CENTER OF TECHNOLOGICAL INITIATIVES, LLC
info@cti-sumy.com.ua

Custom-made machines for you

Welding machine

Welding machine

Welding machine

Semi-automatic welding machine

Contact us:

http://cti-sumy.com.ua
info@cti-sumy.com.ua
sales1@cti-sumy.com.ua
https://www.facebook.com/cti.sumy
https://www.youtube.com/channel/UCyYGmKVZ2FwZnoX4RV8Is0g

This manual is brought to you by CTI LLC.

We have more than 10 years of experience in machine building and will gladly help you to increase productivity at your manufacturing facility!

Taking into account the peculiarities of your manufacture we can make a custom machine that will help you simplify the production process.

We offer to perform the whole range of works:

Designing Production Assembly

GE Fanuc Automation

Computer Numerical Control Products

PMC Model PA1/PA3/SA1/SA2/SA3/SA5/SB/SB2/ SB3/SB4/SB5/SB6/SC/SC3/SC4/NB/NB2/NB6 Ladder Language

Programming Manual

GFZ-61863E/12 March 1999

Warnings, Cautions, and Notes as Used in this Publication

Warning

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

Caution

Caution notices are used where equipment might be damaged if care is not taken.

Note

Notes merely call attention to information that is especially significant to understanding and operating the equipment.

This document is based on information available at the time of its publication. While efforts have been made to be accurate, the information contained herein does not purport to cover all details or variations in hardware or software, nor to provide for every possible contingency in connection with installation, operation, or maintenance. Features may be described herein which are not present in all hardware and software systems. GE Fanuc Automation assumes no obligation of notice to holders of this document with respect to changes subsequently made.

GE Fanuc Automation makes no representation or warranty, expressed, implied, or statutory with respect to, and assumes no responsibility for the accuracy, completeness, sufficiency, or usefulness of the information contained herein. No warranties of merchantability or fitness for purpose shall apply.

DEFINITION OF WARNING, CAUTION, AND NOTE

This manual includes safety precautions for protecting the user and preventing damage to the machine. Precautions are classified into Warning and Caution according to their bearing on safety. Also, supplementary information is described as a Note. Read the Warning, Caution, and Note thoroughly before attempting to use the machine.

WARNING

Applied when there is a danger of the user being injured or when there is a danger of both the user being injured and the equipment being damaged if the approved procedure is not observed.

CAUTION

Applied when there is a danger of the equipment being damaged, if the approved procedure is not observed.

NOTE

The Note is used to indicate supplementary information other than Warning and Caution.

❍ **Read this manual carefully, and store it in a safe place.**

PREFACE

This programming manual describes the method of generating ladder sequence programs for PMC.

It also describes the operation methods of CRT/MDI and SYSTEM P series for sequence programming.

Renaming of PMC Models

This manual presents programming descriptions for the PMC models listed in the following table. Note that some models have been renamed; in the product name column, the old names are enclosed in parentheses, while the new names appear above the old names. However, the previous specifications are still applied to the renamed models. Thus, when using the renamed models, users should:

- Read the old names shown in this manual as the new names.
- Read the old names appearing on the units as the new names.

Applicable models

The models covered by this manual, and their abbreviations are :

Product Name	Abbreviations	Applicable CNC
FANUC PMC–MODEL PA1	PMC–PA1	FANUC Power Mate–MODEL D FANUC Series 21–MODEL A
FANUC PMC–MODEL PA3	PMC–PA3	FANUC Power Mate–MODEL D/F/H FANUC Series 21–MODEL A
FANUC PMC–MODEL SA1 (Note 1) (Old Name : FANUC PMC–MODEL RA1)	PMC–SA1 (PMC–RA1)	FANUC Series 18–MODEL A/B FANUC Series 20 FANUC Series 21–MODEL B FANUC Series 21*i*–MODEL A Loader control function (Note 2)
FANUC PMC–MODEL SA2 (Note 1) (Old Name : FANUC PMC–MODEL RA2)	PMC–SA2 (PMC–RA2)	FANUC Series 18–MODEL A
FANUC PMC–MODEL SA3 (Note 1) (Old Name : FANUC PMC–MODEL RA3)	PMC–SA3 (PMC–RA3)	FANUC Series 18–MODEL A FANUC Series 20 FANUC Series 21–MODEL B
FANUC PMC–MODEL SA5 (Note 1) (Old Name : FANUC PMC–MODEL RA5)	PMC–SA5 (PMC–RA5)	FANUC Series 21*i*–MODEL A
FANUC PMC–MODEL SB (Note 1) (Old Name : FANUC PMC–MODEL RB)	PMC–SB (PMC–RB)	FANUC Series 16–MODEL A
FANUC PMC–MODEL SB2 (Note 1) (Old Name : FANUC PMC–MODEL RB2)	PMC–SB2 (PMC–RB2)	
FANUC PMC–MODEL SB3 (Note 1) (Old Name : FANUC PMC–MODEL RB3)	PMC–SB3 (PMC–RB3)	FANUC Series 16–MODEL A/B FANUC Series 18–MODEL B
FANUC PMC–MODEL SB4 (Note 1) (Old Name : FANUC PMC–MODEL RB4)	PMC–SB4 (PMC–RB4)	FANUC Series 16–MODEL B FANUC Series 18–MODEL B

Product Name	Abbreviations	Applicable CNC
FANUC PMC–MODEL SB5 (Note 1) (Old Name : FANUC PMC–MODEL RB5)	PMC–SB5 (PMC–RB5)	FANUC Series 16–MODEL C FANUC Series 18–MODEL C FANUC Series 16*i*–MODEL A FANUC Series 18*i*–MODEL A FANUC Power Mate *i*–MODEL D/H
FANUC PMC–MODEL SB6 (Note 1) (Old Name : FANUC PMC–MODEL RB6)	PMC–SB6 (PMC–RB6)	
FANUC PMC–MODEL SC (Note 1) (Old Name : FANUC PMC–MODEL RC)	PMC–SC (PMC–RC)	FANUC Series 16–MODEL A
FANUC PMC–MODEL SC3 (Note 1) (Old Name : FANUC PMC–MODEL RC3)	PMC–SC3 (PMC–RC3)	FANUC Series 16–MODEL A/B/C FANUC Series 18–MODEL B/C
FANUC PMC–MODEL SC4 (Note 1) (Old Name : FANUC PMC–MODEL RC4)	PMC–SC4 (PMC–RC4)	FANUC Series 16–MODEL B/C FANUC Series 18–MODEL B/C
FANUC PMC–MODEL NB	PMC–NB	FANUC Series 15–MODEL B
FANUC PMC–MODEL NB2	PMC–NB2	
FANUC PMC–MODEL NB6	PMC–NB6	FANUC Series 15*i*–MODEL A

NOTE

1 These models have been renamed; in the product name column, the old names are enclosed in parentheses, while the new names appear above the old names. However, the previous specifications are still applied to the renamed models.
Thus, when using the renamed models, users should :
- Read the old names shown in this manual as the new names.
- Read the old names appearing on the units as the new names.

2 PMC–SA1 is applied to the loader control side of a CNC having the loader control function.
The CNC models having the loader control function are as follows :
FANUC Series 16–MODEL A/B/C
FANUC Series 18–MODEL A/B/C
FANUC Series 21–MODEL B
FANUC Series 16*i*–MODEL A
FANUC Series 18*i*–MODEL A

Other manuals

However, it does not include all items required for sequence programming. For those required for sequence programming refer to the following manuals.

Name of manual	Reference items	Application
FANUC Power Mate-MODEL D/F CONNECTION MANUAL (B-62833EN)	Interface between PMC and CNC	PMC-PA1 PMC-PA3
FANUC Power Mate-MODEL H CONNECTION MANUAL (B–62683EN)	Interface between PMC and CNC	PMC-PA3
FANUC Power Mate *i*-MODEL D/H CONNECTION MANUAL (FUNCTION) (B–63733EN–1)	Interface between PMC and CNC	PMC-SB5 PMC-SB6
FANUC Series 16/18 CONNECTION MANUAL (B-61803E)	Interface between PMC and CNC	PMC-SA1 PMC-SA2 PMC-SA3 PMC-SB PMC-SB2 PMC-SB3 PMC-SC PMC-SC3
FANUC Series 16/18/160/180-MODEL B CONNECTION MANUAL (FUNCTION) (B-62443E-1)	Interface between PMC and CNC	PMC-SB3 PMC-SB4 PMC-SC3 PMC-SC4
FANUC Series 16/18/160/180-MODEL C CONNECTION MANUAL (FUNCTION) (B-62753EN-1)	Interface between PMC and CNC	PMC-SB5 PMC-SB6 PMC-SC3 PMC-SC4
FANUC Series 16*i*/18*i*/21*i*/160*i*/180*i*/210*i*–MODEL A CONNECTION MANUAL (FUNCTION) (B-63003EN-1)	Interface between PMC and CNC	PMC-SA1 PMC-SA5 PMC-SB5 PMC-SB6
FANUC Series 20–FA/TA CONNECTION MANUAL (B–62173E)	Interface between PMC and CNC	PMC-SA1 PMC-SA3
FANUC Series 21/210–MODEL B CONNECTION MANUAL (FUNCTION) (B–62703EN–1)	Interface between PMC and CNC	PMC-SA1 PMC-SA3
FANUC Series 15-MODEL B BMI INTERFACE CONNECTION MANUAL (B-62073E-1)	Interface between PMC and CNC	PMC-NB PMC-NB2
FANUC PMC-MODEL RC/RC3/RC4/NB C LANGUAGE PROGRAMMING MANUAL (B-61863E-1)	C language programming	PMC-SC PMC-SC3 PMC-SC4 PMC-NB PMC-NB2

Other application model names

The models covered for reference by this manual, and their abbreviations are:

Product name	Abbreviation	CNC for
FANUC PMC-MODEL P	PMC-P	FANUC Power Mate-MODEL C
FANUC PMC-MODEL NA	PMC-NA	FANUC Series 15-MODEL B

Table of Contents

Volume 1

II. PMC OPERATION (CRT/MDI)

III. PMC PROGRAMMER (CRT/MDI)

V.PMC PROGRAMMER (SYSTEM P series)

Volume 2

APPENDIX

I. PMC SEQUENCE PROGRAM

1 SEQUENCE PROGRAM CREATING PROCEDURE

The procedure for creating the sequence program when the CNC machine tool is controlled by use of the PMC is shown in Fig. 1.
Proceed according to the flow shown in Fig. 1.
The procedure is briefly explained below.

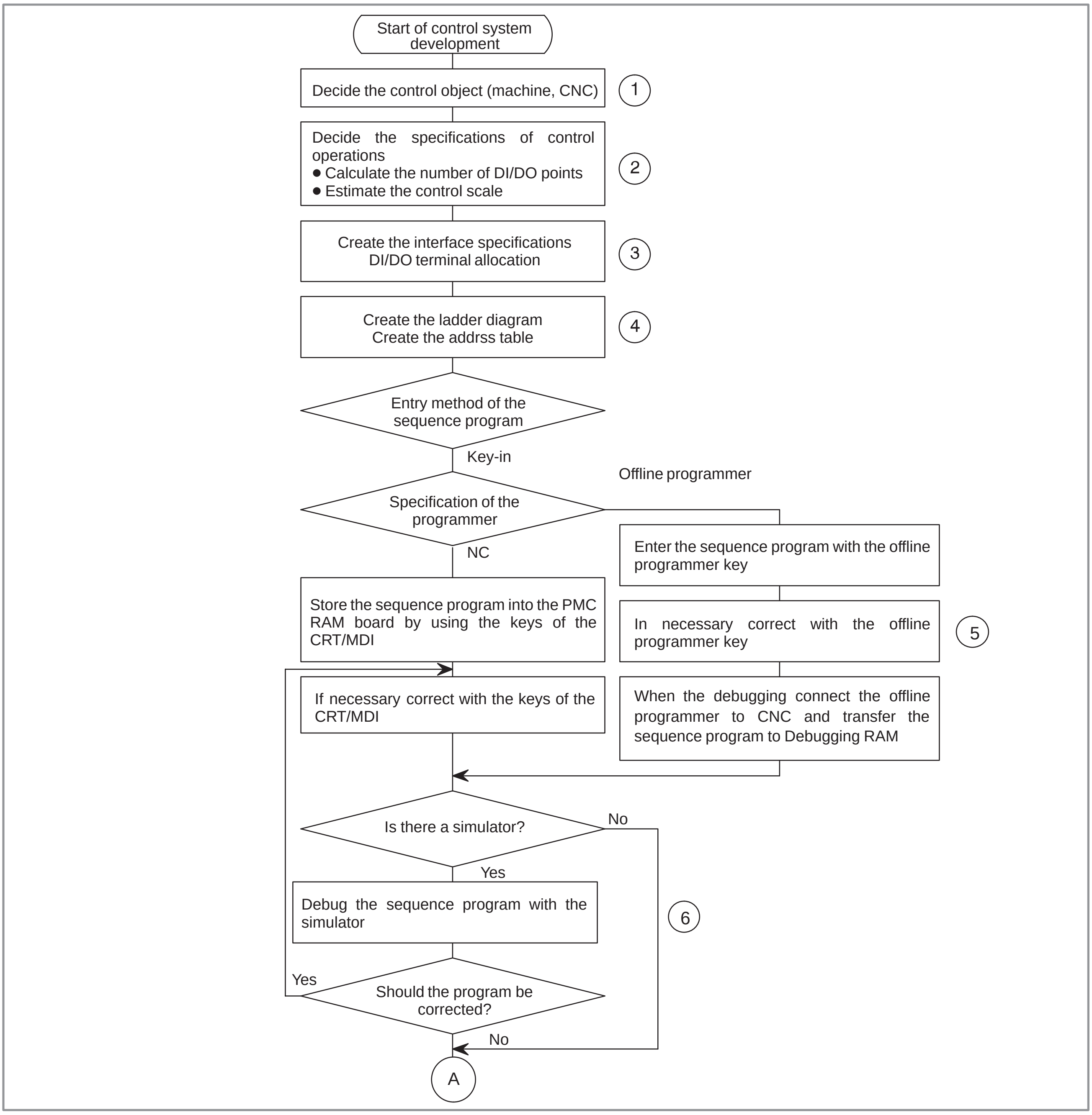


Fig. 1 Sequence program creating procedure (1/2)

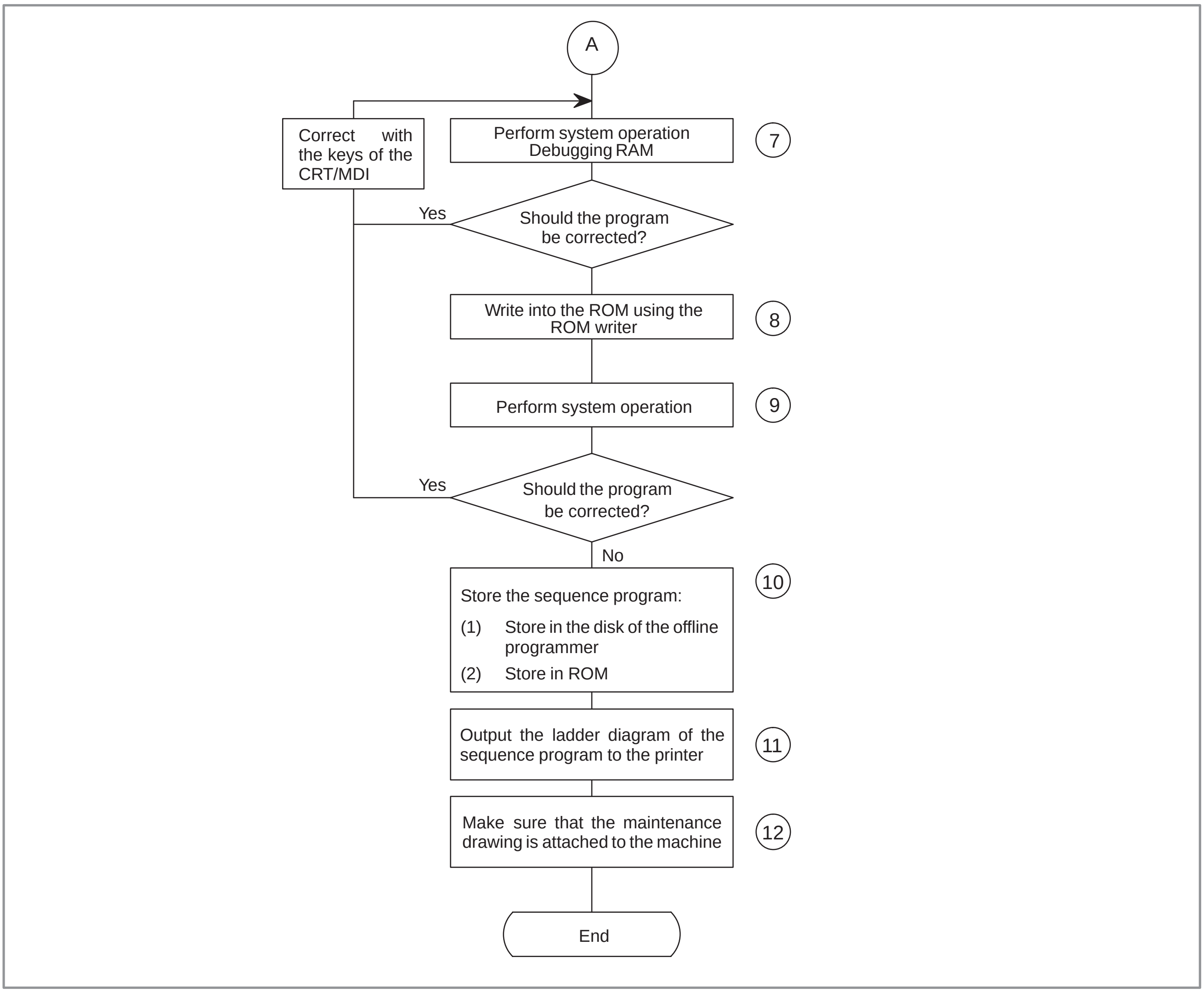


Fig. 1 Sequence program creating procedure (2/2)

1.1 SPECIFICATION OF PMCs

Table 1.1 shows the specification of PMCs.
Note that the program size, processing speed, available function commands, internal addresses, and nonvolatile memory addresses of some PMCs are different from those of other PMCs.

Table 1.1 PMC specifications (1)

Specification of PMC \ Type of PMC	PMC-PA1	PMC-PA3
Program method language	Ladder	Ladder
Number of ladder level	2	2
1st level execution period	8 ms	8 ms
Mean processing time of basic command	4. 5 (μs/ step)	* 0. 15 (μs/ step)
Program capacity ● Ladder (step) ● Symbol, Comment (Note 1) ● Message ● Language only	 Approx. 3, 000 1 to 128KB 0.1 to 64KB –	 Approx. 5, 000 Approx. 12,000 (Only for Power Mate D/H) 1 to 128KB 0.1 to 64KB –
Command Basic command Function command	12 kinds 47 kinds	14 kinds 64 kinds
Internal relay (R) Message request (A) Keepmemory ● Variable timer (T) ● Counter (C) ● Keep relay (K) ● Data table (D) Subprogram (P) Label (L) Fixed timer	1100 byte 25 byte 80 byte 80 byte 20 byte 1860 byte – – Timer No. 100 devices specified	1118 byte 25 byte 80 byte 80 byte 20 byte 1860 byte 512 programs 9999 labels Timer No. 100 devices specified
I/O ● I/O Link (Note 2) (I) (Master) (O) ● I/O Link (I) (Slave) (O) ● I/O card (I) (O)	 1024 points max. 1024 points max. 64 points max. 64 points max. 32 points max. 24 points max.	 1024 points max. 1024 points max. 64 points max. 64 points max. 32 points max. 24 points max.
Sequence program (Note 3)	SRAM	SRAM

● **Power Mate D Data size of each modules**

MEMORY-MODULE	Total capacity	Program size	
		One–Path control	Two–Path control
A	256KB	62KB	24KB
B	512KB	128KB	64KB
C	768KB	128KB	128KB

It is impossible that make the data more than the total capacity of each modules.

● **Power Mate F**

Program size
64KB

● **Power Mate H**

Program size
128KB

NOTE

1 The size of a symbol and that of a comment are fixed to 1KB.
The size of a message is fixed to 0.1KB.
The maximum size of a symbol and that of a comment are 64KB each.
2 I/O Link Master function is not available in the Power Mate–MODEL F.
3 FLASH ROM is used in the Power Mate–MODEL H.
4 As values indicated with an asterisk (*) in the table, former versions of the programming manual and catalogs have listed the mean processing time of basic commands, but this manual lists the execution time for one step. The actual ladder program execution performance (speed) of each PMC has not been changed.
5 Up to 256/256 points of Input/Output points are available or I/O Link (Slave) in the Power Mate–MODEL D/H.

Table 1.1 PMC specifications (2)

Type of PMC / Specification of PMC	PMC–SA1	PMC–SA2	PMC–SA3
Program method language	Ladder	Ladder	Ladder
Number of ladder level	2	2	2
1st level execution period	8 ms	8 ms	8 ms
Mean processing time of basic command	5.0 (μs/ step)	1.5 (μs/ step)	* 0. 15 (μs/ step)
Program capacity ● Ladder (step) ● Symbol, Comment (Note 1) ● Message ● Language only	 Approx. 3, 000 Approx. 5, 000 1 to 128KB 0. 1 to 64KB –	 Approx. 3, 000 Approx. 5, 000 Approx. 8, 000 Approx.12, 000 1 to 128KB 0. 1 to 64KB –	 Approx. 3, 000 Approx. 5, 000 Approx. 8, 000 Approx.12, 000 1 to 128KB 0. 1 to 64KB –
Command Basic command Function command	12 kinds 49 kinds	12 kinds 48 kinds	14 kinds 66 kinds
Internal relay (R) Message request (A) Keep memory ● Variable timer (T) ● Counter (C) ● Keep relay (K) ● Data table (D) Subprogram (P) Label (L) Fixed timer	1100 byte 25 byte 80 byte 80 byte 20 byte 1860 byte – – Timer No. 100 devices specified	1118 byte 25 byte 80 byte 80 byte 20 byte 1860 byte – – Timer No. 100 devices specified	1118 byte 25 byte 80 byte 80 byte 20 byte 1860 byte 512 programs 9999 labels Timer No. 100 devices specified
I/O ● I/O link (I) (O) ● I/O card (I) (O)	 1024 points max. 1024 points max. 156 points max. 120 points max.	 1024 points max. 1024 points max. 156 points max. 120 points max.	 1024 points max. 1024 points max. 156 points max. 120 points max.
Sequence program	EPROM 1Mbit×1 (128KB) (Note 2)	EPROM 1Mbit×1 (128KB)	EPROM 1Mbit×1 (128KB) (Note 2)

NOTE

1 The size of a symbol and that of a comment are fixed to 32KB.
The size of a message is fixed to 2.1KB.
The maximum size of a symbol and that of a comment are 64KB each.

2 FLASH ROM is used in the FANUC Series 20.

3 As values indicated with an asterisk (*) in the table, former versions of the programming manual and catalogs have listed the mean processing time of basic commands, but this manual lists the execution time for one step. The actual ladder program execution performance (speed) of each PMC has not been changed.

4 Application PMC for FANUC Series 16–MODEL A loader control function is PMC–SA1.

Table 1.1 PMC specifications (3)

Specification of PMC \ Type of PMC	PMC–SB1	PMC–SB2	PMC–SB3
Program method language	Ladder	Ladder	Ladder
Number of ladder level	2	2	2
1st level excution period	8 ms	8 ms	8 ms
Mean processing time of basic command	1.0 (µs/ step)	1.0 (µs/ step)	* 0. 15 (µs/ step)
Program capacity ● Ladder (step) ● Symbol, Comment (Note 1) ● Message ● Language only	 Approx. 5, 000 Approx. 8, 000 Approx.12, 000 Approx.16, 000 1 to 128KB 0.1 to 64KB –	 Approx. 5, 000 Approx. 8, 000 Approx.12, 000 Approx.16, 000 Approx.24, 000 1 to 128KB 0.1 to 64KB –	 Approx. 5, 000 Approx. 8, 000 Approx.12, 000 Approx.16, 000 Approx.24, 000 1 to 128KB 0.1 to 64KB –
Command Basic command Function command	12 kinds 49 kinds	12 kinds 49 kinds	14 kinds 68 kinds
Internal relay (R) Message request (A) Keep memory ● Variable timer (T) ● Counter (C) ● Keep relay (K) ● Data table(D) (D) Subprogram (P) Label (L) Fixed timer	1100 byte 25 byte 80 byte 80 byte 20 byte 1860 byte – – Timer No. 100 devices specified	1118 byte 25 byte 80 byte 80 byte 20 byte 1860 byte – – Timer No. 100 devices specified	1618 byte 25 byte 80 byte 80 byte 20 byte 3000 byte 512 programs 9999 labels Timer No. 100 devices specified
I/O ● I/O link (I) (O) ● I/O card (I) (O)	 1024 points max. 1024 points max. 156 points max. 120 points max.	 1024 points max. 1024 points max. 156 points max. 120 points max.	 1024 points max. 1024 points max. 156 points max. 120 points max.
Sequence program	EPROM 1Mbit×1 (128KB)	EPROM 1Mbit×1 (128KB) ROM MODULE 256KB (Note 2)	EPROM 1Mbit×1 (128KB) ROM MODULE 256KB (Note 2)

NOTE

1 The size of a symbol and that of a comment are fixed to 32KB.
The size of a message is fixed to 2.1KB.
The maximum size of a symbol and that of a comment are 64KB each.

2 When the number of steps of the PMC-SB2, SB3 ladder program is approx. 24,000, the capacity of the ROM module must be 256KB.

3 As values indicated with an asterisk (*) in the table, former versions of the programming manual and catalogs have listed the mean processing time of basic commands, but this manual lists the execution time for one step. The actual ladder program execution performance (speed) of each PMC has not been changed.

Table 1.1 PMC specifications (4)

Specification of PMC \ Type of PMC	PMC–SC	PMC–SC3	PMC–NB
Program method language	Ladder C-language	Ladder C-language	Ladder C-language
Number of ladder level	3	3	3
1st level execution period	8 ms	8 ms	8 ms
Mean processing time of basic command	0.15 (µs/ step)	0.15 (µs/ step)	0.15 (µs/ step)
Program capacity ● Ladder (step)	 Approx. 16,000 Approx. 24,000	 Approx. 16,000 Approx. 24,000	 Approx. 8,000 Approx. 16,000 (Note 2) Approx. 24,000 (Note 2)
● Symbol, Comment (Note 1)	1 to 128KB	1 to 128KB	1 to 128KB
● Message	0.1 to 64KB	0.1 to 64KB	0.1 to 64KB
● Language only	896KB max.	896KB max.	896KB max.
Command Basic command	12 kinds	14 kinds	14 kinds
Function command	51 kinds	68 kinds	68 kinds
Internal relay (R)	1600 byte	1618 byte	1618 byte
Message request (A)	25 byte	25 byte	25 byte
Keepmemor			
● Variable timer (T)	80 byte	80 byte	80 byte
● Counter (C)	80 byte	80 byte	80 byte
● Keep relay (K)	20 byte	20 byte	20 byte
● Data table (D)	3000 byte	3000 byte	3000 byte
Subprogram (P)	–	512 programs	512 programs
Label (L)	–	9999 labels	9999 labels
Fixed timer	Timer No. 100 devices specified	Timer No. 100 devices specified	Timer No. 100 devices specified
I/O ● I/O link (I)	1024 points max.	1024 points max.	1024 points max.
(O)	1024 points max.	1024 points max.	1024 points max.
● I/O card (I)	156 points max.	156 points max.	–
(O)	120 points max.	120 points max.	–
Sequence program	ROM MODULE 128KB 256KB 512KB 1MB	ROM MODULE 128KB 256KB 512KB 1MB	ROM MODULE 64KB 128KB 256KB 512KB 1MB

NOTE

1 The size of a symbol and that of a comment of PMC-SC/SC3 are fixed 32KB. The size of message of PMC-SC/SC3 is fixed 2.1KB. The size of a symbol and that of a comment of PMC-NB are fixed 28KB. The size of message of PMC-NB is fixed 2.1KB. The maximum size of a symbol and that of a comment are 64KB each.

2 When the number of steps of the PMC-NB ladder program is not less than 8,000, the OPTION DRAM is required. (A02B-0162-J151, J152)

Table 1.1 PMC specifications (5)

Model	Series 16–MODEL B/Series 18–MODEL B				Series 18–MODEL B
	PMC–SB3	PMC–SC3	PMC–SB4	PMC–SC4	PMC–SA1
Programming method language	Ladder	Ladder C–language	Ladder Step sequence	Ladder C–language Step sequece	Ladder
Number of ladder level	2	3	2	3	2
Level-1 Cycle Time	8 ms	8 ms	8 ms	8 ms	8 ms
Basic Instruction Execution Time	* 0.1 (µs/ step)	0.1 (µs/ step)	* 0.1 (µs/ step)	0.1 (µs/ step)	5.0 (µs/ step)
Program capacity					
● Ladder (step)	Approx. 5, 000 Approx. 8, 000 Approx.12, 000 Approx.16, 000 Approx.24, 000	Approx.16, 000 Approx.24, 000	Approx. 5, 000 Approx. 8, 000 Approx.12, 000 Approx.16, 000 Approx.24, 000	Approx.16, 000 Approx.24, 000	Approx. 3, 000 Approx. 5, 000
● Symbol/Comment	1 to 128KB	1 to 128KB	1 to 128KB	1 to 128KB	1 to 128KB
● Message					
● Language only	0.1 to 64KB –	0.1 to 64KB max. 896KB	0.1 to 64KB	0.1 to 64KB max. 896KB	0.1 to 64KB –
Instruction (Basic) (Functional)	14 kinds 67 kinds	14 kinds 69 kinds	14 kinds 67 kinds	14 kinds 69 kinds	12 kinds 49 kinds
Internal relay (R)	1618 byte	1618 byte	3200 byte	3200 byte	1100 byte
Message request (A)	25 byte	25 byte	125 byte	125 byte	25 byte
Non-volatile					
● Var. Timer (T)	80 byte	80 byte	300 byte	300 byte	80 byte
● Counter (C)	80 byte	80 byte	200 byte	200 byte	80 byte
● Keep relay (K)	20 byte	20 byte	50 byte	50 byte	20 byte
● Data table (D)	3000 byte	3000 byte	8000 byte	8000 byte	1860 byte
Subprogram (P)	512 programs	512 programs	2000 programs	2000 programs	–
Label (L)	9999 labels	9999 labels	9999 labels	9999 labels	–
Fixed timer	Timer No. 100 devices specified	Timer No. 100 devices specified	Timer No. 100 devices specified	Timer No. 100 devices specified	Timer No. 100 devices specified
Input/output					
● I/O link (I) Max.	1024 points max.	1024 points max.	1024 points max.	1024 points max.	1024 points max.
(O) Max.	1024 points max.	1024 points max.	1024 points max.	1024 points max.	1024 points max.
● I/O card (I) Max.	312 points max.	312 points max.	312 points max.	312 points max.	312 points max.
(Note) (O) Max.	240 points max.	240 points max.	240 points max.	240 points max.	240 points max.
Sequence program storage media	Flash ROM 128KB 256KB	Flash ROM 128KB 256KB 512KB 1MB	Flash ROM 128KB 256KB	Flash ROM 128KB 256KB 512KB 1MB	Flash ROM 128KB

NOTE

1. The size of a symbol and that of a comment are fixed 32KB. The size of message is fixed 2.1KB. The maximum size of a symbol and that of a comment are 64KB each.
2. That is the maximum number when 2 I/O cards (with 156 inputs/120 outputs) are used.
3. As values indicated with an asterisk (*) in the table, former versions of the programming manual and catalogs have listed the mean processing time of basic commands, but this manual lists the execution time for one step. The actual ladder program execution performance (speed) of each PMC has not been changed.
4. Application PMC for FANUC Series 16–MODEL B loader control function is PMC–SA1.

Table 1.1 PMC specifications (6)

Model	Series 16–MODEL C/Series 18–MODEL C			
	PMC–SB5	**PMC–SC3**	**PMC–SB6**	**PMC–SC4**
Programming method language	Ladder	Ladder C–language	Ladder Step sequence	Ladder C–language Step sequece
Number of ladder level	2	3	2	3
Level-1 Cycle Time	8 ms	8 ms	8 ms	8 ms
Basic Instruction Execution Time	* 0.1 (μs/ step)	0.1 (μs/ step)	* 0.1 (μs/ step)	0.1 (μs/ step)
Program capacity ● Ladder (step) ● Symbol/Comment ● Message ● Language only	 Approx. 3, 000 Approx. 5, 000 Approx. 8, 000 Approx.12, 000 Approx.16, 000 Approx.24, 000 1 to 128KB 0.1 to 64KB –	 Approx.16, 000 Approx.24, 000 1 to 128KB 0.1 to 64KB max. 896KB	 Approx. 3, 000 Approx. 5, 000 Approx. 8, 000 Approx.12, 000 Approx.16, 000 Approx.24, 000 Approx.32, 000 1 to 128KB 0.1 to 64KB –	 Approx.16, 000 Approx.24, 000 Approx.32, 000 1 to 128KB 0.1 to 64KB max. 896KB
Instruction (Basic) (Functional)	14 kinds 67 kinds	14 kinds 69 kinds	14 kinds 67 kinds	14 kinds 69 kinds
Internal relay (R) Message request (A) Non-volatile ● Var. Timer (T) ● Counter (C) ● Keep relay (K) ● Data table (D) Subprogram (P) Label (L) Fixed timer	1618 byte 25 byte 80 byte 80 byte 20 byte 3000 byte 512 programs 9999 labels Timer No. 100 devices specified	1618 byte 25 byte 80 byte 80 byte 20 byte 3000 byte 512 programs 9999 labels Timer No. 100 devices specified	3200 byte 125 byte 300 byte 200 byte 50 byte 8000 byte 2000 programs 9999 labels Timer No. 100 devices specified	3200 byte 125 byte 300 byte 200 byte 50 byte 8000 byte 2000 programs 9999 labels Timer No. 100 devices specified
Input/output ● I/O link (I) Max. (O) Max. ● I/O card (I) Max. (Note) (O) Max.	 1024 points max. 1024 points max. 312 points max. 240 points max.	 1024 points max. 1024 points max. 312 points max. 240 points max.	 1024 points max. 1024 points max. 312 points max. 240 points max.	 1024 points max. 1024 points max. 312 points max. 240 points max.
Sequence program storage media	Flash ROM 128KB 256KB	Flash ROM 128KB 256KB 512KB 1MB	Flash ROM 128KB 256KB	Flash ROM 128KB 256KB 512KB 1MB

NOTE

1 The size of a symbol and that of a comment are fixed 32KB. The size of message is fixed 2.1KB. The maximum size of a symbol and that of a comment are 64KB each.
2 That is the maximum number when 2 I/O cards (with 156 inputs/120 outputs) are used.
3 Application PMC for FANUC Series 16–MODEL C loader control function is PMC–SA1.

Table 1.1 PMC specifications (7)

Model	Series 21–MODEL B/ Series 210–MODEL B	
	PMC–SA1	**PMC–SA3**
Programming method language	Ladder	Ladder
Number of ladder level	2	2
1st level excution period	8 ms	8 ms
Mean processing time of basic command	5.0 (µs/ step)	* 0.15 (µs/ step)
Program capacity ● Ladder (step) ● Symbol/Comment (Note 3) ● Message ● Language only	 Approx. 3, 000 Approx. 5, 000 1 to 128KB 0.1 to 64KB –	 Approx. 3, 000 Approx. 5, 000 Approx. 8, 000 Approx.12, 000 1 to 128KB 0.1 to 64KB –
Command Basic command Function command	12 kinds 49 kinds	14 kinds 66 kinds
Internal relay (R) Message request (A) Keep memory ● Variable timer (T) ● Counter (C) ● Keep relay (K) ● Data table (D) Subprogram (P) Label (L) Fixed timer	1100 byte 25 byte 80 byte 80 byte 20 byte 1860 byte – – Timer No. 100 devices specified	1118 byte 25 byte 80 byte 80 byte 20 byte 1860 byte 512 programs 9999 labels Timer No. 100 devices specified
I/O ● I/O link (I) (O) ● I/O card (I) (O)	 1024 points max. 1024 points max. 96 points max. 72 points max. (Note 3)	 1024 points max. 1024 points max. 96 points max. 72 points max. (Note 3)
Sequence program	Flash ROM 128KB	Flash ROM 128KB

NOTE

1 The size of a symbol and that of a comment are fixed 32KB. The size of message is fixed 2.1KB. The maximum size of a symbol and that of a comment are 64KB each.
2 When extended memory is not specified in the 4082 series (ordering drawing No.: A02B–0210–H020 or A02B–0210–H022), the program capacity is 64KB.
3 Output points of I/O card in 4082 series are following ; PMC–SA1 : 64points, PMC–SA3 : 64points
4 As values indicated with an asterisk (*) in the table, former versions of the programming manual and brochure have listed the mean processing time of basic commands, but this manual lists the execution time for one step. The actual ladder program execution performance (speed) of each PMC has not been changed.
5 Application PMC for FANUC Series 21–B loader control function is PMC–SA1.

Table 1.1 PMC specifications (8)

Model	Series 16*i*/18*i*/160*i*/180*i*	
	PMC–SB5	PMC–SB6
Programmingmethod	Ladder	Ladder step sequence
Number of ladder levels	2	2
First–level execution period	8 ms	8 ms
Basic instruction processing time	0.085 (μ sec/step)	0.085 (μ sec/step)
Program capacity ● Ladder (step) ● Symbol Comment ● Message	 About 3,000 About 5,000 About 8,000 About 12,000 About 16,000 About 24,000 1KB to 128KB 0.1KB to 64KB	 About 3,000 About 5,000 About 8,000 About 12,000 About 16,000 About 24,000 About 32,000 1KB to 128KB 0.1KB to 64KB
Instruction (Basic instruction) (Functionalinstruction)	14 67	14 67
Internal relay (R) Message request (A) Nonvolatile memory ● Variable timer (T) ● Counter (C) ● Keep replay (K) ● Data table (D) Subprogram (P) Label (L) Fixed timer	1618 bytes 25 bytes (200 points) 80 bytes (40 each) 80 bytes (20 each) 20 bytes 3000 bytes 512 each 9999 each 100 each (Timer number specification)	3200 bytes 125 bytes (1000 points) 300 bytes (150 each) 200 bytes (50 each) 50 bytes 8000 bytes 2000 each 9999 each 100 each (Timer number specification)
I/O ● I/O link (Input) (Note 2) (Output) ● Built–in I/O card (Input) (Output)	 1024 points maximum 1024 points maximum – –	 1024 points maximum 1024 points maximum – –
Sequence program storage media	Flash ROM 128KB 256KB	Flash ROM 128KB 256KB 384KB

NOTE

1 The PMC–SA1 can be used with the loader control function of the FANUC Series 16*i*/18*i*/21*i*/160*i*/180*i*/210*i*.

2 For I/O of the FANUC Series 16*i*/18*i*/21*i*/160*i*/180*i*/210*i*, only the I/O link is used.

Table 1.1 PMC specifications (9)

Model	Series 21*i*/210*i*	
	PMC–SA1 (Note 1)	**PMC–SA5**
Programming method	Ladder	Ladder
Number of ladder levels	2	2
First–level execution period	8 ms	8 ms
Basic instruction processing time	5.0 (μ sec/ step)	0.085 (μ sec/ step)
Program capacity ● Ladder (step) ● Symbol Comment ● Message	 About 3,000 About 5,000 1KB to 128KB 0.1KB to 64KB	 About 3,000 About 5,000 About 8,000 About 12,000 About 16,000 1KB to 128KB 0.1KB to 64KB
Instruction (Basic instruction) (Functional instruction)	12 kinds 49 kinds	14 kinds 66 kinds
Internal relay (R) Message request (A) Nonvolatile memory ● Variable timer (T) ● Counter (C) ● Keep replay (K) ● Data table (D) Subprogram (P) Label (L) Fixed timer	1100 bytes 25 bytes (200 points) 80 bytes (40 each) 80 bytes (20 each) 20 bytes 1860 bytes – – 100 each (Timer number specification)	1118 bytes 25 bytes (200 points) 80 bytes (40 each) 80 bytes (20 each) 20 bytes 1860 bytes 512 each 9999 each 100 each (Timer number specification)
I/O ● I/O link (Input) (Note 2) (Output) ● Built–in I/O card (Input) (Output)	 1024 points maximum 1024 points maximum – –	 1024 points maximum 1024 points maximum – –
Sequence program storage media	Flash ROM 128KB	Flash ROM 128KB

NOTE

1 The PMC–SA1 can be used with the loader control function of the FANUC Series 16*i*/18*i*/21*i*/160*i*/180*i*/210*i*.

2 For I/O of the FANUC Series 16*i*/18*i*/21*i*/160*i*/180*i*/210*i*, only the I/O link is used.

Table 1.1 PMC specifications (10)

Model	FUNAC Power Mate *i*–MODEL D/H	
	PMC–SB5	PMC–SB6
Programmingmethod	Ladder	Ladder step sequence
Number of ladder levels	2	2
Level–1 cycle time	8 ms	8 ms
Basic instruction execution time	0.085 (μ sec/step)	0.085 (μ sec/step)
Program capacity ● Ladder (step)	Approx. 5,000 Approx. 12,000 Approx. 16,000 Approx. 24,000	Approx. 5,000 Approx. 12,000 Approx. 16,000 Approx. 24,000 Approx. 32,000
● Symbol/Comment ● Message	1 to 128KB 0.1 to 64KB	1 to 128KB 0.1 to 64KB
Instruction (Basic) (Functional)	14 kinds 67 kinds	14 kinds 67 kinds
Internal relay (R) Message request (A) Non–volatile ● Var.timer (T) ● Counter (C) ● Keep replay (K) ● Data table (D) Subprogram (P) Label (L) Fixed timer	1618 bytes 25 bytes 80 bytes 80 bytes 20 bytes 3000 bytes 512 programs 9999 labels Timer No.100 devices specified	3200 bytes 125 bytes 300 bytes 200 bytes 50 bytes 8000 bytes 2000 programs 9999 labels Timer No.100 devices specified
Input/Output ● I/O Link (I) Max. (master) (O) Max. ● I/O Link (I) Max. (slave) (O) Max. ● Built–in I/O (I) Max. (O) Max.	 1024 points max. 1024 points max. 256 points max. 256 points max. 32 points max. 24 points max.	 1024 points max. 1024 points max. 256 points max. 256 points max. 32 points max. 24 points max.
Sequence program storage media	Flash ROM 128KB 256KB	Flash ROM 128KB 256KB 384KB

Table 1.1 PMC specifications (11)

Model		Series 15–MODEL B	
		PMC–NB (4048 Series)	PMC–NB2
Programmingmethod language		Ladder C–language	Ladder C–language Step sequence
Number of ladder level		3	3
Level–1 Cycle Time		8 ms	8 ms
Basic instruction Execution Time		0.1 (µs/step)	0.1 (µs/step)
Program capacity • Ladder(step) • Symbol/Comment • Message • Language only	 (Note)	Approx. 8,000 Approx.16,000 Approx.24,000 1 to 128KB 0.1 to 64KB max. 896KB	Approx. 8,000 Approx.16,000 Approx.24,000 1 to 128KB 0.1 to 64KB max. 896KB
Instruction	(Basic) (Function)	14 kinds 69 kinds	14 kinds 69 kinds
Internal relay Message request Non–volatile • Var.Timer • Counter • Keep relay • Data table Subprogram Label Fixed timer	(R) (A) (T) (C) (K) (D) (P) (L)	1618 bytes 25 bytes 80 bytes 80 bytes 20 bytes 3000 bytes 512 programs 9999 labels Max 100 timers specified by timer No.	3200 bytes 125 bytes 300 bytes 200 bytes 50 bytes 8000 bytes 2000 programs 9999 labels Max 100 timers specified by timer No.
Input/output • I/O link • I/O card 	 (I) (O) (I) (O)	 max 1024 points. max 1024 points. – –	 max 1024 points. max 1024 points. – –
Sequence program storage media		Flash ROM 64 KB 128 KB 256 KB 512 KB 1 MB	Flash ROM 64 KB 128 KB 256 KB 512 KB 1 MB

NOTE

Please refer to (4) for PMC–NB(4047 Series).
The above–mentioned table is a value for PMC–NB/NB2 (4048 Series).

Table 1.1 PMC specifications (12)

Model		FANUC Series 15*i*
		PMC–NB6
Programmingmethod		Ladder step sequence (optional)
Number of ladder levels		3
First–level execution period		8 ms
Basic instruction processing time		0.085 μ sec/step
Program capacity		
• Ladder (step)(NOTE 1)		About 32,000 maximum
• Symbol/comment		1 to 128KB
• Message		0.1 to 64KB
Instruction	(Basic instruction)	14
	(Functional instruction)	69
Internal relay	(R)	3200 byte
Message request	(A)	125 bytes (1000 points)
Nonvolatile memory		
• Variable timer	(T)	300 bytes (150 points)
• Counter	(C)	200 bytes (50 points)
• Keep relay	(K)	50 byte
• Data table	(D)	8000 byte
Subprogram	(P)	2000 each
Label	(L)	9999 each
Fixed timer		100 each (timer number specification)
I/O		
• I/O link	(Input)	1024 points maximum
(NOTE 2)	(Output)	1024 points maximum
Sequence program storage media		Flash ROM 128KB 256KB 384KB

NOTE

1 This capacity applies when the largest available storage medium is used.

2 The one and only I/O of the FANUC Series 15*i* is the I/O Link.

1.2 SUMMARY OF SPECIFICATION OF LADDER PROGRAM

Table 1.2 Summary of specification of ladder program (1)

Model			PMC–PA1	PMC–PA3	PMC–P
PMC address	Interfaces between the PMC and CNC (F and G)		Compatible		Incompatible (Note 2)
	Interfaces between the PMC and machine (X and Y)		Compatible		Incompatible (Note 2)
	Others (R, A, C, K, D, T)		Compatible		Incompatible
Ladder program compatibility	ROM format (object)		Incompatible (Note 1)		
	Source format (mnemonic)		Compatible		Incompatible (Note 2)
System	Divided system		Not provided (Note 3)		Provided
	Undivided system		Provided		Not provided
Basic commands			Compatible		
Function commands	DISP (SUB49)		Not provided (Note 4)		Provided
	COM (SUB9)	Coil count specification	Not provided (Note 5)		Provided
		COME (SUB29) specification	Provided		
	JMP (SUB10)	Coil count specification	Not provided (Note 5)		Provided
		JMPE (SUB30) specification	Provided		

NOTE

1 The same ROM cannot be shared by different models. The ROM must be rewritten using the offline programmer.

2 It is possible that convert the signal address by the operation of “SIGNAL ADDRESS CONVERSION” (APPENDIX G).

3 The setting item of system parameter IGNORE DEVIDE CODE is not provided.

4 Use the DISPB (SUB41) command instead.

5 The range of the COM (SUB9) and JMP (SUB10) commands cannot be specified with the number of coils. Specify the range with the COME (SUB29) and JMPE (SUB30) commands. If specify the number of coils, no error messages will be displayed while editing, but “ALARM093” will be displayed when send the data to RAM.

Table 1.2 Summary of specification of ladder program (2)

Model			PMC–SA1	PMC–SA2	PMC–SA3/SA5	PMC–SB	PMC–SB2	PMC–SB3/SB4/SB5/SB6	PMC–SC	PMC–SC3/SC4
PMC address	Interfaces between the PMC and CNC (F and G)		Compatible							
	Interfaces between the PMC and machine (X and Y)		Compatible							
	Subprogram, label (P and L)		Not provided		Provided	Not provided		Provided	Not provided	Provided
	Others (R, A, C, K, D, T)		Compatible (Note 1)							
Ladder program compatibility	ROM format (object)		Incompatible (Note 2)							
	Source format (mnemonic)		Compatible (Note 3)							
System	Divided system		Not provided (Note 4)			Provided	Not provided (Note 4)		Provided	Not provided
	Undivided system		Provided							
Structuring	Sub program		Unusable		Usable	Unusable		Usable	Un-usable	Usable
Basic commands			Compatible							
Function commands	END3 (SUB48)		Not provided						Provided	
	DISP (SUB49)		Not provided (Note 5)			Provided				
	COM (SUB9)	Coil count specification	Not provided (Note 6)			Provided	Not provided (Note 6)		Provided	Not provided (Note 6)
		COME (SUB29) specification	Provided							
	JMP (SUB10)	Coil count specification	Not provided (Note 6)			Provided	Not provided (Note 6)		Provided	Not provided (Note 6)
		JMPE (SUB30) specification	Provided							
	FNC9X (SUB9X)		Not provided						Provided	
	MMCWR (SUB98), MMCWW (SUB99) MMC3R (SUB88), MMC3W (SUB89) (Note 7)		Provided (Note 7)			Provided				
	MOVB (SUB43), MOVW (SUB44) , MOVN (SUB45)		Not provided		Provided	Not provided		Provided	Not provided	Provided
	DIFU (SUB57), DIFD (SUB58)		Not provided		Provided	Not provided		Provided	Not provided	Provided
	AND (SUB60), OR (SUB61) NOT (SUB62), EOR (SUB59)		Not provided		Provided	Not provided		Provided	Not provided	Provided
Function command (for structured programming)	Commands for subprogram END (SUB64) , CALL (SUB65), CALLU (SUB66) , SP (SUB71), SPE (SUB72)		Not provided		Provided	Not provided		Provided	Not provided	Provided
	Extended jump command JMPB (SUB68), JMPC (SUB73) LBL (SUB69)		Not provided		Provided	Not provided		Provided	Not provided	Provided

NOTE

1 The internal relay and the data table in nonvolatile memory for the PMC-SB3, SC, SC3 are extended, compared with those for other models.

2 The same ROM cannot be shared by different models. The ROM must be rewritten using the offline programmer. However, the ROM for the PMC–SA2 can be used for the PMC–SA3 and the ROM for the PMC–SB2 can be used for the PMC–SB3.

3 The program can be converted by reinputting it after it is output in a source format.

4 The setting item of system parameter IGNORE DEVIDE CODE is not provided.

5 Use the DISPB (SUB41) command instead.

6 The range of the COM (SUB9) and JMP (SUB10) commands cannot be specified with the number of coils. Specify the range with the COME (SUB29) and JMPE (SUB30) commands.

7 For the FS18A (PMC–SA1/SA2/SA3), only the MMC–III can be used. For the FS18B, the MMC–III and MMC–IV can be used.
For the FS21B (PMC–SA1/SA3), the MMC–IV can be used.
For the FS16*i*/18*i*/21*i*, the MMC–IV can be used.
For the FS16C/18C, the MMC–IV can be used.

Table 1.2 Summary of specification of ladder program (3)

Model / Series			PMC–NA (4046)	PMC–NB (4047) (4048)	PMC–NB2 (4048)
PMC address	Interfaces between the PMC and CNC (F and G)		Incompatible		
	Interfaces between the PMC and machine (X and Y)		Compatible		
	Subprogram, label (P and L)		Not provided	Provided	
	Others (R, A, C, K, D, T)		Compatible (Note 1)		
Ladder program compatibility	ROM format (object)		Incompatible (Note 2)		
	Source format (mnemonic)		Compatible (Note 3)		
System	Divided system		Provided	Not provided	
	Undivided system		Not provided	Provided	
Structuring	Subprogram		Usable	Unusable	
	Step sequence		Unusable		Usable
Basic commands			Compatible		
Function commands	END3 (SUB48)		Provided		
	DISP (SUB49)		Provided	Not provided	
	COM (SUB9)	Coil count spesification	Provided	Not provided	
		COME (SUB29) specification	Provided		
	JMP (SUB10)	Coil count specification	Provided	Not provided	
		JMPE (SUB30) specification	Provided		
	FNC9X (SUB9X)		Provided		
	LIBRY (SUB60), LEND (SUB61)		Provided	Not provided	
	MMCWR (SUB98), MMCWW (SUB99) MMC3R (SUB88), MMC3W (SUB89) MOVB (SUB43), MOVW (SUB44) MOVN (SUB45) DIFU (SUB57), DIFD (SUB58) AND (SUB60), OR (SUB61) NOT (SUB62), EOR (SUB59)		Not provided	Provided	
Function command (for structured programming)	• Command for subprogram END (SUB64) , CALL (SUB65) , CALLU (SUB66) , SP (SUB71) , SPE (SUB72)		Not provided	Provided	
	• Extended jump command JMPB (SUB68) , JMPC (SUB73) , LBL (SUB69)		Not provided	Provided	

NOTE

1 Management of internal relay address and that of datatable are different between the PMC–NB/NB2 and the PMC–NA.

2 The same ROM cannot be shared by different models.
The ROM must be rewritten using the offline programmer.

3 The data can be converted by outputting in the source format and then inputting again.
Moreover, a part of functional instruction is not compatible between PMC–NB/NB2 and PMC–NA.

Table 1.2 Ladder Compatibility (4)

Model		FANUC Series 15–B		FANUC Series 15*i*
		PMC–NB	PMC–NB2	PMC–NB6
PMC address	PMC–CNC interface (F, G)	Compatible(NOTE 1)		
	PMC–machine interface (X, Y)	Compatible		
	Area used by management software (K)	K17 to K19	K900 to K909	
Ladder compatibility	ROM format (object)	Not compatible	Compatible(NOTE 2)	
	Source format (mnemonic)	Compatible		
Basic instruction		Compatible		
Functional instruction	FNC9X(SUB9X) MMC3R(SUB88) MMC3W(SUB89) MMCWR(SUB98) MMCWW(SUB99)	Yes		No
User program (C)		Yes		Yes(NOTE 3)
User program (step sequence)		No	Yes	Yes (optional)
Sequence program automatic operation		K17#2=1	K900#2=1	K900#2=0

NOTE

1 Compatibility is not maintained for the interface unique to the Series 15*i*.

2 The PMC–NB6 of the Series 15*i* is highly compatible with the PMC–NB2 of the Series 15–B. The PMC–NB2 and PMC–NB6 differ from each other in:

(1) Execution time–dependent ladder

As instruction execution becomes faster, the following changes may occur in the execution timing:

- Change in the execution cycle of the second ladder level
- Change in timing for the second–level split and first–level execution
- Change in timing for ladder execution and I/O transfer

(2) Window functions

The functional instructions that can be used vary between the PMC–NB2 and PMC–NB6. See Chapter 5, ”PMC Functional Instructions” in Part I, ”PMC Sequence Program.”

(3) Screen manipulation

The operating procedure for the PMC screen for the PMC–NB6 of the 15i varies slightly from that for the PMC–NB2 of the 15–B.

See Chapter 7, ”PMC–NB6 Screen Manipulation,” in Part II, ”PMC Manipulation.”

(4) Step sequence

For the PMC–NB6, the step sequence is optional.

3 The C option is necessary.

1.3 WHAT IS A SEQUENCE PROGRAM?

This is paragraph outlines functions of a sequence program before explaining the programming work.

A sequence program is a program for sequence control of machine tools and other systems.

A program is defined as a processing procedure to enable CPU to execute arithmetic processing.

This program is converted into a format (machine language instructions) to enable CPU to execute decoding and arithmetic processing, and stored into the RAM or ROM memory.

The CPU reads out instructions of the program stored into the memory at high speed every instruction, and executes the program by arithmetic operation.

The programming of a sequence program begins with the production of a ladder diagram which serves as a processing procedure for arithmetic processing by CPU.

This ladder program is produced using PMC instructions.

After producing the ladder diagram, the processing sequence of this ladder diagram is converted into machine language instructions, and stored into the memory (program input).

Conversion into the machine language instructions and storage into the memory are done by the PMC programmer. The PMC programmer is a function to produce a program.

The sequence program being stored into the memory is sequentially read out into the PMC's CPU every instruction at high speed and executed.

Fig. 1.3 shows this relation.

The CPU reads out input circuit signals of address X0.0 by RD X0.0 instruction, and sets them into an operation register. Then, the CPU executes AND operation with internal relay states at address R10.0 according to the AND R10.1 instruction, and sets these results into the operation register.

The CPU executes instructions at high speed and outputs arithmetic results to the address Y0.0 output circuit.

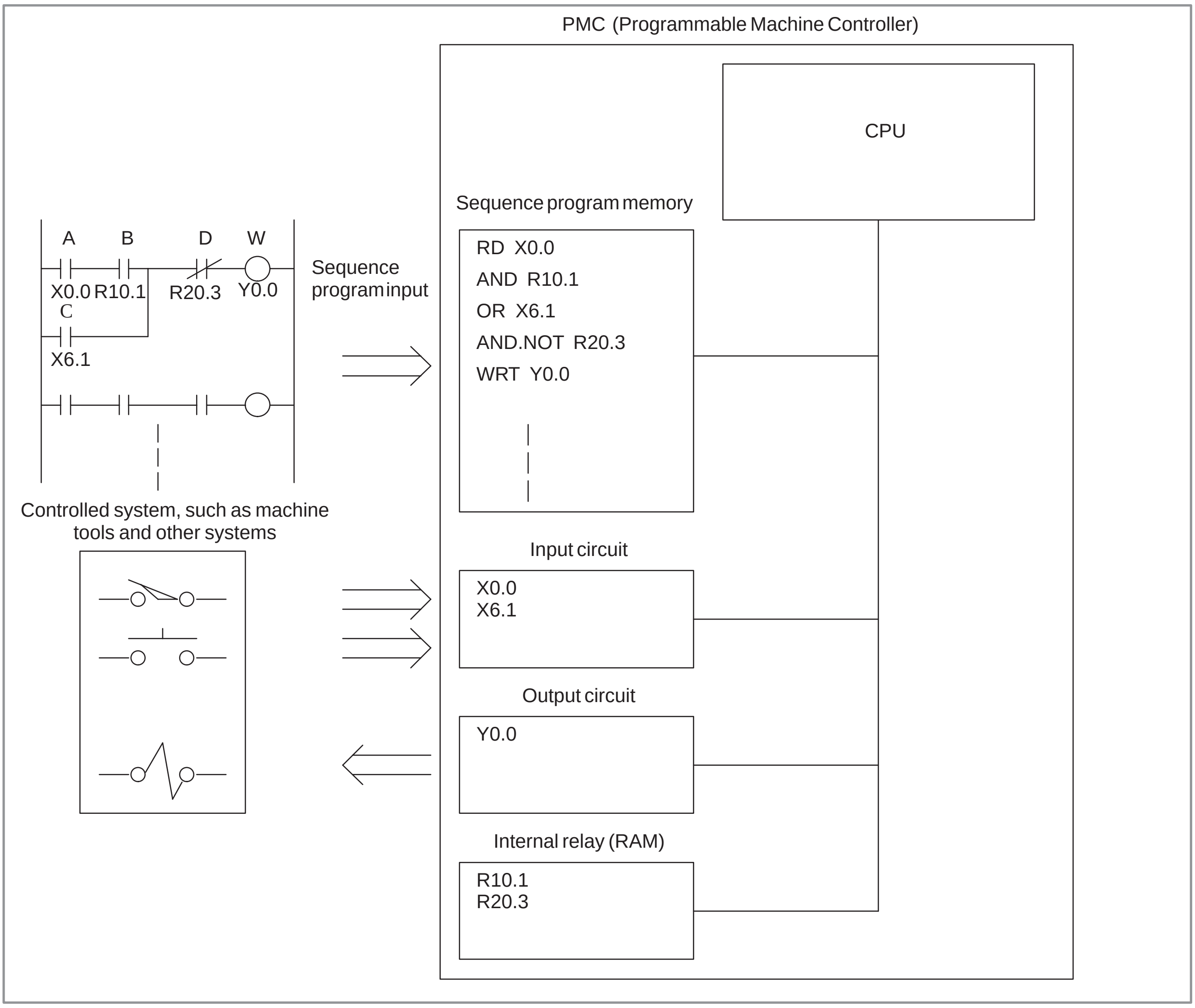


Fig. 1.3 Execution of sequence program by PMC

1.4 CREATION OF INTERFACE SPECIFICATIONS (STEPS 1 TO 3)

After deciding the control object specifications and calculating the number of input/output signal points, create the interface specifications. Use the input/output signal interface tables in the CONNECTING MANUAL for the creation of the interface specifications. Enter the signal names (within six characters) in the input/output signal interface table according to the type of the connected signals. For the input/output signals, see CONNECTING MANUAL.

1.5 CREATION OF LADDER DIAGRAM (STEP 4)

Express the control operations decided by step 2 by use of the ladder diagram (relay circuit diagram). For the functions of the timer, counter, etc. which cannot be expressed with the relay symbols (i.e. the functional instructions), express them with the symbols assigned to the functional instructions.

In the offline programmer and built-in editing function, the sequence program can be entered in the ladder diagram format from the keys of the CRT/MDI panel or from the keys of the keyboard of the SYSTEM P series.

Also, the entered sequence program can be output to the printer in the ladder diagram format using the SYSTEM P series.

Therefore, entry can be performed while the ladder diagram is created on the CRT screen at the time of sequence program entry. Thus no ladder diagram may be prepared in advance.

However, in order to shorten the time occupied by the equipment for the creation of the sequence program or to efficiently create the sequence program, it is recommended to prepare the ladder diagram in advance.

The ladder diagram is used as a maintenance diagram by the personnel in charge of maintenance in FANUC, the machine tool builder and end user in the world. Therefore, the ladder diagram must be easy to understand.

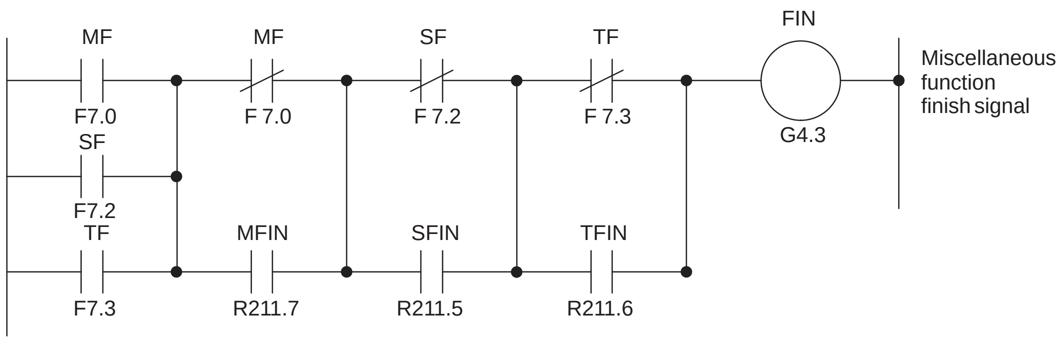
Signal names (max. six characters) can be entered to the input/output signals, comments (max. 30 characters) can be entered to the relay coil, and comments (max. 30 characters) can be entered to the input/output signals of the address tables at the time of entry of the sequence program. Be sure to enter understandable signal names and comments as much as possible.

1.6 CODING (STEP 5)

In the coding, the contents of control expressed in the ladder diagram are converted into PMC instructions. In the case of using the offline programmer or ladder diagram editting, since sequence program entry can be performed in the simple ladder diagram format, it is normally unnecessary to perform coding.

Coding is necessary only when the sequence program is punched on a paper tape and entered from the paper tape.

Examples of the ladder diagram and the coding are shown in Fig. 1.6.



Step number	Instruction	Address number, bit number	Remark
850	RD	F7.0	MF
851	OR	F7.2	SF
852	OR	F7.3	TF
853	RD.NOT.STK	F7.0	MF
854	OR	R211.7	MFIN
856	AND.STK		
857	RD.NOT.STK	F7.2	SF
858	OR	R211.5	SFIN
859	AND.STK		
860	RD.NOT.STK	F7.3	TF
861	OR	R211.6	TFIN
862	AND.STK		
863	WRT	G4.3	FIN

Fig. 1.6

1.7 SEQUENCE PROGRAM ENTRY (STEPS 6, 7)

The sequence program can be entered in five ways as follows:

(1) Entry with CRT/MDI keys
The sequence program is entered in the ladder diagram format by pressing the keys of the CRT/MDI.

(2) Entry with keys of SYSTEM P series keyboard
The sequence program is entered in the mnemonic symbol by pressing the keys of SYSTEM P series keyboard.

(3) Entry from PPR of SYSTEM P series
The sequence program punched on a paper tape is read out of the PPR and stored in the memory of the SYSTEM P series.

(4) Entry form floppy disk of SYSTEM P series
This method is used when a completed sequence program is slightly changed. The sequence program written in the floppy disk is stored in the memory of SYSTEM P series.

(5) Entry form ROM Writer
This method is used when a completed sequence program is slightly changed. The sequence program written in the ROM is stored from the PMC Writer or FA Writer into P-G or Debugging RAM.

1.8 SEQUENCE PROGRAM CHECK AND WRITE INTO ROM (STEPS 8 TO 11)

Check the sequence program and write it into the ROM after check is over. The sequence program can be checked in two ways.

(1) Check by simulator
Instead of the machine, connect a simulator (consisting of lamps and switches). Instead of using input signals from the machine, enter signals by turning on and off the switches according to the machine movement. Check the output signals on the basis of the activation of the lamps.

(2) Check by system operation
Perform checks by connecting the machine. Since it sometimes happens that unexpected operations may be executed depending on a sequence program, arrange for safety before starting operations.

(3) Writing into ROM
When check of the sequence program is over, write the sequence program into the ROM. The ROMs to be used are as follows. Then, the ROM into the CNC unit, and deliver it as a regular product to an end user. Writing of the sequence program into the ROM, maintenance and control thereof shall be performed by the machine tool builder. For this purpose, FANUC provides the PMC Writer or FA Writer as the ROM writer and the ROM or the ROM module that is the PC board on which a ROM chip is mounted. Be sure to use these devices for entering a sequence program in ROMs.

1.9 STORAGE AND CONTROL OF SEQUENCE PROGRAM (STEPS 12 TO 14)

(1) Storage and control of sequence program
After debugging, the sequence program should be stored and controlled by the machine tool builder. It can be stored in the following ways:

(a) Storing in ROM
The sequence program can be stored in the ROM. For control, enter the drawing number, edition number, etc. of the machine tool builder into the label provided in the ROM, and attach it to the ROM for control. The same control is necessary for the ROM for product.

(b) Storing in floppy disk
The sequence program can be stored in the floppy disk with offline programmer. Many programs can be stored in one floppy disk.

(c) Storing in paper tape
The sequence program can be stored in the form of a paper tape.

(d) Storing in FANUC floppy disk cassette
The sequence program can be stored in floppy disk cassette.

(2) Compiling and control of maintenance drawing
The sequence program can be output to the printer in the ladder diagram format using the offline programmer or built-in editing function. Be sure to attach the ladder diagram to the machine as a maintenance drawing together with the machine tool magnetic circuit diagrams, etc.

2 SEQUENCE PROGRAM

Since PMC sequence control handled by software and operates on principles different from a general relay circuit, the sequence control method must be fully understood in order to design the PMC sequence.

2.1 EXECUTION PROCEDURE OF SEQUENCE PROGRAM

In a general relay sequence circuit, each relay operates at approximately the same time. In the figure below for example, when relay A operates, the relay D and E operate at approximately the same time. (When both contacts B and C are off.) In PMC sequence control, each relay of the circuit operates sequentially. When relay A operates, relay D operates, then relay E (see Fig. 2.1 (a)). Thus each relay operates in sequence which can be written as a ladder diagram. (programmed sequence)

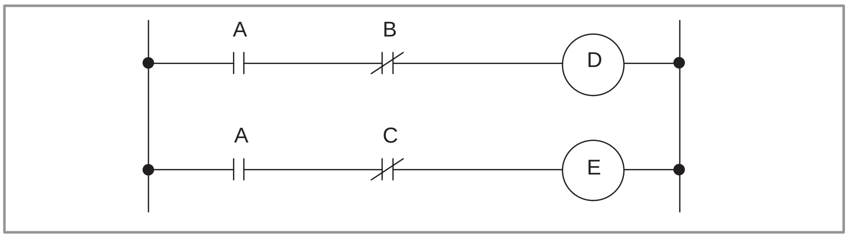


Fig. 2.1 (a) Circuit examples

Although the PMC sequential operation is performed at high speed, the speed will change with the order to be executed.

Fig. 2.1 (b) (A) and (B) illustrate operations varying from the relay circuit to PMC program.

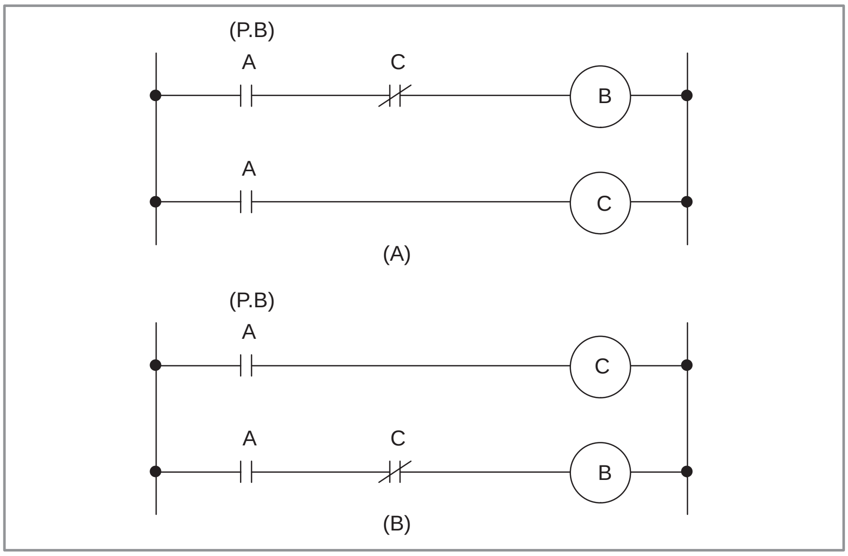


Fig. 2.1 (b) Circuit examples

(1) Relay circuit
Operations are the same in both Fig. 2.1 (b) (A) and (B). Turning on A (P.B) causes current to flow to coils B and C, which turns on B and C. When C turns on, B turns off.

(2) PMC program
In Fig. 2.1 (b) (A), as in the relay circuit, turning on A (P.B) turns on B and C, and after one cycle of the PMC sequence, turns off B. But in Fig. 2.1 (b) (B), turning on A (P.B) turns on C, but does not turn on B.

2.2 REPETITIVE OPERATION

The sequence program is executed from the beginning of coding to the end of coding of the ladder diagram in the sequence written. When the sequence program ends, the program starts over from the beginning. This is called repetitive operation.

The execution time from the beginning to the end of the ladder diagram is called the sequence processing time, which varies according to the control scale (the number of steps) and the size of the 1st level sequence. The shorter the process time is, the better the signal response becomes.

2.3 PRIORITY OF EXECUTION (1ST LEVEL, 2ND LEVEL AND 3RD LEVEL)

A sequence program consists of three parts: 1st level sequence, 2nd level sequence and 3rd level sequence. The 3rd level sequence part is added to the models usable the 3rd level sequence. (see Fig. 2.3 (a)).

The 1st level sequence part operates every 8 ms (high–speed sequential operation).

If the 1st level sequence part is long, the total operating time, including the 2nd level sequence part, is extended. Therefore the 1st level sequence part must be programmed to be processed in as short time as possible.

The 2nd level sequence part operates every 8×n ms. Here n is a dividing number for the 2nd level sequence part. The 2nd level sequence part is divided automatically when the sequence program is transferred to the RAM for debugging in the CNC unit or it is written on ROM after the program is created. The time for one cycle of the sequence program is then displayed on the offline programmer screen.

The 3rd level sequence part operates during idle time of PMC.

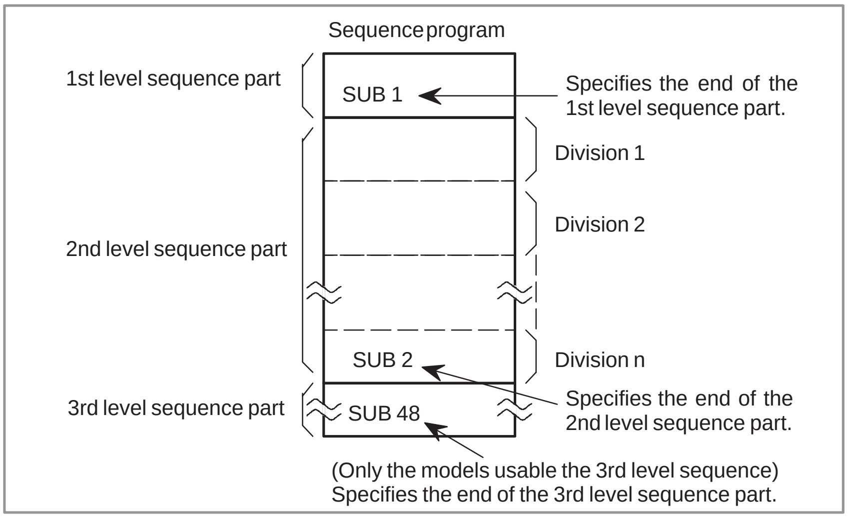


Fig. 2.3 (a) Construction of sequence program

(1) Division of the 2nd level sequence part

The 2nd level sequence part must be divided in order to execute the 1st level sequence part. For example a sequence program is executed in the following sequence when the dividing number is n. (See Fig. 2.3 (b), 2.3 (c))

After the last 2nd level sequence part (division n) is executed, the sequence program is executed again from the beginning. Thus, when the dividing number is n, the cycle of execution is 8mms (8ms×n). The 1st level sequence operates every 8 msec, and the 2nd level sequence every 8×n msec. If the steps of the 1st level sequence is increased, the steps of the 2nd level sequence operating within 8 msec becomes less, thereby increasing the dividing number and making the processing time longer. Therefore, it is desirable to program so as to reduce the 1st level sequence to a minimum.

In the, PMC–SA1, –SA2, –SB and –SB2, 1.25 ms of 8 ms is assigned to execution of the 1st and 2nd level sequences. The remaining time is assigned to NC processing.

In the PMC–SC, 5 ms of 8 ms is assigned to execution of the 1st and 2nd level sequences. The standard setting value is 5 ms when system parameter LADDER EXEC = 100%. The remaining time is assigned to execution of the 3rd level sequence and the program.

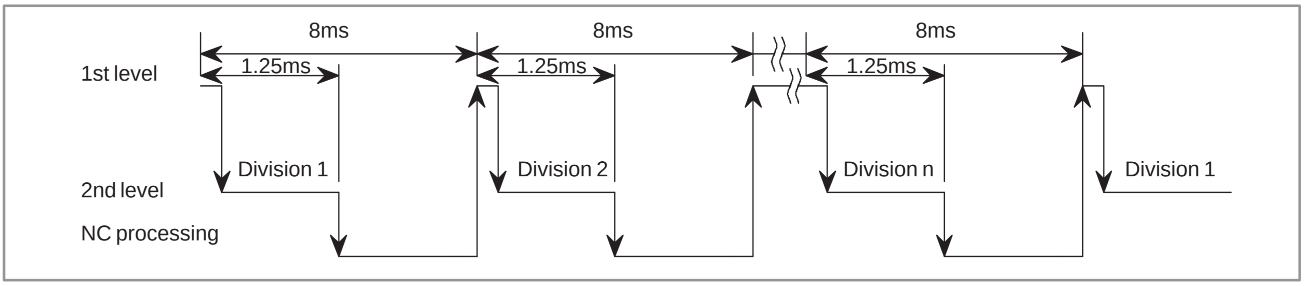


Fig. 2.3 (b) Sequence in which the Sequence Program Is Executed (PMC–SA1, –SA2, –SB and –SB2)

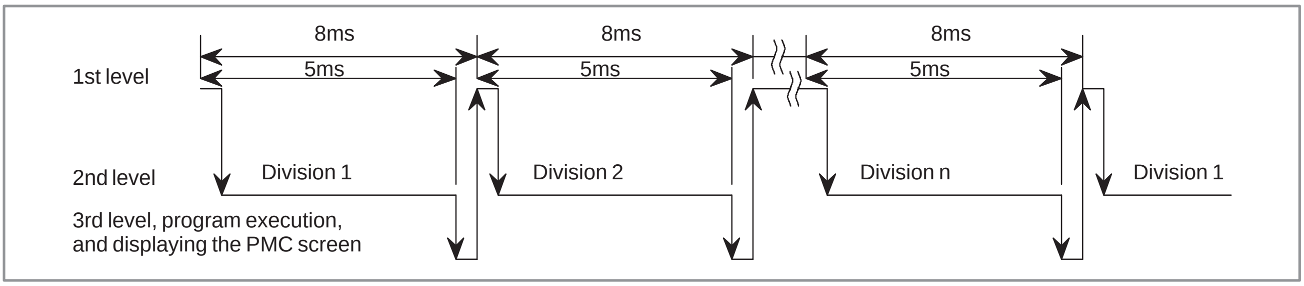


Fig. 2.3 (c) Sequence in which the Sequence Program Is Executed (PMC–SC)

(2) 1st level sequence part
Only short–width pulse signals are processed. These signals include emergency stop, overtravel of each axis, reference point return deceleration, external deceleration, skip, measuring position arrival and feed hold signals.

(3) 3rd level sequence
The purpose of the 3rd level sequence is to execute such programs as display processing or control status monitor having no direct relation to the machine control (operator message, alarm display, etc.), to lighten the load of the 2nd level program having a direct relation to the machine control by transferring former programs to the 3rd level, and to shorten the PMC execution time (cycle time).
For PMC–RC, when 3rd level program is not used, command SUB 48 (END3) following SUB 2 instruction.

(4) Divided system and undivided system
There is a model can use the divided system and undivided system among the PMCs. In the divided system, a ladder program is divided before being executed if all ladder program run regardless of the sequence state (see Fig. 2.3 (d)).
For an actual ladder program, not all ladder program run. The PMC cannot therefore be used effectively.
The PMC can execute the ladder program in the system for terminating one cycle of the program using the time to execute the actual ladder program (undivide system) as well as in the divided system.
The time required for the one cycle can be reduced by the effective use of jump instructions in the ladder program.
Since the sequence using many functional instructions requires a lot of processing time, the undivided system should be specified so that the PMC is used more effectively (see Fig. 2.3 (e)).
To operate the PMC in the undivided system, set system parameter IGNORE DIVIDE CODE to YES.
The PMC model usable only the undivided system, does not have setting system parameter IGNORE DIVIDE CODE. It is always

operated under the undivided system.

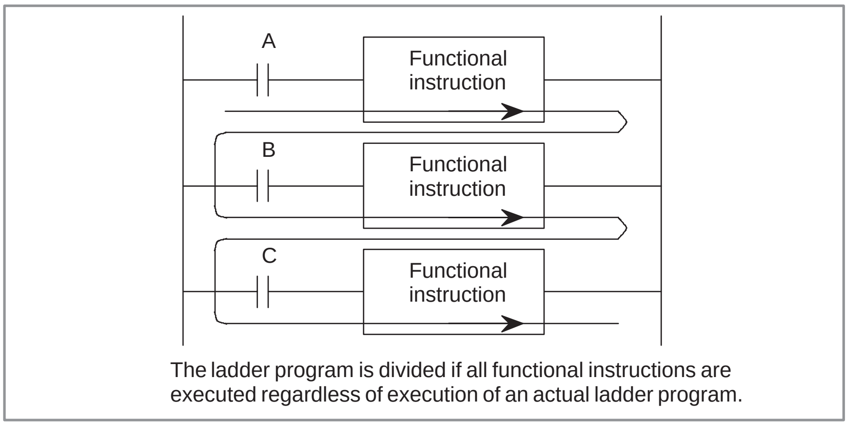


Fig. 2.3 (d) Divisions in the divided system

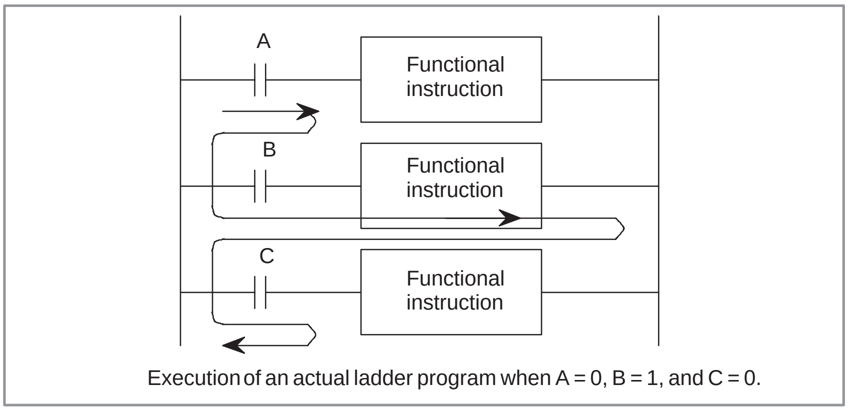


Fig. 2.3 (e) Execution of a ladder program

(a) Example of effective use of the undivided system

Example 1)

Many M codes are usually used. Since more than one M code is not used in the same block, the decoded M code is divided into several parts. Machine instructions are used as these decoded parts.

The M code is divided into M codes having two digits such as M21, M22, M24, M28, and so on.

Example 2)

To reduce the number of ROM types using the same ladder program for multiple machines, a PMC parameter must be specified so that any of the following ladder program run.

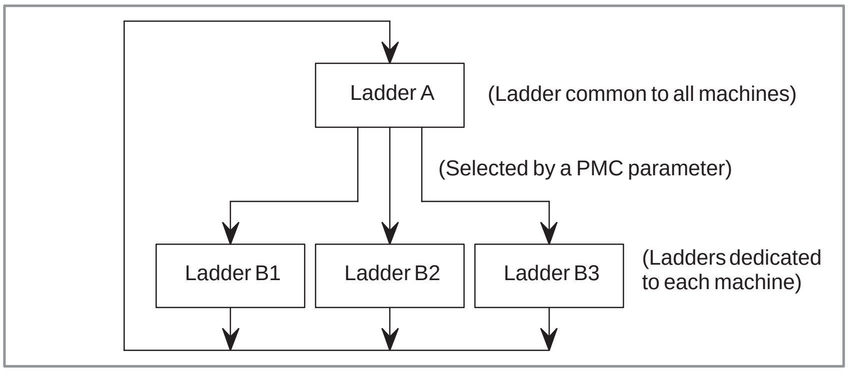


Fig. 2.3 (f)

(5) Construction of sequence program in the case of using Sub–program.

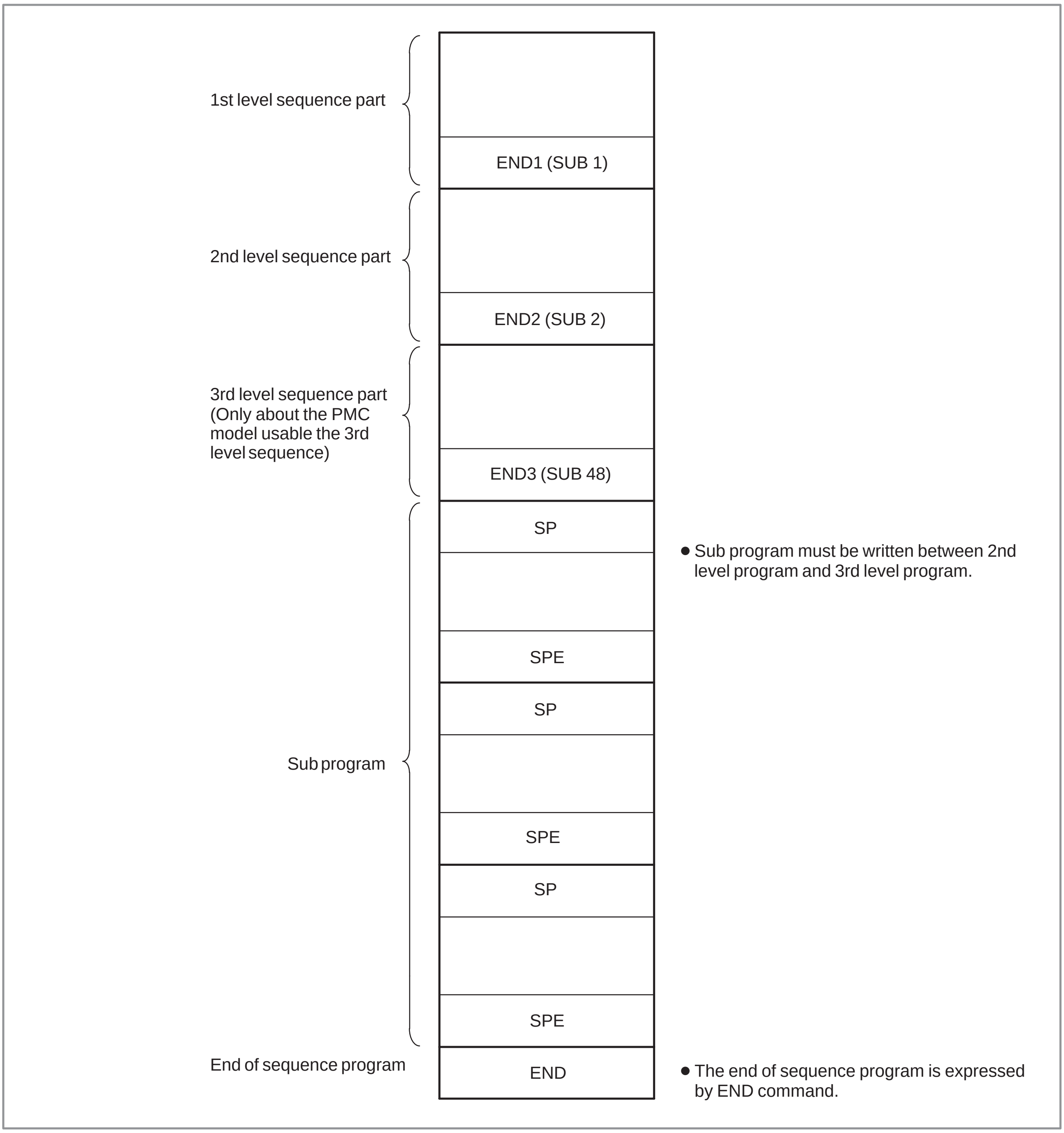


Fig. 2.3 (g)

2.4 SEQUENCE PROGRAM STRUCTURING

○ : Can be used
× : Cannot be used

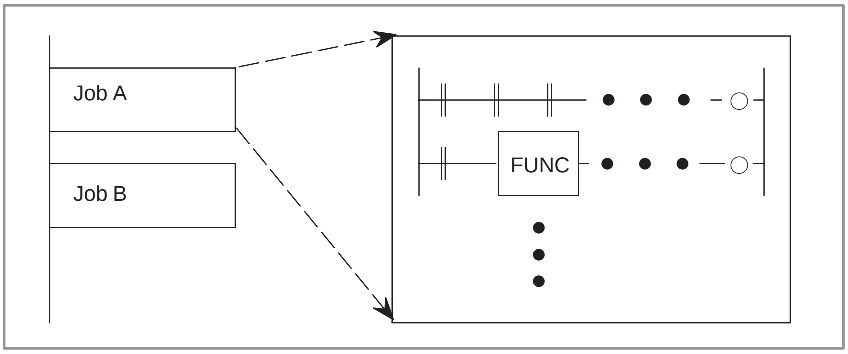
PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

With the conventional PMC, a ladder program is described sequentially. By employing a ladder language that allows structured programming, the following benefits are derived:

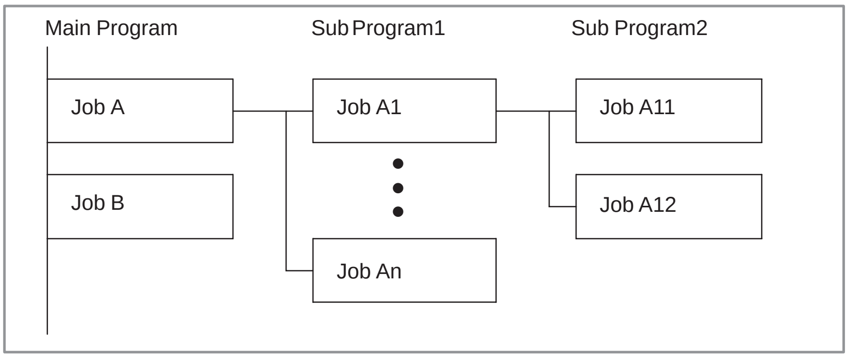
- A program can be understood and developed easily.
- A program error can be found easily.
- When an operation error occurs, the cause can be found easily.

Three major structured programming capabilities are supported.

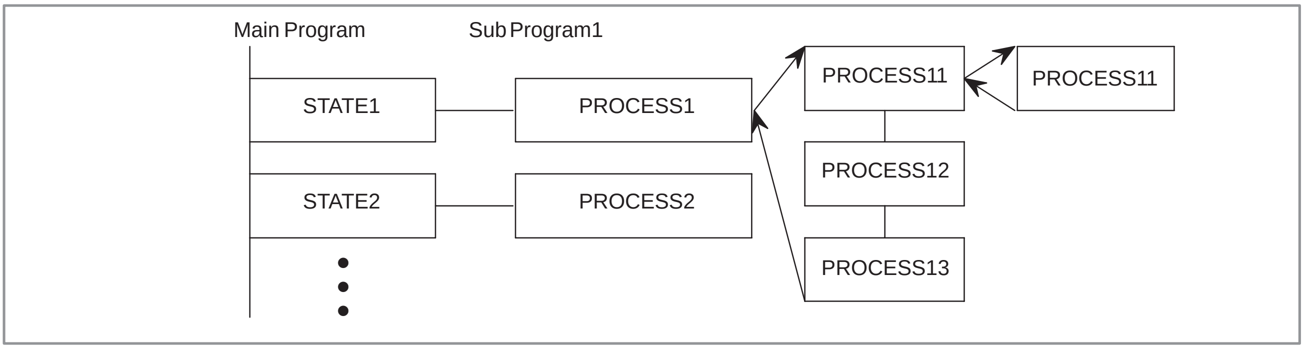
(1) Subprogramming
A subprogram can consist of a ladder sequence as the processing unit.



(2) Nesting
Ladder subprograms created in (1) above are combined to structure a ladder sequence.



(3) Conditional branch
The main program loops and checks whether conditions are satisfied. If a condition is satisfied, the corresponding subprogram is executed. If the condition is not satisfied, the subprogram is skipped.



For details, see Chapter 9.

2.5 PROCESSING I/O SIGNALS

Input signals (M function, T function, etc.) from the CNC and those (cycle start, feed hold, etc.) from the machine tool are sent to the PMC.

Signals for the CNC (cycle start, feed hold, etc.) and those for the machine tool (tunret rotation, spindle stop, etc.) are output from the PMC.
Fig. 2.4 shows the relationship between these signals and the PMC.

Input signals are entered in the input memory of PMC and output signals are issued from PMC.
As shown in Fig. 2.5, the input signals are synchronized only in the 2nd level sequence part.

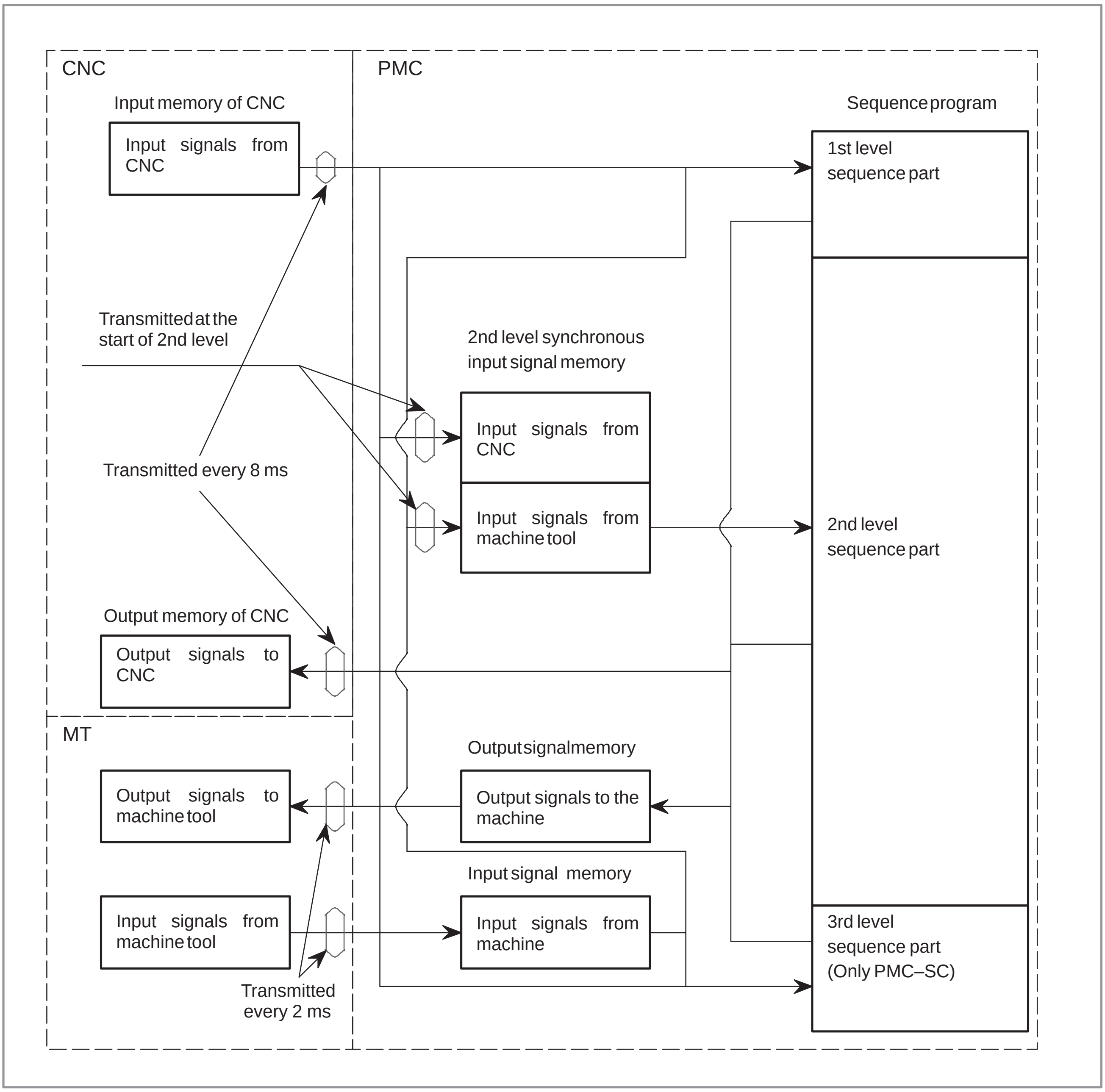


Fig. 2.5 PMC I/O signals

2.5.1 Input Signal Processing

(1) Input memory of CNC

The input signals from CNC are loaded in memory of CNC and are transferred to the PMC at intervals of 8 ms.
Since the 1st level and the 3rd level sequence part directly refer to these signals and process operations, these signals do not synchronize with input signals from the CNC.
See item 2.5.3.

(2) Input signals from machine tool (DI/DO card)
Input signals from the machine tool are transferred to the input signal memory from the input circuit (DI/DO card). 1st level and 3rd level sequence part directly processes by reading signals loaded in the input signal memory.

(3) Input signal memory
The input signal memory stores signals transferred from the machine tool at intervals of 2 ms period.
The PMC 1st level sequence part and 3rd level sequence part are used to read and process signals stored in this memory.
In this case, state of signals set in the input signal memory synchronizes with that of 1st level sequence part but not with that of 3rd level sequence part.
See item 2.5.3.

(4) 2nd level synchronous input signal memory
The 2nd level synchronous input signal memory stores signals processed by the 2nd level sequence section.
State of the signals set in this memory synchronizes with that of the 2nd level sequence part.
Input signal memory and input signals from the CNC are transferred to the 2nd level synchronous input signal memory only at the beginning of execution of the 2nd level sequence section. Therefore, the status of the 2nd level synchronous input signal memory does not change from the beginning to end of the execution of the 2nd level sequence part.
Programmer function makes the processing so that the 1st level sequence section and 3rd level sequence section use the input signal memory and input signals from the CNC side and the 2nd level sequence section uses the 2nd level synchronous input signal memory.

2.5.2 Output Signal Processing

(1) CNC output memory

The output signals are transferred from the PMC to the CNC output memory at intervals of 8 ms.

(2) Output signals to machine tool (DI/DO card)
Output signals to the machine tool are transferred from the PMC output signal memory to the machine tool.

(3) Output signal memory
The output signal memory is set by the PMC sequence program. Signals stored in this memory are transferred to the machine side at a 2 ms period.

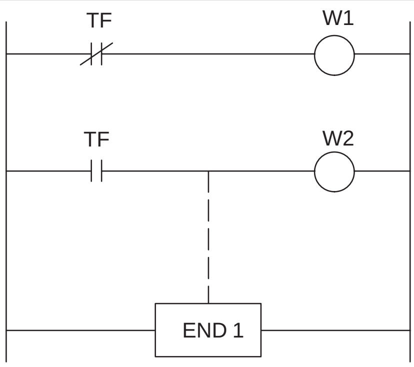
NOTE

The status of the CNC input memory, input signals from machine, CNC output memory and output signals to machine can be checked by using the PC self–diagnosis function.

The self–diagnosis number specified is the address number used by the sequence program.

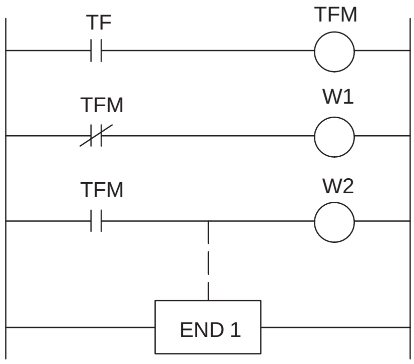
2.5.3 I/O Signals to CNC

Signals input from the CNC are transferred to the PMC at intervals of 8 ms.
Signals output to the CNC are transferred from the PMC at intervals of 8 ms.
PMC I/O signals are generally transferred at intervals of 8 ms.
In this case, note that state of the input signals from the CNC does not synchronize with that of the 1st level sequence program and the 2nd level sequence program. By this reason, if an input signal from the CNC may change while execution of the 1st level sequence program, for example, some trouble may occur like example in Fig. 2.5.3 (a).
To avoid such trouble, write the state of signal TF in an internal relay at the start of the 1st level sequence, then the 1st level sequence program shall refer to the internal relay as signal TF. See Fig. 2.5.3 (b).



If after TF=0 is load, signal state changes to TF=1, state of W1=1 and W2=1 may momentary occur

Fig. 2.5.3 (a)



Make signal TF synchronized one, and state of W1=1 and W2=1 may not occur.

Fig. 2.5.3 (b)

2.5.4 Difference of Status of Signals between 1st Level and 2nd Level

The status of the same input signal may be different in the 1st level and 2nd level sequences. That is, at 1st level, processing is performed using input signal memory and at 2nd level, processing is performed using the 2nd level synchronous input signal memory. Therefore, it is possible for a 2nd level input signal to delay by a cycle of 2nd level sequence execution at the worst, compared with a 1st level input signal.
This must be kept in mind when writing the sequence program.

Signal statesO
- A.M ON (short time width pulse signal)
- B OFF
- C OF

Differences drawn in Fig. 2.5.4 (a) and Fig. 2.5.4 (b) when the 1st level sequence has been executed are as follows:

(a) Fig. 2.5.4 (a)
W2 may not be 1 even when W1=1. (Because the A.M signal may be different at the 1st and 2nd levels.)

(b) Fig. 2.5.4 (b)
If W1=1, W2=1.
When performing the sequence shown in Fig. 2.5.4 (a), proceed as follows:
At 1st level, perform a high–speed sequence when the A.M signal changes (operating).
At 2nd level, perform sequence processing when the A.M signal does not change (stopped).

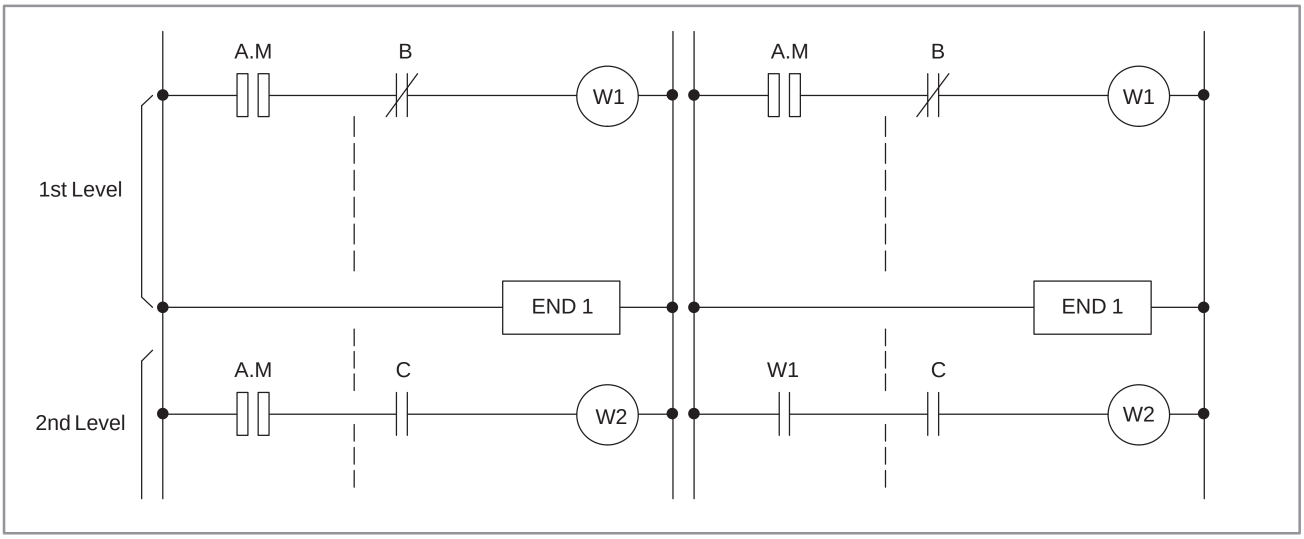


Fig. 2.5.4 (a) Fig. 2.5.4 (b)

2.6 INTERLOCKING

Interlocking is externally important in sequence control safety.

Interlocking with the sequence program is necessary. However, interlocking with the end of the electric circuit in the machine tool magnetics cabinet must not be forgotten. Even though logically interlocked with the sequence program (software), the interlock will not work when trouble occurs in the hardware used to execute the sequence program. Therefore, always provide an interlock inside the machine tool magnetics cabinet panel to ensure operator safety and to protect the machine from damage.

2.7 SEQUENCE PROGRAM PROCESSING TIME

The exact sequence processing time is displayed on the CRT screen when the sequence programs have been completed. The time is 2nd level sequence division number n x 8 ms.
This section explains how to estimate processing times that are important in sequence control when the ladder diagram, the basis of sequence program control, is almost complete.

(1) Processing time calculation units
Sequence processing time estimation is based on the basic instructions (AND, OR, etc.). The execution time for a functional instruction is given in the execution constant column of the Functional Instruction Table. Converted to a basic instruction; that is the number of basic instructions that a functional instruction is equivalent to.
Processing time is determined for the above using the equation in item below.

(2) Processing time estimation equation
The number of division (n) in the 2nd level sequence is determined and the processing time is calculated using the following equations:

Sequence processing time =
n (number of division) $\times$ 8 msec

$$n = \frac{(LT)\mu sec}{(ET)\mu sec - (HT)\mu sec} + 1$$

(n is an integer, fractions are omitted)

(a) (HT) is the execution time for the 1st level sequence section.
(HT)={(number of steps in basic instruction)+(sum of functional instruction execution time constants) $\times$10} $\times$(IT) µsec
Execution time constant for END.1 (206) must be included in HT.

(b) (LT) is the execution time for the 2nd level sequence section.
(LT)={(number of steps in basic instruction)+(sum of functional instruction execution time constants) $\times$10} $\times$(IT) µsec
END.2 execution time (127) must be included.

(c) (ET) is the execution time assigned to the 1st and 2nd level parts out of 8 ms.
For PMC–SB
(ET) = 1.25 ms = 1250µs
For PMC–SC (standard setting when LADDER EXEC = 100%)
(ET) = 5 ms = 5000µs

(d) IT) is the execution constant for calculating the processing time.
The value is as follows:
(IT) = 0.15µs

(3) Processing time calculation example

(a) 1st level sequence

Basic instruction: 100 steps
Functional instruction:

CTR:	2 times,
COMPB:	2 times
CTR execution time constant:	26
COMPB execution time constant:	24
END.1 execution time constant:	206

HT={100+(26 × 2+24 × 2+206) × 10} × 0.15 =474 μsec

(b) 2nd level sequence

Basic instruction:	6,000 steps
Functional instruction:	
TMR:	35 times,
DECB:	25 times,
ROTB:	2 times
TMR execution time constant:	23
DECB execution time constant:	20
ROTB execution time constant:	33
END.2 execution time constant:	32

LT={6,000+(23 35+20 25+33 2+32) 10} 0.15=3004.5msec

(c) Determination of the number of divisions (n)

$$n= \frac{3004.5\ \mu sec}{1250\mu sec - 474\ \mu sec} +1 = 4.87$$

(d) Processing time calculation

Sequence processing time=4 (number of division) × 8 msec=32 msec

2.8 SEQUENCE PROGRAM MEMORY CAPACITY

In the PMC–SB, one 1M–bit EPROM is used for storing the sequence program. In the PMC–SA1, –SA2, –SB and –SB2, a 128KB, 256KB, 512KB, or 1MB ROM module is used for this purpose.
Table 2.8 (a) shows the maximum memory capacity available for the sequence program. The number of bytes in parentheses indicates the size of the area dedicated to the programs other than the sequence program.

Table 2.8 (a) Maximum Memory Capacity for a Sequence Program

PMC	ROM	Ladder	Symbol and comment	Message	Total	
PMC–SA1, SA2, SB, SB2	1M–bit EPROM	64KB	64KB each	64KB	126KB	Note)
PMC–SC	128KB ROM module	96KB	64KB each	64KB	126KB	Note)
PMC–SC3	256KB ROM module	96KB	64KB each	64KB	254KB	Note)
PMC–NB	512KB ROM module	96KB	64KB each	64KB	288KB (222KB)	
	1MB ROM module	96KB	64KB each	64KB	288KB (734KB)	

NOTE
All ladder, symbol, comment, and message data items cannot be created using each maximum memory capacity. Reduce the memory capacity for any of the data items and create them so that they add up to the total capacity of each ROM.

Generate a sequence program within a range of bytes shown in Table 2.8 (a). Calculate the number of bytes of sequence program instructions and data based on Table 2.8 (b).
When the program is initialized, symbol and comment areas are allocated 32KB (extendable and reducible in 1KB units) on memory. When the program is initialized, a message area is also allocated 2.1KB (extendable and reducible in 1KB units). Therefore, program the basic instructions and functional instructions listed in Table 2.8 (b) in the remaining capacity which is the difference of the number of bytes shown in Table 2.8 (a) and the number of bytes for symbols, comments, and messages.

Table 2.8 (b) Sizes of sequence program instructions and data

Basic instruction	Functional instruction	Functional instruction parameters	Message data	Symbol	Comment
4 bytes	4 bytes	4 bytes	1 byte/character (alphanumeric characters) 2 bytes/kana characters	10 bytes	1 byte/character

3 ADDRESS

An address shows a signal location. Addresses include input/output signals with respect to the machine, the input/output signals with respect to the CNC, the internal relays, the counters, the keep relays (PMC parameters), and data table. Each address consists of an address number (for every 8 signals) and a bit number (0 to 7). Enter the symbol table showing the relationship between the signal names and the addresses into the programmer by using the keys of the CRT/MDI or the keys of the keyboard of the offline programmer as in the case of the sequence program.
For programming, see Chapter III, IV and V.

(1) Addresses related to PMC
Four types of addresses as shown in Fig. 3 are necessary for creation of the PMC sequence program.

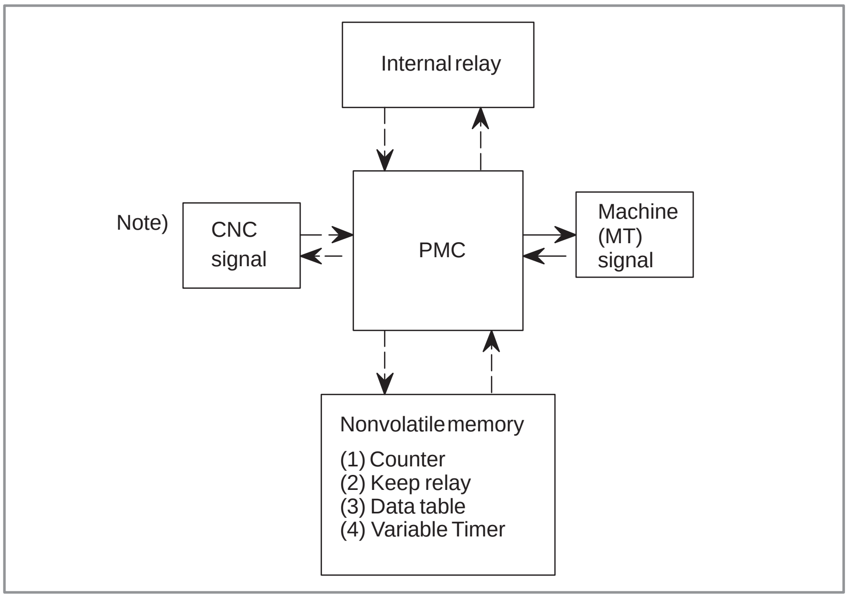


Fig. 3 Addresses related to PMC

(a) The input/output signals with respect to the PMC, which are indicated by the solid lines, are transferred via the receiver and the driver of the I/O board.

(b) The input/output signals with respect to the PMC, which are indicated by the broken lines, are transferred only in the memory such as the RAM.
All of these signals can be displayed on the CRT/MDI panel.

(2) Address regulations
The address comprises the address number and the bit number in the format as shown below.

```
X 127.   7
   |     |
   |     └──> Bit number (0 to 7)
   └────────> Address number (within four numerics after alphabet)
```

An alphabet must be specified at the beginning of the address number to indicate the type of the signal as shown in Table 3. When specifying the address in the byte unit by the functional instruction, specify X127. In this case, "." and the bit number are not necessary.

Table 3 Alphabetic characters in address numbers (1)

Character	Signal description	Model			
		Power Mate – D		Power Mate– F	Power Mate– H
		PMC PA1	PMC– PA3	PMC– PA3	PMC– PA3
X	Signal from the machine to the PMC (MT to PMC)	X0 to X127 (I/O Link Master) X1000 to X1003 (Built–in I/O Card) X1020 to X1051 (I/O Link Slave)		X1000 to X1005 X1020 to X1027 (Slave)	X0 to X127 (I/O Link Master) X1000 to X1003 (Built–in I/O Card) X1020 to X1051 (I/O Link Slave)
Y	Signal from the PMC to the machine (PMC to MT) (Caution 3)	Y0 to Y127 (I/O Link Master) Y1000 to Y1002 (Built–in I/ O Card) Y1020 to Y1051 (I/O Link Slave)		Y1000 to Y1003 Y1020 to Y1027 (Slave)	Y0 to Y127 (I/O Link Master) Y1000 to Y1002 (Built–in I/O Card) Y1020 to Y1051 (I/O Link Slave)
F	Signal from the NC to the PMC (NC to PMC)	F0 to F255 F1000 to F1255 (Dual path control)		F0 to F255	F0 to F255
G	Signal from the PMC to the NC (PMC to NC)	G0 to G255 G1000 to G1255 (Dual path control)		G0 to G255	G0 to G255
R	Internal relay (Caution 1)	R0 to R999 R9000 to R9099	R0 to R999 R9000 to R9117	R0 to R999 R9000 to R9117	R0 to R999 R9000 to R9117
A	Message request signal	A0 to A24		A0 to A24	A0 to A24
C	Counter	C0 to C79		C0 to C79	C0 to C79
K	Keep relay (Caution 2)	K0 to K19		K0 to K19	K0 to K19
T	Variable timer	T0 to T79		T0 to T79	T0 to T79
D	Data table	D0 to D1859		D0 to D1859	D0 to D1859
L	Label Number	–	L1 to L9999	L1 to L9999	L1 to L9999
P	Subprogram Number	–	P1 to P512	P1 to P512	P1 to P512

CAUTION

1 R9000 to R9117 are areas reserved for the PMC system program; these areas cannot be used for output by a sequence program.

2 K17 to K19 are areas reserved for the PMC system program; these areas cannot be used for output by a sequence program.

3 I/O Link Master function is not available in the Power Mate–MODEL F.
You cannot use the address X0–127 and Y0–127.

Table 3 Alphabetic characters in address numbers (2)

Character	Signal description	Model				
		FS20A		FS18A		
		PMC–SA1	PMC–SA3	PMC–SA1	PMC–SA2	PMC–SA3
X	Signal from the machine to the PMC (MT to PMC)	X0 to X127 X1000 to X1013 (Caution 1)		X0 to X127 X1000 to X1019		
Y	Ssignal from the PMC to the machine (PMC to MT)	Y0 to Y127 Y1000 to Y1013 (Caution 1)		Y0 to Y127 Y1000 to Y1014		
F	Signal from the NC to the PMC (NC to PMC)	F0 to F255 F1000 to F125		F0 to F255 F1000 to F1255		
G	Signal from the PMC to the NC (PMC to NC)	G0 to G255 G1000 to G1255		G0 to G255 G1000 to G1255		
R	Internal relay (Caution 2)	R0 to R999 R9000 to R9099	R0 to R999 R9000 to R9117	R0 to R999 R9000 to R9099	R0 to R999 R9000 to R9117	
A	Message request signal	A0 to A24		A0 to A24		
C	Counter	C0 to C79		C0 to C79		
K	Keep relay (Caution 3)	K0 to K19		K0 to K19		
D	Data table	D0 to D1859		D0 to D1859		
T	Variable timer	T0 to T79		T0 to T79		
L	Label number	–	L1 to L9999	–		L1 to L9999
P	Subprogram number	–	P1 to P512	–		P1 to P512

CAUTION

1 X1000 to X1007 and Y1000 to Y1007 are configured as a matrix.

2 R9000 to R9117 are areas reserved for the PMC system program; these areas cannot be used for output by a sequence program.

3 K17 to K19 are areas reserved for the PMC system program; these areas cannot be used for output by a sequence program.

Table 3 Alphabetic characters in address numbers (3)

Chara–cter	Signal description	Model					
		PMC–SB	PMC–SB2	PMC–SB3	PMC–SC	PMC–SC3	PMC–NB
X	Signal from the machine to the PMC (MT to PMC)	X0 to X127 X1000 to X1039					X0 to X127
Y	Signal from the PMC to the machine (PMC to MT)	Y0 to Y127 Y1000 to Y1029					Y0 to Y127
F	Signal from the NC to the PMC (NC to PMC)	F0 to F255 F1000 to F1255					F0 to F319
G	Signal from the PMC to the NC (PMC to NC)	G0 to G255 G1000 to G1255					G0 to G511
R	Internal relay (Caution 1)	R0 to R999 R9000 to R9099	R0 to R999 R9000 to R9117	R0 to R1499 R9000 to R9117	R0 to R1499 R9000 to R9099	R0 to R1499 R9000 to R9117	R0 to R1499 R9000 to R9117
A	Message request signal	A0 to A24					
C	Counter	C0 to C79					
K	Keep relay (Caution 2)	K0 to K19					
D	Data table	D0 to D1859		D0 to D2999			
T	Variable timer	T0 to T79					
L	Label number	–	–	L1 to L9999	–	L1 to L9999	
P	Subprogram number	–	–	P1 to P512	–	P1 to P512	

CAUTION

1. R9000 to R9117 are areas reserved for the PMC system program; these areas cannot be used for output by a sequence program.
2. K17 to K19 are areas reserved for the PMC system program; these areas cannot be used for output by a sequence program.

Table 3 Alphabetic characters in address numbers (4)

Character	Signal description	Model				
		Series 16–MODEL B/Series 18–MODEL B				Series 18–MODEL B
		PMC–SB3	PMC–SC3	PMC–SB4	PMC–SC4	PMC–SA1
X	Signal from the machine to the PMC (MT to PMC)	X0 to X127 X1000 to X1019 X1020 to X1039				X0 to X127 X1000 to X1019 X1020 to X1039
Y	Signal from the PMC to the machine (PMC to MT)	Y0 to Y127 Y1000 to Y1014 Y1020 to Y1034				Y0 to Y127 Y1000 to Y1014 Y1020 to Y1034
F	Signal from the NC to the PMC (NC to PMC)	F0 to F255 F1000 to F1255	F0 to F255 F1000 to F1255	F0 to F511 F1000 to F1511 F2000 to F2511	F0 to F511 F1000 to F1511 F2000 to F2511	F0 to F255 F1000 to F1255
G	Signal from the PMC to the NC (PMC to NC)	G0 to G255 G1000 to G1255	G0 to G255 G1000 to G1255	G0 to G511 G1000 to G1511 G2000 to G2511	G0 to G511 G1000 to G1511 G2000 to G2511	G0 to G255 G1000 to G1255
R	Internal relay	R0 to R1499 R9000 to R9117	R0 to R1499 R9000 to R9117	R0 to R2999 R9000 to R9199	R0 to R2999 R9000 to R9199	R0 to R999 R9000 to R9099
A	Message request signal	A0 to A24	A0 to A24	A0 to A124	A0 to A124	A0 to A24
C	Counter	C0 to C79	C0 to C79	C0 to C199	C0 to C199	C0 to C79
K	Keep relay	K0 to K19	K0 to K19	K0 to K39 K900 to K909	K0 to K39 K900 to K909	K0 to K19
T	Data table	T0 to T79	T0 to T79	T0 to T299	T0 to T299	T0 to T79
D	Variable timer	D0 to D2999	D0 to D2999	D0 to D7999	D0 to D7999	D0 to D1859
L	Label number	L1 to L9999	L1 to L9999	L1 to L9999	L1 to L9999	–
P	Subprogram number	P1 to P512	P1 to P512	P1 to P2000	P1 to P2000	–

Table 3 Alphabetic characters in address numbers (5)

Character	Signal description	Model			
		Series 16–MODEL C/Series 18–MODEL C			
		PMC–SB5	PMC–SC3	PMC–SB6	PMC–SC4
X	Signal from the machine to the PMC (MT to PMC)	X0 to X127 X1000 to X1019 X1020 to X1039			
Y	Signal from the PMC to the machine (PMC to MT)	Y0 to Y127 Y1000 to Y1014 Y1020 to Y1034			
F	Signal from the NC to the PMC (NC to PMC)	F0 to F255 F1000 to F1255	F0 to F255 F1000 to F1255	F0 to F511 F1000 to F1511 F2000 to F2511	F0 to F511 F1000 to F1511 F2000 to F2511
G	Signal from the PMC to the NC (PMC to NC)	G0 to G255 G1000 to G1255	G0 to G255 G1000 to G1255	G0 to G511 G1000 to G1511 G2000 to G2511	G0 to G511 G1000 to G1511 G2000 to G2511
R	Internal relay	R0 to R1499 R9000 to R9117	R0 to R1499 R9000 to R9117	R0 to R2999 R9000 to R9199	R0 to R2999 R9000 to R9199
A	Message request signal	A0 to A24	A0 to A24	A0 to A124	A0 to A124
C	Counter	C0 to C79	C0 to C79	C0 to C199	C0 to C199
K	Keep relay	K0 to K19	K0 to K19	K0 to K39 K900 to K909	K0 to K39 K900 to K909
T	Data table	T0 to T79	T0 to T79	T0 to T299	T0 to T299
D	Variable timer	D0 to D2999	D0 to D2999	D0 to D7999	D0 to D7999
L	Label number	L1 to L9999	L1 to L9999	L1 to L9999	L1 to L9999
P	Subprogram number	P1 to P512	P1 to P512	P1 to P2000	P1 to P2000

Table 3 Alphabetic characters in address numbers (6)

Character	Signal description	Model	
		Series 21/210–MODEL B	
		PMC–SA1	PMC–SA3
X	Signal from the machine to the PMC (MT to PMC)	X0 to X127 X1000 to X1011	
Y	Signal from the PMC to the machine (PMC to MT)	Y0 to Y127 Y1000 to Y1008 (Note)	
F	Signal from the NC to the PMC (NC to PMC)	F0 to F255 F1000 to F1255	
G	Signal from the PMC to the NC (PMC to NC)	G0 to G255 G1000 to G1255	
R	Internal relay	R0 to R1999 R9000 to R9099	R0 to R1499 R9000 to R9117
A	Message request signal	A0 to A24	
C	Counter	C0 to C79	
K	Keep relay	K0 to K19	
D	Data table	–	D0 to D1859
T	Variable timer	T0 to T79	
L	Label number	–	L1 to L9999
P	Subprogram number	–	P1 to P512

NOTE

The Y addresses for the 4082 series are Y0 to Y127 and Y1000 to Y1007.

Table 3 Alphabetic characters in address numbers (7)

Symbol	Type of signal	Model FANUC Series 16*i*/18*i*/160*i*/180*i* PMC–SB5	 PMC–SB6
X	Signal from the machine to PMC (MT → PMC)	X0 to X127 (Note)	X0 to X127 (Note)
Y	Signal from the PMC to machine (PMC → MT)	Y0 to Y127 (Note)	Y0 to Y127 (Note)
F	Signal from the NC to PMC (NC → PMC)	F0 to F255 F1000 to F1255	F0 to F511 F1000 to F1511 F2000 to F2511
G	Signal from the PMC to NC (PMC → NC)	G0 to G255 G1000 to G1255	G0 to G511 G1000 to G1511 G2000 to G2511
R	Internal relay	R0 to R1499 R9000 to R9117	R0 to R2999 R9000 to R9199
A	Message request signal	A0 to A24	A0 to A124
C	Counter	C0 to C79	C0 to C199
K	Keep relay	K0 to K19	K0 to K39 K900 to K909
D	Variable timer	T0 to T79	T0 to T299
T	Data table	D0 to D2999	D0 to D7999
L	Label number	L1 to L9999	L1 to L9999
P	Subprogram number	P1 to P512	P1 to P2000

NOTE

With the 16/18–C, the addresses (X1000 and up, Y1000 and up) for the built–in I/O card are reserved. However, these areas cannot be used for I/O. Never use X1000 and up, or Y1000 and up.

Table 3 Alphabetic characters in address numbers (8)

Symbol	Type of signal	Model FANUC Series 21*i*/210*i* PMC–SA1	Model FANUC Series 21*i*/210*i* PMC–SA5
X	Signal from the machine to PMC (MT → PMC)	X0 to X127 (Note)	
Y	Signal from the PMC to machine (PMC → MT)	Y0 to Y127 (Note)	
F	Signal from the NC to PMC (NC → PMC)	F0 to F255 F1000 to F1255	
G	Signal from the PMC to NC (PMC → NC)	G0 to G255 G1000 to G1255	
R	Internal relay	R0 to R999 R9000 to R9099	R0 to R999 R9000 to R9117
A	Message request signal	A0 to A24	
C	Counter	C0 to C79	
K	Keep relay	K0 to K19	
D	Variable timer	T0 to T79	
T	Data table	D0 to D1859	
L	Label number	–	L1 to L9999
P	Subprogram number	–	P1 to P512

NOTE

With the 16/18–C, the addresses (X1000 and up, Y1000 and up) for the built–in I/O card are reserved. However, these areas cannot be used for I/O. Never use X1000 and up, or Y1000 and up.

Table 3 Alphabetic characters in address numbers (9)

Sym-bol	Signal description	Model	
		FANUC Power Mate *i*–MODEL D/H	
		PMC–SB5	PMC–SB6
X	Input signal from the machine to the PMC (MT to PMC)	X0 to X127 (I/O Link Master) X1000 to X1003 (Built–in I/O) X1020 to X1051 (I/O Link Slave)	X0 to X127 (I/O Link Master) X1000 to X1003 (Built–in I/O) X1020 to X1051 (I/O Link Slave)
Y	Output signal from the PMC to the machine (PMC to MT)	Y0 to Y127 (I/O Link Master) Y1000 to Y1003 (Built–in I/O) Y1020 to Y1051 (I/O Link Slave)	Y0 to Y127 (I/O Link Master) Y1000 to Y1003 (Built–in I/O) Y1020 to Y1051 (I/O Link Slave)
F	Input signal from the NC to the PMC (NC to PMC)	F0 to F255 F1000 to F1255	F0 to F511 F1000 to F1511 F2000 to F2511
G	Output signal from the PMC to the NC (PMC to NC)	G0 to G255 G1000 to G1255	G0 to G511 G1000 to G1511 G2000 to G2511
R	Internal relay	R0 to R1499 R9000 to R9117	R0 to R2999 R9000 to R9199
A	Message request signal	A0 to A24	A0 to A124
C	Counter	C0 to C79	C0 to C199
K	Keep relay	K0 to K19	K0 to K39 K900 to K909
T	Variable timer	T0 to T79	T0 to T299
D	Data table	D0 to D2999	D0 to D7999
L	Label number	L1 to L9999	L1 to L9999
P	Subprogram number	P1 to P512	P1 to P2000

Table 3 Alphabetic characters in address numbers (10)

Character	Signal description	Model	
		Series 15–MODEL B	
		PMC–NB (4048)	PMC–NB2
X	Input signal from the machine to the PMC (MT to PMC)	X0 to X127	
Y	Output signal from the PMC to the machine (PMC to MT)	Y0 to Y127	
F	Input signal from the NC to the PMC (NC to PMC)	F0 to F319	
G	Output signal from the PMC to the NC (PMC to NC)	G0 to G511	
R	Internal relay	R0 to R1499 R9000 to R9099	R0 to R1499 R9000 to R9117
A	Message request signal	A0 to A24	A0 to A124
C	Counter (Non–volatile memory)	C0 to C79	C0 to C199
K	Keep relay (Non–volatile memory)	K0 to K19	K0 to K39 K900 to K909
D	Data table (Non–volatile memory)	D0 to D2999	D0 to D7999
T	Variable timer (Non–volatile memory)	T0 to T79	T0 to T299
L	Label number	L1 to L9999	
P	Subprogram number	P1 to P512	P1 to P2000

CAUTION

1 R9000 to R9199 are areas reserved for the PMC system program;
these areas cannot be used for output by a sequence program.

2 K17 to K19 or K900 to K909 are areas reserved for the PMC system program;
these areas cannot be used for output by a sequence program.

3 Please refer to (3) PMC–NB(Series 4047).

Table 3 Alphabetic Characters for PMC Address Number (11)

Character	Signal description	Model
		FANUC Series 15*i*
		PMC–NB6
X	Input signal from machine to PMC (MT→PMC)	X0 to X127
Y	Output signal from PMC to machine (PMC→MT)	Y0 to Y127
F	Input signal from the NC to PMC (NC→PMC)	F0 to F511
G	Output signal from the PMC to NC (PMC→NC)	G0 to G511
R	Internal relay	R0 to R2999 R9000 to R9199
A	Message request signal	A0 to A124
C	Counter	C0 to C199
K	Keep relay	K0 to K39 K900 to K909
T	Variable timer	T0 to T299
D	Data table	D0 to D7999
L	Label number	L1 to L9999
P	Subprogram number	P1 to P2000

3.1 ADDRESSES BETWEEN PMC AND CNC (PMC↔NC)

Addresses of the interfaces are outlined below. For details, see CONNECTING MANUAL of Series 16.

(1) Basic machine interface

(a) PMC←CNC related signals
The addresses for Series 15 are from F0 to F511, for the others are from F0 to F255.
For details of the signals, see CONNECTING MANUAL of CNC.

(b) PMC→CNC related signals
The addresses for Series 15 are from G0 to G511, for the others are from G0 to G255.
For details of the signals, see CONNECTING MANUAL of CNC.

3.2 ADDRESSES BETWEEN PMC AND MACHINE TOOL (PMC↔MT)

3.2.1 Addresses Between PMC and Machine Tool

(1) When the FANUC I/O Link is used

(a) PMC ← MT
Addresses are from X0 to X127.

(b) PMC → MT
Addresses are from X0 to X127.

Up to 1024 input and 1024 output points can be assigned to any address within the above range in byte units.

(2) When the built–in I/O card is used (except Series 15)

(a) PMC ← MT
Addresses are from X1000 to X1019.

(b) PMC → MT
Addresses are from X1000 to X1014.

The addresses in the above range are always specified. They cannot therefore be changed when the I/O points are assigned to them.

(3) NC signals whose addresses are fixed and that are input from the machine tool

Be sure to assign the following signals to be input from the machine tool to the specified addresses because the NC unit refers to the following fixed addresses during processing.

NOTE
If both I/O Link and built–in I/O card are provided, the address of the I/O card is valid.
(Except Series 15)

Table 3.2.1 (a) Input signals whose addresses are fixed (Series 16/Series 18)

	Signal	Symbol	Address	
			When the I/O Link is used	When the built–in I/O card is used
T system	Signal indicating that X–axis measurement position is reached	XAE	X4.0	X1004.0
	Signal indicating that Z–axis measurement position is reached	ZAE	X4.1	X1004.1
	Function B for directly entering the measurement value of tool compensation in the positive X direction	+MIT1	X4.2	X1004.2
	Function B for directly entering the measurement value of tool compensation in the negative X direction	–MIT1	X4.3	X1004.3
	Function B for directly entering the measurement value of tool compensation in the positive Z direction	+MIT2	X4.4	X1004.4
	Function B for directly entering the measurement value of tool compensation in the negative Z direction	–MIT2	X4.5	X1004.5
M system	Signal indicating that X–axis measurement position is reached	XAE	X4.0	X1004.0
	Signal indicating that Y–axis measurement position is reached	YAE	X4.1	X1004.1
	Signal indicating that Z–axis measurement position is reached	ZAE	X4.2	X1004.2
Common	Skip signal	SKIP	X4.7	X1004.7
	Emergency stop signal	*ESP	X8.4	X1008.4
	Deceleration signal for 1st axis reference position return	*DEC1	X9.0	X1009.0
	Deceleration signal for 2nd axis reference position return	*DEC2	X9.1	X1009.1
	Deceleration signal for 3rd axis reference position return	*DEC3	X9.2	X1009.2
	Deceleration signal for 4th axis reference position return	*DEC4	X9.3	X1009.3
	Deceleration signal for 5th axis reference position return	*DEC5	X9.4	X1009.4
	Deceleration signal for 6th axis reference position return	*DEC6	X9.5	X1009.5
	Deceleration signal for 7th axis reference position return	*DEC7	X9.6	X1009.6
	Deceleration signal for 8th axis reference position return	*DEC8	X9.7	X1009.7

If the NC is a TT system, the signals for tool post 2 listed in Table 3.2.1 (b) are always assigned to the following addresses.
In addition, the system does not have the signals for tool post 1, DEC5 to DEC8 (X9.4 to X9.7).

Table 3.2.1 (b) Input signals whose addresses are fixed (TT) (Series 16/Series 18)

	Signal	Symbol	Address	
			When the I/O Link is used	When the built–in I/O card is used
TT system	Signal indicating that X–axis measurement position is reached	XAE	X13.0	X1013.0
	Signal indicating that Z–axis measurement position is reached	ZAE	X13.1	X1013.1
	Function B for directly entering the measurement value of tool compensation in the positive X direction	+MIT1	X13.2	X1013.2
	Function B for directly entering the measurement value of tool compensation in the negative X direction	–MIT1	X13.3	X1013.3
	Function B for directly entering the measurement value of tool compensation in the positive Z direction	+MIT2	X13.4	X1013.4
	Function B for directly entering the measurement value of tool compensation in the negative Z direction	–MIT2	X13.5	X1013.5
	Skip signal	SKIP	X13.7	X1013.7
	Deceleration signal for 1st axis reference position return	*DEC1	X7.0	X1007.0
	Deceleration signal for 2nd axis reference position return	*DEC2	X7.1	X1007.1
	Deceleration signal for 3rd axis reference position return	*DEC3	X7.2	X1007.2
	Deceleration signal for 4th axis reference position return	*DEC4	X7.3	X1007.3

Table 3.2.1 (c) Input signals whose addresses are fixed (Series 15)

Type of I/O unit	Emergency stop address	Skip signal address	Measurement position reached signal	
			AE1 (XAE)	AE2 (ZAE)
Connection unit	X6.4	X11.6	X8.3	X8.4
I/O unit	X6.4	X11.6	X8.3	X8.4

3.2.2 Assignment of I/O Module Addresses

The sequence program addresses of each module should be decided by the machine tool builder. These decided addresses are set to the programmer memory by using programmer.

The address information being set to the programmer is written together with a sequence program into ROM when a sequence program is written into ROM. No I/O address is changeable in the written stage of the address information into ROM. These addresses are determined by the connecting position (group number and base number) of the I/O base unit, each module position (slot number) mounted inside the I/O base unit and each module name.

Fig. 3.2.2 (a) and Fig. 3.2.2 (b) indicate the configuration of the I/O base unit.

For the specifications and details of connections of the I/O interface module, I/O module, CPU module, and other modules, see Connection Manual of each CNC.

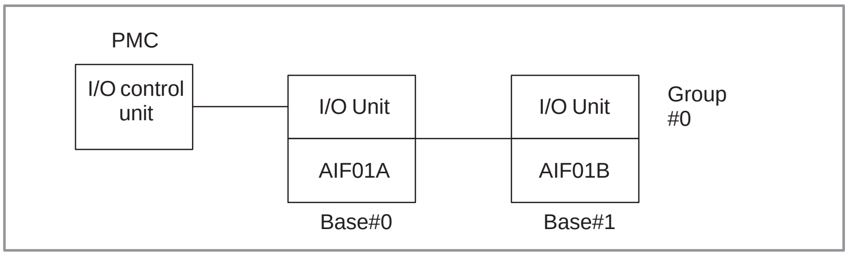


Fig. 3.2.2 (a)

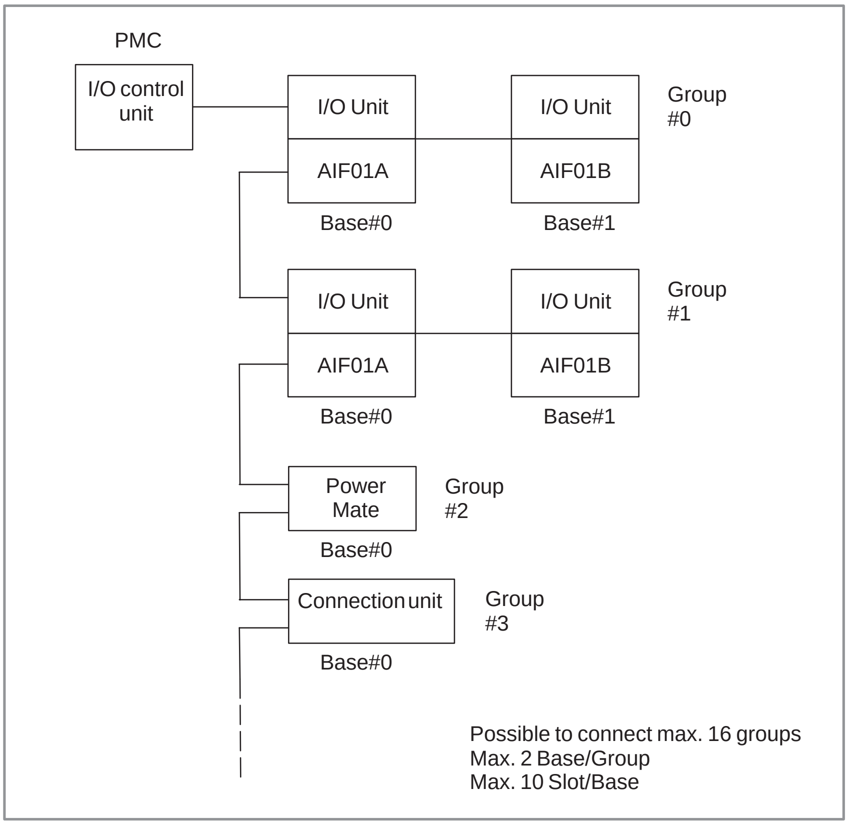


Fig. 3.2.2 (b)

(1) Group No.
Up to two I/O units can be connected using the additional I/O interface module AIF01B, based on I/O interface AIF01A. Up to two I/O units extended from AIF01A are called a group. When only one interface module is not enough to accommodate the required I/O modules, or when multiple I/O units are separately located remote from the machine, connect the first AIF01A and the second AIF01A with the cable. Up to 16 groups of I/O units can be connected.

(2) Base No.
In one group, there are 2 max. I/O base units. The I/O unit with the I/O interface module IF01A is assigned to base No. 0 and another is assigned to base No.1.

(3) Slot No.
A maximum of 5 or 10 I/O modules can be mounted on the I/O base unit ABU05A, ABU10A, respectively. The module mounting position on the I/O base unit is expressed with slot Nos. In each base unit, the mounting position of the I/O interface module is assigned to slot No. 0, and slot No. 1, 2, 3 ... are assigned in order from the left. In the case of I/O base unit (BU10B) for 10 slots, slot No. 1, 2 follow slot No. 8. The last slot No. 1, 2 are assigned for the next base address. Each module can be mounted on an arbitrary slot. It is possible to mount modules by skipping some slots.

(4) Module name
For module names, see Table 3.2.2 (a) to (c).
An actual module name begins with A. When specifying a module, omit the first letter A from the module name.

Example) When specifying module AID16D, enter ID16D.

Table 3.2.2 (a) Input modules

No.	Input format	Module name (Actual module name)	Rated voltage	Rated current	Polarity	Response time	Number of input points	Terminal	Indication by LED
1	Non–insulation DC input	ID32A (AID32A)	24VDC	7.5mA	Both	20 ms max.	32	Connector	Not provided
		ID32B (AID32B)	24VDC	7.5mA	Both	2 ms max.	32	Connector	Not provided
2	Insulation DC input	ID16C (AID32C)	24VDC	7.5mA	NEG	20 ms max.	16	Terminal board	Provided
		ID16D (AID32D)	24VDC	7.5mA	POS	20 ms max.	16	Terminal board	Provided
		ID32E (AID32E)	24VDC	7.5mA	Both	20 ms max.	32	Connector	Not provided
		ID32F (AID32F)	24VDC	7.5mA	Both	2 ms max.	32	Connector	Not provided
3	Non–insulation DC input	IA16G (AIAHG)	100 to 120VAC	14.5mA (AC120V)		ON : 20ms max OFF : 45ms max	16	Terminal board	Provided

Polarity NEG : 0 V common (current output)
POS : 24 V common (current output)

Table 3.2.2 (b) Output modules

No.	Output format	Module name (Actual module name)	Rated voltage	Rated current	Polarity	Number of points	Number of points/ common	Terminal	Indication by LED	Fuse
1	Insulation DC output	OD08C (AOD08C)	12 to 24 VDC	2A	NEG	8	8	Terminal board	Provided	Provided
		OD08D (AOD08D)		2A	POS	8	8	Terminal board	Provided	Provided
		OD16C (AOD16C)		0.5A	NEG	16	8	Terminal board	Provided	Not provided
		OD16D (AOD16D)		0.5A	POS	16	8	Terminal board	Provided	Not provided
		OD32C (AOD32C)		0.3A	NEG	32	8	Connector	Not provided	Not provided
		OD32D (AOD32D)		0.3A	POS	32	8	Connector	Not provided	Not provided
2	AC output	OA05E (AOA05E)	100 to 240 VAC	2A	–	5	1	Terminal board	Provided	Provided
		OA08E (AOA08E)		1A	–	8	4	Terminal board	Provided	Provided
		OA12E (AOR12G)	100 to 120 VAC	0.5A	–	12	6	Terminal board	Provided	Provided
3	Relay output	OR08G (AOR08G)	250 VAC/ 30 VDC max.	4A	–	8	1	Terminal board	Provided	Not provided
		OR16G (AOR16G)		2A	–	16	4	Terminal board	Provided	Not provided

Polarity NEG : 0 V common (current output)
POS : 24 V common (current output)

Table 3.2.2 (c) Other modules

No.	Name	Module name (actual module name)	Occupied address	Specifications
1	FANUC CNC SYSTEM FANUC Power Mate	FS04A	Input: 4 bytes Output: 4 bytes	FANUC Series 0–C (with FANUC I/O Link supported) FANUC Power Mate–MODEL A/B/C/D/E/F/H
		FS08A	Input: 8 bytes Output: 8 bytes	
		OC02I	Input: 16 bytes	FANUC Power Mate–MODEL D/H
		OC02O	Output: 16 bytes	
		OC03I	Input: 32 bytes	
		OC03O	Output: 32 bytes	
2	Analog input module	AD04A (AAD04A)	Input: 8 bytes	
4	Operator's panel connection unit I/O card E	OC01I	Input: 12 bytes	Ordering information: A16B–2200–0660 (sink type) A16B–2201–0730 (source type)
		OC01O	Output: 8 bytes	
5	Operator's panel connection unit I/O card D	/ 8	Input: 8 bytes	Ordering information: A16B–2200–0661 (sink type) A16B–2201–0731 (source type)
		/ 4	Output: 4 bytes	
6	Machine operator's panel interface unit	OC02I	Input: 16 bytes	
		OC02O	Output: 16 bytes	
		OC03I	Input: 32 bytes	
		OC03O	Output: 32 bytes	
7	I/O Link connection unit	/□	Input: □ bytes Output: □ bytes	Specify the same value (1 to 8) as the number of input or output bytes in □.
		OC02I	Input: 16 bytes	When one manual pulse generator is used
		OC02O	Output: 16 bytes	
		OC03I	Input: 32 bytes	When three manual pulse generators are used
		CM15I	Input: 15 bytes	When no manual pulse generator is used
8	Area for the I/O Unit MODEL B	#□	Input: □ bytes Output: □ bytes	Specify 1 to 10 bytes in □.
		##	Input: 4 bytes	Specify an area for reading information about whether the power of each unit in the I/O Unit MODEL B is on.
9	Special modules Special modules, which are not listed in Tables 3.2.2 (a) to (c)	/□	Input: □ bytes Output: □ bytes	Specify the same value (1 to 8) as the number of input or output bytes in □.
		OC02I	Input: 16 bytes	Used when the number of input or output bytes is 9 to 16.
		OC02O	Output: 16 bytes	
		OC03I	Input: 32 bytes	Used when the number of input or output bytes is 17 to 32.
		OC03O	Output: 32 bytes	

No.	Name	Module name (actual module name)	Occupied address	Specifications
10	Distribution I/O connector panel I/O modules(NOTES 3, 4)	CM03I (/3)	Input 3 bytes	Basic unit only
		CM06I (/6)	Input 6 bytes	Expansion unit 1 is used.
		CM09I	Input 9 bytes	Expansion unit 2 is used.
		CM12I (OC01I)	Input 12 bytes	Expansion unit 3 is used.
		CM13I	Input 13 bytes	The first MPG unit is used.
		CM14I	Input 14 bytes	The second MPG unit is used.
		CM15I	Input 15 bytes	The third MPG unit is used.
		CM16I (OC02I)	Input 16 bytes	DO alarm detection is used.
		CM02O (/2)	Output 2 bytes	Basic unit only
		CM04O (/4)	Output 4 bytes	Expansion unit 1 is used.
		CM06O (/6)	Output 6 bytes	Expansion unit 2 is used.
		CM08O (/8)	Output 8 bytes	Expansion unit 3 is used.
11	Distribution I/O operator’s panel I/O modules(NOTES 3, 4)	CM06I (/6)	Input 6 bytes	
		CM13I	Input 13 bytes	The first MPG unit is used.
		CM14I	Input 14 bytes	The second MPG unit is used.
		CM15I	Input 15 bytes	The third MPG unit is used.
		CM16I (OC02I)	Input 16 bytes	DO alarm detection is used.
		CM04O (/4)	Output 4 bytes	
		CM08O (/8)	Output 8 bytes	

NOTE

1 See Section 3.2.3, “I/O Link connection unit assignment method,” for how to assign the I/O Link connection unit.
2 See Section 3.2.4, “I/O Link MODEL B assignment method,” for how to assign the I/O Link MODEL B.
3 See Section 3.2.7, “Distribution panel I/O connection panel I/O module and distribution I/O operator’s panel I/O module assignment method,” for how to assign the connection panel I/O module and operator’s panel I/O module.
4 If the version of the programming system (FAPT LADDER, FAPT LADDER–II) is too old to match a module above, use the compatible module indicated in parentheses. When a compatible module having the same number of points is not available, use a compatible module having a greater number of points.

(5) How to set address to each module
The character and the mount position of each module is now decided with the group number, base number, slot number, and module name, so the address of each module can now be decided, corresponding these data and the input/output addresses. After display the I/O unit address screen as shown below on the programmer's CRT, set necessary data on the screen, Then the module address is now assigned. The occupying DI/DO points (bytes) of each module are stored in the programmer, so just assign the address of the head byte of each module, and the addresses of the other bytes in the module are automatically assigned by the programmer.
For instance, when the module ID32A is assigned address X5 as in Fig. 3.2.2 (d), the necessary 4 bytes are automatically secured. For details on operation, see Chapters III, IV, "Programmer". The input/output addresses of each module can be freely decided in this method at the machine tool builder, so the address can be decided when making the ladder diagram, as long as it does not duplicate with the addresses of each module.

	ADDRESS	GROUP	BASE	SLOT	NAME
	X000				
	X001				
	X002				
	X003				
	X004				
	X005	0	0	5	ID32A
Automatical set	X006	0	0	5	ID32A
	X007	0	0	5	ID32A
	X008	0	0	5	ID32A
	X009				

Fig. 3.2.2 (d) I/O unit address screen

NOTE
When assigning Connection unit 1, Connection unit 2 or Connection unit for operator's panel, set base number to 0 and slot number to 1.

(6) Notes when setting addresses

(a) The head bytes of the analog input module (AD04A) and analog output module (DA02A) must be assigned to even number addresses of input address (X□□□), and output address (Y□□□) each. When reading the A/D–converted digital value from the input address (X□□□) or when writing the D/A–converting value to the output address (Y□□□), readout and write–in must always be done in word (16 bits) units.

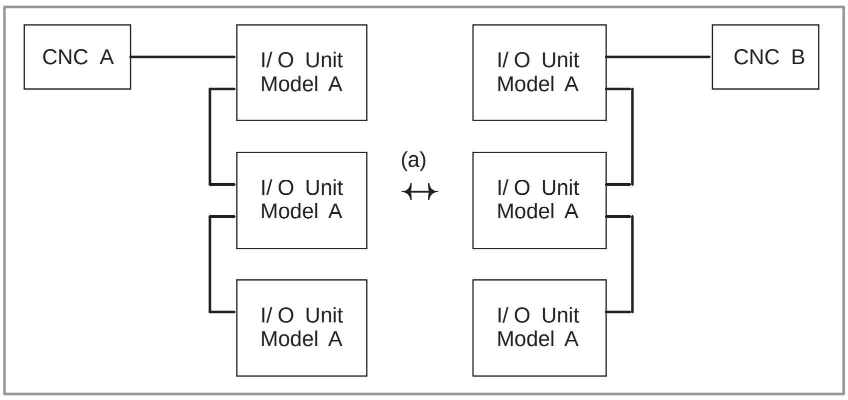
ADDRESS	GROUP	BASE	SLOT	NAME
X000	0	0	1	ID16C
X001	0	0	1	ID16C
X002	0	0	2	ID16D
X003	0	0	2	ID16D
X004	1	0	1	IA16G
X005	1	0	1	IA16G
X006	1	0	2	IA16G
X007	1	0	2	IA16G
X008	2	0	1	ID16D
X009	2	a 0	1	ID16D

Fig. 3.2.2 (e)

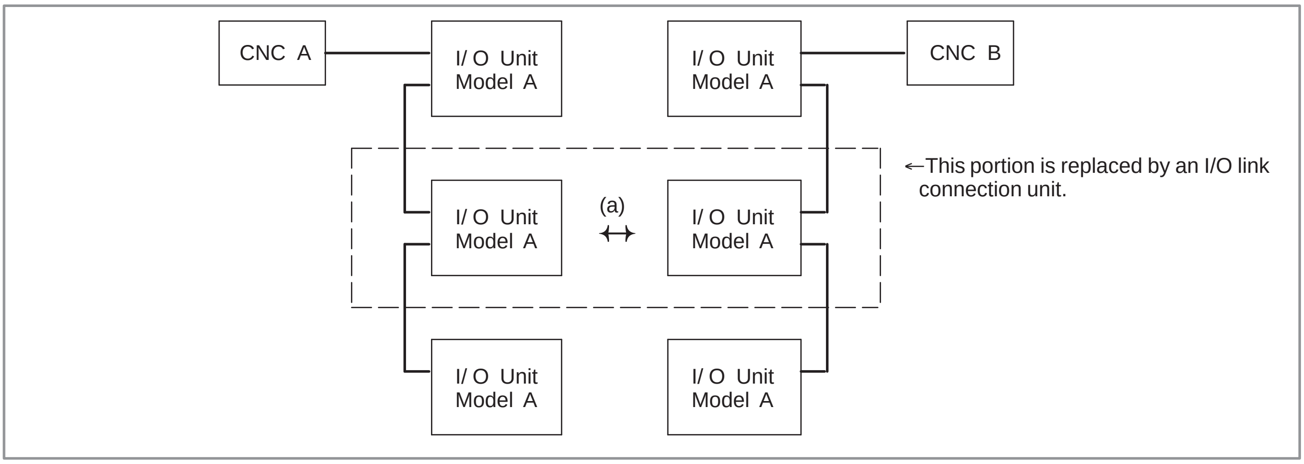
3.2.3 I/O Link Connection Unit Assignment

Concept:

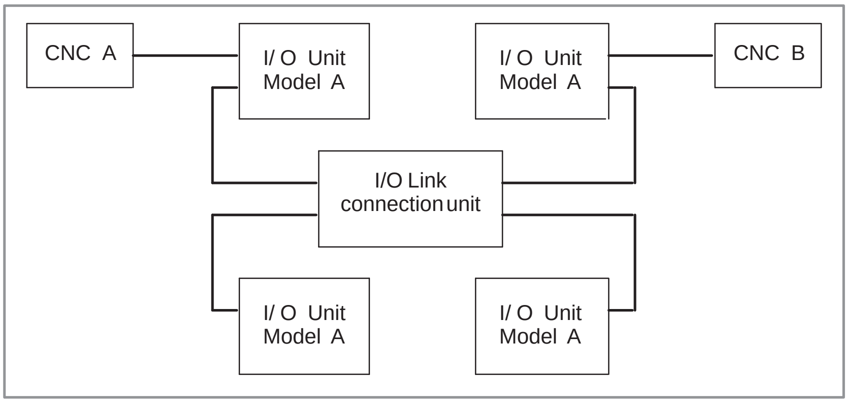
In conventional data transfer, when data is to be transferred between CNC A and CNC B, the I/O units indicated by (a) (figure below) must be connected with each other. (In this case, data can be transferred using any I/O unit.)



The I/O link connection unit replaces these I/O units, thus eliminating the need to connect them with, for example, cables.



Consequently, when the I/O link connection unit is used, the connections become as shown below.



Method of assignment:

The assignment data depends on what type of I/O unit is to be replaced with an I/O link connection unit.

Occupied address	Input unit name at the time of assignment	Output unit name at the time of assignment
1 to 8	/ □ (□ represents a number from 1 to 8.)	/ □ (□ represents a number from 1 to 8.)
16	OC02I	OC02O
32	OC03I	OC03O

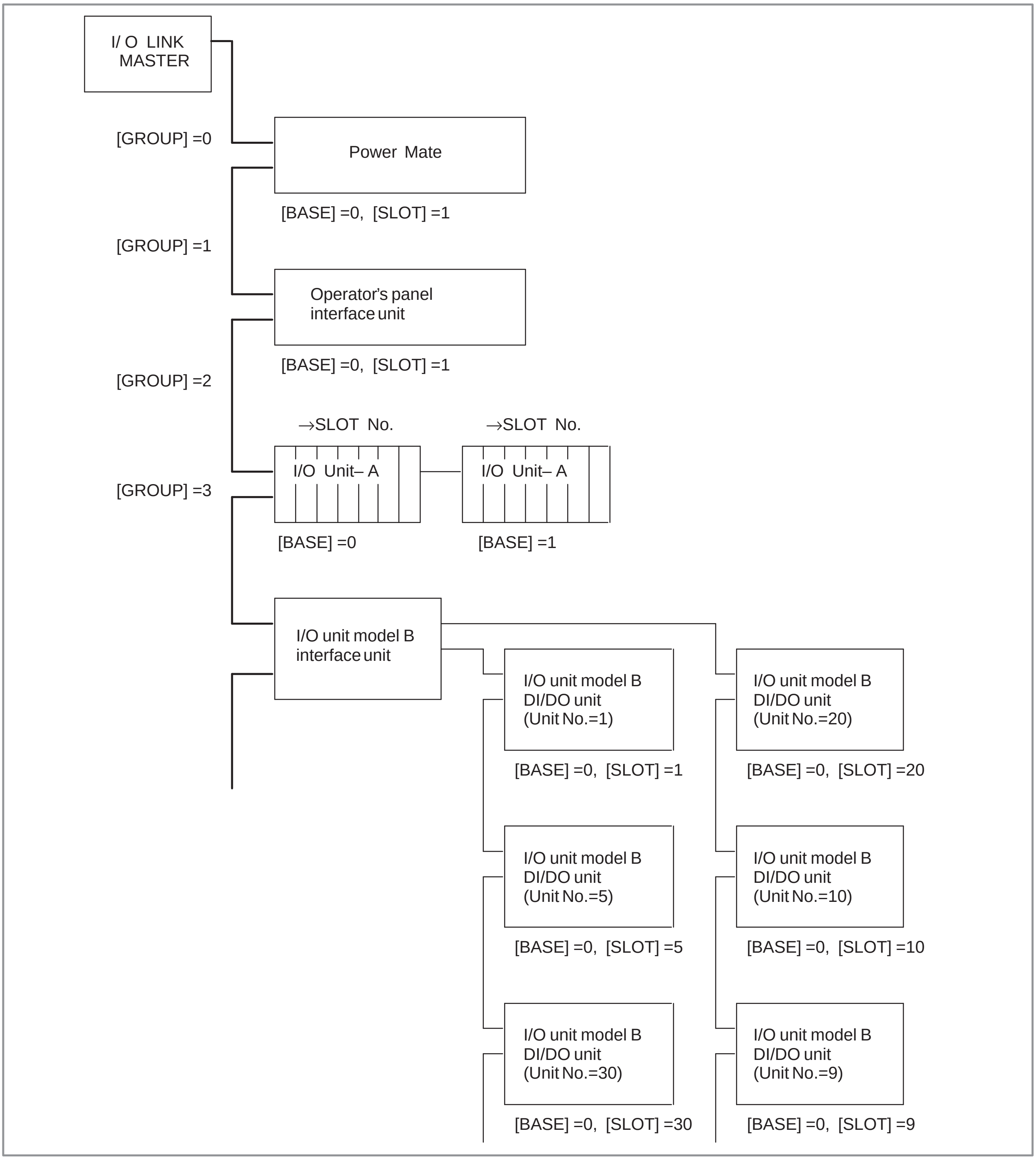
Setting:

When a connection unit that occupies 16–byte addresses is attached to the input side in GROUP = 1, enter "1.0.1.OC02I."

3.2.4 I/O Unit MODEL B Assignment

- Related hardware publications:
  FANUC I/O Unit–MODEL B Connection and Maintenance Manual (B–62163E)

I/O Unit MODEL Bs can be used together with a Power Mate operator panel interface unit, connection unit, and I/O Unit MODEL As.
In this case, the I/O Unit MODEL Bs occupy one group; that is, no other type of unit can be present in that group.
An example of connection is shown below.



Method of assignment:

Specify a group number in [GROUP]. Always specify 0 in [BASE]. Specify the unit number of an I/O unit model B in [SLOT]. But when you assign the power–on/off intormation, specify 0 in [SLOT].
The data specified by [SLOT] and [NAME] is as follows:
[SLOT] = 0, 1, ...30:
Unit number (1 to 30) of an I/O unit model B DI/DO unit
[NAME]: Addresses occupied by an I/O unit model B

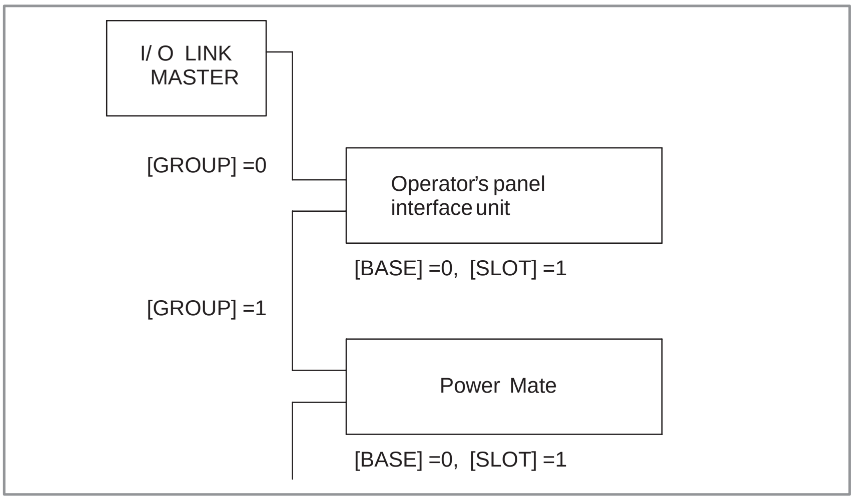
Input/output size of ([base unit] + [extended unit])	Assigned name	Occupied address
1 byte	#1	Input/output: 1 byte
2 bytes	#2	Input/output: 2 bytes
3 bytes	#3	Input/output: 3 bytes
4 bytes	#4	Input/output: 4 bytes
6 bytes	#6	Input/output: 6 bytes
8 bytes	#8	Input/output: 8 bytes
10 bytes	#10	Input/output: 10 bytes
Power–on/off information	##	Input: 4 bytes

Setting:

When an I/O unit model B assigned unit number 10 and occupying an area of 3 bytes is attached to the input with GROUP = 1, enter "1.0.10.#3."

3.2.5 Power Mate–MODEL D/H Assignment

When a Power Mate–MODEL D/H is used as I/O Link slave, it need to be assigned on the I/O Link master side.
On the I/O link slave side, fixed addresses are used, so that no address needs to be assigned. (See Table 3 for the addresses used.)
An example of connection is shown below.



Method of assignment:

Specify a group number in [GROUP].
Always specify 0 in [BASE].
Always specify 1 in [SLOT].
The data specified by [NAME] is as follows:

I/O points (input/output)	Input unit name at the time of assignment	Output unit name at the time of assignment
32/32	FS04A	FS04A
64/64	FS08A	FS08A
128/128	OC02I	OC02O
256/256	OC03I	OC03O

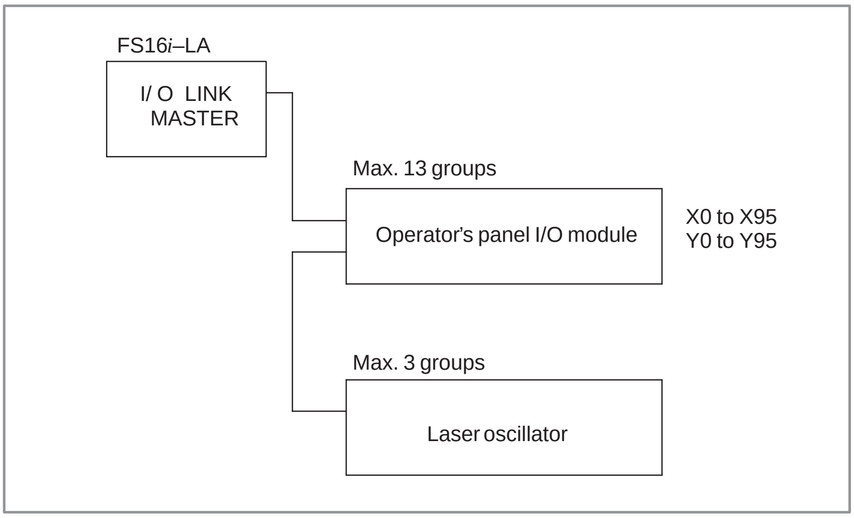
Setting:

When a Power Mate–D of 256/256 points is connected with group 1, input the undermentioned assignment data.
·Input side : "1.0.1.OC03I"
·Output side : "1.0.1.OC03O"

3.2.6 FS16*i*–LA Assignment

On the FS16*i*–LA, the laser oscillator is connected as part of the I/O Link. Three groups (DI/DO=256/256 points) are used for the laser oscillator interface. 13 groups, 768/768 points (X0 to X95, Y0 to Y95) are used for the ladder diagram. For details on the laser oscillator connection, see the FS16i Series CONNECTION MANUAL (B–63003EN), FANUC I/O Link connection.

Connection Example



Method of Assignment

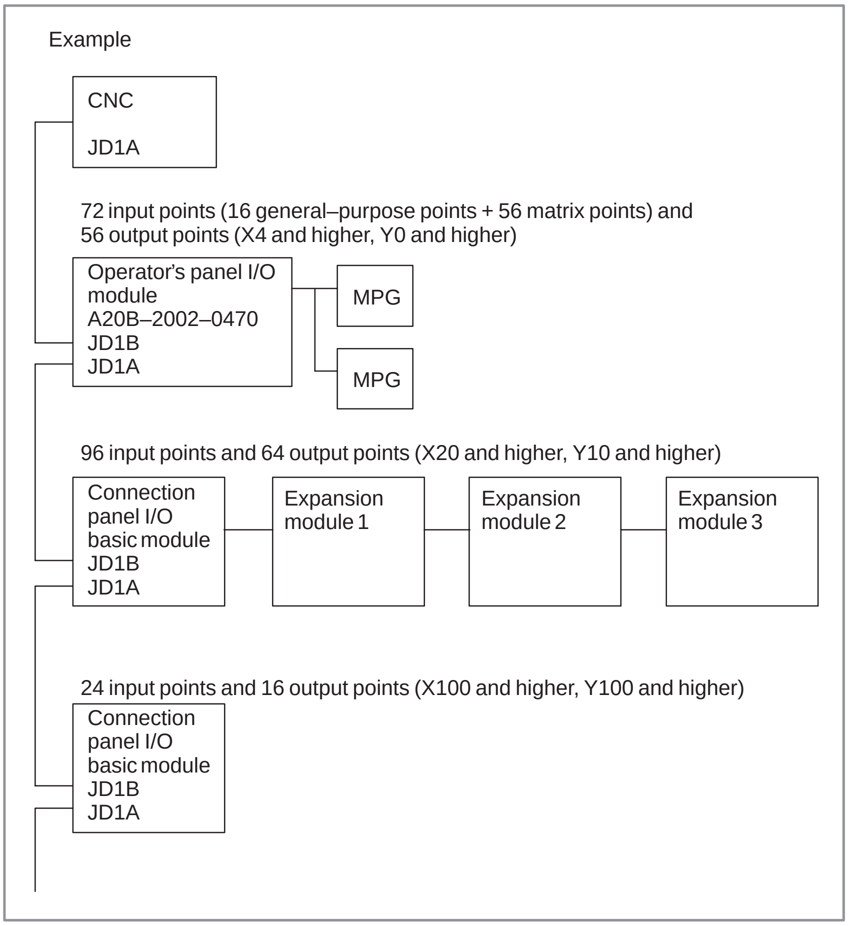
Specify 0 to 12 in [GROUP].
For details on how to assign [BASE] and [SLOT], see 3.2.2 Assignment of I/O Module Addresses.

CAUTION
1 Addresses X96 to X127 and Y96 to Y127 cannot be used in ladder diagrams.
2 When addresses X96 to X127 and Y96 to Y127 are assigned, I/O points are not linked.
3 Connect the laser group to the end of the I/O link.
4 The I/O Link restart function cannot be used.
5 The OVERRIDE mode of the forced I/O function cannot be used.
6 PMC–SB5 does not support a laser.

3.2.7 Distribution I/O Connection I/O Module and Distribution I/O Operator's Panel I/O Module Assignment Methods

To assign connection panel I/O and operator's panel I/O modules to the I/O Link, set the group number with an I/O Link serial number (use a smaller number toward the I/O Link master CNC, like 0, 1, and 2). Also, set the base number and slot number to 0 and 1, respectively. If a combination of the basic module and expansion modules is used as the connection panel I/O module, assign all the modules in one I/O Link group as one unit. Unlike the I/O Unit MODEL–A, it is unnecessary to specify a slot number. An assignment example is shown below.

Assignment example



	Group	Base	Slot	Name
X004	0	0	1	CM14I
X020	1	0	1	CM12I
X100	2	0	1	CM03I
Y000	0	0	1	CM08O
Y010	1	0	1	CM08O
Y100	2	0	1	CM02O

Assignment name

To assign the connection panel I/O and operator's panel I/O modules for the I/O Link, it is necessary to use programming software that supports these modules. If the programming software does not support the modules, use "compatible names for assignment" described later.

Connection panel I/O

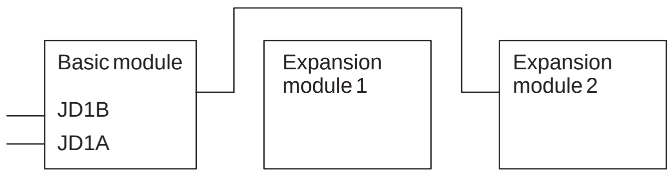
Refer to the following manuals for an explanation of how the connection panel I/O module signals are mapped:

- "FANUC Series 16*i*/18*i* Connection Manual (Hardware)" B–63003EN
- "FANUC Series 21*i* Connection Manual (Hardware)" B–63083EN
- "FANUC Series 15*i* Connection Manual (Hardware)" B–63323EN
- "FANUC Power Mate *i* Connection Manual (Hardware)" B–63173EN

The assignment that is made for different configurations (such as basic module configuration and combination basic/expansion module configuration) is explained below.

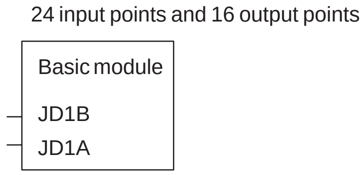
CAUTION

Expansion modules must be connected in ascending order with respect to the module number (1, 2, then 3). An expansion module number cannot be skipped.



It is impossible to assign expansion module 2 without expansion module 1, which will be installed later, as shown above.

(1) Basic module only



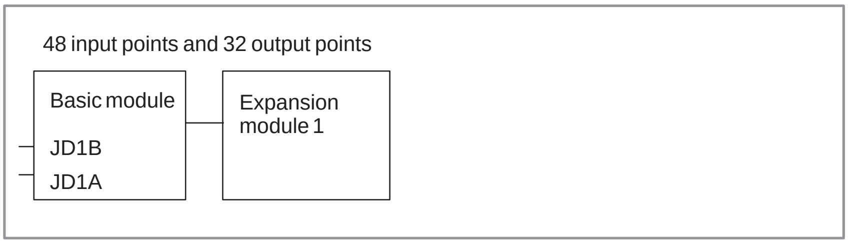
(a) If DO alarm detection is not used

- No manual pulse generator:
  Input X = "CM03I" and output Y = "CM02O"
- One manual pulse generator:
  Input X = "CM13I" and output Y = "CM02O"
- Two manual pulse generators:
  Input X = "CM14I" and output Y = "CM02O"
- Three manual pulse generators:
  Input X = "CM15I" and output Y = "CM02O"

(b) If DO alarm detection is used

- Input X = "CM16I" and output Y = "CM02O" no matter how many manual pulse generators are used

(2) Basic module + expansion module 1



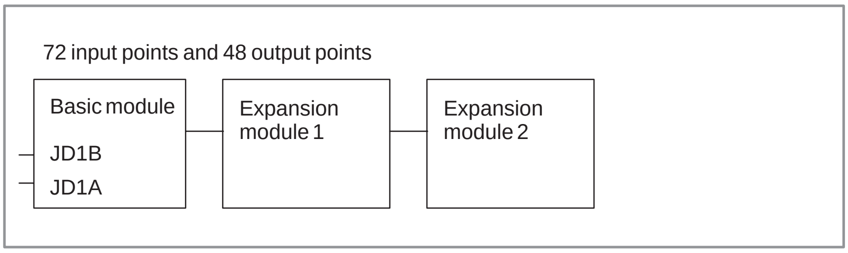
(a) If DO alarm detection is not used

- No manual pulse generator:
  Input X = “CM06I” and output Y = “CM04O”
- One manual pulse generator:
  Input X = “CM13I” and output Y = “CM04O”
- Two manual pulse generators:
  Input X = “CM14I” and output Y = “CM04O”
- Three manual pulse generators:
  Input X = “CM15I” and output Y = “CM04O”

(b) If DO alarm detection is used

- Input X = “CM16I” and output Y = “CM04O” no matter how many manual pulse generators are used

(3) Basic module + expansion module 1 + expansion module 2



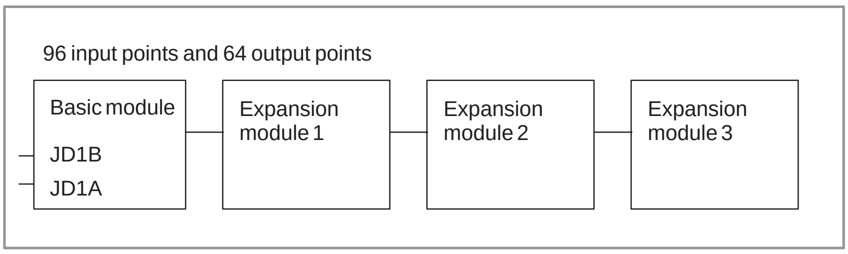
(a) If DO alarm detection is not used

- No manual pulse generator:
  Input X = “CM09I” and output Y = “CM06O”
- One manual pulse generator:
  Input X = “CM13I” and output Y = “CM06O”
- Two manual pulse generators:
  Input X = “CM14I” and output Y = “CM06O”
- Three manual pulse generators:
  Input X = “CM15I” and output Y = “CM06O”

(b) If DO alarm detection is used

- Input X = “CM16I” and output Y = “CM06O” no matter how many manual pulse generators are used

(4) Basic module + expansion module 1 + expansion module 2 + expansion module 3



(a) If DO alarm detection is not used

- No manual pulse generator:
  Input X = “CM12I” and output Y = “CM08O”
- One manual pulse generator:
  Input X = “CM13I” and output Y = “CM08O”
- Two manual pulse generators:
  Input X = “CM14I” and output Y = “CM08O”
- Three manual pulse generators:
  Input X = “CM15I” and output Y = “CM08O”

(b) If DO alarm detection is used

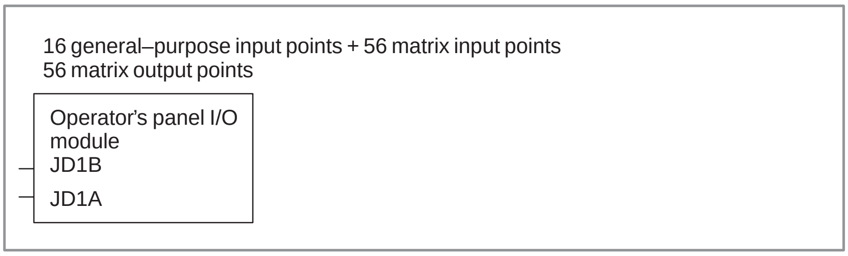
- Input X = “CM16I” and output Y = “CM08O” no matter how many manual pulse generators are used

Operator’s panel I/O module

Refer to the following manuals for an explanation how the operator’s panel I/O module signals are mapped:

- “FANUC Series 16*i*/18*i* Connection Manual (Hardware)” B–63003EN
- “FANUC Series 21*i* Connection Manual (Hardware)” B–63083EN
- “FANUC Series 15*i* Connection Manual (Hardware)” B–63323EN
- “FANUC Power Mate *i* D/H Connection Manual (Hardware)” B–63173EN

(1) Operator’s panel I/O module
(A20B–2002–0470 supporting matrix inputs)



(a) If DO alarm detection is not used

- No manual pulse generator:
  Input X = “CM12I” and output Y = “CM08O”
- One manual pulse generator:
  Input X = “CM13I” and output Y = “CM08O”
- Two manual pulse generators:
  Input X = “CM14I” and output Y = “CM08O”
- Three manual pulse generators:
  Input X = “CM15I” and output Y = “CM08O”

(b) If DO alarm detection is used

- Input X = “CM16I” and output Y = “CM08O” no matter how many manual pulse generators are used

(2) Operator’s panel I/O modules
(A20B–2002–0520 and A20B–2002–0521)

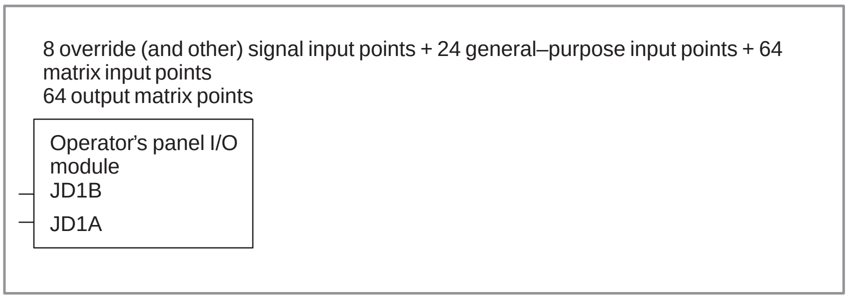
(a) If DO alarm detection is not used

- No manual pulse generator:
  Input X = “CM06I” and output Y = “CM04O”
- One manual pulse generator:
  Input X = “CM13I” and output Y = “CM04O”
- Two manual pulse generators:
  Input X = “CM14I” and output Y = “CM04O”
- Three manual pulse generators:
  Input X = “CM15I” and output Y = “CM04O”

(b) If DO alarm detection is used

- Input X = “CM16I” and output Y = “CM04O” no matter how many manual pulse generators are used

(3) Distribution I/O machine operator’s panels (A20B–8001–0721, A20B–8001–0720, and A20B–8001–0210)



(a) If DO alarm detection is not used

- No manual pulse generator:
  Input X = “CM12I” and output Y = “CM08O”
- One manual pulse generator:
  Input X = “CM13I” and output Y = “CM08O”
- Two manual pulse generators:
  Input X = “CM14I” and output Y = “CM08O”
- Three manual pulse generators:
  Input X = “CM15I” and output Y = “CM08O”

(b) If DO alarm detection is used

- Input X = “CM16I” and output Y = “CM08O” no matter how many manual pulse generators are used

Compatible names for assignment

If your programming unit does not support the connection I/O or operator's panel I/O module, use the following compatible names for I/O Link assignment.

Assignment name	Compatible name
CM03I	/3
CM06I	/6
CM09I	OC01I
CM12I	OC01I
CM13I	OC02I
CM14I	OC02I
CM15I	OC02I
CM16I	OC02I
CM02O	/2
CM04O	/4
CM06O	/6
CM08O	/8

3.3 INTERNAL RELAY ADDRESSES (R)

In each model, the following signals (bytes) can be used as internal relays. This area is cleared to zero when the power is turned on.

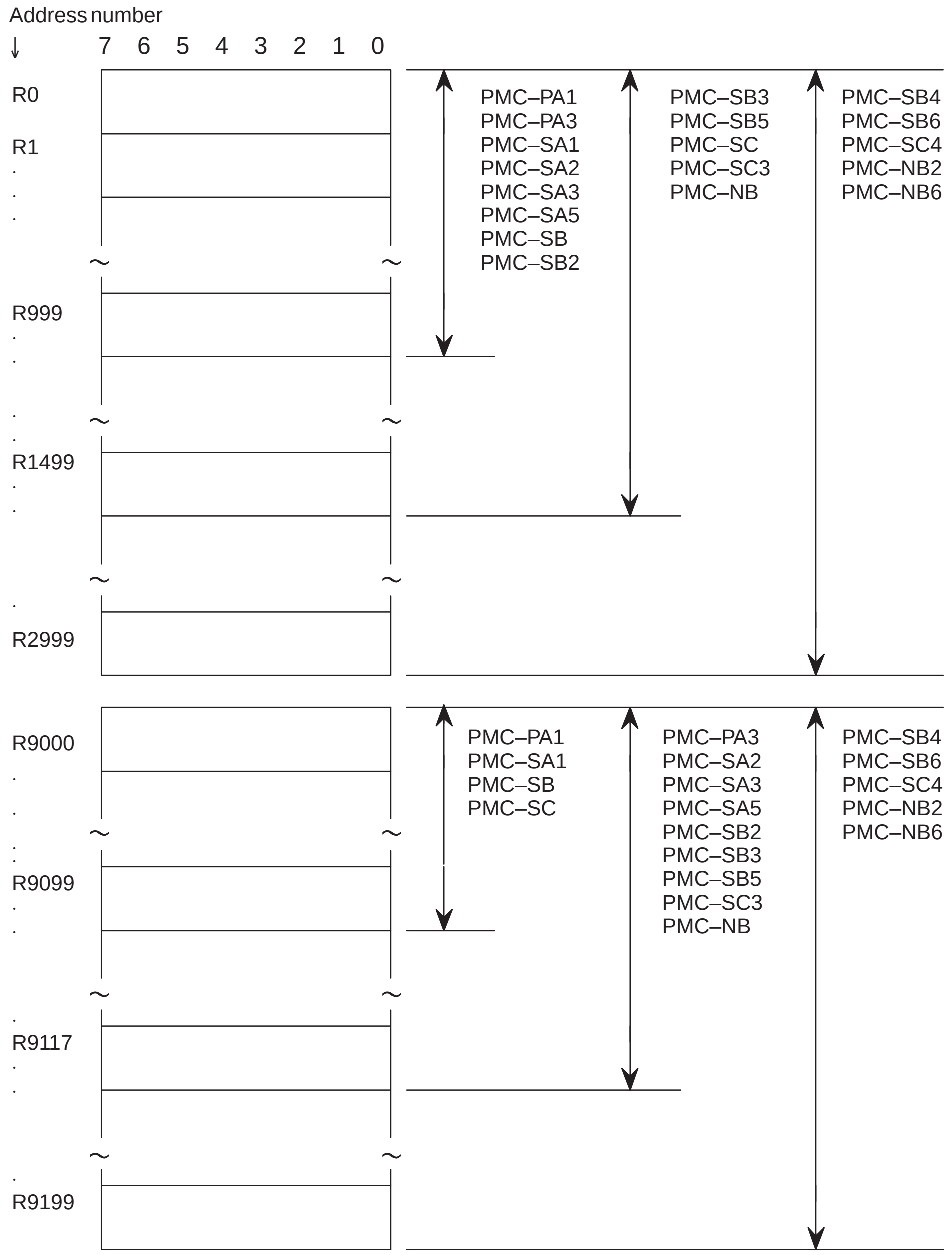
Model	PA1	PA3
Number of bytes	1100	1118

Model	SA1	SA2	SA3/SA5
Number of bytes	1100	1118	1118

Model	SB	SB2	SB3/SB5	SB4/SB6
Number of bytes	1100	1118	1618	3200

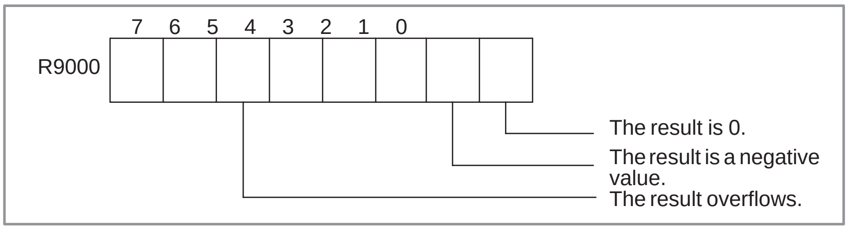
Model	SC	SC3	SC4
Number of bytes	1600	1618	3200

Model	NB	NB2	NB6
Number of bytes	1618	3200	3200

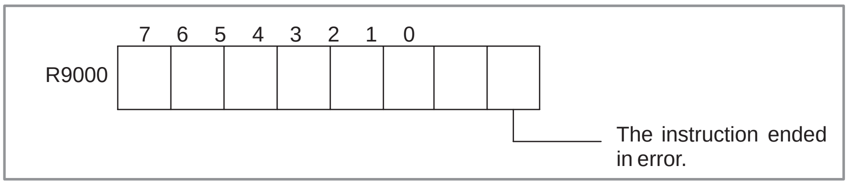

Address number
7 6 5 4 3 2 1 0
R0
R1
R999
R1499
R2999
PMC–PA1
PMC–PA3
PMC–SA1
PMC–SA2
PMC–SA3
PMC–SA5
PMC–SB
PMC–SB2
PMC–SB3
PMC–SB5
PMC–SC
PMC–SC3
PMC–NB
PMC–SB4
PMC–SB6
PMC–SC4
PMC–NB2
PMC–NB6
R9000
R9099
R9117
R9199
PMC–PA1
PMC–SA1
PMC–SB
PMC–SC
PMC–PA3
PMC–SA2
PMC–SA3
PMC–SA5
PMC–SB2
PMC–SB3
PMC–SB5
PMC–SC3
PMC–NB
PMC–SB4
PMC–SB6
PMC–SC4
PMC–NB2
PMC–NB6

3.3.1 Area Managed by the System Program

(1) R9000 (Operation output register for the ADDB, SUBB, MULB, DIVB, and COMPB functional instructions)

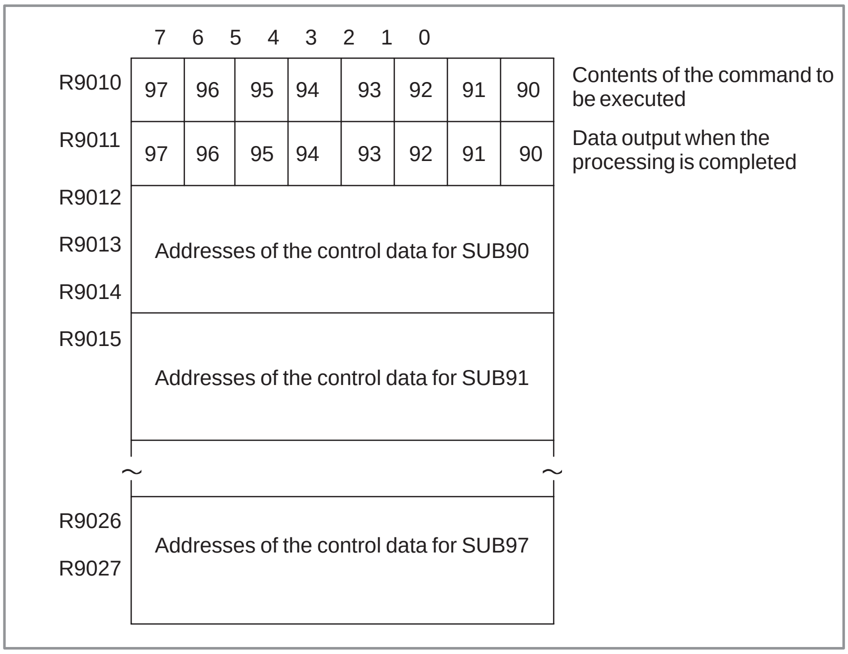


(2) R9000 (Error output for the EXIN, WINDR, WINDW, MMCWR, MMCWW, MMC3R, and MMC3W functional instructions)

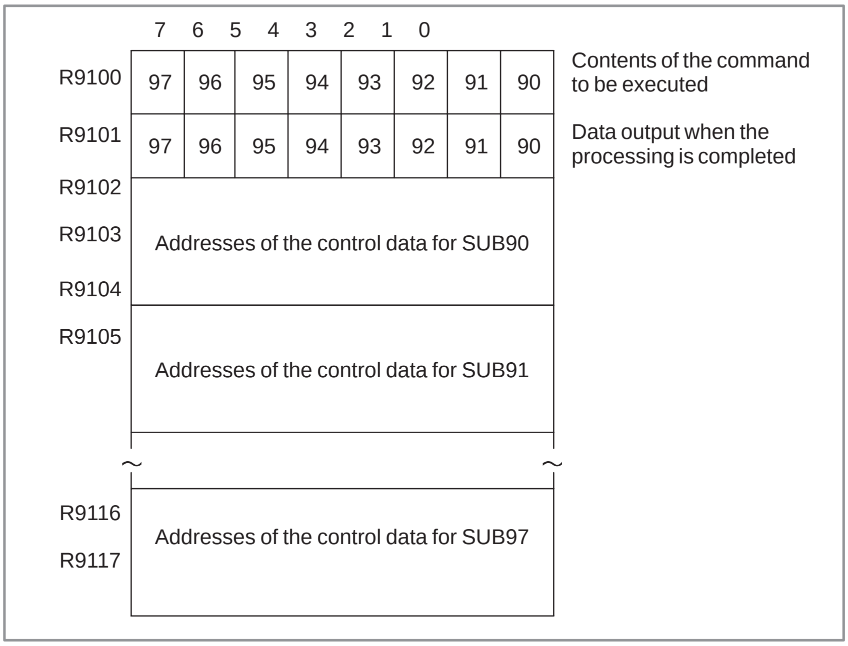


(3) R9002 to R9005 (Operation output registers for the DIVB functional instruction)
The data remaining after the DIVB functional instruction is executed is output.

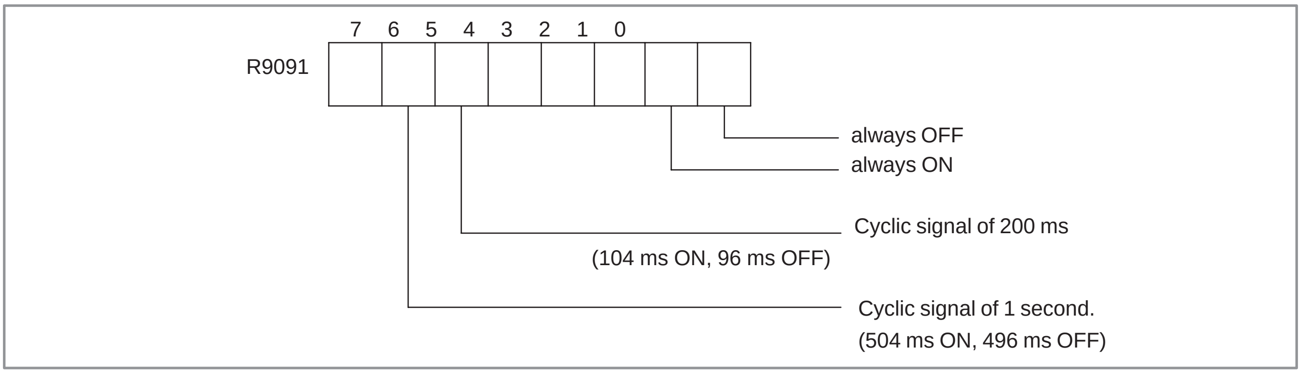
(4) R9010 to R9027 (Interface area for the FNC9x functional instruction) (PMC–SC only)
The area is provided as an interface between the FNC9x functional instruction to be executed and a desired function.



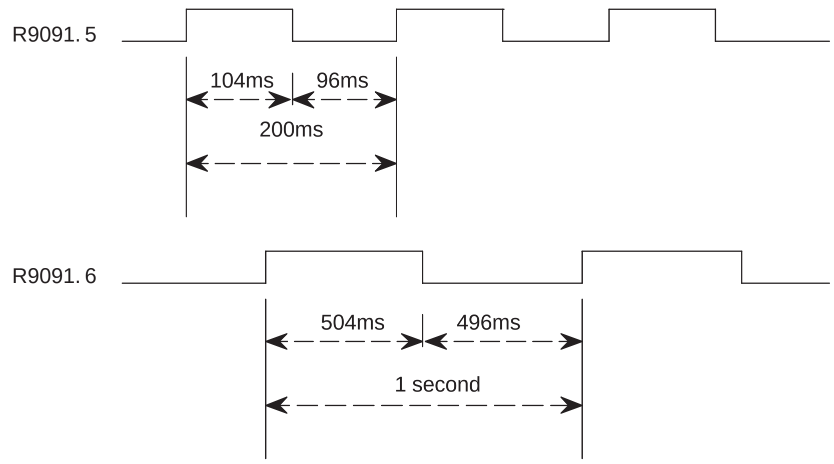
(5) R9100 to R9117 (Interface area for the FNC9x functional instruction) (PMC–SC3/SC4/NB/NB2 only)
The area is provided as an interface between the FNC9x functional instruction to be executed and a desired function.



(6) R9091 (System timer)
4 signals can be used as system timer.
The specifications of every signal are as following.



CAUTION
In the beginning, every signal is OFF.
The signals of R9091.0 and R9091.1 are always set at the beginning of 1st level in every cycle.
Every pulse signal (ON–OFF) includes ±8 ms errors.


R9091. 5
104ms
96ms
200ms
R9091. 6
504ms
496ms
1 second

3.4 ADDRESSES FOR MESSAGE SELECTION DISPLAYED ON CRT (A)

This area is used as message display request. In each model, the following number of messages can be used. Where “Number of Messages” = “Number of Bytes” $\times$ 8

This area is cleared to zero when the power is turned on. For information about using the message, see the subsection “5.43”.

Model	PA1	PA3
Number of bytes	25	25
Number of messages	200	200

Model	SA1	SA2	SA3/ SA5
Number of bytes	25	25	25
Number of messages	200	200	200

Model	SB	SB2	SB3/ SB5	SB4/ SB6
Number of bytes	25	25	25	125
Number of messages	200	200	200	1000

Model	SC	SC3	SC4
Number of bytes	25	25	125
Number of messages	200	200	1000

Model	NB	NB2	NB6
Number of bytes	25	125	125
Number of messages	200	1000	1000

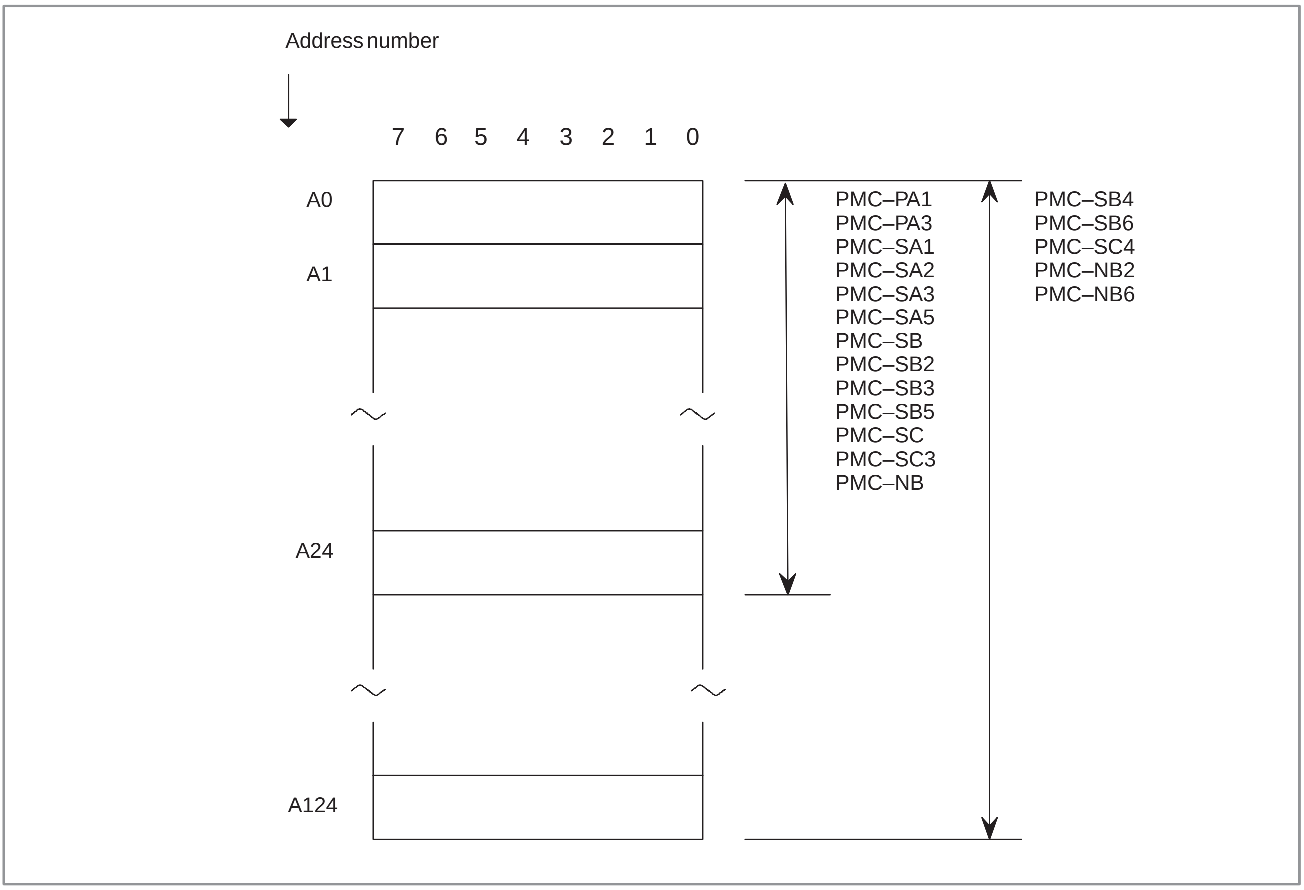


Fig. 3.4 Address of message display reguest

3.5 ADDRESS OF COUNTER (C)

This area is used as counters. In each model, the following number of counters can be used. Where “Number of Counters” = “Number of Bytes” / 4
Since this area is nonvolatile, the contents of the memory do not disappear even when the power is turned off.

Model	PA1	PA3
Number of bytes	80	80
Number of counters	20	20

Model	SA1	SA2	SA3/ SA5
Number of bytes	80	80	80
Number of counters	20	20	20

Model	SB	SB2	SB3/ SB5	SB4/ SB6
Number of bytes	80	80	80	200
Number of counters	20	20	20	50

Model	SC	SC3	SC4
Number of bytes	80	80	200
Number of counters	20	20	50

Model	NB	NB2	NB6
Number of bytes	80	200	200
Number of counters	20	50	50

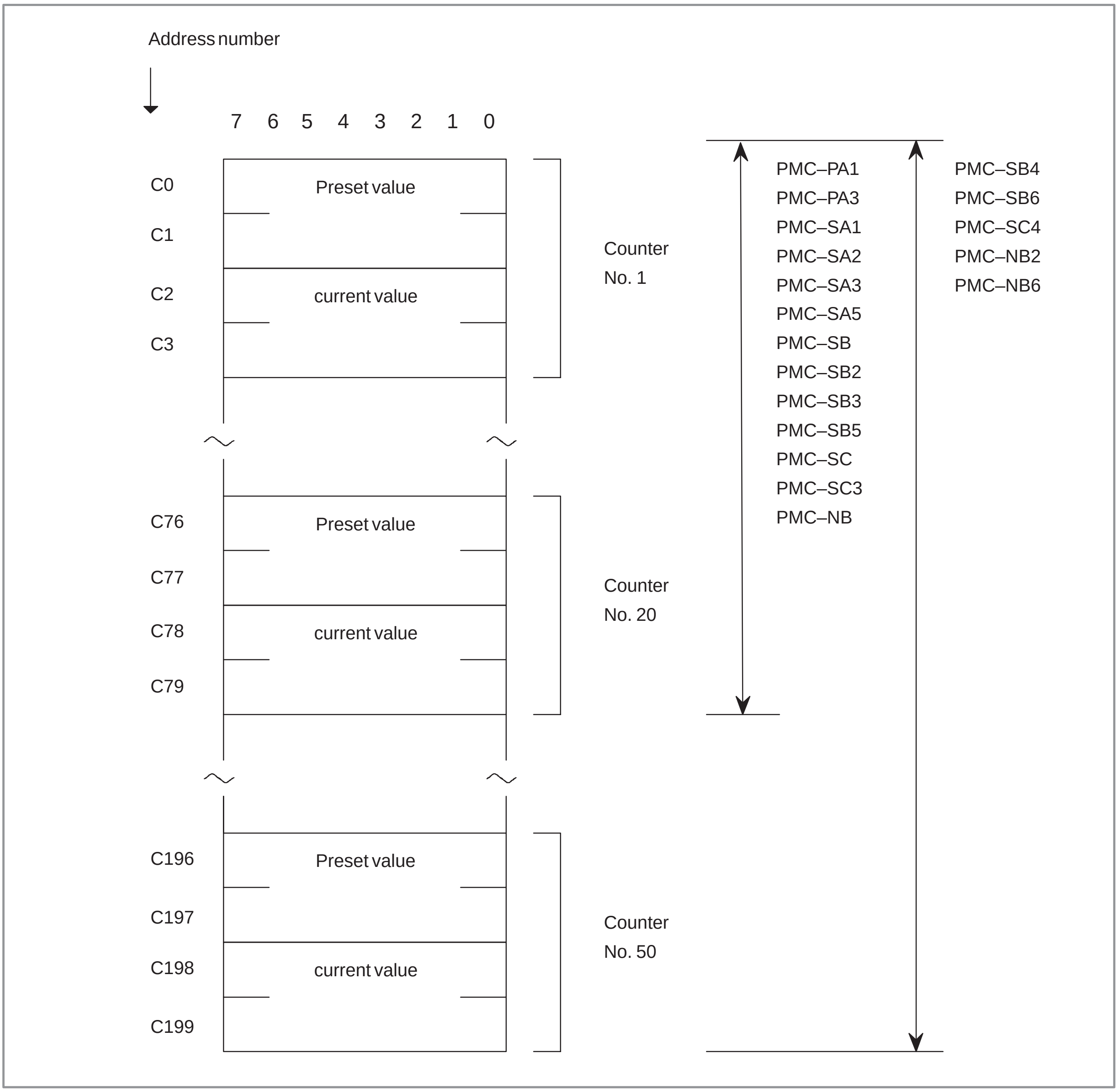


Fig. 3.5 Address of Counter

3.6 ADDRESS OF KEEP RELAY AND NONVOLATILE MEMORY CONTROL (K)

The area is used as keep relays and PMC parameters. In each model, the following number of bytes can be used. Since this area is nonvolatile, the contents of the memory do not disappear even when the power is turned off.

Model	PA1	PA3
Number of bytes	20	20
Nonvolatile memory control address	K16	K16
PMC control software parameter	K17 to K19	K17 to K19

Model	SA1	SA2	SA3/ SA5
Number of bytes	20	20	20
Nonvolatile memory control address	K16	K16	K16
PMC control software parameter	K17 to K19	K17 to K19	K17 to K19

Model	SB	SB2	SB3/ SB5	SB4/ SB6
Number of bytes	20	20	20	50
Nonvolatile memory control address	K16	K16	K16	K16
PMC control software parameter	K17 to K19	K17 to K19	K17 to K19	K900 to K909

Model	SC	SC3	SC4
Number of bytes	20	20	50
Nonvolatile memory control address	K16	K16	K16
PMC control software parameter	K17 to K19	K17 to K19	K900 to K909

Model	NB	NB2	NB6
Number of bytes	20	50	50
Nonvolatile memory control address	K16	K16	K16
PMC control software parameter	K17 to K19	K900 to K909	K900 to K909

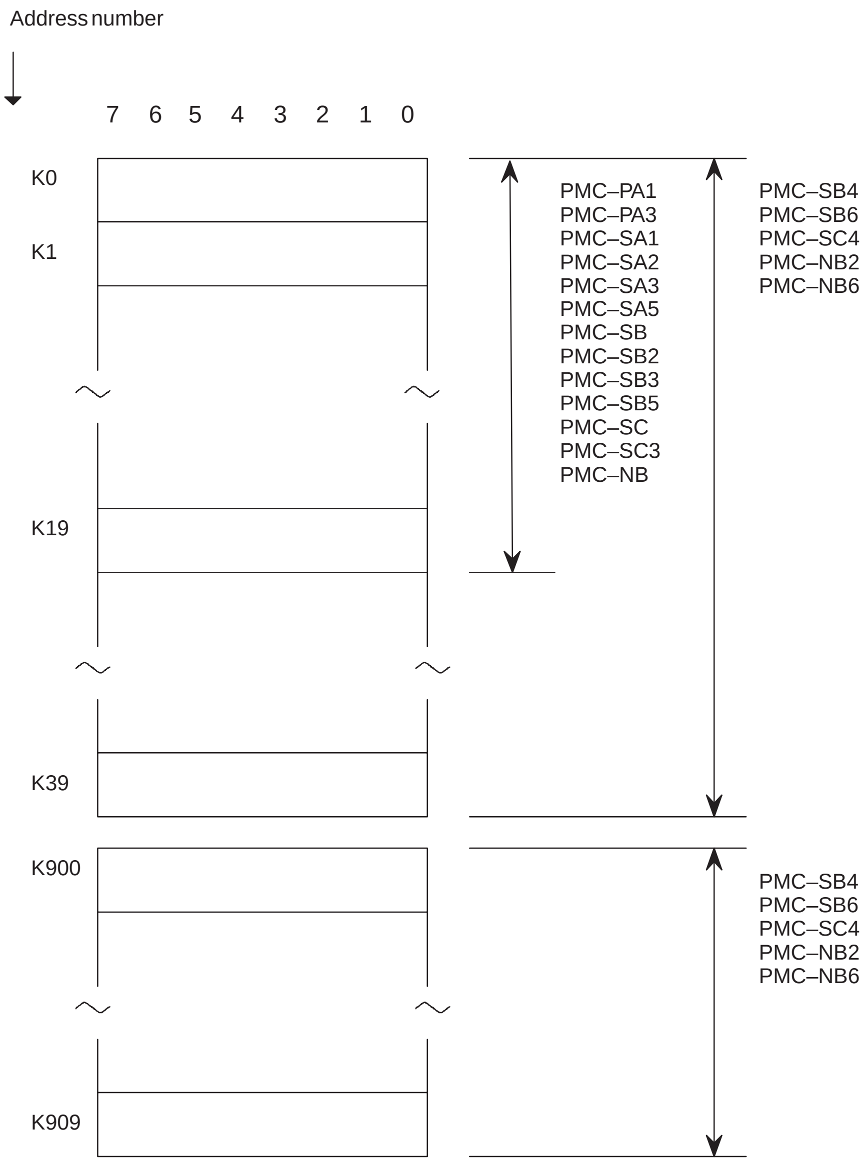


Fig. 3.6 Address of keep relay and nonvolative memory control

For the information about using “Nonvolatile memory control”, see the section “6.1”.

PMC control software parameter area is used by PMC control software. For more information about PMC control software parameter, see the section “II 4.3”.

3.7 ADDRESS OF DATA TABLE (D)

Data table is the area of nonvolatile memory. In each model, the following number of bytes can be used.

Model	PA1	PA3
Number of bytes	1860	1860

Model	SA1	SA2	SA3/ SA5
Number of bytes	1860	1860	1860

Model	SB	SB2	SB3/ SB5	SB4/ SB6
Number of bytes	1860	1860	3000	8000

Model	SC	SC3	SC4
Number of bytes	3000	3000	8000

Model	NB	NB2	NB6
Number of bytes	3000	8000	8000

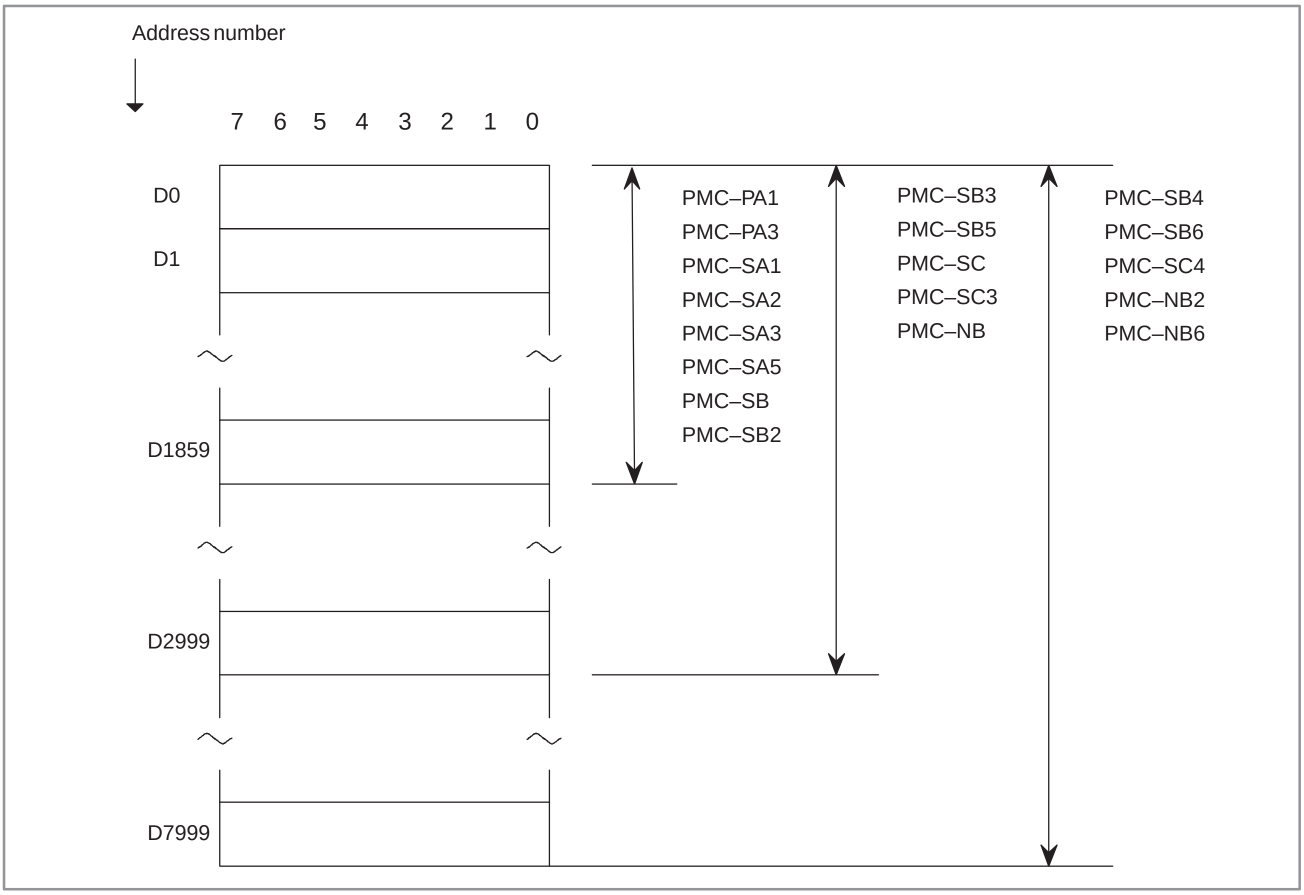


Fig. 3.7 Address of data table

3.8 TIMER ADDRESSES (T)

This area is used by TMR instruction as variable timers. In each model, the following number of timers can be used. Where "Number of timers" = "Number of Bytes" / 2
Since this area is nonvolatile, the contents of the memory do not disappear even when the power is turned off.

Model	PA1	PA3
Number of bytes	80	80
Number of timers	40	40

Model	SA1	SA2	SA3/ SA5
Number of bytes	80	80	80
Number of timers	40	40	40

Model	SB	SB2	SB3/ SB5	SB4 SB6
Number of bytes	80	80	80	300
Number of timers	40	40	40	150

Model	SC	SC3	SC4
Number of bytes	80	80	300
Number of timers	40	40	150

Model	NB	NB2	NB6
Number of bytes	80	300	300
Number of timers	40	150	150

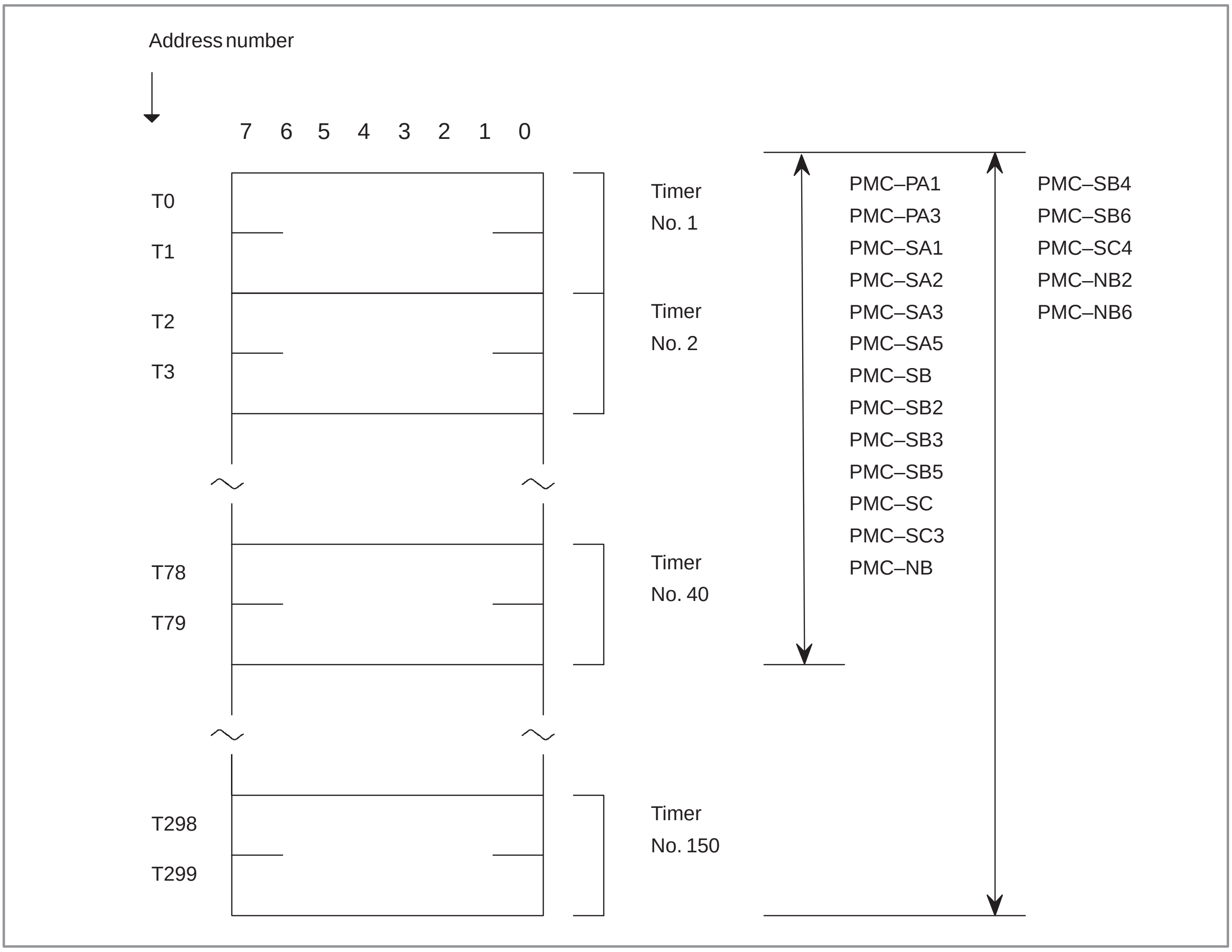


Fig. 3.8 Timer address

3.9 LABEL ADDRESSES (JMPB, JMPC, LBL) (L)

Label addresses are used to specify jump destination labels (positions in a sequence program) in the JMPB and JMPC instructions. The same label number can appear in different LBL instructions in the same sequence program as long as it is unique in the program unit (main program, subprogram). In each model, the following number of label can be used.

Model	PA1	PA3
Number of labels	–	9999

Model	SA1	SA2	SA3/ SA5
Number of labels	–	–	9999

Model	SB	SB2	SB3/ SB5	SB4/ SB6
Number of labels	–	–	9999	9999

Model	SC	SC3	SC4
Number of labels	–	9999	9999

Model	NB/ NB2	NB6
Number of labels	9999	9999

3.10 SUBPROGRAM NUMBERS (CALL, CALLU, SP) (P)

Subprogram numbers are used to specify jump destination subprogram labels in the CALL and CALLU instructions. Subprogram number must be unique in the entire sequence program. In each model, the following number of subprograms can be used.

Model	PA1	PA3
Number of subprograms	–	512

Model	SA1	SA2	SA3/ SA5
Number of subprograms	–	–	512

Model	SB	SB2	SB3/ SB5	SB3/ SB6
Number of subprograms	–	–	512	2000

Model	SC	SC3	SC4
Number of subprograms	–	512	2000

Model	NB	NB2	NB6
Number of subprograms	512	2000	2000

4 PMC BASIC INSTRUCTIONS

Designing a sequence program begins with writing a ladder diagram. The ladder diagram is written using relay contact symbols and functional instruction code. (These will be described later.) Logic written in the ladder diagram is entered as a sequence program in the Programmer.

There are two sequence program entry methods. One is the entry method with the mnemonic language (PMC instructions such as RD, AND and OR). The other is the relay symbol method (┤├, ┤/├ and -○-) in which the sequence program is entered by using the relay contact symbols and the functional instruction symbols of the ladder diagram. When the relay symbol method is used, the ladder diagram format can be used and programming can be performed without understanding the PMC instructions (basic instructions such as RD, AND and OR).

Actually, however, the sequence program entered by the relay symbol method is also internally converted into the instruction corresponding to the PMC instruction. When the sequence program is punched on a paper tape and then entered to the programmer, programming must be performed with the PMC instructions.

Also, the meanings of the functional instructions described later must be understood fully. See Subsection 4.1 and Section 5.

On how to enter the sequence program into the programmer by using the PMC instructions and relay symbols, see Chapter III or V.

The following should be noted first before reading the explanation on PMC instructions.

This manual describes the entry method using mnemonic language.

(1) Signal address
Relay coils and contacts written in a ladder diagram are each given an address, represented with an address number and a bit number. (See Fig. 4 (a)) It is possible for the head zero. For details of address, see Section 3.

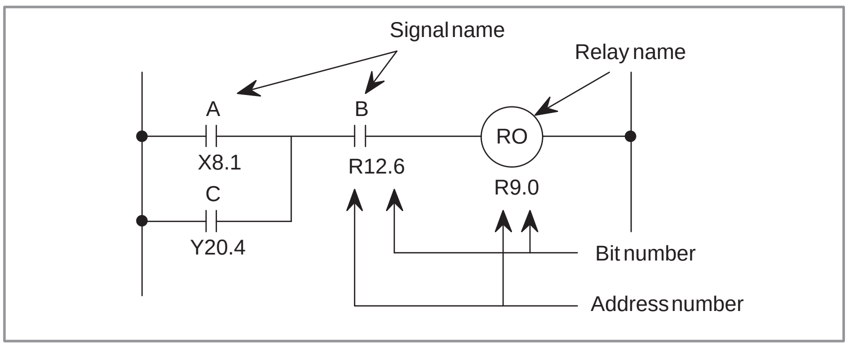


Fig. 4 (a) Address of signal

(2) Type
There are two types of PMC instructions, basic and functional.

(a) Basic instruction
Basic instructions are most often used when designing sequence programs. They perform one–bit operations, such as AND, or OR. There are 12 types.

(b) Functional instruction
Functional instructions ease programming of machine movements that are difficult to program with basic instructions. Refer to Chapter V about the type of functional instruction.

(3) Storage of logical operation results
A register is provided for storing the intermediate results of a logical operation during operation of a sequence program. This register consists of 9 bits. (See Fig. 4 (b)) .

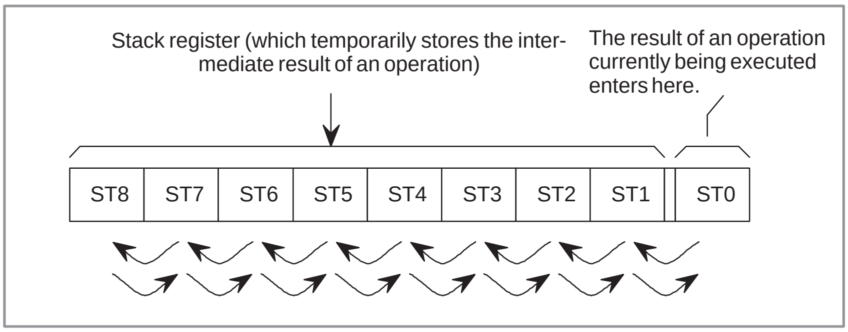


Fig. 4 (b)

Execution of an instruction (RD.STK or the like) to temporarily store the intermediate results of an operation as in the above figure, shifts left and stacks the status stored so far; conversely, execution (AND.STK or the like) to retrieve a stacked signal shifts it right. The signal stacked last is retrieved first.
Refer to explanations of each instruction for concrete applications and operations.

4.1 DETAILS OF BASIC INSTRUCTIONS

The type of instructions and contents of processing are listed in the Table 4.1 (a).

Information format 1:
This is used when writing instructions on a coding sheet, punching out them on a paper tape or displayed on the CRT/MDI or offline programmer.

Information format 2:
This is used when inputting instructions through programmer.
This format is to simplify an input operation.
RN, for instance, means RD.NOT and represents an input operation using both keys, “R” and “N”.

Details of each basic instruction will be given here.

Table 4.1 (a) Basic instruction and processing

No.	Instruction		Contents of processing
	Format 1 (coding)	Format 2 (keys ofFAPT LADDER)	
1	RD	R	Reads the status of a specified signal and sets it in ST0.
2	RD.NOT	RN	Inverts the logical status of a specified signal, reads and sets it in ST0.
3	WRT	W	Outputs the results of logical operations (status of ST0) to a specified address.
4	WRT.NOT	WN	Inverts the results of logical operations (status of ST0) and outputs it to a specified address.
5	AND	A	Induces a logical product.
6	AND.NOT	AN	Inverts the status of a specified signal and induces a logical product.
7	OR	O	Induces a logical sum.
8	OR.NOT	ON	Inverts the status of a specified signal and induces a logical sum.
9	RD.STK	RS	Shifts the stack register left one bit, read and sets the status of a specified signal in ST0.
10	RD.NOT.STK	RNS	Shifts the stack register left one bit reads the inveried logical status of a specified signal, and sets it in ST0.
11	AND.STK	AS	Sets the logical product of ST0 and ST1, and shifts the stack register right one bit.
12	OR.STK	OS	Sets the logical sum of ST0 and ST1, and shifts the stack register right by one bit.
13	SET	SET	Calculates the logical OR of the contents of ST0 and the status of the signal at the specified address and outputs the result to the specified address.
14	RST	RST	Calculates the logical AND of the inverted contents of ST0 and the specified address and outputs the result to the address.

Basic instructions available on each models are as shown in the “Table 4.1 (b)”.

Table 4.1 (b) Basic instruction

No.	Instruction	Model								
		PMC–PA1	PMC–PA3	PMC–SA1/SA2	PMC–SB/SB2	PMC–SC	PMC–SA3/SA5	PMC–SB3/SB4/SB5/SB6	PMC–SC3/SC4	PMC–NB/NB2/NB6
1	RD	○	○	○	○	○	○	○	○	○
2	RD.NOT	○	○	○	○	○	○	○	○	○
3	WRT	○	○	○	○	○	○	○	○	○
4	WRT.NOT	○	○	○	○	○	○	○	○	○
5	AND	○	○	○	○	○	○	○	○	○
6	AND.NOT	○	○	○	○	○	○	○	○	○
7	OR	○	○	○	○	○	○	○	○	○
8	OR.NOT	○	○	○	○	○	○	○	○	○
9	RD.STK	○	○	○	○	○	○	○	○	○
10	RD.NOT.STK	○	○	○	○	○	○	○	○	○
11	AND.STK	○	○	○	○	○	○	○	○	○
12	OR.STK	○	○	○	○	○	○	○	○	○
13	SET	×	○	×	×	×	○	○	○	○
14	RST	×	○	×	×	×	○	○	○	○

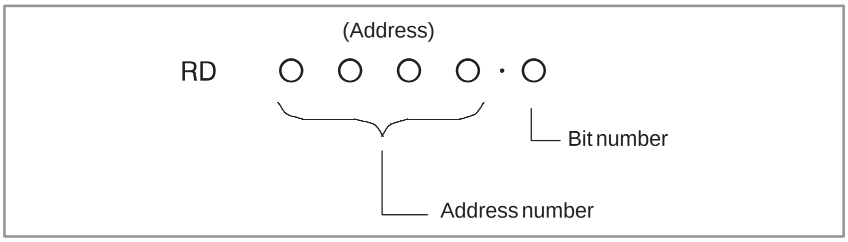
× : Cannot be used ○ : Can be used

NOTE

SET/RST are not available on PMC–SA3 for Series 20.

4.1.1 RD

(1) Format



(2) Reads the status (1 or 0) of a signal at a specified address and sets it in ST0.

(3) Is used when beginning coding with contact A (⊣⊢). See the ladder diagram of Fig. 4.1.1 and entries in the coding sheet of Table 4.1.1 for an example of using the RD instruction.

(4) The signal read by the RD instruction may be any signal entered as the logical condition for one coil (output).

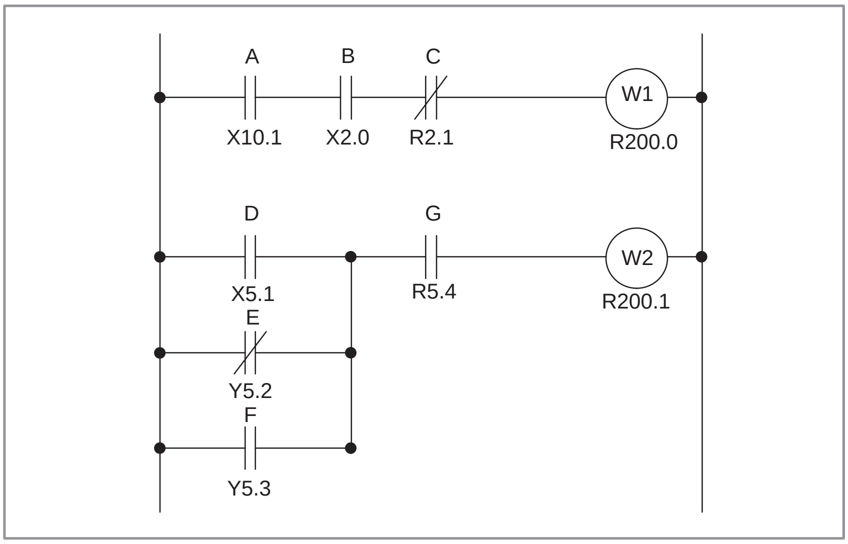


Fig. 4.1.1 Ladder diagram

Table 4.1.1 Coding for Fig. 4.1.1

Coding sheet

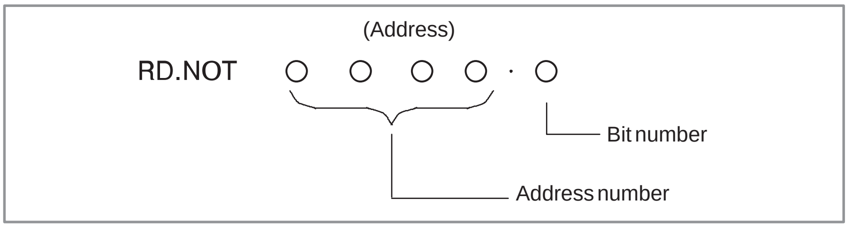
Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	X10 .	1	A
2	AND	X 2 .	0	B
3	AND . NOT	R2 .	1	C
4	WRT	R200 .	0	W1 output
5	RD	X5 .	1	D
6	OR . NOT	Y5 .	2	E
7	OR	Y5 .	3	F
8	AND	R5 .	4	G
9	WRT	R200 .	1	W2 output
10				

Status of operating result

ST2	ST1	ST0
		A
		A.B
		A.B.$\overline{C}$
		A.B.$\overline{C}$
		D
		D+$\overline{E}$
		D+$\overline{E}$+F
		(D+$\overline{E}$+F)·G
		(D+$\overline{E}$+F)·G

4.1.2 RD. NOT

(1) Format



(2) Inverts the status of a signal at a specified address and set it in ST0.

(3) Is used when beginning coding with contact B (-|/|-). See the ladder diagram of Fig. 4.1.2 and entries in the coding sheet of Table 4.1.2 for an example of using the RD.NOT instruction.

(4) The signal read by the RD.NOT instruction may be any contact B entered as the logical condition of one coil.

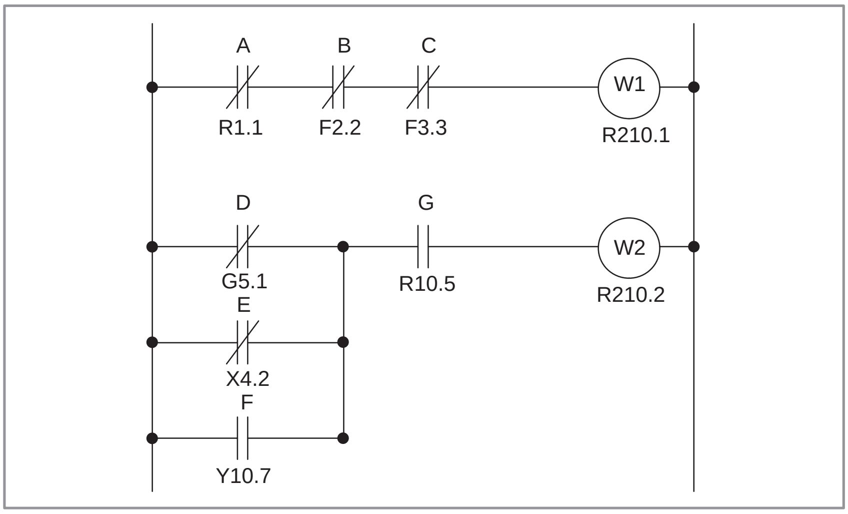


Fig. 4.1.2 Ladder diagram

Table 4.1.2 Coding for Fig. 4.1.2

Coding sheet

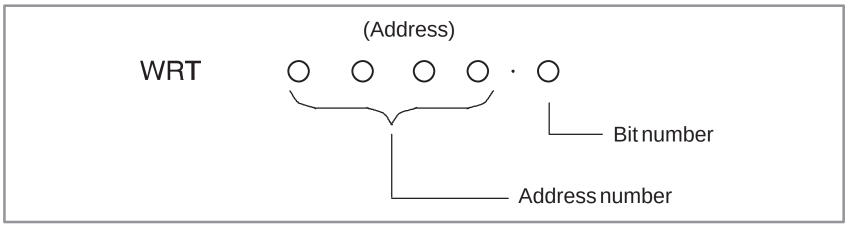
Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD. NOT	R1 .	1	A
2	AND . NOT	F2 .	2	B
3	AND . NOT	F3 .	3	C
4	WRT	R210 .	1	W1 output
5	RD. NOT	G5 .	1	D
6	OR . NOT	X4 .	2	E
7	OR	Y10 .	7	F
8	AND	R10 .	5	G
9	WRT	R210 .	2	W2 output

Status of operating result

ST2	ST1	ST0
		$\overline{A}$
		$\overline{A}.\overline{B}$
		$\overline{A}\cdot\overline{B}\cdot\overline{C}$
		$\overline{A}\cdot\overline{B}\cdot\overline{C}$
		$\overline{D}$
		$\overline{D}$+E
		$\overline{D}+\overline{E}$+F
		$(\overline{D}+\overline{E}+F)\cdot G$
		$(\overline{D}+\overline{E}+F)\cdot G$

4.1.3 WRT

(1) Format



(2) Outputs the results of logical operations, that is, the status of ST0 to a specified address.

(3) The results of one logical operation can also be output to two or more addresses. How to use the WRT instruction in this case is shown in Fig. 4.1.3 and Table 4.1.3.

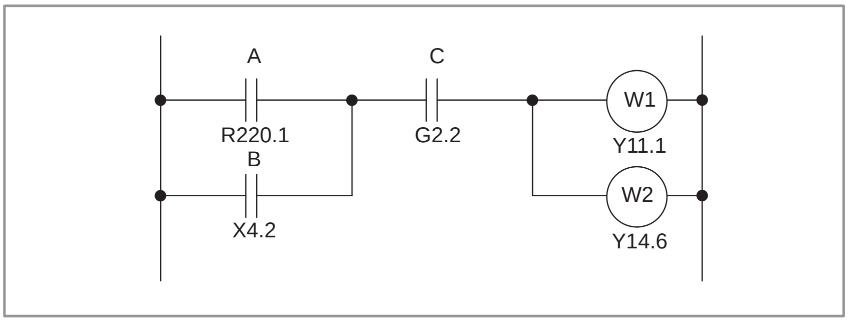


Fig. 4.1.3 Ladder diagram

Table 4.1.3 Coding for Fig. 4.1.3

Coding sheet

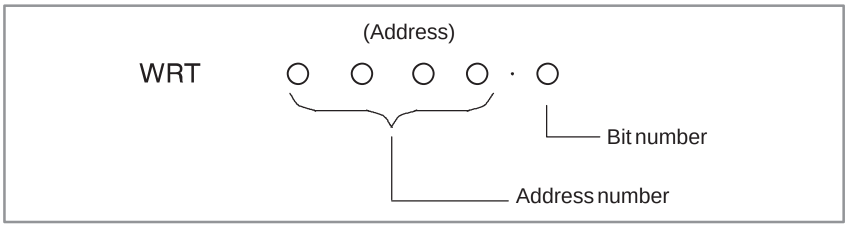
Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	R220 .	1	A
2	OR	X4 .	2	B
3	AND	G2 .	2	C
4	WRT	Y11 .	1	W1 output
5	WRT	Y14 .	6	W2 output

Status of operating result

ST2	ST1	ST0
		A
		A+B
		(A+B)·C
		(A+B)·C
		(A+B)·C

4.1.4 WRT. NOT

(1) Format



(2) Inverts the results of logical operations, that is, the status of ST0 and outputs it to a specified address. Fig. 4.1.4 and Table 4.1.4 show an example on using the WRT.NOT instruction.

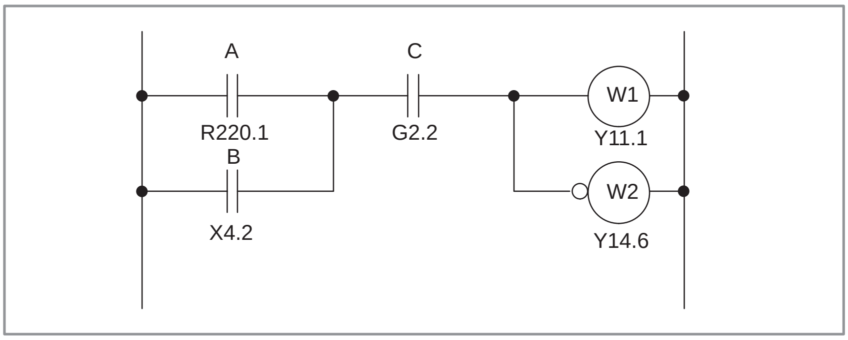


Fig. 4.1.4 Ladder diagram

Table 4.1.4 Coding for Fig. 4.1.4

Coding sheet

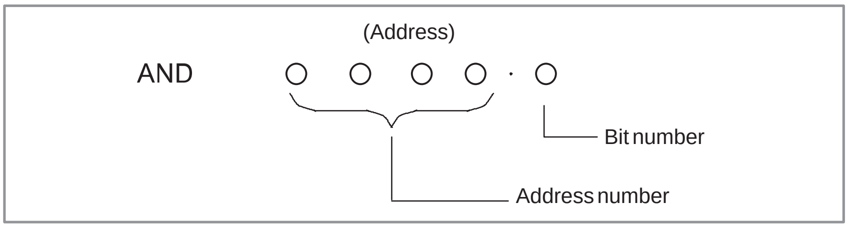
Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	R220 .	1	A
2	OR	X4 .	2	B
3	AND	G2 .	2	C
4	WRT	Y11 .	1	W1 output
5	WRT. NOT	Y14 .	6	W2 output

Status of operating result

ST2	ST1	ST0
		A
		A+B
		(A+B) · C
		(D+E) · F
		(A+B) · C

4.1.5 AND

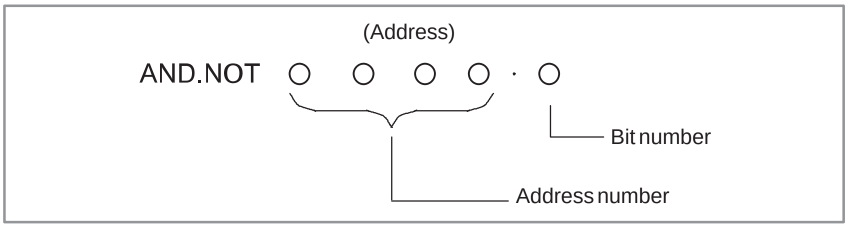
(1) Format



(2) Induces a logical product.
(3) See Fig. 4.1.1 and Table 4.1.1 for an example of using the AND instruction.

4.1.6 AND. NOT

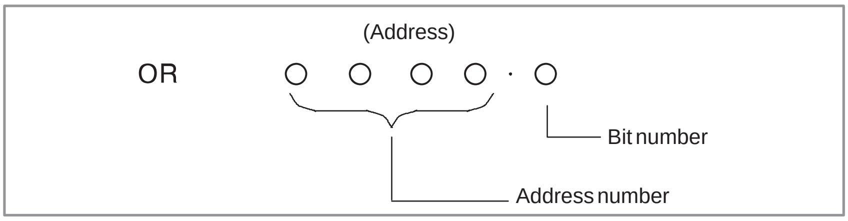
(1) Format



(2) Inverts the status of a signal at a specified address and induces a logical product.
(3) See Fig. 4.1.1 and Table 4.1.1 for an example of using the AND.NOT instruction.

4.1.7 OR

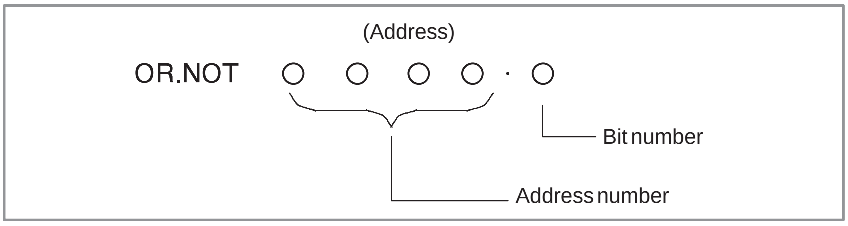
(1) Format



(2) Induces a logical sum.
(3) See Fig. 4.1.1 and Table 4.1.1 for an example of using the OR instruction.

4.1.8 OR. NOT

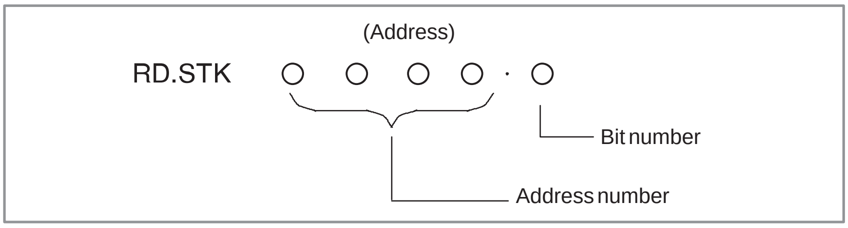
(1) Format



(2) Inverts the status of a signal at a specified address and induces a logical sum.
(3) See Fig. 4.1.1 and Table 4.1.1 for an example of using the OR.NOT instruction.

4.1.9 RD. STK

(1) Format



(2) Stacks the intermediate results of a logical operations. After shifting the stack register left one bit, sets a signal at a specified address to ST0.

(3) Is used when the signal to be specified is contact A (⊣⊢).

(4) See Fig. 4.1.9 and Table 4.1.9 for an example of using the RD.STK instruction.

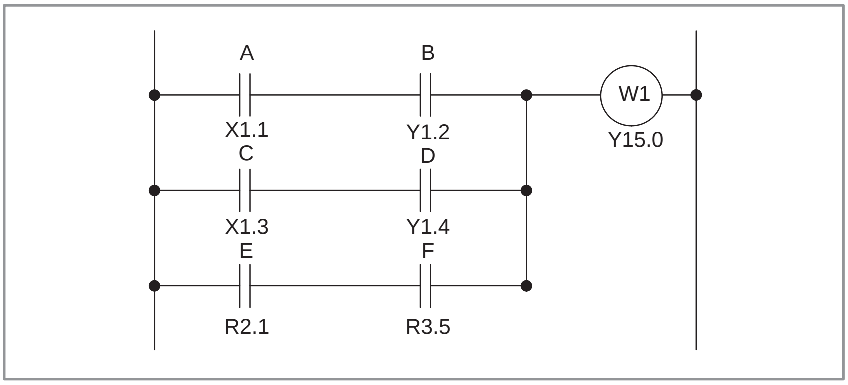


Fig. 4.1.9 Ladder diagram

Table 4.1.9 Coding for Fig. 4.1.9

Coding sheet

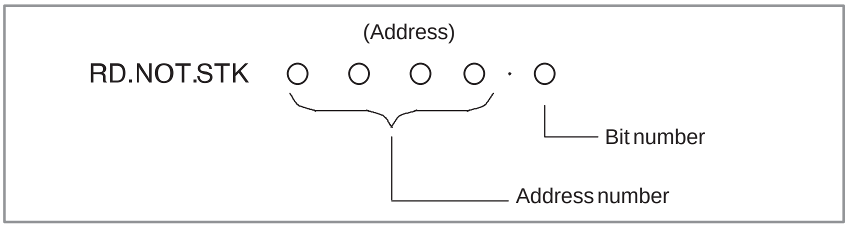
Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	X1 .	1	A
2	AND	Y1 .	2	B
3	RD. STK	X1 .	3	C
4	AND	Y1 .	4	D
5	OR. STK			
6	RD. STK	R2 .	1	E
7	AND	R3 .	5	F
8	OR.STK			
9	WRT	Y15 .	0	W1 output
10				

Status of operating result

ST2	ST1	ST0
		A
		A·B
	A·B	C
	A·B	C·D
		A·B+C·D
	A·B+C·D	E
	A·B+C·D	E·F
		A·B+C·D+E·F
		A·B+C·D+E·F

4.1.10 RD. NOT. STK

(1) Format



(2) Stacks the intermediate results of a logical operations. Shifts the stack register left one bit, inverts the status of a signal at a specified address and sets it in ST0.

(3) Is used when the signal to be specified is contact B (–|/|–).

(4) See Fig. 4.1.10 and Table 4.1.10 for an example of using the RD.NOT.STK instruction.

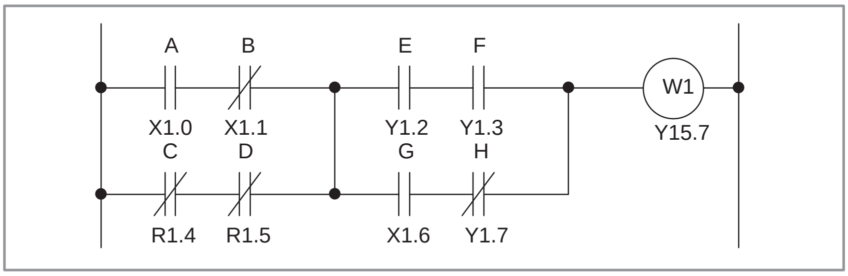


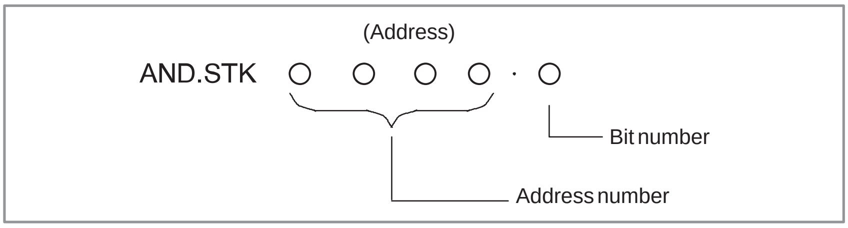
Fig. 4.1.10 Ladder diagram

Table 4.1.10 Coding for Fig. 4.1.10

Coding sheet					Status of operating result		
Step Number	**Instruction**	**Address No.**	**Bit No.**	**Remarks**	**ST2**	**ST1**	**ST0**
1	RD	X1 .	0	A			A
2	AND. NOT	X1 .	1	B			$A \cdot \overline{B}$
3	RD.NOT.STK	R1 .	4	C		$A \cdot \overline{B}$	$\overline{C}$
4	AND. NOT	R1 .	5	D		$A \cdot \overline{B}$	$\overline{C} \cdot \overline{D}$
5	OR. STK						$A \cdot \overline{B}+\overline{C} \cdot \overline{D}$
6	RD. STK	Y1 .	2	E		$A \cdot \overline{B}+\overline{C} \cdot \overline{D}$	E
7	AND	Y1 .	3	F		$A \cdot \overline{B}+\overline{C} \cdot \overline{D}$	$E \cdot F$
8	RD.STK	X1 .	6	G	$A \cdot \overline{B}+\overline{C} \cdot \overline{D}$	$E \cdot F$	G
9	AND. NOT	Y1 .	7	H	$A \cdot \overline{B}+\overline{C} \cdot \overline{D}$	$E \cdot F$	$G \cdot \overline{H}$
10	OR. STK					$A \cdot \overline{B}+\overline{C} \cdot \overline{D}$	$E \cdot F+G \cdot \overline{H}$
11	AND. STK						$(A \cdot \overline{B}+\overline{C} \cdot D) \cdot (E \cdot F+G \cdot \overline{H})$
12	WRT	Y15 .	7	W1 output			$(A \cdot \overline{B}+\overline{C} \cdot D) \cdot (E \cdot F+G \cdot \overline{H})$
13							
14							

4.1.11 AND. STK

(1) Format

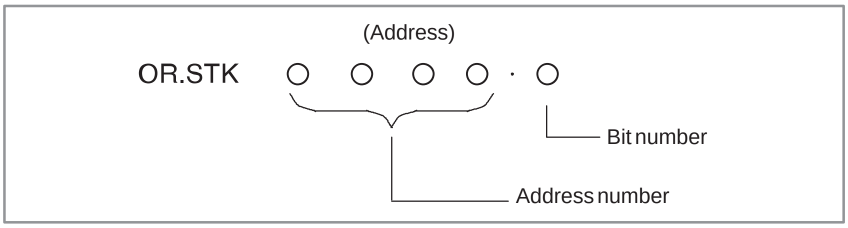


(2) Induces a logical product from the operation results in ST0 and ST1, sets the result in ST1, and shifts the stack register right one bit.

(3) See Fig. 4.1.10 and Table 4.1.10 for an example of using the AND.STK instruction.

4.1.12 OR. STK

(1) Format



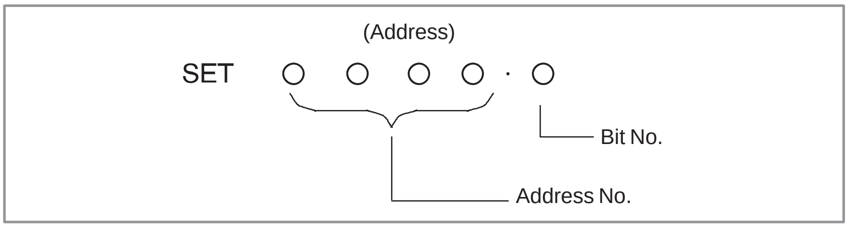
(2) Induces a logical sum from the operation results in ST0 and in ST1, sets the result in ST1, and shifts the stack register right one bit.

(3) See Fig. 4.1.9 and Table 4.1.9 or Fig. 4.1.10 and Table 4.1.10 for examples of using the OR.STK instruction.

NOTE
In Table 4.1.9 putting OR.STK at step 5 between steps 7 and 8 brings about the same result. But it is recommended to code as shown in Table 4.1.9, because coding OR.STK or AND.STK in succession is prone to cause an error.

4.1.13 SET

(1) Format



(2) Logical sum of the logical operation result ST0 with the content of the specified address is outputted to the same address.

(3) Refer to the figure below for an example of using the SET instruction.

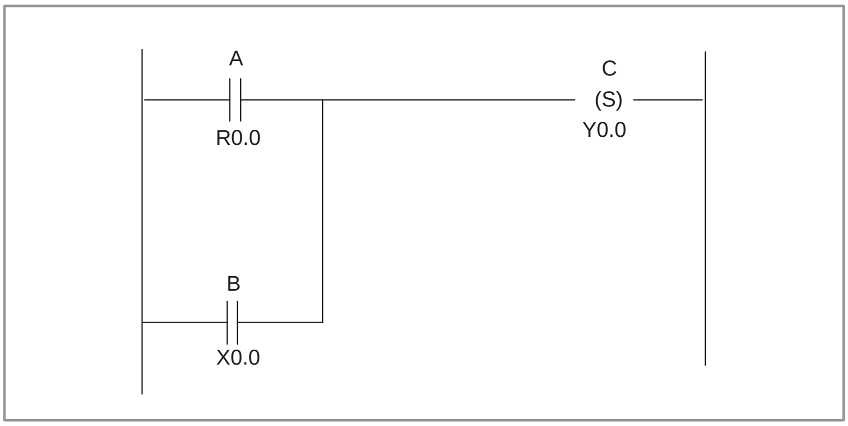


Fig. 4.1.13 Ladder diagram

Table 4.1.13 Coding for Fig. 4.1.13

Coding sheet

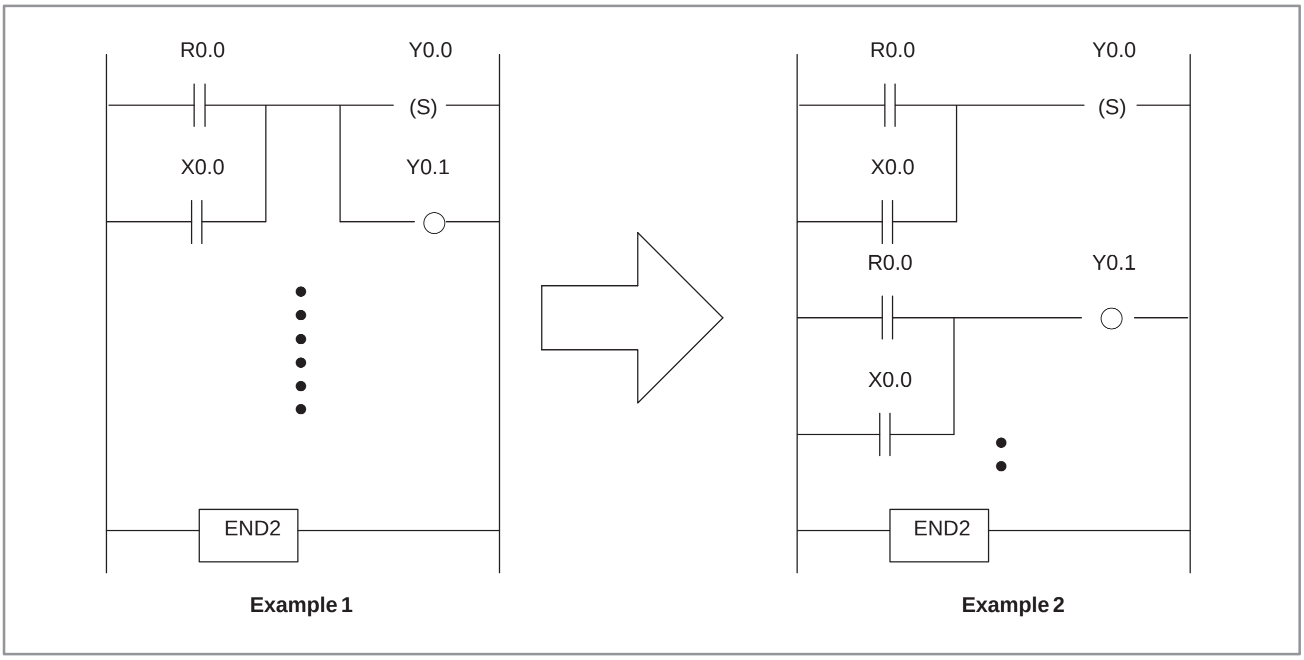
Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	R0 .	0	A
2	OR	X0 .	0	B
3	SET	Y0 .	0	Y0.0 output

Status of operating result

ST2	ST1	ST0
	A	C
	A+B	C
–	–	(A+B) +C

(4) Remarks

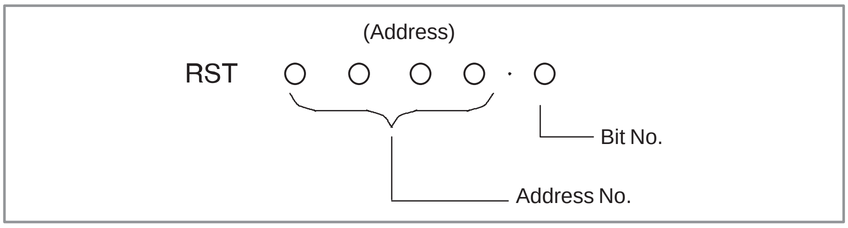
(a) Restriction of using
Do not use SET/RST like the following example 1, use them alone like the following example 2.



- The relation between COM and COME.
  The operation of SET/RST in the section of COM/COME is as follows.
  COM condition ON (ACT=1) : It operates usually.
  COM condition OFF (ACT=0) : SET does not operate.

4.1.14 RST

(1) Format



(2) Logical product of inverted logical operation result ST0 with the content of the specified address is outputted to the same address.

(3) Refer to the figure below for an example of using the RST instruction.

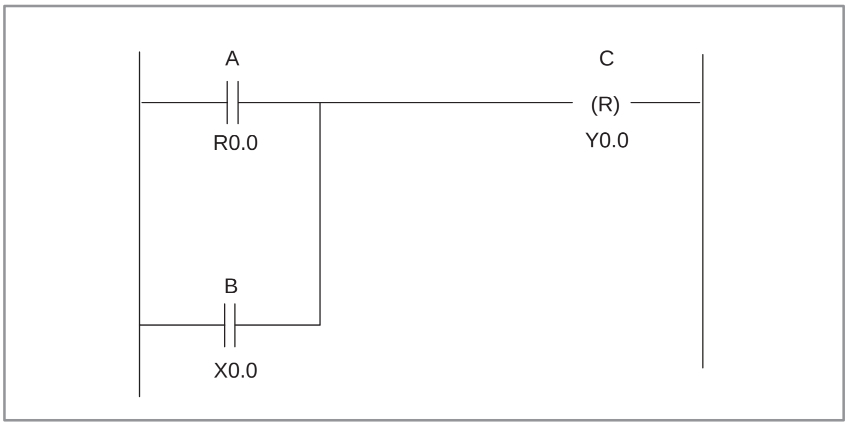


Fig. 4.1.14 Ladder diagram

Table 4.1.14 Coding for Fig. 4.1.14

Coding sheet

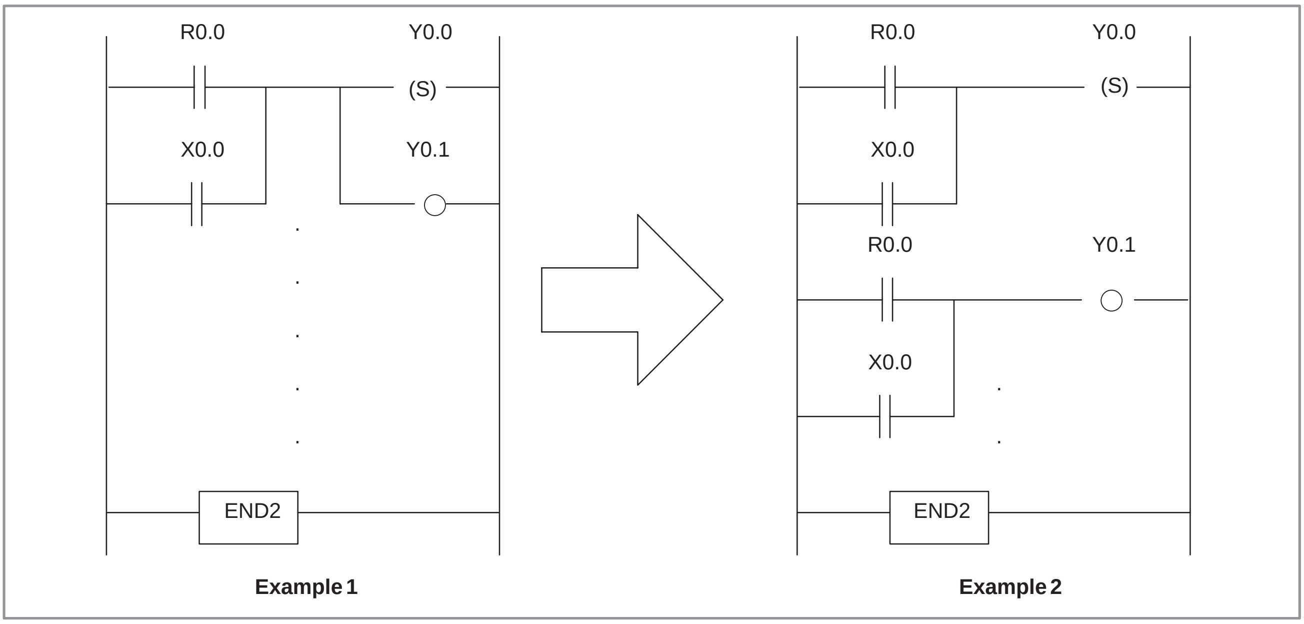
Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	R0 .	0	A
2	OR	X0 .	0	B
3	SET	Y0 .	0	Y0.0 output

Status of operating result

ST2	ST1	ST0
	A	C
	A+B	C
	–	$\overline{(A+B)}+C$

(4) Remarks

(a) Restriction of using
Do not use SET/RST like the following example 1, use them alone like the following example 2.



- The relation between COM and COME.

  The operation of SET/RST in the section of COM/COME is as follows.
  COM condition ON (ACT=1) : It operates usually.
  COM condition OFF (ACT=0) : RST does not operate.

5 FUNCTIONAL INSTRUCTIONS

In preparing a sequence program, some functions such as the function for controlling rotation via the shorter path, are difficult to program with basic instructions, which perform only one–bit logical operations. Therefore, functional instructions are available to facilitate programming. See Table 5 (a).

Table 5 (a) Types and processing of functional instructions (1)

Instruction			Processing	Model	
Format 1 (Ladder)	Format 2 (paper tape punch program)	Format 3 (program input)		PMC–PA1	PMC–PA3
END1	SUB1	S1	End of a first–level ladder program	○	○
END2	SUB2	S2	End of a second–level ladder program	○	○
END3	SUB48	S48	End of a third–level ladder program	×	×
TMR	TMR	S3 or TMR	Timer processing	○	○
TMRB	SUB24	S24	Fixed timer processing	○	○
TMRC	SUB54	S54	Timer processing	○	○
DEC	DEC	S4 or DEC	Decoding	○	○
DECB	SUB25	S25	Binary decoding	○	○
CTR	SUB5	S5	Counter processing	○	○
CTRC	SUB55	S55	Counter processing	○	○
ROT	SUB6	S6	Rotation control	○	○
ROTB	SUB26	S26	Binary rotation control	○	○
COD	SUB7	S7	Code conversion	○	○
CODB	SUB27	S27	Binary code conversion	○	○
MOVE	SUB8	S8	Data transfer after logical AND	○	○
MOVOR	SUB28	S28	Data transfer after logical OR	○	○
MOVB	SUB43	S43	Transfer of 1 byte	×	○
MOVW	SUB44	S44	Transfer of 2 bytes	×	○
MOVN	SUB45	S45	Transfer of an arbitrary number of bytes	×	○
COM	SUB9	S9	Common line control	○	○

× : Cannot be used ○ : Can be used

Table 5 (a) Types and processing of functional instructions (2)

Instruction			Processing	Model	
Format 1 (Ladder)	Format 2 (paper tape punch program)	Format 3 (program input)		PMC–PA1	PMC–PA3
COME	SUB29	S29	End of common line control	○	○
JMP	SUB10	S10	Jump	○	○
JMPE	SUB30	S30	End of a jump	○	○
JMPB	SUB68	S68	Label jump 1	×	○
JMPC	SUB73	S73	Label jump 2	×	○
LBL	SUB69	S69	Label	×	○
PARI	SUB11	S11	Parity check	○	○
DCNV	SUB14	S14	Data conversion	○	○
DCNVB	SUB31	S31	Extended data conversion	○	○
COMP	SUB15	S15	Comparison	○	○
COMPB	SUB32	S32	Binary comparison	○	○
COIN	SUB16	S16	Coincidence check	○	○
SFT	SUB33	S33	Shift register	○	○
DSCH	SUB17	S17	Data search	○	○
DSCHB	SUB34	S34	Binary data search	○	○
XMOV	SUB18	S18	Indexed data transfer	○	○
XMOVB	SUB35	S35	Binary indexed data transfer	○	○
ADD	SUB19	S19	Addition	○	○
ADDB	SUB36	S36	Binary addition	○	○
SUB	SUB20	S20	Subtraction	○	○
SUBB	SUB37	S37	Binary subtraction	○	○
MUL	SUB21	S21	Multiplication	○	○
MULB	SUB38	S38	Binary multiplication	○	○
DIV	SUB22	S22	Division	○	○
DIVB	SUB39	S39	Binary division	○	○
NUME	SUB23	S23	Constant definition	○	○
NUMEB	SUB40	S40	Binary constant definition	○	○
DISP	SUB49	S49	Message display	×	×
DISPB	SUB41	S41	Extended message display	○	○
EXIN	SUB42	S42	External data input	○	○
WINDR	SUB51	S51	Window data read	○	○
WINDW	SUB52	S52	Window data write	○	○

× : Cannot be used ○ : Can be used

Table 5 (a) Types and processing of functional instructions (3)

Instruction			Processing	Model	
Format 1 (Ladder)	Format 2 (paper tape punch program)	Format 3 (program input)		PMC–PA1	PMC–PA3
PSGNL	SUB50	S50	Position signal output	○	○
PSGN2	SUB63	S63	Position signal output 2	○	○
DIFU	SUB57	S57	Rising edge detection	×	○
DIFD	SUB58	S58	Falling edge detection	×	○
EOR	SUB59	S59	Exclusive OR	×	○
AND	SUB60	S60	Logical AND	×	○
OR	SUB61	S61	Logical OR	×	○
NOT	SUB62	S62	Logical NOT	×	○
END	SUB64	S64	End of a subprogram	×	○
CALL	SUB65	S65	Conditional subprogram call	×	○
CALLU	SUB66	S66	Unconditional subprogram call	×	○
SP	SUB71	S71	Subprogram	×	○
SPE	SUB72	S72	End of a subprogram	×	○
AXCTL	SUB53	S53	PMC axes control	○	○

× : Cannot be used ○ : Can be used

Table 5 (a) Types and processing of functional instructions (4)

Instruction	SUB number	Processing	Model									
			PMC–SA1	PMC–SA2	PMC–SA3	PMC–SB	PMC–SB2	PMC–SB3	PMC–SC	PMC–SC3	PMC–NB/NB2	PMC–NB6
END1	1	End of a first–level ladder program	○	○	○	○	○	○	○	○	○	○
END2	2	End of a second–level ladder program	○	○	○	○	○	○	○	○	○	○
END3	48	End of a third–level ladder program	×	×	×	×	×	×	○	○	○	○
TMR	3	Timer processing	○	○	○	○	○	○	○	○	○	○
TMRB	24	Fixed timer processing	○	○	○	○	○	○	○	○	○	○
TMRC	54	Timer processing	○	○	○	○	○	○	○	○	○	○
DEC	4	Decoding	○	○	○	○	○	○	○	○	○	○
DECB	25	Binary decoding	○	○	○	○	○	○	○	○	○	○
CTR	5	Counter processing	○	○	○	○	○	○	○	○	○	○
CTRC	55	Counter processing	○	○	○	○	○	○	○	○	○	○
ROT	6	Rotation control	○	○	○	○	○	○	○	○	○	○
ROTB	26	Binary rotation control	○	○	○	○	○	○	○	○	○	○
COD	7	Code conversion	○	○	○	○	○	○	○	○	○	○
CODB	27	Binary code conversion	○	○	○	○	○	○	○	○	○	○
MOVE	8	Data transfer after Logical AND	○	○	○	○	○	○	○	○	○	○
MOVOR	28	Data transfer after logical OR	○	○	○	○	○	○	○	○	○	○
MOVB	43	Transfer of 1 byte	×	×	○	×	×	○	×	○	○	○
MOVW	44	Transfer of 2 bytes	×	×	○	×	×	○	×	○	○	○
MOVN	45	Transfer of an arbitrary number of bytes	×	×	○	×	×	○	×	○	○	○
COM	9	Common line control	○	○	○	○	○	○	○	○	○	○
COME	29	End of common line control	○	○	○	○	○	○	○	○	○	○
JMP	10	Jump	○	○	○	○	○	○	○	○	○	○
JMPE	30	End of a jump	○	○	○	○	○	○	○	○	○	○
JMPB	68	Label jump 1	×	×	○	×	×	○	×	○	○	○
JMPC	73	Label jump 2	×	×	○	×	×	○	×	○	○	○
LBL	69	Label	×	×	○	×	×	○	×	○	○	○
PARI	11	Parity check	○	○	○	○	○	○	○	○	○	○
DCNV	14	Data conversion	○	○	○	○	○	○	○	○	○	○
DCNVB	31	Binary data conversion	○	○	○	○	○	○	○	○	○	○
COMP	15	Comparison	○	○	○	○	○	○	○	○	○	○
COMPB	32	Binary comparison	○	○	○	○	○	○	○	○	○	○
COIN	16	Coincidence check	○	○	○	○	○	○	○	○	○	○
SFT	33	Shift register	○	○	○	○	○	○	○	○	○	○
DSCH	17	Data search	○	○	○	○	○	○	○	○	○	○
DSCHB	34	Binary data search	○	○	○	○	○	○	○	○	○	○

× : Cannot be used   ○ : Can be used

Table 5 (a) Types and processing of functional instructions (5)

Instruc-tion	SUB number	Processing	Model									
			PMC–SA1	PMC–SA2	PMC–SA3	PMC–SB	PMC–SB2	PMC–SB3	PMC–SC	PMC–SC3	PMC–NB/NB2	PMC–NB6
XMOV	18	Indexed data transfer	○	○	○	○	○	○	○	○	○	○
XMOVB	35	Binary indexed data transfer	○	○	○	○	○	○	○	○	○	○
ADD	19	Addition	○	○	○	○	○	○	○	○	○	○
ADDB	36	Binary addition	○	○	○	○	○	○	○	○	○	○
SUB	20	Subtraction	○	○	○	○	○	○	○	○	○	○
SUBB	37	Binary subtraction	○	○	○	○	○	○	○	○	○	○
MUL	21	Multiplication	○	○	○	○	○	○	○	○	○	○
MULB	38	Binary multiplication	○	○	○	○	○	○	○	○	○	○
DIV	22	Division	○	○	○	○	○	○	○	○	○	○
DIVB	39	Binary division	○	○	○	○	○	○	○	○	○	○
NUME	23	Constant definition	○	○	○	○	○	○	○	○	○	○
NUMEB	40	Binary constant definition	○	○	○	○	○	○	○	○	○	○
DISP	49	Message display	×	×	×	○	○	○	○	○	×	×
DISPB	41	Extended message display	○	○	○	○	○	○	○	○	○	○
EXIN	42	External data input	○	○	○	○	○	○	○	○	○	○
SPCNT	46	Spindle control	×	×	×	×	×	×	×	×	○	○
WINDR	51	NC window data read	○	○	○	○	○	○	○	○	○	○
WINDW	52	NC window data write	○	○	○	○	○	○	○	○	○	○
FNC9X	9X	Arbitrary functional instruction (X = 0 to 7)	×	×	×	×	×	×	×	○	○	×
MMC3R	88	MMC3 window data read	○	○	○	○	○	○	○	○	○	×
MMC3W	89	MMC3 window data write	○	○	○	○	○	○	○	○	○	×
MMCWR	98	MMC window data read	○	○	○	○	○	○	○	○	○	○
MMCWW	99	MMC window data write	○	○	○	○	○	○	○	○	○	○
DIFU	57	Rising edge detection	×	×	○	×	×	○	×	○	○	○
DIFD	58	Falling edge detection	×	×	○	×	×	○	×	○	○	○
EOR	59	Exclusive OR	×	×	○	×	×	○	×	○	○	○
AND	60	Logical AND	×	×	○	×	×	○	×	○	○	○
OR	61	Logical OR	×	×	○	×	×	○	×	○	○	○
NOT	62	Logical NOT	×	×	○	×	×	○	×	○	○	○
END	64	End of a subprogram	×	×	○	×	×	○	×	○	○	○
CALL	65	Conditional subprogram call	×	×	○	×	×	○	×	○	○	○
CALLU	66	Unconditional subprogram call	×	×	○	×	×	○	×	○	○	○
SP	71	Subprogram	×	×	○	×	×	○	×	○	○	○
SPE	72	End of a subprogram	×	×	○	×	×	○	×	○	○	○
AXCTL	53	PMC axes control	○	○	○	○	○	○	○	○	×	×

× : Cannot be used ○ : Can be used

Table 5 (a) Types and processing of functional instructions (6)

Name	SUB number	Processing	Model				
			Series 16/18–MODEL B		Series 16/18–MODEL B/C		Series 18–MODEL B
			PMC–SB3	PMC–SB4	PMC–SC3	PMC–SC4	PMC–SA1
END1	1	First level program end	○	○	○	○	○
END2	2	Second level program end	○	○	○	○	○
END3	48	Third level program end	×	×	○	○	×
TMR	3	Timer processing	○	○	○	○	○
TMRB	24	Fixed timer processing	○	○	○	○	○
TMRC	54	Timer processing	○	○	○	○	○
DEC	4	Decoding	○	○	○	○	○
DECB	25	Binary decoding	○	○	○	○	○
CTR	5	Counter processing	○	○	○	○	○
CTRC	55	Counter processing	○	○	○	○	○
ROT	6	Rotation control	○	○	○	○	○
ROTB	26	Binary rotation control	○	○	○	○	○
COD	7	Code conversion	○	○	○	○	○
CODB	27	Binary code conversion	○	○	○	○	○
MOVE	8	ANDed data transfer	○	○	○	○	○
MOVOR	28	ORed data transfer	○	○	○	○	○
MOVB	43	Byte data transfer	○	○	○	○	×
MOVW	44	Word data transfer	○	○	○	○	×
MOVN	45	Block data transfer	○	○	○	○	×
COM	9	Common line control	○	○	○	○	○
COME	29	Common line control end	○	○	○	○	○
JMP	10	Jump	○	○	○	○	○
JMPE	30	Jump end	○	○	○	○	○
JMPB	68	Label jump 1	○	○	○	○	×
JMPC	73	Label jump 2	○	○	○	○	×
LBL	69	Label	○	○	○	○	×
PARI	11	Parity check	○	○	○	○	○
DCNV	14	Data conversion	○	○	○	○	○
DCNVB	31	Extended data conversion	○	○	○	○	○
COMP	15	Comparison	○	○	○	○	○
COMPB	32	Binary comparison	○	○	○	○	○
COIN	16	Coincidence check	○	○	○	○	○
SFT	33	Shift register	○	○	○	○	○
DSCH	17	Data search	○	○	○	○	○
DSCHB	34	Binary data search	○	○	○	○	○
XMOV	18	Indexed data transfer	○	○	○	○	○
XMOVB	35	Binary indexed data transfer	○	○	○	○	○
ADD	19	Addition	○	○	○	○	○

× : Cannot be used ○ : Can be used

Table 5 (a) Types and processing of functional instructions (7)

Name	SUB number	Processing	Model				
			Series 16/18–MODEL B		Series 16/18–MODEL B/C		Series 18–MODEL B
			PMC–SB3	PMC–SB4	PMC–SC3	PMC–SC4	PMC–SA1
ADDB	36	Binary Addition	○	○	○	○	○
SUB	20	Subtraction	○	○	○	○	○
SUBB	37	Binary subtraction	○	○	○	○	○
MUL	21	Multiplication	○	○	○	○	○
MULB	38	Binary multiplication	○	○	○	○	○
DIV	22	Division	○	○	○	○	○
DIVB	39	Binary division	○	○	○	○	○
NUME	23	Definition of constant	○	○	○	○	○
NUMEB	40	Definition of binary constant	○	○	○	○	○
DISP	49	Message display	△	△	△	△	×
DISPB	41	Extended message display	○	○	○	○	○
EXIN	42	External data input	○	○	○	○	○
AXCTL	53	PMC axis control	○	○	○	○	○
WINDR	51	Window data read	○	○	○	○	○
WINDW	52	Window data write	○	○	○	○	○
FNC9X	9X	Arbitrary functional ins.	×	×	○	○	×
MMC3R	88	MMC3 window data read	○	○	○	○	○
MMC3W	89	MMC3 window data write	○	○	○	○	○
MMCWR	98	MMC2 window data read	○	○	○	○	○
MMCWW	99	MMC2 window data write	○	○	○	○	○
DIFU	57	Rising edge detection	○	○	○	○	×
DIFD	58	Falling edge detection	○	○	○	○	×
EOR	59	Exclusive OR	○	○	○	○	×
AND	60	Logical production	○	○	○	○	×
OR	61	Logical Add	○	○	○	○	×
NOT	62	Logical Negation	○	○	○	○	×
END	64	End of subprograms	○	○	○	○	×
CALL	65	Conditional subprogram call	○	○	○	○	×
CALLU	66	Unconditional subprogram call	○	○	○	○	×
SP	71	Subprogram	○	○	○	○	×
SPE	72	End of a subprogram	○	○	○	○	×

× : Cannot be used ○ : Can be used △ : Can be used (with some restrictions)

NOTE

On the PMC–SB3/SB4/SC3/SC4, DISP is provided only for the compatibility with Series 16/18 MODEL A. On the Series 16/18 MODEL B, it is recommended to use DISPB instead of DISP because some extended functions such as high speed display and display of double sized character are available only with DISPB. On the Series 16/18 MODEL B, if both DISP and DISPB are used in the same sequence program, double sized character can not be displayed by DISPB.

