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 (Volume 1 of 2)

GFZ-61863E/14 July 2001

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–MODEL A FANUC Series 21–MODEL B FANUC Series 20*i*–MODEL A FANUC Series 0*i*–MODEL A FANUC Series 21*i*–MODEL A/B 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–MODEL A FANUC Series 21–MODEL B FANUC Series 0*i*–MODEL A
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

Product Name	Abbreviations	Applicable CNC
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
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 Series 16–MODEL C FANUC Series 18–MODEL C FANUC Series 16*i*–MODEL A FANUC Series 18*i*–MODEL A FANUC Series 21*i*–MODEL A FANUC Power Mate *i*–MODEL D/H
FANUC PMC–MODEL SB7	PMC–SB7	FANUC Series 16*i*–MODEL B FANUC Series 18*i*–MODEL B FANUC Series 21*i*–MODEL B
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*/18*i*/21*i*–MODEL A/B

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 16*i*/18*i*/21*i*/160*i*/180*i*/210*i*–MODEL B CONNECTION MANUAL (FUNCTION) (B-63523EN-1)	Interface between PMC and CNC	PMC-SB7
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 Series 15*i*/150*i*-MODEL A CONNECTION MANUAL (FUNCTION) (B–63323EN–1)	Interface between PMC and CNC	PMC-NB6
FANUC PMC 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)

Volume 2

VI. STEP SEQUENCE FUNCTION

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.

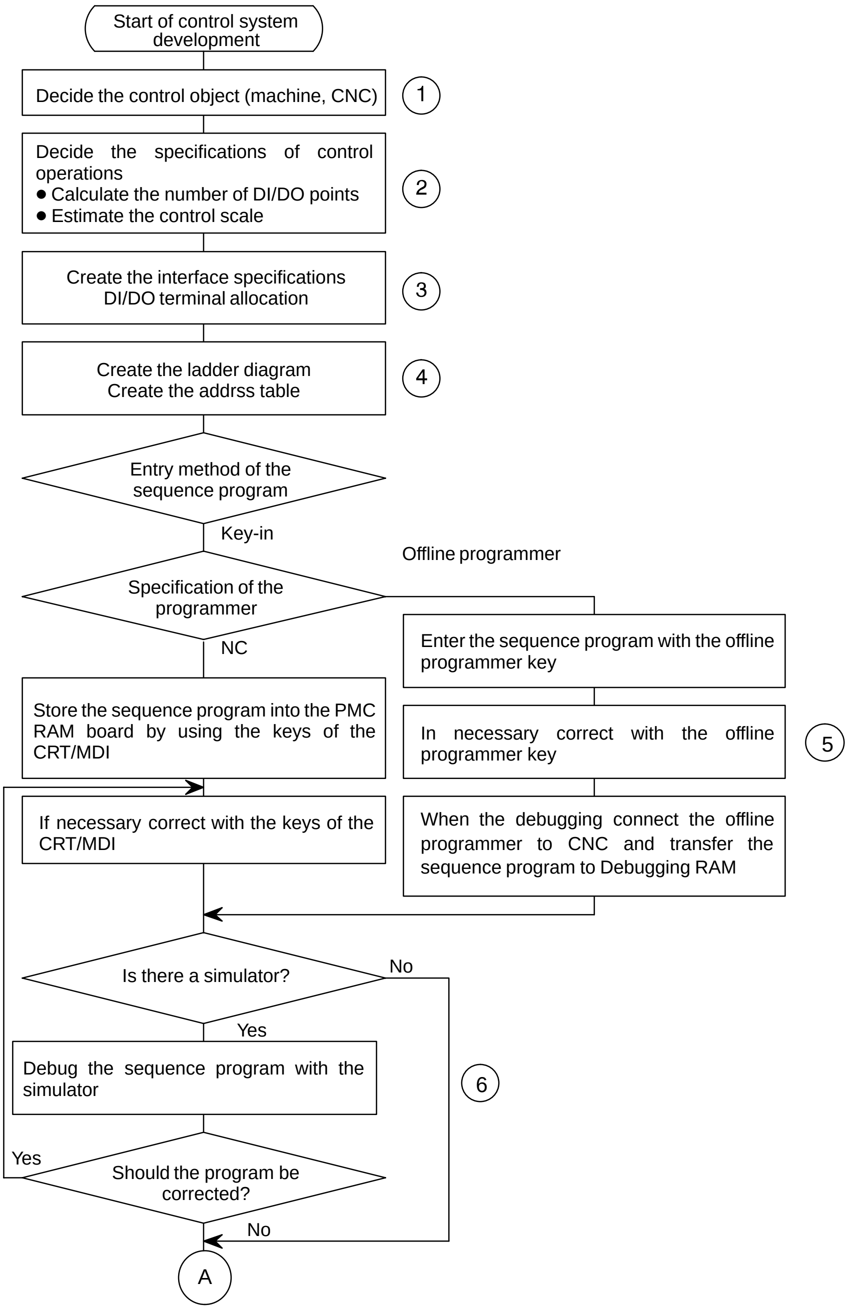


Fig. 1 Sequence program creating procedure (1/2)

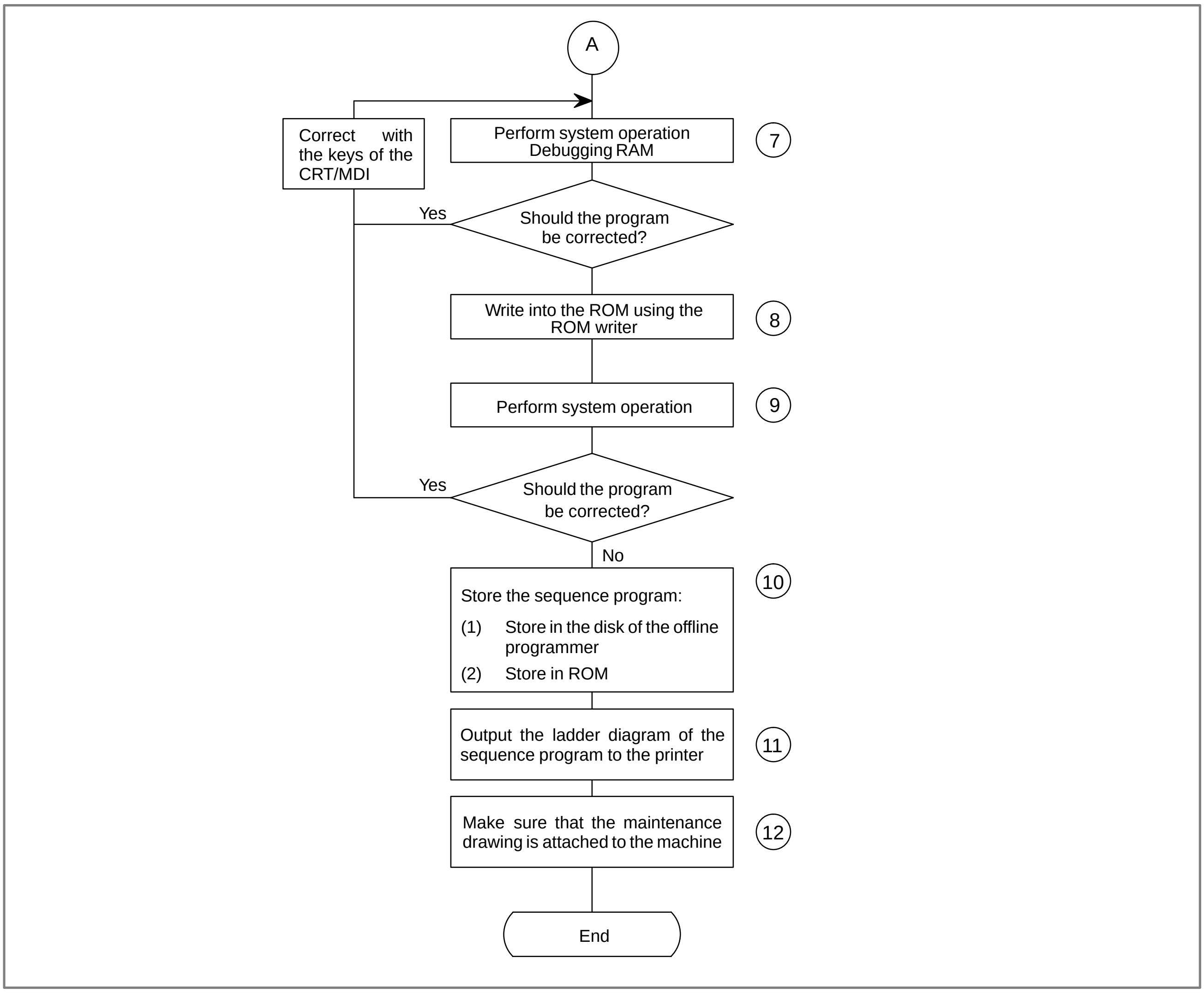


Fig. 1 Sequence program creating procedure (2/2)

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) (Note 1,3) ● Symbol, Comment (Note 2,3) ● Message (Note 3) ● 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 4) (I) (Master) (O) ● I/O Link (Note 7) (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 5)	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. This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.
2. 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.
3. These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.
4. I/O Link Master function is not available in the Power Mate–MODEL F.
5. FLASH ROM is used in the Power Mate–MODEL H.
6. 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.
7. 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) (Note 1,3) ● Symbol, Comment (Note 2,3) ● Message (Note 3) ● 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 4)	EPROM 1Mbit×1 (128KB)	EPROM 1Mbit×1 (128KB) (Note 4)

NOTE

1 This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.

2 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.

3 These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.

4 FLASH ROM is used in the FANUC Series 20.

5 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.

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

Table 1.1 PMC specifications (3)

Type of PMC Specification 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) (Note 1,3,4) ● Symbol, Comment (Note 2,4) ● Message (Note 4) ● 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) (Note 5) (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 3)	EPROM 1Mbit×1 (128KB) ROM MODULE 256KB (Note 3)

NOTE

1. This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.
2. 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.
3. 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.
4. These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.
5. 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) (Note 1,3) ● Symbol, Comment (Note 2,3) ● Message (Note 3) ● Language only	 Approx. 16,000 Approx. 24,000 1 to 128KB 0.1 to 64KB 896KB max.	 Approx. 16,000 Approx. 24,000 1 to 128KB 0.1 to 64KB 896KB max.	 Approx. 8,000 Approx. 16,000 (Note 4) Approx. 24,000 (Note 4) 1 to 128KB 0.1 to 64KB 896KB max.
Command Basic command Function command	12 kinds 51 kinds	14 kinds 68 kinds	14 kinds 68 kinds
Internal relay (R) Message request (A) Keepmemor ● Variable timer (T) ● Counter (C) ● Keep relay (K) ● Data table (D) Subprogram (P) Label (L) Fixed timer	1600 byte 25 byte 80 byte 80 byte 20 byte 3000 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	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. – –
Sequence program	ROM MODULE 128KB (16,000 steps option) 256KB (24,000 steps option) 512KB (24,000 steps option) 1MB (24,000 steps option)	ROM MODULE 128KB (16,000 steps option) 256KB (24,000 steps option) 512KB (24,000 steps option) 1MB (24,000 steps option)	Flash ROM 64KB (8,000 steps) 128KB (16,000 steps) 256KB (24,000 steps) 512KB (24,000 steps) 1MB (24,000 steps)

NOTE

1 This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.
2 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.
3 These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.
4 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) (Note 1,3) ● Symbol/Comment (Note 2,3) ● Message (Note 3) ● Language only	 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. 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 1 to 128KB 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) 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	1100 byte 25 byte 80 byte 80 byte 20 byte 1860 byte – – Timer No. 100 devices specified
Input/output ● I/O link (I) Max. (O) Max. ● I/O card (I) Max. (Note 4) (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.	 1024 points max. 1024 points max. 312 points max. 240 points max.
Sequence program storage media	Flash ROM 128KB (16,000 steps option or less) 256KB (24,000 steps option)	Flash ROM 128KB (16,000 steps option) 256KB (24,000 steps option) 512KB (24,000 steps option) 1MB (24,000 steps option)	Flash ROM 128KB (16,000 steps option or less) 256KB (24,000 steps option)	Flash ROM 128KB (16,000 steps option) 256KB (24,000 steps option) 512KB (24,000 steps option) 1MB (24,000 steps option)	Flash ROM 128KB

NOTE

1 This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.
2 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.
3 These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.
4 That is the maximum number when 2 I/O cards (with 156 inputs/120 outputs) are used.
5 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.
6 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) (Note 1,3) ● Symbol/Comment (Note 2,3) ● Message (Note 3) ● 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 4) (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 (16,000 steps option or less) 256KB (24,000 steps option)	Flash ROM 128KB (16,000 steps option) 256KB (24,000 steps option) 512KB (24,000 steps option) 1MB (24,000 steps option)	Flash ROM 128KB (16,000 steps option or less) 256KB (24,000 steps option)	Flash ROM 128KB (16,000 steps option) 256KB (24,000 steps option) 512KB (24,000 steps option) 1MB (24,000 steps option)

NOTE

1 This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.
2 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.
3 These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.
4 That is the maximum number when 2 I/O cards (with 156 inputs/120 outputs) are used.
5 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) (Note 1,4) ● Symbol/Comment (Note 2,4) (Note 4) ● 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 Approx.16, 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 5)	 1024 points max. 1024 points max. 96 points max. 72 points max. (Note 5)
Sequence program	Flash ROM 128KB (Note 3)	Flash ROM 128KB (Note 5)

NOTE

1 This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.

2 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.

3 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.

4 These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.

5 Output points of I/O card in 4082 series are following ; PMC–SA1 : 64points, PMC–SA3 : 64points

6 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.

7 Application PMC for FANUC Series 21–B loader control function is PMC–SA1.

Table 1.1 PMC specifications (8)

Model	FANUC Series 16*i*/160*i*/18*i*/180*i*		
	PMC–SA5 (Loader control)	**PMC–SB5**	**PMC–SB6**
Programmingmethod	Ladder	Ladder	Ladder step sequence
Number of ladder levels	2	2	2
First–level execution period	8 ms	8 ms	8 ms
Basic instruction processing time	5.0 μ sec/step	0.085 μ sec/step	0.085 μ sec/step
Program capacity ● Ladder (step) (Note 1,2) ● Symbol & Comment (Note 2) ● Message (Note 2)	 About 3,000 About 5,000 About 8,000 About 12,000 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 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) (Functional instruction)	12 48	14 66	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 (Timer number specification)	1100 bytes 25 bytes (200 points) 80 bytes (40 each) 80 bytes (20 each) 20 bytes 1,860 bytes – – 100 each	1618 bytes 25 bytes (200 points) 80 bytes (40 each) 80 bytes (20 each) 20 bytes 3,000 bytes 512 each 9999 each 100 each	3200 bytes 125 bytes (1000 points) 300 bytes (150 each) 200 bytes (50 each) 50 bytes 8,000 bytes 2000 each 9999 each 100 each
I/O ● I/O link (Input) (Note 4) (Output)	 1024 points maximum 1024 points maximum	 1024 points maximum 1024 points maximum	 2048 points maximum (Note 5) 2048 points maximum (Note 5)
Sequence program storage media	Flash ROM 128KB	Flash ROM 128KB (16,000 steps option or less) 256KB (24,000 steps option or less)	Flash ROM 128KB (16,000 steps option or less) 256KB (24,000 steps option) 384KB(32,000/40,000 steps option)

NOTE

1 This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.
2 These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.
3 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*.
4 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.
5 1024 points maximum (channel 1)+1024 points maximum (channel 2)= 2048 points maximum. You can use the I/O Link channel 2 only when CNC hardware support the I/O Link 2–channel and optional I/O Link expansion is provided with CNC.

Table 1.1 PMC specifications (9)

Model	FANUC Series 21*i*/210*i*			FANUC Series 21*i*/210*i*
	PMC–SA1	**PMC–SA5 (Loader control)**	**PMC–SA5**	**PMC–SB6**
Programming method	Ladder	Ladder	Ladder	Ladder Step sequence
Number of ladder levels	2	2	2	2
First–level execution period	8 ms	8 ms	8 ms	8 ms
Basic instruction processing time	5.0 μ sec/ step	5.0 μ sec/ step	0.085 μ sec/ step	0.085 μ sec/ step
Program capacity ● Ladder (step) (Note 1,2) ● Symbol & Comment (Note 2) ● Message (Note 2)	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 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	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) (Functional instruction)	12 49	12 49	14 66	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 (Timer number specification)	1100 bytes 25 bytes (200 points) 80 bytes (40 each) 80 bytes (20 each) 20 bytes 1860 bytes – – 100 each	1100 bytes 25 bytes (200 points) 80 bytes (40 each) 80 bytes (20 each) 20 bytes 1,860 bytes – – 100 each	1118 bytes 25 bytes (200 points) 80 bytes (40 each) 80 bytes (20 each) 20 bytes 1860 bytes 512 each 9999 each 100 each	3200 bytes 125 bytes (1000 points) 300 bytes (150 each) 200 bytes (50 each) 50 bytes 8,000 bytes 2000 each 9999 each 100 each
I/O ● I/O link (Input) (Note 4) (Output)	1024 points maximum 1024 points maximum	1024 points maximum 1024 points maximum	1024 points maximum 1024 points maximum	2048 points maximum (Note 5) 2048 points maximum (Note 5)
Sequence program storage media	Flash ROM 128KB	Flash ROM 128KB	Flash ROM 128KB	Flash ROM 128KB (16,000 steps option or less) 256KB (24,000 steps option) 384KB(32,000/40,000 steps option)

NOTE

1 This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.
2 These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.
3 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*.
4 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.
5 1024 points maximum (channel 1)+1024 points maximum (channel 2)= 2048 points maximum. You can use the I/O Link channel 2 only when CNC hardware support the I/O Link 2–channel and optional I/O Link expansion is provided with CNC.

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) (Note 1,2) ● Symbol/Comment (Note 2) ● Message (Note 2)	 Approx. 5,000 Approx. 12,000 Approx. 16,000 Approx. 24,000 1 to 128KB 0.1 to 64KB	 Approx. 5,000 Approx. 12,000 Approx. 16,000 Approx. 24,000 Approx. 32,000 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 (200 points) 80 bytes (40 points) 80 bytes (20 points) 20 bytes 3000 bytes 512 programs 9999 labels Timer No.100 devices specified	3200 bytes 125 bytes (200 points) 300 bytes (150 points) 200 bytes (50 points) 50 bytes 8000 bytes 2000 programs 9999 labels Timer No.100 devices specified
Input/Output ● I/O Link (master) (I) Max. (O) Max. ● I/O Link (slave) (I) Max. (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 (16,000 steps option or less) 256KB (24,000 steps option)	Flash ROM 128KB (16,000 steps option or less) 256KB (24,000 steps option) 384KB (32,000 steps option)

NOTE

1 This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.

2 These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/ comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.

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 1,2) (Note 2) (Note 2)	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 (8,000 steps) 128 KB (16,000 steps) 256 KB (24,000 steps) 512 KB (24,000 steps) 1 MB (24,000 steps)	Flash ROM 64 KB (8,000 steps) 128 KB (16,000 steps) 256 KB (24,000 steps) 512 KB (24,000 steps) 1 MB (24,000 steps)

NOTE

1. This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.
2. These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/ comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.
3. 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 1,2)	About 32,000 maximum
• Symbol/comment	(Note 2)	1 to 128KB
• Message	(Note 2)	0.1 to 64KB
Instruction	(Basic instruction)	14
	(Functional instruction)	64
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 3)	(Output)	1024 points maximum
Sequence program storage media		Flash ROM 128KB (16,000steps option or less) 256KB (24,000 steps) 384KB (32,000steps option)

NOTE

1 This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.

2 These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.

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

Table 1.1 PMC specifications (13)

Model	Series 21*i*–B	Series 16*i*/18*i*/21*i*–B	
	PMC–SA1	PMC–SA1 (Loader control)	PMC–SB7
Programmingmethod	Ladder	Ladder	LadderC–language
Number of ladder levels	2	2	3
First–level execution period	8 ms	8 ms	8 ms
Basic instruction processing time	5.0 µ sec/step	5.0 µ sec/step	0.033 µ sec/step
Program capacity			
• Ladder (step) (Note 1,2)	About 3,000 About 5,000	About 3,000 About 5,000 About 8,000 About 12,000	About 3,000 About 5,000 About 8,000 About 12,000 About 16,000 About 24,000 About 32,000 About 40,000 About 48,000 About 64,000
• Symbol & comment (Note 2)	1 to 128KB	1 to 128KB	1KB or more
• Message (Note 2)	0.1 to 64KB	0.1 to 64KB	8KB or more
Instruction (Basic instruction)	12	12	14
(Functionalinstruction)	48	48	69
Internal Relay (R)	1100 bytes	1,100 byte	8,500 byte
Extra Relay (E)	–	–	8,000 byte
Message Request (A)	200 points (25 byte)	200 points (25 byte)	2,000 points (500 byte, 2 bit/point)
Nonvolatile Memory			
• Data Tables (D)	1,860 byte	1,860 byte	10,000 byte
• Variable Timers (T)	80 bytes (20 each)	40 points (80 byte)	250 points (1,000 byte, 4 byte/point)
• Fixed Timers	20 bytes	100 points	500 points (Timer number specify)
• Counters (C)	1860 bytes	20 points (80 byte)	100 points (400 byte, 4 byte/point)
Fixed Counters (C)	–	–	100 points (200 byte, 2 byte/point)
• Keep Relays (K)	–	20 byte	120 byte
Subprograms (P)	100 each	–	2000
Labels (L)		–	9999
I/O I/O link			
• Input	1024 points maximum	1024 points maximum	2048 points maximum (Note 3)
• Output	1024 points maximum	1024 points maximum	2048 points maximum (Note 3)
Sequence program storage media	Flash ROM 128KB	Flash ROM 128KB	Flash ROM 128KB (16,000steps option or less) 256KB (24,000steps option) 384KB(32,000/40,000steps option) 512KB (48,000steps option) 768KB (64,000steps option)

NOTE

1 This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.
2 These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.
3 Maximum of basic input/output points are 1024/1024 points. I/O Link expansion option extends the maximum to 2048/2048 points.

Table 1.1 PMC specifications (14)

Model	Series 0*i*–A	
	PMC–SA1	**PMC–SA3**
Programmingmethod	Ladder	Ladder
Number of ladder levels	2	2
Level–1 cycle time	8 ms	8 ms
Basic instruction execution time	5.0 μ sec/step	0.15 μ sec/step
Program capacity • Ladder (step) (Note 1) • Symbol/Comment (Note 2,3) • Message (Note 2,3)	 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 Approx. 16,000 1 to 128KB 0.1 to 64KB
Instruction (Basic) (Functional)	12 kinds 49 kinds	14 kinds 66 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	1100 bytes 25 bytes (200 points) 80 bytes (40 each) 80 bytes (20 each) 20 bytes 1860 bytes – – Timer No.100 devices specified	1118 bytes 25 bytes (200 points) 80 bytes (40 each) 80 bytes (20 each) 20 bytes 1860 bytes 512 programs 9999 labels Timer No.100 devices specified
Input/Output • I/O Link (I) Max. (master) (O) Max. • Built–in I/O (I) Max. (O) Max.	 1024 points maximum 1024 points maximum 96 points max. 64 points max.	 1024 points maximum 1024 points maximum 96 points max. 64 points max.
	Flash ROM 128KB	Flash ROM 128KB

NOTE

1 This is the number of ladder steps for the program only with basic instructions. The use of functional instructions may vary the number of ladder steps.

2 These have no limit of size for each. However, the total size of sequence program (the sum total of ladder, symbols/comments, messages, etc.) never exceed the storage size of sequence program. The size of them influences the capacity of others.

3 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.

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)

<table>
<tr><th colspan="3">Model</th><th>PMC–SA1</th><th>PMC–SA2</th><th>PMC–SA3/SA5</th><th>PMC–SB</th><th>PMC–SB2</th><th>PMC–SB3/SB4/SB5/SB6</th><th>PMC–SC</th><th>PMC–SC3/SC4</th></tr>
<tr><td rowspan="4">PMC address</td><td colspan="2">Interfaces between the PMC and CNC (F and G)</td><td colspan="8">Compatible (Note 8)</td></tr>
<tr><td colspan="2">Interfaces between the PMC and machine (X and Y)</td><td colspan="8">Compatible</td></tr>
<tr><td colspan="2">Subprogram, label (P and L)</td><td colspan="2">Not provided</td><td>Provided</td><td colspan="2">Not provided</td><td>Provided</td><td>Not provided</td><td>Provided</td></tr>
<tr><td colspan="2">Others (R, A, C, K, D, T)</td><td colspan="8">Compatible (Note 1)</td></tr>
<tr><td rowspan="2">Ladder program compatibility</td><td colspan="2">ROM format (object)</td><td colspan="8">Incompatible (Note 2)</td></tr>
<tr><td colspan="2">Source format (mnemonic)</td><td colspan="8">Compatible (Note 3)</td></tr>
<tr><td rowspan="2">System</td><td colspan="2">Divided system</td><td colspan="3">Not provided (Note 4)</td><td>Provided</td><td colspan="2">Not provided (Note 4)</td><td>Provided</td><td>Not provided</td></tr>
<tr><td colspan="2">Undivided system</td><td colspan="8">Provided</td></tr>
<tr><td>Structuring</td><td colspan="2">Sub program</td><td colspan="2">Unusable</td><td>Usable</td><td colspan="2">Unusable</td><td>Usable</td><td>Un-usable</td><td>Usable</td></tr>
<tr><td colspan="3">Basic commands</td><td colspan="8">Compatible</td></tr>
<tr><td rowspan="11">Function commands</td><td colspan="2">END3 (SUB48)</td><td colspan="6">Not provided</td><td colspan="2">Provided</td></tr>
<tr><td colspan="2">DISP (SUB49)</td><td colspan="3">Not provided (Note 5)</td><td colspan="5">Provided</td></tr>
<tr><td rowspan="2">COM (SUB9)</td><td>Coil count specification</td><td colspan="3">Not provided (Note 6)</td><td>Provided</td><td colspan="2">Not provided (Note 6)</td><td>Provided</td><td>Not provided (Note 6)</td></tr>
<tr><td>COME (SUB29) specification</td><td colspan="8">Provided</td></tr>
<tr><td rowspan="2">JMP (SUB10)</td><td>Coil count specification</td><td colspan="3">Not provided (Note 6)</td><td>Provided</td><td colspan="2">Not provided (Note 6)</td><td>Provided</td><td>Not provided (Note 6)</td></tr>
<tr><td>JMPE (SUB30) specification</td><td colspan="8">Provided</td></tr>
<tr><td colspan="2">FNC9X (SUB9X)</td><td colspan="6">Not provided</td><td colspan="2">Provided</td></tr>
<tr><td colspan="2">MMCWR (SUB98), MMCWW (SUB99)
MMC3R (SUB88), MMC3W (SUB89) (Note 7)</td><td colspan="3">Provided (Note 7)</td><td colspan="5">Provided</td></tr>
<tr><td colspan="2">MOVB (SUB43), MOVW (SUB44) ,
MOVN (SUB45)</td><td colspan="2">Not provided</td><td>Provided</td><td colspan="2">Not provided</td><td>Provided</td><td>Not provided</td><td>Provided</td></tr>
<tr><td colspan="2">DIFU (SUB57), DIFD (SUB58)</td><td colspan="2">Not provided</td><td>Provided</td><td colspan="2">Not provided</td><td>Provided</td><td>Not provided</td><td>Provided</td></tr>
<tr><td colspan="2">AND (SUB60), OR (SUB61)
NOT (SUB62), EOR (SUB59)</td><td colspan="2">Not provided</td><td>Provided</td><td colspan="2">Not provided</td><td>Provided</td><td>Not provided</td><td>Provided</td></tr>
<tr><td rowspan="2">Function command (for structured programming)</td><td colspan="2">Commands for subprogram
END (SUB64) ,
CALL (SUB65), CALLU (SUB66) ,
SP (SUB71), SPE (SUB72)</td><td colspan="2">Not provided</td><td>Provided</td><td colspan="2">Not provided</td><td>Provided</td><td>Not provided</td><td>Provided</td></tr>
<tr><td colspan="2">Extended jump command
JMPB (SUB68), JMPC (SUB73)
LBL (SUB69)</td><td colspan="2">Not provided</td><td>Provided</td><td colspan="2">Not provided</td><td>Provided</td><td>Not provided</td><td>Provided</td></tr>
</table>

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.

8 In the PMC–SB4, SB6, and SC4, interface extension is made. The extended portion of the interface is not compatible with other PMCs.

Table 1.2 Summary of specification of ladder program (3)

<table>
<tr><th colspan="3">Model
Series</th><th>PMC–NA (4046)</th><th>PMC–NB (4047) (4048)</th><th>PMC–NB2 (4048)</th></tr>
<tr><td rowspan="4">PMC address</td><td colspan="2">Interfaces between the PMC and CNC (F and G)</td><td colspan="3">Incompatible</td></tr>
<tr><td colspan="2">Interfaces between the PMC and machine (X and Y)</td><td colspan="3">Compatible</td></tr>
<tr><td colspan="2">Subprogram, label (P and L)</td><td>Not provided</td><td colspan="2">Provided</td></tr>
<tr><td colspan="2">Others (R, A, C, K, D, T)</td><td colspan="3">Compatible (Note 1)</td></tr>
<tr><td rowspan="2">Ladder program compatibility</td><td colspan="2">ROM format (object)</td><td colspan="3">Incompatible (Note 2)</td></tr>
<tr><td colspan="2">Source format (mnemonic)</td><td colspan="3">Compatible (Note 3)</td></tr>
<tr><td rowspan="2">System</td><td colspan="2">Divided system</td><td>Provided</td><td colspan="2">Not provided</td></tr>
<tr><td colspan="2">Undivided system</td><td>Not provided</td><td colspan="2">Provided</td></tr>
<tr><td rowspan="2">Structuring</td><td colspan="2">Subprogram</td><td>Usable</td><td colspan="2">Unusable</td></tr>
<tr><td colspan="2">Step sequence</td><td colspan="2">Unusable</td><td>Usable</td></tr>
<tr><td colspan="3">Basic commands</td><td colspan="3">Compatible</td></tr>
<tr><td rowspan="9">Function commands</td><td colspan="2">END3 (SUB48)</td><td colspan="3">Provided</td></tr>
<tr><td colspan="2">DISP (SUB49)</td><td>Provided</td><td colspan="2">Not provided</td></tr>
<tr><td rowspan="2">COM (SUB9)</td><td>Coil count spesification</td><td>Provided</td><td colspan="2">Not provided</td></tr>
<tr><td>COME (SUB29) specification</td><td colspan="3">Provided</td></tr>
<tr><td rowspan="2">JMP (SUB10)</td><td>Coil count specification</td><td>Provided</td><td colspan="2">Not provided</td></tr>
<tr><td>JMPE (SUB30) specification</td><td colspan="3">Provided</td></tr>
<tr><td colspan="2">FNC9X (SUB9X)</td><td colspan="3">Provided</td></tr>
<tr><td colspan="2">LIBRY (SUB60), LEND (SUB61)</td><td>Provided</td><td colspan="2">Not provided</td></tr>
<tr><td colspan="2">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)</td><td>Not provided</td><td colspan="2">Provided</td></tr>
<tr><td rowspan="2">Function command (for structured programming)</td><td colspan="2">• Command for subprogram
END (SUB64) , CALL (SUB65) , CALLU (SUB66) ,
SP (SUB71) , SPE (SUB72)</td><td>Not provided</td><td colspan="2">Provided</td></tr>
<tr><td colspan="2">• Extended jump command
JMPB (SUB68) , JMPC (SUB73) , LBL (SUB69)</td><td>Not provided</td><td colspan="2">Provided</td></tr>
</table>

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 PMC–NB	FANUC Series 15–B PMC–NB2	FANUC Series 15*i* 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.

Table 1.2 Ladder Compatibility (5)

Model		16*i*/18*i*/21*i*–A			16*i*/18*i*/21*i*–B
		SA5	SB5	SB6	SB7
PMC address	PMC–CNC interface (F, G)	F0 to F255 G0 to G255		Expanded to F0 to F511 and G0 to G511	Expanded to F0 to F767 and G0 to G767
	PMC–machine interface (X, Y)	X0 to X127 Y0 to Y127		X200 to X327 and Y200 to Y327 are added.	Compatible
	Area used by management software (K)	K17 to K19		Changed to K900 to K909	Expanded to K900 to K919
Basic instruction		Compatible			
Function instruction	END3 CTRB MOVD	No			Yes (additional)
	DISP	Yes			No

The above table lists the differences to be noted on upward conversion to the PMC–SB5, PMC–SB6, or PMC–SB7. Simple conversion in the reverse direction cannot be performed generally because functions such as the PMC address ranges are limited.

CAUTION

1 The above table does not contain simple addition of reserved areas for PMC addresses that are not used by the ladder.

2 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

A ladder which can be operated on the 16*i*/18*i*/21*i*–A must also be checked for operation on the 16*i*/18*i*/21*i*–B system.

3 Changes in memory capacities required for a sequence program (PMC–SB7)
As described in Section 2.8, the memory capacities required for the system and symbol/comment data have been changed. As a result, the capacity of flash ROM is increased as compared with the PMC–SA5/SB5/SB6, even for the same source program. If a created sequence program exceeds the capacity of flash ROM, add the step count option or delete unnecessary symbols and comments.

4 Screen display and operation
The PMC–SB7 has much the same screen display/operation system as the PMC–SA5/SB5/SB6. Some operations for the PMC–SA5/SB5/SB6 have been modified to improve operability and functionality, however. The same goes for the PMC–SA1. For details, see Chapter 1, "SCREEN OPERATION FOR PMC–SA1/SB7" in Part V.

5 PMC parameter input/output (PMC–SB7)
With increase in size of a PMC parameter (T, C, K, and D areas), the data format used for inputting PMC parameters from a memory card or floppy disk (FANUC Handy File) or outputting them to it on the PMC I/O screen has been extended.
- PMC parameters output by the PMC–SA5/SB5/SB6 can be read by the PMC–SB7.
- PMC parameters output by the PMC–SB7 cannot be read by the PMC–SA5/SB5/SB6.

An extension relay (E area) has been added as a PMC address. A PMC parameter input or output by the PMC–SB7 contains the E area. If a PMC parameter output by the PMC–SB7 is read, the E area is initialized to the status when the PMC parameter is output.

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.

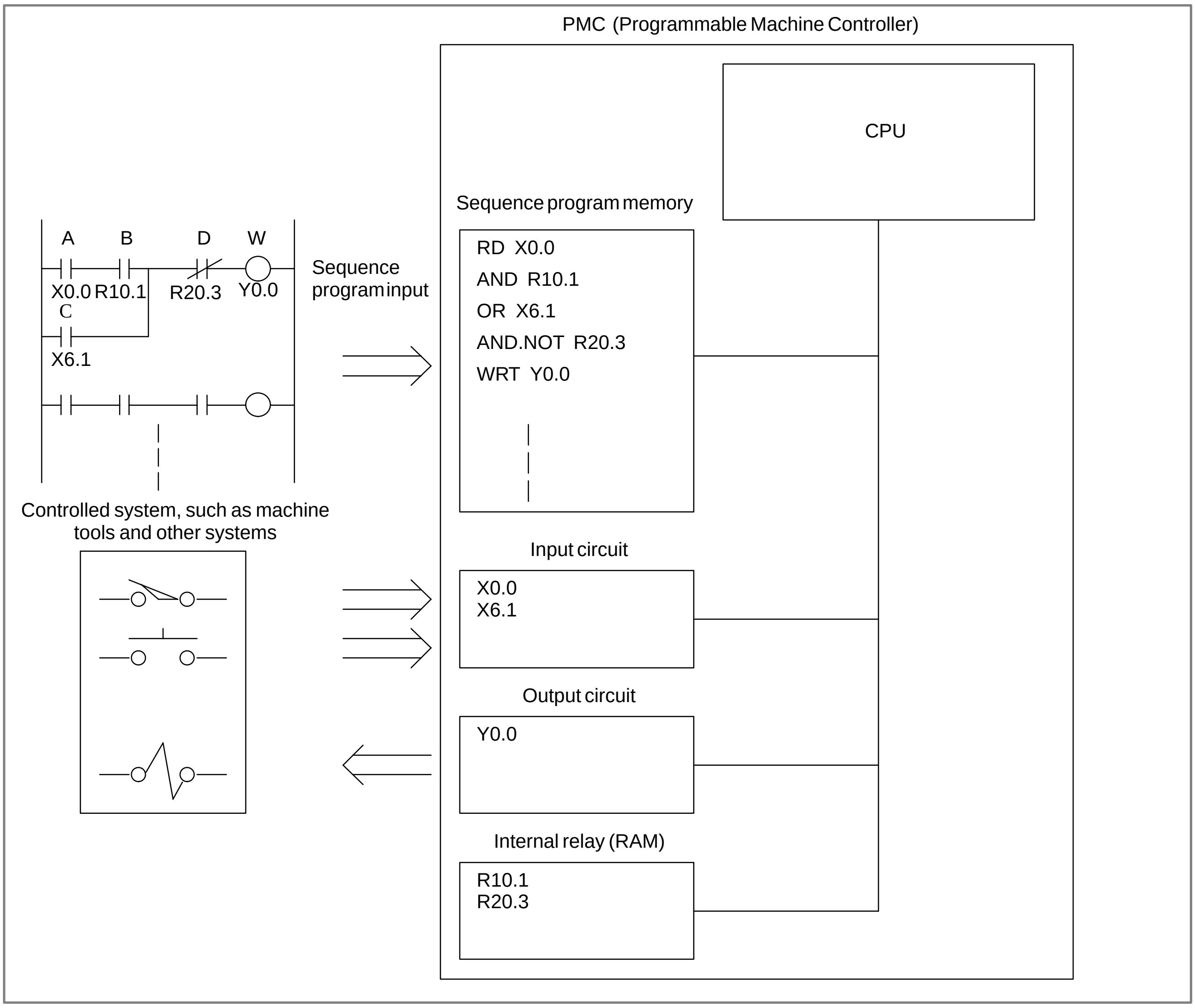


Fig. 1.3 Execution of sequence program by PMC

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 CONNECTION 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.

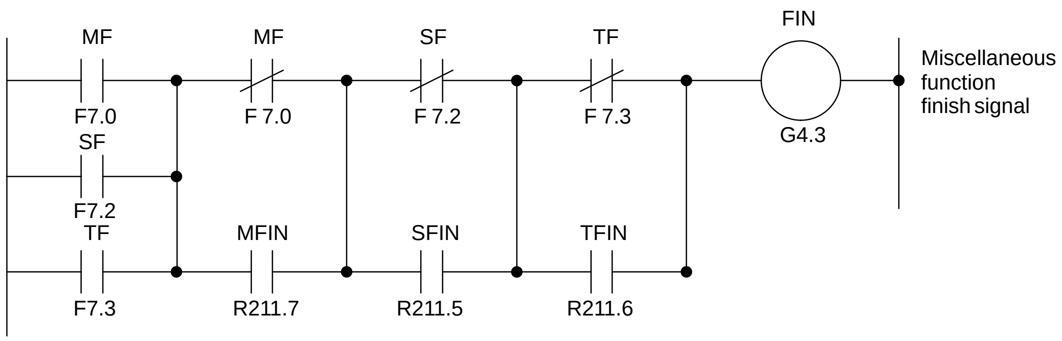
Signal names (max. six characters) can be entered to the input/output signals, comments (max. 30 characters) can be entered to the relay coil, and comments (max. 30 characters) can be entered to the input/output signals of the address tables at the time of entry of the sequence program. Be sure to enter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

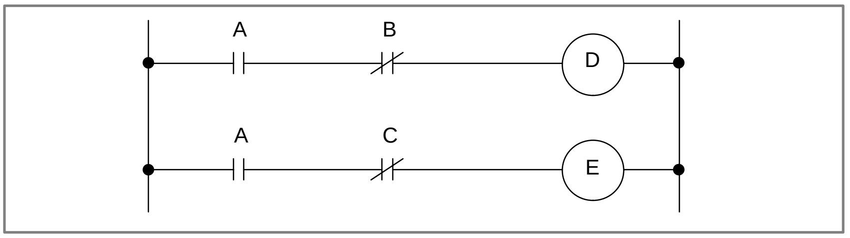


Fig. 2.1 (a) Circuit examples

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.

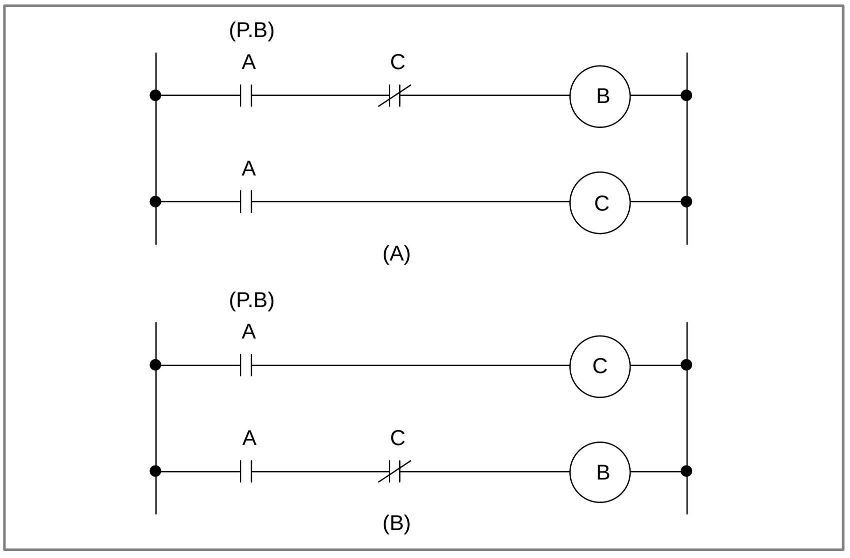


Fig. 2.1 (b) Circuit examples

(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)).

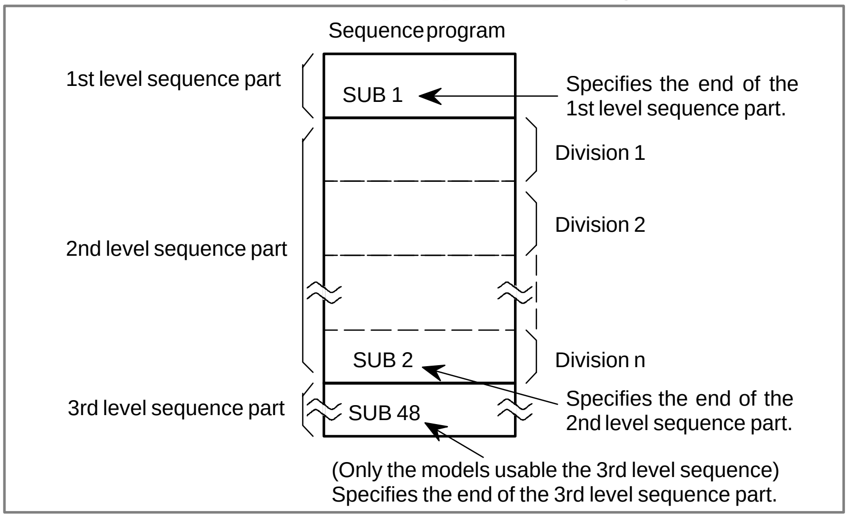


Fig. 2.3 (a) Construction of sequence program

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.

(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.

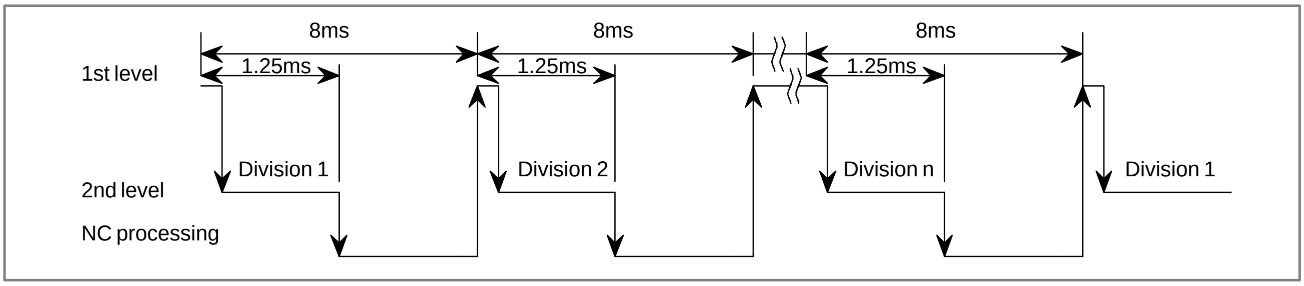


Fig. 2.3 (b) Sequence in which the Sequence Program Is Executed (PMC–SA1, –SA2, –SB and –SB2)

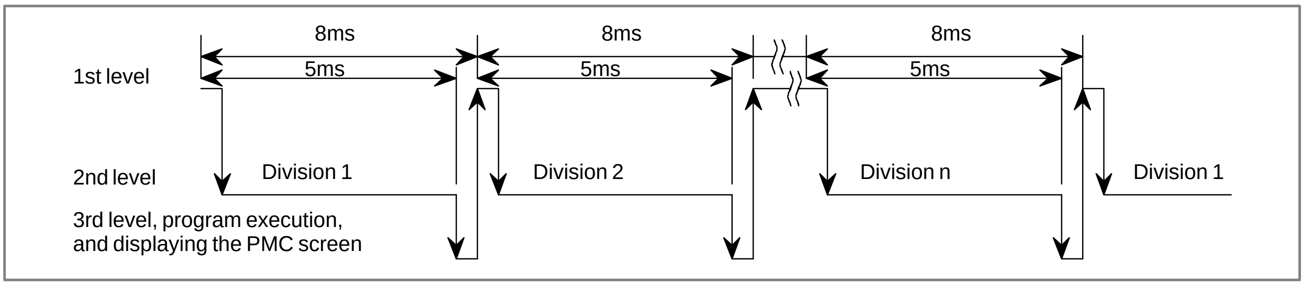


Fig. 2.3 (c) Sequence in which the Sequence Program Is Executed (PMC–SC)

(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.

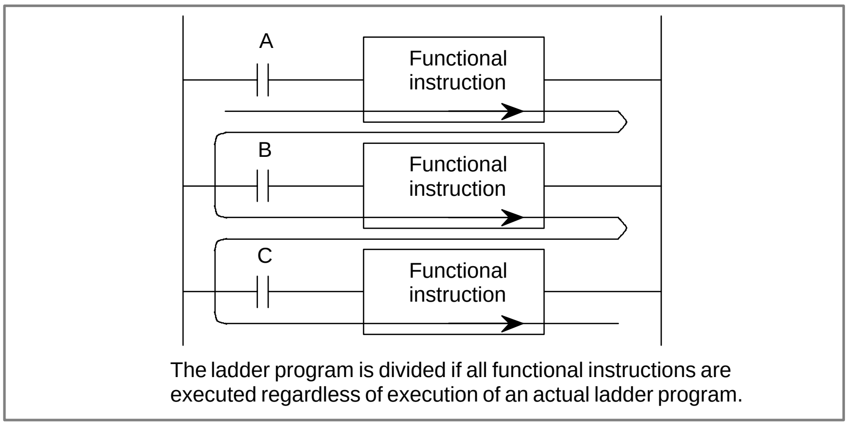


Fig. 2.3 (d) Divisions in the divided system

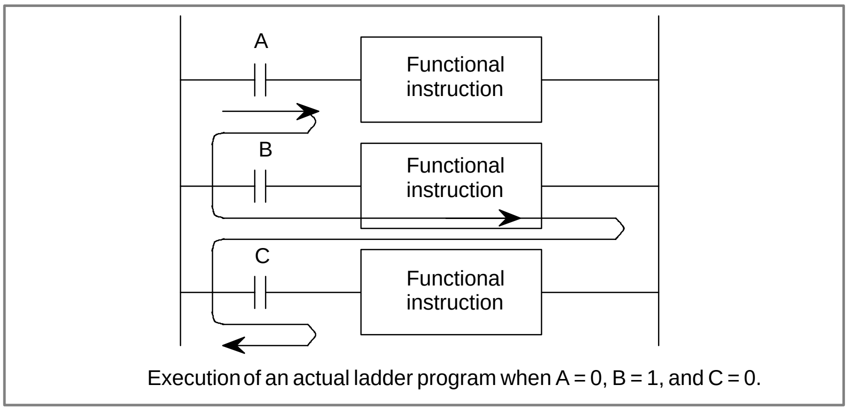


Fig. 2.3 (e) Execution of a ladder program

(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.

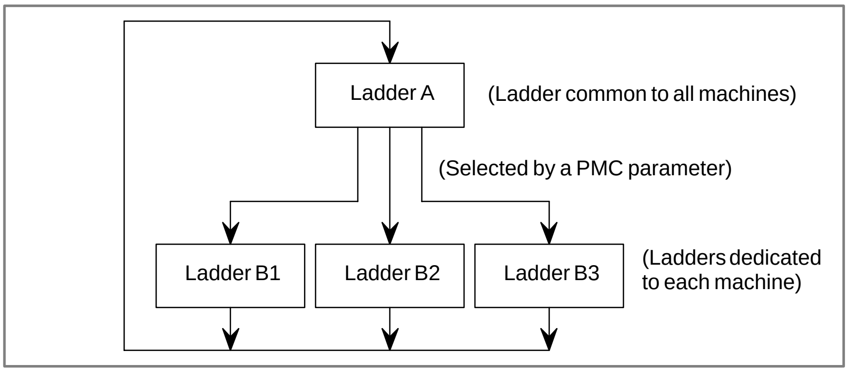


Fig. 2.3 (f)

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

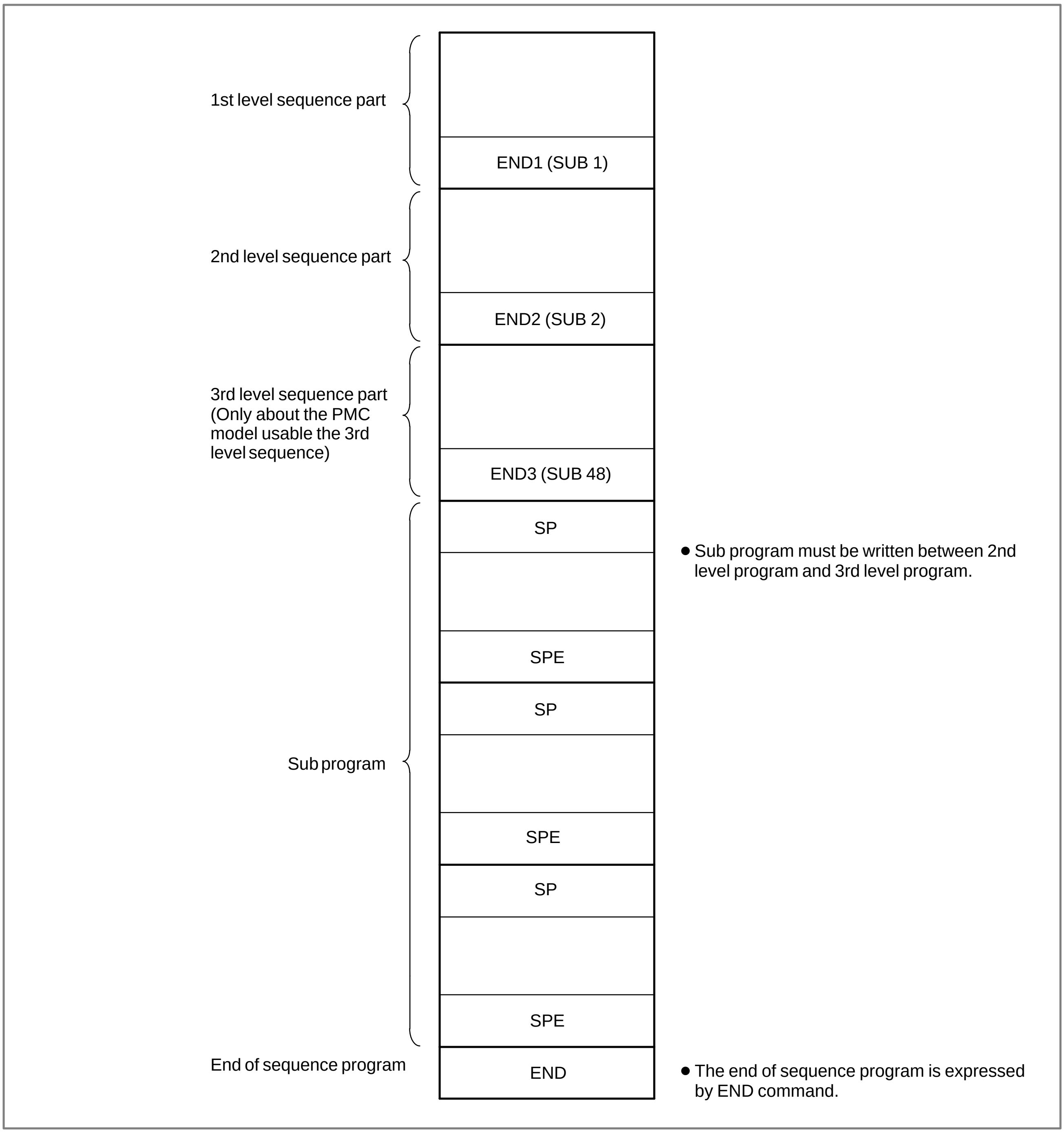


Fig. 2.3 (g)

2.4 SEQUENCE PROGRAM STRUCTURING

○ : Can be used
× : Cannot be used

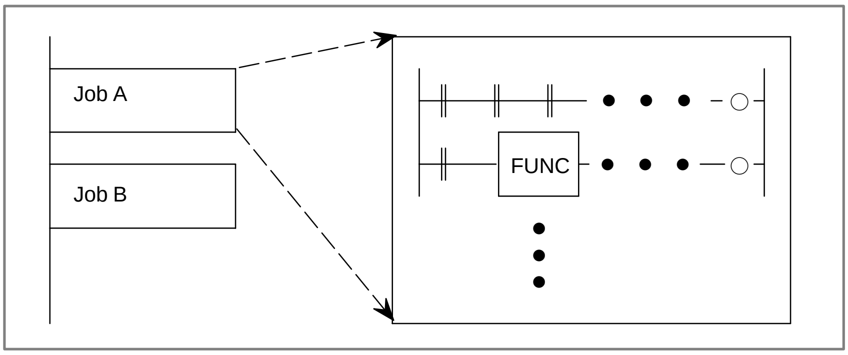
PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

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

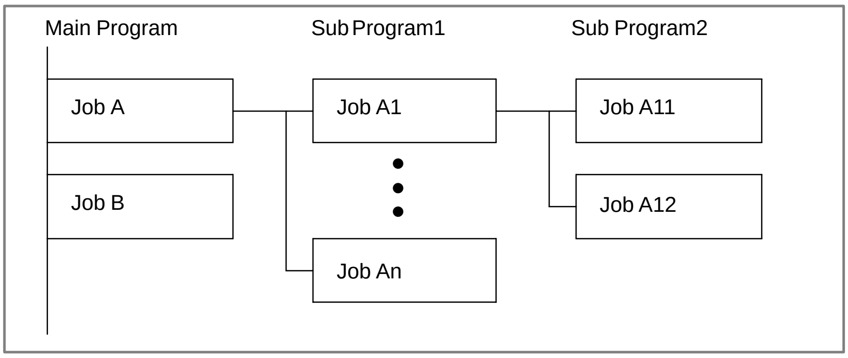
- A program can be understood and developed easily.
- A program error can be found easily.
- When an operation error occurs, the cause can be found easily.

Three major structured programming capabilities are supported.

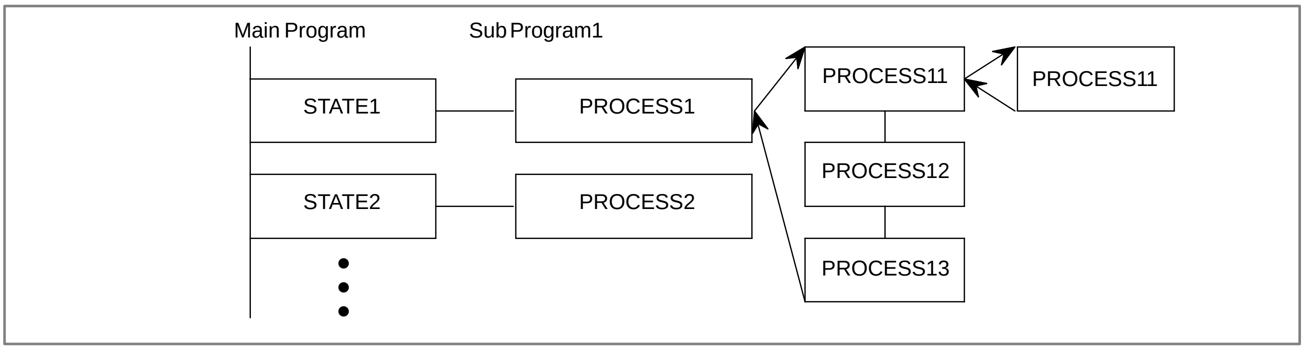
(1) Subprogramming
A subprogram can consist of a ladder sequence as the processing unit.



(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.

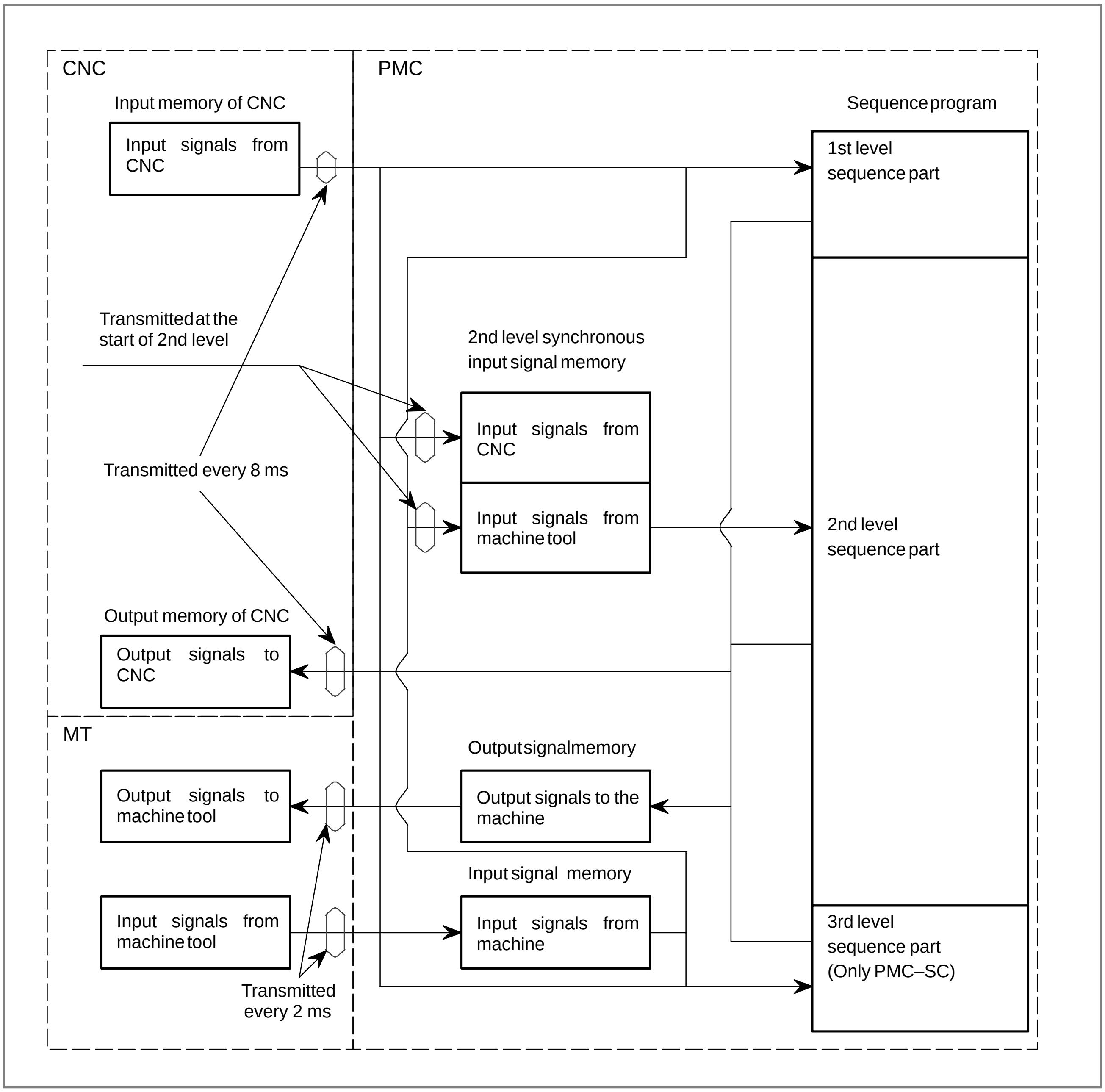


Fig. 2.5 PMC I/O signals

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.

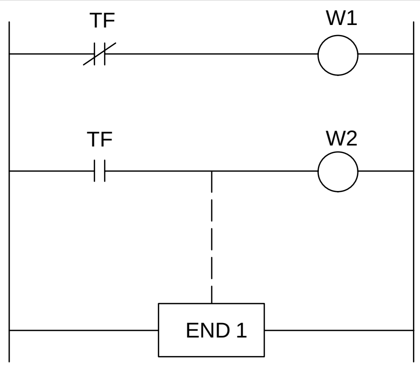
(2) Output signals to machine tool (DI/DO card)
Output signals to the machine tool are transferred from the PMC output signal memory to the machine tool.

(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.

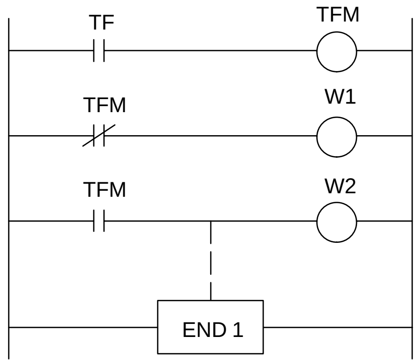
2.5.3 I/O Signals to CNC

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).

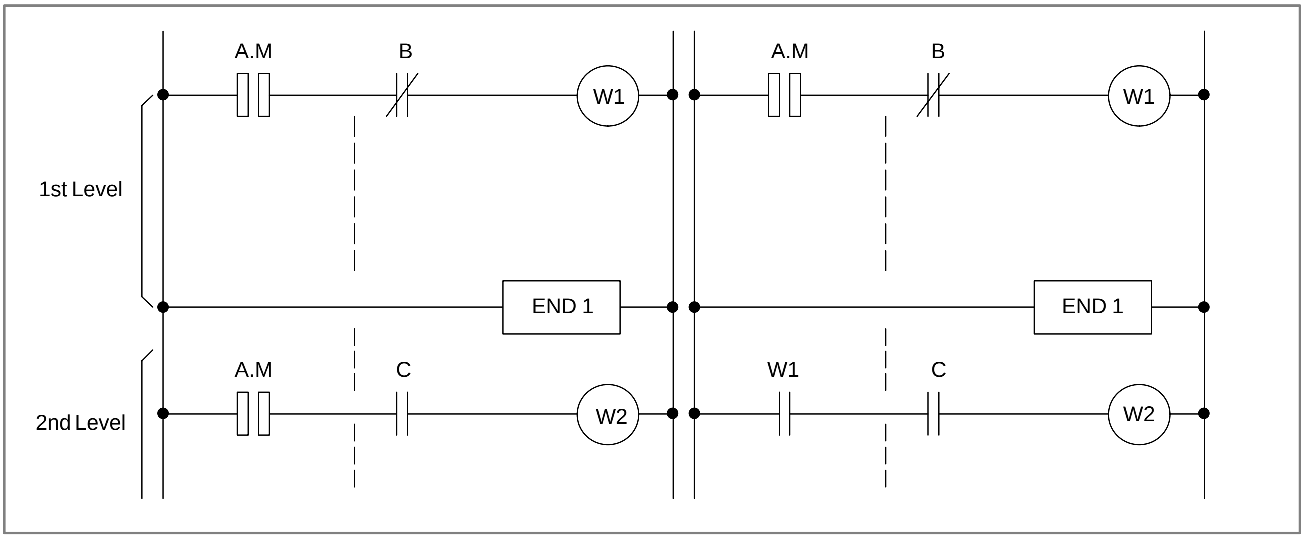


Fig. 2.5.4 (a) Fig. 2.5.4 (b)

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

> **NOTE**
> For the PMC–SB/SC, see the execution time constant of each function instruction in Table 5 (b) in Section I–5, ”PMC FUNCTION INSTRUCTIONS.”

2.8 SEQUENCE PROGRAM MEMORY CAPACITY

The following tables list memory capacities required for a sequence program. Create a sequence program so that the total capacity of these items does not exceed the sequence program memory capacity.

Table 2.8 (a) PMC–SB7

Type	Item	Memory capacity (Note 1)
Ladder (Note 2)	Basic instruction	4 bytes
	Functional instruction	4 bytes
	Functional instruction parameter	4 bytes
Symbol/comment (Note 2)	One symbol or comment	24 bytes
	One half–width character in a comment	1 byte (Note 3)
Message (Note 2)	One half–width alphanumeric character in a message	1 byte (Note 4)
Others	System used area	About 15K bytes

Table 2.8 (b) PMC–SB4/SB6/SC4/NB2/NB6

Type	Item	Memory capacity (Note 1)
Ladder (Note 2)	Basic instruction	4 bytes
	Functional instruction	4 bytes
	Functional instruction parameter	4 bytes
Symbol/comment (Note 2)	One symbol or comment	12 bytes
	One half–width character in a comment	1 byte
Message (Note 2)	One half–width alphanumeric character in a message	1 byte (Note 4)
Others	System used area	About 4K bytes

Table 2.8 (c) PMC–SB6 (I/O links expanded)

Type	Item	Memory capacity (Note 1)
Ladder (Note 2)	Basic instruction	4 bytes
	Functional instruction	4 bytes
	Functional instruction parameter	4 bytes
Symbol/comment (Note 2)	One symbol or comment	12 bytes
	One half–width character in a comment	1 byte
Message (Note 2)	One half–width alphanumeric character in a message	1 byte (Note 4)
Others	System used area	About 9.5K bytes

Table 2.8 (d) PMC–SA1/SA3/SA5/SB/SB2/SB3/SB5/SC/SC3/PA1/PA3/NB

Type	Item	Memory capacity (Note 1)
Ladder (Note 2)	Basic instruction	4 bytes
	Functional instruction	4 bytes
	Functional instruction parameter	4 bytes
Symbol/comment (Note 2)	One symbol or comment	10 bytes
	One half–width character in a comment	1 byte
Message (Note 2)	One half–width alphanumeric character in a message	1 byte (Note 4)
Others	System used area	About 2K bytes

NOTE

1 The total capacity of a sequence program (including all items such as ladder, symbols/comments, and messages) cannot exceed the capacity of the sequence program storage memory. If the ladder, symbol/comment, or message area is large, the size of another area may be limited.

2 The PMC programmer may adjust arrangement of the areas in the sequence program memory to improve processing efficiency. As a result, up to 1K (1024) bytes may be added to the total capacity of each type of data.

3 A full–width character requires double the capacity.

4 For each of half–width katakana characters and special characters, and full–width hiragana characters, kanji characters, and special characters, a capacity of 1 byte is required per digit of the notation (including characters preceding and following the character such as @) by character code input. For details of notation by character code input, refer to the paragraph describing DISPB in Chapter 5, “FUNCTIONAL INSTRUCTIONS.”

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.

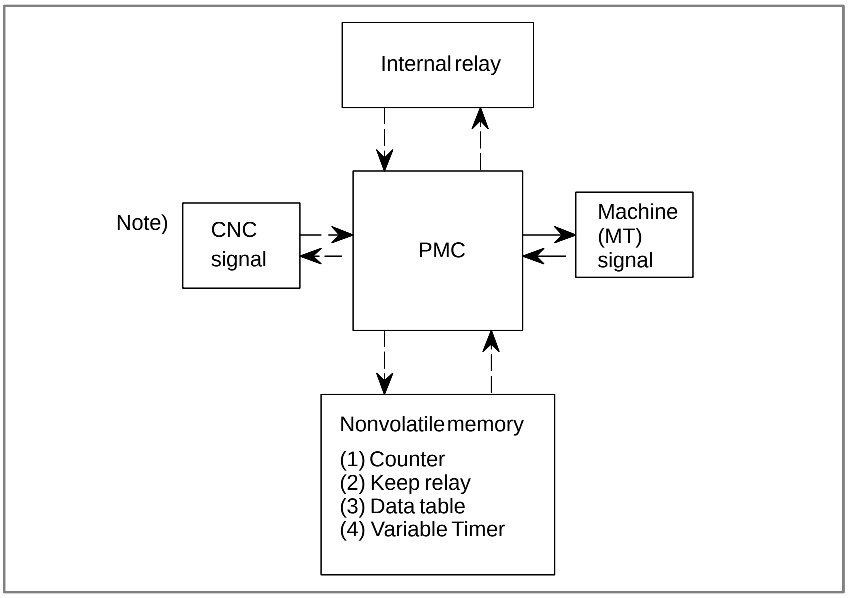


Fig. 3 Addresses related to PMC

(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)

Sym-bol	Type of signal	Model FANUC Series 16*i*/160*i*/18*i*/180*i*–A PMC–SB5	Model FANUC Series 16*i*/160*i*/18*i*/180*i*–A PMC–SB6
X	Signal from the machine to PMC (MT → PMC)	X0 to X127 (Note 1)	X0 to X127 X200 to X327 (Note 2) (Note 1)
Y	Signal from the PMC to machine (PMC → MT)	Y0 to Y127 (Note 1)	Y0 to Y127 Y200 to Y327 (Note 2) (Note 1)
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

1 The addresses (X1000 and up, Y1000 and up) cannot be assigned for I/O. Never use X1000 and up, or Y1000 and up.

2 I/O of the I/O Link channel 2 can be assigned to the addresses (X200 to X327, Y200 to Y327). You can use the I/O Link channel 2 only when CNC hardware support the I/O Link 2–channel and optional I/O Link expansion is provided with CNC. And to assign to these areas, it is necessary to use programming software that supports the I/O Link expansion.

Table 3 Alphabetic characters in address numbers (8)

<table>
<tr><th rowspan="3">Sym-bol</th><th rowspan="3">Type of signal</th><th colspan="3">Model</th></tr>
<tr><th colspan="3">FANUC Series 21i/210i–A</th></tr>
<tr><th>PMC–SA1</th><th>PMC–SA5</th><th>PMC–SB6</th></tr>
<tr><td>X</td><td>Signal from the machine to PMC
(MT → PMC)</td><td colspan="2">X0 to X127 (Note 1)</td><td>X0 to X127 (Note 1)
X200 to X327 (Note 2)</td></tr>
<tr><td>Y</td><td>Signal from the PMC to machine
(PMC → MT)</td><td colspan="2">Y0 to Y127 (Note 1)</td><td>Y0 to Y127 (Note 1)
Y200 to Y327 (Note 2)</td></tr>
<tr><td>F</td><td>Signal from the NC to PMC
(NC → PMC)</td><td colspan="2">F0 to F255
F1000 to F1255</td><td>F0 to F511
F1000 to F1511
F2000 to F2511</td></tr>
<tr><td>G</td><td>Signal from the PMC to NC
(PMC → NC)</td><td colspan="2">G0 to G255
G1000 to G1255</td><td>G0 to G511
G1000 to G1511
G2000 to G2511</td></tr>
<tr><td>R</td><td>Internal relay</td><td>R0 to R999
R9000 to R9099</td><td>R0 to R999
R9000 to R9117</td><td>R0 to R2999
R9000 to R9199</td></tr>
<tr><td>A</td><td>Message request signal</td><td colspan="2">A0 to A24</td><td>A0 to A124</td></tr>
<tr><td>C</td><td>Counter</td><td colspan="2">C0 to C79</td><td>C0 to C199</td></tr>
<tr><td>K</td><td>Keep relay</td><td colspan="2">K0 to K19</td><td>K0 to K39
K900 to K909</td></tr>
<tr><td>D</td><td>Variable timer</td><td colspan="2">T0 to T79</td><td>T0 to T299</td></tr>
<tr><td>T</td><td>Data table</td><td colspan="2">D0 to D1859</td><td>D0 to D7999</td></tr>
<tr><td>L</td><td>Label number</td><td>–</td><td>L1 to L9999</td><td>L1 to L9999</td></tr>
<tr><td>P</td><td>Subprogram number</td><td>–</td><td>P1 to P512</td><td>P1 to P2000</td></tr>
</table>

NOTE

1 The addresses (X1000 and up, Y1000 and up) cannot be assigned for I/O. Never use X1000 and up, or Y1000 and up.

2 I/O of the I/O Link channel 2 can be assigned to the addresses (X200 to X327, Y200 to Y327). You can use the I/O Link channel 2 only when CNC hardware supports the I/O Link 2–channel and optional I/O Link expansion is provided with CNC. And to assign to these areas, it is necessary to use programming software that supports the I/O Link expansion.

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

Table 3 Alphabetic Characters for PMC Address Number (12)

Address	Type of signal	Control type	
		Series 16*i*/18*i*/21*i*–B	
		PMC–SA1	PMC–SB7
X	Input signal from machine tool to PMC (MT→PMC)	X0 to X127	X0 to X127 X200 to X327*1 X1000 to X1127*2
Y	Output signal from PMC to machine tool (PM→CMT)	Y0 to Y127	Y0 to Y127 Y200 to Y327*1 Y1000 to Y1127*2
F	Input signal from NC to PMC (NC→PMC)	F0 to F255	F0 to F767*3 F1000 to F1767*4 F2000 to F2767*4 F3000 to F3767*5
G	Output signal from PMC to NC (PM→CNC)	G0 to G255	G0 to G767*3 G1000 to G1767*4 G2000 to G2767*4 G3000 to G3767*5
R	Internal relay	R0 to R999 R0 to R9099	R0 to R7999 R9000 to R9499*6
E	Extra relay*7	–	E0 to E7999
A	Message request	A0 to A24	A0 to A249
	Message display state*8	–	A9000 to A9249
C	Counter	C0 to C79	C0 to C399 C5000 to C5199*9
K	Keep relay	K0 to K19	K0 to K99 K900 to K919*10
T	Variable timer	T0 to T79	T0 to T499 T9000 to T9499*11
D	Date table	D0 to D1859	D0 to D9999
L	Label number	–	L1 to L9999
P	Subprogram number	–	P1 to P2000

CAUTION

1 These addresses are used for channel 2 of I/O Link. I/O link expansion option is necessary.
2 This area is reserved for PMC. I/O can not be assigned in it.
Don't use it in sequence program.
3 This area contains PMC reserve. Actual available addresses depend on the configuration of CNC system.
4 This area is used for multi path system. It contains PMC reserve. Actual available addresses depend on the configuration of CNC system.
5 This area is for PMC reserve. Don't use it in sequence program.
6 This area is used for PMC system software as special relay. Please use these according to the explanation of each address.
7 This area can be used equally as internal relay(R).
These relays (E) are non–volatile type. However, these can be input/output to/from memory card, etc, as PMC parameter.
8 These addresses are message display state signals that have one–to–one correspondence to message display request signal. It is impossible to write into these addresses.
9 This area is used for counter instruction (CTRB) that requires preset value as fixed number.
10 This area is used for PMC control software. Please use these signals according to each explanation.
11 This area is for PMC reserve. Don't use it in sequence program.

Table 3 Alphabetic Characters for PMC Address Number (12)

Character	Signal description	FANUC Series 0*i*	
		PMC–SA1	PMC–SA3
X	Signal from machine to PMC (MTP→MC)	X0 to X127 X1000 to X1011	
Y	Signal from PMC to machine (PMCMT)	Y0 to Y127 Y1000 to Y1008	
F	Signal from NC to PMC (NCPMC)	F0 to F255 F1000 to F1255	
G	Signal from PMC to NC (PMCNC)	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	
T	Variable timer	T0 to T79	
D	Data table	D0 to D1859	
L	Label number	–	L1 to L9999
P	Subprogram number	–	P1 to P512

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 of channel 1 are from X0 to X127.
Addresses of channel 2 are from X200 to X327. (PMC–SB6/SB7 for Series 16*i* /160*i* /18*i* /180*i* /21*i* /210*i*)

(b) PMC → MT
Addresses of channel 1 are from Y0 to Y127.
Addresses of channel 2 are from Y200 to Y327. (PMC–SB6/SB7 for Series 16*i* /160*i* /18*i* /180*i* /21*i* /210*i*)

Up to 1024 input and 1024 output points can be assigned to any address of each channel 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.

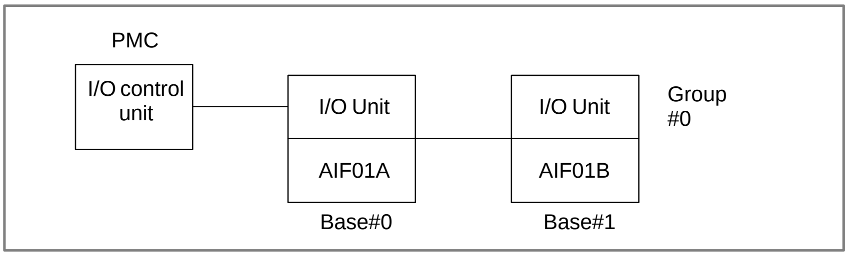


Fig. 3.2.2 (a)

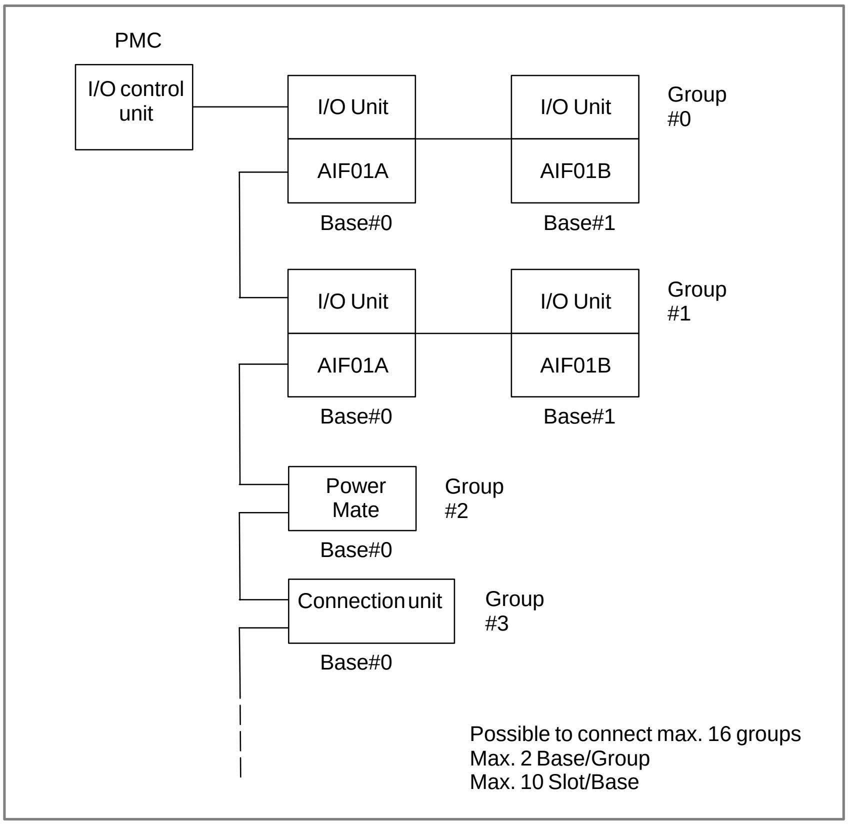


Fig. 3.2.2 (b)

(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

Input format	Module name (Actual module name)	Rated voltage	Rated current	Polarity	Response time	Number of input points	Terminal	Indication by LED
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
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
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

Output format	Module name (Actual module name)	Rated voltage	Rated current	Polarity	Number of points	Number of points/ common	Terminal	Indication by LED	Fuse
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
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
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

Name	Module name (actual module name)	Occupied address	Specifications
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	
Analog input module	AD04A (AAD04A)	Input: 8 bytes	
Analog output module	DA02A (ADA02A)	Output: 4 bytes	
Connection unit (one piece)	CN01I	Input: 12 bytes	
Connection unit (two pieces)	CN01O	Output: 8 bytes	
Connection unit (one piece)	CN02I	Input: 24 bytes	
Connection unit (two pieces)	CN02O	Output: 16 bytes	
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	
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	
Machine operator's panel interface unit	OC02I	Input: 16 bytes	
	OC02O	Output: 16 bytes	
	OC03I	Input: 32 bytes	
	OC03O	Output: 32 bytes	
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	
	OC02O	Output: 16 bytes	
	OC03I	Input: 32 bytes	
	OC03O	Input: 32 bytes	
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.
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	

Name	Module name (actual module name)	Occupied address	Specifications
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.
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	
External I/O card A, D for Power Mate	/ 6	Input 6 bytes	
	/ 4	Output 4 bytes	
External I/O card B, E for Power Mate	OC01I	Input 12 bytes	
	OC01O	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 On the PMC–SB6/SB7 for Series 16*i*/18*i*/21*i*/160*i*/180*i*/210*i*, when programming software supports the I/O Link expansion, up to 16 groups of I/O Module can be assigned to the I/O Link channel 2 area. To link the I/O Link channel 2 actually, the option of the I/O Link expansion is necessary.

CAUTION

For I/O Unit MODEL A, to use 3, 5, 6, or 7 bytes for assignment, change the module name as follows.
Do not use 10241, /3, /5, /6, or /7 for the module name.
Module name

Before change	→	After change
10241	→	/4
/3	→	/4
/5	→	/8
/6	→	/8
/7	→	/8

(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 (c), 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
X006	0	0	5	ID32A
X007	0	0	5	ID32A
X008	0	0	5	ID32A
X009				

Automatical set

Fig. 3.2.2 (c) 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.

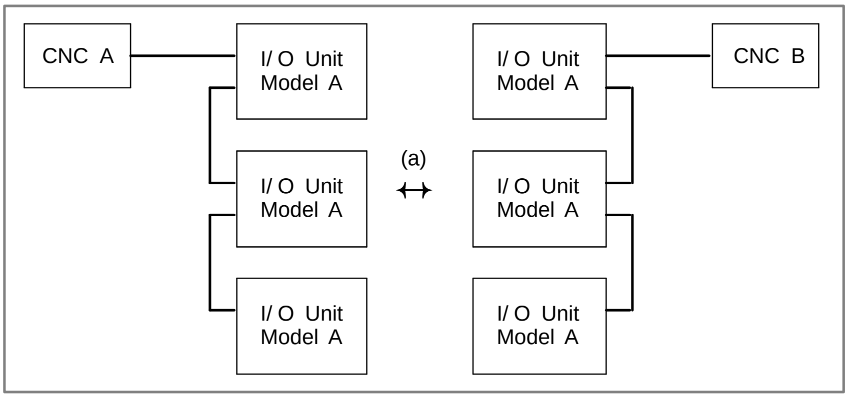
ADDRESS	GROUP	BASE	SLOT	NAME
X000	0	0	1	ID16C
X001	0	0	1	ID16C
X002	0	0	2	ID16D
X003	0	0	2	ID16D
X004	1	0	1	IA16G
X005	1	0	1	IA16G
X006	1	0	2	IA16G
X007	1	0	2	IA16G
X008	2	0	1	ID16D
X009	2	a 0	1	ID16D

Fig. 3.2.2 (d)

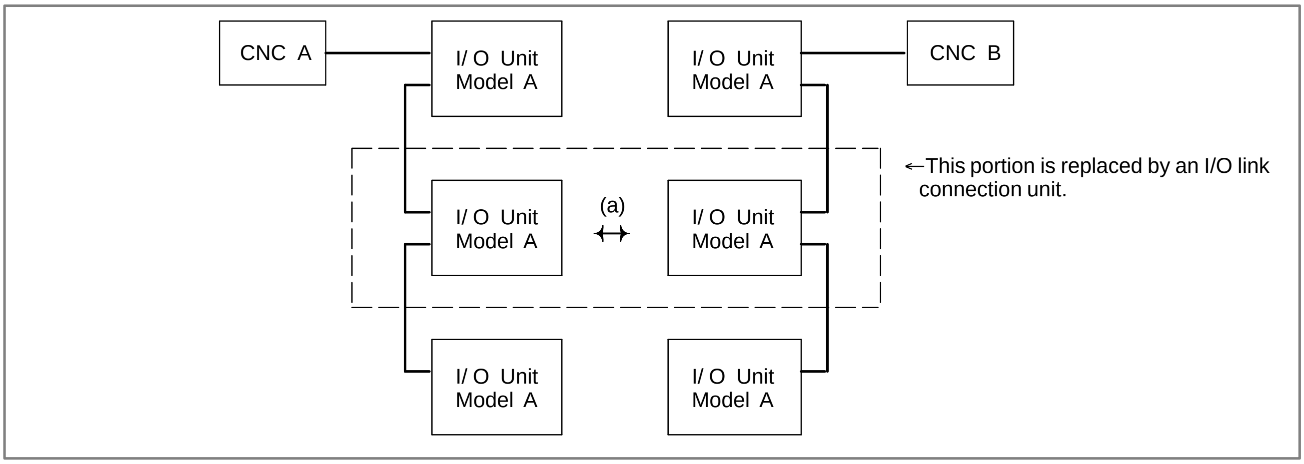
3.2.3 I/O Link Connection Unit Assignment

Concept:

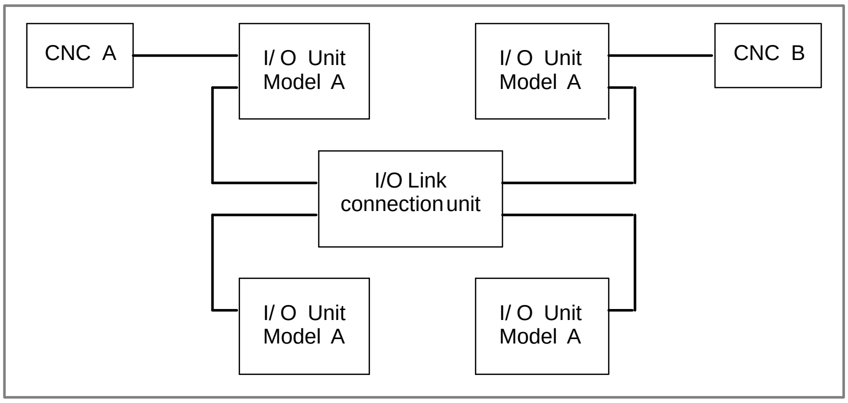
In conventional data transfer, when data is to be transferred between CNC A and CNC B, the I/O units indicated by (a) (figure below) must be connected with each other. (In this case, data can be transferred using any I/O unit.)



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:

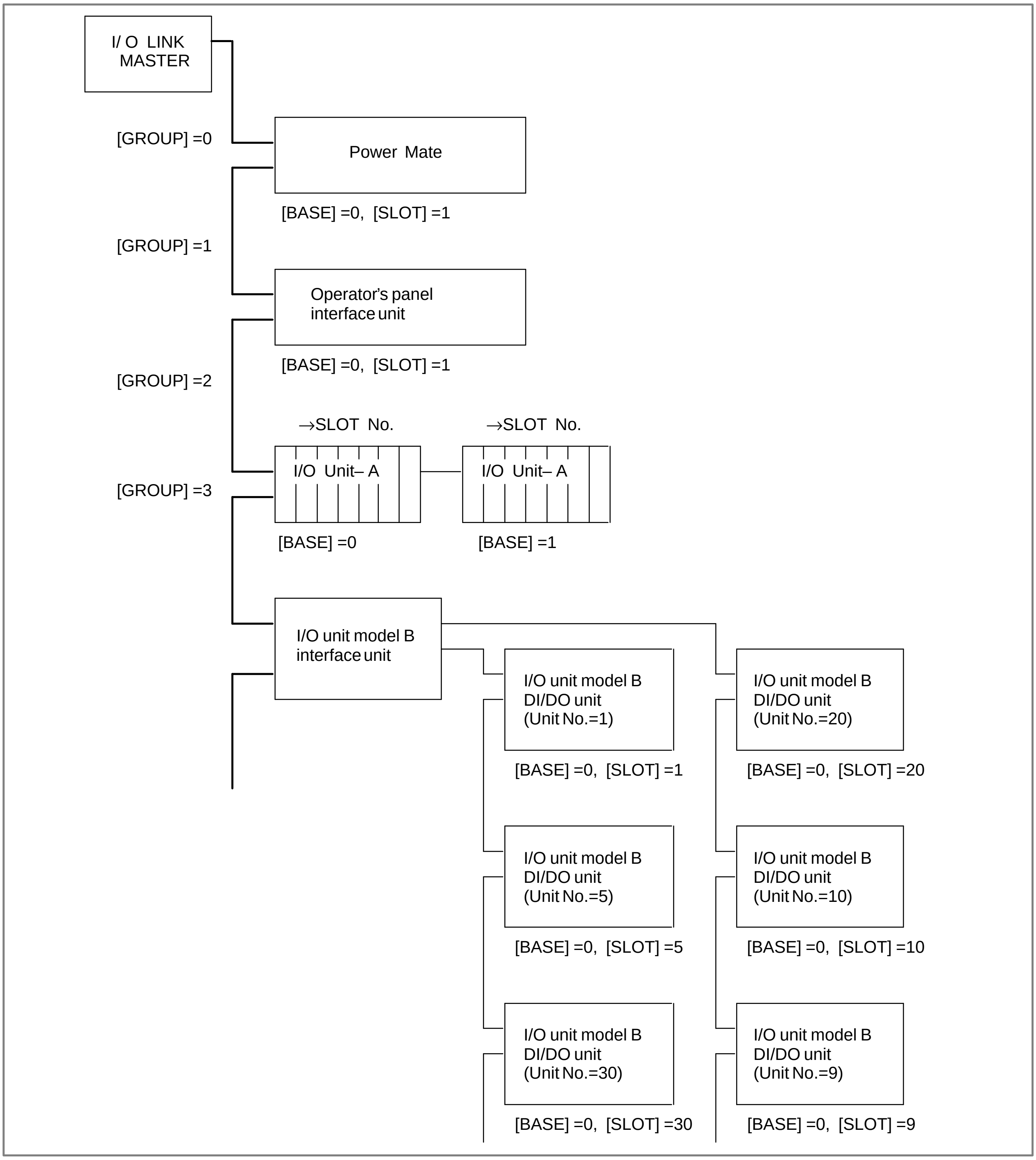
When a connection unit that occupies 16–byte addresses is attached to the input side in GROUP = 1, enter "1.0.1.OC02I."

> **NOTE**
> On the PMC–SB6/SB7 for Series 16*i*/18*i*/21*i*/160*i*/180*i*/210*i*, when programming software supports the I/O Link expansion, I/O Link connection unit can be assigned to the I/O Link channel 2 area. To link the I/O Link channel 2 actually, the option of the I/O Link expansion is necessary.

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

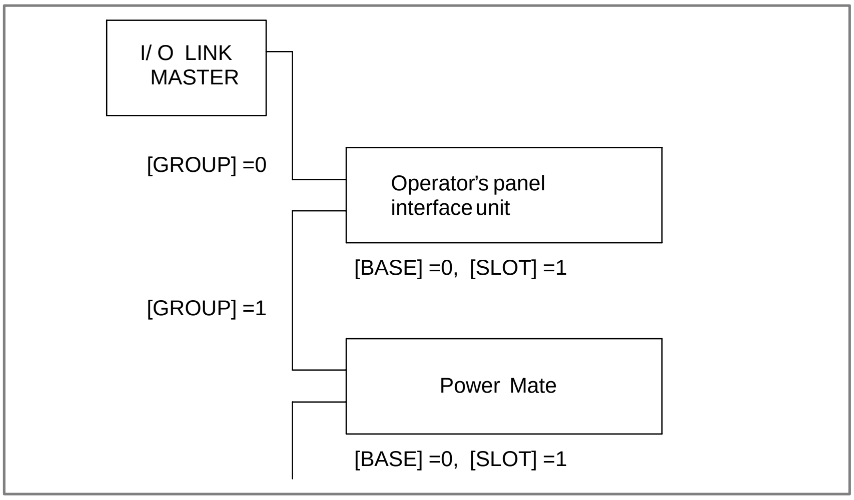
Setting:

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."

NOTE
On the PMC–SB6/SB7 for Series 16*i*/18*i*/21*i*/160*i*/180*i*/210*i*, when programming software supports the I/O Link expansion, I/O Unit–MODEL B can be assigned to the I/O Link channel 2 area. But total number of groups of channel 1 and channel 2 of I/O Unit –MODEL B available at the same time is up to 8 when I/O device is connected to channel 2. To link the I/O Link channel 2 actually, the option of the I/O Link expansion is necessary.

3.2.5 Power Mate Assignment

When a Power Mate–D/H or Power Mate *i*–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:

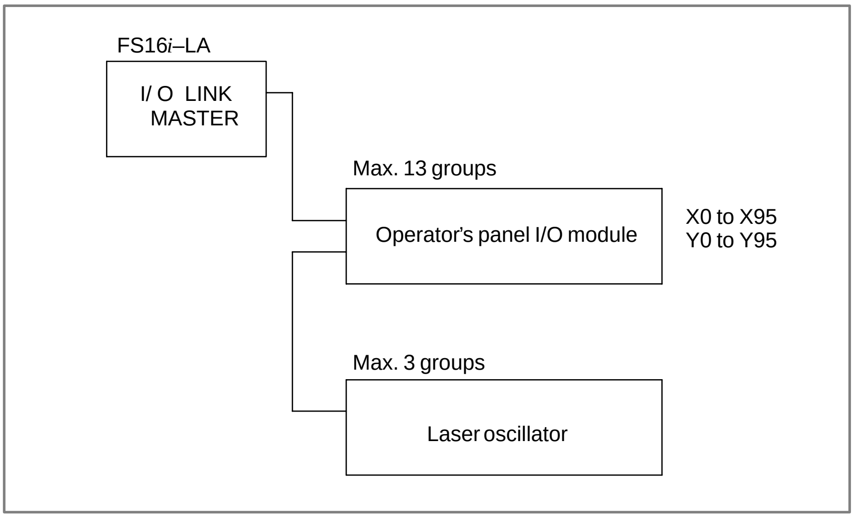
When a Power Mate–D of 256/256 points is connected with group 1, input the undermentioned assignment data.
·Input side : "1.0.1.OC03I"
·Output side : "1.0.1.OC03O"

NOTE
On the PMC–SB6/SB7 for Series 16*i*/18*i*/21*i*/160*i*/180*i*/210*i*, when programming software supports the I/O Link expansion, Power Mate–D/H or Power Mate *i*–D/H can be assigned to the I/O Link channel 2 area. To link the I/O Link channel 2 actually, the option of the I/O Link expansion is necessary.

3.2.6 FS16*i*–LA Assignment

On the FS16*i*–LA, the laser oscillator is connected as part of the I/O Link. 3 groups (DI/DO=256/256 points) are used for the laser oscillator interface. 13 groups, 768/768 points (X0 to X95, Y0 to Y95) of the I/O Link channel 1 can be used for the ladder diagram. 16 groups, 1024/1024 points (X200 to X327, Y200 to Y327) of the I/O Link channel 2 can be used for the ladder diagram. For details on the laser oscillator connection, see the FS16*i* Series CONNECTION MANUAL (B–63003EN), FANUC I/O Link connection.

Connection Example



Method of Assignment

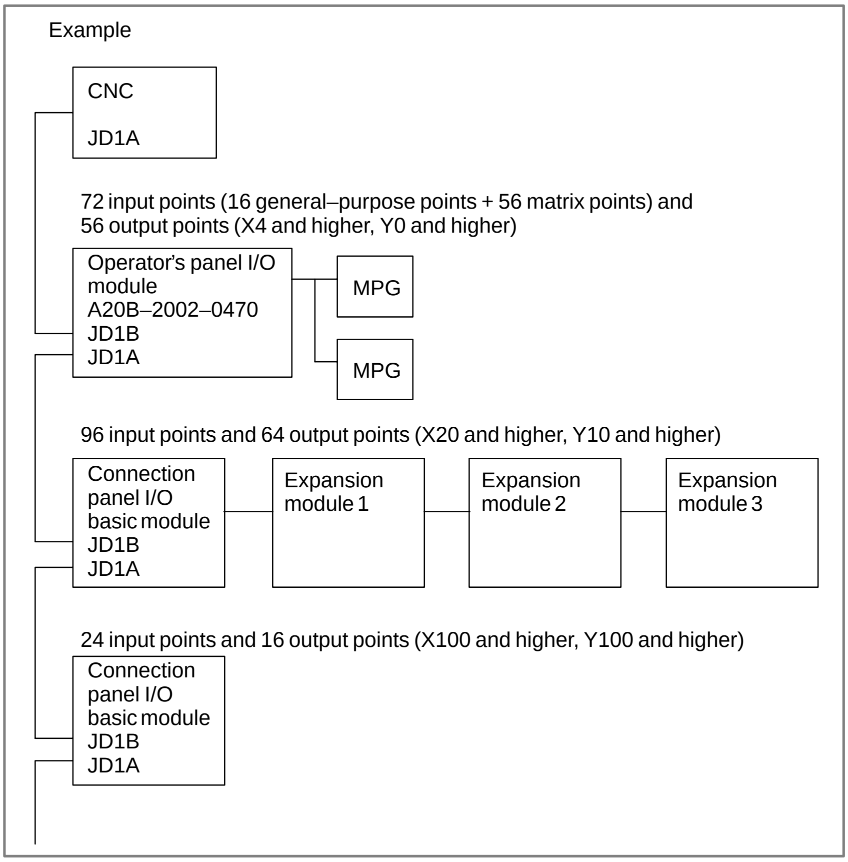
Specify 0 to 12 in [GROUP].
For details on how to assign [BASE] and [SLOT], see 3.2.2 Assignment of I/O Module Addresses.

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

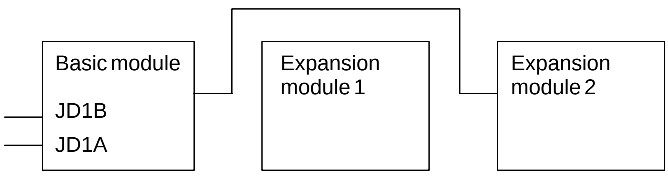
Refer to the following manuals for an explanation of how the connection panel I/O module signals are mapped:

- "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.

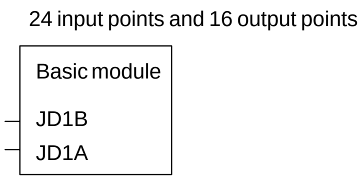
CAUTION

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



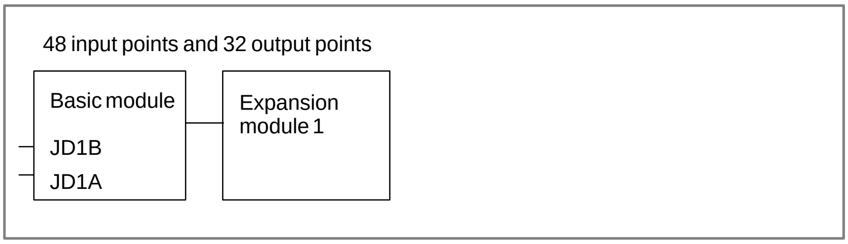
(a) If DO alarm detection is not used

- No manual pulse generator:
  Input X = "CM03I" and output Y = "CM02O"
- One manual pulse generator:
  Input X = "CM13I" and output Y = "CM02O"

(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



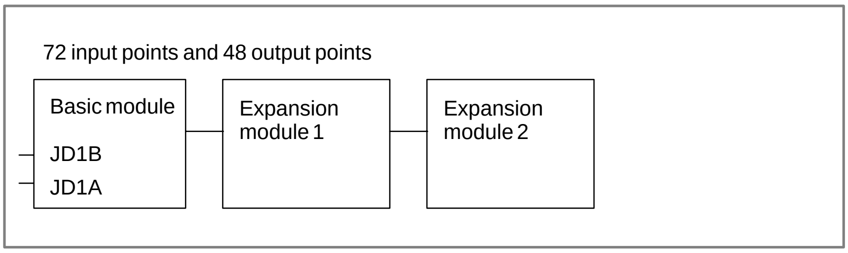
(a) If DO alarm detection is not used

- No manual pulse generator:
  Input X = "CM06I" and output Y = "CM04O"

(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



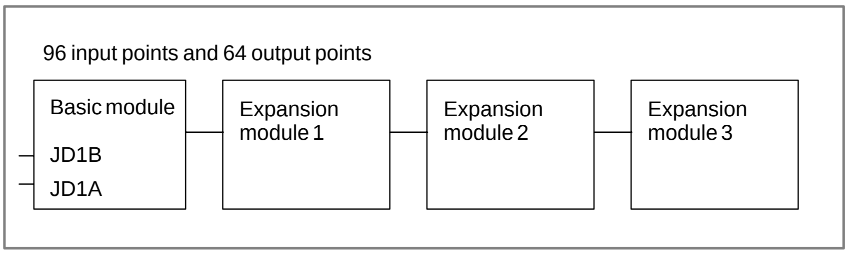
(a) If DO alarm detection is not used

- No manual pulse generator:
  Input X = "CM09I" and output Y = "CM06O"
- One manual pulse generator:
  Input X = "CM13I" and output Y = "CM06O"

(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

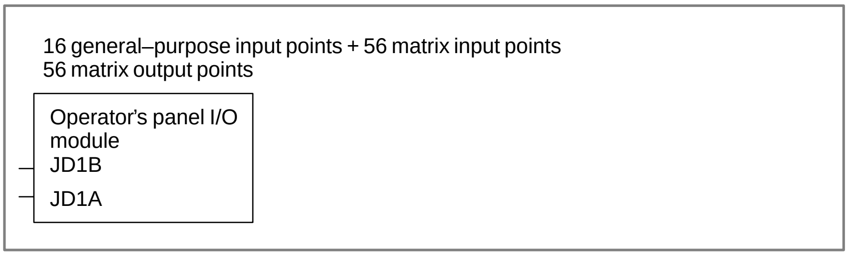
- Input X = “CM16I” and output Y = “CM08O” no matter how many manual pulse generators are used

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)

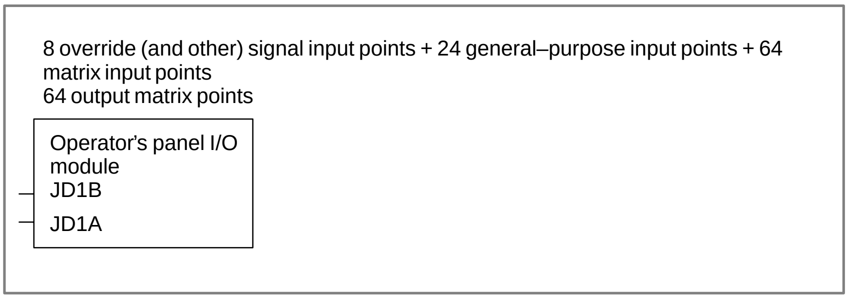
(a) If DO alarm detection is not used

- 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

NOTE

On the PMC–SB6/SB7 for Series 16*i*/18*i*/21*i*/160*i*/180*i*/210*i*, when programming software supports the I/O Link expansion, I/O Connection, I/O Module and Operator's Panel I/O Module can be assigned to the I/O Link channel 2 area. But Manual Pulse Generator can not be assigned to channel 2 area.

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.

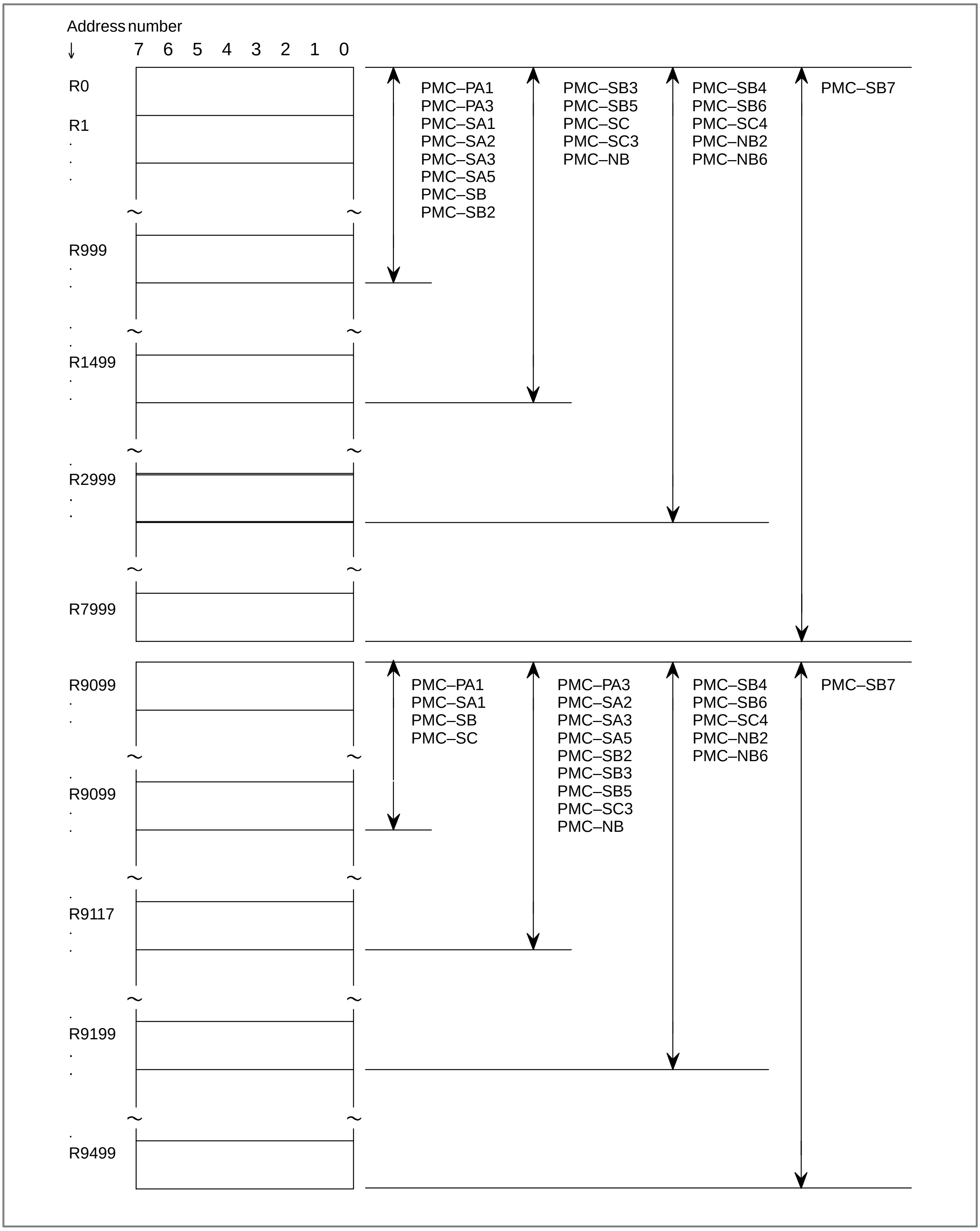
Model	PA1	PA3
Number of bytes	1100	1118

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

Model	SB	SB2	SB3/SB5	SB4/SB6	SB7
Number of bytes	1100	1118	1618	3200	8500

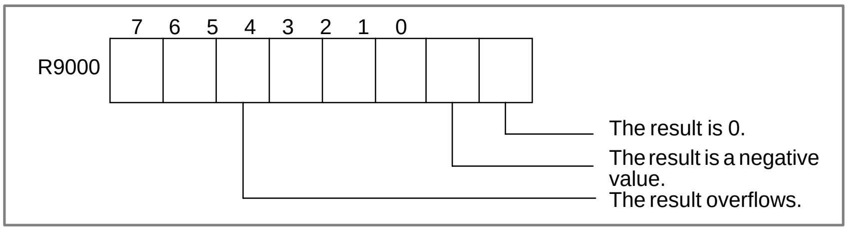
Model	SC	SC3	SC4
Number of bytes	1600	1618	3200

Model	NB	NB2	NB6
Number of bytes	1618	3200	3200

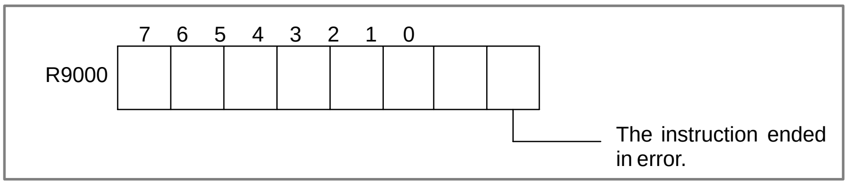

Address number
7 6 5 4 3 2 1 0
R0
R1
R999
R1499
R2999
R7999
R9099
R9099
R9117
R9199
R9499
PMC–PA1
PMC–PA3
PMC–SA1
PMC–SA2
PMC–SA3
PMC–SA5
PMC–SB
PMC–SB2
PMC–SB3
PMC–SB5
PMC–SC
PMC–SC3
PMC–NB
PMC–SB4
PMC–SB6
PMC–SC4
PMC–NB2
PMC–NB6
PMC–SB7
PMC–PA1
PMC–SA1
PMC–SB
PMC–SC
PMC–PA3
PMC–SA2
PMC–SA3
PMC–SA5
PMC–SB2
PMC–SB3
PMC–SB5
PMC–SC3
PMC–NB
PMC–SB4
PMC–SB6
PMC–SC4
PMC–NB2
PMC–NB6
PMC–SB7

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)

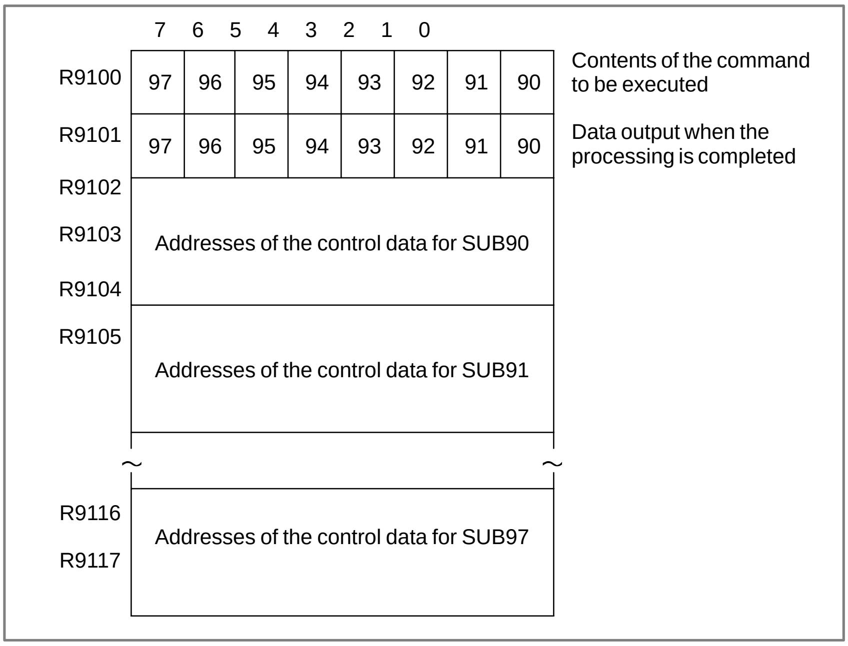


(3) R9002 to R9005 (Operation output registers for the DIVB functional instruction)
The data remaining after the DIVB functional instruction is executed is output.

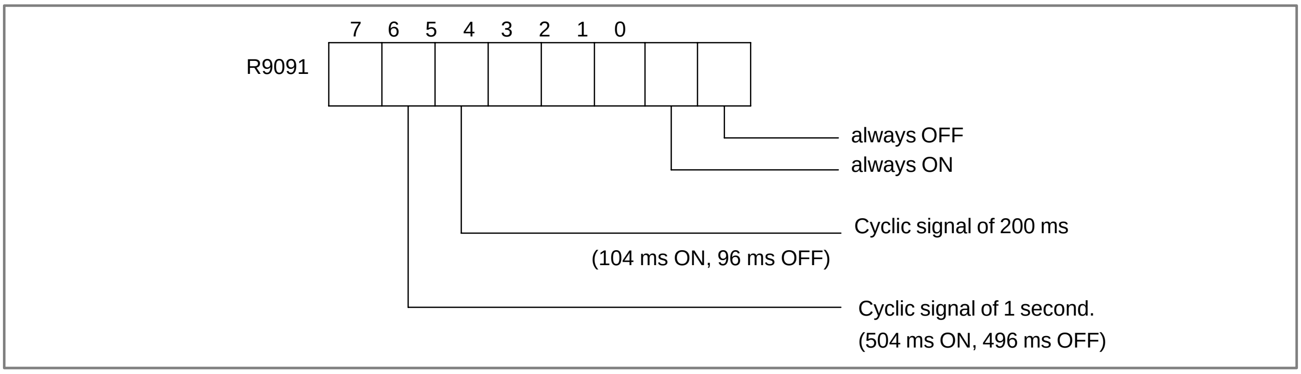
(4) R9010 to R9027 (Interface area for the FNC9x functional instruction) (PMC–SC only)
The area is provided as an interface between the FNC9x functional instruction to be executed and a desired function.

	7	6	5	4	3	2	1	0	
R9010	97	96	95	94	93	92	91	90	Contents of the command to be executed
R9011	97	96	95	94	93	92	91	90	Data output when the processing is completed
R9012 R9013	Addresses of the control data for SUB90								
R9014 R9015	Addresses of the control data for SUB91								
~	~								
R9026 R9027	Addresses of the control data for SUB97								

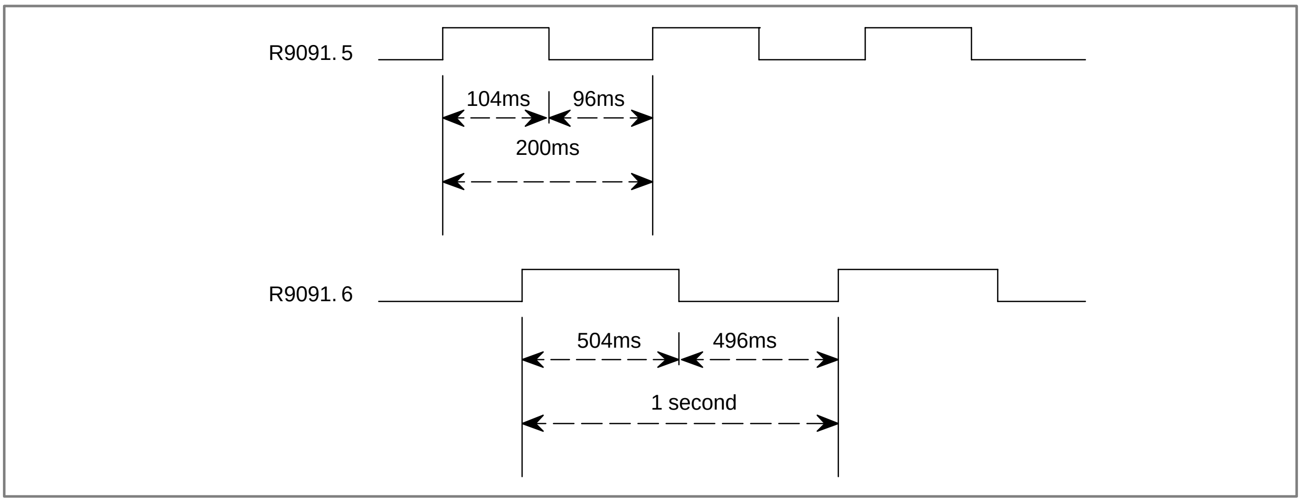
(5) R9100 to R9117 (Interface area for the FNC9x functional instruction) (PMC–SC3/SC4/NB/NB2 only)
The area is provided as an interface between the FNC9x functional instruction to be executed and a desired function.



(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.



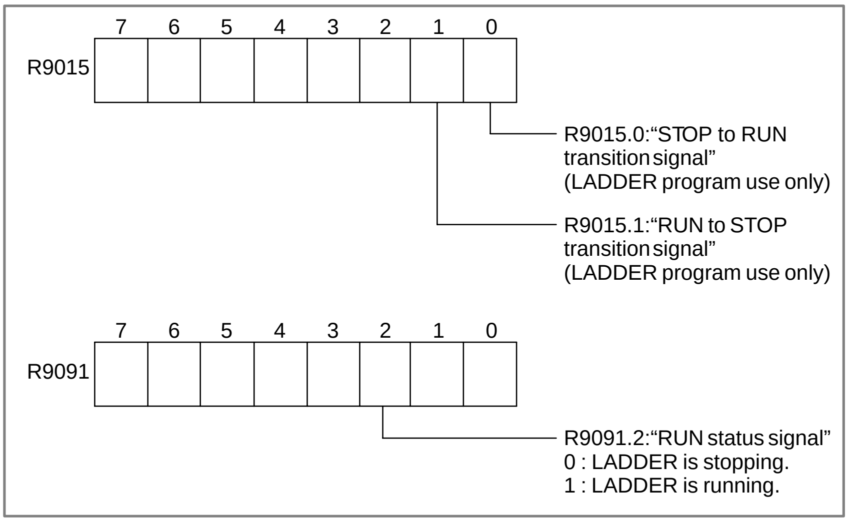
(7) RUN to STOP Transition Signal, STOP to RUN Transition Signal and RUN Status Signal (PMC–SB7)

(a) Overview

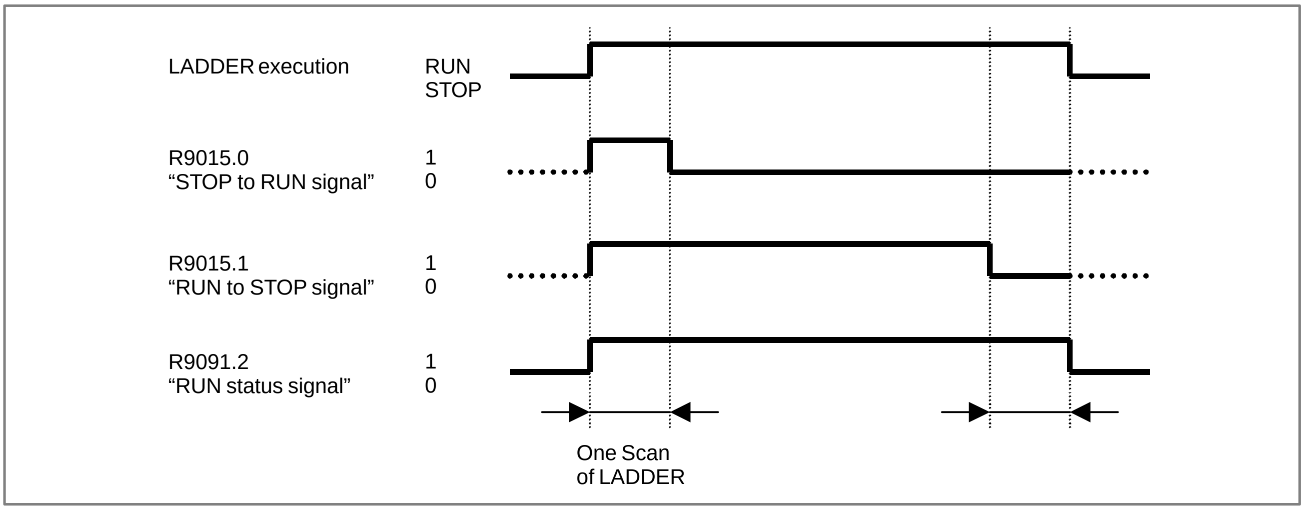
(i) “RUN to STOP transition signal” and ”STOP to RUN transition signal”
The “RUN to STOP transition signal” and the ”STOP to RUN transition signal” for LADDER program execution are to handle these events in a LADDER program.

(ii) “RUN status signal”
“RUN status signal” is to notify the status of LADDER and PMC C program execution to other systems or programs, such a network board, C executor program, FOCAS1 Ethernet or HSSB library etc.

(b) Address



(c) Behavior



(i) “STOP to RUN transition signal” (R9015.0)
When a STOP to RUN event is detected on system software, this signal will be turned on during the 1st scan of LADDER program after LADDER started. This signal has individual status corresponding the scan of each LADDER execution level like R9000. This signal is completely turned on during whole of the 1st scan in any execution level of LADDER program.

– When does a STOP to RUN event happen?
– LADDER starting at every power on cycle
– pressing a ”RUN” soft–key on a PMC screen
– “RUN” commanded by FAPT LADDER–III or LADDER EDITING PACKAGE

Referring this signal in a LADDER program, you can recognize and handle a ”STOP to RUN” transition event. You can program a pre–processing for LADDER execution.

NOTE
This signal is available only in LADDER program. Don’t refer this signal on other systems or programs, such a network board, C executor program, FOCAS1 Ethernet or HSSB liberally etc. because this signal has individual status in each LADDER execution.

(ii) “RUN to STOP transition signal” (R9015.1)
When a RUN to STOP event is detected on system software, this signal will be turned off during the last scan of LADDER program before LADDER stopped. This signal has individual status corresponding the scan of each LADDER execution level like R9000. This signal is completely turned off during the last scan before any execution level of LADDER program stops.

– When does a RUN to STOP event happen?
– pressing a ”STOP” soft–key on a PMC screen
– ”STOP” commanded by FAPT LADDER–III or LADDER EDITING PACKAGE

– pressing a ”EDIT”–>”IOMDL” soft–key on a PMC screen
– pressing a ”EDIT”–>”MESAGE” soft–key on a PMC screen
– pressing a ”EDIT”–>”SYMBOL” soft–key on a PMC screen
– pressing a ”EDIT”–>”CLEAR” soft–key on a PMC screen
– pressing a ”SYSPRM” soft–key on a PMC screen
– storing a LADDER or PMC C language program to PMC on PMC ”I/O” screen
– storing a LADDER program to PMC using FAPT LADDER–III or LADDER EDITING PACKAGE

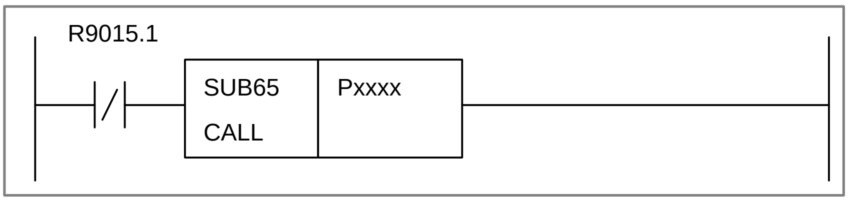
Referring this signal in a LADDER program, you can recognize and handle a ”RUN to STOP” transition event. You can program a post–processing for LADDER execution (i.e. pre–processing for stopping of LADDER execution). For example, set or reset any appropriate signals into certain condition for the safety.

NOTE
1 This signal is available only in LADDER program. Don't refer this signal on other systems or programs, such a network board, C executor program, FOCAS1 Ethernet or HSSB liberally etc. because this signal has individual status in each LADDER execution level.
2 You can not handle this event using this signal at a power off sequence and system alarm of CNC in which the execution of LADDER and I/O scanning are completely shut down.

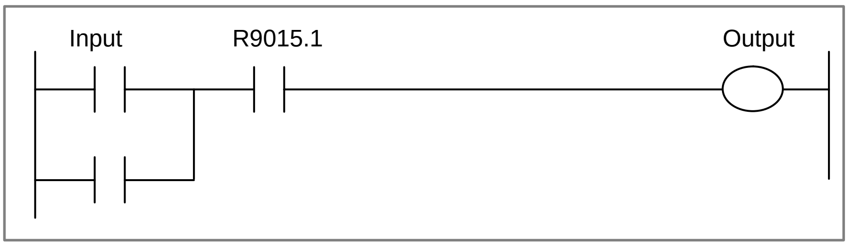
(iii) “RUN status signal” (R9091.2)
Referring this signal on other systems or programs, such a network board, C executor program, FOCAS1 Ethernet or HSSB liberally etc. you can know the status of LADDER and PMC C language program execution.

(d) Example of use

(i) Calling a subprogram at RUN to STOP condition.



(ii) Forcing off a specified output signal programmed in 1st level at RUN to STOP transition.



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” × 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	SB7
Number of bytes	25	25	25	125	500
Number of messages	200	200	200	1000	2000

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

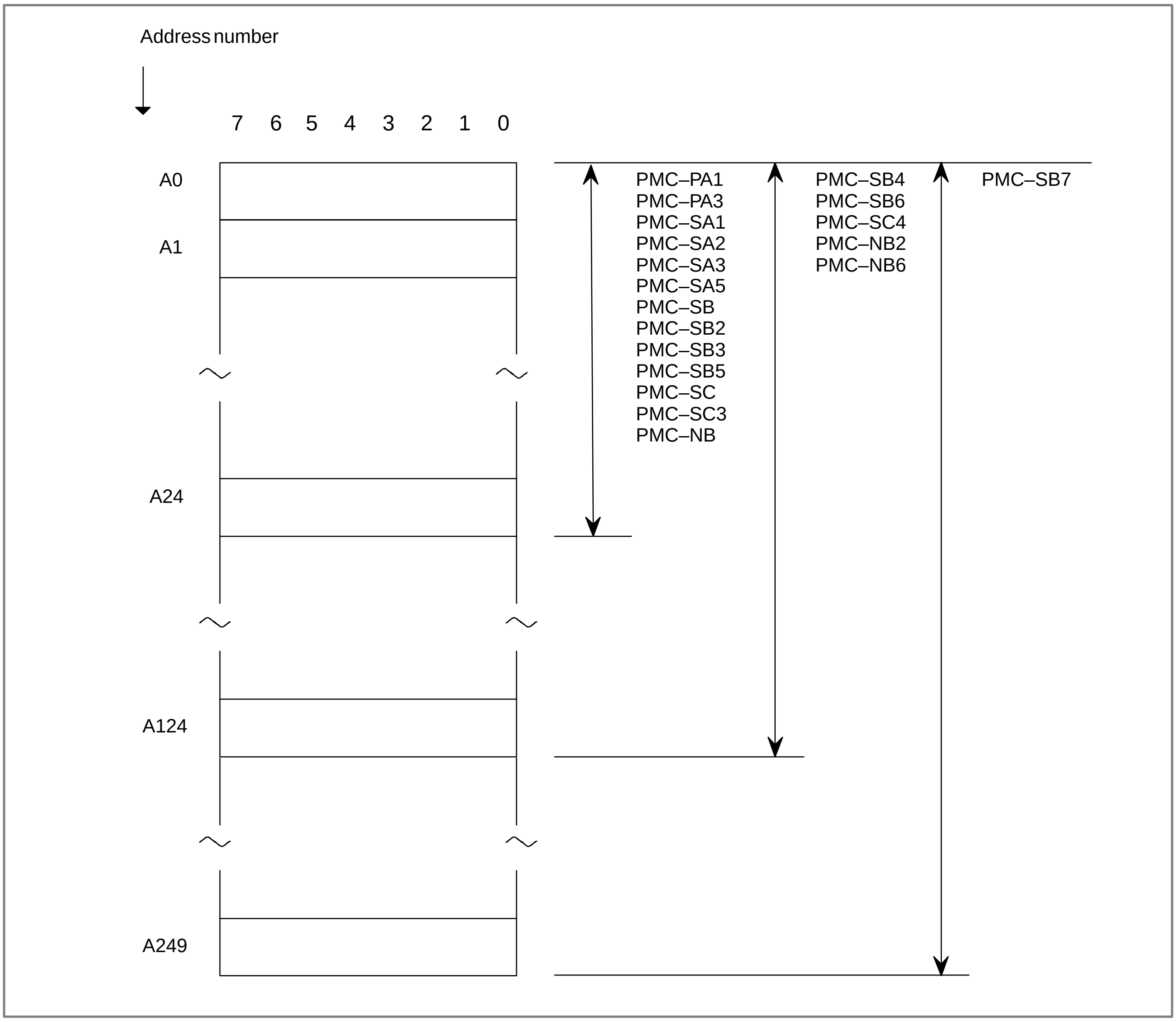


Fig. 3.4 Address of message display reguest

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	SB7
Number of bytes	80	80	80	200	400
Number of counters	20	20	20	50	100

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

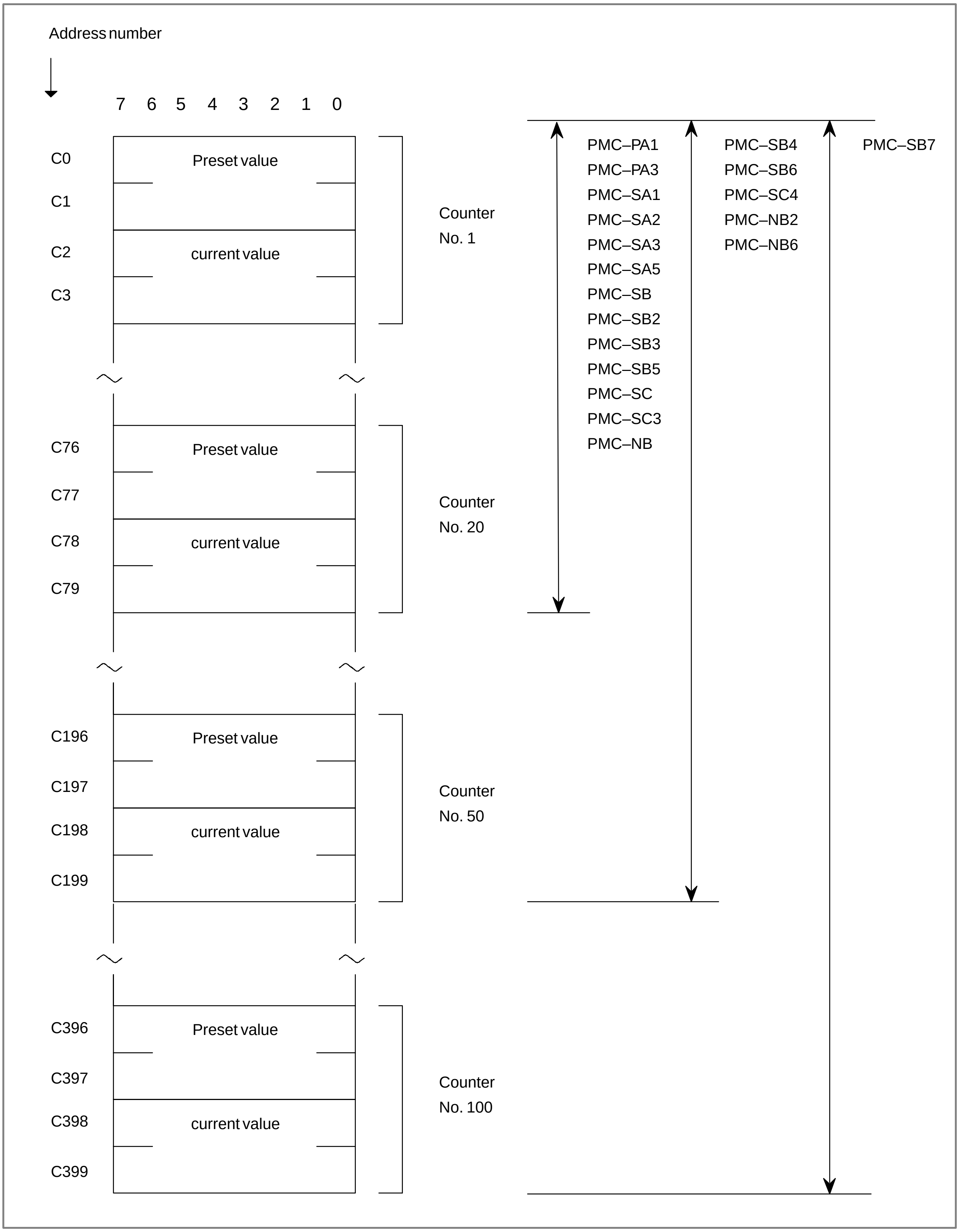


Fig. 3.5 Address of Counter

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	SB7
Number of bytes	20	20	20	50	120
Nonvolatile memory control address	K16	K16	K16	K16	K16
PMC control software parameter	K17 to K19	K17 to K19	K17 to K19	K900 to K909	K900 to K919

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

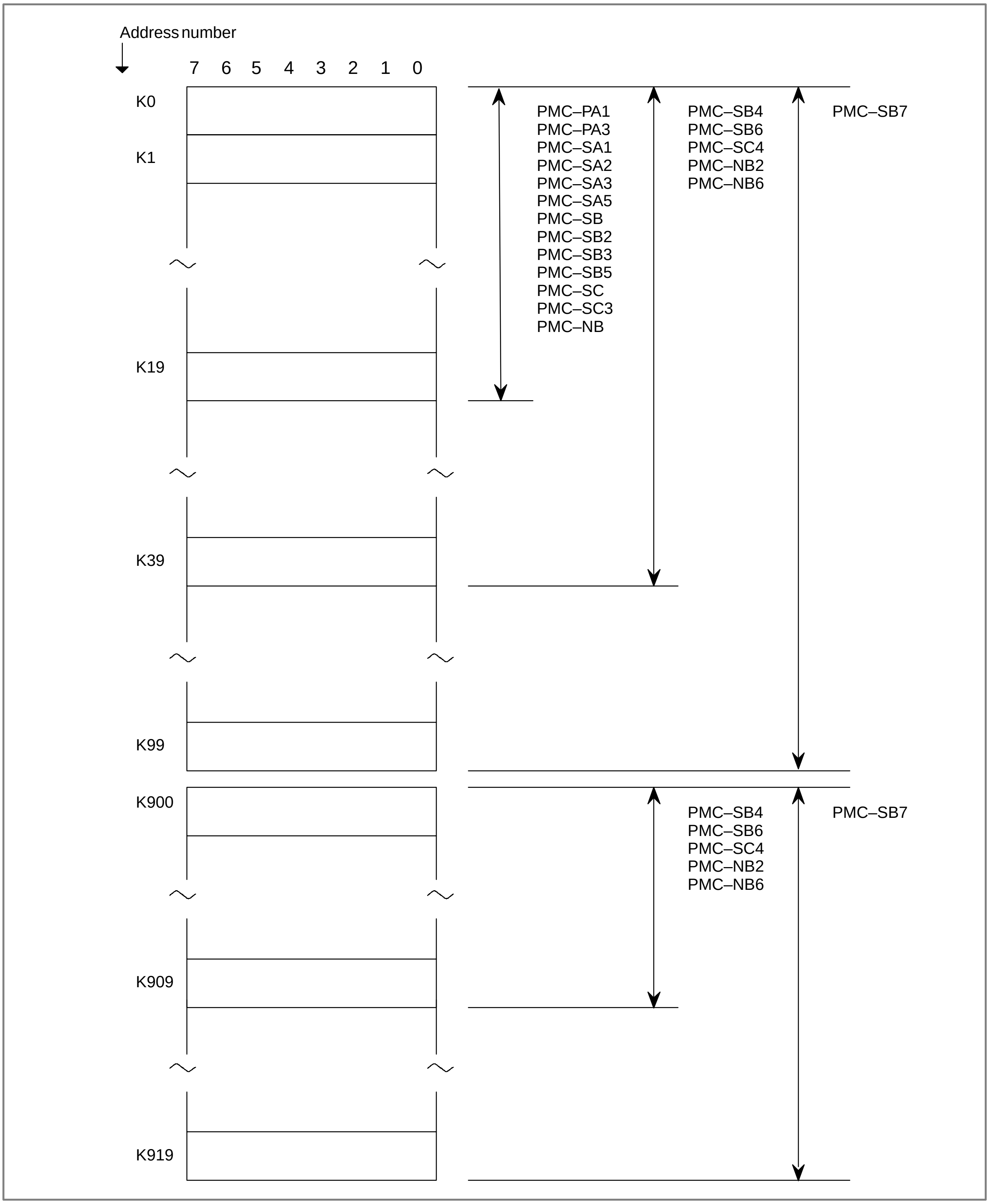


Fig. 3.6 Address of keep relay and nonvolative memory control

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	SB7
Number of bytes	1860	1860	3000	8000	10000

Model	SC	SC3	SC4
Number of bytes	3000	3000	8000

Model	NB	NB2	NB6
Number of bytes	3000	8000	8000

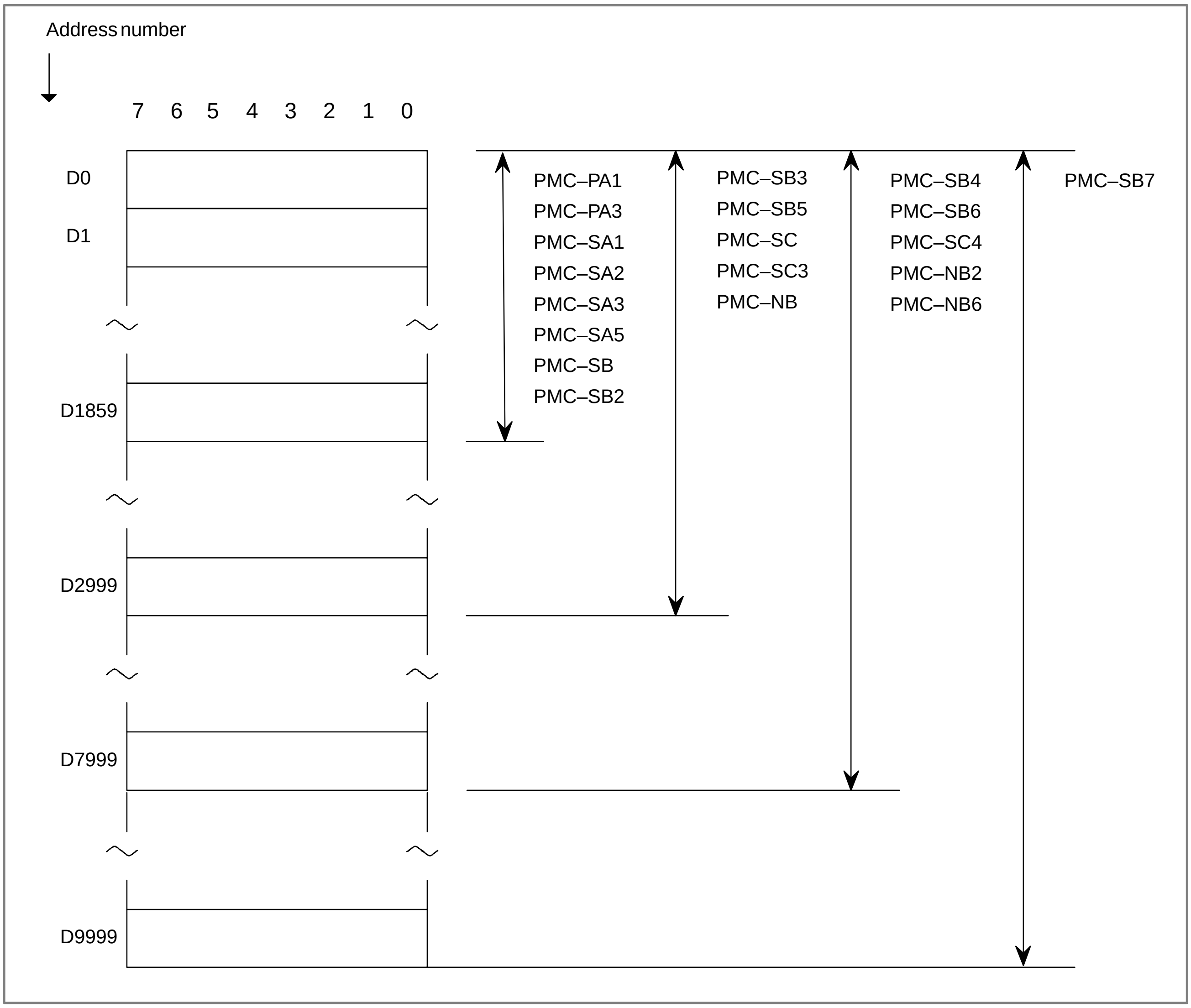


Fig. 3.7 Address of data table

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	SB7
Number of bytes	80	80	80	300	500
Number of timers	40	40	40	150	250

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

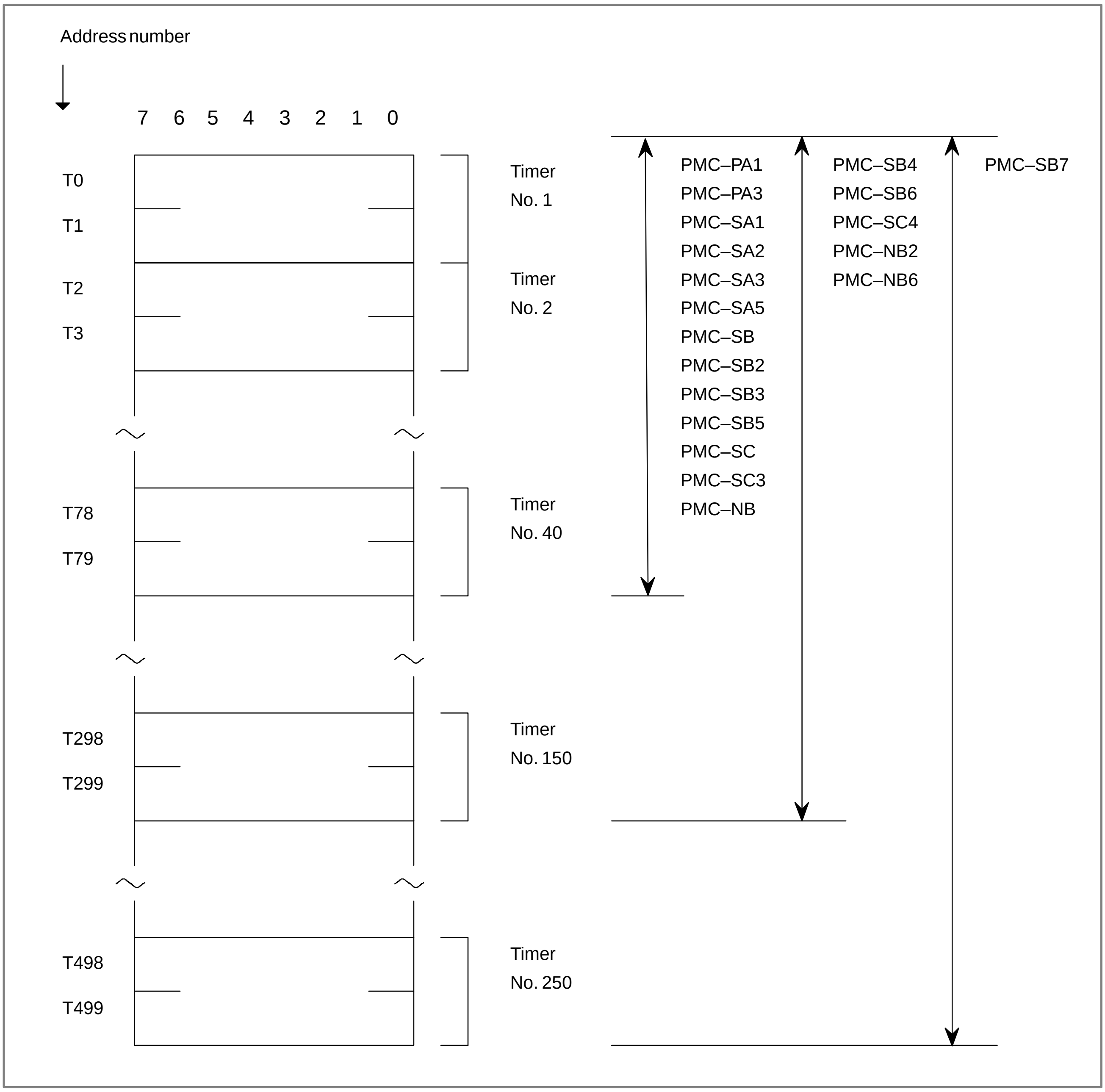


Fig. 3.8 Timer address

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	SB7
Number of labels	–	–	9999	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	SB7
Number of subprograms	–	–	512	2000	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.

