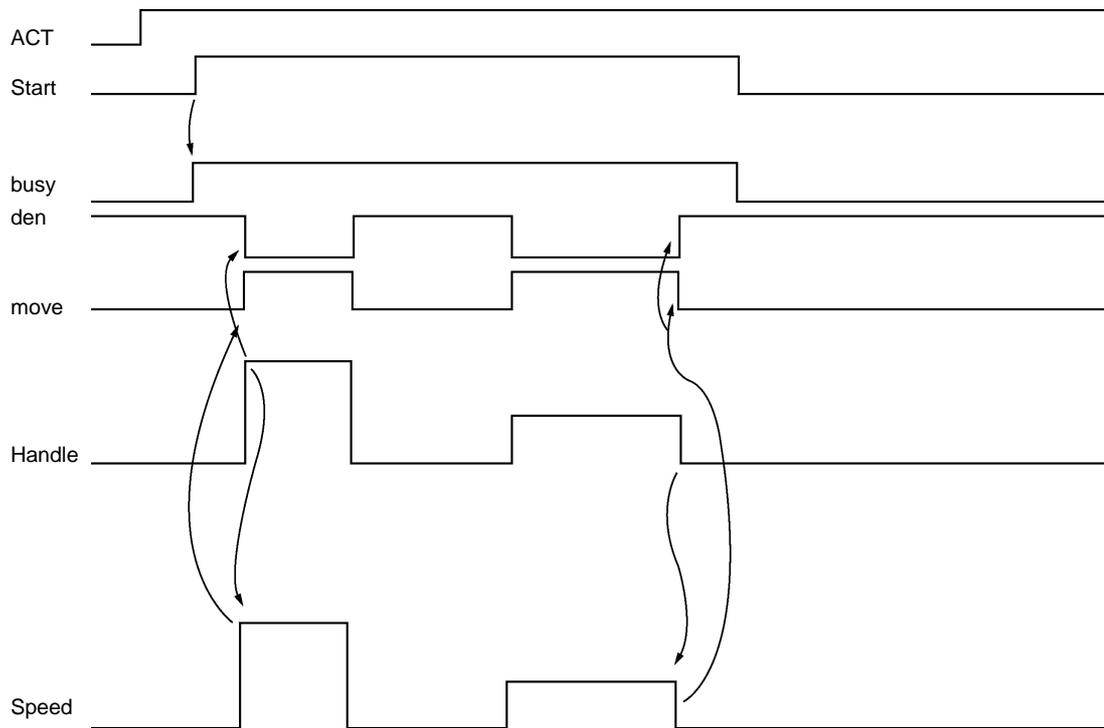
(Note 1) The axis moves by reference point return feed only during start ON. Turn the start OFF after confirming that the reference point has been reached.