Table 5 (a) Types and processing of functional instructions (8)

Name	SUB number	Processing	Model	
			Series 16–MODEL C/Series 18–MODEL C	
			PMC–SB5	PMC–SB6
END1	1	First level program end	○	○
END2	2	Second level program end	○	○
END3	48	Third level program end	×	×
TMR	3	Timer processing	○	○
TMRB	24	Fixed timer processing	○	○
TMRC	54	Timer processing	○	○
DEC	4	Decoding	○	○
DECB	25	Binary decoding	○	○
CTR	5	Counter processing	○	○
CTRC	55	Counter processing	○	○
ROT	6	Rotation control	○	○
ROTB	26	Binary rotation control	○	○
COD	7	Code conversion	○	○
CODB	27	Binary code conversion	○	○
MOVE	8	ANDed data transfer	○	○
MOVOR	28	ORed data transfer	○	○
MOVB	43	Byte data transfer	○	○
MOVW	44	Word data transfer	○	○
MOVN	45	Block data transfer	○	○
COM	9	Common line control	○	○
COME	29	Common line control end	○	○
JMP	10	Jump	○	○
JMPE	30	Jump end	○	○
JMPB	68	Label jump 1	○	○
JMPC	73	Label jump 2	○	○
LBL	69	Label	○	○
PARI	11	Parity check	○	○
DCNV	14	Data conversion	○	○
DCNVB	31	Extended data conversion	○	○
COMP	15	Comparison	○	○
COMPB	32	Binary comparison	○	○
COIN	16	Coincidence check	○	○
SFT	33	Shift register	○	○
DSCH	17	Data search	○	○
DSCHB	34	Binary data search	○	○
XMOV	18	Indexed data transfer	○	○
XMOVB	35	Binary indexed data transfer	○	○
ADD	19	Addition	○	○

× : Cannot be used ○ : Can be used

Table 5 (a) Types and processing of functional instructions (9)

Name	SUB number	Processing	Model	
			Series 16–MODEL C/Series 18–MODEL C	
			PMC–SB5	PMC–SB6
ADDB	36	Binary Addition	○	○
SUB	20	Subtraction	○	○
SUBB	37	Binary subtraction	○	○
MUL	21	Multiplication	○	○
MULB	38	Binary multiplication	○	○
DIV	22	Division	○	○
DIVB	39	Binary division	○	○
NUME	23	Definition of constant	○	○
NUMEB	40	Definition of binary constant	○	○
DISP	49	Message display (Note)	△	△
DISPB	41	Extended message display	○	○
EXIN	42	External data input	○	○
AXCTL	53	PMC axis control	○	○
WINDR	51	Window data read	○	○
WINDW	52	Window data write	○	○
FNC9X	9X	Arbitrary functional ins.	×	×
MMC3R	88	MMC3 window data read	○	○
MMC3W	89	MMC3 window data write	○	○
MMCWR	98	MMC2 window data read	○	○
MMCWW	99	MMC2 window data write	○	○
DIFU	57	Rising edge detection	○	○
DIFD	58	Falling edge detection	○	○
EOR	59	Exclusive OR	○	○
AND	60	Logical production	○	○
OR	61	Logical Add	○	○
NOT	62	Logical Negation	○	○
END	64	End of subprograms	○	○
CALL	65	Conditional subprogram call	○	○
CALLU	66	Unconditional subprogram call	○	○
SP	71	Subprogram	○	○
SPE	72	End of a subprogram	○	○

× : Cannot be used ○ : Can be used △ : Can be used (with some restrictions)

NOTE

On the PMC–SB5/SB6, DISP is provided only for the compatibility with Series 16 MODEL A/B. On the Series 16/18 MODEL C, it is recommended to use DISPB instead of DISP because some extended functions such as high speed display and display of double sized character are available only with DISPB. On the Series 16/18 MODEL C, if both DISP and DISPB are used in the same sequence program, double sized character can not be displayed by DISPB.

Table 5 (a) Types and processing of functional instructions (10)

Name	SUB number	Processing	Model	
			Series 21–MODEL B	
			PMC–SA1	PMC–SA3
END1	1	First level program end	○	○
END2	2	Second level program end	○	○
END3	48	Third level program end	×	×
TMR	3	Timer processing	○	○
TMRB	24	Fixed timer processing	○	○
TMRC	54	Timer processing	○	○
DEC	4	Decoding	○	○
DECB	25	Binary decoding	○	○
CTR	5	Counter processing	○	○
CTRC	55	Counter processing	○	○
ROT	6	Rotation control	○	○
ROTB	26	Binary rotation control	○	○
COD	7	Code conversion	○	○
CODB	27	Binary code conversion	○	○
MOVE	8	ANDed data transfer	○	○
MOVOR	28	ORed data transfer	○	○
MOVB	43	Byte data transfer	×	○
MOVW	44	Word data transfer	×	○
MOVN	45	Block data transfer	×	○
COM	9	Common line control	○	○
COME	29	Common line control end	○	○
JMP	10	Jump	○	○
JMPE	30	Jump end	○	○
JMPB	68	Label jump 1	×	○
JMPC	73	Label jump 2	×	○
LBL	69	Label	×	○
PARI	11	Parity check	○	○
DCNV	14	Data conversion	○	○
DCNVB	31	Extended data conversion	○	○
COMP	15	Comparison	○	○
COMPB	32	Binary comparison	○	○
COIN	16	Coincidence check	○	○
SFT	33	Shift register	○	○
DSCH	17	Data search	○	○
DSCHB	34	Binary data search	○	○
XMOV	18	Indexed data transfer	○	○
XMOVB	35	Binary indexed data transfer	○	○
ADD	19	Addition	○	○

× : Cannot be used ○ : Can be used

Table 5 (a) Types and processing of functional instructions (11)

Name	SUB number	Processing	Model	
			Series 21–MODEL B	
			PMC–SA1	PMC–SA3
ADDB	36	Binary Addition	○	○
SUB	20	Subtraction	○	○
SUBB	37	Binary subtraction	○	○
MUL	21	Multiplication	○	○
MULB	38	Binary multiplication	○	○
DIV	22	Division	○	○
DIVB	39	Binary division	○	○
NUME	23	Definition of constant	○	○
NUMEB	40	Definition of binary constant	○	○
DISP	49	Message display	×	×
DISPB	41	Extended message display	○	○
EXIN	42	External data input	○	○
AXCTL	53	PMC axis control	○	○
WINDR	51	Window data read	○	○
WINDW	52	Window data write	○	○
FNC9X	9X	Arbitrary functional ins.	×	×
MMC3R	88	MMC3 window data read	○	○
MMC3W	89	MMC3 window data write	○	○
MMCWR	98	MMC2 window data read	○	○
MMCWW	99	MMC2 window data write	○	○
DIFU	57	Rising edge detection	×	○
DIFD	58	Falling edge detection	×	○
EOR	59	Exclusive OR	×	○
AND	60	Logical production	×	○
OR	61	Logical Add	×	○
NOT	62	Logical Negation	×	○
END	64	End of subprograms	×	○
CALL	65	Conditional subprogram call	×	○
CALLU	66	Unconditional subprogram call	×	○
SP	71	Subprogram	×	○
SPE	72	End of a subprogram	×	○

× : Cannot be used ○ : Can be used

Table 5 (a) Types and Processing of Functional Instructions (6) Part 1

Name	SUB number	Processing	Model	
			Series 16*i* MODEL A/Series 18*i* MODEL A	
			PMC–SB5	PMC–SB6
END1	1	First level program end	○	○
END2	2	Second level program end	○	○
END3	48	Third level program end	×	×
TMR	3	Timer processing	○	○
TMRB	24	Fixed timer processing	○	○
TMRC	54	Timer processing	○	○
DEC	4	Decoding	○	○
DECB	25	Binary decoding	○	○
CTR	5	Counter processing	○	○
CTRC	55	Counter processing	○	○
ROT	6	Rotation control	○	○
ROTB	26	Binary rotation control	○	○
COD	7	Code conversion	○	○
CODB	27	Binary code conversion	○	○
MOVE	8	ANDed data transfer	○	○
MOVOR	28	ORed data transfer	○	○
MOVB	43	Transfer of one byte	○	○
MOVW	44	Transfer of two bytes	○	○
MOVN	45	Transfer of arbitrary bytes	○	○
COM	9	Common line control	○	○
COME	29	Common line control end	○	○
JMP	10	Jump	○	○
JMPE	30	Jump end	○	○
JMPB	68	Label jump 1	○	○
JMPC	73	Label jump 2	○	○
LBL	69	Label specification	○	○
PARI	11	Parity check	○	○
DCNV	14	Data conversion	○	○
DCNVB	31	Binary data conversion	○	○
COMP	15	Comparison	○	○
COMPB	32	Binary comparison	○	○
COIN	16	Coincidence check	○	○
SFT	33	Shift register	○	○
DSCH	17	Data search	○	○
DSCHB	34	Binary data search	○	○
XMOV	18	Indexed data transfer	○	○
XMOVB	35	Binary indexed data transfer	○	○
ADD	19	BCD addition	○	○

× : Cannot be used ○ : Can be used

Table 5 (a) Types and Processing of Functional Instructions (6) Part 2

Name	SUB number	Processing	Model	
			Series 16*i* MODEL A/Series 18*i* MODEL A	
			PMC–SB5	PMC–SB6
ADDB	36	Binary addition	○	○
SUB	20	BCD subtraction	○	○
SUBB	37	Binary subtraction	○	○
MUL	21	BCD multiplication	○	○
MULB	38	Binary multiplication	○	○
DIV	22	BCD division	○	○
DIVB	39	Binary division	○	○
NUME	23	Definition of constant	○	○
NUMEB	40	Definition of binary constant	○	○
DISP	49	Message display (Note)	△	△
DISPB	41	Extended message display	○	○
EXIN	42	External data input	○	○
AXCTL	53	PMC axis control	○	○
WINDR	51	Window data read	○	○
WINDW	52	Window data write	○	○
FNC9X	9X	Arbitrary functional instruction	×	×
MMC3R	88	MMC3 window data read	×	×
MMC3W	89	MMC3 window data write	×	×
MMCWR	98	MMC2 window data read	○	○
MMCWW	99	MMC2 window data write	○	○
DIFU	57	Rising edge detection	○	○
DIFD	58	Falling edge detection	○	○
EOR	59	Exclusive OR	○	○
AND	60	Logical product	○	○
OR	61	Logical add	○	○
NOT	62	Logical negation	○	○
END	64	End of subprograms	○	○
CALL	65	Conditional subprogram call	○	○
CALLU	66	Unconditional subprogram call	○	○
SP	71	Subprogram	○	○
SPE	72	End of a subprogram	○	○

× : Cannot be used ○ : Can be used △ : Can be used (with some restrictions)

NOTE

With PMC–SB5/SB6 of the Series 16*i*/18*i* MODEL A, the DISP instruction can be used only to ensure compatibility with the Series 16 MODEL A/B.
With the Series 16*i*/18*i* MODEL A, FANUC recommends the use of the DISPB instruction that provides extended functions such as high–speed display and kanji character display.
With the Series 16*i*/18*i* MODEL A, if both the DISP instruction and DISPB instruction are used in the same sequence program, the kanji display function of the DISPB instruction cannot be used.

Table 5 (a) Types and Processing of Functional Instructions (7) Part 1

Name	SUB number	Processing	Model	
			Series 21*i* MODEL A	
			PMC–SA1	PMC–SA5
END1	1	First level program end	○	○
END2	2	Second level program end	○	○
END3	48	Third level program end	×	×
TMR	3	Timer processing	○	○
TMRB	24	Fixed timer processing	○	○
TMRC	54	Timer processing	○	○
DEC	4	Decoding	○	○
DECB	25	Binary decoding	○	○
CTR	5	Counter processing	○	○
CTRC	55	Counter processing	○	○
ROT	6	Rotation control	○	○
ROTB	26	Binary rotation control	○	○
COD	7	Code conversion	○	○
CODB	27	Binary code conversion	○	○
MOVE	8	ANDed data transfer	○	○
MOVOR	28	ORed data transfer	○	○
MOVB	43	Transfer of one byte	×	○
MOVW	44	Transfer of two bytes	×	○
MOVN	45	Transfer of arbitrary bytes	×	○
COM	9	Common line control	○	○
COME	29	Common line control end	○	○
JMP	10	Jump	○	○
JMPE	30	Jump end	○	○
JMPB	68	Label jump 1	×	○
JMPC	73	Label jump 2	×	○
LBL	69	Label specification	×	○
PARI	11	Parity check	○	○
DCNV	14	Data conversion	○	○
DCNVB	31	Binary data conversion	○	○
COMP	15	Comparison	○	○
COMPB	32	Binary comparison	○	○
COIN	16	Coincidence check	○	○
SFT	33	Shift register	○	○
DSCH	17	Data search	○	○
DSCHB	34	Binary data search	○	○
XMOV	18	Indexed data transfer	○	○
XMOVB	35	Binary indexed data transfer	○	○
ADD	19	BCD addition	○	○

× : Cannot be used ○ : Can be used

Table 5 (a) Types and Processing of Functional Instructions (7) Part 2

Name	SUB number	Processing	Model	
			Series 21*i* MODEL A	
			PMC–SA1	PMC–SA5
ADDB	36	Binary addition	○	○
SUB	20	BCD subtraction	○	○
SUBB	37	Binary subtraction	○	○
MUL	21	BCD multiplication	○	○
MULB	38	Binary multiplication	○	○
DIV	22	BCD division	○	○
DIVB	39	Binary division	○	○
NUME	23	Definition of constant	○	○
NUMEB	40	Definition of binary constant	○	○
DISP	49	Message display	×	×
DISPB	41	Extended message display	○	○
EXIN	42	External data input	○	○
AXCTL	53	PMC axis control	○	○
WINDR	51	Window data read	○	○
WINDW	52	Window data write	○	○
FNC9X	9X	Arbitrary functional instruction	×	×
MMC3R	88	MMC3 window data read	×	×
MMC3W	89	MMC3 window data write	×	×
MMCWR	98	MMC2 window data read	○	○
MMCWW	99	MMC2 window data write	○	○
DIFU	57	Rising edge detection	×	○
DIFD	58	Falling edge detection	×	○
EOR	59	Exclusive OR	×	○
AND	60	Logical product	×	○
OR	61	Logical add	×	○
NOT	62	Logical negation	×	○
END	64	End of subprograms	×	○
CALL	65	Conditional subprogram call	×	○
CALLU	66	Unconditional subprogram call	×	○
SP	71	Subprogram	×	○
SPE	72	End of a subprogram	×	○

× : Cannot be used ○ : Can be used

The execution time constant is a ratio of the execution time of a functional instruction to the execution time of 10 basic instruction steps (1.5 μs). Execution time constants are used when a ladder program is executed in the separate mode.

Table 5 (b) Execution Time Constants of Functional Instructions (1)

Instruction	SUB Number	Processing	Model	
			PMC–SB	PMC–SC
END1	1	End of a first–level ladder program	171	1033
END2	2	End of a second–level ladder program	26	45
END3	48	End of a third–level ladder program	–	0
TMR	3	Timer processing	19	33
TMRB	24	Fixed timer processing	19	34
TMRC	54	Timer processing	17	29
DEC	4	Decoding	21	28
DECB	25	Binary decoding	16	23
CTR	5	Counter processing	21	35
CTRC	55	Counter processing	18	26
ROT	6	Rotation control	37	53
ROTB	26	Binary rotation control	27	39
COD	7	Code conversion	20	29
CODB	27	Binary code conversion	19	29
MOVE	8	Data transfer after Logical AND	19	27
MOVOR	28	Data transfer after logical OR	13	19
COM	9	Common line control	11	14
COME	29	End of common line control	0.1	0.1
JMP	10	Jump	12	16
JMPE	30	End of a jump	9	11
PARI	11	Parity check	13	19
DCNV	14	Data conversion	25	37
DCNVB	31	Binary data conversion	132	233
COMP	15	Comparison	22	36
COMPB	32	Binary comparison	20	31
COIN	16	Coincidence check	21	36
SFT	33	Shift register	15	22
DSCH	17	Data search	237	287
DSCHB	34	Binary data search	351	596
XMOV	18	Indexed data transfer	26	38
XMOVB	35	Binary indexed data transfer	27	37

Table 5 (b) Execution Time Constants of Functional Instructions (2)

Instruction	SUB Number	Processing	Model	
			PMC–SB	PMC–SC
ADD	19	Addition	22	33
ADDB	36	Binary addition	25	39
SUB	20	Subtraction	21	32
SUBB	37	Binary subtraction	25	39
MUL	21	Multiplication	42	63
MULB	38	Binary multiplication	28	45
DIV	22	Division	44	66
DIVB	39	Binary division	33	53
NUME	23	Constant definition	18	25
NUMEB	40	Binary constant definition	13	20
DISP	49	Message display	51	93
DISPB	41	Extended message display	177	297
EXIN	42	External data input	29	49
WINDR	51	NC window data read	101	293
WINDW	52	NC window data write	101	293
FNC9X	9X	Arbitrary functional instruction (X=0 to 7)	–	21
MMC3R	88	MMC3 window data read	342	375
MMC3W	89	MMC3 window data write	385	421
MMCWR	98	MMC window data read	100	293
MMCWW	99	MMC window data write	100	293

Execution time constant:

This constant represents how many times the execution time of a functional instruction corresponds to the execution time of 10 basic instructions (about 1.5μs) . The execution time of a basic instruction is about 0.15 μs.

The general format and restrictions common to each functional instruction are given below, details on each instructions will follow later. Refer to this paragraph without fail, since it covers the provisions on using a functional instruction and other important items.

(1) Format
Since the functional instructions cannot be represented with relay symbols, the format shown in Fig. 5 (a) must be used. The format includes control conditions, an instruction, parameters, W1, R9000 to R9005 (Functional instruction operation result register).

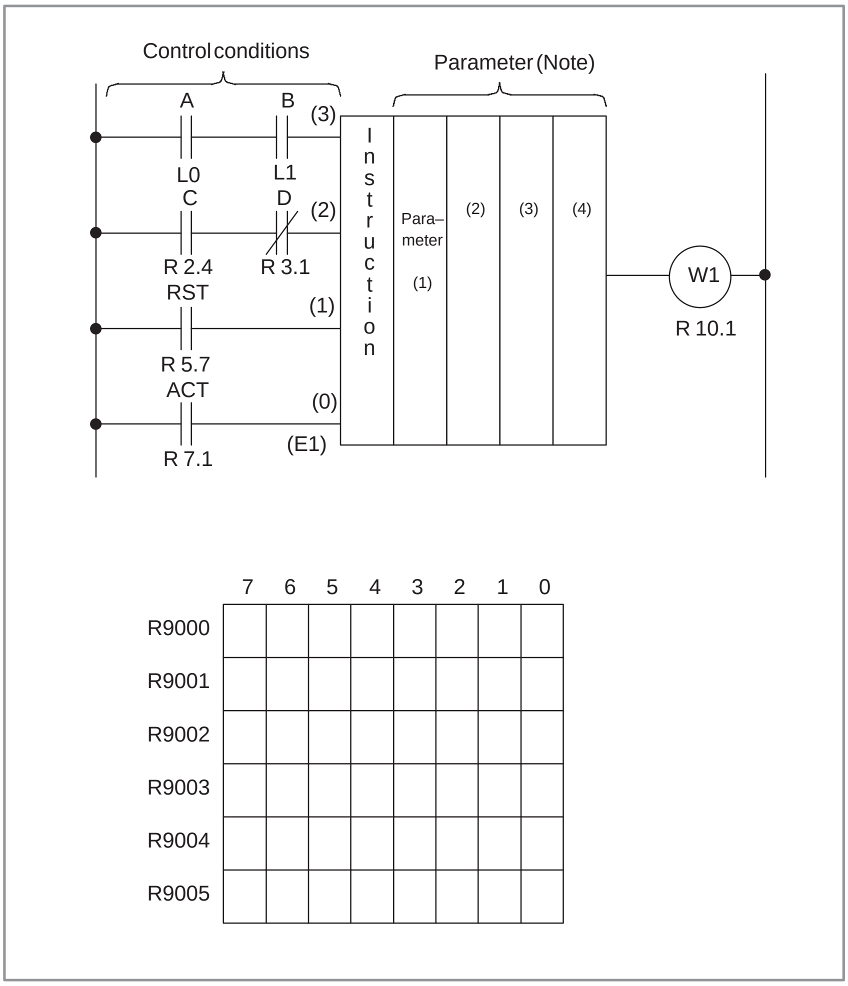


Fig. 5 (a) Function instruction format

Table 5 (c) Coding of function instruction

Coding sheet

Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	R1 .	0	A
2	AND	R1 .	1	B
3	RD. STK	R2 .	4	C
4	AND. NOT	R3 .	1	D
5	RD. STK	R5 .	7	RST
6	RD. STK	R7 .	1	ACT
7	SUB	○○		Instruction
8	(PRM) (Note 2)	○○○○		Parameter 1
9	(PRM)	○○○○		Parameter 2
10	(PRM)	○○○○		Parameter 3
11	(PRM)	○○○○		Parameter 4
12	WRT	R10 .	1	W1 output

Status of operating result

ST3	ST2	ST1	ST0
			A
			$A \cdot B$
		$A \cdot B$	C
		$A \cdot B$	$C \cdot \overline{D}$
	$A \cdot B$	$C \cdot \overline{D}$	RST
$A \cdot B$	$C \cdot \overline{D}$	RST	ACT
$A \cdot B$	$C \cdot \overline{D}$	RST	ACT
$A \cdot B$	$C \cdot \overline{D}$	RST	ACT
$A \cdot B$	$C \cdot \overline{D}$	RST	ACT
$A \cdot B$	$C \cdot \overline{D}$	RST	ACT
$A \cdot B$	$C \cdot \overline{D}$	RST	ACT
$A \cdot B$	$C \cdot \overline{D}$	RST	W1

NOTE

1 Numbers in parentheses under control conditions indicate the position of the stored register.

2 (PRM) of steps 8 to 11 under Instruction means that P must be input when a parameter is input from the programmer, and PRM is not required to be input when a parameter is input from a paper tape.

(2) Control condition

The number and meaning of control conditions vary with each functional instruction. The control conditions are entered in the stick register as shown in Table 5 (b). The sequence is fixed and cannot be changed or omitted.

NOTE

For the functional instructions, with a RST as a control condition, the RST has the highest priority. Accordingly when RST=1, the RST processing is done even when ACT=0.

(3) Instruction

The types of instructions are shown in Table 5 (a). The Programmer has exclusive keys for functional instructions TMR and DEC. They are input by T and D keys, respectively. The other functional instructions are given by “S” key and a following number. When instructions are input by relay symbols, software keys are used to input them. Refer to chapter III or V for details.

(4) Parameter

Unlike basic instructions, functional instructions can handle numeric values. Thus the reference data or addresses containing data are entered under Parameter. The number and meaning vary with each functional instruction. The P key is used to enter parameters in the Programmer.

(5) W1
The operation results of a functional instruction, when represented with one bit of 1 or 0, is output to W1 whose address can be determined freely by the programmer. Its meaning varies with each functional instruction. Note that some functional instructions have no W1.

(6) Data to be processed
Data handled by functional instructions are of binary coded decimal (BCD) code and binary code.
In the conventional PMCs, the numeric data is processed mainly based on the BCD code. However, in the PMC–SB/SC, it is recommended to handle all pieces of numeric data with the binary code. The reasons for this are:

(a) In the Series 16, the numeric data (M, S, T, B code) between the CNC and the PMC should be of the binary code.

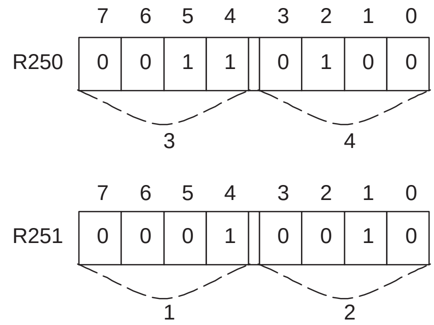
(b) Numeric data on which the CPU performs processing must be in binary format. When numeric data is always processed in binary format, therefore, neither BCD–to–binary nor binary–to–BCD conversion is necessary, thus enabling faster PMC processing.

(c) When the data is of the binary code, the range of the numeric data processable becomes wide. Also, negative numeric data can be processed easily, and the arithmetic operation functions are strengthened. The binary numeric data is handled, as a rule, on the basis of 1 byte (–128 to+127), 2 bytes (–32768 to +32767), and 4 bytes (–99999999 to +99999999).

(d) When various numeric data items are entered or displayed using the keys on the CRT/MDI panel, all the numeric data items in binary are conveniently specified or displayed in decimal. Therefore, no problem arises, though the data stored in the internal memory is of the binary code. Pay attention to this only when referring to the memory by the sequence program. See (7). In the functional instructions, binary data is mainly handled.

(7) Example of numeric data

(a) BCD code data
The basic data handled with the BCD code is of 1 byte (0 to 99) or 2 bytes (0 to 9999). The BCD 4–digit data is entered into two bytes of continuous addresses as shown below.
Example: When BCD data 1234 is stored to addresses R250 and R251.



Specify smaller address R250 by a functional instruction.
(Note) The low order digits are entered to the smaller address.

(b) Binary code data

The basic data handled with the binary code is of 1 byte (–128 to +127), 2 bytes (–32,768 to +32,767) and 4 bytes (–99,999,999 to +99,999,999). The data is stored at addresses R200, R201, R202 and R203 as shown below.

1 byte data (–128 to +127)

	7	6	5	4	3	2	1	0
R200	±	2^6	2^5	2^4	2^3	2^2	2^1	2^0

± : 0 : Positive, 1 : Negative

2 byte data (–32,768 to + 32,767)

	7	6	5	4	3	2	1	0
R200	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
R201	±	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8

4 byte data (–99999999 to +99999999)

	7	6	5	4	3	2	1	0
R200	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
R201	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8
R202	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
R203	±	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}

By a functional instruction, specify smaller address R200.
A negative value is set by the two’s complement code.

Example: 1 byte data

7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	1	(+1)
1	1	1	1	1	1	1	1	(–1)
0	1	1	1	1	1	1	1	(+127)
1	0	0	0	0	0	0	1	(–127)

(8) Addresses of numerical data handled in the function instructions

When numerical data handled in the function instructions are 2 bytes or 4 bytes, addresses of numerical data specified by parameters of function instructions are better to take even numbers.

The use of even addresses slightly reduces the execution time of functional instructions.

These parameters of the functional instructions mainly handling binary data are marked with an asterisk as follows.

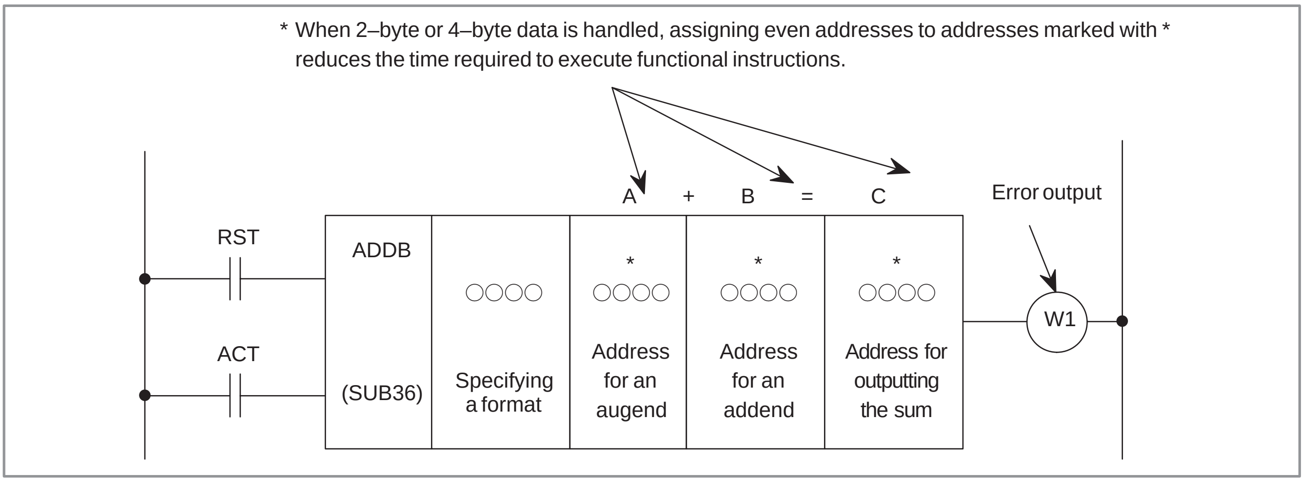


Fig. 5 (b)

In even addresses, the number after R is even with internal relays, and the number after D is even in data tables.

(9) Functional instruction calculation result register (R9000 to R9005) (See Fig. 5 (c))
The result of calculation of the functional instruction is set in the register.
This register is used commonly to the functional instructions.
Therefore, refer to the information in the register immediately after the functional instruction is executed. Otherwise, the previous information disappears when the next functional instruction is executed.
The calculation information in the register cannot be transferred between different levels of the sequence program. For example, it is impossible to read the set information by referring to registers R9000's by the 2nd level program. When the subtraction instruction (SUBB) is executed by the 1st level program.
The calculation information set in the register is guaranteed up to the point just before the functional instruction for setting the next calculation information is executed between the same level of programs. The calculation information set in this register differs according to the functional instruction. It can be read out by the sequence program, but cannot be written.

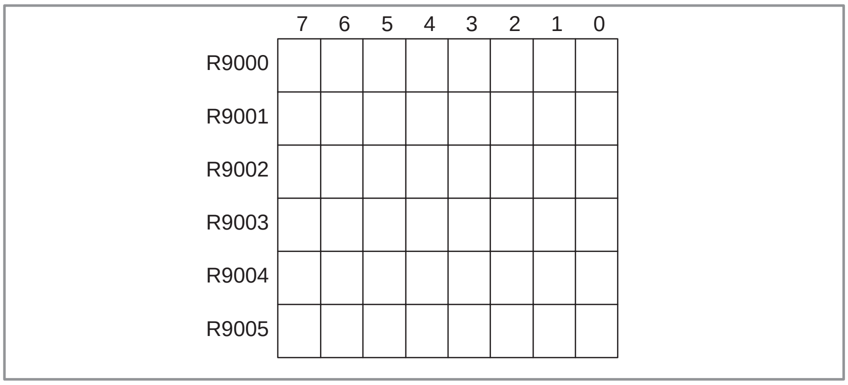


Fig. 5 (c)

This register is a 6 byte register (R9000 to R9005), and the data of 1 bit unit or 1 byte unit can be referred to.
When reading the data of bit 1 of R9000, specify RD R9000.1.

5.1 END1 (1ST LEVEL SEQUENCE PROGRAM END)

5.1.1 Function

Must be specifies once in a sequence program, either at the end of the 1st level sequence, or at the beginning of the 2nd level sequence when there is no 1st level sequence.

5.1.2 Format

Fig. 5.1.2 shows the format of END.1 and Table 5.1.2 shows the coding.

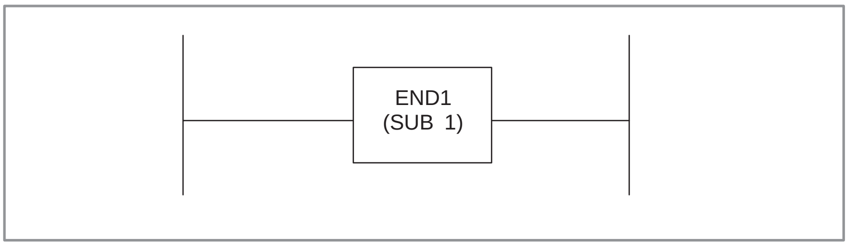


Fig. 5.1.2 Format of END.1

Table 5.1.2 Coding of END.1

Coding sheet

Step Number	Instruction	Address Number	Bit Number	Remarks
	SUB	1		End of 1st level

5.2 END2 (2ND LEVEL SEQUENCE PROGRAM END)

5.2.1 Function

Specify at the end of the 2nd level sequence.

5.2.2 Format

Fig.5.2.2 shows the expression format and Table 5.2.2 shows the coding format.

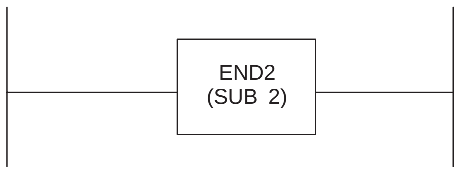


Fig. 5.2.2 Format of END.2

Table 5.2.2 Coding of END.2

Coding sheet

Step Number	Instruction	Address Number	Bit Number	Remarks
	SUB	2		2nd level sequence program end

5.3 END3 (END OF 3RD LEVEL SEQUENCE) (PMC–SC/SC3/SC4/NB/NB2 ONLY)

5.3.1 Function

Specify this command at the end of the 3rd level sequence program, i.e. it indicates the end of the sequence program. If there is no 3rd level sequence program, specify this command immediately after END.2 command.

5.3.2 Format

Fig.5.3.2 shows description format and Table 5.3.2 shows coding format.

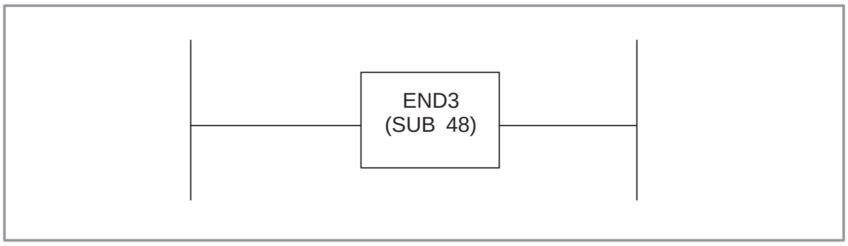


Fig. 5.3.2 END.3 description format

Table 5.3.2 END.3 coding format

Coding sheet

Step Number	Instruction	Address Number	Bit Number	Remarks
	SUB	48		End of 3rd level program

5.4 TMR (TIMER)

5.4.1 Function

This is an on–delay timer.

5.4.2 Format

Fig.5.4.4 (a) shows description format and Table 5.4.4 shows coding format.

5.4.3 Control Condition

ACT=0: Turns off the timer relay (TM○○).
ACT=1: Initiates the timer.

5.4.4 Timer Relay (TM○○)

When the time preset is reached with ACT=1 as shown in Fig.5.4.4 (b), the timer relay turns on. The address of the timer relay is determined by designer.

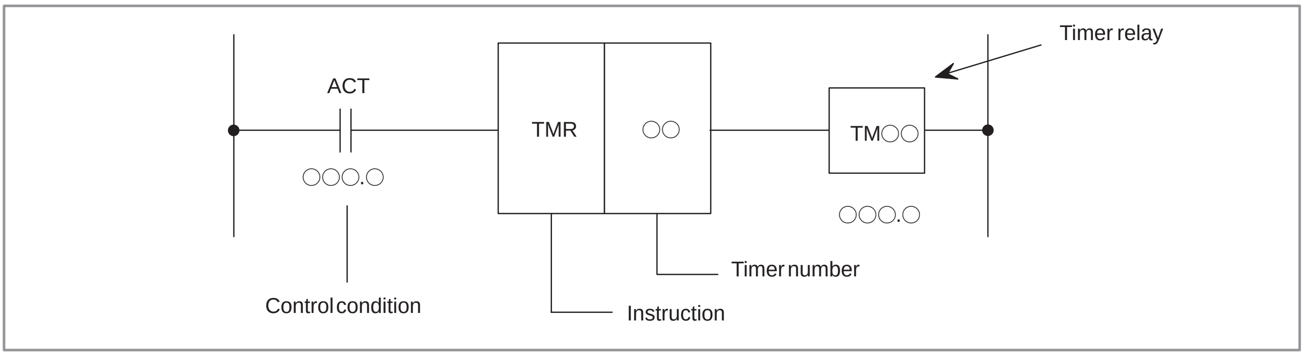


Fig. 5.4.4 (a) Format of TMR

Table 5.4.4 Coding of TMR

Step Number	Instruction	Address Number	Bit Number	Remarks
1	RD	○○○○.	○	ACT
2	TMR	○○		
3	WRT	○○○.	○	TM○○

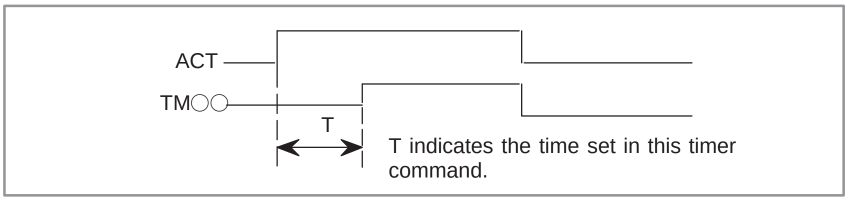


Fig. 5.4.4 (b) Operation of the timer

5.4.5 Setting Timers

The timer can be set via the CRT/MDI unit of the CNC (See Chapter II). The setting time is every 48 ms for timer number 1 to 8 and every 8 ms for timer number 9 to 40. A time less than 48 ms is discarded for timer number 1 to 8. The time set by timers 9 to 40 is every 8 ms. Any remainder is discarded. For example, if 38 ms is set, the remainder 6 (38=8×4+6) is discarded, and only 32 ms is actually set.

Model / Type of timer	PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4
48 ms timer number	1 to 8	1 to 8	1 to 8	1 to 8	1 to 8	1 to 8	1 to 8	1 to 8	1 to 8	1 to 8
8 ms timer number	9 to 40	9 to 40	9 to 40	9 to 40	9 to 40	9 to 40	9 to 40	9 to 40	9 to 40	9 to 150

Model / Type of timer	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
48 ms timer number	1 to 8	1 to 8	1 to 8	1 to 8	1 to 8	1 to 8	1 to 8	1 to 8
8 ms timer number	9 to 40	9 to 150	9 to 40	9 to 40	9 to 150	9 to 40	9 to 150	9 to 150

5.4.6 Timer Accuracy

Type of timer	Setting time	Error
48 ms timer	48 ms to 1572.8 s	–48 to 0 ms
8 ms timer	8 ms to 262.1 s	–8 to 0 ms

Variation in time is caused only by operation time of the Timer Instruction. For example, when a timer instruction is used in the 2nd level sequence part, the variation does not include the delay time (Max. 2nd level sequence one cycle time) until the sequence actuates after the set time is reached.

5.4.7 Parameter

Set the timer number.

WARNING
If the timer number is duplicated, or falls outside the valid range, the operation will be unpredictable.

5.5 TMRB (FIXED TIMER)

5.5.1 Function

This timer is used as a fixed on–delay timer. The variable timer in section 5.4 sets time of the timer into the nonvolatile memory, and can be reset via the CRT/MDI when necessary.
Time present in this fixed timer is written to the ROM together with the sequence program, so the timer time once set cannot be changed unless the whole ROM is exchanged.

5.5.2 Format

The format is expressed as follows (Fig.5.5.2).

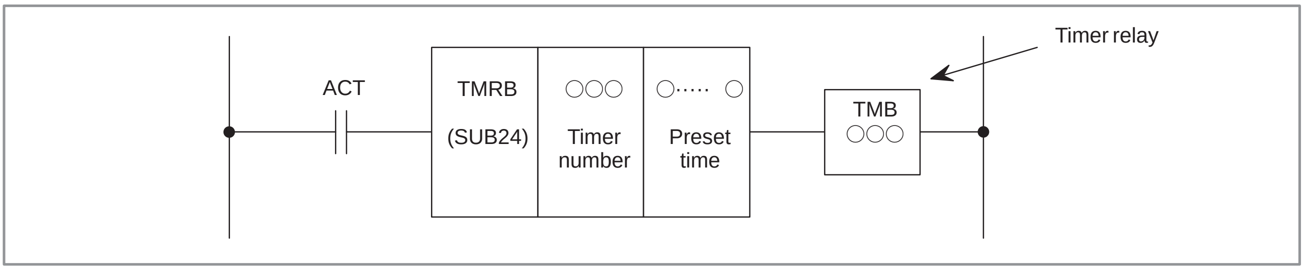


Fig. 5.5.2 Format of TMRB

5.5.3 Control Conditions

ACT=0: Turns off timer relay (TMB○○○).
ACT=1: Start timer.

5.5.4 Timer Relay (TMB○○○)

As shown in Fig.5.5.4, timer relay is set ON after certain time preset in the parameter of this instruction pasts after ACT=1.
The designer will decide the address of the internal relay in the timer relay.

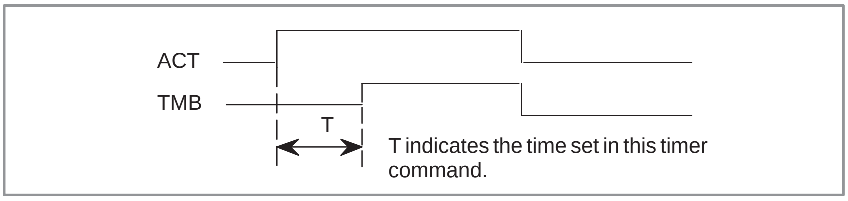


Fig. 5.5.4 Timer operation

5.5.5 Parameter

(a) Timer number
Sets timer number (1 to 100) of the fixed timers.

(b) Preset time (8 to 262,136 ms)
Processing is done every 8 ms in this fixed timer.
The preset time is therefore integral times of 8 ms and the odds are omitted.
For example, when set 38 ms, $38=8\times4+6$, the odd 6 is omitted, and the preset time becomes 32 ms.
The range of the preset time is 8 to 262,136 ms.

5.5.6 Precision of the Timer

Time varies –8 to 0 ms from the setting time.
The varing time in this timer is caused only the error occurred when the timer instruction performs operation process.
Error caused by sequence program processing time (time of 1 cycle of the second level), etc. are not included.

5.6 TMRC (TIMER)

5.6.1 Function

This is the on–delay timer.
A timer setting time is set at an arbitrary address. The selection of an address determines whether the timer is a variable timer or fixed timer. No limit is imposed on the number of timers provided areas can be allocated.

5.6.2 Format

Fig.5.6.2 and Table 5.6.2 show the expression format and the coding format, respectively.

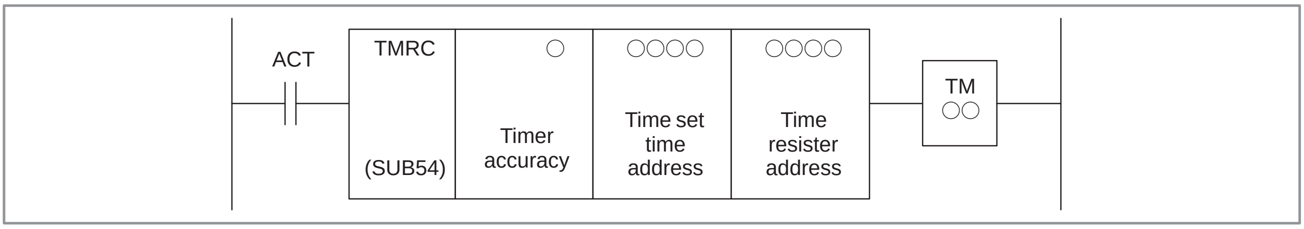


Fig. 5.6.2 TMRC expression format

Table 5.6.2 TMRC coding format

Step Number	Instruction	Address Number	Bit Number	Remarks
1	RD	○○○○.	○	
2	SUB	54		TMRC command
3	(PRM)	○		Timer accuracy
4	(PRM)	○○○		Timer set time address
5	(PRM)	○○○○		Timer register address
6	WRT	○○○○.	○	TM○○

5.6.3 Control Condition

ACT=0 : Turns off the timer relay (TM○○).
ACT=1 : Starts the timer.

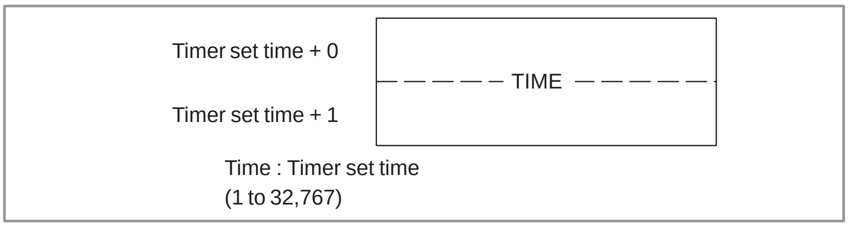
5.6.4 Timer Accuracy

Timer precision	Setting value	Setting time	Error
8 ms	0	1 to 262,136	–8 to 0 ms
48 ms	1	1 to 1,572,816	–48 to +0 ms
1 second (Note)	2	1 to 32,767	0 to +1 s
10 s (Note)	3	1 to 327,670	0 to +10 s
1 m (Note)	4	1 to 32,767	0 to +1 m

NOTE
This function is usable only with the following models:
FS16C/18C PMC–SB5/SB6
FS16*i*/18*i* PMC–SB5/SB6
FS21*i* PMC–SA5

5.6.5 Timer Set Time Address

Sets the first address of the timer set time field.
The continuous 2–byte memory space is required for the timer set time field.
Field D is normally used as this field.

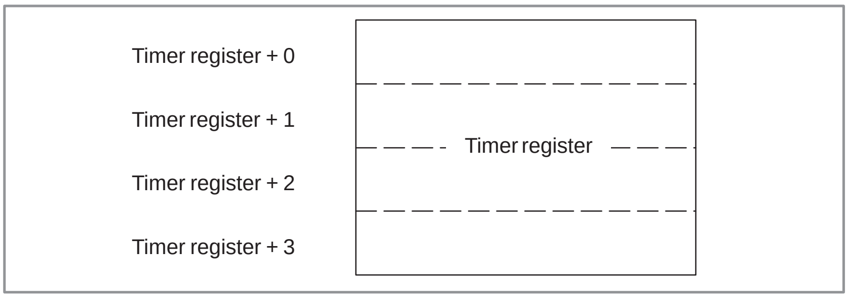


The timer set time is converted into the binary value in 8 ms (48 ms) units. The timer set time is shown as follows:

8 ms - - - - - - 8 to 262,136 ms
48 ms - - - - - 48 to 1,572,816 ms
1 s - - - - - - - 1 to 32,767s
10 s - - - - - - 1 to 327,670s
1 m - - - - - - - 1 to 32,767m

5.6.6 Timer Register Address

Set the start address of a timer register area.
A timer register area must be allocated to a continuous four–byte memory area starting from the set address. Normally, the R area is used as a timer register area. This area should be used by the PMC system, and therefore should not be used by the sequence program.



5.6.7 Timer Relay (TM○○)

As shown in Fig. 5.6.7, after ACT is set to 1, the timer relay is turned on once the time specified in this command has elapsed.

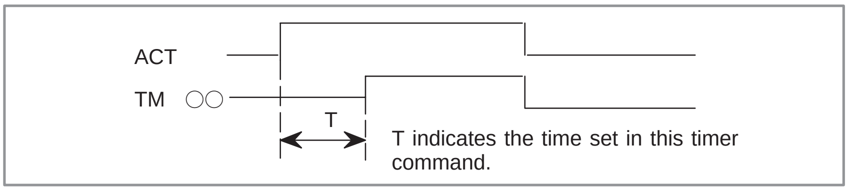


Fig. 5.6.7 Timer operation

5.7 DEC (DECODE)

5.7.1 Function

Outputs 1 when the two–digit BCD code signal is equal to a specified number, and 0 when not. Is used mainly to decode M or T function.

5.7.2 Format

Fig.5.7.2 and Table 5.7.2 show the expression format and Table 5.7.2 show the coding format.

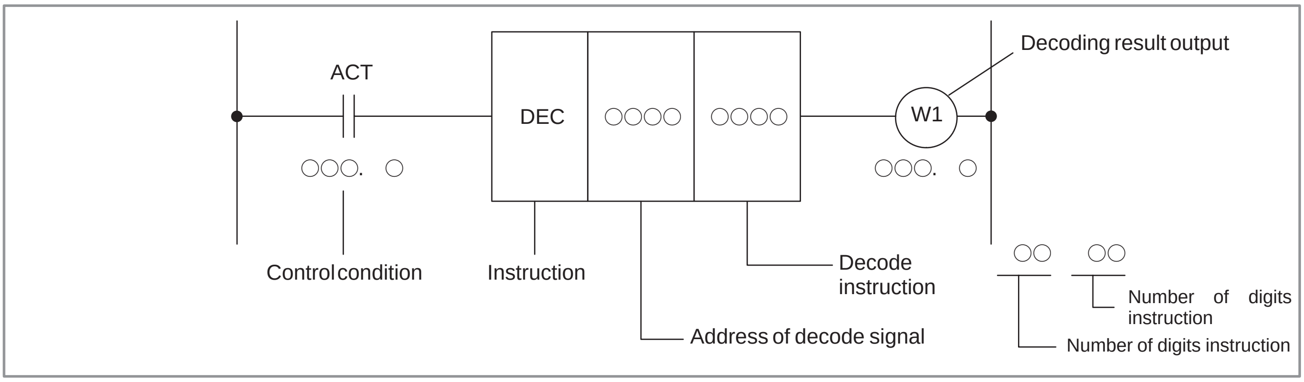


Fig. 5.7.2 Format of DEC

Table 5.7.2 Coding of DEC

Step Number	Instruction	Address Number	Bit Number	Remarks
1	RD	○○○.	○	ACT
2	DEC	○○○○		
3	(PRM)	○○○○		
4	WRT	○○○.	○	W1, Decoding result output

5.7.3 Control Condition

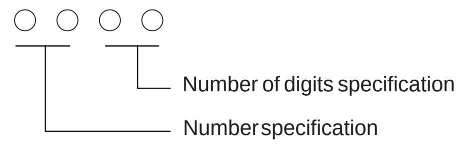
ACT=0 : Turns the decoding result output off (W1).
ACT=1 : Performs decoding.
When the specified number is equal to the code signal, W1=1; when not, W1=0.

5.7.4 Code Signal Address

Specify the address containing two–digit BCD code signals.

5.7.5 Decode Specification

There are two paths, the number and the number of digits.
Decode specification



(i) Number:
Specify the decode number.
Must always be decoded in two digits.

(ii) Number of digits:
- 01 : The high–order digit of two decimal digits is set to 0 and only the low–order digit is decoded.
- 10 : The low–order digit is set to 0 and only the high–order digit is decoded.
- 11 : Two decimal digits are decoded.

5.7.6 W1 (Decoding Result Output)

W1 is 1 when the status of the code signal at a specified address is equal to a specified number, 0 when not. The address of W1 is determined by designer.

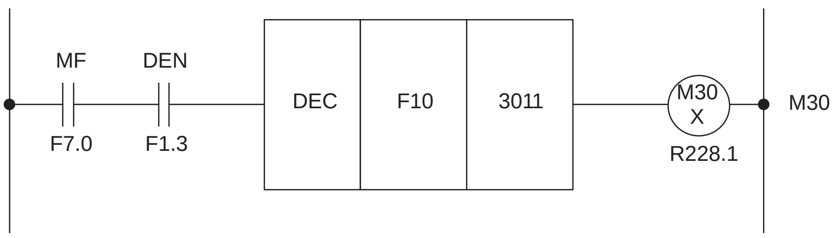


Fig. 5.7.6 Ladder diagram using the DEC instruction

Table 5.7.6 Coding for Fig.5.7.6

Coding sheet

Step Number	Instruction	Address Number	Bit Number	Remarks
1	RD	F 7.0		
2	AND	F 1 .3		
3	DEC	F 10		
4	(PRM)	3011		
5	WRT	R228 .	1	M30X

5.8 DECB (BINARY DECODING)

5.8.1 Function (Fig.5.8.2 (a), (b))

DECB decodes one, two, or four–byte binary code data. When one of the specified eight consecutive numbers matches the code data, a logical high value (value 1) is set in the output data bit which corresponds to the specified number. When these numbers do not match, a logical low value (value 0) is set.
Use this instruction for decoding data of the M or T function.

In PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i*/Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*, the setting of the format specification parameter is extended. With this setting, DECB can decode multiple (8 × n) bytes.
For the details of the setting of a format specification parameter, refer to “5.8.4 Parameters”.

5.8.2 Format

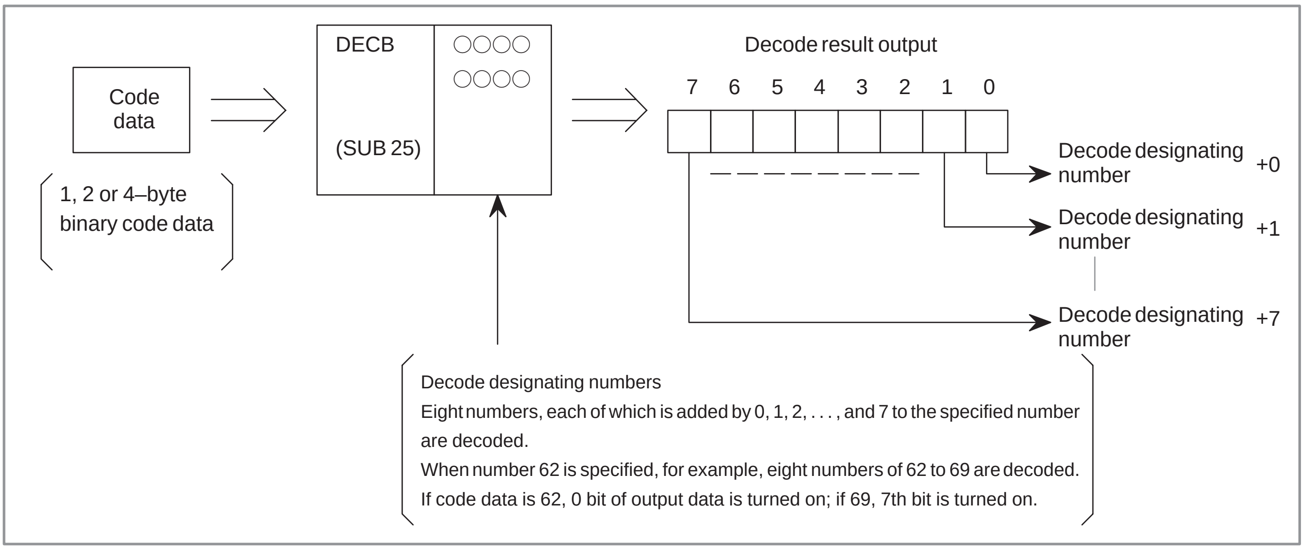


Fig. 5.8.2 (a) Function of DECB (basic specification)

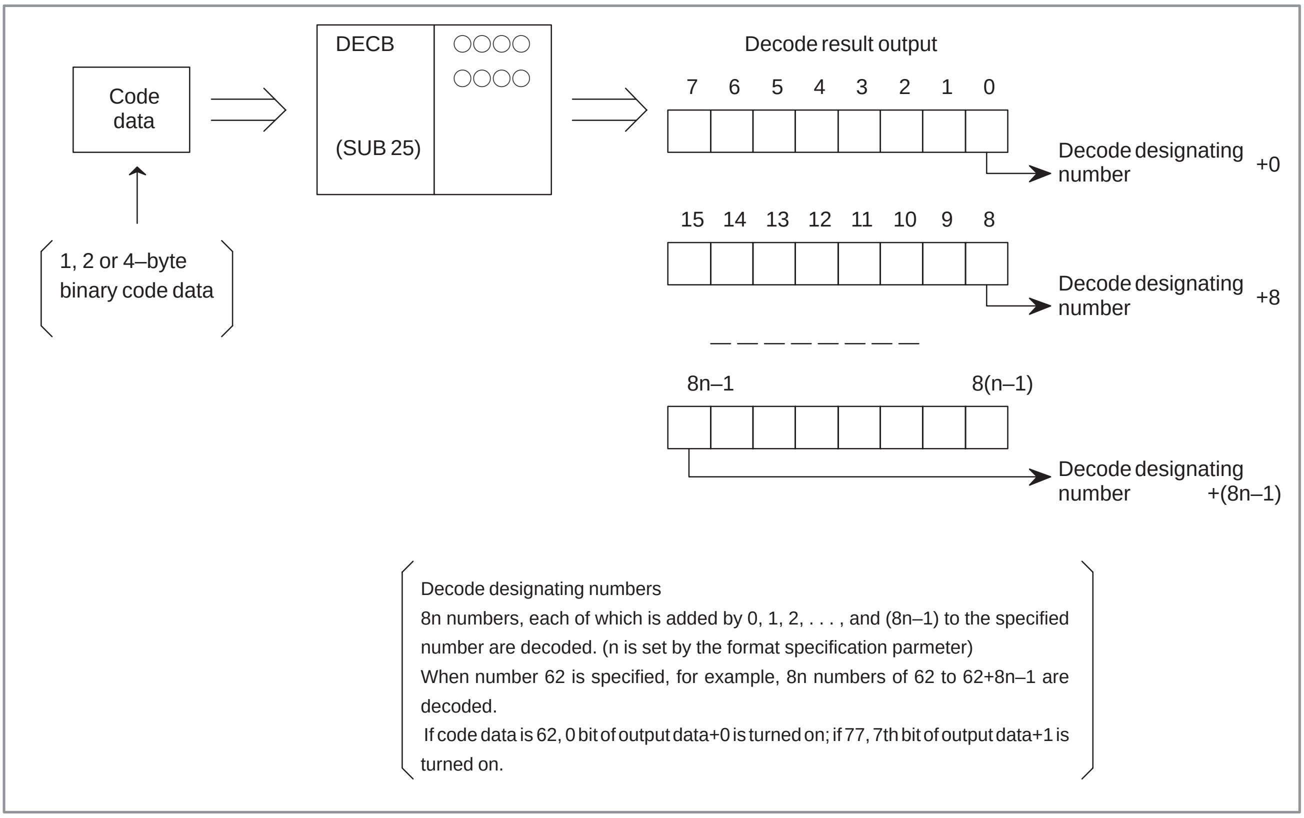


Fig. 5.8.2 (b) Function of DECB (extended specification)
(only for PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*)

Fig.5.8.2 (c), (d) show the expression format.

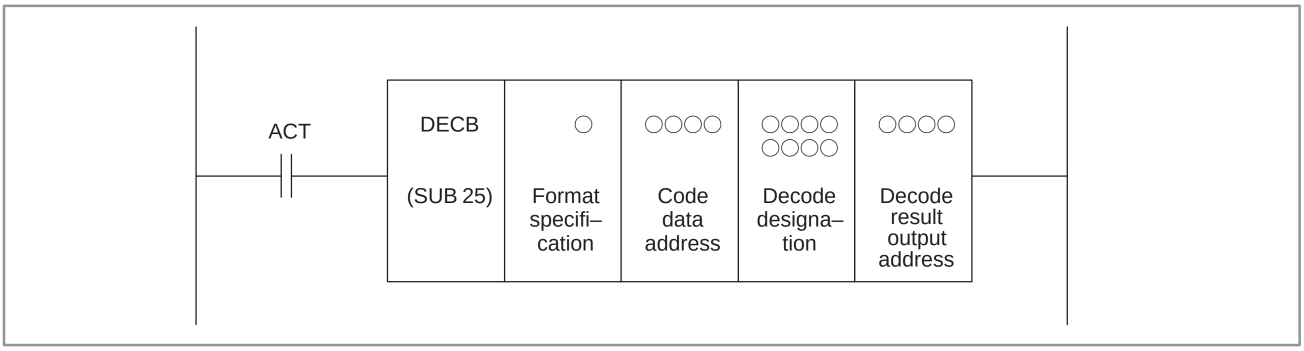


Fig. 5.8.2 (c) Expression format of DECB (basic specification)

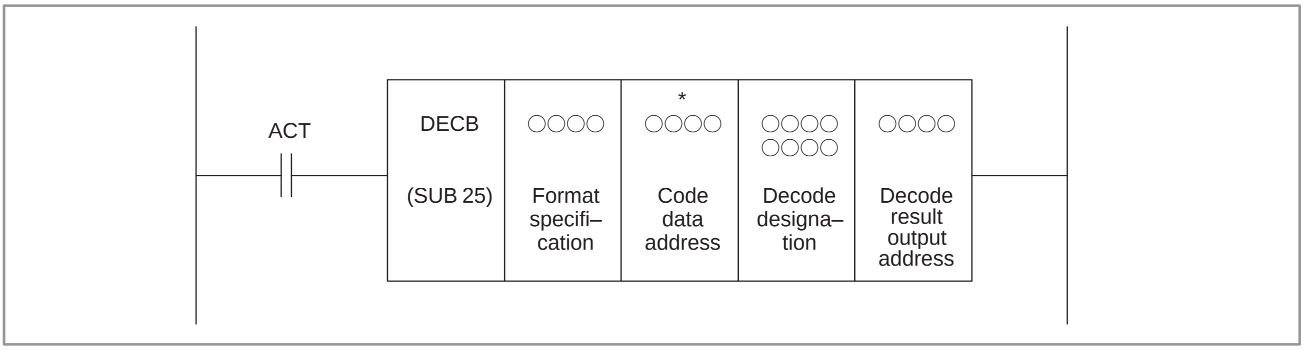


Fig. 5.8.2 (d) Expression format of DECB (extended specification)
(only for PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*)

5.8.3 Control Conditions

(a) Command (ACT)

ACT=0 : Resets all the output data bits.

ACT=1 : Decodes data.
Results of processing is set in the output data address.

5.8.4 Parameters

(a) Format specification

Set the size of code data to the 1st digit of the parameter.

0001 : Code data is in binary format of 1 byte length

0002 : Code data is in binary format of 2 byte length

0004 : Code data is in binary format of 4 byte length

In PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*, when setting format specification in the following extended format, DECB can decode multiple ($8 \times$n) bytes by 1 instruction.

0nn1 : In case of decoding multiple ($8 \times$nn) bytes and code data is binary format of 1 byte length

0nn2 : In case of decoding multiple ($8 \times$nn) bytes and code data is binary format of 2 byte length

0nn4 : In case of decoding multiple ($8 \times$nn) bytes and code data is binary format of 4 byte length

The nn is the numerical value from 02 to 99. When setting 00 or 01, it works for decoding 8 numbers.

```
Format specification (extended specification) :
   0  n  n  x

            The byte length setting of code data
               1 :  1 byte length
               2 :  2 byte length
               4 :  4 byte length

 The multiple decoding number setting
   00–01 :
      It decodes 8 continuous numbers.
      The decode result output address needs a memory of 1 byte length.
   02–99 :
      It decodes multiple (8 × nn) continuous numbers.
      The decode result output address needs a memory of nn bytes length.
```

(b) Code data address

specifies an address at which code data is stored.

(c) Number specification decode designation

Specifies the first of the 8 continuous numbers to be decoded.

(d) Decode result address

Specifies an address where the decoded result shall be output.
A one–byte area is necessary in the memory for the output.

In PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*, when executing this instruction in extended specification, the area of setting by the format specification for the nn bytes is necessary.

5.9 CTR (COUNTER)

5.9.1 Function

CTR is used as a counter. Counters are used for various purposes for NC Machine tools.
Numerical data such as preset values and count values can be used with either BCD format or binary format by a system parameter.

> **NOTE**
> When a incollect BCD data was set to a BCD type counter, the morement of CTR cannot be sured.

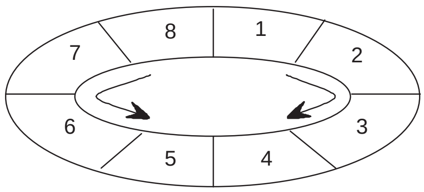
This counter has the following functions to meet various applications.

(a) Preset counter
Outputs a signal when the preset count is reached. The number can be preset from the CRT/MDI panel, or set in the sequence program.

(b) Ring counter
Upon reaching the preset count, returns to the initial value by issuing another count signal.

(c) Up/down counter
The count can be either up or down.

(d) Selection of initial value
Selects the initial value as either 0 or 1.
A combination of the preceding functions results in the ring counter below.



Such a counter permits the position of a rotor to be memorized.

5.9.2 Format

Fig.5.9.2 show the expression format and Table 5.9.2 show the coding format.

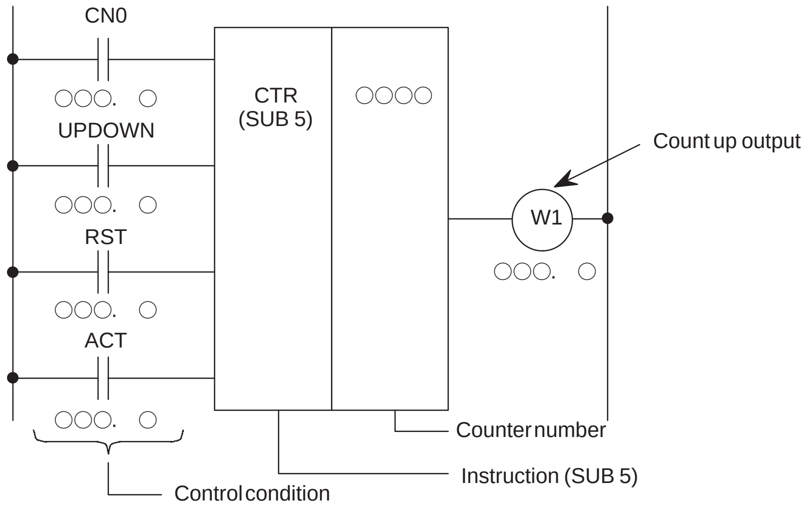


Fig. 5.9.2 Format of CRT instruction

Table 5.9.2 Coding for Fig.5.9.2

Coding sheet

Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	○○○ .	○	CN0
2	RD. STK	○○○ .	○	UPDOWN
3	RD. STK	○○○ .	○	RST
4	RD. STK	○○○ .	○	ACT
5	SUB	5		CTR instruction
6	(PRM)	○○		Counter number
7	WRT	○○○ .	○	W1 output number

Memory status of control condition

ST3	ST2	ST1	ST0
			CN0
		CN0	UPDOWN
	CN0	UPDOWN	RST
CN0	UPDOWN	RST	ACT
CN0	UPDOWN	RST	ACT
CN0	UPDOWN	RST	ACT
CN0	UPDOWN	RST	W1

5.9.3 Control Conditions

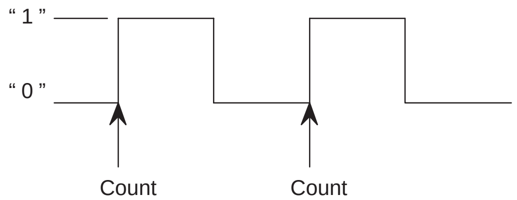
(a) Specify the initial value. (CN0)
CN0=0: Begins the value of the counter with 0.
0, 1, 2, 3 ····· n.
CN0=1: Begins the value of the counter with 1 (0 is not used).
1, 2, 3 ····· n.

(b) Specify up or down counter.
UPDOWN=0:
Up counter. The counter begins with 0 when CN0=0; 1 when 1.
UPDOWN=1:
Down counter. The counter begins with the preset value.

(c) Reset (RST)
RST=0: Releases reset.
RST=1: Enables reset.
W1 becomes 0. The integrated value is reset to the initial value.

NOTE
Set RST to 1, only when reset is required.

(d) Count signal (ACT)



ACT { 0 : Counter does not operate. W1 does not change.
1 : Count is made by catching the rise of ACT.

5.9.4 Counter Number

Counters of 2 bytes (2 bytes for each of the preset values and cumulative values) are available. The usable numbers are listed below.

Model	PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4
Counter number	1 to 20	1 to 20	1 to 20	1 to 20	1 to 20	1 to 20	1 to 20	1 to 20	1 to 20	1 to 50

Model	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
Counter number	1 to 20	1 to 50	1 to 20	1 to 20	1 to 50	1 to 20	1 to 50	1 to 50

WARNING
If the counter number is duplicated, or falls outside the valid range, the operation will be unpredictable.

5.9.5 Countup Output (W1)

When the count is up to a preset value, W1=1. The address of W1 can be determined arbitrarily.
When the counter reaches the set value, W1 is set to 1.
When the counter reaches 0 or 1, W1 is set to 1.

5.9.6 Examples of Using the Counter

[Example 1]
As a preset counter (See Fig.5.9.6 (a))
The number of workpieces to be machined is counted. When the number reaches the preset count, a signal is output.

- L1 is a circuit to make logic 1.
- Since the count ranges from 0 to 9999, contact B of L1 is used for making CN0=0.
- Since it is to be up counter, contract B of L1 is used make UPDOWN=0.
- The reset signal of the counter uses input signal CRST.M from the machine tool.
- The count signal is M30X, which was decoded from the CNC output M code. M30X contains contact B of CUP to prevent counting past the preset value, as long as reset is not enabled after countup.

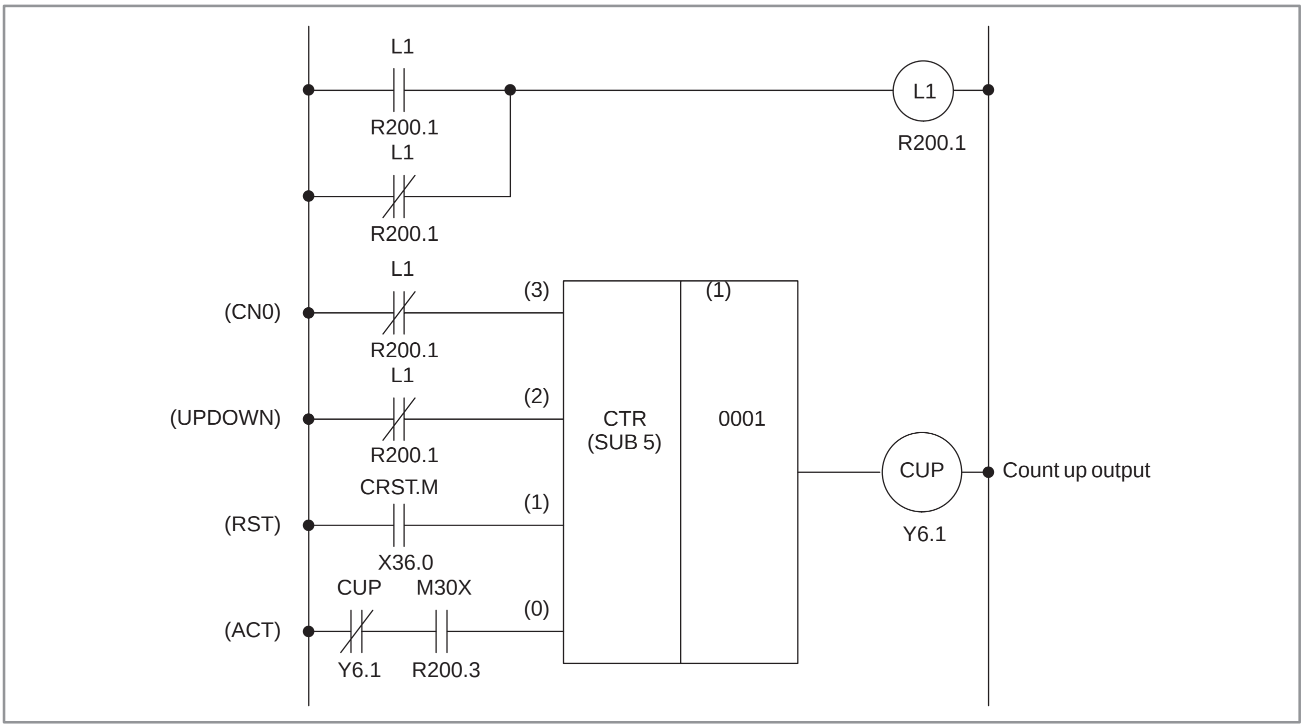


Fig. 5.9.6 (a) Ladder diagram for the counter, example 1

[Example 2]

Use of the counter to store the position of a rotor. (See Fig.5.9.6 (b))

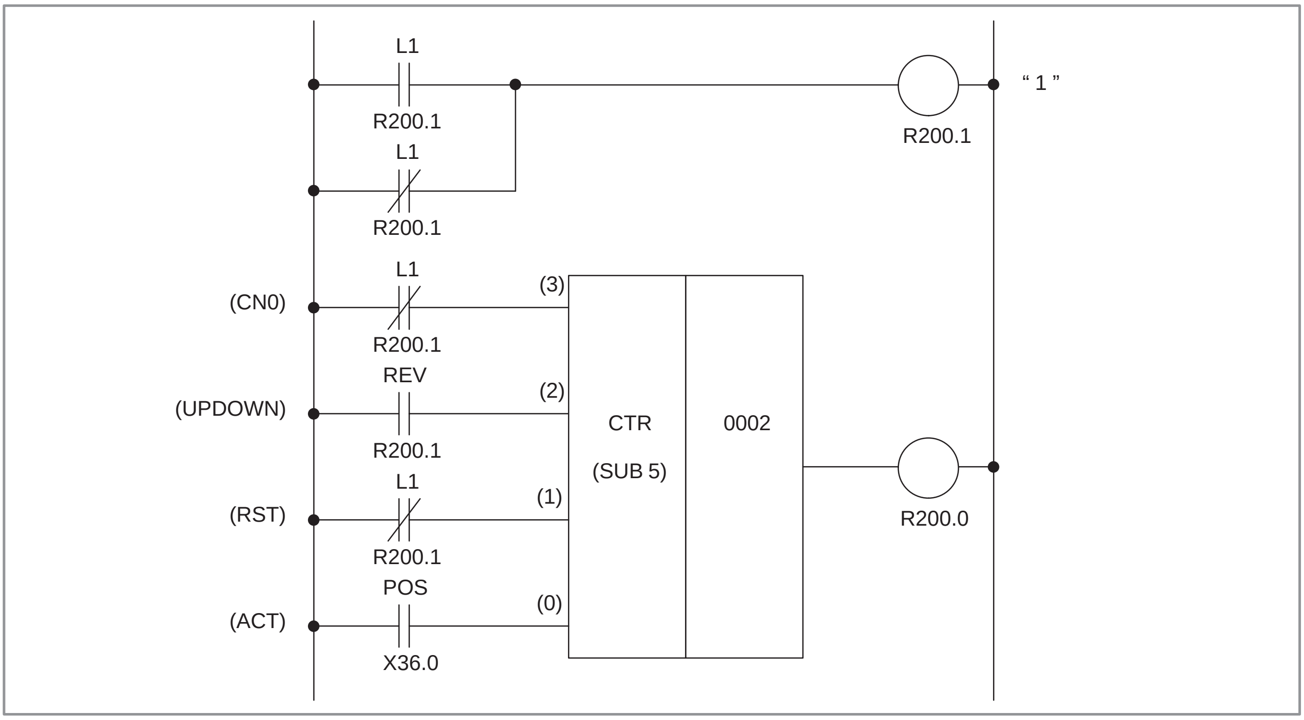


Fig. 5.9.6 (b) Ladder diagram for the counter, example 2

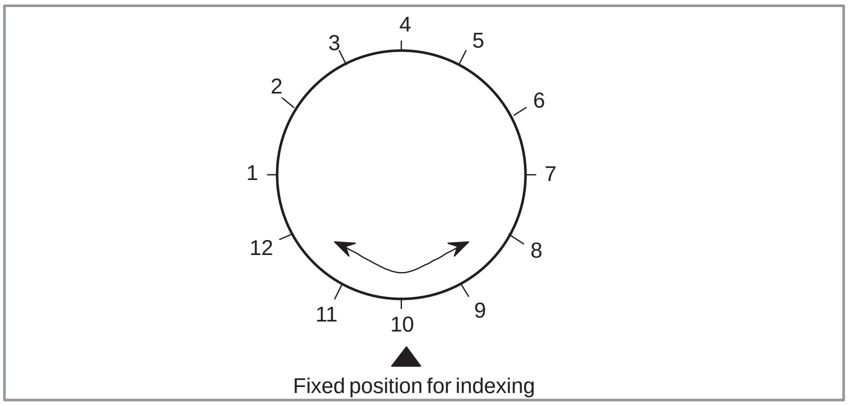


Fig. 5.9.6 (c) Indexing for a rotor

Fig.5.9.6 (b) shows a ladder diagram for a counter to store the position of a rotor of Fig.5.9.6 (c).

(1) Control conditions

(a) Count start number
When a 12–angle rotor shown in Fig.5.9.6 (c) is used, the count starting number is 1. Contact A of L1 is used for making CN0=1.

(b) Specify up and down
The signal REV changes according to the then direction of rotation. It becomes 0 for forward rotation and 1 for reverse rotation. Thus, the counter is an up counter for forward rotation and a down counter for reverse rotation.

(c) Reset
In this example, since W1 is not used, RST=0, and contact B of L1 is used.

(d) Count signal
The count signal POS turns on and off 12 times each time the rotor rotates once.

(2) Counter number and W1
In this example, the second counter is used. The result of W1 is not used, but its address must be determined.

(3) Operation

(a) Setting the preset value
Since the rotor to be controlled is 12–angle as shown in Fig.5.9.6 (c), 12 must be preset in the counter. It is set from the CRT/MDI panel.

(b) Setting the current value
When the power is turned on, the position of the rotor must be equated with the count on the counter. The count is set via the CRT/MDI panel. Once a current value is set, then correct current positions will be loaded to the counter every time.

(c) The POS signal turns on and off each time the rotor rotates.
The number of times of the POS signal turns on and off is counted by the counter, as below.
1, 2, 3, . . . 11, 12, 1, 2, . . .
for forward rotation
1, 12, 11, . . . 3, 2, 1, 12 . . .
for reverse rotation

5.10 CTRC (COUNTER)

5.10.1 Functions

The numeral data of this counter are all binary. This counter has the following functions and can be used according to the application:

(a) Preset counter
Preset the count value and if the count reaches this preset value, outputs to show that.

(b) Ring counter
This is the ring counter which is reset to the initial value when the count signal is input after the count reaches the preset value.

(c) Up/down counter
This is the reversible counter to be used as both the up counter and down counter.

(d) Selection of the initial value
Either 0 or 1 can be selected as the initial value.

5.10.2 Format

Fig.5.10.2 and Table 5.10.2 show the expression format and the coding format, respectively.

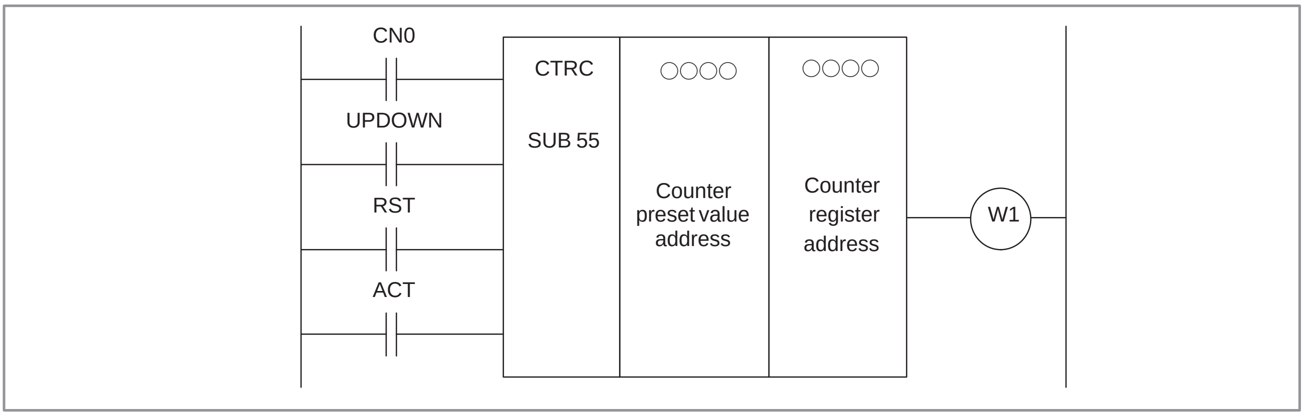


Fig. 5.10.2 CTRC expression format

Table 5.10.2 CRTC coding format

Step Number	Instruction	Address Number	Bit Number	Remarks
1	RD	○○○○.	○	CN0
2	RD.STK	○○○○.	○	UPDOWN
3	RD.STK	○○○○.	○	RST
4	RD.STK	○○○○.	○	ACT
5	SUB	55		CRTC command
6	(PRM)	○○○○		Counter preset address
7	(PRM)	○○○○		Counter register address
8	WRT	○○○○.	○	W1

5.10.3 Control Conditions

(a) Specifying the initial value (CN0)
CN0=0 : The count value starts with “0”. 0, 1, 2, 3, . . . n
CN0=1 : The count value starts with “1”. 1, 2, 3, . . . n

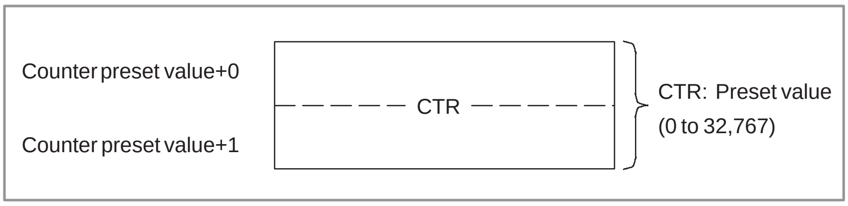
(b) Specifying up or down count (UPDOWN)
UPDOWN=0:
Up counter.
The initial value is “0” when CN0=0 or “1” when CN0=1.
UPDOWN=1:
Down counter. The initial value is the preset value.

(c) Reset (RST)
RST=0 : Reset cancelled.
RST=1 : Reset. W1 is reset to “0”. The accumulated value is reset to the initial value.

(d) Count signal (ACT)
ACT=0 : The counter does not operate. W1 does not change.
ACT=1 : The counter operates at the rise of this signal.

5.10.4 Counter Preset Value Address

The first address of the counter preset value field is set.
The continuous 2–byte memory space from the first address is required for this field. Field D is normally used.



The counter preset value is binary. Therefore, it ranges from 0 to 32767.

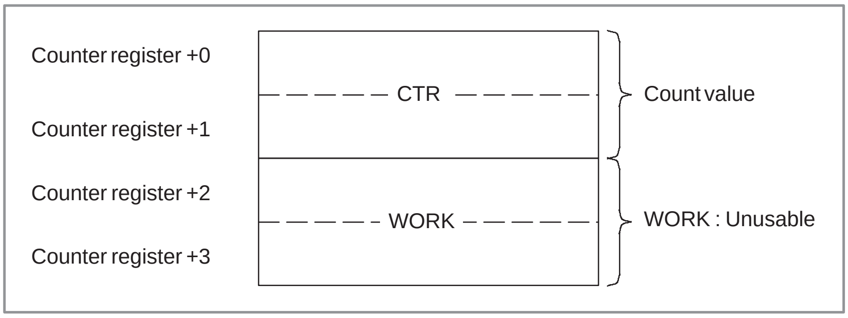
5.10.5 Counter Register Address

The first address of the counter register field is set.
The continuous 4–byte memory space from the first address is required for this field. Field D is normally used.

> **NOTE**
> When field R is specified as the counter register address, the counter starts with count value “0” after powered on.

5.10.6 Count–up Output (W1)

If the count value reaches the preset value, W1 is set to “1”.
The W1 address can be determined freely.



5.11 ROT (ROTATION CONTROL)

5.11.1 Function

Controls rotors, such as the tool post, ATC, rotary table, etc., and is used for the following functions.

(a) Selection of the rotation direction via the shorter path

(b) Calculation of the number of steps between the current position and the goal position

(c) Calculation of the position one position before the goal or of the number of steps up to one position before the goal

5.11.2 Format

Fig.5.11.2 shows the expression format and Table 5.11.2 shows the coding format.

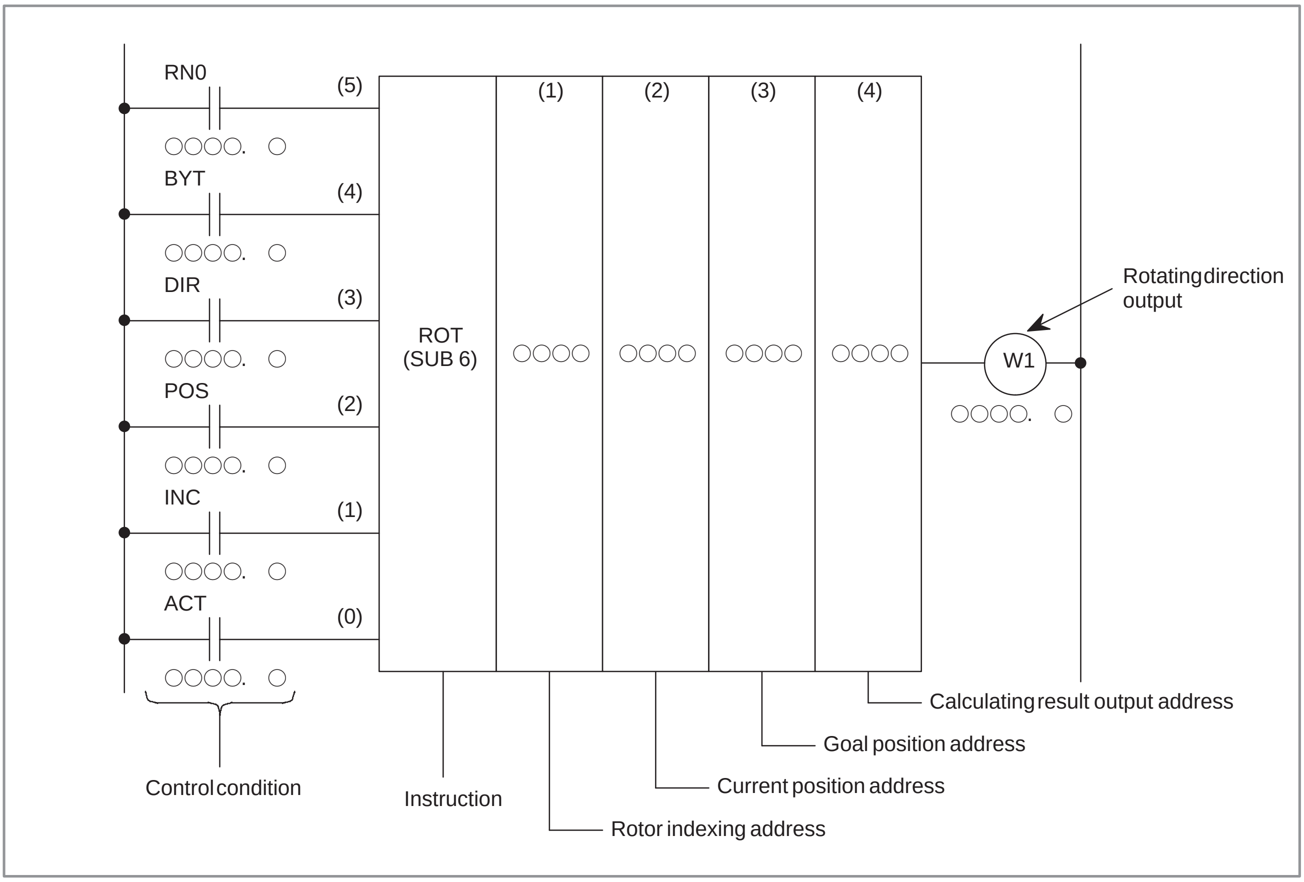


Fig. 5.11.2 ROT instruction format

Table 5.11.2 Coding for Fig.5.11.2

Coding sheet

Step Number	Instruc-tion	Address No.	Bit No.	Remarks
1	RD	○○○○ .	○	RN0
2	RD. STK	○○○○ .	○	BYT
3	RD. STK	○○○○ .	○	DIR
4	RD. STK	○○○○ .	○	POS
5	RD. STK	○○○○ .	○	INC
6	RD. STK	○○○○ .	○	ACT
7	SUB	6		ROT
8	(PRM)	○○○○		Rotor indexing number
9	(PRM)	○○○○		Current position
10	(PRM)	○○○○		Goal position address
11	(PRM)	○○○○		Calculating result output address
12	WRT	○○○ .	○	
13				
14				
15				

Status of operating result

ST5	ST4	ST3	ST2	ST1	ST0
					RN0
				RN0	BYT
			RN0	BYT	DIR
		RN0	BYT	DIR	POS
	RN0	BYT	DIR	POS	INC
RN0	BYT	DIR	POS	INC	ACT
RN0	BYT	DIR	POS	INC	ACT
RN0	BYT	DIR	POS	INC	ACT
RN0	BYT	DIR	POS	INC	ACT
RN0	BYT	DIR	POS	INC	ACT
RN0	BYT	DIR	POS	INC	ACT
RN0	BYT	DIR	POS	INC	W1

5.11.3 Control Conditions

(a) Specify the starting number of the rotor.
 RN0=0 : Begins the number of the position of the rotor with 0.
 RN0=1 : Begins the number of the position of the rotor with 1.

(b) Specify the number of digits of the process data (position data).
 BYT=0 : BCD two digits
 BYT=1 : BCD four digits

(c) Select the rotation direction via the shorter path or not.
 DIR=0 : No direction is selected. The direction of rotation is only forward.
 DIR=1 : Selected. See (8) for details on the rotation direction.

(d) Specify the operating conditions.
 POS=0 : Calculates the goal position.
 POS=1 : Calculates the position one position before the goal position.

(e) Specify the position or the number of steps.
 INC=0 : Calculates the number of the position. If the position one position before the goal position is to be calculated, specify INC=0 and POS=1
 INC=1 : Calculates the number of steps. If the difference between the current position and the goal position is to be calculated, specify INC=1 and POS=0.

(f) Execution command
 ACT=0 : The ROT instruction is not executed. W1 does not change.
 ACT=1 : Executed. Normally, set ACT=0. If the operation results are required, set ACT=1.

5.11.4 Rotor Indexing Number

Specify the rotor indexing number.

5.11.5 Current Position Address

Specify the address storing the current position.

5.11.6 Goal Position Address

Specify the address storing the goal position (or command value), for example the address storing the CNC output T code.

5.11.7 Operation Result Output Address

Calculate the number of steps for the rotor to rotate, the number of steps up to the position one position before, or the position before the goal. When the calculating result is to be used, always check that ACT=1.

5.11.8 Rotating Direction Output (W1)

The direction of rotation for control of rotation via the shorter path is output to W1. When W1=0, the direction is forward (FOR) when 1, reverse (REV). The definition of FOR and REV is shown in Fig.5.11.8. If the number given to the rotor is ascending, the rotation is FOR; if descending, REV. The address of W1 can be determined arbitrarily. When, however, the result of W1 is to be used, always check that ACT=1.

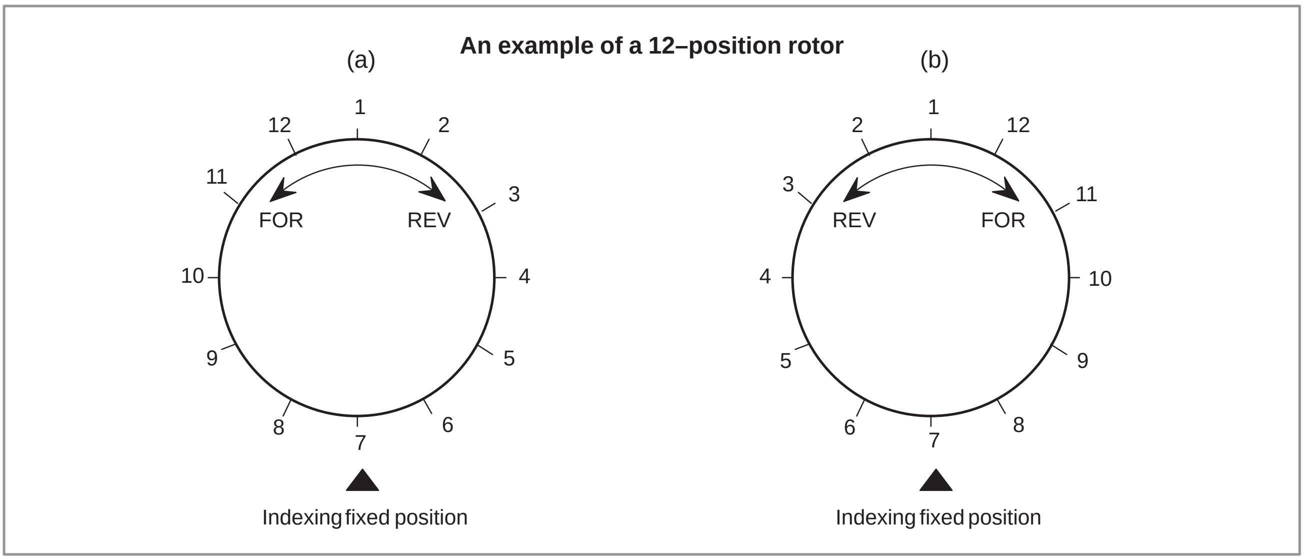


Fig. 5.11.8 Rotation direction

5.12 ROTB (BINARY ROTATION CONTROL)

5.12.1 Function

This instruction is used to control rotating elements including the tool post, ATC (Automatic Tool Changer), rotary table, etc. In the ROT command (5.11) a parameter indicating the number of rotating element indexing positions is a fixed data in programming. For ROTB, however, you can specify an address for the number of rotating element index positions, allowing change even after programming. The data handled are all in the binary format. Otherwise, ROTB is coded in the same way as ROT.

5.12.2 Format

Fig.5.12.2 shows the expression format of ROTB

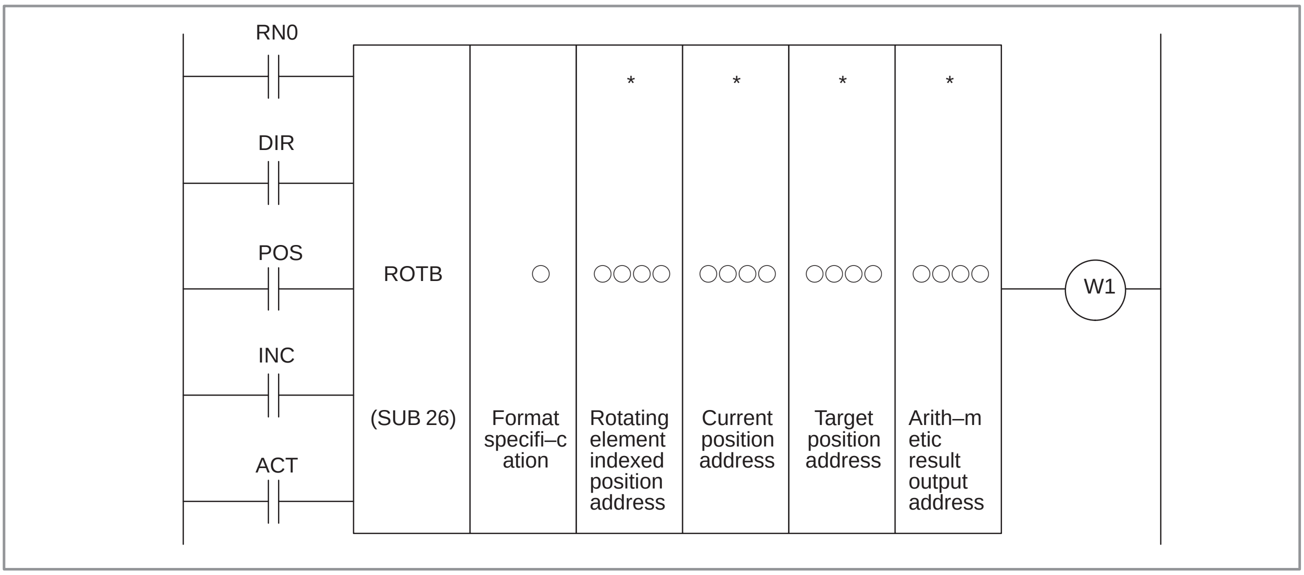


Fig. 5.12.2 Expression format of ROTB

5.12.3 Control Conditions

The control conditions do not differ basically from those for ROT command described in section 5.11. However, BYT has been eliminated from ROTB (it forms part of the ROTB parameters).
For the reset, see ROT.

5.12.4 Parameters

(a) Format
Specifies data length (1, 2, or 4 bytes). Use the first digit of the parameter to specify the number of bytes.
1 : 1 byte
2 : 2 bytes
4 : 4 bytes
All numerical data (number of indexed positions for the rotating elements, current address, etc.) are in the binary format.
Therefore, they require the memory space specified by data length.

(b) Rotating element indexed position address
Specifies the address containing the number of rotary element positions to be indexed.

(c) Other parameters
For the functions and use of the other parameters, see Section 5.11.

5.12.5 Output for Rotational Direction (W1)

See Section 5.11.

5.12.6 Example of Using the ROTB Instruction

Fig. 5.12 (b) illustrates a ladder diagram for a 12–position rotor to be controlled for rotation via the shorter path and for deceleration at the position one position before the goal.

- The goal position is specified with CNC 32B of binary code (address F26 to F29).
- The current position is entered with the binary code signal (address X41) from the machine tool.
- The result of calculating the position one position before the goal is output to address R230 (work area).
- Operation starts with the output TF (address F7.3) from the CNC.
- The coincidence check instruction (COIN) is used to detect the deceleration and stop positions.

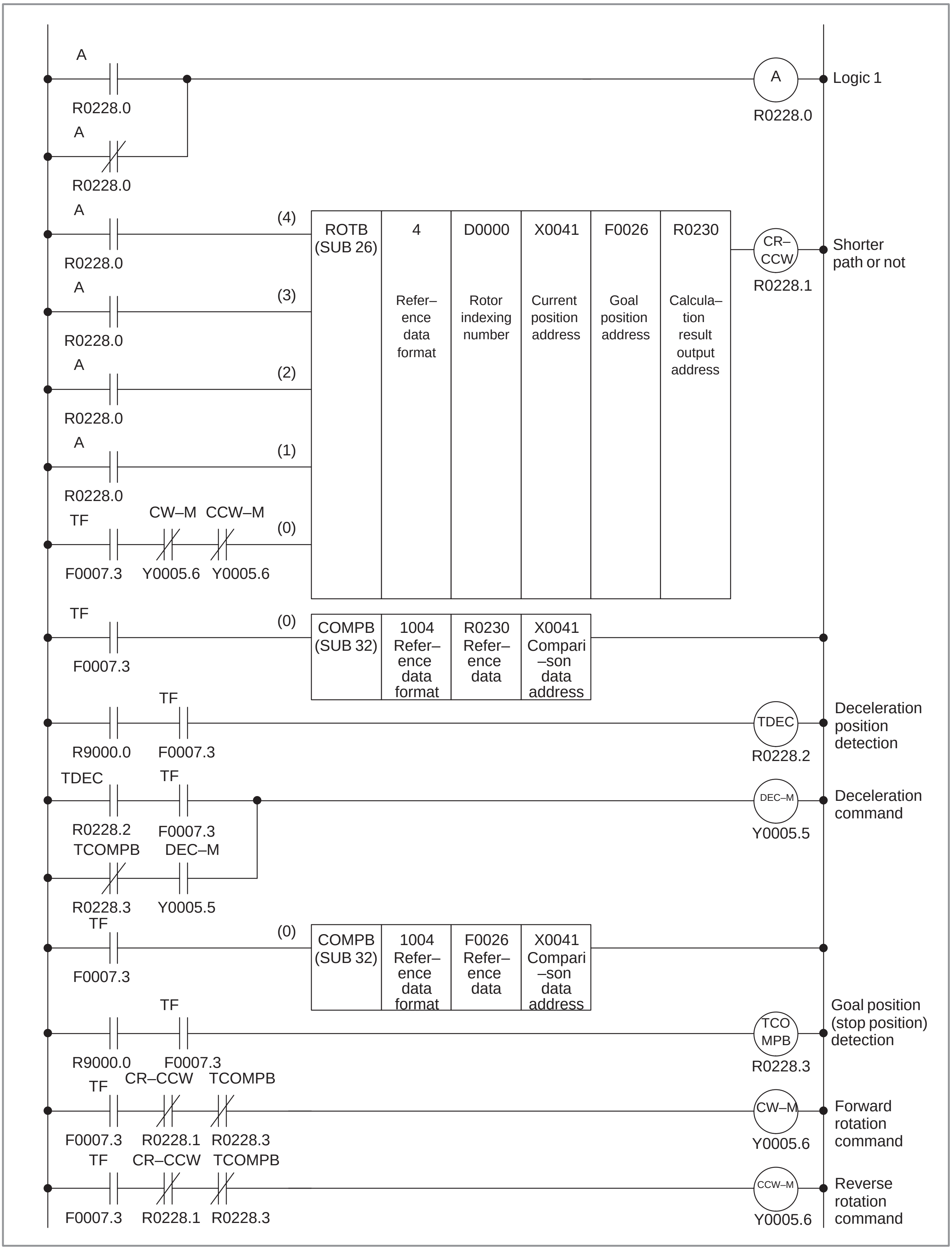


Fig. 5.12.6 Example of a ladder diagram for the ROTB instruction

5.13 COD (CODE CONVERSION)

5.13.1 Function

Converts BCD codes into an arbitrary two– or four–digits BCD numbers. For code conversion shown in Fig.5.13.1 the conversion input data address, conversion table, and convert data output address must be provided.

Set a table address, in which the data to be retrieved from the conversion table is contained, to conversion table input data address in a two–digits BCD number. The conversion table is entered in sequence with the numbers to be retrieved in the two– or four–digits number. The contents of the conversion table of the number entered in the conversion input data address is output to the convert data output address. As shown in Fig.5.13.1, when 3 is entered in the conversion input data address, the contents 137 located at 3 in the conversion table is output to the convert data output address.

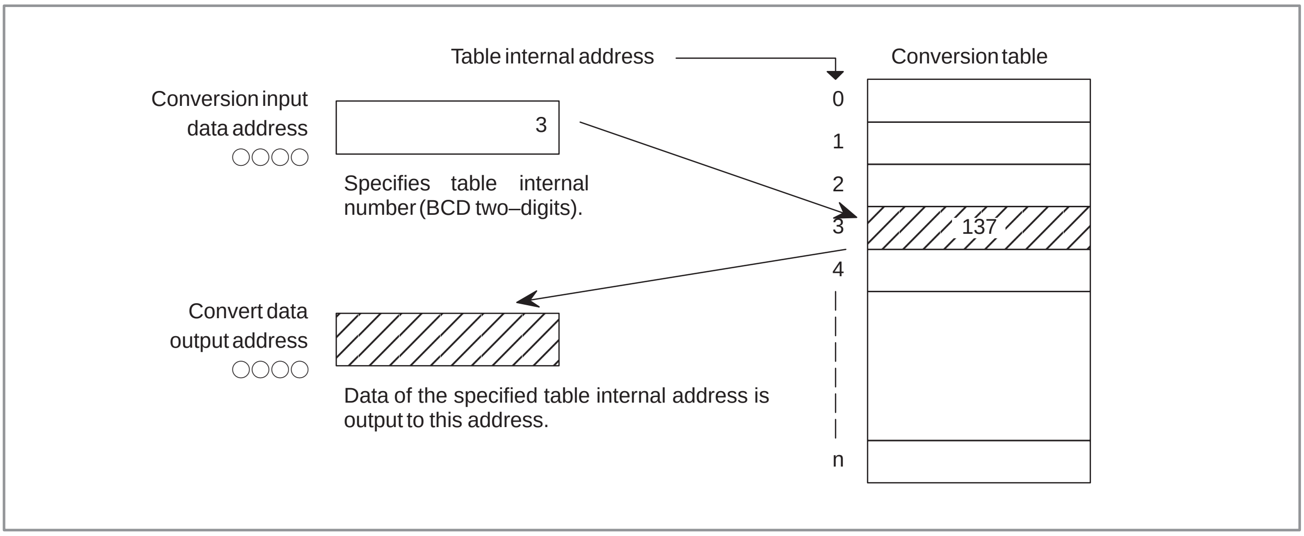


Fig. 5.13.1 Code conversion diagram

5.13.2 Format

Fig.5.13.2 shows the format for the COD instruction and Table 5.13.2 shows the coding format.

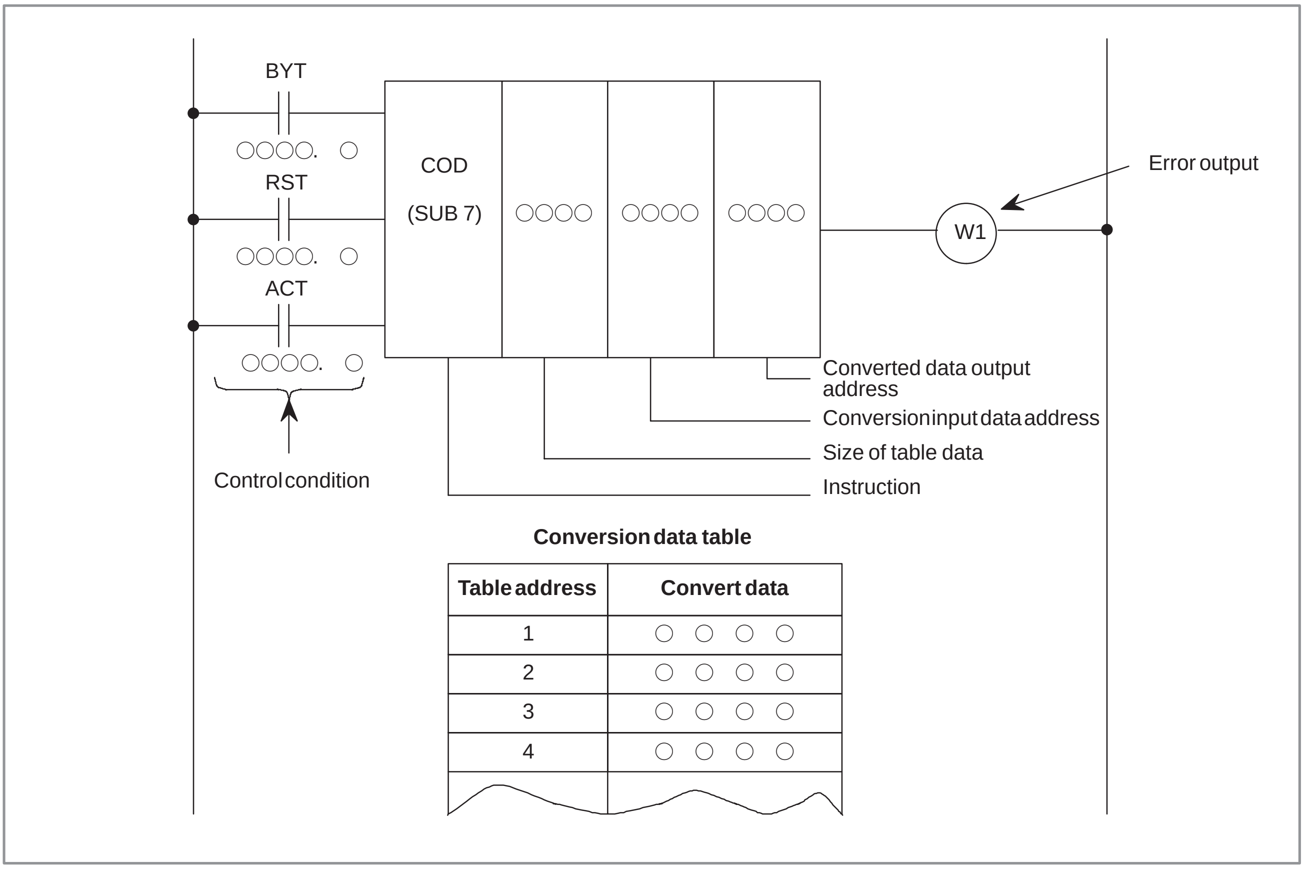


Fig. 5.13.2 COD instruction

Table 5.13.2 Coding for Fig.5.13.2

Coding sheet

Step Number	Instruc-tion	Address No.	Bit No.	Remarks
1	RD	○○○ .	○	BYT
2	RD. STK	○○○ .	○	RST
3	RD. STK	○○○ .	○	ACT
4	SUB	7		COD instruction
5	(PRM)	○○○○		Size of table data (1)
6	(PRM)	○○○○		Conversion input data address (2)
7	(PRM)	○○○○		Convert data output address (3)
8	(PRM)	○○○○		Convert data at table address 0 (4)
9	(PRM)	○○○○		Convert data at table address 1 (5)
10	:	:		:
11	WRT	○○○ .	○	Error output

Memory status of control condition

ST3	ST2	ST1	ST0
			BYT
		BYT	RST
	BYT	RST	ACT
	↓	↓	↓
			W1

5.13.3 Control Conditions

(a) Specify the data size.
- BYT=0 : Specifies that the conversion table data is to be BCD two digits.
- BYT=1 : Specifies that the conversion table data is to be BCD four digits.

(b) Error output reset
- RST=0 : Disable reset
- RST=1 : Sets error output W1 to 0 (resets).

(c) Execution command
- ACT=0 : The COD instruction is not executed. W1 does not change.
- ACT=1 : Executed.

5.13.4 Size of Table Data

A conversion table data address from 0 to 99 can be specified.
Specify n+1 as the size of table when n is the last table internal number.

5.13.5 Conversion Input Data Address

The conversion table address includes a table address in which converted data is loaded. Data in the conversion table can be retrieved by specifying a conversion table address.
One byte (BCD 2–digit) is required for this conversion input data address.

5.13.6 Convert Data Output Address

The convert data output address is the address where the data stored in the table is to be output. The convert data BCD two digits in size, requires only a 1–byte memory at the convert data output address.
Convert data BCD four digits in size, requires a 2–byte memory at the convert data output address.

5.13.7 Error Output (W1)

If an error occurs in the conversion input address during execution of the COD instruction, W1=1 to indicate an error.
For example, W1=1 results if a number exceeding the table size specified in the sequence program is specified as the conversion input address. When W1=1, it is desirable to effect an appropriate interlock, such as having the error lamp on the machine tool operator’s panel light or stopping axis feed.

5.13.8 Conversion Data Table

The size of the conversion data table is from 00 to 99.
The conversion data can be either BCD two digits or four digits, which is specified depends on the control conditions

5.14 CODB (BINARY CODE CONVERSION)

5.14.1 Function

This instruction converts data in binary format to an optional binary format 1–byte, 2–byte, or 4–byte data.
Conversion input data address, conversion table, and conversion data output address are necessary for data conversion; as shown in Fig.5.14.1.
Compared to the 5.13 ”COD Function Instruction”, this CODB function instruction handles numerical data 1–, 2– and 4–byte length binary format data, and the conversion table can be extended to maximum 256.

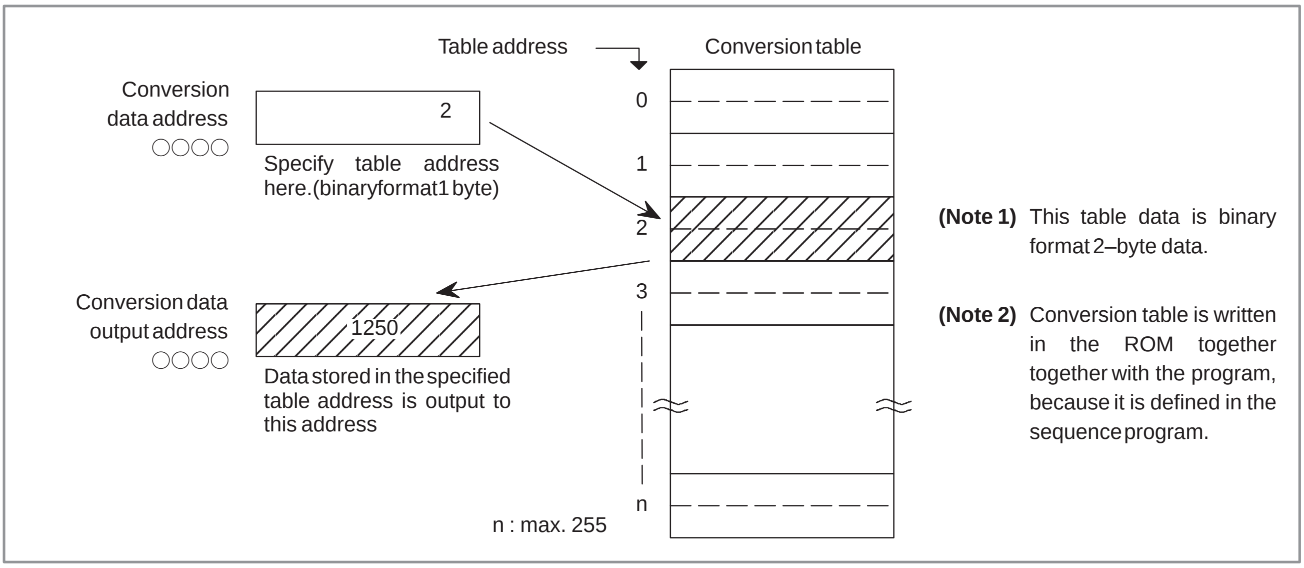


Fig. 5.14.1 Code conversion diagram

5.14.2 Format

Fig.5.14.2 shows the expression format of CODB.

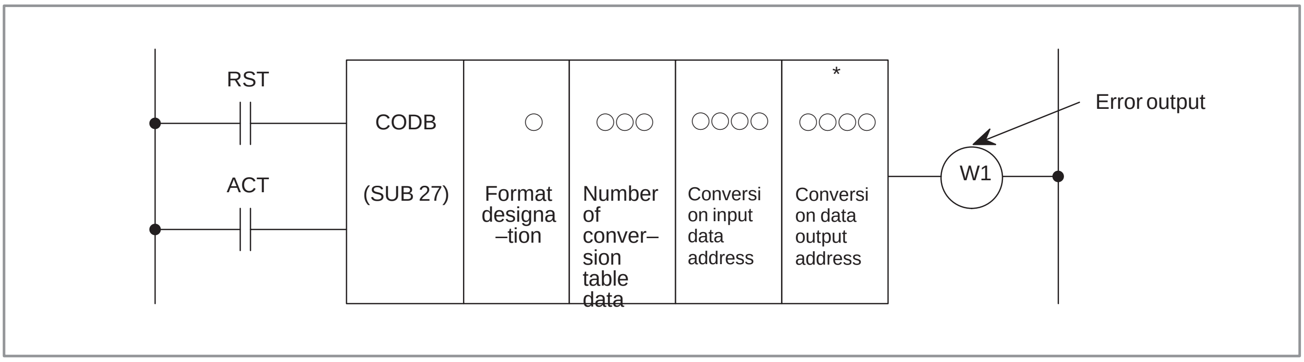


Fig. 5.14.2 Expression format of CODB

5.14.3 Control Conditions

(a) Reset (RST)
RST=0 : Do not reset.
RST=1 : Reset error output W1 (W1=0).

(b) Activate command (ACT)
ACT=0 : Do not execute CODB instruction
ACT=1 : Execute CODB instruction.

5.14.4 Parameters

(a) Format designation
Designates binary numerical size in the conversion table.
1 : Numerical data is binary 1–byte data.
2 : Numerical data is binary 2–byte data.
4 : Numerical data is binary 4–byte data.

(b) Number of conversion table data
Designates size of conversion table. 256 (0 to 255) data can be made.

(c) Conversion input data address
Data in the conversion data table can be taken out by specifying the table number. The address specifying the table number is called conversion input data address, and 1–byte memory is required from the specified address.

(d) Conversion data output address
Address to output data stored in the specified table number is called conversion data output address.
Memory of the byte length specified in the format designation is necessary from the specified address.

5.14.5 Conversion Data Table

Size of the conversion data table is maximum 256 (from 0 to 255).
This conversion data table is programmed between the parameter conversion data output address of this instruction and the error output (W1).

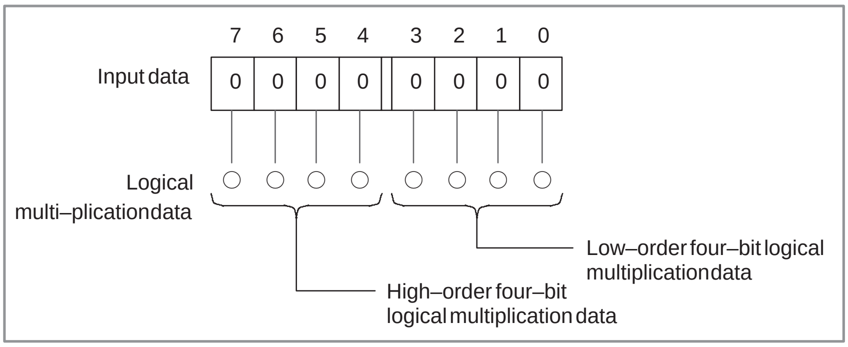
5.14.6 Error Output (W1)

If there are any abnormality when executing the CODB instruction, W1=1 and error will be output.

5.15 MOVE (LOGICAL PRODUCT TRANSFER)

5.15.1 Function

ANDs logical multiplication data and input data, and outputs the results to a specified address. Can also be used to remove unnecessary bits from an eight–bit signal in a specific address, etc.
(Logical multiplication data) (Input data) to a specified address
The input data is one byte (eight bits).



5.15.2 Format

Fig.5.15.2 shows the expression format and Table 5.15.2 shows the coding format.

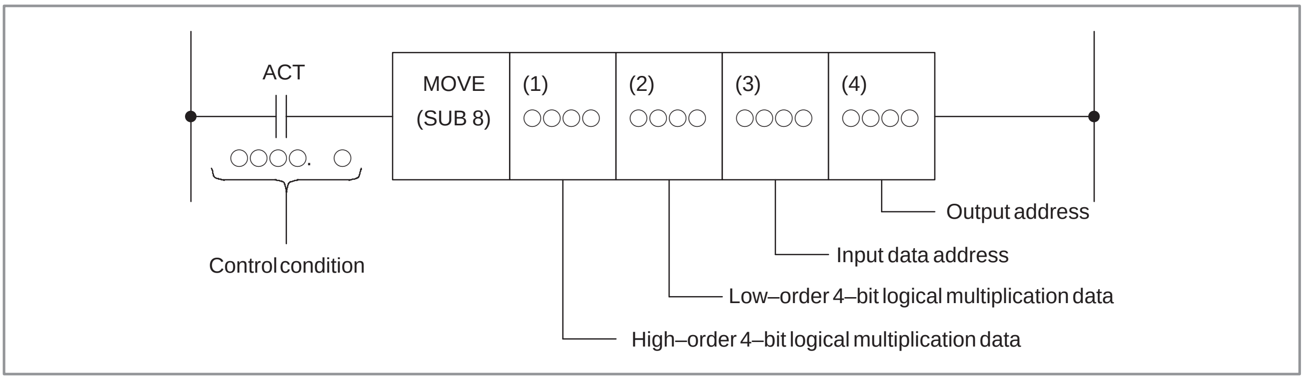


Fig. 5.15.2 Move instruction format

Table 5.15.2 Coding for Fig.5.15.2

Coding sheet

Step Number	Instruc-tion	Address No.	Bit No.	Remarks
1	RD	○○○ .	○	ACT
2	SUB	8		MOVE instruction
3	(PRM)	○○○○		High–order 4–bit logical multiplication data (1)
4	(PRM)	○○○○		Low–order 4–bit logical multiplication data (2)
5	(PRM)	○○○○		Input data address (3)
6	(PRM)	○○○○		Output data address (4)

Memory status of control condition

ST3	ST2	ST1	ST0
			ACT

5.15.3 Execution Command

ACT=0 : Move instruction not executed.
ACT=1 : Executed.

5.15.4 Example of Using the MOVE Instruction

If a code signal and another signal co–exist at address X35 for an input signal from the machine tool, to compare the code signal and a code signal at another address, the rest of signals in address X35 becomes an obstacle. Thus, the MOVE instruction can be used to output only the code signal at address X35 address R210.

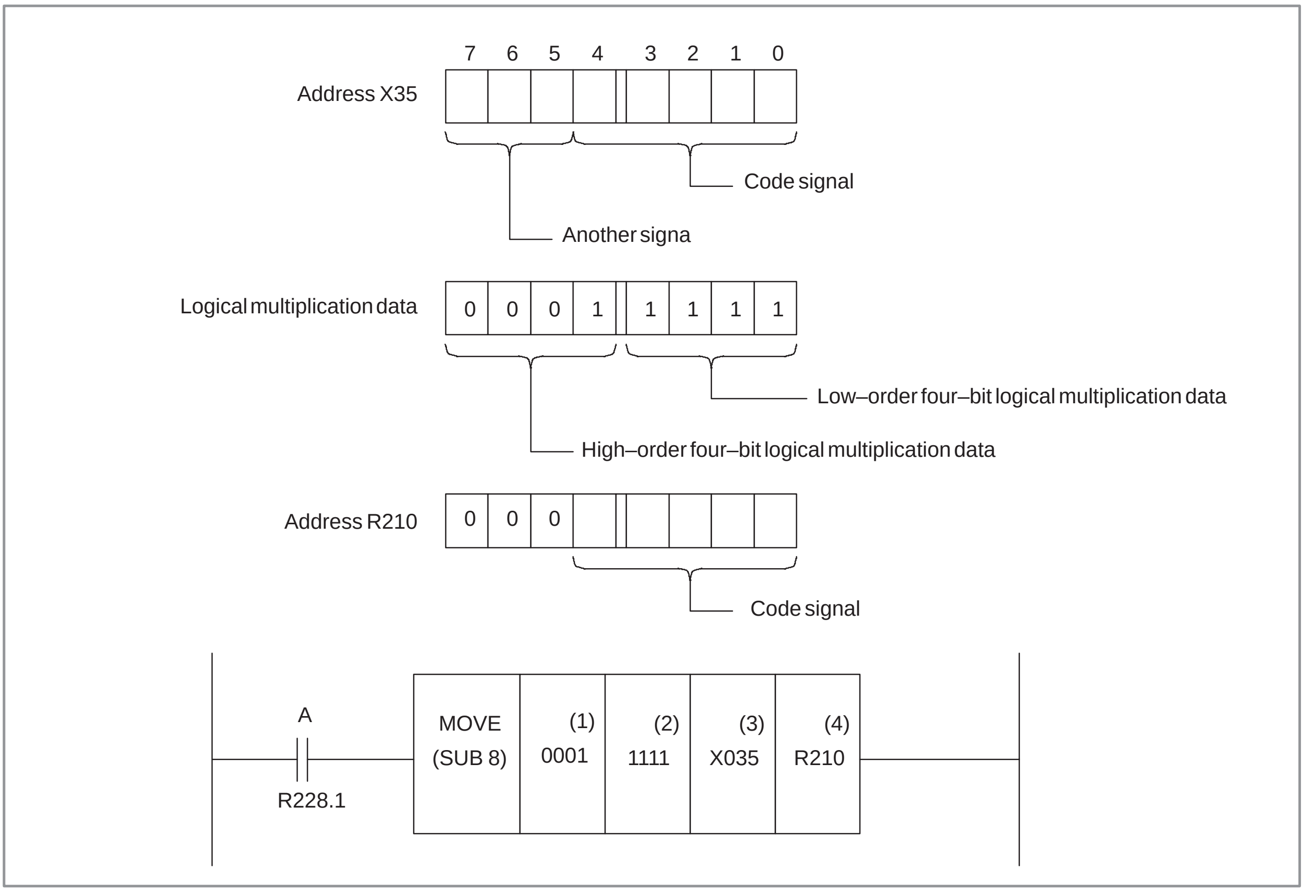
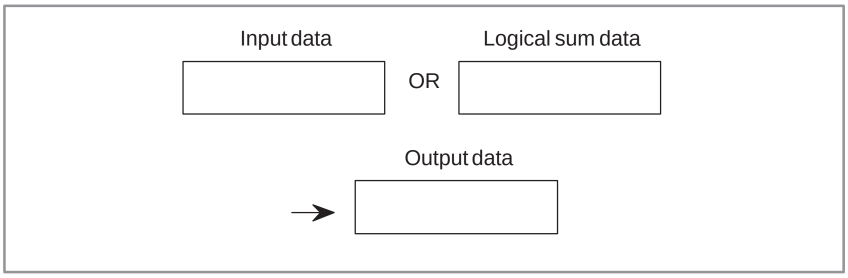


Fig. 5.15.4 MOVE instruction ladder diagram

5.16 MOVOR (DATA TRANSFER AFTER LOGICAL SUM)

5.16.1 Function

This instruction ORs the input data and the logical sum data and transfers the result to the destination.



5.16.2 Format

Fig.5.16.2 shows the expression format of MOVOR.

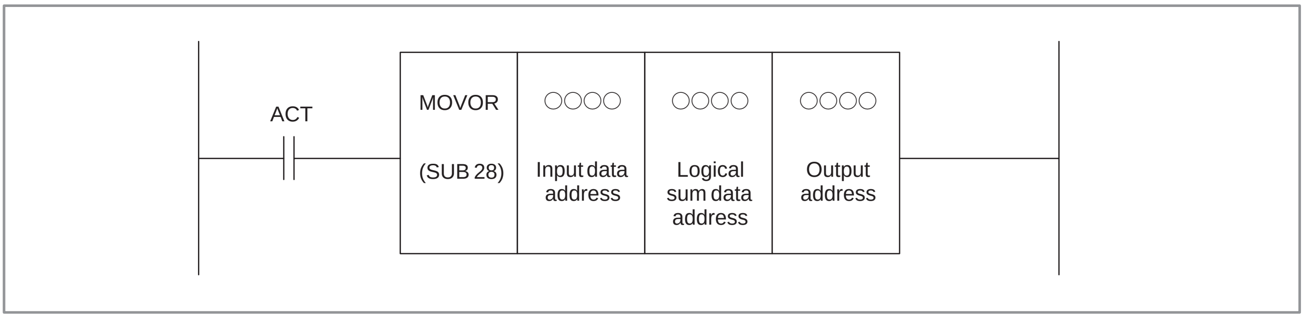


Fig. 5.16.2 Expression format of MOVOR

5.16.3 Control Conditions

(a) Command (ACT)
ACT=0: Do not execute MOVOR.
ACT=1: Execute MOVOR.

5.16.4 Parameters

(a) Input data address
Specifies the address for the input data.

(b) Logical sum data address
Specifies the address of the logical sum data with which to OR the transferred data.

(c) Output address
This is the address to contain the logical sum obtained. It is also possible to obtain the logical sum (OR) of the input and the logical sum data and output the result in the logical sum data address. For this, you must set the logical sum data address for the output address.

5.17 COM (COMMON LINE CONTROL)

5.17.1 COM (Common Line Control)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	×	×	×	×	×	○	×	×	×	×	×	○	×	×	×	×	×

5.17.1.1 Function

The specified number of coils or the coils in a region up to the common line control end instruction (COME) are turned off. (See Fig.5.17.1.1)
Relay number specification is set when a numeric other than zero is specified in a parameter for the number of turned off coils.
Specification of the region up to the common line control end instruction is set when zero is specified for the number of turned off coils.
When the common line control end instruction is programmed in the relay number specification, error is indicated when programming is completed.

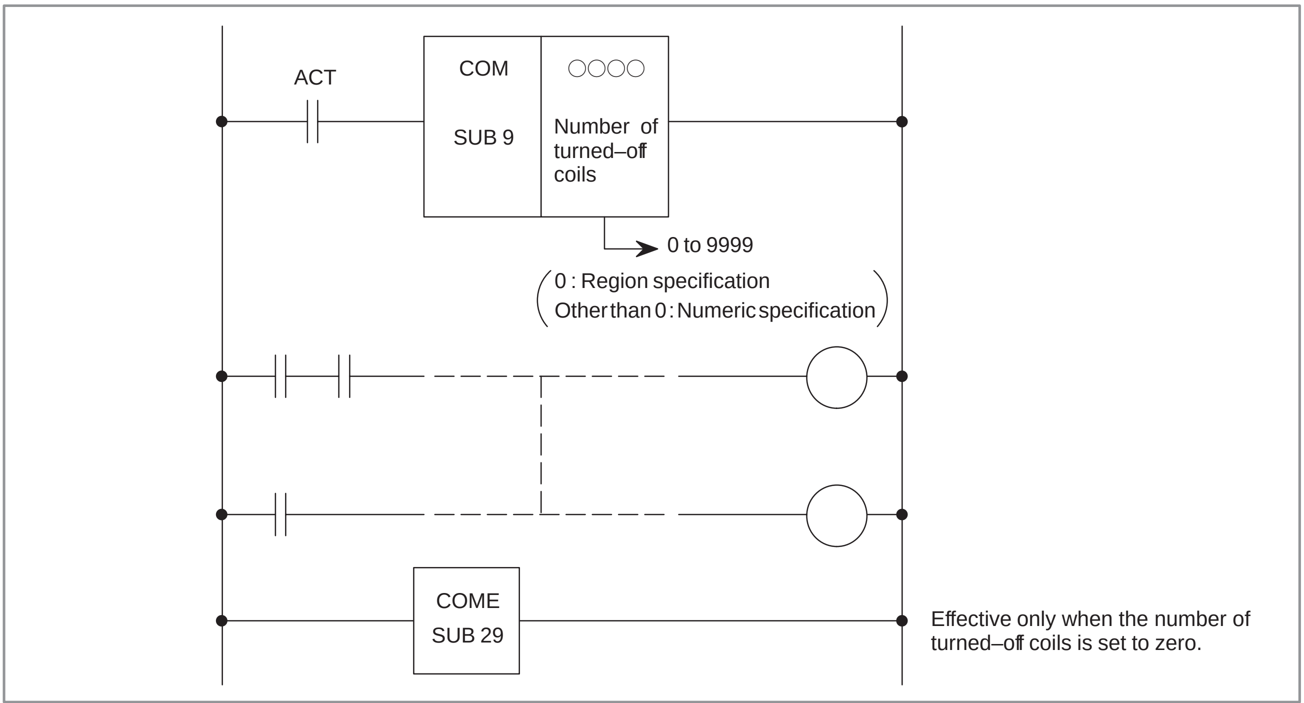


Fig. 5.17.1.1 Function of COM

5.17.1.2 Format

Fig.5.17.1.2 (a) shows the expression format of COM

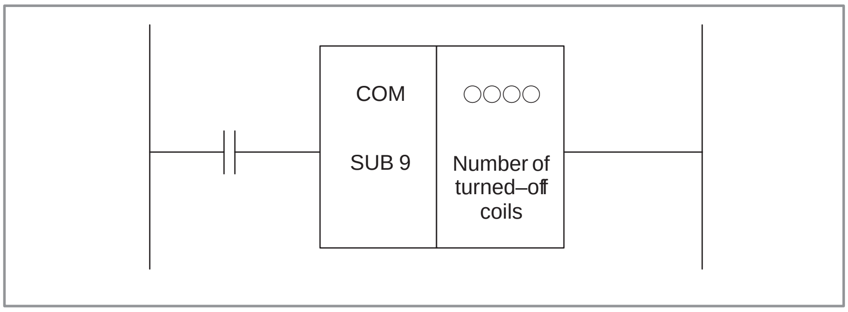


Fig. 5.17.1.2 (a) Expression format of COM

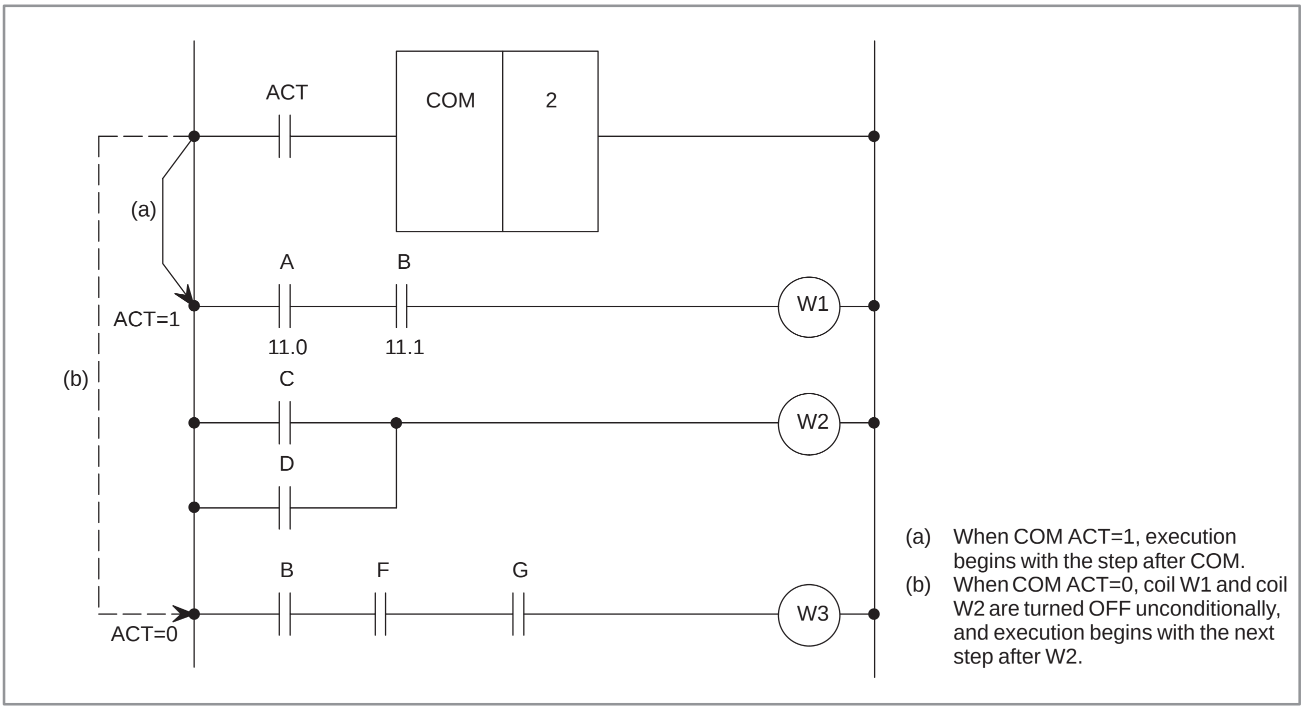


Fig. 5.17.1.2 (b) Ladder diagram for the COM instruction

5.17.2 Control Conditions

ACT=0 : The specified number of coils or the coils within the region specified are unconditionally turned off (set to 0).

ACT=1 : No processing is performed.

Processing is performed from the step next to the COM instruction.

5.17.3 Parameter

(a) Number of turned–off coils
Specify 0 to 9999.
0 : Region specification
Other than 0: Coil number specification

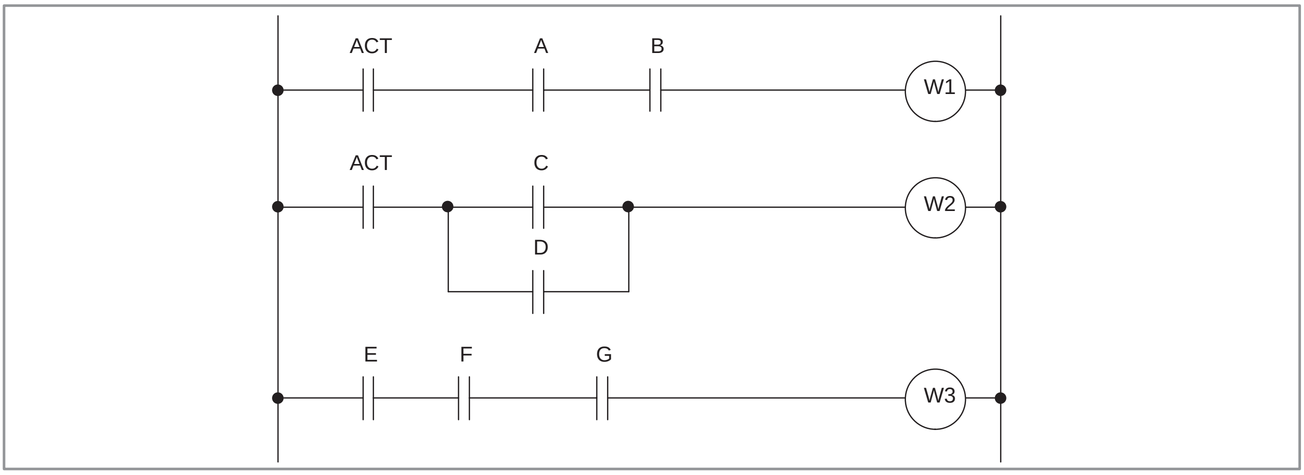


Fig. 5.17.3 (a) Relay circuit example

NOTE

1 A functional instruction in a range specified by COM executes processing, regardless of COM ACT. However, if COM ACT=0, the coil of the execution result becomes 0.
2 Another COM instruction cannot be specified in the range specified by the COM instruction.
3 If COM ACT=0, the coil written in by a WRT. NOT instruction in a range specified by COM becomes 1 unconditionally.
4 The number of coils cannot be specified in PMC–SA2, or PMC–SB2. Assume the number of coils to be 0 and specify the region with the common line control end (COME) command.

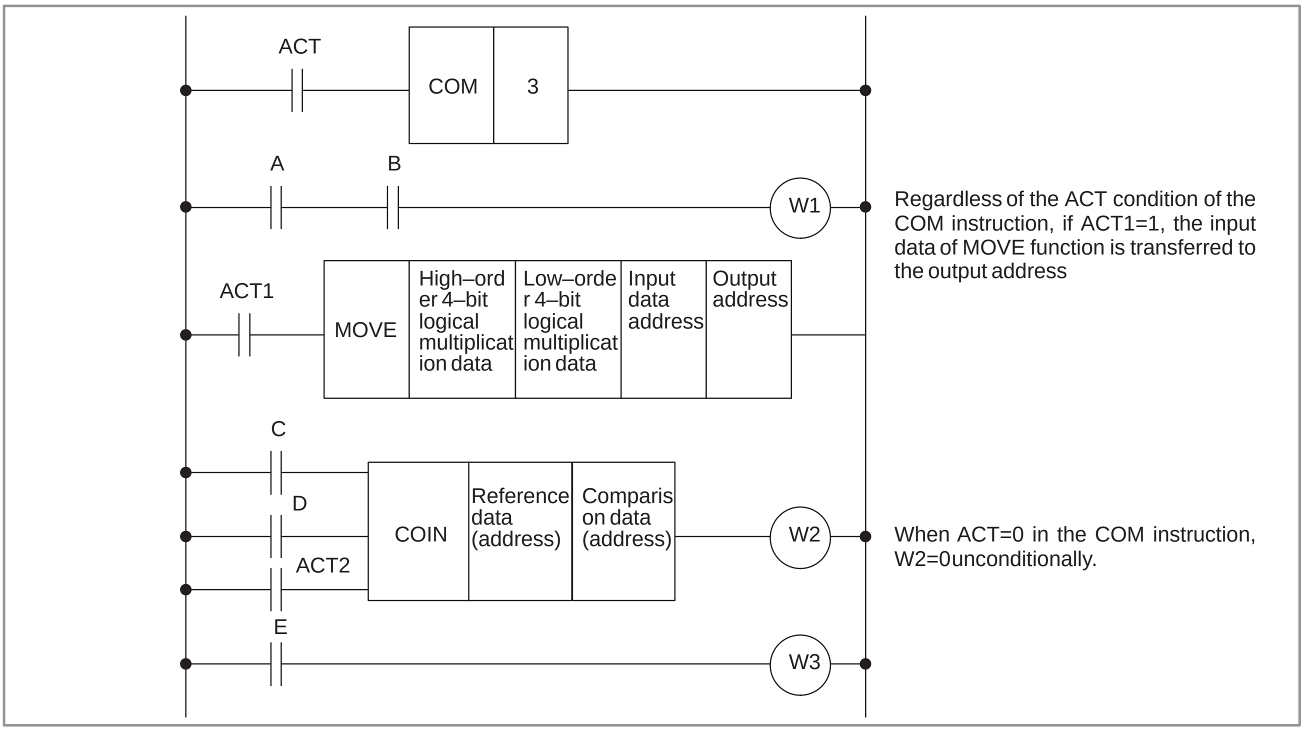


Fig. 5.17.3 (b)

5.17.4 COM (Common Line Control)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
○	○	○	○	○	○	×	○	○	○	○	○	×	○	○	○	○	○

5.17.5 Function

The COM instruction controls the coils in a range up to a common line control end instruction (COME). (See Fig.5.17.5) Specify 0 as the number of coils, and specify a range to be controlled using the common line end instruction.
When the common line end instruction is not specified, the message COM FUNCTION MISSING is displayed.

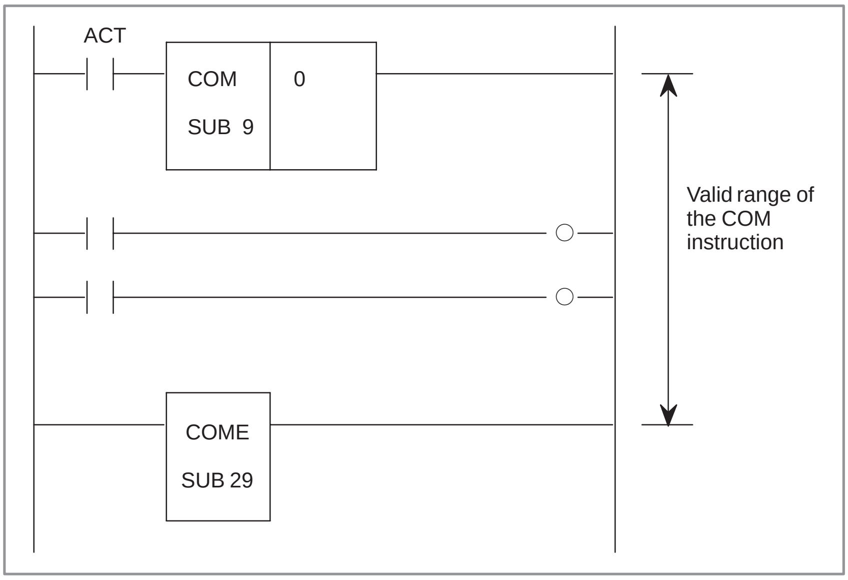


Fig. 5.17.5 Function of COM

5.17.6 Format

Fig.5.17.6 shows the expression format of the functional instruction COM.

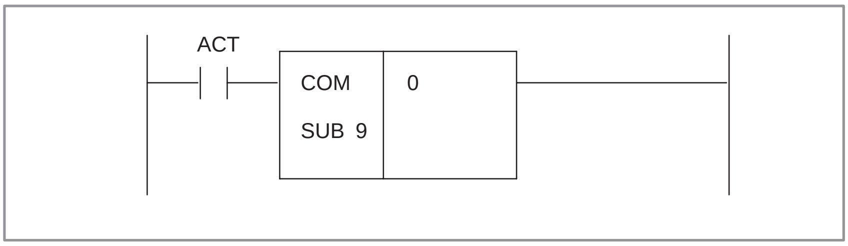


Fig. 5.17.6 Expression Format of COM

5.17.7 Control Conditions

ACT = 0 : The coils in the specified range are unconditionally turned off (set to 0).

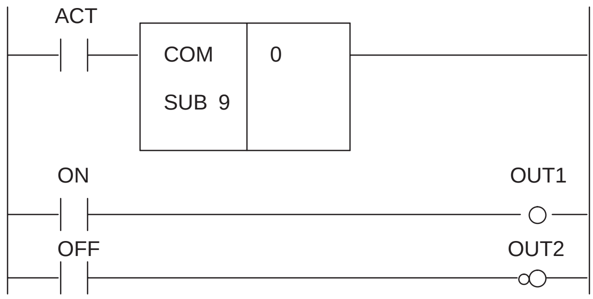
ACT = 1 : The same operation as when COM is not used is performed.

5.17.8 Parameters

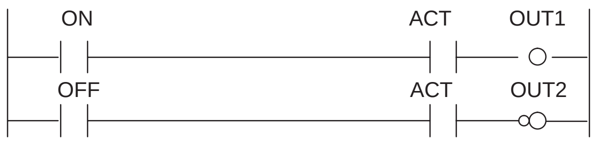
(a) Specify 0. (Range specification only)

NOTE

1 COM instruction operation
Suppose the following Ladder diagram including a COM instruction exists:



Then, for the coil "OUTx," this Ladder diagram has the same effect as the following Ladder diagram:



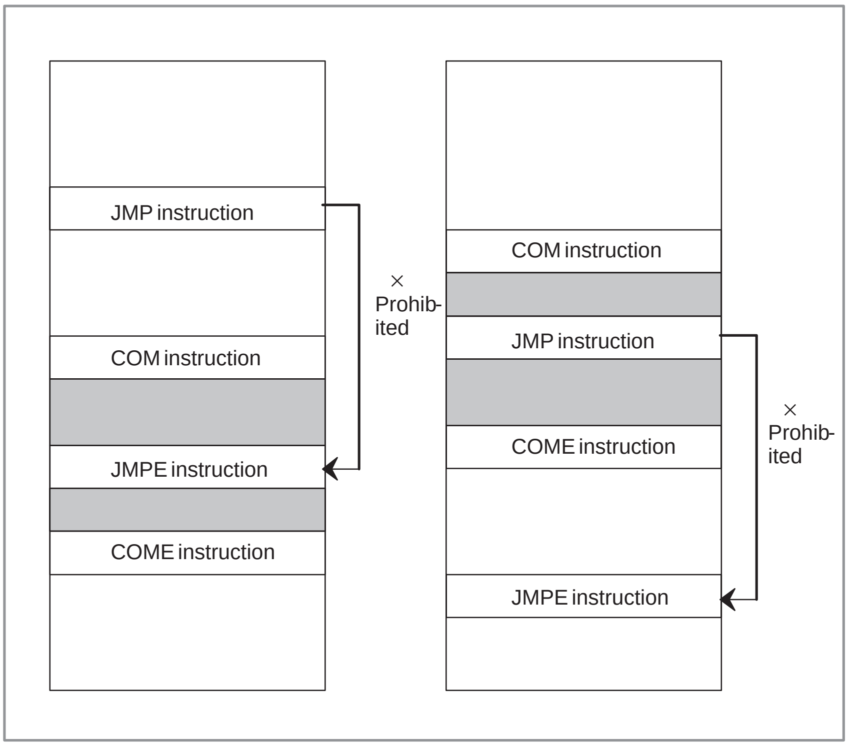
So, the functional instructions in the range specified with a COM instruction are processed, regardless of the setting of ACT of the COM instruction. Note, however, that the coil for the execution of a functional instruction is unconditionally set to 0 when COM ACT = 0.

2 In the range specified with a COM instruction, no additional COM instruction can be specified.

3 As explained in the figures in Note 1, the coil for WRT.NOT in the range specified with a COM instruction is unconditionally set to 1 when COM ACT = 0.

5.17.9 Caution

Do not create a program in which a combination of JMP and JMPE instructions is used to cause a jump to and from a sequence between the COM and COME instructions; the ladder sequence may not be able to operate normally after the jump.



5.18 COME (COMMON LINE CONTROL END)

5.18.1 Function

This instruction indicates the division in the region specification of the common line control instruction (COM).
This instruction cannot be used alone. It must he used together with the COM instruction.

5.18.2 Format

Fig.5.18.2 shows the expression format of COME

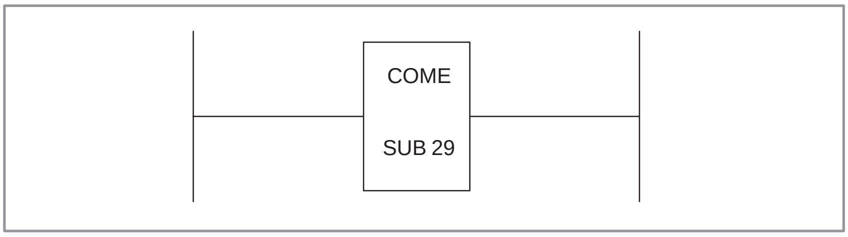


Fig. 5.18.2 Expression format of COME

5.19 JMP (JUMP)

5.19.1 JMP (Jump)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	×	×	×	×	×	○	×	×	×	×	×	○	×	×	×	×	×

5.19.2 Function

This instruction jumps the specified number of coils or the logic instructions (including the functional instructions) contained within the region up to the jump end instruction (JMPE).
Coil number specification is set when a numeral other than zero is specified in the parameter for the number of coils.
Specification of the region up to the jump end instruction is set when zero is set for the number of coils. Nesting of jump instructions is not allowed.

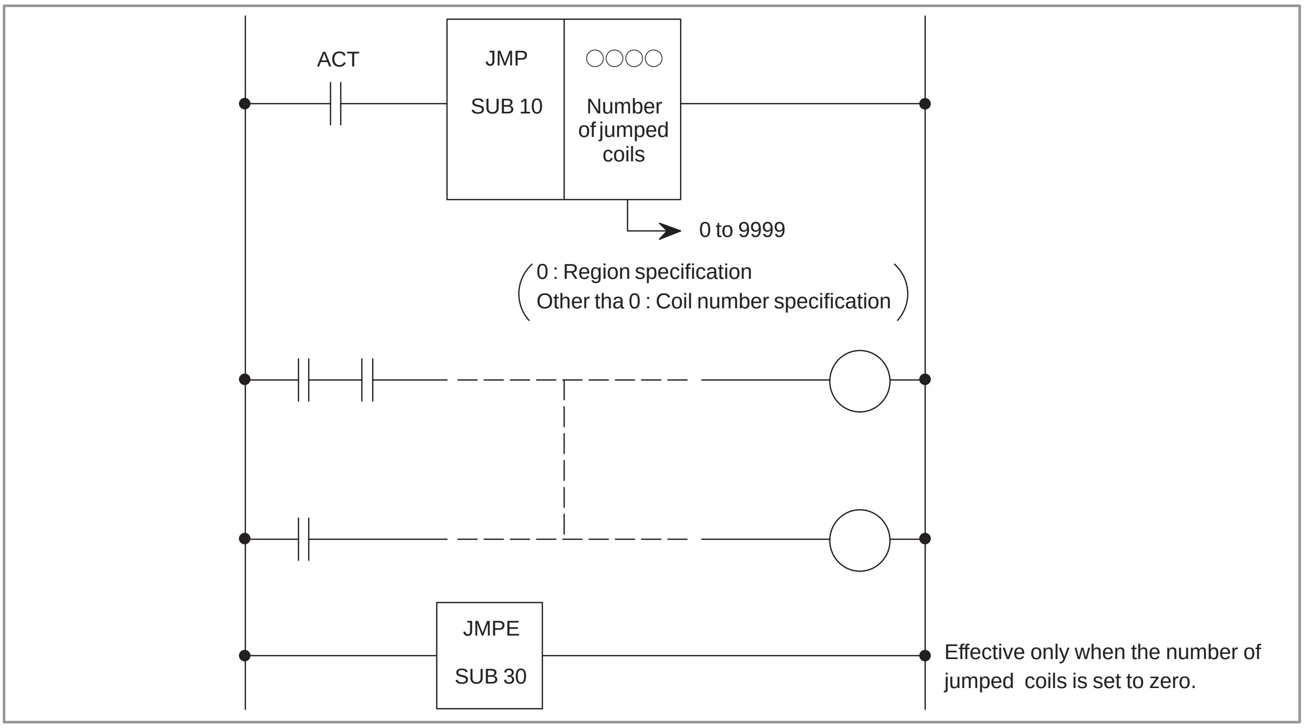
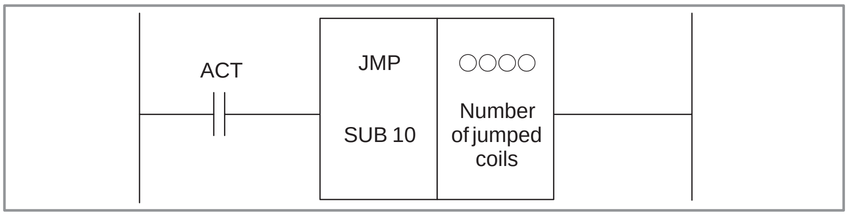


Fig. 5.19.2

5.19.3 Format



5.19.4 Control Conditions

ACT=0 : Nojump.Processing begins with the step after the JMP instruction.
ACT=1 : The logic instructions contained within the specified number of coils or the specified region are jumped. Processing is performed from the next step.

5.19.5 Parameter

(a) Number of jumped coils

Specify 0 to 9999.
0 : Region specification jump
Other than 0 : Coil number specification jump
When the jump end instruction is programmed in the coil number specification, error is indicated when programming is completed.

Table 5.19.5 JMP instruction coding

Step Number	Instruction	Address Number	Bit Number	Remarks
1	RD	○○○.	○	ACT
2	SUB	10		JMP instruction
3	(PRM)	○○○○		Number of coils to be jumped

NOTE
The number of coils can be specified only for the PMC–SB/SC. Assume the number of coils to be 0 and specify the region with the jump end (JMPE) command.

5.19.6 Operation

Fig.5.19.6 shows a ladder diagram for the JMP instruction. When ACT=0, the next step to the JMP instruction is executed. When ACT=1, logical operations are skipped according to the specified number of coils. Note that, when ACT=1, even if signal A changes from 1 to 0 or vice versa as shown in Fig.5.19.6, W1 remains in a status before ACT=1. Similarly, W2 remains unchanged, even if signals B, C, and D change. If a sequence is executed in ladder split mode, even the use of the JMP instruction does not reduce the execution time of the sequence (see Section I.2.3, “Processing Priority”).

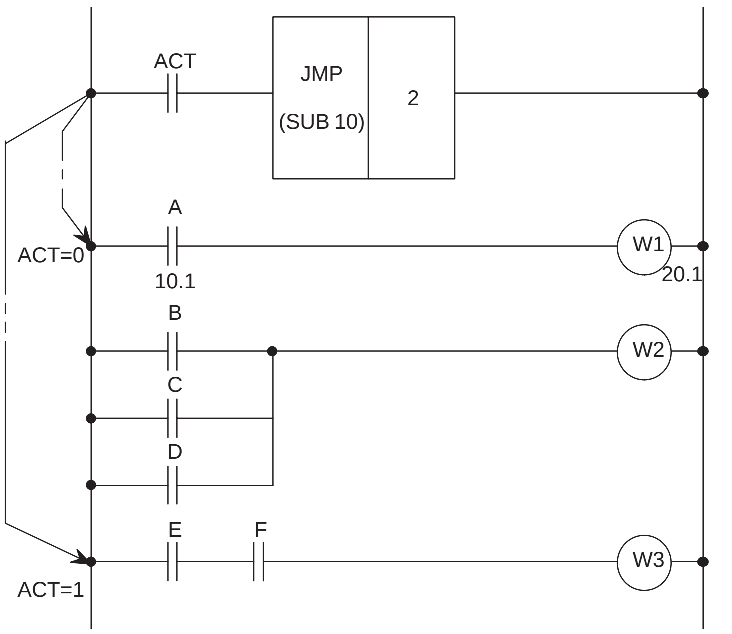


Fig. 5.19.6 Ladder diagram for the JMP instruction

5.19.7 JMP (Jump)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
○	○	○	○	○	○	×	○	○	○	○	○	×	○	○	○	○	○

5.19.7.1 Function

The JMP instruction causes a departure from the normal sequence to executing instructions. When a JMP instruction is specified, processing jumps to a jump and instruction (JMPE) without executing the logical instructions (including functional instructions) in the range delimited by a jump end instruction (JMPE). (See Fig.5.19.7.1) Specify 0 as the number of coils, and specify a range to be skipped using the jump end instruction.
When the jump end instruction is not specified, the message JUMP FUNCTION MISSING is displayed.

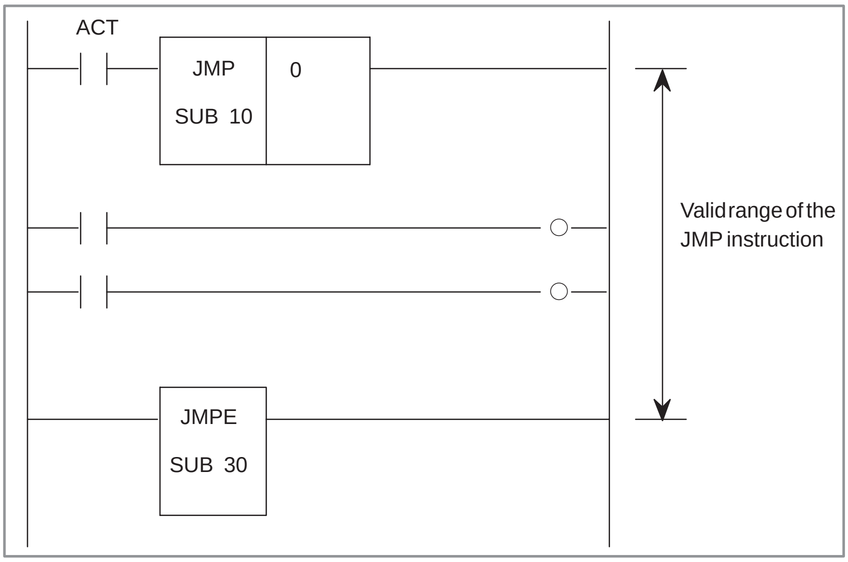


Fig. 5.19.7.1 Function of JMP

5.19.7.2 Format

Fig.5.19.7.2 shows the expression format of the functional instruction JMP.

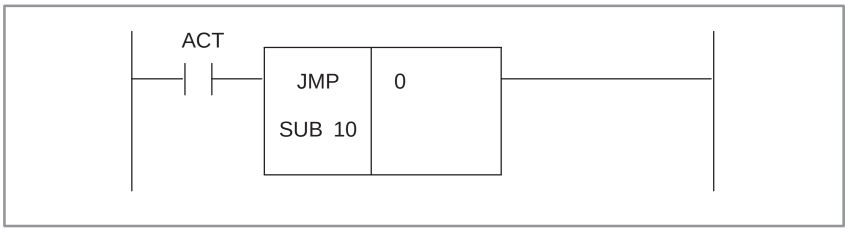


Fig. 5.19.7.2 Expression format of JMP

5.19.7.3 Control Conditions

ACT=1 : The logical instructions (including functional instructions) in the specified range are skipped; program execution proceeds to the next step.

ACT=0 : The same operation as when JMP is not used is performed.

5.19.7.4 Parameters

(a) Specify 0. (Range specification only)

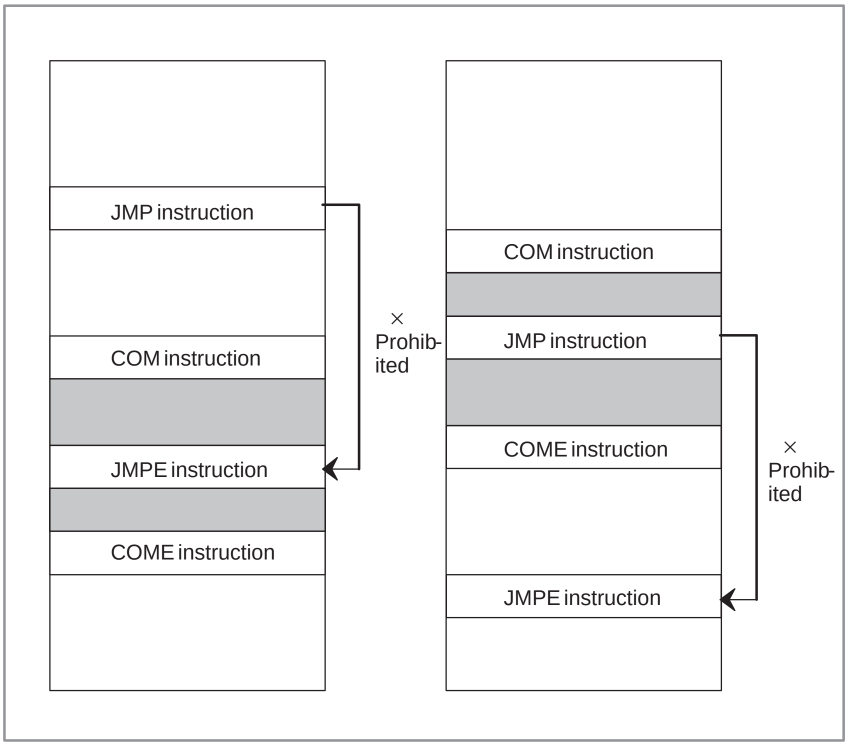
NOTE

JMP instruction operation

When ACT = 1, processing jumps to a jump end instruction (JMPE); the logical instructions (including functional instructions) in the specified jump range are not executed. When the Ladder program is executed in the nonseparate mode, this instruction can reduce the Ladder execution period (scan time).

5.19.8 Caution

Do not create a program in which a combination of JMP and JMPE instructions is used to cause a jump to and from a sequence between the COM and COME instructions; the ladder sequence may not be able to operate normally after the jump.

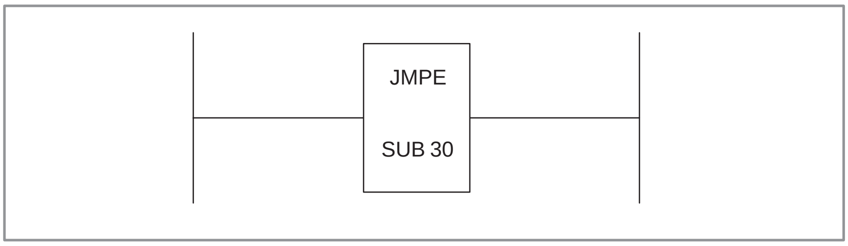


5.20 JMPE (JUMP END)

5.20.1 Function

This instruction indicates the division in the region specification of the jump instruction (JMP).
It cannot be used alone. It must be used together with the JMP instruction.

5.20.2 Format



5.21 PARI (PARITY CHECK)

5.21.1 Function

Checks the parity of code signals, and outputs an error if an abnormality is detected. Secifies either an even– or odd–parity check. Only one–byte (eight bits) of data can be checked.

5.21.2 Format

Fig.5.21.2 shows the expression format and Table 5.21.2 shows the coding format.

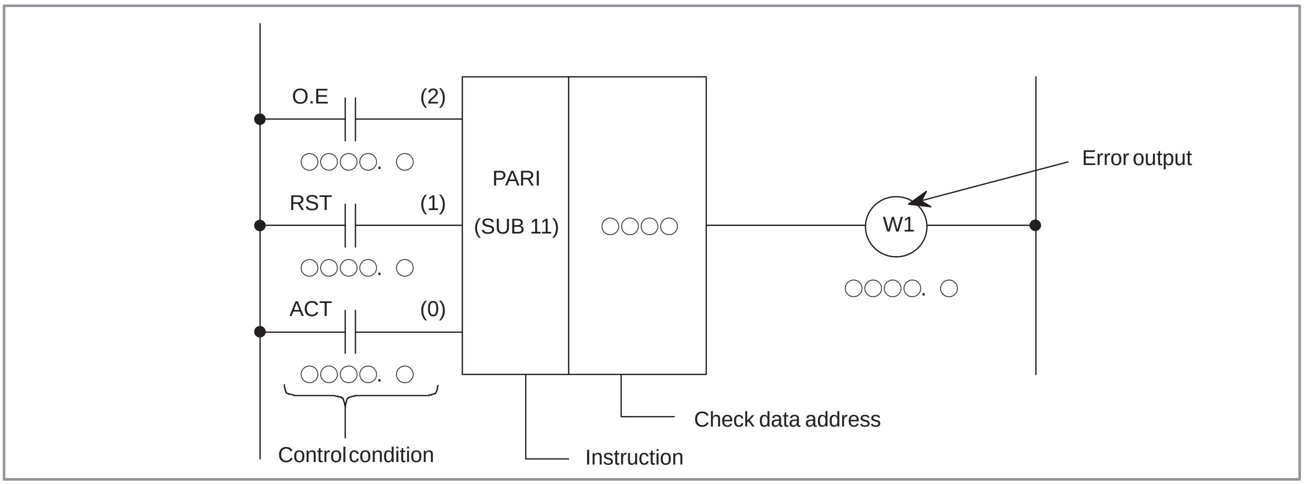


Fig. 5.21.2 PARI instruction format

Table 5.21.2 PARI instruction coding

Coding sheet

Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	○○○.	○	ACT
2	RD. STK	○○○.	○	ACT
3	RD. STK	○○○.	○	ACT
4	SUB	11		PARI instruction
5	(PRM)	○○○○		Check data address
6		○○○○.	○	Error output

Memory status of control condition

ST3	ST2	ST1	ST0
			O.E
		O.E	RST
	O.E	RST	ACT
			W1

5.21.3 Control Conditions

(a) Specify even or odd.
O.E=0 : Even–parity check
O.E=1 : Odd–parity check

(b) Reset
RST=0 : Disables reset.
RST=1 : Sets error output W1 to 0. That is, when a parity error occurs, setting RST to 1 results in resetting.

(c) Execution command
ACT=0 : Parity checks are not performed. W1 does not alter.
ACT=1 : Executes the PARI instruction, performing a parity check.

5.21.4 Error Output (W1)

If the results of executing the PARI instruction is abnormal, W1=1 and an error is posted. The W1 address can be determined arbitrarily.

5.21.5 Example of Using the PARI Instruction

Fig.5.21.5 shows odd–parity checking of a code signal entered at address X036.

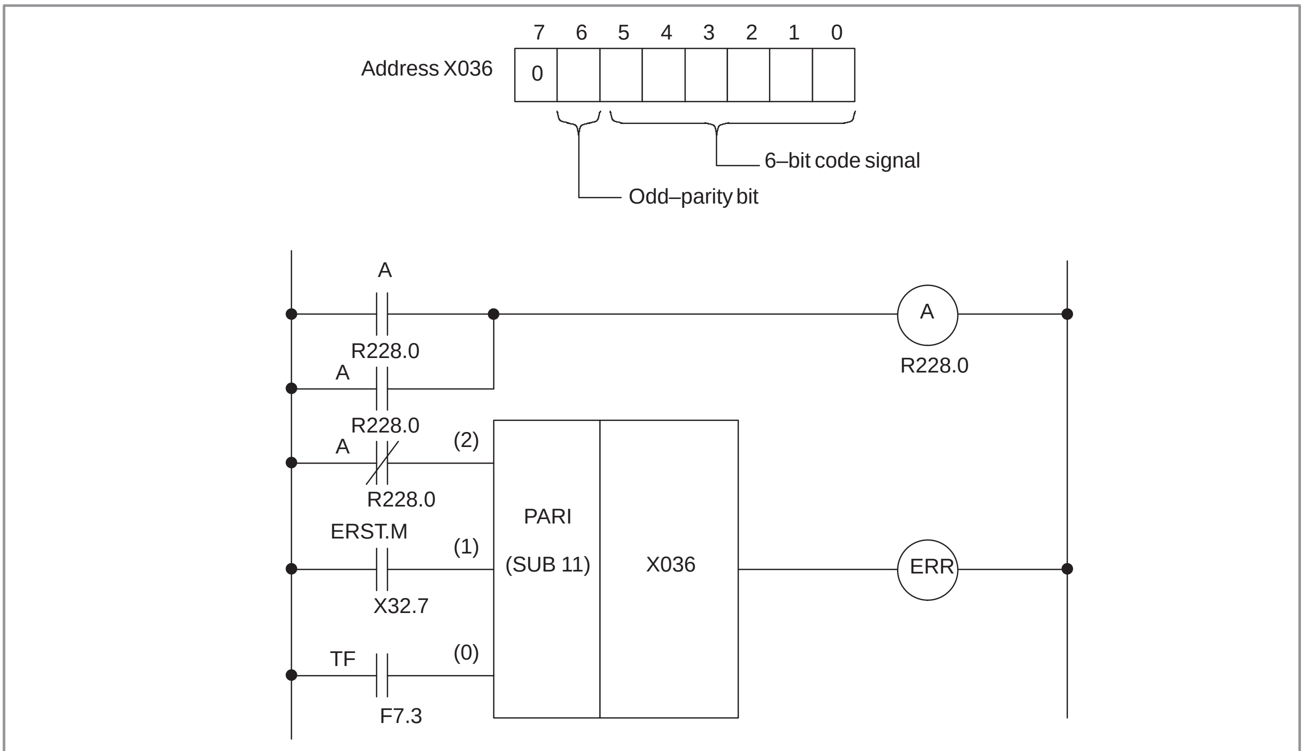


Fig. 5.21.5 Ladder diagram for the PARI instruction

> **NOTE**
> For bits 0 to 7, bits other than those for the parity check must be 0.

5.22 DCNV (DATA CONVERSION)

5.22.1 Function

Converts binary–code into BCD–code and vice versa.

5.22.2 Format

Fig.5.22.2 shows the expression format and Table 5.22.2 shows the coding format.

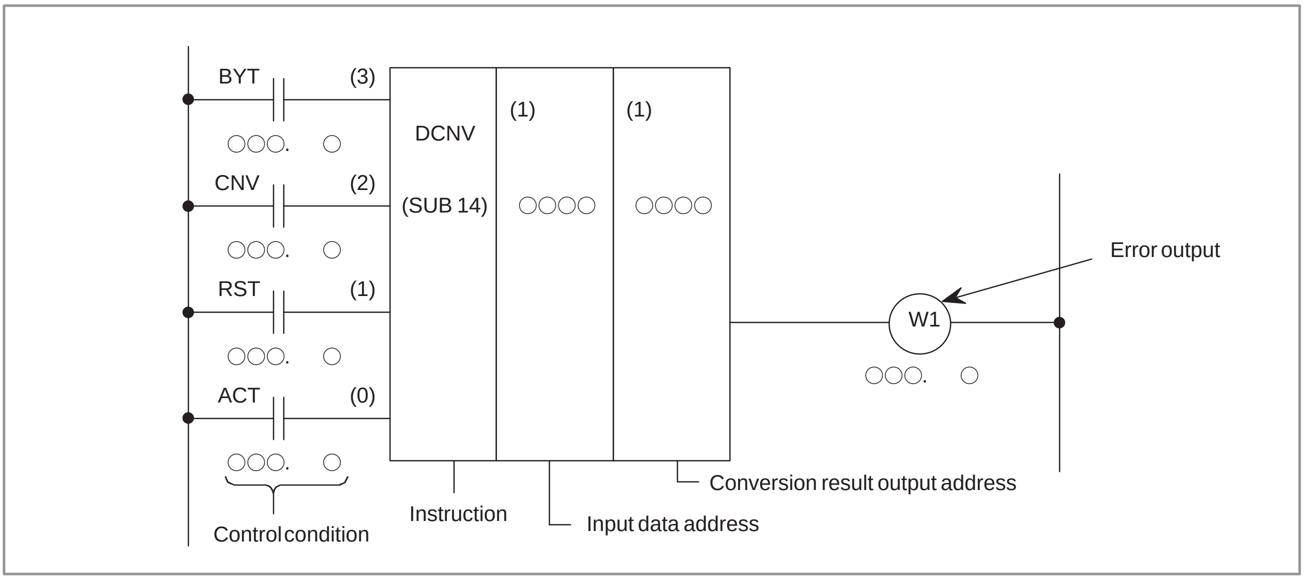


Fig. 5.22.2 DCNV instruction format

Table 5.22.2 DCNV instruction coding

Coding sheet

Step Number	Instruc-tion	Address No.	Bit No.	Remarks
1	RD	○ ○○○ .	○	BYT
2	RD. STK	○○○ .	○	CNV
3	RD. STK	○○○ .	○	RST
4	RD. STK	○○○ .	○	ACT
5	SUB	14		DCNV instruction
6	(PRM)	○○○○		(1) Input data address
7	(PRM)	○○○○		(2) Conversion result output address
8	WRT	○○○ .	○	W1 error output

Memory status of control condition

ST3	ST2	ST1	ST0
			BYT
		BYT	CNV
	BYT	CNV	RST
BYT	CNV	RST	ACT
↓	↓	↓	↓
			W1

5.22.3 Control Conditions

(a) Specify data size.
BYT=0 : Process data in length of one byte (8 bits)
BYT=1 : Process data in length of two byte (16 bits)

(b) Specify the type of conversion
CNV=0 : Converts binary–code into BCD–code.
CNV=1 : Converts BCD–code into binary–code.

(c) Reset
RST=0 : Disables reset.
RST=1 : Resets error output W1. That is, setting RST to 1 when W1, makes W1=0.

(d) Execution command
ACT=0 : Data is not converted. W1 does not alter.
ACT=1 : Data is converted.

5.22.4 Error Output (W1)

W1=0 : Normal
W1=1 : Conversion error
W1=1 if the input data which should be BCD data, is binary data, or if the data size (byte length) specified in advance is exceeded when converting binary data into BCD data.

5.23 DCNVB (EXTENDED DATA CONVERSION)

5.23.1 Function

This instruction converts 1, 2, and 4–byte binary code into BCD code or vice versa. To execute this instruction, you must preserve the necessary number of bytes in the memory for the conversion result output data.

5.23.2 Format

Fig.5.23.2 shows the expression format of DCNVB

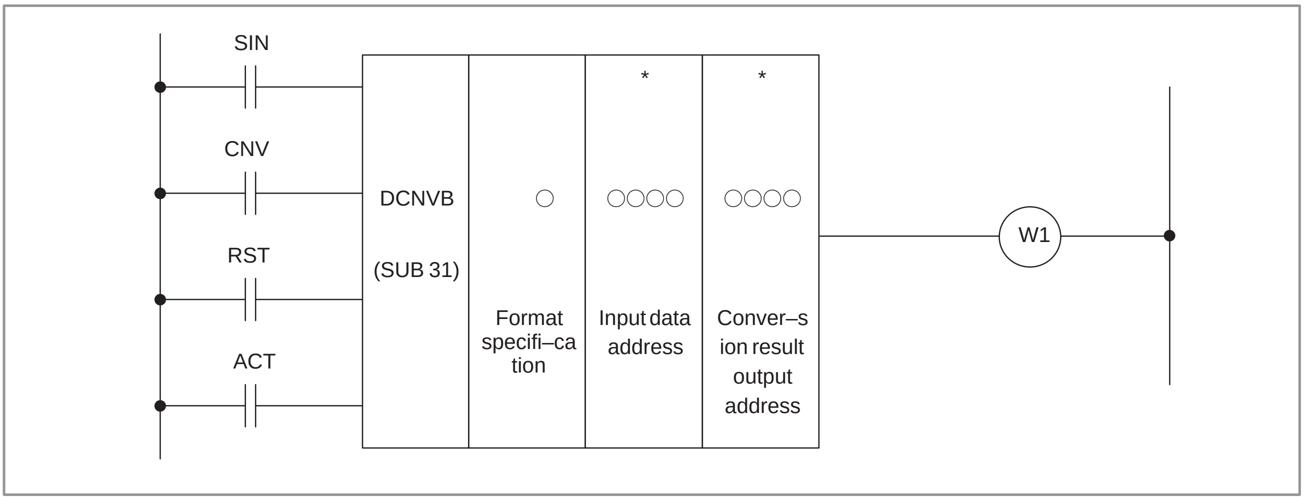


Fig. 5.23.2 Expression format of DCNVB

5.23.3 Control Conditions

(a) Sign of the data to be converted (SIN)
This parameter is significant only when you are converting BCD data into binary coded data. It gives the sign of the BCD data.
Note that though it is insignificant when you are converting binary into BCD data, you cannot omit it.
SIN=0 : Data (BCD code) to be input is positive.
SIN=1 : Data (BCD code) to be input is negative.

(b) Type of conversion (CNV)
CNV=0 : Convert binary data into BCD data
CNV=1 : Convert BCD data into binary data.

(c) Reset (RST)
RST=0 : Release reset
RST=1 : Reset error output W1. In other words, set W1=0.

(d) Execution command (ACT)
ACT=0 : Data is not converted. The value of W1 remains unchanged.
ACT=1 : Data is converted.

5.23.4 Parameters

(a) Format specification
Specify data length (1,2, or 4 bytes).
Use the first digit of the parameter to specify byte length.
1 : one byte
2 : two bytes
4 : four bytes

(b) Input data address
Specify the address containing the input data address.

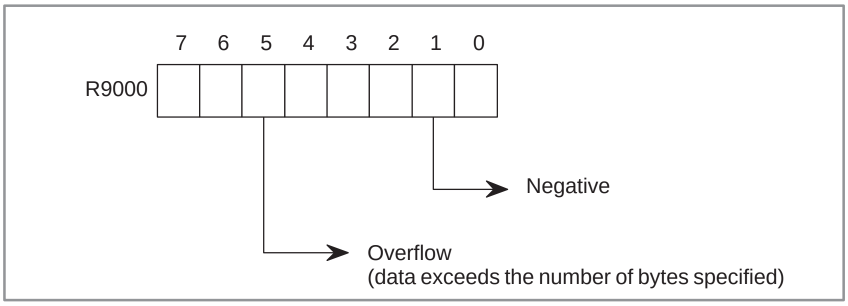
(c) Address for the conversion result output
Specify the address to output the data converted to BCD or binary format.

5.23.5 Error Output (W1)

W1=0 : Correct conversion
W1=1 : Abnormally
(The data to be converted is specified as BCD data but is found to be binary data, or the specified number of bytes cannot contain (and hence an overflow occurs) the BCD data into which a binary data is converted.)

5.23.6 Operation Output Register (R9000)

This register is set with data on operation. If register bit 1 is on, they signify the following.
For the positive/negative signs when binary data is converted into BCD data, see R9000.



5.24 COMP (COMPARISON)

5.24.1 Function

Compares input and comparison values.

5.24.2 Format

Fig.5.24.2 shows the expression format and Table 5.24.2 shows the coding format.

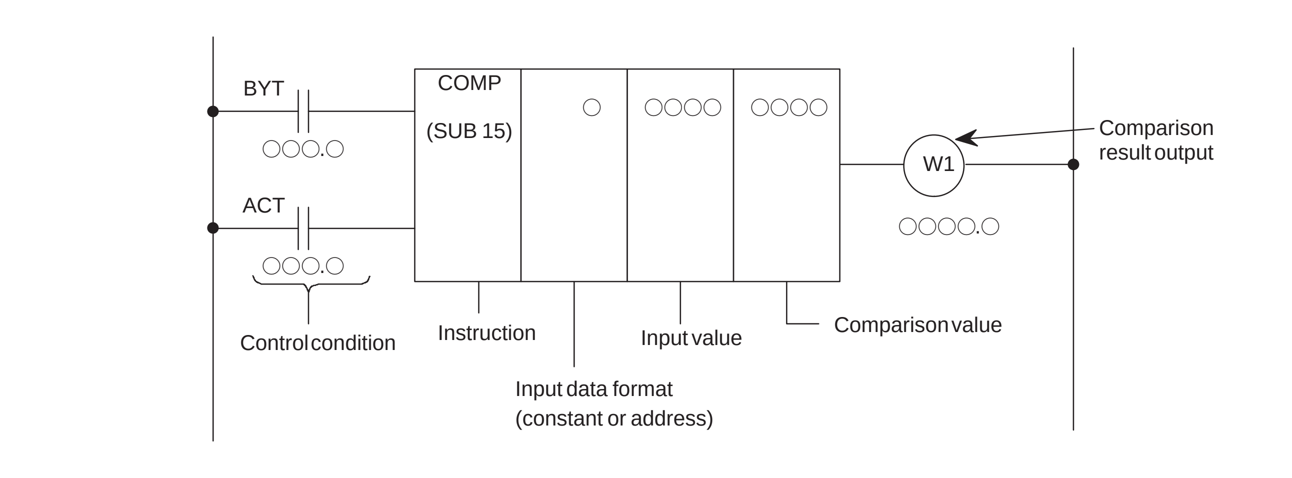


Fig. 5.24.2 COMP instruction format

Table 5.24.2 COMP instruction coding

Coding sheet

Step Number	Instruc-tion	Address No.	Bit No.	Remarks
1	RD	○○○ .	○	BYT
2	RD. STK	○○○ .	○	ACT
3	SUB	15		COMP instruction
4	(PRM)	○		Input data format
5	(PRM)	○○○○		Input data
6	(PRM)	○○○○		Comparison data address
7	WRT	○○○ .	○	W1: Comparison result output

Memory status of control condition

ST3	ST2	ST1	ST0
			BYT
		BYT	ACT
		↓	↓
			W1

5.24.3 Control Conditions

(a) Specify the data size.
BYT=0 : Process data (input value and comparison value) is BCD two digits long.
BYT=1 : Process data (input value and comparison value) is four digits long.

(b) Execution command
ACT=0 : The COMP instruction is not executed. W1 does not alter.
ACT=1 : The COMP instruction is executed and the result is output to W1.

5.24.4 Input Data Format

0 : Specifies input data with a constant.
1 : Specifies input data with an address
Not specify input data directly, but specify an address storing input data.

5.24.5 Input Data

The input data can be specified as either a constant or the address storing it. The selection is made by a parameter of format specification.

5.24.6 Comparison Data Address

Specifies the address storing the comparison data.

5.24.7 Comparison Result Output

W1=0 : Reference data > Comparison data
W1=1 : Reference data ≦ Comparison data

5.25 COMPB (COMPARISON BETWEEN BINARY DATA)

5.25.1 Function

This instruction compares 1, 2, and 4–byte binary data with one another. Results of comparison are set in the operation output register (R9000). Sufficient number of bytes are necessary in the memory to hold the input data and comparison data.

5.25.2 Format

Fig.5.25.2 shows the expression format of COMPB.

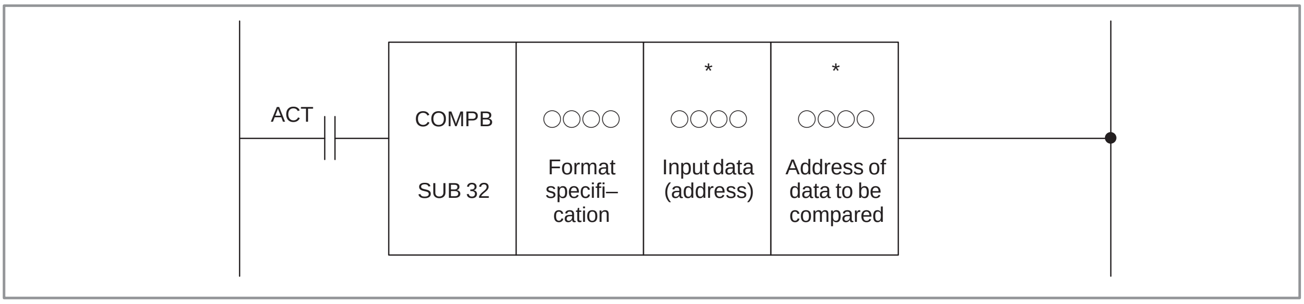


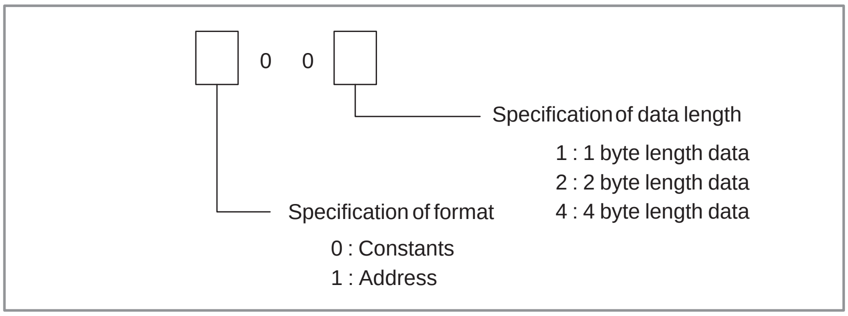
Fig. 5.25.2 Expression format of COMPB

5.25.3 Control Conditions

(a) Command (ACT)
ACT=0 : Do not execute COMPB.
ACT=1 : Execute COMPB.

5.25.4 Parameters

(a) Format specification
Specify data length (1,2, or 4 bytes) and format for the input data ('constants data' or 'address data').

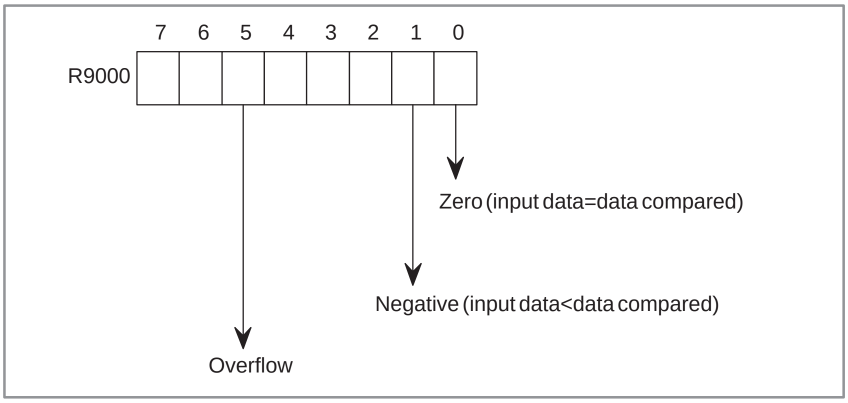


(b) Input data (address)
Format for the input data is determined by the specification in a).

(c) Address of data to be compared
Indicates the address in which the comparison data is stored.

5.25.5 Operation Output Register (R9000)

The data involved in the operation are set in this register. This register is set with data on operation. If register bit 1 is on, they indicate the following:



5.26 COIN (COINCIDENCE CHECK)

5.26.1 Function

Checks whether the input value and comparison value coincide.
This instruction is available with BCD data.

5.26.2 Format

Fig.5.26.2 shows the expression format and Table 5.26.2 shows the coding format.

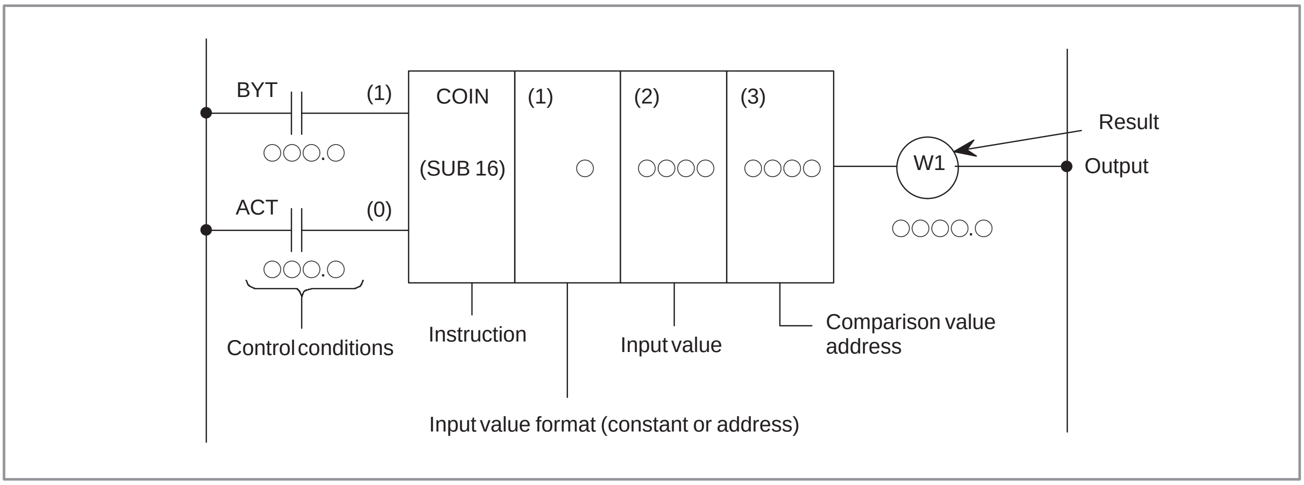


Fig. 5.26.2 COIN instruction format

Table 5.26.2 COIN instruction coding

Coding sheet

Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	○○○ .	○	BYT
2	RD. STK	○○○ .	○	ACT
3	SUB	16		COIN instruction
4	(PRM)	○		Reference value format
5	(PRM)	○○○○		Reference value
6	(PRM)	○○○○		Comparison value address
7	WRT	○○○ .	○	W1: Checking result output

Memory status of control condition

ST3	ST2	ST1	ST0
			BYT
		BYT	ACT
			W1

5.26.3 Control Conditions

(a) Specify the data size.
BYT=0 : Process data (input value, and comparison values). Each BCD is two digits long.
BYT=1 : Each BCD four digits long.

(b) Execution command
ACT=0 : The COIN instruction is not executed. W1 does not change.
ACT=1 : The COIN instruction is executed and the results is output to W1.

5.26.4 Input Data Format.

0 : Specifies input data as a constant.
1 : Specifies input data as an address.

5.26.5 Input Data

The input data can be specified as either a constant or an address storing it. The selection is made by a parameter of format designation.

5.26.6 Comparison Data Address

Specifies the address storing the comparison data.

5.26.7 Comparison Result Output

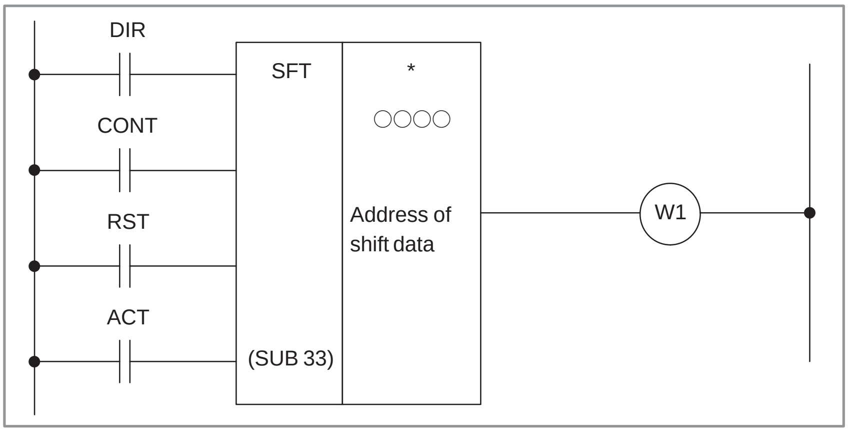
W1=0 : Input data $\neq$ Comparison data
W1=1 : Input data = Comparison data

5.27 SFT (SHIFT REGISTER)

5.27.1 Function

This instruction shifts 2–byte (16–bit) data by a bit to the left or right. Note that W1=1 when data "1" is shifted from the left extremity (bit 15) in left shift or from the right extremity (bit 0) in right shift.

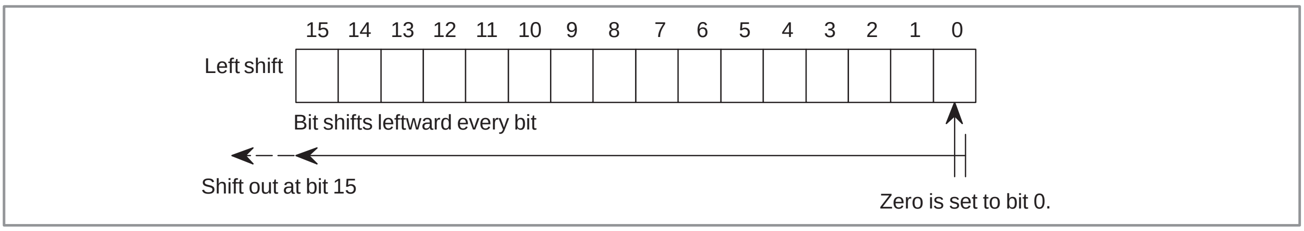
5.27.2 Format



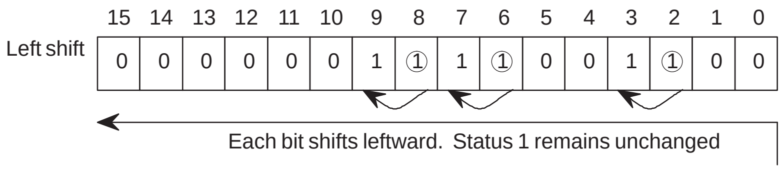
5.27.3 Control Conditions

(a) Shift direction specification (DIR)
DIR=0 : Left shift
DIR=1 : Right shift

(b) Condition specification (CONT)
CONT=0:
On "1" bit shifts by one bit in the specified direction.
The condition of an adjacent bit (eighter right or left adjacent bit according to the specification of shift direction DIR) is set to the original bit position of the on "1" bit.
Also, "0" is set to bit 0 after shifting in the left direction or set to hit 15 after shifting in the right direction.
In case of leftward shift;



CONT=1:
Shift is the same as above, but 1s are set to shifted bits.



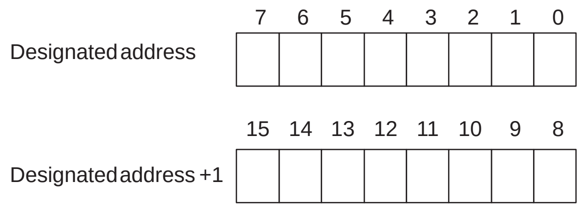
(c) Reset (RST)
The shifted out data (W1=1) is reset (W1=0).
RST=0 : W1 is not reset.
RST=1 : W1 is reset (W1=0).

(d) Actuation signal (ACT)
Shift processing is done when ACT=1. For shifting one bit only, execute an instruction when ACT=1, and then, set ACT to 0 (ACT=0).

5.27.4 Parameters

(a) Shift data addresses

Sets shift data addresses. These designated addresses require a continuous 2–byte memory for shift data.
Bit numbers are represented by bit 0 to 15 as shown below. When addresses are designated for programming, an address number is attached every 8 bits, and the designable bit numbers are 0 to 7.



5.27.5 W1

W1=0 : “1” was not shifted out because of the shift operation.
W1=1 : “1” was shifted out because of the shift operation.

5.28 DSCH (DATA SEARCH)

5.28.1 Function

DSCH is only valid for data tables (see section 6.3) which can be used by the PMC. DSCH searches the data table for a specified data, outputs an address storing it counting from the beginning of the data table. If the data cannot be found, an output is made accordingly.

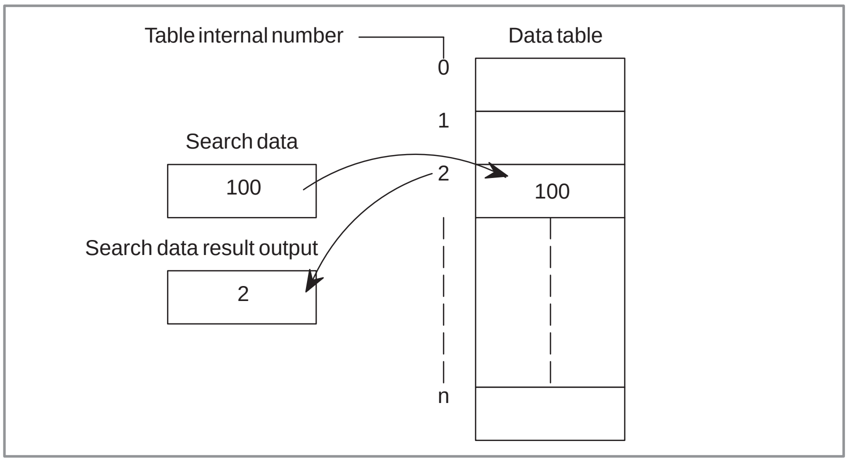


Fig. 5.28.1

> **NOTE**
> Parameter of this functional instruction and the data table heading address specified here are table internal number 0. The table internal number specified here, however, is different from that mentioned in 6.3.

5.28.2 Format

Fig.5.28.2 shows the expression format and Table 5.28.2 shows the coding format.

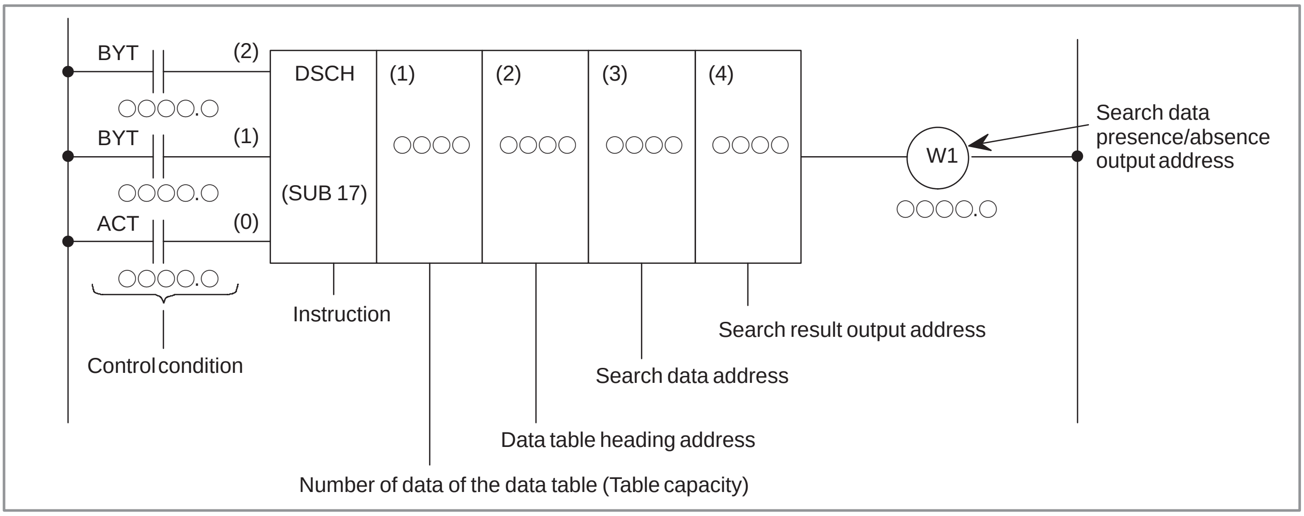


Fig. 5.28.2 DSCH instruction format

Table 5.28.2 DSCH instruction coding

Coding sheet

Step Number	Instruc-tion	Address No.	Bit No.	Remarks
1	RD	○○○ .	○	BYT
2	RD. STK	○○○ .	○	RST
3	RD. STK	○○○ .	○	ACT
4	SUB	17		DSCH instruction
5	(PRM)	○○○○		Number of data of the data table
6	(PRM)	○○○○		Data table heading address
7	(PRM)	○○○○		Search data address
8	(PRM)	○○○○		Search result output address
9	WRT	○○○ .	○	Search data presence/absence output adress

Memory status of control condition

ST3	ST2	ST1	ST0
			BYT
		BYT	RST
	BYT	RST	ACT
			W1

5.28.3 Control Conditions

(a) Specify data size.
BYT=0 : Data stored in the data table, BCD two digits long.
BYT=1 : Data stored in the data table, BCD four digits long.

(b) Reset
RST=0 : Release reset
RST=1 : Enables a reset, that is, sets W1 to 0.

(c) Execution command
ACT=0 : The DSCH instruction is not executed. W1 does not change.
ACT=1 : The DSCH is executed, and the table internal number storing the desired data is output., If the data cannot be found, W1=1.

5.28.4 Number Of Data of the Data Table

Specifies the size of the data table. If the beginning of the data table is 0 and the end is n, n+1 is set as the number of data of the data table.

5.28.5 Data Table Head Address

Addresses that can be used in a data table are fixed. When preparing a data table, the addresses to be used must be determined beforehand, specify the head address of a data table here.

5.28.6 Search Data Address

Indicates the address of the data to be searched.

5.28.7 Search Result Output Address

If the data being searched for is found, the internal number of the table storing the data is output to this field. This address field is called a search result output address field.
The search result output address field requires memory whose size is the number of bytes conforming to the size of the data specified by BYT.

5.28.8 Search Data Presence/Absence Output

W1=0 : The data to be searched exists.
W1=1 : The data to be searched does not exist.

5.29 DSCHB (BINARY DATA SEARCH)

5.29.1 Function

Alike the DSCH instruction of Section 5.28, this function instruction instructs data search in the data table.
There are two differences; the numerical data handled in this instruction are all in binary format; and number of data (table capacity) in the data table can be specified by specifying the address, thus allowing change in table capacity even after writing the sequence program in the ROM.

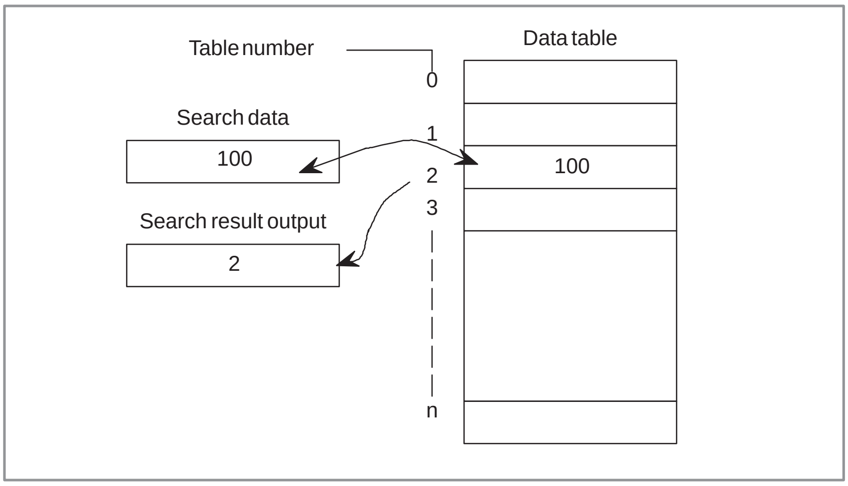


Fig. 5.29.1

5.29.2 Format

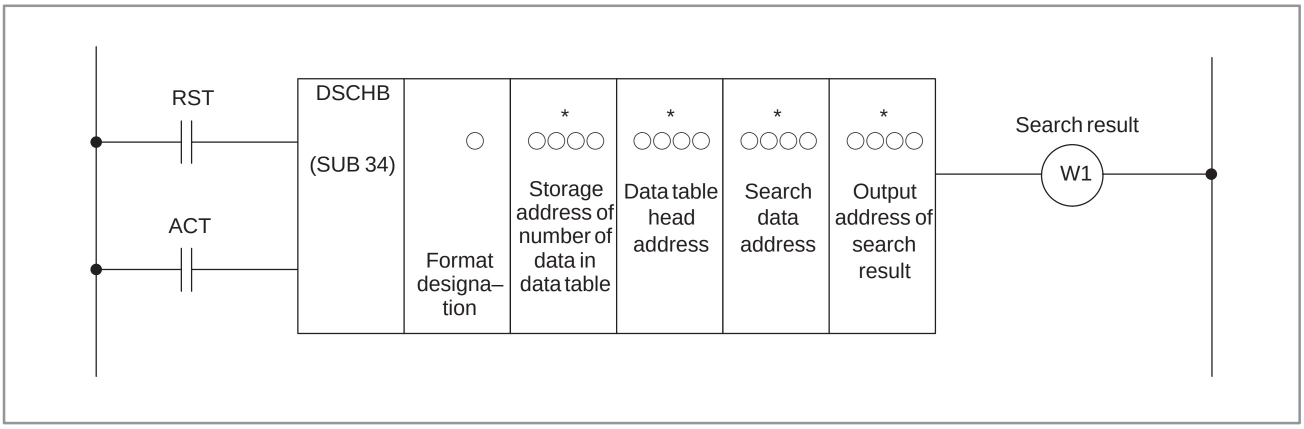


Fig. 5.29.2

5.29.3 Control Conditions

(a) R eset (RST)
RST=0 : Release reset
RST=1 : Reset. W1=”0”.

(b) Activation command
ACT=0 : Do not execute DSCHB instruction. W1 does not change.
ACT=1 : Execute DSCHB instruction. If the search data is found, table number where the data is stored will be output. If the search data is not found, W1 becomes 1.

5.29.4 Parameter

(a) Format designation
Specifies data length. Specify byte length in the first digit of the parameter.
1 : 1–byte long data
2 : 2–byte long data
4 : 4–byte long data

(b) Storage address of number of data in data table
Specifies address in which number of data in the data table is set. This address requires memory of number of byte according to the format designation.
Number of data in the table is n+1 (headnumber in the table is 0 and the last number is n).

(c) Data table head address
Sets head address of data table.

(d) Search data address
Address in which search data is set.

(e) Search result output address
After searching, if search data is found, the table number where the data is stored will be output. The searched table number is output in this search result output address. This address requires memory of number of byte according to the format designation.

5.29.5 Search Result (W1)

W1=0 : Search data found.
W1=1 : Search data not found.

5.30 XMOV (INDEXED DATA TRANSFER)

5.30.1 Function

Reads or rewrites the contents of the data table. Like the DSCH instruction, XMOV is only valid for data tables which can be used by the PMC.

NOTE
The data table heading address specified here is table internal number 0. The table internal number specified here, however, is different from that mentioned in 6.3.

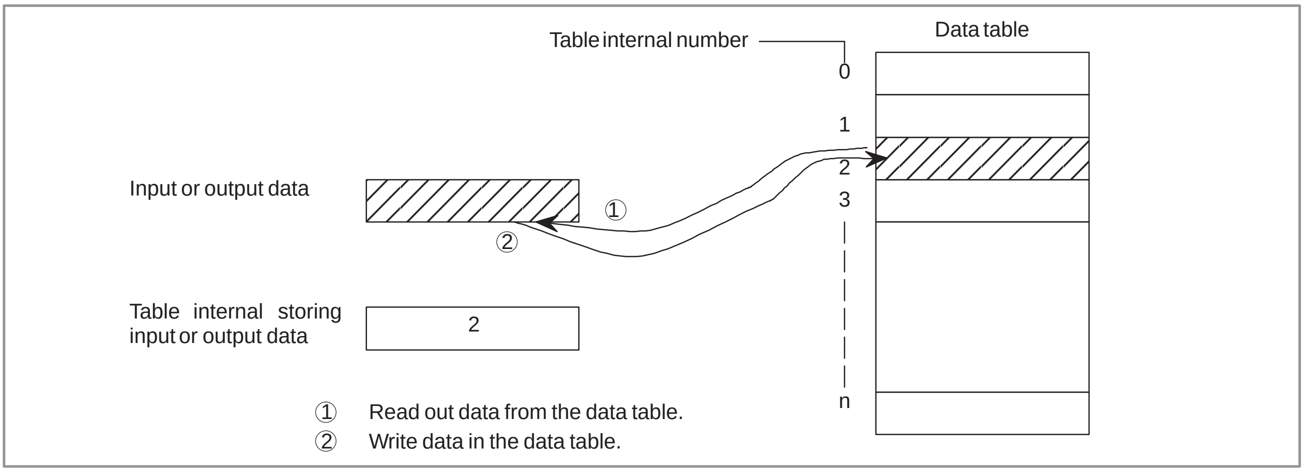


Fig. 5.30.1 Reading and writing of data

5.30.2 Format

Fig.5.30.2 shows the expression format and Table 5.30.2 shows the coding format.

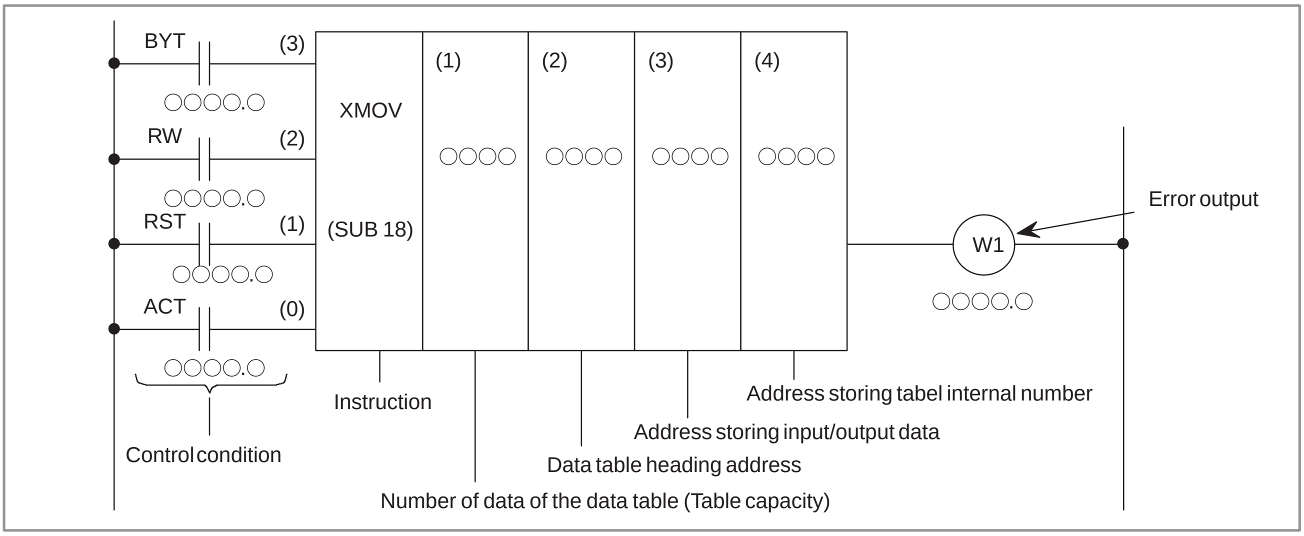


Fig. 5.30.2 XMOV instruction format

Table 5.30.2 XMOV instruction coding

Coding sheet

Step Number	Instruc-tion	Address No.	Bit No.	Remarks
1	RD	○○○ .	○	BYT
2	RD. STK	○○○ .	○	RW
3	RD. STK	○○○ .	○	RST
4	RD. STK	○○○ .	○	ACT
5	SUB	18		XMOV instruction
6	(PRM)	○○○○		Number of data of the data table
7	(PRM)	○○○○		Data table heading address
8	(PRM)	○○○○		Address storing input/output data
9	(PRM)	○○○○		Address storing table internal number
10	WRT	○○○ .	○	Error output

Memory status of control conditions

ST3	ST2	ST1	ST0
			BYT
		BYT	RW
	BYT	RW	RST
BYT	RW	RST	ACT
↓	↓	↓	↓
			W1

5.30.3 Control Conditions

(a) Specify the number of digits of data.
  BYT=0 : Data stored in the data table, BCD in two digits long.
  BYT=1 : Data stored in the data table, BCD in four digits long.

(b) Specify read or write
  RW=0 : Data is read from the data table.
  RW=1 : Data is write in the data table.

(c) Reset
  RST=0 : Release reset.
  RST=1 : Enables reset, that is, sets W1 to 0.

(d) Execution command
  ACT=0 : The XMOV instruction is not executed. W1 does not change.
  ACT=1 : The XMOV instruction is executed.

5.30.4 Number of Data Of the Data Table

Specifies the size of the data table. If the beginning of the data table is 0 and the end is n, n+1 is set as the number of data of the data table.

5.30.5 Data Table Head Address

Address that can be used in a data table are fixed. When preparing a data table, the addresses to be used must be determined beforehand, and the head address placed in that data table .

5.30.6 Address Storing Input/Output Data

The input/output data storage address is the address storing the specified data, and is external to the data table. The contents of the data table is read or rewritten.

5.30.7 Address Storing the Table Internal Number

The table internal number storage address is the address storing the table internal number of the data to be read or rewritten.
This address requires memory specified by the formaat designation (BYT).

5.30.8 Error Output

W1=0 : There is no error.
W1=1 : There is an error.
An error occurs if a table internal number exceeding the previously programmed number of the data table is specified.

5.31 XMOVB (BINARY INDEX MODIFIER DATA TRANSFER)

5.31.1 Function

Alike the XMOV instruction of Section 5.30, this function instruction instructs reading and rewriting of data in the data.
There are two differences; the numerical data handled in this instruction are all in binary format; and number of data (table capacity) in the data table can be specified by specifying the address, thus allowing change in table capacity even after writing the sequence program in the ROM.

In PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i*/Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*, the setting of the format specification parameter is extended. With this setting, XMOVB can read/write the multiple data in 1 instruction. For the details of the setting of a format specification parameter, refer to "5.31.4 Parameters".

(a) Read data from data table

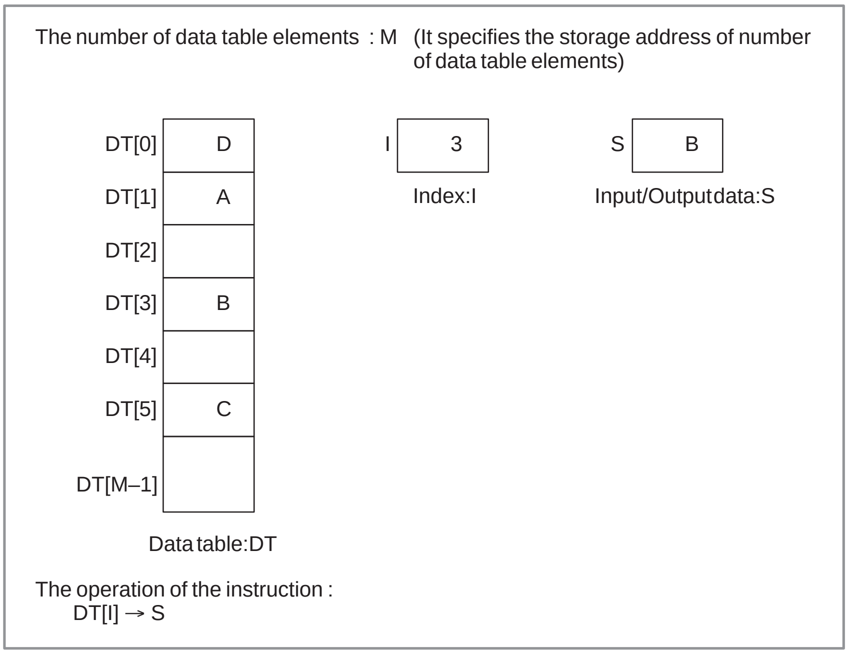


Fig. 5.31.1 (a) Read data from data table (basic specification)

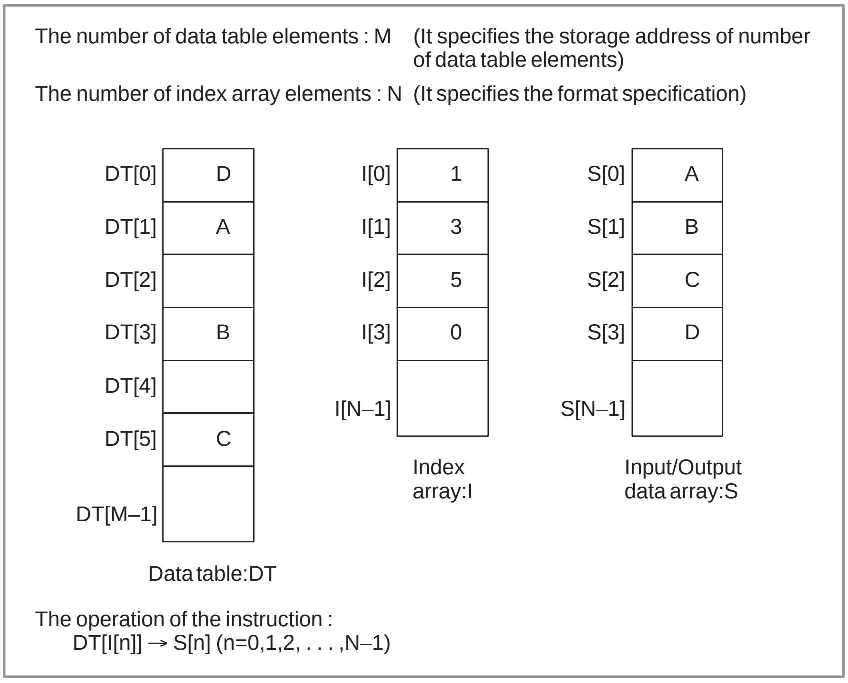


Fig. 5.31.1 (b) Read data from data table (expended specification) (only for PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*)

(b) Write data to data table

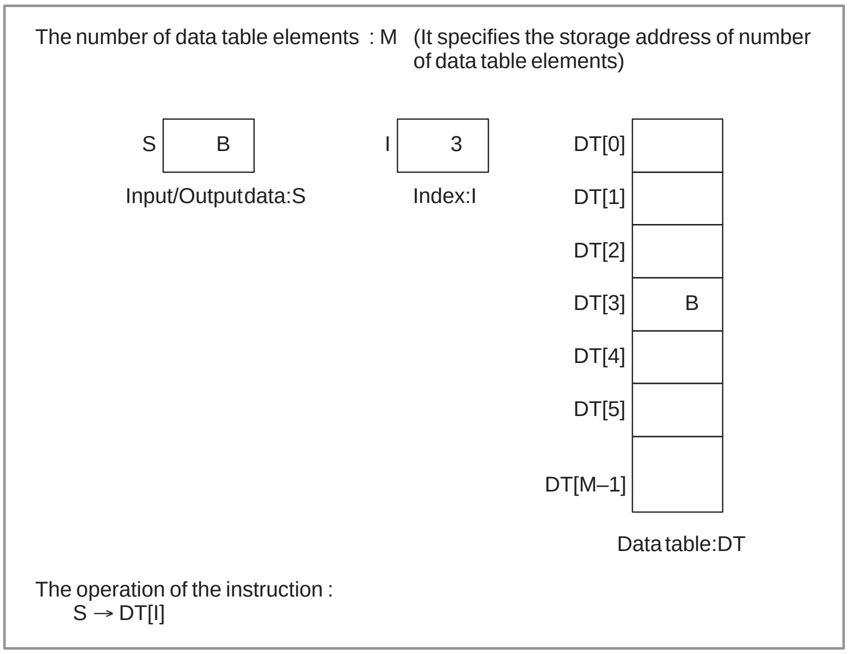


Fig. 5.31.1 (c) Write data to data table (basic specification)

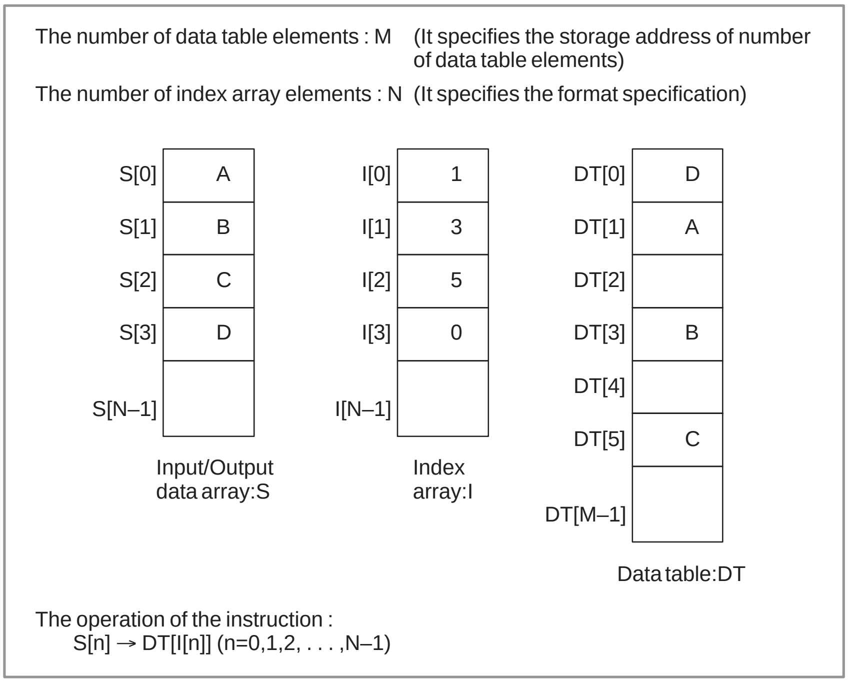


Fig. 5.31.1 (d) Write data to data table (expended specification) (only for PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*)

5.31.2 Format

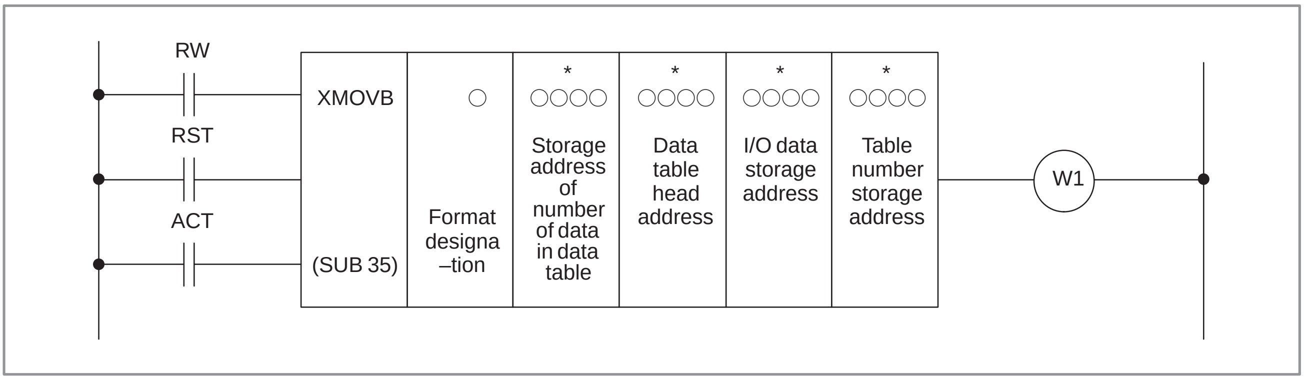


Fig. 5.31.2 (a) XMOVB instruction format

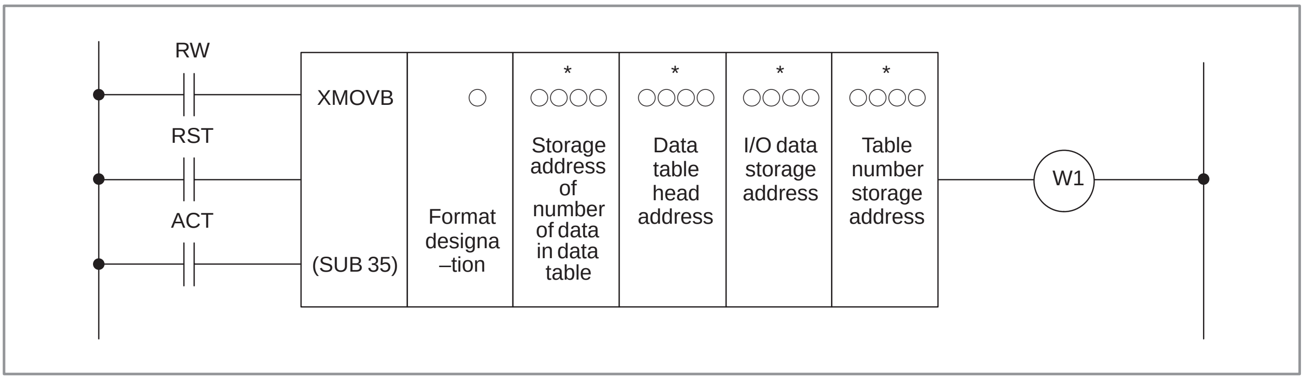


Fig. 5.31.2 (b) XMOVB (expended specification) (only for PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*)

5.31.3 Control Conditions

(a) Read, write designation (RW)
RW=0 : Read data from data table.
RW=1 : Write data to data table.

(b) Reset (RST)
RST=0 : Reset release.
RST=1 : Reset. W1=0.

(c) Activation command (ACT)
ACT=0: Do not execute MOV instruction.
There is no change in W1.
ACT=1: Execute MOV instruction.

5.31.4 Parameters

(a) Format designation
Specifies data length. Specify byte length in the first digit of the parameter.
0001 :1–byte long data
0002 :2–byte long data
0004 :4–byte long data

In PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i*/Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*, when setting format specification in the following extended format, XMOVB can read/write multiple data in data table in 1 instruction. Specifies data length (1, 2, or 4) to the 1st digit as above–mentioned. Specifies the number of the index array elements to the 2nd and 3rd digit. Specifies 0 to the 4th digit.

0nn1 : In case of reading/writing multiple (nn) data in data table by 1 byte length
0nn2 : In case of reading/writing multiple (nn) data in data table by 2 byte length
0nn4 : In case of reading/writing multiple (nn) data in data table by 4 byte length

The nn is the numerical value from 02 to 99. When setting 00 or 01, it works as the basic specification in which one data transfer is performed by one instruction.

Format specification (extended specification) :
0 n n x

The byte length setting
1 : 1 byte length
2 : 2 byte length
4 : 4 byte length

The number of the index array elements
00–01 :
It works as the basic specification.
02–99 :
Read/Write multiple (nn) data from/to data table.

(b) Storage address of number of data table elements
Set to the memory at the byte length which set the number of the data table elements in “(a) Format specification” and set the address to this parameter. The effective range of number of data table elements is as follows with the byte length which set in “(a) Format specification”.

1 byte length : 1 to 255
2 byte length : 1 to 32767
(Actually, set a value below the size of the D area.)
4 byte length : 1 to 99999999
(Actually, set a value below the size of the D area.)

(c) Data table head address
Sets head address in the data table.
The memory of (byte length) × (number of data table elements) which was set in “(a) Format specification” and “(b) Storage address of number of data table elements” is necessary.

(d) Input/Output data storage address
In case of the reading, set the address of the memory which stores a reading result. In case of the writing, set the address of the memory which stores a writing result. The memory with the byte length which set in “(a) Format specification” is necessary.

In PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*, when setting format specification in the extended format, set the head address of the array. (In case of the reading, set the head address of the array in which a reading result is stored. In case of the writing, set the head address of the array in which a writing result is stored.) The memory of (byte length) × (number of index array elements) which was set in “(a) Format specification” is necessary.

(e) Index storage address
Set the address of the memory in which an index value is stored. The memory with the byte length set in “(a) Format specification” is necessary. The effective range of number of data in index is as follows according to the byte length set in “(A) Format specification”.
Actually, set the value which is smaller than the value to set in “(b) Storage address of number of data table elements” to the index.
When setting an index value above the value to set in “(b) Storage address of number of data table elements”, it causes an error output W1=1 in instruction execution.

1 byte length : 0 to 254
2 byte length : 0 to 32766
4 byte length : 0 to 99999998

In PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*, when setting format specification in the extended format, set an address at the head of the array in which an index value is stored. The memory of (byte length) × (number of data in index array) which was set in “(a) Format specification” is necessary.

5.31.5 Error Output (W1)

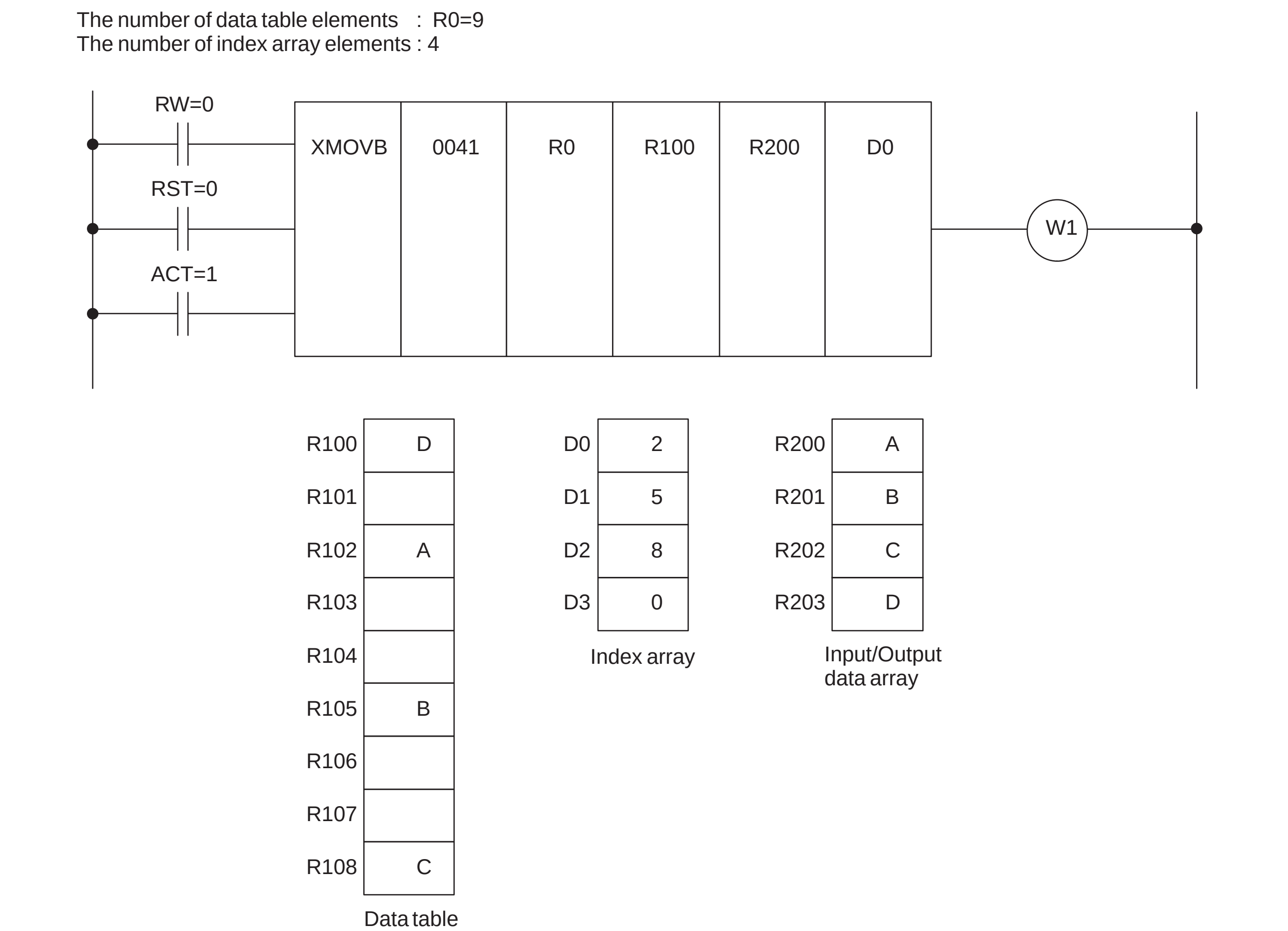
W1=0 : No error

W1=1 : Error found. In the case where the index value set in “(e) Index storage address” exceeds the value set in “(b) Storage address of number of data table elements”, it becomes W1=1. The reading or writing of the data table isn’t executed.

In PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*, when setting format specification in the extended format, in the case the value set in “(b) Storage address of number of data table elements”, it becomes W1=1. The reading or writing of a data table is executed for the normal index values but not executed as for the wrong index values.

5.31.6 Example for Extended Specification

(a) Read data from data table (extended specification)

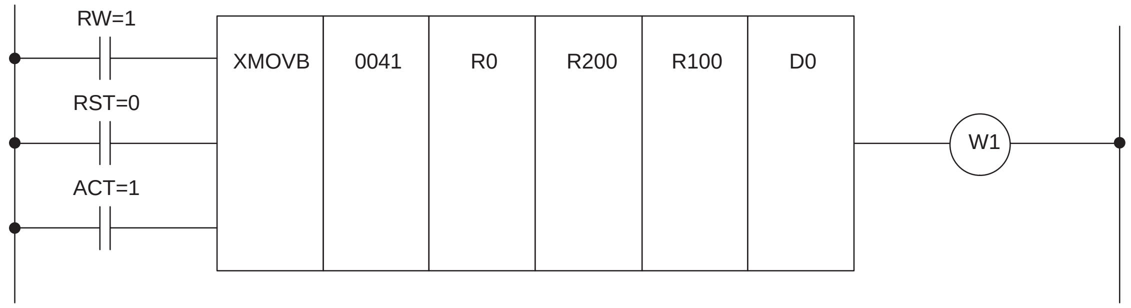


The operation of the instruction :
(1) R102 → R200
(2) R105 → R201
(3) R108 → R202
(4) R100 → R203

Fig.5.31.6 (a) Example for XMOVB (extended specification)
(only for PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*)

(b) Write data to data table (extended specification)

The number of data table elements : R0=9
The number of index array elements : 4



Input/Output data array

R100	A
R101	B
R102	C
R103	D

Index array

D0	2
D1	5
D2	8
D3	0

Data table

R200	D
R201	
R202	A
R203	
R204	
R205	B
R206	
R207	
R208	C

The operation of the instruction :
(1) R100 → R202
(2) R101 → R205
(3) R102 → R208
(4) R103 → R200

Fig. 5.31.6 (b) Example for XMOVB (extended specification)
(only for PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*)

5.32 ADD (ADDITION)

5.32.1 Function

Adds BCD two–or four–digit data.

5.32.2 Format

Fig.5.32.2 shows the expression format and Table 5.32.2 shows the coding format.

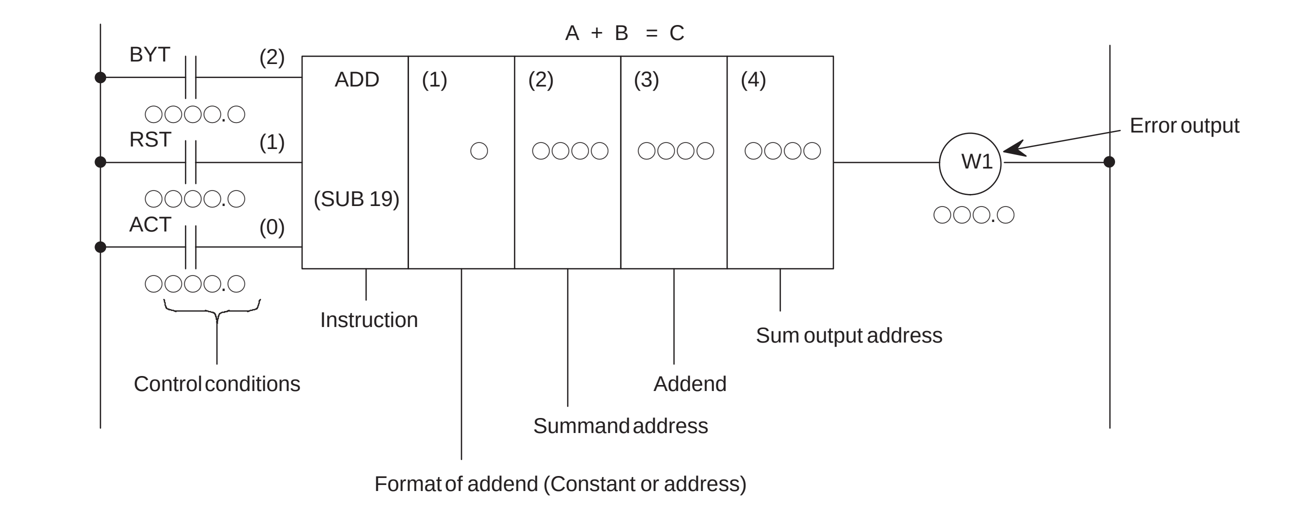


Fig. 5.32.2 ADD instruction format

Table 5.32.2 DSCH instruction coding

Coding sheet

Step Number	Instruc-tion	Address No.	Bit No.	Remarks
1	RD	○○○ .	○	BYT
2	RD. STK	○○○ .	○	RST
3	RD. STK	○○○ .	○	ACT
4	SUB	19		ADD instruction
5	(PRM)	○		Addend format
6	(PRM)	○○○○		Summand address
7	(PRM)	○○○○		Addend (address)
8	(PRM)	○○○○		Sum output address
9	WRT	○○○ .	○	Error output

Memory status of control conditions

ST3	ST2	ST1	ST0
			BYT
		BYT	RST
	BYT	RST	ACT
			W1

5.32.3 Control Conditions

(a) Specify the number of digits of data.
BYT=0 : Data is BCD two digits long.
BYT=1 : Data is BCD four digits long.

(b) Reset
RST=0 : Release reset.
RST=1 : Resets error output W1, that is, sets W1 to 0.

(c) Execution command
ACT=0 : The ADD instruction is not executed.
ACT=1 : The ADD instruction is executed.

5.32.4 Data Format of Addend

0 : Specifies addend with a constant.
1 : Specifies addend with an address.

5.32.5 Summand Address

Set the address storing the summand.

5.32.6 Addend (Address)

Addressing of the addend depends on 4).

5.32.7 Sum Output Address

Set the address to which the sum is to be output.

5.32.8 Error Output

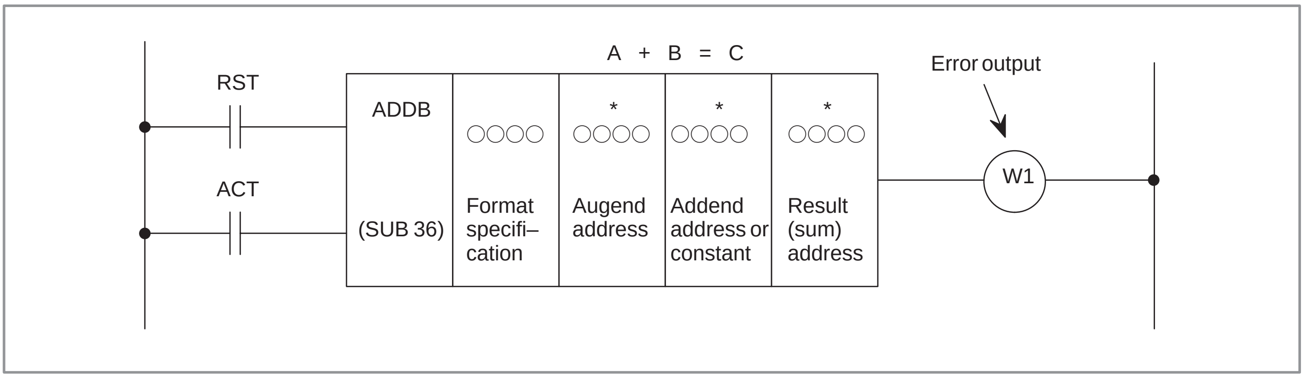
If the sum exceeds the data size specified in 3), a), W1=1 is set to indicate an error.

5.33 ADDB (BINARY ADDITION)

5.33.1 Function

This instruction performs binary addition between 1–, 2–, and 4–byte data. In the operation result register (R9000), operating data is set besides the numerical data representing the operation results. The required number of bytes is necessary to store each augend, the added, and the operation output data.

5.33.2 Format



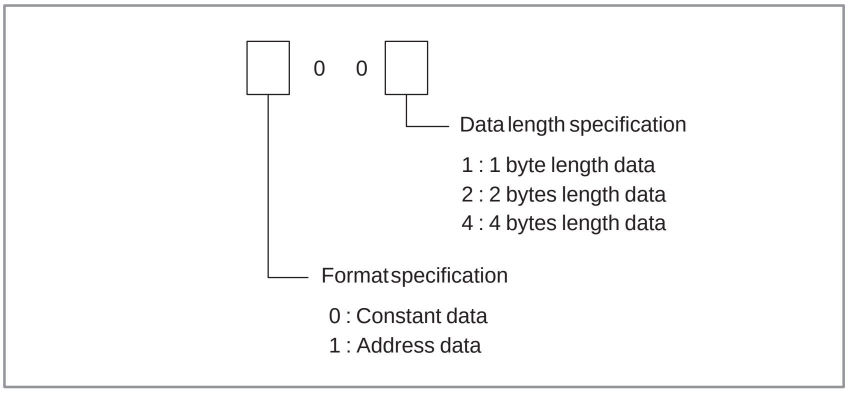
5.33.3 Control Conditions

(a) Reset (RST)
RST=0 : Release reset
RST=1 : Resets error output W1. In other words, makes W1=0.

(b) Command (ACT)
ACT=0 : Do not execute ADDB. W1 does not change now.
ACT=1 : Execute ADDB.

5.33.4 Parameters

(a) Format specification
Specifies data length (1,2, and 4 bytes) and the format for the addend (constant or address).



(b) Augend address
Address containing the augend.

(c) Addend data (address)
Specification in (a) determines the format of the addend.

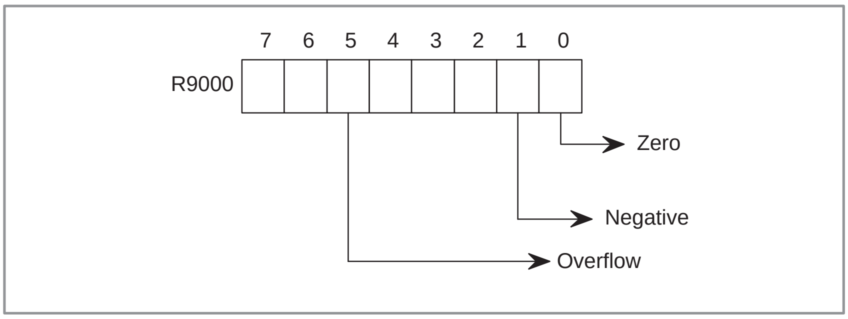
(d) Result output address
Specifies the address to contain the result of operation.

5.33.5 Error Output (W1)

W1=0 : Operation correct
W1=1 : Operation incorrect
W1 goes on (W1=1) if the result of addition exceeds the specified data length.

5.33.6 Operation Output Register (R9000)

This register is set with data on operation. If register bit is on, they signify the following operation data:



5.34 SUB (SUBTRACTION)

5.34.1 Function

Subtracts BCD two–or four–digit data.

5.34.2 Format

Fig.5.34.2 shows the expression format and Table 5.34.2 shows the coding format.

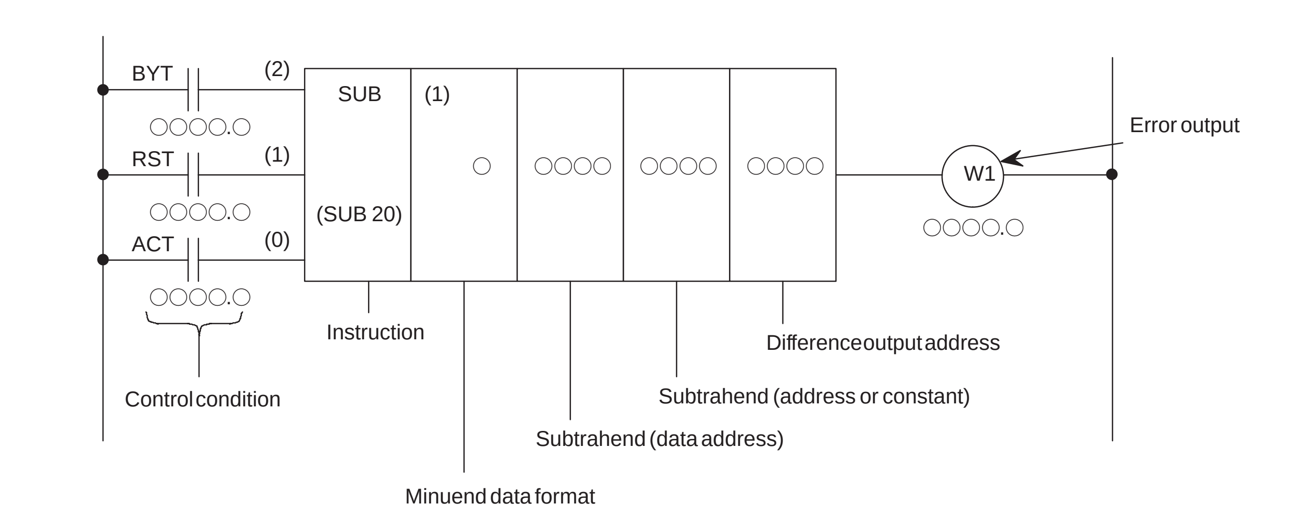


Fig. 5.34.2 SUB Instruction format

Table 5.34.2 SUB instruction format

Coding sheet

Step Number	Instruc-tion	Address No.	Bit No.	Remarks
1	RD	○○○ .	○	BYT
2	RD. STK	○○○ .	○	RST
3	RD. STK	○○○ .	○	ACT
4	SUB	20		SUB instruction
5	(PRM)	○		Data format of subtrahend
6	(PRM)	○○○○		Minuend address
7	(PRM)	○○○○		Subtrahend (address)
8	(PRM)	○○○○		Difference output address
9	WRT	○○○ .	○	Error output

Memory status of control conditions

ST3	ST2	ST1	ST0
			BYT
		BYT	RST
	BYT	RST	ACT
			W1

5.34.3 Control Conditions

(a) Specification of the number of digits of data.
BYT=0 : Data BCD two digits long
BYT=1 : Data BCD four digits long

5.34.4 Reset

RST=0 : Release reset.
RST=1 : Resets error output W1, that is, sets W1 to 0.

5.34.5 Execution Command

ACT=0 : The SUB instruction is not executed. W1 does not change.
ACT=1 : The SUB instruction is executed.

5.34.6 Data Format of Subtrahend

0 : Specifies subtrahend with a constant.
1 : Specifies subtrahend with an address.

5.34.7 Minuend Address

Set the address storing the minuend.

5.34.8 Subtrahend (Address)

Addressing of the subtrahend depends on (6).

5.34.9 Difference Output Address

Sets the address to which the difference is output.

5.34.10 Error Output

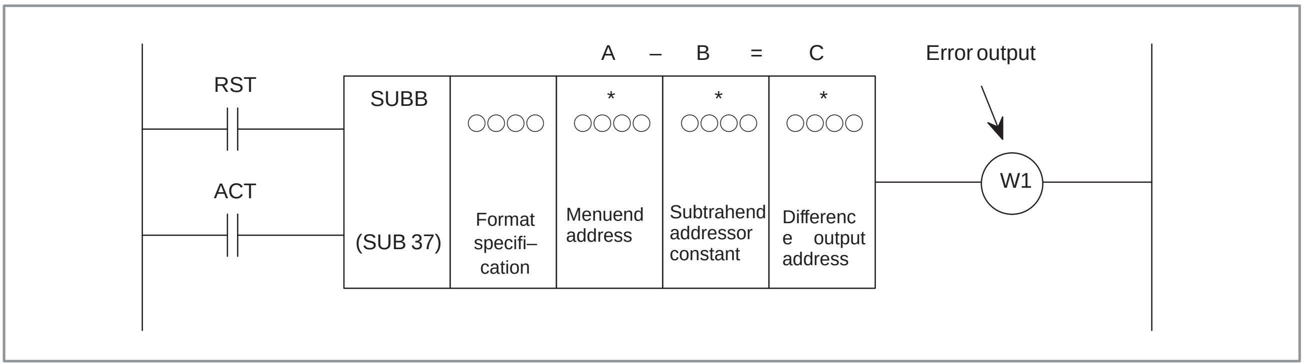
W1 is set 1 to indicate an error if the difference is negative.

5.35 SUBB (BINARY SUBTRACTION)

5.35.1 Function

This instruction subtracts one data from another, both data being in the binary format of 1, 2 or 4 bytes.
In the operation result register (R9000), operation data is set besides the numerical data representing the operation. A required number of bytes is necessary to store the subtrahend, minuend, and the result (difference).

5.35.2 Format



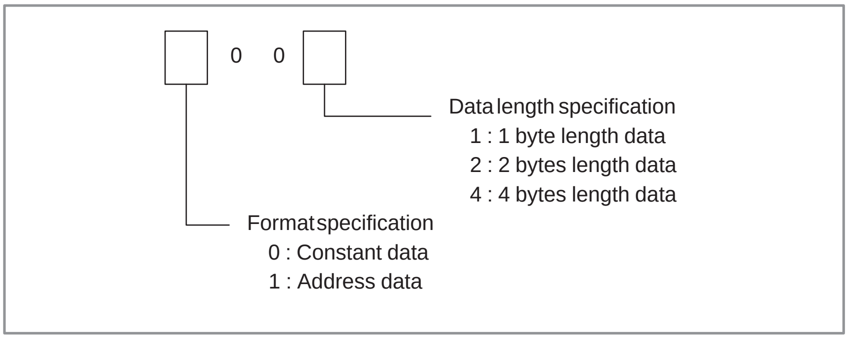
5.35.3 Control Conditions

(a) Reset (RST)
RST=0 : Release reset
RST=1 : Resets error output W1. (Set W1 to 0.)

(b) Command (ACT)
ACT=0 : Do not execute SUBB. W1 does not change now.
ACT=1 : Execute SUBB.

5.35.4 Parameters

(a) Format specification
Specifies data length (1, 2, and 4 bytes) and the format for the subtrahend (constant or address).



(b) Minuend address
Address containing the minuend.

(c) Minuend data (address)
Specification in (a) determines the format of the minuend.

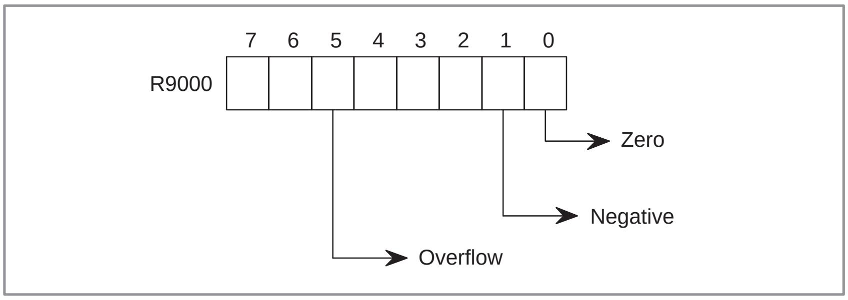
(d) Result output address
Specifies the address to contain the result of operation.

5.35.5 Error Output (W1)

W1=0 : Operation correct
W1=1 : Operation incorrect
W1 goes on (W1=1) if the result of subtraction exceeds the specified data length.

5.35.6 Operation Output Register (R9000)

This register is set with data on operation. If register bit is on, they signify the following operation data:



5.36 MUL (MULTIPLICATION)

5.36.1 Function

Multiplies BCD two–or four–digit data. The product must also be BCD two–or four–digit data.

5.36.2 Format

Fig.5.36.2 shows the expression format and Table 5.36.2 shows the coding format.

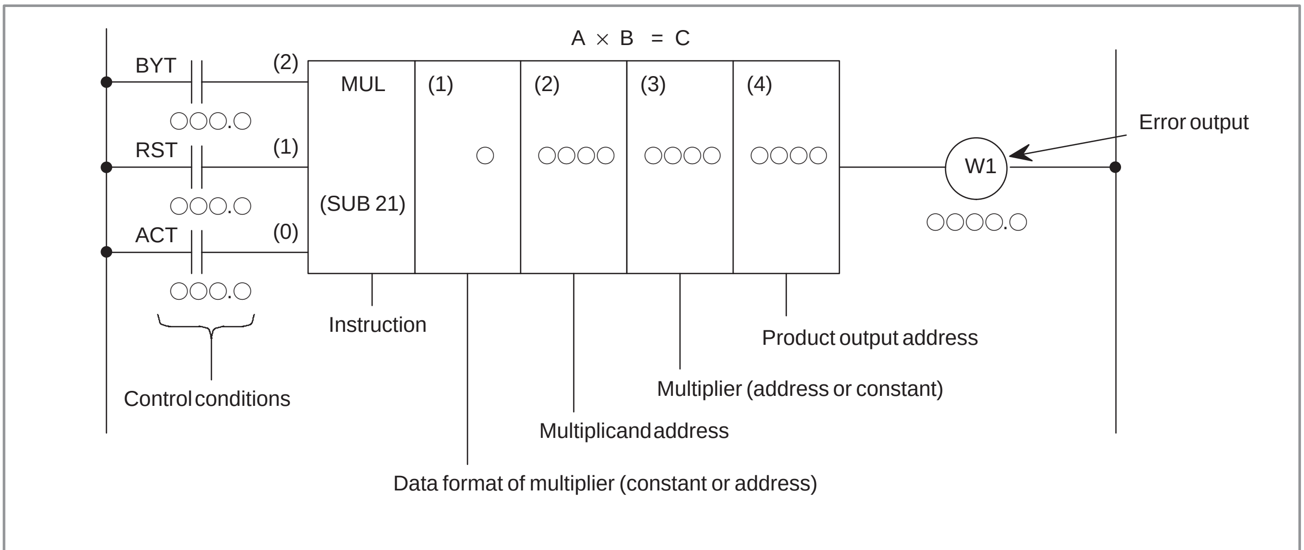


Fig. 5.36.2 MUL instruction format

Table 5.36.2 MUL instruction coding

Coding sheet

Step Number	Instruc-tion	Address No.	Bit No.	Remarks
1	RD	○○○ .	○	BYT
2	RD. STK	○○○ .	○	RST
3	RD. STK	○○○ .	○	ACT
4	SUB	21		MUL instruction
5	(PRM)	○		Data format of multiplier
6	(PRM)	○○○○		Multiplicand address
7	(PRM)	○○○○		Multiplier (address)
8	(PRM)	○○○○		Product output address
9	WRT	○○○ .	○	Error output

Memory status of control conditions

ST3	ST2	ST1	ST0
			BYT
		BYT	RST
	BYT	RST	ACT
			W1

5.36.3 Control Conditions

(a) Specify the number of digits of data.
BYT=0 : Data is BCD two digits long.
BYT=1 : Data is BCD four digits long.

(b) Reset
RST=0 : Releases reset.
RST=1 : Resets error output W1, that is, sets W1 to 0.

(c) Execution command
ACT=0 : The MUL instruction is not executed. W1 does not change.
ACT=1 : The MUL instruction is executed.

5.36.4 Data Format of Multiplier

0 : Specifies multiplier with a constant.
1 : Specifies multiplier with an address.

5.36.5 Multiplicand Address

Sets the address storing the multiplicand.

5.36.6 Multiplier (Address)

Addressing of the multiplier depends on 4).

5.36.7 Product Output Address

Set the address to which the product is output.

5.36.8 Error Output

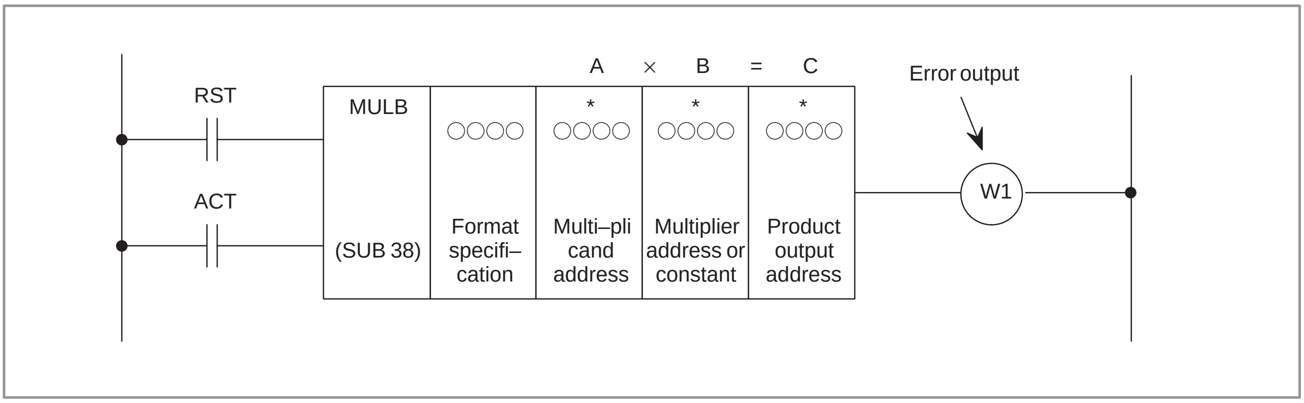
W1=1 is set to indicate an error if the product exceeds the size specified in 3), a).

5.37 MULB (BINARY MULTIPLICATION)

5.37.1 Function

This instruction multiplies 1–, 2–, and 4–byte binary data items. In the operation result register (R9000), operation data is set besides the numerical data representing the operation.
A required number of bytes is necessary to store multiplicand, multiplier, and the result (product).

5.37.2 Format



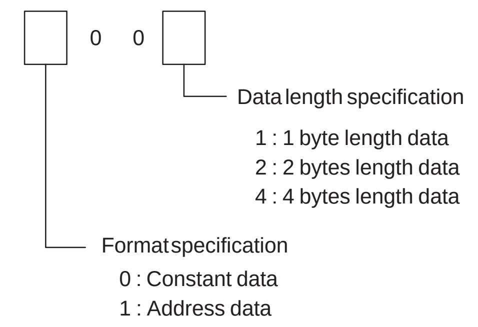
5.37.3 Control Conditions

(a) Reset (RST)
RST=0 : Release reset
RST=1 : Resets error output W1. In other words, makes W1=0.

(b) Command (ACT)
ACT=0: Do not execute MULB. W1 does not change now.
ACT=1: Execute MULB.

5.37.4 Parameters

(a) Format specification
Specifies data length (1, 2, and 4 bytes) and the format for the multiplier (constant or address).



(b) Multiplicand address
Address containing the multiplicand.

(c) Multiplier data (address or constant)
Specification in (a) determines the format of the multiplier.

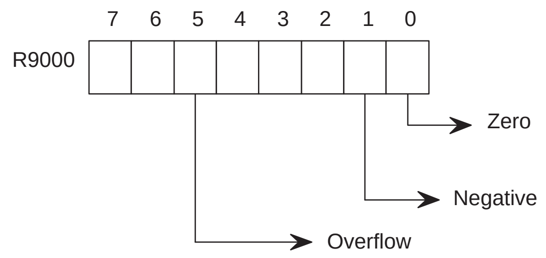
(d) Result output address
Specifies the address to contain the result of operation.

5.37.5 Error Output (W1)

W1=0 : Operation correct
W1=1 : Operation incorrect
W1 goes on (W1=1) if the result of multiplication exceeds the specified data length.

5.37.6 Operation Output Register (R9000)

This register is set with data on operation. If register bit is on, they signify the following operation data:



5.38 DIV (DIVISION)

5.38.1 Function

Divides BCD two–or four–digit data. Remainders are discarded.

5.38.2 Format

Fig.5.38.2 shows the expression format and Table 5.38.2 shows the coding format.