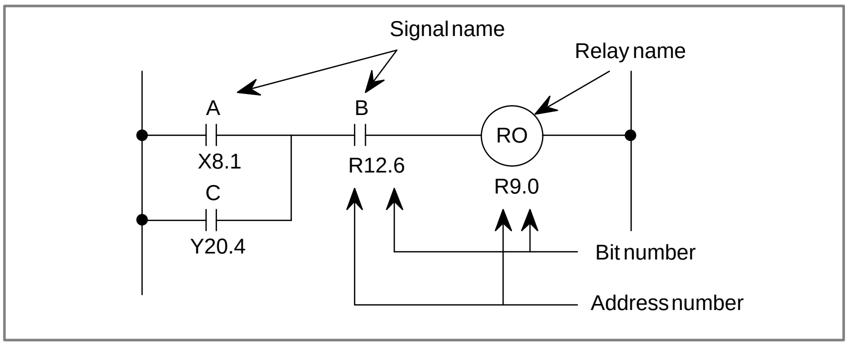


Fig. 4 (a) Address of signal

(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)) .

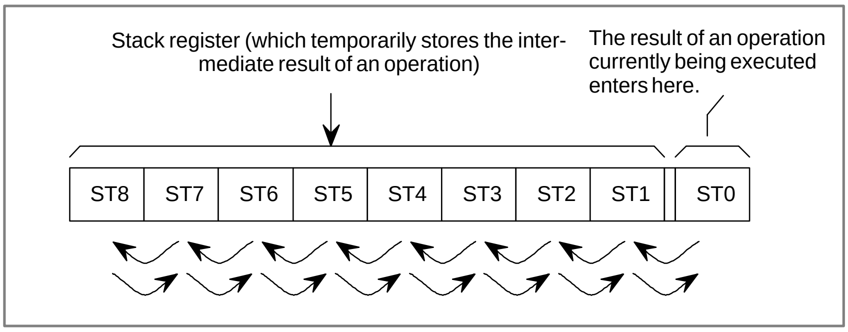


Fig. 4 (b)

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

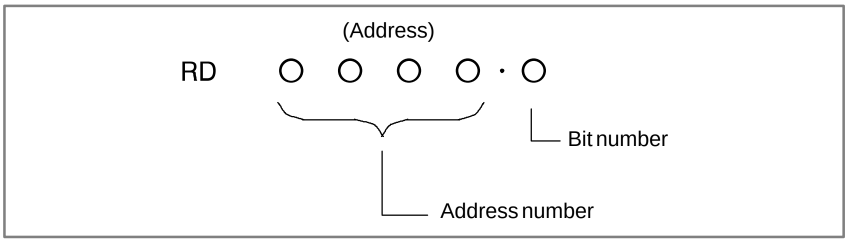
No.	Instruction	Model								
		PMC– PA1	PMC– PA3	PMC– SA1/ SA2	PMC– SB/ SB2	PMC– SC	PMC– SA3/ SA5	PMC– SB3/ SB4/ SB5/ SB6/ SB7	PMC– SC3/ SC4	PMC– NB/ NB2/ NB6
1	RD	○	○	○	○	○	○	○	○	○
2	RD.NOT	○	○	○	○	○	○	○	○	○
3	WRT	○	○	○	○	○	○	○	○	○
4	WRT.NOT	○	○	○	○	○	○	○	○	○
5	AND	○	○	○	○	○	○	○	○	○
6	AND.NOT	○	○	○	○	○	○	○	○	○
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).

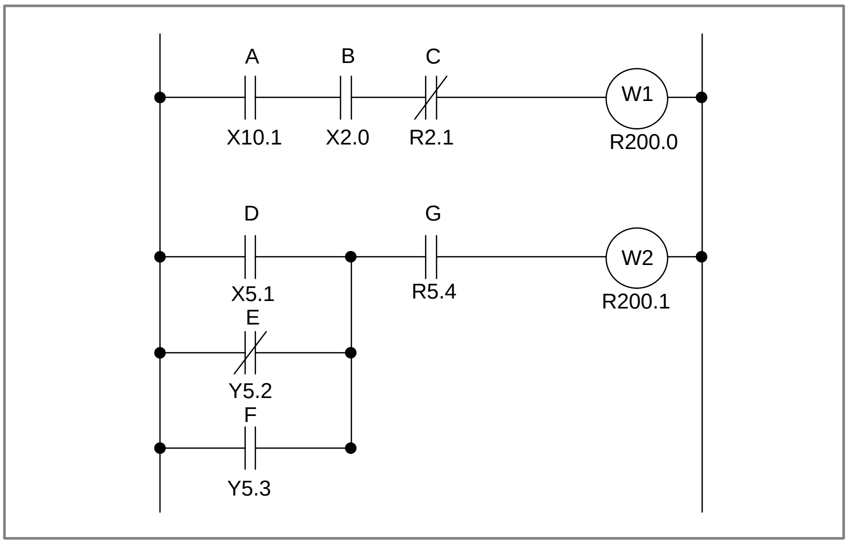


Fig. 4.1.1 Ladder diagram

Table 4.1.1 Coding for Fig. 4.1.1

Coding sheet

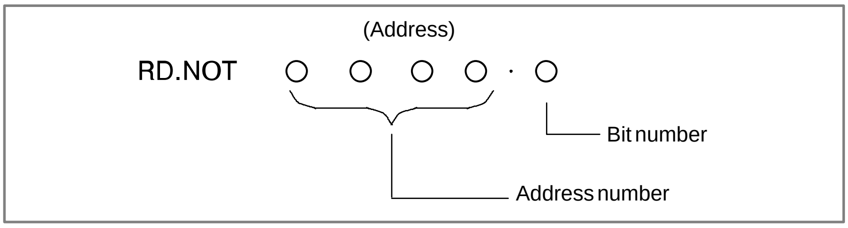
Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	X10 .	1	A
2	AND	X 2 .	0	B
3	AND . NOT	R2 .	1	C
4	WRT	R200 .	0	W1 output
5	RD	X5 .	1	D
6	OR . NOT	Y5 .	2	E
7	OR	Y5 .	3	F
8	AND	R5 .	4	G
9	WRT	R200 .	1	W2 output
10				

Status of operating result

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

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.

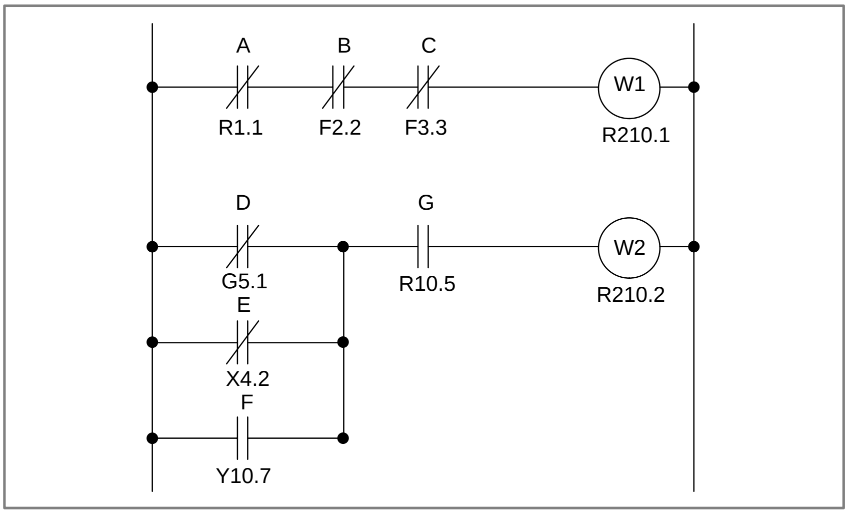


Fig. 4.1.2 Ladder diagram

Table 4.1.2 Coding for Fig. 4.1.2

Coding sheet

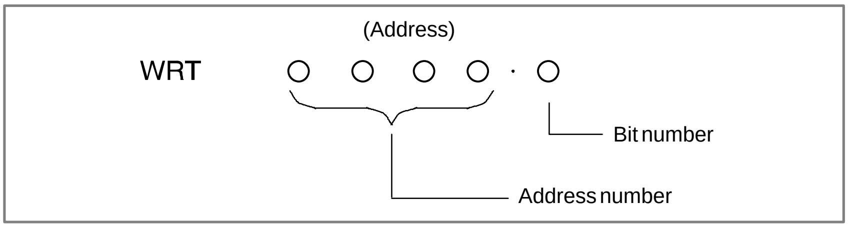
Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD. NOT	R1 .	1	A
2	AND . NOT	F2 .	2	B
3	AND . NOT	F3 .	3	C
4	WRT	R210 .	1	W1 output
5	RD. NOT	G5 .	1	D
6	OR . NOT	X4 .	2	E
7	OR	Y10 .	7	F
8	AND	R10 .	5	G
9	WRT	R210 .	2	W2 output

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.

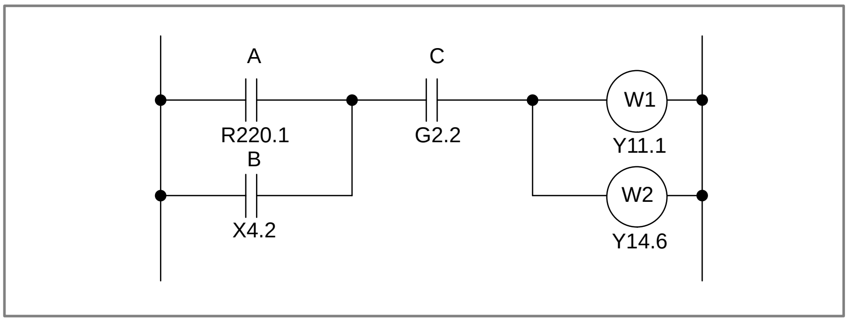


Fig. 4.1.3 Ladder diagram

Table 4.1.3 Coding for Fig. 4.1.3

Coding sheet

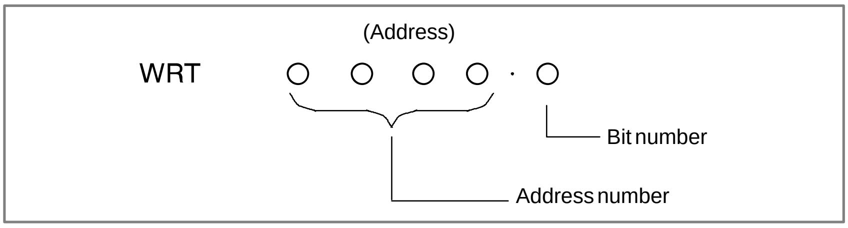
Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	R220 .	1	A
2	OR	X4 .	2	B
3	AND	G2 .	2	C
4	WRT	Y11 .	1	W1 output
5	WRT	Y14 .	6	W2 output

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.

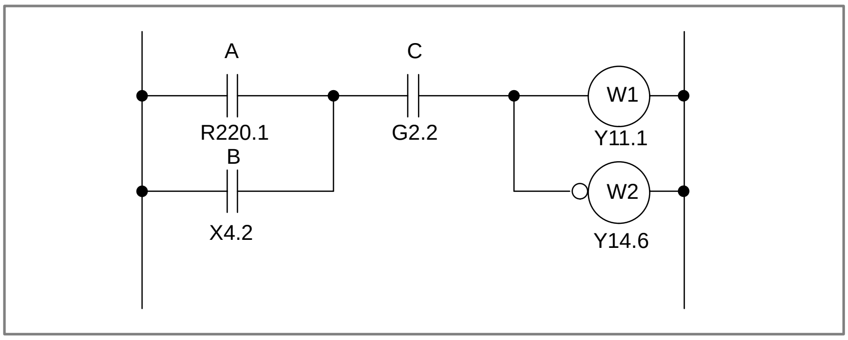


Fig. 4.1.4 Ladder diagram

Table 4.1.4 Coding for Fig. 4.1.4

Coding sheet

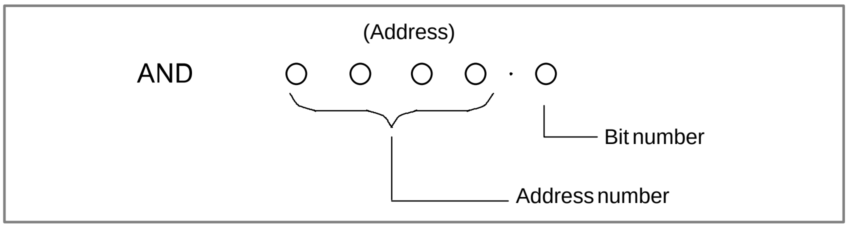
Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	R220 .	1	A
2	OR	X4 .	2	B
3	AND	G2 .	2	C
4	WRT	Y11 .	1	W1 output
5	WRT. NOT	Y14 .	6	W2 output

Status of operating result

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

4.1.5 AND

(1) Format

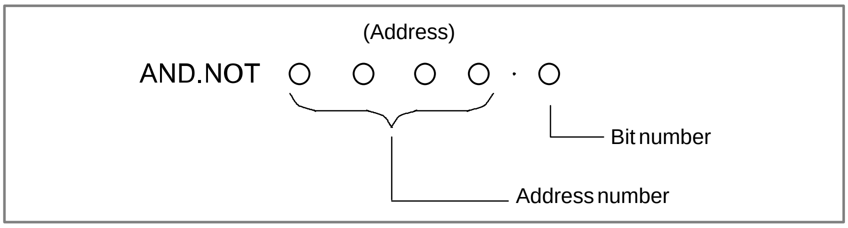


(2) Induces a logical product.

(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

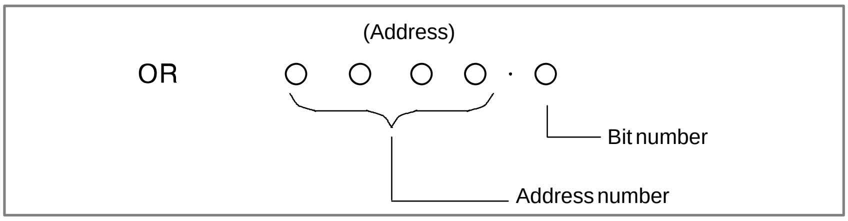


(2) Inverts the status of a signal at a specified address and induces a logical product.

(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

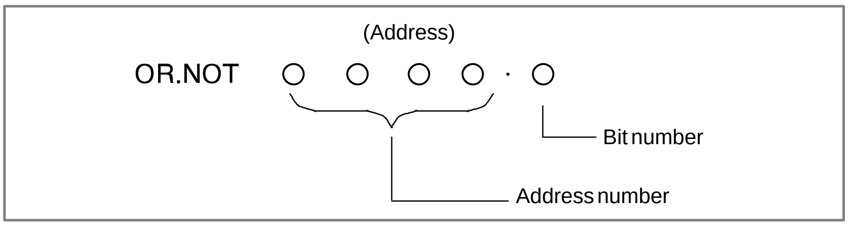


(2) Induces a logical sum.

(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

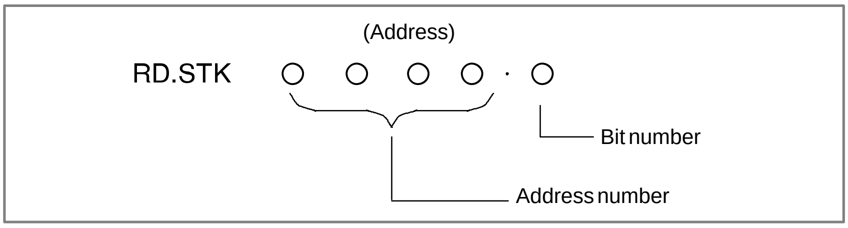


(2) Inverts the status of a signal at a specified address and induces a logical sum.

(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.

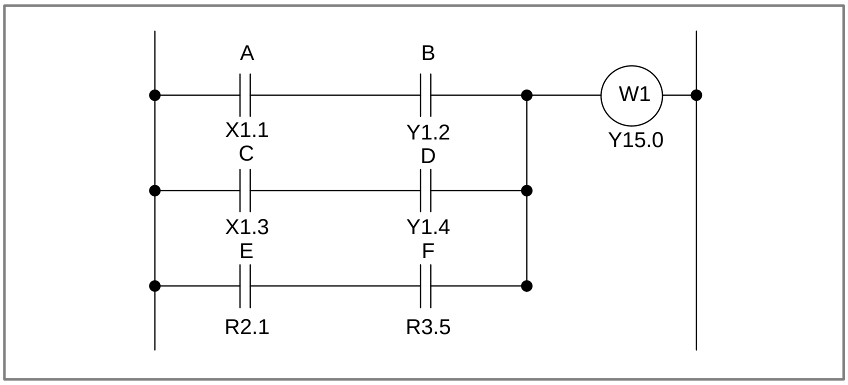


Fig. 4.1.9 Ladder diagram

Table 4.1.9 Coding for Fig. 4.1.9

Coding sheet

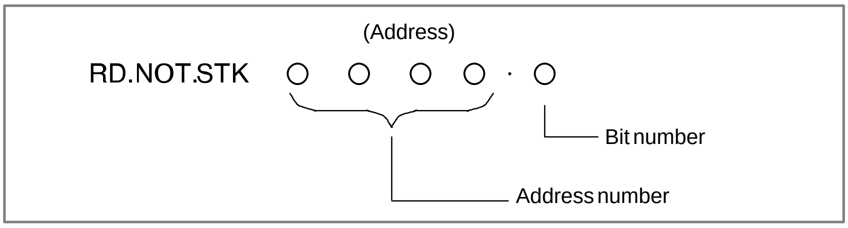
Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	X1 .	1	A
2	AND	Y1 .	2	B
3	RD. STK	X1 .	3	C
4	AND	Y1 .	4	D
5	OR. STK			
6	RD. STK	R2 .	1	E
7	AND	R3 .	5	F
8	OR.STK			
9	WRT	Y15 .	0	W1 output
10				

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.

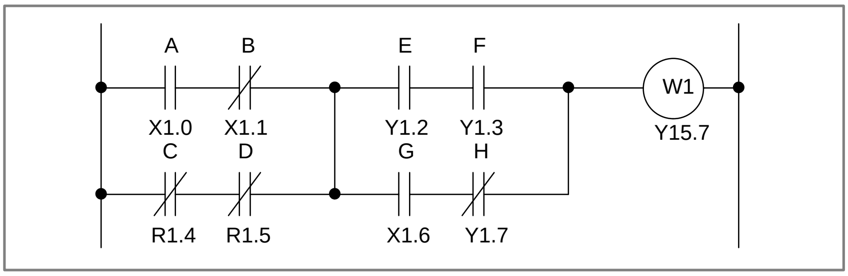


Fig. 4.1.10 Ladder diagram

Table 4.1.10 Coding for Fig. 4.1.10

Coding sheet

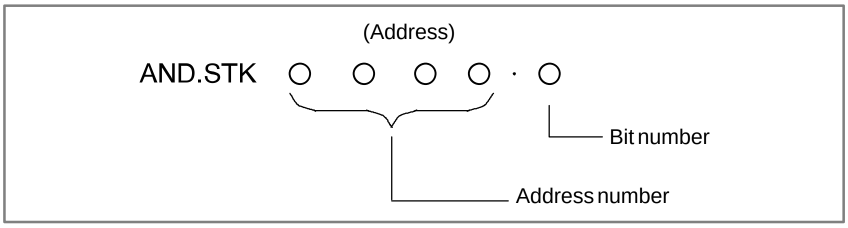
Step Number	Instruction	Address No.	Bit No.	Remarks
1	RD	X1 .	0	A
2	AND. NOT	X1 .	1	B
3	RD.NOT.STK	R1 .	4	C
4	AND. NOT	R1 .	5	D
5	OR. STK			
6	RD. STK	Y1 .	2	E
7	AND	Y1 .	3	F
8	RD.STK	X1 .	6	G
9	AND. NOT	Y1 .	7	H
10	OR. STK			
11	AND. STK			
12	WRT	Y15 .	7	W1 output
13				
14				

Status of operating result

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

4.1.11 AND. STK

(1) Format

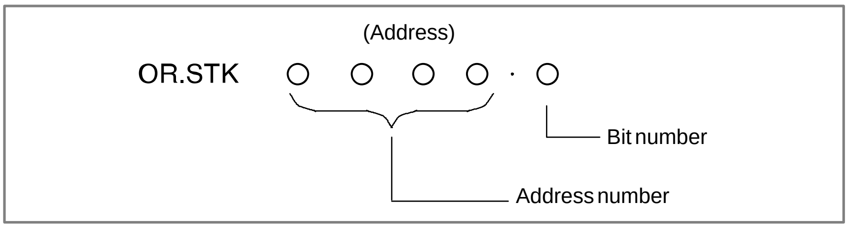


(2) Induces a logical product from the operation results in ST0 and ST1, sets the result in ST1, and shifts the stack register right one bit.

(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



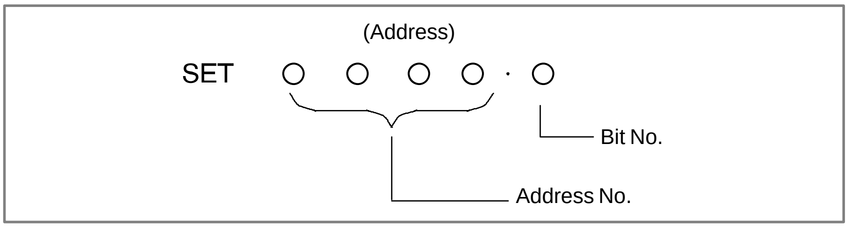
(2) Induces a logical sum from the operation results in ST0 and in ST1, sets the result in ST1, and shifts the stack register right one bit.

(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.

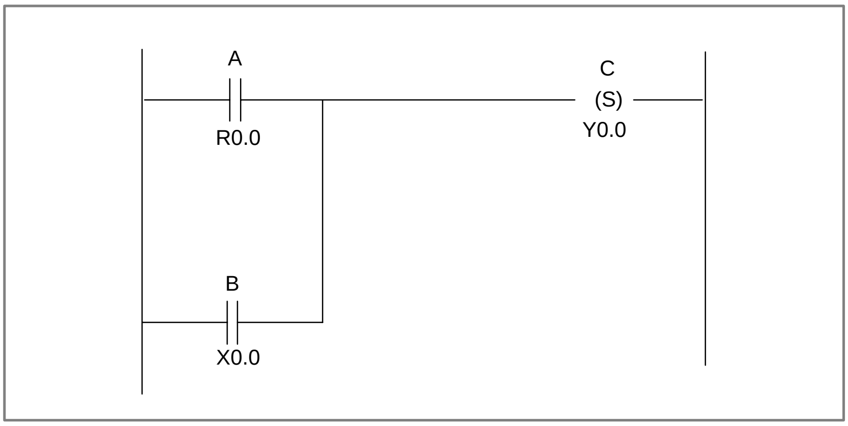


Fig. 4.1.13 Ladder diagram

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

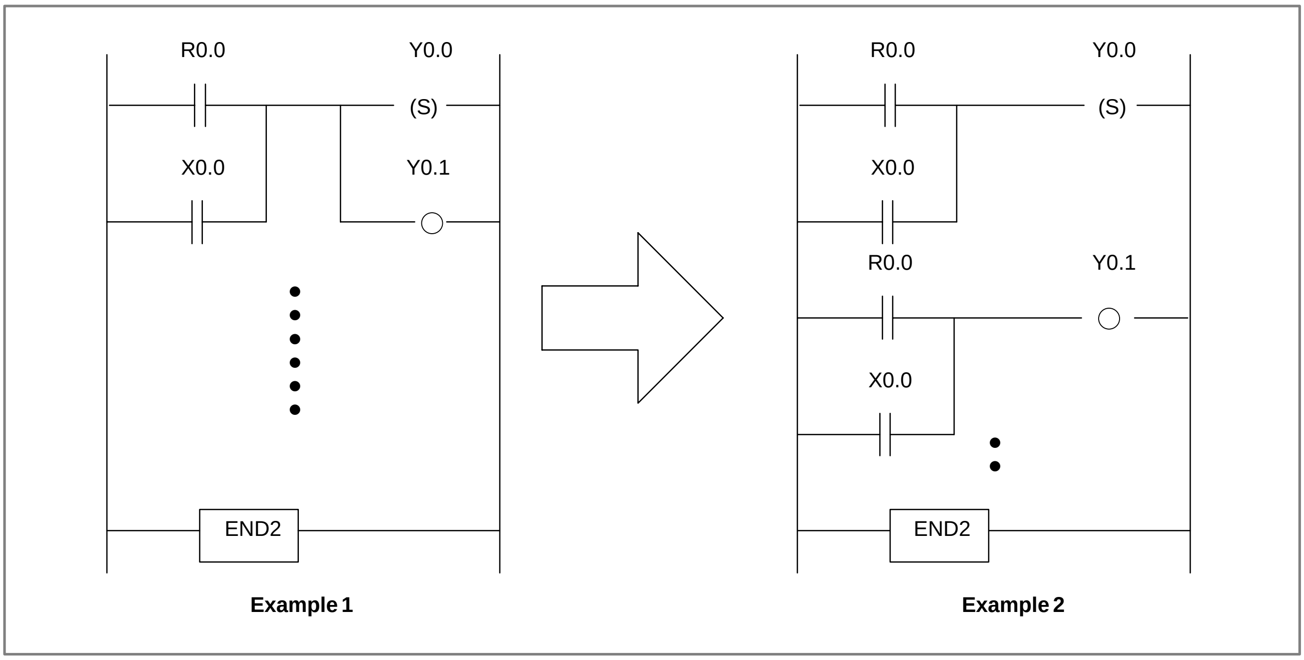
ST2	ST1	ST0
	A	C
	A+B	C
	–	(A+B) +C

(4) Remarks

The use of the PMC–PA3, SA2, SA3, SB2, SB3, and SC3 is restricted as follows:

(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.

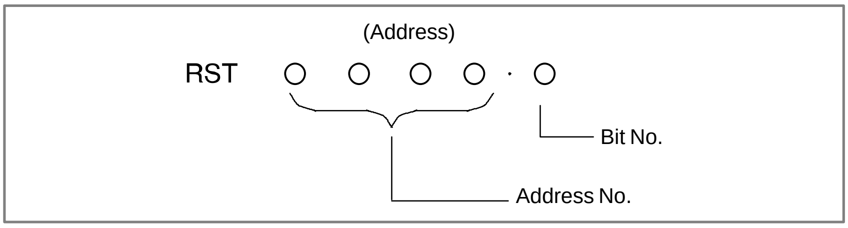
  The operation of SET/RST in the section of COM/COME is as follows.

  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.

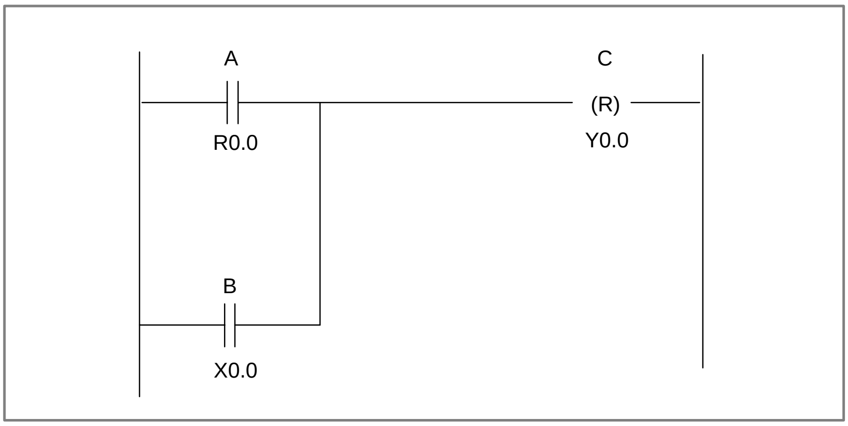


Fig. 4.1.14 Ladder diagram

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

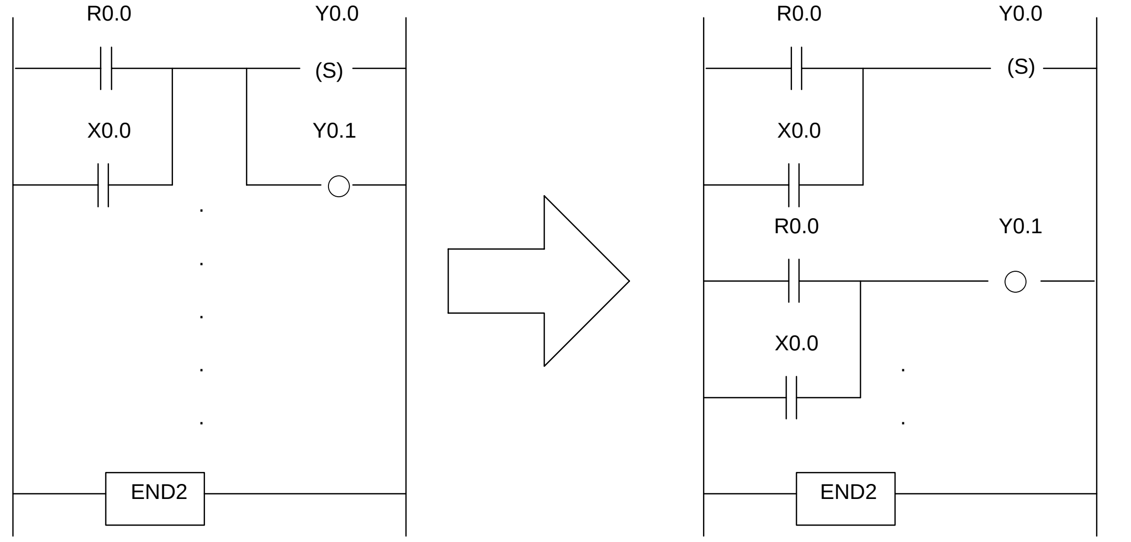
ST2	ST1	ST0
	A	C
	A+B	C
	–	$\overline{(A+B)}$ +C

(4) Remarks

The use of the PMC–PA3, SA2, SA3, SB2, SB3, and SC3 is restricted as follows:

(a) Restriction of using

Do not use SET/RST like the following example 1, use them alone like the following example 2.



Example 1 **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	○	○
NOP	SUB70	S70	No operation	×	△

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

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

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
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	×	×	○	×	×	○	×	○	○	○

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
AXCTL	53	PMC axes control	○	○	○	○	○	○	○	○	×	×
NOP	70	No operation	×	×	×	×	×	×	×	×	△	○

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

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	○	○	○	○	×
NOP	70	No operation	△	△	△	×	×

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

CAUTION

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 (Caution)	△	△
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	○	○
NOP	70	No operation	△	△

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

CAUTION

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 Series 0*i*–MODEL A	
			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 Series 0*i*–MODEL A	
			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	×	○
NOP	70	No operation	×	×

× : 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 (Caution)	△	△
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	○	○
NOP	70	No operation	○	○

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

CAUTION

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	×	○
NOP	70	No operation	○	○

× : Cannot be used ○ : Can be used

Table 5 (a) Types and Processing of Functional Instructions (8) Part 1

Command Name	SUB Number	Description	Model: Series 21*i*–MODEL B PMC–SA1	Model: Series 16*i* /18*i* /21*i* – MODEL B PMC–SB7
END1	1	1st Level program end	○	○
END2	2	2nd Level program end	○	○
END3	48	3rd Level program end	×	○
TMR	3	Timer	○	○
TMRB	24	Fixed timer	○	○*2
TMRC	54	Timer	○	○*2
DEC	4	Decode	○	○
DECB	25	Binary decode	○	○
CTR	5	Counter	○	○
CTRB	56	Fixed counter	×	○*1
CTRC	55	Counter	○	○
ROT	6	Rotational control	○	○
ROTB	26	Binary rotational control	○	○
COD	7	Code conversion	○	○
CODB	27	Binary code conversion	○	○
MOVE	8	Move ANDed data	○	○
MOVOR	28	Move ORed data	○	○
MOVB	43	Move 1 byte	×	○
MOVW	44	Move 2 bytes (Word)	×	○
MOVD	47	Move 4 bytes (Double word)	×	○*1
MOVN	45	Move 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	×	○
PARI	11	Parity check	○	○
DCNV	14	Data convert	○	○
DCNVB	31	Binary data convert	○	○
COMP	15	Comparison	○	○
COMPB	32	Binary comparison	○	○
COIN	16	Coincidence check	○	○
SFT	33	Shift register	○	○
DSCH	17	Data search	○	○
DSCHB	34	Binary data search	○	○

× : Not available ○ : Available

Table 5 (a) Types and Processing of Functional Instructions (8) Part 2

Com-mand Name	SUB Number	Description	Model	
			Series 21*i*–MODEL B	Series 16*i* /18*i* /21*i* – MODEL B
			PMC–SA1	PMC–SB7
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	Define constant	○	○
NUMEB	40	Define binary constant	○	○
DISP	49	Display message	×	×*3
DISPB	41	Display message	○	○*2
EXIN	42	External data input	○	○
SPCNT	46	Spindle control	×	×*3
AXCTL	53	PMC axis control	○	○
WINDR	51	CNC window data read	○	○
WINDW	52	CNC window data write	○	○
FNC9X	9X	Arbitrary functional instruction (X=0 to 7)	×	○*4
MMC3R	88	MMC3 window data read	×	×*3
MMC3W	89	MMC3 window data write	×	×*3
MMCWR	98	MMC window data read	○	○
MMCWW	99	MMC window data write	○	○
PSGNL	50	Position signal output	×	×*3
PSGN2	63	Position signal output 2	×	×*3
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 ladder program	×	○
CALL	65	Conditional subprogram call	×	○
CALLU	66	Unconditional subprogram call	×	○
SP	71	Subprogram	×	○

× : Not available ○ : Available

Table 5 (a) Types and Processing of Functional Instructions (8) Part 3

Command Name	SUB Number	Description	Model	
			Series 21*i*–MODEL B	Series 16*i* /18*i* /21*i* – MODEL B
			PMC–SA1	PMC–SB7
SPE	72	End of subprogram	×	○
NOP	70	No operation (Net comment)	○	○

× : Not available ○ : Available

NOTE
1 These are new functions that are added to PMC–SB7.
2 These specifications are improved for PMC–SB7.
3 These are ignored like NOP if these are programmed in ladder. Therefore, you can use these functions to keep compatibility of your ladder program for several machines. You must certainly keep ACT=0 in case of some functions that need ACT input.
4 This is effective when you use PMC C language option.

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)

Instruc-tion	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).

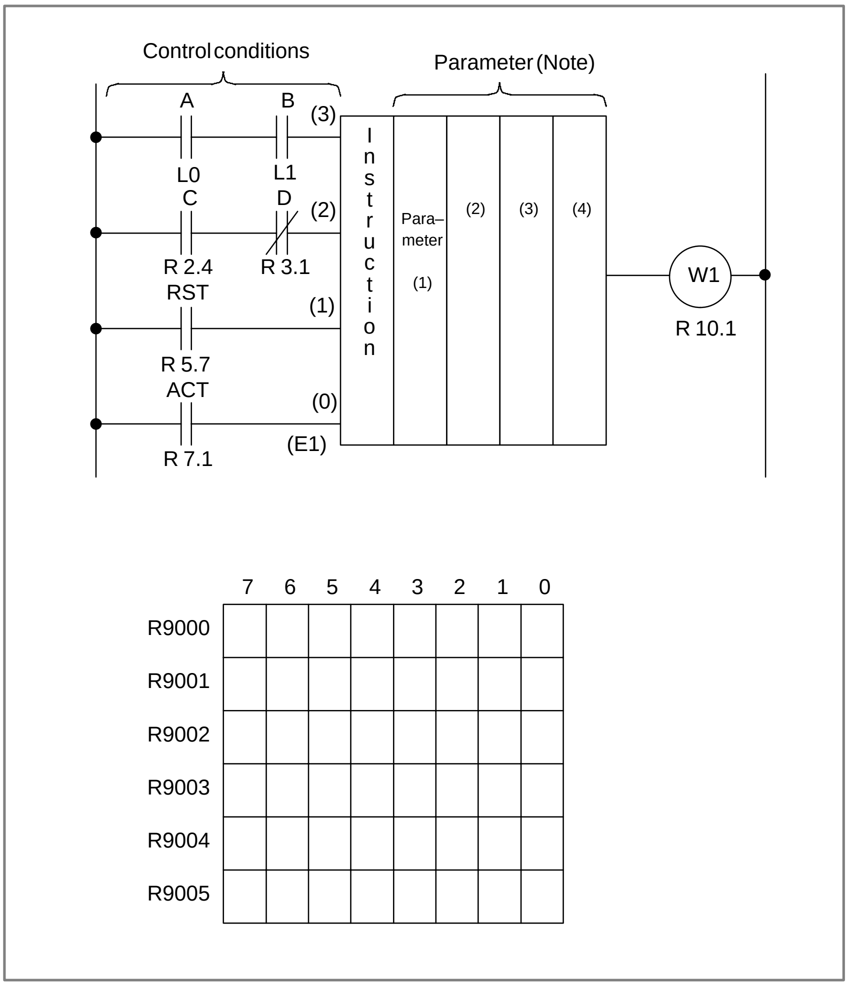


Fig. 5 (a) Function instruction format

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.

CAUTION

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.

	7	6	5	4	3	2	1	0
R250	0	0	1	1	0	1	0	0
	3				4			

	7	6	5	4	3	2	1	0
R251	0	0	0	1	0	0	1	0
	1				2			

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.

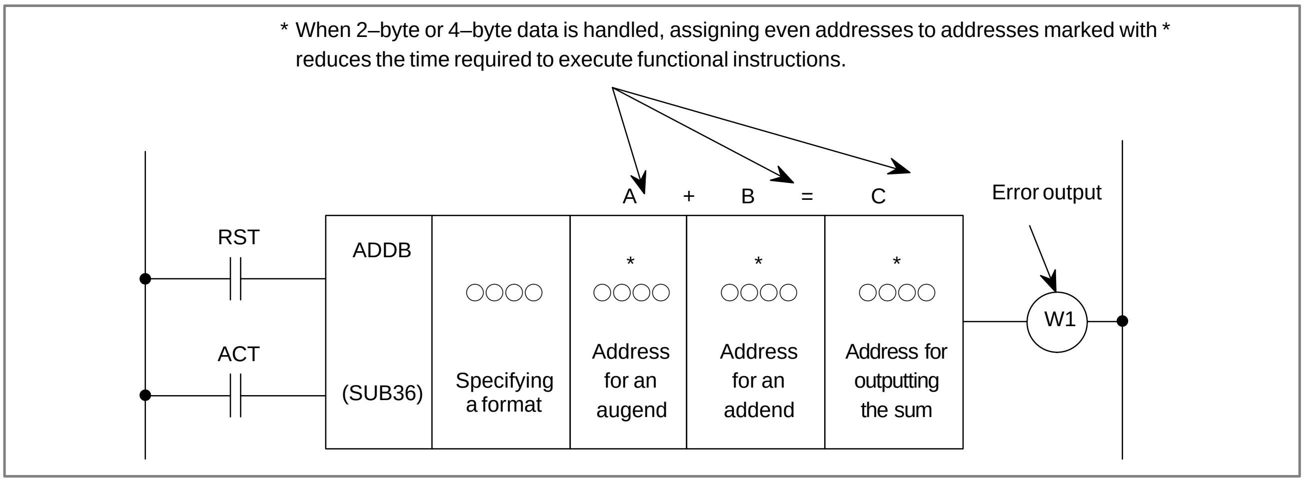


Fig. 5 (b)

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.

	7	6	5	4	3	2	1	0
R9000								
R9001								
R9002								
R9003								
R9004								
R9005								

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.

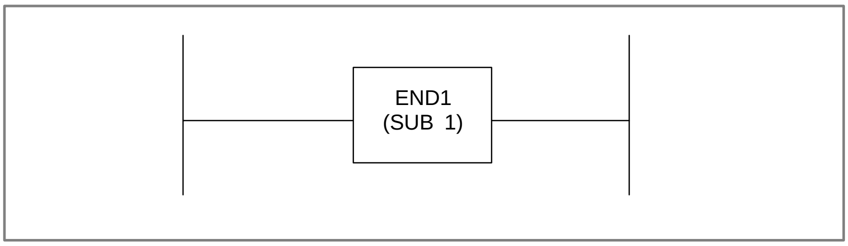


Fig. 5.1.2 Format of END.1

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.

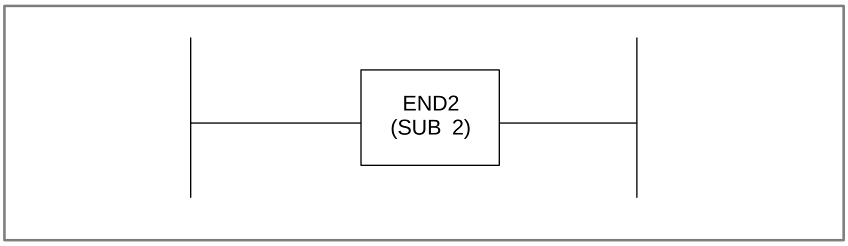


Fig. 5.2.2 Format of END.2

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/NB6/NB7 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.

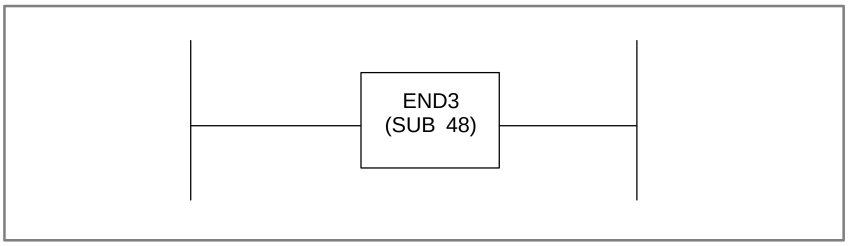


Fig. 5.3.2 END.3 description format

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.

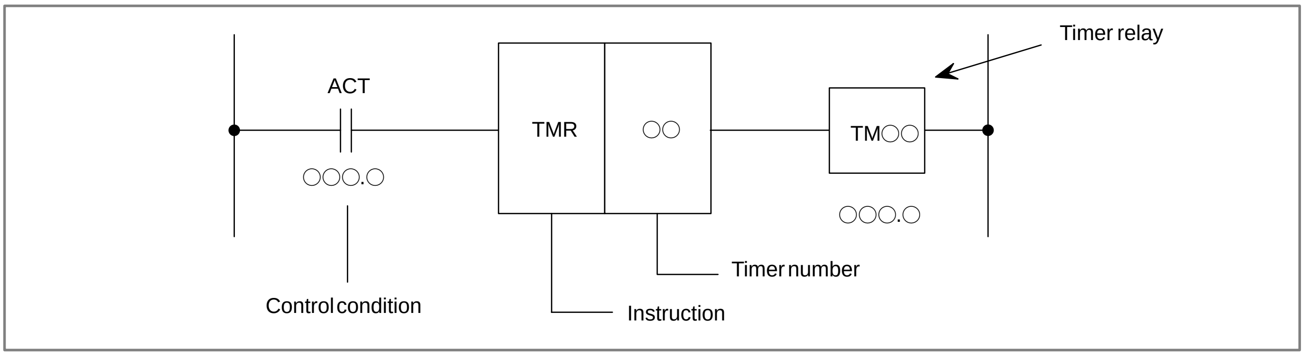


Fig. 5.4.4 (a) Format of TMR

Table 5.4.4 Coding of TMR

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

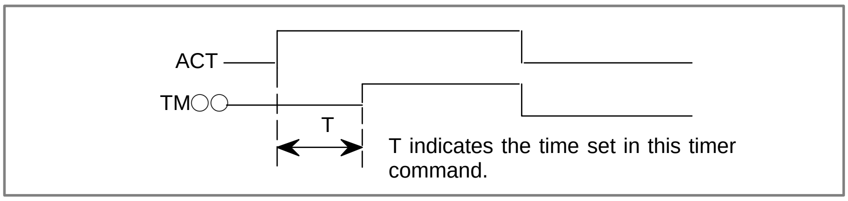


Fig. 5.4.4 (b) Operation of the timer

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	SB7	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	1 to 8
8 ms timer number	9 to 40	9 to 150	9 to 250	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	0 to +48 ms
8 ms timer	8 ms to 262.1 s	0 to +8 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).

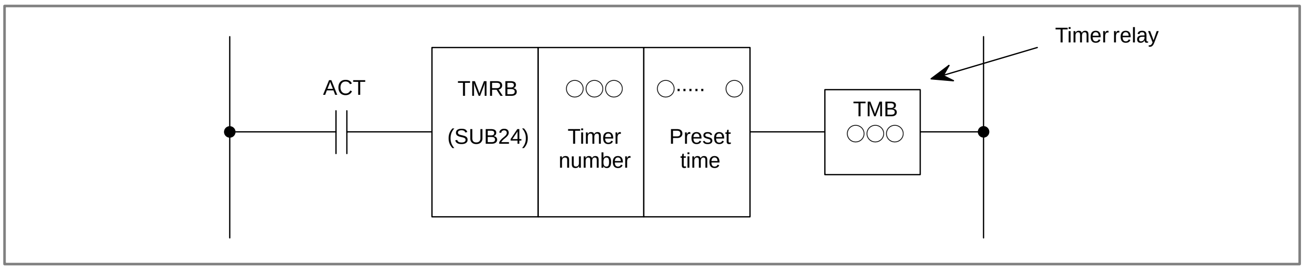


Fig. 5.5.2 Format of TMRB

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.

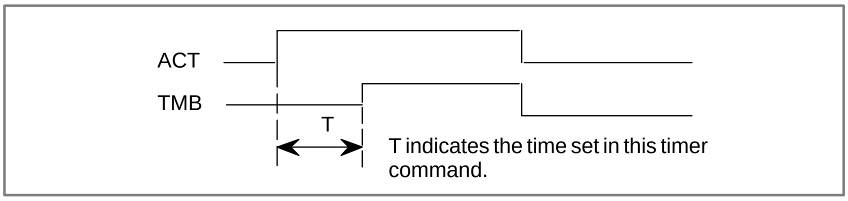


Fig. 5.5.4 Timer operation

5.5.5 Parameter

(a) Timer number
Sets timer number (1 to 100) of the fixed timers.
For the PMC–SB7, set a number between 1 and 500.

> **WARNING**
> If the same timer number is used more than once or if a timer number out of the valid range is used, operation is unpredictable.

(b) Preset time is 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.
For the PMC–SB7, preset time is 1 to 32,760,000 msec (about 546 minutes).

5.5.6 Precision of the Timer

Time varies 0 to level 1 sweep interval 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.

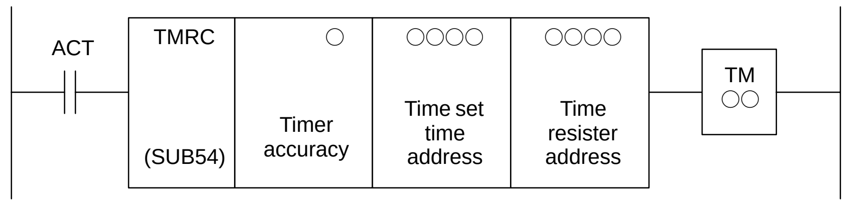


Fig. 5.6.2 TMRC expression format

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	1 to +8 ms
48 ms	1	1 to 1,572,816	1 to +48 ms
1 s (Note)	2	1 to 32,767	1 to +1 s
10 s (Note)	3	1 to 327,670	1 to +10 s
1 m (Note)	4	1 to 32,767	1 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

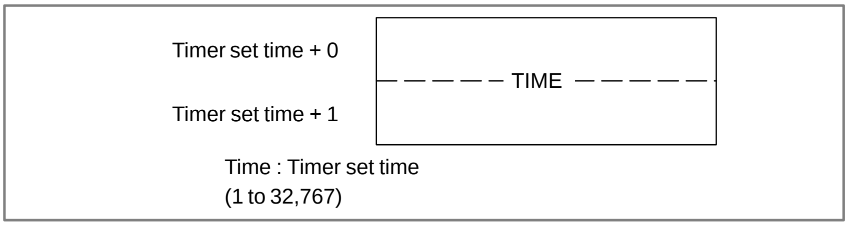
For PMC–SB7 :

Timer accuracy	Setting number	The range of setting time (Note)	Margin of error
8msec	0	8msec to about 262.1sec	0 to + Level 1 Sweep Interval
48msec	1	48msec to about 26.2 min	0 to + Level 1 Sweep Interval
1sec	2	1sec to about 546 min	0 to + Level 1 Sweep Interval
10sec	3	10sec to about 91 h	0 to + Level 1 Sweep Interval
1min	4	1min to about 546 h	0 to +1sec
1msec	5	1msec to about 32.7 sec	0 to + Level 1 Sweep Interval
10msec	6	10msec to about 327.7 sec	0 to + Level 1 Sweep Interval
100msec	7	100msec to about 54.6 min	0 to + Level 1 Sweep Interval

NOTE
The range of the value is 0 to 32767.

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.

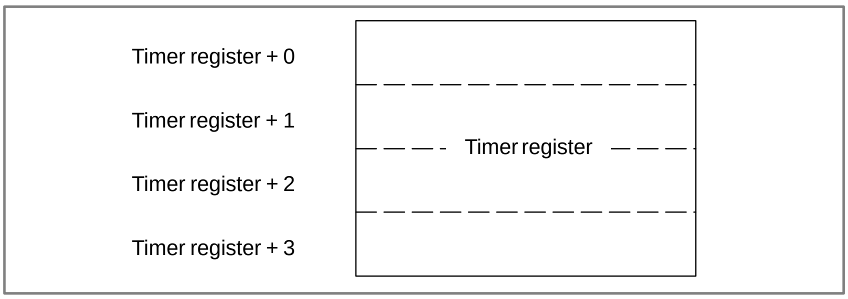


The timer set time is converted into the binary value in 8 ms (48 ms) units.
The timer set time is shown as follows:

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.

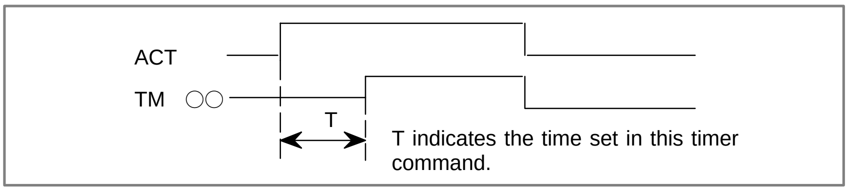


Fig. 5.6.7 Timer operation

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.

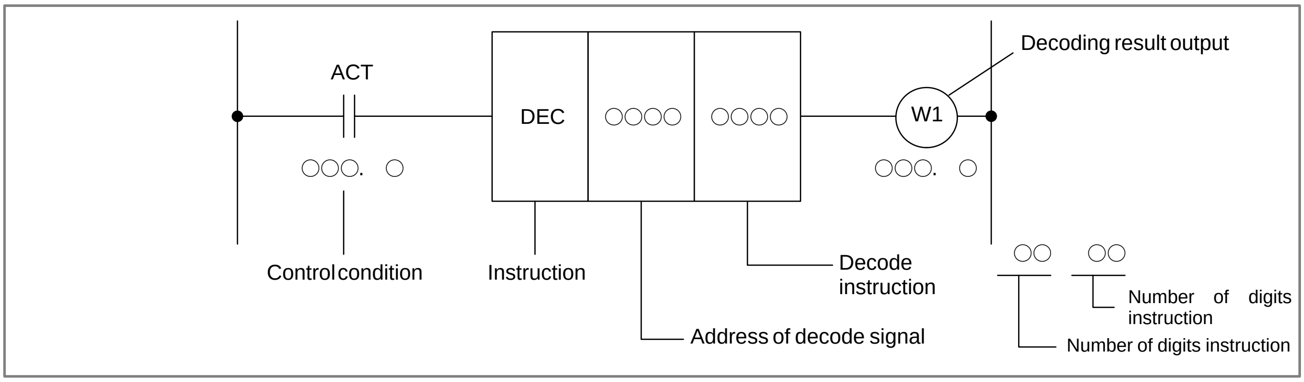


Fig. 5.7.2 Format of DEC

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

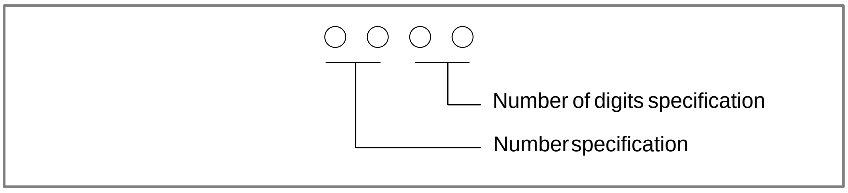
ACT=0 : Turns the decoding result output off (W1).
ACT=1 : Performs decoding.
When the specified number is equal to the code signal, W1=1; when not, W1=0.

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.

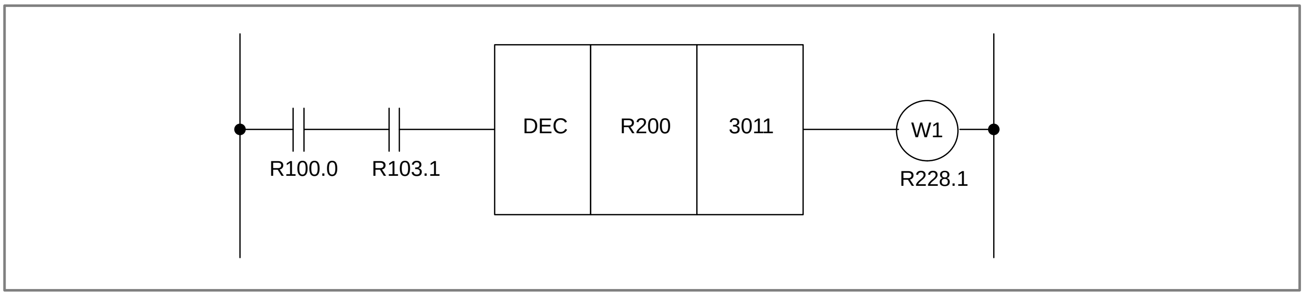


Fig. 5.7.6 Ladder diagram using the DEC instruction

Table 5.7.6 Coding for Fig.5.7.6
Coding sheet

Step Number	Instruction	Address Number	Bit Number	Remarks
1	RD	R100.0		
2	AND	R103.1		
3	DEC	R200		
4	(PRM)	3011		
5	WRT	R228 .	1	

5.8 DECB (BINARY DECODING)

5.8.1 Function

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/SB7 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 ($8 \times n$) bytes.
For the details of the setting of a format specification parameter, refer to "5.8.4 Parameters".

5.8.2 Format

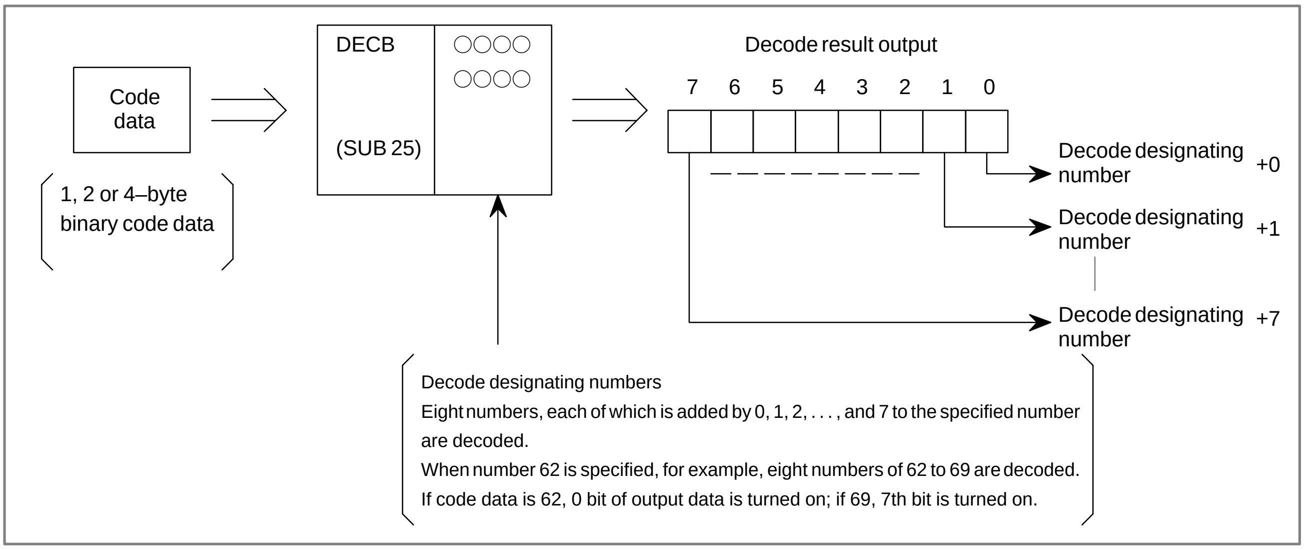


Fig. 5.8.2 (a) Function of DECB (basic specification)

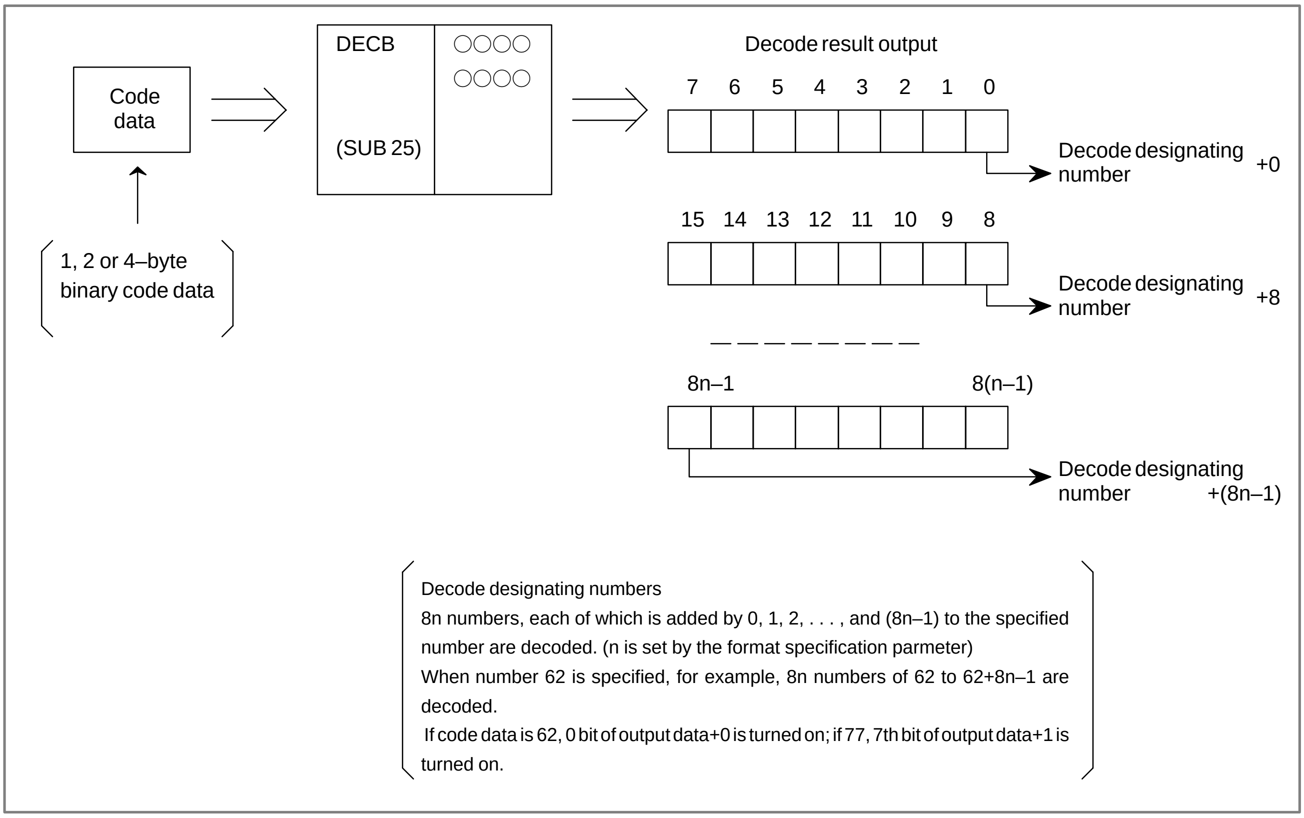


Fig. 5.8.2 (b) Function of DECB (extended specification)
(only for PMC–SB5/SB6/SB7 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i*, PMC–SA5 for Series 21*i*/210*i*, and PMC–NB6 for Series 15*i*)

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

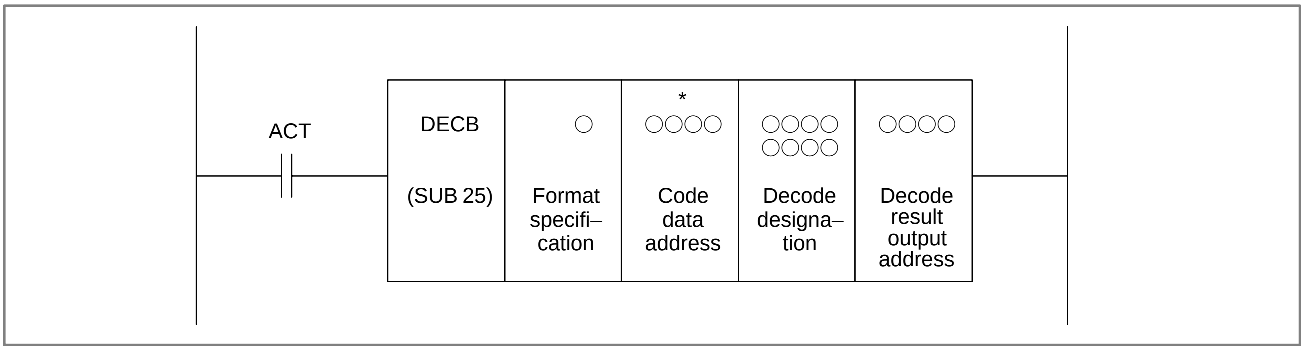


Fig. 5.8.2 (c) Expression format of DECB (basic specification)

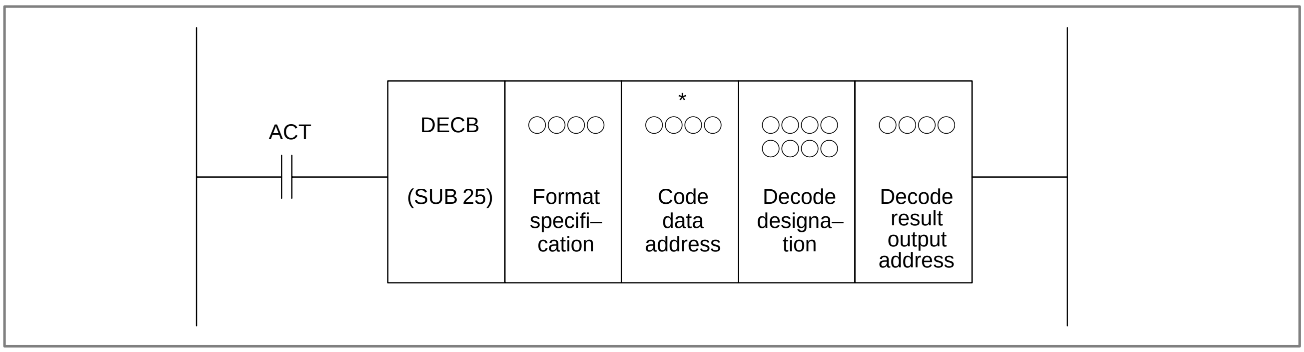


Fig. 5.8.2 (d) Expression format of DECB (extended specification)
(only for PMC–SB5/SB6/SB7 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i*, PMC–SA5/SB6/SB7 for Series 21*i*/210*i*, and PMC–NB6 for Series 15*i*)

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/SB7 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 × n) bytes by 1 instruction.

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

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

0nn4 : In case of decoding multiple (8 × 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/SB7 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–NB6 for Series 15*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.

WARNING
When a incollect BCD data was set to a BCD type counter, the morement of CTR cannot be sured.
If changing the counter type, be sure to reconfigure the counter data.

This counter has the following functions to meet various applications.

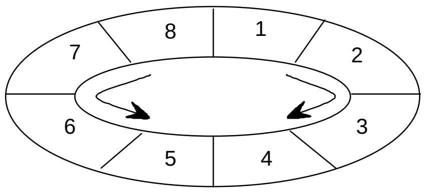
(a) Preset counter
Outputs a signal when the preset count is reached. The number can be preset from the CRT/MDI panel, or set in the sequence program.

(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.



Presetting : 8
Initial value : 1

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.

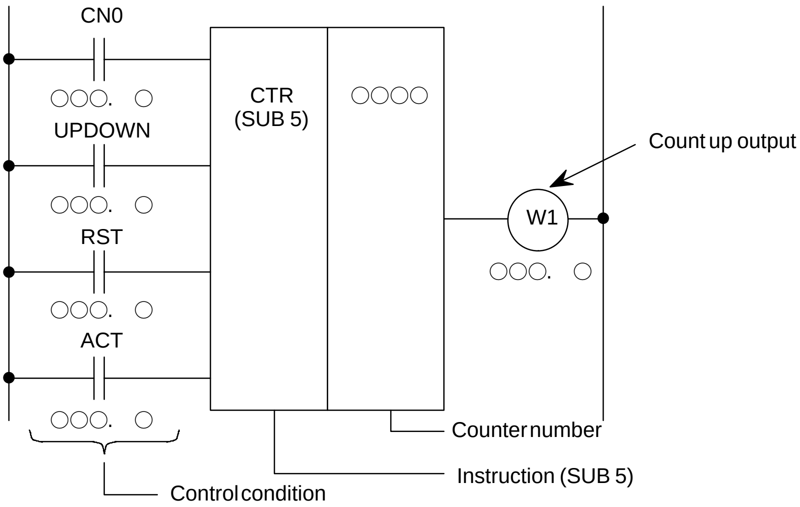


Fig. 5.9.2 Format of CRT instruction

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

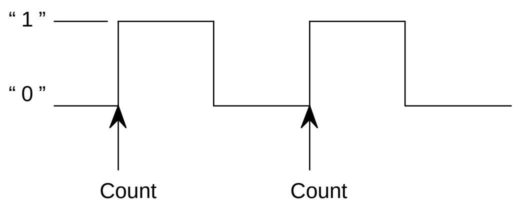
(a) Specify the initial value. (CN0)
CN0=0: Begins the value of the counter with 0.
0, 1, 2, 3 ····· n.
CN0=1: Begins the value of the counter with 1 (0 is not used).
1, 2, 3 ····· n.

(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

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	SB7	SC	SC3	SC4	NB	NB2	NB6
Counter number	1 to 20	1 to 50	1 to 100	1 to 20	1 to 20	1 to 50	1 to 20	1 to 50	1 to 50

The preset value and cumulative value that can be set are as follows:
Binary counter: 0 to 32767
BCD counter: 0 to 9999

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.

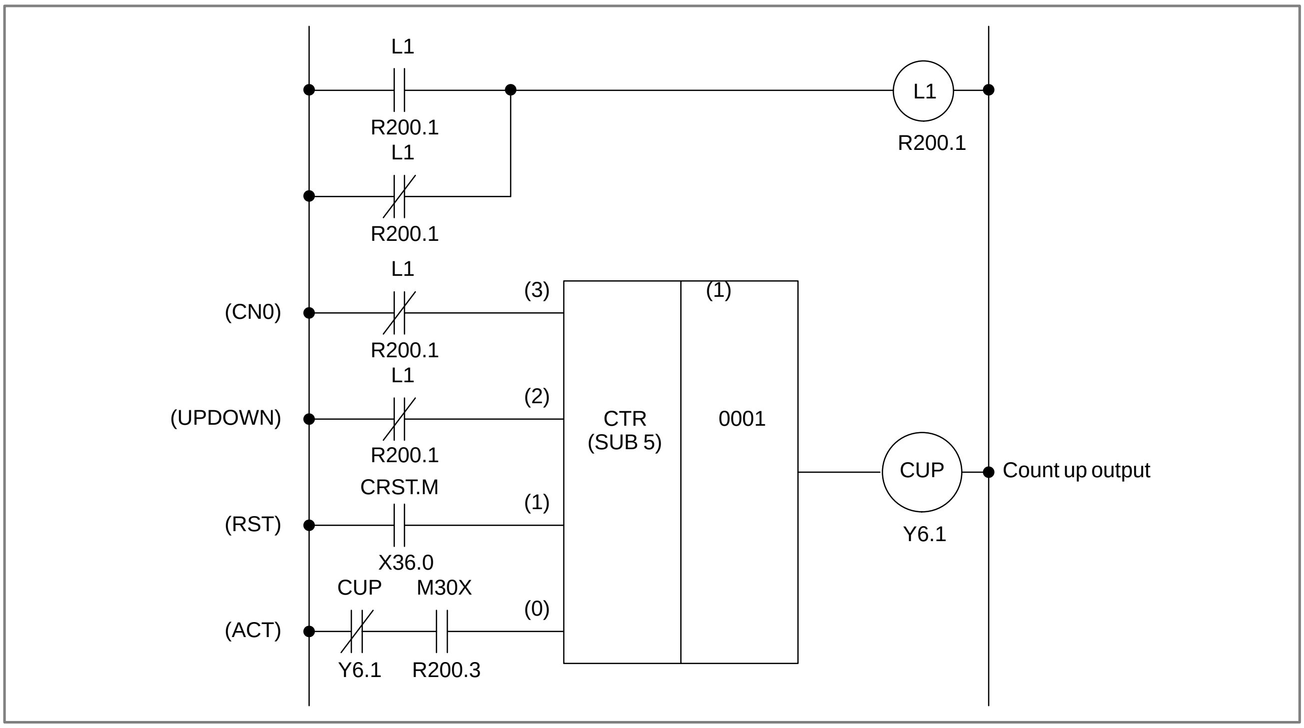


Fig. 5.9.6 (a) Ladder diagram for the counter, example 1

[Example 2]

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

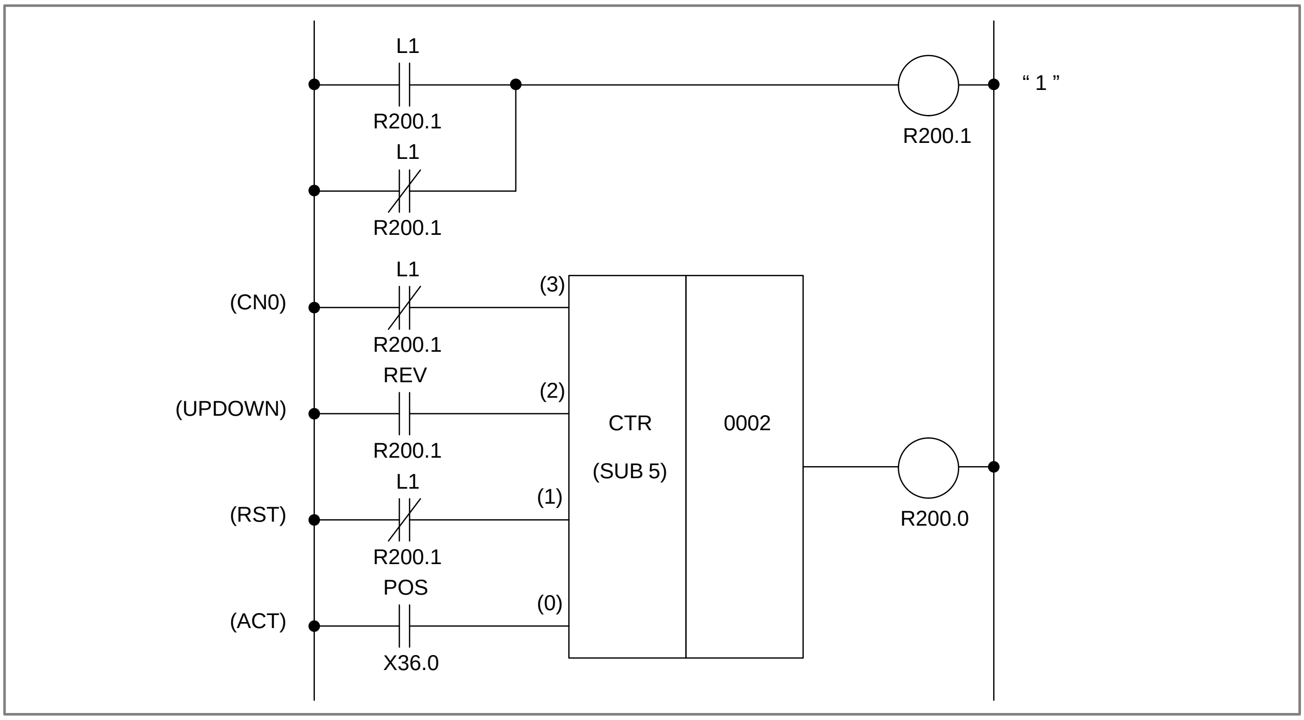


Fig. 5.9.6 (b) Ladder diagram for the counter, example 2

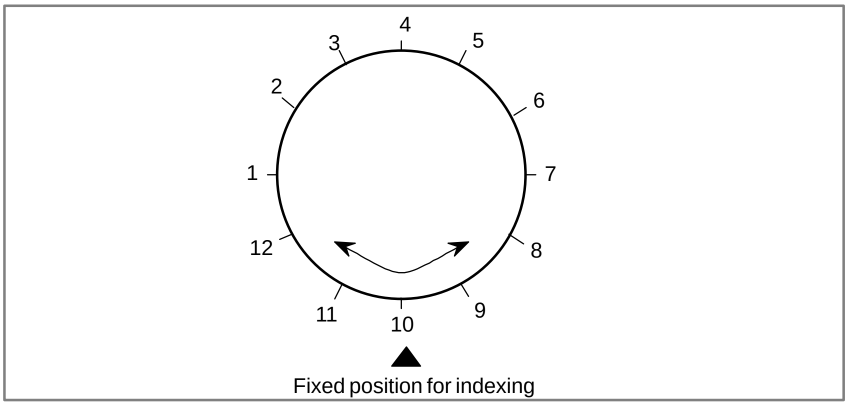


Fig. 5.9.6 (c) Indexing for a rotor

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 CTRB (FIXED COUNTER)

PMC–SB7
O

5.10.1 Functions

CTRB is used as a counter. Numerical data such as preset values and count values can be used with binary format. This counter has the following functions to meet various applications.

(a) Preset counter
Preset the count value. 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 up counter and down counter.

(d) Selection of 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 cording format, respectively.

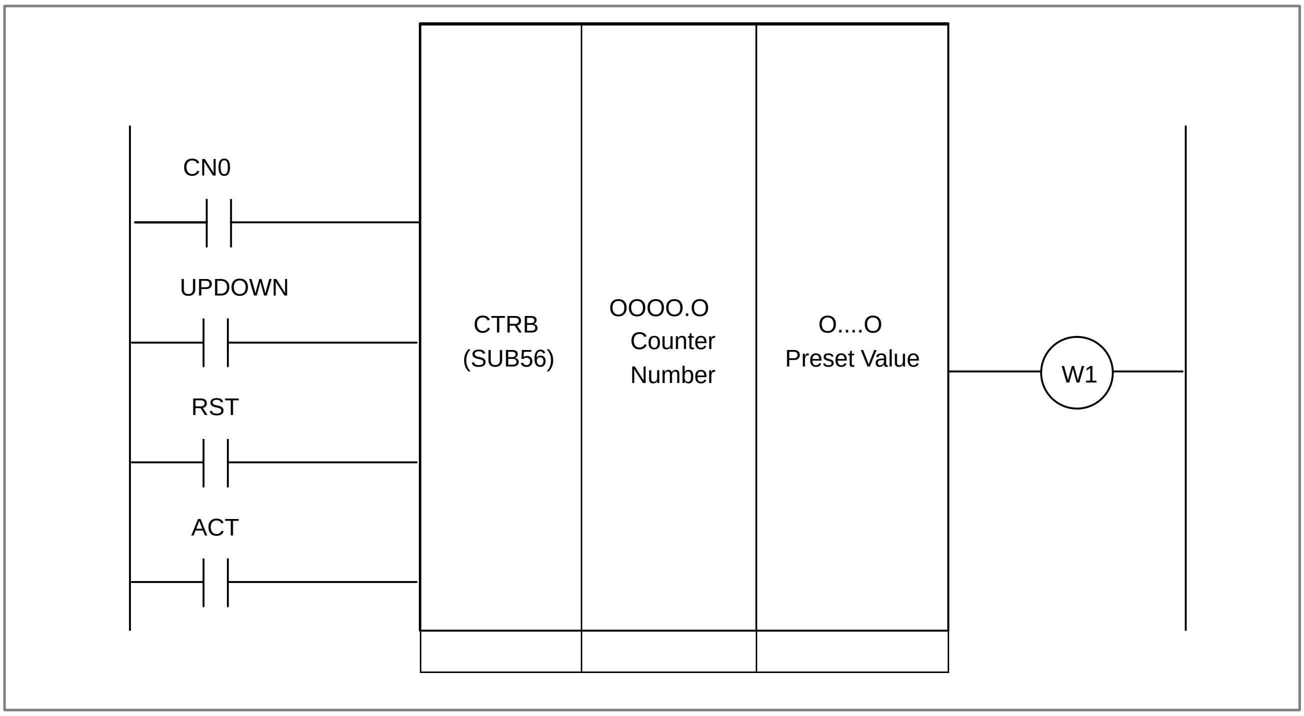


Fig. 5.10.2 CTRB expression format

Table 5.10.2 CTRB cording format

Step number	Instruction	Address number	Bit number	Remarks
1	RD	OOOO.O		CN0
2	RD.STK	OOOO.O		UPDOWN
3	RD.STK	OOOO.O		RST
4	RD.STK	OOOO.O		ACT
5	SUB	56		CTRB command
6	(PRM)	OOO		Counter Number
7	(PRM)	O...O		Preset Value
8	WRT	OOOO.O		W1

5.10.3 Control Condition

(a) Specifying the initial value (CN0)
CN0=0 : The counter value starts with ”0”. 0,1,2,3,.......,n
CN0=1 : The counter value starts with ”1”. 1,2,3,4.........,n

(b) Specifying up or down (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 : Cancels reset.
RST=1 : Resets. W1 is reset to 0. The accumulated value is reset to the initial value.

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

(3) Counter number
Set 1 to 100 as counter number.

(4) Preset value
Following value can be set as preset value.
Binary counter: 0 to 32767

*CTRB is always binary counter. System parameter is ineffective.

(5) Count–up output (W1)
When the counter value reaches the preset value, W1 is set to 1. The W1 address can be specified arbitrarily.

(6) Accumulate value
The address C5000– are used for accumulate value of the CTRB. One value has 2 bytes.
Counter number 1 corresponds to C5000 and number 2 corresponds to C5002.

5.11 CTRC (COUNTER)

5.11.1 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.

5.11.2 Format

Fig.5.11.2 and Table 5.11.2 show the expression format and the coding format, respectively.

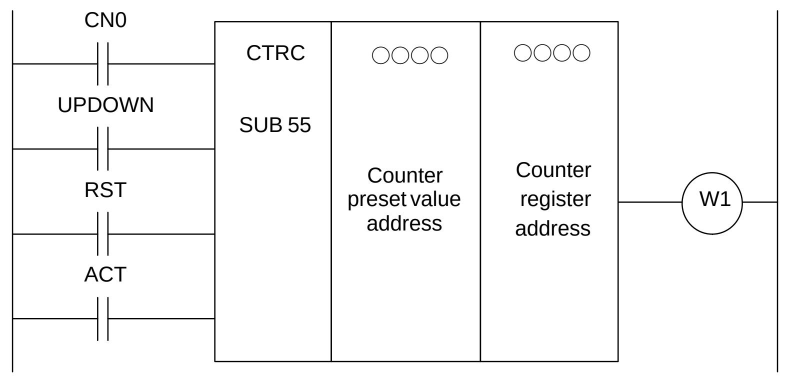


Fig. 5.11.2 CTRC expression format

Table 5.11.2 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

5.11.3 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

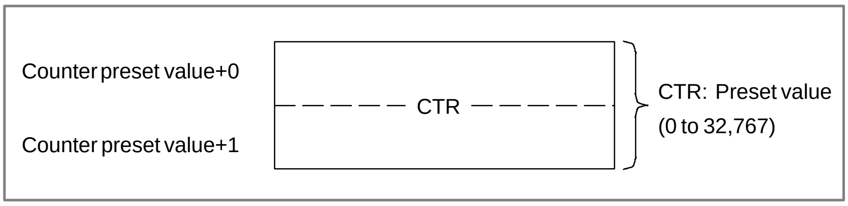
(b) Specifying up or down count (UPDOWN)
UPDOWN=0:
Up counter.
The initial value is “0” when CN0=0 or “1” when CN0=1.
UPDOWN=1:
Down counter. The initial value is the preset value.

(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.

5.11.4 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.

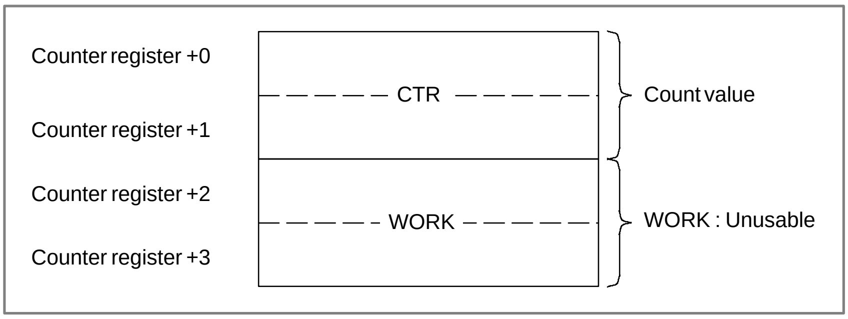
5.11.5 Counter Register Address

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.

5.11.6 Count–up Output (W1)

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



5.12 ROT (ROTATION CONTROL)

5.12.1 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

5.12.2 Format

Fig.5.12.2 shows the expression format and Table 5.12.2 shows the coding format.

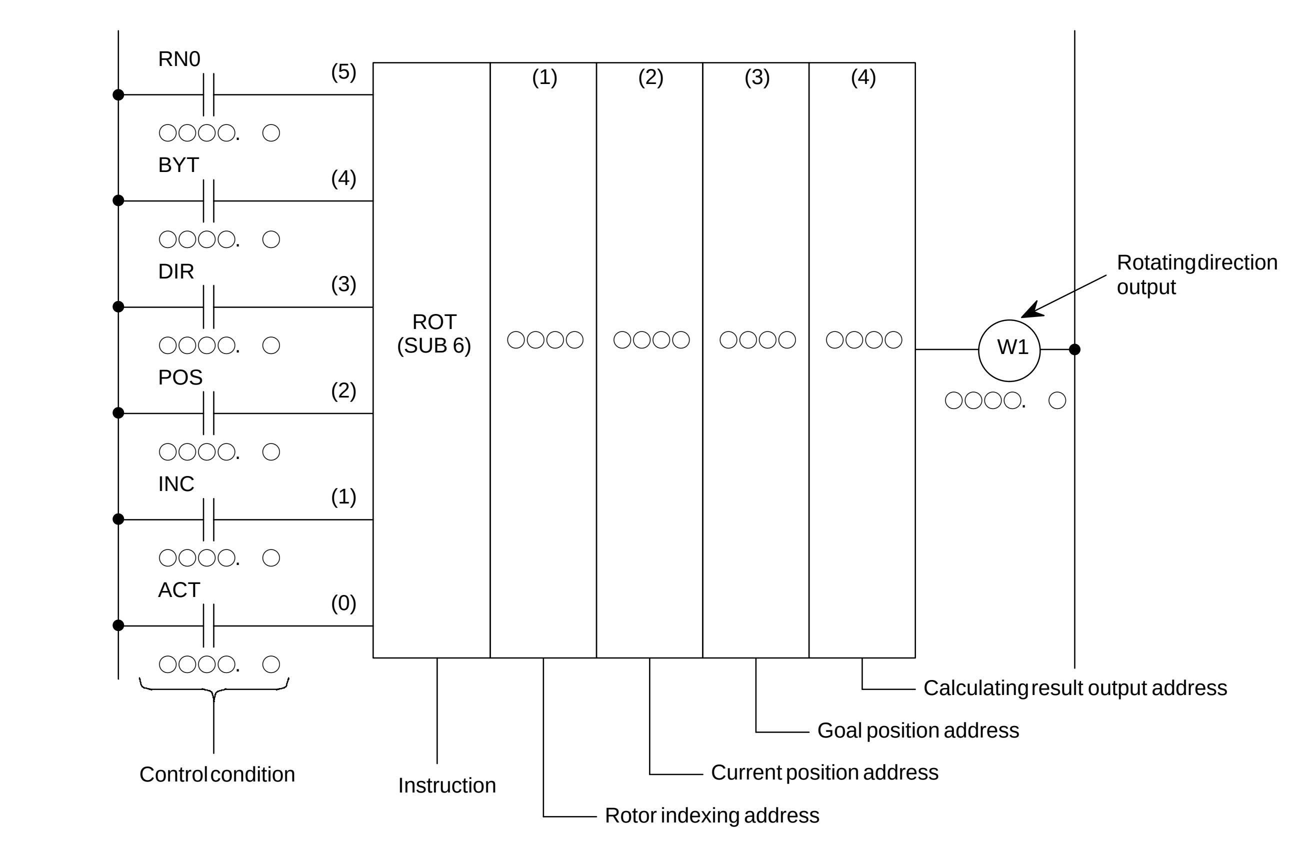


Fig. 5.12.2 ROT instruction format

Table 5.12.2 Coding for Fig.5.12.2

Coding sheet / Status of operating result

Step Number	Instruction	Address No.	Bit No.	Remarks	ST5	ST4	ST3	ST2	ST1	ST0
1	RD	○○○○ .	○	RN0						RN0
2	RD. STK	○○○○ .	○	BYT					RN0	BYT
3	RD. STK	○○○○ .	○	DIR				RN0	BYT	DIR
4	RD. STK	○○○○ .	○	POS			RN0	BYT	DIR	POS
5	RD. STK	○○○○ .	○	INC		RN0	BYT	DIR	POS	INC
6	RD. STK	○○○○ .	○	ACT	RN0	BYT	DIR	POS	INC	ACT
7	SUB	6		ROT	RN0	BYT	DIR	POS	INC	ACT
8	(PRM)	○○○○		Rotor indexing number	RN0	BYT	DIR	POS	INC	ACT
9	(PRM)	○○○○		Current position	RN0	BYT	DIR	POS	INC	ACT
10	(PRM)	○○○○		Goal position address	RN0	BYT	DIR	POS	INC	ACT
11	(PRM)	○○○○		Calculating result output address	RN0	BYT	DIR	POS	INC	ACT
12	WRT	○○○ .	○		RN0	BYT	DIR	POS	INC	W1
13										
14										
15										

5.12.3 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.

5.12.4 Rotor Indexing Number

Specify the rotor indexing number.

5.12.5 Current Position Address

Specify the address storing the current position.

5.12.6 Goal Position Address

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

5.12.7 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.

5.12.8 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.5.12.8. 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.

An example of a 12–position rotor

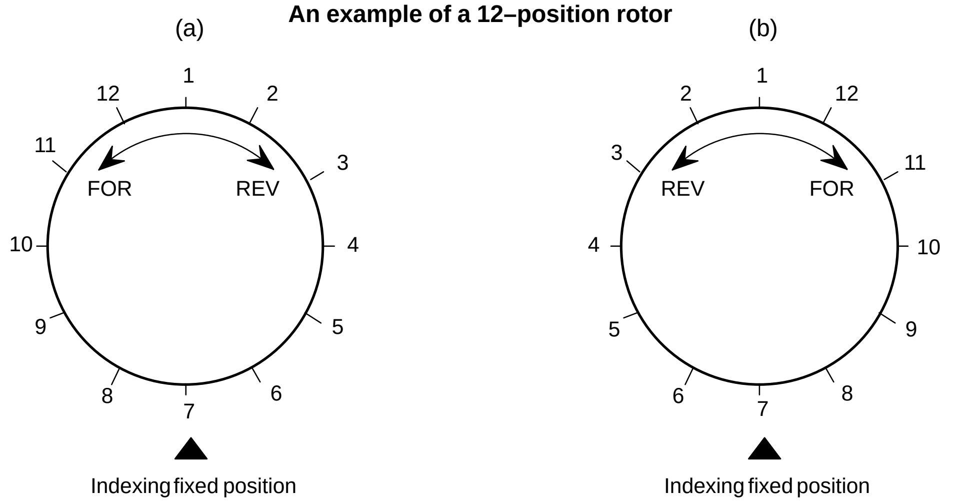


Fig. 5.12.8 Rotation direction

5.13 ROTB (BINARY ROTATION CONTROL)

5.13.1 Function

This instruction is used to control rotating elements including the tool post, ATC (Automatic Tool Changer), rotary table, etc. In the ROT command (5.12) 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.

5.13.2 Format

Fig.5.13.2 shows the expression format of ROTB

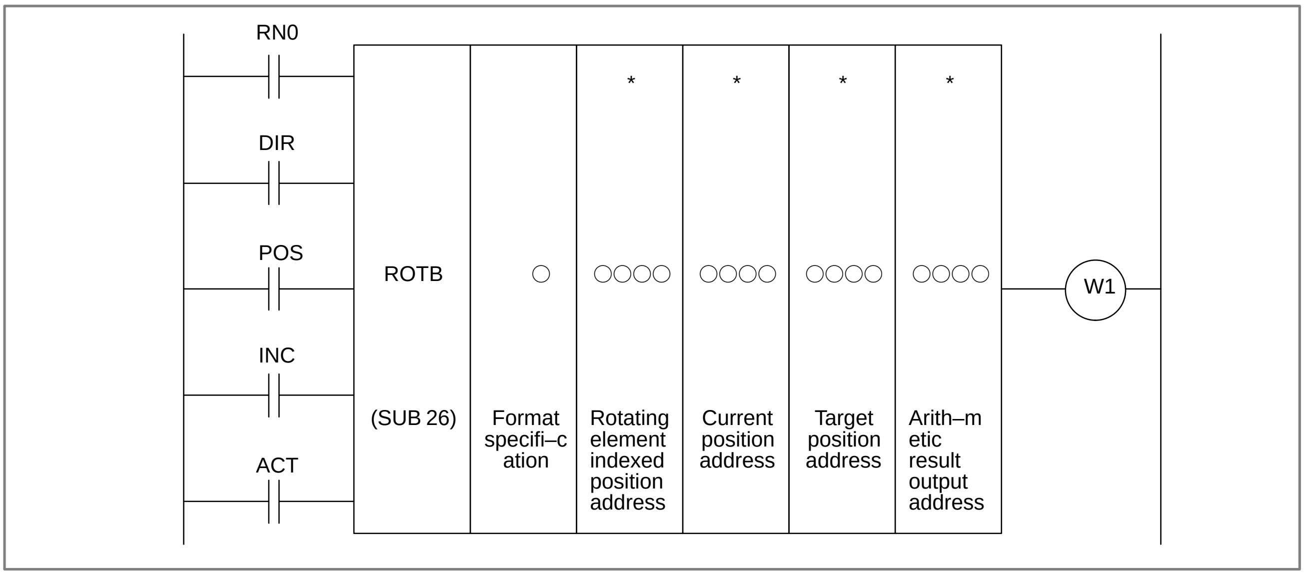


Fig. 5.13.2 Expression format of ROTB

5.13.3 Control Conditions

The control conditions do not differ basically from those for ROT command described in section 5.12. However, BYT has been eliminated from ROTB (it forms part of the ROTB parameters).
For the reset, see ROT.

5.13.4 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 5.12.

5.13.5 Output for Rotational Direction (W1)

See Section 5.12.

5.13.6 Example of Using the ROTB Instruction

Fig. 5.13 (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.

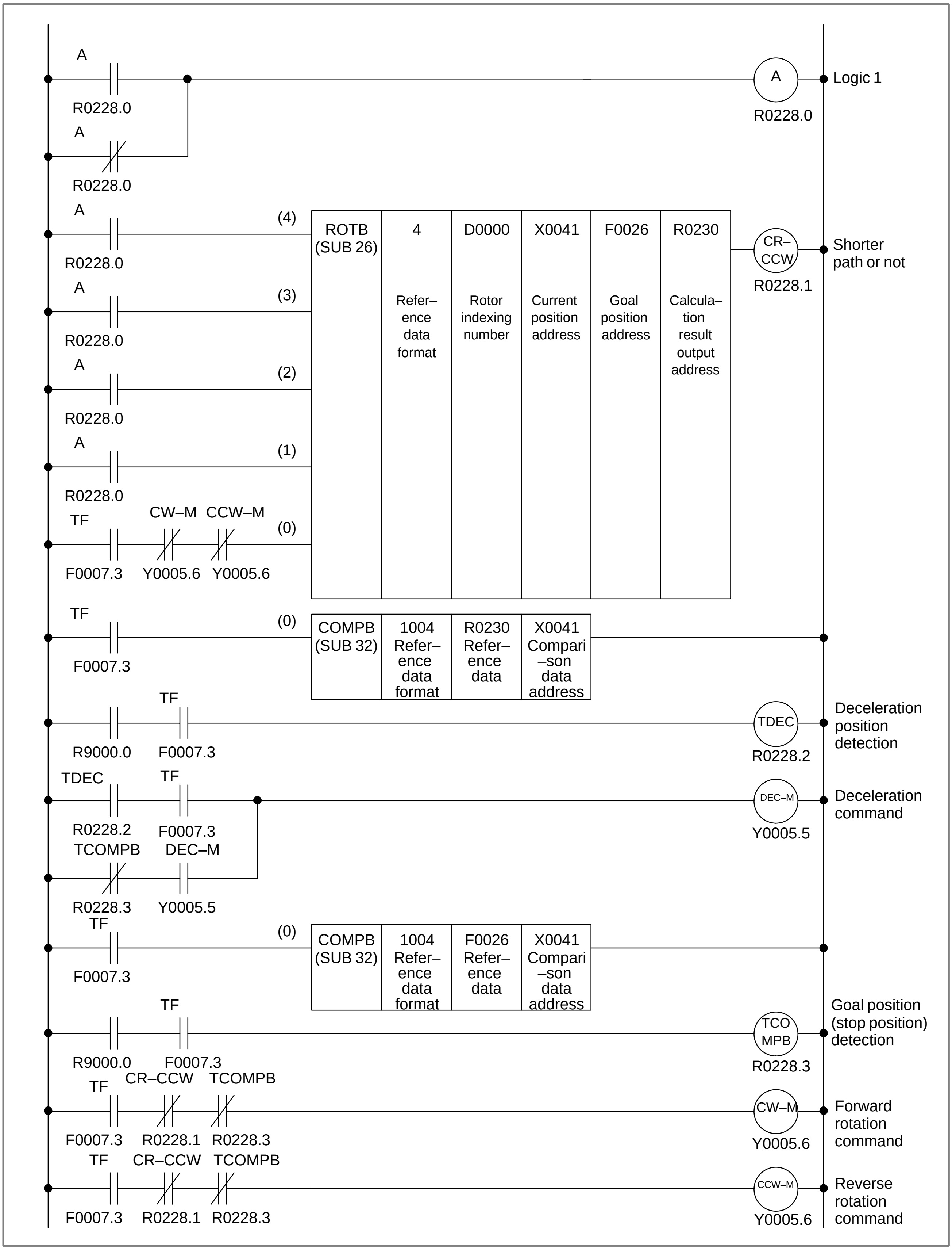


Fig. 5.13.6 Example of a ladder diagram for the ROTB instruction

5.14 COD (CODE CONVERSION)

5.14.1 Function

Converts BCD codes into an arbitrary two– or four–digits BCD numbers. For code conversion shown in Fig.5.14.1 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.5.14.1, 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.

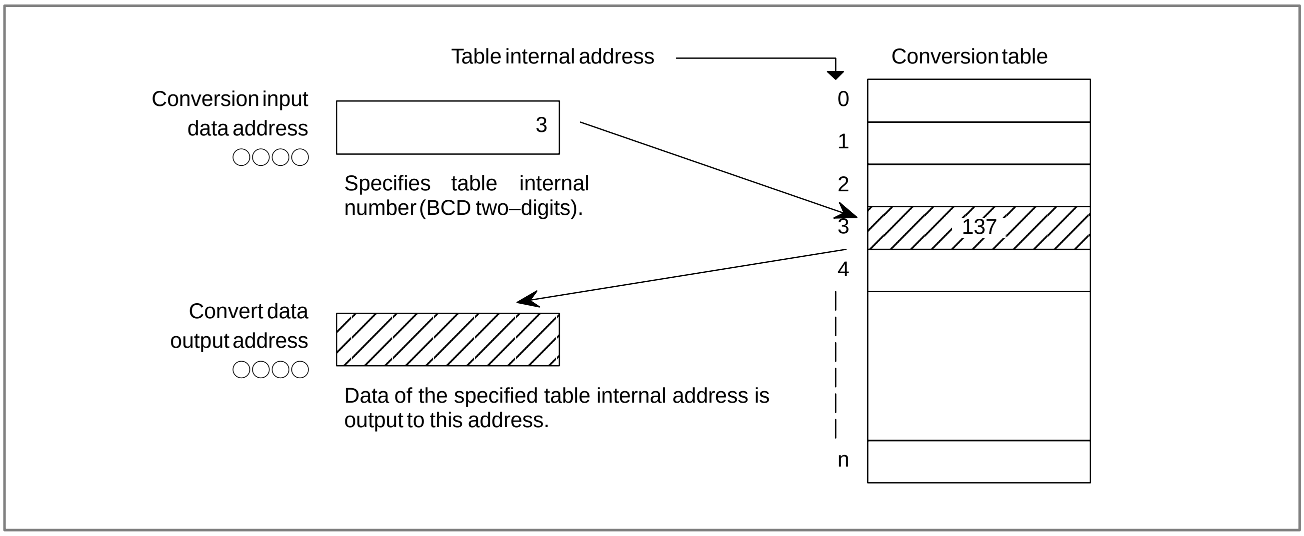


Fig. 5.14.1 Code conversion diagram

5.14.2 Format

Fig.5.14.2 shows the format for the COD instruction and Table 5.14.2 shows the coding format.

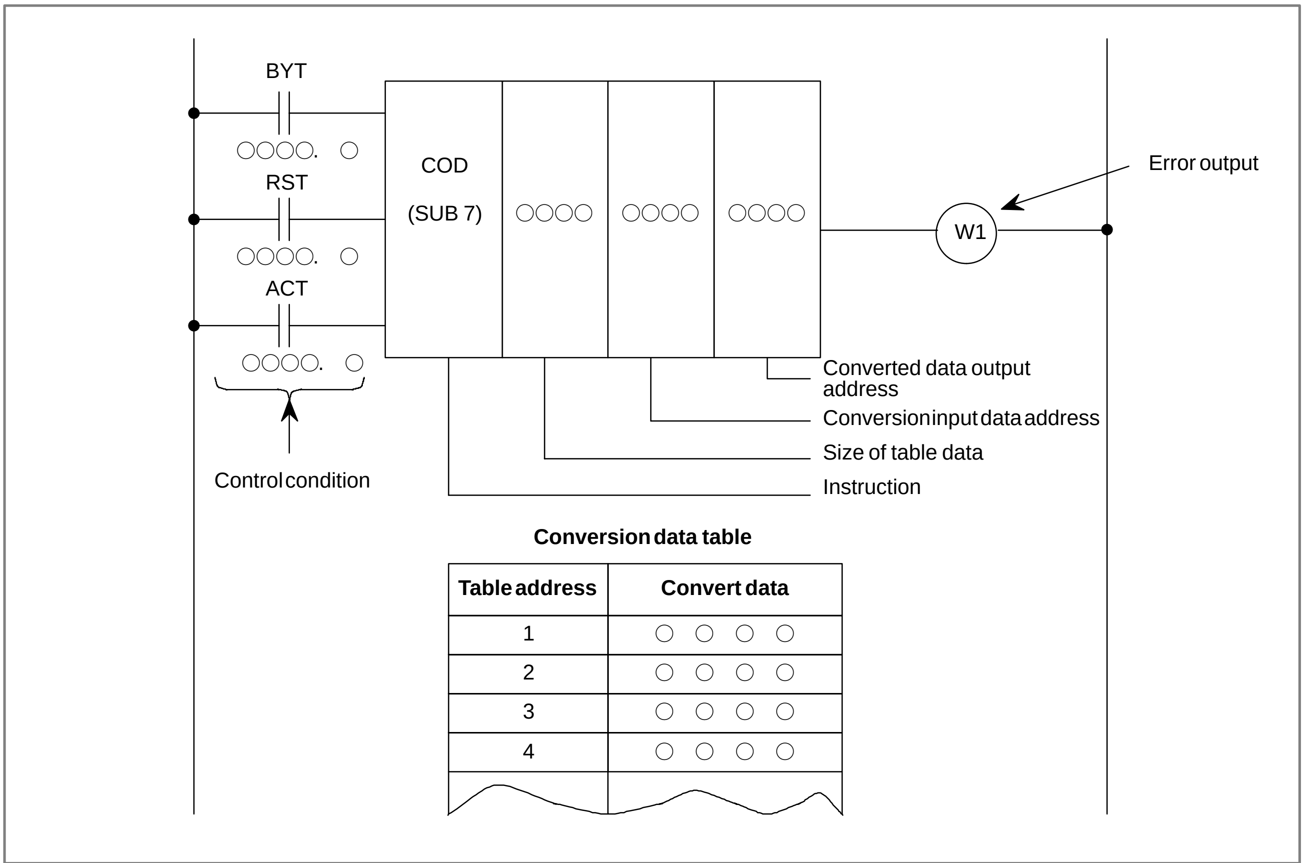


Fig. 5.14.2 COD instruction

Table 5.14.2 Coding for Fig.5.14.2

Coding sheet

Step Number	Instruction	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

5.14.3 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.

5.14.4 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.

5.14.5 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.

5.14.6 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.

5.14.7 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.

5.14.8 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

5.15 CODB (BINARY CODE CONVERSION)

5.15.1 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.5.15.1.
Compared to the 5.14 ”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.

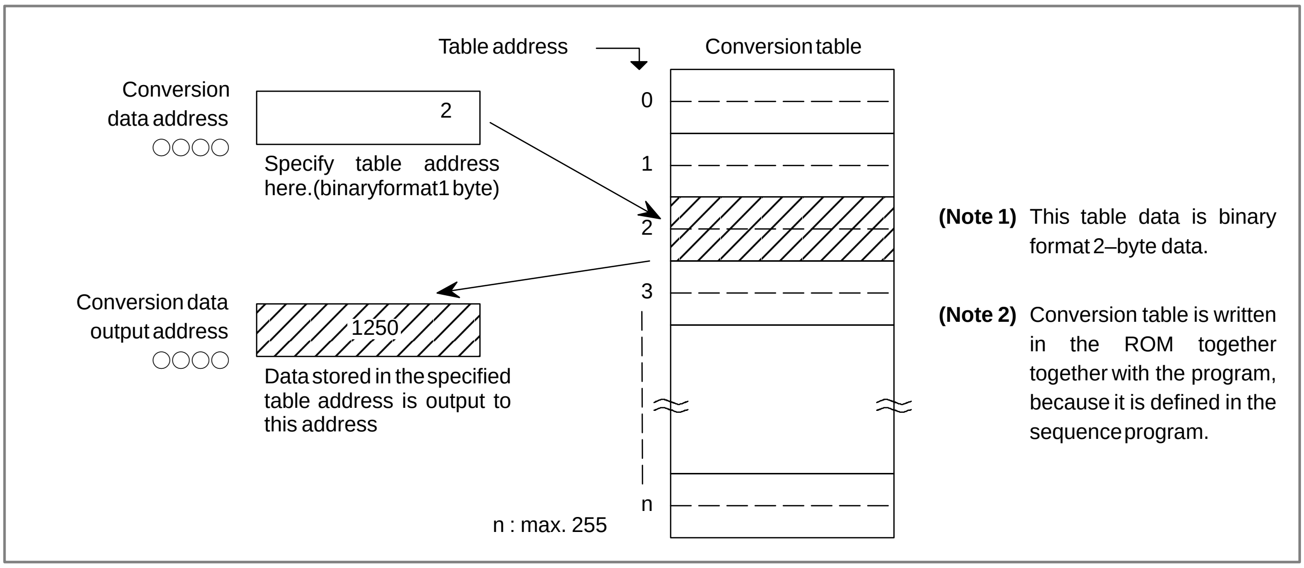


Fig. 5.15.1 Code conversion diagram

5.15.2 Format

Fig.5.15.2 shows the expression format of CODB.

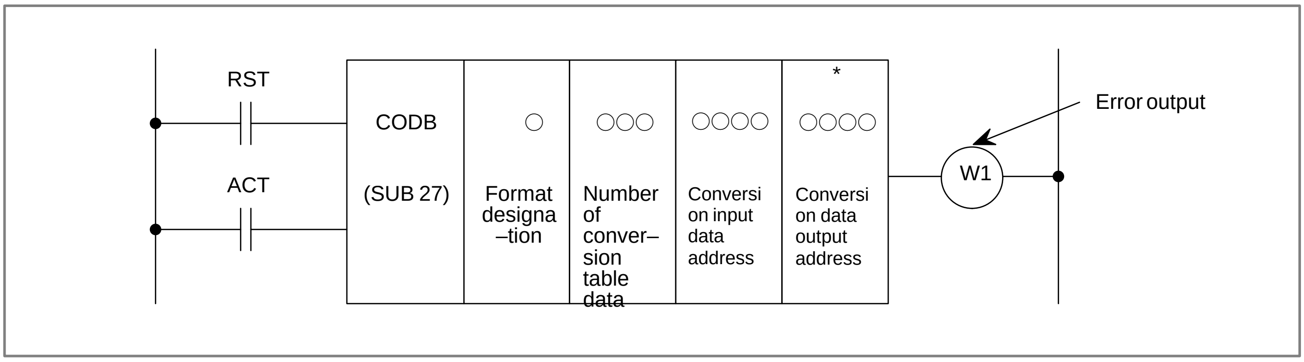


Fig. 5.15.2 Expression format of CODB

5.15.3 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.

5.15.4 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.

5.15.5 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).

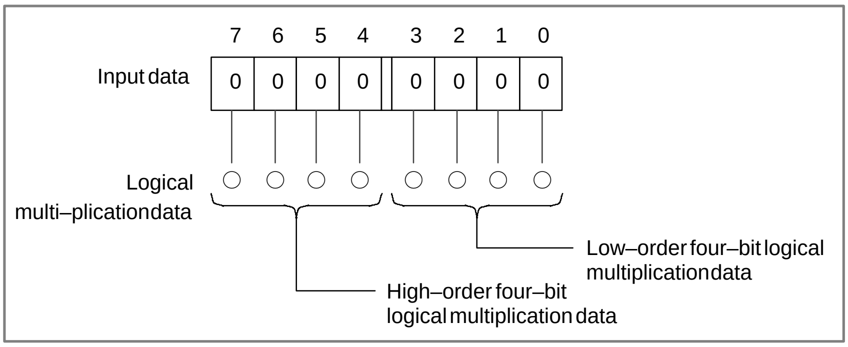
5.15.6 Error Output (W1)

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

5.16 MOVE (LOGICAL PRODUCT TRANSFER)

5.16.1 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).



5.16.2 Format

Fig.5.16.2 shows the expression format and Table 5.16.2 shows the coding format.

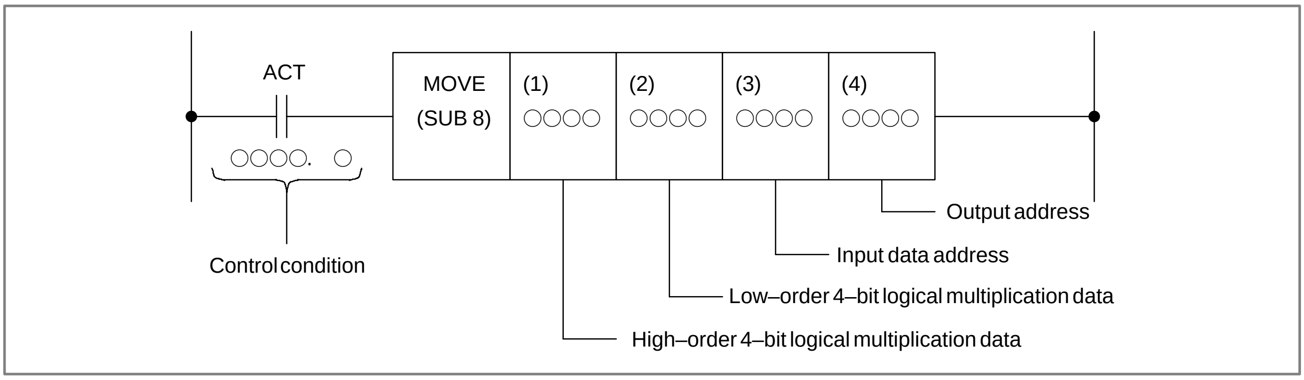


Fig. 5.16.2 Move instruction format

Table 5.16.2 Coding for Fig.5.16.2

Coding sheet

Step Number	Instruction	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

5.16.3 Execution Command

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

5.16.4 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.

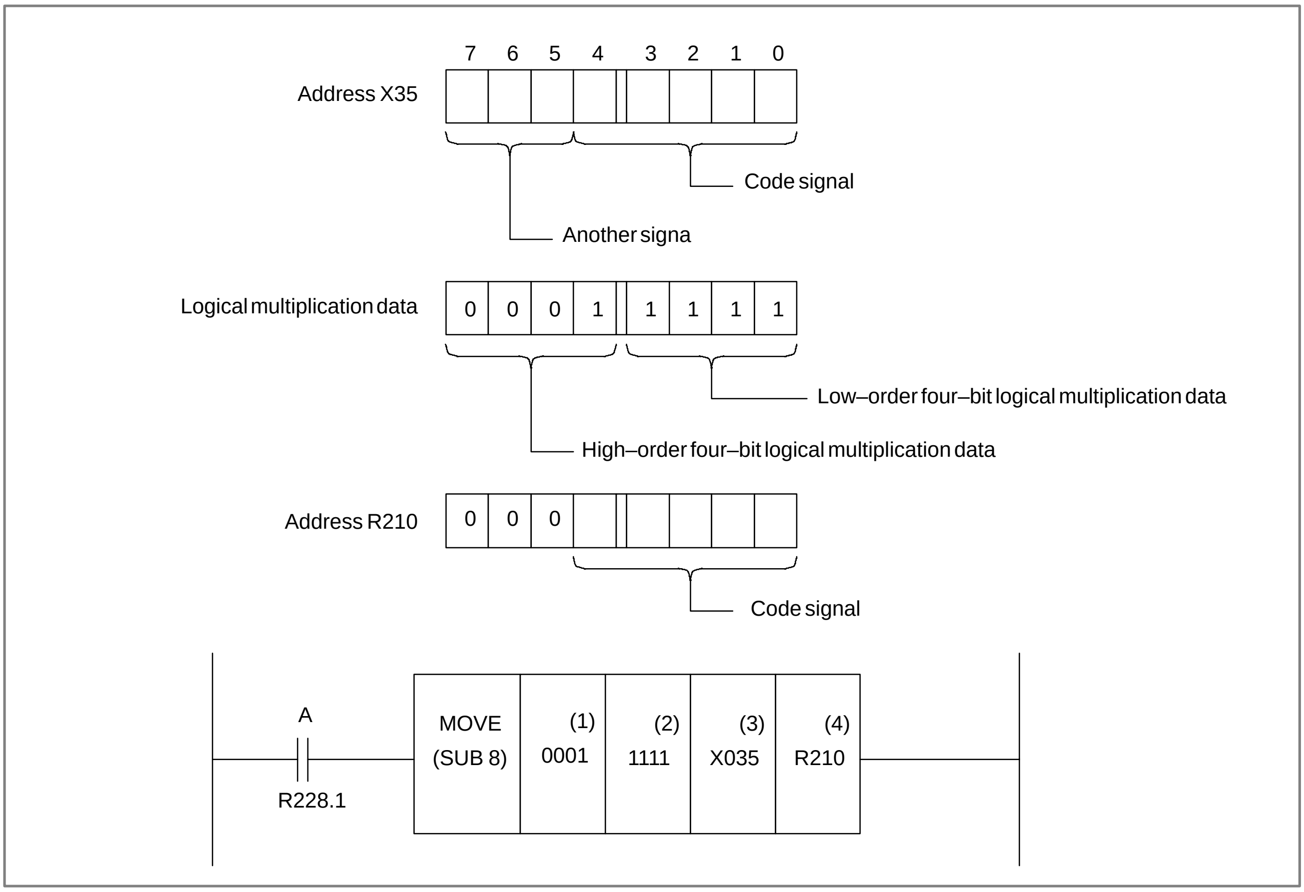
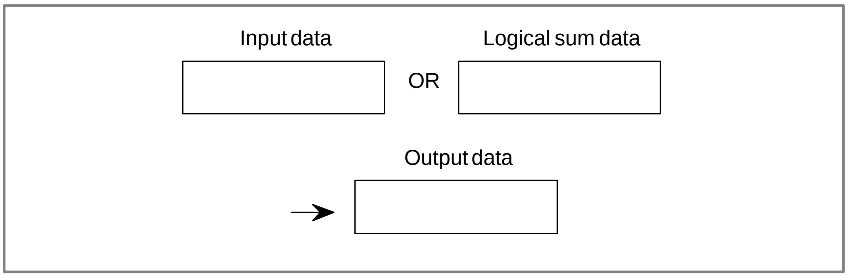


Fig. 5.16.4 MOVE instruction ladder diagram

5.17 MOVOR (DATA TRANSFER AFTER LOGICAL SUM)

5.17.1 Function

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



5.17.2 Format

Fig.5.17.2 shows the expression format of MOVOR.

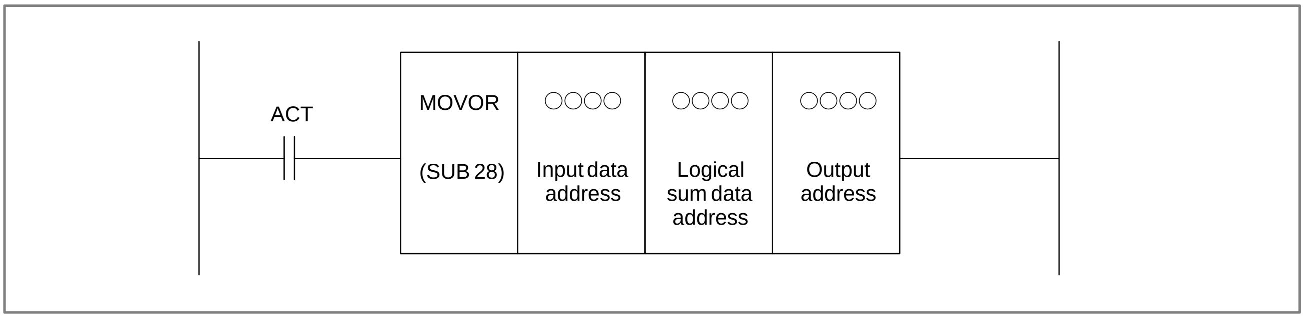


Fig. 5.17.2 Expression format of MOVOR

5.17.3 Control Conditions

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

5.17.4 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.

5.18 COM (COMMON LINE CONTROL)

5.18.1 COM (Common Line Control)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	×	×	×	×	×	○	×	×	×	×	×	×	○	×	×	×	×	×

This function can be used for specifying the number of coils only on the PMC–SB/SC. On other PMCs, specify 0 for the number of coils and use the common line control end instruction to use this function. For how to use the instruction, see Subsection 5.18.4.

5.18.1.1 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.5.18.1.1)
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.

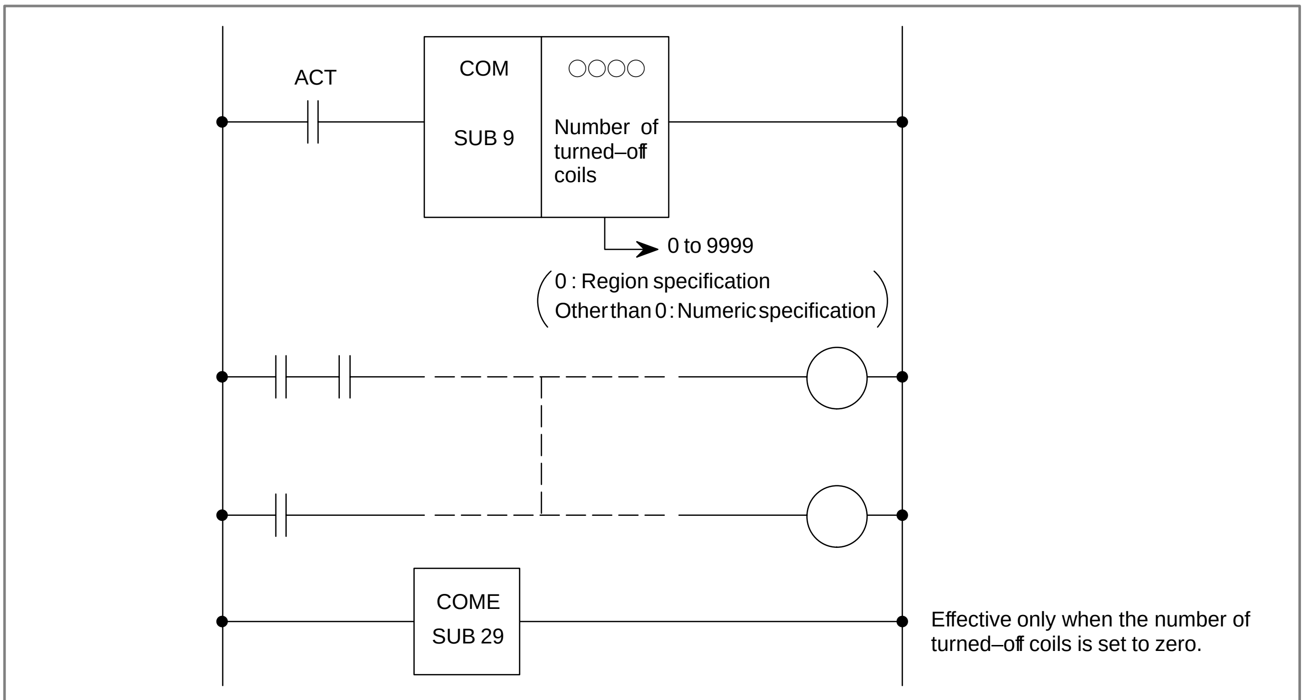


Fig. 5.18.1.1 Function of COM

5.18.1.2 Format

Fig.5.18.1.2 (a) shows the expression format of COM

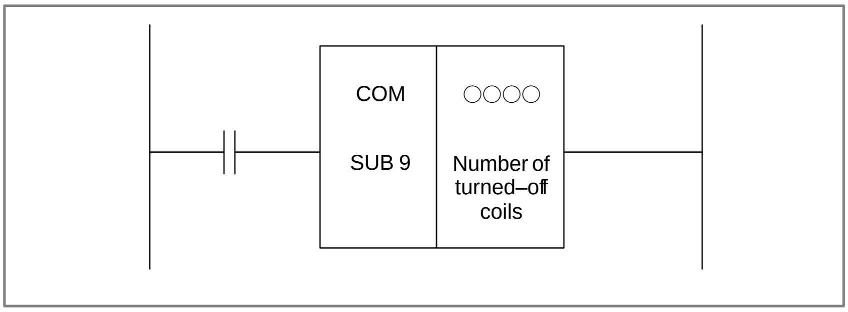


Fig. 5.18.1.2 (a) Expression format of COM

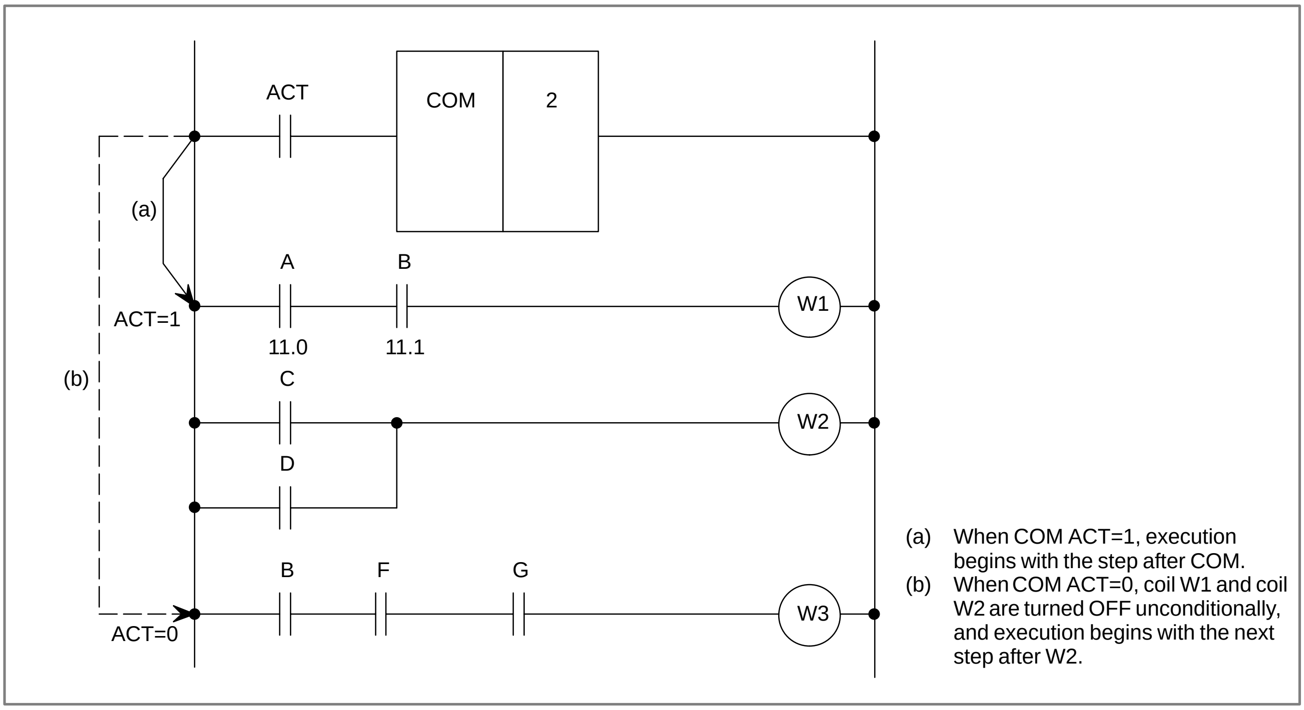


Fig. 5.18.1.2 (b) Ladder diagram for the COM instruction

5.18.2 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.

5.18.3 Parameter

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

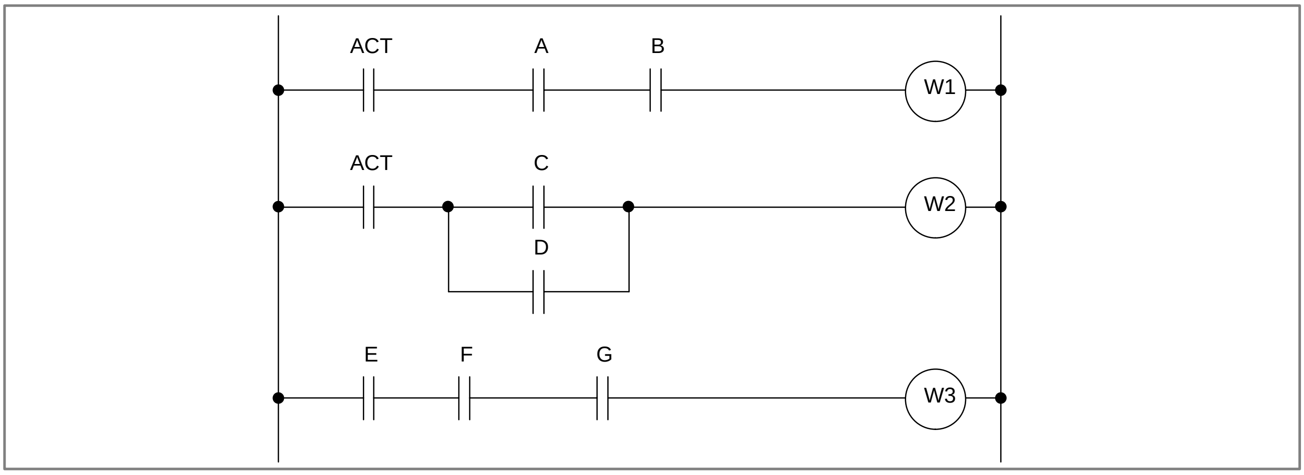


Fig. 5.18.3 (a) Relay circuit example

CAUTION

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.

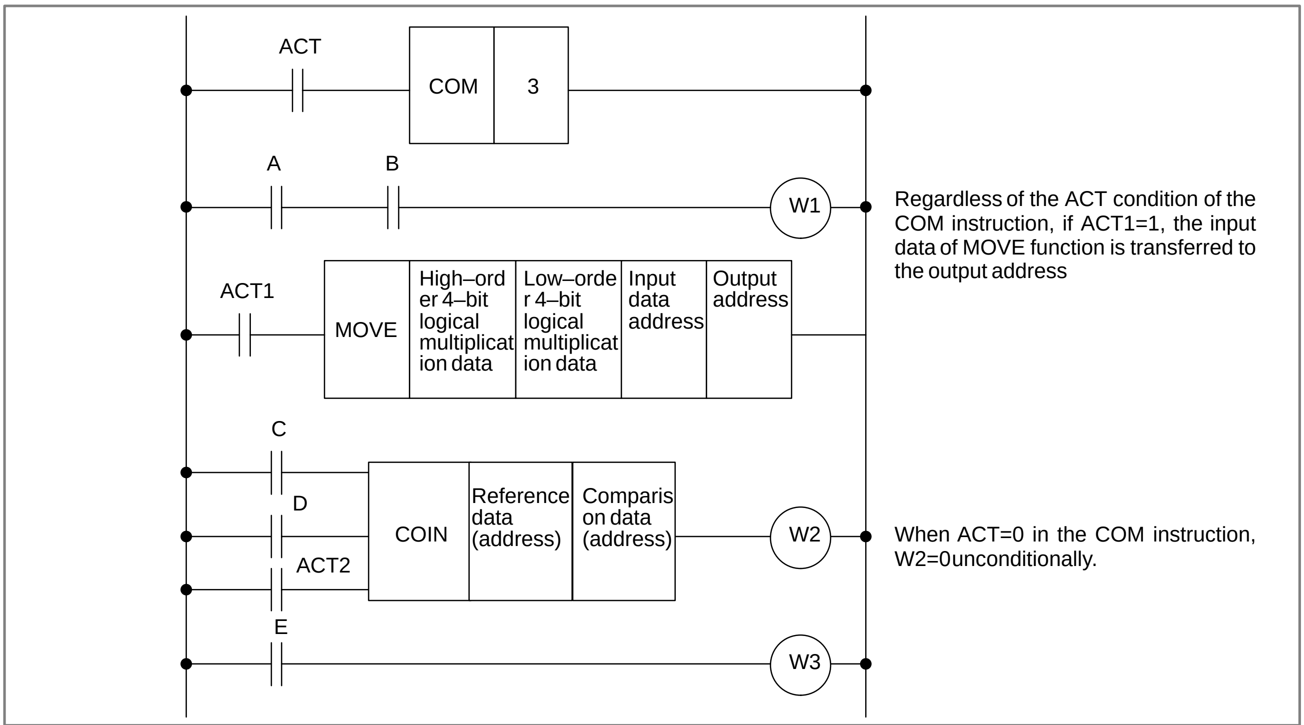


Fig. 5.18.3 (b)

5.18.4 COM (Common Line Control)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

5.18.5 Function

The COM instruction controls the coils in a range up to a common line control end instruction (COME). (See Fig.5.18.5) 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.

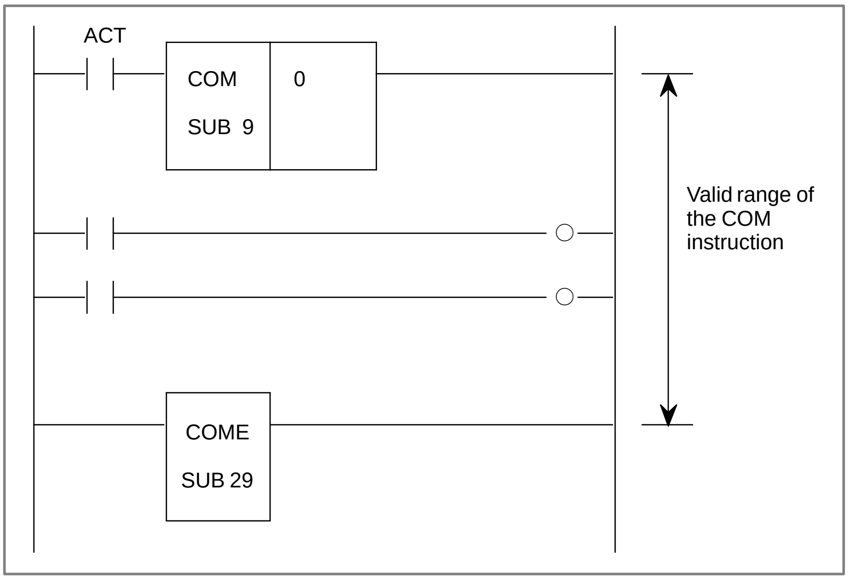


Fig. 5.18.5 Function of COM

5.18.6 Format

Fig.5.18.6 shows the expression format of the functional instruction COM.

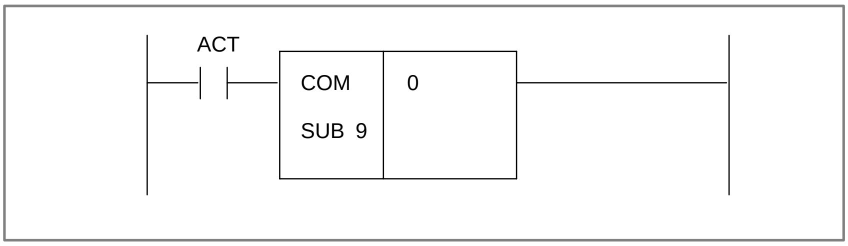


Fig. 5.18.6 Expression Format of COM

5.18.7 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.

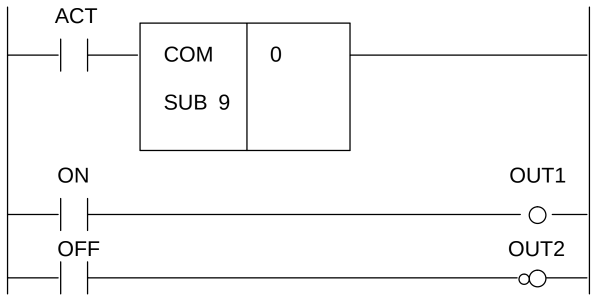
5.18.8 Parameters

(a) Specify 0. (Range specification only)

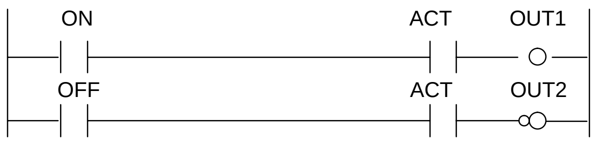
NOTE

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:



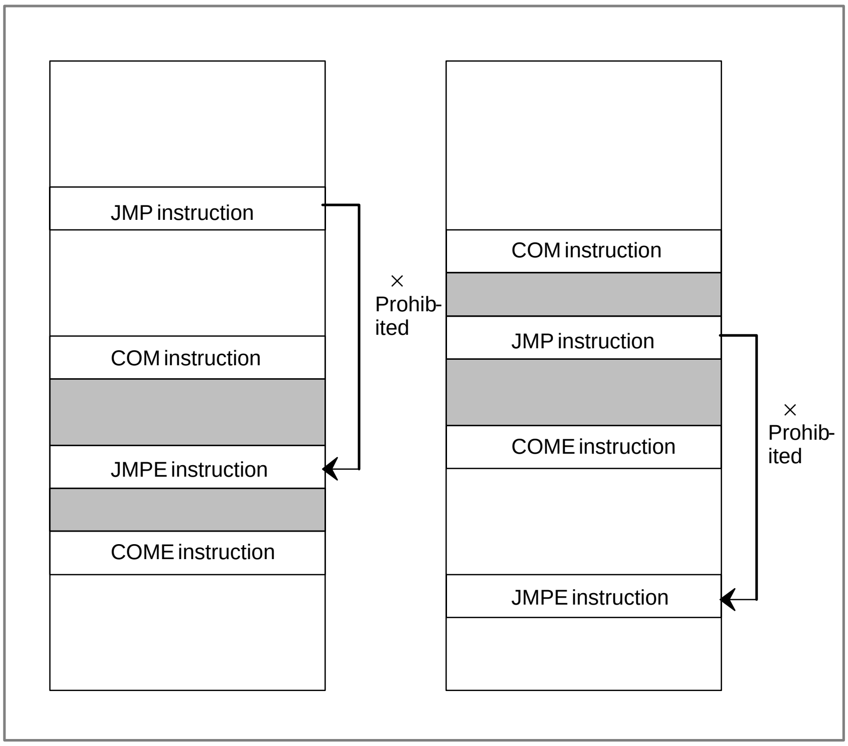
So, the functional instructions in the range specified with a COM instruction are processed, regardless of the setting of ACT of the COM instruction. Note, however, that the coil for the execution of a functional instruction is unconditionally set to 0 when COM ACT = 0.

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.

5.18.9 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.



5.19 COME (COMMON LINE CONTROL END)

5.19.1 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.

5.19.2 Format

Fig.5.19.2 shows the expression format of COME

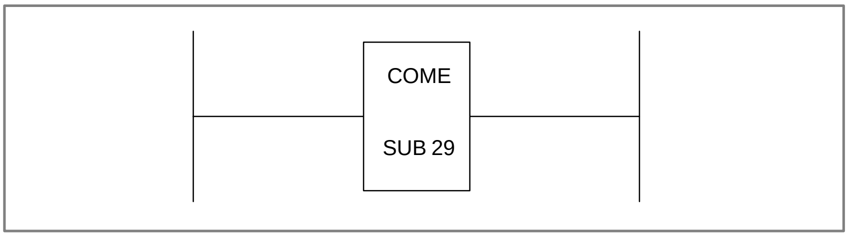


Fig. 5.19.2 Expression format of COME

5.20 JMP (JUMP)

5.20.1 JMP (Jump)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	×	×	×	×	×	○	×	×	×	×	×	×	○	×	×	×	×	×

This function can be used for specifying the number of coils only on the PMC–SB/SC. On other PMCs, specify 0 for the number of coils and use the jump end instruction to use this function. For how to use the instruction, see Subsection 5.20.7.

5.20.2 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.

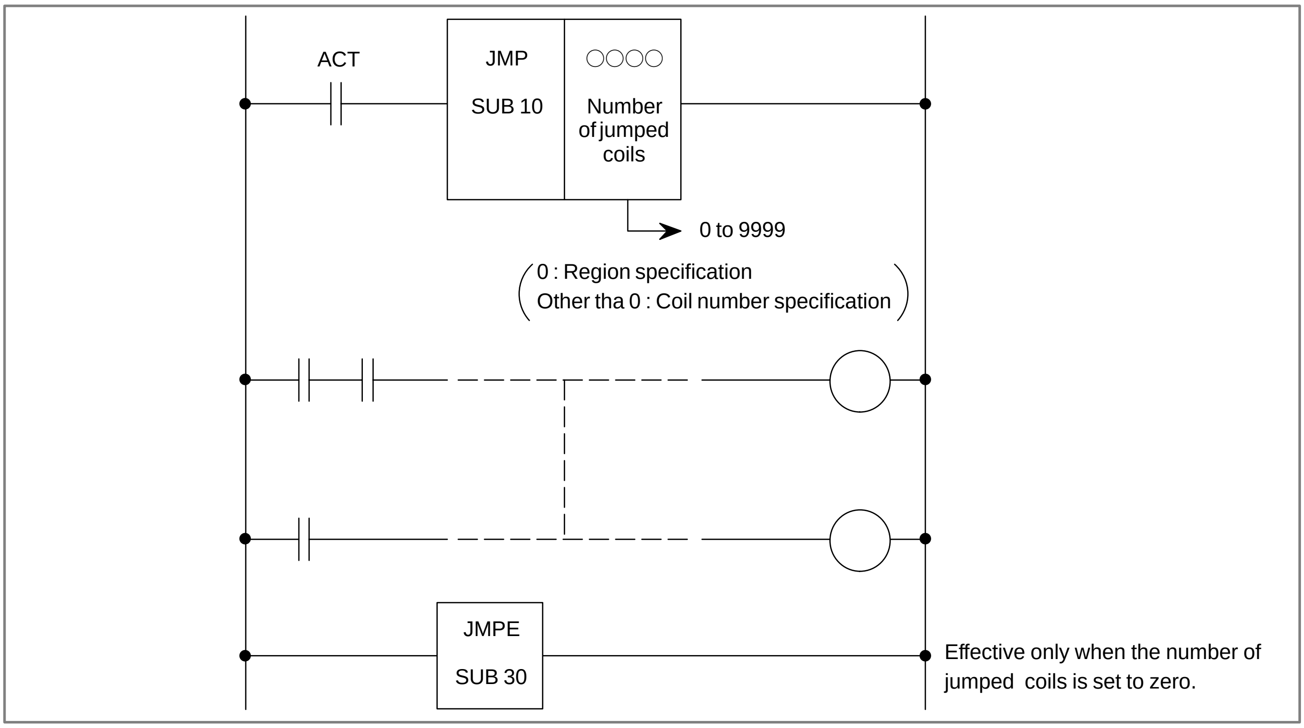
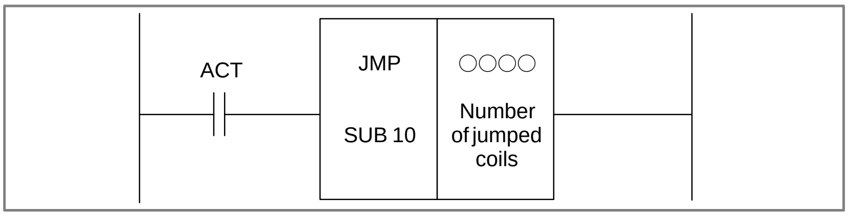


Fig. 5.20.2

5.20.3 Format



5.20.4 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.

5.20.5 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 5.20.5 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.

5.20.6 Operation

Fig.5.20.6 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.5.20.6, 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”).

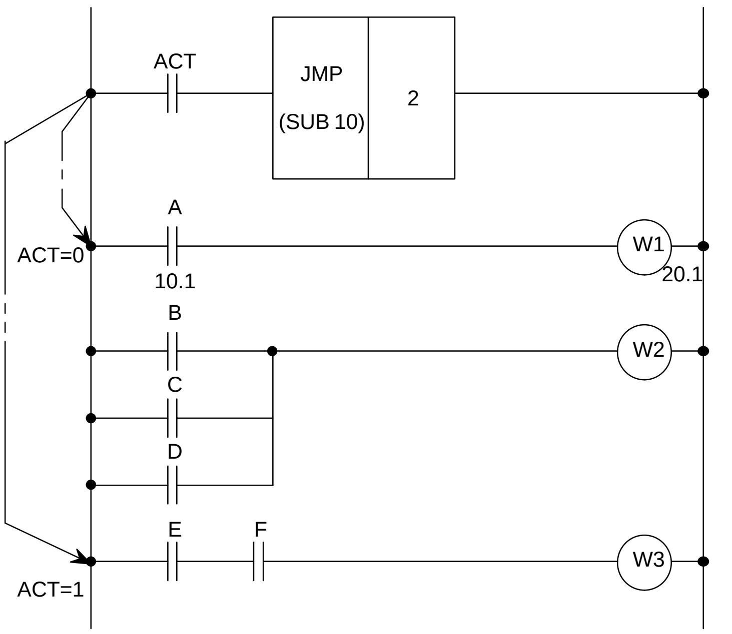


Fig. 5.20.6 Ladder diagram for the JMP instruction

5.20.7 JMP (Jump)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

5.20.7.1 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.5.20.7.1) 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.