(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

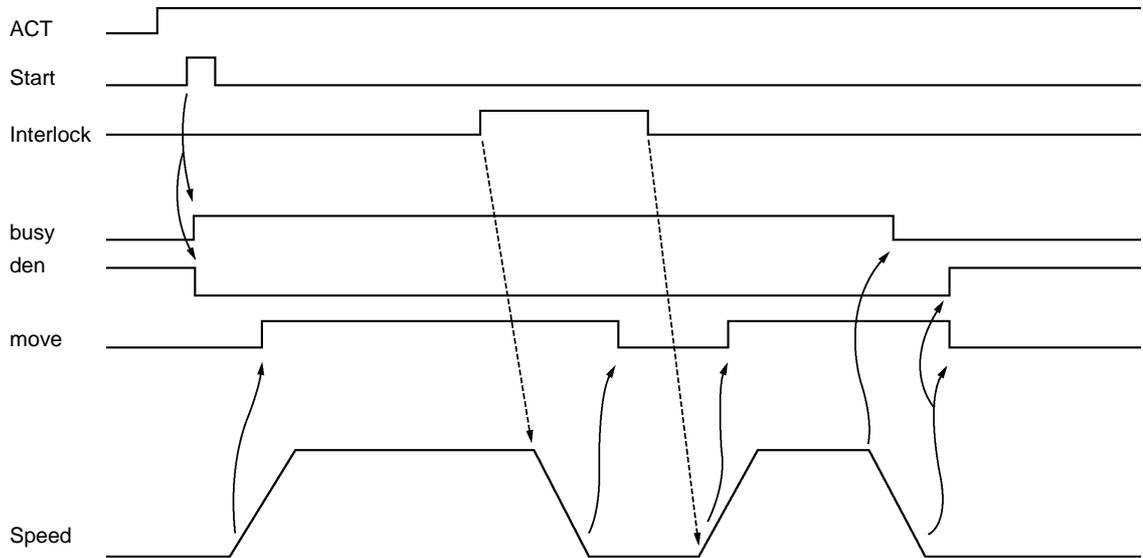


(4) For handle feed mode

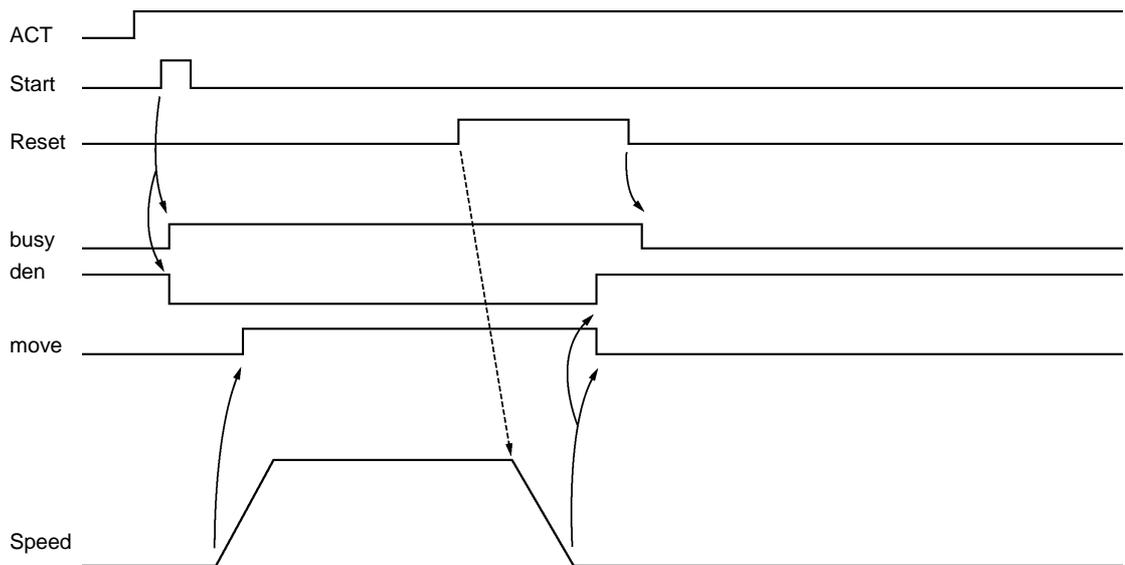


(Note) Handle feed is possible only during start ON.