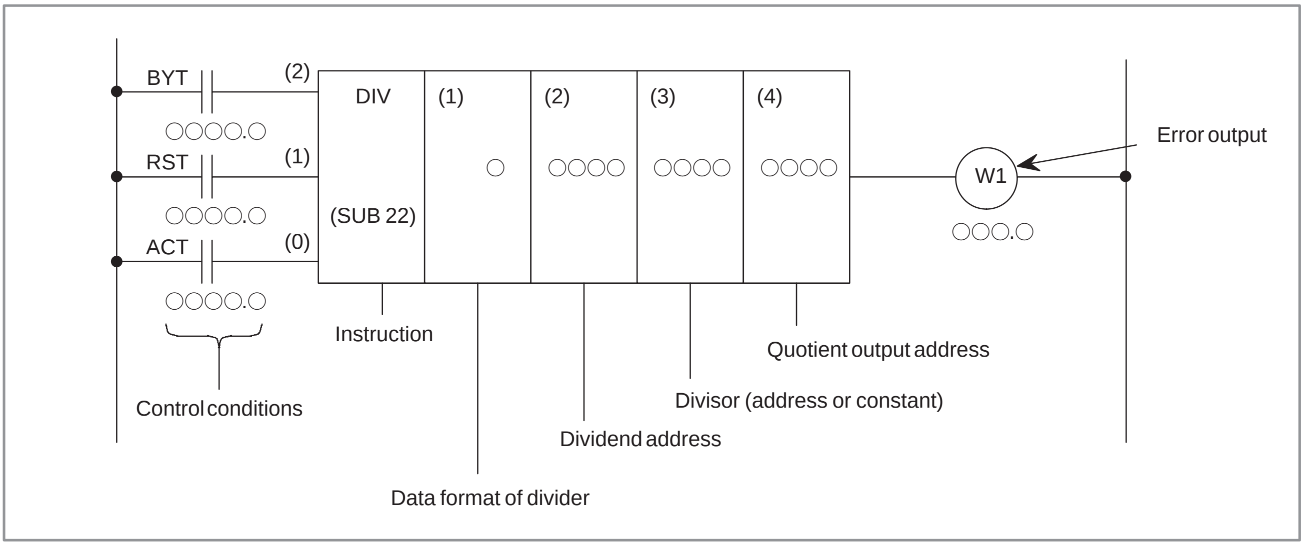


Fig. 5.38.2 DIV instruction format

Table 5.38.2 DIV instruction coding

Coding sheet

Step Number	Instruc-tion	Address No.	Bit No.	Remarks
1	RD	○○○ .	○	BYT
2	RD. STK	○○○ .	○	RST
3	RD. STK	○○○ .	○	ACT
4	SUB	22		DIV instruction
5	(PRM)	○		Data format of divider
6	(PRM)	○○○○		Dividend address
7	(PRM)	○○○○		Divider (address)
8	(PRM)	○○○○		Quatient output address
9	WRT	○○○ .	○	Error output

Memory status of control conditions

ST3	ST2	ST1	ST0
			BYT
		BYT	RST
	BYT	RST	ACT
			W1

5.38.3 Control Conditions

(a) Specify the number of digits of data.
BYT=0 : Data is BCD two digits long.
BYT=1 : Data is BCD four digits long.

(b) Reset
RST=0 : Releases reset.
RST=1 : Resets error output W1, that is, sets W1 to 0.

(c) Execution command
ACT=0 :The DIV instruction is not executed. W1 does not change.
ACT=1 :The DIV instruction is executed.

5.38.4 Divisor Data Format Designation

0 : Specifies divisor data by constant.
1 : Specifies divisor data by address.

5.38.5 Dividend Address

Sets the address storing the dividend.

5.38.6 Divisor (Address)

Addressing of the divisor depends on 4).

5.38.7 Quotient Output Address

Sets the address to which the quotient is output.

5.38.8 Error Output

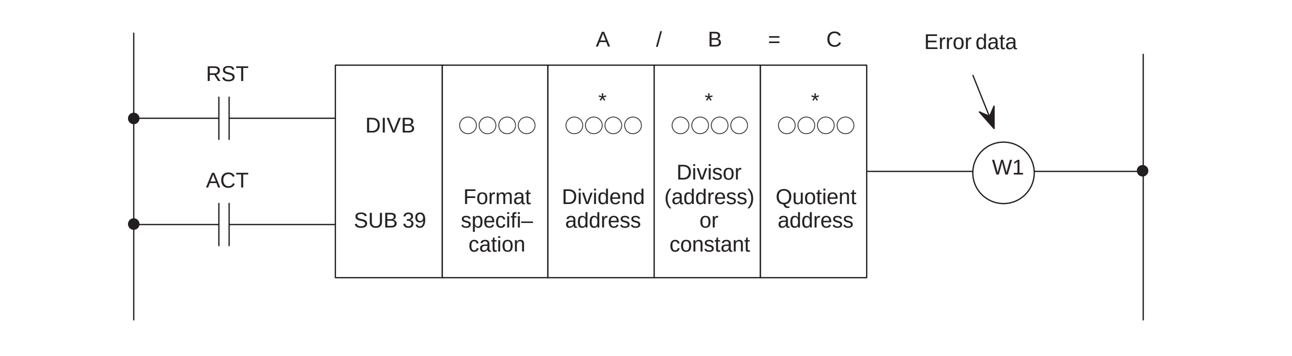
W1=1 is set to indicate an error if the divider is 0.

5.39 DIVB (BINARY DIVISION)

5.39.1 Function

This instruction divides binary data items 1, 2, and 4 byte in length. In the operation result register (R9000), operation data is set and remainder is set to R9002 and following addresses.
A required number of bytes is necessary to store the dividend, divisor, and the result (quotient).

5.39.2 Format



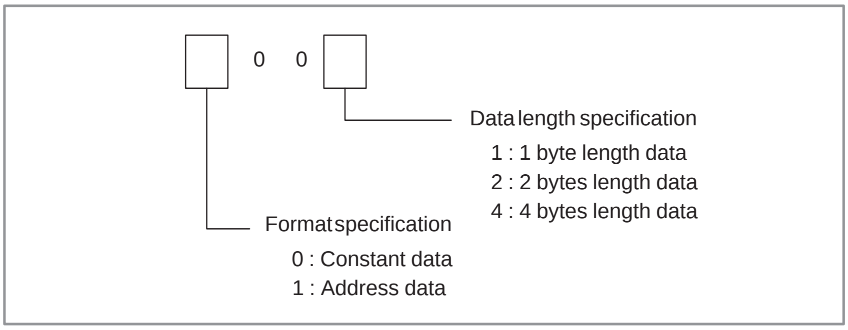
5.39.3 Control Conditions

(a) Reset (RST)
RST=0 : Release reset
RST=1 : Resets error output W1. In other words, makes W1=0.

(b) Command (ACT)
ACT=0 : Do not execute DIVB. W1 does not change now.
ACT=1 : Execute DIVB.

5.39.4 Parameters

(a) Format specification
Specifies data length (1, 2, and 4 bytes) and the format for the divisor (constant or address).



(b) Dividend address
Address containing the dividend

(c) Divisor data (address)
Specification in (a) determines the format of the divisor.

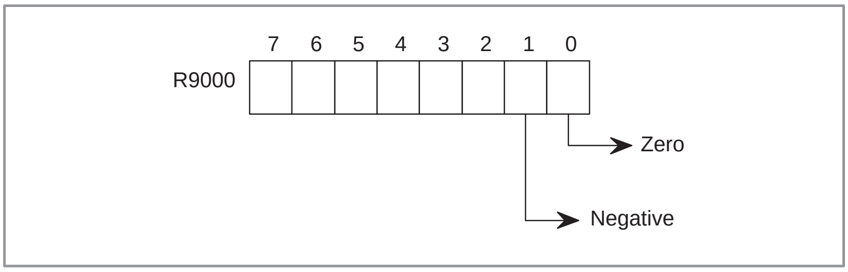
(d) Result output address
Specified the address to contain the result of operation.

5.39.5 Error Putput (W1)

W1=0 : Operation correct
W1=1 : Operation incorrect
W1 goes on (W1=1) if the divisor is 0.

5.39.6 Operation Output Register (R9000)

This register is set with data on operation. If register bit is on, they signify the following operation data:



5.39.7 Remainder Output Address

Depending on its length, the remainder is stored in one or more of registers R9002 to R9005.

5.40 NUME (DEFINITION OF CONSTANT)

5.40.1 Function

Defines constants, when required. In this case, constants are defined with this instructions.

5.40.2 Format

Fig.5.40.2 shows the expression format and Table 5.45.40.2 shows the coding format.

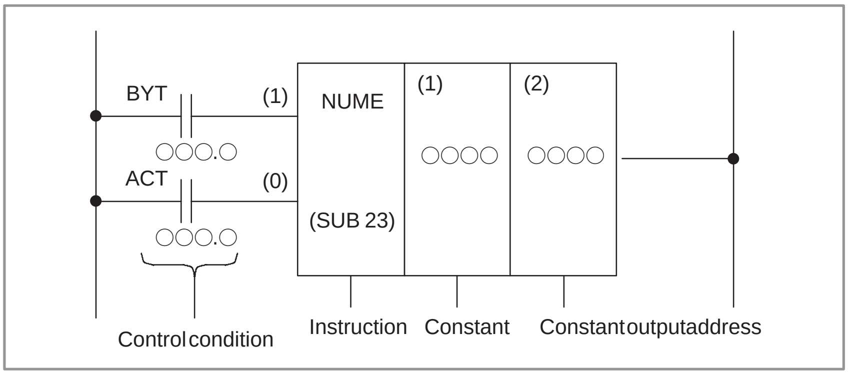


Fig. 5.40.2 NUME instruction format

Table 5.40.2 NUME instruction coding

Coding sheet

Step Number	Instruc-tion	Address No.	Bit No.	Remarks
1	RD	○○○ .	○	BYT
2	RD. STK	○○○ .	○	ACT
3	SUB	23		NUME instruction
4	(PRM)	○○○○		Constant
5	(PRM)	○○○○		Constant output address

Memory status of control conditions

ST3	ST2	ST1	ST0
			BYT
		BYT	ACT

5.40.3 Control Conditions

(a) Specify the number of digits of a constant.
BYT=0 : Constant is BCD two digits long.
BYT=1 : Constant is BCD four digits long.

(b) Execution command
ACT=0 :The NUME instruction is not executed.
ACT=1 :The NUME instruction is executed.

5.40.4 Constant

Sets the constant as the number of digits specified in Item (a) in Subsec. 5.40.3.

5.40.5 Constant Output Address

Sets the address to which the constant defined in Subsec. 5.40.4 is output.

5.41 NUMEB (DEFINITION OF BINARY CONSTANTS)

5.41.1 Function

This instruction defines 1, 2, or 4–bytes long binary constant. Data entered in decimal during programming is converted into binary data during program execution. The binary data is stored in the specified memory address(es).

In PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*, the setting of the format specification parameter is extended. With this setting, NUMEB can store multiple data by 1 instruction. This extended specification is effective when initializing a large memory area with value. For the details of the setting of a format specification parameter, refer to “5.41.4 Parameters”.

5.41.2 Format

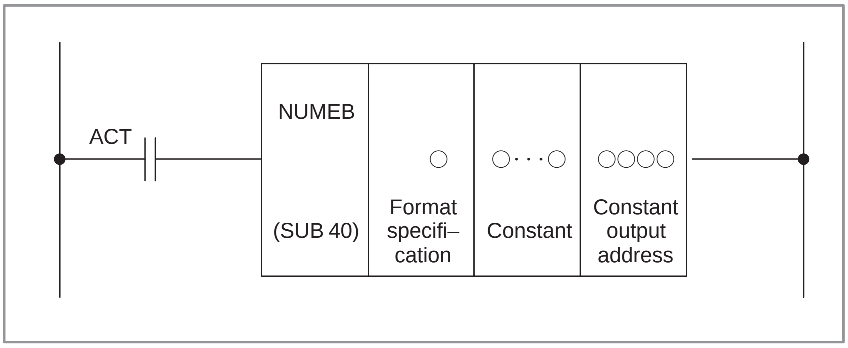


Fig. 5.41.2 (a) NUMBER instruction format

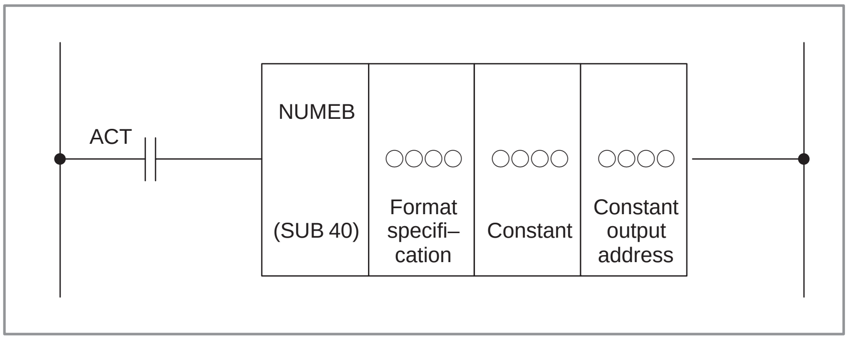


Fig. 5.41.2 (b) Expression format of NUMEB (extended specification) (only for PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*)

5.41.3 Control Conditions

(a) Command (ACT)
ACT= 0 : Do not execute NUMEB.
ACT= 1 : Execute NUMEB.

5.41.4 Parameters

(a) Format specification
Specifies data length (1, 2, or 4 bytes).
Use the first parameter digit to specify byte length:
0001 : Binary data of 1 byte length
0002 : Binary data of 2 byte length
0004 : Binary data of 4 byte length

In PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*, when setting format specification in the following extended format, NUMEB can store multiple data by 1 instruction.
Specify data length (1, 2, or 4) to the 1st digit as above–mentioned.
Specify the number of the array in which is a constant to the 2nd and 3rd digit is defines.
Specify 0 to the 4th digit.
0nn1 : In case of defining multiple (nn) data by 1 byte length
0nn2 : In case of defining multiple (nn) data by 2 byte length
0nn4 : In case of defining multiple (nn) data by 4 byte length

The nn is the numerical value from 02 to 99. When setting 00 or 01, it works as the basic specification that works for one data.

Format specification (extended specification) :
0 n n x

The byte length setting of constant
1 : 1 byte length
2 : 2 byte length
4 : 4 byte length

Number of data in the array
00–01 :
It defines constant at 1 memory.
02–99 :
It defines constants at multiple (nn) memory.

(b) Constant
Defined constants in decimal format. Set a constant data within the effective range for the byte length which is set in "(a) Format specification".

(c) Constant output address
Specifies the address of the area for output of the binary data. The memory of the number of bytes which is set in "(a) Format specification" is necessary.

In PMC–SB5/SB6 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5 for Series 21*i*/210*i*, when setting format specification in the extended format, it is necessary to reserve memory of (byte length) × (number of array elements which define constant) which was set in "(a) Format specification".

5.42 DISP (MESSAGE DISPLAY) (PMC–SB/SB2/SB3/SB4/SB5/SB6/SC/SC3/SC4 ONLY)

5.42.1 Function

DISP is used to display messages on the CRT screen, CNC of which enters alarm status. Message data to be displayed is specified after the parameters of the functional instruction. One DISP functional instruction can define up to 16 types of message. Display is performed by setting the control condition ACT to 1. In order to display and then clear a message, set the display–request bit corresponding to the message data number to 1 and 0, respectively.

Up to one alarm message (message data putting the CNC in alarm status) can be displayed on one screen. When one message is cleared, a message is displayed. Similarly, each time one of the message is displayed. One operator message (message data not putting the CNC in alarm status) can be displayed on a screen. When an operator message is cleared in a state when four operator messages are displayed, the subsequent operator message is displayed.

5.42.2 Format

Fig.5.42.2 shows the instruction format and Table 5.42.2 shows the coding format.

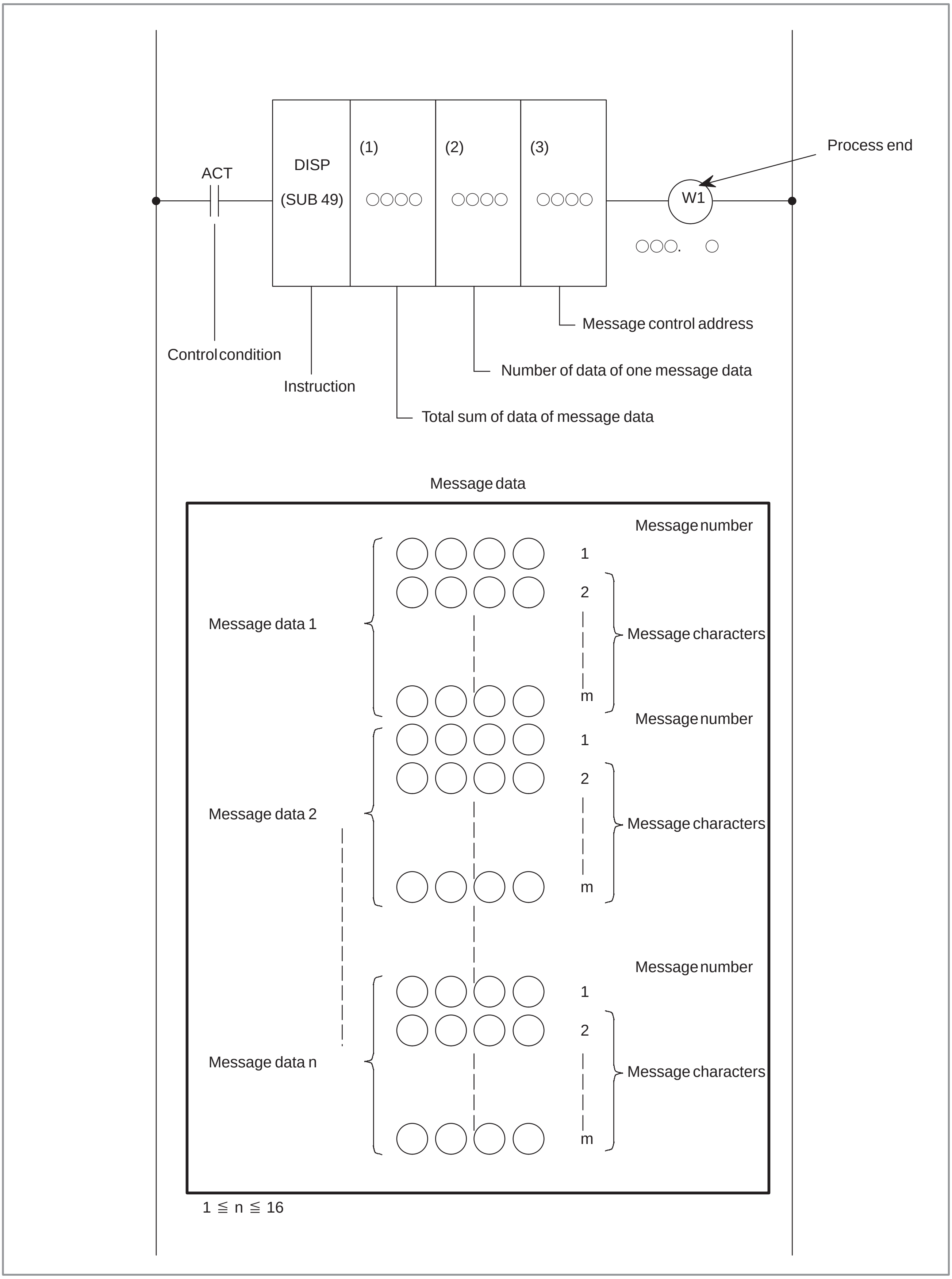


Fig. 5.42.2 DISP instruction format

Table 5.42.2 DISP instruction coding

Coding sheet / **Memory status of control conditions**

Step Number	Instruction	Address No.	Bit No.	Remarks	ST2	ST1	ST0
	RD	○○○.	○	ACT			ACT
	SUB	49		DISP			ACT
	(PRM)	○○○○		Total sum of data of message data			
	(PRM)	○○○○		Number of data of one message item			
	(PRM)	○○○○		Message control address			
	(PRM)	○○○○		Message number			
	(PRM)	○○○○		2			
	(PRM)	○○○○		3			
	:	:		: Message characters			
	:	:		:			
	(PRM)	○○○○		m			
	(PRM)	○○○○		Message number			
	(PRM)	○○○○		2			
	(PRM)	○○○○		3			
	:	:		: Message characters			
	:	:		:			
	(PRM)	○○○○		m			
	:	:		:			
	:	:		:			
	(PRM)	○○○○		Message number			
	(PRM)	○○○○		2			
	(PRM)	○○○○		3			
	:	:		: Message characters			
	:	:		:			
	(PRM)	○○○○		m			
	WRT	○○○○		Process end (W1)			W1

5.42.3 Control Condition

ACT=0: Nothing is processed. W1 does not change.
ACT=1: The specified message data is displayed or cleared.
ACT must remain 1 until processing end is reported by W1.

5.42.4 Parameters

(a) Total sum of message data of data:
$m \times n$

(b) Number of data of one massage data:
m Note)

(c) Message control address: Specifies the address of the RAM of internal relay area (see (7) for details).

> **NOTE**
> The number of data used by each message data item, m, must be the same. Since 00 is ignored, it can be set for unnecessary data. For example, for particular messages with a different number of displayed characters, set 00 so that the number of data, m, are the same.

5.42.5 Message Data

(a) Message number:

The specified number produces an appropriate event as follows.

1000 to 1999 (alarm message):

The CNC is put in alarm status and the number and following data are displayed. The maximum number of the displayed characters is up to 32, except for the message number. When an alarm status occurs, the operation being executed stops. To release the alarm status, set the display–request bit (see Fig.5.42.7) to 0.

2000 to 2099 (operator message):

The CNC is not put in alarm status and the number and following data are displayed. The maximum number of the displayed characters is 255, except for the message number.

2100 to 2999 (operator message):

The CNC is not put in alarm status and the number is not displayed. Only the following data (up to 255 characters) is displayed.

5000–5999 (alarm messages on path 2):

Path 2 is placed in the alarm state. A displayed message number is a specified number from which 4000 is subtracted. The number of displayed characters excluding this number is 32 or less. If the alarm state arises during axis movement, a gradual stop occurs. The alarm state can be released by setting the display request bit to 0.

7000–7999 (alarm messages on path 3):

Path 3 is placed in the alarm state. The displayed message number is a specified number from which 6000 is subtracted. The number of displayed characters excluding this number is 32 or less. If the alarm state arises during axis movement, a gradual stop occurs. The alarm state can be released by setting the display request bit to 0.

NOTE

If all characters in the operator message are kana characters, up to 254 kana characters are displayed.

(b) Message character

An alphanumeric character is specified with a two–digit decimal (two characters per step). Table 5.42.6 shows the correspondence between characters and specified numbers.

The above message data is always specified because it is written on ROM. The message data cannot therefore be changed as desired. However, arbitrary numeric data of up to four BCD digits can be displayed according to the specified variable data. The spindle tool number which changes whenever ACT tools are changed and the number of the tool at the tool–change position can be displayed, for example. For specifying variable data, see (10) below.

5.42.6 Error Output (W1)

W1=0 : Processing ends. Normally, W1=0. If W1=0 after W1=1, processing ends.
W1=1 : In process. W1=1 when ACT=1.

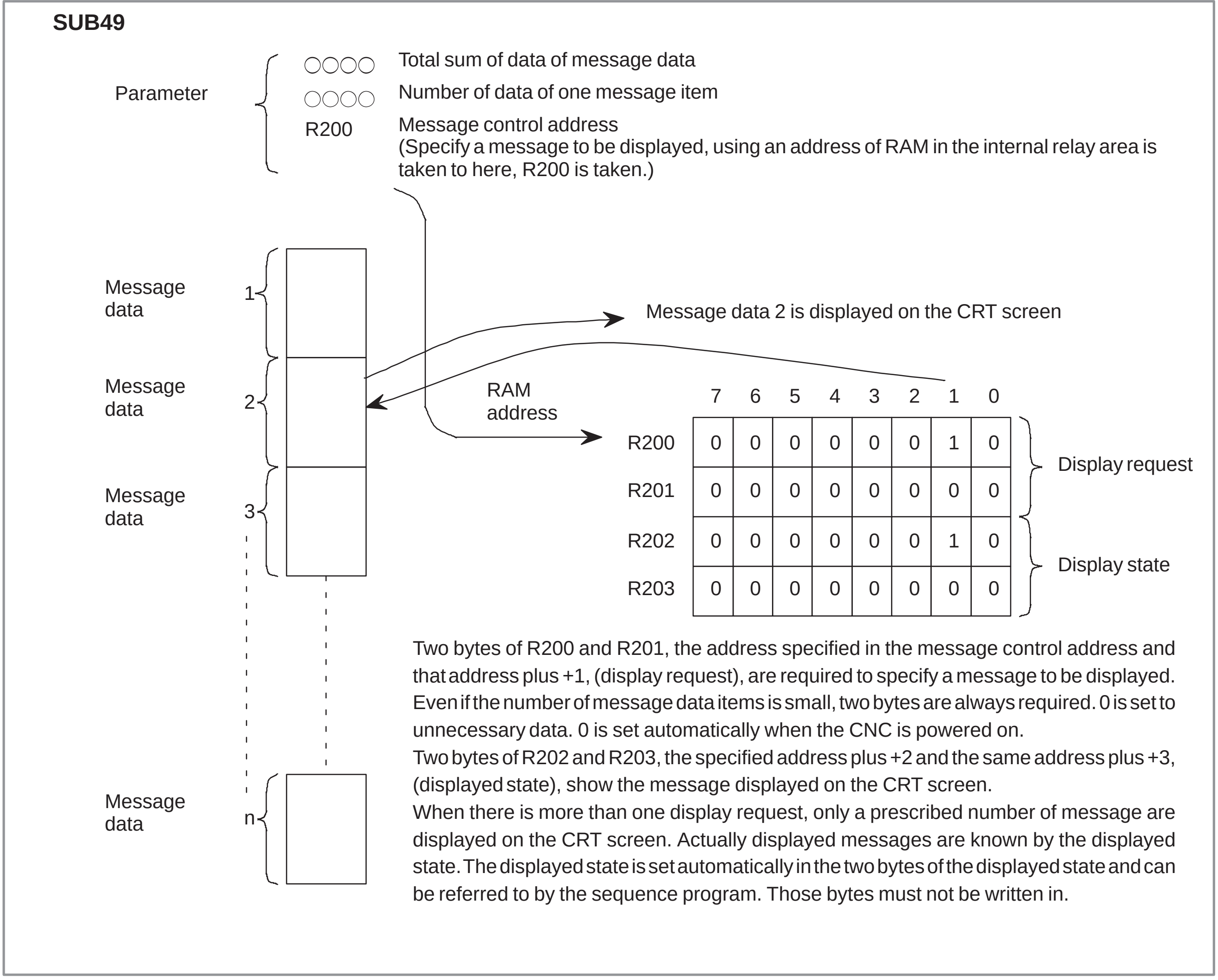
Table 5.42.6 Correspondince between characters and specified numbers

Specified number	Corresponding character	Specified number	Corresponding character	Specified number	Corresponding character	Specified number	Corresponding character
32	␣ (space)	64	@	160	to	192	タ
33	!	65	A	161	°	193	チ
34	”	66	B	162	「	194	ツ
35	#	67	C	163	」	195	テ
36	$	68	D	164	、	196	ト
37	%	69	E	165	·	197	ナ
38	&	70	F	166	ヲ	198	ニ
39	’	71	G	167	ァ	199	ヌ
40	(	72	H	168	ィ	200	ネ
41	)	73	I	169	ゥ	201	ノ
42	*	74	J	170	ェ	202	ハ
43	+	75	K	171	ォ	203	ヒ
44	,	76	L	172	ャ	204	フ
45	- *1)	77	M	173	ュ	205	ヘ
46	.	78	N	174	ョ	206	ホ
47	/	79	O	175	ッ	207	マ
48	0	80	P	176	- *3)	208	ミ
49	1	81	Q	177	ア	209	ム
50	2	82	R	178	イ	210	メ
51	3	83	S	179	ウ	211	モ
52	4	84	T	180	エ	212	ヤ
53	5	85	U	181	オ	213	ユ
54	6	86	V	182	カ	214	ヨ
55	7	87	W	183	キ	215	ラ
56	8	88	X	184	ク	216	リ
57	9	89	Y	185	ケ	217	ル
58	:	90	Z	186	コ	218	レ
59	;	91	[	187	サ	219	ロ
60	<	92	¥	188	シ	220	ワ
61	=	93	]	189	ス	221	ン
62	>	94	∧	190	セ	222	”
63	?	95	- *2)	191	ソ	223	°

*1) minus *2) Under bar *3) Long bar

5.42.7 Parameters and Message Data

The parameters and message data used by this functional instruction are as follows.



Two bytes of R200 and R201, the address specified in the message control address and that address plus +1, (display request), are required to specify a message to be displayed. Even if the number of message data items is small, two bytes are always required. 0 is set to unnecessary data. 0 is set automatically when the CNC is powered on.
Two bytes of R202 and R203, the specified address plus +2 and the same address plus +3, (displayed state), show the message displayed on the CRT screen.
When there is more than one display request, only a prescribed number of message are displayed on the CRT screen. Actually displayed messages are known by the displayed state. The displayed state is set automatically in the two bytes of the displayed state and can be referred to by the sequence program. Those bytes must not be written in.

One DISP functional instruction requires the four consecutive bytes following the address specified in the above message control address in order to check the display request and displayed status.

When messages are displayed or cleared, message data 1 to n (n ≦ 16) and display–request bits correspond to each other as shown in Fig.5.42.7.

To display and clear a message data item, set the corresponding bit to 1 and 0, respectively, and the control condition ACT to 1.

If the sequence program checks messages displayed on the screen, message data 1 to n and display–request bits correspond to each other as shown in Fig.5.42.7.

Message data for which 1 is set among the 16 displayed status bits, is the message data currently being displayed.

Note)		7	6	5	4	3	2	1	0
Display request	Specified address	Message data 8	Message data 7	Message data 6	Message data 5	Message data 4	Message data 3	Message data 2	Message data 1
	Specified address	Message data 16	Message data 15	Message data 14	Message data 13	Message data 12	Message data 11	Message data 10	Message data 9
Display state	Specified address +2	Message data 8	Message data 7	Message data 6	Message data 5	Message data 4	Message data 3	Message data 2	Message data 1
	Specified address +3	Message data 16	Message data 15	Message data 14	Message data 13	Message data 12	Message data 11	Message data 10	Message data 9

Fig. 5.42.7 Correspondence between message data and display request/displayed status

NOTE
"Specified address" means an address specified in the message control address of a DISP instruction parameter.

5.42.8 Remarks on Using the DISP Instruction

(a) CNC external data input function

The DISP instruction displays mes–sages using external data input function or external message display, which in–volves external work–number search, external tool offset, external work co–ordinate system shift, etc. as well as message display. The DISP instruction cannot display messages when any of these functions is being executed. To check this, EPCA (any address in inter–nal relay area) and EPCB (any address in control relay area) are used as interlock signal. The sequence program sets EPCA to 1 while the message is displayed, and to 0 upon competion of processing. The sequence program sets EPCB to 1 while any function other than the above is being processed, and to 0 upon completion of processing.
When EPCB = 1, messages must not be displayed (DISP ACT must not be 1). Set ACT to 1 after making sure that EPCB = 0.
When the function other than message display is executed, execute after making sure that EPCA = 0. DISP instruction and external data input function (external tool offset, external work number search) must be programmed in the same sequence level.

(b) External data input function address
During DISP instruction execution (EPCA = 1), the PMC → CNC interface of the external data input function must not be used for processing of external tool offset, external work–number search or external work coordinate system shifting. If EPCA = 1, use the JUMP instruction, for example, to skip writing data, so that nothing is written in the interface.

(c) ACT and W1 of the DISP instruction

(a) Timing of ACT ON

If EPCB = 0, ACT may be set to 1 with any timing. For instance, when all display–request bits are off or when the status displayed on the screen and the display requests are the same, that is, when there are no new display requests, even if ACT = 1, the DISP instruction processes nothing and the operation terminates (W1 = 0).

Even if another display–request bit is set on and ACT is set to 1 with a prescribed number of messages (four alarm messages or one operator messages) displayed on the screen, no message is displayed for that request, but W1 = 0 after W1 = 1 and W1 = 1 again during execution of the next cycle. In other words, W1 only changes back and forth between 1 and 0.

(b) Using two or more DISP instructions

If EPCB = 0, ACT of each DISP instruction may be turned on simultaneously. Until the DISP instruction whose ACT was set to 1 earlier, has been completed (W1 = 0), executing of the next DISP instruction is kept waiting. W1 of the DISP instruction kept waiting remains 0 at this time. Consequently, no messages more than those specified number are displayed, as discussed in (a).

From (i) and (ii) above, set ACT to 1 whenever EPCB = 0. Do not set ACT to 1 when EPCB = 1.

5.42.9 Examples of Using The DISP Instruction

(a) Display three types of messages with the following conditions.

SPER = 1 and “SPINDLE ALARM” (Message data 1)
ATCER = 1 and “ACT ALARM” (Message data 2)
WORK = 1 and “WORK SET UP” (Message data 3)

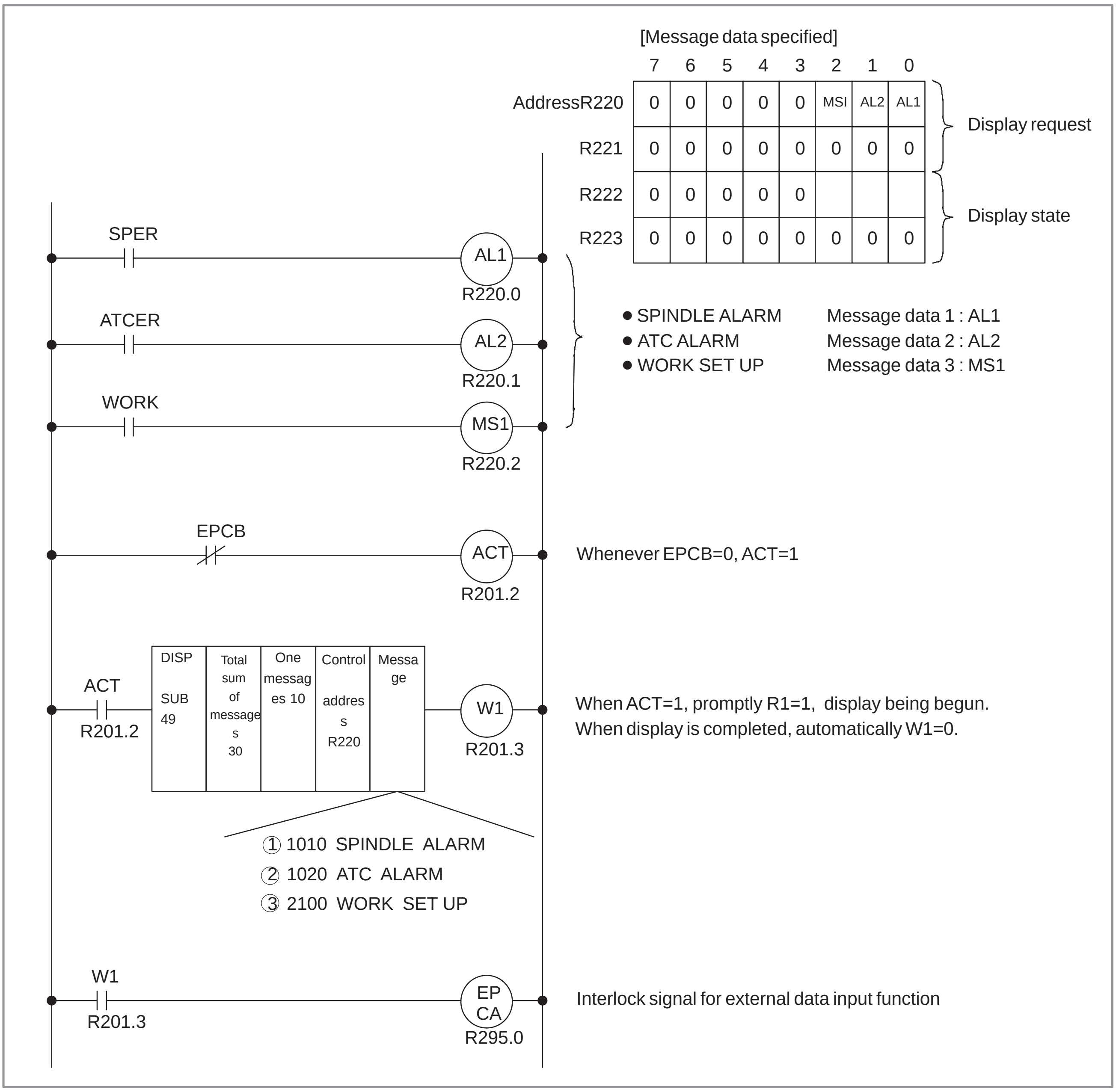


Fig. 5.42.9 (a)

Step Number	Instruction	Address No.	Bit No.	Remarks		ST2	ST1	ST0
	RD	R201.2						ACT
	SUB	49						
	(PRM)	30		Total sum of data of message data				
	(PRM)	10		Number of data of one message				
	(PRM)	R220		Message control address				
	(PRM)							
	(PRM)	1010		Message No.	Message data 1 (10 data m=10)			
		8380		SP				
		7378		IN				
		6876		DL				
		6932		E_				
		6576		AL				
		6582		AR				
		7700		M				
		(Note1)						
		0000						
		0000						
		1020		Message No.	Message data 2 (10 data m=10)			
		6584		AT				
		6732		C_				
		6576		AL				
		6582		AR				
		7700		M				
		0000						
		0000						
		0000						
		0000						
		2100		Message No.	Message data 3 (10 data m=10)			
		8779		W0				
		8275		RK				
		3200		_				
		0192		ﾀ				
		0222		”				
		0221		ﾝ				
		0196		ﾄ				
		0222		”				
	(PRM)	0216		ﾘ				ACT
	WRT	R201.3		Process end (W1)				W1
	RD	R201.3						W1
	WRT	R295.0						W1

NOTE

1 00 is ignored data.

2 Display example (The following is displayed on the screen in message data 1).
1010 SPINDLE ALARM

(b) Using three DISP instructions and one external tool offset

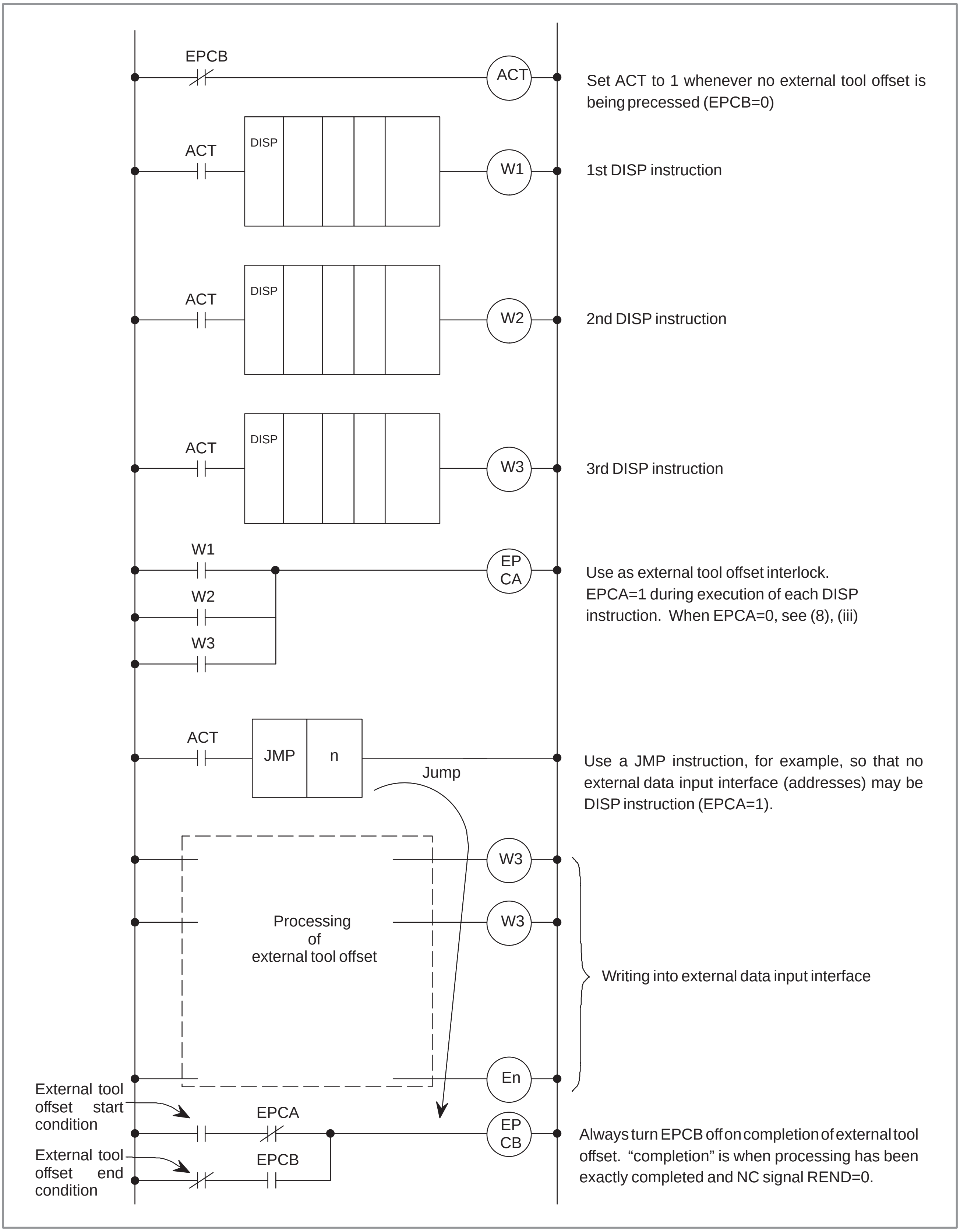


Fig. 5.42.9 (b)

5.42.10 Variable Data Display by Specifying Variable Data

Conform to the following instruction format. Variable data, i.e., any numeric value of up to four BCD digits, can be displayed.

SUB49	Instruction format
○○○○	Total number of steps in message data
○○○○	Number of steps in one message data item
○○○○	Message control address
○○○○	Message number
Message characters	
○○○○	
990m	Variable data specification
□□□□	Variable data address
○○○○	Message number
Message characters	
○○○○	
○○○○	Message number
990m	Variable data specification
□□□□	Variable data address
Message characters	
990m	Variable data specification
□□□□	Variable data address

NOTE
1. One step is used at variable data specification 990m.
2. The number of steps is the same for each message data item. The number of characters to be displayed varies according to the value specified for m.
3. Multiple variable data items can be used in one message data item.

(1) Specifying variable data

9 9 0 m

- m: Specifies the number of digits in the variable data. (1 ≦ m ≦ 4)
- 9 9: Variable data

(2) Variable data address
□□□□: Address of the area in which variable data is stored

(3) Variable data
Specify variable data consisting of up to four BCD digits (the number of digits specified for m) to be displayed at the address specified by the variable data address using the sequence program.
For example, variable data 1234 is specified at variable data address R300 in BCD as shown below:

AddressR300	0011	0100
R301	0001	0010

(4) Example
To display TOOL NO 123
SUB49
0007 Total number of steps in message data
0007 Number of steps in one message data item
R300 Message control address
2100 Message number
8479 TO
7396 OL
3278 N
7932 O
9903 Variable data specification
R350 Variable data address

AddressR350	0011	0100
R351	0001	0010

5.43 DISPB

5.43.1 Function

This instruction displays messages on the CRT/MDI screen. You can also specify the message number to generate an alarm in the CNC. This instruction supports special functions (numerical data display and kanji character display) in addition to the same basic functions as those of the message display instruction (DISP), described in Section 5.42.
However, it performs a special additional function, namely, it displays numerical data.
You can program up to 200 messages. You must use the special message addresses in your program (see Sec. 3, ‘Address’) to simplify use of the messages. The following are the features of this function.

(a) In the program you define the total number of messages by using DISPB, and set ACT=1.It does not matter if ACT is already set at ‘1’. If, however, ACT = 0, DISPB will not process the messages at all. When ACT = 1, messages are displayed according to the contents of the message display request memory (addresses A0 to A24) and the message data table.
Relation between the message display request memory address and the message data table appears in Table 5.43.1.

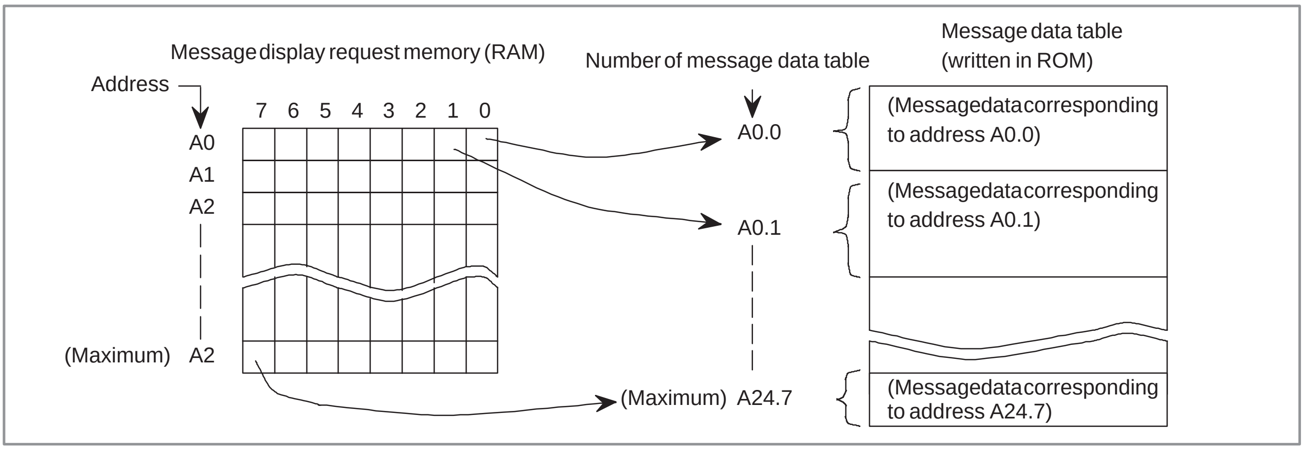


Fig. 5.43.1 Message display request memory and message data table

(i) Message display request memory (RAM)
Addresses A0 to A24 constitute a 200–bit area. This is a display request memory for up to 200 messages, each bit corresponding to a message.
If you want to display a message on the CRT, set the corresponding display request memory 1. Set 0 to erase the message of CRT.

(ii) Message data table
This table stores messages corres–ponding to the message display request bits. The table is stored in the EPROM together with the sequence program. Message data table numbers correspond to the message display request memory addresses.
The message data table capacity is prepared by the maximum capacity of a message, or, 255 characters (255 bytes). Produce a message data within this capacity.

A character prepared in CRT/MDI key consists of one byte, and 4 bytes are necessary for a message number (consisting of 4 characters) in the next item. A character not covered by the CRT/MDI keys requires two bytes (a half–width kana character) or four bytes (a kanji character or other full–width character). For details, see Section 5.43.6.

(iii) Message number
This message number consisting of 4 digits must always be defined at the start of each message data. The CRT display is as specified below by this message number.

- FS16–M/T, FS18–M/T, FS15, FS20, Power Mate–D (single path control), Power Mate–F and Power Mate–H

Message number	CNC screen	Display contents
1000 to 1999	Alarm message screen	Alarm message ● CNC is turned to alarm state.
2000 to 2099	Operator message screen	Operator message
2100 to 2999		Operator message (without message number) ● Only message data, no message number, is displayed.

- FS16–TT and FS18–TT

Message number	CNC screen	Display contents
1000 to 1999	Alarm message screen (The 1st tool post side)	Alarm message ● The 1st tool post side of CNC is turned to alarm state.
2000 to 2099	Operator message screen	Operator message
2100 to 2999		Operator message (without message number)
5000 to 5999	Alarm message screen (The 2nd tool post side)	Alarm message ● The 2nd tool post side of CNC is turned to alarm state. ● The displayed message number is a value by witch 4000 is subtracted from specified number.

- For 3–path control

Message number	CNC screen	Display contents
1000 to 1999	Alarm screen (on path 1)	Alarm message ● Path 1 is placed in the alarm state.
2000 to 2099	Operator message screen	Operator message
2100 to 2999		Operator message (with no message number)
5000 to 5999	Alarm screen (on path 2)	Alarm message ● Path 2 is placed in the alarm state. ● The displayed message number is a specified number from which 4000 is subtracted.
7000 to 7999	Alarm screen (on path 3)	Alarm message ● Path 3 is placed in the alarm state. ● The displayed message number is a specified number from which 6000 is subtracted.

- Power Mate–D (dual path control)

Message number	CNC screen	Display contents
1000 to 1999	Alarm message screen (The 1st path side)	Alarm message ● The 1st path side of CNC is turned to alarm state.
2000 to 2099	Operator message screen (The 1st path side)	Operator message
2100 to 2999		Operator message (without message number)
5000 to 5999	Alarm message screen (The 2nd path side)	Alarm message ● The 2nd path side of CNC is turned to alarm state. ● The displayed message number is a value by witch 4000 is subtracted from specified number.
6000 to 6099	Operator message screen (The 2nd path side)	Operator message ● The displayed message number is a value by witch 4000 is subtracted from specified number.
6100 to 6999		Operator message (without message number)

Moreover, the DPL/ MDI display with Power Mate is as specified below by this message number.

- Power Mate–D (single path control), Power Mate–F and Power Mate–H

Message number	CNC screen	Display contents
1000 to 1999	Alarm message screen	Message number ● CNC is turned to alarm state. ● Only message number, no message data, is displayed.
2000 to 2099 2100 to 2999	Operator message screen	Operator message ● Only message data, no message number, is displayed.

- Power Mate–D (dual path control)

Message number	CNC screen	Display contents
1000 to 1999	Alarm message screen (The 1st path side)	Message number ● The 1st path side of CNC is turned to alarm state. ● Only message number, no message data, is displayed.
2000 to 2099 2100 to 2999	Operator message screen (The 1st path side)	Operator message ● Only message data, no message number, is displayed.
5000 to 5999	Alarm message screen (The 2nd path side)	Message number ● The 2nd path side of CNC is turned to alarm state. ● Only message number, no message data, is displayed. ● The displayed message number is a value by witch 4000 is subtracted from specified number.
6000 to 6099 6100 to 6999	Operator message screen (The 2nd path side)	Operator message ● Only message data, no message number, is displayed.

NOTE

1 The number of message number which you can display at the same time to the alarm screen on DPL/MDI is up to 3.
2 The number of character which you can display to the operator message screen on DPL/MDI is up to 32 characters. The message data since the 33rd character is not displayed.
3 A "~" character (code A0H) is displayed as space character to the screen on DPL/MDI.
4 The DPL/MDI cannot display kanji (double–byte) characters.

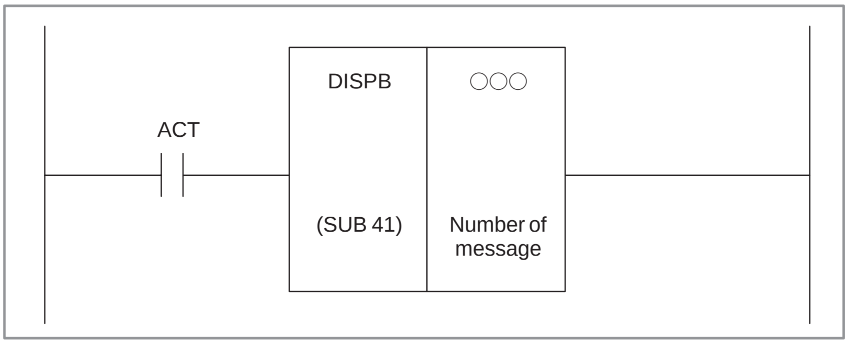
(b) You need not use numerical codes for message data input. Instead, when programming, directly key in the characters making up the messages (from the CRT/MDI keyboard). For the characters that CRT/MDI does not provide for, you must enter these characters by numerical data with special symbols "@". For details, refer to Subsec. 5.43.6).

(c) Use external data input command (described later) where you must combine the DISPB instruction with external data input function (for external tool compensation, external workpiece No. search, etc.). Such use of the DISPB instruction does not affect the interface of external data input function though the common interface is used between DISPB instruction and external data input function.

(d) If you write the message data items in the ROM after programming, you cannot change them any more (they will become fixed data items). However, you can still change and display only the numerical data forming part of the messages if you specify addresses storing the numerical data as the message data and assign the required numerical data in these addresses through sequence program.
Use of this function makes it possible for you to display frequently varying numerical data (such as tool number etc.) during automatic operations.

(e) A message is displayed on the CNC alarm message/operator message screen.
When using the DISPB instruction, you must satisfy the following conditions:
To use DISPB, the optional External Data Input function or External Message Display is necessary for CNC.

5.43.2 Format



5.43.3 Conditions

ACT=0 : Do not display messages on the CRT.
ACT=1 : Display the messages on the CRT.

5.43.4 Parameters

(a) Number of messages
Specifies the total number of messages (up to 200).

Function	PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
Number of messages	1 to 200	1 to 200	1 to 200	1 to 200	1 to 200	1 to 200	1 to 200	1 to 200	1 to 200	1 to 200	1 to 200	1 to 1000	1 to 200	1 to 200	1 to 1000	1 to 200	1 to 1000	1 to 1000

5.43.5 Numerical Data Display

To change the numerical data contained within the messages, enter in the messages the number of digits making up the data and the memory address to contain the data. To differentiate between the numerical data from the other message data, write it within [ ] in the message.
Since the brackets, [ ], are used to contain numerical data, they are not themselves treated as symbols to be included in the messages.

(a) Numerical data format

[Ibid, ○○○○]

→ Memory address storing the numerical data.
The numerical data must be of binary format.

→ Set the bid data after character i:

b : Number of bytes (1, 2, or 4)
i : Number of digits in the integer part (0 to 8)
d : Number of digits in the fractional part (0 to 8)

NOTE
1 Sum of integer part digits and fractional part digits must be within 8.
2 Blank is displayed for digits exceeding 8 digits.

(b) Example
The following message includes a 3–digit tool number at the spindle and the offset data (○.○○) for this tool. And these data is contained in a 2–byte memory address:

SPINDLE TOOL No. = [I 230,□□□□]
OFFSET DATA = [I 212, ΔΔΔΔ]

5.43.6 Defining Characters not found in the CRT/MDI

Message characters not covered by the CRT/MDI keys (kanji and half–width kana characters) can be input as follows:

(a) Half–width kana characters

(i) Data format
Numerical code enclosed by @ and @

(ii) Input method
Enter the numerical codes corresponding to the characters to be input, by referring to the character code table (Table 5.43). Each character requires two bytes. Characters covered by the CRT/MDI keys can also be input in this way.

(iii) Example
To input ATC? チョウサ OK when characters A, T, C, O, and K are registered in the CRT/MDI unit, enter the following:

```
ATC   @20  3F  C1  AE  B3  BB@OK
        _   ?   ﾁ   ｮ   ｳ   ｻ
```

(b) Kanji (full–width) characters

○ : Can be used
× : Cannot be used

Power Mate	FS20 FS21A	FS21B	FS21*i*	FS18A		FS16A			FS16B FS18B		FS16C FS18C		FS16*i* FS18*i*	FS15B		FS15*i*
PA1 PA3	SA1 SA3	SA1 SA3	SA1 SA5	SA1 SA3	SA2	SB SB3	SB2	SC SC3	SB3 SB4	SC3 SC4	SB5 SB6	SC3 SC4	SB5 SB6	NB	NB2	NB6
×	×	○	○	○	×	○	×	○	○	○	○	○	○	×	×	○

NOTE

1 The PMC–SA1 for the FS18–A can be used when the PMC management software series is 4071.
2 The PMC–SB for the FS16–A can be used when the PMC management software series is 4063.
3 The PMC–SC/SC3 for the FS16–A cannot be used depending on the series and edition of the CNC software.
4 For the FS16–A, set the following CNC parameter:
   – No. 6300 bit 6 = 0: Kanji characters are used for the DISPB instruction (default).
   1: Kanji characters are not used for the DISPB instruction.
   When kanji characters are used, the DISP instruction cannot be used.
5 On the CNC, the external data input option or external message option must be selected.

(i) Data format
Numerical code enclosed by @02 and 01@

(ii) Input method
Enter the numerical codes corresponding to the characters to be input, by referring to the kanji, hiragana, and special code table in Appendix O. Each character requires four bytes.

(iii) Example

To input ATC? 調査 OK when characters A, T, C, O, and K are registered in the CRT/MDI unit, enter the following:

```
ATC   @20  3F@@02 4434 3A3A 01@OK
       _    ?      調   査
```

NOTE

1 To define @, enter @40...@, where 40 is the code corresponding to @.

@40 . . . . . @
Code for @

2 To renew the message line displayed on the CRT/MDI screen, input as:
@ OA @ at the end of the data.

3 When using numerical codes, @ code occupies 1 byte, and space code occupies 2 bytes. (Space code = 20, 2 and 0 occupies 1 byte each).

4 The following control codes are used:
02 : 2–byte code (kanji and hiragana characters)
01 : 1–byte code (alphanumerics and half–width kana characters)
Do not specify 02 or 01 between @02 and 01@, as follows. The characters may not be correctly displayed.
@02 ... 02 ... 01@   @02 ... 01 ... 01@

Table 5.43.6 Character code table

	2	3	4	5	A	B	C	D
0	␣ (Space)	0	@	P	to	_ *3)	タ	ミ
1	!	1	A	Q	。	ア	チ	ム
2	#	2	B	R	「	イ	ツ	メ
3	#	3	C	S	」	ウ	テ	モ
4	$	4	D	T	、	エ	ト	ヤ
5	%	5	E	U	·	オ	ナ	ユ
6	&	6	F	V	ヲ	カ	ニ	ヨ
7	’	7	G	W	ァ	キ	ヌ	ラ
8	(	8	H	X	ィ	ク	ネ	リ
9	)	9	I	Y	ゥ	ケ	ノ	ル
A	*	:	J	Z	ェ	コ	ハ	レ
B	+	;	K	[	ォ	サ	ヒ	ロ
C	,	<	L	¥	ャ	シ	フ	ワ
D	± *1)	=	M	]	ュ	ス	ヘ	ン
E	·	>	N	∧	ョ	セ	ホ	゛
F	/	?	O	__*2)	ッ	ソ	マ	゜

*1) Minus, *2) Under bar, *3) Long bar *4) Dakuten *5) Han–dakuten

5.43.7 Notes when this Functional Instruction is Used in Subroutine

Refer to Sec. 9.3.

5.43.8 Foreign Language Display

Power Mate/ FS21A	FS20/ FS21B	FS18A	FS16A	FS16B FS18B		FS16C FS18C		FS21*i*	FS16*i* FS18*i*		FS15B	FS15*i*
PA1 PA3	SA1 SA3	SA1, SA2 SA3	SB, SB2, SB3 SC, SC3	SB3 SC3	SB4 SC4	SB5 SC3	SB6 SC4	SA1 SA5	SB5	SB6	NB NB2	NB6
×	×	×	×	×	○	×	○	×	×	○	×	×

(a) General

In the message data areas corresponding to contiguous message display request memory locations, message data can be displayed in any of several languages.

The language in which a message is displayed is selected by shifting the message display request bit according to the address bit shift amount set in setting parameter 2.

A0.0 Language 1
A0.1 Language 2
A0.2 Language 3
A0.3 Language 4
A0.4 Language 5

When A0.0 is turned on after setting the message display request bit shift amount to 2, the message display request bit is shifted by 2 bits to display language 3.

The parameters set on the setting parameter 2 screen are listed below.

- MESSAGE SHIFT VALUE
  Message display request bit shift amount
- MESSAGE SHIFT START ADDRESS
  Start bit address of the message display request bit area to be shifted

(b) Examples

Example 1:

Message data in any of four languages is set starting at A0.0 in the order of Japanese, English, Italian, German, Japanese and so on. The Italian message data is displayed.

Set the parameters as follows:

MESSAGE SHIFT VALUE : 2
MESSAGE SHIFT START ADDRESS : A0.0
(MESSAGE SHIFT VALUE =
0:Japanese/1:English/2:Italian/3:German)

Manipulate A0.0, A0.4, A1.0, A1.4, and so forth with the ladder.

Message table

A0.0	Japanese 1
A0.1	English 1
A0.2	Italian 1
A0.3	German 1
A0.4	Japanese 2
A0.5	English 2
A0.6	Italian 2
A0.7	German 2
;	:
Am.n	:

When A0.0 is turned on, Italian 1 is displayed. (The message data is shifted by 2 bits).

When A0.4 is turned on, Italian 2 is displayed. (The message data is shifted by 2 bits).

Example 2:

As common alarm messages, English message data is displayed with A0.0 through A9.7. Operator messages are set starting at A10.0 in the order of Japanese, English, Italian, German and so on, and German message data is displayed.
Set the parameters as follows:
MESSAGE SHIFT VALUE : 3
MESSAGE SHIFT START ADDRESS : A10.0
(MESSAGE SHIFT VALUE =
0:Japanese/1:English/2:Italian/3:German)
Manipulate A10.0, A10.4, A11.0, A11.4, and so forth with the ladder.
When any of A0.0 to A9.7 is turned on, the message corresponding to the bit is displayed.

Message table

A0.0	English A	(ALARM)
A0.1	English B	(ALARM)
A0.2	English C	(ALARM)
~		~
A10.0	Japanese 1	(OPE)
A10.1	English 1	(OPE)
A10.2	Italian 1	(OPE)
A10.3	German 1	(OPE)
A10.4	Japanese 2	(OPE)
A10.5	English 2	(OPE)
A10.6	Italian 2	(OPE)
A10.7	German 2	(OPE)
:	:	
:	:	
A m.n	:	

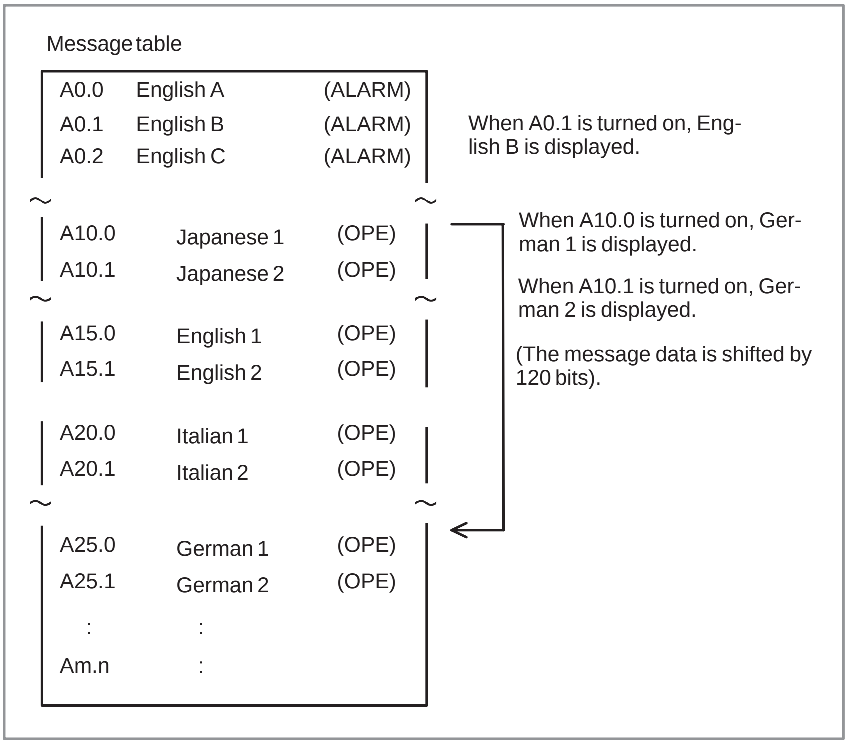
When A0.1 is turned on, English B is displayed.

When A10.0 is turned on, German 1 is displayed. (The message data is shifted by 3 bits).

When A10.4 is turned on, German 2 is displayed. (The message data is shifted by 3 bits).

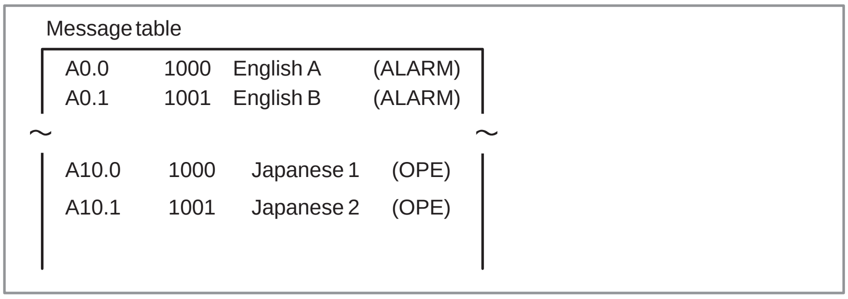
Example 3:

As common alarm messages, English message data is displayed with A0.0 through A9.7. Operator messages are set starting at A10.0 in the order of Japanese, English, Italian, German and so on, with 40 successive messages assigned to each language. For these messages, German message data is displayed.
Set the parameters as follows:
MESSAGE SHIFT VALUE : 120 (40 x 3)
MESSAGE SHIFT START ADDRESS : A10.0
(MESSAGE SHIFT VALUE =
0:Japanese/40:English/80:Italian/120:German)
Manipulate A10.0 through A14.7 with the ladder.
When any of A0.0 to A9.7 is turned on, the message corresponding to the bit is displayed.



(c) Notes

The same message number should be assigned to a message in each language that has the same meaning.



5.44 EXIN (EXTERNAL DATA INPUT)

5.44.1 Function

This instruction is used for external data (external tool compensation, external message function, external program number search, external workpiece coordinates shift, etc.) input. You must use this instruction when combining the message display instruction (DISP, DISPB) with the external data input function. If you are not used DISP or DISPB, you need not use this instruction either. Instead, use the external data input interface PMC↔CNC directly in your program.
The DISPB instruction uses the interface between the PMC and CNC provided by the external data input function during display. The DISP instruction prevents the interface signal transferred between the PMC and CNC from being changed due to external cutter compensation or others.
You can use the EXIN instruction only when the PMC↔CNC interface is of BMI (Basic Machine Interface) and optional external data input function is provided with CNC.
An 4–byte control data as described below is required for external data input function (option).

In PMC–SA5/SB5/SB6, the expended specification that needs 6 bytes of control data is supported. With this setting, the extended operation can use ED16 to ED31 signals (for program number O8 digits etc.). To use the extended specification, it is necessary to set to NC parameter 6300#7 (EEXIN)=1.

NC parameter

	#7	#6	#5	#4	#3	#2	#1	#0
6300	EEXIN							

data format : bit type

EEXIN : EXIN function of PMC

0 : basic specification

1 : extended specification

> **NOTE**
> To use program number O8 digits, the option with program number O8 digits and NC parameter 6300#7 (EEXIN)=1 are necessary.

5.44.2 Format

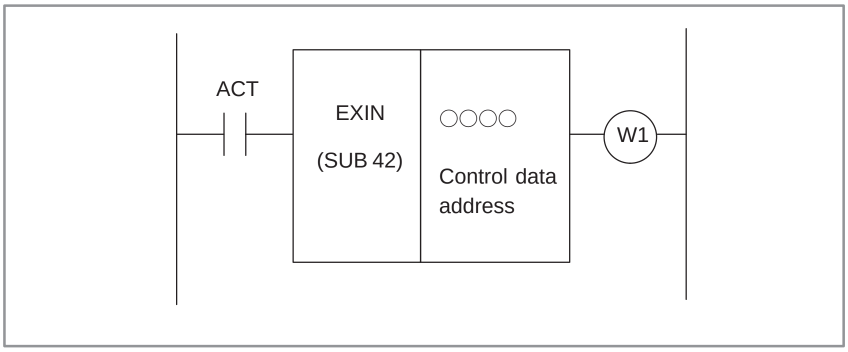


Fig. 5.44.2 EXIN instruction format

5.44.3 Control Conditions

ACT=0 : Do not process external data input/output.
ACT=1 : Process external data input/output.
ACT is to be maintained ’1’ till the end of external data input/output. After external data input, reset ACT (W1 = 1).

5.44.4 Parameter

(a) Control data (except PMC–NB/NB2)

The control data needs 4 continuous bytes from the specification address. The path is specified to the 1st byte. The addresses G0 to G2 of the interface from PMC to NC are specified by after 3 bytes. For 2nd path, the addresses G1000 to G1002 are specified. Fot 3rd path, the addresses G2000 to G2002 are specified. (Be sure to set the strobe signal (ESTB) to ON.)

In PMC–SA5/SB5/SB6, in case of the extended specification (program number O8 digits etc.), a control data is extended. In this case, the control data address needs 6 continuous bytes from the specified address. THe path is specified in the 1st byte. The addresses G0 to G2 and G210 to G211 of the interface from PMC to NC are specified in later 5 bytes. For 2nd path, the addresses G1000 to G1002 and G1210 to G1211 are specified. For 3rd path, the addresses G2000 to G2002 and G2210 to G2111 are specified. (Be sure to set the strobe signal (ESTB) to ON.)

Basic specification

CTL+0	HEAD NO.
+1	ED0 to ED7
+2	ED8 to ED15
+3	EA0 to EA6, ESTB
+4	

Extended specification
(program number O8 digits etc.)

CTL+0	HEAD NO.
+1	ED0 to ED7
+2	ED8 to ED15
+3	ED16 to ED23
+4	ED24 to ED31
+5	EA0 to EA6, ESTB
+6	

[For single path control]

CTL+0 : 0
CTL+1 to CTL+3 : Data to be specified for G0 to G2
In case of the extended specification (program number O8 digits etc.), it sets CTL+1 to CTL+5 as follows.
CTL+1 to CTL+2 : Data to be specified for G0 to G1
CTL+3 to CTL+4 : Data to be specified for G210 to G211
CTL+5 : Data to be specified for G2

[For multi path control]

(i) 1st path
CTL+0 : 0
CTL+1 to CTL+3 : Data to be specified for G0 to G2
In case of the extended specification (program number O8 digits etc.), it sets CTL+1 to CTL+5 as follows.
CTL+1 to CTL+2 : Data to be specified for G0 to G1
CTL+3 to CTL+4 : Data to be specified for G210 to G211
CTL+5 : Data to be specified for G2

(ii) 2nd path
CTL+0 : 2
CTL+1 to CTL+3 : Data to be specified for G1000 to G1002
In case of the extended specification (program number O8 digits etc.), it sets CTL+1 to CTL+5 as follows.
CTL+1 to CTL+2 : Data to be specified for G1000 to G1001
CTL+3 to CTL+4 : Data to be specified for G1210 to G1211
CTL+5 : Data to be specified for G1002

(iii) 3rd path
CTL+0 : 3
CTL+1 to CTL+3 : Data to be specified for G2000 to G2002
In case of the extended specification (program number O8 digits etc.), it sets CTL+1 to CTL+5 as follows.
CTL+1 to CTL+2 : Data to be specified for G2000 to G2001
CTL+3 to CTL+4 : Data to be specified for G2210 to G2211
CTL+5 : Data to be specified for G2002

NOTE
Refer to the ”Series 16 or 18 Connection Manual” for detailed data to be specified concerning external data input.

(b) A consecutive area in eight bytes is necessary as the control data.
In 15– M/ T, set command data in this CTL+0 – +7 by the same data form as G32– 39 of BMI interface.
In 15– TT, set command data in this CTL+0 – +7 by the same data form as G112– 119 of BMI interface.

CTL+0	EISTB, EOREND etc.
+1	EIA0 to EIA7
+2	EID32 to EID39
+3	EID40 to EID47
+4	EID0 to EID7
+5	EID8 to EID15
+6	EID16 to EID23
+7	EID25 to EID31

A consecutive area in 16 bytes is necessary as the control data.
In 15–M/T, set command data in first CTL+0 – +7 by the same data form as G32–39 of BMI interface. The data output from NC is written in CTL+8 – +15 in the same data form as BMI interface F32–39.
In 15–TT, set command data in first CTL+0 – +7 by the same data form as G112–119 of BMI interface. The data output from NC is written in CTL+8 – +15 in the same data form as BMI interface F112–119.

CTL+0	EISTB, EOREND etc
+1	EIA0 to EIA7
+2	EID32 to EID39
+3	EID40 to EID47
+4	EID0 to EID7
+5	EID8 to EID15
+6	EID16 to EID23
+7	EID25 to EID31

to to

to to

CTL+8	EOSTB, EIREND etc
+9	EOA0 to EOA7
+10	EOD32 to EOD39
+11	EOD40 to EOD47
+12	EOD0 to EOD7
+13	EOD8 to EOD15
+14	EOD16 to EOD23
+15	EOD25 to EOD31

NOTE
Refer to the following manuals in detail of BMI interface.
"FANUC Series 15–MODEL B Connection Manual (BMI interface)"

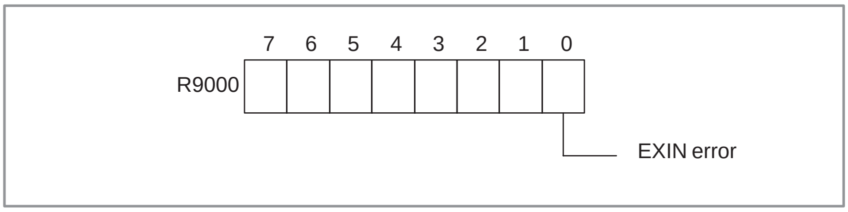
5.44.5
End of Transfer (W1)

This indicates end of transfer of external data. This transfer end condition shows the end of a series of external data input sequence. This functional instruction executes a series of transfer sequence, and finally sets ESTB = 0 in the PMC → NC interface. As a result, W1 is set to 1 (W1 = 1) after confirming that EREND = 0.
When W1 = 1, transfer of data is over. Reset ACT now.

CAUTION
1 The EXIN command cannot input multiple external data items at the same time. Be sure to issue the next EXIN command (ACT = 1) after external data transfer ends (W1 = 1).
2 Be sure to specify an interlock when the external data input function is used by commands other than the function commands, DISP, DISPB, and EXIN.

5.44.6
Operation Output Register

If any of the following errors occurs during external data input, the bit in the operation output register is set. In this case, external data transfer ends (W1 = 1).



(Description of errors)

- When the EXIN command (ACT = 1) is started, the strobe signal (ESTB) or EREND signal is already on. The external data may be input by commands other than the function commands, DISP, DISPB, and EXIN.
- An invalid head number was specified for 16–TT or 18–TT. (Data other than 0 to 2 was specified.)
- The specification of HEAD.NO is incorrect.
(Data other than 0 to 3 is set for 3–path control.)

5.44.7
Notes when this Functional Instruction is Used in Subroutine

Refer to Sec. 9.3.

5.45 WINDR (READING CNC WINDOW DATA)

5.45.1 Function

This function reads various data items via the window between the PMC and the CNC.
The "WINDR" is classified into two types. One type completes reading a data during one scan time. Another type completes reading a data during a few scan time. The former is called the function of a high–speed response and the latter is called the function of a low–speed response.

5.45.2 Format

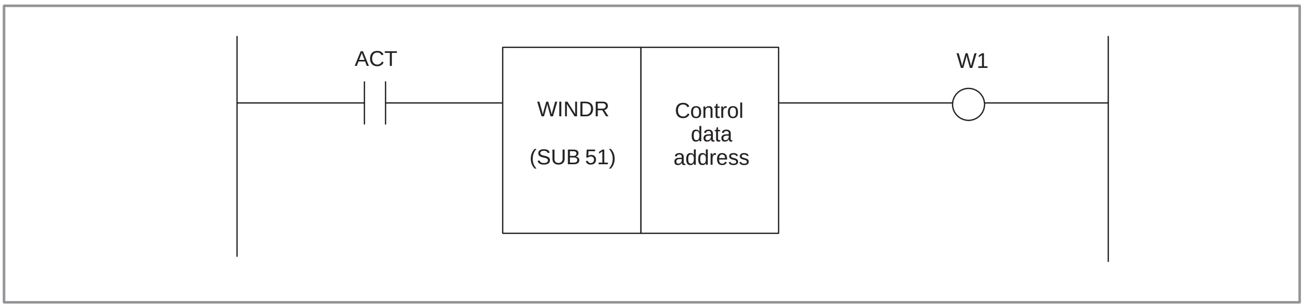


Fig. 5.45.2

5.45.3 Control Condition

ACT=0 : The WINDR function is not executed.

ACT=1 : The WINDR function is executed. Using the function of a high–speed response, it is possible to read the data continuously by always keeping ACT on. However, using the function of a low–speed response, as soon as reading a data is completed, reset "ACT" once (ACT=0).

5.45.4 Parameter

(a) Control data address

The PMC byte address is used to specify the area where control data is stored.

5.45.5 Control Data

CTL+0	Function code
+2	Completion code
+4	Data length
+6	Data number
+8	Data attribute
+10	Read data
∼	∼
+n	

* Set the control data area by sequence program before executing the “WINDR” or “WINDW”.

* Only the size of the read data is necessary for the data area below to to “CTL+10” usually.

See Appendix B WINDOW FUNCTION DESCRIPTION.

CAUTION

1 In the functional instructions “WINDR” and “WINDW”, the control data area may be temporarily rewritten. Therefore, set the control data area by sequence program before the “WINDR” or “WINDW” is executed even when you specify the none volatile memory area like “D” address for the control data area. Because, when the power supply is turned off during the control data is rewritten, this rewritten data may be memorized in a none volatile memory. Therefore, note that the “WINDR” or “WINDW” might be executed with the wrong control data when the power supply is turned on next if the control data area is not set by sequence program.

2 Set the control data in the same program level as the “WINDR” or “WINDW” is executed. If you set the control data in the different program level, note that the “WINDR” or “WINDW” might not be executed correctly, because the control data is rewritten during the execution of “WINDR” or “WINDW”.

3 In the diagnosis screen, it might be seen that the value of control data is changing. This is not abnormal. Because the display processing and the execution processing of a sequence program are asynchronously executed. Therefore, the value when the control data is rewritten (above–mentioned) is occasionally displayed. Even in this case, the “WINDR” or “WINDW” is executed correctly.

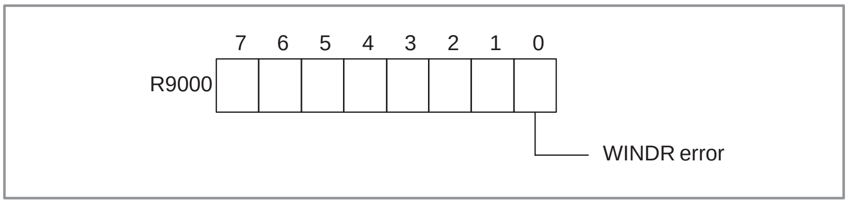
5.45.6 Reading Completion (W1)

W1=0 : “W1” is usually reset. The “W1=0” indicates that the “WINDR” is not executed or the ”WINDR” being executed now.

W1=1 : “W1” is set when the reading a data is completed by the reading command (ACT=1). If the function of a low–speed response is used, as soon as reading a data is completed, reset “ACT” (ACT=0).

5.45.7 Operation Output register

If an error occurs during execution of the “WINDR” or “WINDW”, the bit in the operation output register is set. At the same time, the reading completion is set (W1=1). Details of the error are output to the completion code (CTL+2) in the control data area. See Appendix B WINDOW FUNCTION DESCRIPTION.



5.45.8 Notes when this Functional Instruction is Used in Subroutine

When you use the function of a low–speed response, there are a few limitation. Refer to “9.3 NOTE FOR SUBROUTINES WHEN YOU USE SUBROUTINES” When you use the function of a high–speed response, there is no limitation.

5.46 WINDW (WRITING CNC WINDOW DATA)

5.46.1 Function

This function writes various data items via the window between the PMC and the CNC.
The “WINDR” is classified into the function of a low–speed response.

5.46.2 Format

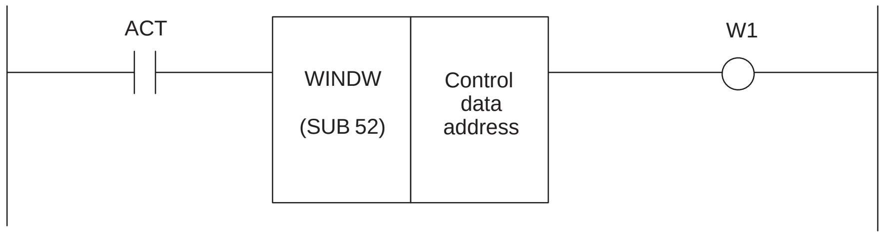


Fig. 5.46.2

5.46.3 Control Condition

ACT=0 : The WINDW function is not executed.

ACT=1 : The WINDW function is executed. As soon as writing a data is completed, reset “ACT” once (ACT=0).

5.46.4 Parameter

(a) Control data address

The PMC byte address is used to specify the area where control data is stored.

5.46.5 Control Data

CTL+0	Function code
+2	Completion code
+4	Data length
+6	Data number
+8	Data attribute
+10	Writing data
~	~
+42	

* Set the control data area by sequence program before executing the ”WINDR” or ”WINDW”.

See Appendix B WINDOW FUNCTION DESCRIPTION.

CAUTION

1 In the functional instructions “WINDR” and “WINDW”, the control data area may be temporarily rewritten. Therefore, set the control data area by sequence program before the “WINDR” or “WINDW” is executed even when you specify the none volatile memory area like “D” address for the control data area. Because, when the power supply is turned off during the control data is rewritten, this rewritten data may be memorized in a none volatile memory. Therefore, note that the “WINDR” or “WINDW” might be executed with the wrong control data when the power supply is turned on next if the control data area is not set by sequence program.

2 Set the control data in the same program level as the “WINDR” or “WINDW” is executed. If you set the control data in the different program level, note that the “WINDR” or “WINDW” might not be executed correctly, because the control data is rewritten during the execution of “WINDR” or “WINDW”.

3 In the diagnosis screen, it might be seen that the value of control data is changing. This is not abnormal. Because the display processing and the execution processing of a sequence program are asynchronously executed. Therefore, the value when the control data is rewritten (above–mentioned) is occasionally displayed. Even in this case, the “WINDR” or “WINDW” is executed correctly.

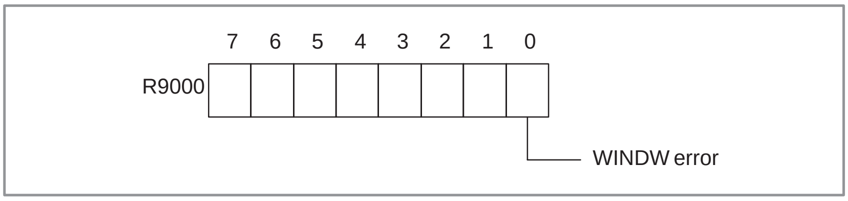
5.46.6 Writing Completion (W1)

W1=0 : "W1" is usually reset. The "W1=0" indicates that the "WINDW" is not executed or the "WINDW" being executed now.

W1=1 : "W1" is set when the writing a data is completed by the writing command (ACT=1). As soon as writing a data is completed, reset "ACT" (ACT=0).

5.46.7 Operation Output Register

If an error occurs during execution of the "WINDR" or "WINDW", the bit in the operation output register is set. At the same time, the writing completion is set (W1=1). Details of the error are output to the completion code (CTL+2) in the control data area. See Appendix B WINDOW FUNCTION DESCRIPTION.



5.46.8 Notes when this Functional Instruction is Used in Subroutine

When you use the function of a low–speed response, there are a few limitation. Refer to "9.3 NOTE FOR SUBROUTINES WHEN YOU USE SUBROUTINES"

5.47 ARBITRARY FUNCTIONAL INSTRUCTIONS

5.47.1 FNC 90 to 97 (Arbitrary Functional Instructions) (Only for PMC–RC/RC3/NB/NB2)

5.47.1.1 Function

These functional instructions (SUB90 to SUB97) are used to execute the arbitrary functional instructions. These instructions consist of the addresses specifying the start condition, process end output, and control condition.

5.47.1.2 Format

Fig.5.47.1.2 shows the notation format. Table 5.47.1.2 shows the coding format.

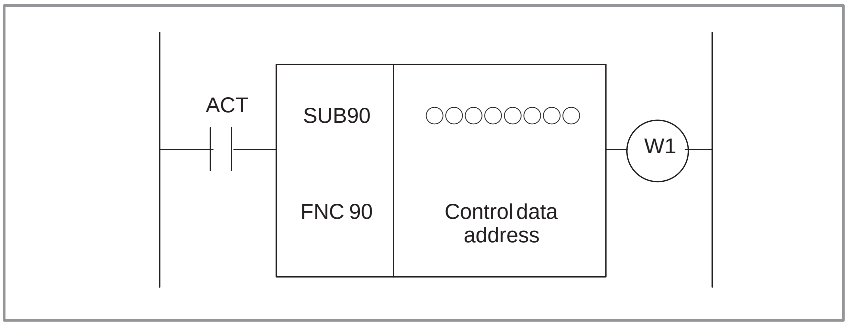


Fig. 5.47.1.2 FUNC 90 notation format

Table 5.47.1.2 FUNC 90 coding format

Step number	Com-mand	Address No.	Bit No.	Remarks
1	RD	○○○○.	○	ACT
2	SUB	90		FUNC90 command
3	(PRM)	○○○○		Control data address
4	WRT	○○○○.	○	W1

5.47.1.3 Control Condition

(a) Execution command (ACT)
This is used as the start condition of an arbitrary functional instruction.

5.47.1.4 Parameter

(a) Control data address
Specifies the first address in the control data area.

5.47.1.5 Control Data

Set the control data to be used by an arbitrary functional instruction.
If the control data is determined as follows, for example, the person who created the ladder program determines a control address to set the control data using the ladder program.

Control data addres	7·····	2	1	0
CTL + 0		CNO	UPDOWN	RST
CTL + 1				
CTL + 2	Counter number			
CTL + 3				

5.47.1.6 Process End Output (W1)

This is used as the process end output of an arbitrary functional instruction.

NOTE
If this functional instruction is displayed by the PCLAD display function, an arbitrary functional instruction is displayed as SUB9X, FNC99X.

5.47.2 Creating an Arbitrary Function

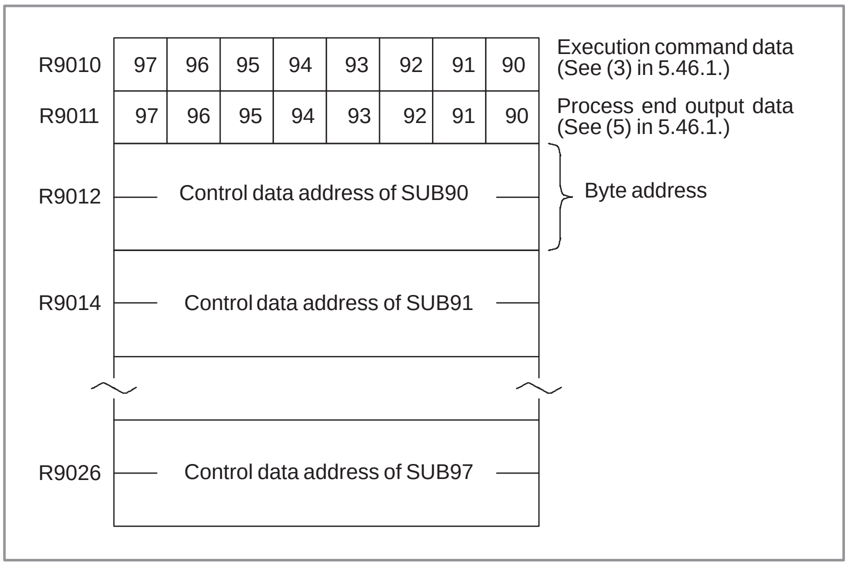
5.47.2.1 Arbitrary Functional Instruction and Interface

(a) Execution command (ACT)
The contents of the execution command can be referenced by bit 1 at R9010.

(b) Control data address
The address where the control data is stored can be referenced in the byte address format at R9012 or later.

(c) Process end output (W1)
The data output when the process terminates can be referenced by bit 1 at R9011.

5.47.2.2 Use of the R Field



5.47.2.3 Creating an Arbitrary Function

Reference the start condition (ACT) of the arbitrary function by bit 1 at R9010. Reference the address at which the control data is stored in the byte address format by the fields at R9012 and later. Set the end signal (W1) of an arbitrary function in bit 1 at R9011. For example, to execute the arbitrary function using SUB90, reference the start condition by R9010.0. Reference the control data address in the byte address format by R9012. Set the end signal at R9011.0.

5.48 MMCWR (READING MMC WINDOW DATA) (OTHER THAN PMC–PA1/PA3)

5.48.1 Function

This command reads up to 32 bytes of data via the window between PMC and MMC. The data can be determined as required between the PMC ladder program and MMC application program.

5.48.2 Format

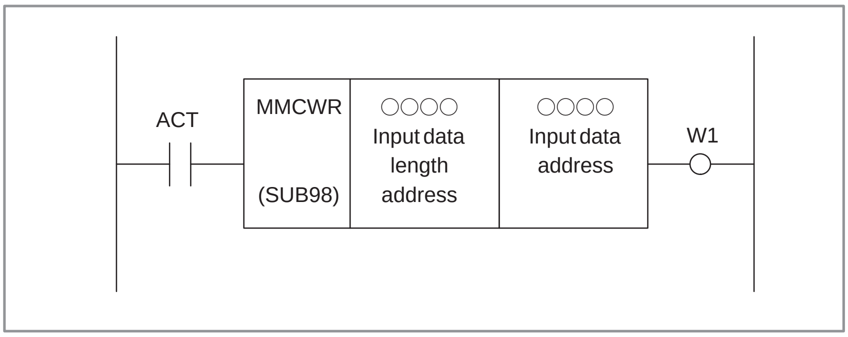


Fig. 5.48.2 MMCWR instrument format

Table 5.48.2 MMCWR coding format coding sheet

Step number	Com-mand	Address No.	Bit No.	Remarks
1	RD	○○○.	○	ACT
2	SUB	98		
3	(PRM)	○○○○		Input data length address
4	(PRM)	○○○○.		Input data address
5	WRT	○○○.	○	W1, processing completion

5.48.3 Control Condition

ACT=0 : The MMCWR function is not executed.
ACT=1 : The MMCWR function is executed. Hold ACT = 1 until processing is completed and specify ACT = 0 immediately after processing is completed (W1 = 1).

5.48.4 Parameters

(a) Input data length address (two bytes)
Specifies the length of input data transferred from MMC. When transfer is completed, the length of data actually transferred is stored. The maximum data length is 32 bytes.

(b) Input data address
Specifies the area containing data transferred from MMC. An area large enough for the specified input data length is required.

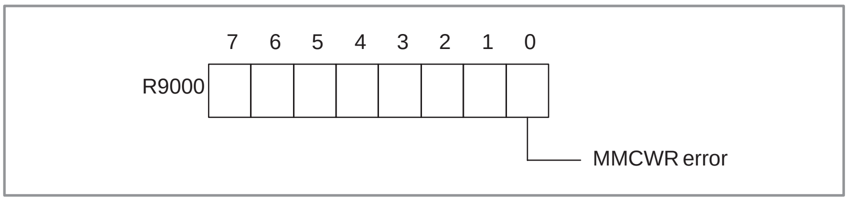
5.48.5 Processing Completion (W1)

W1=0 : This value is usually set. W1 = 1 indicates that processing is completed. As soon as processing is completed, specify ACT=0.

W1=1 : This value is set when data transfer from MMC is completed or if an error occurs.

5.48.6 Operation Output Register

If an MMC window transfer error occurs, the bit in the operation output register is set to indicate the error. If an error occurs, the transferred data is not stored in the input data area.



5.48.7 Completion Status Information

The completion status information is specified in R9002 and R9003. The completion codes and contents, W1, and error bits are as follows:

–11 ⋯ Initialization at MMC is not completed.
(W1 = 0, R9000#0 = 0)
–10 ⋯ Processing is in progress (W1 = 0, R9000#0 = 0)
0 ⋯ Processing is completed.(W1 = 1, R9000#0 = 0)
2 ⋯ Data length error (W1 = 1, R9000#0 = 1)
(0, a negative value, or a value exceeding 33 bytes was specified for the data length. The length of data actually transferred exceeded the specified value.)
6 ⋯ MMC is not provided (W1 = 1, R9000#0 = 1)

5.48.8 Notes when this Functional Instruction is Used in Subroutine

Refer to Sec. 9.3.

5.49 MMCWW (WRITING MMC WINDOW DATA) (OTHER THAN PMC–PA1/PA3)

5.49.1 Function

This command writes data containing up to 32 bytes via the window between PMC and MMC. The data can be determined as required between the PMC ladder program and MMC application program.

5.49.2 Format

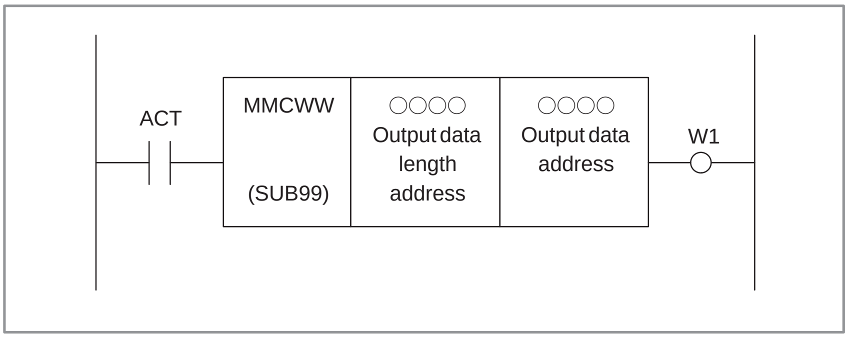


Fig. 5.49.2 MMCWW instruction format

Table 5.49.2 MMCWW coding format coding sheet

Step number	Com-mand	Address No.	Bit No.	Remarks
1	RD	○○○.	○	ACT
2	SUB	99		
3	(PRM)	○○○○		Output data length address
4	(PRM)	○○○○.		Output data address
5	WRT	○○○.	○	W1, processing completion

5.49.3 Control Condition

ACT=0 : The MMCWW function is not executed.

ACT=1 : The MMCWW function is executed. Hold ACT = 1 until processing is completed and specify ACT = 0 immediately after processing is completed.

5.49.4 Parameters

(a) Output data length address (two bytes)
Specifies the length of output data transferred to MMC. The maximum data length is 32 bytes.

(b) Output data address
Specifies the area storing data to be transferred to MMC. An area large enough for the specified output data length is required.

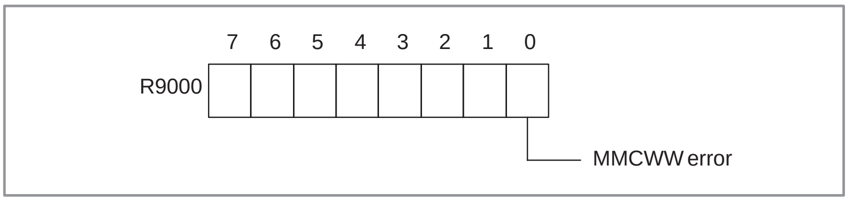
5.49.5 Processing Completion (W1)

W1=0 : This value is usually set. W1 = 1 indicates that processing is completed. As soon as processing is completed, specify ACT=0.

W1=1 : This value is set when data transfer to MMC is completed or if an error occurs.

5.49.6 Operation Output Register

If an MMC window transfer error occurs, the bit in the operation output register is set to indicate the error. If an error occurs, the transferred data is not transferred to MMC.



5.49.7 Completion Status Information

The completion status information is specified in R9002 and R9003. The completion codes and contents, W1, and error bits are as follows:

–11 ··· Initialization at MMC is not completed.(W1 = 0, R9000#0 = 0)
–10 ··· Processing is in progress. (W1 = 0, R9000#0 = 0)
0 ··· Processing is completed. (W1 = 1, R9000#0 = 0)
2 ··· Data length error (W1 = 1, R9000#0 = 1)
(0, a negative value, or a value exceeding 33 bytes was specified for the data length.)
6 ··· MMC is not provided. (W1 = 1, R9000#0 = 1)

5.49.8 Notes when this Functional Instruction is Used in Subroutine

Refer to Sec. 9.3.

5.50 MOVB (TRANSFER OF 1 BYTE)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.50.1 Function

The MOVB instruction transfers 1–byte data from a specified source address to a specified destination address.

5.50.2 Format

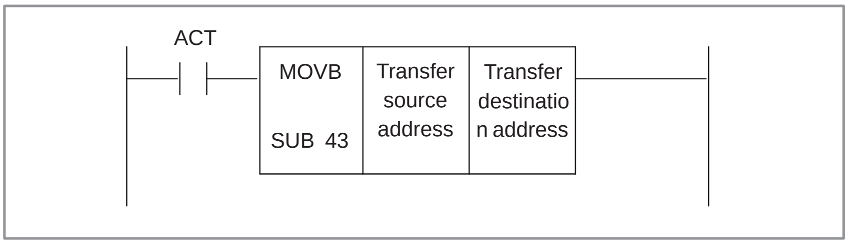


Fig. 5.50.2 MOVB instruction format

5.50.3 Control Conditions

(a) Execution specification
ACT=0 : No data is transferred.
ACT=1 : One–byte data is transferred.

5.51 MOVW (TRANSFER OF 2 BYTES)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.51.1 Function

The MOVW instruction transfers 2–byte data from a specified source address to a specified destination address.

5.51.2 Format

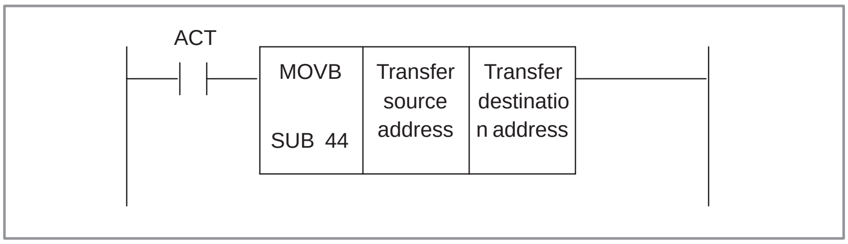


Fig. 5.51.2 MOVB instruction format

5.51.3 Control Conditions

(a) Execution specification
ACT=0 : No data is transferred.
ACT=1 : Two–byte data is transferred.

5.52 MOVN (TRANSFER OF AN ARBITRARY NUMBER OF BYTES)

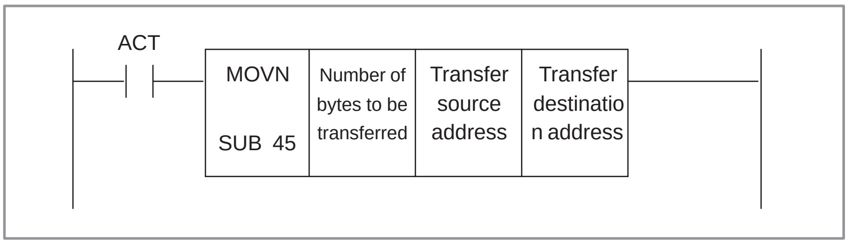
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.52.1 Function

The MOVN instruction transfers data consisting of an arbitrary number of bytes from a specified source address to a specified destination address.

5.52.2 Format



5.52.3 Control Conditions

(a) Execution specification
ACT=0 : No data is transferred.
ACT=1 : A specified number of bytes are transferred.

5.52.4 Parameters

(a) Number of bytes to be transferred
Specify the number of bytes to be transferred. An odd number can also be specified. A number from 1 to 200 can be specified.

5.53 DIFU (RISING EDGE DETECTION)

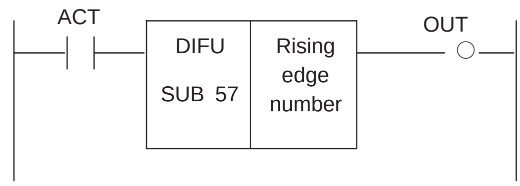
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.53.1 Function

The DIFU instruction sets the output signal to 1 for one scanning cycle on a rising edge of the input signal.

5.53.2 Format



5.53.3 Control Conditions

(a) Input signal
On a rising edge (0→1) of the input signal, the output signal is set to 1.

(b) Output signal
The output signal level remains at 1 for one scanning cycle of the ladder level where this functional instruction is operating.

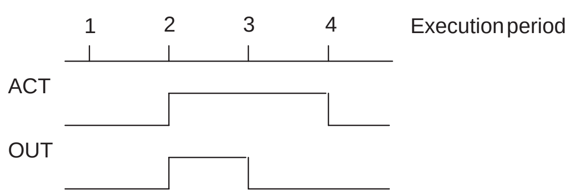
5.53.4 Parameters

(a) Rising edge number

Model	PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2
Rising edge number	–	1 to 256	–	–	1 to 256	1 to 256	–	–	1 to 256	1 to 500	1 to 256	1 to 500	–	1 to 256	1 to 500	1 to 256	1 to 500

WARNING
If the same number is used for another DIFU instruction or a DIFD instruction (described later) in one Ladder diagram, operation is not guaranteed.

5.53.5 Operation



5.54 DIFD (FALLING EDGE DETECTION)

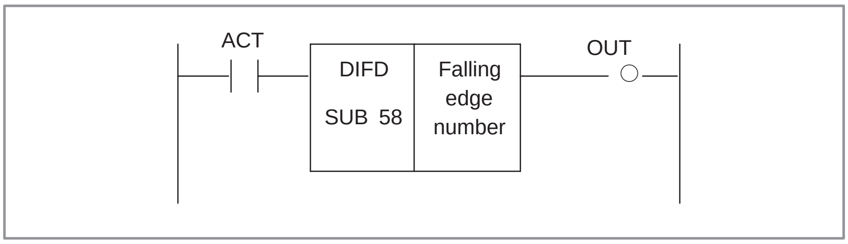
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.54.1 Function

The DIFD instruction set the output signal to 1 for one scanning period on a falling edge of the input signal.

5.54.2 Format



5.54.3 Control Conditions

(a) Input signal
On a falling edge(1→0)of the input signal, the output signal is set to 1.

(b) Output signal
The output signal level remains at 1 for one scanning period of the ladder level where this functional instruction is operating.

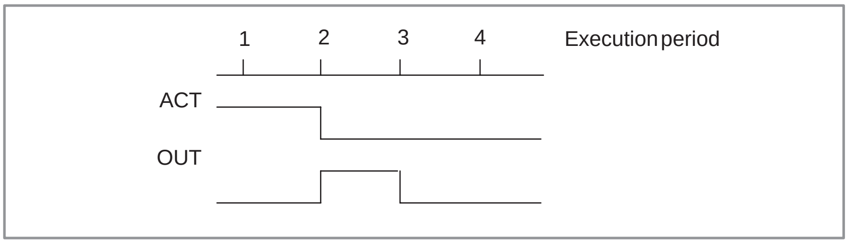
5.54.4 Parameters

(a) Falling edge number

Model	PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2
Falling edge number	–	1 to 256	–	–	1 to 256	1 to 256	–	–	1 to 256	1 to 500	1 to 256	1 to 500	–	1 to 256	1 to 500	1 to 256	1 to 500

WARNING
If the same number is used for another DIFD instruction or a DIFU instruction (described above) in one ladder diagram, operation is not guaranteed.

5.54.5 Operation



5.55 EOR (EXCLUSIVE OR)

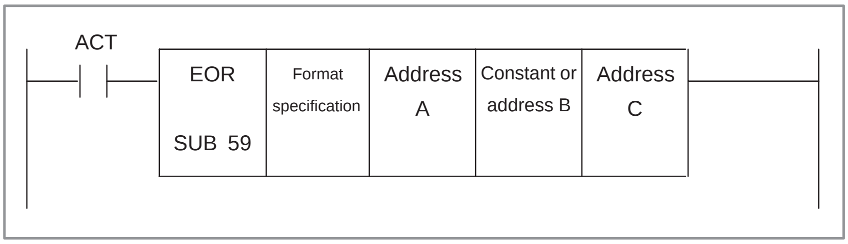
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.55.1 Function

The EOR instruction exclusive–ORs the contents of address A with a constant (or the contents of address B), and stores the result at address C.

5.55.2 Format



5.55.3 Control Conditions

(a) Input signal
ACT=0 : The EOR instruction is not executed.
ACT=1 : The EOR instruction is executed.

5.55.4 Parameters

(a) Format specification
Specify a data length (1, 2, or 4 bytes), and an input data format (constant or address specification).

(b) Address A
Input data to be exclusive–ORed. The data that is held starting at this address and has the data length specified in format specification is treated as input data.

(c) Constant or address B
Input data to be exclusive–ORed with. When address specification is selected in format specification, the data that is held starting at this address and has the data length specified in format specification is treated as input data.

(d) Address C
Address used to store the result of an exclusive OR operation. The result of an exclusive OR operation is stored starting at this address, and has the data length specified in format specification.

5.55.5 Operation

When address A and address B hold the following data:

Address A	1	1	1	0	0	0	1	1
Address B	0	1	0	1	0	1	0	1

The result of the exclusive OR operation is as follows:

Address C	1	0	1	1	0	1	1	0

5.56 LOGICAL AND

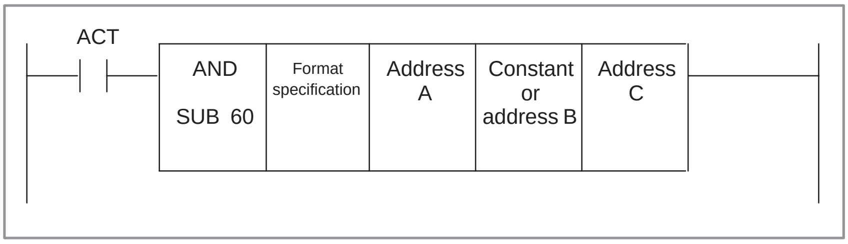
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.56.1 Function

The AND instruction ANDs the contents of address A with a constant (or the contents of address B), and stores the result at address C.

5.56.2 Format



5.56.3 Control Conditions

(a) Input signal
ACT=0 : The AND instruction is not executed.
ACT=1 : The AND instruction is executed.

5.56.4 Parameters

(a) Format specification
Specify a data length (1, 2, or 4 bytes), and an input data format (constant or address specification).

(b) Address A
Input data to be ANDed. The data that is held starting at this address and has the data length specified in format specification is treated as input data.

(c) Constant or address B
Input data to be ANDed with. When address specification is selected in format specification, the data that is held starting at this address and has the data length specified in format specification is treated as input data.

(d) Address C
Address used to store the result of an AND operation. The result of an AND operation is stored starting at this address, and has the data length specified in format specification.

5.56.5 Operation

When address A and address B hold the following data:

Address A	1	1	1	0	0	0	1	1
Address B	0	1	0	1	0	1	0	1

The result of the AND operation is as follows:

Address C	0	1	0	0	0	0	1	0

5.57 LOGICAL OR

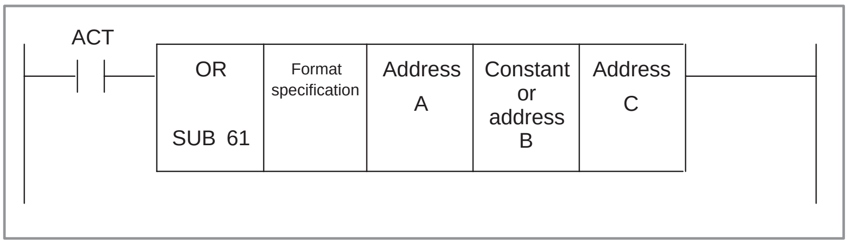
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.57.1 Function

The OR instruction ORs the contents of address A with a constant (or the contents of address B), and stores the result at address C.

5.57.2 Format

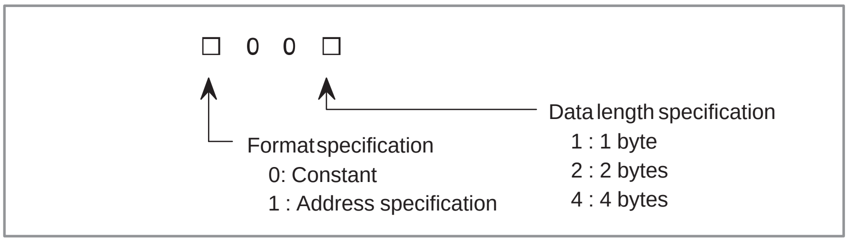


5.57.3 Control Conditions

(a) Input signal
ACT=0 : The OR instruction is not executed.
ACT=1 : The OR instruction is executed.

5.57.4 Parameters

(a) Format specification
Specify a data length (1, 2, or 4 bytes), and an input data format (constant or address specification).



(b) Address A
Input data to be ORed. The data that is held starting at this address and has the data length specified in format specification is treated as input data.

(c) Constant or address B
Input data to be ORed with. When address specification is selected in format specification, the data that is held starting at this address and has the data length specified in format specification is treated as input data.

(d) Address C
Address used to store the result of an OR operation. The result of an OR operation is stored starting at this address, and has the data length specified in format specification.

5.57.5 Operation

When address A and address B hold the following data:

Address A	1	1	1	0	0	0	1	1
Address B	0	1	0	1	0	1	0	1

The result of the OR operation is as follows:

Address C	1	1	1	1	0	1	1	1

5.58 NOT (LOGICAL NOT)

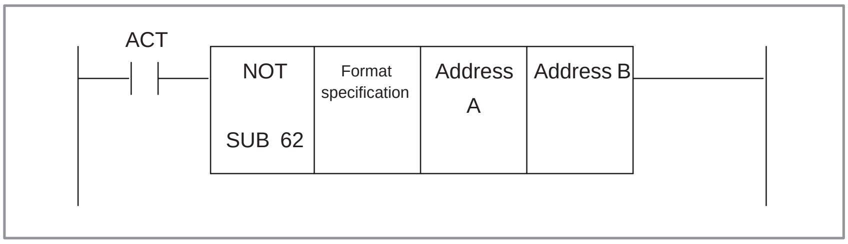
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.58.1 Function

The NOT instruction inverts each bit of the contents of address A, and stores the result at address B.

5.58.2 Format

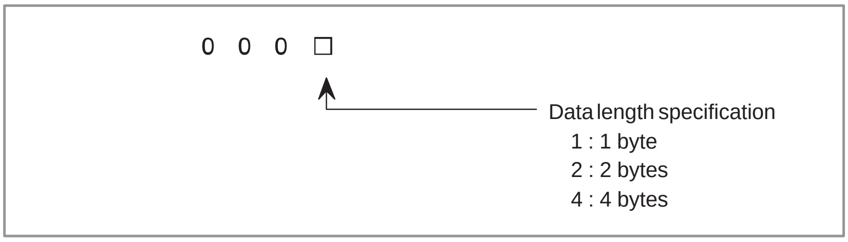


5.58.3 Control Conditions

(a) Input signal
ACT=0 : The NOT instruction is not executed.
ACT=1 : The NOT instruction is executed.

5.58.4 Parameters

(a) Format specification
Specify a data length (1, 2, or 4 bytes).



(b) Address A
Input data to be inverted bit by bit. The data that is held starting at this address and has the data length specified in format specification is treated as input data.

(c) Address B
Address used to output the result of a NOT operation. The result of a NOT operation is stored starting at this address, and has the data length specified in format specification.

5.58.5 Operation

When address A holds the following data:

Address A	1	1	1	0	0	0	1	1

The result of the NOT operation is as follows:

Address B	0	0	0	1	1	1	0	0

5.59 MMC3 R (MMC–III WINDOW DATA READ)

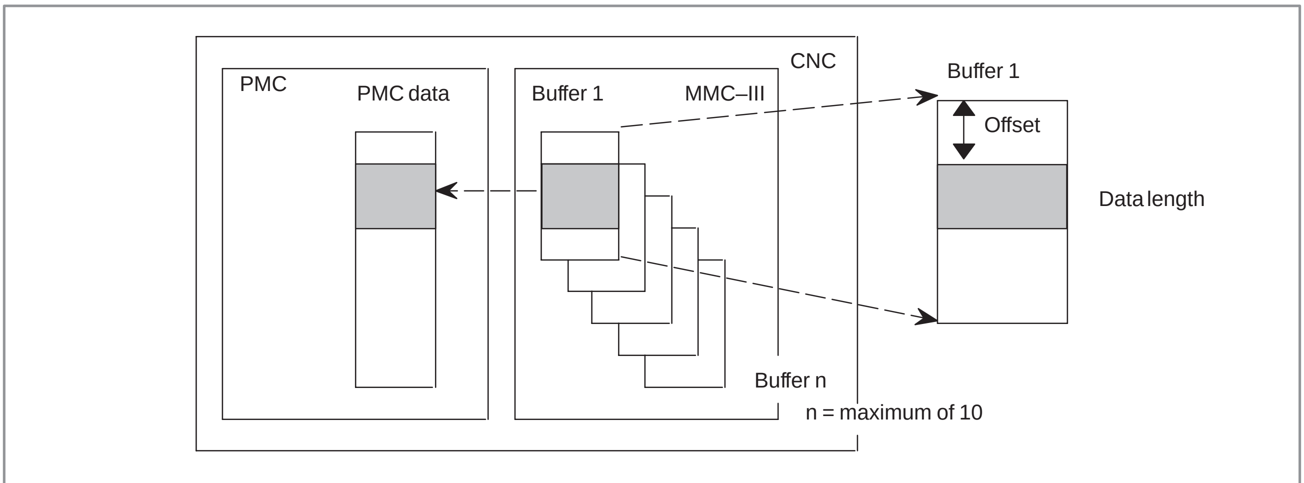
○ : Can be used
Δ : Can be used (with some restrictions)
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	×	Δ	○	○	○	○	○	○	○	×	×	○	○	○	○	○	×

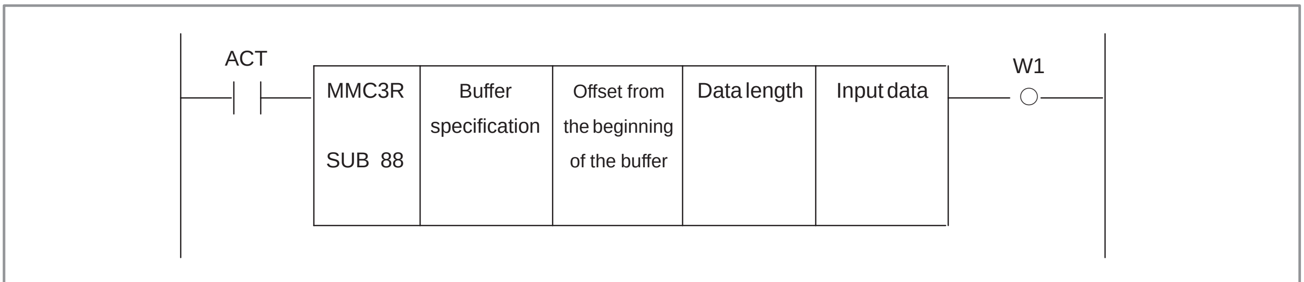
NOTE
This functional instruction cannot be used with SA1 of the Series 16*i*/18*i*/21*i*–MODEL A.

5.59.1 Function

The MMC3R instruction reads MMC–III application data via a PMC–MMC window. Which buffer in the MMC–III is to be read can be specified. The contents of read data can be freely determined by a PMC Ladder program and MMC–III application program.



5.59.2 Format



5.59.3 Control Conditions (ACT)

ACT=0 : The MMC3R instruction is not executed.
ACT=1 : Data is read.

5.59.4 Parameters

(a) Address for storing buffer specifications (2 bytes)
A buffer from which data is to be read is specified. Up to 10 buffers can be specified. Specify the address where the buffer specification is held.

> **NOTE**
> For the method of buffer registration, refer to the relevant MMC–III manual.

(b) Address for storing an offset from the beginning of a buffer (2 bytes)
An offset from the beginning of a read buffer is specified. Specify the address where the offset is held.

(c) Data length storage address (2 bytes)
The length of data to be read from the MMC–III is specified. Specify the address where the length of data is held. The maximum allowable data length is 256 bytes.

(d) Input data storage address
Specify the address where data to be read from the MMC–III is stored. A contiguous area not smaller than the length of data specified in c) above is required.

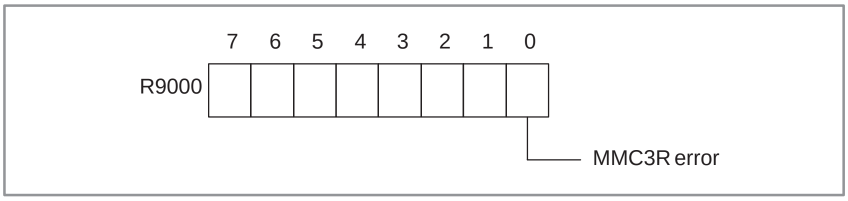
5.59.5 Processing Completion (W1)

W1=0 : When ACT = 0, W1 = 0 is set. If W1 = 0 is set when ACT = 1, it indicates that read processing is in progress.

W1=1 : Indicates that read processing has terminated. Whether read processing has terminated normally or abnormally can be checked with the state of R9000 described below.

5.59.6 Operation Output Register

When W1 indicates the termination of read processing, a termination state is set.



MMC3R=0 : Normal termination
MMC3R=1 : Abnormal termination

5.59.7 Completion Status Information

When ACT = 1, completion status information is set in the operation register R9002.

–11 : MMC initialization not completed (W1=0, R9000#0=0)
0 : Normal termination (W1=1, R9000#0=0)
2 : Data length error (W1=1, R9000#0=1)
The specified length of data is 0, negative data is specified, or the maximum allowable data length is exceeded.
6 : The MMC–III is not attached. (W1=1, R9000#0=1)
3 : Buffer specification error (W1=1, R9000#0=1)

5.59.8 Notes when this Functional Instruction is Used in Subroutine

Refer to Sec. 9.3.

5.60 MMC3W (MMC–III WINDOW DATA WRITE)

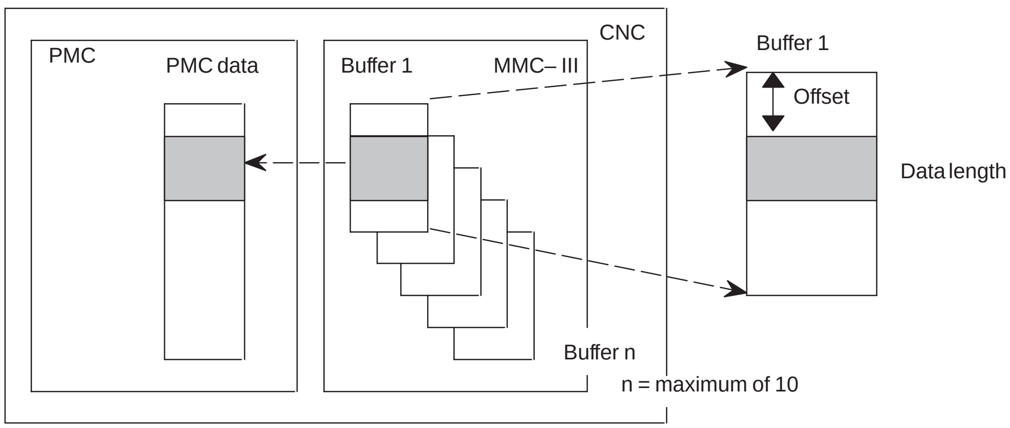
○ : Can be used
Δ : Can be used (with some restrictions)
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	×	Δ	○	○	○	○	○	○	○	×	×	○	○	○	○	○	×

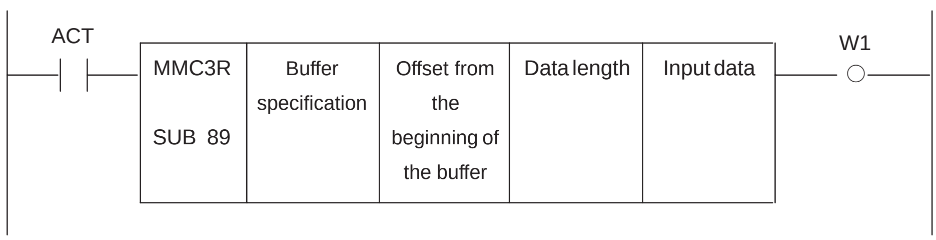
NOTE
This functional instruction cannot be used with SA1 of the Series 16*i*/18*i*/21*i*–MODEL A.

5.60.1 Function

The MMC3W instruction writes data to MMC–III application data via a PMC–MMC window. Which buffer in the MMC–III is to be written to can be specified. The contents of write data can be freely determined by a PMC Ladder program and MMC–III application program.



5.60.2 Format



5.60.3 Control Conditions (ACT)

ACT=0 : The MMC3W instruction is not executed.
ACT=1 : Data is written.

5.60.4 Parameters

(a) Address for storing buffer specifications (2 bytes)
A buffer to which data is to be written is specified. Up to 10 buffers can be specified. Specify the address where the buffer specification is held.

> **NOTE**
> For the method of buffer registration, refer to the relevant MMC–III manual.

(b) Address for storing an offset from the beginning of a buffer (2 bytes)
An offset from the beginning of a write buffer is specified. Specify the address where the offset is held.

(c) Data length storage address (2 bytes)
The length of data to be written to the MMC–III is specified. Specify the address where the length of data is held. The maximum allowable data length is 256 bytes.

(d) Output data storage address
Specify the address where data to be written to the MMC–III is stored. A contiguous area not smaller than the length of data specified in c) above is required.

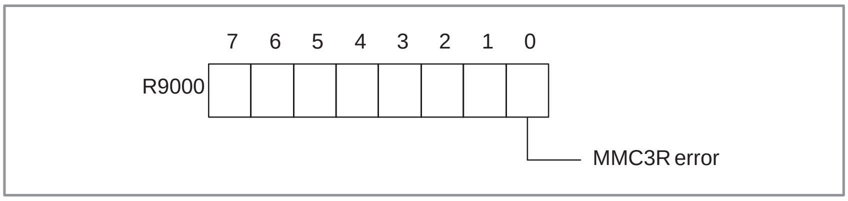
5.60.5 Processing Completion (W1)

W1=0 : When ACT = 0, W1 = 0 is set. If W1 = 0 is set when ACT = 1, it indicates that write processing is in progress.

W1=1 : Indicates that write processing has terminated. Whether write processing has terminated normally or abnormally can be checked with the state of R9000 described below.

5.60.6 Operation Output Register

When W1 indicates the termination of write processing, a termination state is set.



MMC3W=0 : Normal termination
MMC3W=1 : MMC3R = 1: Abnormal termination

5.60.7 Completion Status Information

When ACT = 1, completion status information is set in the operation register R9002.

–11 : MMC initialization not completed (W1=0, R9000#0=0)
0 : Normal termination (W1=1, R9000#0=0)
2 : Data length error (W1=1, R9000#0=1)
The specified length of data is 0, negative data is specified, or the maximum allowable data length is exceeded.
6 : The MMC–III is not attached. (W1=1, R9000#0=1)
3 : Buffer specification error (W1=1, R9000#0=1)

5.60.8 Notes when this Functional Instruction is Used in Subroutine

Refer to Sec. 9.3.

5.61 SPCNT (SPINDLE CONTROL)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	○	○	○

5.61.1 Function

SPCNT performs the following processing using spindle speed data (32–bit binary data) that is input from the NC or some other device to the PMC:

(a) Gear selection (Up to four gears from GR1 to GR4 can be used.)

(b) Calculating a spindle motor rotation command (13–bit binary data) when automatic gear selection is enabled

(c) Calculating a spindle motor rotation command (13–bit binary data) when direct gear selection is enabled

(d) Calculating a spindle motor rotation command when a spindle override is specified

(e) Clamping the spindle motor speed to an upper or lower limit

As shown in Fig.5.61.1, a spindle motor rotation command is calculated from the spindle speed data. The maximum value (8191) of the spindle motor command is equivalent to an analog voltage at 10V.

> **NOTE**
> The motor speed is clamped after spindle override is specified.

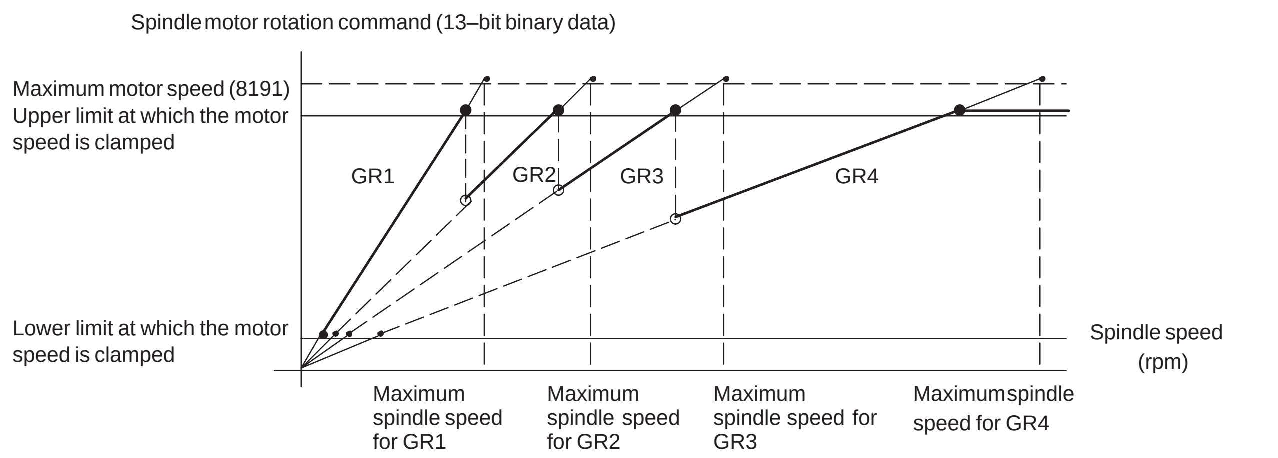
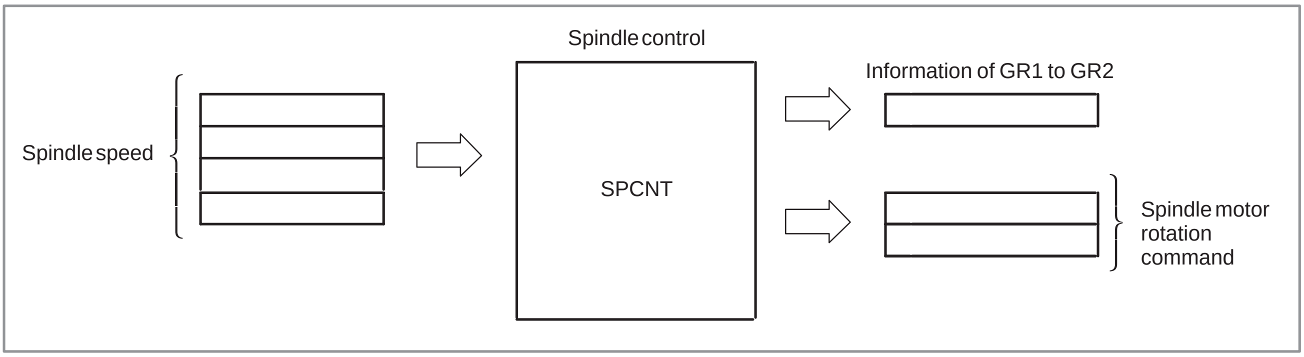


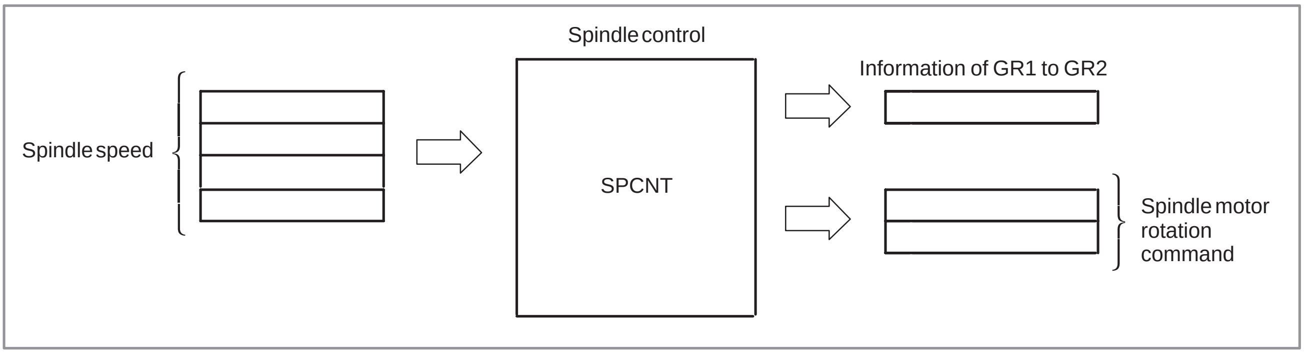
Fig. 5.61.1 Spindle Speeds and Corresponding Spindle Motor Rotation Commands

The spindle motor rotation command is calculated as 13–bit binary data. If the spindle amplifier is a D/A converter provided in the machine and can only handle 12–bit binary data, for example, the calculated spindle motor rotation command must be halved before being output (shifted right one bit position in a shift register).

(i) Spindle control with automatic gear selection
This functional instruction uses spindle speed data (32–bit binary data) and the maximum spindle speeds set in parameters GR1 to GR4 of this functional instruction to select a gear, calculate the spindle motor rotation command for that selected gear, and output the result to the control data address.
Based on this output information, the sequence program must perform gear switching as necessary and output the rotation command to the spindle motor.

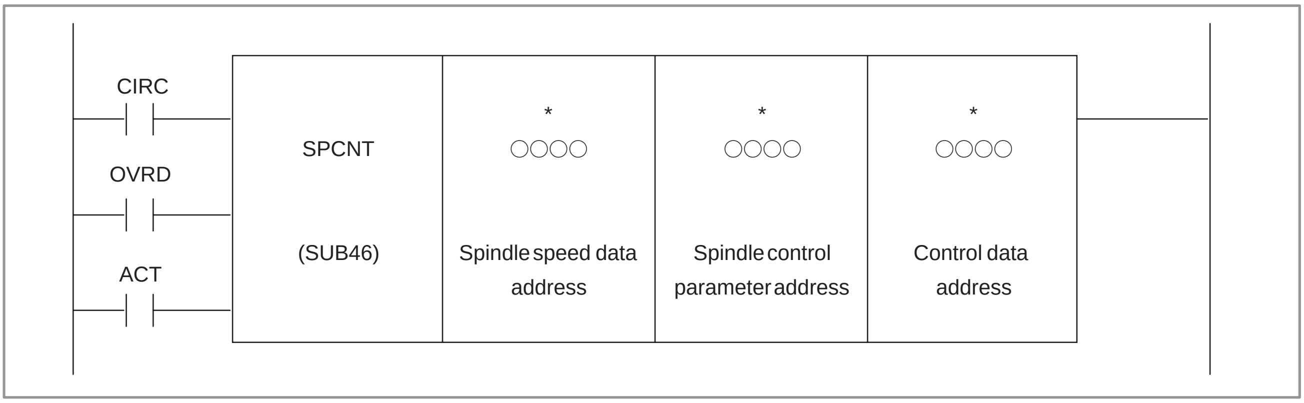


(ii) Spindle control with direct gear selection
When direct gear selection is set, this functional instruction does not perform gear selection. A gear to be used is selected by the sequence program. The target gear must be set at the control data address, which is a parameter of this functional instruction, using the sequence program.
According to the set gear, the functional instruction calculates and outputs a spindle motor rotation command. In this case, the spindle motor rotation command has a linear relationship with the spindle speed. The line for the selected gear is assumed to extend to its lower limit (indicated by a dotted line). See Fig.5.61.1.



The spindle motor speed can be clamped at the upper and lower limits also with direct gear specification. When the CNC performs constant surface speed control, spindle control with direct gear specification is generally performed.

5.61.2 Format



5.61.3 Control Conditions

(a) Direct gear specification (CIRC)
CIRC=0 : Disables direct gear specification.
(Enables automatic gear selection.)
CIRC=1 : Enables direct gear specification.

(b) Override specification (OVRD)
OVRD=0 : Disables the override function
OVRD=1 : Enables the override function.

(c) Instruction execution specification (ACT)
ACT=0 : The SPCNT instruction is not executed.
ACT=1 : The SPCNT instruction is executed.

5.61.4 Parameters

(a) Spindle speed data address
Specifies an even–numbered address at which the spindle speed data (32–bit binary data) is stored.

(b) Spindle control parameter address
Specifies an even–numbered address at which the parameters for spindle control are stored. Binary data is set in contiguous 24–byte memory locations starting at the specified address.

Spindle control parameter +0	Lower spindle motor speed limit data
Spindle control parameter +4	Upper spindle motor speed limit data
Spindle control parameter +8	Maximum spindle speed for gear 1
Spindle control parameter +12	Maximum spindle speed for gear 2
Spindle control parameter +16	Maximum spindle speed for gear 3
Spindle control parameter +20	Maximum spindle speed for gear 4
Spindle control parameter +24	

This 24–byte memory area is specified by addressing, and so it can be allocated in any addressable memory location. For this type of data, however, a data table in nonvolatile memory is most suitable. For maintenance convenience, the memory area should be allocated in the first data table (table group 1).

(i) Lower spindle motor speed limit data
Sets the lower spindle motor speed limit obtained from the following expression:

$$\text{Lower spindle motor speed limit data} = \frac{\text{Minimum speed (rpm) specified for the spindle motor}}{\text{Maximum speed (rpm) obtainable by the spindle motor}} \times 8191$$

A value from 0 to 8191 can be specified as the lower speed limit data. The maximum spindle motor speed is achieved when 10 V is applied to the motor.

(ii) Upper spindle motor speed limit data
Sets the upper spindle motor speed limit obtained from the following expression:

$$\text{Upper spindle motor speed limit data} = \frac{\text{Maximum speed (rpm) specified for the spindle motor}}{\text{Maximum speed (rpm) obtainable by the spindle motor}} \times 8191$$

(iii) Maximum spindle speed for GR1
Sets a maximum spindle speed (rpm) for GR1. The maximum spindle speed must be set in this parameter even when GR1 gear is not provided. The maximum spindle speed is the speed of the spindle when the motor operates at its maximum speed.

(iv) Maximum spindle speed for GR2
Sets a maximum spindle speed (rpm) for GR2. When GR2 is not provided, this parameter must be set to 0.

(v) Maximum spindle speed for GR3
Sets a maximum spindle speed (rpm) for GR3. When GR3 is not provided, this parameter must be set to 0.

(vi) Maximum spindle speed for GR4
Sets a maximum spindle speed (rpm) for GR4. When GR4 is not provided, this parameter must be set to 0.

(c) Control data address
Contiguous 4–byte memory locations starting at the even–numbered address specified in the control data address parameter must be specified.

	7	6	5	4	3	2	1	0	
Specified address+0	R08	R07	R06	R05	R04	R03	R02	R01	Spindle motor rotation command
Specified address+1				R13	R12	R11	R10	R09	
Specified address+2					GR4	GR3	GR2	GR1	Spindle gear selection
Specified address+3	SOV128	SOV64	SOV32	SOV16	SOV8	SOV4	SOV2	SOV1	Spindle override

(i) Spindle gear selection

7	6	5	4	3	2	1	0
				GR4	GR3	GR2	GR1

[For automatic gear selection]
This functional instruction finds an appropriate gear using the spindle speed data and the maximum spindle speed for each gear, then outputs the result to GR1 to GR4.

[For direct gear selection]
The sequence program sets the gear to be used in GR1 to GR4.

This functional instruction calculates the spindle motor rotation commands for all speeds from the upper motor speed limit to the lower speed limit (extended portion indicated by dotted line). See Fig. 5.61.

(ii) Spindle motor rotation command

7	6	5	4	3	2	1	0
R08	R07	R06	R05	R04	R03	R02	R01
			R13	R12	R11	R10	R09

The spindle motor rotation command (13–bit binary data) calculated by this functional instruction is set at these control data addresses. This instruction specifies a spindle motor rotation command with a spindle override applied.

(iii) Spindle override

7	6	5	4	3	2	1	0
SOV128	SOV64	SOV32	SOV16	SOV8	SOV4	SOV2	SOV1

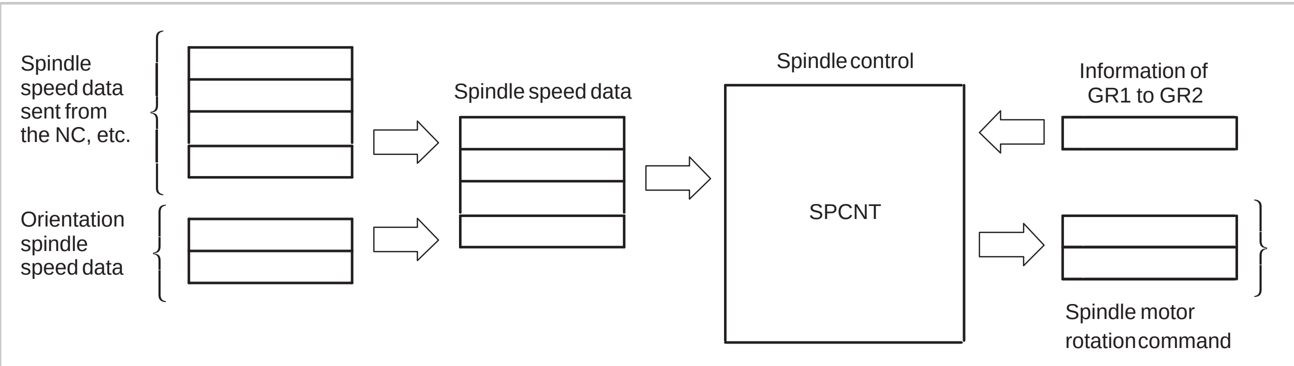
The sequence program must set a spindle override in binary. A spindle override from 0% to 255% can be set in binary.

5.61.5 Use of Spindle Control

Spindle control is primarily used to control the spindle speed during normal cutting. It can, however, also be used to:

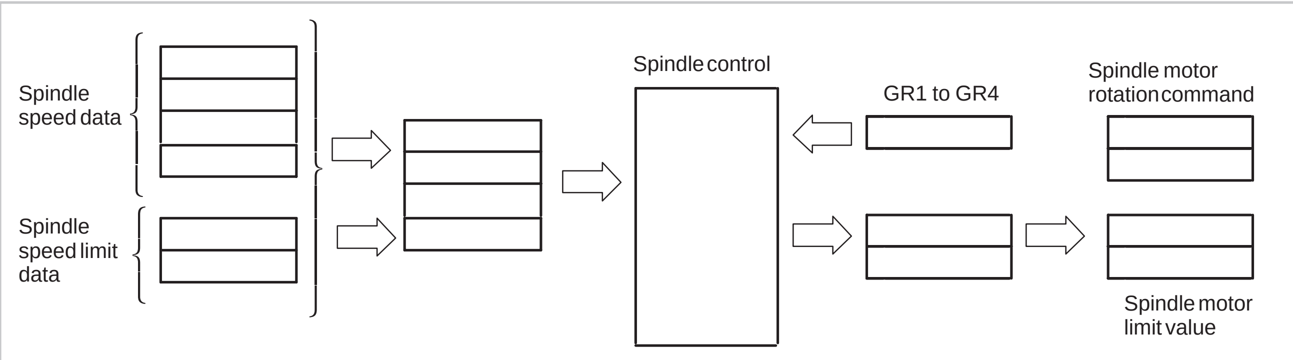
(a) Rotate the spindle motor at a specific speed when the gear is switched
The sequence program can output appropriate 13–bit binary data as a spindle motor rotation command to rotate the spindle motor at a specific speed, without using this functional instruction.

(b) Rotate the spindle at a specific speed during spindle orientation
This is enabled by specifying appropriate spindle speed data in the functional instruction (SPCNT). During spindle orientation, the spindle is rotated at the specified orientation spindle speed with the currently selected gear (gear selection is not performed). Gear selection is disabled by setting CIRC to 1 (direct gear specification).



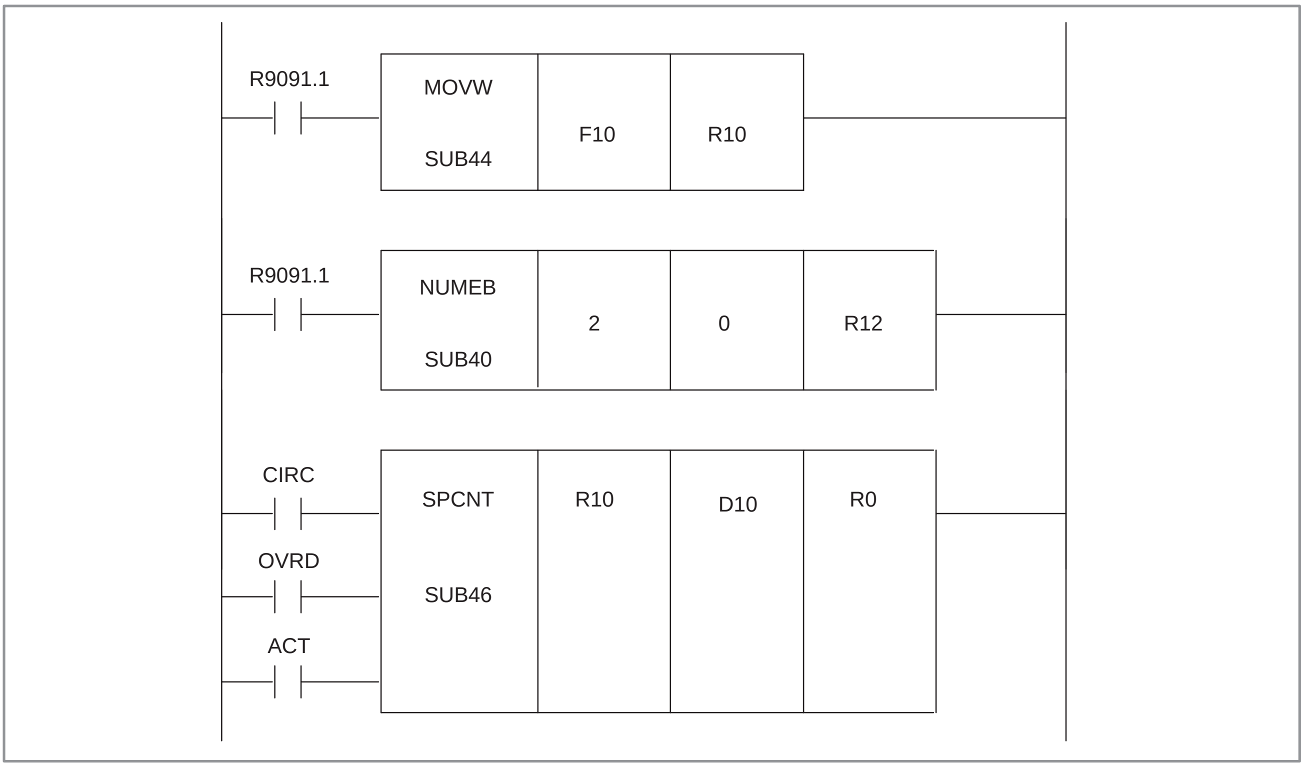
(c) Control the spindle in a tapping cycle
In a tapping cycle, spindle rotation is reversed at the bottom of a hole. Using the HIGH gear to reverse the rotation requires a lower analog voltage than using the LOW gear. So, using the HIGH gear reduces the machining time.
To widen the usable range of the HIGH gear, set CIRC to 1 to disable automatic gear selection.

(d) Clamp the spindle speed
When the BMI interface is used between the NC and PMC, spindle should be controlled by the PMC (sequence program), as described in the BMI manual.
Clamping the spindle speed is one of the spindle control operations. The spindle control functional instruction SPCNT (SUB46) can be used to clamp the spindle speed. The clamping method is outlined below. For precise control, conform to the specifications of the machine supplied by the machine tool builder.



(e) Example
Suppose that the parameters are set as follows:
Minimum speed specified for the spindle motor = 1000 rpm
Maximum speed specified for the spindle motor = 35000 rpm
Maximum speed obtainable by the spindle motor = 40000 rpm
(Maximum speed when 10 V is applied to the spindle motor)
Maximum speed for gear 1 = 25000 rpm
Maximum speed for gear 2 = 40000 rpm
Maximum speed for gear 3 = 6000 rpm
Maximum speed for gear 4 = 100000 rpm
Spindle speed data addresses = F10 to F11 (RO0 to RO15)
The specified spindle speed signal is used.
(For details, refer to the BMI connection manual.)
Spindle control parameter addresses = D10 to D33
Control data addresses = R0 to R3

(1) Create a functional instruction.



(2) Set the spindle speed data
Copy the spindle speed data (RO0 to RO15) to spindle speed data addres specified at the first porameter of SPCNT.

R10	RO0 to 7	Copy from F10
	RO8 to 15	Copy from F11
	0	Clear by 0
	0	Clear by 0

(3) Set the spindle control parameters.
The lower spindle motor speed limit data and the upper spindle motor speed limit data are obtained as follows (see i) and ii) of b) in 4)):

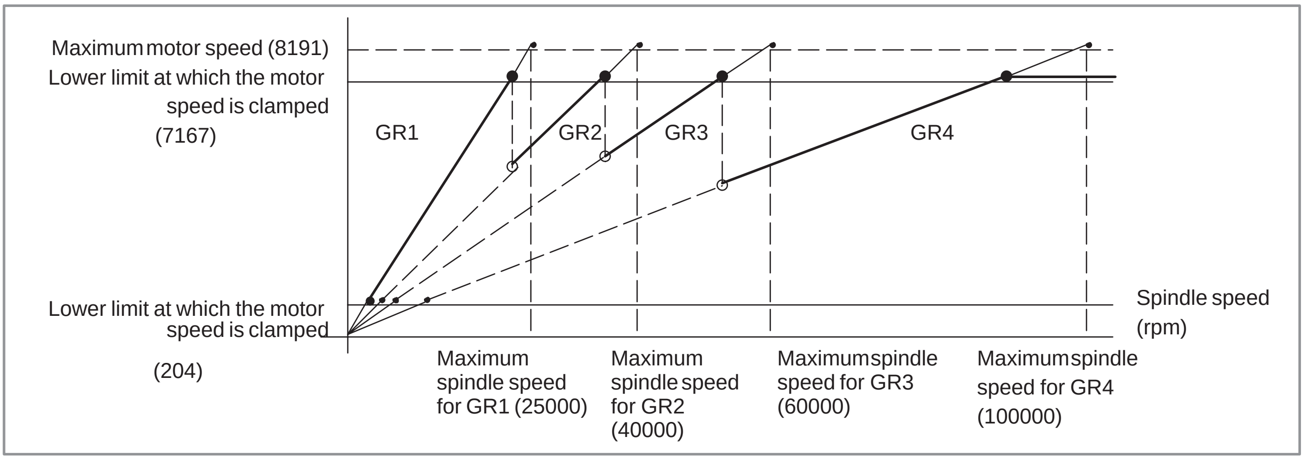
$$\text{Lower spindle motor speed limit data} = \frac{1000}{40000} \times 8191 = 204 \text{ (rpm)}$$

$$\text{Upper spindle motor speed limit data} = \frac{35000}{40000} \times 8191 = 7167 \text{ (rpm)}$$

Then, the spindle control parameters are set as follows:

D10 toD13	204	Lower spindle motor speed limit data
D14 to D17	7167	Upper spindle motor speed limit data
D18 to D21	25000	Maximum spindle speed for gear 1
D22 to D25	40000	Maximum spindle speed for gear 2
D26 to D29	60000	Maximum spindle speed for gear 3
D30 to D33	100000	Maximum spindle speed for gear 4

(4) Calculate the spindle motor rotation command for the spindle speed



From the above graph, the following table can be obtained:

Table 5.61.5 Maximum and Minimum spindle speeds for each gear

	Minimum spindle speed (rpm)	Maximum spindle speed (rpm)
GR1	625	21877
GR2	21878	35004
GR3	35005	52506
GR4	52507	87499

(When CIRC = 0, OVRD = 0)

Thus, if the spindle speed data is 55000 (rpm), when the spindle override is not applied (OVRD = 0) and the direct gear specification is not set (CIRC = 0), the spindle motor rotation command and the spindle gear to be used are obtained as follows:

R0 to R1	4505	Spindlemotor rotation command
R2	8 (GR4)	Spindle gear selection

5.62 END (END OF A LADDER PROGRAM)

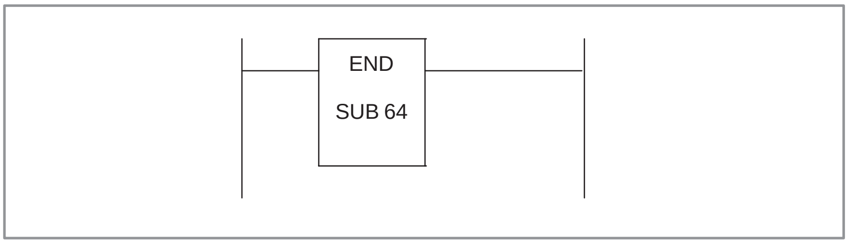
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.62.1 Function

The END functional instruction designates the end of a ladder program. END must be placed at the end of the ladder program.

5.62.2 Format



5.63 CALL (CONDITIONAL SUBPROGRAM CALL)

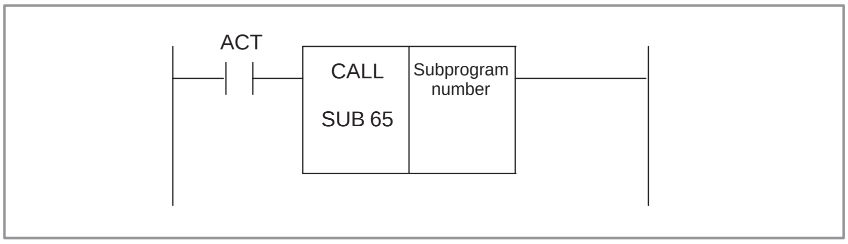
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.63.1 Function

The CALL functional instruction calls a subprogram. When a subprogram number is specified in CALL, a jump occurs to the subprogram if a condition is satisfied.

5.63.2 Format



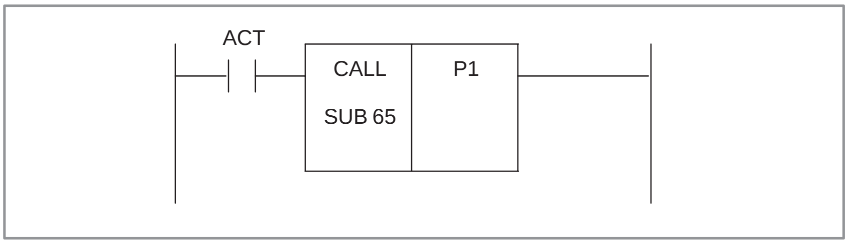
5.63.3 Control Conditions

(a) Input signal
ACT=0 : The CALL instruction is not executed.
ACT=1 : The CALL instruction is executed.

5.63.4 Parameters

(a) Subprogram number
Specifies the subprogram number of a subprogram to be called. The subprogram number must be specified in the P address form. A number from P1 to P512 can be specified.

Example : To call subprogram 1



> **NOTE**
> Be careful when using the CALL instruction with the COM, COME, JMP, or JMPE functional instruction. For details, see Chapter 9 in Part I.

5.64 CALLU (UNCONDITIONAL SUBPROGRAM CALL)

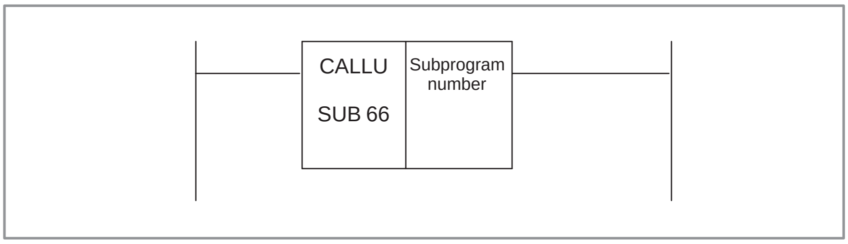
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.64.1 Function

The CALLU functional instruction calls a subprogram. When a subprogram number is specified, a jump occurs to the subprogram.

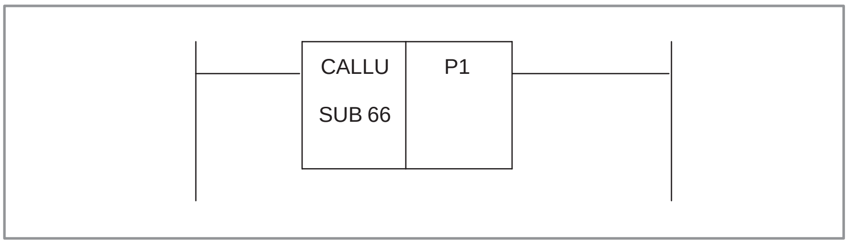
5.64.2 Format



5.64.3 Parameters

(a) Subprogram number
Specifies the subprogram number of a subprogram to be called. The subprogram number must be specified in the P address form. A number from P1 to P512 can be specified.

Example : To call subprogram 1



5.65 SP (SUBPROGRAM)

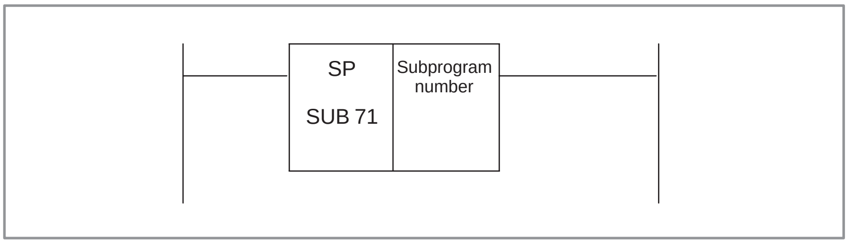
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.65.1 Function

The SP functional instruction is used to create a subprogram. A subprogram number is specified as a subprogram name. SP is used with the SPE functional instruction (mentioned later) to specify the subprogram range.

5.65.2 Format

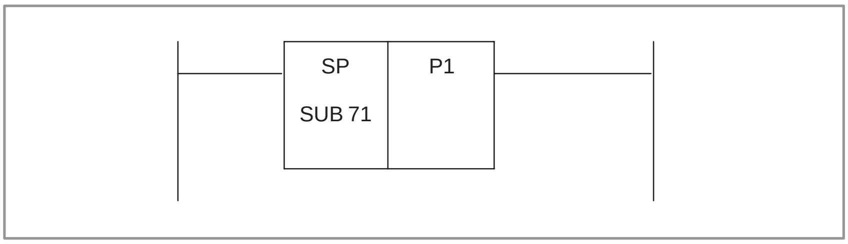


5.65.3 Parameters

(a) Subprogram number
Specifies the subprogram number of a subprogram to be coded following this instruction. The subprogram number must be specified in the P address form. A number from P1 to P512 can be specified. The specified subprogram number must be unique within the sequence program.

Function	PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
Subprogram number	×	P1 to P512	×	×	P1 to P512	P1 to P512	×	×	P1 to P512	P1 to P2000	P1 to P512	P1 to P2000	×	P1 to P512	P1 to P2000	P1 to P512	P1 to P2000	P1 to P2000

Example: When the subprogram number is set to 1



5.66 SPE (END OF A SUBPROGRAM)

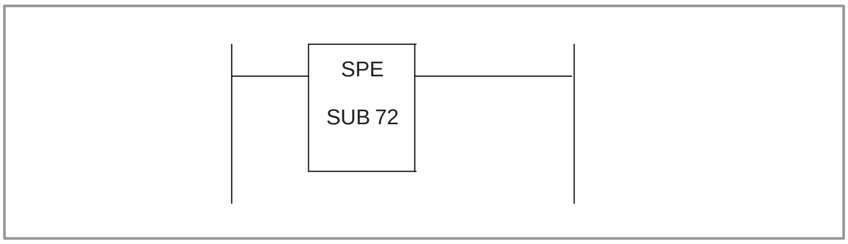
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	×	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.66.1 Function

The SPE functional instruction is used to create a subprogram. SPE is used with the SP functional instruction. It specifies the range of a subprogram. When this functional instruction has been executed, control is returned to the functional instruction that called the subprogram.

5.66.2 Format



5.67 JMPB (LABEL JUMP)

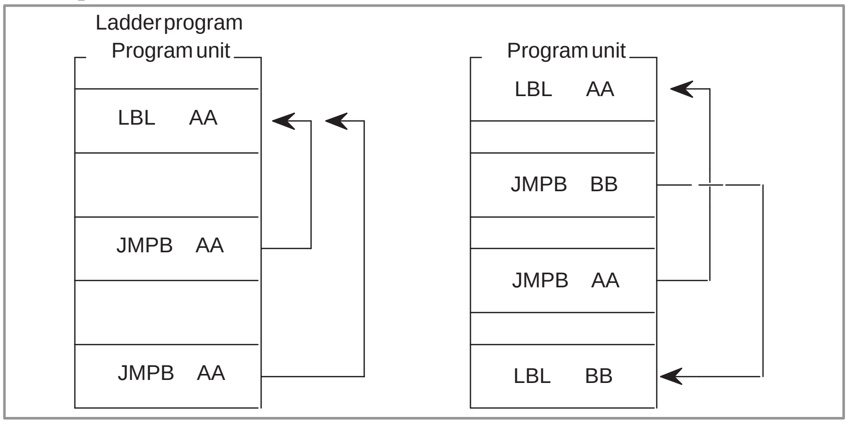
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

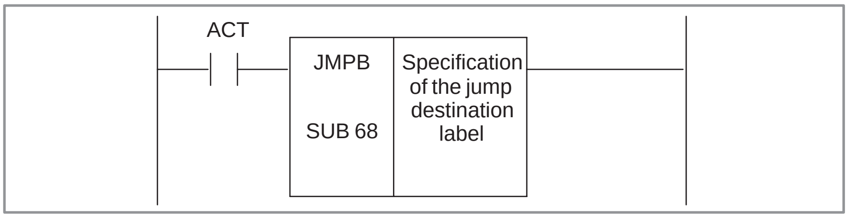
5.67.1 Function

The JMPB functional instruction transfers control to a Ladder immediately after the label set in a Ladder program. The jump instruction can transfer control freely before and after the instruction within the program unit (main program or subprogram) in which the instruction is coded. (See the description of the LBL functional instruction, which is be explained later.)
As compared with the conventional JMP functional instruction, JMPB has the following additional functions:

- More than one jump instruction can be coded for the same label.
- Jump instructions can be nested.



5.67.2 Format



5.67.3 Control Conditions (ACT)

ACT=0 : The next instruction after the JMPB instruction is executed.
ACT=1 : Control is transferred to the Ladder immediately after the specified label.

5.67.4 Parameters

(a) Label specification
Specifies the label of the jump destination. The label number must be specified in the L address form. A value from L1 to L9999 can be specified.

> **NOTE**
> 1 For the specifications of this instruction, see Chapter 10 in Part I.
> 2 When this instruction is used to jump back to a previous instruction, care must be taken not to cause an infinite loop.

5.68 JMPC (LABEL JUMP)

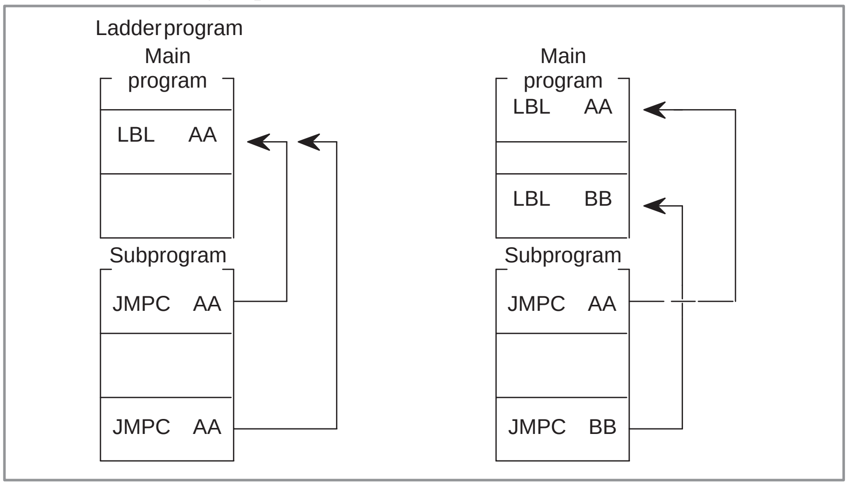
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

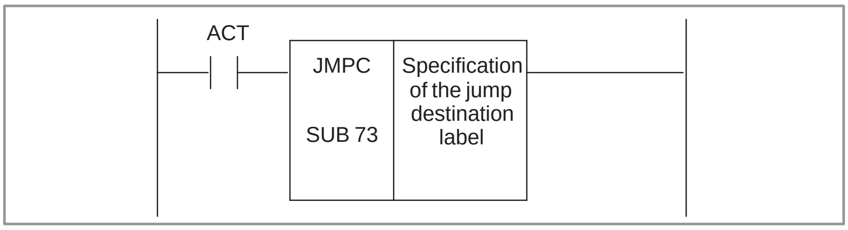
5.68.1 Function

The JMPC functional instruction returns control from a subprogram to the main program. Be sure to code the destination label in the main program. The specifications of this JMPC functional instruction are the same as those of the JMPC functional instruction, except that JMPC always returns control to the main program.

- More than one jump instruction can be coded for the same label.



5.68.2 Format



5.68.3 Control Conditions (ACT)

ACT=0 : The instruction after the JMPC instruction is executed.
ACT=1 : Control is transferred to the Ladder after the specified label.

5.68.4 Parameters

(a) Label specification
Specifies the label of the jump destination. The label number must be specified in the L address form. A number from L1 to L9999 can be specified.

NOTE
1 For the specifications of this instruction, see Chapter 10 in Part I.
2 When this instruction is used to jump back to a previous instruction, care must be taken not to cause an infinite loop.

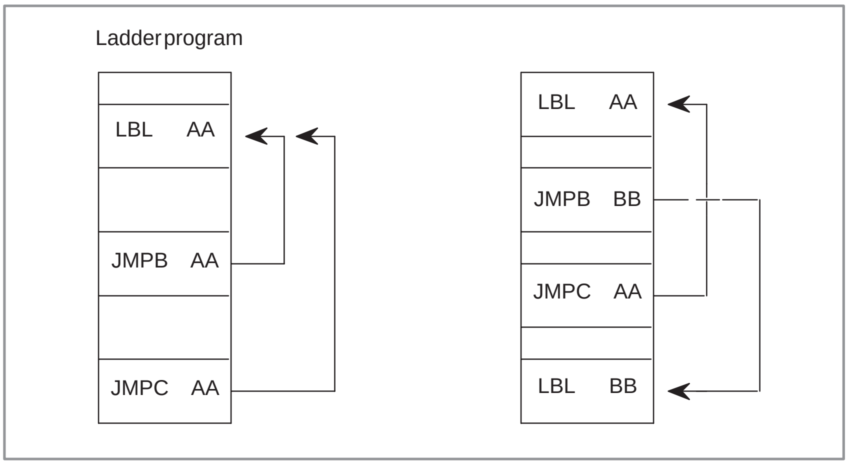
5.69 LBL (LABEL)

○ : Can be used
× : Cannot be used

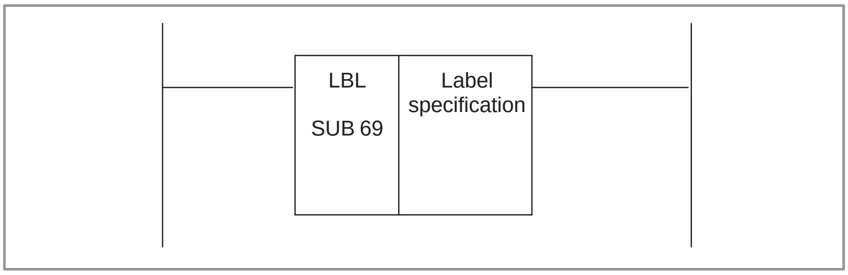
PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

5.69.1 Function

The LBL functional instruction specifies a label in a Ladder program. It specifies the jump destination for the JMPB and JMPC functional instructions. (See the explanation of the JMPB and JMPC functional instructions.)



5.69.2 Format



5.69.3 Parameters

(a) Label specification
Specifies the jump destination for the JMPB and JMPC functional instructions. The label number must be specified in the L address form. A label number from L1 to L9999 can be specified. A label number can be used more than once as long as it is used in a different program unit (main program, subprogram).

NOTE
For the use of this instruction, see Chapter 10 of Part I.

5.70 AXCTL (AXIS CONTROL BY PMC)

○ : Available
× : Unavailable

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	×	×	×

NOTE
1 Option for Axis control by PMC function is required.
2 This functional instruction can not be used on the CNC that does not have option for Axis control by PMC.

5.70.1 Function

This function simplifies the handshake of DI/DO signal for the axis control by PMC.

5.70.2 Format

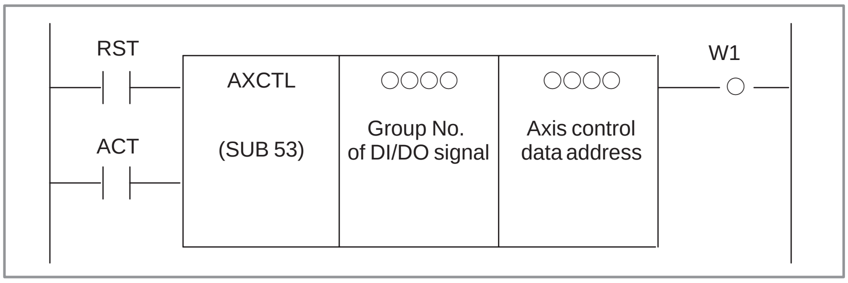


Fig. 5.70.2 AXCTL instruction format

Table 5.70.2 AXCTL instruction coding

Step Number	Instruc-tion	Address Number	Bit Number	Remarks
1	RD	○○○○.	○	RST
2	RD. STK	○○○○.	○	ACT
3	SUB	53		
4	(PRM)	○○○○		Number of DI/DO signal
5	(PRM)	○○○○		Axis control data address
6	WRT	○○○○.	○	W1, processing completion

5.70.3 Control Condition

ACT=0 : The AXCTL function is not executed.
ACT=1 : The AXCTL function is executed.
ACT is to be maintained ‘1’ till the end of AXCTL processing. And reset ACT immediately after the processing is complete (W1 = 1).
RST=0 : Release reset.
RST=1 : Set the reset signal (ECLRx) to 1. All the buffered commands are invalidated and the command being executed is stopped. Set RST at the same time as the reset of CNC when CNC becomes the state of alarm.

NOTE
When RST and ACT become 1 at the same time, RST is prior to ACT.

5.70.4 Parameters

(a) Group number of DI/DO signal

Specify the DI/DO signal group by the number.

1 : group A(G142 to G149, F130 to F132)
2 : group B(G154 to G161, F133 to F135)
3 : group C(G166 to G173, F136 to F138)
; Cannot be used on Power Mate–D/F
4 : group D(G178 to G185, F139 to F141)
; Cannot be used on Power Mate–D/F
5 : group E (G226 to G233, F228 to F230)
; Can be used only on Power Mate–H
6 : group F (G238 to G245, F231 to F233)
; Can be used only on Power Mate–H

Add 1000 to the above number as follows if you use HEAD2 of FS16/18–TT or two path of Power Matw–D.

1001 : group A (G1142 to G1149, F1130 to F1132)
1002 : group B (G1154 to G1161, F1133 to F1135)
1003 : group C (G1166 to G1173, F1136 to F1138)
; Cannot be used on Power Mate
1004 : group D (G1178 to G1185, F1139 to F1141)
; Cannot be used on Power Mate

When 3–path control is used with the Series 16*i*/18*i*, the following addresses are used for DI/DO signals:

2001 : Group A (G2142 to G2149, F2130 to F2132)
2002 : Group B (G2154 to G2161, F2133 to F2135)
2003 : Group C (G2166 to G2173, F2136 to F2138)
2004 : Group D (G2178 to G2185, F2139 to F2141)

(b) Axis control data address

Select the addresses of the locations that contain PMC axis control data.

+0	FANUC reserved	Specify 0.
1	Control command	Specify the command to set EC0x–EC6x.
2	Command data 1	Specify the data to set EIF0x–EIF15x.
3		
4	Command data 2	Specify the data to set EID0x–EID31x.
5		
6		
7		(x=A / B / C / D)

The following functions are available.

Operation	Control	Command data 1	Command data 2
Rapid traverse	00H	Feedrate Need not to set if CNC PRM. 8002#0 = 0.	Total travel amount
Cutting feed (feed per min.)	01H	Feedrate (Note 1)	Total travel amount
Cutting feed (Note 2) (feed per revolution)	02H	Feedrate per revolution	not used
Skip (feed per min.) (Note 2)	03H	Feedrate	Total travel amount
Dwell	04H	not used	Dwell time
Reference pos. return	05H	not used	not used
Continuous feed (Note 3)	06H	Feedrate	Feed direction (Note 4)
1st ref. pos. return 2nd ref. pos. return 3rd ref. pos. return 4th ref. pos. return (Note 2)	07H 08H 09H 0AH	Feed rate Need not to set if CNC PRM. 8002#0 = 0.	not used
External pulse synchronization (Note 2) (Note 3)	0BH 0DH 0EH 0FH	Pulse weighting (Only M series)	not used
Speed command (Note 2) (Note 5) (Note 6) (Note 2)	10H	Feedrate	not used
Machine coordinate positioning. (Rapid traverse) (Note 2) (Note 6)	20H	Feedrate Need not to set if CNC PRM. 8002#0 = 0.	Position of machine coordinate. (absolute)

NOTE

1 When you specify 0 for feedrate, CNC does not work. Please release this state by RST = 1.
2 It is not available in PMC–MODEL PA1/PA3.
3 When you end a continuous feed or external pulse synchronization, set RST to 1. And, continuous feed can't be used with buffering inhibits signal = 1. You must set the signal to 0.
4 Specify the direction by most significant bit of command data 2.
5 Command control axis must be specified to rotary axis by setting parameter ROTx (No. 1006#0) to 0.
6 Not applicable to the Power Mate.
7 For details such as the range of command data, please refer to the connecting manual for each CNC models.

CAUTION

The above table is not up–to–date. For the latest information, refer to the descriptions about PMC axis control in the “CNC Connection Manual (Functions).”

Example 1) In case of cutting feed (feed per min.)

+0	0H	Not used (Specify 0).
1	01H	Command code for cutting feed. (feed per min.)
2	Feedrate	unit : mm/min.
3		
4	Total travel amount	
5		unit : 0.001mm
6		(x=A / B / C / D)
7		

Example 2) In case of machine coordinate positioning.

+0	0H	Not used (Specify 0).
1	20H	Command code for machine coordinate positioning.
2	0 or Feedrate	In case of CNC PRM8002#0= 0 not used. = 1 Feedrate.
3		
4	Position in machine coordinate system	(Absolute)
5		
6		
7		

NOTE

It is necessary to set the CNC parameters relating to the axis movement.

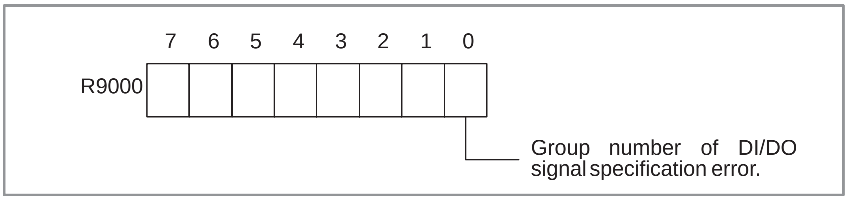
5.70.5 End of Command (W1)

W1=0 : It is 0 usually. W1=1 indicates that AXCTL instruction is completed.
Specify ACT=0 immediately after processing is completed. (W1=1).

W1=1 : It will become 1 when the command of the axis control by PMC is buffered on CNC (when EMBUFx=0) or when axis movement is completed (when EMBUFx=1).

5.70.6 Operation Output Register (R9000)

When error occurs by processing the axis control by PMC, the bit of the operation output register will be set. At the same time, processing is over.



NOTE
1 W1 becomes 1 regardless of the state of ACT.
2 It is not related to the state of the alarm signal (EIALx).

5.70.7 Remarks

(1) The following signals cannot be operated from this function. Please operate by LADDER.

- Axis control stop signal
  ESTPx (G142#5, G154#5, G166#5, G178#5)
- Servo–off signal
  ESOFx (G142#4, G154#4, G166#4, G178#4)
- Block stop signal
  ESBKx (G142#3, G154#3, G166#3, G178#3)
- Block stop inhibit signal
  EMSBKx (G143#7, G155#7, G167#7, G179#7)
- Controlled axis selection signal
  EAX1–EAX8 (G136#0 to #7)
- Override signal *FV0E–*FV7E (G151#0 to #7)
- Override cancel signal OVCE (G150#5)
- Rapid traverse override signal
  ROV2E, ROV1E (G150#1, #0)
- Dry run signal DRNE (G150#7)
- Manual rapid traverse selection signal RTE (G150#7)
- Skip signal SKIP/ESKIP (X4#7, #6)
- Buffering inhibit signal
  EMBUFx (G142#2, G154#2, G166/#2, G178#2)
  (x=A/B/C/D)

WARNING
Movement cannot be sured when controlled axis selection signal (EAXx) is changed in the state of ACT=1.

(2) Buffering inhibit signal (EMBUFx)
0 : The commands are buffered on the CNC.
Even if one command is being executed, the CNC accepts the next command as long as there is vacancy in the buffer on CNC.
W1 will become 1 when the command of the axis control by PMC is buffered on CNC.
1 : Prohibits the buffering on CNC.
W1 will become 1 when the movement of the instructed axis control by PMC is completed.

5.71 PSGNL (POSITION SIGNAL OUTPUT)

○ : Can be used
Δ : Usable in some CNC models
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3 SB4	SB5	SB6	SC	SC3 SC4	NB	NB2	NB6
○	○	×	×	×	×	×	×	×	Δ	Δ	×	×	×	×	×

NOTE
The PMC–SB5/SB6 can be used only in the Power Mate *i*–D/H.

5.71.1 Function

This function outputs a signal that indicates the are in which the current position in the mechanical coordinate system is located. The area is specified by parameter.

5.71.2 Format

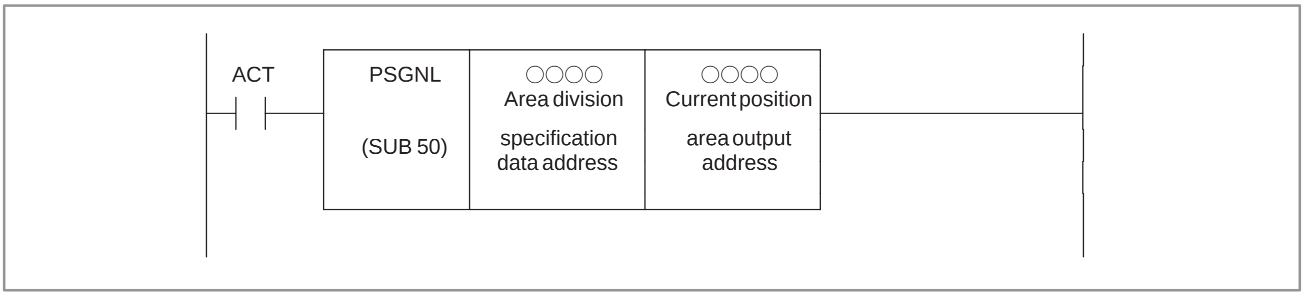


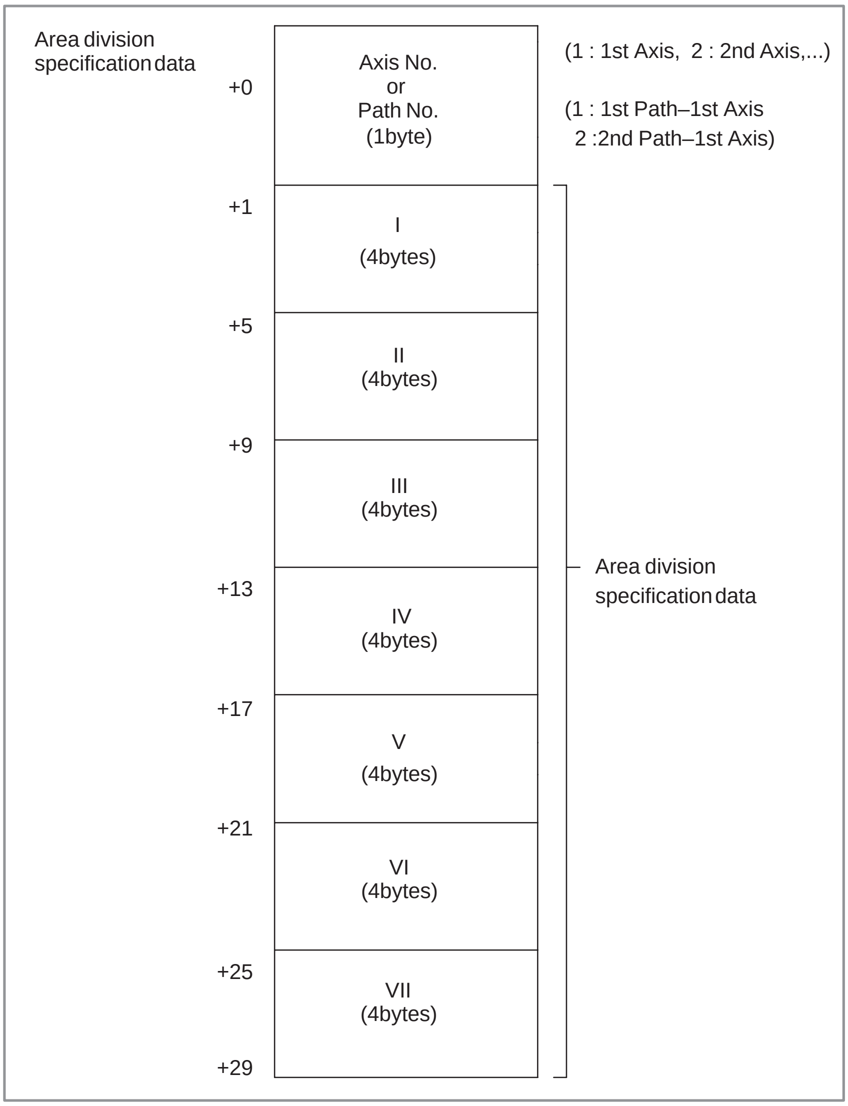
Fig. 5.71.2 PSGNL instruction format

5.71.3 Control Condition

(a) Execution specification (ACT)
ACT=0 : The PSGNL instruction is not executed.
ACT=1 : The PSGNLnstruction is executed.

5.71.4 Parameters

(a) Area division specification data address
Set the top address of area division specification data 29 bytes of continuous memory are necessary in nonvolatile memory area for area division specification data.



- In case of axis–No. specification
  Please set axis–No. to select. (1 byte data of binary format)

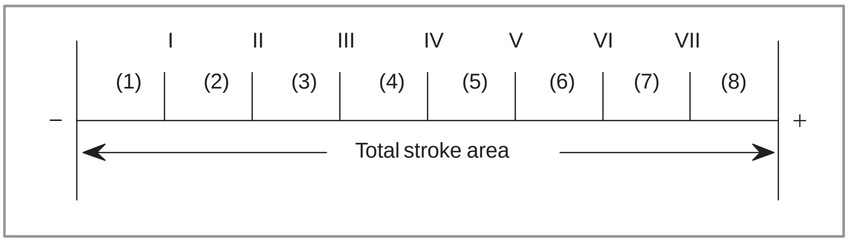
(Example) Axis No.=1 : For machine coordinates of the 1st axis
Axis No.=2 : For machine coordinates of the 2nd axis

- In case of path specification (Power Mate–MODEL D dual path control)
  Please set path–No. of axis to select. (1 byte data of binary format)

(Example)
Path spec.=1: For machine coordinates of the 1st axis on the 1st path)
Path spec.=2: For machine coordinates of the 1st axis on the 2nd path)

Each area division specification data (I, II, III, ..., VII) is 4bytes binary format data. (Scale is 0.001mm or 0.001inch)

<Example of area division>



As shown in the above diagram, check can be performed for the 8 areas (1) to (8) by dividing the total stroke area by 7 division points.

> **NOTE**
> 1 Please set the division points data in ascending order (I < II < .....<VI < VII). If they are not in ascending order, the sequence program cannot operate normally.
> 2 Even if you need division points only under 7, you must set the division specification data for7.

(b) Current position area output address
The address which is output the divided area that the currrent position in the machine coordinates system located.

	7	6	5	4	3	2	1	0
Cureent position area output address	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

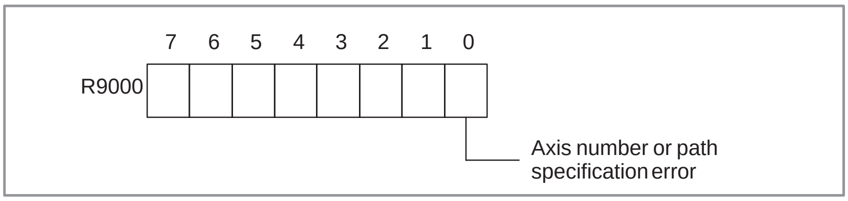
Corresponding bit is set to 1 indicates the area in which the current position in the machine coordinates system is located.

(Example)

Bit 0 becomes 1 if the current position in the machine coordinate system is greater than VII. Bit 1 becomes 1 if the current position in the machine coordinate system is greater than VI but not greater than VII.

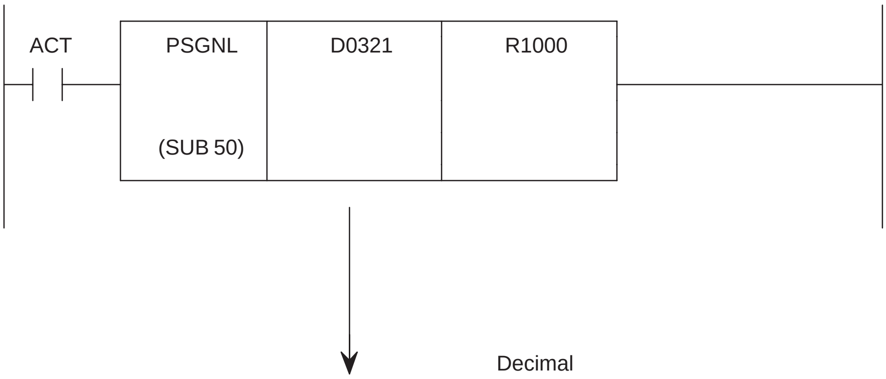
5.71.5 Operation Output Register (R9000)

If an error occurs in position signal output processing, the corresponding bit of the operation output register is set.



5.71.6 Example of Using Position Signals

This example illustrates how to output the position signal of the current position of the first axis of path 1 in the machine coordinate system (the area split specification data and current position area output addresses are set to D0321 and R1000, respectively).



	Decimal
D0321	1
D0322	–300000
D0326	–200000
D0330	–100000
D0334	0
D0338	100000
D0342	200000
D0346	300000

If ACT = 1 for the above ladder and area split specification data, the current specification area output (R1000) is as follows:

- R1000.0=1: The current position in the machine coordinate system is greater than 300.000 mm.
- R1000.1=1: The current position in the machine coordinate system is greater than 200.000 mm but not greater than 300.000 mm.
- R1000.2=1: The current position in the machine coordinate system is greater than 100.000 mm but not greater than 200.000 mm.
- R1000.3=1: The current position in the machine coordinate system is greater than 0 mm but not greater than 100.000 mm.
- R1000.4=1: The current position in the machine coordinate system is greater than –100.000 mm but not greater than 0 mm.
- R1000.5=1: The current position in the machine coordinate system is greater than –200.00 mm but not greater than –100.000 mm.
- R1000.6=1: The current position in the machine coordinate system is greater than –300.000 mm but not greater than –200.000 mm.
- R1000.7=1: The current position in the machine coordinate system is not greater than –300.000 mm.

5.72 PSGN2 (POSITION SIGNAL OUTPUT 2)

○ : Can be used
Δ : Usable in some CNC models
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3 SB4	SB5	SB6	SC	SC3 SC4	NB	NB2	NB6
○	○	×	×	×	×	×	×	×	Δ	Δ	×	×	×	×	×

NOTE
The PMC–SB5/SB6 can be used only in the Power Mate *i*–D/H.

5.72.1 Function

Turn W1=1 which th ecurrent position in the machine coordinates system is in the area specified by parameters.

5.72.2 Format

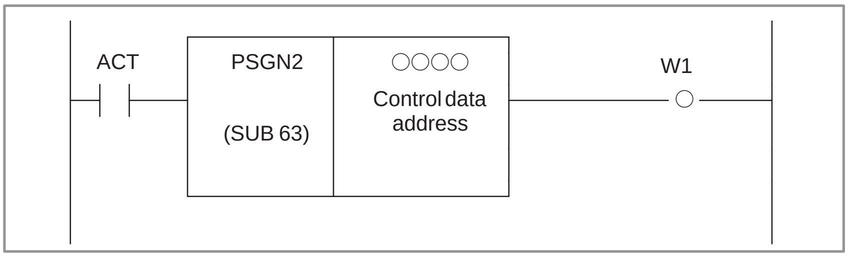


Fig. 5.72.2 PSGN2 instruction format

5.72.3 Control Condition

(a) Execution specification (ACT)
ACT=0 : The PSGN2 instruction is not executed.
ACT=1 : The PSGN2 instruction is executed.

5.72.4 Parameters

(a) Control data address

Please set the top address of control data.

For the area specification data, 9bytes of continuous memory area in the nonvolatile memory is necessary.

Address	Content	Remarks
Control data+0 (basic specification)	Axis No. or Path No. (1byte)	(1 : 1st Axis, 2 : 2nd Axis,...) (1 : 1st Path–1st Axis 2 :2nd Path–1st Axis)
+1	Boundary Point 1 (4bytes)	
+5	Boundary Point 2 (4bytes)	
+9		

- In case of axis–No. specification
  Please set axis–No. to select. (1 byte data of binary format)

(Example) Axis No.=1 : For machine coordinates of the 1st axis
Axis No.=2 : For machine coordinates of the 2nd axis

- In case of path specification (Power Mate–MODEL D dual path control)
  Please set path–No. of axis to select. (1 byte data of binary format)

  In the PMC–SB5/SB6 for the Power Mate *i*, control data can be used under the following expanded specification. Generally, use of the expanded specification offers faster operation. So it is recommended that the expanded specification be used.
  The control data requires 12 consecutive bytes in nonvolatile memory.

Address	Content	Remarks
Control data + 0 (expanded specification)	Fixed to 0 (2bytes)	
+2	Path specification (1byte)	(0 = path 1, 2 = path 2, ...)
+3	Axis number (1byte)	(1 = first axis, 2 = second axis, ...)
+4	Boundary point 1 (4bytes)	
+8	Boundary point 2 (4bytes)	
+12		

- Path specification
  A path is specified. (One–byte data in binary form)

(Example) Path specification = 0: Path 1 is specified.
Path specification = 2: Path 2 is specified.
Path specification = 3: Path 3 is specified.

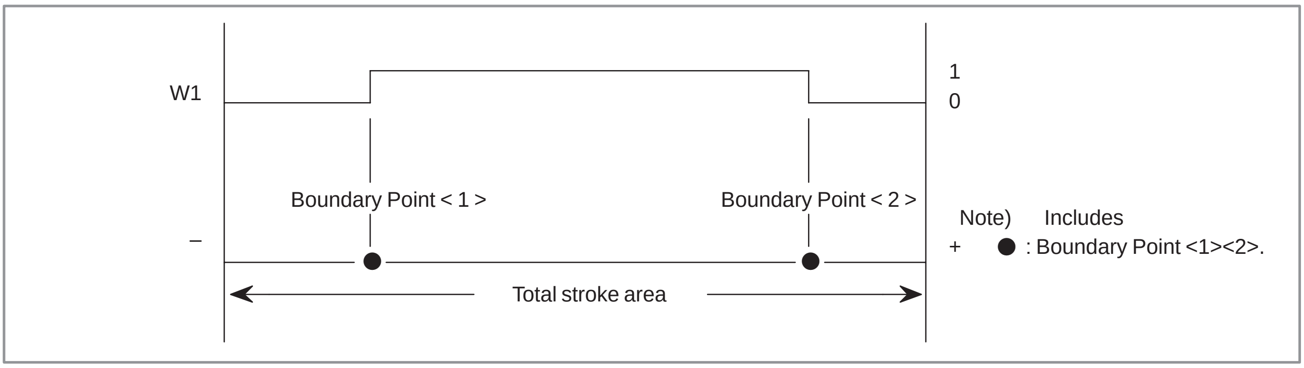
- Axis number specification
  An axis number is specified. (Binary one–byte data)

(Example) Axis number = 1: The machine coordinates for the first axis are specified.
Axis number = 2: The machine coordinates for the second axis are specified.

(Boundary points <1> and <2> are binary four–byte data. Their unit of measurement is 0.001 mm or 0.001 inch.

> **NOTE**
> Data for both boundary points <1> and <2> must be specified in ascending order (boundary point 1 ≦ boundary point 2).

<Example of area division>

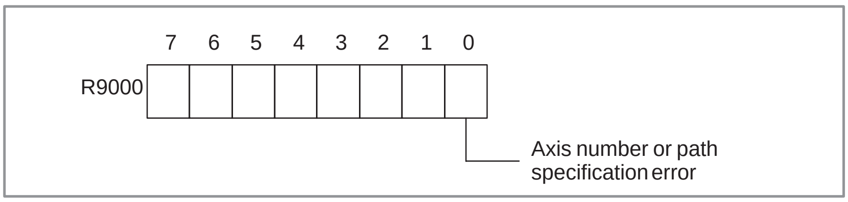


5.72.5 Current Position Area Output (W1)

W1=0 : The current position in the machine coordinates system is outside of the area specified by parameters.
W1=1 : The current position n the machine coordinates system is inside of the area specified by parameters.

5.72.6 Operation Output Register (R9000)

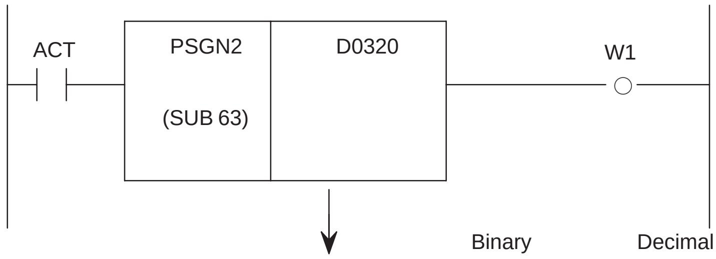
If an error occurs in position signal output processing, the corresponding bit of the operation output register is set.
In this case, W1 = 0.



5.72.7 Example of Using Position Signals

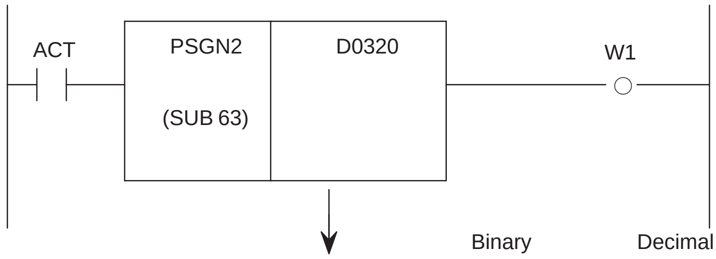
- This example illustrates how to output the position signal for the current position of the second axis of path 1 in the machine coordinate system if it is the position between –100.000 mm and 200.000 mm. (The control data address is set to D0320.)

(Basic specification)



	Binary	Decimal
D0320	00000010	(2)
D0321	01100000	(–100000)
D0322	01111001	
D0323	11111110	
D0324	11111111	
D0325	01000000	(200000)
D0326	00001101	
D0327	00000011	
D0328	00000000	

(Basic specification)



	Binary	Decimal
D0320	00000000	0
D0321	00000000	
D0322	00000000	0
D0323	00000010	2
D0324	01100000	–100000
D0325	01111001	
D0326	11111110	
D0327	11111111	
D0328	01000000	200000
D0329	00001101	
D0330	00000011	
D0331	00000000	

If ACT = 1 for the above ladder and control data, W1 = 1 when: –100.000 ≦ current position (second axis) in the machine coordinate system ≦ 200.000 mm

6 NONVOLATILE MEMORY

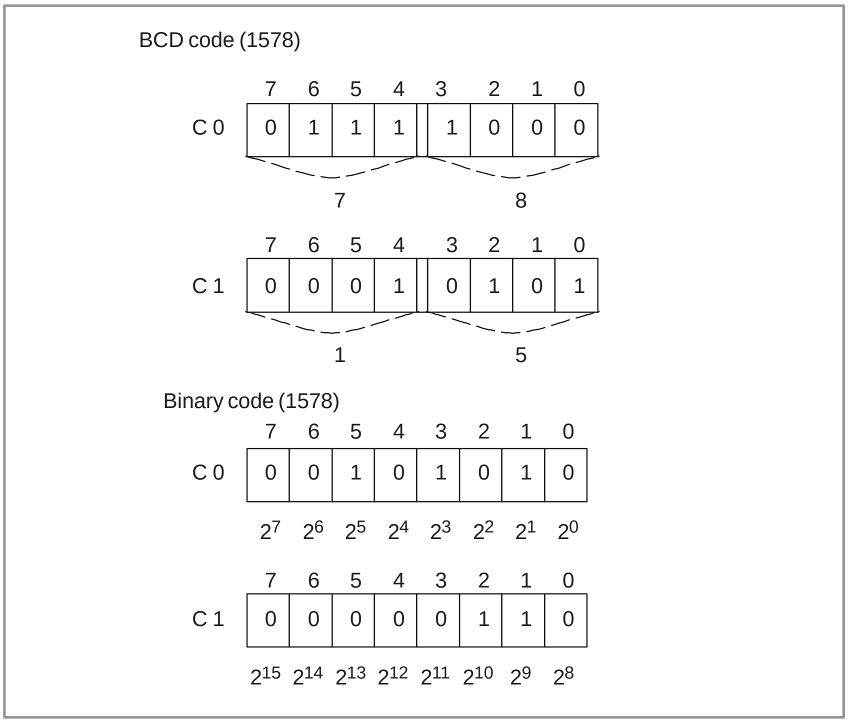
6.1 TIMER, COUNTER, KEEP RELAY, NONVOLATILE MEMORY CONTROL, DATA TABLE

Nonvolatile memory is considered nonvolatile if its contents are not erased when the power is turned off.

(1) Used for the timer
Time can be set and displayed from the CRT/MDI panel. The set time can be read or written by a sequence program instruction.

(2) Used for the counter
This area is used to store the preset and cumulative values of the counter. Values can be set and displayed from the CRT/MDI panel. These values can be read and written by a sequence program instruction. Refer to section 3.5 for details of addresses.
The data format is two bytes of BCD or binary, and the higher-order digits are entered at the smaller address.
Whether counter is processed by BCD format or binary format is selected by a system parameter.

Example) PMC counter addresses are C0 and C1 and the set value is 1578.



To change low-order digits of the set value by a sequence program instruction with 1 byte processing, specify C0 as the output address of the functional instruction parameters to enter new data.

(3) Keep relay
This memory is used as parameters, keep relays, etc. for sequence control. Setting and display are possible from the CRT/MDI panel and sequence program instructions can be used for reading and writing. Since data set or displayed from the CRT/MDI panel is binary eight bits, each of the eight digits of data is set or displayed as 0 or 1.

(4) Nonvolatile memory control (MWRTF, MWRTF2) (Address K16)
This memory is used when the position of a moving part of the machine tool, such as a lathe turret, is stored in code (BCD, etc.) and to maintain it while power is off.

	#7	#6	#5	#4	#3	#2	#1	#0
K16	MWRTF2	MWRTF						

Setting and display are possible from the CRT/MDI panel, and sequence program instructions can be used for reading and writing. If, for example, power is turned off for some reason during rotation of the turret, the turret stops and a difference between the contents of the memory storing the position and the actual position of the turret occurs. When power is turned on again, the machine tool will be out of sequence. To prevent this, use the nonvolatile memory control, and a sequence program as follows.

(a) Set MWRTF in nonvolatile memory control to 1 before starting the turret.

(b) Start the turret.

(c) Set MWRTF to 0 after the turret stops.

(d) MWRTF remains 1 if power is turned off between a) and c).

(e) When power is turned on again, automatically MWRTF2 = 1 and an error is reported to the sequence program. Thus, the sequence program processes (a) to (d), check for the error of MWRTF2, and outputs an alarm when MWRTF2 = 1 (error).

(f) In response, the operator should set MWRTF and MWRTF2 to 0 from the CRT/MDI panel.

(g) Resume operation after the contents of the memory and the turret position are aligned.

(5) Data table
A sizable amount of numeric data (data table) can be used for sequence control by the PMC. See section 6.3 for details.

6.2 READING AND WRITING OF NONVOLATILE MEMORY DATA

All the nonvolatile memory data can be read and written by the sequence program. The memory read and written by the PMC sequence program is actually not a nonvolatile memory, but a nonvolatile memory image (RAM) storing the same data as the nonvolatile memory. When the power supply is turned off, the data in the nonvolatile memory image disappears. Immediately after the power is turned on, the nonvolatile memory data is automatically transferred to the nonvolatile memory image. Before the power is turned off, the data is correctly restored.

When the nonvolatile memory image is rewritten by the sequence program, the data is automatically transferred to the CMOS or bubble memory.

When the sequence program rewrite nonvolatile memory image of area, the rewritten data is automatically transferred to the nonvolatile memory.

Rewriting of nonvolatile memory can also be done by rewriting optional addresses of the nonvolatile memory image in an optional timing. The changed data will be automatically transferred to the nonvolatile memory.

Therefore, there is not special processing necessary when the sequence program writes or reads nonvolatile memory. It will only take some time to write in the nonvolatile memory (512 ms).

6.3 PMC DATA TABLE

(1) Introduction

PMC sequence control sometimes requires a sizable amount of numeric data (herein after referred to as data table. If contents of such data table are free to set or to read, they can be used as various PMC sequence control data, such as tool numbers of tools on the ATC magazine.

Each table size can be set optionally in the memory for data table, and 1-, 2-, or 4-byte binary or BCD format data can be used per each table, thus consigning a simple-to-use table.

Data in the data table can be set in the nonvolatile memory or displayed via the CRT/MDI panel.

Data set in the data table can also be easily read or written by the sequence program using function instructions as data search (DSCHB), or index modification data transfer (XMOVB).

NOTE
For details of the usable range, see the description of PMC sequence program addresses in Part I.

(2) Configuration of the PMC data table and notes on programming

(a) Configuration of the data table
PMC data table consists of table control data and data table. Table control data control the size and data format (BCD or binary) of the tables.
This table control data must first be set from CRT/MDI before preparing data table. In the sequence program, the table control data cannot be read or written. When the contents of the nonvolatile memory are read or written using the Floppy Cassette, the table control data is read or written together. Figure 6.3 (a) is a general configuration of the data table, and Figure 6.3 (b) is a detailed configuration of the data table. Also refer to 3.7 for data table configuration.

(b) Data table head address
If the data table starts from an odd address, for example, when a data table is created with an odd number of one-byte data, the DSCHB instruction operates slower than when the data table starts from an even address. It is recommended that the starting address of a data table be an even number.

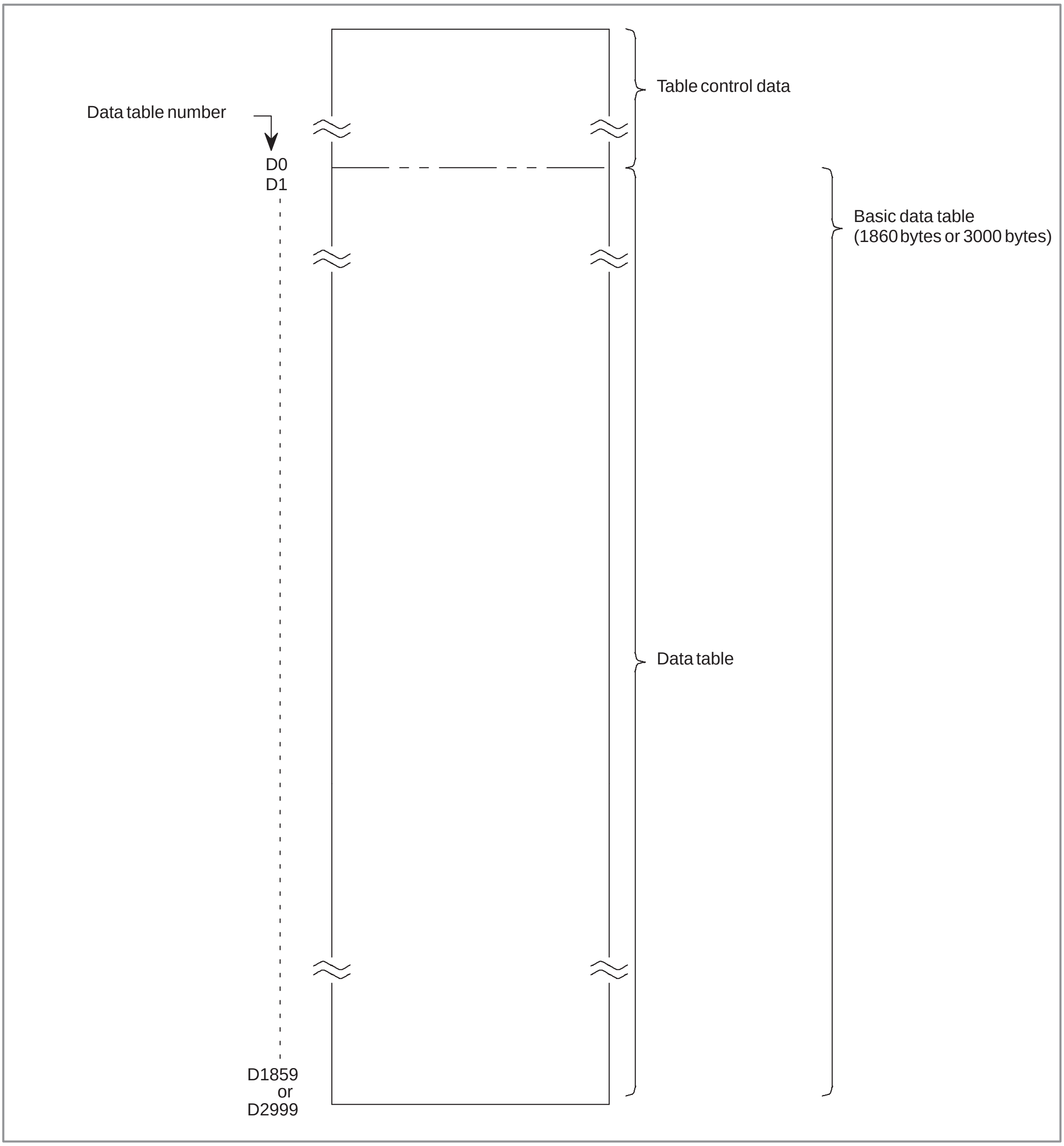


Fig. 6.3 (a) General configuration of data table

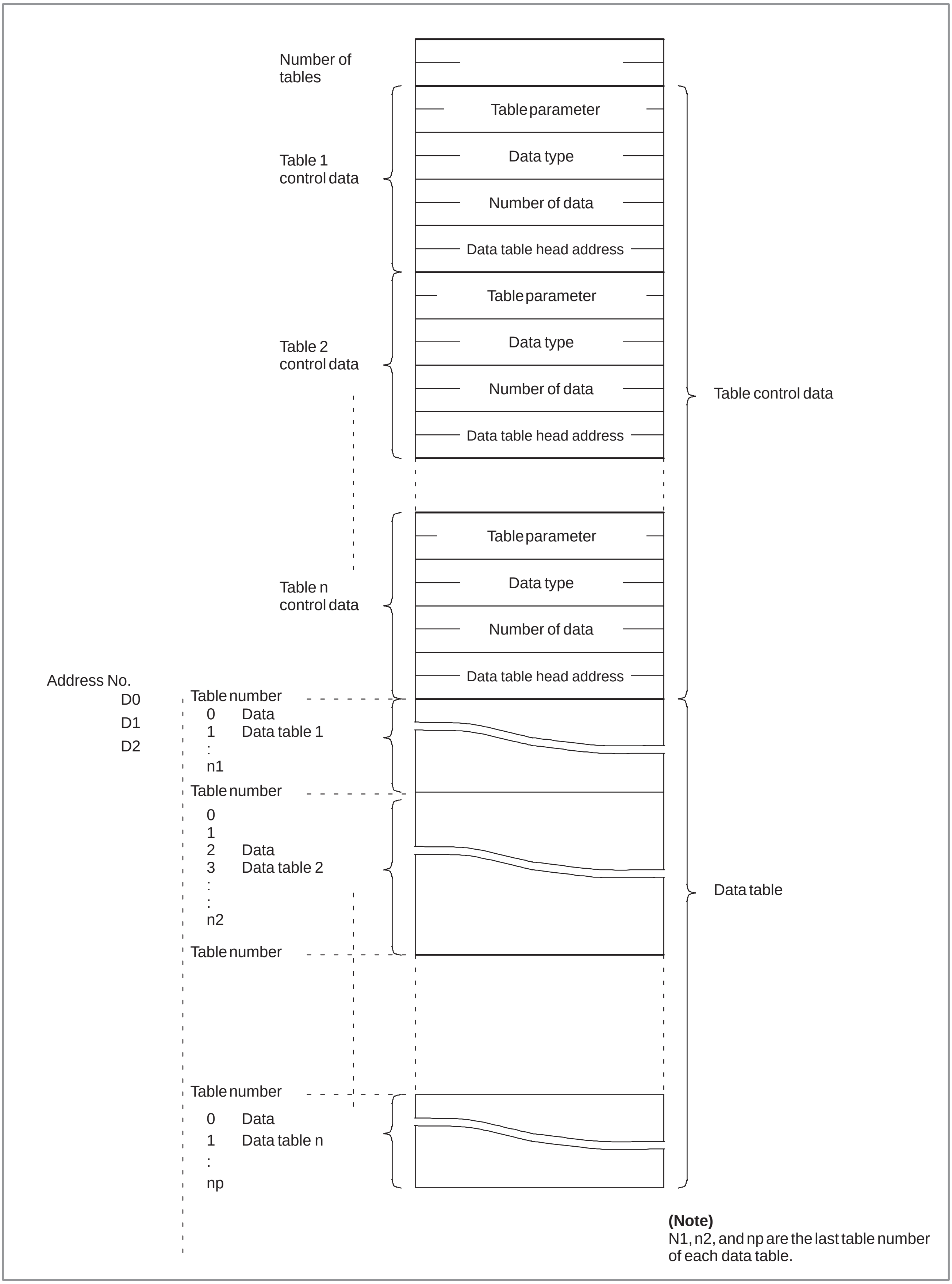


Fig. 6.3 (b) Detailed configuration of data table

(3) Table control data
The table control data controls a data table
If the table control data is not properly set, a data table described in Item (4) cannot be properly created.
Referring to the description in Item (3), set the table control data, then create a data table.

(a) Number of groups of tables
Specify the number of groups of data tables in binary.

(b) Control data for table groups 1 to n
Each data table has table control data consisting of the starting address of the table, table parameters, data type, and the number of data items.

(i) Starting address of the table
Specify the starting address of the table from D0 to D1859 or D0 to D2999.

(ii) Table parameter

#7	#6	#5	#4	#3	#2	#1	#0
						MASK	COD

COD { 0 : A data table is specified in binary.
1 : A data table is specified in BCD.

MASK { 0 : The contents of the data table are not protected.
1 : The contents of the data table are protected.

(iii) Data type
Specify the length of data in the data table.

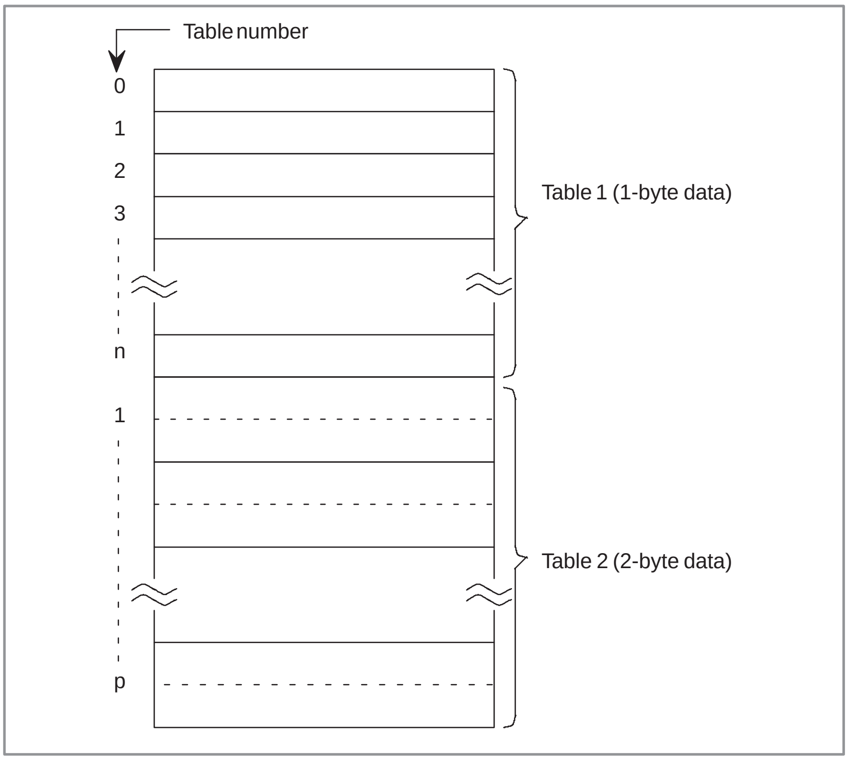
{ 0 : One byte
1 : Two bytes
2 : Four bytes

(iv) Number of data items
Specify the number of data items used in the data table.

(4) Data table
Data table can be created within the range of the memory (D address) for the data table and separated some groups. This number of groups is decided with the number of tables of table control data.

The maximum of the number of table groups.

Except series 15b PMC–NB	max 100 tables
PMC–NB	max 50 tables



Each data table can be used in 1, 2 or 4 byte data. Table parameter of table control data decides whether to use 1 or 2 byte data. Therefore, 1 table number is taken for a 1-byte data when table data is 1 byte; 2 byte data when table data is 2 bytes.

(5) Entering data in a data table
Specify a location number in the data table from the CRT/MDI panel, then enter the data. A number for each location in the table is defined for each data table group.

NOTE
Reading and Writing of the data table are available from the sequence program.

7 LADDER DIAGRAM FORMAT

A designer examines and checks the ladder diagram in the process of design. However, it should be noted that other persons (maintenance servicemen, for example) read the ladder diagram far longer than the designer.
Accordingly, the ladder diagram must be written to be easily understood by all persons.
For this purpose, applicable symbols, writing method, and other methods are specified as detailed below.

7.1 ADDRESSES, SIGNAL NAMES, COMMENTS, AND LINE NUMBERS

Addresses, signal names, comments, and line numbers must be inserted into a ladder diagram to enable all users to easily read the ladder diagram.

7.1.1 Addresses

Each address consists of an address number and a bit number, and it is represented as follows.

```
R 202. 2
   |    |
   |    |_________ Bit number (0 to 7)
   |
   |______________ Address number
                   (A numeric of 4 digits or less after an alphabetic character)
```

An alphabetic character is prefixed to the start of each address number to represent the kinds of signals as shown in Table 7.1.1.

Table 7.1.1 Alphabetic symbols of address numbers

Symbol	Type of signal
X	Input signal entered from machine tool to PMC (MT→PMC)
Y	Output signal sent from PMC to machine tool (PMC→MT)
F	Input signal entered from CNC to PMC (CNC→PMC)
G	Output signal sent from PMC to CNC (PMC→CNC)
R	Internal relay
A	Message display request
C	Counter
K	Keep relay
D	Data table
T	Variable Timer
L	Label number
P	subprogram number

7.1.2 Signal Names

Suitable symbols shall be attached to I/O signals as signal names according to the following procedure.

(1) The names of all signals containing CNC signals and machine tool signals are represented within 6 characters.
Alphanumeric characters and special symbols described in this manual are all employable.

(2) For CNC↔PMC signal names, signal names shown in the PMC address table are employable as they are.

(3) CNC signals to be entered from the machine tool and CNC signals to be sent to the machine tool are identified from each other by prefixing X or Y to the start of these CNC↔PMC signal names, respectively. A single block input signal is represented as XSRK by prefixing X, while a start lamp output signal is represented as YSTL by prefixing Y, for example. However, when X or Y is prefixed to the start of an CNC↔PMC signal name, certain signal names exceed 6 characters. In such a case, omit the last character from such a signal name (*SECLP↔X*SECL)

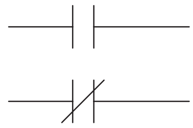
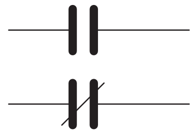
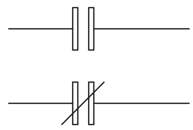
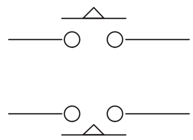
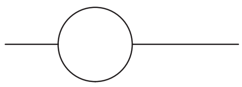
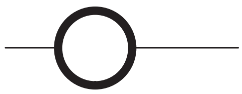
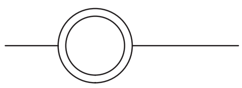
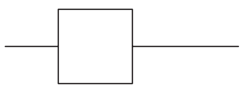
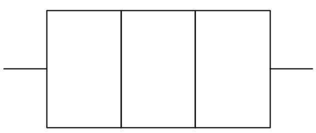
7.1.3 Comments

A comment of within 30 characters can be inserted to a relay coil in a sequence program and each symbol in a symbol table.
When relay coils are output signals to the machine tool, insert a detailed description of the signal to all relay coils as the comment to be inserted to the relay coil. Also insert a detailed description of the signal when other auxiliary relays are significant in sequence control.
Be sure to insert detailed descriptions of machine tool–related input signals, in particular, as comments in symbol tables.
Detailed comments are necessary as it is difficult to guess the meanings of signals specific to the machine tool by judging the symbol name alone.

7.1.4 Line Numbers

A line number should be attached to each line of the ladder diagram. For details, refer to Sec. 7.3.

7.2 SYMBOLS USED IN THE LADDER DIAGRAM

Symbol	Description
 A contact B contact	These are the contacts of relays in the PMC, and are used for other input from the machine side and CNC
 A contact B contact	These are input signals from the CNC.
 A contact B contact	These are input signals from the machine side (including the built-in manual control panel).
 A contact B contact	These are timer contacts in the PMC
	This is a relay coil whose contact is used only in the PMC.
	This is a relay coil whose contact is output to CNC.
	This is a relay coil whose contact is output to the machine side.
	This is the coil of a timer in the PMC.
	This is a PMC fucnctional instruction. The actual form varies depending on the instruction.

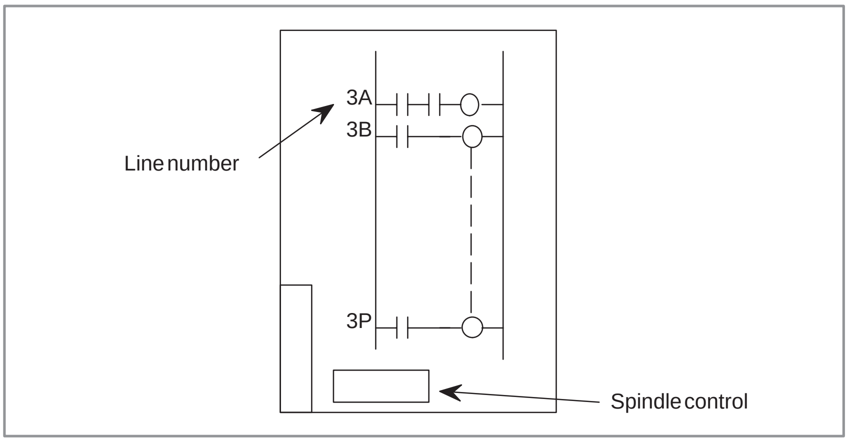
NOTE

If the coil is represented by  or  , the relay is within the PMC, and the contact uses  or  .

7.3 LADDER DIAGRAM FORMAT

(1) Format
The size should be A3 or A4 (JIS standard).

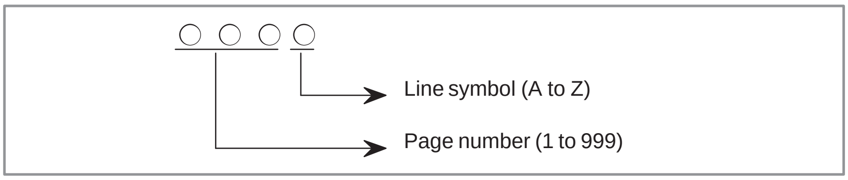
(2) Columns are used for wiring.



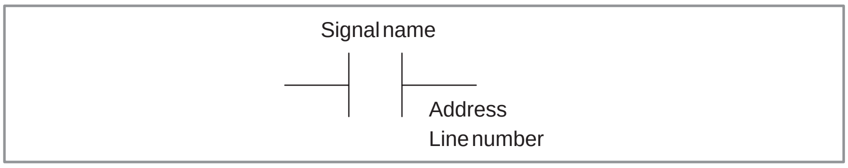
(3) Divide the circuits into several functions. And program the same function in a single program.

Example) Mode control.spindle control, turret control, APC control.

(4) Assign a line number to each line as follows:



(5) Write a relay contact with a signal name of the relay coil, line number and address.



(6) For complicated timing, timing chart should be on the same page of the ladder diagram.

(7) The meaning of the code numbers for the S, T, and M functions should be listed on the ladder diagram.

(8) The 1st level sequence part should be written at the beginning of the ladder diagram.

(9) The following data should be written on the first page of the ladder diagram:

(i) The sequence program design number
Machine tool builder shall assign design numbers of sequence program and ROMs and manage them.

(ii) Description of symbol

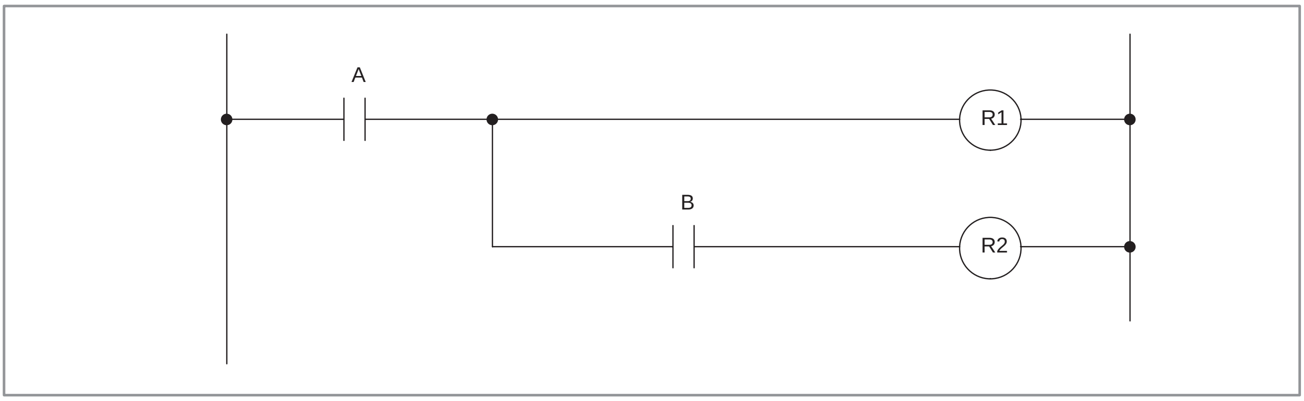
(iii) Setting table of timer, counter, and PMC parameters and meaning of them.

(iv) Description of functional instruction.

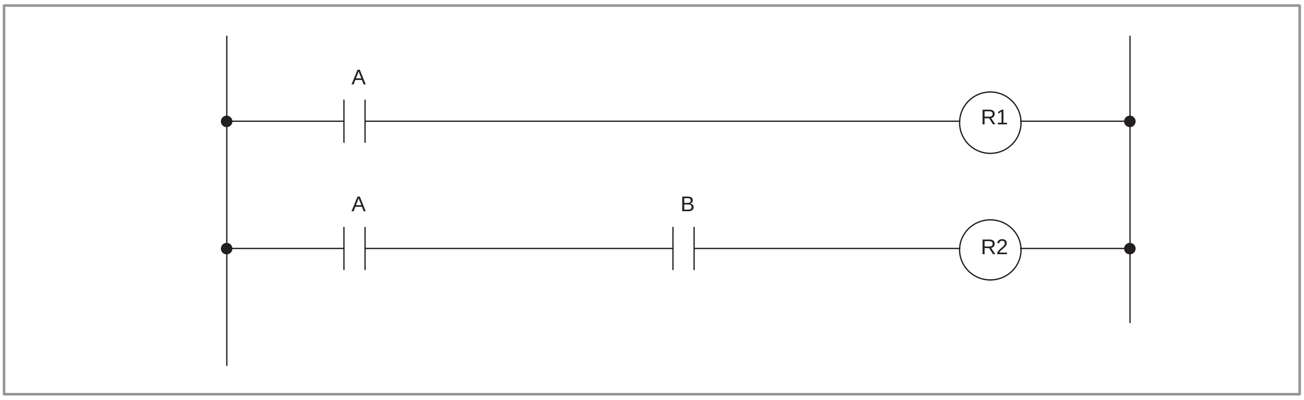
(10) Easy-to-understand name should be assigned.

7.4 INFINITE NUMBER OF RELAY CONTACTS

A general relay sequence circuit has a finite number of contacts, so several relays use one contact in common so as to reduce the number of contacts used as much as possible.



The PMC is considered to have an infinite number of relay contact and is written as in the figure below.



8 MISCELLANEOUS ITEM

To create a ladder program related to the axis-control function by the PMC, refer to the subsection, ”Axis-control function by the PMC,” in the Connecting Manual.

9 SEQUENCE PROGRAM STRUCTURING

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

With the conventional PMC, a Ladder program is described sequentially. By employing a Ladder language that allows structured programming, the following benefits are derived:

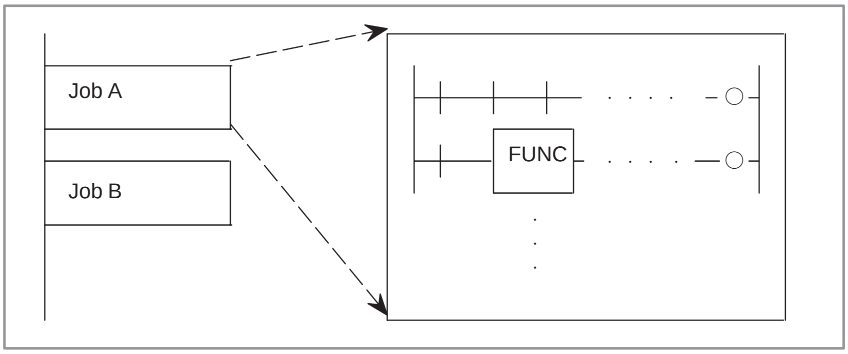
- A program can be understood and developed easily.
- A program error can be found easily.
- When an operation error occurs, the cause can be found easily.

9.1 EXAMPLES OF STRUCTURED PROGRAMMING

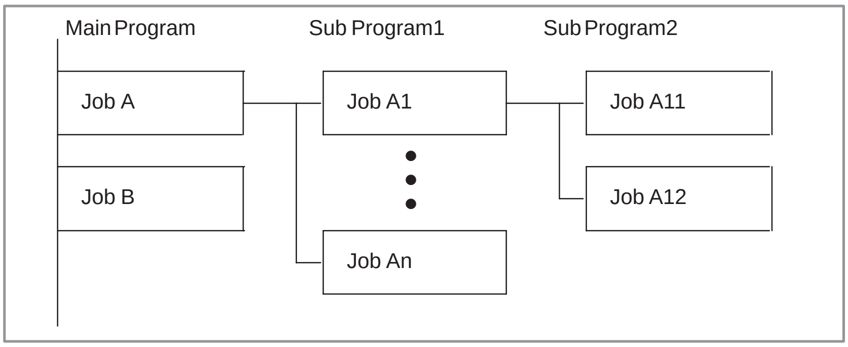
9.1.1 Implementation Techniques

Three major structured programming capabilities are supported.

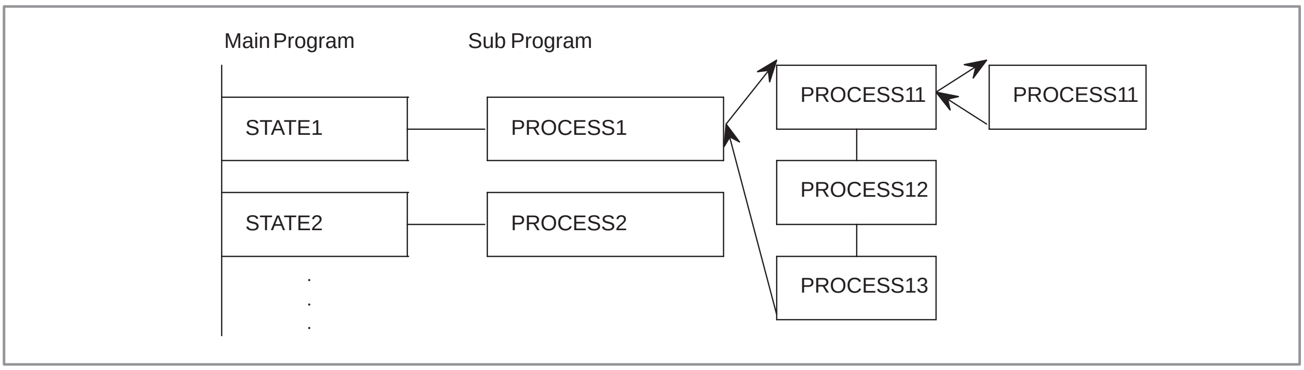
(1) Subprogramming
A subprogram can consist of a Ladder sequence as the processing units.



(2) Nesting
Ladder subprograms created in 1 above are combined to structure a Ladder sequence.



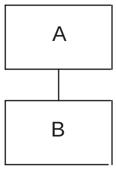
(3) Conditional branch
The main program loops and check whether conditions are satisfied. If a condition is satisfied, the corresponding subprogram is executed. If the condition is not satisfied, the subprogram is skipped.



9.1.2 Applications

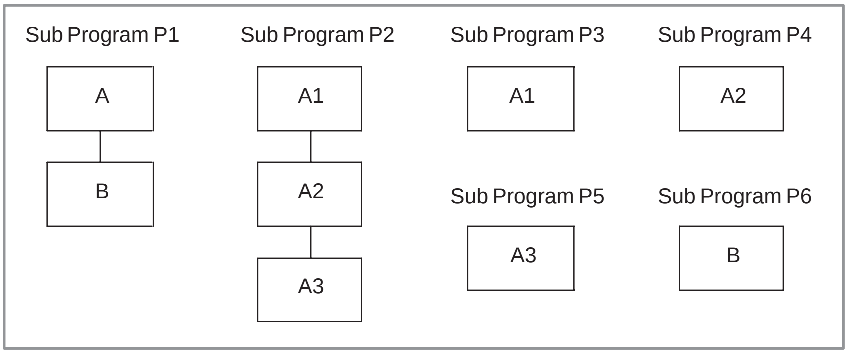
(1) Example

Suppose that there are four major jobs.

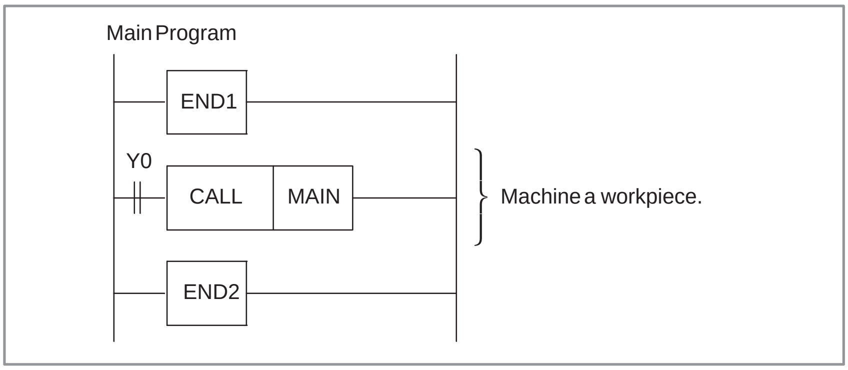


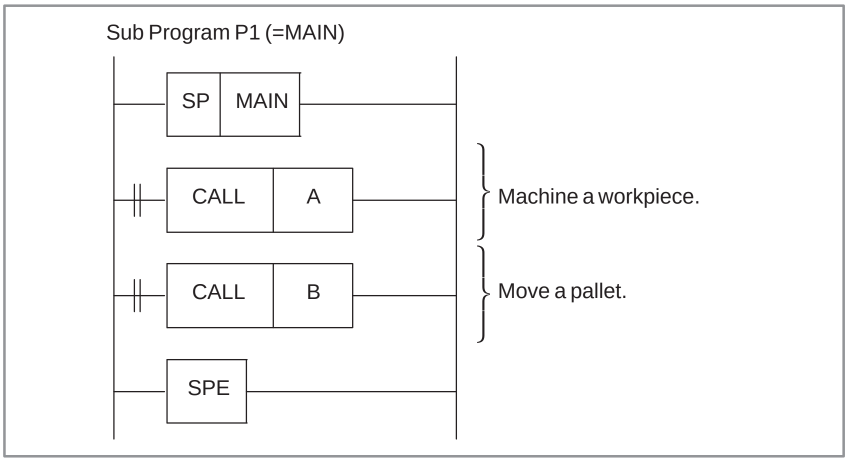
If Y0 is 1, a request to machine a workpiece is assumed, and processing is performed (with a condition).
A : ① Pick up a workpiece from a pallet. (A1)
②Machine the workpiece. (A2)
③Return the workpiece to the pallet. (A3)
B : ④Move the pallet.

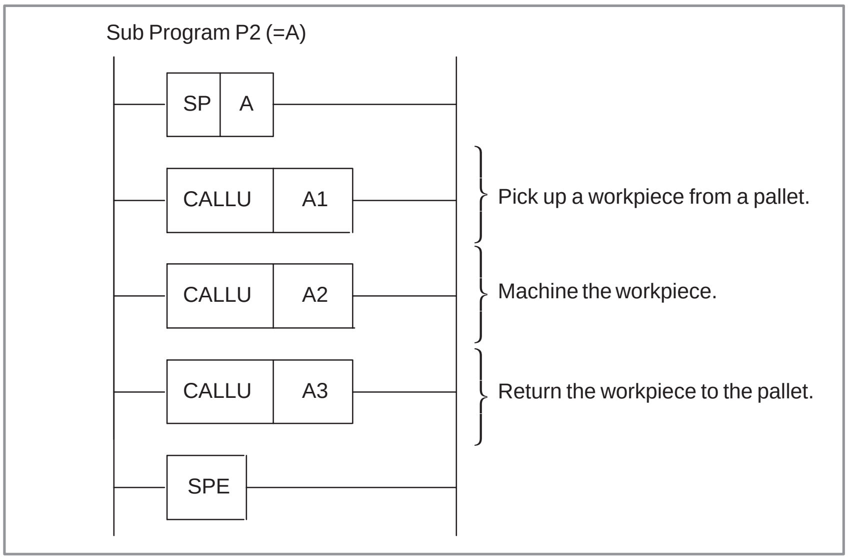
(2) Program structure

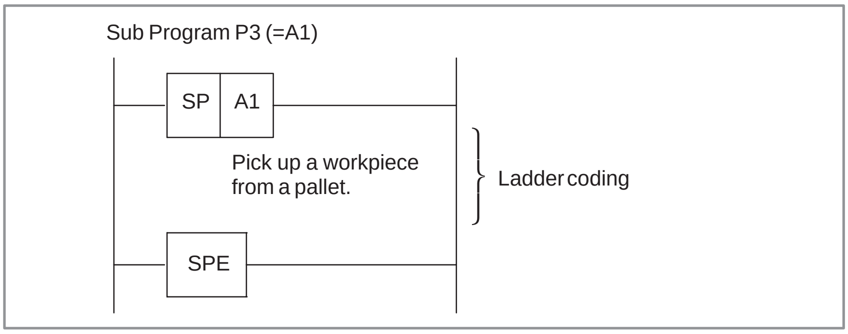


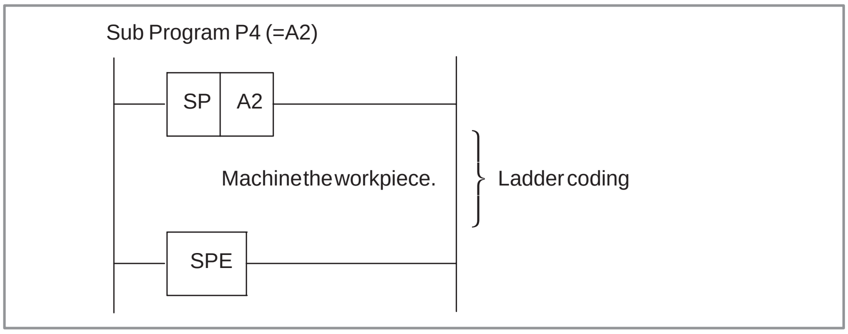
(3) Program description

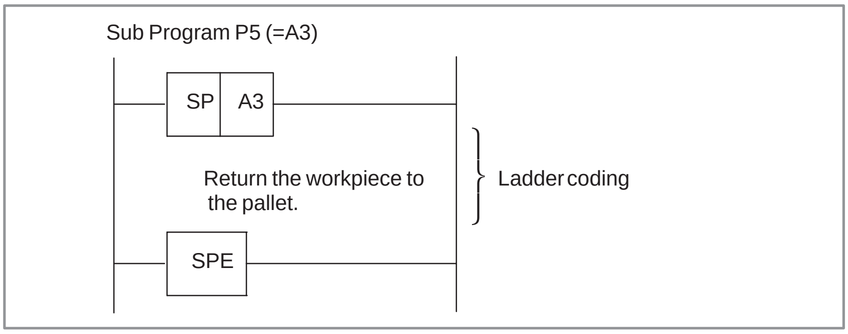


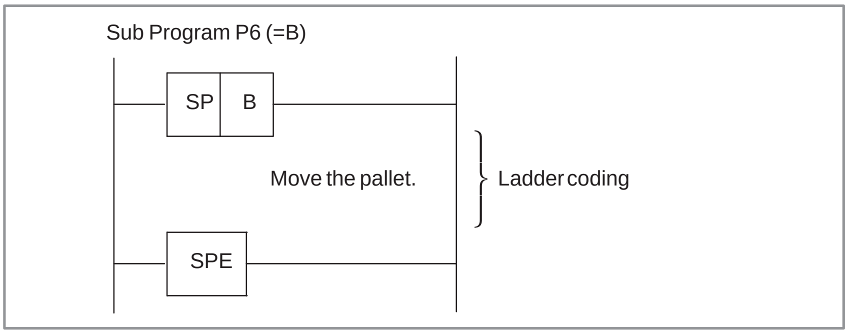



Sub Program P2 (=A)
SP
A
CALLU
A1
Pick up a workpiece from a pallet.
CALLU
A2
Machine the workpiece.
CALLU
A3
Return the workpiece to the pallet.
SPE


Sub Program P3 (=A1)
SP
A1
Pick up a workpiece from a pallet.
Ladder coding
SPE


Sub Program P4 (=A2)
SP
A2
Machine the workpiece.
Ladder coding
SPE


Sub Program P5 (=A3)
SP
A3
Return the workpiece to the pallet.
Ladder coding
SPE



9.1.3 Specifications

(1) Main program

The main program is the Ladder program consisting of the first- and second-level Ladder programs. One, but only one, main program can be created. A subprogram cannot be called from the first-level Ladder program. Any number of subprograms however, can be called from the second-level Ladder program. The functional instructions JMP and COM must be completed within each main program or subprogram.

(2) Subprogram

A subprogram is a program called by the second-level Ladder program. It is a program unit starting with the functional instruction SP and ending with the functional instruction SPE. Up to 512 subprograms can be created for one PMC.

(3) Nesting

A subprogram can call another subprogram. The maximum nesting depth is eight levels. Recursive calls are not allowed.

9.2 SUBPROGRAMMING AND NESTING

9.2.1 Function

Conditional JUMP (or unconditional JUMP) is coded in the main program, and the name of a subprogram to be executed is specified. In the subprogram, the name of the subprogram and a Ladder sequence to be executed are coded.

When a subprogram is named Pn (program name), and this name is specified in conditional JUMP, the subprogram is executed by calling it.

A symbol and comment can be added to Pn to assign a subroutine name.

In the example shown in Fig. 9.2.1, the main program calls three subprograms. These calls are all conditional calls. Subprogram P1 is named SUBPRO. It calls subprogram PROCS1 unconditionally.

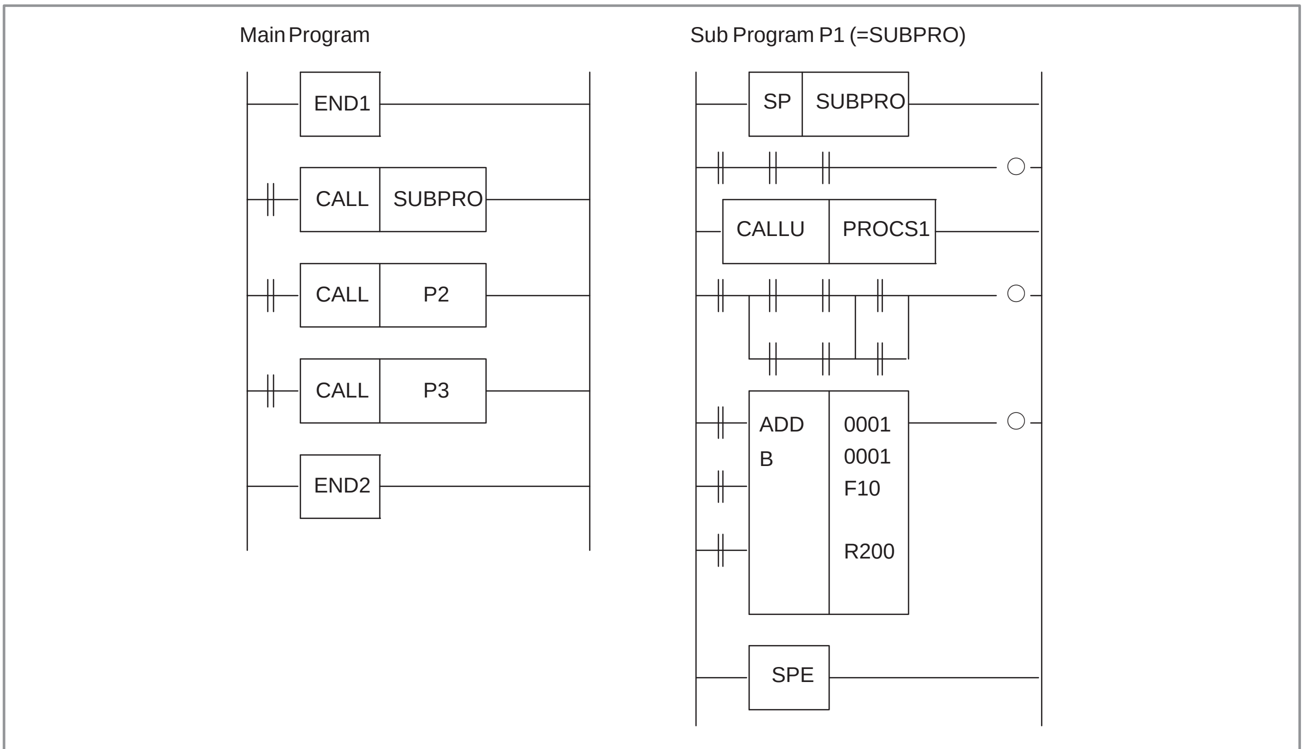
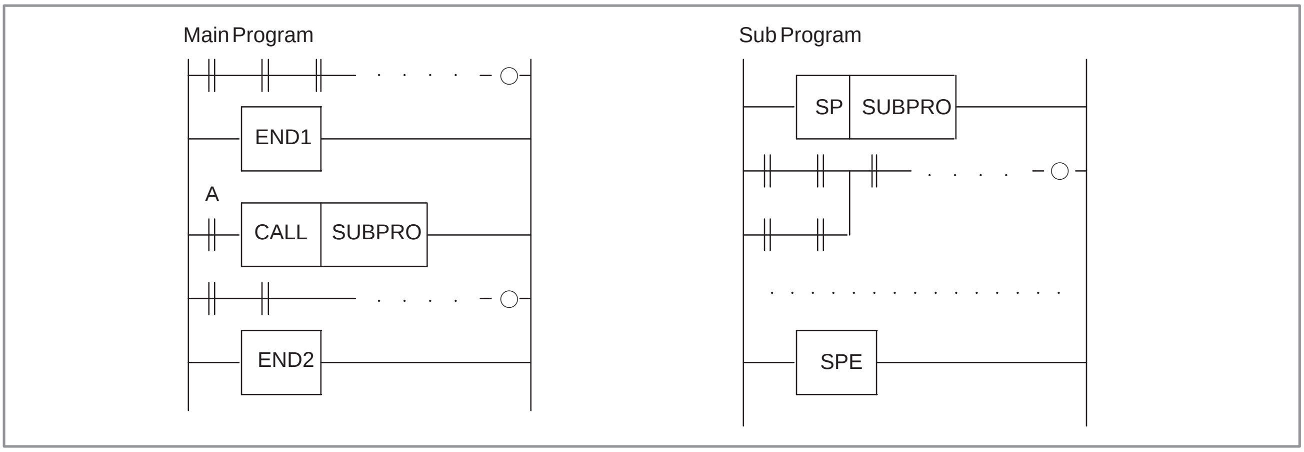
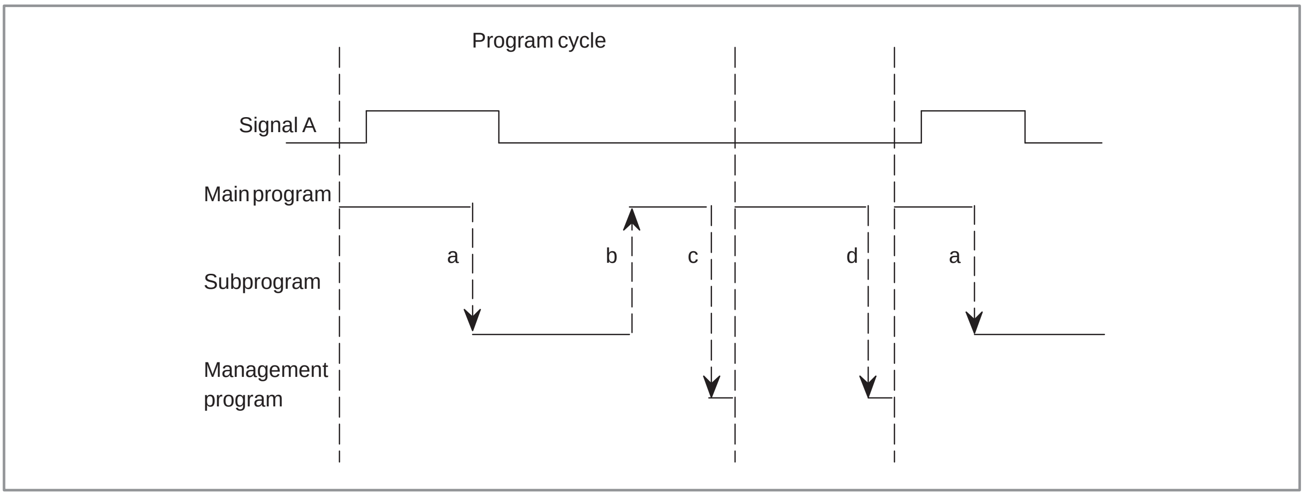


Fig. 9.2.1 Example of subprogramming and nesting

9.2.2 Execution Method

The main program is always active. Subprograms on the other hand, are active only when called by another program.
In the following example, subprogram SUBPRO is called by signal A.





Flow of execution

a : Functional instruction CALL calls a subprogram in order to transfer control to the subprogram.

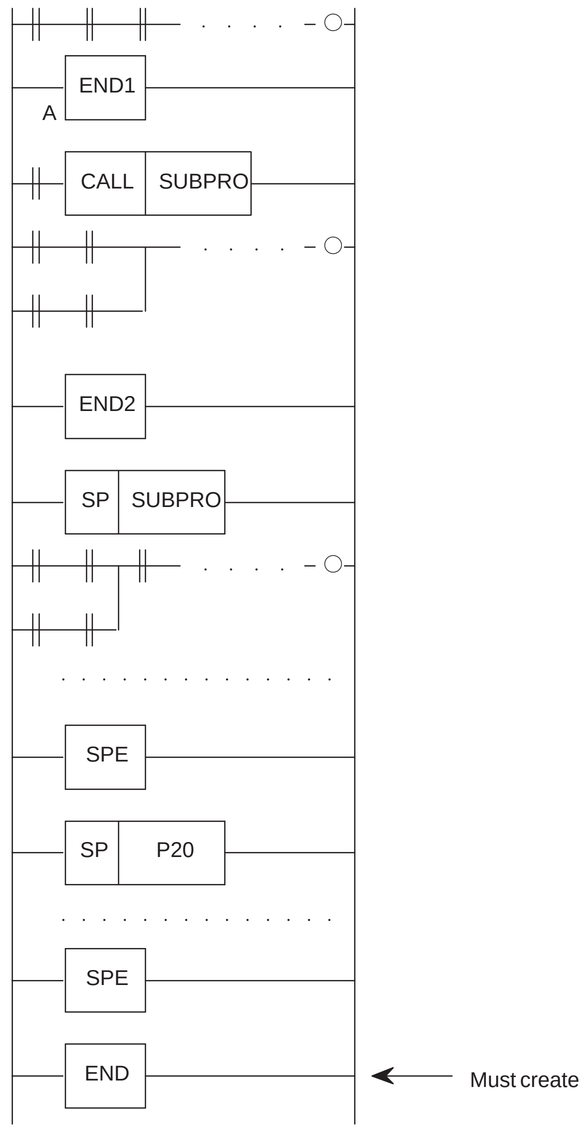
b : When the end of the subprogram is reached, control is returned to the main program.

c : When the end of the main program is reached, the management program performs Ladder program postprocessing.

9.2.3 Creating a Program

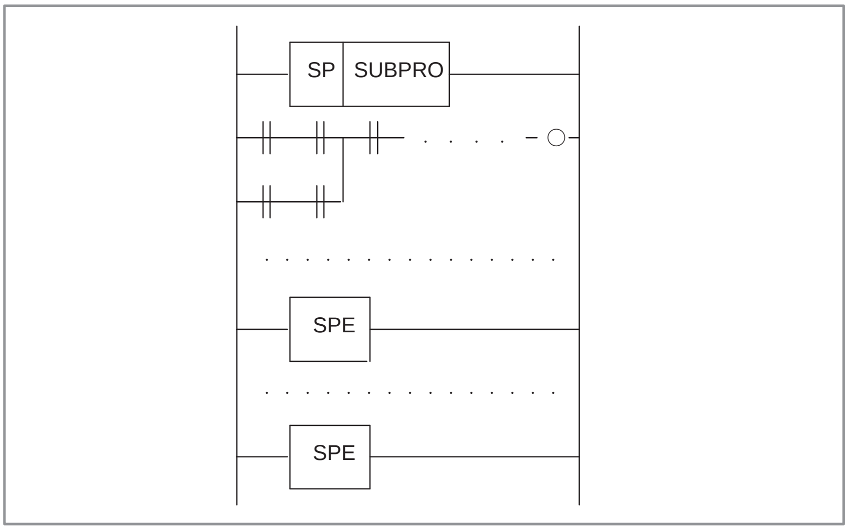
Create subprograms in the same way as the first-, second-, and third-level Ladder programs.

Example of creation

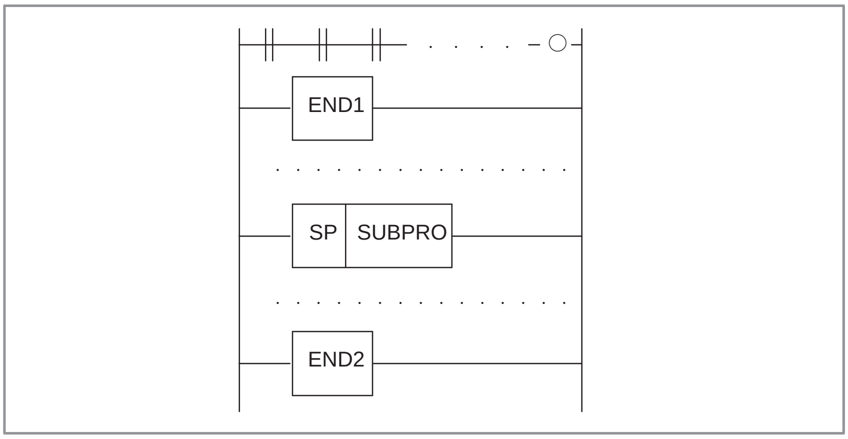


Inhibit items

(1) Subprograms are nested.



(2) A subprogram is created within the first-, second-, or third-level Ladder program.



9.3 CAUTIONS FOR SUBROUTINES

a) DISPB

b) EXIN

c) WINDR (only low-speed response)

d) WINDW (only low-speed response)

e) MMCWR

f) MMCWW

g) MMC3R

h) MMC3W

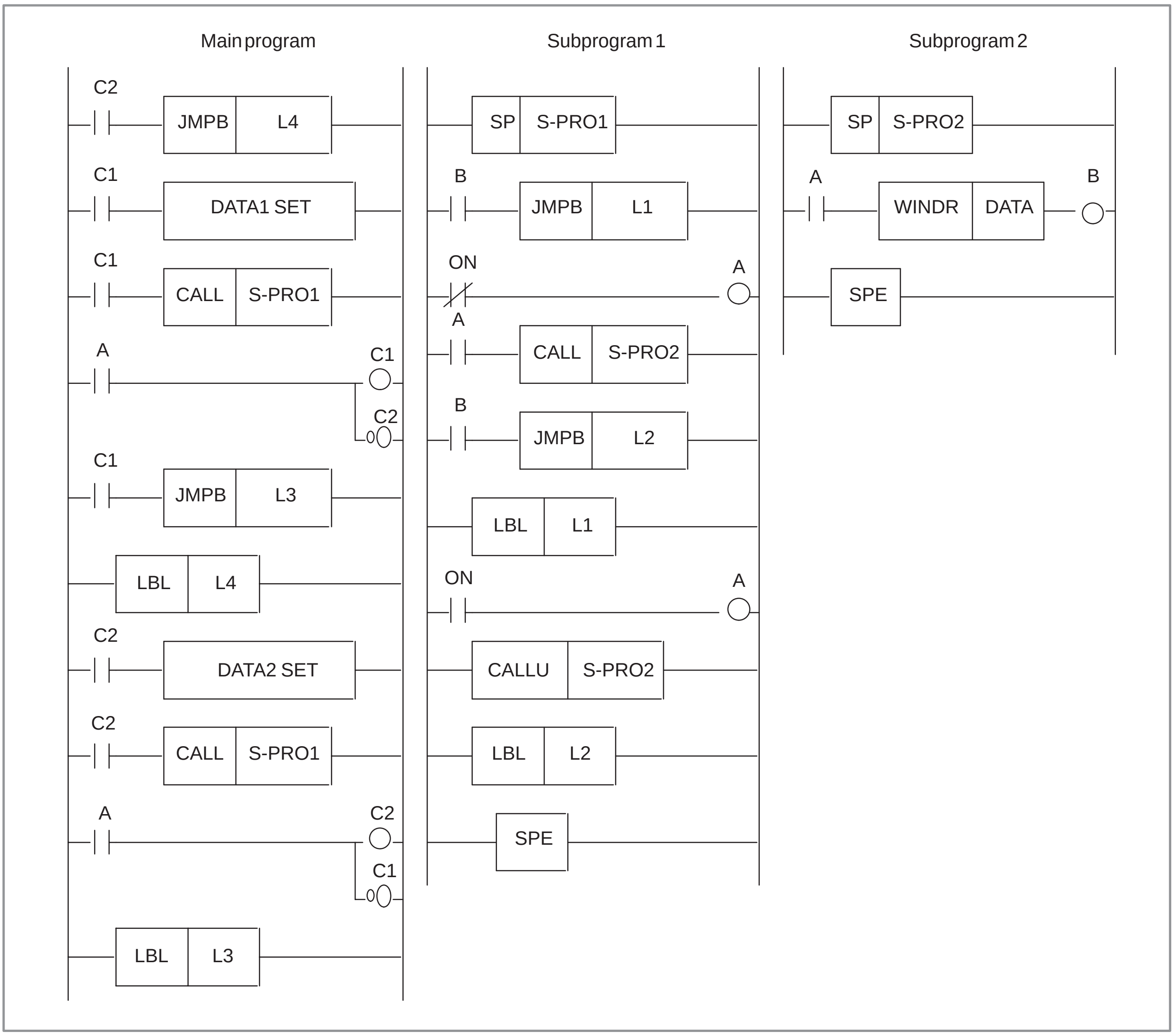
When you use the above-mentioned functional instructions, ACT=1 must be held until the transfer completion information(W1) becomes 1. Therefore, be careful of the following when using those instructions in subprograms.

- Do not stop calling the subprogram at the state which has not been completed yet, that is executed still while using the instructions in the subprogram.
  (In other words, do not set the ACT of the CALL instruction to 0)
  → If you do it the function of the instructions after that is not guaranteed.
- Call the subprogram from other subprograms at the state which has not been completed yet while using the instructions in the subprogram.
  → The movement of the above-mentioned functional instruction after that is not guaranteed so that the last functional instruction may be processing the instruction.

Then, when the subprogram, in which the above-mentioned functional instruction is used, is called from two or more places, it is necessary to control the subprogram exclusively. The case of the WINDR instruction (low-speed response) is given as an example here.

Example)

When subprogram is called from two places. (The WINDR instruction is used)



Description)

Subprogram 1 controls ACT(A)and W1(B)of WINDR (subprogram 2).
By "A" controlled in subprogram 1, the main program decides which relay (C1,C2) to be effective.
When the WINDR instruction is completed, the following data will be set and the other CALL instruction is started.
It keeps working in this way.

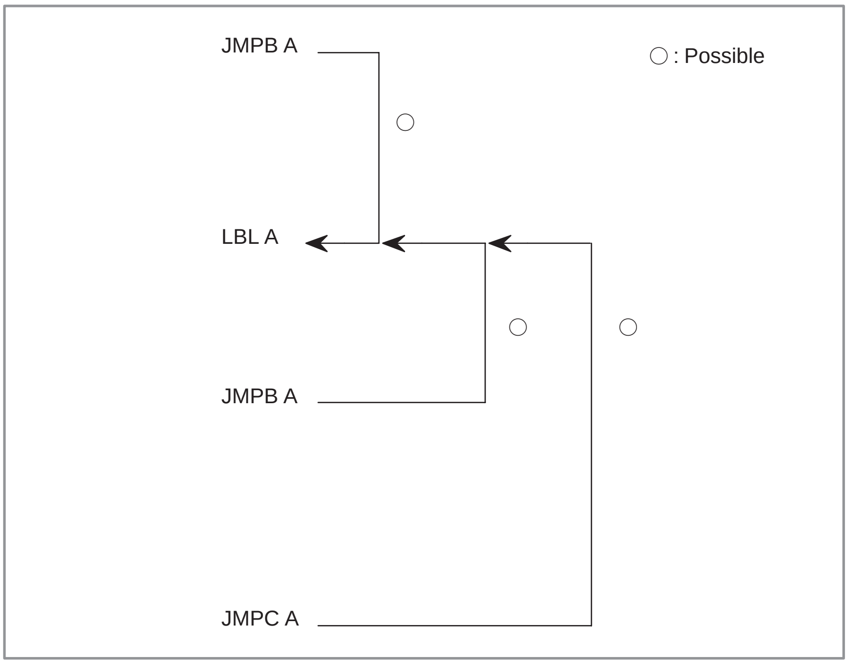
10 JMP INSTRUCTIONS WITH LABEL SPECIFICATION

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	×	○	○	○	○	○

10.1 SPECIFICATIONS

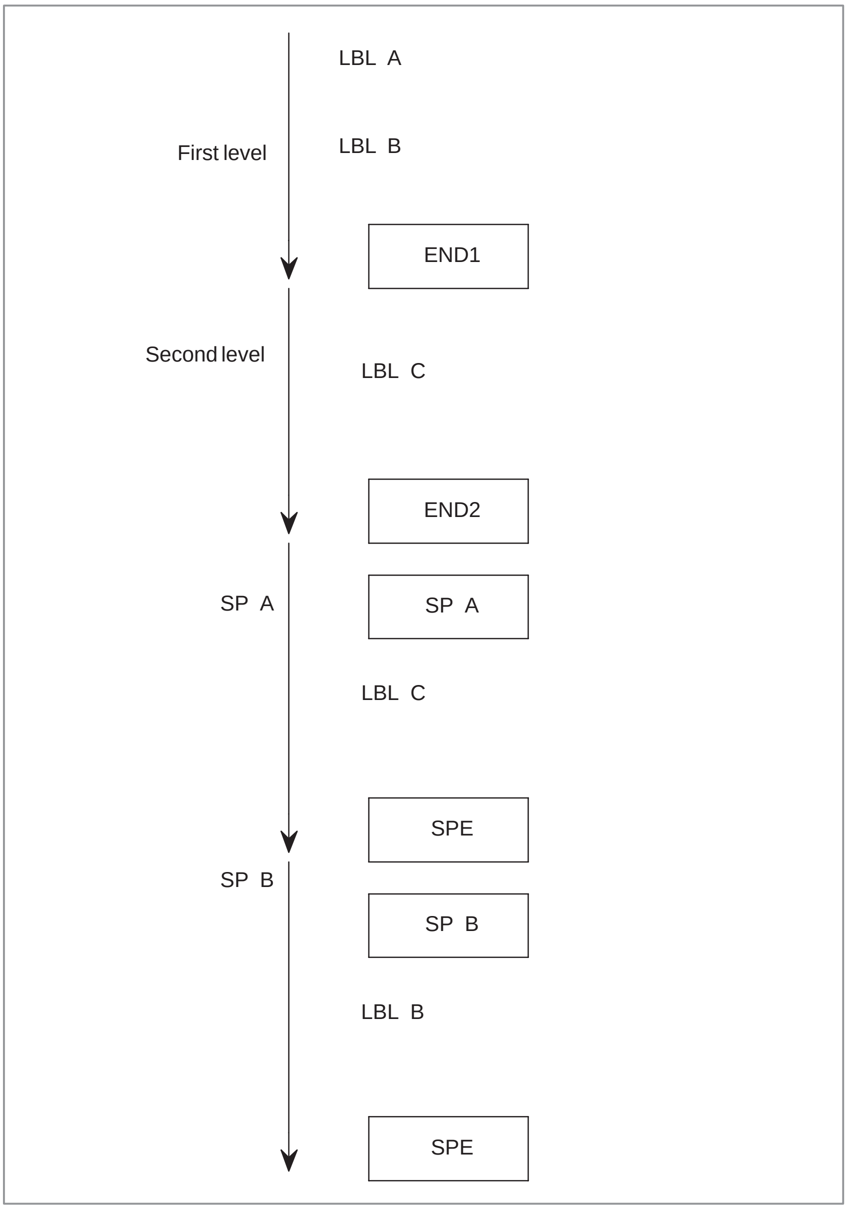
(1) Relationship between JMPB/JMPC and LBL
(Forward and backward jumps to the same label are possible.)



CAUTION

The specifications allow backward jumps. A backward jump, however, may result in an infinite loop or cause the execution time of the first-level Ladder program to exceed 1.5 ms (or 5 ms). Create a program carefully so an infinite loop does not occur.

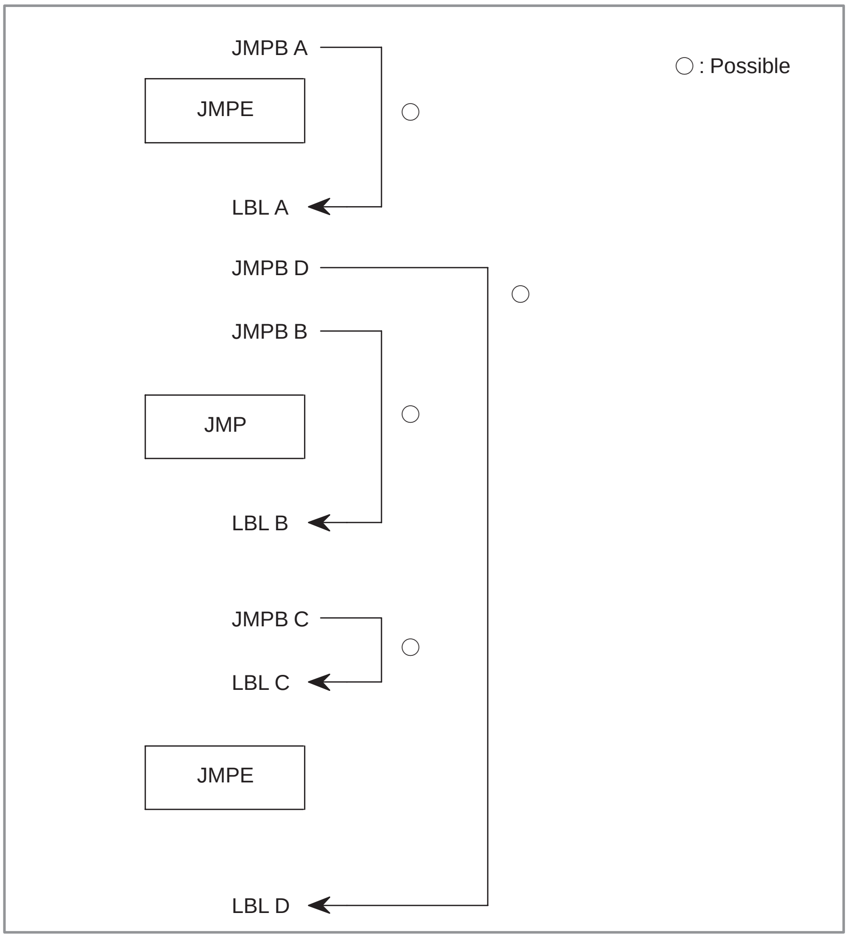
(2) Same label
(A label can be used more than once as long as it is unique within the main program or each subprogram.)



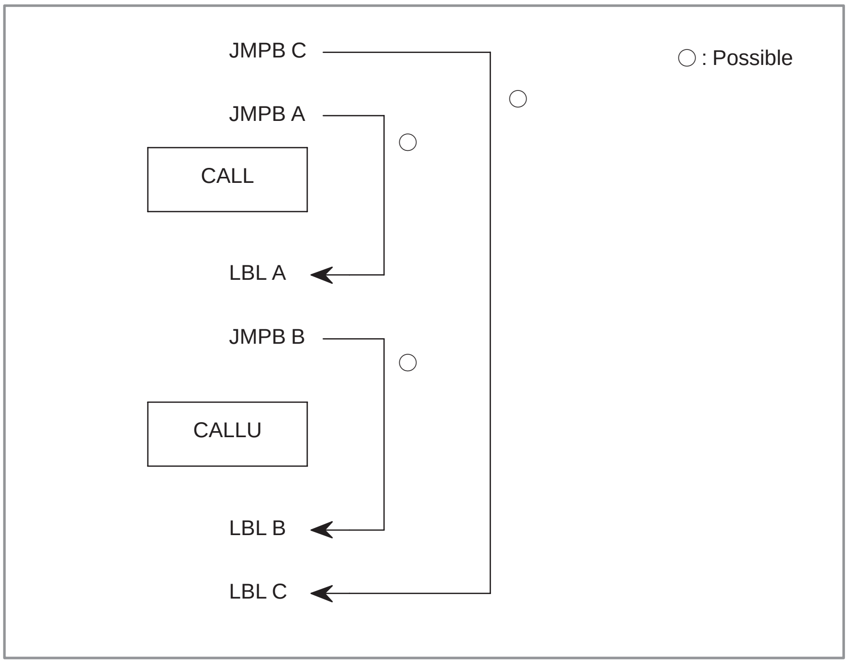
NOTE
As mentioned in (8) of Section 10.2, the same label must not exist in the first- and second-level Ladder programs.