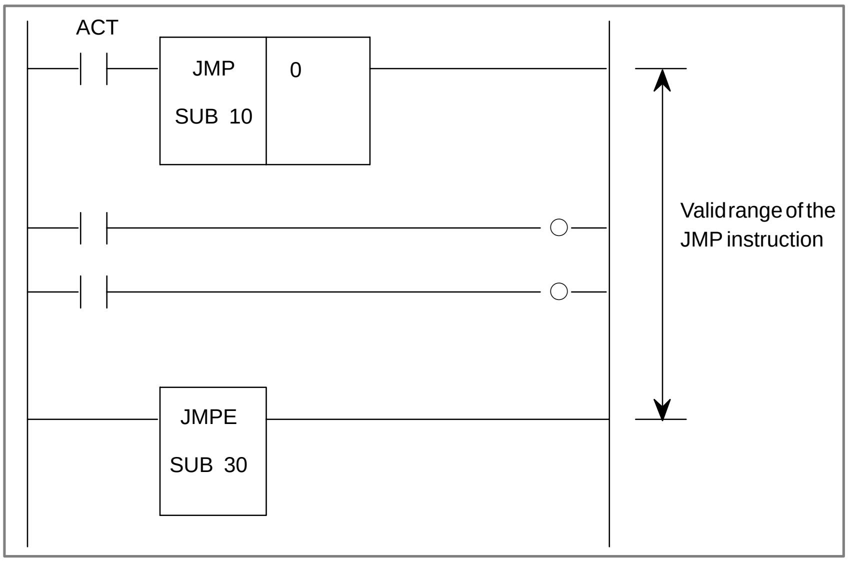


Fig. 5.20.7.1 Function of JMP

5.20.7.2 Format

Fig.5.20.7.2 shows the expression format of the functional instruction JMP.

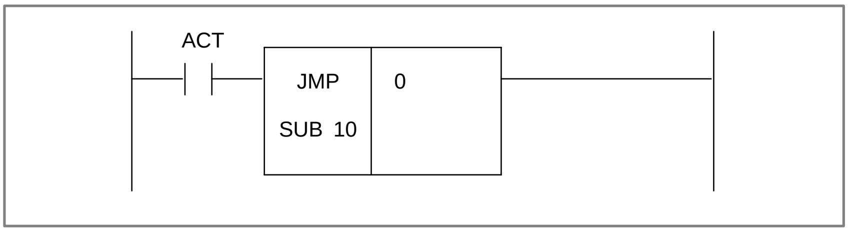


Fig. 5.20.7.2 Expression format of JMP

5.20.7.3 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.

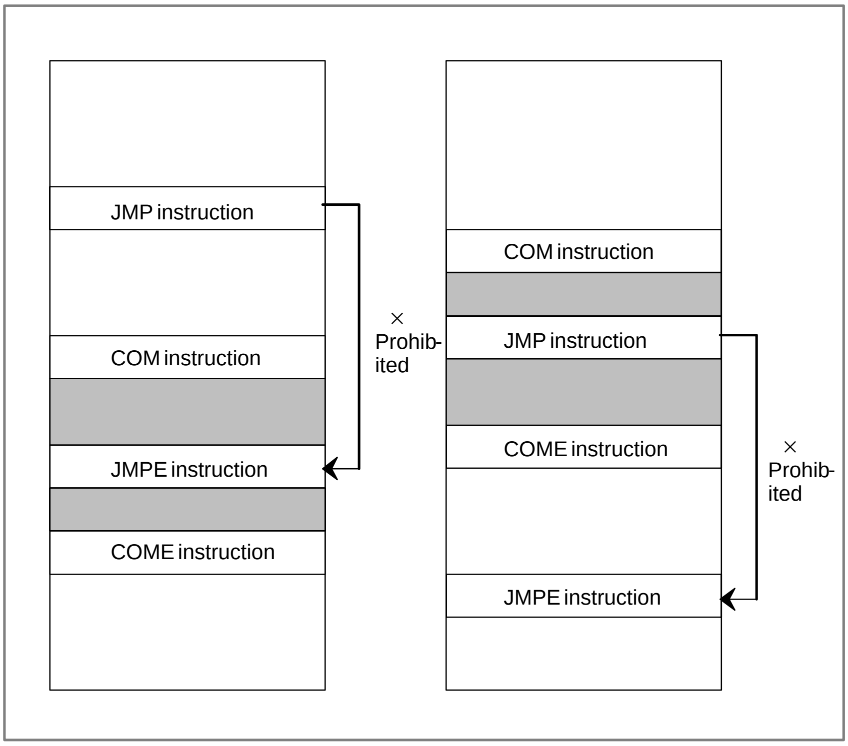
5.20.7.4 Parameters

(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).

5.20.8 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.

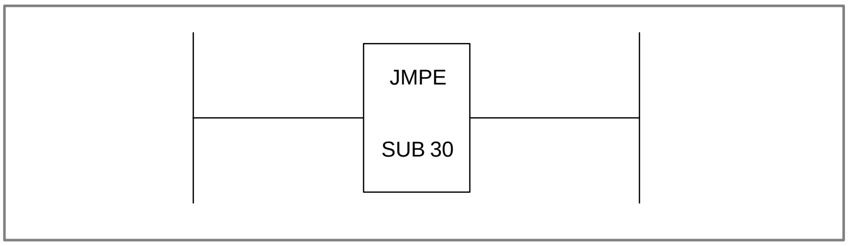


5.21 JMPE (JUMP END)

5.21.1 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.

5.21.2 Format



5.22 PARI (PARITY CHECK)

5.22.1 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.

5.22.2 Format

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

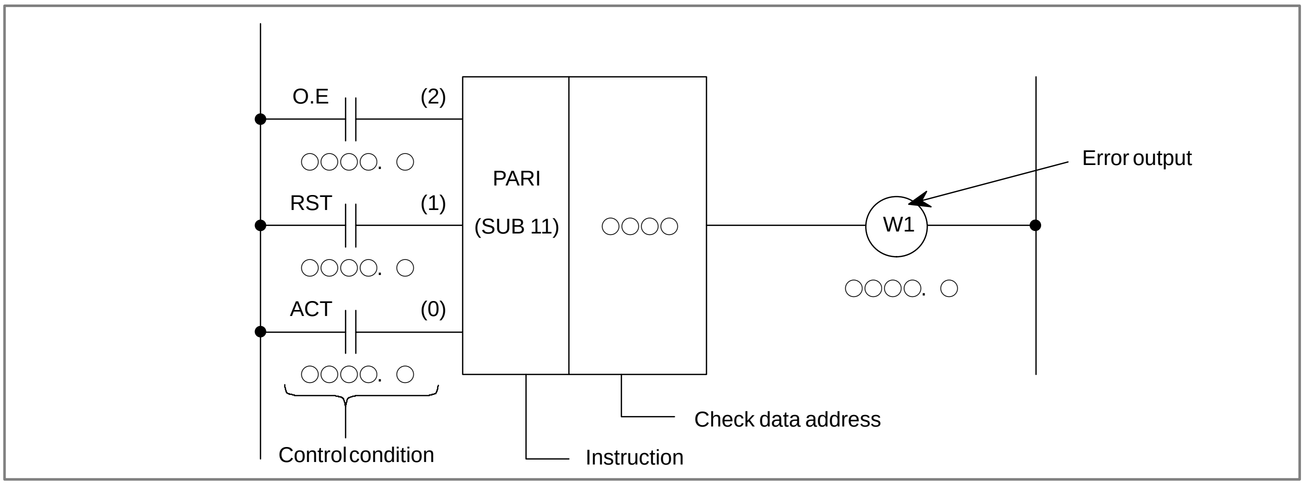


Fig. 5.22.2 PARI instruction format

Table 5.22.2 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

5.22.3 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.

5.22.4 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.

5.22.5 Example of Using the PARI Instruction

Fig.5.22.5 shows odd–parity checking of a code signal entered at address X036.

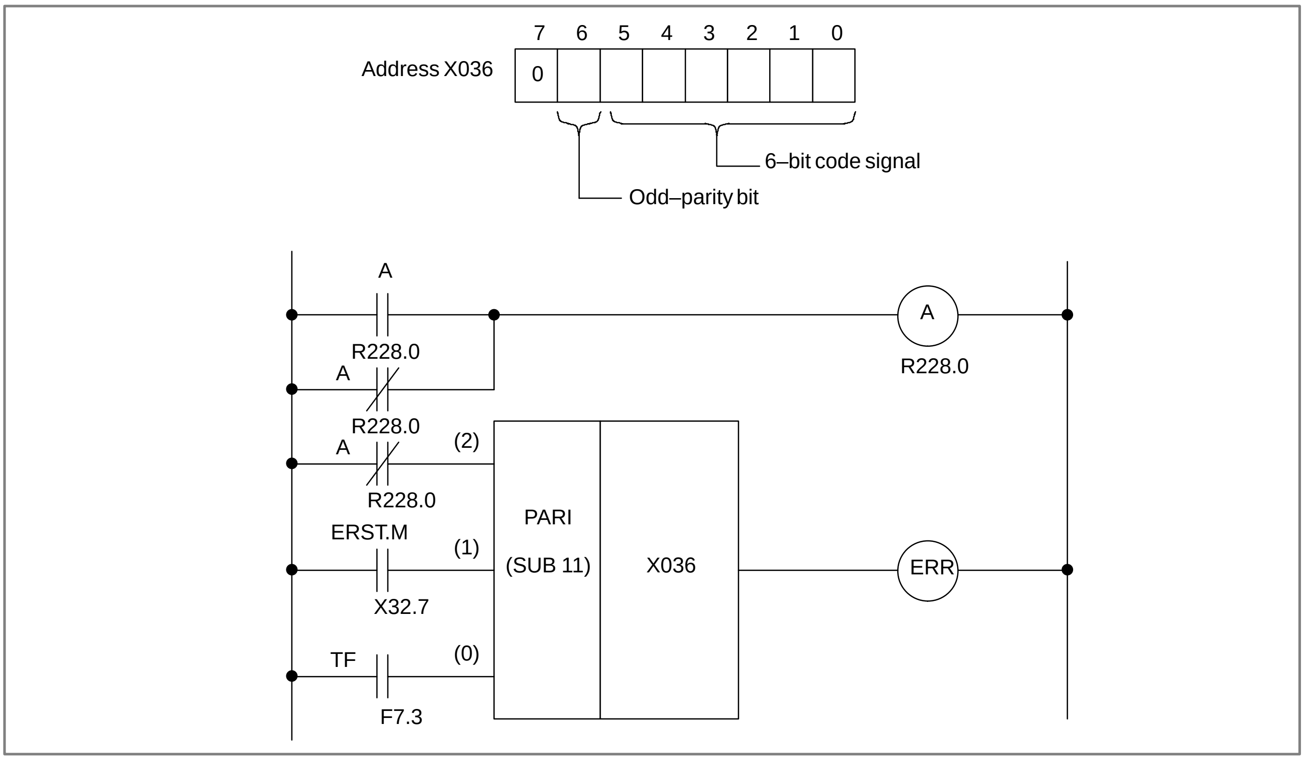


Fig. 5.22.5 Ladder diagram for the PARI instruction

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

5.23 DCNV (DATA CONVERSION)

5.23.1 Function

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

5.23.2 Format

Fig.5.23.2 shows the expression format and Table 5.23.2 shows the coding format.

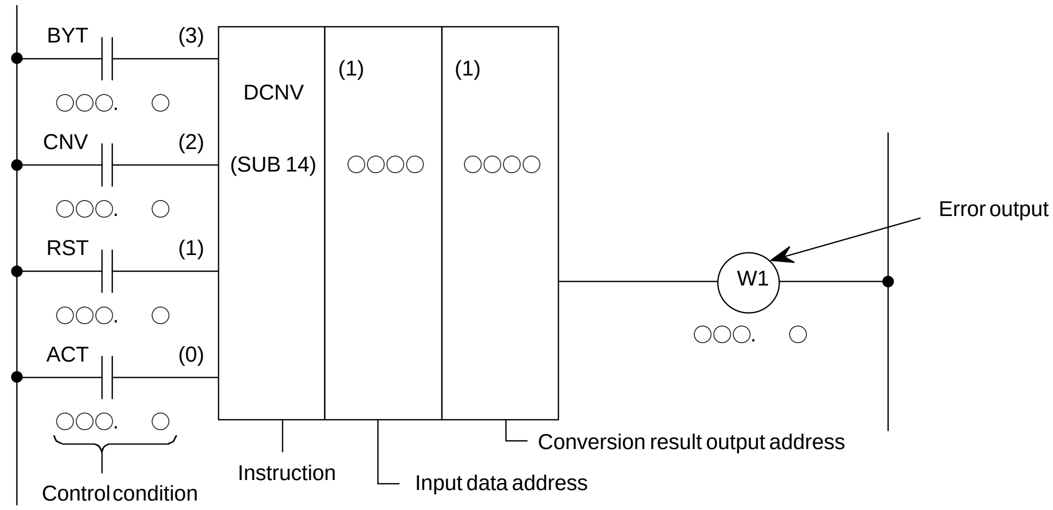


Fig. 5.23.2 DCNV instruction format

Table 5.23.2 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

5.23.3 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.

5.23.4 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.

5.24 DCNVB (EXTENDED DATA CONVERSION)

5.24.1 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.

5.24.2 Format

Fig.5.24.2 shows the expression format of DCNVB

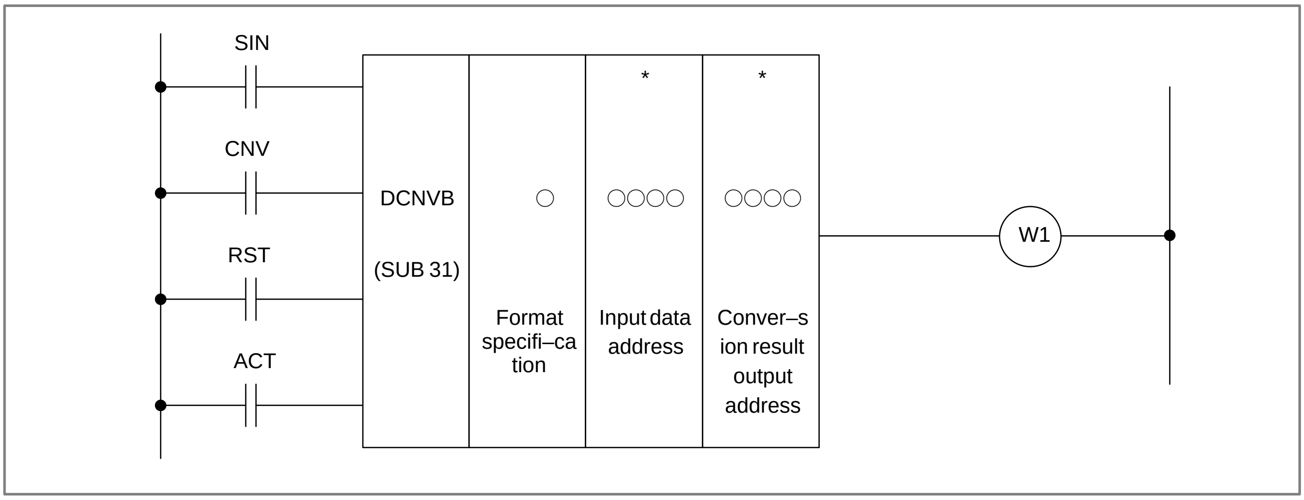


Fig. 5.24.2 Expression format of DCNVB

5.24.3 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.

5.24.4 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.

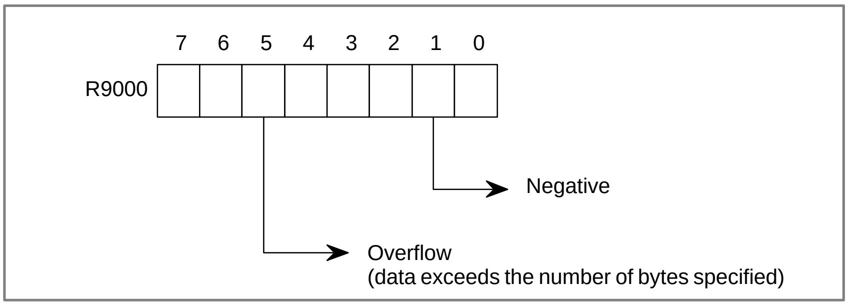
(c) Address for the conversion result output
Specify the address to output the data converted to BCD or binary format.

5.24.5 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.)

5.24.6 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.



5.25 COMP (COMPARISON)

5.25.1 Function

Compares input and comparison values.

5.25.2 Format

Fig.5.25.2 shows the expression format and Table 5.25.2 shows the coding format.

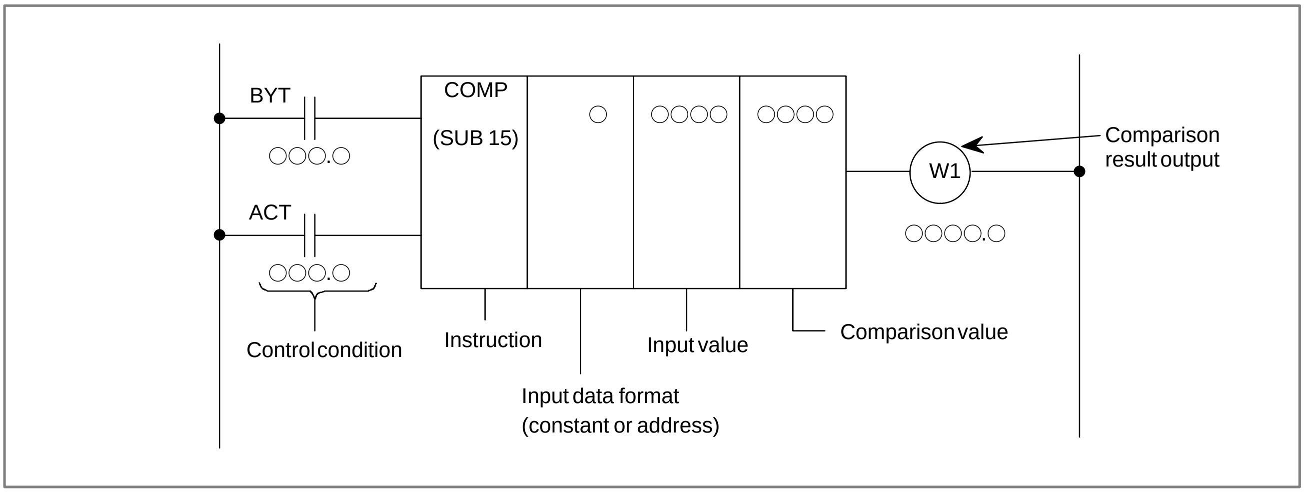


Fig. 5.25.2 COMP instruction format

Table 5.25.2 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

5.25.3 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.

5.25.4 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.

5.25.5 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.

5.25.6 Comparison Data Address

Specifies the address storing the comparison data.

5.25.7 Comparison Result Output

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

5.26 COMPB (COMPARISON BETWEEN BINARY DATA)

5.26.1 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.

5.26.2 Format

Fig.5.26.2 shows the expression format of COMPB.

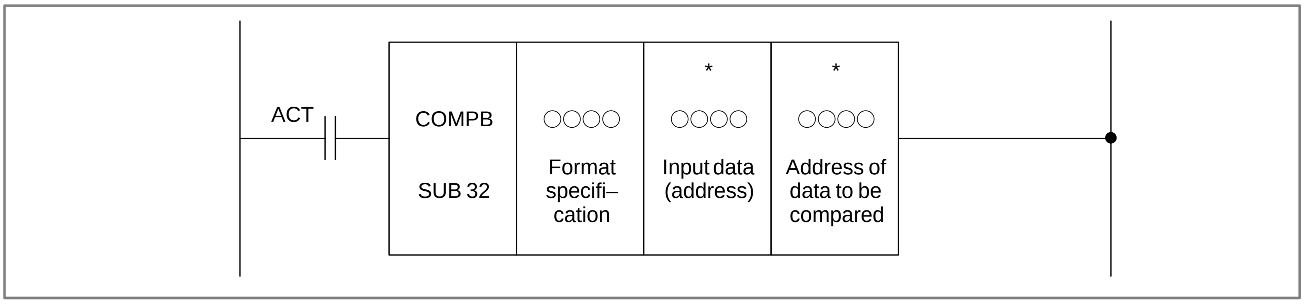


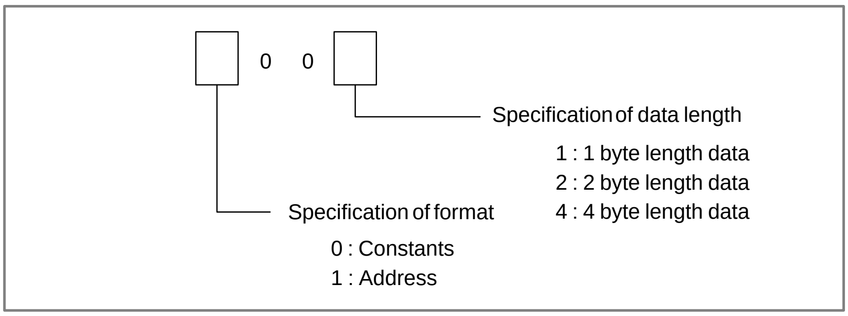
Fig. 5.26.2 Expression format of COMPB

5.26.3 Control Conditions

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

5.26.4 Parameters

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

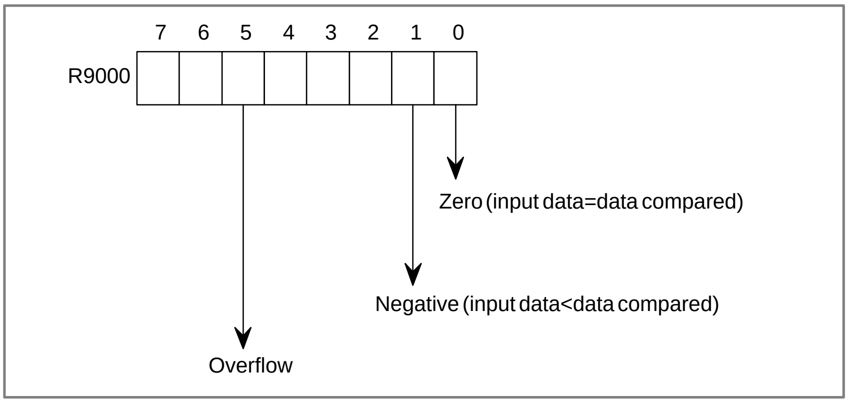


(b) Input data (address)
Format for the input data is determined by the specification in a).

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

5.26.5 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:



5.27 COIN (COINCIDENCE CHECK)

5.27.1 Function

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

5.27.2 Format

Fig.5.27.2 shows the expression format and Table 5.27.2 shows the coding format.

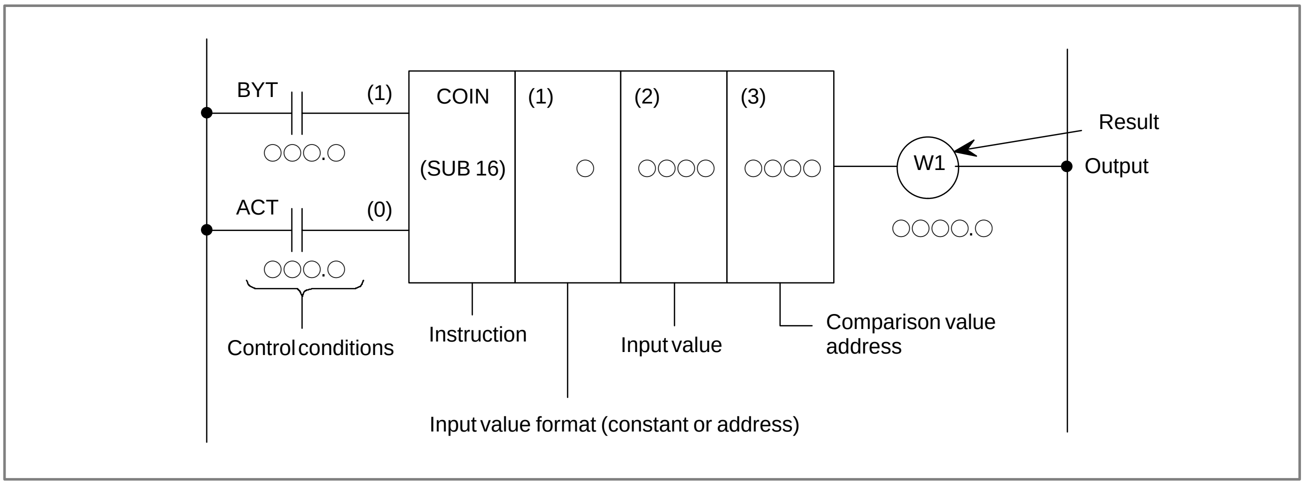


Fig. 5.27.2 COIN instruction format

Table 5.27.2 COIN instruction coding

Coding sheet

Step Number	Instruc- tion	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

5.27.3 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.

5.27.4 Input Data Format.

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

5.27.5 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.

5.27.6 Comparison Data Address

Specifies the address storing the comparison data.

5.27.7 Comparison Result Output

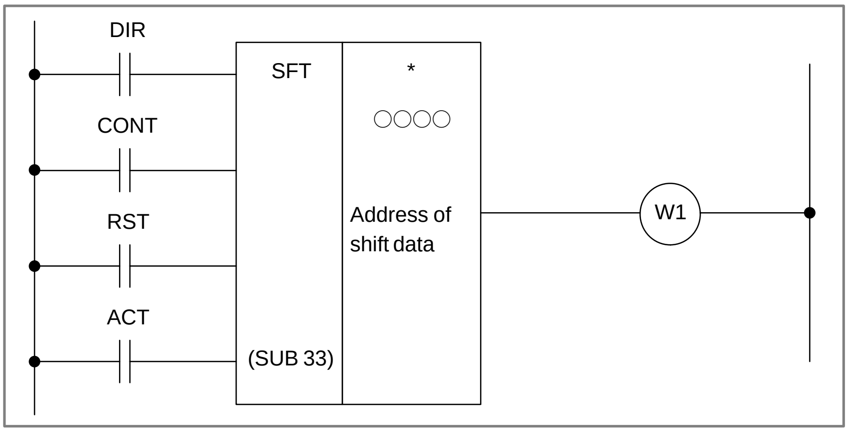
W1=0 : Input data $\neq$ Comparison data
W1=1 : Input data = Comparison data

5.28 SFT (SHIFT REGISTER)

5.28.1 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.

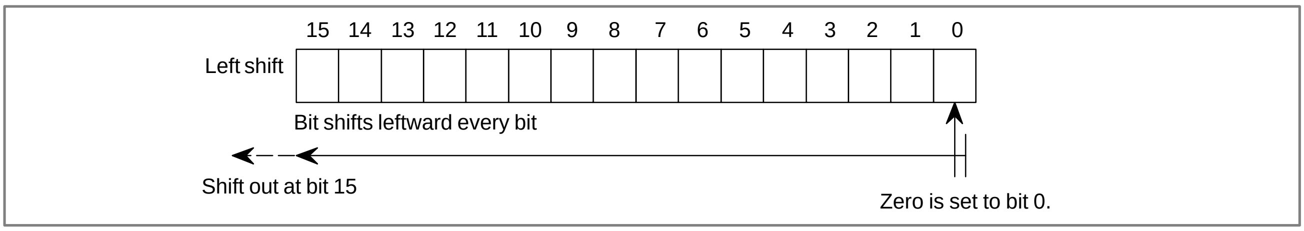
5.28.2 Format



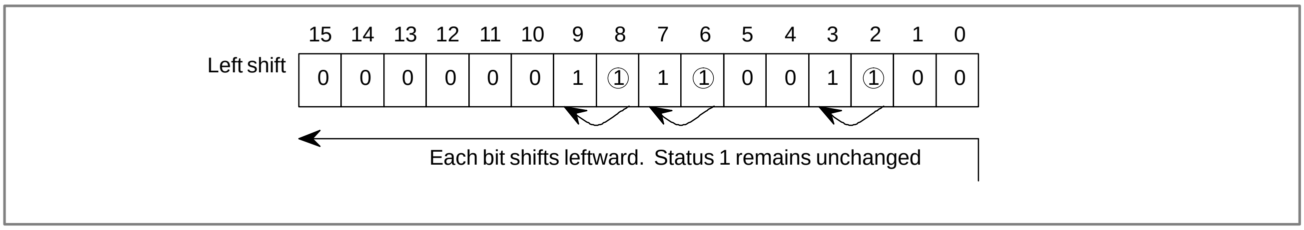
5.28.3 Control Conditions

(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.



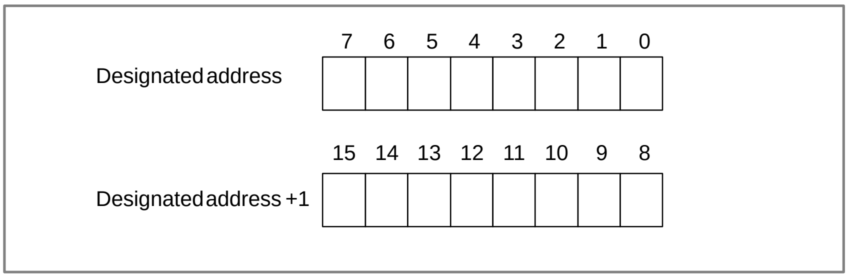
(c) Reset (RST)
The shifted out data (W1=1) is reset (W1=0).
RST=0 : W1 is not reset.
RST=1 : W1 is reset (W1=0).

(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).

5.28.4 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.



5.28.5 Shifted Out

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

5.29 DSCH (DATA SEARCH)

5.29.1 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.

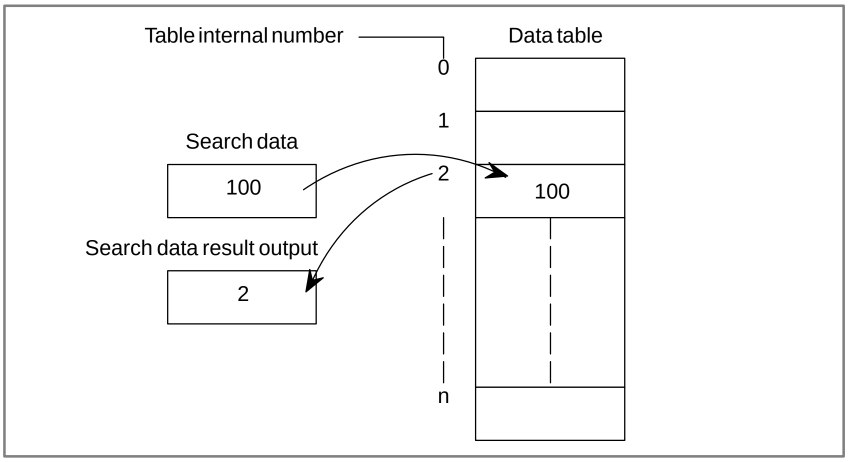


Fig. 5.29.1

> **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.

5.29.2 Format

Fig.5.29.2 shows the expression format and Table 5.29.2 shows the coding format.

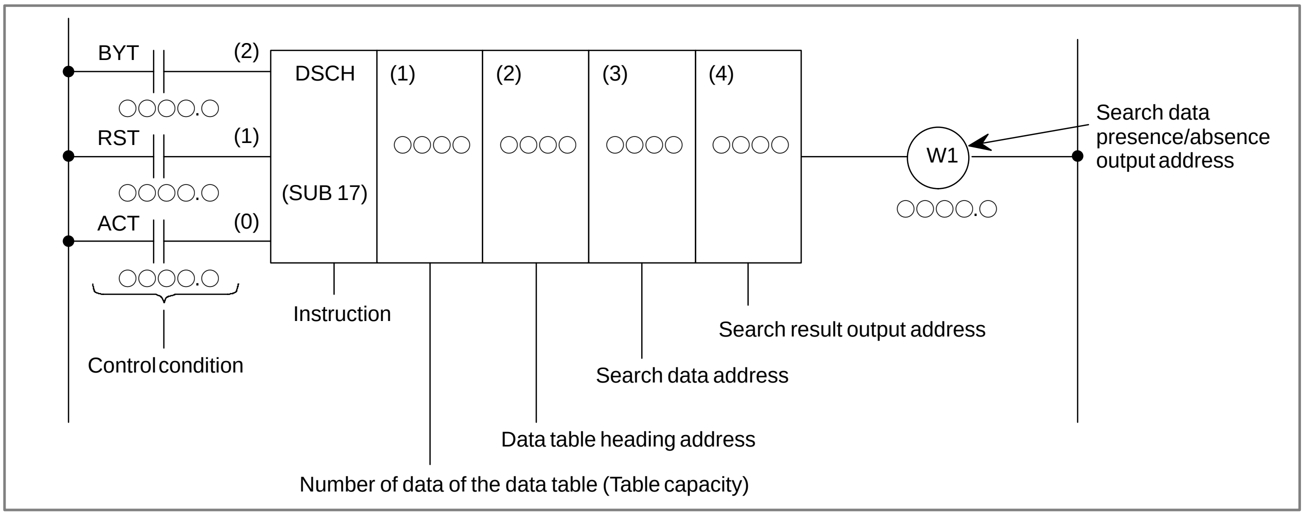


Fig. 5.29.2 DSCH instruction format

Table 5.29.2 DSCH instruction coding

Coding sheet

Step Number	Instruction	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

5.29.3 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.

5.29.4 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.

5.29.5 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.

5.29.6 Search Data Address

Indicates the address of the data to be searched.

5.29.7 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.

5.29.8 Search Data Presence/Absence Output

W1=0 : The data to be searched exists.
W1=1 : The data to be searched does not exist.

5.30 DSCHB (BINARY DATA SEARCH)

5.30.1 Function

Alike the DSCH instruction of Section 5.29, 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.

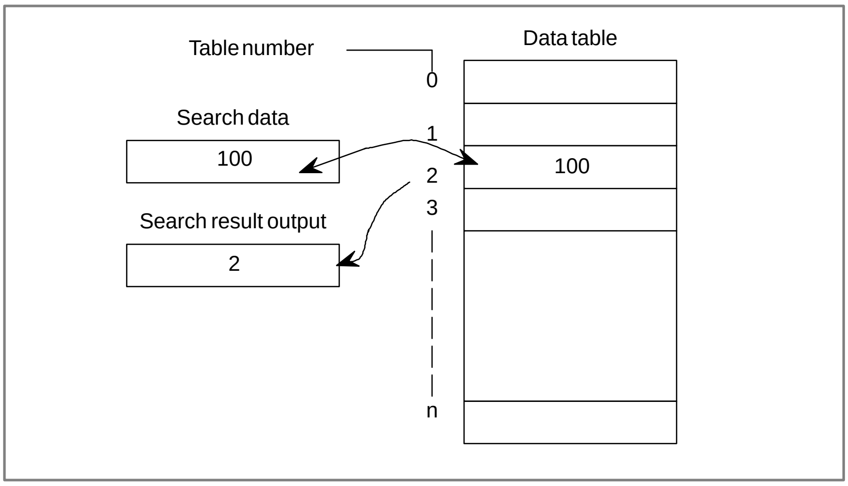


Fig. 5.30.1

5.30.2 Format

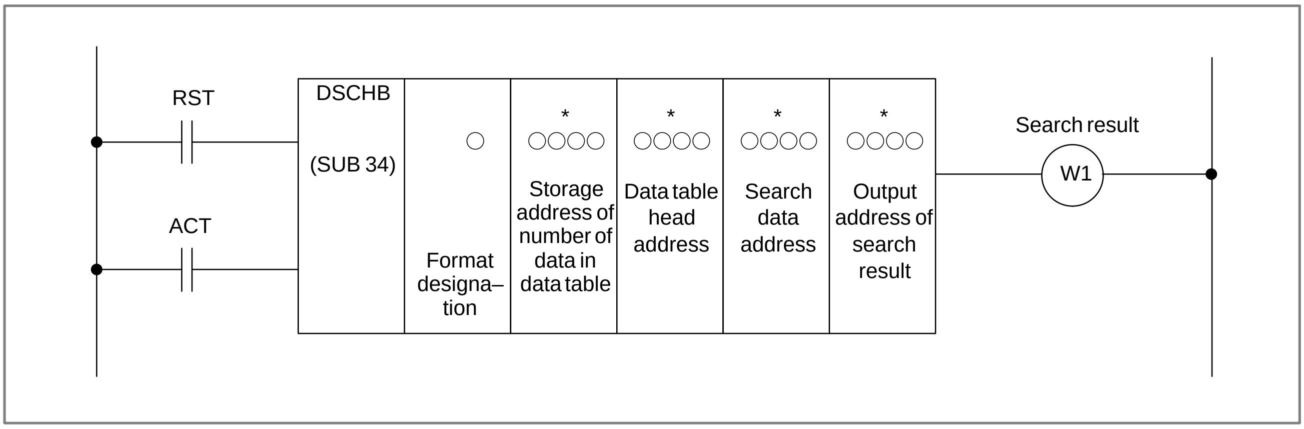


Fig. 5.30.2

5.30.3 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.

5.30.4 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.

5.30.5 Search Result (W1)

W1=0 : Search data found.
W1=1 : Search data not found.

5.31 XMOV (INDEXED DATA TRANSFER)

5.31.1 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.

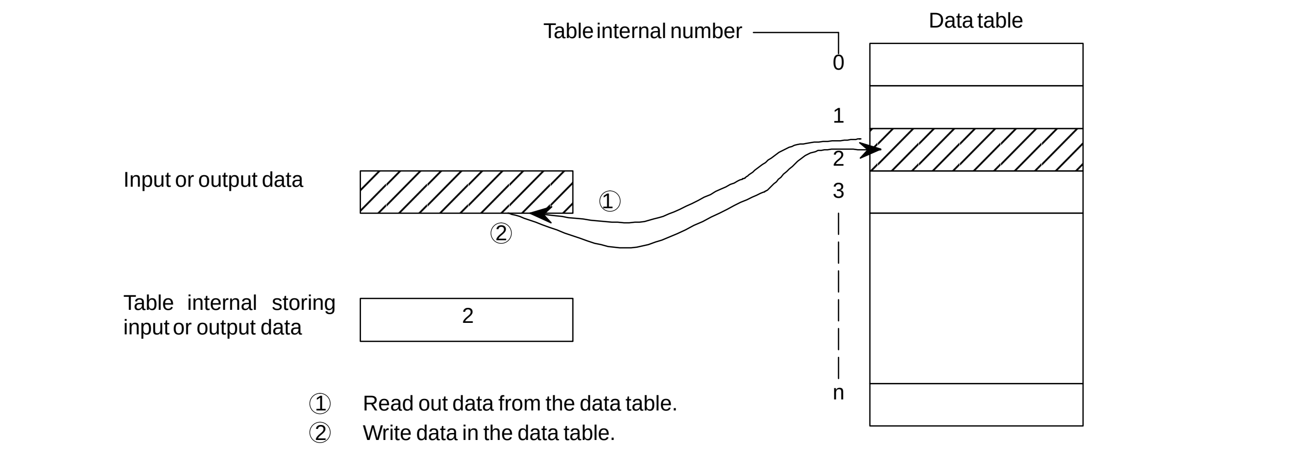


Fig. 5.31.1 Reading and writing of data

5.31.2 Format

Fig.5.31.2 shows the expression format and Table 5.31.2 shows the coding format.

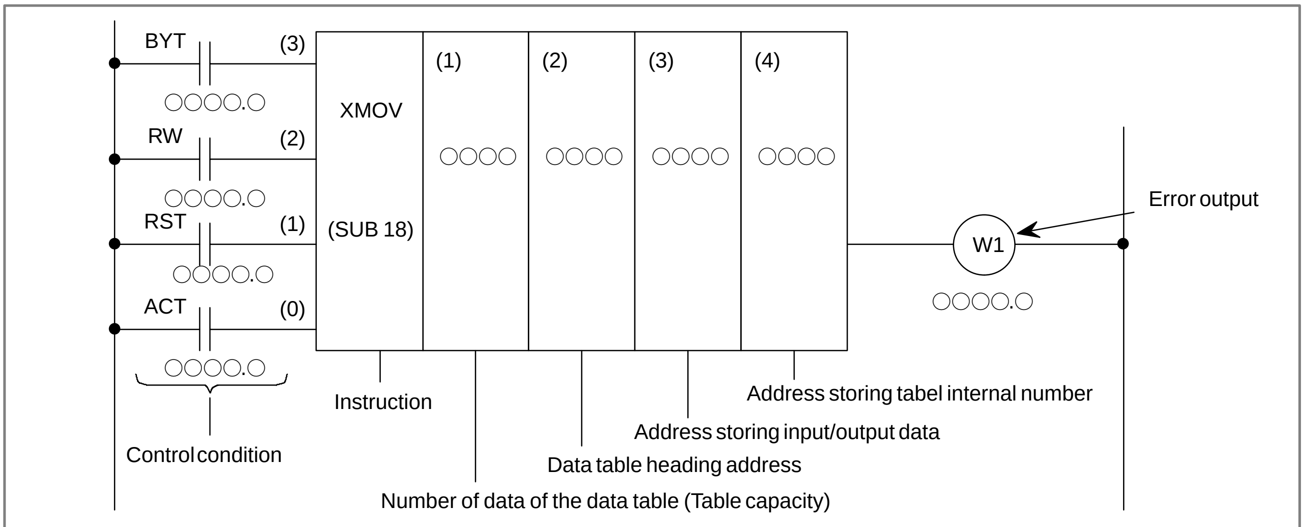


Fig. 5.31.2 XMOV instruction format

Table 5.31.2 XMOV instruction coding

Coding sheet					Memory status of control conditions			
Step Number	**Instruc-tion**	**Address No.**	**Bit No.**	**Remarks**	**ST3**	**ST2**	**ST1**	**ST0**
1	RD	○○○ .	○	BYT				BYT
2	RD. STK	○○○ .	○	RW			BYT	RW
3	RD. STK	○○○ .	○	RST		BYT	RW	RST
4	RD. STK	○○○ .	○	ACT	BYT	RW	RST	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				W1

5.31.3 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.

5.31.4 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.

5.31.5 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 .

5.31.6 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.

5.31.7 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).

5.31.8 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.

5.32 XMOVB (BINARY INDEX MODIFIER DATA TRANSFER)

5.32.1 Function

Alike the XMOV instruction of Section 5.31, 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/SB7 for Series 16*i*/160*i*/18*i*/180*i*/Power Mate *i*, PMC–SA5 for Series 21*i*/210*i*, and PMC–NB6 for Series 15*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 "5.32.4 Parameters".

(a) Read data from data table

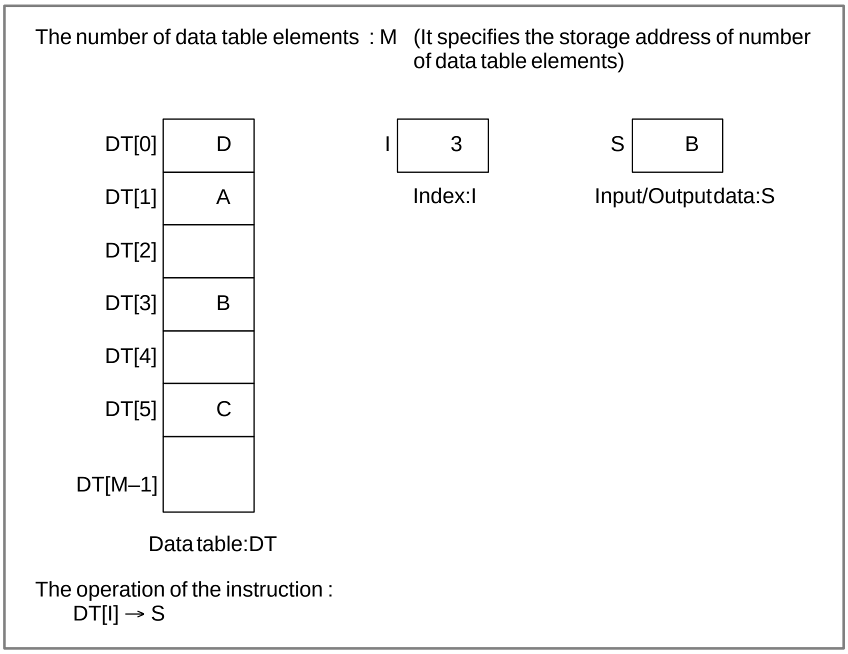
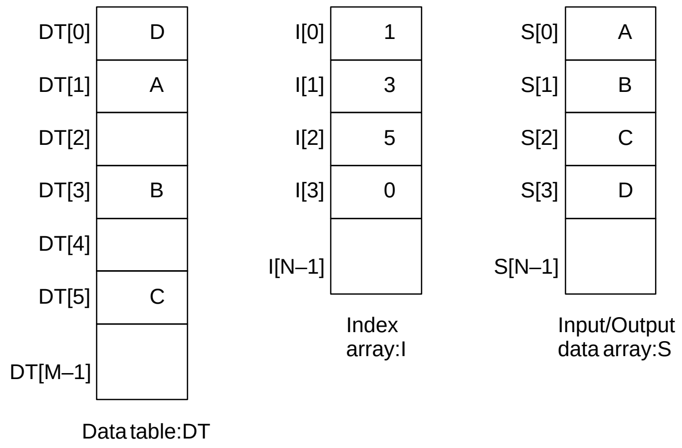


Fig. 5.32.1 (a) Read data from data table (basic specification)

The number of data table elements : M (It specifies the storage address of number of data table elements)

The number of index array elements : N (It specifies the format specification)

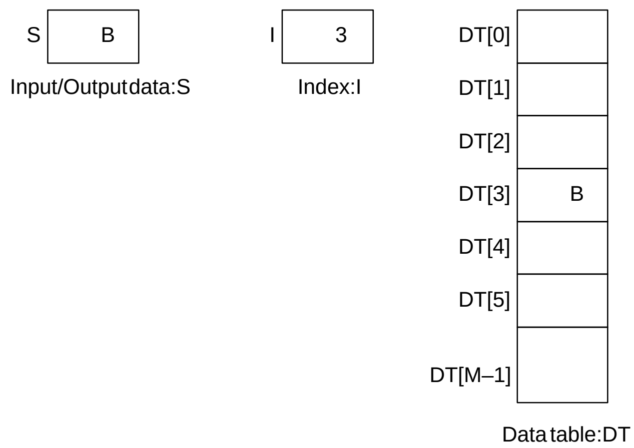


The operation of the instruction :
DT[I[n]] → S[n] (n=0,1,2, . . . ,N–1)

Fig. 5.32.1 (b) Read data from data table (expended specification) (only for PMC–SB5/SB6/SB7 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i*, PMC–SA5/SB6/SB7 for Series 21*i*/210*i*, and PMC–NB6 for Series 15*i*)

(b) Write data to data table

The number of data table elements : M (It specifies the storage address of number of data table elements)

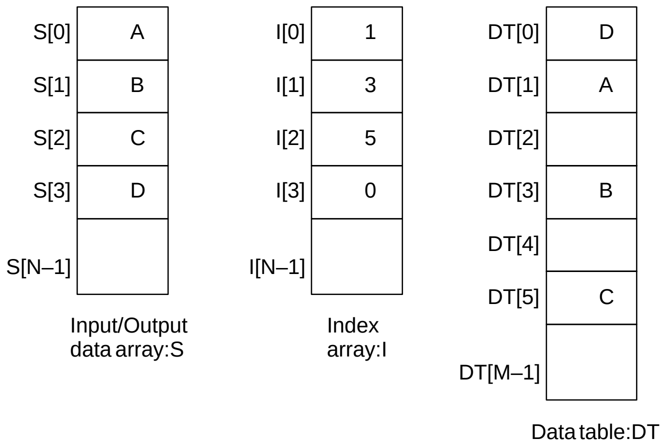


The operation of the instruction :
S → DT[I]

Fig. 5.32.1 (c) Write data to data table (basic specification)

The number of data table elements : M (It specifies the storage address of number of data table elements)

The number of index array elements : N (It specifies the format specification)



The operation of the instruction :
S[n] → DT[I[n]] (n=0,1,2, . . . ,N–1)

Fig. 5.32.1 (d) Write data to data table (expended specification) (only for PMC–SB5/SB6/SB7 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i*, PMC–SA5/SB6/SB7 for Series 21*i*/210*i*, and PMC–NB6 for Series 15*i*)

5.32.2 Format

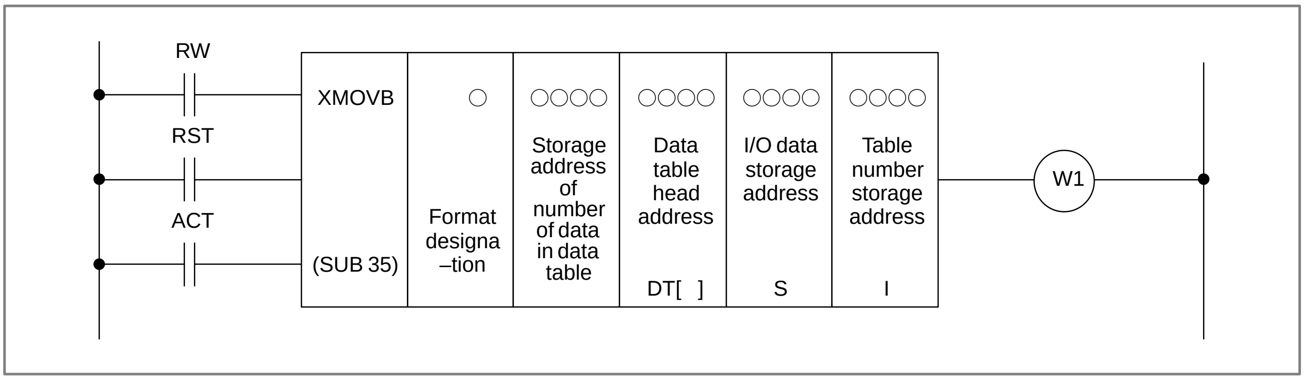


Fig. 5.32.2 (a) XMOVB instruction format

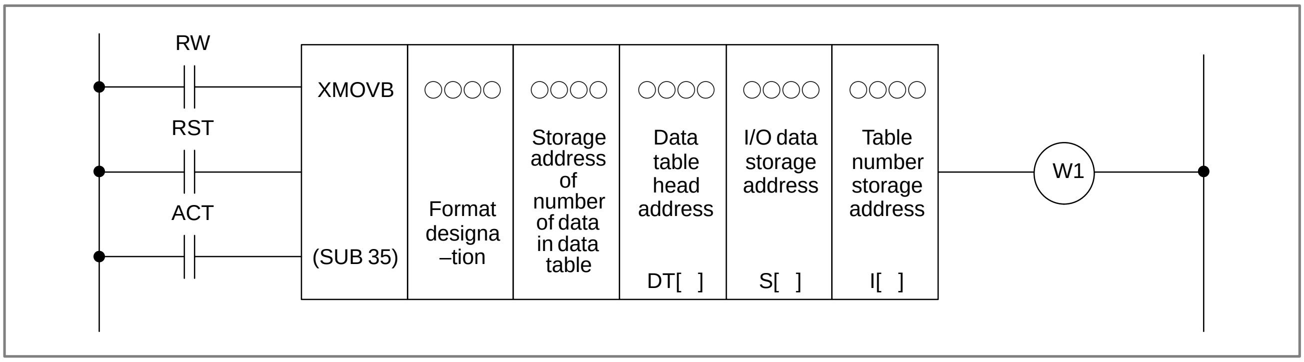


Fig. 5.32.2 (b) XMOVB (expended specification) (only for PMC–SB5/SB6/SB7 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i*, PMC–SA5 for Series 21*i*/210*i*, and PMC–NB6 for Series 15*i*)

5.32.3 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 XMOVB instruction.
There is no change in W1.
ACT=1: Execute XMOVB instruction.

5.32.4 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/SB7 for Series 16*i*/160*i*/18*i*/180*i*/Power Mate *i*, PMC–SA5/SB6/SB7 for Series 21*i*/210*i*, and PMC–NB6 for Series 15*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/SB7 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i*, PMC–SA5/SB6/SB7 for Series 21*i*/210*i*, and PMC–NB6 for Series 15*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/SB7 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i*, PMC–SA5/SB6/SB7 for Series 21*i*/210*i*, and PMC–NB6 for Series 15*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.

5.32.5 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/SB7 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i*, PMC–SA5/SB6/SB7 for Series 21*i*/210*i*, and PMC–NB6 for Series 15*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.

5.32.6 Example for Extended Specification

(a) Read data from data table (extended specification)

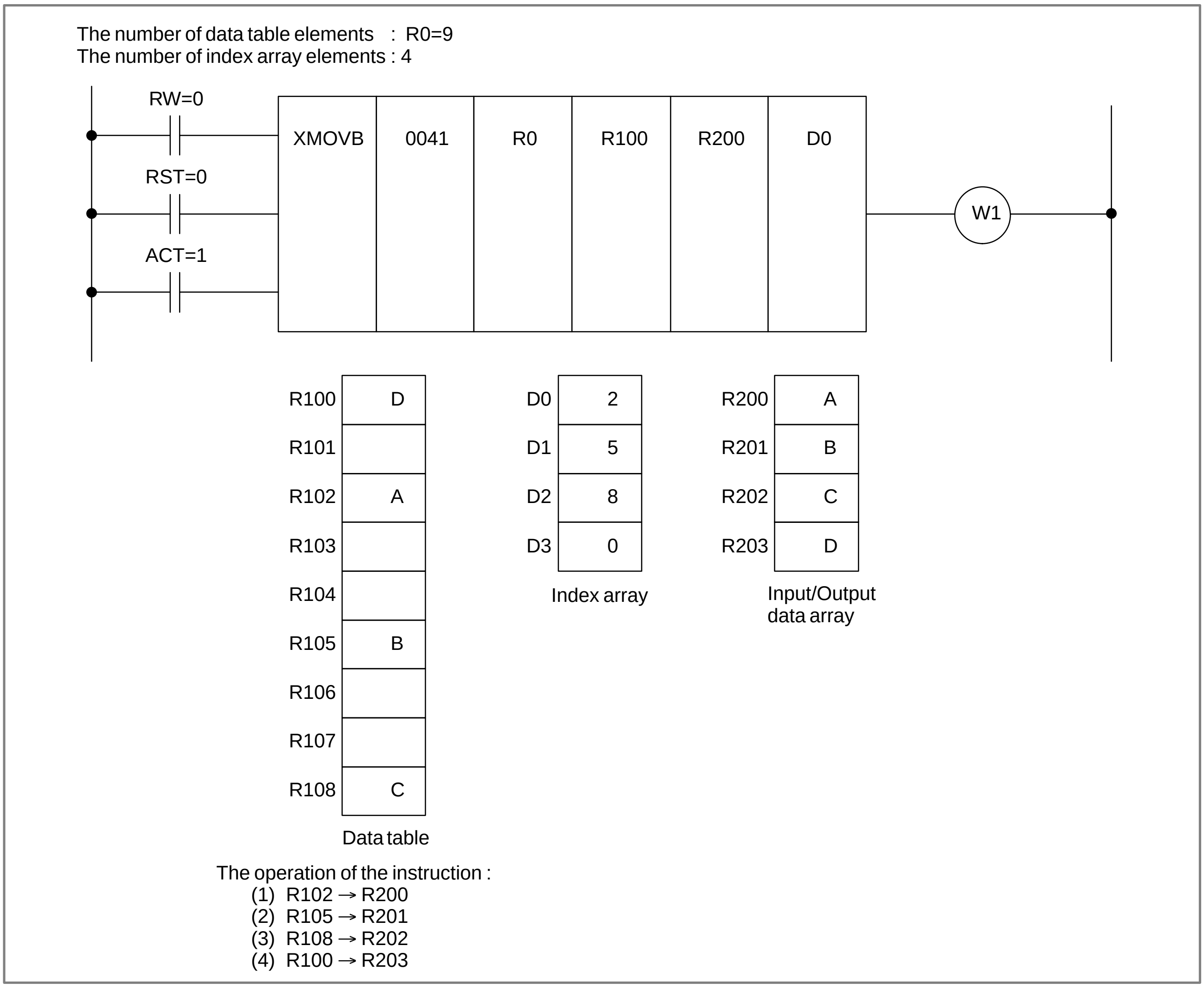
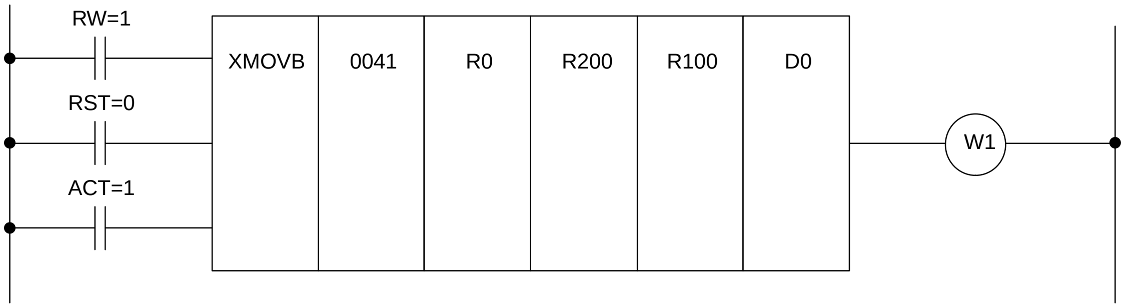


Fig.5.32.6 (a) Example for XMOVB (extended specification) (only for PMC–SB5/SB6/SB7 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i*, PMC–SA5/SB6/SB7 for Series 21*i*/210*i*, and PMC–NB6 for Series 15*i*)

(b) Write data to data table (extended specification)

The number of data table elements : R0=9
The number of index array elements : 4



R100	A
R101	B
R102	C
R103	D

Input/Output data array

D0	2
D1	5
D2	8
D3	0

Index array

R200	D
R201	
R202	A
R203	
R204	
R205	B
R206	
R207	
R208	C

Data table

The operation of the instruction :
(1) R100 → R202
(2) R101 → R205
(3) R102 → R208
(4) R103 → R200

Fig. 5.32.6 (b) Example for XMOVB (extended specification)
(only for PMC–SB5/SB6/SB7 for Series 16*i*/160*i*/18*i*/180*i* Power Mate *i* and PMC–SA5/SB6/SB7 for Series 21*i*/210*i*)

5.33 ADD (ADDITION)

5.33.1 Function

Adds BCD two–or four–digit data.

5.33.2 Format

Fig.5.33.2 shows the expression format and Table 5.33.2 shows the coding format.

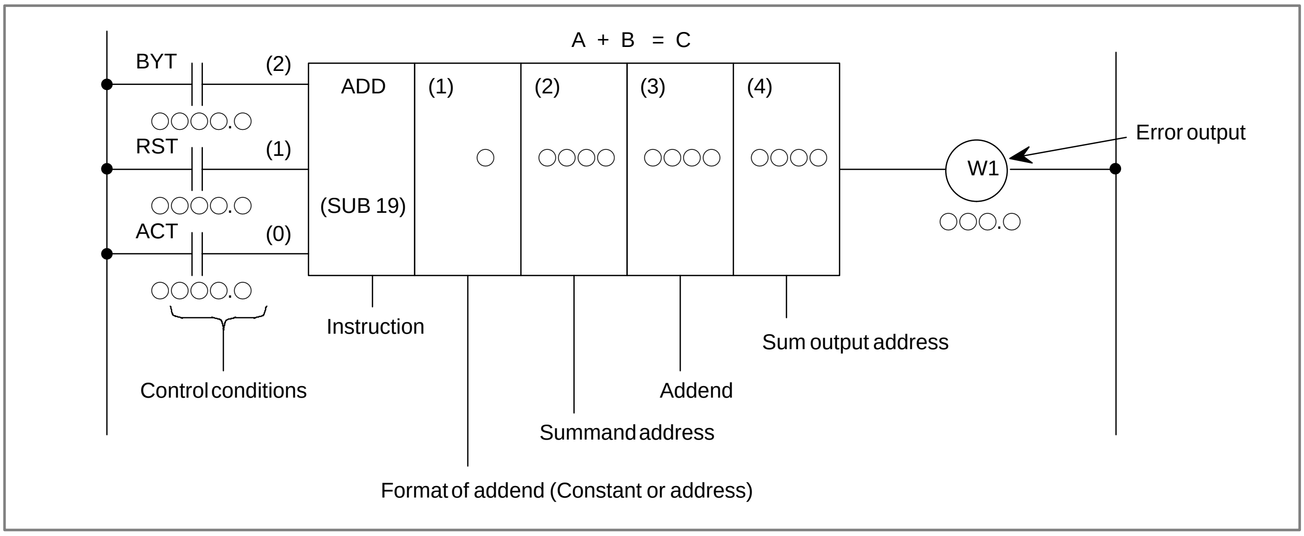


Fig. 5.33.2 ADD instruction format

Table 5.33.2 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

5.33.3 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.

5.33.4 Data Format of Addend

0 : Specifies addend with a constant.
1 : Specifies addend with an address.

5.33.5 Summand Address

Set the address storing the summand.

5.33.6 Addend (Address)

Addressing of the addend depends on 5.32.4.

5.33.7 Sum Output Address

Set the address to which the sum is to be output.

5.33.8 Error Output

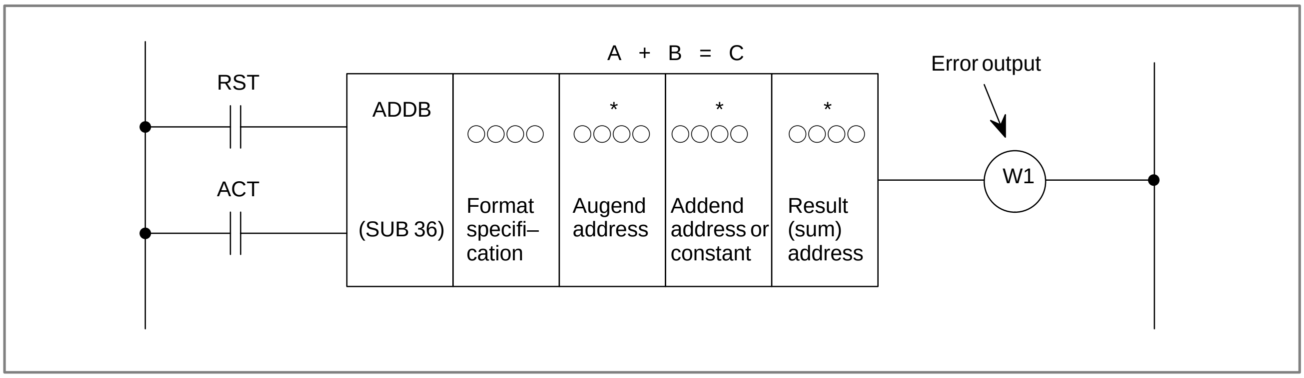
If the sum exceeds the data size specified in 5.32.3–a), W1=1 is set to indicate an error.

5.34 ADDB (BINARY ADDITION)

5.34.1 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.

5.34.2 Format



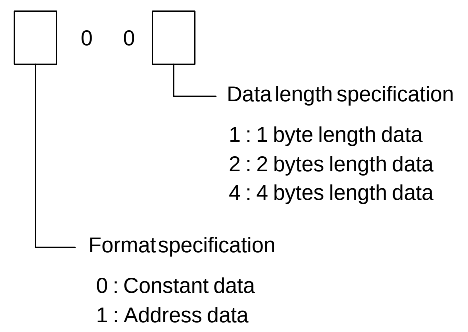
5.34.3 Control Conditions

(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.

5.34.4 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.

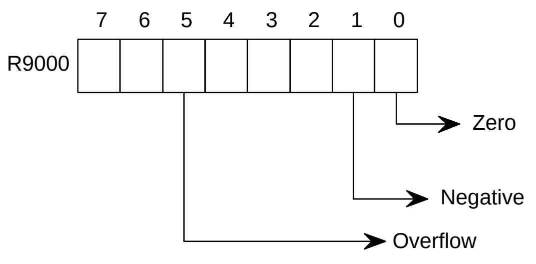
(d) Result output address
Specifies the address to contain the result of operation.

5.34.5 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.

5.34.6 Operation Output Register (R9000)

This register is set with data on operation. If register bit is on, they signify the following operation data:



5.35 SUB (SUBTRACTION)

5.35.1 Function

Subtracts BCD two–or four–digit data.

5.35.2 Format

Fig.5.35.2 shows the expression format and Table 5.35.2 shows the coding format.

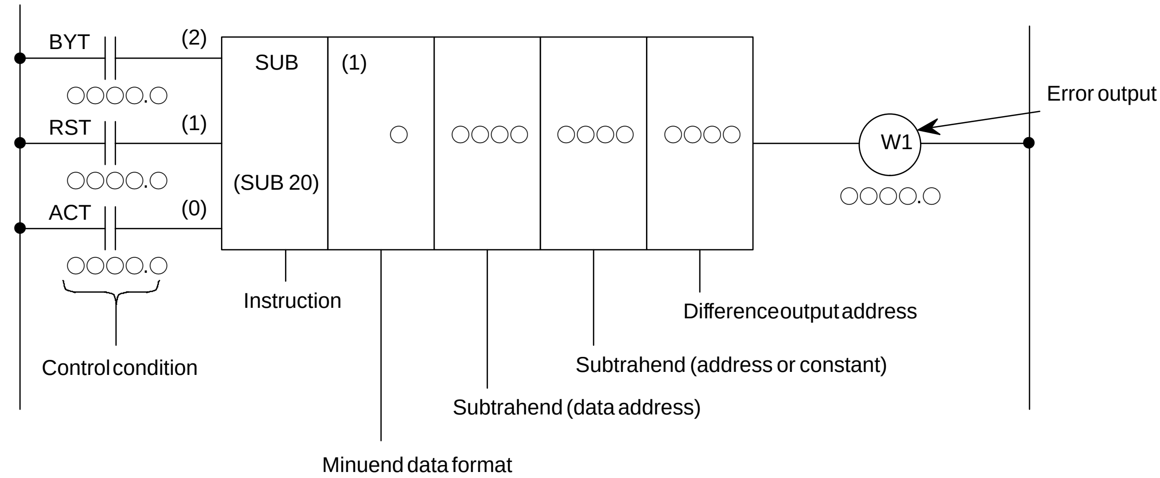


Fig. 5.35.2 SUB Instruction format

Table 5.35.2 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

5.35.3 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

5.35.4 Reset

RST=0 : Release reset.
RST=1 : Resets error output W1, that is, sets W1 to 0.

5.35.5 Execution Command

ACT=0 : The SUB instruction is not executed. W1 does not change.
ACT=1 : The SUB instruction is executed.

5.35.6 Data Format of Subtrahend

0 : Specifies subtrahend with a constant.
1 : Specifies subtrahend with an address.

5.35.7 Minuend Address

Set the address storing the minuend.

5.35.8 Subtrahend (Address)

Addressing of the subtrahend depends on 5.35.6.

5.35.9 Difference Output Address

Sets the address to which the difference is output.

5.35.10 Error Output

W1 is set 1 to indicate an error if the difference is negative.

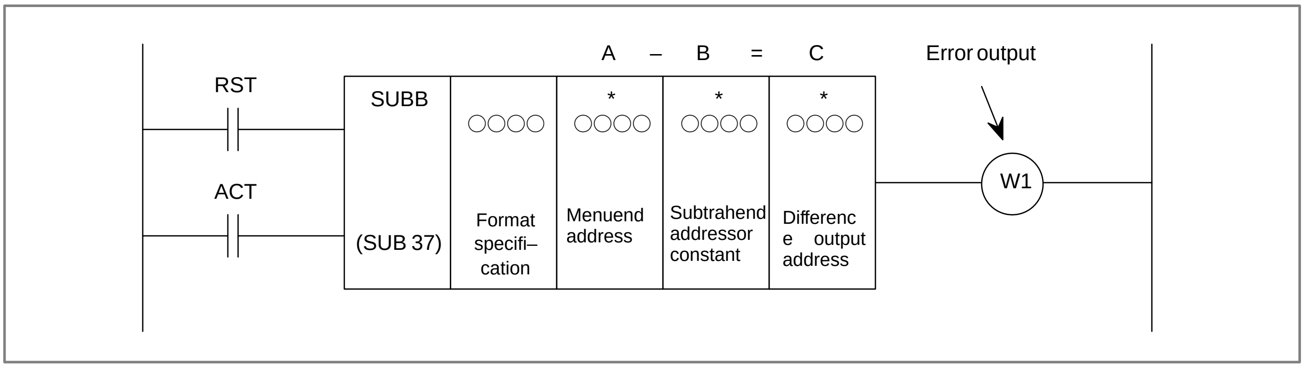
5.36 SUBB (BINARY SUBTRACTION)

5.36.1 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).

5.36.2 Format



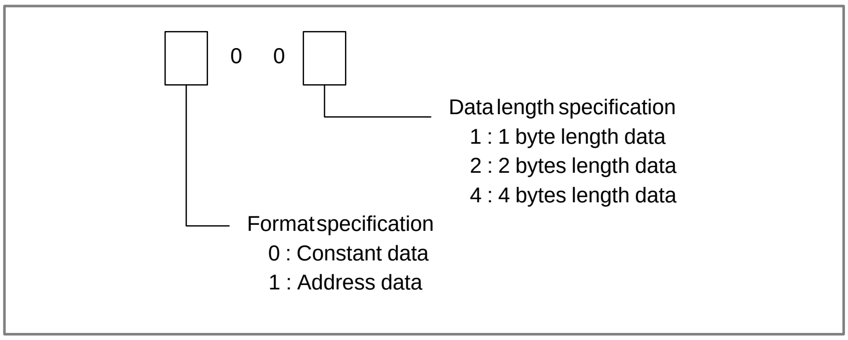
5.36.3 Control Conditions

(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.

5.36.4 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.

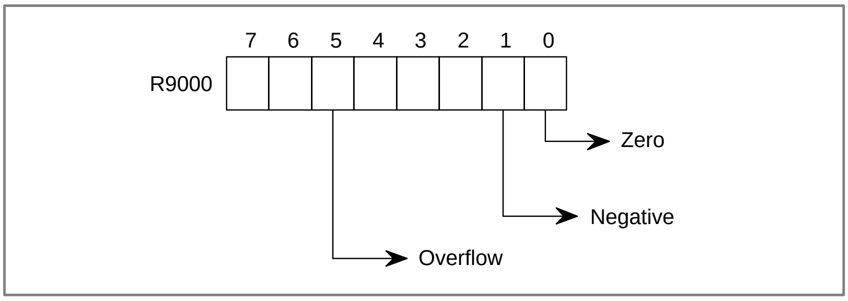
(d) Result output address
Specifies the address to contain the result of operation.

5.36.5 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.

5.36.6 Operation Output Register (R9000)

This register is set with data on operation. If register bit is on, they signify the following operation data:



5.37 MUL (MULTIPLICATION)

5.37.1 Function

Multiplies BCD two–or four–digit data. The product must also be BCD two–or four–digit data.

5.37.2 Format

Fig.5.37.2 shows the expression format and Table 5.37.2 shows the coding format.

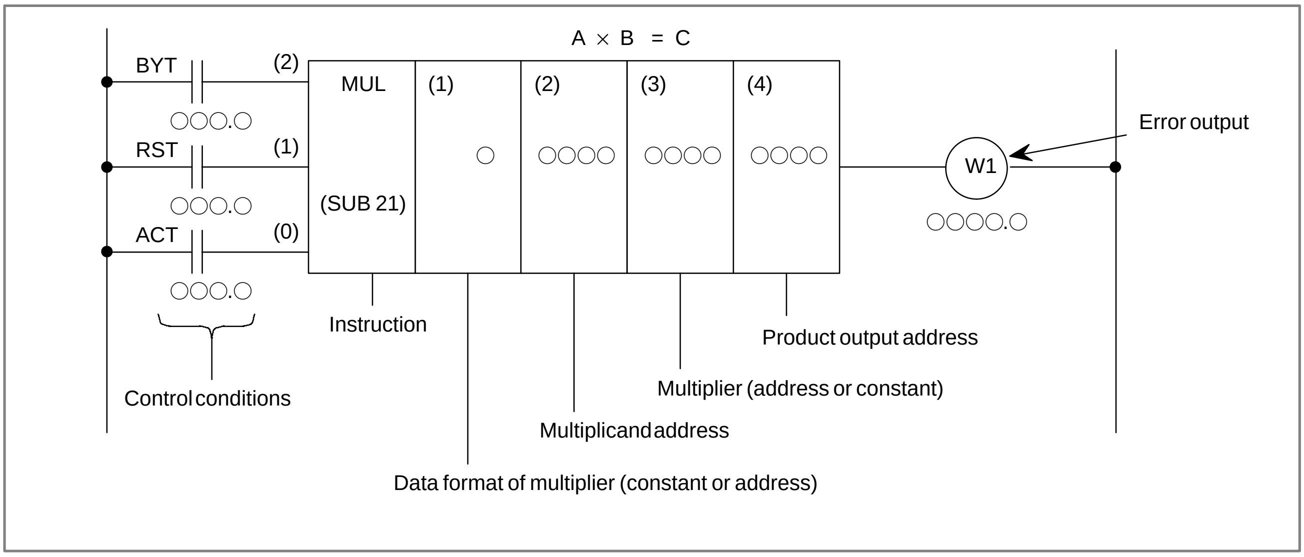


Fig. 5.37.2 MUL instruction format

Table 5.37.2 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

5.37.3 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.

5.37.4 Data Format of Multiplier

0 : Specifies multiplier with a constant.
1 : Specifies multiplier with an address.

5.37.5 Multiplicand Address

Sets the address storing the multiplicand.

5.37.6 Multiplier (Address)

Addressing of the multiplier depends on 4).

5.37.7 Product Output Address

Set the address to which the product is output.

5.37.8 Error Output

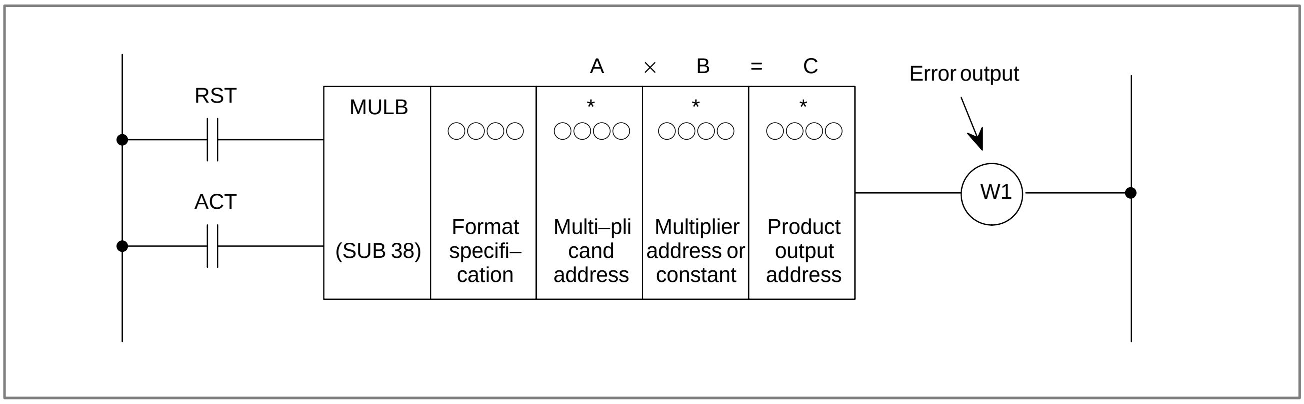
W1=1 is set to indicate an error if the product exceeds the size specified in 5.37.3–a).

5.38 MULB (BINARY MULTIPLICATION)

5.38.1 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).

5.38.2 Format



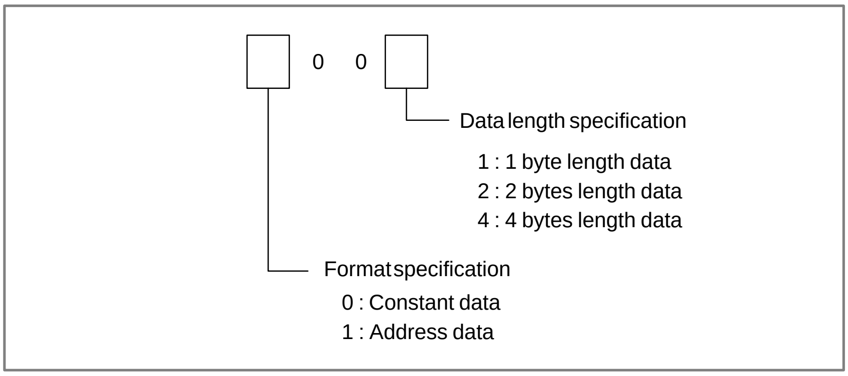
5.38.3 Control Conditions

(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.

5.38.4 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.

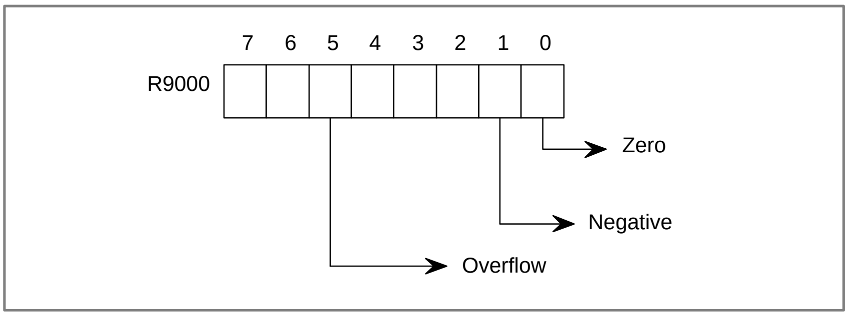
(d) Result output address
Specifies the address to contain the result of operation.

5.38.5 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.

5.38.6 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 DIV (DIVISION)

5.39.1 Function

Divides BCD two–or four–digit data. Remainders are discarded.

5.39.2 Format

Fig.5.39.2 shows the expression format and Table 5.39.2 shows the coding format.

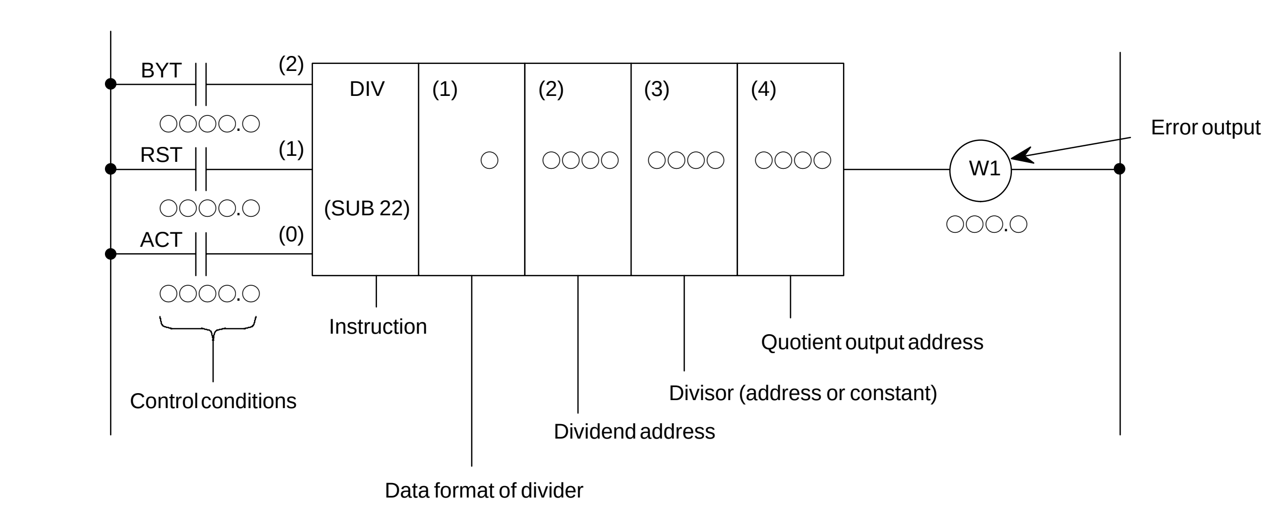


Fig. 5.39.2 DIV instruction format

Table 5.39.2 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

5.39.3 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.

5.39.4 Divisor Data Format Designation

0 : Specifies divisor data by constant.
1 : Specifies divisor data by address.

5.39.5 Dividend Address

Sets the address storing the dividend.

5.39.6 Divisor (Address)

Addressing of the divisor depends on 5.39.4.

5.39.7 Quotient Output Address

Sets the address to which the quotient is output.

5.39.8 Error Output

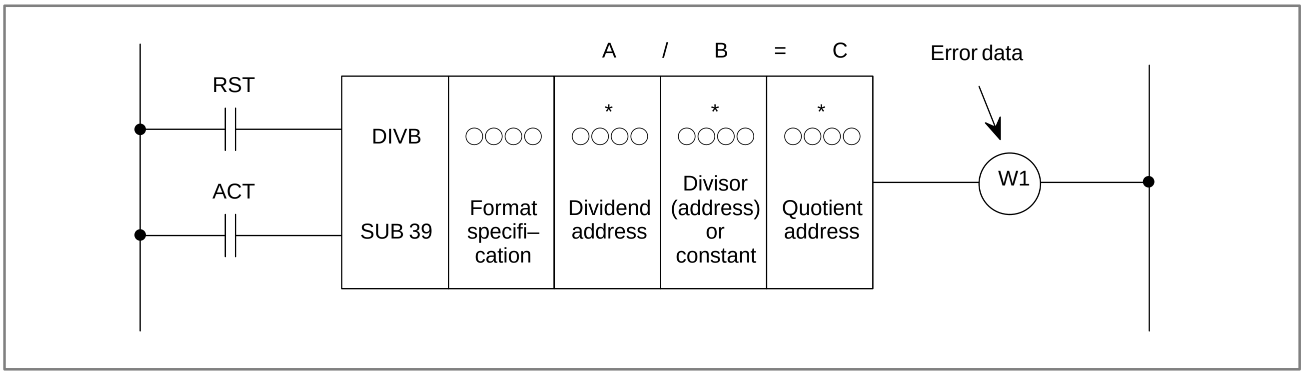
W1=1 is set to indicate an error if the divider is 0.

5.40 DIVB (BINARY DIVISION)

5.40.1 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).

5.40.2 Format



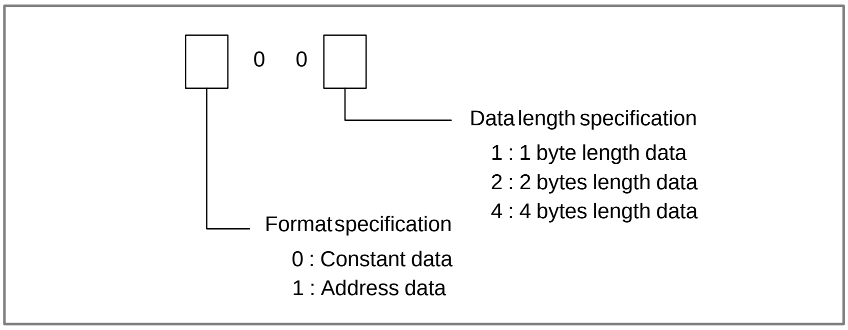
5.40.3 Control Conditions

(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.

5.40.4 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.

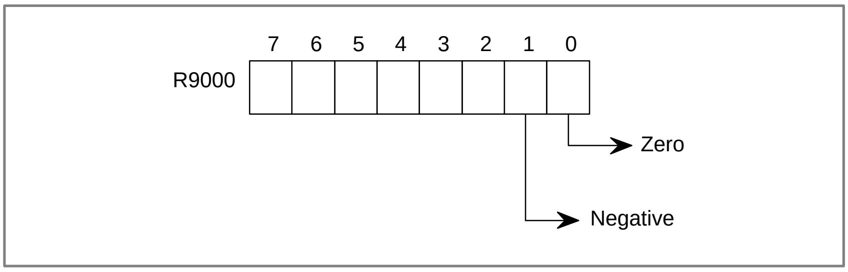
(d) Result output address
Specified the address to contain the result of operation.

5.40.5 Error Putput (W1)

W1=0 : Operation correct
W1=1 : Operation incorrect
W1 goes on (W1=1) if the divisor is 0.

5.40.6 Operation Output Register (R9000)

This register is set with data on operation. If register bit is on, they signify the following operation data:



5.40.7 Remainder Output Address

Depending on its length, the remainder is stored in one or more of registers R9002 to R9005.

5.41 NUME (DEFINITION OF CONSTANT)

5.41.1 Function

Defines constants, when required. In this case, constants are defined with this instructions.

5.41.2 Format

Fig.5.41.2 shows the expression format and Table 5.41.2 shows the coding format.

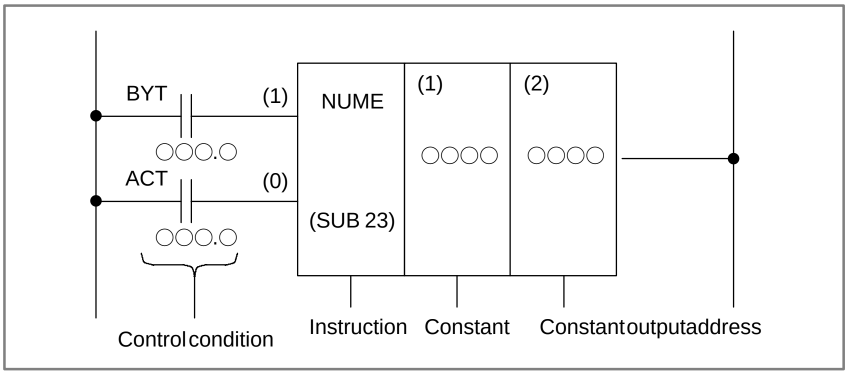


Fig. 5.41.2 NUME instruction format

Table 5.41.2 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

5.41.3 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.

5.41.4 Constant

Sets the constant as the number of digits specified in Item (a) in Subsec. 5.41.3.

5.41.5 Constant Output Address

Sets the address to which the constant defined in Subsec. 5.41.4 is output.

5.42 NUMEB (DEFINITION OF BINARY CONSTANTS)

5.42.1 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–SA5/SB5/SB6/SB7 for Series 16*i*/160*i*/18*i*/180*i*/21*i*/210*i*, Power Mate *i* and PMC–NB6 for Series 15*i*/150*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”.

5.42.2 Format

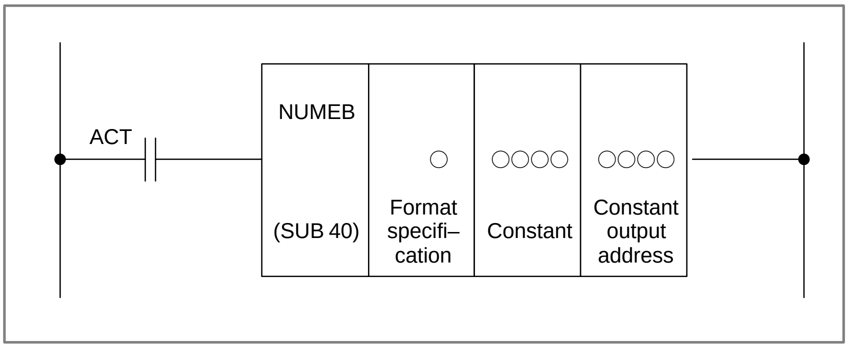


Fig. 5.42.2 (a) NUMBER instruction format

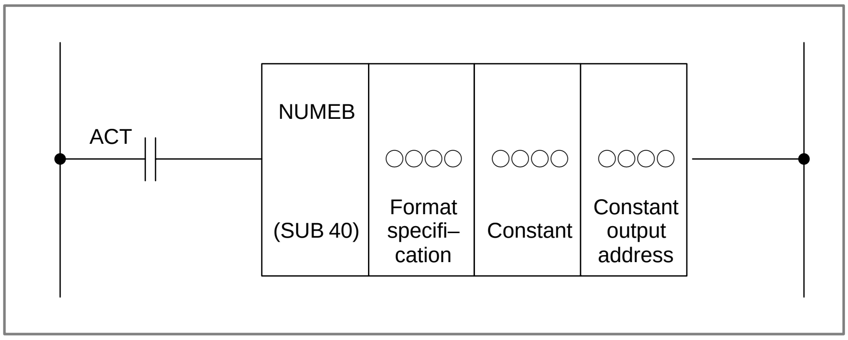


Fig. 5.42.2 (b) Expression format of NUMEB (extended specification) (only for PMC–SA5/SB5/SB6/SB7 for Series 16*i*/160*i*/18*i*/180*i* /21*i*/210*i* Power Mate *i* and PMC–NB6 for Series 15*i*/150*i*)

5.42.3 Control Conditions

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

5.42.4 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–SA5/SB5/SB6/SB7 for Series 16*i*/160*i*/18*i*/180*i*/21*i*/210*i* Power Mate *i* and PMC–NB6 for Series 15*i*/150*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–SA5/SB5/SB6/SB7 for Series 16*i*/160*i*/18*i*/180*i*/21*i*/210*i* Power Mate *i* and PMC–NB6 for Series 15*i*/150*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”.

5.43 DISP (MESSAGE DISPLAY) (PMC–SB/SB2/SB3/SB4/SB5/SB6/SC/SC3/SC4 ONLY)

5.43.1 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.

5.43.2 Format

Fig.5.43.2 shows the instruction format and Table 5.43.2 shows the coding format.

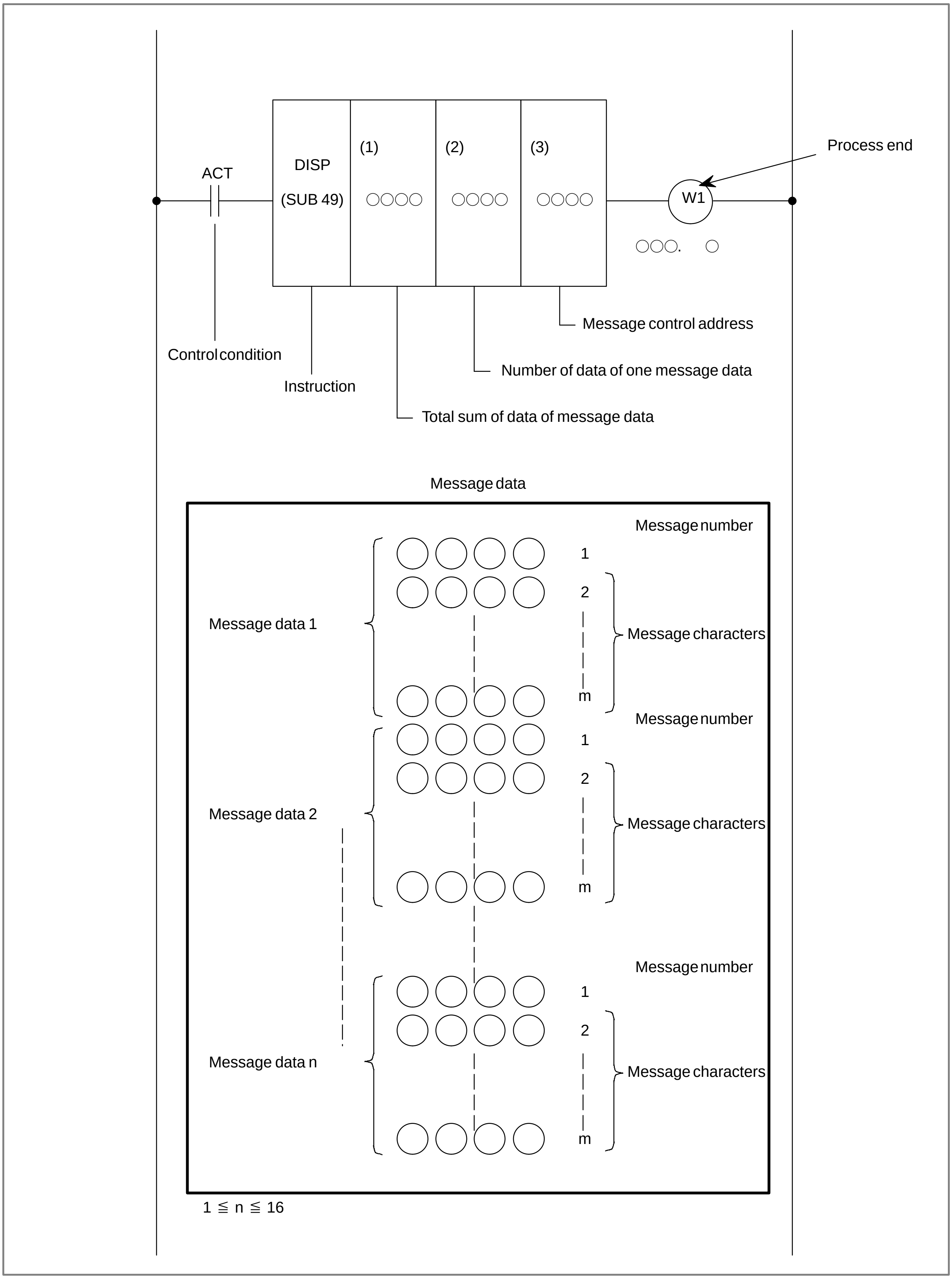


Fig. 5.43.2 DISP instruction format

Table 5.43.2 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

5.43.3 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.

5.43.4 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 5.43.7 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.

5.43.5 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.5.43.7) 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 5.43.6 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 5.42.10 below.

5.43.6 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.

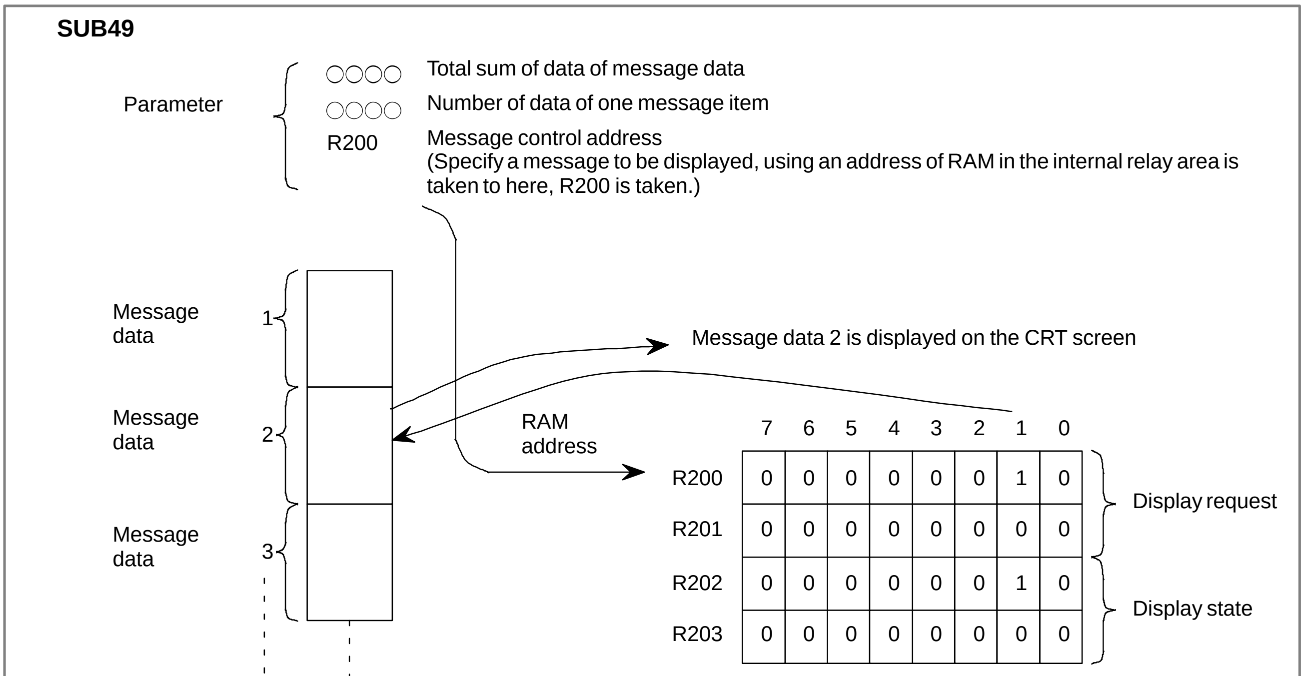
Table 5.43.6 Correspondince between characters and specified numbers

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

5.43.7 Parameters and Message Data

The parameters and message data used by this functional instruction are as follows.



	7	6	5	4	3	2	1	0
R200	0	0	0	0	0	0	1	0
R201	0	0	0	0	0	0	0	0
R202	0	0	0	0	0	0	1	0
R203	0	0	0	0	0	0	0	0

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.5.43.7.

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.5.43.7.

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. 5.43.7 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.

5.43.8 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.

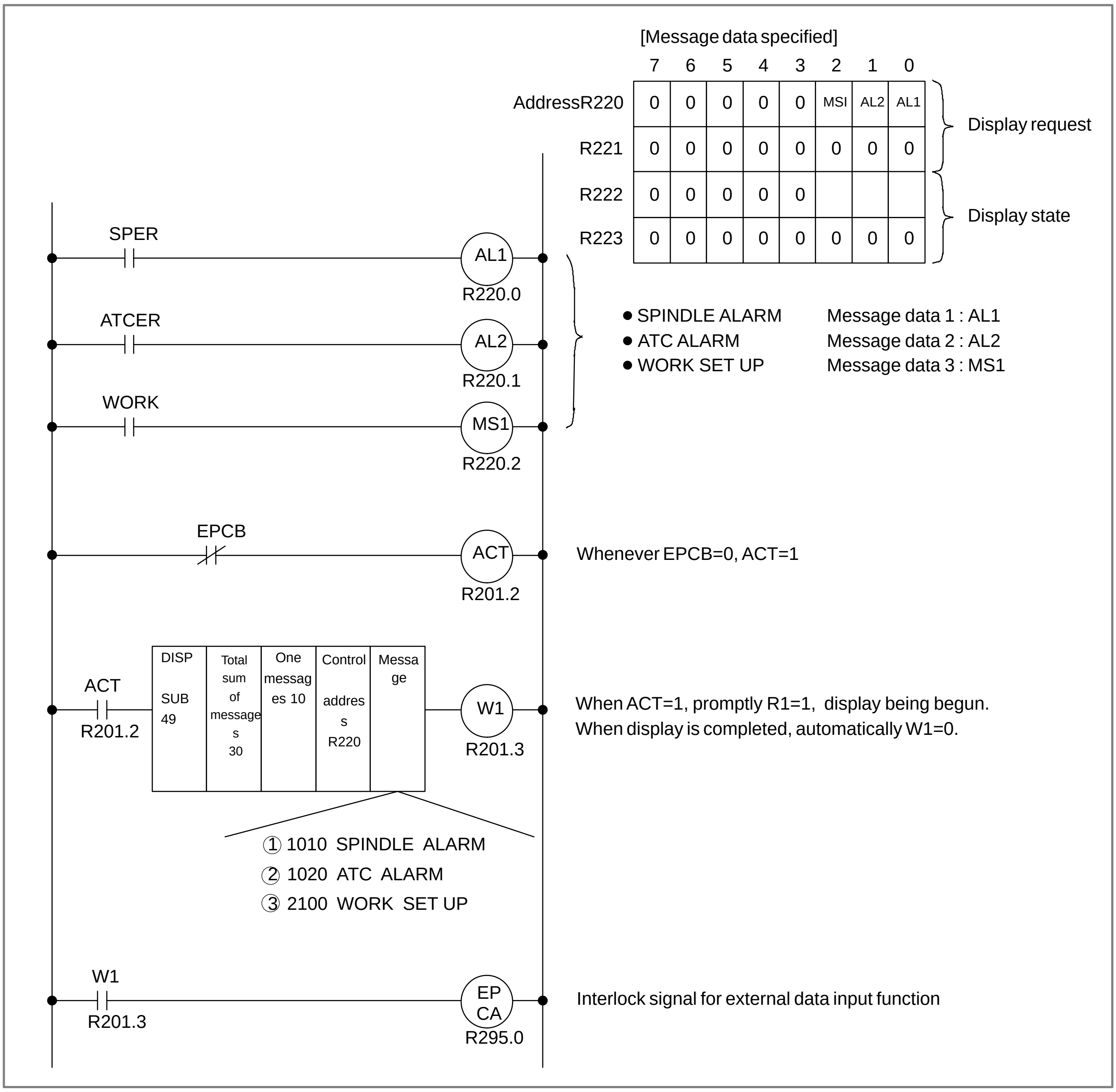
(b) Using two or more DISP instructions

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 5.42.1.

From (a) and (b) above, set ACT to 1 whenever EPCB = 0. Do not set ACT to 1 when EPCB = 1.

5.43.9 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)



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.				
		8380		SP				
		7378		IN				
		6876		DL				
		6932		E_	Message data 1			
		6576		AL	(10 data m=10)			
		6582		AR				
		7700		M				
		(Note1)						
		0000						
		0000						
		1020		Message No.				
		6584		AT				
		6732		C_				
		6576		AL				
		6582		AR	Message data 2 (10 data m=10)			
		7700		M				
		0000						
		0000						
		0000						
		0000						
		2100		Message No.				
		8779		W0				
		8275		RK				
		3200		_	Message data 3			
		0192		タ	(10 data m=10)			
		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

Fig. 5.43.9 (a)

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

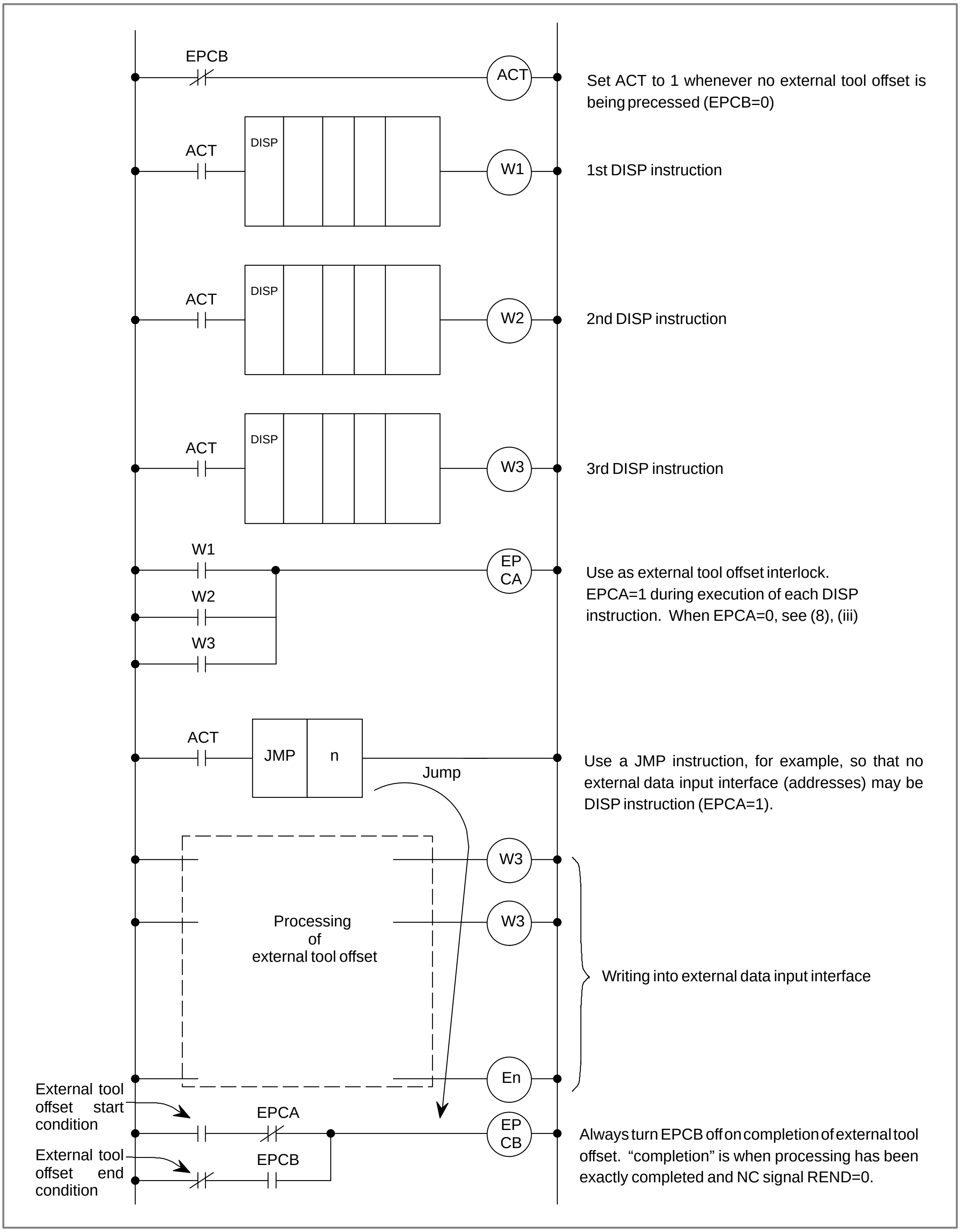


Fig. 5.43.9 (b)

5.43.10 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)
- 990: 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

5.44 DISPB

5.44.1 Function

This instruction displays messages on the CNC 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 5.43.
However, it performs a special additional function, namely, it displays numerical data.
You can program up to 2000 messages. You must use the special message addresses in your program (see Chapter 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 A) and the message data table.
When multiple messages are requested simultaneously, all the messages may not be displayed. The display of messages depends on the number of messages which can be displayed in CNC screen.
In PMC–SB7, the status of the messages which is displayed actually in CNC screen is shown in the message display status memory.
Relation between the message display request memory address and the message data table appears in Table 5.44.1(a).

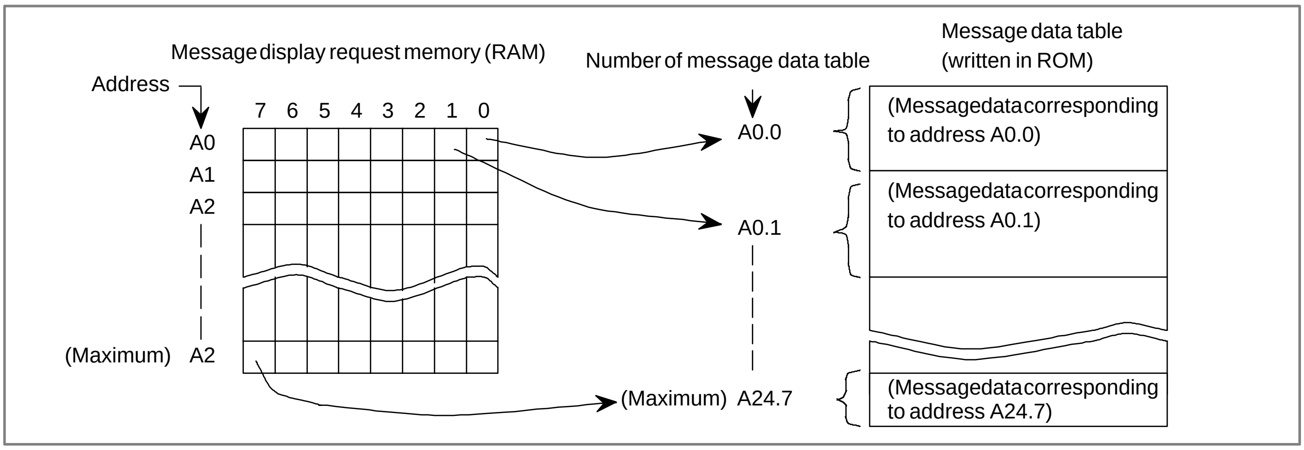


Fig. 5.44.1(a) Message display request memory and message data table

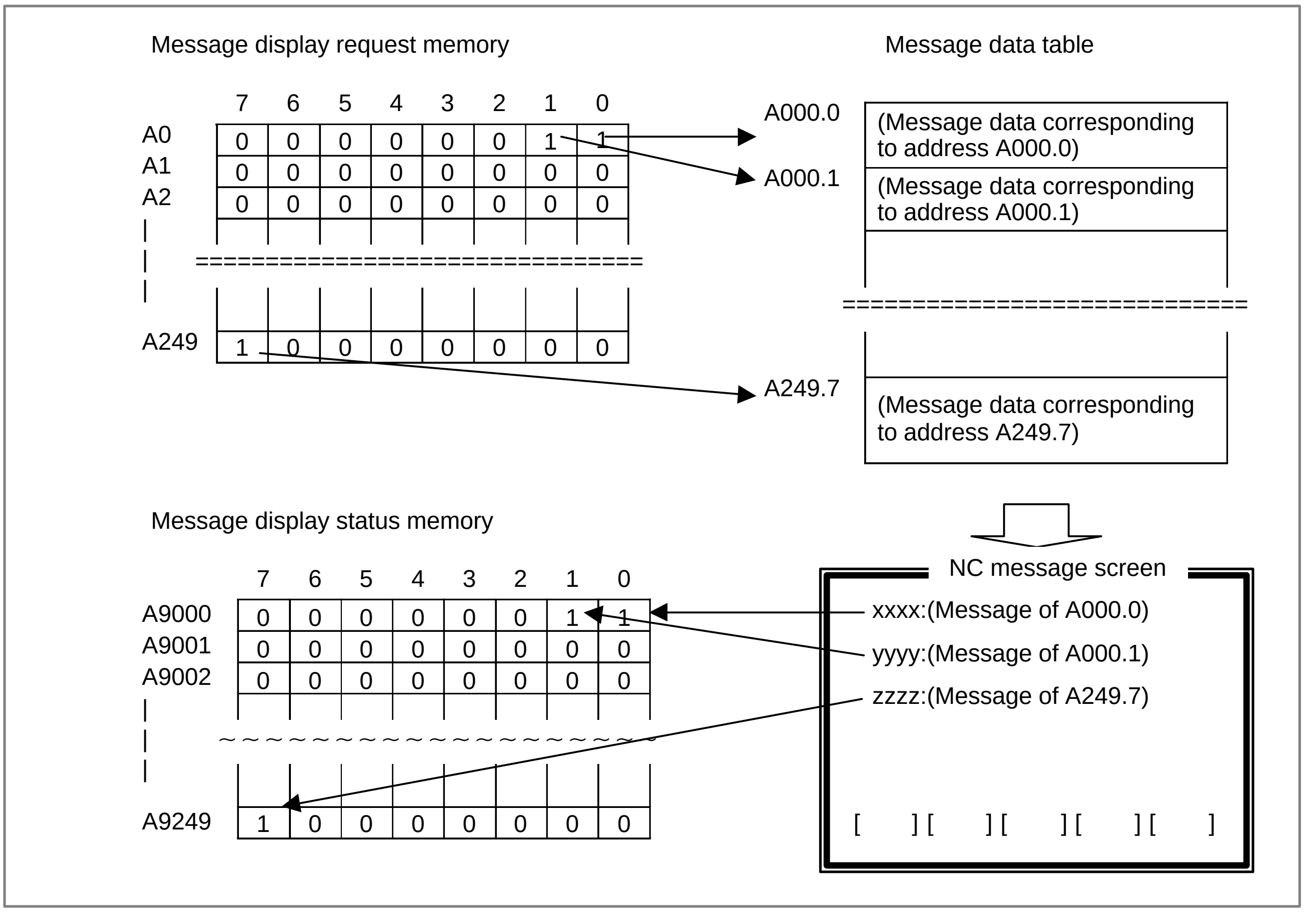


Fig. 5.44.1(b) Message display request memory, message display status memory and Message data table in PMC–SB7

(i) Message display request memory (RAM)
The message display request memory consists bits at A addresses on each PMC model. One bit corresponds to one type of message data.
If you want to display a message on the CRT screen, set the corresponding display request memory 1. Set 0 to erase the message of CNC screen.

(ii) Message display status memory (PMC–SB7)
This memory locates at the address A9000 to A9249 and has 2000bits. Each bit corresponds to a message. While displaying a message in CNC screen, the corresponding bit is set to 1. The ladder can not write on this memory.

(iii) 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 5.44.6.

(iv) 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, FS15B, FS15*i*, 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	Operator message screen	Operator message • Only message data, no message number, is displayed.
2100 to 2999		

- 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	Operator message screen (The 1st path side)	Operator message • Only message data, no message number, is displayed.
2100 to 2999		
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	Operator message screen (The 2nd path side)	Operator message • Only message data, no message number, is displayed.
6100 to 6999		

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.
5 In the FS15*i*, each alarm message must consist of up to 30 characters.

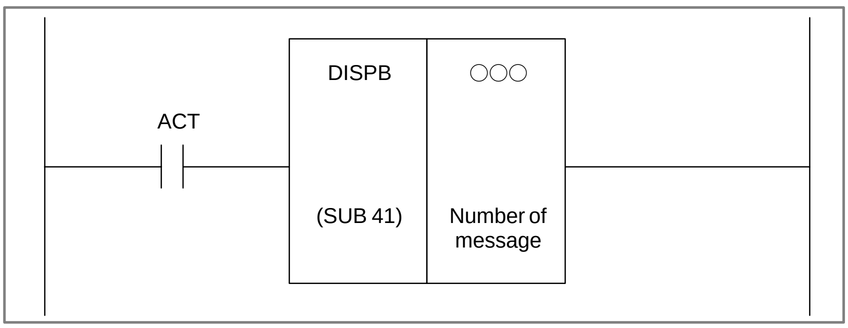
(b) You need not use numerical codes for message data input. Instead, when programming, directly key in the characters making up the messages (from the CRT/MDI keyboard). For the characters that CRT/MDI does not provide for, you must enter these characters by numerical data with special symbols "@". For details, refer to Subsec. 5.44.6).

(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.

5.44.2 Format



5.44.3 Conditions

ACT=0 : Do not display messages on the CRT.
ACT=1 : Display the messages on the CRT.

5.44.4 Parameters

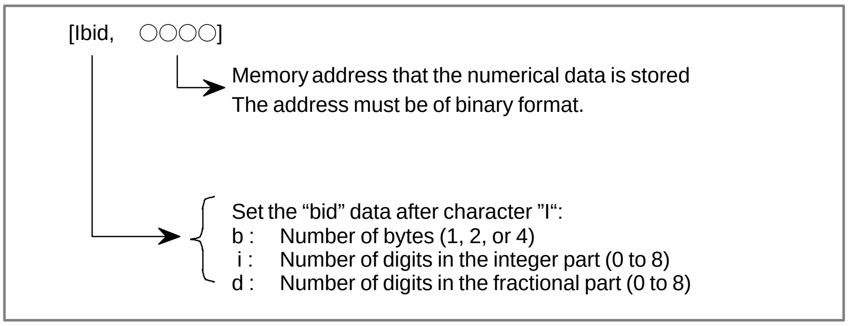
(a) Number of messages
Specifies the total number of messages (up to 200).

Function	PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	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 2000	1 to 200	1 to 200	1 to 1000	1 to 200	1 to 1000	1 to 1000

5.44.5 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



NOTE
1 Sum of integer part digits and fractional part digits must be within 8.
2 Blank is displayed for digits exceeding 8 digits.
3 Do not use any space between the brackets, [ ].

(b) Example
The following message includes 3 digits tool number at the spindle and the offset data (○.○○) for this tool. And these data are contained in memory address of 2bytes:
SPINDLE TOOL No. = [I230, □□□□]
OFFSET DATA = [I212, ΔΔΔΔ]