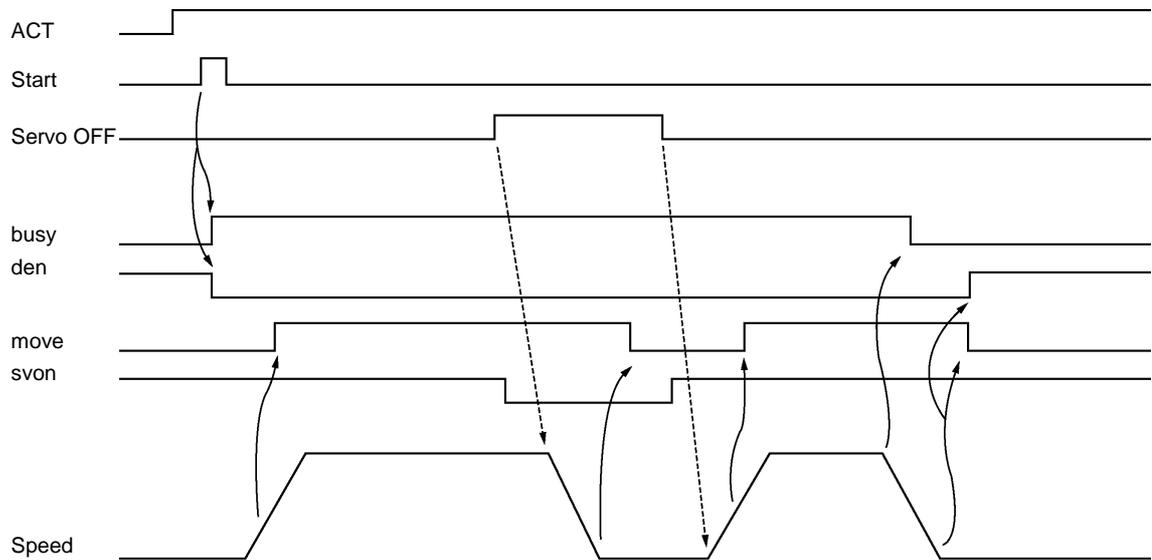
(5) When the interlock signal is ON (= 1)



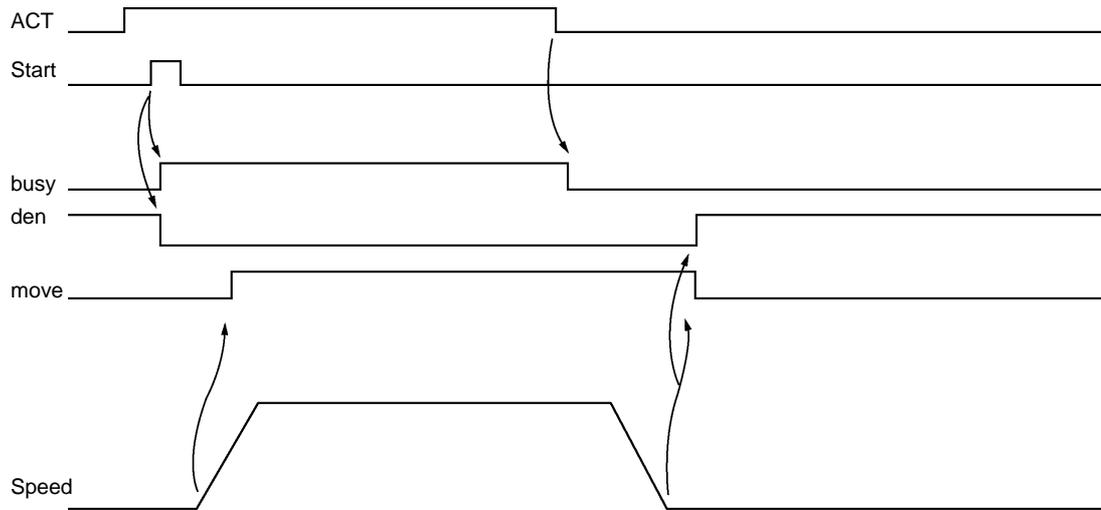
(6) When the reset signal is ON (= 1)



(7) When the servo OFF signal is ON (= 1)



(8) When the ACT signal is OFF (= 0)



11.3.3.3 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)

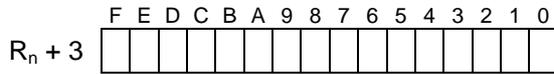
| Alarm No. | Details |
|-----------|---|
| 01 | Control signal illegal (A signal other than a registered control signal has been commanded.) |
| 02 | Axis No. illegal |
| 03 | Operation mode illegal (0 to 6) |
| 04 | Movement data range exceeded -99999999 to +99999999 |
| 05 | |
| 06 | |
| ⋮ | |
| 10 | Zero point return not complete (absolute value command not possible) |
| 11 | |
| 12 | |

(2) 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 (-) |

11.3.3.4 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.