(3) Number of labels
First-and second-level Ladder programs : Up to 256 labels
Subprogram : Up to 256 labels for each subprogram
Label number : L1 to L9999

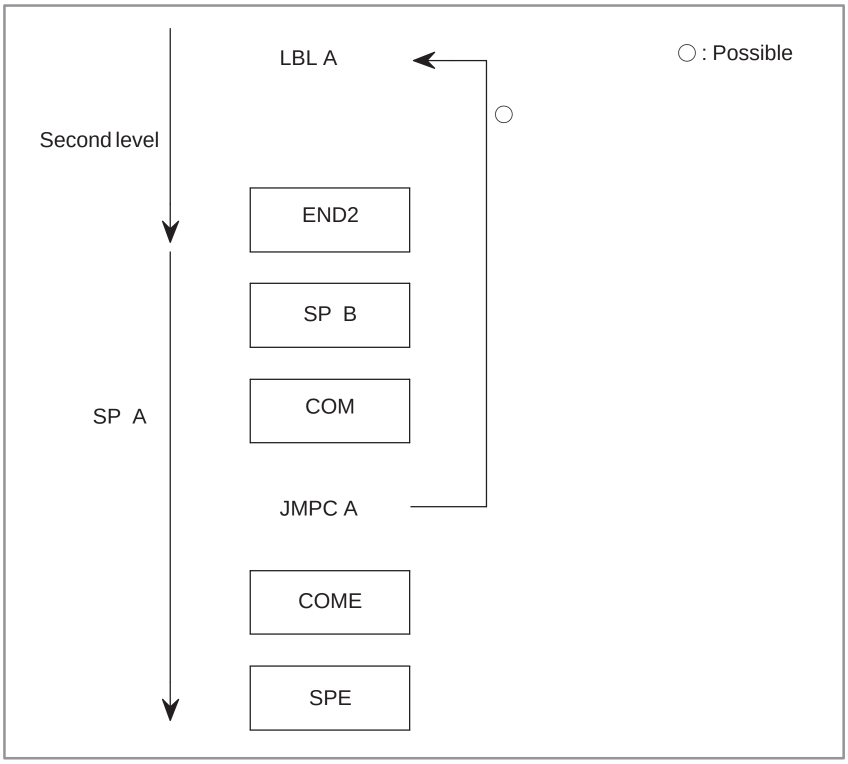
(4) Relationship between JMP/JMPE and JMPB/JMPC
(JMPB and JMPC can be used with JMP and JMPE freely.)



(5) Relationship between CALL/CALLU and JMPB/JMPC
(JMPB and JMPC can be used with CALL and CALLU freely.)

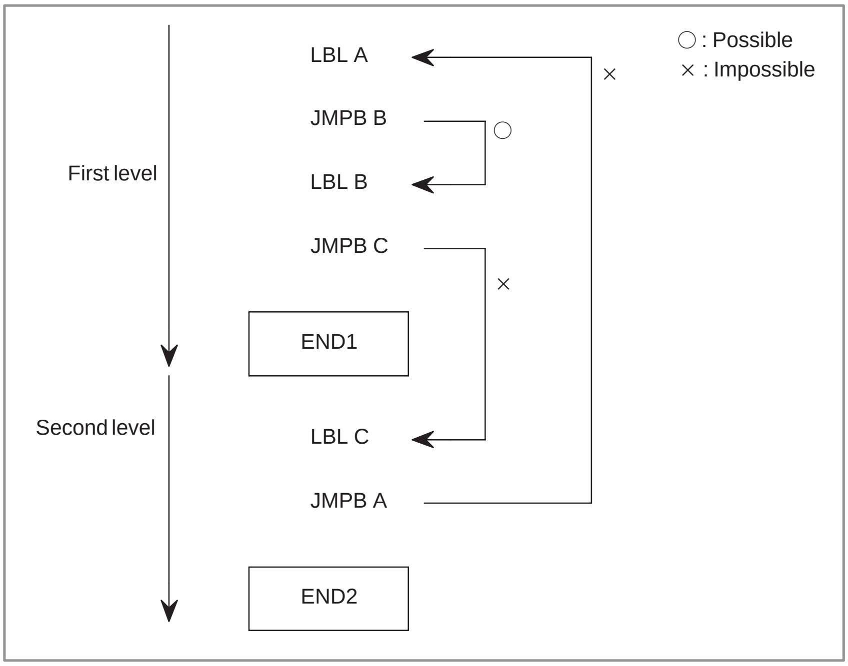


(6) Position of JMPC
(JMPC coded between COM and COME can cause a jump.)

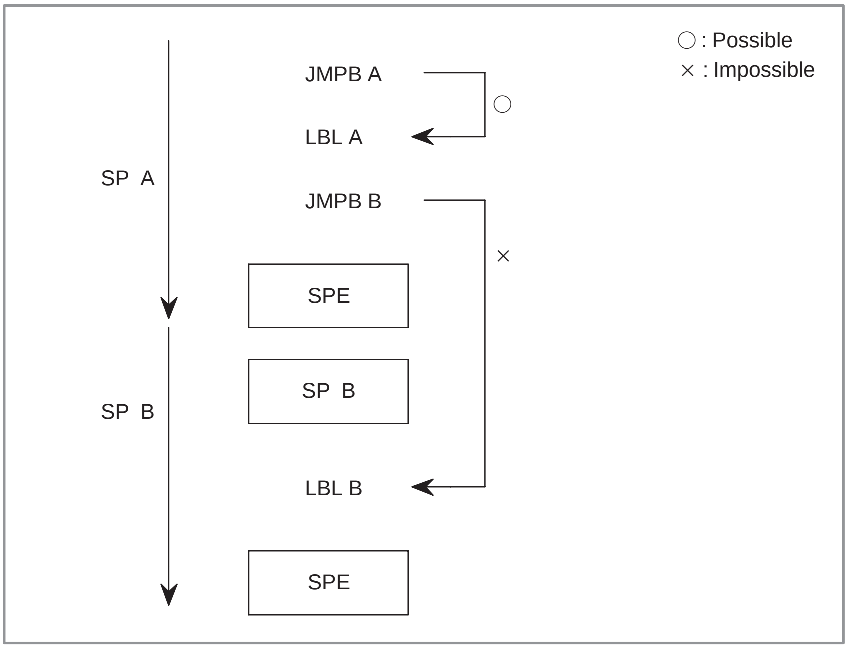


10.2 RESTRICTIONS

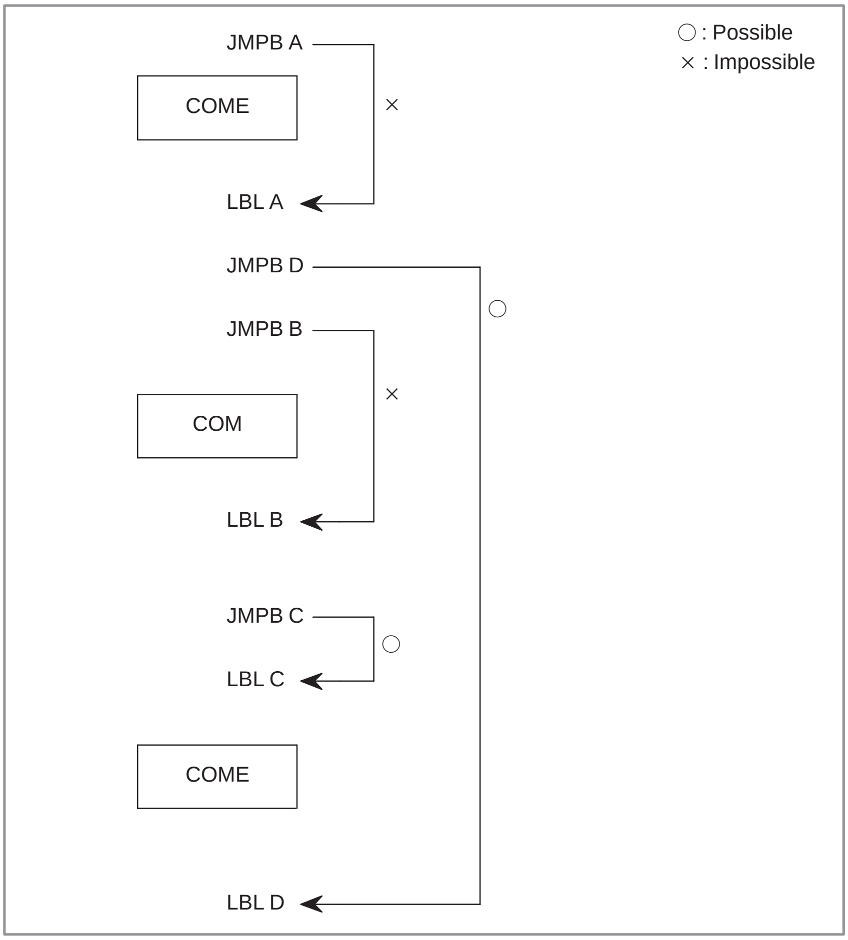
(1) Jump destination of JMPB (1)
(A jump over END1 or END2 is inhibited.)



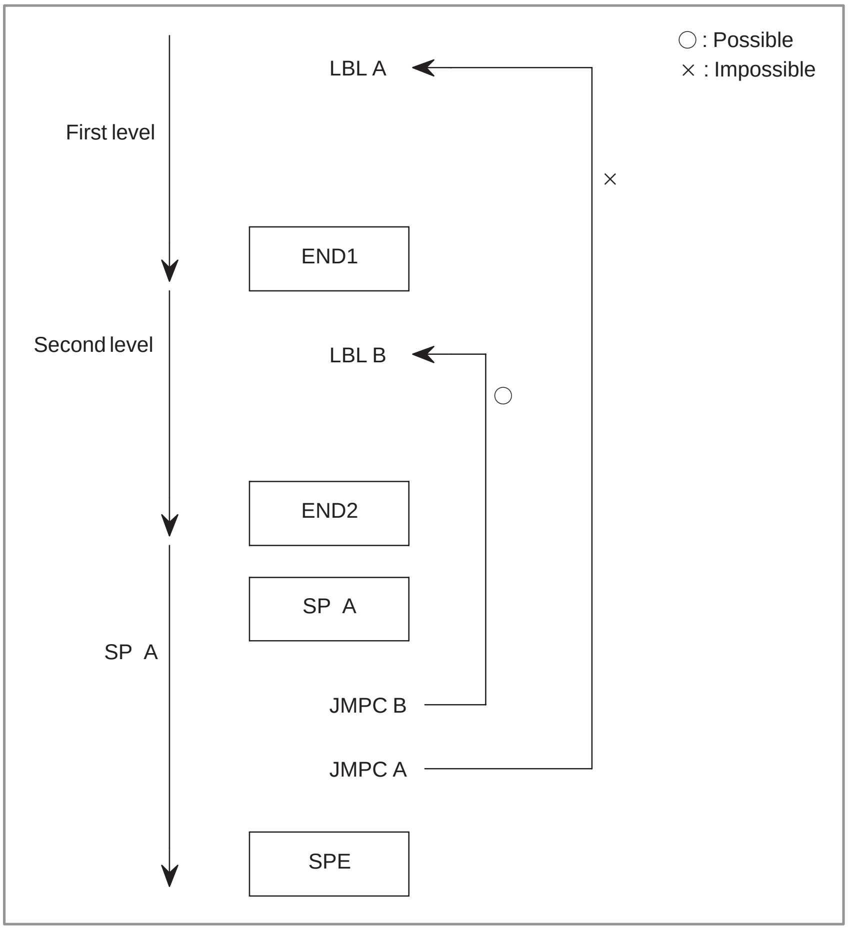
(2) Jump destination of JMPB (2)
(A jump must be performed within a subprogram.)



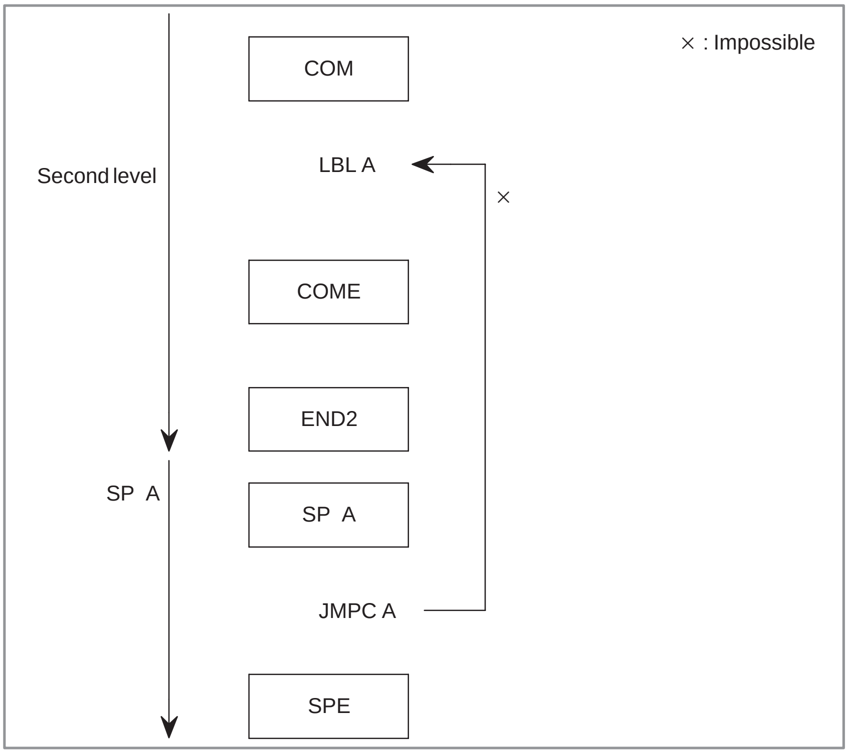
(3) Jump destination of JMPB (3)
(A jump over COM or COME is inhibited.)



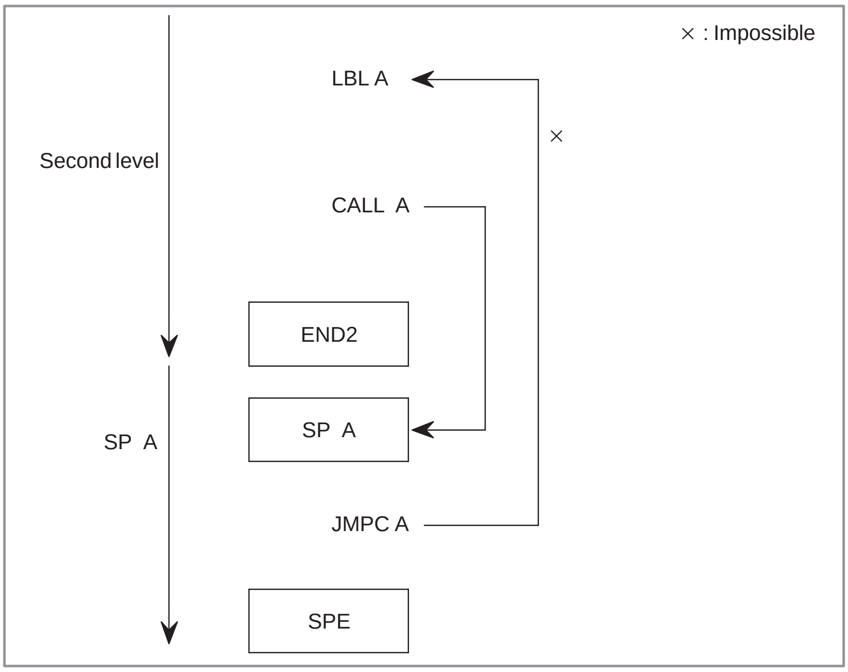
(4) Jump destination of JMPC (1)
(A jump to the first-level Ladder program is inhibited.)



(5) Jump destination of JMPC (2)
(A jump to a label between COM and COME is inhibited.)



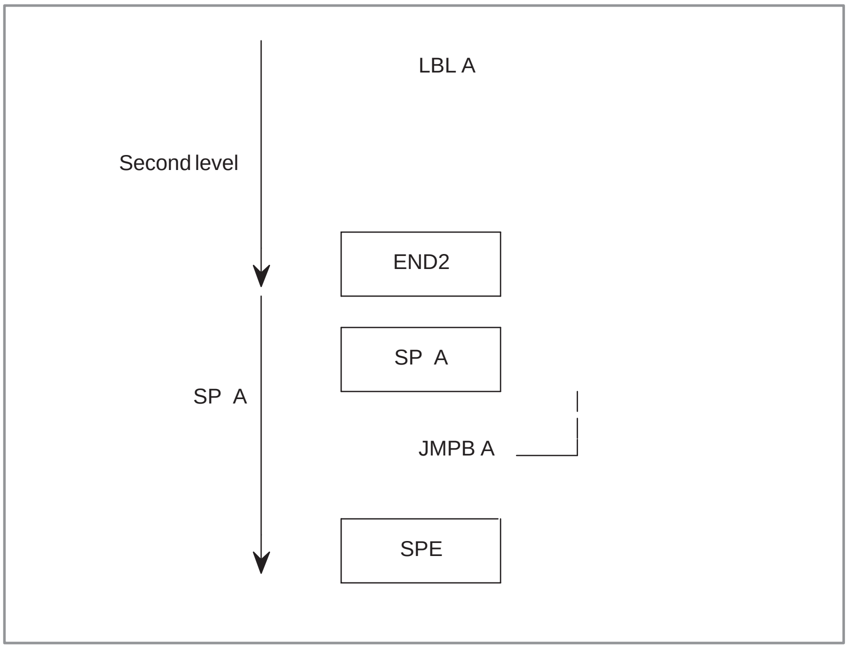
(6) Jump destination of JMPC (3)
(Control must not be returned to a label that appears earlier than the instruction that has called the subprogram.)



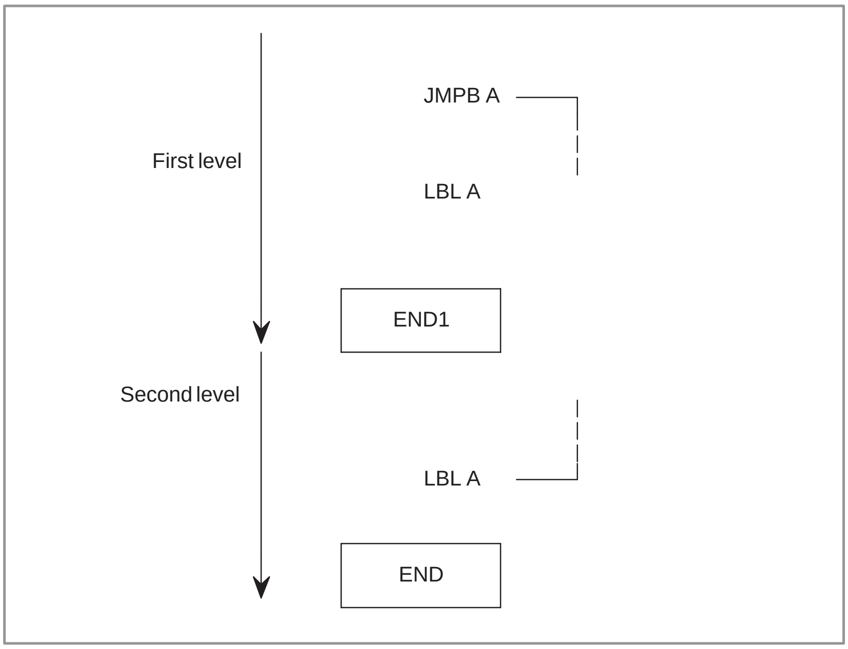
CAUTION

Although Ladder diagrams can be edited, editing a Ladder diagram may cause an infinite loop. So, be careful not to program such processing.

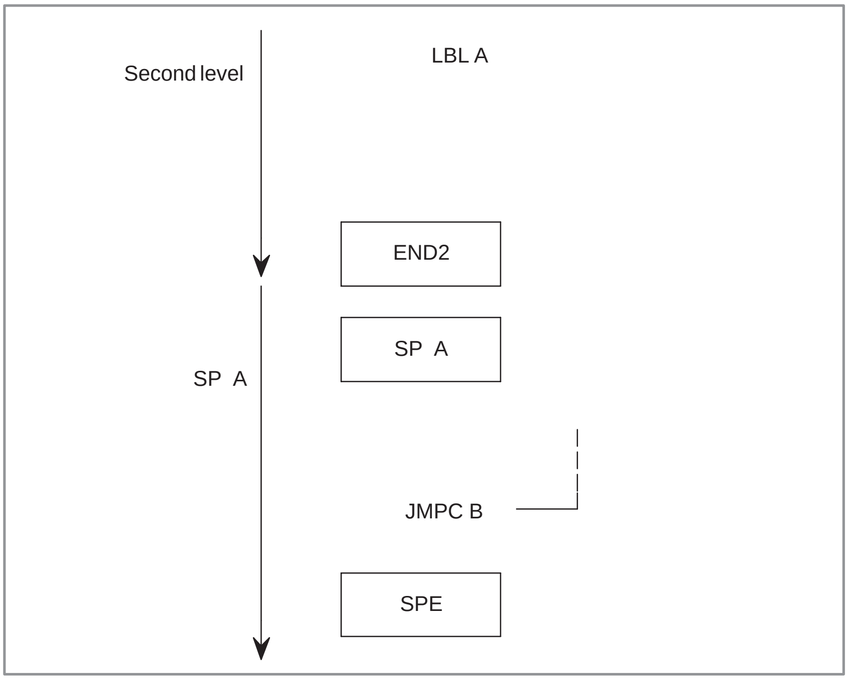
(7) LBL for JMPB (1)
(There is no LBL in the same subprogram.)



(8) LBL for JMPB (2)
(The same LBL is found in the first- and second-level Ladder programs.)



(9) LBL for JMPC
(There is no LBL in the second-level Ladder program.)



11 INTERRUPT–TYPE PMC FUNCTION

NOTE
This function is available only in the PMC–SB5/SB6 for the Power Mate *i*–H. It requires the interrupt–type PMC option.

11.1 OVERVIEW

This function enables the discontinuation of the current sequence program (second–level sequence section) and the execution of another sequence program (conventional first–level sequence section) when urgent execution of the latter sequence program becomes necessary.
The interrupt program is called for execution on the rising and/or falling edges of an interrupt input (one of eight points X1003.0 to X1003.7).

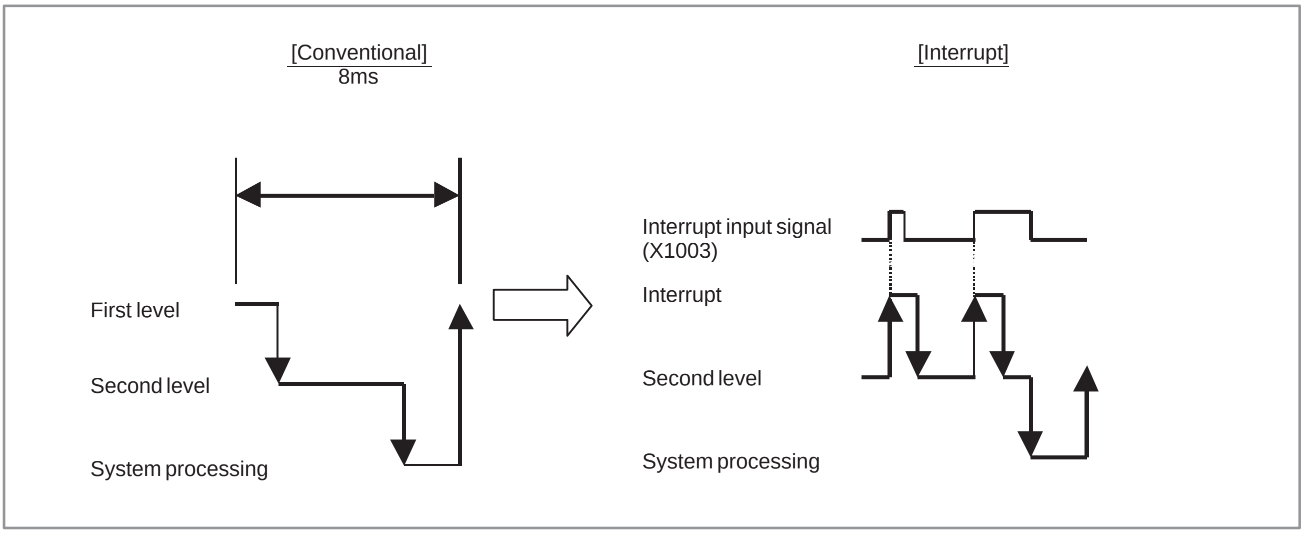


Fig. 11.1 Sequence program execution order

11.2 SETTING

The following NC parameters specify the conditions for the interrupt input signals.

- NC parameters (bit type)

	#7	#6	#5	#4	#3	#2	#1	#0
8731	EPMC7	EPMC6	EPMC5	EPMC4	EPMC3	EPMC2	EPMC1	EPMC0

EPMCn Whether to use bit n of X1003 as an interrupt–type PMC

0 : Not used.
1 : Used.

	#7	#6	#5	#4	#3	#2	#1	#0
8732	UPEG7	UPEG6	UPEG5	UPEG4	UPEG3	UPEG2	UPEG1	UPEG0

UPEGn Whether to use the interrupt–type PMC on the rising edge of a signal defined by bit n of X1003

0 : Not used.
1 : Used.

	#7	#6	#5	#4	#3	#2	#1	#0
8733	DWEG7	DWEG6	DWEG5	DWEG4	DWEG3	DWEG2	DWEG1	DWEG0

DEWGn Whether to use the interrupt–type PMC on the falling edge of a signal defined by bit n of X1003

0 : Not used.
1 : Used.

(Example) The following settings specify that the rising edge of bit 0 of X1003, the falling edge of bit 1 of X1003, and both the rising and falling edges of bit 7 of X1003 be used as conditions for interrupt program execution.

	#7	#6	#5	#4	#3	#2	#1	#0
8731	1	0	0	0	0	0	1	1
8732	1	0	0	0	0	0	0	1
8733	1	0	0	0	0	0	1	0

11.3 INTERRUPT PROCESSING

11.3.1 Interrupt Program

This function uses a conventional first–level sequence section as an interrupt program.

Only one interrupt program is supported. More than one interrupt input condition (up to 8 points from bit 0 to bit 7 of X1003) can be specified as the conditions for executing the interrupt program. In this case, if any one condition is satisfied, the interrupt program is executed. To define a different process in the interrupt program for each interrupt input signal, create the interrupt program by referencing Section 11.4, "Sequence Program Examples."

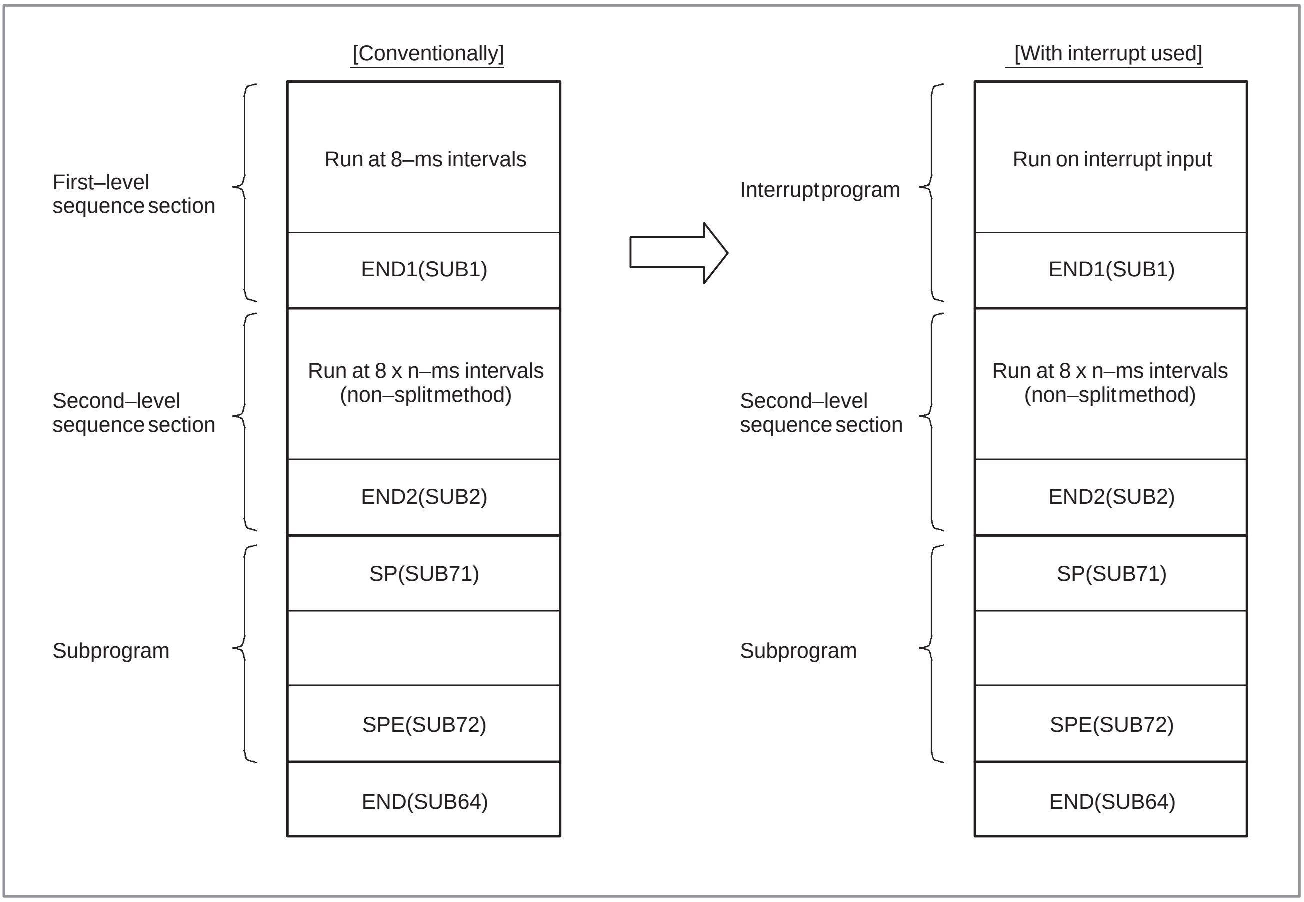


Fig. 11.3.1 Sequence program configuration

11.3.2 Input/Output Signal Processing

Input/output signals (F, G, X, Y addresses) between the NC and machine are processed asynchronously except for built–in I/O units (X1000 to X1003, X1007, Y1000 to Y1002). (Usually, processing for NC <–> PMC (F/G), Link master (X0 to X127), and I/O Link slave (X1020 to X1051, Y1020 to Y1051) is performed at 8–, 2–, or 8–ms intervals, respectively.)
When an interrupt–type PMC is used, input signals (X1000 to X1003, X1007) from a built–in I/O unit are read immediately before the interrupt program is executed. Output signals (Y1000 to Y1002) to a built–in I/O unit are written immediately after the interrupt program is executed. Built–in I/O signals are processed at 8–ms intervals even when the interrupt program is not executed.
The same signal is read from both X1007 and X1003, but signal changes in X1007 can be read faster. Therefore, ladder programs should use X1007 rather than X1003 when referencing an input signal.
To enable the interrupt program to judge interrupt input trigger conditions, the interrupt request status (which of the signals defined by bit 0 to bit 7 of X1003 causes the interrupt) is output to an internal relay (R9021).

- Interrupt relay R9021 (interrupt request status)

	#7	#6	#5	#4	#3	#2	#1	#0
R9021								

This relay indicates the status of an interrupt request corresponding to each interrupt input signal (bit 0 to bit 7 of X1003).
When a bit is set to 1, it indicates that the corresponding signal is requesting an interrupt.

11.3.3 Response Time

The interrupt program is executed within 0.5 ms (software response time) after an interrupt input signal is received. If there is an interrupt–inhibited interval, the execution of the interrupt program is deferred by the corresponding time. If another interrupt input signal is received when the interrupt program is already running, the newly received signal is kept waiting until the current execution ends.

11.3.4 Execution Time

The execution time of the interrupt program must be within about 6 ms, even in the worst case.
If an interrupt program whose execution time is longer than 6 ms is executed, or if too many interrupt requests are made, a PMC alarm (WN08 INTERRUPT LADDER TIME OVER) is issued.
The maximum execution time of the interrupt program can be checked using the internal relay (R9022 to R9023).

- Internal relay R9022 to R9023 (interrupt program maximum execution time)

R9022	Interrupt ladder program maximum execution time [10 μs]

This data is the maximum allowable execution time for the interrupt ladder program. (0 to 655,350 μs)

- PMC alarm message (alarm screen)

Alarm message	Meaning and response
WN08 INTERRUPT LADDAR TIME OVER	The execution time of the interrupt ladder has exceeded the allowable value (about 6 ms). The interrupt program is too large, or too many interrupt requests were issued. So, the second level of the ladder has operated. (Response) Make the interrupt program smaller, or reduce the number of interrupt requests.

CAUTION
Check the execution time of the interrupt program with the internal relay mentioned above, and change the program to keep the regular interrupt program execution time within 2 ms.

11.3.5 Interrupt Enable/Disable/Clear

The WINDW machine instruction (with function code 10000) is used to enable, disable, and clear (that is, nullify all internally stored requests for) interrupts in the second–level program.
If an interrupt request arises when an interrupt has been disabled, interrupt program execution is deferred until an interrupt is enabled.
The second–level program is initially in an interrupt disabled state. Any interrupt request is rejected before the NC preparation completed signal MA (F1.7) is input. To enable immediately when the power is applied, execute the WINDW machine instruction (with function code 10000), using the preparation completed signal MA (F1.7) signal. (See Section 11.4, “Sequence Program Examples.”)

- Window function (with function code 10000 for enabling, disabling, and clearing interrupts)

[Description of data]
Interrupt–type PMC interrupts are enabled, disabled, and cleared.

[Input data structure]

Address	Contents	Remarks
Top address+0	(Function code) 10000	
+2	(Completion code) – (Need not be set)	
+4	(Data length) – (Need not be set)	
+6	(Data number) N (N = data specification)	N=1 to 3 1: Enable interrupts. (All conditions) 2: Disable interrupts. (All conditions) 3: Clear interrupts. (All conditions)
+8	(Data attribute) – (Need not be set)	
+10		

[Completion code types]

0: Interrupt setting has been completed normally.

3: The specified data number is incorrect. (Data other than 1, 2, or 3 was specified.)

6: The interrupt–type PMC option is not available. Alternatively, conditions for setting the interrupt input signals specified in the relevant NC parameters are incorrect.

[Output data structure]

Address	Contents
Top address+0	(Function code) 10000
+2	(Completion code) ? (See the above descriptions about the completion codes.)
+4	(Data length) ? (Data at input)
+6	(Data number) N (Data at input)
+8	(Data attribute) ? (Data at input)
+10	

11.3.6 Cautions

Machine instructions not usable in the interrupt program

(1) TMR (timer), TMRB (fixed timer), and TMRC (timer)
The least input increment for the timer values is 8 ms, so it may not be possible to measure time accurately.

(2) CTR (counter), CTRC (counter), DIFU (rising edge detection), DIFD (falling edge detection)
Because the rising or falling signal edge is handled, it is necessary to execute the interrupt program twice in order to check for signal changes. Furthermore, the DIFU and DIFD outputs are held to 1 until the interrupt program is executed again.

(3) DISPB (message display), EXIN (external data input), low–speed WINDR/WINDW (NC window data read/write), and AXCTL (PMC axis control)
It takes at least two cycles (one cycle = 8 ms) to complete the execution of each of these instructions. In addition, the execution of the EXIN, low–speed WINDR/WINDW, and AXCTL instructions involve exclusive control. If the interrupt program is terminated before completion, these instructions are disabled in the second–level sequence program.

(4) CALL (conditional subprogram call) and CALLU (unconditional subprogram call)
These instructions are unusable, in the same way as the conventional first–level program.

Continuous interrupt processing
After the interrupt program has been started by a certain interrupt signal, if another interrupt signal occurs, it is processed after the current interrupt processing is completed.

(Example) Rising edges of bits 0 and 1 of X1003

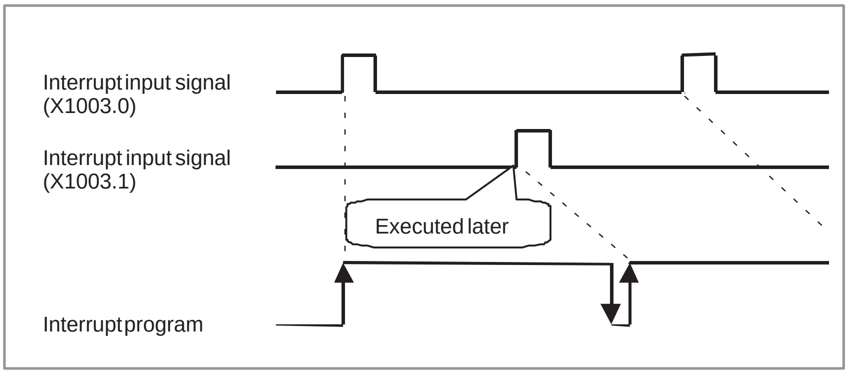


Fig. 11.3.6 (a) Continuous interrupt operations triggered by different signals

After the interrupt program has been started by a certain interrupt signal, if the same interrupt signal occurs again, it is ignored.

(Example) Rising edge of bit 0 of X1003

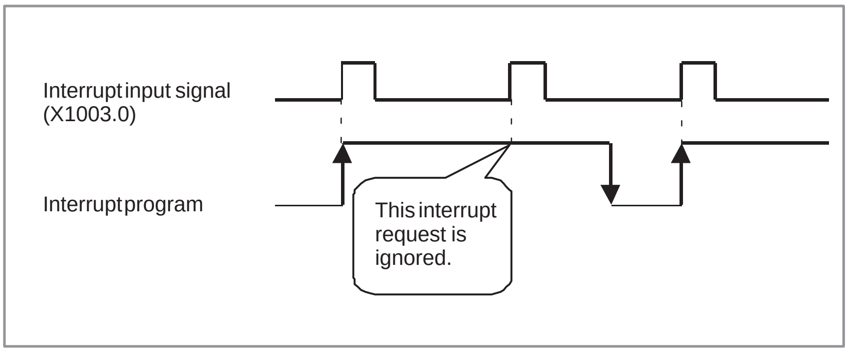
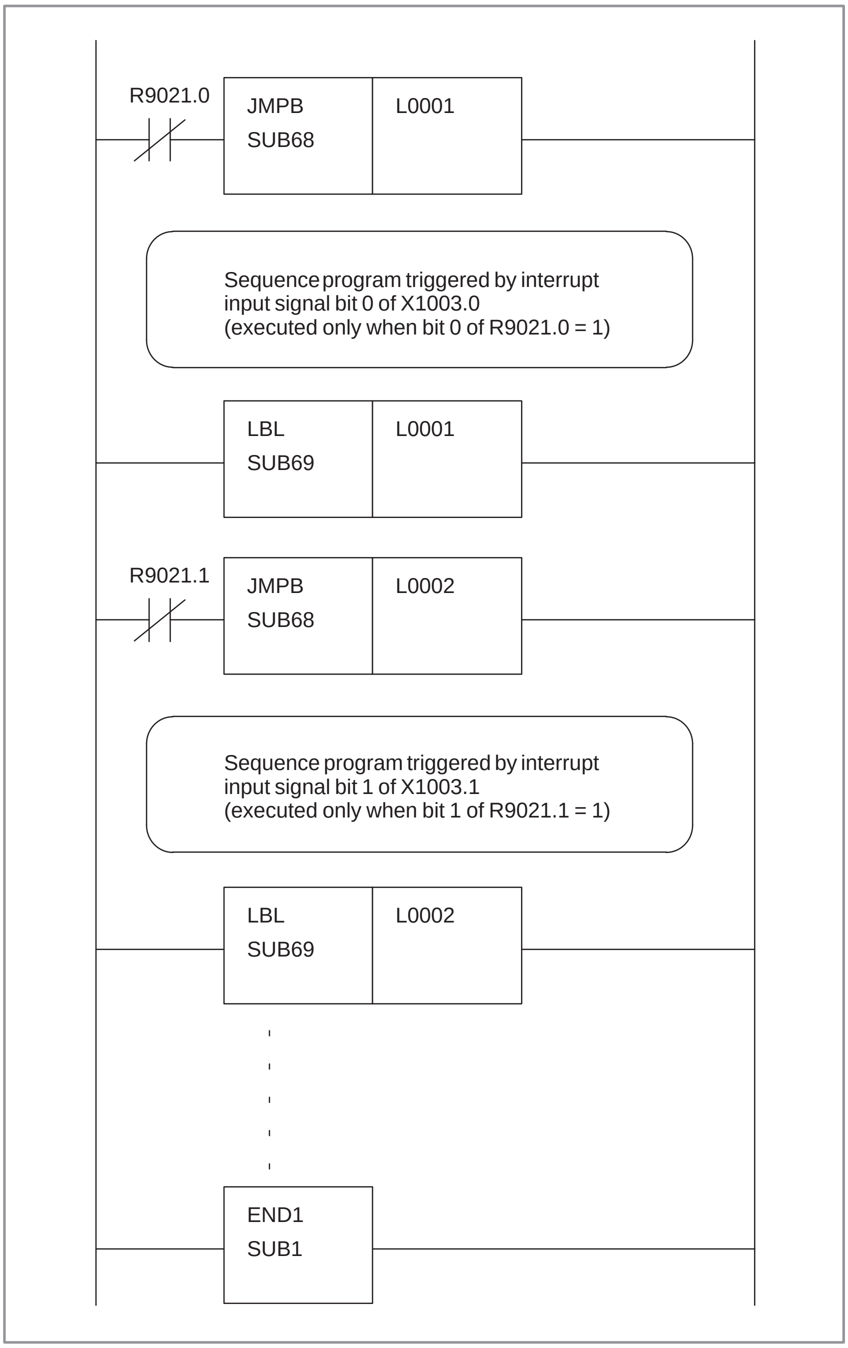


Fig. 11.3.6 (b) Continuous interrupt operations triggered by the same signal

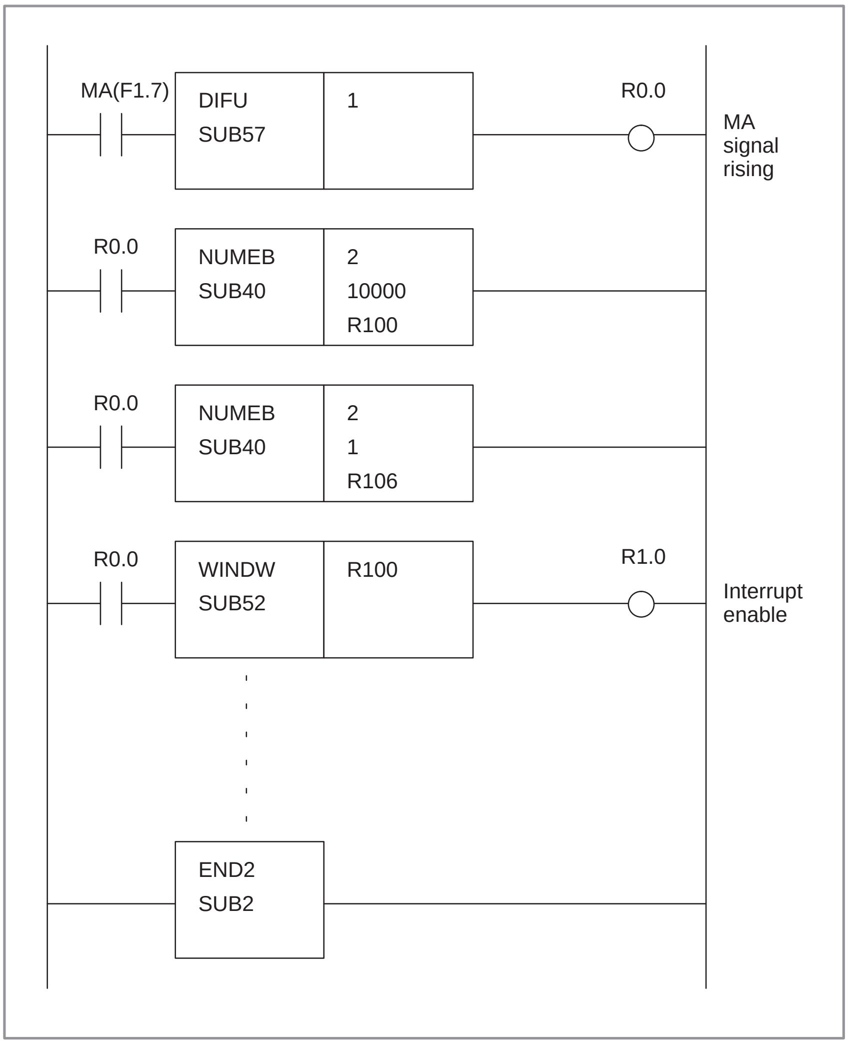
If both the rising and falling edges of a certain interrupt input signal are specified for interrupt, the falling (rising) edge is ignored if it is detected during the interrupt program execution requested on the rising (falling) edge. So, it is necessary to complete the interrupt program execution before the interrupt input signal changes.

11.4 SEQUENCE PROGRAM EXAMPLES

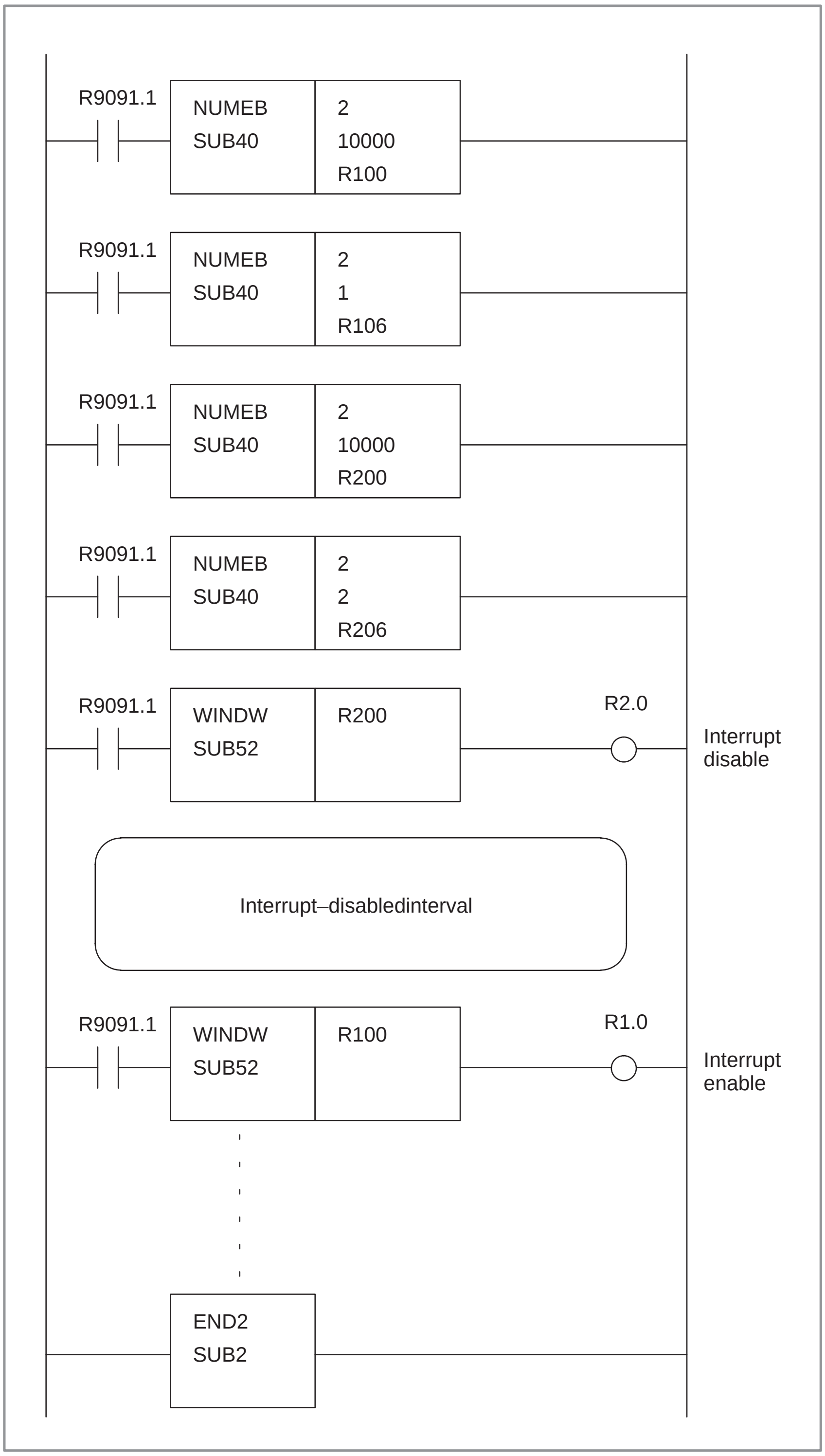
(1) Interrupt program that handles interrupt requests separately
Use of R9021 together with a label jump enables processing interrupt requests separately as shown below.



(2) Enabling interrupts immediately when the power is applied



(3) Specifying an interrupt–disabled interval



II. PMC OPERATION (CRT/MDI)

1 GENERAL

The following PMC data can be set and displayed by using the CRT/MDI panel.

1) PMC I/O signal display and internal relay display (PMCDGN)
   PMCDGN has following screens.
   a) Title data display
   b) Status screen
   c) Alarm screen
   d) Trace function
   e) Memory display
   f) Signal Wareform display function
   g) User task execution status display function
2) PMC data setting and display (PMCPRM)
   The following PMC data are provided.
   a) Timer
   b) Counter
   c) Keep relay
   d) Data table
3) Display of sequence program ladder diagram (PMCLAD)
4) PMC screen (PMCMDI) for the user
   Press the function key <CUSTOM> on the CRT/MDI panel first.

> **NOTE**
> This function key is effective when a user program exists in the PMC–RC.

Switch the NC and PMC menus as described below.
NC screen to PMC screen

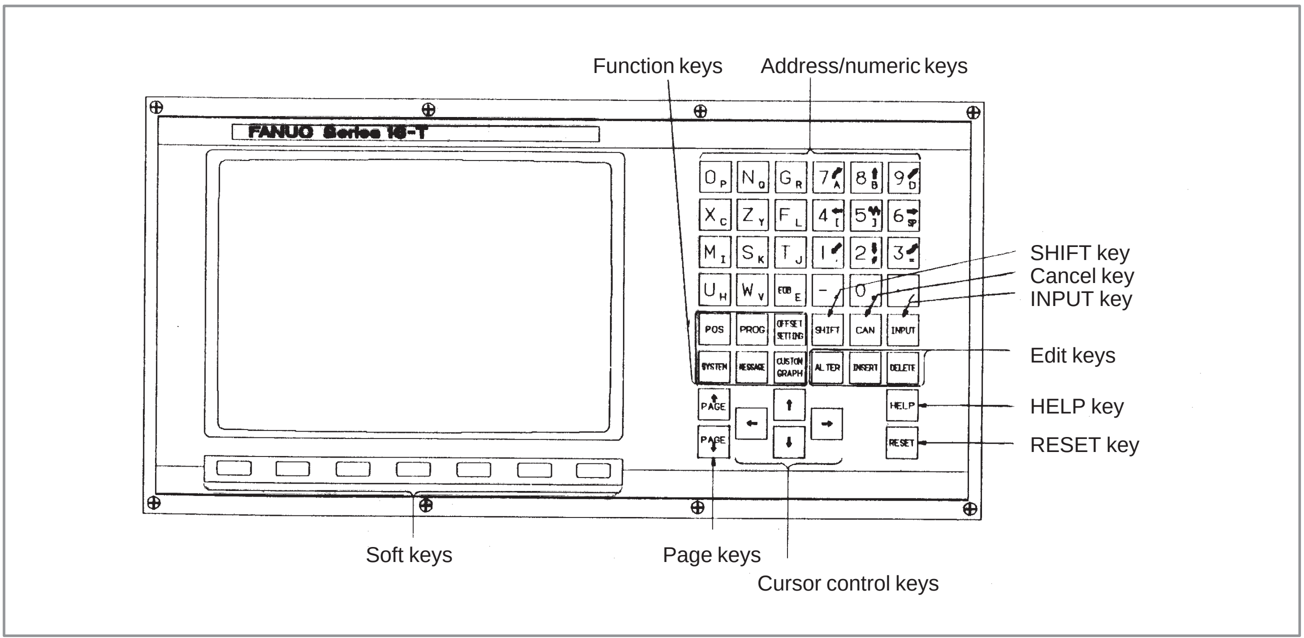
Press the SYSTEM function key on the CRT/MDI panel. Selecting the PMC soft key displays the PMC basic menu.
PMC screen to NC screen

- Pressing the RETURN key (the leftmost key) on the PMC basic menu screen changes the menu to the NC soft key menu.
- Selecting a function key on the PMC screen changes the screen to the corresponding NC screen.

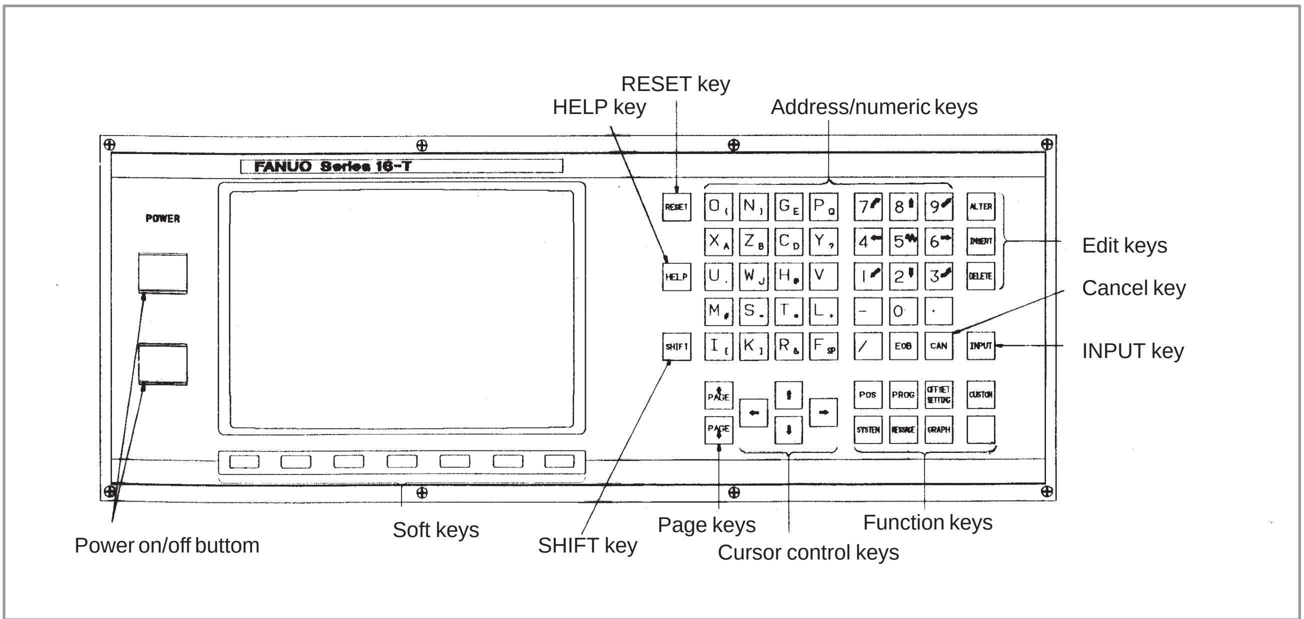
Figs. 1 l) to 1 a) show the standard CRT/MDI panels.

> **NOTE**
> A key in < > is a function key on the CRT/MDI panel. A key in [ ] is a soft key described below.

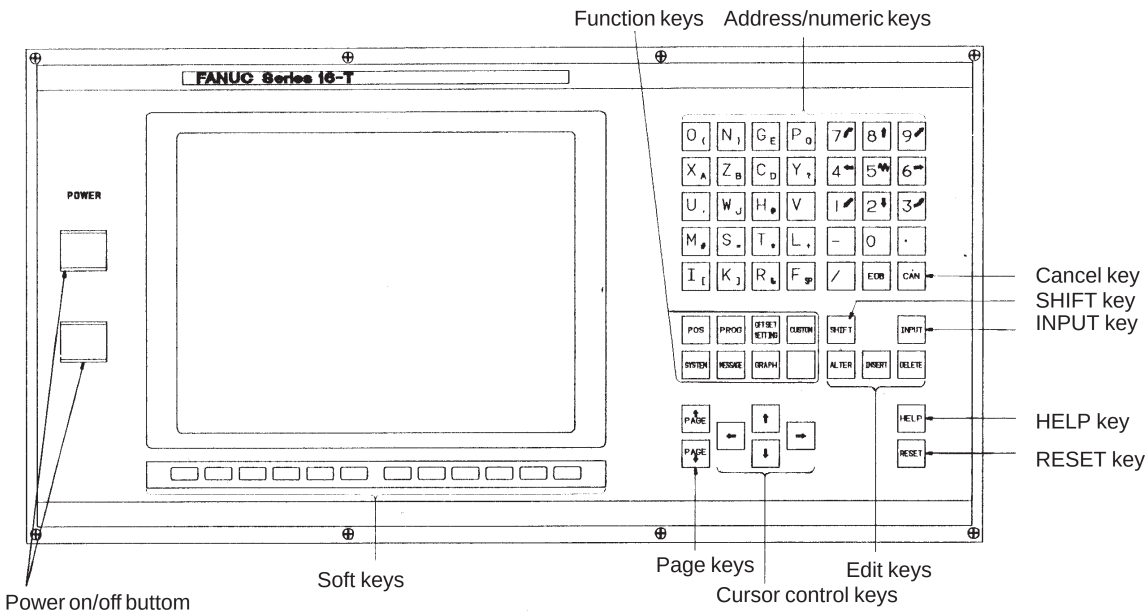
a) 9″ small monochrome/color CRT/MDI panel for 16–TA/18–TA (Horizontal type)



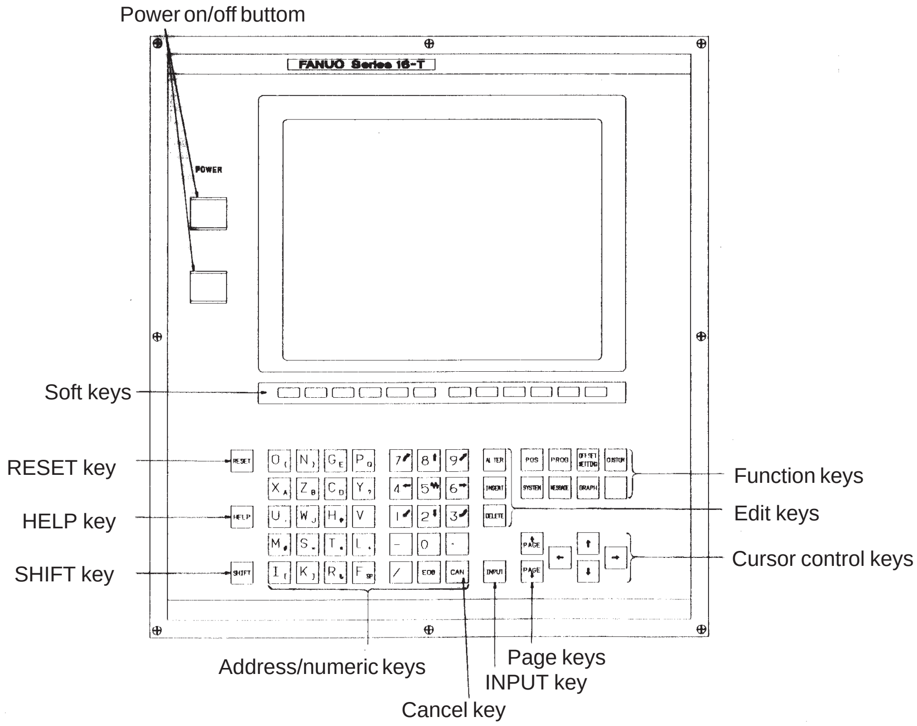
b) 9″ monochrome/color CRT/MDI panel for 16–TA/18–TA (Horizontal type)



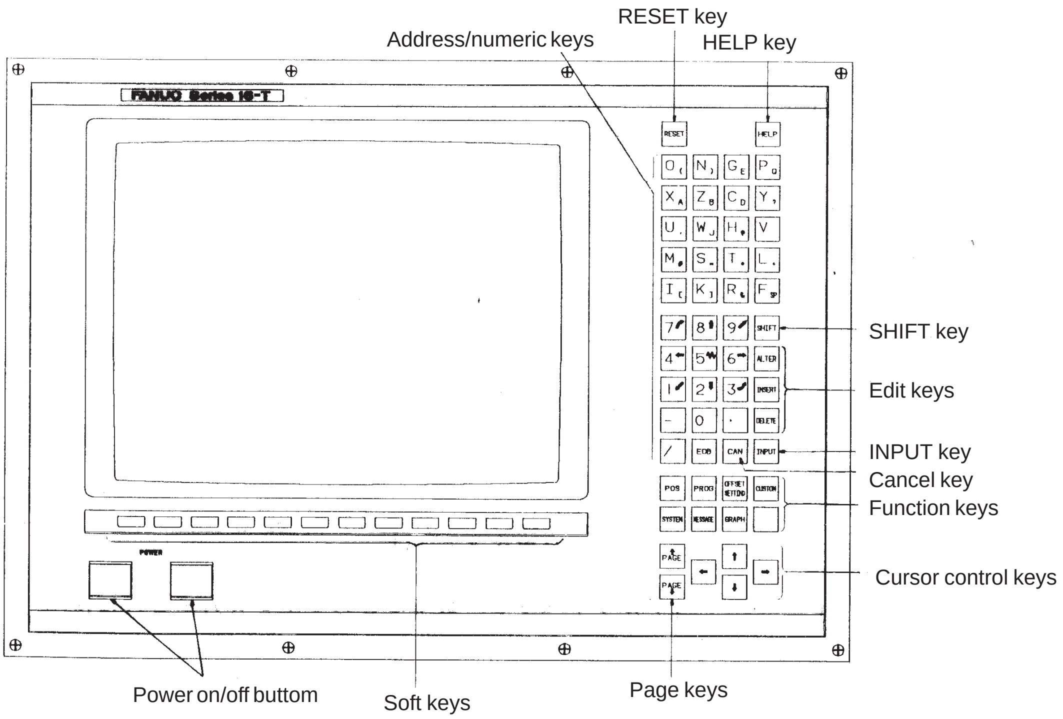
c) 10″ color LCD/MDI panel for 16–TA/18–TA (Horizontal type)



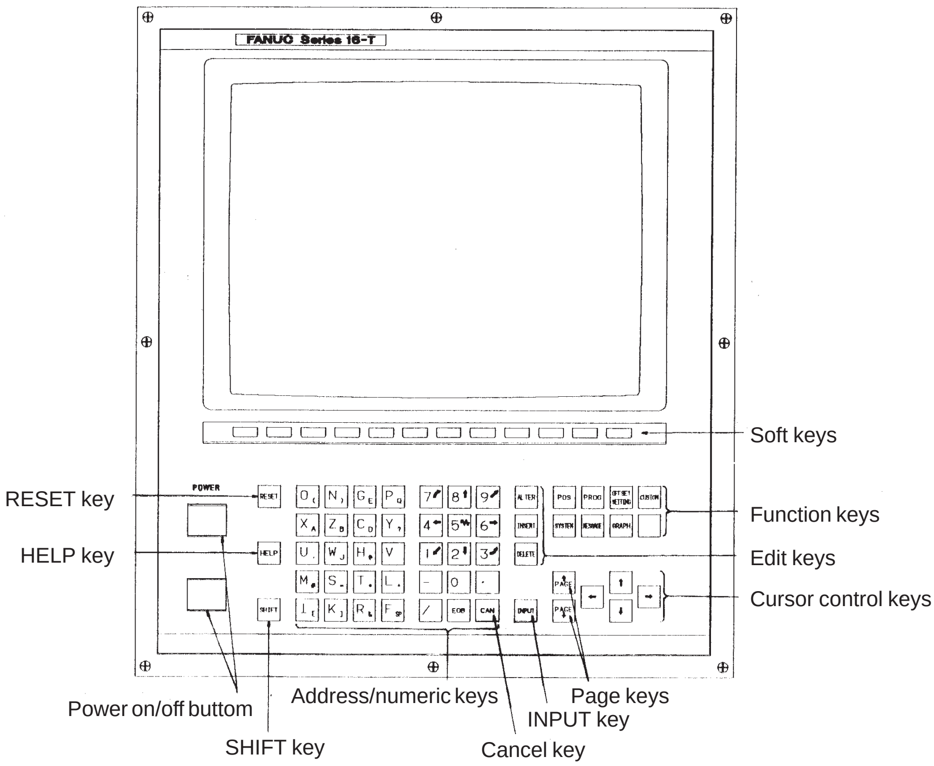
d) 10″ color LCD/MDI panel for 16–TA/18–TA (Vertical type)



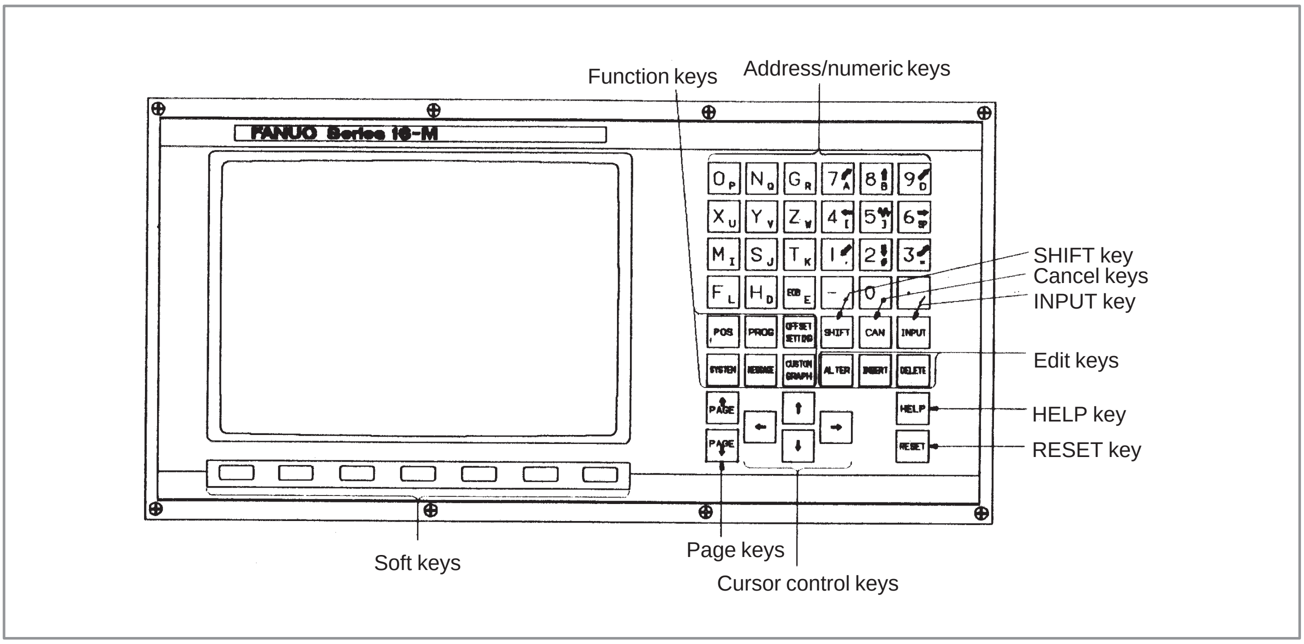
e) 14″ color CRT/MDI panel for 16–TA/18–TA (Horizontal type)



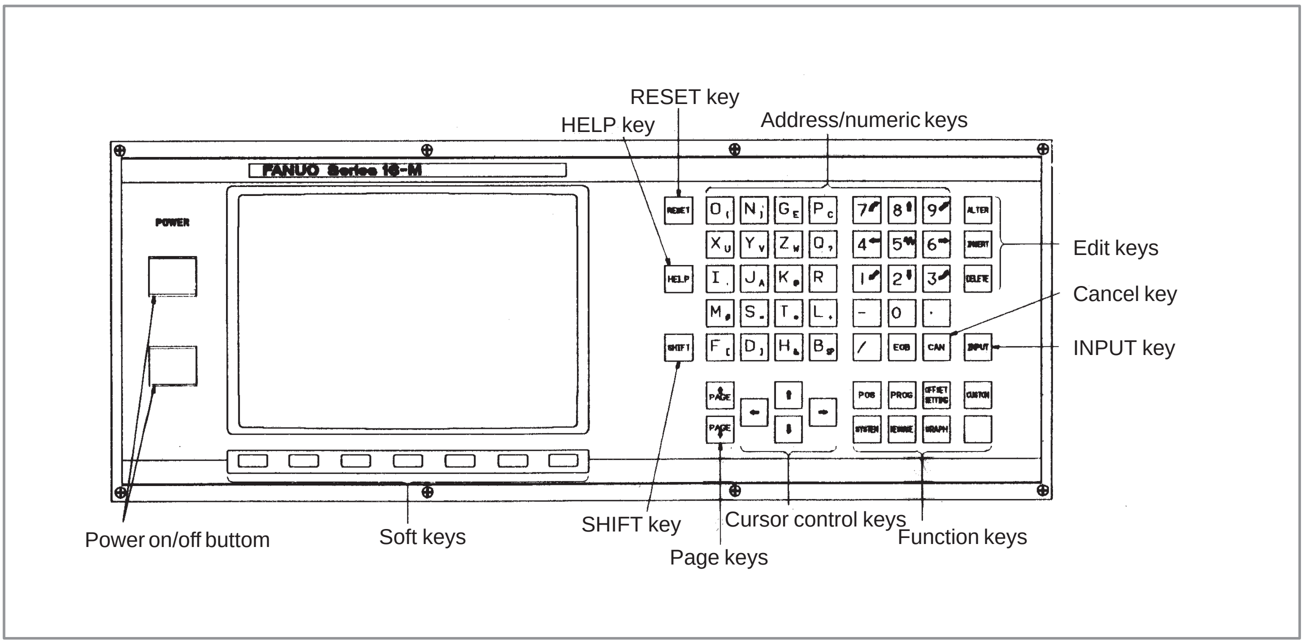
f) 14″ color CRT/MDI panel for 16–TA/18–TA (Vertical type)



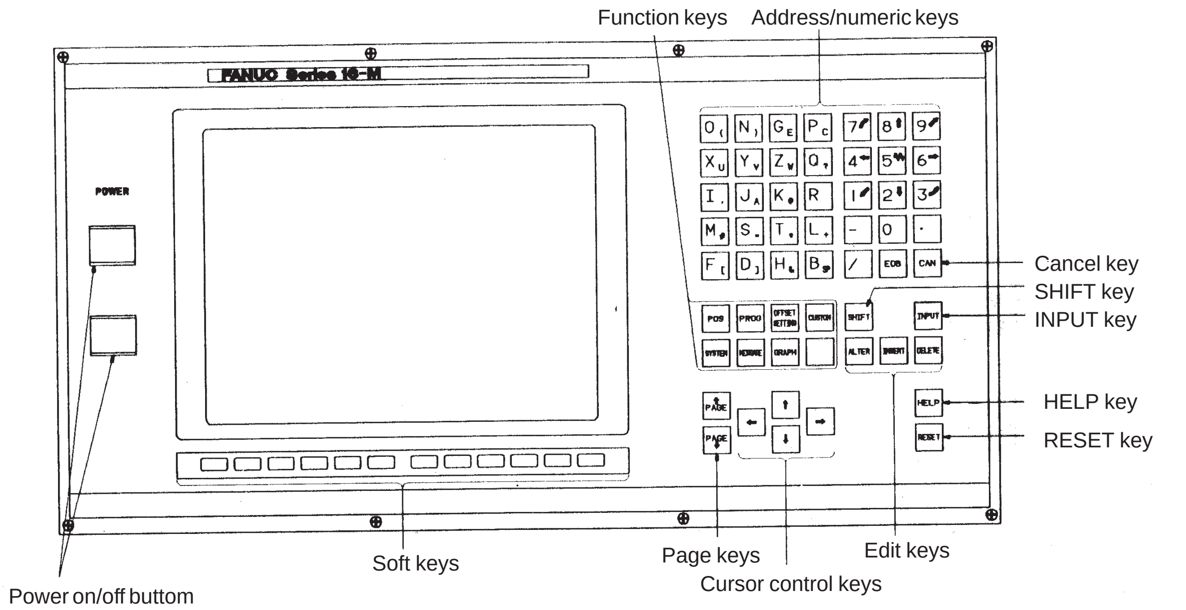
g) 9″ small monochrome/color CRT/MDI panel for 16–MA/18–MA (Horizontal type)



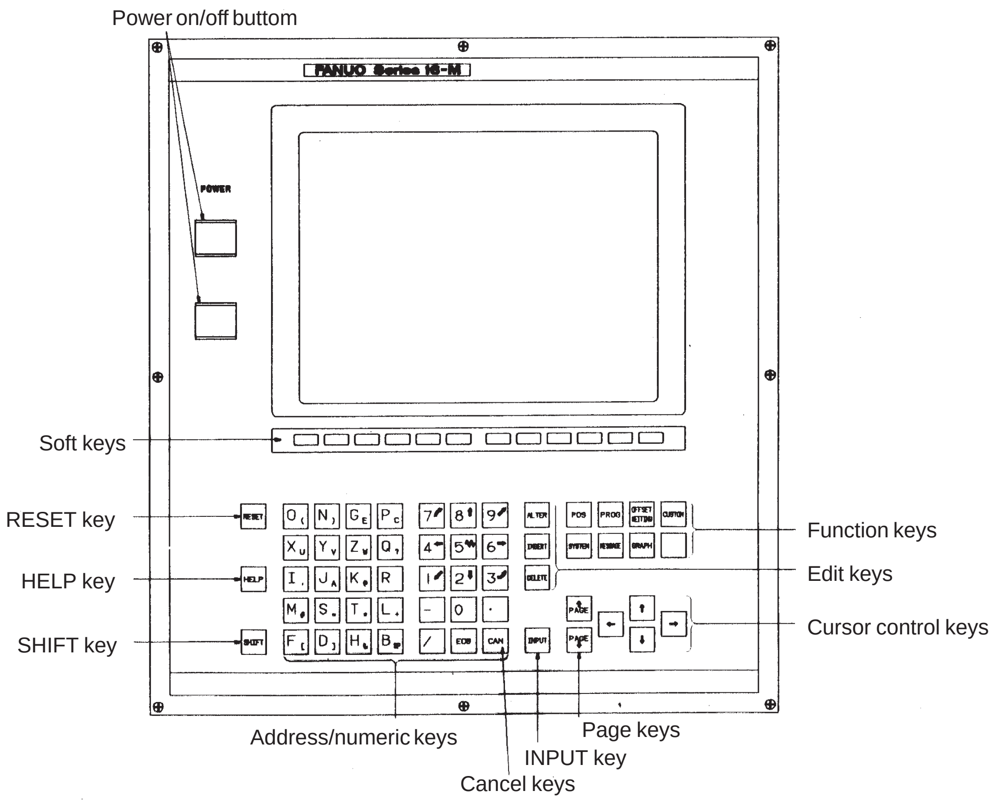
h) 9″ monochrome/color CRT/MDI panel for 16–MA/18–MA (Horizontal type)



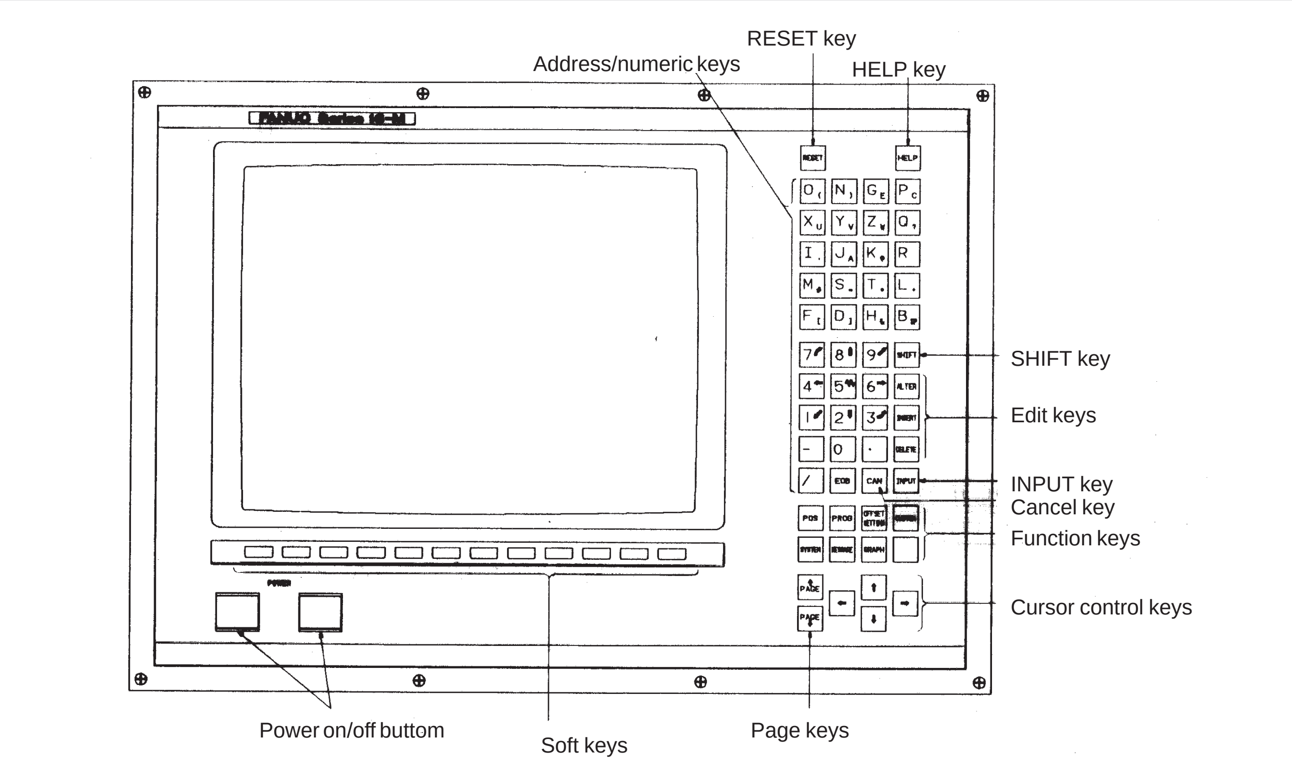
i) 10″ color LDC/MDI panel for 16–MA/18–MA (Horizontal type)



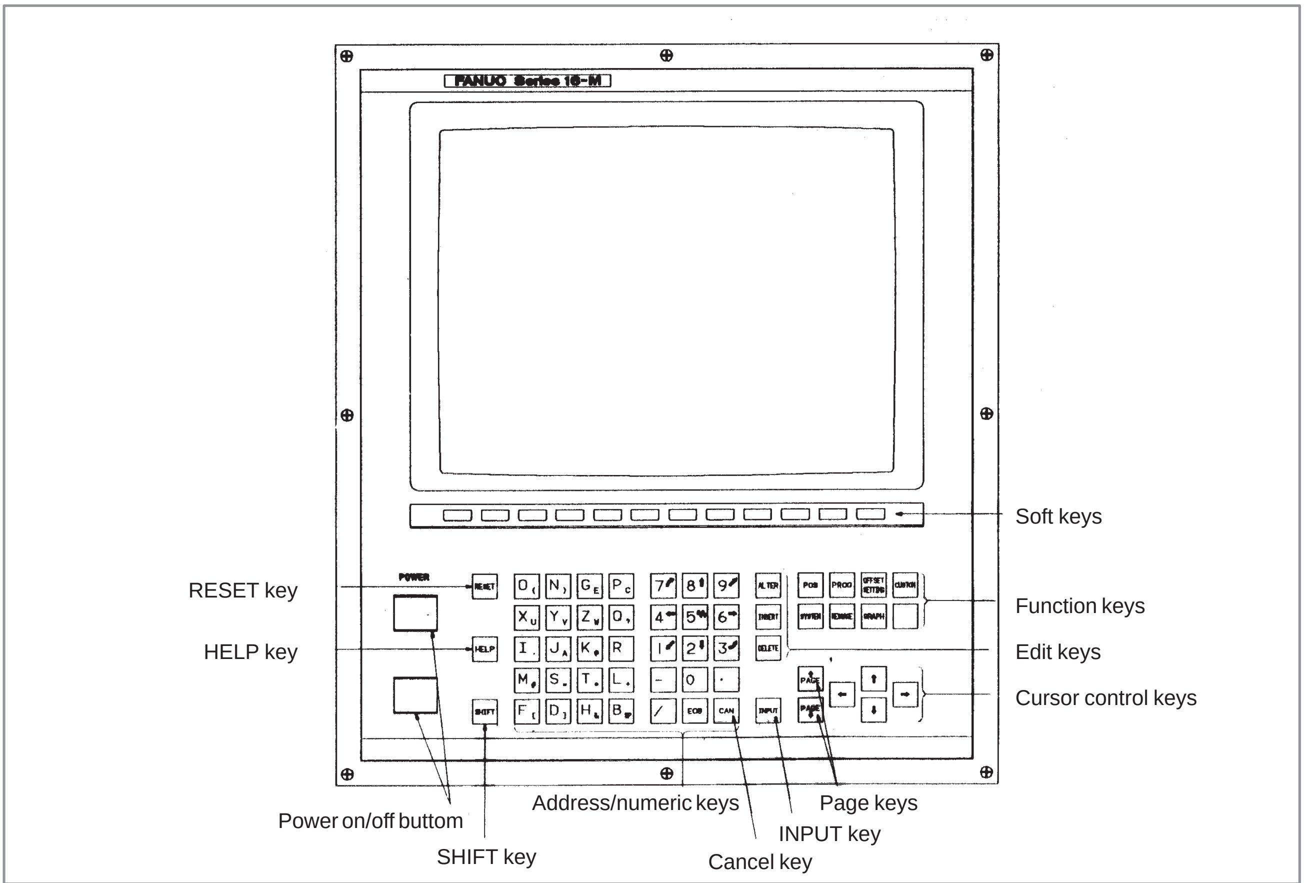
j) 10″ color LCD/MDI panel for 16–MA/18–MA (Vertical type)



k) 14″ color CRT/MDI panel for 16–MA/18–MA (Horizontal type)



l) 14″ color CRT/MDI panel for 16–MA/18–MA (Vertical type)



1.1 FOR MDI UNITS OTHER THAN STANDARD MDI UNITS (FOR FS20 PMC–SA1 AND SA3)

Note the followings when you input PMC–address on the original MDI boards made by MTBs without using Standard MDI Unit supplied by FANUC.

(1) If the MDI has the keys to input PMC–address (X, Y, F, G, R, A, C, K, D, T), You can operate as same as FANUC Seires 18 (PMC–SA1/SA3).

(2) If MDI does not have those keys, input PMC–address as follows. When inputting PMC–address (in PCLAD, STATUS and so on), you can substitute number keys (0 to 9) and a hyphen key (–) for PMC–address capital keys (X, Y, F, G, etc.). PMC–address capital keys are corresponding to the number keys as follows.

PMC–address keys	G	F	Y	X	A	R	T	K	C	D
number keys	0–	1–	2–	3–	4–	5–	6–	7–	8–	9–

(Example) If you want to input “X0.0 [SRCH] ”, input “3–0.0 [SRCH] ”.

1.2 AUTOMATIC OPERATION WHEN THE POWER IS TURNED ON

When a valid sequence program is contained in the PMC, automatic operation can be started immediately after power–on by keep relay setting. This eliminates the need to display the PMC screen and run a sequence program each time the power is turned on. The keep relay setting method depends on the PMC model. See Section 4.3.3.

1.3 CLEARING THE SEQUENCE PROGRAM

When the power for the CNC is turned on for the first time, a RAM PARITY or NMI alarm may occur in the PMC. This is caused by invalid data in the sequence program storage area in the PMC. The sequence program must be cleared to prevent this.

The automatic operation (see 1.2 above) can also be stopped by clearing the sequence program in the PMC.

The sequence program can be cleared in either of the following two ways:

1. Turn on the power while pressing X and O.
2. Turn on the power, display the PMC screen, and use the programmer function of the PMC (EDIT/CLEAR).

> **NOTE**
> In case of loader control function, turn on the power while pressing X and 5.

1.4 LOADING THE STANDARD LADDER (FOR Power Mate–D/F PMC–PA1 AND PA3)

The PMC–PA1 and PA3 contained in the Power Mate have a sequence program called the “standard ladder” in their ROM to operate the Power Mate without creating a sequence program.

Operation)

Parameter in the Power Mate

	#7	#6	#5	#4	#3	#2	#1	#0
8703								FLA

#0 (FLA) = 0 : The FANUC standard ladder is not used.
1 : The FANUC standard ladder is used.

(1) Set bit 0 (FLA) of NC parameter 8703 to 1.

This generates alarm 000 (power–off request) in the Power Mate.

(2) Turn off the power, then turn it on again.

If the PMC contains a sequence program (PMC alarm ER22 PROGRAM NOTHING does not occur), turn on the power while clearing the sequence program (pressing X and O).

(3) The FANUC standard ladder is loaded.

NOTE
If the sequence program is not cleared in the PMC, the FANUC standard ladder is not loaded. The existing sequence program remains.

1.5 FS15*i* PMC–NB6 OPERATING PROCEDURE

See Chapter 7, ”PMC–NB6 Manipulation” for an explanation of how to operate the FS15*i* PMC–NB6.

1.6 LADDER PASSWORD FUNCTION

A password can be specified for a ladder program. Specified passwords are stored as sequence program data. A ladder program for which the password has been specified cannot be displayed or edited.
Symbols, comments and messages, however, can be displayed and edited whether a password is specified or not.

(1) Applicable model

PMC–SA1/SA5/SB5/SB6 for Series 16*i*/18*i*/21*i*–A
PMC–SA1/SB3/SB4/SC3/SC4 for Series 16/18–MODEL B
PMC–SB5/SB6 for Series 16/18–MODEL C
PMC–SA1/SA3 for Series 21/210–MODEL B
PMC–NB/NB2 for Series 15–MODEL B
PMC–PA3 for Power Mate–H

(2) Types of passwords

A password consists of up to eight alphanumeric characters. The following two types of passwords are used.
Display permissible : R password (READ)
Display and editing permissible : RW password (READ+WRITE)

Table 1.6 (a) Screens requiring password release and corresponding password types

Selected screen (soft key)	Password
PMCLAD	READ
ONLEDT	READ+WRITE
M.SRCH (display)	READ
M.SRCH (input)	READ+WRITE
LADDER	READ+WRITE
CLRLAD	READ+WRITE
CLRALL	READ+WRITE
DBGLAD	READ
ONLEDT	READ+WRITE

Table 1.6 (b) Screens requiring password release and corresponding password types (DPL/MDI)

Selected screen	Password
LADDER	READ+WRITE

NOTE

1 See the following items for the selected screens listed in Table 1.6 (a).

PMCLAD	:	5. PMC LADDER DIAGRAM DISPLAY (PMCLAD) in Part II
M.SRCH	:	3.5 Display the Contents of Memory (M.SRCH) in Part II
LADDER	:	5.2 Sequence program generation (LADDER) in Part III
CLRLAD	:	5.6.2 Clear the ladder program (CLRLAD) in Part III
CLRALL	:	5.6.5 Clear the sequence program (CLRALL) in Part III
DBGLAD	:	8.4 Ladder Debug Function in Part III
ONLEDT	:	5.8 On–line Editing in Part II 8.4.2 Soft key menu for ladder debug function in Part III

2 For an explanation of the selection screen of Table 1.6 (b), see the following section:

LADDER: III 11.4 Ladder Mnemonic Editing

3 With DPL/MDI of the Power Mate, the use of the following characters only is supported for clearing passwords:

Alphabetic characters : D, F, G, K, P, T, X, Y

Numeric characters : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

If a character other than those listed above is used for a password, the password cannot be cleared using the DPL/MDI.

(3) Setting a password

Set a password for a ladder program on the editing/password screen on FAPT LADDER (for personal computers).

(4) Releasing password protection

A ladder program for which the password has been specified cannot be displayed or edited until the password is input correctly. Once password protection is released, the protection remains being released until the power is turned off then on again.

(a) When operation which requires releasing the password protection is performed, the system displays either of the following messages to require the protection to be released, depending on the type of password.

“KEY IN PASSWORD(R)” ⋯ READ PASSWORD
“KEY IN PASSWORD(R/W)” ⋯ READ+WRITE PASSWORD

(b) Enter the password and press the [INPUT] key.
*The entered password is not displayed. (Echo back is not performed.)

(c) When the password is correctly specified, the protection is released and the corresponding operation becomes available. See Table 1.6 (a). If the password is incorrectly specified, the message “FALSE PASSWORD” is displayed.

> **NOTE**
> The sequence program is cleared by turning on the power with the X and O keys being held down, whether password protection is specified or not.

(5) Special password

○ : Usable
× : See Note.
△ : Not usable

Power Mate/ FS21A	FS20/FS21B	FS18A	FS16A	FS16B FS18B		FS16C FS18C			FS21*i*	FS16*i* FS18*i*		FS15B	
PA1 PA3	RA1 RA3	SA1, SA2 SA3	SB, SB2, SB3 SC, SC3	SB3 SC3	SB4 SC4	SB5 SC3	SB6	SC4	SA1 SA5	SB5	SB6	NB	NB2
×	×	×	×	×	△	×	○	△	×	×	○	×	○

> **NOTE**
> Usable editions
> PMC–SB4 : Series 4066 Edition 08 or later
> PMC–SC4 : Series 4068 Edition 07 or later
> Edit card : Series 4073 Edition 06 or later

When a password beginning with the character # is set for RW password, the subprogram after P1500 can be edited in spite of the protection by this password.

```
LADDER <<MAIN>>    PROGRAM:(STEP SEQUENCE DEMO PROGRAM)   MONIT STOP
                    P1500 (      ) USER PROGRAM NO.1

  LEVEL1     LEVEL2    LEVEL3
  □ P0001    □ P0002   □ P0004  □ P0005  □ P0006    □ P0007
  □ P0008    □ P0009   □ P0014  □ P0015  □ P0016    □ P0017
  □ P0021    □ P0022   □ P0024  □ P0025  □ P0026    □]P0027
  .       .         .        .        .          .
  .       .         .        .        .          .
  □ P1500    □ P1501   □]P1502
```

example 1)

When the cursor is positioned to the subprogram P1500 and [ZOOM] key is pressed, this subprogram P1500 can be edited in spite of the protection by the password.

example2)

When the cursor is positioned to the subprogram P1 and [ZOOM] key is pressed, if the protection by the password is not released, the message ”KEY IN PASSWORD(R/W)” is displayed and this subprogram can be edited by inputting a correct password.

1.7 PMC OPERATION FOR LOADER CONTROL FUNCTION

Note the following when PMC of loader control function is operated.

- Operate PMC after switching to the screen for the loader control. (The control of the main and the loader changes by pushing the SHIFT key and the HELP key at the same time.)
- Connector JD5A of main board is used when communicating with RS232–C.
- When ladder data is input and output to the memory card on the PMC I/O screen or an edit card is used,the edit card or the memory card is installed at connector CNMC of the loader board.
- Connector JD1A of loader board is used when using I/O Link function.

2 PMC MENU SELECTION PROCEDURE BY SOFTKEY

Pressing the function key <SYSTEM> of CRT/MDI and the PMC soft key changes the screen to the PMC basic screen. The soft keys are displayed at the bottom of the screen.

1) PMC basic menu

If the control provides a built–in programmer function, a programmer basic menu is selected by depressing the next key. The PMC basic menu and programmer basic menu are alternately selected from each other by depressing the next key.
For programmer basic menus and operation, see Chapter III "PMC PROGRAMMER".

NOTE

1 In the following description, the relation between soft keys and menu is described based on the 9″ CRT/MDI panel. The 10″, 14″ CRT/MDI panel is provided with 10 soft keys which are those of the 9″ CRT/MDI panel, and thus, it displays many menus as compared with the 9″ CRT/MDI panel.

2 The following operations are necessary for using the built–in programmer function:

Model	Operation
PMC–SA1/SA2/SA3/SB/SB2/SB3 (FS16/18–MODEL A), PMC–SA1(FS16–MODEL A loader control)	Mount the editing module. (A02B–0120C–C160)
PMC–PA1/PA3 (Power Mate–D/H), PMC–SA1/SB3/SB4 (FS16/18–MODEL B), PMC–SB5/SB6 (FS16/18–MODEL C), PMC–SA1/SA5/SB5/SB6 (16*i*/18*i*/21*i*–MODEL A), PMC–SA1/SA3 (FS20, FS21/210–B), PMC–SA1 (FS16–MODEL B/C, 16*i*/18*i*/21*i*–MODEL A, FS21–B loadercontrol function)	Mount an editing card.
PMC–SC/SC3(FS16/18–MODELA), PMC–SC3/SC4(FS16/18–MODELB), PMC–NB/NB2(FS15B)	The function is already contained.
Common to all the models listed above	Set bit 1 of K17 to 1.

The FS18–MODEL A contains the PMC–SA1,SA2,or SA3. The series number is 4070.
The FS20 contains the PMC–SA1 or SA3. The series number is 4080.
The FS21/210–B contains PMC–SA1 or SA3. The series number is 4084.
The FS21–B(with loader control) contains PMC–SA1. The series number is 4086.

Built–in programmer function

```
PMC DIAGNOSIS FUNCTIONS          MONIT RUN

 SELECT ONE OF FOLLOWING SOFT KEYS

   PMCLAD  : DYNAMIC LADDER DISPLAY
   PMCDGN  : DIAGNOSIS FUNCTION
   PMCPRM  : PARAMETER(T/C/K/D)
   RUN/STOP: RUN/STOP SEQUENCE PROGRAM
   EDIT    : EDIT SEQUENCE PROGRAM
   I/O     : I/O SEQUENCE PROGRAM
   SYSPRM  : SYSTEM PARAMETER
   MONIT   : PMC MONITOR

[PMCLAD] [PMCDGN] [PMCPRM] [       ] [       ]
```

Fig. 2 PMC basic menu screen (9″CRT)

> **NOTE**
> Without built–in programmer function of PMC–SA1, –SA2, –SA3, –SB, –SB2, –SB3, –SB4, –SB5, or –SB6 there are only RUN/STOP and I/O functions.

2) Keys on CRT/MDI panel

The following keys are related to PMC operation on CRT/MDI panel.

a) <SYSTEM> key

Selects from CNC menu to PMC basic menu.

b) <PAGE↑> key

Screen page return key.

c) <PAGE↓> key

Screen page advance key.

d) <↑> key

Cursor shift (upward) key.

e) <↓> key

Cursor shift (downward) key.

f) <←> key

Cursor shift (leftward) key. Search function with this key is provided in PMCLAD EDIT, LADDER (See chapter II.5 and Chapter III.5.2.5 for details).

g) <→> key

Cursor shift (rightward) key. Search function with this key is provided in PMCLAD EDIT, LADDER (See chapter II.5 and Chapter III.5.2.5 for details).

h) Soft key

These keys show operating functions corresponding to individual operations when various PMC operations are done. The soft key functions change (key menus are selected) according to operations.

i) Next key

This key is used for extending menus of soft keys. By pressing this key, a menu changes, and by pressing it again, the menu is reset as before.

j) Return key

Various PMC operations are conducted by pressing soft keys related to menus.
The menus sequentially change when pressing corresponding soft key. Use this return key to reset a menu to the original one.

3) Status display

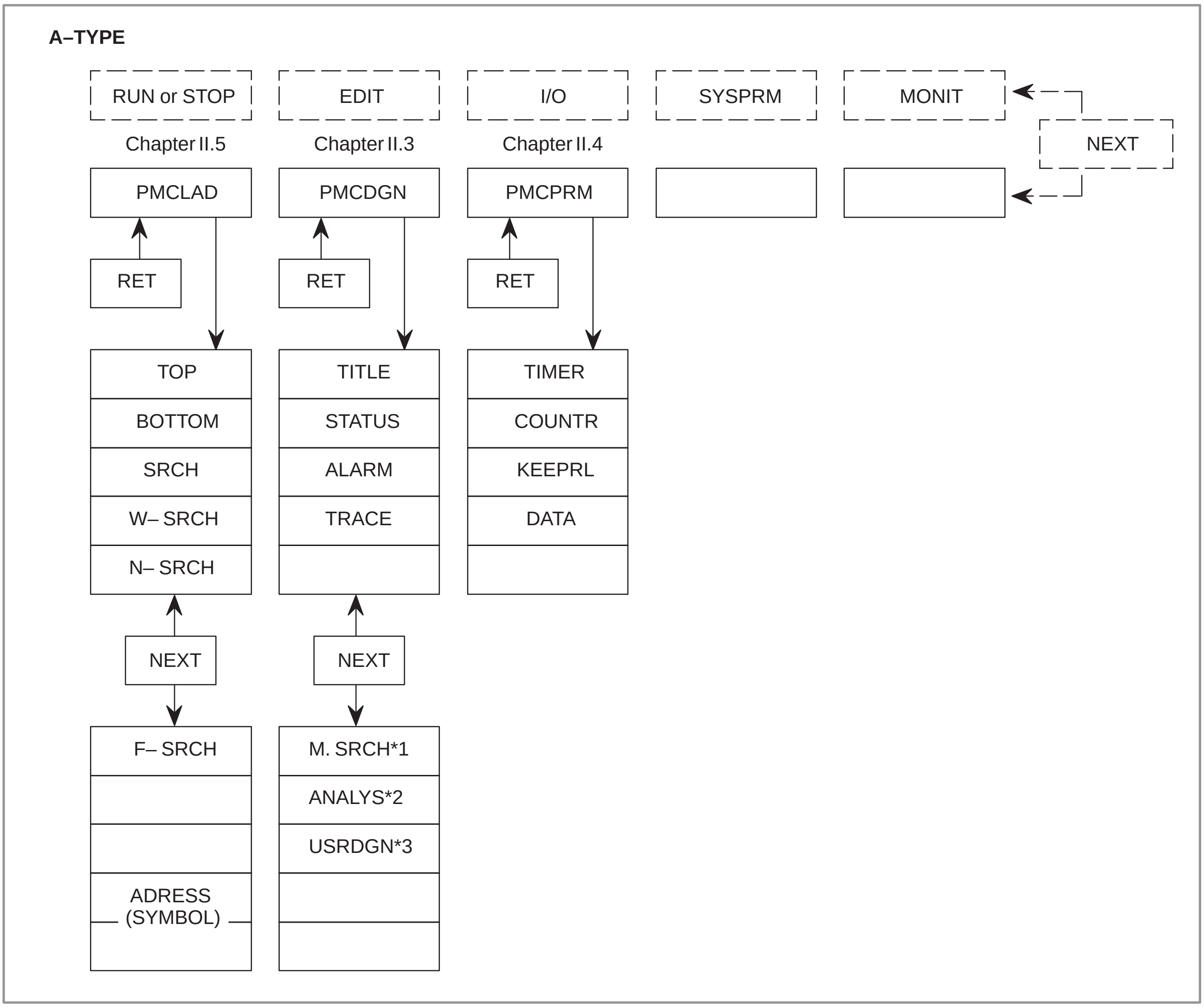
The alarm status and the name of the sequence program storage that is currently effective are displayed on all the PMC menus.
In addition, PMC–SC/SC3/SC4/NB display the states while the debugging function is used.

```
Data entry        →   >
Status display    →                                   DBG  –RAM–  ALM
Soft key display      [      ] [      ] [      ] [      ] [      ]
```

- ALM : An alarm occurred in the PMC (For details, see Section 3.3.)
- RAM : The currently effective sequence program storage is a RAM module.
- ROM : The currently effective sequence program storage is a ROM module.
- EPROM: Currently effective sequence program storage is EPROM. (EPROM for PMC–SA1, PMC–SA2, PMC–SB, and PMC–SB2)
- DBG : A break issued by the debugging function of PMC–SC/SC3/SC4/NB in effective.
- BRK : The break issued by the debugging function of PMC–SC/SC3/SC4/NB has terminated.

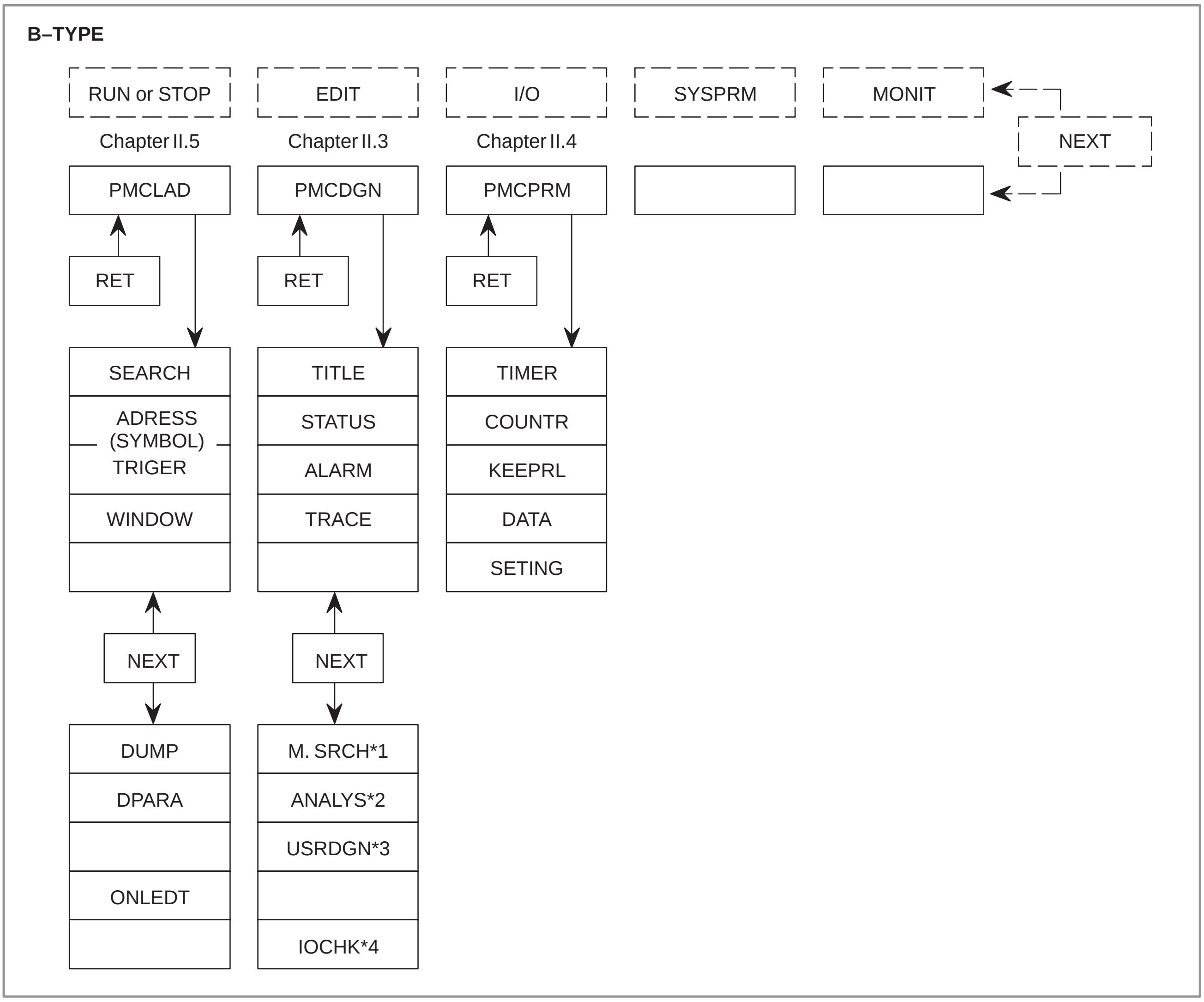
4) Relation between PMC menus and soft keys

There are 2 types, A and B, in the series of CNC.



NOTE

The soft keys indicated by *1, *2, *3 are supported only for certain models. See the conditions in the description of each relevant function.



NOTE
The soft keys indicated by *1, *2, *3 are supported only for certain models. See the conditions in the description of each relevant function.

The softkey' s type for the series of CNC.

CNC type	Power Mate		FS20		FS18			FS16					FS15B	
PMC type	PA1	PA3	SA1	SA3	SA1	SA2	SA3	SB	SB2	SB3	SC	SC3	NB	NB2
Softkey type	A	A	A	A	AB	A	B	AB	A	B	AB	AB	B	B

Type A or B is selected depending on the Series of PMC control software.

Series of PMC control software and type of softkey are related as follows.

	Type A	Type B
FS16A	4061	4063
FS18A	4070	4071

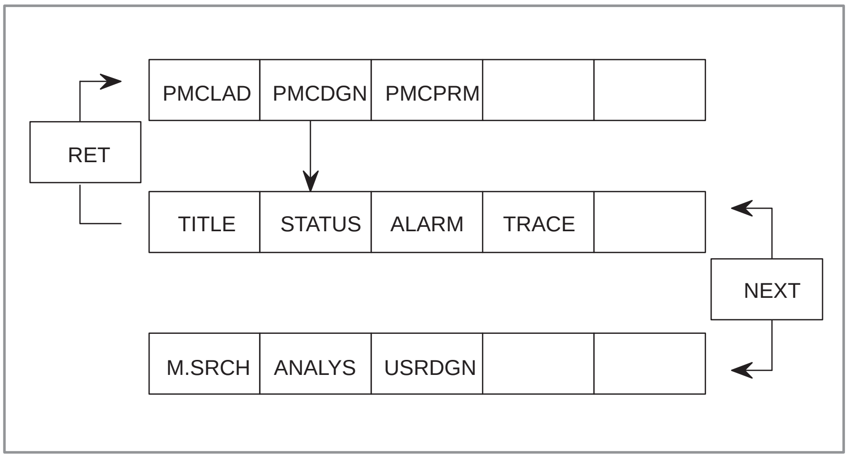
(Reference) Series of PMC control software is displayed on the [PMCDGN] and [TITLE] screen as shown below.

```
PMC CONTROL PROGRAM
SERIES : 4063 EDITION : 08
```

Series of PMC control software

3 PMC I/O SIGNAL DISPLAY AND INTERNAL RELAY DISPLAY (PMCDGN)

PMC I/O signals, internal relays, and other PMC diagnosis are displayed on the screen by depressing soft key [PMCDGN].



3.1 DISPLAYING TITLE DATA

Title Data refers to the title of the sequence program created by the machine tool builder.

They consist of the following ten items :

Machine tool builder name	(32 characters)
Machine tool name	(32 characters)
NC and PMC types	(32 characters)
Sequence program number	(4 characters)
Version	(2 characters)
Sequence program drawing number	(32 characters)
Date when the sequence program was created	(16 characters)
Sequence program programmer	(32 characters)
ROM programmer	(32 characters)
Comment	(32 characters)

In addition to the title display :

1) Series and version of the PMC control software.
2) Type of the PMC.
3) For Editing module or Editing card, the series and version.
4) Memory areas used for each sequence data, and execution time of ladder program.
5) Type of PMC control module and PMC sequence program.
6) For the non-dividing system, the present, maximum and minimum values for the execution time of ladder program.

NOTE
When a C board is installed in the Series 16*i*/18*i*, the title data for C can be displayed. With the arrow keys [←] and [→], the user can switch the display between the ladder title and C title data.

To display the previous or next screen on the 9″ CRT/MDI, use the <PAGE ↑> or <PAGE ↓> key.

```
PMC TITLE DATA #1                  MONIT  RUN
 PMC PROGRAM NO.  : 1234
 EDITION NO.      : 12
 PMC CONTROL PROGRAM
     SERIES : 4063 EDITION : 08
    (SERIES : 4065 EDITION : 08)
 PMC TYPE CONTROL : RB3  PROGRAM : RB3
     MEMORY USED  : 007.8 KB
     LADDER       : 007.0 KB
     SYMBOL       : 000.0 KB
     MESSAGE      : 000.8 KB
     SCAN TIME    : 008   MS
     SCAN MAX     : 016   MS  MIN : 008  MS

[TITLE ] [STATUS ] [ALARM ] [TRACE ] [      ]
```

Fig. 3.1 (a) Title data 1

```
PMC TITLE DATA #2                                   MONIT RUN

 MACHINE TOOL BUILDER NAME :
 ○ · · · · · · · · · · · · · ○

 MACHINE TOOL NAME :
 ○ · · · · · · · · · · · · · ○

 CNC & PMC TYPE NAME :
 ○ · · · · · · · · · · · · · ○

 PROGRAM DRAWING NO. :
 ○ · · · · · · · · · · · · · ○


[TITLE ] [STATUS ] [ALARM ] [TRACE ] [       ]
```

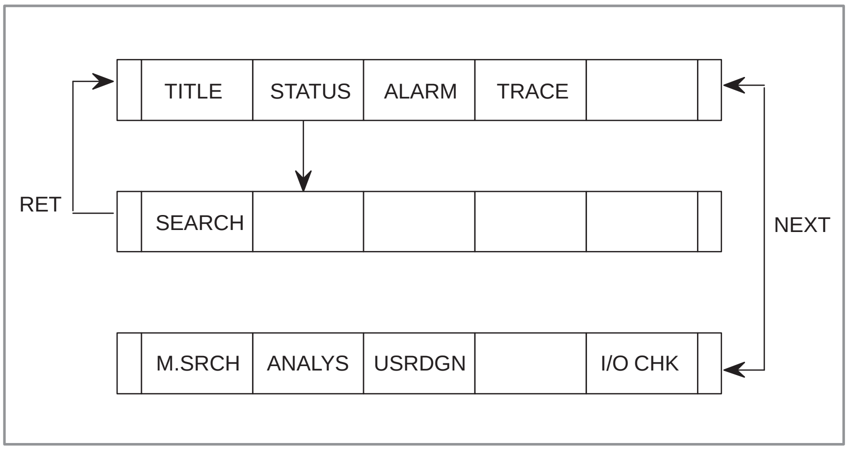
Fig. 3.1 (b) Title data 2

```
PMC TITLE DATA #3                                   MONIT RUN

 DATE OF PROGRAMING :
 ○ · · · · · · ○

 PROGRAM DESIGNED BY :
 ○ · · · · · · · · · · · · · ○

 ROM WRITTEN BY :
 ○ · · · · · · · · · · · · · ○

 REMARKS :
 ○ · · · · · · · · · · · · · ○


[TITLE ] [STATUS ] [ALARM ] [TRACE ] [       ]
```

Fig. 3.1 (c) Title data 3

3.2 DISPLAY OF SIGNAL STATUS (STATUS)

The contents at all addresses (X, Y, F, G, R, A, C, K, D, T, M, N) disignated in programs can be displayed on the CRT screen. This display is all done by “0” and “1” bit patterns, and symbol data is displayed together at address bits where symbol data are difined.



1 Depress [STATUS] soft key. The CRT screen changes as shown in Fig. 3.2, and the soft key menu is changed.

2 Depress [SEARCH] key after keying in an address to be displayed.

3 A continuous 8 byte data is displayed by a bit pattern from the designated address in the top stage of the CRT screen.

4 Depress [SEARCH] key or page key to display another address.

```
 PMC SIGNAL STATUS                           MONIT RUN

ADDRESS    7     6     5     4     3     2     1    0
EXDAT1  ED7   ED6   ED5   ED4   ED3   ED2   ED1  ED0
G0000      0     0     0     0     0     0     0    0
EXDAT2 ED15  ED14  ED13  ED12  ED11  ED10  ED9  ED8
G0001      0     0     0     0     0     0     0    0
       ESTB   EA6   EA5   EA4   EA3   EA2   EA1  EA0
G0002      0     0     0     0     0     0     0    0
       ERDRQ  EOREND
G0003      0     0     0     0     0     0     0    0
       MFIN5 MFIN4 MFIN3 MFIN2 MFIN1
G0004      0     0     0     0     0     0     0    0
        BFIN  AFL               TFIN  SFIN EFIND MFIN
G0005      0     0     0     0     0     0     0    0
        DLK               OVC       *ABSM  BRN  SRN
G0004      0     0     0     0     0     0     0    0
       RLSOT EXLM2 *FLWP              ST      STLK RVS
G0007      0     0     0     0     0     0     0    0

[SEARCH] [      ] [      ] [      ] [      ] [      ]
```

Fig. 3.2 Status display of PMC I/O signals and internal relays

3.3 ALARM SCREEN (ALARM)

If an alarm is issued in the PMC, pressing the PMC soft key displays the alarm message as shown in Fig. 3.3. ALM blinks at the lower right corner of the screen.
If a fatal error occurs, a sequence program does not start.

```
PMC ALARM MESSAGE                          MONIT STOP

 ER00 PROGRAM DATA ERROR (ROM)









                                               ALM
 [TITLE ] [STATUS] [ALARM ] [TRACE ] [        ]
```

Fig. 3.3 Alarm screen

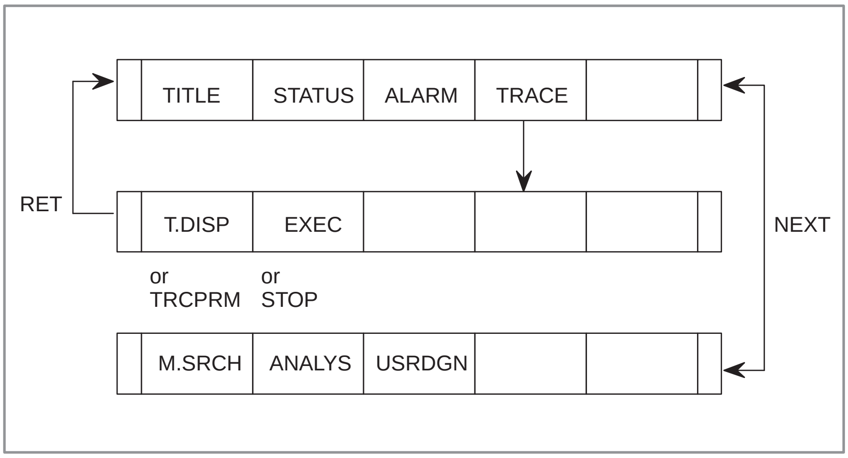
For displayed messages, see the appendix, “Alarm Message List.”

3.4 TRACE FUNCTION (TRACE)

This function checks the signal history which cannot be checked in the status display. Using one- or two-byte addressing, the function records a state when the signal changes. In two-byte addressing, discontinuous addresses can be set.

3.4.1 Operation

Pressing the [TRACE] key on the PMCDGN screen displays the trace screen when signals are being read. When signals are not being read, the parameter setting screen for reading signals is displayed. After displaying either screen, pressing the [TRCPRM] key on the trace screen displays the parameter setting screen and pressing the [T.DISP] key on the parameter setting screen displays the trace screen.



3.4.2 Parameter Setting Screen

Data to be used for reading signals needs to be specified to check the signal history.

1) Parameters

TRACE MODE : Sets a mode used for reading signals
- 0 : 1-byte data
- 1 : 2-byte data (discontinuous addresses can be specified)
- 2 : Word data (with continuous addresses)

ADDRESS TYPE : Sets addresses used
- 0 : PMC address
- 1 : Physical address

ADDRESS : Sets addresses at which a signal is traced

MASK DATA : Sets a masked bit or bits (signals can be read with unnecessary bits masked)
Range : 00 to FF

The above trace parameters are retained if the power is turned off.

3.4.3 Starting or Stopping the Trace Function

EXEC : Starts reading signals

NOTE

1 Pressing the [EXEC] key again clears the results of the previous trace.
If the trace parameters are not set correctly, the trace is not performed.
When signals are being sampled using the function for displaying signal waveforms, the trace is not performed.
2 The result data of the trace is stored latest 256-byte. If the power is turned off, the results of the trace are cleared.
3 Signals R9000 to R9007 cannot be traced.
4 A signal is traced at intervals of 8 ms. If the signal changes within 8 ms, the changed signal state cannot be traced.
5 When the trace address type is specified as a physical address, specify an effective memory address. If an ineffective address is specified to execute the trace, a system error may occur.

STOP : Stops reading signals.

```
PMC SIGNAL TRACE                             MONIT RUN

 TRACE MODE         : 1
(0:1BYTE/1:2BYTE/2:WORD)
 1ST TRACE ADDRESS CONDITION
     ADDRESS TYPE   : 1 (0:PMC /1:PHY)
     ADDRESS        : FFE480
     MASK DATA      : 11
 2ND TRACE ADDRESS CONDITION
     ADDRESS TYPE   : 0 (0:PMC /1:PHY)
     ADDRESS        : Y0
     MASK DATA      : FF



[T.DISP] [ EXEC ] [      ] [      ] [      ]
```

Fig. 3.4.3 Trace parameter setting screen

3.4.4 Trace Screen

Signal history can be checked using data specified on the parameter setting screen. The result of the latest trace is displayed at the cursor position. The cursor moves on the screen as the results of the trace are obtained. If the cursor moves off the screen, the results of the trace can be followed by pressing the page key to display the subsequent screen.

```
PMC SIGNAL TRACE                                    MONIT RUN

      1ST ADDRESS = X0000 (FF)     2ND ADDRESS = Y0000 (FF)
 NO.   7  6  5  4    3  2  1  0    7  6  5  4    3  2  1  0
 0000  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0001  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0002  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0003  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0004  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0005  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0006  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0007  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0008  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0009  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0010  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0011  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0012  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0013  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0014  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
 0015  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .

[TRCPRM ] [ STOP  ] [         ] [         ] [         ]
```

EXEC

Fig. 3.4.4 Trace screen

3.4.5 Automatic Tracing Function at Power on

Trace operation can be started automatically, immediately after power–on, by setting trace parameters beforehand and by setting the keep relay to start the trace function automatically upon power–on. The keep relay setting method depends on the PMC model. See Section 4.3.3.

3.5 DISPLAYING THE CONTENTS OF MEMORY

◎: Standard
○ : optional
× : cannot be used
△ : Can be used for the 4084 series.

Power Mate–D/F		Power Mate–H	FS20/ FS21A		FS21B		FS18A			FS18B	FS16A		FS16A/B FS18B		FS16C FS18C		FS21*i* FS16*i* FS18*i*				FS16A	FS16A /B/C FS18B/C	FS16B/C FS18B/C	FS15B	
PA1	PA3	PA3	SA1	SA3	SA1	SA3	SA1	SA2	SA3	SA1	SB	SB2	SB3	SB4	SB5	SB6	SA1	SA5	SB5	SB6	SC	SC3	SC4	NB	NB2
×	×	◎	×	×	△	△	○	○	○	○	○	○	○	○	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎

- The ladder editing module is required for the PMC–SB/SB2/SB3 of the Series 16 –MODEL A and for the PMC–SA1/SA2/SA3 of the Series 18–MODEL A.
- This function is provided as a standard function with PMC–SA1/SB3/SB4/SB5/SB6/SC3/SC4 of the Series 16/18–MODEL B/C, and PMC–SA1/SA3 of the Series 21/210–MB.

3.5.1 Operation

1) Pressing the [M.SRCH] soft key changes the screen to that shown in Fig. 3.5. The displayed soft keys also change.
2) Enter a physical address in hexadecimal from which the contents of the memory are to be displayed. Then pressing the [SEARCH] key displays 256 bytes of stored data starting from the specified address.

   Example) Entering 100000, then pressing the [SEARCH] key displays the contents of the memory starting from 100000H.
3) An address can be changed using the <PAGE ↓> or <PAGE ↑> key.
4) Pressing either the [BYTE], [WORD], or [D.WORD] soft key displays data of the corresponding type.

CAUTION
If an address at which the memory is not used is specified, a system error occurs. Be sure to specify the correct address.

3.5.2 Function for Storing Data in Memory

To store data in memory, set bit 4 of keep relay K17 to 1, move the cursor to a position at which the address of the data to be changed in RAM is displayed, and enter data in units of data type in hexadecimal.

Example) Entering 0F41, then pressing the [INPUT] key stores 0F41 at the address specified by the cursor.

CAUTION
Some values cause a system error.

```
PMC CONTENTS OF MEMORY                                        MONIT RUN

 100000  0000 0000 0000 0000 0000 0000 0000 0000  ................
 100010  4142 4344 4546 4748 494A 4B4C 4D4E 4F50  ABCDEFGHIJKLMNOP
 100020  2020 2020 2020 2020 2020 2020 2020 2020
 100030  5152 5354 5556 5758 595A 2020 2020 2020  QRSTUVWXYZ

 100040  0000 0000 0000 0000 0000 0000 0000 0000  ................
 100050  0000 0000 0000 0000 0000 0000 0000 0000  ................
 100060  0000 0000 0000 0000 0000 0000 0000 0000  ................
 100070  0000 0000 0000 0000 0000 0000 0000 0000  ................
 100080  4641 4E55 4320 434F 2E2C 5444 0000 0000  FANUC CO.LTD....
 100090  0000 0000 0000 0000 0000 0000 0000 0000  ................
 1000A0  0000 0000 0000 0000 0000 0000 0000 0000  ................
 1000B0  0000 0000 0000 0000 0000 0000 0000 0000  ................

 1000C0  0000 0000 0000 0000 0000 0000 0000 0000  ................
 1000D0  0000 0000 0000 0000 0000 0000 0000 0000  ................
 1000E0  0000 0000 0000 0000 0000 0000 0000 0000  ................
 1000F0  0000 0000 0000 0000 0000 0000 0000 0000

>

 [  SEARCH  ] [  INPUT   ] [          ] [          ] [          ]
```

Fig. 3.5.2 Memory display

3.6 FUNCTION FOR DISPLAYING SIGNAL WAVEFORMS (ANALYS)

◎: Standard
○ : optional
×: cannot be used
△ : Can be used for the 4084 series.

Power Mate–D/F		Power Mate–H	FS20/ FS21A		FS21B		FS18A			FS18B	FS16A		FS16A/B FS18B		FS16C FS18C		FS21*i* FS16*i* FS18*i*				FS16A	FS16A /B/C FS18B/C	FS16B/C FS18B/C	FS15B	
PA1	PA3	PA3	SA1	SA3	SA1	SA3	SA1	SA2	SA3	SA1	SB	SB2	SB3	SB4	SB5	SB6	SA1	SA5	SB5	SB6	SC	SC3	SC4	NB	NB2
×	×	○	×	×	△	△	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○	◎	◎	◎	◎

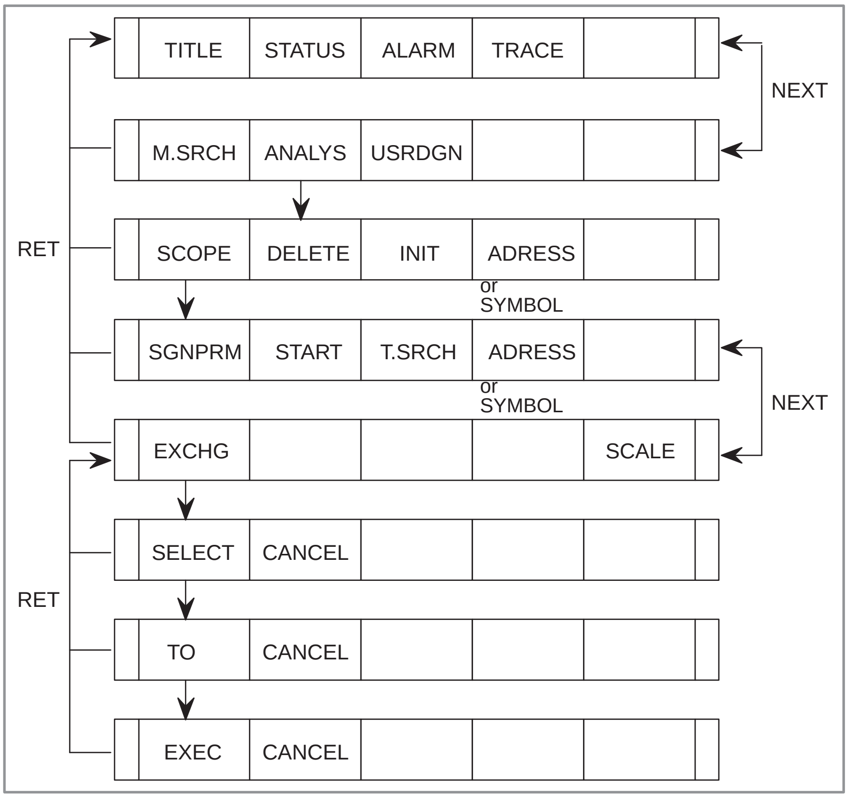
- The ladder edit module is required with PMC–SB/SB2/SB3 of the Series 16–MODEL A, and PMC–SA2/SA3 of the Series 18–MODEL A.
- The ladder edit card is required with PMC–RA1/RB3/RB4 of the Series 16/18–MODEL B, PMC–SA1/SB5/SB6 of the Series 16/18–MODEL C, PMC–SA1/SA3 of the Series 21/210–MODEL B, and PMC–PA3 of the Power Mate–MODEL–H.
- Work RAM is required with PMC–SC/SC3 of the Series 16–MODEL A.
- This function is provided as a standard function with PMC–SC3/SC4 of the Series 16/18–MODEL B/C.
- The ladder edit card is required with PMC–SA5/SB5/SB6 of the Series 16*i*/18*i*/21*i*–MODEL A.

3.6.1 Specifications

1) Maximum number of signals traced at the same time: 16
2) Maximum sampling period: 10 s
3) Sampling interval: 8 ms

3.6.2 Operation

Pressing the [ANALYS] key on the PMCDGN screen displays the parameter screen for diagnosing signals. Pressing the [SCOPE] soft key on the parameter screen displays the screen showing signal diagnosis. To return to the parameter screen, press the [SGNPRM] soft key.



3.6.3 Parameter Screen

To trace the state of a signal, the trace conditions need to be specified on the parameter screen. In a 9″ screen, a trace address can be specified by pressing the <PAGE ↓> key. (See Fig. 3.4.4.)

1) Setting parameters

   Move the cursor to a parameter to be specified. Enter a value and press the [INPUT] key. To delete the value of the parameter, move the cursor to the parameter, then press the [DELETE] soft key.

   a) SAMPLING TIME

      Specify the maximum trace time in the range of 1 to 10 s.

   b) TRIGGER ADDRESS

      Specify a trigger address from which the tracing starts on the PMC address. A symbol name can be used.

   c) CONDITION

      Specify the conditions at which the tracing starts.

      0 : When the [START] key is pressed
      1 : When the [START] key is pressed and the trigger address signal rises
      2 : When the [START] key is pressed and the trigger address signal falls

> **NOTE**
> Conditions 1 and 2 are effective when a trigger address is specified.

d) TRIGGER MODE

Sampled data for up to 10 seconds is stored in the trace buffer. A signal is stored in the buffer within 8 ms.
This parameter specifies the starting and end points for obtaining data.

0 : AFTER

In this mode, signal states are obtained in the period specified in parameter SAMPLING TIME from the time when the trigger conditions are satisfied.

1 : ABOUT

In this mode, signal states are obtained in the period specified in the parameter SAMPLING TIME with the time at the middle when the trigger conditions are satisfied.

2 : BEFORE

In this mode, signal states are obtained in the period specified in parameter SAMPLING TIME before the trigger conditions are satisfied.

3 : ONLY

In this mode, the signal states are obtained only when the trigger conditions are satisfied.

NOTE
Trigger mode 1 and 2 are effective when condition 1 or 2 is set.

e) SIGNAL ADDRESS

Specify up to 16 addresses at which the tracing is performed with PMC addresses or symbol names.

2) Initializing signal diagnosis data

Pressing the [INIT] soft key on the parameter screen initializes parameter data and trace data.

3) Displaying symbols for trigger addresses and trace addresses

Pressing the [ADRESS] soft key displays trigger and trace addresses for which symbols are defined and the key changes to the [SYMBOL] key. Pressing the [SYMBOL] key displays the symbols for trigger and trace addresses and the key changes to the [ADRESS] key.

```
PMC SIGNAL PARAMETER                          MONIT RUN

   SAMPLING TIME   :    10(1-10SEC)

   TRIGGER ADDRESS :  *ESP

   CONDITION       :     1
   (0:START 1:TRIGGER-ON 2:TRIGGER-OFF)

   TRIGGER MODE :        1
   (0:AFTER 1:ABOUT 2:BEFORE 3:ONLY)


>

 [SCOPE ] [DELETE] [INIT  ] [ADRESS] [       ]
```

ADRESS/SYMBOL

Fig. 3.6.3 (a)   Parameter setting screen 1

```
PMC SIGNAL PARAMETER                          MONIT RUN

              SIGNAL ADDRESS

        1 : ED0              9 : X0000.0
        2 : ED1             10 : X0000.1
        3 : ED2             11 : X0000.2
        4 : ED3             12 : X0000.3
        5 : ED4             13 : X0000.4
        6 : ED5             14 : X0000.5
        7 : ED6             15 : X0000.6
        8 : ED7             16 : X0000.7

>

 [SCOPE ] [DELETE] [INIT  ] [ADRESS] [       ]
```

ADRESS/SYMBOL

Fig. 3.6.3 (b)   Parameter setting screen 2

3.6.4 Signal Diagnosis Screen

After parameters are specified on the parameter screen, select the signal diagnosis screen.

Pressing the [START] soft key starts to trace the specified signal.

While the signals are traced, “EXECUTING” is displayed. When the tracing is finished, the period in which the specified signal was traced is displayed on the screen.

When the optional graphic function is provided, the waveform is displayed by using the graphic function.

When the function is not provided, waveform is displayed with “■” indicating the signal is on and “_” indicating the signal off.

In the ONLY mode, even when the optional graphic function is provided,“■” and “_” is used to display the waveform as shown in Fig. 3.6.4.

1) Starting or stopping the data sampling

   Pressing the [START] key starts sampling. Pressing the [STOP] key stops sampling and the sampled data is displayed.

2) Displaying traced data by specifying a period

   Enter a period in ms in which traced data is to be displayed. Pressing the [T.SRCH] key displays the traced data.

   Example) Entering 800, then pressing the [T.SRCH] key displays the waveform from 512 ms to 1024 ms.

3) Displaying symbols for trigger and trace addresses

   When symbols are defined for trigger and trace addresses, the symbols and addresses are displayed

4) Exchanging positions at which traced data is displayed

   Pressing the [EXCHG] key moves the cursor to the first traced address. Position the cursor to the trace address to be exchanged, using the <↑> or <↓> key, then press the [SELECT] key. Next, position the cursor to the trace address with which the selected trace address is to be exchanged, then press the [TO] key. Finally, press the [EXEC] key. The trace data is exchanged.

   During the above operation, all other soft keys are disabled until the [EXEC] key has been pressed. To cancel the exchange, press the [CANCEL] key.

5) Changing the time division (This function is available when the graphic function is used.)

   When displaying the signal waveform, the time division can be changed.

   Setting 8 . . . . . . 8 ms/divisions
   16 . . . . . 16 ms/divisions
   32 . . . . . 32 ms/divisions

   Pressing only the [SCALE] key increments the minimum scale from 8 to 32 ms, as follows:

6) Shifting traced data upward or downward

   Pressing the <PAGE ↑> key shifts traced data upward. Pressing the <PAGE ↓> key shifts traced data downward.

7) Shifting traced data left or right

Pressing the "←" key shifts traced data to the left. Pressing the "→" key shifts traced data to the right.

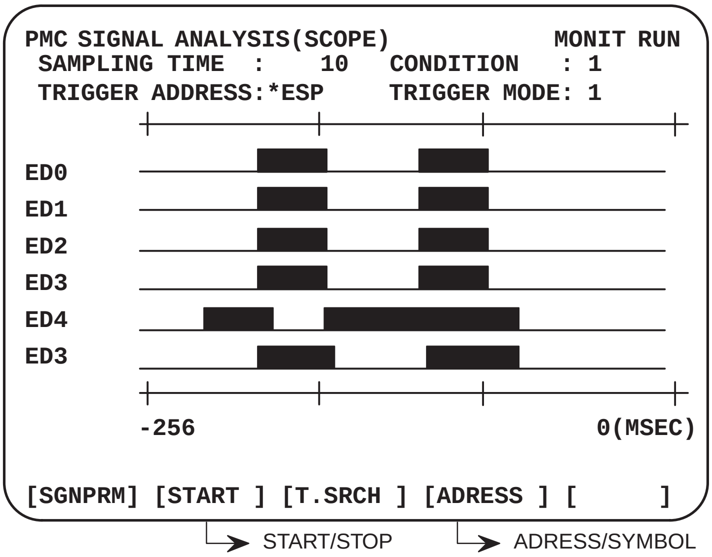


Fig. 3.6.4 Screen displaying signal diagnosis

3.6.5 Reading Signals Automatically at Power on

Since parameter and sampling data is stored in nonvolatile memory, data is retained when the power is turned off.

Data sampling can be started automatically, immediately after power–on, by setting sampling parameters and setting the keep relay beforehand. The keep relay setting method depends on the PMC model. See Section 4.3.3.

3.7 DISPLAYING THE RUNNING STATE OF A USER TASK (USRDGN)

Pressing the [USRDGN] key dynamically displays the running states of user tasks (including the third level of a ladder program) in the PMC (Fig. 3.7).

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2
×	×	×	×	×	×	×	×	×	×	Δ (Note)	Δ (Note)	○	○	○	○	○

Work RAM is necessary (A02B-0120-H987 for the PMC-SC and PMC-SC3 and A02B-0162-J151 or A02B-0162-J152 for the PMC-NB).

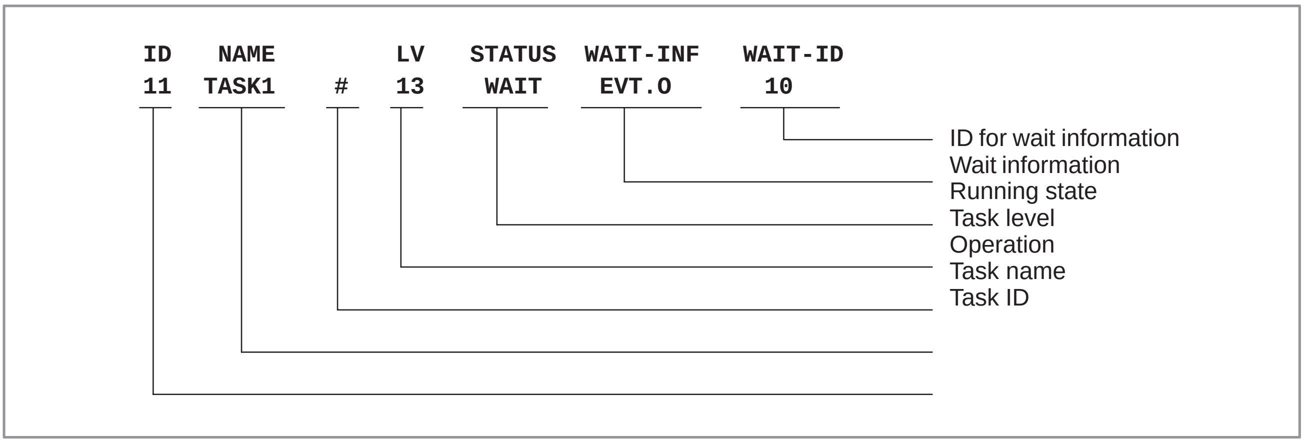
For details, refer to the “PMC-SC/SC3/SC4/NB Programming Manual (C language)” (B-61863E-1).

> **NOTE**
> C language board is required.

```
PMC MONIT USER TASK #1              MONIT RUN

ID NAME      LV STATUS WAIT-INF WAIT-ID
   LAD3      10 READY
10 TASK_01 @ 10 ACTIVE
11 TASK_02 # 11 READY
12 TASK_03   12 WAIT    TIM
13 TASK_04   13 WAIT    EVT.O        1
14 TASK_05   14 WAIT    EVT.A        3
15 TASK_06   15 WAIT    PKT       2340
16 TASK_07      STOP
17 TASK8      17 READY

 >

[       ][       ][       ][       ][       ]
```

Fig. 3.7 Screen displaying the running states of user tasks

[Displayed items]



1) Operation

Code	Description
#	RS-232C being used
@	NC command edit being used

2) Running state

Code	Description
ACTIVE	Running
READY	Ready
WAIT	Waiting
STOP	Task stopped
ERROR	The system deleted the task because the task had called library that is not supported.

3) Wait information

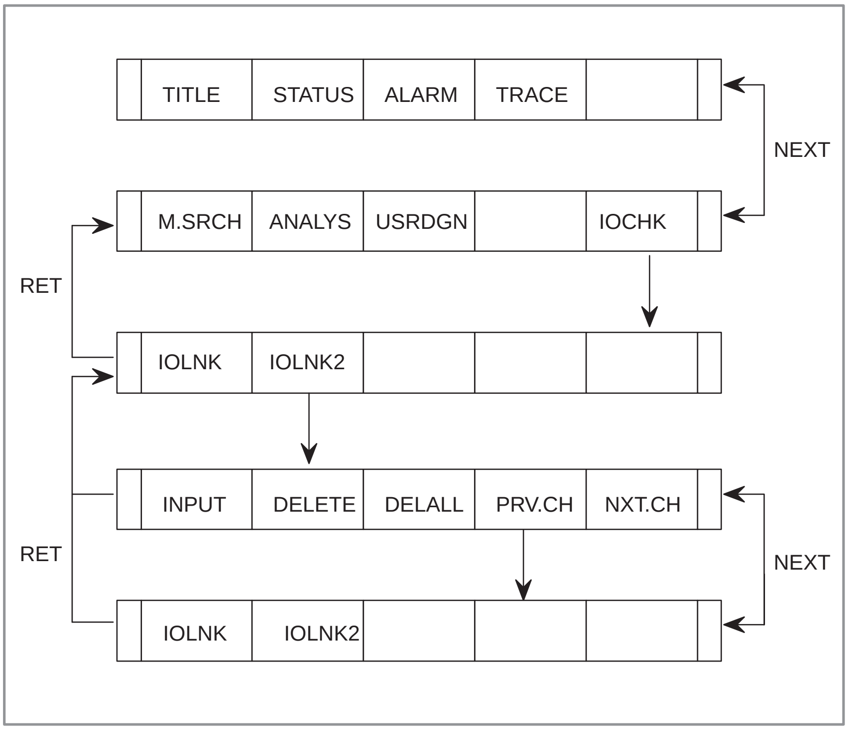
Code	Description
TIM	Waiting for time-out
EVT.A	Waiting for AND condition of event flags
EVT.O	Waiting for OR condition of event flags
SEM	Waiting for semaphore
MBX.R	Waiting for READ of the mail box
MBX.W	Waiting for WRITE of the mail box
PKT	Waiting for a packet to be received
PCMDI	Waiting for the PCMDI command to be issued

3.8 DISPLAYING AND SETTING THE CONFIGURATION STATUS OF I/O DEVICES (IOCHK)

○ : Supprted
× : Not supported

Power Mate–D/F/G	Power Mate–H	FS21 TA/TB	FS20	FS18	FS16–A	FS16–B	FS18–B	FS21*i*	FS16*i*	FS18*i*	FS15–B
×	○	×	×	×	×	○	○	○	○	○	○

In case of FS16–B/FS18–B :




```
PMC I/O CHECK

SELECT ONE OF FOLLOWING SOFT KEYS

IOLNK   : I/O LINK CHECK
IOLNK2  : I/O LINK–II SETTING
```

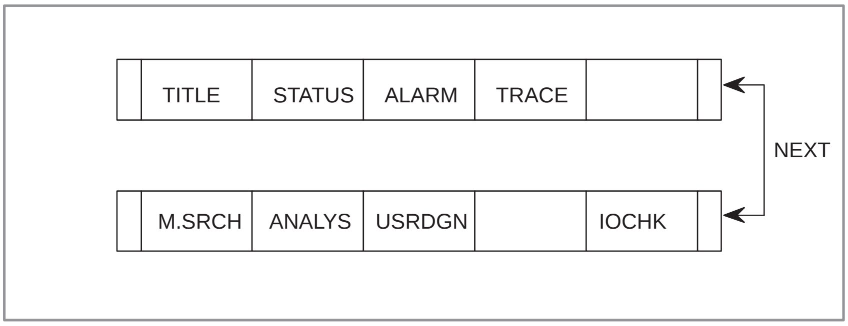
Fig. 3.8 I/O check menu screen

The I/O check screen has two functions. By pressing the soft key, the following screens are displayed.

[IOLNK] : I/O Link connecting check screen.

[IOLNK2]: I/O Link–II parameter setting screen.

In case of FS15–B :
FS15–B has not supported [IOLNK2] screen. By pressing [IOCHK] key, I/O Link connecting check screen is selected directly.



3.8.1 I/O Link Connecting Check Screen

The I/O Link connecting check screen displays the types and ID codes of the connected I/O devices for each group. When I/O device is not connected, ”NO I/O DEVICE” is displayed. When input to or output from an I/O devices is abnormal, check if the configuration of the connected I/O devices correct is by referring the screen.

```
I/O CHECK

          GROUP     ID      KIND OF UNIT

           00       80      CONNECTION UNIT
           01       82      OPERATOR PANEL
           02       84      I/O UNIT MODEL A
           03       96      CONNECTION UNIT
           04       4A      POWER MATE
```

Fig. 3.8.1 (a) Example of the I/O Link screen

Table 3.8.1 I/O devices and ID codes

Displayed I/O device name	ID	Actual I/O device
CONNECTION UNIT	80	Connection unit
OPERATOR PANEL	82	Operator's panel connection unit
I/O-B3	83	Expanded I/O B3
I/O UNIT MODEL A	84 to 87	I/O UNIT MODEL A
I/O UNIT MODEL B	9D to 9E	I/O UNIT MODEL B
POWER MATE	4A	Power Mate
CONNECTION UNIT	96	I/O Link connection unit
I/O MODULE	A9 to AA	Distributed I/O
OTHER UNIT	—	Other than above

When the screen is displayed like fig.3.8.1(a) The I/O devices are composed like following fig.3.8.1(2).

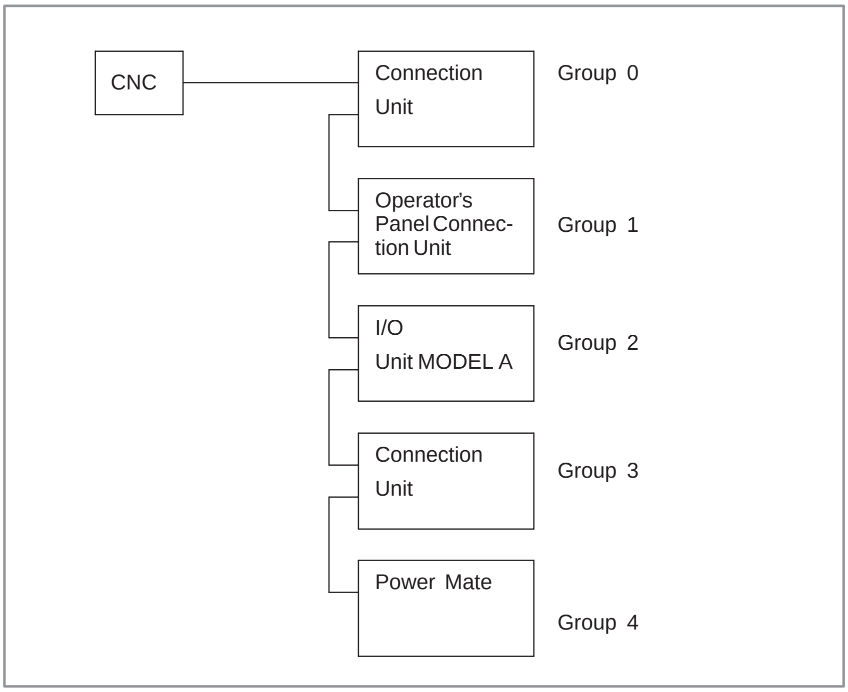


Fig. 3.8.1 (b) I/O Link configuration

3.8.2 I/O Link–II Parameter Setting Screen

In case of using the I/O Link–II function, set the following I/O Link–II parameter on this screen. Depending on the kind of I/O Link–II interface board, master/slave screen is displayed automatically.
Please refer to
FANUC I/O Link–II operating manual (B–62714EN)
about details of I/O Link–II and each parameter.

(1) Set parameters.
Move the cursor to the parameter by using the cursor key.
Type the data and press the soft key[INPUT] or MDI key<INPUT>.
The set parameter is saved to the I/O Link–II board when the data is input.

(2) Change channel.
Change the channel by the soft key [PRV.CH],[NXT.CH]. These keys are not displayed when the single channel is used.

(3) Delete parameter.
Move the cursor to the parameter by using the cursor key.
Press the soft key[DELETE].

(4) Delete all parameters.
Press the soft key[DELALL].
Press the soft key[EXEC] to delete all parameters.
Press the soft key[CANCEL] to cancel the deletion.

(5) Change page.
This screen is composed of two pages when the 9 inch CRT is used.
Change the page by using (PAGE) key of MDI.

(6) Re–start I/O Link–II
Press the soft key [START] to re–start I/O Link–II after editing the parameter.
When the re–start is completed normally, ”LINK STARTED” is displayed.
If the re–start fails, ”START ERROR” is displayed. In this case, check the parameter that is set.

Example of parameter setting of master.

```
PMC I/O LINK–II CH 1 (1/2)

 GENERAL:
   MAX SLAVE NO.    = 03 (1–31)
   SCAN TIME        = 0100 (1–9999)*2MSEC
   STATUS ADDRESS   = R0500
 DI/DO SETTING:
   DI/DO MAP MODE   = 1 (1,2)
   DI/DO DATA SIZE  = 16 (0–64)
   DO ADDRESS       = R0100
   DI ADDRESS       = R0150

[INPUT ] [DELETE ] [DELALL] [PRV.CH] [NXT.CH]
```

```
PMC I/O LINK–II CH 1 (2/2)

MESSAGE I/O SETTING:
MESSAGE SIZE    = 032 (0–128)
OUTPUT ADDRESS  = R0200
INPUT ADDRESS   = R0250
STATUS:
REFRESH TIME    = 40 MSEC
I/O LINK–II     = 6546/01 (MASTER)

[INPUT ] [DELETE ] [DELALL] [PRV.CH] [NXT.CH]
```

Fig. 3.8.2 (a) Example of the I/O Link–II screen.(master)

Example of parameter setting of slave.

```
PMC I/O LINK–II CH 1 (1/2)

 GENERAL:
   MAX SLAVE NO.    = 03 (1–31)
   STATION NO.      = 02 (1–31)
   STATUS ADDRESS   = R0900
 DI/DO SETTING:
   DI/DO MAP MODE   = 0 (0,2)
   DI/DO DATA SIZE  = 16 (0–64)
   DO ADDRESS       = R0000
   DI ADDRESS       = R0032

[INPUT ] [DELETE ] [DELALL] [PRV.CH] [NXT.CH]
```

```
   PMC I/O LINK–II CH 1 (2/2)

 MESSAGE I/O SETTING:
   MESSAGE SIZE     = 032 (0–128)
   OUTPUT ADDRESS   = R0256
   INPUT ADDRESS    = R0296
 STATUS:
   I/O LINK–II      = 6545/01 (SLAVE )

[INPUT ] [DELETE ] [DELALL] [PRV.CH] [NXT.CH]
```

Fig. 3.8.2 (b) Example of the I/O Link–II screen.(slave)

3.9 FORCED INPUT/OUTPUT FUNCTION

○ : Usable
△: See Note.
× : Not usable

Power Mate	FS20/ FS21A	FS18A	FS16A	FS16B FS18B		FS16C FS18C		FS21*i*	FS16*i* FS18*i*		FS15B
PA1 PA3	SA1 SA3	SA1, SA2 SA3	SB, SB2, SB3 SC, SC3	SB3 SC3	SB4 SC4	SB5 SC3	SB6 SC4	SA5	SB5	SB6	NB NB2
×	×	×	×	×	×	×	×	△	△	△	×

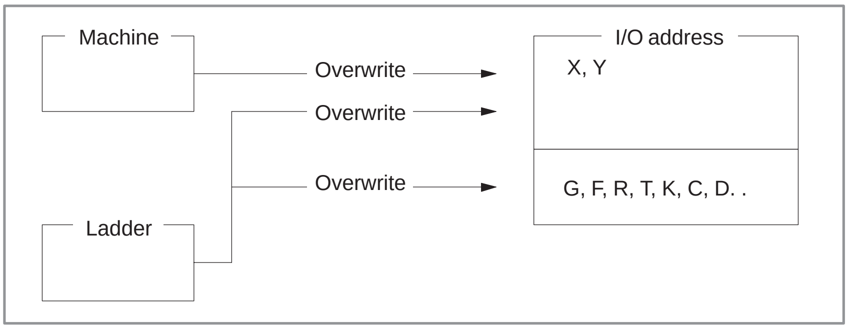
NOTE
With the FS16*i*/18*i*, the edit card or C board are required.
With the FS21*i*, the edit card is required.
With SA1 of the FS16*i*/18*i*/21*i*, only FORCING mode is valid.

3.9.1 Overview

This function can forcibly enter a value for the signal of an arbitrary PMC address. With this function, for example, an X value can be forcibly entered to enable sequence program debugging without using a machine, and a Y value can be forcibly entered to enable the signal wiring on the machine to be checked efficiently without using a sequence program. This function is added to the status display function.

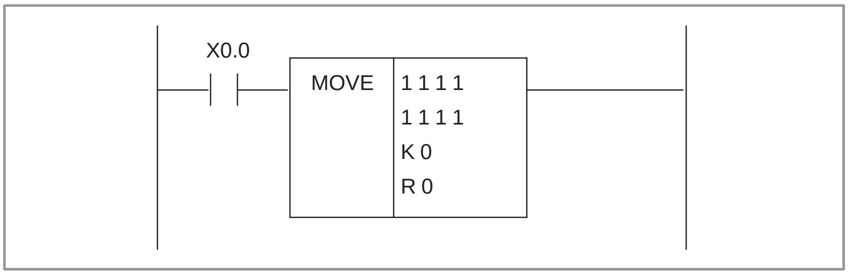
(1) Input mode
Two input modes are available. The user can choose between the two modes, depending on the application.

(a) FORCING mode
This mode is applicable to all addresses. When input/output scan is performed by a sequence program, however, a signal modified by the forced input/output function is overwritten, and the result of modification made by the forced input/output function is lost.



Example 1:

In this example, the forced input/output function is applied to R0 in the ladder program below.



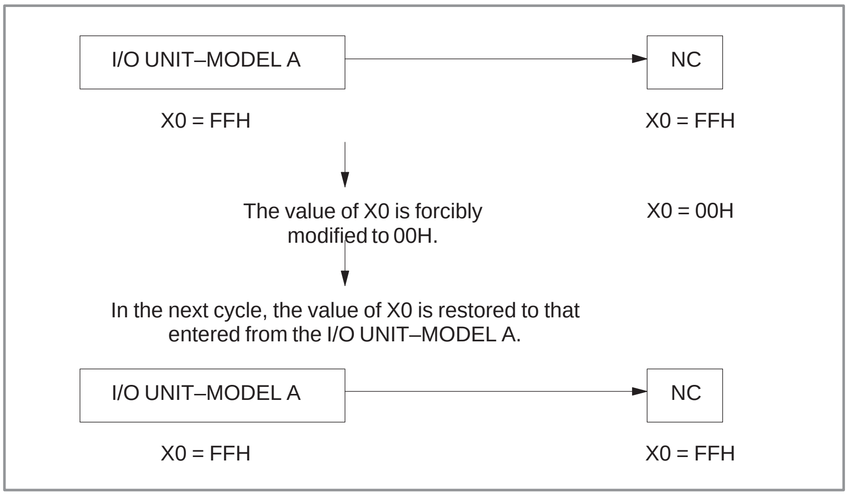
a. The initial signal states are as follows:
X0.0 = off, K0 = 55H, R0 = 00H

b. FFH is forcibly entered to R0.
X0.0 = off, K0 = 55H, R0 = FFH

c. When X0.0 is turned on, R0 assumes the result of output by the sequence program as follows:
X0.0 = on, K0 = 55H, R0 = 55H

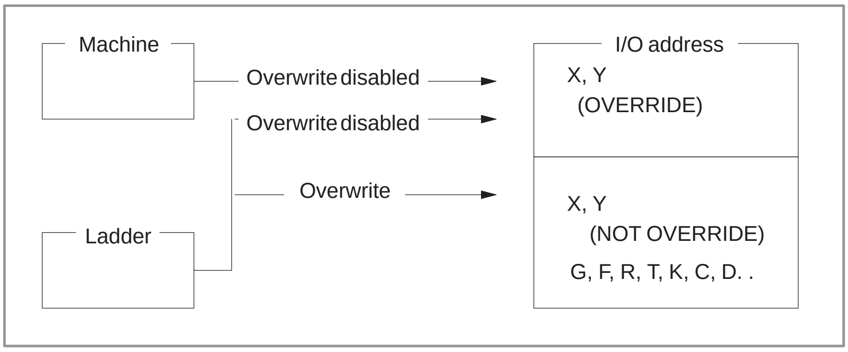
Example 2:

In this example, the forced input/output function is applied to X0 in a configuration where the I/O UNIT–MODEL A is connected to X0 over an I/O link.
The input value from the I/O UNIT–MODEL A is transferred to X0 at certain intervals. So, even if the value of X0 is forcibly modified, X0 is overwritten in the next cycle. Thus, the value of X0 is restored to the value input from the I/O UNIT–MODEL A.



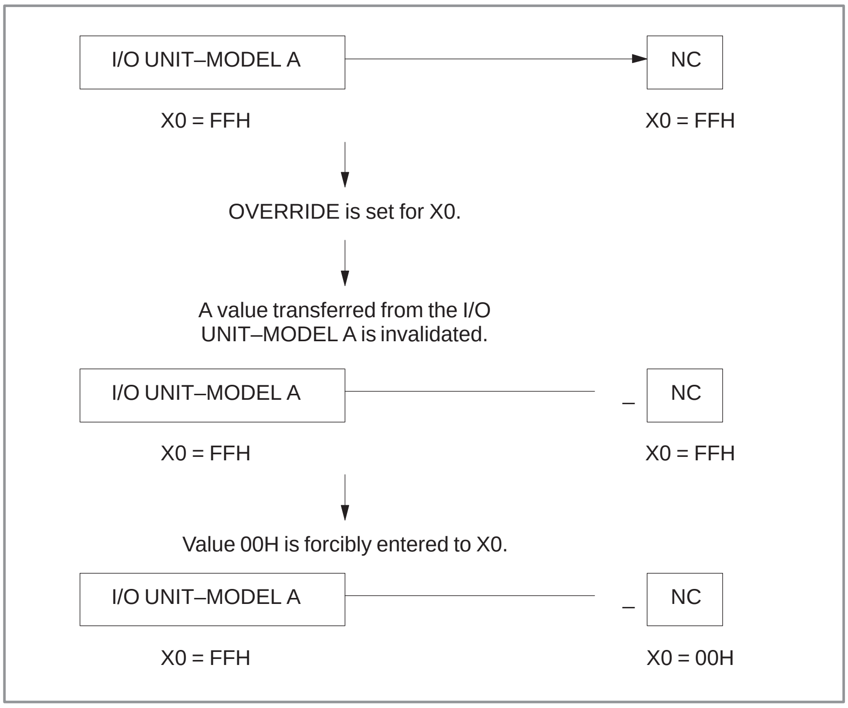
Cyclical transfer may also be performed for addresses that are not assigned. So, use the forced input/output function for X in FORCING mode to debug a sequence program when no machine is connected or assigned. Use the OVERRIDE mode to debug a sequence program in the case of I/O connection.

(b) OVERRIDE mode
That state in which values modified by the forced input/output function cannot be overwritten by a sequence program or machine signal is referred to as OVERRIDE mode. In OVERRIDE mode, OVERRIDE can be set for arbitrary X and Y signals.
FORCING is applied to those X and Y addresses where OVERRIDE is not set, and also to addresses other than the X and Y addresses.



Example:

In this example, the forced input/output function is applied to X0 in a configuration where the I/O UNIT–MODEL A is connected to X0 with an I/O link.



Thus, the forced input/output function for X addresses in OVERRIDE mode can be used to debug a sequence program while a machine is connected.
When a Y address is placed in the OVERRIDE state, a value that has been forcibly modified by the forced input/output function is output to the machine.

CAUTION

1 In OVERRIDE mode, input/output signals are updated at 8–ms intervals in sync with the first ladder level. When an I/O link is used which is usually updated at 2–ms intervals, an input/output signal timing delay occurs. For this reason, note that a sequence that depends on input/output signal timing may perform an unpredictable operation.

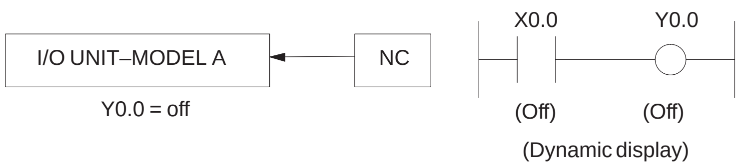
2 Note that, when OVERRIDE mode is set, the interval of the second level may be extended slightly.

3 Even if OVERRIDE is set for a Y address, the resultant value of a ladder operation before being modified by the forced input/output function is displayed as the coil on/off value in ladder dynamic display.

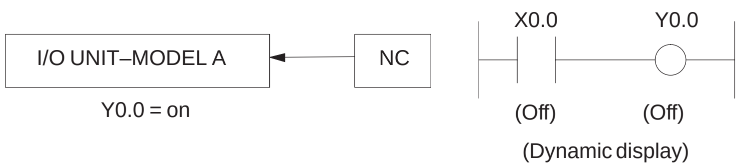
A value, after being modified by the forced input/output function, is output to the machine. So, note that the on/off indication in ladder dynamic display does not match a value output to the machine.

Example:

In this example, the forced input/output function is used for Y0.0 in the ladder below in a configuration where the I/O UNIT–MODEL A is connected to Y0 with an I/O link. Before OVERRIDE setting, the on/off indication in dynamic display matches a value output to the I/O UNIT–MODEL A as shown below.



If 1 is entered forcibly after setting OVERRIDE for Y0.0, the value modified by the forced input/output function is output to the I/O UNIT–MODEL A as shown below.



3.9.2 Setting/Operation for Enabling Forced Input/Output

Use the procedures below to set the input/output modes.

(1) Operation for enabling FORCING mode
Use the procedure below.

(a) Mount an edit card or C board.

(b) Turn on the power.

(c) Set the PROGRAM ENABLE (bit 1 of K17 or bit 1 of K900) setting parameter to YES.

(2) Operation for enabling OVERRIDE mode
Use the procedure below.

(a) Mount an edit card or C board.

(b) Turn on the power.

(c) Set the PROGRAM ENABLE (bit 1 of K17 or bit 1 of K900) setting parameter to YES.

(d) Set the OVERRIDE ENABLE setting parameter (OVERRIDE) to YES.

(e) Turn the power off, then back on.

WARNING

Special care must be exercised when modifying a signal with the forced input/output function. If the forced input/output function is used incorrectly, the operation of the machine may be unpredictable. Never use this function when persons are near the machine.

CAUTION

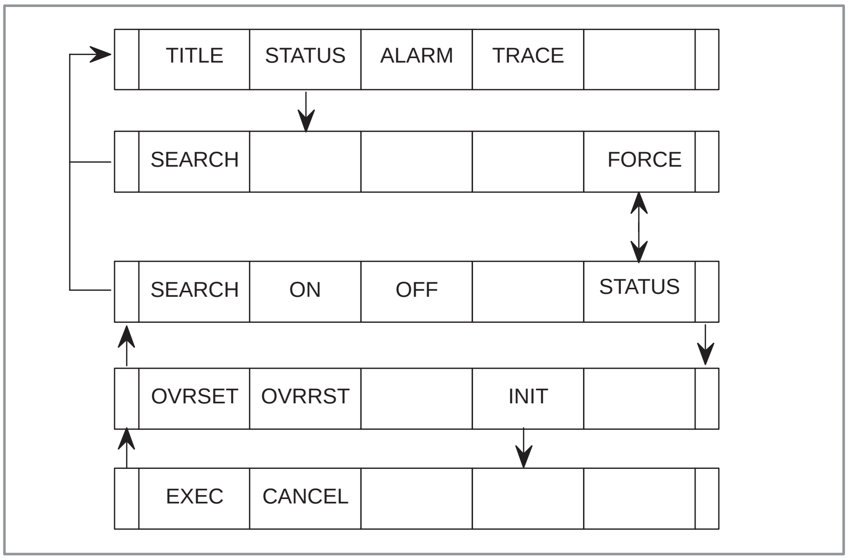
1 When shipping a machine, disable this function.

2 This function is disabled by extracting the edit card or setting the PROGRAM ENABLE setting parameter to NO.

3 The setting of OVERRIDE is not maintained when the power is turned off. When the power is turned on again, the setting of OVERRIDE is cleared for all X and Y addresses.

3.9.3 Screen Display

(1) Soft keys



(2) Details of the soft keys

(a) [SEARCH]
Searches for an address to be displayed.

(b) [ON] (Note 1)
Forcibly changes the value of a signal to 1.

(c) [OFF] (Note 1)
Forcibly changes the value of a signal to 0.

(d) [FORCE]/[STATUS] (Note 1)
Switches between the status display screen and forced input/output screen.

(e) [OVRSET] (Note 2)
Sets OVERRIDE for a signal.

(f) [OVRRST] (Note 2)
Clears an OVERRIDE setting for a signal.

(g) [INIT] (Note 2)
Clears OVERRIDE for all the X and Y areas.

NOTE
1 The soft key is enabled in FORCING mode.
2 The soft key is enabled in OVERRIDE mode.

(3) Forced input/output screens

FORCING Mode Status Screen

```
PMC SIGNAL STATUS                              MONIT  RUN

  NO.   7    6    5    4    3    2    1    0
X0000   0    0    0    0    0    0    0    0
X0001   0    0    0    0    0    0    0    1
X0002   0    0    0    0    0    0    1    0
X0003   0    0    0    0    0    1    0    0
X0004   0    0    0    0    1    0    0    0
X0005   0    0    0    1    0    0    0    0
X0006   0    0    1    0    0    0    0    0
X0007   0    1    0    0    0    0    0    0

[SEARCH] [       ] [       ] [       ] [ FORCE ]
```

FORCING Mode Setting Screen

```
PMC SIGNAL FORCING                             MONIT  RUN

  NO.   7    6    5    4    3    2    1    0
X0000   0    0    0    0    0    0    0    0
X0001   0    0    0    0    0    0    0    1
X0002   0    0    0    0    0    0    1    0
X0003   0    0    0    0    0    1    0    0
X0004   0    0    0    0    1    0    0    0
X0005   0    0    0    1    0    0    0    0
X0006   0    0    1    0    0    0    0    0
X0007   0    1    0    0    0    0    0    0

[SEARCH] [  ON   ] [  OFF   ] [       ] [STATUS]
```

OVERRIDE Mode Status Screen

```
PMC SIGNAL STATUS          OVERRIDE      MONIT RUN

  NO.   7     6     5     4     3     2     1     0
X0000   0     0     0     0     0     0     0     0
X0001   0     0     0     0     0     0     0    >1
X0002   0     0     0     0     0     0    >1     0
X0003   0     0     0     0     0    >1     0     0
X0004   0     0     0     0    >1     0     0     0
X0005   0     0     0    >1     0     0     0     0
X0006   0     0    >1     0     0     0     0     0
X0007   0    >1     0     0     0     0     0     0

[SEARCH] [       ] [       ] [       ] [ FORCE ]
```

OVERRIDE Mode Setting Screen

```
PMC SIGNAL FORCING         OVERRIDE      MONIT RUN

  NO.   7     6     5     4     3     2     1     0
X0000   0     0     0     0     0     0     0     0
X0001   0     0     0     0     0     0     0   0>1
X0002   0     0     0     0     0     0   1>1     0
X0003   0     0     0     0     0   1>1     0     0
X0004   0     0     0     0   0>1     0     0     0
X0005   0     0     0   0>1     0     0     0     0
X0006   0     0   0>1     0     0     0     0     0
X0007   0   1>1     0     0     0     0     0     0

[SEARCH] [  ON   ] [  OFF  ] [       ] [STATUS]

[OVRSET] [ OVRRST ] [       ] [ INIT  ] [      ]
```

3.9.4 Modifying the Values of Signals by Forced Input/Output

The method described below applies to both FORCING and OVERRIDE modes.

(1) Modifying signal values on a bit–by–bit basis
Position the cursor to a desired input bit, then enter a desired value by using one of the following three methods:

(a) Enter 1, then press the INPUT key or the [ON] soft key.
The signal is forcibly turned on.

(b) Enter 0, then press the INPUT key or the [OFF] soft key.
The signal is forcibly turned off.

(c) Press the INPUT key.
The on/off state of the signal is reversed.

```
PMC SIGNAL FORCING      OVERRIDE      MONIT RUN

   NO.  7    6    5    4    3    2    1    0
X0000   0    0    0    0    0    0    0    0
```

(2) Modifying signal values on a byte–by–byte basis
Move the cursor to a desired input byte, then enter a desired value by using one of the following three methods:

(a) Enter a binary number of no more than 8 digits, then press the INPUT key. (If an entered number is shorter than 8 digits, the number is entered starting from bit 0.)

Example: When 100 is entered, the number is entered at the following bit positions:

Bit No. 7 6 5 4 3 2 1 0
0 0 0 0 0 1 0 0

(b) Press the [ON] soft key.
All bits of the specified byte are set to 1.

(c) Press the [OFF] soft key.
All bits of the specified byte are set to 0.

```
PMC SIGNAL FORCING      OVERRIDE      MONIT RUN

   NO.  7    6    5    4    3    2    1    0
X0000   0    0    0    0    0    0    0    0
```

3.9.5 Setting/Clearing OVERRIDE

(1) Setting OVERRIDE
OVERRIDE can be set as described below.
By using the [OVRSET] soft key, place the desired bit/byte in the OVERRIDE state.

(a) Setting OVERRIDE on a bit–by–bit basis
Move the cursor to the desired bit, then press the [OVRSET] soft key.

```
PMC SIGNAL FORCING      OVERRIDE      MONIT RUN

   NO.  7    6    5    4    3    2    1    0
X0000   0    0    0    0    0    0    0    0
```

Then, the display changes as shown below.

```
PMC SIGNAL FORCING      OVERRIDE      MONIT RUN

   NO.  7    6    5    4    3    2    1    0
X0000   0    0    0    0    0  0>0    0    0
```

(b) Setting OVERRIDE on a byte–by–byte basis
Position the cursor to the desired byte, then press the [OVRSET] soft key.

```
PMC SIGNAL FORCING      OVERRIDE      MONIT RUN

   NO.  7    6    5    4    3    2    1    0
X0000   0    0    0    0    0    0    0    0
```

Then, the display changes as shown below.

```
PMC SIGNAL FORCING      OVERRIDE      MONIT RUN

   NO.  7    6    5    4    3    2    1    0
X0000 0>0  0>0  0>0  0>0  0>0  0>0  0>0  0>0
```

When OVERRIDE is set, the signal resumes the state existing before OVERRIDE setting. When OVERRIDE is set for a pulse signal, for example, the signal state existing when the [OVRSET] soft key is pressed is overridden.

Each bit for which OVERRIDE is set has ”>” added in the status display.

In OVERRIDE mode, those signal states that are not in the OVERRIDE state are also displayed.

For an X signal:

(Input signal from the machine) 0 > 1 (input signal to the ladder)

For a Y signal

(Output signal from the ladder) 0 > 1 (output signal to the machine)

(2) Clearing OVERRIDE
By using the [OVRRST] soft key, clear the OVERRIDE state of the desired bit/byte.
This operation returns the state of a signal to the state existing before OVERRIDE setting.

(a) Clearing OVERRIDE on a bit–by–bit basis
Position the cursor to the desired bit, then press the [OVRRST] soft key.

```
PMC SIGNAL FORCING        OVERRIDE      MONIT RUN

   NO.  7    6    5    4    3    2    1    0
X0000   0    0    0    0    0 1> 0    0    0
```

Then, the display changes as shown below.

```
PMC SIGNAL FORCING        OVERRIDE      MONIT RUN

   NO.  7    6    5    4    3    2    1    0
X0000   0    0    0    0    0    1    0    0
```

(b) Clearing OVERRIDE on a byte–by–byte basis
Position the cursor to the desired byte, then press the [OVRRST] soft key.

```
PMC SIGNAL FORCING        OVERRIDE      MONIT RUN

   NO.  7    6    5    4    3    2    1    0
X0000 1>0  0>1  1>0  0>0  1>0  1>1  1>0  0>1
```

Then, the display changes as shown below.

```
PMC SIGNAL FORCING        OVERRIDE      MONIT RUN

   NO.  7    6    5    4    3    2    1    0
X0000   1    0    1    0    1    1    1    0
```

(c) Clearing OVERRIDE for all X and Y areas
Press the [INIT] soft key. Then, the message ”CLEAR OVERRIDES OK?” is displayed.
By using the soft key [EXEC]/[CANCEL], execute or cancel the clearing of OVERRIDE.

4 PMC PARAMETERS SETTING AND DISPLAY (PMCPRM)

4.1 OUTLINE

Parameters of TIMER, COUNTER, KEEP RELAY and DATA TABLE, which are nonvolatile, are set and displayed with CRT/MDI panel.
To use this function, press the soft key [PMCPRM] of PMC basic menu screen.

NOTE
The address and contents of the nonvolatile memory are described in 3.5 to 3.8 of I–3. “ADDRESS” and I–6. “NONVOLATILE MEMORY”.

4.2 INPUT PMC PARAMETERS FROM MDI PANEL

1 Place the sequence program in the STOP state.

2 When the sequence program is in the RUN state, perform the setting below.

(1) Set NC to “MDI” mode or “Emergency Stop” status.

(2) Set “PWE” of NC setting screen or Program Protect Signal(“KEY4”) to 1. (See the following table.)

	PWE	KEY4	
TIMER	○		
COUNTER	○	○	: Alternative
KEEP RELAY	○		
DATA TABLE	○	○	: Alternative

3 Press the following soft keys to select the screens.

[TIMER] : TIMER screen
[COUNTR]: COUNTER screen
[KEEPRL] : KEEP RELAY screen
[DATA] : DATA TABLE screen

4 By using cursor keys, move cursor to the position for setting value.

5 Press the INPUT key after typing the value.

6 Set “PWE” or “KEY4” to 0 after setting value.

4.2.1 Multiple Data Input

1 This function is effective on the screen of TIMER, COUNTER, KEEP RELAY, and DATA TABLE.

2 Up to 10 data can be inputted at once.

3 The cursor is moved to the final data position of inputted data.

(1) Input method

- “ ; (EOB)” is used for separating data.

  Press the INPUT key after typing “100; 200; 300”.

- “ ; =” is used for inputting the same value as preceding data.

  Press the INPUT key after typing “100; =; =; 200; =”, and it becomes “100, 100, 100, 200, 200”.

- “ ; ; ” is used for skipping an input address.

  Press the INPUT key after typing “100; ; 100”.

  The second data is not inputted.

4.3 SETTING AND DISPLAY SCREEN

4.3.1 Timer Screen (TIMER)

The TIMER times of the functional instruction TMR(SUB 3) are set and displayed on this screen.

Page No.(Change pages with the page keys.)
The TIMER No.s used by TIMER instruction
The addresses refered by sequence program

```
PMC PRM  (TIMER)   #001            MONIT RUN

 NO.  ADDRESS   DATA      NO.  ADDRESS   DATA
 01     T00     2016      11     T20     1000
 02     T02       48      12     T22        8
 03     T04      960      13     T24        0
 04     T06     1008      14     T26       32
 05     T08        0      15     T28        0
 06     T10        0      16     T30        0
 07     T12       96      17     T32     2000
 08     T14        0      18     T34        0
 09     T16        8      19     T36        8
 10     T18       16      20     T38    10000

[TIMER ] [COUNTR ] [KEEPRL ] [ DATA ] [      ]
```

TIMER times(See the following table.)

TIMER No.s	Minimum time	Maximum time
1 to 8	48 (ms)	1572.8 (s)
9 to 40 or 9 to 150 (*1)	8 (ms)	262.136 (s)

*1 The usable numbers vary from one model to another. For details, see Section 5.4 in Part I.

4.3.2 Counter Screen (COUNTR)

The maximum(PRESET) values and CURRENT values of the functional instruction CTR(SUB 5) are set and displayed on this screen.

```
PMC PRM  (COUNTER)  #001           MONIT RUN

  NO.   ADDRESS             PRESET     CURRENT
  01      C00                    4           1
  02      C04                    4           2
  03      C08                    4           3
  04      C12                    5           4
  05      C16                    4           5
  06      C20                  545           6
  07      C24                    5           3
  08      C28                    6           2
  09      C32                    6           1
  10      C36                    6           4


[TIMER ] [COUNTR ] [KEEPRL ] [ DATA ] [      ]
```

The COUNTER No.s used by CTR instruction

The addresses refered by sequence program

Page No.(Change pages with the page keys.)

The maximum(PRESET) values of COUNTER
(The minimum values are specified in CTR instruction.)

The CURRENT values of COUNTER

0–9999 in BCD(0–32767 in Binary) can be set as the PRESET and CURRENT values.

4.3.3 Keep Relay (KEEPRL)

The KEEP RELAYs and the Data for Controlling nonvolatile memory are set and displayed on this screen.

```
PMC PRM  (KEEP RELAY)                  MONIT RUN

NO. ADDRESS    DATA       NO. ADDRESS   DATA
01    K00     00000000    11    K10    00000000
02    K01     00000000    12    K11    00000000
03    K02     00000000    13    K12    00000000
04    K03     00000000    14    K13    00000000
05    K04     00000000    15    K14    00000000
06    K05     00000000    16    K15    00000000
07    K06     00000000    17    K16    00000000
08    K07     00000000    18    K17    00000110
09    K08     00000000    19    K18    00000000
10    K09     00000000    20    K19    00000000


[TIMER ] [COUNTR ] [KEEPRL ] [ DATA ] [       ]
```

The address used by sequence program

■ : This area is reserved for special use. (Note)

NOTE

1 The Data for Controlling Nonvolatile Memory(K16)
Refer to I–6.1(4)”Nonvolatile Memory Control”.

2 The Data for PMC Management Software(K17,18,19)
Be careful of using the following KEEP RELAYs, because they are used by PMC Management Software.

The Data for PMC Management Software

Model	PA1	PA3
PMC control software data 1	K17	K17
PMC control software data 2	K18	K18
Not used	K19	K19

Model	SA1	SA2	SA3/ SA5
PMC control software data 1	K17	K17	K17
PMC control software data 2	K18	K18	K18
Not used	K19	K19	K19

Model	SB	SB2	SB3/ SB5	SB4/ SB6
PMC control software data 1	K17	K17	K17	K900
PMC control software data 2	K18	K18	K18	K901
Not used	K19	K19	K19	K902 to K909

Model	SC	SC3	SC4
PMC control software data 1	K17	K17	K900
PMC control software data 2	K18	K18	K901
PMC control software data 3	K19	K19	K902
Not used			K903 to K909

Model	NB	NB2
PMC control software data 1	K17	K900
PMC control software data 2	K18	K901
PMC control software data 3	K19	K902
Not used		K903 to K909

PMC control software data 1 (K17 or K900)

	#7	#6	#5	#4	#3	#2	#1	#0
K17 or K900	DTBLDSP	ANASTAT	TRCSTART	MEMINP	SELCTMDL	AUTORUN	PRGRAM	LADMASK

#7 DTBLDSP
0 : The PMC parameter data table control screen is displayed.
1 : The PMC parameter data table control screen is not displayed.

#6 ANASTAT
0 : In the function for displaying signal waveforms, sampling starts when the [START] soft key is pressed.
1 : In the function for displaying signal waveforms, sampling starts automatically when the power is turned on.

* This bit is effective only for applicable models specified in 3.6, “Function for Displaying Signal Waveforms (ANALYS),” in Part II.

#5 TRCSTAT
- 0 : In the signal trace function, tracing starts when the [EXEC] soft key is pressed.
- 1 : In the signal trace function, tracing starts automatically when the power is turned on.

#4 MEMINP
- 0 : Data cannot be entered in the memory content display function.
- 1 : Data can be entered in the memory content display function.

* This bit is effective only for applicable models specified in 3.5, “Display the Contents of Memory (M.SRCH),” in Part II.

#3 SELCTMDL
- 0 : The sequence program stored in ROM (EPROM) is enabled.
- 1 : The sequence program stored in the RAM module or ROM module (only for PMC–SB2/SB3) is enabled.

* This bit enables either the EPROM module or ROM/RAM module when both modules are provided. It is effective for the PMC–SA1, SA2, SA3, SB, SB2, and SB3. (It is not effective for the Series 20 or Series 16/18 MODEL–B.)

#2 AUTORUN
- 0 : In RAM operation, a sequence program is not executed when the power is turned on.
- 1 : In RAM operation, a sequence program is executed automatically when the power is turned on (as in ROM operation).

* For the PMC of the Series 16/18 MODEL–B/C, Series 16*i*/18*i*/21*i*–MODEL A, this bit has the following meanings.
- 0 : The sequence program is executed automatically when the power is turned on.
- 1 : The sequence program is executed when the [RUN] soft key is pressed.

#1 PRGRAM
- 0 : The built–in programmer function is not operated. (The programmer menu is not displayed, either.)
- 1 : The built–in programmer function is operated. (The programmer menu is displayed.)

> **CAUTION**
> Set this bit to 0 before shipment from the factory.
> If the bit setting is left as 0, the operator may stop execution of the ladder diagram by mistake, and cause an accident.

#0 LADMASK
- 0 : Ladder dynamic display (PCLAD) is performed.
- 1 : Ladder dynamic display (PCLAD) is not performed.

PMC control software data 2 (K18 or K901)

	#7	#6	#5	#4	#3	#2	#1	#0
K18 or K901	IGNDINT		CHKPRTY	CALCPRTY	TRNSRAM	TRGSTAT	DBGSTAT	IGNKEY

#7 IGNDINT 0 : When the screen is switched to the PCMMDI screen, the CRT is initialized.

1 : When the screen is switched to the PCMMDI screen, the CRT is not initialized.

* The flag is used to determine whether PMC control software initializes the CRT when the screen is switched to the PCMMDI screen. Design application software sot that the CRT is initialized when this flag is on.

#5 CHKPRTY 0 : The parity check is performed for the system ROM and program ROM/RAM.

1 : The parity check is not performed for the system ROM and program ROM/RAM.

#4 CALCPRTY 0 : The built–in programmer function performs RAM parity calculation.

1 : The built–in programmer function does not performs RAM parity calculation.

#3 TRNSRAM 0 : A ladder program is not automatically sent to the backup RAM after on–line editing is completed.

1 : A ladder program is automatically sent to the backup RAM after on–line editing is completed.

#2 TRGSTAT 0 : The trigger stop function does not automatically start when the power is turned on.

1 : The trigger stop function automatically starts when the power is turned on.

#1 DBGSTAT 0 : In the C language debug function, the break processing does not automatically start when the power is turned on.

1 : In the C language debug function, the break processing automatically starts when the power is turned on.

* This flag is effective for the PMC–SC/SC3/SC4.

#0 IGNKEY 0 : Function keys are enabled when the user program displays the user screen.

1 : Function keys are disabled when the user program displays the user screen.

* This flag is effective for the PMC–SC/SC3/SC4/NB/NB2, When this bit is set to 1 in the user screen, the screen cannot be switched to the NC screen using the function keys. For this reason, a program that always sets this bit to 0 or that changes the screen to the NC screen is required.

* Be sure to set this flag to 1 when the CNC screen display function is used to display the user menu on an open CNC.

PMC control software data 3 (K19 or K902)

	#7	#6	#5	#4	#3	#2	#1	#0
K19 or K902					LCD–MONO		C–REJECT	FROM–

#3 LCD–MONO 0 : Ladder–related display is brightness–adjusted when a monochrome LCD is used with the VGA setting.

1 : Ladder–related display is not brightness–adjusted but is displayed in reverse video when a monochrome LCD is used with the VGA setting.

#1 C–REJECT 0 : A C–language program is activated.

1 : A C–language program is forcibly not activated.

* The flag is effective for the PMC–RC/RC3/RC4.

#0 FROM–WRT 0 : The program is not automatically written to F–ROM.

1 : After a lodder program on C program has been edited, the program is automatically written to F–ROM.

CAUTION
Be sure to set bits not used in the PMC control software data to 0.

In case of PMC–PA1/PA3 on Power Mate

	#7	#6	#5	#4	#3	#2	#1	#0
K17	DTBLDSP	ANASTAT	TRCSTART	MEMINP		AUTORUN	PRGRAM	LADMASK

#7 DTBLDSP 0 : The PMC parameter data table control screen is displayed.

1 : The PMC parameter data table control screen is not displayed.

#6 ANASTAT 0 : Sampling is started with the signal waveform display function by using the execution soft key.

1 : Sampling is started with the signal waveform display function, automatically after the power is turned on.
(This setting cannot be used with the Power Mate–D/F.)

#5 TRCSTAT 0 : Trace operation is started with the signal trace function by using the trace execution soft key.

1 : Trace operation is started with the signal trace function, automatically after the power is turned on.

#4 MEMINP 0 : Data cannot be entered with the memory contents display function.

1 : Data can be entered with the memory contents display function.
(This setting cannot be used with the Power Mate–D/F.)

#2 AUTORUN
0 : The sequence program is executed automatically after the power is turned on.
(This setting cannot be used with the Power Mate–D/F.)
1 : The sequence program is executed by using the sequence program execution soft key.

#1 PRGRAM
0 : The built–in programmer function is not operated. (The programmer menu is not displayed either.)
1 : The built–in programmer function is operated. (The programmer menu is displayed.)

#0 LADMASK
0 : Ladder dynamic display (PMCLAD) is performed.
1 : Ladder dynamic display (PMCLAD) is not performed.

	#7	#6	#5	#4	#3	#2	#1	#0
K18			CHKPRTY	CALCPRTY	TRANSRAM	TRGSTAT		

#5 CHKPRTY
0 : System ROM and program ROM/RAM parity checks are performed.
1 : System ROM and program ROM/RAM parity checks are not performed.

#4 CALCPRTY
0 : A RAM parity calculation is performed with the built–in programmer function.
1 : A RAM parity calculation is not performed with the built–in programmer function.

#3 TRNSRAM
0 : Upon the completion of online editing, the ladder program is not automatically transferred to RAM for editing.
1 : Upon the completion of online editing, the ladder program is automatically transferred to RAM for editing. (This setting cannot be used with the Power Mate–D/F.)

#2 TRGSTAT
0 : When the power is turned on, the trigger stop function is not started automatically.
1 : When the power is turned on, the trigger stop function is started automatically.
(This setting cannot be used with the Power Mate–D/F.)

	#7	#6	#5	#4	#3	#2	#1	#0
K19								FROM–WRT

#0 FROM–WRT
0 : After a ladder is edited, the ladder is not automatically written to F–ROM.
1 : After a ladder is edited, the ladder is automatically written to F–ROM.
(This setting cannot be used with the Power Mate–D/F.)

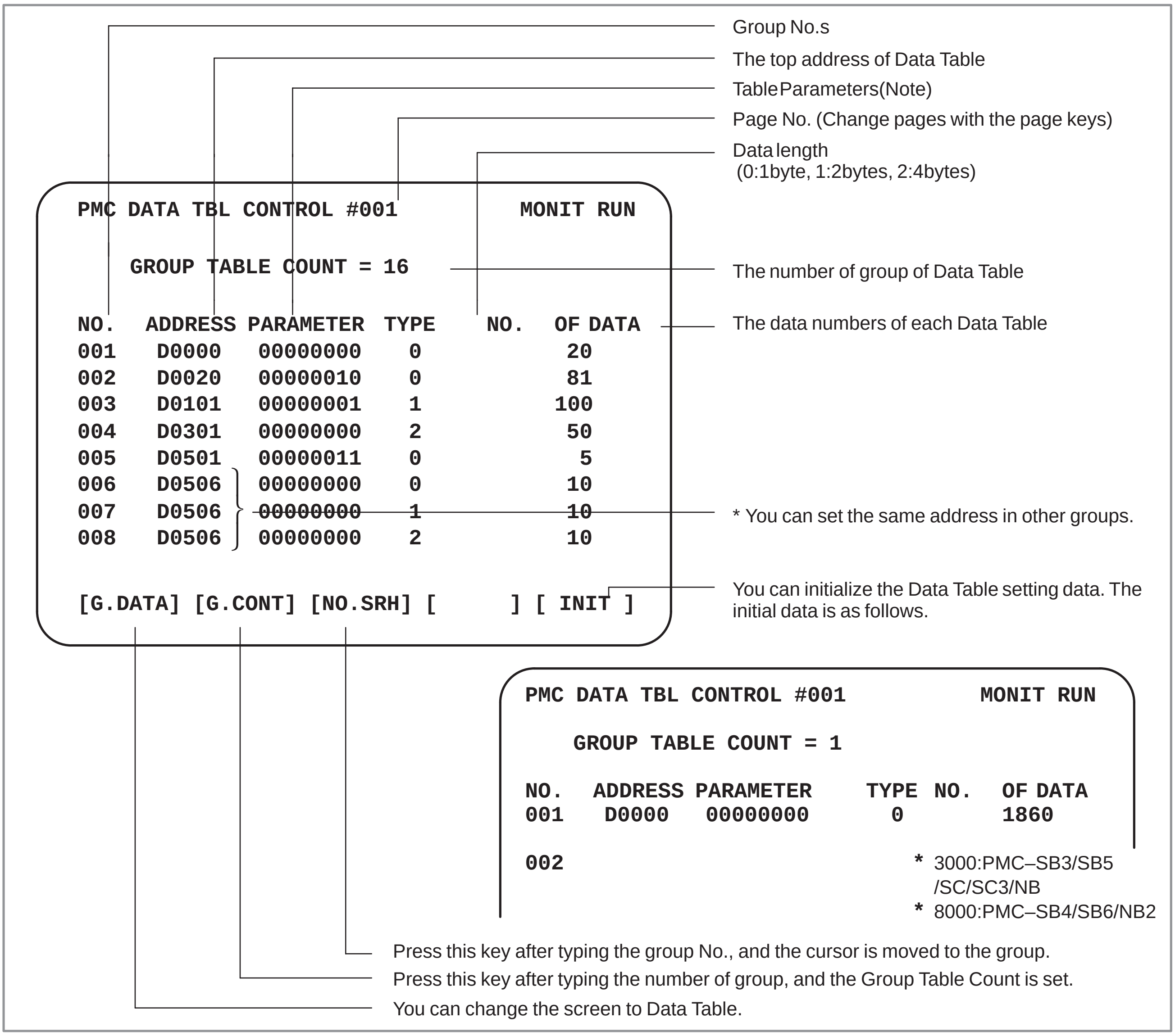
> **CAUTION**
> The unused area of the data for the PMC management software must always be set to 0.

4.3.4 Data Table (DATA)

DATA TABLE consists of two screens, that is, Data Table Controlling Data screen and Data Table screen.

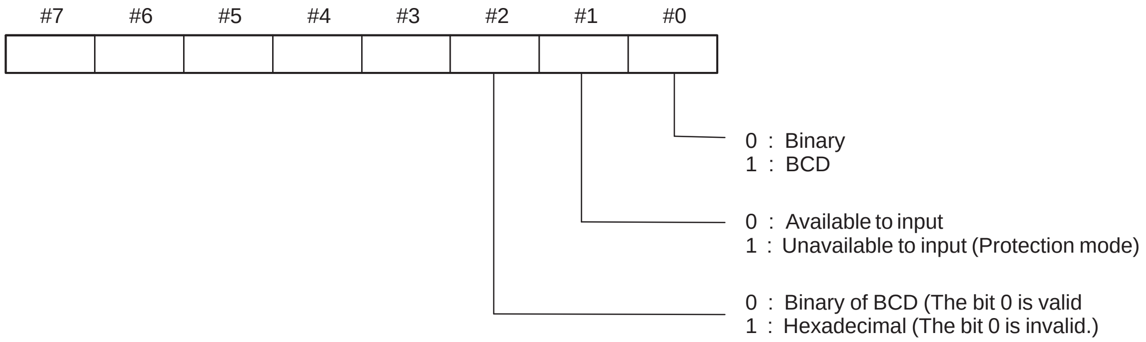
(1) Data Table Controlling Data Screen

Data Table Controlling Data Screen for controlling Data Table is displayed by pressing the soft key [DATA].



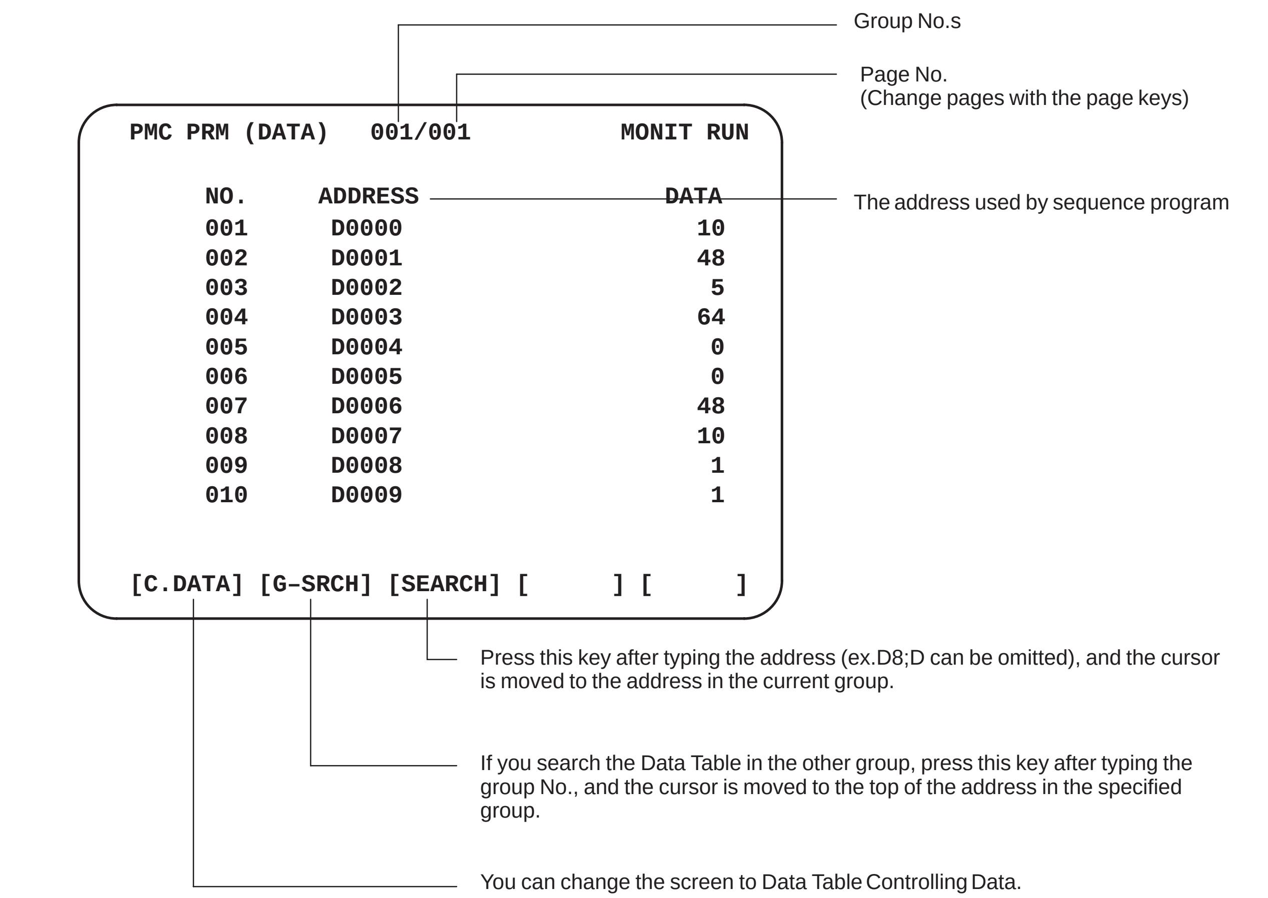
NOTE

Table Parameter



(2) Data Table Screen

If the Data Table Controlling Data is set, Data Table Screen is displayed by pressing the soft key [G.DATA].



4.4 SETTING SCREEN

Part of KEEP RELAY parameters can be set on SETTING Screen.

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2
×	○	Δ	×	○	○	Δ	×	○	○	○	○	Δ	Δ	○	○	○

> **NOTE**
> Δ: Can be used for the specific series of CNC.
> (Series 16 : B005/11 to, B105/08 to, B305/04 to, B009/03 to, All serieses of model C)
> (Series 18 : BD03/12 to, BE03/09 to, BG23/03 to, BG03/06 to, BD09/02 to, BE09/14 to, All serieses of model C)
> PMC–PA3 can be used only with Power Mate–H.

- The display items are different according to the type of CNC.
- The parameter is set by a soft key or the <INPUT> key with 0 or 1.
- Once an item has been set, the cursor moves to the next item.

[PMC–SA1/SA3/SB/SB3/SB4 on SETTING screen]

```
PMC PRM (SETTING)                     MONIT RUN

PROGRAMMER ENABLE    = 0(0:NO 1:YES)                (K17. 1)
LADDER START (RAM)   = 0(0:MANUAL 1:AUTO)           (K17. 2)
SELECT ROM/RAM       = 0(0:ROM 1:RAM)               (K17. 3)
SIGNAL TRACE START   = 0(0:MANUAL 1:AUTO)           (K17. 5)
DATA TBL CNTL SCREEN = 0(0:YES 1:NO)                (K17. 7)
SIGNAL TRIGGER START = 0(0:MANUAL 1:AUTO)           (K18. 2)
TRANS LADDER(ONLEDT) = 0(0:MANUAL 1: AUTO)          (K18. 3)


[ NO   ] [ YES  ] [      ] [      ] [      ]
```

* The bracketed addresses show the related KEEP RELAYs.

[PMC–PA3 on SETTING screen]

```
PMC PRM (SETTING)                        MONIT RUN

PROGRAMMER ENABLE    = 0(0:NO 1:YES)              (K17. 1)
LADDER START         = 0(0:AUTO 1:MANUAL)         (K17. 2)
RAM WRITE ENABLE     = 0(0:NO 1:YES)              (K17. 4)
SIGNAL TRACE START   = 0(0:MANUAL 1:AUTO)         (K17. 5)
DATA TBL CNTL SCREEN = 0(0:YES 1:NO)              (K17. 7)
SIGNAL TRIGGER START = 0(0:MANUAL 1:AUTO)         (K18. 2)


[ NO   ] [ YES  ] [      ] [      ] [      ]
```

* The bracketed address show the related KEEP RELAYs

[PMC–RC/RC3/RC4 on SETTING screen]

```
PMC PRM (SETTING)                        MONIT RUN

PROGRAMMER ENABLE    = 0(0:NO 1:YES)              (K17. 1)
LADDER START (RAM)   = 0(0:MANUAL 1:AUTO)         (K17. 2)
RAM WRITE ENABLE     = 0(0:NO 1:YES)              (K17. 4)
SIGNAL TRACE START   = 0(0:MANUAL 1:AUTO)         (K17. 5)
SIGNAL ANALYS START  = 0(0:MANUAL 1:AUTO)         (K17. 6)
DATA TBL CNTL SCREEN = 0(0:YES 1:NO)              (K17. 7)
FUNC KEY INP(CUSTOM) = 0(0:AVAL 1:IGNORE)         (K18. 0)
DEBUG FUNC START     = 0(0:MANUAL 1:AUTO)         (K18. 1)
SIGNAL TRIGGER START = 0(0:MANUAL 1:AUTO)         (K18. 2)
TRANS LADDER (ONLEDT)= 0(0:MANUAL 1:AUTO)         (K18. 3)
INITPMC–MDI SCREEN   = 0(0:YES 1:NO)              (K18. 7)


[ NO   ] [ YES  ] [      ] [      ] [      ]
```

* The bracketed addresses show the related KEEP RELAYs.

[PMC–NB/NB2 on SETTING screen]

```
PMC PRM (SETTING)                                MONIT RUN

PROGRAMMER ENABLE             = 0 (0:NO 1:YES)
AUTOMATIC LADDER START        = 0 (0:MANUAL 1:AUTO)
RAM WRITE ENABLE IN [M.SRC]   = 0 (0:NO 1:YES)
SIGNAL TRACE START            = 0 (0:MANUAL 1:AUTO)
SIGNAL ANALYSIS START         = 0 (0:MANUAL 1:AUTO)
DATA TABLE CONTROL SCREEN     = 0 (0:YES 1:NO)
NC/PC KEY EFFECTIVE           = 0 (0:AVAL 1:IGNORE)
DEBUG FUNCTION START          = 0 (0:MANUAL 1:AUTO)
SIGNAL TRIGGER START          = 0 (0:MANUAL 1:AUTO)
TRANSFER LADDER (ONLINE–EDIT) = 0 (0:MANUAL 1:AUTO)
INITIALIZE PMC–MDI SCREEN     = 0 (0:YES 1:NO)
WRITE TO F–ROM (EDIT)         = 0 (0:NO 1:YES)
REJECT LANGUAGE               = 0 (0:NO 1:YES)
SIGNAL ANALYSIS DISPLAY MODE  = 0 (0:GRAPHIC 1:TEXT)
SPECIFY NC WINDOW FORMAT      = 0 (0:AUTO 1:MANUAL)
NC WINDOW FORMAT (TOOL DATA)  = 0 (0:EXPAND 1:STANDARD)


[ NO   ] [ YES  ] [       ] [       ] [       ]
```

NB NB2
(K17. 1, K900.1)
(K17. 2, K900.2)
(K17. 4, K900.4)
(K17. 5, K900.5)
(K17. 6, K900.6)
(K17. 7, K900.7)
(K18. 0, K901.0)
(K18. 1, K901.1)
(K18. 2, K901.2)
(K18. 3, K901.3)
(K18. 7, K901.7)
(K19. 0, K902.0)
(K19. 1, K902.1)

* The bracketed addresses show the related KEEP RELAYs.

SIGNAL TRIGGER ENABLE

Displayed in case of PMC–NB(4047).

Stop function of ladder diagram display by trigger of signal is set.

The trigger stop function can be used by selecting “YES” ,and turning off and on the power.

WRITE TO F–ROM (EDIT)

Setting to write the LADDER data in F–ROM, when the edit of LADDER ends.

When you select “YES” and then get out of the EDIT screen, a message confirming if you write to F–ROM is displayed.

REJECT LANGUAGE

It is setting of the start of the program of C language.

When “YES” is selected, the program of C language is not started.

SIGNAL ANALYSIS DISPLAY MODE

The display form in the signal waveform display function is set.

The display form can be selected.

Select “TEXT” and it is displayed by the character.

Select “GRAPHIC” and it is displayed by the line.

SPECIFY NC WINDOW FORMAT

The form in functional instruction WINDR and WINDW are set.

When “AUTO” is selected, the format is automatically distinguished by the state of bit 4 of NC parameter 7401.

When “MANUAL” is selected, the format is selected by “NC WINDOW FORMAT (TOOL DATA)”.

NC WINDOW FORMAT (TOOL DATA)

The format in functional instruction WINDR and WINDW are set.

When “MANUAL” is selected by “SPECIFY NC WINDOW FORMAT”, this item is effective.

The window instruction of a new format can be used by selecting “EXPAND”.

(The same meaning as bit 4 of NC parameter 7401 is 1.)

An old window instruction can be used by selecting “STANDARD”

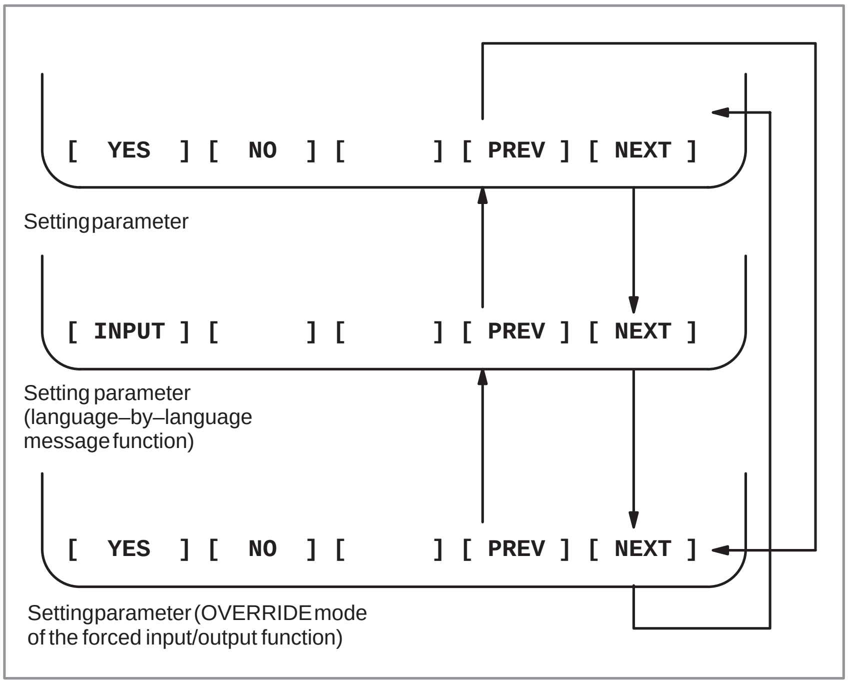
(The same meaning as bit 4 of NC parameter 7401 is 0.)

4.4.1 Other Setting Screens

Enable or disable the following function and mode:

- Language–by–language message function
- OVERRIDE mode of the forced input/output function

(1) Setting screen display
Each setting screen can be displayed by pressing the [NEXT] or [PREV] soft key on the setting parameter screen.



Each setting parameter can be set when the respective conditions are satisfied.

(a) Language–by–language message function

- The PROGRAM ENABLE setting parameter is set to YES (bit 1 of K17 or bit 1 of K900 is set to 1).
- The PMC model is SB6.

(b) OVERRIDE mode of the forced input/output function

- The PROGRAM ENABLE setting parameter is set to YES (bit 1 of K17 or bit 1 of K900 is set to 1)
- The PMC model is SA5, SB5, or SB6.
- The editing function is provided.

(2) Setting operations

(a) Setting the language–by–language message function
Position the cursor to the each setting item with the arrow keys, enter the desired data, then press the [INPUT] soft key.

```
PMC PRM (MESSAGE SHIFT)                   MONIT RUN

MESSAGE SHIFT VALUE          =       0
MESSAGE SHIFT START ADDRESS = A0000.0






[INPUT] [      ] [      ] [  PREV  ] [ NEXT ]
```

(i) Parameters

- MESSAGE SHIFT VALUE
  Enter a desired message display request bit shift amount.
  A value from 0 to 999 can be entered. The initially displayed value is 0.
  Entered data is maintained even after the power is turned off.
- MESSAGE SHIFT START ADDRESS
  Enter a shift start bit address in the message display request bit area.
  An address A value can be entered. The initially displayed value is A0.0.
  Entered data is maintained even after the power is turned off.

CAUTION
Data entered for MESSAGE SHIFT START ADDRESS is valid only when a value other than 0 is entered for MESSAGE SHIFT VALUE.

(b) OVERRIDE mode of the forced input/output function

- Set OVERRIDE mode by using the soft key or by entering 0 or 1 followed by the <INPUT> key.

```
PMC PRM (OVERRIDE)                        MONIT RUN

OVERRIDE ENABLE          = 0   (0:NO  1:YES)










[  NO  ] [  YES  ] [      ] [ PREV ] [ NEXT ]
```

OVERRIDE ENABLE 0 : OVERRIDE mode is disabled.
1 : OVERRIDE mode is enabled.

CAUTION
If the setting of this parameter is modified, the new setting becomes effective when the power is next turned on. After this parameter has been modified, the power must be turned off then back on.

4.5 NOTE

If you make a keyboard without cursor keys, you must move cursor by searching the address or so. In case of TIMER,COUNTER and KEEP RELAY, press the soft key [TIMER],[COUNTR] or [KEEPRL] after typing the address(Ex.1,2).
In case of Data Table Controlling Data, press the soft key [DATA](or [NO.SRH] if Data Table screen has already been displayed) after typing the group No.(Ex.3). In case of the Data Table, press the soft key [SEARCH] after typing the address in the Data Table screen which contains the address you want to search(Ex.4).

Ex.1) In case of setting the TIMER NO.11(ADDRESS T20)

1 Press the soft key [TIMER] after typing T20(or T21;T can be omitted.).

2 Press the INPUT key after typing the value.

Ex.2) In case of setting PRESET and CURRENT values of the COUNTER NO.02(ADDRESS C04)

1 PRESET → Press the soft key [COUNTER] after typing C4 (or C5;C can be omitted).

CURRENT → Press the soft key [COUNTER] after typing C6 (or C7;C can be omitted).

2 Press the INPUT key after typing the value.

> **CAUTION**
> It is not the number(NO.) but the address(ADDRESS) that you type in searching.

Ex.3) In case of the ADDRESS,PARAMETER,TYPE and NO. OF DATA of the Data Table Controlling Data NO.002.

1 Press the soft key [NO.SRH] after typing 2, and the cursor is moved to the ADDRESS position.

2 Press the INPUT key after typing the ADDRESS(ex.D20;D must not be omitted), and the cursor is automatically moved to the next position(PARAMETER). The cursor is moved only by pressing the INPUT key.

3 In the same way, set the PARAMETER,TYPE and NO. OF DATA. If you finish setting the NO. OF DATA, the cursor is moved to the position(ADDRESS) in the same line.

Ex.4) In case of setting D22 in the Data Table of the group 2

1 Press the soft key [G.DATA] on the Data Table Controlling Data screen, and the Data Table screen is displayed.

2 Press the soft key [G–SRCH] after typing 2 on the Data Table screen, and the Data Table of the group 2 is displayed.

3 Press the soft key [SEARCH] after typing D22(D can be omitted).

4 Press the INPUT key after typing the value.

5 PMC LADDER DIAGRAM DISPLAY (PMCLAD)

Displaying the PMC ladder diagram on CRT/MDI panel is available. This ladder diagram display function offers functions effectively used for locating troubles in addition to the simple ladder diagram display.

The following functions are done using the soft keys.

(1) Search and display of optional relay coil on ladder diagrams.

(2) Ladder diagram dynamic display.

(3) Stop of ladder diagram display by trigger of signal (on or off).

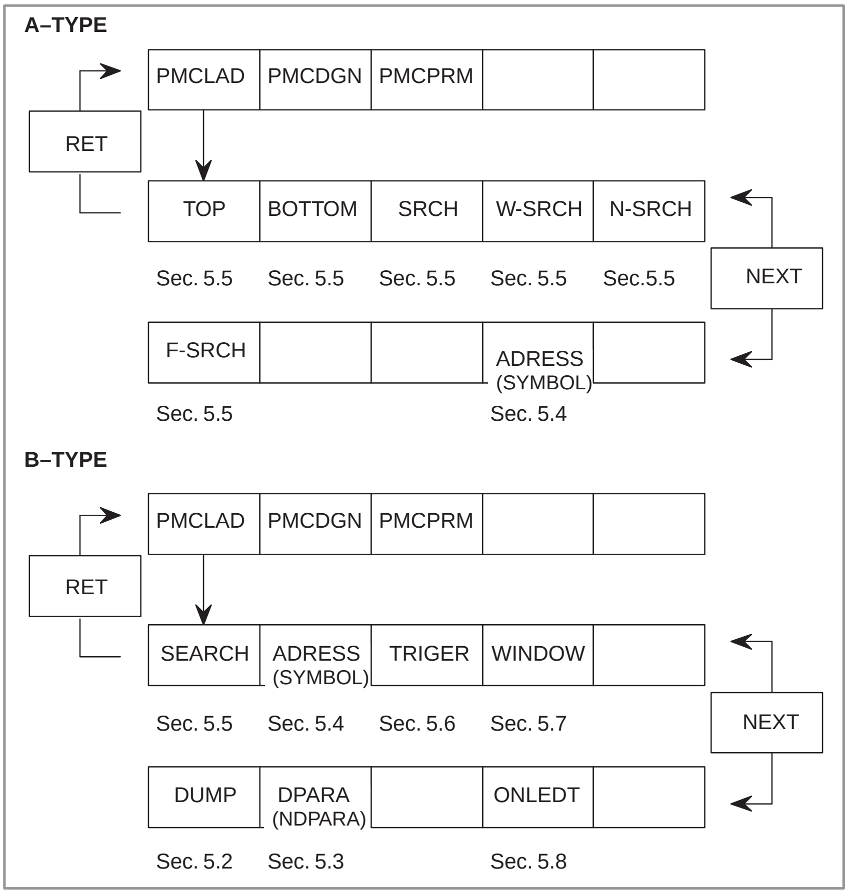
(4) Screen-dividing display.

(5) Monitor display of signal condition.

(6) Monitor display of parameter in functional instructions.

(7) ON LINE edit.

For this operation, depress [PMCLAD] soft key of PMC basic menu to bring the following menu.



5.1 LADDER DIAGRAM DISPLAY

The following functions can be done the ladder diagram display screen.

(a) Specified relay coil of ladder diagrams can be searched and displayed.

(b) Ladder diagram dynamic display.

The logical on-off states during a sequence program execution are displayed on a ladder diagram by changing the brightness in case of a monochrome CRT or by changing colors in case of a color CRT.

(1) Ladder diagram display

Press [PMCLAD] soft key, then the ladder diagram will be displayed. Eight relay contacts and relay coils in total are displayed in the horizontal direction of the CRT screen.

If the number of relay contacts exceed the above value, they are displayed in 2 or more lines.

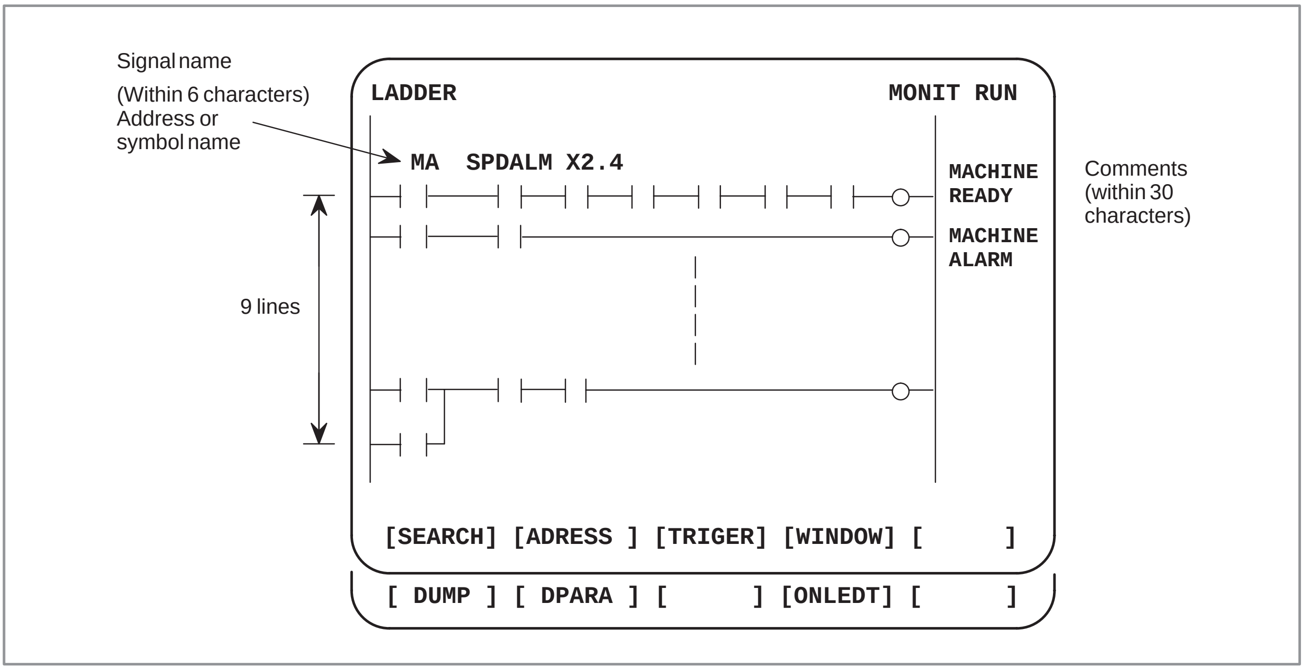


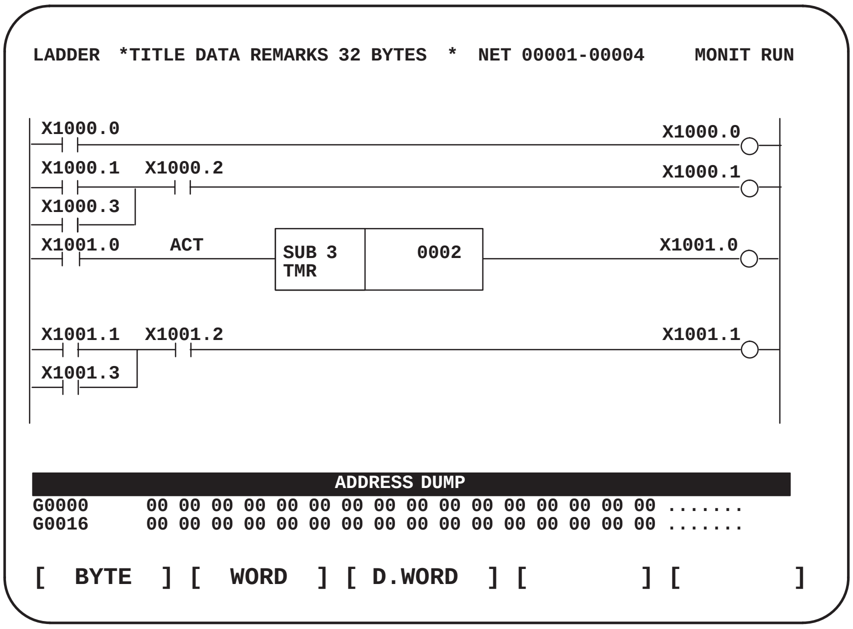
Fig. 5.1 Ladder diagram display

5.2 DUMP DISPLAY ON LADDER DIAGRAM

Ladder diagram and signal status dump can be displayed together.

The dump is displayed over 2 lines at the last line of ladder diagram by pressing the [DUMP] soft key.

PAGE↑↓ keys or [SEARCH] soft key is used for changing of PMC address.



The [DUMP] soft key has the following functions.

(1) [BYTE] : Byte type display (1 BYTE)

“G0000 00 14 00 00 01 00 00 00 00 00 00 00 00 00 00 00”

“G0016 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00”

(2) [WORD] : Word type display (2 BYTE)

“G0000 1400 0000 0001 0000 0000 0000 0000 0000”

“G0016 0000 0000 0000 0000 0000 0000 0000 0000”

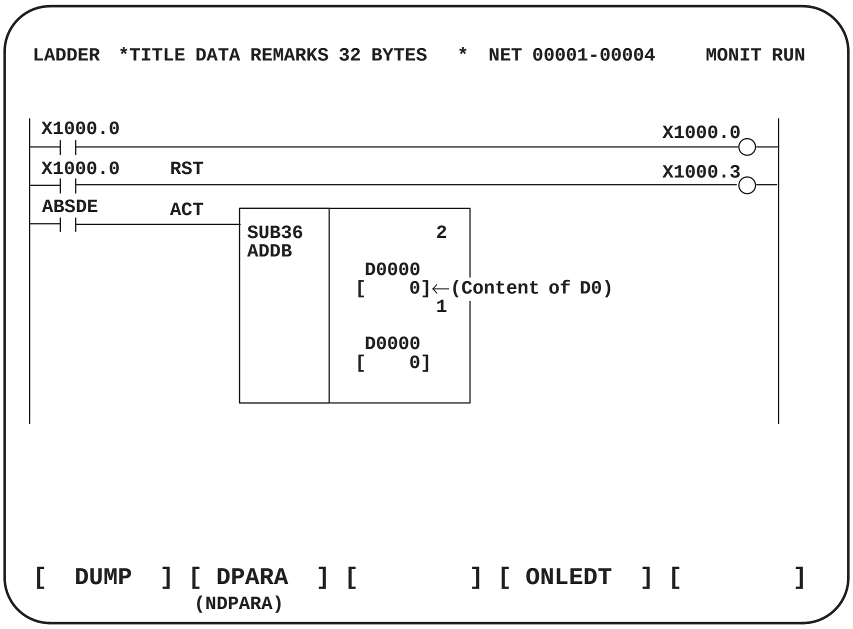
(3) [D.WORD] : Long word type display (4 BYTE)

“G0000 00001400 00000001 00000000 00000000”

“G0016 00000000 00000000 00000000 00000000”

5.3 PARAMETER DISPLAY ON LADDER DIAGRAM

The value of parameter of a functional instruction is displayed in the functional instruction of a ladder diagram.



The function of the soft key is as follows :

(1) [DPARA] : The value of parameter is displayed in functional instruction.

(2) [NDPARA] : The value of parameter is not displayed in functional instruction.

5.3.1 The Value of Functional Instruction Parameter

No.	Functional instruction	Data no. s	Data length of instruction parameter (1: byte, 2: word, 4: d. word)						Displaying form
			1	2	3	4	5	6	
1	END1	0							
2	END2	0							
3	TMR (NOTE3)	2	4	4					Binary
4	DEC	1	1						BCD
5	CTR (NOTE4)	2	2	2					Binary
6	ROT	3		2	2	2			BCD
7	COD	2		1	2				BCD
8	MOVE	2			1	1			HEX
9	COM	0							
10	JMP	0							
11	PARI	1	1						
12									
13									
14	DCNV	2	2	2					(Note 1)
15	COMP	2		2	2				BCD
16	COIN	2		2	2				BCD
17	DSCH	3		2	2	2			BCD
18	XMOV	3		2	2	2			BCD
19	ADD	3		2	2	2			BCD
20	SUB	3		2	2	2			BCD
21	MUL	3		2	2	2			BCD
22	DIV	3		2	2	2			BCD
23	NUME	1		2					BCD
24	TMRB (NOTE3)	1		4					Binary
25	DECB	2		1/2/4		1			
26	ROTB	4		1/2/4	1/2/4	1/2/4	1/2/4		Binary
27	CODB	2			1	1/2/4			
28	MOVOR	3	1	1	1				HEX
29	COME	0							
30	JMPE	0							
31	DCNVB	2		1/2/4	1/2/4				(Note1)
32	COMPB	2		1/2/4	1/2/4				Binary
33	SFT	1							HEX
34	DSCHB	4		1/2/4	1/2/4	1/2/4	1/2/4		Binary
35	XMOVB	4		1/2/4	1/2/4	1/2/4	1/2/4		Binary
36	ADDB	3		1/2/4	1/2/4	1/2/4			Binary
37	SUBB	3		1/2/4	1/2/4	1/2/4			Binary
38	MULB	3		1/2/4	1/2/4	1/2/4			Binary
39	DIVB	3		1/2/4	1/2/4	1/2/4			Binary
40	NUMEB	1			1/2/4				Binary
41	DISPB	0							
42	EXIN	1	4						HEX
43	MOVB	2	1	1					Binary
44	MOVW	2	2	2					Binary
45	MOVN	2	4	4					Binary
46									
47									
48	END3	0							
49	DISP	1			4				HEX
50	PSGNL	2	1	1					HEX
51	WINDR	1	2						Binary
52	WINDW	1	2						Binary
53	AXCTL	1		4					HEX
54	TMRC (NOTE3)	2		4	4				Binary

No.	Functional instruction	Data no. s	Data length of instruction parameter (1: byte, 2: word, 4: d. word)						Displaying form
			1	2	3	4	5	6	
55	CTRC (NOTE4)	2	2	2					Binary
56									
57	DIFU	0							
58	DIFD	0							
59	EOR	3		1/2/4	1/2/4	1/2/4			HEX
60	AND	3		1/2/4	1/2/4	1/2/4			HEX
61	OR	3		1/2/4	1/2/4	1/2/4			HEX
62	NOT	2			1/2/4	1/2/4			HEX
63	PSGN2	1	1						HEX
64	END	0							
65	CALL	0							
66	CALLU	0							
67									
68	JMPB	0							
69	LBL	0							
70									
71	SP	0							
72	SPE	0							
73	JMPC	0							
74									
87									
88	MMC3R	4	2	2	2	2			Unsign
89	MMC3W	4	2	2	2	2			Unsign
90	FNC90	1	2						Binary
91	FNC91	1	2						Binary
92	FNC92	1	2						Binary
93	FNC93	1	2						Binary
94	FNC94	1	2						Binary
95	FNC95	1	2						Binary
96	FNC96	1	2						Binary
97	FNC97	1	2						Binary
98	MMCWR	2	2	2					Unsign
99	MMCWW	2	2	2					Unsign

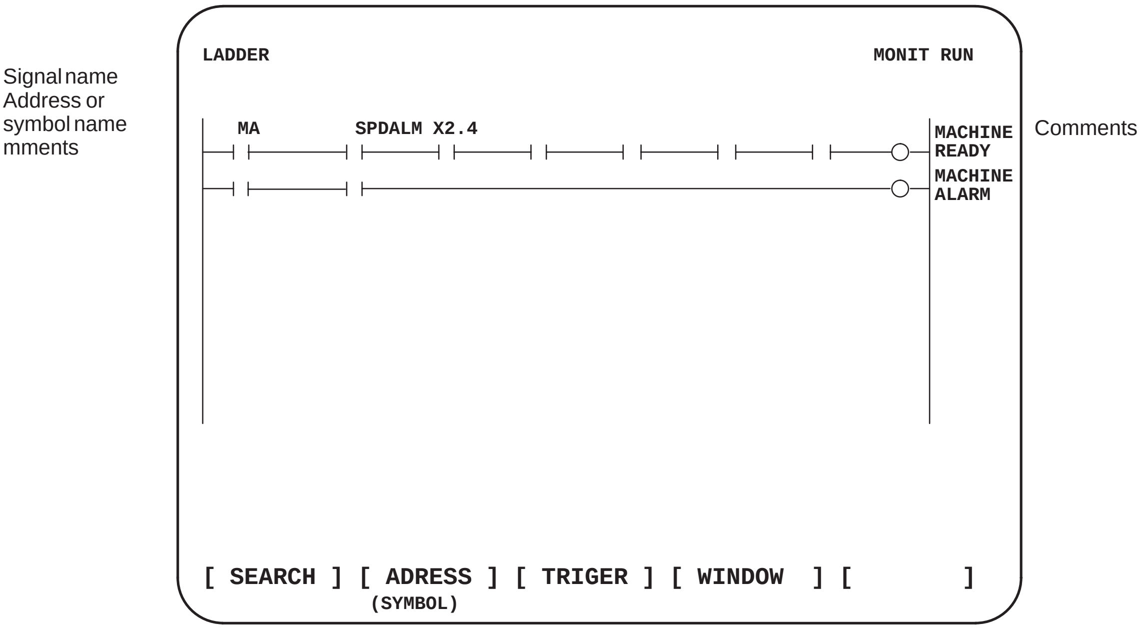
NOTE

1 The data length of BCD is displayed for 1 is 2-figures, 2 is 4-figures.
2 The value of parameter is not displayed in this instruction.
3 The timer is displayed the content of timer number (3: TMR, 24: TMRB, 54: TMRC).
4 The counter is displayed the content of counter number (5: CTR, 55: CTRC).

5.4 SYMBOL AND COMMENT DISPLAY

If symbol data and comments are defined to the PMC address, a comment is displayed for symbol display and relay coil.

By pressing soft key [ADRESS], the symbol displayed relay is address-displayed.
By pressing soft key [SYMBOL], the symbol displayed relay is symbol-displayed.
(See III. PMC programer, 5. 4 Symbol data setting)



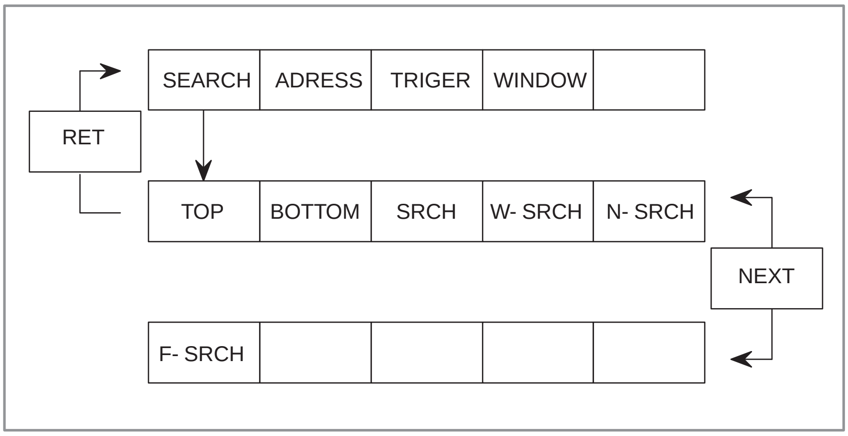
(1) [ADRESS] : is used to display the address name.

(2) [SYMBOL] : is used to display the symbol name.

5.5 SEARCH OF SPECIFIED RELAY COIL POINTS IN LADDER DIAGRAM

Specified relay coil points of ladder diagrams can be displayed on the screen.

For this operation, press [SEARCH] soft key to bring the following menu.



The function of the soft key is as follows :

(1) [TOP] : Displays the first NET of the ladder from the beginning of the screen.

(2) [BOTTOM] : Displays the last NET of the ladder from the beginning of the screen.

(3) [SRCH] : When the address and bit number or symbol name to be searched are typed in and the [SRCH] key is pressed, the specified address or symbol is searched from the top of the current screen. If the specified relay cannot be found until the last NET of the ladder, the relay are searched again from the first ladder until the NET where they started being searched.

(4) [W-SRCH] : This is used for searching a relay coil. Press [W-SRCH] soft key after keying in an address and bit number or symbol name. If the same address and bit number or the same symbol name is detected, the screen containing it will be displayed.

(5) [N-SRCH] : Displays the ladder with the specified NET number from the beginning of the screen. Moreover, when pressing the [N-SRCH] key without keying the NET number, the display is scrolled down by one NET.

(6) [F-SRCH] : When the functional instruction name or functional instruction number is typed in and the [F-SRCH] key is pressed, the functional instruction is searched.

5.6 STOP OF LADDER DIAGRAM DISPLAY BY TRIGGER OF SIGNAL

○ : Can be used
× : Cannot be us

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2
×	○	×	×	○	○	×	×	○	○	○	○	Δ	Δ	○	○	○

NOTE
Δ: Can be used for the specific series of CNC
(Series 16 : B005/11 to, B105/08 to, B305/04 to, B009/03 to, All serieses of model C)
(Series 18 : BD03/12 to, BE03/09 to, BG23/03 to, BG03/06 to, BD09/02 to, BE09/14 to, All serieses of model C)
PMC–PA3 can be used only with Power Mate–H.

The ladder display can be stopped by manual operation or trigger of signal.

The former ladder diagram display renews signal status every moment. But by using this function, all the ladder diagram at the specified moment can be checked.

The stop conditions as a trigger are specified by rising or falling edge detection of the designated signal.

* Display of setting trigger

The setting address, condition and counter are displayed at the title line.

"MODE : ON : X0000.0 : 0 : 0001 "

COUNT: Trigger checking number (default 1)

POINT : Trigger checking point (default 0)
0 —— the top of the 1st level
1 —— after END1 execution
2 —— after END2 execution
3 —— after END3 execution

ADR : Trigger setting address

ON : Rising edge detection (TRGON) ,
OFF : Falling edge detection (TRGOFF)

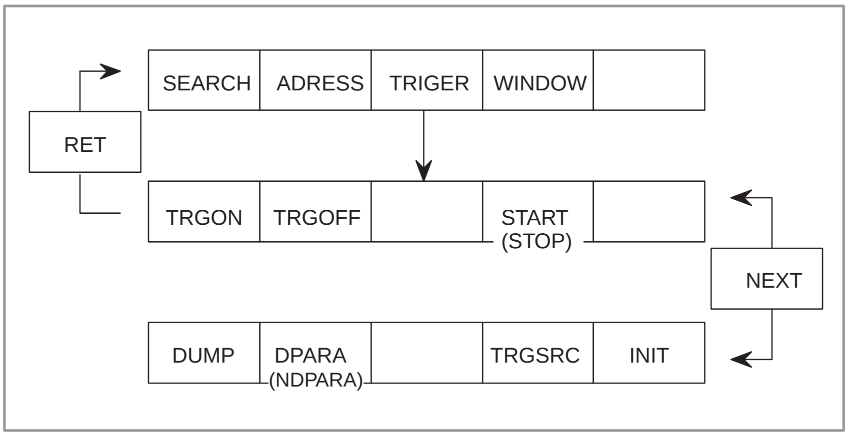
* Setting form adr ; p1 ; p2 + [TRGON/TRGOFF] soft key
Note) " ; " = "EOB"

adr (trigger address) ; p1 (trigger point) ; p2 (trigger checking number (1 to 65535))

* Because parameters are stored in the nonvolatile memory, they are not lost even if the power is turned off.

When bit 2 of keep relay K18 is set to 1 after parameters for sampling are specified, the trigger function automatically starts when the power is turned on.

For this operation, press [TRIGER] soft key to bring the following menu.



The function of the soft key is as follows :

(1) [TRGON] : Trigger is set on condition that the ladder status stops when the status of designated signal is rising.

(2) [TRGOFF] : Trigger is set on condition that the ladder status stops when the status of designated signal is falling.

(3) [START] : Change start/stop of trigger execution. While this function is executing, “TRG” is blinking.

(4) [TRGSRC] : Search and blink the instruction stopped by trigger.

(5) [INIT] : The setting of trigger is initialized.

5.7 DIVIDING DISPLAY OF LADDER DIAGRAM

This function is used for dividing display of ladder diagram.

The maximum number of division is 6.

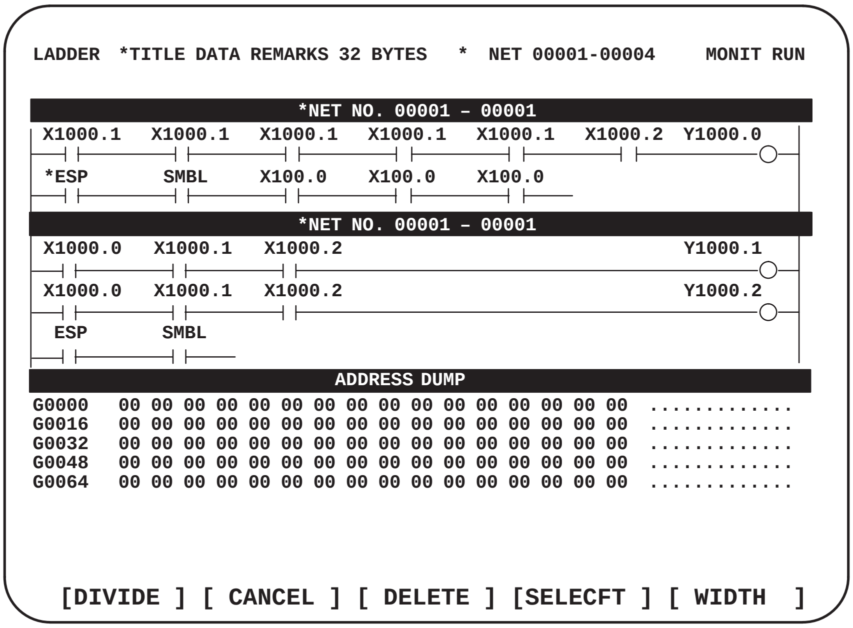
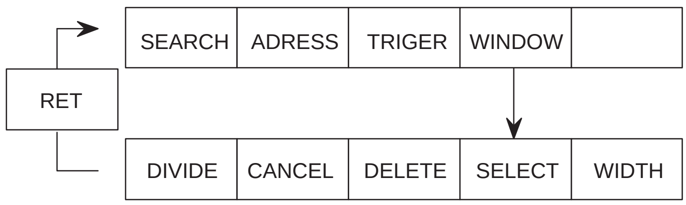


Fig. 5.7 Dividing display of ladder diagram

> **NOTE**
> For DUMP display, dump screen is displayed at the last part of screen.

For this operation, press [WINDOW] soft key to bring the following menu.



The function of the soft key is as follows :

(1) [DIVIDE] : The screen will be divided.
The dividing display of ladder diagram can be displayed for the designated NET number.
(NET number + [DIVIDE])

(2) [CANCEL] : The dividing display of ladder diagram display ends.
(The screen returns to normal display.)

(3) [DELETE] : The screen division subject to operation is ended.

(4) [SELECT] : Change the screen subject to division operation.
The screen in operation is displayed by “purple” title line, another screen is displayed by “blue” title line.
In monochrome CRT, the screen is displayed by changing brightness.

(5) [WIDTH] : Change the width of division by using [EXPAND] or [SHRINK] soft key.

(6) [EXPAND] : The divided screen is expanded.

(7) [SHRINK] : The divided screen is shrank.

5.8 ON–LINE EDIT

○ : Can be used
Δ : Option
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2
×	Δ	×	×	○	○	×	×	Δ	Δ	○	○	Δ	Δ	○	○	○

> **NOTE**
> Δ: Can be used for the specific series of CNC
> (Series 16 : B005/11 to, B105/08 to, B305/04 to, B009/03 to, All serieses of model C)
> (Series 18 : BD03/12 to, BE03/09 to, BG23/03 to, BG03/06, BD09/02 to, BE09/14 to, All serieses of model C)
> PMC–SA3 is usable only with the Power Mate–H.

For the PMC MODEL PA, SA series and –SB series, the editing card (module) is necessary.

When bit 1 in the keep relay K17 is 1, this function is available and [ONLEDT] soft key is displayed.

When the ladder program is executing, a part of the ladder program can be changed.

- Change the type of contact (A contact, B contact)
- Change address of contact and coil.
- Change address parameter of functional instruction.

This function don’t change the size.
(Cannot be Addition, deletion and changable data size)

When bit 3 of keep relay K18 is set to 1, the results of online editing are automatically reflected on the ladder program for editing. When bit 3 of keep relay K18 is set to 0, reflect the results of online editing on the ladder program for editing, using the COPY function for the I/O screen. Otherwise, the results of editing will be lost upon power–off. For the Moreover, when the CNC being used is the Series 15–MODEL B, Series 16/18–MODEL B/C, Series 21/210–MODEL B, Series 16*i*/18*i*/21*i*–MODEL A, or Power Mate–MODEL H, write to flash ROM.

How to store the results of editing

PMC other than NB		Press the COPY key on the I/O screen.
NB	Without DRAM	Write the program into FROM.
	With DRAM	Press the COPY key on the I/O screen. Write the program into FROM.

Operation

Press the [ONLEDT] soft key to enable the editing of a ladder program. The editing procedure is the same as that using the programmer function, described in Part III.

6 USER PMC SCREEN (PMCMDI)

6.1 FOR THE FS16 (PMC–SC OR PMC–SC3)

This user PMC screen is open to users, and it employs function key <CUSTOM>. It is applicable only when C language programming has been made. For details, see the PMC–SC/SC3/SC4/NB programming manual for C language (B–61863E–1).

NOTE
Pressing the <CUSTOM> key several times changes the screen to the PMCMDI screen. Because the <CUSTOM> key is also used to execute other functions.

6.2 FOR THE FS15 (PMC–NB)

This user PMC screen is open to users. To display this screen, display the PMC screen and press the OTHERS key or call the pl–pcmdi function in C language. It is applicable only when the program has been written in C language. For details, see the PMC–SC/SC3/SC4/NB programming manual for C language (B–61863E–1).

7 PMC–NB6 MANIPULATION

7.1 OVERVIEW

In the NC system, clicking the [PMC] soft key on the [SYSTEM] menu enables the setting and display of data related to the PMC. The following screens are used to specify and display the PMC–related data.

(1) Displaying PMC input/output signals and internal relay (PMCDGN)

(a) Title data screen

(b) Status screen

(c) Alarm screen

(2) PMC data setting and display (PMCPRM)

(d) Timer

(e) Counter

(f) Keep relay

(g) Data table

(3) Specifying PMC setting data (SETING)

(h) General setting data

(i) Setting data related to editing and debugging

(j) Online monitor parameter

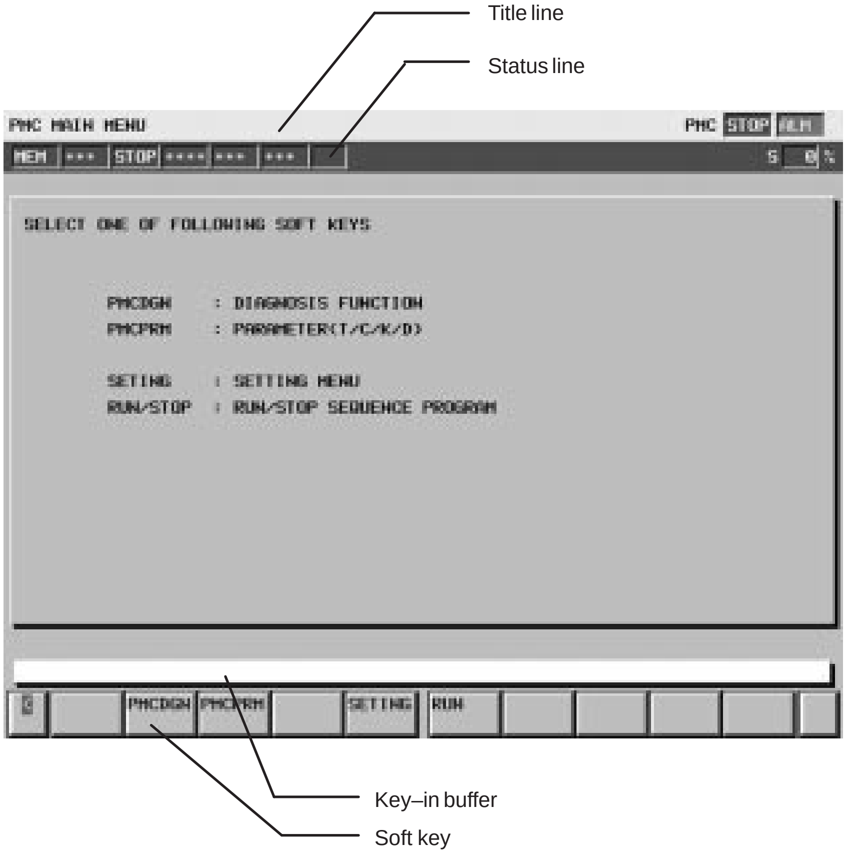
(4) Writing, reading, and collating sequence programs and PMC parameters (I/O)

* This is a PMC program restart function. For an explanation of its operation, see III.12, “PMC–NB6 Programmer.”

7.2 SOFT KEY–BASED PMC MENU SELECTION PROCEDURE

While the <SYSTEM> function key on the LCD/MDI is held down, clicking the [PMC] soft key displays the following PMC basic menu.

7.2.1 PMC Basic Menu



(1) Title line
This line displays the title of each PMC system screen.
It also displays the status of the PMC system at the right–hand end.

RUN STOP . . . . Whether the sequence program is running is indicated. [RUN] means that the sequence program is running. [STOP] means that the sequence program is at a halt.

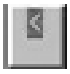
ALM . . . . This character string appears if a PMC alarm has occurred.

(2) Status line
This line displays NC information. The display is the same as that on the NC system screen.

(3) Key–in buffer
This area displays the data that was typed in.

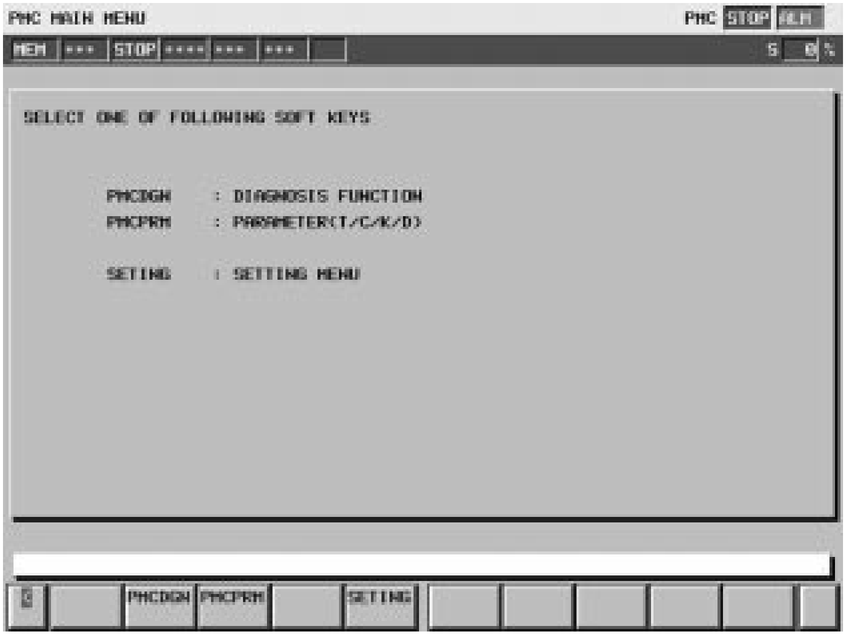
(4) Soft key

The soft key field consists of a soft key at both ends and ten soft keys in between. The left–end key has the following meaning:

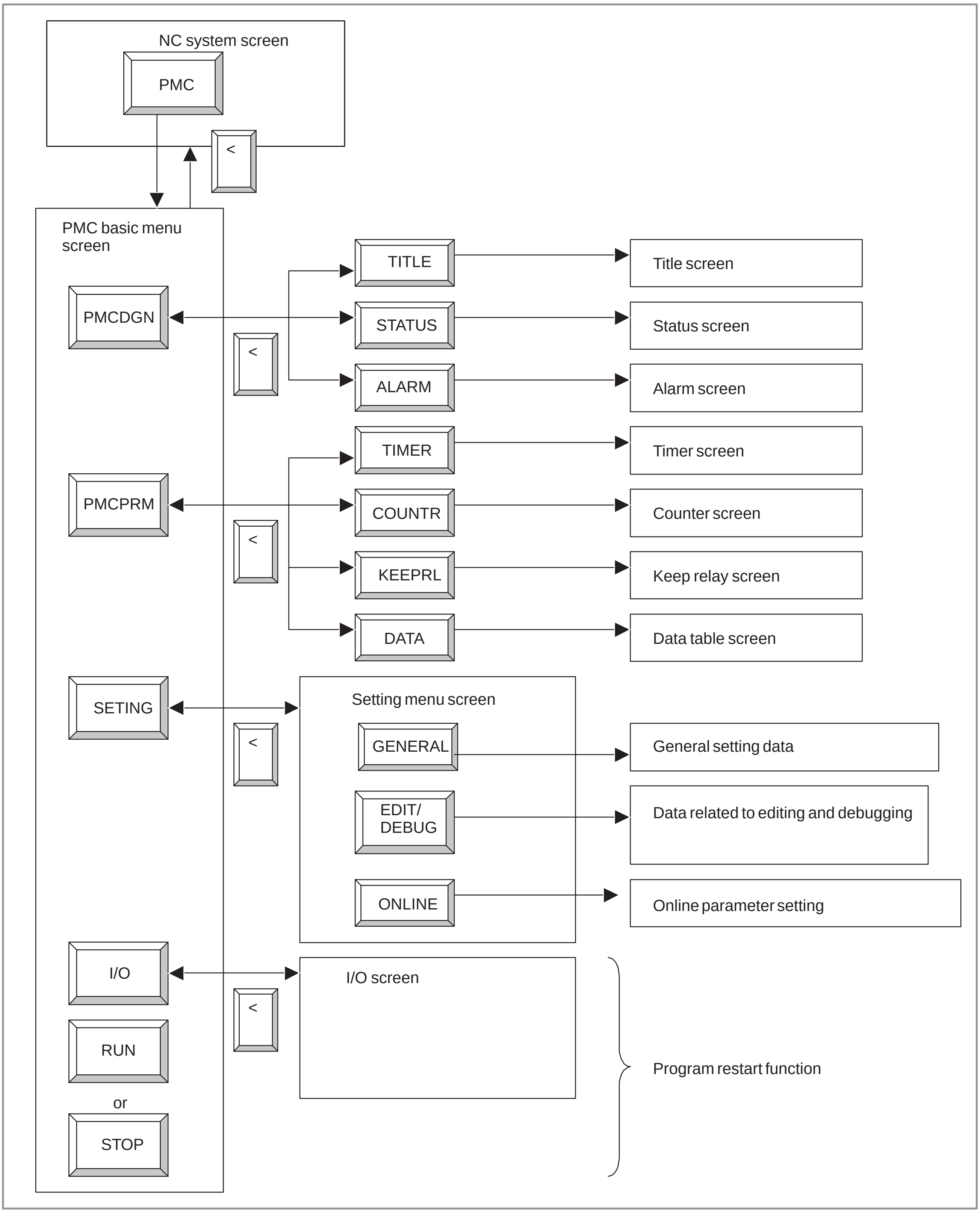


Return key . . Clicking this key returns you to the previous screen.

If the built–in debug function is disabled (bit 1 of K900 = 0), the basic menu appears as follows:



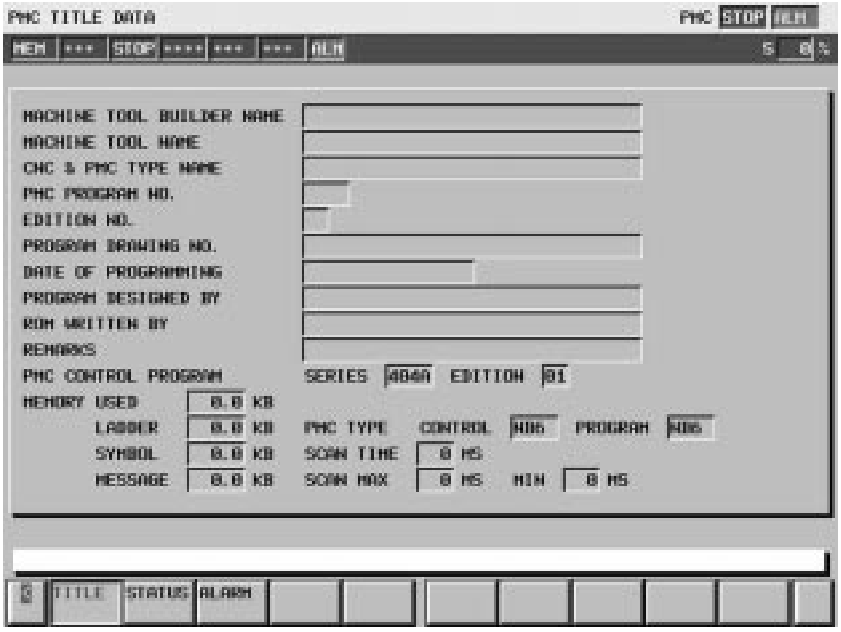
7.2.2
PMC Screen Transition and Related Soft Keys



7.3 DISPLAYING PMC INPUT/OUTPUT SIGNALS AND INTERNAL RELAY (PMCDGN)

7.3.1 Title Data Display (TITLE)

The title data corresponds to the title of a sequence program. It consists of the following items:



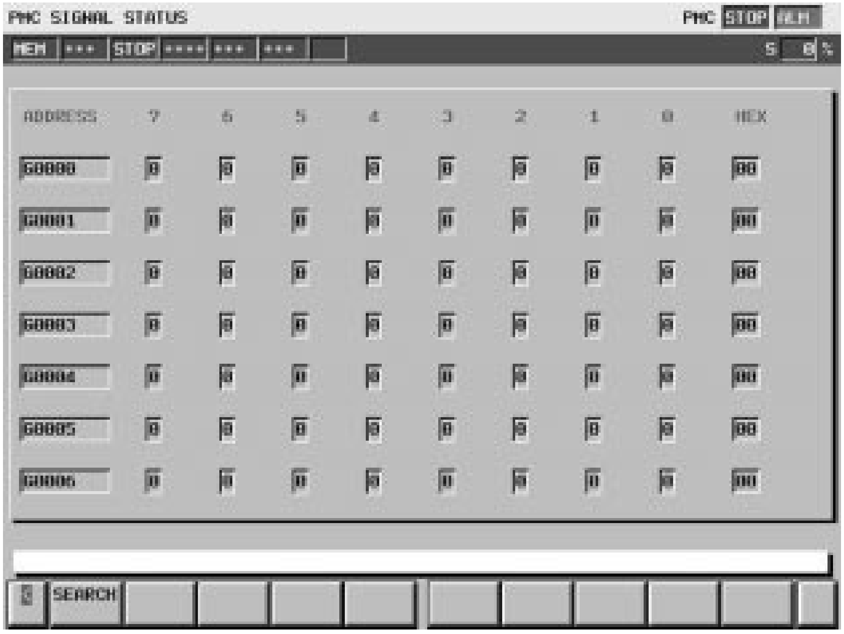
- Machine tool builder name (32 characters)
- Machine name (32 characters)
- NC/PMC type (32 characters)
- Sequence program number (4 characters)
- Edition (2 characters)
- Sequence program drawing (32 characters)
- Sequence program creation date (16 characters)
- Sequence program creator name (32 characters)
- ROM writer operator name (32 characters)
- Comment (32 characters)

In addition, the following data is displayed:

- PMC basic software series and edition
- Amount of memory occupied by each set of sequence data
- PMC basic software type and sequence program PMC type
- Current, maximum, and minimum execution time of the ladder program

7.3.2 Signal Status Display (STATUS)

This screen displays the contents at all the addresses (X, Y, F, G, R, A, C, T, K, D, M, and N) specified in programs. Each content display is a string of 0 and 1 with a hexadecimal indication at the right end.



Operating procedure

(1) Click the [STATUS] soft key. The screen shown above appears.

(2) Specify the desired address by keying it in, then click the [SEARCH] soft key.

(3) A sequence of data starting at the specified address is displayed as a bit pattern.

(4) To specify another address for display, click a cursor key, page key, or the [SEARCH] soft key.

7.3.3 Alarm Screen (ALARM)

If an alarm condition occurs in the PMC, clicking the [PMC] soft key from the NC system displays the following alarm message instead of the PMC basic menu. The soft keys displayed on this screen remain the same as on the PMC basic module screen. In addition, character string “ALM” appears on the title line.
If the alarm condition is fatal, no sequence program will be executed.

For an explanation of the alarm messages displayed on this screen, see APPENDIX M, “ALARM MESSAGE LIST.”

7.4 PMC DATA SETTING AND DISPLAY (PMCPRM)

7.4.1 Overview

This screen enables the entry and display of parameters for the timers, counters, keep relay, and data tables, which are held in nonvolatile memory. To use this procedure, first click the [PMCPRM] soft key on the PMC basic module.

7.4.2 Method for Entering PMC Parameters

Usually, no data can be entered for PMC parameters because they are protected. The following two methods can be used to make it possible to enter data for them.

- If the sequence program is running (RUN state) (usually, this method should be used when the machine is operating.)
  - i) Place the NC in MDI mode or bring it to an emergency stop.
  - ii) Set “PWE” on the NC setting screen to 1 (see the following table).
  - iii) Alternatively, set the program protect signal (KEY4) to 1 (only if counters or data tables are involved).
  - iv) The parameters are released from protection; so data can be entered for them (see the following table).

	PWE	KEY4
Timer	○	
Counter	○	○
Keep relay	○	
Data table	○	○

  - v) After entering data for the parameters, return “PWE” or the KEY4 signal to the previous state.
- If the sequence program can be stopped (STOP state), for example, while it is being debugged
  - i) Stop the sequence program.
  - ii) The parameter protection is released; so data can be entered for them.

WARNING
If a sequence program is stopped while the machine is operating, the machine may behave unexpectedly. Before stopping the sequence program, make sure that nobody is near the machine and that the tool cannot interfere with the workpiece or machine. Incorrect operation of the machine presents an extreme risk of death or serious injury to the user. Damage the tool, workpiece, and/or the machine is also likely.

An attempt to enter data for protected parameters causes the error message “WRITE PROTECT” to be displayed.

7.4.3 Continuous Data Entry

It is possible to enter data continuously on each of the timer, counter, keep relay, and data table screens. After continuous data entry, the cursor appears at the bottom of the most recently entered data.

- Entry

(1) Use “;” (EOB) as a data delimiter.

(Example) “100;200;300;” + “INPUT” key

(2) Use “;=” to enter the same value as the previous data.

(Example) Entering “100;=;=;200;=” + “INPUT” key causes the following data to be entered: 100,100,100,200,200

(3) Use “;;” to skip an address for entry.

(Example) Entering “100;;200;” + “INPUT” key causes the second data item to be skipped from entry.

7.4.4 Timer Screen (TIMER)

This screen is used to set and display the timer values for the machine instruction timers (SUB3).

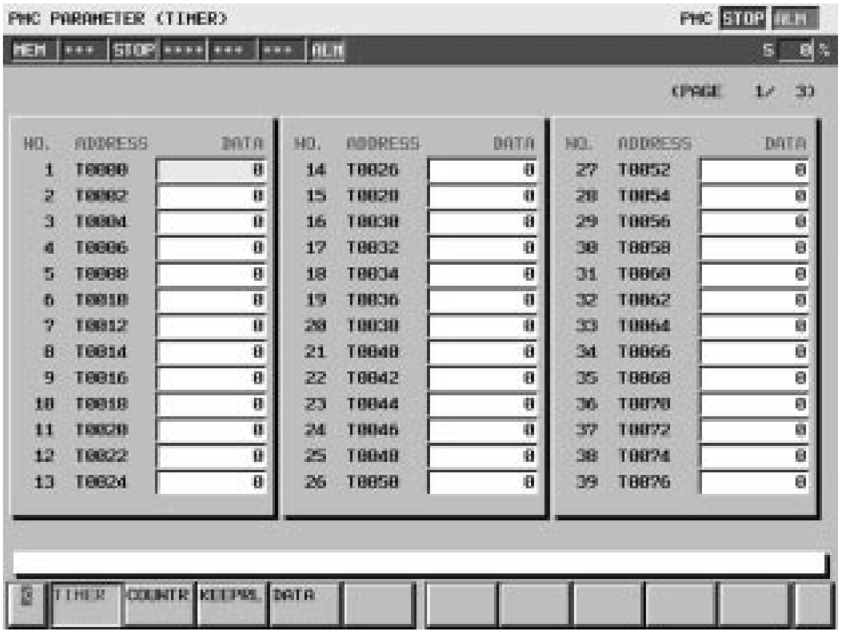


Table contents

- NO.: Counter number specified for a machine instruction counter
- ADDRESS: Address referenced by a sequence program
- DATA: Timer value

Timer number	Minimum timer value that can be specified	Maximum timer value that can be specified
1 to 8	48ms	1572.8 s
9 to 150	8ms	262.1 s

7.4.5 Counter Screen (COUNTR)

This screen is used to set and display the maximum and minimum counter values for machine instruction counters (SUB5).

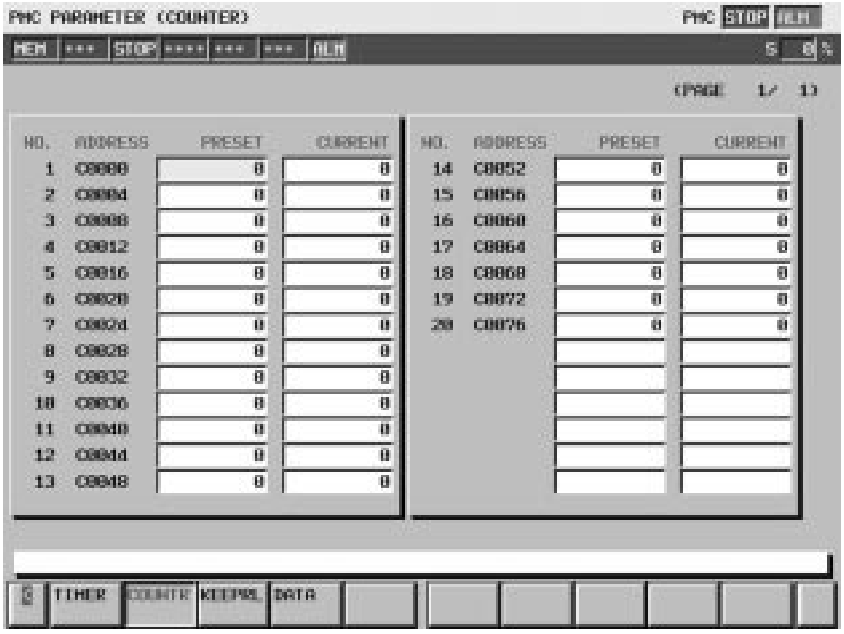


Table contents

- NO.: Counter number specified for a machine instruction counter
- ADDRESS: Address referenced by a sequence program
- PRESET: Maximum counter value (a minimum counter value is specified by a counter instruction)
- CURRENT: Current counter value

Counter types and maximum values

Counter type	PRESET maximum value	CURRENT maximum value
BINARY	32767	32767
BCD	9999	9999

7.4.6 Keep Relay Screen (KEEPRL)

This screen is used to set and display control data for the keep relay and nonvolatile memory control data.

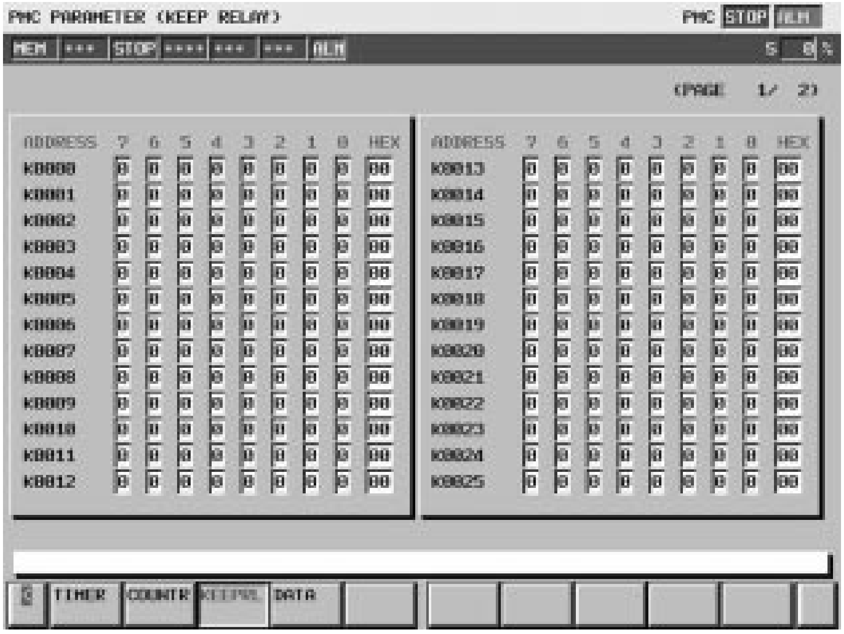


Table contents

- ADDRESS: Address referenced by the sequence program
- DATA (0 to 7): Contents (in bit representation)
- HEX: Contents (in hexadecimal representation)

The keep relay data is held in nonvolatile memory; it is not erased from memory even when the power is switched off.
Each PMC–NB6 area is as follows:

Area available to the user	K0 to K15 K17 to K39
Nonvolatile memory control address	K16
Area for use by management software(CAUTION)	K900 to K909

CAUTION
Area for use by management software
This keep relay is intended for use by PMC management software; the sequence program cannot use it. Reset all areas that are not to be used to 0.

(1) Nonvolatile memory control (MWRTF, MWRTF2) (address K16)
This control data is used to record the position of a movable machine part (such as a lathe turret) as coded data (such as BCD) in nonvolatile memory so that it can be held even after the power is switched off.

	#7	#6	#5	#4	#3	#2	#1	#0
K16	MWRTF2	MWRTF						

For details, refer to (4), “Nonvolatile memory control” in Section 6.1 of Chapter 6 “Nonvolatile Memory” of Part 1.

(2) Area for use by management software (addresses K900 to K909)

Model	NB&
PMC management software data 1	K900
Not used	K901
Not used	K902
Not used	K903 to K909

	#7	#6	#5	#4	#3	#2	#1	#0
K900	MWRTF2			MEMINP		AUTORUN	PRGRAM	LADMASK

#7 DTBLDSP 0 : The PMC parameter data table control screen is displayed.
1 : The PMC parameter data table control screen is not displayed.

#4 MEMINP 0 : Data cannot be entered on the signal status screen.
1 : Data can be entered on the signal status screen.
* The signal status screen is displayed by FAPT LADDER–II or the online function of the ladder editing package.

#2 AUTORUN 0 : The sequence program is executed automatically when the power is switched on.
1 : The sequence program is executed by clicking the sequence program execution soft key.

#1 PRGRAM 0 : The built–in debug function is not operated.
1 : The built–in debug function is operated.

#0 LADMASK 0 : Ladder dynamic display is used.
1 : Ladder dynamic display is not used.
* The ladder dynamic display status screen is displayed by FAPT LADDER–II or the online function of the ladder editing package.

CAUTION
Keep any unused portion of the area for use by the PMC management software reset to 0.

7.4.7 Data Table (DATA)

There are two data table types (data table control data table and data table).

(1) Data table control data screen (C.DATA)
Clicking the [DATA] soft key displays the data table control data setting screen for data table management.

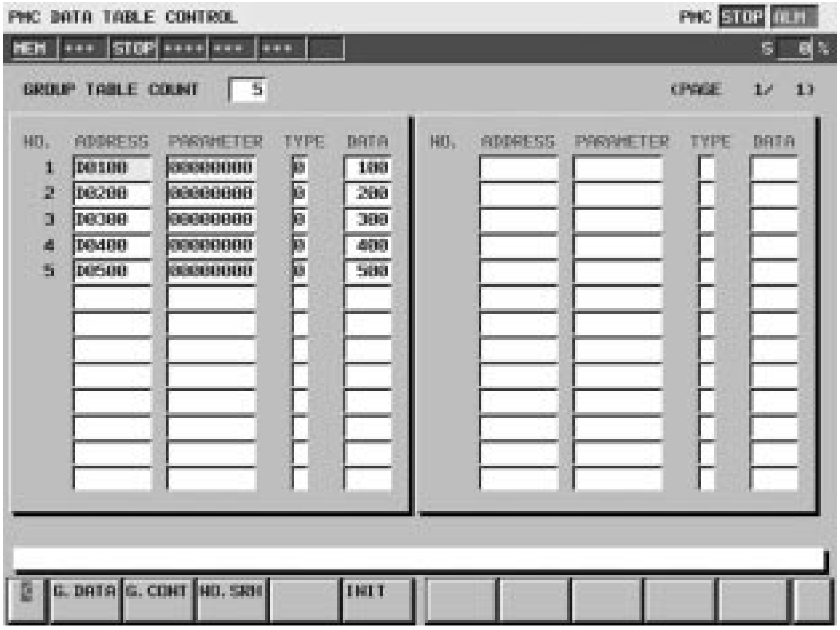


Table contents

- GROUP TABLE COUNT: Number of data items in the data table
- NO.: Group number
- ADDRESS: Data table start address (the same address can be specified for different groups.)
- PARAMETER: Table parameter(NOTE)
- TYPE: Data length (0 = 1 byte, 1 = 2 bytes, 2 = 4 bytes)
- DATA: Number of data items in each data table

Soft key definitions

[G.DATA]: Clicking this soft key switches to the screen for setting and displaying the data in the data table.