5.44.6 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 5.44.6). 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
```

> **NOTE**
> Spaces are used between each numerical code in example to understand easily, but do not use them actually.

(b) Kanji (full–width) characters

○ : Can be used
× : Cannot be used

Power Mate	FS20 FS21A	FS21B	FS0*i*	FS21*i*	FS18A		FS16A			FS16B FS18B		FS16C FS18C		FS16*i* FS18*i*	FS15B		FS15*i*
PA1 PA3	SA1 SA3	SA1 SA3	SA1 SA3	SA1 SA5	SA1 SA3	SA2	SB SB3	SB2	SC SC3	SB3 SB4	SC3 SC4	SB5 SB6	SC3 SC4	SB5 SB6 SB7	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 P. 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

@20 → _ ; 3F → ? ; 4434 → 調 ; 3A3A → 査

NOTE

1 To define @, enter @40...@, where 40 is the code corresponding to @40 . . . . . @
   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 (alphanumeric 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@
5 Spaces are used between each numerical code in example to understand easily, but do not use them actually.

Table 5.44.6 Character code table

	2	3	4	5	A	B	C	D
0	␣ (Space)	0	@	P	~	_ *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

5.44.7 Notes when this Functional Instruction is Used in Subroutine

See I–9.3 for details.

5.44.8 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 SB7	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. See II–4.4.1 for details.

- 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.

Message table

A0.0	English A	(ALARM)
A0.1	English B	(ALARM)
A0.2	English C	(ALARM)
~		~
A10.0	Japanese 1	(OPE)
A10.1	Japanese 2	(OPE)
~		~
A15.0	English 1	(OPE)
A15.1	English 2	(OPE)
A20.0	Italian 1	(OPE)
A20.1	Italian 2	(OPE)
~		~
A25.0	German 1	(OPE)
A25.1	German 2	(OPE)
:	:	
Am.n	:	

When A0.1 is turned on, English B is displayed.

When A10.0 is turned on, German 1 is displayed.

When A10.1 is turned on, German 2 is displayed.

(The message data is shifted by 120 bits).

(c) Notes

The same message number should be assigned to a message in each language that has the same meaning.

Message table

A0.0	1000	English A	(ALARM)
A0.1	1001	English B	(ALARM)
~			~
A10.0	1000	Japanese 1	(OPE)
A10.1	1001	Japanese 2	(OPE)

5.45 EXIN (EXTERNAL DATA INPUT)

5.45.1 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 from PMC to NC directly in your program.
The DISPB instruction uses the interface from PMC to NC provided by the external data input function during display. The DISP instruction prevents the interface signal transferred from the PMC to NC from being changed due to external cutter compensation or others.
You can use the EXIN instruction only when the PMC→NC interface is of BMI (Basic Machine Interface) and optional external data input function is provided with NC.
An 4–byte control data as described below is required for external data input function (option).

In PMC–SA5/SB5/SB6/SB7, 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 (FS16*i*/18*i*/21*i*)**

	#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.

5.45.2 Format

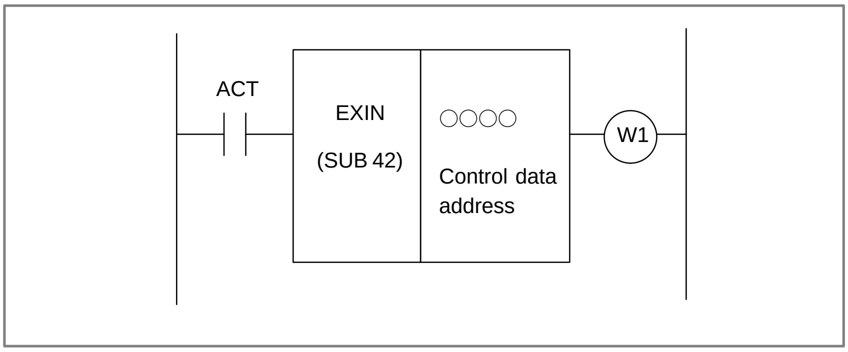


Fig. 5.45.2 EXIN instruction format

5.45.3 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).

5.45.4 Parameter

(a) Control data (except PMC–NB/NB2/NB6)

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/SB7, 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

Address	Contents
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.)

Address	Contents
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) Control data (PMC–NB/NB2/NB6)
When the external input function is used.
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.

Address	Data
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

<u>When the external output function is used.</u>
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.

Address	Data	Address	Data
			to to
CTL+0	EISTB, EOREND etc	CTL+8	EOSTB, EIREND etc
+1	EIA0 to EIA7	+9	EOA0 to EOA7
+2	EID32 to EID39	+10	EOD32 to EOD39
+3	EID40 to EID47	+11	EOD40 to EOD47
+4	EID0 to EID7	+12	EOD0 to EOD7
+5	EID8 to EID15	+13	EOD8 to EOD15
+6	EID16 to EID23	+14	EOD16 to EOD23
+7	EID25 to EID31	+15	EOD25 to EOD31
	to to		

NOTE
Refer to the following manuals in detail of BMI interface.
“FANUC Series 15–MODEL B Connection Manual (BMI interface)”
“FANUC Series 15*i*/150*i*–MODEL A Connection Manual (Function)”

5.45.5 End of Transfer (W1)

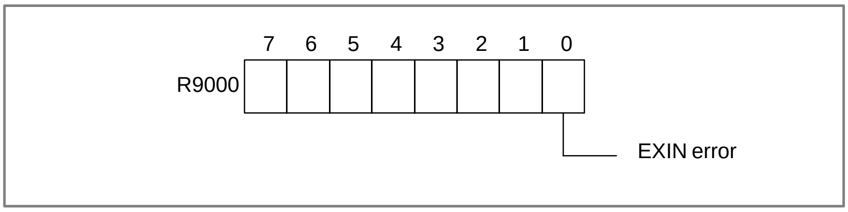
This indicates end of transfer of external data. This transfer end condition shows the end of a series of external data input sequence. This functional instruction executes a series of transfer sequence, and finally sets ESTB = 0 in the PMC → NC interface. As a result, W1 is set to 1 (W1 = 1) after confirming 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.

5.45.6 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.)

5.45.7 Notes when this Functional Instruction is Used in Subroutine

See I–9.3 for details.

5.46 WINDR (READING CNC WINDOW DATA)

5.46.1 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.

5.46.2 Format

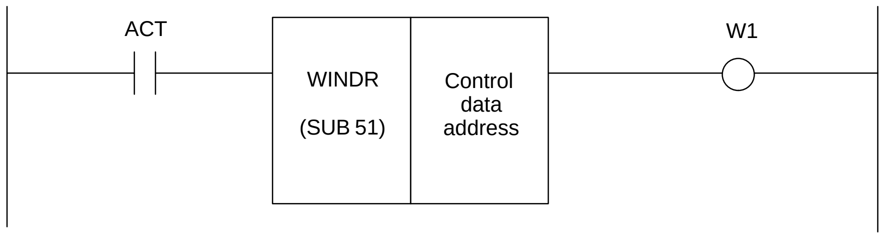


Fig. 5.46.2

5.46.3 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).

5.46.4 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	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 Appendixies B to F 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.

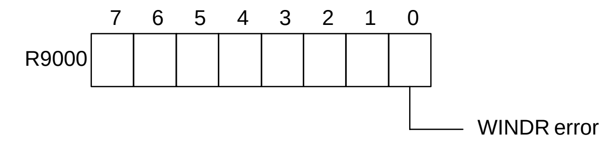
5.46.6 Reading Completion (W1)

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).

5.46.7 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 reading completion is set (W1=1). Details of the error are output to the completion code (CTL+2) in the control data area. See Appendixies B to F 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 “I–9.3 NOTE FOR SUBROUTINES WHEN YOU USE SUBROUTINES” When you use the function of a high–speed response, there is no limitation.

5.47 WINDW (WRITING CNC WINDOW DATA)

5.47.1 Function

This function writes various data items via the window between the PMC and the CNC.

5.47.2 Format

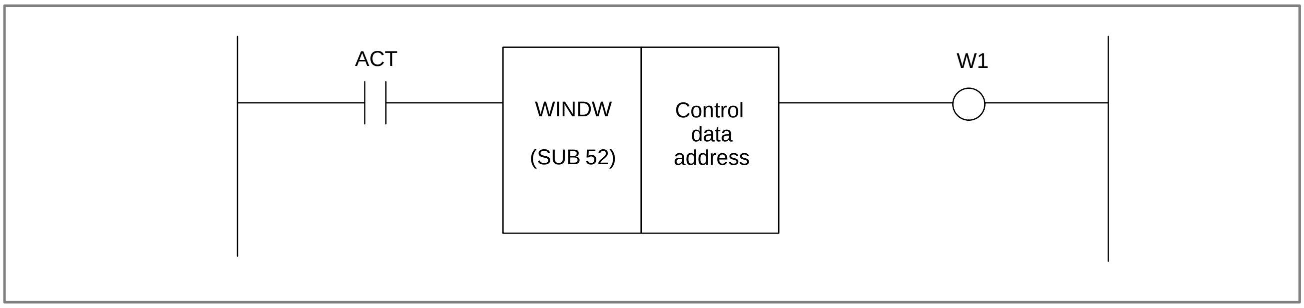


Fig. 5.47.2

5.47.3 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).

5.47.4 Parameter

(a) Control data address

The PMC byte address is used to specify the area where control data is stored.

5.47.5 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 Appendixies B to F 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.

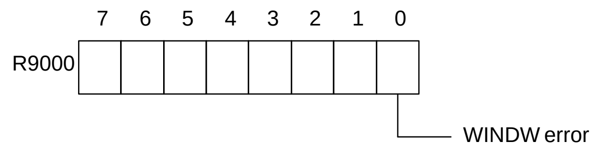
5.47.6 Writing Completion (W1)

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).

5.47.7 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 Appendixies B to F WINDOW FUNCTION DESCRIPTION.



5.47.8 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 "I–9.3 NOTE FOR SUBROUTINES WHEN YOU USE SUBROUTINES"

5.48 ARBITRARY FUNCTIONAL INSTRUCTIONS

5.48.1 FNC 90 to 97 (Arbitrary Functional Instructions) (Only for PMC–RC/RC3/NB/NB2)

5.48.1.1 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.

5.48.1.2 Format

Fig.5.48.1.2 shows the notation format. Table 5.48.1.2 shows the coding format.

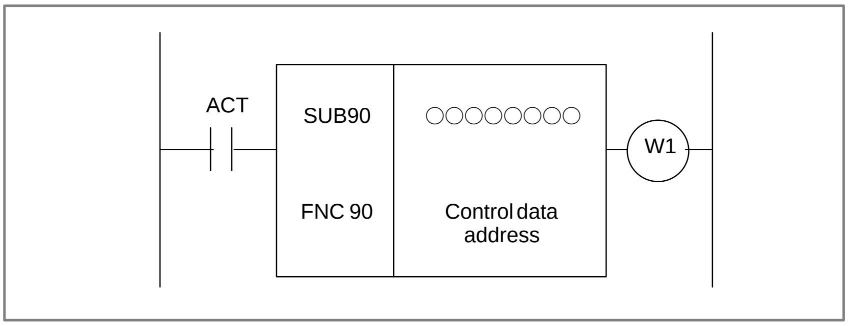


Fig. 5.48.1.2 FUNC 90 notation format

Table 5.48.1.2 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

5.48.1.3 Control Condition

(a) Execution command (ACT)
This is used as the start condition of an arbitrary functional instruction.

5.48.1.4 Parameter

(a) Control data address
Specifies the first address in the control data area.

5.48.1.5 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				

5.48.1.6 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.

5.48.2 Creating an Arbitrary Function

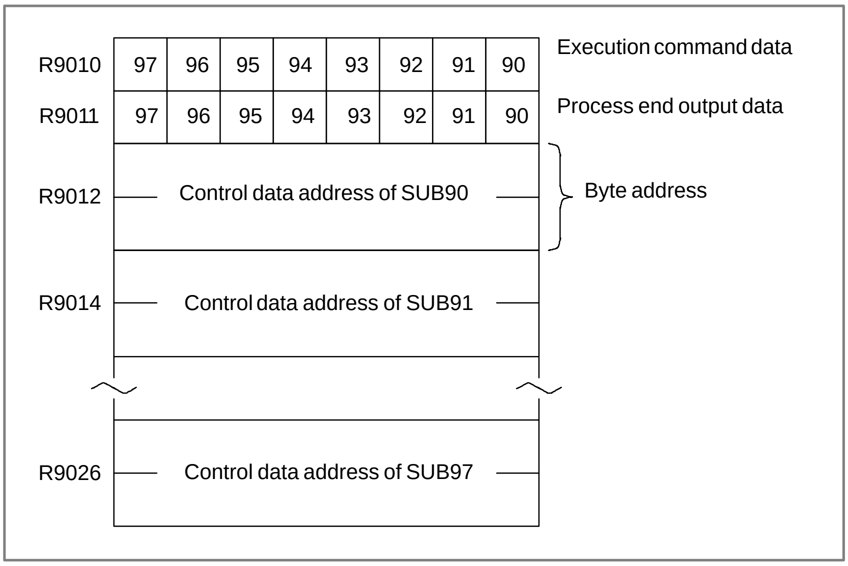
5.48.2.1 Arbitrary Functional Instruction and Interface

(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.

5.48.2.2 Use of the R Field



5.48.2.3 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.

5.49 MMCWR (READING MMC WINDOW DATA) (OTHER THAN PMC–PA1/PA3)

5.49.1 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.

5.49.2 Format

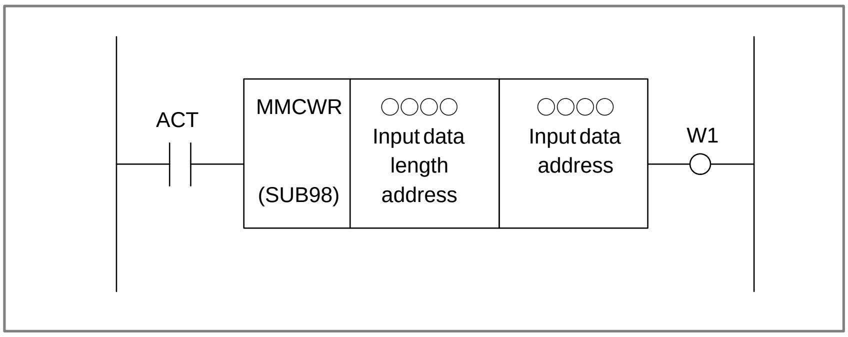


Fig. 5.49.2 MMCWR instrument format

Table 5.49.2 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

5.49.3 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).

5.49.4 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.

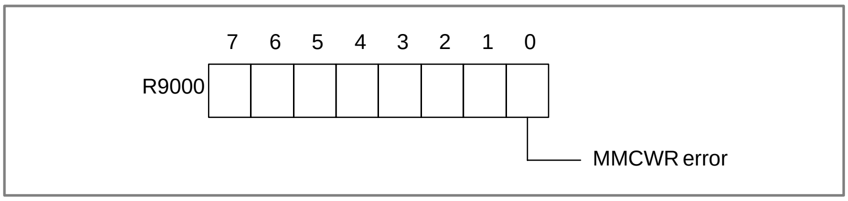
5.49.5 Processing Completion (W1)

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 from 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 stored in the input data area.



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. The length of data actually transferred exceeded the specified value.)
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. I–9.3.

5.50 MMCWW (WRITING MMC WINDOW DATA) (OTHER THAN PMC–PA1/PA3)

5.50.1 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.

5.50.2 Format

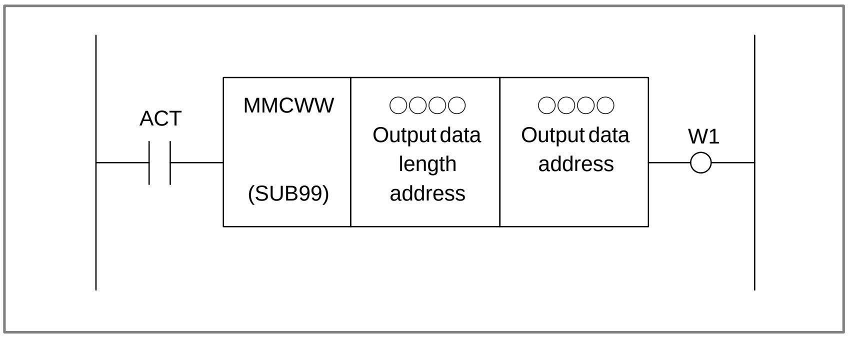


Fig. 5.50.2 MMCWW instruction format

Table 5.50.2 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

5.50.3 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.

5.50.4 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.

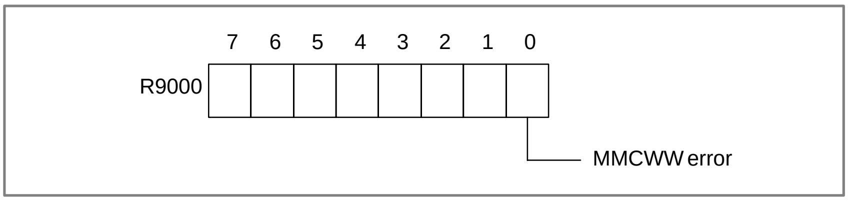
5.50.5 Processing Completion (W1)

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.

5.50.6 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.



5.50.7 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)

5.50.8 Notes when this Functional Instruction is Used in Subroutine

Refer to Sec. I–9.3.

5.51 MOVB (TRANSFER OF 1 BYTE)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.51.1 Function

The MOVB instruction transfers 1–byte data from a specified source address to a specified destination address.

5.51.2 Format

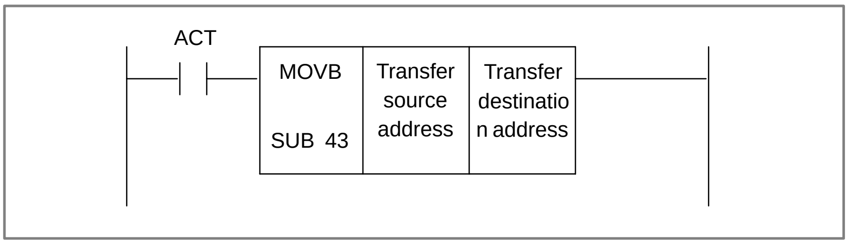


Fig. 5.51.2 MOVB instruction format

5.51.3 Control Conditions

(a) Execution specification
ACT=0 : No data is transferred.
ACT=1 : One–byte data is transferred.

5.52 MOVW (TRANSFER OF 2 BYTES)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.52.1 Function

The MOVW instruction transfers 2–byte data from a specified source address to a specified destination address.

5.52.2 Format

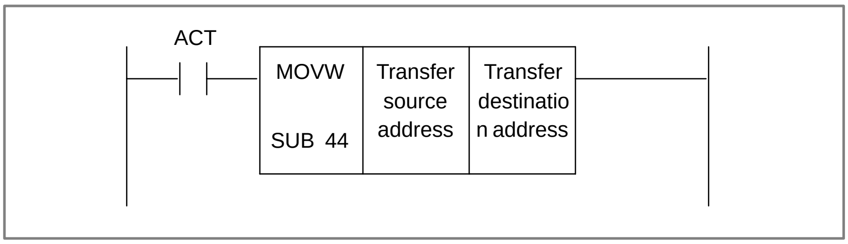


Fig. 5.52.2 MOVB instruction format

5.52.3 Control Conditions

(a) Execution specification
ACT=0 : No data is transferred.
ACT=1 : Two–byte data is transferred.

5.53 MOVN (TRANSFER OF AN ARBITRARY NUMBER OF BYTES)

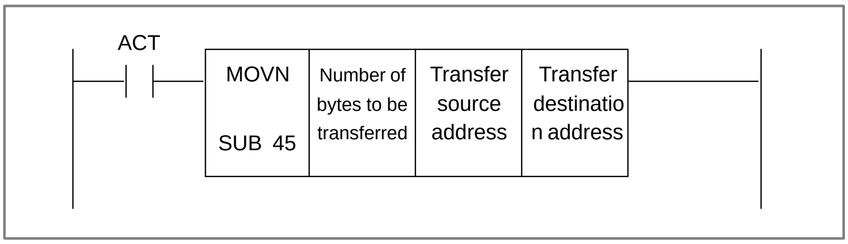
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.53.1 Function

The MOVN instruction transfers data consisting of an arbitrary number of bytes from a specified source address to a specified destination address.

5.53.2 Format



5.53.3 Control Conditions

(a) Execution specification
ACT=0 : No data is transferred.
ACT=1 : A specified number of bytes are transferred.

5.53.4 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 9999 can be specified.

> **NOTE**
> With the PMC–PA3/SA3/SB3/SB4/SC3/SC4/NB/NB2, a value from 1 to 200 must be set as the number of transferred bytes.

5.54 MOVD (TRANSFER OF 4 BYTES)

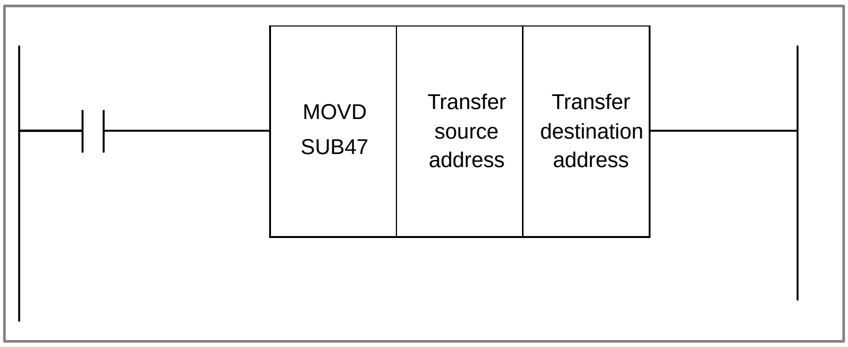
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	×	×	×	×	×	×	×	×	×	×	×	○	×	×	×	×	×	×

5.54.1 Function

The MOVD instruction transfers 4 bytes data from a specified source address to a specified destination address.

5.54.2 Format



5.54.3 Control Conditions

ACT=0: No data is transferred.
ACT=1: A specified number of bytes are transferred.

5.55 DIFU (RISING EDGE DETECTION)

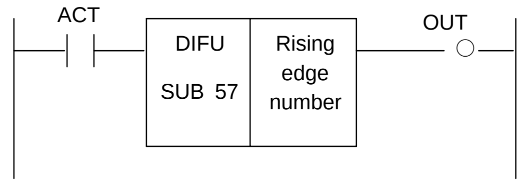
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.55.1 Function

The DIFU instruction sets the output signal to 1 for one scanning cycle on a rising edge of the input signal.

5.55.2 Format



5.55.3 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.

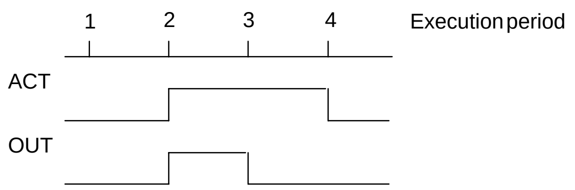
5.55.4 Parameters

(a) Rising edge number

Model	PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	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 1000	–	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.

5.55.5 Operation



5.56 DIFD (FALLING EDGE DETECTION)

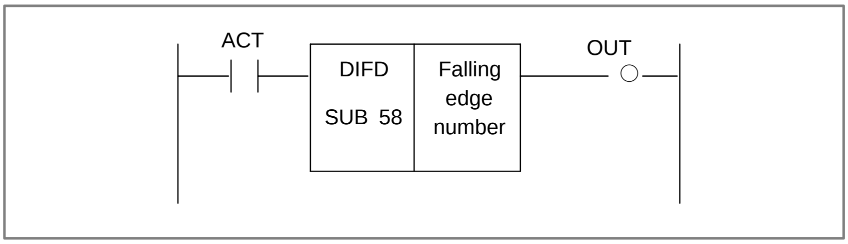
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.56.1 Function

The DIFD instruction set the output signal to 1 for one scanning period on a falling edge of the input signal.

5.56.2 Format



5.56.3 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.

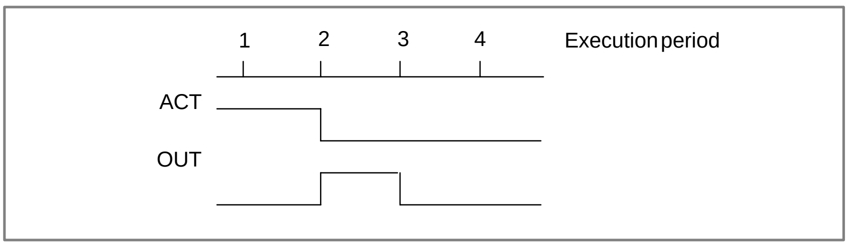
5.56.4 Parameters

(a) Falling edge number

Model	PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	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 1000	–	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.

5.56.5 Operation



5.57 EOR (EXCLUSIVE OR)

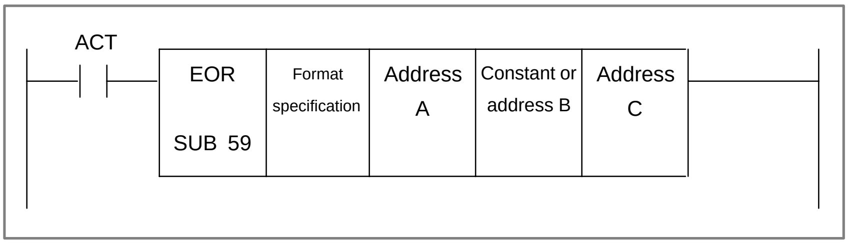
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.57.1 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.

5.57.2 Format

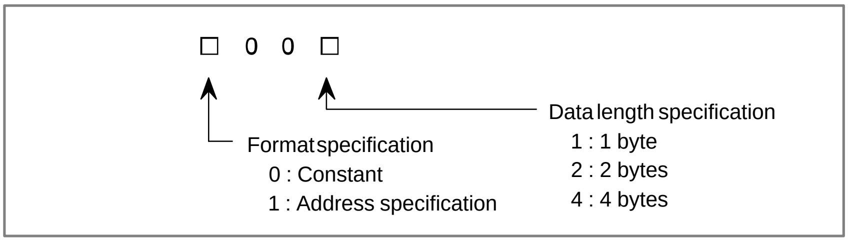


5.57.3 Control Conditions

(a) Input signal
ACT=0 : The EOR instruction is not executed.
ACT=1 : The EOR 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 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.

5.57.5 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 exclusive OR operation is as follows:

Address C	1	0	1	1	0	1	1	0

5.58 LOGICAL AND

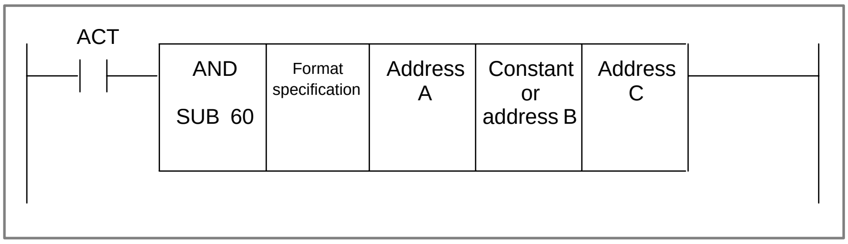
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.58.1 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.

5.58.2 Format

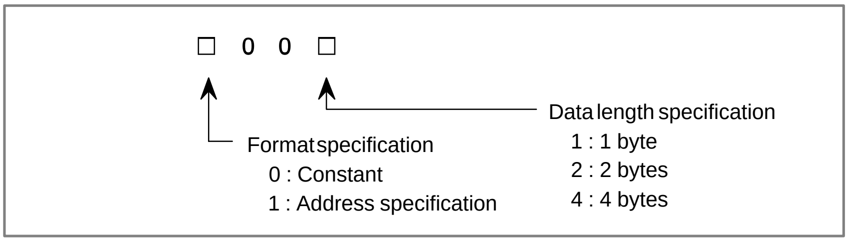


5.58.3 Control Conditions

(a) Input signal
ACT=0 : The AND instruction is not executed.
ACT=1 : The AND instruction is executed.

5.58.4 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.

5.58.5 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

5.59 LOGICAL OR

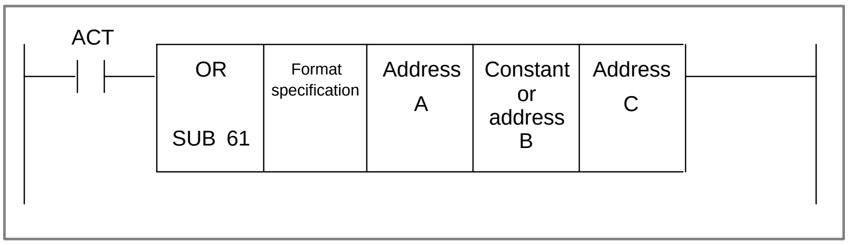
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.59.1 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.

5.59.2 Format

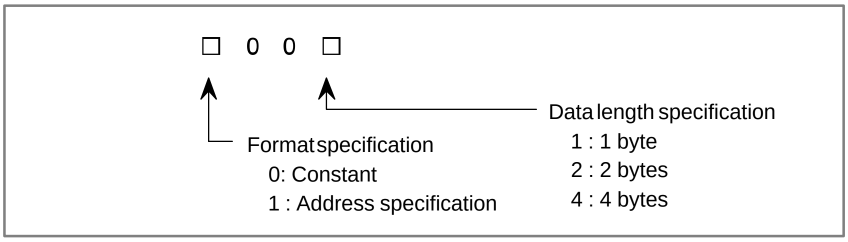


5.59.3 Control Conditions

(a) Input signal
ACT=0 : The OR instruction is not executed.
ACT=1 : The OR instruction is executed.

5.59.4 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.

5.59.5 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

5.60 NOT (LOGICAL NOT)

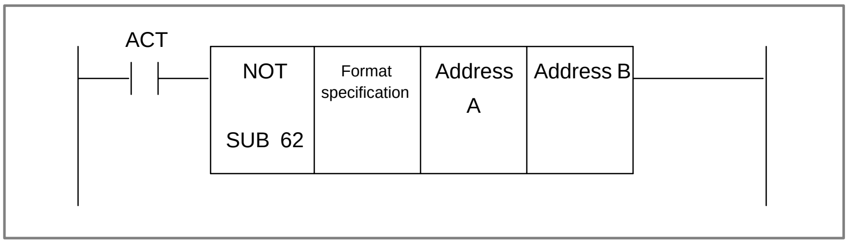
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.60.1 Function

The NOT instruction inverts each bit of the contents of address A, and stores the result at address B.

5.60.2 Format

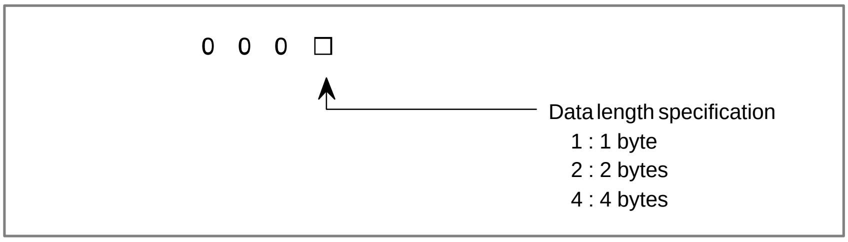


5.60.3 Control Conditions

(a) Input signal
ACT=0 : The NOT instruction is not executed.
ACT=1 : The NOT instruction is executed.

5.60.4 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.

5.60.5 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

5.61 MMC3 R (MMC–III WINDOW DATA READ)

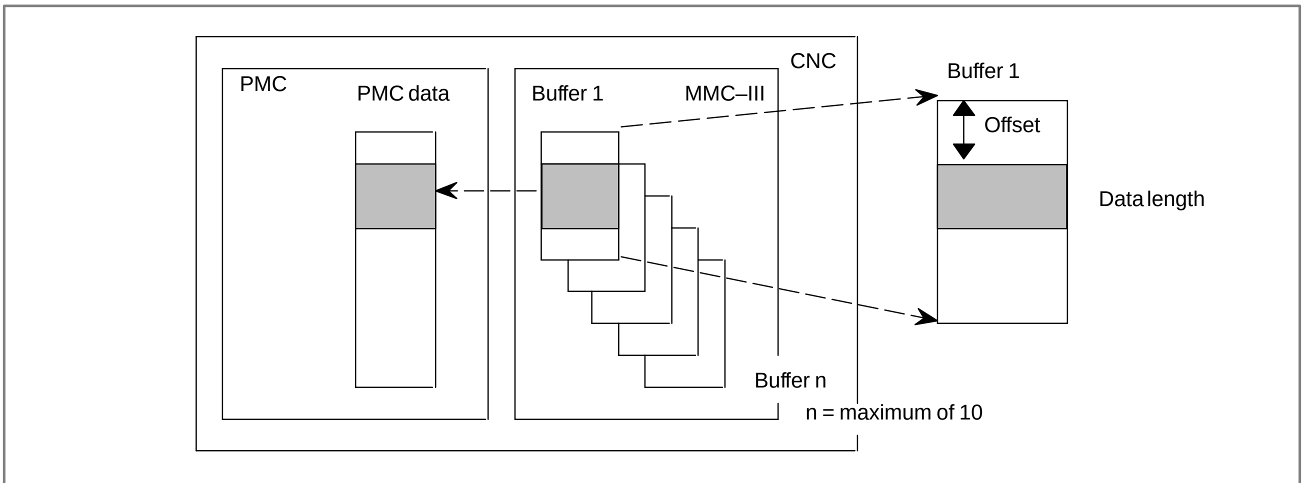
○ : Can be used
Δ : Can be used (with some restrictions)
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	×	Δ	○	○	×	○	○	○	○	×	×	○	○	○	○	○	○	×

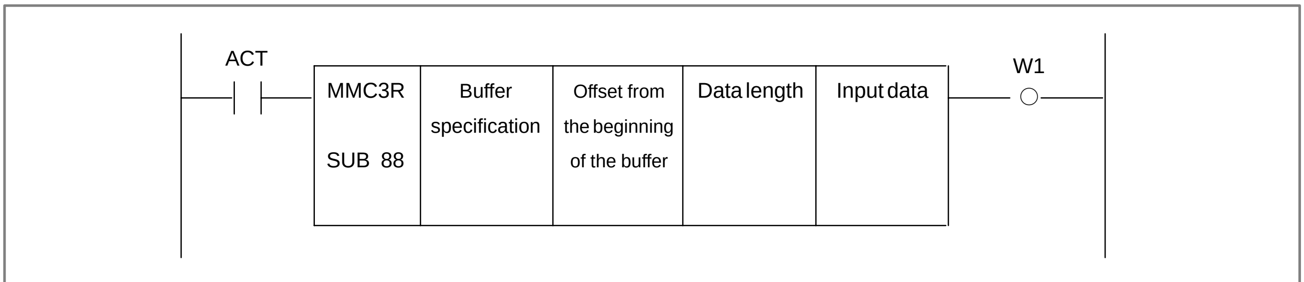
NOTE
This functional instruction cannot be used with SA1 of the Series 16*i*/18*i*/21*i*–MODEL A/B and Series 15*i*.

5.61.1 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.



5.61.2 Format



5.61.3 Control Conditions (ACT)

ACT=0 : The MMC3R instruction is not executed.
ACT=1 : Data is read.

5.61.4 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.

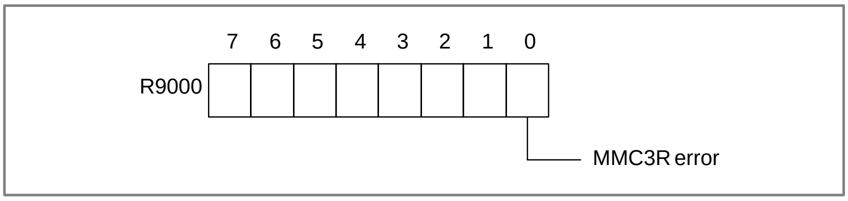
5.61.5 Processing Completion (W1)

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.

5.61.6 Operation Output Register

When W1 indicates the termination of read processing, a termination state is set.



MMC3R=0 : Normal termination
MMC3R=1 : Abnormal termination

5.61.7 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)

5.61.8 Notes when this Functional Instruction is Used in Subroutine

Refer to Sec. I–9.3.

5.62 MMC3W (MMC–III WINDOW DATA WRITE)

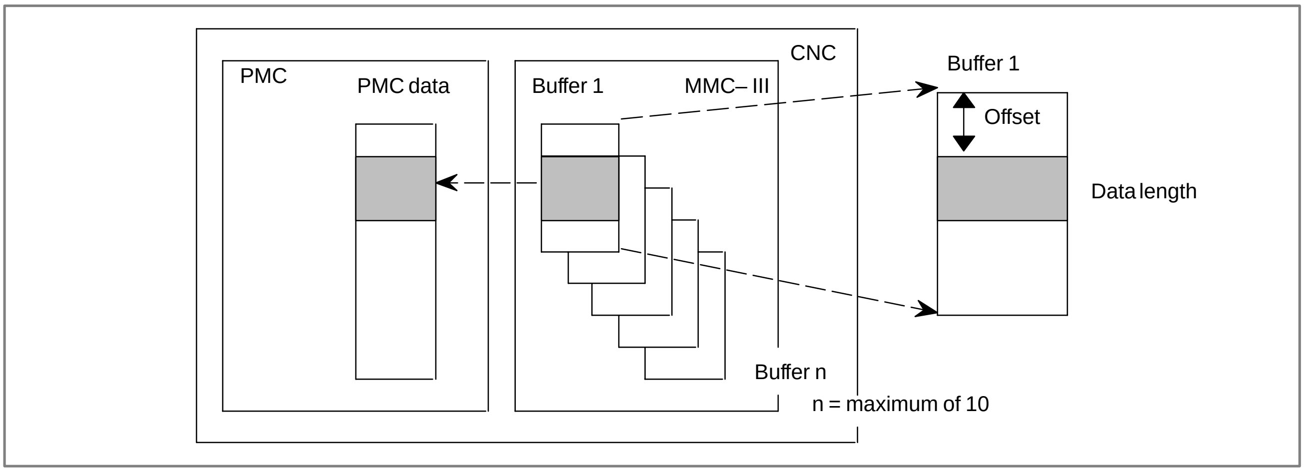
○ : Can be used
Δ : Can be used (with some restrictions)
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	×	Δ	○	○	×	○	○	○	○	×	×	×	○	○	○	○	○	×

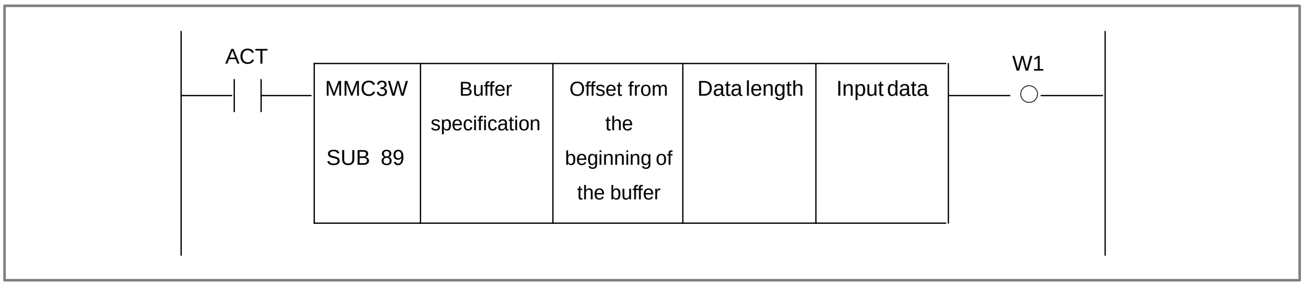
NOTE
This functional instruction cannot be used with SA1 of the Series 16*i*/18*i*/21*i*–MODEL A/B and Series 15*i*.

5.62.1 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.



5.62.2 Format



5.62.3 Control Conditions (ACT)

ACT=0 : The MMC3W instruction is not executed.
ACT=1 : Data is written.

5.62.4 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.

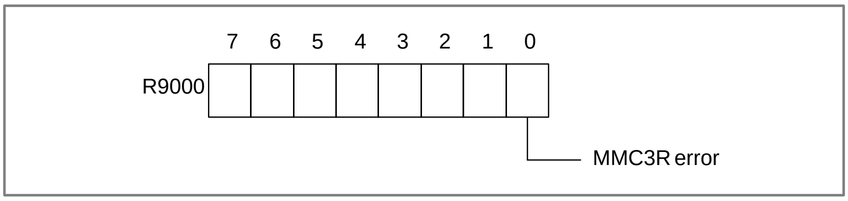
5.62.5 Processing Completion (W1)

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.

5.62.6 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

5.62.7 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 The specified length of data is 0, negative data is specified, or the maximum allowable data length is exceeded.	(W1=1, R9000#0=1)
6 :	The MMC–III is not attached.	(W1=1, R9000#0=1)
3 :	Buffer specification error	(W1=1, R9000#0=1)

5.62.8 Notes when this Functional Instruction is Used in Subroutine

Refer to Sec. I–9.3.

5.63 SPCNT (SPINDLE CONTROL)

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	○	○	○

5.63.1 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.5.63.1, 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.

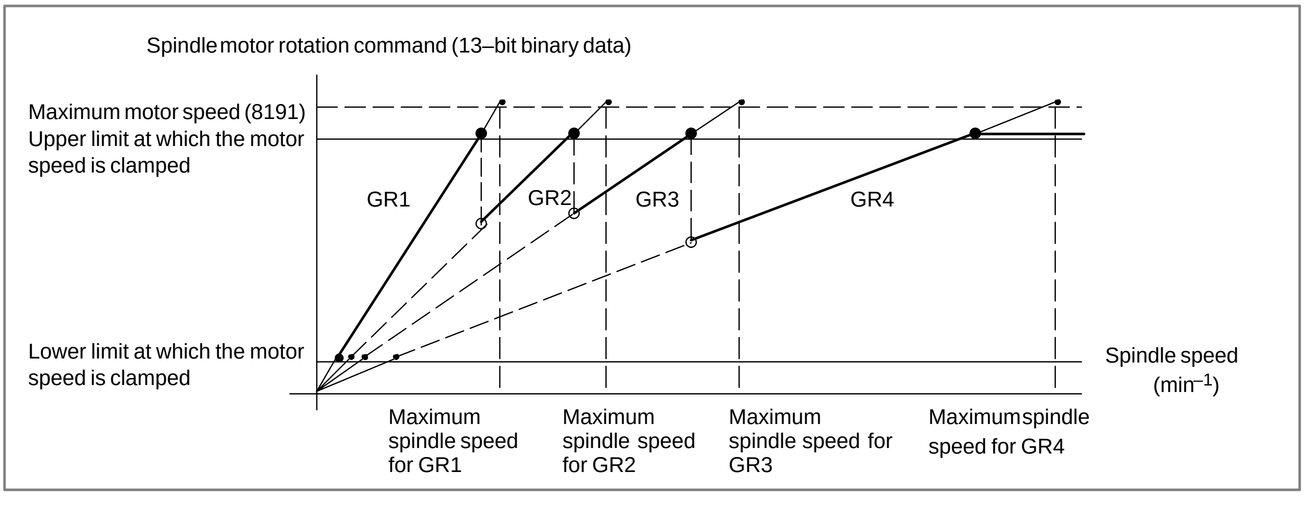
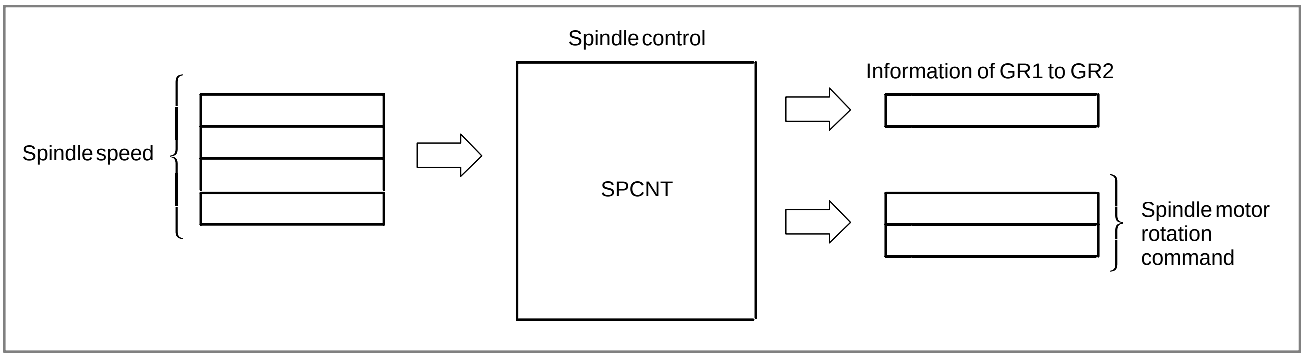


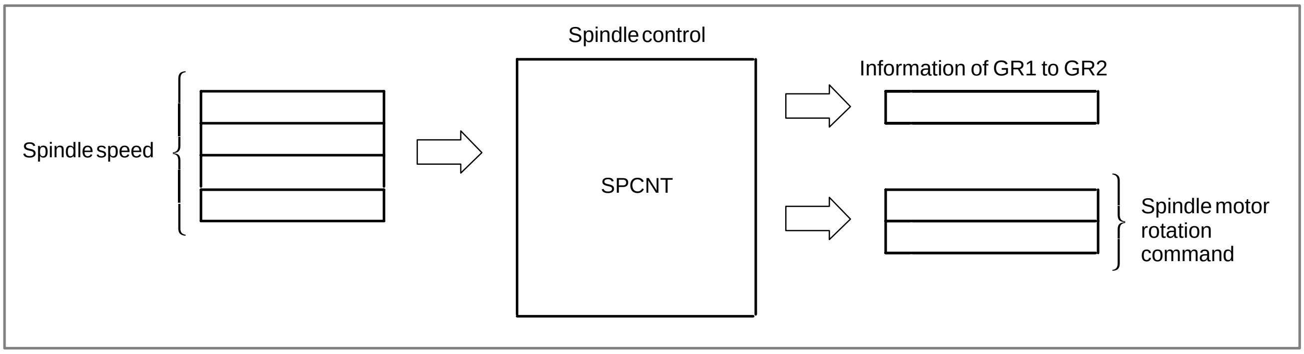
Fig. 5.63.1 Spindle Speeds and Corresponding Spindle Motor Rotation Commands

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.

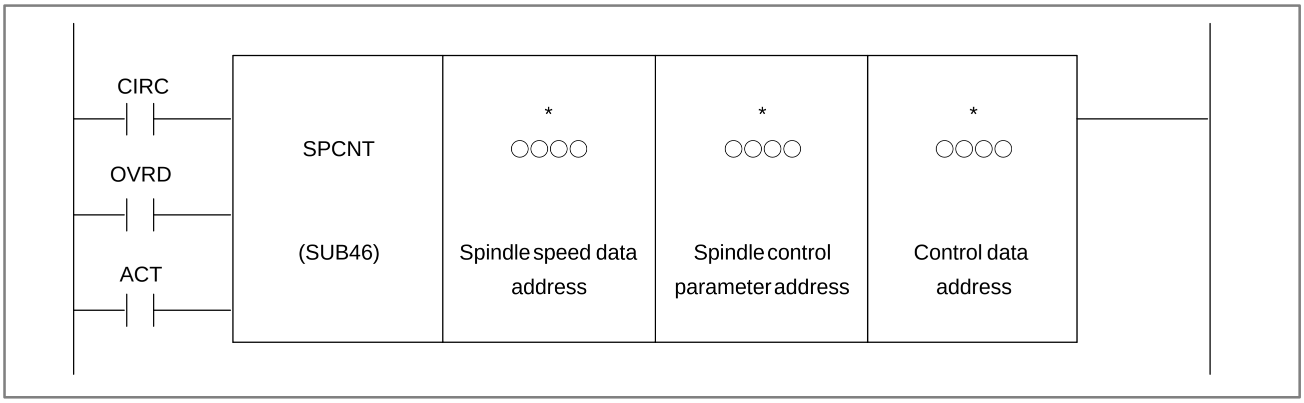


(ii) Spindle control with direct gear selection
When direct gear selection is set, this functional instruction does not perform gear selection. A gear to be used is selected by the sequence program. The target gear must be set at the control data address, which is a parameter of this functional instruction, 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.5.63.1.



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.

5.63.2 Format



5.63.3 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.

5.63.4 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	

(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 (min^{-1}) 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 (min^{-1}) 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 (min^{-1}) 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 (min^{-1}) 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. 5.63.1.

(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

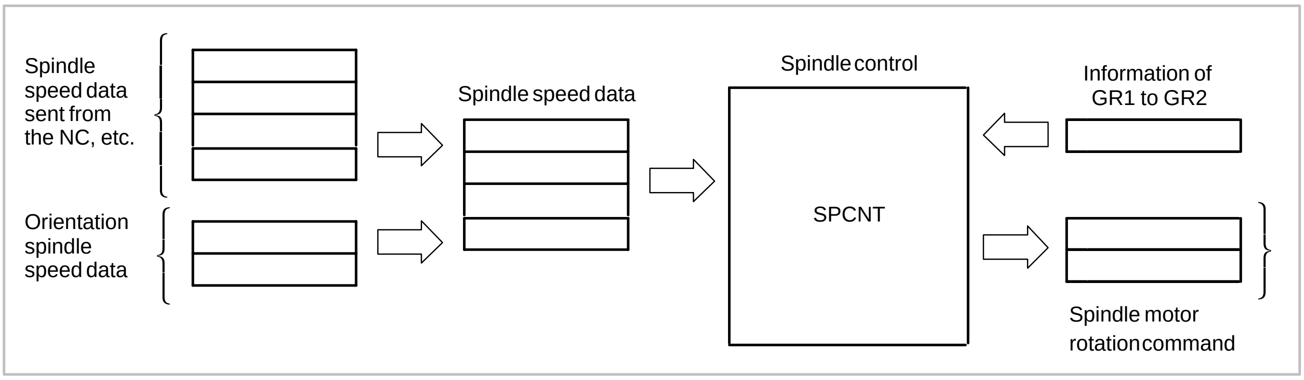
The sequence program must set a spindle override in binary. A spindle override from 0% to 255% can be set in binary.

5.63.5 Use of Spindle Control

Spindle control is primarily used to control the spindle speed during normal cutting. It can, however, also be used to:

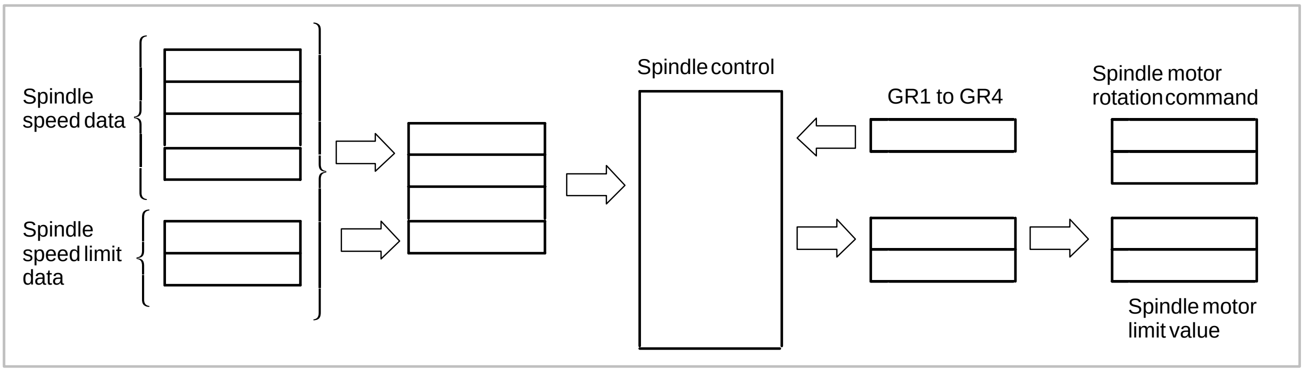
(a) Rotate the spindle motor at a specific speed when the gear is switched
The sequence program can output appropriate 13–bit binary data as a spindle motor rotation command to rotate the spindle motor at a specific speed, without using this functional instruction.

(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).



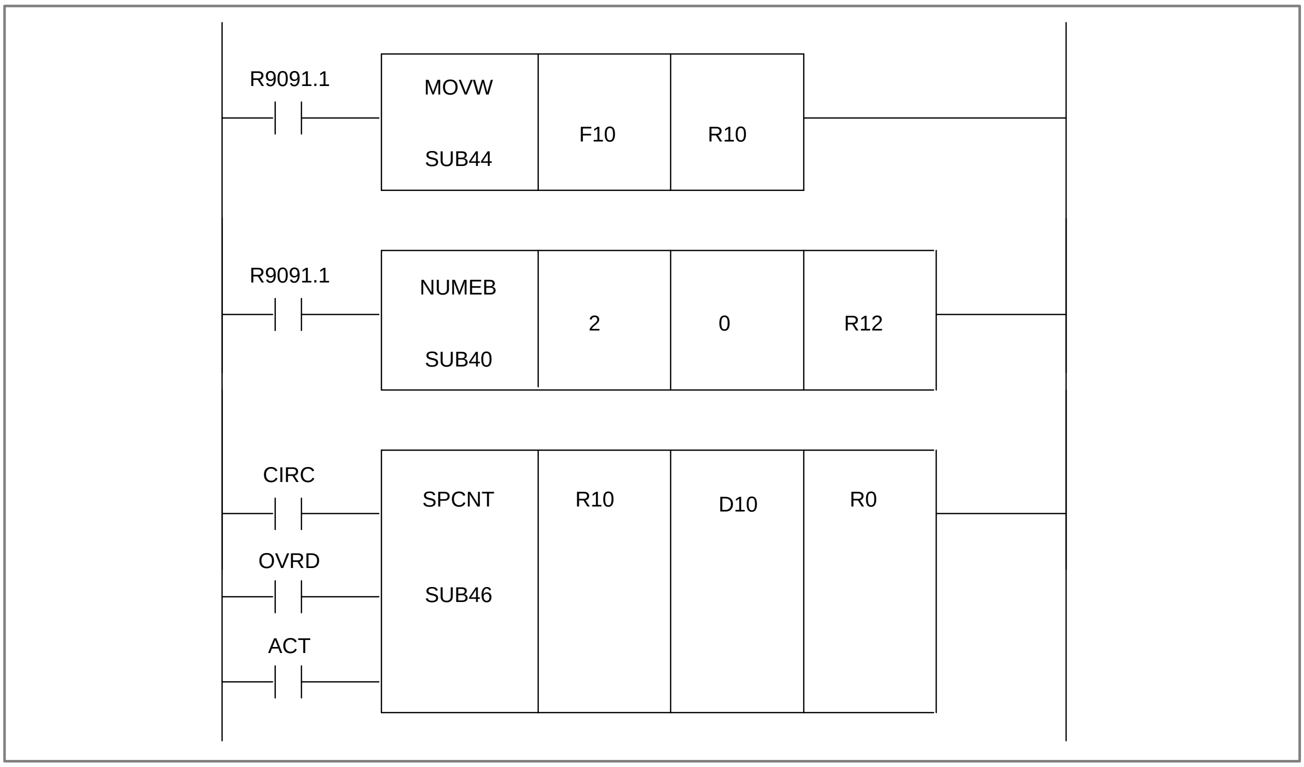
(c) Control the spindle in a tapping cycle
In a tapping cycle, spindle rotation is reversed at the bottom of a hole. Using the HIGH gear to reverse the rotation requires a lower analog voltage than using the LOW gear. So, using the HIGH gear reduces the machining time.
To widen the usable range of the 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 min^{-1}
Maximum speed specified for the spindle motor = 35000 min^{-1}
Maximum speed obtainable by the spindle motor = 40000 min^{-1}
(Maximum speed when 10 V is applied to the spindle motor)
Maximum speed for gear 1 = 25000 min^{-1}
Maximum speed for gear 2 = 40000 min^{-1}
Maximum speed for gear 3 = 6000 min^{-1}
Maximum speed for gear 4 = 100000 min^{-1}
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)):

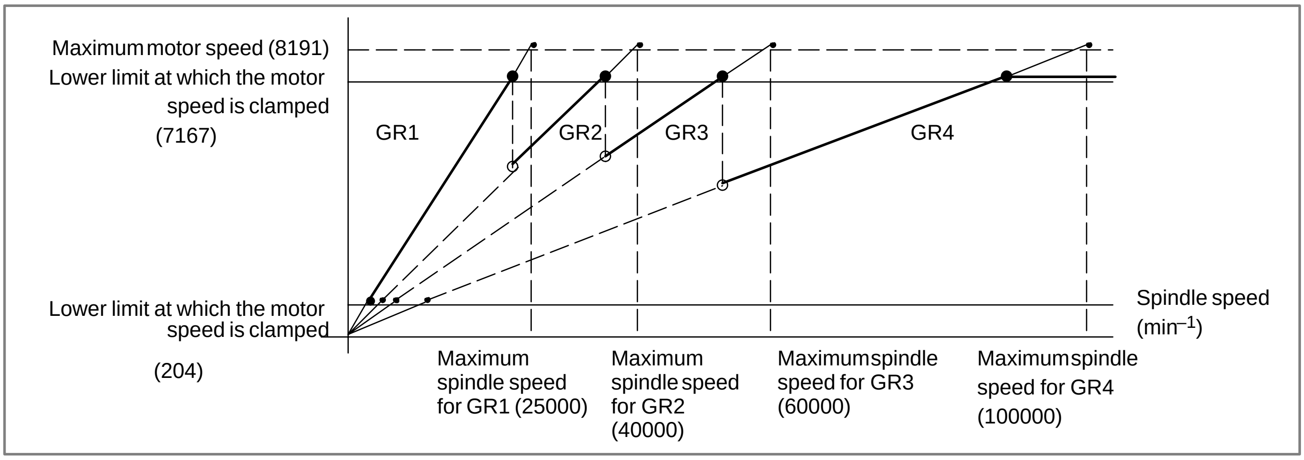
$$\text{Lower spindle motor speed limit data} = \frac{1000}{40000} \times 8191 = 204\ (\text{min}^{-1})$$

$$\text{Upper spindle motor speed limit data} = \frac{35000}{40000} \times 8191 = 7167\ (\text{min}^{-1})$$

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 5.63.5 Maximum and Minimum spindle speeds for each gear

	Minimum spindle speed (min^{-1})	Maximum spindle speed (min^{-1})
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 (min^{-1}), 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

5.64 END (END OF A LADDER PROGRAM)

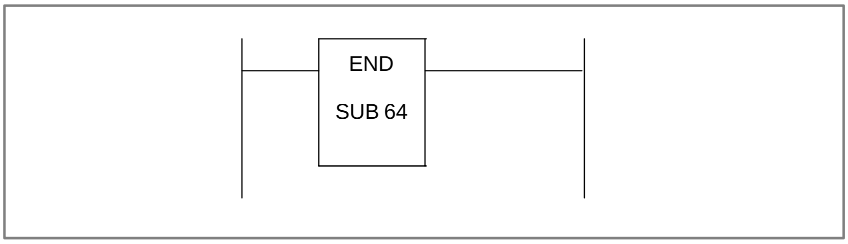
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.64.1 Function

The END functional instruction designates the end of a ladder program. END must be placed at the end of the ladder program.

5.64.2 Format



5.65 CALL (CONDITIONAL SUBPROGRAM CALL)

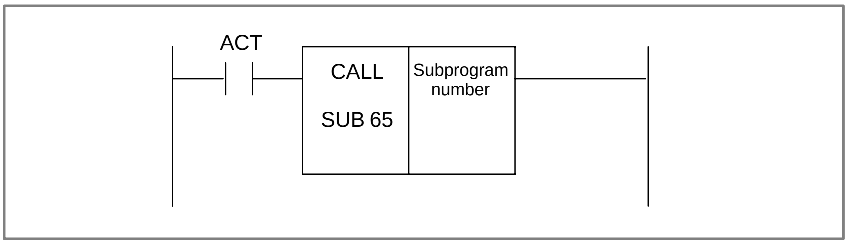
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.65.1 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.

5.65.2 Format



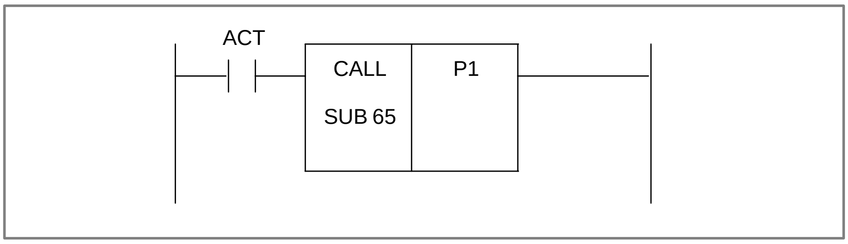
5.65.3 Control Conditions

(a) Input signal
ACT=0 : The CALL instruction is not executed.
ACT=1 : The CALL instruction is executed.

5.65.4 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



CAUTION
Be careful when using the CALL instruction with the COM, COME, JMP, or JMPE functional instruction. For details, see Chapter 9 in Part I.

5.66 CALLU (UNCONDITIONAL SUBPROGRAM CALL)

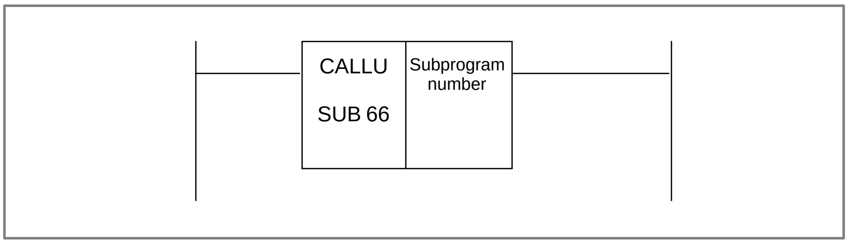
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.66.1 Function

The CALLU functional instruction calls a subprogram. When a subprogram number is specified, a jump occurs to the subprogram.

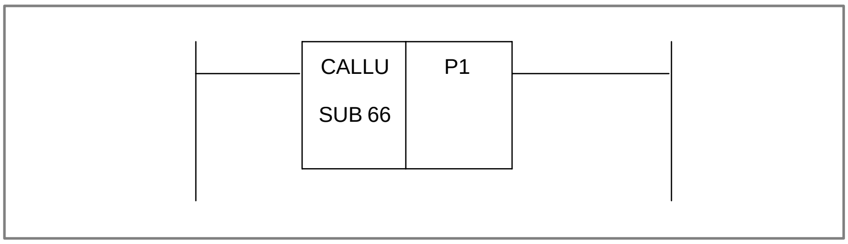
5.66.2 Format



5.66.3 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



5.67 SP (SUBPROGRAM)

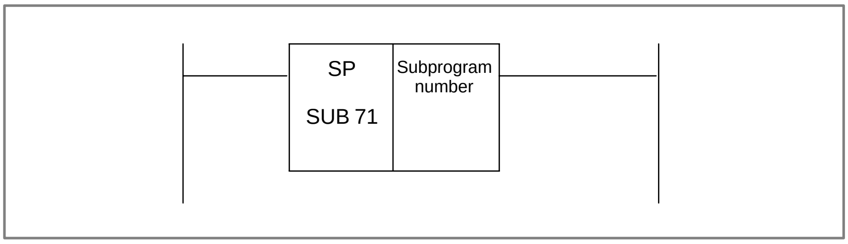
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.67.1 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.

5.67.2 Format

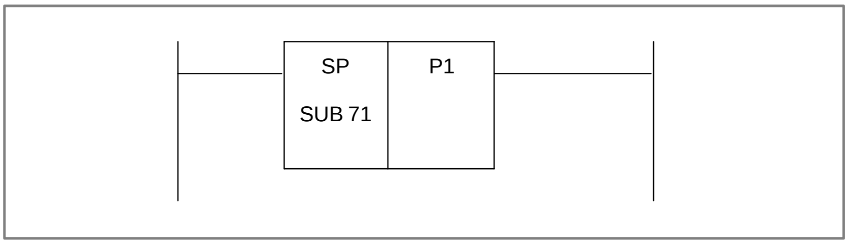


5.67.3 Parameters

(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	SB7	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 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



5.68 SPE (END OF A SUBPROGRAM)

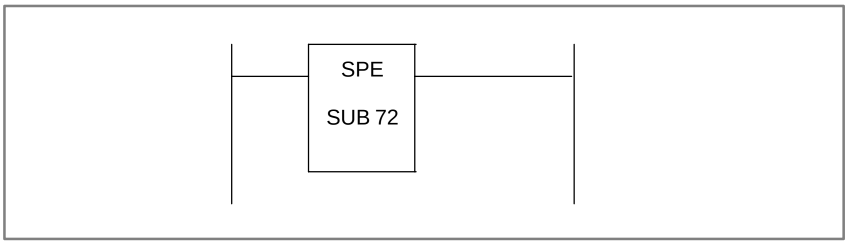
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.68.1 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.

5.68.2 Format



5.69 JMPB (LABEL JUMP)

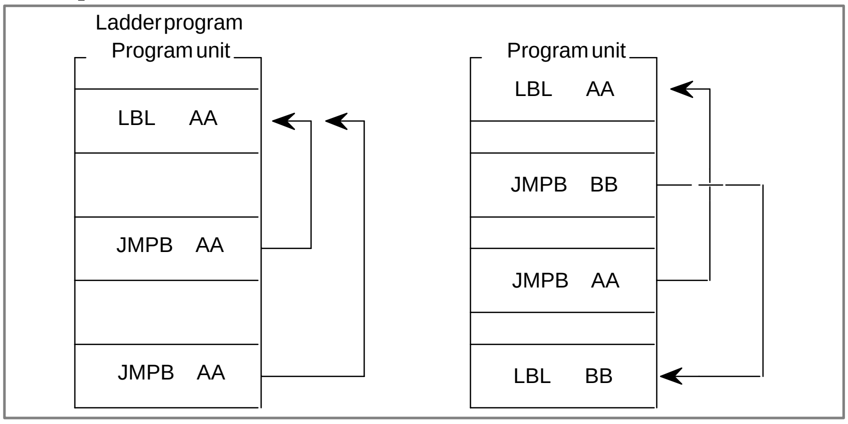
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

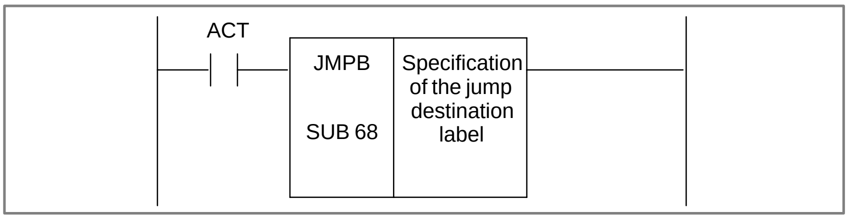
5.69.1 Function

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.



5.69.2 Format



5.69.3 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.

5.69.4 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.

CAUTION
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.

5.70 JMPC (LABEL JUMP)

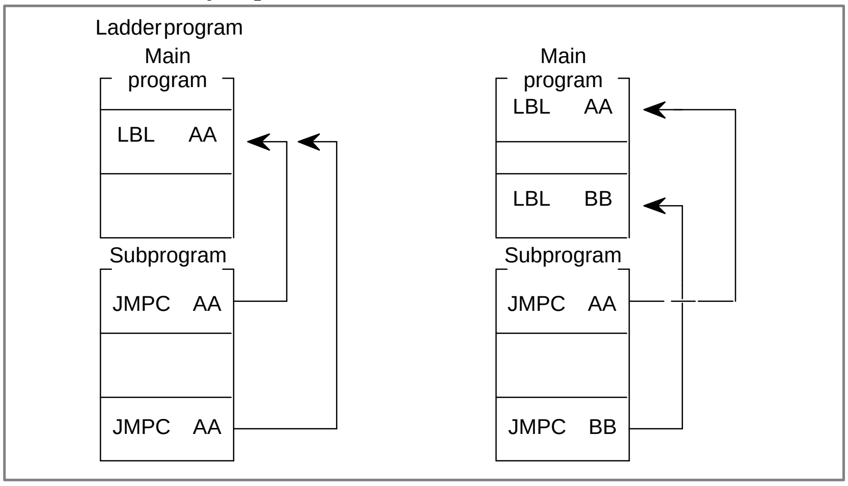
○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

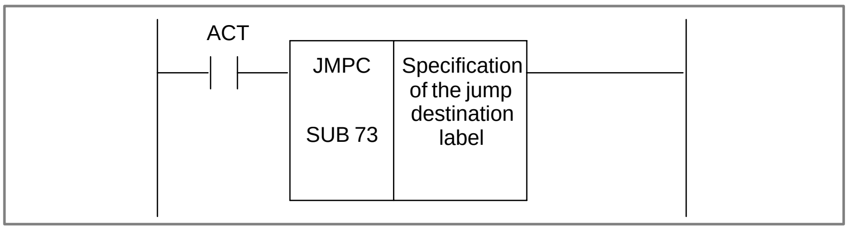
5.70.1 Function

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.



5.70.2 Format



5.70.3 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.

5.70.4 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.

CAUTION
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.

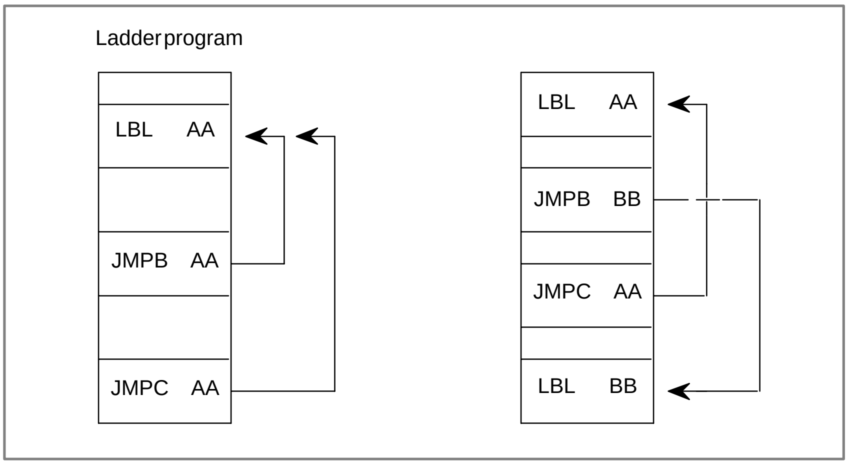
5.71 LBL (LABEL)

○ : Can be used
× : Cannot be used

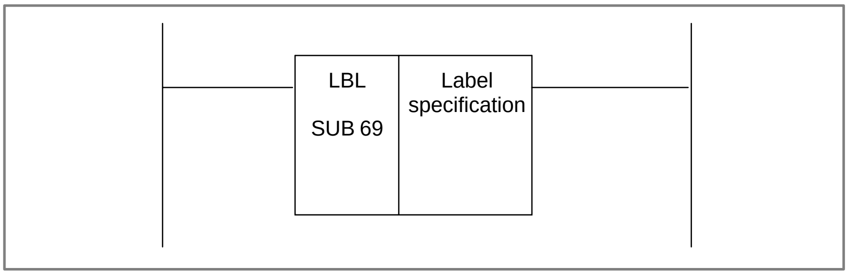
PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

5.71.1 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.)



5.71.2 Format



5.71.3 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.

5.72 AXCTL (AXIS CONTROL BY PMC)

○ : Available
× : Unavailable

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	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.

5.72.1 Function

This function simplifies the handshake of DI/DO signal for the axis control by PMC.

5.72.2 Format

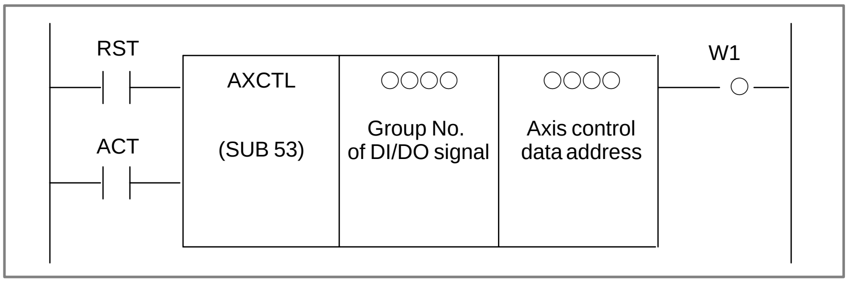


Fig. 5.72.2 AXCTL instruction format

Table 5.72.2 AXCTL instruction coding

Step Number	Instruc-tion	Address Number	Bit Number	Remarks
1	RD	○○○○.	○	RST
2	RD. STK	○○○○.	○	ACT
3	SUB	53		AXCTL command
4	(PRM)	○○○○		Number of DI/DO signal
5	(PRM)	○○○○		Axis control data address
6	WRT	○○○○.	○	W1, processing completion

5.72.3 Control Condition

ACT=0 : The AXCTL function is not executed.
If RST is 1, PMC axis control instruction reset processing is performed. If a modification is made to a ladder program with RST set to 1, the AXCTL instruction cannot sometimes be executed successively when the ladder program is executed again.
When modifying a ladder program, set RST and ACT to 0.

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.

5.72.4 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 2nd path side 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 3rd path side 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) Group number of DI/DO signal for PM*i*–D/H

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)
4 : Group D(G178 to G185, F139 to F141)
5 : Group E (G464 to G471, F500 to F502)
6 : Group F (G476 to G483, F503 to F505)
7 : Group G(G488 to G495, F506 to F508)
8 : Group H(G500 to G507, F509 to F511)

For 2nd path side of PM*i*–D/H

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)
1004 : Group D (G1178 to G1185, F1139 to F1141)
1005 : Group E (G1464 to G1471, F1500 to F1502)
1006 : Group F (G1476 to G1483, F1503 to F1505)
1007 : Group G (G1488 to G1495, F1506 to F1508)
1008 : Group H (G1500 to G1507, F1509 to F1511)

> **NOTE**
> With the PMC–SB5 (RB5), group E and subsequent groups cannot be used. The SB6 (RB6) is required to use these groups.

(c) 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	Total travel amount
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)	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)

CAUTION

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.
However this function applicables to the Power Mate *i*.
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.
2–3	0 or Feedrate	In case of CNC PRM8002#0= 0 not used. = 1 Feedrate.
4–7	Position in machine coordinate system	(Absolute)

> **CAUTION**
> It is necessary to set the CNC parameters relating to the axis movement.

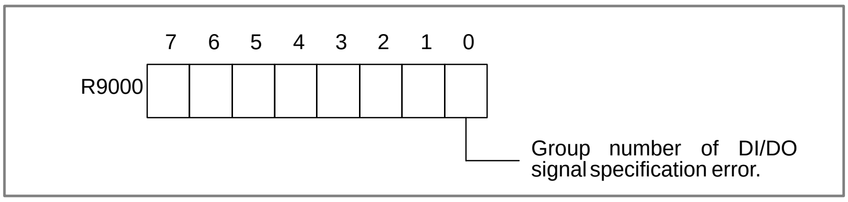
5.72.5 End of Command (W1)

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).

5.72.6 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).

5.72.7 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.

5.73 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	SB7	SC	SC3 SC4	NB	NB2	NB6
○	○	×	×	×	×	×	×	×	Δ	Δ	×	×	×	×	×	×

> **NOTE**
> The PMC–SB5/SB6 can be used only in the Power Mate *i*–D/H.

5.73.1 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.

5.73.2 Format

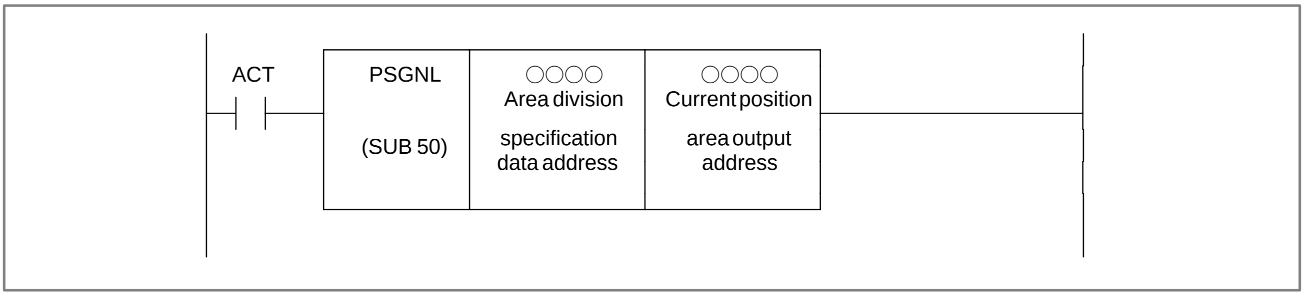


Fig. 5.73.2 PSGNL instruction format

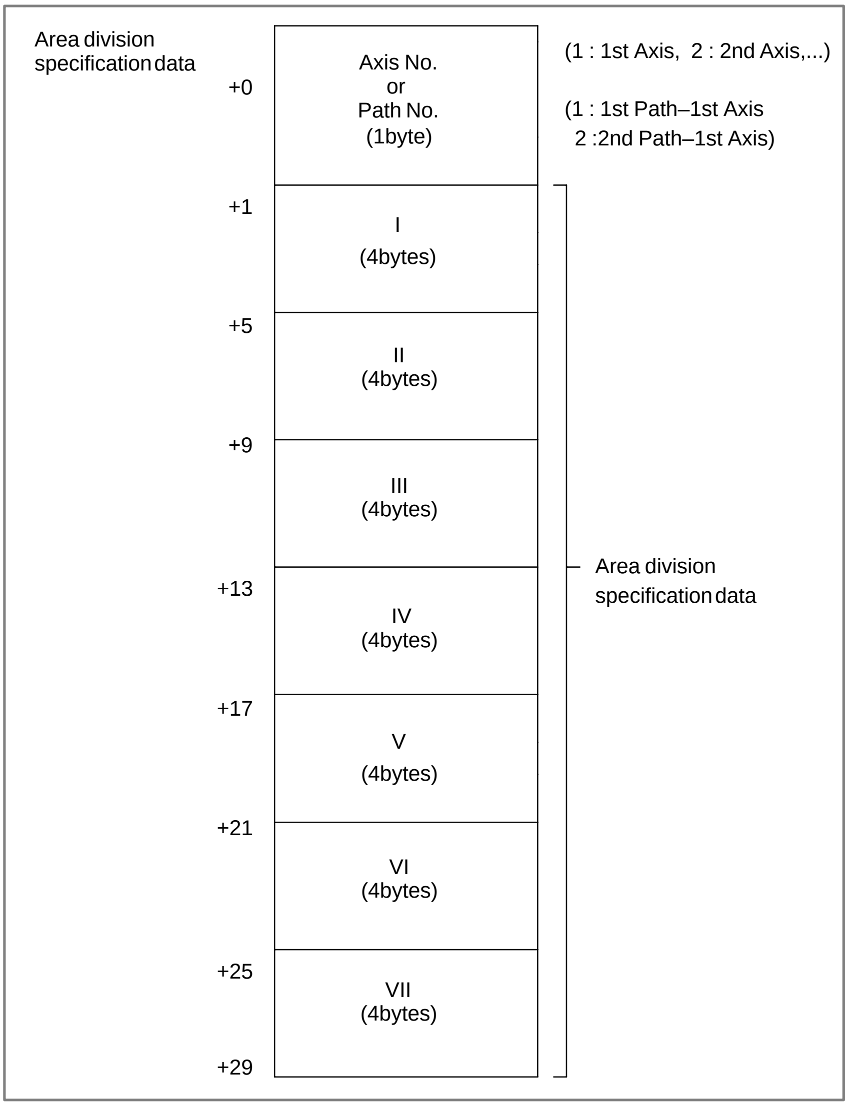
5.73.3 Control Condition

(a) Execution specification (ACT)
ACT=0 : The PSGNL instruction is not executed.
ACT=1 : The PSGNLnstruction is executed.

5.73.4 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

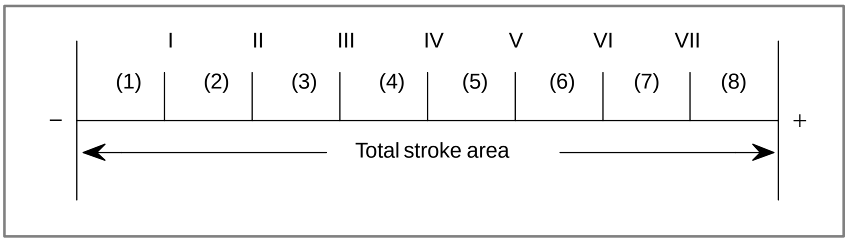
- In case of path specification (Power Mate–MODEL D dual path control)
  Please set path–No. of axis to select. (1 byte data of binary format)

(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)

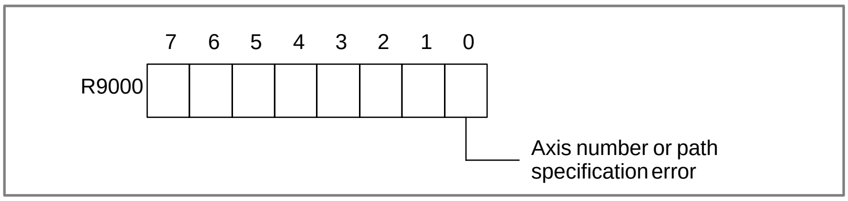
Corresponding bit is set to 1 indicates the area in which the current position in the machine coordinates system is located.

(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.

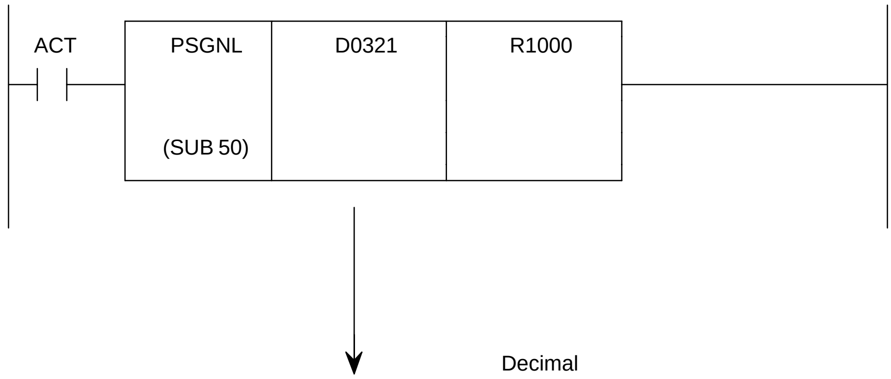
5.73.5 Operation Output Register (R9000)

If an error occurs in position signal output processing, the corresponding bit of the operation output register is set.



5.73.6 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.

5.74 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	SB7	SC	SC3 SC4	NB	NB2	NB6
○	○	×	×	×	×	×	×	×	Δ	Δ	×	×	×	×	×	×

NOTE
The PMC–SB5/SB6 can be used only in the Power Mate *i*–D/H.

5.74.1 Function

Turn W1=1 which th ecurrent position in the machine coordinates system is in the area specified by parameters.

5.74.2 Format

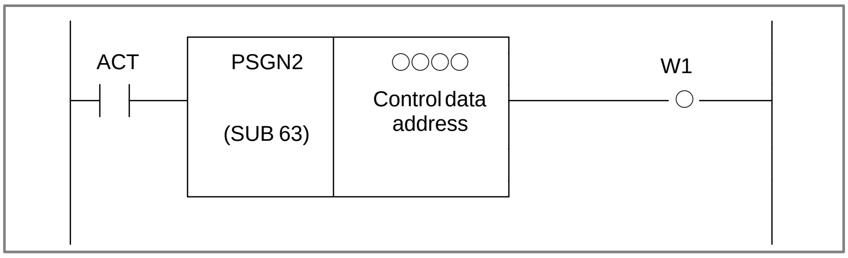


Fig. 5.74.2 PSGN2 instruction format

5.74.3 Control Condition

(a) Execution specification (ACT)
ACT=0 : The PSGN2 instruction is not executed.
ACT=1 : The PSGN2 instruction is executed.

5.74.4 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.

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 two 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.

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.

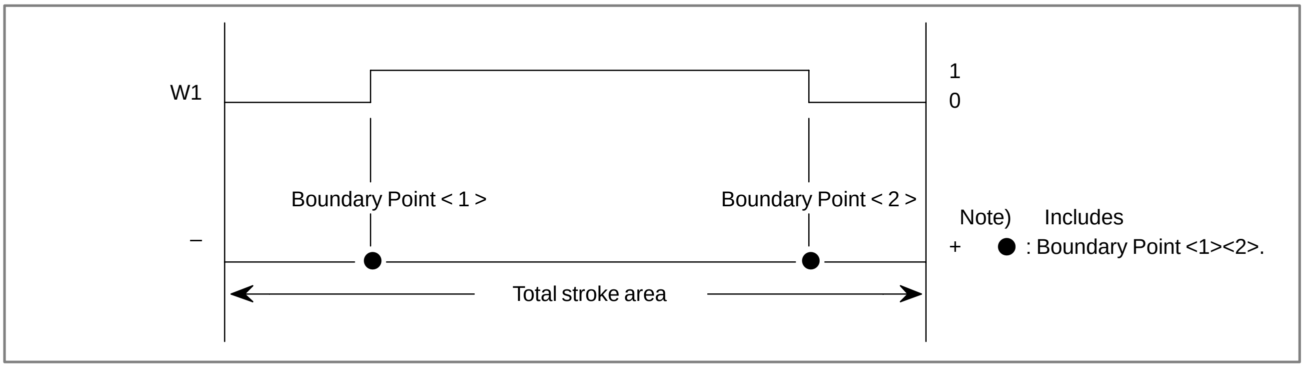
- Axis number specification
  An axis number is specified. (Binary one–byte data)

(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.

CAUTION
Data for both boundary points <1> and <2> must be specified in ascending order (boundary point 1 ≦ boundary point 2).

<Example of area division>

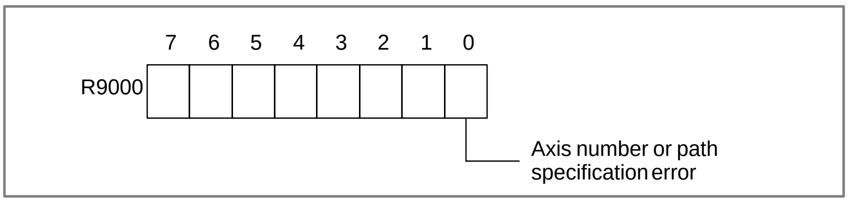


5.74.5 Current Position Area Output (W1)

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.

5.74.6 Operation Output Register (R9000)

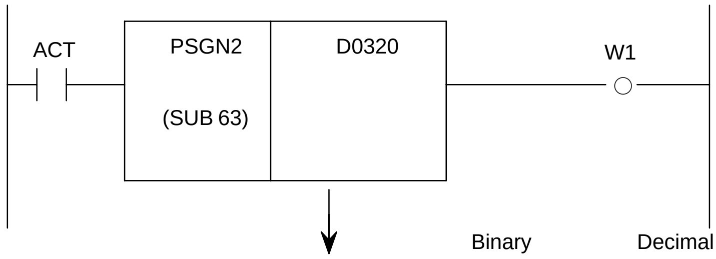
If an error occurs in position signal output processing, the corresponding bit of the operation output register is set.
In this case, W1 = 0.



5.74.7 Example of Using Position Signals

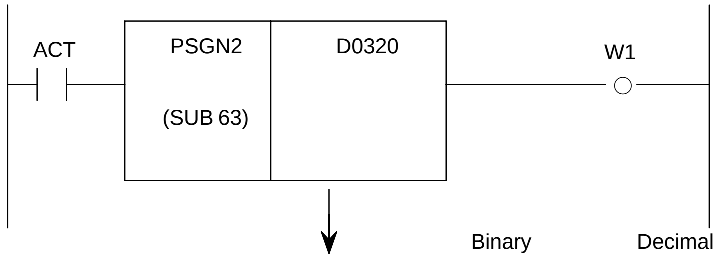
- This example illustrates how to output the position signal for the current position of the second axis of path 1 in the machine coordinate system if it is the position between –100.000 mm and 200.000 mm. (The control data address is set to D0320.)

(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

5.75 NOP (NO OPERATION)

○ : Can be used
Δ : Can be used depending of the version of software
× : Cannot be used

PM–D/H	PA1	PA3						
	×	Δ						
FS16/18A	SA1	SA2	SA3	SB	SB2	SB3	SC	SC3
	×	×	×	×	×	×	×	×
FS16/18B	SA1	SB3	SB4					
	×	Δ	Δ					
FS16/18 B/C	SC3	SC4						
	Δ	Δ						
FS16/18C	SA1	SB5	SB6					
	Δ	Δ	Δ					
FS20	SA1	SA3						
	×	×						
FS21B	SA1	SA3						
	×	×						
FS0*i*	SA1	SA3						
	×	×						
FS16*i*/18*i*/21*i* PM*i*	SA1	SA5	SB5	SB6	SB7			
	○	○	○	○	○			
FS15B	NB	NB2						
	Δ	Δ						
FS15*i*	NB6							
	○							

5.75.1 Function

During creation of a ladder program using the offline programmer, if the program is compiled with specifying the setting with which a net comment or form feed code is used and the point of the net comment is output, position information of the net comment or form feed code is output as the NOP instruction. This instruction performs no operation during execution of the ladder.

CAUTION

1 Do not move or delete the NOP instruction using the built–in edit function. During decompilation, position information of net comments and form feed are lost.

2 For the model which does not support this instruction, always compile a program without specifying the setting with which the net comment pointers are output. If the NOP instruction is executed on a model which does not support the instruction, a system error may occur.
Set the following items on the programmer compiler option screen as shown below:
FAPT LADDER–II Select [None] for [Net Comment].
FAPT LADDER–III Uncheck [Output Net Comment Pointers].

3 This instruction can be used on the following models on which the listed version or later of software is installed:

PM–D/H	PMC–PA1	4074/01 to, 4078/06 to (Cannot be used on series 4075)
FS16B/18B	PMC–SB3/SB4	4066/08 to
	Edit card	4073/06 to
FS16C/18C	PMC–SA1/SB5/SB6	4067/09 to
	Edit card	4090/04 to
FS16B/18B/16C/18C	PMC–SC3/SC4	4068/08 to
FS15B	PMC–NB/NB2	4048/06 to, 4049/01 to (Cannot be used on series 4075)

6 NONVOLATILE MEMORY

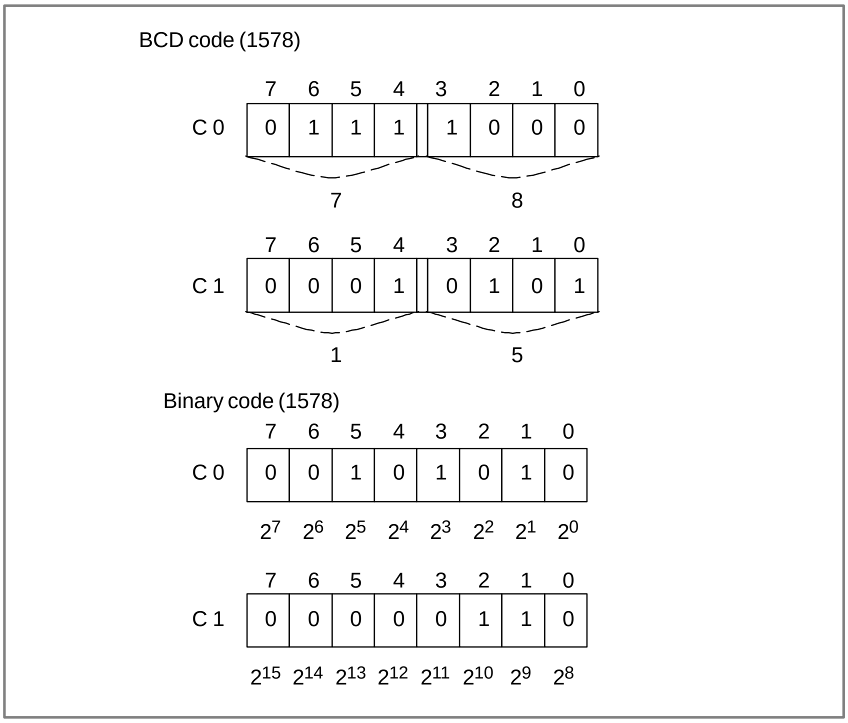
6.1 TIMER, COUNTER, KEEP RELAY, NONVOLATILE MEMORY CONTROL, DATA TABLE

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 I–3.

(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. Fig 6.3 (a) is a general configuration of the data table, and Fig 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.

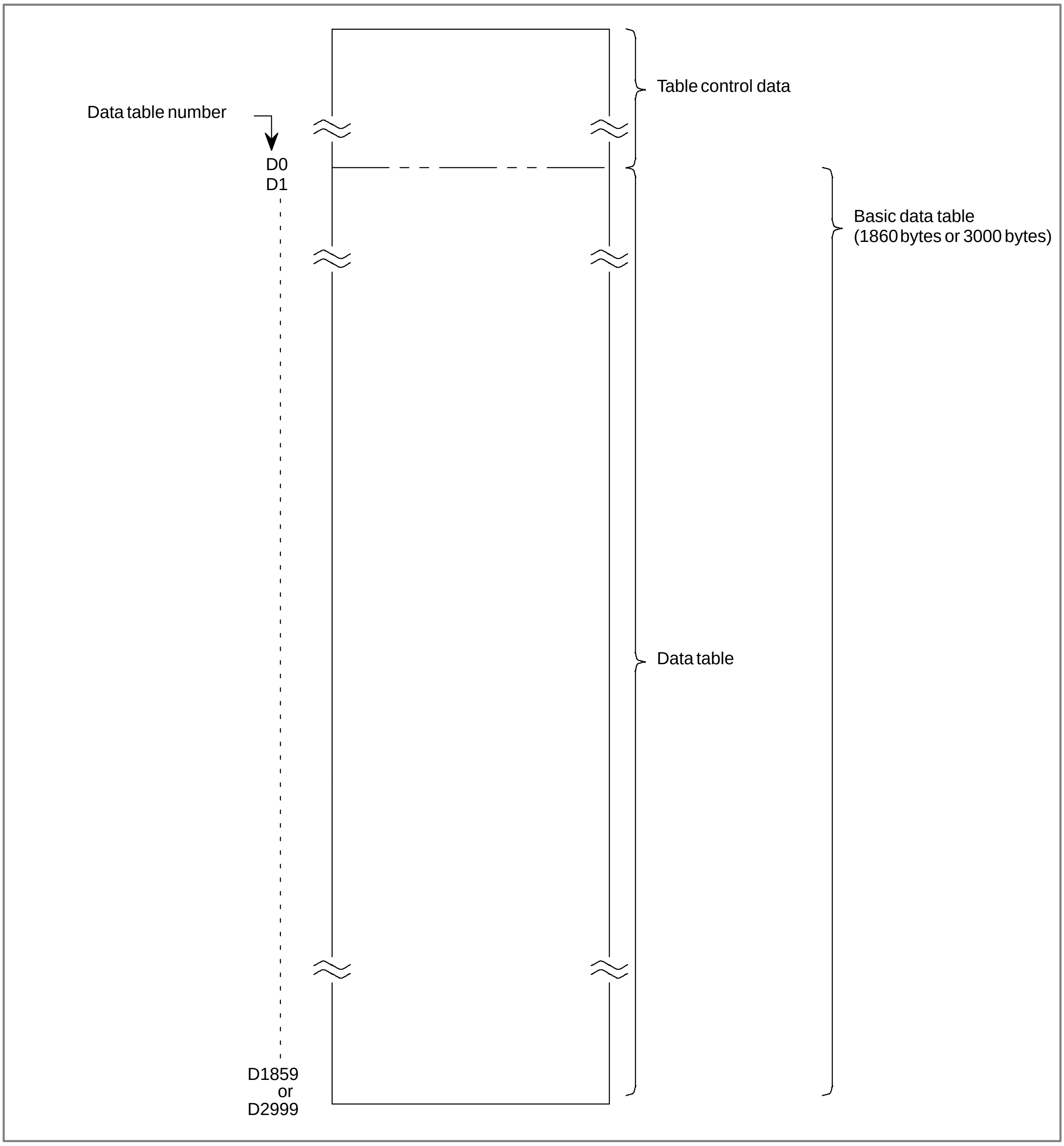


Fig. 6.3 (a) General configuration of data table

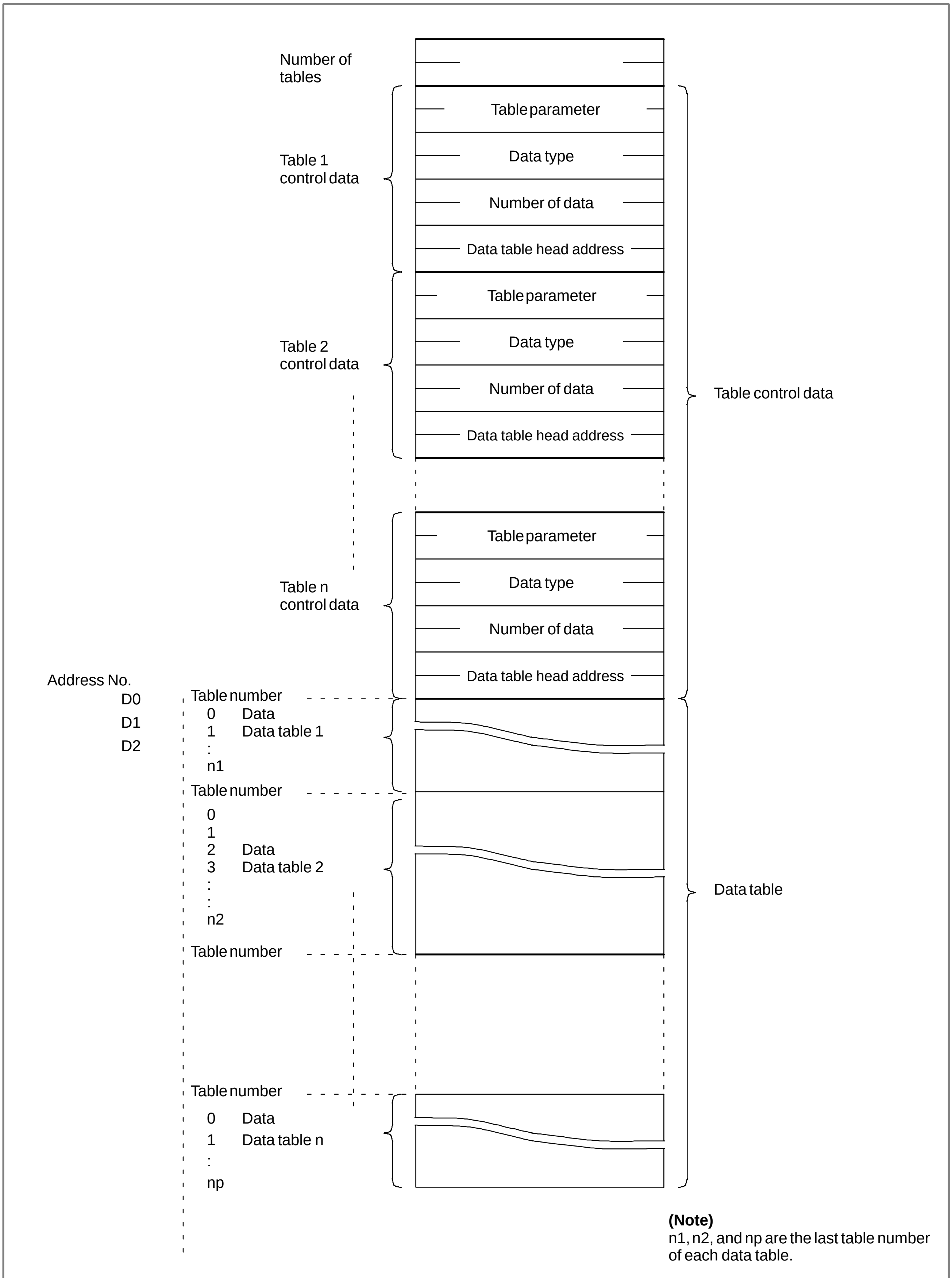


Fig. 6.3 (b) Detailed configuration of data table

(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.

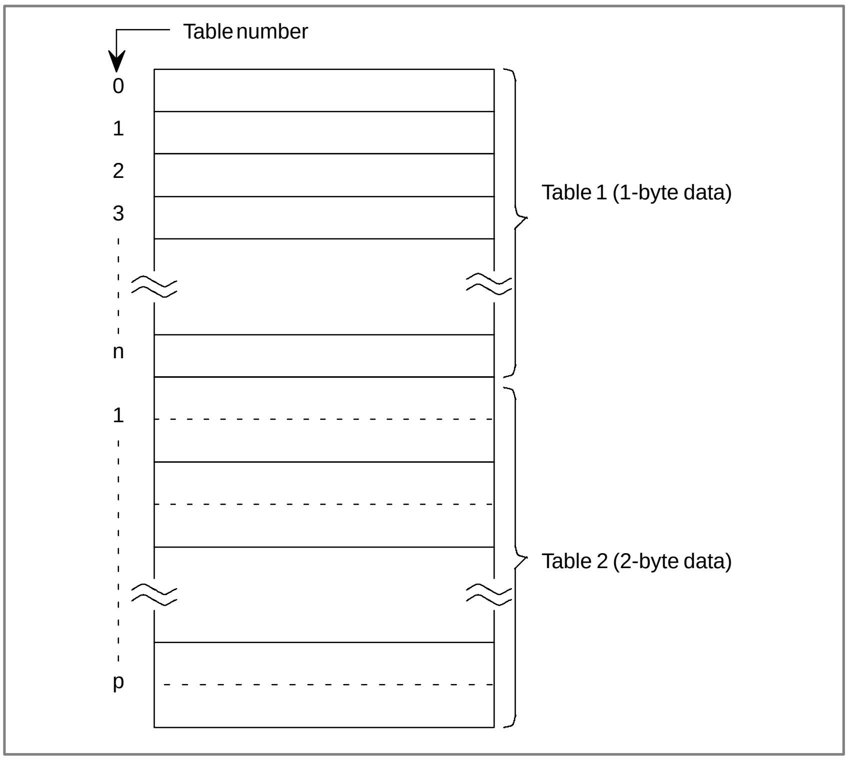
(iii) Data type
Specify the length of data in the data table.
- 0 : One byte
- 1 : Two bytes
- 2 : Four bytes

(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 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)

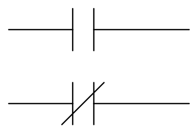
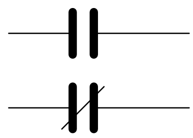
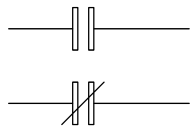
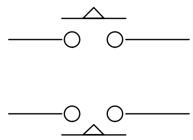
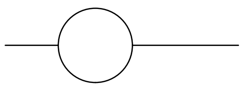
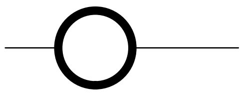
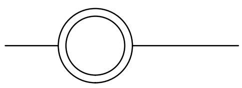
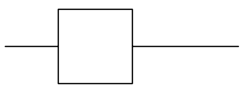
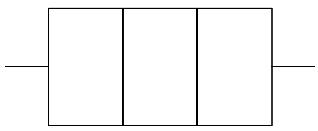
7.1.3 Comments

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.

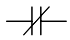
7.1.4 Line Numbers

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.

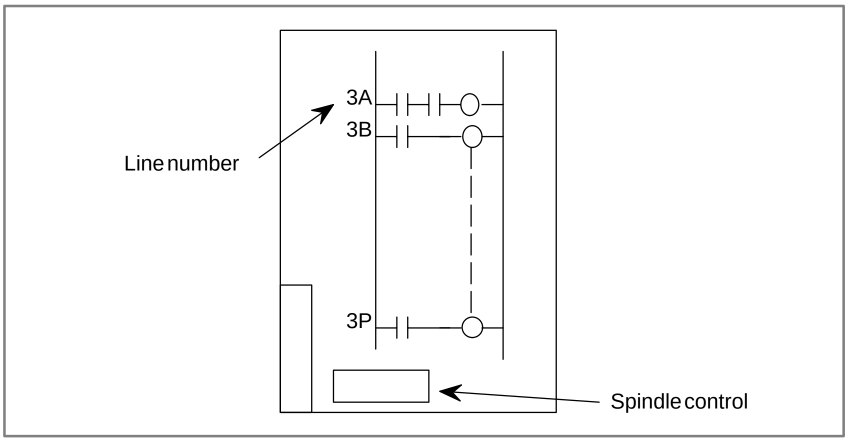
NOTE

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).

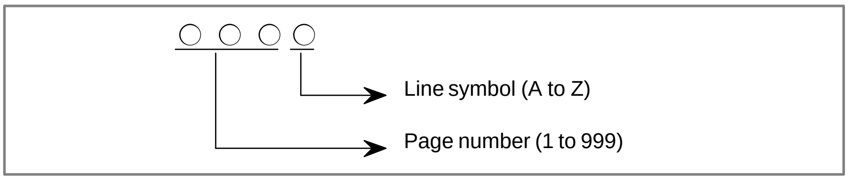
(2) Columns are used for wiring.



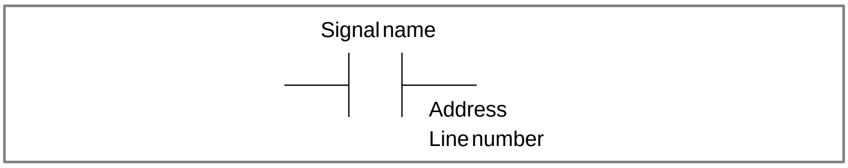
(3) Divide the circuits into several functions. And program the same function in a single program.

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

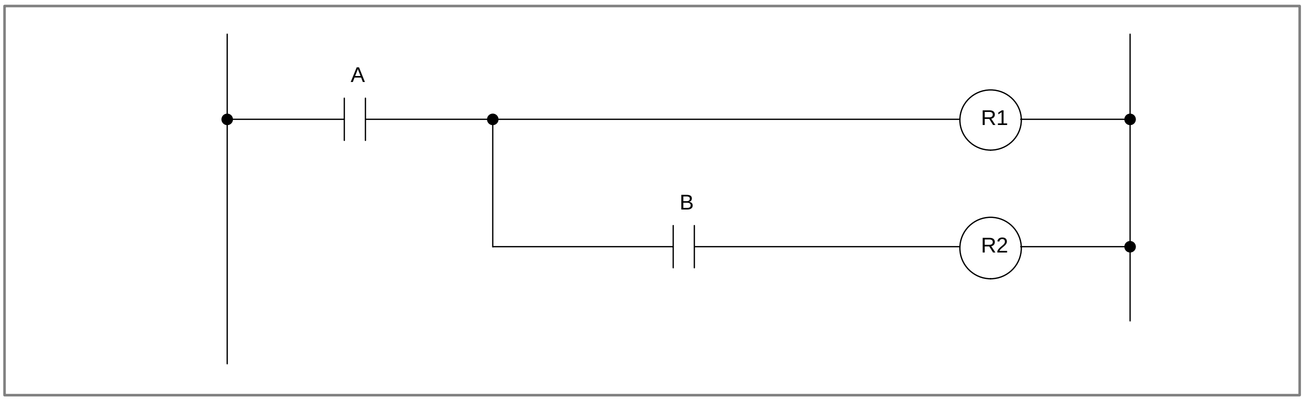
(iii) Setting table of timer, counter, and PMC parameters and meaning of them.

(iv) Description of functional instruction.

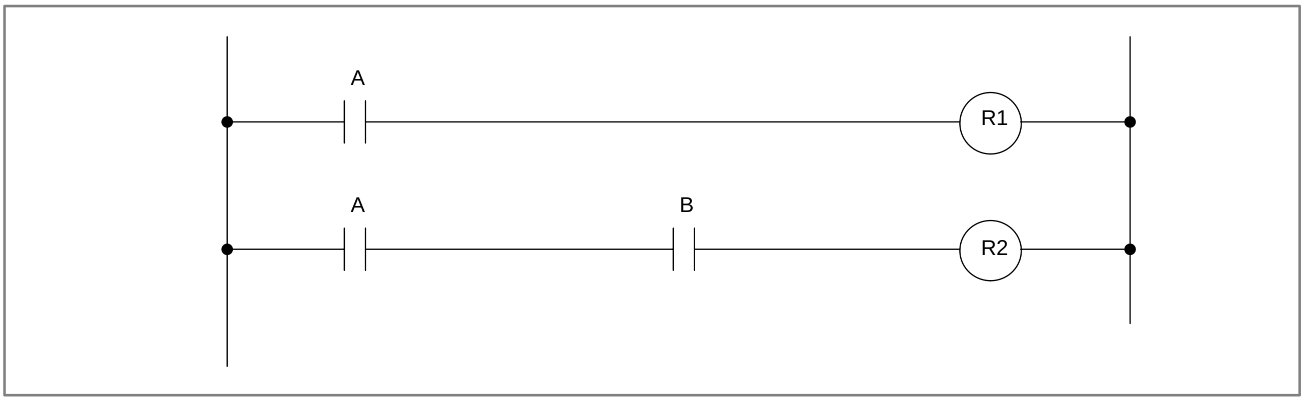
(10) Easy-to-understand name should be assigned.

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	SB7	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:

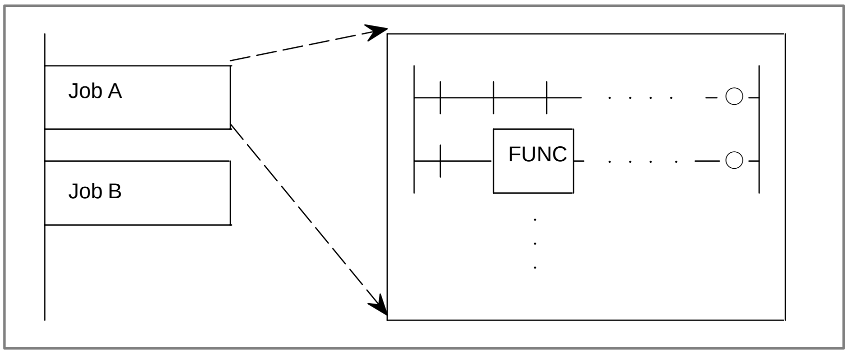
- A program can be understood and developed easily.
- A program error can be found easily.
- When an operation error occurs, the cause can be found easily.

9.1 EXAMPLES OF STRUCTURED PROGRAMMING

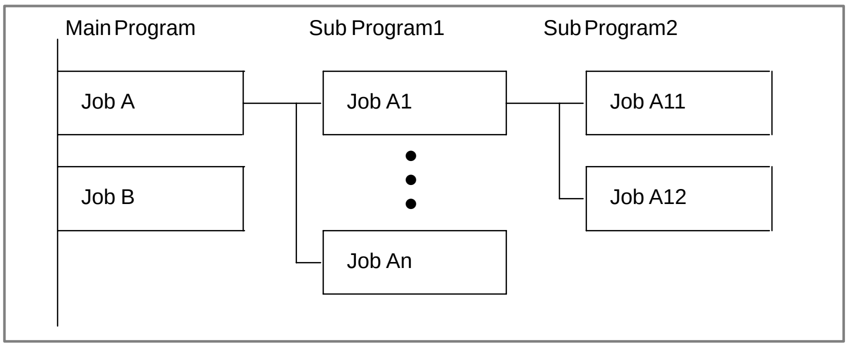
9.1.1 Implementation Techniques

Three major structured programming capabilities are supported.

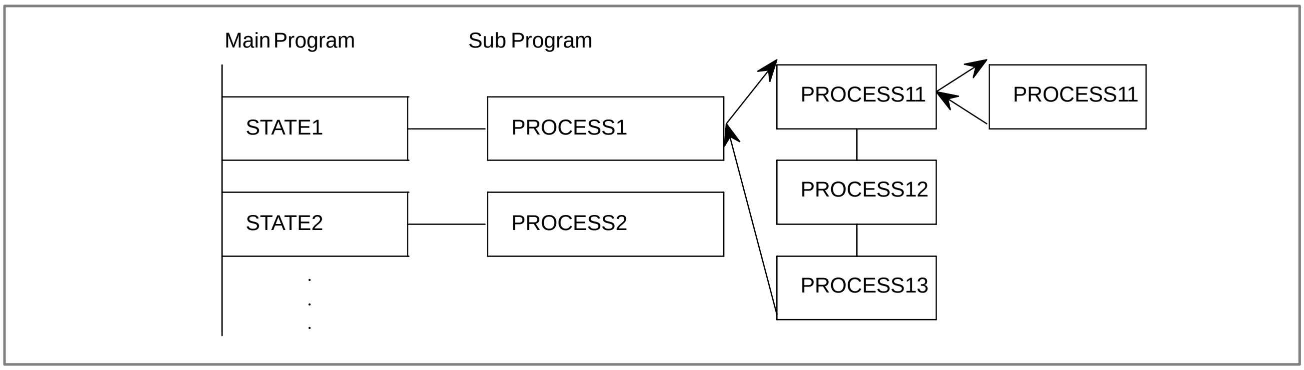
(1) Subprogramming
A subprogram can consist of a Ladder sequence as the processing units.



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



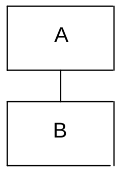
(3) Conditional branch
The main program loops and check whether conditions are satisfied. If a condition is satisfied, the corresponding subprogram is executed. If the condition is not satisfied, the subprogram is skipped.



9.1.2 Applications

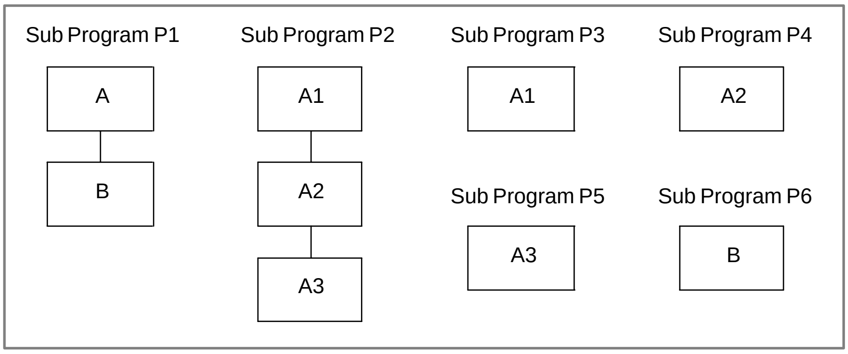
(1) Example

Suppose that there are four major jobs.

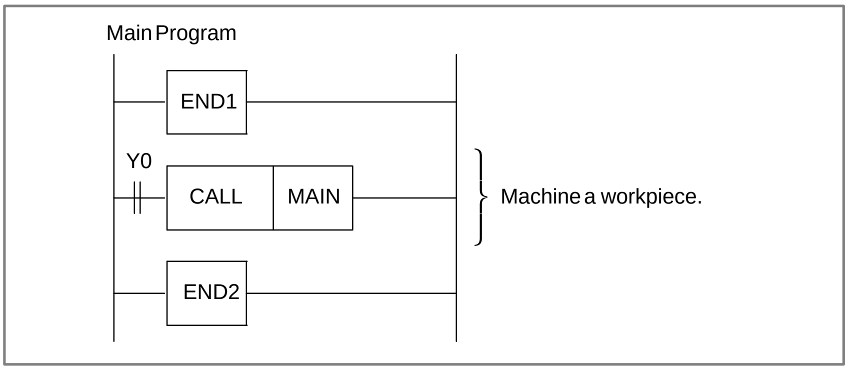


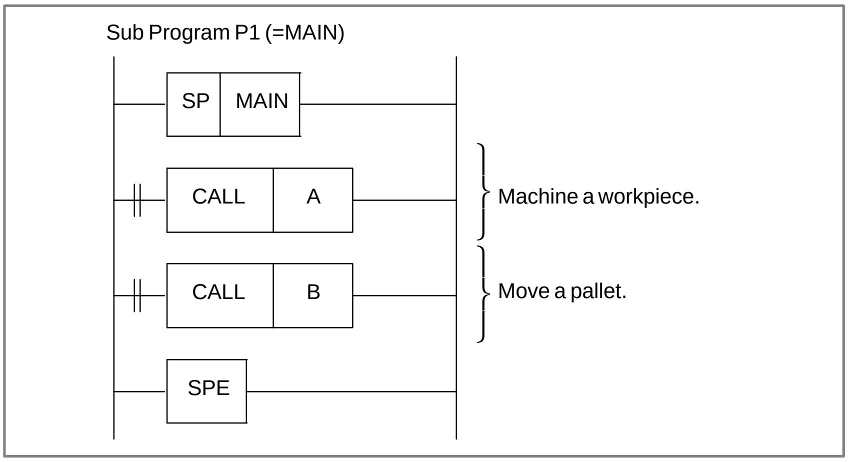
If Y0 is 1, a request to machine a workpiece is assumed, and processing is performed (with a condition).
A : ① Pick up a workpiece from a pallet. (A1)
②Machine the workpiece. (A2)
③Return the workpiece to the pallet. (A3)
B : ④Move the pallet.

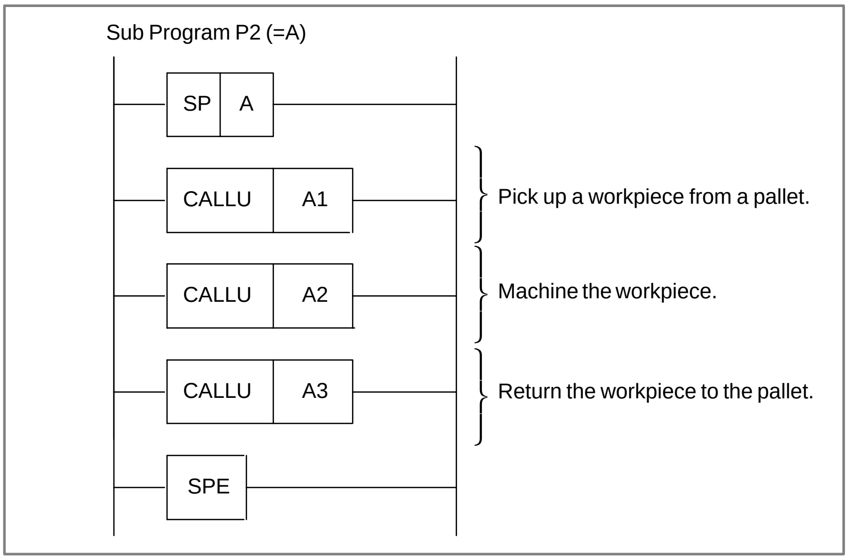
(2) Program structure

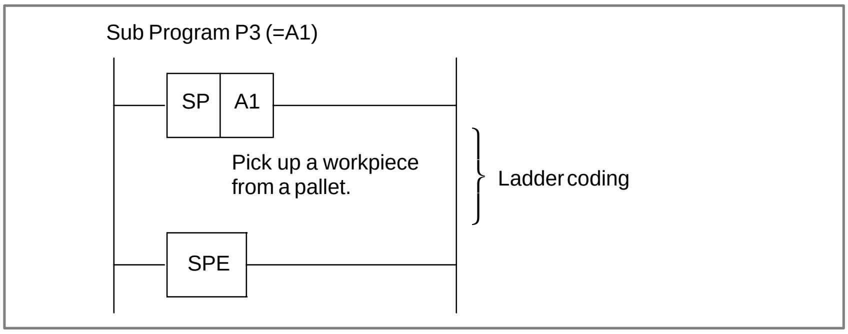


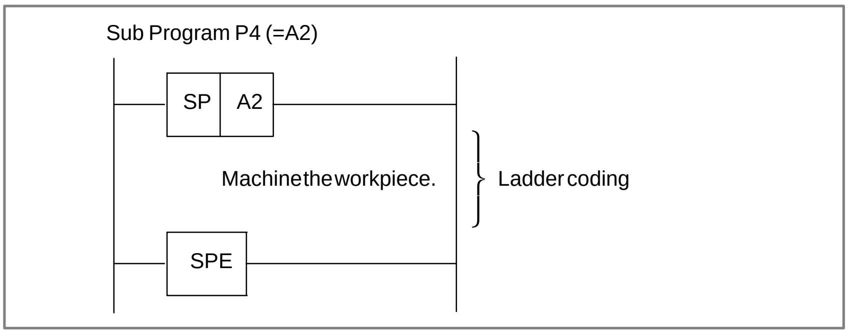
(3) Program description

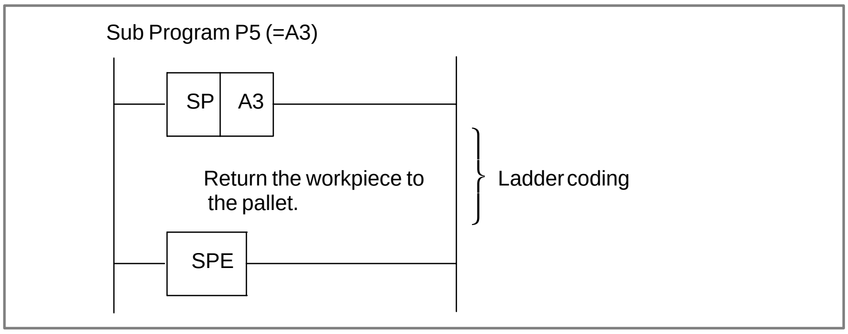


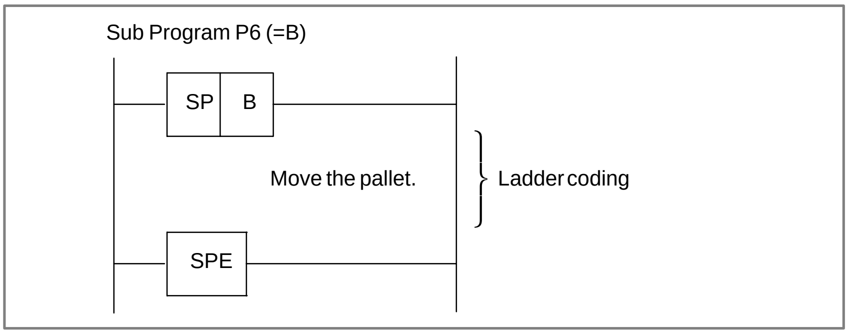



Sub Program P2 (=A)
SP
A
CALLU
A1
Pick up a workpiece from a pallet.
CALLU
A2
Machine the workpiece.
CALLU
A3
Return the workpiece to the pallet.
SPE


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.

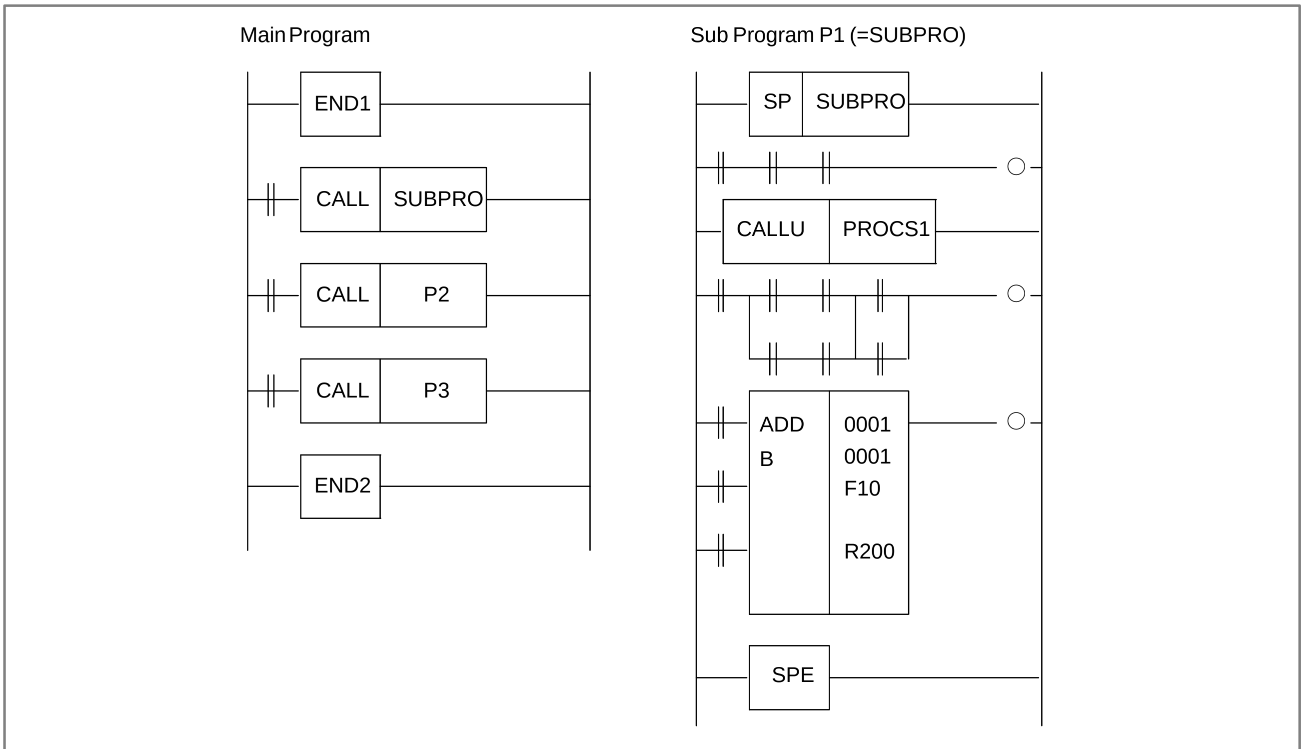
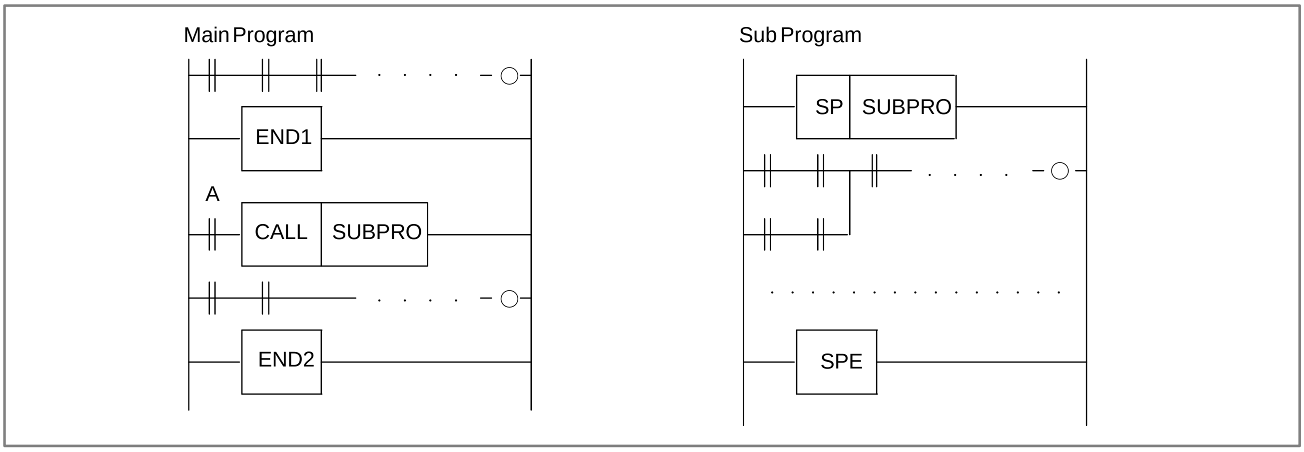
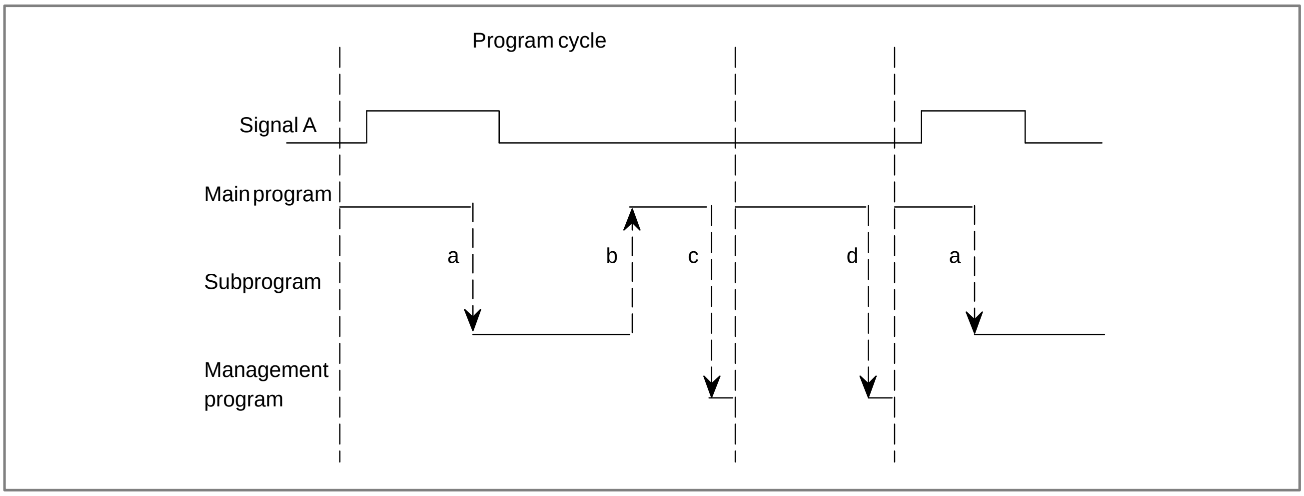


Fig. 9.2.1 Example of subprogramming and nesting

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.

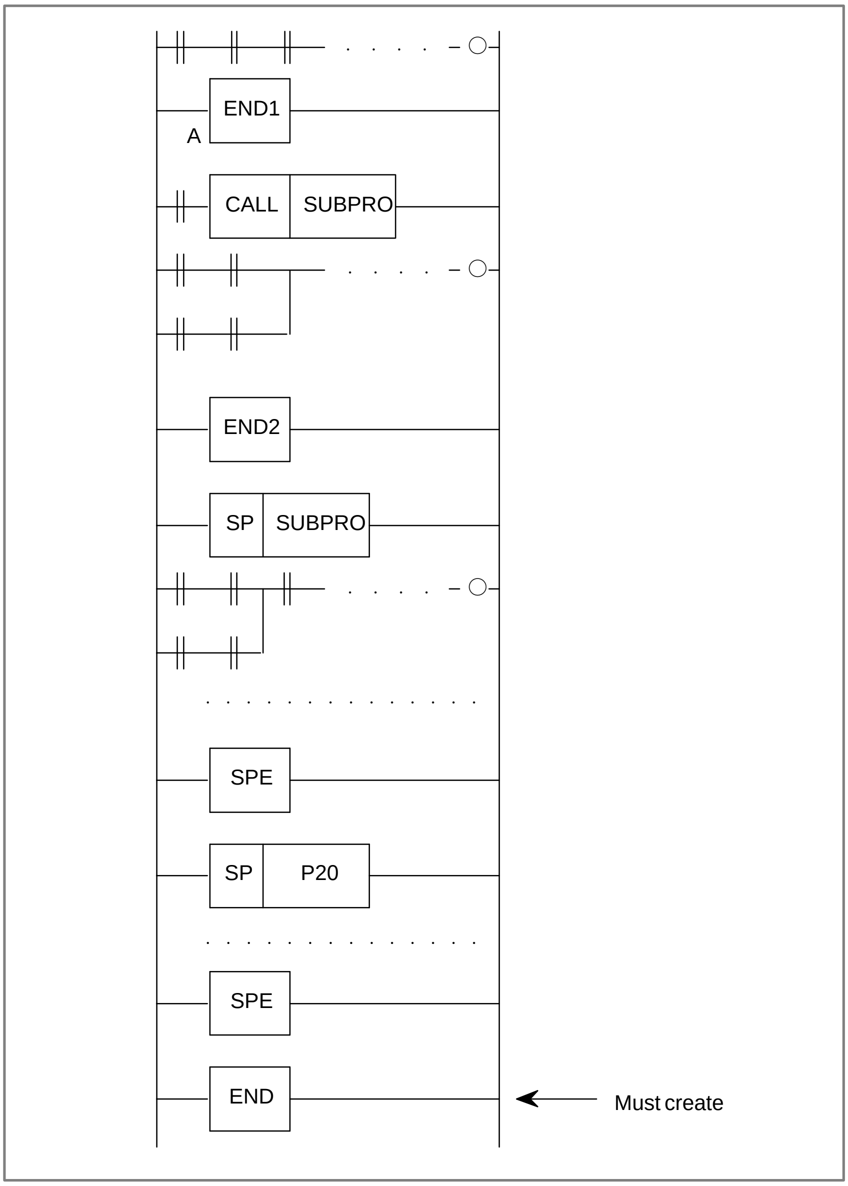
b : When the end of the subprogram is reached, control is returned to the main program.

c : When the end of the main program is reached, the management program performs Ladder program postprocessing.

9.2.3 Creating a Program

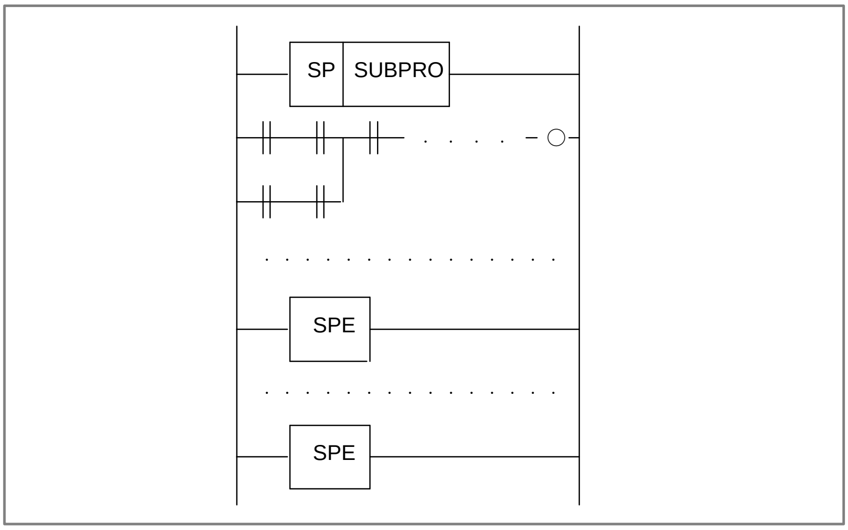
Create subprograms in the same way as the first-, second-, and third-level Ladder programs.

Example of creation

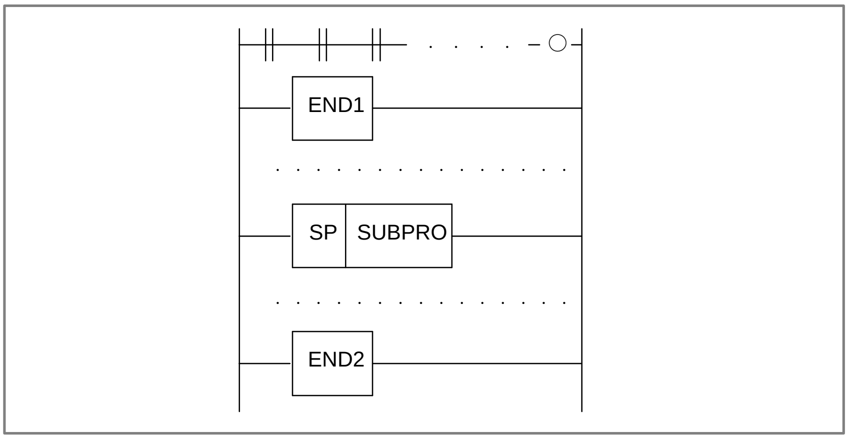


Inhibit items

(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

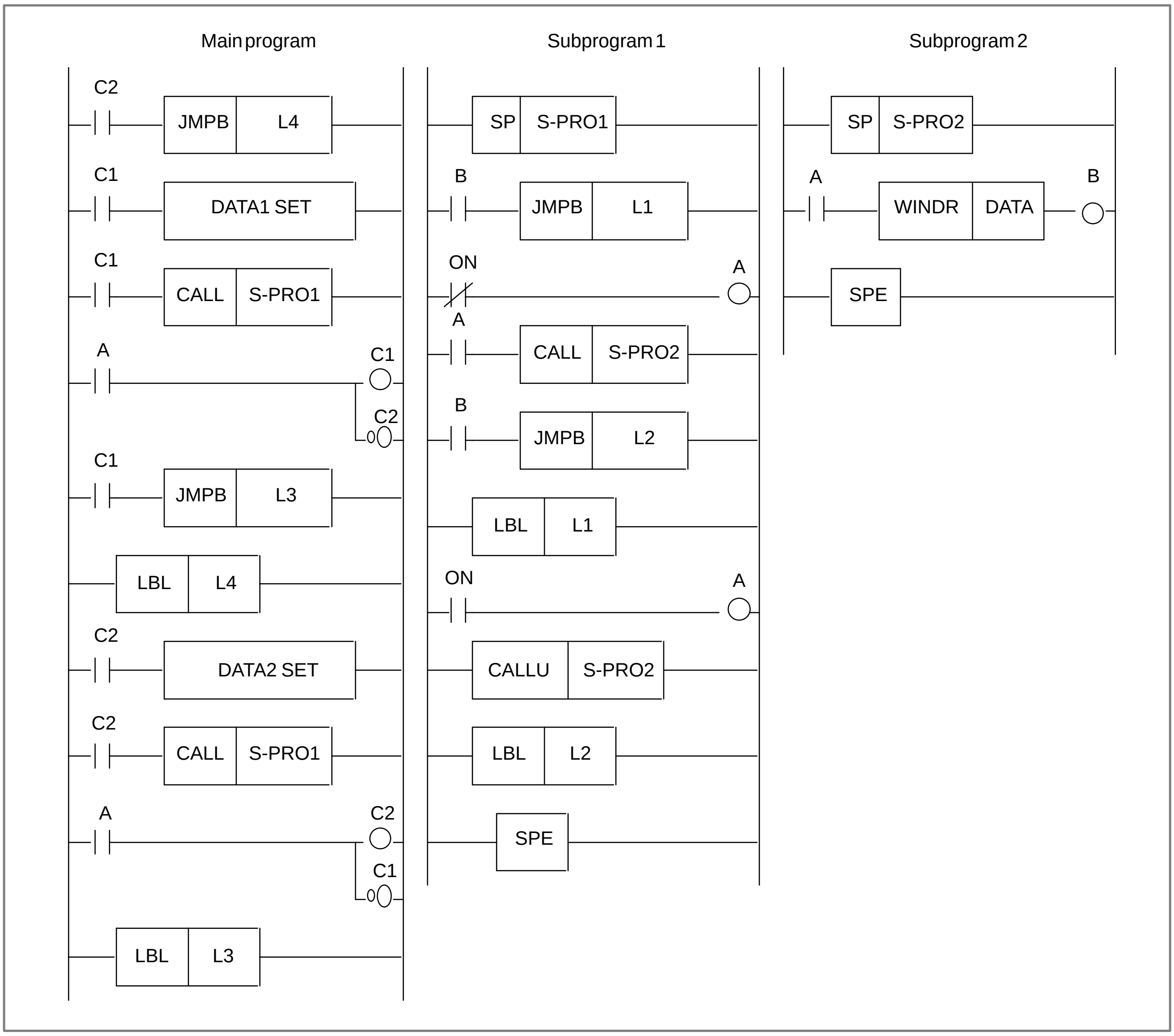
When you use the above-mentioned functional instructions, ACT=1 must be held until the transfer completion information(W1) becomes 1. Therefore, be careful of the following when using those instructions in subprograms.

- 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.

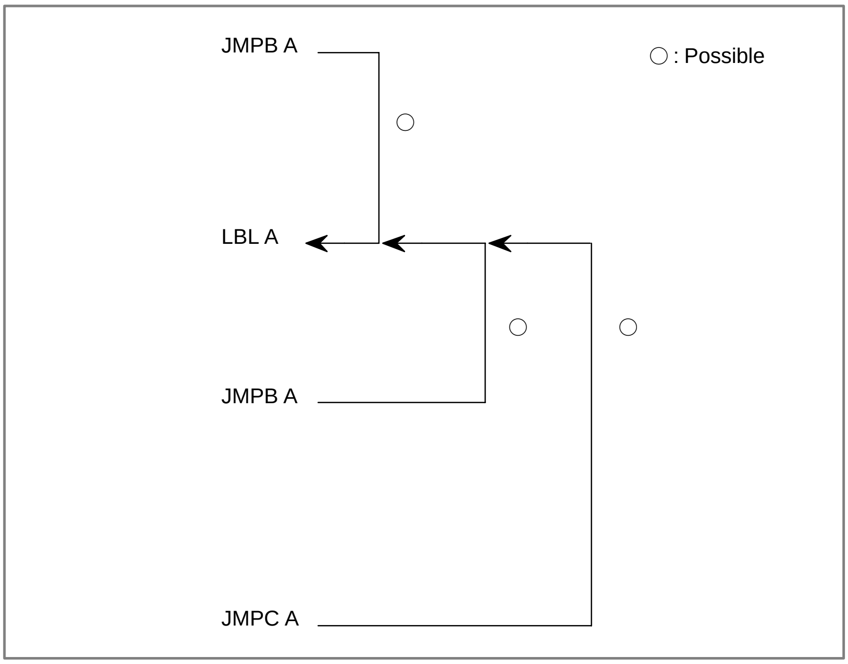
10 JMP INSTRUCTIONS WITH LABEL SPECIFICATION

○ : Can be used
× : Cannot be used

PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SB7	SC	SC3	SC4	NB	NB2	NB6
×	○	×	×	○	○	×	×	○	○	○	○	○	×	○	○	○	○	○

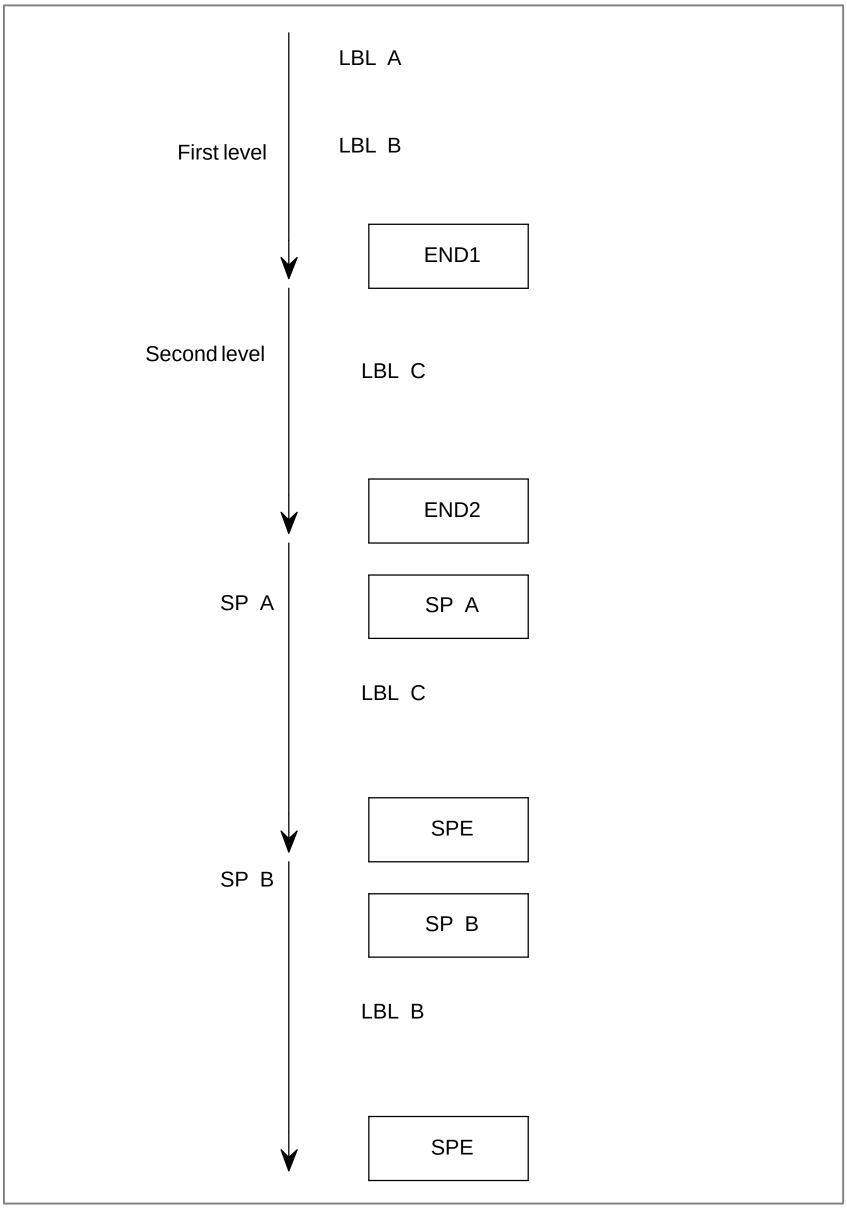
10.1 SPECIFICATIONS

(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.

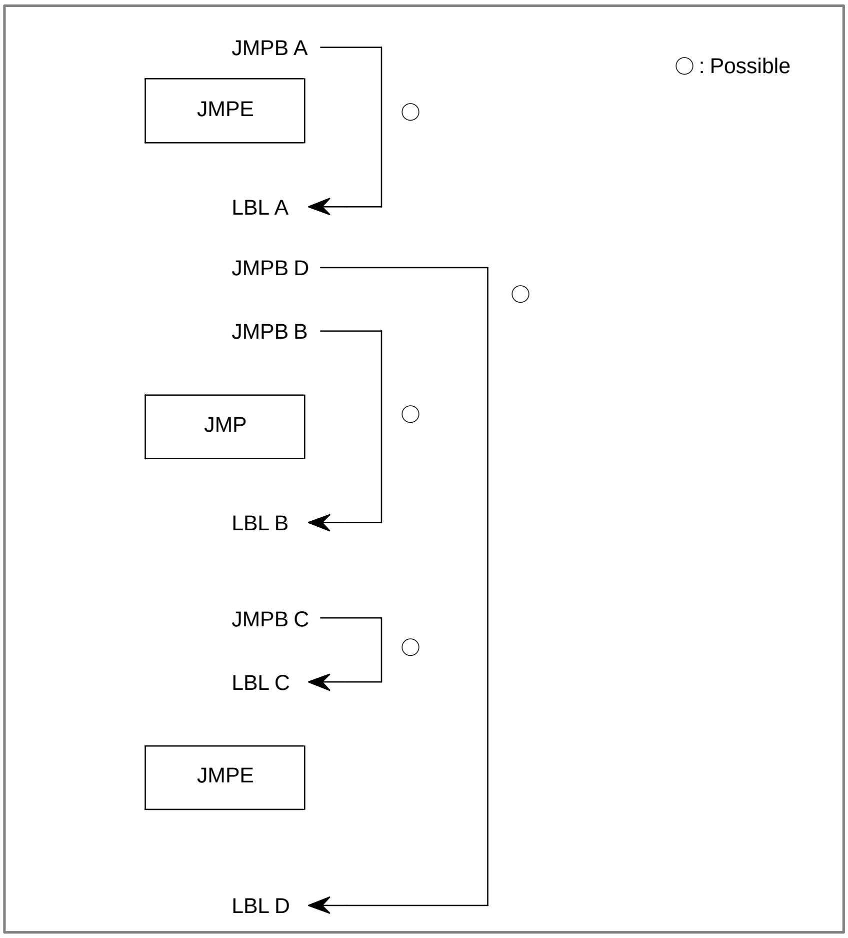
(2) Same label
(A label can be used more than once as long as it is unique within the main program or each subprogram.)



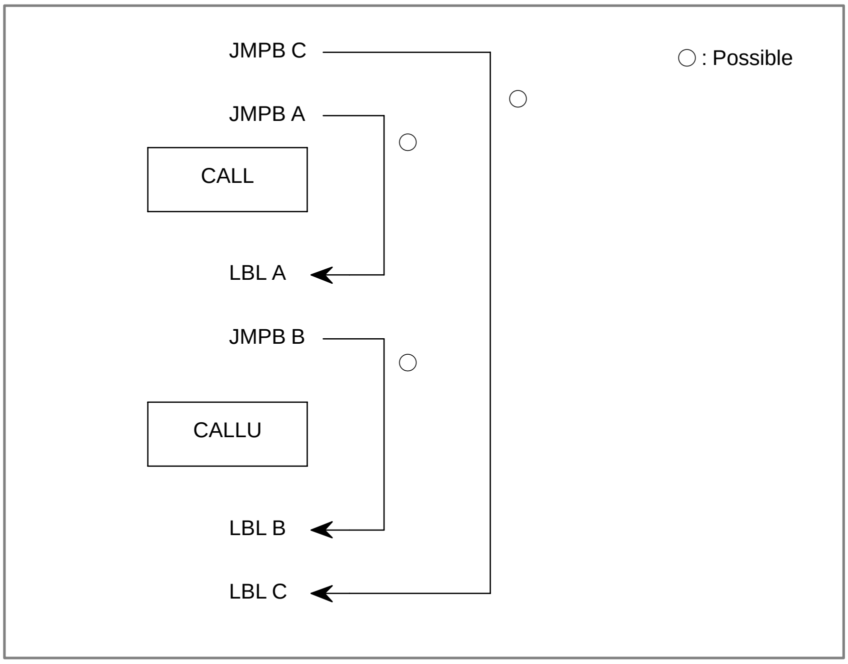
CAUTION
As mentioned in (8) of Section 10.2, the same label must not exist in the first- and second-level Ladder programs.

(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

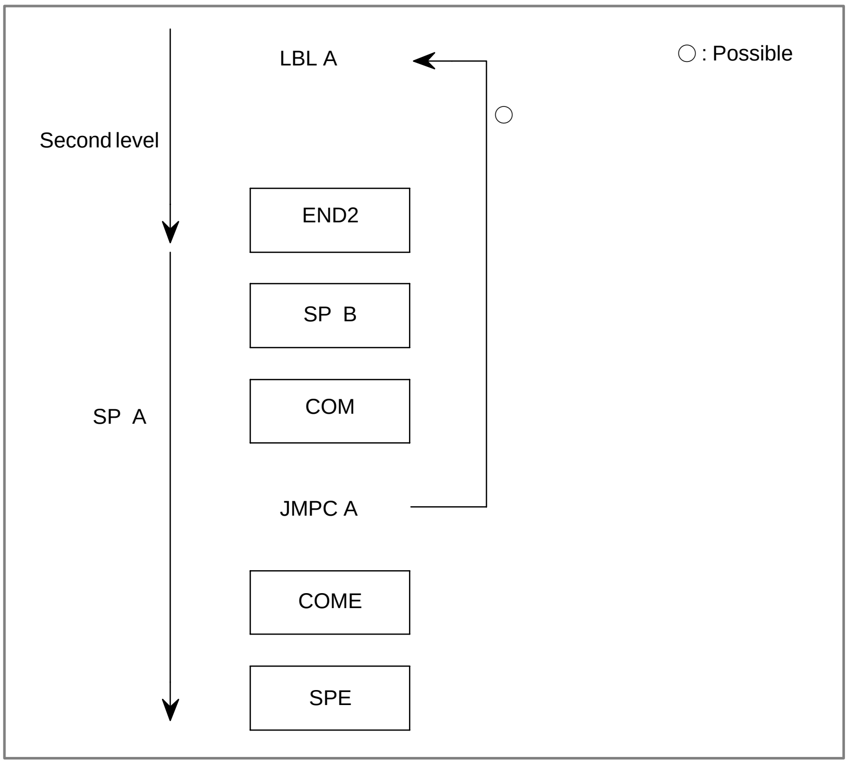
(4) Relationship between JMP/JMPE and JMPB/JMPC
(JMPB and JMPC can be used with JMP and JMPE freely.)



(5) Relationship between CALL/CALLU and JMPB/JMPC
(JMPB and JMPC can be used with CALL and CALLU freely.)

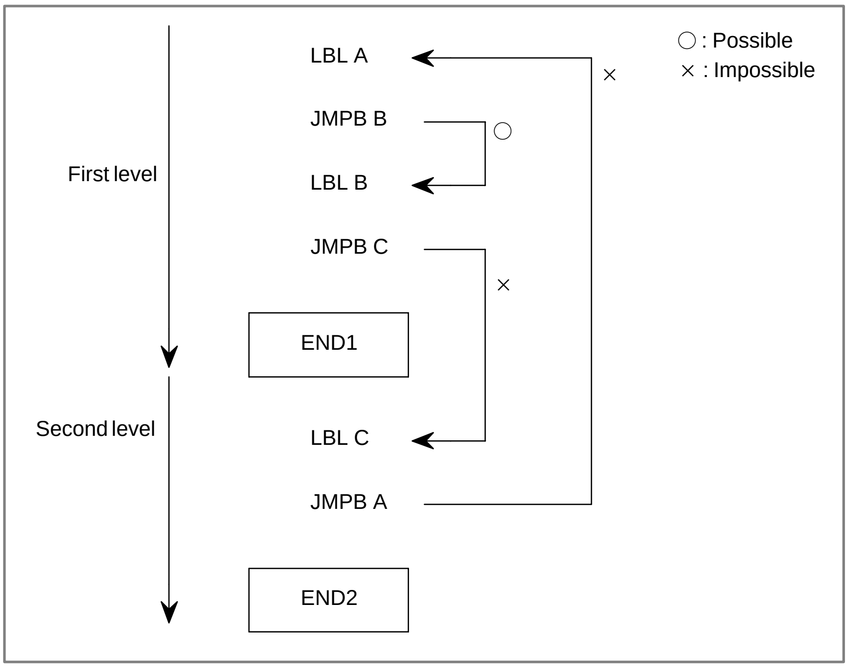


(6) Position of JMPC
(JMPC coded between COM and COME can cause a jump.)

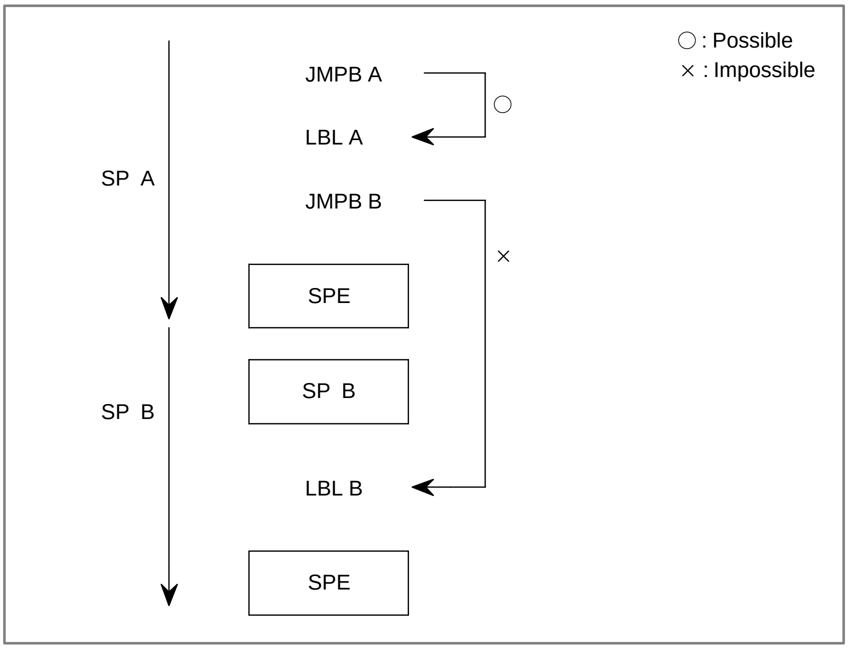


10.2 RESTRICTIONS

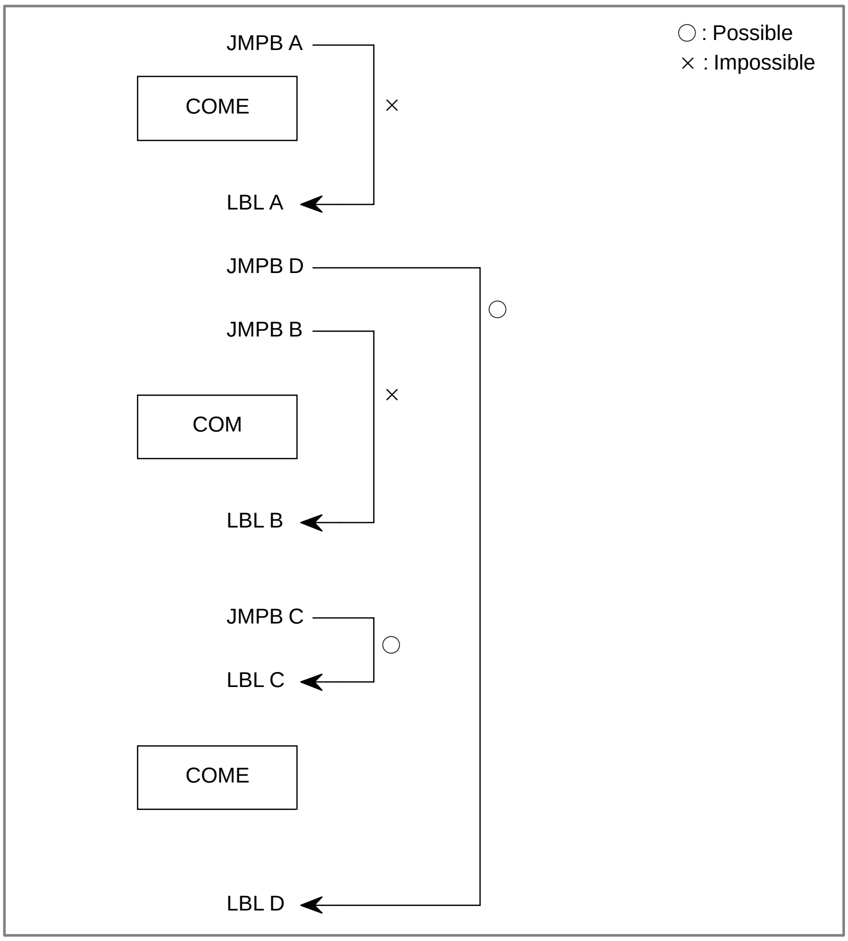
(1) Jump destination of JMPB (1)
(A jump over END1 or END2 is inhibited.)



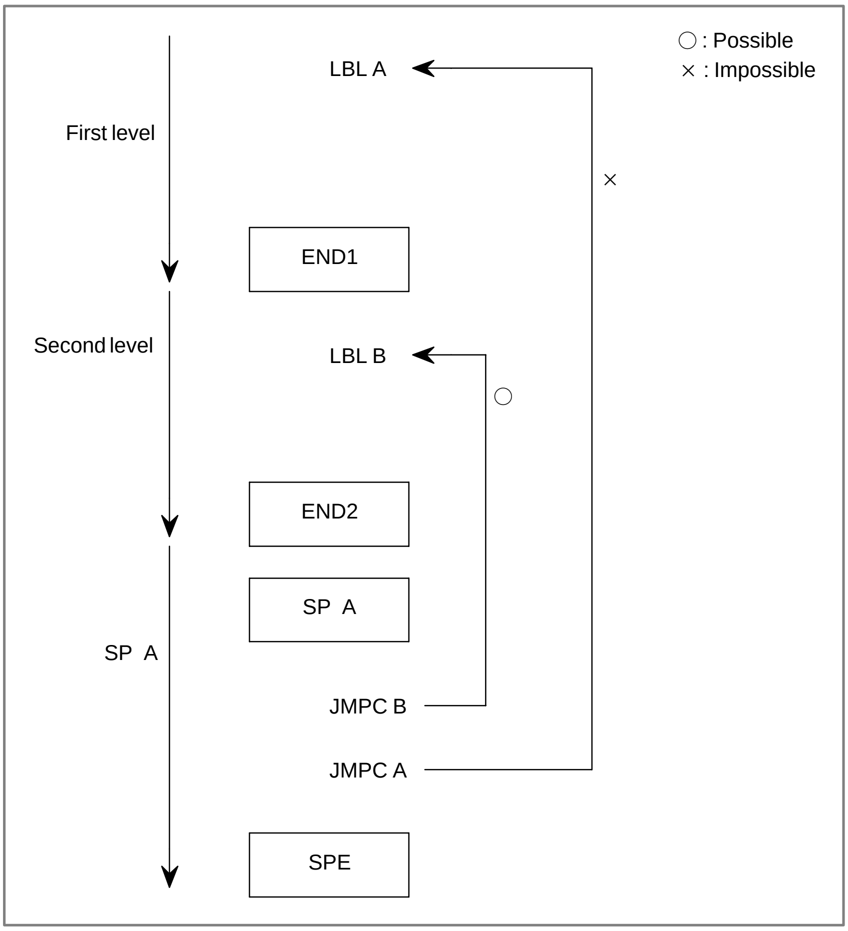
(2) Jump destination of JMPB (2)
(A jump must be performed within a subprogram.)



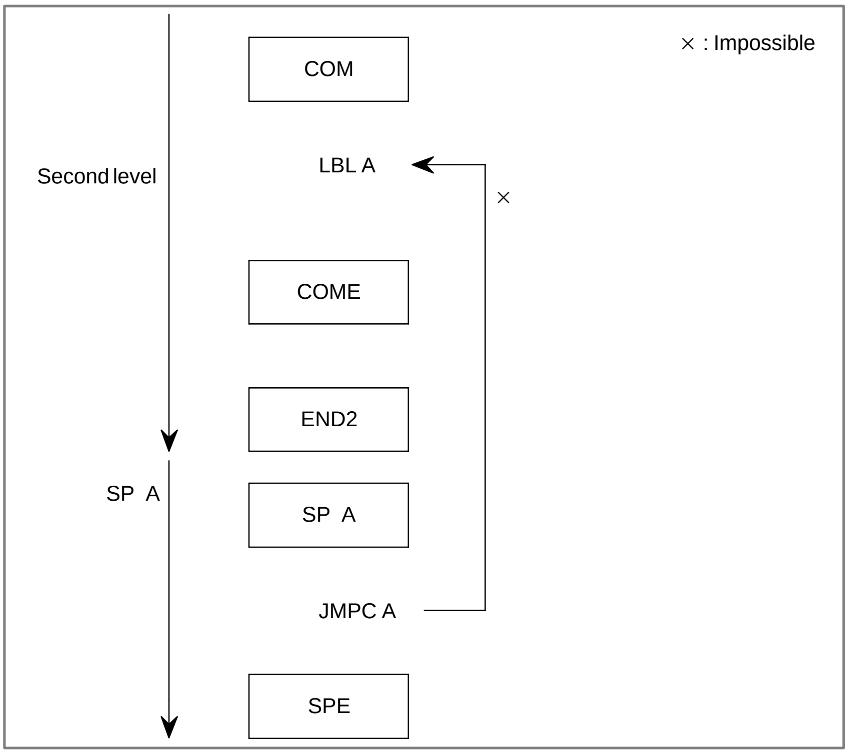
(3) Jump destination of JMPB (3)
(A jump over COM or COME is inhibited.)



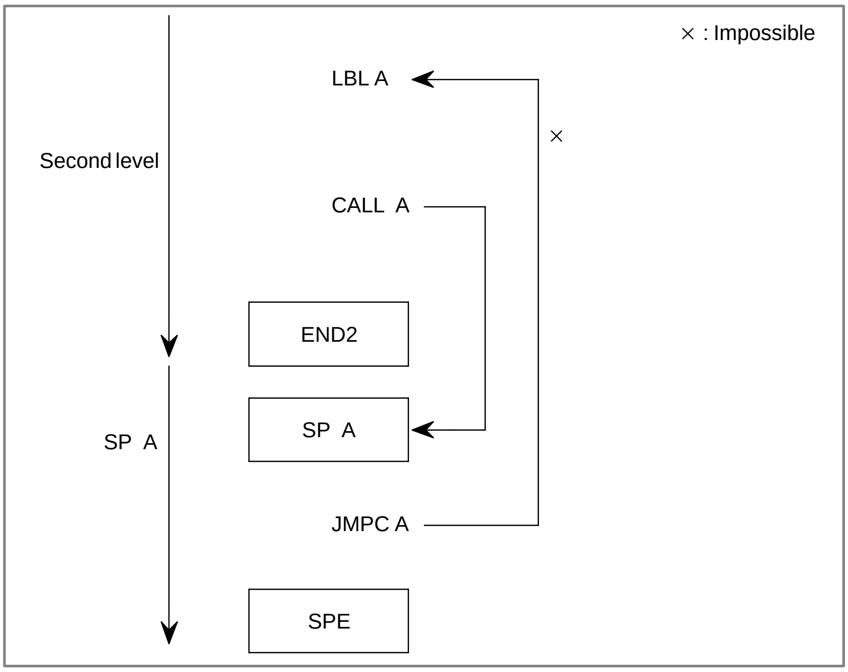
(4) Jump destination of JMPC (1)
(A jump to the first-level Ladder program is inhibited.)



(5) Jump destination of JMPC (2)
(A jump to a label between COM and COME is inhibited.)

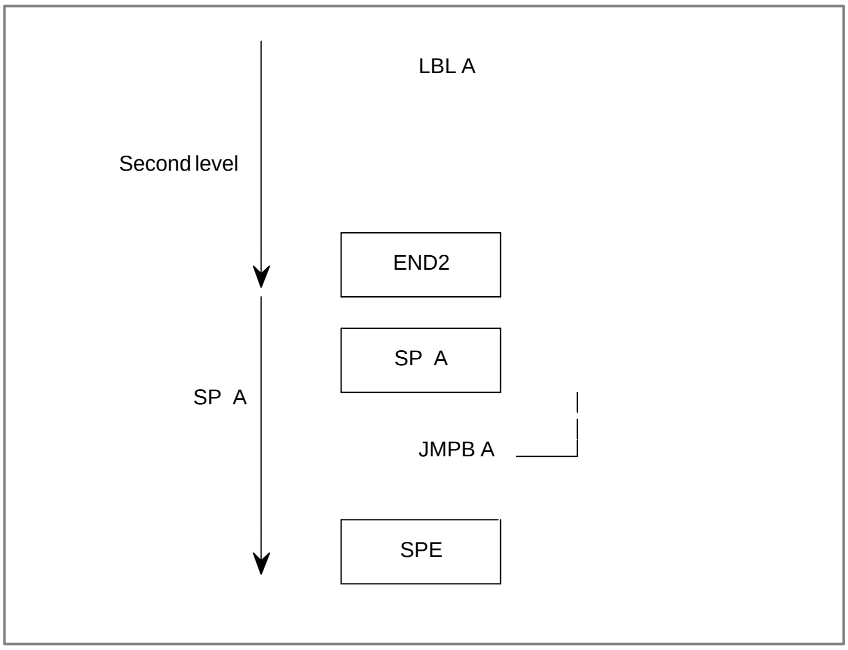


(6) Jump destination of JMPC (3)
(Control must not be returned to a label that appears earlier than the instruction that has called the subprogram.)

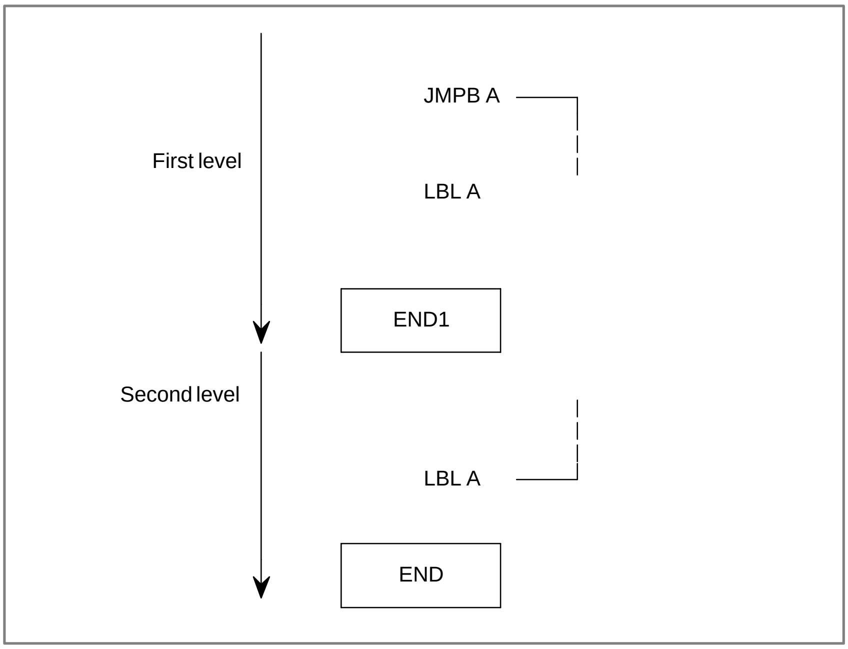


CAUTION
Although Ladder diagrams can be edited, editing a Ladder diagram may cause an infinite loop. So, be careful not to program such processing.

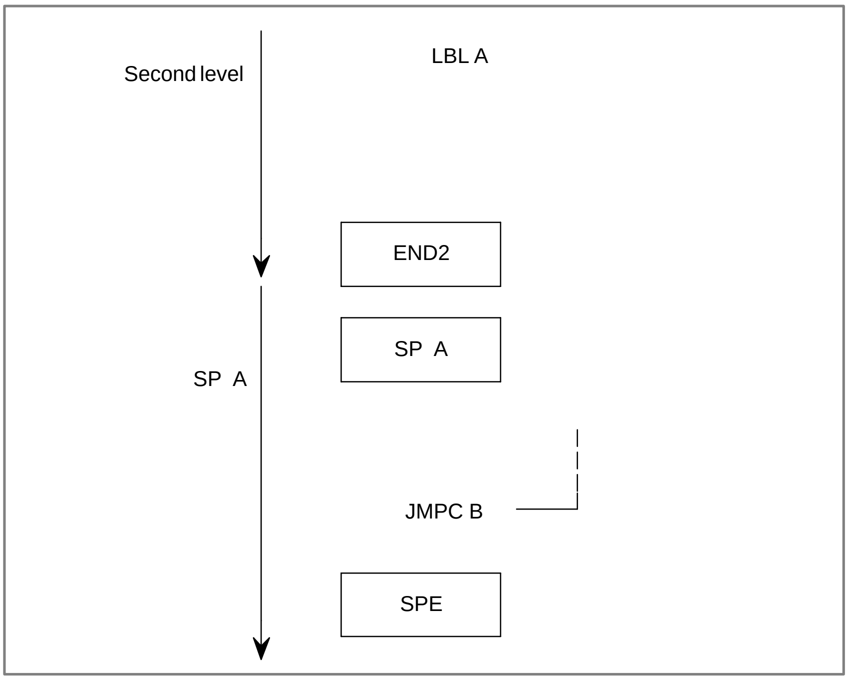
(7) LBL for JMPB (1)
(There is no LBL in the same subprogram.)



(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).

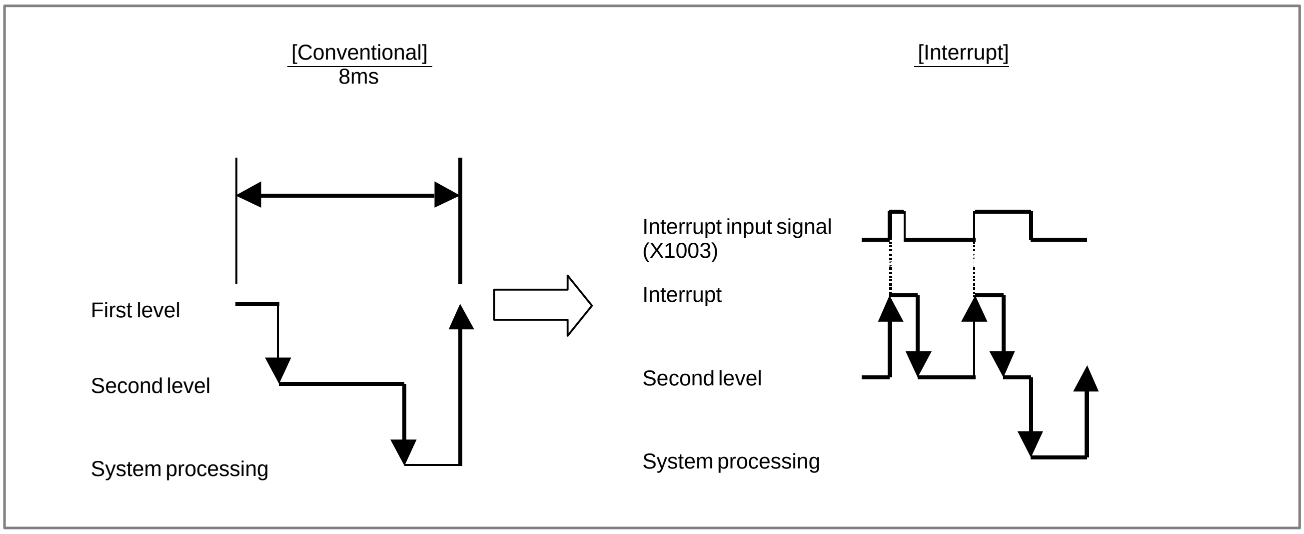


Fig. 11.1 Sequence program execution order

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."

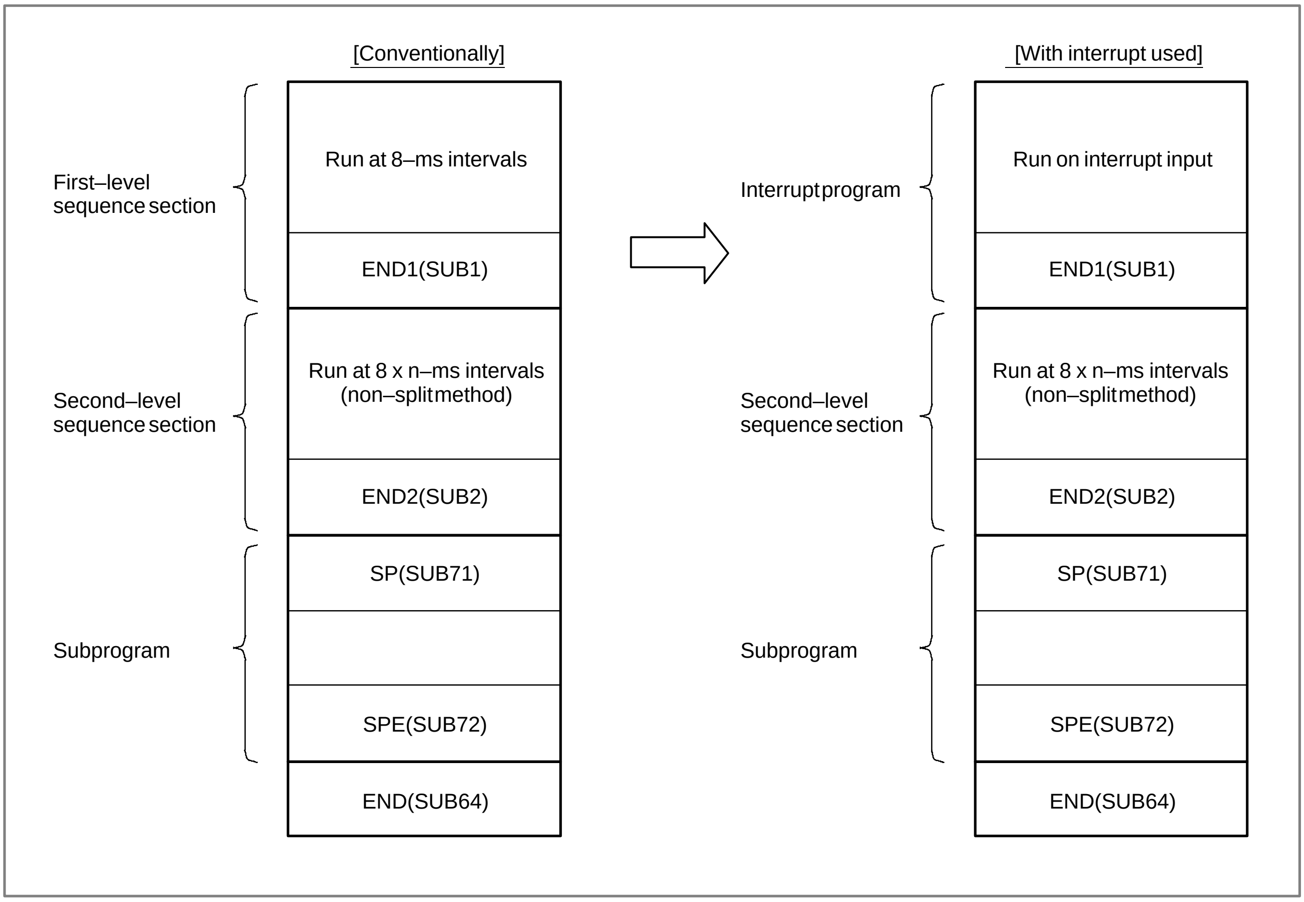


Fig. 11.3.1 Sequence program configuration

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]

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]

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

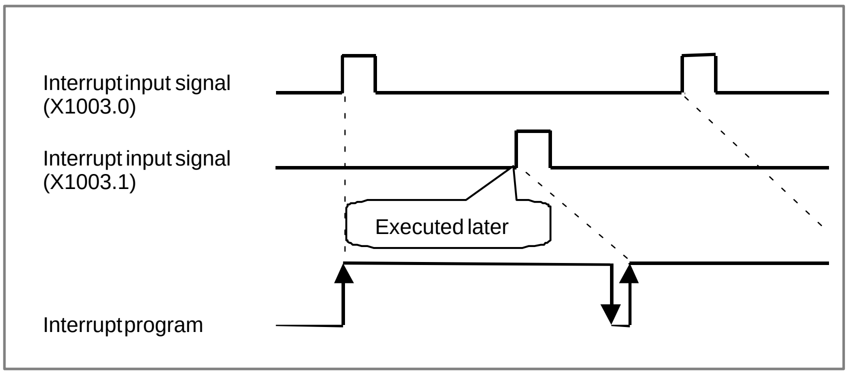


Fig. 11.3.6 (a) Continuous interrupt operations triggered by different signals

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

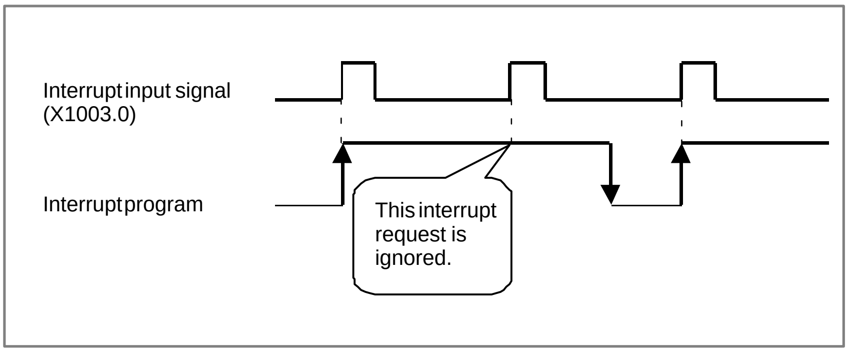
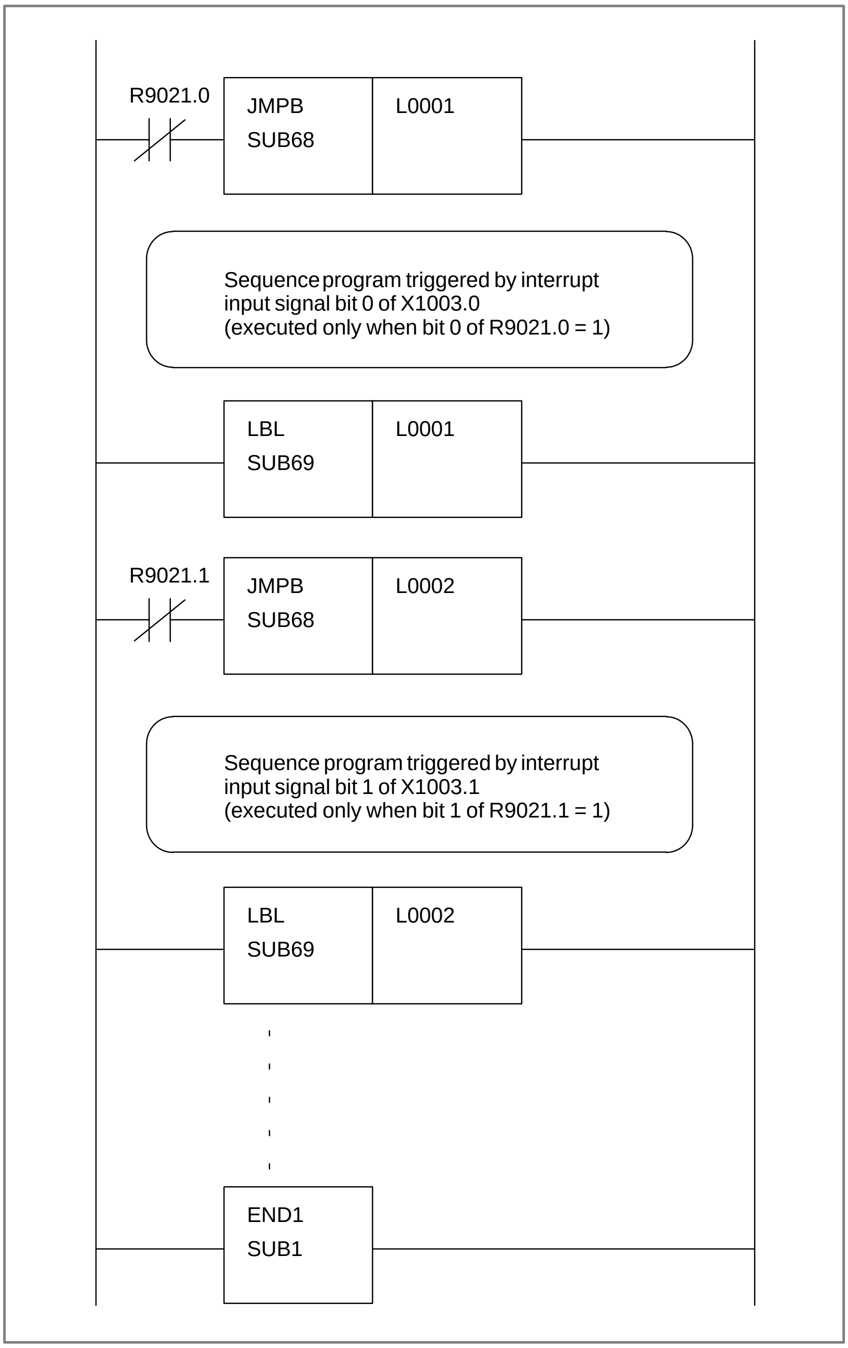


Fig. 11.3.6 (b) Continuous interrupt operations triggered by the same signal

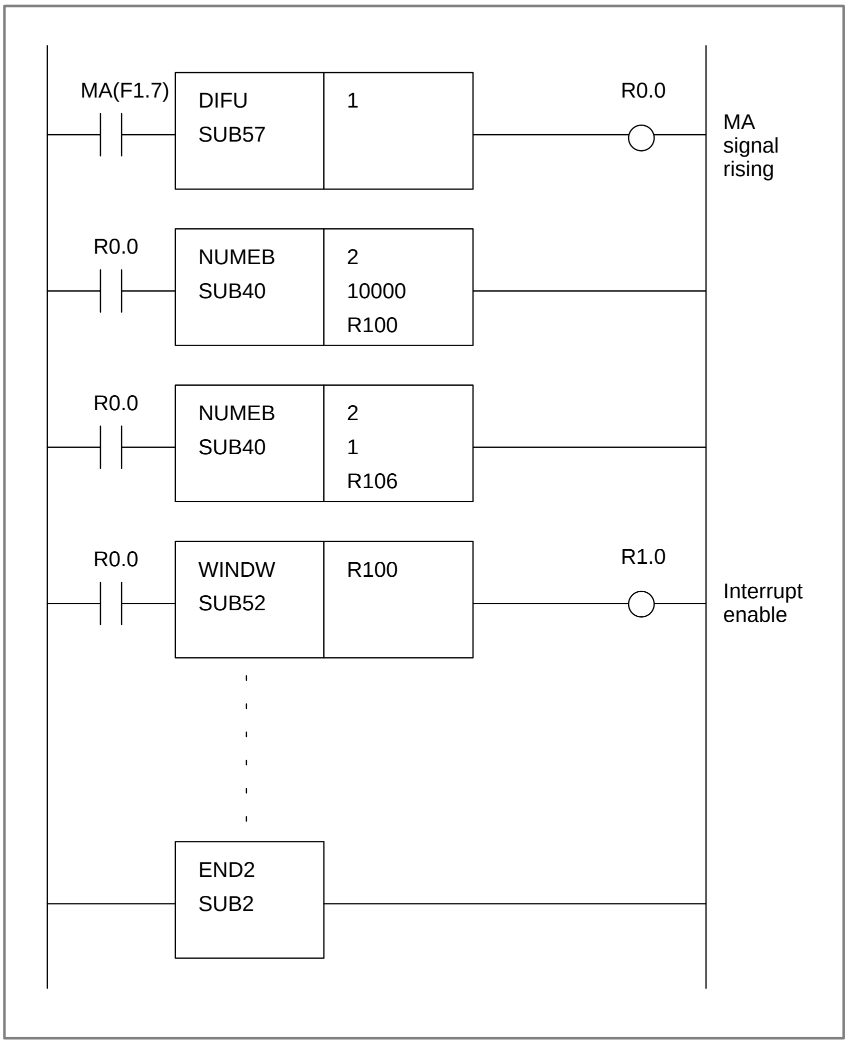
If both the rising and falling edges of a certain interrupt input signal are specified for interrupt, the falling (rising) edge is ignored if it is detected during the interrupt program execution requested on the rising (falling) edge. So, it is necessary to complete the interrupt program execution before the interrupt input signal changes.

11.4 SEQUENCE PROGRAM EXAMPLES

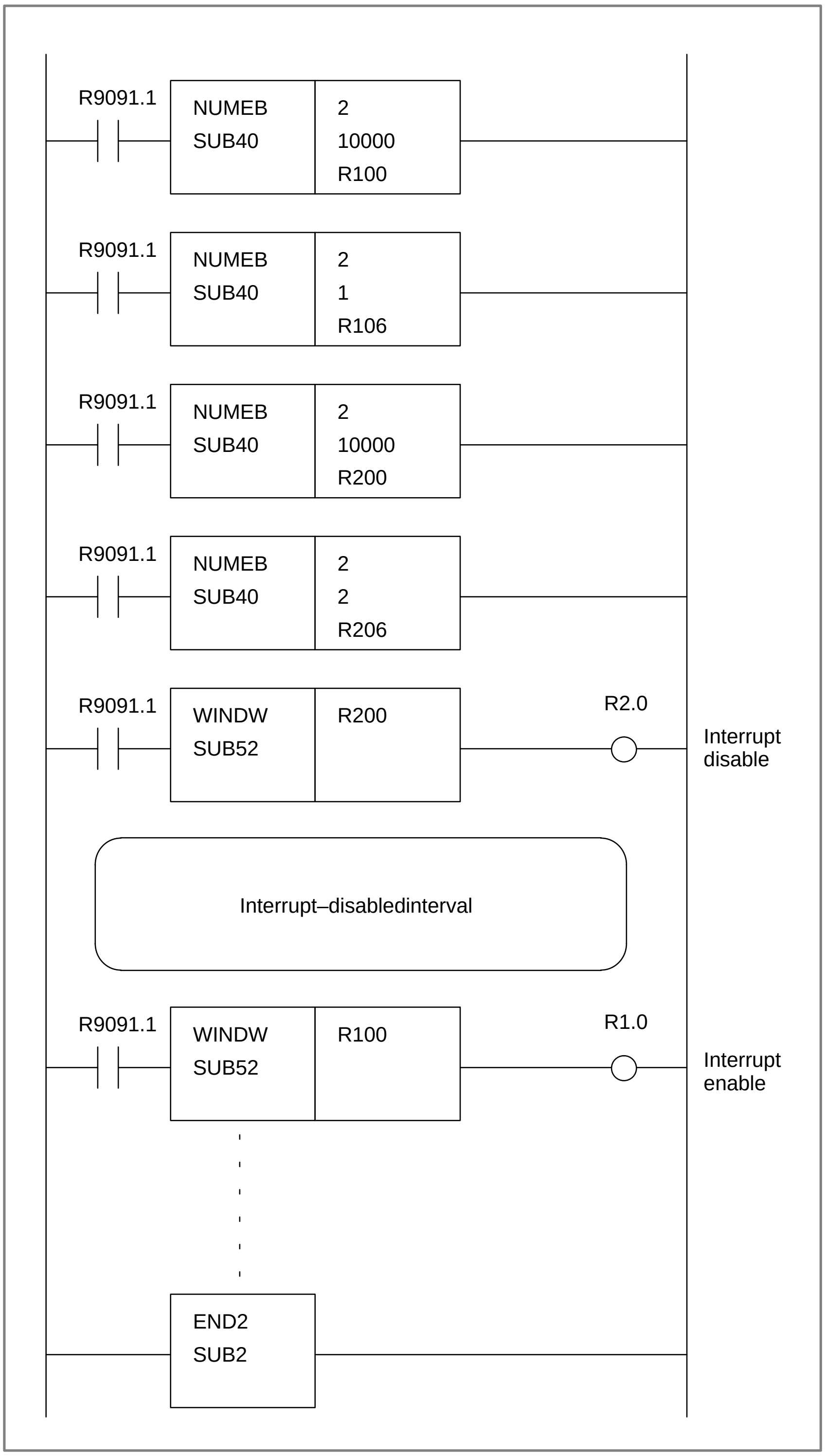
(1) Interrupt program that handles interrupt requests separately
Use of R9021 together with a label jump enables processing interrupt requests separately as shown below.



(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

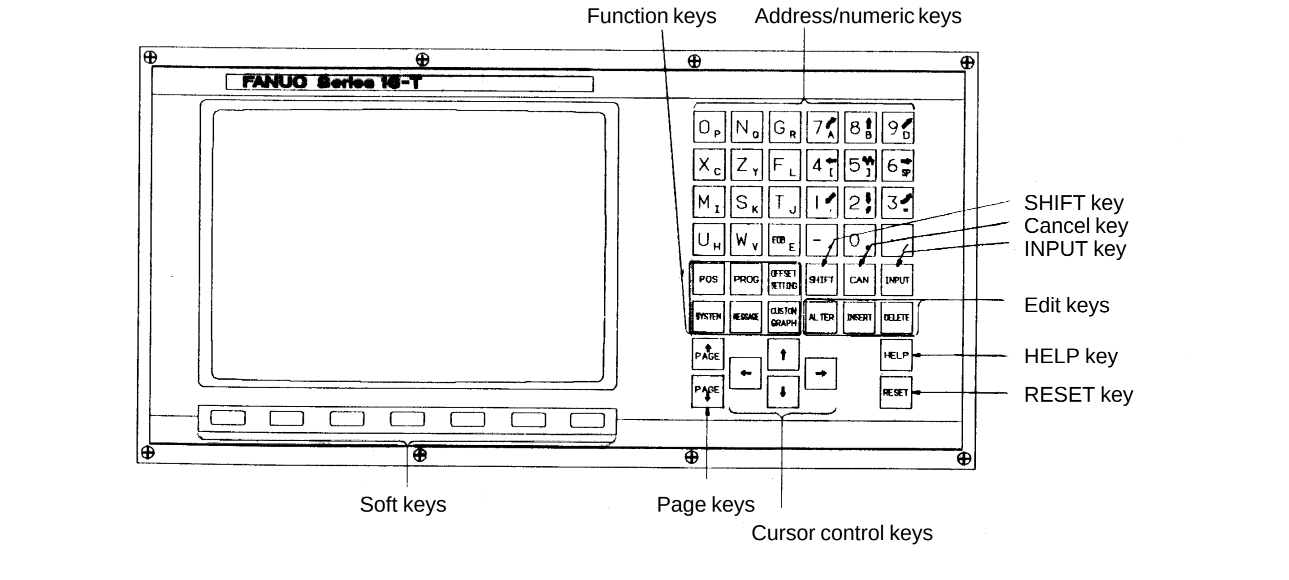
Press the SYSTEM function key on the CRT/MDI panel. Selecting the PMC soft key displays the PMC basic menu.
PMC screen to NC screen

- 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.

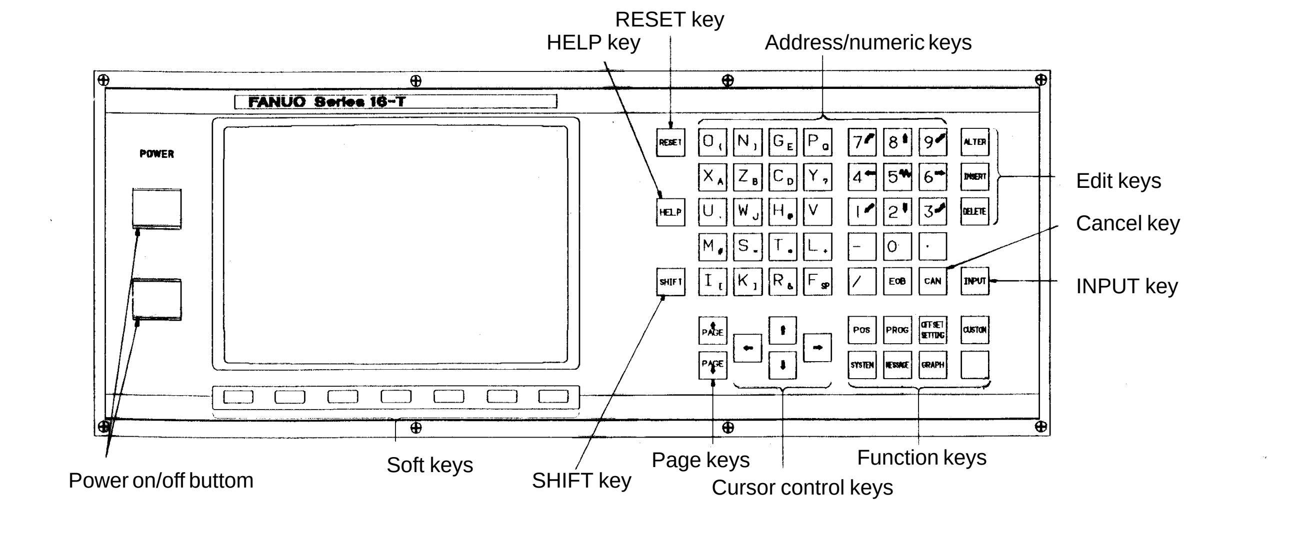
Figs. 1 a) to 1 l) show the standard CRT/MDI panels.

> **NOTE**
> A key in < > is a function key on the CRT/MDI panel. A key in [ ] is a soft key described below.

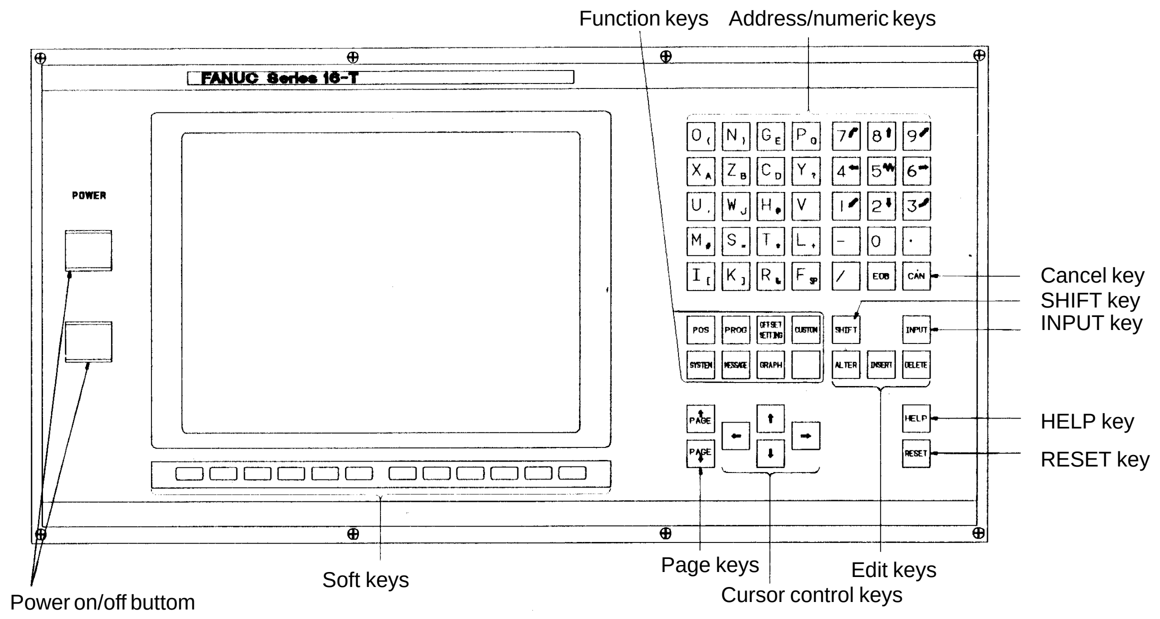
a) 9″ small monochrome/color CRT/MDI panel for 16–TA/18–TA (Horizontal type)



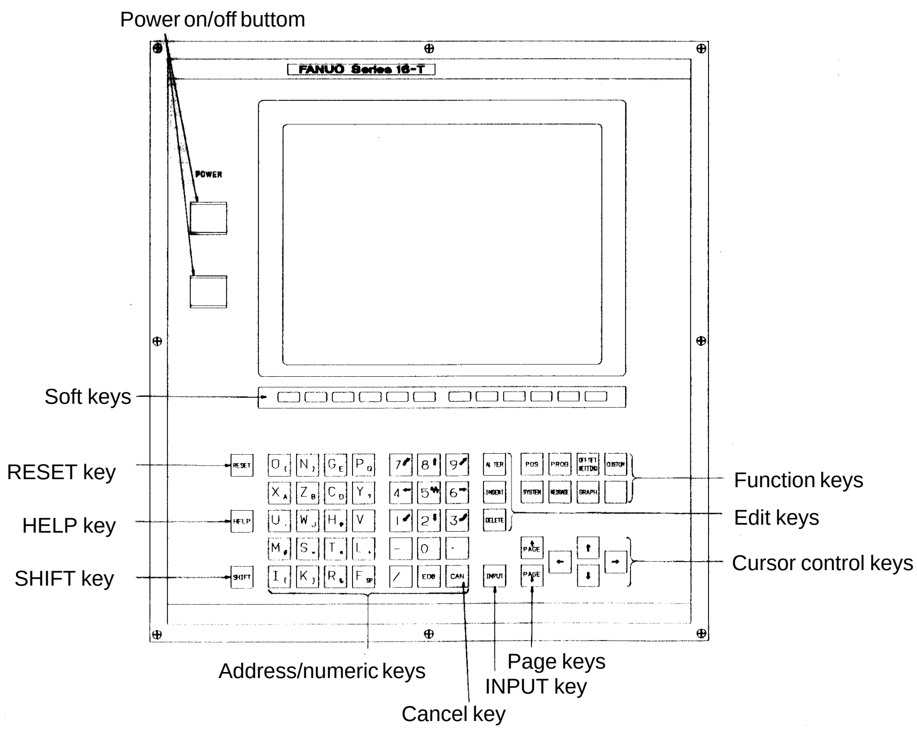
b) 9″ monochrome/color CRT/MDI panel for 16–TA/18–TA (Horizontal type)



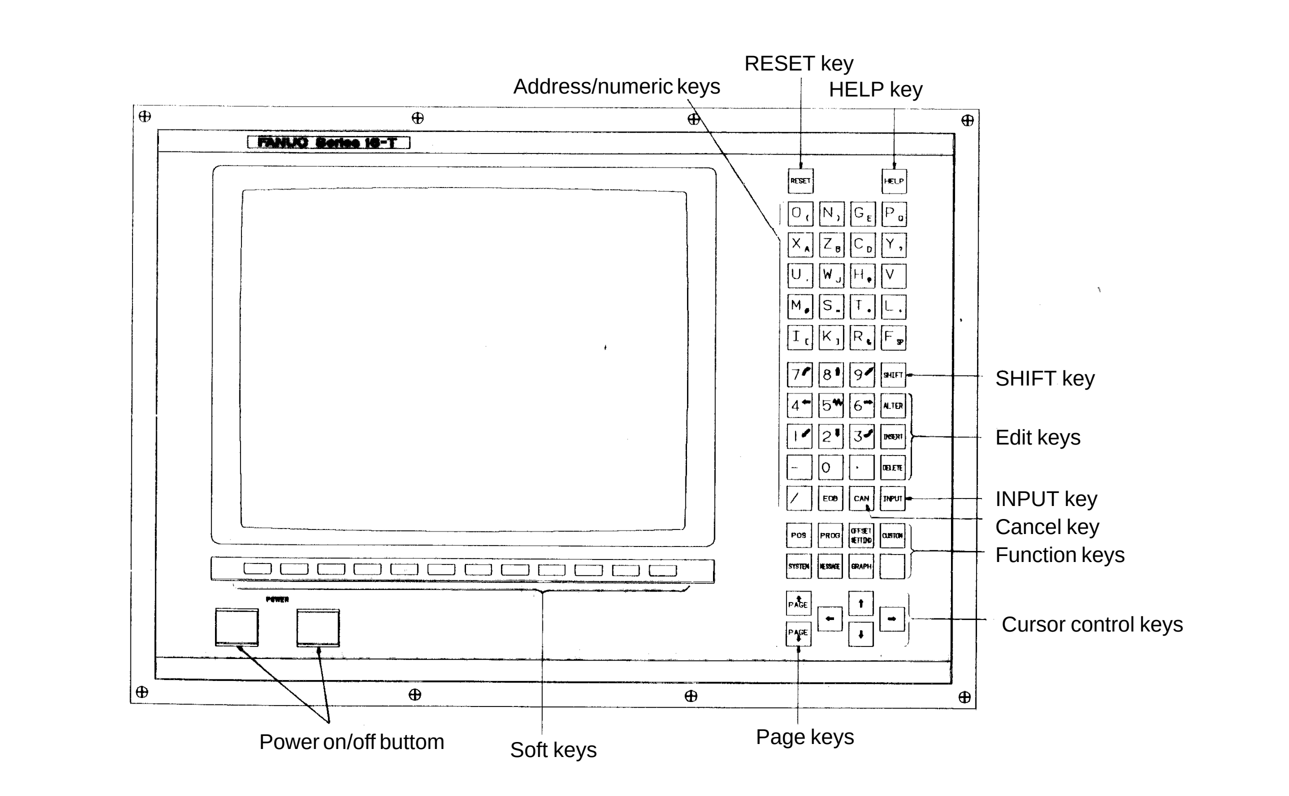
c) 10″ color LCD/MDI panel for 16–TA/18–TA (Horizontal type)



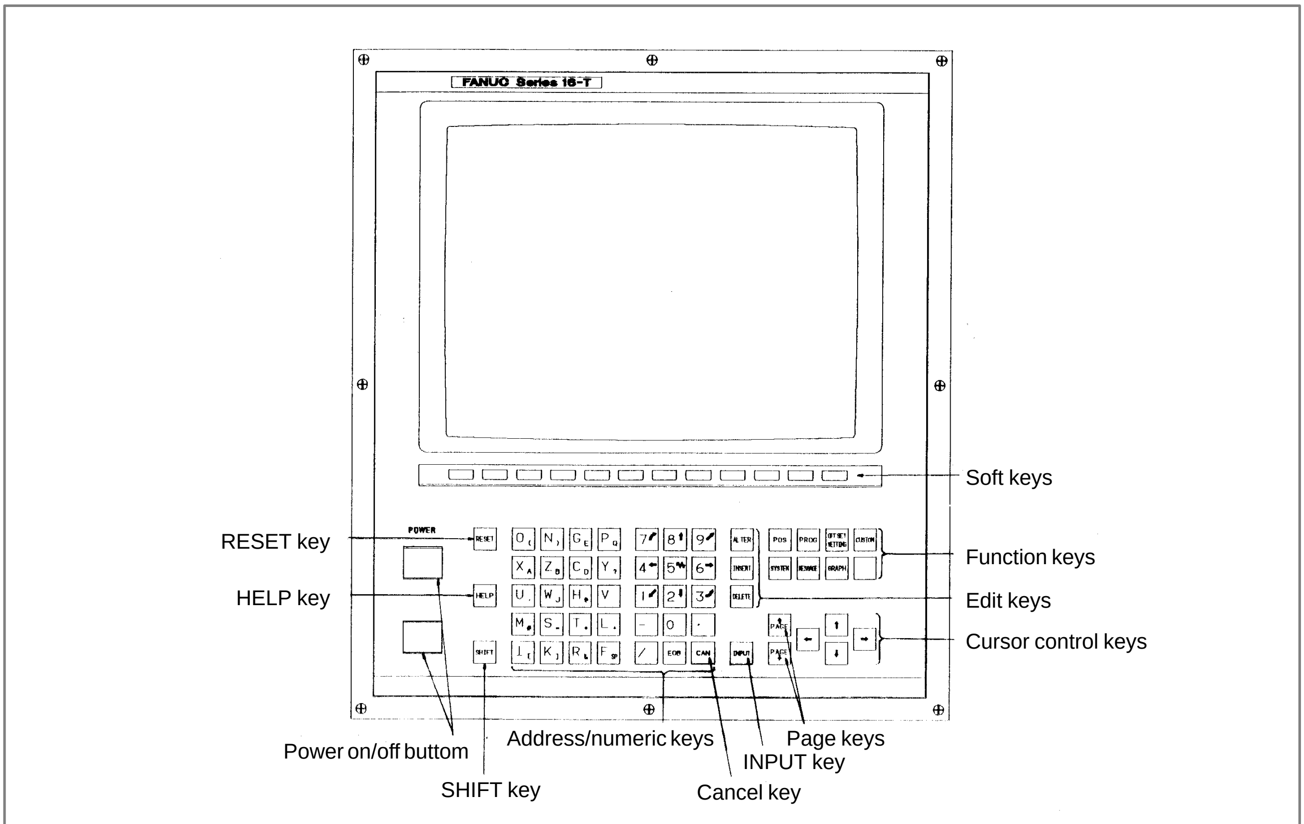
d) 10″ color LCD/MDI panel for 16–TA/18–TA (Vertical type)



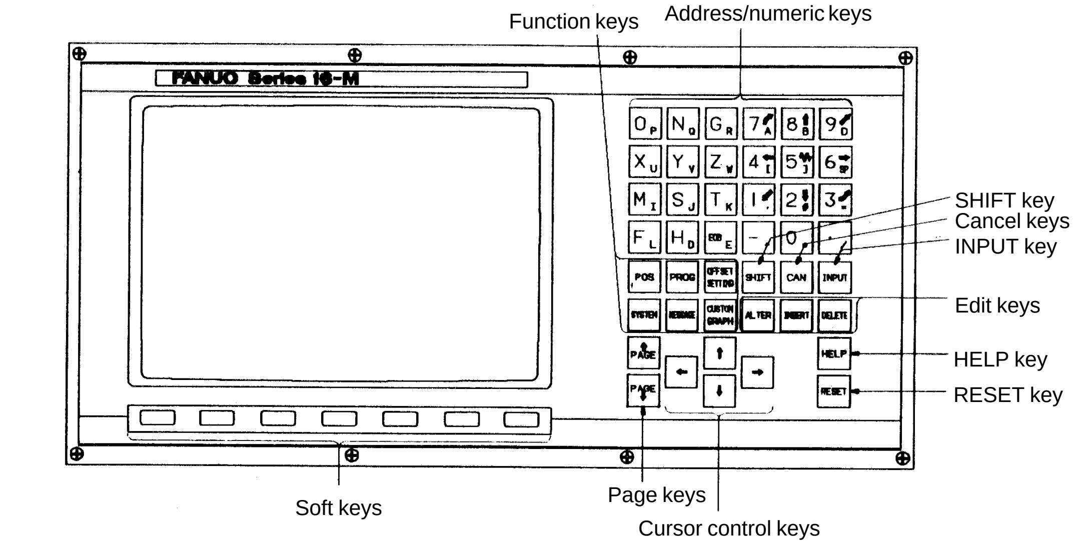
e) 14″ color CRT/MDI panel for 16–TA/18–TA (Horizontal type)



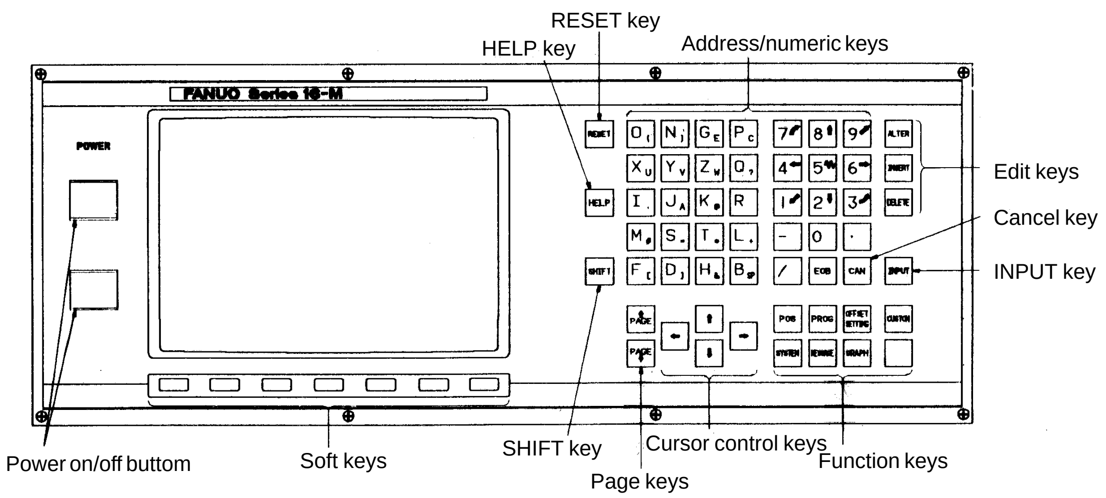
f) 14″ color CRT/MDI panel for 16–TA/18–TA (Vertical type)



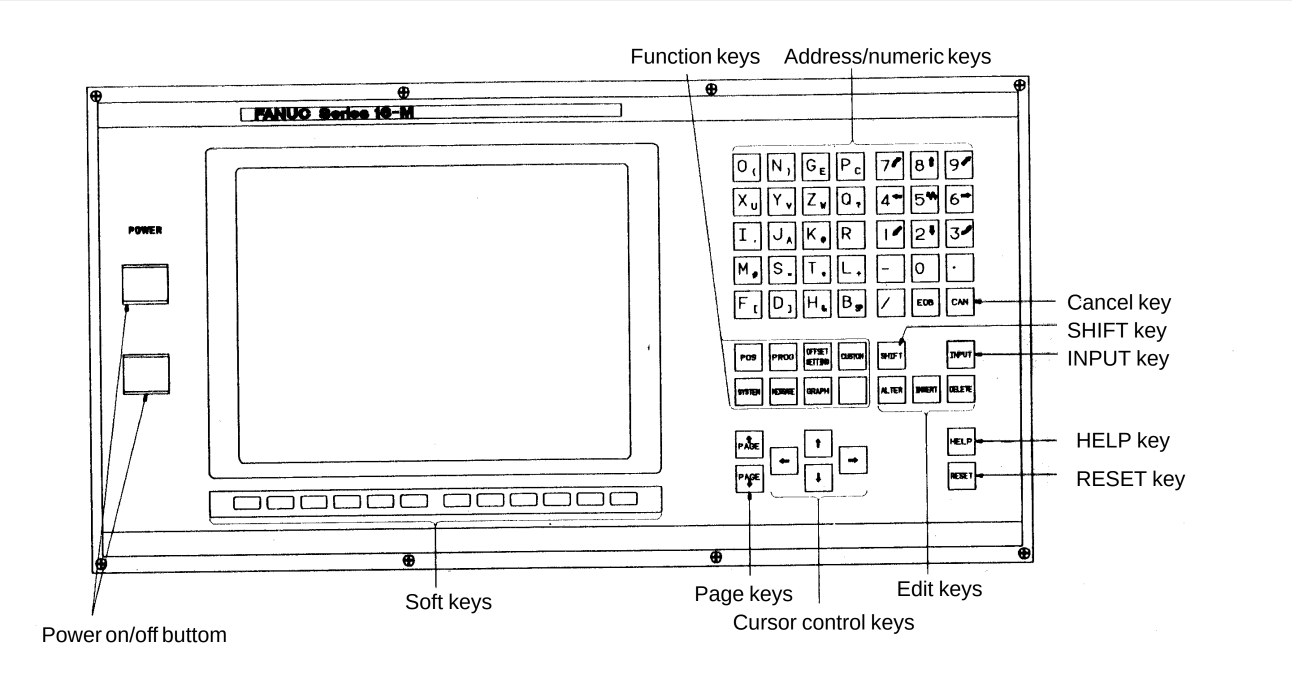
g) 9″ small monochrome/color CRT/MDI panel for 16–MA/18–MA (Horizontal type)



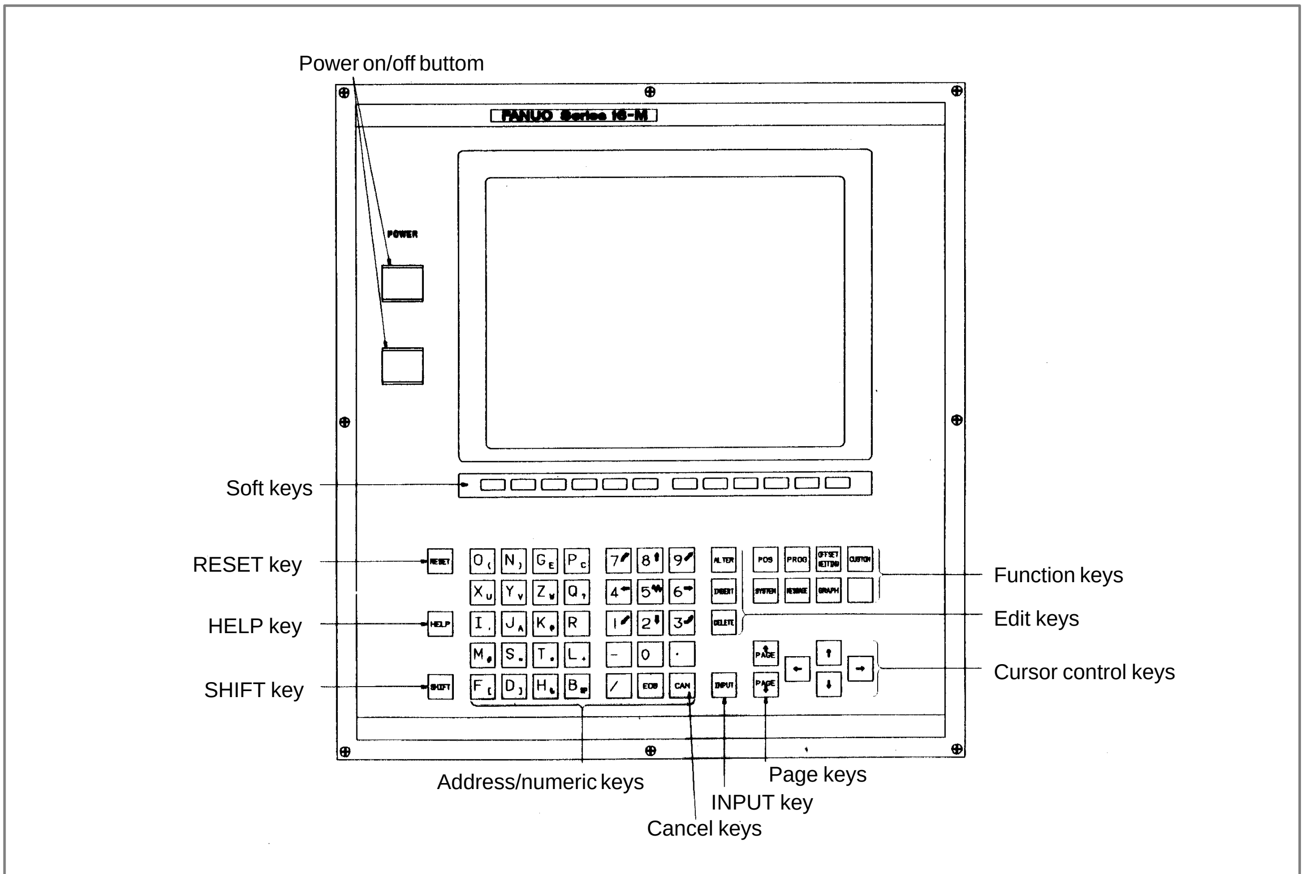
h) 9″ monochrome/color CRT/MDI panel for 16–MA/18–MA (Horizontal type)



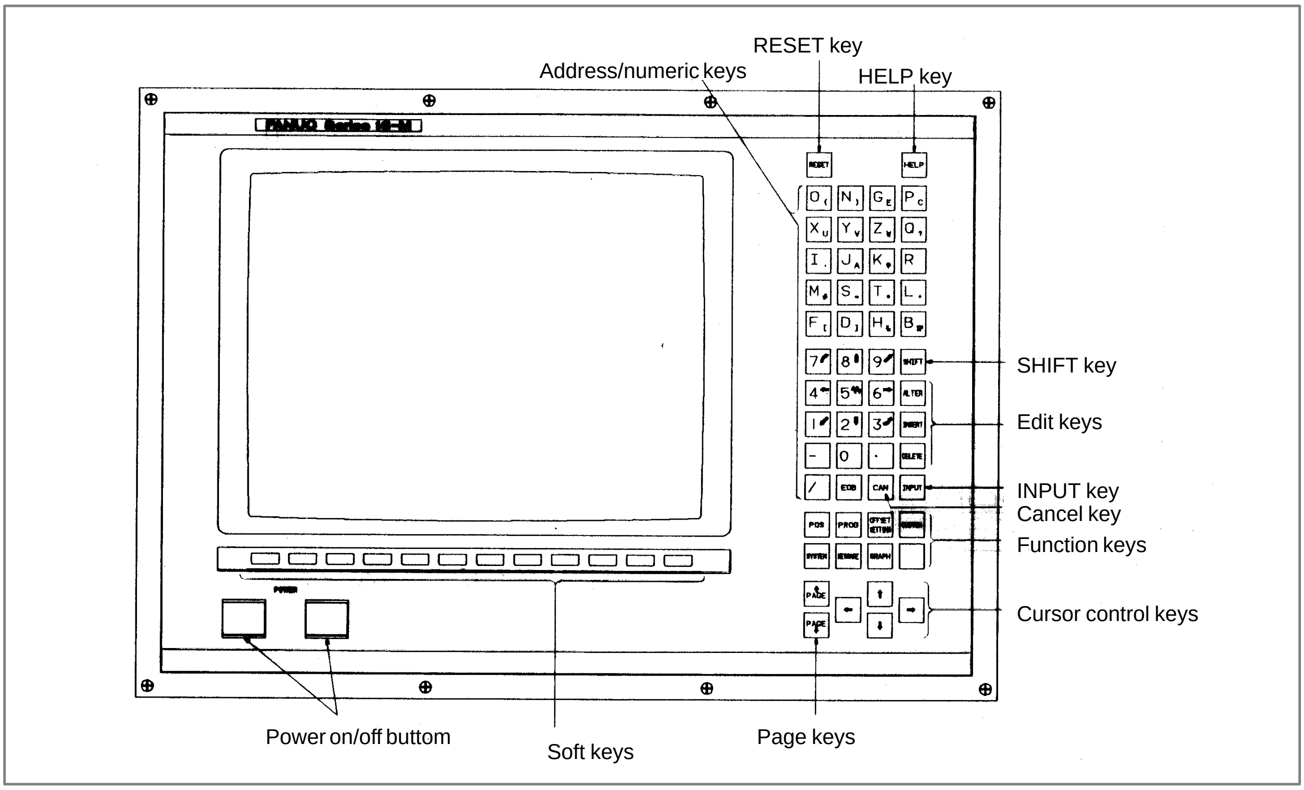
i) 10″ color LDC/MDI panel for 16–MA/18–MA (Horizontal type)



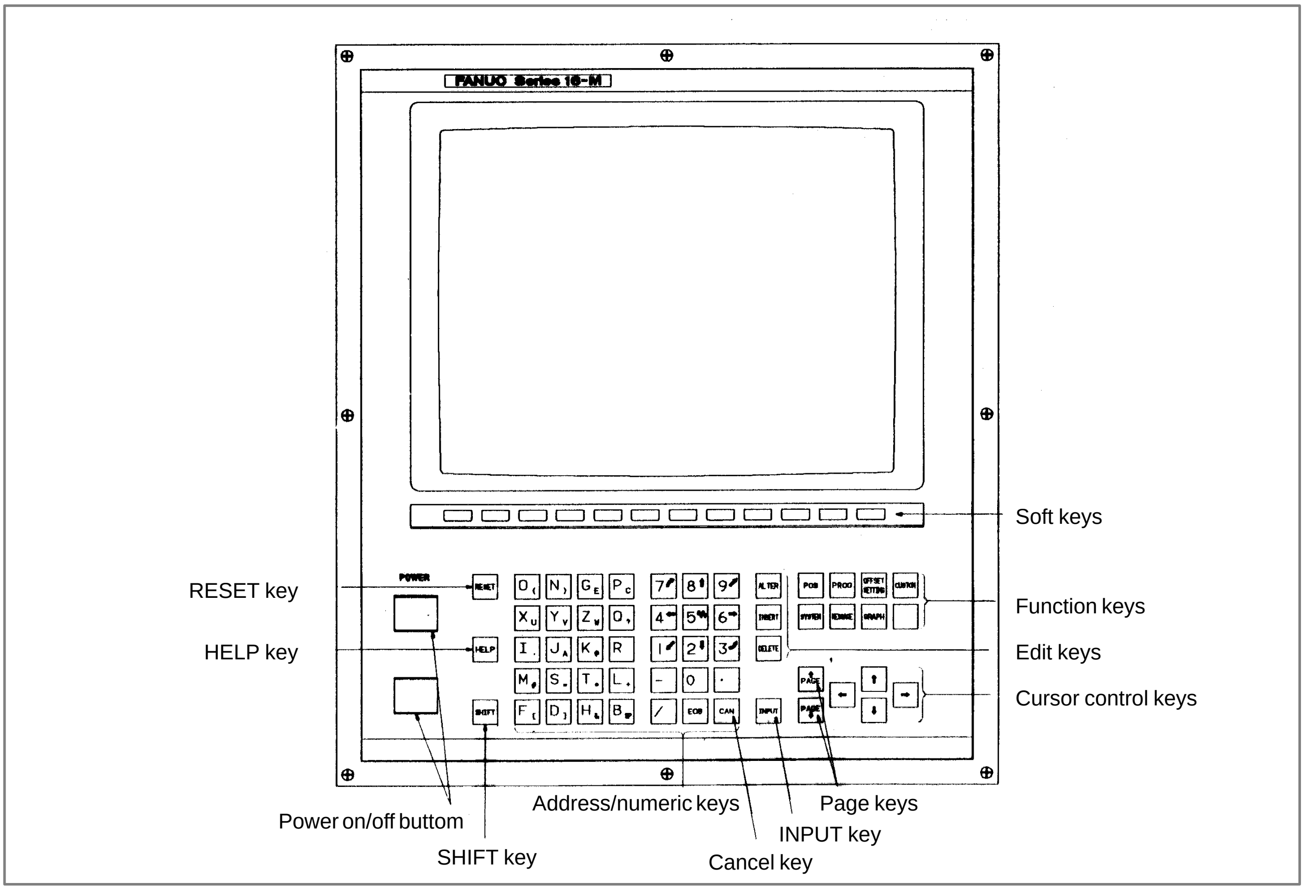
j) 10″ color LCD/MDI panel for 16–MA/18–MA (Vertical type)



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.

CAUTION
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 Part IV, ”PMC–NB6 Manipulation” for an explanation of how to operate the FS15*i* PMC–NB6.

1.6 FS16*i*/18*i*/21*i*–B PMC–SA1/SB7 OPERATING PROCEDURE

See Part V, ”PMC–SA1/SB7 Manipulation” for an explanation of how to operate the FS16*i*/18*i*/21*i*–B PMC–SA1/SB7.

1.7 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
(The 4047 series is not supported.)
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 1.7 (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 1.7 (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 1.7 (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	:	5.6.1 Clear the sequence program in Part III
CLRALL	:	5.6.1 Clear the sequence program 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 1.7 (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.

1.8 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/C), 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.

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.