- | | |
|---|---|
| <p>bit 0: Start</p> <p>1: Interlock</p> <p>2: Reset</p> <p>3: Servo OFF</p> <p>4: Axis removal</p> <p>5: Axis removal 2</p> <p>6:</p> <p>7:</p> | <p>bit 8 : Absolute value command</p> <p>9:</p> <p>A:</p> <p>B:</p> <p>C:</p> <p>D:</p> <p>E:</p> <p>F:</p> |
|---|---|

bit 0: Start

Starting begins at the at the rising edge (OFF -> ON) of the start signal, based on the control information data.
 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.

11.3.3.5 Axis Designation

The axis No. of the PLC axis is designated.

$R_n + 4$

| |
|------------------|
| Axis designation |
|------------------|

- 0: 1st axis
- 1: 2nd axis

11.3.3.6 Operation Mode

The operation mode for the PLC axis is designated.

For example, in the handle mode, $R_n+5=6$ (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.

11.3.3.7 Feedrate

When the operation mode is cutting feed or jog feed ($R_n + 5 = 1$ to 3), the PLC axis feedrate is designated with a binary code.

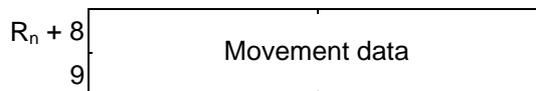


Designation value
1 to 1000000 mm/min. (0.1 inch/min.)

- (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 ($R_n + 6, 7$) is possible.

11.3.3.8 Movement Data

When the operation mode is rapid traverse or cutting feed, the movement data is designated with a binary code.

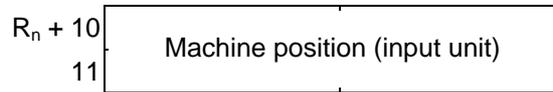


Designation value
0 to ± 99999999 (0.001mm/0.0001inch)

- (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.

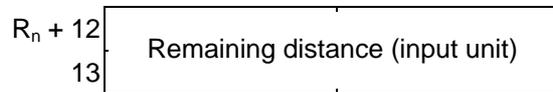
11.3.3.9 Machine Position

The machine position output to the machine system is expressed. The machine position becomes the rfp (reference point) when the reference point is reached.



11.3.3.10 Remaining Distance

The remaining distance of the movement data output to the machine system is expressed.



11.3.4 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 | |
|------------|-------|-------------|--|
| Y2E0 | *PCD1 | PLC axis | PLC axis near point detect 1st axis |
| Y2E1 | *PCD2 | PLC axis | PLC axis near point detect 2nd axis |
| Y2E2 | | | |
| Y2E3 | | | |
| Y2E4 | | | |
| Y2E5 | | | |
| Y2E6 | | | |
| Y2E7 | | | |

(Note) The responsiveness when the dog signal is set in PLC middle-speed processing is worse than when set in PLC high-speed processing.

11.3.5 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 |
|------------|------|---------------------------|
| Y2E0 | | |
| Y2E1 | | |
| Y2E2 | | |
| Y2E3 | | |
| Y2E4 | PCH1 | PLC axis 1st handle valid |
| Y2E5 | PCH2 | PLC axis 2nd handle valid |
| Y2E6 | | |
| Y2E7 | | |

When Y2E4 and Y2E5 are ON, each handle changes to PLC axis dedication. Y248 to Y24C, Y24F, Y250 to Y254 and Y257 usually used in the control device are used for the axis selection of each handle.