[G.CONT]: After the number of groups is entered, clicking this soft key asserts the number of groups for the data table.

[NO.SRH]: After a group number is entered, clicking this soft key moves the cursor to the specified group.

[INIT]: Clicking this soft key initializes the data table.

* The initial data is as follows:

NO.	ADDRESS	PARAMETER	TYPE	DATA
001	D0000	00000000	0	8000

NOTE

The meanings of the table parameters are as follows:

#7	#6	#5	#4	#3	#2	#1	#0

#0:
0: Binary format
1: BCD format

#1:
0: Input protection disabled
1: Input protection enabled

#2:
0: Binary or BCD format (with bit 0 valid)
1: Hexadecimal format (with bit 0 invalid)

(2) Data table screen

If the data table control data is specified, clicking the [G.DATA] soft key on the data table control data screen displays the data table setting screen.

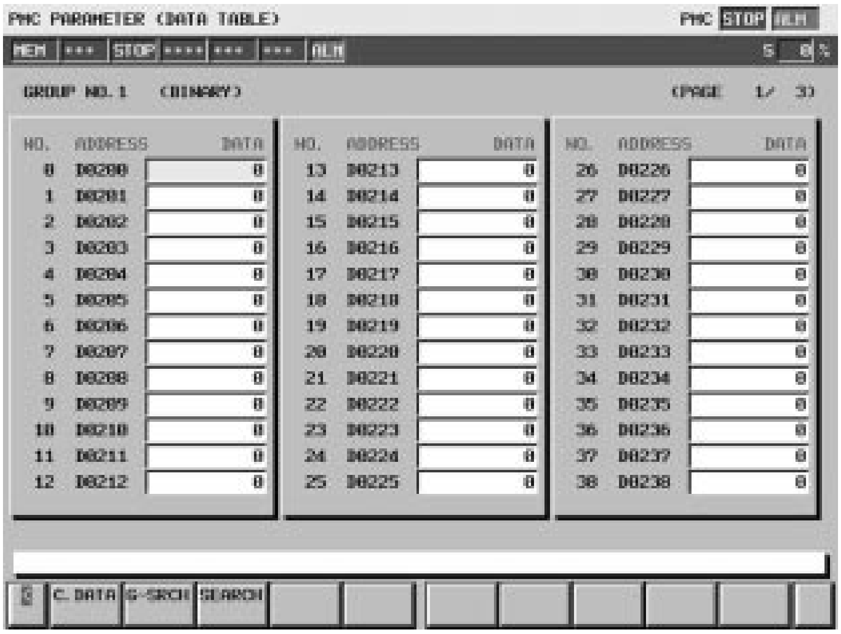


Table contents

- NO.
- ADDRESS: Address used by the sequence program
- DATA

Soft key definitions

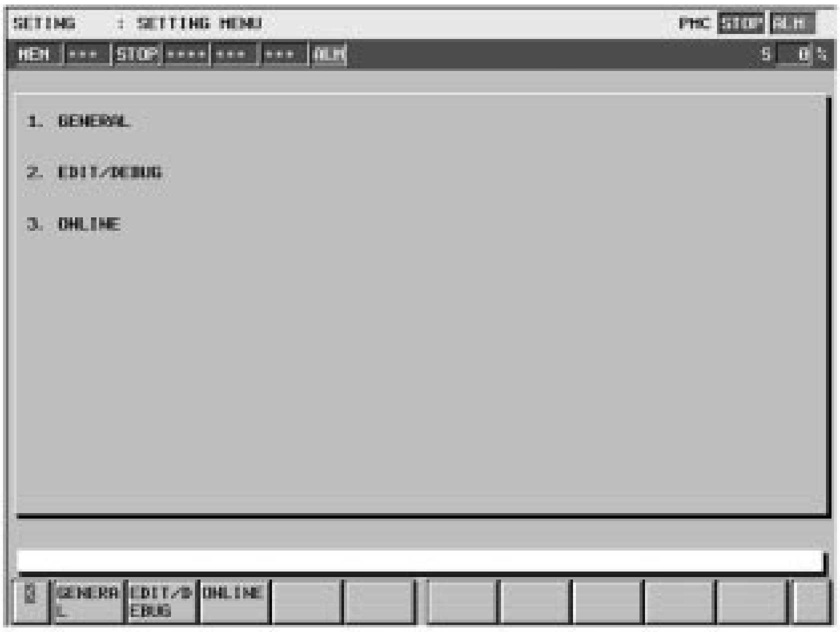
[C.DATA]: Clicking this soft key switches to the data table control data screen.

[G–SRCH]: After the entry of a group number for a data table to be searched in another group, clicking this key moves the cursor to the beginning of that group.

[SEARCH]: After the entry of an address, clicking this key moves the cursor to the specified address within the currently selected group. When entering the address, the "D" can be omitted from the beginning of the address. After entering "101" for example, clicking this key moves the cursor to data using D101.

7.5 SETTING MENU (SETING)

Clicking the [SETING] soft key on the PMC basic module screen displays the following setting menu screen.



Menu contents

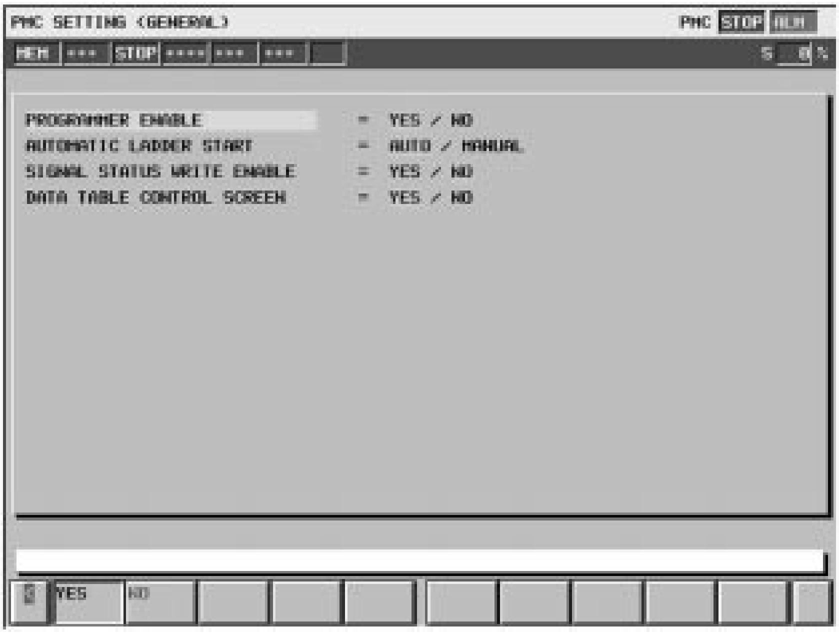
1. GENERAL: Screen for displaying general setting data
2. EDIT/DEBUG: Screen for displaying setting data related to editing and debugging
3. ONLINE: Screen for displaying the communication settings for the online–function. (It is displayed by setting “PROGRAMMER ENABLE” to “YES” on the GENERAL screen.)

Clicking the soft keys explained above displays the respective setting screens.

Some of the settings on this screen are saved to the keep relay. It is possible to prevent changes to that part of the settings on this setting screen, using a sequence program for writing to the keep relay.

7.5.1 Screen for Displaying General Settings (GENERAL)

Clicking the [GENERAL] soft key displays the following screen.



- PROGRAMMER ENABLE

  YES: The built–in debug function is used.

  NO: The built–in debug function is not used.

- AUTOMATIC LADDER START

  AUTO: The sequence program is executed automatically when the power is switched on.

  MANUAL: The sequence program is executed by clicking the sequence program execution soft key.

- SIGNAL STATUS WRITE ENABLE

  YES: The online function can be used to enter data on the signal status screen.

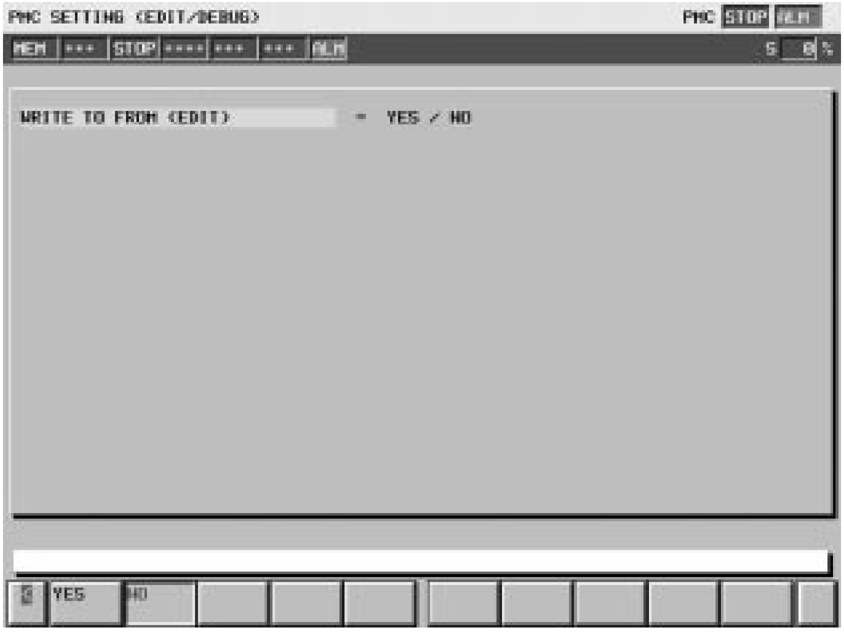
  NO: The online function is prevented from entering data on the signal status screen.

- DATA TABLE CONTROL SCREEN

  YES: The PMC parameter data table control screen is displayed.

  NO: The PMC parameter data table control screen is not displayed.

7.5.2 Screen for Displaying the Setting Data Related to Editing and Debugging



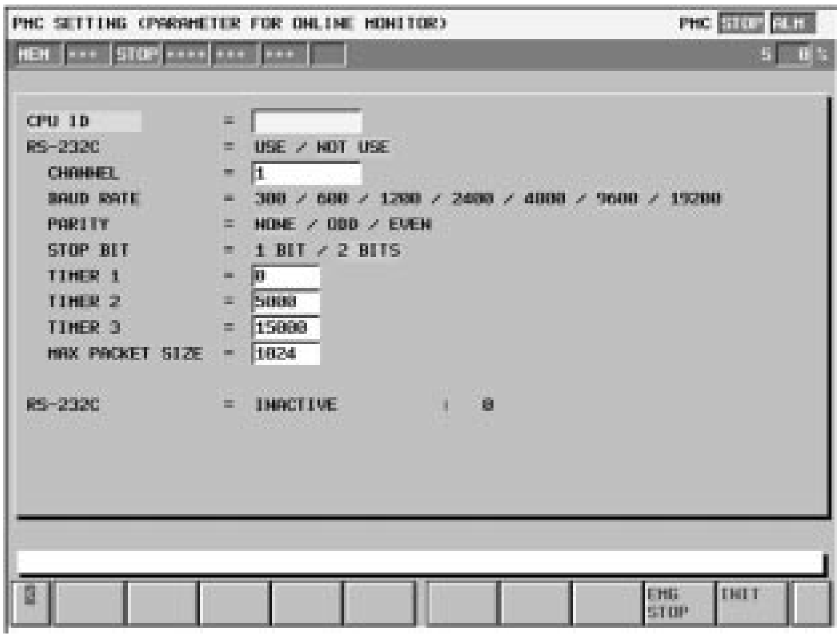
- WRITE TO FROM (EDIT)

  YES: The ladder program is automatically written to F–ROM after editing.

  NO: The ladder program is not automatically written to F–ROM after editing.

7.5.3 Screen for Setting/Displaying Online Monitor Parameters (ONLINE)

If PROGRAMMER ENABLE is set to YES on the GENERAL screen, the [ONLINE] soft key appears on the setting menu screen. Clicking this soft key displays the following screen.



Menu descriptions

- CPU ID
  The CPU ID value is displayed. The value can also be entered here, but its entry is usually not necessary.
- RS–232C (prompt)

  USE: An RS–232C port can be connected to FAPT LADDER–II.

  NOT USE: No RS–232C port is used.

  Note) If no RS–232C is to be connected to FAPT LADDER–II, select NOT USE.
- CHANNEL
  A channel number to be used is displayed. The number can also be entered.
- BAUD RATE

  300: A baud rate of 300 is specified.
  600: A baud rate of 600 is specified.
  900: A baud rate of 900 is specified.
  1200: A baud rate of 1200 is specified.
  2400: A baud rate of 2400 is specified.
  4800: A baud rate of 4800 is specified.
  9600: A baud rate of 9600 is specified.
  19200: A baud rate of 19200 is specified.
- PARITY

  NONE: No parity is specified.

  ODD: Odd parity is specified.

  EVEN: Even parity is specified.

- STOP BIT

  1 BIT: The number of stop bits is set to 1.

  2 BITS: The number of stop bits is set to 2.

- TIMER 1

  The value in communication parameter timer 1 is displayed. The value can also be entered, but its specification is usually not necessary.

- TIMER 2

  The value in communication parameter timer 2 is displayed. The value can also be entered, but its specification is usually not necessary.

- TIMER 3

  The value in communication parameter timer 3 is displayed. The value can also be entered, but its specification is usually not necessary.

- MAX PACKET SIZE

  The maximum packet size for the communication parameter is displayed. The size can also be entered, but its specification is usually not necessary.

- RS–232C (status display)

  The status of an RS–232C port is displayed.

  INACTIVE: No RS–232C port is in use.

  STOPPING: An RS–232C port is closed.

  STARTING: An RS–232C port is open.

  STAND–BY: An RS–232C port is waiting to be connected to FAPT LADDER–II.

  CONNECTED: An RS–232C port has been connected to FAPT LADDER–II.

If the ladder editing package is included in the system configuration, the F–BUS prompt menu appears above the RS–232C status display menu, and the F–BUS status display menu appears below the RS–232C status display menu.

- F–BUS (prompt)

  USE: An F–BUS port can be connected to the ladder editing package.

  NOT USE: No F–BUS port will be connected to the ladder editing package.

- F–BUS (status display)

  The status of an F–BUS port is displayed.

  INACTIVE: No F–BUS port is in use.

  STOPPING: An F–BUS port is closed.

  STARTING: An F–BUS port is open.

  STAND–BY: An F–BUS port is waiting to be connected to FAPT LADDER–II.

  CONNECTED: An F–BUS port has been connected to FAPT LADDER–II.

Soft key descriptions

[EMGSTOP]: Clicking this key causes communication to be terminated. It is used if it is impossible to terminate a connection due to abnormal communication.

[INIT]: Clicking this key initializes the parameter settings.

III. PMC PROGRAMMER (CRT/MDI)

1 GENERAL

This PMC programmer is used to set PMC system parameters and also generate and execute sequence programs by using soft keys on the CRT/MDI panel. For this operation, the PMC debugging RAM must be mounted in the CNC in advance.

For the CRT/MDI panel keys, refer to PMC operation in PARTII, Chapter 1 and 2.

1) Setting and display of PMC system parameters (SYSPRM)

   The following system parameters are available.

   a) Selection of counter data types (BCD or binary)
   b) Selection of division/non–division of ladder program (only PMC–SC)
   c) Parameters for executing C language programs (only for PMC–SC)

2) Editing of sequence programs (EDIT)

   The following editing functions are provided.

   a) Clear of memory
   b) Title data input
   c) Input, insert, search, and delete of sequence programs by ladder diagram format
   d) Input, insert, delete, and search of symbol data
   e) Address setting to each module when I/0 unit is used
   f) Message data input

3) Execution of sequence programs (RUN/STOP)

   The following function is provided to execute sequence programs

   a) Sequence program start and stop

4) To write, verify, and read of sequence programs and PMC data, and to write and read of I/0 sequence programs, followings are provided.

   a) Input/output of sequence programs to and from FANUC floppy disk cassette
   b) Input/output of sequence programs to and from debugging RAM
   c) Input/output of sequence programs to and from ROM
   d) Input/output of PMC parameter data to and from FANUC FD cassette

5) Displaying the contents of memory for the user C program and debugging the user C program (MONIT)

   a) Displaying the GDT map of the user C program
   b) Displaying memory information for the user C program
   c) Debugging the user C program

1.1 ABOUT THE FS15*i* PMC–NB6 PROGRAMMER

See Chapter 12, "PMC–NB6 Program Manipulation Screen" for an explanation of the FS15*i* PMC–NB6.

2 COMPONENT UNITS AND CONNECTIONS

This section describes only the 16/18–MODEL A. For other models, refer to the order list and the connection manual for each model.

The units required for generating a sequence program and connection methods are described below.

2.1 COMPONENT UNITS

1) PCB and module for PMC

This is PCB and module for PMC. The type of board is as follows;

a) Series 16

i) PMC–SB (Main CPU board)

- PMC control (A20B–2900–0560, –0143)
- Debugging control (A20B–2900–0530)
- PMC user ROM
- Editing module (A02B–0120–C160)

ii) PMC–RC (Option 3 board)

- PMC control module

A20B–2900–0390
(When using language programs, work RAM is required.)
A20B–2900–0391
A20B–2900–0143

b) Series 18

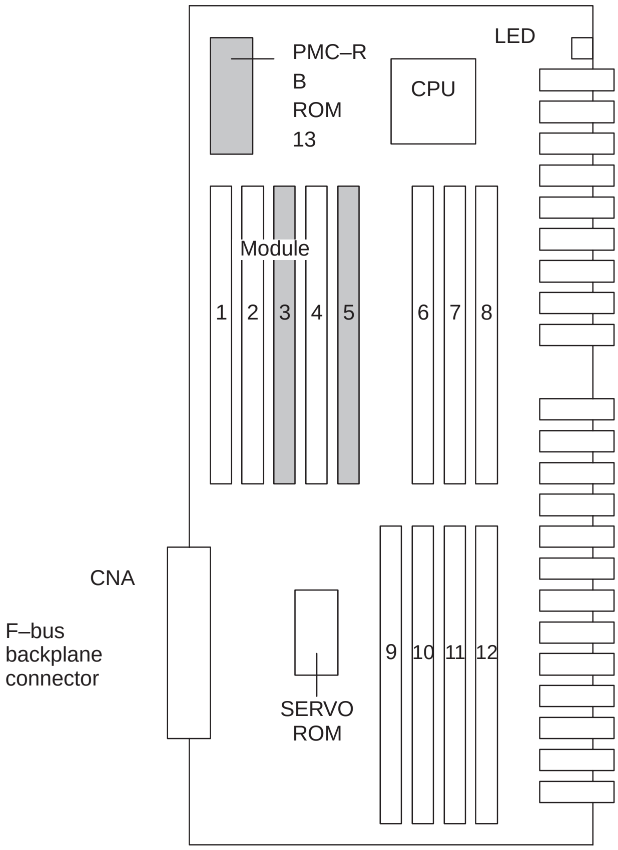
i) PMC–SA1/SA2 (Main CPU board)

- PMC control module (A20B–2900–0142) for PMC–SA1
  (A20B–2900–0920) for PMC–SA2
- Debugging RAM module (A20B–2900–0530)
- PMC user ROM
- Editing module (A02B–0120–0160)

Common with PMC–SB

Configuration of the main CPU board (Series 16)

Drawing number : A16B–2200–0900



Connector name	Connector name	Application
CRT	JA1	CRT video signal
MDI	JA2	MDI keyboard
R232–1	JD5A	RS–232–C serial port
R232–2	JD5B	RS–232–C serial port
MPG	JA3	Manual pulse generator
IOLINK	JD1A	FANUC I/O LINK
SPDL–1	JA7A	Serial spindle
A–OUT1	JA8A	Analog output
APCBAT	JA4A	APC battery
AMP1	JV1	1st axis servo amplifier
AMP2	JV2	2nd axis servo amplifier
AMP3	JV3	3rd axis servo amplifier
AMP4	JV4	4th axis servo amplifier
ENC1	JF1	1st axis pulse coder
ENC2	JF2	2nd axis pulse coder
ENC3	JF3	3rd axis pulse coder
ENC4	JF4	4th axis pulse coder
SCALE1	JF21	1st axis scale
SCALE2	JF22	2nd axis scale
SCALE3	JF23	3rd axis scale
SCALE4	JF24	4th axis scale

Fig. 2.1 (a) Layout of parts on Main CPU board (Series 16)

Table 2.1 (a) Modules of Main CPU board (Series 16)

No.	Module	Drawing number	Functional outline
1	ROM module	A20B–2900–0290 to 0293	ROM for CAP I or macros
2	ROM module	A20B–2900–0290 to 0292	ROM for the CNC system
3	SRAM module	A20B–2900–0530	RAM for debugging the PMC–SB
4	SRAM module	A20B–2900–0530,–0531 A20B–2900–0540,–0541	RAM for part programs and parameters
5	PMC control module	A20B–2900–0560 (For PMC–SB) A20B–2900–0143 (For PMC–SC)	PMC operation control
6	CRT control module	A20B–2900–0150 to 0152	CRT display control
7	System control module	A20B–2900–0101 to 0103	Clear, battery backup, spindle control, etc.
8	I/O interface module	A20B–2900–0110	MDI, MPG, RS–232–C, etc.
9	Servo control module	A20B–2900–0160	Digital servo control of the 3rd and 4th axes
10	Servo control module	A20B–2900–0160	Digital servo control of the 1st and 2nd axes
11	Servo interface module	A20B–2900–0370,–0380	3rd/4th axis amplifier/pulse coder interface
12	Servo interface module	A20B–2900–0370,–0380	1st/2nd axis amplifier/pulse coder interface

Configuration of the option 3 board (Series 16)

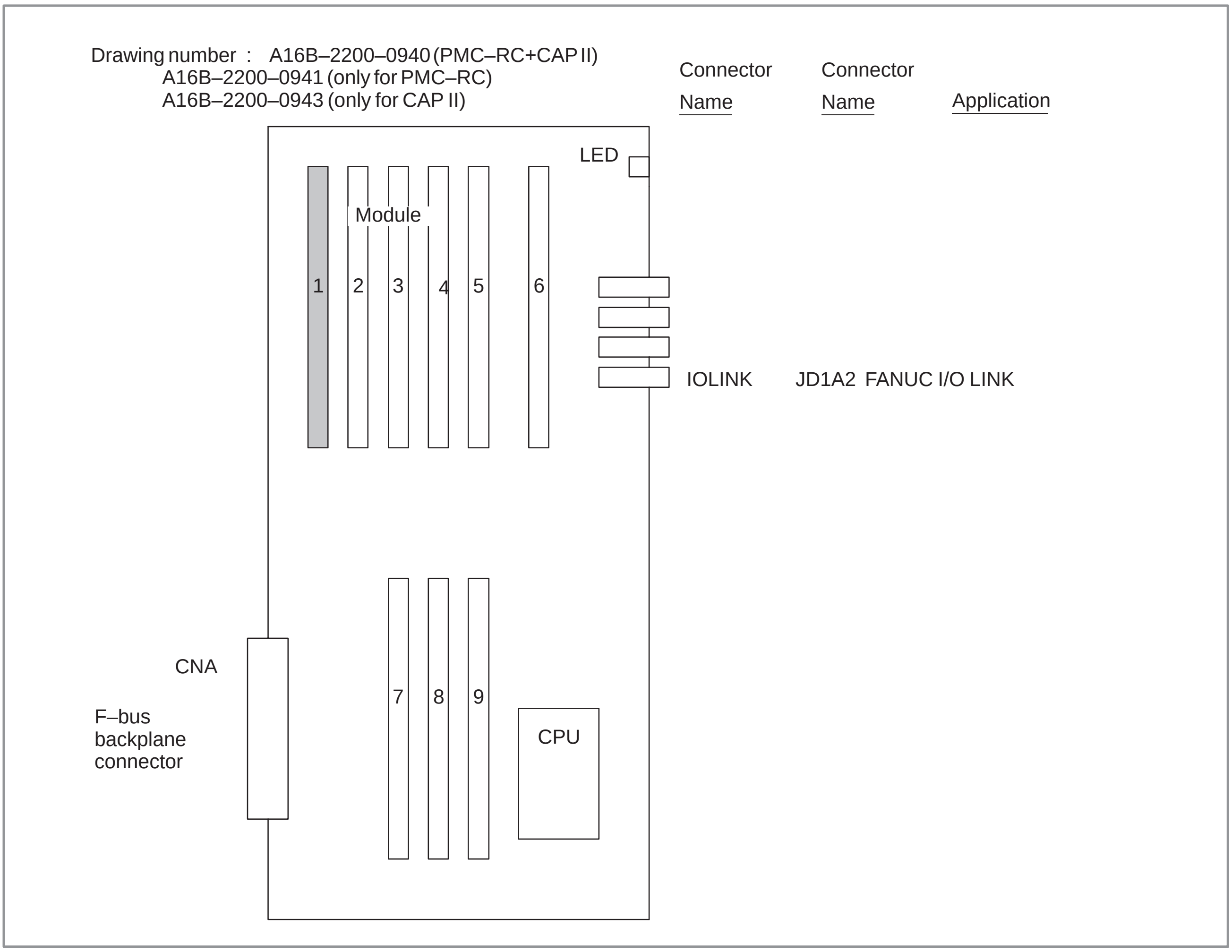


Fig. 2.1 (b) Layout of Parts on Option 3 Board (Series 16)

Table 2.1 (b) Modules of Option 3 Board (Series 16)

No.	Module	Drawing number	Functional outline
1	ROM module	A20B–2900–0290 to 0293	User ROM for PMC–SC (Mount the RAM module during debugging.)
2	ROM module	A20B–2900–0292	System ROM for PMC–SC
3	DRAM module	A20B–2900–0553	Work RAM for PMC–SC
4	PMC control module	A20B–2900–0560	PMC operation control and I/O Link control
5	PMC CPU module	A20B–2900–0390	For ladder capacity 2400 steps or C language
		A20B–2900–0391	Other than the above

Configuration of the Main CPU Board (Series 18)

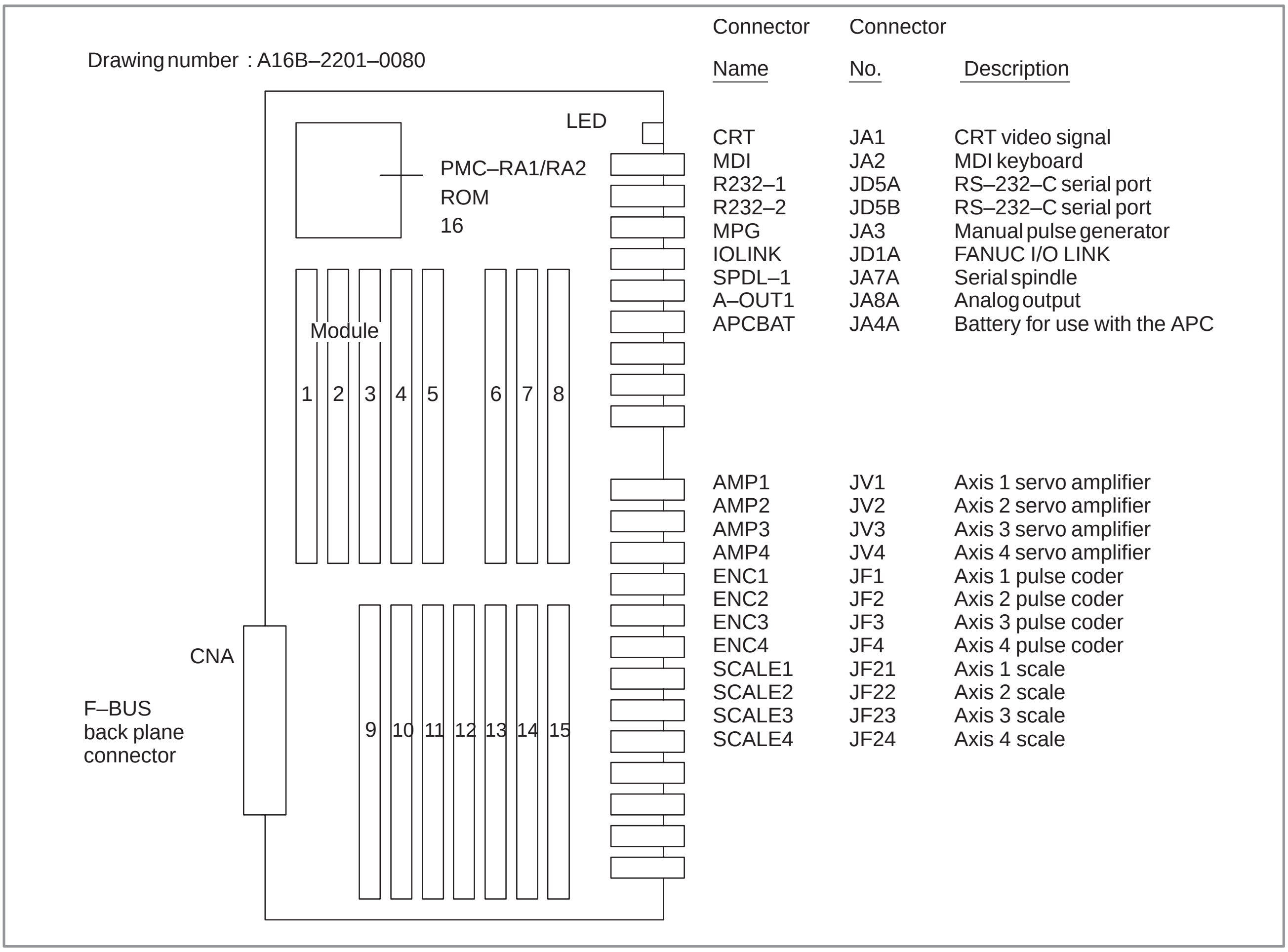


Fig. 2.1 (c) Parts layout for the main CPU board (Series 18)

Table 2.1 (c) Module list for the main CPU board (Series 18)

No.	Module name	Drawing no.	Function outline
1	ROM module	A20B–2900–0290 to 0293	ROM for macros or CAP 1
2	ROM module	A20B–2900–0290 to 0292	ROM for the CNC system
3	SRAM module	A20B–2900–0530	RAM for PMC–SA1/SA2 debug
4	SRAM module	A20B–2900–0530,–0531 A20B–2900–0540,–0541	RAM for parameters and tape memory
5	PMC control module	A20B–2900–0142 (PMC–SA1) A20B–2900–0920 (PMC–SA2)	PMC operation control
6	Main CPU module	A20B–2900–0930	FS18 Main processor
7	System control module	A20B–2900–0900 to 0902	Clear, battery backup, spindle control, servo/graphics software flash ROM
8	I/O interface module	A20B–2900–0110	MDI, MPG, RS–232–C
9	Graphics control module	A20B–2900–0310	Graphics display control
10	Graphics CPU module	A20B–2900–0590	Graphics control CPU
11	CRT control module	A20B–2900–0154 to 0156	CRT display control
12	Servo control module	A20B–2900–0160	Digital servo control for axes 3 and 4
13	Servo control module	A20B–2900–0160	Digital servo control for axes 1 and 2
14	Servo interface module	A20B–2900–0380	Amplifier, pulse coder, and interface for axes 3 and 4
15	Servo interface module	A20B–2900–0380	Amplifier, pulse coder, and interface for axes 1 and 2

2) Debugging RAM

This is used for debugging sequence programs. Since this debugging RAM memory is backed up by the battery, the memory data contents are not erased even when turning off the power supply.

> **CAUTION**
> If a RAM parity error occurs or when power is first turned on after installation, the RAM for debugging must be cleared.

(Procedure)

Turn on power to the CNC while pressing the X and O keys simultaneously. The contents of the RAM for debugging are then cleared.

3) Editing module

This is a built–in programmer for PMC–SA1, PMC–SA2, SA3, PMC–SB, PMC–SB2, or SB3 that enables editing sequence programs.

4) ROM

After debugging, write a sequence program into ROM.

5) ROM WRITER

This unit is used for writing or reading out a sequence program to ROM.

6) Offline programmer

This is used to transfer a sequence program.

By connecting the Offline programmer to PMC–SA1, –SA2, –SB, –SB2, –SB3, –SC, or –SC3, the storage of sequence programs in the floppy, and the output of a sequence program into printer can be done.

2.2 CONNECTING COMPONENT UNITS

(1) Connecting the debugging RAM module

a) PMC–SB, –SB2 and –SB3 : Connect the module to portion 3 shown in Fig. 2.1 (a).

b) PMC–SC and PMC–SC3 : Connect the module to portion 1 shown in Fig. 2.1 (b).

c) PMC–SA1, –SA2 and –SA3: Connect the module to portion 3 shown in Fig. 2.1 (c).

(2) Connecting the editing module for PMC–SA1, –SA2, –SA3, –SB, –SB2 and –SB3 Connect the module to portion 3 shown in Fig. 2.1 (a).

(3) Connecting ROM

a) PMC–SB, –SB2 and –SB3 : Connect EPROM to portion 13 shown in Fig. 2.1 (a).

b) PMC–SC and PMC–SC3 : Connect the ROM module to portion 1 shown in Fig. 2.1 (b).

c) PMC–SA1, –SA2 and –SA3: Connect EPROM to portion 16 shown in Fig. 2.1 (c).

○ : Enabled
Δ : Enabled depending on the option
× : Disabled

	SA1	SA2	SA3	SB	SB2	SB3	SC	SC3
RAM module	○	○	○	○	○	○	○	○
Editing module	○	○	○	○	○	○	×	×
EPROM	○	○	○	○	○	○	×	×
ROM module	×	×	×	×	Δ	Δ	○	○

NOTE

1 When 24,000 optional PMC–SB2 and PMC–SB3 ladder steps are available, 256K bytes of the ROM module can be used. In this case, connect the ROM module to portion 3 shown in Fig. 2.1 (a).

2 Either a RAM module, editing module, or ROM module can be connected to each board of PMC–SA1, –SA2, –SA3, –SB, –SB2 and –SB3.

3 Either a RAM module or ROM module can be connected to each board of PMC–SC and PMC–SC3.

(4) Connecting the off–line programmer
Connect the off–line programmer to the reader/punch interface on the CNC. There are several connectors for the reader/punch interface on the CNC. The connector to be used is specified during I/O processing for the PMC. For details, see Section 7.

3 SELECTION OF PROGRAMMER MENUS BY SOFTKEYS

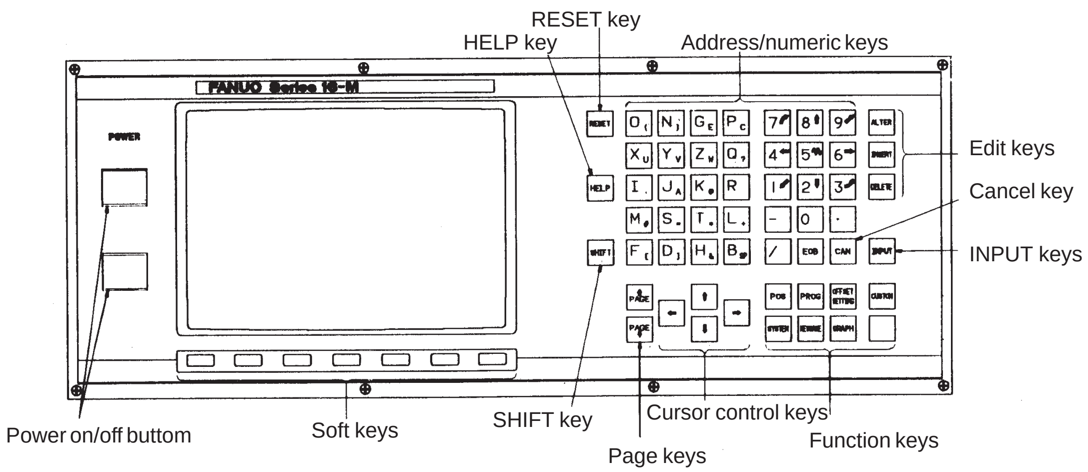
To operate the PMC programmer, set bit 1 in K17 of the keep relay area for PMC parameters to 1, enabling the programmer basic menu to be displayed. To display the programmer basic menu, press <SYSTEM> and [PMC] soft key on the MDI keyboard then, press the [NEXT] key.

The programmer basic menu is displayed at the lower part of the CRT screen to signify the keys as shown in the following figure.

(1) Programmer basic menu

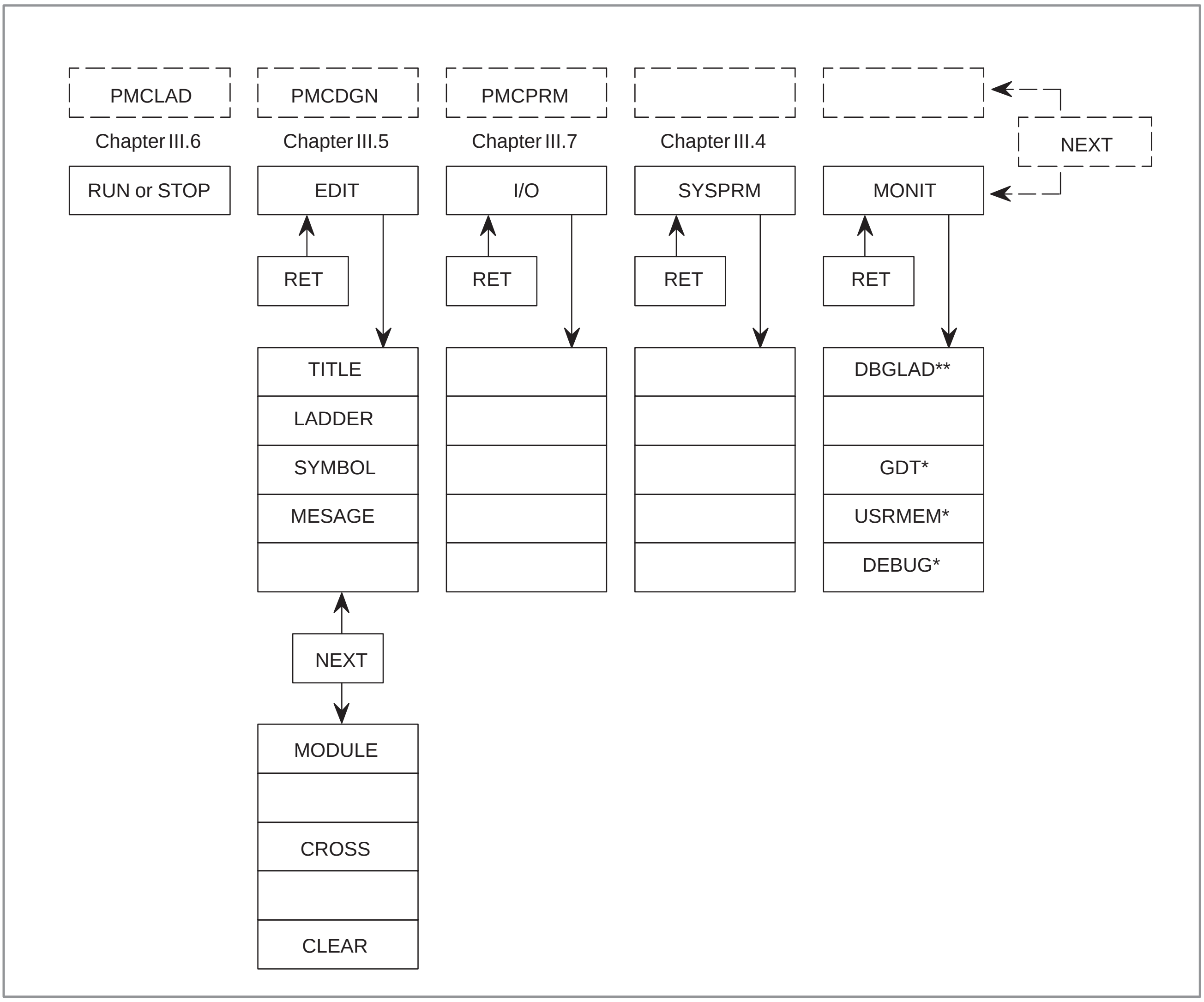
The programmer basic menu and PMC basic menu are selected to each other alternately by pressing the [NEXT] key. For the PMC basic menu and operation, see PMC operation in Chapter II.

NOTE
In the following description, the relation between soft keys and menus is described based on 9–inch CRT/MDI panel. The 14–inch CRT/MDI panel is different from the 9–inch CRT/MDI panel about the number of soft keys. Five soft keys are mounted on the 9–inch CRT/MDI panel, while ten soft keys are mounted on the 14–inch CRT/MDI panel.



(2) Relation between programmer menus and soft keys

The relation between programmer menus and soft keys are different according to each function as shown in the following figure. These menus are selected by pressing related keys. For the menu contents, see the description given later. Refer to this figure for operation.



NOTE
1 Mark “*” is valid for PMC–SC/SC3/SC4/NB function.
2 Mark “**” is valid for PMC–SA3/SB3 with Editing module or PMC–SC/SC3 function.

4 SPECIFYING AND DISPLAYING SYSTEM PARAMETERS (SYSPRM)

Display the system parameter screen by pressing soft key [SYSPRM] on the basic programmer menu. Move the cursor to necessary system parameters and specify them according to the menu displayed on the screen. When this function is selected, if the sequence program is in operation, the PMC management software automatically stops this function.

(1) COUNTER DATA TYPE

Specifies whether the counter value is used in binary or BCD by functional instruction CTR.

> **CAUTION**
> After changing a counter data type, set up the counter value again.

(2) LADDER EXEC (valid for PMC–SC/SC3/SC4/NB/NB2)

Specifies the increment or decrement of processing time of the 1st and 2nd level parts of the ladder program in the range of 1% to 150%. This increases or decreases the scanning time of the ladder program. This parameter influences the processing time of the 3rd level part of the ladder program and the language program.

If 100% is specified, the time of 5 ms for an 8 ms cycle is used to process the 1st and 2nd level parts of the ladder program. The remaining 3 ms is used to process the 3rd level part of the ladder program, language program, and PMC screen display.

If 120% is specified, the time of 6 ms is used to process the 1st and 2nd level parts of the ladder program. This reduces the scanning time of the ladder program, thus enabling the ladder program to be executed at high speed. Note that the processing time required for the 3rd level part of the ladder program, language program, and PMC screen display is substantially reduced. If the undivided system is specified too, this parameter is validated.

If a value less than 40% is specified, 40% is assumed. If a value greater than 120% is specified, 120% is assumed.

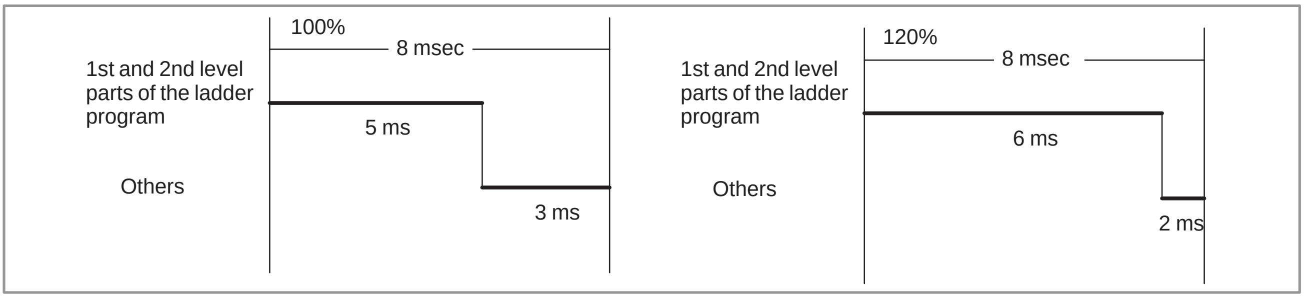
The processing time of the 1st and 2nd parts of the ladder program is obtained by the following formula:

Processing time of the 1st and

$$\text{2nd parts of the ladder program} = 5\ \text{msec} \times \frac{\text{(LADDER EXEC)}}{100}$$

The processing time of the 3rd level part of the ladder program, language program, and PMC screen display

= 8 ms – (processing time of the 1st and 2nd level parts of the ladder program)



(3) LANGUAGE EXEC RATIO (valid for PMC–SC/SC3/SC4/NB/NB2)

Specifies the division ratio of execution for PMC screen display and language program.

(0 to 99%)

Since the execution priority of PMC screen display is higher than language program tasks, it is usually hard for the tasks to execute processing while displaying PMC screen. Then this parameter can be used to set the division ratio for each. Cyclic processing of language program is therefore possible during PMC screen display. Only language program tasks are running if PMC screen is not displayed.

(4) IGNORE DIVID CODE (valid for PMC–SB and–RC)

Specifies whether the ladder program is executed in the divided system (IGNORE DIVID CODE = NO) or in the undivided system.

(5) LANGUAGE ORIGIN (valid for PMC–SC/SC3/SC4/NB/NB2)

Specifies the first address of the link control statement data in the language program.

Be sure to specify 0 when the language program is not stored.

LANGUAGE AREA and SIZE indicate the area where the language program is stored. Store the language program in the specified area.

When the language program is stored, the Language Origin is automatically set by moving the cursor to this item and pressing [ORIGIN] soft key.

(6) MAX LADDER AREA SIZE (valid for PMC–SC/SC3/NB)

Specify the maximum size of the ladder program. This parameter can be used to increase or decrease the size of the work area used by language programs. The setting of the parameter takes effect only after power is turned on. When the setting is to be changed, therefore, power must be turned off.

For details, see the FANUC PMC–MODEL SC/SC3/SC4/NB PROGRAMMING MANUAL C LANGUAGE (B–61863E–1). The default is the size in kilobytes resulting from conversion of the ladder step option.

(7) FS0 OPERATOR PANEL

Specifies whether the Series 0 operator's panel is connected. When YES is selected, specify the actual addresses of DI and DO connected to the operator's panel, the address of the key image transferred from the operator's panel, and the address of the LED image to be transferred to the operator's panel.

(a) KEY DI ADDRESS

Specify a PMC address representing the first address of the external DI actually connected (X0 to X127 or X1000 to X1019).

(b) LED DO ADDRESS

Specify a PMC address representing the first address of the external DO actually connected (Y0 to Y127 or Y1000 to Y1014).

(c) KEY BIT IMAGE ADDRESS

Specify a PMC address representing the first address of the key image to be referenced by the user program. Usually specify an arbitrary internal relay area.

(d) LED BIT IMAGE ADDRESS

Specify a PMC address representing the first address of the key image to be generated by the user program. Usually specify an arbitrary internal relay area.

(8) STEP SEQUENCE

When creating new programs with the built–in editing function, set this parameter first, then execute CLEAR ALL or perform clear operation (turn on power while holding down X and O) at power on.

When selecting the step sequence method: STEP SEQUENCE = YES

When selecting the ladder method: STEP SEQUENCE = NO

```
PMC SYSTEM PARAMETER
 COUNTER DATA TYPE     =  BINARY/BCD

 FS0 OPERATOR PANEL    =  YES/NO

 KEY DI ADDRESS        =  X100

 LED DO ADDRESS        =  Y100

 KEY BIT IMAGE ADDRESS =  R900

 LED BIT IMAGE ADDRESS = R910
>

[BINARY] [ BCD   ] [      ] [      ] [      ]
```

Fig. 4(a) PMC–SA series system parameter screen

```
PMC SYSTEM PARAMETER

 COUNTER DATA TYPE     =  BINARY/BCD

 IGNORE DIVIDE CODE    =  NO/YES






>

[BINARY] [ BCD   ] [      ] [      ] [      ]
```

Fig. 4(b) PMC–SB series system parameter screen (1st page)

```
PMC SYSTEM PARAMETER                 MONIT STOP

 COUNTER DATA TYPE       =  BINARY /BCD

 LADDER EXEC             =  100% (1–150)

 LANGUAGE EXEC RATIO     =  50% (0–99)

 IGNORE DIVIDE CODE      =  NO/YES

 LANGUAGE ORIGIN         =  841000H
 (LANGUAGE AREA = 840000H,SIZE = 768KB)
 MAX LADDER AREA SIZE =  90KB (1–96)

[BINARY] [ BCD   ] [      ] [      ] [      ]
```

Fig. 4(c) PMC– SC, SC3 or NB system parameter screen (1st page)

```
PMC SYSTEM PARAMETER (1/2)         MONIT STOP

 COUNTER DATA TYPE     =  BINARY/BCD

 IGNORE DIVIDE CODE    =  YES/NO

>

[BINARY] [ BCD   ] [      ] [      ] [      ]
```

Fig. 4(d) PMC–SB4/SB6/SC4 system parameter screen (1st page)

```
PMC SYSTEM PARAMETER (1/2)         MONIT STOP

 COUNTER DATA TYPE      =  BINARY/BCD

 LADDER EXEC            =  % (1–150)

 LANGUAGE EXEC RATIO    =  % (0–99)

 LANGUAGE ORIGIN        =  H
  (LANGUAGE AREA =       H, SIZE =    KB)

 STEP SEQUENCE          =  YES/NO
 >

[ BINARY ] [  BCD  ] [      ] [      ] [      ]
```

Fig. 4(e) PMC–SC4/NB2 system parameter screen (1 st page)

Press the [NEXT] key to select the following screen for PMC–SB series, PMC–SC series, and PMC–NB :

```
PMC SYSTEM PARAMETER (2/2)          MONIT STOP

 FS0 OPERATOR PANEL     =  YES/NO

 KEY DI ADDRESS         =  X100

 LED DO ADDRESS         =  Y100

 KEY BIT IMAGE ADDRESS  =  R900

 LED BIT IMAGE ADDRESS  =  R910

 >

[ YES  ] [  NO   ] [      ] [      ] [      ]
```

Fig. 4(f) PMC–SB series, PMC–SC series, or PMC–NB system parameter screen (2nd page)

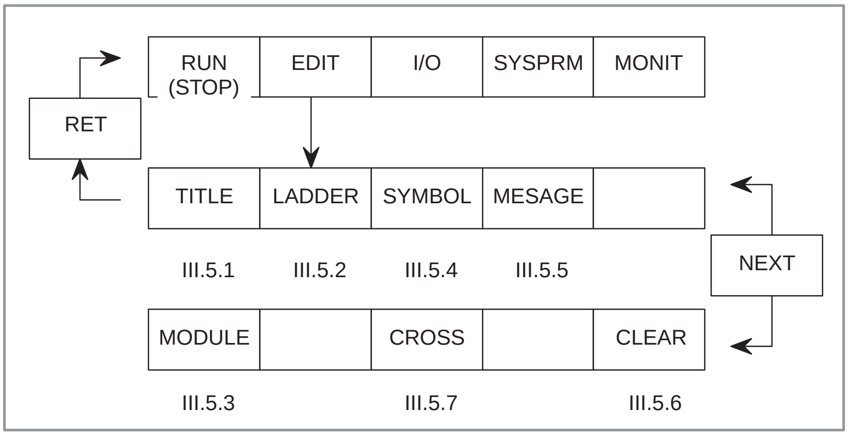
5 EDITING OF SEQUENCE PROGRAM (EDIT)

Press soft key [EDIT] of the programmer basic menu to bring the following menu. For setting the CLEAR or I/O unit address, press the [NEXT] key to bring another menu.

Each menu of [EDIT] can be selected by EDIT key, or menu of other EDIT can be selected by each EDIT menu. When this function is selected, if the sequence program is in operation, the PMC management software automatically stops this function.

(Operation)

Perform each operation by pressing necessary menu soft keys. Press [RETURN] key for resetting to the programmer basic menu.




```
PMC EDITION MENU                              MONIT STOP

 SELECT ONE OF FOLLOWING SOFT KEYS

  TITLE      :  TITLE DATA
  LADDER     :  LADDER DIAGRAM
  SYMBOL     :  SYMBOL & COMMENT DATA
  MESAGE     :  MESSAGE DATA
  MODULE     :  I/O MODULE DATA
  CLEAR      :  CLEAR DATA
  CROSS      :  CROSS REFERENCE



 [TITLE ] [LADDER] [MESAGE] [      ] [      ]

 [MODULE] [      ] [CROSS ] [      ] [CLEAR ]
```

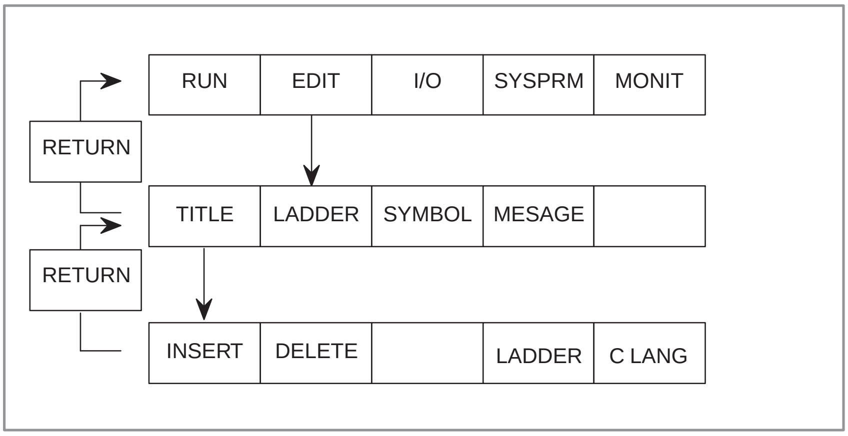
Fig. 5 Editing basic menu

5.1 SPECIFYING AND DISPLAYING TITLE DATA (TITLE)

The title data refers to the title of the sequence program created by the machine tool builder. The data consists of the following ten items:

- Machine tool builder name (32 characters)
- Machine tool name (32 characters)
- NC and PMC types (32 characters)
- Sequence program number (16 characters)
- Version (4 characters)
- Sequence program drawing number (32 characters)
- Date of sequence program creation (16 characters)
- Sequence program programmer (32 characters)
- ROM programmer (32 characters)
- Comment (32 characters)

The title for the 9″ CRT consists of three screens. The screens are changed by pressing <PAGE↑> or <PAGE↓> .



NOTE
When a C language board is installed in the Series 16*i*/18*i*, the title data for C can be edited. With the soft key [LADDER], the display can be switched to the ladder title data. With the soft key [C LANG], the display can be switched to the C title data.

5.1.1 Entering Title Data

(1) Move the cursor to the desired title data item. Use the cursor keys [↑], [↓], [→], [←] to move the cursor.

(2) Press the address key and numeric keys to enter the title data, and press the <INPUT> key.

5.1.2 Deleting Title Data

(1) Move the cursor to the desired title data item. Use the cursor keys [↑], [↓], [→], [←] to move the cursor.

(2) After keying in the title data by pressing the desired address keys and numeric keys, press the <INPUT> key.

5.1.3 Editing Character Strings of Title Data

When the length of the cursor is the same as the maximum number of characters, pressing the [INSERT] key enables the operator to edit character strings. Then, the length of the cursor is changed to that of one character.

(1) Move the cursor to the desired insertion position with the cursor keys and enter a character string. Then, the character string is inserted.

(2) Pressing the [DELETE] key deletes the character at the cursor.

```
PMC TITLE DATA #1                      MONIT  RUN

        PMC PROGRAM NO. :     1234
        EDITION NO.     :     12

        PMC CONTROL PROGRAM
        SERIES : 4061 EDITION : 01

           MEMORY USED : 44.0KB
           LADDER      : 32.0KB
           SYMBOL      : 10.2KB
           MESSAGE     : 01.8KB
           SCAN TIME   : 048 MSEC

 [INSERT] [DELETE] [      ] [      ] [      ]
```

Fig. 5.1.3 (a) Title edit screen 1

```
PMC TITLE DATA #2                      MONIT  RUN

 MACHINE TOOL BUILDER NAME :
 ○ · · · · · · · · · · · · · ○

 MACHINE TOOL NAME :
 ○ · · · · · · · · · · · · · ○

 CNC & PMC TYPE NAME :
 ○ · · · · · · · · · · · · · ○

 PROGRAM DRAWING NO. :
 ○ · · · · · · · · · · · · · ○

 [INSERT] [DELETE] [      ] [      ] [      ]
```

Fig. 5.1.3 (b) Title edit screen 2

```
PMC TITLE DATA #3                               MONIT  RUN

 DATE OF PROGRAMING :
 ○ . . . . . . . . . . . . . . ○

 PROGRAM DESIGNED BY :
 ○ . . . . . . . . . . . . . . ○

 ROM WRITTEN BY :
 ○ . . . . . . . . . . . . . . ○

 REMARKS :
 ○ . . . . . . . . . . . . . . ○


 [INSERT] [DELETE] [       ] [       ] [       ]
```

Fig. 5.1.3 (c) Title edit screen 3

5.2 SEQUENCE PROGRAM GENERATION (LADDER)

Input, insert, delete, and search a sequence program as described below. The relation between these functions and soft keys is as shown below.

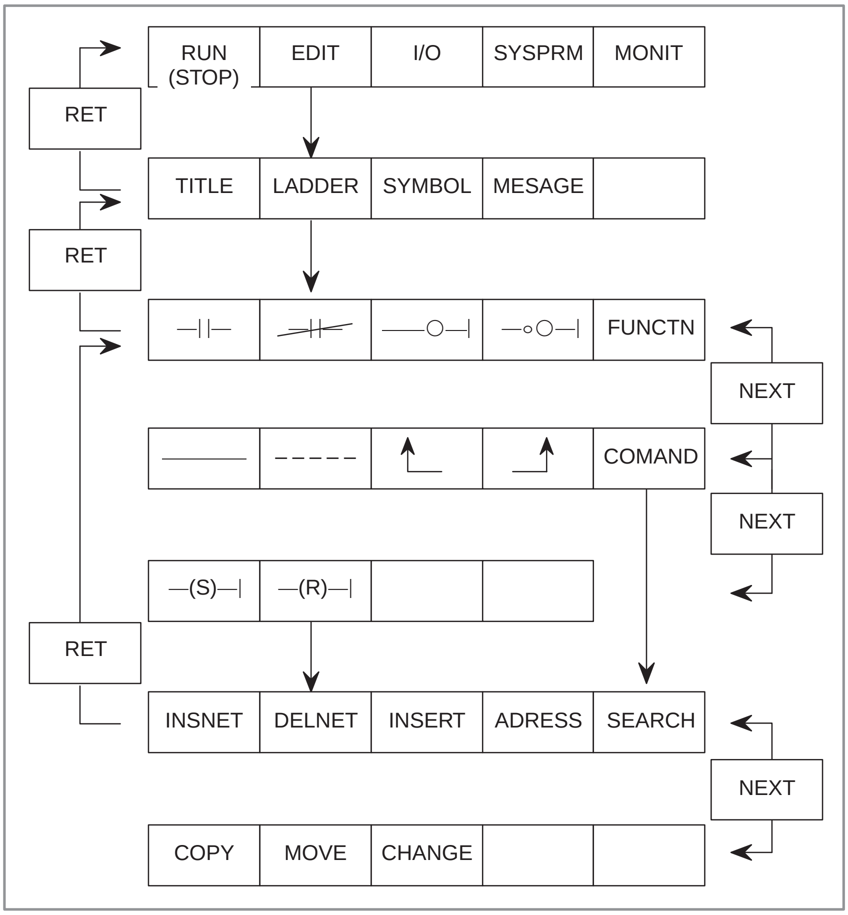


Fig. 5.2 Sequence program generation softkeys

> **NOTE**
> "—(S)—|"and "—(R)—|" are valid for PMC–PA3, –SA3, –SB3, –SB4, –SC3, –SC4, and –NB.

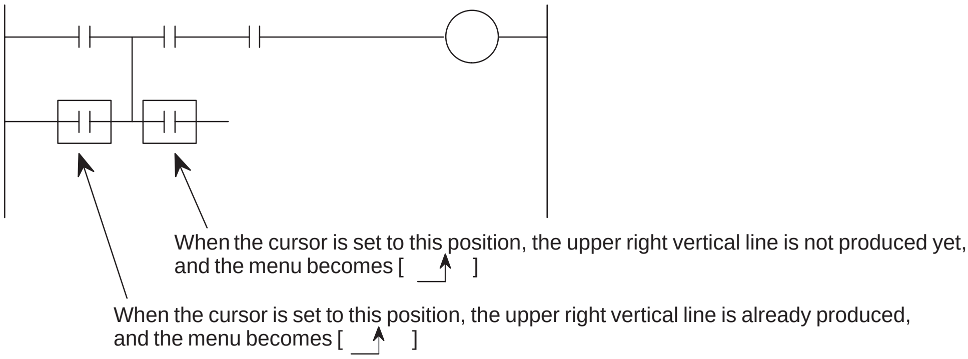
Each of EDIT · LADDER software functional instruction keys can be selected by the [COMAND] key. Type in one of the following character strings and press software key [COMAND]. The character string within parentheses "[ ]" can be omitted. "n" after the character string indicates that a value can be input. For example, if the [COMMAND] key is pressed after "D2" is typed in, the operation can be performed in the same manner when the <DELNET> key is pressed after "2" is typed in.

I[NSERT]	D[ELNET][n]	n:value
A[DRESS]	SY[MBOL]	
S[EARCH]	C[OPY][n]	
M[OVE][n]		

Generate and search a program by pressing soft keys of the above menu.

NOTE

Soft keys ([ ] or [ ]) ([ ] or [ ]) are used for producing or deleting an upper left vertical line or upper right vertical line on the ladder diagram. The solid line display vertical line indicates the production, while the dotted line display vertical line shows the deletion. Which one is available is determined by the ladder diagrams and cursor positions.



5.2.1 Sequence Program Input

Press soft key [LADDER] for inputting a sequence program. The soft key menu changes as shown in Fig. 5.2.

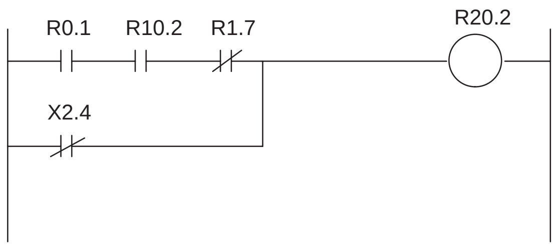
If a sequence program is not input yet, the right and left vertical lines only of the ladder diagram are displayed on CRT/MDI.

Start inputting a program with this screen condition. If a previous program remains unerased from RAM module for debug, clear it according to the instruction in 5.6 before starting the program input.

Input a ladder diagram by moving the cursor to the desired input position by using the cursor key.

The following description shows an example of the input of a program of the basic instruction and a program of the functional instruction.

(1) In case of basic instruction program input;



The contacts and coils inputtable in one line are as specified below.

9–inch CRT/MDI
10–inch CRT/MDI } 7 contact + relay coil
14–inch CRT/MDI

They cannot be input into one line more than specified. If they exceed the specified range, provide a dummy relay coil halfway.

1 Press soft key [—| |—] after moving the cursor to the start position.
Symbol [—| |—] is input to the cursor position and HORIZONTAL LINE ILLEGAL is displayed at the lower right part of the CRT screen. This is a caution message to show that the ladder diagram horizontal line is not entered yet. Input address and bit data next.

2 Press <INPUT> key after inputting R0.1 by using address key and numeric keys. The address is set on the contact, and cursor shifts rightward.

3 Input A contact with address R10.2 by the above method **1**, **2**.
(Note) The order of processes **1** and **2** are interchangeable.

4 Input B contact R1.7

Press soft key [—|/|—], input address R1.7, and then, press <INPUT> key.
The address is set on the B contact and the cursor shifts rightward.

5 Press soft key [—○—|]with the cursor kept as it is.

A right horizontal line is automatically drawn, and a relay coil symbol is entered near the right vertical line.

6 Press <INPUT> key after inputting.

The cursor automatically shifts to the input start position of the next line.

7 Input the OR condition next,

Press soft key [——], input address X2.4 and then, press INPUT key. The address is set on the B contact and the cursor shifts rightward.

8 Press NEXT key, since the soft key of the right horizontal line of OR is necessary.

9 Press soft key [—|/|—] to input a horizontal line.

When inputting the horizontal bar key ([——], [- - - -]), key in a numerical value and press this bar key, and then the horizontal line for the frequency will be drawn. However, this horizontal line will not be drawn over the LINE.

10 Press soft key [—↑], and input necessary upper right vertical line or OR.

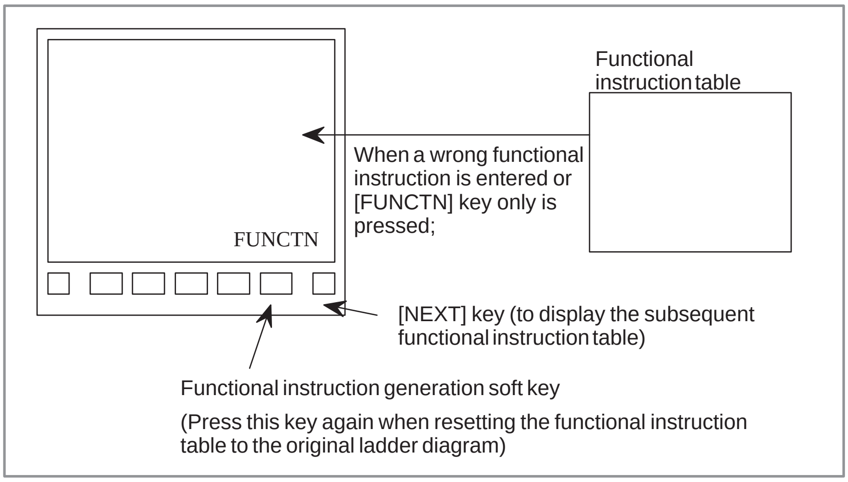
CAUTION

1. When the ladder program displayed on the screen is incomplete (when, for example, addresses have not been entered) or erroneous, the screen cannot be scrolled even when a page key is pressed. Before attempting to scroll the screen, therefore, ensure that the ladder program is complete and error–free.
   However, be careful since the program net (a block corresponding to a range from RD to WRT Instruction) containing an error is deleted when the screen is switched to an CNC screen.
2. 7 contacts + a coil are specified to be inputtable per line from CRT/MDI, any more contacts exceeding the specified value are not inputtable.
   However, this limitation is not applicable to mnemonic sequence programs generated by Offline programmer. When a sequence program, transferred from the offline programmer to the PMC, exceeds the length which can be displayed on a single line, the program is displayed using two or more lines, linked with a continuation symbol. This continuation symbol is not erasable usually, except when all programs from RD instruction to WRT instruction are erased.
3. If the power is turned off while a ladder program is being displayed in edit mode, that ladder program will be lost. Always save the program and exit the editing screen before turning off the power.
4. The termination processing of the ladder (JMP, COM, and other processing) is done when the EDIT screen is switched to another screen by pressing RET key, it takes several tens second until the screen is switched completely, if the ladder is large.
5. In the Series 15–MODEL B, Series 16/18–MODEL B/C, and Series 16*i*/18*i*/21*i*–MODEL A CNC that use Flash Memory, the program is not automatically written into Flash Memory once editing ends. After editing, perform the processing for writing to Flash Memory (see 7.3.3, “FROM” in Chapter 7 of Part III). Otherwise, the editing results will be lost when the power is turned off.
6. When the user presses the RET key to switch from the edit screen to another screen, the parameters of functional instructions TMR, TMRB, CRT, DIFU, and DIFD are checked for a parameter number range error and duplicate parameter number in the ladder termination processing. If a range error is found, the editing cannot be terminated. If the use of a duplicate parameter number is found, the guidance message is displayed.

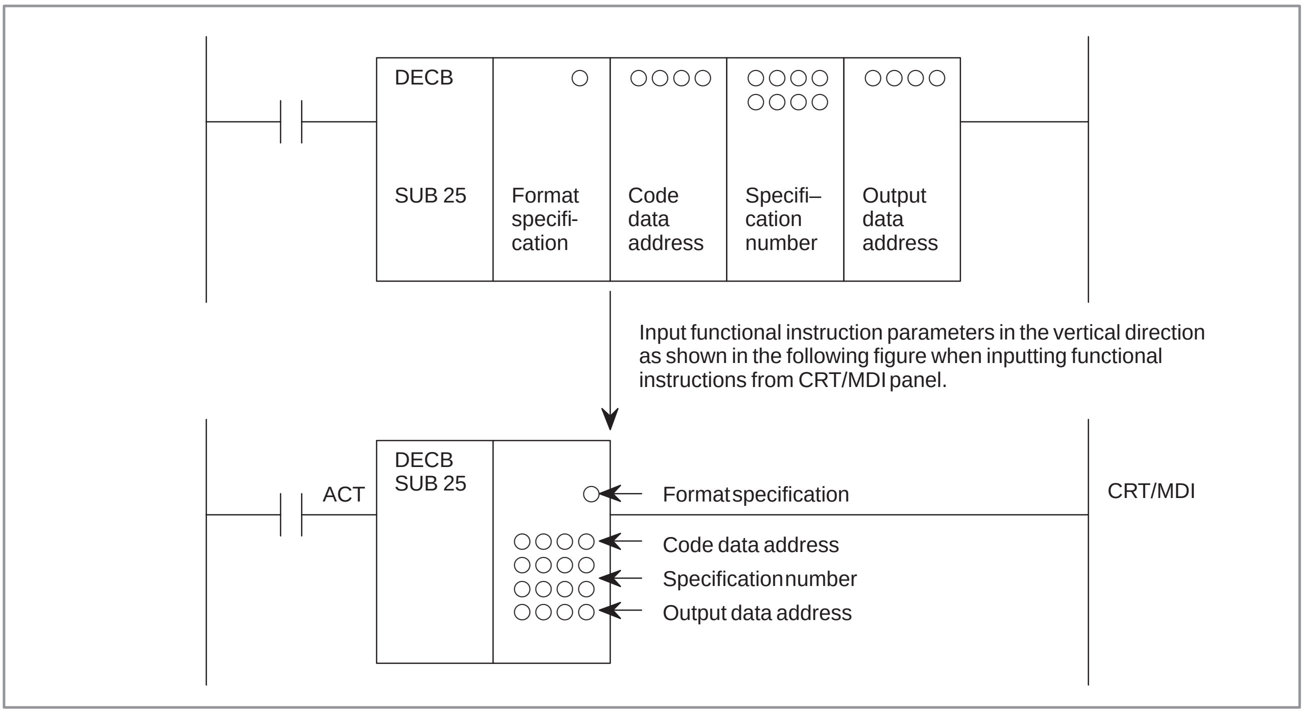
(2) In case of functional instruction program input;

For inputting a functional instruction, press [FUNCTN] soft key, and then, input instruction symbol of the functional Instruction and SUB number.
A function command can be input by pressing the [FUNCTN] key after keying the Function Command No. When pressing the [FUNCTN] key without keying in the Function Command No., the function command table is displayed. Key in Function Command No. and press [INPUT] key.
If you don’t keep the instruction symbol and SUB number into mind, you can display a functional instruction table covering the correspondence between instruction on symbols and SUB numbers automatically by inputting a wrong instruction symbol or a wrong SUB number and then pressing the soft key [FUNCTN] key or by pressing soft key [FUNCTN] key only without inputting any other key.
If an aimed functional instruction is not found in the displayed functional instruction table, press [NEXT] key or [PAGE] key to brings its subsequent table.

Press [FUNCTN] key when resetting the functional instruction table to the original ladder diagram.



> **NOTE**
> If the system is left undone without inputting any data after pressing soft key [FUNCTN], the other soft keys are not employ–able. In such a case, press [FUNCTN] key again.



1 Input a control condition.

Press soft key [–| |–], input the address and bit data, and then, press <INPUT> key. The cursor shifts rightward.

2 Input an instruction.

Press soft key [FUNCTN], input SUB number 25, and then, press <INPUT> key. A functional instruction diagram appears as shown in the above figure.

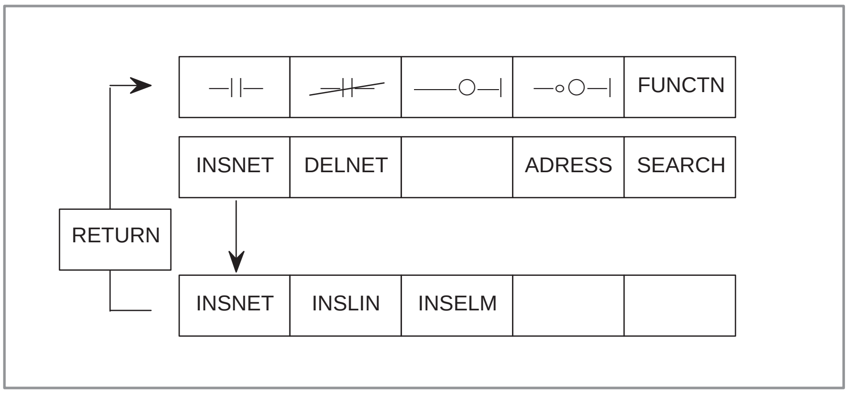
3 Input an instruction parameter.

Input the first parameter, format specification, and then, press <INPUT> key. The cursor automatically lowers downward. Input three residual parameters in order.

5.2.2 Alteration of Sequence Programs

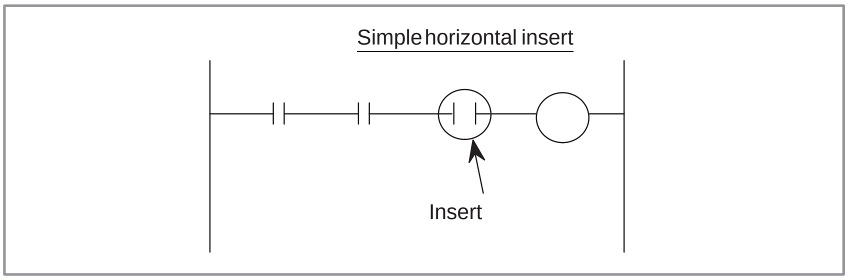
The method of altering a generated sequence program is the same as described in 5.2.1. Move the cursor to the program part to be altered and input change data.

5.2.3 Insert of Sequence Program

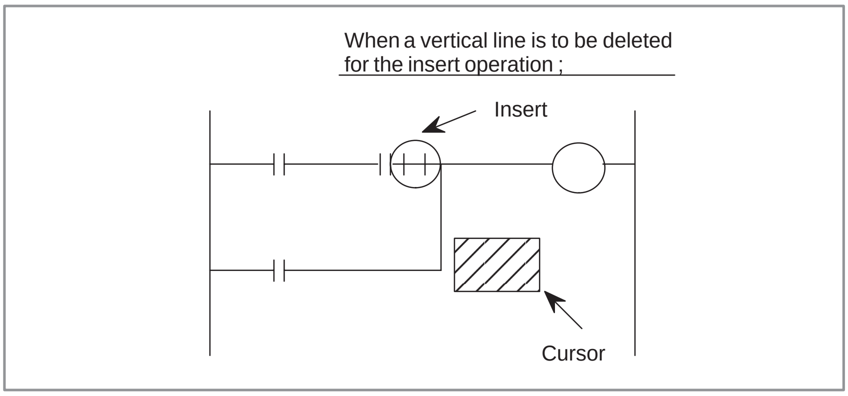


A sequence program is inserted in four ways on the ladder diagram as described below.

(1) To insert a relay contacts in the horizontal direction.



Move the cursor to the position where a sequence program is to be inserted, and input the program by the method specified in 5.2.1.



1 Set the cursor to the above position.

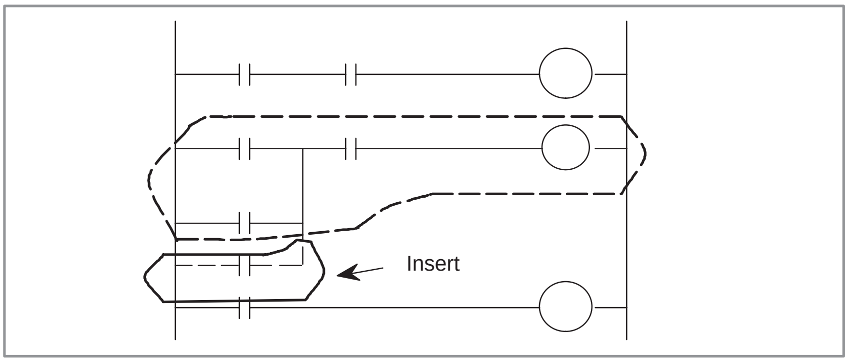
2 Press soft key [↑—] for erasing the upper left vertical line. The upper left vertical line to the cursor disappears.

3 Press soft key [—↑] to produce a upper right vertical line to the cursor, then, press soft key [——]. Both verti–cal line and horizontal line are pro–duced.

4 Shift the cursor to a line of contact insert position.

5 Press soft key [—| |—] to add contacts.

(2) For inserting vertical line;



For inserting a vertical line as shown in the above figure, the inserting area is required, correspondingly. In order to produce the area, shift the entire part after the part to be inserted by one line by moving the cursor to the ladder diagram within the dotted line range (an optional part is allowable) and then pressing soft key [INSNET] (see Fig. 5.2).

The lower ladder diagram shifts downward by one line, each time the [INSNET] key is pressed to produce the area to which a line is to be inserted.

If a surplus insert area remains unused after the insert processing ends (if an area corresponding to 3 lines has been reserved when two lines have been inserted, for example), the area may be left as it is. No problem arises.

1 Move the cursor to the ladder diagram bounded by a dotted line.

2 Press soft key [INSNET].

The lower ladder diagram shifts downward by one line.

3 Pressing [INSNET] key without keying in numeric values will cause one line to be inserted.

4 Pressing [INSNET] key with keying in numeric values will cause the line to be inserted the number of numeric values input.

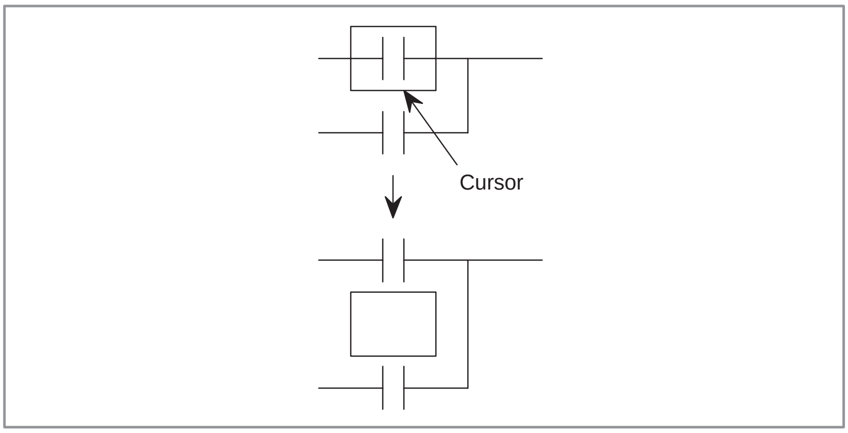
5 After setting the cursor to a position where the contacts is to be inserted, press soft key [—| |—]. After setting address data, press [INPUT] key. The cursor shifts rightward.

6 Press soft key [↑—] to produce an OR circuit.

(3) Inserting the 1 NET sequence program lines

Space lines are inserted one by one.

1 Type in the number of lines to be inserted and press the [INSLIN] key. The lines corresponding to the input number are inserted. (If the number of lines to be inserted is not typed in but the [INSLIN] key is pressed, one line is inserted.)

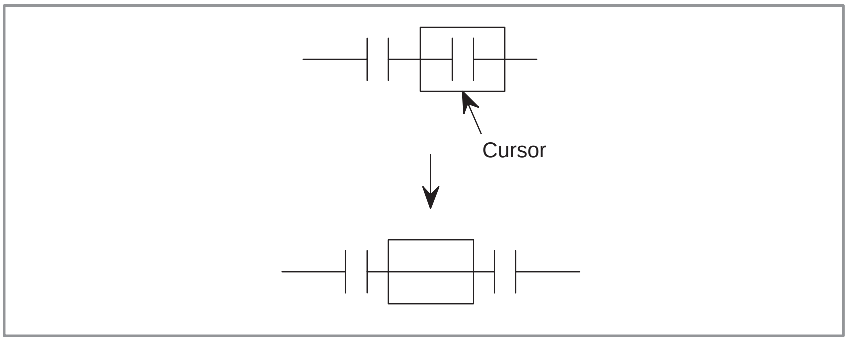


If the [INSLIN] key is pressed when the cursor is in the position specified as shown in the above figure on the left, the line is inserted as shown in the above figure on the right.

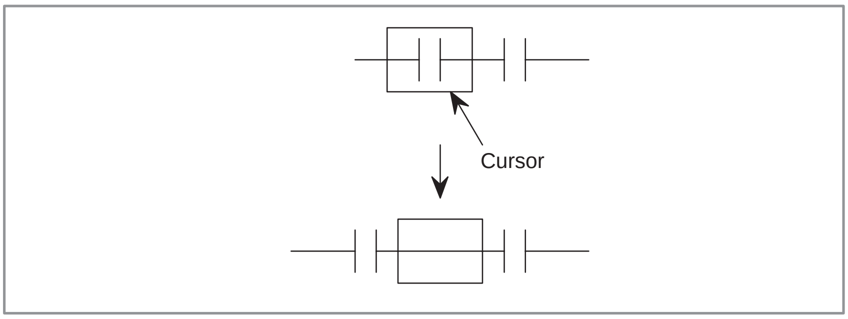
(4) Inserting the 1 NET sequence program elements

Elements can be inserted one by one.

1 Type in the number of elements to be inserted and press the [INSELM] key. The elements corresponding to the input number are inserted. If the number of elements prefixed by character “A” is typed in and the [INSELM] key is pressed, the elements are inserted after the cursor.
(If the number of elements to be inserted is not typed in but the [INSELM] key is pressed, one element is inserted.)



When the [INSELM] key is pressed when the cursor is positioned as shown in the above figure on the left, the element is inserted as shown in the above figure on the right.



If “A” is typed in when the cursor is positioned as shown in the above figure on the left and the [INSELEM] key is pressed, the element is inserted as shown in the above figure on the right.

5.2.4 Delete of Sequence Program

(1) Delete a part of sequence program by using three kinds of soft keys after setting the cursor to the portion from which the sequence program is to be deleted.

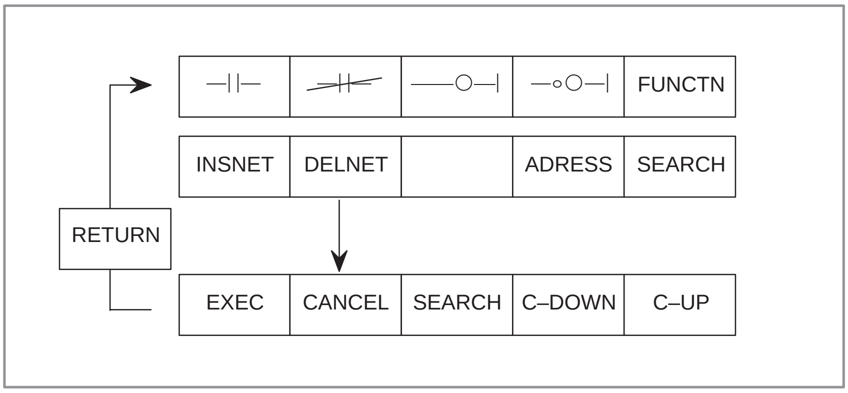
[—] : Delete of horizontal lines, relay contacts, relay coils, etc.

[↑—] : Delete of upper left vertical line to the cursor

[—↑] : Delete of upper right vertical line to the cursor

(2) Delete a net of the sequence program (the part from the RD instruction to the WRT instruction) with the [DELNET] key.

(3) Deleting NETs one by one



1 Deletion

Move the cursor to the NET to be deleted and press the [DELETE] key. The NET to be deleted brightly displayed on the screen.

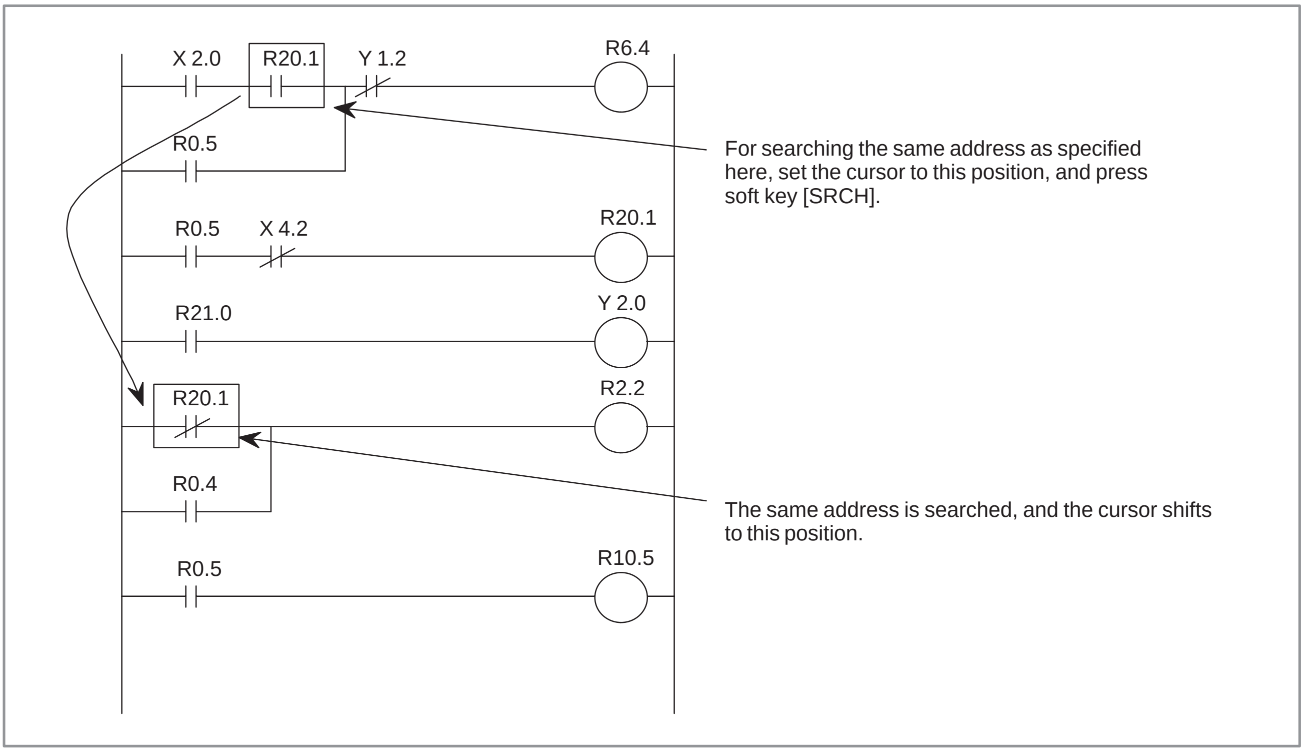
2 Deleting multiple NETs

Move the cursor with the cursor DOWN key, [C–DOWN] key, or [SEACH] key to blink the NETs to be deleted. Type in a value and press the [C–DOWN] key to move the cursor the number of times specified by this value.

3 Execution Press the [EXEC] key.

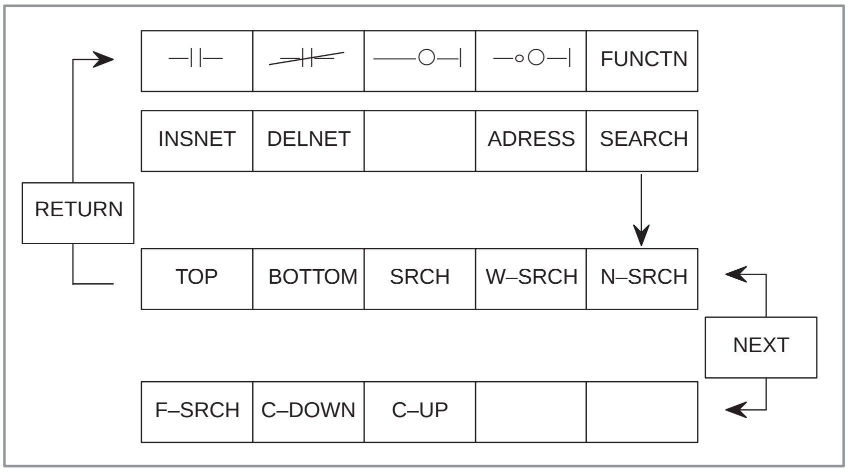
Cancel Press the [CANCEL] key.

4 If the NET to be deleted is already known, move the cursor to the first NET, type in the number of NETs, and press the [DELNET] key to omit steps **1** and **2**.



5.2.5 Search of Sequence Program

Search a sequence program by using the following soft keys. For the following soft keys, see Fig. 5.2.



(1) Soft key [TOP]

When this key is pressed, the start of the sequence program is displayed on the screen and the cursor shifts to this start position.

(2) Soft key [BOTTOM]

When this key is pressed, the last of the sequence program is displayed on the screen, and the cursor shifts to this position.

(3) Soft key [SRCH]

This key is used to search a specified address. It searches the specified address from the program of the cursor part to the last program of this screen, and displays the address on the screen.

a) Method of specifying the address by the cursor

Set the cursor to the relay contact part of the address to be searched and press soft key [SRCH].

The system searches the same address as the address specified by the cursor from the cursor program on the presently displayed screen to the end of the program (SUB 48).

When the same address is found, the program part is displayed on the screen, and the cursor shifts to the address part. If the same address is not found as a result of this search, an error is displayed.

b) Method of specifying the address by inputting it

Input an address to be searched by using address and numeric keys and press soft key [SRCH]. The same address as specified is searched from the program of the cursor part on the presently displayed screen to the last of the program (SUB 48).

When the same address is found, the program part is displayed on the screen, and the cursor shifts to the address part.

If the same address is not found as a result of this search, an error is displayed.

(4) Soft key [W–SRCH]

This key specified an address of the relay coil to be searched, and searches the relay coil of the specified address from the program at the cursor part to the end of the program (SUB 48) on this screen. Then, it displays the relay coil on the screen.

Two methods are available to specify the address of the relay coil to be searched.

a) Method of specifying the address by cursor

Set the cursor to the relay contact of the relay coil to be searched, and press soft key [W–SRCH].

The corresponding relay coil is searched from the program of the cursor part to the end of the program (SUB 48).

When the relay coil is found, the program part is displayed on the screen, and the cursor shifts to the relay coil.

If no corresponding relay coil is found as a result of search, an error occurs.

b) Method of specifying the address by inputting it

Input the address of the relay coil to be searched by both address and numeric keys, and then, press soft key [W–SRCH].

The specified address relay coil is searched from the program of the cursor part on the presently displayed screen to the end of program (SUB 48).

When the specified address relay coil is found, the program part is displayed on the screen, and the cursor shifts to the relay coil.

If no relay coil is found as a result of search, an error is displayed.

(5) Soft key [N–SRCH]

Display the ladder with the specified NET number from the top of the screen.

If the number is not typed in but the [N–SRCH] key is pressed, the display is scrolled down by one NET.

(6) Soft key [F–SRCH]

Type in the functional instruction number and press the [F–SRCH] key to start searching the functional instruction. When the [F–SRCH] key is pressed during execution of a functional instruction, the functional instruction with the same number as this instruction is searched.

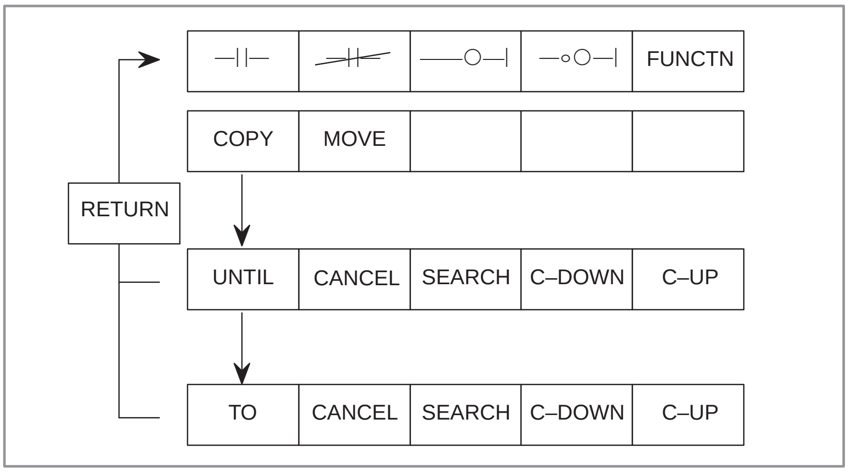
(7) Searching with cursor keys (<←>, <→>, <↑>, <↓>)

- Type in the address or symbol and press the cursor key to start searching the address. When the ”→” key is pressed, the operation is performed in the same manner when the [SRCH] key is pressed.
- Type in NET NO. and press the cursor key to start searching the NET NO.
- Type in the functional instruction name or functional instruction number with “S” and press the cursor key to start searching the functional instruction.

Example) Type in “END1” or “S1” and press the cursor key to search functional instruction END1.

5.2.6 Copying the Sequence Program

The sequence program with multiple NETs can be copied in NETs. Specify the NET to be copied and the copy position with the cursor. The number of copies can be also specified.



1 Copying

Move the cursor to the NET to be copied and press the [COPY] key. The NET to be copied blinks on the screen.

2 Copying multiple NETs

Move the cursor with the cursor UP/DOWN key, [C–UP] key, [C–DOWN] key, or [SEARCH] key to blink the NETs to be copied. Type in a value and press [C–UP] or [C–DOWN] key to scroll up or down the screen by the number of times specified by this value.

3 Setting the NET to be copied

Press the [UNTIL] key.

4 Specifying the copying address

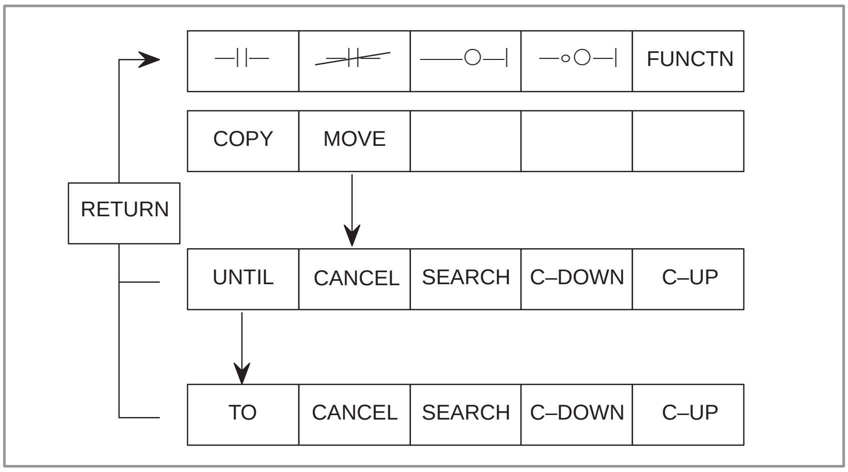
Press the [TO] key to start copying a NET. The NET is copied into the position above the cursor. If the number of copies is typed in before the [TO] key is pressed, the NET is copied the specified number of times.

5 If the cursor is moved to the first NET and the number of NETs is typed in when the NETs to be copied are known, steps **1** through **3** can be omitted by pressing the [COPY] key.

> **NOTE**
> An error NET cannot be copied.

5.2.7 Moving the Sequence Program

The sequence program with multiple NETs can be moved in NETs. Specify the NET to be moved and the move position with the cursor. The number of times of moving NETs can be also specified.



1 Moving

Move the cursor to the NET to be copied and press the [MOVE] key. The NET to be moved blinks on the screen.

2 Moving multiple NETs

Moving the cursor with the cursor UP/DOWN key, [C–UP] key, [C–DOWN] key, or [SEARCH] key to blink the NETs to be moved. Type in a value and press [C–UP] or [C–DOWN] key to scroll up or down the screen by the number of times specified by this value.

3 Setting the NET to be moved

Press the [UNTIL] key.

4 Specifying the copying address

Press the [TO] key to start copying a NET.

The NET is moved to the position above the cursor.

5 If the cursor is moved to the first NET and the number of NETs is typed in when the NETs to be moved are known, steps **1** through **3** can be omitted by pressing the [MOVE] key.

> **NOTE**
> An error NET cannot be copied.

5.2.8 Editing Symbol Data and Comment at Once

While editing a sequence program, symbol data and comment can be edited.

(1) The symbol data and comment assigned to undefined address can be edited.

a) Move the cursor to the position where a contact or coil is to be inputted.

b) Enter an address, enter the symbol and comment enclosed in characters other than alphanumeric characters, then press the soft key of [contact or coil].

(Example) When the contact X8.4 is assigned the symbol "*ESP" and the comment "EMERGENCY STOP".
Operation : Depress [contact] soft key after entering "X8.4/ *ESP/EMERGENCY STOP/".

(2) The symbol data and comment assigned to the address already defined can be edited.

a) Move the cursor on the address part where symbol data or comment will be edited.

b) Enter the symbol and comment enclosed in characters other than alphanumeric characters, then press the <INPUT> key.

(Example) When the contact X8.4 is assigned the symbol "*ESP" and the comment "EMERGENCY STOP".
Operation : Depress the <INPUT> key after entering "/*ESP/ EMERGENCY STOP/".

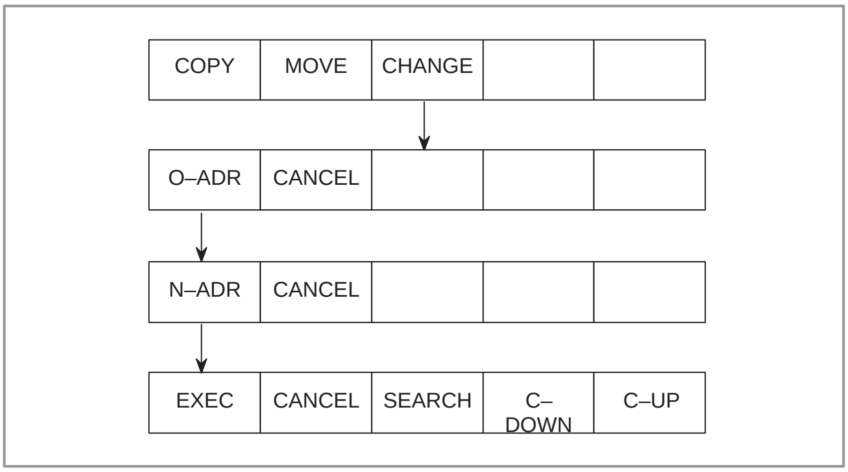
(3) The symbol data and comment only can be edited by the similar operation to the above (1) and (2).

a) For entering "X8.4/*ESP/" or "/*ESP/" with the "INPUT" key, the symbol data only can be edited.

b) For entering "X8.4//EMERGENCY STOP/" or "//EMERGENCY STOP/" with the "INPUT" key, the comment only can be edited.

5.2.9 Address Change of Sequence Program

The address in a sequence program can be replaced with another address by the procedure below.



(1) Changing the address while checking it one by one

a) Press the [CHANGE] key.

b) Input the original address and press the [O–ADR] key.

c) Input the new address and press the [N–ADR] key.

d) Press the [EXEC] key for executing the change. After completion, the cursor will shift downward to the nearest address to be changed.

If the [EXEC] key is pressed again at the point, the address change can be continued.

(2) Changing the address within the specified range

a) Press the [CHANGE] key, and move the cursor to the address to be changed.

b) Input the original address and press the [O–ADR] key.

c) Input the new address and press the [N–ADR] key.

d) The specified range will be brightened by using the [C–DOWN] or [C–UP] key.
All the addresses within the specified range can be changed.

e) Press the [EXEC] key for executing the change.

(3) Address designation by a wild card

The address to be changed can be designated by using the "*" code as a wild card.

(Example) "X*.0" means X0000.0, X0001.0, ... , X9999.0.
"X0000.*" means X0000.0, ... , X0000.7.
"X*" means X0000, X0001, ... , X9999.

The wild card can be used for both of the original address (O–ADR) and new adress (N–ADR).

The following are examples by wild card.

a) "X0.*" to "D100.*"
X0000.0 → D0100.0
X0000.1 → D0100.1
:
X0000.7 → D0100.7

b) "X*.0" to "X*.7"
X0000.0 → X0000.7
X0001.0 → X0001.7
:
X9999.0 → X9999.7

[Limit items]

- The address of data part in Functional instruction "DISP" cannot be changed.
- If the original address (O–ADR) and new address (N–ADR) are different in address name and the byte part of new address (N–ADR) is specified by a wild card, the change can not be done.

  Example) D1234.0 → X*.7, D* → X*

5.3 I/O UNIT ADDRESS SETTING (MODULE)

Set and delet the address of each module in I/O unit as follows. The relation between these functions and soft keys is as shown in the following figure.

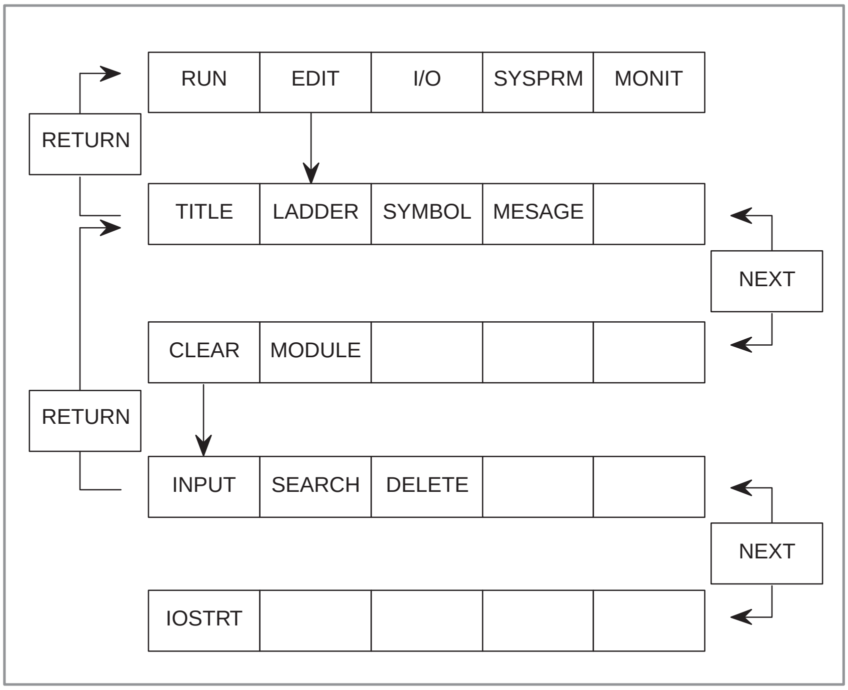


Fig. 5.3 Address setting for I/O unit

(1) Address setting for each module

1 Press the soft key [MODULE].

The following module address setting screen is displayed.

2 Move the cursor to the address to be set, and input data in the order of GROUP, BASE, SLOT, and NAME and press INPUT key. Input a dot (.) as a delimiter of each data.

Example) When setting the AID16A module with group = 0, base = 0, and slot = 5
0. 0. 5. ID16A

Table 3.2.2 in Section I–3.2 lists the necessary names for the NAME column.

3 Set all data of the module employed to aimed addresses by using the cursor key and page key.

```
PMC I/O MODULE

 ADDRESS GROUP BASE SLOT  NAME  ADDRESS GROUP BASE SLOT  NAME
   X000    2    0     2   ID16C   Y000    3    0     1   #1      <-- I/O Unit MODEL B
   X001    2    0     2   ID16C   Y001    3    0     5   #1
   X002    3    0     9   #2      Y002
   X003    3    0     9   #2      Y003
   X004    3    0    30   #2      Y004    2    0     1   OD16C   <-- I/O Unit MODEL A
   X005    3    0    30   #2      Y005    2    0     1   OD16C
   X006    3    0    10   #2      Y006    2    1     1   OD32D
   X007    3    0    10   #2      Y007    2    1     1   OD32D
   X008    3    0    20   #2      Y008    2    1     1   OD32D
   X009    3    0    20   #2      Y009    2    1     1   OD32D
   X010    3    0     0   ##      Y010    2    0     3   OD16C
   X011    3    0     0   ##      Y011    2    0     3   OD16C
   X012    3    0     0   ##      Y012
   X013    3    0     0   ##      Y013
   X014    0    0     1   FS04A   Y014    0    0     1   FS04A
                                                                 <-- Power Mate
GROUP.BASE.SLOT.NAME =
>2.0.4.OD08C
```

CAUTION

I/O module data items are made valid in the power–on sequence. When changing settings, be sure to turn off the power and turn it on to validate the settings.
However, the power need not be turned off and on again to validate settings when the programmer function version displays the soft key IOSTRT, (described later (item 4). Press the IOSTRT key after changing data.

(2) Delete of address

A preset address of each module can be deleted as follows:

1 Move the cursor to the address to be deleted, and press soft key [DELET] (see Fig. 5.3).

2 The preset address data are deleted.

(3) Soft key [SEARCH]

Searches the type–in address.

1 Type in the address to be searched and press the [SEARCH] key.

2 The typed–in address starts being displayed from the top of the screen.

(4) Validate the assignment data.

I/O module data is validated when the power is turned on. If I/O module data is changed without changing the configuration of the I/O devices, the new I/O module data is validated when the IOSTRT key is pressed.

The conditions where the IOSTRT key is displayed depend on the version of the programmer function.

(5) Error and warning messages issued during the editing of assignment data

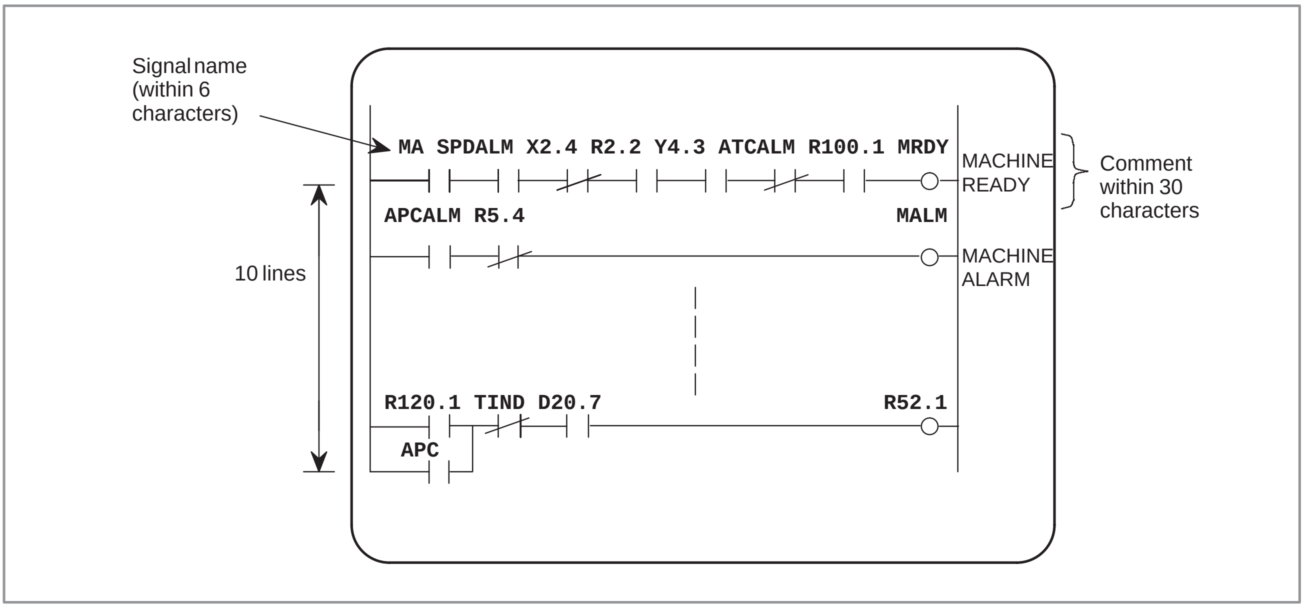
No.	Error or warning message	Description
1	ERR : GROUP NO. (0–15)	The group number must be from 0 to 15.
2	ERR : BASE NO. (0–3)	The base number must be from 0 to 3.
3	WARN : BASE NO. MUST BE 0	The base number must be 0 for the I/O Unit–B. It is forcibly set to 0.
4	ERR : SLOT NO. (1–10)	The slot number must be from 1 to 10 for the I/O Unit–A.
5	ERR : SLOT NO. (0, 1–30)	The slot number must be from 0 to 30 for the I/O Unit–B.
6	ERR : SLOT NO. MUST BE 0	The slot number must be 0 to set the power–on/off information for the I/O Unit–B.
7	ERR : ILLEGAL NAME	An invalid or unsupported assignment name has been entered. Enter a correct name.
8	INPUT INVALID	An invalid character string has been entered. Reenter with a correct format.
9	INPOSSIBLE WRITE	An attempt has been made to edit ROM data. ROM data cannot be edited.
10	ERR : ADDRESS ALREADY ASSIGNED	The specified address is already assigned. Assign another address or retry after deleting the existing data.
11	ERR : ADDRESS OVER	An address exceeds the upper limit (X127, Y127). Check the addresses used for the unit to be set.
12	ERR : SLOT ALREADEY DEFINED	The specified slot is already assigned. Check the existing data.
13	WARN : SLOT ALREADY DEFINED	The specified slot is already assigned. Check the existing data.
14	ERR : UNIT TYPE MISMATCH (IN OR OUT)	An X address cannot be assigned to an output module. A Y address cannot be assigned to an input module.
15	ERR : UNIT TYPE MISMATCH (MODEL)	I/O Unit–A and I/O Unit–B are assigned in the same group. I/O Unit–A and I/O Unit–B cannot exist in the same group.

5.4 SYMBOL DATA SETTING (SYMBOL)

A signal name (within 6 alphanumeric characters) can be attached to I/O signals and internal relays employed in sequence programs.

Also, a comment (within 30 alphanumeric characters) can be attached to the relay coils in addition to the symbol name.

Symbol data and comment are displayed together with a ladder diagram on the CRT/MDI screen as follows.



If symbol data and comment are defined in signal addresses of the program, the signal name and comment are displayed as Shown in the above figure.

If neither symbol data nor comment is defined at an address, the address is displayed as it is.

A maximum of 64 KBytes can be used for the ladder, symbol, comment, and data. After the program is initialized, the capacity of the symbol area and that of the comment area are usually 32KB (28KB for PMC–NB) each. When additional data is entered causing the total amount of data in either area to exceed 32KB (28KB for PMC–NB), the area is automatically extended in 1KB units.

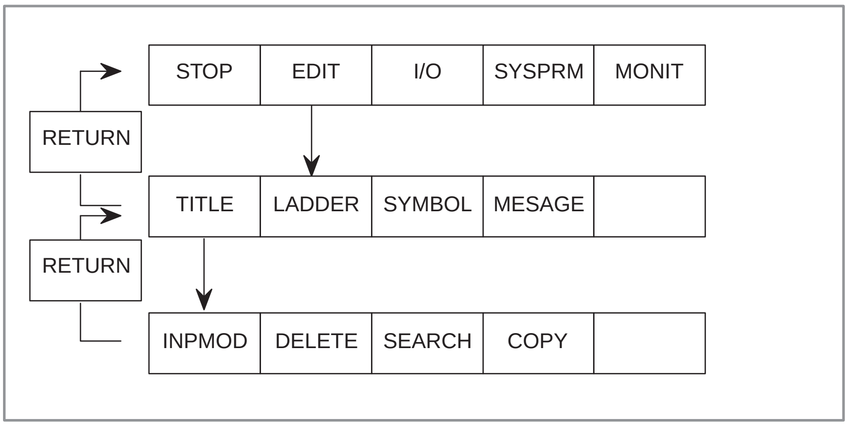


Fig. 5.4(a) Setting and display of symbol data

When soft key [SYMBOL] is pressed, the following screen is displayed, and the soft key operation is done hereafter.

```
SYMBOL & COMMENT 001
ADDRESS SYMBOL COMMENT
      *---------* *------*-------*--------------*
```

Fig. 5.4(b) Symbol data screen

5.4.1 Symbol Data and Comment Input

Refer to Fig. 5.4 (a) and Fig. 5.4 (b). Input symbol data and comment on the screen shown in Fig. 5.4 (b).

1 Press <INPUT> key after inputting an address where a symbol and a comment are to be set.

The input address is set to the ADDRESS column of Fig. 5.4 (b), and the cursor shifts to the address. The input addresses are arrange and set in the alphabetic sequence, and they can be inserted halfway.

2 For setting a symbol, shift the cursor rightward by using the cursor key.

3 After setting symbol data (within 6 alphanumeric characters), press INPUT key.

The symbol data are set in the SYMBOL column of Fig. 5.4 (b), and the cursor shifts to the COMMENT column.

4 For inputting a comment, set comment data (within 30 alphanumeric characters), and press INPUT key.

Comment data are set to the COMMENT column in Fig. 5.4 (b).

Repeat steps **1** to **4** hereafter.

5.4.2 Symbol Data Search (SRCH)

Display the screen shown in Fig. 5.4 (b) and search symbol data

(1) After setting an address or symbol data to be searched, press soft key [SRCH].

Specified address or symbol data is searched and displayed on the screen.

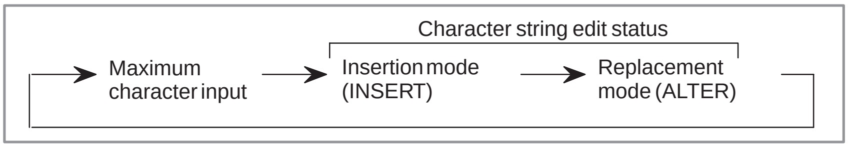
The cursor shifts to the corresponding address part.

5.4.3 Delete of Symbol Data and Comment

Move the cursor to the address to be deleted in the ADDRESS column of Fig. 5.4 (b), and press soft key [DELETE].

5.4.4 Editing Character Strings of Symbol Data and Comment Data

The edit modes can be changed by pressing the [INPMOD] soft key as follows:



”INSERT” is displayed on the screen in the insertion mode. ”ALTER” is displayed on the screen in the replacement mode.

- When the <INPUT> key is pressed

  (1) In the character string edit status

  Insertion mode : The entered character is inserted at the cursor. If the [INPUT] key is pressed after no character is entered, one space is inserted.

  Replacement mode : The character at the cursor is replaced with the entered character. If the [INPUT] key is pressed after no character is entered, the character at the cursor is replaced with one space.

  (2) When the length of the cursor is the same as the number of characters that can be entered.

  The original character string are replaced with the entered characters.

- When the <DELETE> key is pressed

  (1) In the character string edit status

  Insertion mode : The character at the cursor is deleted.

  Replacement mode : The character at the cursor is replaced with a space.

  (2) When the length of the cursor is the same as the number of characters that can be entered

  The character string at the cursor is deleted.

5.4.5 Function for Editing Symbol Data and Comment Data at One Time

An address, symbol, and comment can be entered at one time.

(1) Editing the symbol and comment assigned to address not defined

Enter an address, enter the symbol and comment enclosed in characters other than alphanumeric characters, then press the <INPUT> key.

The cursor may be located anywhere. A comment can be omitted.

Example) G0.4 / *EMG / EMERGENCYSW/ <INPUT> key
Address Symbol Comment

(2) Editing the symbol and comment assigned to address already defined

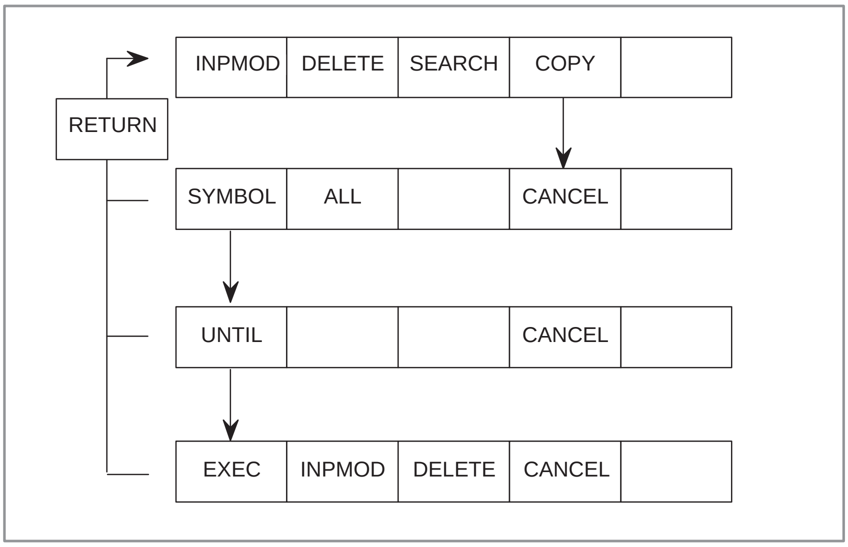
Move the cursor to the desired line of the address whose symbol and comment are to be edited.

Next, enter the symbol and comment enclosed in characters other than alphanumeric characters.

Then, press the <INPUT> key. A comment can be omitted.

5.4.6 Function of Copying Symbol and Comment Data

Copy the specified data to re–edit and register it.



(1) Press the [COPY] soft key.

(2) Select data to be copied with the corresponding soft key.

[ALL] : The address, symbol data, and comment data are copied.

[SYMBOL] : The address and symbol data are copied.

When either of the above soft keys is pressed, the line at the cursor is specified as the beginning of the range of the data to be copied.

(3) Specify the range with the [↓] and [↑] cursor keys.

- A range of more than 15 lines cannot be specified. Up to 15 lines can be displayed on one screen.
- A range cannot be specified at a position above the cursor position. When the copy range of the data is specified, the data is displayed differently.

(4) Press the [UNTIL] soft key to determine the copy range.

(5) Edit the address and symbol data according to the procedure described in Sections 5.4.1 and 5.4.4.

(6) When updating the data is completed, press the [EXEC] soft key to register the copied data.

5.5 MESSAGE DATA SETTING (MESSAGE)

Message data are used for PMC functional instruction DISPB (SUB 41).

The setting and display methods are as shown below.

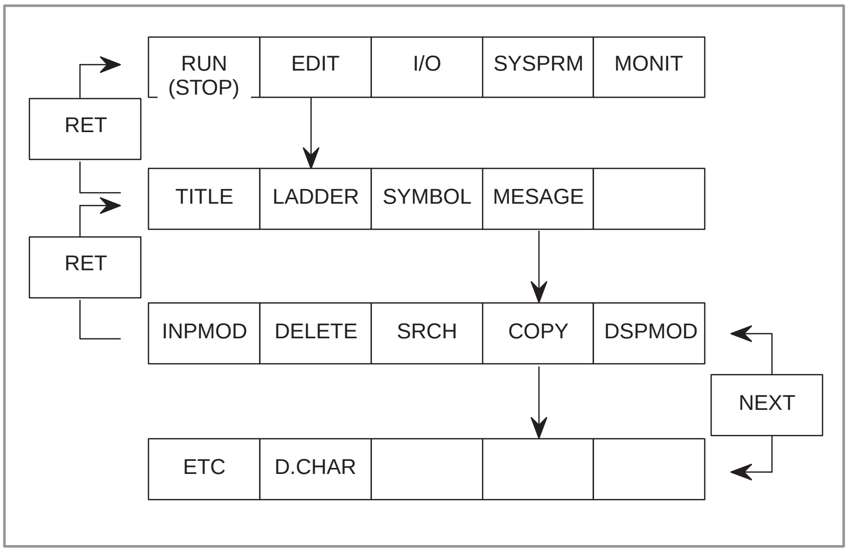


Fig. 5.5 (a)

When soft key MESAGE is pressed, the next screen is displayed, and setting operation can be done hereafter.

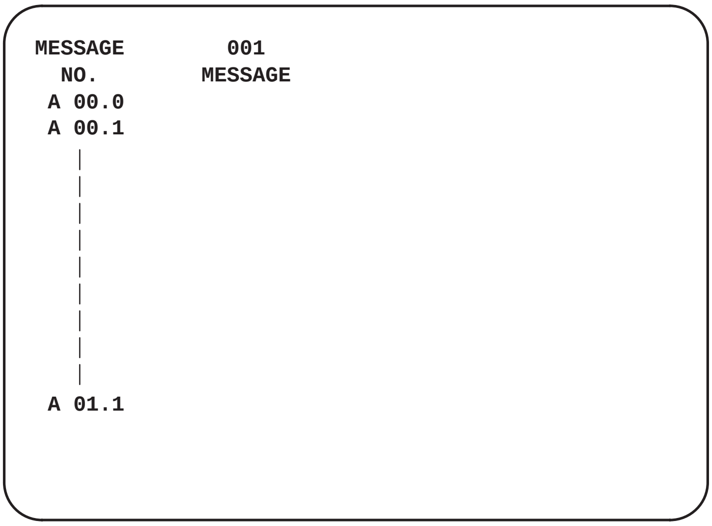


Fig. 5.5 (b) Message data screen

After initialization, the capacity of the message area is approx. 2.1KB. When additional data is entered causing the total amount of data in the area to exceed 2.1KB, the message area is automatically extended in 1KB units to a maximum of 64KB.

5.5.1 Message Data Input

Refer to Fig. 5.5 (a) and Fig. 5.5 (b). Display the screen shown in Fig. 5.5 (b).

1 Display a number to set a message data by using PAGE key.

2 Shift the cursor to this number by the cursor key.

3 After setting message data, press INPUT key.

If the message data has already been entered, it is deleted and the set data is entered.

Repeat steps **1** to **3** hereafter.

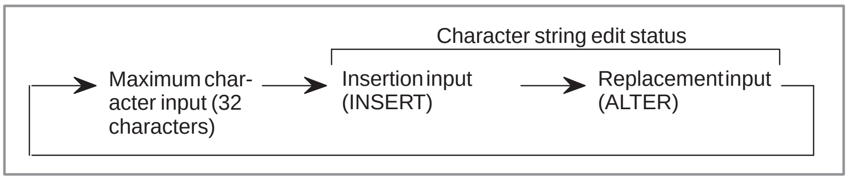
5.5.2 Searching for an Address (SRCH)

(1) Specify an address to be searched for, and press the [SRCH] soft key.

Then, the specified address is searched for and displayed on the screen.

5.5.3 Editing a Character String in Message Data

Edit modes can be changed by pressing the [INPMOD] soft key as follows:



The edit mode is changed every time the [INPMOD] soft key is pressed.

[INSERT] is displayed on the screen in the insertion mode.

[ALTER] is displayed on the screen in the replacement mode.

- When the <INPUT> key is pressed

  (1) In the character string edit status

  Insertion mode : The entered character is inserted at the cursor.

  Replacement mode : The character at the cursor is replaced with the entered character.

  (2) When the maximum number of characters are entered

  The original character string at the cursor is replaced with the entered characters.

- When the <DELETE> key is pressed

  (1) In the character string edit status

  Insertion mode : The character at the cursor is deleted.

  Replacement mode : The character at the cursor is replaced with a space.

  (2) When the maximum number of characters are entered

  The character string at the cursor is deleted.

  (3) When the cursor is located at the address field

  The entire message data specified at the address is deleted.

5.5.4 Input with a Katakana Identification Code

If no “@” key on the MDI key, pressing the [ETC] soft key enables the operator to enter the data enclosed between at signs (@).

When the soft key is pressed, ”ETC CODE” is displayed on the screen.

5.5.5 Copying Message Data (COPY)

Move the cursor to the message number to be copied and press the [COPY] key.

Then press the [EXEC] key after moving the cursor to the message number in which it is copied.

5.5.6 Inputting a Multi–byte Character (D.CHAR)

The input mode becomes multi–byte character by pressing the [D.CHAR] key.

(@02, 01@ are added to input data automatically.)

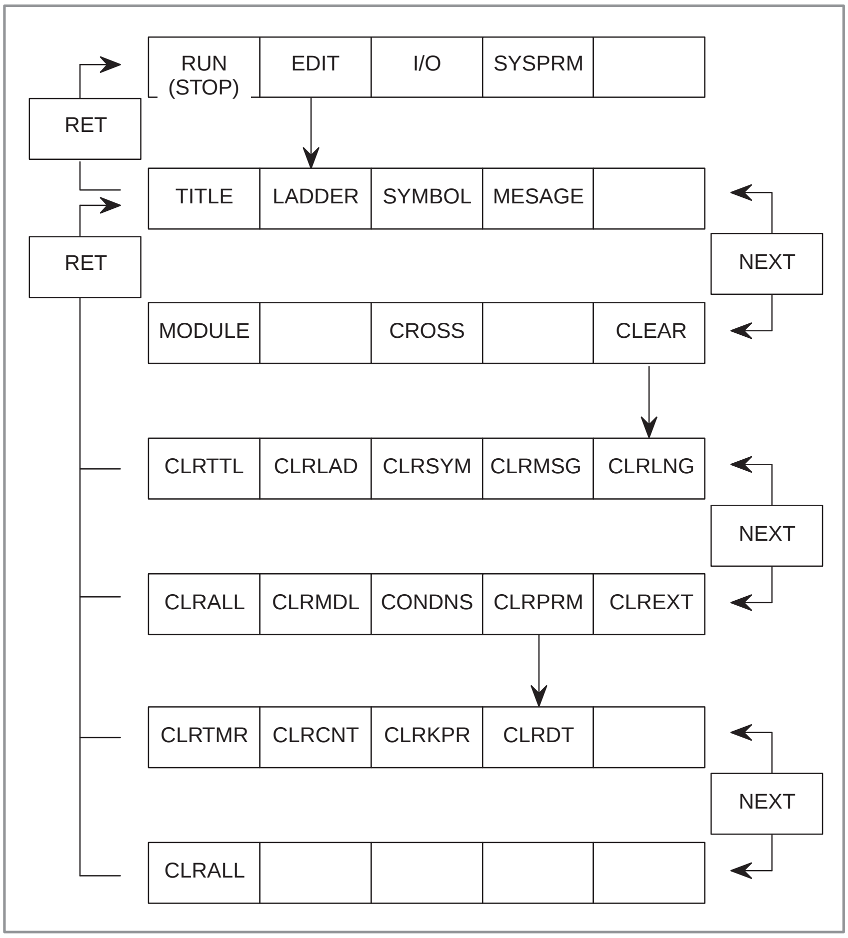
For example, “4873 [INPUT]” is processed as “@02487301@”.

5.5.7 Displaying Input Code (DSPMOD)

The ASCII code enclosed with @ characters is displayed in the form of screen display by pressing the [DSPMOD] key.

Example) Katakana : “@B6C532@” → “カナ2” is displayed.
Multi–byte character : “@0248733E6F44643B5F01@100”
→ “非常停止100” is displayed.

5.6 CLEARING THE SEQUENCE PROGRAM AND CONDENSATION OF THE SEQUENCE PROGRAM



5.6.1 Clearing the Sequence Program

Clears each data in the sequence program

The function of the key is as follows:

(1) [CLRTTL] : Clears the title data.

(2) [CLRLAD] : Clears the ladder program.

(3) [CLRSYM] : Clears the symbol and comment data. If the extend symbol and comment data is cleared, the field is restored to the original size.

(4) [CLRMSG] : Clears the message data. If the extend message data is cleared, the field is restored tothe original size.

(5) [CLRLNG] : The C language area is cleared. Clear the C language area before transferring a C program. When a C language board is installed in the Series 16*i*/18*i*, this item is displayed.

(6) [CLRALL] : Clear all data described in the above (1) to (4). Clear also the C language programs for models which create C language programs, such as models PMC–SC, SC3, SC4, NB, and NB2. Press this soft key when the message "PLEASE CLEAR ALL" is displayed.

(7) [CLRMDL] : Clears the I/O module data.

(8) [CONDNS] : Compress the sequence program in 1KB units. The detail will be explained chapter 5.6.2.

(9) [CLRPRM] : Clears each parameter data. The detail will be explained chapter 5.6.3.

(10) [CLREXT] : Clears the expand nonvolatile memory (valid for PMC–SC/SC3/SC4/NB/NB2)

NOTE

When using a system that incorporates flash ROM, clear the flash ROM before writing to it. If the power is turned on again without performing this operation, sequence program data is not cleared.

5.6.2 Compress the Sequence Program

Compresses the sequence program in 1KB units.

(1) [CONDNS] : Compresses the unused area in the message, symbol, or comment area in the sequence program in 1KB units when the capacity of the unused area extends 1KB. The unused area, which is the size less than 1KB, will not be compressed.

5.6.3 Clearing the PMC Parameter

Clears each PMC parameter.

The function of the key is as follows:

(1) [CLRTMR] : Clears timer data.

(2) [CLRCNT] : Clears counter data.

(3) [CLRKPR] : Clears keep relay data.

(4) [CLRDT] : Clears data table.

(5) [CLRALL] : Clear all data described in the above (1) to (4).

NOTE

These functions require the same condition as PMC data setting in operation For [CLRALL], all conditions are required. See “Chapter II.4. PMC PARAMETER SETTING AND DISPLAY”

5.7 CROSS REFERENCE DISPLAY

The cross reference is displayed for PMC address and functional instruction used in a sequence program.

Cross reference display has the following functions.

(1) Display NET number by specifying the PMC address.

(2) Display the address list by specifying PMC address name (G, F, Y, · ·).

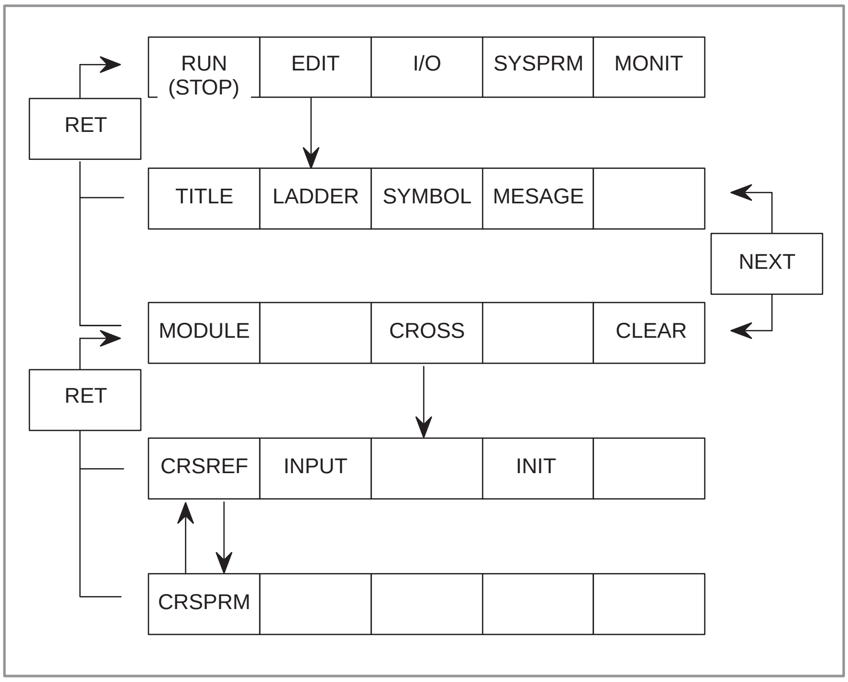
(3) Display a functional instruction list.

(4) Display NET number by specifying the functional instruction number.

5.7.1 Operation

Pressing the [CROSS] key displays the cross reference screen for setting parameters.

Press soft key [CRSREF] in the parameter setting screen for displaying the cross reference of address and functional instruction in use. Press soft key [CRSPRM] to return to the parameter setting screen from cross reference display.



5.7.2 Parameter Setting Screen

To display cross reference, the address and reference type need to be specified on the parameter screen.

Display the reference of addresses which are used.

1. In “SELECT CROSS TYPE”, input “1”.
2. In “1: REFERENCE ADDRESS”, input addresses which should be displayed. (maximum number of input: 8)
3. Press [CRSREF] key.

   The address, symbol, relay and the NET number will be displayed as shown in Fig. 5.7.2 (b).

```
PMC CROSS REFERENCE
 SELECT CROSS TYPE  =  1
 ( 1:ADDRESS 2:ADRS KIND 3:FUNCTION.NO )
    1:REFERENCE  ADDRESS
              1 =  X0000.0   5 =
              2 =            6 =
              3 =            7 =
              4 =            8 =
    2:ADRS KIND     =
      ( G /F /Y /X /A /R /K /C /D /P /L )
    3:FUNCTION.NO   =     ( ALL=0 )



 [CRSREF] [INPUT ] [       ] [ INIT  ] [       ]
```

Fig. 5.7.2 (a) Cross reference setting (TYPE1)

```
PMC CROSS REFERENCE


    X0000.0         ABCDE
   ──┤ ├── :     1       2
   ──( )── :     4       32






 [CRSPRM] [       ] [       ] [       ] [       ]
```

Fig. 5.7.2 (b) Cross reference display (TYPE1)

Display the reference of each address name

1. In “SELECT CROSS TYPE”, input “2”.
2. In “2: ADRS KIND”, input the address name.
3. Press [CRSREF] key.

   The bit/byte addresses and the related symbol in the sequence program will be displayed as shown in Fig.5.7.2 (d).

```
PMC CROSS REFERENCE
 SELECT CROSS TYPE  =  2
 ( 1:ADDRESS 2:ADRS KIND 3:FUNCTION.NO )
    1:REFERENCE  ADDRESS
              1 =  X0000.0   5 =
              2 =            6 =
              3 =            7 =
              4 =            8 =
    2:ADRS KIND     =  X
      ( G /F /Y /X /A /R /K /C /D /P /L )
    3:FUNCTION.NO   =     ( ALL=0 )



 [CRSREF] [INPUT ] [       ] [ INIT  ] [       ]
```

Fig. 5.7.2 (c) Cross reference setting (TYPE2)

```
PMC CROSS REFERENCE

 HEAD CHARACTER =  X

  USED ADDRESS  =  X0000.0
  SYMBOL NAME   =  ABCDE

                     X0000
                     SYMBOL NOTHING




 [CRSPRM] [       ] [       ] [       ] [       ]
```

Fig. 5.7.2 (d) Cross reference display (TYPE2)

Display a functional instruction list in use

1. In “SELECT CROSS TYPE”, input “3”.
2. In “3: FUNCTION. NO”, input “0”.
3. Press [CRSREF] key.

   The functional instruction name and the functional instruction number in the sequence program will be displayed as shown in Fig.5.7.2 (f).

```
PMC CROSS REFERENCE
 SELECT CROSS TYPE  =  3
 ( 1:ADDRESS 2:ADRS KIND 3:FUNCTION.NO )
    1:REFERENCE  ADDRESS
              1 =  X0000.0   5 =
              2 =            6 =
              3 =            7 =
              4 =            8 =
    2:ADRS KIND     =
      ( G /F /Y /X /A /R /K /C /D /P /L )
    3:FUNCTION.NO   =     ( ALL=0 )



 [CRSREF] [INPUT ] [      ] [ INIT  ] [      ]
```

Fig. 5.7.2 (e) Cross reference setting (TYPE3)

```
PMC CROSS REFERENCE


 FUNCTION NAME    END1(  1)    END2(  2)
                   COD(  7)









 [CRSPRM] [      ] [      ] [      ] [      ]
```

Fig. 5.7.2 (f) Cross reference display (TYPE3)

Display the reference of functional instruction (FUNCTION. NO = number of the functional instruction)

1. In “SELECT CROSS TYPE”, input “3”.
2. In “3: FUNCTION. NO”, input the functional instruction number.
3. Press [CRSREF] key.

   The functional instruction name, functional instruction number and NET number in the sequence program will be displayed as shown in Fig.5.7.2 (g).

```
PMC CROSS REFERENCE


 FUNCTION NO.  =  7
 FUNCTION NAME =  COD
 USED NET NO.  =      6  14









 [CRSPRM] [       ] [       ] [       ] [       ]
```

Fig. 5.7.2 (g) Displaying of cross reference (TYPE3)

6 EXECUTION OF A SEQUENCE PROGRAM

6.1 START AND STOP OF A SEQUENCE PROGRAM

Start and Stop of a sequence program are described as follows.

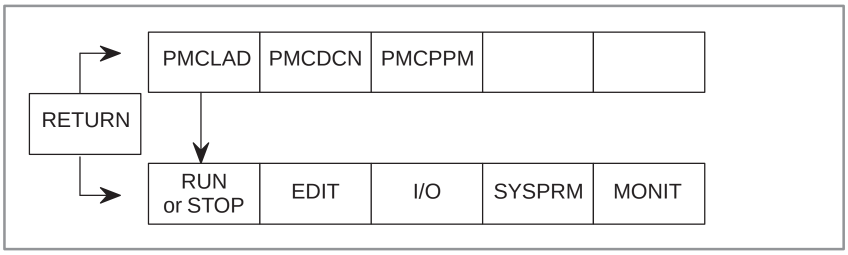


Fig. 6.1 The sequence program execution software key

An operable sequence program is usually automatically started when power is turned on if the program is stored in ROM. However, the program is not started if it is stored in RAM.

(1) Start of a sequence program (RUN)

When a sequence program is at the stopped state, pressing the [RUN] key causes the sequence program to run displaying the software key as [STOP].

The ladder program starts from the beginning. However, whether C–language programs start from the beginning depends on the function selected in advance.

a) When a C–language program starts from the beginning
Functions selected in advance: Ladder editing, reading the system parameter, reading a sequence program using input/output processing

b) When a C–language program does not start from the beginning but restarts from the next step after stopping
Function selected in advance: Functions other than the function in item (a)

> **NOTE**
> Both ladder and C–language programs start from the beginning immediately after the power is turned on.

(2) Stop of a sequence program (STOP)

When a sequence program is at the run state, pressing the [STOP] key causes the sequence program to stop displaying the software key as [RUN].

6.2 STARTING THE SEQUENCE PROGRAM

The sequence program can be automatically started immediately after power–on, when bit 2 of keep relay K17 (PMC parameter) is set to 1.

> **NOTE**
> For the Series 16/18–MODEL B/C, Series 16*i*/18*i*/21*i*–MODEL A, automatic start is specified when bit 2 of K17 is set to 0.

6.3 FORCIBLY STOPPING THE SEQUENCE PROGRAM

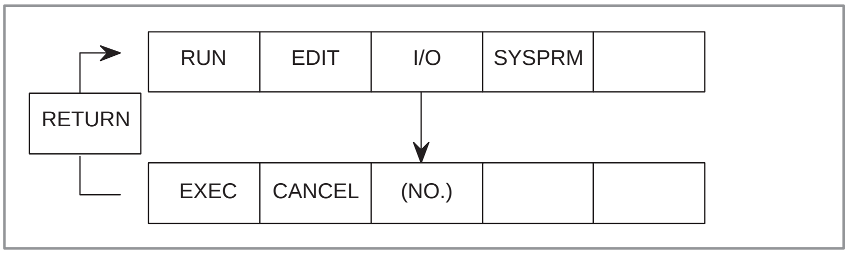
To forcibly stop starting the sequence program in ROM or RAM, at power–on, turn on the power by pressing the [Z] key while pressing the [CAN] key. (Except for PMC–NB/NB2)

This method is effective for locating the error when a system error occurs after power is turned on and when the error may be caused by the sequence program.

Never perform this operation in a normal state.

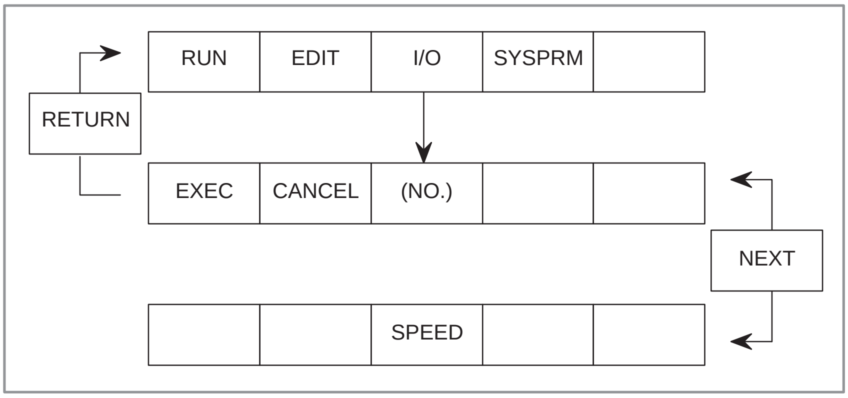
7 WRITING, READING, AND VERIFYING THE SEQUENCE PROGRAM AND PMC PARAMETER DATA

When the [I/O] key is pressed, the sequence program and PMC data are written, read, or collated for the specified device. Operations are performed with cursor keys and soft keys.



7.1 OVERVIEW

When the [I/O] key is pressed, the sequence program and PMC data are written, read, or collated for the specified device. Operations are performed with cursor keys and soft keys.




```
PMC I/O PROGRAM                        MONIT STOP

     CHANNEL   =   1

     DEVICE    = HOST

      DATA KIND =
      (ALL:LADDER + LANGUAGE)
      FUNCTION  =



>
                                             ALM
 [ EXEC ] [CANCEL] [ HOST ] [FDCAS ] [F–ROM ]
```

[Case of FS16/18]

```
PMC I/O PROGRAM                        MONIT STOP

     CHANNEL   =   1

     DEVICE    = HOST

      DATA KIND =

      FUNCTION  =



>

 [ EXEC ] [CANCEL] [ HOST ] [FDCAS ] [ROMWRT]

 [      ] [OTHERS] [SPEED ] [      ] [ COPY ]
```

NOTE
The sequence program can be output while the ladder is being executed, but the output speed is low. When the sequence program is input while the ladder is being executed, the execution of the ladder is automatically stopped.

7.1.1 C Input/Output

Conventionally, C programs have been managed together with ladder programs. For this reason, when only a ladder program is to be modified as shown in Fig. 7.1.1 (a), the program linked with a C program needs to be replaced.
With the FANUC Series 16*i*/18*i*/160*i*/180*i*, ladder programs and C programs are managed separately. So, each program can be edited and replaced independently of the other programs as shown in Fig. 7.1.1 (b).

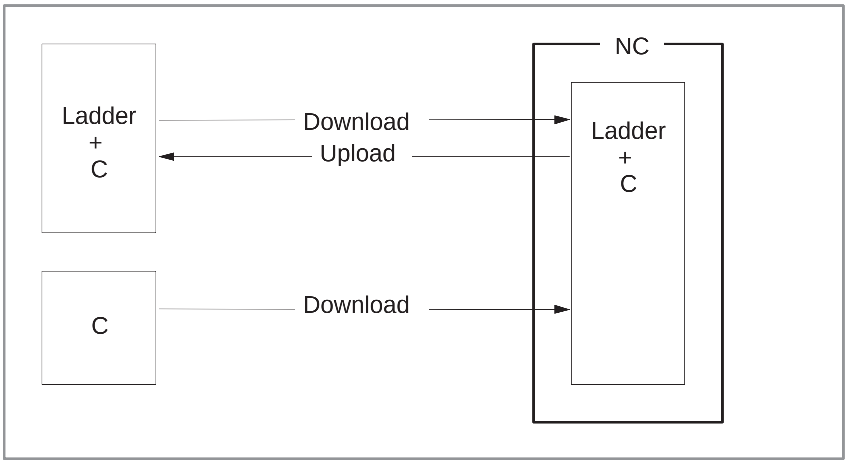


Fig. 7.1.1 (a) Ladder and C structure for FANUC Series 16B/16C

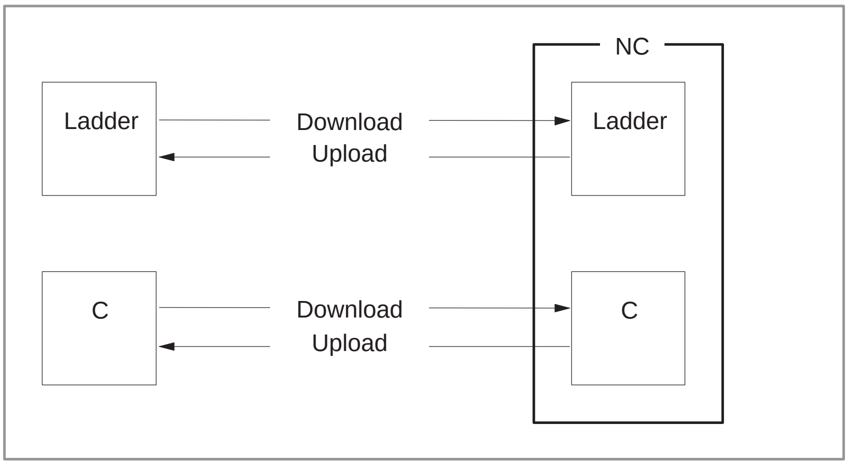


Fig. 7.1.1 (b) Ladder and C structure for FANUC Series 16*i*/18*i*/160*i*/180*i*

7.2 SET ITEMS

(1) CHANNEL

Specify which connector the reader/punch interface (such as RS–232C) is connected to. CHANNEL must be set when HOST, FDCAS, or OTHERS is selected for DEVICE.

(2) DEVICE

Select the device with which the PMC inputs or outputs data, using soft keys.

Soft key	Description
HOST	Transfers data with a FAPT LADDER (P–G, P–G Mate, or personal computer). (See Subsection 7.3.1 for details.)
FDCAS	Transfers data with a FANUC FD cassette. (See Subsection 7.3.2 for details.)
F–ROM	Transfers data with flash EEPROM. This is where the sequence program is stored. (See Subsection 7.3.3 for details.)
M–CARD	Transfers data with a memory card. (See Subsection 7.3.4 or details.)
OTHERS	Transfers data with other input/output devices. (See Subsection 7.3.5 for details.)
SPEED	Used to set transfer conditions when RS–232C is used. (See Subsection 7.3.6 for details.)
ROMWRT	Transfers data with a ROM WRITER. (See Subsection 7.3.6 for details)

NOTE

Some functions cannot be used with some PMC models. See Section 7.3.

(3) DATA KIND

Select the type of output data using soft keys. DATA KIND must be set when FDCAS, M–CARD, or OTHERS is selected for DEVICE.

DATA KIND	CONTENTS
ALL	Output the data of sequence program and C program
LADDER	Output the data of sequence program (Ladder, Symbol, Comment, Message, etc.)
PARAM (Note 1)	Output PMC Parameters (TIMER, COUNTER, KEEP RELAY, DATA TABLE, etc.)
C–LANG (Note 2)	Outputs a C program.

NOTE

1 The conditions of outputting PMC parameters
   1) When sequence program is stopped
      You can input/output them.
   2) When sequence program is executed
      You must satisfy the following conditions.
      Output (WRITE) Set NC to "EDIT" mode.
      Input (READ) . . . Set NC to "Emergency Stop" status, and, set "PWE" of NC parameters to 1.

2 When a C language board is installed in the Series 16*i*/18*i*, the item of C–LANG is displayed, and C program output is enabled.

(4) FUNCTION

Select the direction of data transfer between the PMC and input/output device.

Item	Description
WRITE	Outputs data from the PMC to an input/output device.
READ	Inputs data from an input/output device to the PMC.
COMPARE	Collates data in the PMC and an input/output device. (Note) PMC data cannot be collated.
DELETE	Deletes a file in FDCAS or M–CARD.
LIST	Lists the files in FDCAS or M–CARD.
BLANK	Performs blank check for F–ROM.
ERASE	Clears the data in F–ROM.
FORMAT	Initializes M–CARD (clears all data).

(5) FILE NO.

FILE NO. is displayed when FDCAS or M–CARD is selected for DEVICE. Specify the file number or file name for WRITE, READ, COMPARE, or DELETE processing. Note the following restrictions on the file name when FDCAS or M–CARD is selected for DEVICE:

	FDCAS (FANUC FD CASSETTE)	M–CARD (MEMORY CARD)
Number of characters in the file name	Up to 17 characters following @ or #. The file is written after the existing files.	Up to 8 characters following @ or #(*1).
When the same name as an existing file is specified	An error occurs. Delete the existing file and reoutput the new file.	The new file is written over the existing file (the contents of the existing file are lost).
When –1 is specified for the file name	The file is written after the existing files.	The system names the file and writes it(*2).
When 0 is specified for the file name	The file is written and all the existing files are deleted.	The system names the file and writes it(*2).

NOTE

1 Name the file in the MS–DOS format (up to eight characters for the file name with up to three characters for the extension).
Example) FILE NO. = @12345678.123
FILE NO. = @LADDER.EXE

2 If the file name is not specified, the system names the file as follows:

DATA KIND	File name
ALL	model–name.ALL
LADDER	model–name.LAD
PARAM	model–name.PRM

The model name is PMC–NB for the PMC–NB and PMC–SA for the PMC–SA1 or PMC–SA3.

7.3 OPERATIONS

7.3.1 Transfer to and from a FAPT LADDER

(a) Setting the channel

Move the cursor to "CHANNEL = ." Check that an RS–232C cable is connected to the main board. Enter the number (1 or 2) corresponding to the connector. The correspondence between the connector and CHANNEL is as follows:

CHANNEL = 1 : JD5A (main board)
CHANNEL = 2 : JD5B (main board)

(b) Setting the transfer conditions

Press the [SPEED] soft key and set each condition. See Subsection 7.3.6 for details.

(c) Writing, reading, or collating the sequence program

Item	Operation
DEVICE	Press the [HOST] soft key.
	Press the [EXEC] soft key and to make the NC ready for operation.

Select necessary items on a FAPT LADDER and start transfer.

NOTE
WRITE, READ, or COMPARE is automatically switched by operation on a FAPT LADDER.

7.3.2 Transfer to and from a FANUC FD Cassette

Reads or writes the sequence program, Pascal or C programs, or PMC data.

```
PMC I/O PROGRAM                        MONIT STOP

     CHANNEL   =   1

     DEVICE    = FDCAS

     DATA KIND = ALL
     (ALL:LADDER + LANGUAGE)
     FUNCTION  = WRITE

     FILE NO.  = –1
     (–1:ADD,0:INIT,OR @ NAME)
>
                                           ALM
 [ EXEC ] [CANCEL] [ HOST ] [FDCAS ] [F–ROM ]
```

(a) Setting the channel

Enter the number of the channel used at "CHANNEL = ." See (a) in Subsection 7.3.1 for details.

(b) Setting the transfer conditions

Press the [SPEED] soft key and set each condition. See Subsection 7.3.6 for details.

(c) Writing a file

Item	Operation
DEVICE	Press the [FDCAS] soft key.
FUNCTION	Press the [WRITE] soft key.
DATA KIND	Select the type of data to be output (see (3) in Section 7.2).
FILE NO.	Name the file within 17 characters. –1 is displayed if no name is entered (see (5) in Section 7.2).

Press the [EXEC] soft key to start outputting the file.

(d) Reading a file

Item	Operation
DEVICE	Press the [FDCAS] soft key.
FUNCTION	Press the [READ] soft key.
FILE NO.	Enter the number or name of the file to be input.

Press the [EXEC] soft key to start inputting the file.

(e) Collating a file

Item	Operation
DEVICE	Press the [FDCAS] soft key.
FUNCTION	Press the [COMPAR] soft key.
FILE NO.	Enter the number or name of the file to be collated.

Press the [EXEC] soft key to start collating the file.

Note

PMC data cannot be collated. The data the file is to be collated with depends on the file.

(f) Deleting a file

Item	Operation
DEVICE	Press the [FDCAS] soft key.
FUNCTION	Press the [DELETE] soft key.
FILE NO.	Enter the number or name of the file to be deleted.

Press the [EXEC] soft key to start deleting the file.

(g) Listing the files

Item	Operation
DEVICE	Press the [FDCAS] soft key.
FUNCTION	Press the [LIST] soft key.

Press the [EXEC] soft key to start listing the files.

7.3.3 Storage to Flash ROM

○ : Supported
× : Not supported

Power Mate–D/F/G	Power Mate–H FS20	FS21/ 210MB	FS18	FS16–A	FS16–B FS18–B	FS16–C FS18–C	FS21*i* FS16*i* FS18*i*	FS15B
×	○	○	×	×	○	○	○	○

Formerly, a RAM module or ROM module was necessary for storing programs. Using Flash Memory, however, programs can be ROM–stored on the PMC board.

> **CAUTION**
> 1 If the power is turned off without performing the writing operation, the updated sequence program is not stored.
> 2 The CNC must be placed in the emergency stop state when data is read from or written to Flash Memory.
> 3 Even if the sequence program is cleared with the X and O keys at power–on, the contents of Flash Memory are not cleared. Therefore, when the power is turned on again, the sequence program in Flash Memory is read. To clear the contents of Flash Memory, write Flash Memory after clearing the sequence program with X and O.

```
PMC I/O PROGRAM                                  MONIT STOP

     CHANNEL   =   1

     DEVICE    = F–ROM

      DATA KIND =
      (ALL:LADDER + LANGUAGE)
      FUNCTION  = WRITE

 RAM SIZE = A ( MAX SIZE = B )

 PROGRAM ALREADY EXISTS (EXEC?)
>
                                                      ALM
 [ EXEC ] [CANCEL] [ HOST ] [FDCAS ] [F–ROM ]
```

(a) Writing data to Flash Memory

Item	Operation
DEVICE	Press the [F–ROM] soft key.
FUNCTION	Press the [WRITE] soft key.

Press the [EXEC] soft key to start outputting data.

NOTE
1 Operation in PMC–NB
(a) If data is stored in Flash Memory, a message is displayed to confirm writing.
(b) RAM SIZE indicates the size of the sequence program. MAX SIZE indicates the size of the ROM option in the PMC.
A: 64K bytes, B: 128K bytes, C: 256K bytes,
D: 512K bytes, E: 1M bytes
2 When a C board is installed in the Series 16*i*/18*i*, ladder programs and C programs must be written into Flash Memory separately.
(a) When a ladder program is written into Flash Memory, select [LADDER] from the DATA KIND item.
When a C program is written into Flash Memory, select [C–LANG] from the DATA KIND item.
(b) Select [WRITE] from the FUNCTION item.
(c) Press the [EXEC] soft key to start writing to Flash Memory.

(b) Reading data from Flash Memory

Item	Operation
DEVICE	Press the [F–ROM] soft key.
FUNCTION	Press the [READ] soft key.

Press the [EXEC] soft key to start inputting data.

(c) Comparing data in Flash Memory

Item	Operation
DEVICE	Press the [F–ROM] soft key.
FUNCTION	Press the [COMPAR] soft key.

Press the [EXEC] soft key to start comparing data.

(d) Checking if data is stored in Flash Memory

Item	Operation
DEVICE	Press the [F–ROM] soft key.
FUNCTION	Press the [BLANK] soft key.

Press the [EXEC] soft key to check if data is stored in Flash Memory.

NOTE
Operation in PMC–NB
When data is stored in Flash Memory : BLANK ERROR is displayed.
When no data is stored in Flash Memory : BLANK COMPLETE is displayed.

(e) Deleting data in Flash Memory

Item	Operation
DEVICE	Press the [F–ROM] soft key.
FUNCTION	Press the [ERASE] soft key.

Press the [EXEC] soft key to start deleting data.

NOTE
In FS16B/18B, [READ], [COMPAR], [BLANK] and [ERASE] functions are unavailable.

7.3.4 Storage to a Memory Card

○ : Supported
× : Not supported

Power Mate–D/F/G	Power Mate–H FS20	FS21/ 210MB	FS18	FS16–A	FS16–B FS18–B	FS16–C FS18–C	FS21*i* FS16*i* FS18*i*	FS15B
×	○	○	×	×	○	○	○	○

NOTE
This function is not supported on DPL/MDI of Power Mate–H.

Sequence programs and data are input from or output to a memory card as described below. The memory card to which data is input from or output to can directly send or receive data to or from the programming unit (FAPT LADDER).

The supported function and the kind of memory card is shown as below.

Any kind of card has to be conformed to TYPE 1 to 2 of PCMCIA (The Personal Computer Memory Card International Association) 2.0 (or later) or TYPE 1 to 2 of JEIDA (Japanese Electronics Development Association) 4.0 (or later) . And the format is based on MS–DOS FAT file system. The size of memory–card that can be used is up to 32Mbytes.

The case of FS20,FS18B,FS16B FS18C, FS16C PMC

○ : Supported
× : Not supported

	SRAM Card	Flash Memory Card
Read of a file	○	○
Format of a card	○	×
Write of a file	○	×
Delete of a file	○	×
List of a file	○	○

The case of FS16*i*, FS18*i*, FS15B(PMC–NB)

○ : Supported
× : Not supported

	SRAM Card	Flash Memory Card	
		Supported Card	Unsupported Card
Read of a file	○	○	○
Format of a card	○	○	×
Write of a file	○	○	×
Delete of a file	○	×	×
List of a file	○	○	○

(1) Flash memory card writing

The following kinds of flash memory card are supported.

- Intel Series 2 Flash Memory Cards (or compatible cards)

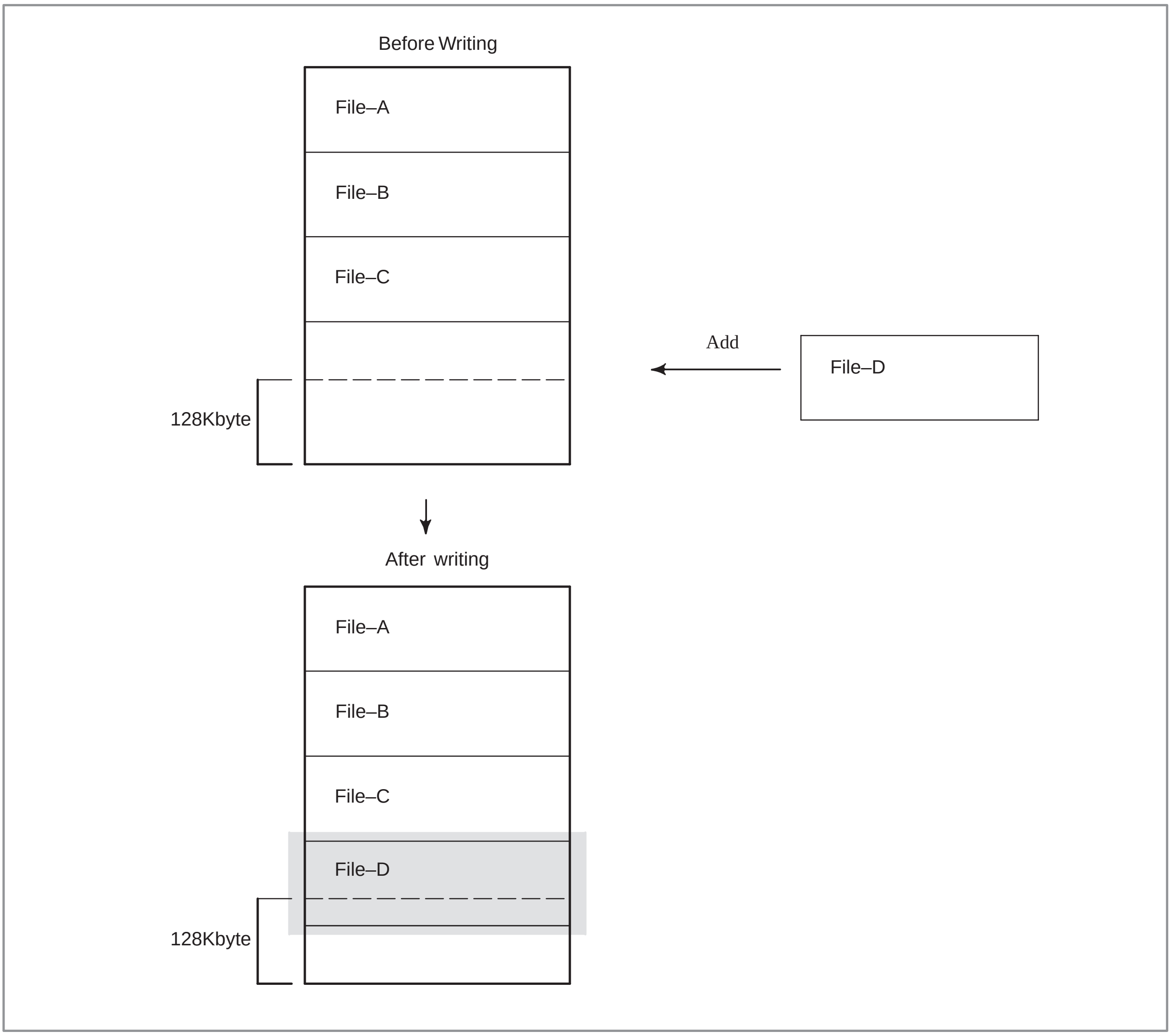
Attribute memory is needed for any card.

Files can be written on the card that is formatted by MS–DOS. But there are following limitations.

- It is impossible to alter a file that is already written.
- A card that is formatted by Flash File System can not be used. (Neither Read nor List)
- The data can not be written in the last 128Kbyte of the card. So, available size of a card is (Card_size – 128Kbyte). Please refer to the following figure.

CAUTION

If the CNC screen display function is used with the FS160*i*/180*i*/210*i* incorporating a combination display/PC, memory card input/output is impossible.



"CLOSE ERROR" is displayed and File–D cannot be saved.

In the part of the ,the data of File–D is written. But "read" and "list" functions are not available for File–D.

After this operation, any file cannot be written to this card.

There are following limitations due to the system that formats the flash memory card.

(a) When the card formatted and written files by FANUC products is used by other systems.

	Ramu–zou Note1)	CardPro Note2)
Read of a file	○	○
Add of file	Not supported function	×
List of file	○	○

NOTE
1 Ramu–zou is a memory card reader/writer that is made by ADTEK SYSTEM SCIENCE.
2 CardPro is a memory card reader/writer that is made by Data I/O.

(b) When the card formatted and written files by other system is used by FANUC products.

	Ramu–zou	CardPro Note3)
Read of a file	○	○
Add of file	○	×
List of file	○	○

NOTE
If you use the CardPro to format a flash memory card, type the following command.
CPFORMAT drive–name: /F:FLASHFAT /NOCIS

(2) Operation

The operation is almost the same as Subsection 7.3.2 except that steps (a) and (b) are not necessary for a memory card.

```
PMC I/O PROGRAM                          MONIT STOP

     CHANNEL   =   1

     DEVICE    = M–CARD

     DATA KIND = PARAM
     (ALL:LADDER + LANGUAGE)
     FUNCTION  = WRITE

     FILE NO. = –1

     (–1:ADD, 0:INIT, OR@ NAME)
>
                                              ALM
 [M–CARD] [OTHER] [      ] [      ] [      ]
```

(a) Formatting the memory card

Item	Operation
DEVICE	Press the [M–CARD] soft key.
FUNCTION	Press the [FORMAT] soft key.

Press the [EXEC] soft key to start formatting.

(b) Writing a file

Item	Operation
DEVICE	Press the [M–CARD] soft key.
FUNCTION	Press the [WRITE] soft key.
DATA KIND	Select the type of data to be output. (See (3) in Section 7.2)
FILE NO.	Name the file within 8 characters. –1 is displayed if no name is entered. (See (5) in Section 7.2)

Press the [EXEC] soft key to start outputting the file.

If the file name is not specified, the system names the file as follows:

DATA KIND	File name
ALL	model–name.ALL
LADDER	model–name.LAD
PARAM	model–name.PRM

The model name is PMC–NB for the PMC–NB and PMC–RA for the PMC–RA1 or RA3.

(c) Reading a file

Item	Operation
DEVICE	Press the [M–CARD] soft key.
FUNCTION	Press the [READ] soft key.
FILE NO.	Enter the number or name of the file to be input.

Press the [EXEC] soft key to start inputting the file.

(d) Collating a file

Item	Operation
DEVICE	Press the [M–CARD] soft key.
FUNCTION	Press the [COMPAR] soft key.
FILE NO.	Enter the number or name of the file to be collated.

Press the [EXEC] soft key to start collating the file.

NOTE
PMC data cannot be collated.

(e) Deleting a file

Item	Operation
DEVICE	Press the [M–CARD] soft key.
FUNCTION	Press the [DELETE] soft key.
FILE NO.	Enter the number or name of the file to be deleted.

Press the [EXEC] soft key to start deleting the file.

(f) Listing the files

Item	Operation
DEVICE	Press the [M–CARD] soft key.
FUNCTION	Press the [LIST] soft key.

Press the [EXEC] soft key to start listing the files.

7.3.5 Data Input to and Output from other Devices

○ : Supported
× : Not supported

Power Mate	FS20	FS21/ 210MB	FS18	FS16	FS18B	FS16B	FS16–C FS18–C	FS21*i* FS16*i* FS18*i*	FS15B
○	○	○	○	○	○	○	○	○	○

Reads or writes the sequence program, Pascal or C programs, or PMC data.

```
PMC I/O PROGRAM                      MONIT STOP

     CHANNEL   =   1

     DEVICE    = OTHERS

      DATA KIND = ALL
      (ALL:LADDER + LANGUAGE)
      FUNCTION  = WRITE



>
                                           ALM
 [M-CARD] [OTHERS] [SPEED ] [      ] [      ]
```

(a) Setting the channel

Enter the number of the channel used at ”CHANNEL = .” See (a) in Subsection 7.3.1 for details.

(b) Setting the transfer conditions

Press the [SPEED] soft key and set each condition. See Subsection 7.3.6 for details.

(c) Outputting data (PMC to input/output device)

Item	Operation
DEVICE	Press the [OTHERS] soft key.
FUNCTION	Press the [WRITE] soft key.
DATA KIND	Select the type of data to be output (see (3) in Section 7.2).
	Set the input/output device so that it is ready to accept data (wait state).

Press the [EXEC] soft key to start outputting data.

(d) Inputting data (input/output device to PMC)

Item	Operation
DEVICE	Press the [OTHERS] soft key.
FUNCTION	Press the [READ] soft key.
	Press the [EXEC] soft key and wait until data input finishes.

The input/output device starts outputting data.

CAUTION

If DATA ERROR is displayed when a C program is written with the Series 16*i*/18*i*, perform the following:

1 Clear the C language area by pressing soft keys [EDIT], [CLEAR], [CLRLNG], then [EXEC].
2 Read the C program again.
3 On the system parameter screen, set LANGUAGE ORIGIN.
4 Write the C program into flash ROM.

(e) Collating data

Item	Operation
DEVICE	Press the [OTHERS] soft key.
FUNCTION	Press the [COMPAR] soft key.
	Press the [EXEC] soft key and wait until data collation finishes.

The input/output device starts outputting data.

NOTE

PMC data cannot be collated. The data the file is to be collated with depends on the file.

7.3.6 Setting the Transfer Speed ([SPEED] Soft Key)

○ : Supported
× : Not supported

Power Mate	FS20	FS21/ 210MB	FS18	FS16	FS18B	FS16B	FS16–C FS18–C	FS21*i* FS16*i* FS18*i*	FS15B
○	○	○	○	○	○	○	○	○	○

```
PMC SPEED OTHERS                          MONIT STOP

     BAUD RATE  = 3
      (0:1200,1:2400,2:4800,3:9600,4:19200)

      PARITY BIT = 0
      (0:NONE,1:ODD,2:EVEN)

      STOP BIT   = 1
      (0:1BIT,1:2BIT)

      WRITE CODE = 1
      (0:ASCII,1:ISO)
>
                                                ALM
 [INPUT ] [      ] [      ] [      ] [ INIT ]
```

The items shown above must be set when RS–232C is used for communication. Move the cursor to each item and enter a number. Pressing the [INIT] soft key sets each item to the initial value. “WRITE CODE = ” is displayed only when OTHERS is selected for DEVICE.

The table below lists the setting for communication with a FAPT LADDER.

Item	Setting on the personal computer	Setting on the PMC (SPEED screen)
Baud rate (bps)	9600	BAUD RATE = 3 (9600bps)
Character length	8 bits	————
Parity check	No parity	PARITY BIT = 0 (NONE)
Number of stop bits	2 bits	STOP BIT = 1 (2BIT)
X parameter	None	————

7.3.7 Transfer to and from a ROM WRITER

○ : Supported
× : Not supported

Power Mate	FS20	FS21/ 210MB	FS18	FS16	FS18B	FS16B	FS16–C FS18–C	FS21*i* FS16*i* FS18*i*	FS15B
×	×	×	○	○	×	×	×	×	×

Reads or writes the sequence program, Pascal or C programs, or PMC data.

This function is valid for the built–in programer function.

(a) Writing a file

Item	Operation
DEVICE	Press the [ROMWRT] soft key.
FUNCTION	Press the [WRITE] soft key.

Press the [EXEC] soft key to start outputting the data.

(b) Reading a file

Item	Operation
DEVICE	Press the [ROMWRT] soft key.
FUNCTION	Press the [READ] soft key.

Press the [EXEC] soft key to start inputting the data.

(c) Collating a file

Item	Operation
DEVICE	Press the [ROMWRT] soft key.
FUNCTION	Press the [COMPAR] soft key.

Press the [EXEC] soft key to start collating the data.

7.3.8 Notes on Using an MDI Keyboard without Cursor Keys (when using the FS20 PMC–MODEL SA1/SA3)

When the machine tool builder creates a MDI keyboard which has no cursor keys on the PMC–MODEL SA1/SA3 of the FS 20, note the following methods of operation. Ladder diagrams cannot be edited using the ladder–diagram–edit memory card.

On each setting screen, when an item is specified, the cursor automatically moves to the next item to be specified. When the item at the cursor need not be modified, specify the same value again. When the item at the bottom of the screen has been specified, the cursor automatically moves to the item at the top of the screen (CHANNEL setting). When the return key (leftmost soft key) is pressed to exit from the I/O screen, the cursor automatically moves to the top of the screen. Examples of setting items are shown below.

Example 1) When a ladder program is output to an off–line programmer (such as the P–G or a personal computer)

1 CHANNEL setting : Enter the desired channel number, then press the <INPUT> key or [(NO.)] key. To use the current value, just press the <INPUT> key or [(NO.)] key.

2 DEVICE setting : Press the [HOST] key. The cursor returns to the CHANNEL setting position to enable CHANNEL setting.

Example 2) When a ladder program is written into an F–ROM

1 CHANNEL setting : No specification required. To move the cursor, perform the operation described in **1** of Example 1.

2 DEVICE setting : Press the [F–ROM] key.

3 FUNCTION setting : No specification required. To change the CHANNEL setting, press the [WRITE] key to return the cursor to the CHANNEL setting position.

Example 3) When a ladder program or a PMC parameter is read from or written into an FDCAS (M–CARD)

1 CHANNEL setting : See **1** of Example 1 (or **1** of Example 2).

2 DEVICE setting : Press the [FDCAS] ([M–CARD]) key.

3 DATA KIND setting: Press the [LADDER] key for ladder operation. Press the [PARAM] key for PMC–parameter operation.

4 FUNCTION setting : Press the [READ]/[WRITE] key.

5 FILE NO. setting : Enter the desired file number or file name, then press the <INPUT> key or [EXEC] key. When the current value is used, just press the <INPUT> key. The cursor automatically returns to the CHANNEL setting position. The setting can be modified.

In each example, pressing the [EXEC] key after setting data executes the corresponding processing.

7.4 I/O ERROR MESSAGES

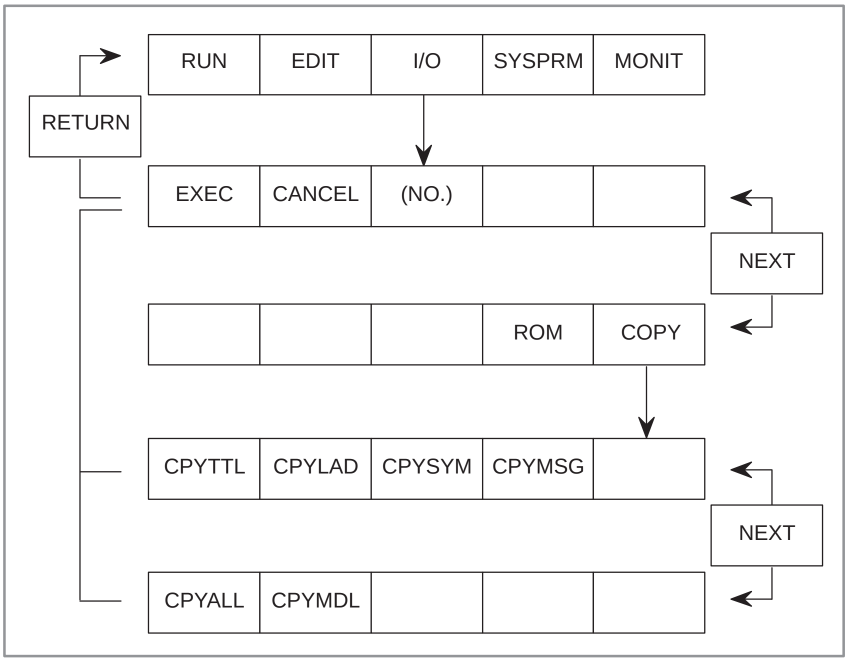
	Message	CONTENTS → OPERATION
FLSH ROM	PROGRAM ALREADY EXISTS	A program is already stored in the FLASH ROM (during blank check).
	PROGRAM ALREADY EXISTS (EXEC ?)	A program is already stored in the FLASH ROM (during writing or deleting data). Action) Press the EXEC key again when the message is displayed. Data is then written or deleted.
	PROGRAM NOTHING	No program is in the FLASH ROM.
	ERASE ERROR	The FLASH ROM is faulty and must be replaced. Consult your FANUC service office.
	WRITE ERROR	
	READ ERROR	
	ANOTHER USED	The FLASH ROM is being used by a device other than the PMC.
	MUST BE IN EMG STOP NOT EMG STOP	The CNC is not in the emergency stop state.
	NO OPTION	There is no ROM cassette option.
	SIZE ERROR	The size of the program exceeds the FLASH ROM size (during writing of the sequence program). Action) Use the CONDENSEM function (EDIT/CLEAR screen). If the error persists, the FLASH ROM size must be increased. The size of the program exceeds the RAM size (during reading of the sequence program). Action) The RAM size must be increased.
HOST.FDCAS.OTHERS	I/O OPEN ERROR nn	nn = –1: The RS–232C interface is being used by a device other than the PMC. Action) Check if another device is using the RS–232C interface. Check that, on the online setting screen (Section 8.5.1 in Part III), NOT USE is set for the RS–232C item. nn = 6: There is no RS–232C option. nn = 20: The RS–232C interface is connected incorrectly. Action) Check that the connection and the setting, such as channel and baud rate, are correct.
	I/O WRITE ERROR nn	nn = 20: The RS–232C interface is connected incorrectly. Action) Check that the connection and the setting, such as channel and baud rate, are correct. nn = 22: Communication cannot be performed normally. Action) Check if a cable is disconnected.
	I/O READ ERROR nn	nn = 20: The RS–232C interface is connected incorrectly. Action) Check that the connection and the setting, such as channel and baud rate, are correct. nn = 22: Communication cannot be performed normally. Action) Check if a cable is disconnected.
	ADDRESS IS OUT OF RANGE (xxxxxx)	Data other than that stored in the PMC debugging RAM area has been transferred. xxxxxx: Transfer address
	DATA ERROR	Invalid data was read. Action) Check the cable and setting (SPEED). When a C program is read into the Series 16*i*/18*i*: Action) Clear the C language area by pressing soft keys [EDIT], [CLEAR], [CLRLNG], then [EXEC].
	PROGRAM DATA ERROR	Data output contains an error Action) On the alarm screen, check the details of the alarm.

	Message	CONTENTS → OPERATION
MEMORY CARD	CREATE ERROR	The file name is invalid. Action) Name the file is the MS–DOS format (see(5) of Section 7.2).
	NO MORE SPACE WRITE ERROR	The memory card capacity is insufficient. Action) Replace the memory card or delete unnecessary files and retry.
	NOT READY	The memory card is not mounted. Action) Confirm if the memory card is mounted correctly.
	MOUNT ERROR	The memory card is not formatted. Action) Format the memory card (see (a) of Subsection 7.3.4).
	WRITE PROTECT	The memory card is protected. Action) Remove the protection of the memory card.
	BATTERY ALARM	The battery of the memory card is not enough. Action) Exchange the battery of the memory card.
	FILE NOT FOUND	Specified file number or file name is not found. Action) Confirm the file number or the file name by LIST.
	DELETE ERROR	The file cannot be deleted. Action) Change the attribute of the file.
	PROGRAM ALREADY EXISTS	The file name already exists. Action) Change to other file name.
	I/O WRITE ERROR nn I/O READ ERROR nn I/O COMPARE ERROR nn I/O DELETE ERROR nn I/O LIST ERROR nn I/O FORMAT ERROR nn	nn=30 : The memory card is not mounted. Action) Confirm if the memory card is mounted correctly. nn=31 : The data cannot be written to the memory card. Action) Remove the protection of the memory card. Exchange the memory card for the S–RAM card. nn=32 : The battery of the memory card is not enough. Action) Exchange the battery of the memory card. nn=102: The memory card capacity is insufficient. Action) Replace the memory card or delete unnecessary files and retry. nn=135: The memory card is not formatted. nn=105: ditto Action) Format the memory card. nn=114: Specified file is not found. Action) Confirm the file number or the file name by LIST. nn=115: Specified file is protected. Action) Confirm the attribute of the file.
ROM WRITER	SIZE OVER WRITE	The ROM size is smaller than the program size. Response) Increase the ROM size.
	ROM WRITER ERROR nnnnn	An error has occurred in the ROM writer. Response) Refer to the “ROM Writer Operator’s Manual.”
Common	COMPARE ERR XXXXXX = AA:BB CONT?(Y/N)	The data between DEVICE and PMC is different. XXXXXX : Address aa : The data in PMC bb : The data in DEVICE Action) If you continue it, press Y key. Otherwise, press N key.
	DATA ERROR	Invalid data was read. Action) Check the cable and setting (SPEED). When a C program is read into the Series 16*i*/18*i*: Action) Clear the C language area by pressing soft keys [EDIT], [CLEAR], [CLRLNG], then [EXEC].
	PROGRAM DATA ERROR	Data output contains an error. Action) On the alarm screen, check the details of the alarm.

7.5 SEQUENCE PROGRAM COPY FUNCTION

The data items of the sequence program stored in EPROM can be copied into the debugging RAM module for PMC–SA1, PMC–SA2, PMC–SB, and PMC–SB2.

The following shows the relationship between the function and soft keys.



7.5.1 Copy Title Data [CPYTTL]

Copies title data.

7.5.2 Copy a Ladder Program [CPYLAD]

Copies a ladder program.

7.5.3 Copy Symbol Data and Comment Data [CPYSYM]

Copies symbol data and comment data.

7.5.4 Copy Message Data [CPYMSG]

Copies message data.

7.5.5 Copy the Sequence Programs [CPYALL]

Copies all the sequence programs into the debugging RAM.

7.5.6 Copy I/O Module Data [CPYMDL]

Copies I/O module data.

> **NOTE**
> If the I/O module data is different from the currently selected data during copying in Subsection 7.5.5 or 7.5.6, turn off the power and restart the system.

7.6 RESTRICTIONS

Two channels cannot be used for the reader/punch interface at the same time. Before performing these I/O operations, be sure to terminate the system other than the PMC and processing through the reader/punch interface in the PMC program.

8 FUNCTIONS FOR DISPLAYING MEMORY AREAS AND DEBUGGING THE PROGRAM (MONIT)

Press the [MONIT] soft key on the basic programmer menu to display the basic monitor menu shown in Fig. 8. Pressing an appropriate soft key enables the user to display memory areas used for a user program written in the C language or to debug a program.

○ : Can be used
× : Cannot be used
Δ : Can be used (with some restrictious)

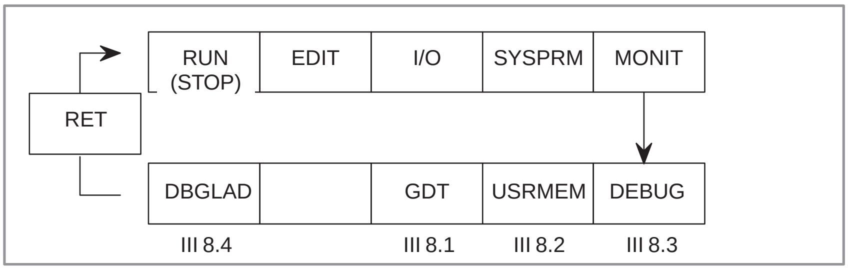
PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2
×	×	×	×	×	×	×	×	×	×	Δ	Δ	○	○	○	○	○

Work RAM is necessary (A02B–0120–H987 for the PMC–SC and SC3 and A02B–0162–J151 or A02B–0162–J152 for the PMC–NB).

These functions facilitate debugging a user program created by the machine tool builder in the C language. If the settings are erroneous, a system error may occur or the system may be shut down. Specify the settings correctly.

For details of operation, refer to the “PMC–SC/SC3/NB Programming Manual (C language)” (B–61863E–1).

The following figure shows the soft key related to these functions.




```
PMC MONITOR MENU                         MONIT STOP

 SELECT ONE OF FOLLOWING SOFT KEYS

  DBGLAD    : DEBUG LADDER DIAGRAM
  GDT       : DESCRIPTOR TABLE MAP
  USRMEM    : USER MEMORY INFORMATION
  DEBUG     : DEBUG FUNCTION

 [DBGLAD] [ ONLINE ] [ GDT  ] [USRMEM] [DEBUG ]
```

Fig. 8 Basic monitor menu

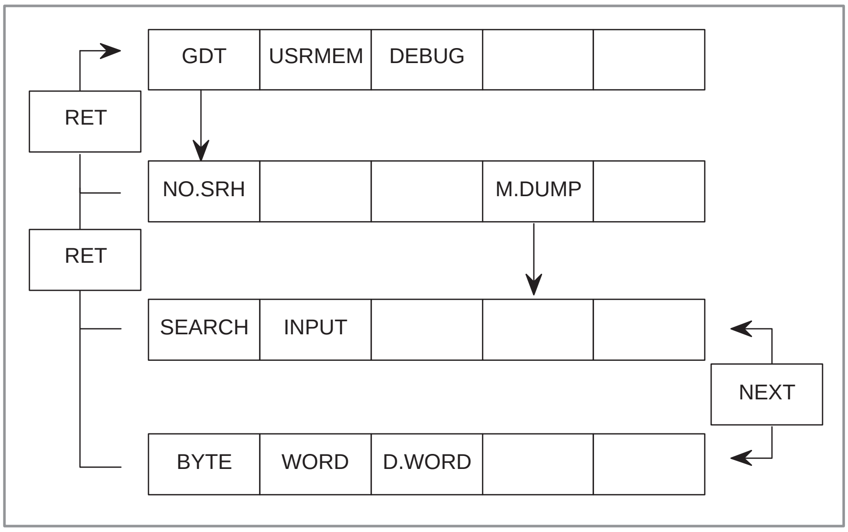
8.1 DISPLAYING THE GDT (GLOBAL DESCRIPTOR TABLE)

Information of a User Program Coded in C

GDT Nos. 32 to 95 defined in a user program can be displayed.

The specified GDTs can also be dumped.

The following figure shows the soft keys related to this function.



8.1.1 Operation

(1) Press the [GDT] soft key to display the user GDT information shown in Fig. 8.1.1 (a).

(2) Use the [NO. SRH] key to search for the GDT table with a desired number.

(3) Press the [M. DUMP] key to dump the data of the GDT number which is displayed at the top.

(4) Press the [NEXT] key on the memory dump screen.

Pressing the [BYTE] key displays the data in units of bytes.

Pressing the [WORD] key displays the data in units of words, where one word equals two bytes. Pressing the [D. WORD] key displays the data in units of double words, or four bytes.

See Fig. 8.1.1 (b).

(5) When bit 4 of keep relay K17 is set to 1, the contents of RAM can be changed in units of the specified length on the memory dump screen by moving the cursor to the data to be changed.

CAUTION

Depending on the settings, a user program may operate erroneously, causing a system error. Be sure to specify the correct settings.

```
PMC DESCRIPTOR TABLE(GDT)     MONIT RUN

 NO. ACCESS USE    BASE      LIMIT
 032   RW   16   0016000AH  0000056FH
 033   RW   16   0016005AH  0000023FH
 034   RW   16   00160300H  00000040H
 035   RW   16   00160340H  00000234H
 036   ER   16   00823000H  00000058H
 037   ER   16   0084FB7CH  0000070AH
 038   NULL DESCRIPTOR
 039   ER   16   0084FF88H  0000292FH
 040   RW   16   00160A6CH  0000005AH
 041   RW   16   00160600H  00000402H
>

[NO.SRH] [      ] [      ] [M.DUMP] [      ]
```

Fig. 8.1.1(a) User GDT information

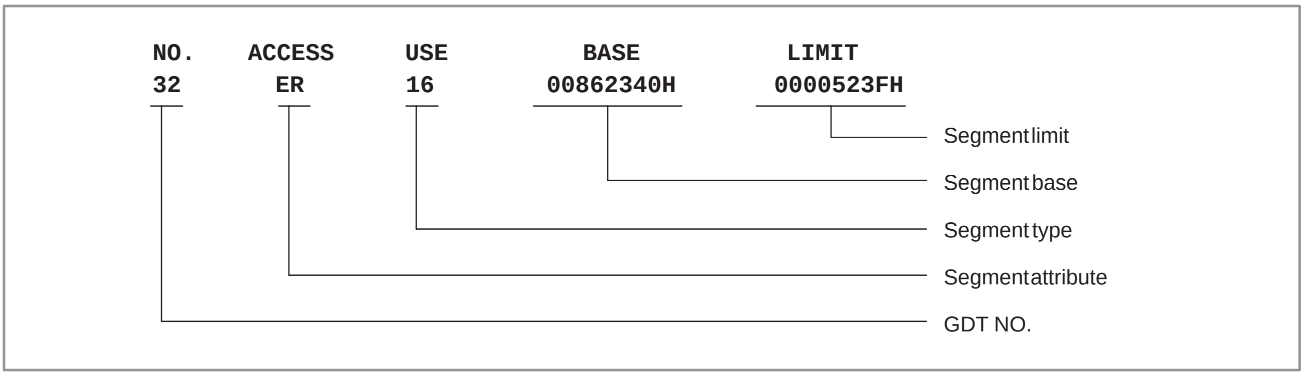
```
PMC DESCRIPTOR TABLE(GDT)     MONIT RUN

 NO. ACCESS USE    BASE      LIMIT
 032   RW   16   0016000AH  0000056FH
 033   RW   16   0016005AH  0000023FH


[NO.SRH] [      ] [      ] [M.DUMP] [      ]
```

```
PMC MEMORY DUMP(GDT NO.032)    MONIT RUN

 103:0000 0000 0000 0000 0000 0000 ......
 103:0010 0000 0000 0000 0000 0000 ......
 103:0020 0000 0000 0000 0000 0000 .....
```

Dumped information of GDT NO.32

Fig. 8.1.1(b) Memory dump

8.1.2 Descriptions of Displayed Items



(1) Access attribute of a segment

Code	Description
RO	Read–only data segment
RW	Read/write data segment
ROD	Read–only downward–expansion data segment
RWD	Read/write downward–expansion data segment
EO	Execute–only code segment
ER	Execute/read code segment

(2) Segment type

Code	Description
16	16–bit segment
32	32–bit segment

NOTE
A user program created with the IC286 compiler is segmented in 16–bit units.

(3) Undefined segment

NULL DESCRIPTOR is displayed for an undefined segment.

8.2 DISPLAYING THE MEMORY ALLOCATION INFORMATION OF A USER PROGRAM CODED IN C.

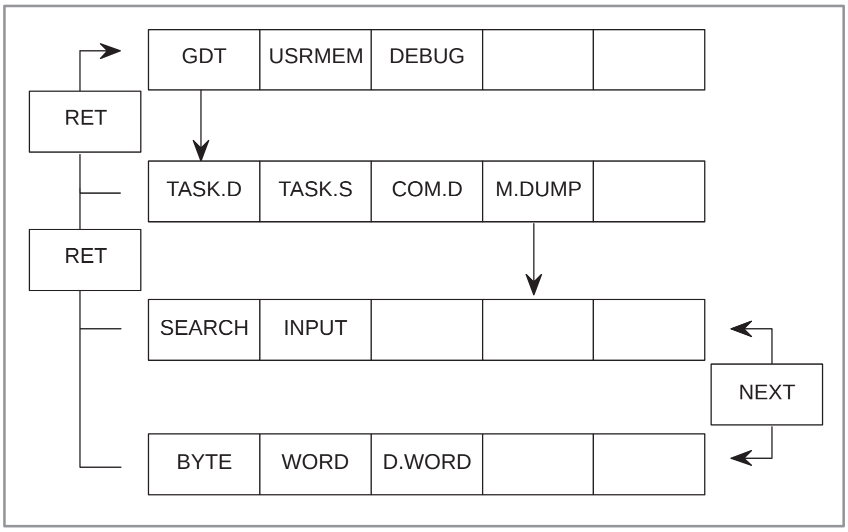
The segment information of the following areas defined by a user program for each task can be displayed and the contents of the areas can be dumped.

- Data area
- Stack area
- Common memory area

The PMC management software dynamically allocates the areas mentioned above at locations which are different from those defined by the user program.

The system allocates the data area at activation. When the system is not activated after the user program has been loaded, the data area is located at the address defined by the user program. Be sure to refer to the area after the system starts.

The following figure shows the soft keys related to this function.



8.2.1 Operation

(1) Press the [USRMEM] soft key. Depending on which soft key is pressed next (see below), the task memory information of a user program is displayed on the screen as shown in Fig. 8.2.1 (a) to (c).

Soft keys

[TASK. D] : Displays the information of allocating task data.
[TASK. S] : Displays the information of allocating task stacks.
[COM.D] : Displays the information of common memory allocation.

(2) A task data area and stack area are displayed for each task ID. The information for all the common memory defined by user link control statements is displayed.

(3) Pressing the [M. DUMP] key on each allocation information screen enables the contents of the memory related to the item which is displayed at the top to be dumped.

(4) Operation on the memory dump screen is the same as that described in Section 8.1.

(5) When bit 4 of keep relay K17 is set to 1, the contents of RAM can be changed in units of the specified length on the memory dump screen by moving the cursor to the data to be changed.

CAUTION

Depending on the settings, a user program may operate erroneously, causing a system error. Be sure to specify the correct settings.

```
PMC USER MEMORY(TASK DATA)     MONIT RUN

 ID NAME      GDT    BASE       LIMIT
 10 TASK–O01  039  00160050H  00010100H
 11 TASK–O02  040  00160060H  00004100H
 12 TASK–O03  041  00160070H  00005100H
 13 TASK–O04  042  00160080H  00000160H
 14 TASK–O05  043  00160210H  00000170H
 15 TASK–O06  044  00160110H  00000110H





>

[TASK.D] [TASK.S] [COM.D ] [M.DUMP] [      ]
```

Fig. 8.2.1(a) Information of a task data area

```
PMC USER MEMORY(TASK STACK)    MONIT RUN

 ID NAME      GDT    BASE       LIMIT
 10 TASK–O01  239  00161050H  00010100H
 11 TASK–O02  240  00161060H  00004100H
 12 TASK–O03  241  00161070H  00005100H
 13 TASK–O04  242  00161080H  00000160H
 14 TASK–O05  243  00161210H  00000170H
 15 TASK–O06  244  00161110H  00000110H





>

[TASK.D] [TASK.S] [COM.D ] [M.DUMP] [      ]
```

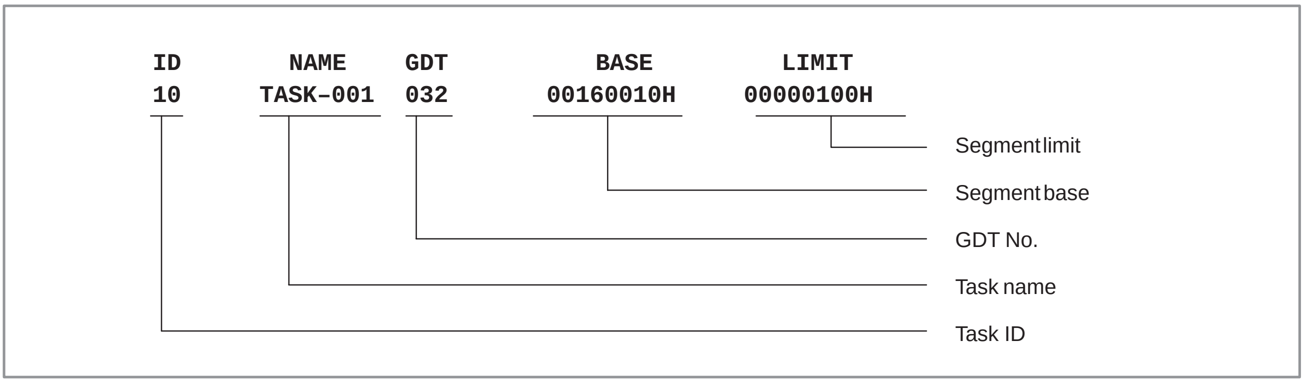
Fig. 8.2.1(b) Information of a task stack area

```
PMC USER MEMORY(COMMON DATA)   MONIT RUN

 NO.          GDT    BASE       LIMIT
 01           042  00162010H  00000100H
 02           045  00162020H  000A0100H
 03           047  00162030H  0000D000H
 04           048  00162040H  0000A100H

>

[TASK.D] [TASK.S] [COM.D ] [M.DUMP] [      ]
```

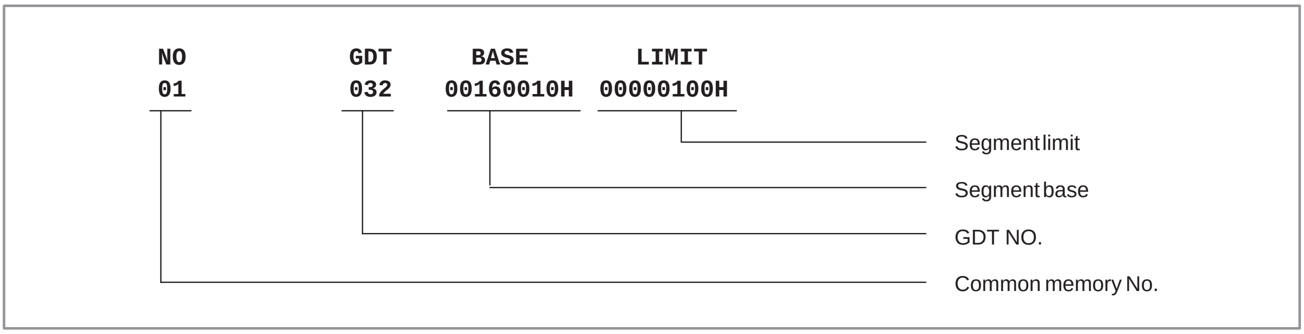
Fig. 8.2.1(c) Information of a common memory data area

8.2.2 Displayed Items

(1) Items displayed for a task data area and stack area



(2) Items displayed for a common memory area



8.3 DEBUGGING

There are two ways to check if a user program operates as intended. One is to execute the program while displaying the sequence on an external unit such as a display monitor. The other is to execute the program to a specified point (breakpoint), and check if the internal data items such as program work areas are correct.

This PMC debugging function checks programs using breakpoints.

8.3.1 Specifications

(1) Number of breakpoints: Up to 4

(2) Number of portions to be traced: 8

(3) Capacity of memory used for storing traced data: Up to 256 bytes, up to 32 bytes for each traced portion

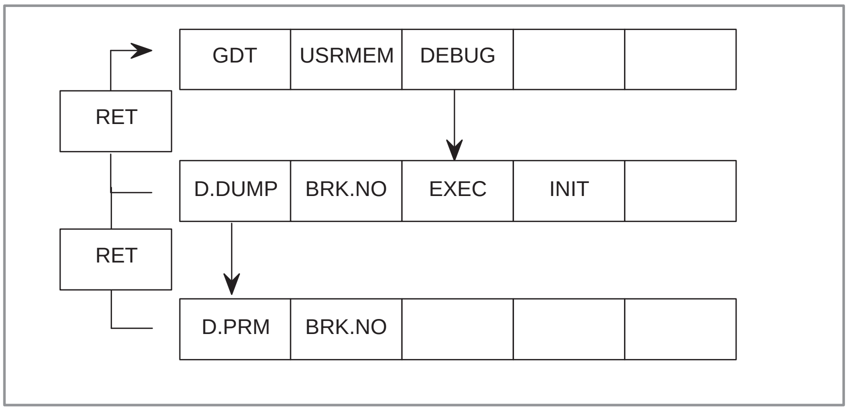
8.3.2 Operation

Press the [DEBUG] soft key to display the parameter screen for debugging. Pressing the [D.DUMP] key on the parameter screen displays the contents of the CPU registers and specified internal data items at the breakpoint.

To return from the data display screen to the parameter screen, press the [D.PRM] soft key.

After the parameters are set, but before the program is interrupted, DBG blinks at the bottom right of the PMC screen. The breakpoint numbers BP1 to BP4 are also displayed at the bottom of the debug function screen. When the program is interrupted at a breakpoint, BRK blinks at the bottom right of the PMC screen. At this time, the breakpoint number, from BP1 to BP4, is displayed in reverse at the bottom of the debug function screen.

The following figure shows soft keys related to this function.



8.3.3 Parameter Screen

When the debug function is used, it is necessary to set the break conditions on the parameter screen. When using a 9" screen, press the <PAGE↓> key to set a trace data area for a breakpoint.

(1) Setting parameters

(a) BREAK SEG.ADR

Specify the effective address of the breakpoint using a segment address. When data is accessed, specify the break address using a segment address.
Use a key, such as EOB, to delimit a segment and an offset. Do not use alphanumeric keys.

NOTE
In data access, an even boundary or 4–byte boundary is assumed according to the type of ACCESS LENGTH, described later.
Example)
When the break address is GDT.NO = 32, OFFSETADDRESS = 101, specify 103; 101, obtained using the following formula:
32 (GDT.NO) x 8 + 3 = 259 = 103 (Hex)

- When ACCESS LENGTH = WORD is specified with BREAK SEG.ADR = 103; 101
  An access to 103; 100–101 causes a break.
- When ACCESS LENGTH = D.WORD is specified with BREAK SEG.ADR = 103; 101
  An access to 103; 100–103 causes a break.

(b) BREAK COND.

Specify a break condition.

0 (EXEC) : A program is interrupted at the specified effective address.

1 (WRITE) : A program is interrupted when it writes data to the specified address.

2 (READ/WRITE) : A program is interrupted when it writes data to or read data from the specified address.

(c) ACCESS LENGTH

Specify the address type of a breakpoint.

0 (BYTE) : An address is specified in units of bytes for read/write operation at the specified address and for when a program is interrupted at the specified effective address.

1 (WORD) : An address is specified in units of words for read/write operation at the specified address.

2 (D.WORD) : An address is specified in units of two words for read/write operation at the specified address.

(d) PASS COUNT

Specify the number of times a break condition is satisfied before the program is interrupted, in the range of 1 to 65535.

(e) TASK ID

Specify the task ID of a program. This parameter is convenient for identifying the program when it is to be interrupted at a breakpoint located in a function called from multiple tasks or is located in common memory.

(f) TASK STATUS

Specify how to handle the task when a program is interrupted.

0 (PASS) : The task continues after the program is interrupted.

1 (STOP) : The user task stops when the program is interrupted. The ladder program does not stop.

NOTE

To restart the user program, press the [STOP] key to stop the sequence program and then press the [RUN] key to start the program on the basic menu using the RUN/STOP function.

(g) BREAK AVAIL.

Specify whether the parameters for each breakpoint are valid or invalid.

(h) NO. TRACE ADR.

Using segment addresses, specify up to eight addresses from which data is traced when a program is interrupted at a breakpoint. Up to 32 bytes are stored for each address.

Use a key, such as EOB, to delimit a segment and an offset. Do not use alphanumeric keys.

To initialize these addresses only, enter 0; 0.

NOTE

If the addresses are specified erroneously, the following two items, TYPE and LENGTH, cannot be specified.

(i) TYPE

Specify an address type with which traced data is displayed.

0 (BYTE) : Data is displayed in units of bytes.

1 (WORD) : Data is displayed in units of words.

2 (D.WORD) : Data is displayed in units of double words.

(j) LENGTH

Specify the length of traced data to be displayed.

(2) Starting processing for a breakpoint

When the parameters for each breakpoint are correctly specified, press the [EXEC] soft key on the parameter screen to start the processing for the currently selected breakpoint. The breakpoint number, from BP1 to BP4, is displayed at the bottom of the screen.

(3) Initializing data used for debugging

To initialize the parameters and memory used for storing traced data, press the [INIT] soft key on the parameter screen. The parameter and memory for the currently selected breakpoint are then initialized.

(4) Changing a breakpoint

Up to four breakpoints can be specified. For each breakpoint, parameters are specified and traced data is stored. To select a desired breakpoint, press the [BRK.NO] soft key on the parameter screen. The breakpoint is selected in the order of BP1, BP2, BP3, and BP4.

```
PMC DEBUG (PARAM)             MONIT RUN

   BREAK POINT NO.1

 BREAK SEG.ADR = 0000:00000000
 BREAK COND.   =     0 ( 0:E 1:W 2:RW )
 ACCESS LENGTH =     0 ( 0:B 1:W 3:D  )
 PASS COUNT    = 32767
 TASK ID       =     1 ( 0:ALL / 10–25 )
 TASK STATUS   =     0 ( 0:PASS 1:STOP )
 BREAK AVAIL.  =     0 ( 0:NO   1:YES  )


>

[D.DUMP] [BRK.NO] [ EXEC ] [ INIT ] [      ]
```

Fig. 8.3.3 (a) Screen for specifying a break condition

```
PMC DEBUG (PARAM)             MONIT RUN

 BREAK POINT NO.1
 NO. DUMP ADR.        TYPE       LENGTH
                (0:B/1:W/2:D) (32BYTE)
 01  0000:00000000      0        10
 02  0000:00000000      1         9
 03  0000:00000000      2         8
 04  0000:00000000      0         7
 05  0000:00000000      1         6
 06  0000:00000000      2         5
 07  0000:00000000      0         4
 08  0000:00000000      1         3
>

[D.DUMP] [BRK.NO] [ EXEC ] [ INIT ] [      ]
```

Fig. 8.3.3 (b) Screen for specifying data to be traced

8.3.4 Screen for Displaying Traced Data

When a program is interrupted under the break condition specified on the parameter screen, BRK blinks at the bottom right of the PMC screen. The breakpoint number at which the program has been interrupted is displayed in reverse at the bottom of the debug function screen.

To display the traced data, press the [D.DUMP] soft key on the parameter screen, then press the [BRK.NO] key to select the screen for displaying the traced data corresponding to the breakpoint.

The following items are displayed.

(1) REGISTER

Displays the contents of the CPU registers.

(2) MEMORY

Displays the contents of memory at addresses of the traced data specified on the parameter screen.

When the contents are displayed on multiple pages, scroll the screen, if necessary, using the <PAGE↑>, <PAGE↓>, <↑>, or, <↓> key.

```
PMC DEBUG (DUMP)              MONIT RUN
 BREAK POINT NO.1(0000:00000000)
 REGISTER
  EAX=00000000 EBX=00000000 ECX=00000000
  EDX=00000000 ESI=00000000 EDI=00000000
  EBP=00000000 ESP=00000000 IEP=00000000
  DS=0000   ES=0000   FS=0000   GS=0000
  SS=0000   CS=0000   EFLAGS=00000000
 CONTENS OF MEMORY
01 0000:00000000 00000000   00000000
02 0000:00000000 00000000   00000000
03 0000:00000000 00 00 00 00 00 00 00 00
04 0000:00000000 0000  0000  0000  0000
>

[D.PRM ] [BRK.NO] [      ] [      ] [       ]
```

Fig. 8.3.4 Screen for displaying traced data

8.3.5 Enabling Automatic Debugging at Power–on

As parameters used for debugging and traces data are stored in the retained memory, they are not lost when the power is turned off.

When bit 1 of keep relay K18 is set to 1 after break condition parameters are correctly specified, debugging is automatically enabled when the power is turned on.

8.3.6 Notes

(1) Specify a break address (BREAK SEG.ADR) in the area used by the user program.

If a break address is specified in the area which is used by the PMC management software, the system may hang up.

(2) Debug function is incorporated in the CPU, reduces the CPU speed. Do not use the function during normal system operation.

8.4 LADDER DEBUGGING FUNCTION

○ : Can be used
× : Cannot be used
Δ : To use this function, a ladder editing module is required

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2
×	Δ	×	×	Δ	○	×	×	Δ	○	○	○	×	○	○	○	○

NOTE
PMC–PA3 is usable with the Power Mate–H.

Using this function, Step Operations and Stop Operations listed below are possible. Step Operations are to execute ladder by specified step (single instruction, single net, and specified block). Stop Operations are to stop the execution of ladder when specified condition becomes true.

(1) Step Operation to execute one instruction from current position.

(2) Step Operation to execute one net (one circuit) from current position.

(3) Step Operation to execute from current position to specified contact or coil instruction.

(4) Stop Operation to execute from the first step and stop the execution at specified contact or coil instruction.

(5) Stop Operation to stop the execution of ladder by a trigger of signal condition. (Optionally, a trigger counter can be specified.)

(6) Stop Operation to stop the execution of ladder after executing one scan. (Optionally, a scan counter can be specified.)

8.4.1 Screen of Ladder Debugging Function

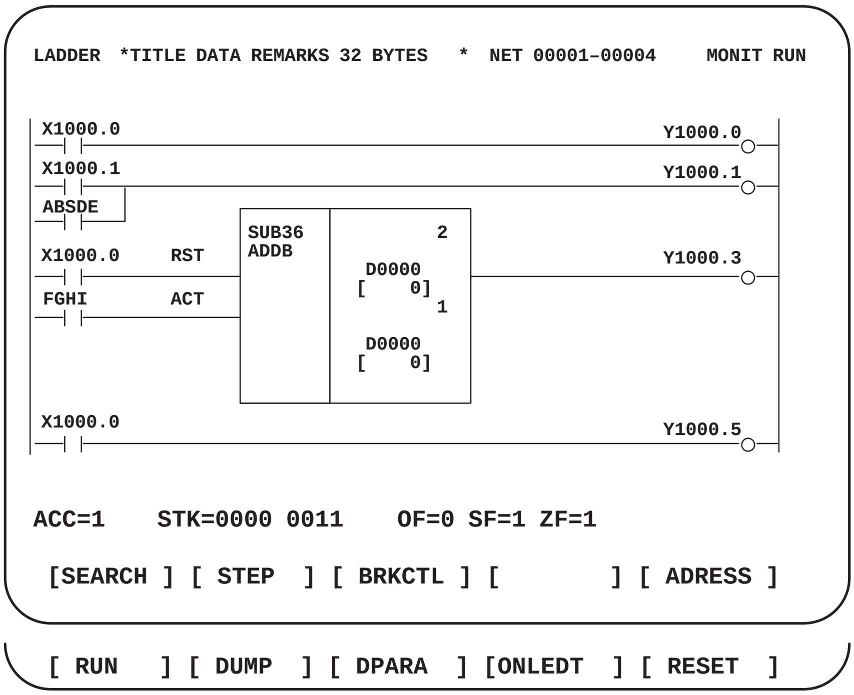
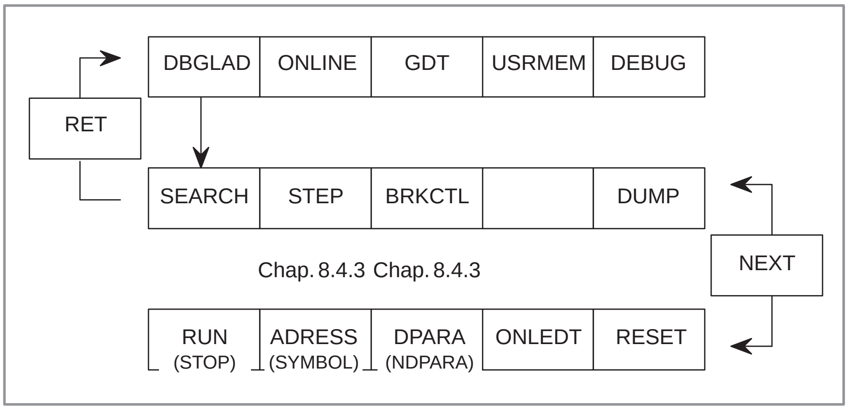


Fig. 8.4.1 Screen of ladder debugging function

8.4.2 Soft Key Menu of Ladder Debugging Function

For this operation, press [DBGLAD] soft key to bring the following menu.



The function of the soft key is as follows.

(1) [SEARCH] : is used to specify several types of search functions.

(2) [STEP] : is used to specify several types of Step Functions. This function can not be used when the ladder program is being executed.

(3) [BRKCTL] : is used to specify Stop Functions.
This function is to stop the execution of ladder when specified condition becomes true. This function can not be used when the ladder program is being executed.

(4) [DUMP] : is used to display the contents of PMC address in the 2 lines at the bottom of CRT where the last NET is normally displayed.

(5) [RUN] : is used to switch the monitor mode from STOP to RUN, or vice versa.

(6) [ADRESS] : is used to switch the symbol display mode from SYMBOL to ADDRESS, or vice versa.

(7) [DPARA] : is used to switch the mode for displaying the contents of functional instruction parameters from NDPARA (No Display Parameter) mode to DPARA (Display Parameter) mode, or vice versa.

(8) [ONLEDT] : is used to edit the ladder program without stopping the execution. Editing is limited within the operations which do not change the size of ladder.

(9) [RESET] : is used to initialize the Step Function and Stop Function.

> **NOTE**
> See Chapter II. 5.3 and 5.4 for details of (7) or (8).

8.4.3 Step Operation [STEP]

Using this function, Step Operations such as single step, single net, and block steps until specified instruction are possible.

[Function]

(1) Step Operation to execute one instruction from current position.

(2) Step Operation to execute one net (one circuit) from current position.

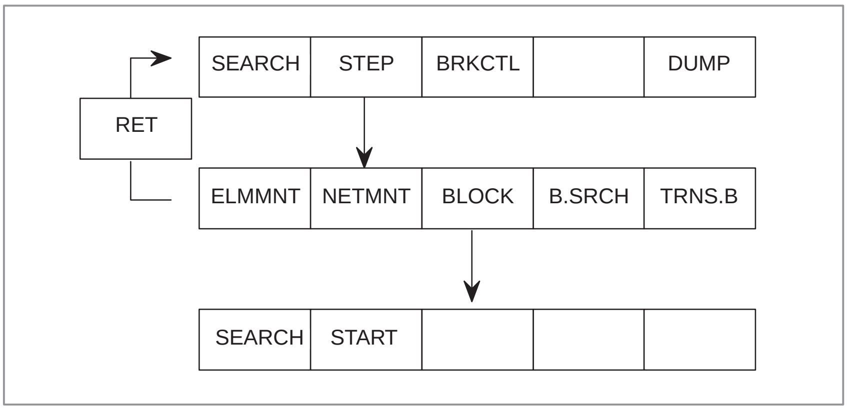
(3) Step operation to execute from current position to specified contact or coil instruction.

[Displaying of Step] See “Fig. 8.4.1”

“ACC=1 STK=0000 0011 OF=0 SF=0 ZF=1”

ACC	:	result of operation	
STK	:	contents of stack	(1 byte)
OF	:	overflow	(0=NO, 1=YES)
SF	:	sign	(0=NO, 1=YES)
ZF	:	zero	(0=NO, 1=YES)

For this operation, press [STEP] soft key to bring the following menu.



The function of the soft key is as follows.

(1) [STEP] : A blinking cursor shows the current position at which the execution is stopped. y moving the cursor, a position at which the execution is to be stopped can be specified.

(2) [ELMMNT] : is used to execute one instruction from current position.

(3) [NETMNT] : is used to execute one net from current position.

(4) [BLOCK] : is used to execute from current position to specified instruction. If specified instruction is not executed because it is skipped by conditional JMP or CALL instructions, the execution will stop at the END instruction, END1 (SUB 1), END2 (SUB 2), or END3 (SUB 48), of the current level.

(5) [B.SRCH] : is used to search the instruction at which the execution is currently stopped.

(6) [TRNS.B] : is used to transfer the current status of input signals to the synchronous buffer so that succeeding instructions could operate on refreshed inputs when the execution is continued from current position. (For more about the synchronous buffer, see Chapter I.2.5 Processing I/O Signal)

NOTE
Normally, transferring to the synchronous buffer is automatically performed at the beginning of the 2nd level ladder.

8.4.4 Stop Function of Break with Condition [BRKCTL]

Using this function, the execution of the ladder can be stopped when specified condition becomes true. Then, the signal condition can be checked.

[Function]

(1) Stop operation to execute from the first step and then to stop at specified contact or coil instruction. (Optionally, a trigger counter can be specified to stop after the instruction is executed specified times.)

(2) Stop operation to stop the execution of ladder when a trigger condition specified by signal becomes true. (Optionally, a trigger counter can be specified to stop after the trigger becomes true specified times.)

(3) Stop operation to stop the execution of ladder after executing a scan. (Optionally, a scan counter can be specified to stop after executing specified times of scans.) The execution is started by pressing [START] key.

[Displaying of specified trigger]

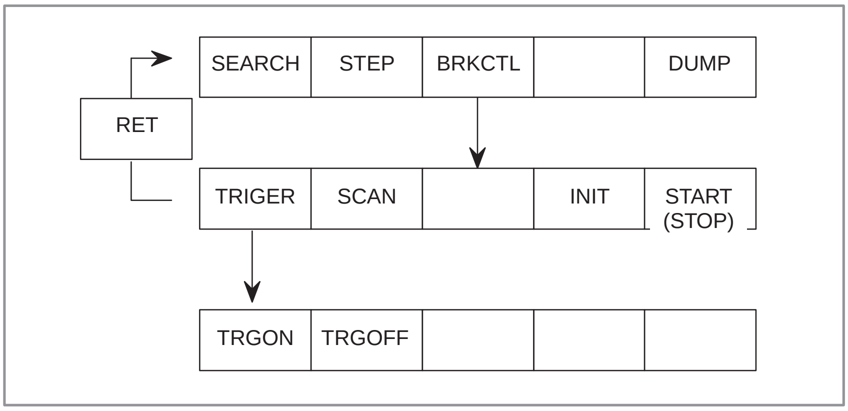
“MODE :ON :X0000. 0 : 0 : 0000/12345”

COUNT : trigger counter (present counter/specified counter)
POINT : trigger point at which the trigger condition is checked (default is 0)
0 the top of the 1st level seq
1 after END1
2 after END2
3 after END3
ADR : specified trigger address
ON : signal turn on (TRGON), OFF : turn off

[Displaying of specified scan]

“SCAN COUNT : 00000/00003”

counter of scan (present counter/specified counter)



The function of [BRKCTL] soft key is as follows.

(1) [TRIGER] : is used to specify the trigger condition by signal. Trigger condition has to be specified according to the following syntax. And then, the execution is started by pressing [START] key.

“ADR ; PONIT (0–3) ; COUNT + [TRGON/ TRGOFF]”

ADR : specified trigger address
POINT : trigger point at which the trigger condition is checked (default is 0)
0 —— the top of the 1st level sequence
1 —— after END1
2 —— after END2
3 —— after END3
COUNT : counter of checked trigger (default is 1) (1 to 65535)

(2) [TRGON] : is used to specify “turn on” trigger to stop the execution when the signal is transitioned from off to on status.

(3) [TRGOFF] : is used to specify “turn off” trigger to stop the execution when the signal is transitioned from on to off status.

(4) [DUMP] : is used to display the contents of PMC address in the 2 lines at the bottom of CRT where the last NET is normally displayed.

(5) [SCAN] : is used to specify a stop function by scan counter. To specify a scan counter, input as follows.
“counter + [SCAN]”. (counter: 1 to 65535)
When the counter is not specified, it is recognized as 1.
After specifying the scan counter, the execution is started by pressing [START] key.

(6) [INIT] : is used to initialize the stop function with break condition.

(7) [START] : is used to start the execution after specifying the condition to stop.

8.5 ONLINE FUNCTION

○ : Usable
Δ : See Note
× : Not usable

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2
×	Δ	Δ	×	Δ	○	×	×	○	○	○	○	×	○	○	○	○

NOTE
PMC–PA3 is usable with the Power Mate–D/H.
PMC–SA1 is usable with the loader control function of the Series 21*i*.
PMC–SA3 is usable with the FANUC NC Board.

With the online function of the FAPT LADDER–II or ladder editing package, the following can be performed using the personal computer:

- Ladder monitor display
- Online ladder editing
- PMC parameter display and editing
- Signal state monitor display and modifications
- Input/output to and from the PMC (loading from the PMC, storing to the PMC)
- Writing to flash ROM

Before this online function can be used, communication conditions must be set in the PMC built into the CNC.

8.5.1 Online Setting Screen

```
PARAMETERS FOR ONLINE MONITOR      MONIT STOP

 CPU ID     =
 RS–232C    = USE/ NOT USE
  CHANNEL   =   1
  BAUD RATE = 300/600/1200/2400/4800/ 9600 19200
  PARITY    = NONE/ODD/EVEN
  STOP BIT  = 1 BIT/2 BITS
  TIMER 1   =     0
  TIMER 2   =  5000
  TIMER 3   = 15000
  MAX PACKET SIZE =  1024
 F–BUS      = USE/ NOT USE
 RS–232C = INACTIVE     :    0
 F–BUS   = INACTIVE     :    0
>

                EMG ST  INIT
```

Fig. 8.5.1 Online monitor setting screen

EMG ST : Terminates communication forcibly. Use this key when communication becomes abnormal, such that the connection cannot be terminated normally.

INIT : Initializes the parameters to their default values.

NOTE
When the configuration disables the use of the ladder editing package, the F–BUS item is not displayed.
When the 9–inch CRT is being used, two pages are used for screen display. Switch between the pages with the <Page Up> and <Page Down> keys.

8.5.2 Setting Method

1. For communication with FAPT LADDER–II
   (1) Check that NOT USE is set for the RS–232C item.
   (2) Set the CHANNEL and BAUD RATE parameters.
   (3) Move the cursor to the RS–232C item with the and keys.
   (4) Select USE with the "←" or "→" key.
2. For communication with the ladder editing package
   (1) Move the cursor to the F–BUS item with the and keys.
   (2) Select USE with the "←" or "→" key.

This completes the setting for communication.
The online function is operated from the FAPT LADDER–II or ladder editing package.

For details of operation, refer to the operator's manual for the FAPT LADDER–II or ladder editing package.

8.5.3 Setting on the NC Parameter Screen

When the NC (such as the Series 160*i* or 180*i*) does not support the display of the PMC screen, or if communication is to be started automatically at power–on without screen manipulation, select a communication destination with NC parameter No. 24.

When the value of NC parameter No. 24 is changed, the PMC online monitor screen is set as indicated below.

NC parameter No. 24	Setting on the PMC online monitor screen		
	RS–232C	F–BUS	Description
0	NOT USE	USE	Selects the ladder editing package.
1	USE (Channel 1)	NOT USE	Selects FAPT LADDER–II (channel 1).
2	USE (Channel 2)	NOT USE	Selects FAPT LADDER–II (channel 2).
255	NOT USE	NOT USE	Terminates communication forcibly (EMG ST).

NOTE

With NC parameter No. 24, the PMC online monitor screen setting is changed when the power is turned on.
After this parameter has been modified, the power must be turned off then back on for the new setting to become effective.
When the setting of the online monitor screen is to remain unchanged, a value other than 0, 1, 2, and 255 must be set for this parameter.

8.5.4 Explanation of Communication Status

On the online monitor screen, a message is displayed to indicate the RS–232C or F–BUS communication status. The table below lists the displayed messages and their explanations.

Displayed message	Explanation
INACTIVE	The unit is inactive.
STOPPING	The unit is being stopped. (Wait for the termination of communication)
STARTING	The unit is being started. (Wait for the termination of communication over another communication path)
STAND–BY	The unit is active and in standby mode.
CONNECTED	The unit is active and being connected.
NO OPTION	The file cannot be opened because the corresponding option is not provided.
BAD PARAMETER	An invalid open parameter is specified.
TIMEOUT ERROR	A time–out has occurred and communication is aborted.
BCC ERROR	A BCC (packet parity) error has occurred.
PARITY ERROR	A parity error has occurred.
OVER–RUN ERROR	A reception overrun has occurred, from which the unit cannot recover.
SEQUENCE ERROR	Packets are out of sequence. (Incorrect procedure)
DATA ERROR	Correct packets cannot be received through retransmission.
QUEUE OVERFLOW	The transmit/receive queue has overflowed.
DISCONNECTED	Communication has been terminated successfully.
NO CONNECTION	The cable is disconnected.

9 ERROR MESSAGES (FOR EDIT)

Error messages (For EDIT 1)

Message	Contents and solution
ADDRESS BIT NOTHING	The address of the relay/coil is not set.
FUNCTION NOT FOUND	There is no functional instruction of the input number.
COM FUNCTION MISSING	The functional instruction COM(SUB9) is not correctly dealt with. Correspondence of COM and COME(SUB29) is incorrect. Or, the number of coil controlled by COM is specified by the model which the number cannot be specified.(It is possible to specify the number of coil only on PMC–RB/RC.)
EDIT BUFFER OVER	There is no empty area of the buffer for the editing. (solution) Please reduce NET under editing.
END FUNCTION MISSING	Functional instruction END1, END2, END3 and END do not exist Or, there are error net in END1, END2, END3, END. Or, order of END1, END2, END3, and END is not correct.
ERROR NET FOUND	There is an error net.
ILLEGAL FUNCTION NO.	The wrong number of the functional instruction is searched.
FUNCTION LINE ILLEGAL	The functional instruction is not correctly connected.
HORIZONTAL LINE ILLEGAL	The horizontal line of the net is not connected.
ILLEGAL NET CLEARED	Because the power had been turn off while editing LADDER, some net under editing was cleared.
ILLEGAL OPERATION	Operation is not correct. The value is not specified and only INPUT key was pushed. The address data is not correctly inputted. Because the space to display the instruction on screen is not enough, the functional instruction cannot be made.
SYMBOL UNDEFINED	The symbol which was inputted is not defined.
INPUT INVALID	There is an incorrect input data. Non–numerical value was inputted with COPY,INSLIN,C–UP, C–DOWN etc. The input address was specified for write coil. An illegal character was specified for the data table.
NET TOO LARGE	The input net is larger than the editing buffer. (solution) Please reduce the net under editing.
JUMP FUNCTION MISSING	The functional instruction JMP(SUB10) is not correctly dealt with. Correspondence of JMP and JMPE(SUB30) is incorrect. The number of coil to jump is specified by the model which the number of coil cannot specified. (It is possible to specify the coil number only on PMC–SB/SC.)
LADDER BROKEN	LADDER is broken.
LADDER ILLEGAL	There is an incorrect LADDER.
IMPOSSIBLE WRITE	You try to edit sequence program on the ROM.
OBJECT BUFFER OVER	The sequence program area was filled. (solution) Please reduce the LADDER.
PARAMETER NOTHING	There is no parameter of the functional instruction.
PLEASE COMPLETE NET	The error net was found in LADDER. (solution) After correcting the error net, please continue operating.
PLEASE KEY IN SUB NO.	Please input the number of the functional instruction. (solution) If you do not input the functional instruction, please push soft key ”FUNC” again.
PROGRAM MODULE NOTHING	You tried to edit though there was neither RAM for debugging nor ROM for sequence program.
RELAY COIL FORBIT	There is an unnecessary relay or coil.
RELAY OR COIL NOTHING	The relay or the coil does not suffice.
PLEASE CLEAR ALL	It is impossible to recover the sequence program. (solution) Please clear the all data.

Error messages (For EDIT 2)

Message	Contents and solution
SYMBOL DATA DUPLICATE	The same symbol name is defined in other place.
COMMENT DATA OVERFLOW	The comment data area was filled. (solution) Please reduce the number of the comment.
SYMBOL DATA OVERFLOW	The symbol data area was filled. (solution) Please reduce the number of the symbol.
VERTICAL LINE ILLEGAL	There is an incorrect vertical line of the net.
MESSAGE DATA OVERFLOW	The message data area was filled. (solution) Please reduce the number of the message.
1ST LEVEL EXECUTE TIME OVER	The 1st level of LADDER is too large to complete execution in time. (solution) Please reduce the 1st level of LADDER.
PARA NO.RANGE ERR : functional–instructionname	A parameter number used for a functional instruction falls outside the allowable range. (solution) Change the parameter number to a value within the allowable range.
PARA NO.DUPLICATE : functional–instructionname EXIT ?	A parameter number used for a functional instruction is specified more than once. (solution) Change the number to a number that has not yet been used if duplicate execution causes an error.

Error messages (during automatic F–ROM writing after ladder editing)

Message	Contents and solution
PROGRAM ALREADY EXISTS	A program already exists in flash ROM. (Upon BLANK execution)
PROGRAM ALREADY EXISTS (EXEC?)	A program already exists in flash ROM. (Action) While the message is being displayed, press the EXEC key again to execute WRITE or ERASE. (Upon WRITE or ERASE execution)
PROGRAM NOTHING	No program exists in flash ROM.
ERASE ERROR F–ROM WRITE ERROR 13 F–ROM WRITE ERROR 28	Flash ROM has failed and requires replacement. Contact a FANUC service representative.
WRITE ERROR F–ROM WRITE ERROR 12 F–ROM WRITE ERROR 29	
READ ERROR	
ANOTHER USED F–ROM WRITE ERROR 9 F–ROM WRITE ERROR 36	A unit other than the PMC is using the flash ROM.
MUST BE IN EMG STOP NOT EMG STOP F–ROM WRITE ERROR 10 F–ROM WRITE ERROR 37	The CNC is not in emergency stop mode.
NO OPTION	The ROM cassette option is not provided.
SIZE ERROR IMPOSSIBLE WRITE (SIZE OVER) NO SPACE F–ROM WRITE ERROR 1 F–ROM WRITE ERROR 15 F–ROM WRITE ERROR 35	The size of a sequence program is larger than that of the flash ROM. (Upon WRITE execution) (Action) Try the CONDENSE function (on the EDIT/CLEAR screen). If the problem persists, increase the flash ROM size. The size of a sequence program to be read is larger than that of RAM. (Upon READ execution) (Action) Increase the RAM size.

10 ERROR MESSAGES (FOR I/O)

Error messages (For I/O 2)

Message	Contents and solution
I/O OPEN ERROR nn	An error occurs when the reader/puncher interface was started. nn = –1 Because the interface is used with NC etc., the interface is not able to be opened by PMC side. (solution) After other functions finishes using the line, please execute again. 6 There is no option for the interface. 20 The interface cannot be opened. (solution) Please confirm the connection of the cable. Please confirm setting of the baud rate etc.
I/O WRITE ERROR nn	An output error occurred in the reader/puncher interface. nn = 20 The state of the interface is not correct. (solution) Please confirm the connection of the cable. Please confirm setting the baud rate etc. 22 Opponent side is not ready to receive. (solution) Please confirm the power supply on the opponent side. Or, please initialize the interface.
I/O READ ERROR nn	An input error occurred in the reader/puncher interface. nn = 20 The state of the interface is not correct. (solution) Please confirm the connection of the cable. Please confirm setting the baud rate etc. 21 The data is not sent from the opponent side. (solution) Please confirm the power supply on the opponent side.
I/O LIST ERROR nn	An error occurred in directory read processing from FD Cassette. nn = 20 The state of the interface is not correct. (solution) Please confirm the connection of the cable. Please confirm setting of the baud rate etc.
COMPARE ERR xxxxxx = aa:bb CONT?(Y/N)	A compare error occurred. xxxxxx : The Address where the compare error occurred. aa : The data on PMC side bb : The data on device side Enter 'Y' to continue processing.
ADDRESS IS OUT OF RANGE(xxxxxx)	The data transferred to the address out of the PMC debugging RAM area. xxxxxx : Transferred address. (solution) Please confirm the address of the transferring data. LADDER : Please confirm the model setting. C language : Please confirm setting the address in the link control statement and build file.
ROM WRITER ERROR nnnnnn	An error occurred in the ROM writer.

11 PMC PROGRAMMER (DPL/MDI) (ONLY FOR THE Power Mate–D/F/H)

The DPL/MDI panel is used to set PMC system parameters and create and execute the sequence program.

(1) Setting and displaying PMC system parameters (SYSTEM PARAM)

– The type of counter data (BCD or binary) can be selected.

(2) Editing the sequence program (EDIT)

– The sequence program can be edited (input, addition, search, and deletion) by using the ladder mnemonics display.

(3) Executing the sequence program (RUN/STOP)

– The execution of the sequence program can be started and stopped.

(4) Storing the sequence program into flash EEPROM (I/O)

– The sequence program can be stored into flash EEPROM (only for the Power Mate–H).

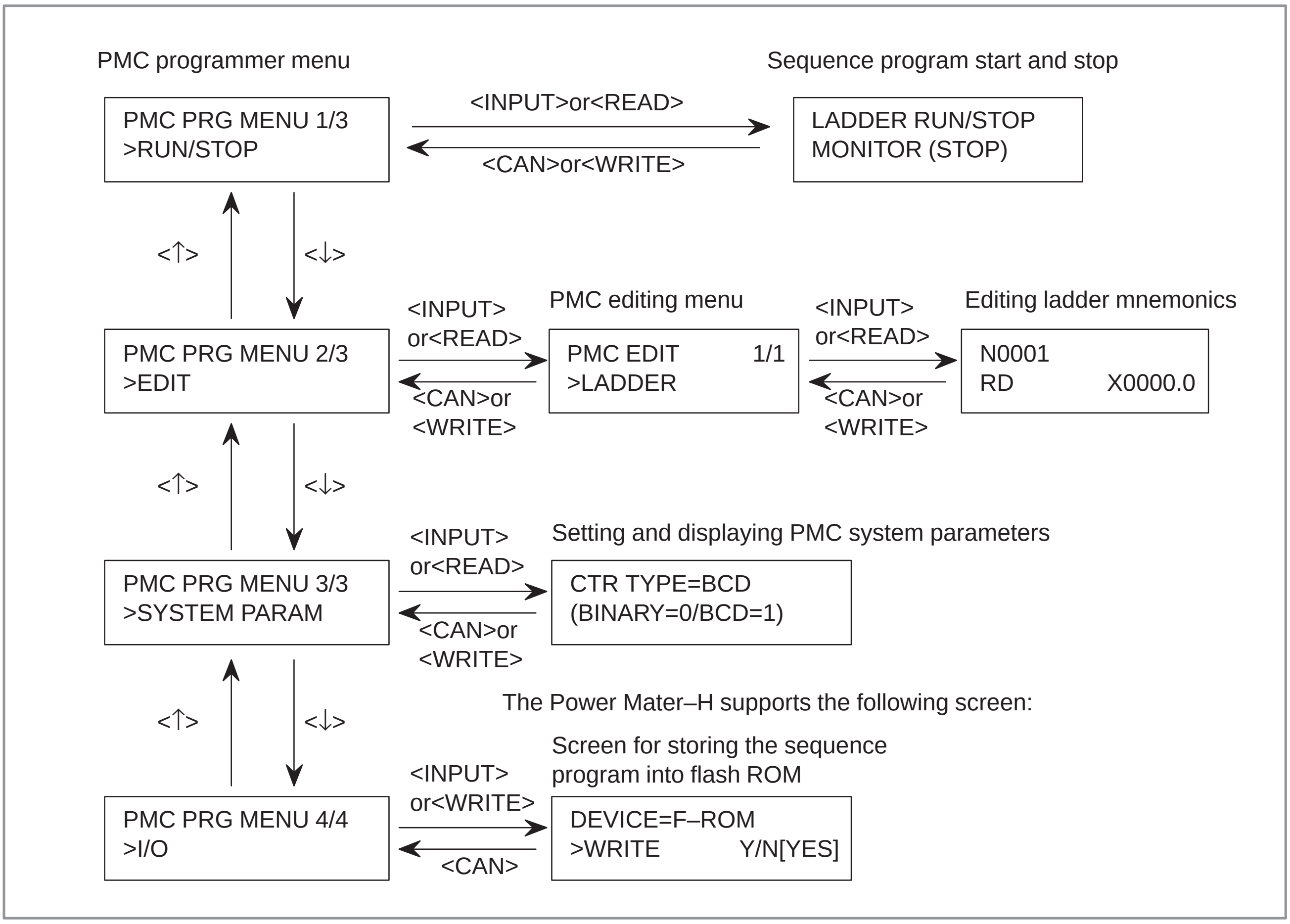
The DPL/MDI panel is shown below.

Fig. 11 DPL/MDI panel for Power Mate

NOTE

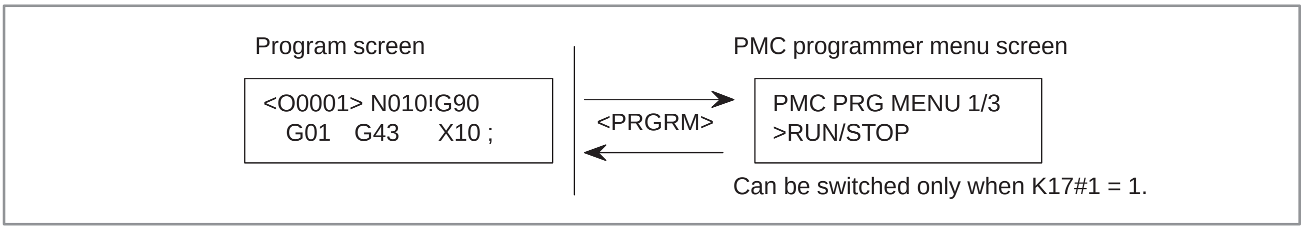
1. With the PMC programmer (DPL/MDI) function, the characters indicated at the lower–left part of each key are used.
2. When the <D/R> key is pressed once, the left–hand character is valid. When the <D/R> key is pressed twice, the right–hand character is valid.
   Example : When the <D/R> key is pressed once, D is keyed in. When the <D/R> key is pressed twice, R is keyed in. When a password is cleared, however, only the characters on the left side are valid.

The screen configuration for the PMC programmer (DPL/MDI) function is as follows:

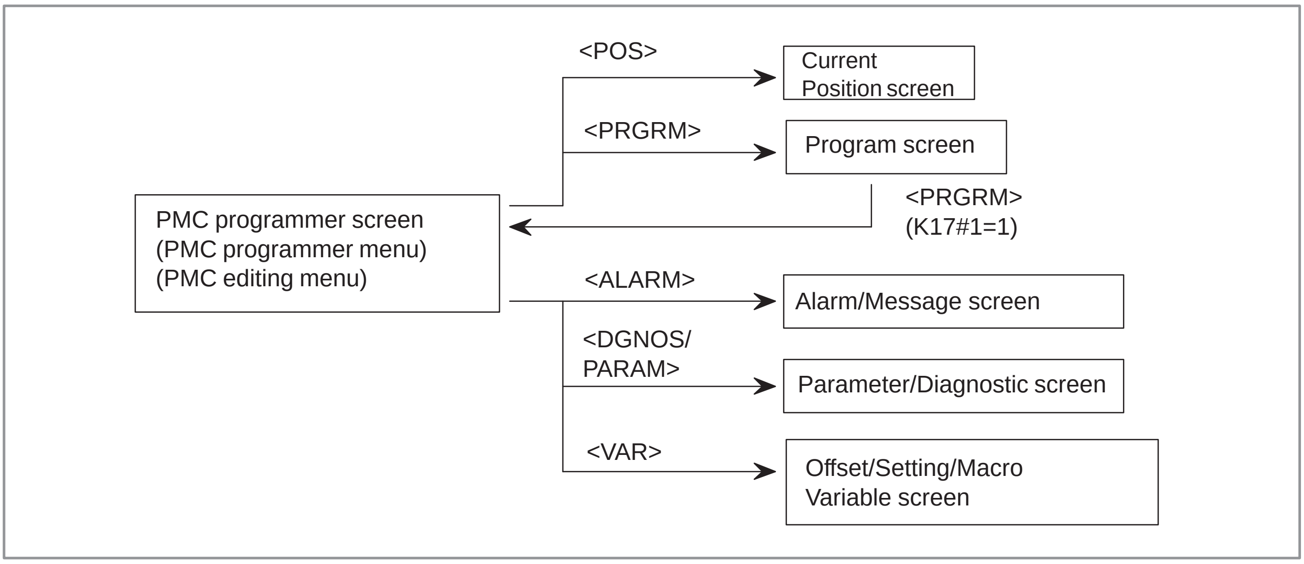


11.1 SELECTING THE PMC PROGRAMMER MENU

To operate the PMC programmer, set K17#1 of the keep relay area for PMC parameters to 1, then press the <PRGRM> key two times on the DPL/MDI (press the <PRGRM> key further when the program screen is selected), thus causing the PMC programmer menu to be displayed.



To return to the CNC screen, press the <POS>, <PRGRM>, <VAR>, <DGNOS/PARAM>, or <ALARM> key.



The following keys on the DPL/MDI panel are used for PMC operation:

1 <POS>, <PRGRM>, <VAR>, <DGNOS/PARAM>, <ALARM> key
Returns to the CNC screen.

2 <↑> key
Shifts the cursor upward.

3 <↓> key
Shifts the cursor downward.

4 <INPUT>, <READ> key
Selects a function when the PMC programmer menu or PMC editing menu is displayed.

5 <CAN>, <WRITE> key
Returns to the previous menu from the PMC programmer menu or PMC editing menu.

11.2 SETTING AND DISPLAYING SYSTEM PARAMETERS (SYSTEM PARAM)

Selecting SYSTEM PARAM on the PMC programmer menu displays the system parameter screen. If the sequence program is running, selecting this function automatically stops the program.

1 Display the PMC programmer menu.

2 Display the SYSTEM PARAM item by pressing the <↓> or <↑> key.

```
PMC  PRG  MENU       3/3
>SYSTEM   PARAM
```

3 Press the <INPUT> or <READ> key. The system parameter screen appears.

```
CTR   TYPE = BIN
(BINARY=0/BCD=1)
```

4 The current counter data type is displayed on the screen.

(a) Specify the type of the counter value to be used for the CTR functional instruction, as binary or BCD (enter <0> for binary or <1> for BCD).

(b) Press the <INPUT> key.

The counter data type is set.

5 Pressing the <CAN> or <WRITE> key displays the PMC programmer menu.

NOTE

If the PMC parameter keep relay K19#0 is set to 1, the screen for writing a sequence program into Flash Memory is displayed upon the completion of editing. (This is applicable to the Power Mate–H only).

```
DEVICE=F–ROM
>WRITE Y/N [YES]
```

Write a sequence program into Flash Memory as explained in Section 11.7.

11.3 EDITING THE SEQUENCE PROGRAM (EDIT)

Selecting EDIT on the PMC programmer menu displays the editing menu.

1 Display the PMC programmer menu.

2 Display the EDIT item by pressing the <↓> or <↑> key.

```
PMC  PRG  MENU        2/3
>EDIT
```

3 Press the <INPUT> or <READ> key. The PMC editing menu appears.

```
PMC  EDIT             1/1
>LADDER
```

To end editing and display the PMC programmer menu, press the <CAN> or <WRITE> key.

11.4 EDITING LADDER MNEMONICS

When ladder mnemonic editing (LADDER) is selected from the PMC edit menu, the ladder mnemonic edit screen is displayed.

When this function is selected, the sequence program stops.

11.4.1 Starting Ladder Mnemonics Editing

When ladder mnemonic editing (LADDER) is selected from the PMC edit menu, the ladder mnemonic edit screen is displayed.
When this function is selected, the sequence program stops.

1 Display the PMC edit menu screen.

2 Display the LADDER item by pressing the <↓> or <↑> key.

```
PMC  EDIT                    1/1
>LADDER
```

3 Press the <INPUT> or <READ> key.
When a password is set for the ladder: Proceed to step 4.
When no password is set for the ladder: Proceed to step 6.

4 If a password is set, a password clear request is displayed.

```
PASSWORD (R/W)
```

NOTE
For a ladder for which a password has been set, the ladder mnemonic editing function cannot be started unless the correct password is entered. Once the password is cleared, the password remains cleared until the power is turned off then back on.

5 Enter the password, then press the <INPUT> key.

NOTE
The entered password is not displayed (not echoed back on the screen).

If the entered password is incorrect, the following error message is displayed.

```
FALSE PASSWORD
```

If the <INPUT> key is pressed at this time, the screen display returns to the password clear request screen.
If the <CAN> key is pressed at this time, the screen display returns to the PMC edit menu.
If the entered password is correct, the password is cleared.

6 A sequence program is displayed.

Step number

```
> N0001
  RD                X0000.0
```

Instruction

11.4.2 Confirming the Ladder Mnemonics

1 Cursor scroll (scroll per step)
Pressing the <↑> cursor key displays the instruction one step before that currently displayed. Pressing the <↓> cursor key displays the instruction one step after that currently displayed.

2 Specifying the step number
Entering <NO.>, <step number>, then <INPUT> displays the instruction having the entered step number.
(The <↓> cursor key can be used instead of the <INPUT> key.)
(Example) <NO.>, <123>, <↓>

```
N0123
SUB   50   PSGNL
```

3 Relay search
Entering <address number> then <↓> searches for the relay including the entered address.
(Example) <X0.2>, <↓>

```
N0105
AND                 X0000.2
```

4 Relay coil search
Entering <WRT>, <address number>, then <↓> searches for the relay coil including the entered address.
(Example) <WRT>, <Y33.5>, <↓>

```
N0187
WRT. NOT            Y0033.5
```

5 Functional instruction search
Entering <SUB>, <functional instruction number>, then <↓> searches for the entered functional instruction.
(Example) <SUB>, <50>, <↓>

```
N0123
SUB   50   PSGNL
```

NOTE

1 Relay search, relay coil search, and functional instruction search are started from the current screen. If the relay, relay coil, or instruction is not found by the end of the ladder program, search is performed from the beginning of the ladder program to the step at which search was started. If still not found, ”NOT FOUND” is displayed.

```
N0105   NOT   FOUND
AND           X0000.2
```

2 Display of some instructions may differ from that for FAPT LADDER.

P–G, personal–computer FAPT LADDER	Ladder mnemonics editing
(a) RD.NOT.STK	RD.N.STK
(b) TMR timer–number	SUB 03 TMR P001 timer–number
(c) DEC code–signal–address (PRM) decode–instruction	SUB 04 DEC P001 code–signal–address P002 decode–instruction

The above also applies when modifying the ladder mnemonics.

11.4.3 Modifying the Ladder Mnemonics

1 Changing an instruction

(a) Display the instruction to be changed.

(b) Enter a new instruction.

(c) Press the <ALTER> key.

(Example) <OR>, <Y32.4>, <ALTER>

```
N1234
AND                  R0123.4
```

Before change

```
N1234
OR                   Y0032.4
```

After change

NOTE
If changing the instruction causes the memory capacity to be exceeded, the <ALTER> key is ignored without changing the instruction.

2 Deleting an instruction

(a) Display the instruction to be deleted.

(b) Press the <DELET> key.
The instruction is deleted and the next instruction is displayed.

3 Inserting an instruction

(a) Display the instruction after which an instruction is to be inserted.

(b) Enter the instruction to be inserted.

(c) Press the <INSRT> key.

(Example) <AND>, <STK>, <INSRT>

```
N1234
AND                  R0123.4
```

Before insertion

```
N1234
AND.STK
```

After insertion

NOTE
If inserting the instruction causes the memory capacity to be exceeded, the <INSRT> key is ignored without inserting the instruction.

4 Deleting the ladder program

(a) Enter <–9999>.

(b) Press the <DELET> key.

The whole ladder program is deleted.

11.4.4 Ending Ladder Mnemonics Editing

1 Press the <CAN> or <WRITE> key.

2 ”EXECUTING” is displayed.

```
N0001
    EXECUTING
```

3 The PMC editing menu appears.

NOTE

1 If the sequence program contains an error, the PMC editing menu is not displayed but an error message appears on the screen.
(Example) Error message

```
END  FUNCTION
MISSING
```

Pressing the <↑> or <↓> cursor key displays the ladder mnemonics editing screen.

2 Pressing the <POS>, <PRGRM>, <VAR>, <DGNOS/PARAM>, or <ALARM> key during the editing of the sequence program displays the CNC screen by forcibly terminating editing even if the program contains an error.

3 If the PMC parameter keep relay K19#0 is set to 1, the screen for writing a sequence program to flash ROM is displayed upon the completion of editing. (This is applicable to the Power Mate–H only).

```
DEVICE=F–ROM
>WRITE   Y/N [YES]
```

Write a sequence program into flash ROM as explained in Section 11.7.

11.5 STARTING AND STOPPING THE SEQUENCE PROGRAM (RUN/STOP)

Selecting RUN/STOP on the PMC programmer menu displays the sequence program start/stop screen.

1 Display the PMC programmer menu.

2 Display the RUN/STOP item by pressing the <↓> or <↑> key.

```
PMC PRG MENU          1/3
>RUN/STOP
```

3 Press the <INPUT> or <READ> key. The sequence program start/stop screen appears.

```
LADDER RUN/STOP
  MONITOR  [RUN]
```

4 The current execution state of the sequence program is displayed on the screen.
Pressing the <↓> or <↑> key switches the state between running and stopped.

5 Pressing the <CAN> or <WRITE> key displays the PMC programmer menu.

> **CAUTION**
> When the sequence program cannot be started(RUN), the alarm of PMC occurred. Please confirm the alarm status referring to “11.11 Error List”.

11.6 ERROR MESSAGES (FOR LADDER MNEMONICS EDITING)

	Displayed error message	Error description (operator action)
1	COIL NOTHING	No coil is specified for a functional instruction using a coil.
2	COM FUNCTION MISSING	The use of the COM (SUB9) functional instruction is incorrect.
3	END FUNCTION MISSING	The END1 or END2 functional instruction is missing (or ERROR NET).
4	JUMP FUNCTION MISSING	The use of the JMP (SUB10) functional instruction is incorrect.
5	LADDER BROKEN	The ladder program is corrupted.
6	OBJECT BUFFER OVER	The user program RAM is full. (Note) (Perform condensation or reduce the size of the ladder program.)
7	PLEASE CLEAR ALL	The sequence program has become unrecoverable due to power–off during editing.
8	1ST LEVEL EXEC TIME OVER	The ladder first level is too great.

CAUTION

Use a memory card for ladder diagram editing or the CONDENSE function of FAPT LADDER (for personal computers). These methods may, however, not be effective.

11.7 STORING THE SEQUENCE PROGRAM INTO FLASH EEPROM (I/O) (ONLY FOR THE Power Mate–H)

Selecting I/O on the PMC programmer menu displays the screen for storing the sequence program into flash EEPROM. Before attempting to store the sequence program into flash EEPROM, place the CNC in the emergency stop state.

(1) Display the PMC programmer menu.

(2) Display the I/O item by pressing the <↓> or <↑> key.

```
PMC  PRG  MENU        4/4
>I/O
```

(3) Press the <INPUT> or <WRITE> key. The sequence program storage screen appears. Pressing the <↓> or <↑> key switches display between [YES] and [NO].

```
DEVICE=F–ROM            <↓>          DEVICE=F–ROM
>WRITE!Y/N[YES]    ------------->    >WRITE!Y/N[NO  ]
                   <-------------
                        <↑>
```

(4) When [NO] is displayed, pressing the <INPUT> key displays the sequence program storage screen. When [YES] is displayed, pressing the <INPUT> key starts writing the sequence program into flash EEPROM.
”EXECUTING” is displayed during writing.

```
WRITE  TO  F–ROM
     EXECUTING
```

”EXECUTING” BLINKS.

Once the sequence program has been written normally, ”COMPLETE” is displayed.

```
WRITE  TO  F–ROM
     COMPLETE
```

> **NOTE**
> If an error occurs, an error message appears on the screen.

Example

Example error message

```
NOT  EMG  STOP
```

To return to the sequence program storage screen, press the <↑> or <↓> key.

(5) Pressing the <CAN> key displays the PMC programmer menu.

11.8 ERROR DETAILS

The table below lists the details of the errors which may occur during storage into F–ROM using the DPL/MDI (only for the Power Mate–H).

Error message	Description
PROGRAM DATA ERROR	The ladder data in RAM is invalid. Alternatively, there is no RAM or ROM.
SIZE ERROR	The program exceeds the maximum size which can be written into F–ROM.
NOT EMG STOP	The CNC is not in the emergency stop state.
OPEN ERROR	The OPEN processing has failed.
ERASE ERROR	The ERASE processing has failed. The F–ROM cannot be erased. Alternatively, the F–ROM is defective.
WRITE ERROR	The WRITE processing has failed. The F–ROM cannot be written. Alternatively, the F–ROM is defective.

11.9 INPUT/OUTPUT LADDER/ PMC–PARAMETER BY MDI/DPL

11.9.1 Input/Output Method to Office Programmer (P–g Mate/Mark II) (Fixed 9600bit/Sec.)

- Method of Inputting/Outputting Ladder

(1) Select “Diagnose screen” by key in <DGNOS> key.

(2) Key in <READ>key or <WRITE> key.

(3) Turn on <F8> key from the office programmer menu screen, and key in menu number “5<NL>” or “3<NL>”.

11.9.2 Input/Output Method to FANUC FLOPPY CASSETTE (Fixed 4800bit/Sec.)

- Method of Inputting Ladder and PMC–Parameter.

(1) Select “Diagnose screen” by key in <DGNOS>key.

(2) Key in <NO.>key and optionally key in [File No.].

(3) Key in <READ>key.

CAUTION

In case of input PMC–Parameter, it is necessary to set following conditions.

(a) Emergency stop condition, and NC–Parameter PWE=1.

(b) Stop condition the Ladder program.

- Method of Outoutting Ladder.

(1) Select “Diagnose screen” by key in <DGNOS>key.

(2) Key in <NO.>key and optionally key in [Files No.].

(3) Key in <WRITE>.

- Method of Outputting PMC–Patameter.

(1) Select “PMC STATUS screen” by key in <DIGNOS>key.

(2) Key in <No.> key and optionally key in [File No.].

(3) Key in <WRITE>.

CAUTION

In case of output PMC–Parameter, it is necessary to set following condition.

(a) Edit mode.

(b) Stop condition the Ladder program.

11.10 ON–LINE DEBUGGING FUNCTION (ONLY FOR Power Mate–D/H)

The on–line debugging function enables the monitoring and modification of ladder programs and signal status on personal computer's screen using a personal computer connected to the Power Mate through an RS–232C cable.

> **NOTE**
> The additional option of the Ladder On–line debugging function is necessary to use the on–line debugging function on PMC–PA3 for Power Mate–D.

FANUC FAPT LADDER–II is necessary to use the on–line debugging function. (This software is a programming system for developing FANUC PMC sequence programs which operate on IBM PC/AT and compatible computers.)

Software name	Specification	Personal computer
FAPT LADDER–II	A08B–9201–J503	IBM PC/AT and compatible

In this section, only the parameter of on–line monitor driver for Power Mate–H and attention in use is described. Other points(connection of cable with personal computer, details of the operation, etc.) are described in the following manual.

Name of Manual	Spec.No.	Reference Items
FAPT LADDER–II OPERATOR'S MANUAL	B–66184EN	On–line function

11.10.1 Starting and Stopping the On–line Debugging Function

When using the on–line debugging function to connect a personal computer to the PMC, first start the driver that provides the communication function of the PMC.
When starting or stopping the driver, it is necessary to set either of the following parameters.

- Parameter screen for on–line monitor(「PARAMETERS FOR ONLINE MONITOR」)

  Pressing the [MONIT] then [ONLINE] soft keys on the PMC menu screen causes the on–line monitor parameter screen to appear.

  Parameter「RS–232C」=「USE」 : On–line monitor driver is used.

  「NOT USE」: On–line monitor driver is not used.

> **NOTE**
> The CRT/MDI is necessary when the parameter is set on the "PARAMETERS FOR ONLINE MONITOR" screen.

● Parameter in the Power Mate–H (No.0101#6)

	#7	#6	#5	#4	#3	#2	#1	#0
0101								

#6 = 0 : On–line monitor driver is not used.
1 : On–line monitor driver is used.

> **NOTE**
> In case of Power Mate–D Dual path control, only the parameter of first path side is effective.

When either of the following conditions consists, the on–line monitor driver is started.

●Parameter "RS–232C" is "USE"

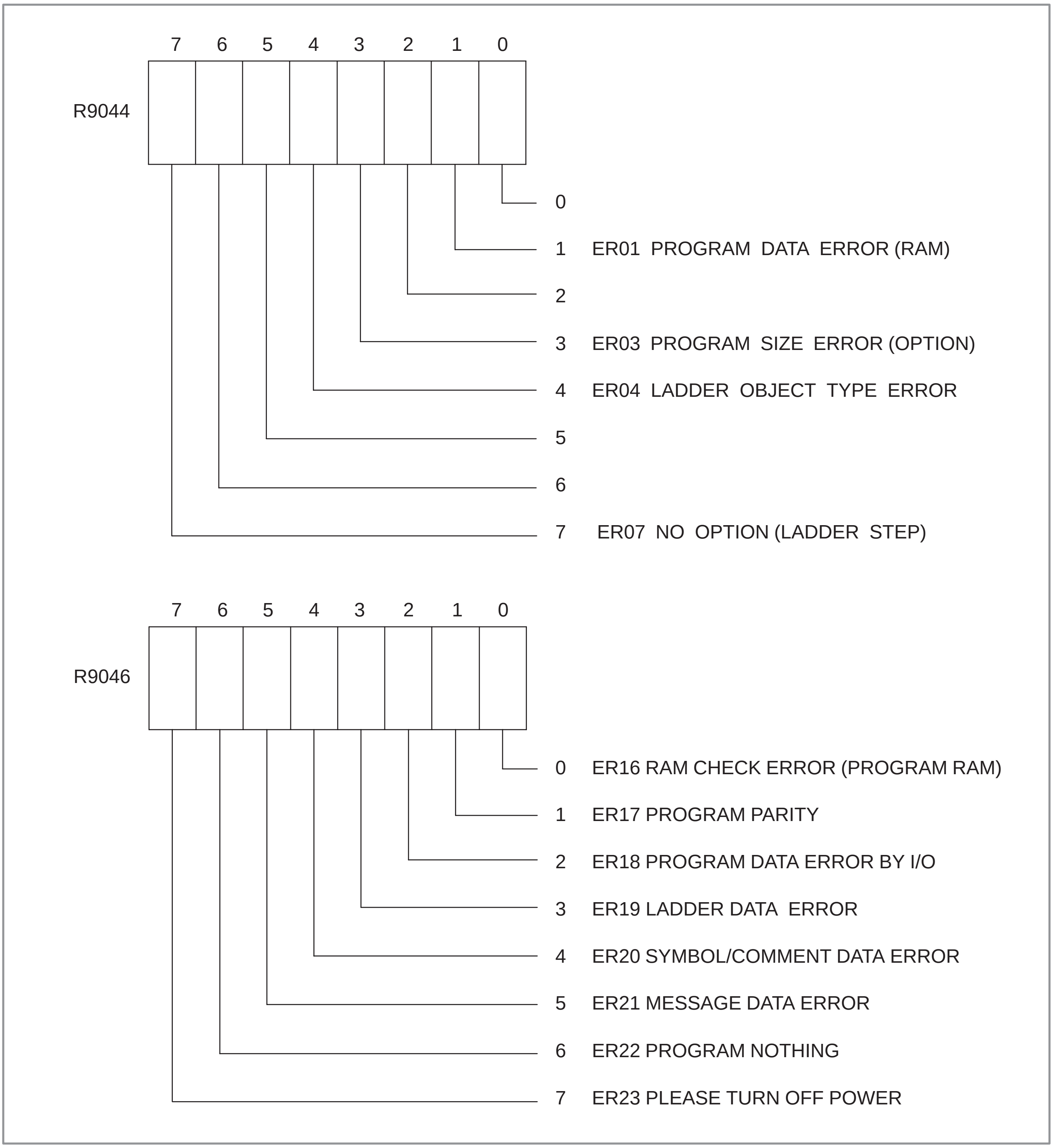
●Bit 6 of parameter No.0101 is "1"

> **CAUTION**
> 1 The on–line monitor driver occupies the line while it is operating.
> In this state, other input/output functions cannot use the line.
> If other input/output functions use the line, it is necessary to display the above–mentioned parameter and stop the on–line monitor driver.
> 2 While the on–line monitor driver is operating, the following functions cannot be used.
> ●[PMCLAD], [I/O], [EDIT], [SYSPRM] on CRT/MDI
> ●[EDIT], [SYSTEM PARAM], [I/O] on DPL/MDI
> 3 In case of operating NC, the screen display of NC(Position, etc.) might be slow when using input/output functions(Load from PMC, Store to PMC, etc.). There is no problem in the operation of NC. It is recommended to using input/output functions while NC is not operating.
> 4 When the screen made by C language executor is displayed, the communication speed decreases. It is recommended to use input/output functions after moving to other screens(Position, etc.).

11.11 ERROR LIST

If in alarm is issued in the PMC, the alarn message is displayed on the CRT (PMC ALARM MESSAGE screeen). But in case of DPL/MDI, it is displayed only by R–relay status (ON or Off).
Refer to the “APPENDIX L.ALARM MESSAGE LIST” for more information.

(1) Error ststus at power on or PROGRAM DOWN LOAD.

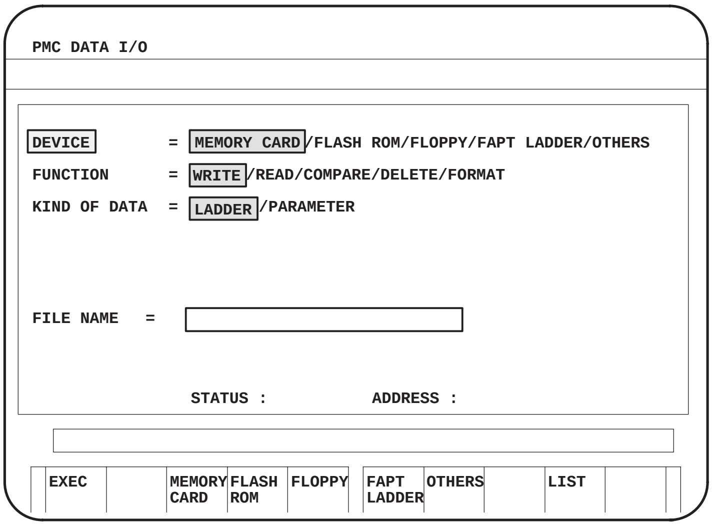


12 PMC–NB6 PROGRAM OPERATION SCREEN

12.1 WRITING, READING, AND COMPARING SEQUENCE PROGRAMS AND PMC PARAMETERS

12.1.1 I/O Screen

When the [I/O] soft key on the PMC main menu is clicked, the following screen appears.



On this screen, sequence programs and PMC parameters can be written to a specified device, read from the device, and compared. The question selection cursor, which moves vertically from one question to another, is displayed, as is the option selection cursor, which moves horizontally from one option to another. The displayed soft keys differ depending on the position of the question selection cursor.

The following types of devices can be used for input/output. The desired device type can be selected by positioning the question selection cursor to “DEVICE” and either moving the option selection cursor to that type or selecting the soft key corresponding to that type.

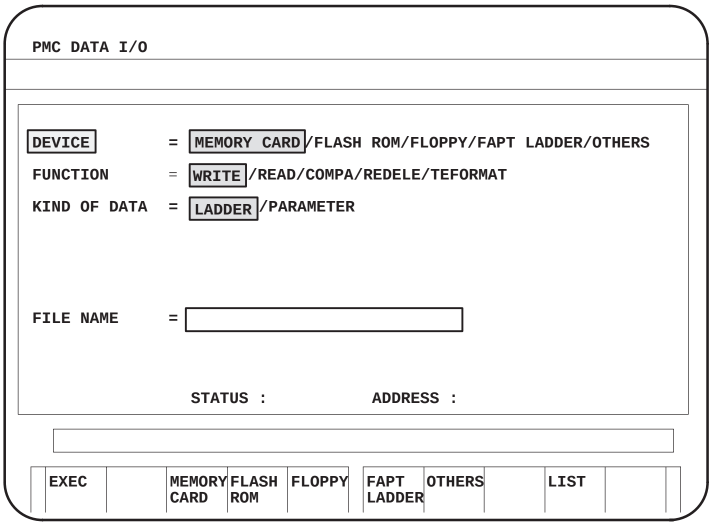
- MEMORY CARD: Data can be output to and input from a memory card.
- FLASH ROM: Data can be output to and input from flash ROM.
- FLOPPY: Data can be output and input to and from handy files and floppy cassettes.
- FAPT LADDER: Data can be output to and input from FAPT LADDER–II.
- OTHERS: Data can be output to and input from other input/output devices.

> **WARNING**
> If a sequence program is input while a Ladder program is being executed, the execution of the Ladder program stops automatically.

NOTE

1. The [I/O] soft key is displayed by setting bit 1 of keep relay K900 to 1.
2. For an explanation of I/O–related error messages, see section III.7.4, “I/O–Related Error Messages.”
3. The “STATUS” field near the bottom of the screen displays the execution status of the writing, reading, comparison, and other functions.
4. The “ADDRESS” field near the bottom of the screen displays the address being executed while the writing, reading, comparison, and other functions are being executed, in real–time as processing progresses.

12.1.2 Outputting to and Inputting from Memory Cards



When “MEMORY CARD” is selected for DEVICE, output to and input from memory cards are enabled.

- FUNCTION

  Select the data input/output command: Position the cursor to FUNCTION, then move the option selection cursor to the desired command or select it with the corresponding soft key.

Soft keys displayed when the question selection cursor is positioned to “FUNCTION”

Explanation of options

WRITE: Outputs data from the PMC to a memory card.

READ: Inputs data from a memory card to the PMC.

COMPARE: Compares the sequence programs on the PMC with those on a memory card.

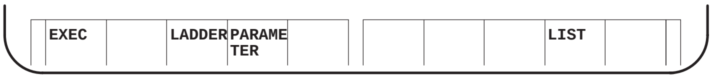
DELETE: Deletes files from a memory card.

FORMAT: Formats a memory card.

CAUTION
When “FORMAT” is selected and executed, all data in the memory card is lost. Be careful when executing this function.

- KIND OF DATA
  KIND OF DATA is displayed only when “WRITE” is selected for “FUNCTION.”
  Set the type of data to be output by moving the cursor horizontally to that type or by clicking the corresponding soft key.

Soft keys displayed when the question selection cursor is positioned to “KIND OF DATA”



Explanation of options

LADDER: Outputs sequence programs only.

PARAMETER: Outputs PMC parameters.

- FILE NO.
  FILE NO. is displayed only when “READ,” ”COMPARE,” or “DELETE” is selected for “FUNCTION.”
  Enter the file number in the edit box.
- FILE NAME
  FILE NAME is displayed when “WRITE,” “READ,” “COMPARE,” or “DELETE” is selected for “FUNCTION.”
  Enter the file name in the edit box.
  When “READ,” “COMPARE,” or “DELETE” is selected for “FUNCTION,” the file name corresponding to the file number entered in “FILE NO.” is displayed automatically.
  The file name must be in MS–DOS format: a file name of up to eight characters followed by an extension of up to three characters.
  When “WRITE” is selected for “FUNCTION” and the file name is not entered, the following names are automatically assumed.