```
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] [       ] [       ]
```

Built–in programmer function

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.

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.5.25 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.5.25 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.

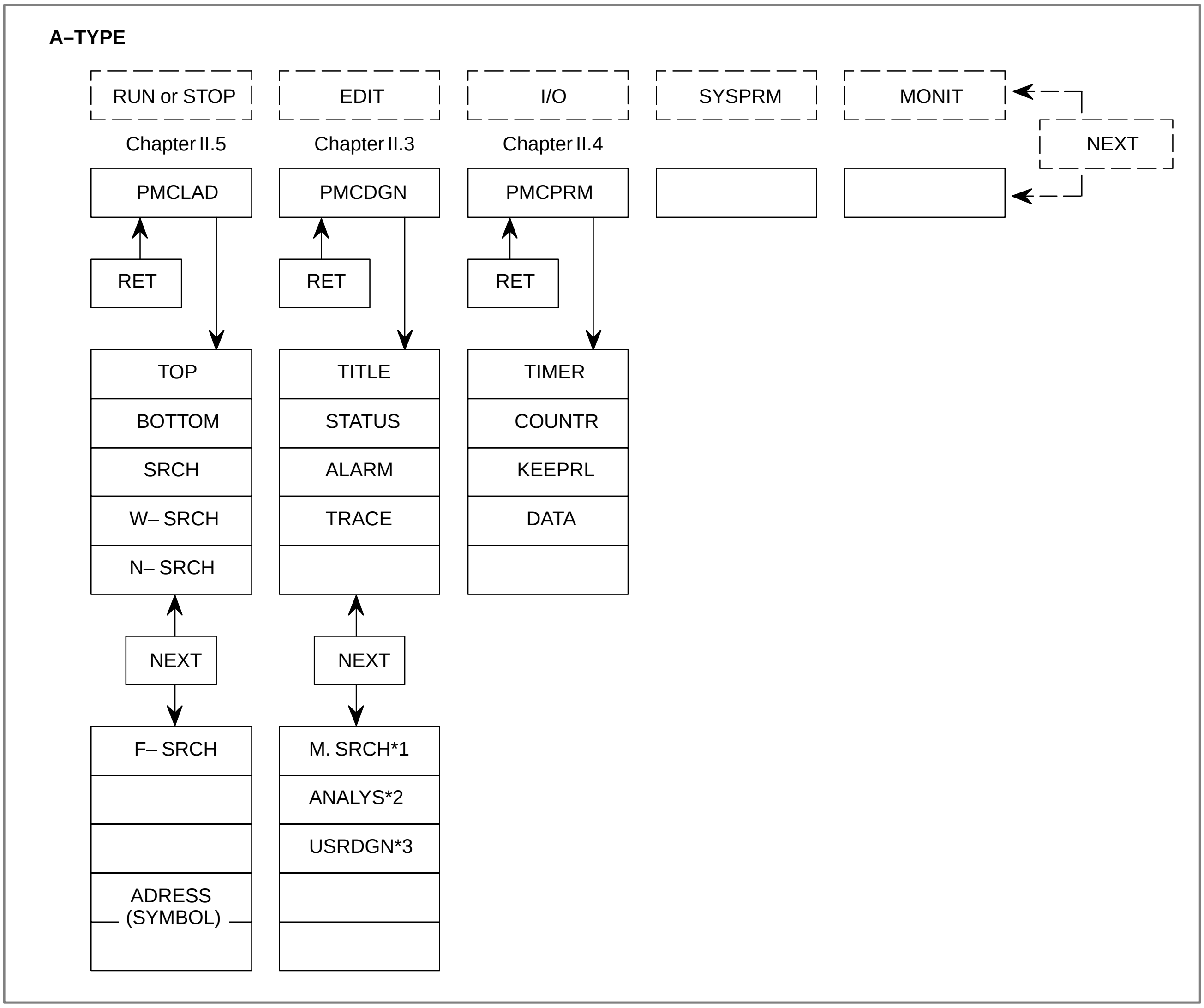
3) Status display

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.

- 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.

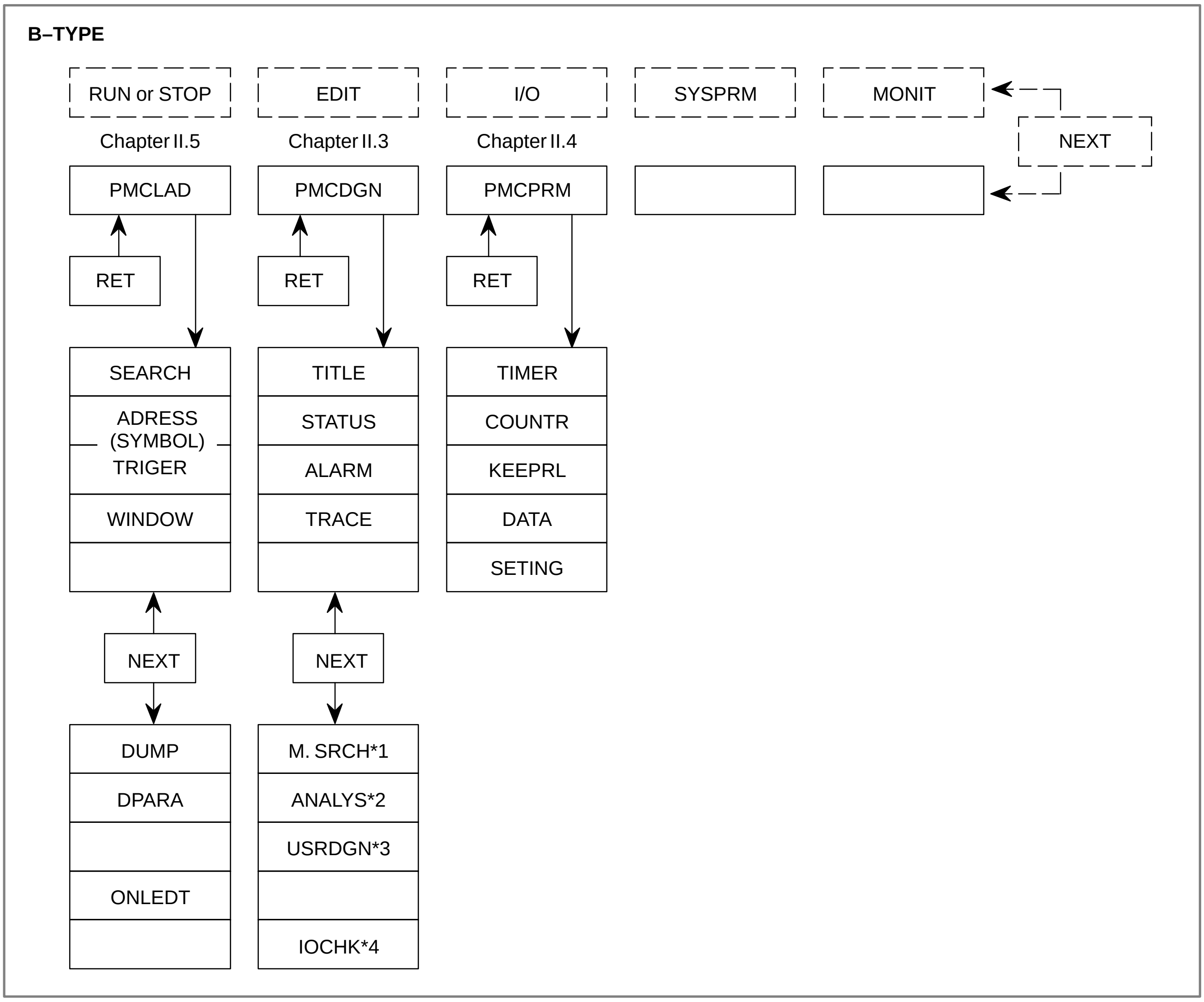
4) Relation between PMC menus and soft keys

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

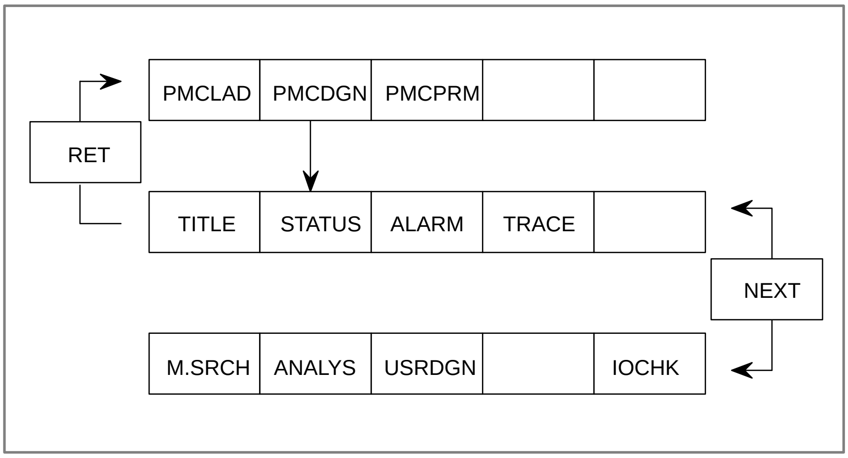
(Reference) Series of PMC control software is displayed on the [PMCDGN] and [TITLE] screen as shown below.

```
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 ] [       ]
```

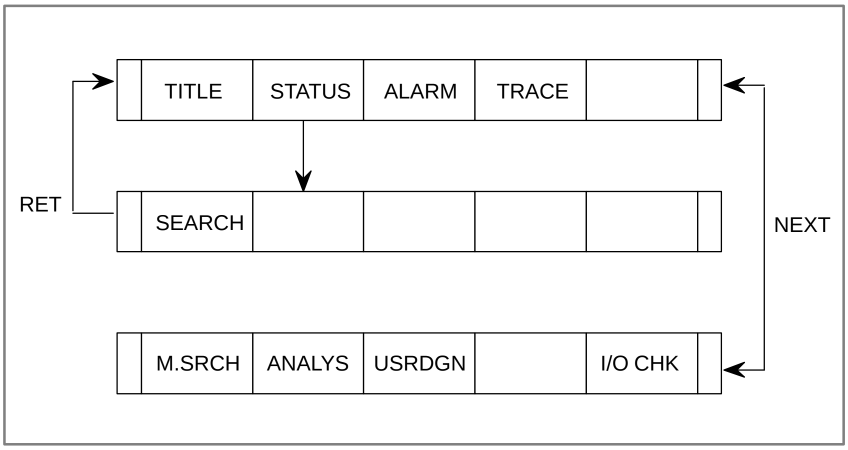
Fig. 3.1 (b)   Title data 2

```
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.

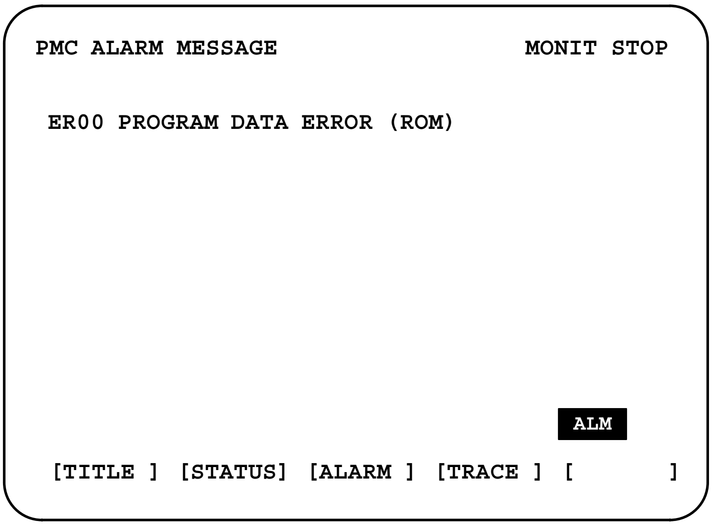


Fig. 3.3 Alarm screen

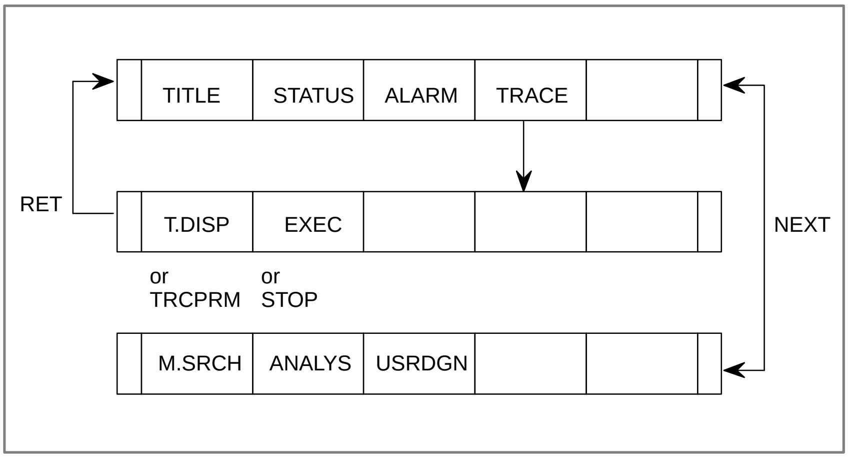
For displayed messages, see the appendix, “Alarm Message List.”

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. 3.5.2. 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.

> **WARNING**
> 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
×	×	○	×	×	△	△	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○	◎	◎	◎	◎

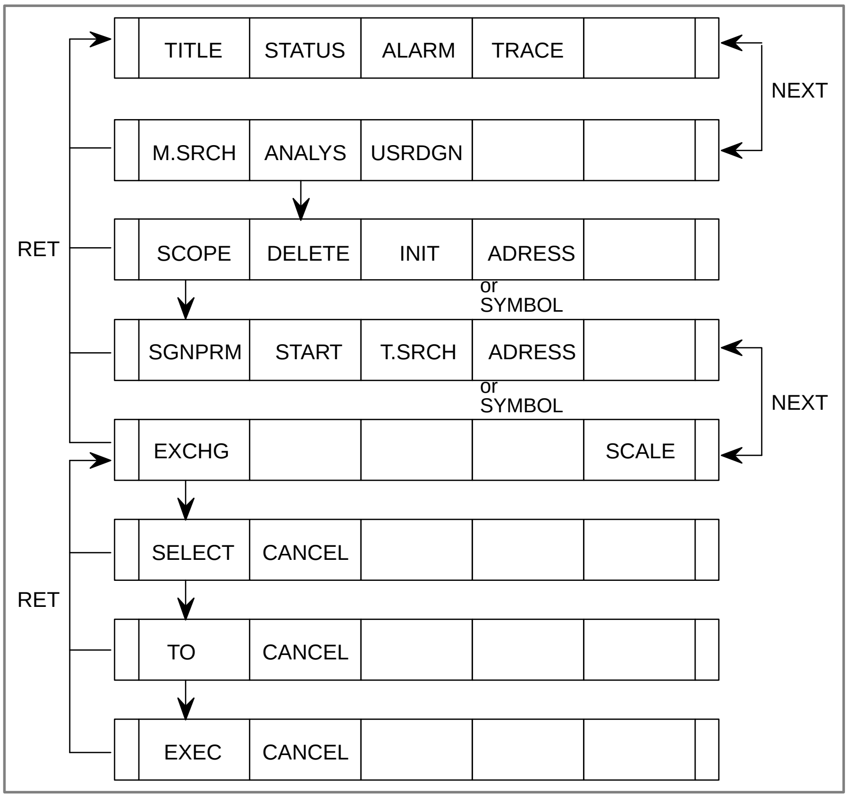
- The ladder edit module is required with PMC–SB/SB2/SB3 of the Series 16–MODEL A, and PMC–SA2/SA3 of the Series 18–MODEL A.
- The ladder edit card is required with PMC–RA1/RB3/RB4 of the Series 16/18–MODEL B, PMC–SA1/SB5/SB6 of the Series 16/18–MODEL C, PMC–SA1/SA3 of the Series 21/210–MODEL B, and 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.

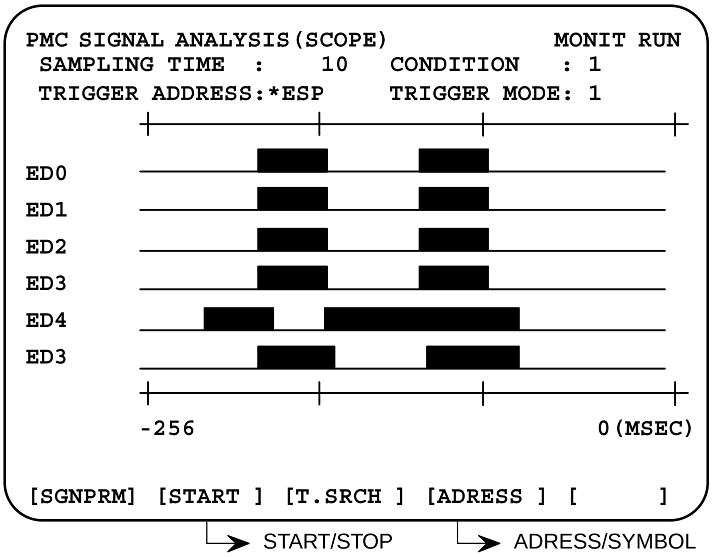


Fig. 3.6.4 Screen displaying signal diagnosis

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	○	○	○	○	○

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).

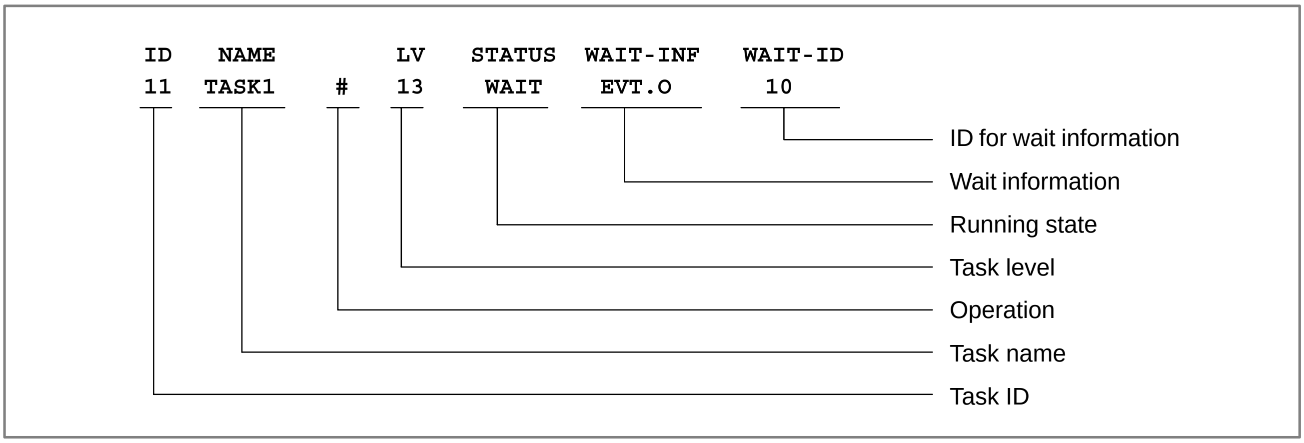
For details, refer to the “PMC-SC/SC3/SC4/NB Programming Manual (C language)” (B-61863E-1).

> **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_O1 @ 10 ACTIVE
11 TASK_O2 # 11 READY
12 TASK_O3   12 WAIT    TIM
13 TASK_O4   13 WAIT    EVT.O        1
14 TASK_O5   14 WAIT    EVT.A        3
15 TASK_O6   15 WAIT    PKT       2340
16 TASK_O7      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

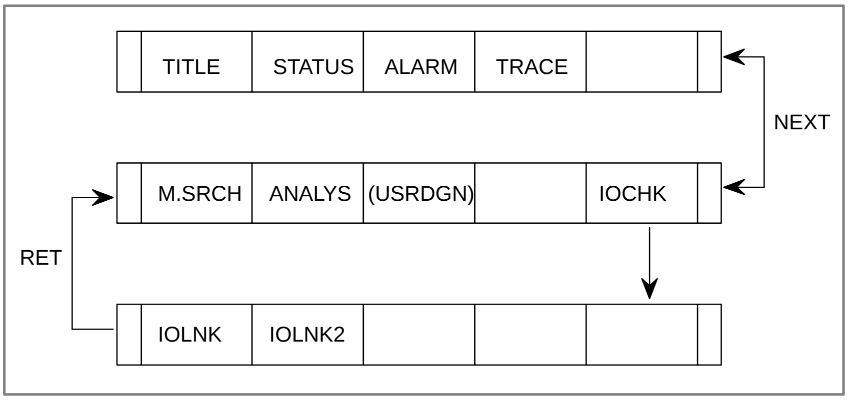
3.8 DISPLAYING AND SETTING THE CONFIGURATION STATUS OF I/O DEVICES (IOCHK)

Check kind of I/O modules connected to I/O Link.
Set communication parameters of I/O Link–II, PROFIBUS–DP, DeviceNet, FL–net.

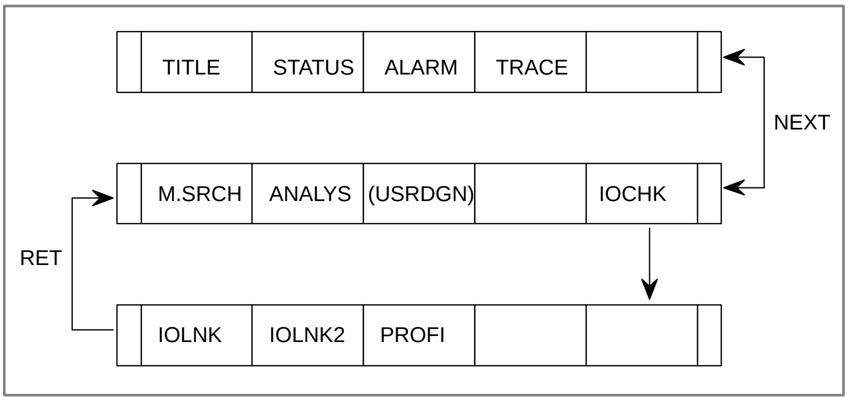
○ : Supprted
× : Not supported

Power Mate– D/F/G	Power Mate–H	Power Mate–*i*	FS21– TA/TB	FS20	FS16/ 18–A	FS16/ 18–B	FS16/ 18–C	FS21–B	FS16*i*/ 18*i*/21*i*	FS15–B
×	○	×	×	×	×	○	○	○	○	○

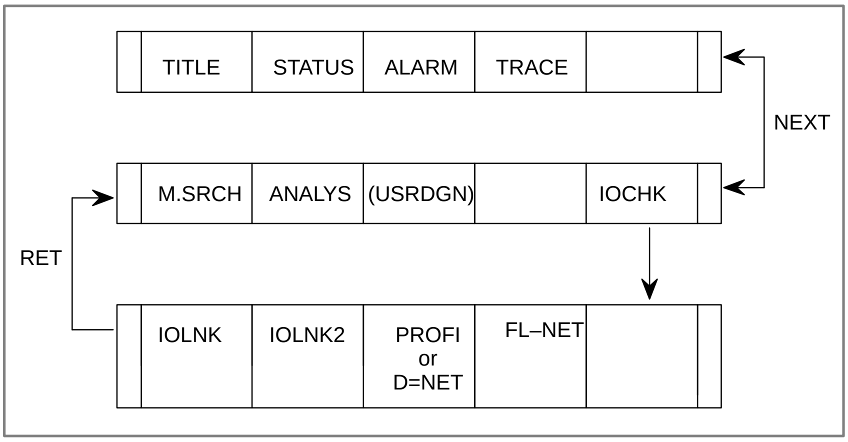
In case of FS16 / 18–B / FS21–B / Power Mate–H:



In case of FS16–C / 18–C:



In case of FS16*i* / 18*i* / 21*i*/ Power Mate *i*:



```
PMC I/O CHECK

SELECT ONE OF FOLLOWING SOFT KEYS

IOLNK   : I/O LINK CHECK
IOLNK2  : I/O LINK-II SETTING
PROFI   :  PROFIBUS PARAMETER
D_NET   :  DEVICENET PARAMETER
FL-NET  :  FL-NET SETTING
```

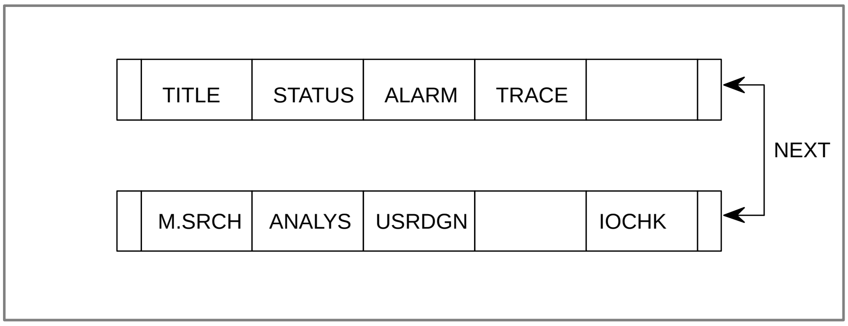
Fig. 3.8 Menu of I/O check screen

There are following sub screens under the I/O Check screen.
Softkeys are displayed when each function can be used.
Please refer to the manual about requirement of the functions and detail of the sub screens.

Softkey : Name of sub screen
[IOLNK] : I/O Link Connecting Check screen

[IOLNK2]: I/O Link–II Parameter Setting screen
.....”FANUC I/O Link–II CONNECTING MANUAL (B–62714EN)”
[PROFI] : PROFIBUS–DP Parameter Setting screen
.....”FANUC PROFIBUS–DP Board OPERATOR’S MANUAL (B–62924EN)”
[D_NET] : DeviceNet Parameter Setting screen
.....”FANUC DeviceNet Board OPERATOR’S MANUAL (B–63404EN)”
[FL–NET]: FL–net Parameter Setting screen
.....”FANUC FL–net Board OPERATOR’S MANUAL (B–63434EN)”

In case of FS15–B:



Only I/O Link Connecting Check screen is supported.
When [IOCHK]key is pressed, I/O Link Connecting Check screen isdisplayed 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.
If there is a problem of input or output signals for I/O devices, check connection of I/O Link by referring to this screen.

(1) When CNC hardware dose not support the I/O Link expansion function.

```
PMC I/O LINK(1/1)
             CHANNEL 1
          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 FANUC SERVO AMPLIFIER UNIT ß series (I/O Link OPTION)
CONNECTION UNIT	96	I/O Link connection unit
I/O MODULE	A9 to AA	Distributed I/O
OTHER UNIT	—	Other than above

The screen displayed like fig.3.8.1 (a) means that the I/O devices are composed like following fig.3.8.1 (b).

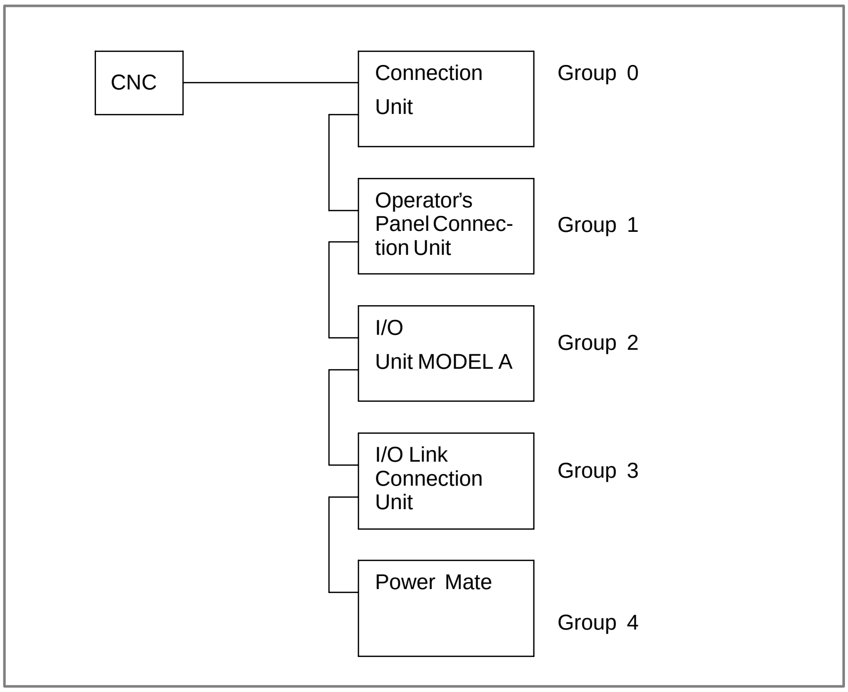


Fig. 3.8.1 (b) I/O Link configuration

(2) When CNC hardware supports the I/O Link expansion function.

```
PMC I/O LINK(1/1)
   CHANNEL 1                        CHANNEL 2
GROUP ID KIND OF UNIT            GROUP ID KIND OF UNIT

 00   80 CONNECTION UNIT          00   80 I/O UNIT
                                          MODEL A
 01   82 OPERATOR PANEL           01   82 POWER MATE
 02   84 I/O UNIT MODEL A         02   84 POWER MATE
 03   96 CONNECTION UNIT
 04   4A POWER MATE
```

Fig. 3.8.1 (c) Example of the I/O Link connecting check screen

When the screen is displayed like fig.3.8.1(c), the I/O devices are composed like following fig.3.8.1 (d)

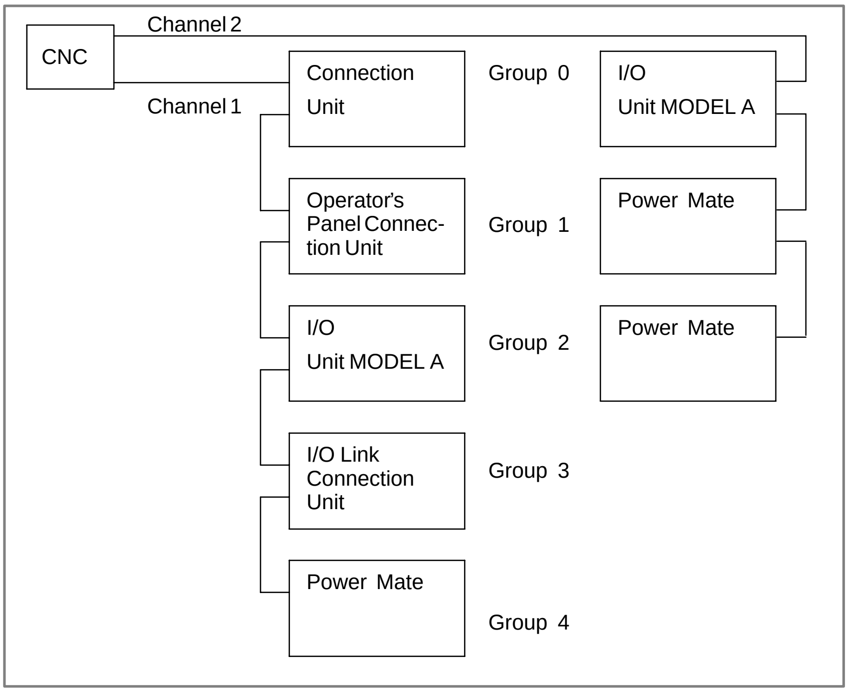


Fig. 3.8.1 (d) I/O Link configuration

When PMC model is not PMC–SB6 for Series 16*i* /160*i* /18*i* /180*i* /21*i* / 210*i* or optional I/O Link expansion is not provided with CNC, "NOT SUPPORTED" is displayed on CHANNEL 2.

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 Operator's 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	Power Mate *i*–D/H	FS20/ FS21A	FS18A	FS16A	FS16B FS18B		FS16C FS18C		FS21*i*	FS16*i* FS18*i*		FS15B
PA1 PA3	SB5 SB6	SA1 SA3	SA1, SA2 SA3	SB, SB2, SB3 SC, SC3	SB3 SC3	SB4 SC4	SB5 SC3	SB6 SC4	SA5	SB5	SB6	NB NB2
×	△	×	×	×	×	×	×	×	△	△	△	×

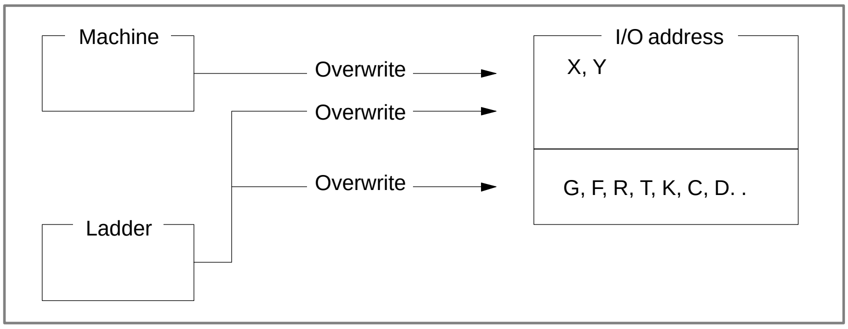
NOTE
With the FS16*i*/18*i*/21*i*, the edit card or C board are required.
With the Power Mate *i*–D/H, the edit card is required.
With SA1 of the FS16*i*/18*i*/21*i*, only FORCING mode is valid.

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.

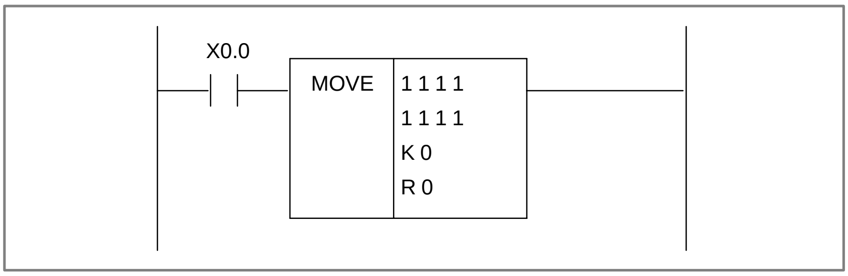
(1) Input mode
Two input modes are available. The user can choose between the two modes, depending on the application.

(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.



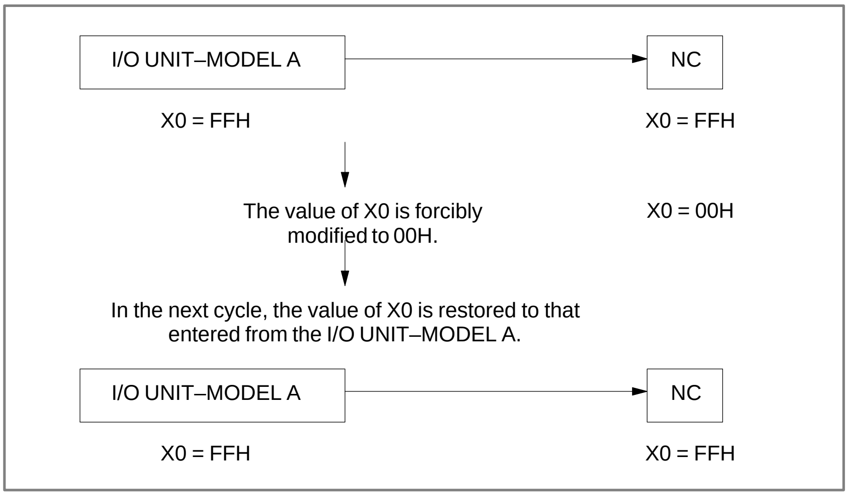
a. The initial signal states are as follows:
X0.0 = off, K0 = 55H, R0 = 00H

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

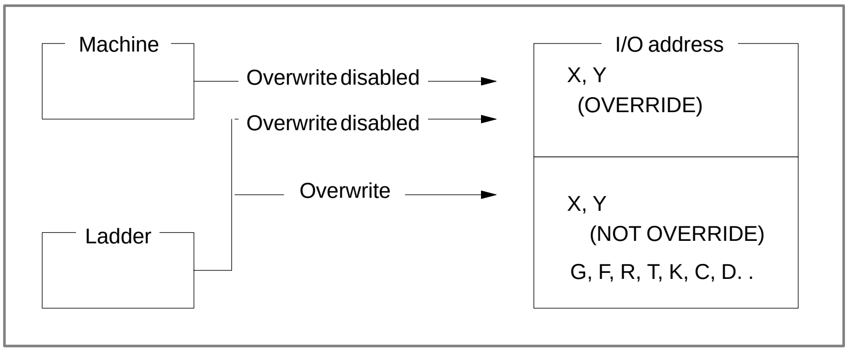
Example 2:

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.



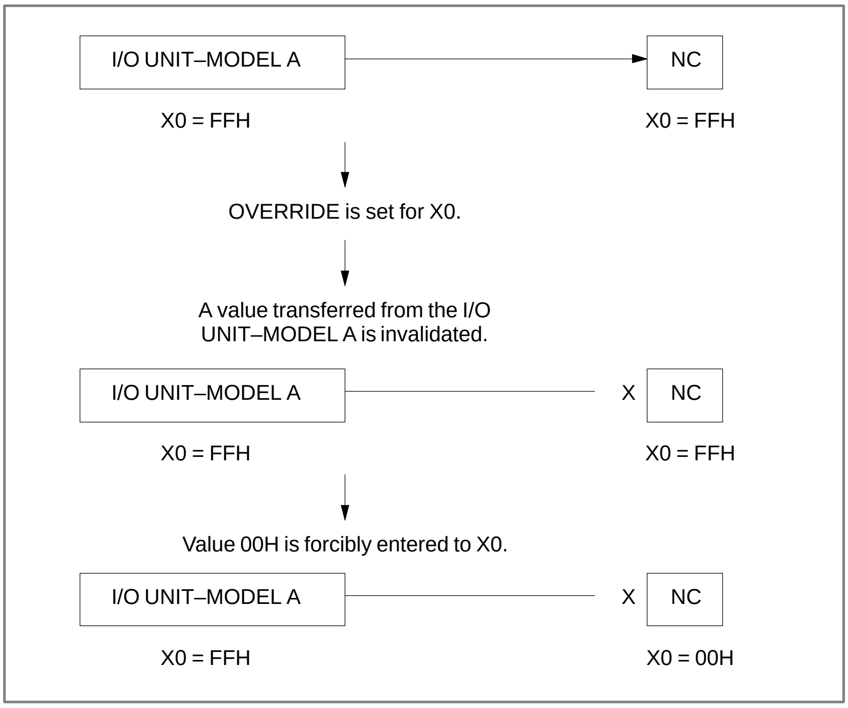
Cyclical transfer may also be performed for addresses that are not assigned. So, use the forced input/output function for X in FORCING mode to debug a sequence program when no machine is connected or assigned. Use the OVERRIDE mode to debug a sequence program in the case of I/O connection.

(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.

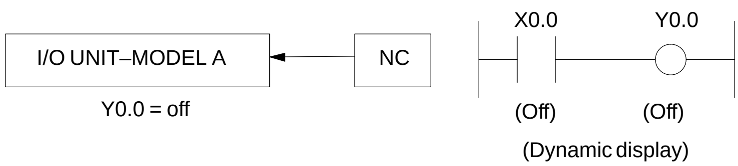
2 Note that, when OVERRIDE mode is set, the interval of the second level may be extended slightly.

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.

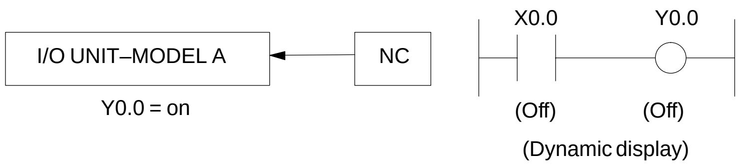
A value, after being modified by the forced input/output function, is output to the machine. So, note that the on/off indication in ladder dynamic display does not match a value output to the machine.

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

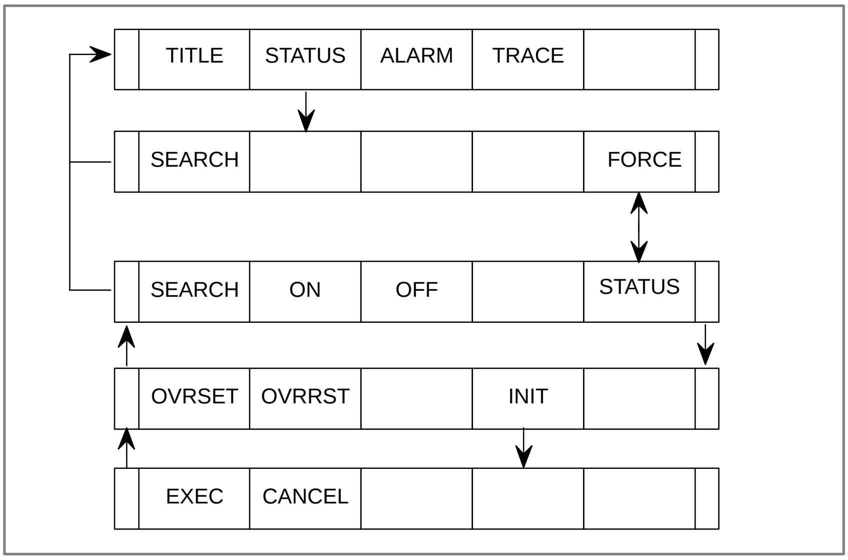
Special care must be exercised when modifying a signal with the forced input/output function. If the forced input/output function is used incorrectly, the operation of the machine may be unpredictable. Never use this function when persons are near the machine.

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.

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.)

```
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 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
Reserved	K19	K19

Model	SA1	SA2
PMC control software data 1	K17	K17
PMC control software data 2	K18	K18
Reserved	K19	K19

Model	SB	SB2
PMC control software data 1	K17	K17
PMC control software data 2	K18	K18
Reserved	K19	K19

Model	SA3/ SA5	SB3/ SB5	SB4/ SB6
PMC control software data 1	K17	K17	K900
PMC control software data 2	K18	K18	K901
PMC control software data 3	K19	K19	K902
Reserved			K903 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
Reserved			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
Reserved		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, and Power Mate *i*–D/H 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.)

WARNING
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. Set to 0.)

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.)

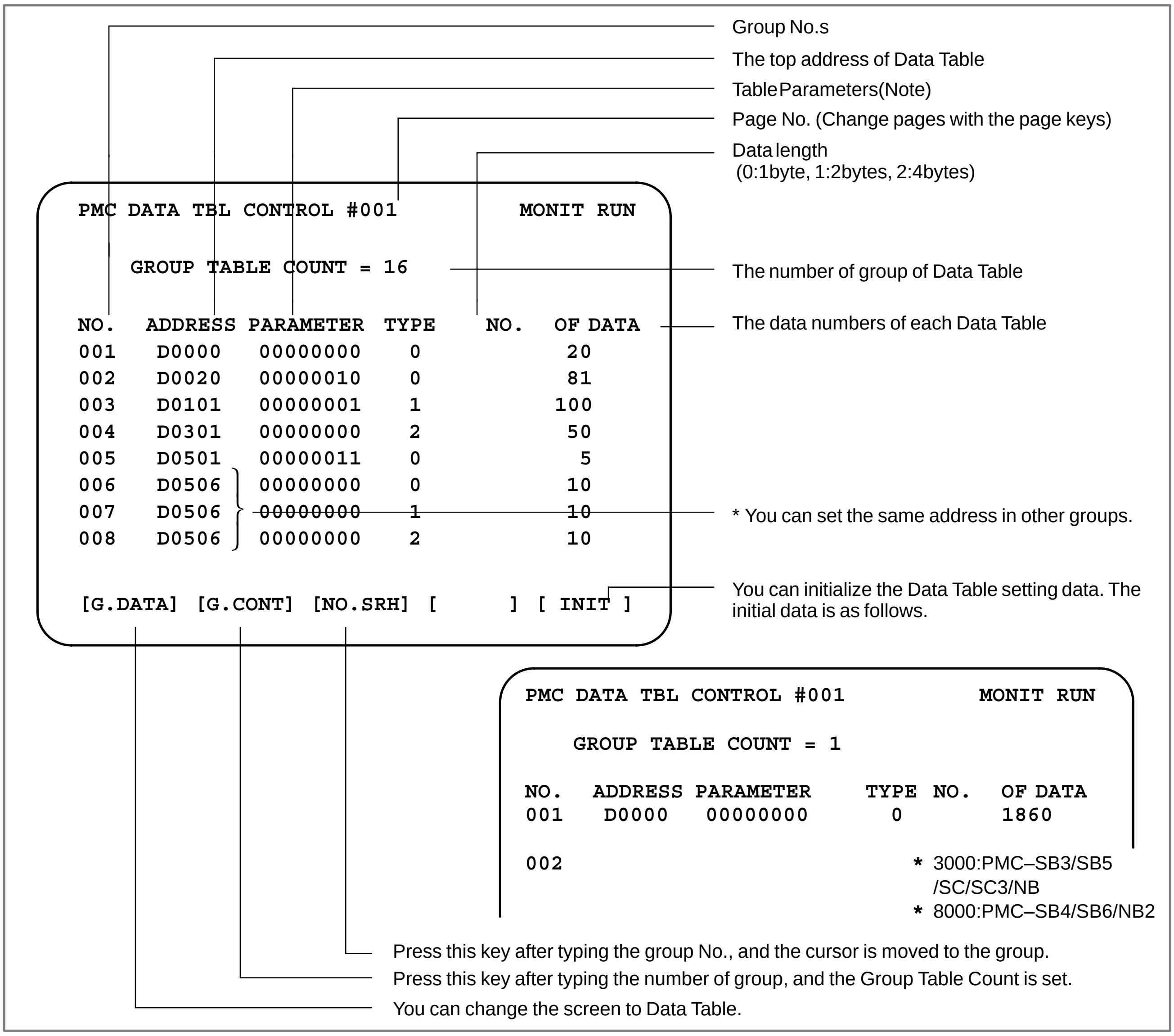
> **CAUTION**
> The unused area of the data for the PMC management software must always be set to 0.

4.3.4 Data Table (DATA)

DATA TABLE consists of two screens, that is, Data Table Controlling Data screen and Data Table screen.

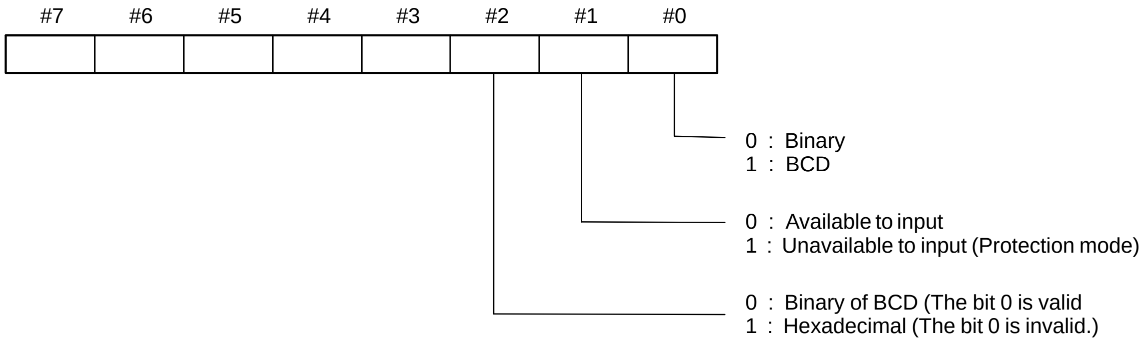
(1) Data Table Controlling Data Screen

Data Table Controlling Data Screen for controlling Data Table is displayed by pressing the soft key [DATA].



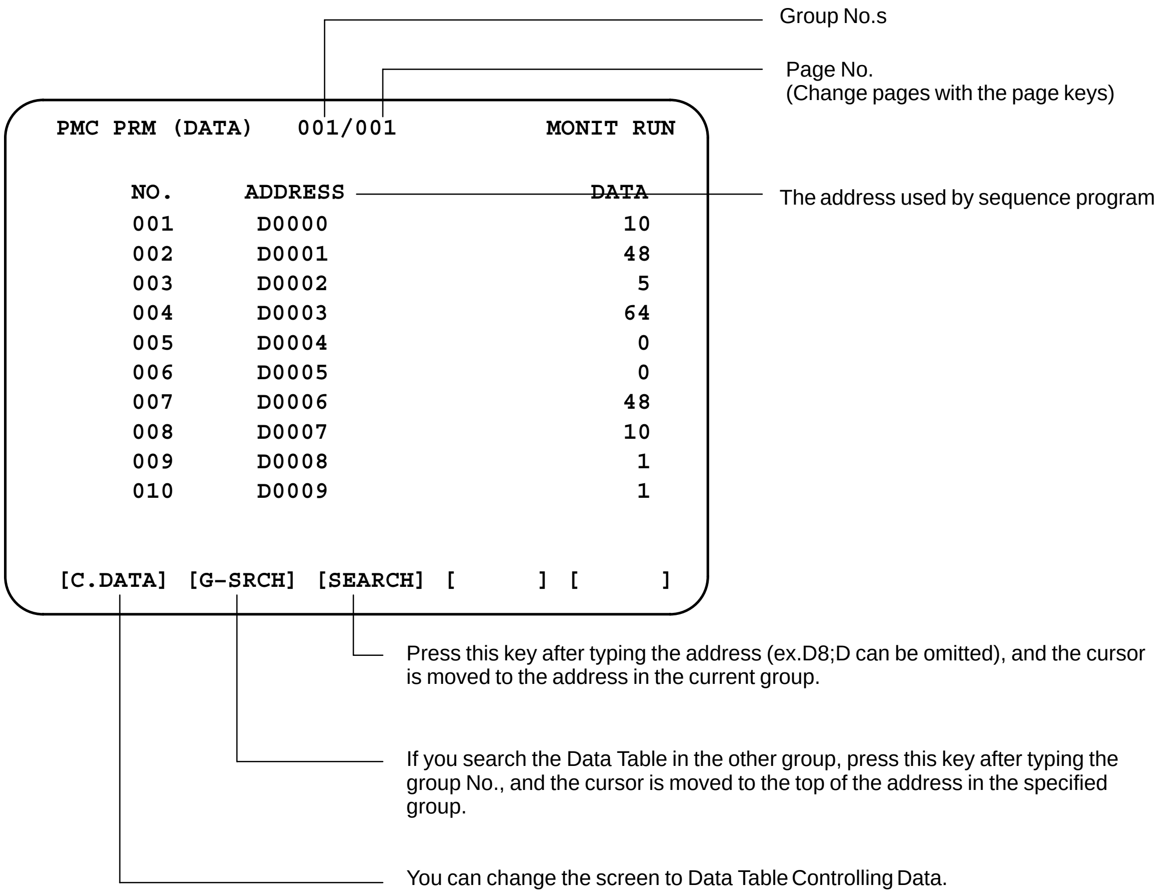
NOTE

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–SC/SC3/SC4 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–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–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.)

GRAY SCALE DISPLAY

This parameter sets the ladder–related display method when a monochrome LCD is used. When you find the screen display on the monochrome LCD not easy to view, change this parameter setting.

When USE is selected, the difference in brightness is used for screen display.

When UNUSED is selected, reverse video is used for screen display.

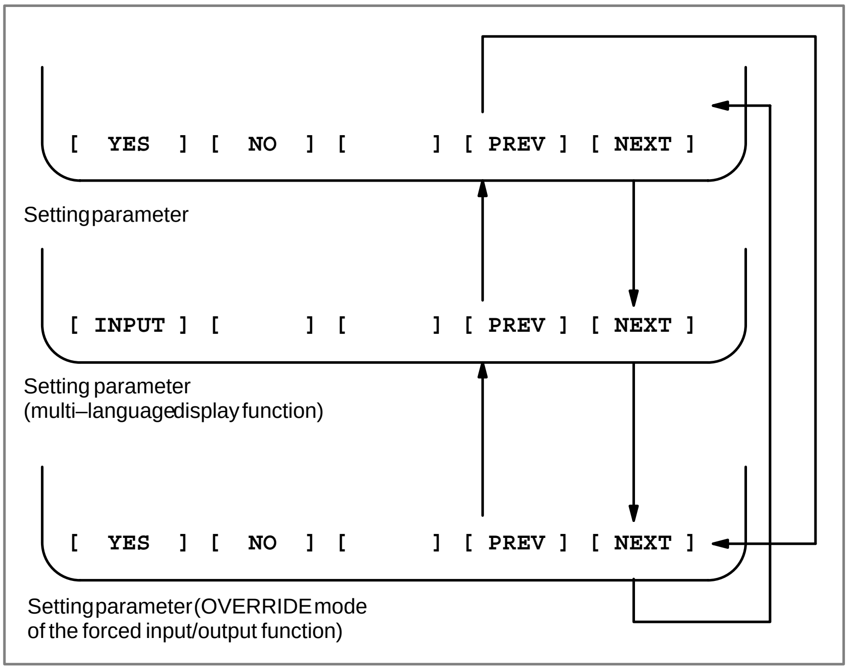
> **NOTE**
> A change made to this parameter does not become valid until the power is turned on next. After the setting of this parameter has been changed, the power must be turned off then back on for the setting to become effective.

4.4.1 Other Setting Screens

Enable or disable the following function and mode:

- Multi–language display 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) Multi–language display 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 Multi–language display 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.

NOTE
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).

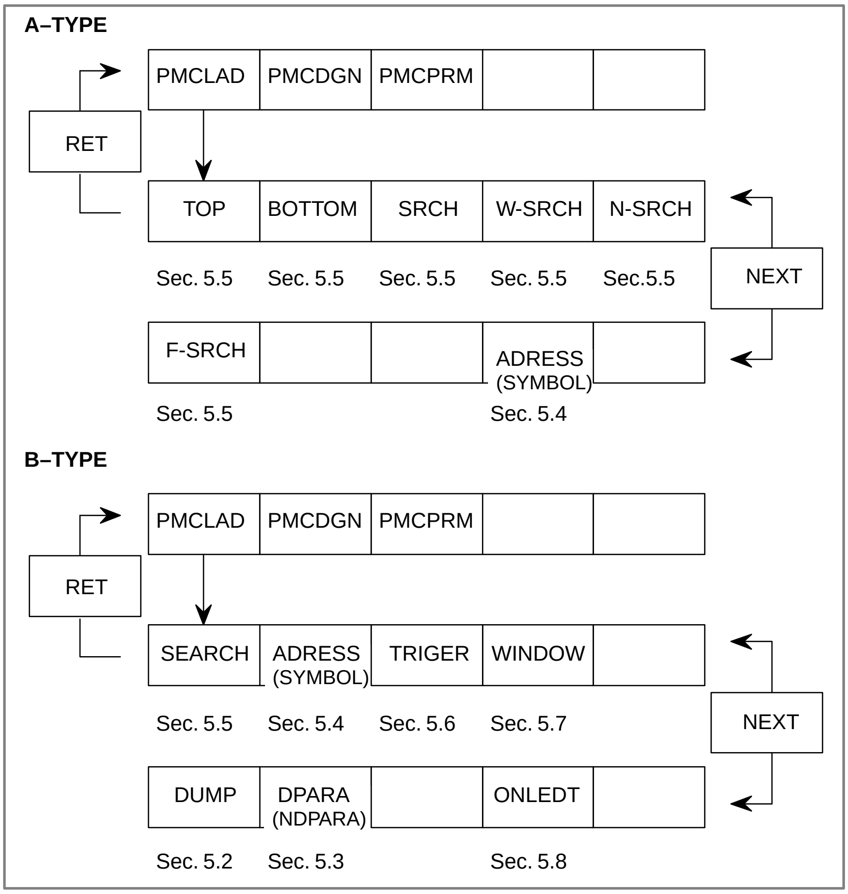
(4) Screen-dividing display.

(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.

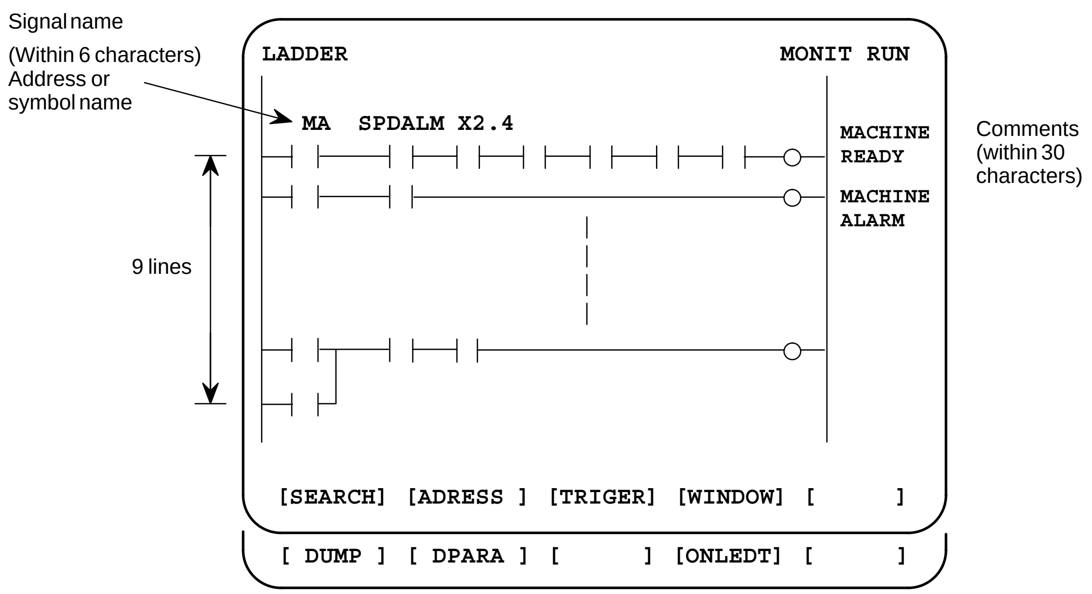


Fig. 5.1 Ladder diagram display

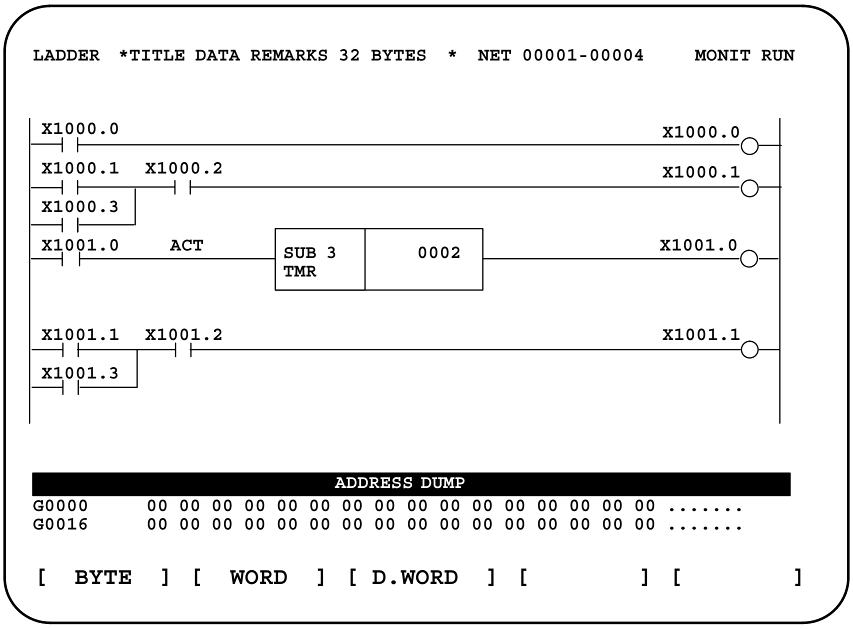
NOTE
When online communication is being performed with USE selected on the online setting screen (see Section III–8.5), pressing the [PMCLAD] soft key displays the message ”REJECT (ONLINE MONITOR ACTIVE),” disabling access to the display. In such a case, select NOT USED on the online setting screen, and stop online communication on the PC side.

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”

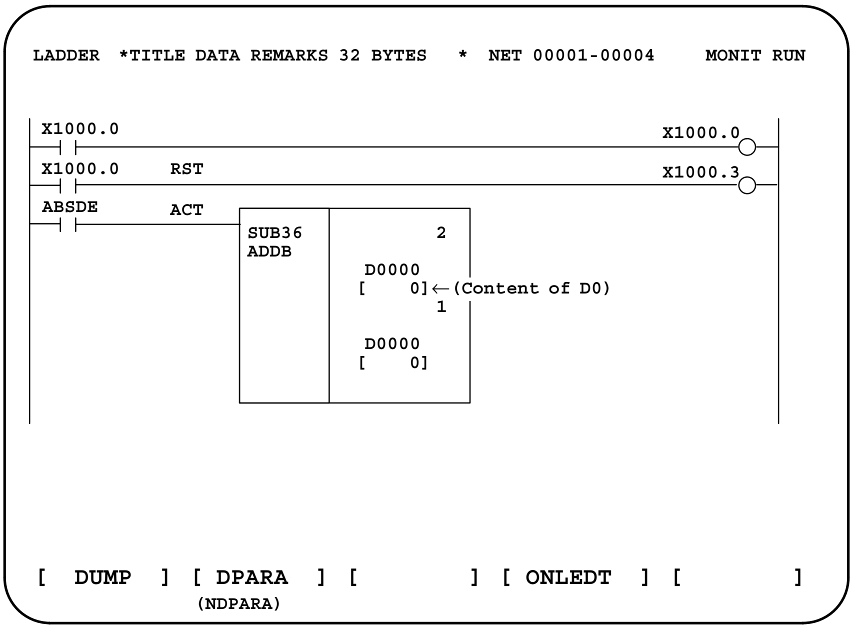
(3) [D.WORD] : Long word type display (4 BYTE)

“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

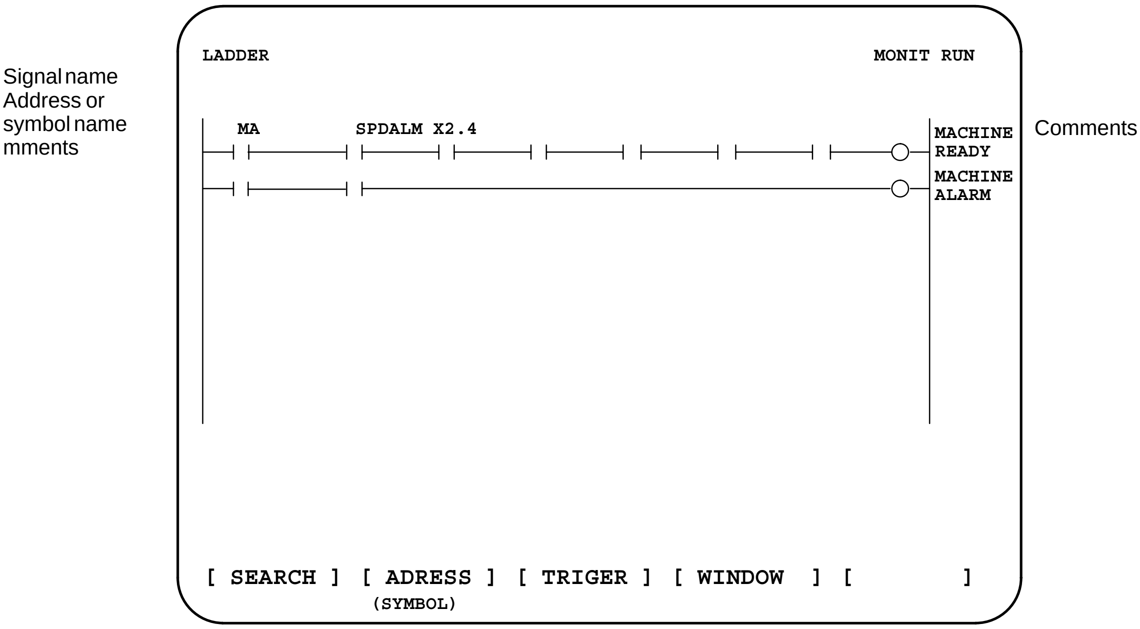
NOTE

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)



The function of the soft key [ADRESS] is as follows.

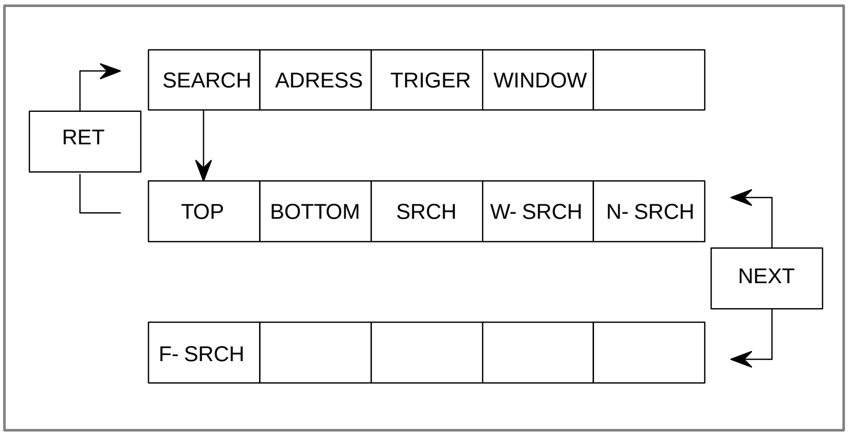
(1) [ADRESS] : is used to display the address name.

(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 [SEARCH] 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.

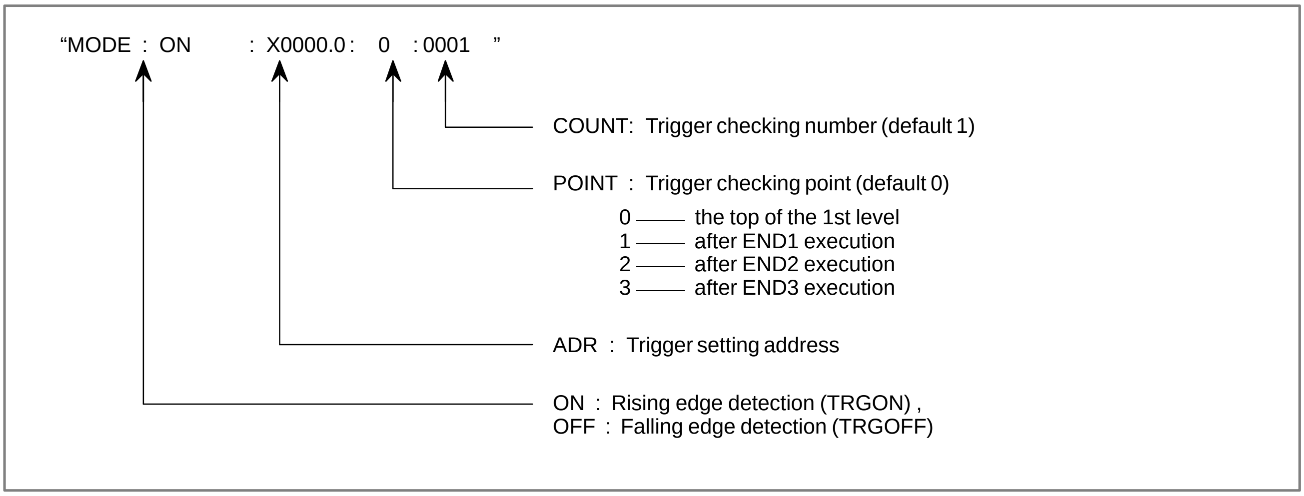
The ladder display can be stopped by manual operation or trigger of signal.

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.



* Setting form adr ; p1 ; p2 + [TRGON/TRGOFF] soft key

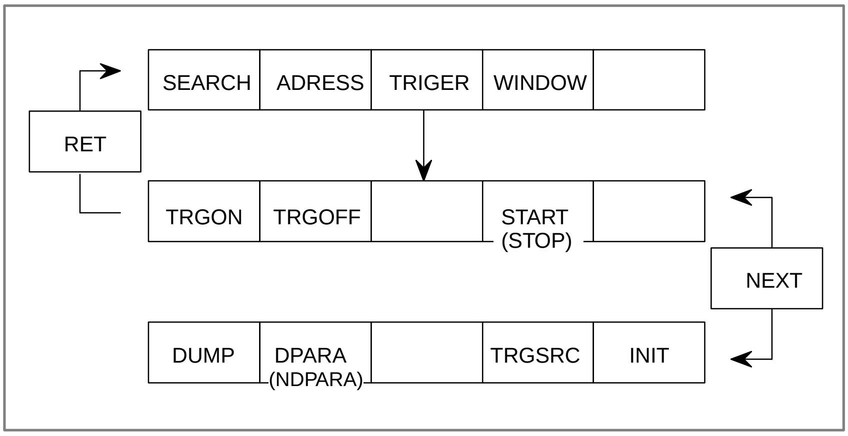
NOTE
" ; " = "EOB"

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.

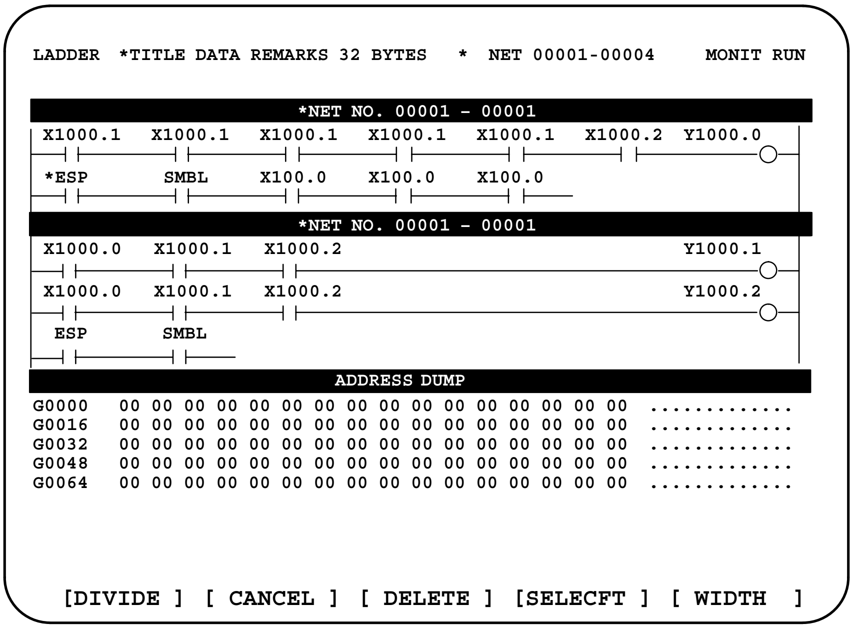
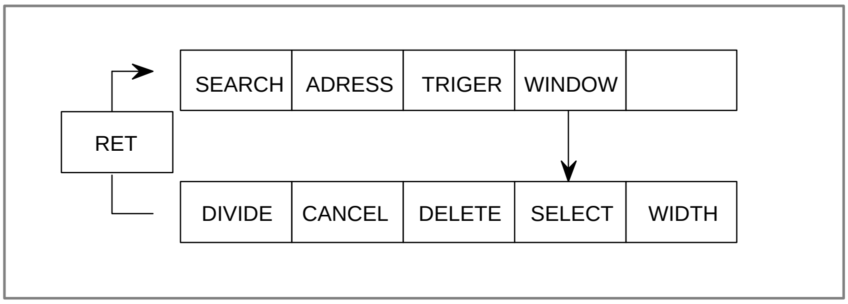


Fig. 5.7 Dividing display of ladder diagram

> **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).

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 Part IV, “PMC–NB6 Manupiration Screen” for an explanation of the FS15*i* PMC–NB6.

1.2 ABOUT THE FS16*i*/18*i*/21*i*–B PMC–SA1/SB7 PROGRAMMER

See Part V, “PMC–SA1/SB7 Manupiration Screen” for an explanation of the FS16*i*/18*i*/21*i* PMC–SA1/SB7.

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

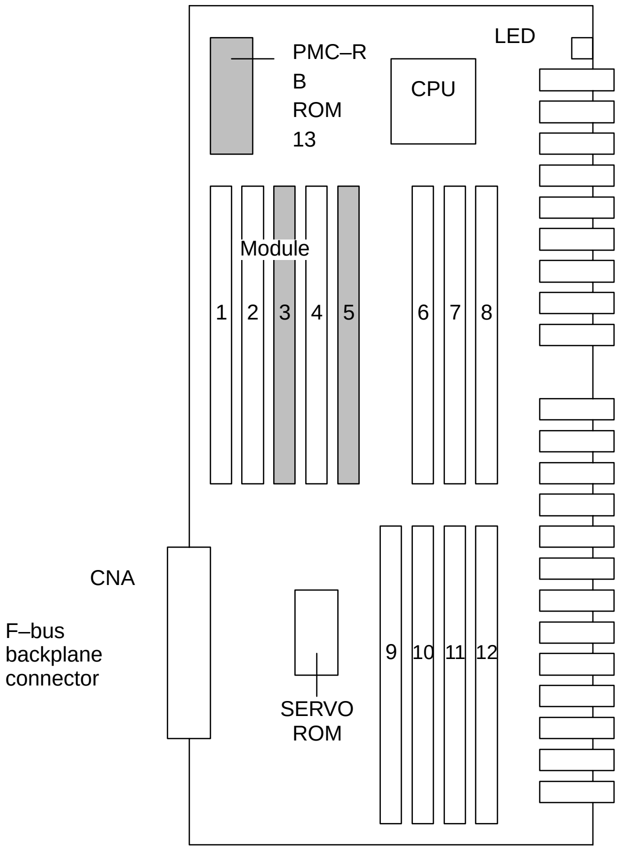
      i) PMC–SA1/SA2 (Main CPU board)

      - 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)

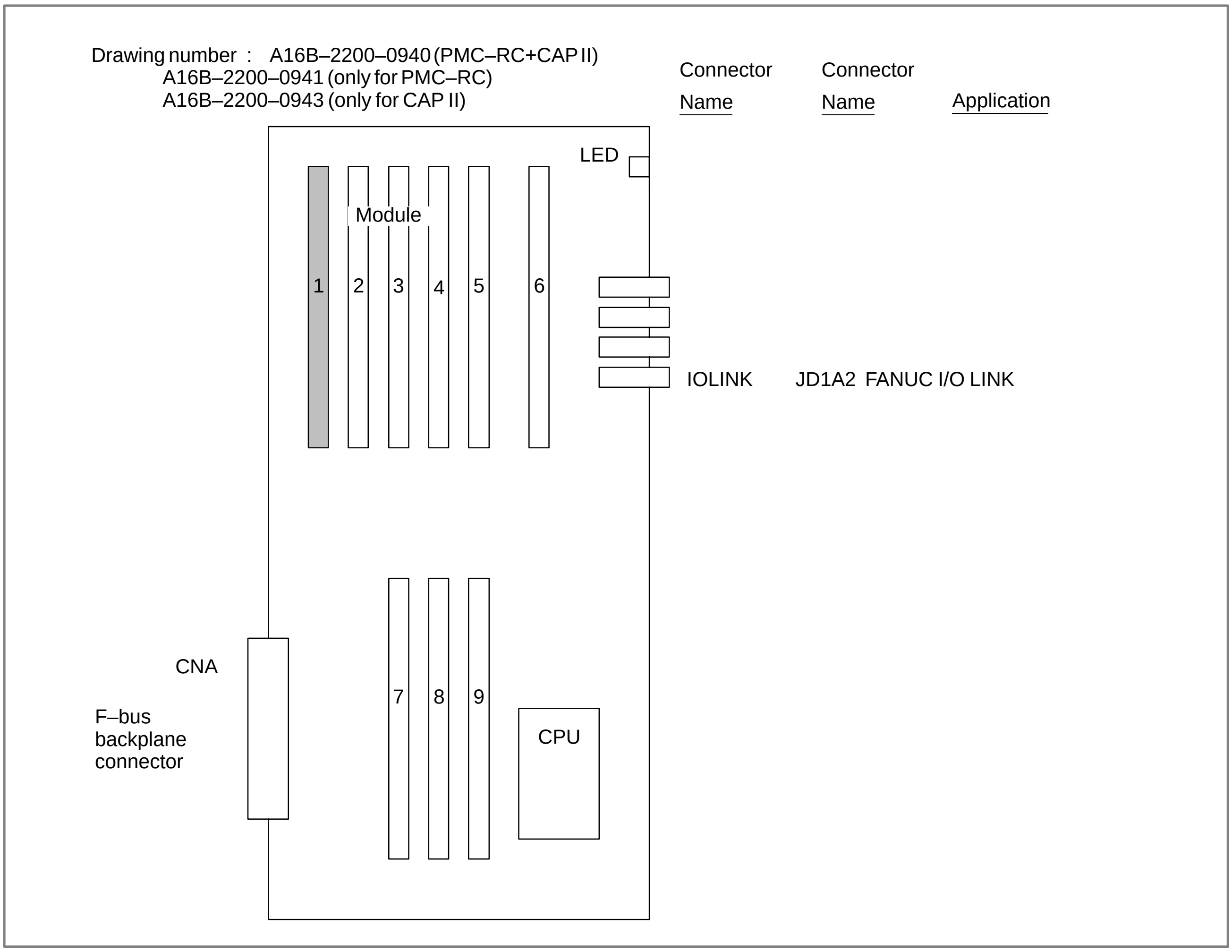


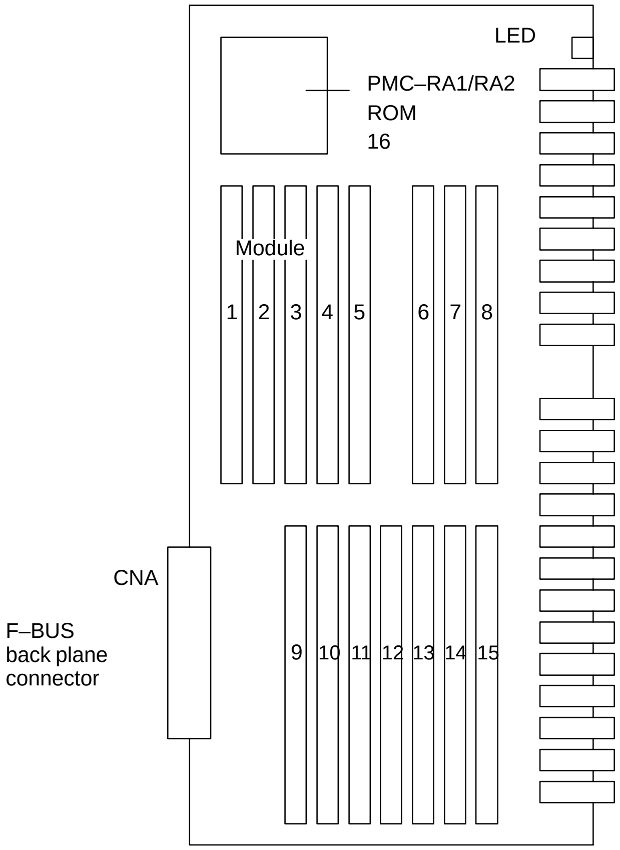
Fig. 2.1 (b) Layout of Parts on Option 3 Board (Series 16)

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 24000 steps or C language
		A20B–2900–0391	Other than the above

Configuration of the Main CPU Board (Series 18)

Drawing number : A16B–2201–0080



Connector Name	Connector No.	Description
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	Battery for use with the APC
AMP1	JV1	Axis 1 servo amplifier
AMP2	JV2	Axis 2 servo amplifier
AMP3	JV3	Axis 3 servo amplifier
AMP4	JV4	Axis 4 servo amplifier
ENC1	JF1	Axis 1 pulse coder
ENC2	JF2	Axis 2 pulse coder
ENC3	JF3	Axis 3 pulse coder
ENC4	JF4	Axis 4 pulse coder
SCALE1	JF21	Axis 1 scale
SCALE2	JF22	Axis 2 scale
SCALE3	JF23	Axis 3 scale
SCALE4	JF24	Axis 4 scale

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

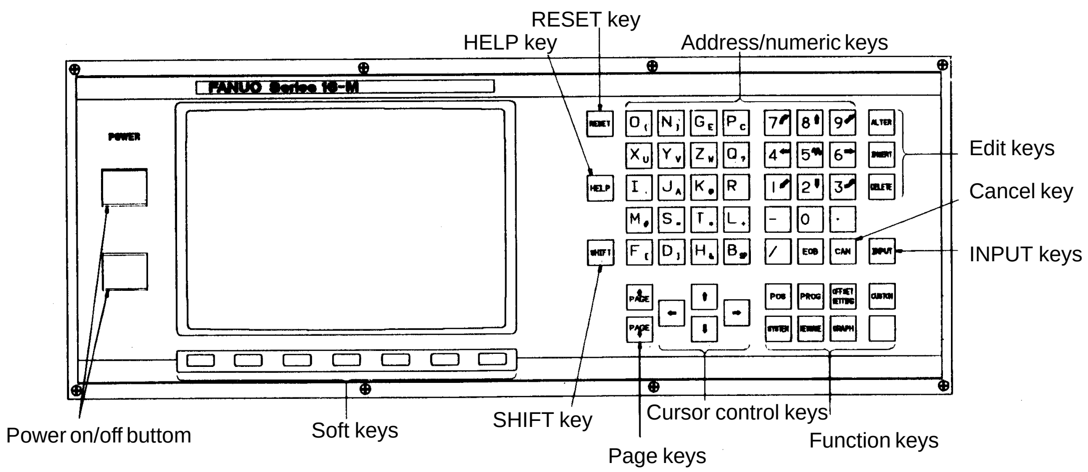
To operate the PMC programmer, set bit 1 in K17 of the keep relay area for PMC parameters to 1, enabling the programmer basic menu to be displayed. To display the programmer basic menu, press <SYSTEM> and [PMC] soft key on the MDI keyboard then, press the [NEXT] key.

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

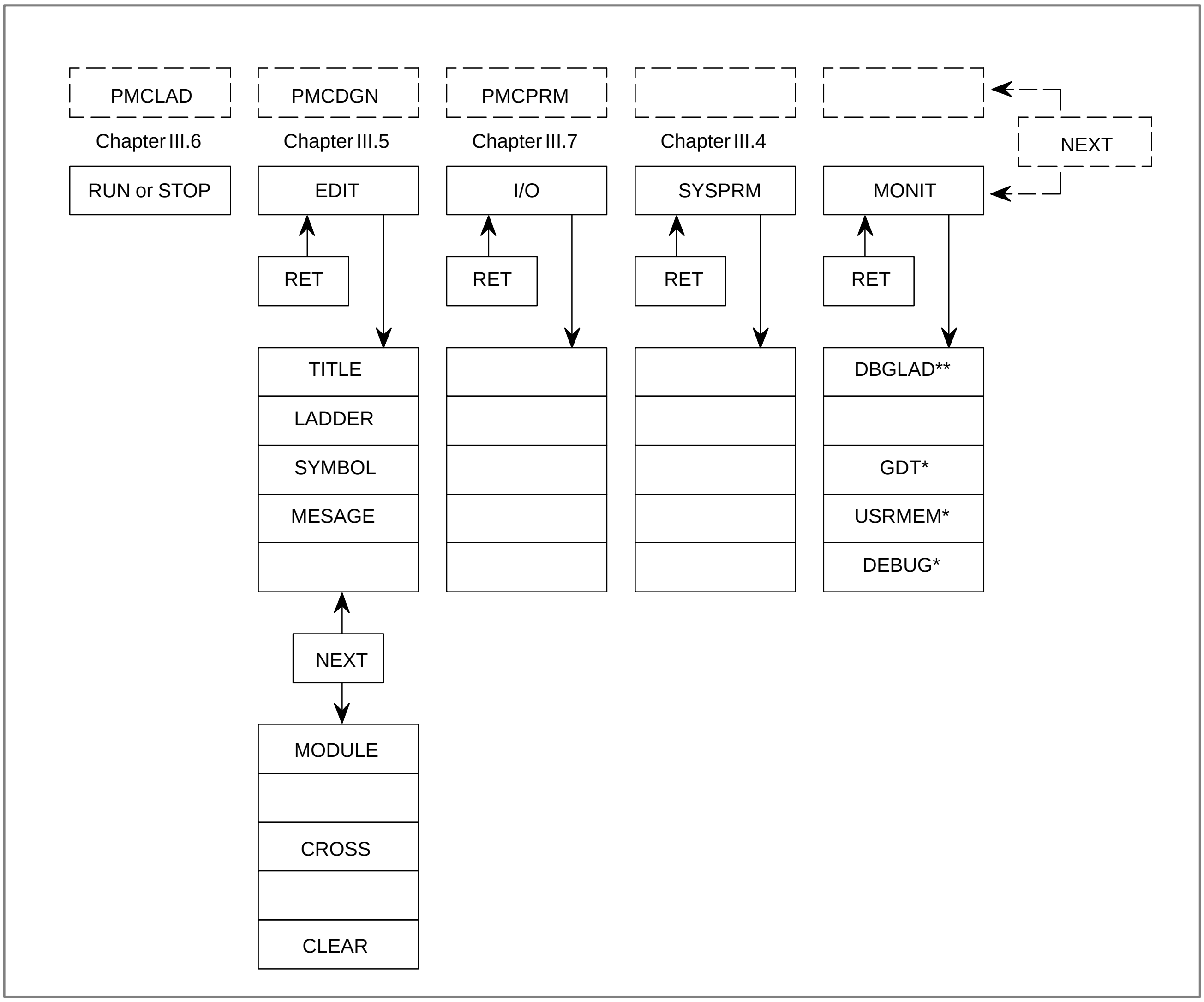
The programmer basic menu and PMC basic menu are selected to each other alternately by pressing the [NEXT] key. For the PMC basic menu and operation, see PMC operation in Chapter II.

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.