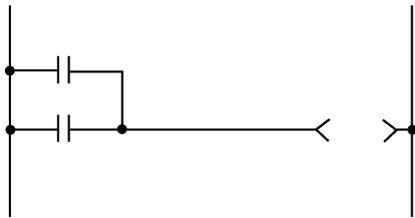
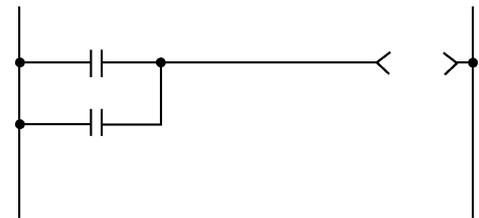
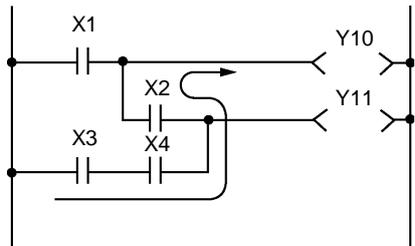
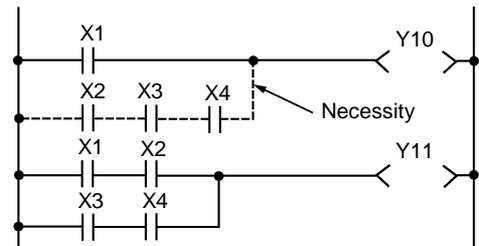
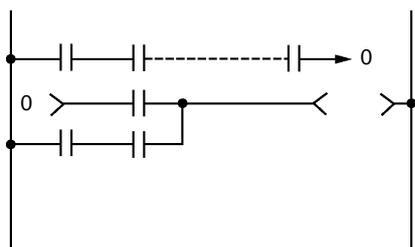
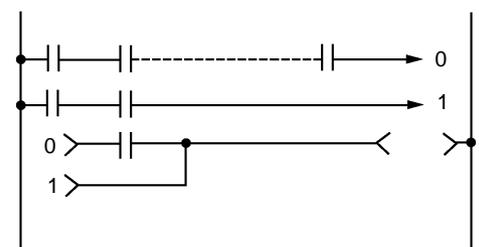
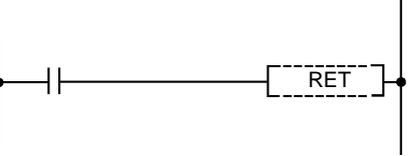
PLC axes are counted as PLC such as first axis and second axis. Therefore, if you will operate the first handle in the first axis of PLC, turn ON Y2E4, Y248 to Y24C and Y24F.

(Note) The handle feed magnification is also used for NC control axes.

12. Appendix

12.1 Example of Faulty Circuit

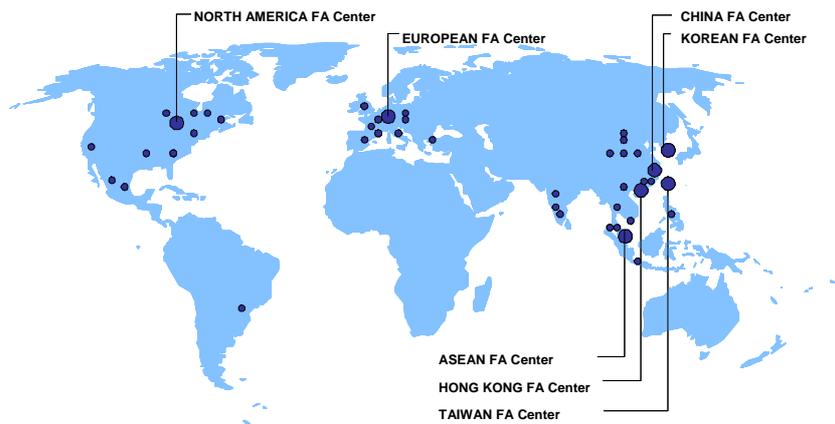
Wrong configurations of circuits are shown below. Correct the circuitry, if any.

| Faulty circuit producing errors | → Correct circuit |
|---|--|
| <p>(1) Circuit containing OR</p>  |  |
| <p>(2) Rounding circuit</p>  <p>Whether or not the Y10 condition includes X3, X4 and X2 is unknown.</p> |  |
| <p>(3) Modification of loopback circuit</p>  |  |
| <p>(4) Presence of a contact before RET, FEND, or MCR circuit</p>  |  |

Revision History

| Date of revision | Manual No. | Revision details |
|------------------|------------|--|
| Aug. 2002 | BNP-B2269C | First edition created. |
| Nov. 2005 | BNP-B2269D | <p>Contents were revised to correspond to the system software version C6.</p> <p>The following sections were added.</p> <ul style="list-style-type: none">• 9.6 Chopping• 9.7 CC-Link <p>Addition/deletion/revision was made according to the specifications.</p> |
| | | |

Global service network



North America FA Center (MITSUBISHI ELECTRIC AUTOMATION INC.)