DATA KIND	File name
LADDER	PMC–BN6.LAD
PARAM	PMC–NB6.PRM

CAUTION
When both “FILE NO.” and “FILE NAME” are displayed at the same time, and a value is entered for “FILE NO.” and another file name is entered in “FILE NAME,” the value entered in “FILE NO.” is erased and the file name entered in “FILE NAME” becomes effective.

Explanation of soft keys

[EXEC]: Executes the function selected for "FUNCTION." During execution, the soft key disappears and the [CANCEL] soft key appears to the right of the key.

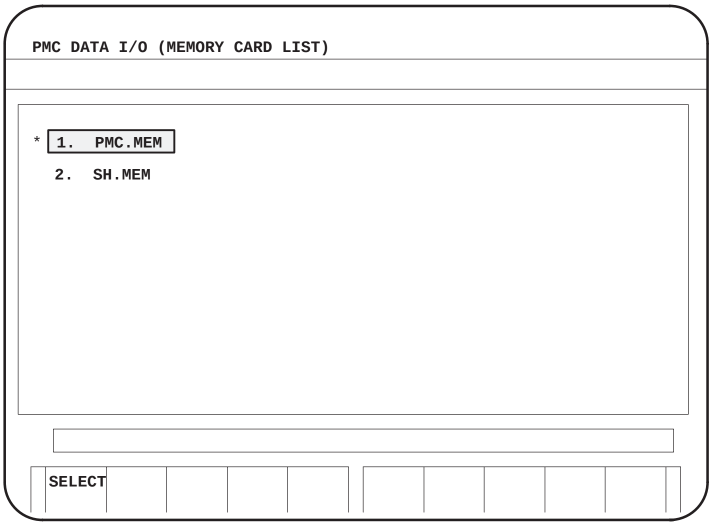
[CANCEL]: Cancels the execution of the function. When the function terminates normally, the soft key disappears.

[LIST]: Replaces the current display with the memory card list screen. See Section 12.1.3, "List screen" for details.

> **NOTE**
> For an explanation of supported flash memory cards, see "(1) Writing to flash memory cards" in Section III.7.3.4.

12.1.3 List Screen

When the [LIST] soft key is clicked, the following screen appears.

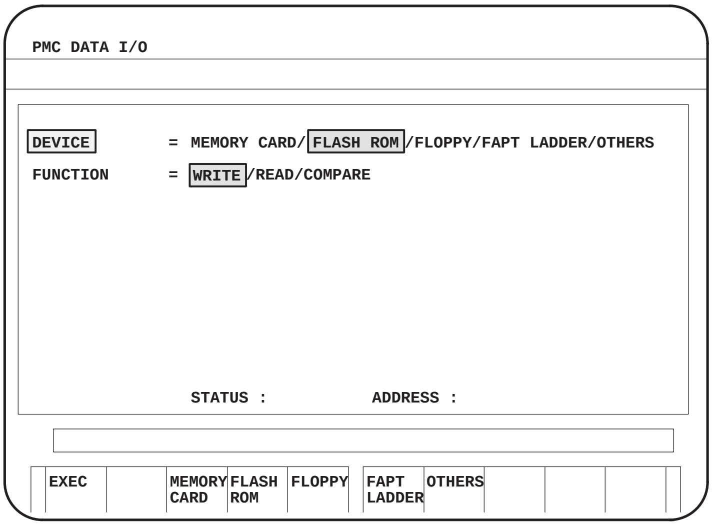


The contents of the memory card are displayed. From this screen, a single file can be selected.
Move the cursor vertically to the desired file and click the [SELECT] soft key to select it. When the file is selected, an asterisk "*" appears to the left of the file name.
To return to the previous screen (the "MEMORY CARD" setting screen in the above case), click the leftmost soft key. By default, when you select a file and return to the previous screen, the cursor is positioned to "READ" in "FUNCTION" and the file number and name selected from the list screen are displayed in "FILE NO." and "FILE NAME."

Explanation of the soft key

[SELECT]: Selects a file. An asterisk (*) appears to the left of the selected file. Clicking [SELECT] again on the selected file deselects the file and causes the "*" to disappear.

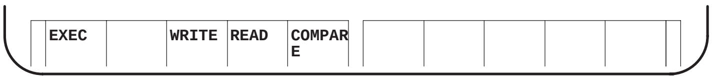
12.1.4 Outputting to and Inputting from Flash ROM



When “FLASH ROM” is selected for DEVICE, output to and input from flash ROM are enabled.

- FUNCTION
  The available data input/output commands are displayed. Select the desired command by moving the cursor horizontally to that command or select it with the corresponding soft key.

Soft keys displayed when the cursor is positioned to ”FUNCTION”



Explanation of options

WRITE: Outputs sequence programs from the PMC to flash ROM.

READ: Inputs sequence programs from flash ROM to the PMC.

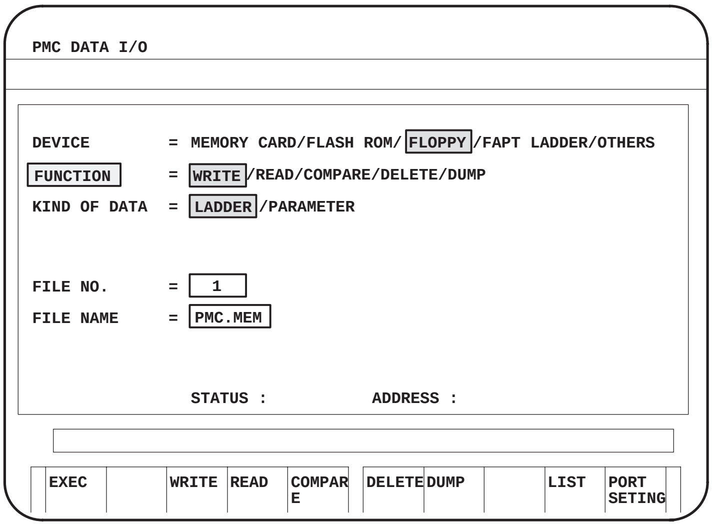
COMPARE: Compares the sequence programs on the PMC with those on flash ROM.

Explanation of soft keys

[EXEC]: Executes the function selected for “FUNCTION.” During execution, the soft key disappears and the [CANCEL] soft key appears to the right of the key.

[CANCEL]: Cancels the execution of the function. When the function terminates normally, the soft key disappears.

12.1.5 Outputting to and Inputting from Floppy Disks



When “FLOPPY” is selected for DEVICE, output to and input from handy files or floppy cassettes are enabled.

- FUNCTION
  The available data input/output commands are displayed. Select the desired command by moving the cursor horizontally to that command or select it with the corresponding soft key.

Explanation of options

WRITE: Outputs data from the PMC to a floppy disk.

READ: Inputs data from a floppy disk to the PMC.

COMPARE: Compares the sequence programs on the PMC with those on a floppy disk.

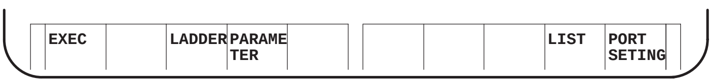
DELETE: Deletes a file from a floppy disk.

DUMP: Outputs a dump of a specified range of physical addresses to a floppy disk.

NOTE
The DUMP command is displayed only when bit 4 of keep relay K900 is set to 1.

- KIND OF DATAKIND OF DATA is displayed only when “WRITE” is selected for “FUNCTION.”
  Set the type of data to be output by moving the cursor horizontally to that type or by clicking the soft key corresponding to it.

Soft keys displayed when the question selection cursor is positioned to “KIND OF DATA”



LADDER: Outputs sequence programs only.

PARAMETER: Outputs PMC parameters.

- HOW TO WRITE
  HOW TO WRITE is displayed only when "WRITE" or "DUMP" is selected at "FUNCTION."
  Specify the style of output (writing): Set the desired style by moving the cursor horizontally to that style or by clicking the soft key corresponding to it.

Soft keys displayed when the question selection cursor is positioned to "HOW TO WRITE"

EXEC		ADD	INIT					LIST	PORT SETING

ADD: Adds data to the end of an existing file.

INT: Adds data to the beginning of a file. The previous contents are erased.

- FILE NO.
  FILE NO. is displayed only when "READ," "COMPARE," or "DELETE" is selected for "FUNCTION."
  Enter the file number in the edit box.
- FILE NAME
  Enter the file name in the edit box.
  When "READ," "COMPARE," or "DELETE" is selected for "FUNCTION," the file name corresponding to the file number entered in "FILE NO." is displayed automatically.
  The file name must be in MS–DOS format: a file name of up to eight characters followed by an extension of up to three characters.
  When "WRITE" or "DUMP" is selected for "FUNCTION" and no file name is entered, a name is automatically assigned. (For "DUMP," the name is "PMC–NB6.DMP.")

CAUTION

1 When a value is entered in "FILE NO." and another file name is entered in "FILE NAME," the value entered in "FILE NO." is erased and the file name entered in "FILE NAME" becomes effective.

2 Specifying the same name as that of an existing file results in an error.

- START ADDRESS
  START ADDRESS is displayed under "FILE NAME" only when "DUMP" is selected for "FUNCTION."
  In the edit box, enter the first address to be dumped.
- SIZE
  SIZE is displayed under "START ADDRESS" only when "DUMP" is selected for "FUNCTION."
  In the edit box, enter the size of the addresses to be dumped.

CAUTION

A system error may occur depending on the entered address. Be careful when using this function.

Explanation of soft keys

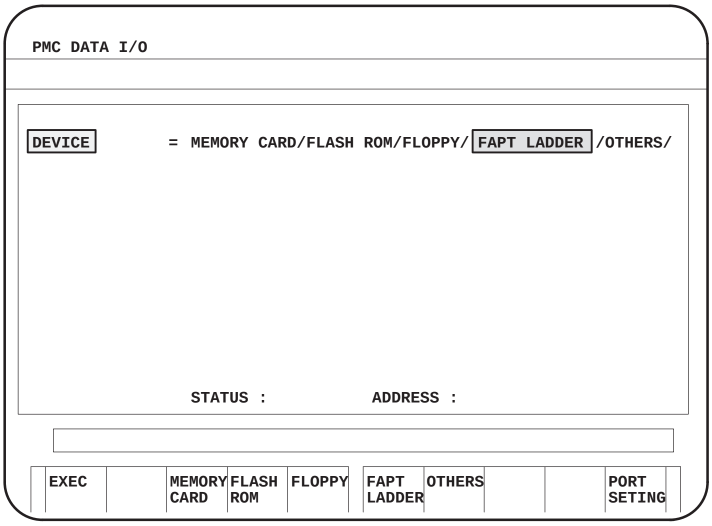
[EXEC]: Executes the function selected for "FUNCTION." During execution, the soft key disappears and the [CANCEL] soft key appears to the right of the key.

[CANCEL]: Cancels the execution of the function. When the function terminates normally, the soft key disappears.

[LIST]: Replaces the current display with the MEMORY CARD list screen. See Section 12.1.3, "List screen" for details.

[PORT SETING]: Replaces the current display with the screen for setting communication parameters. See Section 12.1.8, "PORT SETTING screen" for details.

12.1.6 Outputting to and Inputting from FAPT LADDER



When "FAPT LADDER" is selected for DEVICE, output to and input from FAPT LADDER–II are enabled. There are no other questions.

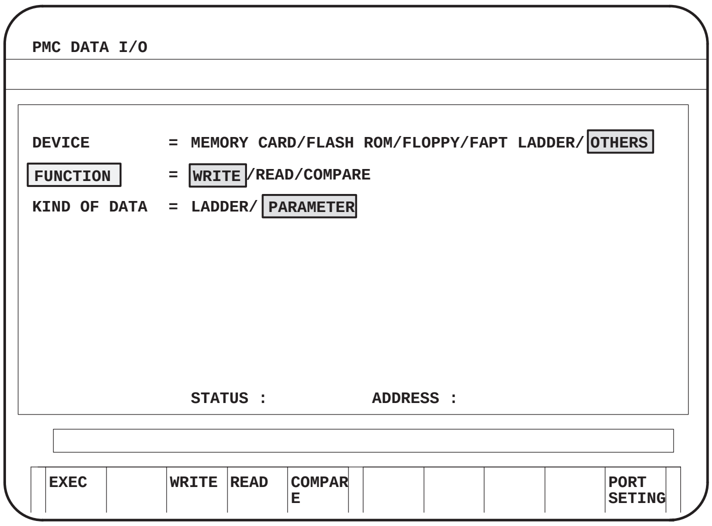
Explanation of soft keys

[EXEC]: Places the NC in the standby state and causes "CANCEL" to appear to its left. The WRITE, READ, and other functions are executed by FAPT LADDER–II.

[CANCEL]: Cancels the execution of the function. When the function terminates normally, the soft key disappears.

[PORT SETING]: Replaces the current display with the screen for setting communication parameters. See Section 12.1.8, "PORT SETTING screen" for details.

12.1.7 Outputting to and Inputting from Other Input/Output Devices



When “OTHERS” is selected for DEVICE, output to and input from other input/output devices are enabled.

- FUNCTION
  The available data input/output commands are displayed. Select the desired command by moving the cursor horizontally to that command or select it with the corresponding soft key.

Explanation of options

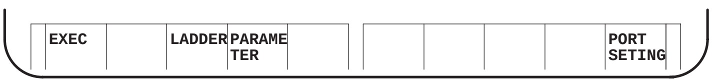
WRITE: Outputs data from the PMC to another input/output device.

READ: Inputs data from another input/output device to the PMC.

COMPARE: Compares the sequence programs on the PMC with those on another input/output device.

- KIND OF DATAKIND OF DATA is displayed only when “WRITE” is selected for “FUNCTION.”
  Set the type of data to be output by moving the cursor horizontally to that type or by clicking the soft key corresponding to it.

Soft keys displayed when the question selection cursor is positioned to “KIND OF DATA”



LADDER: Outputs sequence programs only.

PARAMETER: Outputs PMC parameters.

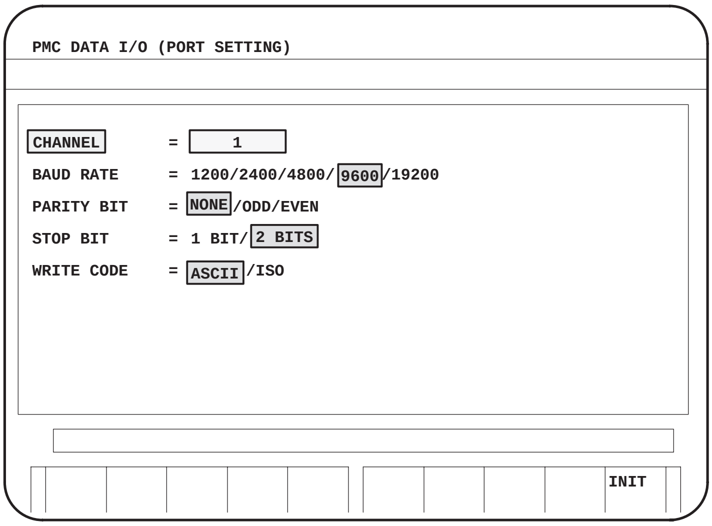
Explanation of soft keys

[EXEC]: Executes the function selected for “FUNCTION.” During execution, the soft key disappears and the [CANCEL] soft key appears to the right of the key.

[CANCEL]: Cancels the execution of the function. When the function terminates normally, the soft key disappears.

[PORT SETING]: Replaces the current display with the screen for setting communication parameters. See Section 12.1.8, “PORT SETTING screen” for details.

12.1.8 PORT SETTING Screen



When any of “FLOPPY,” “FAPT LADDER,” or “OTHERS” is selected at the DEVICE question, the “PORT SETING” soft key is displayed. When the key is clicked, the PORT SETTING screen appears. This screen allows the setting of the communication data required for communication using the RS–232C and other ports. Communication data can be set for each of the three types of DEVICEs independently of the others.

Explanation of each question

- CHANNEL
  Check that an RS–232C cable is connected to the main board of the control unit. Directly enter the number corresponding to the connected connector.
  1 . . . . JD5A
  2 . . . . JD5B
- BAUD RATE
  1200: Sets the baud rate to “1200.”
  2400: Sets the baud rate to “2400.”
  4800: Sets the baud rate to “4800.”
  9600: Sets the baud rate to “9600.”
  19200: Sets the baud rate to “19200.”
- PARITY
  NONE: Sets no parity.
  ODD: Sets “odd” parity.
  EVEN: Sets “even” parity.
- STOP BIT
  1 BIT: Sets the number of stop bits to “1.”
  2 BITS: Sets the number of stop bits to “2.”
- WRITE CODE
  WRITE CODE is displayed when “OTHERS” is selected for “DEVICE.”
  ASCII: Sets the input/output code to “ASCII.”
  ISO: Sets the input/output code to “ISO.”

Explanation of soft keys

[INIT]: Sets all the parameters to their initial values.

	DEVICE = FAPT LADDER	**DEVICE = FLOPPY**	**DEVICE = OTHERS**
CHANNEL	1	1	1
BAUD RATE	9600	4800	4800
PARITY BIT	NONE	NONE	NONE
STOP BIT	2 BITS	2 BITS	2 BITS
WRITE CODE	(None)	(None)	ISO

12.2 STARTING AND STOPPING SEQUENCE PROGRAMS

(1) Starting a sequence program (RUN)
When a program is stopped, clicking the [RUN] soft key causes the program to start and the status line display to change to "PMC RUN." The sequence program starts from the beginning. The soft key changes to [STOP].

(2) Stopping a sequence program (STOP)
When a program is executed, clicking the [STOP] soft key causes the program to stop and the status line display to change to "PMC STOP." The soft key changes to [RUN].

WARNING
If the sequence program is stopped while the machine is operating, the machine may behave in an unexpected way. Before stopping the sequence program, ensure that there are no people near the machine and that the tool cannot collide with the workpiece or machine.
Otherwise, there is an extreme risk of death or serious injury, as well as the likelihood of the tool, workpiece, and machine being damaged.

(3) Automatic operation of a sequence program
When AUTOMATIC LADDER START is set to AUTO (bit 2 of the keep relay K900 = 0) on the setting screen, a sequence program can be executed automatically when the power is turned on.

IV. STEP SEQUENCE FUNCTION

1 GENERAL

1.1 STEP SEQUENCE METHOD

The ladder method is most often used for programming the sequence control governed by a programmable controller. This method, shown in Fig.1.1(a) , was derived from relay-panel control circuits. Since it has been in use for years, many sequence control engineers are already familiar with it. This method is also used in PMC sequence programming.

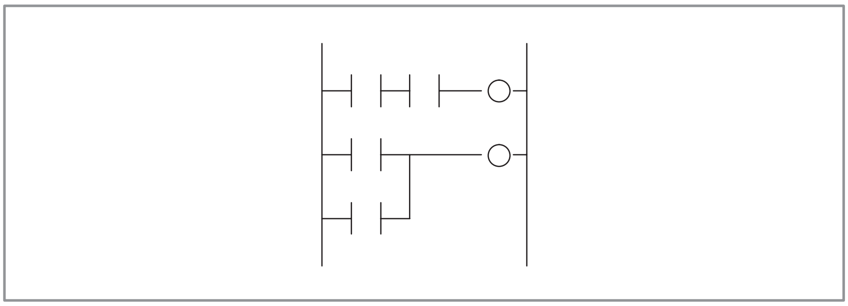


Fig. 1.1 (a) Ladder method

The greater the number of functions implemented by the PMC for a CNC system, the larger and the more complicated the sequence program becomes. A large-scale system requires a larger program and a greater number of processes, making it hard for the ladder method to control the overall process. This is because the ladder method does not describe the order of control. While the ladder method is suitable for describing partial control, it is hard to apply it to the description of the flow of control overall.

To overcome this problem, structured programming has been introduced into sequence control. A PMC that supports the subprogram function enables the use of modular programs. As shown in Fig.1.1(b), a large-scale program is divided into subprograms for each function, simplifying the unit of processing. Since the programmer determines how to divide the main program into subprograms and the control flow used to call the subprograms, however, the programs are not necessarily easy-to-understand by other programmers.

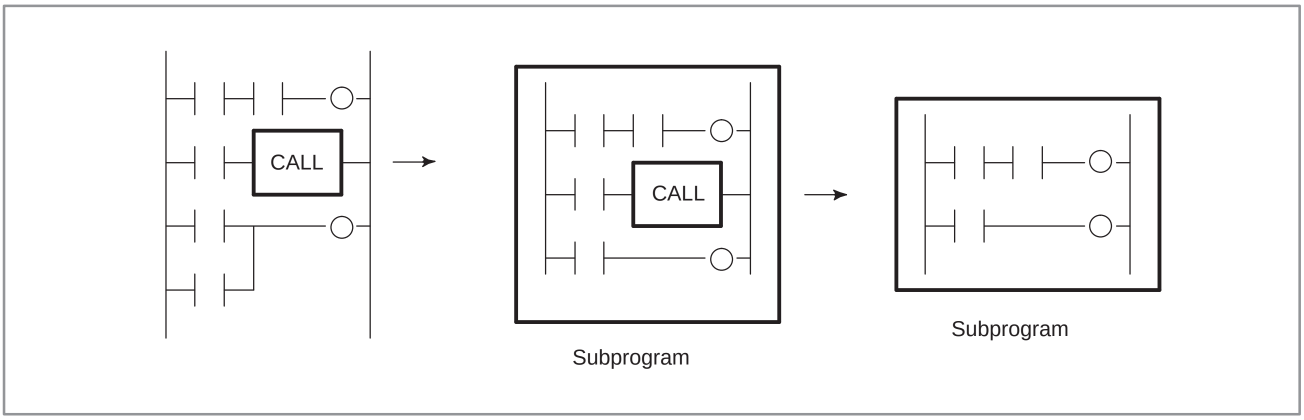


Fig. 1.1 (b) Module method

Given these conditions, a step sequence method has been created to describe programs structurally. It is well-suited to the control of entire processes and provides an easy-to-understand visualized flow of the process. The step sequence programming features the direct representation of the control flow on a flow chart, as shown in Fig.1.1(c). Each block of processing is described as a subprogram, using the ladder method. The entire program is then created by combining these subprograms.

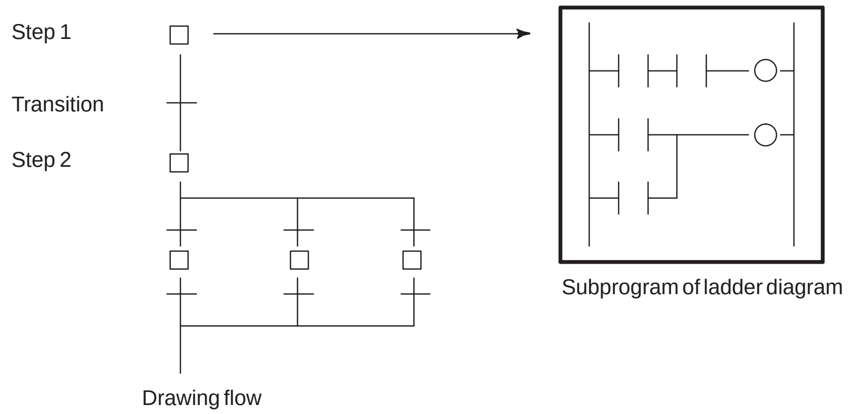


Fig. 1.1(c) Step sequence method

The step sequence method has the following features:

(1) Increased programming efficiency

- Since the flow of processes can be programmed directly, simple, correct programming is enabled, reducing the time required for programming.
- Even for complicated control, programming proceeds from the main flow to detailed flow in each process, creating a structured, top-down program, which is easy-to-understand by persons other than the original creator.
- Structured modules can be used again easily.

(2) Easy debugging and maintenance

- Graphical display enables the operator to easily understand the execution state of a program visually.
- Erroneous steps in a program can be found easily.
- A part of a program can be easily modified.

(3) High-speed program

- Since only the subprograms required for a certain process are executed, the cycle time is reduced.

(4) Transition from ladder programs

- Since steps and transitions consist of conventional ladder programs, conventional ladder programs can be converted to new step sequence programs, without discarding ladder-program resources.

In step sequence programming, a sequence control program is divided into two types of subprograms, steps and transitions. Steps describe processes. Transitions connect steps and determine whether the transition conditions from one step to another evaluate true. As shown in Fig.1.1(d), a step sequence program is described using graphical symbols.

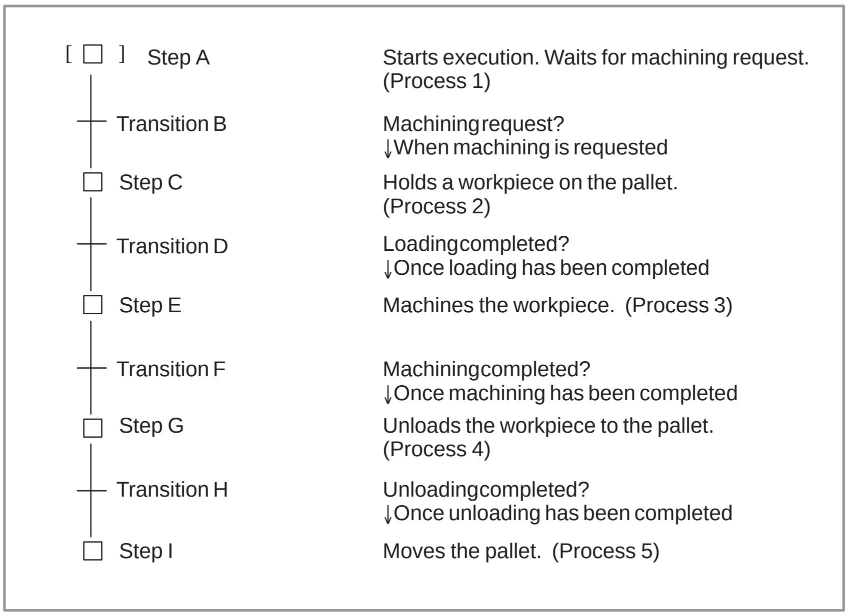


Fig. 1.1 (d) Example of machining the workpiece

As shown in this example, the program flow from process 1 through process 5 is expressed visually. Detailed programs related to the movements performed aspart of each process, and the signals used for determining whether transition conditions for proceeding to the next step are satisfied, are not described here. To program complicated control flows, many other functions are supported, such as divergence, jump, and nesting functions. The details of these functions are described later.

Step sequence programming is suitable for creating programs which control processes sequentially. Programs used for controlling a unit which operates according to a certain sequence, such as a loader, ATC, and other peripheral units, are best suited to step sequence programming. For programs which control units with no particular sequence, such as that of the operator’s panel which is always monitoring the emergency stop signal or mode signals, however, are not well-suited to step sequence programming. The PMC supports the advantages of both methods, ladder and step sequence programming, by calling subprograms written according to a step sequence and those written as a ladder, from the main program.

1.2 GRAPHICAL SYMBOLS

This manual uses the graphical symbols listed in Table 1.2 to describe step sequence flowcharts. Depending on the character font being used, the actually displayed symbols may differ slightly from those listed here. These graphical symbols are described in the subsequent chapters.

Table 1.2 List of graphical symbols

Contents	Display of programming manual	Display	
		CNC Device	FAPT LADDER of Personal Computer
Step	□ Sn	□ Sn	□ Sn
Initial Step	[□] Sn	[□] Sn	[□] Sn
Transition	┼ Pn	┼ Pn	┼ Pn
Divergence of Selective Sequence			
Convergence of Selective Sequence			
Divergence of Simultaneous Sequence			
Convergence of Simultaneous Sequence			
Jump	→ Ln	– > Ln	– > Ln
Label	Ln →	< – Ln	< – Ln
Block Step	□] Sn	□] Sn	□] Sn
Initial Block Step	[□] Sn	[□] Sn	[□] Sn
End of Block Step	⊥	⊥	⊥

1.3 PROGRAMMING

Follow the procedure below to create a step sequence program. Use a personal computer on which the FAPT LADDER software package is installed to code (edit) a program. Use a CNC to execute, debug and correct the ladder subprogram.

(1) Create step sequence program (editing)

(2) Create a subprogram of ladder diagram (editing)

(3) Compile

(4) Transfer to the CNC device (with the memory card or RS232C)

(5) Write to the FlashROM

(6) Execute

(7) Diagnosis and debugging

(8) Correct a subprogram of ladder diagram (editing)

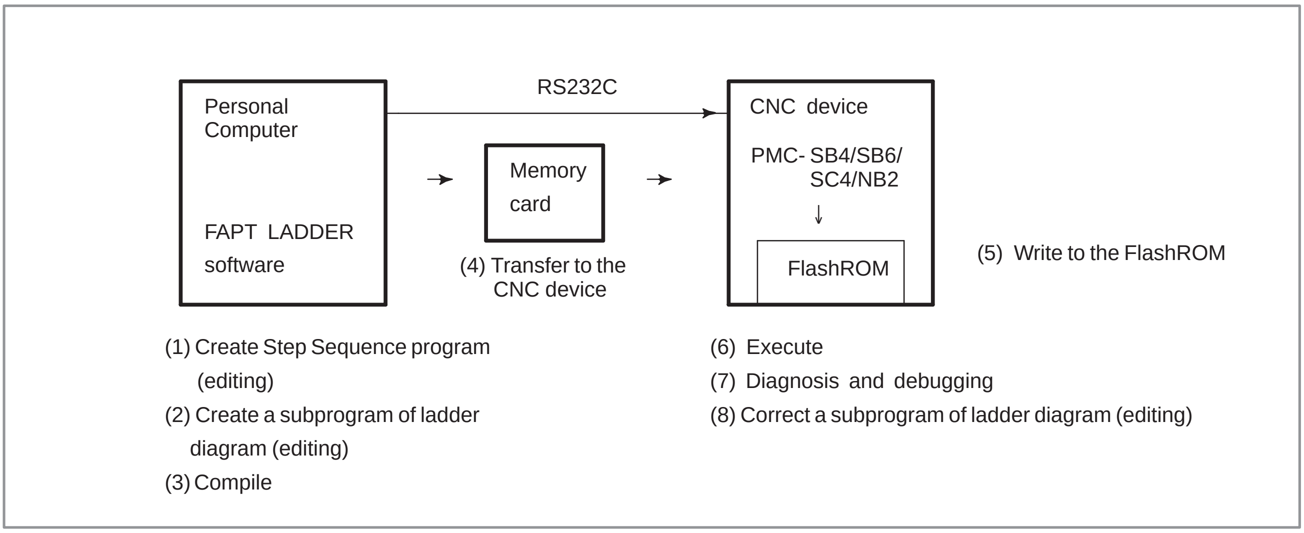


Fig. 1.3 Programming to create a program

Table1.3 lists the step sequence functions supported by a personal computer (on which the FAPT LADDER software package is installed) and CNC.

Table 1.3 Step sequence functions

○ : usable

Functions	PMC-SB4/ SB6	PMC-SC4	PMC-NB2	FAPT LADDER of PERSONAL COMPUTER
Display and edit of a program				
• Display of subprogram list	○	○	○	○
• Create a new subprogram				○
• Delete a subprogram				○
• Edit a subprogram of StepSequence form				○
• Edit a subprogram of ladder diagram	○	○	○	○
• Compile				○
• Decompile				○
Input and output				
• Input and output with a memory card	○	○	○	○
• Input and output with RS232C	○	○	○	○
• Write to a FlashROM	○	○	○	
Execution of program				
• execution of a ladder diagram	○	○	○	
• execution of Step Sequence program	○	○	○	
Diagnosis and debugging (note1)				
• Diagnosis of Step Sequence program	○	○	○	
• Diagnosis of a ladder diagram	○	○	○	
• Set and display a monitoring timer	○	○	○	

NOTE

While step sequence functions are being used, some of the diagnosis and debug functions supported by the ladder method cannot be used. For details, see 6.4 (Support Functions).

2 STEP SEQUENCE BASICS

2.1 TERMINOLOGY

A step sequence program is created using a variety of graphical symbols, as shown in Fig. 2.1 (a). The main terms used in the step sequence are described below.

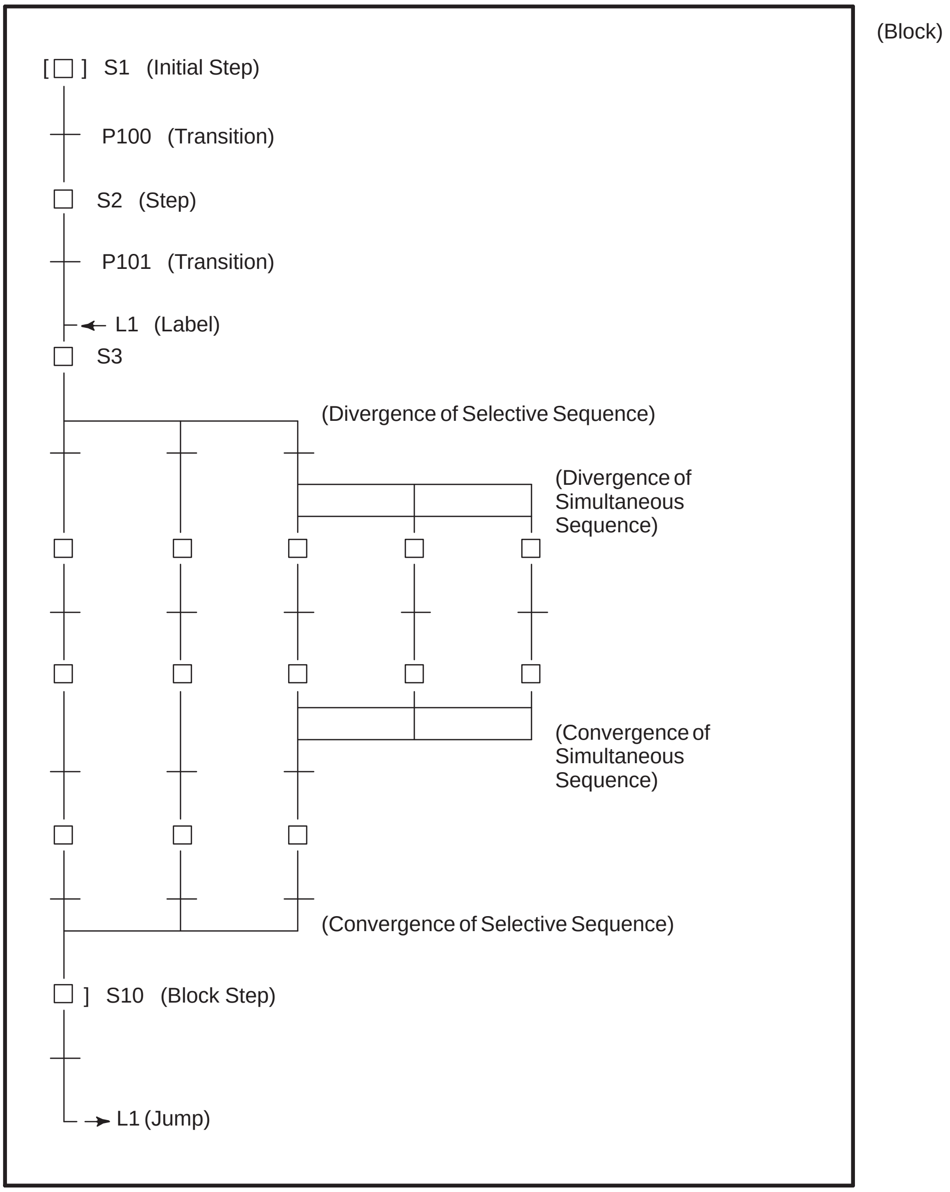


Fig. 2.1 (a) Step sequence elements

(1) Step

A step indicates a process, which is the basic processing unit in a step sequence program. In a step, specify the S address (Sn), which is a step number, and P address (Pm), which indicates a subprogram (action program) specifying the details of processing in each step.

(2) Step state transition

When a step sequence program is executed, the process proceeds as program processing advances, the state of each step changs accordingly. Each step can assume any of the logical states listed in Table 2.1, its state changes as shown in Fig. 2.1 (b). Activation refers to the changing of a step from the inactive state to the active state. Inactivation refers to the changing of a step from the active state to the inactive state.

Table 2.1 Step state

State		Processing	Display
Active	Execution	Activated step. The action program (subprogram) is being executed.	■ Sn
Inactive	Transition to halt	Transition from execution to halt. The action program (subprogram) is executed once only, then the step automatically transits to halt.	□ Sn
	Halt	Not activated state. The action program (subprogram) has not yet been executed.	

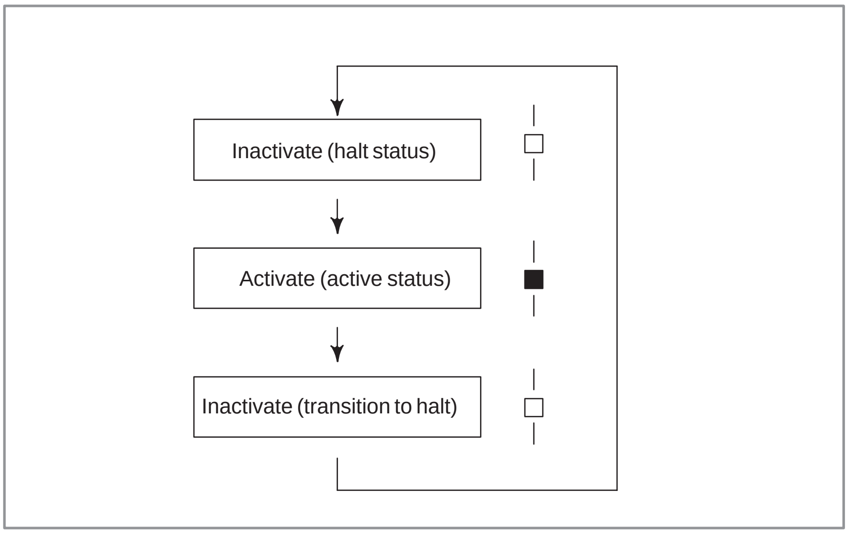


Fig. 2.1 (b) Step state transition

(3) Transition

A transition denotes the transition conditions. When these evaluate true, the step of the corresponding state changes from the inactive to active state or vice the reverse. Specify the P address (Pn), which indicates a subprogram describing the transition conditions in detail.

As shown in Fig. 2.1 (c), step S2 changes its state from inactive to active when the conditions described in transition P10 evaluate true, while step S2 changes its state from active to inactive when the conditions described in transition P20 evaluate true.

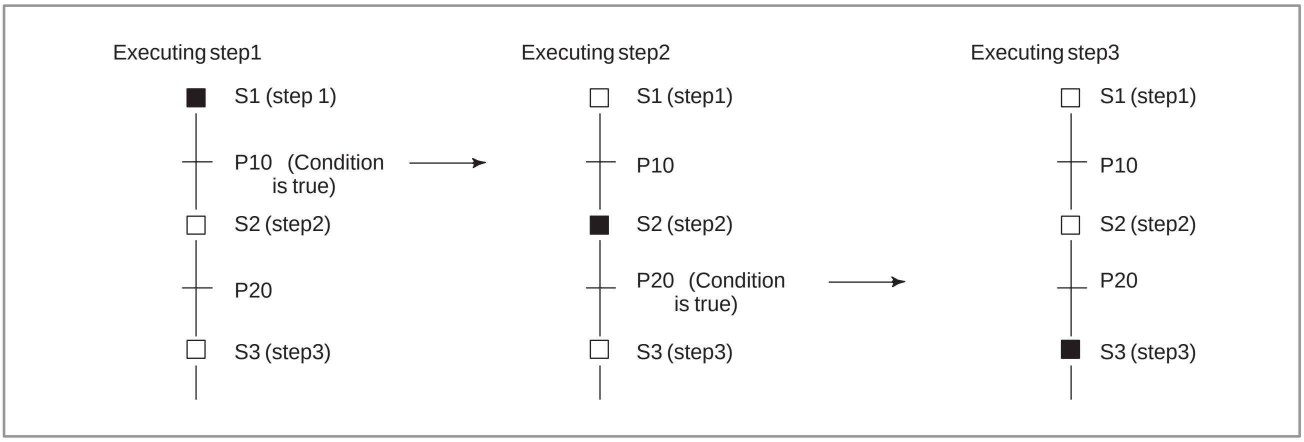


Fig. 2.1 (c) Transition of step state by the transition

Note that the step immediately before a transition must be active in order to switch the next step from inactive to active when the conditions specified in the transition evaluate true. As shown in Fig. 2.1 (d), step S3 does not change to the active state, even when transition P20 evaluates true, if step S1 is active and step S2 is inactive. An active state passes from a certain step to the next step when the corresponding transition conditions evaluate true, the execution of the step sequence program advancing one step.

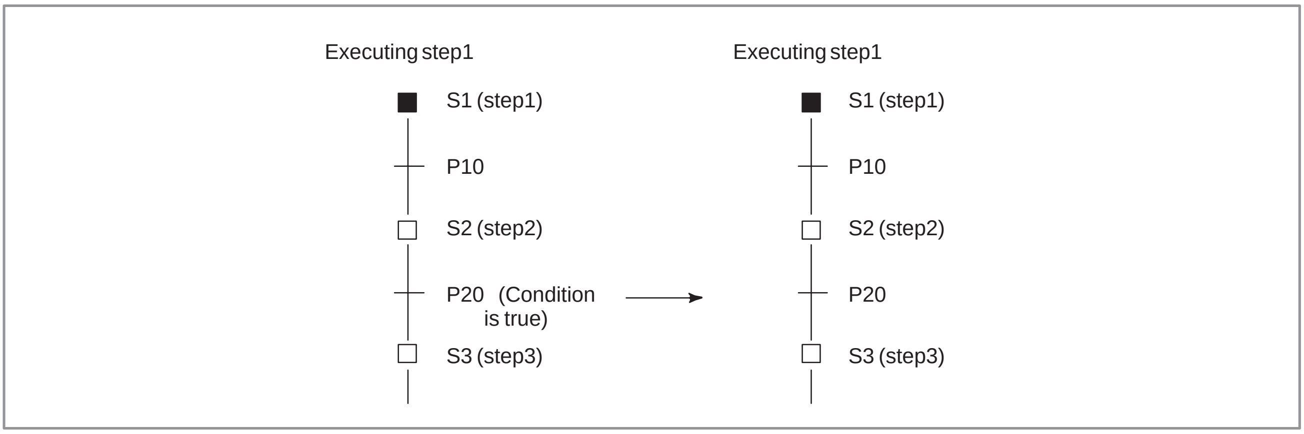


Fig. 2.1 (d) Transition of step state by transition

(4) Initial Step

While a normal step can be activated by a transition, the initial step is activated automatically when execution of the program starts, as shown in Fig. 2.1 (e).

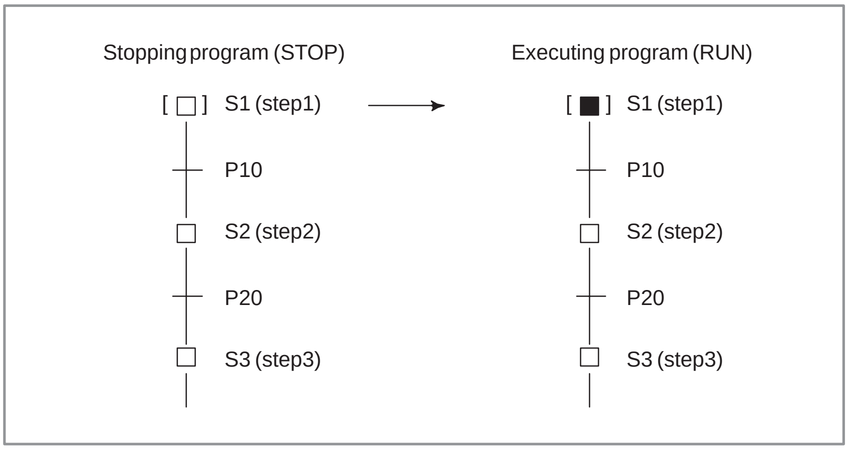


Fig. 2.1 (e) Activate of initial step

Although the initial step, which is usually executed first, is often placed at the top of a program, it can also be specified at some point within a program. It is always activated first. After being deactivated once, it can be subsequently be activated again. In this case, it acts in the same way as a normal step.

(5) Divergence and Convergence of Selective Sequence

To describe a complicated sequence, selective sequences can be used. A selective sequence offers multiple choices, from among which the condition becomes true first activates the corresponding step, as shown in Fig. 2.1 (f). The divergent paths join to generate the mai sequence.

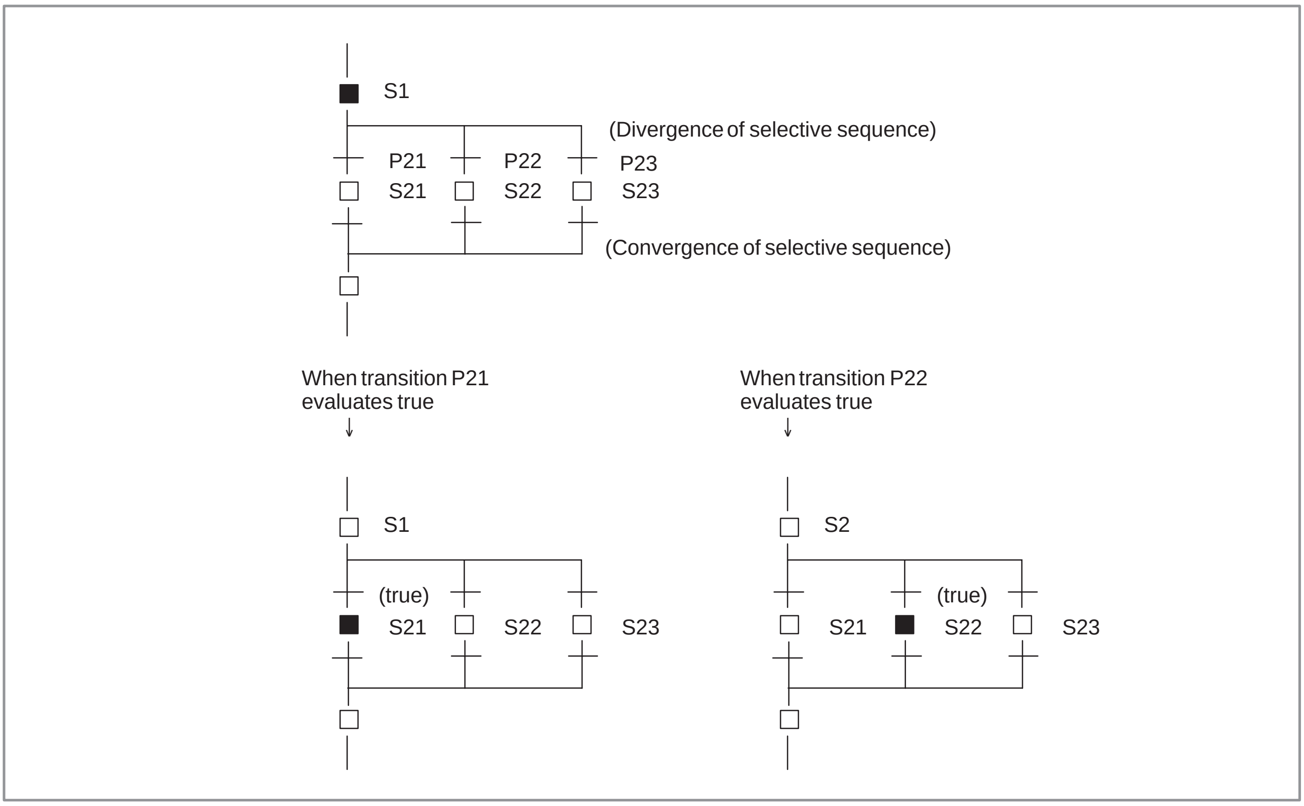


Fig. 2.1 (f) Selective sequence

(6) Divergence and Convergence of Simultaneous Sequence

A Simultaneous sequence can be used to execute multiple processes simultaneously. In a Simultaneous sequence, as shown in Fig. 2.1 (g), one transition activates multiple steps. The activated multiple steps are executed independently. Once all steps along the multiple paths have been completed, the divergent paths join to generate the main sequence.

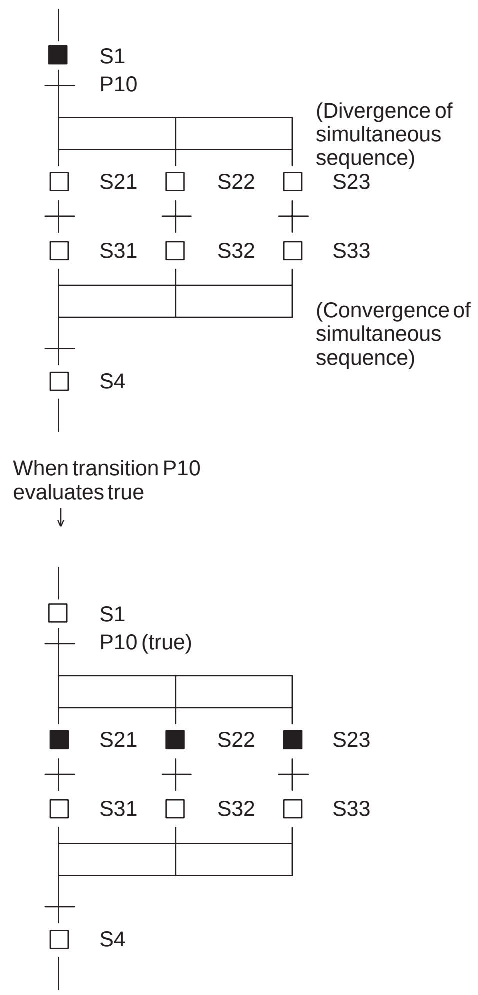


Fig. 2.1(g) Simultaneous sequence

(7) Jump and Label

The jump function is used to describe a non–serial sequence, such as a repeated loop. As shown in Fig. 2.1 (h), when a jump designation is activated, the sequence jumps to the step having the corresponding jump destination label, after which that step is activated. To specify a label number, the L address is used in the same way as a jump instruction in ladder programming. A jump can be made to a previous or subsequent step.

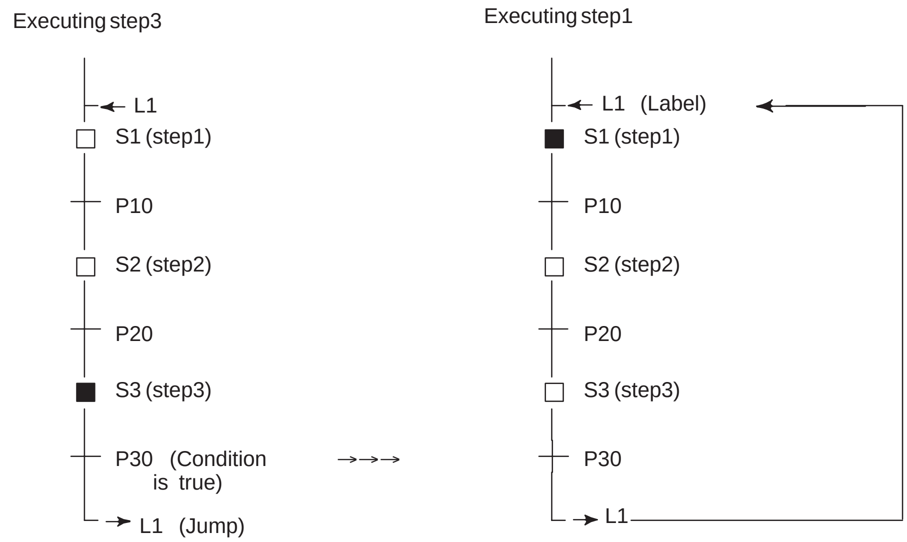


Fig. 2.1 (h) Jump and Label

(8) Block

A block refers to a group of consecutive steps and transitions. A block can be a step sequence program. The more complicated the sequence becomes, the larger and more complex the block is. A program can be divided into multiple blocks in the same way as for subprograms in ladder programming, based on the concept of modular programming. Each block is identified by a P address, which corresponds to the subprogram number in ladder programming.

A block is executed as the main program in a step sequence, or called from another step sequence program as a subprogram.

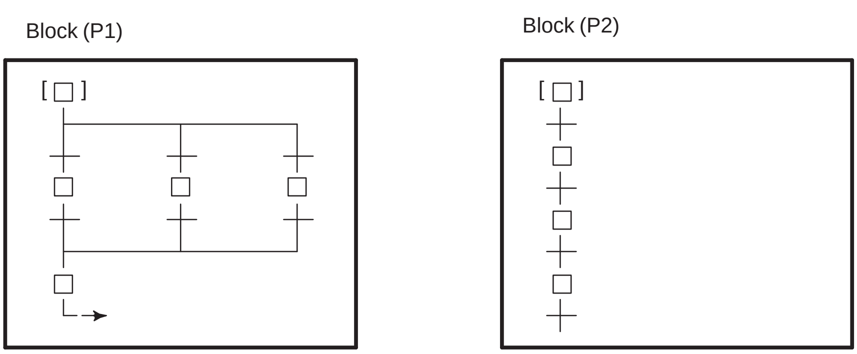


Fig. 2.1 (i) Block

(9) Calling block

To execute a block as the main program in a step sequence, call the block with the CALLU (SUB 66) or CALL (SUB65) instruction in the same way as for ladder subprogram calling from the second level ladder program.

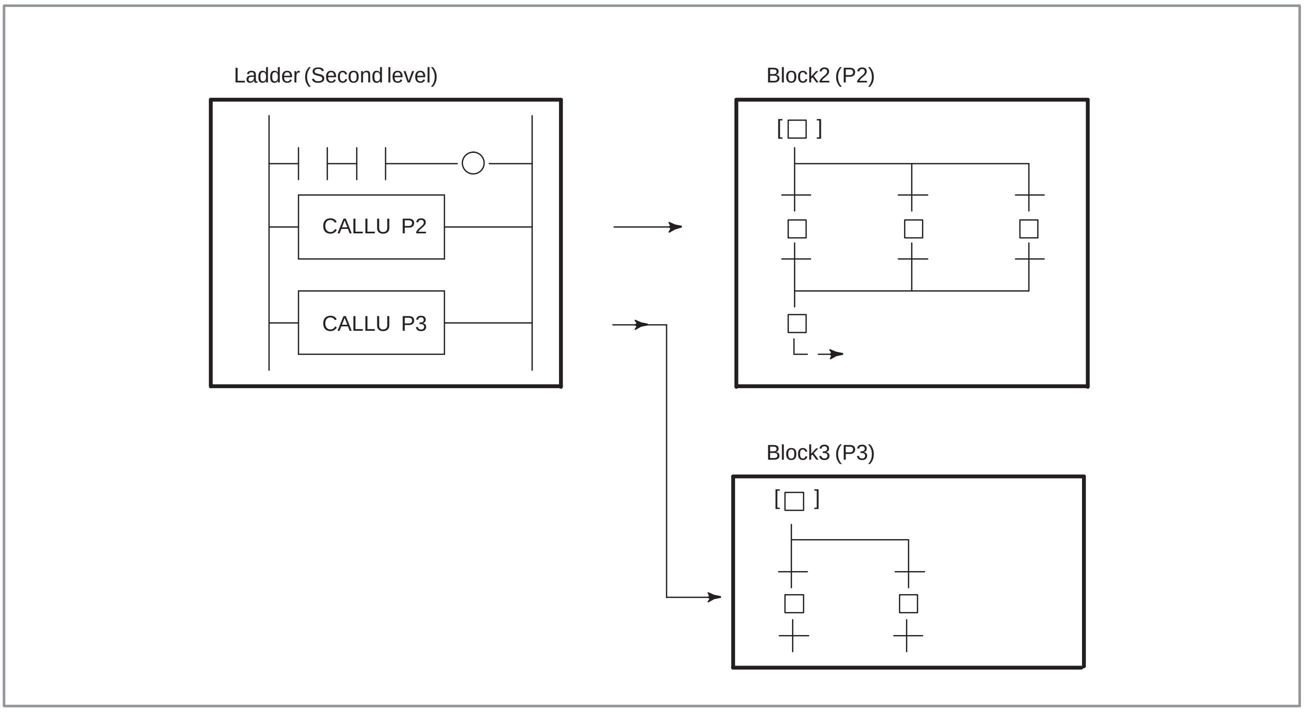


Fig. 2.1 (j) Calling block

(10) Block step (calling step sequence program)

To call a block from the step sequence program as a subprogram, specify a block step in the step sequence program which calls the block, as shown in Fig. 2.1 (k). This is called bloc nesting.

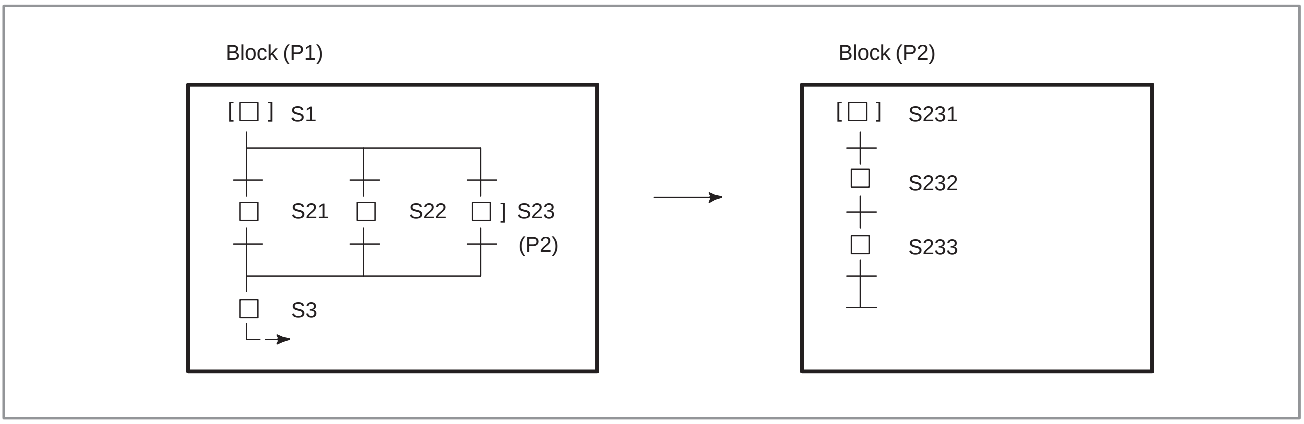


Fig. 2.1 (k) Block nesting

The program shown in Fig. 2.1 (k) is equivalent to in Fig. 2.1 (l). which does not use a block step.

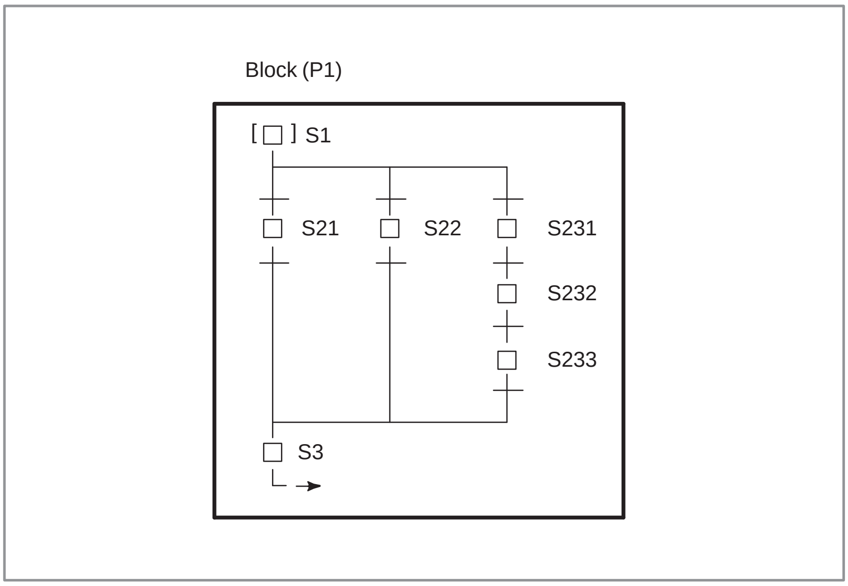


Fig. 2.1 (l) Program without block step

(11)End of block step

Use an end block step to terminate nested–block–step calling and to return to the calling sequence.

2.2 EXECUTION OF STEP SEQUENCE

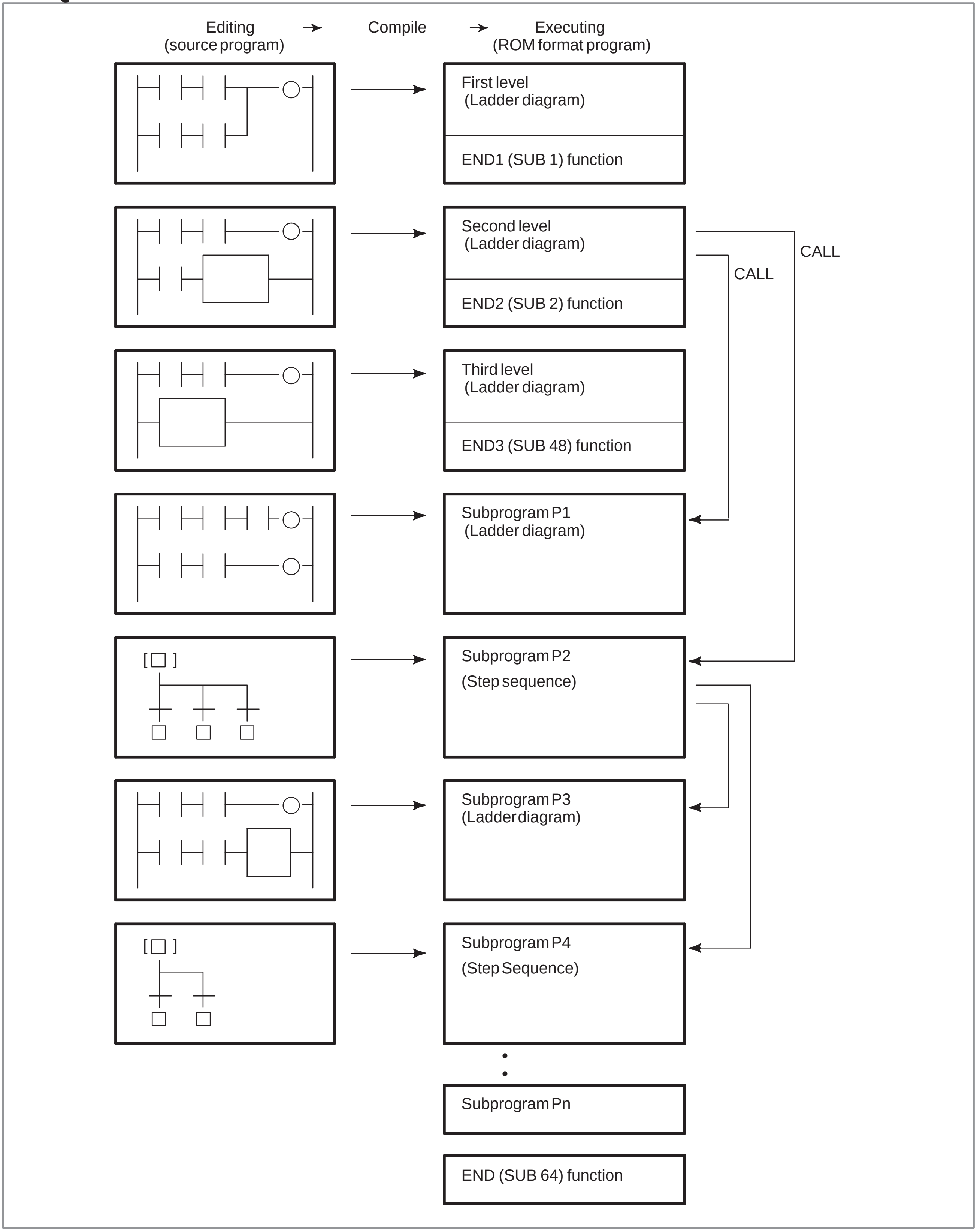


Fig. 2.2 (a) Structure of program

In the step sequence method, a program is created (edited) in units of subprograms. The edited source program is compiled andconverted to an executable ROM–format program, thenlinked, as shown in Fig. 2.2 (a). A ROM–format program is a kind of a modular program, created using conventional subprograms. A step sequence block is also a type of a subprogram. Step sequence blocks are linked to the end of the first level to third level ladder programs, together with other ladder subprograms.

In the same way as in the ladder method, a program is activated at certain intervals, namely every 8 ms, as shown in Fig. 2.2 (b). The first level and second level ladders are executed for a certain period (T ms), then the third level ladder is executed for the remaining time. The period in which the first level and second level ladders are executed varies with the PMC model and the setting of the system parameter (LADDER EXEC). Whether the third level ladder can be used depends on the PMC model.

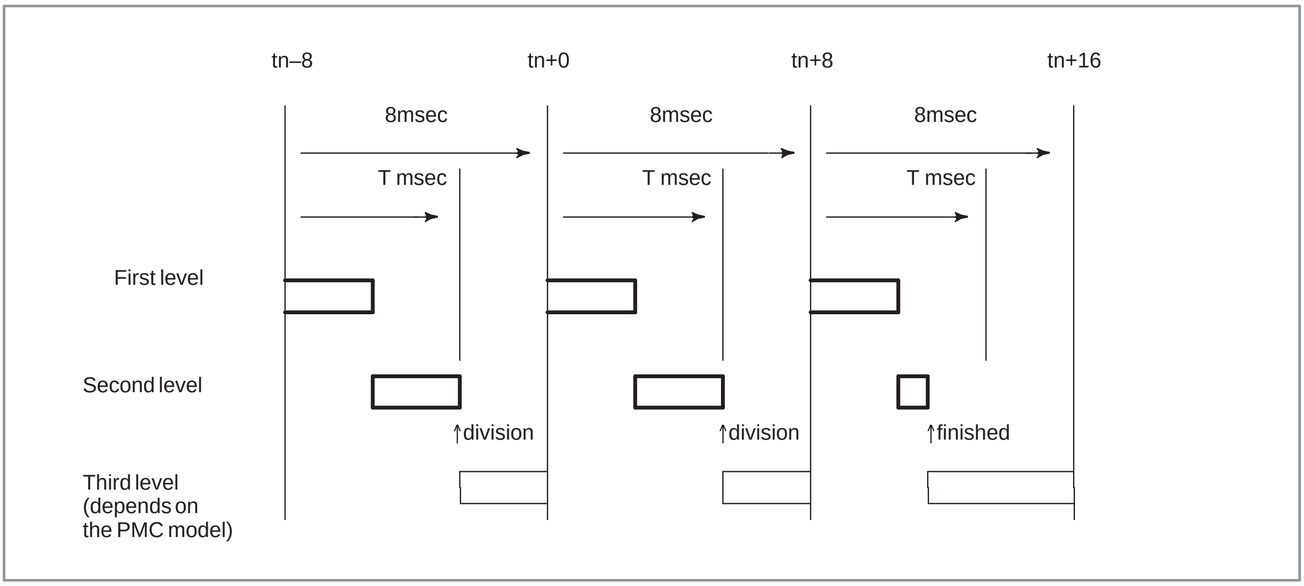


Fig. 2.2 (b) Execution of program cyclically

After the first level ladder has been executed, the second level ladder i executed for the remaining time. If the second level ladder cannot be fully executed within one execution period, it is suspended part–way, with the remainder being executed in the nextperiod. This type of execution is called divided execution. Where the second level ladder is divided varies with the execution time of the first level ladder and that of the executed instructions of the second level ladder. Divided execution is divided into two types, divided system and undivided system. In the divided system, the position where the second level ladder is divided is determined in advance, a divided instruction code being inserted at that position. In the undivided system, in contrast, where the second level ladder is divided is not determined in advance, the ladder being automatically divided upon the determined period elapsing. A PMC which allows step sequence programming executes the second level ladder in undivided system.

In divided execution, the second level ladder is executed at an interval that is a multiple of 8 ms (e. g., 8, 16, 24 ms). Input signals referenced in the second level ladder, such as addresses X and F, are refreshed in synchronization with the execution period for the second level ladder, so that they do not change during the execution.

All subprograms, created using either the ladder or step sequence method, are called from the second level ladder. Hence, the execution time of the second level ladder includes those of ladder subprograms, step sequence programs (blocks), steps, and transitions. Since only the activated step and the transition which checks the transition condition from the step to the next step are executed in a step sequence program, the second level ladder is executed much more frequently than may be expected from the total number of steps.

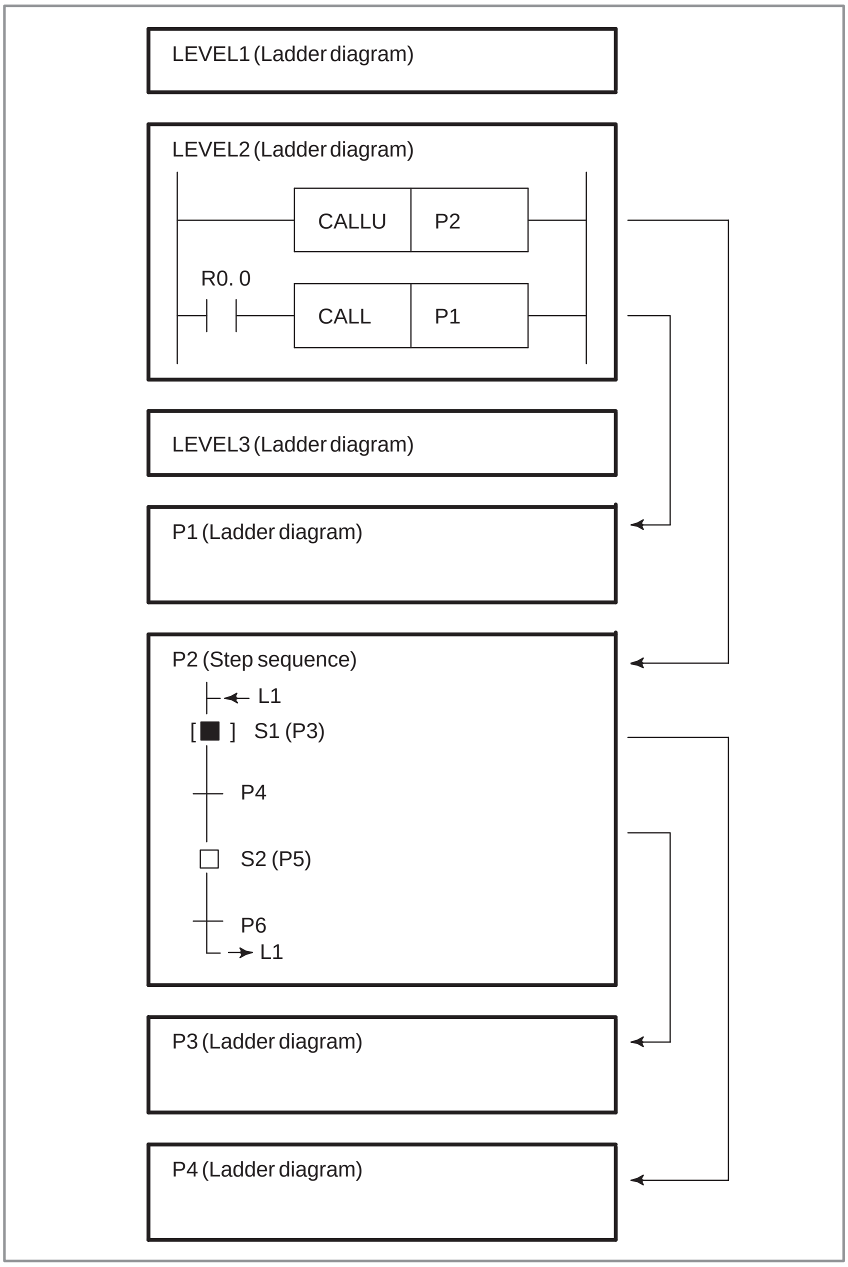


Fig. 2.2 (c) Execution of step sequence

In the step sequence program shown in Fig. 2.2 (c), when step S1 is activated, subprograms are executed according to the timing illustrated in Fig. 2.2 (d).

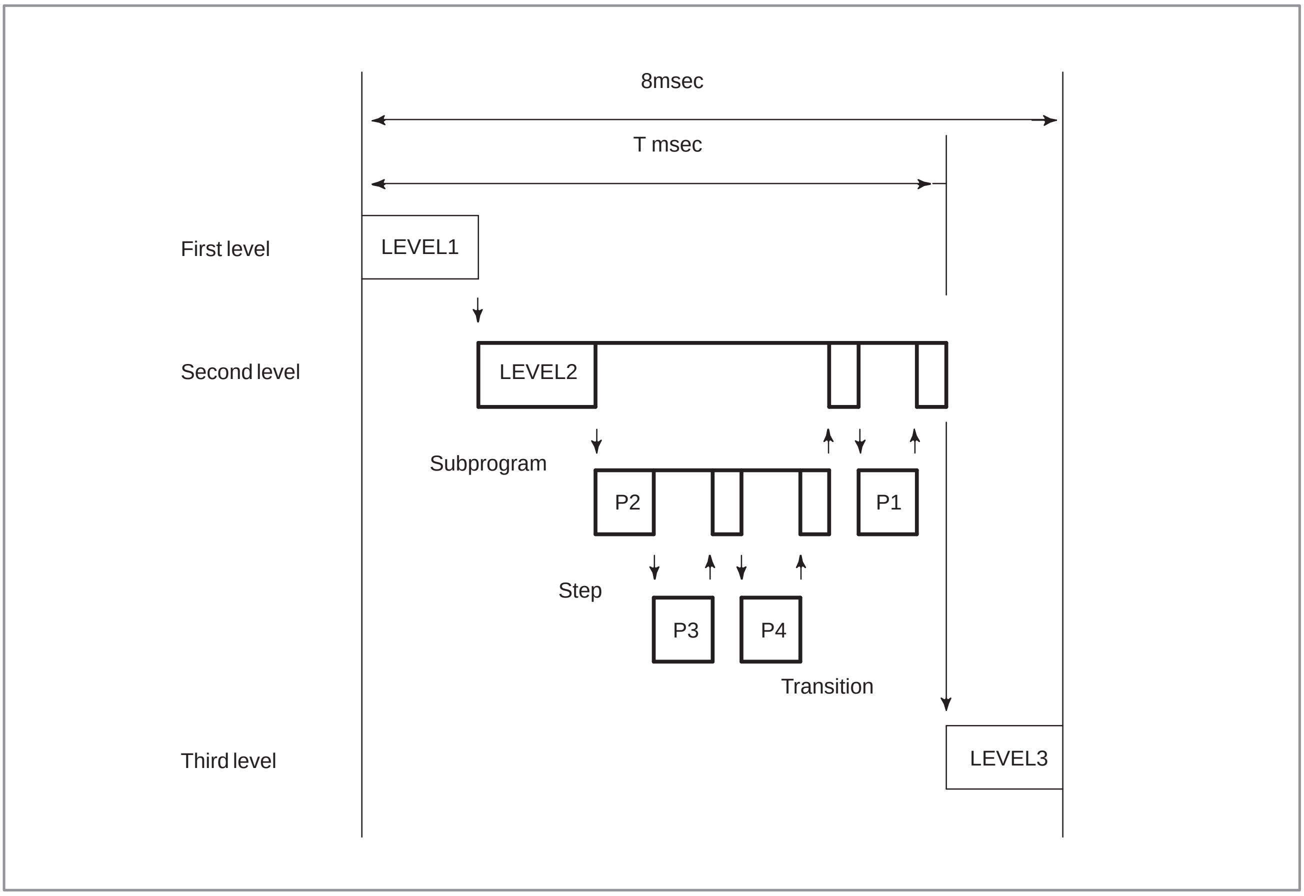


Fig. 2.2 (d) Timing of execution of step sequenceprogram

In this case, step sequence program P2, step P3, transition P4, and ladder subprogram P1 are executed. Step P5 and transition P6 are not executed.

3 CONFIGURATION AND OPERATION OF STEP–SEQUENCE PROGRAMS

3.1 STEP

A step is a unit of processing in a program.

[Display]

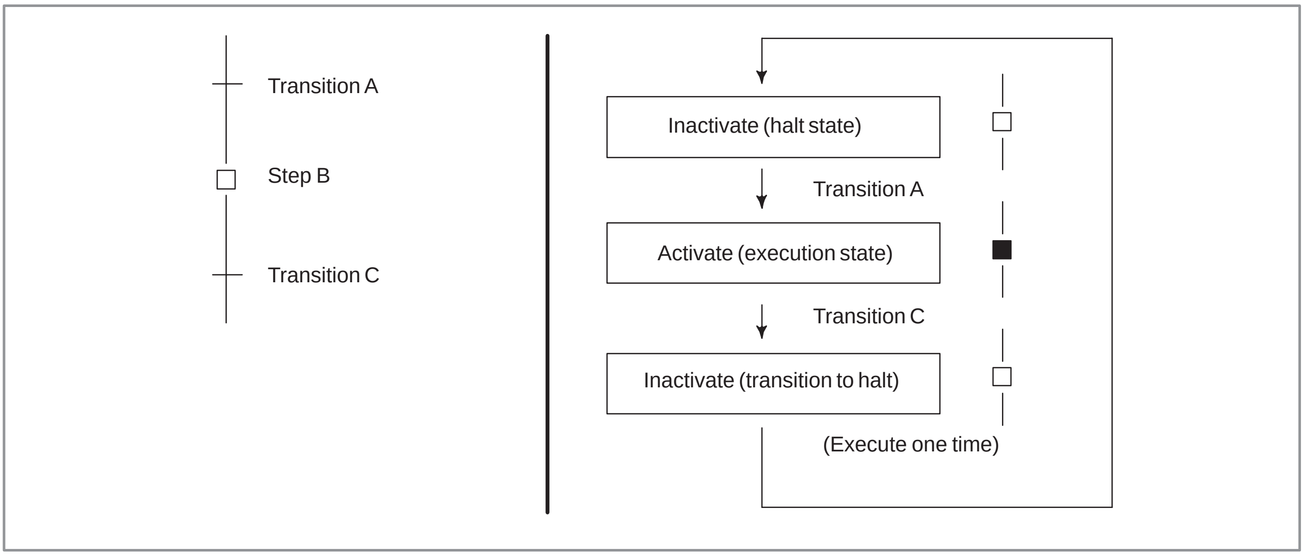
[Contents]

- Define a step number (Sn), necessary for controlling execution, and subprogram number (Pm) specifying actua processing, for a step.
- Assign a step number to a step.
- The same step number cannot be used twice in a program.
- A step has three logical states: the execution, transition to halt, and halt states. The execution state is also called the active state. The transition to halt and halt states are collectively called the inactive state.

State		Contents of operation	Display	Sn.0 NOTE)
Activate	Execution	Activated step. The action program (subprogram) is being executed.	■ Sn	1
Inactivate	Transition to halt	Transition from execution to halt. The action program (subprogram) is executed once only, then the step automatically transits to halt.	□ Sn	0
	Stop	Not activated state. The action program (subprogram) has not yet been executed.	□ Sn	0

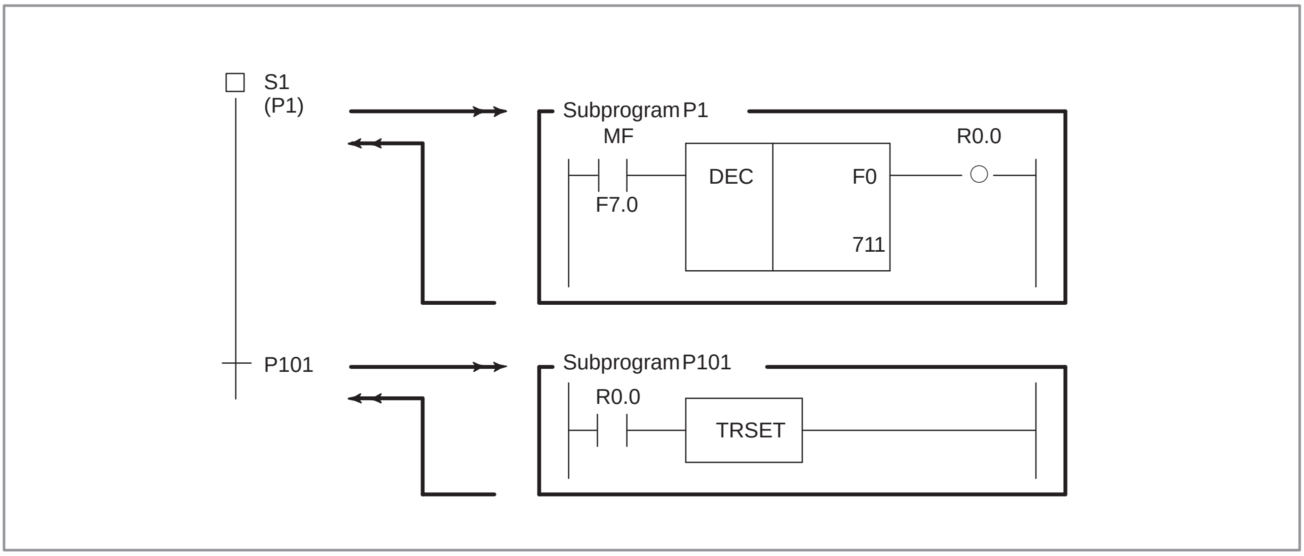
NOTE
Refer to 4.2 PMC ADDRESS (S ADDRESS)

Example) State transition of Step B



[Example]

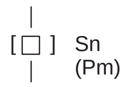
After the M7 code is decoded, control is transferred to the next step using a DEC functional instruction.



3.2 INITIAL STEP

An initial step is automatically activated when execution of the program starts. Once it has been activated, it operates in the same way as a normal step. The program can be returned to this step through other steps.

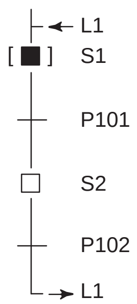
[Display]



[Contents]

- Define a step number (Sn), necessary for controlling execution, and subprogram number (Pm) specifying the actual processing, for an initial step.
- All initial steps are activated when the other steps are not activated.
- Each block must contain at least one initial step. No limit is applied to the number of initial steps contained in a block.
- A block having no initial step cannot be executed if called.
- Assign a step number to an initial step.
- The same step number cannot be used more than once in a program.
- In parallel branch, one initial step is required for each path. (See example 2.)

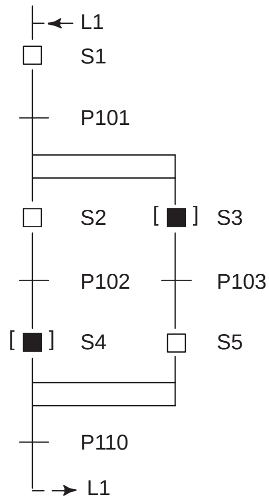
[Example1]



When a program is executed, step P1, specified by an initial step, is activated first.

Initial step S1 is executed in the same way as normal step once S1 has been executed.

Example2



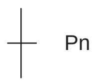
When a program is executed, steps S3 and S4, specified by an initial step, are activated first.

Once steps S3 and S4 have been executed, initial steps S3 and S4 are executed in thesame way as normal step when the program starts from step S1.

3.3 TRANSITION

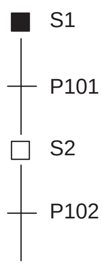
A transition specifies the conditions governing the transition from the step to the next step.

[Display]



[Contents]

- Only one transition is required between steps.
- Transition between steps is performed as described below.



While S1 is activate, only S1 and P101 are executed.
Other steps and transition are not executed.
When the transition P102 evaluates true unles S2 is not being executed, the state is ignored.

When the transition P101 evaluates true, control passes fro S1 to S2. In this case, when the condition i true, S1 is terminated regardless of the state of S1, and S2 is activated.

- When a signal is set to 1 in a transition, it remains the state even if the control is transferred to the subsequentstep. To set the signal to 0, use another subprogram to do so.

[Example]

Refer an example described on the Step function (3.1).

3.4 DIVERGENCE OF SELECTIVE SEQUENCE

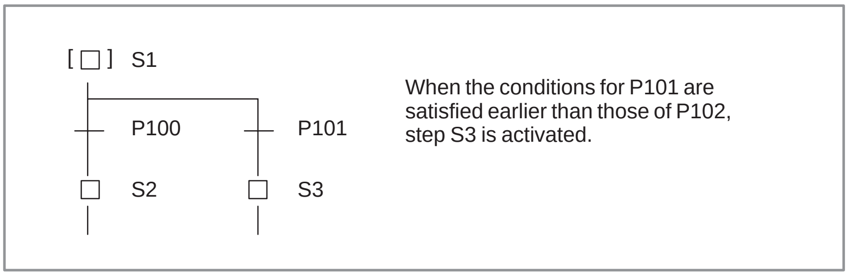
A selective sequence branches to two or more sequences. When the transition evaluates true, the corresponding step is activated.

[Display]

[Contents]

- Transitions are placed after a divergence of selective sequence.
- The step connected to the transition for which the conditions are true is first activated.
- When the conditions for any transition are true simultaneously, the leftmost step is activated.
- A selective sequence can create up to 16 paths.

[Example]



3.5 CONVERGENCE OF SELECTIVE SEQUENCE

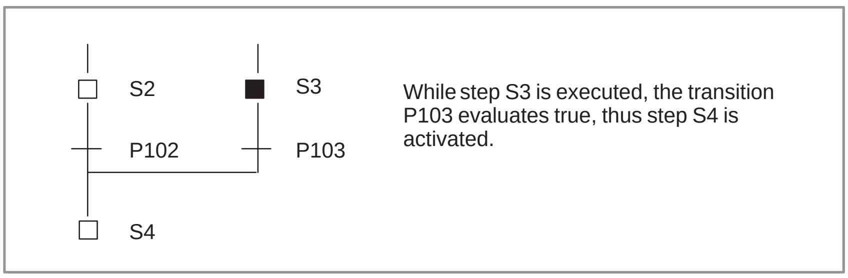
It combines two or more divergent paths to the main sequence.

[Display]

[Contents]

The number of divergent paths must match that of the convergent paths.

[Example]



3.6 DIVERGENCE OF SIMULTANEOUS SEQUENCE

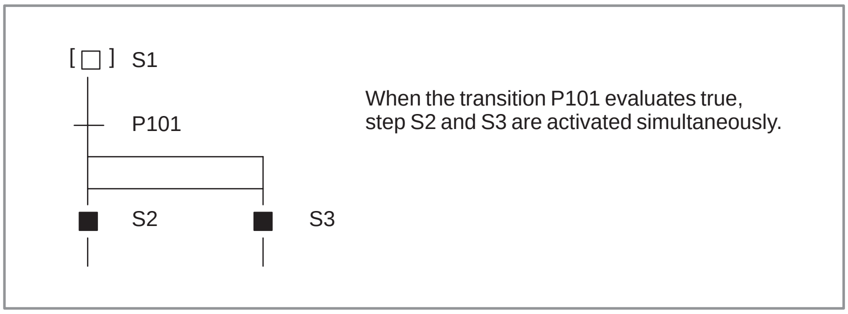
A simultaneous sequence branches to two or more sequences, and all steps are activated simultaneously.

[Display]

[Contents]

- A transition must be placed before a divergence of simultaneous sequence.
- All branched steps are activated simultaneously, then executed.
- A simultaneous sequence can create up to 16 paths.

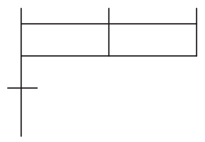
[Example]



3.7 CONVERGENCE OF SIMULTANEOUS SEQUENCE

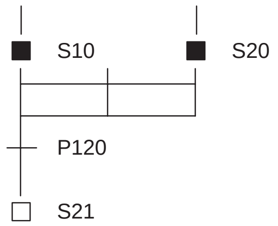
It combines two or more divergent paths to the main sequence.

[Display]



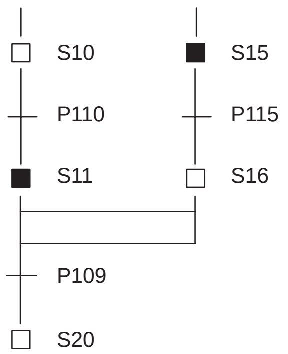
[Contents]

- A convergence of simultaneous sequence is processed as follows.



When the transition P120 evaluates true, step S10 and S20 are terminated and step S21 is activated.

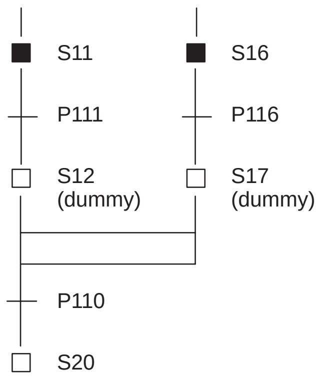
- Wait processing is processed as follows.
  case1)



When the transition P109 evaluates true unless both of step S11 and S16 are active, control does not pass to step S20.

When the transition P109 evaluates true while both of S11 and S16 are active, S11 and S16 are terminated and S20 is activated. In the case, P109 provides the termination conditions for both S11 and S16.

case2)



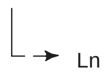
To specify the termination conditions for S11 and S16 separately, place the conditions in P111 and P116 and specify two dummy steps, S12 and S17, as shown

A dummy step also requires a step number and subprogram number. Alsospecify a dummy transition condition, which becomes always true, in P110.

3.8 JUMP

A jump controls the execution of steps non–sequentially, together with a transition.

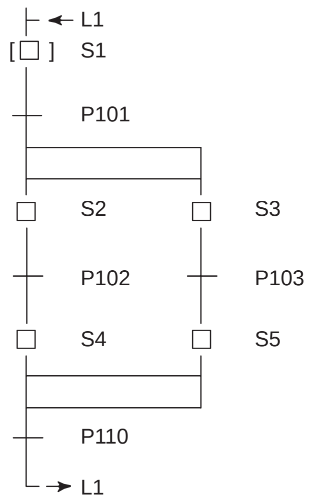
[Display]



[Contents]

- Specify a jump destination label (Ln).
- The step to which control is transferred (jumped) is activated.
- The jump destination must be within the same program.
- A jump cannot be performed from outside a simultaneous sequence to within the simultaneous sequence, or from within a simultaneous sequence to outside.
- A jump cannot be performed between parallel–branched paths.

[Example]



When steps S4 and S5 ar executed and the transition P110 evaluates true, the program is repeated from initial step S1.

3.9 LABEL

A label specifies the jump destination.

[Display]

[Contents]

Specify the jump destination label (Ln).

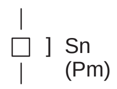
[Example]

Refer to an example described on the jump function (3.8).

3.10 BLOCK STEP

A block step specifies the step sequence subprogram to be executed.

[Display]

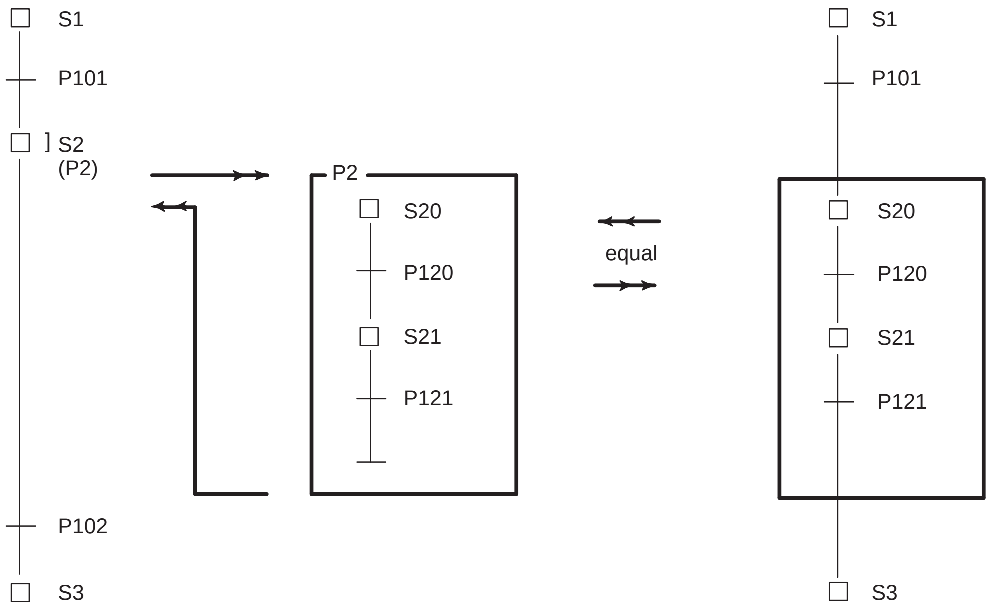


[Contents]

Define a step number (Sn), which controls the execution of a bloc step, and a sub–program (Pm) specifying the actual process, for a block step.

NOTE

- Assign a step number to a block step.
- The same step number cannot be used twice in a program.
- A transition must be placed after a block step.
  Example)



- Transition P102 cannot be omitted due to the syntax of the step sequence method. Specify a dummy transition, which becomes always true, for transition P102.
- Transition P121 must specify the transition condition for the termination of the step S21.
- When the conditions of transitions P102 and P121 are switched, step S21 will not be correctly executed.

3.11 INITIAL BLOCK STEP

This is an initial step on the block step.

[Display]

[Contents]

- Define a step number (Sn), necessary for controlling execution, and subprogram number (Pm)specifying the actual processing, for an initial step.
- This step has the same function and graphical symbol asan initial step.

3.12 END OF BLOCK STEP

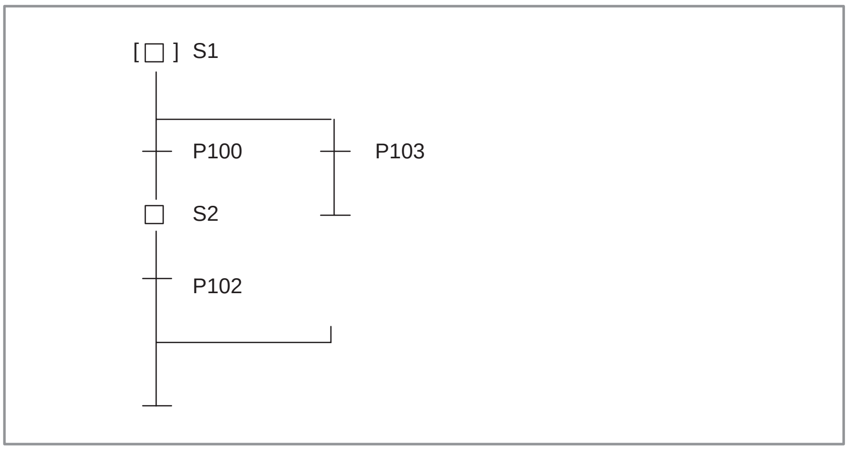
This terminates a block step.

[Display]

[Contents]

- Use this step to terminate a block step.
- Each block requires at least one end block step. No limit is applied to the number of end block steps.

[Example]



4 EXTENDED LADDER INSTRUCTIONS

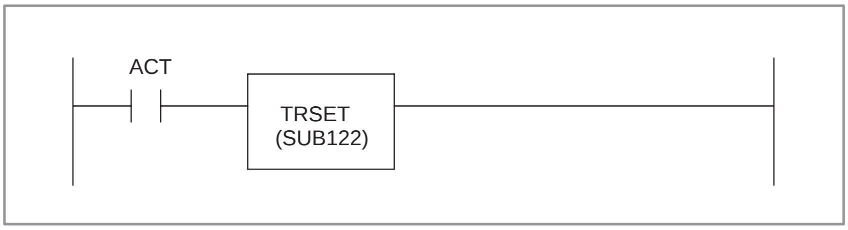
To enable the specification of steps and transitions, the components of a step sequence program, by means of the ladder method, the following signals and functional instructions are provided. These signals and instructions can only be used in subprograms in which step sequence step and transitions are specified.

4.1 FUNCTIONAL INSTRUCTION TRSET

[Function]

- This instruction describes that the conditions for a transition have been true.
- This instruction is used in a subprogram which is called from a transition.

[Format]



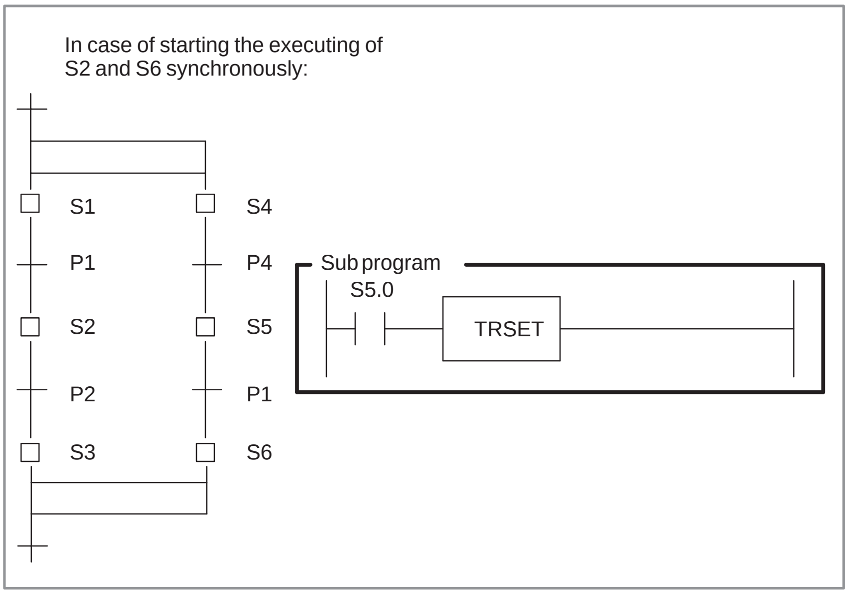
4.2 PMC ADDRESS (S ADDRESS)

[Contens]

- This address is used to read the logical state of a specified step.
  0 : Transition to halt state, or halt state
  1 : Execution state
- This address is used for creating a program in which detailed transitions of the execution states between steps are considered. Specify the number of the step to be read.
  Example) To reference the state of the step S100
  S100. 0
- This address allows any subprogram to reference the state of any step.
- Data cannot be written into state signal Sn. 0.
- A ladder can be configured for the TRSET transition instruction using state signal Sn. 0. Referencing state signal Sn. 0, however, adversely affects the portability and comprehensibility. Use this feature sparingly.

[Example]

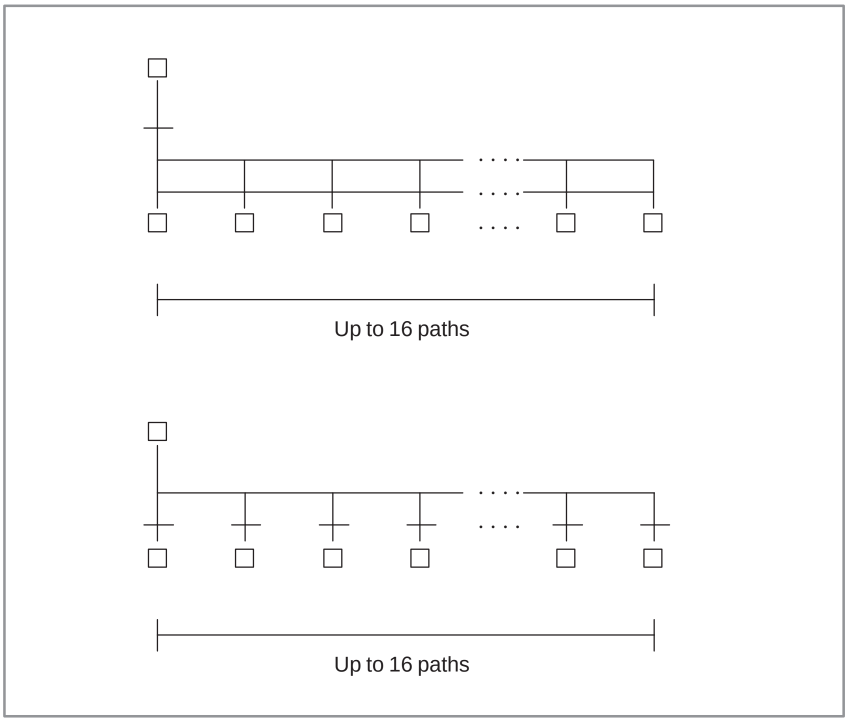
This address is used to reference the activation states of steps in a step in which this address has been specified, and performs complicated wait processing in a program including a simultaneous sequence.



5 SPECIFICATION OF STEP SEQUENCE

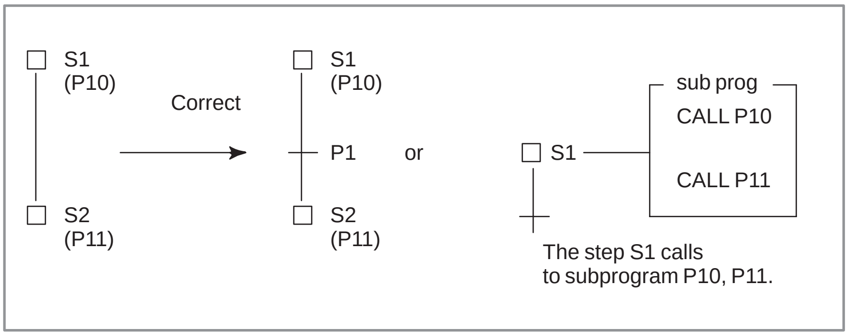
5.1 SPECIFICATION

Contents/Kind of PMC	PMC–RB4/RB6/RC4/NB2
Number of subprogram	Up to 2000 (P1 to P2000)
Number of step	Up to 1000 (S1 to S1000)
Number of label	Up to 9999 (L1 to L9999)
Number of jump in block	Up to 256
Nesting depth of block step	Up to 8 levels
Size of block	64 lines × 32 columns
Number of paths	Up to 16 paths

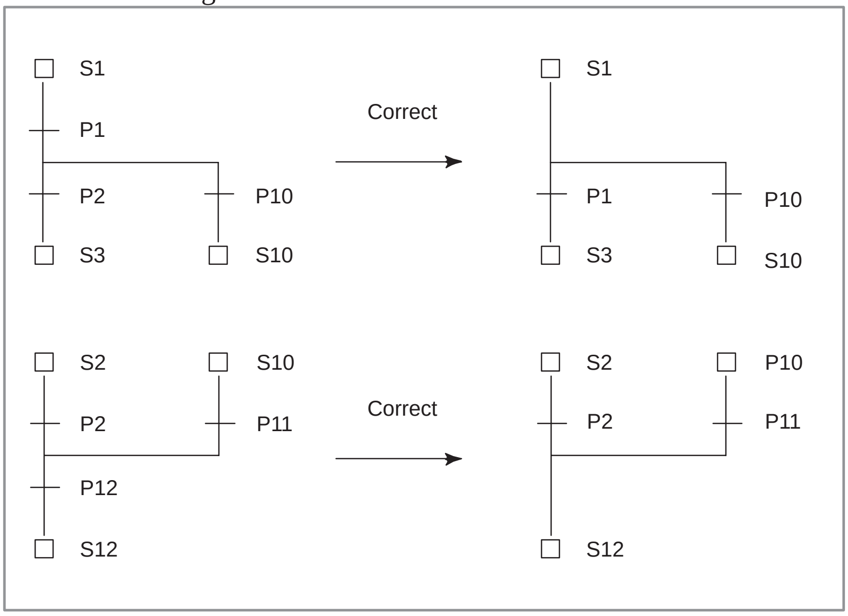


5.2 GENERAL RULES

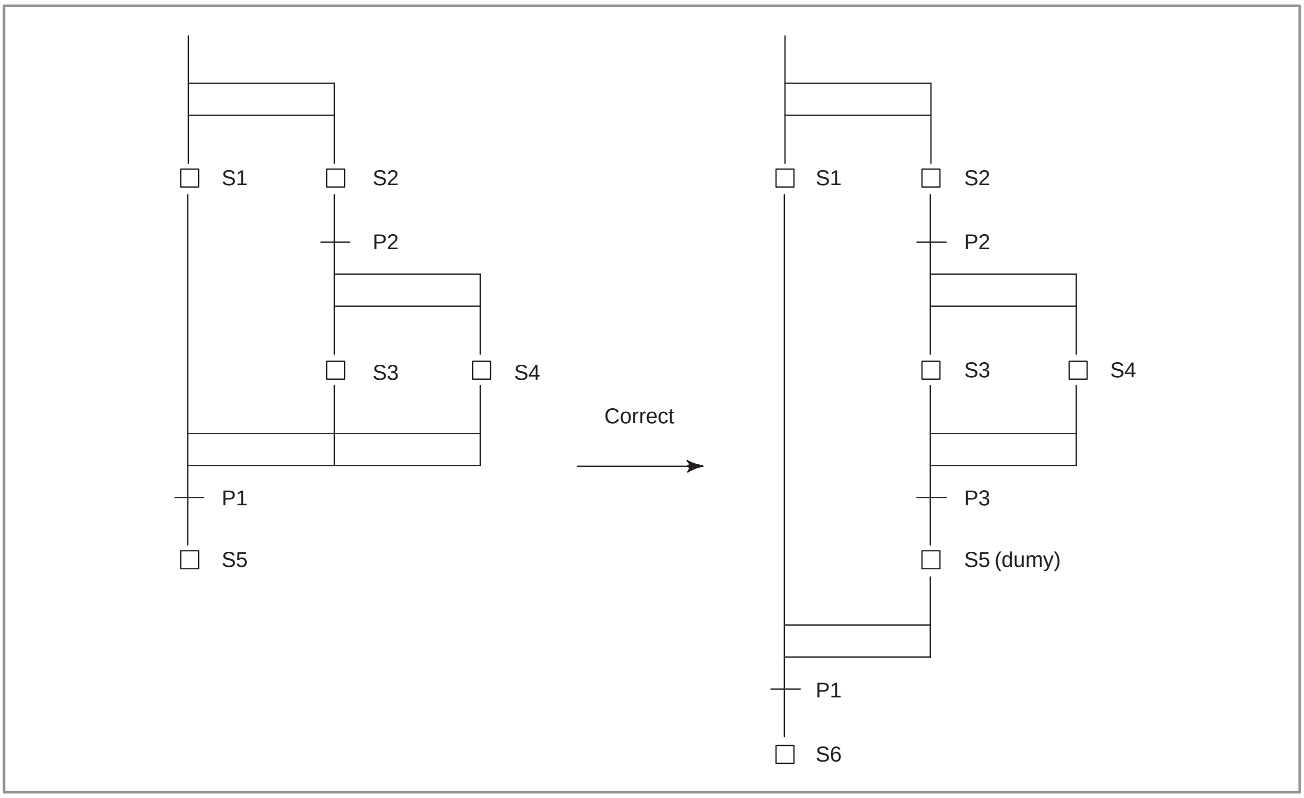
- One transition must exist between step and step.



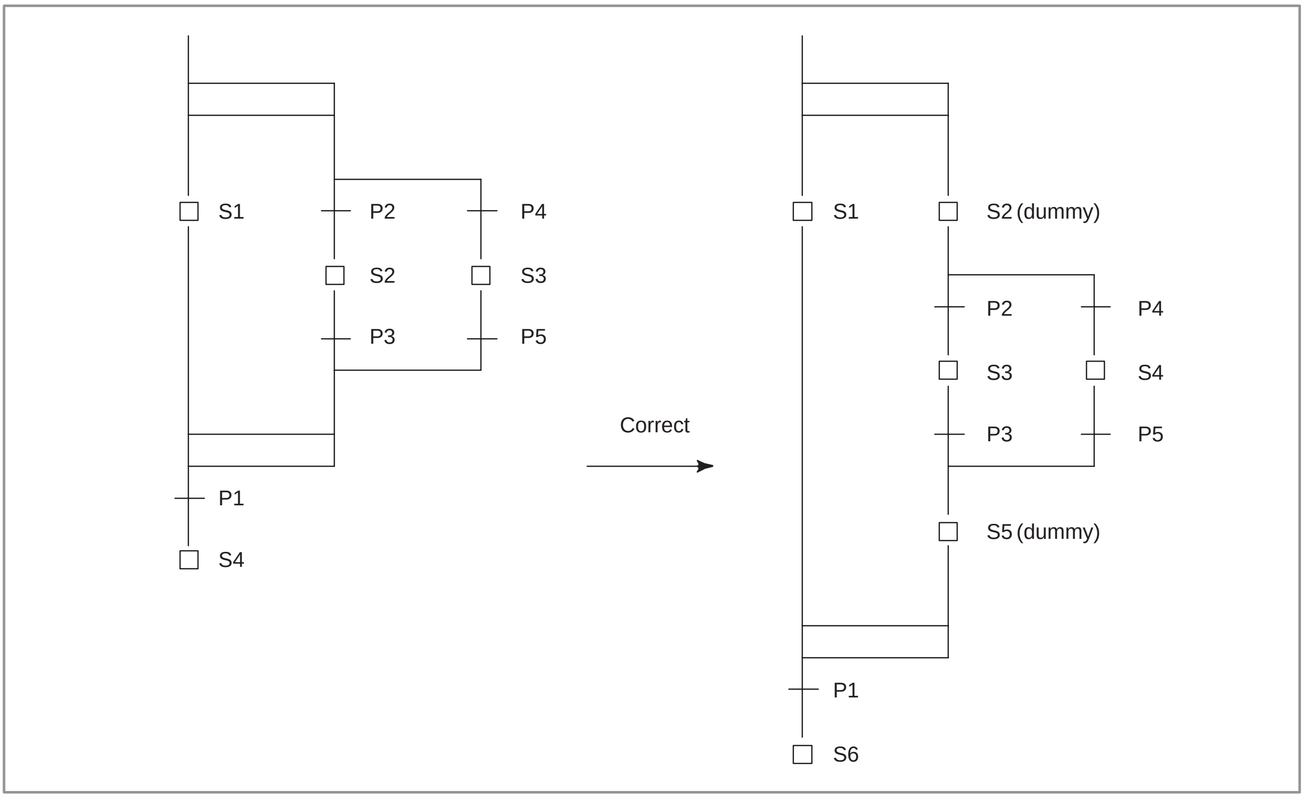
- The transition shall never be repeated even at the point of the divergence and the convergence.



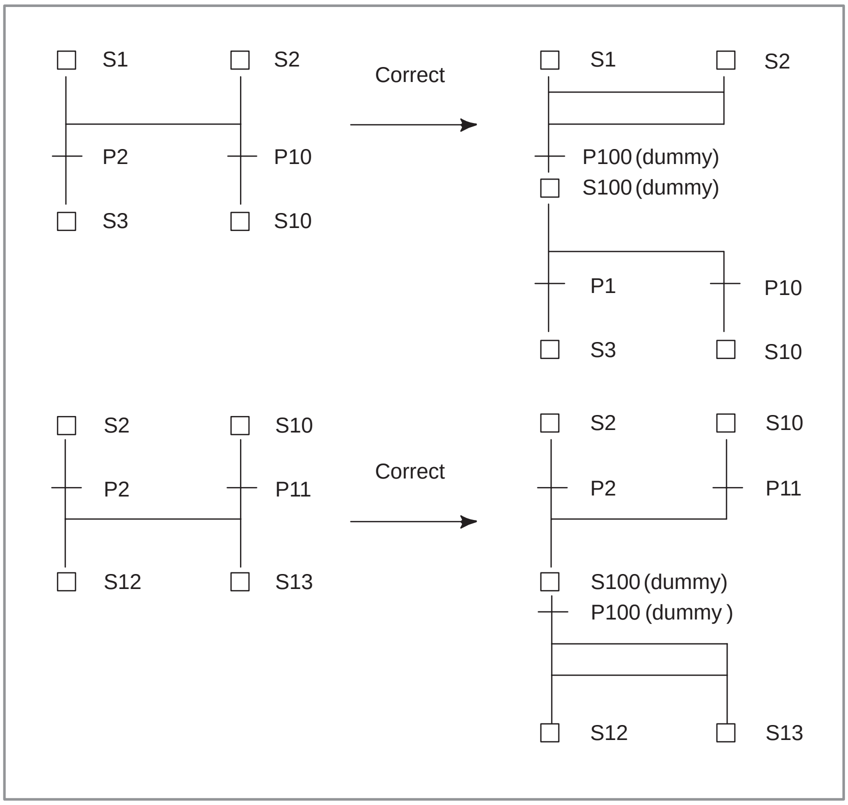
- When a simultaneous sequence is specified in another simultaneous sequence, one convergence must not be used for each sequence.



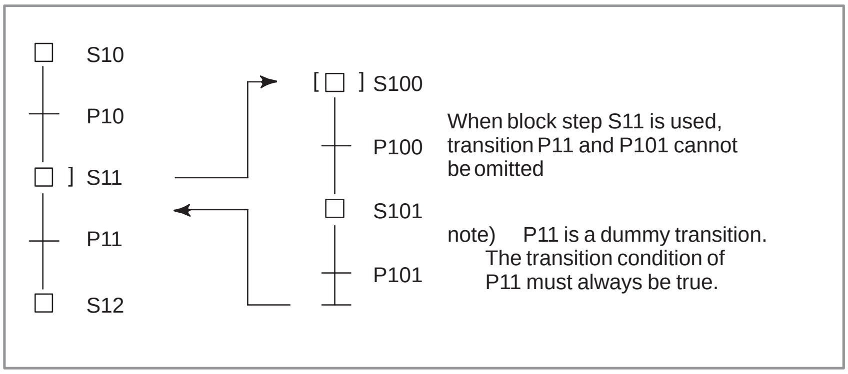
- When a selective sequence is specified in a simultaneous sequence, dummy steps must be required both after the divergence and before convergence.



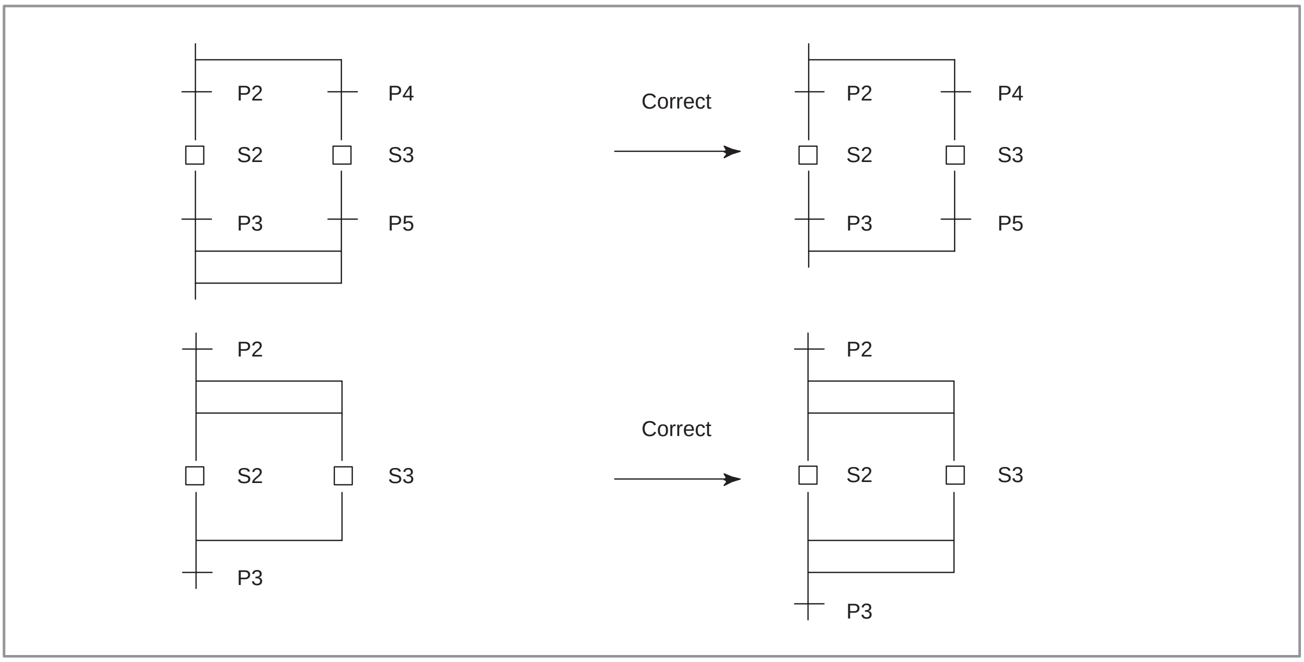
- In case of branching again immediately after the convergence, a step/transition is required between the divergence and convergence.



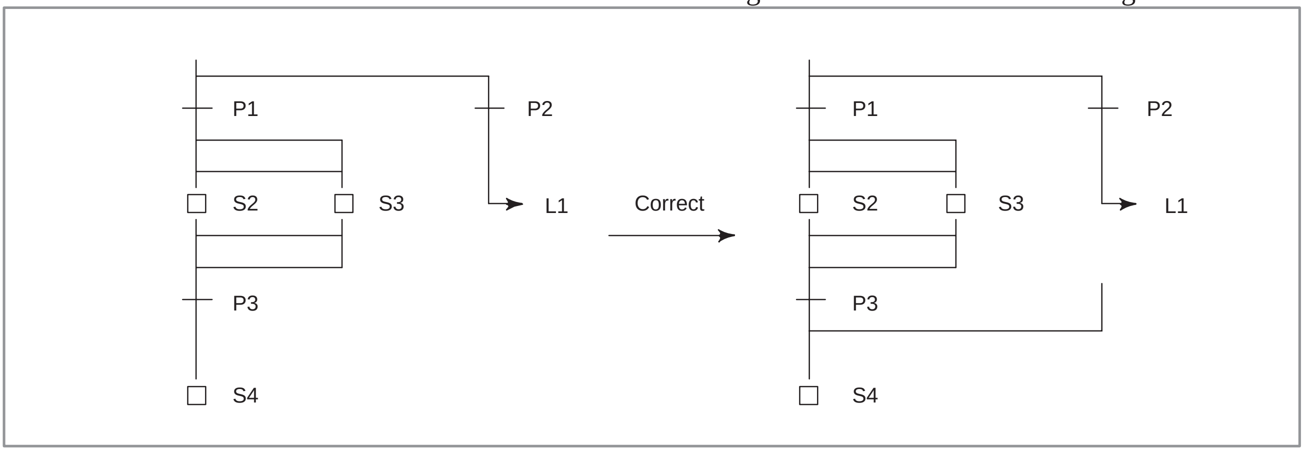
- Immediately after the block step, a dummy transition which is always true is needed.



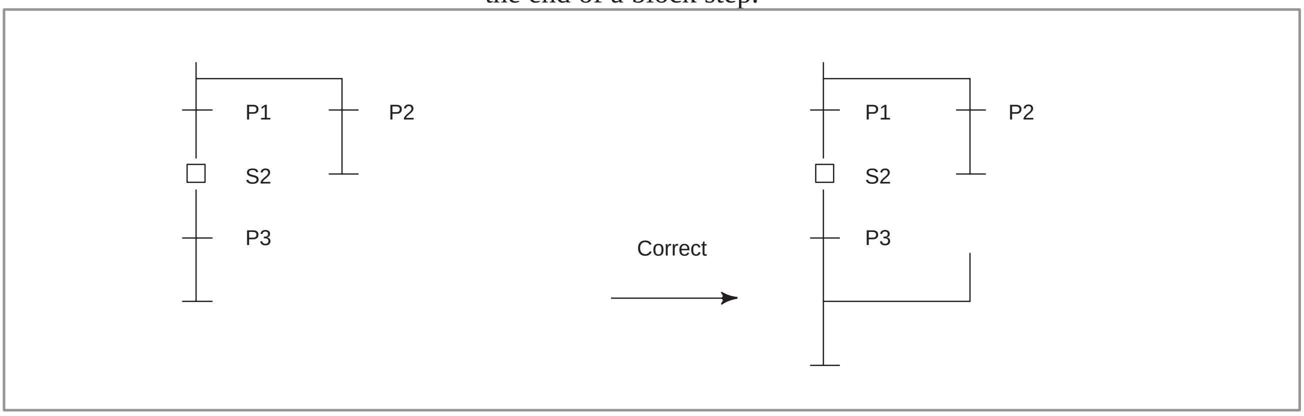
- The divergence must be terminated with the same type of convergence.



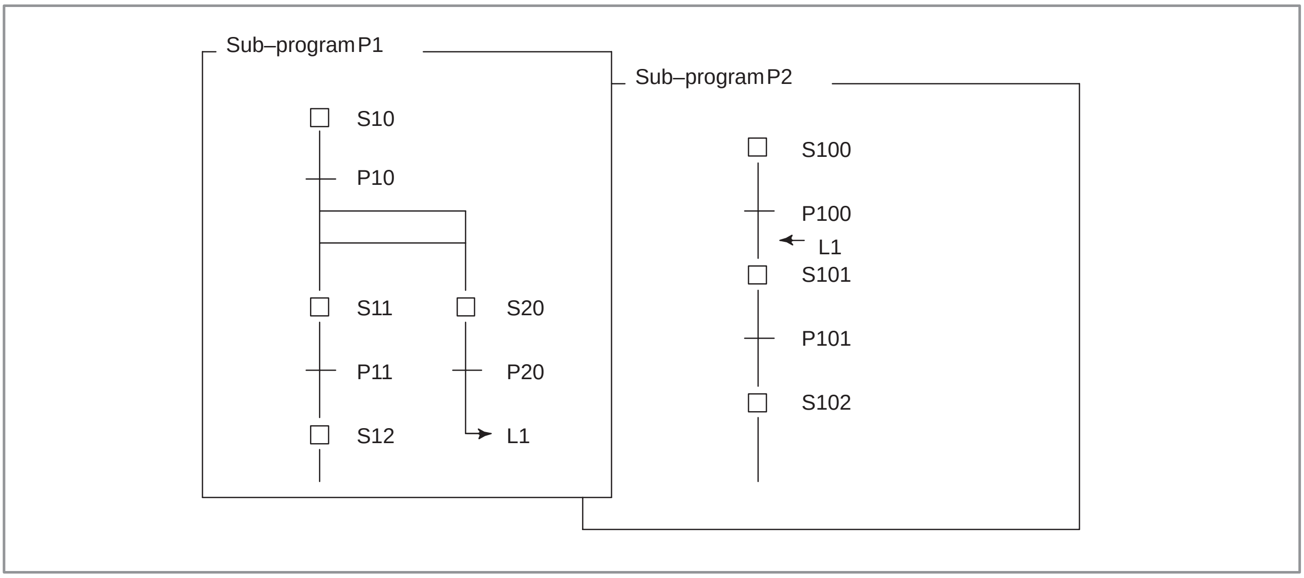
- The number of convergences must match that of divergences.



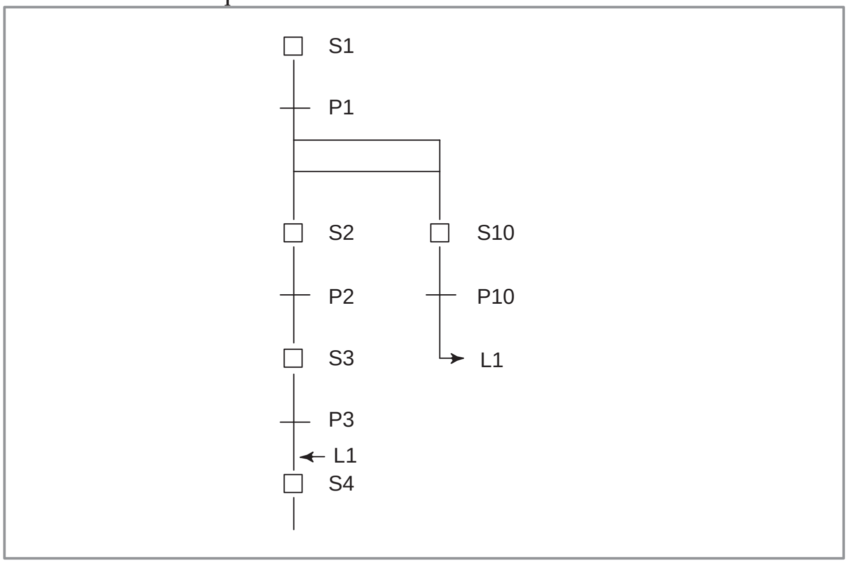
- The number of convergences must match that of divergences, even at the end of a block step.



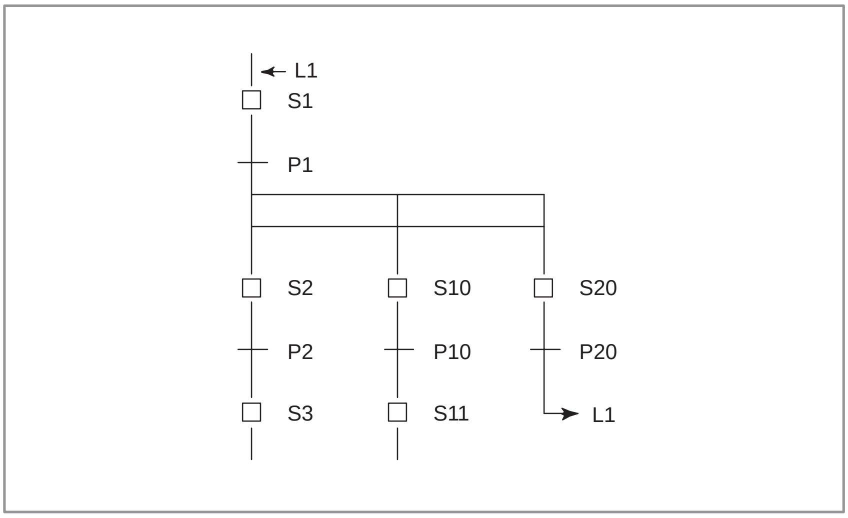
- It is not possible to jump to the other subprogram.



- It is not possible to jump from a simultaneous sequence to another simultaneous sequence.



- It is not allowed to jump from inside of the simultaneous sequence to outside.



5.3 EXCLUSIVE CONTROL FOR FUNCTIONAL INSTRUCTIONS

The use of the following functional instructions is restricted in steps and transitions.

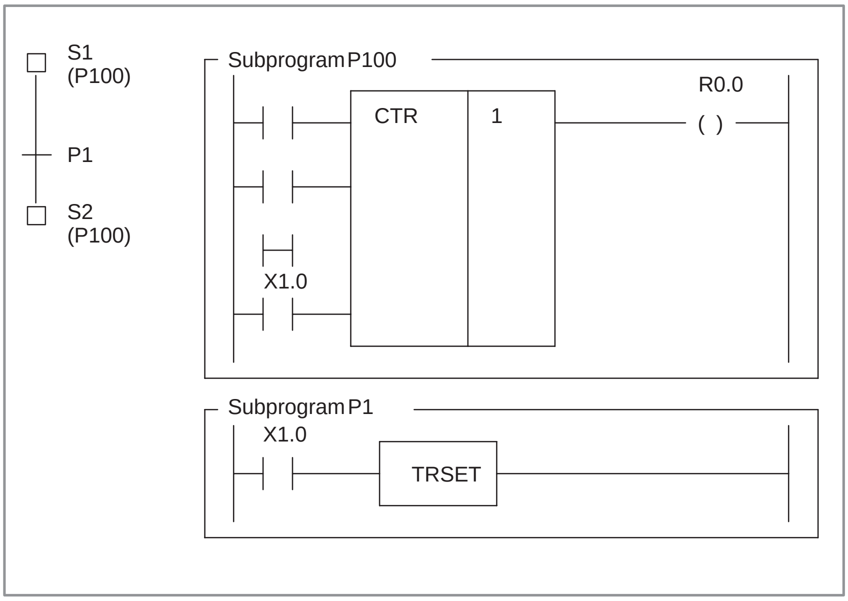
Group	Description		Functional instructions
A	The instructions operate when a signal changes		CTR (SUB5) CTRC (SUB60) TMR (SUB3) TMRB (SUB24) TMRC (SUB54) DIFU (SUB57) DIFD (SUB58)
	Con–dition	Multiple functional instructions having the same number are used.	
	Prob–lem	Not activated. Correct operation cannot be guaranteed.	
B	Restriction due to the interface.		WINDR (SUB51) WINDW (SUB52) DISP (SUB49) DISPB (SUB41) EXIN (SUB40)
	Con–dition	Data is input or output by using two subprograms.	
	Prob–lem	Invalid return value. Not terminated.	

(1) Functional instructions of group A

Since these functional instructions operate when the corresponding signals change, they may not operate correctly when called from multiplesteps.

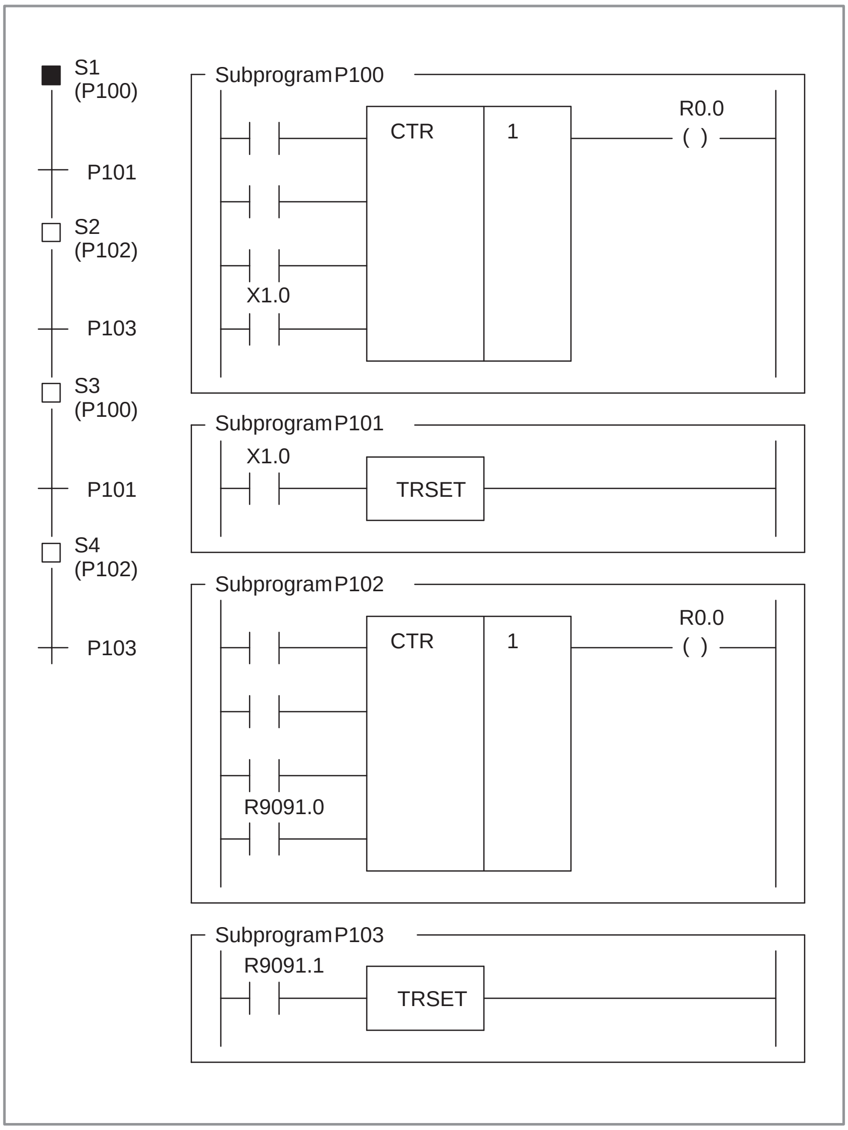
Example)

While multiple CTR functional instructions are used, when control passes from S1 to S2 with ACT of CTR not set to off, CTR is not counted when called from step S2.



Correct program

Divide the subprogram so that ACT of CTR is called after it is set to off.



(2) Functional instructions of group B

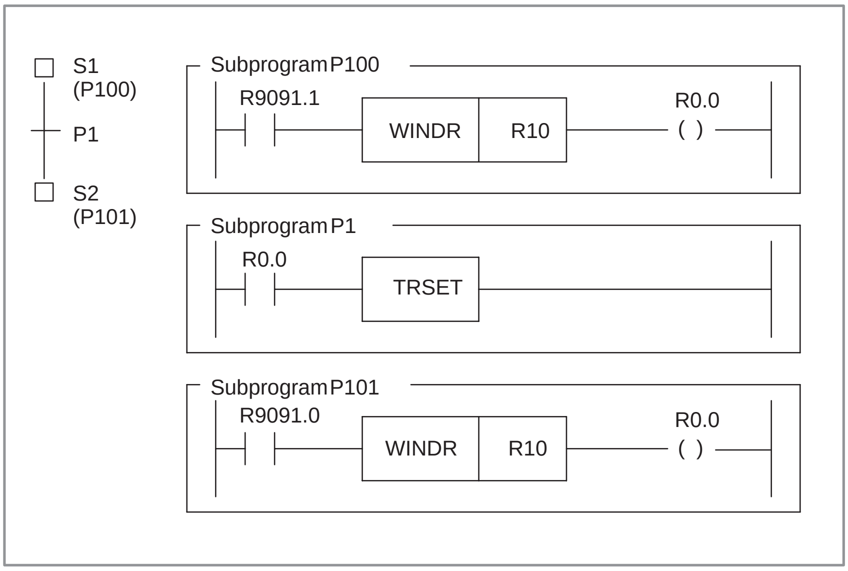
While an instruction is being executed through the interface with the NC, other same instructions cannot be executed. PMC control software does not receive the process when the instruction is not at a same position (net).

If ACT is set to on and off in different instructions (or subprograms), these processes are not terminated.

NOTE

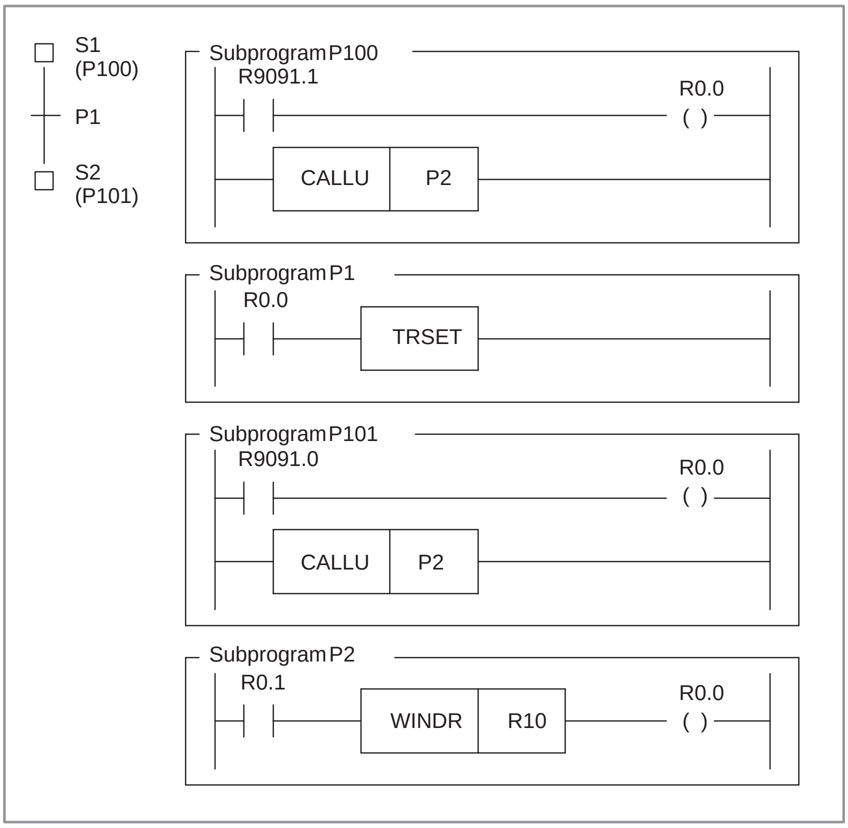
In the window instructions (WINDR and WINDW), low–speed–type is included the functional instructions of group B.

Example)



Correct program

Correct the program so that ACT is set to on and off within one subprogram.



6 CRT/MDI OPERATION

The following operations are supported to enable the diagnosis and debugging of a step sequence program.

(1) Displaying the sequence diagram

(2) Displaying the run time of the step sequence program

(3) Monitoring the run time of the step sequence program

6.1 DISPLAYING OF SEQUENCE PROGRAM

The diagnosis and debugging of a step sequence program have four screens.

(1) Program configuration list (main screen)

(2) Step sequence screen

(3) List screen

(4) Ladder screen

6.1.1 Program Configuration List (Main Screen)

Press the [STPSEQ] key and display the program configuration list.

```
STPSEQ <<MAIN>>   PROGRAM:(STEP SEQUENCE DEMO PROGRAM)   MONIT RUN
                    P0001 (       ) SUB PROGRAM NO.1

  LEVEL1     LEVEL2    LEVEL3
  □ P0001    □ P0002   □ P0004   □ P0005   □ P0006    □ P0007
  □ P0008    □ P0009   □ P0014   □ P0015   □ P0016    □ P0017
  □ P0021    □ P0022   □ P0024   □ P0025   □ P0026    □]P0027
  .      .         .         .        .          .
  .      .         .         .        .          .
  □ P0101    □ P0202   □]P0304   □]P0405   □]P0406    □]P0407

 [    UP    ] [   DOWN   ] [   TIME   ] [  P–ADRS   ] [   ZOOM   ]
```

Items displayed on the screen

Display	Contents	Display by [ZOOM] key
LEVEL1	Ladder first level	Ladder diagram
LEVEL2	Ladder second level	Ladder diagram
LEVEL3	Ladder third level note1)	Ladder diagram
□ Pxxx	Subprogram	Ladder diagram
□] Pxxx	Subprogram	Step sequence diagram

Pxxx indicates a subprogram number.

NOTE
The third level ladder can be omitted.

[ZOOM] key

To display the contents of a program, position the cursor to the program number and press the [ZOOM] key. The step sequence diagram (Fig. 6.1.2 (b)) or ladder diagram (Fig. 6.1.3 (c) is automatically displayed according to the type of the program.

[TIME] key

Press the [TIME] key to display the time display screen (Fig. 6.2.1) and time monitor screen (Fig. 6.3).

[P–ADRS/P–SYMB] key

Displays the addresses specified to subprograms, using addresses or symbols, if symbols have been assigned. When the [P–ADRS] key ispressed, the addresses are displayed. When the [P–SYMB] key is pressed, the symbols are displayed.

6.1.2 Step Sequence Screen

(1) Position the cursor to a program indicated by □], then press the [ZOOM] key.

```
STPSEQ <<MAIN>>    PROGRAM:(STEP SEQUENCE DEMO PROGRAM)    MONIT RUN
                     P0407 (MAIN    ) STEP SEQUENCE NO.1

  LEVEL1     LEVEL2    LEVEL3
  □ P0001    □ P0002   □ P0004   □ P0005   □ P0006    □ P0007
  □ P0008    □ P0009   □ P0014   □ P0015   □ P0016    □ P0017
  □ P0021    □ P0022   □ P0024   □ P0025   □ P0026    □]P0027
  .      .         .         .        .         .
  .      .         .         .        .         .
  □ P0101    □ P0202   □]P0304   □]P0405   □]P0406    □]P0407

 [    UP    ] [   DOWN   ] [   TIME   ] [  P–ADRS   ] [   ZOOM    ]
```

Fig. 6.1.2 (a) Program configuration list (main screen)

Example)

When the cursor is positioned to □] P0407 and press the [ZOOM] key, the subprogram P407 isdisplayed.

(2) Displayed Step Sequence

Activated steps are indicated by red □ (highlighted □ on a monochrome display). (In this manual, activated steps are indicated by ■.)

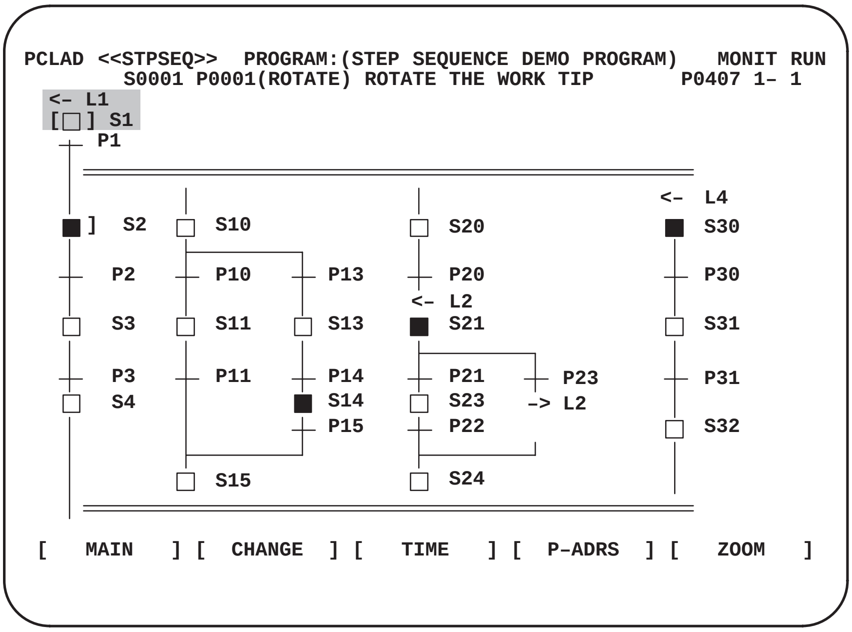


Fig. 6.1.2 (b) Step sequence screen

Meaning of display

Display	Contents	Display by [ZOOM] key
[□] Sxxx	Initial step	Ladder diagram
□ Sxxx	Step	Ladder diagram
□] Sxxx	Block step	Step sequence diagram
+ Pxxx	Transition	Ladder diagram
————	Selective sequence	Cannot zoom.
════	Simultaneous sequence	Cannot zoom.
└→L2	Jump	Cannot zoom.
←–L2	Label	Cannot zoom.

Pxxx means the subprogram number.

[ZOOM] key

To display the contents of a program, position the cursor to the program number and press the [ZOOM] key. The step sequence diagram (Fig.6.1.2 (b)) or ladder diagram (Fig.6.1.3 (c)) is automatically displayed according to the type of the program.

[MAIN] key

Press the [MAIN] key to return to the program configuration list.

[CHANGE] key

Press the [CHANGE] key to list the subprograms referenced in the step sequence program.

[TIME] key

Press the [TIME] key to display the time display screen (Fig. 6.2.1).

[P–ADRS/P–SYMB/S–ADRS/S–SYMB] key

Displays the addresses specified with steps and transitions, using addresses or symbols, if symbols have been assigned. And the display of steps is changed to display the S addresses or P addresses.

Press the [P–ADRS] key to display the addresses of P addresses.
Press the [P–SYMB] key to display the symbols of P addresses.
Press the [S–ADRS] key to display the addresses of S addresses.
Press the [S–SYMB] key to display the symbols of S addresses.

(3) Displaying the list screen

While the step sequence screen is displayed and press the [CHANGE] key, a list screen of the subprograms referenced in this step sequence program is displayed.

```
PCLAD <<LIST>>    PROGRAM:(STEP SEQUENCE DEMO PROGRAM)   MONIT RUN
        S0001 P0001 (ROTATE) ROTATE THE WORK TIP
  □] P0001 □] P0002 □] P0004 □] P0005 □] P0006 □] P0007
  ____________________________________________________
  □ P0008   □ P0009  □ P0014  □ P0015  □ P0016   □ P0017
  □ P0021   □ P0022  □ P0024  □ P0025  □ P0026   □]P0027
  .     .        .        .       .          .
  .     .        .        .       .          .
  □ P0101   □]P0202  □]P0304  □]P0405  □]P0406   □]P0407

 [    UP    ] [   DOWN   ] [          ] [          ] [  ZOOM   ]


 [   MAIN   ] [  CHANGE  ] [          ] [          ] [          ]
```

[ZOOM] soft ke

To display a program, position the cursor to the program number and press the [ZOOM] key. The step sequence screen (Fig. 6.1.2 (b)) or ladder screen (Fig. 6.1.3 (c)) is automatically displayed according to the type of the program.

[MAIN] key

Press the [MAIN] key to return to the program configuration list.

[TIME] key

Press the [TIME] key to display the time display screen (Fig. 6.2.1).

[P–ADRS/P–SYMB] key

Displays the addresses specified to subprograms, using addresses or symbols, if symbols have been assigned. When the [P–ADRS] key is pressed, the addresses are displayed. When the [P–SYMB] key is pressed, the symbols are displayed.

[CHANGE] key

Press the [CHANGE] key to return to the step sequence diagram.

6.1.3 Ladder Screen

(1) Position the cursor to a program indicated by □, then press the [ZOOM] key.

```
STPSEQ <<MAIN>>   PROGRAM:(STEP SEQUENCE DEMO PROGRAM)   MONIT RUN

 LEVEL1     LEVEL2    LEVEL3
 □ P0001    □ P0002   □ P0004   □ P0005   □ P0006    □ P0007
 □ P0008    □ P0009   □ P0014   □ P0015   □ P0016    □ P0017
 □ P0021    □ P0022   □ P0024   □ P0025   □ P0026    □]P0027
 .       .         .         .        .          .
 .       .         .         .        .          .
 □ P0101    □]P0202   □]P0304   □]P0405   □]P0406    □]P0407

[    UP    ] [   DOWN   ] [   TIME   ] [  P–ADRS  ] [   ZOOM   ]
```

Fig. 6.1.3 (a) Program configuration list (main screen)

Example)

When the cursor is positioned to LEVEL1, press the[ZOOM] key, the first level ladder is displayed.

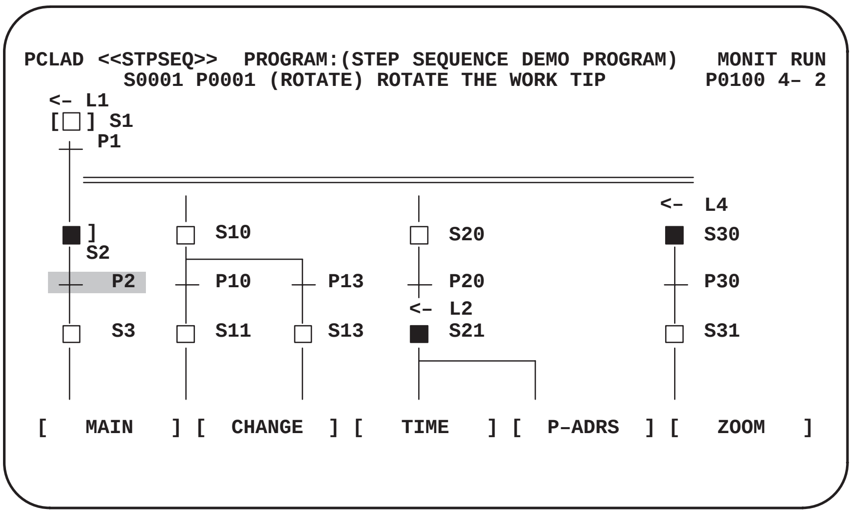


Fig. 6.1.3 (b) Step Sequence screen

Example)

When the cursor is positioned to “P2”, press the [ZOOM] key, subprogram P2 is displayed.

(2) Ladder Screen

The signals currently set to on are displayed in white (highlighted on a monochrome display).

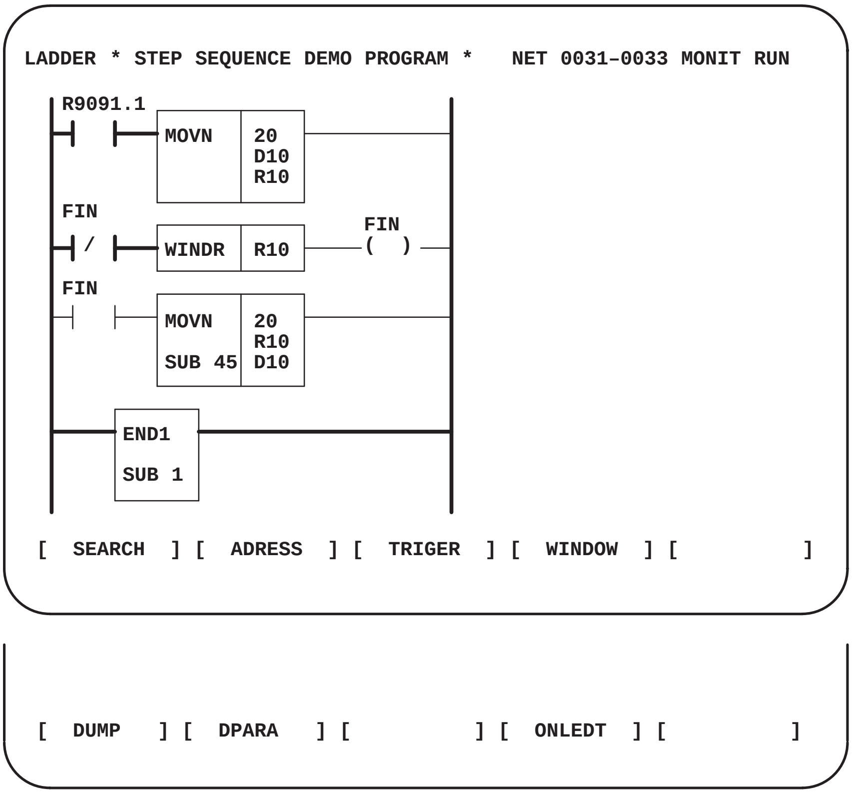
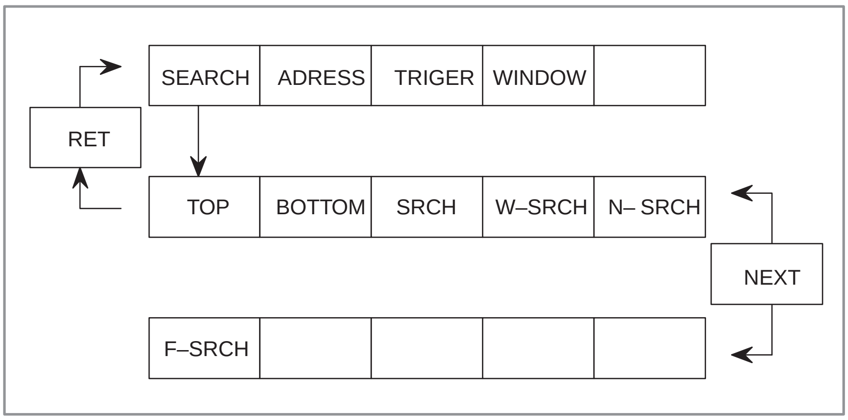


Fig. 6.1.3 (c) Ladder screen

[SEARCH] key

Used for search within a subprogram.



[TOP] key

Displays the top of a subprogram.

[BOTTOM] key

Displays the bottom of a subprogram.

[SRCH] key

Searches for the specified address.

[W–SRCH] key

Displays the ladder in which the specified address is used as a coil address.

[N–SRCH] key

Displays the ladder having the specified net number, at the top of the screen.

[F–SRCH] key

Displays the specified functional instruction, at the to of the screen.

[ADRESS/SYMBOL] key

Displays the addresses specified with relays and coils, using addresses or symbols, if symbols have been assigned. When the [ADRESS] key is pressed, the addresses are displayed. When the [SYMBOL] key is pressed, the symbols are displayed.

[TRIGER] key

With a manual operation or a signal trigger function, a renewal screen of a ladder monitoring function is stopped. By this function, the signal status when one signal is changed is certainly checked.

[WINDOW] key

Splits the screen into two sections, allowing the display of two ladder positions in a subprogram.

[DUMP] key

Displays the contents of addresses at the bottom of the screen.

[DPARA] key

Displays the data specified with functional instructions.

[ONLEDT] key

While a sequence program is executing, a part of the ladder diagram can be changed.

6.2 TIMER SCREEN

The elapsed time of a step sequence program is displayed.

6.2.1 Time Screen

```
STPSEQ <<STATUS>> PROGRAM:(STEP SEQUENCE DEMO PROGRAM)   MONIT RUN

  STEP NO.    STATUS  ELAPSE MONITOR       STEP NO.    STATUS.
S0001(     )  EXEC   1000000 T(1) OVER   S0010(TILE )  EXEC
S0002(     )  EXEC       100 T(3)        S0011(     )
S0003(     )  EXEC     10000 T(4)        S0012(     )
S0004(     )         1000000             S0013(     )
 .           .   .                         .          .
 .           .   .                         .          .

 [    UP    ] [   DOWN   ] [  SEARCH  ] [  RESET   ] [  MONIT  ]
```

Fig. 6.2.1 Time screen

Meaning of display

Display	Contents
STEP NO.	Step number S0001 : Step number (123456) : symbol display
STATUS	Step state EXEC : Active space : Inactive
ELAPSE	Actual elapsed time (per msec) The time is increasing during active state.
MONITOR	Monitor time T (1) : monitoring time number OVER : An elapsed time is over monitoring time

[UP] [DOWN] key

Scrolls the screen up or down, in units of pages, to display the operation time of other steps. Acts in the same way as the page up or down key.

[MONIT] key

Displays the screen used for setting the timer to monitor the operation time. (See 6.3)

[SEARCH] key

Search and display the specified step number.

example) Display the S100 address.

Key in "100" and press the [SEARCH] key.

[RESET] key

For all of monitoring steps, the error status which occurred by the monitoring function is canceled.

To cancel the status per steps, press the [DELETE] key on the monitor time screen. (Please refer to 6.3 Monitor Time Screen below)

6.2.2 Monitoring Elapsed Time

When an activated state remains set for longer than the specified time, the state may be determined as being erroneous. The elapsed time can be specified for up to eight steps.

When an activated state remains set for longer than the specified time,

(1) OVER is displayed at the corresponding step number on the STPSEQ/TIME screen.

(2) Execution of the ladder continues.

(3) The bit of address R9118 which corresponds with the step number is set to

1. The processes for the error status can be program by the ladder diagram. And the following message is displayed on the PMC/ALARM screen.

   ”ER48 STEP SEQUENCE TIME OVER (xxH)”

   ”xx” displays the content of address R9118 in hexadecimal code.

Time Number	Corresponding Address	Time Number	Corresponding Address
1	R9118.0	5	R9118.4
2	R9188.1	6	R9118.5
3	R9188.2	7	R9188.6
4	R9188.3	8	R9188.7

6.3 MONITOR TIME SCREEN

Operation time limits can be specified for a step sequence program. Up to eight steps can be monitored.

```
STPSEQ<<MONITOR>> PROGRAM:(STEP SEQUENCE DEMO PROGRAM)  MONIT RUN

      NO.   STEP NO.        ELAPSE   MONITOR
      T(1)  S0001(     )   1000000      2000
      T(2)  S0010(MOVE )       100      1000
      T(3)  S0002(     )       100      2000
      T(4)  S0003(     )     10000     20000
      T(5)
      T(6)
      T(7)
      T(8)

 [  DELETE  ] [          ] [          ] [          ] [          ]
```

Fig. 6.3 monitor time screen

Meaning of display

Display	Meaning
NO.	Monitor time number T (1) : means monitor time 1.
STEP NO.	Step number S0001 : Step number (123456) : symbol display
ELAPSE	Actual elapsed time (per msec) The time is increasing during active state.
MONITOR	Monitor time (per msec)

[DELETE] key

Delete the definition of monitor time.

Operation

Definition of monitor

(1) Position the cursor at the input position and input a step (or symbol).

```
      NO.   STEP NO.        ELAPSE   MONITOR
      T(1)  S0001(     )   1000000      2000
      T(2)  S0010(MOVE )
```

Key in “MOVE” and push [INPUT] key.

(2) Position the cursor at the input position and define a monitor time.

```
      NO.   STEP NO.        ELAPSE   MONITOR
      T(1)  S0001(     )   1000000      2000
      T(2)  S0010(MOVE )      1000       100
```

Keyin “100” and push [INPUT] key.

Deletion of monitor

Position the cursor at the deletion and press [DELETE] key.

```
NO.   STEP NO.         ELAPSE   MONITOR
T(1)  S0001(     )    1000000      2000
T(2)  S0010(MOVE )        100      1000
```

Alteration of monitor

Position the cursor at the alteration position and input again.

```
NO.   STEP NO.         ELAPSE   MONITOR
T(1)  S0001(     )    1000000      2000
T(2)  S0100(     )       2000      1000
```

Key in “S100” and push <INPUT> key.

6.4 EDITING FUNCTION OF LADDER DIAGRAM

The display and editing of a step sequence program per subprogram aresupported.
A step sequence program is allowed to be displayed and a ladder diagram is allowed to be displayed and edited.

6.4.1 Program Configuration List (Main Screen)

Press the [EDIT] and [LADDER] key and display the program configuration list.

```
STPSEQ <<MAIN>>   PROGRAM:(STEP SEQUENCE DEMO PROGRAM)   MONIT STOP
                   P0001 ( ) SUB PROGRAM NO.1

  LEVEL1     LEVEL2    LEVEL3
  □ P0001    □ P0002   □ P0004   □ P0005   □ P0006    □ P0007
  □ P0008    □ P0009   □ P0014   □ P0015   □ P0016    □ P0017
  □ P0021    □ P0022   □ P0024   □ P0025   □ P0026    □]P0027
  .      .         .         .        .          .
  .      .         .         .        .          .
  □ P0101    □ P0202   □]P0304   □]P0405   □]P0406    □]P0407

 [    UP    ] [   DOWN   ] [          ] [  P–ADRS   ] [   ZOOM   ]
```

Items displayed on the screen

Display	Contents	Display by [ZOOM] key
LEVEL1	Ladder first level	Ladder diagram
LEVEL2	Ladder second level	Ladder diagram
LEVEL3	Ladder third level (Note)	Ladder diagram
□ Pxxx	Subprogram	Ladder diagram
□] Pxxx	Subprogram	Step sequence diagram

Pxxx indicates a subprogram number.

> **NOTE**
> The third level ladder can be omitted.

[ZOOM] key

To display the contents of a program, position the cursor to the program number and press the [ZOOM] key. The step sequence diagram (Fig. 6.4.2 (b)) or ladder diagram (Fig. 6.4.3 (c)) is automatically displayed according to the type of the program.

[P–ADRS/P–SYMB] key

Displays the addresses specified to subprograms, using addresses or symbols, if symbols have been assigned. When the [P–ADRS] key is pressed, the addresses are displayed. When the [P–SYMB] key is pressed, the symbols are displayed.

6.4.2 Step Sequence Screen

(1) Position the cursor to a program indicated by □], then press the [ZOOM] key.

```
LADDER <<MAIN>>    PROGRAM:(STEP SEQUENCE DEMO PROGRAM)   MONIT STOP
                    P0407 ( ) STEP SEQUENCE NO.1

  LEVEL1     LEVEL2     LEVEL3
  □ P0001    □ P0002    □ P0004    □ P0005    □ P0006    □ P0007
  □ P0008    □ P0009    □ P0014    □ P0015    □ P0016    □ P0017
  □ P0021    □ P0022    □ P0024    □ P0025    □ P0026    □]P0027
  .       .         .         .        .          .
  .       .         .         .        .          .
  □ P0101    □ P0202    □]P0304    □]P0405    □]P0406    □]P0407

 [    UP    ] [   DOWN    ] [   TIME    ] [  P–ADRS   ] [   ZOOM    ]
```

Fig. 6.4.2 (a) Program configuration list (main screen)

Example)

When the cursor is positioned to □] P0407 and press the [ZOOM] key , the subprogram P407 is displayed.

(2) Displayed Step Sequence

Activated steps are indicated by red □ (highlighted □ on a monochrome display). (In this manual, activated steps are indicated by ■.)

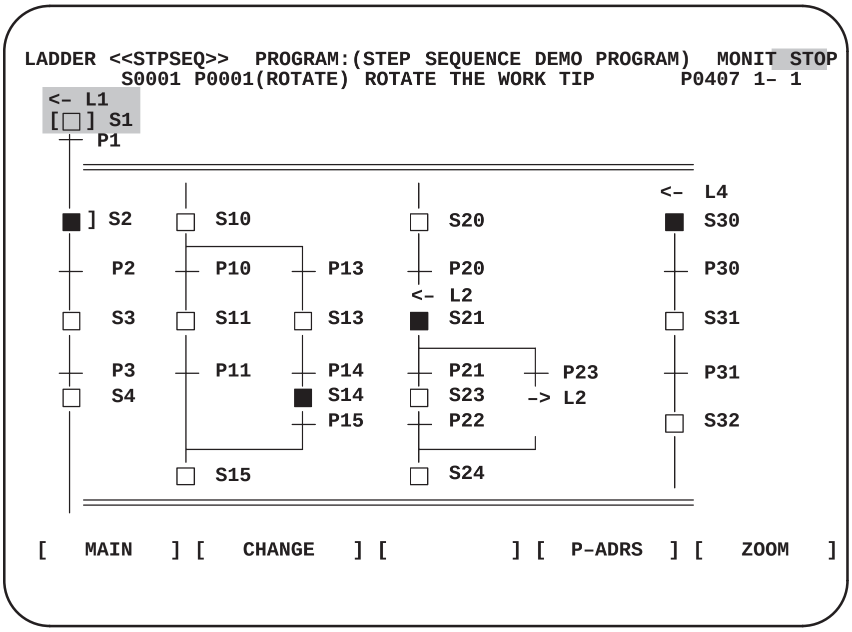


Fig. 6.4.2 (b) Step sequence screen

Meaning of display

Display	Contents	Display by [ZOOM] key
[□] Sxxx	Initial step	Ladder diagram
□ Sxxx	Step	Ladder diagram
□] Sxxx	Block step	Step sequence diagram
+ Pxxx	Transition	Ladder diagram
————	Selective sequence	Cannot zoom.
════	Simultaneous sequence	Cannot zoom.
└→L2	Jump	Cannot zoom.
←–L2	Label	Cannot zoom.

Pxxx means the subprogram number.

[ZOOM] key

To display the contents of a program, position the cursor to the program number and press the [ZOOM] key. The step sequence diagram (Fig. 6.4.2 (b)) or ladder diagram (Fig. 6.4.3 (c)) is automatically displayed according to the type of the program.

[MAIN] key

Press the [MAIN] key to return to the program configuration list.

[CHANGE] key

Press the [CHANGE] key to list the subprograms referenced in the step sequence program.

[P–ADRS/P–SYMB/S–ADRS/S–SYMB] key

Displays the addresses specified with steps and transitions, using addresses or symbols, if symbols have been assigned.

And the display of steps is changed to display the S addresses or P addresses.

Press the [P–ADRS] key to display the addresses of P addresses.
Press the [P–SYMB] key to display the symbols of P addresses.
Press the [S–ADRS] key to display the addresses of S addresses.
Press the [S–SYMB] key to display the symbols of S addresses.

(3) Displaying the list screen

While the step sequence screen is displayed and press the [CHANGE] key, a list screen of the subprograms referenced in this step sequence program is displayed.

```
LADDER <<LIST>>    PROGRAM:(STEP SEQUENCE DEMO PROGRAM)  MONIT STOP
        S0001 P0001 (ROTATE) ROTATE THE WORK TIP
  □] P0001 □] P0002 □] P0004 □] P0005 □] P0006 □] P0007
  ____________________________________________________
  □ P0008   □ P0009  □ P0014  □ P0015  □ P0016   □ P0017
  □ P0021   □ P0022  □ P0024  □ P0025  □ P0026   □]P0027
  .      .        .        .       .          .
  .      .        .        .       .          .
  □ P0101   □ P0202  □]P0304  □]P0405  □]P0406   □]P0407

 [    UP    ] [  CHANGE  ] [          ] [  P–ADRS  ] [  ZOOM   ]
```

[ZOOM] soft key

To display a program, position the cursor to the program number and press the [ZOOM] key. The step sequence screen (Fig. 6.4.2 (b)) or ladder screen (Fig. 6.4.3 (c)) is automatically displayed according to the type of the program.

[MAIN] key

Press the [MAIN] key to return to the program configuration list.

[CHANGE] key

Press the [CHANGE] key to return to the step sequence diagram.

[P–ADRS/P–SYMB] key

Displays the addresses specified to subprograms, using addresses or symbols, if symbols have been assigned. When the [P–ADRS] key is pressed, the addresses are displayed. When the [P–SYMB] key is pressed, the symbols are displayed.

6.4.3 Ladder Screen

(1) Position the cursor to a program indicated by □, then press the [ZOOM] key.

```
LADDER <<MAIN>>   PROGRAM:(STEP SEQUENCE DEMO PROGRAM)   MONIT STOP

 LEVEL1     LEVEL2   LEVEL3
 □ P0001    □ P0002  □ P0004  □ P0005  □ P0006   □ P0007
 □ P0008    □ P0009  □ P0014  □ P0015  □ P0016   □ P0017
 □ P0021    □ P0022  □ P0024  □ P0025  □ P0026   □]P0027
 .      .       .        .        .        .
 .      .       .        .        .        .
 □ P0101    □ P0202  □]P0304  □]P0405  □]P0406   □]P0407
[   UP    ] [  DOWN  ] [         ] [  P–ADRS  ] [  ZOOM   ]
```

Fig. 6.4.3 (a) Program configuration list (main screen)

Example)

When the cursor is positioned to LEVEL1, press the [ZOOM] key, the first level ladder is displayed.

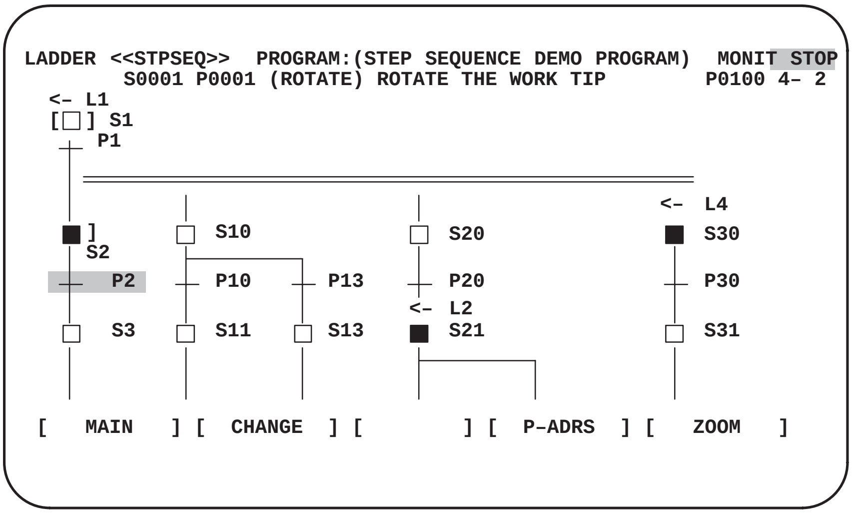


Fig. 6.4.3 (b) Step Sequence screen

Example)

When the cursor is positioned to ”┼P2”, press the [ZOOM] key, the subprogram P2 is displayed.

(2) Ladder Screen

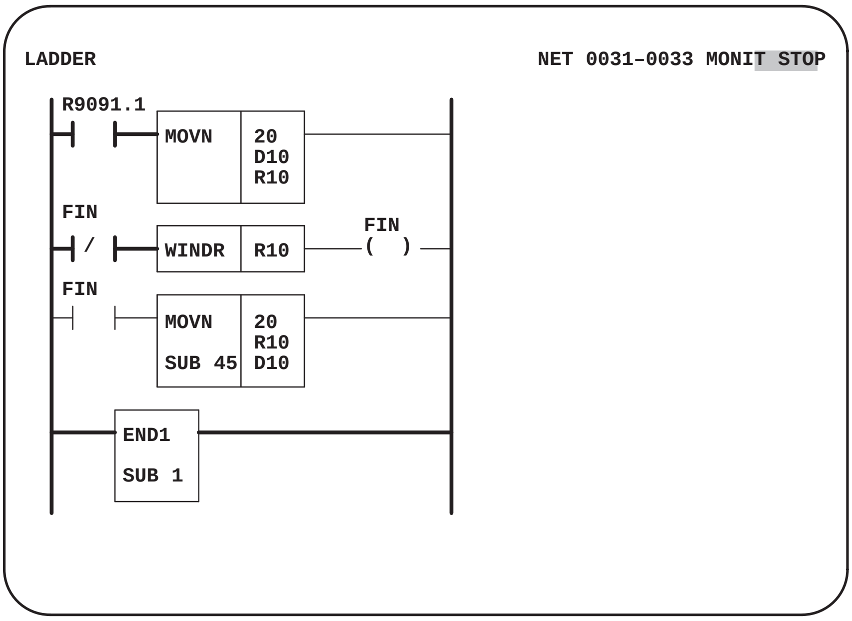


Fig. 6.4.3 (c) Ladder screen

Please refer to the following section about the operations of editing a ladder diagram.

III PMC PROGRAMMER(CRT/MDI) 5.2 Sequence Program Generation(LADDER)

6.5 CORRESPONDING FUNCTION

The following ladder diagnosis and debugging functions can be used together with the step sequence functions.

○ : can be used
Δ : can be used on condition
× : cannot be used

Functions	PMC–SB4/SB6	PMC–SC4	PMC–NB2
PMC Ladder diagram display (PMCLAD)	×	×	×
PMC I/O signal display (PMCDGN)			
Title screen (TITLE)	○	○	○
Signal status screen (STATUS)	○	○	○
Alarm screen (ALARM)	○	○	○
Trace screen (TRACE)	○	○	○
Contents of Memory (MEMORY)	○	○	○
Signal Waveforms screen (ANALYS)	Δ NOTE1	○	○
Running State of a User Task (USRDGN)	×	○	○
PMC Parameters screen (PMCPRM)			
Timer screen (TIMER)	○	○	○
Counter screen (COUNTR)	○	○	○
Keep relay screen (KEEPRL)	○	○	○
Data table screen (DATA)	○	○	○
Simple setting screen (SETING)	○	○	○
Step Sequence screen (STPSEQ)			
Displaying Step Sequence screen	○	○	○
Displaying Ladder screen	○	○	○
SEARCH	○	○	○
Display address and symbol	○	○	○
Trigger function (TRIGER)	○	○	○
Divided screen function (WINDOW)	○	○	○
Contents of memory (DUMP)	○	○	○
Contents of parameter (DPARA)	○	○	○
online editting (ONLEDT)	Δ	○	○
Time screen (TIME)	○	○	○
Monitor time screen (MONIT)	○	○	○
Execute or stop the sequence program (RUN)	○	○	○
Edit function (EDIT)	NOTE1		
Title screen (TITLE)	Δ	○	○
Ladder diagram (LADDER)	Δ	○	○
Symbol screen (SYMBOL)	Δ	○	○
Message screen (MESAGE)	Δ	○	○
Definition of I/O (MODULE)	Δ	○	○
Cross reference (CROSS)	Δ	○	○
Memory clear (CLEAR)	Δ	○	○
Input and output			
FAPT LADDER (HOST)	○	○	○
Floppy cassette (FDCAS)	○	○	○
FlashROM (F–ROM)	○	○	○
Memory card (M–CARD)	Δ NOTE2	○	○
Other I/O device (OTHERS)	○	○	○
System Parameter (SYSPRM)	Δ NOTE1	○	○
Debug function (MONIT)	NOTE1		
Ladder debug function (DBGLAD)	×	×	×
Descriptor table screen (GDT)	×	○	○
User memory screen (USRMEM)	×	○	○
User program debug function (DEBUG)	×	○	○

CAUTION
1 An Editor card is needed.
2 It is possible to use while an Editor card is not mounted. (Can be used with the 16*i*/18*i*/21*i*.)

6.6 COMPATIBILITY OF LADDER DIAGRAM

The PMC–SB4, SB6, SC4 and NB2 can be used with either the ladder method or step sequence method. When a step sequence program is transferred to the old version of the PMC, ER08 OBJECT UNMATCH is displayed on the PMC/ALARM screen.

The model setting of FAPT LADDER determines whether the ladder or step sequence method is used. The STEP SEQUENCE item has been added to the system parameter screen for future expansion. Specify the parameter according to the model setting of FAPT LADDER.

To create a program with the built–in edit function, after the parameter has been set execute CLEAR ALL. Alternatively, while holding down “X” and “O” key, turn the power off and on.

When the step sequence method is used: STEP SEQUENCE = YES.

When the ladder method is used: STEP SEQUENCE = NO.

```
PMC SYSTEM PARAMETER  (1/2)        MONIT STOP

  COUNTER DATA TYPE    =  BINARY / BCD

  STEP SEQUENCE        =     YES / NO







>

 [BINARY] [  BCD  ] [      ] [      ] [      ]
```

Fig. 6.6 (a) PMC–SB4/SB6 system parameter screen (first page)

```
PMC SYSTEM PARAMETER  (2/2)        MONIT STOP

  FS0 OPERATOR PANEL    =     YES / NO

  KEY DI ADDRESS        =

  LED DO ADDRESS        =

  KEY BIT IMAGE ADDRESS =

  LED BIT IMAGE ADDRESS =


>

 [  YES  ] [  NO  ] [      ] [      ] [      ]
```

Fig. 6.6 (b) PMC–SB4/SB6 system parameter screen (second page)

```
PMC SYSTEM PARAMETER  (1/2)        MONIT STOP

  COUNTER DATA TYPE     =  BINARY / BCD

  LADDER EXEC           =     % (1–150)

  LANGUAGE EXEC RATIO   =    % (0–99)

  LANGUAGE ORIGIN       =       H
  (LANGUAGE AREA =       H, SIZE =     KB)

  STEP SEQUENCE         =     YES / NO

>

 [BINARY] [  BCD  ] [      ] [      ] [      ]
```

Fig. 6.6 (c) PMC–SC4/NB2 system parameter screen (first page)

```
PMC SYSTEM PARAMETER  (2/2)        MONIT STOP

  FS0 OPERATOR PANEL    =     YES / NO

  KEY DI ADDRESS        =

  LED DO ADDRESS        =

  KEY BIT IMAGE ADDRESS =

  LED BIT IMAGE ADDRESS =


>

 [  YES  ] [  NO  ] [      ] [      ] [      ]
```

Fig. 6.6 (d) PMC–SC4/NB2 system parameter screen (second page)

V. PMC PROGRAMMER (SYSTEM P series)

1 GENERAL

The FAPT LADDER system can easily prepare sequence programs, symbol data, titles, and message of PMC–SB and PMC–SC, and also easily define addresses of the modules to be installed in an I/O unit by using SYSTEM P series.

Major functions of this FAPT LADDER are as described below.

(1) Input, display and editing of sequence programs

(2) Transfer of sequence programs (including write into EPROM for PMC or ROM module.)

(3) Collation of sequence programs

(4) Program error display

The SYSTEM P series is used in the stage of preparing a sequence program only and separated from PMC after the sequence program has been completed. The SYSTEM P series can be connected to PMC only when the PMC is operated with the RAM card and cannot be connected when PMC is operated with a EPROM for PMC or ROM module.

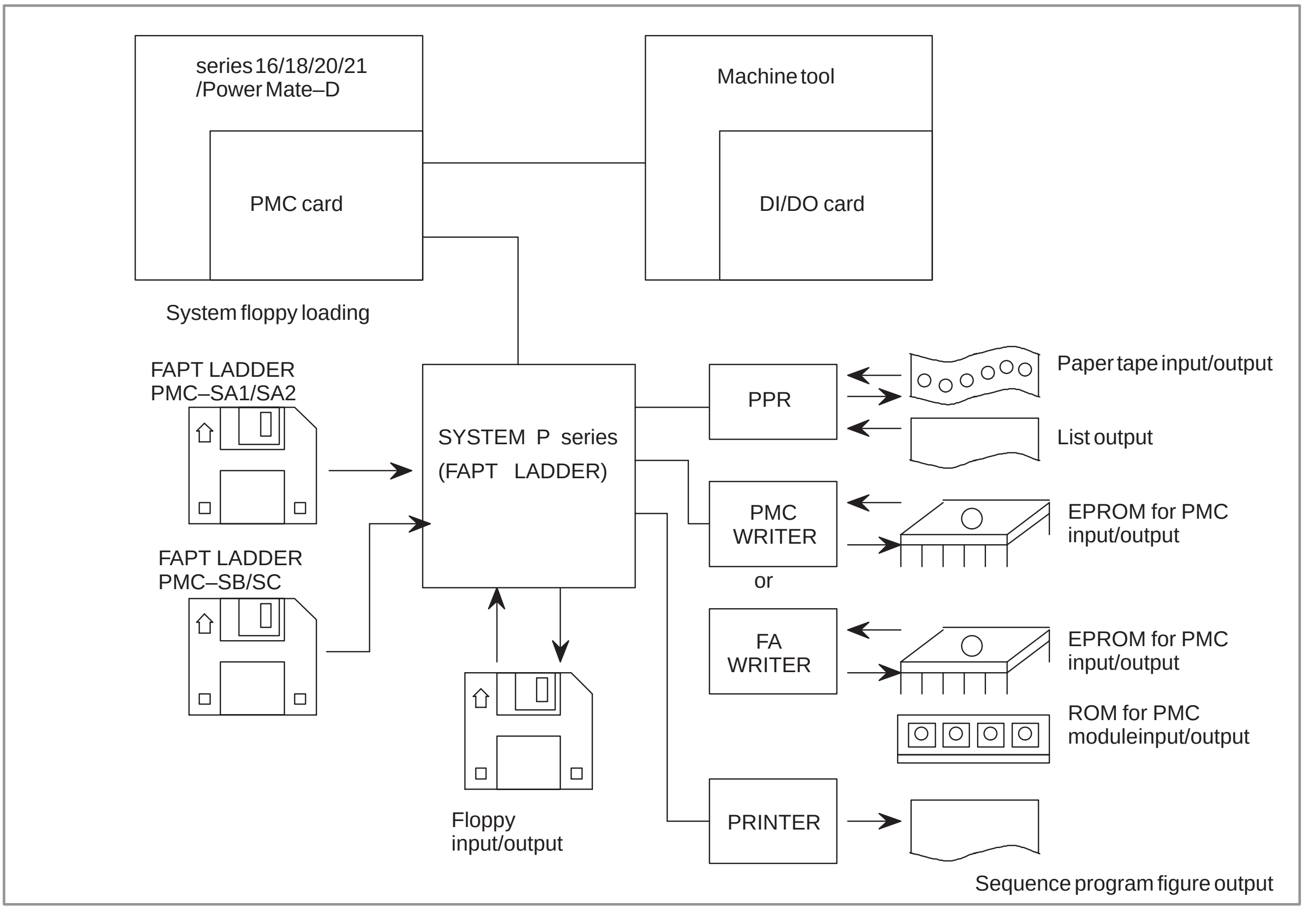


Fig. 1

2 FUNCTIONS OF PROCESSING

(1) Input of sequence programs

Input sequence programs using the following units when sequence programs are loaded into the memory of the SYSTEM P series.

a) SYSTEM P series keyboard

b) PPR tape reader (paper tape)

c) Floppy

d) PMC memory

e) EPROM for PMC or ROM module

(2) Sequence program display

Sequence programs can be displayed on the 12” graphic display of SYSTEM P series as follows.

a) Sequence programs can be displayed using mnemonic symbols.

b) Sequence programs can also be displayed in the ladder diagram format.

(3) Editing of sequence programs

A sequence program can be edited by using the SYSTEM P series keyboard in the following three ways.

a) Alteration

b) Insertion

c) Deletion

(4) Transfer of sequence programs

Sequence programs can be transferred as follows.

a) From SYSTEM P series memory to PMC memory

b) From PMC memory to SYSTEM P series memory

c) From SYSTEM P series memory to floppy

d) From floppy to SYSTEM P series memory

e) From SYSTEM P series memory to EPROM or ROM module for PMC (Write into EPROM for PMC or ROM module)

f) From EPROM for PMC or ROM module to SYSTEM P series memory

(5) Collation of sequence programs

Sequence programs can be checked by collating them between the following memories.

a) SYSTEM P series memory – PMC memory

b) SYSTEM P series memory – floppy

c) SYSTEM P series memory – EPROM for PMC or ROM module

d) SYSTEM P series memory – paper tape

(6) Hard copy

a) Since FANUC PPR is connectable to SYSTEM P series, the paper tape output and list output (mnemonic symbol) are obtainable.

b) A ladder diagram can be printed out.

(7) Program error display

Sequence program errors are displayed on the screen of the SYSTEM P series.

Error codes are displayed at the lower right of the screen as ALARM=XXX.
Refer to list of error codes in Appendix.

3 COMPONENT UNITS AND CONNECTIONS

3.1 COMPONENT UNITS

(1) SYSTEM P series

This system serves as a programmer to generate and edit sequence programs.

(2) Series 16

This system transfers a generated sequence program to CNC.

(3) FANUC PPR

This PPR inputs/outputs a sequence program by using a paper tape, and also output a source list to the printer.

(4) FANUC printer

This printer prints out the sequence program.

(5) FANUC PMC writer

This unit is used for writing a sequence program to the EPROM for PMC or ROM module when the sequence program has been completed.

(6) FANUC FA Writer

This unit is used for writing data to the EPROM or ROM module for the PMC after a sequence program has been created.

3.2 CONNECTIONS OF UNITS

For details of the connections of SYSTEM P series unit power supply, PPR, and other units as well as their operation, refer to the following operator's manuals.

SYSTEM P–G Mark II: B–66014E

SYSTEM P–G Mate: B–66003E

This chapter mainly describes the connections between SYSTEM P series and I/O devices.

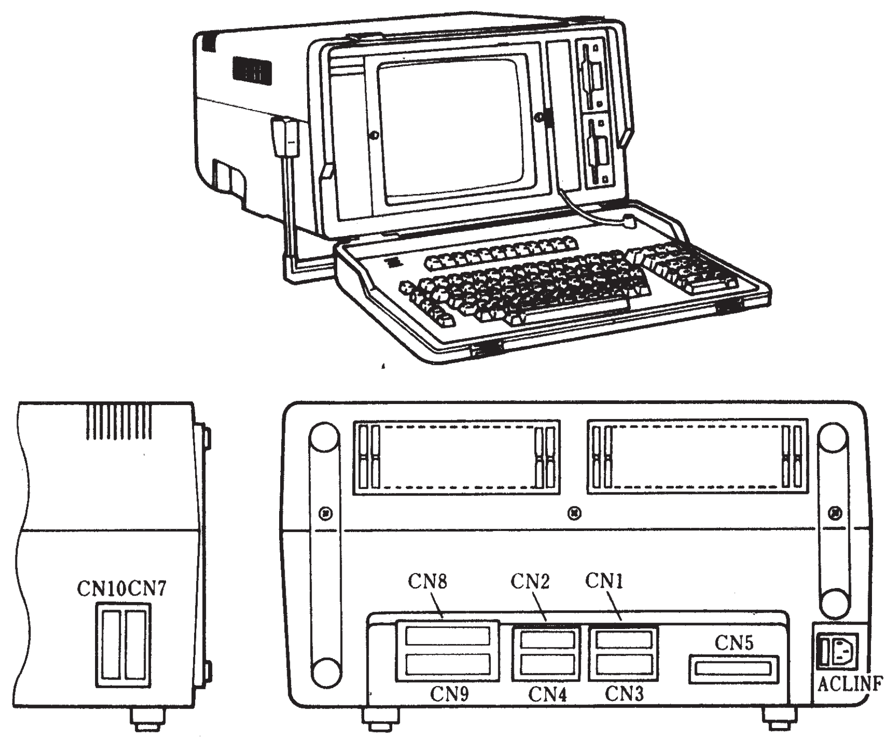


Fig. 3.2 (a) External view of SYSTEM P Mark II

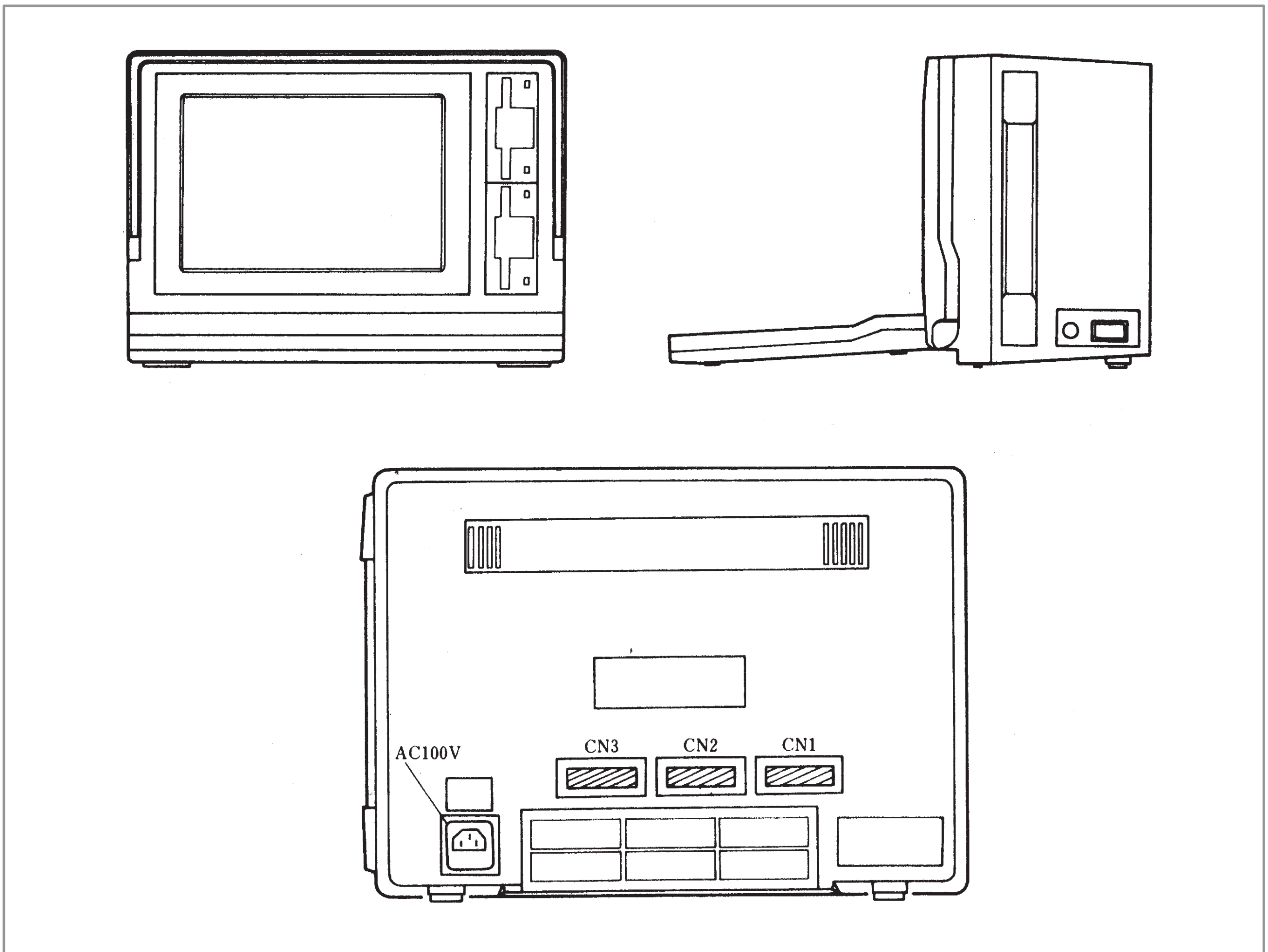


Fig. 3.2 (b) External view of SYSTEM P Mate

Since a volatile RAM is employed as the SYSTEM P series memory, all programs (FAPT LADDER system programs and sequence program) being loaded into memory are operation should be started with the input of FAPT LADDER system programs (called system loading).

If the SYSTEM P series power supply is turned off halfway in the curse of inputting a sequence program from the keyboard, the sequence program must be stored in advance, and this FAPT LADDER provides an output function to a floppy for this purpose.

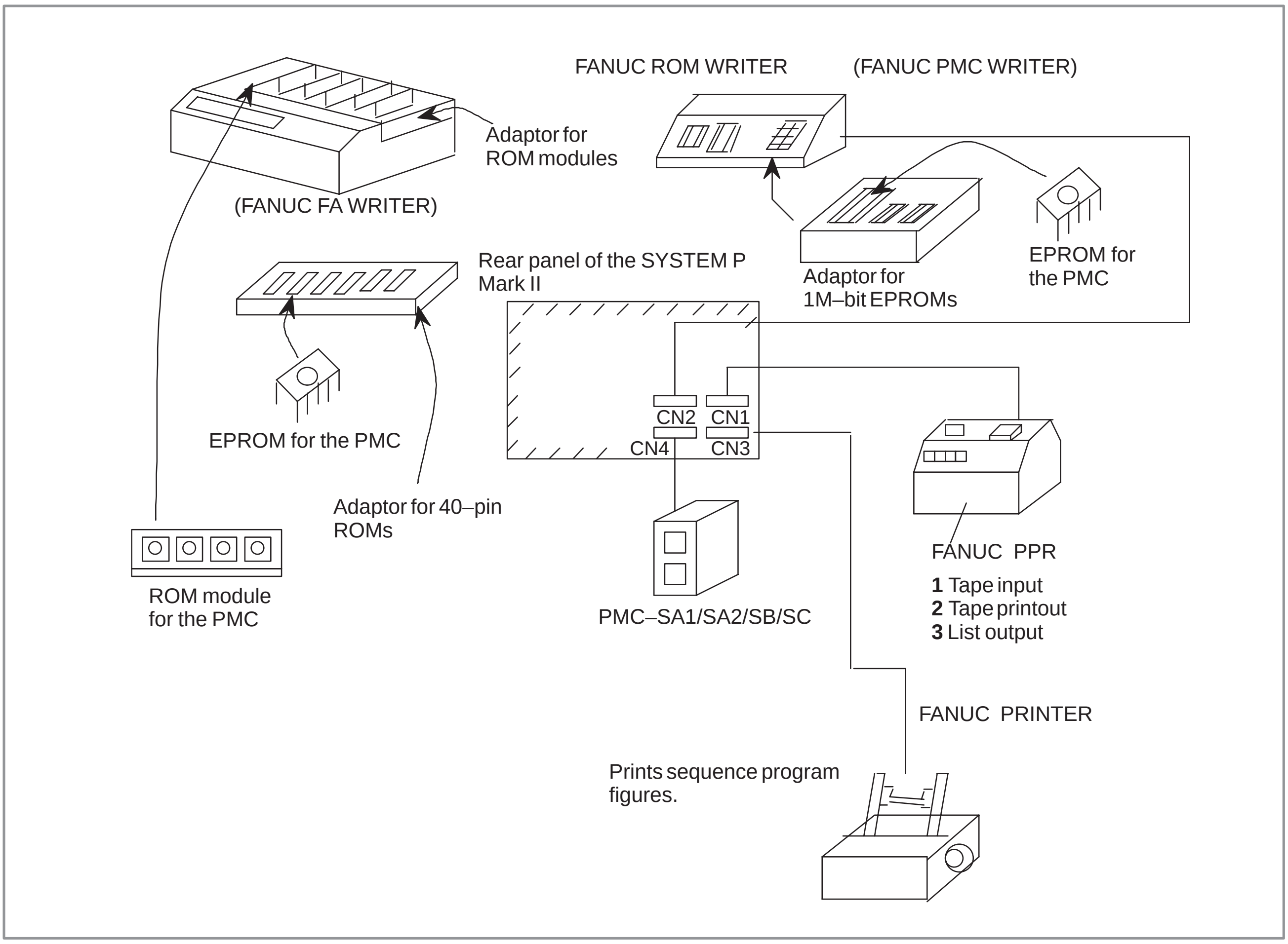


Fig. 3.2 (c) Connection of SYSTEM P series with each unit

(1) Connect FANUC PPR to connector CN1.

(2) Connect FANUC PMC writer or FANUC FA writer to connector CN2.

(3) Connect FANUC printer to connector CN3.

(4) Connect connector CN4 to PMC–SA1/SA2/SB/SB2/SC. It is connected to a channel preset by a PMC I/O.
For details, refer to ”Setting and display of I/O in PMC programmer (CRT/MDI) in III”.

Connector JD5A on MAIN PCB → 1 CHANNEL

Connector JD5B on MAIN PCB → 2 CHANNEL

3.3 KEYBOARD OF SYSTEM P SERIES

Figs. 3.3(a) – (b) show the panel of the SYSTEM P series keyboard.

It is not necessary to memorize the meanings of keys on the keyboard.

Descriptions of these keys and menus are displayed on the SYSTEM P series screen by operation, and you can easily operate the SYSTEM P series board while monitoring the SYSTEM P series screen.

In this chapter, you should understand an outline of functions of these keys.

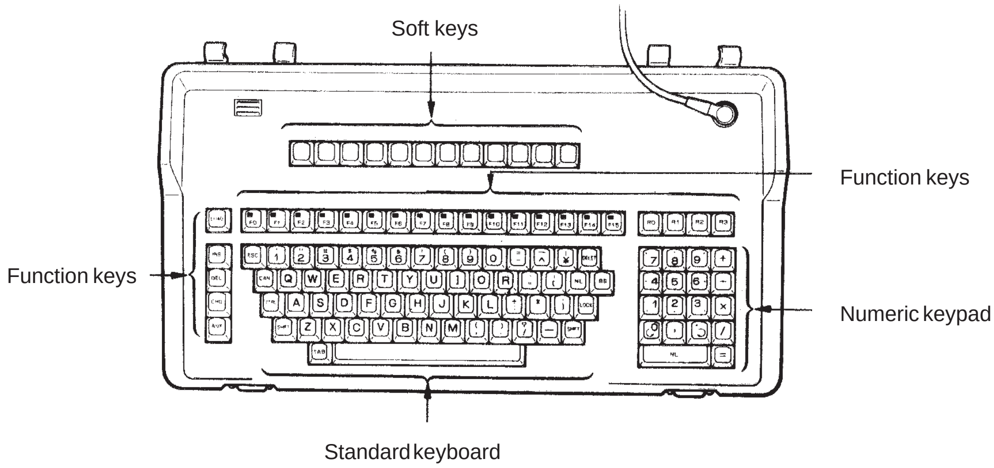


Fig. 3.3 (a) Panel of the SYSTEM P Mark II key board

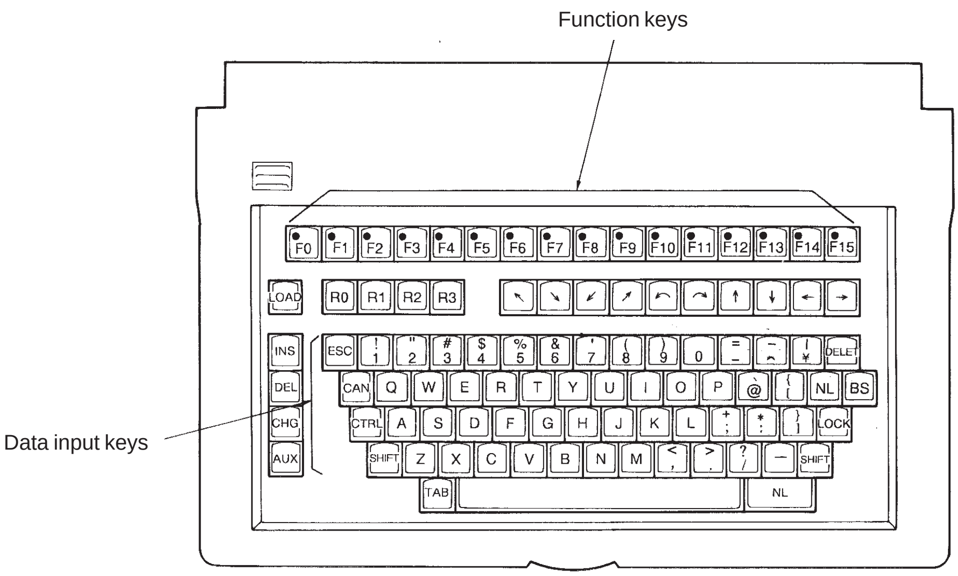


Fig. 3.3 (b) Panel of the SYSTEM P Mate key board

3.3.1 LOAD Key (System Program Loading Key)

This key is used to load the FAPT LADDER system program into the SYSTEM P series memory through a floppy disk at the first time after turning on power.

3.3.2 F Keys (F1 to F0)

F key is used to select an I/O device among I/O devices connected at that time.

These F keys are provided with an LED. When depressing a key, the LED lights, and when depressing the key once more, the LED goes out. The lighting condition of of this LED indicates that an I/O has been designated. No I/O device is operable when its corresponding LED is not lighting.

The correspondence between F keys and I/O devices is as shown below. (I) shows an input, while (O) shows an output.

(1) <F1> key: FANUC PPR paper tape reader (I)

(2) <F2> key: Floppy disk input (I)

(3) <F3> key: Not used

(4) <F4> key: Display of ladder diagram on SYSTEM P series screen (O)

(5) <F5> key: FANUC PPR printer (O)

(6) <F6> key: FANUC PPR paper tape puncher (O)

(7) <F7> key: Floppy disk output (O)

(8) <F8> key: PMC–PA1/PA2/SA1/SA2/SB/SB2/SC (I/O)

(9) <F9> key: FANUC PMC writer, FANUC FA writer (I/O)

(10) <F10> key: FANUC printer (O)

(The ladder diagram is printed on the printer.)

(11) <F13> key: FANUC Floppy Cassette/FANUC FA Card adapter (I)

(12) <F14> key: FANUC Floppy Cassette/FANUC FA Card adapter (O)

Combination of F key and menu number of FAPT LADDER decided which function is to be executed.

3.3.3 R Keys (R0 to R3)

Four R keys <R0> to <R3> are provided. The meaning of these keys differ according to the screen conditions at their operating time, even in case of the same key.

(1) R key menu screen

This screen is obtained just after loading a FAPT LADDER system program (1/2) or when pressing <NL> key only in a menu screen. Refer to Fig. 4.2.2.

<R0> FAPT ladder start.

. A menu screen appears.

<R1> Editing a ladder diagram starts.

<R2> Not used in FAPT ladder. (Not accepted when pressing these keys)

<R3> Request key (see 4.8)

Press NL keys, if a wrong key was pressed by mistake. The screen is reset to the condition before pressing the wrong R key.

(2) Other than R key menu screen

<R0> This key operation is accepted when EDIT is displayed at the lower left part of the screen (called EDIT screen hereafter) during sequence program editing. The screen is switched to the sequence program, symbol, message, I/O module, and title, each time this R key is pressed.

<R1> **1** When this key is pressed during printing of a ladder diagram on an external printer, the printer stops every page to be ready for key entry.

2 When this key is pressed during data transfer between SYSTEM P series and PMC–SB/SC, data transfer is stopped.

3 The signal display in a sequence program is alternately selected to symbols and addresses, each time this R1 is pressed during the display of the sequence program on the screen.

<R2> Data on the last page are displayed, each time this key is pressed on the EDIT screen.

<R3> **1** Data on the next page are displayed, each time this key is pressed on the EDIT screen.

2 Transfer is aborted when this key is pressed during ROM data transfer between SYSTEM P series and PMC–WRITER or floppy.

3.3.4 Data Keys and Screen Scroll Key

Data keys are used to enter data. To switch the output of such keys between the upper character and lower–character, use the [SHIFT] key or [LOCK] key. Pressing the [SHIFT] key together with an arbitrary key changes the output of the arbitrary key to the upper character, and pressing the [LOCK] key changes the output of all keys to upper character. To release the upper character mode, press the [LOCK] key again.
Special keys are described below.

(1) <NL> key

Data entry from the SYSTEM P series keyboard are input into SYSTEM P series by depressing <NL> key.
Two <NL> keys are located on the keyboard for easily operation.

(2) <CAN> key

Data being entered from the keyboard are cancelled.

(3) BS key

Data being entered from the keyboard are sequentially deleted leftward, each time this key is depressed.

(4) Arrow keys <↑> <↓> <←> <→>

These keys are accepted only when a ladder diagram is being displayed on the screen, and used for scrolling the ladder diagram.

NOTE
None of [INS] [DEL] [CHG] [AUX] keys and K key is employable in the FAPT LADDER.

3.4 SETTING OF I/O DEVICE

(1) SYSTEM P series Mate

An initial I/O device setting of 'FAPT LADDER' for SYSTEM P series Mate is as follows.

Table 3.4(a) FAPT LADDER (Mate) of table

Channel	I/O device	F key
CN1	PMC–RAM	F8
CN2	PMC WRITER FA WRITER	F9
CN3	External printer	F10

Alter the setting of the I/O device by under–mentioned 'IO command' when using FANUC PPR.

(Setting method of IO command)

i) Press the R3 key in the menu screen of R keys.

'REQUEST =' is displayed in the left bottom of screen and becomes the state which can be typed in.

ii) Type in IO PPR, CN1 <NL>.

PPR is allocated to channel 1.

iii) Type in as follows when channel 1 allocation is returned to PMC–RAM.

IO, NC, CN1, F8, BR10 <NL>

(2) SYSTEM P Mark II

The initial setting of I/O devices of FAPT LADDER for the SYSTEM P Mark II is as follows.

Table 3.4(b) FAPT LADDER (Mark II)

Channel	I/O device	F key
CN1	FANUC PPR	F1, F5, F6
CN2	PMC WRITER FA WRITER	F9
CN3	External printer	F10
CN4	PMC–RAM	F8

(3) When a FANUC Floppy Cassette or FANUC FA Card adapter is used, change the setting of the I/O device by executing the following I/O command:

i) Press the [R3] key on the menu screen for the [R] keys.

Then REQUEST = appears at the lower left of the screen allowing data to be entered.

ii) Type IO BCA, CN2, F13, F14, then press the <NL> key. The FANUC Floppy Cassette or FA Card adapter is allocated to channel 2.

iii) To initialize the setting of channel 2 again, type IO AUX, CN2, F9, then press the <NL> key.

(4) Setting of the ROM writer

The PMC–SA1, –SA2 or –SB uses one of 1MB EPROM (27C1024).

The PMC–SC uses one of ROM module (128KB, 256KB, or 512KB).

For this reason, when the PMC–SA1, –SA2 or –SB is used, both the FA Writer and PMC Writer can be used. When the PMC–SC is used, only the FA Writer is available.

When the PMC–SA1, –SA2 or –SB is used, the ROM writer used can be selected on the REQUEST screen as follows.

1 On the REQUEST screen, enter WRITER then press the <NL> key.

2 The following message appears. To select the FA Writer, enter 0 or press the <NL> key. To select the PMC Writer, enter 1.

SET KIND OF ROM WRITER (0:FA WRITER, 1:PMC WRITER)
WRITER=

The current setting of the ROM writer can be checked on the system parameter screen.

(1) PMC Writer

The PMC Writer is required when the PMC–SA1, –SA2, –SB or SB2 is available. To use a 1MB EPROM (27C1024), the 1M EPROM adapter (A13B0147–B001) is required. Set the EPROM select switch to the 271024 position before using the 1M EPROM adapter.

(2) FA Writer

When the FA Writer is used with the PMC–SA1, –SA2 or –SB, the EPROM adapter (1MB) for the FA Writer is required. When the FA writer is used with the PMC–RC, the ROM module adapter is required.

4 OPERATION

4.1 GENERAL

Various operations of FAPT ladder are done onthe specified screen.

Fig. 4.1 shows the relation between various operations and corresponding screens.

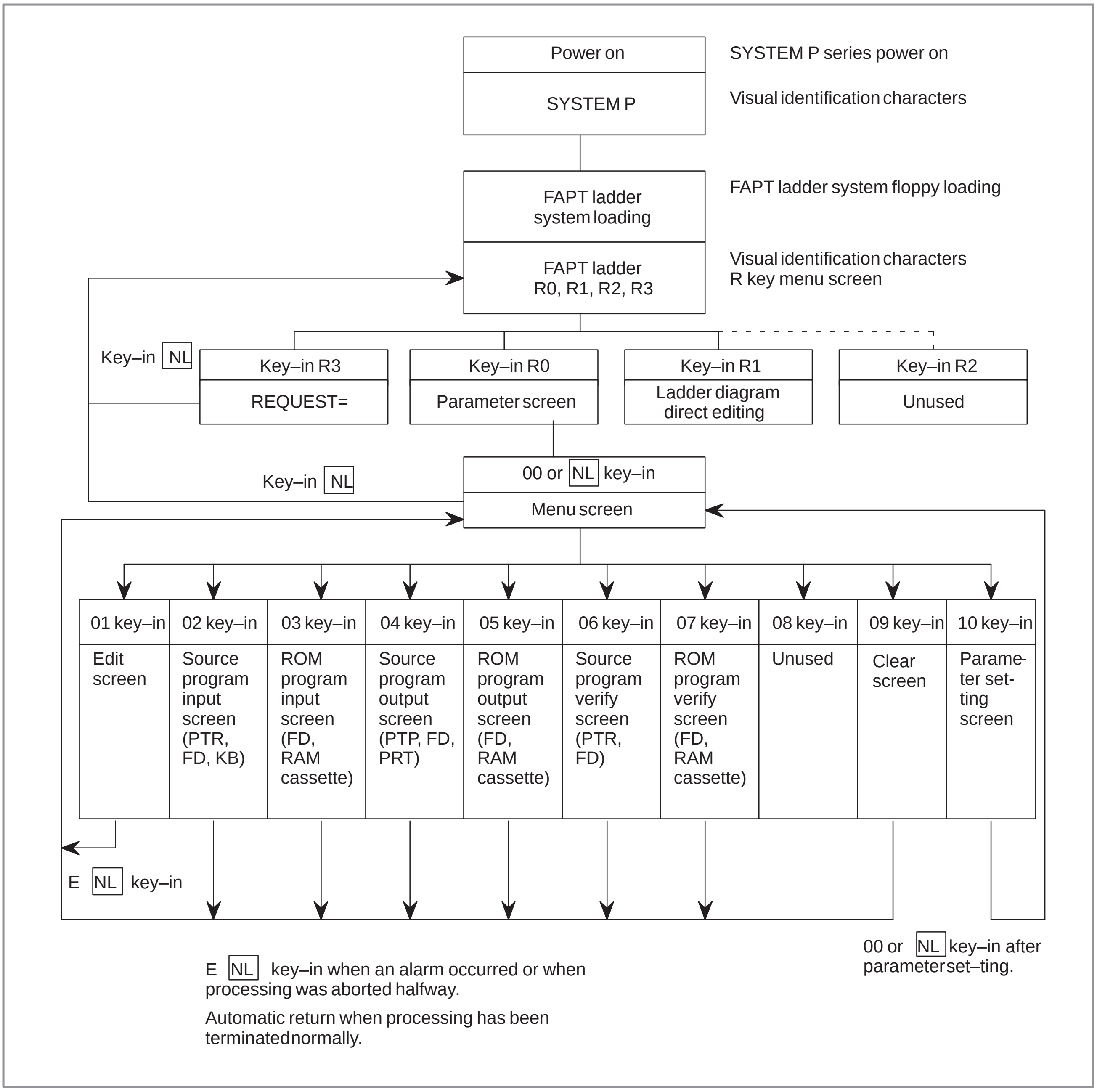


Fig. 4.1 (a) Relation between various operations and screens

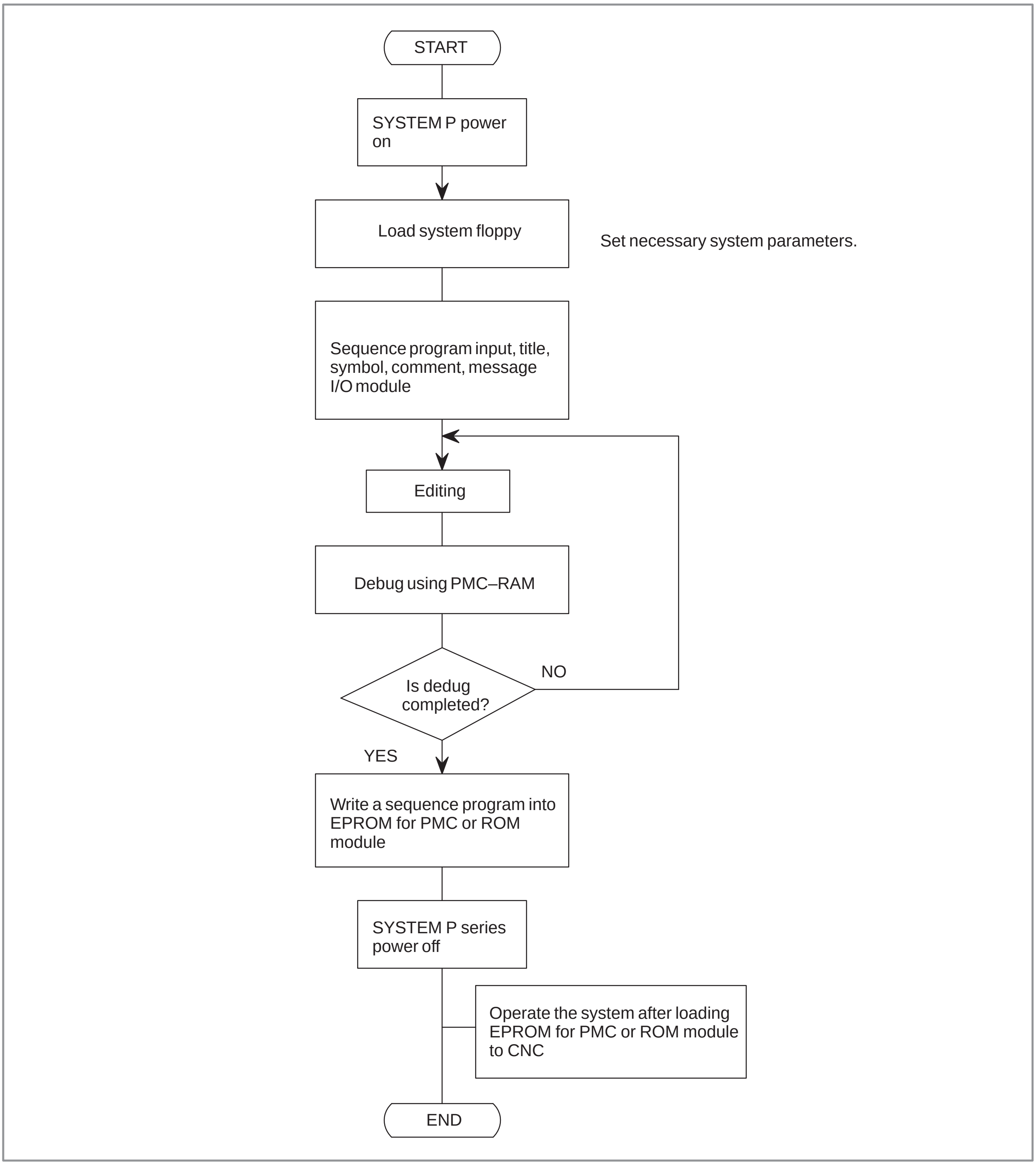


Fig. 4.1 (b) Outline of operation

4.2 PREPARATION BEFORE OPERATION

4.2.1 System Floppy

The system floppy disk contains the system of FAPT LADDER for PMC–SA1/SA2/SB/SB2/SC.

4.2.2 Limitations with the SYSTEM P Mate

To apply the FAPT LADDER system for PMC–SA1/SA2/SB/SB2/SC to the SYSTEM P Mate will overlay each of the following functions.

- The function to display the ladder diagram on the screen and output it on an external printer, which is operated using the menu numbers 03 and F4 or 04 to F10.
- The function to input/output the ROM formatted program and make its comparison, which is operated combining the menu numbers 03, 05 or 07 and F2, F7, F8, F9, F13 or F14.
- Ladder diagram direct editing, which is operated by pressing <R1> key on the R key menu screen and executing ladder diagram direct editing.

The SYSTEM P Mate has less memory than the SYSTEM P Mark II and cannot load the system program on the system floppy disk at a time. The remainder left unloaded will be loaded automatically when each of the functions above is used. However, only in the case the system floppy disk has not been installed into the drive, the message "MOUNT SYSTEM FLOPPY DISK" is displayed as follows:

```
  SET SYSTEM FD & KEY I 'OK' OR 'NO'
FDD =OK ODRIVEJ (VOL =01)
FDD =
```

Install the system floppy disk into the drive #0 or #1 and key in 'OK 0' or 'OK 1'. If the system floppy disk is installed into the drive #0, it is possible to key in only 'OK' without specifying the drive number.

4.2.3 Loading of Floppy

FAPT LADDER system programs are loaded into the floppy. Also, sequence programs can be written from SYSTEM P series into the floppy or input from the floppy.

The loading method of the floppy is described in detail in the operator's manual for SYSTEM P series.

The following describes the loading direction of the floppy.

Loading direction of floppy

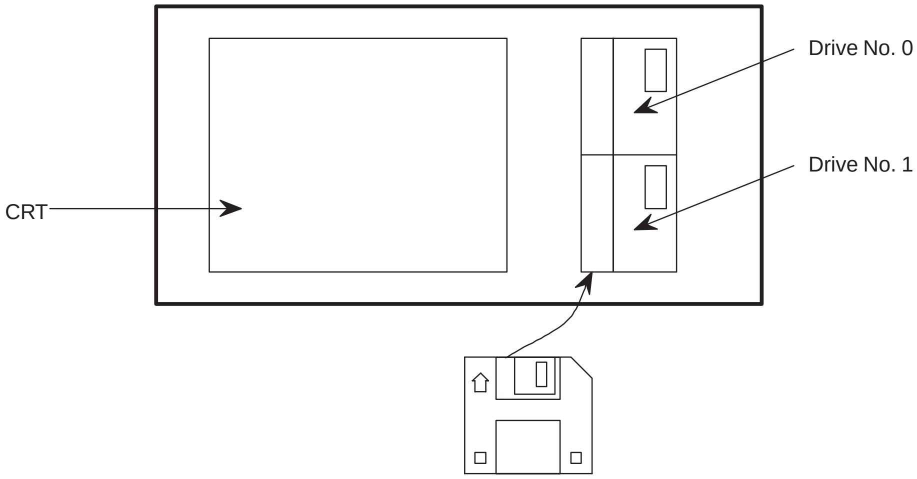


Fig. 4.2.3 Loading direction of floppy

4.2.4 FAPT LADDER System Floppy Loading

(1) Turn on the SYSTEM P series power supply.

(2) Set the system floppy or prepared exclusive system floppy into the floppy disk.

(3) Continue depressing <LOAD> key for 2 to 3 seconds on the keyboard.

(4) The system loading is started. After this system loading, ”FAPT LADDER” is displayed on the CRT screen and R key menu also appears.

This R key menu screen is shown in Fig. 4.2.6.

After this screen is displayed, take out system floppy or exclusive system floppy.

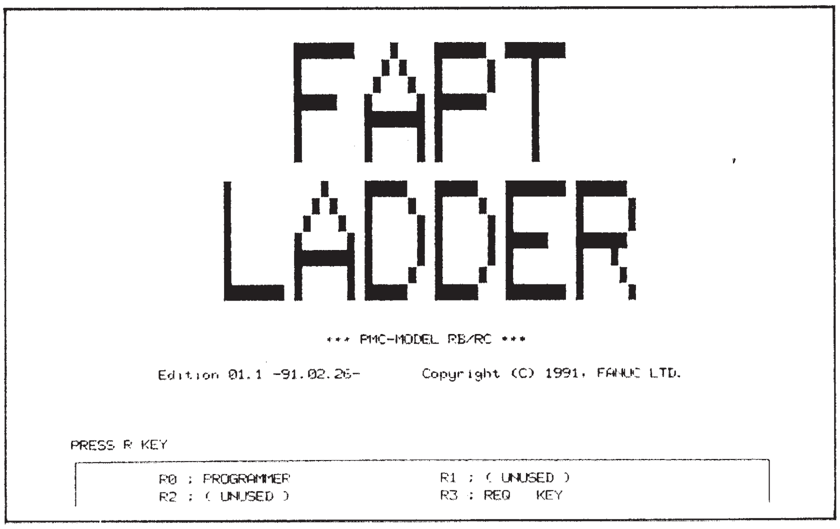


Fig. 4.2.4 R key menu screen

4.2.5 Programmer Menu Screen

A programmer menu screen (hereinafter called menu screen) is displayed by pressing <R0> key from the R key menu screen. Key in a menu number to be executed Fig. 4.2.5 shows the menu screen.

The parameter setting screen is displayed by pressing <R0> key just after loading the 1/2 system floppy.

Set parameters as required, referring to 4.2.6.

Proceed to the menu screen by pressing <NL> key.

Parameters are displayable and settable from the menu screen, too.

Programmer menu screen (The programmer menu and function keys are displayed.)

The following figure shows the screen to be displayed when the programmer key (R0) is pressed.

The programmer menu, function keys with I/O indication, and statuses are displayed on this screen.

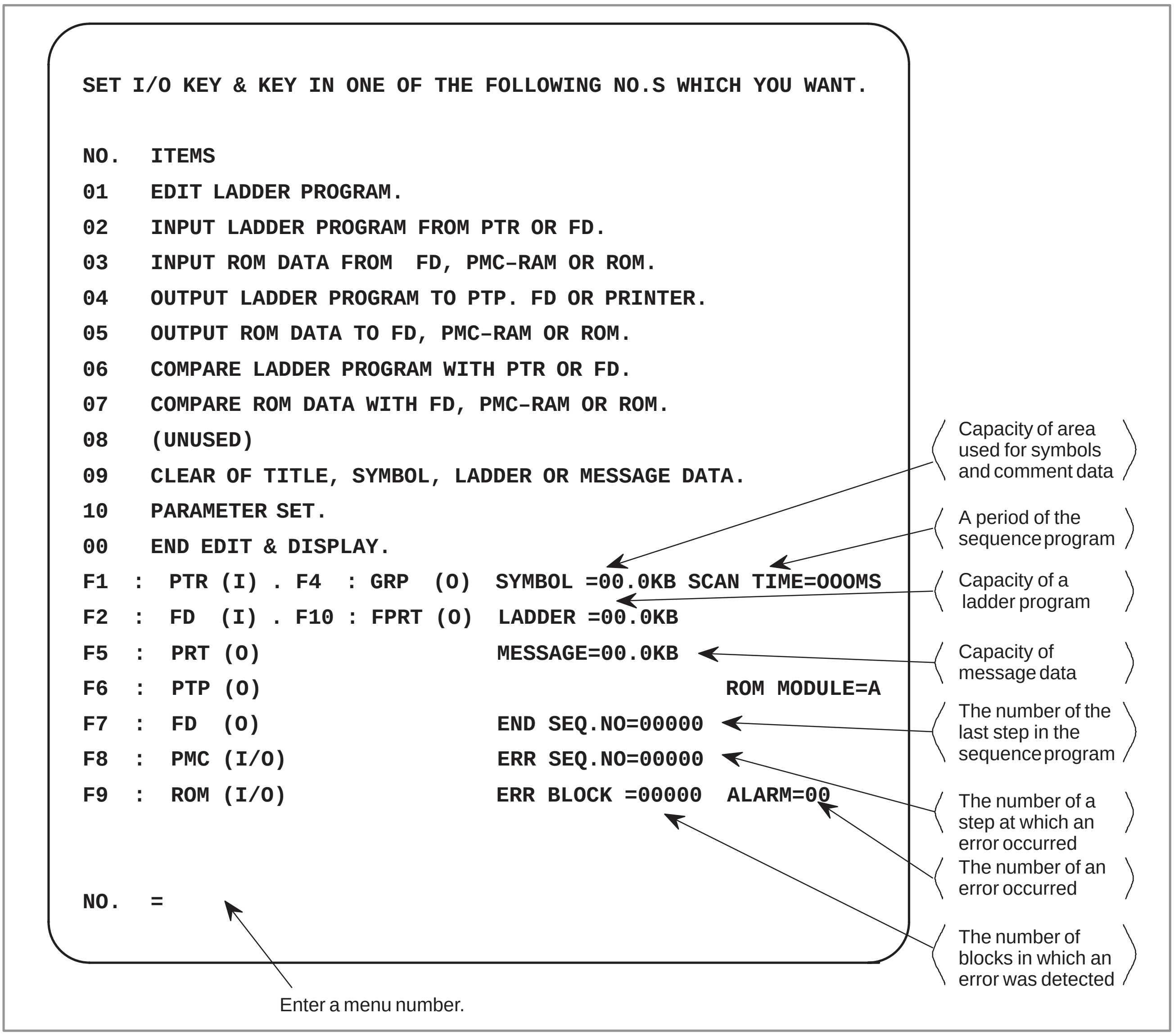


Fig. 4.2.5 Programmer menu screen

4.2.6 Parameter Setting and Display

Set parameters before inputting a sequence program without fail. Set necessary parameters by changing from the menu screen to the parameter setting screen (Fig.4.2.6), provided that the parameter setting screen is automatically displayed just after loading the system floppy.

(Operation in step **1** is not required in the procedure below.)

1 Key in menu number “10 <NL>” from the menu screen.

Turn off all F keys. The screen is switched, and the parameter setting screen shown in Fig. 4.2.6 is displayed.
The initial value of each parameter is as shown in Fig. 4.2.6.

```
KEY IN ONE OF THE FOLLOWING NO.S WHICH YOU WANT TO SET PARA,S.

NO.  ITEMS                     CURRENT PARAMETERS

01   (UNUSED)                  ;
02   COUNTER DATA TYPE         ; BINARY
03   OPERATOR PANEL            ; NO
      KEY/LED ADDRESS          ;      /
      KEY/LED BIT IMAGE ADRS.  ;      /
04   PMC TYPE                  ; PMC–RC
05   LANGUAGE ORIGIN           ; 000000H
06   (UNUSED)                  ;
07   LADDER EXEC.              ; 100%      (1–150%)
08   (UNUSED)                  ;
09   IGNORE DIVIDE CODE        ; NO
10   (UNUSED)                  ;
00   NOTHING TO SET
                               ; ROM WRITER=FA WRITER
NO.=
```

Fig. 4.2.6 Parameter setting screen (PMC–RC)

2 Key in “00 <NL>” to proceed to the menu screen, if displayed parameters are employed as they are.

3 Set parameters according to the following procedure when it is necessary to change the displayed parameters.

No operation is required for an item in which no change is required.

a) Set a counter data type. The initial value is set to the binary format.

1 Key in “02 <NL>”

2 Select a binary or BCD notation, and key in the corresponding number “@@<NL>”.

b) OPERATOR PANEL

Specifies whether the operator’s panel is used.

The initial value is already set to NO (unused).

1 Enter 03 and press the <NL> key.

2 The following message appears at the lower left of the screen.

EXAMPLE 0:NO, 1:YES
OP PANEL=

3 To disable the operator’s panel, enter 0 and press the <NL> key. To enable the operator’s panel, enter 1 and press the <NL> key.

4 Selecting YES in step 3 displays the following message:

SET KEY/LED ADDRESS (KEY ADRS, LED ADRS.)
ADDR=

5 Enter a Y–address to specify the KEY address and a Y–address to specify the LED address. For example, enter X0,Y0 and press the <NL> key.

6 Entering data as shown above displays the following message:

SET KEY/LED IMAGE ADDRESS (KEY ADRS, LED ADRS.)
ADDR=

7 Enter addresses other than X– and F–addresses. For example, enter R0,R10 and press the <NL> key.

c) Select the type of PMC.

The initial value has been set to the PMC–SB or –SA1.

1 Type 04 and press the <NL> key.

2 The following message appears at the lower left of the screen:

EXAMPLE 0:PMC–SB, 1:PMC–SC
PMC TYPE=
or
EXAMPLE 0:PMC–SA1, 1:PMC–SA2
PMC TYPE=

3 To select the PMC–SB or –SA1, enter 0 and press the <NL> key. To select the PMC–SC or –SA2, enter 1 and press the <NL> key.

4 When the type of PMC is changed, all data items including ladder data are cleared. The following message is displayed for confirmation:

CLEAR ALL DATA TO CHANGE PMC TYPE (0:NO, 1:YES)
CLEAR/KEEP=

5 To cancel changing the type of the PMC, enter 0 and press the <NL> key. To change the type of the PMC, enter 1 and press the <NL> key.

d) LANGUAGE ORIGN (for PMC–SC only)

The initial value is already set to 0.

1 Enter 05 and press the <NL> key.

2 Enter @@@@@@ (hexadecimal) and press the <NL> key to specify the first address of the TCB in a C program.

e) LADDER EXEC (only for PMC–SC)

The parameter value for LADDER EXEC is fixed to 100% for the PMC–RB. For the PMC–RC, the parameter value can be set as follows.

1 key in “07 <NL>”.

2 Key in “@@@ <NL>” by numeric characters within a range of 1% to 150%.

After setting, key in “@@ <NL>” or “<NL>” to set the menu screen.

This parameter is not supported for PMC–SA1, PMC–SA2, PMC–SB or PMC–SB2.

f) IGNORE DIVIDE CODE (only for PMC–SB/SC)

It is possible to select whether to execute a ladder program by dividing it into smaller units or without dividing it.

This parameter can be specified as follows for PMC–SB and PMC–SC:

1 Enter 09 and press the <NL> key.

2 To execute the ladder program by dividing it into smaller units, enter 0 and press the <NL> key. To execute it without dividing it, enter 1 and press the <NL> key.

This parameter is not supported for PMC–SA1 or PMC–SA2.

The ladder program is always executed without being divided.

4.3 PROGRAM EDITING

4.3.1 Data Display and Setting (Title, Symbol, Ladder Program, Comment, Message, I/O Module)

Display the EDIT screen by keying in ”1 <NL>” from the menu screen. Press <R0> by necessary times until a desired screen appears from the title to I/O module. The screen is switched in the sequence shown in Fig. 4.3.1, each time <R0> key is pressed.

Individual screens are reset to the menu screen by ”E <NL>”. In this paragraph, only the input and editing operation of each data from the keyboard is described.

For the I/O operations using a paper tape or a floppy, see 4.4 and 4.5.

(1) Title data (title data list screen).

Set the following data on sequence program as a comment.

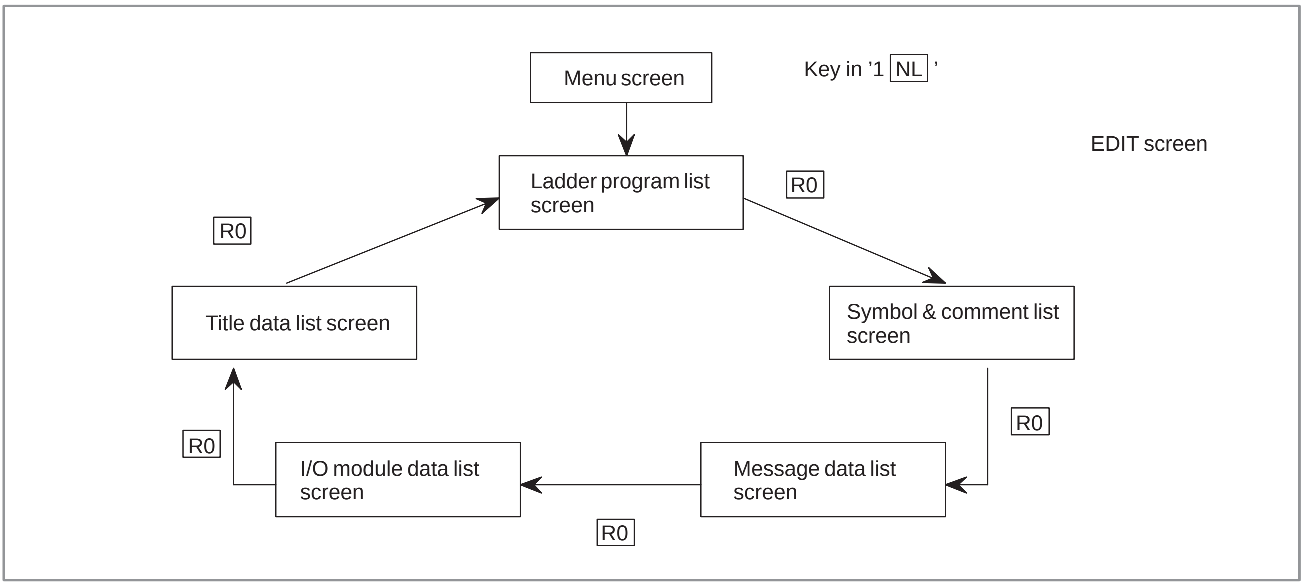


Fig. 4.3.1 (a) Switching sequence of data display screen

```
*** TITLE DATA LIST ***
01 MACHINE TOOL BUILDER NAME
02 MACHINE TOOL NAME
03 PMC & NC NAME               FANUC PMC–MODEL RB & F16MA
04 PMC PROGRAM NO.
05 EDITION NO.
06 PROGRAM DRAWING NO.
07 DATE OF PROGRAMMING
08 PROGRAM DESIGNED BY
09 ROM WRITTEN BY
10 REMARKS

   PMC CONTROL PROGRAM         SERIES : 4061 EDITION : 01

   MEMORY USED : 00.0 KBYTE SCAN TIME : 008 MSEC


0003 ALTERED
EDIT
```

Fig. 4.3.1 (b) Title data list screen

a) MACHINE TOOL BUILDER NAME

Set the machine tool builder name (max. 32 characters).

Key in “A1 @@@..........@@@ NL”.

Machine tool builder name to be set

Example) “A1 ***MACHINE(LTD) NL”

b) MACHINE TOOL NAME

Set the machine tool name (max. 32 characters).

Key in “A2 @@@..........@@@ NL”.

Machine tool name to be set

Example) ”A2 ***MACHINE NL”

c) CNC & PMC NAME

Set the CNC and PMC name (max. 32 characters).

Key in “A3 @@@..........@@@ NL”.

NC and PMC name to be set

Example) “A3 F16MA.&.PMC–N NL”

d) PMC PROGRAM NO.

Set the sequence program number (max. 4 characters).

Key in “A4 @@@@ NL”.

Number to be set

Example) “A4 0001 NL”

e) EDITION NO.

Set the edition number (max. 2 characters).

Key in “A5 @@ NL”.

Edition number to be set

Example) “A5 G NL”

f) PROGRAM DRAWING NO.

Set the sequence program drawing number (max. 32 characters).

Key in “A6 @@@...........@@@ NL”.

Drawing number to be set

Example) “A6 0001–0002–000A NL”

g) DATE OF PROGRAMMING

Set the sequence programming date (max. 16 characters).

Key in “A7 @@..........@@ NL”.

Date to be set

Example) “A7 1990.10.23 NL”

h) PROGRAM DESIGNED BY

Set the sequence program designer name (max. 32 characters).

Key in "A8 @@@..........@@@ NL".

Name to be set

Example) "A8 MR.***&MISS *** NL"

i) ROM WRITTEN BY

Set the name of the programmer who wrote a program into ROM cassette (max. 32 characters).

Key in "A9 @@@..........@@@ NL".

Name to be set

Example) "A9 MR.***&MISS *** NL"

j) REMARKS

Set remarks (memo) (max. 32 characters).

Key in "A10 @@..........@@ NL".

Remarks to be set

Example) "A10 MEMO–COMMENT NL"

Set title data about all items in the above format for both entry and alteration.

All characters are settable so long as they can be keyed in from the SYSTEM P series keyboard. Set easy–to–understand data about individual items.

(2) Symbol and comment data (SYMBOL & COMMENT LIST screen).

A symbol means a signal name to be attached to each PMC I/O signal. The comment data is a comment statement of the signal name.

The symbol is optionally settable within maximum 6 characters, while the comment data are optionally settable within maximum 30 characters.

a) Input from keyboard (Insert)

Key in "G0.1 SYMNAM COMMENT NL".

Comment data

Symbol name

Address

Mode selection(IS..., AS...) and line selection (I..., A...) need not be specified when symbols or comment data are input or edited. Also addresses (G, F, X, Y,...) can be entered in any sequence.

b) Alter

The operation is completely the same as that described in 2) a).

c) Insert

The operation is completely the same as that described in 2) a).

d) Delete

i) Delete every line

Key in "D@@@ NL".

@@@: Line number to be deleted

ii) Sequential delete

Key in "D@@@@,@@@ NL".

@@@ (second): Delete end line number

@@@@ (first): Delete start line number

e) Search

i) Search by line number

Key in "L@@@@ NL".

@@@@: Line number to be searched

ii) Search by symbol name

Key in "L@@@@ SYMNAM NL".

SYMNAM: Symbol name to be searched

@@@@: Line number with which the search is to be started

iii) Address search

Key in "L@@@@ F0.1 NL".

F0.1: Address to be searched

@@@@: Line number with which the search is to be started

(3) Message data (MESSAGE DATA LIST screen)

Message data are alarm and operator message data to be displayed by using functional instruction DISPB (SUB 41).

```
     ***MESSAGE DATA LIST***
0001  A00.0   0123456··········89XY  <-- Message data
0002          ABCD······
0003                                  <-- Line number
0004
0005
0006
0007
0008
0009  A00.1                           <-- Address
  .
  .
  .
EDIT =
```

Fig. 4.3.1 (c) Message data list screen

Maximum 255 characters are entered to one address as message data. Input message data every maximum 32 characters/line by dividing them into 8 lines.

a) Input and alter from keyboard

Set message data in the alter format for both entry and alter. All characters are settable so long as they can be keyed in from the SYSTEM P series keyboard.

Key in message data every line in the following format.

"A @@@ MESSAGE–DATA1 NL".

Message data (maximum 32 characters)

Line number (maximum 3 digits)

A means alter.

b) Delete

Delete message data every line in the following format.

Key in "D@@@ NL".

Line number to be deleted

c) Search

Search message data by address.

"A@@.@ NL"

Address of message data to be searched

(4) I/O module data (I/O MODULE DATA LIST screen)

I/O module data are used for determining addresses in a sequence program of each I/O module.

```
***I/O MODULE DATA LIST***
ADDRESS GROUP BASE SLOT NAME
X000
X001
X002
X003
 .
 .
 .
 .
```

Fig. 4.3.1 (d) I/O module data list screen

a) Input and alter from keyboard

Set I/O module data in the following format when inputting or altering them from the keyboard.
Key in I/O module data in the format of:

"@@@ @ @ @@ @@@@@ NL"

I/O module name (maximum 5 digits)

Slot number (maximum 2 digits within a range of 1 to 10)

Base number (1 digit within a range of 0 to 3)

Group number (1 digit within a range of 0 to 15)

Address (input X0, Y0, ...)

b) Delete

Delete I/O module data every address by specifying it as follows:

Key in "@@@@ NL".

Address of I/O module data to be deleted (input X0, Y0, ...)

NOTE

1. If the same slot number is specified when the group and base numbers are equal to each other, alarm No. 88 occurs.
2. If an output module is specified at an input address or an input module is specified at an output address, alarm No. 87 occurs.
3. If a module is set doubly to a preset address, alarm No. 81 occurs.

Example) When two IO modules, b are set as shown in the following figure;

address		group	base	slot	name
	a.				
X000		2	0	1	FS08A
X001		2	0	1	FS08A
X002		2	0	1	FS08A
X003		2	0	1	FS08A
X004		2	0	1	FS08A
X005		2	0	1	FS08A
X006		2	0	1	FS08A
X007		2	0	1	FS08A
X008	b.				
X009		2	1	8	ID16C
X010		2	1	8	ID16C
X011					
X012					
X013					
X014	c.	2	1	8	ID16C
X015		2	1	8	ID16C

If an attempt is made to set a module to X014 like c., alarm No. 88 occurs due to the reason in 1).

d.

If an attempt is made to set an output module to X006 like d., alarm No. 87 occurs due to the reason in 2).

If an attempt is made to set input module ID32B to X006 like d., alarm No. 81 occurs due to the reason in 3). In this case, this input module must be set after deleting a. and b. modules once.

The module names (FS08A, CT01A, etc.) used for input and output in common are out of the objects of check in 1) and 2).

4.3.2 Programming from Keyboard

Input a sequence program from the keyboard.

Set the EDIT screen (LADDER PROGRAM LIST screen).

Press menu number ”1 <NL>” on the menu screen, or press <R0> key on the symbol or I/O module screen. Turn off all F keys at this time.

Key in ”IS0 <NL>” (Insert Succession) to set the sequential insert mode, and then, input a sequence program.

”*IS MODE*” is displayed at the lower right part of the screen. key in desired instructions sequentially in the following format.

(Key in sequence)

1 IS0 <NL> (Sequential input start command)

→ *IS MODE* is displayed at the lower right part of the screen.

2 R X0.1 <NL>

3 W R1.1 <NL>

4 IE <NL> (Sequential input end command)

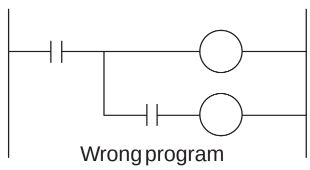
→ *IS MODE* display disappears from the lower right part of the screen.

NOTE

1 Instructions to be keyed in are entered by abbreviated symbols as shown above for the purpose of preventing a key–in failure and improving the operability by reducing the number of key–in times. It is also allowable to input these instructions by using their full names, like ”RD X0.1 <NL>”. Table 4.3.2 shows the correspondence between abbreviated symbols and full names.

2 No severe format checking is performed for mnemonic program. For example, the following program may be correct with mnemonic programming.
However this program cannot be displayed as ladder diagram nor printed out on the printer.

Usually do not program as shown below:



Wrong program

Table 4.3.2 Keyboard input format and screen display format

Input format from keyboard (Simple symbol)	Display format on screen (Full name)
R X0. 1	RD X0. 1
RN X0. 2	RD. NOT X0. 2
RNS X0. 3	RD.NOT.STK X0. 3
W R0. 4	WRT R0. 4
WN R0. 5	WRT.NOT R0.5
O Y1. 0	OR Y1. 0
ON Y1. 1	OR.NOT Y1.1
OS	OR. STK
AG2. 0	AND G2. 0
AN G2. 1	AND.NOT G2. 1
AS	AND. STK
T 5	TMR 5
D F0	DEC F0
S 5	SUB 5
P 1234	(Parameter)

4.3.3 Alter

Correct a generated sequence program by alter operation.

Set the EDIT screen (LADDER PROGRAM LIST screen) first and display the generated source program. Turn off all F keys at this time.

a) Alter every instruction

Key in "A@@@@@ R X0.1 NL"

Instruction to be altered

Line number of the instruction to be altered (maximum 5 digits)

A means alter.

b) Sequential alter

i) Key in "AS@@@@@ NL" (Alter Succession) to set the sequential alter mode.

@@@@@: Line number to be sequentially altered (maximum 5 digits) "AS MODE" is displayed at the lower right part of the screen.

ii) Instructions are sequentially altered starting with the Line specified by @@@@@, each time the key–in operation is done in the "R X0.1 <NL>" format.

iii) After sequential alter, key in "AE <NL>" (Alter End).

Example) Example of sequential alter of sequence program
For altering all step numbers 20 to 23;

(Key in sequence)

1 AS20 <NL> (Sequential alter start command)
→ *AS MODE* is displayed at the lower right part of screen.

00020 RD Y0.1 **2** R Y0.1 <NL>

00021 WRT R0.1 **3** W R0.1 <NL>
00022 RD F1.1 **4** R Y1.2 <NL>
00023 WRT R1.1 **5** W R1.2 <NL>
6 AE <NL> (Sequential alter end command)
→ *AS MODE* display disappears from the lower right part of the screen.

c) Wiring change function

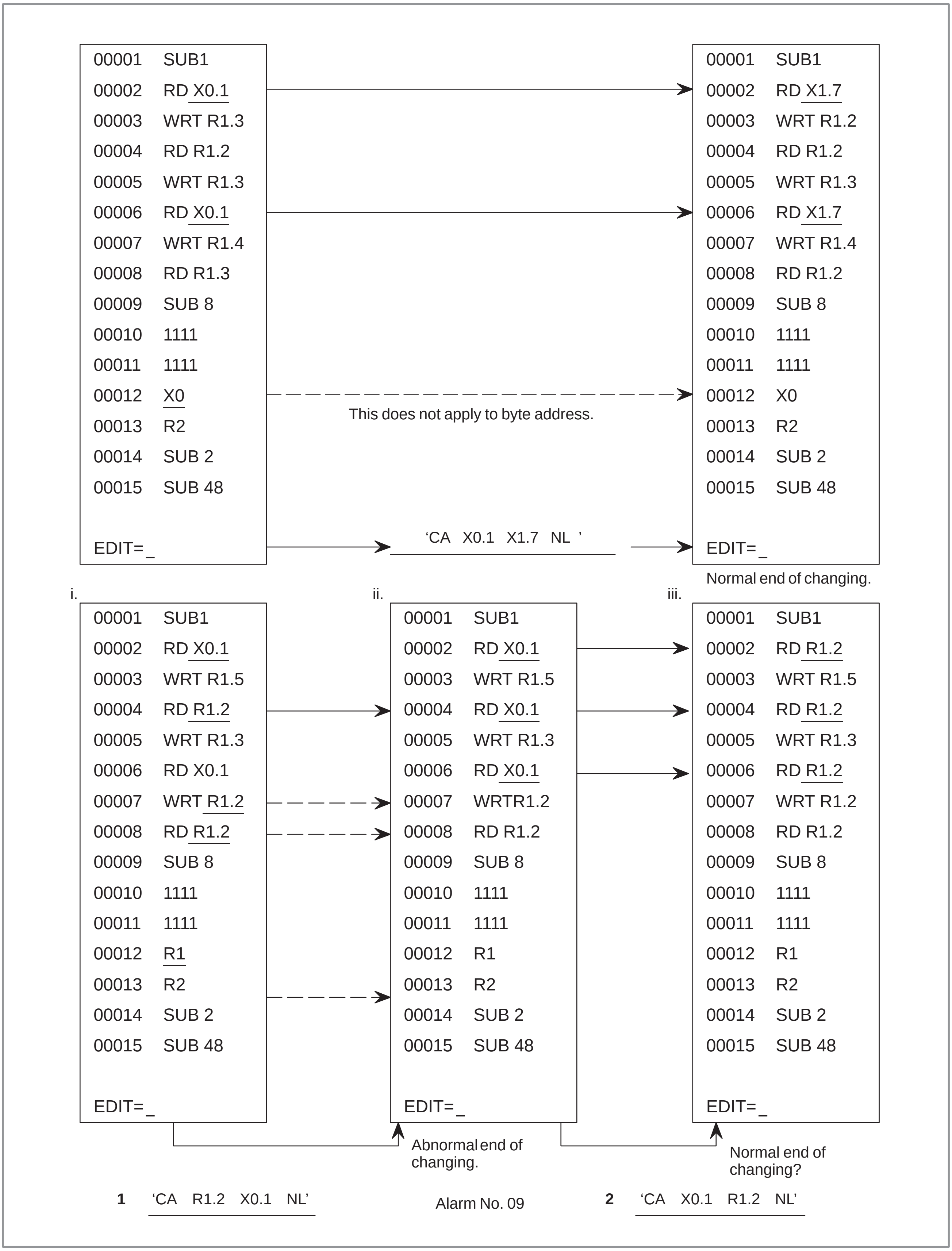
All of address used in Ladder Program is changed to a new address independently of a command. Only bit address can be changed.

Type in ’CA Address 1 Address 2 <NL>

(Symbol can not be changed)
A new address
A previous address to be changed
Abbreviation of CHANGE ALL

Example) ’CA R0.1 R1.2 <NL>’
— All ”R0.1” used in Ladder Program is changed to ”R1.2”.

> **NOTE**
> If an address is specified which can not be changed to a new address, an alarm 09 occurs when the specified line will be changed. In that case, previous lines correctly changed to that line can be acceptable.



As shown above, an alarm No. 09 occurs when a ladder program i is changed by an operation of **1** and a ladder program ii will be produced.

Then, it may be impossible to return a ladder program ii to a ladder program i by an operation **2**.

(Special use of wiring change function)

All address used in ladder program of specified line number of subsequent, is changed a new address independently of a command.

Only bit address can be changed.

Operate carefully with enough recognition of above.

’C@@@@@ Address 1 Address 2 NL’

- @@@@@: Change start line number
- Address 1: A previous address to be changed
- Address 2: A new address (Symbol cannot be changed)

Example) ’C7 R0.1 R1.2 <NL>’

— All ”R0.1” used in ladder program of 7th line or subsequent, are changed to ”R1.2”.

4.3.4 Insert

Insert a new program to the generated sequence program.

Set the EDIT screen (LADDER PROGRAM LIST screen) first.

a) Insert every instruction

Key in “I@@@@@ R X0.1 NL”.

- I: I means insert.
- @@@@@: Line number just before the instruction to be inserted (maximum 5 digits)
- R X0.1: Instruction to be inserted

b) Sequential insert

i) Key in ”IS@@@@@ <NL>” (Insert Succession) to set the sequential insert mode, and *IS MODE* is displayed at the lower right part of the screen.

@@@@@:Line number just before the instruction to be inserted (maximum 5 digits)

ii) Instructions are sequentially inserted starting with the line next to the line specified by @@@@@, each time the key–in operation is done in the ”R X0.1 NL” format.

iii) After sequential insert, key in ”IE <NL>” (Insert End).

Example) Sequential insert of a sequence program
For inserting multiple instructions after step number 20

(Key in sequence)

1 IS20 <NL> (Sequential insert start command)
→ *IS MODE* is displayed at the lower right part of screen.

00020 RD Y200.0
00021 WRT R300.7

2 R.S R200.1 <NL>
3 R.S R200.2 <NL>
4 R 5 <NL>
5 P 9 <NL>
6 IE <NL> (Sequential insert end command)
→ *IS MODE* display disappears from the lower right part of the screen.

4.3.5 Delete

i) Delete every instruction

Key in "D@@@@@ NL".

Line number to be deleted (maximum 5 digits)

D means delete.

ii) Sequential delete

Key in "D@@@@@,@@@@@ NL".

Line number to complete delete (maximum 5 digits)

Line number to start delete

4.3.6 Location Search

Search a sequence number by a line number or instruction

i) Search by line number

Key in "L@@@@@ NL".

Line number to be searched (maximum 5 digits)

L means location search.

ii) Search by instruction (Search by address)

Key in "L@@@@@ R X0.1 NL".

Instruction to be searched

Line number with which the search is to be started

NOTE
Input data after changing the symbol display into address display by passing R1 key, if the address of the instruction to be searched is defined by a symbol and displayed by the symbol.

iii) Search by instruction (Search by symbol)

Key in “L@@@@@ R ACT NL”.

- L@@@@@ : Line number with which the search is to be started
- R ACT : Instruction to be searched (ACT: Symbol name)

NOTE
This search applies to such a case as the address of the instruction to be searched is defined by a symbol and the symbol is displayed.

iv) Search by the bit address or its symbol name

The specified address (only bit address) or its symbol name is searched from the specified line number independently of a command.

Type in ’L@@@@ ????? NL’.

- L@@@@ : Search start line number
- ????? : Bit address or its symbol name

Example) ’L1 R1.0 NL’
— Start searching bit address ”R1.0” from 1st line.
’L7 SMB NL’
— Start searching symbol name ”SM BL” defined at bit address from 7th line.

v) Continuous search

A specified command, address (only bit address) or its symbol name is searched from 2nd line displayed on the screen.

Type in ’L ????? NL’.

- ????? : Command, bit address or its symbol name to be searched

Type in ’F NL’.

- F — FIND: Search the same command, bit address or its symbol name as that searched just before, from 2nd line displayed on the screen.

Example) ’L R R0.1 <NL>’
— Search the command ”RD R0.1” from 2nd line displayed on the screen.
’L R0.1 <NL>’
— Search the bit address ”R0.1” from 2nd line displayed on the screen.
’L SYMBOL <NL>’
— Search the symbol name ”SYMBOL” defined at bit address from 2nd line displayed on the screen.

4.3.7 Display of Ladder Diagram

The ladder diagram can be displayed on the programmer function EDIT screen.

Set the screen to EDIT screen (LADDER PROGRAM LIST)

a) Turn on F4 key.

b) Depress <NL> key

The ladder diagram is displayed on the screen.

For displaying the sequence program in the mnemonic format from the ladder diagram, turn off F4 key, and depress <NL> key.

The ladder diagram at an optional point can be displayed by the step number search or instruction search method.

If a ladder diagram cannot be displayed on one screen, it can be displayed by scrolling it leftward, rightward, upward, and downward as shown in the following table.

c) Edition during LADDER diagram display

Sequence programs can be edited even on the LADDER diagram screen display, (This function is convenience when sequence programs are edited with seeing LADDER diagram print out list.) From ’EDIT=’ in the LADDER diagram screen display, sequence programs can be edited by the same operation as in editing programs in the ’LADDER PROGRAM LIST’ screen.

Scroll direction		Key
Left	(Left ladder on screen is displayed.)	← 4 NL
Right	(Right ladder on screen is displayed.)	6 → NL
Upper	(Upper ladder on the screen is displayed.)	↑ 8 NL
Lower	(Lower ladder on the screen is displayed.)	2 ↓ NL
Upper half page		R2 NL
Lower half page		R2 NL

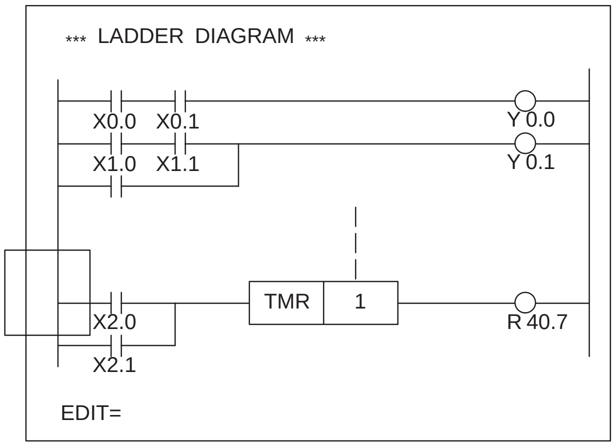
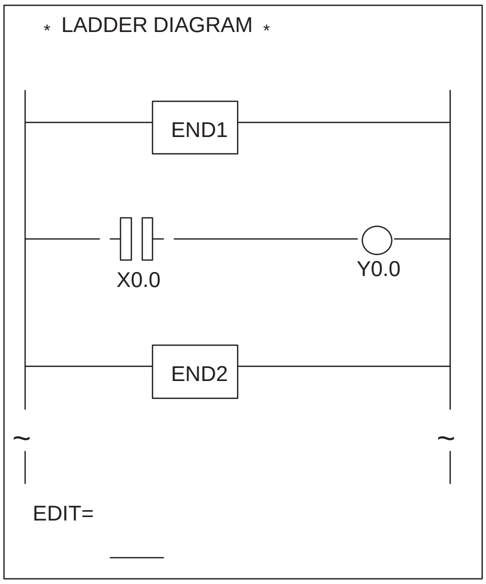


Fig. 4.3.7 Ladder diagram display screen

Example)



1 From ’EDIT=’ IN the left figure, key in as follows.
’ISO NL’
’R X1.0 NL’
’W Y1.0 NL’
’IE NL’

2 Press R0 key to display the ’LADDER PROGRAM LIST’ screen.
Two lines ’RDX1.0’ AND ’WRT Y1.0’ are added before ’SUB1 (END1)’.

3 When the LADDER diagram is displayed again, the diagram after adding the above two lines is displayed.

4.3.8 Help Screen

Editing command explanation screen can be displayed from Ladder program edition screen. (LADDER PROGRAM LIST or LADDER DIAGRAM)

key in ’H <NL>’ to display the following screen.

```
  *** HELP LIST ***
                                            (@@@@@=SEQUENCE NO.)
<INSERT>
  I@@@@@ OPERATION CODE           : INSERT
  IS@@@@@                         : INSERT SUCCESSION START
  IE                              : INSERT SUCCESSION END
<ALTER>
  A@@@@@ OPERATION CODE           : ALTER
  AS@@@@@                         : ALTER SUCCESSION START
  AE                              : ALTER SUCCESSION END
<DELETE>
  D@@@@(,@@@@@)                   : DELETE (SUCCESSION END)
<LOCATION SEACH>
  L@@@@@                          : SEQUENCE NO. SEARCH
  L@@@@@ OPERATION CODE           : OPERATION CODE SEARCH
  L@@@@@ <ADDRESS OR SYMBOL>      : ADDRESS OR SYMBOL SEARCH
  L OPERATION CODE                : SEARCH FROM DISPLAY 2ND LINE
  L <ADDRESS OR SYMBOL>           : SEARCH FROM DISPLAY 2ND LINE
  F                               : FIND FROM DISPLAY 2ND LINE
<CHANGE ALL ADDRESS>
  CA ADDRESS1 ADDRESS2            :CHANGE ALL ADDRESS1 TO ADDRESS2
```

Key in “<NL>” to return it to ladder program editing screen.

4.3.9 Editing end

Key is “E <NL>” (End) after editing a sequence program, and the EDIT screen is reset to the menu screen.

4.4 INPUT OF PROGRAM

4.4.1 Source Program

Read source programs (parameters, titles, symbols, ladders, messages, and I/O modules) from an input unit designated by an F key on the menu screen, and load them into SYSTEM P series memory.

(1) Paper tape format of source programs

Paper tape format of source programs is of ISO code. No EIA code paper tape can be used.

a) Parameter date

Feed % @0 CR ±±±±± % CR Feed

b) Title date

Feed % @1 CR 1 MACHINE TOOL CR ±±±±± % CR Feed

c) Symbol date

Feed % @2 CR F0.1 SYMBOL COMMENT CR ±±±± % CR Feed

d) Ladder program

Feed % @3 CR RD X0.1 CR ±± WRT WORK01 CR ±± % CR Feed

e) Message date

Feed % @4 CR A0.0 MESSAGE = 1 CR ±± A24.7 MESSAGE = 24 CR %
WRT WORK01 CR

CR ~ Feed

f) I/O module date

Feed % @5 CR X 0 0 0 5 ID32C CR ±±±±± % CR Feed

(2) Input method from PPR reader

1 Turn on F1 key.

2 Key in menu number “2 <NL>”.

3 The screen is switched, and the entry of a source program is started.

4 After the source program has been normally entered, the screen is automatically reset to the programmer menu screen. If an error was detected during entry, “PART–” is displayed on the lower left part of the screen. Check error contents, and key in “E NL”. The screen is reset to the programmer menu screen.

(3) Entry method from floppy

1 Turn on F2 key.

2 Key in menu number “2 <NL>”.

3 The following message is displayed at the lower part of the screen.

SET FD & KEY IN “OK” “KILL” OR “NO”
FD0=OK <DRIVE> <@NAME OR : NUMBER>
FD0=

Fig. 4.4.1 shows the menu screen in the floppy entry mode.

[Screen when source programs are input form floppy]

```
SET I/O KEY & KEY IN ONE OF THE FOLLOWING NO.S WHICH YOU WANT.

NO.ITEMS
01 EDIT LADDER PROGRAM.
02 INPUT LADDER PROGRAM FROM PTR OR FD.
03 INPUT ROM DATA FROM FD. PMC–RAM OR ROM.
04 OUTPUT LADDER PROGRAM TO PTP. FD OR PRINTER.
05 OUTPUT ROM DATA TO FD. PMC–RAM OR ROM.
06 COMPARE LADDER PROGRAM WITH PTR OR FD.
07 COMPARE ROM DATA WITH FD. PMC–RAM OR ROM.
08 (UNUSED)
09 CLEAR OF TITLE. SYMBOL. LADDER OR MESSAGE DATA.
10 PARAMETER SET.
00 END EDIT & DISPLAY.
F1  :  PTR (I) . F4  : GRP  (O)   SYMBOL =00.0KB SCAN TIME–OO8MS
F2  :  FD  (I) . F10 : EPRT (O)   LADDER =00.0KB
F5  :  PRT (O)                    MESSAGE=00.0KB
F6  :  PTP (O)
F7  :  FD  (O)                    END SEQ.NO=00000
F8  :  PMC (I/O)                  ERR SEQ.NO=00000
F9  :  ROM (I/O)                  ERR BLOCK =00000
SET FD & KEY IN ’OK’   ’KILL’ OR ’NO’
FD0 = OK ’@FILE NAME’
         <DRIVE><@NAME OR NUMBER>
NO.=
```

Example

Key in file names to be input from floppy as shown in the example.

Fig. 4.4.1 Floppy input menu screen

4 Insert the floppy into the disk, and enter the following data. Characters in < > need not be keyed in.

OK @LADDER1 NL

File name (provisional file name)

5 The screen is switched, and the entry of source programs is started from the floppy.

6 The following procedure is the same as in 4.4.1 2) **4**.

7 A file name is inputtable up to maximum 17 characters. All characters on the SYSTEM P series keyboard are employable for this entry. The kinds of capitals are not limited.

"@" (at mark) shows a file name input identifier. Key in it just before the file name as shown in example $$ without fail.

CAUTION

If sequence program instructions are sequentially entered while a sequence program is loaded in the SYSTEM P series memory, the instructions are entered into the SYSTEM P series memory following the previously loaded program. Clear SYSTEM P series memory, if a new program is entered from the floppy. (see 4.7)
The SYSTEM P series memory is cleared by turning off the SYSTEM P series power supply.

4.4.2 ROM Format Program

(1) Transfer of sequence program from the PMC–SA1/SA2/SB/SC

The created sequence program is transferred from the PMC–SA1/SA2/SB/SC.

First, connect the SYSTEM P Series and the CNC with a Reader/Puncher interface cable. (Refer to Appendix 1 for details of the cable.) For the method and location of connection, refer to the section "3.2 Configuration devices and their connection". In the following procedure, operations **1** to **6** are NC side operations.

The keys enclosed in [ ] are soft keys.

1 Pressing soft keys [SYSTEM] and [PMC] displays the PMC screen. Steps **2** to **4** below must be performed when [I/O] is not displayed on the PMC screen. For a 9–inch CRT, press soft key [NEXT] to check that [I/O] is not on the screen.

2 Pressing soft keys [PMCPRM] and [KEEPRL] on the PMC screen displays the keep relay setting screen.

3 Set K17.1 to 1 on the keep relay setting screen.

4 Pressing soft key [RETURN] displays the PMC screen.

5 On the PMC screen, pressing soft key [I/O] displays the I/O screen. For a 9–inch CRT, press soft key [NEXT] before pressing soft key [I/O].

6 Pressing soft key [EXEC] on the I/O screen puts the system in the EXECUTING state.

7 Turn on the F8 key on the SYSTEM P series menu screen. (Turn on the F12 key at the same time when the C–language program is included.)

8 If the menu number '3 [NL]' is keyed in, the message shown below will be displayed. PMC–SA1/SA2/SB/SC is not displayed. Key in the type of ROM module to be used from now on. (Refer to Note 1 when selecting ROM module B, C or D.)

SELECT THE TYPE OF ROM MODULE ACCORDING
TO THE FOLLOWING NO.
ROM MODULE 0:A 1:B, 2:C, 3:D
NO.=

By means of the above–described operations, the program transfer is started. The transfer screen is displayed on the SYSTEM P Series screen and the transfer counter counts. The screen returns to the menu screen after the end of transfer.

(2) Input from a floppy disk

1 Turn on the F2 key. (Turn on the F12 key at the same time when the C language program is included.)

2 If the menu number ‘3 [NL]’ is keyed in, the message shown below will be displayed. PMC–SA1/SA2/SB/SC is not displayed. Key in the type of ROM module to be used from now on. (Refer to Note 1 when selecting ROM module B, C or D.)

SELECT THE TYPE OF ROM MODULE ACCORDING
TO THE FOLLOWING NO.
ROM MODULE 0:A 1:B, 2:C, 3:D
NO.=

3 The following message is displayed at the lower left part of the screen.

SET FD & KEY IN ‘OK’,‘KILL’ OR ‘NO’
FD=OK <@FILE NAME>
FD0=OK <DRIVE><@NAME OR : NUMBER>
FD0=

4 Insert the floppy into the disk and enter the following data

5 ’OK @LADDER2 [NL]

File name

6 The screen is switched and the ROM format program is started from the floppy disk.

7 After reading is ended, the screen is automatically changed to the program menu screen if no problem occurs. When an error is detected during reading, ’PART’ = is displayed on the left lower part of the screen. Check the error and key in ’E [NL]’ to return the screen to the program menu screen.

(3) Method of inputting from the FA writer and PMC writer

1 Check the setting of the ROM writer. (See Section 3.4, ”Setting of I/O Device.”)

2 Put the FA Writer in the REMOTE mode by the [REMOTE/LOCAL] key before using it.

3 Turn on the F9 key. (Turn on the F12 key at the same time when the C language program is included.)

4 If the menu number ‘3 <NL>’ is keyed in, the message shown below will be displayed. PMC–SA1/SA2/SB/SC is not displayed. Key in the type of ROM module to be used from now on. (Refer to Note 1 when selecting ROM module B, C or D.)

SELECT THE TYPE OF ROM MODULE ACCORDING
TO THE FOLLOWING NO.
ROM MODULE 0:A 1:B, 2:C, 3:D
NO.=

5 The screen is switched and the message shown below is displayed.

SET EPROM OR ROM MODULE & KEY IN ‘OK’ OR ‘NO’
KEY IN=

6 Check the above message. For the PMC–SA1/SA2/SB, insert the EPROM for the PMC into the FA Writer or PMC Writer. For the PMC–RC, insert the ROM module for the PMC into the FA Writer or PMC Writer. Note, however, that ROM modules are not available with the PMC Writer.

7 Key in ’OK <NL>’ or ’NO <NL>’.
When ’OK <NL>’ is keyed in, the sequence program written into the EPROM and ROM module for PMC is entered into P–G memory.
The screen returns to the menu screen if it ends with no problems occurring.
When ’NO <NL>’ is keyed in, the screen returns to the menu screen.

NOTE
When using the SYSTEM P Mate, if ROM module B, C or D is selected, overlay occurs. When cassette B or C is selected, set the work floppy disk for external memory in drive 1.

4.5 OUTPUT OF PROGRAM

4.5.1 Source Program

By selecting ’04 <NL>’ (OUTPUT LADDER PROGRAM) from menu no.4, the following detail menu is displayed.

```
SET I/O KEY & KEY IN ONE OF THE FOLLOWING NO.S WHICH YOU WANT.

NO.  ITEMS
01  OUTPUT ALL DATA.
02  OUTPUT SYSTEM PARAMETER.
03  OUTPUT TITLE DATA
04  OUTPUT SYMBOL DATA.
05  OUTPUT MESSAGE DATA.
06  OUTPUT I/O MODUL DATA.
07  OUTPUT LADDER PROGRAM (MNEMONIC).
08  OUTPUT LADDER DIAGRAM (ONLY FANUC PRINTER).
09  OUTPUT CROSS REFERENCE (SEQUENCE NO.)
00  END
F5 : PRT (O)  ,  F10 : FANUC PRINTER (O)
F6 : PTP (O)  ,  F13 : CROSS REFERENCE (NO.8)
F7 : FD  (O)



NO. =
```

Select a desired data and device from the above details menu screen by combining the menu numbers and F keys.

(1) OUTPUT ALL DATA

All data of system parameters, titles, symbols, messages, I/O modules and ladder programs (source format) are output to a device specified by an F key.
Turn on an F key corresponding to the device to be output, and key in detail menu number ’01 <NL>’.
If F10 key is turned on, all data are output to the FANUC printer (external printer) and the ladder diagram is output last. If F13 key is turned on furthermore, the ladder diagram is output with a cross reference.

(2) OUTPUT SYSTEM PARAMETER

System parameter data are output to a device specified by an F key.
Turn on an F key corresponding to the device to be output, and key in detail menu number ’02 <NL>’.

(3) OUTPUT TITLE, DATA

Title data are output to device specified by an F key.
Turn on an F key corresponding to a device to be output, and key in detail menu No. ’03 <NL>’.

(4) OUTPUT SYMBOL DATA

Symbol data are output to device specified by an F key.
Turn on an F key corresponding to a device to be output and key in detail menu number ’04 <NL>’.
The screen is switched and the following display appears.

    OUTPUT = ’L@@@@ (,@@@@)’
    OUTPUT =_

Specify the output range by line numbers as follows.

Example)

Key in ’L1, 100 NL’

Output end liner number (If this parameter is omitted, data are output to the last one.)

Output start line number

(5) OUTPUT MESSAGE DATA

Message data are output to a device specified by an F key.
Turn on an F key corresponding to a device to be output, and key in detail menu number ’05 <NL>’.
The screen is switched and the following display appears.