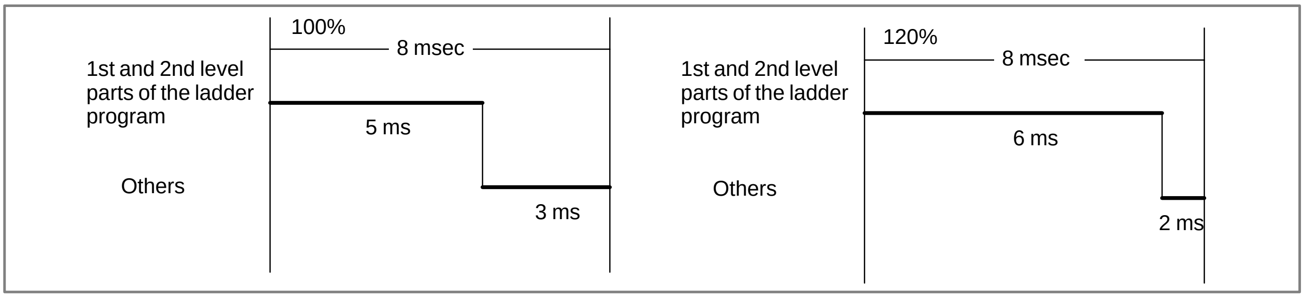
The processing time of the 1st and 2nd parts of the ladder program is obtained by the following formula:

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, X200 to X327, 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, Y200 to Y327, 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 (R) 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 (R) 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

(9) I/O Link 2 CHANNEL (PMC–SB6 for Series 16*i*/18*i*/21*i*/160*i*/180*i*/210*i*)

When creating a new program for assignment of the I/O Link channel 2 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 assigning the I/O Link channel 2
(Note1): I/O LINK 2 CHANNEL = YES

When assigning only the I/O Link channel 1
(Note2): I/O LINK 2 CHANNEL = NO

NOTE

1. PMC model for FAPT LADDER–II /Ladder Editing Package is PMC–SB6 (IO–2) or PMC–SB6 (STEP SEQ, IO–2)(case of using the step sequence method).
2. PMC model of FAPT LADDER–II /Ladder Editing Package is PMC–SB6 or PMC–SB6 (STEP SEQ)(case of using the step sequence method).
3. When assigning the I/O Link channel 2, capacity of the memory used by system increases maximum 5376 bytes compared to the case of not assigning the I/O Link channel 2. Please adjust the volume of Ladder, symbol, comment, and message, in order to reduce the sequence program size to fit for the ROM capacity.
4. FAPT LADDER–II /Ladder Editing Package is used when converting sequence program for 1 channel (PMC–SB6 or PMC–SB6 (STEP SEQ)) into sequence program for expansion (PMC–SB6 (IO–2) or PMC–SB6 (STEP SEQ, IO–2)). Please refer to ”FAPT LADDER–II OPERATOR'S MANUAL (B–66184EN/02) APPENDIX 3.3 Convert The PMC Type of Sequence Program” and ”FANUC OPEN CNC LADDER EDITING PACKAGE OPERATOR'S MANUAL (B–62884EN/01–6) APPENDIX D.3 CONVERT THE PMC TYPE OF SEQUENCE PROGRAM”.

```
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)

```
PMC SYSTEM PARAMETER (1/2)         MONIT STOP

 COUNTER DATA TYPE     =  BINARY/BCD

 STEP SEQUENCE         =  YES/NO

 I/O LINK 2 CHANNEL    =  YES/NO




>

[ YES  ] [  NO   ] [      ] [      ] [      ]
```

Fig. 4(g) PMC–SB6 system parameter screen (1st page)

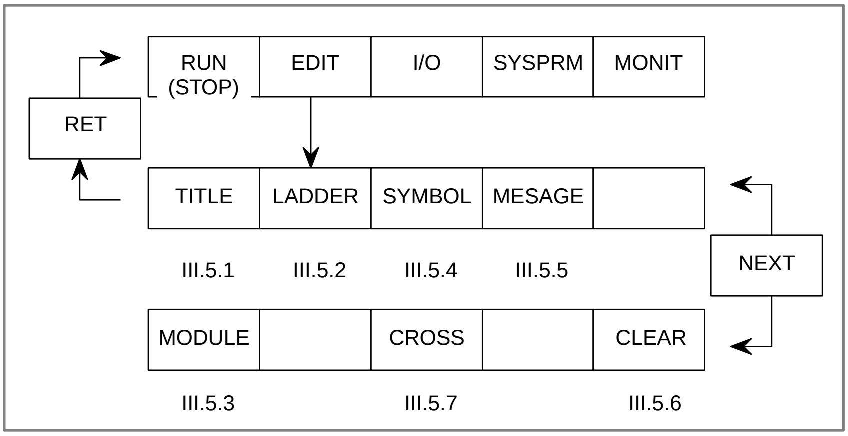
5 EDITING OF SEQUENCE PROGRAM (EDIT)

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 ]
```

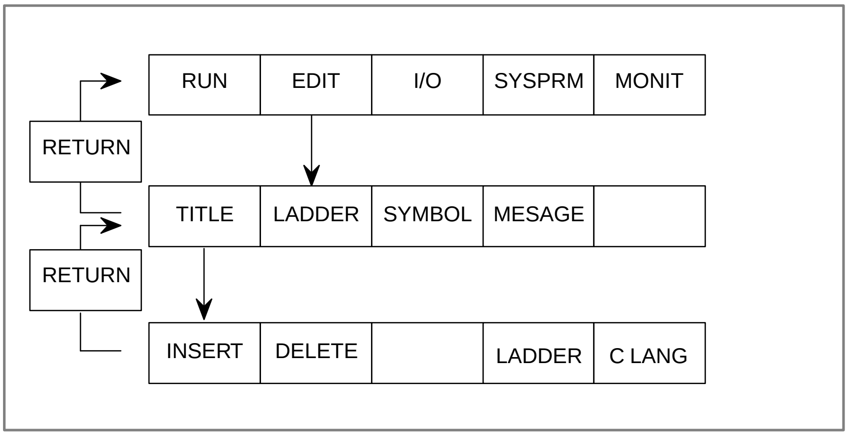
Fig. 5 Editing basic menu

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.

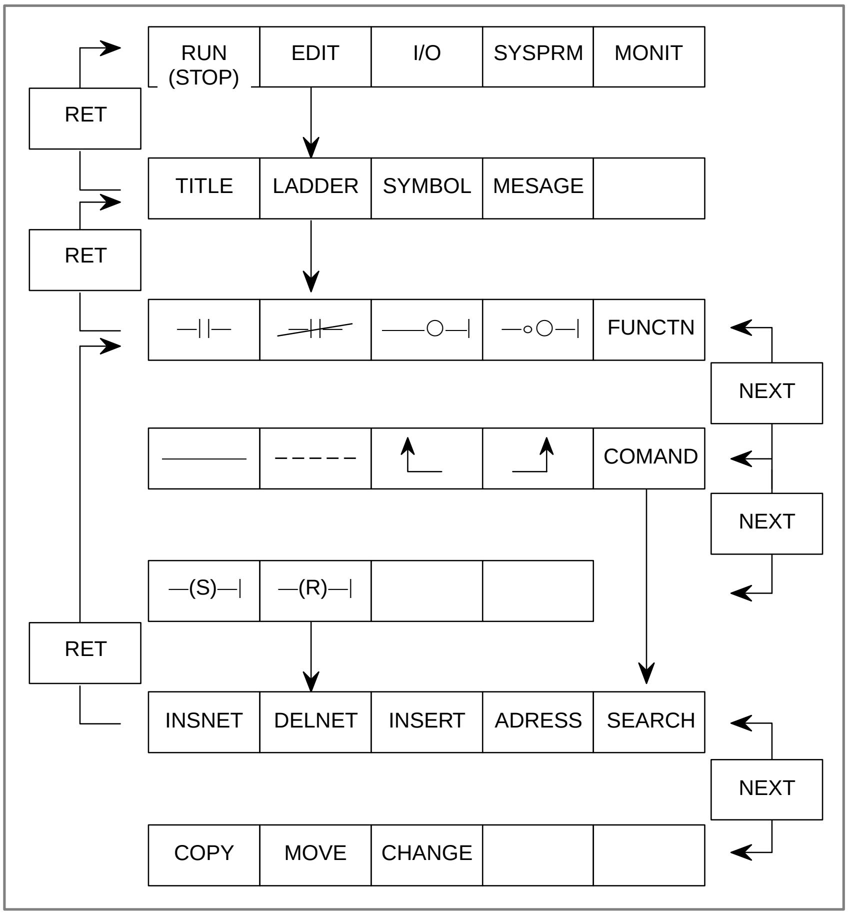


Fig. 5.2 Sequence program generation softkeys

> **NOTE**
> "—(S)—|"and "—(R)—|" are valid for PMC–PA3, –SA3, –SB3, –SB4, –SC3, –SC4, and –NB.

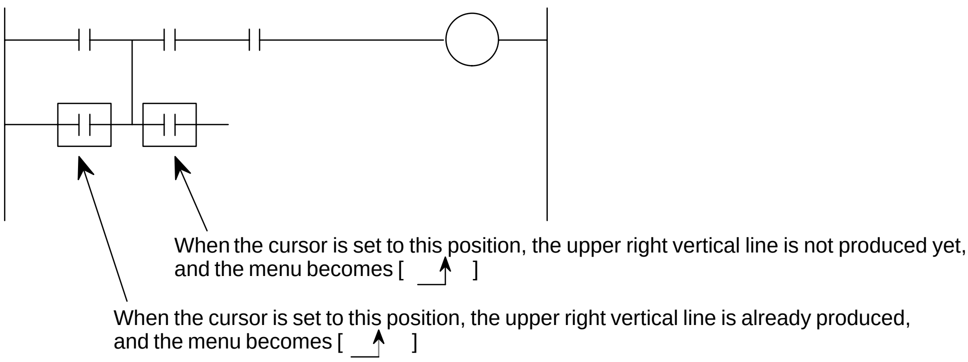
Each of EDIT · LADDER software functional instruction keys can be selected by the [COMAND] key. Type in one of the following character strings and press software key [COMAND]. The character string within parentheses "[ ]" can be omitted. "n" after the character string indicates that a value can be input. 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.

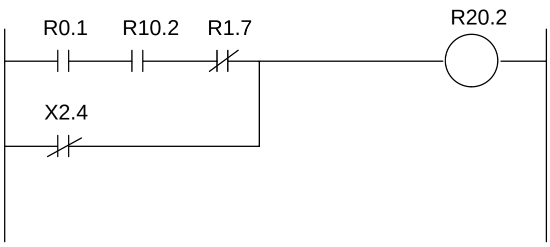
If a sequence program is not input yet, the right and left vertical lines only of the ladder diagram are displayed on CRT/MDI.

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.

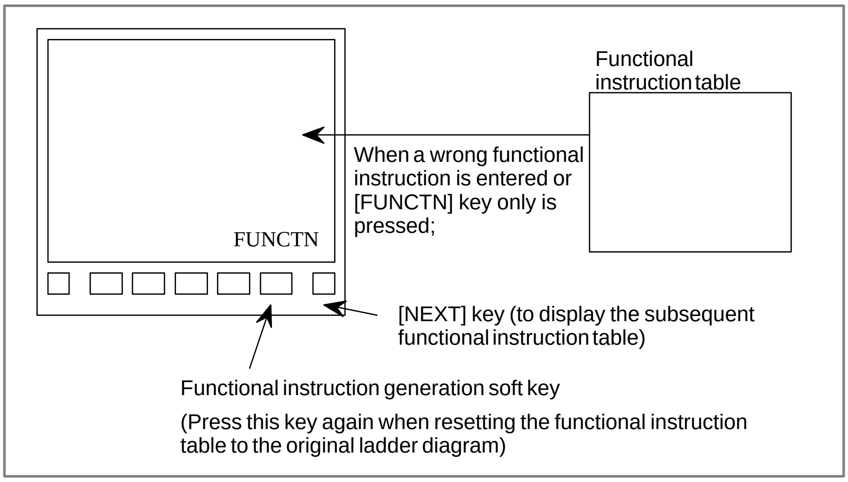
5 In the Series 15–MODEL B, Series 16/18–MODEL B/C, and Series 16*i*/18*i*/21*i*–MODEL A CNC that use Flash Memory, the program is not automatically written into Flash Memory once editing ends. After editing, perform the processing for writing to Flash Memory (see 7.3.3, “FROM” in Chapter 7 of Part III). 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;

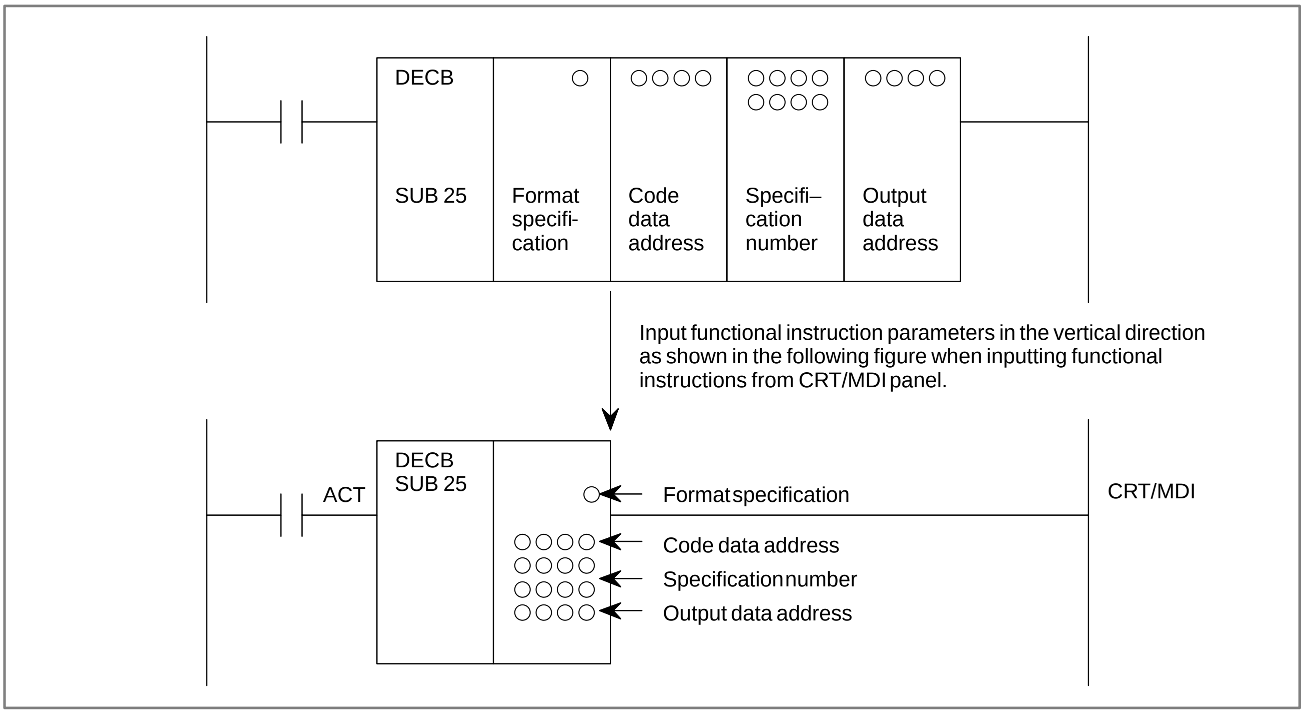
For inputting a functional instruction, press [FUNCTN] soft key, and then, input instruction symbol of the functional Instruction and SUB number.
A function command can be input by pressing the [FUNCTN] key after keying the Function Command No. When pressing the [FUNCTN] key without keying in the Function 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.

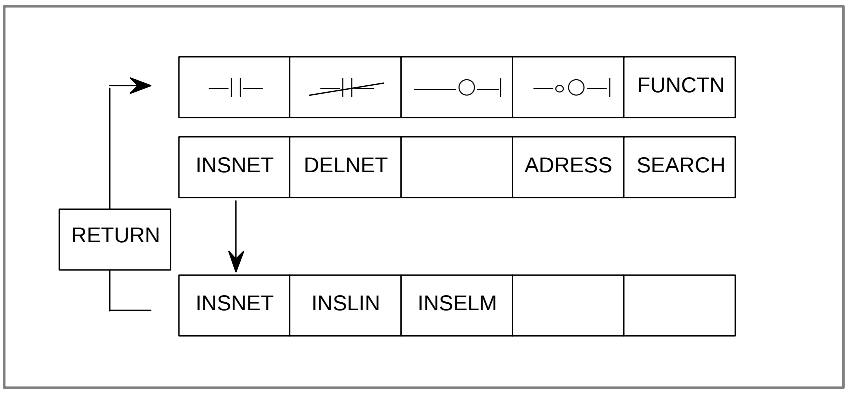
3 Input an instruction parameter.

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

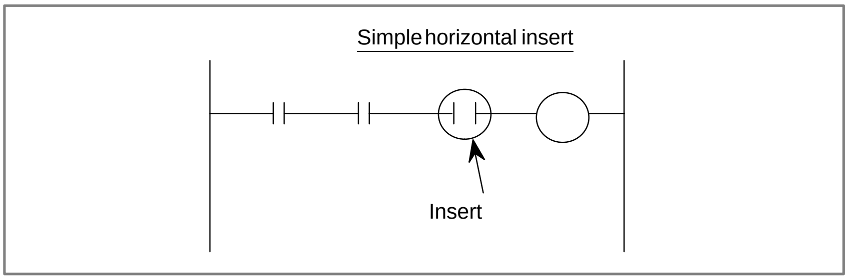
The method of altering a generated sequence program is the same as described in 5.2.1. Move the cursor to the program part to be altered and input change data.

5.2.3 Insert of Sequence Program

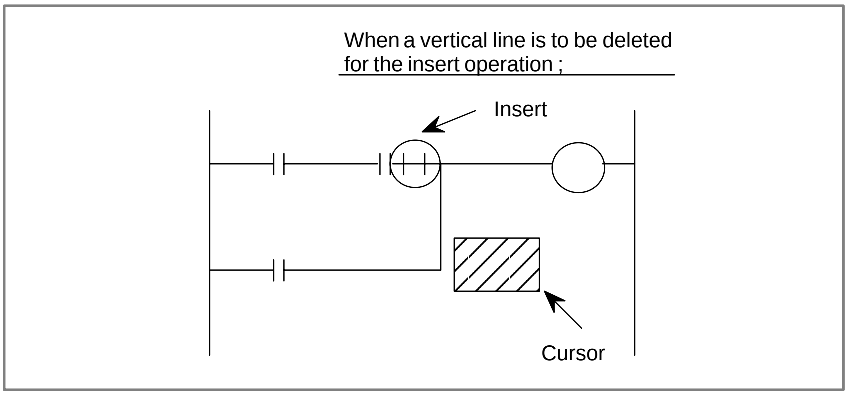


A sequence program is inserted in four ways on the ladder diagram as described below.

(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.

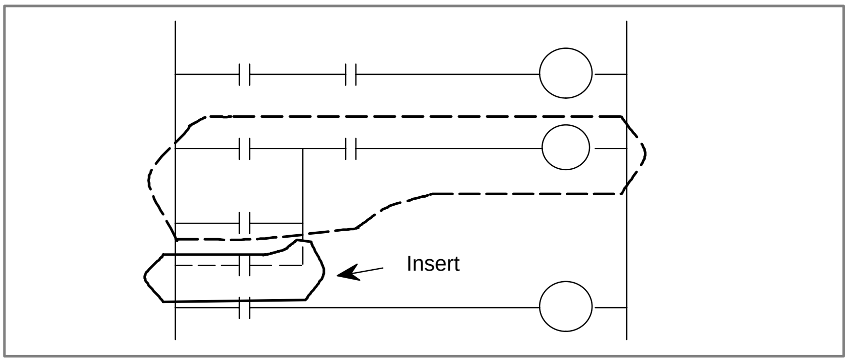
2 Press soft key [└—] for erasing the upper left vertical line. The upper left vertical line to the cursor disappears.

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.

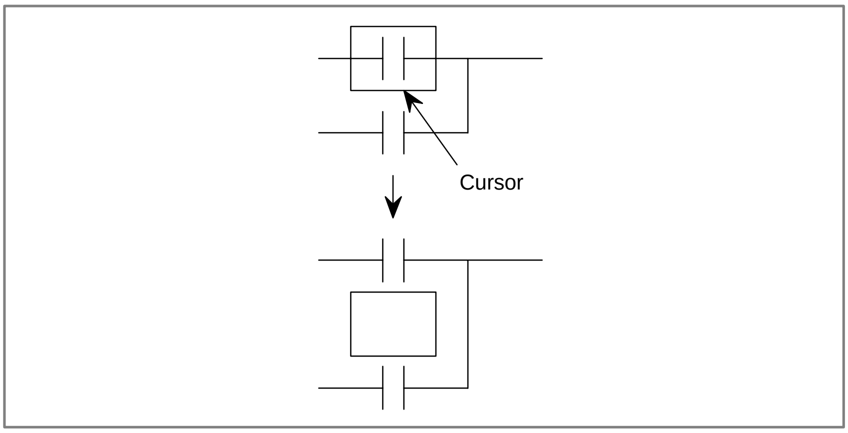
5 After setting the cursor to a position where the contacts is to be inserted, press soft key [⊣⊢]. After setting address data, press [INPUT] key. The cursor shifts rightward.

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.)

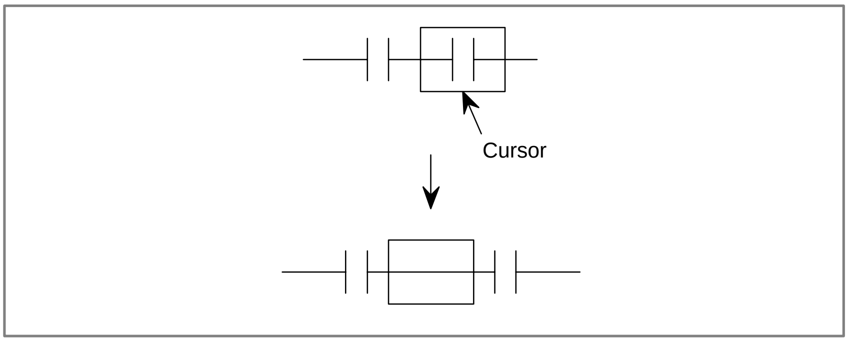


If the [INSLIN] key is pressed when the cursor is in the position specified as shown in the above figure on the left, the line is inserted as shown in the above figure on the right.

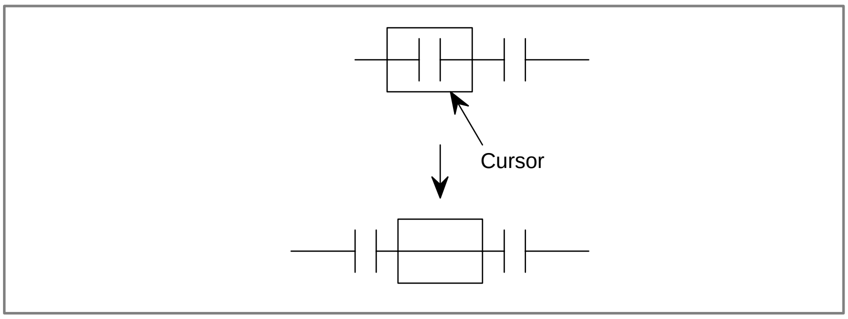
(4) Inserting the 1 NET sequence program elements

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.

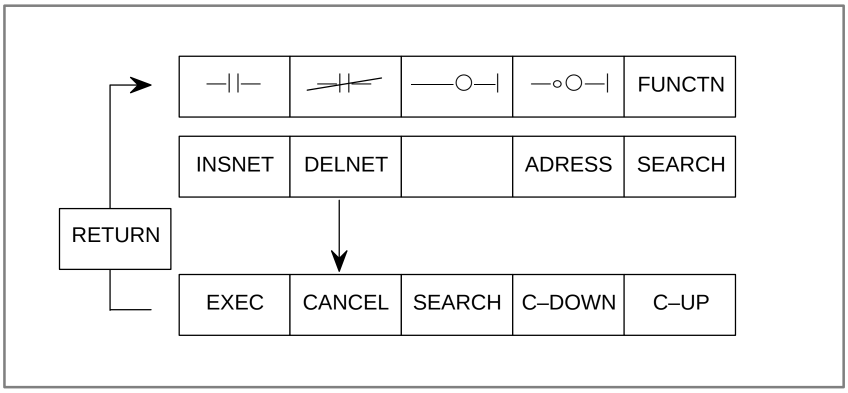
[—] : Delete of horizontal lines, relay contacts, relay coils, etc.

[↑—] : 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.

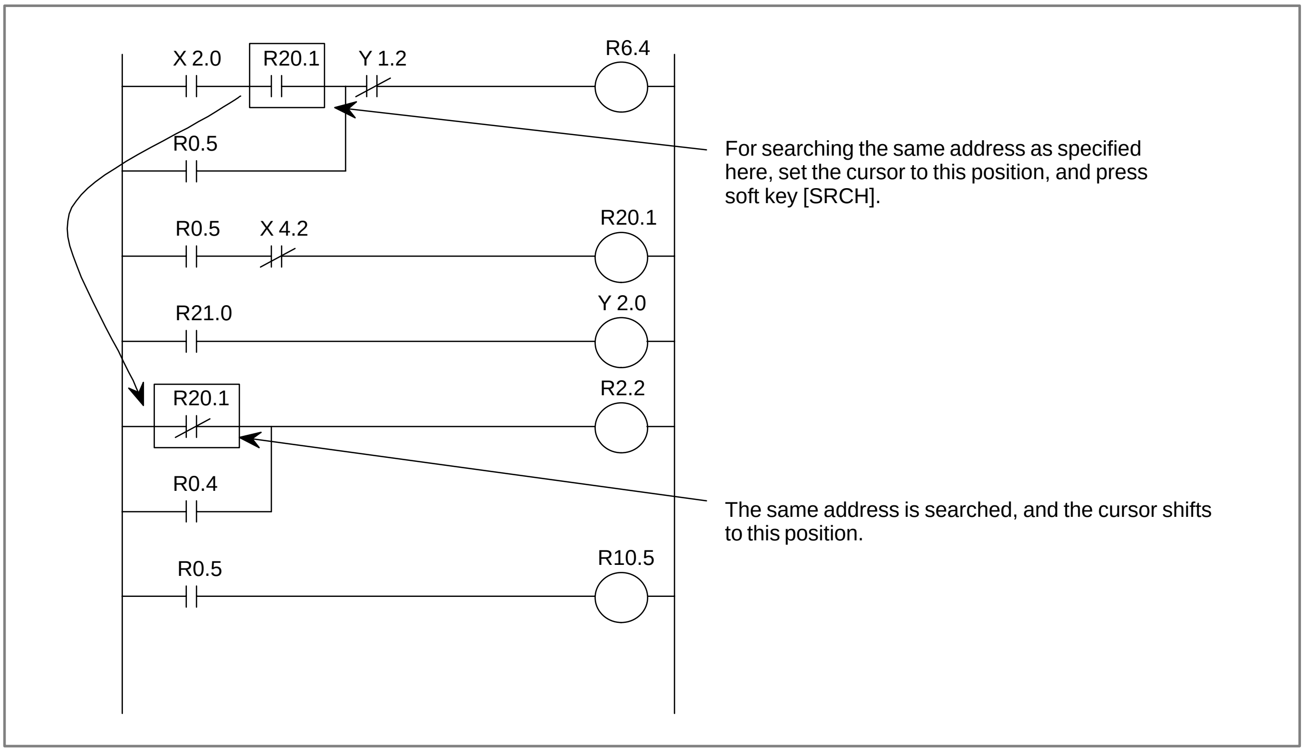
2 Deleting multiple NETs

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.

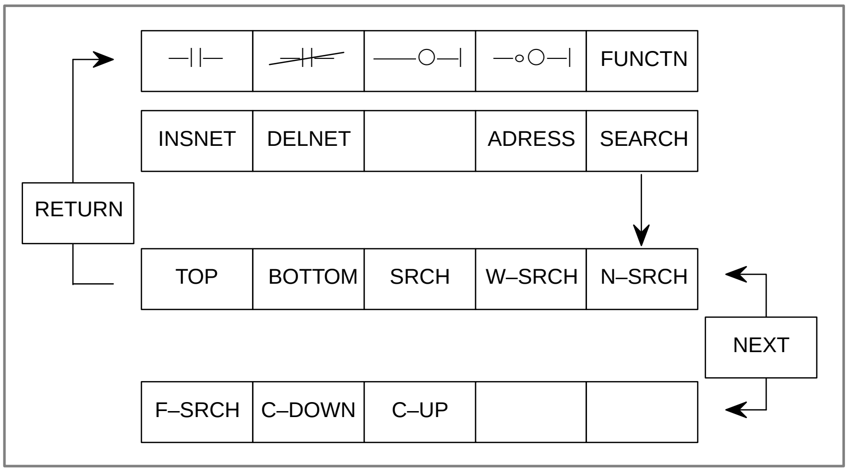
Cancel Press the [CANCEL] key.

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.

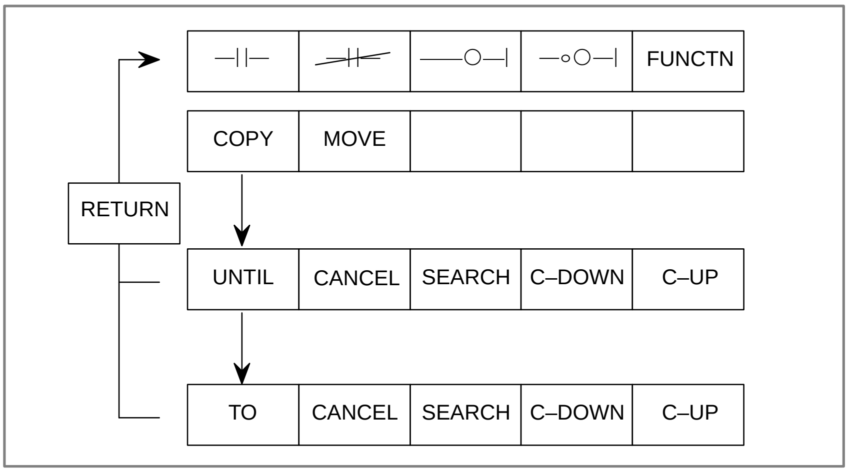
(7) Searching with cursor keys (<←>, <→>, <↑>, <↓>)

- 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

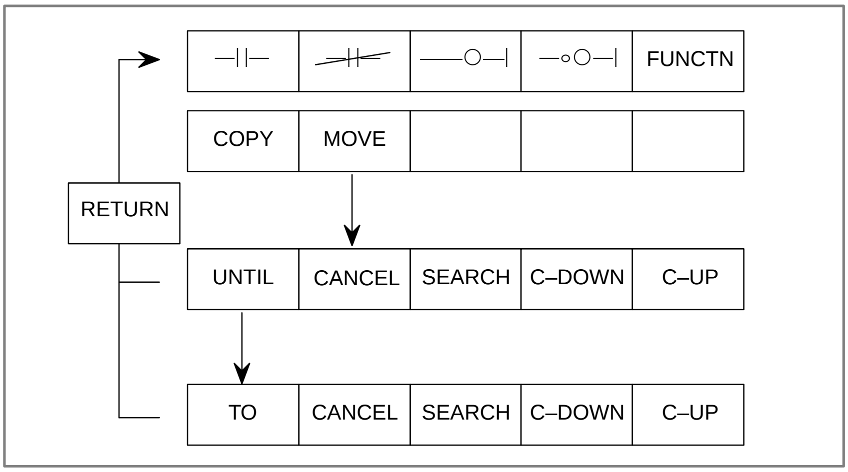
Press the [TO] key to start copying a NET. The NET is copied into the position above the cursor. If the number of copies is typed in before the [TO] key is pressed, the NET is copied the specified number of times.

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/".

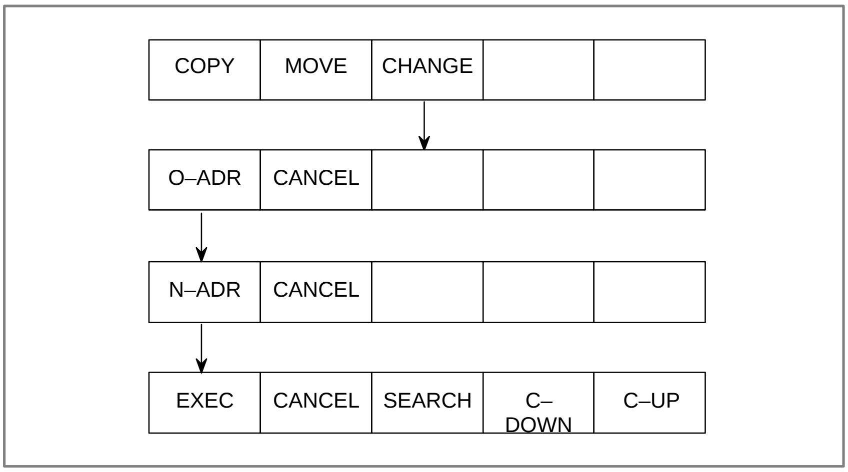
(3) The symbol data and comment only can be edited by the similar operation to the above (1) and (2).

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.

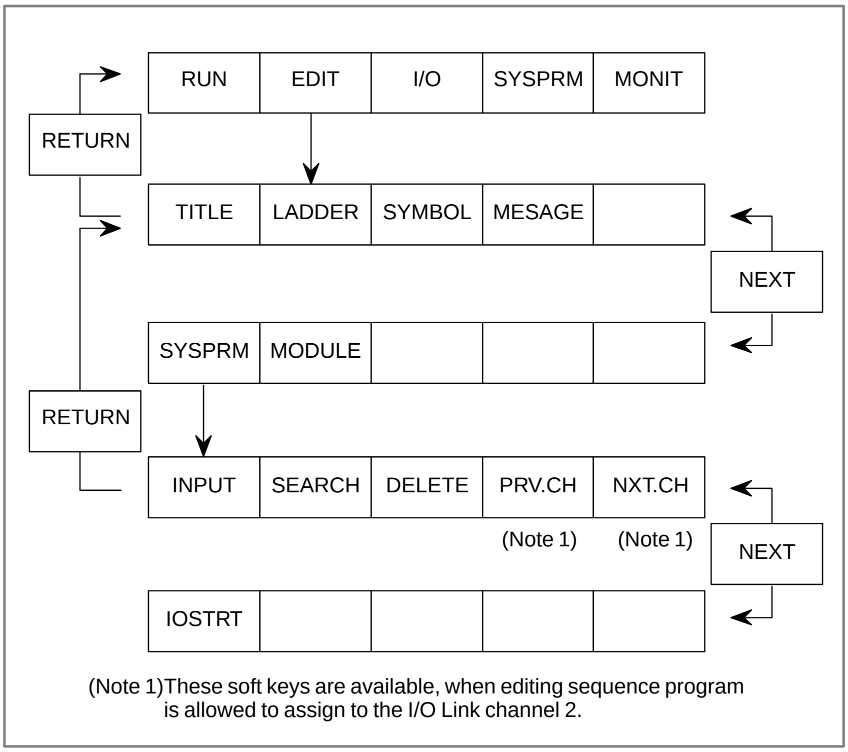


Fig. 5.3 Address setting for I/O unit

(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
   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
   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

GROUP.BASE.SLOT.NAME  =
>2.0.4.OD08C
```

I/O Unit MODEL B

I/O Unit MODEL A

Power Mate

> **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) Soft key [PRV.CH] or [NXT.CH]

1 It is available to assign the I/O Link expansion. This soft key is to change the I/O Link channel.

(6) Error and warning messages issued during the editing of assignment data

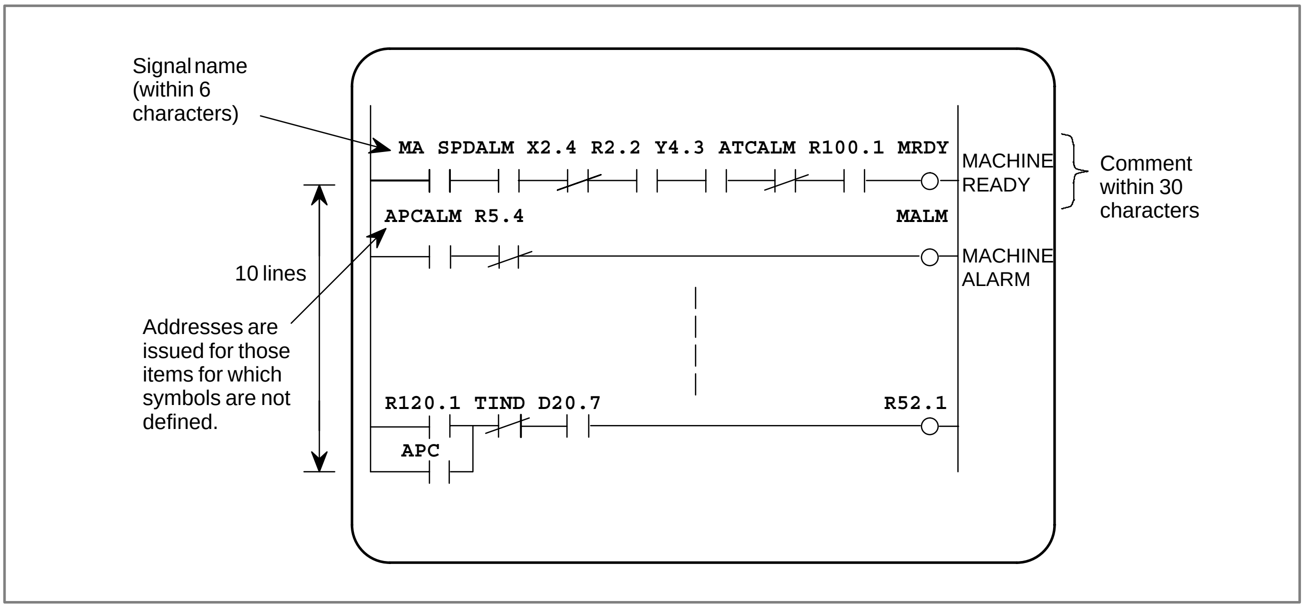
No.	Error or warning message	Description
1	ERR : GROUP NO. (0–15)	The group number must be from 0 to 15.
2	ERR : BASE NO. (0–3)	The base number must be from 0 to 3.
3	WARN : BASE NO. MUST BE 0	The base number must be 0 for the I/O Unit–B. It is forcibly set to 0.
4	ERR : SLOT NO. (1–10)	The slot number 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.

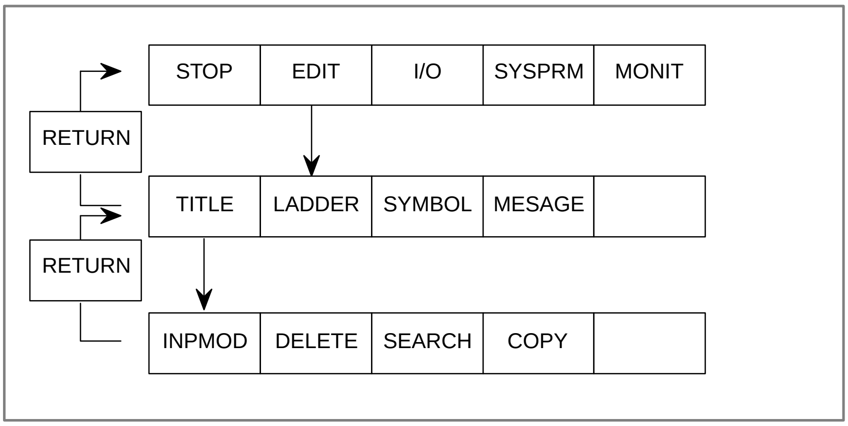


Fig. 5.4(a) Setting and display of symbol data

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.

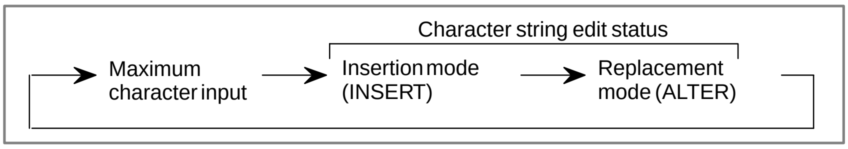
The cursor shifts to the corresponding address part.

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

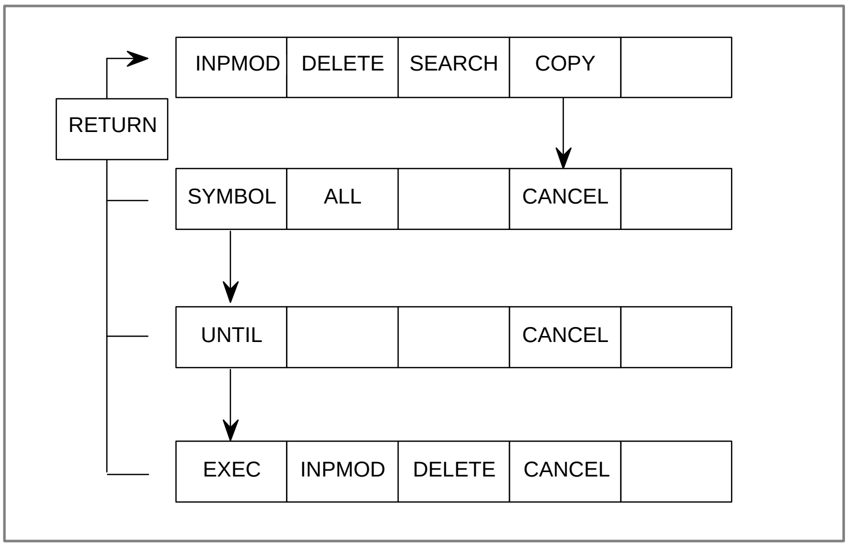
Move the cursor to the desired line of the address whose symbol and comment are to be edited.

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.

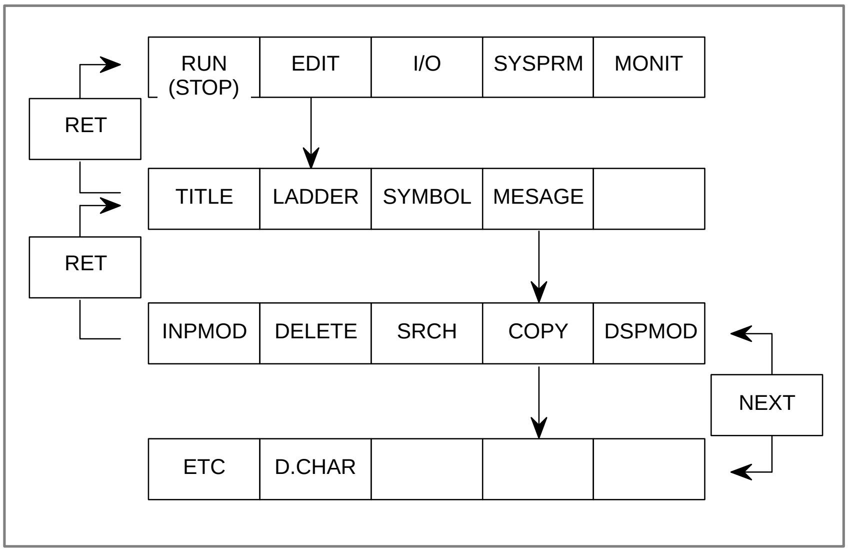


Fig. 5.5 (a)

When soft key MESAGE is pressed, the next screen is displayed, and setting operation can be done hereafter.

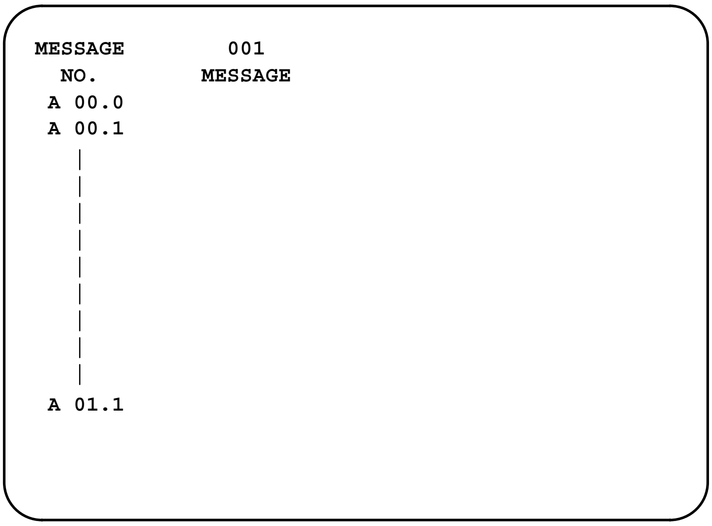


Fig. 5.5 (b) Message data screen

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.

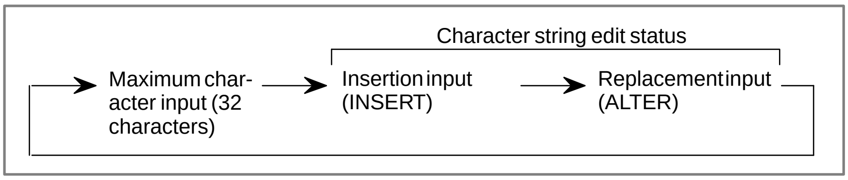
5.5.2 Searching for an Address (SRCH)

(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.)

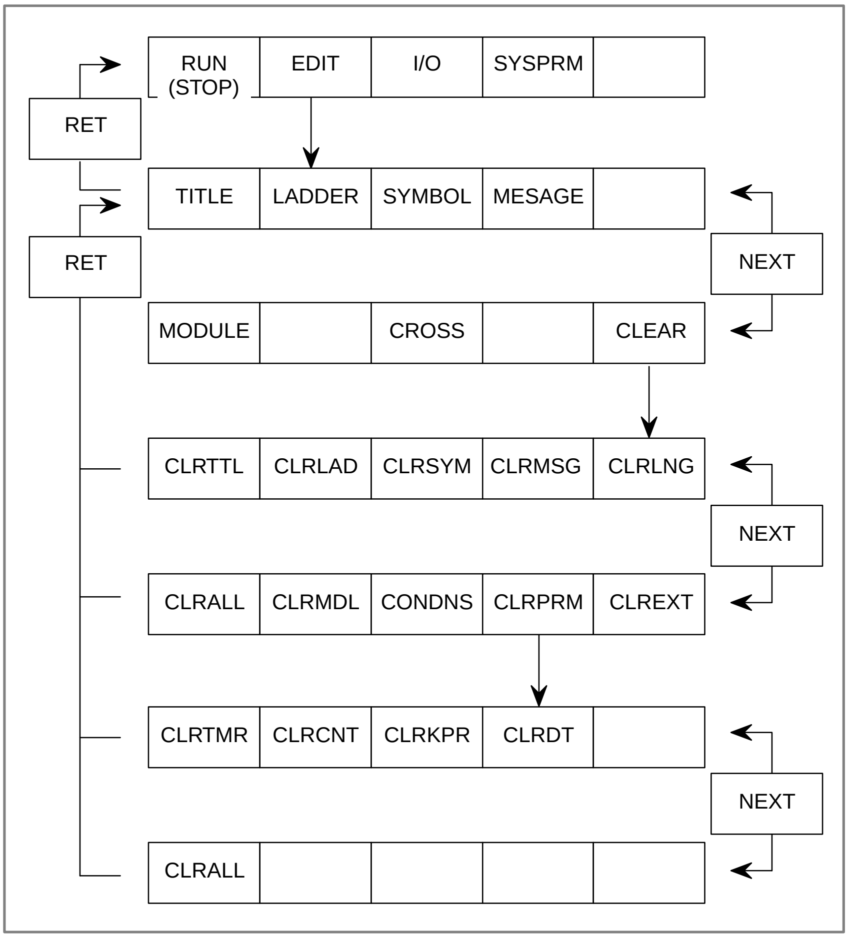
For example, “4873 [INPUT]” is processed as “@02487301@”.

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)

CAUTION

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).

CAUTION

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, · ·).

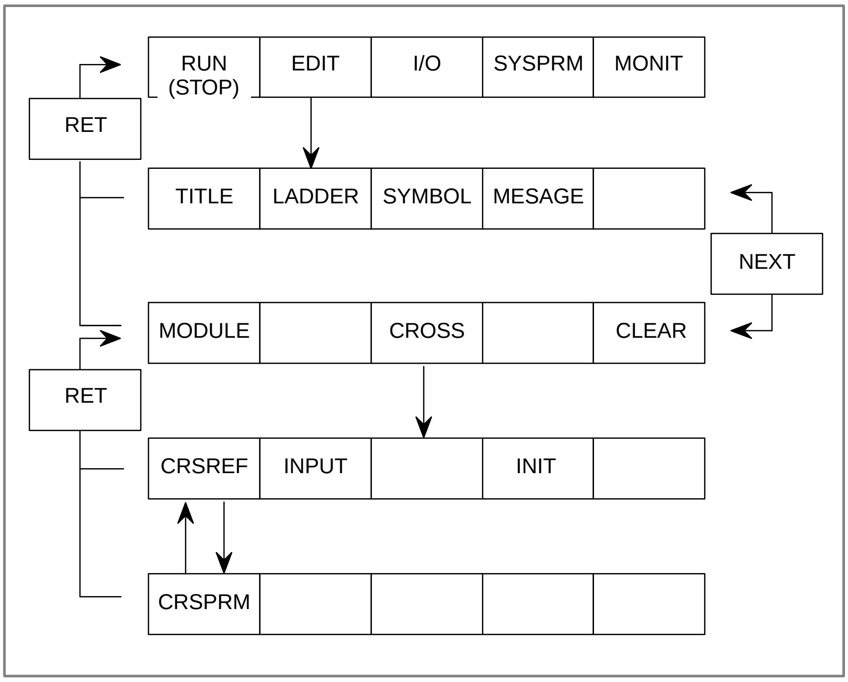
(3) Display a functional instruction list.

(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.

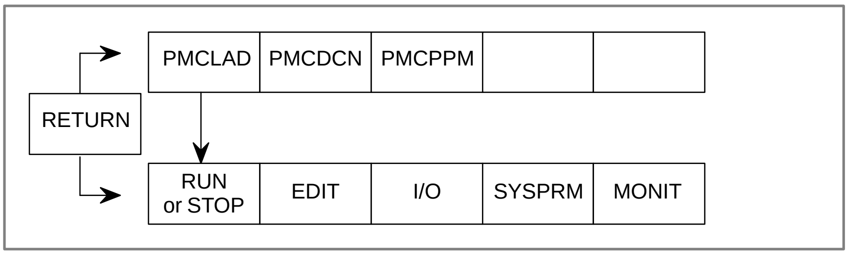


Fig. 6.1 The sequence program execution software key

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*/15*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)

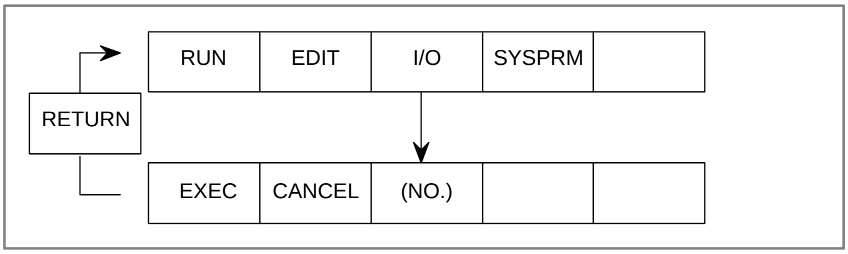
This method is effective for locating the error when a system error occurs after power is turned on and when the error may be caused by the sequence program.

Never perform this operation in a normal state.

WARNING
In normal operation, do not use the Z + CAN keys to stop the sequence program forcibly.

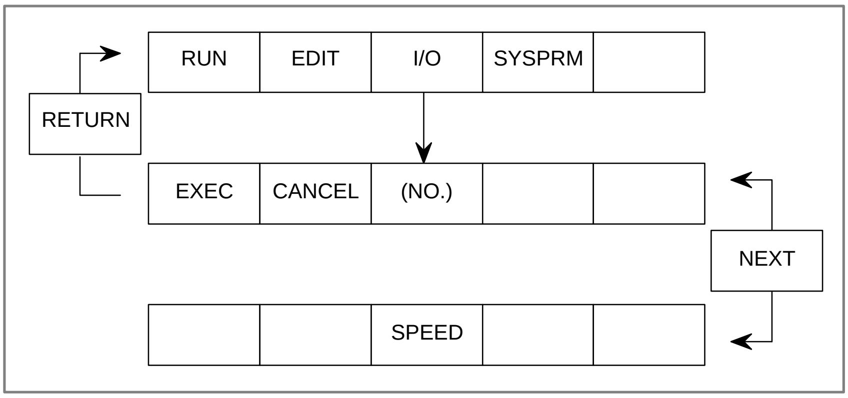
7 WRITING, READING, AND VERIFYING THE SEQUENCE PROGRAM AND PMC PARAMETER DATA

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 ]
```

CAUTION
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).

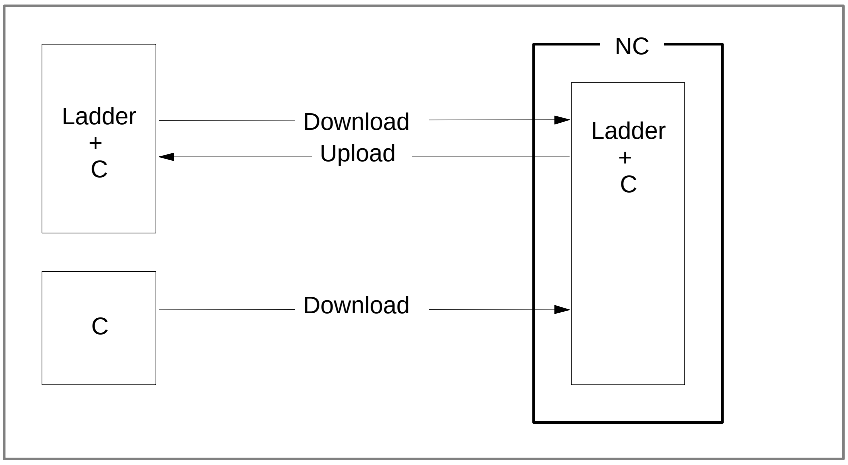


Fig. 7.1.1 (a) Ladder and C structure for FANUC Series 16B/16C

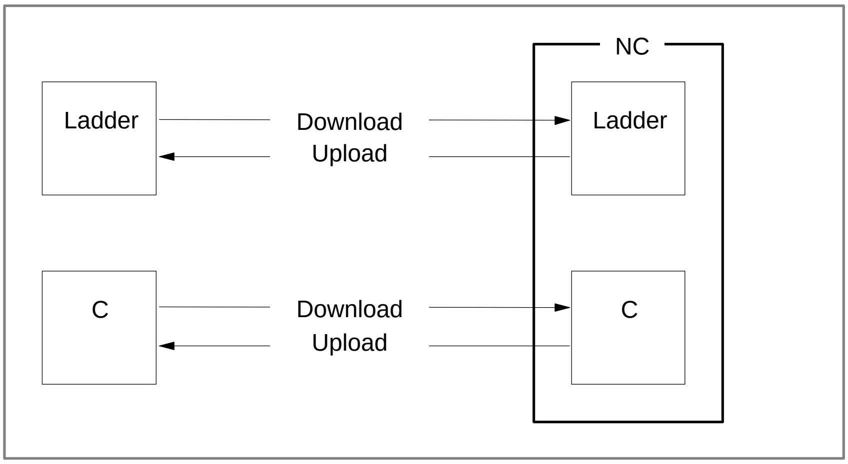


Fig. 7.1.1 (b) Ladder and C structure for FANUC Series 16*i*/18*i*/160*i*/180*i*

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 7.3.7 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
1 WRITE, READ, or COMPARE is automatically switched by operation on a FAPT LADDER.
2 This operation can not be used with PMC–SB7.

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.

CAUTION
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.

CAUTION
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.

NOTE
With the Power Mate–H, FS20, FS21/210MB, FS16–A, FS18–A, FS16–B, FS18–B, FS16–C, FS18–C, and FS15B, cards having a capacity of more than 32M bytes cannot be used.

The case of FS20,FS18B,FS16B FS18C, FS16C PMC

○ : Supported
× : Not supported

	SRAM Card	Flash Memory Card	ATA Card
Read of a file	○	○	×
Format of a card	○	×	×
Write of a file	○	×	×
Delete of a file	○	×	×
List of a file	○	○	×

The case of FS15B(PMC–NB)

○ : Supported
× : Not supported

	SRAM Card	Flash Memory Card		ATA 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	○	○	○	×

The case of FS16*i*, FS18*i*, FS21*i*

○ : Supported
× : Not supported

	SRAM Card	Flash Memory Card		ATA 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

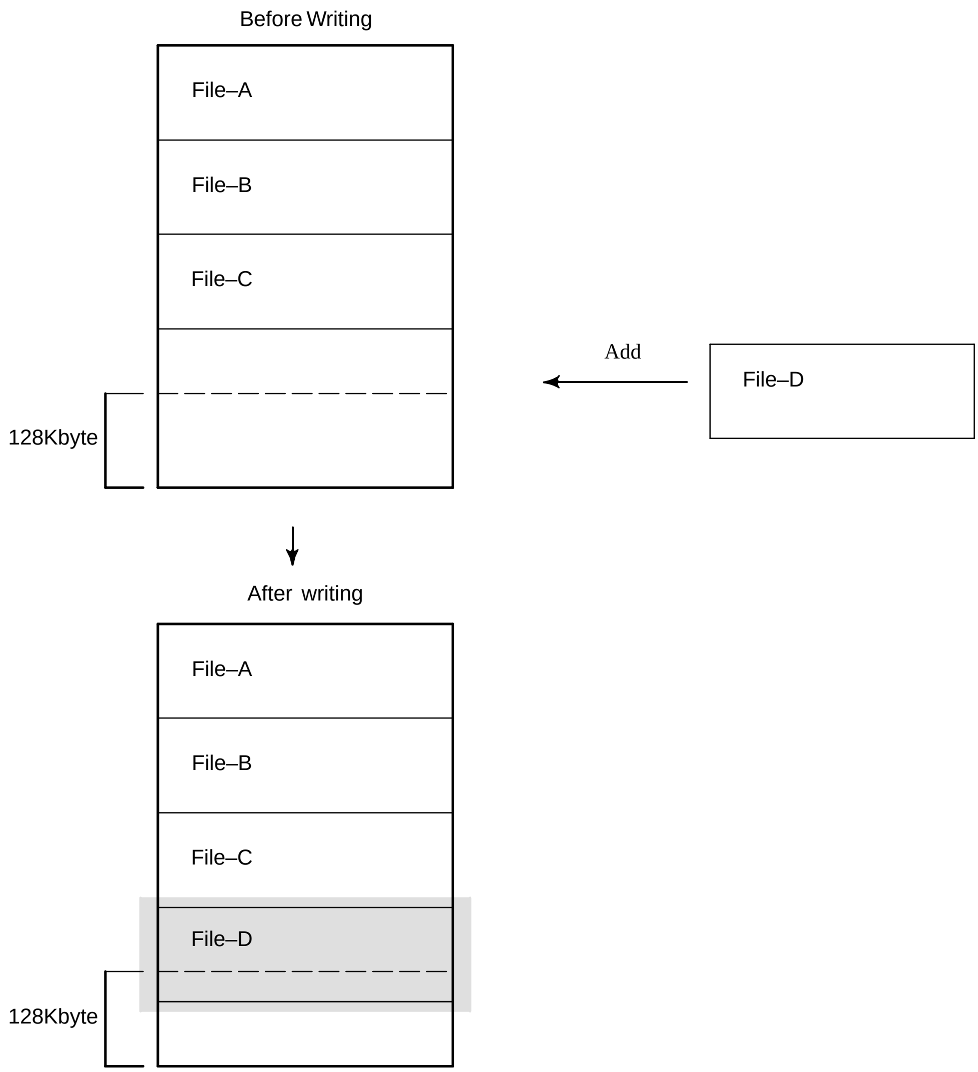
The following kinds of flash memory card are supported.

- 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.



“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.

CAUTION
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

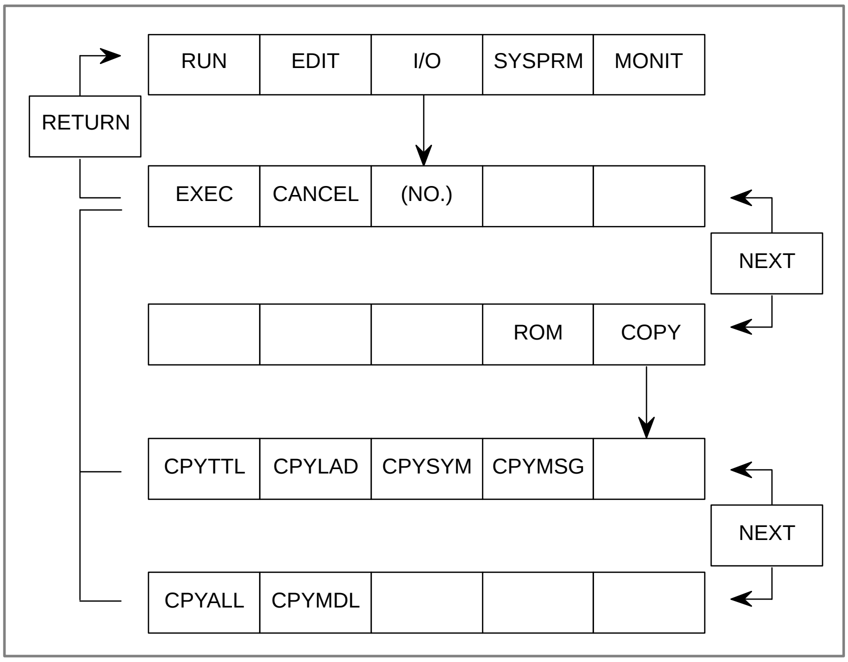
	Message	CONTENTS → OPERATION
FLSH ROM	PROGRAM ALREADY EXISTS	A program is already stored in the FLASH ROM (during blank check).
	PROGRAM ALREADY EXISTS (EXEC ?)	A program is already stored in the FLASH ROM (during writing or deleting data). Action) Press the EXEC key again when the message is displayed. 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.

CAUTION

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)

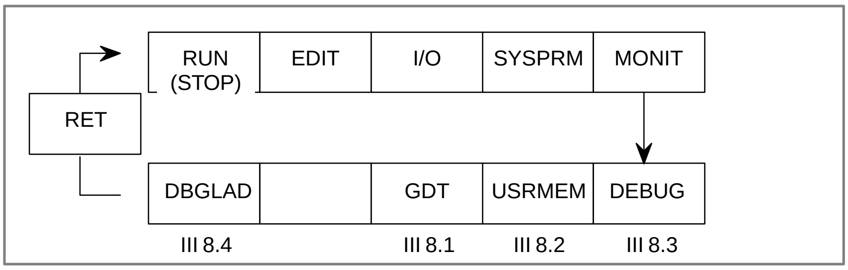
PA1	PA3	SA1	SA2	SA3	SA5	SB	SB2	SB3	SB4	SB5	SB6	SC	SC3	SC4	NB	NB2
×	×	×	×	×	×	×	×	×	×	Δ	Δ	○	○	○	○	○

Work RAM is necessary (A02B–0120–H987 for the PMC–SC, SC3, and SC4 and A02B–0162–J151 or A02B–0162–J152 for the PMC–NB and NB2).

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/SC4/NB/NB2 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

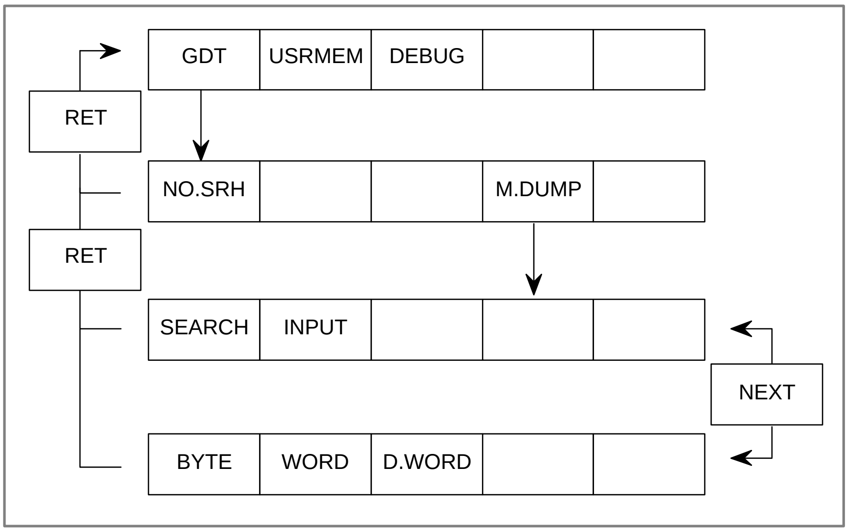
8.1 DISPLAYING THE GDT (GLOBAL DESCRIPTOR TABLE)

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.

WARNING

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

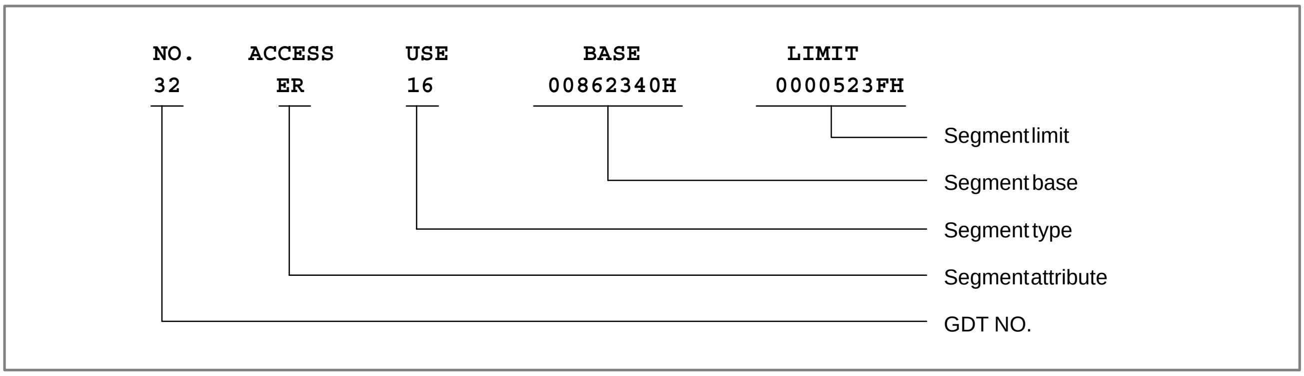
```
PMC DESCRIPTOR TABLE(GDT)     MONIT RUN

 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.

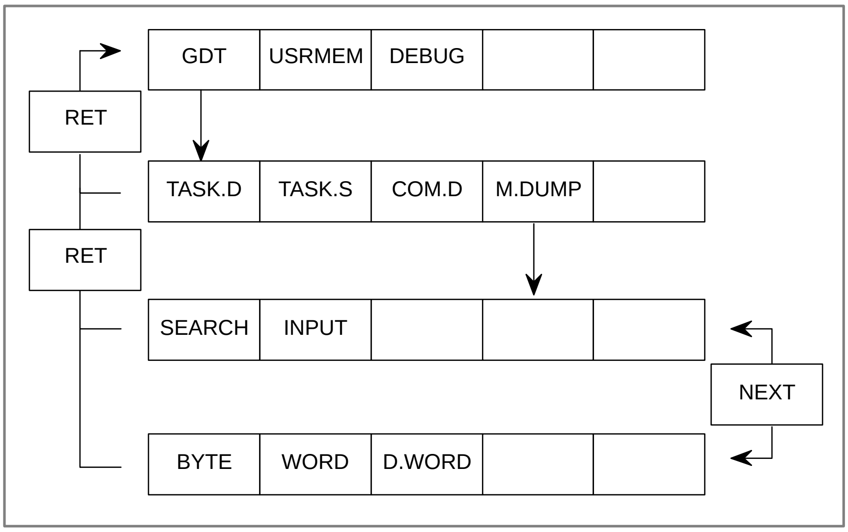
The segment information of the following areas defined by a user program for each task can be displayed and the contents of the areas can be dumped.

- 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] [      ]
```

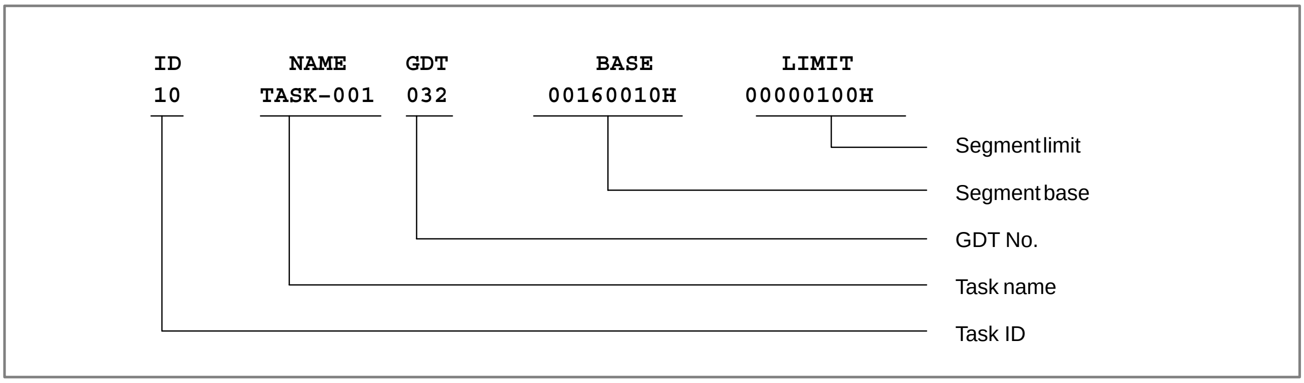
Fig. 8.2.1(b) Information of a task stack area

```
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] [      ]
```

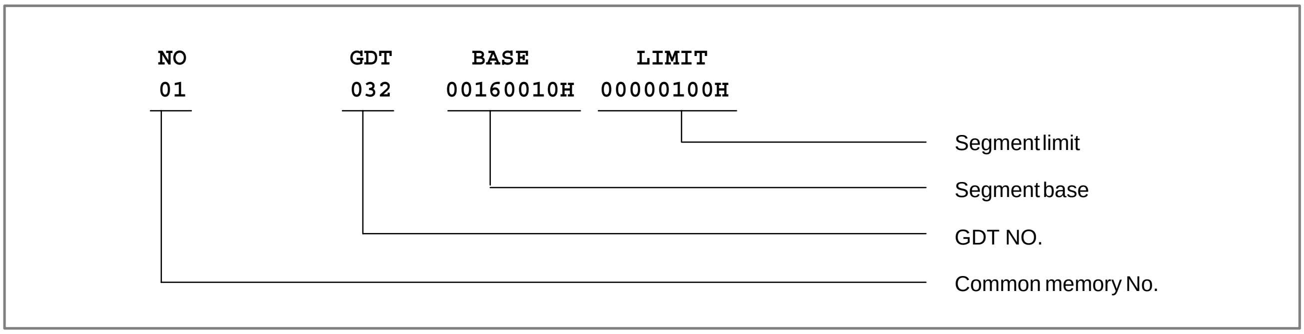
Fig. 8.2.1(c) Information of a common memory data area

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

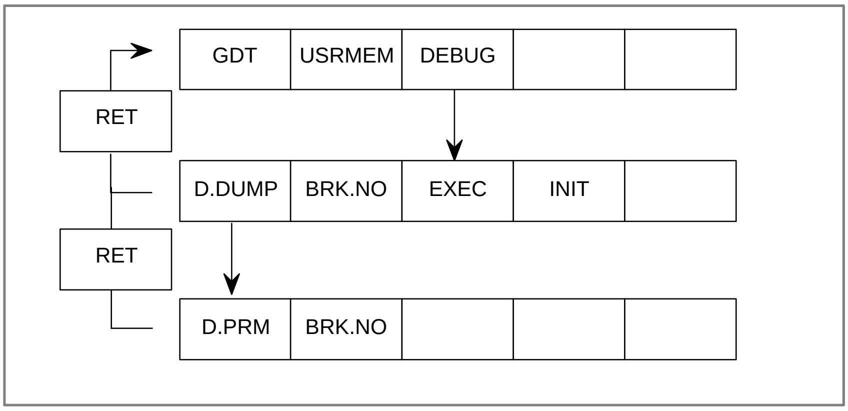
8.3.2 Operation

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.

CAUTION

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.

CAUTION

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	SB7	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

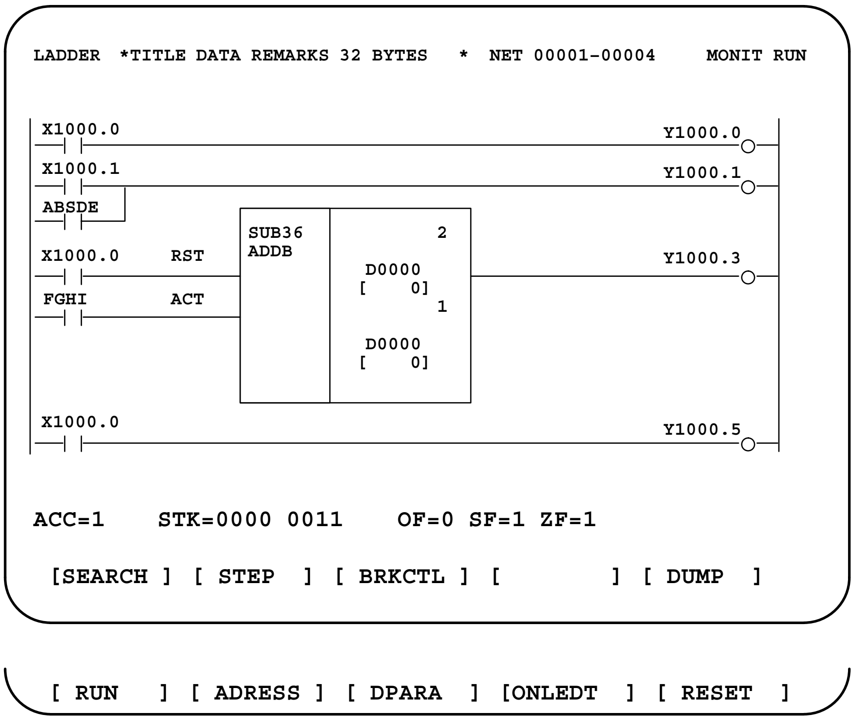
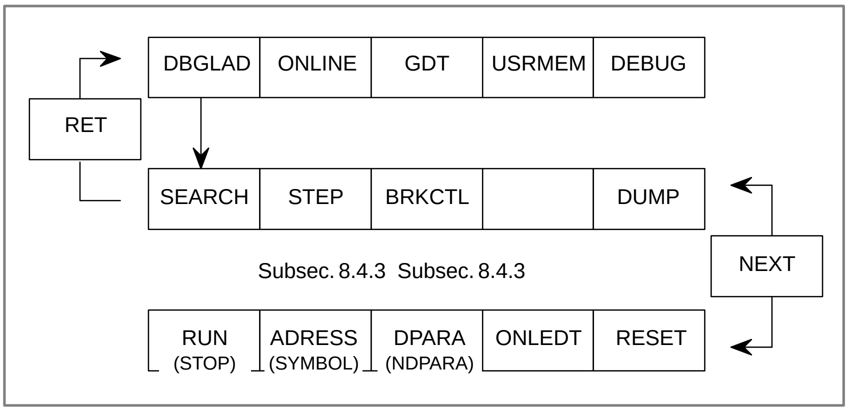


Fig. 8.4.1 Screen of ladder debugging function

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.

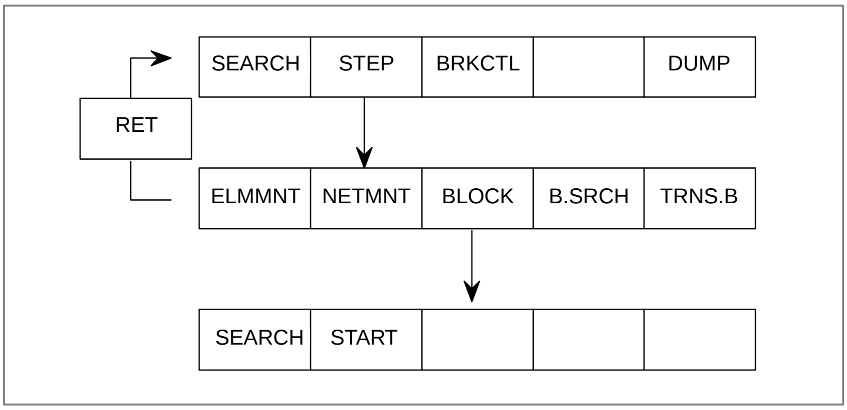
(3) Step operation to execute from current position to specified contact or coil instruction.

[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)

CAUTION
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

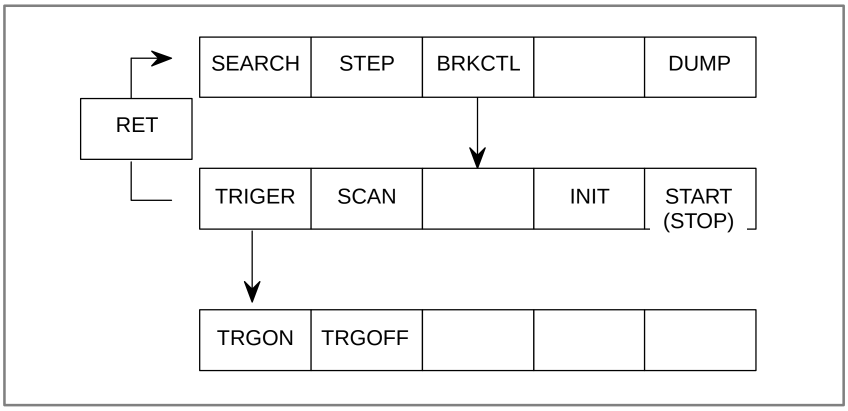
ADR : specified trigger address

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
1 PMC–PA3 is usable with the Power Mate–D/H.
2 PMC–SA1 is usable with the loader control function of the Series 21*i*.
3 PMC–SA3 is usable with the FANUC NC Board.
4 If the system is set up to use the on–line function, that line is occupied, and cannot be used with other input/output functions. To use the line with other input/output functions, set up the system so as not to use the on–line function.
If the system is set up to use the on–line function, the following functions cannot be used:
- [PMCLAD], [I/O], [EDIT], [SYSPRM], [TRACE], [ANALYS], [USRDGN], [DGNLAD], [GDT], [USRMEM]

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 (This cannot be used with the PMC–SA1.)
- 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

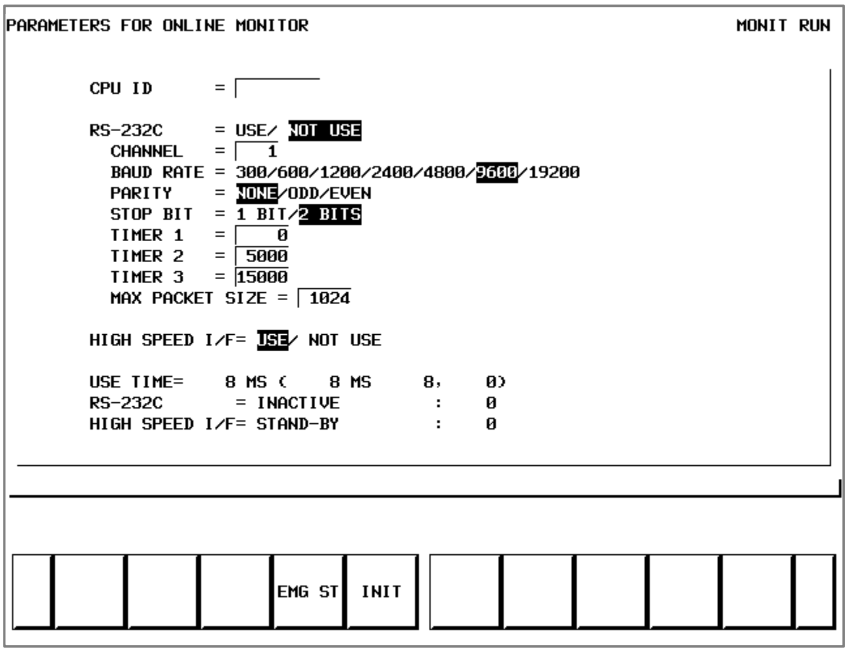


Fig. 8.5.1 Online monitor setting screen

Meanings of Soft key

EMG ST : Terminates communication forcibly. Use this key if communication becomes abnormal and the connection cannot be terminated normally.

INIT : Initializes the parameters to their default values.

NOTE

1 In case of configuration of CNC with which neither Ethernet nor HSSB is available, the item of ”HIGH SPEED I/F” is not displayed.

2 In case of display which has 5+2 soft key, two pages are used for this setting screen.
Switch the page by <Page Up> or <Page Down> key.

8.5.2 Setting of online connection

To communicate with FAPT LADDER–II, FAPT LADDER–III or Ladder Editing Package, you need to put the PMC system in waiting situation of the connection. There are two ways for setting this, setting at PMC screen and setting in NC parameter. Also, there are three connection types, for example Ethernet, RS–232C or HSSB.

8.5.2.1 How to set at PMC Screen

To display the soft key [MONIT] in the PMC main menu screen, set ”PROGRAMMER ENABLE” to ”YES” in the setting screen. When pushing the soft key [MONIT]–>[ONLINE], the online setting screen is displayed. (Fig. 8.5.1)

1. Case of connection by RS–232C (FAPT LADDER–II, FAPT LADDER–III)
   (1) Check that ”NOT USE” is selected at the ”RS–232C” item.
   (2) Set the parameter of ”CHANNEL” and ”BAUD RATE”.
   (3) Move the cursor to the ”RS–232C” item with Up or Down Cursor key.
   (4) Select ”USE” with Left or Right Cursor key.
2. Case of connection by Ethernet (FAPT LADDER–III, Ladder Editing Package)
   (1) Move the cursor to the ”HIGH SPEED I/F” item with Up or Down Cursor key.
   (2) Select ”USE” with Left or Right Cursor key.
3. Case of connection by HSSB (Ladder Editing Package)
   (1) Move the cursor to the ”HIGH SPEED I/F” item with Up or Down Cursor key.
   (2) Select ”USE” with Left or Right Cursor key.

NOTE
1 When both ”RS–232C = USE” and ”HIGH SPEED I/F = USE” are selected, the PMC system will communicate with the application which is connected at first. If PMC system is already connecting with an application, it can not connect with other applications.
2 When you use the online function by Ethernet, the setting of Ethernet parameters at CNC is necessary in advance.
3 Loader control function can not connect with FAPT LADDER–III or Ladder Editing Package by Ethernet.

8.5.2.2 How to set at NC parameter

(1) Display the No.24 of NC parameter.

(2) To connect by Ethernet or HSSB, input ”0”, ”11” or ”12”.
To connect by RS–232C, input ”1”, ”2”, ”11” or ”12”.

Contents of NC parameter No.24.

NC parameter No.24	Setting of the PMC online monitor screen		
	RS–232C	**HIGH SPEED I/F**	**Meanings**
0	NOT USE	USE	The item of ”RS–232C” is changed to ”NOT USE”. The item of ”HIGH SPEED I/F” is changed to ”USE”.
1	USE (Channel 1)	NOT USE	The item of ”CHANNEL” is set ”1” and the item of ”RS–232C” is changed to ”USE”. The item of ”HIGH SPEED I/F” is changed to ”NOT USE”.
2	USE (Channel 2)	NOT USE	The item of ”CHANNEL” is set ”2” and the item of ”RS–232C” is changed to ”USE”. The item of ”HIGH SPEED I/F” is changed to ”NOT USE”.
11	USE (Channel 1)	USE	The item of ”CHANNEL” is set ”1” and the item of ”RS–232C” and ”HIGH SPEED I/F” are changed to ”USE”.
12	USE (Channel 2)	USE	The item of ”CHANNEL” is set ”2” and the item of ”RS–232C” and ”HIGH SPEED I/F” are changed to ”USE”.
3 to 10 13 to 254	(Reserved)	(Reserved)	Don’t use this setting.
255	NOT USE	NOT USE	Terminates communication forcibly.It is the same effect as soft key [EMG ST].

NOTE
1 If the value of No.24 in NC parameter is changed, the setting of the online monitor screen of PMC is automatically changed too. This NC parameter is made effective immediately after changing setting of the parameter.
2 Even if the setting of the online monitor screen of PMC is changed, the value of No.24 in NC parameter is not changed

8.5.2.3 How to set at NC parameter (Power Mate–D/H)

(1) Display the No.101 of NC parameter.

(2) To connect by RS–232C, input ”1” to No.101#6.

	#7	#6	#5	#4	#3	#2	#1	#0
0101		1						

#6 0 : Online monitor driver is not used.
1 : Online monitor driver is used.

> **NOTE**
> 1 In case of Power Mate–D 2 path–control, only a parameter for 1 path is available.
> 2 The online monitor driver occupies the line while it is operating. In this state, other input/output functions cannot use the line. For other input/output functions to use the line, it is necessary to display the above–mentioned parameter and stop the online monitor driver.
> 3 While the online 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
>
> 4 In case of operating NC, the screen display of NC (Position, etc.) might be slow when using input/output functions (Load program, Store program, etc.). There is no problem in the operation of NC. It is recommended to use input/output functions while NC is not operating.
> 5 When the screen made by C language executor is displayed, the communication may be slow down. It is recommended to use input/output functions after moving to other screens (Position, etc).

8.5.3 Online Function by Ethernet

8.5.3.1 Setting of Ethernet parameters

When you try to connect FAPT LADDER–III or Ladder Editing Package (Window version) with CNC by Ethernet, it is necessary to set some Ethernet parameters. The setting of Ethernet parameters can be set in the following Ethernet parameter screen of CNC. Please refer to ”FANUC Ethernet Board/DATA SERVER Board OPERATOR’S MANUAL” (B–63354EN) about the detail of the setting screen and setting parameters. The setting item necessary for Ethernet connection for PMC online function is as follows.

– IP ADDRESS (Set the IP address of CNC. 192.168.0.1 etc.)

– SUBNET MASK (Set the mask address of the IP address. 255.255.255.0 etc.)

– ROUTER IP ADDRESS (If you use the router, set the Router IP Address.)

– PORT NUMBER (TCP) (8193 etc.)

```
ETHERNET PARAMETER
                                 PAGE: 1/ 2
 MAC ADDRESS             XXXXXXXXXXXX
 NUMBER OF SCREENS                 14
 MAXIMUM PATH                       1
 HDD EXISTENCE                      0

 IP ADDRESS          192.168.0.1
 SUBNET MASK         255.255.255.0
 ROUTER IP ADDRESS
```

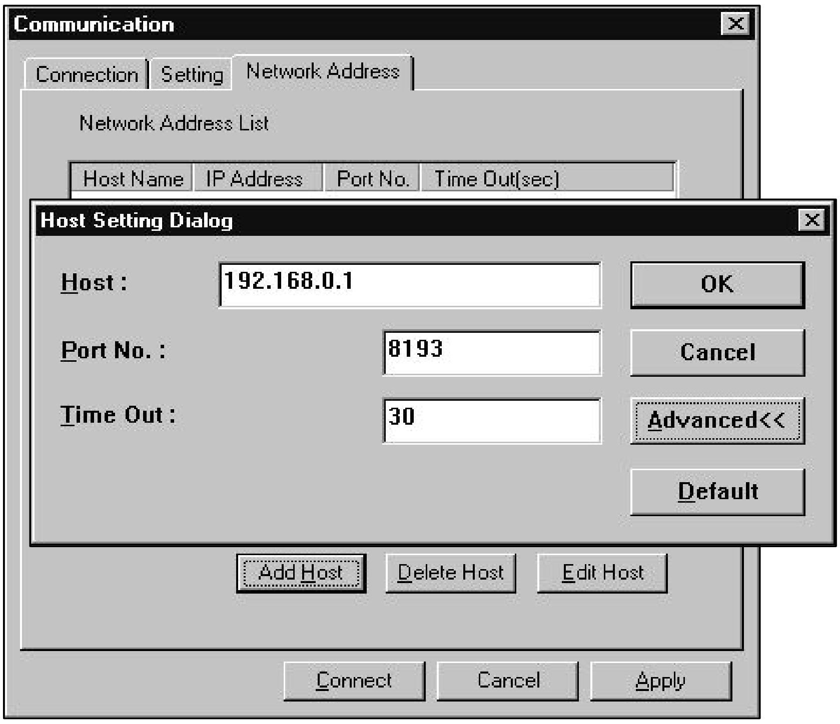
```
ETHERNET PARAMETER
                                 PAGE: 2/ 2
(DNC1/ETHERNET)
 PORT NUMBER(TCP)                 8193

 PORT NUMBER(UDP)                    0

 TIME INTERVAL                       0
```

8.5.3.2 Starting online communication by offline programmer (Ethernet connection)

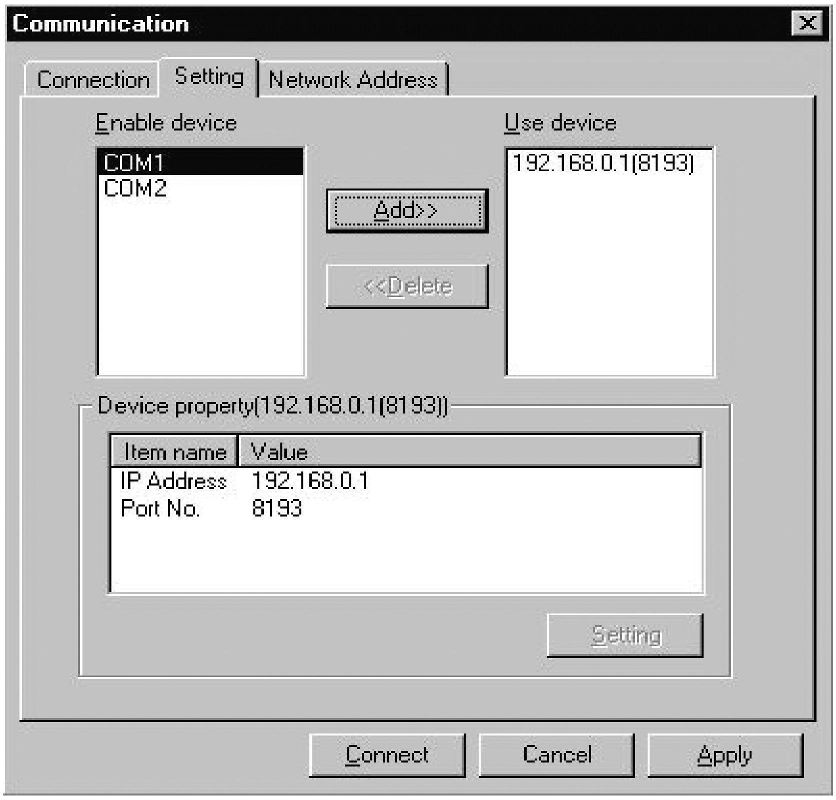
The procedures for online connection with PMC and the offline programmer (FAPT LADDER–III, Ladder Editing Package for windows) by Ethernet are as follows.
(Example: FAPT LADDER–III)

(1) Start up FAPT LADDER–III, and click the [Communication] on [Tool] menu.

(2) Select the [Network Address] tab and push the <Add Host> button. Input the "IP Address" and "Port No." inputted by "8.5.3.1 Setting of Ethernet parameters".



(3) Select the [Setting] tab, and add the IP Address to ”Use device”.



(4) Push the <Connect> button for start of the communication.

Ladder Editing Package can be connected by the same operation too. Refer to ”FAPT LADDER–III OPERATOR’S MANUAL”(B–66234EN) as for the detail of operation of FAPT LADDER–III. Refer to ”LADDER EDITING PACKAGE (Windows) OPERATOR’S MANUAL” (B–63484EN) as for the detail operation of Ladder Editing Package.

CAUTION

1 When one of the following screens is displayed at PMC, the online communication can not be used. Change to other screens from the following screens, and use the online function.
[PMCLAD], [I/O], [EDIT], [SYSPRM], [TRACE], [USRDGN], [DBGLAD], [GDT], [USRMEM]

2 When the online function is used with RS–232C, the selected channel is occupied by the PMC system. To use other input/output functions with RS–232C, specify other channel setting than the one used by online function.

3 Loader control function can not connect with FAPT LADDER–III or Ladder Editing Package by Ethernet.

8.5.4 Communication Status

The communication status of RS–232C and HIGH SPEED I/F are displayed at the online monitor screen during the online communication.

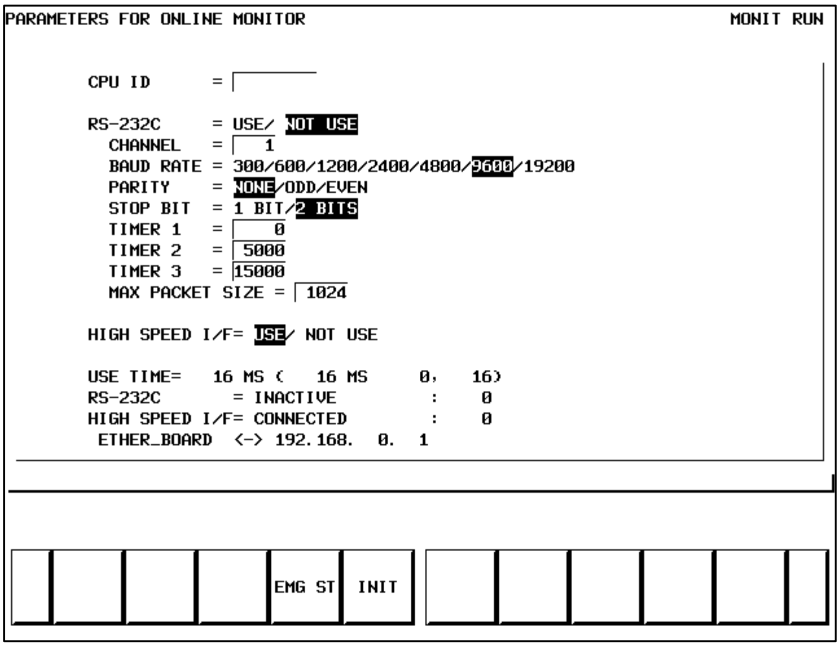


Fig. 8.5.4 Online monitor setting screen

USE TIME	:	The maximum time in the communication processing is displayed.
RS–232C	:	The communication condition of RS–232C is displayed.
HIGH SPEED I/F	:	The communication condition of HIGH SPEED I/F is displayed.
ETHER_BOARD	:	Displayed during the communication with Ethernet board. The IP address of the communication partner is displayed.
HSSB	:	Displayed during the communication with HSSB.

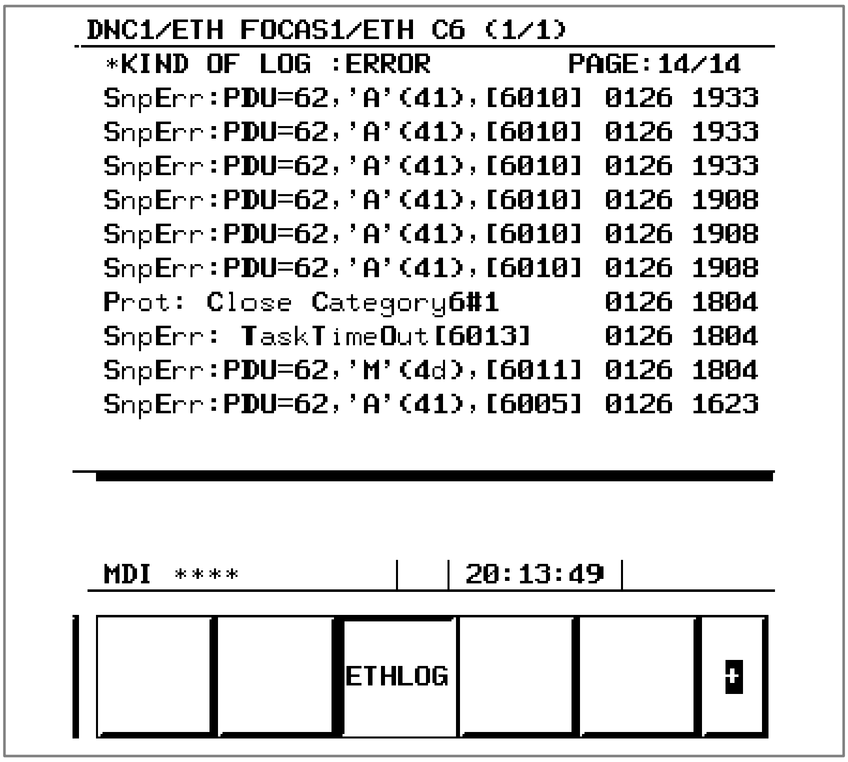
The display messages and the meanings are shown in the table of below.

Displayed messages	Meanings
INACTIVE	The communication is inactive.
STOPPING	The communication is being stopped.(Wait for the termination of communication)
STARTING	The communication is being started.(Wait for the termination of communication over another communication path)
STAND–BY	The communication is active and in standby mode.
CONNECTED	The communication is active and being connected.
NO OPTION	The port can be not opened because there is not option of RS–232C.
BAD PARAMETER	Invalid open parameters are specified.
TIMEOUT ERROR	A time–out has occurred and communication is aborted.

Displayed messages	Meanings
TIMEOUT(K) ERROR	A time–out has occurred and communication is aborted.
BCC ERROR	A Block Check Code (packet parity) error has occurred.
PARITY ERROR	A parity error has occurred.
OVER–RUN ERROR	A reception overrun has occurred and the communication can not recover.
SEQUENCE ERROR	Packets are out of sequence.(Incorrect procedure)
DATA ERROR	Incorrect packets have been received through retry process.
QUEUE OVERFLOW	The transmit/receive queue has overflowed.
DISCONNECTED	Communication has been terminated successfully.
NO CONNECTION	The cable is disconnected.

8.5.5 About Connection Log of Ethernet

If any errors have occurred during Ethernet connection, the contents of the errors are displayed at ”ETHLOG” screen of CNC. Refer to this screen when the communication does not start.



Connection Log	Meanings and countermeasures
SnpErr:PDU=*m,n,*[*x*] *date time* SnpErr:PDU=*n,*[*x*] *date time* SnpErr:TaskTimeOut[*x*] *date time*	An error has occurred during the online communication. *m, n* : Online communication information that is internal information of a system. *x* : Error information 6001 PMC does not support the Ethernet.Confirm the Series/Edition of PMC software. 6003 Unsupported command data was received.Confirm the Series/Edition of Ethernet board software. 6004 There was an error in command dataConfirm the Series/Edition of Ethernet board software. 6005 PMC does not receive command data.Confirm the communication status at the online setting screen of PMC. 6010 PMC does not receive command data.Confirm if ”HIGH SPEED I/F=USE” is selected and other application is not connected at the online setting screen of PMC. 6011 Time–out error occurred at PMC.Increase the value of ”Time Out” in [Network Address] of [Communication] menu for FAPT LADDER–III or Ladder Editing Package. 6012 PMC does not receive command data because it is busy for processing.Confirm the communication status at the online setting screen of PMC. 6013 Time–out error occurred at PMCIncrease the value of ”Time Out” in [Network Address] of [Communication] menu for FAPT LADDER–III or Ladder Editing Package. 6101 PMC received an unsupported function code.Confirm the Series/Edition of PMC software. *date* : The date when the error occurred. Ex.) ”0126” means January 26th. *time* : The time when the error occurred. Ex.) ”1623” means 4:23 PM.

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)

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)

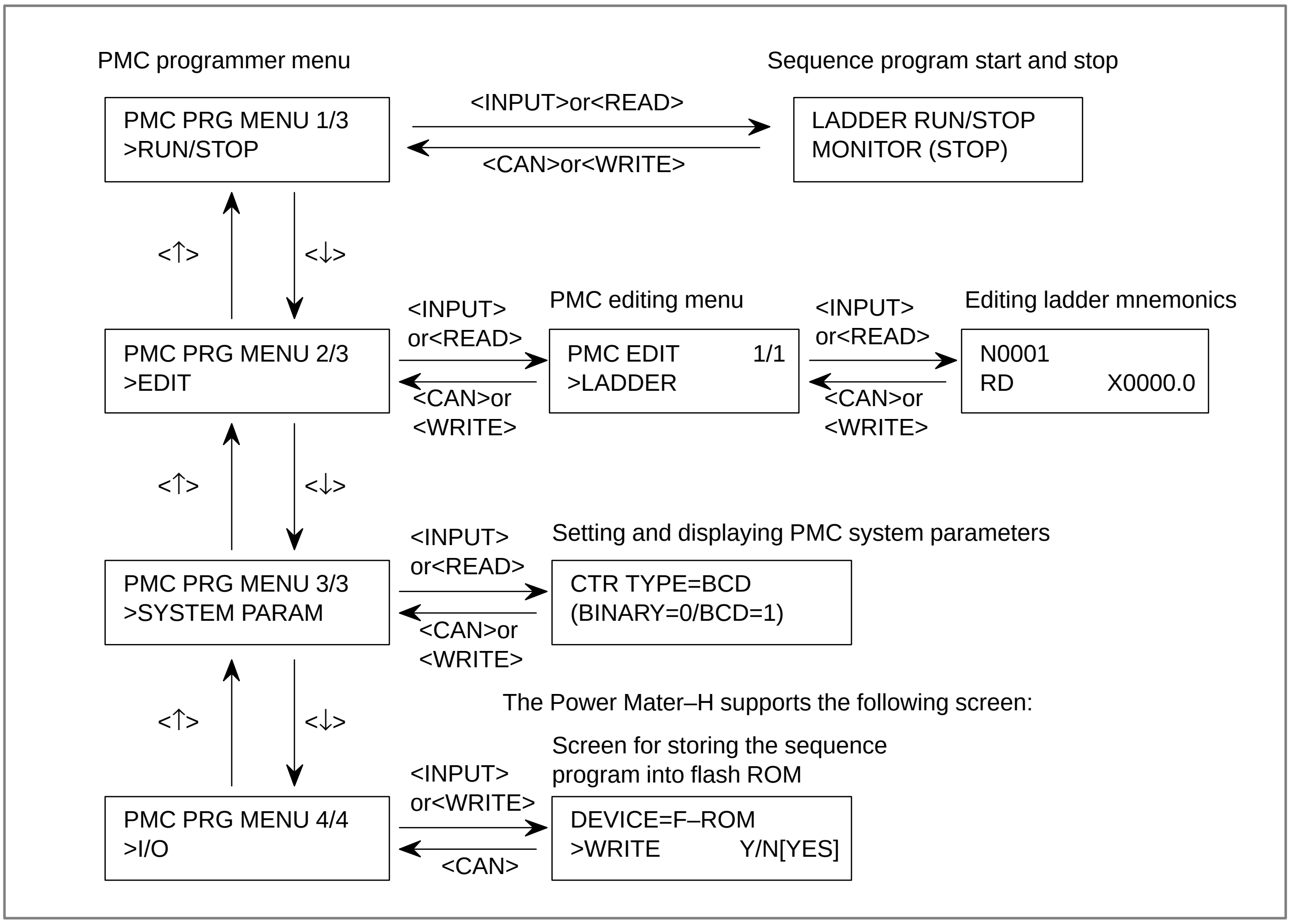
– The sequence program can be stored into flash EEPROM (only for the Power Mate–H and Power Mate *i*–D/H).

The DPL/MDI panel is shown below.

Fig. 11 DPL/MDI panel for Power Mate

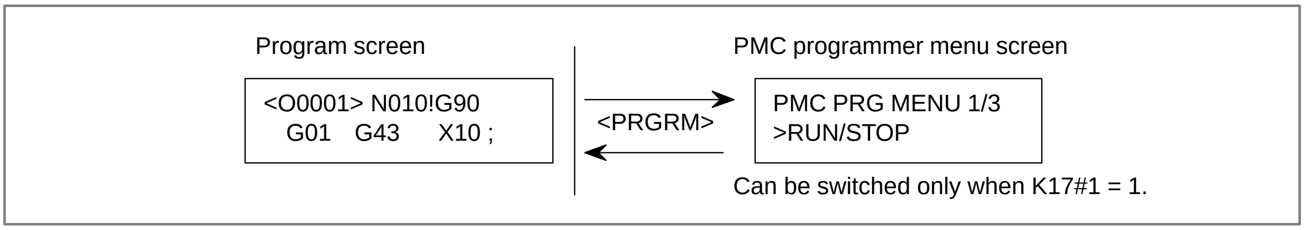
> **NOTE**
> 1 With the PMC programmer (DPL/MDI) function, the characters indicated at the lower–left part of each key are used.
> 2 When the <D/R> key is pressed once, the left–hand character is valid. When the <D/R> key is pressed twice, the right–hand character is valid.
> Example : When the <D/R> 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:

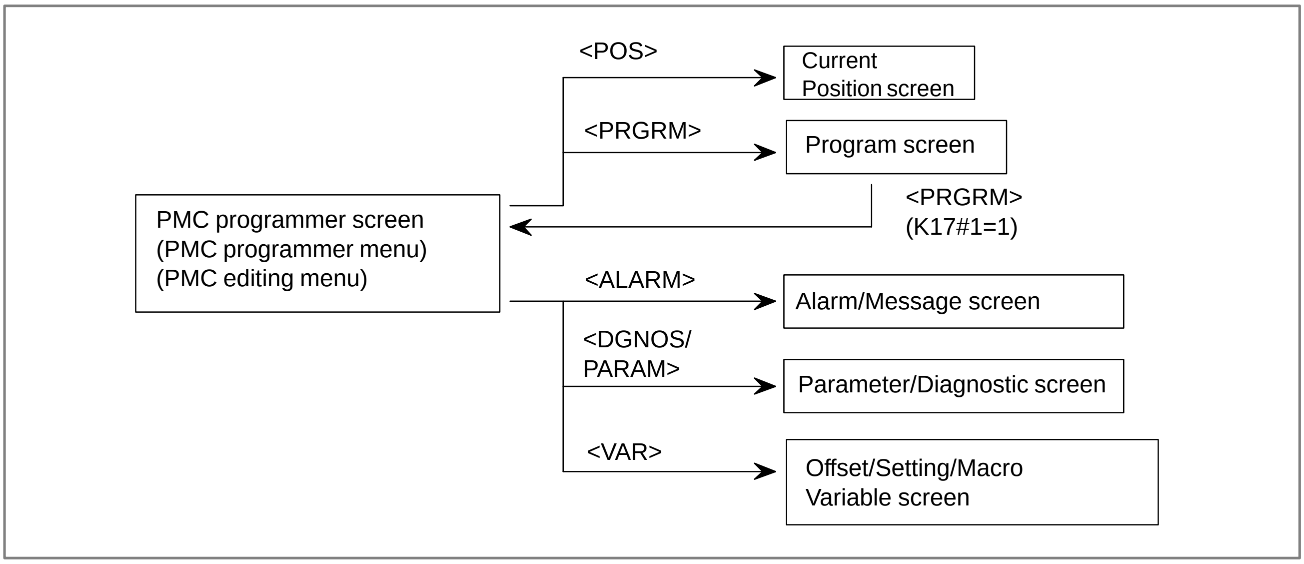


11.1 SELECTING THE PMC PROGRAMMER MENU

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.

CAUTION

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 and Power Mate *i*–D/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

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

CAUTION

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.

NOTE

In editing the functional instruction TMR, TMRB, CTR, DIFU, DIFD, the range and the multiple use of parameters are checked.
When the parameter is out of range error, it can not be inputted. When parameter is multiple use error, the error message is displayed on the screen.
(Example) Error message of multiple use.

```
N00010  TMR  (DUP)
   P001              50
```

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 When K19#0 of the keep relay area for PMC parameter is set to 1, the screen for storing the sequence program into flash EEPROM is displayed after ladder editing. (Only for Power Mate–H, Power Mate *i*–D/H)

```
DEVICE=F–ROM
>WRITE   Y/N [YES]
```

Please refer to the item 11.4 and storing the sequence program into flash EEPROM.

4 When parameter is multiple use error, the error message is displayed on the screen.
(Example) Error message

```
FUNC.    PARAM   NO.
DUPLICATE
```

Pressing the <↑> or <↓> cursor key displays the ladder mnemonics editing screen. Pressing <CAN> or <WRITE> key again displays the PMC programmer menu.

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.
9	FUNC, PARAM NO. OUT OF RANGE	There is out of range error in the parameter of functional instruction TMR, TMRB, CTR, DIFU, DIFD. It is displayed when mnemonics editing is finished.
10	FUNC. PARAM NO. DUPLICATE	There is multiple use error in the parameter of functional instruction TMR, TMRB, CTR, DIFU, DIFD. It is displayed when mnemonics editing is finished.
11	Nxxxxx yyyy (RNG) P0001 nnn	There is out of range error in the parameter of functional instruction TMR, TMRB, CTR, DIFU, DIFD. xxxxx : Step number yyyy : Functional instruction nnn : Parameter
12	Nxxxxx yyyy (DUP) P0001 nnn	There is multiple use error in the parameter of functional instruction TMR, TMRB, CTR, DIFU, DIFD. xxxxx : Step number yyyy : Functional instruction nnn : Parameter

11.7 STORING THE SEQUENCE PROGRAM INTO FLASH EEPROM (I/O) (ONLY FOR THE Power Mate–H AND Power Mate *i*–D/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 Flash ROM using the DPL/MDI (only for the Power Mate–H and Power Mate *i*–D/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 and Power Mate *i*–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.

> **CAUTION**
> 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.

> **CAUTION**
> 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.

R9044

0	
1	ER01 PROGRAM DATA ERROR (RAM)
2	
3	ER03 PROGRAM SIZE ERROR (OPTION)
4	ER04 LADDER OBJECT TYPE ERROR
5	
6	
7	ER07 NO OPTION (LADDER STEP)

R9046

0	ER16 RAM CHECK ERROR (PROGRAM RAM)
1	ER17 PROGRAM PARITY
2	ER18 PROGRAM DATA ERROR BY I/O
3	ER19 LADDER DATA ERROR
4	ER20 SYMBOL/COMMENT DATA ERROR
5	ER21 MESSAGE DATA ERROR
6	ER22 PROGRAM NOTHING
7	ER23 PLEASE TURN OFF POWER

IV. PMC–NB6 MANIPULATION

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) Ladder diagram display and editing (PMCLAD)

(2) Displaying PMC input/output signals and internal relay (PMCDGN)

   (a) Title data screen
   (b) Status screen
   (c) Alarm screen
   (d) Trace screen
   (e) I/O Link monitor screen

(3) PMC data setting and display (PMCPRM)

   (f) Timer
   (g) Counter
   (h) Keep relay
   (i) Data table

(4) Specifying PMC setting data (SETING)

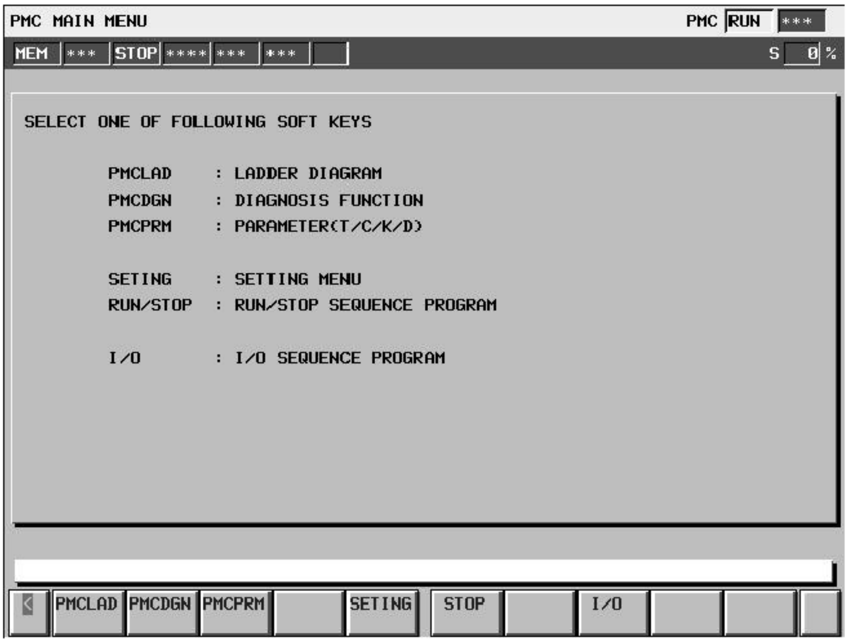
   (j) General setting data
   (k) Setting data related to editing and debugging
   (l) Online monitor parameter

(5) Writing, reading, and collating sequence programs and PMC parameters (I/O)

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.

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

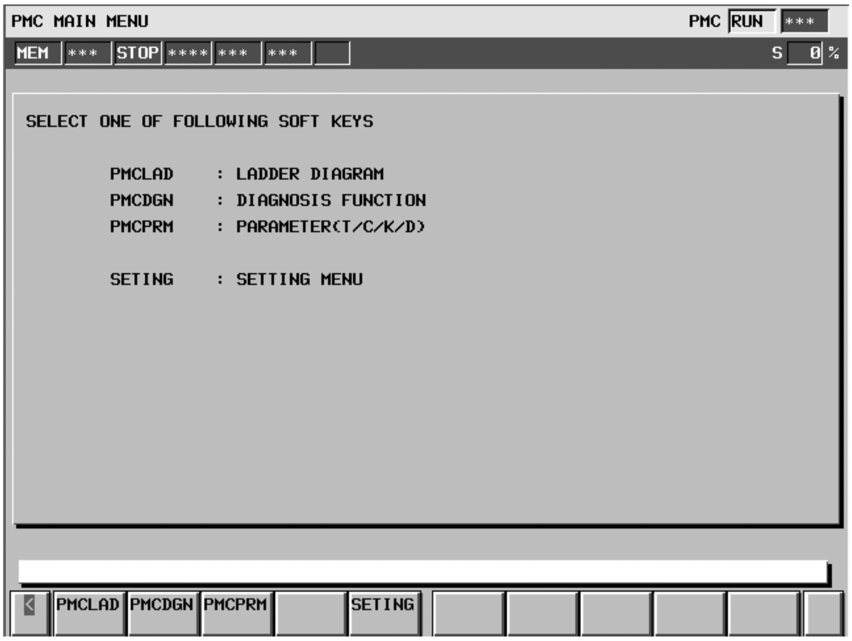
This area displays the data that was typed in.

(4) Soft key

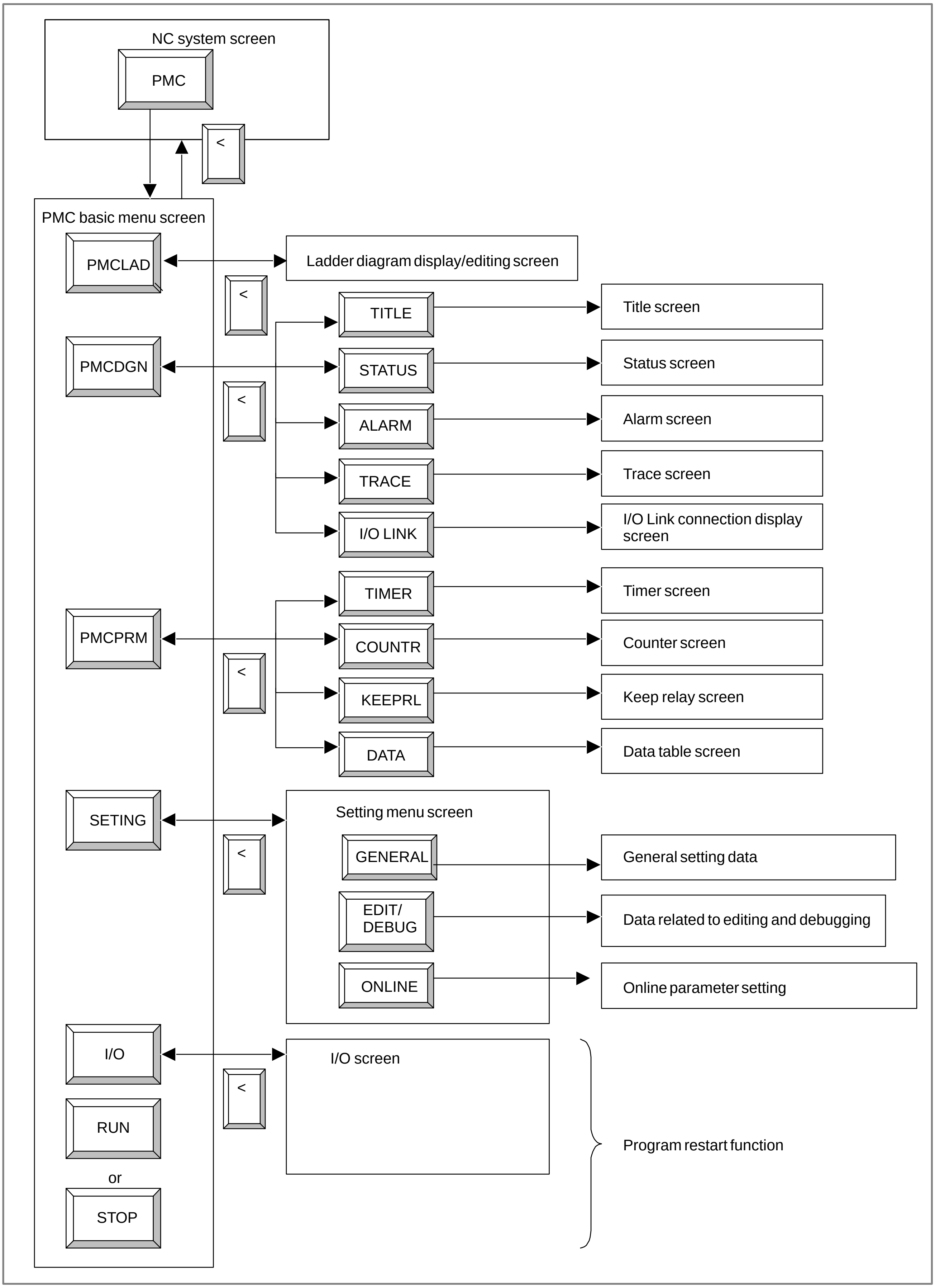
The soft key field consists of a soft key at both ends and ten soft keys in between. The left–end key has the following meaning:

 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:


PMC MAIN MENU
PMC RUN ***
MEM *** STOP **** *** ***
S 0 %
SELECT ONE OF FOLLOWING SOFT KEYS
PMCLAD : LADDER DIAGRAM
PMCDGN : DIAGNOSIS FUNCTION
PMCPRM : PARAMETER(T/C/K/D)
SETING : SETTING MENU
PMCLAD
PMCDGN
PMCPRM
SETING

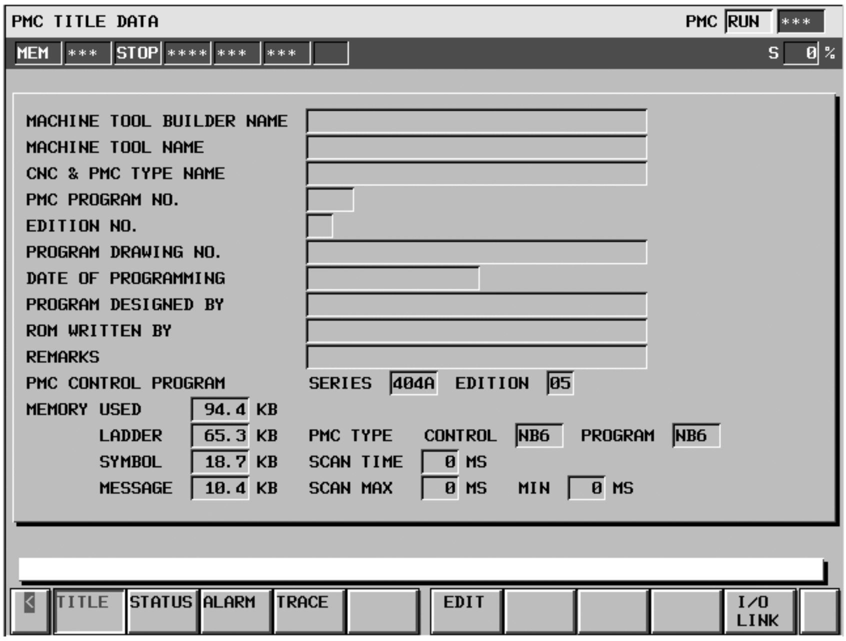
2.2 PMC SCREEN TRANSITION AND RELATED SOFT KEYS


NC system screen
PMC
<
PMC basic menu screen
PMCLAD
Ladder diagram display/editing screen
<
TITLE
Title screen
PMCDGN
STATUS
Status screen
<
ALARM
Alarm screen
TRACE
Trace screen
I/O LINK
I/O Link connection display screen
TIMER
Timer screen
PMCPRM
COUNTR
Counter screen
<
KEEPRL
Keep relay screen
DATA
Data table screen
SETING
Setting menu screen
<
GENERAL
General setting data
EDIT/ DEBUG
Data related to editing and debugging
ONLINE
Online parameter setting
I/O
I/O screen
<
RUN
or
STOP
Program restart function

3 DISPLAYING PMC INPUT/OUTPUT SIGNALS AND INTERNAL RELAY (PMCDGN)

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

When the [EDIT] soft key is pressed, the Title Editor Screen appears.

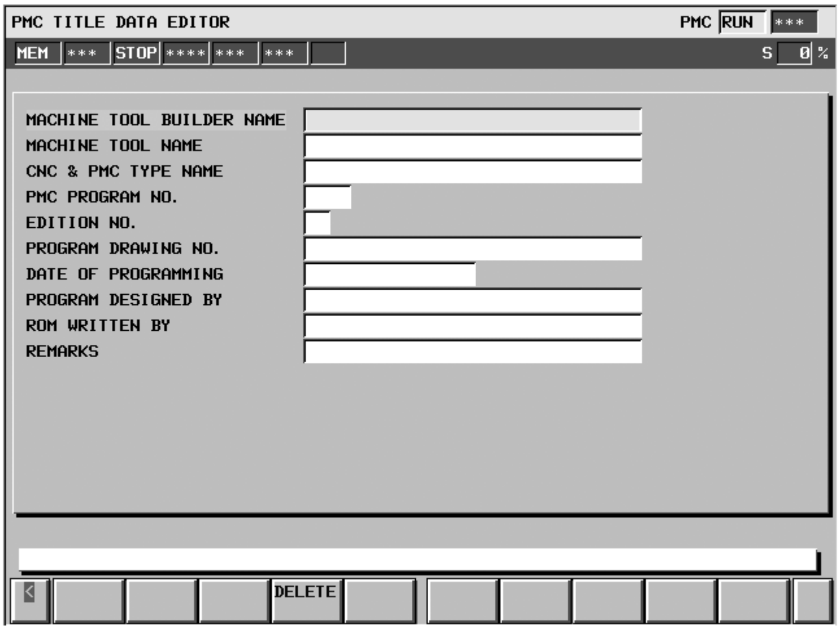
NOTE

1 When "PROGRAMMER ENABLE" is set to "YES" on the GENERAL screen of Setting screen, the [EDIT] soft key appears on the Title Data Display screen.

2 If the [EDIT] soft key is pressed while Online function is being executed, the "ONLINE MONITOR IS ACTIVE" message is displayed and the Title Editor screen does not appear.

3.2 TITLE DATA EDITOR SCREEN

When the [EDIT] soft key is pressed on the Title Data Display screen, the Title Data Editor screen appears.



On this screen, you can edit the following ten 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)

Move the cursor to the item and input the comment. If you initialize the item on the cursor, press the [DELETE] soft key.

If you do not save the edited data into Flash ROM, edited data is cleared by power OFF.

When finishing the edit, the following message is displayed to confirm to write to Flash ROM.

DO YOU WANT TO WRITE PROGRAM INTO FLASH ROM?

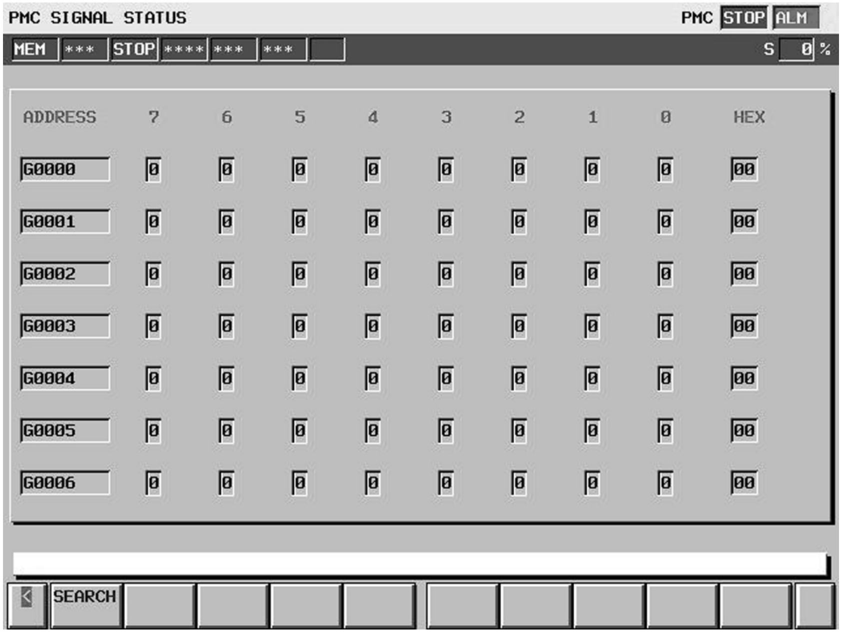
The following soft key is displayed. ”YES” soft key saves the data. ”NO” soft key does not save the data.

Pressing these soft keys causes a return to Title Data Display screen.

> **NOTE**
> When ”WRITE TO FROM (EDIT)” is set to ”YES” on the EDIT/DEBUG screen of Setting screen, saving operation into Flash ROM is enabled.

3.3 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.

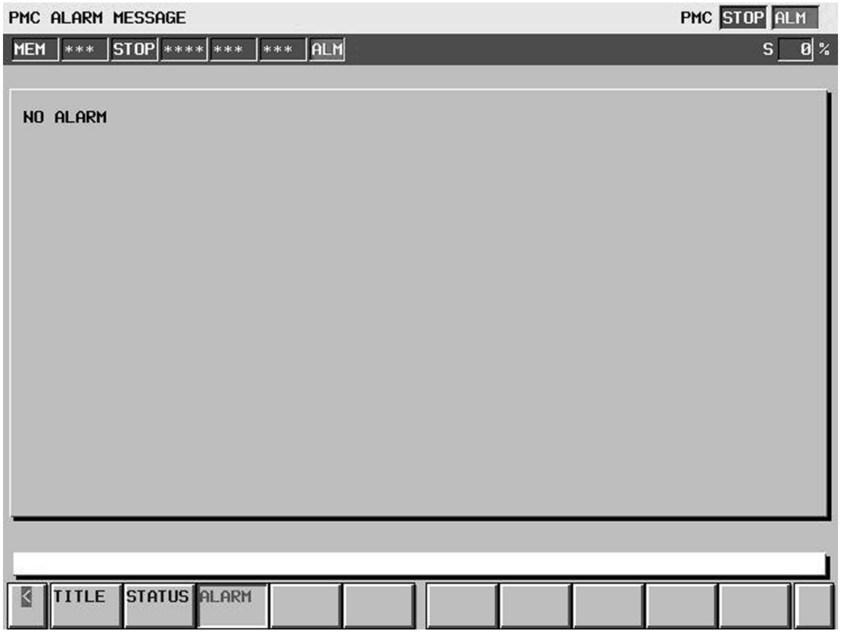
(3) A sequence of data starting at the specified address is displayed as a bit pattern.

(4) To specify another address for display, click a cursor key, page key, or the [SEARCH] soft key.

3.4 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.

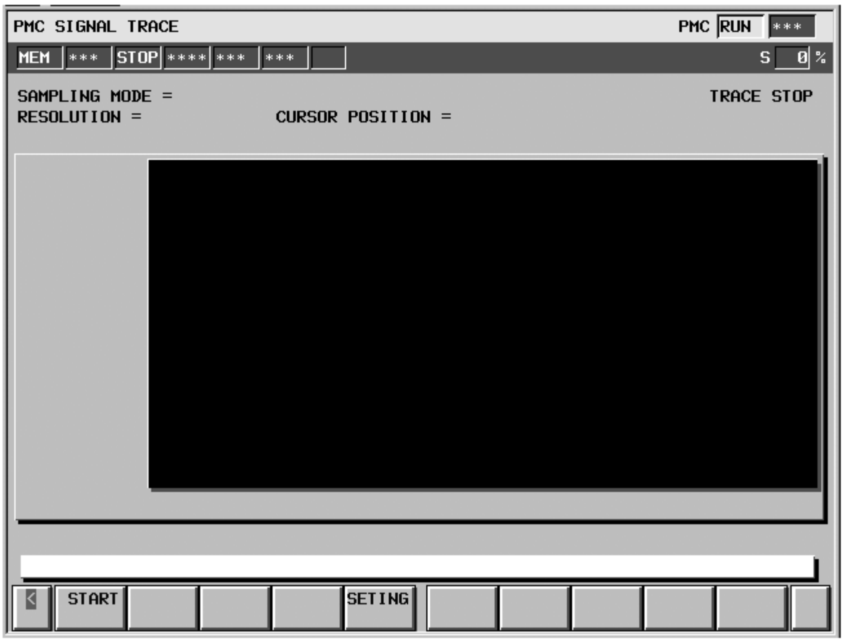


For an explanation of the alarm messages displayed on this screen, see APPENDIX M, “ALARM MESSAGE LIST.”

3.5 TRACE SCREEN

3.5.1 Overview

On the trace screen, you can trace specified signals. The result of the trace is displayed as the time chart of signals.
There are two tracing modes. One is the ”Time Cycle” mode that samples the state of the signals at every specified cycle time. The other is the ”Signal Transition” mode that samples the status of the signals when the signals that are watched at every specified time are changed.
Pushing [TRACE] soft key displays the following screen.

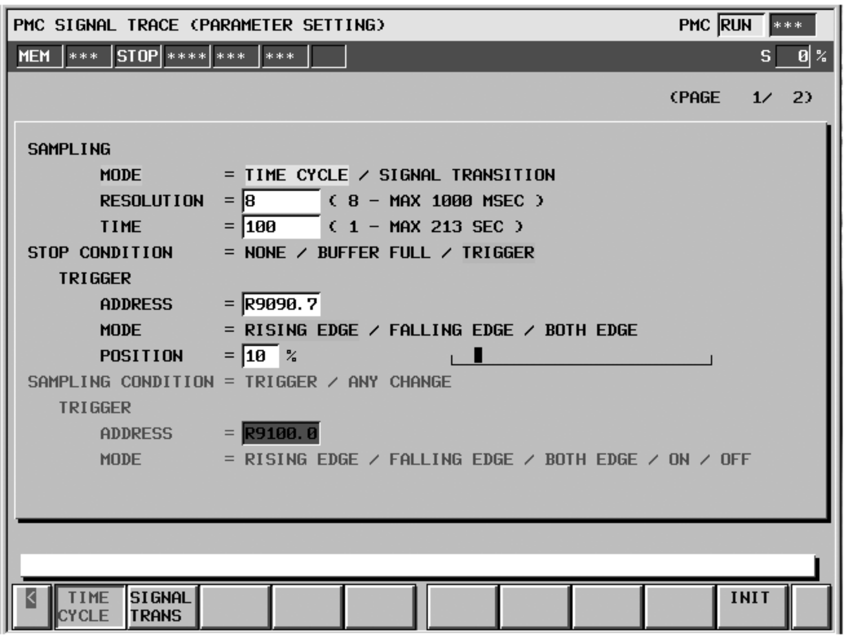


To execute tracing, setting of the trace parameter is necessary. Pushing the [SETING] soft key displays ”Parameter Setting” screen. The trace function is able to run automatically by the setting on ”PMC Setting” screen when the power is switched on. In this case, setting of the trace parameter is necessary in advance.

NOTE
As for the setting of automatic running for the trace function, please refer to ”II 7.5.1 Screen for Displaying General Settings (GENERAL)”

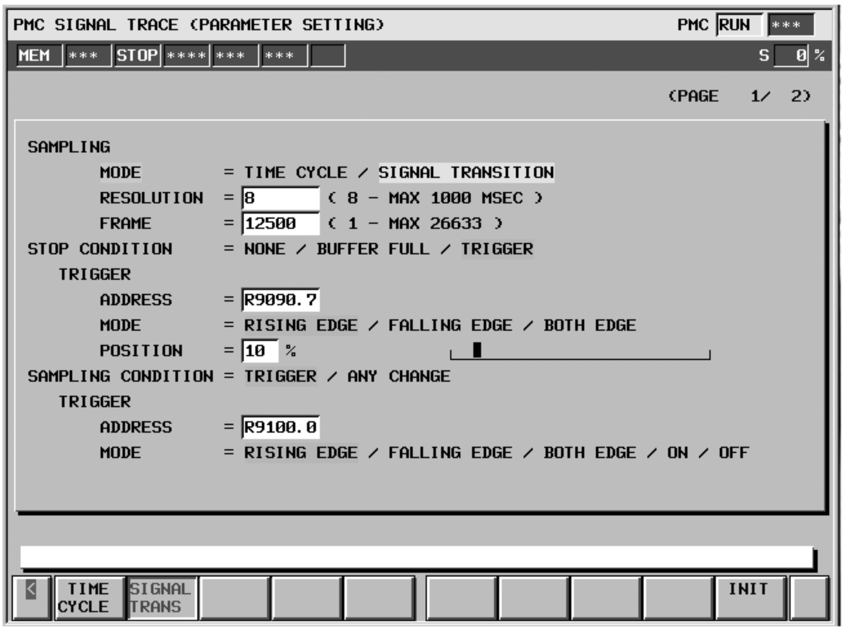
3.5.2 Setting of Trace Parameter

Pushing the [SETING] soft key displays the ”Parameter Setting” screen. The following is the screen example of the trace execution by ”TIME CYCLE” mode.



”Parameter Setting” has two screens. The page number is displayed on up right of the screen. The page key changes these screens.
Pushing [<] soft key displays ”PMC Signal Trace” screen.

The following is the screen example of the trace execution by ”SIGNAL TRANSITION” mode.



- SAMPLING MODE
  Determines the sampling mode. Select one by cursor key or soft key.

  Soft keys display when the cursor is put on ”SAMPLING MODE”.

Explanation:

TIME CYCLE	Samples at every specified cycle time.
SIGNAL TRANSITION	Samples when the signal changes.
INIT	Initializes all the settings. (This soft key is always displayed in page 1.)

- SAMPLING RESOLUTION
  The resolution of sampling is inputted. The default value is the minimum resolution (8msec). The range of the value is from 8msec to 1000msec. Inputted value is rounded down to the multiple of 8msec.
- SAMPLING TIME
  This parameter is displayed when ”TIME CYCLE” is set on ”SAMPLING MODE”. The execution time of trace is inputted. The value of ”SAMPLING SESOLUTION” or the number of specified signal address changes the range of the value that is able to input. The range is displayed on the right of the edit box.
- SAMPLING FRAME
  This parameter is displayed when ”SIGNAL TRANSITION” is set on ”SAMPLING MODE”. The number of sampling is inputted. The value of ”SAMPLING SESOLUTION” or the number of specified signal addresses changes the range of the value that is able to input. The range is displayed on the right of the edit box.
- STOP CONDITION
  Determines the condition to stop the trace. Select one by cursor key or soft key.

  Soft keys display when the cursor is put on ”STOP CONDITION”.

Explanation:

NONE	Does not stop the tracing automatically.
BUFFER FULL	Stops the tracing when the buffer becomes full.
TRIGGER	Stops the tracing by trigger

- STOP CONDITION TRIGGER ADDRESS
  When ”TRIGGER” is set on ”STOP CONDITION”, this parameter is enabled. Input signal address or symbol name as stop trigger.
  Only bit address or corresponding symbols can be inputted. Byte address cannot be inputted.

  Soft keys display when the cursor is put on ”STOP CONDITION TRIGGER ADDRESS”.

Explanation:

DELETE Clears the value in the edit box.

SYMBOL Changes the address display to the symbol name display and changes the soft key to [ADDRESS]. After that, following soft keys are displayed.

ADDRESS Changes the symbol name display to the address display and changes the soft key to [SYMBOL].

- STOP CONDITION TRIGGER MODE
  When ”TRIGGER” is set on ”STOP CONDITION”, this parameter is enabled. Input trigger mode when the trace is stopped. Select one by cursor key or soft key.

Soft keys display when the cursor is put on ”STOP CONDITION TRIGGER MODE”.

Explanation:

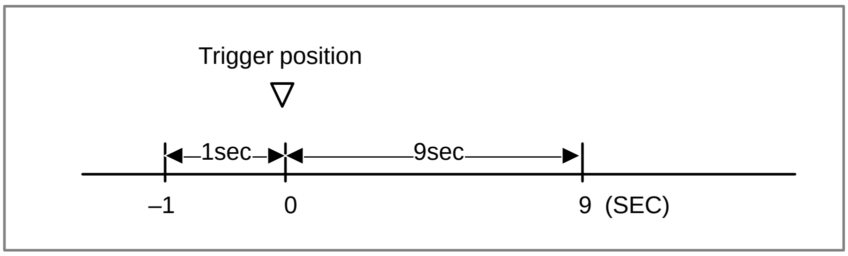
RISING EDGE Stops the tracing automatically by rising up of the trigger signal.

FALLING EDGE Stops the tracing automatically by falling down of the trigger signal.

BOTH EDGE Stops the tracing automatically by rising up or falling down of the trigger signal.

- STOP CONDITION TRIGGER POSITION
  When ”TRIGGER” is set on ”STOP CONDITION”, this parameter is enabled. Input the ratio of the sampling time or number which specifies the position where specified trigger condition is on.
  If you would like to examine the transitions of the signal before the trigger condition, you should set a big value in this parameter. If you would like to examine the transitions of the signal after the trigger condition, you should set a small value in this parameter.

Example: The case that sampling time is 10 second and trigger position is set as ”10%”.



The graph is displayed on the right of the edit box. The edge of the left hand is as 0% and the edge of the right hand is as 100%. The position indicated by the input value is displayed as a gauge.

- SAMPLING CONDITION
  When ”SIGNAL TRANSITION” is set on ”TRACE MODE”, this parameter is enabled. Select one by cursor key or soft key.

Soft keys display when the cursor is put on ”SAMPLING CONDITION”.

Explanation:

TRIGGER	Samples the status of specified signals when the specified sampling condition is on.
ANY CHANGE	Samples the status of specified signals when the signals change

The address of the signals that should be sampled is set on Parameter Setting screen page2.

- SAMPLING CONDITION TRIGGER ADDRESS
  When ”SIGNAL TRANSITION” is set on ”TRACE MODE”, and ”TRIGGER” is set on ”SAMPLING CONDITION”, this parameter is enabled. Input signal address or symbol name as sampling trigger.
  Only bit address or corresponding symbol can be inputted. Byte address cannot be inputted.

Soft keys display when the cursor is put on ”SAMPLING CONDITION TRIGGER ADDRESS”.

The contents of the soft keys are same as ”STOP CONDITION TRIGGER ADDRESS”.

- SAMPLING CONDITION TRIGGER MODE
  When ”SIGNAL TRANSITION” is set on ”TRACE MODE”, and ”TRIGGER” is set on ”SAMPLING CONDITION”, this parameter is enabled. Input trigger mode that determines the condition of specified trigger.

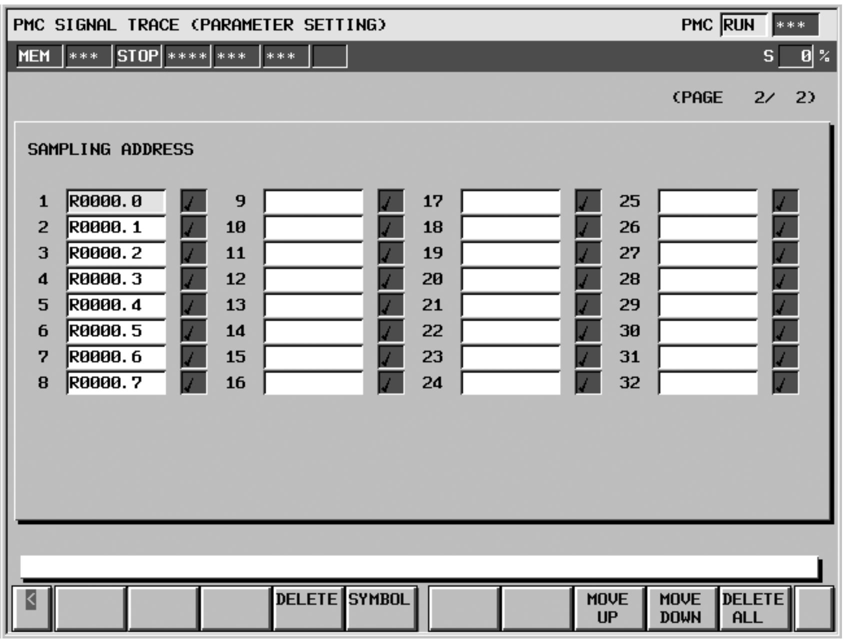
Soft keys display when the cursor is put on ”SAMPLING CONDITION TRIGGER MODE”.

Explanation:

RISING EDGE	Samples the status of specified signals by rising up of the trigger signal.
FALLING EDGE	Samples the status of specified signals by falling down of the trigger signal.
BOTH EDGE	Samples the status of specified signals by rising up or falling down of the trigger signal.
ON	Samples the status of specified signals during the trigger signal is on.
OFF	Samples the status of specified signals during the trigger signal is off.

- SAMPLING ADDRESS

  In page 2 of Parameter Setting screen, you can set the addresses or symbols that should be sampled.



Move the cursor into edit box and input PMC signal address or symbol.
In case of inputting discrete bit addresses, any bit address can be inputted. Moreover, when you input byte address, all bits of the address (bit0–bit7) are set automatically.
Maximum 32 points of signal address can be inputted. Increasing the number of the signal address changes the capacity of ”SAMPLING TIME” or ”SAMPLING FRAME” in page1. If the capacity is changed, the warning message is displayed.

Example of warning message:

In case of ”TIME CYCLE” mode
”SAMPLING TIME IS REDUCED TO *n* SEC.”

In case of ”SIGNAL TRANSITION” mode
”SAMPLING FRAME IS REDUCED TO *n* COUNT.”