Illinois CNC Service Center
500 CORPORATE WOODS PARKWAY, VERNON HILLS, IL. 60061, U.S.A.
TEL: +1-847-478-2500 (Se) FAX: +1-847-478-2650 (Se)

California CNC Service Center
5665 PLAZA DRIVE, CYPRESS, CA. 90630, U.S.A.
TEL: +1-714-220-4796 FAX: +1-714-229-3818

Georgia CNC Service Center
2810 PREMIERE PARKWAY SUITE 400, DULUTH, GA., 30097, U.S.A.
TEL: +1-678-258-4500 FAX: +1-678-258-4519

New Jersey CNC Service Center
200 COTTONTAIL LANE SOMERSET, NJ. 08873, U.S.A.
TEL: +1-732-560-4500 FAX: +1-732-560-4531

Kentucky CNC Service Satellite
8025 PRODUCTION DRIVE, FLORENCE, KY., 41042, U.S.A.
TEL: +1-859-342-1700 FAX: +1-859-342-1578

Michigan CNC Service Satellite
2545 38TH STREET, ALLEGAN, MI., 49010, U.S.A.
TEL: +1-847-478-2500 FAX: +1-269-673-4092

Ohio CNC Service Satellite
62 W. 500 S., ANDERSON, IN., 46013, U.S.A.
TEL: +1-847-478-2608 FAX: +1-847-478-2690

Texas CNC Service Satellite
1000, NOLEN DRIVE SUITE 200, GRAPEVINE, TX. 76051, U.S.A.
TEL: +1-817-251-7468 FAX: +1-817-416-1439

Canada CNC Service Center
4299 14TH AVENUE MARKHAM, ON. L3R 0J2, CANADA
TEL: +1-905-475-7728 FAX: +1-905-475-7935

Mexico CNC Service Center
MARIANO ESCOBEDO 69 TLALNEPANTLA, 54030 EDO. DE MEXICO
TEL: +52-55-9171-7662 FAX: +52-55-9171-7698

Monterrey CNC Service Satellite
ARGENTINA 3900, FRACC. LAS TORRES, MONTERREY, N.L., 64720, MEXICO
TEL: +52-81-8365-4171 FAX: +52-81-8365-4171

Brazil MITSUBISHI CNC Agent Service Center
(AUTOMOTION IND. COM. IMP. E EXP. LTDA.)
ACESSO JOSE SARTORELLI, KM 2.1 18550-000 BOITUVA – SP, BRAZIL
TEL: +55-15-3363-9900 FAX: +55-15-3363-9911

European FA Center (MITSUBISHI ELECTRIC EUROPE B.V.)

Germany CNC Service Center
GOTHAER STRASSE 8, 40880 RATINGEN, GERMANY
TEL: +49-2102-486-0 FAX: +49-2102486-591

South Germany CNC Service Center
KURZE STRASSE. 40, 70794 FILDERSSTADT-BONLANDEN, GERMANY
TEL: +49-711-3270-010 FAX: +49-711-3270-0141

France CNC Service Center
25, BOULEVARD DES BOUVETS, 92741 NANTERRE CEDEX FRANCE
TEL: +33-1-41-02-83-13 FAX: +33-1-49-01-07-25

Lyon CNC Service Satellite

U.K CNC Service Center
TRAVELLERS LANE, HATFIELD, HERTFORDSHIRE, AL10 8XB, U.K.
TEL: +44-1707-282-846 FAX: +44-1707-278-992

Italy CNC Service Center
ZONA INDUSTRIALE VIA ARCHIMEDE 35 20041 AGRATE BRIANZA, MILANO ITALY
TEL: +39-039-60531-342 FAX: +39-039-6053-206

Spain CNC Service Satellite
CTRA. DE RUBI, 76-90 -APDO.420 08190 SAINT CUGAT DEL VALLES, BARCELONA SPAIN
TEL: +34-935-65-2236 FAX:

Turkey MITSUBISHI CNC Agent Service Center
(GENEL TEKNİK SİSTEMLER LTD. STI.)
DARULACEZE CAD. FAMAS İS MERKEZİ A BLOK NO.43 KAT2 80270 OKMEYDANI İSTANBUL, TURKEY
TEL: +90-212-320-1640 FAX: +90-212-320-1649

Poland MITSUBISHI CNC Agent Service Center (MPL Technology Sp. z o. o)
UL SLICZNA 34, 31-444 KRAKOW, POLAND
TEL: +48-12-632-28-85 FAX:

Wrocław MITSUBISHI CNC Agent Service Satellite (MPL Technology Sp. z o. o)
UL KOBIERZYCKA 23, 52-315 WROCLAW, POLAND
TEL: +48-71-333-77-53 FAX: +48-71-333-77-53

Czech MITSUBISHI CNC Agent Service Center
(AUTOCONT CONTROL SYSTEM S.R.O.)
NEMOCNICNI 12, 702 00 OSTRAVA 2 CZECH REPUBLIC
TEL: +420-596-152-426 FAX: +420-596-152-112

ASEAN FA Center (MITSUBISHI ELECTRIC ASIA PTE. LTD.)

Singapore CNC Service Center
307 ALEXANDRA ROAD #05-01/02 MITSUBISHI ELECTRIC BUILDING SINGAPORE 159943
TEL: +65-6473-2308 FAX: +65-6476-7439

Thailand MITSUBISHI CNC Agent Service Center (F. A. TECH CO., LTD)
898/19,20,21,22 S.V. CITY BUILDING OFFICE TOWER 1 FLOOR 12,14 RAMA III RD BANGPONGPANG, YANNAWA, BANGKOK 10120, THAILAND
TEL: +66-2-682-6522 FAX: +66-2-682-6020

Malaysia MITSUBISHI CNC Agent Service Center
(FLEXIBLE AUTOMATION SYSTEM SDN. BHD.)
60, JALAN USJ 10/1B 47620 UEP SUBANG JAYA SELANGOR DARUL EHSAN MALAYSIA
TEL: +60-3-5631-7605 FAX: +60-3-5631-7636

JOHOR MITSUBISHI CNC Agent Service Satellite
(FLEXIBLE AUTOMATION SYSTEM SDN. BHD.)
NO. 16, JALAN SHAHBANDAR 1, TAMAN UNGUU TUN AMINAH, 81300 SKUDAI, JOHOR MALAYSIA
TEL: +60-7-557-8218 FAX: +60-7-557-3404

Indonesia MITSUBISHI CNC Agent Service Center
(PT. AUTOTEKNINDO SUMBER MAKMUR)
WISMA NUSANTARA 14TH FLOOR JL. M.H. THAMRIN 59, JAKARTA 10350 INDONESIA
TEL: +62-21-3917-144 FAX: +62-21-3917-164

India MITSUBISHI CNC Agent Service Center (MESSUNG SALES & SERVICES PVT. LTD.)
B-36FF, PAVANA INDUSTRIAL PREMISES M.I.D.C., BHOASRI PUNE 411026, INDIA
TEL: +91-20-2711-9484 FAX: +91-20-2712-8115

BANGALORE MITSUBISHI CNC Agent Service Satellite
(MESSUNG SALES & SERVICES PVT. LTD.)
S 615, 6TH FLOOR, MANIPAL CENTER, BANGALORE 560001, INDIA
TEL: +91-80-509-2119 FAX: +91-80-532-9480

Delhi MITSUBISHI CNC Agent Parts Center (MESSUNG SALES & SERVICES PVT. LTD.)
1197, SECTOR 15 PART-2, OFF DELHI-JAIPUR HIGHWAY BEHIND 32ND MILESTONE GURGAON 122001, INDIA
TEL: +91-98-1024-8895 FAX:

Philippines MITSUBISHI CNC Agent Service Center
(FLEXIBLE AUTOMATION SYSTEM CORPORATION)
UNIT NO.411, ALABANG CORPORATE CENTER KM 25, WEST SERVICE ROAD SOUTH SUPERHIGHWAY, ALABANG MUNTINLUPA METRO MANILA, PHILIPPINES 1771
TEL: +63-2-807-2416 FAX: +63-2-807-2417

Vietnam MITSUBISHI CNC Agent Service Center (SA GIANG TECHNO CO., LTD)
47-49 HOANG SA ST. DAKAO WARD, DIST.1 HO CHI MINH CITY, VIETNAM
TEL: +84-8-910-4763 FAX: +84-8-910-2593

China FA Center (MITSUBISHI ELECTRIC AUTOMATION (SHANGHAI) LTD.)

China CNC Service Center
2/F, BLOCK 5 BLDG. AUTOMATION INSTRUMENTATION PLAZA, 103 CAOBAO RD. SHANGHAI 200233, CHINA
TEL: +86-21-6120-0808 FAX: +86-21-6494-0178

Shenyang CNC Service Center
TEL: +86-24-2397-0184 FAX: +86-24-2397-0185

Beijing CNC Service Satellite
9/F, OFFICE TOWER1, HENDERSON CENTER, 18 JIANGUOMENNEI DAJIE, DONGCHENG DISTRICT, BEIJING 100005, CHINA
TEL: +86-10-6518-8830 FAX: +86-10-6518-8030

China MITSUBISHI CNC Agent Service Center
(BEIJING JIAYOU HIGHTECH TECHNOLOGY DEVELOPMENT CO.)
RM 709, HIGH TECHNOLOGY BUILDING NO.229 NORTH SI HUAN ZHONG ROAD, HAIDIAN DISTRICT, BEIJING 100083, CHINA
TEL: +86-10-8288-3030 FAX: +86-10-6518-8030

Tianjin CNC Service Satellite
RM909, TAIHONG TOWER, NO220 SHIZILIN STREET, HEBEI DISTRICT, TIANJIN, CHINA 300143
TEL: +86-22-2653-9090 FAX: +86-22-2635-9050

Shenzhen CNC Service Satellite
RM02, UNIT A, 13/F, TIANAN NATIONAL TOWER, RENMING SOUTH ROAD, SHENZHEN, CHINA 518005
TEL: +86-755-2515-6691 FAX: +86-755-8218-4776

Changchun Service Satellite
TEL: +86-431-50214546 FAX: +86-431-5021690

Hong Kong CNC Service Center
UNIT A, 25/F RYODEN INDUSTRIAL CENTRE, 26-38 TA CHUEN PING STREET, KWAI CHUNG, NEW TERRITORIES, HONG KONG
TEL: +852-2619-8588 FAX: +852-2784-1323

Taiwan FA Center (MITSUBISHI ELECTRIC TAIWAN CO., LTD.)

Taichung CNC Service Center
NO.8-1, GONG YEH 16TH RD., TAICHUNG INDUSTRIAL PARK TAICHUNG CITY, TAIWAN R.O.C.
TEL: +886-4-2359-0688 FAX: +886-4-2359-0689

Taipei CNC Service Satellite
TEL: +886-4-2359-0688 FAX: +886-4-2359-0689

Tainan CNC Service Satellite
TEL: +886-4-2359-0688 FAX: +886-4-2359-0689

Korean FA Center (MITSUBISHI ELECTRIC AUTOMATION KOREA CO., LTD.)

Korea CNC Service Center
DONGSEO GAME CHANNEL BLDG. 2F. 660-11, DEUNGCHON-DONG KANGSEO-KU SEOUL, 157-030 KOREA
TEL: +82-2-3660-9607 FAX: +82-2-3663-0475

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 manual may not be reproduced in any form, in part or in whole, without written permission from Mitsubishi Electric Corporation.

© 2002-2005 MITSUBISHI ELECTRIC CORPORATION
ALL RIGHTS RESERVED.



| | |
|------------|-----------------|
| MODEL | M60/60S series |
| MODEL CODE | 008—221 |
| Manual No. | BNP-B2269D(ENG) |

Specifications subject to change without notice.
Printed in Japan on recycled paper.