OUTPUT = ’A@@.@ (,@@.@)’
OUTPUT =_

Specify the output range by addresses as follows.

Example)

Key in ’A1.0,10.1’

Output end address (If this parameter is omitted, data are output to the last one.)

Output start address

(6) OUTPUT I/O MODULE DATA

I/O module data are output to a device specified by an F key.
Turn on an F key corresponding to a device to be output, and key in detail menu number ’06 <NL>’.

(7) OUTPUT LADDER PROGRAM (MNEMONIC)

Ladder program (source format) data are output to a device specified by an F key.
Turn on an F key corresponding to a device to be output, and key in detail menu number ’07 <NL>’.

The screen is switched and the following display appears.

OUTPUT = ’L@@@@ (,@@@@)’
OUTPUT =_

Specify the output range by line numbers as follows.

Example)

Key in ’L1,100 NL’

Output end line number (If this parameter is omitted, data are output to the last one.)

Output start line number

(8) OUTPUT LADDER DIAGRAM (ONLY FANUC PRINTER)

A ladder diagram is output to the FANUC printer (external printer).
Key in detail menu number ’08 <NL>’, and then, turn on F10 key.
Turn on F13 key furthermore, if it is desired to output the ladder diagram with a cross reference.
The screen is switched and the following display appears.

OUTPUT = ’L@@@@ (,@@@@)’
OUTPUT =_

Specify the output range by line numbers as follows.
(Partial output is also possible.)

Example)

Key in ’L1,100 <NL>’

Output end line number (If this parameter is omitted, data are output to the last one.)

Output start line number

```
  ITEMS

* ALL ADDRESS              ’ALL’
* HEAD CHARACTER           ’G’
* ADDRESS’G14.6’
* ADDRESS TO ADDRESS       ’G14.6,R142.5’
* ADDRESS TO END           ’G14.6–END’

* END ; PUSH ’NL’ KEY

 ADDR=
```

NOTE

- It takes time more or less from the end of operation on end to the start of printer operation when outputting the LADDER diagram with cross reference. (EXECUTING is displayed on the screen.)
  This time depends upon the size and complexity of sequence programs. The cross reference is displayed by the page number and the line number of the LADDER diagram every contact.
  See Appendix printout example.
- If R1 key is pressed when each data is being output to the FANUC printer (External printer), the output is cancelled.

(9) OUTPUT CROSS REFERENCE (SEQUENCE NO)

Addresses (symbols, comments) are printed with cross reference Nos. by FANUC external PRINTER.
These Nos. correspond to the Mnemonic format list (screen) or Ladder diagram (RD command line number).
Key in above detailed memo No. ’09 <NL>’ and turn F10 key on. The screen changes to display the key in example and ’ADDR=’ as below.

Key in addresses to be output according to examples.

	Key in example	Address to be output
ALL	ALL <NL>	All addresses (G,F,Y,X,A,R,T,K,C,D in order)
Address initial	R <NL>	All address with the specified initial
Bit address	X1.0<NL>	Only bit address specified address
Byte address	R58 <NL>	Bit 0 – 7 of specified
Address range specification	F8.0, X7.2<NL>	Specified addresses in order of G,F,Y,X,A,R,T,K, C,D
	X0.2–END<NL>	All address after specified address

```
                *** CROSS REFERENCE LIST ***
                                                                PAGE=1
ADDRESS  SYMBOL        COMMENT DATA

G0000.0  *IT
    653
G0000.1  *CST
    653
G0000.4  *ESP
     22   568   901   912   1177   1189   1288   2800
G0000.5  *SP
     45   2802
G0000.7  ERS
    3435  3512
G0001.0  *AIT
     656
```

NOTE

1 When the same address performs double writing,”* MULTIPLE COIL USED *” is displayed.
2 If the F10 key is set to OFF and output performed, the cross reference table is displayed on the screen.

4.5.2 Paper Command

A 12–inch chart is also applicable to the FANUC printer (external printer).

(The standard chart size is 11 inches.) Enter the command for changing the chart by the following operation.

(1) Press [R3] key from the R key menu screen.

(2) ’REQUEST=’ is displayed at the lower left part of the screen.

(3) Key in ’PAPER <NL>’.

(4) The following message is displayed at the lower left part of the screen.

KEY IN NUMBER OF PAPER LENGTH
EXAMPLE 11–INCH;0,12–INCH;1.
LINE NUM.=

(5) Key in ’0 <NL>’ for 11–inch chart, or ’1 <NL>’ for 12–inch chart.

4.5.3 ROM Format Program

(1) Transfer of sequence program into PMC–SA1/SA2/SB/SC

A generated sequence program is transferred into PMC–SA1/SA2/SB/SC.
Connect SYSTEM P series to CNC by using a Reader/Puncher interface cable. (For this cable, see Appendix 1.) for the connection method and places, see Section 3.2.
Steps **1** to **6** show the operation on the CNC side.

1 Pressing soft keys <SYSTEM> and [PMC] displays the PMC screen. Steps **2** to **4** below must be performed when [I/O] is not displayed on the PMC screen. For a 9–inch CRT, press soft key [NEXT] to check that [I/O] is not on the screen.

2 Pressing soft keys [PMCPRM] and [KEEPRL] on the PMC screen displays the keep relay setting screen.

3 Set K17.1 to 1 on the keep relay setting screen.

4 Pressing soft key [RETURN] displays the PMC screen.

5 On the PMC screen, pressing soft key [I/O] displays the I/O screen. For a 9–inch CRT, press soft key [NEXT] before pressing soft key [I/O].

6 Pressing soft key [EXEC] on the I/O screen puts the system in the EXECUTING state.

7 Turn on F8 key from the SYSTEM P series menu screen. (Also turn on F12 key when the C language program is included.)

8 Key in menu number “5 <NL>”.

Now, the program transfer is started.
In SYSTEM P series, the transfer screen is displayed and the transfer counter is counted up. After transfer, the screen is reset to menu screen. In CNC screen, the COUNTER display is counted up.

***Procedure when a program cannot be transferred from SYSTEM P series to RAM of PMC**

i) When an alarm 31 occurs on SYSTEM P series screen;

Cause 1 : Reader/Puncher interface cable is defective.
Remedy : Use the specified cable.
Cause 2 : Reader/Puncher interface connector is not connected to correct channel SYSTEM P series.
Remedy : Connect the connector correctly.

ii) When the transfer counter of SYSTEM P series screen is counted up and normally terminated, but data are not transferred to the PMC RAM correctly;

Cause 1 : Reader/Puncher interface connector is not connected to CNC.
Remedy : Connect it correctly.
Cause 2 : CNC screen is not set to ”I/O of PMC” screen.
Remedy : Set the I/O screen by the soft key.
Cause 3 : An error occurs in ACI channel due to a certain cause.
Remedy : Turn off the power supply once, and turn it on again.

(2) Output method to floppy

1 Turn on F7 key. (Also turn on F12 key when the C language program is included.)

2 Set the floppy to the disk.

3 Key in menu number “5 <NL>”.

4 The screen is switched and the following message is displayed:

SET FD & KEY IN “OK”,“KILL” OR “NO”.
FD0= OK <INT OR ADD><P OR NP,></DATE,>
<DRIVE>@NAME
FD0 =

When loading data starting with the start of the floppy, specify INT. When loading data after the loaded files, specify ADD. After outputting all data, the screen is reset to the program menu screen. The menu screen is also reset by keying in “NO <NL>”.

(3) Method of outputting data to FA writer or PMC writer (EPROM for PMC/ROM module write)

1 Check the setting of the ROM writer. (See Section 3.4, “Setting of I/O Device.”)

2 Put the FA writer in the REMOTE mode by the [REMOTE/LOCAL] key before using it.

3 Turn on F9 key. (Turn on F12 key when the C language program is included.)

4 Key in menu number “5 NL”.

5 The screen is switched to the title screen, and the following message is displayed.

SET EPROM OR ROM MODULE & KEY IN ”OK” OR “NO”.
KEY IN =

Check the above message. For the PMC–SA1/SA2/SB, insert the EPROM for the PMC into the FA Writer or PMC Writer. For the PMC–RC, insert the ROM module for the PMC into the FA Writer or PMC Writer. Note, however, that ROM modules are not available with the PMC Writer.

6 Key in “OK <NL>” or “NO <NL>”.

When “OK <NL>” is keyed in, data are output from the SYSTEM P series memory to the EPROM for PMC or ROM module. After normal end, the screen is reset to the menu screen.

```
   *** TRANSFER ROM DATA TO PMC WRITER ***

01  MACHINE TOOL BUILDER NAME
02  MACHINE TOOL NAME
03  PMC & NC NAME
04  PMC PROGRAM NO
05  EDITION NO
06  PROGRAM DRAWING NO
07  DATE OF PROGRAMING
08  PROGRAM DESIGNED BY
09  ROM WRITTEN BY
10  REMARKS

    PMC CONTROL PROGRAM   SERIES : 4061   EDITION :01

    MEMORY USED : 00.0 KBYT SCAN TIME : 008 MSEC



  SET EPROM OR ROM MODULE & KEY IN 'OK' OR 'NO'
KEY IN =
```

```
   *** TRANSFER ROM DATA TO PMC WRITER ***

01  MACHINE TOOL BUILDER NAME
02  MACHINE TOOL NAME
03  PMC & NC NAME
04  PMC PROGRAM NO
05  EDITION NO
06  PROGRAM DRAWING NO
07  DATE OF PROGRAMING
08  PROGRAM DESIGNED BY
09  ROM WRITTEN BY
10  REMARKS

    PMC CONTROL PROGRAM   SERIES : 4061   EDITION :01
    MEMORY USED : 00.0 KBYT SCAN TIME : 008 MSEC


                                       MODE=BLANK
        ROML=10  ROMH=EF  MEM FF   AD=000000
                                           ALARM=083

OUTPUT=
```

Display mode

(BLANK : Blank check
PROGRAM : Write
VERIFY : Compare)

Error number

Enter 'E NL', and restart from menu.

4.6 COLLATION OF PROGRAM

4.6.1 Collation of Source Programs

Enter source programs from the designated input unit, and compare them. The operation method is the same as source program entry, except that ”6” shall be designated as the menu number.

(1) Comparison with PTR

1 Turn on F1 key.

2 Key in menu number “6 NL”.

(2) Comparison with FD

1 Turn on F2 key.

2 Key in menu number “6 <NL>”.

3 The screen is switched, and the following message is displayed.

SET FD & KEY IN “OK”, “KILL” OR “NO”,
FD0=OK<DRIVE> <@NAME OR : NUMBER>
FD0=

Specify the file name to be compared.

After normal end, the screen is automatically reset to the menu screen.

Also, this menu screen is reset by keying in “KILL

4.6.2 ROM Format Program

Compare ROM format program by reading it from the specified input device. The operation method is the same as in ROM format program input, except that menu number “7” is specified.

(1) Comparison with FD

1 Turn on F2 key.

2 Key in menu number “7 <NL>”.

3 The following operation is the same as in 4.6.1 2)– **3** and later.

(2) Comparison with PMC–RAM

Display the I/O of PMC screen on the CRT/MDI before executing the following operation.

1 Turn on F8 key.

2 Key in menu number “7 <NL>”.

Note when comparing P–G and PMC–RAM : The comparison between P–G and PMC–RAM should be performed immediately after the data transfer. (When the comparison is made after the output of ROM format data, the parity portion of data may become error.)

(3) Comparison with EPROM for PMC and ROM module

1 Turn on F9 key.

2 Key in menu number “7 <NL>”.

The screen is switched, and the comparison of ROM program is started. After normal end, the screen is automatically reset to the menu screen.

4.7 DELETION OF PROGRAMS

Delete ladder programs, symbols, message, titles, and I/O module data being loaded into SYSTEM P series memory according to the following procedure.

1 Put the screen to menu screen.

2 Key in menu No. “9 <NL>”.

3 The screen is switched, and the following message is displayed at the lower left part of the screen. See Fig. 4.7.

KEY IN “1,2,3,4 OR 5” OR “NO”
CLEAR/KEEP =

4 Key in data number of the data to be deleted or key in “NO <NL>”, if it is not desired to delete any data. After processing, the screen is automatically reset to the programmer menu screen.

```
KEY IN ONE OF THE FOLLOWING NO.S WHICH YOU WANT TO CLEAR DATA

NO.  ITEMS
01   TITLE DATA
02   SYMBOL DATA
03   LADDER DATA
04   MESSAGE DATA
05   I/O MODULE DATA
06   ALL DATA CLEAR



KEY IN ’1. 2. 3. 4. 5 OR 6 OR ’NO’
CLEAR/KEEP =
```

Fig. 4.7 Delection of sepuence programs

Example)

i) When all title data are to be deleted;
Key in “1 <NL>”.

ii) When all symbol data are to be deleted;
Key in “2 <NL>”.

iii) When all ladder programs are to be deleted;
Key in “3 <NL>”.

iv) When all message data are to be deleted;
Key in “4 <NL>”.

v) When I/O module data are to be deleted;
Key in “5 <NL>”.

vi) When all titles, symbols, ladders, messages and I/O module data are to be deleted;
Key in “6 <NL>”.

vii) When no data are to be deleted;
Key in “NO <NL>”.

4.8 SPECIAL USES OF THE R3 KEY

Key in <NL> alone at the menu screen to display the R key menu screen. Key in R3 at the R screen, and the display 'REQUEST=' will appear at bottom left of the screen, making key inputs possible. Key in <NL> on this screen to return to the R key menu screen.

R3 executes a large number of processings. For the FAPT LADDER system, however, note the following two points:

(1) Floppy file name output

1 Press R3 key at the R key menu screen.

2 This will change the screen contents, displaying 'REQUEST=' at its left bottom.

3 Key in FDLIST <NL>.

4 The file name will appear on the CRT display. To print out the file name, turn on the F5 (printer) key in advance.

(2) Change of I/O devices (for output to a printer other than that of PPR)

1 Key in IO PRT, CN3, F5 <NL> while the screen displays 'REQUEST='. When the F5 key has been turned on in advance, the data is printed on the printer connected to connector CN3 on the SYSTEM P series rear side.

4.9 DIRECT EDITING BY LADDER DIAGRAM

4.9.1 Outline

Using the P–G Mate/Mark II software keys (in the case of P–G Mate, the F keys), sequence program creation and editing can be performed directly by the ladder diagram.

In the following explanation, [P–G Mate] is called [Mate] and [P–G Mark II] is called [Mark II].
When it is possible to use this function, in the R key menu screen

R1: EDIT

is displayed. (In systems where [UNUSED] is displayed, it cannot be used.)

The following items are present in the edit function.

- Ladder diagram direct editing by software key and cursor (input, addition, deletion and substitution)
- Copying, moving and deletion of multiple lines of the ladder
- Optional relay and coil reference
- Comment display on ladder diagram

4.9.2 Limitations in SYSTEM P Mate

(1) This function operates only when the P–G Mate main unit is version 04 and later. (When the power supply is turned on, it is displayed in the lower right part of the initial screen.)

(2) The function keys <F keys> are used instead of the soft keys (P–G Mark II). In the description that follows, an explanation for the soft keys (P–G Mark II) is given. When P–G Mate is used, operate with the function keys. At this time, in order to make the F key respond and display the screen bottom line, the F key lamp illuminates to correspond to those items displayed with shaded characters on the screen.

4.9.3 Selection of Program Menu by Soft Keys

The program menu appears in order to operate this function.

The program menu is displayed when the <R1> key is pressed from the R menu screen. The program menu is displayed above the soft keys (in the case of P–G Mate, the function keys) as shown in the screen below, and gives significance to the keys.

(1) Keyboard

Refer to Section “3.3 SYSTEM P keyboard”.

(2) Relationship betweeen program menus and soft keys

The relationship between the program menus and the soft keys is shown in the following for each function. These menus are changed by pressing the related keys. For menu contents, refer to the explanations described later. Utilize this figure when operating.

R0 : PROGRAMMER
R1 : EDIT
R2 : UNUSED
R3 : REQUEST

Press the <R1> key

R keys menu screen

—| |—	—|/|—	—( )—|	|—o( )—|	FUNCTN	———	— — —	↑ or ↑	↑ or ↑	COMAND

[COMAND]

[COMAND]

INSNET	DELNET	INSERT	ADRESS	SEARCH	COPY	MOVE			

1 2 3 4 5

1 [DELNET]

EXEC	CANCEL	SEARCH	C–DOWN	C–UP					

2 [INSERT]

INSNET	INSLIN	INSELM							

3 [ADRESS]

INSNET	DELNET	INSERT	SYMBOL	SEARCH	COPY	MOVE			

4 [SEARCH]

TOP	BOTTOM	SRCH	W–SRCH	N–SRCH	F–SRCH	C–DOWN	C–UP		

5 [COPY/MOVE]

UNTIL	CANCEL	SEARCH	C–DOWN	C–UP					

TO	CANCEL	SEARCH	C–DOWN	C–UP					

4.9.4 Sequence Program Input

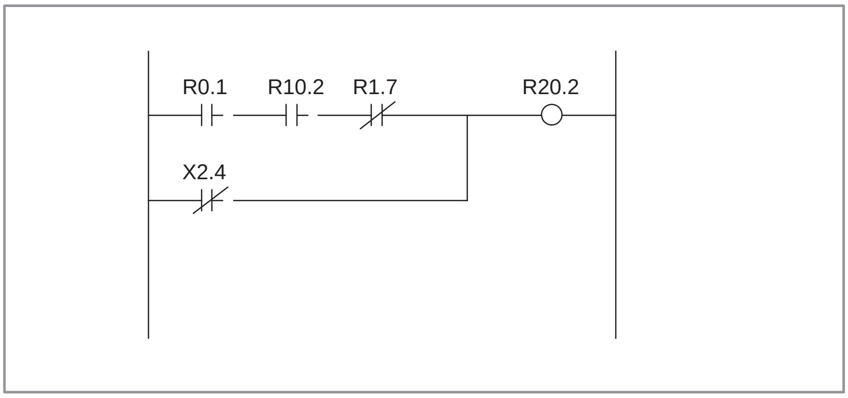
In order to input the sequence program, press the <R1> key from the R key menu. The soft key menu program is displayed, and in the case that the sequence program has not yet been input, only the left and right vertical lines of the ladder diagram are displayed on the screen.

Start inputting a program with the screen in this state.

Input a ladder diagram program by moving the cursor to the desired input position using the cursor key.

The following description shows an example of the input of a program of basic instruction and a program of functional instruction.

(1) Basic instruction program input



1 Press the soft key [—| |—] after moving the cursor to the start position.

Symbol [—| |—] is input at the cursor position and HORIZONTAL LINE ILLEGAL is displayed at the lower right part of the screen. This is a cautionary message which shows that the ladder diagram horizontal line is not yet completely created. Input the continuation address and bit data.

2 Press the <NL> key after inputting R0.1 using the keyboard. The address is set on the contact and the cursor shifts rightward.

3 Input A contact with address R10.2 by the above methods **1** and **2**.

4 Input B contact R1.7.

Press the soft key [—|/|—], input address R1.7, and then press the <NL> key. The address is set on the B contact and the cursor shifts rightward.

5 Press software key [—○—|] with the cursor position unchanged.

A right horizontal line is automatically drawn, and a relay coil symbol is entered near the right vertical line.

6 Press the <NL> key after inputting address R20.2.

The cursor automatically shifts to the input start position of the next line.

7 Next, input the OR condition.

Press the soft key [—|/|—], input address X2.4, and then press the <NL> key. The address is set on the B contact and the cursor shifts rightward.

8 Press the soft key [———] to input a horizontal line

When inputting the horizontal bar key [———], by keying in a numerical value and pressing this bar key, a horizontal line for the frequency will be drawn. However, this horizontal line will not be drawn over the LINE.

9 Because the upper right line OR is necessary, press the soft key [ ] and input the upper right vertical line to end.

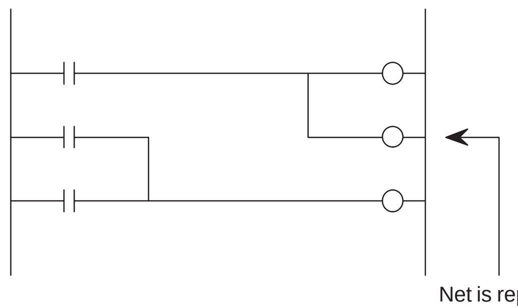
NOTE

1 When the ladder program displayed on the screen is incomplete (when, for example, addresses have not been entered) or erroneous, the screen cannot be scrolled even when a page key is pressed. Before attempting to scroll the screen, therefore, ensure that the ladder program is complete and error–free.

2 Since 8 contacts + coil are specified to be inputtable per line from the screen, any more contacts in excess of this amount cannot be input. However, this restriction does not apply to a sequence program created with mnemonic format.
When a sequence program, transferred from the offline programmer to the PMC, exceeds the length which can be displayed on a single line, the program is displayed using two or more lines, linked with a continuation symbol.
This continuation symbol is not erasable by software key [±±±±].

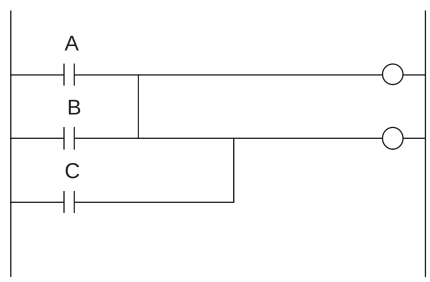
3 Below is shown an example with an error net, or part of it, erased with no error display.

1 Case of multiple nets on 1 LINE



Downward from the net is erased

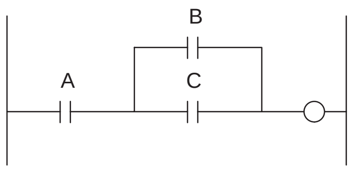
2 Case of multiple WRT results in 1 NET difference as shown in the diagram below.



Section C is erased

NOTE

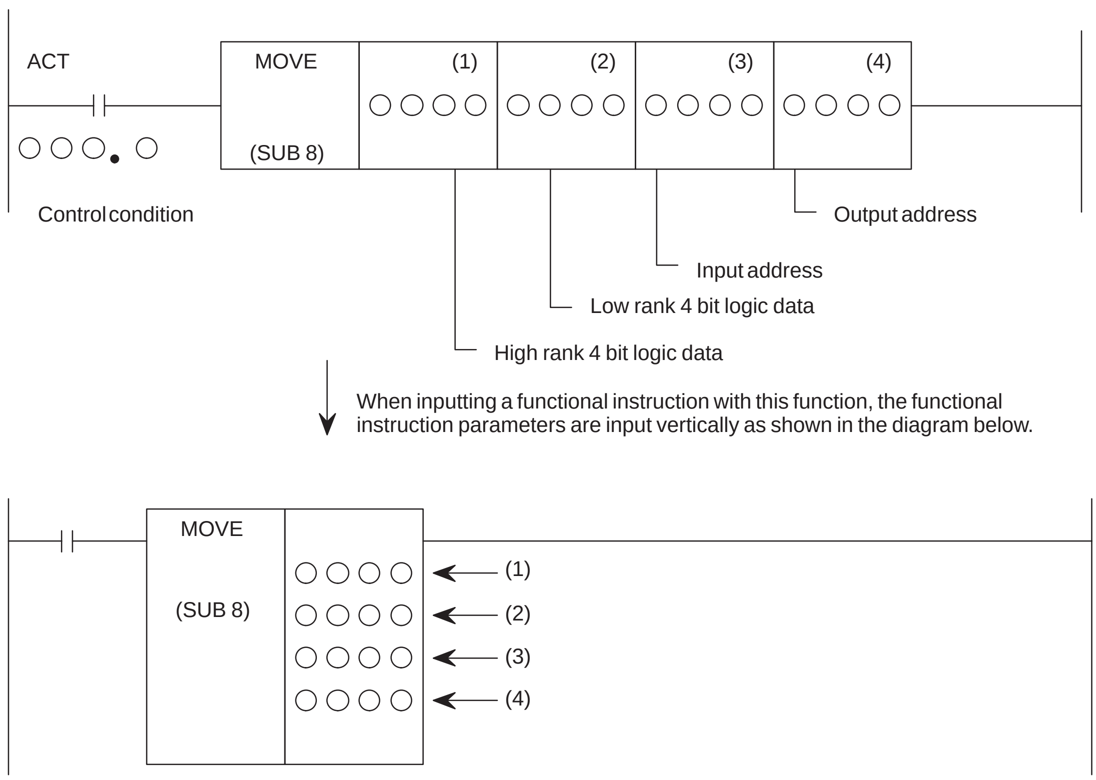
3 Case of exceeding the highest rank WRT in 1 NET



Section B is erased

(2) Case of functional instruction program input

To input a functional instruction, input the soft key [FUNCTN], and then input the functional instruction name or SUB number.
Further, when inputting a functional instruction, after keying in the functional instruction number, it does not matter if the [FUNCTN] key is pressed.
When you can not remember the instruction name or SUB number, the functional instruction table corresponding to the instruction symbol and SUB number can be displayed on the screen.
The functional instruction table is automatically displayed after inputting an incorrect instruction name or SUB number and then pressing the [FUNCTN] key, or by pressing the [FUNCTN] key only without inputting any other key.
In order to return from the functional instruction table to the original ladder diagram, press the [FUNCTN] key.



1 Input a control condition.

Press soft key [—| |—], input the address and bit data, and then press the <NL> key. The cursor shifts rightward.

2 Input an instruction

Press the soft key [FUNCTN], input SUB number 8, and then press the <NL> key. A functional instruction diagram appears as shown in the above figure.

3 Input an instruction parameter

Input the high rank 4 bit logic data of the first parameter, and then press the <NL> key. The cursor automatically lowers downwards. Input the three residual parameters in order.

4.9.5 Substitution of Sequence Programs

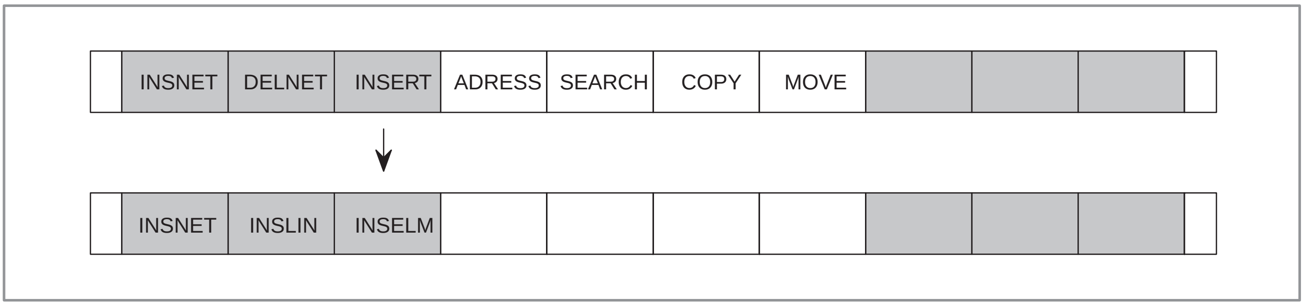
The method of substituting a created sequence program is the same as that described earlier in Section 4.9.4.

Move the cursor to the program part you want to alter and input the change data.

4.9.6 Additions to Sequence Programs

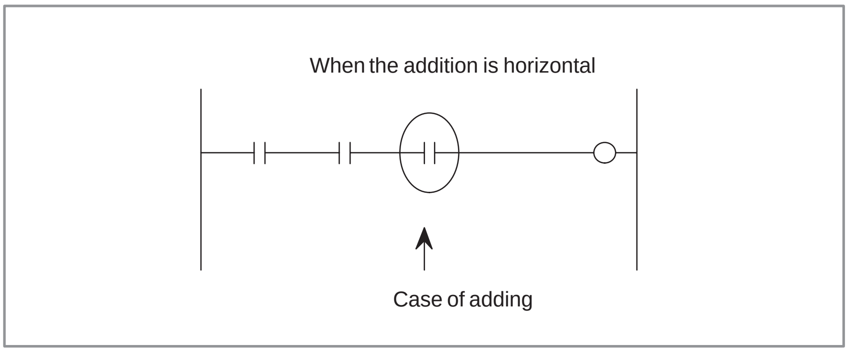
From the soft key program menu, press the soft key [COMAND] and operate with the soft keys shown below.

When you want to end the program menu shown below, press the soft key at the extreme left.

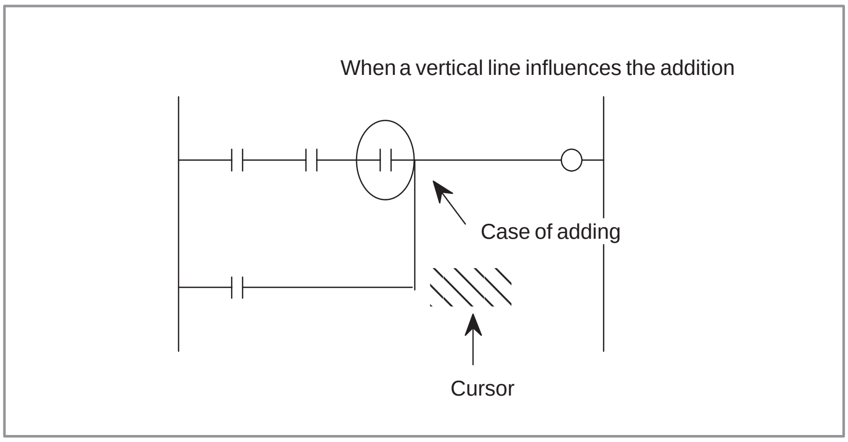


A sequence program is added in four ways on the ladder diagram as described below.

(1) Case of adding a relay contact in the horizontal direction



Move the cursor to the position where you want to add, and input te program by the method described in Section 4.9.4.



1 Move the cursor to the above position.

2 Press the soft key [↑_] in order to erase the upper left vertical line.

The upper left line, vertical to the cursor disappears.

3 Press the soft key [_↑] in order to produce an upper right line vertical to the cursor. Then, press the soft key [———]. Both vertical and horizontal lines are created.

4 Shift the cursor to a line of contact addition position.

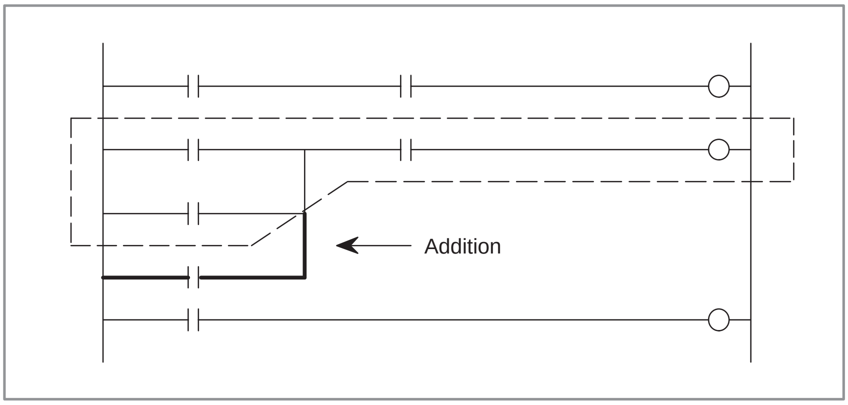
5 Press the soft key [—| |—] to add contacts.

(2) Adding a vertical line

For adding a vertical line as shown in the above diagram, the area to be added is required. In order to produce this area, shift the entire part after the part to be added by one line by moving the cursor to the ladder diagram within the dotted line range (an optional part is allowable) and then pressing the soft key [INSNET].

The lower ladder diagram shifts downward by one line, each time the [INSNET] key is pressed thereby producing the area to which a line is to be added.

If a surplus addition area remains unused after the addition processing ends (for example, if an area corresponding to 3 lines has been reserved when two lines have been added), there is no problem if the area is left remaining.



1 Move the cursor to the ladder diagram bounded by a dotted line.

2 Press the soft key [INSNET].

3 Pressing the [INSNET] key without keying in numeric values will cause one line to be inserted.

4 Pressing the [INSNET] key with keying in numeric values will cause the line to be inserted the number of times specified by the numeric value input.

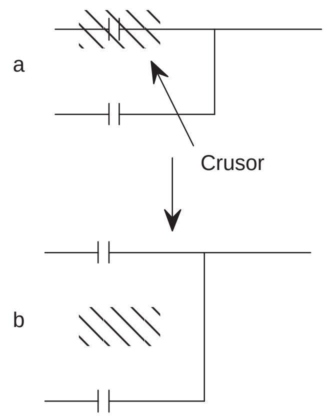
5 After setting the cursor to a position to which you want to add, press the soft key [—| |—]. After setting address data, press the <NL> key. The cursor shifts rightward.

6 Press the shift key [↑] to create an OR circuit.

(3) Inserting the 1 NET sequence program LINE.

Space lines are inserted in units of 1 LINE.

1 Key in the number of lines you want to insert and press the [INSLIN] key. The inputted number of lines will be inserted. (If the number of lines to be inserted is not keyed in, but the [INSLIN] key is pressed, one line will be inserted.)

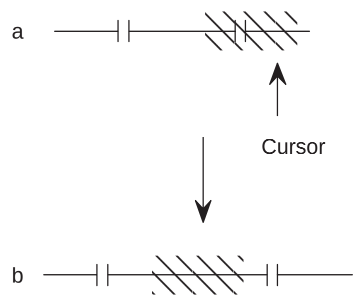


If the [INSLIN] key is pressed with the cursor in the above position, the state shown in the diagram on the right will occur.

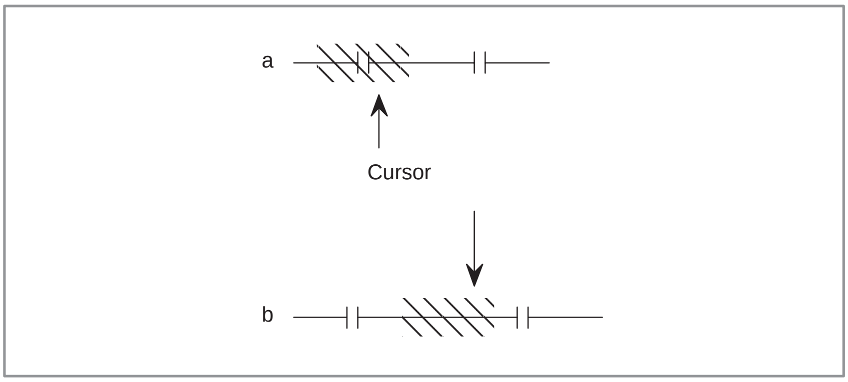
(4) Inserting the 1 NET sequence program elements

Elements are inserted in 1 element units.

1 Key in the number of elements you want to insert and press the [INSELM] key. The inputted number of elements will be inserted. If a number of elements prefixed by the character ”A” are keyed in and the [INSELM] key is pressed, the elements are inserted after the cursor.
(If the number of elements to be inserted is not keyed in, but the [INSELM] key is pressed, one element is inserted.)



If the [INSELM] key is pressed with the cursor in the position on the left, the state shown in the diagram on the right will occur.



If the character ”A” is keyed in and the [INSELM] key is pressed with the cursor in the position on the left, the state shown in the diagram on the right will occur.

4.9.7 Deleting a Sequence Program

(1) For deleting part of a program, use the following three kinds of soft keys and delete after setting the cursor to the unnecessary part.

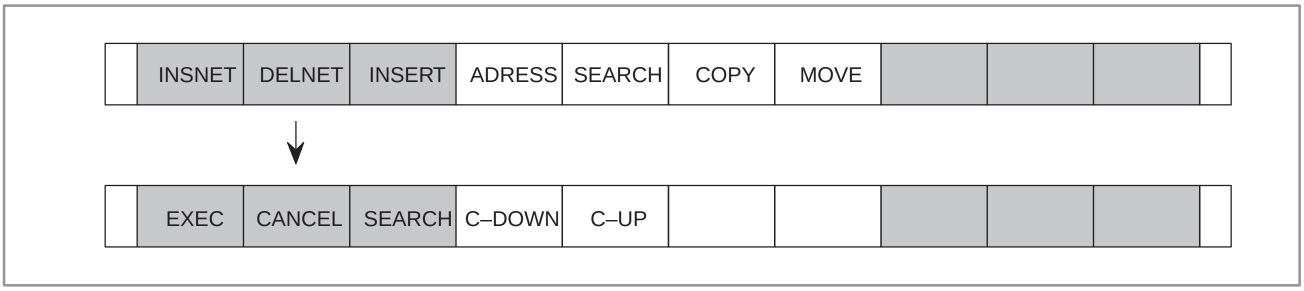
[– – – –] : Deletion of horizontal lines, relay contacts coils, etc.

[↑] : Deletion of upper left vertical line to the cursor

[↑] : Deletion of upper right vertical line to the cursor

(2) For the deletion of a program net (part corresponding to the section from RD instruction to WRT instruction), use the [DELNET] key.

(3) Deleting multiple NETs in NET units



1 Deletion

Move the cursor to the NET you want to delete and press the [DELETE] key. The net you want to delete will be displayed in red. (In the case of Mate, in reversal display.)

2 Deleting multiple nets

Move the cursor with the cursor DOWN key, [C–DOWN] key, or [SEARCH] key to display in red the NET you want to delete. (In the case of Mate, in reversal display.) Further, key in a numerical value and press the [C–DOWN] key to move the cursor the number of times specified by this value.

3 Execution . . . . . . Press the [EXEC] key

Cancellation . . . . Press the [CANCEL] key

4 If you already know the NET you want to delete, move the cursor to the first NET, key in the number of NETs, and press the [DELNET] key to omit steps **1** and **2**.

4.9.8 Searching a Sequence Program

Search a sequence program by using the following soft keys.

(1) Soft key [TOP]

When this key is pressed, the start of the sequence program is desplayed on the screen and the cursor also sifts to the program start position.

(2) Soft key [BOTTOM]

When this key is pressed, the last of the sequence program is displayed on the screen and the cursor also shifts to this program end position.

(3) Soft key [SRCH]

In this search, you specify an address you want to search and it searches the specified address from the program of the cursor part on this screen to the last part of the program and displays the address on the screen. There are two methods to specify the address you want to search.

(a) Method of specifying the address by the cursor

Set the cursor to the relay contact part of the address you want to search and press the soft key [SRCH].

The system searches the same address as the address specified by the cursor from the cursor part of the program currently displayed on the screen to the end of the program.

When the same address is found, the program part is displayed on the screen, and the cursor shifts to that address part. If the same address is not found as a result of this search, the cursor remains in the same position.

When finishing, press the soft key on the extreme left.

(b) Method of specifying the address by input

Input the address you want to searcch by using address and numeric keys, then press the soft key [SRCH].

The same address as specified is searched from the program of the cursor part currently displayed on the screen to the last part of the program.

When the same address is found, the program part is displayed on the screen, and the cursor shifts to that address part.

If the same address is not found as a result of this search, an error is displayed.

(4) Soft key [W–SRCH]

This key specifies an address of the relay coil to be searched, and then searches the relay coil of the specified address from the program at the cursor part to the end of the program on this screen. Then, it displays the relay coil on the screen.

Two methods are available to specify the address of the relay coil to be searched.

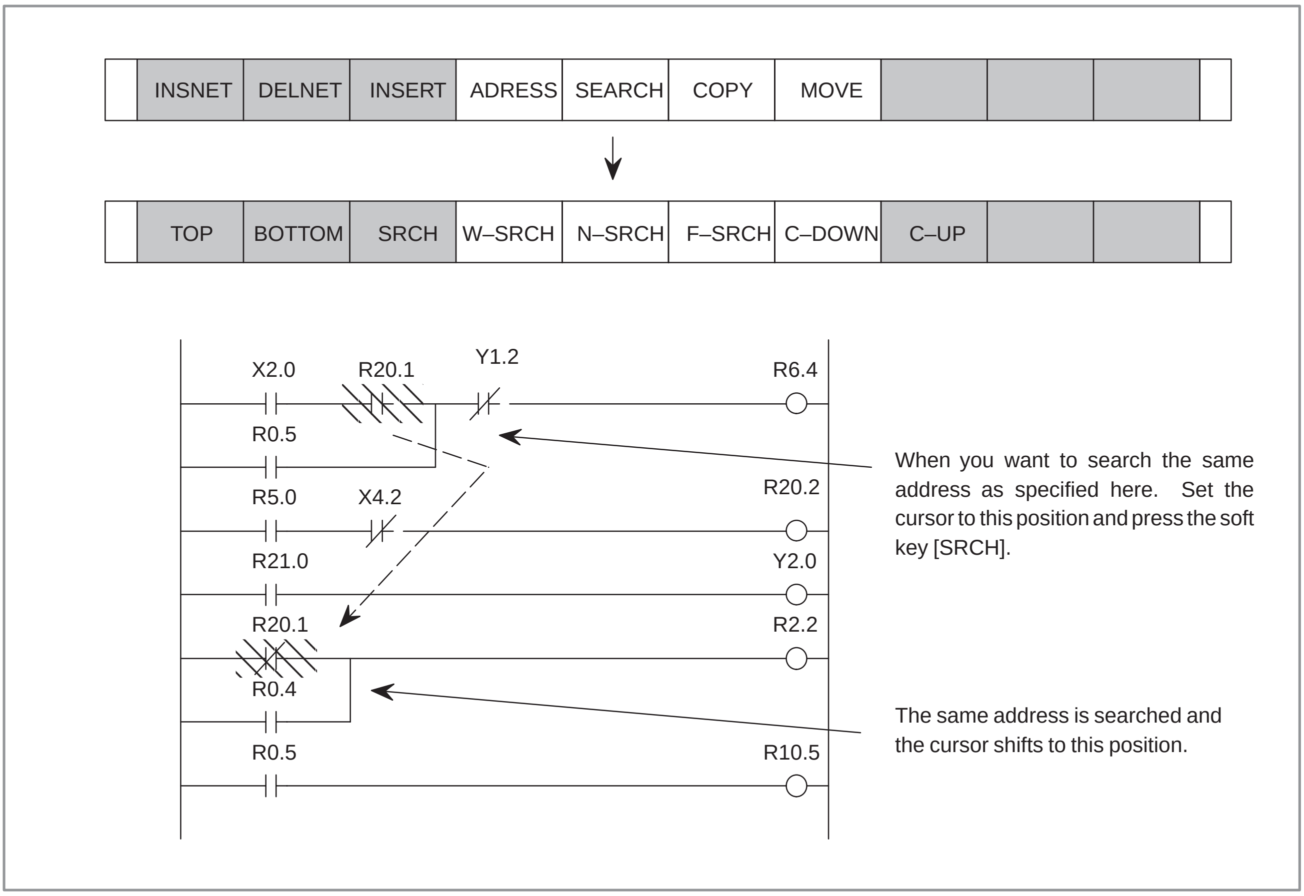
(a) Method of specifying the address by cursor

Set the cursor to the relay contact of the relay coil to be searched, and press the soft key [W–SRCH].

The corresponding relay coil is searched from the program of the cursor part to the end of the program.

When the relay coil is found, the program part is displayed on the screen, and the cursor shifts to the relay coil.

If no corresponding relay coil is found as a result of the search, an error occurs.



(b) Method of specifying the address by input

Input the address of the relay coil to be searched by both address and numeric keys, and then press the soft key [W–SRCH].

The specified address relay coil is searched from the program of the cursor part currently displayed on the screen to the end of the program.

When the specified address relay coil is found, the program part is displayed on the screen, and the cursor shifts to the relay coil.

If no relay coil is found as a result of the search, an error occurs.

(5) Soft key [N–SRCH]

This displays the ladder with the specified NET number from the top of the screen. If the number is not keyed in, but the [N–SRCH] key is pressed, the display is scrolled down by one NET.

(6) Soft key [S–SRCH]

Key in the functional instruction name or number and press the [S–SRCH] key to start searching the functional instruction. When the [S–SRCH] key is pressed during execution of a functional instruction, the functional instruction with the same number as this instruction is searched.

(7) Searching with cursor keys (<←> , <→>)

- Key in the address or symbol and press the cursor to start searching the NET No.
- Key in the NET NO. and press the cursor key to start searchng the NET NO.

- Key in the functional instruction name or functional instruction number starting with “S” and press the cursor key to start searching the functional instruction.

Example) Key in “END1” or “S1” and press the cursor to search functional instruction END1.

4.9.9 Copying a Sequence Program

The sequence program with multiple NETs is copied in units of NETs. Specify the NET to be copied and specify the copy position with the cursor. When copying, the number of copies can also be specified.

1 Copying

Move the cursor to the NET you want to copy and press the [COPY] key. The NET you want to copy will be displayed in yellow (in the case of Mate, in reversal display).

2 Copying multiple NETs

Move the cursor with the cursor UP/DOWN key, [C–UP] key, [C–DOWN] key, or [SEARCH] key to display in yellow the NET to be copied. (In the case of Mate, in reversal display.) Further, if you in a numerical value and press the [C–UP] or [C–DOWN] key, you can scroll up or down the screen by the number of times specified by this value.

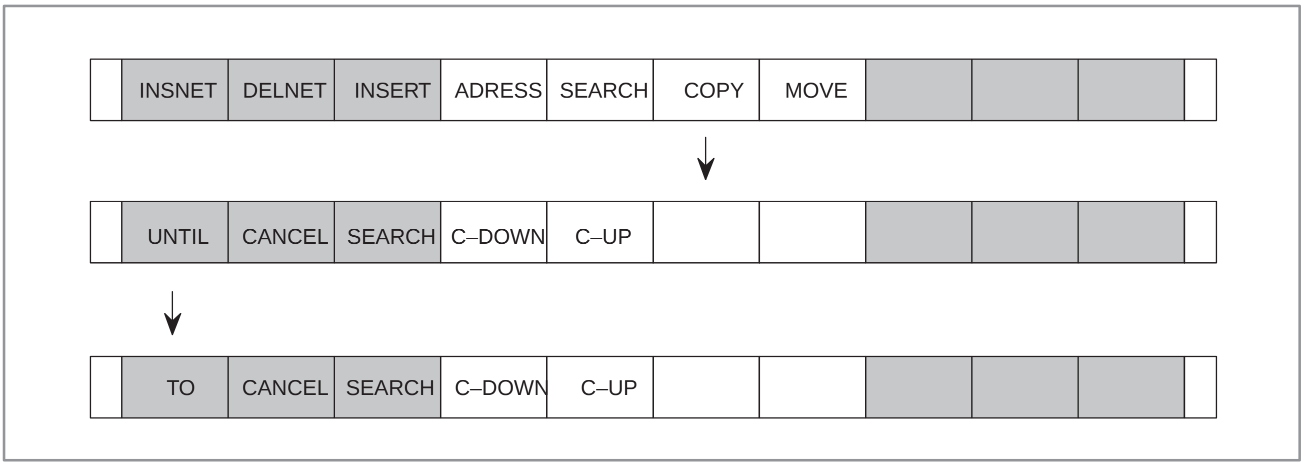
3 Setting the NET to be copied

Press the [UNTIL] key.

4 Specifying the copying address

Copying is performed by the [TO] key. At this time, the NET is copied in the direction above the cursor. If the number of copies is keyed in before the [TO] key is pressed, the NET is copied that specified number of times.

5 Further, if the NET you want to copy is already known, if the cursor is moved to the first NET and the number of NETs is keyed in, then by pressing the [COPY] key, steps **1** to **3** can be omitted.



NOTE
An error NET cannot be copied.

4.9.10 Moving a Sequence Program

A sequence program with multiple NETS is moved in units of NETs. Specify the NET to be moved and specify the move position with the cursor. When moving, the number of moves can also be specified.

1 Moving

Move the cursor to the NET you want to move and press the [MOVE] key. The NET you want to move will be displayed in yellow. (In the case of Mate, in reversal display.)

2 Moving multiple NETs

Move the cursor with the cursor UP/DOWN key, [C–UP] key, [C–DOWN] key, or [SEARCH] key to display in yellow the NET to be moved. (In the case of Mate, in reversal display.) Further, if you key in a numerical value and press the [C–UP] or [C–DOWN] key, you can scroll up or down the screen by the number of times specified by this value.

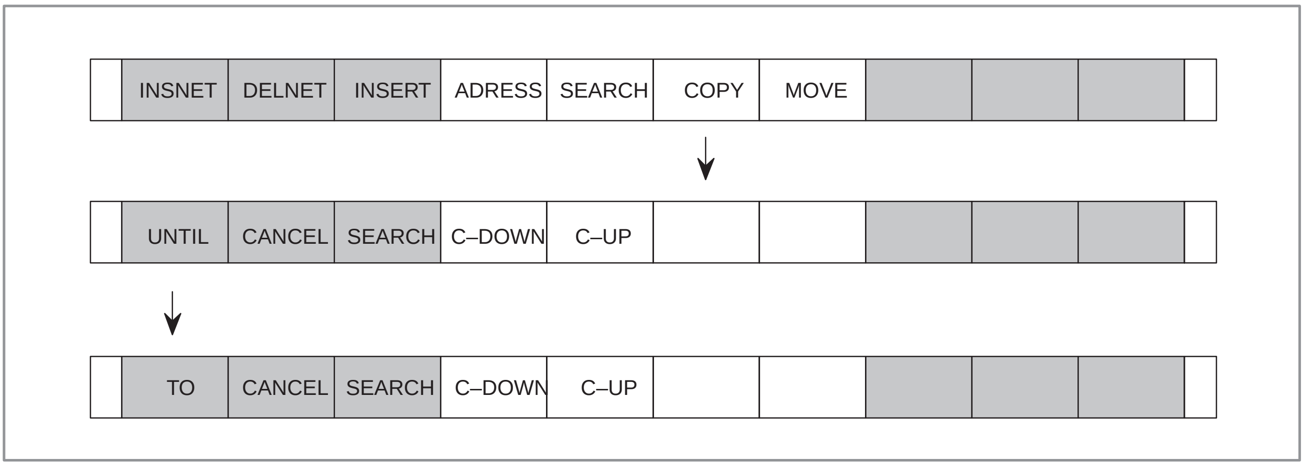
3 Setting the NET to be moved

Press the [UNTIL] key.

4 Specifying the moving address

Moving is performed by the [TO] key. At this time, the NET is moved in the direction above the cursor.

5 Further, if the NET you want to move is already known, if the cursor is moved to the first NET and the number of NETs is keyed in, then by pressing the [MOVE] key, steps **1**–**3** can be omitted.



NOTE
An error NET cannot be moved.

4.9.11 Symbol Data Display

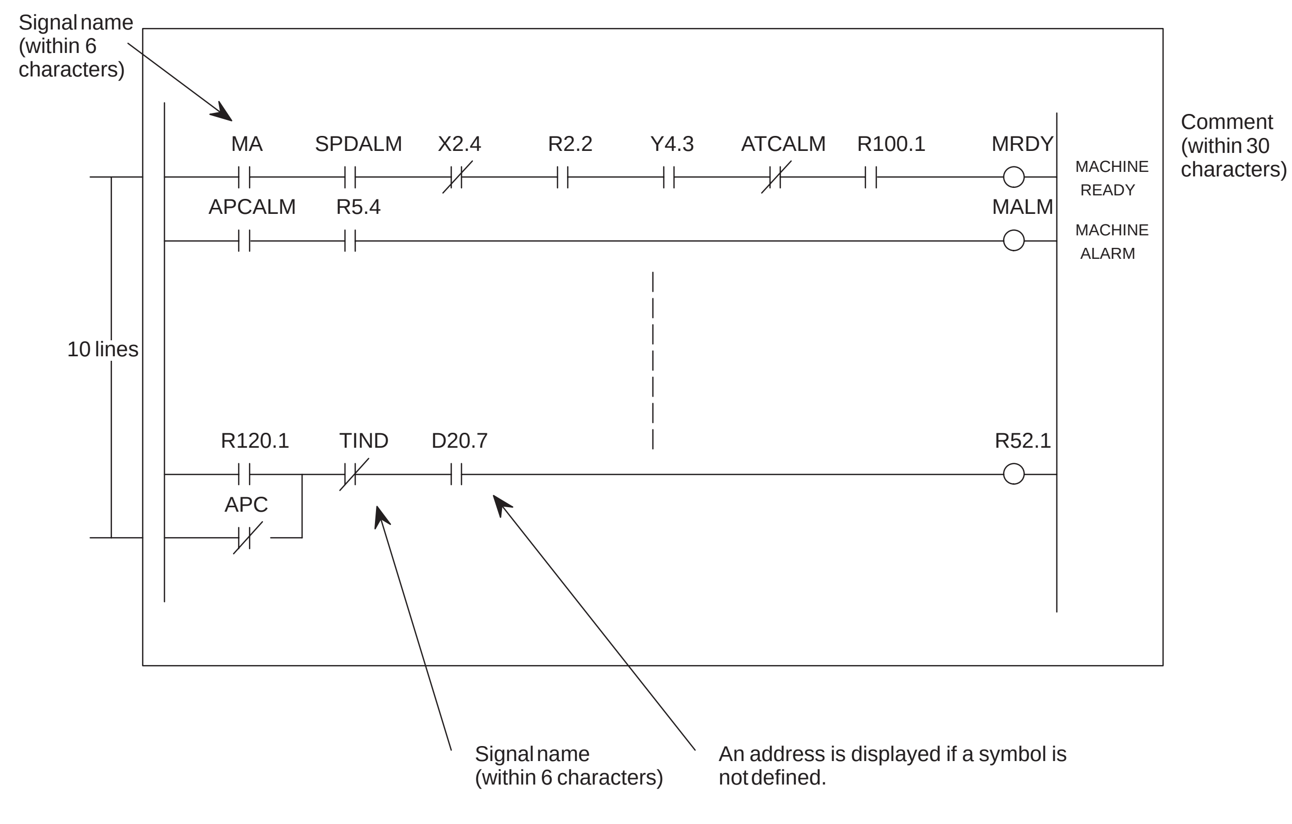
(1) Symbol and comment data display

Symbol data and comment are displayed together with a ladder diagram on the screen as follows.

When symbol data and comment are defined in signal addresses in the program, the signal name and comment are displayed as shown in the above diagram.
When converting the symbol and address display, press the shift key [ADRESS or SYMBOL].

(2) Symbol input and search in the sequence program

When symbol data is defined in signal addresses in the sequence program, input and reference can be performed by the symbols. (Address and symbol are only different in operation.)
If neither symbol data nor comment is defined at an address, the address is displayed as it is.



4.9.12 Compressed Input by [COMAND] Key

The main function of each soft key can be directly selected from the [COMAND] key.

After keying in the characters shown below, press the [COMAND] key. [ ] shows parts that can be omitted. Further, the ”n” appearing after the characters signifies that it is also posssible to input a numerical value. For example, after keying in ”D2”, pressing the [COMAND] key results in the same operation as keying in 2 and pressing the [DELNET] key.

I [NSERT]

D [ELNET] [n]

A [DRESS]

SY [MBOL]

S [EARCH]

C [OPY] [n]

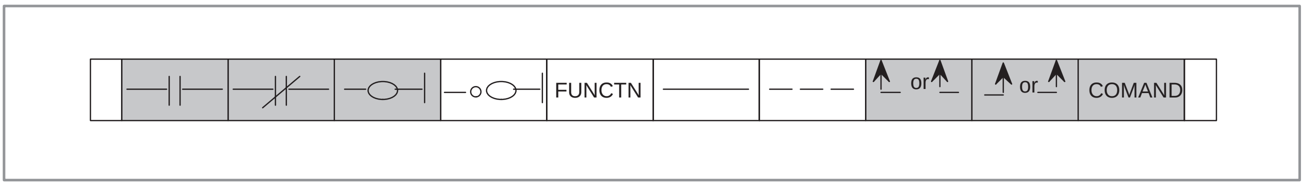
M [OVE] [n]
n :numerical value

The creation and search of programs is performed by pressing the software keys of the above menu.

> **NOTE**
> The software keys [↑ or ↑] and [↑ or ↑] are used to create or delete the upper left vertical line or the upper right vertical line on the ladder diagram. The solid line display of the vertical line indicates creation; the dotted line display of the vertical line indicates deletion. As to which menu will appear above the software keys, is decided by the ladder diagram form and the cursor position.

4.9.13 Ending Edit of a Sequence Program

In the program menu shown below, press the extreme left software key.



> **NOTE**
> When an error NET exists, ERROR NET NO. is displayed and you cannot end the edit. End after correcting the erroneous NET.

4.10 INPUT/OUTPUT OF LADDER PROGRAM WITH P–G AND FLOPPY CASSETTE/FA CARD

4.10.1 General

The ladder program can be stored in or fetched out of a floppy cassette/FA card by connecting P–G and floppy cassette adapter/FA card adapter by using this function enables reading the program stored in a floppy cassette/FA card by using PMC RAM into P–G or reading the program stored in a floppy cassette/FA card by using P–G into PMC RAM. The usable adapters are as follows:

- FANUC cassette adapter 3 (A13B–0131–B001)/cassette F1 (A87L–0001–0038)
- FANUC floppy cassette adapter (A13B–0150–B001)/floppy cassette (A87L–0001–0039)
- FA card adapter (A13B–0148–B001)/FA card (A87B–0001–0108)

4.10.2 Setting I/O Commands

When using the FANUC floppy cassette adapter/FA card adapter, change the settings of the input/output devices by the following ‘IO commands’.

1 Press the R3 key on the R key menu screen. ‘REQUEST=’ is displayed lower left on the screen, and keying in is permitted.

2 Key in ‘IO BCA, CN2, F13, F14 [NL]’. The floppy cassette adapter/FA card adapter is assigned to channel 2.

3 To return the assignment to channel 2 to PMC WRITER, key in ‘IO AUX, CN2, F9 [NL]’.

4.10.3 Program Input

1 Turn on F13 key.

(Turn on F12 too, when C language program is included.)

2 Key in the menu No. ‘3 [NL]’.

3 (For PMC–SC only) Enter the type of a ROM module to be used. (See the following note for selecting ROM module B or C.)

SELECT THE TYPE OF ROM MODULE ACCORDING TO THE FOLLOWING NO.
ROM MODULE 0:A, 1:B, 2:C
No. =

4 The message is displayed lower left on the screen.
SET BC & KEY IN ‘OK’ OR ‘NO’
BC = OK <FILE NO. OR NEXT>
BC =

5 Set the floppy cassette/FA card in the adapter, and enter the following data.

6 ‘OK 1 [NL]’ (specify file No.) or ‘OK NEXT [NL]’ (read the next file).

7 The screen changes, and reading the program from the floppy cassette/FA card starts.

8 When the program reading ends normally, the screen will automatically return to the programmer menu. If any error is detected during the program reading, ‘PART=’ is displayed lower left on the screen. Check the error contents, and key in ‘E [NL]’. The screen will return to the programmer menu.

> **NOTE**
> When ROM module B or C is selected during use of SYSTEM P Mate, the program is overlaid. In this case, insert the work floppy disk for the external memory into drive 1.

4.10.4 Program Output

1 Turn on F14 key.

(Turn on F12 too, when C language program is included.)

2 Key in the menu No. ‘5 [NL]’.

3 The message is displayed lower left on the screen.
SET BC & KEY IN ‘OK’ OR ‘NO’
BC = OK <INT OR ADD OR FILE NO.>
BC =

4 Set the floppy cassette/FA card in the adapter, and enter the following data.

5 ‘OK INT [NL]’ (write at the floppy head),
‘OK ADD [NL]’ (write in the next file) or ‘OK1 [NL]’ (specify file No.).

> **CAUTION**
> When specifying file number, put the numbers in the ascending order. If the file No. located at the middle of a floppy disk is specified, the files after that will be deleted.

6 The screen changes, and writing the program into the floppy cassette/FA card starts.

7 When the program writing ends normally, the screen will automatically return to the programmer menu. If any error is detected during the program reading, ‘PART=’ is displayed lower left on the screen. Check the error contents, and key in ‘E [NL]’. The screen will return to the programmer menu.

4.10.5 Program Collation

1 Turn on F13 key.

(Turn on F12 too, when C language program is included.)

2 Key in the menu No. '7 [NL]'.

3 The following operations are the same as those after **3** in 'Program input'.

NOTE

For the program which is output from PMC–SA1/SA2/SB/SC RAM board to the floppy cassette/FA card by specifying LADDER of ALL, there is no problem in the input/collation. It is impossible to make input/collation for the program which is output by specifying PARAM.

5 FILE EDITING FUNCTION

5.1 GENERAL

This function edits floppy disk data in the unit of file. When key in only <NL> the menu screen of R key appears key in R3 key on the R key menu. 'REQUEST=' will be displayed on the left below part of the screen to show a key–in enable condition.

> **NOTE**
> The format for file designation is as follows:
> [drive No.] @ file name
> : file No.

The file attributes are as shown below.

(1) File number
(2) File name
(3) File creation date
(4) Identification of protection file (protect)
(5) File size
(6) Multi–volume number

These file attributes are attached when writing data into floppy disk.

When writing, the next floppy disk set request message is displayed, so specify date and protection file.

File editing command table

Contents of jobs	Name of command (Instruction)	Contents inputted from keyboard (NL key is inputted at the end of a command)
Display of file name, or file size	FDLIST	FDLI { [D, [P,] [S,] [F,] [L,] File designation }
Change of file name, date, etc.	RENAME	RENA file designation [, {P / NP}] [,/date] [,@ new file name]
Deletion of file	SCRATCH	SCRA file designation
File area condensation	CONDENSE	COND [drive No.]
Copy of file (This command is effective for SYSTEM P series with 2–floppy disk unit.)	REMOVE	REMO [{M / A} ,] [file designation] [, {INT / ADD}] [, {P / NP}] [,/date] [,@ new. file name]

```
Set FD, and key in 'OK', KILL, or 'NO'.
FD=OK        <INT or ADD,> <P or NP,> </data>
             <drive><@name>
FD=_
```

Set the floppy disk and key in as follows.

```
{ OK   { INT.   { P ,    [/date] [drive] [@ file name]
{        ADD      NP
{ NO
{ KILL
```

When reading, the following floppy set request message is displayed.

```
Set FD, and key in 'OK', KILL, or 'NO'.
FD=OK          <drive.> <@name or : nember>
FD=_
```

Set floppy disk and key in as follows.

```
{ OK     [drive number]      { @ file name
{                              :  file number
{ NO
{ KILL
```

In file editing function, the above floppy disk set request message key input parameter can also be used. Now, parameter used in common here here has the following meaning. Specify 'OK', 'NO', 'KILL' and instruct the answer to the set request.

OK . . . . After instructing execution of read and write, specify parameter.

NO . . . . Cancel read/write to floppy only.

KILL . . Cancel the specified process.

Parameters instructing details of read and write is as follows.

Parameter	Function	Notes
INT	When writing, write from the head of the floppy	When omitted, it is regarded as ADD. If INT is specified to protection file, an error generaters.
ADD	When writing, add after exising fie	
P	Prepare as protection file	When omitted, it is regarded as NP. Ready files can be changed by RENAME command.
NP	Prepare as ordinary file	
Date	Specify file preparation date with 6 numbers	Blank when omitted.
Drive number	Specify drive number 0 or 1 set with read/write floppy disk. 0 ; Upper unit 1; Lower unit	When omitted, it is regarded as 0. See Note).
@ File name	Specify file name (Max. 17 characters). When reading, the first name correspond–ing to the specified names is vallid.	Always specify when writing. When reading, if omitted, the file is valid.
; File number	When reading, specify the necessary file number after the :.	With the FDLIST command, file number and file name list can be displayed.

NOTE

When specifying drive number and file name or file number, specify without separating, as follows.
Example) 0 @ ABC or 1 : 5

When displaying set request message, drive number is decided by the system, and ’FD0=’ or ’FD1=’ is displayed, instead of the ’FD=’ message, to check the drive (unit) to be used. If a drive number is specified then, it will be ignored. (FD0 shows drive 0, and FD1, drive 1).

5.2 CONFIGURATION OF COMMAND

(1) General form of command

Operation	Space	Operand
Command name or its abbreviated form (4 leading characters)	–	List of one or more parameters delimited by delimiter symbol ' , ' (comma).

A command name consists of plural alphabetic characters, and it can be abbreviated by four leading characters.

An operand consists of parameters peculiar to commands and parameters specified in floppy disk mounting request message.

(2) Execution of operands and commands

If operands are fully designated, a command is executed without displaying any floppy disk mounting request message.

However, a certain command may require many parameters. If these parameters cannot be recalled, specify the command name only. Necessary parameters are indicated in the floppy disk mounting request message. Accordingly, parameters can be input from the keyboard according to this display. The message may be displayed twice separately according to commands. (Old and new names are requested separately in RENAME command, for example.)

5.3 FDLIST COMMAND — FILE ATTRIBUTE DISPLAY

This command displays the attributes of files in the floppy disk, such as file name, file size, etc.

a) Input format

FDLIST { [D,] [P,] [S,] [F,] / [L,] }

[Drive No.] { @ file name / : file No. }

b) Operand

D : Display of file creation date consisting of 6 characters
P : Identification display of protection files
S : Display of file size
F : Display of size of unoccupied area
L: Executes all display by parameters D, P, S, F.

@ file name	Displays a file having the designated file name or designated
: file No.	file number only. If this designation is omitted, all files are treated as processing objects.

c) This command displays the information (attributes) on the floppy disk files.

If no attribute to be displayed is designated, the file number, file name, and multi–volume number only are displayed. The following example shows the display of all information (L designation)

```
NO.   FILE NAME        V.  DATE SIZE  P.
001   DATA1                830928 72  P
002   DATA2                831028 60
003   DATA3                831028  8  P
**** DELETED FILE ****       10
005   DATA4                901022 10  P
006   DATA5                901022  5
        FILE    USED AREA       = 155
        DELETED FILE AREA  =  10
        FREE    AREA            = 1019
```

NOTE
The numeric characters shown in SIZE, FILE USED AREA, DELETED FILE AREA, and FREE AREA are displayed assuming that 256 characters are 1.

5.4 RENAME COMMAND — FILE ATTRIBUTE CHANGE

This command designates a change of the file name, file creation date, and the designation of protection file.

a) Input format

```
                      { @ file name      [ { P  ]
RENAME  [drive No.]   {  : file No.      [ { NP ]  [,/date] [@ new file name]
└───────────────────────────────────┘  └──────────────────────────────────┘
   Designation of file to be changed       Designation of file to be changed
```

b) Function

File attributes are renewed when they are designated by operand parameters. Attributes which are not designated are stored as they are. Protection files can be cancelled, but neither dates nor file names are changeable. The designation of protection files must be cancelled once before changing their attributes.

If all operands are omitted, the system displays an input message to request the designation of a file to be changed. When the file to be changed is designated by keying operation, a message is displayed to input attributes of the file to be changed by keying operation. Designate new data.

If the file to be changed only is designated together with the command, the system asks the file attributes to be changed.

(RENAME: 5 <NL>, for example)

Old attributes (B) and new attributes (A) are displayed by executing this command as shown below, for example.

Example)

```
RENAME :3, @ NEWNAME <F11>NL>
  RENA : 3,@NEWNAME
  NO. FILE NAME V. DATE P.
B: 003 DATA3        901020
A: 003 NEWNAME      901020
```

5.5 SCRATCH COMMAND — DELETION OF FILES

This command deletes files of floppy disk.

a) Input format

$$\text{SCRATCH [drive No.]} \left\{ \begin{array}{l} \text{@ file name} \\ \text{: file No.} \end{array} \right.$$

b) Function

This command deletes the designated file. Even if the file name is displayed by FDLIST, the file name is not displayed any longer. The area occupied by the deleted file must be released by CONDENSE command before writing new data into the area.

5.6 CONDENSE COMMAND — RELEASE OF DELETED AREA

This command releases the deleted file area to be employable.

a) Input format

CONDENSE [drive No.]

b) Function

The area occupied by the file deleted by SCRATCH command cannot be employed for writing new data under that condition. By executing this command, all unemploy–able areas can be released. Since it takes time to execute this command, it is recommended to arrange these areas when there are many files to be deleted and the residual capacity of the floppy disk is small.

5.7 REMOVE COMMAND — FILE COPY

This command copies files to another floppy disk by using two floppy disk units.

a) Input format

REMOVE [{M / A,}] [input drive No.] [@ file name / file No.] [{INT / ADD}] [,{P / NP}, [/date]] [, @ new file name]

Designation of copying method (for [{M / A,}])

Input designation (for [input drive No.] [@ file name / file No.])

Output designation (for [{INT / ADD}] [,{P / NP}, [/date]] [, @ new file name])

b) Operand

No. M,A designation:

Specified file is copied with specified file attributes.

M (manual):

Copies designated files one by one.

Floppy disk mounting request message is displayed every file to ask if the file is to be copied or not. If input file is not designated, all files becomes the object to be copied.

If new file name is not designated, input file name is produced. Accordingly, the output designation of REMOVE command is meaningless, because the request for output designation is performed every file.

i) When a file name is designated as an input;

All coincident files having the designated length are treated as processing objects. If @A is desig–nated, for example, all files starting with A are asked.

ii) When a file number is designated as an input;

Files from a file having the designated file number to the last file are treated as processing objects.

A (auto):

All files conforming to the designated conditions are copied. The output designation file name is meaningless. However, P, NP and / date are designated to all copied files with new attributes.

i) If the file name or file number is not designated by input designation, all files of the input floppy disk are copied.

ii) If the file name or file number is designated in input designation, processing is done in the same way as in M designation.

c) Function

This command copies floppy disk files to another floppy disk. Files are output to a floppy disk opposite to the drive number (0, if omitted) designated by input designation. These files are copied by the following three methods.

i) Copy of one file only (Neither M nor A is designated.)

(Ex. 1) REMOVE : 3,P, @ NEW <NL>

In this example, the input/output designations are as follows; Input designation : 3rd file of drive No.0 floppy disk

Output designation : File name is ”NEW”, drive No.1 floppy disk with protection

ii) The system asks every file to check if the file is to be copied or not. (M designation)

(Ex. 2) REMOVE M,1 @ A <NL>

In this example, the system asks to copy or not every file with file name starting with ”A” of drive No.1 floppy disk.

A change of attributes such as file name, date, and file protection can be designated.

iii) All designated files are copied. (A designation)

(Ex. 3) REMOVE A,: 3, INT,/830920 <NL>

In this example, files with file name starting with ”A” and with file No.3 and later of drive No.0 floppy disk are copied to drive No.1 floppy disk from the head of it with designated date ”830930”. The copied file names cannot be changed in this method.

When this command is executed, input file attributed (I) and output file attributes (O) are displayed. The next display example shows the execution of ’REMOVE A, 1 @TO, INT, P <NL>’.

NO.	FILE NAME			V.DATE	P.
I:001	T01 ZX	1.	100/40	830920	
O:001	T01 ZX	1.	100/40	830920	P
I:002	T02 ZX	1.	150/50	830920	
O:002	T02 ZX	1.	150/50	830920	P
I:003	T04 ZX	1.	100/50	830920	
O:003	T04 ZX	1.	100/50	830920	P
I:004	T05 ZX	1.	20/50	830920	
O:004	T05 ZX	1.	20/50	830920	P

If REMOVE command is only designated, key in operands according to the request message. The following are general designation format for file copy.

i) Without M, A designation (One file is copied.)

$$\text{REMOVE}\ \left[\left\{\begin{matrix}0\\ \text{—}\\ 1\end{matrix}\right.\right]\ \left\{\begin{matrix}@\ \text{file name}\\ \\ :\ \text{file No.}\end{matrix}\right.\ \left[,\left\{\begin{matrix}\text{INT}\\ \\ \underline{\text{ADD}}\end{matrix}\right.\right]\ \left[,\left\{\begin{matrix}\text{P}\\ \\ \text{NP}\end{matrix}\right.\right]$$

[,/ date] [, @ new file name] <NL>

ii) With M designation (request message is displayed for each objected file.)

$$\text{REMOVE}\ \ \text{M,}\left[\left\{\begin{matrix}0\\ \text{—}\\ 1\end{matrix}\right.\right]\ \left\{\begin{matrix}@\ \text{file name}\\ \\ :\ \text{file No}\end{matrix}\right.$$

iii) With A designation (All subjected files are copied.)

$$\text{REMOVE}\ \left[\left\{\begin{matrix}0\\ \text{—}\\ 1\end{matrix}\right.\right]\ \left\{\begin{matrix}@\ \text{file name}\\ \\ :\ \text{file No.}\end{matrix}\right.\ \left[,\left\{\begin{matrix}\text{INT}\\ \\ \underline{\text{ADD}}\end{matrix}\right.\right]\ \left[,\left\{\begin{matrix}\text{P}\\ \\ \text{NP}\end{matrix}\right.\right]$$

[,/ date] <NL>

APPENDIX

A ERROR CODES LIST (FOR FAPT LADDER P–G)

Error codes	Details of errors
01	Sequence program area over
02	No. of divisions has exceeded 99.
03	High level program time over
05	An error block was detected.
07	No designated step number is found.
08	An undefined instruction was specified.
09	An undefined address was specified.
10	Parameter data error
11	An address was employed in OR.STK and AND.STK.
12	An unemployable subroutine number was specified.
13	An unemployable timer number was specified.
14	A comparison error occurred.
15	A jump instruction was specified, exceeding END1 and END2.
16	A common instruction was specified, exceeding END1 and END2.
17	An instruction format error
18	An attempt was made to delete a parameter.
19	An attempt was made to add a parameter.
20	An erroneous system parameter data
21	A parameter was specified in a mode other than subroutine mode.
24	END2 is not specified.
25	WRT instruction is not specified in WRT instruction subroutine.
27	END1 is not specified.
29	A data sent from PMC–SB/SC is in error.
30	R1 key is pressed during data transmission between SYSTEM P series and PMC–SB/SC.
31	Input/output unit error
32	Read error
33	Hardware error of floppy disk
34	No designated file name is found.
41	An error occurred when inputting ROM data from ROM writer.
43	An error occurred when writing ROM data into ROM writer.
44	An error is deleted during comparison between SYSTEM P series–memory data and floppy data.
45	An error occurred when comparing ROM data with ROM writer data.
46	Key input data over
47	No designated symbol name is found.
48	A numeric value was directly specified to address parameters.
49	Counter number error
50	Decode functional instruction error
51	Symbol name (max. 6 characters) over

Error codes	Details of errors
52	Input data error
53	Comment data are in error.
54	Symbol table over
55	Comment data area over
56	Designated symbol name is already employed.
57	Symbol table sequence is in error.
58	Designated symbol name is not found.
59	END1 was detected in COM mode.
60	END1 was detected in JMP mode.
61	END2 was detected in COM mode.
62	END2 was detected in JMP mode.
63	END 3 was detected in COM mode.
64	END 3 was detected in JMP mode.
65	END 3 is not specified.
66	COM functional instruction was specified in COM mode.
67	JMP functional instruction was specified in JMP mode.
68	Message address error
69	Message data area over
70	Message data error
71	No symbol table is prepared.
72	NC model error in title
73	Title number error
74	Title data error
75	I/O port address error
76	Group number error
77	Base number error
78	Slot number error
79	I/O module name error
80	I/O port data are not prepared yet.
81	I/O port data were doubly specified.
82	Specified symbol or address is missing.
83	An invalid unit is loaded in the ROM WRITER or the specification of ROM WRITER does not meet the unit.
84	ROM module type is different from the specified one.
87	Output (or input) module was specified as an input (or output) address.
88	The same slot number was specified in the same group and the same base number.
89	The model of PMC is different.
93	The number of coils is specified by the COM or JMP command. (This causes an error for PMC–SA1 and PMC–SA2.)
150	Parity error of transfer data (check the cable.)
151	Excessive or insufficient data to be transferred (Check the cable.)
152	An EPROM or ROM module is not inserted in the ROM writer, or specification of the ROM writer is invalid.
153	Blank check error (Ultraviolet ray is not sufficiently irradiated or the EPROM, ROM module is defective.)

Error codes	Details of errors
154	Write error (EPROM or ROM module is defective.)
155	Verifyerror (EPROM or ROM module is defective.)
156	Data output level error (EPROM or ROM module is defective.)
157	Timer test error is ROM writer (ROM writer is defective.)
158	I/O test error in ROM write (ROM writer is defective.)
159	A/D converter test error in ROM writer (ROM writer is defective.)
160	Power test error in ROM writer (ROM writer is defective.)
161	Power (VPP) is defective (EPROM, ROM module or ROM writer is defective.)
162	Power supply (VCC) is defective (EPROM, ROM module or ROM writer is defective.)
163	ROM test error in ROM writer (ROM writer is defective.)
164	RAM test error in ROM writer (ROM writer is defective.)
170	An initialization error in the external memory floppy disk.
171	The inputted ROM format data is greater than the specified cassette type. The PASCAL origin of the inputted PASCAL load module is unsuitable.
172	The specified ROM format data cannot be edited with the P–G Mate. Outputting data in the ROM format is possible, however.

B WINDOW FUNCTION DESCRIPTION (PMC–PA1/PA3/SA1/SA2/SA3/SA5/SB/SB2/SB3/SB4/SB5/SB6/SC/SC3/SC4)

B.1 FUNCTION

This window function is a functional instruction by which the data on the CNC is read or is written.

B.2 LOW–SPEED RESPONSE AND HIGH–SPEED RESPONSE OF WINDOW FUNCTION

In the way to process, there are window function high speed and one processed at low speed.

In case of a low–speed response, The data is read or written by the control between CNC and PMC

Therefore, it is necessary to ACT=1 of the window instrucion must be held until the transfer completion information (W1) becomes 1 (interlock).

In a high–speed response, it is not necessity for take the interlock because the data is directly read.

CAUTION

The window instruction of a low–speed response is controlled exclusively with the other window instructions of low–speed response.
Therefore, when the data is read or written continuously, it is necessary to clear ACT of the functional instruction once when the completion information (W1) become 1.
It does not work about ACT=1 of the other window instructions of low–speed response such as W1=1 and ACT=1 of the window instruction of a low–speed response.
The window instruction of a high–speed response is not exclusively controlled like a low–speed response. Therefore, when the data is read or written continuously, yow need not make ACT=0.
The scan number of times to complete the processing is summarized on the following table.

TYPE	SCAN TIMES UNTIL PROCESSING ENDS
LOW	TWO SCAN TIMES OR MORE(This depends on the state of CNC)
HIGH	1SCAN TIME

NOTE

Enter the desired function code (to which 1000 is added when data of the second tool post (HEAD2) is read or written in the TT series, or when data of the second path is read or written in two–path control of the Power Mate–D.
To perform path 3 read/write operation in 3–path control, enter a function code + 2000.

B.3 LIST OF WINDOW FUNCTIONS

Number	Description	Function code	R/W
1	Read CNC system information	0	R
2	Read the tool offset	13	R
3	Write a tool offset ✻Low–speed response	14	W
4	Read the work origin offset *PM	15	R
5	Write work origin offset *PM ✻Low–speed response	16	W
6	Read parameters *SB56 ✻Low–speed response	17	R
7	Write parameters ✻Low–speed response	18	W
8	Read setting data *SB56 ✻Low–speed response	19	R
9	Write setting data ✻Low–speed response	20	W
10	Read custom macro variables *SB56 ✻Low–speed response	21	R
11	Write custom macro variables ✻Low–speed response	22	W
12	Read the CNC alarm state	23	R
13	Read the current program number	24	R
14	Read the current sequence number	25	R
15	Read an actual velocity for controlled axes	26	R
16	Read an absolute position (absolute coordinate value) on controlled axes	27	R
17	Read a machine position (machine coordinate value) on controlled axes	28	R
18	Read a skip operation (G31) stop position (coordinate value) on controlled axes	29	R
19	Read a servo delay amount	30	R
20	Read acceleration/deceleration delay amount on controlled axes	31	R
21	Read modal data	32	R
22	Read diagnosis data *SB56 ✻Low–speed response	33	R
23	Read a feed motor load current value (A/D conversion data)	34	R
24	Reading tool life management data (tool group No.) *PM *21T	38	R
25	Reading tool life management data (number of tool group s) *PM *21T	39	R
26	Reading tool life management data (number of tools) *PM *21T	40	R
27	Reading tool life management data (usable life of tool) *PM *21T	41	R
28	Reading tool life management data (tool usage counter) *PM *21T	42	R
29	Reading tool life management data (tool length compensation No. (1): Tool No.) *PM *21T	43	R
30	Reading tool life management data (tool length compensation No. (2): Tool order No.) *PM *21T	44	R
31	Reading tool life management data (cutter compensation No. (1): Tool No.) *PM *21T	45	R
32	Reading tool life management data (cutter compensation No. (2): Tool order No.) *PM *21T	46	R
33	Reading tool life management data (tool information (1): Tool No.) *PM *21T	47	R
34	Reading tool life management data (tool information (2): Tool order No.) *PM *21T	48	R
35	Reading tool life management data (tool No.) *PM *21T	49	R
36	Reading the actual spindle speed	50	R

Number	Description	Function code	R/W
37	Entering data on the program check screen ⁎low–speed response *PM *21T	150	W
38	Reading clock data (date and time)	151	R
39	Writing torque limit data for the digital servo motor ⁎low–speed response	152	W
40	Reading load information of the spindle motor (serial interface)	153	R
41	Reading a parameter *PM *21T	154	R
42	Reading setting data *PM *21T	155	R
43	Reading diagnosis data *PM *21T	156	R
44	Reading a character string of the CNC program being executed in the buffer *C	157	R
45	Reading the relative position of a controlled axis	74	R
46	Reading the remaining travel	75	R
47	Reading CNC status information	76	R
48	Reading value of the P– code macro variable *SB56 ⁎low–speed response	59	R
49	Writing value of the P– code macro variable ⁎low–speed response	60	W
50	Reading the Tool life management data (Tool life counter type)	160	R
51	Registering the Tool life management data (Tool group) ⁎low–speed response	163	W
52	Writing the Tool life management data (Tool life) ⁎low–speed response	164	W
53	Writing the Tool life management data (Tool life counter) ⁎low–speed response	165	W
54	Writing the Tool life management data (Tool life counter type) ⁎low–speed response	166	W
55	Writing the Tool life management data (Tool length offset number (1): Tool number) ⁎low–speed response	167	W
56	Writing the Tool life management data (Tool length offset num–ber (2): Tool operation sequence number) ⁎low–speed response	168	W
57	Writing the Tool life management data (Cutter compensation number (1): Tool number) ⁎low–speed response	169	W
58	Writing the Tool life management data (Cutter compensation nu–mber (2): Tool operation sequence number) ⁎low–speed response	170	W
59	Writing the Tool life management data (Tool condition (1): Tool number) ⁎low–speed response	171	W
60	Writing the Tool management data (Tool condition (2): Tooloperation sequence number) ⁎low–speed response	172	W
61	Writing the Tool life management data (Tool number) ⁎low–speed response	173	W
62	Reading the Estimate disturbance torque data	211	R
63	Reading the current program number (8–digit program numbers) *PM *21T	90	R
64	Writing (registering) tool life management data (tool group number) *PM *21T	200	R
65	Reading tool life management data (tool length offset number 1) *PM *21T	227	R
66	Reading tool life management data (tool diameter offset number 1) *PM *21T	228	R
67	Reading tool life management data (tool information 1) *PM *21T	201	R
68	Writing tool life management data (tool group number) ⁎low–speed response	202	R
69	Writing tool life management data (tool length offset number 1) ⁎low–speed response	229	W
70	Writing tool life management data (tool radius offset number 1) ⁎low–speed response	230	W
71	Writing tool life management data (tool information 1) ⁎low–speed response	231	W
72	Reading actual spindle speeds	138	R
73	Reading fine torque sensing data (statistical calculation results)	226	R

Number	Description	Function code	R/W
74	Reading fine torque sensing data (store data)	232	R
75	Specification of the number of the program for I/O Link	194	W

*1 Function codes that have R in the R/W column are window read functions specifiable with the WINDR function command. Function codes that have W in the R/W column are window write functions specifiable with the WINDW function command.

*2 For window functions mark with “Low–speed response,” reading and writing parameters, setting data, diagnostic data and so on starts after the PMC receives the response for request of reading and writing from the CNC. On the contrary, the other window functions can read or write data at once in response to the request from PMC.

*3 Functions marked with *PM are not provided for the Power Mate–D or F.

*4 Functions marked with *21T are not provided for the Series 21T.

*5 Functions marked with *SB5/6 support high–speed window response for the SB5/SB6.

*6 Functions marked with *C are not provided for the SB5/SB6.

B.4 FORMATS AND DETAILS OF CONTROL DATA

(1) In the explanation of the window functions, minuses (–) in the data structure fields indicate that input data need not be set in these fields or that output data in these fields is not significant.

(2) All data is in binary unless otherwise specified.

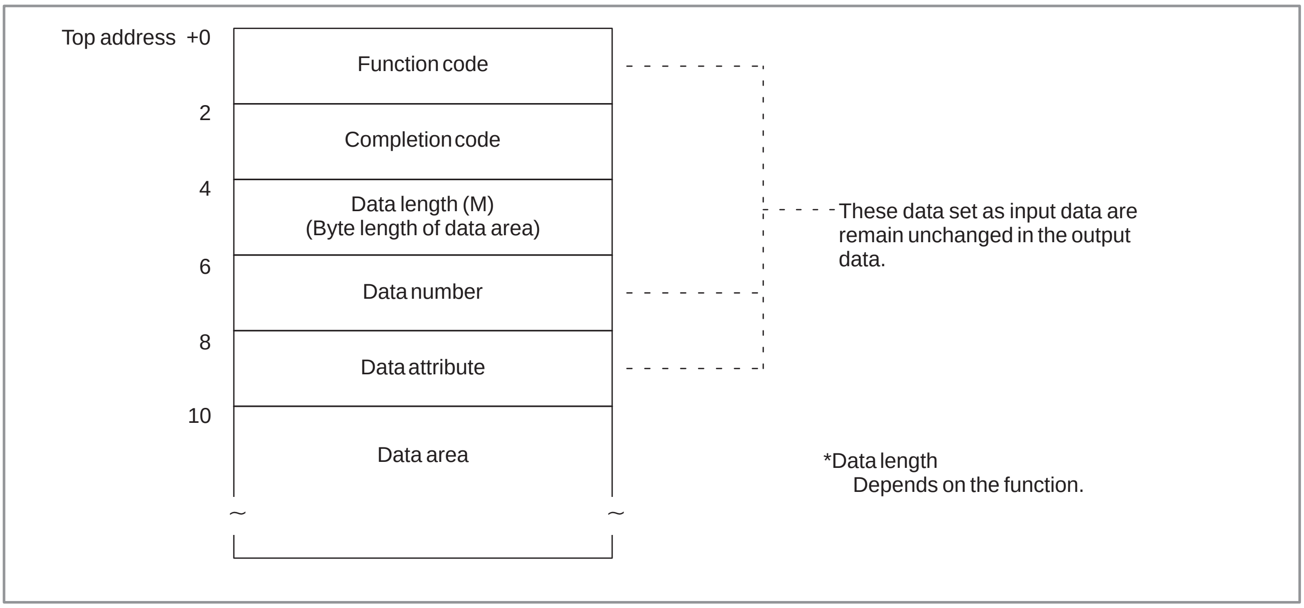
(3) All data block lengths and data lengths are indicated in bytes.

(4) Output data is valid only when window processing terminates normally.

(5) Output data always includes one of the following completion codes. Note, however, that all of the completion codes listed are not always provided for each function.

Completion code	Meaning
0	Normal termination
1	Error (invalid function code)
2	Error (invalid data block length)
3	Error (invalid data number)
4	Error (invalid data attribute)
5	Error (invalid data)
6	Error (necessary option missing)
7	Error (write–protected)

Input and output control data has the following structure.



B.4.1 Reading CNC System Information

[Description]

System information peculiar to the CNC can be read. Such system information includes the series name of the CNC (16 as series name, for example), the machine type applied to the CNC, such s a machining center (M) and a lathe (T), the series code and version of the ROM containing the CNC system software, and the number of controlled axes.

[Input data structure]

Address	Contents
Top address + 0	(Function code) 0
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) 0
8	(Data attribute) 0
10	(Data area) — (Need not be set)
42	

[Completion codes]

0 : CNC system information has been read normally.

[Output data structure]

Address	Item	Value
Top address + 0	(Function code) 0	
2	(Completion code) 0 (Always terminates normally.)	
4	(Data length) 14	
6	(Data number) —	
8	(Data attribute) —	
10	CNC series name (2 bytes)	ASCII characters (16)
12	Machine type M/T/TT (2 bytes)	ASCII characters (M, T. TT, . . .)
14	ROM series of CNC system software(4 bytes)	ASCII characters (B 0 0 0 1, . . .)
18	ROM version of CNC system software(4 bytes)	ASCII characters (0 0 0 1, 0 0 0 2, , . .)
22	Number of controlled axes (2 bytes)	ASCII characters (2, 3, 4, . . .)

NOTE

1 Data is stored from the upper digit in each lower byte.
2 In the Power Mate–D and –F, the data corresponding to the CNC series name and machine type are left as spaces.
3 In two–path control of the Power Mate–D, the data for the first path is the same as that for the second path.

B.4.2 Reading a Tool Offset

[Description]

A tool offset value recorded in the CNC can be read.

Wear offset data, geometry offset data, cutter compensation data, and tool length offset data can be read as a tool offset.

[Input data structure]

Top address + 0	(Function code) 13
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N = offset number)
8	(Data attribute) M (M = offset type)
10	(Data area) — (Need not be set) ~
42	

(a) Offset types (for machining centers, Power Mate–D, F)

	Cutter	Tool length
Wear	0	2
Figure	1	3

If the type of tool offset need not be specified, enter 0.

NOTE
In the Power Mate–D and –F, read tool offsets without specifying the classification (i.e. cutter compensation, tool length, tool wear, and tool geometry).

(b) Offset types (for lathes)

	X axis	Z axis	Tool tip R	Virtual tool tip	Y axis
Wear	0	2	4	6	8
Figure	1	3	5	7	9

[Completion codes]

0 : The tool offset has been read normally.

3 : The offset number specified for reading is invalid. (This completion code is returned when the specified offset number data is not from 1 to the maximum number of offsets.)

4 : There are mistakes in the data attribute that specifies the type of the offset to be read.

6 : For the offset number specified for reading, an additional tool offset number option is required, but it is missing.
The offset number is not available for Power Mate–D/F.

[Output data structure]

Top address + 0	(Function code) 13	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (Normally set to 4) (L: Byte length of offset value)	
6	(Data number) N (N = offset number)	
8	(Data attribute) M (M = offset type),,	Value
10	Tool offset value	Signed binary (A negative value is represented in 2's complement.) Upper 3 bytes are always "0" for virtual tool tip

Output data unit

<table>
<tr><th colspan="2"></th><th>Input system</th><th>Increment system IS–B</th><th>Increment system IS–C</th></tr>
<tr><td colspan="2" rowspan="2">Machining center system
Power Mate–D, F</td><td>mm, deg system</td><td>0.001</td><td>0.0001</td></tr>
<tr><td>inch system</td><td>0.0001</td><td>0.00001</td></tr>
<tr><td rowspan="4">Lathe system</td><td>Radius specification</td><td rowspan="2">mm, deg system</td><td>0.001</td><td>0.0001</td></tr>
<tr><td>Diameter specification</td><td>0.002</td><td>0.0002</td></tr>
<tr><td>Radius specification</td><td rowspan="2">inch system</td><td>0.0001</td><td>0.00001</td></tr>
<tr><td>Diameter specification</td><td>0.0001</td><td>0.00001</td></tr>
</table>

B.4.3 Writing a Tool Offset (✻Low–speed Response)

[Description]

The tool offset value can be directly written into the CNC.

Wear offset data, geometry offset data, cutter compensation data, and tool length offset data can be written as a tool offset.

[Input data structure]

Address		Value
Top address + 0	(Function code) 14	
2	(Completion code) — (Need not to be set)	
4	(Data length) 4	
6	(Data number) N (N = offset number)	
8	(Data attribute) M (M = offset type)	
10	Tool offset value	Signed binary (A negative value is represented in 2's complement.) Upper 3 bytes are always "0" for virtual tool tip

(a) Offset types (for machining centers, Power Mate–D, F)

	Cutter	Tool length
Wear	0	2
Figure	1	3

If the type of tool offset need not be specified, enter 0.

In the Power Mate–D and –F, write tool offsets without specifying the classification (i.e. cutter compensation, tool length, tool wear, and tool geometry).

(b) Offset types (for lathes)

	X axis	Z axis	Tool tip R	Virtual tool tip	Y axis
Wear	0	2	4	6	8
Figure	1	3	5	7	9

Input data unit

<table>
<tr><th colspan="2"></th><th>Input system</th><th>Increment system IS–B</th><th>Increment system IS–C</th></tr>
<tr><td colspan="2" rowspan="2">Machining center system
Power Mate–D, F</td><td>mm, deg system</td><td>0.001</td><td>0.0001</td></tr>
<tr><td>inch system</td><td>0.0001</td><td>0.00001</td></tr>
<tr><td rowspan="4">Lathe system</td><td>Radius specification</td><td rowspan="2">mm, deg system</td><td>0.001</td><td>0.0001</td></tr>
<tr><td>Diameter specification</td><td>0.002</td><td>0.0002</td></tr>
<tr><td>Radius specification</td><td rowspan="2">inch system</td><td>0.0001</td><td>0.00001</td></tr>
<tr><td>Diameter specification</td><td>0.0002</td><td>0.00002</td></tr>
</table>

[Completion codes]

0 : The tool offset has been written normally.

2 : The data byte length for the tool offset specified for writing is invalid. (It is not set to 4.)

3 : The offset number specified for writing is invalid. (This completion code is returned when the specified offset number data is not from 1 to the maximum number of offsets.)

4 : There are mistakes in the data attribute that specifies the type of the offset to be written.

6 : For the offset number specified for writing, the additional tool offset number option is required, but it is missing.
The specified offset number is out of range. (Power Mate–D, F)

[Output data structure]

Address	Field	Value
Top address + 0	(Function code) 14	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L: Input data)	
6	(Data number) N (N = Input data)	
8	(Data attribute) M (Input data)	
10	Tool offset value: Input data	Signed binary (A negative value is represented in 2's complement.)

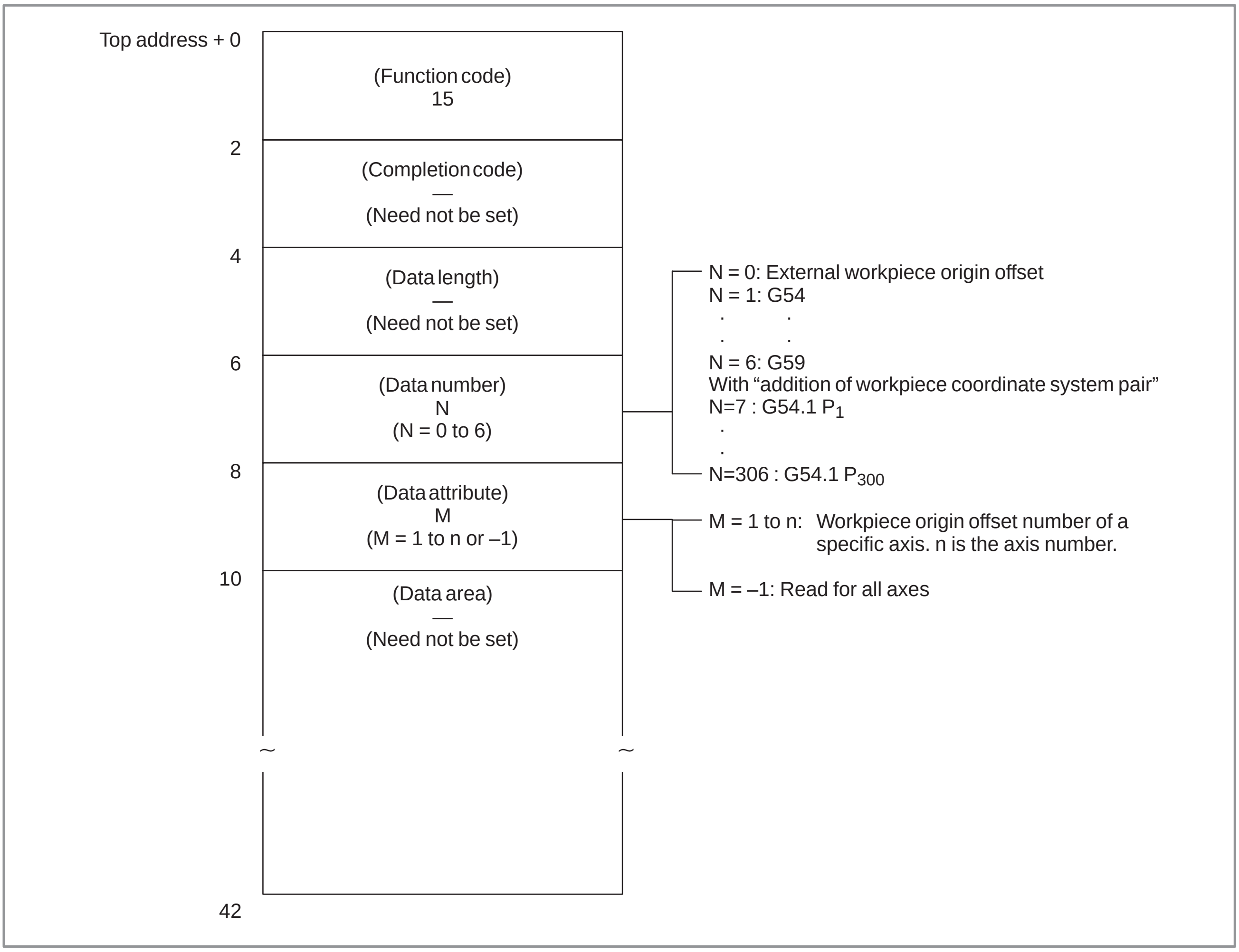
B.4.4 Reading a Workpiece Origin Offset Value (not Supported by the Power Mate–D or –F)

[Description]

The workpiece origin offset recorded in the CNC can be read.

A workpiece origin offset is provided for each controlled axis (the first axis to the eighth axis) in the CNC. Either the workpiece origin offset for a specific axis can be read, or the workpiece origin offsets for all axes can be read at one time. If the additional axis option is not provided, however, the workpiece origin offset for the additional axis cannot be read.

[Input data structure]



[Completion codes]

0 : The workpiece origin offset has been read normally.

3 : The specified data number is invalid because the number is not from 0 to 6.

4 : The specified data attribute is invalid because the attribute data is neither –1 nor a value from 1 to n (n is the number of axes). Alternatively, the specified axis number is greater than the number of controlled axes.

6 : There is no workpiece coordinate shift option added.

[Output data structure]

Top address + 0	(Function code) 15	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L: Byte length of the workpiece origin offset value)	L = 4: The workpiece origin offset value for a specific axis is read. L = 4*n: Workpiece origin offsets for all axes are read.
6	(Data number) N (N = Input data)	
8	(Data attribute) M (M = Input data)	Value
10	Workpiece origin offset value	Signed binary number (A negative value is represented in 2's complement.)

Output data unit

		Input system	Increment system IS–B	Increment system IS–C
Machining center system Power Mate–D, F		mm, deg system	0.001	0.0001
		inch system	0.0001	0.00001
Lathe system	Radius specification	mm, deg system	0.001	0.0001
	Diameter specification		0.002	0.0002
	Radius specification	inch system	0.0001	0.00001
	Diameter specification		0.0001	0.00001

B.4.5
Writing a Workpiece Origin Offset Value (✻Low–speed Response) (not Supported by the Power Mate–D or –F)

[Description]

Data can be written directly as a workpiece origin offset value in the CNC.

A workpiece origin offset is provided for each controlled axis (the first axis to the eighth axis) in the CNC. Either the workpiece origin offset value for a specific axis can be written, or the workpiece origin offset values for all axes can be written at one time. If the additional axis option is not provided, however, the workpiece origin offset value for the additional axis cannot be written.

[Input data structure]

Address	Field	Notes
Top address + 0	(Function code) 16	
2	(Completion code) — (Need not be set)	
4	(Data length) L (L: Byte length of the workpiece origin offset value)	L = 4 : Workpiece origin offset value for a specific axis is written. L = 4*n Workpiece origin offset values for all axes are written.
6	(Data number) N (N = 0 to 6)	N = 0: External workpiece origin offset N = 1: G54 . . . . N = 6: G59
8	(Data attribute) M (M = 1 to n, or –n)	M = 1 to n: Workpiece origin offset number of a specific axis. n is the axis number. M = –1: Write for all axes
		Value
10	Workpiece origin offset value	Signed binary (A negative value is represented in 2's complement.)

Input data unit

<table>
<tr><th colspan="2"></th><th>Input system</th><th>Increment system IS–B</th><th>Increment system IS–C</th></tr>
<tr><td colspan="2" rowspan="2">Machining center system
Power Mate–D, F</td><td>mm, deg system</td><td>0.001</td><td>0.0001</td></tr>
<tr><td>inch system</td><td>0.0001</td><td>0.00001</td></tr>
<tr><td rowspan="4">Lathe system</td><td>Radius specification</td><td rowspan="2">mm, deg system</td><td>0.001</td><td>0.0001</td></tr>
<tr><td>Diameter specification</td><td>0.002</td><td>0.0002</td></tr>
<tr><td>Radius specification</td><td rowspan="2">inch system</td><td>0.0001</td><td>0.00001</td></tr>
<tr><td>Diameter specification</td><td>0.0002</td><td>0.00002</td></tr>
</table>

[Completion codes]

0 : The workpiece origin offset has been written normally.

2 : The specified data length is invalid.

3 : The data number is invalid because the specified number is not from 0 to 6.

4 : The specified data attribute is invalid because the attribute data is neither –1 nor a value from 1 to n (n is the number of axes). Alternatively, the specified axis number is greater than the number of controlled axes.

6 : There is no workpiece coordinate shift option added.

[Output data structure]

Top address + 0	(Function code) 16	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L: Input data)	
6	(Data number) N (N = Input data)	
8	(Data attribute) M (M = Input data)	Value
10	Workpiece origin offset value	Signed binary number (A negative value is represented in 2's complement.)

B.4.6 Reading a Parameter (*Low–speed Response)

[Description]

Parameter data in the CNC can be read.

There are four types of parameters in the CNC: Bit parameters having a definite meaning for each bit, byte parameters holding 1–byte data, word parameters holding 2–byte data, and double word parameters holding 4–byte data. Therefore, the length of the read data varies according to the parameter number specified.

Note that bit parameters cannot be read in bit units. The eight bits (one byte) for a parameter number must be read at a time.

For axis parameters, data for a specific axis can be read, or data for all axes can be read at a time.

Specify pitch error compensation data in data Nos. 10000 to 11023 (1024 points in total).

For details of parameter data, refer to the Operator's manual of the CNC.

[Input data structure]

Top address + 0	(Function code) 17	
2	(Completion code) — (Need not be set)	
4	(Data length) — (Need not be set)	
6	(Data number) N (N = parameter number)	
8	(Data attribute) M (M = 1 to n or –1)	M = 0: No axis M = 1 to n: A specific axis M = –1: All axes
10	(Data area) — (Need not be set)	When all axes are specified by spindle parameters (parameters 4000 to 4799), only two axes are specified.
~	~	
42		

[Completion codes]

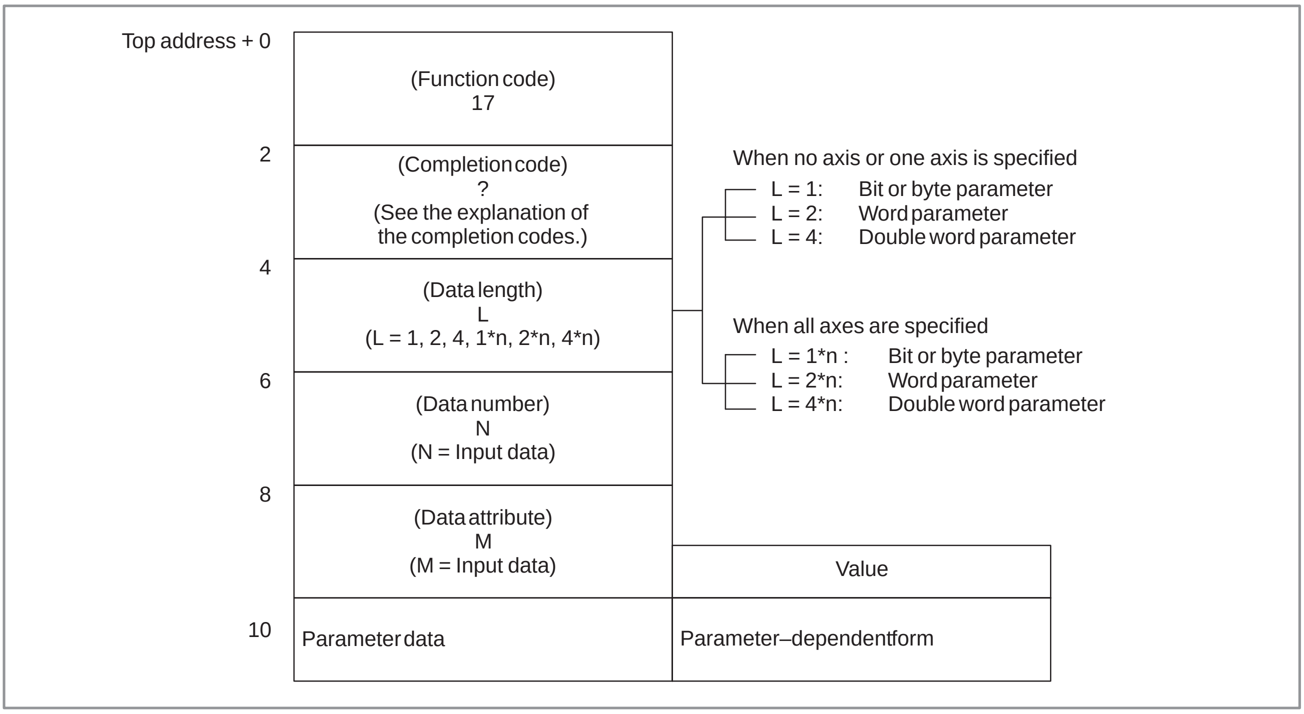
0 : Parameter data has been read normally.

3 : The parameter number specified for reading is invalid.

4 : The specified data attribute is invalid because it is neither 0, –1, nor a value 1 to n (n is the number of axes).

6 : Although a certain option, such as the pitch error compensation option, is required for the data of the parameter number specified for reading, it is not provided.

[Output data structure]



For the SB5/SB6, macro executor parameters 9000 to 9011 cannot be read.

B.4.7 Writing a Parameter (✻Low–speed Response)

[Description]

Data can be written in a parameter in the CNC.

There are four types of parameters in the CNC: Bit parameters having a definite meaning for each bit, byte parameters holding 1–byte data, word parameters holding 2–byte data, and double word parameters holding 4–byte data. Therefore, the length of the written data varies according to the parameter specified.

Note that bit parameters cannot be written in bit units. The eight bits (one byte) for the parameter number must be written at a time. This means that when a bit needs to be written, the whole data for the corresponding parameter number shall be read first, modify the target bit in the read data, then the data shall be rewritten.

For axis parameters, data for a specific axis can be read, or data for all axes can be read at a time.

For details of parameter data, refer to the Operator's manual of the CNC.

Some parameters cause a P/S alarm 000 when data is written. (The power must be turned off before continuing operation.)

[Input data structure]

Top address + 0	(Function code) 18	
2	(Completion code) — (Need not be set.)	
4	(Data length) L (L = 1, 2, 4, 1*n, 2*n, 4*n)	When no axis or one axis is specified L = 1: Bit or byte parameter L = 2: Word parameter L = 4: Double word parameter When all axes are specified L = 1*n: Bit or byte parameter L = 2*n: Word parameter L = 4*n: Double word parameter
6	(Data number) N (N = parameter number)	
8	(Data attribute) M (M = 0, 1 to n, or –1)	M = 0: No axis M = 1 to n: A specific axis M = –1: All axes When all axes are specified by spindle parameters (parameters 4000 to 4799), only two axes are specified.
		Value
10	Parameter data	Parameter–dependent form

[Completion codes]

0 : Parameter data has been written normally.

2 : The data byte length of the parameter specified for writing is invalid.

3 : The parameter number specified for writing is invalid.

4 : The specified data attribute is invalid because it is neither 0, –1, nor a value from 1 to n (n is the number of axes).

6 : Although a certain option, such as the pitch error compensation option, is required for the data of the parameter number specified for writing, it is not provided.

[Output data structure]

Top address + 0	(Function code) 18	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = Input data)	
6	(Data number) N (N = Input data)	
8	(Data attribute) M (M = Input data)	Value
10	Parameter data: Input data	Parameter–dependentform

B.4.8 Reading Setting Data (✳Low–speed Response)

[Description]

The CNC setting data can be read.

There are four types of setting data in the CNC: Bit setting data having a definite meaning for each bit, byte setting data stored in bytes, word setting data stored in 2–byte units, and double–word setting data stored in 4–byte units. Therefore, the length of the read data varies according to the setting data specified.

Note that bit setting data cannot be read in bit units. The eight bits (one byte) for the setting data number must be read at a time.

For axis parameters, data for a specific axis can be read, or data for all axes can be read at a time.

For details of setting data, refer to the Operator's manual of the CNC.

[Input data structure]

Top address + 0	(Function code) 19	
2	(Completion code) — (Need not be set)	
4	(Data length) — (Need not be set)	
6	(Data number) N (N = Setting data number)	
8	(Data attribute) M (M = 1 to n or –1)	M = 0 : No axis M = 1 to n: A specific axis M = –1: All axes
10	(Data area) — (Need not be set) ~	
42		

[Completion codes]

0 : Setting data has been read normally.

3 : The setting number specified for reading is invalid.

4 : The specified data attribute is invalid because it is neither 0, –1, nor a value from 1 to n (n is the number of axes).

[Output data structure]

Top address + 0	(Function code) 19	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 1, 2, 4, 1*n, 2*n, 4*n)	
6	(Data number) N (N = Input data)	
8	(Data attribute) M (M = Input data)	Value
10	Setting data	Parameter–dependentform

When no axis or one axis is specified
- L = 1: Bit or byte parameter
- L = 2: Word parameter
- L = 4: Double word parameter

When all axes are specified
- L = 1*n: Bit or byte parameter
- L = 2*n: Word parameter
- L = 4*n: Double word parameter

B.4.9 Writing Setting Data (✱Low–speed Response)

[Description]

Data can be written as setting data in the CNC.

For details of setting data, refer to the Operator's manual of the CNC.

[Input data structure]

Address	Content	
Top address + 0	(Function code) 20	
2	(Completion code) — (Need not be set.)	
4	(Data length) L (L = 1, 2, 4, 1*n, 2*n, 4*n)	When no axis or one axis is specified L = 1: Bit or byte parameter L = 2: Word parameter L = 4: Double word parameter When all axes are specified L = 1*n: Bit or byte parameter L = 2*n: Word parameter L = 4*n: Double word parameter
6	(Data number) N (N = Setting data number)	
8	(Data attribute) M (M = 0, 1 to n, or –1)	M = 0: No axis M = 1 to n: A specific axis M = –1: All axes
		Value
10	Setting data	Setting data–dependent form

[Completion codes]

0 : Setting data has been written normally.

2 : The byte length of the setting data specified for writing is invalid.

3 : The setting data number specified for writing is invalid.

4 : The specified data attribute is invalid because it is neither 0, –1, nor a value from 1 to n (n is the number of axes).

5 : Data exceeding the allowable range was specified as setting data to be written. For example, when data outside the range from 0 to 3 is specified as the setting data to be written for I/O data, this completion code is returned.

[Output data structure]

Top address + 0	(Function code) 20	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (N = Input data)	
6	(Data number) N (N = Input data)	
8	(Data attribute) M (M = Input data)	Value
10	Setting data: Input data	Setting data–dependent form

B.4.10 Reading a Custom Macro Variable (✱Low–speed Response)

[Description]

A custom macro variable in the CNC can be read.

Custom macro variables may or may not be read depending on the variable type.

(1) Local variables

Local variables (#1 to #33) cannot be read.

(2) Common variables

Common variables (#100 to #149 and #500 to #531) can be read in floating–point representation. When the option to add common variables is provided, however, common variables range from #100 to #199 and #500 to #999.

> **NOTE**
> 1 Power Mate–D (two–path control), Power Mate–F: #100 to 199, #500 to 699.
> Memory module A of one–path control: #100 to #149, #500 to #531.
> Memory module B/C of one–path control: #100 to #199, #500 to #699.
> 2 On the Power Mate *i*–MODEL D/H, common variables #100 to #199 and #500 to #699 can be read and written.
> Set the variable number within these ranges as the data number of the input data, and read and write custom macro variables.

(3) System variables

System variables (#1000 and up) can be read in floating–point representation.

For details of the custom macro variables, refer to the Operator's Manual for the CNC.

> **NOTE**
> For the SB5/SB6, system variables cannot be read.

[Input data structure]

Address	Content
Top address + 0	(Function code) 21
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N = Custom macro variable number)
8	(Data attribute) M (M: Number of decimal places)
10	(Data area) — (Need not be set)
~	~
42	

[Completion codes]

0 : The custom macro variable has been read normally.

3 : The number of a custom macro variable that cannot be read was specified as the data number. Only common variables can be read as custom macro variables by this library command.

5 : The custom macro variable is not within the range from 0.0000001 to 99999999.

6 : The custom macro option is not provided.
The specified variable number is out of range. (Power Mate–D, F)

[Output data structure]

Address	Data	Value
Top address + 0	(Function code) 21	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L: Byte length of custom macro variable data)	
6	(Data number) N (N = Input data)	
8	(Data attribute) M (M: Number of decimal places)	
10	Custom macro variable data (4 bytes) Mantissa (custom macro B)	Signed binary (A negative value is represented in 2's complement.)
14	Custom macro variable data (2 bytes) Exponent (custom macro B): The number of decimal digits	Signed binary 0 to 8 (no negative values)

L = 6: Custom macro B
The mantissa of a floating–point number is indicated in 4 bytes, and the exponent is indicated in 2 bytes.

M = 0: The number of decimal places is not specified.
M = 1≦ n≦ 7:
The number of decimal places is specified. n stands for the number of decimal places.

B.4.11 Writing a Custom Macro Variable (✻Low–speed Response)

[Description]

Data can be written in a custom macro variable in the CNC.

For details of common variables, refer to the Operator’s manual of the CNC.

[Input data structure]

Offset	Data	Value
Top address + 0	(Function code) 22	
2	(Completion code) — (Need not be set)	
4	(Data length) L (L: Byte length of custom macro variable data)	L = 6: Custom macro B The mantissa of a floating–point number is indicated in 4 bytes, and the exponent is indicated in 2 bytes.
6	(Data number) N (N = Custom macro variable number)	
8	(Data attribute) 0	Value
10	Custom macro variable data (4 bytes) Mantissa (custom macro B)	Signed binary (A negative value is represented in 2’s complement.)
14	Custom macro variable data (2 bytes) Exponent (custom macro B): The number of decimal digits	Signed binary (A negative value is represented in 2’s complement.)

[Completion codes]

0 : The custom macro variable has been written normally.

2 : The specified data length is invalid because it is not 6.

3 : A custom macro variable number that cannot be written as the data number was specified.

6 : The custom macro option has not been provided.

The specified variable number is out of range. (Power Mate–D, F)

[Output data structure]

Top address	Field	Value
+ 0	(Function code) 22	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L: Input data)	
6	(Data number) N (N = Input data)	
8	(Data attribute) — (Need not be set)	Value
10	Custom macro variable data: Input data Mantissa (custom macro B)	Signed binary (A negative value is represented in 2's complement.)
14	Custom macro variable data: Input data Exponent (custom macro B): The number of decimal digits	Signed binary

B.4.12 Reading the CNC Alarm Status

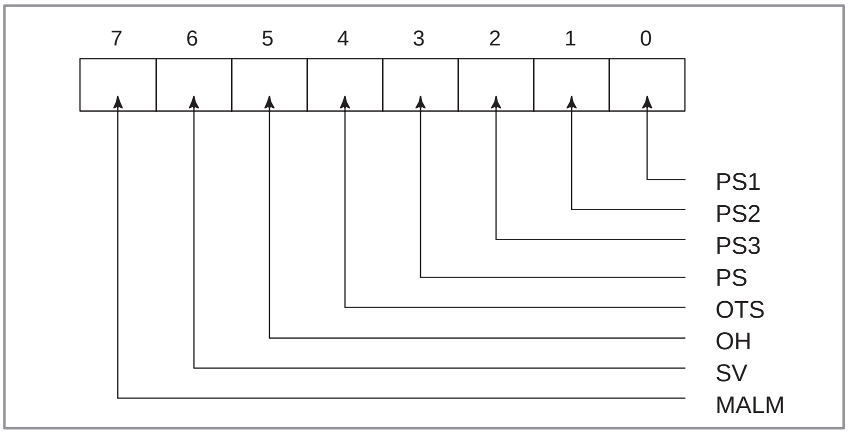
B.4.12.1 Except Power Mate–D/F/H

[Description]

When the CNC is in the alarm status, the alarm status data can be read.

The following alarm status data can be read:

(1) First byte of alarm status data



PS1 : P/S alarm 100 (PWE (parameter write enable) is set to 1.)

PS2 : P/S alarm 000 (Turn off the power before continuing operation. Some parameters activate this alarm status when they are written.)

PS3 : P/S alarm 101 (The part program recording area is disordered. This alarm is activated when the power to the CNC is turned off during part program editing or reading of a machining program. To release the alarm, then press the RESET key while holding down the PROG key.)

PS : A P/S alarm other than the above alarm is generated

OTS : Stroke limit alarm

OH : Overheat alarm

SV : Servo alarm

MALM: Memory alarm

(2) Second byte of alarm status data

7	6	5	4	3	2	1	0
(Not used)	(Not used)	(Not used)	(Not used)	(Not used)	(Not used)	SPA	APAL

APAL: APC alarm

SPA : Spindle alarm

[Input data structure]

Address	Content
Top address + 0	(Function code) 23
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) 0
8	(Data attribute) 0
10	(Data area) — (Need not be set) ~
42	

B.4.12.2 For Power Mate–D/F/H

(1) Overview

PMC application programs can read CNC alarm information.

(2) Alarm information

1) Alarm status

Information concerning the alarm type

2) Detailed alarm

Information concerning the alarm number and axis information

(3) Input data configuration

Top address+0	+2	+4	+6	+8	+10
Function code	Completion code	Data length	Data number	Data attribute	Data area

Function code : 23 (fixed)

Completion code : No specification required.

Data length : No specification required.

Data number : Number of alarms which can be stored. (Up to 30). If 31 or more are specified, the value is assumed to be 30.

Data attribute : Other than 0 : Alarm status information

: Detailed alarm information, indicated in two–byte bit–type data described below (multiple bits can be specified.)

Data area : No specification required.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	bit

Bit 0 : P/S alarm 100 (PS1)

(PWE, parameter write enable, is set to 1.)

Bit 1 : P/S alarm 000 (PS2)

(Turn off the power. Writing data into certain parameters may cause this alarm.)

Bit 2 : P/S alarm 101 (PS3)

(Part program storage has been disrupted. This alarm is issued when the CNC is turned off during tape editing or machining program reading. To release this alarm, press the RESET key while holding down the PROG key.)

Bit 3 : A P/S alarm (PS) other than those described above has been issued. (Up to 255)

Bit 4 : Stroke limit alarm (OTS)

Bit 5 : Overheat alarm (OH)

Bit 6 : Servo alarm (SV)

Bit 7 : Not used

Bit 8 : APC alarm (APAL)

Bit 9 : Spindle alarm (SPA)

Bit 10 : P/S alarm 5000 or greater (PS_2)

Bit 11 to Bit 15 : Not used

Data attribute : 0 : Alarm status information

(4) Output data configuration

Top address+0	+2	+4	+6	+8	+10
Function code	Completion code	Data length	Data number	Data attribute	Data area

Function code : 23 (fixed)

Completion code : Always 0.

Data length : 2 when the input data attribute is set to 0 and no alarm is issued.

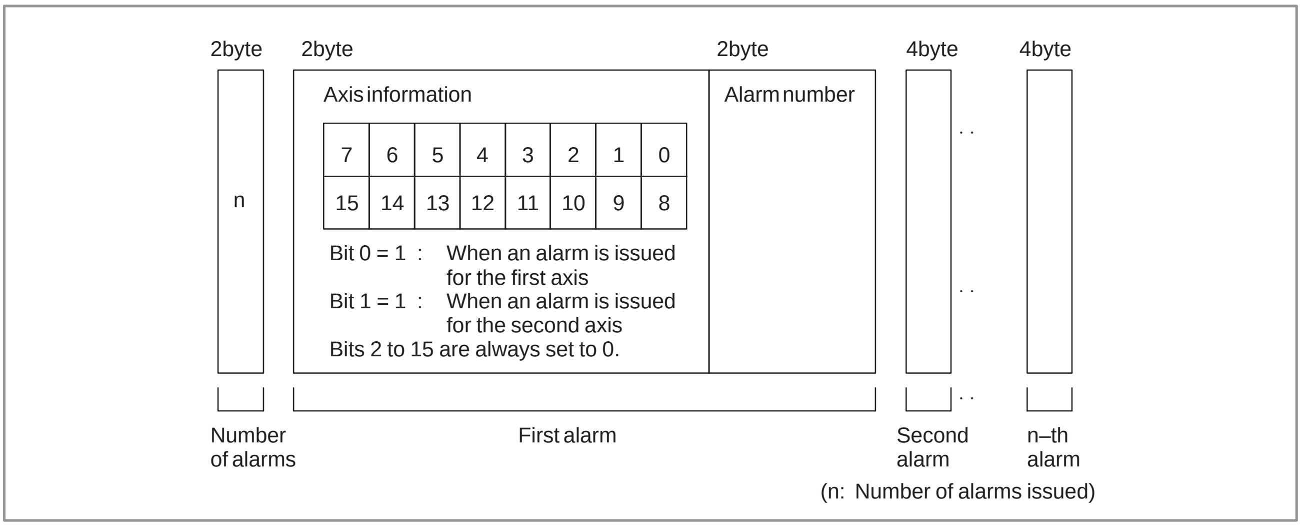
2 + 4*n when the input data attribute is set to other than 0 (n stands for the number of alarms issued).

Data number : Same as that for the input data.

Data attribute : Same as that for the input data.

Data area : Two–byte bit–type data when the input data attribute is set to 0 (each bit indicates the same information as that for the input data).

(2 + 4*n)–byte data, described below, for all alarm states specified in the input data attribute when the input data attribute is other than 0.



(5) Completion code

0 : CNC alarm status has been read normally.

[Completion codes]

0 : This alarm status in the CNC has been read normally.

[Output data structure]

Top address + 0	(Function code) 23	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 2	
6	(Data number) —	
8	(Data attribute) —	Value
10	CNC alarm status data	2 byte bit data. For the meanings of the bits, see [Description] in this section.

B.4.12.3 Power Mate *i*–D/F/H

(1) Overview

PMC application programs can read CNC alarm information.

(2) Alarm information

1) Alarm status

Information concerning the alarm type

2) Detailed alarm

Information concerning the alarm number and axis information

(3) Input data configuration

Top address+0	+2	+4	+6	+8	+10
Function code	Completion code	Data length	Data number	Data attribute	Data area

Function code : 186

Completion code : No specification required.

Data length : No specification required.

Data number : No specification required. : For reading alarm status.

1 to 30 : When reading detailed alarm is specified, number of alarms which can be stored is set. (Up to 30).
If 31 or more are specified, the value is assumed to be 30.

Data area : No specification required.

Data attribute : 0 : Alarm status information

Other than 0 :
Detailed alarm information, indicated in two–byte of bit–type data described as “Alarm type data” below (multiple bits can be specified.)

[Alarm type data]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	bit

Bit 0 : P/S alarm 100 (PS1)
PWE, parameter write enable, is set to 1.

Bit 1 : P/S alarm 000 (PS2)
Turn off the power. Writing data into certain parameters may cause this alarm.

Bit 2 : P/S alarm 101 (PS3)
Part program storage has been disrupted. This alarm is issued when the CNC is turned off during tape editing or machining program reading. To release this alarm, press the RESET key while holding down the PROG key.

Bit 3 : A P/S alarm (PS) other than those described above has been issued. (Up to 255)

Bit 4 : Stroke limit alarm. (OTS)

Bit 5 : Overheat alarm. (OH)

Bit 6 : Servo alarm. (SV)

Bit 7 : Not used.

Bit 8 : APC alarm. (APAL)

Bit 9 : Spindle alarm. (SPA)

Bit 10 : P/S alarm 5000 or greater. (PS_2)

Bit 11 to Bit 15 : Not used.

(4) Output data configuration

Top address+0	+2	+4	+6	+8	+10
Function code	Completion code	Data length	Data number	Data attribute	Data area

Function code : 186

Completion code : Always 0.

Data length : When the data attribute is set to 0 : 2
(When reading alarm status information)

: When the data attribute is other than 0: 2+4*n
(When reading detailed alarm information)
(n: Number of alarm)

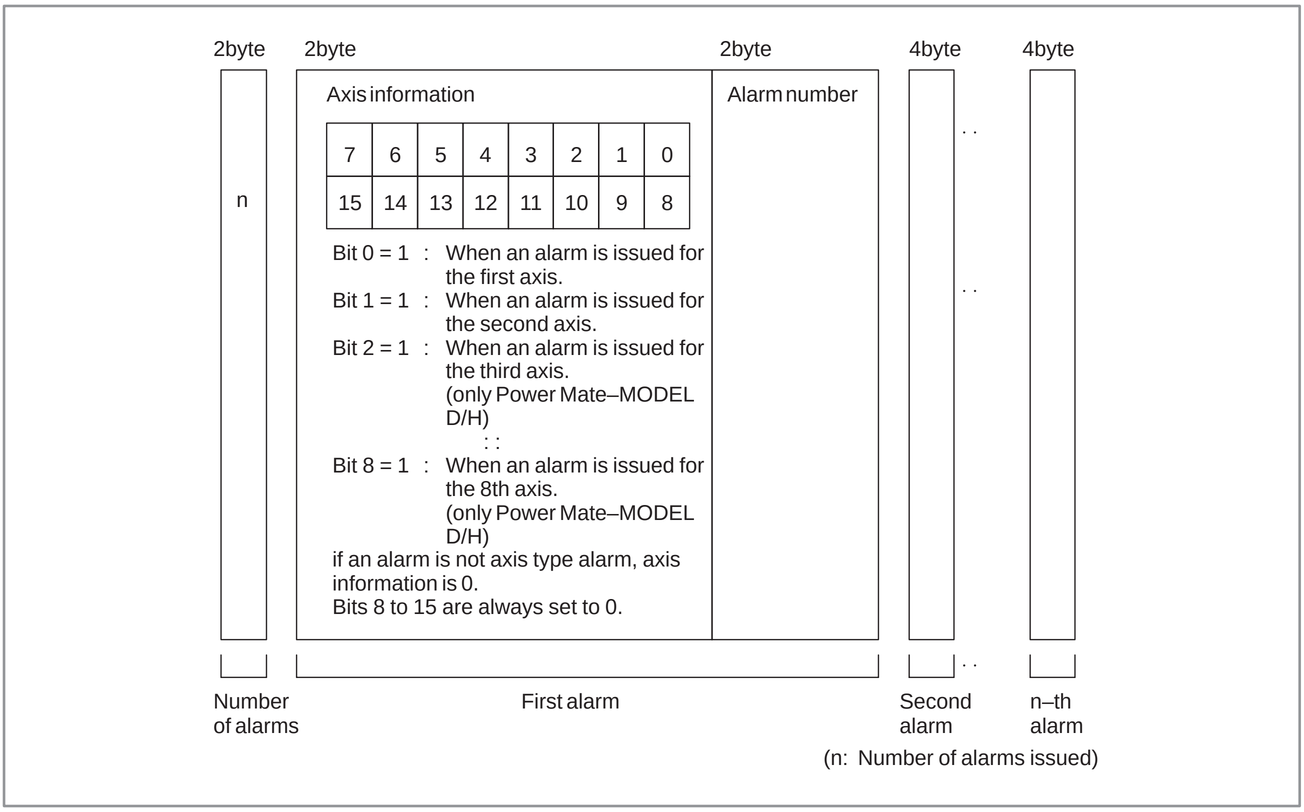
Data number : Same as that for the input data.

Data attribute : Same as that for the input data.

Data area : When the data attribute is set to 0 :
The alarm status information is output in the form of above–mentioned “Alarm type the data”.

: When the data attribute is other than 0:
The detailed alarm information corresponding to the alarm type data specified in the data attribute is output in the undermentioned data format.

[Structure of detailed alarm information]



(5) Completion code

0 : CNC alarm status has been read normally.

B.4.13 Reading the Current Program Number

[Description]

The program number of a machining program being executed or selected on the CNC can be read.
When a subprogram is executed on the CNC, the program number of the main program can also be read. Note that the program number that can be read is the first program number (first loop main program).
This function accepts only 4–digit program numbers. When the specification supports 8–digit program numbers, specify function code 90 to read 8–digit program numbers.

[Input data structure]

Top address + 0	(Function code) 24
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) 0
8	(Data attribute) 0
10	(Data area) — (Need not be set)
42	

[Completion codes]

0 : The program number of the currently executing program was read successfully.

6: The program number is an 8–digit program number. (Use function code 90.)

[Output data structure]

Top address + 0	(Function code) 24	
2	(Completion code) ? (See the explanation above.)	
4	(Data length) 4	
6	(Data number) —	
8	(Data attribute) —	Value
10	Current program number: ON	Unsigned binary, 2 bytes long
12	Program number of main program: OMN	

(a) Current program number (ON)

The program number of the program being executed is set.

(b) Program number of main program (OMN)

When the currently executing program is a subprogram, the program number of its main program (first loop main program) is set, When the currently executing program is not a subprogram, 0 is set.

B.4.14 Reading the Current Sequence Number

[Description]

The sequence number of a machining program being executed on the CNC can be read. If sequence numbers are not assigned to all blocks of the machining program, the sequence number of the most recently executed block is read.

[Input data structure]

Top address + 0	(Function code) 25
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) 0
8	(Data attribute) 0
10	(Data area) — (Need not be set)
42	

[Completion codes]

0 : The current sequence number has been read normally.

[Output data structure]

Offset	Field	Value
Top address + 0	(Function code) 25	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 4)	Note that the data length must be set to 4 bytes even though the current program number is 2 bytes long (the sequence number is indicated by 5 digits).
6	(Data number) —	
8	(Data attribute) —	Value
10	Current sequence number	Unsigned binary

B.4.15 Reading the Actual Velocity of Controlled Axes

[Description]

The actual velocity of a movement on CNC–controlled axes can be read. Note that the read speed is the composite velocity for the controlled axes. When movement involves only the basic three axes, the X, Y, and Z axes, the composite velocity equals the actual velocity. When movement, however, involves the fourth axis, such as a rotation axis or a parallel axis, as well as some of the basic three axes, the composite velocity for all the relevant axes does not equal the actual velocity.

[Input data structure]

Top address + 0	(Function code) 26
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) 0
8	(Data attribute) 0
10	(Data area) — (Need not be set) ~
42	

[Completion codes]

0 : The actual velocity for the controlled axes has been read normally.

[Output data structure]

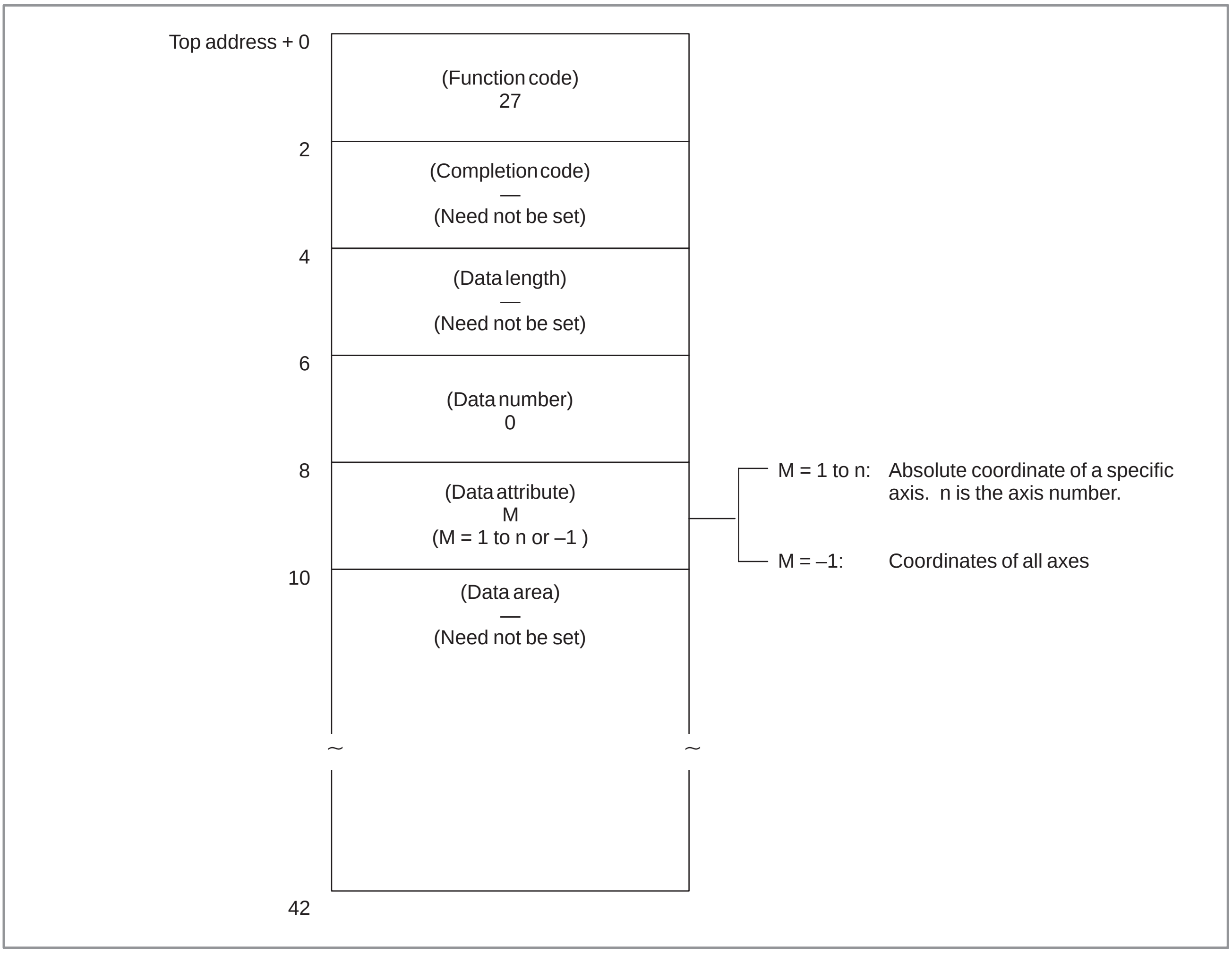
Top address + 0	(Function code) 26	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 4)	
6	(Data number) —	
8	(Data attribute) —	Value
10	Actual velocity for controlled axes	Unsigned binary <Data increments> •Input in mm 1 mm/min •Input in inches 0.01 inch/min.

B.4.16 Reading the Absolute Position (Absolute Coordinates) of Controlled Axes

[Description]

The absolute coordinates of the CNC–controlled axes for movement can be read. The read absolute coordinates equal the absolute coordinates (absolute position) indicated on the current position display screen in the CNC. (The screen is displayed by pressing function button POS.)

[Input data structure]



[Completion codes]

0 : The absolute coordinates of the controlled axes have been read normally.

4 : Data specified as the data attribute is invalid because it is neither –1 nor a value from 1 to n (n is the number of axes). Alternatively, the specified axis number is greater than the number of controlled axes.

[Output data structure]

Offset	Field	Value
Top address + 0	(Function code) 27	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 4*n, n is the number of axes specified.)	
6	(Data number) —	
8	(Data attribute) L (L: Input data)	
10	Absolute coordinate of the controlled axis specified (4 bytes)	Signed binary (A negative value is represented in 2's complement.)

When the number of controlled axes is 4

Offset	Field	Value
10	Absolute coordinate of the first axis (4 bytes)	Signed binary (A negative value is represented in 2's complement.)
14	Absolute coordinate of the second axis (4 bytes)	
18	Absolute coordinate of the third axis (4 bytes)	
22	Absolute coordinate of the fourth axis (4 bytes)	

Output data unit

<table>
<tr><th colspan="2"></th><th>Input system</th><th>Increment system IS–B</th><th>Increment system IS–C</th></tr>
<tr><td colspan="2" rowspan="2">Machining center system
Power Mate–D, F</td><td>mm, deg system</td><td>0.001</td><td>0.0001</td></tr>
<tr><td>inch system</td><td>0.0001</td><td>0.00001</td></tr>
<tr><td rowspan="4">Lathe system</td><td>Radius specification</td><td rowspan="2">mm, deg system</td><td>0.001</td><td>0.0001</td></tr>
<tr><td>Diameter specification</td><td>0.001</td><td>0.0001</td></tr>
<tr><td>Radius specification</td><td rowspan="2">inch system</td><td>0.0001</td><td>0.00001</td></tr>
<tr><td>Diameter specification</td><td>0.0001</td><td>0.00001</td></tr>
</table>

B.4.17 Reading the Machine Position (Machine Coordinates) of Controlled Axes

[Description]

The machine coordinates of CNC–controlled axes for movement can be read. The read value equals the machine coordinate indicated on the current position display screen displayed in the CNC. (This screen can be displayed by pressing the function button POS.)

[Input data structure]

Address	Content	
Top address + 0	(Function code) 28	
2	(Completion code) — (Need not be set)	
4	(Data length) — (Need not be set)	
6	(Data number) 0	
8	(Data attribute) M (M = 1 to n or –1)	M = 1 to n: Machine coordinate of a specific axis. n is the axis number. M = –1: Coordinates of all axes
10	(Data area) — (Need not be set) ~	
42		

NOTE

Values that are read follow the setting of parameter No. 1001 bit 1. So, the setting of parameter No. 3104 is sometimes disabled, and the read value may not match the value displayed on the CNC.

[Completion codes]

0 : The machine coordinates of the controlled axes have been read normally.

4 : Data specified as the data attribute is invalid because it is neither –1 nor a value from 1 to n (n is the number of axes). Alternatively, the specified axis number is greater than the number of the controlled axes.

[Output data structure]

Address	Content	Value
Top address + 0	(Function code) 28	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 4*n, n is the number of axes specified.)	
6	(Data number) —	
8	(Data attribute) M (M: Input data)	
10	Machine coordinate of the controlled axis specified (4 bytes)	Signed binary (A negative value is represented in 2's complement.)

When the number of controlled axes is 4

Address	Content	Value
10	Machine coordinate of the first axis (4 bytes)	Signed binary (A negative value is represented in 2's complement.)
14	Machine coordinate of the second axis (4 bytes)	
18	Machine coordinate of the third axis (4 bytes)	
22	Machine coordinate of the fourth axis (4 bytes)	

Output data unit

		Input system	Increment system IS–B	Increment system IS–C
Machining center system Power Mate–D, F		mm, deg system	0.001	0.0001
		inch system	0.0001	0.00001
Lathe system	Radius specification	mm, deg system	0.001	0.0001
	Diameter specification		0.001	0.0001
	Radius specification	inch system	0.0001	0.00001
	Diameter specification		0.0001	0.00001

B.4.18 Reading a Skip Position (Stop Position of Skip Operation (G31)) of Controlled Axes

[Description]

When a block of the skip operation (G31) is executed by the CNC and the skip signal goes on to stop the machine, the absolute coordinates of the stop position on the axes of movement can be read.

[Input data structure]

Address	Content
Top address + 0	(Function code) 29
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) 0
8	(Data attribute) M (M = 1 to n or –1)
10	(Data area) — (Need not be set) ~
42	

M = 1 to n: Skip coordinate on a specific axis. n is the axis number.

M = –1: Coordinates on all axes

[Completion codes]

0 : The coordinates of the skip stop position for the controlled axes have been read normally.

4 : Data specified for the data attribute is invalid because it is neither –1 nor a value from 1 to n (n is the number of axes). Alternatively, the specified axis number is greater than the number of controlled axes.

[Output data structure]

Address	Data	Value
Top address + 0	(Function code) 29	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 4*n, n is the number of axes specified.)	
6	(Data number) —	
8	(Data attribute) M (M: Input data)	
10	Skip coordinate of the controlled axis specified(4 bytes)	Signed binary (A negative value is represented in 2's complement.)

When the number of controlled axes is 4

Address	Data	Value
10	Skip coordinate of the second axis (4 bytes)	Signed binary (A negative value is represented in 2's complement.)
14	Skip coordinate of the third axis (4 bytes)	
18	Skip coordinate of the fourth axis (4 bytes)	
22	Skip coordinate of the first axis (4 bytes)	

Output data unit

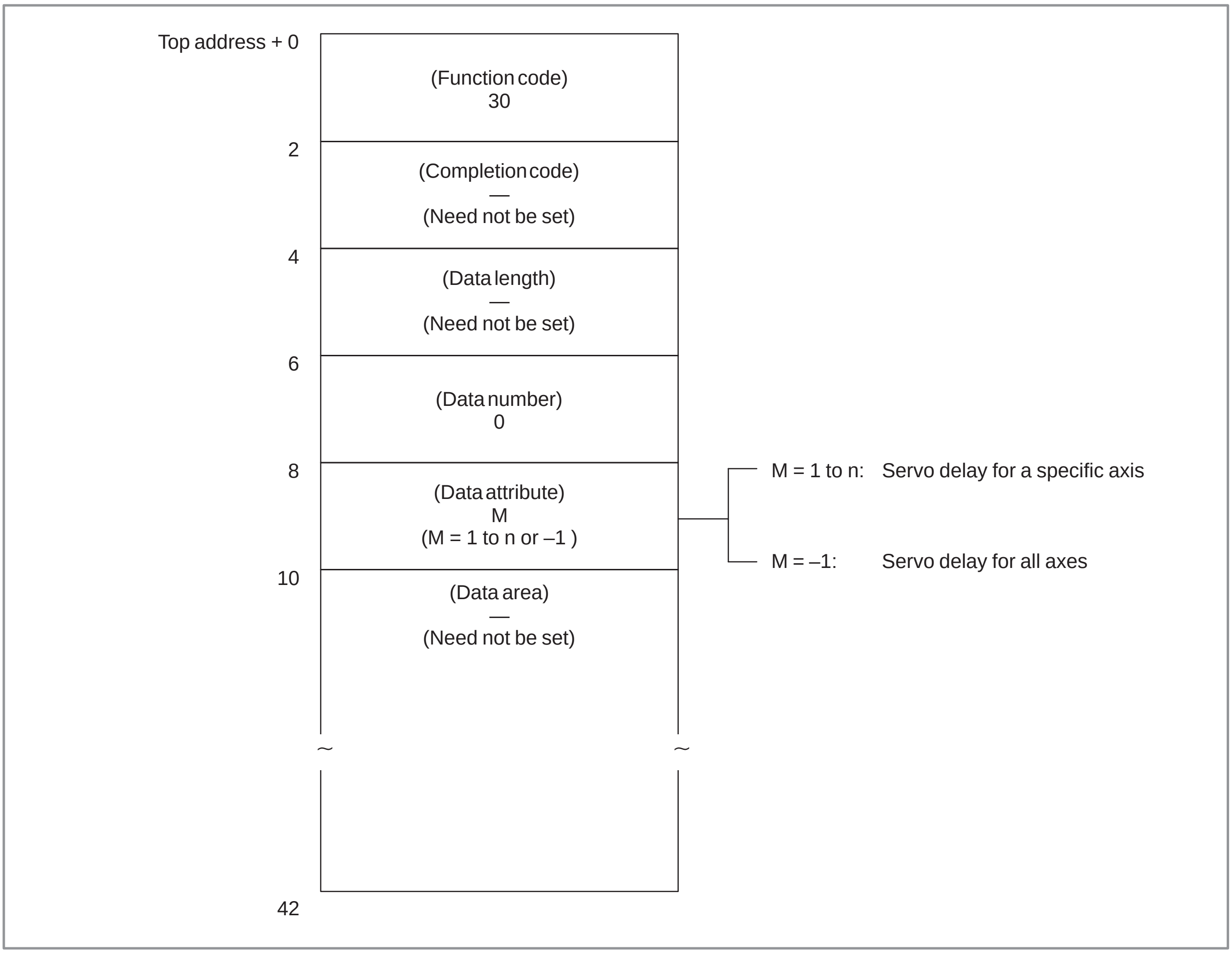
		Input system	Increment system IS–B	Increment system IS–C
Machining center system Power Mate–D, F		mm, deg system	0.001	0.0001
		inch system	0.0001	0.00001
Lathe system	Radius specification	mm, deg system	0.001	0.0001
	Diameter specification		0.001	0.0001
	Radius specification	inch system	0.0001	0.00001
	Diameter specification		0.0001	0.00001

B.4.19 Reading the Servo Delay for Controlled Axes

[Description]

The servo delay, which is the difference between the specified coordinates of CNC–controlled axes and the actual servo position, can be read.

[Input data structure]



[Completion codes]

0 : The servo delay for the controlled axes have been read normally.

4 : The data specified as the data attribute is invalid because it is neither –1 nor a value from 1 to n (n is the number of axes). Alternatively, the specified axis number is greater than the number of controlled axes.

[Output data structure]

Top address + 0	(Function code) 30	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 4*n, n is the number of axes specified.)	
6	(Data number) —	
8	(Data attribute) M (M: Input data)	Value
10	Servo delay for the controlled axis specified (4 bytes)	Signed binary (A negative value is represented in 2's complement.)

When the number of controlled axes is 4

		Value
10	Servo delay for the first axis (4 bytes)	Signed binary (A negative value is represented in 2's complement.)
14	Servo delay for the second axis (4 bytes)	
18	Servo delay for the third axis (4 bytes)	
22	Servo delay for the fourth axis (4 bytes)	

B.4.20 Reading the Acceleration/ Deceleration Delay on Controlled Axes

[Description]

The acceleration/deceleration delay, which is the difference between the coordinates of controlled axes programmed in the CNC and the position after acceleration/deceleration is performed, can be read.

[Input data structure]

Address	Content
Top address + 0	(Function code) 31
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) 0
8	(Data attribute) M (M = 1 to n or –1)
10	(Data area) — (Need not be set)
42	

- M = 1 to n: Acceleration/deceleration delay for a specific axis
- M = –1: Acceleration/deceleration delay for all axes

[Completion codes]

0 : The acceleration/deceleration delay for the control axis has been read normally.

4 : The data specified as the data attribute is invalid because it is neither –1 nor a value from 1 to n (n is the number of axes). Alternatively, the specified axis number is greater than the number of controlled axes.

[Output data structure]

Address	Item	Value
Top address + 0	(Function code) 31	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 4*n, n is the number of axes specified.)	
6	(Data number) —	
8	(Data attribute) M (M: Input data)	
10	Acceleration/deceleration delay for the controlled axis specified (4 bytes)	Signed binary (A negative value is represented in 2's complement.)

When the number of controlled axes is 4

Address	Item	Value
10	Acceleration/deceleration delay for the first axis (4 bytes)	Signed binary (A negative value is represented in 2's complement.)
14	Acceleration/deceleration delay for the second axis (4 bytes)	
18	Acceleration/deceleration delay for the third axis (4 bytes)	
22	Acceleration/deceleration delay for the fourth axis (4 bytes)	

Output data unit

		Input system	Increment system IS–B	Increment system IS–C
Machining center system Power Mate–D, F		mm, deg system	0.001	0.0001
		inch system	0.0001	0.00001
Lathe system	Radius specification	mm, deg system	0.001	0.0001
	Diameter specification		0.001	0.0001
	Radius specification	inch system	0.0001	0.00001
	Diameter specification		0.0001	0.00001

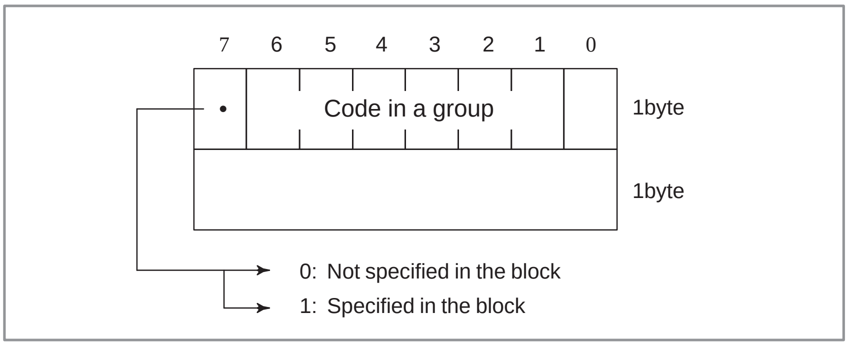
B.4.21 Reading Modal Data

[Description]

Modal information can be read from the CNC.

(1) Format and types of modal data for the G function

Data corresponding to the specified identification code is read and stored in the data area. Whether the data is specified in the block specified in the attribute of the data is determined by the value at the most significant bit.



NOTE
G codes for machining centers are also used for the Power Mate–D and –F except those marked with *. G codes marked with ** are not provided for the Power Mate–F.

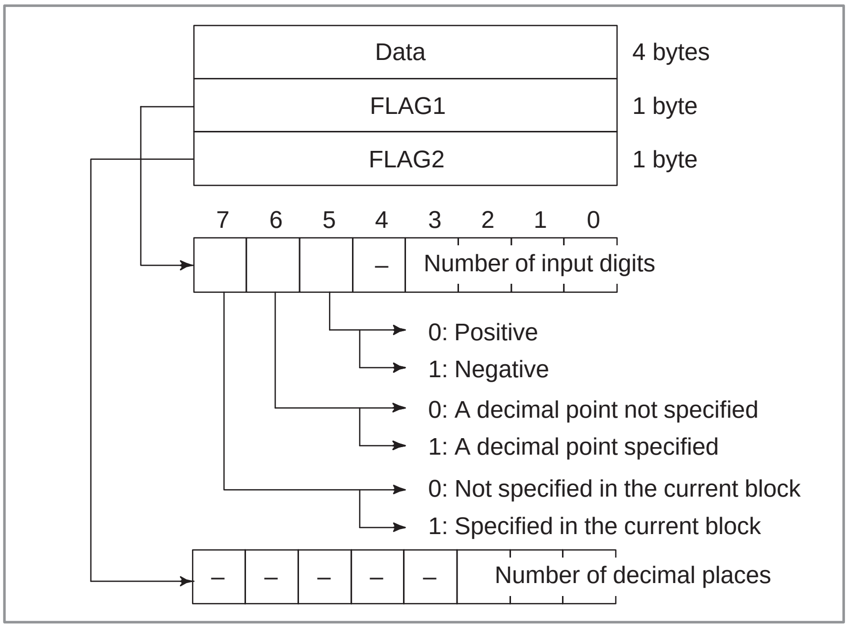
(1/2)

Identification code	Data type: G code for machining center (M)	Data: Code in a group	Data type: G code for lathe (T, G): A series	B series	C series	Data: Code in a group
0	G00 G01 **G02 **G03 *G33	0 1 2 3 4	G00 G01 G02 G03 G32 G33 G34 G90 G92 G94 G71 G72 G73 G74 (G71–G74: G series only)	G00 G01 G02 G03 G33 G34 G77 G78 G79 G71 G72 G73 G74 (G71–G74: G series only)	G00 G01 G02 G03 G33 G34 G20 G21 G24 G72 G73 G74 G75 (G72–G75: G series only)	0 1 2 3 4 8 9 5 6 7 10 11 12 13
1	G17 G18 G19	0 8 4	G96 G97	G96 G97	G96 G97	1 0
2	G90 G91	0 1		G90 G91	G90 G91	0 1
3			G68 G69	G68 G69	G68 G69	1 0
4	G94 G95	0 1	G98 G99	G94 G95	G94 G95	0 1
5	G20 G21	0 1	G20 G21	G20 G21	G70 G71	0 1

(2/2)

	Data type	Data	Data type			Data
Identification code	**G code for machining center (M)**	**Code in a group**	**G code for lathe (T, G)** **A series**	**B series**	**C series**	**Code in a group**
6	✳G40 ✳G41 ✳G42	0 1 2	G40 G41 G42	G40 G41 G42	G40 G41 G42	0 1 2
7	G43 G44 G49	1 2 0	G25 G26	G25 G26	G25 G26	0 1
8	G73 G74 G76 G80 G81 G82 G83 G84 G85 G86 G87 G88 G89	10 11 12 0 1 2 3 4 5 6 7 8 9	G22 G23	G22 G23	G22 G23	1 0
9	✳G98 ✳G99	0 1	G80 G83 G84 G85 G87 G88 G89	G80 G83 G84 G85 G87 G88 G89	G80 G83 G84 G85 G87 G88 G89	0 1 2 3 5 6 7
10	✳G50 ✳G51	0 1		G98 G99	G98 G99	0 1
11	G66 G67	1 0	G66 G67	G66 G67	G66 G67	1 0
13	✳G54 ✳G55 ✳G56 ✳G57 ✳G58 ✳G59	0 1 2 3 4 5	G54 G55 G56 G57 G58 G59	G54 G55 G56 G57 G58 G59	G54 G55 G56 G57 G58 G59	0 1 2 3 4 5
14	✳G61 ✳G62 ✳G63 ✳G64	1 2 3 0				
15	✳G68 ✳G69	1 0				
16	✳G15 ✳G16	0 1				
17	G40.1 G41.1 G42.1	1 2 0				
18	G25 G26	0 1				
19			G50.2 G51.2	G50.2 G51.2	G50.2 G51.2	0 1
20	G13.1 G12.1	0 1	G13.1 G12.1	G13.1 G12.1	G13.1 G12.1	0 1

(2) Format and types of modal data for other than the G function



The specification of whether a decimal point is specified or not, in FLAG1, and the specification of the number of decimal places, in FLAG2, are valid only for F code. Even if a decimal point is not specified, the number of decimal places may not be 0.

Data type		
Identification code	**Specified address**	
–2	Enter identification codes 100 to 126 at one time.	
100	B	(second auxiliary function)
101	D	
102	E	(reserved)
103	F	
104	H	
105	L	
106	M	
107	S	
108	T	
109	R	
110	P	
111	Q	
112	A	
113	C	
114	I	
115	J	
116	K	
117	N	
118	O	
119	U	
120	V	
121	W	
122	X	
123	Y	
124	Z	
125	M2	
126	M3	

NOTE
The Power Mate–D/F is not provided with the second auxiliary function.

[Input data structure]

Address	Contents
Top address + 0	(Function code) 32
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N: Data type)
8	(Data attribute) M (M: Specified block)
10	(Data area) — (Need not be set)
20	

N = 0 to : See the list of data explained above.
N = –1: All data for G function
N = –2: All data for other than G function

M = 0 : Current block
M = 1 : Next block
M = 2 : Block after the next block

When all data items are specified to be read, the data items are all output simultaneously in the order specified in the above data table.

[Completion codes]

0 : Modal information has been read normally.

3 : Invalid data is specified as the data number.

4 : Invalid data is specified as the data attribute.

[Output data structure]

Top address +	Data	Value
0	(Function code) 32 (See the explanation above	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 2, 6, 2*n, 6*m)	
6	(Data number) N (N: Input data)	
8	(Data attribute) M (M: Input data)	
10	Modal data for G function (2 bytes)	See the data format for the G function. The upper byte must always be set to 0.

L = 2 : G function
L = 2*n : All data for G function
L = 6 : Other than G function
L = 6*m : All data for other than G function
(n: Number of groups for the G function)
(m: Number of types other than for the G function)

Or

Top address +	Data	Value
8	(Data attribute) M (M: Input data)	
10	Data part of modal data for other than G function(4 bytes)	See the data format for other than the G function.
14	Flag part of modal data for other than G function(2 bytes)	See the flag format of the data for other than the G function. The upper byte must always be set to 0.

When all data items are specified to be read, the data items are all output simultaneously in the order specified in the above data table.

B.4.22 Reading Diagnosis Data (∗Low–speed Response)

[Description]

The information displayed on the diagnosis data screen in the CNC can be read.

[Input data structure]

Top address + 0	(Function code) 33
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N: Diagnosis No.)
8	(Data attribute) M (M: 0, 1 to n, or –1)
10	(Data area) — (Need not be set)

M = 0 : No axis
M = 1 to n: One axis
M = –1: All axes

[Completion codes]

0 : Diagnosis data has been read from the CNC normally.

3 : The specified diagnosis data number is invalid.

4 : The data specified as the data attribute is invalid because it is neither 0, –1, nor a value from 1 to n (n is the number of axes).

6 : An option required for reading the specified diagnosis data, such as the remote buffer option, is not provided.

[Output data structure]

Top address + 0	(Function code) 33	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 1, 2, 4, 1*n, 2*n, 4*n)	When no axis or one axis is specified L = 1 : Bit or byte parameter L = 2: Word parameter L = 4: Double Word parameter When all axes are specified L = 1*n Bit or byte parameter L = 2*n: Word parameter L = 4*n: Double word parameter
6	(Data number) N (N: Input data)	
8	(Data attribute) M (M: Input data)	Value
10	Diagnosis data	Data–dependentform

B.4.23 Reading A/D Conversion Data

[Description]

The load current for the CNC control axis can be converted to analog voltage, and input to the A/D converter in the CNC to obtain digital data. The output of the A/D converter can then be read.

[Input data structure]

Address	Content
Top address + 0	(Function code) 34
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N: Type of analog voltage)
8	(Data attribute) M (M – 1 to 8: Axis specification)
10	(Data area) — (Need not be set)

(a) Type of analog voltage (data number)

N	Type of analog voltage
0	General–purpose analog voltage information (for four channels)
2	Load information for the CNC–controlled axes

NOTE

Only one–path control of the Power Mate MODEL–D is provided with one channel of general–purpose analog voltage information.

(b) Specifying a CNC–controlled axis (data attribute)

Specify a CNC–controlled axis for which the voltage conversion data for the load current is to be read. Data must be specified according to the following table:

Specification of CNC controlled–axis	
Specified data	**Connector in the CNC**
1	JV1 (MAIN BOARD)
2	JV2 (MAIN BOARD)
3	JV3 (MAIN BOARD)
4	JV4 (MAIN BOARD)
5	JV5 (OPTION BOARD)
6	JV6 (OPTION BOARD)
7	JV7 (OPTION BOARD)
8	JV8 (OPTION BOARD)

[Completion codes]

0 : A/D conversion data has been read normally.

3 : The data specified for the data number is invalid.

4 : The data specified for the data attribute is invalid, or the specified axis number is greater than the number of controlled axes.

6 : No analog input module is connected.

[Output data structure]

		Value
Top address + 0	(Function code) 34	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 2	
6	(Data number) N (Input data)	
8	(Data attribute) M (Input data)	
10	A/D conversion data (2 bytes) AD For general purpose analog	Binary number from 0 to 255
10	A/D conversion data (2 bytes) AD For CNC controlled axis load information	Binary number from 0 to ±6554

(a) A/D conversion data (AD) of CNC controlled axis load information

The load current for the specified CNC controlled axis is converted into analog voltage, the input to the A/D converter to output a digital data.

The value actually set in the AD field is obtained from the following formula:

$$(AD) \times \frac{N}{6554} = \text{Load current } [A_{peak}]$$

AD = A/D conversion data [Value read by the window function (±)]
N = Nominal current limit for the amplifier corresponding to the motor
For the nominal current limits, see the table below or the descriptions of the control motor.

(b) A/D conversion data (A/D) for general–purpose analog voltage information

In A/D conversion data (A/D), 0 corresponds to –10V, 128 corresponds to 0V, 255 corresponds to +10V, and other values correspond in a direct proportion to these values.

$$(AD - 128) \times \frac{N}{128} = \text{Load current } [A_{peak}]$$

AD = A/D conversion data [Value read by the window function (±)]
N = Nominal current limit for the amplifier corresponding to the motor
For the nominal current limits, see the table below or the descriptions of the control motor.

Servo amplifier module		**Applicable motor model**	**Output current at rated output**	**Nominal current limit**
Model	**Connected axis**			
SVM1–12 SVM2–12/12 SVM2–12/20 SVM2–12/40 SVM3–12/12/12 SVM3–12/12/20 SVM3–12/20/20 SVM3–12/12/40 SVM3–12/20/40	 L and M axes L axis L axis L, M, and N axes L and M axes L axis L and M axes L axis	α 0.5/3000 α 1/3000 α 2/2000 α 2/3000	$2.9A_{rms}$	12Ap
SVM1–20 SVM2–12/20 SVM2–20/20 SVM2–20/40 SVM3–12/12/20 SVM3–12/20/20 SVM3–20/20/20 SVM3–12/20/40 SVM3–20/20/40	 M axis L and M axes L axis N axis M and N axes L, M, and N axes M axis L and M axes	α C3/2000 α C6/2000 α C12/2000	$5.8A_{rms}$	20Ap
SVM1–40S		α 3/3000 α 6/2000 α M3/3000 α L3/3000	$5.8A_{rms}$	40Ap
SVM2–12/40 SVM2–20/40 SVM2–40/40 SVM3–12/12/40 SVM3–12/20/40 SVM3–20/20/40	M axis M axis M axis N axis N axis N axis	α 3/3000 α 6/2000 α 12/2000 α M3/3000 α L3/3000 α C22/1500	$12.2A_{rms}$	40Ap

Servo amplifier module		Applicable motor model	Output current at rated output	Nominal current limit
Model	**Connected axis**			
SVM1–40L SVM2–40/80	 L axis	α 3/3000 α 6/2000 α 12/2000 α 22/1500 α M3/3000 α L3/3000 α C22/1500	$12.2A_{rms}$	40Ap
SVM1–80 SVM2–40/80 SVM2–80/80	 L axis L and M axes	α 6/3000 α 12/3000 α 22/2000 α 30/1200 α M6/3000 α M9/3000 α L6/3000 α L9/3000	$18.4A_{rms}$	80Ap
SVM1–130		α 30/2000 α 40/2000	$26.7A_{rms}$	130Ap
		α 22/3000 α 30/3000 α 40/2000 (with a fan) α L25/3000 α L50/2000	$51.0A_{rms}$ (Note3)	

NOTE

1 The rated output is guaranteed at the rated input voltage. If an input voltage fluctuation occurs even within the allowable fluctuation range, the rated output may not be obtained.
2 The current limits (peak values) are standard values. The operation value variation due to a circuit constant is about +10%.
3 SVM1–130 requires forced air cooling when the α 22/3000, α 30/3000, α 40/2000 (with a fan), α L25/3000, or α L50/2000 is driven. At this time, the rated output current is 51.0 Arms.

B.4.24 Reading Tool Life Management Data (Tool Group No.) (not available for Power Mate–D/F, Series 21–TA)

[Description]

By specifying a tool No., the No. of the tool group to which the specified tool belongs can be read from tool life management data.

[Input data structure]

Address	Content
Top address + 0	(Function code) 38
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) 0
8	(Data attribute) M (M : Tool No.)
10	(Data area) — (Need not be set)
42	

NOTE

If 0 is specified for the tool No., the No. of the tool group currently used is read. In this case, if a tool group No. has not been specified since the power to the CNC was turned on, 0 is output.
If the same tool belongs to two or more tool groups, the Nos. of all tool groups to which the tool belongs are displayed.

[Completion codes]

0 : The tool group No. has been read normally.

4 : The value specified for the data attribute is invalid.

5 : The specified tool No. was not found.

6 : The tool life management option has not been added.

[Output data structure]

Top address + 0	(Function code) 38	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 4 × n)	L = 4 to 4 × n n is the number of tool groups to which the specified tool belongs.
6	(Data number) —	
8	(Data attribute) M (M: Input data)	Value
10	Tool group No. (4 bytes)	Unsigned binary

When the specified tool belongs to two or more tool groups

10	Tool group No. (4 bytes)	Unsigned binary
14	Tool group No. (4 bytes)	
18	Tool group No. (4 bytes)	

B.4.25 Reading Tool Life Management Data (Number of Tool Groups) (not available for Power Mate–D/F, Series 21–TA)

[Description]

The number of tool groups in tool life management data can be read.

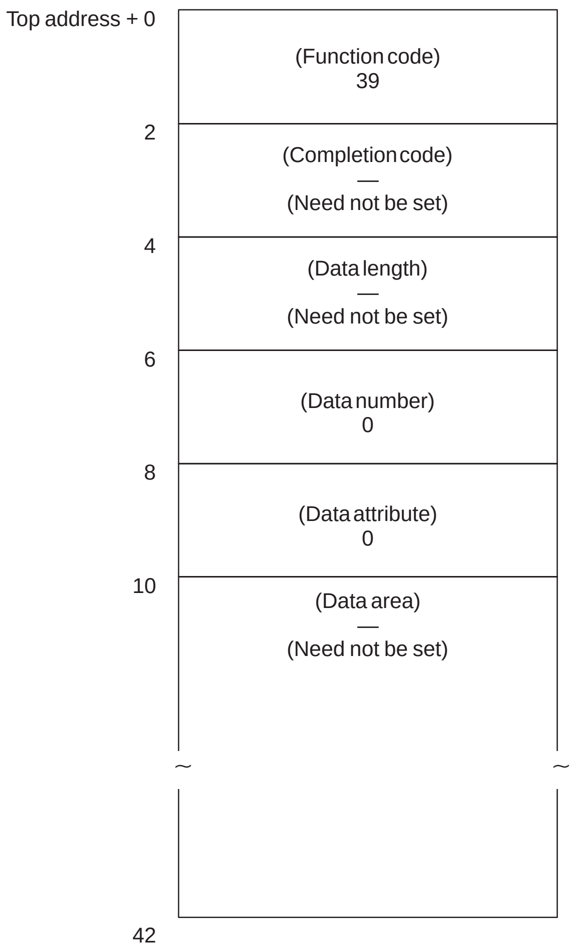
The number of tool groups that can be registered varies depending on the setting of parameter 6800 of the CNC, as indicated in the following table.

Parameter 6800

GS2	GS1	Number of tools The numbers in parentheses apply when the additional option is used	
		M series	T series
0	0	1 to 16 (1 to 64)	1 to 16 (1 to 16)
0	1	1 to 32 (1 to 128)	1 to 32 (1 to 32)
1	0	1 to 64 (1 to 256)	1 to 64 (1 to 64)
1	1	1 to 128 (1 to 512)	1 to 16 (1 to 128)

M series: For Machining Centers T series: For Lathes

[Input data structure]



[Completion codes]

0 : The number of tool group Nos. has been read normally.

6 : The tool life management option has not been added.

[Output data structure]

Address	Contents	Value
Top address + 0	(Function code) 39	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 4	
6	(Data number) —	
8	(Data attribute) —	
10	Number of tool groups (4 bytes)	Unsigned binary

B.4.26 Reading Tool Life Management Data (Number of Tools) (not available for Power Mate–D/F, Series 21–TA)

[Description]

By specifying a tool group No., the number of tools that belong to the tool group can be read from tool life management data.

The number of tools that can be registered varies depending on the setting of parameter 6800 of the CNC, as indicated in the following table.

Parameter 6800

GS2	GS1	Number of tools The numbers in parentheses apply when the additional option is used	
		M series	T series
0	0	1 to 16 (1 to 64)	1 to 16 (1 to 16)
0	1	1 to 32 (1 to 128)	1 to 32 (1 to 32)
1	0	1 to 64 (1 to 256)	1 to 64 (1 to 64)
1	1	1 to 128 (1 to 512)	1 to 16 (1 to 128)

M series: For Machining Centers T series: For Lathes

[Input data structure]

Address	Contents
Top address + 0	(Function code) 40
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N: Tool group No.)
8	(Data attribute) 0
10	(Data area) — (Need not be set)
42	

NOTE
If 0 is specified for the tool group No., the number of tools that belong to the tool group currently used is read. In this case, if a tool group No. has not been specified since the power to the CNC was turned on, 0 is output.

[Completion codes]

0 : The number of tools has been read normally.

3 : The specified tool group No. is invalid.

6 : The tool life management option has not been added.

[Output data structure]

Address	Contents	Value
Top address + 0	(Function code) 40	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 4	
6	(Data number) N (N: Input data)	
8	(Data attribute) —	
10	Number of tools (4 bytes)	Unsigned binary

B.4.27 Reading Tool Life Management Data (Tool Life) (not available for Power Mate–D/F, Series 21–TA)

[Description]

By specifying a tool group No., the life of tools belonging to the tool group can be read from tool life management data.

Whether to display the tool life in minutes or the number of cycles is selected by bit 2 of parameter 6800 (LTM) for the CNC.

[Input data structure]

Address	Contents
Top address + 0	(Function code) 41
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N: Tool group No.)
8	(Data attribute) 0
10	(Data area) — (Need not be set)
~	~
42	

NOTE

If 0 is specified for the tool group No., the tool life of the tool group currently used is read. In this case, if a tool group No. has not been specified since the power to the CNC was turned on, 0 is output.

[Completion codes]

0 : The tool life has been read normally.

3 : The specified tool group No. is invalid.

6 : The tool life management option has not been added.

[Output data structure]

Top address + 0	(Function code) 41	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 4	
6	(Data number) N (N: Input data)	
8	(Data attribute) —	Value
10	Tool life (4 bytes)	Unsigned binary Unit: Time (minutes) or number of cycles

B.4.28 Reading Tool Life Management Data (Tool Life Counter) (not available for Power Mate–D/F, Series 21–TA)

[Description]

By specifying a tool group No., the tool life counter for the specified tool group can be read from tool life management data.

[Input data structure]

Address	Contents
Top address + 0	(Function code) 42
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N: Tool group No.)
8	(Data attribute) 0
10	(Data area) — (Need not be set) ~ ~
42	

NOTE

If 0 is specified for the tool group No., the tool life counter for the tool group currently used is read. In this case, if a tool group No. has not been specified since the power to the CNC was turned on, 0 is output.

[Completion codes]

0 : The tool life has been read normally.

3 : The specified tool group No. is invalid.

6 : The tool life management option has not been added.

[Output data structure]

Address	Field	Value
Top address + 0	(Function code) 42	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 4	
6	(Data number) N (N: Input data)	
8	(Data attribute) —	
10	Tool life counter (4 bytes)	Unsigned binary Unit: Time (minutes) or number of cycles

B.4.29 Reading Tool Life Management Data (Tool Length Compensation No. (1): Tool No.) (not available for Power Mate–D/F, Series 21–TA)

[Description]

By specifying a tool group No. and a tool No., the tool length compensation No. for the specified tool can be read from tool life management data. This function is available only with the M series CNCs.

[Input data structure]

Top address + 0	(Function code) 43
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N: Tool group No.)
8	(Data attribute) M (M: Tool No.)
10 ~ 42	(Data area) — (Need not be set)

NOTE

If 0 is specified for both the tool group No. and tool No., the Nos. of the tool group and tool currently used are read. In this case, if a tool group No. has not been specified since the power to the CNC was turned on, 0 is output.
For the T series CNCs, 0 is always output.

[Completion codes]

0 : The tool length compensation No. has been read normally.

3 : The specified tool group No. is invalid.

4 : The specified tool No. is invalid.

5 : The specified tool No. was not found in the specified tool group.

6 : The tool life management option has not been added.

[Output data structure]

Top address + 0	(Function code) 43	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 4	
6	(Data number) N (N: Input data)	
8	(Data attribute) M (M: Input data)	Value
10	Tool length compensation No. (4 bytes)	Unsigned binary

B.4.30 Reading Tool Life Management Data (Tool Length Compensation No. (2): Tool Order No.) (not available for Power Mate–D/F, Series 21–TA)

[Description]

By specifying a tool group No. and tool order No., the tool length compensation No. for the specified tool can be read from tool life management data. This function is available only with the M series CNCs.

[Input data structure]

Address	Content
Top address + 0	(Function code) 44
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N: Tool group No.)
8	(Data attribute) M (M: Tool order No.)
10	(Data area) — (Need not be set)
42	

NOTE

If 0 is specified for the tool group No., the No. of the tool group currently used is read. In this case, if a tool group No. has not been specified since the power to the CNC was turned on, 0 is output.

When 0 is specified for the tool order No., if the specified tool group has been used, the tool currently used is read. In this case, if the specified tool group has not been used, the first tool in the group is read.

For the T series CNCs, 0 is always output.

[Completion codes]

0 : The tool length compensation No. has been read normally.

3 : The specified tool group No. is invalid.

4 : The specified tool order is invalid.

5: The tool having the specified tool order is not registered in the specified tool group.

6 : The tool life management option has not been added.

[Output data structure]

Top address + 0	(Function code) 44	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 4	
6	(Data number) N (N: Input data)	
8	(Data attribute) M (M: Input data)	Value
10	Tool length compensation No. (4 bytes)	Unsigned binary

B.4.31 Reading Tool Life Management Data (Cutter Compensation No. (1): Tool No.) (not available for Power Mate–D/F, Series 21–TA)

[Description]

By specifying a tool group No. and a tool No., the cutter compensation No. for the specified tool can be read from tool life management data. This function is available only with the M series CNCs.

[Input data structure]

Address	Contents
Top address + 0	(Function code) 45
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N: Tool group No.)
8	(Data attribute) M (M : Tool No.)
10	(Data area) — (Need not be set) ~
42	

NOTE

If 0 is specified for both tool group No. and tool No., the Nos. of the tool group and tool currently used are read. If a tool group No. has not been specified since the power to the CNC was turned on, 0 is output.
For the T series CNCs, 0 is always read.

[Completion codes]

0 : The cutter compensation No. has been read normally.

3 : The specified tool group No. is invalid.

4 : The specified tool No. is invalid.

5 : The specified tool No. was not found in the specified tool group.

6 : The tool life management option has not been added.

[Output data structure]

Top address + 0	(Function code) 45	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 4	
6	(Data number) N (N: Input data)	
8	(Data attribute) M (M: Input data)	Value
10	Cutter compensation No. (4 bytes)	Unsigned binary

B.4.32 Reading Tool Life Management Data (Cutter Compensation No. (2): Tool Order No.) (not available for Power Mate–D/F, Series 21–TA)

[Description]

By specifying a tool group No. and a tool order No., the cutter compensation No. for the specified tool can be read from tool life management data. This function is available only with the M series CNCs.

[Input data structure]

Top address + 0	(Function code) 46
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N: Tool group No.)
8	(Data attribute) M (M: Tool order No.)
10	(Data area) — (Need not be set) ~
42	

NOTE

If 0 is specified for the tool group No., the No. of the tool group currently used is referenced. In this case, if a tool group No. has not been specified since the power to the CNC was turned on, 0 is output.

When 0 is specified for the tool order No., if the specified tool group has been used, the tool currently used is read. In this case, if the specified tool group has not been used, the first tool in the group is referred to.

For the T series CNCs, 0 is always output.

[Completion codes]

0 : The cutter compensation No. has been read normally.

3 : The specified tool group No. is invalid.

4 : The specified tool order No. is invalid.

5: The tool having the specified tool order is not registered in the specified tool group.

6 : The tool life management option has not been added.

[Output data structure]

Top address + 0	(Function code) 46	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 4	
6	(Data number) N (N: Input data)	
8	(Data attribute) M (M: Input data)	Value
10	Cutter compensation No. (4 bytes)	Unsigned binary

B.4.33 Reading Tool Life Management Data (Tool Information (1) : Tool No.) (not available for Power Mate–D/F, Series 21–TA)

[Description]

By specifying a tool group No. and a tool No., the information for the specified tool can be read from tool life management data.

[Input data structure]

Address	Content
Top address + 0	(Function code) 47
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N: Tool group No.)
8	(Data attribute) M (M: Tool No.)
10	(Data area) — (Need not be set)
42	

NOTE

If 0 is specified for both tool group No. and tool No., the Nos. of the tool group and tool currently used are referenced. If neither a tool group No. nor a tool No. has been specified since the power to the CNC was turned on, 0 is output.

[Completion codes]

0 : The tool group No. has been read normally.

3 : The specified tool group No. is invalid.

4 : The specified tool No. is invalid.

5 : The specified tool No. was not found in the specified tool group.

6 : The tool life management option has not been added.

[Output data structure]

Address	Field	Value
Top address + 0	(Function code) 47	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 4	
6	(Data number) N (N: Input data)	
8	(Data attribute) M (M: Input data)	Value
10	Number of tools (4 bytes)	0: See Note) on the previous page. 1: The tool is registered. 2: The tool has reached the end of its life. 3: The tool was skipped. The three high–order bytes are fixed to 0.

B.4.34 Reading Tool Life Management Data (Tool Information (2): Tool Order No.) (not available for Power Mate–D/F, Series 21–TA)

[Description]

By specifying a tool group No. and a tool order No., the information for the specified tool can be read from tool life management data.

[Input data structure]

Top address + 0	(Function code) 48
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N: Tool group No.)
8	(Data attribute) M (M: Tool order No.)
10	(Data area) — (Need not be set) ~ ~
42	

NOTE

If 0 is specified for the tool group No., the No. of the tool group currently used is read. If a tool group No. has not been specified since the power to the CNC was turned on, 0 is output.

When 0 is specified for the tool order No., if the specified tool group has ever been used, the tool currently used is read. In this case, if the specified tool group has not been used, the first tool in the group is referred to.

[Completion codes]

0 : The tool group No. has been read normally.

3 : The specified tool group No. is invalid.

4 : The specified tool order No. is invalid.

5: The tool having the specified tool order is not registered in the specified tool group.

6 : The tool life management option has not been added.

[Output data structure]

Offset	Field	Value
Top address + 0	(Function code) 48	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 4	
6	(Data number) N (N: Input data)	
8	(Data attribute) M (M: Input data)	
10	Tool information (4 bytes)	0: See Note) on the previous page. 1: The tool is registered. 2: The tool has reached the end of its life. 3: The tool was skipped. The three high–order bytes are fixed to 0.

B.4.35 Reading Tool Life Management Data (Tool No.) (not available for Power Mate–D/F, Series 21–TA)

[Description]

By specifying a tool group No. and a tool order No., the No. of the corresponding tool can be read from tool life management data.

[Input data structure]

Top address + 0	(Function code) 49
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (N: Tool group No.)
8	(Data attribute) M (M: Tool order No.)
10	(Data area) — (Need not be set)
42	

NOTE

When 0 is specified for the tool group No., the tool group currently used is referenced. If neither a tool group No. nor a tool No. has been specified since the power to the CNC was turned on, however, 0 is output for the tool group No. When 0 is specified for the tool order No., if the specified tool group has been used, the tool currently used is referred to. If the specified tool group has not been used, the first tool in the group is referenced.

[Completion codes]

0 : The tool No. has been read normally.

3 : The specified tool group No. is invalid.

4 : The specified tool order No. is invalid.

6 : The tool life management option has not been added.

[Output data structure]

		Value
Top address + 0	(Function code) 49	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 4	
6	(Data number) N (N: Input data)	
8	(Data attribute) M (M: Input data)	
10	Tool No. (4 bytes)	Unsigned binary

B.4.36 Reading the Actual Spindle Speed

[Description]

The actual speed of the spindle can be read from the CNC.

[Input data structure]

Address	Content
Top address + 0	(Function code) 50
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) 0
8	(Data attribute) 0
10	(Data area) — (Need not be set) ~
42	

[Completion codes]

0 : The actual speed of the spindle has been read normally.

[Output data structure]

Top address + 0	(Function code) 50	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 4	
6	(Data number) —	
8	(Data attribute) —	Value
10	Actual spindle speed	Unsigned binary <Data unit> rpm

B.4.37 Entering Data on the Program Check Screen (✻Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

On the program check screen of the CNC, data can be entered for the spindle tool No. and the next tool No. This function is available only with the M series CNCs.

This function is effective only when bit 2 of parameter 3108 is 1.

[Input data structure]

Address	Contents	
Top address + 0	(Function code) 150	
2	(Completion code) — (Need not be set)	
4	(Data length) 4	
6	(Data number) N (N = 0, 1)	N = 0 : Spindle tool No. (8 digits) N = 1 : Next tool No. (8 digits)
8	(Data attribute) 0	Value
10	Data for the spindle tool No. (4 bytes) or data for the next tool No. (4 bytes)	Unsigned binary

[Completion codes]

0 : Data has been entered on the program check screen normally.

2 : The data length in bytes is invalid.

3 : The data No. is invalid.

[Output data structure]

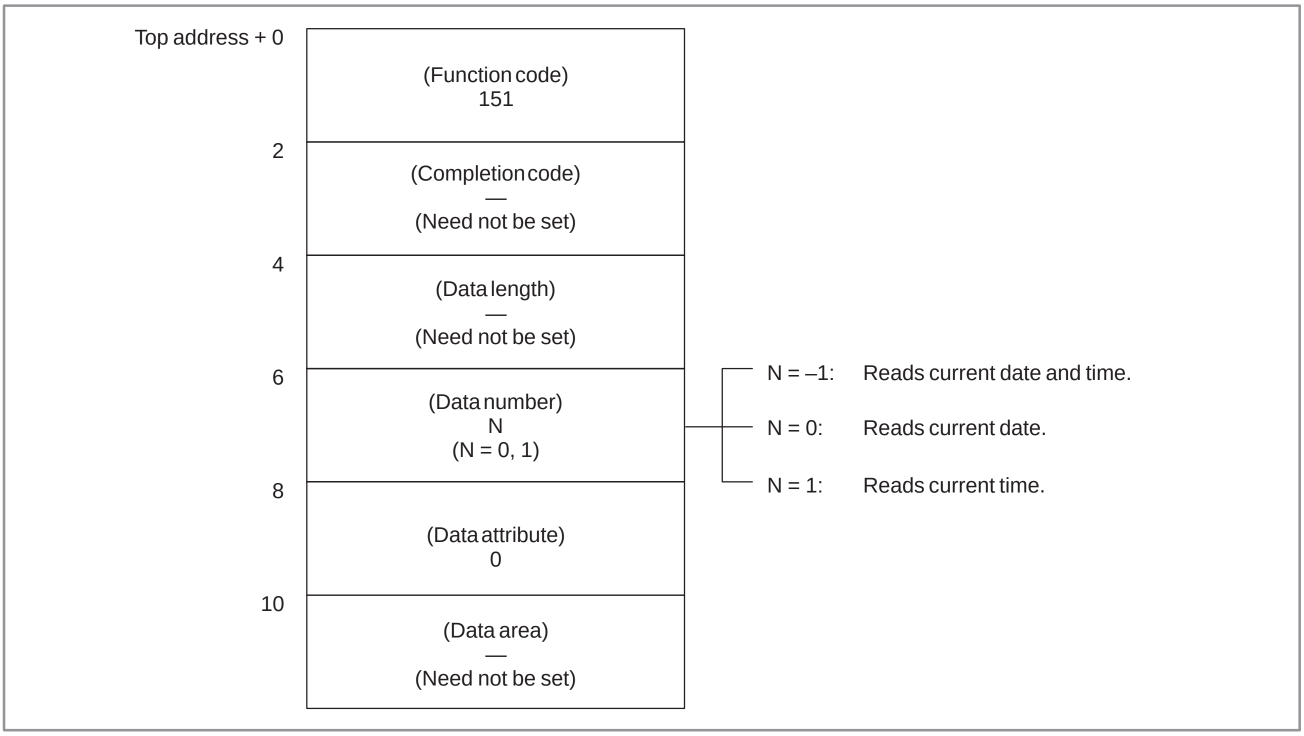
Top address + 0	(Function code) 150	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 4 (Input data)	
6	(Data number) N (Input data)	N = 0 : Spindle tool No. (8 digits) N = 1 : Next tool No. (8 digits)
8	(Data attribute) — (Input data)	Value
10	Data for the spindle tool No. (4 bytes) or data for the next tool No. (4 bytes)	Unsigned binary

B.4.38 Reading Clock Data (Date and Time) (not available for Power Mate–F)

[Description]

The current date (year, month, day) and time (hours, minutes, seconds) can be read from the clock built into the CNC.

[Input data structure]



[Completion codes]

0 : Data of the clock built into the CNC has been read normally.

3 : A value other than 0, 1, and –1 was specified for the data No.

[Output data structure]

Address	Item	Value
Top address + 0	(Function code) 151	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 6/12	
6	(Data number) N (Input data)	
8	(Data attribute) — (Input data)	
		Value
10	Current date (year) or time (hours)	Unsigned binary
12	Current date (month) or time (minutes)	
14	Current date (day) or time (seconds)	

When both the current date and current time are specified to be read by entering [–1] for the data No.

Address	Item	Value
	— (Input data)	Value
10	Current date (year)	Unsigned binary
12	Current date (month)	
14	Current date (day)	
16	Current time (hours)	
18	Current time (minutes)	
20	Current time (seconds)	

[Example] September 10th, 1990

Data area	
+0	1990
+2	9
+4	10

[Example] 23:59:59 (hours:minutes:seconds)

Data area	
+0	23
+2	59
+4	59

B.4.39 Entering Torque Limit Data for the Digital Servo Motor (∗Low–speed Response)

[Description]

Torque limit values for the digital servo motor can be entered.

[Input data structure]

Address	Field	Value
Top address + 0	(Function code) 152	
2	(Completion code) — (Need not be set)	
4	(Data length) 2	
6	(Data number) 0	
8	(Data attribute) M (M: 1 to n)	M = 1 to n: Axis No.
10	Torque limit data (1 byte) The high–order byte is always set to 0.	Unsigned binary <Unit: %> Values from 0 to 255 correspond to 0% to 100%.

[Example] To specify a torque limit of 50%, enter 128.

[Completion codes]

0 : Torque limit data has been entered normally.

4 : The specified data attribute is invalid. That is, a value other than 1 to n (number of axes) was specified, or the specified axis No. was greater than the number of controlled axes.

[Output data structure]

Top address + 0	(Function code) 152	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 2 (Input data)	
6	(Data number) — (Input data)	
8	(Data attribute) M (M: Input data)	Value
10	Torque limit data (1 byte): Input data The high–order byte is always set to 0.	Unsigned binary <Unit: %> Values from 0 to 255 correspond to 0% to 100%.

B.4.40 Reading Load Information of the Spindle Motor (Serial Interface)

[Description]

Load information of the serial spindle can be read.

The equation to normalize the load information is shown below

$$\text{Load (\%)} = \frac{L}{32767} \times \lambda$$

L: Data read from the window

λ: The percentage of the maximum output of the motor to the continuous rated output of the motor (When the maximum output is 180% and the continuous rated output is 100%, the percentage is 180.)

NOTE
λ is equal to the value of parameter No. 4127.

[Input data structure]

Address	Content
Top address + 0	(Function code) 153
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) N (Need not be set)
8	(Data attribute) 0
10	(Data area) — (Need not be set)
42	

N= 0: Read the load of the No.1 spindle.
1: Read the load of the No.2 spindle.
2: Read the load of the No.3 spindle.
3: Read the load of the No.4 spindle.
–1: Read the loads of the No.1 and No.2 spindles.
–2: Read the loads of the No.1 to No.3 spindles.
–3: Read the loads of the No.1 to No.4 spindles.

[Completion codes]

0 : Load information of the serial spindle has been read normally.

[Output data structure]

Top address + 0	(Function code) 153	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 2, 4)	L = 2: Specifies the first axis. L = 4: Specifies all axes.
6	(Data number) N (N: Input data)	N= 0: Read the load of the No.1 spindle. 1: Read the load of the No.2 spindle. 2: Read the load of the No.3 spindle. 3: Read the load of the No.4 spindle. –1: Read the loads of the No.1 and No.2 spindles. –2: Read the loads of the No.1 to No.3 spindles. –3: Read the loads of the No.1 to No.4 spindles.
8	(Data attribute) —	Value
10	Load information of the serial spindle (first or second axis)	Unsigned binary, 2 bytes long

When two axes are specified

8	(Data attribute) —	Value
10	First axis in the load information of the serial spindle	Unsigned binary, 2 bytes long
12	Second axis in the load information of the serial spindle	

When three axes are specified

Offset	(Data attribute)	Value
8	—	
10	First axis in the load information of the serial spindle	Unsigned binary, 2 bytes long
12	Second axis in the load information of the serial spindle	
14	Third axis in the load information of the serial spindle	

When four axes are specified

Offset	(Data attribute)	Value
8	—	
10	First axis in the load information of the serial spindle	Unsigned binary, 2 bytes long
12	Second axis in the load information of the serial spindle	
14	Third axis in the load information of the serial spindle	
16	Fourth axis in the load information of the serial spindle	

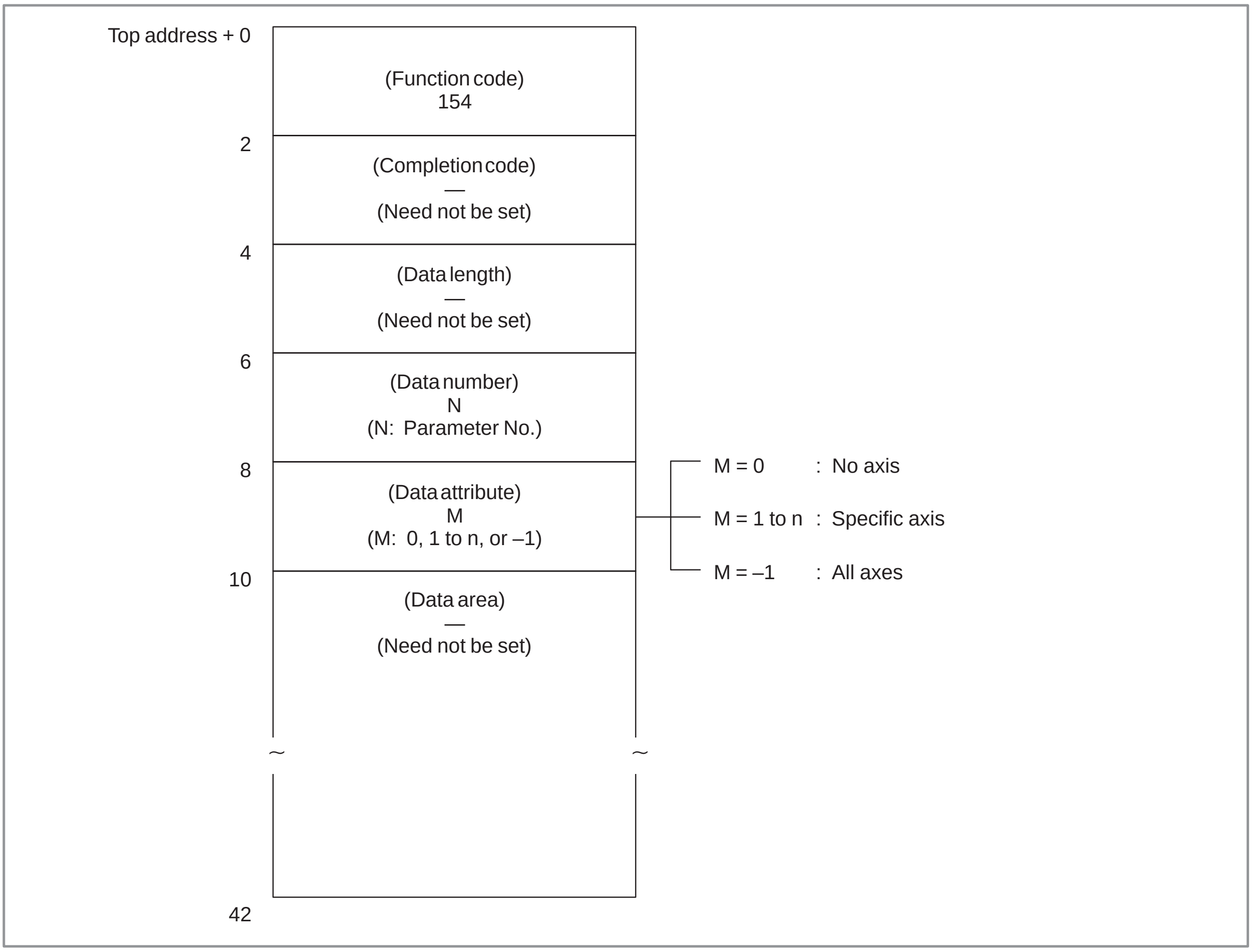
B.4.41 Reading a Parameter (not available for Power Mate–D/F, Series 21–TA)

[Description]

Parameter data in the CNC can be read directly from the CNC via the FANUC bus.

This function is basically the same as the function described in Section 3.6 ”Reading a Parameter,” except that the function code is 154 and some of the completion codes are different.

[Input data structure]



[Completion codes]

0 : Parameter data has been read normally.

3 : The parameter No. specified to be read is invalid.

4 : A value other than 0, –1, and 1 to n (number of axes) was specified for the data attribute.

6 : An option required for setting the parameter to be read, such as the error compensation option, is not provided.

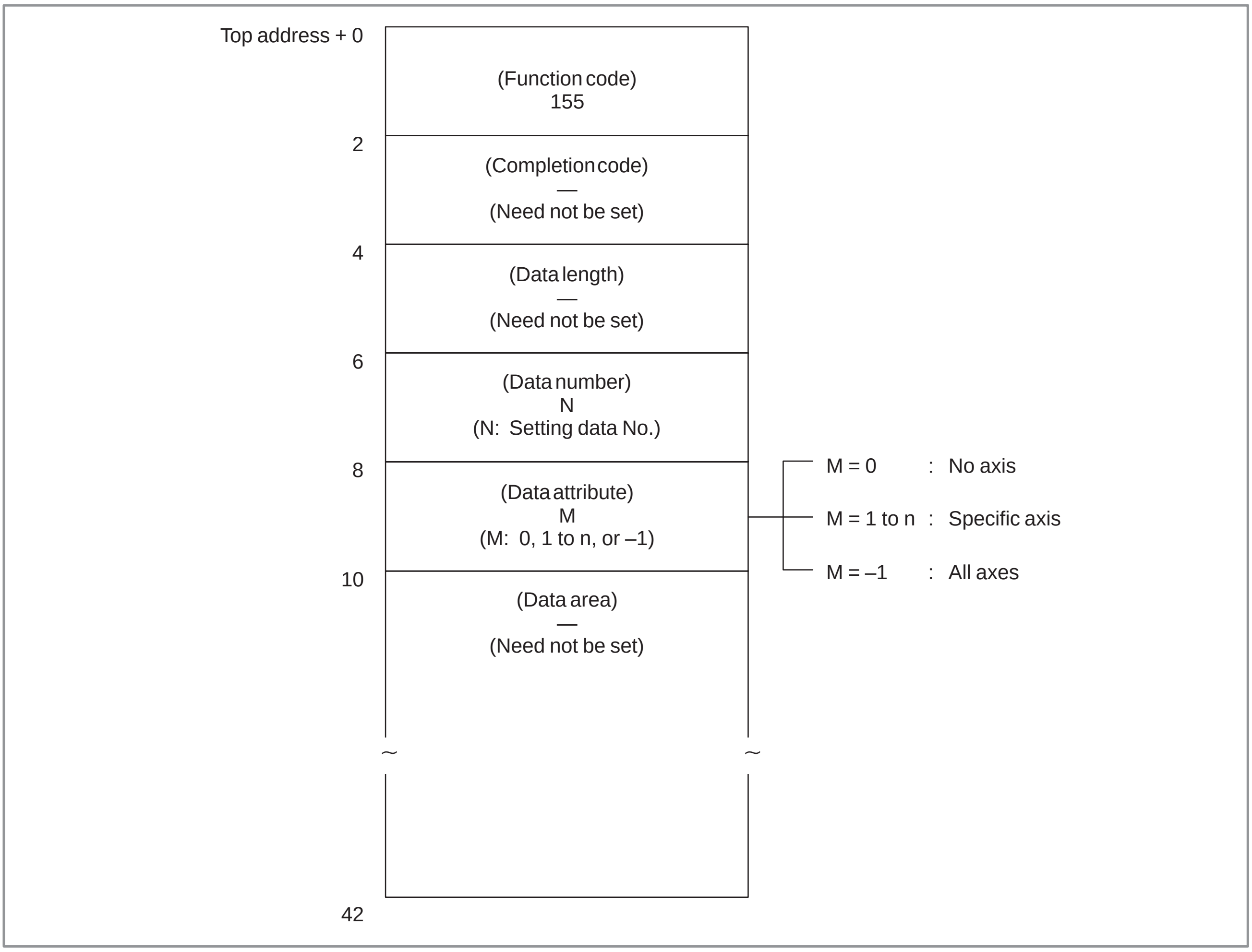
B.4.42 Reading Set Data (not available for Power Mate–D/F, Series 21–TA)

[Description]

Set data stored in the CNC can be read directly from the CNC via the FANUC bus.

This function is basically the same as the function described in Section 3.8 ”Reading Set Data,” except that the function code is 155 and some of the completion codes are different.

[Input data structure]



[Completion codes]

0 : Set data has been read normally.

3 : The set data No. specified to be read is invalid.

4 : A value other than 0, –1, and 1 to n (number of axes) was specified for the data attribute.

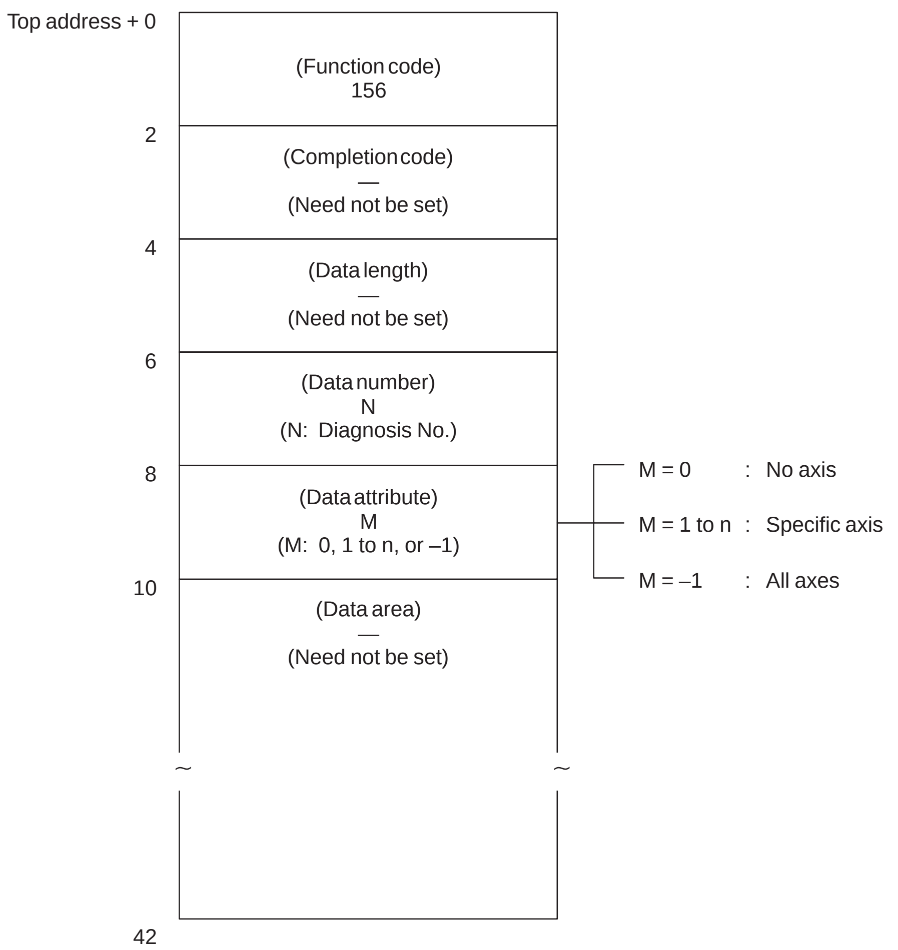
B.4.43 Reading Diagnosis Data (not available for Power Mate–D/F, Series 21–TA)

[Description]

Data displayed on the diagnosis data screen of the CNC can be read directly from the CNC via the FANUC bus.

This function is basically the same as the function described in Section 3.22 ”Reading Diagnosis Data,” except that the function code is 156 and some of the completion codes are different.

[Input data structure]



[Completion codes]

0 : Diagnosis data has been read normally from the CNC.

3 : The diagnosis No. specified to be read is invalid.

4 : A value other than 0, –1, and 1 to n (number of axes) was specified for the data attribute.

6 : An option required for using the diagnosis data to be read, such as the remote buffer option, is not provided.

B.4.44 Reading a Character String of the CNC Program Being Executed in the Buffer

[Description]

In a machining program being executed on the CNC, the block currently executed, the next block, and the next block but one can be read in the CNC program format. That is, these blocks can be read in the form of a character string of ASCII codes. This function is available only with the M series CNCs.

Comments in a block can also be read.

The maximum number of characters in a character string is fixed to 64.

[Input data structure]

Address	Content
Top address + 0	(Function code) 157
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) — (Need not be set)
8	(Data attribute) M (M: Specified block)
10	(Data area) — (Need not be set)
74	

M = 0 : Current block
M = 1 to n : Next block
M = –1 : Next block but one

NOTE

1 When data specified by the NC is a macro statement, the character string cannot be read correctly.

2 When data attribute M is set to 2, the next block but one can be read only when the next block is an instruction for tool diameter compensation C.

[Completion codes]

0 : The character string of the CNC program being executed in the buffer has been read normally.

4 : The value specified for the data attribute is invalid.

[Output data structure]

Top address + 0	(Function code) 157	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 64	64 characters
6	(Data number) —	
8	(Data attribute) M (M: Specified block)	
10	NC command data for the first character RC(1)	ASCII code If a block consists of less than 64 characters, the remaining bytes are filled with 20H (space).
11	NC command data for the second character RC(2)	
~	~	~
73	NC command data for the 64th character RC(64)	

B.4.45 Reading the Relative Position on a Controlled Axis

[Description]

The relative coordinates of the machine moving along an axis controlled by the CNC can be read.

[Input data structure]

Address	Contents
Top address + 0	(Function code) 74
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) 0
8	(Data attribute) M (M: 1 to n or –1)
10	(Data area) — (Need not be set)
~	~
42	

M = 1 to n : Reads the relative coordinates of each axis. n is an axis No.

M = –1 : Reads the relative coordinates of all axes.

[Completion codes]

0 : The relative coordinates on the controlled axis have been read normally.

4 : The specified data attribute is invalid. That is, a value other than –1 and 1 to n (number of axes) was specified, or the specified axis No. was greater than the number of controlled axes.

[Output data structure]

Address	Item	Value
Top address + 0	(Function code) 74	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 4*n. n is the number of specified axes.)	
6	(Data number) —	
8	(Data attribute) M (M: Input data)	
10	Relative coordinates on the specified controlled axis (4 bytes)	Signed binary (A negative value is represented in 2's complement.)

When the number of controlled axes is 4

Address	Item	Value
10	Relative coordinates on the first axis (4 bytes)	Signed binary (A negative value is represented in 2's complement.)
14	Relative coordinates on the second axis (4 bytes)	
18	Relative coordinates on the third axis (4 bytes)	
22	Relative coordinates on the fourth axis (4 bytes)	

Output data unit

Input system	Increment system IS–B	Increment system IS–C
mm, deg system	0.001	0.0001
inch system	0.0001	0.00001
mm, deg system	0.001	0.0001
	0.001	0.0001
inch system	0.0001	0.00001
	0.0001	0.00001

Double values can be read for a machining center system or when radius specification is used for the relevant axis of a lathe system.

B.4.46 Reading the Remaining Travel

[Description]

The remaining travel of the machine along an axis controlled by the CNC can be read. The read value equals the remaining travel indicated on the current position display screen on the CNC. (This screen can be called by pressing the function button POS.)

[Input data structure]

Top address + 0	(Function code) 75
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) 0
8	(Data attribute) M (M: 1 to n or –1)
10	(Data area) — (Need not be set) ~ ~
42	

M = 1 to n: Reads the remaining travel along each axis. n is an axis No.

M = –1 : Reads the remaining travel along all axes.

[Completion codes]

0 : The remaining travel along the controlled axis has beenread normally.

4 : The specified data attribute is invalid. That is, a value other than –1 and 1 to n (number of axes) was specified, or the specified axis No. was greater than the number of controlled axes.

[Output data structure]

Top address + 0	(Function code) 75	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) L (L = 4*n. n is the number of specified axes.)	
6	(Data number) —	
8	(Data attribute) M (M: Input data)	Value
10	Remaining travel along the specified controlled axis (4 bytes)	Signed binary (A negative value is represented in 2's complement.)

When the number of controlled axes is 4

		Value
10	Remaining travel along the first axis (4 bytes)	Signed binary (A negative value is represented in 2's complement.)
14	Remaining travel along the second axis (4 bytes)	
18	Remaining travel along the third axis (4 bytes)	
22	Remaining travel along the fourth axis (4 bytes)	

Output data unit

		Input system	Increment system IS–B	Increment system IS–C
Machining center system Power Mate–D, F		mm, deg system	0.001	0.0001
		inch system	0.0001	0.00001
Lathe system	Radius specification	mm, deg system	0.001	0.0001
	Diameter specification		0.0005	0.00005
	Radius specification	inch system	0.0001	0.00001
	Diameter specification		0.00005	0.000005

B.4.47 Reading CNC Status Information

[Description]

Status information (status indication on the screen) can be read from the CNC.

The types of status information that can be read are as follows.

(1) Indication of which mode is selected, automatic or manual

(2) Status of automatic operation

(3) Status of movement along the axis and dwelling

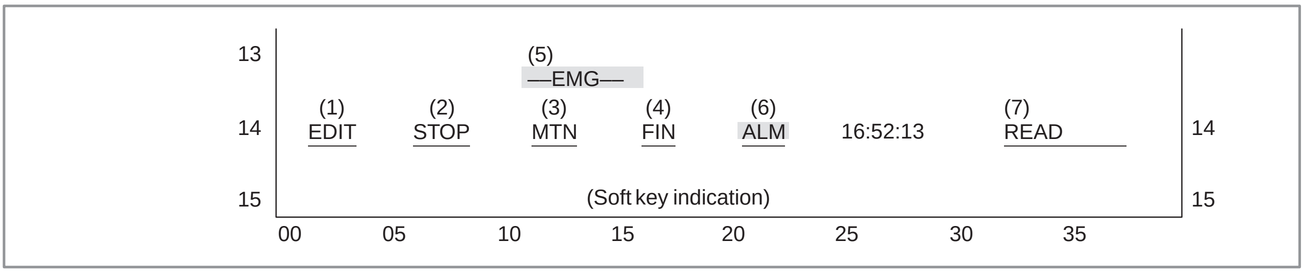
(4) Status of M, S, T, and B functions

(5) Statuses of emergency stop and the reset signal

(6) Alarm status

(7) Status of program edit

(Indication)



[Input data structure]

Offset	Content
Top address + 0	(Function code) 76
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data number) 0
8	(Data attribute) 0
10	(Data area) — (Need not be set)
42	

[Completion codes]

0 : CNC status information has been read normally.

[Output data structure]

Address	Item	Value
Top address + 0	(Function code) 76	
2	(Completion code) ? (See the explanation of the completion codes.)	
4	(Data length) 14	
6	(Data number) — (Input data)	
8	(Data attribute) — (Input data)	
10	Indication of which mode is currently selected, automatic or manual (2 bytes)	0 : MDI 1 : MEMory 2 : **** (Other states) 3 : EDIT 4 : HaNDle 5 : JOG 6 : Teach in JOG 7 : Teach in HND 8 : INC. feed 9 : REFerence 10: ReMoTe
12	Status of automatic operation (2 bytes)	0 : **** (Reset states) 1 : STOP 2 : HOLD 3 : STaRT
14	Status of movement along the axis or dwelling (2 bytes)	0 : *** (Other states) 1 : MoTioN 2 : DWell
16	Status of M, S, T, and B functions (2 bytes)	0 : *** (Other states) 1 : FIN
18	Status of emergency stop (2 bytes)	0 : (Releases the emergency stop state) 1 : — —EMerGency — — 2 : — RESET — (The reset signal is on.)
20	Alarm status (2 bytes)	0 : *** (Other states) 1 : ALarM 2 : BATtery low
22	Status of program edit (2 bytes)	0 : ******* (Non editing) 1 : EDIT 2 : SeaRCH 3 : OUTPUT 4 : INPUT 5 : COMPARE 6 : LabelSKip 7 : OFST 8 : WSFT 9 : ReSTaRt

B.4.48 Reading Value of the P–code Macro Variable (∗Low–speed Response)

[Description]

This function gets the value of variable for Macro–compiler (P–code macro variable) of specified number.

The extended P–code macro variable is not able to be read.

[Input data structure]

Address	Content
Top address + 0	(Function code) 59
+ 2	(Completion code) — (Need not be set)
+ 4	(Data length) — (Need not be set)
+ 6	(Data number) N (P–code macro variable number)
+ 10	(Data attribute) 0
+ 12	(Data area) — (Need not be set)
+ 18	

CAUTION

The ’Data number’ occupies 4 bytes instead of 2 bytes of usual data structure.

[Completion codes]

0 : Success to read the value of P–code macro variable.

3 : The P–code macro variable specified by 'Data number' is not able to be read.

5 : The value of the P–code macro variable is out of range ($\pm$0.0000001 – $\pm$99999999).

6 : No option, or no Macro ROM module.

[Output data structure]

Offset	Field	Value
Top address + 0	(Function code) 59	
+ 2	(Completion code) ? (See the explanation above)	
+ 4	(Data length) 6	
+ 6	(Data number) N (Same as input data)	
+ 10	(Data attribute) — (Same as input data)	Value
+ 12	Value of P–code macro variable (4bytes)	Signed binary (Minus number is represented by 2's complemental)
+ 16	Figures after decimal point of the value (2bytes)	Signed binary (Minus number is represented by 2's complemental)

B.4.49 Writing Value of the P–code Macro Variable (✳Low–speed Response)

[Description]

This function stores the value into the variable for Macro–compiler (P–code macro variable) of specified number.

The extended P–code macro variable is not able to be written into.

[Input data structure]

Address	Field	Value
Top address + 0	(Function code) 60	
+ 2	(Completion code) — (Need not be set)	
+ 4	(Data length) 6	
+ 6	(Data number) N (P–code macro variable number)	
+ 10	(Data attribute) 0	
		Value
+ 12	Value of P–code macro variable (4bytes)	Signed binary (Minus number is represented by 2's complemental)
+ 16	Figures after decimal point of the value (2bytes)	Signed binary (Minus number is represented by 2's complemental)

CAUTION

The 'data number' occupies 4 bytes instead of 2 bytes of usual data structure.

[Completion codes]

0 : Success to store the value into P–code macro variable.

2 : The data length has illegal data (is not 6).

3 : The P–code macro variable specified by ’Data number’ is not able to be written.

6 : No option, or no Macro ROM module.

[Output data structure]

Top address + 0	(Function code) 60
+ 2	(Completion code) ? (See the explanation above)
+ 4	(Data length) 6 (Same as input data)
+ 6	(Data number) N (Same as input data)
+ 10	(Data attribute) — (Same as input data)
+ 12	Value of P–code macro variable (4bytes)
+ 16	Figures after decimal point of the value (2bytes)

B.4.50 Reading the Tool Life Management Data (Tool Life Counter Type) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function gets the Tool life counter type of specified tool group in the Tool life management data. (M series only)

[Input data structure]

Offset	Content
Top address + 0	(Function code) 160
+ 2	(Completion code) — (Need not be set)
+ 4	(Data length) — (Need not be set)
+ 6	(Data number) N (N = Tool group number)
+ 8	(Data attribute) 0
+ 10	(Data area) — (Need not be set)
+ 12	

NOTE

About Tool group number (in 'Data number')
"0" as Tool group number indicates the Tool group currently used.
When Tool group has never specified since power–on, "0" of Tool group number results "0" as counter type.
"0" of counter type will be returned on T series.

[Completion codes]

0 : Success to read the Tool life counter type.

3 : The Tool group number is out of range from 0 to 512, or exceeds the maximum number of registered Tool group.

6 : No option for Tool life management.

[Output data structure]

Top address + 0	(Function code) 160	
+ 2	(Completion code) ? (See the explanation above)	
+ 4	(Data length) 2	
+ 6	(Data number) N (Same as input data)	
+ 8	(Data attribute) — (Same as input data)	Value
+ 10	Tool life counter type (2bytes)	0 : No counter type 1 : Frequency 2 : Real time (in minutes)

B.4.51 Registering the Tool Life Management Data (Tool Group) (∗Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function registers the Tool group in Tool life management data, with Tool number, length of life and Tool life counter type. On T series, the Tool life counter type will be specified by the NC parameter ”LTM” (No.6800#2), and this function cannot set/change the counter type.

[Input data structure]

Top address + 0	(Function code) 163	
+ 2	(Completion code) — (Need not be set)	
+ 4	(Data length) 8	
+ 6	(Data number) 0	
+ 8	(Data attribute) M (M = Tool number)	Value
+ 10	Tool group number (2bytes)	Unsigned binary 1 to 512
+ 12	Tool life counter type (2bytes)	1 : Frequency 2 : Real time in minutes
+ 14	Length of Tool life (4bytes)	Unsigned binary 1 to 9999 (Frequency) 1 to 4300 (Real time in minutes)

[Completion codes]

0 : Success to register the Tool group.

3 : The Tool group number is out of range from 1 to 512, or exceeds the maximum number of registered Tool group.

4 : The Tool number in ’Data attribute’ has wrong value.

5 : The length of Tool life in ’Data area’ is out of range. The Tool life counter type does not match on T series.

6 : No option for Tool life management.

[Output data structure]

Top address + 0	(Function code) 163
+ 2	(Completion code) ? (See the explanation above)
+ 4	(Data length) 8 (Same as input data)
+ 6	(Data number) — (Same as input data)
+ 8	(Data attribute) M (Same as input data)
+ 10	Tool group number (2bytes) (Same as input data)
+ 12	Tool life counter type (2bytes) (Same as input data)
	Length of Tool life (4bytes) (Same as input data)

B.4.52 Writing the Tool Life Management Data (Tool Life) (∗Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function sets the length of Tool life of the specified Tool group in the Tool life management data.

[Input data structure]

Address	Item	Value
Top address + 0	(Function code) 164	
+2	(Completion code) — (Need not be set)	
+4	(Data length) 4	
+6	(Data number) N (N = Tool group number)	
+8	(Data attribute) 0	
+10	Length of Tool life (4bytes)	Unsigned binary 1–9999 (Frequency) 1–4300 (Real time in minutes)

[Completion codes]

0 : Success to set the length of Tool life.

3 : The Tool group number is out of range from 1 to 512, or exceeds the maximum number of registered Tool group.

5 : The length of Tool life is out of range.

6 : No option for Tool life management.

[Output data structure]

Top address +0	(Function code) 164
+2	(Completion code) ? (See the explanation above)
+4	(Data length) 4 (Same as input data)
+6	(Data number) N (Same as input data)
+8	(Data attribute) — (Same as input data)
+10	Length of Tool life (4bytes)

B.4.53 Writing the Tool Life Management Data (Tool Life Counter) (∗Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function sets the Tool life counter in the specified Tool group in the Tool life management data.

[Input data structure]

Address	Content	Value
Top address + 0	(Function code) 165	
+2	(Completion code) — (Need not be set)	
+4	(Data length) 4	
+6	(Data number) N (N = Tool group number)	
+8	(Data attribute) 0	
+10	Length of Tool life (4bytes)	Unsigned binary 1–9999 (Frequency) 1–4300 (Real time in minutes)

[Completion codes]

0 : Success to set the Tool life counter.

3 : The Tool group number is out of range from 1 to 512, or exceeds the maximum number of registered Tool group.

5 : The value for Tool life counter is out of range.

6 : No option for Tool life management.

[Output data structure]

Top address + 0	(Function code) 165
+2	(Completion code) ? (See the explanation above)
+4	(Data length) 4 (Same as input data)
+6	(Data number) N (Same as input data)
+8	(Data attribute) — (Same as input data)
+10	Length of Tool life (4bytes)

B.4.54 Writing the Tool Life Management Data (Tool Life Counter Type) (✻Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function sets the Tool life counter type of specified Tool group in the Tool life management data. (M series only)

[Input data structure]

Top address + 0	(Function code) 166	
+2	(Completion code) — (Need not be set)	
+4	(Data length) 2	
+6	(Data number) N (N = Tool group number)	
+8	(Data attribute) 0	Value
+10	Tool life counter type (2bytes)	1 : Frequency 2 : Real time in minutes

[Completion codes]

0 : Success to set the Tool life counter type.

3 : The Tool group number is out of range from 1 to 512, or exceeds the maximum number of registered Tool group.

5 : The value for Tool life counter type is wrong.

6 : No option for Tool life management.

[Output data structure]

Top address + 0	(Function code) 166
+2	(Completion code) ? (See the explanation above)
+4	(Data length) 2 (Same as input data)
+6	(Data number) N (Same as input data)
+8	(Data attribute) — (Same as input data)
+10	Tool life counter type (2bytes)

B.4.55 Writing the Tool Life Management Data (Tool Length Offset Number (1) : Tool Number) (✻Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function sets the Tool length offset number of the specified Tool group in the Tool life management data. (M series only)

[Input data structure]

Top address + 0	(Function code) 167	
+2	(Completion code) — (Need not be set)	
+4	(Data length) 4	
+6	(Data number) N (N = Tool group number)	
+8	(Data attribute) M (M = Tool number)	Value
+10	Tool length offset number (4bytes)	Unsigned binary 1–255

[Completion codes]

0 : Success to set the Tool length offset number.

3 : The Tool group number is out of range from 1 to 512, or exceeds the maximum number of registered Tool group.

4 : The Tool number in 'Data attribute' has wrong value.

5 : The Tool number is not found in the Tool group.

6 : No option for Tool life management.

[Output data structure]

Top address + 0	(Function code) 167
+2	(Completion code) ? (See the explanation above)
+4	(Data length) 4 (Same as input data)
+6	(Data number) N (Same as input data)
+8	(Data attribute) M (Same as input data)
+10	Tool length offset number (4bytes)

NOTE
The effective value for Tool length offset number depends on Tool compensation number available on NC.

B.4.56 Writing the Tool Life Management Data (Tool Length Offset Number (2) : Tool Operation Sequence Number) (∗Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function sets the Tool length offset number of the Tool of the specified Tool operation sequence number in the Tool life management data. (M series only)

[Input data structure]

Address	Item	Value
Top address + 0	(Function code) 168	
+2	(Completion code) — (Need not be set)	
+4	(Data length) 4	
+6	(Data number) N (N = Tool group number)	
+8	(Data attribute) M (M = Tool operation sequence number)	
+10	Tool length offset number (4bytes)	Unsigned binary 1–255

[Completion codes]

0 : Success to set the Tool length offset number.

3 : The Tool group number is out of range from 1 to 512, or exceeds the maximum number of registered Tool group.

4 : The Tool operation sequence number is wrong.

6 : No option for Tool life management.

[Output data structure]

Top address + 0	(Function code) 168
+2	(Completion code) ? (See the explanation above)
+4	(Data length) 4 (Same as input data)
+6	(Data number) N (Same as input data)
+8	(Data attribute) M (Same as input data)
+10	Tool length offset number (4bytes)

NOTE
The effective value for Tool length offset number depends on Tool compensation number available on NC.

B.4.57 Writing the Tool Life Management Data (Cutter Compensation Number (1) : Tool Number) (∗Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function sets the Cutter compensation number of the specified Tool group in the Tool life management data. (M series only)

[Input data structure]

Address	Item	Value
Top address + 0	(Function code) 169	
+2	(Completion code) — (Need not be set)	
+4	(Data length) 4	
+6	(Data number) N (N = Tool group number)	
+8	(Data attribute) M (M = Tool number)	
+10	Cutter compensation number (4bytes)	Unsigned binary 1–255

[Completion codes]

0 : Success to set the Cutter compensation number.

3 : The Tool group number is out of range from 1 to 512, or exceeds the maximum number of registered Tool group.

4 : The Tool number in 'Data attribute' has wrong value.

5 : The Tool number is not found in the Tool group.

6 : No option for Tool life management.

[Output data structure]

Address	Content
Top address + 0	(Function code) 169
+2	(Completion code) ? (See the explanation above)
+4	(Data length) 4 (Same as input data)
+6	(Data number) N (Same as input data)
+8	(Data attribute) M (Same as input data)
+10	Cutter compensation number (4bytes)

NOTE
The effective value for Cutter compensation number depends on Tool compensation number available on NC.

B.4.58 Writing the Tool Life Management Data (Cutter Compensation Number (2) : Tool Operation Sequence Number) (∗Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function sets the Cutter compensation number of the Tool of the specified Tool operation sequence number in the Tool life management data. (M series only)

[Input data structure]

Address	Data	Value
Top address + 0	(Function code) 170	
+2	(Completion code) — (Need not be set)	
+4	(Data length) 4	
+6	(Data number) N (N = Tool group number)	
+8	(Data attribute) M (M = Tool operation sequence number)	
+10	Cutter compensation number (4bytes)	Unsigned binary 1–255

[Completion codes]

0 : Success to set the Cutter compensation number.

3 : The Tool group number is out of range from 1 to 512, or exceeds the maximum number of registered Tool group.

4 : The Tool operation sequence number is wrong.

6 : No option for Tool life management.

[Output data structure]

Top address + 0	(Function code) 170
+2	(Completion code) ? (See the explanation above)
+4	(Data length) 4 (Same as input data)
+6	(Data number) N (Same as input data)
+8	(Data attribute) M (Same as input data)
+10	Cutter compensation number (4bytes)

NOTE
The effective value for Cutter compensation number depends on Tool compensation number available on NC.

B.4.59 Writing the Tool Life Management Data (Tool Condition (1) : Tool Number) (∗Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function sets the Tool condition of the specified Tool group in the Tool life management data.(M series only)

[Input data structure]

Top address + 0	(Function code) 171	
+2	(Completion code) — (Need not be set)	
+4	(Data length) 2	
+6	(Data number) N (N = Tool group number)	
+8	(Data attribute) M (M = Tool number)	Value
+10	Tool condition (2bytes)	1 : Tool state clear 2 : Tool state skip

[Completion codes]

0 : Success to set the Tool condition.

3 : The Tool group number is out of range from 1 to 512, or exceeds the maximum number of registered Tool group.

4 : The Tool number in ’Data attribute’ has wrong value.

5 : The Tool number is not found in the Tool group.

6 : No option for Tool life management.

[Output data structure]

Top address + 0	(Function code) 171
+2	(Completion code) ? (See the explanation above)
+4	(Data length) 2 (Same as input data)
+6	(Data number) N (Same as input data)
+8	(Data attribute) M (Same as input data)
+10	Tool condition (2bytes)

This function changes Tool condition as below.

command	before call	after call
clear	skip (#) skip (#) consumed (*)	usable ( ) in use (@) usable ( )
skip	unused ( ) in use (@) consumed (*)	skip (#) skip (#) skip (*)

B.4.60 Writing the Tool Management Data (Tool Condition (2) : Tool Operation Sequence Number) (∗Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function sets the Tool condition of the Tool of the specified Tool operation sequence number in the Tool life management data.

[Input data structure]

Top address + 0	(Function code) 172	
+2	(Completion code) — (Need not be set)	
+4	(Data length) 2	
+6	(Data number) N (N = Tool group number)	
+8	(Data attribute) M (M = Tool operation sequence number)	Value
+10	Tool condition (2bytes)	1 : Tool state clear 2 : Tool state skip

[Completion codes]

0 : Success to set the Tool condition.

3 : The Tool group number is out of range from 1 to 512, or exceeds the maximum number of registered Tool group.

4 : The Tool operation sequence number is wrong.

6 : No option for Tool life management.

[Output data structure]

Top address + 0	(Function code) 172
+2	(Completion code) ? (See the explanation above)
+4	(Data length) 2 (Same as input data)
+6	(Data number) N (Same as input data)
+8	(Data attribute) M (Same as input data)
+10	Tool condition (2bytes)

This function changes Tool condition as shown in B.4.60.

B.4.61 Writing the Tool Life Management Data (Tool Number) (✻Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function registers a tool to the specified Tool group in the Tool life management data.

[Input data structure]

Top address + 0	(Function code) 173	
+2	(Completion code) — (Need not be set)	
+4	(Data length) 4	
+6	(Data number) N (N = Tool group number)	
+8	(Data attribute) M (M = Tool operation sequence number)	Value
+10	Tool number (4bytes)	Unsigned binary 1–9999

[Completion codes]

0 : Success to register the Tool number.

3 : The Tool group number is out of range from 1 to 512, or exceeds the maximum number of registered Tool group.

4 : The Tool operation sequence number is wrong.

6 : No option for Tool life management.

[Output data structure]

Top address + 0	(Function code) 173
+2	(Completion code) ? (See the explanation above)
+4	(Data length) 4 (Same as input data)
+6	(Data number) N (Same as input data)
+8	(Data attribute) M (Same as input data)
+10	Tool number (4bytes)

B.4.62 Reading the Estimate Disturbance Torque Data

Power Mate	FS20	FS18	FS16
×	×	Δ	Δ

Δ : The support is decided by CNC series
× : No support

(1) servo axis

[Description]

The load torques except a necessary torque for acceleration/ deceleration of the digital servo axis are read.

[Input data structure]

Address	Contents	
Top address + 0	(Function code) 211	
+2	(Completion code) – (Need not to be set)	
+4	(Data length) – (Need not to be set)	
+6	(Data number) 0	
+8	(Data attribute) M (M=1 to n or – 1)	M=1 to n : Estimate disturbance torque data for specific axis. "n" is the axis number. M= – 1 : Estimate disturbance torque data for all axes.
+10	(Data area) – (Need not to be set)	
+12		

[Completion codes]

0 : The estimate disturbance torque data have been read normally.

4 : The data specified as the data attribute is invalid because it is neither –1 nor a value from 1 to n (n is the number of axes). Alternatively, the specified axis number is greater than the number of controlled axes.

[Output data structure]

Address	Contents	Value
Top address + 0	(Function code) 211	
+2	(Completion code) ? (See the explanation of the completion codes.)	
+4	(Data length) L (L=2 × n, n is the number of axes specified.)	
+6	(Data number) 0	
+8	(Data attribute) M (M : Input data)	Value
+10	Estimate disturbance torque data for-the controlled axis specified (2 bytes)	(A negative value is represented in 2's complement.)
+12		

When the number of controlled axes is 4

Address	Contents	Value
		Value
+10	Estimate disturbance torque data forfirst axis (2 bytes)	Signed binary (A negative value is represented in 2's complement.)
+12	Estimate disturbance torque data forsecond axis (2 bytes)	
+14	Estimate disturbance torque data forthird axis (2 bytes)	
+16	Estimate disturbance torque data forfourth axis (2 bytes)	
+18		

(2) spindle axis

[Description]

The load torques except a necessary torque for acceleration/ deceleration of the serial spindle axis are read.

[Input data structure]

Address	Content	Remarks
Top address + 0	(Function code) 211	
+2	(Completion code) — (Need not to be set)	
+4	(Data length) — (Need not to be set)	
+6	(Data number) 1	
+8	(Data attribute) M (M=1 to n or – 1)	M= 1 to n: Read the load of each spindle. (n is the spindle number.) –1: Read the loads of the No.1 and No.2 spindles. –2: Read the loads of the No.1 to No.3 spindles. –3: Read the loads of the No.1 to No.4 spindles.
+10	(Data area) – (Need not to be set)	
+12		

[Completion codes]

0 : The estimate disturbance torque data have been read normally.

4 : The data specified as the data attribute is invalid because it is neither – 1 nor a value from 1 to n (n is the number of axes) . Alternatively, the specified axis number is greater than the number of controlled axes.

[Output data structure]

Top address + 0	(Function code) 211	
+2	(Completion code) ? (See the explanation of the completion codes.)	
+4	(Data length) L (L=2 × n, n is the number of axes specified.)	
+6	(Data number) 1	
+8	(Data attribute) M (M : Input data)	Value
+10	Estimate disturbance torque data forthe controlled axis specified (2 bytes)	Signed binary (A negative value is represented in 2’s complement.)
+12		

When the number of controlled axes is 2

		Value
+10	Estimate disturbance torque data for first axis (2 bytes)	Signed binary (A negative value is represented in 2’s complement.)
+12	Estimate disturbance torque data forsecond axis (2 bytes)	

[supporting soft]

CNC FS16 :	B005 SERIES	Edition K or later
	B105 SERIES	Edition H or later
	B205 SERIES	Edition H or later
FS18 :	BD03 SERIES	Edition L or later
	BE03 SERIES	Edition I or later
	BF03 SERIES	Edition I or later
SERVO :	9060 SERIES	Edition J or later
SPINDLE :	9A50 SERIES	Edition Q or later

* Some of the series not listed above are supported by the first edition.

B.4.63 Reading the Current Program Number (8–digit Program Numbers) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function reads CNC program numbers extended to 8 digits from the usual 4 digits.
Basically, this function is the same as function number 24 excluding the different data length of function code 90.

[Input data structure]

Address	Content
Top address 0	(Function code) 90
2	(Completion code) – (Need not be set)
4	(Data length) – (Need not be set)
6	(Data number) 0
8	(Data attribute) 0
10	(Data area) – (Need not be set)
18	

[Completion codes]

–1 : The read command of the currently executing program could not be executed. That is, the same command could not be executed as the data of the program number was being updated on the CNC.

0 : The program number of the currently executing program has been read normally.

[Output data structure]

Top address	Item	Value
0	(Function code) 90	
2	(Completion code) ? (See the explanation of the completion codes, above.)	
4	(Data length) 8	
6	(Data number) –	
8	(Data attribute) –	
10	Number of the program currently being executed ON	Unsigned binary format, 4–byte length
14	Program number of the main program ON	
18		

(a) Number of the program currently being executed (ON)
The program number of the program currently being executed is set.

(b) Program number of the main program (OFF)
If the program currently being executed is a subprogram, the program number of its main program is set.
If the program currently being executed is not a subprogram, 0 is set.

B.4.64 Reading Tool Life Management Data (Tool Group Number) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function reads the tool group number to which the tool number is currently registered.

[Input data structure]

Address	Content
Top address 0	(Function code) 200
2	(Completion code) – (Need not be set)
4	(Data length) – (Need not be set)
6	(Data number) 0
8	(Data attribute) M (M=Tool number)
12	(Data area) – (Need not be set)
16	

NOTE

When the tool number is set to “0”, the tool group number of the currently used tool is read.
If a tool group number is not specified after the power is turned ON, tool group number “0” is read. Also, if a tool number is registered to two or more tool group numbers, the tool group numbers of all tool groups to which the tool number is registered are read.

[Completion codes]

0 : The tool group number was read successfully.

4 : The tool number in ’Data Attribute’ has a wrong value.

5 : The tool number is not registered.

6 : The tool life management option has not been added on.

[Output data structure]

Top address		Value	
0	(Function code) 200		
2	(Completion code) ? (See the explanation above.)		
4	(Data length) L (L=4×n)		L=4 to 4 x n n is the number of tool group numbers when multiple groups is specified.
6	(Data number) –		
8	(Data attribute) M (M: entered data)		
12	Tool group number (4 bytes)	Unsigned binary	
16			

Or, when the tool number is registered to multiple groups

		Value
12	Tool group number (4 bytes)	Unsigned binary
16	Tool group number (4 bytes)	
20	Tool group number (4 bytes)	
24		

B.4.65 Reading Tool Life Management Data (Tool Length Offset Number 1) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function reads the tool length offset number according to the specified tool group number and tool number. (M series only)

[Input data structure]

Top address	
0	(Function code) 227
2	(Completion code) – (Need not be set)
4	(Data length) – (Need not be set)
6	(Data number) N (N=Tool group number)
8	(Data attribute) M (M=Tool number)
12	(Data area) – (Need not be set)
16	

NOTE
When the tool group number and tool number are set to “0”, the currently used tool group and tool number are referenced.
If a tool group number is not specified after the power is turned ON, tool group number “0” is read.
“0” is always read on the T series.

[Completion codes]

0 : The tool length offset number was read successfully.

3 : The specified tool group number is incorrect.

4 : The specified tool number is incorrect.

5 : The specified tool number is not registered to the specified tool group.

6 : The tool life management option has not been added on.

[Output data structure]

Top address		Value
0	(Function code) 227	
2	(Completion code) ? (See the explanation above.)	
4	(Data length) 4	
6	(Data number) N (N: entered data)	
8	(Data attribute) M (M: entered data)	
12	Tool offset number (4 bytes)	Unsigned binary
16		

B.4.66 Reading Tool Life Management Data (Tool Diameter Offset Number 1) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function reads the tool radius offset number according to the specified tool group number and tool number. (M series only)

[Input data structure]

Address	Contents
Top address 0	(Function code) 228
2	(Completion code) – (Need not be set)
4	(Data length) – (Need not be set)
6	(Data number) N (N=Tool group number)
8	(Data attribute) M (M=Tool number)
12	((Data area)) – (Need not be set)
16	

NOTE

When the tool group number and tool number are set to “0”, the currently used tool group and tool number are referenced.

If a tool group number is not specified after the power is turned ON, tool group number “0” is read.

“0” is always read on the T series.

[Completion codes]

0 : The tool radius offset number was read successfully.

3 : The specified tool group number is incorrect.

4 : The specified tool number is incorrect.

5 : The specified tool number is not registered to the specified tool group.

6 : The tool life management option has not been added on.

[Output data structure]

Top address	Field	Value
0	(Function code) 228	
2	(Completion code) ? (See the explanation above.)	
4	(Data length) 4	
6	(Data number) N (N: entered data)	
8	(Data attribute) M (M: entered data)	
12	Tool offset number (4 bytes)	Unsigned binary
16		

B.4.67 Reading Tool Life Management Data (Tool Information 1) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function reads the tool information (status) according to the specified tool group number and tool number.

[Input data structure]

Address	Content
Top address 0	(Function code) 201
2	(Completion code) – (Need not be set)
4	(Data length) – (Need not be set)
6	(Data number) N (N=Tool group number)
8	(Data attribute) M (M=Tool number)
12	((Data area)) – (Need not be set)
16	

NOTE

When the tool group number and tool number are set to “0”, the currently used tool group and tool number are referenced.

If a tool group number is not specified after the power is turned ON, tool group number “0” is read.

[Completion codes]

0 : The tool information was read successfully.

3 : The specified tool group number is incorrect.

4 : The specified tool number is incorrect.

5 : The specified tool number is not registered to the specified tool group.

6 : The tool life management option has not been added on.

[Output data structure]

Address	Item	Value
Top address 0	(Function code) 201	
2	(Completion code) ? (See the explanation above.)	
4	(Data length) 4	
6	(Data number) N (N: entered data)	
8	(Data attribute) M (M: entered data)	Value
12	Tool offset number (4 bytes)	0: See Note in previous item/ 1: Tool is registered. 2: End of tool life. 3: Tool skipped All above three bytes must be “0”.
16		

B.4.68 Writing (Registering) Tool Life Management Data (Tool Group Number) (✻Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function registers the tool group number to tool life management data. Set the tool number, life value and life counter type to the specified tool group. On the T series, since the life counter type is specified by CNC parameter LTM (No. 6800#2), it cannot be set nor changed here.

[Input data structure]

Address	Field	Value
Top address 0	(Function code) 202	
2	(Completion code) ? (Need not be set)	
4	(Data length) 8	
6	(Data number) 0	
8	(Data attribute) M (M: entered data)	
12	Tool group number (2 bytes)	Unsigned binary 1 to 512
14	Tool life counter type (2 bytes)	1: Count 2: Time (minutes)
16	Tool life value (4 bytes)	Unsigned binary 1 to 9999 (count) 2 to 4300 (time)
20		

[Completion codes]

0 : The tool length was registered successfully.

3 : The tool group number exceeded the range 1 to 512 or maximum number of registered groups.

4 : The tool number in 'Data Attribute' has a wrong value.

5 : The tool life value is out–of–range. On the T series, the tool life counter type is different.

6 : The tool life management option has not been added on.

[Output data structure]

Address	Content
Top address 0	(Function code) 202
2	(Completion code) ? (See the explanation above.)
4	(Data length) 8 (Entered data)
6	(Data number) – (Entered data)
8	(Data attribute) M (Entered data)
12	Tool group number (2 bytes) (Entered data)
14	Tool life counter type (2 bytes) (Entered data)
16	Tool life value (4 bytes) (Entered data)
20	

B.4.69 Writing Tool Life Management Data (Tool Length Offset Number 1) (∗Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function sets the tool length offset number of a specified tool group in the tool life management data. (M series only)

[Input data structure]

Top address	Item	Value
0	(Function code) 229	
2	(Completion code) – (Need not be set)	
4	(Data length) 4	
6	(Data number) N (N=Tool group number)	
8	(Data attribute) M (M=Tool number)	
12	Tool length offset number (4 bytes)	Unsigned binary 1 to 255
16		

[Completion codes]

0 : The tool length offset number was written successfully.

3 : The tool group number exceeded the range 1 to 512 or maximum number of registered groups.

4 : The specified tool number is incorrect.

5 : The specified tool number is not registered to the specified tool group.

6 : The tool life management option has not been added on.

[Output data structure]

Address	Data
Top address 0	(Function code) 229
2	(Completion code) ? (See the explanation above.)
4	(Data length) 4 (Entered data)
6	(Data number) N (Entered data)
8	(Data attribute) M (Entered data)
12	Tool length offset number (4 bytes)
16	

NOTE
The tool length offset number that can be actually specified is reliant on the tool offsets available on the NC.

B.4.70 Writing Tool Life Management Data (Tool Radius Offset Number 1) (*Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function sets the tool radius offset number of a tool belonging to a specified tool group in the tool life management data. (M series only)

[Input data structure]

Top address		
0	(Function code) 230	
2	(Completion code) – (Need not be set)	
4	(Data length) 4	
6	(Data number) N (N=Tool group number)	
8	(Data attribute) M (M=Tool number)	Value
12	Tool length offset number (4 bytes)	Unsigned binary 1 to 255
16		

[Completion codes]

0 : The tool radius offset number was written successfully.

3 : The tool group number exceeded the range 1 to 512 or maximum number of registered groups.

4 : The specified tool number is incorrect.

5 : The specified tool number is not registered to the specified tool group.

6 : The tool life management option has not been added on.

[Output data structure]

Top address	
0	(Function code) 230
2	(Completion code) ? (See the explanation above.)
4	(Data length) 4 (Entered data)
6	(Data number) N (Entered data)
8	(Data attribute) M (Entered data)
12	Tool length offset number (4 bytes)
16	

NOTE
The tool length offset number that can be actually specified is reliant on the tool offsets available on the NC.

B.4.71 Writing Tool Life Management Data (Tool Information 1) (✳Low–speed Response) (not available for Power Mate–D/F, Series 21–TA)

[Description]

This function sets the tool information of a tool belonging to a specified tool group in the tool life management data. (M series only)

[Input data structure]

Top address		
0	(Function code) 231	
2	(Completion code) – (Need not be set)	
4	(Data length) 2	
6	(Data number) N (N=Tool group number)	
8	(Data attribute) M (M=Tool number)	Value
12	Tool group number (2 bytes)	1: Clears tool status. 2: Skips tool status.
14		

[Completion codes]

0 : The tool information was written successfully.

3 : The tool group number exceeded the range 1 to 512 or maximum number of registered groups.

4 : The specified tool number is incorrect.

5 : The specified tool number is not registered to the specified tool group.

6 : The tool life management option has not been added on.

[Output data structure]

Address	Contents
Top address 0	(Function code) 231
2	(Completion code) ? (See the explanation above.)
4	(Data length) 2 (Entered data)
6	(Data number) N (Entered data)
8	(Data attribute) M (Entered data)
12	Tool length offset number (2 bytes)
14	

The following table shows how the tool status changes before and after this function is specified.

command	Pre–command Status		Post–command Status	
clear	skip skip used	(#) (#) (*)	unused in use unused	( ) (@) ( )
skip	unused in use used	( ) (@) (*)	skip skip skip	(#) (#) (*)

B.4.72 Reading Actual Spindle Speeds

(1) Actual spindle speed

[Description]

This function reads the actual speed of the No.1 to No.4 serial spindles.

[Input data structure]

Address	Content	Remarks
Top address +0	(Function code) 138	
+2	(Completion code) – (Need not be set)	
+4	(Data length) – (Need not be set)	
+6	(Data number) 0	
+8	(Data attribute) M (M=Spindle number)	M=1 to n : Read spindles on each axis. (n is the spindle number.) –1 : Read spindles on No.1 and No.2 axes –2 : Read spindles on No.1 to No.3 axes –3 : Read spindles on No.1 to No.4 axes
+10	(Data area) – (Need not be set)	

[Completion codes]

0: The actual spindle speed was read successfully.

4: The spindle speed in ’Data Attribute’ has wrong values, that is , a value outside of the range –1 to –(n–1) or 1 to n (n: number of spindles).

[Output data structure]

Address	Field	Value
Top address +0	(Function code) 138	
+2	(Completion code) ? (See the explanation above.)	
+4	(Data length) L (L=4×n)	
+6	(Data number) –	
+8	(Data attribute) M (Entered data)	Value
+10	Actual speed of specified spindle	Signed binary <data unit> rpm
+14		

Or, 4 controlled axes:

Address	Field	Value
		Value
+10	Actual speed of No.1 spindle	Signed binary <Data unit> rpm
+14	Actual speed of No.2 spindle	
+18	Actual speed of No.3 spindle	
+22	Actual speed of No.4 spindle	
+26		

(2) Position coder–less actual spindle speed

[Description]

This function reads the actual spindle speed (position coder–less actual spindle speed) obtained by calculating the spindle motor speed of the No.1 to No.4 serial spindles.

[Input data structure]

Address	Field	Remarks
Top address +0	(Function code) 138	
+2	(Completion code) – (Need not be set)	
+4	(Data length) – (Need not be set)	
+6	(Data number) 0	
+8	(Data attribute) M (M=Spindle number)	M=11 to (10+n) : Read spindles on each axis. (n is the spindle number.) –11 : Read spindles on No.1 and No.2 axes –12 : Read spindles on No.1 to No.3 axes –13 : Read spindles on No.1 to No.4 axes
+10	(Data area) – (Need not be set)	
+12		

[Completion codes]

0 : The actual spindle speed was read successfully.

4 : The spindle speed in 'Data Attribute' has wrong values, that is , a value outside of the range –11 to –(9+1) or 11 to (10+n) (n: number of spindles).

[Output data structure]

Address		Value
Top address +0	(Function code) 138	
+2	(Completion code) ? (See the explanation above.)	
+4	(Data length) L (L=4×n)	
+6	(Data number) –	
+8	(Data attribute) M (Entered data)	
+10	Position coder–less actual spindle speed	Signed binary <data unit> rpm
+14		

Or, 4 controlled axes:

Address		Value
+10	Position coder–less actual No.1 spindle speed	Signed binary <Data unit> rpm
+14	Position coder–less actual No.2 spindle speed	
+18	Position coder–less actual No.3 spindle speed	
+22	Position coder–less actual No.4 spindle speed	
+26		

B.4.73 Reading Fine Torque Sensing Data (Statistical Calculation Results)

[Description]

This function reads the statistical calculation results (average value, maximum value, distribution) in the fine torque sensing function.

[Input data structure]

Offset	Content	Remarks
Top address +0	(Function code) 226	
+2	(Completion code) – (Need not be set)	
+4	(Data length) – (Need not be set)	
+6	(Data number) N (N=1 to n or –1)	
+8	(Data attribute) 0	M=1 to n : Read spindles on each axis. (n is the spindle number.) –1 : Read all axes
+10	(Data area) – (Need not be set)	

[Completion codes]

0 : The statistical calculation results were read successfully.

3 : The fine torque sensing data in ’Data Attribute’ has a wrong value, that is, a value outside of the range –1 or 1 to n (n: number of spindles).

6 : The fine torque sensing option has not been added on.

[Output data structure]

Address	Content	Value
Top address +0	(Function code) 226	
+2	(Completion code) ? (See the above explanation.)	
+4	(Data length) L (L=6×n)	
+6	(Data number) –	
+8	(Data attribute) M (Entered data)	
+10	Average value of target axis	Signed binary
+12	Maximum value of target axis	
+14	Distribution of target axis	
+16		

Or, 4 controlled axes:

Address	Content	Value
+10	Average value of target axis 1	Signed binary (Output only for number of axes specified to parameter Nos. 6390 to 6363)
+12	Maximum value of target axis 1	
+14	Distribution of target axis 1	
+16	Average value of target axis 2	
+18	Maximum value of target axis 2	
+20	Distribution of target axis 2	
	:	
	:	
	:	
+32	Average value of target axis 4	
+34		

B.4.74 Reading Fine Torque Sensing Data (Store Data)

(1) Store counter

[Description]

This function reads the number of stored torque data items.

[Input data structure]

Address	Content	Remarks
Top address +0	(Function code) 232	
+2	(Completion code) – (Need not be set)	
+4	(Data length) – (Need not be set)	
+6	(Data number) 0	
+8	(Data attribute) M (M=Tool number)	M= 0 : Latest data counter = 1 : Sample data counter
+10	(Data area) – (Need not be set)	
+12		

[Completion codes]

0 : The store counter was read successfully.

3 : Incorrect data number, that is, a value other than 0 is specified.

4 : The fine torque sensing data in ’Data Attribute’ has wrong values, that is, a value other than 01.

6 : The fine torque sensing option has not been added on.

[Output data structure]

Address	Content	Value
Top address +0	(Function code) 232	
+2	(Completion code) ? (See the explanation above.)	
+4	(Data length) 4	
+6	(Data number) 0 (Entered data)	
+8	(Data attribute) M (Entered data)	
+10	Store counter value	Unsigned binary
+14		

(2) Stored torque data (latest data)

[Description]

This function reads the latest stored data among stored torque data.

[Input data structure]

Address	Content	Remarks
Top address +0	(Function code) 232	
+2	(Completion code) – (Need not be set)	
+4	(Data length) – (Need not be set)	
+6	(Data number) N (N=11 to (10+N))	N : Number of axis to be read + 10
+8	(Data attribute) M (M=Data type)	M =0 : Latest data =1 : Sample data
+10	(Data area) – (Need not be set)	

[Completion codes]

0: The stored torque data (latest data) was read successfully.

3: Incorrect data number, that is, a value other than 11 to (10+n) (n: number of spindles) is specified.

4: The fine torque sensing data in 'Data Attribute' has a wrong value, that is, a value other than 0 or 1.

6: The fine torque sensing option has not been added on.

[Output data structure]

Address	Field	
Top address +0	(Function code) 232	
+2	(Completion code) ? (See the explanation above.)	
+4	(Data length) L Entered data	
+6	(Data number) N (Entered data)	
+8	(Data attribute) M (Entered data)	Value
+10	Latest stored data	Unsigned binary
+14		

NOTE

1 When data has not been stored, data is not output, and processing ends successfully with L set to 0.

2 When sample data is selected by data attribute, the sample data corresponding to the latest stored data is output.

Example) When 10000 sample data items (data numbers 0 to 9999) and latest data items 5000 (data numbers 0 to 4999) are stored, data number 4999 in the latest data is output when data attribute M is set to "0", and data number 4999 in the sample data is output when data attribute M is set to "1".

3 When sample data is selected by data attribute, and there is no sample data corresponding to the latest stored data, data is not output, and processing ends successfully with L set to 0.

Example) When 5000 sample data items (data numbers 0 to 4999) and 10000 latest data items (data numbers 0 to 9999) are stored, data is not output, and processing ends successfully with L set to 0 when data attribute M is set to "1".

(3) Stored torque data (any data)

[Description]

This function reads the latest stored data among stored torque data.

[Input data structure]

Offset	Contents	Remarks
Top address +0	(Function code) 232	
+2	(Completion code) – (Need not be set)	
+4	(Data length) 6	
+6	(Data number) N (N=1 to n)	N: Number of axis to be read
+8	(Data attribute) M (M=Data type)	M =0 : Latest data =1 : Sample data
+10	Data number n	
+14	Number of data items I	
+16		

NOTE

The valid range of data number n is calculated as follows:

$$0 \leq n \leq (524288 \times \frac{1}{a} \times \frac{1}{b})-1$$

where,

a= { 1: Number of target axes 1
2: Number of target axes 2
4: Number of target axes 3 and 4 }

b= { 1: Sample data store function OFF
2: Sample data store function ON }

The valid range of number of data items I is calculated as follows:

$1 \leq I \leq 20$

[Completion codes]

0 : The stored torque data (any data) was read successfully.

2 : Incorrect data length, that is, a value other than 6 is specified.

3 : Incorrect data number, that is, a value other than 11 to (10+n) (n: number of spindles) is specified.

4 : The fine torque sensing data in 'Data Attribute' has a wrong value, that is, a value other than 0 or 1.

5 : Incorrect data area is specified. See Note for details of value ranges.

6 : The fine torque sensing option has not been added on.

[Output data structure]

		Value
Top address +0	(Function code) 232	
+2	(Completion code) ? (See the above explanation.)	
+4	(Data length) L (L=6 + number of data items I × 2)	
+6	(Data number) N (Entered data)	
+8	(Data attribute) M (Entered data)	
+10	Data number n (Entered data)	Signed binary
+14	Number of data items I (Entered data)	
+16	Distribution of target axes	
+18	Number n data	
+20	Number n+1 data	
+22	Number n+2 data	
+24	:	
	:	
	:	
	Number n+I–1 data	

NOTE

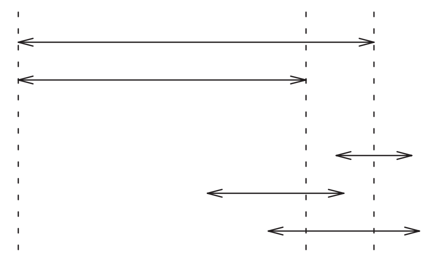
1 When the number of actually stored data items is exceeded even though data number n is in the valid range, data is not output and processing ends successfully by number of data items I set to 0.

Example) When the number of target axes is 2, and the sample data store function is enabled (parameter No.6350#2=1), data numbers 0 to 13107 are valid. However, if an attempt is made to read (example (1) in figure below) data from data number n = 131020 when the number of actually stored data items is 131000 (data numbers 0 to 130999), data is not output, and the number of data items I becomes 0.

2 When data number n is within the number of actually stored data items, and (n+I – 1) exceeds the number of actually stored data items, data of the stored data items is output, and processing ends successfully. In this case, number of data items I is updated to the number of data items that was output.

Example) If an attempt is made to read (example (2) in figure below) number of data items I (120) from data number 130900 under the same conditions as in the example above, the data of data numbers 130900 to 130999 is output, and number of data items I becomes 100.

Also, if an attempt is made to read (example (3) in figure below) number of data items I (120) from data number 130999 under the same conditions as in the example above, the data of data numbers 130990 to 130999 is output, and number of data items becomes 10.



B.4.75 Specifying the Number of the Program for I/O Link

[Explanation of data]

Specify the number of the program to be input/output using the data input/output function with I/O Link.

[Input data structure]

Address	Contents	Value
Top address +0	(Function code) 194	
+2	(Completion code) – (Need not be set.)	
+4	(Data length) 2	
+6	(Data number) 0	
+8	(Data attribute) 0	
+10	Program number	Signed binary format
+12		

[Completion codes]

0 : The specification of the program number terminated normally.

5 : Invalid data was specified for the program number, i.e., the data falls outside the range of 1 to 9999 or is not –9999.

[Output data structure]

Offset	Field	Value
Top address +0	(Function code) 194	
+2	(Completion code) ? (See the explanation of completion codes, above.)	
+4	(Data length) 2 (Data at input time)	
+6	(Data number) 0 (Data at input time)	
+8	(Data attribute) 0 (Data at input time)	Value
+10	Program number (Data at input time)	Signed binary format
+12		

CAUTION
For details of this function, see the section on data input/output functions using I/O Link in the “CNC Connection Manual (Functions).”

C WINDOW FUNCTION DESCRIPTION (PMC–NB/NB2/NB6)

C.1 FUNCTION

This window function is a functional instruction by which the data on the CNC is read or is written.

Option (FS15B : A02B–0162–J917, FS15*i* : A02B–0261–J950) of NC window is necessary.

C.2 LOW–SPEED RESPONSE AND HIGH–SPEED RESPONSE OF WINDOW FUNCTION

In the way to process, there are window function high speed and one processed at low speed.

In case of a low–speed response, The data is read or written by the control between CNC and PMC.

Therefore, it is necessary to ACT=1 of the window instrucion must be held until the transfer completion information (W1) becomes 1 (interlock).

In a high–speed response, it is not necessity for take the interlock because the data is directly read.

To read tool offset data, tool life management data, and the processing time, the FS15B requires the installation of the NC window B option (A02B–0162–J984), in addition to the NC window option.

CAUTION

1 The window instruction of a low–speed response is controlled exclusively with the other window instructions of low–speed response.
Therefore, when the data is read or written continuously, it is necessary to clear ACT of the functional instruction once when the completion information (W1) become 1.
It does not work about ACT=1 of the other window instructions of low–speed response such as W1=1 and ACT=1 of the window instruction of a low–speed response.
The window instruction of a high–speed response is not exclusively controlled like a low–speed response. Therefore, when the data is read or written continuously, yow need not make ACT=0.
The scan number of times to complete the processing is summarized on the following table.

TYPE	SCAN TIMES UNTIL PROCESSING ENDS
LOW	TWO SCAN TIMES OR MORE (This depends on the state of CNC)
HIGH	1SCAN TIME

(Only FS15B)

2 There is a version which does not support in the reading or writing of the window data by a new format.

ROM VERSION		CONTENT
4047	A – E	It does not support a new format. Please use #4 of NC parameter 7401 as 0.
	F –	It supports a new form. When the window function of a new format is used, please set #4 of NC parameter 7401 as 1.
4078	A –	

Function that is effected by #4 of NC parameter 7401.

FUNCTION	CONTENT	
	7401#4	
Tool life management data	0	The data of tool life management for 128 sets of tools can be read and written.
	1	The data of tool life management for 512 sets of tools can be read and written.
Tool offset data according to the specified tool number	0	This function can not be used.
	1	The tool offset data can be read and written.

3 Functions except the above–mentioned are not related to #4 of NC parameter 7401.
If there is no option of the corresponding function, window instructions can not be used.

C.2.1 Functional Instruction WINDR

CTL0	0	Function code
	2	Completion code
	4	Data length
CTL1	6	Data number
CTL2	8	Axis specification
DATA	10	Data area

* Data length Depends on the function

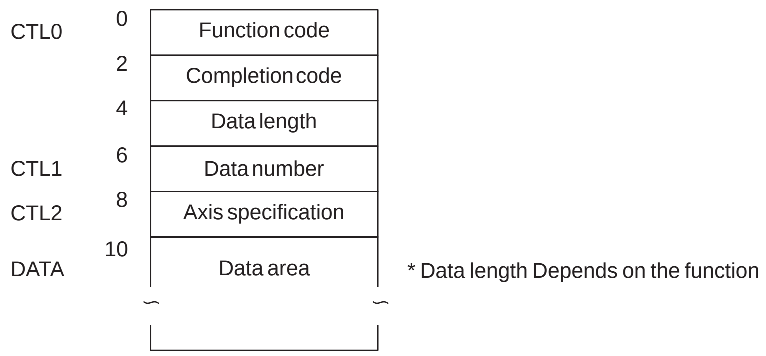
part1

Data type	Type of processing	Type of control data			Data length
		CTL0	CTL1	CTL2	
Tool offset data	(low)	13	Offset number	Offset format	4 byte
Work origin offset	(high)	15	0	Axis number	4 byte
Parameter data Setting data	(low)	17	Parameter number	Axis number	4 byte
Custom macro variables	(low)	21	Custom macro number	0	6 byte
CNC alarm state	(low)	23	0	0	2 byte
Current program number	(low)	24	0	0	6 byte
Current sequence number	(low)	25	0	0	6 byte
Actual velocity for controlled axes	(low)	26	0	0	4 byte
Absolute position on controlled axes	(high)	27	0	Axis number	4 byte
Machine position on controlled axes	(high)	28	0	Axis number	4 byte
Skip operation stop position on controlled axes	(low)	29	0	Axis number	4 byte
Servo delay amount on controlled axes	(high)	30	0	Axis number	4 byte
Acceleration/deceleration delay amount on controlled axes	(high)	31	0	Axis number	4 byte
Modal data (G function) (other than G function)	(low)	32	Data type	Specified block	2 byte 6 byte
Diagnosis data	(low)	33	Diagnosis number	0	2 byte
Feed motor load current value General–purpose analog input	(high) (high)	34 34	200 0	Axis number Axis number	2 byte 2 byte

part2

Data type	Type of processing	Type of control data			Data length
		CTL0	CTL1	CTL2	
Tool life management data					
Tool group No.	(low)	38	0	Tool No.	4 byte
Number of tool groups	(low)	39	Tool group No.	0	4 byte
Number of tools	(low)	40	Tool group No.	0	4 byte
Tool life	(low)	41	Tool group No.	0	4 byte
Tool life counter	(low)	42	Tool group No.	0	4 byte
Tool life counter type	(low)	160	Tool group No.	0	4 byte
Tool length compensation No.1	(low)	43	Tool group No.	Tool No.	4 byte
Tool length compensation No.2	(low)	44	Tool group No.	Tool order number	4 byte
Cutter compensation No.1	(low)	45	Tool group No.	Tool No.	4 byte
Cutter compensation No.2	(low)	46	Tool group No.	Tool order number	4 byte
Tool information 1	(low)	47	Tool group No.	Tool No.	4 byte
Tool information 2	(low)	48	Tool group No.	Tool order number	4 byte
Tool No.	(low)	49	Tool group No.	Tool order number	4 byte
Clock data	(low)	151	Data format	0	6 byte
Relative position of controlled axes	(high)	74	0	Axis number	4 byte
Remaining travel of controlled axes	(high)	75	0	Axis number	4 byte
Estimate disturbance torque data of a digital	(high)	211	0	Axis number	2 byte
Estimate disturbance torque data of a serial spindle	(high)	211	1	Axis number	2 byte
Machining time	(low)	178	Program number	1	6 byte
Load information of the spindle motor	(high)	153	0	Axis number	2 byte
Tool offset data according to the specified tool number	(low)	213	Data format	Tool number	4 byte
Tool life management data					
Tool group number (supporting 8–digit tool numbers)	(low)	200	0	Tool number (4 bytes)	4 bytes
Tool length compensation number 1 (supporting 8–digit tool numbers)	(low)	227	Tool group number	Tool number (4 bytes)	4 bytes
Cutter compensation number 1 (supporting 8–digit tool numbers)	(low)	228	Tool group number	Tool number (4 bytes)	4 bytes
Tool information 1 (supporting 8–digit tool numbers)	(low)	201	Tool group number	Tool number (4 bytes)	4 bytes
Real parameter data	(low)	321	Parameter number (4 bytes)	Axis number	4 bytes

C.2.2
Functional Instruction WINDW



part1

Data type	Type of processing	Type of control data			Data length
		CTL0	CTL1	CTL2	
Tool offset data	(low)	14	Offset number	Offset format	4 byte
Parameter data Setting data	(low)	18	Parameter number	Axis number	4 byte
Custom macro variables	(low)	22	Custom macro number	0	6 byte
Data on the program check screen Spindle tool number Number of the tool to be used next	 (low) (low)	 150 150	 Data type 201	 0 0	 2 byte 2 byte
Torque limit override	(low)	152	0	Axis number	2 byte
Tool life management data Number of tool groups Tool life Tool life counter Tool life counter type Tool length compensation No.1 Tool length compensation No.2 Cutter compensation No.1 Cutter compensation No.2 Tool information 1 Tool information 2 Tool No.	 (low) (low) (low) (low) (low) (low) (low) (low) (low) (low) (low)	 163 164 165 166 167 168 169 170 171 172 173	 0 Tool group No. Tool group No. Tool group No. Tool group No. Tool group No. Tool group No. Tool group No. Tool group No. Tool group No. Tool group No.	 Tool No. 0 0 0 Tool No. Tool order number Tool No. Tool order number Tool No. Tool order number Tool order number	 4 byte 4 byte 4 byte 4 byte 4 byte 4 byte 4 byte 4 byte 4 byte 4 byte 4 byte
Tool offset data according to the specified tool number	(low)	214	Data format	Tool number	4 byte
Superposition move command (for three axes) Superposition move command (for four axes)	(high) (high)	215 215	0 Axis specification mode	0 Axis number	6 byte 8 byte
Feedrate	(high)	216	0	0	6 byte

part2

Data type	Type of processing	Type of control data			Data length
		CTL0	CTL1	CTL2	
Tool life management data					
Tool group number (supporting 8–digit tool numbers)	(low)	202	0	Tool number (4 bytes)	6 bytes
Tool length compensation number 1 (supporting 8–digit tool numbers)	(low)	229	Tool group number	Tool number (4 bytes)	4 bytes
Cutter compensation number 1 (supporting 8–digit tool numbers)	(low)	230	Tool group number	Tool number (4 bytes)	4 bytes
Tool information 1 (supporting 8–digit tool numbers)	(low)	231	Tool group number	Tool number (4 bytes)	4 bytes
Real parameter data	(low)	323	Parameter number (4 bytes)	Axis number	4 bytes

C.3 FORMAT AND DETAILS OF THE CONTROL DATA OF THE WINDR FUNCTIONAL INSTRUCTION

(1) See the description of the window function. The data item marked with a dash (–) in the description of the data structure need not be entered.

(2) The length of all data blocks and data items is represented in bytes.

(3) The read data becomes valid only when the instruction terminates normally.

Completion code	Meaning
–10	The window instruction is being processed. Hold ACT until W1 is set to 1.
0	The instruction terminated normally.
1	An error occurred. The corresponding function number is not found.
2	An error occurred. Possible causes include the following: Wrong data is found in the CTL area. The NC does not have the corresponding function.
3	An error occurred. The specified axis is not provided.
5	An error occurred. It is a mistake of data form. Though the function supports only a new format, a old format is specified.

C.3.1 Reading a Tool Offset (Low–speed Response)

[Description]

The tool offset value (tool compensation) is read from the CNC.

Set data

0	Function code 13
2	Completion code —
4	Data length —
6	Offset number
8	Offset format
10	Data area (4 byte)
14	

Read data

0	Function code 13
2	Completion code
4	Data length 4
6	Offset number
8	Offset format
10	Tool offset value (4 byte)
14	

(Note 1) Offset format

M series (machining center system)

Data type	Format	Offset number (CTL+2, 3)
Tool compensation A Compensation	 1	 Offset number
Tool compensation B Geometry compensation Wear compensation	 1 1	 Offset number Offset number +1000
Tool compensation C Tool length Geometry compensation Wear compensation Cutter Geometry compensation Wear compensation	 1 1 2 2	 Offset number Offset number +1000 Offset number Offset number +1000

T series (lathe system)

Data type	Format	Offset number (CTL+2, 3)
Tool compensation A Compensation along the X–axis Compensation along the Z–axis Tool–tip radius compensation Compensation along the Y–axis Compensation related to the position of the virtual tool	 1 2 3 4 5	 Offset number Offset number Offset number Offset number Offset number Offset number
Tool compensation B Geometry compensation Compensation along the X–axis Compensation along the Z–axis Tool–tip radius compensation Compensation along the Y–axis Wear compensation Compensation along the X–axis Compensation along the Z–axis Tool–tip radius compensation Compensation along the Y–axis Compensation related to the position of the virtual tool	 1 2 3 4 1 2 3 4 5	 Offset number Offset number Offset number Offset number Offset number +1000 Offset number +1000 Offset number +1000 Offset number +1000 Offset number

C.3.2 Reading a Workpiece Origin Offset Value

[Description]

The offset from the workpiece reference point of the current coordinate system (including a shared offset) of the CNC is read.