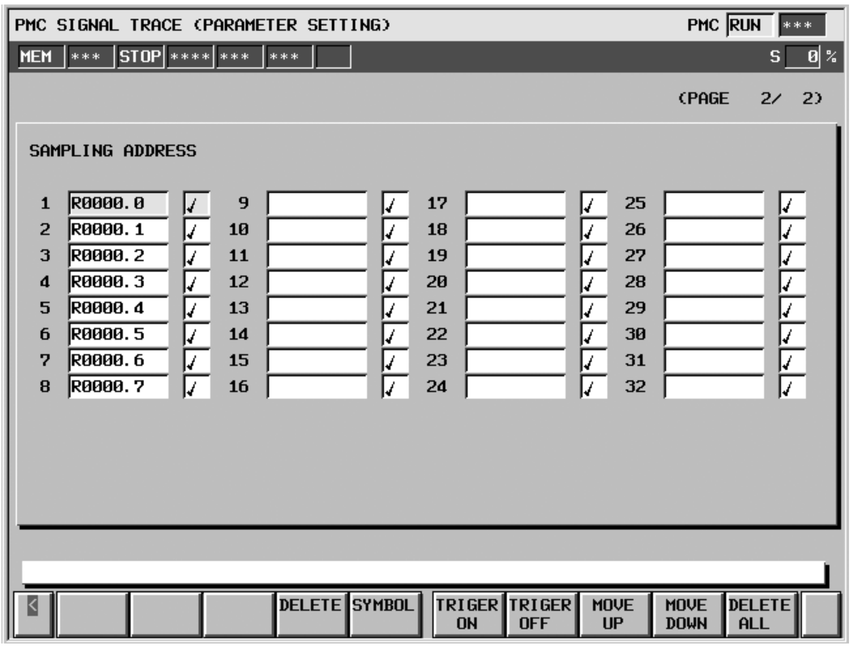
The ”*n*” means the maximum value that is able to input.

Explanation of the soft keys:

DELETE	Clears the value of the edit box on the cursor.
SYMBOL	Changes the address display to the symbol display. However, display of the address that is not defined the symbol does not change. This soft key also changes to ”ADDRESS”. The following soft keys are displayed.
MOVE UP	Exchanges the signal indicated the cursor for the signal above one line.
MOVE DOWN	Exchanges the signal indicated the cursor for the signal below one line.
DELETE ALL	Clears all of the value of the edit box.

ADDRESS — Changes the symbol display into the address display and changes the soft key to ”SYMBOL”.

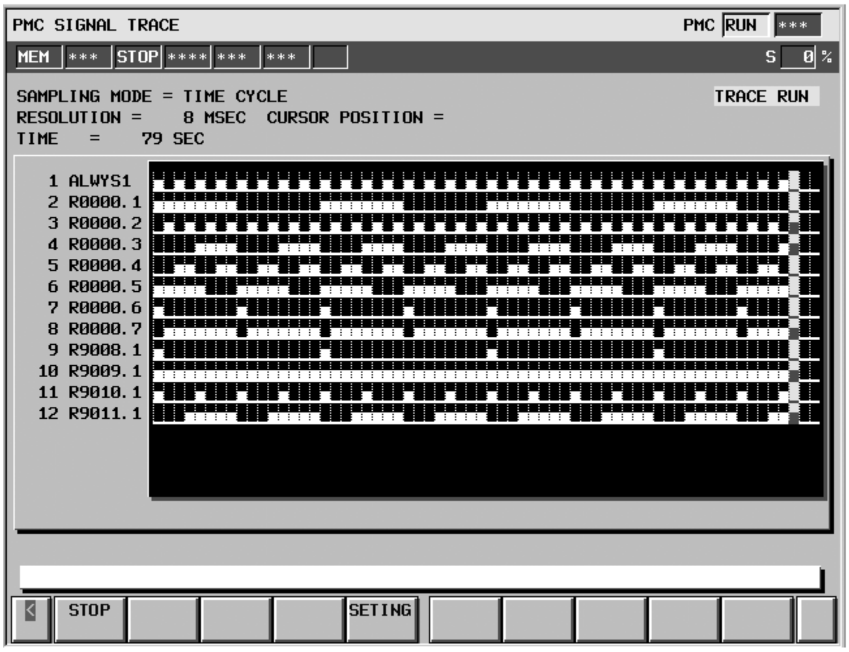
- TRIGGER SETTING OF THE SAMPLING SIGNALS
  When ”SIGNAL TRANSITION” is set on ”TRACE MODE” and ”ANY CHANGE” is set on ”SAMPLING CONDITION”, the check boxes on the right of the sampling address or symbols are displayed as follows. Check the signals that should trigger the sampling in the setting signals.



Pushing [TRIGGER ON] soft key sets the trigger on. Pushing [TRIGGER OFF] soft key sets the trigger off. The default setting is trigger on for all signals. The contents of other soft keys are same as ”SAMPLING ADDRESS”.

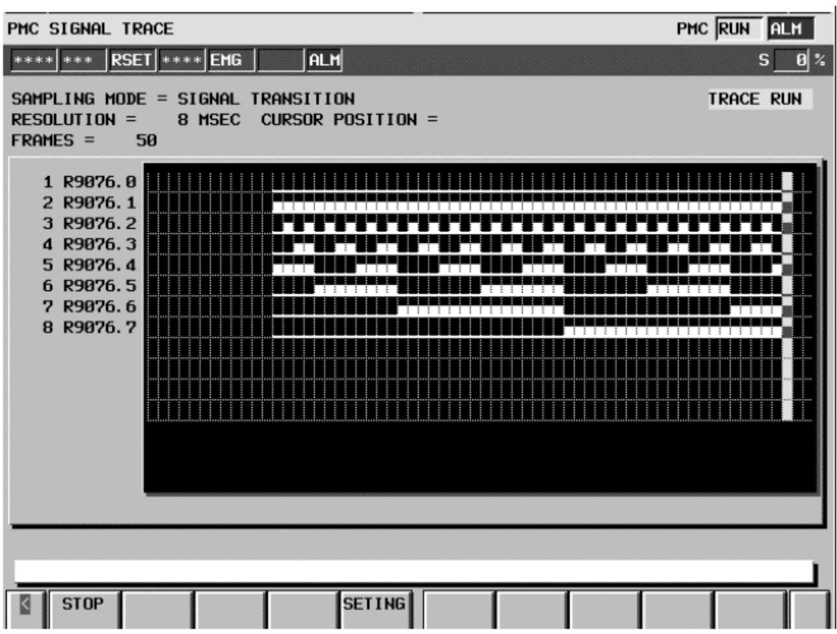
3.5.3 Execution of Trace

On trace screen, pushing [START] soft key starts the execution of trace after you set the trace parameter correctly. The following is the screen example of the trace execution by ”TIME CYCLE” mode.



The result of trace is immediately displayed during execution of the trace. When the stop condition that is set in parameter setting screen is satisfied the execution is finished. Pushing [STOP] soft key aborts the execution.

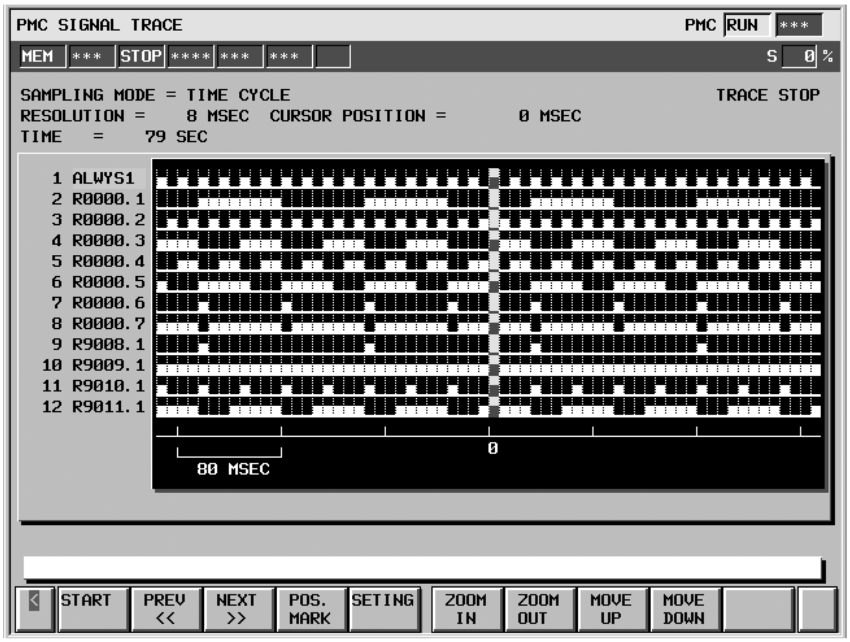
The following is the screen example of the trace execution by ”SIGNAL TRANSITION” mode.



In ”SIGNAL TRANSITION” mode, graphic display is not refreshed until any signal for sampling trigger changes.

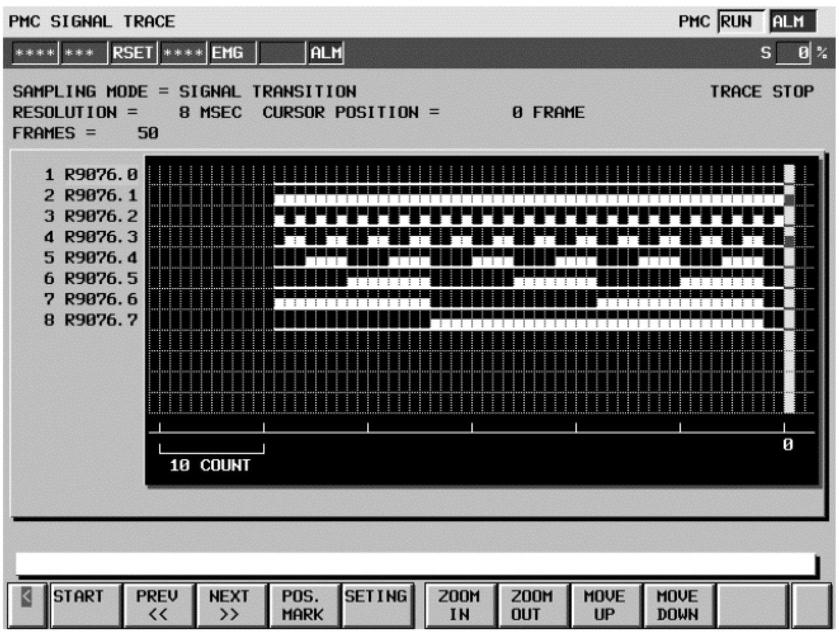
3.5.4 Operation after Execution of Trace

When the execution is finished, the result of trace is displayed. The following is the screen example of trace by ”TIME CYCLE” mode.



The cursor indicating current position is initially displayed on the original point (0 point).
The position of the cursor is displayed in ”CURSOR POSITION” in the upper of the screen. The cursor can move horizontally.

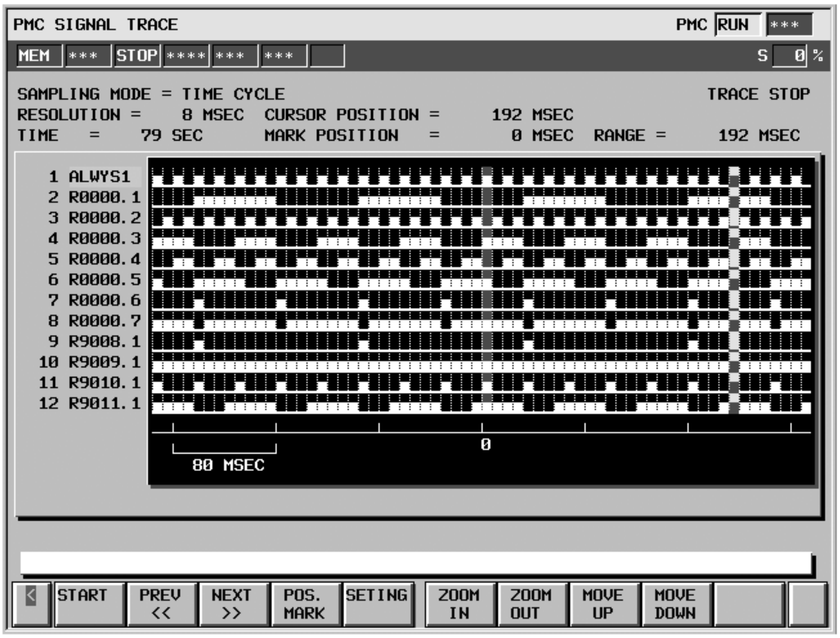
The following is the screen example of trace by ”SIGNAL TRANSITION” mode.



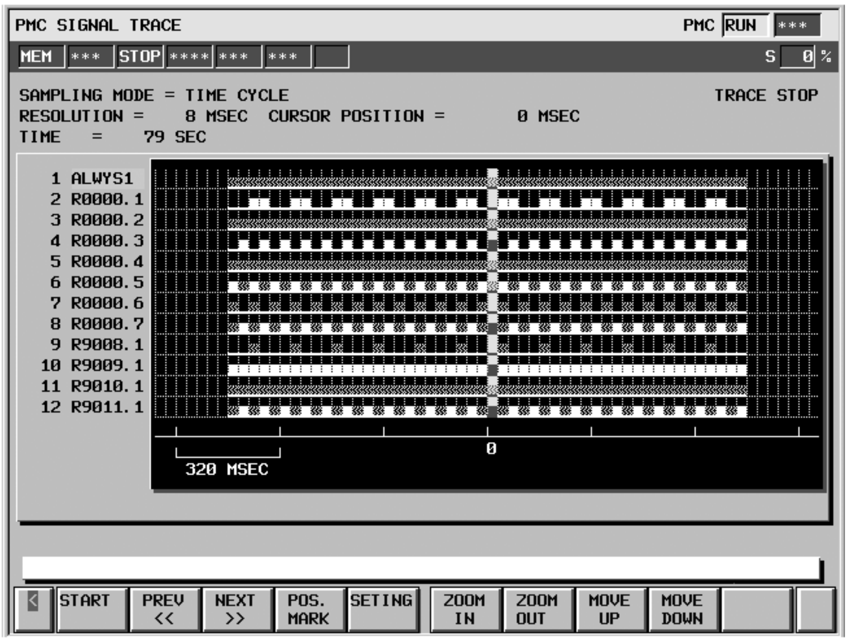
After the execution, following operation is enabled.

(1) SCROLL OF SCREEN
Using cursor up/down key and page up/down key enables the vertical scroll for the specified signal. Using cursor right/left key, [NEXT>>] soft key and [PREV<<] soft key enables the horizontal scroll of the graph.

(2) AUTOMATIC CALCULATION OF THE SELECTED RANGE
Pushing [POS MARK] soft key marks the current position and displays the mark cursor. If the mark cursor duplicates with the current position cursor, the current position cursor has priority of display. The ”MARK POSITION” that shows the position of the mark cursor and ”SELECT RANGE” that shows the range between the mark cursor and the current position cursor are displayed in the upper of screen. Moving the current position cursor changes these values. Pushing [POS MARK] again releases the select range mode.



(3) ZOOM IN/ZOOM OUT OF WAVEFORM
Pushing [ZOOM IN] soft key magnifies the display of chart. Pushing [ZOOM OUT] soft key reduces the display of chart. Pushing these soft keys also change the scale value of the graduation on the graph. When trace is just finished, the default zooming level was the most magnified level. In [ZOOM OUT] mode, gray box is displayed as following screen example when the transitions of signal cannot be expressed accurately enough. The limitation of [ZOOM OUT] displays all of result of the trace in one page.



(4) EXCHANGE OF SAMPLING SIGNAL

Pushing [MOVE UP] soft key exchanges the signal indicated by the signal cursor for the signal one line above. Pushing [MOVE DOWN] soft key exchanges the signal indicated by the signal cursor for the signal one line below. The result of the operation is cancelled by the execution of trace or putting the power off. When you would like to preserve the order of displayed signals against the executing or powering off, please change the order on ”SAMPLING ADDRESS” screen.

3.5.5 Comparison With the Function of FS15–B (PMC–NB/NB2)

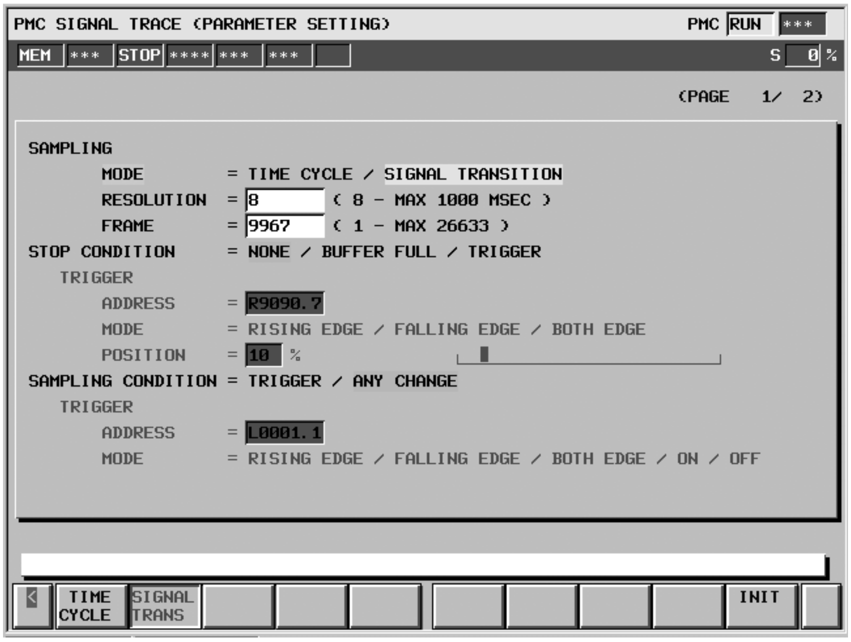
The trace function for PMC–NB6 is different from the trace function, [PMCDGN] [TRACE], or the analysis function, [PMCDGN] [ANALYS], for PMC–NB/NB2 in the specification of display and setting parameter.

However, you can get the same result of the trace as PMC–NB/NB2 by parameter setting.

The way of setting of the parameters to get the same result of the trace as PMC–NB/NB2 is described as follows.

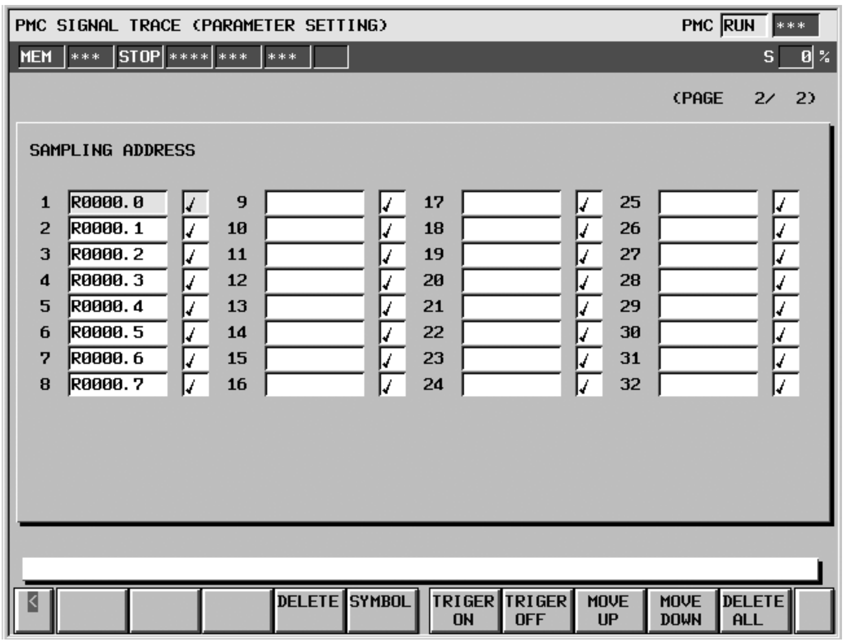
(1) Setting equivalent to the [TRACE] screen for PMC–NB/NB2

The parameters in page 1



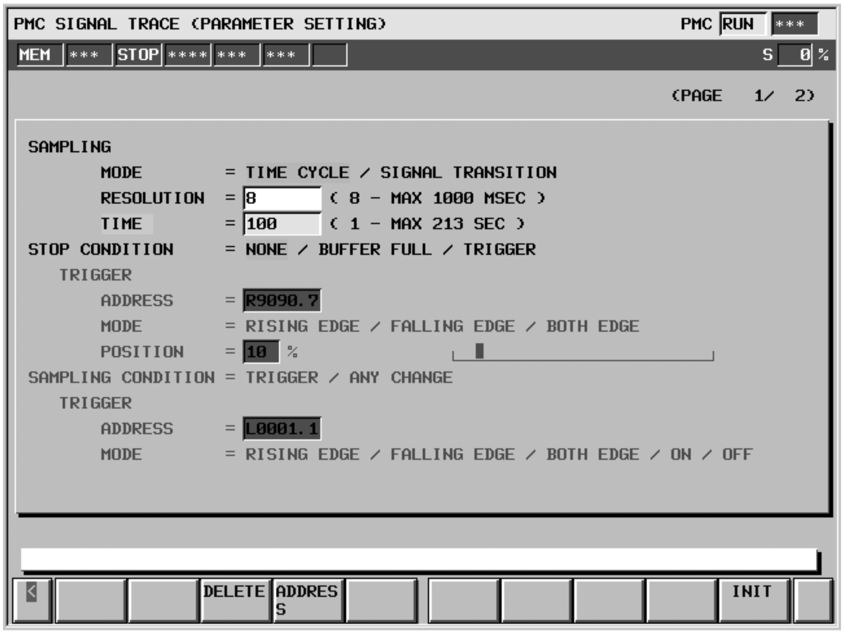
By setting ”SAMPLING MODE” = ”SIGNAL TRANSITION”, sampling equivalent to [TRACE] function of PMC–NB/NB2 is performed.
Furthermore, enhancements on PMC–NB6 are the ability to change the sampling cycle time and the capacity of sampling data by setting ”RESOLUTION” and ”FRAME” parameters. In addition, the result of sampling is displayed in easy to understand format of time chart.

The parameters in page 2

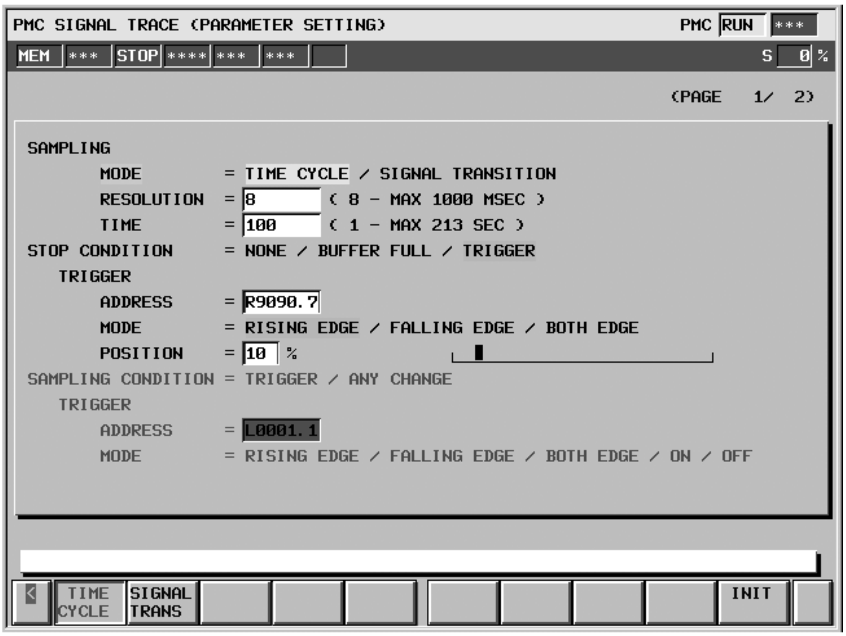


If you input signals from bit0 to bit7 into ”SAMPLING ADDRESS”, this setting is equivalent to input of the byte address for PMC–NB/NB2. The trigger setting of the sampling signals is equivalent to the set of mask data (”MASK DATA”) for PMC–NB/NB2.

(2) Setting equivalent to none trigger start of the [ANALYS] screen for PMC–NB/NB2 (CONDITION = 0 and TRIGGER MODE = 0)



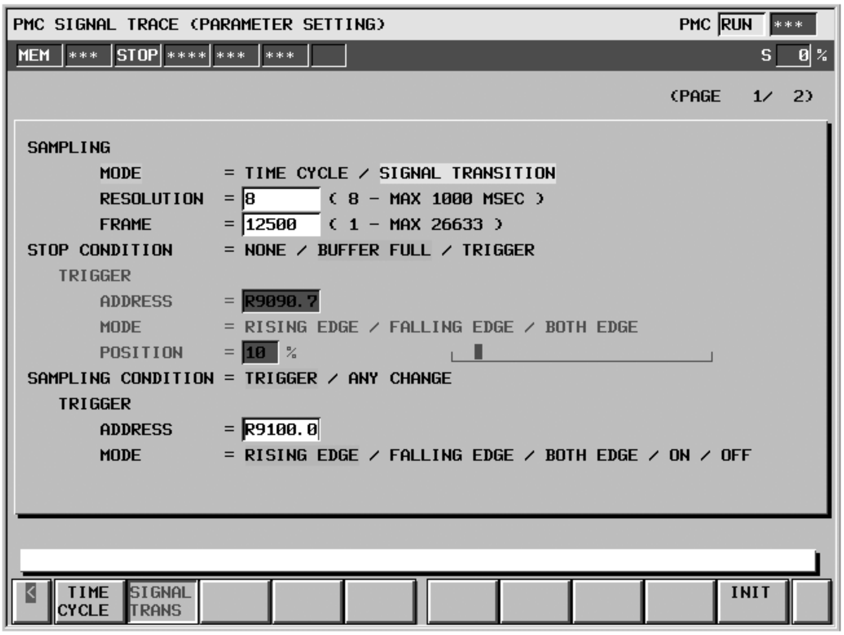
(3) Setting equivalent to ”TRIGGER MODE” = ”0:AFTER”, ”1:ABOUT”, ”2:BEFORE” of the [ANALYS] screen for PMC–NB/NB2



The value of ”TRIGGER POSITION” is equivalent to ”AFTER”, ”ABOUT”, ”BEFORE” for PMC–NB/NB2. The value ”0” means ”AFTER”. The value ”100” means ”BEFORE”. The value from ”1” to ”99” means ”ABOUT”.

The point of the enhancement is the ability to adjust the ratio of data before and after the trigger position.

(4) Setting equivalent to ”TRIGGER MODE” = ”3:ONLY” of the [ANALYS] screen for PMC–NB/NB2



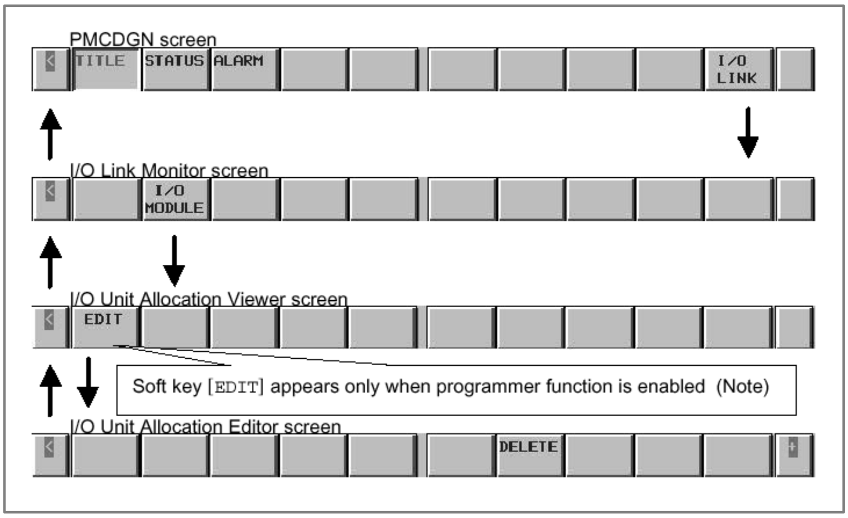
The example samples the signals specified in page 2 when the sampling condition specified in ”SAMPLING CONDITION” is on.

3.6 I/O LINK MONITOR AND ALLOCATION OF I/O UNIT

The I/O Link Monitor screen and the I/O Unit Allocation screens are for checking I/O Link connection and for allocating I/O Unit.

- I/O Link Monitor screen: I/O Units connected via I/O Link are displayed in order of Group number.
- I/O Unit Allocation viewer screen: I/O Units Allocation to X and Y addresses are displayed.
- I/O Unit Allocation Editor screen: I/O Units can be allocated to X and Y addresses.

Change of screens



NOTE
Soft key [EDIT] appears only when programmer function is enabled by setting ”PROGRAMMER ENABLE” to ”YES” at GENERAL screen of PMC Settings. While Online Monitor function is active, you can not reach I/O Unit Allocation Editor screen. To use I/O Unit Allocation Editor function, you have to disable Online Monitor function at PMC Setting screen by choosing ”NOT USE” for ”RS–232–C” and ”F–BUS” settings.

3.6.1 I/O Link Monitor screen

I/O Link Monitor screen shows the types and the ID codes of I/O Units that are connected to I/O Link in order of Group number.
You can confirm the connection of I/O Units type at this screen.
Call this screen by pressing of soft key [I/O LINK] in PMCDGN.

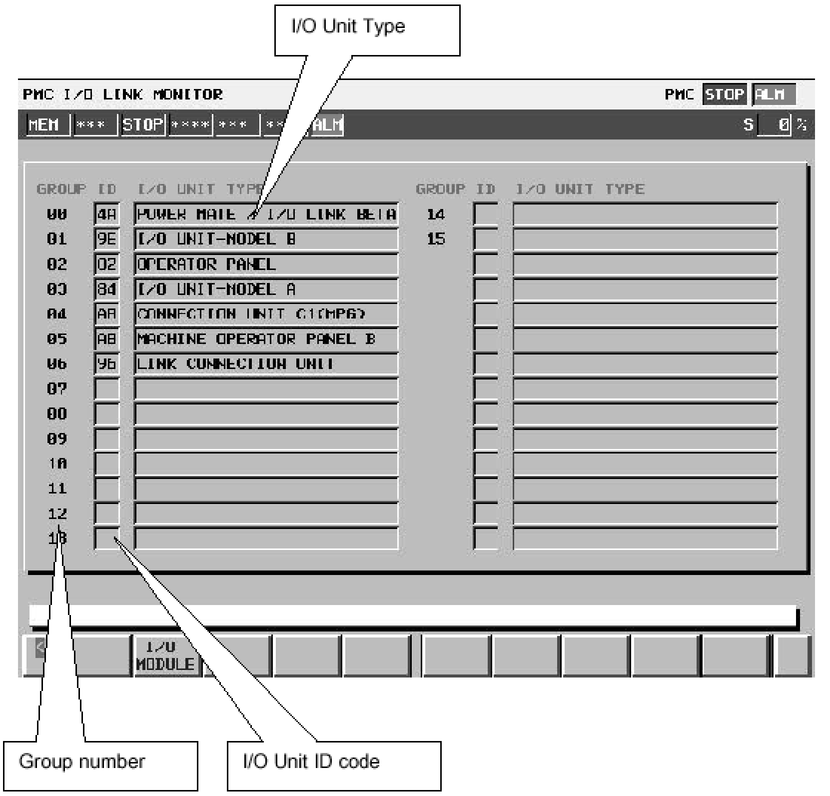


Table 3.6.1. Displayed type and true type of I/O Units

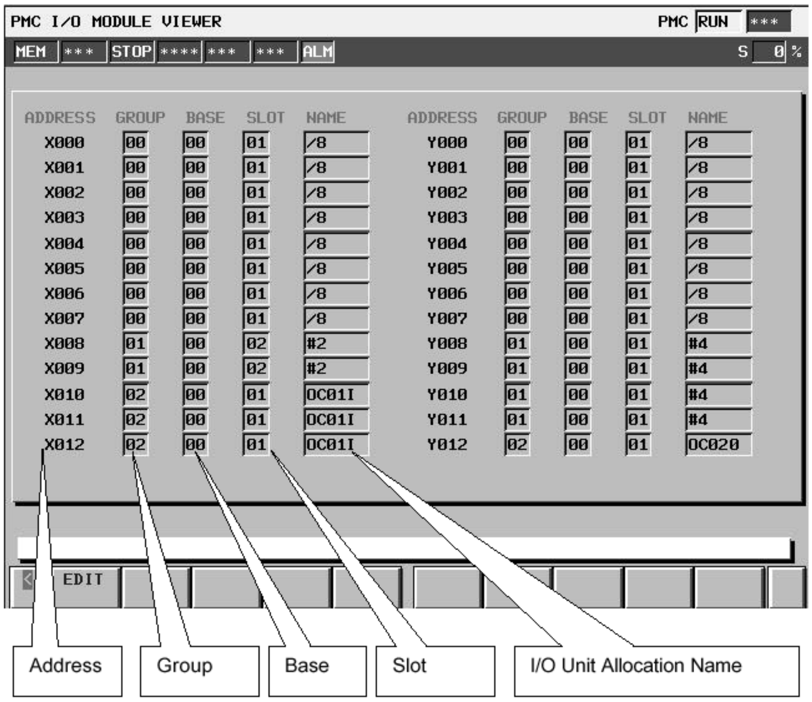
Displayed I/O Unit	ID	True I/O Unit
CONNECTION UNIT	80	Connection Unit
OPERATOR PANEL	82	Connection Unit for Operators Panel
I/O–B3	83	I/O B3
IO UNIT–MODEL B	84, 86, 87	I/O Unit–MODEL A
PLC SERIES 90–30	45	PLC SERIES 90–30
POWER MATE / IO LINK BETA	4A	Power Mate or I/O Link Beta
SERIES 0	50	Series 0
OPERATOR IF BOARD (MPG1)	53	Machine Operators Panel Inter-face
LINK CONNECTION UNIT	96	I/O Link Connecting Unit
I/O UNIT–MODEL B	9E	I/O Unit–MODEL B

Displayed I/O Unit	ID	True I/O Unit
R–J MATE	61	R–J Mate
CONNECTOR PANEL MODULE	A9	I/O module for connector panel
OPERATOR PANEL A1	AA	I/O module for operator's panel
OPERATOR I/F BOARD (MPG3)	6B	Operator Interface(with MPG)
LOADER I/O	AF	I/O Board for Loader
ROBOCUT DIF	B0	DIF Board for ROBOCUT
ROBOCUT MIF	B1	MIF Board for ROBOCUT
I/O CARD	B2	I/O board
ROBOSHOT I/O CARD A	B3	I/O for ROBOSHOT
LOADER I/O (MATRIX)	B4	I/O Board for Loader(Matrix)
PROCESS I/O FA	B5	Process I/O for R–J3
PROCESS IO	89	Process I/O for R–X
I/O LINK ADAPTER	8B	I/O Link adapter
ROBOT CONTROLLER	52	Controller for R–X
PLC SERIES 90	54	PLC SERIES 90
OPERATOR PANEL	95	I/O for Series 0
LASER OSCILLATOR	97	Laser Oscillator
FIXED I/O TYPE A	98	I/O for Robot Type A
FIXED I/O TYPE B	99	I/O for Robot Type B
AS–I CONVERTER	77	AS–i Converter
OPERATOER PANEL B	A8	I/O Module(for Operator Panel 48/32)
MACHINE OPERATOER PANEL A	A8	I/O Module(for Machine Operator Panel of 0 Type)
CONNECTION UNIT C1 (MPG)	A8	Connection Unit C1(with MPG)
MACHINE OPERATOER PANEL B	A8	I/O Module (for Machine Operator Panel)
I/O MODULE WITH LCD	A8	LCD display embedded I/O
UNKNOWN UNIT	–	Unsupported I/O Unit

CAUTION
I/O Units not for Series FS15*i* are written in Table 7.8.1.

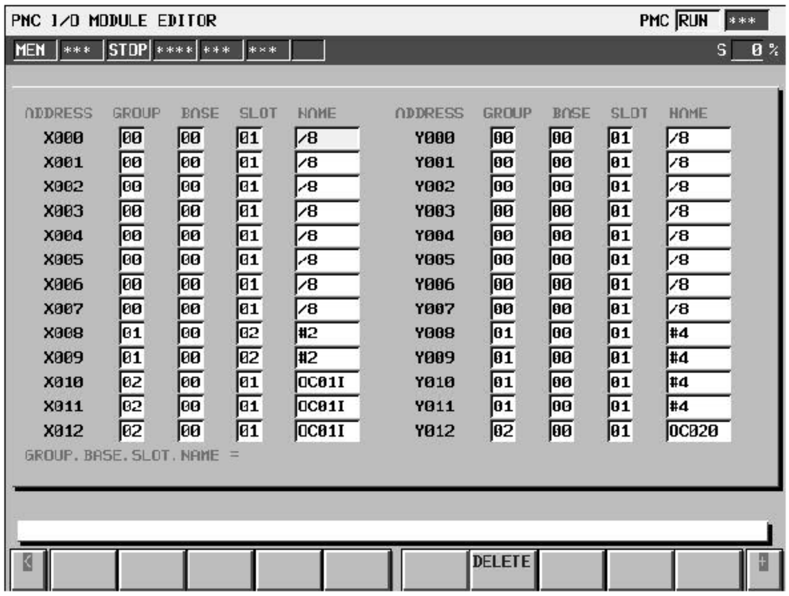
3.6.2 I/O Unit Allocation Viewer screen

I/O Link Allocation Viewer screen shows I/O Units allocation to X and Y addresses.
You can confirm the allocation of I/O Units at this screen.
Call this screen by pressing of soft key [I/O LINK] → [I/O MODULE] in PMCDGN.



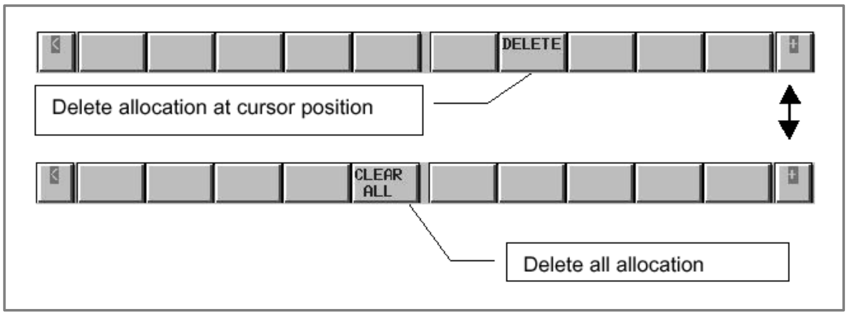
3.6.3 I/O Unit Allocation Editor screen

Allocate I/O Units to X and Y addresses at I/O Unit Allocation Editor screen.
Call this screen by pressing soft key [I/O LINK] → [I/O MODULE] → [EDIT] in PMCDGN.



3.6.3.1 Operations

Soft Keys in I/O Unit Allocation Editor screen

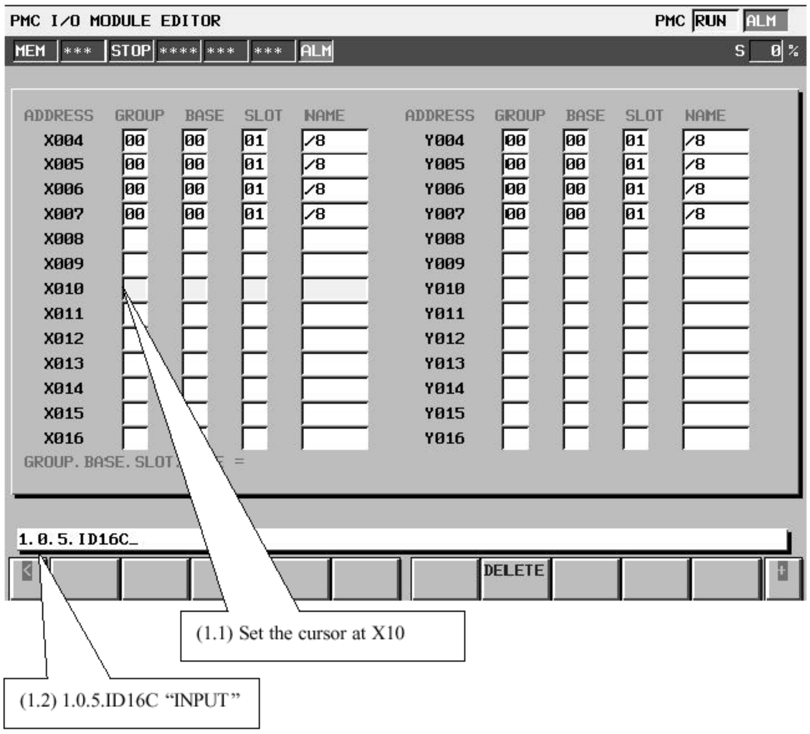


(1) Allocation I/O Units to X and Y addresses

(1.1)Set the cursor at address to which you will allocate new I/O Unit.

(1.2)Type ”Group. Base. Slot. Name” and press INPUT key.

Ex) In case you allocate ”Group=1, Base=0, Slot=5, Name=ID16A” to X10, set the cursor at X10 and <u>1.0.5. ID16C ”INPUT”</u>

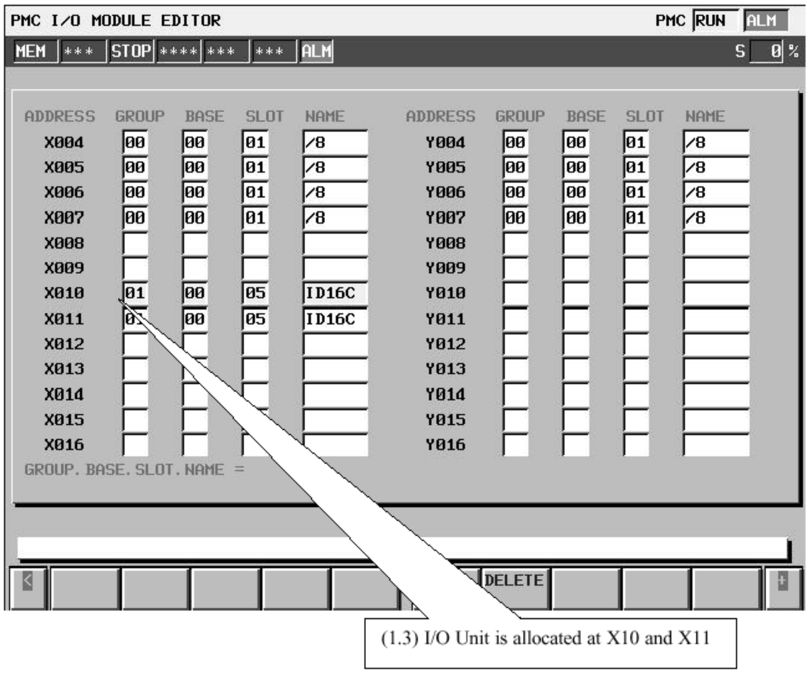


NOTE

As to the allocation name of I/O Unit, please refer to ”FANUC PMC Programming Manual (LADDER Language) Chapter I, 3.2 Table3.2.2”

(1.3)The I/O Unit is allocated to address of the cursor position for the I/O Unit size.

In case of above example (1.2), I/O Unit is allocated at X10 and X11 like as follows.



> **CAUTION**
> 1 Turn off and on the power after you store the Ladder to Flash ROM in order to make the allocation effective.
> 2 The Ladder program is not stopped automatically when you open I/O Unit Allocation Viewer screen or Editor screen.

(2) Delete allocation

(2.1)Set the cursor at allocation of which you will delete and press soft key [DELETE].

(2.2)The allocation is deleted.

(3) Delete all allocation

(3.1)Press the right end soft key.

(3.2)Press soft key [CLEAR ALL].

(3.3)”ARE YOU SURE TO DELETE ALL ALLOCATION DATA?” is displayed.

(3.4)Press soft key [YES].

(3.5)Allocation of X and Y are all deleted.

3.6.3.2 Error Message

Error message	Meaning and solution
GROUP NUMBER IS TOO LARGE	Meaning: Too large Group number is input. Solution: Input Group Number below 16.
BASE NUMBER IS TOO LARGE	Meaning: Too large Base number is input. Solution: I/O Unit–B (##,#1–#10): Input Base number 0 Other I/O Unit: Input 0 or 1.
SLOT NUMBER IS TOO LARGE	Meaning: Too large Slot number is input. Solution: I/O Unit–B: Input Slot number below 31 Other I/O Unit: Input Slot number below 11
SLOT NUMBER IS TOO SMALL	Meaning: Too small Slot number is input. Solution: I/O Unit–B: Input Slot number 0 or above Other I/O Unit: Input Slot number above 0
I/O UNIT NAME MISMATCH	Meaning: I/O Unit for X is allocated to Y address or I/O Unit for Y is allocated to X address. Solution: Confirm the I/O Unit Name and the address
ILLEGAL I/O UNIT NAME	Meaning: I/O Unit name is not correct. Solution: Input correct name described at Chapter I.3.2 Table 3.2.2.
NOT ENOUGH SPACE	Meaning: There is no enough space for the size of I/O Unit you will allocate. Meaning: This message is displayed also in case you will allocate to the address in which other I/O Unit has been allocated. Solution: Re–Input the allocation after you delete the allocation under the cursor and make space.

4 PMC DATA SETTING AND DISPLAY (PMCPRM)

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.

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.

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.

4.4 TIMER SCREEN (TIMER)

This screen is used to set and display the timer values for the machine instruction timers (SUB3).

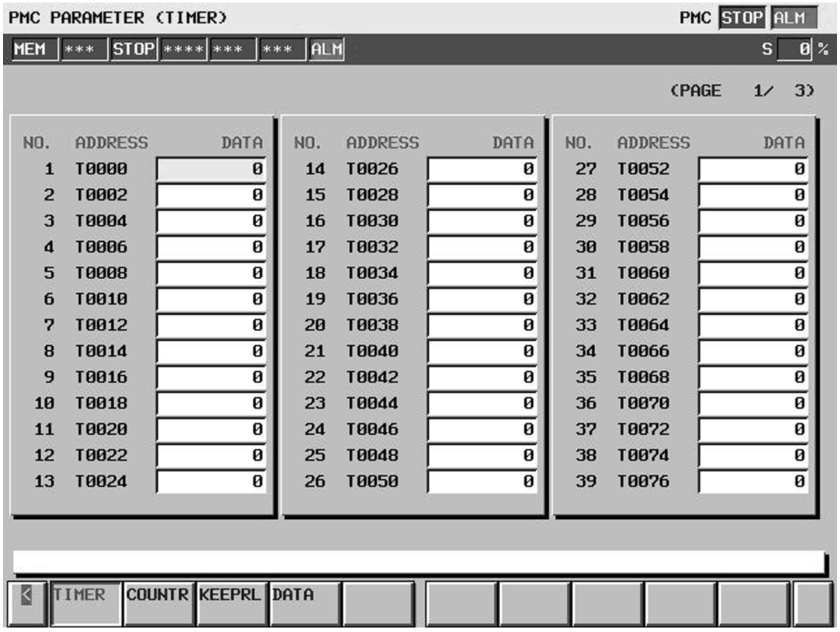


Table contents

- 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

4.5 COUNTER SCREEN (COUNTR)

This screen is used to set and display the maximum and minimum counter values for machine instruction counters (SUB5).

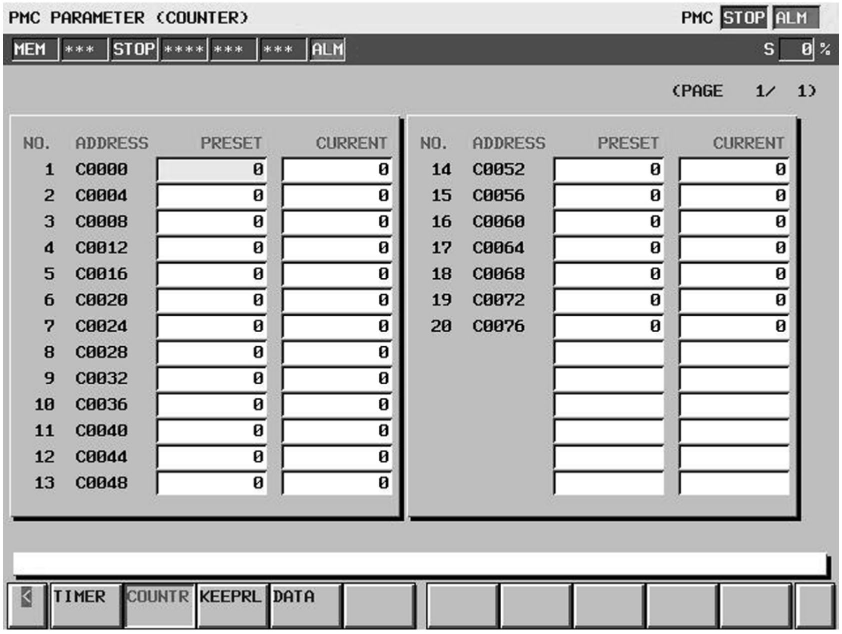


Table contents

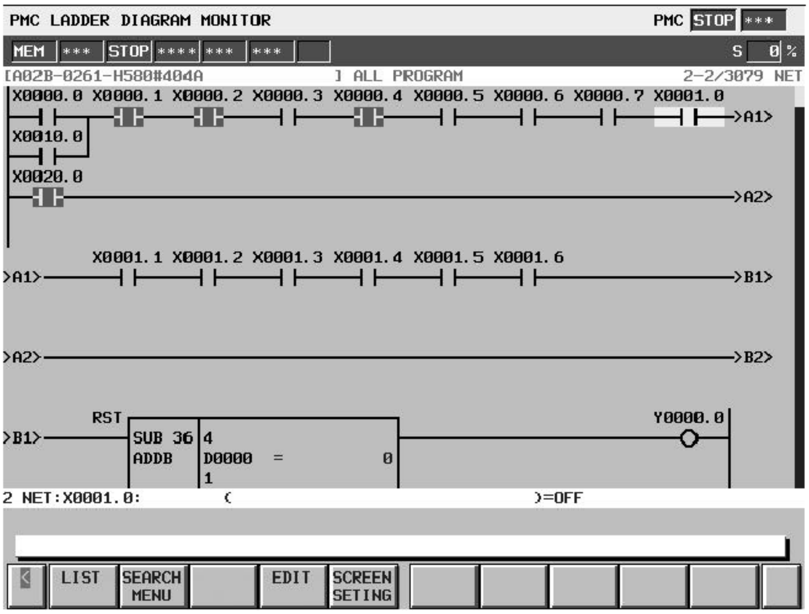
- NO.: Counter number specified for a machine instruction counter
- ADDRESS: Address referenced by a sequence program
- PRESET: Maximum counter value (a minimum counter value is specified by a counter instruction)
- CURRENT: Current counter value

Counter types and maximum values

Counter type	PRESET maximum value	CURRENT maximum value
BINARY	32767	32767
BCD	9999	9999

4.6 KEEP RELAY SCREEN (KEEPRL)

This screen is used for setting and displaying the Keep Relays.



Contents of the table

- ADDRESS: Address refered by sequence Program
- 0 – 7 Contents of each bit
- HEX: Value of the byte data by hexadecimal nortatin

Since Keep Relay is nonvolatile memory, the contents are not lost even if you turn off the power .
Keep Relay area consists of parts as follows.

User area	K0 – K39
Special use area (Note)	K900 – K909

CAUTION
Do not use the Special use area , because the relays in this area are reserved for PMC system software use, and they affect behavior of the PMC software. Set ”0” to any relays that are not mentioned below, to prevent unexpected behavior of PMC.

(1) Special use area (K900 – K909)

PMC type	PMC–NB6
Special use area 1	K900
Reserved	K901 – K909

	#7	#6	#5	#4	#3	#2	#1	#0
K900	DTBL DSP			MEMINP		AUTORUN	PRGRAM	LADMASK

#7 DTBLDSP 0 : The Data Table Controlling Data screen is displayed.
1 : The Data Table Controlling Data screen is not displayed.

#4 MEMINP 0 : The status of signals is not allowed to be modified.
1 : The status of signals is allowed to be modified.

This setting is effective for the following screens.
Embedded PMC:

* LADDER Diagram Monitor screen (Forced I/O function)
  FAPT LADDER–II and Ladder Editting Package
* Signal Status Screen (Online function)

#2 AUTORUN 0 : Sequence program starts automatically at power on.
1 : Sequence program starts by pressing of soft key [RUN]

#1 PRGRAM 0 : Embedded debug function is unavilable.
1 : Embedded debug function is avilable.

This setting is effective for the following screens and functions
Embedded PMC:

* LADDER Diagram Editor screen, I/O Unit Allocation Editor screen
* I/O screen

FAPT LADDER–II and Ladder Editting Package

* Ladder transfer function
* Ladder Editor Screen (Online function)

#0 LADMASK 0 : Ladder diagram is displayed.
1 : Ladder diagram is not displayed.

This setting is effective for the following screens and functions
Embedded PMC:

* LADDER Diagram Editor screen, I/O Unit Allocation Editor screen
* I/O screen

FAPT LADDER–II and Ladder Editting Package

* Ladder transfer function
* Ladder Editor Screen (Online function)

CAUTION
Set always ”0” to any other relays in Special us area than mentioned above, to prevent unexpected behavior of PMC.

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.

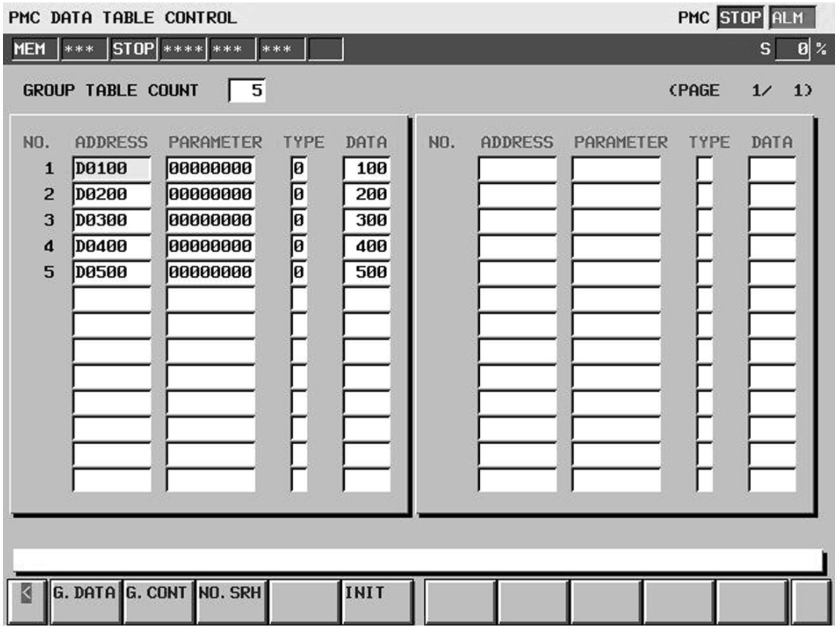


Table contents

- 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.

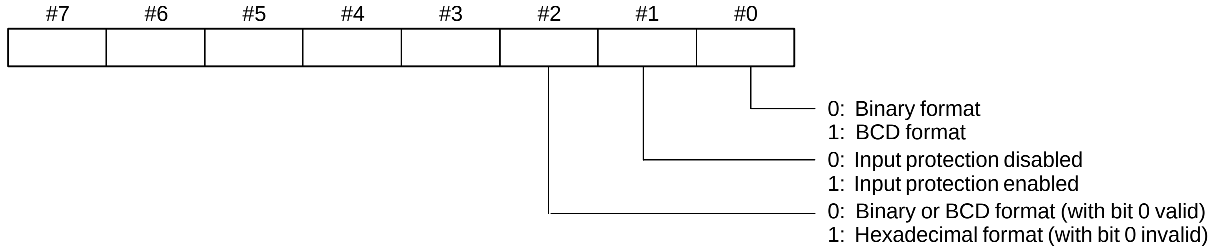
[INIT]: Clicking this soft key initializes the data table.

* 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:



(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.

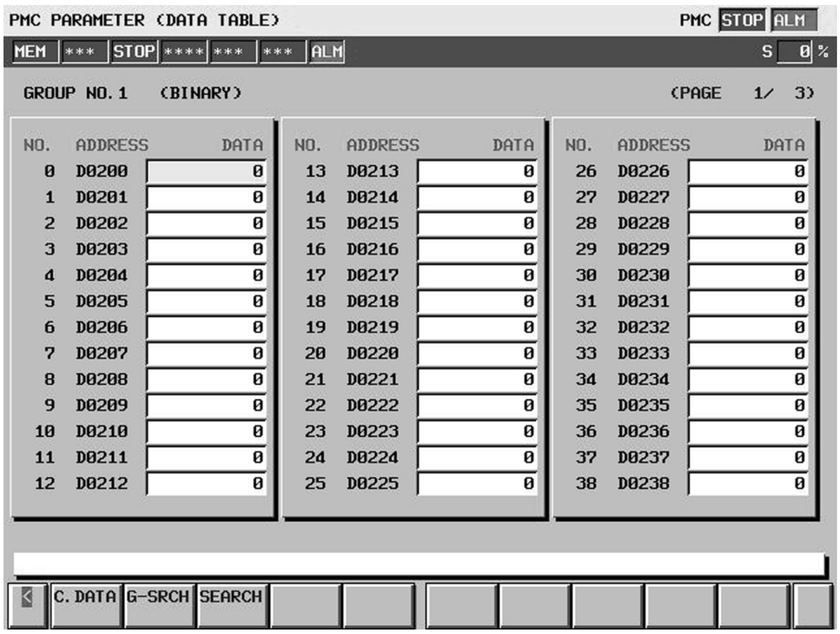


Table contents

- NO.
- ADDRESS: Address used by the sequence program
- DATA

Soft key definitions

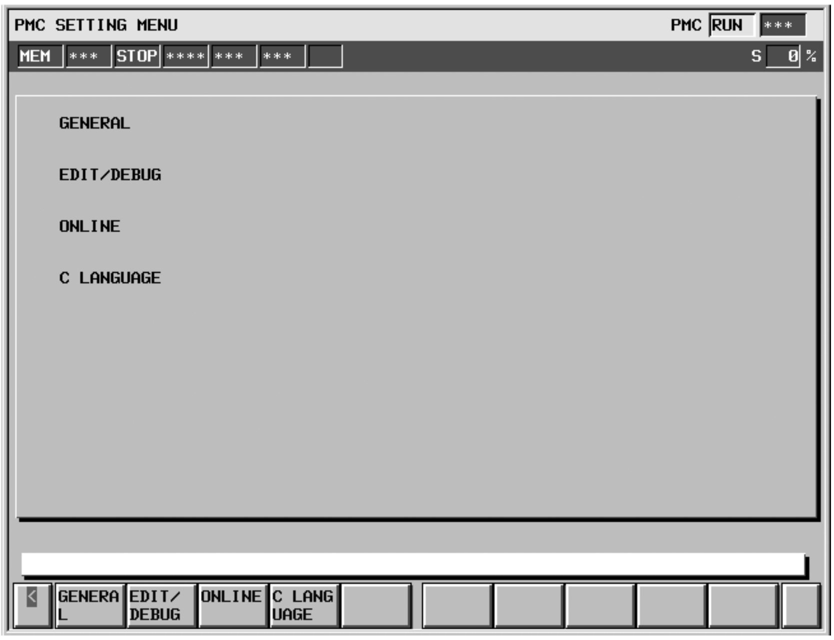
[C.DATA]: Clicking this soft key switches to the data table control data screen.

[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.

5 SETTING MENU (SETING)

Clicking the [SETING] soft key on the PMC basic module screen displays the following setting menu screen.



Menu contents

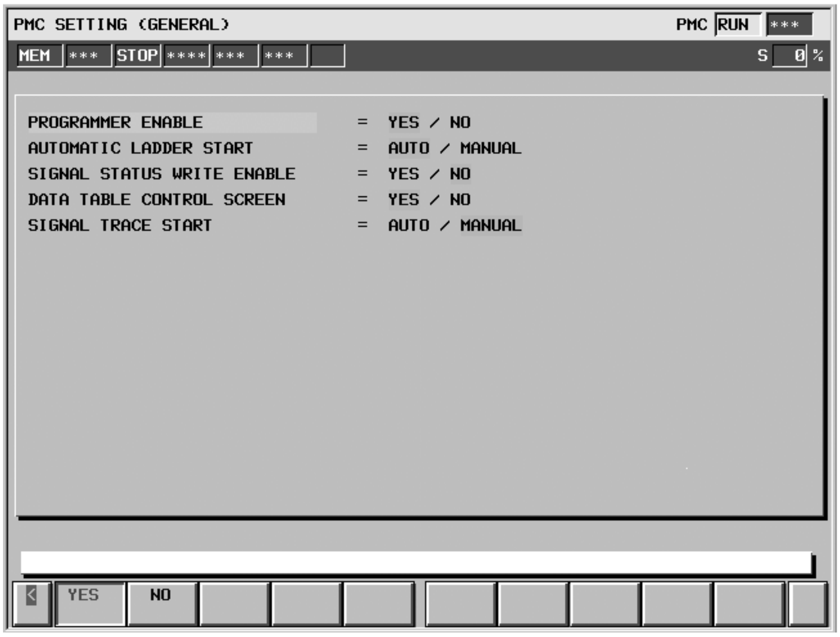
1. GENERAL: Screen for displaying general setting data
2. EDIT/DEBUG: Screen for displaying setting data related to editing and debugging
3. ONLINE: Screen for displaying the communication settings for the online–function. (It is displayed by setting “PROGRAMMER ENABLE” to “YES” on the GENERAL screen.)
4. C LANGUAGE: Screen for displaying setting data related to C Language function.

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.

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.

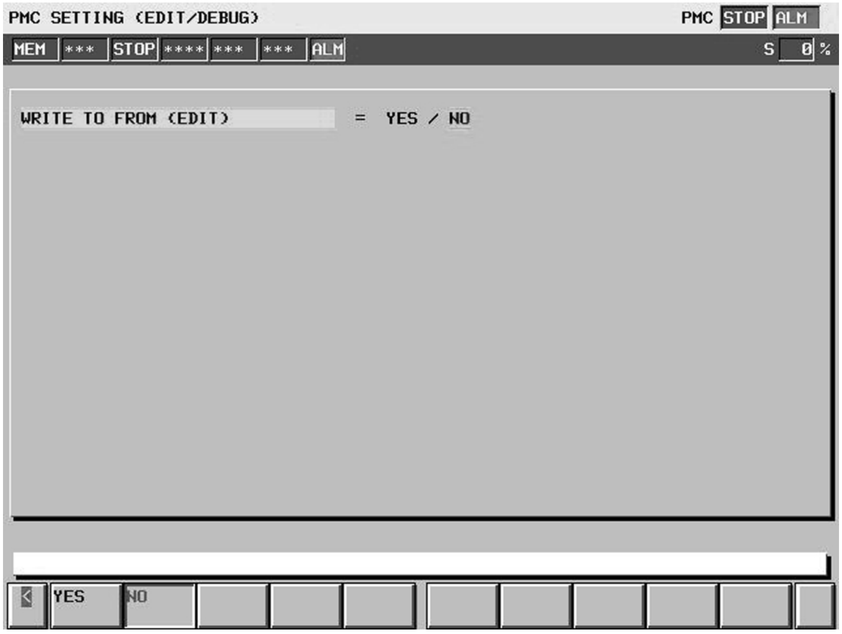
  NO: The PMC parameter data table control screen is not displayed.

- SIGNAL TRACE START

  AUTO: Trace function is executed automatically when the power is switched on.

  MANUAL: Trace function is executed by operating soft key.

5.2 SCREEN FOR DISPLAYING THE SETTING DATA RELATED TO EDITING AND DEBUGGING



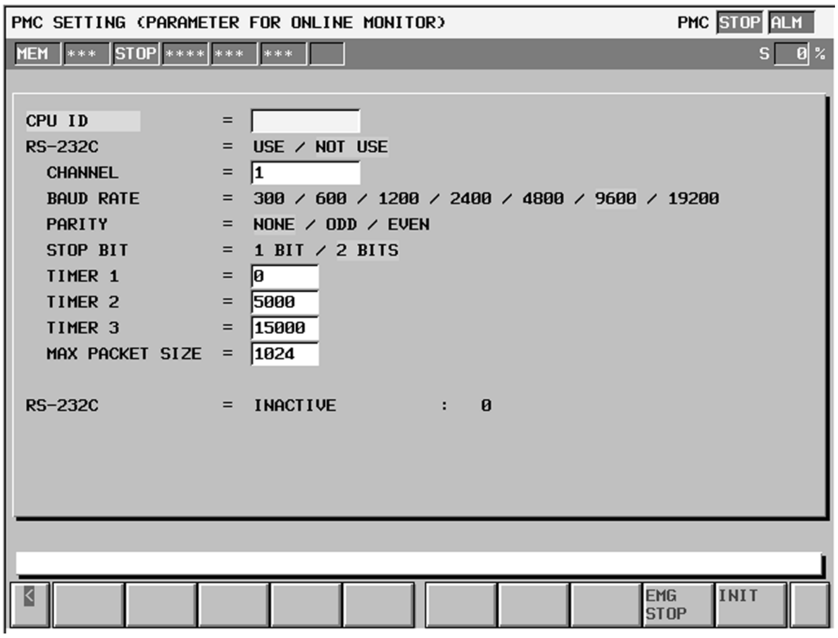
- WRITE TO FROM (EDIT)

  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.

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.

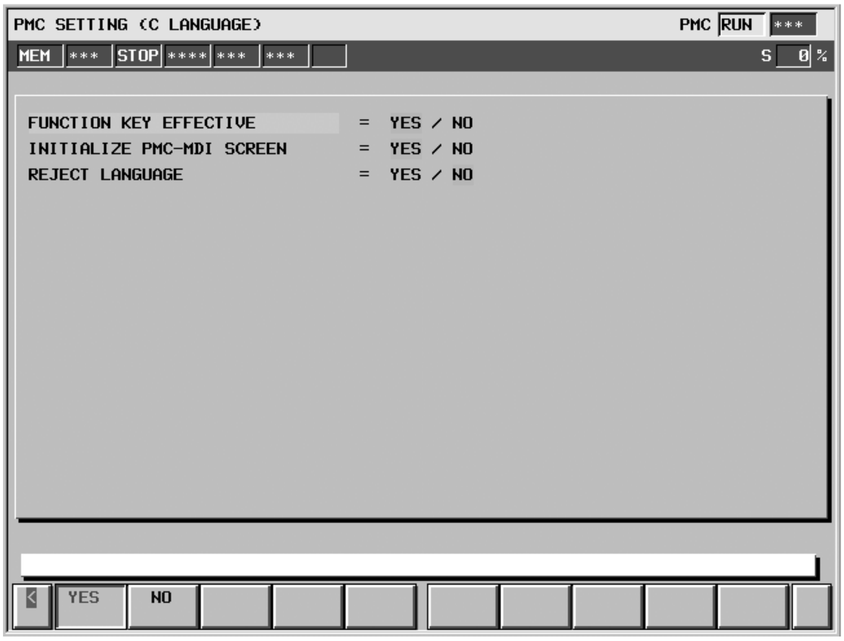
  CONNECTED: An F–BUS port has been connected to FAPT LADDER–II.

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.

5.4 SCREEN FOR DISPLAYING SETTING DATA RELATED TO C LANGUAGE FUNCTION



- FUNCTION KEY EFFECTIVE

  YES: Function keys are enabled when the user program displays the user screen.

  NO: Function keys are disabled when the user program displays the user screen.

- INITIALIZE PMC–MDI SCREEN

  YES: When the screen is switched to the PMCMDI screen, the display screen is initialized.

  NO: When the screen is switched to the PMCMDI screen, the display screen is not initialized.

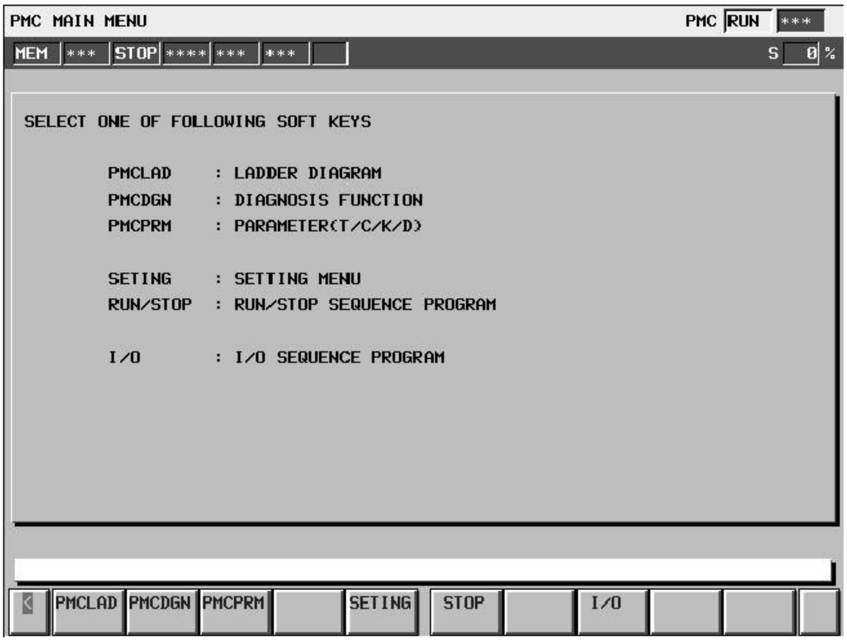
- REJECT LANGUAGE

  YES: The program of C language is not activated.

  NO: The program of C language is activated.

6 LADDER DIAGRAM MONITOR AND EDITOR

The LADDER Diagram Monitor screen displays the LADDER Diagram with the status of the signals in the diagram. You can watch how the LADDER Program works at this LADDER Diagram Monitor screen.
The LADDER Diagram Editor screen allows you to change the LADDER Diagram, adding relays and functional instructions, and change the action of LADDER Diagram.
Press soft key [PMCLAD] at PMC Main Menu to reach the LADDER Diagram Monitor/Editor screen.



NOTE
You can hide [PMCLAD] soft key at PMC Main Menu by setting the keep relay of K900.0 to 1.

LADDER Diagram Monitor/Editor function consists of following screens.

LADDER Diagram Monitor screen (LADDER Diagram Monitor screen)

Displays LADDER Diagram and the current status of relays and others.

LADDER Diagram Editor screen

Edits LADDER Diagram by the net.

Net Editor screen

Edits the structure of a net in LADDER Diagram

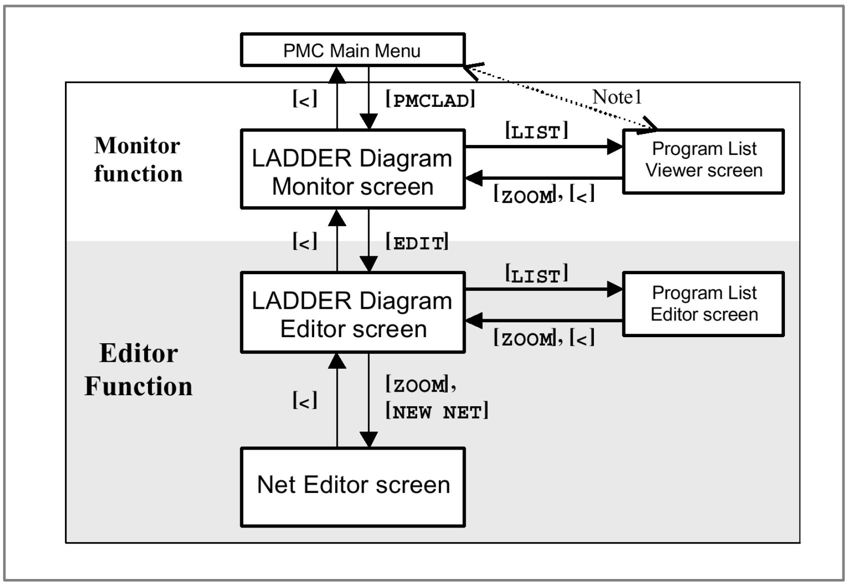
Program List Viewer screen

Chooses the subprogram to be displayed at LADDER Diagram Monitor screen.

Program List Editor screen

Edits LADDER Diagram by the subprogram. Also chooses the subprogram to be displayed at LADDER Diagram Editor screen.

These screens can be reached as described bellow.



NOTE

1 When [PMCLAD] soft key is pressed at first time after CNC power cycle, Program List Viewer screen is displayed. Then use [ZOOM] soft key to select subprogram to reach LADDER Diagram Monitor screen, and the subprogram that you have selected appears on the screen. Once you choose subprogram, pressing [PMCLAD] soft key will directly reach LADDER Diagram Monitor screen. Changing LADDER program by I/O function will make the Program List screen appear at [PMCLAD] soft key again. See ”Program List screen” for more detail. (404A/04 does not have Program List Viewer screen. LADDER Diagram Monitor screen can be reached directly from PMC Main Menu always.)

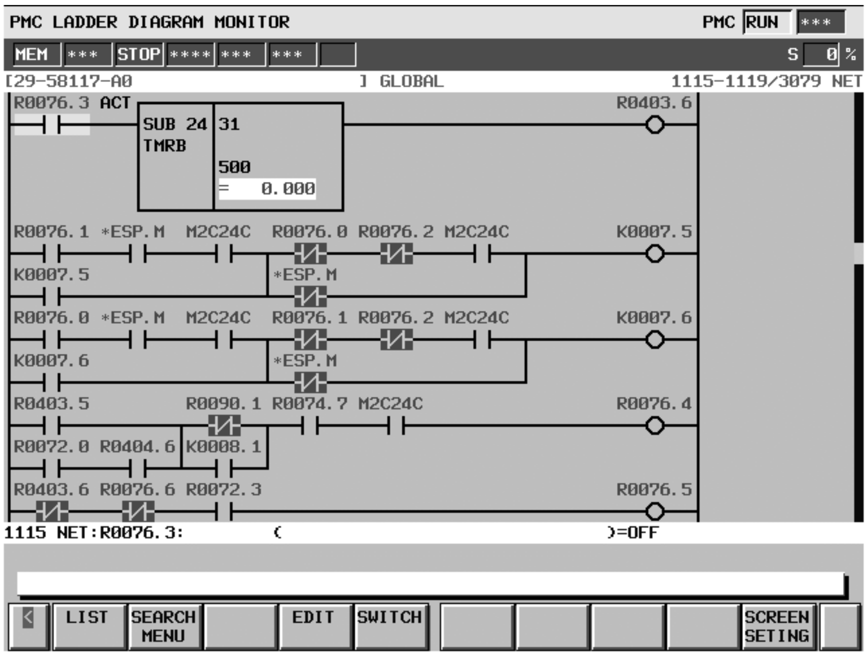
2 404A/04 can not display LADDER program that is protected by Password function. To view a protected LADDER program, you need FAPT LADDER–II, LADDER Editing Package, or upgrade version to 404A/05 or later. 404A/05 or later will ask for password when necessary.

3 [EDIT] soft key in LADDER Diagram Monitor screen appears only when Programmer function is enabled: to enable Programmer function, set ”PROGRAMMER ENABLE” to ”YES” at GENERAL screen of PMC Settings. While Online Monitor function is active, you can not reach LADDER Diagram Editor screen. To use LADDER Diagram Editor function, you have to disable Online Monitor function at PMC Setting screen: choose ”NOT USE”.

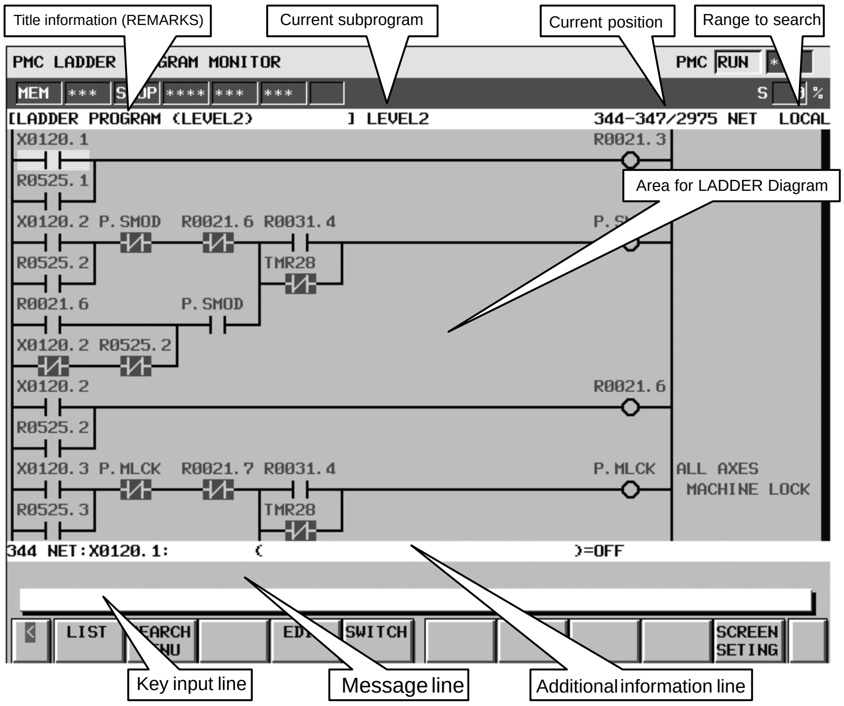
6.1 LADDER DIAGRAM MONITOR SCREEN

LADDER Diagram Monitor screen shows the on/off status of contacts and coils, and the contents of address specified for parameter of functional instructions. You can investigate how the LADDER program is working by this monitor screen.
You can use following operation at this screen, including ”Forced I/O function (Forcing mode)”, by which you can force the relay or the address parameters of functional instructions to a new status or value.

- Switch subprogram to show [LIST]
- Search for address or others [SEARCH MENU]
- Show data table of functional instructions [DATA TABLE]
- Go to LADDER Diagram Editor screen [EDIT]
- Forced I/O function (Forcing mode) ”number” + INPUT key



6.1.1 Screen Structures



(a) Screen Structure

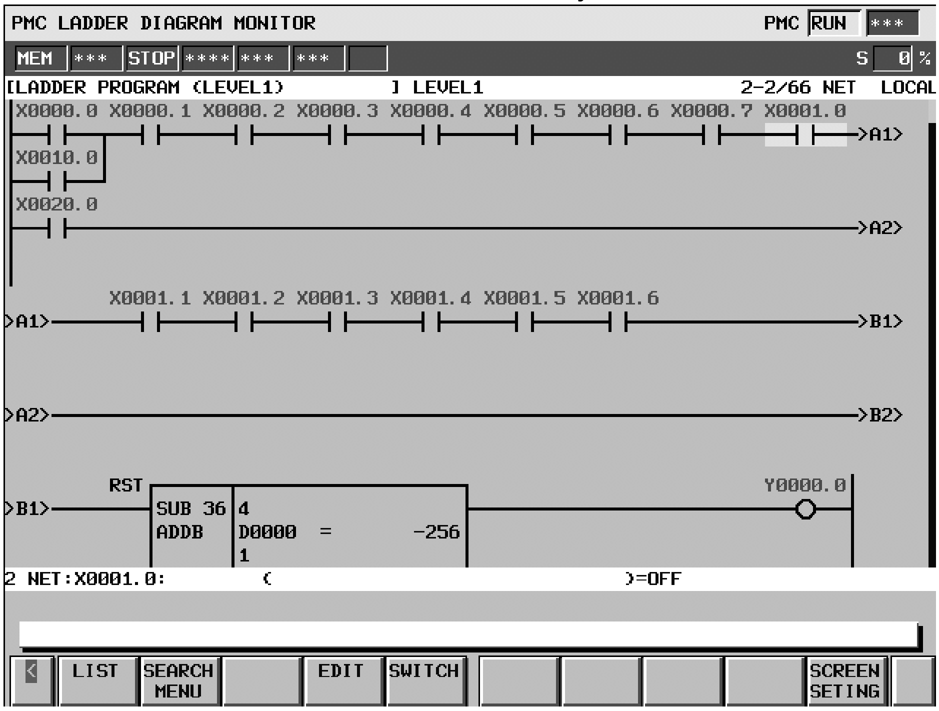
1. Title information (REMARKS) of the LADDER Program, the current subprogram, and the current position information of the Diagram displayed in this screen, are displayed above the LADDER Diagram.
   When you select a subprogram to be displayed, range for search function is indicated at right of the top line as ”LOCAL” or ”GLOBAL”. In case of ”LOCAL”, the range for search function is restricted within the current subprogram. In case of ”GLOBAL”, on the other hand, search function searches whole of LADDER program, and switch current subprogram automatically according to the result of searching.
2. In the additional information line near the bottom of the screen, the following information of the address under the cursor when the cursor is shown.
   - Net number of the net with the cursor
   - Address and its symbol and comment information
   - Current value
3. In the message line, error messages or inquiry messages will be displayed depending on the situation.
4. In the area for LADDER Diagram, 8 or 9 relays side by side, and 10 relays in vertical can be displayed at maximum.

(b) LADDER Diagram

1. The Functional Instruction occupies less space than the Diagram of PMC–NB/NB2 for FS15B. And some names of control conditions are shortened as follows:

UPDOWN	→	UPD	CTR (SUB5), CTRC (SUB55)
CONT	→	CNT	SFT (SUB33)
CIRC	→	CIR	SPCNT (SUB46)
OVRD	→	OVD	SPCNT (SUB46)

2. Nets wider than the screen width are displayed as ”Continuous Net” using continuous marks (”>A1>”).
   Same continuous marks mean they are connected with each other.



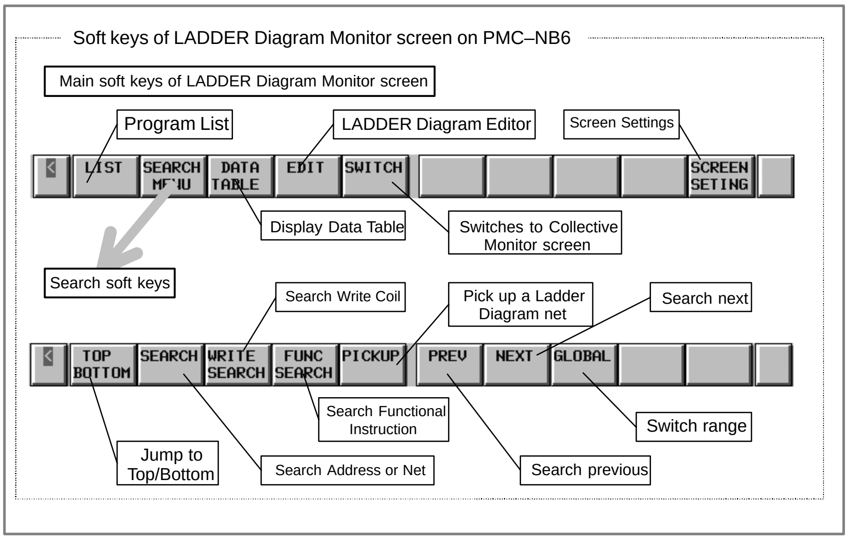
(c) Monitor

1. Contacts and coils are displayed in different colors according to the status of the signal. The status of power flow is not displayed.
2. The contents of address parameters of functional instructions are not shown in default setting. With the extended functional instruction format, you can see the contents of the parameters. See ”Setting Screen” for the detail.

(d) Displaying Symbols and Comments

1. If an address has a symbol name assigned, the symbol name will be displayed instead of the address itself at default setting. You can force not to show symbol by setting. See ”Setting Screen” for the detail.
2. If the address used with coil has a comment string assigned, the comment strings will be displayed at the right margin beside the coil. You can use this margin area to display an additional relay instead of the comment string by setting: at this setting, one more relay can be displayed in each diagram line. See ”Setting Screen” for the detail.

6.1.2 Operations



(a) Operation with Soft–keys

1. [LIST] Go to Program List Viewer screen

   Goes to Program List Viewer screen to choose subprogram to be displayed at LADDER Diagram Monitor screen.
2. [SEARCH MENU] Search & Jump

   Change soft keys to ”Search soft keys”. Use Return key [<] to return to the ”Main soft keys”.
   ”Search soft keys” consists of followings:

   - [TOP BOTTOM] Jump to Top/Bottom

     Jumps to the top of LADDER Program. If the top is displayed already, then jump to the bottom.

   - [SEARCH] Search Address/Net

     Searches the PMC address or the net according to the preceding string. You can specify both of bit address and byte address.
     When digits are entered, the digits are supposed to be a net number and the screen will jump to the net of the net number.
     When a string other than digits is entered, the string is examined as a symbol for some PMC address at first. If the string matches a symbol, then the address that the symbol means will be searched for.
     If no symbol matches the string, then the string is examined as PMC address at next. If the string indicates correct PMC address, then the address will be searched for.
     When cursor is hidden, the net that has the specified net number or contains the specified address will be shown at the top of the screen.
     When cursor is shown, the cursor moves to the relay or the parameter to show the found address directly.

   - [WRITE SEARCH] Search Write Coil

     Searches for the write coils with the address that entered string means. Any contacts with the address are ignored.

   - [FUNC SEARCH] Search Functional Instruction

     Searches for the functional instructions by its SUB number or its mnemonic name such as ”TMR” or ”END2”.

- [PICKUP] Picking up ladder nets.

  Picks up ladder nets with coil which you want to monitor, on Collective Monitor screen. When picks up is complete, the ” ” mark displayed to the top of net..

- [PREV] Search previous

  Repeats to search the same thing backward (upward).

- [NEXT] Search next

  Repeats to search the same thing forward (downward).

- [GLOBAL] / [LOCAL] Change range for searching

  Changes the range for searching between GLOBAL and LOCAL; GOBAL means whole of program, and LOCAL means within the displaying subprogram. Current range for searching is indicated at right of the information line at top of screen.

3. [DATA TABLE] Go to Functional Instruction Data Table Viewer screen

   Goes to Functional Instruction Data Table Viewer screen to examine contents of Data Table of functional instructions such as COD (SUB 7) and CODB (SUB 27), which have Data Table in themselves. This soft key appears only when the cursor is on a functional instruction that has Data Table.

4. [EDIT] Go to LADDER Diagram Editor screen

   Goes to LADDER Diagram Editor screen. This soft key appears only when Programmer function[1] is enabled. And activating Online Monitor function[2] disables this soft key.

5. [SWITCH] Switches to Collective Monitor screen.

   Switches to Collective Monitor screen.

6. [SCREEN SETING] Screen settings

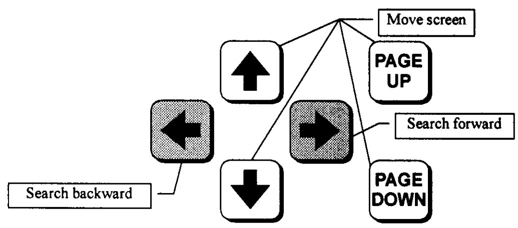
   Goes to setting screen for LADDER Diagram Monitor screen. You can change various settings for LADDER Diagram Monitor screen at the screen. Use Return key [<] to return to LADDER Diagram Monitor screen. See ”Settings for LADDER Diagram Monitor” section for the detail.

(b) Other operations

1. Cursor move keys, Page change keys

   With cursor hidden, you can move diagram in the screen by up/down Cursor move keys and Page change keys.

Function of Cursor keys when cursor is hidden



With cursor displayed, you can move the cursor by all cursor move keys and Page change keys. When cursor is placed on some relay or some address parameter of a functional instruction, the information about the address under cursor is displayed at ”Additional Information Line”.

[1] To enable Programmer function, go to GENERAL screen of PMC Settings, and choose ”YES” at item ”PROGRAMMER ENABLE”.
[2] To deactivate Online Monitor function, go to ONLIEN screen of PMC Settings, and choose ”NOT USE” at settings of ”RS–232C” and ”F–BUS” (if appears).

2. ”number” + INPUT key

   When cursor is shown, you can force the value of the address under cursor by entering new value as ”number” + INPUT key. In this screen, Forced I/O function is limited only to Forcing mode. This Forced I/O function asks you to confirm your intention before it takes effect. Once it is confirmed that you actually want to change value by this function, you can change the value of the same relay or parameter without further confirmation. However, after you move cursor or you operate other functions, you will be asked when you use the Forced I/O function again.

WARNING

1 You have to pay special attention to use Forced I/O function to change status of signals. Inappropriate use of Forced I/O function may cause unexpected reaction of machine. You have to make it sure that **nobody is near the machine** when you use this function.

2 As you use Forcing mode of Forced I/O function to change status of signal, however, the signal may look proof against Forced I/O function, because LADDER program or I/O device writes into the signal repeatedly. In this case, even if the signal looks unchanged, actual signal may be changed in very short moment. You should be careful for the reaction of machine to such signal changes.

NOTE

1 Forced I/O function is enabled when ”SIGNAL STATUS WRITE ENABLE” setting is set to ”YES” in GENERAL screen of PMC Settings. If the setting is ”NO”, INPUT key will be just ignored.

2 Parameters of timer functional instructions, TMR, TMRB, and TMRC, which have special monitor formats, are not supported by Forced I/O function.

(c) Notes for Search function

1. The string followed by [SEARCH] is treated as symbol first. In case that the symbol ”D0” is assigned to the bit address ”R0.0”, the operation ”D0” + [SEARCH] will search the bit address ”R0.0”, instead of byte address ”D0”.[3]
2. To search the symbol that consists of only digit characters, which will usually be treated as net number, you can use leading space to specify explicitly the string is symbol. For example, while ”123” + [SEARCH] will search the 123rd net from top of the LADDER, ”_123” + [SEARCH] (”_” is space) will search address with symbol ”123”.

[3] In this case, you can still search the byte address ”D0” by entering ”D00”. The excess ”0” avoids the conflict with the symbol ”D0”.

3. When the range for searching is GLOBAL, and the target is found in other than displaying subprogram, the screen will automatically switch to the subprogram to which the found target belongs. Searching GLOBAL net number that current subprogram does not contain, for example, the subprogram that contains the net of the net number will appear in the screen, displaying the net.

(d) Shortcuts

1. Right/left cursor move keys that are following some string means searching forward/backward. You can use following strings for this search operation:
   - Digits for net number
   - ”1” for top, ”0” for bottom
   - ”S” + digits, for Functional instructions
   - Other string for symbol or bit/byte address
   - Leading space always makes the string treated as a symbol or address.

     Example: ”_123”+[SEARCH] (”_” means space) will search the symbol ”123” instead of the 123rd net.
2. When cursor is hidden, left/right cursor move keys without string act just like [PREV]/[NEXT] soft keys.
3. String followed by [SEARCH MENU] soft key in ”PMCLAD Main soft keys” starts searching directly. In this case, [SEARCH MENU] soft key acts just like right cursor move key.
4. [SEARCH] soft key in ”Search soft keys” without string searches the address or the functional instruction under cursor forward. If cursor is hidden, or cursor is placed neither on a relay nor on a functional instruction, this operation just repeats the last successful search forward, just like [NEXT] soft key.
5. [WRITE SEARCH] soft key without string searches forward a write coil of the same address with relay under cursor. If cursor is hidden, or cursor is not placed on a relay, this operation will search a write coil of the bit address that is searched at last successful search. If the last search was not made with bit address, the last entered string for searching is used to determine what bit address is to be searched for a write coil.
6. [FUNC SEARCH] soft key without string searches forward the same functional instruction with one under cursor. If cursor is hidden, or cursor is not placed on a functional instruction, this operation will search a functional instruction that is searched at last successful search. If the last search was not made for functional instruction, the last entered string for searching is used to determine what functional instruction is to be searched.
7. [LIST] soft key following string that indicates subprogram, switches subprogram on LADDER Diagram Monitor screen. Examples for strings to specify subprogram are following:

”L1”	Level 1
”P10”, ”10”	Subprogram ”P10”
”0”(zero), ”G”	Whole of LADDER program (Global)

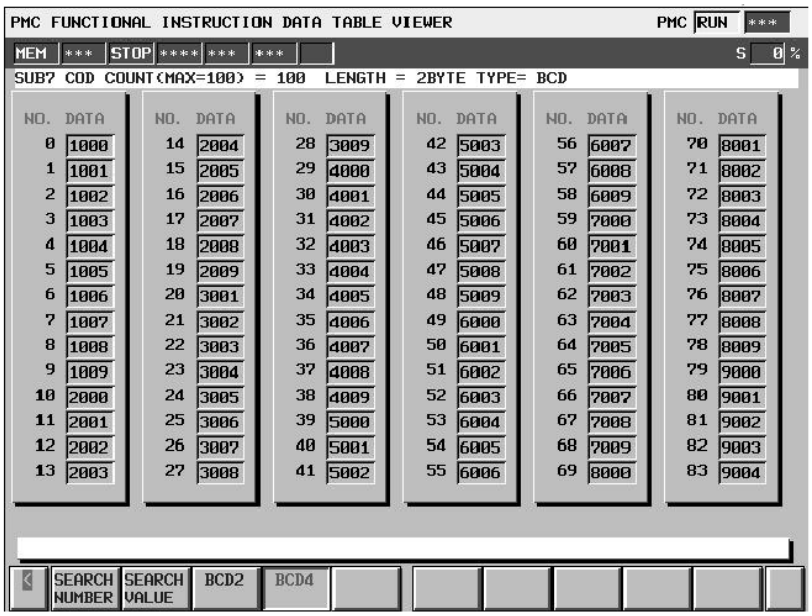
6.1.3 Functional Instruction Data Table Viewer Screen

Functional Instruction Data Table Viewer screen shows the contents of data table that belongs to some functional instructions.
To reach this screen, at LADDER Diagram Monitor screen, press [DATA TABLE] soft key that is displayed when the cursor is on the following functional instructions which have a data table.

- Functional Instruction COD (SUB7)
- Functional Instruction CODB (SUB27)

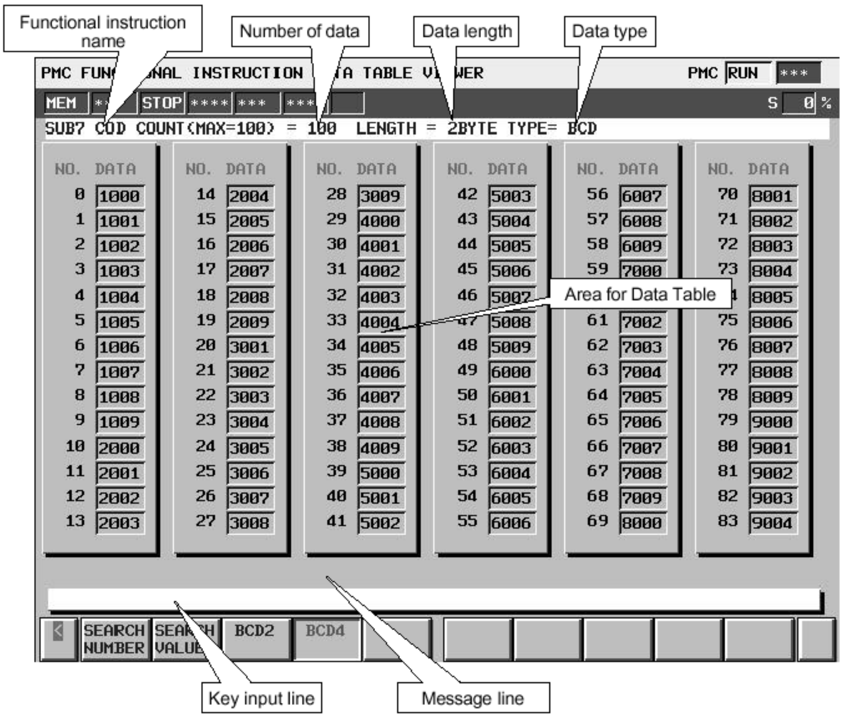
Following operations are available at this screen.

- Search for data table number. [SEARCH NUMBER]
- Search for data value. [SEARCH VALUE]
- Change the displaying data digits. [BCD2],[BCD4]
  (These soft keys can be operated only at Functional Instruction Data Table Viewer screen of Functional Instruction COD.)



Functional Instruction Data Table Viewer screen of Functional Instruction COD.

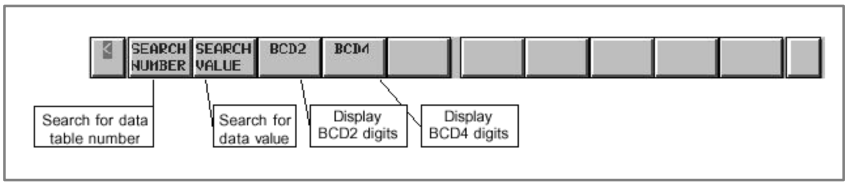
(1) Screen Structures



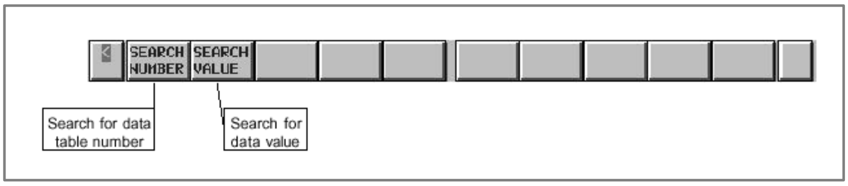
1. The functional instruction name, the number of data table, data length and data type are displayed above the Data Table.
2. In the message line, error messages or inquiry messages will be displayed depending on the situation.
3. In case of functional instruction COD, the data of 6 lines and 14 columns can be displayed in the area for Data table.
   In case of functional instruction CODB, the data of 4 lines and 14 columns can be displayed in the area for Data table.

(2) Operations

Soft keys of Functional Instruction Data Table Viewer screen of functional insturction COD.



Softkeys of Functional Instruction Data Table Viewer screen of functional insturction CODB.



1. [SEARCH NUMBER] Search for data table number

   Searches the data table number which you specified.
   Then the cursor is displayed on the data of the target number.
   And the cursor is disappeared when you operate something.

2. [SEARCH VALUE] Search for data value

   Searches the data value which you specified.
   Then the cursor is displayed on the target data.
   And the cursor is disappeared when you operate something.

3. [BCD2] Display BCD2 digits

   Changes the display data type to 2 digits of BCD. This operation only switches the display data type. So it does never edit the data on memory.

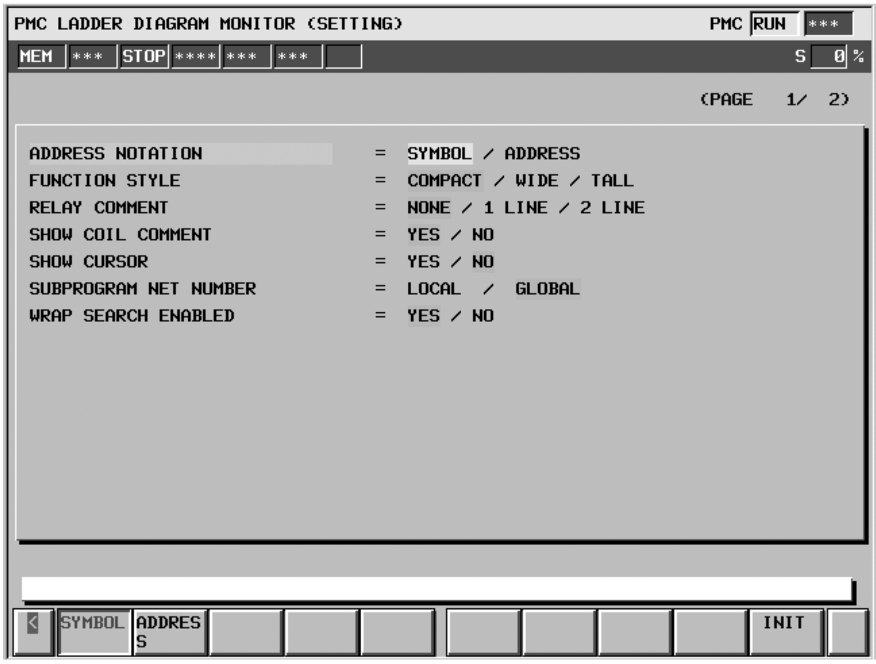
4. [BCD4] Display BCD4 digits

   Changes the display data type to 4 digits of BCD. This operation only switches the display data type. So it does never edit the data on memory.

5. Cursor move keys, Page change keys

   You can scroll screen by right/left cursor move keys and Page change keys.

6.1.4 Setting Screen



1st page of setting screen

(a) Setting items

LADDER Diagram Monitor Setting screen contains the setting items below:

- ADDRESS NOTATION

  Determines whether the bit and byte addresses in the LADDER Diagram are displayed as corresponding symbols, or the addresses themselves.

  SYMBOL (default)

  Addresses that have a symbol are displayed by the symbols. Addresses without symbols are displayed by the addresses themselves.

  ADDRESS

  All addresses are displayed as the addresses themselves even if they have a symbol.

- FUNCTION STYLE

  Change the shape of functional instructions. There are three options as below. You have to choose other than ”COMPACT” to show the current values of address parameters of functional instructions.

  COMPACT (default)

  Occupies least space in diagram. Monitors of current values of address parameters are omitted.

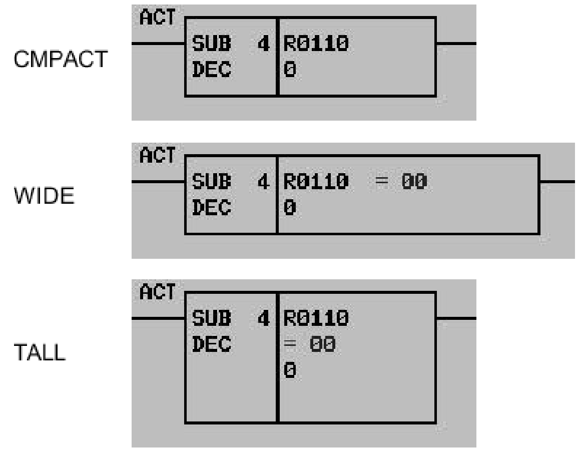
  WIDE

  Extends the box horizontally to reserve spaces for the monitors of current values of address parameters. The box becomes wider than COMPACT.

TALL

Extends the box vertically to reserve spaces for the monitors of current values of address parameters. The box becomes taller than COMPACT.

Display styles of functional instructions



The displays of current values of address parameters change their format according to each parameter. Refer to the table in ”Display Format for Parameters” for detail.
When you place the cursor on an address parameter, its current value is displayed in the ”Additional Information Line” in both formats of binary decimal, and BCD (or hexadecimal binary).

- RELAY COMMENT

Set the style of relay comment. These are three options as below. When you display relay comment, less ladder diagram nets are shown.

NONE (default)

Relay comments are not displayed.

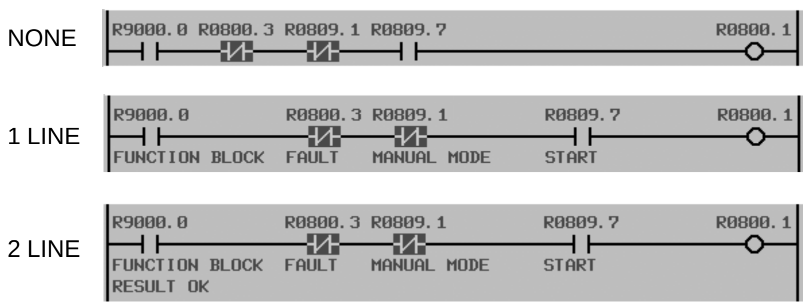
1 LINE

Relay comments are displayed in one line. Up to 15 characters can be displayed. Characters after 16th character will not be displayed. Width of a relay with comment of more than 7 characters will be extended.

2 LINE

Relay comments are displayed in two lines. Up to 30 characters can be displayed. Width of a relay with comment of more than 7 characters will be extended.

Display styles of relay comment



- SHOW COIL COMMENT

Determines whether to show coil comments.

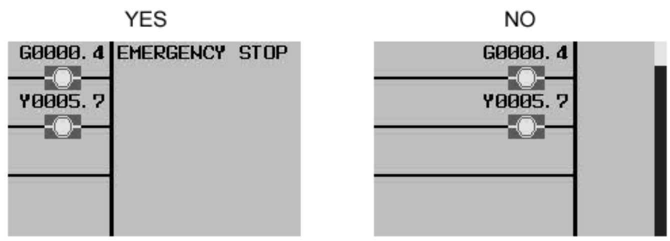
YES (default)

Right margin of 2 lines of 15 characters is reserved for display of coil comments.

NO

Right margin is used to extend diagram by one more relay, instead of showing coil comment. The screen position bar is also displayed at the right edge of the screen in this option.

Show Coil comment setting



- SHOW CURSOR

Determines whether to show cursor.

YES

Cursor is displayed. Cursor move keys will move the cursor. When the cursor is placed on bit or byte addresses, the information of the address is displayed at ”Additional Information Line”. When you search something with the cursor displayed, the cursor goes directly where it is found. This option is recommended for search operation with LADDER program that contains many large nets.

NO (default)

Cursor is not displayed. Up/down cursor move keys will scroll the contents of screen directly. When you search something with the cursor hidden, the net, which contains it, will appear at the top of the screen.

- SUBPROGRAM NET NUMBER

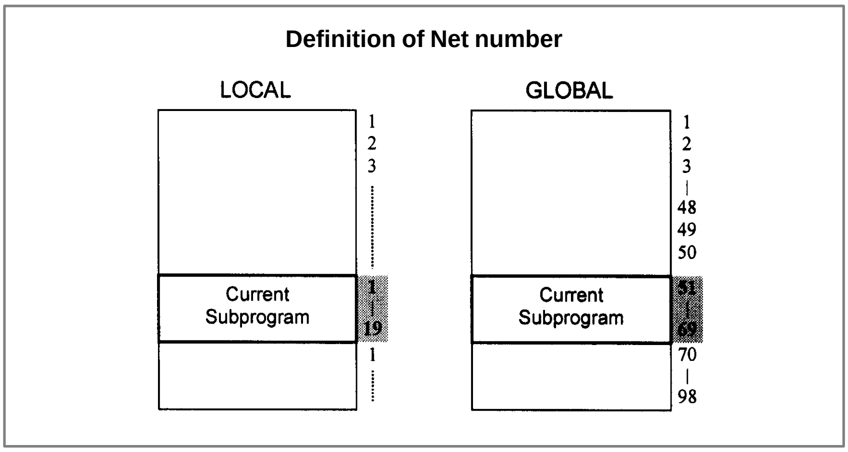
Determines whether a net number is counted as ”LOCAL” starting from the top of current subprogram, or is counted as ”GLOBAL” starting from the top of whole program. This setting also affects the expression of net number at searching nets by number.

LOCAL

Net number starts from 1 at top of current subprogram. Net number is defined only within current subprogram. The net number information at upper right of the screen is displayed in the format ”displaying range / nets in subprogram NET”.

GLOBAL (default)

Net number starts from 1 at top of Level 1 program. Net number is defined identically at whole of program. The net number information at upper right of the screen is displayed in the format ”displaying range / subprogram range NET”.



- WRAP SEARCH ENABLED

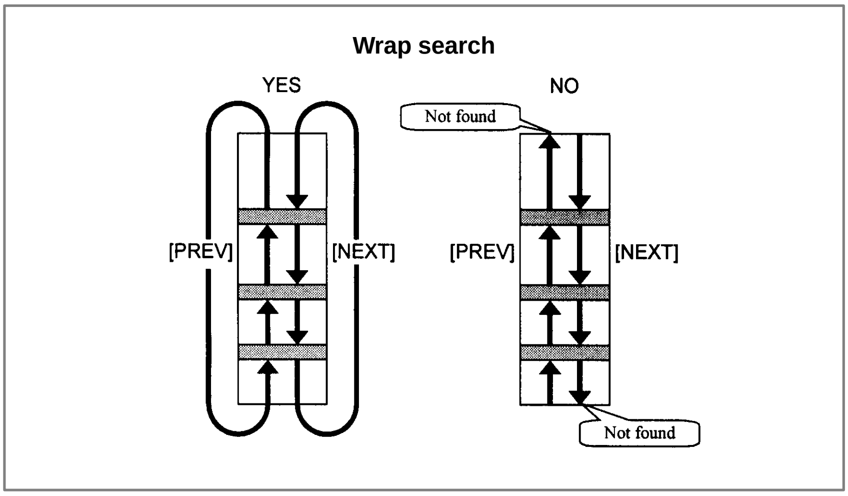
  Allows search process to wrap from top/bottom to bottom/top to continue to search.

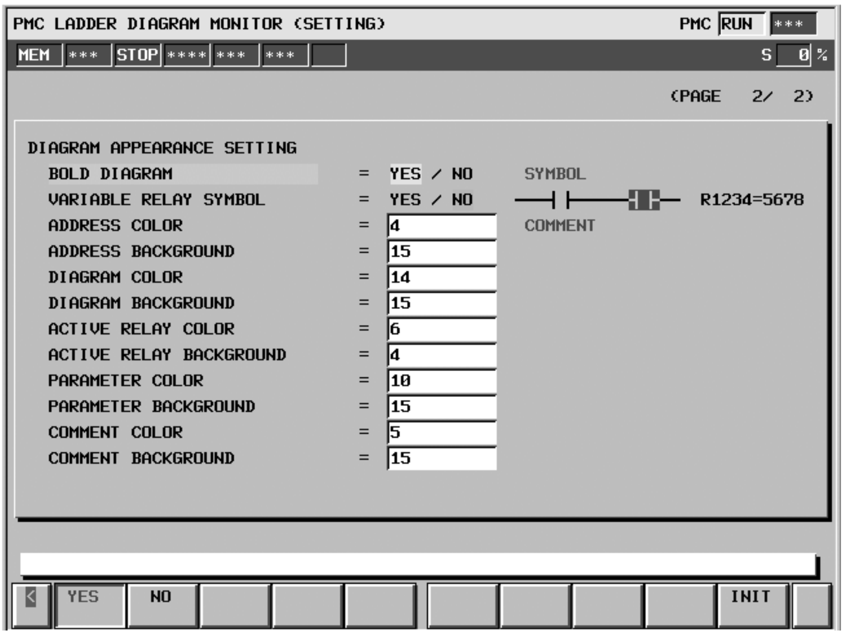
  YES (default)

  Downward search will continue to search from top of LADDER when reaches to bottom. Upward search will also continue to search from bottom when reaches to top.

  NO

  Search process will fail when reached top or bottom, and displays an error message at Message Line.





- DIAGRAM APPEARANCE SETTING

Changes the appearance of LADDER diagram. Lines, relays, and functional instructions that constitute LADDER diagram can be changed in the colors and the shapes. Samples of ”Off Contact”, ”On Contact”, ”Monitor of Parameter in functional instruction” are displayed at right side. These samples change their appearance according to the following current settings.
For the color setting, LADDER diagram is categorized into these three parts, and you can specify their colors independently.

BOLD DIAGRAM

Sets thickness of diagram lines. This setting also affects shapes of relays.

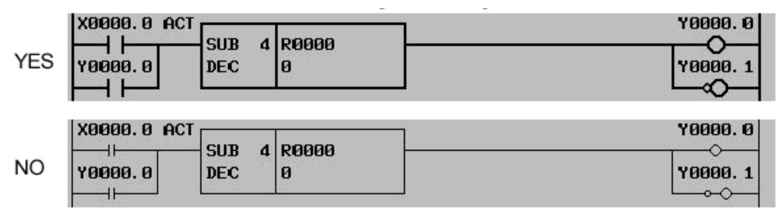
YES (default)

Diagram is drawn with thick lines. Relays are drawn in more distinct shapes. You have to choose this option to make following ”VARIABLE RELAY SYMBOL” setting effective.

NO

Diagram is drawn with thin lines. Relays are drawn in smaller shapes.

Bold Diagram setting



NOTE
The setting ”BOLD DIAGRAM” requires ”Character registration function” of CNC software, which is supported by Series F002 version 03 or later. If CNC software does not support it, diagram is always displayed with thin lines.

VARIABLE RELAY SYMBOL

Determines whether to change the shapes of relays according to their on/off status, or to fix their shapes. This setting is effective only when the setting ”BOLD DIAGRAM” is set to ”YES”.

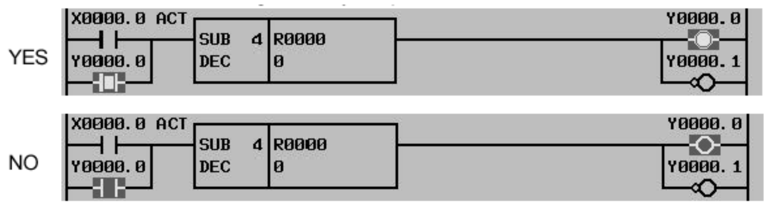
YES

Shapes of relays change according to their on/off status. This option is effective only when the setting ”BOLD DIAGRAM” is set to ”YES”. If it is set to ”NO”, shapes of relays will not change regardless of this setting.

NO (default)

Shapes of relays will not change.

Change of relay shapes at on/off status



NOTE
The setting ”VARIABLE RELAY SYMBOL” requires ”Character registration function” of CNC software, which is supported by Series F002 version 03 or later. If CNC software does not support it, shapes of relay will not change according to their status.

ADDRESS COLOR
ADDRESS BACKGROUND

Colors for the relay address are set. You can specify these colors by entering color number, or by moving right and left cursor. 16 colors (from No.0 to No.15) are available. Foreground color should be different from background one.

DIAGRAM COLOR
DIAGRAM BACKGROUND

General color and its background color for LADDER Diagram. You can specify these colors by entering color number, or by using right and left cursor move keys to change the color number. You can use 16 numbers from 0 to 15; however, some different number may correspond to the same color. You can not specify the same number to the foreground and the background colors.

ACTIVE RELAY COLOR
ACTIVE RELAY BACKGROUND

Color setting for the active relay. When a contact allows power flow, and when a coil receives power, they are active and are displayed with this color setting. When contacts and coils are not active, they are displayed with ”general color”. You can specify these colors in the same manner as color setting of DIAGRAM COLOR and DIAGRAM BACKGROUND.

PARAMETER COLOR
PARAMETER BACKGROUND

Color setting for the monitor of functional instruction parameters. They are displayed when functional instructions are displayed in the shape other than ”COMPACT”. You can specify these colors in the same manner as color setting of DIAGRAM COLOR and DIAGRAM BACKGROUND.

COMMENT COLOR
COMMENT BACKGROUND

Colors for the relay comment are set. You can specify these colors by entering color number, or by moving right and left cursor. 16 colors (from No.0 to No.15) are available. Foreground color should be different from background one.

(b) Soft keys

LADDER Diagram Monitor Setting screen has the soft keys of options and following:

- [INIT] Initialize all settings

  All settings will be initialized to the default values.

6.1.5 Display Format for Parameters

The following table shows all monitor formats for each parameter of each functional instruction.

NOTE
1 "Variable" in "Monitor Format" field means that this parameter changes its size according to the other parameter. Refer to the descriptions for each functional instruction for detail.
2 Functional instruction with "*" mark has Data table.

No.	Name	Parameter	Monitor Format
1	END1	–	–
2	END2	–	–
3	TMR	1	special
4	DEC	1	2–digits BCD
		2	constant
5	CTR	1	special
6	ROT	1	constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD
7	COD *	1	constant
		2	2–DIGITS BCD
		3	4–digits BCD
8	MOVE	1	constant
		2	constant
		3	2–digits HEX
		4	2–digits HEX
9	COM	1	constant
10	JMP	1	constant
11	PARI	1	1–byte binary
14	DCNV	1	no monitor
		2	no monitor
15	COMP	1	constant
		2	4–digits BCD
		3	4–digits BCD
16	COIN	1	constant
		2	4–digits BCD
		3	4–digits BCD
17	DSCH	1	constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD

No.	Name	Parameter	Monitor Format
18	XMOV	1	constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD
19	ADD	1	constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD
20	SUB	1	constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD
21	MUL	1	constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD
22	DIV	1	constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD
23	NUME	1	constant
		2	4–digits BCD
24	TMRB	1	constant
		2	special
25	DECB	1	constant
		2	variable binary
		3	constant
		4	2–digits HEX
26	ROTB	1	constant
		2	variable binary
		3	variable binary
		4	variable binary
		5	variable binary

No.	Name	Parameter	Monitor Format
27	CODB *	1	constant
		2	constant
		3	1–byte binary
		4	variable binary
28	MOVOR	1	2–digits HEX
		2	2–digits HEX
		3	2–digits HEX
29	COME	–	–
30	JMPE	–	–
31	DCNVB	1	constant
		2	no monitor
		3	no monitor
32	COMPB	1	constant
		2	constant or variable binary
		3	variable binary
33	SFT	1	4–digits HEX
34	DSCHB	1	constant
		2	variable binary
		3	variable binary
		4	variable binary
		5	variable binary
35	XMOVB	1	constant
		2	variable binary
		3	variable binary
		4	variable binary
		5	variable binary
36	ADDB	1	constant
		2	variable binary
		3	constant or variable binary
		4	variable binary
37	SUBB	1	constant
		2	variable binary
		3	constant or variable binary
		4	variable binary
38	MULB	1	constant
		2	variable binary
		3	constant or variable binary
		4	variable binary
39	DIVB	1	constant
		2	variable binary
		3	constant or variable binary
		4	variable binary

No.	Name	Parameter	Monitor Format
40	NUMEB	1	constant
		2	constant
		3	variable binary
41	DISPB	1	constant
42	EXIN	1	8–digits HEX
43	MOVB	1	1–byte binary
		2	1–byte binary
44	MOVW	1	2–bytes binary
		2	2–bytes binary
45	MOVN	1	constant
		2	4–bytes binary
		3	4–bytes binary
46	SPCNT	1	4–bytes binary
		2	no monitor
		3	8–digits HEX
48	END3	–	–
51	WINDR	1	2–bytes binary
52	WINDW	1	2–bytes binary
53	AXCTL	1	constant
		2	8–digits HEX
54	TMRC	1	constant
		2	special
		3	special
55	CTRC	1	2–bytes binary
		2	2–bytes binary
57	DIFU	1	constant
58	DIFD	1	constant
59	EOR	1	constant
		2	variable HEX
		3	constant or variable HEX
		4	variable HEX
60	AND	1	constant
		2	variable HEX
		3	constant or variable HEX
		4	variable HEX
61	OR	1	constant
		2	variable HEX
		3	constant or variable HEX
		4	variable HEX
62	NOT	1	constant
		2	variable HEX
		3	variable HEX
64	END	–	–
65	CALL	1	no monitor

No.	Name	Parameter	Monitor Format
66	CALLU	1	no monitor
68	JMPB	1	no monitor
69	LBL	1	no monitor
70	NOP	1	constant
71	SP	1	no monitor
72	SPE	–	–
73	JMPC	1	no monitor

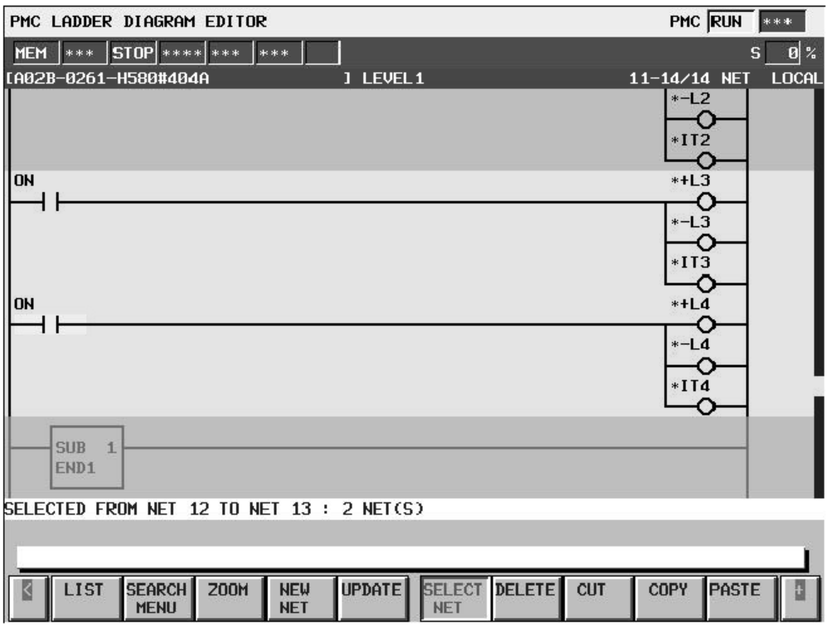
6.1.6 Functional Instructions of Special Monitor forMat

<table>
<tr><td>TMR</td><td>Displays in ”Current/Preset” format by seconds</td></tr>
<tr><td>CTR</td><td>Displays in ”Current/Preset” format by BCD or binary according to the Counter Type setting in LADDER Program.</td></tr>
<tr><td>TMRB</td><td>Displays current value by seconds (Preset value is displayed by milliseconds)</td></tr>
<tr><td>TMRC</td><td>The 2nd parameter shows preset value, and the 3rd parameter shows current value as their monitor displays.
These two monitor displays changes their format according to the 1st parameter as below:
<table>
<tr><th>1st Parameter</th><th>Precision</th><th>Display Format</th></tr>
<tr><td>0</td><td>8 ms</td><td>by second</td></tr>
<tr><td>1</td><td>48 ms</td><td>by second</td></tr>
<tr><td>2</td><td>1 second</td><td>HH:MM:SS</td></tr>
<tr><td>3</td><td>10 second</td><td>HH:MM:SS</td></tr>
<tr><td>4</td><td>1 minute</td><td>HH:MM</td></tr>
</table>
</td></tr>
</table>

6.2 LADDER DIAGRAM EDITOR SCREEN

At LADDER Diagram Editor screen you can edit LADDER program to change its behavior.
To reach LADDER Diagram Editor screen, press [EDIT] soft key at LADDER Diagram Monitor screen.
Following operations are available at LADDER Diagram Editor screen. For more detail of these operations, refer to the descriptions of each key to operate.

● Delete by net	[DELETE]
● Move by net	[CUT] & [PASTE]
● Copy by net	[COPY] & [PASTE]
● Change address of contacts and coils	”bit address” + INPUT key
● Change parameters of functional instructions	”number” or ”byte address” + INPUT key
● Add new net	[NEW NET]
● Change construction of net	[ZOOM]
● Make changes effective	[UPDATE]
● Abandon changes	[RESTORE]

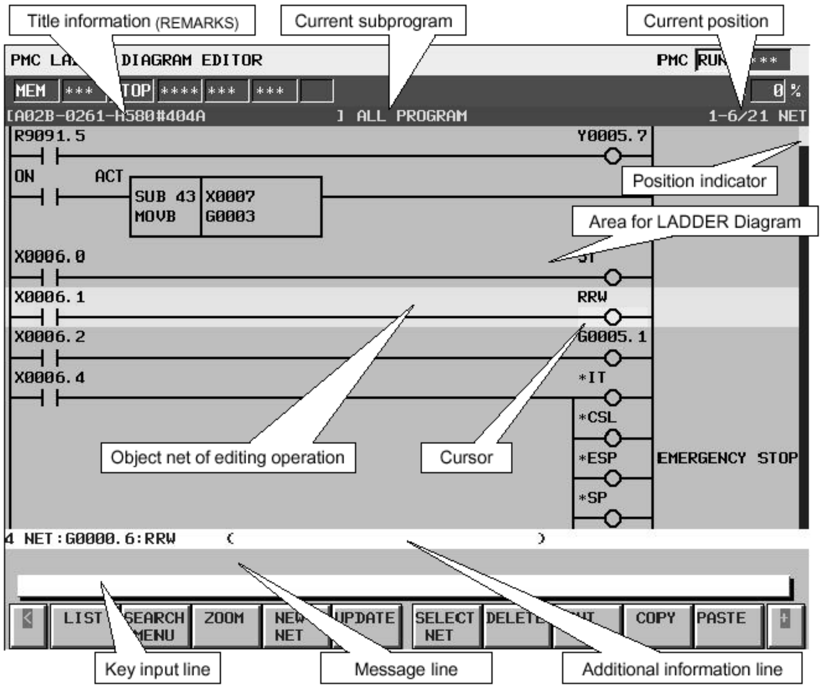


You can modify LADDER even while the LADDER is running. At modifying running LADDER, the modified LADDER will not run just after each operation to modify it. PMC system preserves the LADDER at the point of entering LADDER Diagram Editor screen, and the preserved LADDER will continue to run while editing operation. When exiting LADDER Diagram Editor screen, or pressing [UPDATE] soft key, the modified LADDER starts to run and the modifications take effects.

NOTE

1. [EDIT] soft key in LADDER Diagram Monitor screen appears only when Programmer function is enabled. To enable Programmer function, set ”PROGRAMMER ENABLE” to ”YES” at GENERAL screen of PMC Settings. While Online Monitor function is active, you can not reach LADDER Diagram Editor screen. To use LADDER Diagram Editor function, you have to disable Online Monitor function at PMC Setting screen: choose ”NOT USE”.
2. To edit LADDER protected by password, you have to unlock the protection first. Enter ”password” + INPUT key to unlock it when password is requested.

6.2.1
Screen Structures



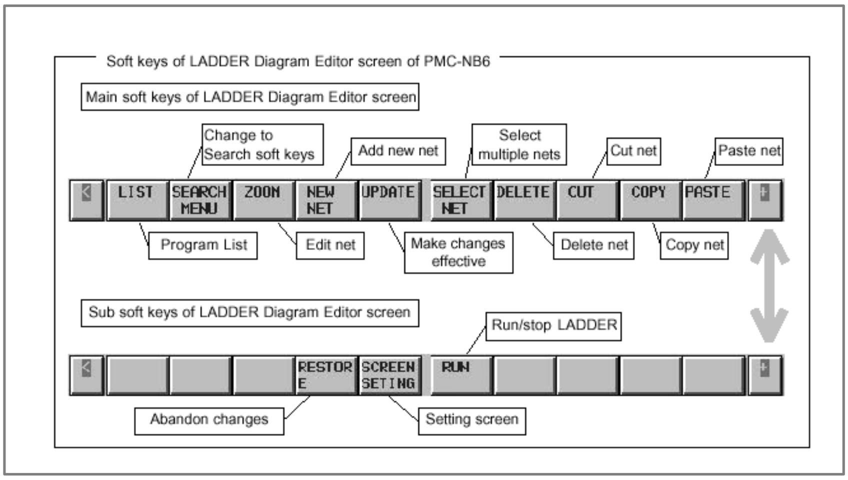
(a) Screen Structure

1. It is basically same with LADDER Diagram Monitor screen, except that no monitor displays of relays and parameters of functional instructions are displayed.
2. At right side of LADDER Diagram area, a position bar is always displayed, which indicates screen position within current subprogram: in LADDER Diagram Monitor screen, this position bar is exclusively displayed with comments of write coil. Sometimes, this position bar hides a part of write coil comments.

(b) LADDER Diagram

1. Style of LADDER Diagram is basically same with LADDER Diagram Monitor screen, except that functional instructions are drawn always in ”COMPACT” format that has no monitor displays.
2. Cursor is shown always. And the net, which will be an object of following editing operations, is emphasized in screen.

6.2.2 Operations



(a) Operation with Soft–keys

1. [LIST] Go to Program List Editor screen

   Goes to Program List Editor screen to choose which subprogram to be edited at LADDER Diagram Editor screen. The Program List Editor screen can also edit subprograms; remove some subprograms, or add new one.

2. [SEARCH MENU] Search & Jump

   Change soft keys to ”Search soft keys”. Use Return key [<] to return to the ”Main soft keys”.
   ”Search soft keys” are quite same with ones of LADDER Diagram Monitor screen. See descriptions of search functions of LADDER Diagram Monitor screen.

3. [ZOOM] Change construction of net

   Goes to Net Editor screen to modify structure of the selected net. For detail of editing operation at Net Editor screen, see descriptions of Net Editor screen.

4. [NEW NET] Add new net

   Create and add new net to cursor position. Pressing this soft key reaches Net Editor screen, so that new net is constructed.
   For detail of editing operation at Net Editor screen, see descriptions of Net Editor screen.

5. [UPDATE] Make changes effective

   Updates running LADDER program to currently edited LADDER program, so that the all modifications will take effects, and remains Editor screen. If it succeeds to update running LADDER, edited LADDER starts to run.
   Before updating LADDER program, edited LADDER program is checked. If some problem is found, a message is displayed, and updating process aborts. At some kind of error, cursor jumps where the error is detected.

WARNING

You have to pay special attention to modify running LADDER program. If you modify LADDER program in wrong way, or update LADDER program with the machine in improper status, it may cause unexpected reaction of the machine. You have to make it sure that modifications you make on LADDER program is appropriate, machine is in proper status, and nobody is near the machine, when you update LADDER program.

CAUTION

At updating LADDER program, it may take rather long time to complete updating process in some cases according to the activity of LADDER program. If it takes too long time, or never complete at updating process, cancel updating process and correct LADDER program, following instructions in section 7.6.2.5 ”How to correct LADDER program that never stops”.

6. [SELECT NET] Select multiple nets

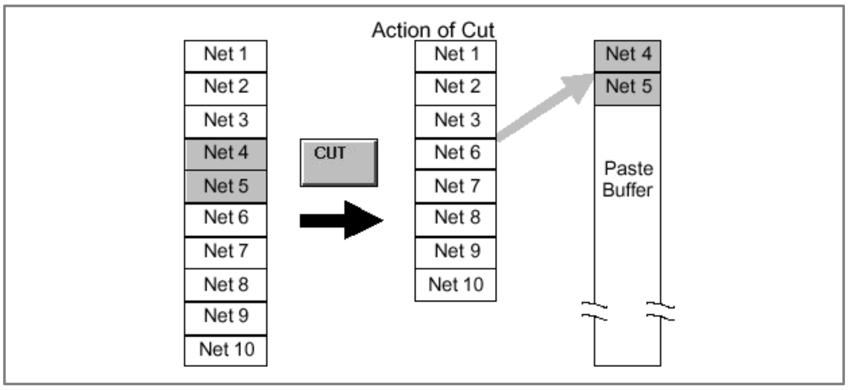
   Selects multiple nets for following operations such as [DELETE].
   Pressing [SELECT NET] soft key leads to a mode to select one or more nets for following editing operation. Use cursor move keys and search functions to select nets, as you like. Being in mode to select nets is indicated by dented [SELECT NET] soft key, and the information of selected nets is shown in additional information line at near bottom of screen.

7. [DELETE] Delete net

   Deletes selected nets. The nets deleted by [DELETE] soft key are lost. If you [DELETE] wrong nets, you have to abandon the all modifications you have made, and restore LADDER program to the original one before editing operation.

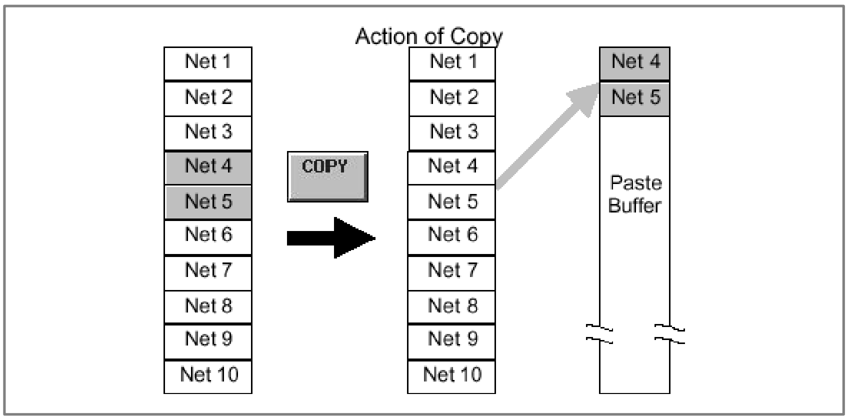
8. [CUT] Cut nets

   Cuts selected nets. The cut nets are preserved in Paste Buffer, and disappear from diagram. The contents of Paste Buffer before [CUT] operation are lost.
   [CUT] and [PASTE] soft keys are used to move nets.
   The Paste Buffer can contain approx. 8000 steps maximum, and [CUT] may fail to cut nets of over 8000 steps.



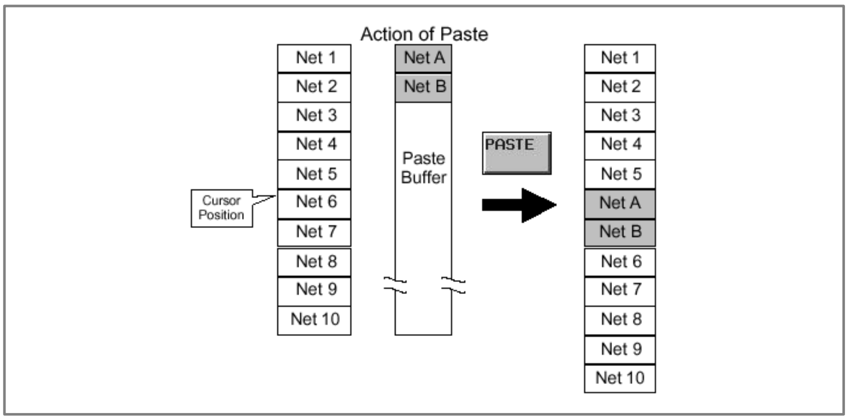
9. [COPY] Copy nets

Copy selected nets into Paste Buffer. No change on diagram will be made. The contents of Paste Buffer before [COPY] operation are lost.
[COPY] and [PASTE] soft keys are used to copy nets.
The Paste Buffer can contain approx. 8000 steps maximum, and [COPY] may fail to copy nets of over 8000 steps.

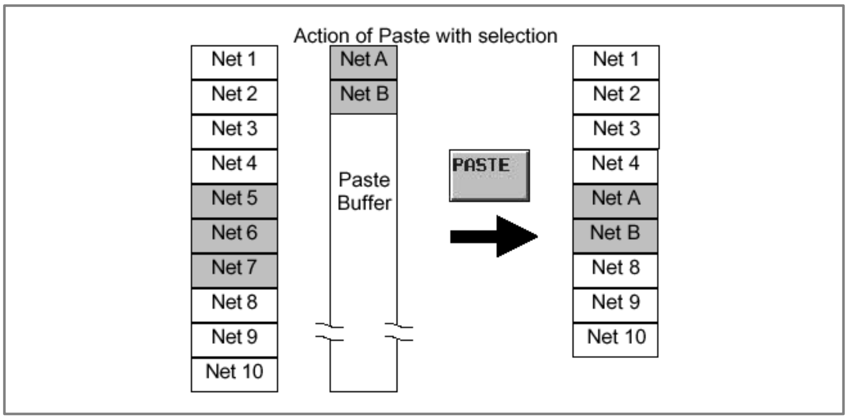


10. [PASTE] Paste nets

Pastes nets at cursor position, which were stored into Paste Buffer by [CUT] or [COPY] soft key.



Pressing [PASTE] soft key while selecting nets using [SELECT NET] soft key, alters the selected nets with the nets in Paste Buffer.



Pasting nets using [PASTE] soft key will not remove contents of Paste Buffer. The contents of Paste Buffer will remain until turning CNC power off.

11. [RESTORE] Abandon changes

    Abandons all changes, and restores LADDER program to the one at entering LADDER Diagram Editor screen, or last updated one using [UPDATE] soft key. This soft key is useful when you make wrong modifications and hard to recover from them.

12. [SCREEN SETING] Screen settings

    Goes to setting screen for LADDER Diagram Editor screen. You can change various settings for LADDER Diagram Editor screen at the screen. Use Return key [<] to return to LADDER Diagram Editor screen. See ”Settings for LADDER Diagram Editor” section for the detail.

13. [RUN] / [STOP] Run and stop LADDER program

    Controls LADDER program execution. [RUN] soft key makes LADDER run, and [STOP] soft key makes LADDER stop. Both soft keys will confirm your intention. When you are sure to run or stop LADDER program, press [YES] to take an action.

WARNING

1 You have to pay special attention to run/stop LADDER program. Running/stopping LADDER program in a wrong timing, or with machine in improper status, may cause unexpected reaction of machine. You have to make it sure that machine is in proper status, and nobody is near the machine when you run/stop LADDER program.

2 At stopping LADDER program, it may take rather long time to complete to stop it in some cases according to the activity of LADDER program. If LADDER takes too long time to stop, or never stop, correct LADDER program, following instructions in section ”7.6.2.5 How to correct LADDER program that never stops”.

14. [<] Exit Editor

    Updates running LADDER program to edited LADDER program, so that the all modifications will take effects, and exits Editor screen. If it succeeds to update running LADDER, edited LADDER starts to run.

    Before updating LADDER program, edited LADDER program

is checked. If some problem is found, a message is displayed, updating process aborts and you can not exit this screen. At some kind of error, cursor jumps where the error is detected.
The function keys such as SYSTEM key will not work when the LADDER Diagram Editor screen is active.

WARNING

1 You have to pay special attention to modify running LADDER program. If you modify LADDER program in wrong way, or update LADDER program with the machine in improper status, may cause unexpected reaction of the machine. You have to make it sure that **modifications you make on LADDER program is appropriate, machine is in proper status**, **and nobody is near the machine**, when you update LADDER program.

2 At updating LADDER program, it may take rather long time to complete updating process in some cases according to the activity of LADDER program. If it takes too long time, or never complete at updating process, cancel updating process and correct LADDER program, following instructions in section ”7.6.2.5 How to correct LADDER program that never stops”.

(b) Other operations

1. Cursor move keys, Page change keys

   Cursor move keys and Page change keys move cursor on screen. When cursor is placed on some relay or some address parameter of a functional instruction, the information about the address under cursor is displayed at ”Additional Information Line”.

2. ”bit address” + INPUT key

   Changes bit address of relay under cursor.

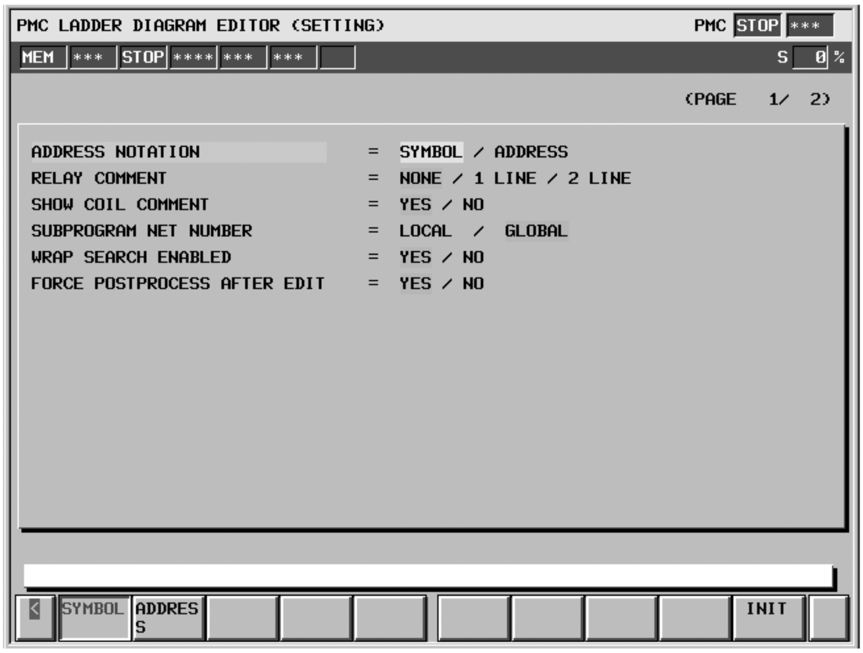
3. ”number” or ”byte address” + INPUT key

   Changes parameter of functional instructions under cursor.
   But, there are some parameters that can not change by this operation. If you see a message that means that this parameter can not be changed, use Net Editor screen to change the parameter.

(c) Shortcuts

1. Same shortcut search operations with LADDER Diagram Monitor screen are available. For their detail, see descriptions about Shortcut operations of LADDER Diagram Monitor screen.
2. Same shortcut operations using [LIST] soft key with LADDER Diagram Monitor screen are available.

6.2.3 Setting Screen



(a) Setting items

LADDER Diagram Editor Setting screen contains the setting items below:

- ADDRESS NOTATION

  Determines whether the bit and byte addresses in the LADDER Diagram are displayed as corresponding symbols, or the addresses themselves.

  SYMBOL (default)

  Addresses that have a symbol are displayed by the symbols. Addresses without symbols are displayed by the addresses themselves.

  ADDRESS

  All addresses are displayed as the addresses themselves even if they have a symbol.

- RELAY COMMENT

  Set the style of relay comment. These are three options as below. When you display relay comment, less ladder diagram nets are shown.

  NONE (default)

  Relay comments are not displayed.

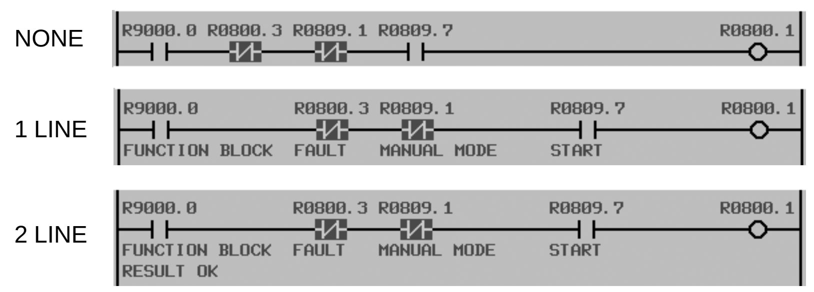
  1 LINE

  Relay comments are displayed in one line. Up to 15 characters can be displayed. Characters after 16th character will not be displayed. Width of a relay with comment of more than 7 characters will be extended.

  2 LINE

  Relay comments are displayed in two lines. Up to 30 characters can be displayed. Width of a relay with comment of more than 7 characters will be extended.

Display style of relay comment



- SHOW COIL COMMENT

Determines whether to show coil comments. Unlike LADDER Diagram Monitor screen, the screen position gauge is always displayed at the right edge of the screen.

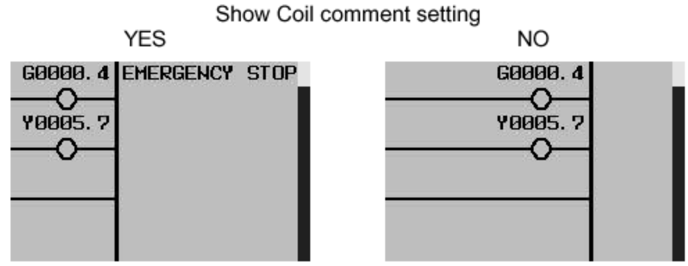
YES (default)

Right margin of 2 lines of 15 characters is reserved for display of coil comments.