The offset from the workpiece reference point for each axis can be read individually. The offset from the workpiece reference point for an additional axis can be read only when the additional axis is provided.

Set data

Offset	Field
0	Function code 15
2	Completion code —
4	Data length —
6	Data number 0
8	Axis number
10	Data area (4 byte)
14	

Read data

Offset	Field
0	Function code 15
2	Completion code
4	Data length 4
6	Data number
8	Axis number
10	Workpiece origin offset value (4 byte)
14	

C.3.3 Reading a Parameter (Setting Data) (Low–speed Response)

[Description]

A parameter of the CNC is read.

Set data

Offset	
0	Function code 17
2	Completion code —
4	Data length —
6	Parameter number
8	Data attribute
10	Data area (4 byte)
14	

Data attribute:
0 : No axis
1 to n : A specified axis

Read data

Offset	
0	Function code 17
2	Completion code
4	Data length 4
6	Parameter number
8	Data attribute
10	Parameter data (4 byte)
14	

(Note) Format of parameter data
If the data is one byte long, it is set in the DATA+0 area.

Offset	
10	DATA+0
11	DATA+1
12	DATA+2
13	DATA+3
14	

C.3.4 Reading a Custom Macro Variable (Low–speed Response)

[Description]

A custom macro variable is read from the CNC.

> **CAUTION**
> The position of the decimal point must be specified beforehand.

Set data

Offset	Field
0	Function code 21
2	Completion code —
4	Data length —
6	Custom macro variable number
8	Data attribute 0
10	Data area (4 byte)
14	Position of decimal point (2 byte)
16	

Read data

Offset	Field
0	Function code 21
2	Completion code
4	Data length 6
6	Custom macro variable number
8	Data attribute
10	Custom macro variable value (4 byte)
14	Position of decimal point (2 byte)
16	

> **CAUTION**
> In the case of reading a Custom Macro Variable of upper 100000.
> Please input “10” to “Data attribute”, and input last four digits of variable number to “Custom macro variable number”.

Examples

The relationship between the read value and the stored variable is:

(Read value) =
(Custom macro variable in the NC) $\times\ 10^{\text{(Position of decimal point)}}$

Read value	Custom macro variable in the NC	Position of decimal point
1	1. 234	0
12		1
123		2
1234		3
12340		4

C.3.5 Reading the CNC Alarm Status (Low–speed Response)

[Description]

If the CNC is in the alarm state, the details of the alarm are read.

Set data

Offset	Set data
0	Function code 23
2	Completion code —
4	Data length —
6	Data number 0
8	Data attribute 0
10	Data area (2 byte)
12	

Read data

Offset	Read data
0	Function code 23
2	Completion code
4	Data length 2
6	Data number
8	Data attribute
10	ALARM 1
11	ALARM 2
12	

The following alarm states can be read:

ALARM 1 Data format

7	6	5	4	3	2	1	0

#0 : Background P/S alarm
#1 : Foreground P/S alarm
#2 : Overheat
#3 : Sub–CPU error
#4 : Excessive synchronization error
#5 : Parameter enable switch enabled
#6 : Overtravel
#7 : PC error

ALARM 2 Data format

7	6	5	4	3	2	1	0

#0 : External alarm message
#1 : Not defined
#2 : P/S alarm preventing the processing from continuing (serious alarm)
#3 : Not defined
#4 : Servo alarm
#5 : I/O error
#6 : Parameter input requiring the power to be turned off
#7 : System error

C.3.6 Reading the Current Program Number (Low–speed Response)

[Description]

The number of a running machining program is read from the CNC.

Set data

Offset	
0	Function code 24
2	Completion code —
4	Data length —
6	Data number 0
8	Data attribute 0
10	Data area (6 byte)
16	

Read data

Offset	
0	Function code 24
2	Completion code
4	Data length 6
6	Data number
8	Data attribute
10	Program number
14	FLAG 1
15	FLAG 2
16	

FLAG 1 Data format
7 6 5 4 3 2 1 0

#0 to 3 : Number of decimal places
#4 : Whether a decimal point is provided
(0: Not provided, 1: Provided)
#5 : Positive or negative
(0: Positive, 1: Negative)
#6 : Whether only one digit is provided
(0: Not only one, 1: Only one)
#7 : Not defined

FLAG 2 Data format
7 6 5 4 3 2 1 0

#3 : Whether at least one address in the range of D to the second miscellaneous function is specified after the NC is reset
(0: Not specified, 1: Specified)
#7 : Whether the read data is specified in the current block of the part program
(0: Not specified, 1: Specified)
Other bits are not defined.

C.3.7 Reading the Current Sequence Number (Low–speed Response)

[Description]

The sequence number of the running machining program is read from the CNC. If the blocks of the running machining program have no sequence numbers, the sequence number of the block most recently executed is read.

Set data

Offset	
0	Function code 25
2	Completion code —
4	Data length —
6	Data number 0
8	Data attribute 0
10	Data area (6 byte)
16	

Read data

Offset	
0	Function code 25
2	Completion code
4	Data length 6
6	Data number
8	Data attribute
10	Sequence number (4 byte)
14	FLAG 1
15	FLAG 2
16	

FLAG 1 Data format

7 6 5 4 3 2 1 0

#0 to 3 : Number of decimal places
#4 : Whether the decimal point is provided (0: Not provided, 1: Provided)
#5 : Positive or negative (0: Positive, 1: Negative)
#6 : Whether only one digit is provided (0: Not only one, 1: Only one)
#7 : Not defined

FLAG 2 Data format

7 6 5 4 3 2 1 0

#3 : Whether at least one address in the range of D to the second miscellaneous function is specified after the NC is reset (0: Not specified, 1: Specified)
#7 : Whether the read data is specified in the current block of the part program (0: Not specified, 1: Specified)Other bits are not defined.

C.3.8 Reading the Actual Velocity of Controlled Axes (Low–speed Response)

[Description]

The actual speed of the feed axes controlled by the CNC is read.

The composite speed of the controlled axes is read. If the X–, Y–, and Z–axes, the basic three axes, are controlled as feed axes, the composite speed of the three axes is read.

Offset	Set data	Offset	Read data
0	Function code 26	0	Function code 26
2	Completion code —	2	Completion code
4	Data length —	4	Data length 4
6	Data number 0	6	Data number
8	Data attribute 0	8	Data attribute
10	Data area (4 byte)	10	Feedrate (4 byte)
14		14	

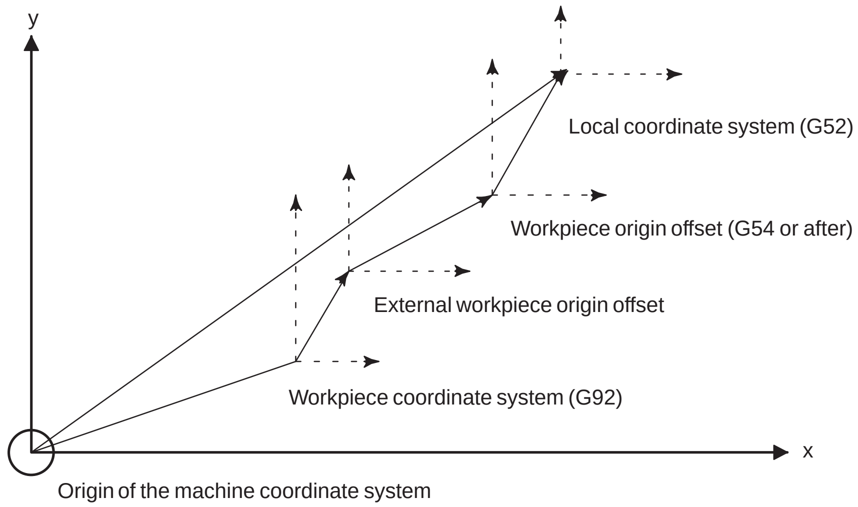
C.3.9 Reading the Absolute Position on a Controlled Axis

[Description]

The absolute position (absolute coordinates) on a feed axis controlled by the CNC is read.

Offset	Set data	Offset	Read data
0	Function code 27	0	Function code 27
2	Completion code —	2	Completion code
4	Data length —	4	Data length 4
6	Data number 0	6	Data number
8	Axis number 0	8	Axis number
10	Data area (4 byte)	10	Absolute position (4 byte)
14		14	

Data specification

Kind of data	Data specification
Current position	1) Indicates the current position in the workpiece coordinate system. The current position is calculated by the following simple expression. The read current position is stored in the 4–byte area of DATA+0 to DATA+3. Current position = machine coordinate value – workpiece offset value The workpiece offset value can be obtained by summing up the folloing offset. (1) External workpiece origin offset (2) Workpiece origin offset (G54 to G59,G54.1Pp) (3) Workpiece coordinate system (G92) (4) Local coordinate system (G52)  2) The unit of the read value is determined as follows: (1) For the machining center system or when the radius is specified for the axis of the lathe systm. The data shows double of the present position with the least input increment as a unit. (2) When the diameter is specified for the lathe system The data shows the present position with the least input increment. (3) When the input unit is multiplied by 10 The data shows twenty–times the present position (radius programming) or ten–times the present position (diameter programming) with the last command increment as a unit. 3) The present position of a moving axis can be read whenever the function instruction is executed.

C.3.10 Reading the Machine Position (Machine Coordinates) of Controlled Axes

[Description]

The machine position (machine coordinates) on a feed axis controlled by the CNC is read.

Set data

Offset	Field
0	Function code 28
2	Completion code —
4	Data length —
6	Data number 0
8	Data attribute
10	Data area (4 byte)
14	

Read data

Offset	Field
0	Function code 28
2	Completion code
4	Data length 4
6	Data number
8	Data attribute
10	Feedrate (4 byte)
14	

1) The unit of the read value is determined as follows:

   (1) For the machining center system or when the radius is specified for the axis of the lathe systm.
   The data shows double of the present position with the least input increment as a unit.

   (2) When the diameter is specified for the lathe system

   The data shows the present position with the least input increment.

   (3) When the input unit is multiplied by 10

   The data shows twenty–times the present position (radius programming) or ten–times the present position (diameter programming) with the last command increment as a unit.

2) The present position of a moving axis can be read whenever the function instruction is executed.

C.3.11 Reading a Skip Position (Stop Position of Skip Operation (G31)) of Controlled Axes (Low–speed Response)

[Description]

The absolute coordinates of the skip position specified in the CNC are read.

Offset	Set data	Offset	Read data
0	Function code 29	0	Function code 29
2	Completion code —	2	Completion code
4	Data length —	4	Data length 4
6	Data number 0	6	Data number
8	Axis number	8	Axis number
10	Data area (4 byte)	10	Skip position (4 byte)
14		14	

1) The unit of the read value is determined as follows:

   (1) For the machining center system or when the radius is specified for the axis of the lathe systm.
   The data shows double of the present position with the least input increment as a unit.

   (2) When the diameter is specified for the lathe system

   The data shows the present position with the least input increment.

   (3) When the input unit is multiplied by 10

   The data shows twenty–times the present position (radius programming) or ten–times the present position (diameter programming) with the last command increment as a unit.

2) Once the skip signal has been input to the NC, movement along the relevant axis is stopped then, after the elapse of the servo delay, the absolute position can be read.

C.3.12 Reading the Servo Delay for Controlled Axes

[Description]

A servo delay, which is the difference between the specified position on a controlled axis and the actual servo position, is read from the CNC.

Offset	Set data	Read data
0	Function code 30	Function code 30
2	Completion code —	Completion code
4	Data length —	Data length 4
6	Data number 0	Data number
8	Data attribute	Data attribute
10	Data area (4 byte)	Servo delay (4 byte)
14		

C.3.13 Reading the Acceleration/ Deceleration Delay on Controlled Axes

[Description]

An acceleration/deceleration delay, which is the difference between the programmed position on a controlled axis and the actual position after the acceleration or deceleration, is read from the CNC.

Offset	Set data	Read data
0	Function code 31	Function code 31
2	Completion code —	Completion code
4	Data length —	Data length 4
6	Data number 0	Data number
8	Axis number	Axis number
10	Data area (4 byte)	Acceleration/ deceleration delay (4 byte)
14		

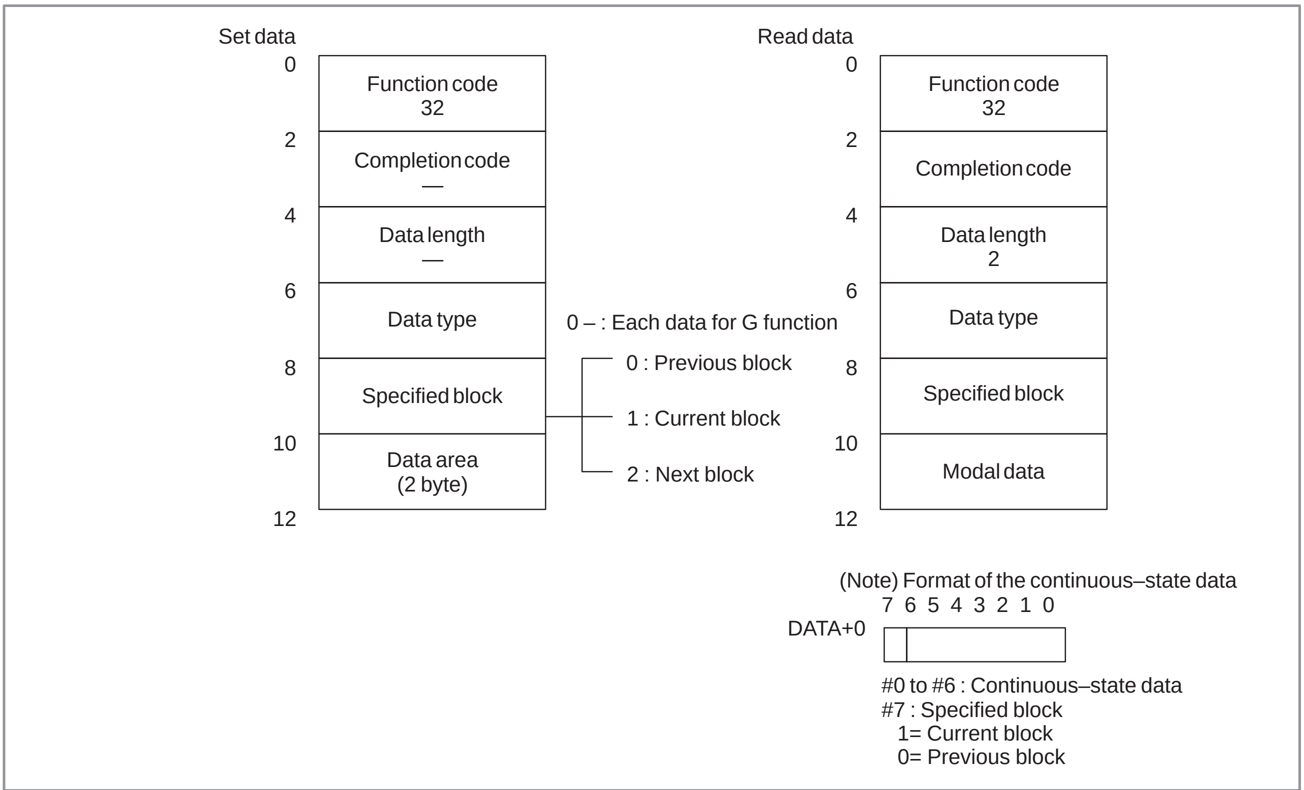
C.3.14 Reading Modal Data (Low–speed Response)

[Description]

The continuous–state data is read from the CNC.

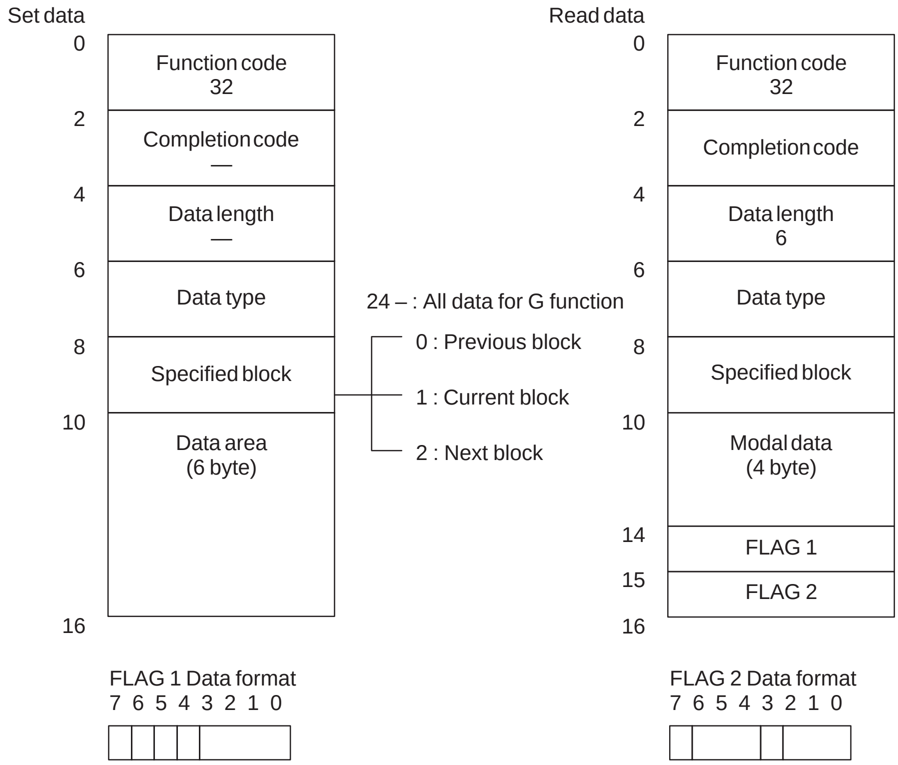
The continuous–state data can be broadly classified into two types: Data of the preparatory function and data of other functions. When CTL2 (specified block) is set to 0, the continuous–state data of the previous block is read. When CTL2 is set to 2, the continuous–state data of the next block is read.

(1) Continuous–state data of the preparatory function



(2) Continuous–state data of a function other than the preparatory function

The following eleven data items of an NC part program can be read: addresses D, E, H, L, M, N, O, S, T, and F, and second miscellaneous function.



FLAG 1:

#0 – 3 : Number of decimal places
#4 : Whether the decimal point is provided
(0: Not provided, 1: Provided)
#5 : Positive or negative
(0: Positive, 1: Negative)
#6 : Whether only one digit is provided
(0: Not only one, 1: Only one)
#7 : Not defined

FLAG 2:

#3 : Whether at least one address in the range of D to the second miscellaneous function is specified after the NC is reset
(0: Not specified, 1: Specified)
#7 : Whether the read data is specified in the current block of the part program
(0: Not specified, 1: Specified)
Other bits are not defined.

(3) Data specification

Kind of data	Data specification
Modal data	1) Modal data of G function

The relationship between the numbers specified in the CTL1 (kinds of data), modal data codes is shown below. Into CTL2 (the specified block), specify 0 (previous data), 1 (present date), or 2 (next data) in accordance with the necessary modal data.
The G code for the lathe system is expressed with the G code system B. Refer to the table indicating the G function system. For example, the G32 of the G code system A corresponds to the G33 of the G code system B. As a result, the code fetched in the DATA + 0 is 4.

Specified number in CTL1 (kinds of data)	G code for machining center system	G code for lathe system (G code system B)	Code fetched in DATA + 0
00	G00 G01 G02 G03 G33 — — —	G00 G01 G02 G03 G33 G77 G78 G79	0 1 2 3 4 8 9 10
01	G17 G18 G19	G97 G96 —	0 1 2
02	G90 G91	G90 G91	1 0
03	G22 G23	G22 G23	0 1
04	G93 G94 G95	— G94 G95	2 0 1
05	G20 G21	G20 G21	1 0
06	G40 G41 G42	G40 G41 G42	0 1 2
07	G43 G44 G49	— — —	1 2 0
08	G80 G81 G82 G83 G84 G85 G86 G87 G88 G89 G73 G74 G76	G80 G81 G82 G83 G84 G85 G86 G87 G88 G89 G83.1 G84.1 G86.1	0 1 2 3 4 5 6 7 8 9 10 11 12
09	G98 G99	G98 G99	0 1
10	G50 G51	— —	0 1

Kind of data	Data specification			
Modal data	**Specified number in CTL1 (kinds of data)**	**G code for machining center system**	**G code for lathe system (G code system B)**	**Code fetched in DATA + 0**
	11	G66 G67 G66.1	G66 G66 G66.1	1 0 2
	12	G96 G97	G68 G69	1 0
	13	G54 G55 G56 G57 G58 G59	G54 G55 G56 G57 G58 G59	0 1 2 3 4 5
	14	G61 G62 G63 G64	G61 — — G64	1 2 3 0
	15	G69 G68 —	G17 G18 G19	0 1 2
	16	G15 G16	— —	0 1
	17	G50.1 G51.1	G50.1 G51.1	0 1

Kind of data	Data specification
Modal data	(see table below)

Table — 1 of G code system for a lathe system

G code system *1) A	B	C	Function
G00	G00	G00	Positioning
G01	G01	G01	Linear interpolation
G02	G02	G02	Circular interpolation CW
G03	G03	G03	Circular interpolation CCW
G04	G04	G04	Dwell
G07	G07	G07	Hypothetical axis interpolation
G09	G09	G09	Exact stop
G10	G10	G10	Data setting
G10.1	G10.1	G10.1	PC data setting
G11	G11	G11	Data setting mode cancel
G17	G17	G17	XpYp plane selection Xp: X axis or its parallel axis
G18	G18	G18	ZpXp plane selection Yp: Y axix or its parallel axis
G19	G19	G19	YpZp plane selection Zp: Z axis or its parallel axis
G20	G20	G70	Inch input
G21	G21	G71	Metric input
G22	G22	G22	Stored stroke check on
G23	G23	G23	Stored stroke check off
G27	G27	G27	Reference point return check
G28	G28	G28	Reference point return
G29	G29	G29	Return from reference point
G30	G30	G30	Return to 2nd, 3rd, 4th reference point
G31	G31	G31	Skip function
G32	G32	G32	Thread cutting
G34	G34	G34	Variable lead thread cutting
G35	G35	G35	Circular thread cutting CW
G36	G36	G36	Circular thread cutting CCW or automatic tool compensation (X axis)
G37	G37	G37	Automatic tool compensation #1 or automatic tool compensation (Z axis)
G37.1	G37.1	G37.1	Automatic tool compensation #1
G37.2	G37.2	G37.2	Automatic tool compensation #2
G37.3	G37.3	G37.3	Automatic tool compensation #3
G40	G40	G40	Tool nose radius compensation cancel
G41	G41	G41	Tool nose radius compensation left
G42	G42	G42	Tool nose radius compensation right
G50	G92	G92	Work coordinates change/maximum spindle spped setting
G50.1	G50.1	G50.1	Programmable mirror image cancel
G51.1	G51.1	G51.1	Programmable mirror image
G52	G52	G52	Local coordinate system setting
G53	G53	G53	Machine coordinate system selection
G54	G54	G54	Work coordinate system 1 selection
G55	G55	G55	Work coordinate system 2 selection
G56	G56	G56	Work coordinate system 3 selection
G57	G57	G57	Work coordinate system 4 selection
G58	G58	G58	Work coordinate system 5 selection
G59	G59	G59	Work coordinate system 6 selection
G61	G61	G61	Exact stop mode
G62	G62	G62	Automatic corner override
G64	G64	G64	Cutting mode
G65	G65	G65	Macro call
G66	G66	G66	Macro modal call A
G66.1	G66.1	G66.1	Macro modal call B
G67	G67	G67	Macro modal call A/B cancel
G68	G68	G68	Mirror image for double currets on
G69	G69	G69	Mirror image for double currets cancel
G70	G70	G72	Finishing cycle
G71	G71	G73	Stock removal in turning

Kind of data	Data specification
Modal data	Table — 2 of G code system for a lathe system (see below)

Table — 2 of G code system for a lathe system

G code system *1)			Function
A	B	C	
G72	G72	G74	Stock removal in facing
G73	G73	G75	Pattern repeating
G74	G74	G76	Peck drilling Z axis
G75	G75	G77	Grooving in X axis
G76	G76	G78	Threading cycle
G80	G80	G80	Canned cycle for drilling cancel
G81	G81	G81	Drilling cycle, spot boring
G82	G82	G82	Drilling cycle, counter boring
G83	G83	G83	Peck drilling cycle
G83.1	G83.1	G83.1	Peck drilling cycle
G84	G84	G84	Tapping cycle
G84.1	G84.1	G84.1	Counter tapping cycle
G85	G85	G85	Boring cycle
G86	G86	G86	Boring cycle
G86.1	G86.1	G86.1	Fine boring cycle
G87	G87	G87	Back boring cycle
G88	G88	G88	Boring cycle
G89	G89	G89	Boring cycle
G90	G77	G20	Outting cycle A
G92	G78	G21	Thread cutting cycle
G94	G79	G24	Outting cycle B
G96	G96	G96	Constant surface speed control
G97	G97	G97	Constant surface speed control
G98	G94	G94	Feed per minute
G99	G95	G95	Feed per revolution
—	G90	G90	Absolute command
—	G91	G91	Incremental command
—	G98	G98	Canned cycle initial level return
—	G99	G99	Canned cycle R point level return

*1) G code system A/B can be selected by parameter setting (basic function).
Gcode sytem C is optinal function. However, when this option is selected, G code system A/B is selectable.

2) Modal data other than the G function

Modal data other than the G function (address in the part program)	CTL1 (kinds of data)		Field from which to fetch data
	For machining system	For turning system	
D	24	—	DATA+0 to DATA+5
E	25	24	
H	26	25	
L	27	26	
M	28	27	
N	29	28	
O	30	29	
S	31	30	
T	32	31	
F	33	32	
Second auxiliary function	34	33	

C.3.15 Reading Diagnosis Data (Low–speed Response)

[Description]

The data on the diagnostic data screen of the CNC is read.

Set data

Offset	
0	Function code 33
2	Completion code —
4	Data length —
6	Diagnosis number
8	Data attribute 0
10	Data area (2 byte)
12	

Read data

Offset	
0	Function code 33
2	Completion code
4	Data length 2
6	Diagnosis number
8	Data attribute 0
10	Diagnosis data (2 byte)
12	

NOTE

1 The valid range of diagnosis numbers is 0 to 103 and 200 to 303. (FS15B)
For the FS15*i*, the valid range of numbers is 1000 and above. (FS15*i*A)

2 Only integer values can be read as diagnosis data. (FS15*i*A)

3 For the FS15*i*, axis data can be read by specifying an axis number for the data attribute.

C.3.16 Reading A/D Conversion Data for the Feed Motor

[Description]

1. The load current for an axis controlled by the CNC is converted to adigital value and the digital value is read.
2. The analog data input to the CNC is converted to a digital value by the A/D converter and the digital value is read.

Set data

Offset	Set data
0	Function code 34
2	Completion code —
4	Data length —
6	Data number
8	Axis number
10	Data area (2 byte)
12	

Read data

Offset	Read data
0	Function code 34
2	Completion code
4	Data length 2
6	Data number
8	Axis number
10	A/D conversion data (2 byte)
12	

An analog voltage ranging from –10V to +10V is input to the A/D converter of the NC. The voltage is converted to a digital value ranging form 0 to +255 and transferred by the window function to the PMC. This value is called the A/D conversion data.

The digital value is proportional to the analog voltage: 0 corresponds to –10V, 128 corresponds to 0V, and 255 corresponds to +10V.

Type of analog voltage input	Data number	Axis number
General–purpose analog input	0	1 2 3 4 5 6
Analog input of a voltage caluculated from the load current for the axis controlled by the NC (AC servo motor only)	200	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Method of calculation of the load current of controlled axis from the read A/D conversion data is as follows.

a) In the case of peak current [Ao–p] of load current is calculated.

$$\text{LOAD CURRENT[Ao–p]} = \frac{\text{(READ DATA)} - 128}{\text{(COEFFICIENT)}} \ \text{[Ao–p]}$$

b) In the case of ratings currents [Arms] of load current are calculated.

$$\text{LOAD CURRENT[Arms]} = \frac{\text{(READ DATA)} - 128}{\text{(COEFFICIENT)} \times \sqrt{2}} \ \text{[Arms]}$$

c) In the case of percent (rate) is calculated.

$$\text{rate of load[\%]} = \frac{\text{LOAD CURRENT[Ao–p]}}{\text{PEAK CURRENT OF SERVO MOTOR}} \times 100\text{[\%]}$$

COEFFICIENT:
It decides depending on the capacity of the amplifier to be used.

PEAK CURRENT OF SERVO MOTOR :
It dicides with the servo motor.

Examples

When the AC motor model ”30s” is used and the read A/D conversion data is 150, method of calculating each load current.

The following is understood from manual of the servo.

AC motor model	Ratings currents(Arms)
30S	16

Moreover, the amplifier of 80A is used for the motor of 30S.

The coefficient is calculated.

The coefficient is a value by which the peak current of amplifier is converted by 128.

$$\text{COEFFICIENT} = \frac{128}{\text{PEAK CURRENT VALUE OF AMPLIFIER}} = \frac{128}{80} = 1.6$$

The peak current of the servo motor is calculated.

$$\begin{aligned}\text{PEAK CURRENT[Ao–p]} &= \text{(ratings currents)} \times \sqrt{2}\\ &= 16 \times \sqrt{2} = 22.62742\\ &\doteq 23 \ \text{[Ao–p]}\end{aligned}$$

Since the rade A/D conversion data is 150, the peak current, the ratings currents and the rate of the load can be calculated.

a) Peak current[Ao–p] of load current

$$\begin{aligned}\text{LOAD CURRENT[Ao–p]} &= \frac{\text{(READ DATA)} - 128}{\text{(COEFFICIENT)}} = \frac{150 - 128}{1.6}\\ &= 13.75 \ \text{[Ao–p]}\end{aligned}$$

b) Ratings currents[Arms] of load current

$$\begin{aligned}\text{RATINGS CURRENTS[Arms]} &= \frac{\text{(READ DATA)} - 128}{\text{(COEFFICIENT)} \times \sqrt{2}} = \frac{150 - 128}{1.6 \times \sqrt{2}}\\ &= 9.72 \ \text{[Arms]}\end{aligned}$$

c) PERCENT(RATE)

$$\begin{aligned}\text{RATE OF LOAD[\%]} &= \frac{\text{LOAD CURRENT[Ao–p]}}{\text{PEAK CURRENT OF SERVO MOTOR}} \times 100\\ &= \frac{13.75}{23} \times 100 = 59.8 \ \text{[\%]}\end{aligned}$$

[Hardware]

When general analog input data is read, one of the following hardware items is required:

1 Sub–CPU board

2 Analog I/O module on the additional axis board

For details of the relationship between input numbers and connectors, refer to the connection manual. When an NC controlled axis load current is read, the hardware described above is not required.

[NC parameters]

When this function is used, the NC parameters listed below need to be set. For details, refer to the parameter descriptions.

1 Bit 6 of parameter No. 1810 = 0 (A/D conversion is performed.)

2 Bit 0 of parameter No. 1811 = 1 (A/D conversion data is output in high–speed mode.)

C.3.17 Reading the Tool Life Management Data (Tool Group Number) (Low–speed Response)

[Description]

The number of the tool group in which the tool number is cataloged is read.

Offset	Set data	Read data
0	Function code 38	Function code 38
2	Completion code —	Completion code
4	Data length —	Data length 4
6	Data number 0	Data number
8	Tool No.	Tool No.
10	Data area (4 byte)	Tool group No. (4 byte)
14		

NOTE
The data can be read only when the tool life management data function is provided.

C.3.18 Reading the Tool Life Management Data (Number of Tool Groups) (Low–speed Response)

[Description]

The number of tool groups contained in the tool life management data is read.

Offset	Set data	Read data
0	Function code 39	Function code 39
2	Completion code —	Completion code
4	Data length —	Data length 4
6	Data number 0	Data number
8	Data attribute 0	Data attribute
10	Data area (4 byte)	Number of tool groups (4 byte)
14		

C.3.19 Reading Tool Life Management Data (Number of Tools) (Low–speed Response)

[Description]

The number of tools cataloged in the specified tool group is read.

Offset	Set data	Read data
0	Function code 40	Function code 40
2	Completion code —	Completion code
4	Data length —	Data length 4
6	Tool group No.	Tool group No.
8	Data attribute 0	Data attribute
10	Data area (4 byte)	Number of tools (4 byte)
14		

C.3.20 Reading Tool Life Management Data (Tool Life) (Low–speed response)

[Description]

The tool life of the specified tool group is read.

Offset	Set data	Read data
0	Function code 41	Function code 41
2	Completion code —	Completion code
4	Data length —	Data length 4
6	Tool group No.	Tool group No.
8	Data attribute 0	Data attribute
10	Data area (4 byte)	Tool life (4 byte)
14		

C.3.21 Reading Tool Life Management Data (Tool Life Counter) (Low–speed Response)

[Description]

The tool life counter of the specified tool group is read.

Offset	Set data	Read data
0	Function code 42	Function code 42
2	Completion code —	Completion code
4	Data length —	Data length 4
6	Tool group No.	Tool group No.
8	Data attribute 0	Data attribute
10	Data area (4 byte)	Tool life counter (4 byte)
14		

C.3.22 Reading Tool Life Management Data (Tool Life Counter Type) (Low–speed Response)

[Description]

The tool life counter type of the specified tool group is read.

Offset	Set data	Read data
0	Function code 160	Function code 160
2	Completion code —	Completion code
4	Data length —	Data length —
6	Tool group No.	Tool group No.
8	Data attribute 0	Data attribute
10	Data area (4 byte)	Tool life counter type (4 byte)
14		

Tool life counter type
1 : The tool life counter indicates the number of times the tool has been used.
2 : The tool life counter indicates the period of time the tool has been used.

C.3.23 Reading Tool Life Management Data (Tool Length Compensation No.1) (Low–speed Response)

[Description]

A tool length compensation number is read according to the specified tool group number and tool number.

Set data

Offset	Content
0	Function code 43
2	Completion code —
4	Data length —
6	Tool group No.
8	Tool No.
10	Data area (4 byte)
14	

Read data

Offset	Content
0	Function code 43
2	Completion code
4	Data length 4
6	Tool group No.
8	Tool No.
10	Tool length compensa-tion No. (4 byte)
14	

If nothing is specified after the H code, the NC transfers 255 (FFH).

C.3.24 Reading Tool Life Management Data (Tool Length Compensation No.2) (Low–speed response)

[Description]

A tool length compensation number is read according to the specified tool group number and tool order number.

Set data

Offset	Set data
0	Function code 44
2	Completion code —
4	Data length —
6	Tool group No.
8	Tool order number
10	Data area (4 byte)
14	

Read data

Offset	Read data
0	Function code 44
2	Completion code
4	Data length 4
6	Tool group No.
8	Tool order number
10	Tool length compensation No. (4 byte)
14	

If nothing is specified after the H code, the NC transfers 255 (FFH).

C.3.25 Reading Tool Life Management Data (Cutter Compensation No.1) (Low–speed Response)

[Description]

A cutter compensation number is read according to the specified tool group number and tool number.

Set data

Offset	Content
0	Function code 45
2	Completion code —
4	Data length —
6	Tool group No.
8	Tool No.
10	Data area (4 byte)
14	

Read data

Offset	Content
0	Function code 45
2	Completion code
4	Data length 4
6	Tool group No.
8	Tool No.
10	Cutter compensation No. (4 byte)
14	

If nothing is specified after the D code, the NC transfers 255 (FFH).

C.3.26 Reading Tool Life Management Data (Cutter Compensation No.2) (Low–speed Response)

[Description]

A cutter compensation number is read according to the specified tool group number and tool order number.

Set data

Offset	Content
0	Function code 46
2	Completion code —
4	Data length —
6	Tool group No.
8	Tool order number
10	Data area (4 byte)
14	

Read data

Offset	Content
0	Function code 46
2	Completion code
4	Data length 4
6	Tool group No.
8	Tool order number
10	Cutter compensation No. (4 byte)
14	

If nothing is specified after the D code, the NC transfers 255 (FFH).

C.3.27 Reading Tool Life Management Data (Tool Information 1) (Low–speed Response)

[Description]

The tool information (status) is read according to the specified tool group number and tool number.

Set data

Offset	Contents
0	Function code 47
2	Completion code —
4	Data length —
6	Tool group No.
8	Tool No.
10	Data area (4 byte)
14	

Read data

Offset	Contents
0	Function code 47
2	Completion code
4	Data length 4
6	Tool group No.
8	Tool No.
10	Tool information (4 byte)
14	

Tool information
1 : The tool is cataloged.
2 : The tool life has expired.
3 : The tool was skipped.

C.3.28 Reading Tool Life Management Data (Tool Information 2) (Low–speed Response)

[Description]

The tool information (status) is read according to the specified tool group number and tool order number.

Offset	Set data	Offset	Read data
0	Function code 48	0	Function code 48
2	Completion code —	2	Completion code
4	Data length —	4	Data length 4
6	Tool group No.	6	Tool group No.
8	Tool order number	8	Tool order number
10	Data area (4 byte)	10	Tool information (4 byte)
14		14	

Tool information
See the description in Section 3.27.

C.3.29 Reading Tool Life Management Data (Tool Number) (Low–speed Response)

[Description]

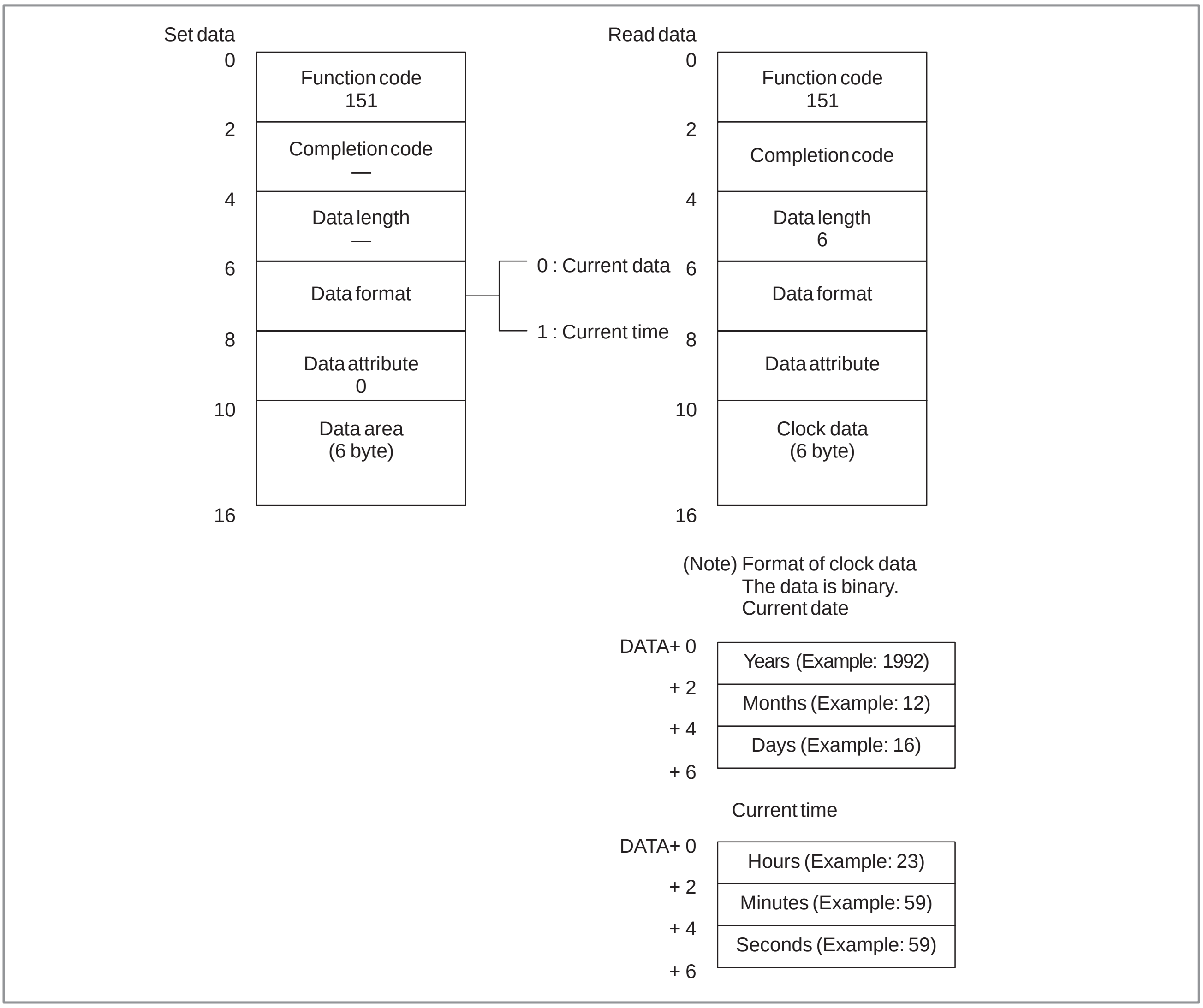
A tool number is read according to the specified tool group number and tool order number.

Offset	Set data	Offset	Read data
0	Function code 49	0	Function code 49
2	Completion code —	2	Completion code
4	Data length —	4	Data length 4
6	Tool group No.	6	Tool group No.
8	Tool order number	8	Tool order number
10	Data area (4 byte)	10	Tool No. (4 byte)
14		14	

C.3.30 Reading Clock Data (Low–speed Response)

[Description]

The current data (year, month, day) and current time (hours, minutes, seconds) can be read from the clock built into the CNC.



C.3.31 Reading the Relative Position on a Controlled Axis

[Description]

The relative position (relative coordinates) on a feed axis controlled by the CNC is read.

Set data

Offset	Content
0	Function code 74
2	Completion code —
4	Data length —
6	Data number 0
8	Axis number
10	Data area (4 byte)
14	

Read data

Offset	Content
0	Function code 74
2	Completion code
4	Data length 4
6	Data number
8	Axis number
10	Relative position (4 byte)
14	

1) The unit of the read value is determined as follows:

   (1) For the machining center system or when the radius is specified for the axis of the lathe systm.
   The data shows double of the present position with the least input increment as a unit.

   (2) When the diameter is specified for the lathe system

   The data shows the present position with the least input increment.

   (3) When the input unit is multiplied by 10

   The data shows twenty–times the present position (radius programming) or ten–times the present position (diameter programming) with the last command increment as a unit.

2) The present position of a moving axis can be read whenever the function instruction is executed.

C.3.32 Reading the Remaining Travel

[Description]

The remaining traveling distance on a feed axis controlled by the CNC is read.

Set data

Offset	Set data
0	Function code 75
2	Completion code —
4	Data length —
6	Data number 0
8	Axis number
10–14	Data area (4 byte)

Read data

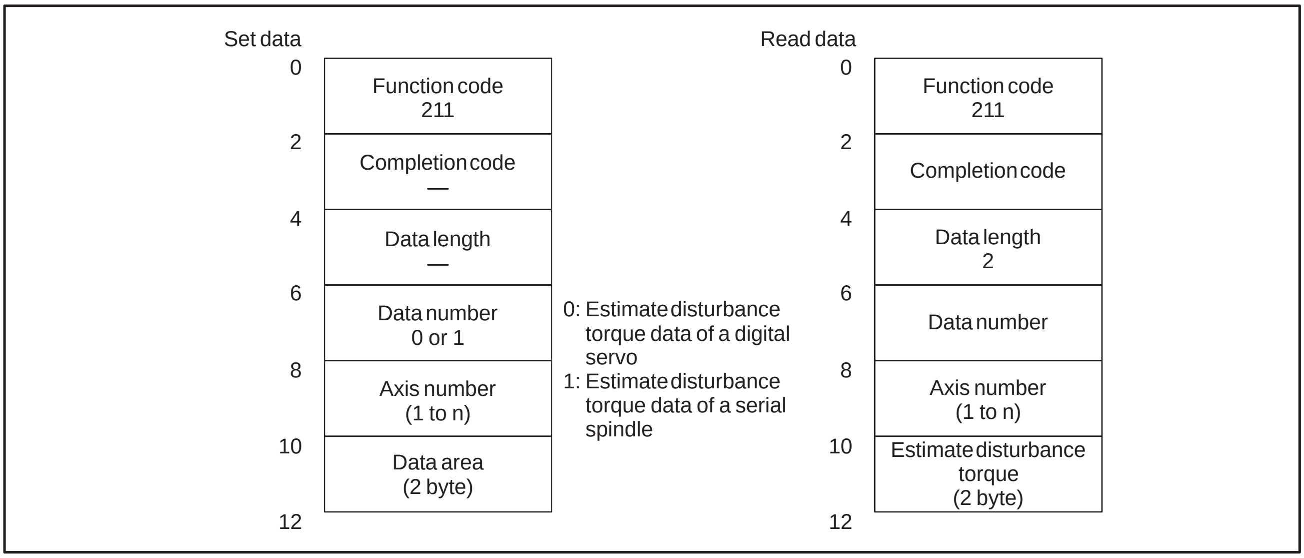
Offset	Read data
0	Function code 75
2	Completion code
4	Data length 4
6	Data number
8	Axis number
10–14	Remaining travel (4 byte)

C.3.33 Reading an Estimate Disturbance Torque Data

[Description]

1) The load torques except a necessary torque for acceleration/ deceleration of the torques of the servo axis are read.

2) The load torques except a necessary torque for acceleration/ deceleration of the torques of the serial spindle axis are read.



Kind of data	Data specification
Estimate disturbance torque data of a digital servo	Please refer to ”FANUC AC SERVO AMPLIFIER AMINTENANCE MANUAL (B–65005E)” for correspondence of the load torque with the value of the read data.
Estimate disteurbance torque data of a serial spindle	Please refer to ”FANUC AC SPINDLE SERVO UNIT (SERIAL INTERFACE) MAINTENANCE MANUAL (B–65045E)” for correspondence of the load torque with the value of the read data. The load torque of the spindle is understood from the undermentioned calculation type. $\text{Load torque} = \dfrac{\text{The read data}}{16384} \times \text{Maximum output torque of spindle}$

* For an explanation of the CNC parameters, refer to the “CNC Parameter Description.”

C.3.34
Reading the Machining Time (Low–speed Response)

[Description]

The machining time currently specified for a program is read.

Set data

0	Function code 178
2	Completion code —
4	Data length —
6	Program number
8	Data attribute 1
10	Data area (6 byte)
16	

Read data

0	Function code 178
2	Completion code
4	Data length 6
6	Program number
8	Data attribute 1
10	Machining time (6 byte)
16	

C.3.35 Reading the Load Current (A/D Conversion Data) for the Spindle Motor

[Description]

The load current for the spindle (spindle motor) is converted to a digital value and the digital value is read. (See Section 3.16, ”Reading the Load Current (A/D Conversion Data) for the Feed Motor.”)

Set data

Offset	
0	Function code 153
2	Completion code —
4	Data length —
6	Data number 0
8	Axis number (1 to n)
10	Data area (2 byte)
12	

The 15B supports two spindles, while the 15*i*A supports four spindles.

Read data

Offset	
0	Function code 153
2	Completion code
4	Data length 2
6	Data number
8	Axis number (1 to n)
10	A/D conversion data (2 byte)
12	

[Hardware]

When general analog input data is read, one of the following hardware items is required:

1 Sub–CPU board

2 Analog I/O module on the additional axis board

With a serial spindle, however, the hardware described above is not required if CNC software of mass production version IV or later is used. For the relationship between input numbers and connectors, refer to the connection manual.

[NC parameters]

When this function is used, the NC parameters listed below must be set. For details, refer to the parameter descriptions.

1 Bit 6 of parameter No. 1810 = 0 (A/D conversion is performed.)

2 Bit 0 of parameter No. 1811 = 1
(A/D conversion data is output in high–speed mode.)

C.3.36 Reading the Tool Offset Data According to the Specified Tool Number

[Description]

The tool number is spedified and the tool offset data is read.

Set data

Offset	Content
0	Function code 213
2	Completion code —
4	Data length —
6	Data format
8	Tool number
10	Data area (4 byte)
16	

Read data

Offset	Content
0	Function code 213
2	Completion code
4	Data length 4
6	Data format
8	Tool number
10	offset data (4 byte)
16	

Kind of the data to be read	The data form CTL1	Tool number CTL2
Tool number	01	Tool display number
Pot number	10	Tool No.
Pot number	11	Tool display number
Tool length compensation value	20	Tool No.
Tool length compensation value	21	Tool display number
Cutter compensation value	30	Tool No.
Cutter compensation value	31	Tool display number

NOTE

Please use the bit 4 of NC parameter as 1.
When the completion code “5” is returned, change the format of the window in the SETTING Screen.
(REFERENCE:chapter II 4.4 SETTING Screen)

C.3.37 Reading Tool Life Management Data (Tool Group Numbers) (Low–speed Type)

[Explanation of data]

The tool group number in which a tool number is registered is read. Note that the tool number area is four bytes in length so that a tool number of up to eight digits can be specified.

[Input data structure]

Settings

Offset	Content
0	Function code 200
+2	Completion code —
+4	Data length —
+6	Data format 0
+8, +10	Tool number
+12, +14	Data area (4 byte)
+16	

After reading

Offset	Content
0	Function code 200
+2	Completion code
+4	Data length 4
+6	Data format
+8, +10	Tool number
+12, +14	Tool group number (4 byte)
+16	

NOTE
This function is provided by the FS15*i*A (PMC–NB6) only.

C.3.38 Reading Tool Life Management Data (Tool Length Compensation Number 1) (Low–speed Type)

[Explanation of data]

The tool length compensation number corresponding to the specified tool group number and tool number is read.
Note that the tool number area is four bytes in length so that a tool number of up to eight digits can be specified.

[Input data structure]

Settings

Offset	Field
0	Function code 227
+2	Completion code —
+4	Data length —
+6	Tool group number
+8, +10	Tool number
+12, +14	Data area (4 byte)
+16	

After reading

Offset	Field
0	Function code 227
+2	Completion code
+4	Data length 4
+6	Tool group number
+8, +10	Tool number
+12, +14	Tool length compensation number (4 byte)
+16	

NOTE
This function is provided by the FS15*i*A (PMC–NB6) only.

C.3.39 Reading Tool Life Management Data (Cutter Compensation Number 1) (Low–speed Type)

[Explanation of data]

The cutter compensation number corresponding to the specified tool group number and tool number is read.
Note that the tool number area is four bytes in length so that a tool number of up to eight digits can be specified.

[Input data structure]

Settings

Offset	Field
0	Function code 228
+2	Completion code —
+4	Data length —
+6	Tool group number
+8 +10	Tool number
+12 +14	Data area (4 byte)
+16	

After reading

Offset	Field
0	Function code 228
+2	Completion code
+4	Data length 4
+6	Tool group number
+8 +10	Tool number
+12 +14	Cutter compensation number (4 byte)
+16	

NOTE
This function is provided by the FS15*i*A (PMC–NB6) only.

C.3.40 Reading Tool Life Management Data (Tool Information 1) (Low–speed Type)

[Explanation of data]

The tool information (state) corresponding to the specified tool group number and tool number is read.
Note that the tool number area is four bytes in length so that a tool number of up to eight digits can be specified.

[Input data structure]

Offset	Settings	After reading
0	Function code 201	Function code 201
+2	Completion code —	Completion code
+4	Data length —	Data length 4
+6	Tool group number	Tool group number
+8 to +10	Tool number	Tool number
+12 to +14	Data area (4 byte)	Tool length compensation number (4 byte)
+16		

Explanation of tool information
1 : The tool is registered.
2 : The tool has expired.
3 : The tool has been skipped.

NOTE
This function is provided by the FS15*i*A (PMC–NB6) only.

C.3.41 Reading Real Parameters (Low–speed Type)

[Explanation of data]

Real parameters are read from the CNC.

[Input data structure]

Offset	Settings	After reading
0	Function code 321	Function code 321
+2	Completion code —	Completion code
+4	Data length —	Data length 4
+6, +8	Parameter number	Parameter number
+10	Data attribute	Data attribute
+12	Decimal point position	Decimal point position
+14, +16	Data area (4 byte)	Parameter value (4 byte)
+18		

Data attribute: 0: no axis; 1 to n: axis specified

NOTE
1 Integer parameters cannot be read.
2 This function is provided by the FS15*i*A (PMC–NB6) only.

Example) The value of a read–out parameter is as follows:

(Value of a read–out parameter) =
(value of the parameter on the NC) $\times$ $10^{\text{(specified decimal point position)}}$

Parameter value	Value on the NC	Decimal point position
1	1.123	0
12		1
123		2
1234		3
12340		4

C.4 FORMAT AND DETAILS OF THE CONTROL DATA OF THE WINDW FUNCTIONAL INSTRUCTION

(1) See the description of the window function. The data item marked with a dash (–) in the description of the data structure need not be entered.
When output, the data item has no meaning.

(2) The length of all data blocks and data items is represented in bytes.

(3) The output data becomes valid only when the instruction terminates normally.

Completion code	Description
–10	The window instruction is being processed. Hold ACT until W1 is set to 1.
0	The instruction terminated normally.
1	An error occurred. The corresponding function number is not found.
2	An error occurred. Possible causes include the following: Wrong data is found in the CTL area. The NC does not have the corresponding function.
3	An error occurred. The specified axis is not provided.
5	An error occurred. it is a mistake of data form. Though the function supports only a new format, a old format is specified.

C.4.1 Writing a Tool Offset Data

[Description]

The data is directly written into the tool offset value (tool compensation) area of the CNC.

Set data

Set data	
0	Function code 14
2	Completion code
4	Data length 4
6	Offset number
8	Offset format
10	Tool offset value (4 byte)
14	

(Note 1) Offset format

M system (machining center system)

Data type	Format	Offset number (CTL+2, 3)
Tool compensation A Compensation	 1	 Offset number
Tool compensation B Geometry compensation Wear compensation	 1 1	 Offset number Offset number +1000
Tool compensation C Tool length Geometry compensation Wear compensation Cutter Geometry compensation Wear compensation	 1 1 2 2	 Offset number Offset number +1000 Offset number Offset number +1000

T system (lathe system)

Data type	Format	Offset number (CTL+2, 3)
Tool compensation A Compensation along the X–axis Compensation along the Z–axis Tool–tip radius compensation Compensation along the Y–axis Compensation related to the position of the virtual tool	 1 2 3 4 5	 Offset number Offset number Offset number Offset number Offset number Offset number
Tool compensation B Geometry compensation Compensation along the X–axis Compensation along the Z–axis Tool–tip radius compensation Compensation along the Y–axis Wear compensation Compensation along the X–axis Compensation along the Z–axis Tool–tip radius compensation Compensation along the Y–axis Compensation related to the position of the virtual tool	 1 2 3 4 1 2 3 4 5	 Offset number Offset number Offset number Offset number Offset number +1000 Offset number +1000 Offset number +1000 Offset number +1000 Offset number

C.4.2 Writing a Parameter (Setting Data)

[Description]

The data is written into the parameter area of the CNC.

The parameters of the CNC are classified into four types according to the smallest unit that has a meaning. Bit parameter: Each bit has a meaning. Byte parameter: Each byte has a meaning. Word parameter: Each set of two bytes has a meaning. Double word parameter: Each set of four bytes has a meaning.

Each bit of a bit parameter cannot be written individually. The eight bits (one byte) of the parameter must be written at a time. To change a bit of a bit parameter, read the entire parameter, change the desired bit, then write the entire parameter.

Set data		
0	Function code 18	
2	Completion code	
4	Data length 4	
6	Parameter number	
8	Data attribute	0 : No axis 1 to n : A specified axis
10	Parameter data (4 byte)	
14		

(Note 1) Format of parameter data
If the data is one byte long, it is set in the DATA+0 area.

10	DATA+0
11	DATA+1
12	DATA+2
13	DATA+3
14	

C.4.3 Writing a Custom Macro Variable

[Description]

The data is written into the custom macro variable area of the CNC.

Set data	
0	Function code 22
2	Completion code
4	Data length 6
6	Custom macro number
8	Data attribute 0
10	Custom macro variable value (4 byte)
14	Position of decimal point (2 byte)
16	

(Note 1) In the case of writing a Custom Macro Variable of upper than 100000.
Please input ”10” to ”Data attribute”, and input last four digits of variable number to ”Custom macro variable number”.

(Note 2) Specification of the position of the decimal point

	7	6	5	4	3	2	1	0
DATA+4								

#0 to #3 : Set number of digit beelow decimal point.
#4 to #7 : Set to ”0”.

Examples

$$(\text{Value written in the NC}) = \frac{(\text{value of custom macro variable})}{10^{(\text{Positon of decimal point})}}$$

Value in the NC	Custom macro variable value	Position of decimal point
1234. 000	1 2 3 4	0
123. 400		1
12. 340		2
1. 124		3
0. 1234		4

C.4.4 Writing a Data on the Program Check Screen

[Description]

The data to be displayed on the program check screen of the CNC is rewritten.

Set data	
0	Function code 150
2	Completion code
4	Data length 2
6	Data type
8	Data attribute 0
10	Data on the program check screen (2 byte)
12	

Data type	Data type	Attribute
M code which is being executed (1 to 5)	1 to 5	0
Spindle speed range	100	0
Spindle tool number	200	0
Number of the tool to be used next	201	0

NOTE

1 As much program check screen data as the amount specified with the data length can be written.

2 When the 15*i* is used, the spindle tool number and the next machining tool number can each be written into a 2– or 4–byte area.

C.4.5 Writing the Torque Limit Override

[Description]

The torque limit override of the specified feed axis is rewritten.

Set data		Value
0	Function code 152	
2	Completion code	
4	Data length 2	
6	Data type 0	
8	Axis number	
10	Torque limit override	Un–signed binary <Unit: %> The values from 0 to 255 correspond to 0% to 100%.
12		

[Example]

If the torque limit override is 50%, please set to 128.

> **NOTE**
> This window is valid for parameters set on the CNC.
> Parameter 1802#4 { 0: Fixed to override 100%
> 1: This window enabled.

C.4.6 Writing the Tool Life Management Data (Tool Group Number)

[Description]

The tool number and the tool life value are written into the specified tool group.

Set data	
0	Function code 163
2	Completion code –
4	Data length 4
6	Data number 0
8	Tool No.
10	Tool group No.
12	(4 byte) Tool life value
14	

Data type	Specification
Number of tool groups	Register the tool group. Set group number, tool life value, and tool life counter for transmission data. 7 6 5 4 3 2 1 0 DATA+0, DATA+1: Group number, Life counter type DATA+1 BIT 7 DATA+2, DATA+3: Tool life value Group number 1 to 512 Tool life value 1 to 9999 (Specified number of time) 1 to 4300 (Specified time) Tool life counter type (DATA+1 BIT 7) 0: Number of time 1: Time (minute)

C.4.7 Writing the Tool Life Management Data (Tool Life)

[Description]

The data is written into the tool life value area of the specified tool group.

Set data	
0	Function code 164
2	Completion code —
4	Data length 4
6	Tool group No.
8	Data attribute 0
10	Tool life (4 byte)
14	

C.4.8 Writing the Tool Life Management Data (Tool Life Counter)

[Description]

The data is written into the tool life counter area of the specified tool group.

Set data	
0	Function code 165
2	Completion code —
4	Data length 4
6	Tool group No.
8	Data attribute 0
10	Tool life counter (4 byte)
14	

C.4.9 Writing the Tool Life Management Data (Tool Life Counter Type)

[Description]

The data is written into the tool life counter type area of the specified tool group.

Set data	
0	Function code 166
2	Completion code —
4	Data length 4
6	Tool group No.
8	Data attribute 0
10	Tool life counter type (4 byte)
14	

Tool life counter type
1 : The tool life counter indicates the number of times the tool has been used.
2 : The tool life counter indicates the period of time the tool has been used.

C.4.10 Writing the Tool Life Management Data (Tool Length Compensation Number 1)

[Description]

The data is written into the tool length compensation number area specified by the tool group number and tool number.

Set data	
0	Function code 167
2	Completion code —
4	Data length 4
6	Tool group No.
8	Tool No.
10	Tool length compensation No. (4 byte)
14	

C.4.11 Writing the Tool Life Management Data (Tool Length Compensation Number 2)

[Description]

The data is written into the tool length compensation number area specified by the tool group number and tool order number.

Set data	
0	Function code 168
2	Completion code —
4	Data length 4
6	Tool group No.
8	Tool order number
10	Tool length compensation No. (4 byte)
14	

C.4.12 Writing the Tool Life Management Data (Cutter Compensation Number 1)

[Description]

The data is written into the cutter compensation number area specified by the tool group number and tool number.

Set data	
0	Function code 169
2	Completion code —
4	Data length 4
6	Tool group No.
8	Tool No.
10	Cutter compensation No. (4 byte)
14	

C.4.13 Writing the Tool Life Management Data (Cutter Compensation Number 2)

[Description]

The data is written into the cutter compensation number area specified by the tool group number and tool order number.

Set data	
0	Function code 170
2	Completion code —
4	Data length 4
6	Tool group No.
8	Tool order number
10	Cutter compensation No. (4 byte)
14	

C.4.14 Writing the Tool Life Management Data (Tool Information 1)

[Description]

The data is written into the tool information (status) area specified by the tool group number and tool number.

Set data	
0	Function code 171
2	Completion code —
4	Data length 4
6	Tool group No.
8	Tool No.
10	Tool information (4 byte)
14	

Tool information
1 : The tool is cataloged.
2 : The tool life has expired.
3 : The tool was skipped.

C.4.15 Writing the Tool Life Management Data (Tool Information 2)

[Description]

The data is written into the tool information (status) area specified by the tool group number and tool order number.

Set data	
0	Function code 172
2	Completion code —
4	Data length 4
6	Tool group No.
8	Tool order number
10	Tool information (4 byte)
14	

C.4.16 Writing the Tool Life Management Data (Tool Number)

[Description]

A tool number is written into (added to) the area specified by the tool group number and tool order number.

Set data	
0	Function code 173
2	Completion code —
4	Data length 4
6	Tool group No.
8	Tool order number
10	Tool No. (4 byte)
14	

C.4.17 Writing the Tool Offset Data According to the Specified Tool Number

[Description]

The tool number is specified and the tool offset data is written.

Set data

Offset	Content
0	Function code 214
2	Completion code —
4	Data length 4
6	Data format
8	Tool number
12	offset data (4 byte)
16	

Kind of the data to be written	The data form CTL1	Tool number CTL2
Change of Tool number	00	Tool No.
Change of Tool number	01	Tool display number
Pot number	10	Tool No.
Pot number	11	Tool display number
Tool length compensation value	20	Tool No.
Tool length compensation value	21	Tool display number
Cutter compensation value	30	Tool No.
Cutter compensation value	31	Tool display number
Addition of Tool number	40	Tool No.
Addition of Tool number	41	Tool display number

NOTE

(Only Series 15B)
Please use the bit 4 of NC parameter as 1.
When the completion code “5” is returned, change the format of the window in the SETTING Screen.
(REFERENCE : chapter II 4.4 SETTING Screen)

C.4.18 Writing the Superposition Move Command

(1) For three axes

[Description]

After the axes for manual handle feed are selected in the manual handle feed mode, the traveling distances (number of pulses) corresponding to three manual pulse generators are written. The set value ranges from –256 to +256.

The specified number of pulses is assumed to be the number of pulses entered from the manual pulse generator. The speed is calculated as follows: (specified number of pulses) × (magnification) × 62.5 (pulses/second) The data in parameters 1413 and 1414 of the CNC is valid for this function.

Set data	
0	Function code 215
2	Completion code
4	Data length 6
6	Data number 0
8	Data attribute 0
10	Superposition move command (6 byte)
16	

(Note 1) Format of superposition move command

DATA+0	First manual pulse generator
DATA+2	Second manual pulse generator
DATA+4	Third manual pulse generator
DATA+6	

(2) For four axes

Set data	
0	Function code 215
2	Completion code
4	Data length 8
6	Axis specification mode
8	Axis number
10	Superposition move command (8 byte)
18	

Axis specification mode

7 6 5 4 3 2 1 0

#0 : Mode selection (0: The BMI signal is selected. 1: An axis number is selected.)

Axis number

	7 6 5 4	3 2 1 0
CTL+8	P2	P1
CTL+9	P4	P3

CTL+8, #0 to #3: Axis number of the first pulse generator (P1)
CTL+8, #4 to #7: Axis number of the second pulse generator (P2)
CTL+9, #0 to #3: Axis number of the third pulse generator (P3)
CTL+9, #4 to #7: Axis number of the fourth pulse generator (P4)

(Note 1) Format of superposition move command

DATA+0	First manual pulse generator
DATA+2	Second manual pulse generator
DATA+4	Third manual pulse generator
DATA+6	Fourth manual pulse generator
DATA+8	

C.4.19 Writing the Feedrate

[Description]

Feedrate writing can be specified only in the feed–per–minute mode. The velocity command is specified with $F \times 10^{-d}$. A flag is provided to validate either the command of the PMC or the feedrate of the CNC. After the flag is set, the velocity command specified in the NC is invalidated. To validate the feedrate of the NC, set the flag to 0.

Set data

Offset	Content
0	Function code 216
2	Completion code
4	Data length 6
6	Data number 0
8	Data attribute 0
10	Feedrate (6 byte)
16	

(Note 1) Format of feedrate

	7 6 5 4 3 2 1 0
DATA+0	(bit 2: flag)
DATA+2	Feedrate, F
DATA+4	Exponent of feedrate, d
DATA+6	

Flag for validating the velocity command : Bit 2 of DATA+0 (0: The NC is valid. 1: The PMC is valid.)

Feedrate, F : DATA+2 and DATA+3

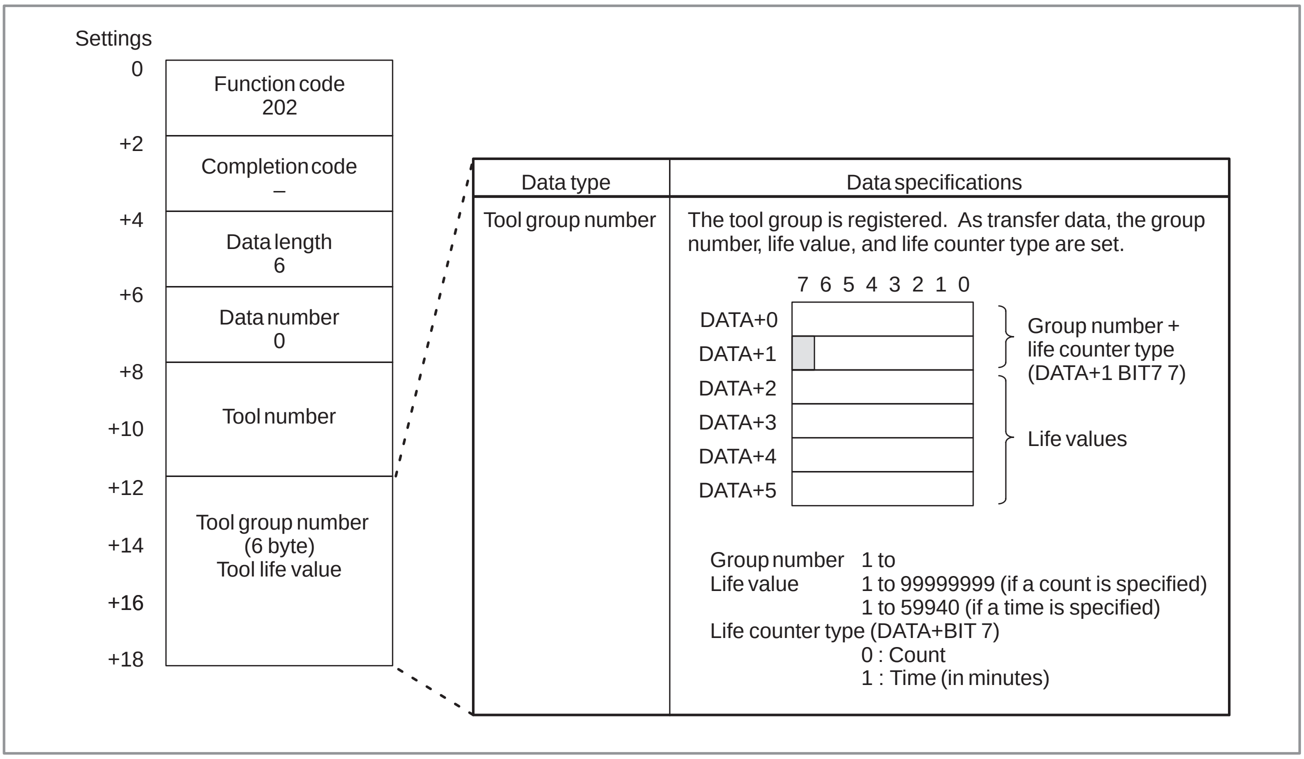
Exponent of feedrate, d : DATA+4 and DATA+5

C.4.20 Writing Tool Life Management Data (Tool Group Numbers)

[Explanation of data]

A tool number and a tool life value are written to a specified tool number. Note that the tool number area is four bytes in length so that a tool number of up to eight digits can be specified.

[Input data structure]



> **NOTE**
> This function is provided by the FS15*i*A (PMC–NB6) only.

C.4.21 Writing Tool Life Management Data (Tool Length Compensation Number 1)

[Explanation of data]

Data is written to the tool length compensation number specified by a tool group number and a tool number.
Note that the tool number area is four bytes in length so that a tool number of up to eight digits can be specified.

[Input data structure]

Settings	
0	Function code 229
+2	Completion code —
+4	Data length 4
+6	Tool group number
+8 +10	Tool number
+12 +14	Tool length compensation number (4 byte)
+16	

NOTE
This function is provided by the FS15*i*A (PMC–NB6) only.

C.4.22 Writing Tool Life Management Data (Cutter Compensation Number 1)

[Explanation of data]

Data is written to the cutter compensation number specified by a tool group number and a tool number.
Note that the tool number area is four bytes in length so that a tool number of up to eight digits can be specified.

[Input data structure]

Settings	
0	Function code 230
+2	Completion code —
+4	Data length 4
+6	Tool group number
+8	Tool number
+10	
+12	Cutter compensation number (4 byte)
+14	
+16	

NOTE
This function is provided by the FS15*i*A (PMC–NB6) only.

C.4.23 Writing Tool Life Management Data (Tool Information 1)

[Explanation of data]

Data is written to the tool information (state) specified by a tool group number and a tool number.
Note that the tool number area is four bytes in length so that a tool number of up to eight digits can be specified.

[Input data structure]

Settings	
0	Function code 231
+2	Completion code —
+4	Data length 4
+6	Tool group number
+8	Tool number
+10	
+12	Tool information (4 byte)
+14	
+16	

Explanation of tool information
1 : The tool is registered.
2 : The tool has expired.
3 : The tool has been skipped.

> **NOTE**
> This function is provided by the FS15*i*A (PMC–NB6) only.

C.4.24 Writing Real Parameters (Low–speed Type)

[Explanation of data]

Real parameters are written to the CNC.

[Input data structure]

Settings		
0	Function code 323	
+2	Completion code —	
+4	Data length –	
+6, +8	Parameter number	
+10	Data attribute	0: no axis 1 to n: axis specified
+12	Decimal point position	
+14, +16	Data area (4 byte)	
+18		

NOTE
1 This function is provided by the FS15*i*A (PMC–NB6) only.
2 Integer parameters cannot be written.

Example)

$$(\text{Value to be set on the NC}) = \frac{(\text{Parameter variable value})}{10^{(\text{specified decimal point position})}}$$

Value to be set on the NC	Custom macro variable value	Decimal point position
1234.000 123.400 12.640 1.234 0.1234	1 2 3 4	0 1 2 3 4

D WINDOW FUNCTION DESCRIPTION (FS16–LA)

D.1 OUTLINE

The following function is added to PMC–CNC window function for FS16–LA.

(1) Transferring a processing condition file in non–volatile memory to data area in CNC memory, and vice versa.

(2) Reading a comment command in a part program.

(3) Reading data commanded to laser oscillator

The following functions are added to the PMC–CNC window function on the FS16*i*–LA:

(1) Reading of comments
Comments specified within programs can be read.

(2) Laser command value data and laser setting data can be read and written, and command value data to the laser oscillator can be read.

> **NOTE**
> Data transfer of machining condition files between data areas is not possible by ladder diagrams on the FS16*i*–LA.

D.2 FUNCTION

D.2.1 Transfer Between Data Area and Non–volatile Memory

(1) Transfer from data area to non–volatile memory. (% low–speed type)

[Contents of data]

The data can be transferred from the data area in CNC to PMC non–volatile memory by PMC–RC application.

Setting the original data set or group in data attribute M.

Setting the written address of non–volatile memory for the offset address from the top address in address N.

Setting the total byte No. of written data set in address L.

Setting the forward structure of data set in data.

And the data can be transferred set or group from data area to non–volatile memory.

[Structure of input data]

Top address + 0	(Function code) 183	
2	(End Code) — (No need to set)	The following End Code is output at reading end. 0 : normal end 3 : The incorrect data is set in Address. 4 : The incorrect data is set in Data attribute.
4	(Data length) L	L = Set the total byte No. of the data structure for 1 set in non–volatile memory. This data must be set in case of transferring the data group. (No–need to set in case of transferring the set)
6	(Address) N	N = Set the forward address by the offset from the top address of non–volatile memory.
8	(Data attribute) M	M = Set the kind of data set or group to transfer 1–10 : Set of processing data 101–103 : Set of piercing d data 201–205 : Set of edge process data 1000 : Group of processing data 1001 : Group of piercing data 1002 : Group of edge process data
10	(Data)	Structure of data set Set the correspondence of 1 set of data item in data area and data item in non–volatile memory by relative address for non–volatile memory.

(2) Transfer from non–volatile memory to data area (% low–speed type)

[Contents of data]

The data can be transferred from the processing condition file registered in non–volatile memory to the data area in CNC by PMC–RC application.

Setting the forward data set or group in data attribute M.

Setting the original read address for the offset address from non–volatile memory top address in address N.

Setting the total byte No. of original data set in address L.

Setting the original structure of set in data.

And the data of set or group can be transferred from non–volatile memory to data area.

[Structure of input data]

Address	Field	Description
Top address + 0	(Function code) 184	
2	(End Code) — (No need to set)	The following End Code is output at reading end. 0 : normal end 3 : The incorrect data is set in Address. 4 : The incorrect data is set in Data attribute.
4	(Data length) L	L = Set the total byte No. of the data structure for 1 set in non–volatile memory. This data must be set in case of transferring the data group. (No–need to set in case of transferring the set)
6	(Group No.) N	N = Set the original address by the offset from the top address of non–volatile memory.
8	(Data attribute) M	M = Set the kind of data set or group to transfer 1–10 : Set of processing data 101–103 : Set of piercing d data 201–205 : Set of edge process data 1000 : Group of processing data 1001 : Group of piercing data 1002 : Group of edge process data
10	(Data)	Structure of data set Set the correspondence of 1 set of data item in data area and data item in non–volatile memory by relative address for non–volatile memory.

(3) Data structure of data area

(a) Processing data set

Address	Data item	Byte No.
0	Feed–rate	4
4	Peak power	2
6	Pulse frequency	2
8	Pulse duty	2
10	Assist gas pres.	2
12	Assist gas select	2
14	Assist gas settling time	2
16	Reference displacement	2
18	Offset amount	4
22	Edge process select	2
24	Start–up process select	2

(b) Piercing data set

Address	Data item	Byte No.
0	Peak power	2
2	Initial frequency	2
4	Initial duty	2
6	Frequency increment	2
8	Duty increment	2
10	Step time	2
12	Step No.	2
14	Piercing time	4
18	Assist gas pres.	2
20	Assist gas select	2
22	Assist gas settling time	2
24	Reference displacement	2

(c) Edge processing data set

Address	Data item	Byte No.
0	Judge angle	2
2	Peak power	2
4	Pulse frequency	2
6	Pulse duty	2
8	Piercing time	4
12	Assist gas pres.	2
14	Assist gas select	2
16	Return distance	4
20	Return feed rate	2
22	Return frequency	2
24	Return duty	2

NOTE

Example of data set

The address in Data is set as follows, for example, in case of the following data structure of processing condition file in non–volatile memory.

Data structure of data area

Address	Data
0	Feed–rate
4	Peak power
6	Pulse frequency
8	Pulse duty
10	Assist gas pres.
12	Assist gas select
14	Assist gas time
16	Ref. displacement
18	Offset amount
22	Edge select
24	Start–up select

The example of data structure for the processing condition file in non–volatile memory

Address	Data
0	Feed–rate
4	Peak power
6	Pulse frequency
8	Pulse duty
10	Focus distance
12	Assist gas pres.
14	Assist gas select
16	Assist gas time
18	Ref. displacement
20	Offset amount
24	Pulse type
26	Edge select
28	Start–up select

Data item in data area	Address	Data setting value
Feed–rate	Top address +10	0
Peak power	+12	4
Pulse frequency	+14	6
Pulse duty	+16	8
Assist gas pres.	+18	12
Assist gas select	+20	14
Assist gas time	+22	16
Ref. displacement	+24	18
Offset amount	+26	20
Edge select	+28	26
Start–up select	+30	28

D.2.2 Reading of the Comment

The data in the parentheses is written in the comment area, if the following M–code is commanded in a part program. This comment can be read from PMC.

```
Mxxx (*******) ;
```

└ Less than 24 characters, including alphabet, numeral, decimal–point and +/–

[Contents of data]

Contents of the data can be read for ASCII code.

[Structure of input data]

Address	Item	Description
Top address + 0	(Function code) 140	
2	(End Code) — (No need to set)	The following End Code is output at writing end. 0 : normal end 1 : The data length of comment data is over 24 bytes
4	(Data length) L	L = Data length of the comments read in. (Added 1 in case of the odd number.)
6	(Group No.) —	
8	(Data attribute) —	
10	(Data)	The read comment data is set.

NOTE
1 The comment is over–written if the next comment is input.
2 M–code number for reading of the comment is set to parameter number 15350.
Setting value is 0 to 999.

D.2.3 Reading and Writing the Laser Command Data and Laser Setting Data

(1) Reading the laser command data and laser setting data (✻high–speed type)

[Contents of the data]

The laser command data and laser setting data for CNC can be read by PMC–RC application. The data are separated to groups and can be read by the group.

[Structure of input data]

Top address +	Contents	Description
0	(Function code) 186	
2	(End Code) — (No need to set)	The following End Code is output at writing end. 0 : normal end 2 : The incorrect data is set for Data length. 3 : The incorrect data is set for Data No.
4	(Data length) L	L = Set the byte No. assigned for the group.
6	(Group No.) N	N = Set the group No.
8	(Data attribute) — (No need to set)	
10	(Data)	

(2) Writing the laser command data and laser setting data (✻low–speed type)

[Contents of the data]

The data can be written to the laser command data for CNC by PMC–RC application. The data are separated to groups and can be written by the group.

[Structure of input data]

Top address + 0	(Function code) 187	
2	(End Code) — (No need to set)	The following End Code is output at writing end. 0 : normal end 2 : The incorrect data is set for Data length. 3 : The incorrect data is set for Data No. 5 : The data beyond the allowable range is set for data command to write.
4	(Data length) L	L = Set the byte No. assigned for the group.
6	(Group No.) N	N = Set the group No.
8	(Data attribute) — (No need to set)	
10	(Data)	

(3) The data structure of the laser command data and laser setting data

Group No.	Address Top add. +	Byte No. for every item	Data length Byte No.	Item
0	10 12	2 2	4	Power control duty const Power control minimum duty
1	10 12 14 16	2 2 2 4	10	Power monitor (Read only) Power offset (Read only) Actual power (Read only) Actual feedrate (Read only)
2	10	2	2	Power input offset coe.
3	10 12	2 2	4	Assist gas select Assist gas flow select
4	10 12 14 16 18 20 22 24 26 28 30 32 34 36 38	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	30	Assist gas flow–1 pre–time pre–pres wrk–pres aft–time aft–pres Assist gas flow–2 pre–time pre–pres wrk–pres aft–time aft–pres Assist gas flow–3 pre–time pre–pres wrk–pres aft–time aft–pres
5	10 12 14	2 2 2	6	Processing peak power Processing pulse frequency Processing pulse duty
6	0 12 14 16	2 2 2 4	10	Piercing peak power Piercing pulse frequency Piercing pulse duty Piercing time
7	10 14 16 18 20 22 24 26 28	4 2 2 2 2 2 2 2 4	22	Feed–rate command Peak power command Pulse frequency command Pulse duty command Assist gas select command Assist gas settling time Assist gas pressure Reference displacement Offset amount
10	10	2	2	Ref. displacement command

E WINDOW FUNCTION DESCRIPTION (FS16–W)

E.1 READING THE WIRE DIAMETER OFFSET

[Description]

The wire diameter offset value recorded in the CNC can be read.

[Input data structure]

Address	Field	Offset	Corner–R	Clearance	Condition
Top address	(Function) 13				
+ 2	(Completion) —				
+ 4	(Data length) L				
+ 6	(Number) N	0–15	16	17	—
+ 8	(Attribute) M	0	0	0	1
+ 10	(Data area) —				

[Output data structure]

Address	Field	Offset	Corner–R	Clearance	Condition
Top address	(Function) 13				
+ 2	(Completion) ?				
+ 4	(Data length) L	4	4	4	8
+ 6	(Number) N	0–15	16	17	—
+ 8	(Attribute) M	0	0	0	1
+ 10	(Data area) D	Offset value	Corner–R value	Clearance value	Actual offset value
+ 14					Direction
+ 16					Offset mode
+ 18					

[Data number]

0–15 : Reads the Offset value.

16 : Reads the Corner–R value.

17 : Reads the Clearance value.

— : Reads the condition.

[Data attribute]

0 : Reads the Offset value, Corner–R value or Clearance value.

1 : Reads the condition.

[Contents of data]

a) Unit of Offset, Corner–R, Clearance and actual offset value

Metric system input : 10^{-3} [mm]

(In case the incremental system is 1/10, output data unit is 10^{-4} [mm].)

Inch system input : 10^{-5} [inch]

b) Direction in condition data

0 : Cancel offset (G40)

1 : Wire diameter compensation left (G41)

2 : Wire diameter compensation right (G42)

c) Offset mode in condition data

0 : Offset mode is 0.

1 : Offset mode is 1.

[Completion codes]

0 : The data has been read normally.

3 : Invalid data is specified as the data number.

4 : Invalid data is specified as the data attribute.

E.2 WRITING THE WIRE DIAMETER OFFSET (✻LOW–SPEED RESPONSE)

[Description]

The wire diameter offset value can be written into the CNC.

[Input data structure]

Address	Field	Offset	Corner–R	Clearance	Condition
Top address	(Function) 14				
+ 2	(Completion) —				
+ 4	(Data length) L	4	4	4	8
+ 6	(Number) N	0–15	16	17	—
+ 8	(Attribute) M	0	0	0	1
+ 10	(Data area) D	Offset value	Corner–R value	Clearance value	0, 1
+ 12					—
+ 14					

[Output data structure]

Address	Field
Top address	(Function) 14
+ 2	(Completion) ?
+ 4	(Data length) L
+ 6	(Number) N
+ 8	(Attribute) M
+ 10	(Data area) D

[Data number]

0–15 : Writes the Offset value.

16 : Writes the Corner–R value.

17 : Writes the Clearance value.

— : Writes the condition.

[Data attribute]

0 : Writes the Offset value, Corner–R value or Clearance value.

1 : Writes the condition.

[Contents of data]

a) Unit of Offset, Corner–R, Clearance and actual offset value

> **NOTE**
> Offset, Corner–R, Clearance or Actual offset value is signed binary in 4 bytes. A negative value is represented in 2's complement.

b) Offset mode in condition data

0 : Offset mode is 0.

1 : Offset mode is 1.

[Completion codes]

0 : The data has been written normally.

2 : Invalid data is specified as the data length.

3 : Invalid data is specified as the data number.

4 : Invalid data is specified as the data attribute.

5 : Invalid data is specified as the data value.

E.3 READING THE PARAMETER (∗LOW–SPEED RESPONSE)

[Description]

Parameter data in the CNC can be read.

There are four types of parameters in the CNC: Bit parameters having a definite meaning for each bit, byte parameters holding 1–byte data, word parameters holding 2–byte data, and double word parameters holding 4–byte data. Therefore, the length of the read data varies according to the parameter number specified.

Note that bit parameters cannot be read in bit units. The eighth bits (one byte) for a parameter number must be read at a time.

For axis parameters (servo parameters), data for a specific axis can be read, or data for all axes can be read at a time.

Specify pitch error compensation data in data Nos. 11000 to 18255.

[Input data structure]

Top address	(Function) 17
+ 2	(Completion) —
+ 4	(Data length) —
+ 6	(Number) N
+ 8	(Attribute) M
+ 10	(Data) —

[Output data structure]

Top address	(Function) 17
+ 2	(Completion) ?
+ 4	(Data length) L
+ 6	(Number) N
+ 8	(Attribute) M
+ 10	(Data) —

[Data length]

L = 1 or 1*n : Reads bit or byte type parameter.
2 or 2*n : Reads word type parameter.
4 or 4*n : Reads 2 words type parameter.
(Note: n is the axis number.)

[Data number]

N = (Parameter number)
or (Pitch error data number)+10000

[Data attribute]

M = 0 : Reads the no axis parameter.
1 to n : Reads the specific axis parameter
–1 : Reads the all axes parameter.
(Note: n is the axis number.)

[Completion code]

0 : Parameter data has been read normally.

2 : Invalid data is specified as the data length.

3 : Invalid data is specified as the data number.

4 : Invalid data is specified as the data attribute.

6 : Option is not provided.

E.4 WRITING THE PARAMETER (*LOW–SPEED RESPONSE)

[Description]

Parameter data in the CNC can be written.

There are four types of parameters in the CNC: Bit parameters having a definite meaning for each bit, byte parameters holding 1–byte data, word parameters holding 2–byte data, and double word parameters holding 4–byte data. Therefore, the length of the written data varies according to the parameter specified.

Note that bit parameters cannot be written in bit unit. The eighth bits (one byte) for the parameter number must be written at a time. This means that when a bit needs to be written, the whole data for the corresponding parameter number shall be read first, then the target bit in the read data shall be written.

For axis parameters (servo parameters), data for a specific axis can be written, or data for all axes can be written at a time.

Specify pitch error compensation data in data Nos. 11000 to 18255.

Some parameters cause a P/S alarm 000 when data is written. (The power must be turned off before continuing operation.)

[Input data structure]

Top address	(Function) 18
+ 2	(Completion) —
+ 4	(Data length) L
+ 6	(Number) N
+ 8	(Attribute) M
+ 10	(Data) Parameter data

[Output data structure]

Top address	(Function) 18
+ 2	(Completion) ?
+ 4	(Data length) L
+ 6	(Number) N
+ 8	(Attribute) M
+ 10	(Data) D

[Data length]

L = 1 or 1*n : Reads bit or byte type parameter.
2 or 2*n : Reads word type parameter.
4 or 4*n : Reads 2 words type parameter.
(Note: n is the axis number.)

[Data number]

N = (Parameter number)
or (Pitch error data number)+10000

[Data attribute]

M = 0 : Writes the no axis parameter.

1 to n : Writes the specific axis parameter

–1 : Writes the all axes parameter.

(Note: n is the axis number.)

[Completion code]

0 : Parameter data has been written normally.

2 : Invalid data is specified as the data length.

3 : Invalid data is specified as the data number.

4 : Invalid data is specified as the data attribute.

6 : Option is not provided.

[Types of parameters] **In the B908 system, data type of ( ) are used.**

No.	Length	No.	Data type	No.	Data type	No.	Data type
0000	Bit	0070	Byte	0140	Byte	0210	2W(—)
0001	Bit	0071	Byte	0141	Byte	0211	—
0002	Bit	0072	Byte	0142	Byte	0212	—
0003	Bit	0073	Byte	0143	Byte	0213	—
0004	Bit	0074	Byte	0144	Byte	0214	—
0005	Bit	0075	Byte	0145	Byte	0215	—
0006	Bit	0076	Byte	0146	Word	0216	—
0007	Bit	0077	Byte	0147	Word	0217	—
0008	Bit	0078	Byte	0148	Word	0218	—
0009	Bit	0079	Byte	0149	Word	0219	—
0010	Bit	0080	Byte	0150	Word	0220	—
0011	Bit	0081	Byte	0151	Word	0221	—
0012	Bit	0082	Byte	0152	Word	0222	—
0013	Bit	0083	Byte	0153	Word	0223	—
0014	Bit	0084	Byte	0154	Word	0224	—
0015	Bit	0085	Byte	0155	Word	0225	—
0016	Bit	0086	Byte	0156	Byte	0226	—
0017	Bit	0087	Byte	0157	Word	0227	—
0018	Bit	0088	Byte	0158	Word	0228	—
0019	Bit	0089	Byte	0159	Word	0229	—
0020	Bit	0090	Byte	0160	Word	0230	—
0021	Bit	0091	Byte	0161	Word	0231	—
0022	Bit	0092	Word	0162	Word	0232	—
0023	Bit	0093	Word	0163	Word	0233	—
0024	Bit	0094	Word	0164	2words	0234	—
0025	Bit	0095	Word	0165	Word	0235	—
0026	Bit	0096	Word	0166	2words	0236	—
0027	Bit	0097	Word	0167	Word	0237	—
0028	Bit	0098	Word	0168	Word	0238	—
0029	Bit	0099	Word	0169	Word	0239	—
0030	Bit	0100	Word	0170	Word	0240	—
0031	Bit	0101	Word	0171	Word	0241	—
0032	Bit	0102	Word	0172	Word	0242	—
0033	Bit	0103	Word	0173	Word	0243	—
0034	Bit	0104	Word	0174	Word	0244	—
0035	Bit	0105	Word	0175	Word	0245	—
0036	Bit	0106	Word	0176	Word	0246	—
0037	Bit	0107	Word	0177	Byte	0247	—
0038	Bit	0108	Word	0178	2W(Byte)	0248	—
0039	Bit	0109	Word	0179	2words	0249	—
0040	Byte	0110	Word	0180	Byte(2W)	0250	—
0041	Byte	0111	Word	0181	Byte	0251	—
0042	Word	0112	2words	0182	Byte	0252	—
0043	Word	0113	Word	0183	Byte	0253	—
0044	Word	0114	Word	0184	Byte	0254	—
0045	Word	0115	Word	0185	Byte	0255	—
0046	Byte	0116	Word	0186	Byte	0256	—
0047	Word	0117	Word	0187	Byte	0257	—
0048	Word	0118	Word	0188	Word	0258	—
0049	Word	0119	Word	0189	Byte	0259	—
0050	Word	0120	Word	0190	Byte	0260	—
0051	Word	0121	Word	0191	2words	0261	—
0052	Word	0122	Word	0192	2words	0262	—
0053	Word	0123	2words	0193	2words	0263	—
0054	Word	0124	Byte	0194	2words	0264	—
0055	Word	0125	Byte	0195	2words	0265	—
0056	Word	0126	Byte	0196	Word	0266	—
0057	Word	0127	Byte	0197	2words	0267	—
0058	Byte(—)	0128	Byte	0198	Byte	0268	—
0059	Byte(—)	0129	Byte	0199	2words	0269	—
0060	Byte	0130	Byte	0200	Byte	0270	—
0061	Byte	0131	Byte	0201	word	0271	—
0062	Byte	0132	Byte	0202	—	0272	—
0063	Byte	0133	Byte	0203	—	0273	—
0064	Byte	0134	Byte	0204	—	0274	—
0065	Byte	0135	Byte	0205	—	0275	—
0066	Byte	0136	Byte	0206	—	0276	—
0067	Byte	0137	Byte	0207	—	0277	—
0068	Word	0138	Byte	0208	—	0278	—
0069	2words	0139	Byte	0209	—	0279	—

No.	Length	No.	Data type	No.	Data type	No.	Data type
0280	—	0350	2words	0620	Byte	0690	Word
0281	—	0351	Byte	0621	Byte	0691	Word
0282	—	0352	Byte	0622	Byte	0692	Word
0283	—	0353	Byte	0623	Byte	0693	Word
0284	—	0354	Byte	0624	Word	0694	Word
0285	—	0355	Byte	0625	Word	0695	Word
0286	—	0356	Byte	0626	Word	0696	Word
0287	—	0357	Word	0627	Word	0697	Word
0288	—	0358	—	0628	Word	0698	Word
0289	—	0359	—	0629	Word	0699	Word
0290	—	0360	Word	0630	Word	0700	Word
0291	—	0361	2words	0631	Word	0701	Word
0292	—	0362	2words	0632	2words	0702	Word
0293	—	0363	2words	0633	2words	0703	Word
0294	—	0364	2words	0634	2words	0704	Word
0295	—	0365	2words	0635	2words	0705	Word
0296	—	0366	2words	0636	2words	0706	Word
0297	—	0367	2words	0637	2words	0707	Word
0298	—	0368	2words	0638	2words	0708	Word
0299	—	0369	2words	0639	2words	0709	Word
0300	Bit	0370	2words	0640	2words	0710	Word
0301	Bit	0371	2words	0641	2words	0711	Word
0302	Bit	0372	2words	0642	2words	0712	Word
0303	Bit	0373	2words	0643	2words	0713	Word
0304	Bit	0374	2words	0644	2words	0714	Word
0305	Bit	0375	2words	0645	2words	0715	Word
0306	Bit	0376	2words	0646	2words	0716	Word
0307	Bit	0377	2words	0647	2words	0717	Word
0308	Bit	0378	2words	0648	Word	0718	Word
0309	Bit	0379	2words	0649	Word	0719	Word
0310	Bit	0380	2words	0650	Word	0720	Word
0311	Bit	0381	2words	0651	Word	0721	Word
0312	Bit	0382	2words	0652	Word	0722	Word
0313	Bit	0383	2words	0653	Word	0723	Word
0314	Bit	0384	2words	0654	Word	0724	Word
0315	Bit	0385	—	0655	Word	0725	Word
0316	Word	0386	2words	0656	Word	0726	Word
0317	—	0387	2words	0657	Word	0727	Word
0318	—	0388	—	0658	Word	0728	Word
0319	—	0389	—	0659	Word	0729	Word
0320	Byte	0390	—	0660	Word	0730	Word
0321	Byte	0391	—	0661	Word	0731	Word
0322	Byte	0392	—	0662	Word	0732	Word
0323	Byte	0393	—	0663	Word	0733	Word
0324	Byte	0394	—	0664	Word	0734	Word
0325	Byte	0395	—	0665	Word	0735	Word
0326	Byte	0396	—	0666	Word	0736	Word
0327	Byte	0397	—	0667	Word	0737	Word
0328	Byte	0398	—	0668	Word	0738	Word
0329	Byte	0399	—	0669	Word	0739	Word
0330	Word	0600	Bit	0670	Word	0740	Word
0331	Word	0601	Bit	0671	Word	0741	Word
0332	2words	0602	Bit	0672	2words	0742	Word
0333	2words	0603	Bit	0673	2words	0743	Word
0334	Word	0604	Bit	0674	2words	0744	Word
0335	Word	0605	Bit	0675	2words	0745	Word
0336	2words	0606	Bit	0676	2words	0746	Word
0337	2words	0607	Bit	0677	2words	0747	2words
0338	2words	0608	Bit	0678	2words	0748	2words
0339	2words	0609	Bit	0679	2words	0749	2words
0340	2words	0610	Bit	0680	Word	0750	2words
0341	2words	0611	Bit	0681	Word	0751	2words
0342	2words	0612	Bit	0682	Word	0752	2words
0343	Byte	0613	Bit	0683	Word	0753	2words
0344	Word	0614	Bit	0684	Word	0754	2words
0345	Byte	0615	Bit	0685	Word	0755	Word
0346	Word	0616	Byte	0686	Word	0756	Word
0347	Byte	0617	Byte	0687	Word	0757	Word
0348	2words	0618	Byte	0688	Word	0758	2words
0349	2words	0619	Byte	0689	Word	0759	2words

No.	Length	No.	Data type	No.	Data type	No.	Data type
0760	2words	0810	2words	0860	Word	0910	—
0761	2words	0811	Byte	0861	Bit	0911	—
0762	2words	0812	Byte	0862	Bit	0912	—
0763	2words	0813	Byte	0863	Byte	0913	—
0764	2words	0814	Byte	0864	Byte	0914	—
0765	2words	0815	Byte	0865	Byte	0915	—
0766	2words	0816	Byte	0866	Byte	0916	—
0767	2words	0817	Byte	0867	Byte	0917	—
0768	2words	0818	Byte	0868	Byte	0918	—
0769	2words	0819	Word	0869	Byte	0919	—
0770	2words	0820	Word	0870	Byte	0920	—
0771	2words	0821	Word	0871	Byte	0921	—
0772	2words	0822	Word	0872	Word	0922	—
0773	2words	0823	Word	0873	Word	0923	—
0774	2words	0824	Word	0874	Word	0924	—
0775	2words	0825	Word	0875	Word	0925	—
0776	2words	0826	Word	0876	Word	0926	—
0777	2words	0827	Word	0877	Word	0927	—
0778	2words	0828	Word	0878	Word	0928	—
0779	2words	0829	Word	0879	Word	0929	—
0780	2words	0830	Word	0880	Word	0930	—
0781	2words	0831	Word	0881	Word	0931	—
0782	2words	0832	Word	0882	Word	0932	—
0783	2words	0833	Word	0883	Word	0933	—
0784	2words	0834	Word	0884	Word	0934	—
0785	2words	0835	—	0885	Word	0935	—
0786	2words	0836	Word	0886	Word	0936	—
0787	2words	0837	Word	0887	Word	0937	—
0788	2words	0838	Word	0888	Word	0938	—
0789	2words	0839	Word	0889	Word	0939	—
0790	2words	0840	Word	0890	Word	0940	—
0791	2words	0841	Word	0891	Word	0941	—
0792	2words	0842	Word	0892	Word	0942	—
0793	2words	0843	Word	0893	Word	0943	—
0794	2words	0844	Word	0894	Word	0944	—
0795	2words	0845	Word	0895	Word	0945	—
0796	2words	0846	Word	0896	Word	0946	—
0797	2words	0847	Word	0897	Word	0947	—
0798	2words	0848	Word	0898	Word	0948	—
0799	2words	0849	Word	0899	—(Word)	0949	—
0800	2words	0850	Word	0900	—(Bit)		
0801	2words	0851	Word	0901	—(Bit)		
0802	2words	0852	Word(2W)	0902	—(Bit)		
0803	2words	0853	Word(2W)	0903	—		
0804	2words	0854	Word(2W)	0904	—(Byte)		
0805	2words	0855	Word(2W)	0905	—(Byte)		
0806	2words	0856	Word(2W)	0906	—		
0807	2words	0857	Word(2W)	0907	—		
0808	2words	0858	Word(2W)	0908	—		
0809	2words	0859	Word(2W)	0909	—		

E.5 READING SETTING DATA

[Data contents]

Setting data on the CNC can be read.

	[Input data structure]		[Output data structure]
Top address	(Function code) 19	Top address	(Function code) 19
+ 2	(Completion code) –	+ 2	(Completion code) ?
+ 4	(Data length) –	+ 4	(Data length) L
+ 6	(Data number) N	+ 6	(Data number) Input data
+ 8	(Data attribute) –	+ 8	(Data attribute) –
+ 10	(Data area) –	+ 10	(Data area) Setting data

[Data number]

See the setting data list.

[Completion code]

0 : Read operation was terminated normally.

3 : An incorrect data number was specified.

E.6 WRITING SETTING DATA (LOW–SPEED TYPE)

[Data contents]

Setting data on the CNC can be written.

[Input data structure]

Top address	(Function code) 20
+ 2	(Completion code) –
+ 4	(Data length) L
+ 6	(Data number) N
+ 8	(Data attribute) –
+ 10	(Data area) Setting data

[Output data structure]

Top address	(Function code) 20
+ 2	(Completion code) ?
+ 4	(Data length) Input data
+ 6	(Data number) Input data
+ 8	(Data attribute) –
+ 10	(Data area) Input data

[Data length]

See the setting data list.

[Data number]

See the setting data list.

[Completion code]

0 : Read operation was terminated normally.

2 : An incorrect data length was specified.

3 : An incorrect data number was specified.

Setting Data List

Setting data	Screen	Data number	Data length	Bit name
X mirror image	Handy	1	1	Bit 0
Y mirror image	Handy	1	1	Bit 1
Axis switching	Handy	1	1	Bit 2
TV check	Handy	1	1	Bit 3
Output code	Handy	1	1	Bit 4
Input unit	Handy	1	1	Bit 5
Parameter writable	Handy	1	1	Bit 6
Input unit multiplication by 10 times	Handy	1	1	Bit 7
Automatic recovery from power failure	Handy	2	1	Bit 3
Automatic power–off M20/M30	Handy	3	1	Bit 0
Automatic power–off M00/M10	Handy	3	1	Bit 1
Automatic power backward movement alarm	Handy	3	1	Bit 2
Automatic power disconnection	Handy	3	1	Bit 3
Input/output device	Handy	4	1	–
Figure magnification ratio	Handy	5	4	–
Figure rotation angle	Handy	6	4	–
Taper machining mode	Taper	15	1	–
Guide type	Taper	16	1	Bit 0
Program surface position	Taper	17	4	–
Workpiece thickness	Taper	18	4	–
Drawing surface position	Taper	19	4	–
Upper guide position	Taper	20	4	–
(Lower guide position)	Taper	21	4	–
(Vertical position U)	Taper	22	4	–
(Vertical position V)	Taper	23	4	–
Enable/disable	PWB	2	1	–
Wire diameter	PWB	7	4	–
Machining groove width	PWB	8	4	–
Workpiece thickness	PWB	9	4	–
Wire deflection	PWB	10	4	–
Effectiveness of concave	PWB	11	4	–
Effectiveness of convex	PWB	12	4	–
Automatic override				
Enable/disable	PWB	2	1	Bit 5
Differential voltage	PWB	13	2	–
Enable/disable	AWF	2	1	Bit 2
Disconnection repair	AWF	2	1	Bit 1
Prepared hole of 0.5	AWF	2	1	Bit 6
Sump machining	AWF	2	1	Bit 7
Portion to be left uncut	AWF	14	1	–
Power reduction ratio (setting)	AWF	24	1	–
Connection position U1	AWF	26	4	–
Connection position V1	AWF	27	4	–
Connection position Z1	AWF	28	4	–
Number of retries (setting)	AWF	40	1	–
Number of allowable disconnections (setting)	AWF	41	1	–
Number of retries (current)	AWF	42	1	–
Number of allowable disconnections (current)	AWF	43	1	–
WIRE REST	LIFE	30	2	–
WIRE CUTTER REST	LIFE	31	2	–
( ) REST	LIFE	32	2	–
( ) REST	LIFE	33	2	–
( ) REST	LIFE	34	2	–
( ) REST	LIFE	35	2	–
( ) REST	LIFE	36	2	–
( ) REST	LIFE	37	2	–

Setting Data List

Setting data	Screen	Data number	Data length	Bit name
OP. BLOCK SKIP /0	Others	38	1	Bit 0
OP. BLOCK SKIP /1	Others	38	1	Bit 1
OP. BLOCK SKIP /2	Others	38	1	Bit 2
OP. BLOCK SKIP /3	Others	38	1	Bit 3
OP. BLOCK SKIP /4	Others	38	1	Bit 4
OP. BLOCK SKIP /5	Others	38	1	Bit 5
OP. BLOCK SKIP /6	Others	38	1	Bit 6
OP. BLOCK SKIP /7	Others	38	1	Bit 7
OP. BLOCK SKIP /8	Others	39	1	Bit 0
OP. BLOCK SKIP /9	Others	39	1	Bit 1
Number or tries	AWF	40	1	–
Number of retries	AWF	41	1	–
Number or tries	AWF	42	1	–
Number of retries	AWF	43	1	–
Program number (for machining distance calculation)	Graphic	44	2	–

E.7 READING THE CNC ALARM STATUS

[Description]

When the CNC is placed in the alarm status, the alarm status data can be read.

[Input data structure]

Top address	(Function) 23
+ 2	(Completion) —
+ 4	(Data length) —
+ 6	(Number) —
+ 8	(Attribute) —
+ 10	(Data area) —

[Output data structure]

Top address	(Function) 23	
+ 2	(Completion) 0	
+ 4	(Data length) 2	
+ 6	(Number) —	
+ 8	(Attribute) —	
+ 10 + 11	(Data area) Alarm status	First byte
+ 12		Second byte

[Contents of data]

(1) Alarm status data in first byte.

#7	#6	#5	#4	#3	#2	#1	#0
EOR	OTM	OTS	OH		SV	OTH	PS

PS : P/S alarm

OTH : Over travel alarm

SV : Servo alarm

OH : Overheat alarm

OTS : First stroke limit alarm

OTM : Second stroke limit alarm

EOR : Edit alarm

(2) Alarm status data in second byte.

#7	#6	#5	#4	#3	#2	#1	#0
							APCER

APCER : Absolute pulco alarm

[Completion codes]

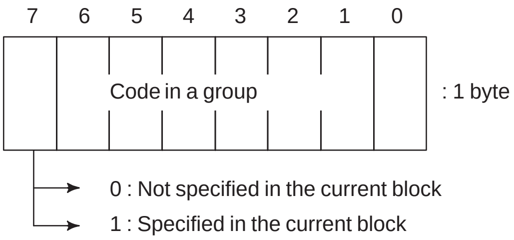
0 : This alarm status in the CNC has been read normally.

E.8 READING MODEL DATA

[Description]

Modal information in the CNC can be read.

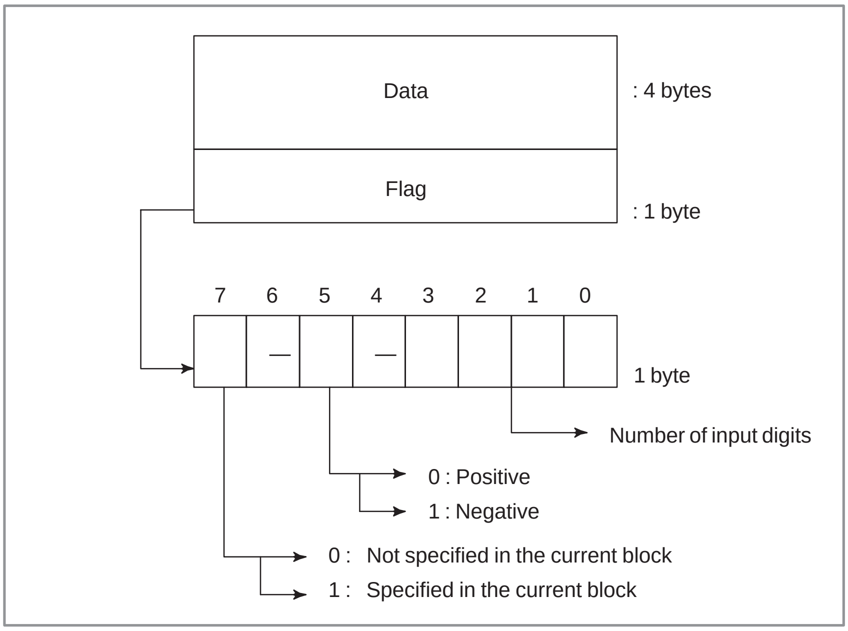
(1) Format and types of modal data for the G function



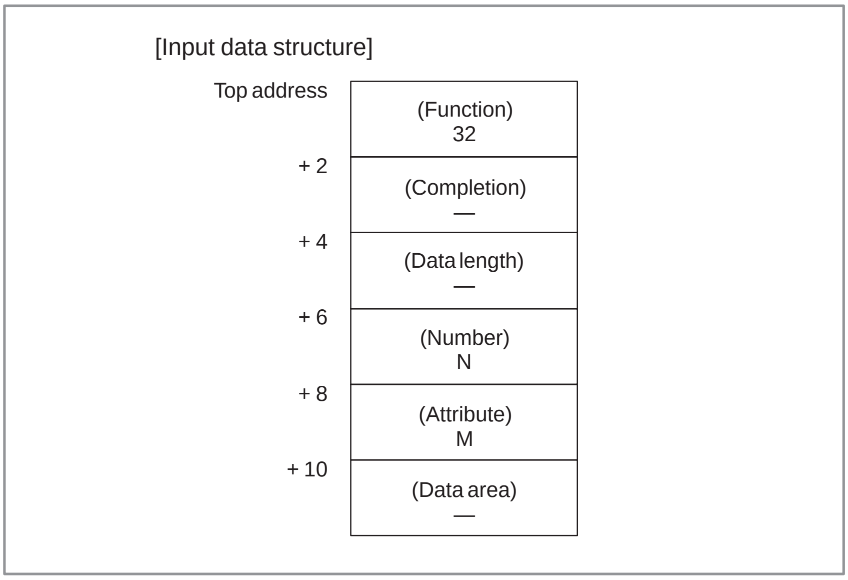
Identification code	Data type	Data
0	G04	0
	G19	1
	G28	5
	G30	7
	G92	14
	G31	15
	G70	16
	G71	17
	G72	18
	G73	19
	G74	20
	G75	21
	G76	22
	G77	23
	G78	24
	G79	25
1	G00	0
	G01	1
	G02	2
	G03	3
2	G17	0
3	G90	0
	G91	1
4	G22	1
	G23	0

Identification code	Data type	Data
5	G94	0
	G95	1
6	G20	0
	G21	1
7	G40	0
	G41	1
	G42	2
8	G50	0
	G51	1
	G52	2
9	G60	0
	G61	1
	G62	2
	G63	3
10	G48	1
	G49	0
11	G65	26
	G66	0
	G67	1

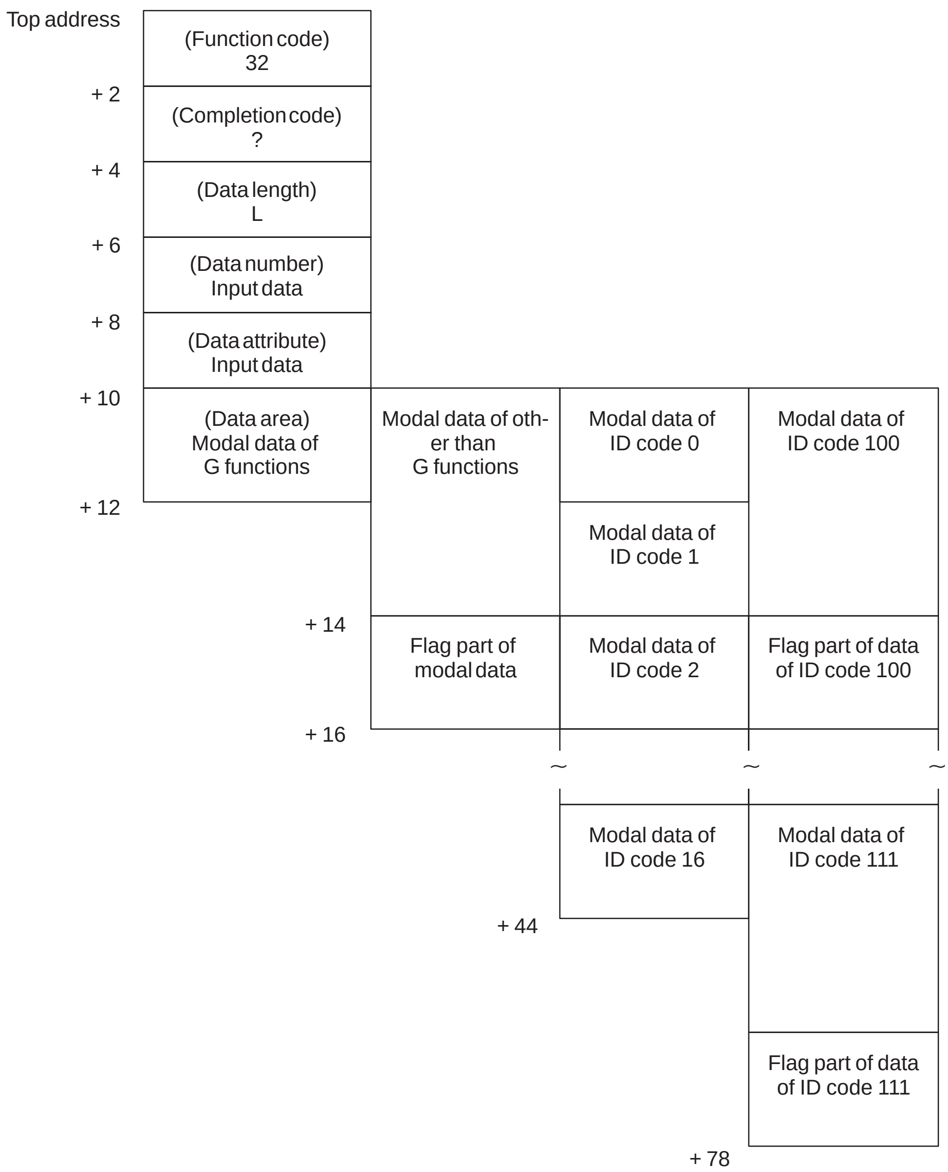
(2) Format and types of modal data for other than the G function



Identification code	Specified address	Meaning of value
100	B	
101	D	
102	E	Offset number
103	F	
104	H	Feedrate
105	L	
106	M	
107	S	
108	T	Tapper data
109	R	
110	P	
111	Q	



[Output data structure]



[Data length]

L = 2 : G function
2* : All data of G functions
6 : Other than G functions
6*12 : All data of other than G functions

[Data number]

N = 0 and up: See each data list.
–1 : All data of G functions
–2 : All data of other than G functions

[Data attribute]

M = 0 : Current block
1 : Next block
2 : Block after the next block

[Completion code]

–1 : The modal data read command could not be executed. This means that, because the modal data was being updated on the CNC, the command could not be executed.

0 : Modal data read operation terminated normally.

3 : An incorrect data number was specified.

4 : An incorrect data attribute was specified.

E.9 READING MACHINING DISTANCES

[Data contents]

The distance (machining distance) from the machining start point to the current point, and the distance (whole distance) from the machining start point to the point where M02 or M30 is specified can be read.

[Input data structure]

Address	Contents
Top address	(Function code) 182
+ 2	(Completion code) –
+ 4	(Data length) –
+ 6	(Data number) N
+ 8	(Data attribute) –
+ 10	(Data area) –

[Output data structure]

Address	Contents
Top address	(Function code) 182
+ 2	(Completion code) ?
+ 4	(Data length) 4
+ 6	(Data number) Input data
+ 8	(Data attribute) –
+ 10	(Data area) Machining distance
+ 14	

[Data number]

N = 0 : Machining distance

N = 1 : Whole length obtained with the dry run function

N = 2 : Whole length obtained by drawing

N = 3 : Whole length obtained with the machining distance calculation function

[Data unit]

Metric input : 10^{-3} [mm]
(When the increment system is 1/10: 10^{-4} [mm])

Inch input : 10^{-5} [inch]

[Completion code]

–1 : Data could not be read. This means that, because the data was being updated on the CNC, the command could not be executed.

0 : Data read operation terminated normally.

3 : An incorrect data number was specified.

E.10 READING THE MEASURED POINT

[Description]

The measured point that are get by positioning can be read. Also, the slit width by slitting and the hole diameter by centering can be read.

[Input data structure]

		Reads measured point	Reads slit width of hole diameter
Top address	(Function) 185		
+ 2	(Completion) —		
+ 4	(Length) —	Reads measured point	Reads slit width of hole diameter
+ 6	(Number) N	Point number	0
+ 8	(Attribute) M	0	1
+ 10	(Data area) —		

[Output data structure]

		Reads measured point	Reads slit width of hole diameter
Top address	(Function) 185		
+ 2	(Completion) ?	Reads measured point	Reads slit width of hole diameter
+ 4	(Length) L	10	4
+ 6	(Number) N	Point number	0
+ 8	(Attribute) M	0	1
+ 10	(Data area)	Type	Slit width of hole diameter
+ 12		Machine coordinate of X axis	
+ 14			
+ 16		Machine coordinate of Y axis	
+ 18			
+ 20			

[Data unit]

Metric system input : 10^{-3} [mm]

(In case the incremental system is 1/10, output data unit is 10^{-4} [mm].)

Inch system input : 10^{-5} [inch]

[Completion codes]

0 : The measured point has been read normally.

3 : Invalid data is specified as data number.

4 : Invalid data is specified as data attribute.

E.11 WRITING THE MEASURED POINT (*LOW–SPEED RESPONSE)

[Description]

The measured point that are get by positioning can be written. Also, the slit width by slitting and the hole diameter by centering can be written.

[Input data structure]

Address	Content
Top address	(Function) 186
+ 2	(Completion) —
+ 4	(Length) 10
+ 6	(Number) Point number
+ 8	(Attribute) 0
+ 10	(Data area) Type
+ 12	Machine coordinate of X axis
+ 14	
+ 16	Machine coordinate of Y axis
+ 18	
+ 20	

[Output data structure]

Address	Content
Top address	(Function) 186
+ 2	(Completion) ?
+ 4	(Length) 10
+ 6	(Number) N
+ 8	(Attribute) 0
+ 10	(Data area)
+ 12	
+ 14	
+ 16	
+ 18	
+ 20	

[Data unit]

Metric system input : 10^{-3} [mm]

(In case the incremental system is 1/10, output data unit is 10^{-4} [mm].)

Inch system input : 10^{-5} [inch]

[Completion codes]

0 : The measured point has been written normally.

3 : Invalid data is specified as data number.

4 : Invalid data is specified as data attribute.

F WINDOW FUNCTION DESCRIPTION (FS16–PA)

F.1 READING OF TOOL SETTING DATA

[Description]

Various Tool setting data recorded in the CNC can be read.

[Input data structure]

Address	Contents
Top address + 0	(Function code) 188
2	(Completion code) — (Need not be set)
4	(Data length) — (Need not be set)
6	(Data Number) N (See 1.1)
8	(Data attribute) M (See 1.1)
10	(Data area) — (Need not be set)

[Completion codes]

0 : The tool setting data has been read normally.

3 : The data number specified for reading is invalid.

4 : There are mistakes in the data attribute that specifies the type of the tool setting data to be read.

6 : For the tool setting data specified for reading, an additional option (graphic or multi–tool control) is required, but it is missing.

[Output data structure]

Top address + 0	(Function code) 188
2	(Completion code) ? (See the explanation of codes)
4	(Data length) ? (See 1.1)
6	(Data Number) N (N : Input data)
8	(Data attribute) M (M : Input data)
10	(Data area) ? (See 1.1) ~
48	

F.1.1 Data Number, Data Attribute, Data Length, Data Area

Data number, data attribute, data length and data area of various tool setting data are as follows.

Various tool setting data	Data number (N)	Data attribute (M)	Data length	Data area
Used tool number	0	0	2 bytes	Binary 1 to 136
Number of turret indexing		1	2 bytes	Binary 1 to 136
Tool number of reference point		2	2 bytes	Binary 1 to 136
Feed amount per revolution of turret		3	4 bytes	Binary 1 to 99999999
Total punch count		4	8 bytes	Binary 1 to 99999999
Tool number	Number of tool setting data 1 to 136	0	2 bytes	Binary 1 to 9999
Punch count		1	4 bytes	Binary 1 to 99999999
Tool position compensation of X		2	4 bytes	Binary ±99999999
Tool position compensation of Y		3	4 bytes	Binary ±99999999
Machine position of tool		4	4 bytes	Binary ±99999999
Tool number for tool change		5	2 bytes	Binary 1 to 136
(Not used)		6	4 bytes	
Tool shape (C) for graphic		7	2 bytes High byte=0	Binary 0 to 4
Tool shape (I) for graphic		8	4 bytes	Binary 0 to 999999
Tool shape (J) for graphic		9	4 bytes	Binary 0 to 999999
Tool shape (K) for graphic		10	4 bytes	Binary 0 to 360000
Tool number for multi–tool	Number of Multi–tool setting data +200 201 to 264	0	2 bytes High byte=0	Binary 0 to 99
Tool angle for multi–tool		1	4 bytes	Binary ±360000
Tool position compensation of Y		2	4 bytes	Binary ±99999999
Tool shape (C) for multi–tool		3	2 bytes High byte=0	Binary 0 to 4
Tool shape (I) for multi–tool		4	4 bytes	Binary 0 to 999999
Tool shape (J) for multi–tool		5	4 bytes	Binary 0 to 999999
Tool shape (K) for multi–tool		6	4 bytes	Binary 0 to 360000

data unit

	Machine	Input of IS–A	Input of IS–B
Tool position compensation	mm	0. 01	0. 001
	inch	0. 001	0. 0001

	Input unit	Input of IS–A	Input of IS–B
Tool shape and angle for graphic	mm	0. 01	0. 001
	inch	0. 001	0. 0001
Tool angle for multi–tool	deg	0. 01	0. 001

F.2 WRITING OF TOOL SETTING DATA (LOW–SPEED RESPONSE)

[Description]

The various tool setting data can be directly written into the CNC.

[Input data structure]

Top address + 0	(Function code) 189
2	(Completion code) — (Need not be set)
4	(Data length) ? (See 1.1)
6	(Data Number) N (See 1.1)
8	(Data attribute) M (See 1.1)
10	(Data) ? (See 1.1)

[Completion code]

0 : The tool setting data has been written normally.

2 : The data length specified for writing is invalid.

3 : The data number specified for writing is invalid.

4 : The data attribute specified for writing is invalid.

5 : The data specified for writing is invalid.

6 : The additional option (multi–tool control or graphic) is required but it is missing.

[Output data structure]

Address	Content
Top address + 0	(Function code) 189
2	(Completion code) ? (See the explanation of codes)
4	(Data length) ? (See F.1)
6	(Data Number) N (N : Input data)
8	(Data attribute) M (M : Input data)
10	(Data area) ? (See F.1) ∼
48	

NOTE
See Sec. F. 1 for data unit.

F.3 READING TOOL SETTING DATA BY SPECIFYING TOOL NUMBER

[Description]

Setting data for a tool (such as registration order, tool punch count, and tool shape) can be read by specifying the tool number.

[Input data structure]

Top address + 0	(Function code) 141
2	(Completion) – (Need not be set)
4	(Data length) – (Need not be set)
6	(Data number) N (N=Tool number)
10	(Data attribute) M (See F.1)
12	(Data area) – (Need not be set)

NOTE

1 The area for specifying the data number consists of four bytes.

2 As the data attribute, specify the type of the tool setting data to be read, in the same way as for function code 188. If 0 is specified as the data attribute, the registration order of the tool is read.

[Completion code]

0: The tool setting data has been read normally.

3: The specified data number is invalid.

4: The specified data attribute is invalid.

6: For the tool setting data specified for reading, an additional option (graphic or multi–tool control) is required, but it is missing.

[Output data structure]

Top address + 0	(Function code) 141
2	(Completion code) ? (See the explanation above)
4	(Data length) ? (See F.1)
6	(Data number) N (N: Input data)
10	(Data attribute) M (M: Input data)
12	(Data area) ? (See F.1)
48	

F.4 OTHER WINDOW FUNCTIONS

The FS16–PA supports the following window functions, described in this manual.

Number			Function code
1	Reading CNC system information		0
2	Reading a tool offset		13
3	Writing a tool offset	*low–speed response	14
4	Reading a workpiece origin offset		15
5	Writing a workpiece origin offset	*low–speed response	16
6	Reading a parameter	*low–speed response	17
7	Writing a parameter	*low–speed response	18
8	Reading setting data	*low–speed response	19
9	Writing setting data	*low–speed response	20
10	Reading a custom macro variable	*low–speed response	21
11	Writing a custom macro variable	*low–speed response	22
12	Reading the CNC alarm state		23
13	Reading the current program number		24
14	Reading the current sequence number		25
15	Reading an actual velocity for a controlled axis		26
16	Reading an absolute position on a controlled axis		27
17	Reading a machine position on a controlled axis		28
18	Reading a skip position on a controlled axis		29
19	Reading a servo delay amount on a controlled axis		30
20	Reading an acceleration/deceleration delay amount on a controlled axis		31
21	Reading modal data		32
22	Reading diagnostic data	*low–speed response	33
38	Reading clock data (date and time)		151
41	Reading a parameter		154
42	Reading setting data		155
43	Reading diagnostic data		156
44	Reading a character string of the CNC program being executed in the buffer		157
45	Reading the relative position on a controlled axis		74
46	Reading the remaining travel on a controlled axis		75
47	Reading CNC status information		76
48	Reading an operator message		83

G SIGNAL ADDRESS CONVERSION (FROM THE PMC–MODEL L/M TO THE PMC–MODEL SB/SC)

G.1 GENERAL

DI/DO signals used in the PMC–MODEL L/M can be converted to signals for the PMC–MODEL SB/SC using the FAPT LADDER program for the PMC–MODEL SB/SC.

G.2 FUNCTION

The DI/DO signals used between the NC unit and the PMC correspond to word addresses consisting of addresses and values. Word addresses of bit type are converted. The program is not logically converted.

The conversion is performed under the following conditions.

(1) A word address of bit type used in a basic instruction is to be converted.

(2) A word address of byte type used in a functional instruction is not converted.

(3) Word addresses used in the standard FANUC Series 0–T/M are converted to those used in the standard FANUC Series 16–T/M. If a value in a word address is 1000.0 or more, the address is not converted.

(4) When the same signal name is used in the FANUC Series 0 and 16, and the addresses corresponding to the signal in the Series 0 and 16 have one–to–one relationship, the word address is converted. For details, see the signal conversion table.

G.3 CONVERSION

(1) Load the FAPT LADDER program for the PMC–SB/SC.

(2) Press the R0 key to display the programmer menu screen.

(3) Press the F2 key. Enter 2 and press the <NL> key. then the following message appears on the screen. Insert a data floppy for the PMC–SB/SC. Select the name of the file corresponding to the conversion from Table G.3 and enter it.

SET FD & KEYIN ‘OK’, ‘KILL’ OR ‘NO’
FD0 = OK<DRIVE> <@NAME OR :NUMBER>
FD0 =

Table G.3 File Name in the Data Floppy for the PMC–RB/RC

	File name
FS0–T → FS16–T	COMV.FS0–T
FS0–M → FS16–M	COMV.FS0–M

(4) Read a source ladder program created with FAPT LADDER for the PMC–L/M from the floppy in the same way as in Item 3.

If an address not listed in the signal conversion table is used in the ladder program file, an error occurs. In this case, enter E, then press the <NL> key to return to the programmer menu screen.

PART= E <NL>

(5) Entering 9 and pressing the <NL> key on the programmer menu screen changes the screen. The following message appears at the lower left corner of the screen. Enter 2, then press the <NL> key to delete the symbol data.

KEYIN ’1, 2, 3, 4, 5 OR 6 OR ‘NO’
CLEAR/KEEP=

G.4 MODIFYING THE CONVERTED SEQUENCE PROGRAM

The above operation terminates the conversion. Check the converted program. If an error occurs in the conversion, modify the program.

Enter 1 on the programmer menu to change the screen to the screen for editing a sequence program. Editing operation is the same as usual.

> **NOTE**
> Some addresses not converted have no error indication. After modifying the program, check that all addresses are correct according to the signal conversion table and the connecting manual.

G.4.1 Modification Procedure

(1) When the Series 0 and 16 differ in the number of parameters used in a functional instruction

Because the Series 0 and 16 differ in the numbers of parameters used for TMR (timer), TMRB (timer), and CTR (counter), errors are indicated at the parameters. Check the program, then delete the parameter. Set the timer and counter again.

(2) When an address not used in functional instructions is specified

When an address used in the ladder program for the Series 0 is not defined in the Series 16, the messages (NO PARAMETER) and #PARAM.ERROR# appear as follows. Set the parameter again and delete the latter message.

Example

```
00001 RD XXX.X
00002 SUB 8
00003 XXXX
00004 XXXX
00005 XXXX
00006 (NO PARAMETER)  . .  Set the parameter again.
00007 #PARAM.ERR#  . . . . .  Delete the message. This
                              message may not appear.
(XXX.X and XXXX are addresses and values.)
```

(3) Deleting SUB48 (END3) (In the PMC–SB)

If SUB48 (END3) is specified in the PMC–SB, an error occurs because the PMC–SB is not provided with SUB48 (END3). When this error occurs, delete third–level programs, or change the third–level programs to second–level programs and delete SUB48.

(4) Address conversion for signals not listed on the signal conversion table Modify the address for a signal by referring to the connection manual.

H CONNECTING THE OPERATOR'S PANEL FOR FS 0 WITH FS16, FS18, FS21, OR Power Mate

H.1 GENERAL

The Series 0 operator's panel consists of key switches, LEDs, a rotary switch, and so on. Because the states of key switches and lamps are coded, the number of the signal lines required for connecting the operator's panel with the CNC may not be the same as the number of actual switches. PMC management software automatically codes the states of the key switches and lamps and transmits data.

Therefore, simple bit images of switches and LEDs must only be manipulated with the PMC ladder program.

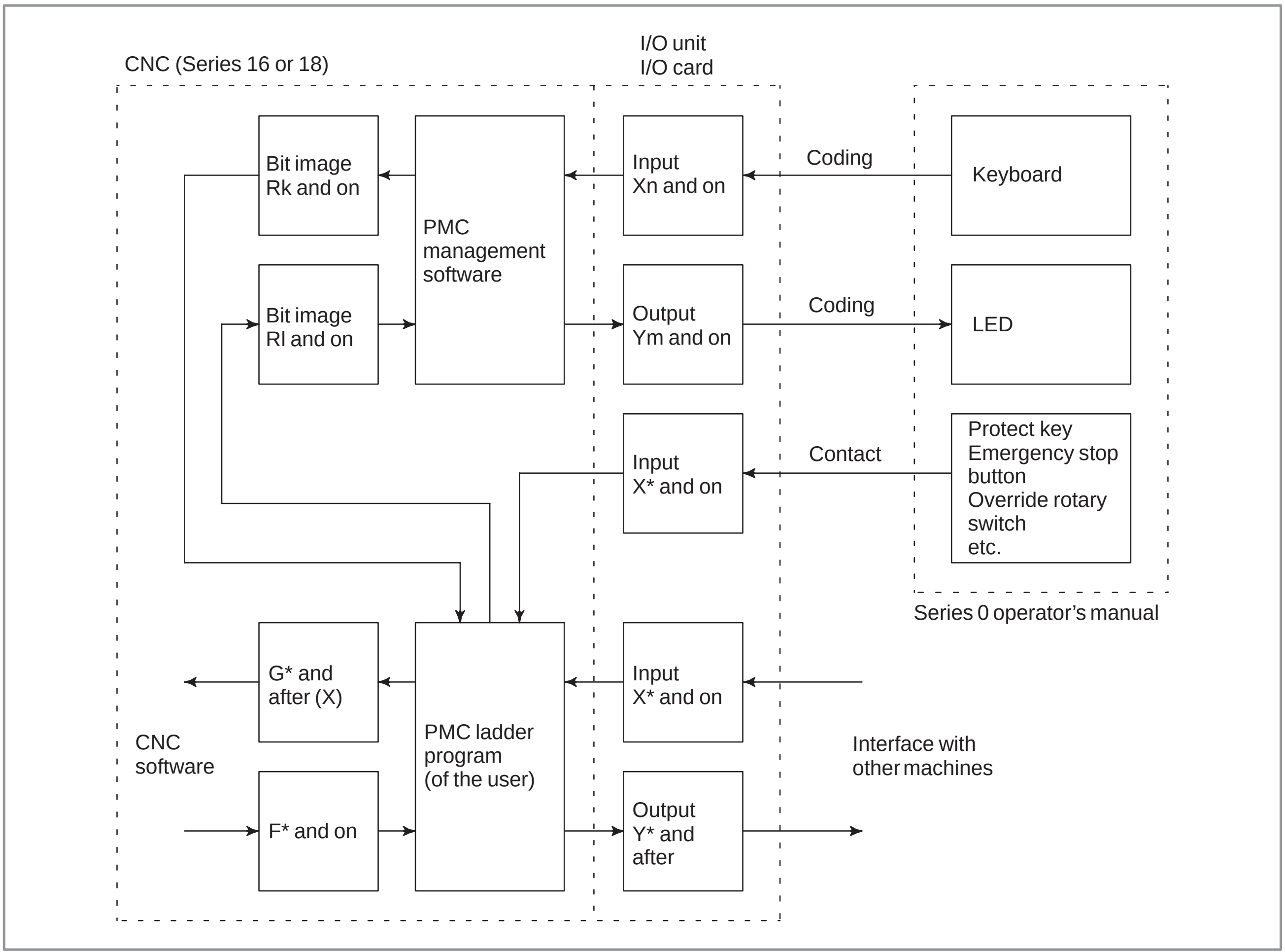


Fig. H.1 (a) Connection between the CNC and the operator's panel

The operator's panel is made up of the following keys, LEDs, etc.

- Key switch (Seat key)
- 42 keys (0–TB)
- 46 keys (0–MB)
- LEDs (red) ....Prepared for all key switches
- Override rotary switch ..... 4 bits
- Emergency stop button .... 1 bit
- Program protect key .... 1 bit

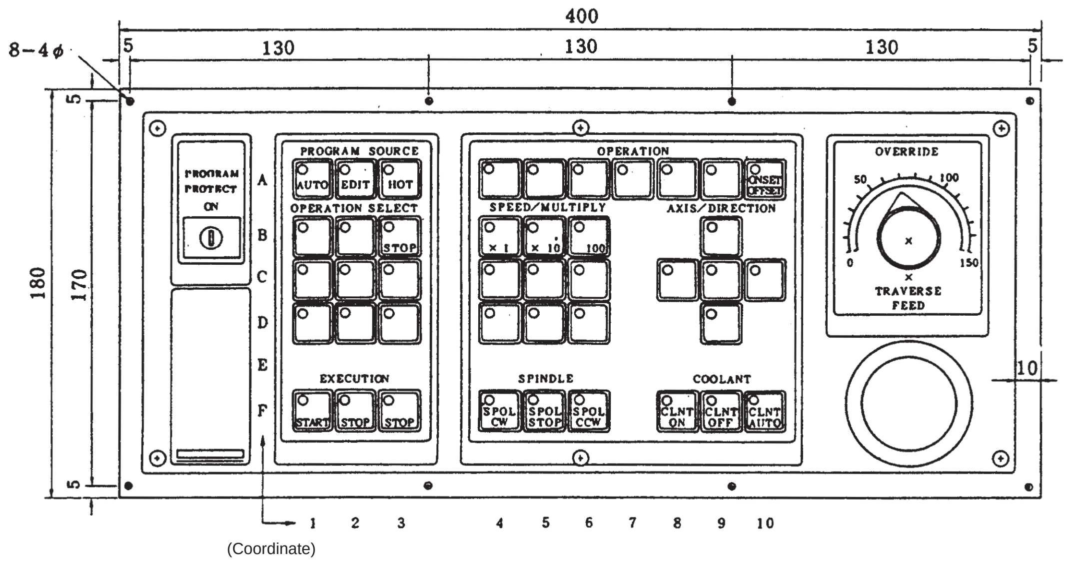


Fig. H.1 (b) Front view of operator's panel for 0–TC

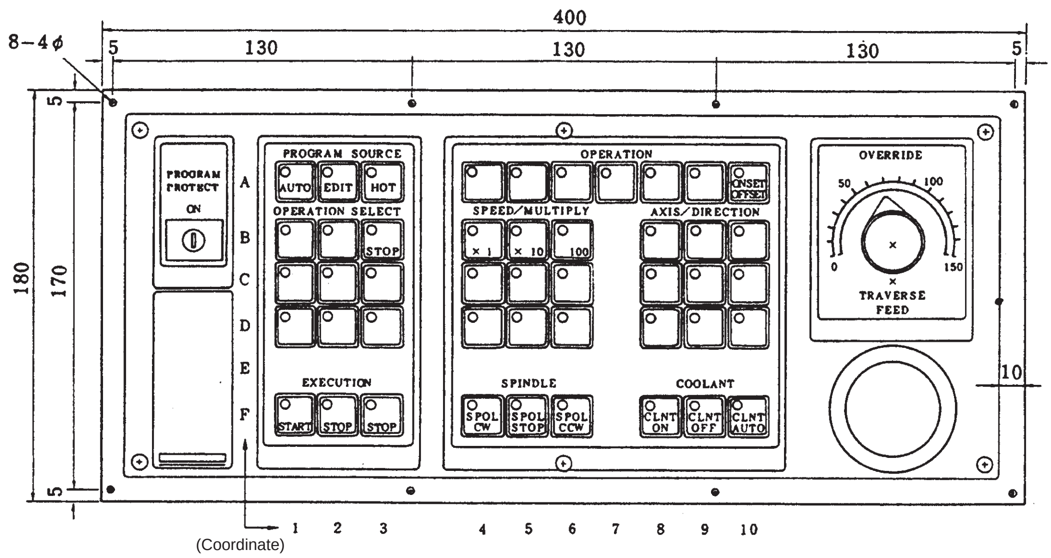


Fig. H.1 (c) Front view of operator's panel for 0–MC

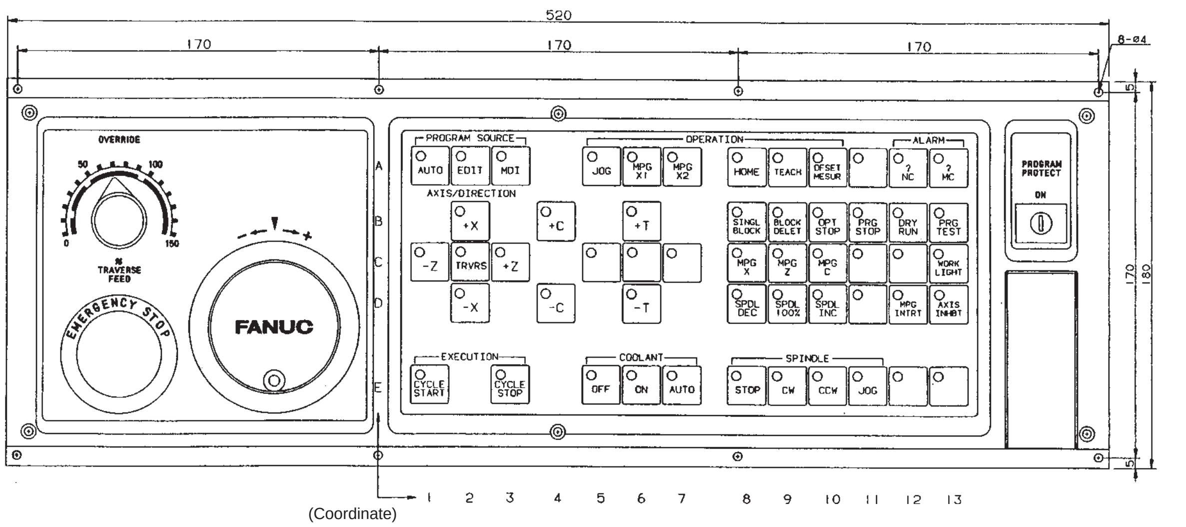


Fig. H.1 (d) External view of operator's panel for 9″ CRT/MDI with full–keyboard (0–TC)

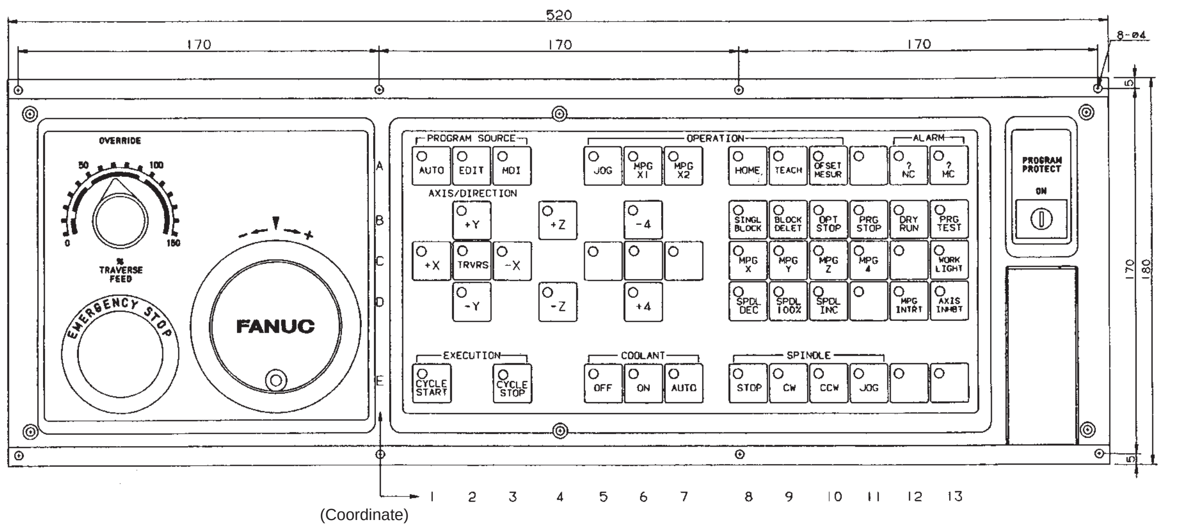
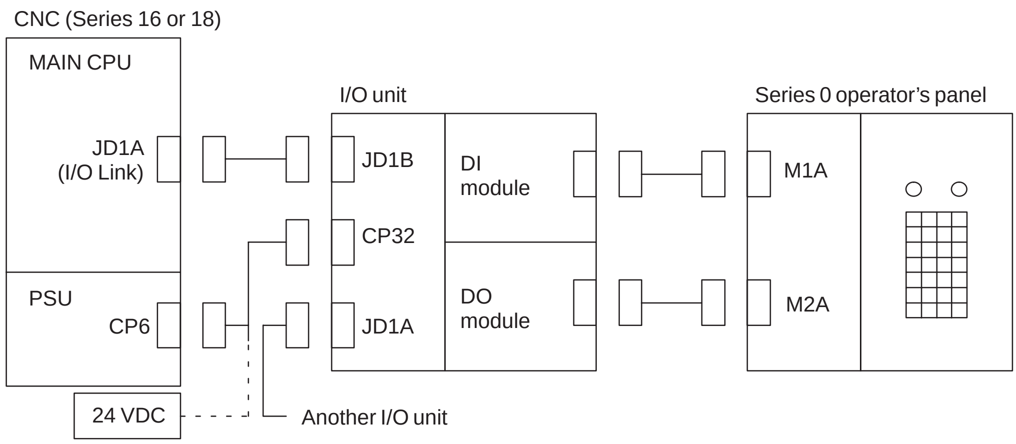


Fig. H.1 (e) External view of operator's panel for 9″ CRT/MDI with full–keyboard (0–MC)

H.2 CONNECTION

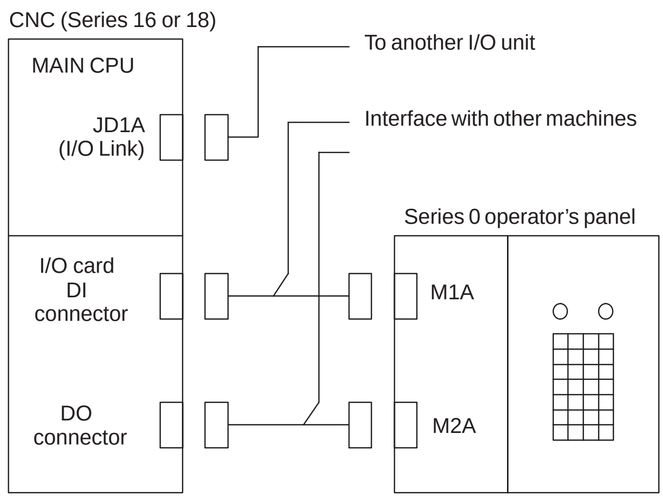
H.2.1 Connecting the I/O Unit



DI module : +24 V common, 20 ms
Example) AID32A1

DO module : 0 V common
Example) AOD32A1

H.2.2 Connecting the I/O Card



H.3 SIGNALS FOR CONNECTING THE OPERATOR'S PANEL

H.3.1 Emergency Stop Signal (*ESP)

This signal is used for the fixed address directly monitored by the CNC.

For connecting the signal, refer to the description of the interface between the CNC and the PMC in the "Series 16 or 18 Connection Manual."

H.3.2 Override Signals (*OV1 to *OV8) and Program Protect Key Signal (KEY)

Their key switch contact signals are directly input to the PMC. Handle them with the PMC ladder program.

For connecting these signals, refer to the description of the interface between the CNC and the PMC in the "Series 16 or 18 Connection Manual."

H.3.3 Key Switch Signals (Xn, Xn+2)

The key switch signals are coded by the PMC management software, and input to the area indicated by address R in the form of to the bit image.

Whether necessary keys are already pressed can be checked by the bit image of the key switches using the user PMC ladder program. (See Tables H.3.4 (a), H.3.4 (b), and H.3.4 (c))

While a key is pressed, the bit corresponding to the key is 1.

Two keys can be pressed at the same time. Create a user PMC program so that it does not require pressing more than two keys at a time. If more than two keys are pressed simultaneously, the relevant data is not entered correctly.

A maximum of 60 ms is required before the corresponding bit is set to 1 or 0 after a key is pressed (released).

Key switch signal addresses (Xn to Xn+2: Table H.3.4 (a)) and their bit image addresses (Rk to Rk+7: Tables H.3.4 (b) and H.3.4 (c)) can be defined using fixed addresses or unused addresses as desired. (In Series 0, the key switch signal addresses are fixed to X20 and after. The bit image addresses are fixed to F292 and after.)

H.3.4 LED Signals (Ym)

Specify the LED signals at PMC address R using the user PMC ladder program in the form of a bit image. PMC management software changes the bit image LED signals to the coded output signals. (See Tables H.3.4 (a), H.3.4 (b), and H.3.4 (c))

While 1 is written in a LED bit image, the relevant LED automatically goes on. When 0 is written in the LED bit image, the relevant LED goes off. All LEDs are off before the power is turned on.

A maximum of 200 ms is required before the LED goes on or off after 1 or 0 is written in a bit image in the PMC.

LED signal address (Ym: Table H.3.4 (a)) and the bit image addresses (Rl to Rl+7: Tables H.3.4 (b), and H.3.4 (c)) can be defined using fixed addresses or unused addresses as desired. (In Series 0, the LED signal address is fixed to Y51. The bit image addresses are fixed to G242 and after.)

Table H.3.4 (a) Key switch and LED signal addresses

	#7	#6	#5	#4	#3	#2	#1	#0
Xn	KD7	KD6	KD5	KD4	KD3	KD2	KD1	KD0
Xn+1								
Xn+2	KST				KA3	KA2	KA1	KA0
Ym	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0

Table H.3.4 (b) Bit Image addresses of key switch and LED signals (for the small operator's panel)

	#7	#6	#5	#4	#3	#2	#1	#0
KEY/LED	F3	F2	F1		D1	C1	B1	A1
Rk/Rl	F4				D2	C2	B2	A2
Rk+1/Rl+1	D4	D3	C4	C3	B4	B3	A4	A3
Rk+2/Rl+2		F6	F5		D5	C5	B5	A5
Rk+3/Rl+3	F8				D6	C6	B6	A6
Rk+4/Rl+4	D8		C8		B8		A8	A7
Rk+5/Rl+5			F9		D9	C9	B9	A9
Rk+6/Rl+6			F10		D10	C10	B10	A10

Table H.3.4 (c) Bit image addresses of key switch and LED signals (for the operator's panel with the full keyboard)

	#7	#6	#5	#4	#3	#2	#1	#0
KEY/LED	E1	C1	A1	E6	D6	C6	B6	A6
Rk/Rl	E2	C2	A2	E7	D7	C7	B7	A7
Rk+1/Rl+1	E3	C3	A3	E8	D8	C8	B8	A8
Rk+2/Rl+2	E5	C4	A4	E9	D9	C9	B9	A9
Rk+3/Rl+3	D2	C5	A5	E10	D10	C10	B10	A10
Rk+4/Rl+4	D4	D5	B2	E11	D11	C11	B11	A11
Rk+5/Rl+5	D1	B1	B4	E12	D12	C12	B12	A12
Rk+6/Rl+6	D3	B3	B5	E13	D13	C13	B13	A13

H.4 SPECIFYING ADDRESSES

The following section describes how to specify key switch and LED signal addresses and the bit image addresses.

H.4.1 Parameter Menu

```
KEY IN ONE OF THE FOLLOWING NO.S WHICH YOU WANT TO SET PARA.S
NO. ITEMS                       CURRENT PARAMETERS
01  (UNUSED)                  ;
02  COUNTER DATA TYPE         ; BINARY
03  OPERATOR PANEL            ; YES
     KEY/LED ADDRESS          ; X0000/Y0000
     KEY/LED BIT IMAGE ADRS.  ; R0900/R0910
04  PMC TYPE                  ; PMC–RB
05  (UNUSED)                  ;
06  (UNUSED)                  ;
07  (UNUSED)                  ;
08  (UNUSED)                  ;
09  IGNORE DIVIDED CODE       ; NO
10  (UNUSED)                  ;
00  NOTHING TO SET            ;
                              ; ROM WRITER = FA WRITER
NO. =
```

H.4.2 Procedure

1) Select 3 from the parameter menu. Then, the following message is displayed:

```
EXAMPLE 0:NO, 1:YES
OP.PANEL=_
```

2) Select 1(:YES). Then, the following message is displayed:

```
SET KEY/LED ADDRESS(KEY ADRS., LED ADRS.)
ADDR=_
```

3) Specify a key or LED address (X or Y). For example, to specify a key switch address as X0 and LED address as Y0, enter X0,Y0 and press the [NL] key ([NL]: New line key). The following message is then displayed:

```
SET KEY/LED BIT IMAGE ADDRESS(KEY ADRS., LED ADRS.)
ADDR=_
```

4) Specify bit image addresses. For example, to specify R900 and R910, enter R900,R910 and press the [NL] key.

Then, the current display returns to the original parameter menu, and the following message appears:

```
  :      :     :
03  OPERATOR PANEL           ; YES
     KEY/LED ADDRESS         ; X0000/Y0000
     KEY/LED BIT IMAGE ADRS. ; R0900/R0910
  :      :     :
```

NOTE

1 After the above procedure, the addresses in Tables 3.1, 3.2–A, and 3.2–B are defined as the following PMC addresses:

Xn	→ X0000	Rk / Rl	→R0900/R0910
Xn+1	→ X0001	Rk+1 / Rl+1	→R0901/R0911
Xn+2	→ X0002	Rk+2 / Rl+2	→R0902/R0912
		Rk+3 / Rl+3	→R0903/R0913
Ym	→ Y0000	Rk+4 / Rl+4	→R0904/R0914
		Rk+5 / Rl+5	→R0905/R0915
		Rk+6 / Rl+6	→R0906/R0916
		Rk+7 / Rl+7	→R0907/R0917

2 Since the PMC addresses for the I/O card are already fixed, specify the signals to be used at the fixed addresses.

Examples

To use X1000, X1001, X1002, and Y1000 for key switches and LEDs, enter the following:

```
SET KEY/LED ADDRESS(KEY ADRS., LED ADRS.)
ADDR= X1000,Y1000 [NL]
```

I EDITING FOR Power Mate–MODEL D (PMC–PA1/PA3)

I.1 OUTLINE

Ladder diagram editing function for FANUC PMC–MODEL PA1/PA3 has high compatibility in a basic specification between ladder diagram editing function for FANUC PMC–MODEL SA1/SA2.

Following abbreviations are used in this chapter.

CNC Model	Product/Card Name	Abbr.
FANUC Power Mate–MODEL D	FANUC PMC–MODEL PA1	PMC–PA1
	FANUC PMC–MODEL PA3	PMC–PA3
	Ladder diagram editing memory card	Editing card
FANUC Series 18	FANUC PMC–MODEL SA1	PMC–SA1
	FANUC PMC–MODEL SA2	PMC–SA2

I.2 COMPATIBILITY WITH CNC BASIC SOFTWARE

Editing card described herein apply to the following software or later.

CNC

· Version 08(H) or later of Power Mate–MODEL D basic software 8830 Series.

PMC

· Version 04(D) or later of PMC–PA1/PA3 control software 4075 Series.

I.3 PMC PROGRAMMER (CRT/MDI OR PDP/MDI) [LADDER EDITING FUNCTION]

This function is used to set PMC system parameters and also generate and execute sequence programs by using soft keys a on the CRT/MDI unit or PDP/MDI unit. You can not use following function because FANUC Power Mate–MODEL D does not use ROM for sequence program.

- Sequence Program Copy Function
- Writing, Reading, and Verification of the Sequence Program and PMC Parameter Data to/from/with ROM.

I.3.1 Component Units and Connections

The units required for generating a sequence program and connection methods are described below.

I.3.1.1 Component units

(1) Editing card

This is used for editing sequence program.

If this card is inserted in CNC at the time of its power–on, PMC displays the programmer menu.

When you want to put on and take off, you must turn off the CNC power.

> **CAUTION**
> Please do not release the write protect switch of editing card for preventing a mistake deleting.

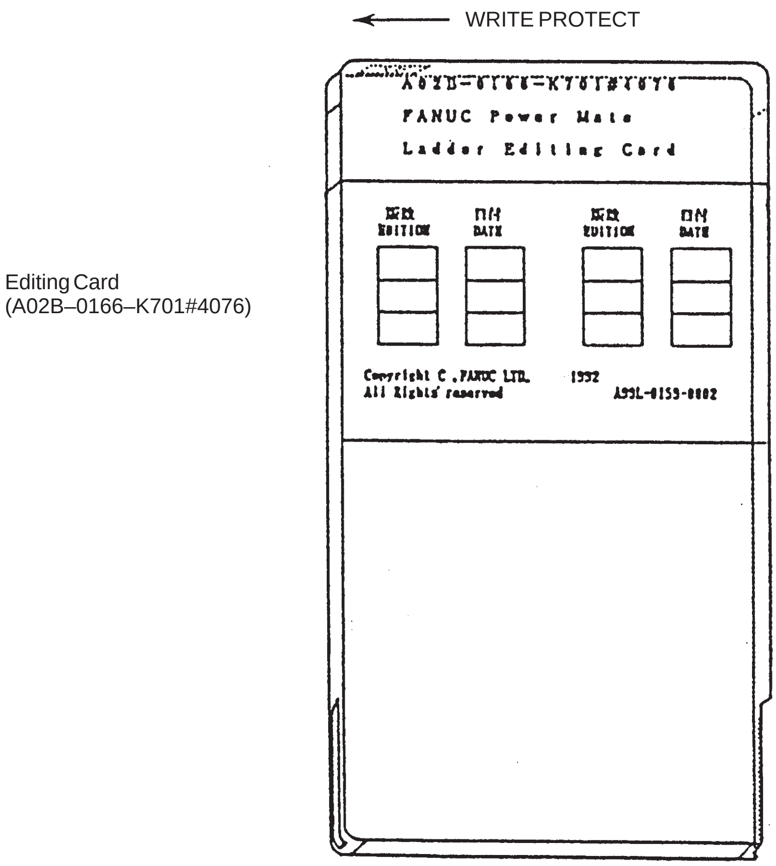


Fig. I.3.1.1

(2) CRT/MDI unit, PDP/MDI unit

CRT/MDI unit or PDP/MDI unit are necessary when you generate or edit sequence program using editing card.

CRT/MDI unit (A02B–0166–C001)

PDP/MDI unit (A02B–0166–C010, A02B–0166–C011)

I.3.1.2 Connection of components

Feed the editing card into connector CNMC of the CNC.

When you want to put on and take off, you must turn off the CNC power. (Refer to the fig. I.3.1.2)

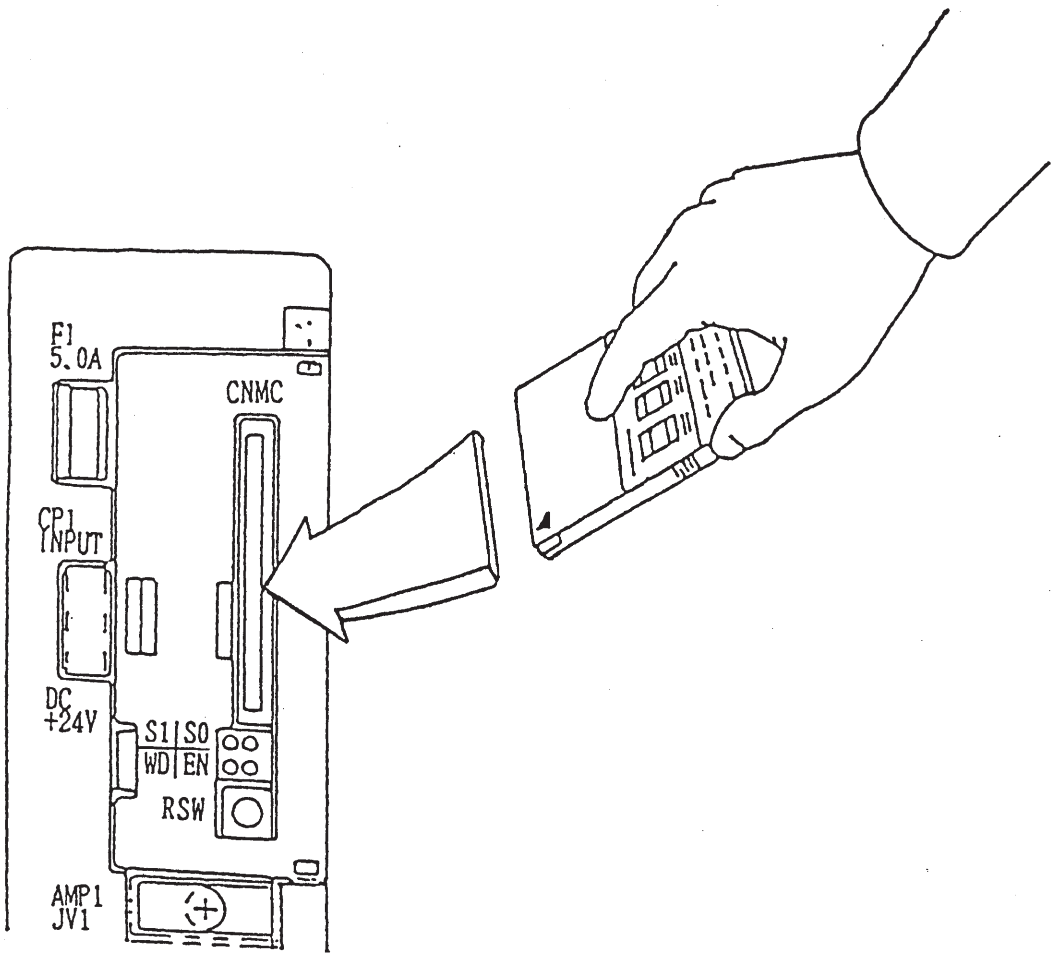


Fig. I.3.1.2

I.3.1.3 Parameter

Please set bit 1 in K17 of keep relay area for PMC parameters.

	#7	#6	#5	#4	#3	#2	#1	#0
K17							PRGRAM	

PRGRAM 0: The programmer function is disabled.
(The programmer menu is not displayed.)

1: The programmer function is enabled.
(The programmer menu is displayed.)

I.3.2 Specification and Display of System Parameters (SYSPRM)

FANUC Power Mate–MODEL D can set only COUNTER DATA TYPE.

The meaning of this parameter is same as PMC–SA1/SA2.

```
PMC SYSTEM PARAMETER
  COUNTER DATA TYPE  =  BINARY/BCD









 [BINARY] [ BCD  ] [      ] [      ] [      ]
```

Fig. I.3.2 PMC–PA1 or PA3 System Parameter Screen

I.3.3 Condense

When the following condition is satisfied, the CONDNS key will be used in FANUC Power Mate–MODEL D.

- Some unused area remain by repeating the addition or the deletion of the symbol/comment and the message in the memory.
- Ladder might be able to be made more by compressing the unused area by pushing [CONDNS] key when the memory is insufficient while ladder is added.

[Example: When you want to expand ladder area by deleting symbol/comment data at the memory status Fig.I.3.3 (a)]

(1) Delete symbol data(0.2KB).

(2) Push [CONDNS] key.

(3) The memory status becomes as Fig.I.3.3 (c) and LADDER can be edited more.

NOTE
1 Sequence program area in Fig. I.3.3 (a) – (c) is 64KB.
2 The underlined memory in Fig. I.3.3 (a) – (c) is the same as the memory display of the TITLE screen.
3 The symbol/comment area in Fig. I.3.3 (a) is 20KB (Unused area 0.9KB is contained.)
4 In case of deleting message and expending another area, it is as same as this example.

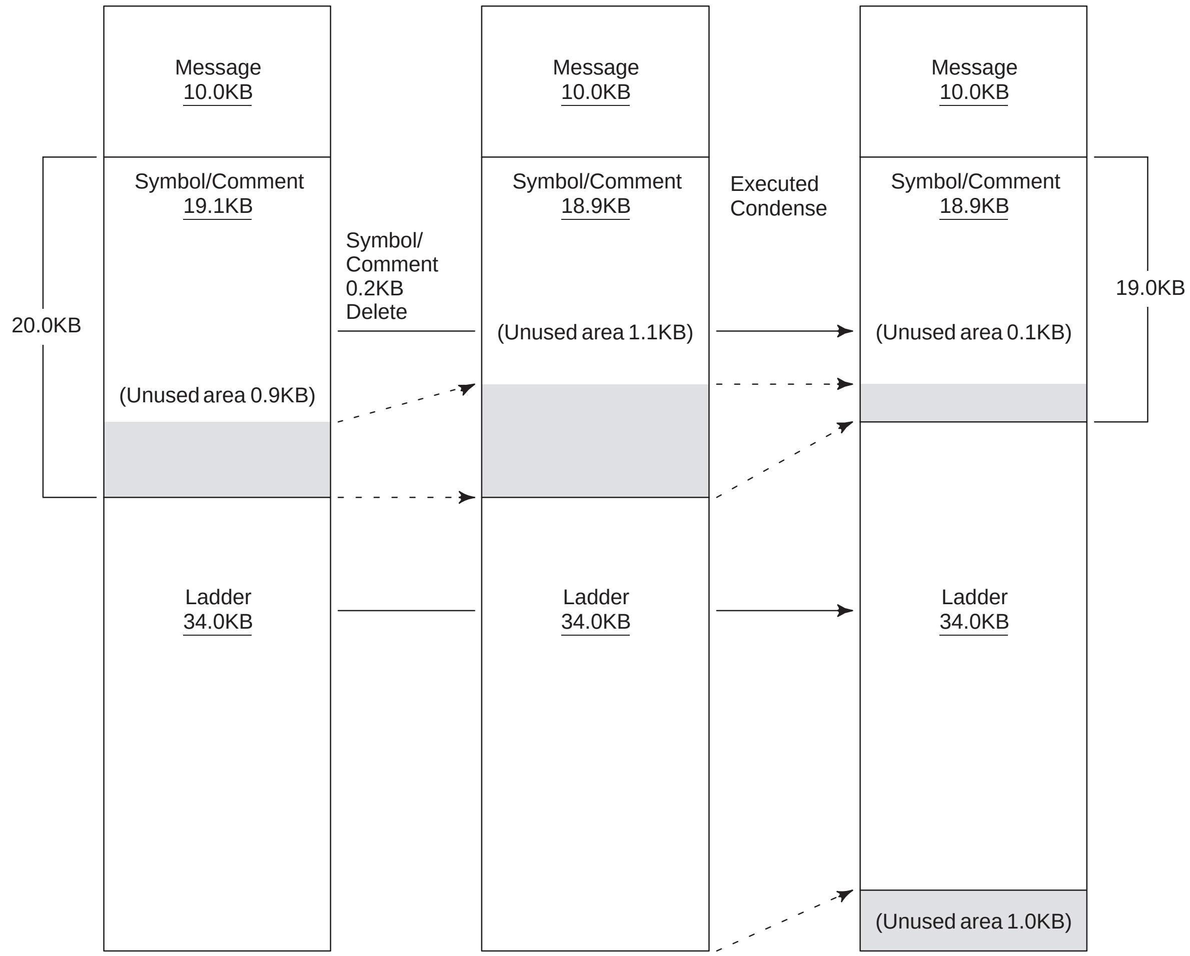


Fig. I.3.3 (a) **Fig. I.3.3 (b)** **Fig. I.3.3 (c)**

I.4 SYSTEM DIAGRAM OF SOFT KEY

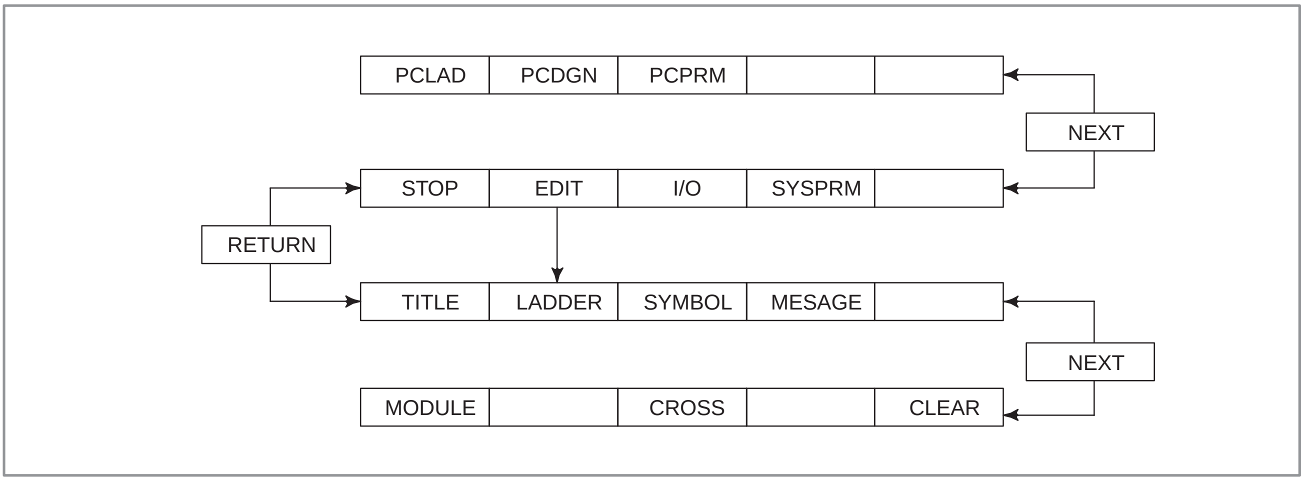


Fig. I.4 (a)

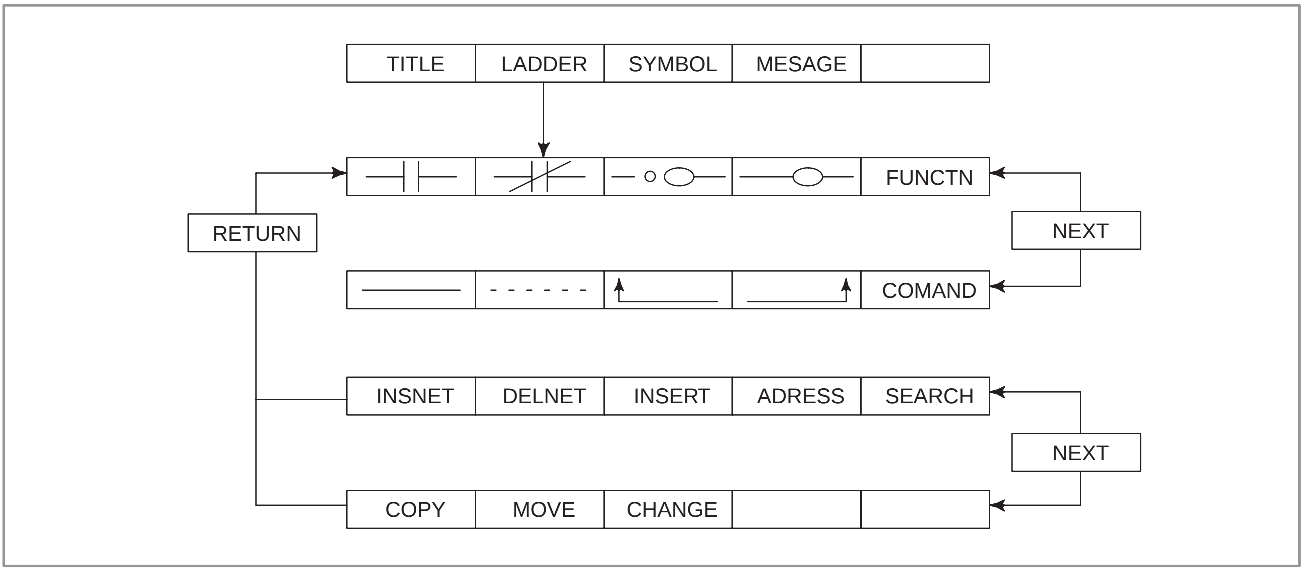


Fig. I.4 (b)

J APPLICABLE FAPT LADDER EDITIONS

The following tables list the editions of offline programs required to program each PMC model.

J.1 FAPT LADDER, FAPT LADDER–II, LADDER EDITING PACKAGES

Drawing number \ Model	PMC–PA1	PMC–PA3	PMC–SA1	PMC–SA2	PMC–SA3	PMC–SA5	PMC–SB	PMC–SB2	PMC–SB3
A02B–9200–J502#JP A02B–9201–J502#EN	5.0 and later	5.0 and later	2.0 and later	2.0 and later	5.0 and later	6.2 and later (Note)	1.0 and later	2.0 and later	5.0 and later
A02B–9200–J603#JP A02B–9201–J603#EN	4.0 and later	4.0 and later	1.0 and later	1.0 and later	4.0 and later	4.2 and later (Note)	–	–	–
A02B–9200–J604#JP A02B–9201–J604#EN	–	–	–	–	–	–	1.0 and later	3.0 and later	4.5 and later
A08B–9201–J503	–	1.0 and later	2.0 and later	–	1.0 and later	2.0 and later	–	–	1.0 and later
A08B–9201–J510	–	–	2.1 and later	–	1.3 and later	2.2 and later	–	–	1.0 and later

Drawing number \ Model	PMC–SB4	PMC–SB5	PMC–SB6	PMC–SC	PMC–SC3	PMC–SC4	PMC–NB	PMC–NB2	PMC–NB6
A02B–9200–J502#JP A02B–9201–J502#EN	7.1 and later	8.5 and later	8.5 and later	1.0 and later	5.0 and later	7.1 and later	6.1 and later	8.5 and later	–
A02B–9200–J603#JP A02B–9201–J603#EN	–	–	–	1.0 and later	4.5 and later	5.0 and later	–	–	–
A02B–9200–J604#JP A02B–9201–J604#EN	5.0 and later	7.0 and later	7.0 and later	–	–	–	1.0 and later	3.0 and later	–
A08B–9201–J503	1.0 and later	1.1 and later	1.1 and later	–	1.0 and later	1.0 and later	1.0 and later	1.0 and later	–
A08B–9201–J510	1.0 and later	1.0 and later	1.0 and later	–	1.0 and later	1.0 and later	2.1 and later	2.1 and later	–

Body of FAPT LADDER	(PC–9801)	A02B–9200–J502#JP
Body of FAPT LADDER	(IBM PC/AT)	A02B–9201–J502#EN
PMC–SA module	(PC–9801)	A02B–9200–J603#JP
PMC–SA module	(IBM PC/AT)	A02B–9201–J603#EN
PMC–SB/SC module	(PC–9801)	A02B–9200–J604#JP
PMC–SB/SC module	(IBM PC/AT)	A02B–9201–J604#EN
PMC–NB module	(PC–9801)	A02B–9200–J606#JP
PMC–NB module	(IBM PC/AT)	A02B–9201–J606#EN
FAPT LADDER–II	(IBM PC/AT)	A08B–9201–J503
Ladder editing package	(IBM PC/AT)	A08B–9201–J510

NOTE
When a PMC–SA5 ladder is to be created, set the model to PMC–SA3.

J.2 FAPT LADDER (SYSTEM P SERIES)

A08B–0035–J595#E (P–G Mark II): FAPT LADDER PMC–SA1/SA2

A08B–0036–J595#E (P–G Mate):

Model / Edition	PMC–PA1	PMC–PA3	PMC–SA1	PMC–SA2	PMC–SA3
1.1 and later	×	×	○	×	×
2.1 and later	×	×	○	○	△
3.1 and later	×	○	×	×	×
4.1 and later	○	○	○	○	△

×: Not supported, ○: Supported, △: Restrictedly supported (Note)

> **NOTE**
> A sequence program cannot be transferred from the PMC–SA1 of the FANUC Series 20 to the offline programmer (edition 6.0 or an earlier edition). If this is attempted, alarm 89 occurs in the offline programmer.

A08B–0036–J964 (P–G Mark II and P–G Mate):
PMC–SA1/SA2/SB/SC/PA1/PA3 data

Model / Edition	PMC–PA1	PMC–PA3	PMC–SA1	PMC–SA2	PMC–SA3
1.1 and later	×	×	○	×	×
2.1 and later	○	○	○	○	×

×: Not supported, ○: Supported, △: Restrictedly supported (Note)

A08B–0035–J595#E (P–G Mark II):
FAPT LADDER PMC–SB/SB2/SC

A08B–0036–J595#E (P–G Mate):

Model / Edition	PMC–SB	PMC–SB2	PMC–SB3	PMC–SC	PMC–SC3	PMC–NB
1.1 and later	○	×	×	○	×	×
4.1 and later	○	○	△	○	×	×

×: Not supported, ○: Supported, △: Restrictedly supported (Note)

NOTE

The edition of FAPT LADDER adopted for the PMC–SA2 or PMC–SB2 can be used to program the PMC–SA3 or PMC–SB3 as long as some functional instructions including structured programming are not used (as long as FAPT LADDER is used within the range of the specifications of the PMC–SA2 or PMC–SB2).
When this edition is used:

(1) The following functional instructions cannot be used. (For details, see Section 5 of Part I.)
– MOVB, MOVW, MOVN
– DIFU, DIFD
– AND, OR, NOT, EOR
– END, CALL, CALLU, SP, SPE
– JMPB, JMPC, LBL

(2) A sequence program created by the editing function (ladder editing module) contained in the PMC–SA3/SB3 cannot be edited after it is read into the offline programmer.

(3) A sequence program created by the offline programmer and transferred to the PMC (sequence program transferred and edited by the built–in editing function) can be edited again after it is read into the offline programmer.

K LEVEL UP OF INPUT/OUTPUT FUNCTION WITH MEMORY CARD

K.1 OUTLINE OF LEVELED UP CONTENTS

The function is leveled up, that is Input/Output function with Memory Card by CNC or Offline Programmer. The leveled up contents are as follows.

(1) The time is reduced in Inputing/Outputing between CNC and Memory Card by PMC I/O function. This is the same between Offline Programmer and Memory Card.

(2) Sequence programs can be inputted from Memory Card by BOOT SYSTEM, by which CNC management software or so can be inputted. (Refer to K.2.3.)

Memory Card function can be used in the following editions of CNC basic software and PMC management software and FAPT LADDER for Personal Computer.

· CNC basic software

	non leveled up	leveled up
FANUC Series 20–FA basic software (D001)	05–06	More than 07
FANUC Series 20–TA basic software (D101)	02	More than 03

· PMC management software

	non leveled up	leveled up
PMC–SA1/SA3 management software (4080)	04–05	More than 06

· FAPT LADDER for Personal Computer

	non leveled up		leveled up
FAPT LADDER PMC–SA1/SA2/SB/SB2/SC SYSTEM (A08B–9200–J502#JP (PC–9801)) (A08–9201–J502#EN (IBM PC/AT))	6.1	6.2	More than 6.3
PMC–SA1/SA2 MODULE (PMC–SA1/SA2/SA3/PA1/PA3) (A08B–9200–J603#JP (PC–9801)) (A08–9201–J603#EN (IBM PC/AT))	4.1	More than 4.2	

K.2 OPERATION

K.2.1 CNC → Offline Programmer

(1) Operation of CNC

1)On PMC I/O screen, specify M–CARD as "DEVICE", WRITE as "FUNCTION", LADDER as "DATA KIND", any file name, which is omissible, as "FILE NO." (See Fig. K.2.1 (a)) and press the soft key [EXEC].

```
PMC I/O PROGRAM                      MONIT  STOP

    CHANNEL    =    1
    DEVICE     = M–CARD
    FUNCTION   = WRITE
    DATA  KIND = LADDER
    FILE  NO.  =
     ( #NAME )

[ EXEC ][CANCEL][ WRITE ][ READ ][COMPAR]

[DELETE][ LIST ][FORMAT ][      ][SETUP ]
```

Fig. K.2.1 (a) PMC I/O screen

(2) Operation of Offline Programmer (FAPT LADDER for Personal Computer)

2) Mount a Memory Card interface on the personal computer.

3) Select [INOUT] (I/O) from the main menu.

4) Select [M–CARD] (Memory Card) from the I/O menu. (See Fig. K.2.1 (b))

```
       I/O              PMC–RA1          <O>[A:¥FLADDER¥          ]
                  F1  KEY :  FA WRITER
                  F2  KEY :  PMC WRITER
                  F3  KEY :  PMC
                  F4  KEY :  Handy File
                  F5  KEY :  Memory Card
                  F10 KEY :  END

|FAWRT|PMC |PMC |FDCAS|M    |  |  |  |  |  |END|
|     |WRT |    |     |CARD |  |  |  |  |  |   |
```

Fig. K.2.1 (b) I/O Menu screen

5) Select [READ] (PROGRAMMER ← Memory Card). (See Fig. K.2.1 (c))

```
I/O (M_CARD)          PMC–RA1       <O> [A:¥FLADDER¥              ]

F1  KEY :  WRITE   (PROGRAMMER –> Memory Card)

F2  KEY :  READ    (PROGRAMMER <– Memory Card)

F10 KEY :  END

WRITE | READ | | | | | | | END |
```

Fig. K.2.1 (c) I/O (I/O M_CARD) screen

6) Specify the followings:

– Name of the Memory Card file
  Specify the name of the file in the Memory Card which is to be converted and the Memory Card drive on which the Memory Card is mounted.

– Name of the ROM format file to be created
  Specify a file name to be given to the converted ROM format data.

```
I/O (FROM MC)         PMC–RA1       <O> [A:¥FLADDER¥              ]

READ (PROGRAMMER <– Memory Card)

Memory Card FILE NAME   :
(Specify the MEMORY CARD drive)

ROM FORMAT FILE NAME    :

EXEC | | | | | | | | END |
```

Fig. K.2.1 (d) I/O (FROM MC) screen

7) After it is decompiled, the converted ROM format file can be edited by the personal computer.

K.2.2 Offline Programmer → CNC

(1) Operation of Offline Programmer (FAPT LADDER for Personal computer)

1) Mount a Memory Card interface on the personal computer.

2) Compile a source program and create a ROM format file.

3) Return to the main menu and select [INOUT] (I/O).

4) From the I/O menu, select {M–CARD] (memory Card).

5) Select [WRITE] (PROGRAMMER → Memory Card). (See Fig. K.2.1(b)).

6) Specify the following:

– Name of the ROM format file
Specify the name of the ROM format file to be converted.

– Name of Memory Card file name
Specify the name to be given to the converted Memory Card file and the Memory Card drive to which the data is output. (The file can be accessed by the Memory Card interface incorporated into the CNC.)

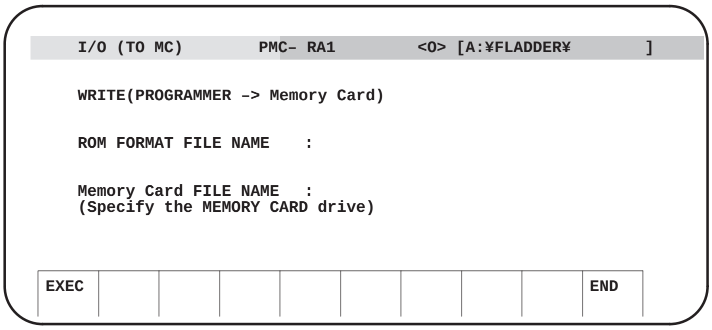


Fig. K.2.2 (a) I/O (TO MC) screen

(2) Operation of CNC

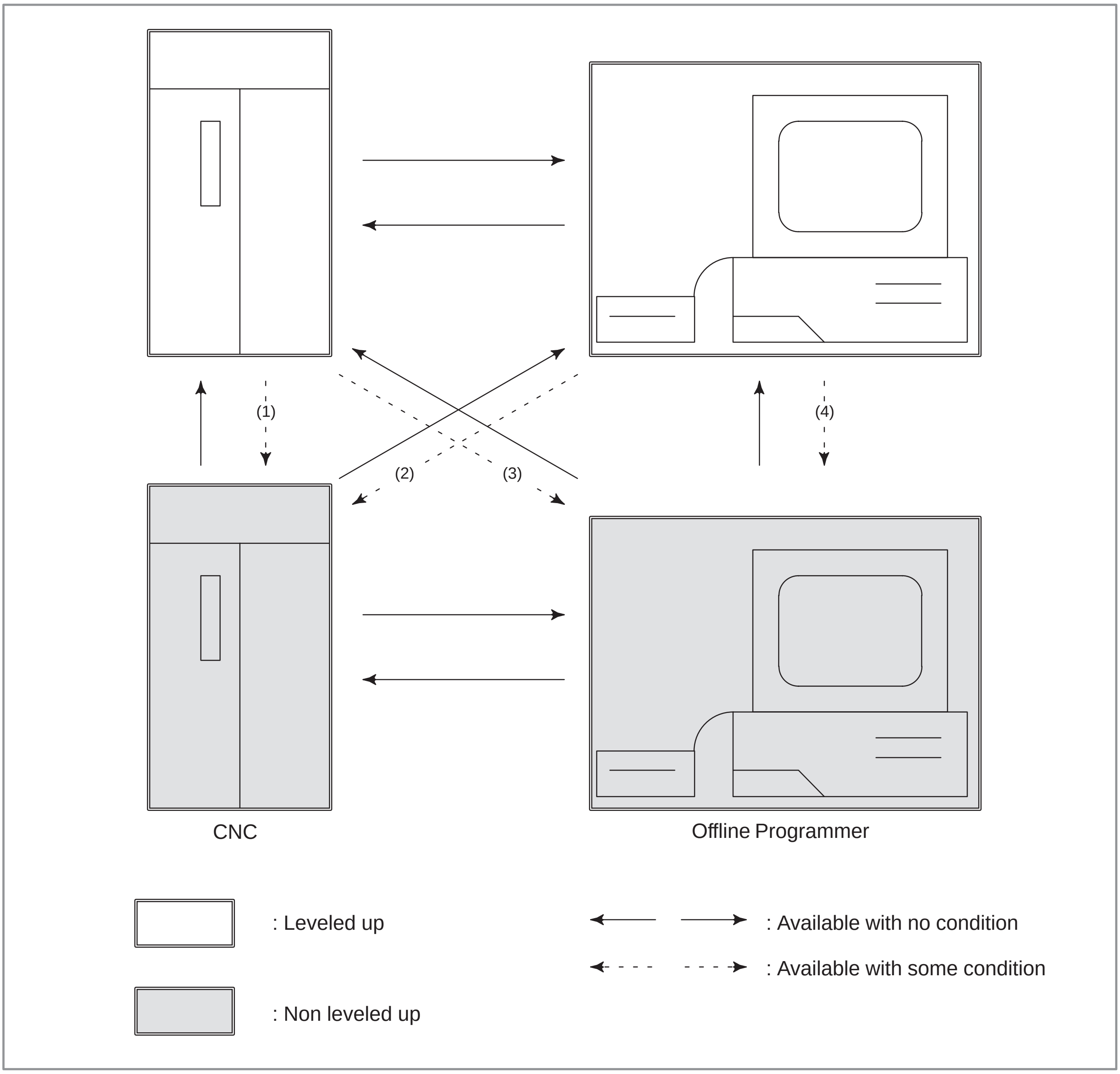
There are 2 methods by which the sequence program can be inputted from Memory Card.

– The method of using I/O function of PMC
On PMC I/O screen, specify M–CARD as “DEVICE”, READ as “FUNCTION”, the file name or file No. you want to input as “FILE NO.” and press the soft key [EXEC].

– The method of using BOOT SYSTEM (When CNC starting up)
Refer to K.2.3.

K.2.3 Note

Sequence programs which are output from leveled up CNC or Offline Programmer to Memory Card can not be input to non leveled up CNC or Offline Programmer. (Refer to the table of K.1)



The case of (1), (2), (3) and (4) are explained as follows.

· In case of (1), (2)

Output operation : There is no special operation
Input operation : Input sequence programs buy BOOT SYSTEM. (Refer to K.2.3)

· In case of (3)

Output operation : Output sequence programs by setting the output format to 1 (:S–FORMAT) on the following SETUP screen PMC I/O. The default output format is 0 (:BINARY).

```
PMC I/O PROGRAM                         MONIT  STOP

    CHANNEL    =   1
    DEVICE     =  M–CARD
    FUNCTION   =  WRITE
    DATA KIND  =  LADDER
    FILE NO.   =
     ( #NAME )






[ EXEC ][CANCEL][ WRITE ][ READ ][COMPAR]
```

```
[DELETE][ LIST ][FORMAT ][      ][SETUP ]
```

```
PMC SETUP M–CARD                   MONIT  STOP


OUTPUT FORMAT (PROGRAM) =   1
 (0:BINARY,1:S–FORMAT)








[ INPUT ][     ][      ][      ][  INIT  ]
```

Press [INIT] key to reset default value 0.

Input operation : Input sequence programs by selecting F6:“I/O” on main menu screen of FAPT LADDER, then F3:“Handy File & Memory Card”.

· In case of (4)

Output operation : Output sequence programs by selecting F6:“I/O” on main menu screen of FAPT LADDER, then F4:“Handy File”.

Input operation : Input sequence programs by selecting F6:“I/O” on main menu screen of FAPT LADDER, then F3:“Handy File & Memory Card”.

K.3 NB/NB2 DATA COMPATIBILITY

Ladder data can be transferred by using a memory card.

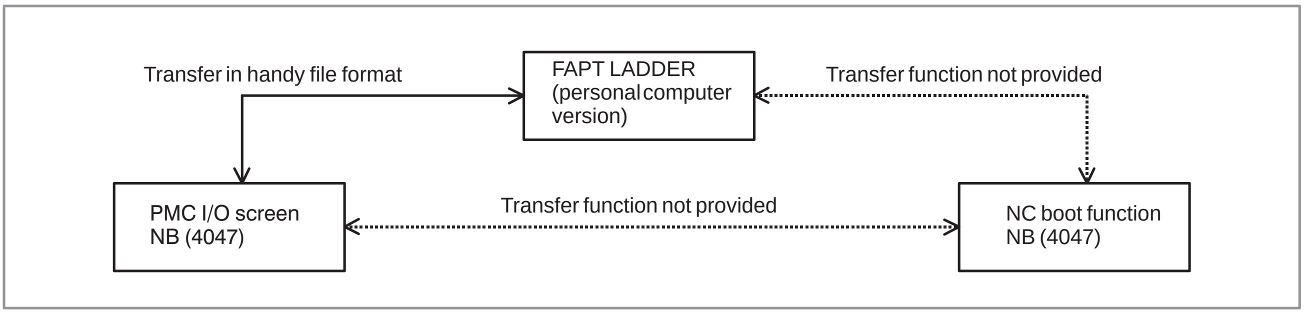
Two data formats are used:

– Handy file format

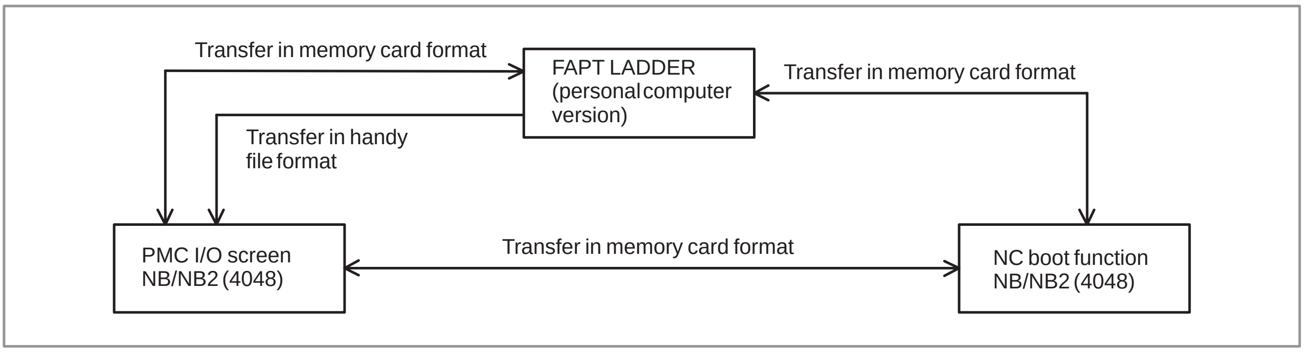
– Memory card format

The handy file format defines the S format data used with RS–232C. The memory card format defines the binary format data used for a boot. Data output to a memory card from the I/O screen of an NB of the 4047 series is handy file format data.
Data output to a memory card from the I/O screen of an NB/NB2 of the 4048 series is memory card format data.

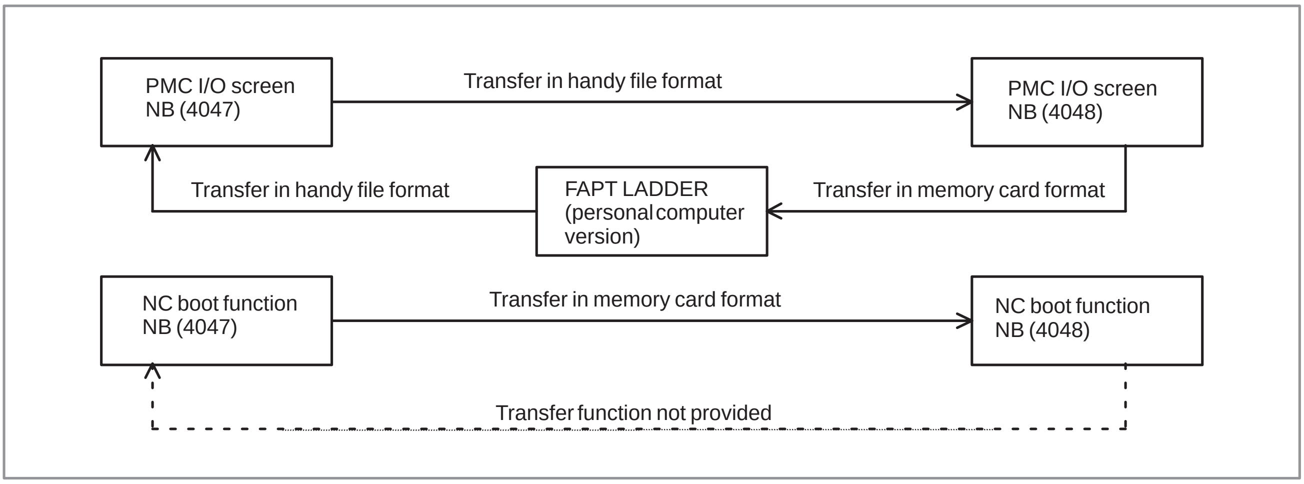
K.3.1 Data Transfer Between NB (4047 Series) and FAPT LADDER



K.3.2 Data Transfer Between NB/NB2 (4048 Series) and FAPT LADDER



K.3.3 Data Transfer Between NB (4047 Series) and NB (4048 Series)



L MIGRATION OF LADDER PROGRAMS BETWEEN DIFFERENT MODELS

L.1 MIGRATION OF LADDER PROGRAMS FROM Power Mate–D/H TO Power Mate *i*–D/H

The PMC–SB5/SB6 for the Power Mate *i*–D/H are upward–compatible with the PMC–PA1/PA3 for the Power Mate–D/H, according to their basic specifications. Because, however, sequence programs are not object–compatible with the PMC–PA1/PA3 for the Mate–D/H, Ladder programs sent from the Power Mate–D/H to a memory card cannot be directly loaded into the Power Mate *i*–D/H. For this reason, to migrate from the Power Mate–D/H to the Power Mate *i*–D/H, you must convert the PMC sequence programs.

This section provides information about the PMC programming compatibility that should be noted when you convert sequence programs.

Differences between the PMC–PA1/PA3 for the Power Mate–D/H and the PMC–SB5/SB6 for the Power Mate *i*–D/H

(1) PMC model change
Sequence programs for one PMC model must be converted to those for the other. These models are Ladder–source compatible with each other, and conversion is possible using the procedure described in the following manual:
Appendix 3.3, "Converting Sequence Programs for One PMC Model to Those for the Other" in the "FAPT LADDER–II Operator's Manual"

(2) Changes in the interface signals between CNC and PMC and between PMC and machine
The F, G, X, and Y address signals have been partially changed.
Change the sequence programs, referring to the following manual:
"FANUC Power Mate *i*–MODEL D/H Connection Manual (Functions)"

(3) Change in the basic command processing time
Because of the increase in command execution time, the following changes may occur:

- Changes in the execution cycle at the second Ladder level
- Changes in the timing of second–level division and first–level execution
- Changes in the timing of Ladder program execution and I/O transfer

(4) PMC addresses expansion
In the PMC–SB5, the inner relay (R) and the data table (D) have been expanded.
In the PMC–SB6, the inner relay (R), message request signal (A), variable timer (T), counter (C), keep relay (K), data table (D), and subprogram number (P) have been expanded.

PMC address type	Model			
	Power Mate–D	Power Mate–D/H	Power Mate *i*–D/H	
	PMC–PA1	PMC–PA3	PMC–SB5	PMC–SB6
Inner relay (R)	R0 to R999 R9000 to R9099	R0 to R999 R9000 to R9117	R0 to R1499 R9000 to R9117	R0 to R2999 R9000 to R9199
Message request signal (A)	A0 to A24			A0 to A124
Variable timer (T)	T0 to T79			T0 to T299
Counter (C)	C0 to C79			C0 to C199
Keep relay (K)	K0 to K19			K0 to K39, K900 to K909
Data table (D)	D0 to D1859		D0 to D2999	D0 to D7999
Subprogram number (P)	–	P1 to P512		P1 to P2000

(5) Function command specification expansion
Expanded specifications have been added to the function commands DECB, NUMEB, XMOVB, and PSGN2. The conventional, basic specifications are still valid and, therefore, sequence programs need not be changed for migration. For an explanation of the expanded specifications, see Section I.5, “PMC Function Commands.”

(6) Changes in a window function
The CNC alarm state read function code has been changed from 23 to 186. In addition, the window function has been changed to the low–speed type. For details, see Section B.4.12, “Reading the CNC alarm state.”

L.2 MIGRATION FROM THE PMC–NB/NB2 TO THE PMC–NB6

If you previously used the Series 15–B PMC–NB/NB2, you must convert from the PMC–NB/NB2 to the PMC–NB6. The conversion procedure is as described below.

(1) Model change from the PMC–NB/NB2 to the PMC–NB6 (source programs)
When changing the model from the PMC–NB/NB2 to the NB6, you must convert the source programs for one PMC model to those for the other using FAPT LADDER–II. Conversion is possible using the procedure described in the following manual:

Appendix 3.3, “Converting Sequence Programs for One PMC Model to Those for the Other” in the “FAPT LADDER–II Operator’s Manual” (B–66184EN)

The mnemonic file format for the PMC–NB6 system parameters used for model conversion is as follows:

%@0		
2 BCD	2. Counter data type	(BINARY or BCD)
3 NO	3. Whether the operator panel exists	(exits: YES, does not exist: NO)
4 PMC–NB6	4. PMC type	(PMC–NB6)
7 100	7. Ladder execution time	(100%)

The specification changes in the NB6, described in Section I.1.2, “Overview of the Ladder Specifications,” must be dealt with separately.

(2) When using FAPT LADDER
When creating Ladder programs for the PMC–NB6 using FAPT LADDER, specify use of the PMC–NB2. When PMC–NB2 is specified, however, the following restrictions are imposed:

a) Do not use the functions supported by the PMC–NB6. (See Section I.1.2, “Overview of the Ladder Specifications” for details.)

b) Do not use sequence programs with C programs linked.

(3) If using sequence programs located on the FANUC Series 15–B

a) If using sequence programs for the PMC–NB2 that are located on the Series 15–B in the Series 15*i*, the same restrictions as those described in (2) are imposed. Provided that these restrictions are observed, programs for the PMC–NB2 can be output to a memory card (by specifying “LADDER” for DATA KIND on the I/O screen) and directly loaded into the Series 15*i*.

b) Sequence programs for the PMC–NB that are located on the Series 15–B cannot be used in the Series 15*i*, without first being converted. They must be converted into programs for the PMC–NB6 with the model change procedure using FAPT LADDER–II, described in (1).

M ALARM MESSAGE LIST

Alarm messages 1 (alarm screen)

Message	Contents and solution
ALARM NOTHING	Normal status
ER00 PROGRAM DATA ERROR (ROM)	The sequence program in the ROM is not written correctly. (solution) Please exchange ROM for the sequence program.
ER01 PROGRAM DATA ERROR (RAM)	The sequence program in the debugging RAM is defective. (solution) Please clear the debugging RAM and input LADDER again. The debugging RAM is not installed though the RAM is selected. (solution) Please install the debugging RAM or install ROM for sequence program and select ROM with K17#3=0.
ER02 PROGRAM SIZE OVER	The size of a sequence program exceeded the maximum allowable ladder size. (solution) The ordered RAM size is smaller than the option. Contact FANUC. Change the value of MAX LADDER AREA SIZE on the SYSPRM screen, then turn the power off then back on (only with PMC–SC).
ER03 PROGRAM SIZE ERROR (OPTION)	The size of sequence program exceeds the option specification size. (solution) Please increase the option specification size. Or, reduce the size of sequence program.
ER04 PMC TYPE UNMATCH	The PMC model setting of the sequence program is not corresponding to an actual model. (solution) Please change the PMC model setting by the offline programmer.
ER05 PMC MODULE TYPE ERROR	The module type of the PMC engine is not correct. (solution) Please exchange the module of PMC engine for a correct one.
ER06 PROGRAM MODULE NOTHING	Both ROM for sequence program and the debugging RAM do not exist (PMC–SC only). For a 3–path system, the PMC model must be SB6. (solution) Contact FANUC.
ER07 NO OPTION (LADDER STEP)	There is no step number option of LADDER.
ER10 OPTION AREA NOTHING (SERIES–NAME)	The PMC–SB management software is not transferred. (solution) There is a mismatch between the order and delivered the software. Contact FANUC.
ER11 OPTION AREA NOTHING (SERIES–NAME)	The PMC C language board management software is not transferred. (solution) There is a mismatch between the order and delivered the software. Contact FANUC.
ER12 OPTION AREA ERROR (SERIES–NAME)	There is a series mismatch between the basic and option of the PMC–SB management software. (solution) Contact FANUC.
ER13 OPTION AREA ERROR (SERIES–NAME)	There is a series mismatch between the basic and option of the PMC C language board management software. (solution) Contact FANUC.
ER14 OPTION AREA VERSION ERROR (SERIES–NAME)	There is an edition mismatch between the basic and option of the PMC–SB management software. (solution) Contact FANUC.
ER15 OPTION AREA VERSION ERROR (SERIES–NAME)	There is an edition mismatch between the basic and option of the PMC C language board management software. (solution) Contact FANUC.

Message	Contents and solution
ER16 RAM CHECK ERROR (PROGRAM RAM)	The debugging RAM cannot be read/written normally. (solution) Please exchange the debugging RAM.
ER17 PROGRAM PARITY	The parity error occurred on ROM for sequence program or the debugging RAM. (solution) ROM: The deterioration of ROM may be deteriorated Please exchange ROM for the sequence program RAM: Please edit the sequence program once on PMC Still the error occurs, exchange the debugging RAM. F–ROM: (PMC–NB/FS–20) Please edit the sequence program once on PMC and write sequence program to F–ROM again.

NOTE

1 The PMC–SB3/SC3 for the Series 16 MODEL–B does not support ER00 and ER06.

2 For the PMC–SB3/SC3 for the Series 16 MODEL–B, the ”debugging RAM” and ”ROM for sequence program,” described in the table, are not supported but the relevant descriptions apply to ordinary RAM.

Alarm messages 2 (alarm screen)

Message	Contents and solution
ER18 PROGRAM DATA ERROR BY I/O	Transferring the sequence program from offline programmer was interrupted by the power off etc. (solution) Please clear the sequence program and transfer the sequence program again.
ER19 LADDER DATA ERROR	Editing the LADDER was interrupted by the power off or by the switch to the CNC screen by the function key etc. (solution) Please edit LADDER once on PMC. Or, please input LADDER again.
ER20 SYMBOL/ COMMENT DATA ERROR	Editing the symbol and comment was interrupted by the power off or by the switch to the CNC screen by the function key etc. (solution) Please edit symbol and comment once on PMC. Or, please input symbol and comment again.
ER21 MESSAGE DATA ERROR	Editing the message data was interrupted by the power off or the switch to the CNC screen by the function key etc. (solution) Please edit message data once on PMC. Or, please input message data again.
ER22 PROGRAM NOTHING	There is no sequence program.
ER23 PLEASE TURN OFF POWER	There is a change in setting LADDER MAX AREA SIZE etc. (solution) Please restart the system to make the change effective.
ER24 LADDER, LANGUAGE AREA OVERLAP	The ladder area overlaps the C language area. (solution) Adjust the C program address range.
ER25 SOFTWARE VERSION ERROR (xx)	xx=PMCAOPT : The version in the PMC–SB management software does not match. xx=PMCBAS–2 : The version in the PMC–NB6 management software does not match.
ER26 PMC CONTROL MODULE ERROT (xx)	xx=PMCAOPT : The PMC–SB management software has not been initialized. xx=PMCBAS–2 : The PMC–NB6 management software has not been initialized.
ER32 NO I/O DEVICE	Any DI/DO unit of I/O Unit or the connection unit etc. is not connected. When built–in I/O card is connected, this message is not displayed. (solution) When built–in I/O card is used: Please confirm whether the built–in I/O card is certainly connected with. When I/O Link is used: Please confirm whether the DI/DO units turning on. Or please confirm the connection of the cable.
ER33 SLC ERROR	The LSI for I/O Link is defective. (solution) Please exchange the module of PMC engine.
ER34 SLC ERROR (xx)	The communication with the DI/DO units of the xx group failed. (solution) Please confirm the connection of the cable connected to the DI/DO units of the xx group. Please confirm whether the DI/DO units turned on earlier than CNC and PMC. Or, please exchange the module of PMC engine on the DI/DO units of the xx group.
ER35 TOO MUCH OUTPUT DATA IN GROUP (xx)	The number of the output data in the xx group exceeded the max. The data, which exceed 32 bytes, become ineffective. (solution) Please refer to the following for the number of the data for each group. “FANUC I/O Unit–MODEL A connecting and maintenance manual” (B–61813E) “FANUC I/O Unit–MODEL B connecting manual”(B–62163E)

Message	Contents and solution
ER36 TOO MUCH INPUT DATA IN GROUP (xx)	The number of the input data in the xx group exceeded the max. The data, which exceed 32 bytes, become ineffective. (solution) Please refer to the following for the number of the data for each group. "FANUC I/O Unit–MODEL A connecting and maintenance manual" (B–61813E) "FANUC I/O Unit–MODEL B connecting manual"(B–62163E)
ER38 MAX SETTING OUTPUT DATA OVER (xx)	The assignment data for a group exceeds 128 bytes. (The assignment data of output side of xx group or later become ineffective.) (solution) Please reduce the assignment data to 128 bytes or less for the number of the output data of each group.
ER39 MAX SETTING INPUT DATA OVER (xx)	The assignment data for a group exceeds 128 bytes. (The assignment data of input side of xx group or later become infective.) (solution) Please reduce the assignment data to 128 bytes or less for the number of the input data of each group.
ER98 ILLEGAL LASER CONNECTION	The I/O unit group for the laser does not match the assignment data. (solution) Make sure that the actual I/O unit configuration matches the assignment data in the ladder.
ER99 X, Y96–127 ARE ALLOCATED	Laser I/O assignments are assigned to X96 to X127 and Y96 to Y127 when the laser I/O link is supported. (solution) Delete I/O assignment data of X96 to X127 and Y96 to Y127.

Alarm messages 3 (alarm screen)

Message	Contents and solution
WN01 LADDER MAX SIZE ERROR	The MAX LADDER AREA SIZE in the system parameter is illegal. (solution) Set the correct value to MAX LADDER AREA SIZE and restart the system.
WN02 OPERATE PANEL ADDRESS ERROR	The address setting data of the operator's panel for FS–0 is illegal. (solution) Please correct the address setting data.
WN03 ABORT NC–WINDOW/EXIN	LADDER was stopped while CNC and PMC were communicating. The functional instruction WINDR, WINDW, EXIN, DISPB, and etc. may not work normally. (solution) When restarting the system, this alarm will be released. Execute the sequence program(Press RUN key) after confirming whether there is a problem in LADDER or not.
WN04 UNAVAIL EDIT MODULE	The LADDER editing module cannot be recognized. (PMC–SA1/SA2/SA3/SB/SB2/SB3, except SA1/SA3 for FS–20) (solution) Please confirm the slot position installed. Please confirm the installed module.
WN05 PMC TYPE NO CONVERSION	A PMC–SA3/SA5 ladder was transferred to PMC–SB5. (solution) Correct the ladder type.
WN06 TASK STOPPED BY DEBUG FUNC	Some user tasks are stopped by break point of the debugging function.
WN07 LADDER SP ERROR (STACK)	When functional instruction CALL(SUB65) or CALLU(SUB66) was executed, the stack of the LADDER overflowed. (solution) Please reduce the nesting of the subprogram to 8 or less.
WN17 NO OPTION (LANGUAGE)	There is no C language option.
WN18 ORIGIN ADDRESS ERROR	The LANGUAGE ORIGIN address of the system parameter is wrong (solution) Please set the address of symbol RC_CTLB_INIT in the map file to the LANGUAGE ORIGIN of the system parameter.
WN19 GDT ERROR (BASE, LIMIT)	The value of BASE, LIMIT or ENTRY of user defined GDT is illegal. (solution) Please correct the address in link control statement and build file.
WN20 COMMON MEM. COUNT OVER	The number of common memories exceeds 8. (solution) Please reduce the number of common memories to 8 or less. It is necessary to correct a link control statement,build file and the source file for the common memory.
WN21 COMMON MEM. ENTRY ERROR	GDT ENTRY of the common memory is out of range. (solution) Please correct the address of GDT ENTRY of the common memory in the link control statement.
WN22 LADDER 3 PRIORITY ERROR	The priority of LADDER LEVEL 3 is out of range. (solution) Please correct the value of LADDER LEVEL 3 in the link control statement within the range of 0 or 10–99 or –1.
WN23 TASK COUNT OVER	The number of user tasks exceeds 16. (solution) Please confirm TASK COUNT in the link control statement. When the number of tasks is changed, it is necessary to correct the link control statement, build file and the composition of the files to be linked.
WN24 TASK ENTRY ADDR ERROR	The selector of the entry address to the user task is out of range. (solution) Please correct the table of GDT in build file to the value within 32(20H)–95(5FH).
WN25 DATA SEG ENTRY ERROR	The entry address of the data segment is out of range. (solution) Please correct DATA SEGMENT GDT ENTRY in the link control statement and the table of GDT in build file within 32(20H)–95(5FH).
WN26 USER TASK PRIORITY ERROR	The priority of the user task is out of range. (solution) Please correct the TASK LEVEL in link control statement within the range of 10–99 or –1. Note: Only one task can have TASK LEVEL –1 (including LADDER LEVEL 3).

Alarm messages 4 (alarm screen)

Message	Contents and solution
WN27 CODE SEG TYPE ERROR	The code segment type is illegal. The code segment of RENAMESEG in the binding control file is wrong. (solution) Please correct the entry of the code segment in the link control statement to correspond to the entry in the build file.
WN28 DATA SEG TYPE ERROR	The data segment type is illegal. The data segment of RENAMESEG in the binding control file is wrong. (solution) Please correct the entry of the code segment in the link control statement to correspond to the entry in the build file.
WN29 COMMON MEM SEG TYPE ERROR	The segment type of common memory is illegal. The segment of RENAMESEG in the building control file of the common memory is wrong. (solution) Please correct the entry of common memory in the link control statement to correspond to the entry in the build file.
WN30 IMOPSSIBLE ALLOCATE MEM.	The memories for the data and stack etc. cannot be allocated. (solution) Please confirm whether the value of code segment in build file and USER GDT ADDRESS in link control statement is correct or not. Or please reduce the value of MAX LADDER AREA SIZE of the system parameter and the size of the stack in link control statement at the least.
WN31 IMPOSSIBLE EXECUTE LIBRARY	The library function cannot be executed. (solution) Please confirm the object model of the library. Or, system ROM of PMC must be replaced with one of later version.
WN32 LNK CONTROL DATA ERROR	Link control statement data is illegal. (solution) Please confirm whether the address of symbol RC_CTLB_INIT in map file is set to LANGUAGE ORIGIN of the system parameter. Or, please make the link control statement again.
WN33 LNK CONTROL VER. ERROR	A link control statement data version error occurred. (solution) Correct the link control statement in the C program.
WN34 LOAD MODULE COUNT OVER	There are more than eight independent load modules. (solution) Reduce the number of independent load modules to eight or less.
WN35 CODE AREA OUT OF RANGE	The code segment area is outside the RAM area. (solution) Check the link map, and place segments within the RAM area.
WN36 LANGUAGE SIZE ERROR (OPTION)	The size of the language area exceeds the option. (solution) Check the free space, and increase the option.
WN37 PROGRAM DATA ERROR (LANG.)	The language program area is invalid. (solution) Clear the language area. [EDIT]→[CLEAR]→[CLRLNG]→[EXEC]
WN38 RAM CHECK ERROR (LANG.)	A RAM check error occurred in the language program area. (solution) Replace the RAM.
WN39 PROGRAM PARITY (LANG.)	A parity error occurred in the language program area. (solution) Reenter each language program. If an error still occurs, replace the RAM.
WN40 PROGRAM DATA ERROR BY I/O (LANG.)	Language program read operation was interrupted. (solution) Reenter the language program.
WN41 LANGUAGE TYPE UNMATCH	There is a C program type mismatch. (solution) Correct the C program.
WN42 UNDEFINE LANGUAGE ORIGIN ADDRESS	No language origin address is set. (solution) Set a language origin address.

System alarm messages 1 (PMC–SC)

Message STATUS LED	Contents and solution
PC1nn CPU INTERPT xxxx yyyyyy STATUS LED ☆★	A CPU error (abnormal interrupt) occurred. nn : CPU exception handling code It is an exception code of i80386. For details, please refer to the manual of the CPU. 00 Division error such as a divisor is 0 in division instruction. 12 Stack exception such as violations of limit of stack segment. 13 General protection exception such as segment limit over. xxxx : Segment selector where system error occurred. The selector of 0103–02FB is used by C language. yyyyyy: Offset address where system error occurred.
PC130 RAM PARITY aa xxxx yyyyyy STATUS LED □★	The parity error occurred on the debugging RAM of PMC. aa : RAM PARITY ERROR information. xxxx : Segment selector where system error occurred. yyyyyy: Offset address where system error occurred.
PC140 NMI BOC bb xxxx yyyyyy STATUS LED ★■	The RAM parity error or NMI(Non Maskable Interrupt) generated in module of PMC engine. bb : RAM PARITY ERROR information. 1, 2, 4, 8 Parity error occurred on basic DRAM. 14, 18 Parity error occurred on option DRAM. 20, 60, A0, E0 Parity error occurred on SRAM. xxxx : Segment selector where system error occurred. yyyyyy: Offset address where system error occurred.

System alarm messages 2 (PMC–SC)

Message STATUS LED	Contents and solution
PC150 NMI SLC aa cc STATUS LED ■★	The communication error occurred in the I/O Link. aa, cc : I/O Link error information. This error may occur by the following causes. 1. When I/O Unit–MODEL A is used, base1, 2 or 3 is not connected though allocated. 2. The connection of cable is insufficient. 3. Defects of cable. 4. Defects of DI/DO units (I/O unit, Power Mate etc.) 5. Defects of PMC board (printed circuit board on host side where I/O Link cable is connected.) (solution) Investigate the cause of error. 1. Please confirm the allocation data (by ”EDIT”→”MODULE” screen) and compare with the actual connection. 2. Please confirm whether the cable is correctly connected. If you cannot find the cause with the ways above, it may be the defect of hardware. Please investigate a defective place by the following methods. 3. Please confirm the specification of the cable referring to ”FANUC I/O Unit–MODEL B connecting manuals(B–62163E)”. 4. Exchange the interface module of I/O Unit, the cable and the PMC board, etc. one by one and, confirm whether this error occurs again. The communication may fail by the noise etc. when this error still occurs after replacing all DI/DO units. Please investigate the cause of noise.
PC160 F–BUS ERROR xxxx:yyyyyyyy PC161 F–BUS ERROR xxxx:yyyyyyyy PC162 F–BUS ERROR xxxx:yyyyyyyy STATUS LED ★□	A bus error occurred on the PMC. xxxx : Segment selector for which a bus error occurred. yyyyyyyy : Offset address where a bus error occurred.
PC170 L–BUS ERROR xxxx:yyyyyyyy PC171 L–BUS ERROR xxxx:yyyyyyyy PC172 L–BUS ERROR xxxx:yyyyyyyy STATUS LED ★□	A bus error occurred on the PMC. xxxx : Segment selector for which a bus error occurred. yyyyyyyy : Offset address where a bus error occurred.
PC199 ROM PARITY eeeeeeee STATUS LED ★☆	The parity error occur in PMC system ROM. eeeeeeee : ROM parity error information.

STATUS LED (green) are LED1, LED2 on PMC–RC. CAP–II is LED3 and LED4.

□ : Off ■ : On ☆★ : Blinking

NOTE

1 The system error on PMC–SA1,SA2,SA3,SB,SB2 and SB3 is displayed as a system error on the CNC side.
(Refer to the “FANUC Series 16–MA Operator’s Manual (B–61874E)” and “FANUC Series 16–TA Operator’s Manual (B–61804E)”.)

2 Error information is needed to investigate on FANUC, please take notes of it.

System alarm messages 3 (PMC–NB/NB2)

Message STATUS LED	Contents and solution
RAM ERROR <a> bbcc xxxx: yyyyyyyy: PC010 STATUS LED ★■ or □★	The parity error occurs on the debugging RAM of PMC. a : RAM which generates RAM parity. B BASIC RAM O OPTION RAM S STATIC RAM bb, cc : RAM PARITY information. xxxx : Segment selector where system error occurred. yyyyyyyy : Offset address where system error occurred.
ROM ERROR aaaaaaaa: PC020 STATUS LED ☆★	The parity error occurs in PMC system ROM. aaaaaaaa : ROM parity information
DIVIDE ERROR xxxx: yyyyyyyy: PC040 STATUS LED ☆★	Division error occurs such as a divisor is 0 in the division instruction. xxxx : Segment selector where system error occurred. yyyyyyyy : Offset address where system error occurred.
BUS ERROR xxxx: yyyyyyyy: PC040 STATUS LED ☆★	The BUS error (access on illegal address). xxxx : Segment selector where system error occurred. yyyyyyyy : Offset address where system error occurred.
STACK FAULT xxxx: yyyyyyyy: PC040 STATUS LED ☆★	The stack exception such as the violation of the limit of the stack. xxxx : Segment selector where system error occurred. yyyyyyyy : Offset address where system error occurred.
GENERAL PROTECTION xxxx: yyyyyyyy: PC040 STATUS LED ☆★	The general protection exception such as segment limit over was generated. xxxx : Segment selector where system error occurred. yyyyyyyy : Offset address where system error occurred.
SLC ERROR aa (cc) : PC050 STATUS LED ■★	The communication error occurred in the I/O Link. aa, cc : I/O Link error information. This error may occur by the following causes. 1.When I/O Unit–MODEL A is used, base1, 2 or 3 is not connected though allocated. 2.The connection of cable is insufficient. 3.Defects of cable. 4.Defects of DI/DO units (I/O unit, Power Mate etc.) 5.Defects of PMC board (printed circuit board on host side where I/O Link cable is connected.) (solution) Investigate the cause of error. 1.Please confirm the allocation data (by ”EDIT”→”MODULE” screen) and compare with the actual connection. 2.Please confirm whether the cable is correctly connected. If you cannot find the cause with the ways above, it may be the defect of hardware. Please investigate a defective place by the following methods. 3.Please confirm the specification of the cable referring to ”FANUC I/O Unit–MODEL B connecting manuals(B–62163E)”. 4.Exchange the interface module of I/O Unit, the cable and the PMC board, etc. one by one and, confirm whether this error occurs again. The communication may fail by the noise etc. when this error still occurs after replacing all DI/DO units. Please investigate the cause of noise.

STATUS LED (green) are LED1, LED2 on PMC–NB.

□ : Off ■ : On ☆★ : Blinking

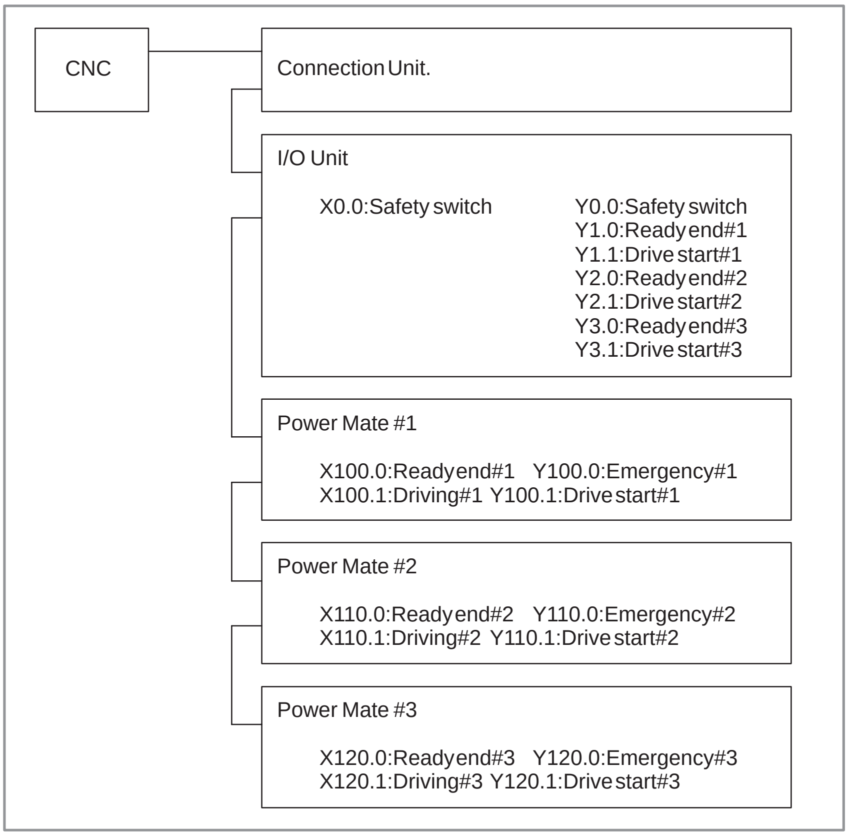
System alarm messages (PMC–RB5/RB6)

Message	Contents and solution
PC0nn CPU INTERRUPT xxxxxxxx	CPU error nn :Exception code xxxxxxxx :Address at which an error occurred
PC004 CPU ERR xxxxxxxx:yyyyyyyy PC006 CPU ERR xxxxxxxx:yyyyyyyy PC009 CPU ERR xxxxxxxx:yyyyyyyy PC010 CPU ERR xxxxxxxx:yyyyyyyy	A CPU error occurred on the PMC. xxxxxxxx and yyyyyyyy represent an internal error code. This error may be caused by a main board failure. (solution) Replace the main board, then check if this error occurs again. If this error still occurs, contact FANUC with the error status information (system configuration, operation, error occurrence timing, error occurrence frequency, and so forth).
PC030 RAM PARITY aa:bb	A RAM parity error occurred on the PMC. aa and bb represent an internal error code. This error may be caused by a main board failure. (solution) Replace the main board, then check if this error occurs again. If this error still occurs, contact FANUC with the error status information (system configuration, operation, error occurrence timing, error occurrence frequency, and so forth) and the internal error code above.
PC040 NMI BOC xxxxxxxx	A non–maskable interrupt (NMI) occurred in the PMC control module. xxxxxxxx :Address at which an error occurred
PC050 NMI SLC aa:bb	A communication error occurred on the I/O Link. aa and bb represent an internal error code. This error may occur as a result of one of the following: (1) When I/O Unit A is used, an I/O assignment is made for base expansion, but no base is connected. (2) Cables are not connected correctly. (3) Cables are faulty. (4) I/O devices (I/O Unit, Power Mate, and so forth) are faulty. (5) The main board is faulty. (solution) (1) Check if the I/O assignment data matches the actual connections of the I/O devices. (2) Check if the cables are connected correctly. (3) Check the cable specifications by referring to the ”FANUC I/O Unit–MODEL A Connection and Maintenance Manual (B–61813E)” or ”FANUC I/O Unit–MODEL B Connection Manual (B–62163E). (4) Replace the interface module, cables, and main board of the I/O Unit, then check whether this error occurs again. If this error still occurs after replacement of all the devices related to the I/O Link according to Action (4), the communication error may have been caused by noise. Attempt to identify the source of the noise.
PC060 FBUS xxxxxxxx:yyyyyyyy PC061 FL–R xxxxxxxx:yyyyyyyy PC062 FL–W aa:xxxxxxxx:yyyyyyyy	A bus error occurred on the PMC. aa, xxxxxxxx, and yyyyyyyy represent an internal error code. This error may be caused by a hardware failure. (solution) Contact FANUC with the error status information (system configuration, operation, error occurrence timing, error occurrence frequency, and so forth), the internal error code above, and the LED statuses on each board.

Message	Contents and solution
PC070 SUB65 CALL (STACK)	A stack error occurred with the ladder functional instruction CALL/CALLU. (solution) Check the correspondence between the CALL/CALLU instruction and the SPE instruction. If the cause of the fault cannot be found, contact FANUC with the error status information and the ladder program.
PC080 SYS EMG xxxxxxxx:yyyyyyyy PC081 FL EMG xxxxxxxx:yyyyyyyy	A system alarm was issued due to other software. (solution) Contact FANUC with the error status information (system configuration, operation, error occurrence timing, error occurrence frequency, and so forth), the internal error code above, and the LED statuses on each board.
PC097 PARITY ERR (LADDER) PC098 PARITY ERR (DRAM) PC099 PARITY ERR (SRAM)	A parity error occurred on the PMC system. This error may be caused by a main board failure. (solution) Replace the main board, then check whether this error occurs again. If this error still occurs, contact FANUC with the error status information (system configuration, operation, error occurrence timing, error occurrence frequency, and so forth).

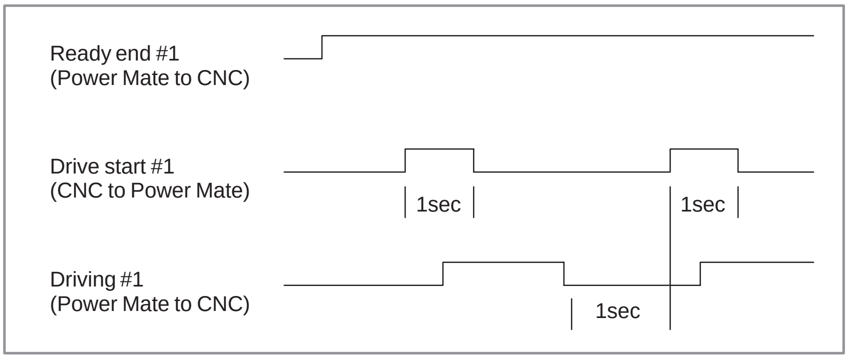
N EXAMPLE OF STEP SEQUENCE PROGRAMS

The CNC is connected two or three Power Mate units.

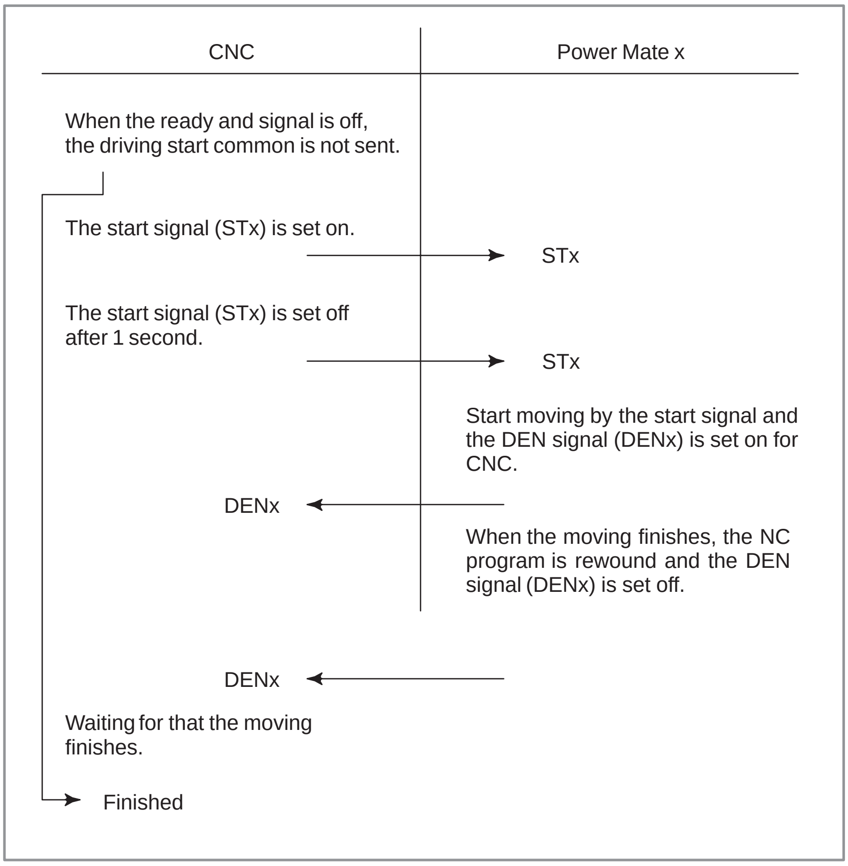


> **NOTE**
> The addresses indicate the single addresses, as viewed from the CNC.

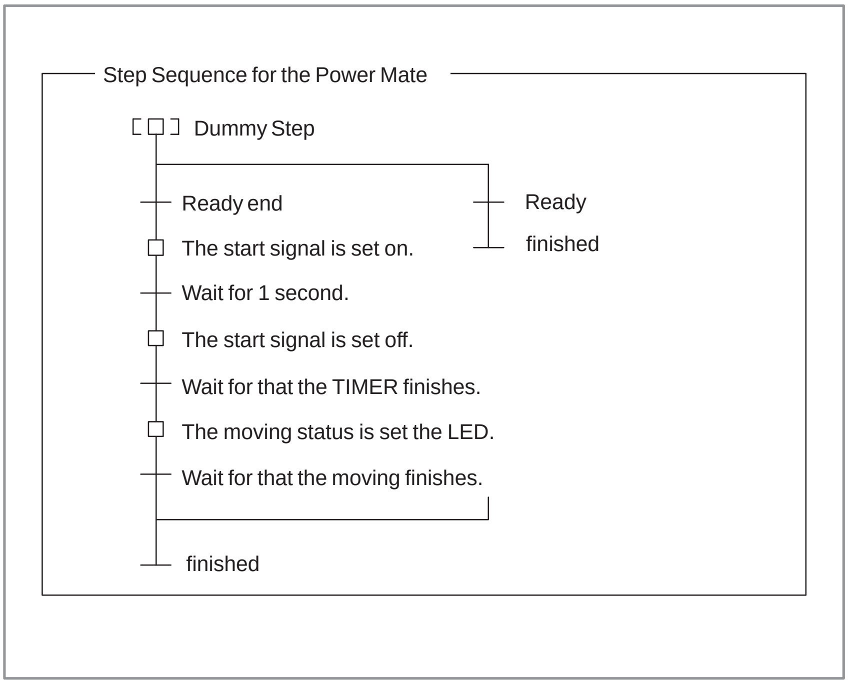
The CNC controls the Power Mate units at the following signal timing.



The following flowchart illustrates the interface with the Power Mate units.

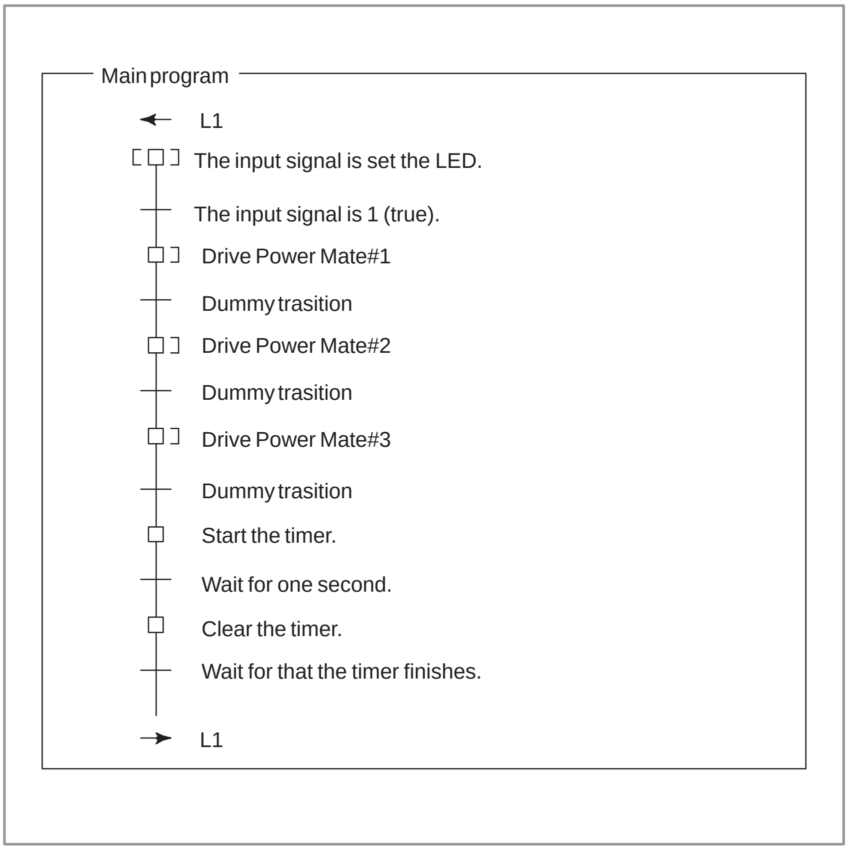


The interface with the Power Mate units is changed to the Step Sequence program.



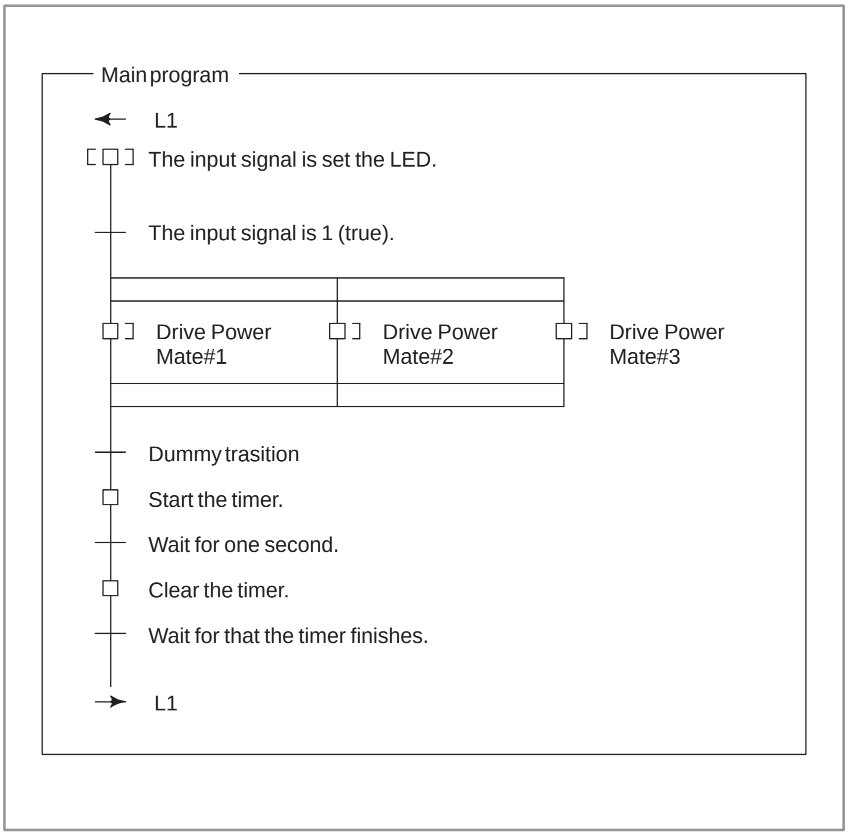
Example 1

The Step sequence program for three sequentially driven Power Mate units:



Example 2

The Step Sequence program for three simultaneously driven Power Mate units:

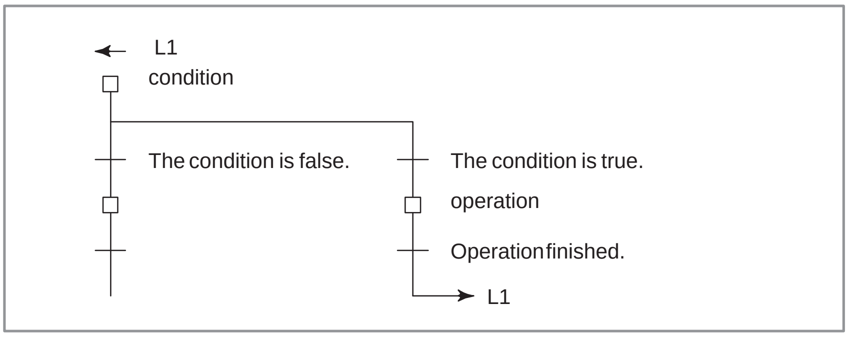


O STEP SEQUENCE CORRESPONDED C LANGUAGE

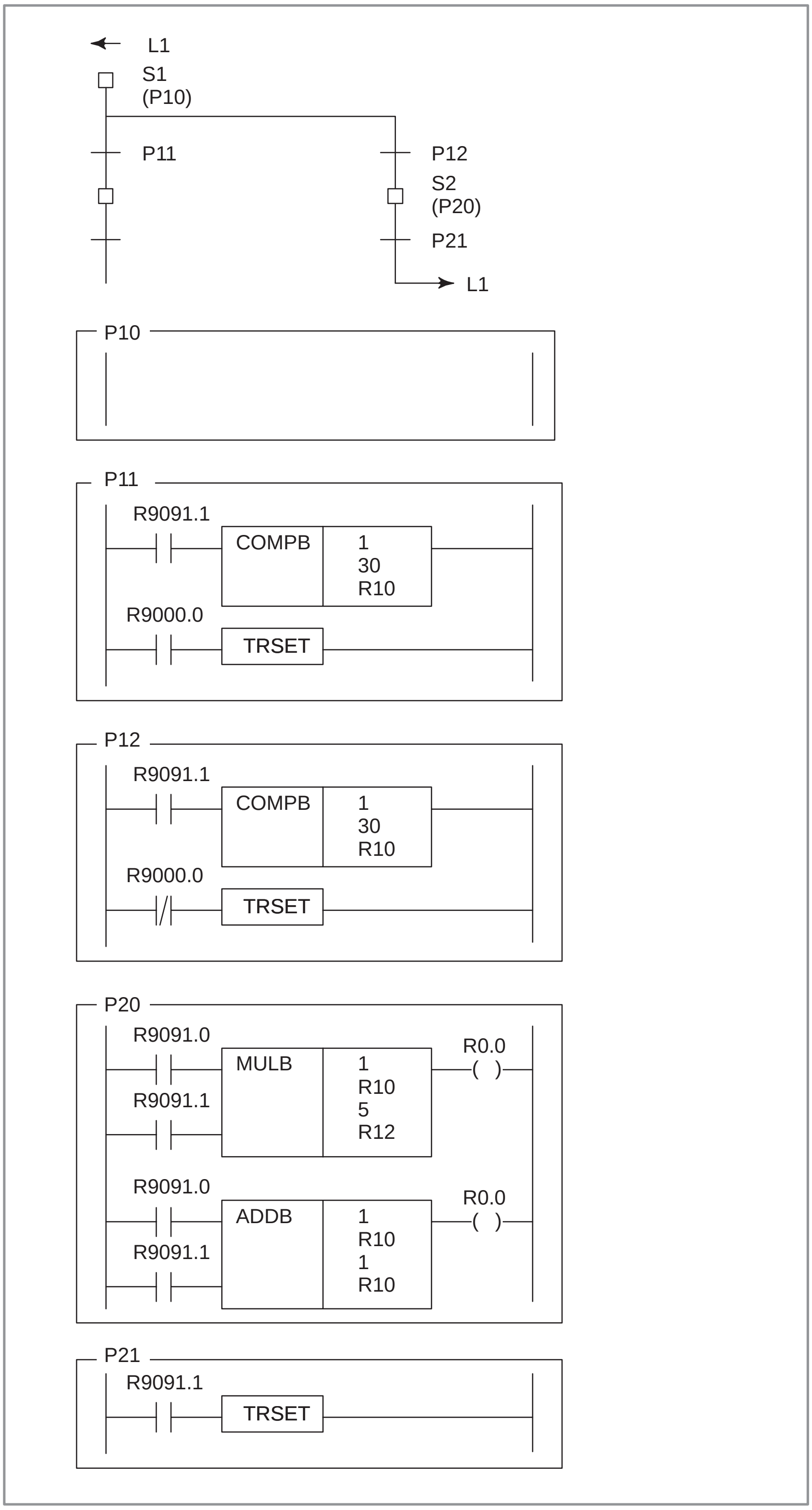
O.1 WHILE STATEMENT

The operation is continued while the condition is true.

Format



Examples

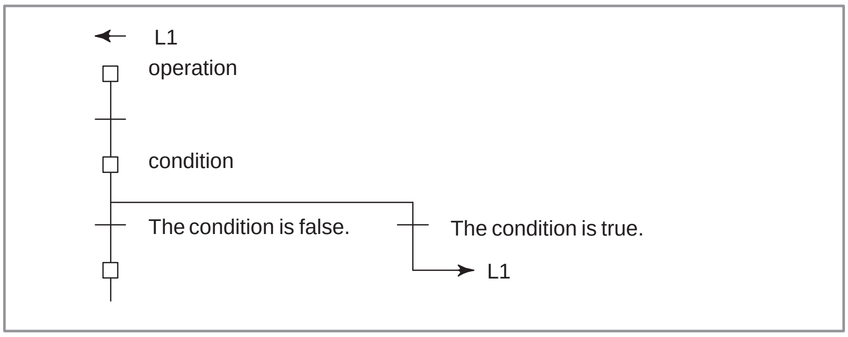

L1
S1
(P10)
P11
P12
S2
(P20)
P21
L1
P10
P11
R9091.1
COMPB
1
30
R10
R9000.0
TRSET
P12
R9091.1
COMPB
1
30
R10
R9000.0
TRSET
P20
R9091.0
MULB
1
R10
5
R12
R0.0
R9091.1
R9091.0
ADDB
1
R10
1
R10
R0.0
R9091.1
P21
R9091.1
TRSET

O.2 DO–WHILE STATEMENT

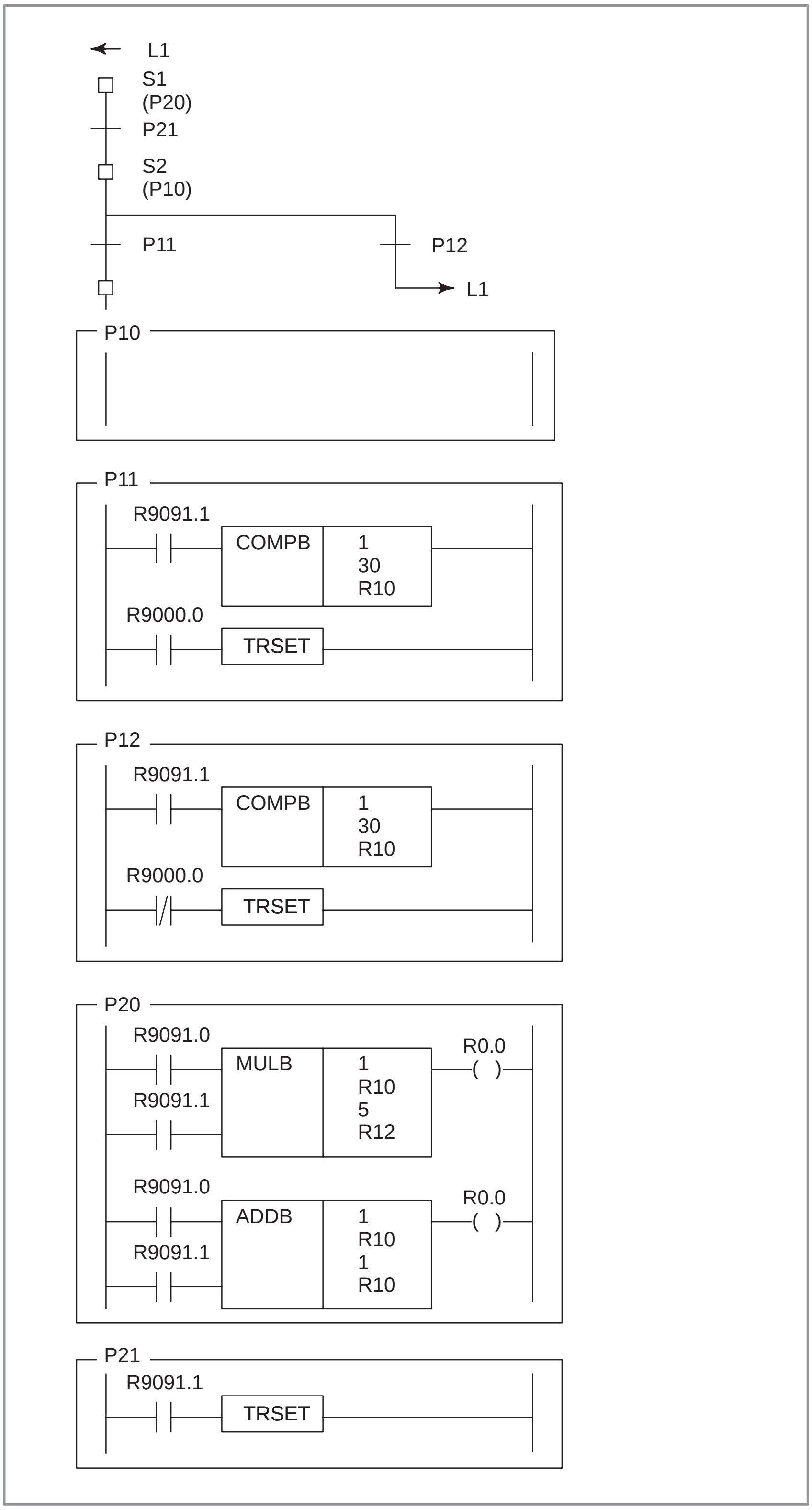
The operation is continued while the condition is true after executing the operation.

The difference between do–while and while is that the operation is executed at least one time.

Format



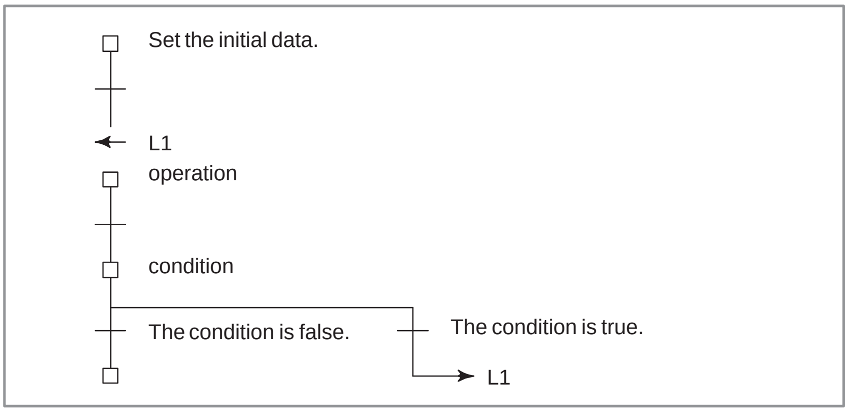
Examples


L1
S1
(P20)
P21
S2
(P10)
P11
P12
L1
P10
P11
R9091.1
COMPB
1
30
R10
R9000.0
TRSET
P12
R9091.1
COMPB
1
30
R10
R9000.0
TRSET
P20
R9091.0
MULB
1
R10
5
R12
R0.0
R9091.1
R9091.0
ADDB
1
R10
1
R10
R0.0
R9091.1
P21
R9091.1
TRSET

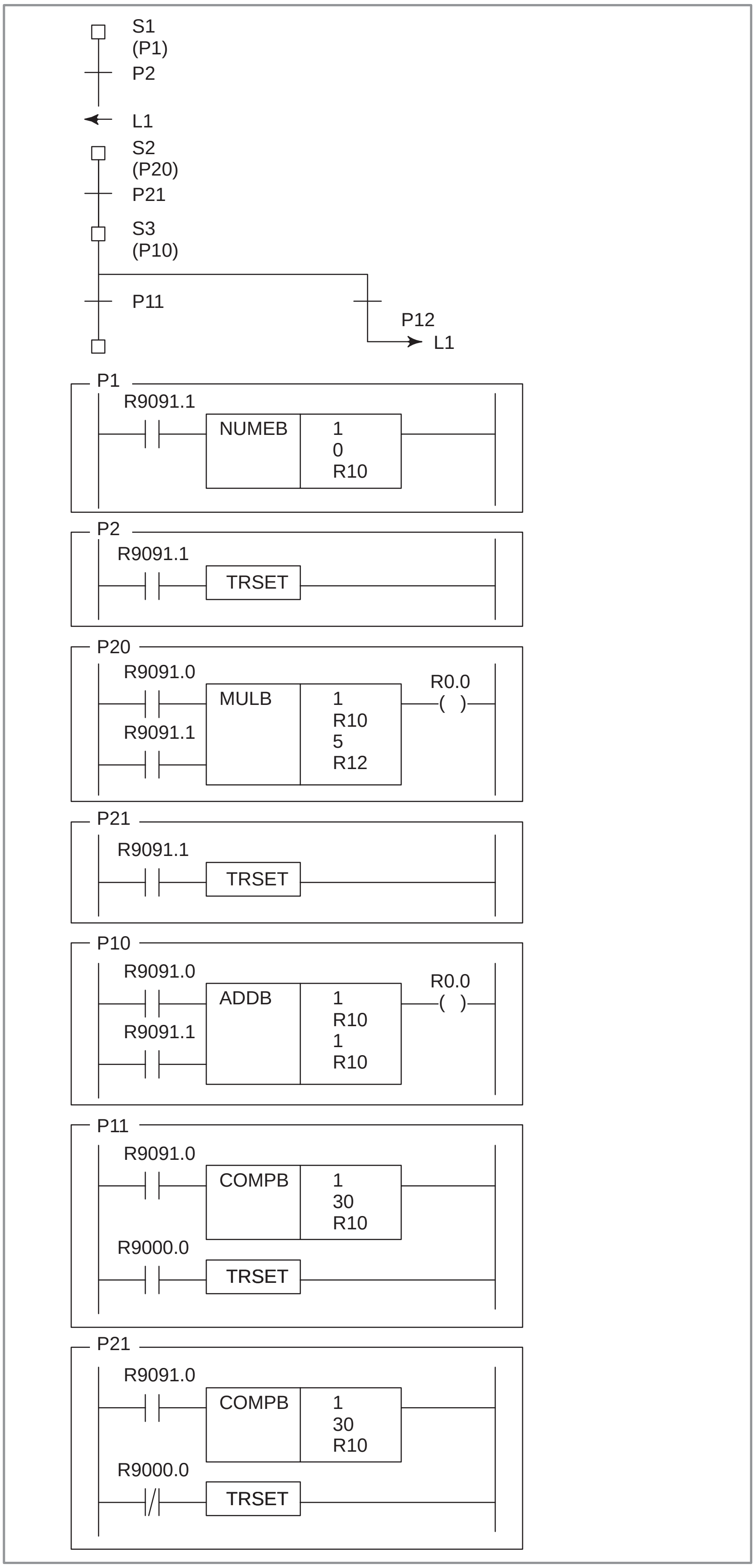
O.3 FOR STATEMENT

After the initial data is set, the operation is continued while the condition is true.

Format



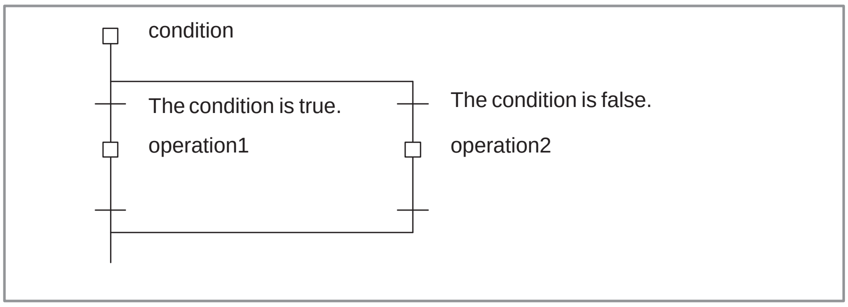
Examples


S1
(P1)
P2
L1
S2
(P20)
P21
S3
(P10)
P11
P12
L1
P1
R9091.1
NUMEB
1
0
R10
P2
R9091.1
TRSET
P20
R9091.0
MULB
1
R10
5
R12
R0.0
R9091.1
P21
R9091.1
TRSET
P10
R9091.0
ADDB
1
R10
1
R10
R0.0
R9091.1
P11
R9091.0
COMPB
1
30
R10
R9000.0
TRSET
P21
R9091.0
COMPB
1
30
R10
R9000.0
TRSET

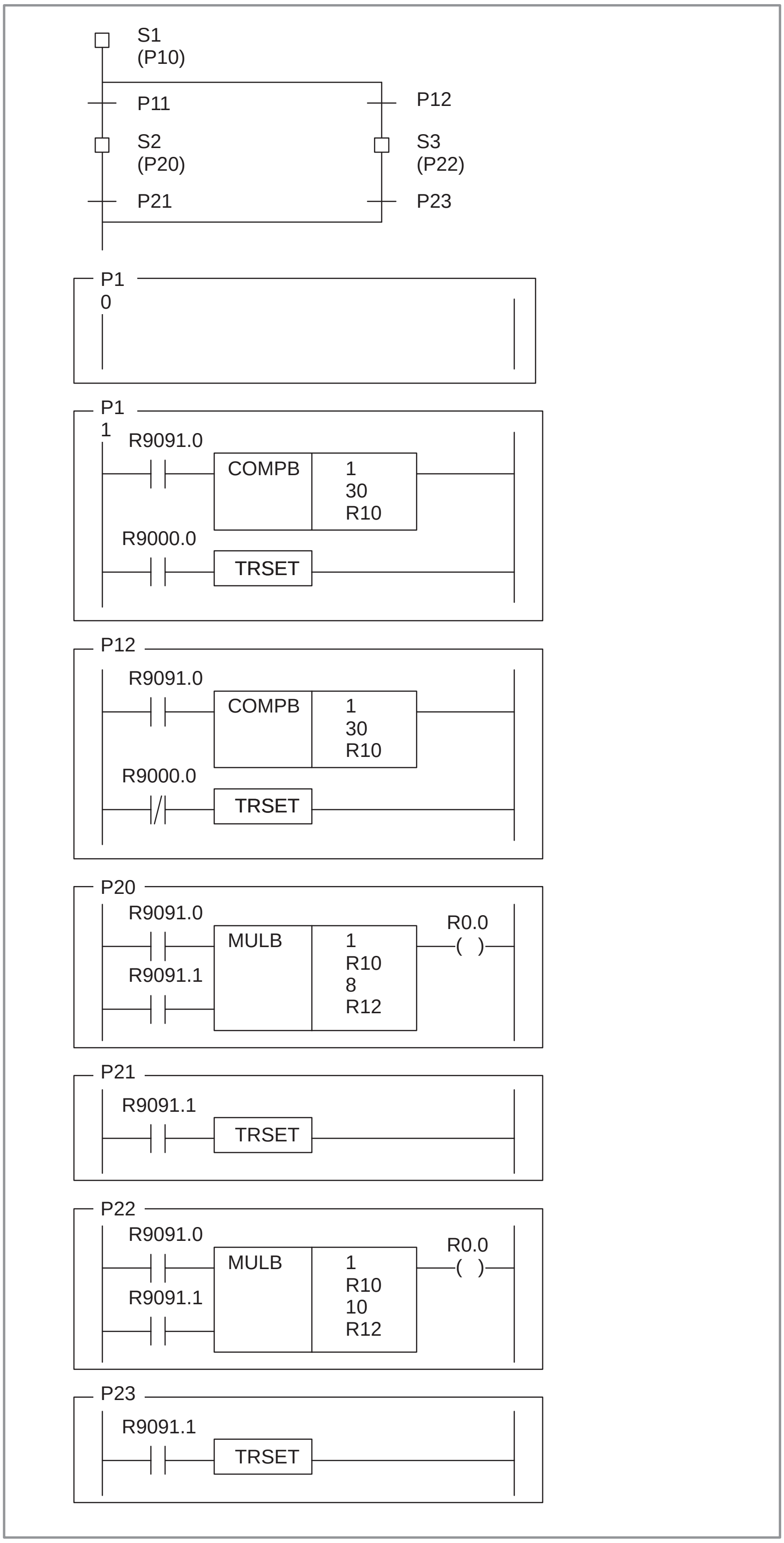
O.4 IF ELSE STATEMENT

If the condition is true, the operation 1 is executed and if the condition is false, the operation 2 is executed.

Format



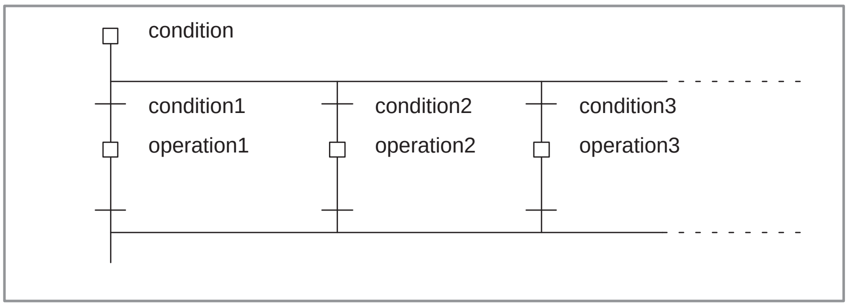
Examples


S1
(P10)
P11
P12
S2
(P20)
S3
(P22)
P21
P23
P1
0
P1
1
R9091.0
COMPB
1
30
R10
R9000.0
TRSET
P12
R9091.0
COMPB
1
30
R10
R9000.0
TRSET
P20
R9091.0
MULB
1
R10
8
R12
R0.0
R9091.1
P21
R9091.1
TRSET
P22
R9091.0
MULB
1
R10
10
R12
R0.0
R9091.1
P23
R9091.1
TRSET

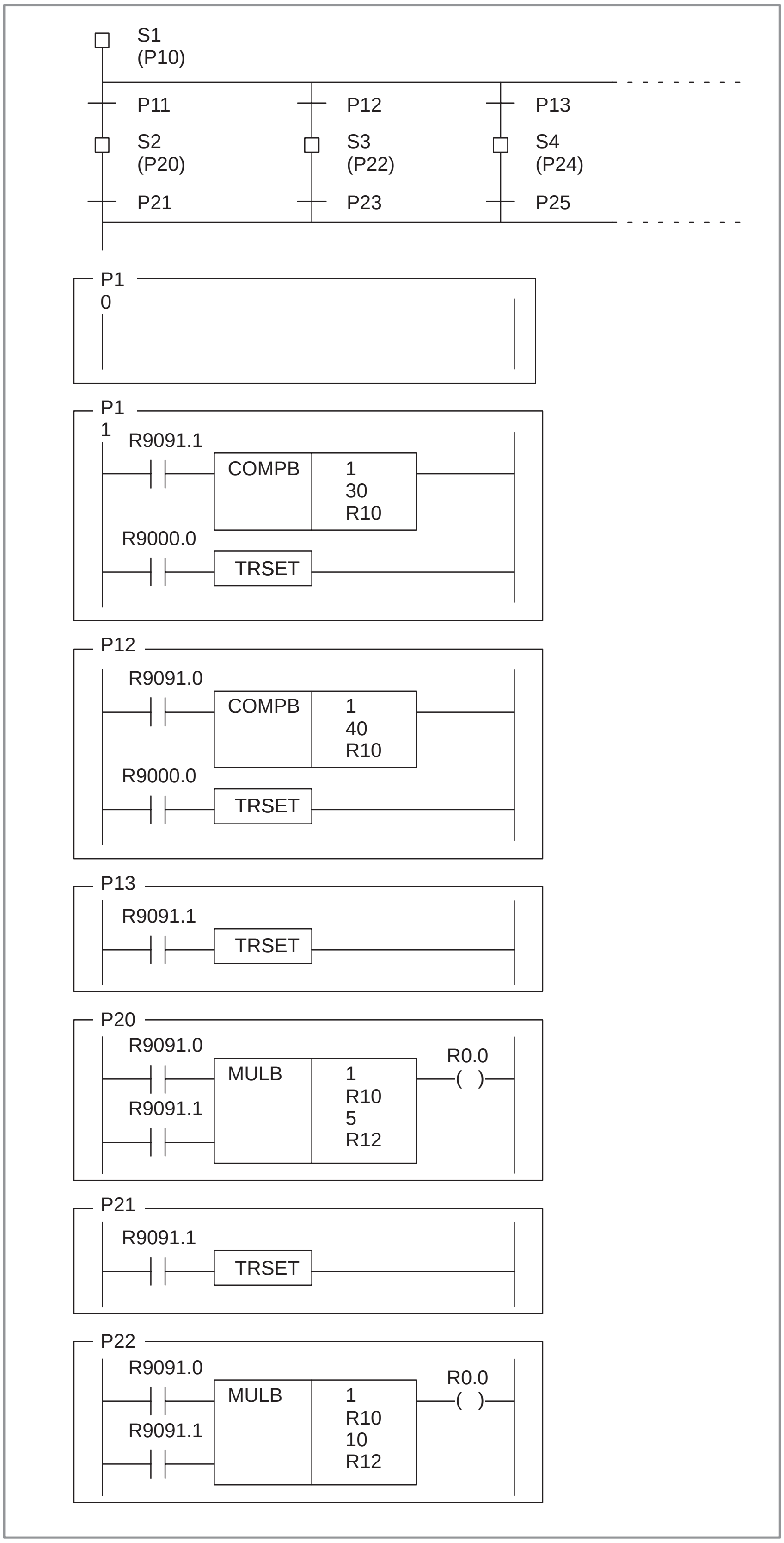
O.5 SWITCH STATEMENT

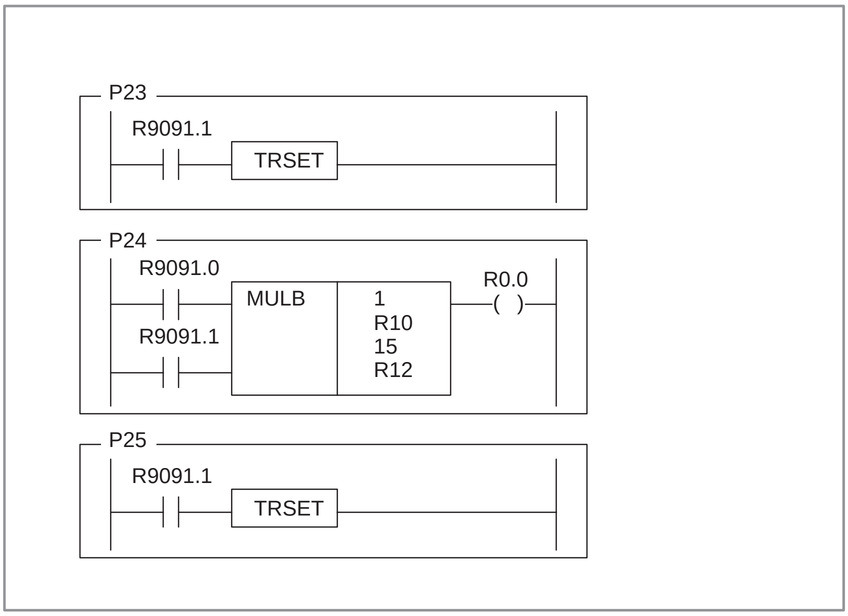
The operation connected to the condition is executed.

Format



Examples


S1
(P10)
P11
P12
P13
S2
(P20)
S3
(P22)
S4
(P24)
P21
P23
P25
P10
P11
R9091.1
COMPB
1
30
R10
R9000.0
TRSET
P12
R9091.0
COMPB
1
40
R10
R9000.0
TRSET
P13
R9091.1
TRSET
P20
R9091.0
MULB
1
R10
5
R12
R0.0
R9091.1
P21
R9091.1
TRSET
P22
R9091.0
MULB
1
R10
10
R12
R0.0
R9091.1


P23
R9091.1
TRSET
P24
R9091.0
MULB
1
R10
15
R12
R0.0
R9091.1
P25
R9091.1
TRSET

P CHINESE CHARACTER CODE, HIRAGANA CODE, AND SPECIAL CODE LIST

Index

Note
Volume 1 : Up to Page 759 / Volume 2 : Page 761 and later

A

B

C

Note
Volume 1 : Up to Page 759 / Volume 2 : Page 761 and later

D

E

Note
Volume 1 : Up to Page 759 / Volume 2 : Page 761 and later

Note
Volume 1 : Up to Page 759 / Volume 2 : Page 761 and later

J

K

L

M

N

O

Note
Volume 1 : Up to Page 759 / Volume 2 : Page 761 and later

Note
Volume 1 : Up to Page 759 / Volume 2 : Page 761 and later

Note
Volume 1 : Up to Page 759 / Volume 2 : Page 761 and later

S

Note
Volume 1 : Up to Page 759 / Volume 2 : Page 761 and later

Note
Volume 1 : Up to Page 759 / Volume 2 : Page 761 and later

X

Revision Record

FANUC PMC–MODEL PA1/PA3/SA1/SA2/SA3/SA5/SB/SB2/SB3/SB4/SB5/SB6/SC/SC3/SC4/NB/NB2/NB6

LADDER LANGUAGE PROGRAMMING MANUAL (B–61863E)

Edition	Date	Contents	Edition	Date	Contents
06	Nov.,'94	Corresponds to 18–B	12	Mar., '99	Addition PMC–NB6
05	May,'94	Addition of PMC–MODEL RB4/RC4 Addition of the following Appendix. • Window function description (FS16–LA) • Window function description (FS16–W) • Window function description (FS16PA) • PMC MODEL RA1/RA3 Supplementary Explanation of Programming	11	Nov., '98	Addition Power Mate *i*
04	Aug.,'93	Addition of PMC–MODEL PA1/PA3/RA3/RB3/RC3/NB.	10	Dec., '97	Corresponds to 16*i*/18*i*/21*i*–MODEL A
03	Mar., '92	Addition of PMC–MODEL RA1/RA2/RB2	09	Mar.,'96	Addition PMC–NB2
02	Aug., '91	All pages are revised. PMC–MODEL RC is added.	08	Oct.,'95	Addition of PMC–RB5/RB6
01	Oct., '90	————	07	Apr.,'95	Total revision