NO

Right margin is used to extend diagram by one more relay, instead of showing coil comment.

Show Coil comment setting



- SUBPROGRAM NET NUMBER

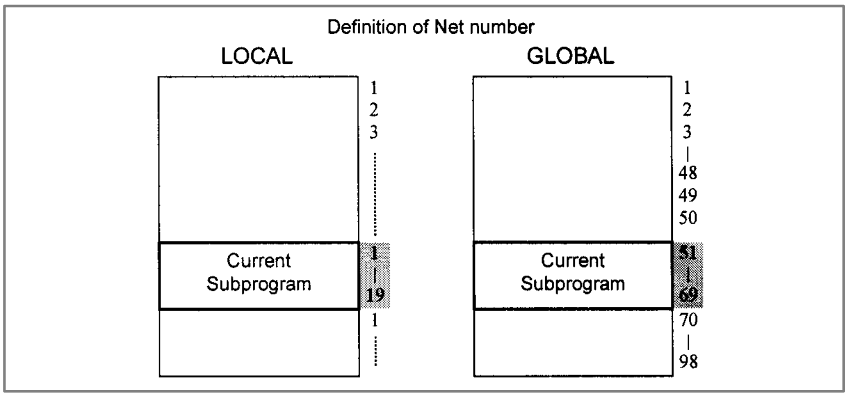
Determines whether a net number is counted as ”LOCAL” starting from the top of current subprogram, or is counted as ”GLOBAL” starting from the top of whole program. This setting also affects the expression of net number at searching nets by number.

LOCAL

Net number starts from 1 at top of current subprogram. Net number is defined only within current subprogram. The net number information at upper right of the screen is displayed in the format ”displaying range / nets in subprogram NET”.

GLOBAL (default)

Net number starts from 1 at top of Level 1 program. Net number is defined identically at whole of program. The net number information at upper right of the screen is displayed in the format ”displaying range / subprogram range NET”.



- WRAP SEARCH ENABLED

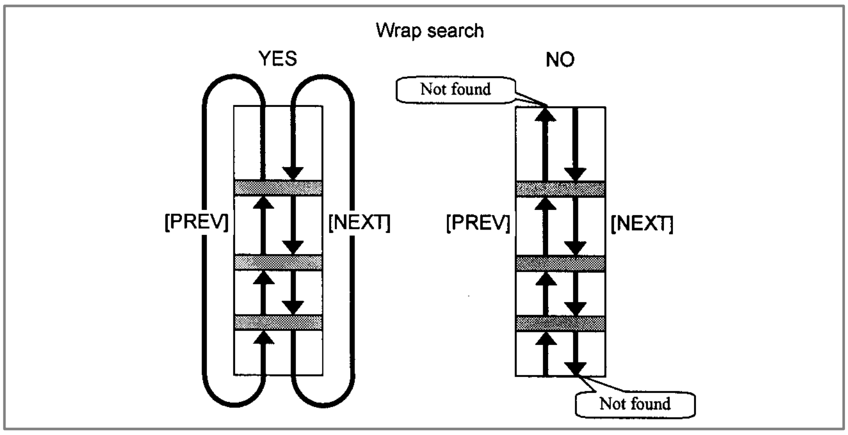
Allows search process to wrap from top/bottom to bottom/top to continue to search.

YES (default)

Downward search will continue to search from top of LADDER when reaches to bottom. Upward search will also continue to search from bottom when reaches to top.

NO

Search process will fail when reached top or bottom, and displays an error message at Message Line.



- FORCE POSTPROCESS AFTER EDIT

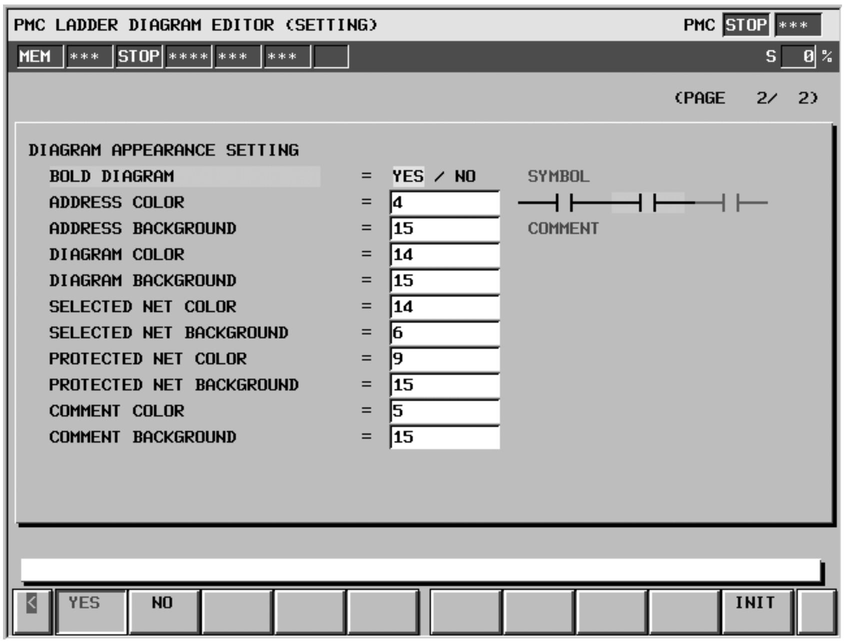
Decides whether the post–processing after editing LADDER program, which makes LADDER program ready to run, will be always done, or will be done only when LADDER program is actually modified, at exiting LADDER Diagram Editor screen.

YES (default)

The post–processing will be always done at exiting LADDER Diagram Editor screen. Even at exiting Editor screen just after entering it, LADDER structure will be checked because of the post–processing.

NO

The post–processing will be done only after LADDER program is actually modified. Before any modification on LADDER program is done, you can always exit Editor screen without any errors, even if the LADDER program has an error originally and can not be executed.



- DIAGRAM APPEARANCE SETTING

Changes the appearance of LADDER diagram. Lines, relays, and functional instructions that constitute LADDER diagram can be changed in the colors and the shapes. Samples of ”Normal net”, ”Selected net”, ”Protected net” are displayed at right side. These samples change their appearance according to the following current settings.
For the color setting, LADDER diagram is categorized into these three parts, and you can specify their colors independently, by entering color number, or by using right and left cursor move keys to change the color number. You can use 16 numbers from 0 to 15; however, some different number may correspond to the same color. You can not specify the same number to the foreground and the background colors of a part.

BOLD DIAGRAM

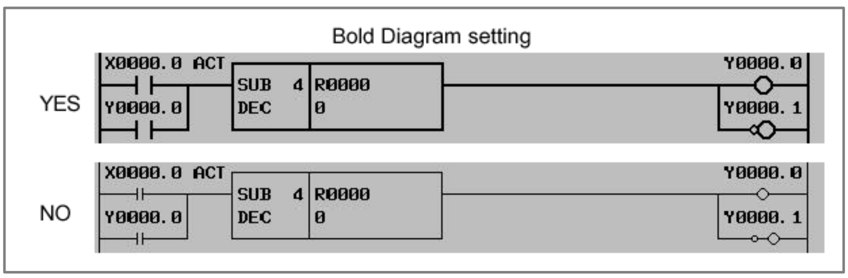
Sets thickness of diagram lines. This setting also affects shapes of relays.

YES (default)

Diagram is drawn with thick lines. Relays are drawn in more distinct shapes. You have to choose this option to make following ”VARIABLE RELAY SYMBOL” setting effective.

NO

Diagram is drawn with thin lines. Relays are drawn in smaller shapes.



NOTE
The setting "BOLD DIAGRAM" requires "Character registration function" of CNC software, which is supported by Series F002 version 03 or later. If CNC software does not support it, diagram is always displayed with thin lines.

ADDRESS COLOR
ADDRESS BACKGROUND

Colors for the relay address are set. You can specify these colors by entering color number, or by moving right and left cursor. 16 colors (from No.0 to No.15) are available. Foreground color should be different from background one.

DIAGRAM COLOR
DIAGRAM BACKGROUND

General color and its background color for LADDER Diagram.

SELECTED NET COLOR
SELECTED NET BACKGROUND

Color setting for the selected net. The nets that will be object of following editing operation are displayed with this color setting.

PROTECTED NET COLOR
PROTECTED NET BACKGROUND

Color setting for the nets protected from editing operations. The protected nets can not be selected as object of editing operations. The nets with following functional instructions are protected from deleting and creating when a subprogram (not whole program) is chosen to edit.

END1	END3	SP
END2	END	SPE

COMMENT COLOR
COMMENT BACKGROUND

Colors for the relay comment are set. You can specify these colors by entering color number, or by moving right and left cursor. 16 colors (from No.0 to No.15) are available. Foreground color should be different from background one.

(b) Soft keys

LADDER Diagram Editor Setting screen has the soft keys of options and following:

- [INIT] Initialize all settings

  All settings will be initialized to the default values.

6.2.4 Error Messages and Their Solutions

Pressing Return key [<] or [UPDATE] soft key will analyze edited LADDER program and try to generate executable objects. If some error is found at analyzing phase, you will see error message on screen, and according to kind of error, cursor may jump to the place where the error is found.

Following table shows error messages expected at LADDER analysis and their meanings and solutions.

Error message	Meaning and solution
CALL/CALLU IN BAD LEVEL.	Meaning: CALL/CALLU is used in wrong place. Solution: CALL/CALLU must be used in Level 2 or in subprograms. Do not use any other places.
COME IN JMP.	Meaning: COME is found between JMP and JMPE, and COM and corresponding COME have different JMP/JMPE status. Solution: COME and corresponding COM must have same JMP/JMPE status. Review COM range and JMP range, to adjust not to overlap with each other: it is possible that one range includes the other completely.
COME WITHOUT COM.	Meaning: There is no COM that corresponds to this COME. Solution: If COM is missing, add it in proper position. If the COME is unnecessary, remove it.
DUPLICATE END1. DUPLICATE END2. DUPLICATE END3.	Meaning: Multiple END1, END2, or END3 are found. Solution: Remove extra END1, END2, or END3.
DUPLICATE LBL.	Meaning: Same L–address is used in plural LBLs. Solution: If some of these LBLs are unnecessary, remove them. If all of these LBLs is necessary, assign other L–addresses to them to make all LBLs unique.
DUPLICATE P ADDRESS.	Meaning: Same P–address is used in plural SPs. Solution: If some of these SPs are unnecessary, remove them. If all of these SPs is necessary, assign other P–addresses to them to make all SPs unique.

Error message	Meaning and solution
DUPLICATE TMR NUMBER.(WARNING) DUPLICATE CTR NUMBER.(WARNING) DUPLICATE TMRB NUMBER.(WARNING) DUPLICATE DIFU/DIFD NUMBER.(WARNING)	Meaning: Plural TMRs, CTRs, TMRBs, DIFUs, or DIFDs have the same number as their parameter. This is warning. Solution: If some of them are unnecessary, remove them. If all of them are necessary, assign other number to parameter of them to make them unique. If two or more instructions with same parameter number will never be active simultaneously at one time, the LADDER program has a possibility to work correctly, however, it is recommended from safety and maintenance points of view, that all these instructions should have different parameter number with each other.
END IN COM. END1 IN COM. END2 IN COM. END3 IN COM.	Meaning: END, END1, END2, or END3 is found between COM and COME. Solution: If COME is missing, add it in proper position. If COM is unnecessary, remove it.
END IN JMP. END1 IN JMP. END2 IN JMP. END3 IN JMP.	Meaning: END, END1, END2, or END3 is found between JMP and JMPE. Solution: If JMPE is missing, add it in proper position. If JMP is unnecessary, remove it.
END IN SP.	Meaning: END is found between SP and SPE. Solution: If SPE is missing, add it in proper position. If END is in wrong place, move it to proper position.
GARBAGE AFTER END. GARBAGE AFTER END2. GARBAGE AFTER END3.	Meaning: There are some nets after END, END2, or END3, which will not be executed. Solution: Remove unnecessary nets, and move necessary nets to proper position so that they will be executed.
ILLEGAL TMR NUMBER. ILLEGAL CTR NUMBER. ILLEGAL TMRB NUMBER. ILLEGAL DIFU/DIFD NUMBER.	Meaning: TMR, CTR, TMRB, DIFU, or DIFD has parameter number that is out of range. Solution: If unnecessary, remove it. Assign correct number not to exceed the maximum number defined by each PMC model.

Error message	Meaning and solution
JMP/JMPE TO BAD COM LEVEL.	Meaning: JMP and corresponding JMPE have different COM/COME status. Solution: JMP and corresponding JMPE must have same COM/COME status. Review JMP range and COM range, to adjust not to overlap with each other: it is possible that one range includes the other completely.
JMPB OVER COM BORDER.	Meaning: JMP and its destination differ in COM/COME status. Solution: JMPB and its destination must have same COM/COME status. Review range of JMPB and COM range, to adjust not to overlap with each other: it is possible that one range includes the other completely.
JMPB OVER LEV-EL.	Meaning: JMPB jumps to different program level. Solution: JMPB can only jump to the same program level, or within a subprogram. If the JMPB is unnecessary, remove it. If LBL for the JMPB is missing, add it in proper position. If it should be JMPC, correct it.
JMPC IN BAD LEVEL.	Meaning: JMPC is used in other than subprogram. Solution: JMPC is used to jump from a subprogram to level 2. If the JMPC is unnecessary, remove it. If it should be JMPB or JMP, correct it.
JMPC INTO COM.	Meaning: JMPC jumps to LBL between COM and COME. Solution: LBL for JMPC must be located out of any COM and COME pair. If the JMPC is unnecessary, remove it. If the LBL is located wrong, move it to correct position. If the L–address of JMPC is wrong, correct it.
JMPE IN COM.	Meaning: JMPE is found between COM and COME, and JMP and corresponding JMPE have different COM/COME status. Solution: JMPE and corresponding JMP must have same COM/COME status. Review JMP range and COM range, to adjust not to overlap with each other: it is possible that one range includes the other completely.
JMPE WITHOUT JMP.	Meaning: There is no JMP that corresponds to this JMPE. Solution: If JMP is missing, add it in proper position. If the JMPE is unnecessary, remove it.

Error message	Meaning and solution
LADDER PRO-GRAM IS BRO-KEN.	Meaning: LADDER program may be broken by some reason. Solution: This LADDER program must be all cleared once, and remake LADDER program.
LBL FOR JMPB NOT FOUND.	Meaning: Can not find proper LBL for JMPB. Solution: If JMPB is unnecessary, remove it. If LBL is missing, add it in proper position.
LBL FOR JMPC IN BAD LEVEL	Meaning: Destination of JMPC is not level 2. Solution: JMPC is used to jump from a subprogram to level 2. If the JMPC is unnecessary, remove it. If another LBL of same L–address that the JMPC is intended to jump exists in the subprogram, assign different L–address to these two LBLs. If it should be JMPB or JMP, correct it.
LBL FOR JMPC NOT FOUND.	Meaning: Can not find proper LBL for JMPC. Solution: If JMPC is unnecessary, remove it. If LBL is missing, add it in proper position: JMPC jumps into level 2. If it should be JMPB or JMP, correct it.
MISSING COME FOR THIS COM.	Meaning: There is no COME that corresponds to this COM. Solution: If COME is missing, add it in proper position. If COM is unnecessary, remove it.
MISSING JMPE FOR THIS JMP.	Meaning: There is no JMPE that corresponds to this JMP. Solution: If JMPE is missing, add it in proper position. If JMP is unnecessary, remove it.
MISSING SPE FOR THIS SP.	Meaning: There is no SPE that corresponds to this SP. Solution: If SPE is missing, add it in proper position. If SP is unnecessary, remove it.
NO END. NO END1. NO END2. NO END3.	Meaning: END, END1, END2, or END3 is not found. Solution: Add END, END1, END2, or END3 in proper position.
NO SUCH SUB-PROGRAM.	Meaning: Subprogram that is called by CALL/CALLU is not found. Solution: If it calls wrong subprogram, correct it. If the subprogram is missing, create it.

Error message	Meaning and solution
NO WRITE COIL.	Meaning: Write coil is necessary, but is not found. Solution: Add proper write coil.
SP IN BAD LEV-EL.	Meaning: SP is found in wrong place. Solution: SP can be used at top of a subprogram. Correct it so that no SP exists in other place.
SP IN LEVEL3.	Meaning: SP is found in level 3. Solution: If END3 is located wrong, move it to correct position. If the SP is unnecessary, remove it.
SP/SPE IN COM.	Meaning: SP or SPE is found between COM and COME. Solution: If COME is missing, add it in proper position. If the COM is unnecessary, remove it.
SP/SPE IN JMP.	Meaning: SP or SPE is found between JMP and JMPE. Solution: If JMPE is missing, add it in proper position. If the JMP is unnecessary, remove it.
SPE WITHOUT SP.	Meaning: There is no SP that corresponds to this SPE. Solution: If SP is missing, add it in proper position. If the SPE is unnecessary, remove it.
TOO MANY LBL.	Meaning: There are too many LBLs. Solution: Remove unnecessary LBLs. If this error still occurs, adjust the construction of program to use less LBLs.
UNAVAILABLE INSTRUCTION.	Meaning: Unsupported instruction for this PMC model is found. Solution: Confirm that this LADDER program is correct one. If this program is correct one, all these unsupported instructions have to be removed.

6.2.5 How to Correct LADDER Program that Never Stops

If the following functional instructions are used with their ACT conditions (or RST conditions) are kept ON improperly, an attempt to stop LADDER may take much time, or may not stop it actually.

- WINDR / WINDW

  In case of window function of high–speed response, this will not cause the problem even if its ACT is stuck to ON.

- EXIN
- AXCTL

  This will cause the problem also when RST condition is stuck to ON.

- JMPB

  When it jumps to the LBL before JMPB itself.

- JMPC

  When it jumps to the LBL which leads to reach the same JMPC again.

When LADDER can not stop, any operation to modify LADDER program may take much time to complete, or may not complete. Processes such as followings will be affected by this problem:

1. Stop LADDER program by [STOP] soft key.
2. Read new LADDER program from memory card or other devices at I/O screen while current LADDER program is running.
3. Update running LADDER program at LADDER Diagram Editor screen, by exiting the screen with return key [<], or by [UPDATE] soft key.

To avoid this problem, find above–mentioned functional instructions whose ACT condition (or RST condition) is kept ON, and follow instructions below to correct LADDER program to use these functional instructions properly:

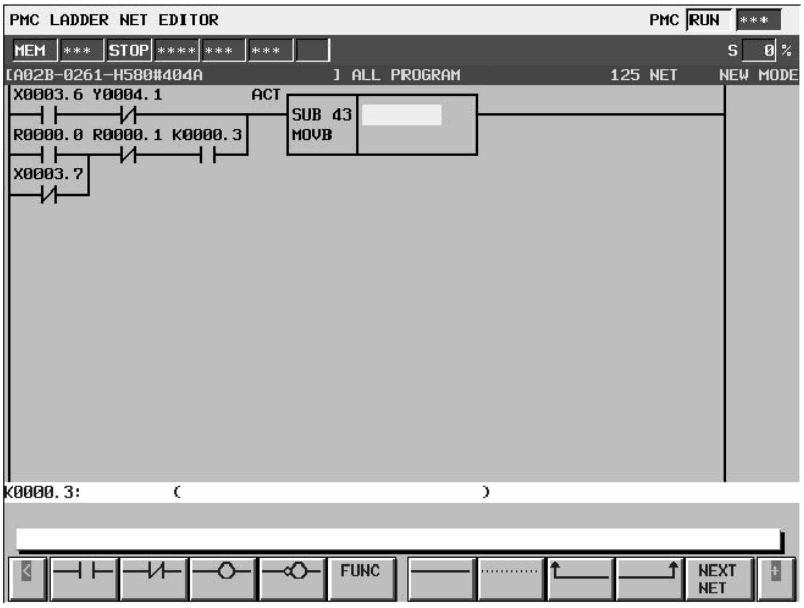
1. Confirm that machine is in safe condition, and then turn off CNC's power.
2. Turn on CNC's power, with "CAN" and "Z" keys keeping held down until CNC's finishes starting up completely, not to run LADDER program automatically.
3. Go to LADDER Diagram Editor screen, find the instruction that cause the trouble, modify logic around the instruction, so that ACT condition (or RST condition) will once turn OFF after every time the process of the instruction completes. In case that JMPB or JMPC repeats same process, examine the condition to jump, and reconstruct the structure of LADDER if necessary.
4. Write LADDER program into flash ROM at I/O screen.
5. Run LADDER program.

If the problem still remains after correction, there may be another functional instruction that causes the trouble in the same way. Repeat finding and correcting them in the same way, until the trouble is resolved.

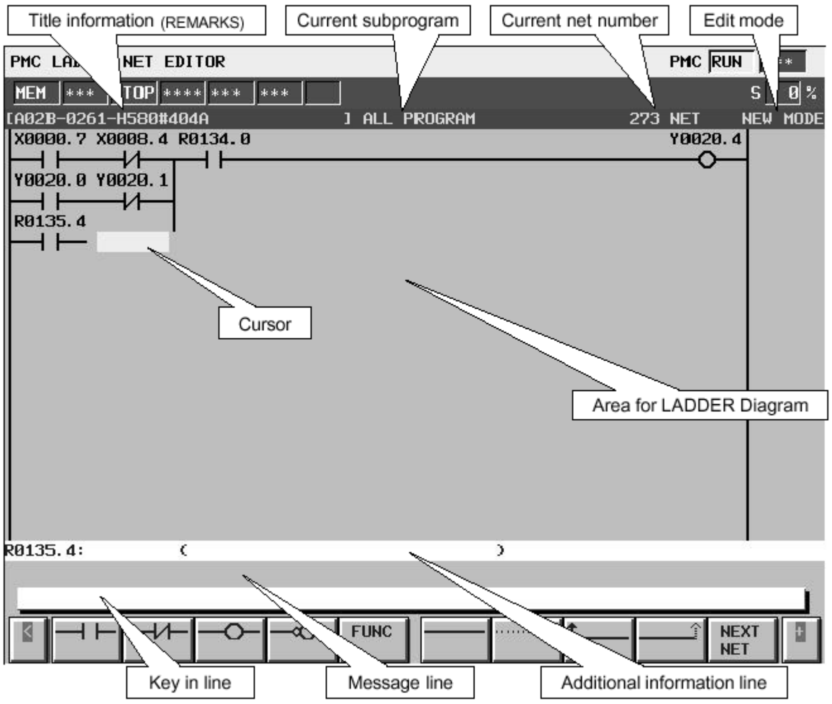
6.3 Net Editor Screen

At Net Editor screen, you can create new net, and modify existing net. When [ZOOM] soft key is used to reach Net Editor screen, this screen is in ”Modify Mode” to modify existing net. When [NEW NET] soft key is used, this screen is in ”New Mode” to create new net from nothing. Following operations are available at this screen:

- Place new contacts and coils ”bit address” + [—| |—], [—O—], etc.
- Change type of contacts and coils [—| |—], [—O—], etc.
- Place new functional instructions [FUNC]
- Change type of functional instructions [FUNC]
- Erase contacts, coils, and functional instructions [- - - - -]
- Draw/erase connecting lines [———], [↑———], [———↑]
- Edit data table of functional instructions [DATA TABLE]
- Insert line/column [INSERT LINE], [INSERT COLUMN], [APPEND COLUMN]
- Change address of contacts and coils ”bit address” + INPUT key
- Change parameters of functional instructions ”number” or ”byte address” + INPUT key
- Abandon modifications [RESTORE]



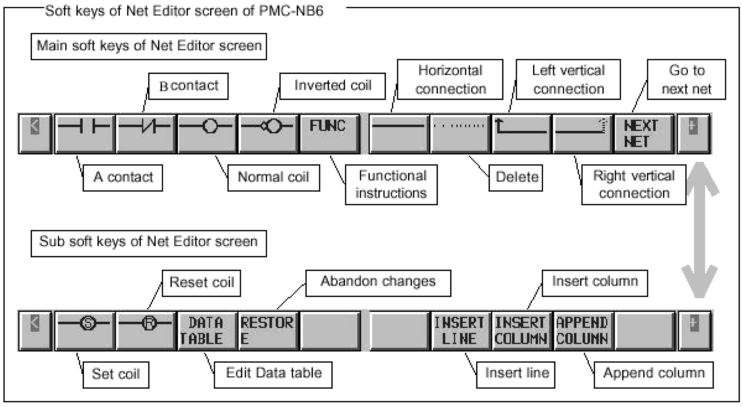
6.3.1 Screen Structures



(a) Screen structure

1. It is basically same with LADDER Diagram Editor screen, except that only one net is in this screen, and that position bar at right edge of screen does not appear at this screen.
2. Current edit mode is indicated at right of the top line as ”NEW MODE” or ”MODFY MODE”. When [ZOOM] soft key at LADDER Editor screen is used to reach Net Editor screen, the screen is in Modify mode, and when [NEW NET] soft key is used, it is in New mode.
3. Current net number is displayed at right of the top line. The net number is same with the net number in previous LADDER Diagram Editor screen.
4. Net Editor screen expands image of net horizontally for a wider net according to its width, while LADDER Diagram Monitor/Editor screen folds nets wider than screen width. When net width is expanded over screen width, attempt to move cursor out of screen will scroll net image to the direction.
   The net of maximum size occupies area of 1024 elements, but actually available area may be little less for internal use according to the internal condition: ”element” means the space that is occupied by single relay.

6.3.2 Operations

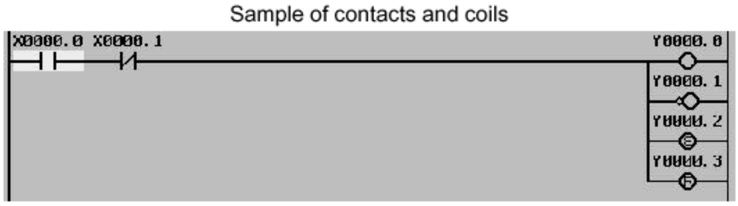


(a) Operation with Soft–keys

1. [—| |—],[—|/|—],[—O—],[—⊘—],[—Ⓢ—],[—Ⓡ—]

   Place relays (contacts and coils), or change type of existing relays.
   When one of these relay soft keys is pressed at cursor on blank place, new relay of the soft key is placed under the cursor. When the soft key follows a string that means a bit address, the bit address is assigned to the newly placed relay. If no bit address is given, last entered bit address is automatically used for the new relay. If no bit address has been entered yet, the new relay will have no address assigned to it. Contacts can be placed at other than rightmost column, and coils can be placed at rightmost column only.
   Moving cursor onto an existing relay, pressing a relay soft key of different type changes the type of relay under the cursor. But, changing coil to contact, and changing contact to coil are forbidden.

Sample of contacts and coils



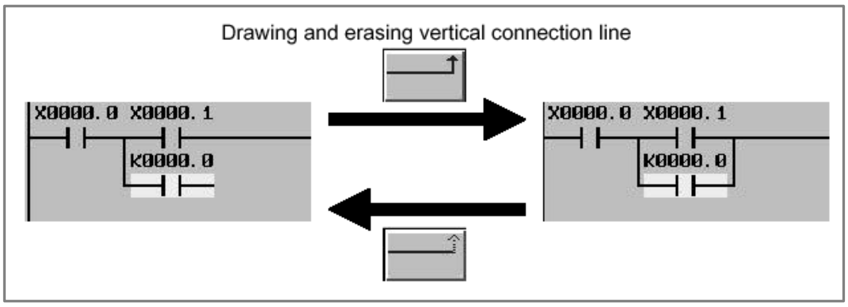
2. [FUNC] Enter and change functional instruction

   Places functional instruction, or changes type of existing functional instruction.
   When [FUNC] soft key is pressed at cursor on blank place, new functional instruction will be placed under the cursor: list of available functional instructions is displayed, then choose type of functional instruction to be entered. When [FUNC] soft key follows a string that means number or name of a functional instruction, the specified functional instruction is entered directly, without the list screen.
   Moving cursor onto an existing functional instruction, pressing [FUNC] soft key changes the type of functional instruction under the cursor.

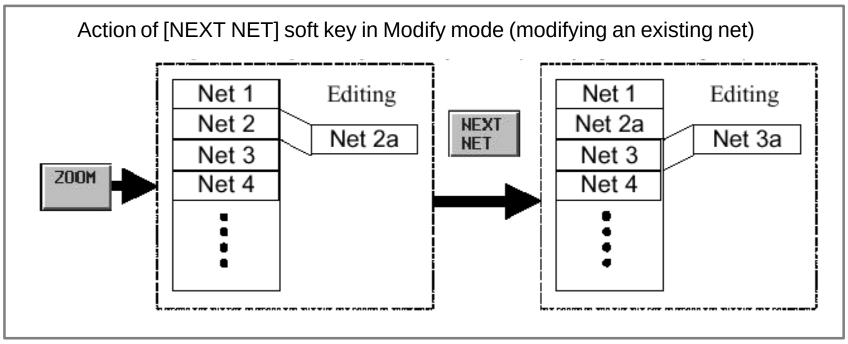
3. [———] Draw horizontal connection

Draws horizontal connection line. Or alters an existing relay to horizontal line.

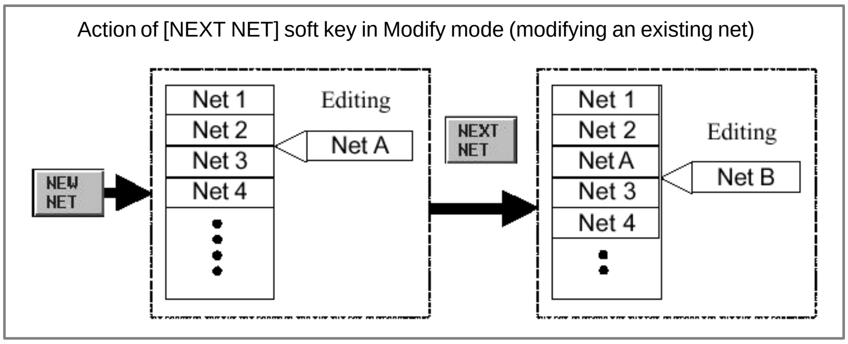
4. [- - - - -] Erase relays and functional instructions
   Erases relays and functional instructions under cursor.
5. [🡡———],[———🡡] Draw and erase vertical connection
   Draw vertical connection line upward from right or left edge of relay or horizontal line under cursor. Or erase existing vertical lines.
   If the relay or line under the cursor has no vertical line upward, these soft keys have solid arrows, and indicate that pressing them means drawing lines. On the other hand, if a vertical line already exists under the cursor, arrows in these soft keys become pale ([⇡———], [———⇡]), and indicate that pressing them means erasing lines.



6. [NEXT NET] Go to next net
   Finishes editing current net, and goes to next net.
   If [ZOOM] soft key at LADDER Diagram Editor screen is used to reach Net Editor screen, [NEXT NET] will finish modifying current net, and the next net will be an object of further editing operation.



If [NEW NET] soft key at LADDER Diagram Editor screen is used to reach Net Editor screen, [NEXT NET] will finish creating current net, insert it into the LADDER program, and start with blank to create another new net to be inserted next to the current net.



7. [DATA TABLE] Edit data table

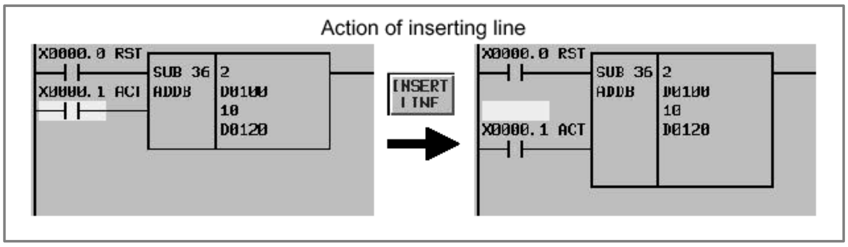
   Reaches Functional Instruction Data Table Editor screen to edit data table of functional instruction under cursor. This soft key appears only when cursor is on a functional instruction that has data table with it.
   For detail of editing operation of data table, see descriptions of ”Functional Instruction Data Table Editor screen”.

8. [RESTORE] Abandon changes

   Abandons all changes, and restores net to the one at starting editing this net. If [NEW NET] soft key at LADDER Diagram Editor screen is used to reach Net Editor screen, it will be back to blank net, and if [ZOOM] soft key is used, it will be back to the old net before modifications in this screen.

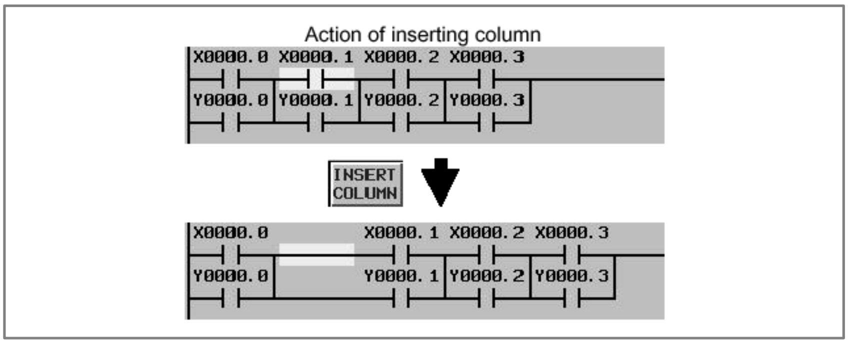
9. [INSERT LINE] Insert line

   Inserts one blank line at cursor position. Diagram elements at or below vertical cursor position will be shifted downward by one line. Inserting line at middle of functional instruction box will expand the box vertically to make a space between the input conditions.



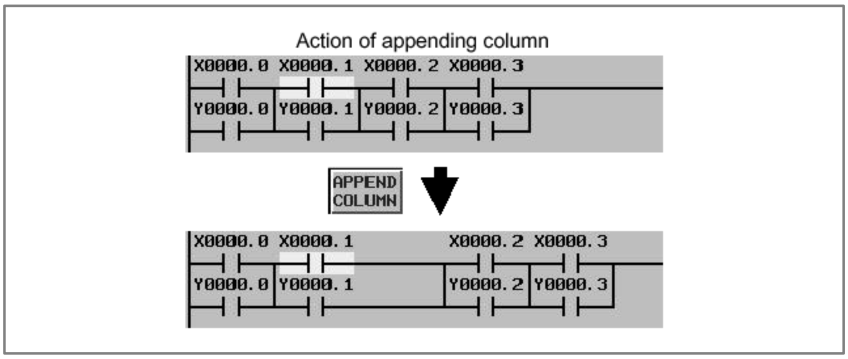
10. [INSERT COLUMN] Insert column

    Inserts one blank column at cursor position. Diagram elements at or on right of horizontal cursor position will be shifted to right by one column. And if there is no room to shift the elements, a new column is added and the Diagram area will be expanded to right.



11. [APPEND COLUMN] Append column

    Inserts one blank column at right of cursor position. Diagram elements on right of horizontal cursor position will be shifted to right by one column. And if necessary, net will be expanded to right.



12. [<] Exit editor screen

    Analyzes current editing net, and store it into LADDER program. If some error is found in the net, it still remains Net Editor screen, and an error message will be displayed. According to a kind of error, cursor may indicate where the error is detected.

(b) Other operations

1. Cursor move keys, Page change keys

   Cursor move keys and Page change keys move cursor on screen. Net Editor screen expands image of net horizontally for a wider net according to its width, while LADDER Diagram Monitor/Editor screen folds nets wider than screen width. When net width is expanded over screen width, attempt to move cursor out of screen will scroll net image to the direction.
   The net of maximum size occupies area of 1024 elements, but actually available area may be little less for internal use according to the internal condition: ”element” means the space that is occupied by single relay.

2. ”bit address” + INPUT key

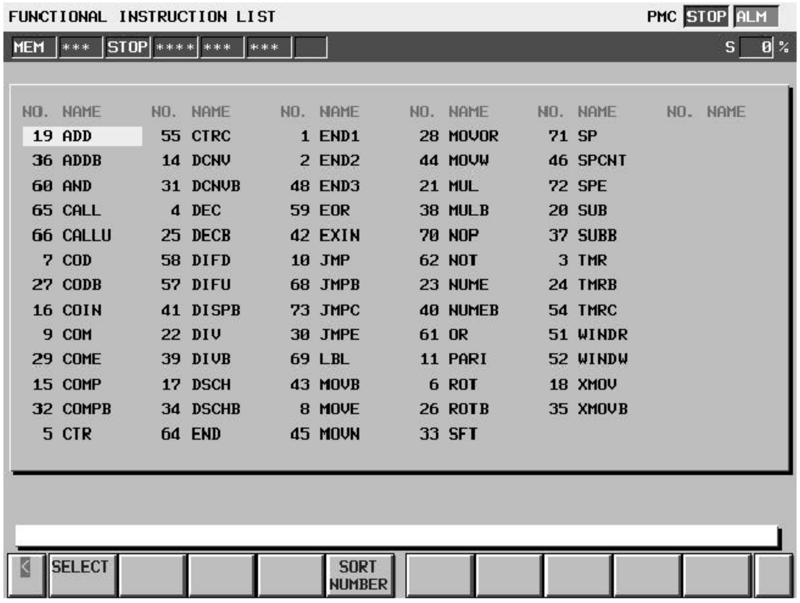
   Changes bit address of relay under cursor.

3. ”number” / ”byte address” + INPUT key

   Changes parameter of functional instructions under cursor.

6.3.3 Functional Instruction List Screen

Pressing [FUNC] soft key at Net Editor screen reaches Functional Instruction List screen at which you choose a functional instruction to be entered from list of all available functional instructions.



Operations at this screen are below:

(a) Operation with Soft–keys

1. [SELECT] Select a functional instruction

   Selects a functional instruction. The functional instruction under cursor at that time is chosen, and entered into the editing net.

2. [SORT NUMBER], [SORT NAME] Rearrange functional instructions list

   Rearrange functional instructions list in two ways. [SORT NUMBER] soft key arranges the list in numerical order with their identifying numbers, on the other hand, [SORT NAME] soft key arranges it in alphabetical order with their names. At beginning, the list is arranged in alphabetical order.

3. [<] Quit selecting

   Quits selecting functional instruction, and return to Net Editor screen.

(b) Other operations

1. Cursor move keys

   Cursor move keys move cursor on screen. According to the cursor position, the functional instruction to be selected changes.

2. INPUT key

   Act just like [SELECT] soft key.

(c) Shortcuts

1. [SELECT] soft key and INPUT key following number or name of a functional instruction will select the specified functional instruction directly, instead of the one under cursor.

2. When [FUNC] soft key in Net editor screen is pressed following a string that means number or name of a functional instruction, the specified functional instruction is entered directly, without displaying Functional Instruction List screen.

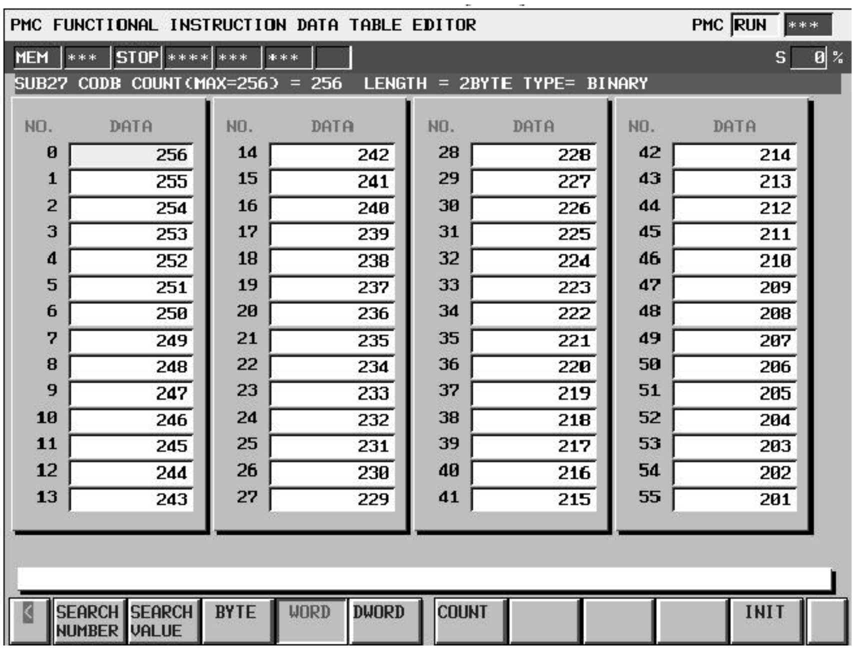
6.3.4 Functional Instruction Data Table Editor Screen

At Functional Instruction Data Table Editor screen, you can edit the contents of data table that belongs to some functional instructions.
To reach this screen, at Net Editor screen, press [DATA TABLE] soft key that is displayed when the cursor is on the following functional instructions which have a data table.

- Functional Instruction COD (SUB7)
- Functional Instruction CODB (SUB27)

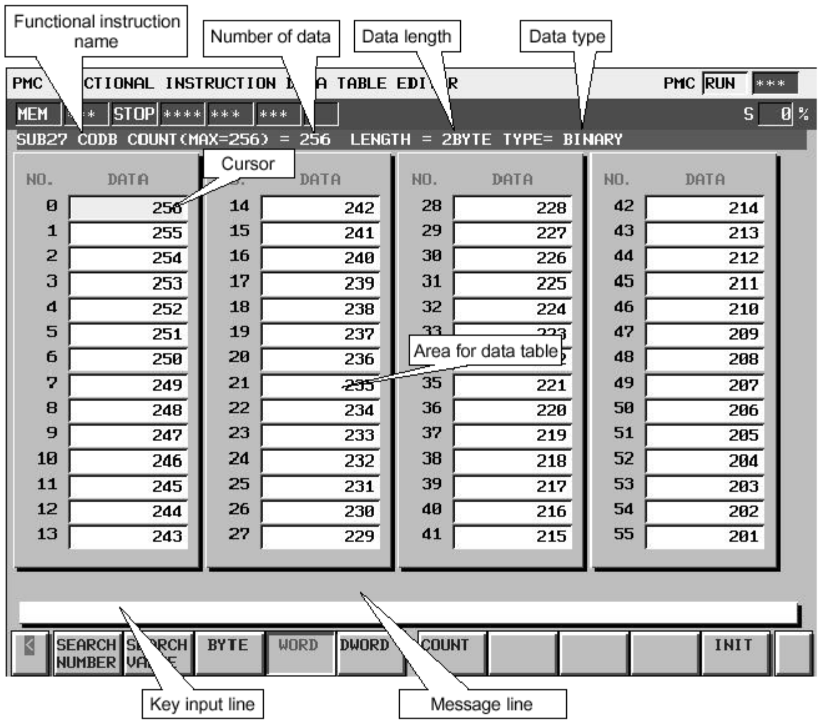
Following edit operations are available at this screen.

- Change the data table value. ”number” + INPUT key
- Change the data length. [BYTE], [WORD], [D.WORD] (These soft keys can be operated only at Functional Instruction Data Table Editor screen of Functional Instruction CODB.)
- Change the number of data. [COUNT]
- Initialize all of data [INIT]



Functional Instruction Data Table Editor screen of Functional Instruction CODB.

(1) Screen Structures



(a) Screen Structure

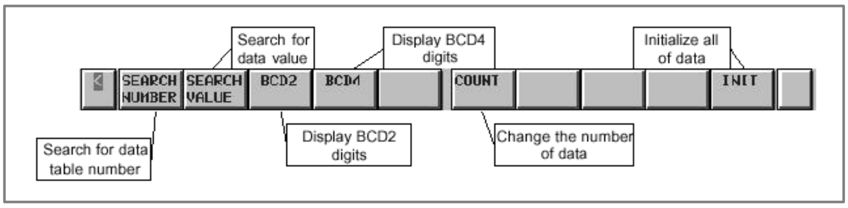
It is same with Functional Instruction Data Table Viewer screen.

(b) Display data

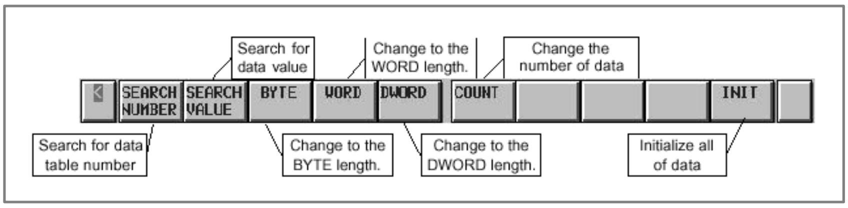
The cursor is shown always. You can edit the data that is pointed by the cursor.

(2) Operations

Soft keys of Functional Instruction Data Table Editor screen of functional instruction COD.



Soft keys of Functional Instruction Data Table Editor screen of functional instruction CODB.



1. [SEARCH NUMBER] Search for data table number

   Searches the data table number which you specified.

2. [SEARCH VALUE] Search for data value

   Searches the data value which you specified.

3. [BCD2] Display BCD2 digits

   Changes the display data type to 2 digits of BCD. This operation only switches the display data type. So it does never edit the data on memory.

4. [BCD4] Display BCD4 digits

   Changes the display data type to 4 digits of BCD. This operation only switches the display data type. So it does never edit the data on memory.

NOTE

In the functional instruction COD (SUB7), the data type of the data table can dynamically be changed either BCD2 or BCD4 by ”BYT” which is one of input condition. So the data type of the data table is decided when the functional instruction COD is executed. Decide the display data digit according to the status of ”BYT” by pressing either [BCD2] soft key or [BCD4] soft key.
After turning the power on, the default displaying data type is BCD4 digits. But if you change data type by pressing [BCD2] soft key or [BCD4] soft key, the data type is kept until you change again.
The data table of functional instruction COD is stored in the memory as BCD4 digits type.
If you change the data type from BCD4 digits to BCD2 digits, the data is displayed without higher 2–digits. But the data of higher 2–digits is kept in the memory. So you return the data type from BCD2 digits to BCD4 digits, the former BCD4 digits is recovered.
The input range of the data obeys the current data type.

5. [BYTE] Change to the BYTE length

   Changes data length to 1 byte. If overflowed data is found, the cursor points it, and this operation aborts. Then, correct it and press [BYTE] soft key again.

6. [WORD] Change to the 2 BYTE length

   Changes data length to 2 byte. If overflowed data is found, the cursor points it, and this operation aborts. Then, correct it and press [WORD] soft key again.

7. [WORD] Change to the 4 BYTE length

   Changes data length to 4 byte.

NOTE

On functional instruction CODB, the data type is decided by the first parameter of it.
So, if you change data type, the first parameter is changed too.
When you added functional instruction CODB to ladder program, the default data type is BYTE.

8. [COUNT] Change the number of data
   Changes the number of data. If you expanded the number of data, ”0” is set to expanded data as default.

NOTE
In case of functional instruction COD, the number of data is decided by the first parameter of it.
In case of functional instruction CODB, the number of data is decided by the second parameter of it.
If you change the number of data, these parameters are changed too.

9. [INIT] Initialize all of data
   Initializes all of data to ”0”. The number of data is not changed.
10. Cursor move keys, Page change keys
   You can move the cursor by all cursor move keys and Page change keys.
11. ”number” + INPUT key
   Changes the data that is pointed by the cursor.
   The input range of the data obeys the data length and the display data type.
   Ex.) The case of functional instruction COD and displaying BCD2 digits
   The available data range: 0 to 99
   Ex.) The case of functional instruction CODB and length of 2 BYTE
   The available data range: –32768 to 32767

   And you can input multiple numbers by the following methods.
   (1) ”;”(EOB) is used for separating data.
   (Ex.) Press the INPUT key after typing ”100;200;300;”
   (2) ”;=” is used for inputting the same value as preceding data.
   (Ex.) Press the INPUT key after typing ”100;=;=;200;=”, and it becomes ”100,100,100,200,200”.
   (3) ”;;” is used for skipping an input address.
   (Ex.) Press the INPUT key after typing ”100;;100”. The second data is not inputted.

6.3.5 Error Messages and Their Solutions

Return key [<] or [NEXT NET] soft key analyzes current editing net, and produce executable object of the net: new object of LADDER program is not yet effective at this time. If some error is found through the analysis, an error message is displayed, and cursor may indicate where the error is detected, according to type of error.

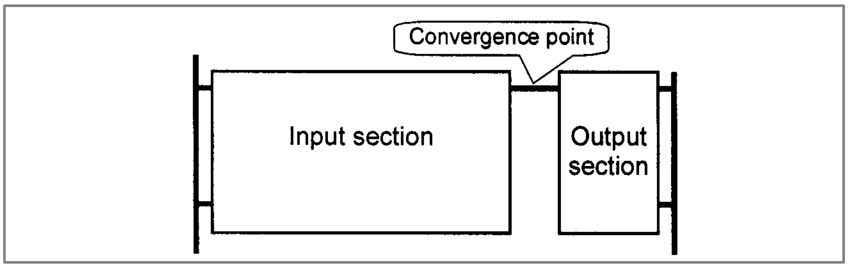
Following table shows error messages expected at net analysis and their meanings and solutions.

Error message	Meaning and solution
ALL COIL MUST HAVE SAME IN-PUT	Meaning: When a net contains more than one coil, the coils should not have any contact beside them that affects only one of the coils. Solution: Left terminals of all coils in a net must be connected to same input point.
BAD COIL LOCA-TION	Meaning: Coil is located in bad position. Solution: Coil can be located only at rightmost column. Any coil located at other place must be erased once, and place necessary coils in correct place.
BAD CONDITION	Meaning: Some condition input of functional instruction is not connected correctly. Solution: Check the connection of all condition inputs of the functional instruction. Especially for functional instruction that has more than one condition input, check if connections to condition inputs interfere with each other.
FUNCTION AF-TER DI-VERGENCE IS FORBIDDEN	Meaning: Functional instruction is used in output section of net: refer to next section for detail. Solution: Functional instruction can not be used in output section of net. If necessary, divide the net into plural nets.
NET IS TOO COM-PLICATED	Meaning: Net is too complicated to analyze. Solution: Examine every connection, and find unnecessarily bending connection, or coils that are connected to different point.
NO CONNECTION	Meaning: There is signal connected to nowhere. Solution: Find gap that is expected to be connected, and correct the connection.

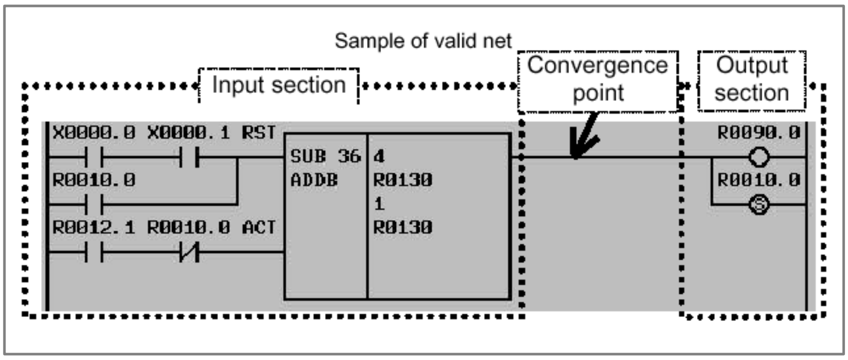
Error message	Meaning and solution
NO INPUT FOR OPERATION	Meaning: No signal is provided for logical operation. Solution: Coil without input, or coil connected to output of functional instruction that has no output, causes this error. If coil is not necessary, remove it. If necessary, connect it to meaningful input.
OPERATION AFTER FUNCTION IS FORBIDDEN	Meaning: No logical operation with functional instruction output is permitted, except write coils. Solution: Output of functional instruction can not be connected to a contact, nor to conjunction with other signal that will be implemented by logical–or operation.
PARAMETER IS NOT SUPPLIED	Meaning: Relay with blank address, or blank parameter of functional instruction, is found. Solution: Enter all of the relay addresses, and parameters of functional instructions.
SHORT CIRCUIT	Meaning: Some contacts are connected with short circuit. Solution: Find contact with terminals connected by short circuit, and correct connections.
TOO MANY FUNCTIONS IN NET	Meaning: Too many functional instructions are in one net. Solution: Only one functional instruction is allowed to constitute a net. If necessary, divide the net into plural nets.
WRITE COIL EXPECTED	Meaning: Write coil is expected, but not found. Solution: Add proper write coil to the net.

6.3.6 Structure of Valid Net

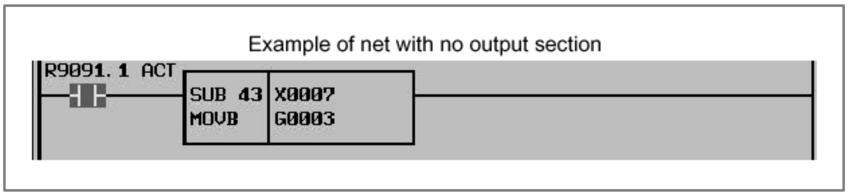
Valid net for PMC–NB6 must have following structure:



”Input section” consists of contacts and functional instruction, and the result of operations of input section is led to ”Convergence point”. After the convergence point, there is ”Output section” that consists of coils only. The ”Convergence point” is the nearest point to right power line, where all connections join with each other to gather into single connection.



Input section contains at least one relay or functional instruction, however, output section may contain nothing.



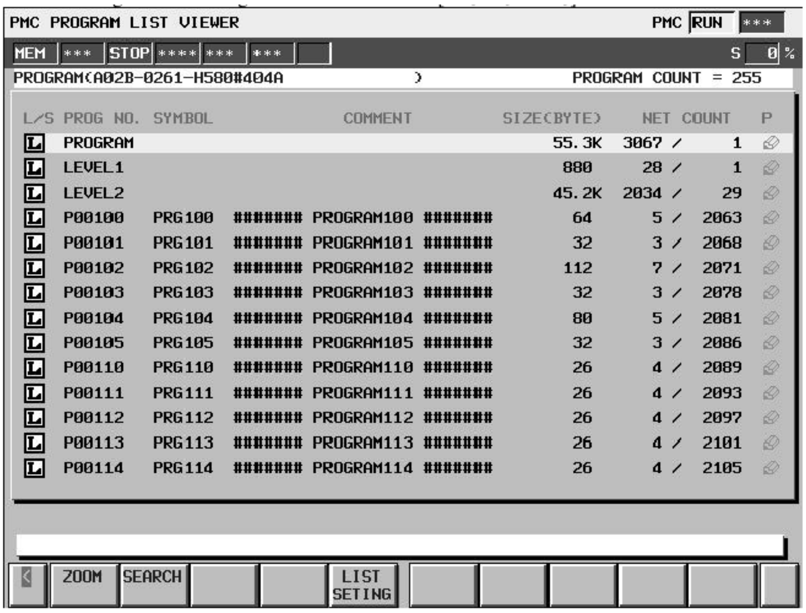
Valid net for PMC–NB6 is also restricted in following rules:

- Only one functional instruction is available for a net.
- Functional instruction can be placed only at last (rightmost) of input section.
- Only coils can be contained in output section.

6.4 PROGRAM LIST VIEWER SCREEN

Program List Viewer screen shows the list of subprogram of the Ladder. You can select one subprogram then the content of program is displayed. This screen also shows the detail information (program type, symbol name, comment, program size, program net count, the start net number in all of the program and protect condition) for subprogram.
To reach this screen, at LADDER Diagram Monitor screen, press [LIST] soft key.
Following operations are available at this screen.

- Show the content of selected program. [ZOOM]
  (Go to LADDER Diagram Monitor screen)
- Search for program. [SEARCH]
- Go to Program List Setting screen. [LIST SETING]



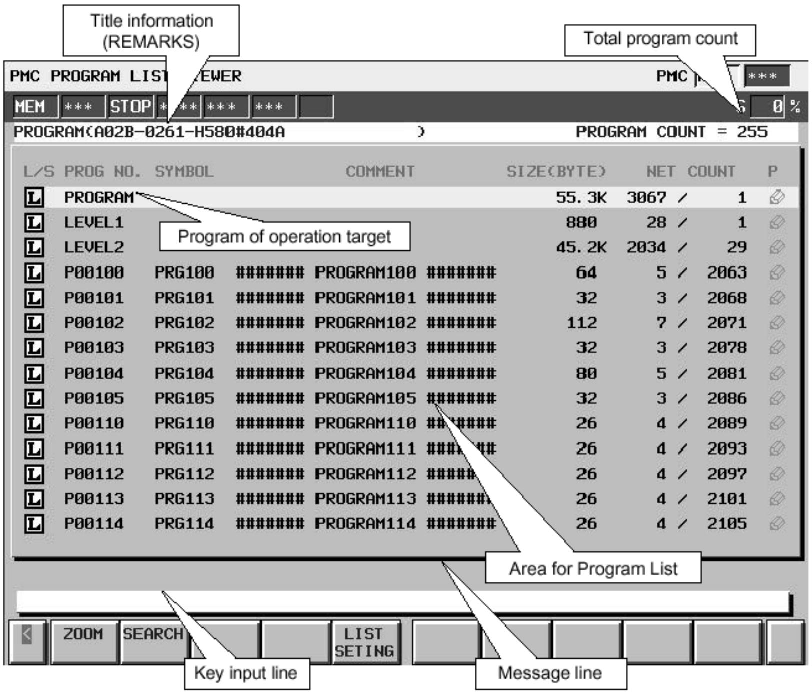
Program List Viewer screen (Detail)

You can select Detail viewer format or Brief viewer format on Program List Viewer screen.
The default viewer format is Detail viewer format.
See "Setting screen for Program List Viewer screen" section for more detail.

6.4.1 Screen Structures

(1) Program List Viewer screen (Detail)

Program List Viewer screen (Detail) shows the detail information that are program size, program net count and so on for program.



(a) Screen Structure

1. The title information (REMARKS) of the Ladder Program and the total program count are displayed above the Program List.
2. In the message line, error messages or inquiry messages will be displayed depending on the situation.
3. In the area for Program List, 14 data can be displayed at maximum.

(b) Area of Program List

1. The icon ”L” means program type of Ladder. It is displayed in the ”L/S” field for each program.

> **NOTE**
> Displaying L requires ”Character registration function” of CNC software, which is supported by Series F002 version 03 or later. If CNC software does not support it, Ladder program type is always displayed ”L”.

2. Program name is displayed in the ”PROG NO.” field for each program.
   There are three kinds of program names.
   PROGRAM : It means the whole program.
   LEVEL*n* (*n*=1,2,3) : It means the Ladder level 1, 2 and 3.
   P*m* (*m*=1 to 2000) : It means subprogram.
3. The symbol is displayed in the ”SYMBOL” field for each program.
   If no symbol is set into the subprogram, the ”SYMBOL” field will be blank.

4. The comment is displayed in the ”COMMENT” field for each program.

   If no comment is set into the subprogram, the ”COMMENT” field will be blank.

5. The program size is displayed in the ”SIZE (BYTE)” field for each program.

   If the program size is not over 1024 byte, the unit is shown in byte. If it is over 1024 byte, the unit is shown in kilo (1024) byte with ”K”.

   Ex.) The case that program size is not over 1024 byte.
   1023 bytes : ”1023” is shown.

   Ex.) The case that program size is over 1024 byte.
   20000 bytes : ”19.5K” is shown.

6. In the ”NET COUNT” field for each program, total number of nets in the program and first net number of the program in the whole Ladder program are displayed as follows.

   Total number of nets in the program* / First global net number of the program in the whole Ladder program*
   * Maximum is 99,999.

7. The status of protection is displayed in the ”P” field for each program.

   Following icons mean the status of protection.

   (Lock) : Monitoring and editing program are disabled.
   (Magnifying glass) : Monitoring program is enabled. Editing program is disabled.
   (Pencil) : Monitoring and editing program are enabled.

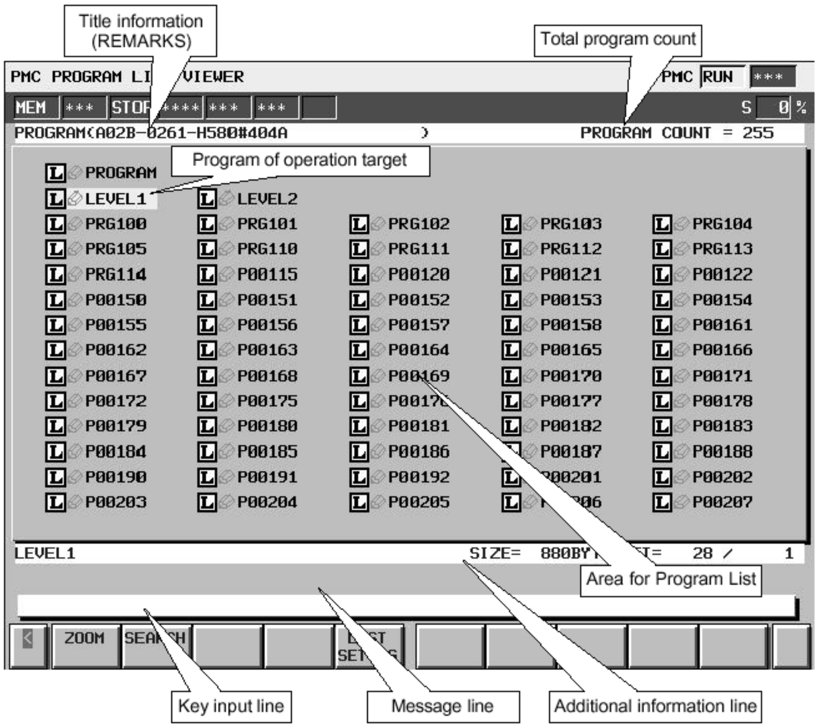
NOTE

The displaying icons requires ”Character registration function” of CNC software, which is supported by Series F002 version 03 or later. If CNC software does not support it, the statuses of protection are displayed as follows.

P	:	Monitoring and editing program are disabled.
R	:	Monitoring program is enabled. Editing program is disabled.
Space	:	Monitoring and editing program are enabled.

(2) Program List Viewer screen (Brief)

Program List Viewer screen (Brief) shows less information than Program List Viewer screen (Detail), to increase the number of items. The program type, the status of protection and the name or symbol are displayed for each program.
The detail information of program that is pointed by the cursor is displayed on additional information line.



(a) Screen Structure

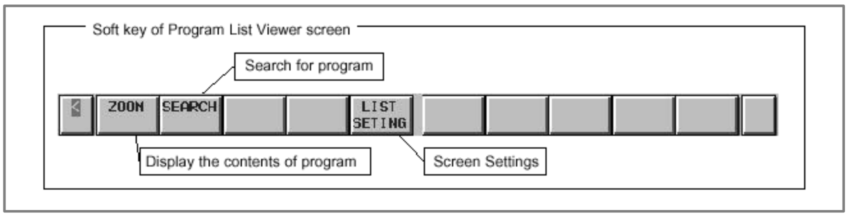
1. The title information (REMARKS) of the Ladder Program and the total program count are displayed above the Program List.
2. In the message line, error messages or inquiry messages will be displayed depending on the situation.
3. In additional information line near the bottom of the screen, the following information is displayed for the program under the cursor.
   - program name
   - symbol, comment
   - program size
   - total number of nets
   - First global net number of the program in the whole Ladder program.
4. In the Program List area, the programs 14 lines and 5 columns can be displayed in maximum.

(b) Area of Program List

The following items are displayed in Program List area of Program List Viewer screen.
The meanings of these items are the same as the items of Program List Viewer screen (Detail).

- Program type.
- Status of protection
- Program name or symbol. (See ”Screen Setting for Program List Viewer screen” section for selecting this item.)

6.4.2 Operations



(a) Operation with Soft–keys

1. [ZOOM] Display the contents of program

   Goes to LADDER Diagram Monitor screen. If you press [ZOOM] soft key without strings, the program under the cursor is displayed at LADDER Diagram Monitor screen.
   If you entered program name (See (c) 1. for detail) or symbol before pressing [ZOOM] soft key, the program according to the preceding string is searched and the program is displayed at LADDER Diagram Monitor screen.
   But, when the selected program is protected to monitor, you have to unlock the protection.

2. [SEARCH] Search for program

   Searches the program. If you entered program name (See (c) 1. for detail) or symbol and press [SEARCH] soft key, the program according to the preceding string is searched, the cursor points the program.

3. [LIST SETING] Screen Settings

   Goes to setting screen for Program List Viewer screen. You can change various settings for Program List Viewer screen at the screen. Use Return key [<] to return to Program List Viewer screen. See ”Settings for Program List Viewer screen” section for the detail.

(b) Other operations

1. Cursor move keys, Page change keys

   You can move cursor by all cursor move keys and Page change keys.
   And if you entered program name (See (c) 1. for detail) or symbol and press Right cursor move key, the program according to the preceding string is searched, the cursor points the program.

2. INPUT key

   You can operate same as [ZOOM] soft key.

(c) Note of searching operations

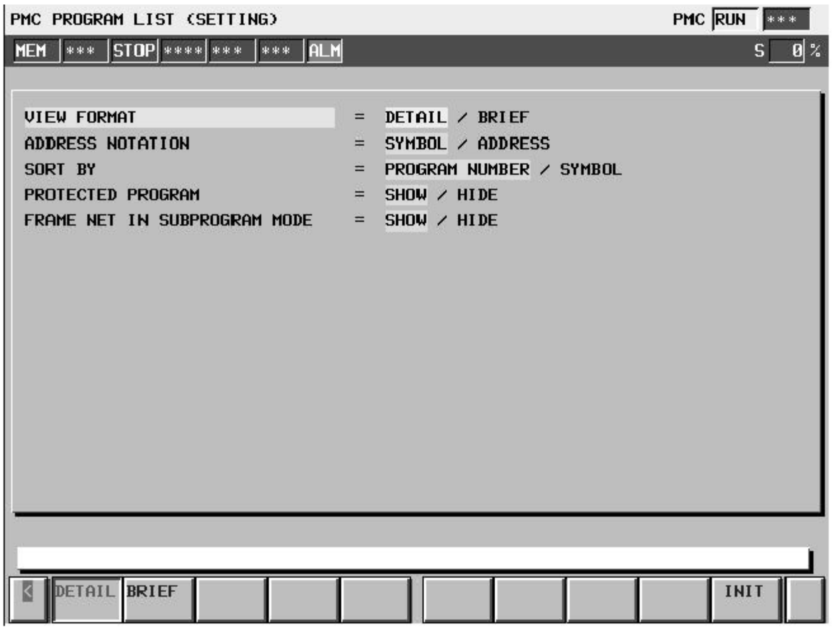
1. When program is searched by program name, the strings that show each program are as follows.

PROGRAM	: ”0”(Zero) or ”G”
LEVEL1, 2, 3	: ”L” + Number Ex.) ”L1”, ”L01”, etc
Pn	: Number or ”P” + Number Ex.) ”1”, ”P1”, ”P01”

2. Search function by [SEARCH] soft key or Right cursor move key tries to suppose the given word as an item to be searched in following order.

(i)	The string for PROGRAM or LEVEL	: ”0”(Zero), ”G”, ”L” + Number
	The number for subprogram	: Number
(ii)	Symbol	
(iii)	The string for subprogram	: ”P” + Number

6.4.3 Setting Screen



(a) Setting items

Program List Viewer/Editor screen contains the setting items below:

- VIEW FORMAT

  Determines whether the list data in the Program List Viewer/Editor screen are displayed as detail format, or brief format themselves.

  DETAIL (default)

  The Program List Viewer/Editor screen shows Detail information for each program.
  The items of detail information are program type, program name, symbol, comment, program size, program net count and status of protection.

  BRIEF

  The Program List Viewer/Editor screen shows less information than DETAIL, to increase the number of items.
  The program type, the status of protection and the name or symbol are displayed for each program.
  The detail information of program that is pointed by the cursor is displayed on additional information line.

- ADDRESS NOTATION

  Determines whether the programs in the Program List Viewer/Editor screen are displayed as corresponding symbols, or the addresses themselves.

  SYMBOL (default)

  Addresses that have a symbol are displayed by the symbols.
  Addresses without symbols are displayed by the addresses themselves.

  ADDRESS

  All addresses are displayed as the addresses themselves even if they have a symbol.

- SORT BY

  Determines whether the programs in the Program List Viewer/Editor screen are displayed in order of program number, or symbol name themselves. And When ADDRESS NOTAION is SYMBOL, this determination is enabled. So if ADDRESS NOTAION is ADDRESS, the programs are always displayed in order of program number.

  PROGRAM NUMBER (default)

  The programs are displayed in order of program number.

  SYMBOL

  Programs that have a symbol are displayed in order of the symbol name. Programs without symbols are displayed in order of program number after programs with the symbols. PROGRAM, LEVEL1, LEVEL2, and LEVEL3 are out of target of sort.

- PROTECTED PROGRAM

  Determines whether protected programs in the Program List Viewer/Editor screen are displayed or not. On each screen, the protected program means as follows.

  On Program List Viewer screen : the program is protected to monitor.

  On Program List Editor screen : the program is protected to edit.

  SHOW (default)

  The protected programs are displayed in the Program List Viewer/Editor screen.

  HIDE

  The protected programs are not displayed in the Program List Viewer/Editor screen.

- FRAME NET IN SUBPROGRAM MODE

  Frame net means functional instruction END1, 2 and 3 on LEVEL 1,2,3, and functional instruction SP and SPE on subprogram.

  It determines whether the frame net in the LADDER Diagram Monitor/Editor screen are displayed or not, when you selected the program and press [ZOOM] soft key in the Program List Viewer/Editor screen.

  SHOW (default)

  The frame net is displayed in the LADDER Diagram Monitor/Editor screen.

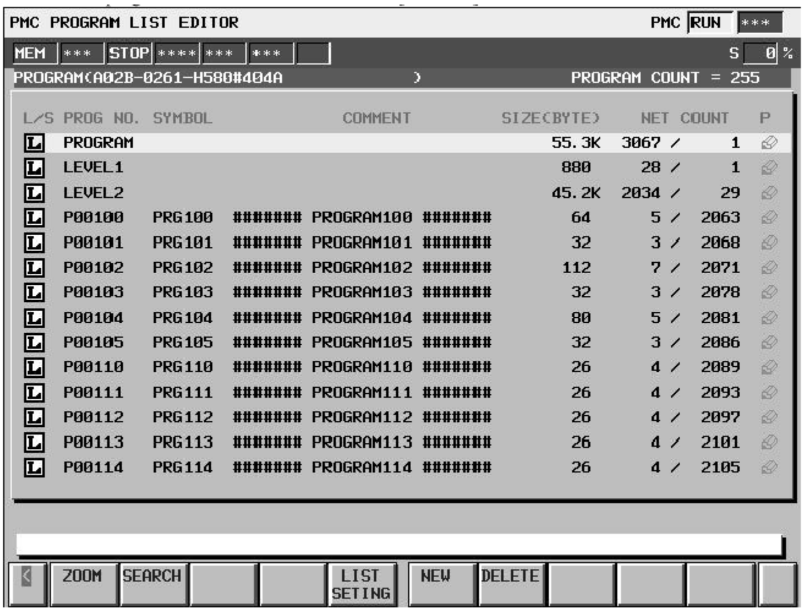
  HIDE

  The frame net is not displayed in the LADDER Diagram Monitor/Editor screen.

6.5 PROGRAM LIST EDITOR SCREEN

At Program List Editor screen you can create new program and delete a program in addition to the function of Program List Viewer screen.
To reach this screen, press [LIST] soft key at LADDER Diagram Editor screen.
Following operations are available at Program List Editor screen. For more detail of these operations, refer to the descriptions of each key to operate.

- Create new program [NEW]
- Delete a program [DELETE]



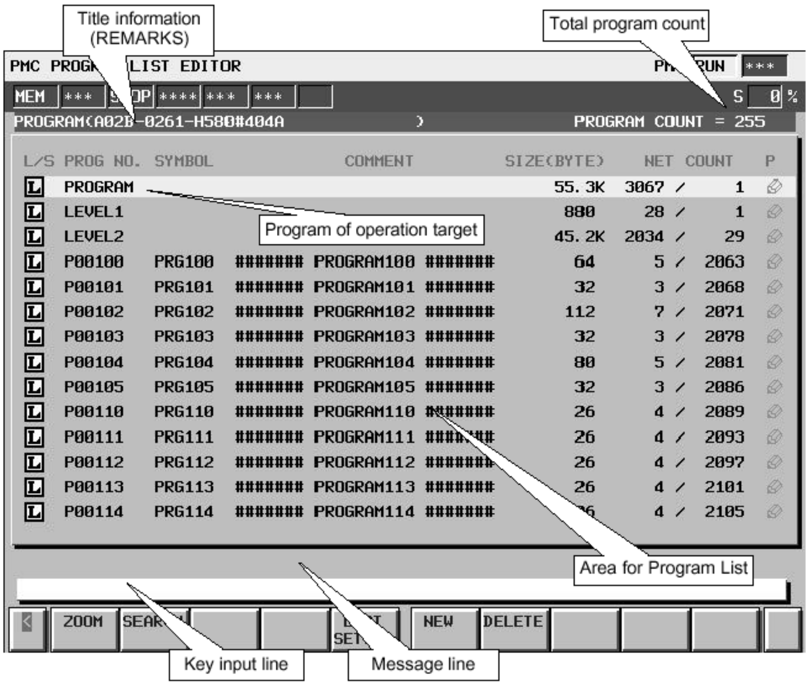
Program List Editor screen (Datail)

You can select Detail viewer format or Brief viewer format on Program List Editor screen.
The default viewer format is Detail viewer format.
See ”Setting screen for Program List Viewer screen” section for more detail.

6.5.1 Screen Structures

(1) Program List Editor screen(Detail)

Program List Editor screen (Detail) shows the detail information that are program size, program net count and so on for program.



(a) (a) Screen Structure

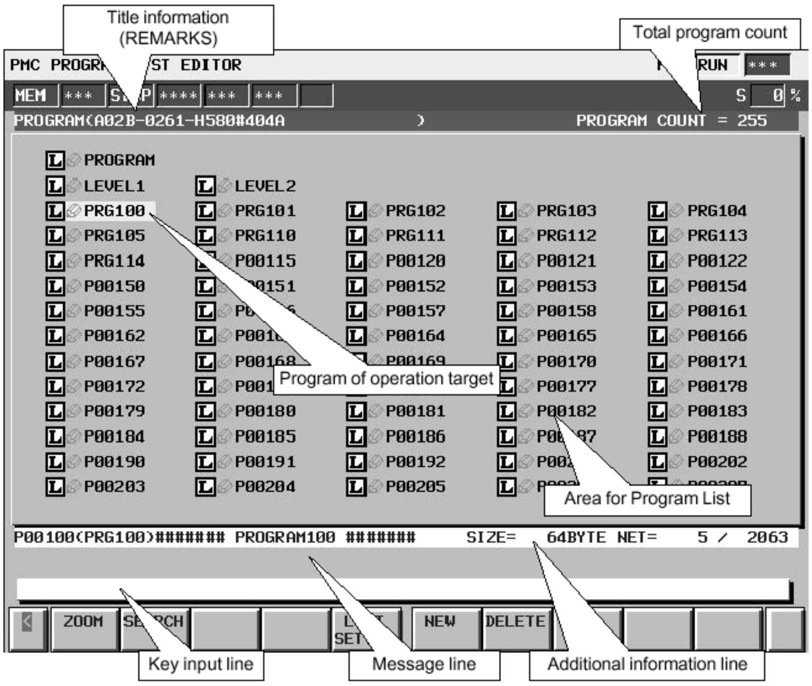
It is basically same with Program List Viewer screen (Detail).

(b) (b) Area of Program List

It is basically same with Program List Viewer screen (Detail).

(2) Program List Editor screen (Brief)

Program List Viewer screen (Brief) shows less information than Program List Viewer screen (Detail), to increase the number of items. The program type, the status of protection and the name or symbol are displayed for each program.
The detail information of program that is pointed by the cursor is displayed on additional information line.



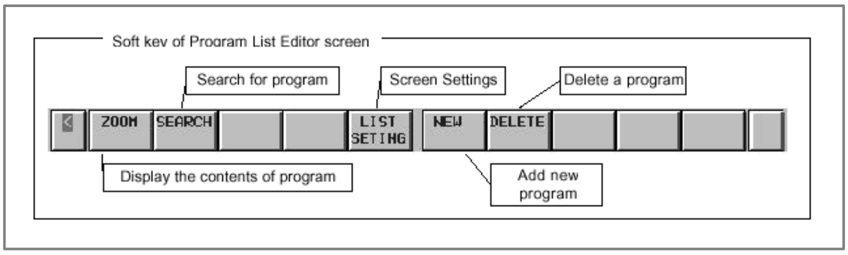
(a) Screen Structure

It is basically same with Program List Viewer screen (Brief).

(b) Area of Program List

It is basically same with Program List Viewer screen (Brief).

6.5.2 Operations



(a) Operation with Soft–keys

1. [ZOOM] Display the contents of program

   Goes to LADDER Diagram Editor screen. The operation is basically same with Program List Viewer screen. See ”Operations for Program List Viewer screen” section for the detail.

2. [SEARCH] Search for program

   The operation is basically same with Program List Viewer screen. See ”Operations for Program List Viewer screen” section for the detail.

3. [LIST SETING] Screen Settings

   The operation is basically same with Program List Viewer screen. See ”Operations for Program List Viewer screen” section for the detail.

4. [NEW] Create new program

   If you entered program name (See ”Note of searching operations for Program List Viewer screen” section for the detail) or symbol and press [NEW] soft key, the program will be checked its existence. If such program is not found, new program will be created. The created program is inserted automatically into the program list and the cursor points it.
   The following ladder nets are created automatically according to the type of created program by this operation.

LEVEL1	: Functional instruction END1
LEVEL2	: Functional instruction END2
LEVEL3	: Functional instruction END3
Subprogram	: Functional instruction SP, SPE

   If the status of protection of the program is enabled to edit, this operation is available.

5. [DELETE] Delete a program

   Deletes a program. If you entered no strings and press [DELETE] soft key, the program under the cursor is deleted. If you entered program name (See ”Note of searching operations for Program List Viewer screen” section for the detail) or symbol and press [DELETE] soft key, the program will be checked its existence, and will be deleted if such program is found.
   But, PROGRAM, LEVEL1 and LEVEL2 should always exist on program list.
   If you delete these programs, the contents of program are abandoned. But these programs do not disappear on program list.
   If the status of protection of the program is enabled to edit, this operation is available.

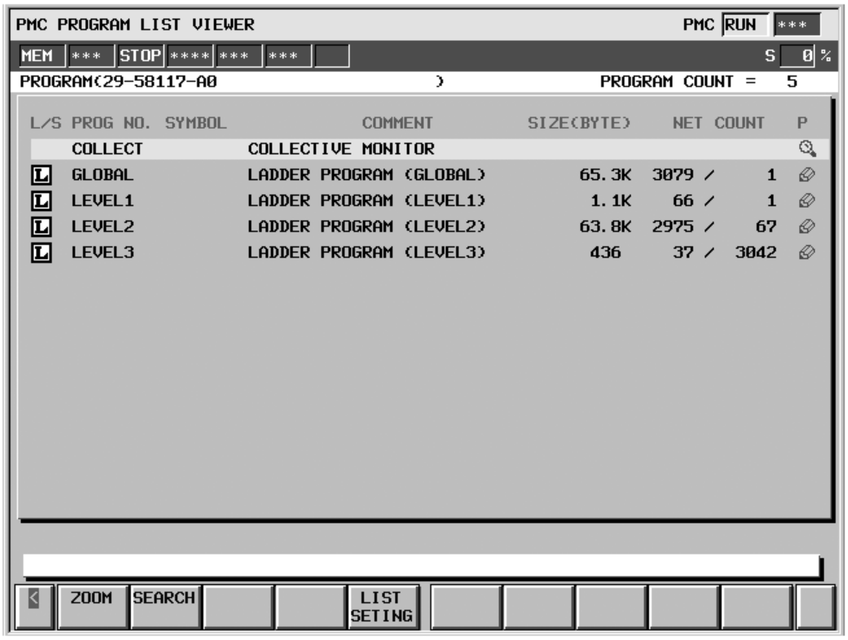
6.5.3 Setting Screen

Screen settings of Program List Editor screen are basically same with them of Program List Viewer screen. See ”Screen settings for Program List Viewer screen” section for the detail.

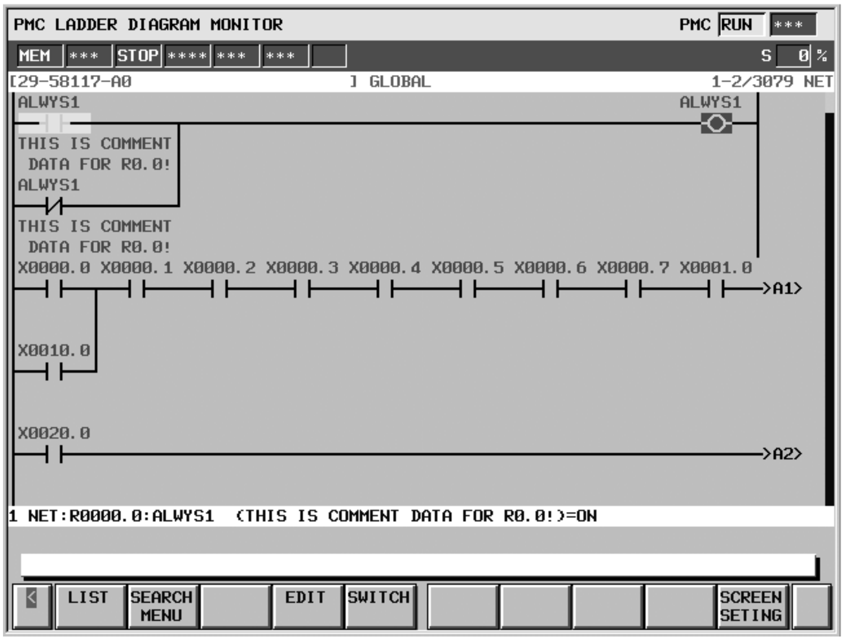
6.6 COLLECTIVE MONITOR FUNCTION

On Collective Monitor screen, only the necessary ladder nets can be displayed by specifying ladder nets.

Collective Monitor screen displayed by pressing soft key [ZOOM] after moving the cursor to ”COLLECT” program position on Program List Viewer screen, or Press soft key [SWITCH] on Ladder Diagram Monitor screen.

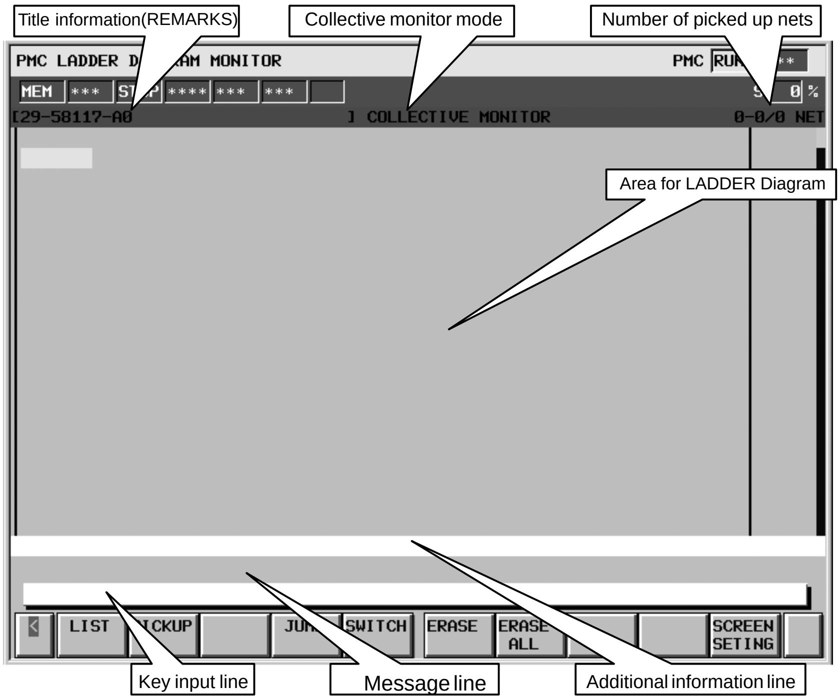


Program List Viewer screen



Ladder Diagram Monitor screen

6.6.1 Structures of Collective Monitor Screen



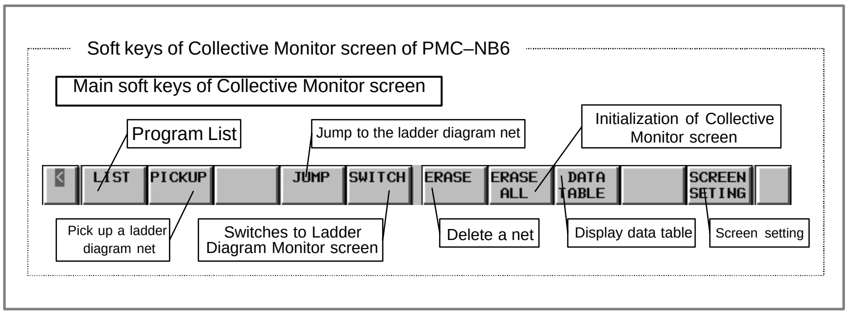
(a) Screen Structure

1. It is basically the same as LADDER Diagram Monitor screen.
2. ”COLLECTIVE MONITOR” displayed in the area for the current subprogram above the screen means that this screen is the Collective Monitor screen.

(b) LADDER Diagram

1. Any ladder diagram is not displayed first. Ladder nets selected by the operation of a coil search and picking up nets, is added.
2. 128 nets or less of ladder net can be displayed on Collective Monitor screen. When adding more new nets, most recently added 128 nets are displayed.

6.6.2 Operations of Collective Monitor Screen



(a) Operations using the soft keys

1. [LIST] Calling Program List Viewer screen.
   Calls Program List Viewer screen. On Program List Viewer screen, you can switch subprograms to be displayed on Ladder Diagram Viewer screen.
2. [PICKUP] Picking up ladder nets.
   Picks up ladder nets with coil which you want to monitor, on Collective Monitor screen. About how to pick up refer to ”7.6.6.3 How to pick up a ladder diagram net”.
3. [JUMP] Jump to a ladder net.
   On Ladder Diagram Monitor screen, search the ladder net at a cursor position on Collective Monitor screen and jumps to the ladder net. This soft key appears by setting ”SHOW CURSOR” to ”YES” on the setting screen.
4. [SWITCH] Switches to Ladder Diagram Monitor screen.
   Switches to Ladder Diagram Monitor screen.
5. [ERASE] Quits to dislpay a ladder diagram net on the screen. (1 net)
   Quits to dislpay a ladder diagram net (only 1 net) whch is picked up on Collective Monitor screen. This soft key appears by setting ”SHOW CURSOR” to ”YES” on the setting screen.
6. [ERASE ALL] Quits to dislpay a ladder diagram net on the screen. (all nets)
   Quits to dislpay ladder diagram nets (all net) whch is picked up on Collective Monitor screen.
7. [DATA TABLE] Go to Functional Instruction Data Table Viewer screen.
   Goes to Functional Instruction Data Table Viewer screen to examine contents of Data Table of functional instructions such as COD (SUB 7) and CODB (SUB 27), which have Data Table in themselves. This soft key appears only when the cursor is on a functional instruction that has Data Table.
8. [SCREEN SETING] Screen settings.
   Calling the setting screen for Collective Monitor screen. You can chenged each settings of a ladder diagram display. Return to Collective Monitor screen when press the return key [<].

(b) Other Operations

1. Cursor move keys, Page change keys
   Cursor move keys and Page change keys move cursor on screen. When cursor is placed on some relay or some address parameter of a functional instruction, the information about the address under cursor is displayed at ”Additional Information Line”.

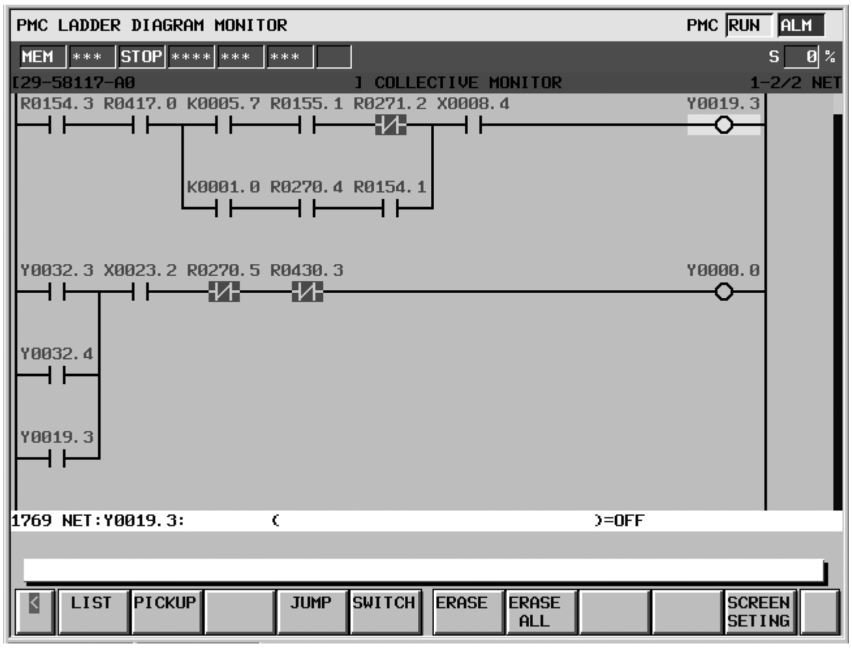
6.6.3 How to Pick Up a Ladder Diagram Net

The operation for picking up ladder nets which you want to monitor on Collective Monitor screen is as follows.

(1) Specification of ladder nets on Collective Monitor screen

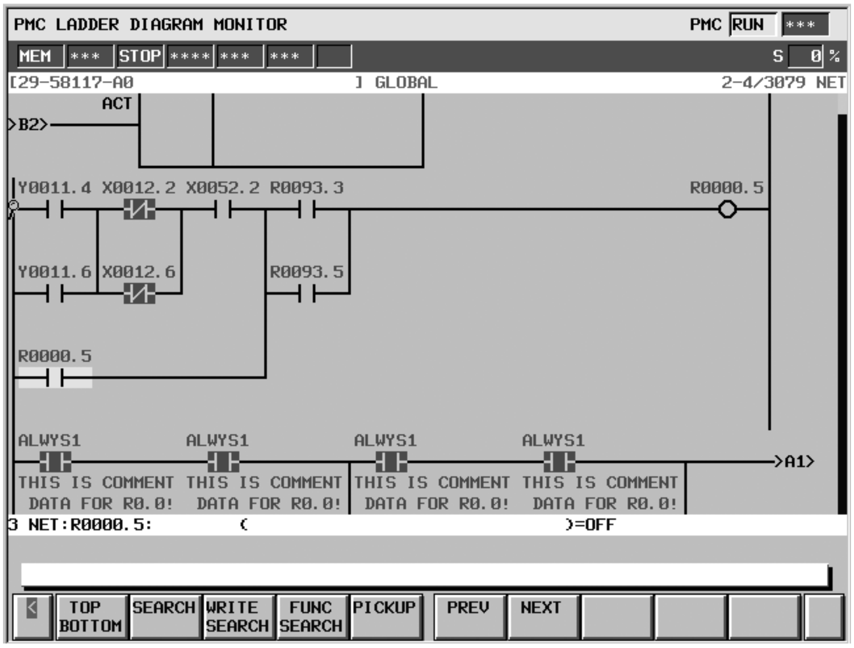
a) Specify the address by key input
   1. Enter the address which you want to monitor. (ex. Y0.0)
   2. Press [PICKUP] soft key.
   3. The net with the coil, which you specified by ”1”, is picked up to the top of screen.

b) Specification of address from the ladder net on the Collective Monitor screen
   1. Move the cursor to a relay on the ladder net which uses address that you want to monitor. (ex. Y19.3)
   2. Press [PICKUP] soft key.
   3. The net with the coil, which uses the address that you specified by ”1”, is picked up to the top of screen and the cursor moves to the specified coil position.

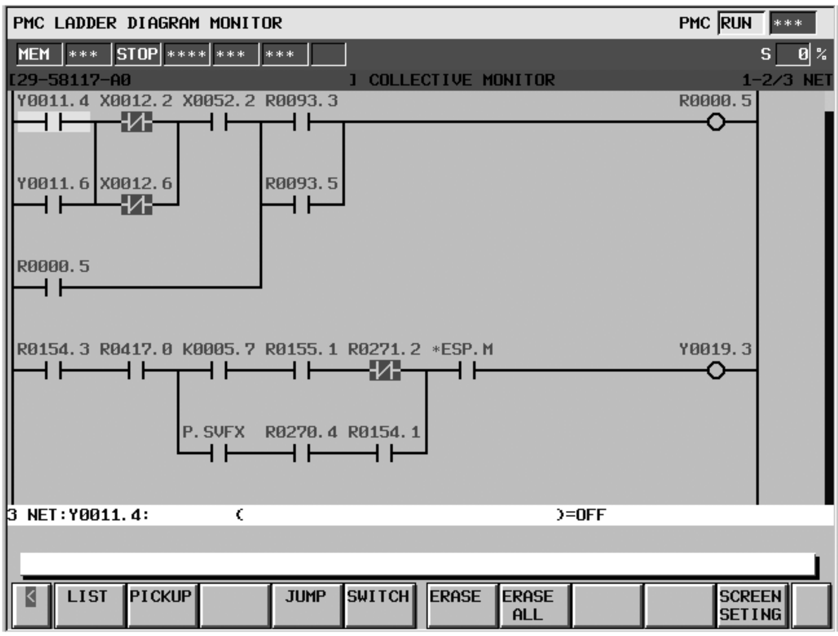


Collective Monitor screen (specification of ladder net on the screen)

1. Move the cursor to a relay on the ladder net which uses address that you want to monitor. (ex. R0.5)
2. Press [SEARCH MENU] soft key.
3. Switches Softleys display. And then, press [PICKUP] soft key.
4. The ” ” mark is displayed to the left of net. When you switshed to Collective Monitor screen, the net with the coil, which uses the address that you specified by ”1”, is picked up to the top of screen.



Ladder Diagram Monitor screen (picked up of ladder net on the screen)



Collective Monitor screen (specification of ladder net on the Ladder Diagram Monitor screen)

7 PASSWORD PROTECTION

The password function protects the LADDER Diagram Monitor screen and the LADDER Diagram Editor screen. If the password function is enabled, it is impossible to display or edit the ladder program. The protection was unlocked when entering the password, which is set on FAPT LADDER–II or LADDER Editing Package.

7.1 SPECIFICATIONS OF PASSWORD

(a) Password type

There are two password types.
One is the ”R–password” to protect against reading. Another is the ”RW–password” to protect against reading and writing.

(b) Available characters for password

The string that satisfies following condition can be set as the password.

- The string length is less than 8 characters.
- Only capital letters and numerals.

(c) Screens protected by password

The following screen is protected by the password.

- Ladder diagram monitor screen
- Ladder diagram editor screen

(d) Indicator of protection status

The status of protected program is displayed in the Program List Viewer screen and the Program List Editor screen. See ” program list screen (detail) for Program List Viewer screen” section for more detail.

(e) How to unlock password protection

Before a screen protected by the password is displayed, the password is inquired. The password is able to be unlocked when the strings of password are entered.

NOTE

1 Once the password has been unlocked, the system never inquires the password except turning the power on again or replacing the ladder program by I/O function.

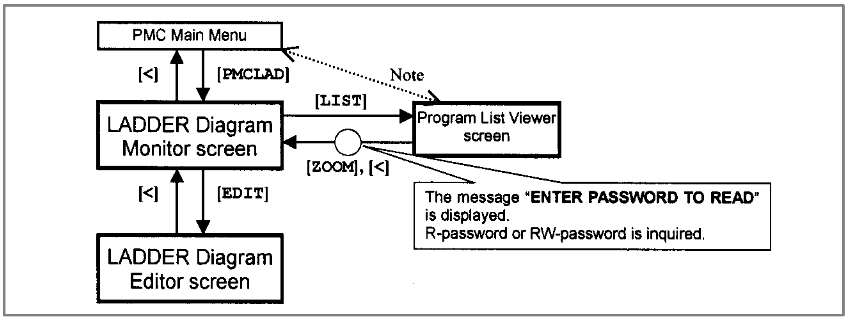
2 404A/04 can not display LADDER program that is protected by Password function. To view a protected LADDER program, you need FAPT LADDER–II, LADDER Editing Package, or upgrade version to 404A/05 or later. 404A/05 or later will ask for password when necessary.

3 When turning the power on pressing both ”X” key and ”O” key, the ladder program will be cleared regardless of password protection.

7.2 PASSWORD AND SWITCHING SCREEN

7.2.1 Using One of R–Password and RW–Password

The LADDER Diagram Monitor screen and the Ladder Diagram Editor screen are protected by the password.



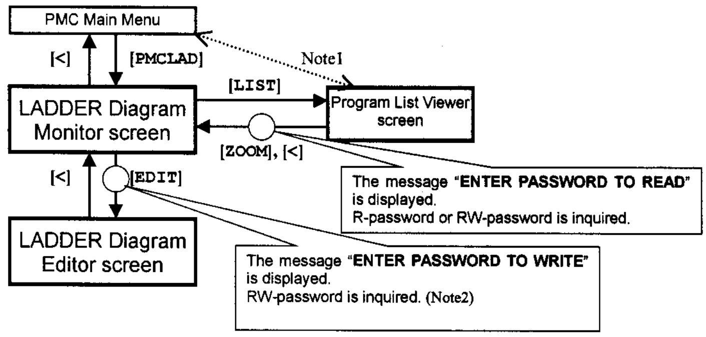
NOTE

When [PMCLAD] soft key is pressed at first time after CNC power cycle, Program List Viewer screen is displayed. Then use [ZOOM] soft key to select subprogram to reach the LADDER Diagram Monitor screen. The message ”ENTER PASSWORD TO READ” is displayed and the password is inquired. If you enter R–password or RW–password, the subprogram that you have selected appears on the screen. Once you choose subprogram, pressing [PMCLAD] soft key will directly reach the LADDER Diagram Monitor screen.

7.2.2 Using Both of R–Password and RW–Password

The LADDER Diagram Monitor screen and the LADDER Diagram Display screen are protected by the following password.

- The LADDER Diagram Monitor screen: R–password or RW–password
- The LADDER Diagram Display screen: RW–password



NOTE

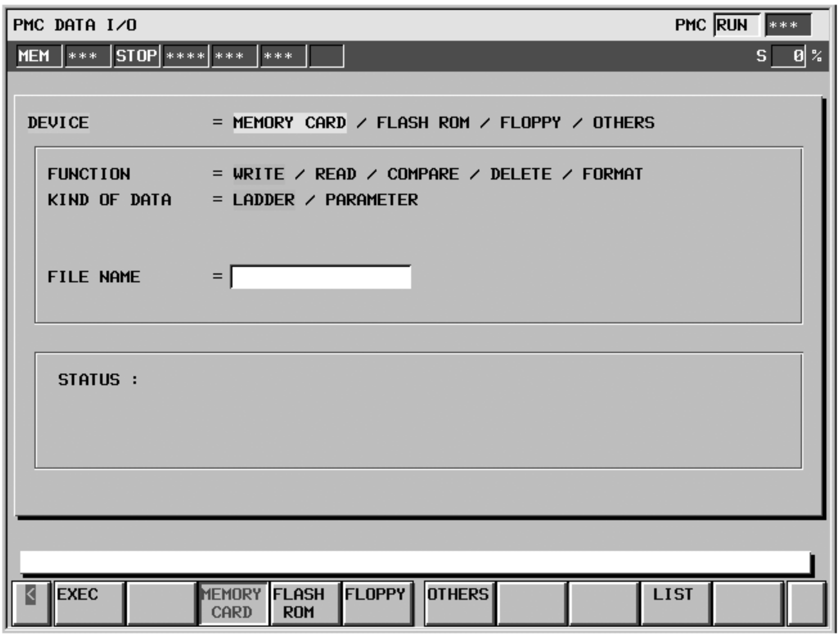
1 When [PMCLAD] soft key is pressed at first time after CNC power cycle, Program List Viewer screen is displayed. Then use [ZOOM] soft key to select subprogram to reach the LADDER Diagram Monitor screen. The message "ENTER PASSWORD TO READ" is displayed and the password is inquired. If you enter R–password or RW–password, the subprogram that you have selected appears on the screen. Once you choose subprogram, pressing [PMCLAD] soft key will directly reach the LADDER Diagram Monitor screen.

2 When [EDIT] soft key is pressed in the LADDER Diagram Monitor screen, the message "ENTER PASSWORD TO WRITE" is displayed and the password is inquired. If you enter RW–password, the LADDEER Diagram Editor screen is displayed. However, if you have already entered RW–password in program list viewer screen, the password is never inquired in the LADDER Diagram Monitor screen.

8 WRITING, READING, AND COMPARING SEQUENCE PROGRAMS AND PMC PARAMETERS

8.1 I/O SCREEN

When the [I/O] soft key on the PMC main menu is pressed, the following screen appears.



On this screen, sequence programs and PMC parameters can be written to the 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 ”DEVIDE” and either moving the option selection cursor to that type or selecting the soft key corresponding the type.

MEMORY CARD	Data can be output to and input from a memory card.
FLASH ROM	Data can be output to and input from frash ROM.
FLOPPY	Data can be output to and input from handy files or floppy cassettes.
OTHERS	Data can be output to and input from other input/output devices.

When you read a file from an I/O device, one of following messages appears and whether to operate the important thing is confirmed.

Sequence program:

THE FILE CONTAINS LADDER PROGRAM. PROCEED TO READ IT?

C language program:

THE FILE CONTAINS C LANGUAGE PROGRAM. PROCEED TO READ IT?

PMC parameters:

THE FILE CONTAINS PMC PARAMETER. PROCEED TO READ IT?

Other:

UNKNOWN FILE FORMAT

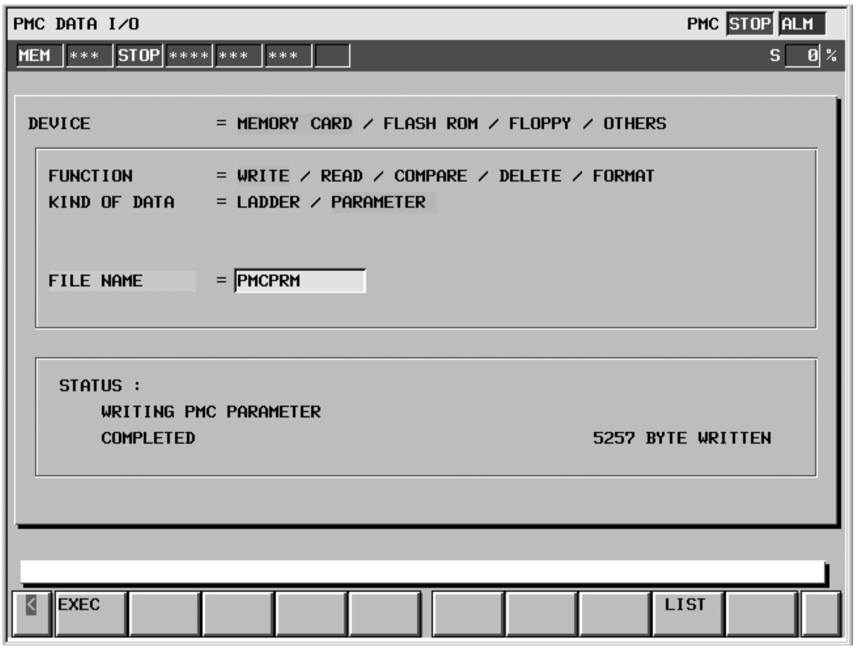
When you proceed to read a sequence program or a C language program, the ladder program stops automatically. When you proceed to read PMC parameters, new PMC parameters will be stored even if the ladder program is running.

WARNING

1 If a Ladder program or a C language program is input while a Ladder program is being executed, the execution of the Ladder program and the C language program stop automatically. You have to pay special attention to stop Ladder program. Stopping Ladder program in a wrong timing, or with machine in improper status, may cause unexpected reaction of machine. You have to make it sure that machine is in proper status, and nobody is near the machine when you stop Ladder program.
2 At stopping Ladder program, it may take rather long time to completely stop it in some cases according to the activity of Ladder program. If Ladder takes too long time to stop, or never stop, correct Ladder program, following instructions in section ”II. PMC OPERATION (CRT/MDI) 7.6.2.5 How to correct LADDER program that never stops”.
3 If the PMC parameters are input while a Ladder program is being executed, You have to special attention to input it. Because changed PMC parameters, may cause unexpected effect to Ladder. You have to make it sure that PMC parameters are not effect to Ladder when you input PMC parameters.

In STATUS in the lower part of the screen, a detailed explanation of execution and the execution status are displayed. During write, read, and comparison, the size of the data already transferred is indicated as the execution (intermediate) result.

The following gives a display example shown when PMC parameters are written to a memory card:

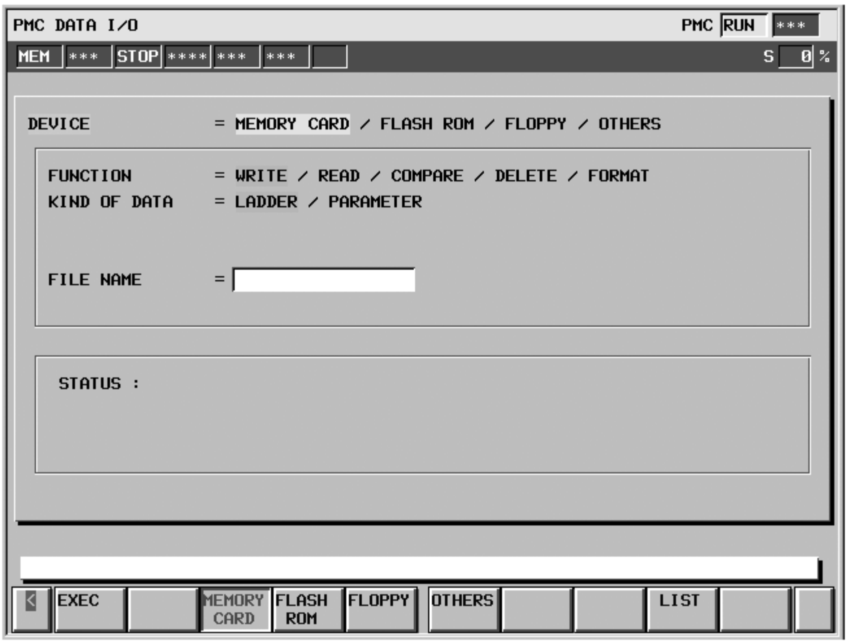


NOTE

1 The [I/O] soft key is displayed by setting bit 1 of keep relay K900 to 1.

2 For an explanation of error messages on I/O screen, see section 12.1.9, ”I/O screen Error Messages”.

8.2 OUTPUTTING TO AND INPUTTING FROM MEMORY CARDS



When “MEMORY CARD” is selected for DEVICE, output to and input from memory cards are enabled.

- FUNCTION
  A data I/O command appears. Select FUNCTION with the item select cursor, then select an item by moving the content select soft key horizontally or pressing an appropriate 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. (Programs in C cannot be output.)
READ:	Inputs data from a memory card to the PMC.
COMPARE:	Compares the sequence programs on the PMC with those on a memory card. (Programs in C cannot be compared.)
DELETE:	Deletes files from a memory card. (Files on a flash card cannot be deleted.)
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.

When you read a file from a memory card, one of following messages appears and whether to operate the important thing is confirmed.

Sequence program:

THE FILE CONTAINS LADDER PROGRAM. PROCEED TO READ IT?

C language program:

THE FILE CONTAINS C LANGUAGE PROGRAM. PROCEED TO READ IT?

PMC parameters:

THE FILE CONTAINS PMC PARAMETER. PROCEED TO READ IT?

Other:

UNKNOWN FILE FORMAT

When you proceed to read a sequence program or a C language program, the ladder program stops automatically. When you proceed to read PMC parameters, new PMC parameters will be stored even if the ladder program is running.

WARNING

1 If a Ladder program or a C language program is input while a Ladder program is being executed, the execution of the Ladder program and the C language program stop automatically. You have to pay special attention to stop Ladder program. Stopping Ladder program in a wrong timing, or with machine in improper status, may cause unexpected reaction of machine. You have to make it sure that machine is in proper status, and nobody is near the machine when you stop Ladder program.

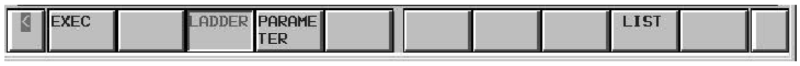
2 At stopping Ladder program, it may take rather long time to completely stop it in some cases according to the activity of Ladder program. If Ladder takes too long time to stop, or never stop, correct Ladder program, following instructions in section ”II. PMC OPERATION (CRT/MDI) 7.6.2.5 How to correct LADDER program that never stops”.

3 If the PMC parameters are input while a Ladder program is being executed, You have to special attention to input it. Because changed PMC parameters, may cause unexpected effect to Ladder. You have to make it sure that PMC parameters are not effect to Ladder when you input PMC parameters.

4 Set bit 1 of keep relay K900 to 0 when the machine tool is shipped.

- 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, “Memory Card List Screen” for details.

The PMC–NB6 supports the following memory cards:

○ : Supported
× : Not supported

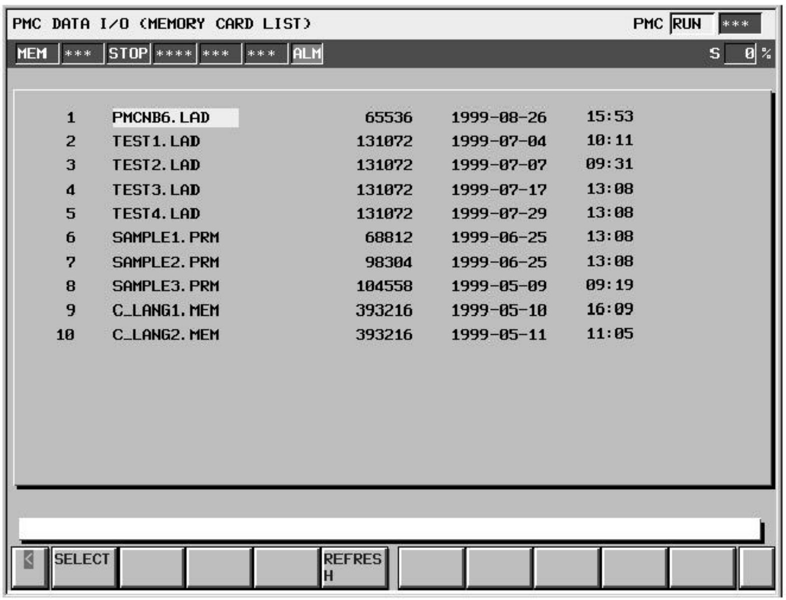
	SRAM Card	Flash Memory Card		ATA Card
		Supported Card	Unsupported Card	
Read of a file	○	○	○	○
Format of a card	○	○	×	○
Write of a file	○	○	×	○
Delete of a file	○	×	×	○
List of a file	○	○	○	○

NOTE

For an explanation of supported flash memory cards, see “(1) Writing to flash memory cards” in Section III.7.3.4.

8.3 MEMORY CARD LIST SCREEN

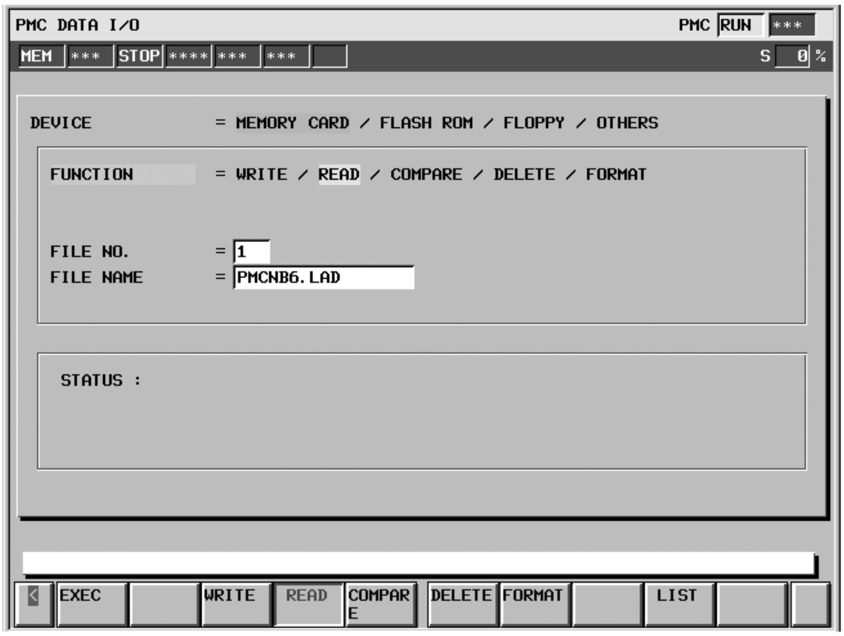
When MEMORY CARD is selected in DEVICE, pressing the [LIST] soft key displays the following screen:



If a memory card holding files is in the slot, the contents of the memory card are displayed as shown in the above.

NOTE
Up to 128 files can be displayed on this screen. When 129 or more files are saved in the memory card, the 129th and subsequent files are ignored.

When a file is selected on this screen, the screen display can be returned to the previous screen. To select a file, place the cursor at the name of the file, then press either the [SELECT] soft key or the INPUT key. After the key entry, the screen display switches to the previous screen automatically. In this case, the cursor is positioned at READ on the FUNCTION menu, and the number and name of the file selected on the list screen are indicated in the FILE NO. and FILE NAME fields, respectively. A display example is shown below.



To return the screen display to the previous screen without selecting a file, press the return key.

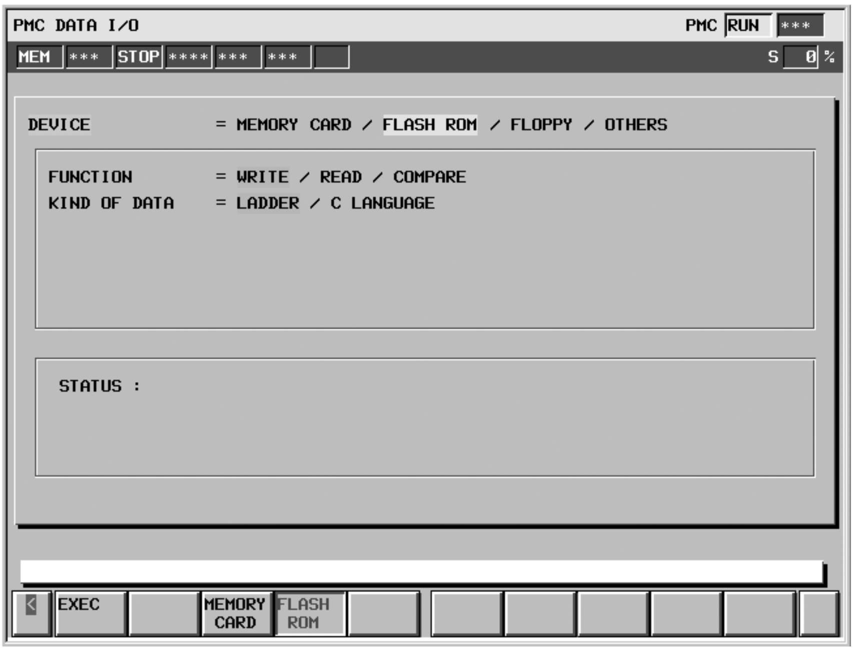
When the memory card is replaced with another card while the list screen is being displayed, the displayed information is not updated automatically. In this case, press the [REFRESH] soft key. The contents of the new memory card are then displayed.

Explanation of soft keys

[SELECT]: Selects a file, and returns the screen display to the previous screen.

[REFRESH]: Redisplays the contents of the memory card.

8.4 OUTPUTTING TO AND INPUTTING FROM FLASH ROM



When “FLASH ROM” is selected for DEVICE, output to and input from flash ROM are enabled.

> **NOTE**
> The above screen is displayed when the C language board is installed. When the C language board is not installed, only LADDER is indicated on the menu of the KIND OF DATA.

- 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.

When you read a file from a flash ROM, one of following messages appears and whether to operate the important thing is confirmed.

Sequence program:

THE FILE CONTAINS LADDER PROGRAM. PROCEED TO READ IT?

C language program:

THE FILE CONTAINS C LANGUAGE PROGRAM. PROCEED TO READ IT?

PMC parameters:

THE FILE CONTAINS PMC PARAMETER. PROCEED TO READ IT?

Other:

UNKNOWN FILE FORMAT

When you proceed to read a sequence program or a C language program, the ladder program stops automatically. When you proceed to read PMC parameters, new PMC parameters will be stored even if the ladder program is running.

WARNING

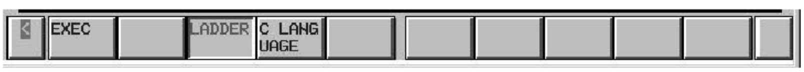
1 If a Ladder program or a C language program is input while a Ladder program is being executed, the execution of the Ladder program and the C language program stop automatically. You have to pay special attention to stop Ladder program. Stopping Ladder program in a wrong timing, or with machine in improper status, may cause unexpected reaction of machine. You have to make it sure that machine is in proper status, and nobody is near the machine when you stop Ladder program.

2 At stopping Ladder program, it may take rather long time to completely stop it in some cases according to the activity of Ladder program. If Ladder takes too long time to stop, or never stop, correct Ladder program, following instructions in section ”II. PMC OPERATION (CRT/MDI) 7.6.2.5 How to correct LADDER program that never stops”.

3 Set bit 1 of keep relay K900 to 0 when the machine tool is shipped.

- KIND OF DATA
  Displayed only when WRITE is selected in FUNCTION. Set the type of the data you want to output by moving the cursor horizontally or by using a soft key. C LANGUAGE appears only when the C language board is installed.

Soft keys displayed when the item select cursor is placed at KIND OF DATA



Explanation

LADDER: Outputs sequence programs only.

C LANGUAGE: Outputs programs in C.

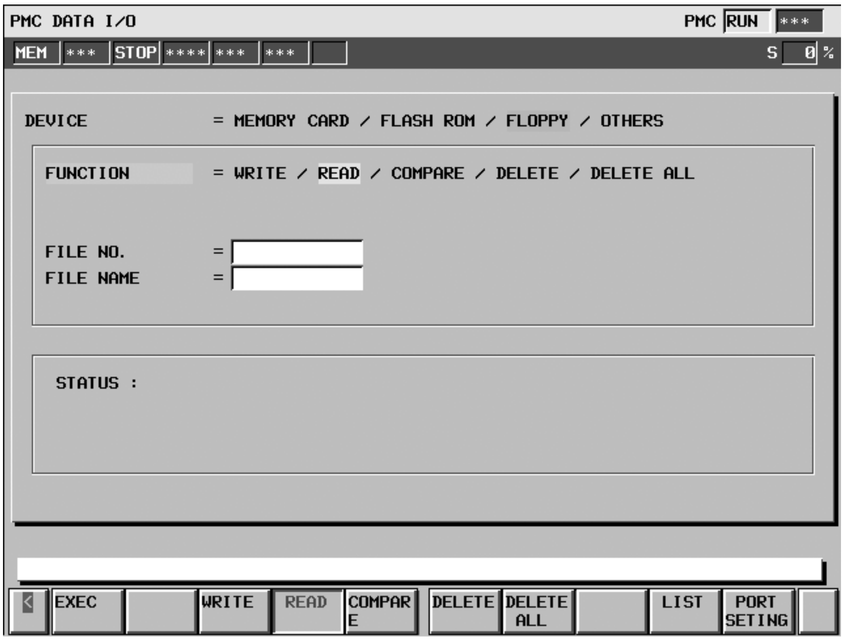
Explanation of soft keys

[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.

NOTE

When a program is written, it may take much time to initialize the flash ROM. During initialization, ”INITIALIZING FLASH ROM” appears in the STATUS field in the lower part of the screen.

8.5 OUTPUTTING TO AND INPUTTING FROM FLOPPY



When ”FLOPPY” is selected for ”DEVICE”, output to and input from floppy cassette or handy file that is connected via RS–232C 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 question selection cursor is positioned to ”FUNCTION”

Explanation of options

WRITE	Outputs data from the PMC to a floppy disk or handy file.
READ	Inputs data from a floppy disk or handy file to the PMC.
COMPARE	Compares the sequence program on the PMC with those on a floppy disk or handy file.
DELETE	Delete a file from a floppy disk or handy file.
DELETE ALL	Deletes all files from a floppy disk or handy file.

CAUTION
”DELETE ALL” command is not use for following floppy cassette.
FLOPPY CASSETTE ADAPTER A13B–0131–B001

When you read a file from a floppy cassete or a handy file, one of following messages appears and whether to operate the important thing is confirmed.

Sequence program:

THE FILE CONTAINS LADDER PROGRAM. PROCEED TO READ IT?

C language program:

THE FILE CONTAINS C LANGUAGE PROGRAM. PROCEED TO READ IT?

PMC parameters:

THE FILE CONTAINS PMC PARAMETER. PROCEED TO READ IT?

Other:

UNKNOWN FILE FORMAT

When you proceed to read a sequence program or a C language program, the ladder program stops automatically. When you proceed to read PMC parameters, new PMC parameters will be stored even if the ladder program is running.

WARNING

1 If a Ladder program or a C language program is input while a Ladder program is being executed, the execution of the Ladder program and the C language program stop automatically. You have to pay special attention to stop Ladder program. Stopping Ladder program in a wrong timing, or with machine in improper status, may cause unexpected reaction of machine. You have to make it sure that machine is in proper status, and nobody is near the machine when you stop Ladder program.

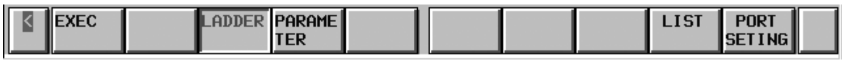
2 At stopping Ladder program, it may take rather long time to completely stop it in some cases according to the activity of Ladder program. If Ladder takes too long time to stop, or never stop, correct Ladder program, following instructions in section ”II. PMC OPERATION (CRT/MDI) 7.6.2.5 How to correct LADDER program that never stops”.

3 If the PMC parameters are input while a Ladder program is being executed, You have to special attention to input it. Because changed PMC parameters, may cause unexpected effect to Ladder. You have to make it sure that PMC parameters are not effect to Ladder when you input PMC parameters.

4 Set bit 1 of keep relay K900 to 0 when the machine tool is shipped.

- 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 command or select it with the corresponding soft key.

  Soft keys displayed when the question selection cursor is positioned to ”KIND OF DATA”



Explanation of options

LADDER Outputs sequence program.

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 NANE
  ”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.
  When you output to or input from floppy disk formatted in DOS format, 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 you output to or input from floppy disk formatted in FANUC format, a file name of up to 17 characters will be input.
  When ”WRITE” is selected for ”FUNCTION” and the file name is not entered, the following names are automatically assumed.

DATA KIND	File name
LADDER	PMCNB6.LAD
PARAM	PMCNB6.PRM

CAUTION

1 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.

2 Specifying the same name as that of an existing file results in an error.

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.

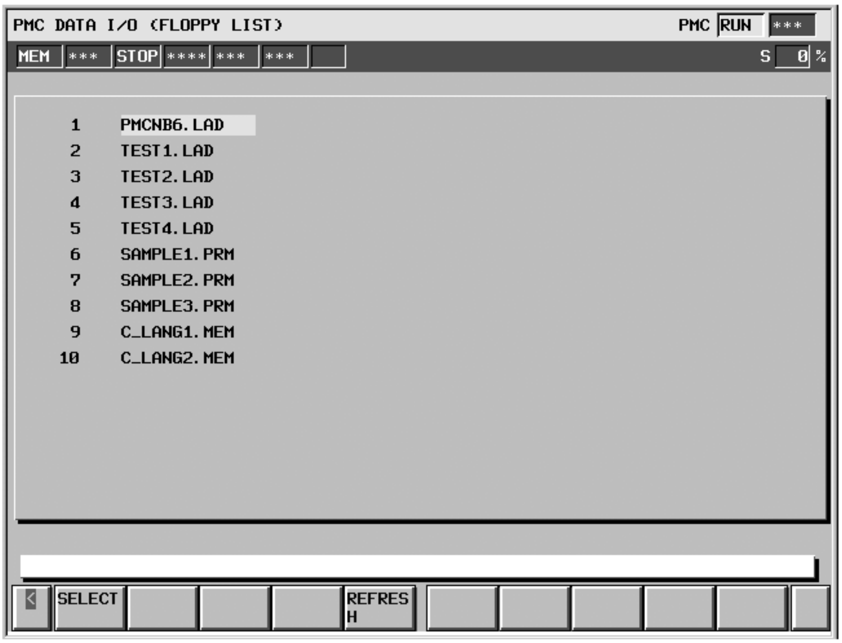
[CANCEL] Cancels the execution of the function. When the function terminates normally, the soft key disappears.

[LIST] Replaces the current display with the Floppy list screen. See Section 12.1.6, ”Floppy 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.

8.6 FLOPPY LIST SCREEN

When ”FLOPPY” is selected for ”DEVICE”, pressing the [LIST] soft key display the following screen.



The contents of the floppy cassettes or the handy files are displayed. When a file is selected on this screen, the screen display can be returned to the previous screen. To select the file, place the cursor at the name of the file, then press either the [SELECT] soft key or the INPUT key. After the key entry, the screen display switches to the previous screen automatically. In this case, the cursor is positioned at ”READ” on the ”FUNCTION” menu, and the number and name of the file selected on the list screen are indicated in the ”FILE NO.” and ”FILE NAME” fields, respectively.

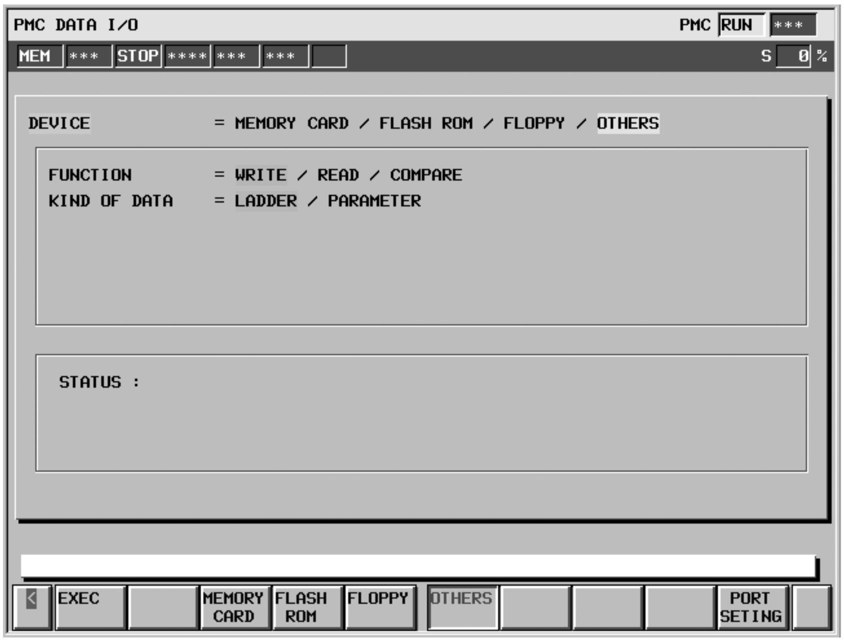
To return the screen display to the previous screen without selecting a file, press the return key.
When the floppy cassette or the handy file is replaced with another one while the list screen is being displayed, the displayed information is not updated automatically. In this case, press the [REFRESH] soft key. The contents are then displayed.

NOTE
Up to 128 files can be displayed on this screen. When 129 or more files are saved, the 129th and subsequent files are ignored.

Explanation of soft keys

[SELECT]	Selects a file, and returns the screen display to the previous screen.
[REFRESH]	Redisplays the contents of floppy cassette or handy file.

8.7 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.

  Soft keys displayed when the question selection cursor is positioned to ”FUNCTION”

Explanation of options

WRITE	Outputs data from the PMC to other input/output device.
READ	Inputs data from other input/output device to the PMC.
COMPARE	Compares the sequence program on the PMC with those on other input/output device.

When you read a file from a I/O device, one of following messages appears and whether to operate the important thing is confirmed.

Sequence program:

THE FILE CONTAINS LADDER PROGRAM. PROCEED TO READ IT?

C language program:

THE FILE CONTAINS C LANGUAGE PROGRAM. PROCEED TO READ IT?

PMC parameters:

THE FILE CONTAINS PMC PARAMETER. PROCEED TO READ IT?

Other:

UNKNOWN FILE FORMAT

When you proceed to read a sequence program or a C language program, the ladder program stops automatically. When you proceed to read PMC parameters, new PMC parameters will be stored even if the ladder program is running.

WARNING

1 If a Ladder program or a C language program is input while a Ladder program is being executed, the execution of the Ladder program and the C language program stop automatically. You have to pay special attention to stop Ladder program. Stopping Ladder program in a wrong timing, or with machine in improper status, may cause unexpected reaction of machine. You have to make it sure that machine is in proper status, and nobody is near the machine when you stop Ladder program.

2 At stopping Ladder program, it may take rather long time to completely stop it in some cases according to the activity of Ladder program. If Ladder takes too long time to stop, or never stop, correct Ladder program, following instructions in section ”II. PMC OPERATION (CRT/MDI) 7.6.2.5 How to correct LADDER program that never stops”.

3 If the PMC parameters are input while a Ladder program is being executed, You have to special attention to input it. Because changed PMC parameters, may cause unexpected effect to Ladder. You have to make it sure that PMC parameters are not effect to Ladder when you input PMC parameters.

4 Set bit 1 of keep relay K900 to 0 when the machine tool is shipped.

- 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 command or select it with the corresponding soft key.

  Soft keys displayed when the question selection cursor is positioned to ”KIND OF DATA”

Explanation of options

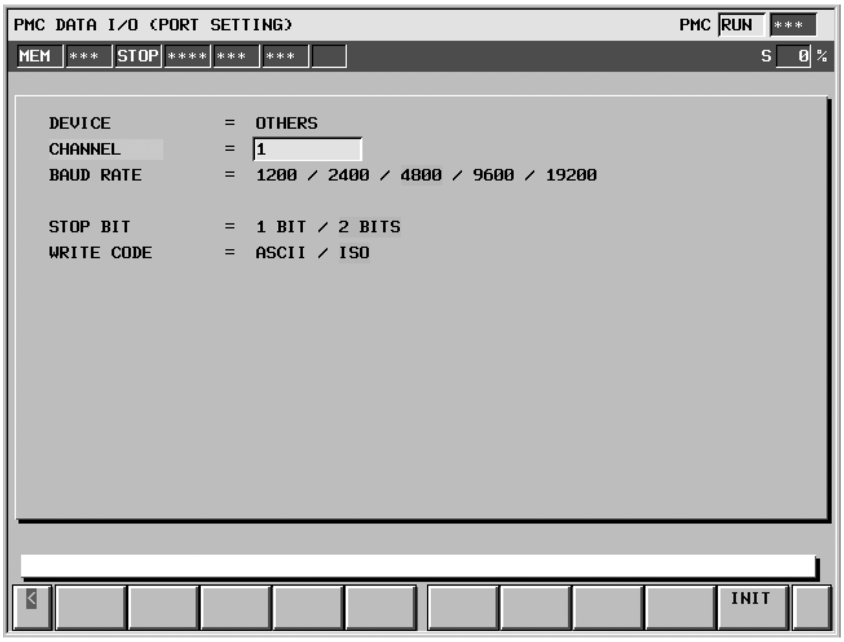
LADDER	Outputs sequence program.
PARAMETER	Outputs PMC parameters.

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.

8.8 PORT SETTING SCREEN

When ”FLOPPY” or ”OTHERS” is selected for ”DEVICE”, the [PORT SETING] soft key is displayed. When the key is pressed, the following screen appears. The following gives a display example shown when ”OTHERS” is selected for ”DEVICE”.



This screen allows the setting of the communication data required for communication using the RS–232C. Communication data can be set for each of the two types of devices independently of the other.
Selected device type is displayed to ”DEVICE” menu on screen.

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

- BAUDRATE
  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”.

- STOP BIT
  1 BIT: Sets the number of stop bits to ”1”.
  2 BIT: Sets the number of stop bits to ”2”.

- WRITE CODE
  ”WRITE CODE” is displayed when ”OTHERS” is selected for ”DEVICE”.
  ASCII: Sets the output code to ”ASCII”.
  ISO: Sets the output code to ”ISO”.

NOTE
Parity is always ”NONE”.

Explanation of soft keys

[INIT] Sets all the parameters to their initial values.

Initial values

	DEVICE = FLOPPY	DEVICE = OTHERS
CHANNEL	1	1
BAUD RATE	4800	4800
STOP BIT	2 BITS	2 BITS
WRITE CODE	(None)	ISO

8.9 I/O SCREEN ERROR MESSAGES (PMC–NB6)

The error messages that may appear on the I/O screen and their meanings and actions are listed below.

- Error messages displayed during memory card I/O operation

Displayed error message	Meaning and action
MEMORYCARD IS NOT READY	No memory card is installed. Action: Check whether a memory card is installed.
MEMORYCARD IS FULL	There is no available space in the memory card. Action: Delete files to create available space.
MEMORYCARD IS WRITE PROTECTED	The memory card is write–protected. Action: Release the write protection of the memory card.
MEMORYCARD IS NOT FORMATTED	The memory card cannot be recognized. Action: Format the memory card.
TOO MANY FILES IN MEMORYCARD	There are too many files. Action: Delete unnecessary files to reduce the number of files.
FILE NOT FOUND	The specified file cannot be found. Action: On the list screen, check the file name or file number.
FILE IS READ–ONLY	Write to the specified file is not permitted. Action: Check the attributes of the file.
FILE NAME IS INVALID	The file name is illegal. Action: Specify the file name in MS–DOS form.
CAN NOT FORMAT MEMORYCARD	The memory card cannot be formatted. Action: The NC cannot format this memory card. Use another unit such as a personal computer to format the memory card.
UNSUPPORTED MEMORYCARD	This memory card is not supported. Action: Replace the memory card with another one.
CAN NOT DELETE FILE	An error occurred when a file was deleted from the memory card. Action: Check the attributes of the file.
MEMORYCARD BATTERY ALARM	The battery of the memory card has become weak. Action: Replace the battery of the memory card.
THIS FILE NAME IS ALREADY USED	The file name is already used. Action: Change the file name to another one.
MEMORYCARD ACCESS ERROR	The memory card cannot be accessed. Action: Replace the memory card with another one.
DIFFERENCE FOUND	File comparison detected a mismatch.

Displayed error message	Meaning and action
MEMORYCARD IS LOCKED BY OTHER FUNCTION	Another PMC user is using the memory card. Action: Wait until the PMC user completes processing, then retry.
MEMORY CARD HEADER ROM DATA ID IS ILLEGAL	An attempt was made to read a file, but its ROM data ID was illegal. Action: This file cannot be read. Check the type of the file.
FLASH ROM HEADER ROM DATA ID IS ILLEGAL	An attempt was made to read a file, but its ROM data ID was illegal. Action: This file cannot be read. Check the type of the file.
COMPARE OF C LANGUAGE PROGRAM IS NOT SUPPORTED	The compare function for programs in C is not supported.
FILE NUMBER CAN NOT SELECTED	The file number cannot be selected. Action: If the file does not exist, the key entry is invalid. If this error occurs even when the cursor is placed at a file name, contact the FANUC service center.
THE FILE NUMBER DOES NOT EXIST	The entered file number is not present. The entered number exceeds the total number of files. Action: Check the total number of files on the list screen.
FILE NUMBER IS RESTRICTED TO ”128”	A value up to 128 can be entered as the file number. Action: Enter a numeric value not exceeding 128.
INTERNAL ERROR (xxxxxxxxxx)	An error due to an internal factor occurred. Details on the error are displayed in parentheses. Action: Contact the FANUC service center, and report the displayed message correctly.

- Error messages displayed during flash ROM I/O operation

Displayed error message	Meaning and action
NOT IN EMG STOP MODE	The system is not in the emergency stop state. Action: Place the system in the emergency stop state.
INVALID LADDER PROGRAM INVALID C LANGUAGE PROGRAM	The transfer program is illegal. Action: Check the program.
DIFFERENCE FOUND	A file comparison detected a mismatch.
FLASH ROM IS LOCKED BY OTHER FUNCTION.	Another PMC user is using the flash ROM. Action: Wait until the PMC user completes processing, then retry.

Displayed error message	Meaning and action
C LANGUAGE BOARD IS NOT IMPLEMENTED	The C language board cannot be recognized. Action: Check whether the C language board is installed.
INTERNAL ERROR (xxxxxxxxxx)	An error due to an internal factor occurred. Details on the error are displayed in parentheses. Action: Contact the FANUC service center, and report the displayed message correctly.

- Error messages displayed during floppy cassettes, handy files or other input/output devices I/O operation.

Displayed error message	Meaning and action
BAD PMC PARAMETER FORMAT	Specified file is not PMC parameter format. Action: Specify file of PMC parameter format, or check the contents of the file.
BAD HANDY FILE FORMAT	Specified file is not handy file format. Action: Specify file of handy file format, or check the contents of the file.
UNKNOWN FILE FORMAT	Can not recognize the format of specified file. Action: Specify file of recognizable format such as PMC parameter format, or check the contents of the file.
FILE NAME OR FILE NUMBER IS REQUIRED	Need file name or file number to identify file to read, compare, or delete. Action: Specify file name or file number for the operation.
COMMUNICATION TIMEOUT	Communication with the I/O device has been time-out. Action: Check the communication parameters such as baud rate, and retry to communicate.
I/O DEVICE IS NOT ATTACHED OR IN ERROR STATUS	Any I/O device is not connected, or some error has occurred in it. Action: Check the power of I/O device is ON. Check the I/O device is connected. Check the cable that connects I/O device with PMC is correct one. If some error has occurred in I/O device, solve it.
RECEIVED BAD DATA: CHECK THE COMMUNICATION PARAMETERS	Invalid data has been received. Action: Check the PMC's communication parameters such as baud rate match the ones of I/O device.
RECEIVED DATA HAS OVERRUN	Too many data have received at once. Action: Check the communication parameters about flow control.

Displayed error message	Meaning and action
OTHERS FUNCTION IS USING THIS CHANNEL	Others function is using this channel. Action: Use the other channel, or stop the function.
BAD COMMUNICATION PARAMETER	Setting parameters of communication are not correct. Action: Check the communication parameters such as baud rate.
OTHER FUNCTION IS USING I/O	Others function such us FAPT LADDER–II is using I/O function. Action: Wait until function that using I/O function do finish, or stop the function.
SEQUENCE PROGRAM IS IN USE BY ONLINE FUNCTION	Can not input/output of sequence program, because On–line function is using sequence program. Action: Wait until On–line function, do finish the using I/O function. In general, both of I/O function and On–line function should not be used at the same time.

9 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.

V. FS16*i*/18*i*/21*i*–B PMC–SA1/SB7 MANIPULATION

1 SCREEN OPERATION FOR PMC–SA1/SB7

1.1 SCREEN OPERATION FOR PMC–SA1

The following points are changed for improvement of operation and feature for PMC–SA1.

- Edit functions which is embedded in PMC control software
- Reinforcement of programmer protection function

1.1.1 Structure of Software Key in PMC

When pressing soft–key [PMC] in CNC system menu screen, PMC main menu appears. The following is the summary of PMC screen tree. The shaded part is the improved screen for PMC–SA1.

[PMC]	[PMCLAD]			LADDER DIAGRAM MONITOR
	[PMCDGN]	[TITLE]		TITLE DATA
		[STATUS]		SIGNAL STATUS
		[ALARM]		ALARM
		[TRACE]		SIGNAL TRACE
		[IOCHK]		I/O LINK CONNECTING CHECK
	[PMCPRM]	[TIMER]		TIMER
		[COUNTR]		COUNTER
		[KEEPRL]		KEEP RELAY
		[DATA]	[G.DATA]	DATA TABLE
		[SETING]		SETTING
	[RUN]/[STOP]			START AND STOP OF SEQUENCE PROGRAM
	[EDIT]	[TITLE]		TITLE DATA EDITOR
		[LADDER]		LADDER DIAGRAM EDITOR
		[SYMBOL]		SYMBOL DATA EDITOR
		[MESAGE]		MESSAGE DATA EDITOR
		[MODULE]		I/O UNIT ADDRESS SETTING
		[CROSS]		CROSS REFERENCE
		[CLEAR]		CLEARING SEQUENCE PROGRAM
	[I/O]			INPUT AND OUTPUT OF SEQUENCE PROGRAM AND PMC PARAMETER
	[SYSPRM]			SYSTEM PARAMETER
	[MONIT]	[ONLINE]		ONLINE SETTING

1.1.2 Addition of System Keep Relays

The following system keep relays are added.

Item	Meaning	PMC–SA1
EDIT ENABLE	Allows editing the ladder program.	K18.6
ALLOW PMC STOP	Allows start /stop of ladder program.	K19.2

Please refer to ”5. SETTING FUNCTION” for detail.

1.2 SCREEN OPERATION FOR PMC–SB7

The following points are changed for improvement of operation and feature for PMC–SB7.

- Edit functions which is embedded in PMC control software
- Display in Japanese
- Ladder monitor/editing screen
- Integration of Signal waveforms(ANALYS) and trace(TRACE)
- Extension of symbol data to 16 characters
- Reinforcement of programmer protection function

1.2.1 Structure of Software Key in PMC

When pressing soft–key [PMC] in CNC system menu screen, PMC main menu appears. The following is the summary of PMC screen tree. The shaded parts are the improved screens for PMC–SB7.

[PMC]	[PMCLAD]	[EDIT]		**LADDER DIAGRAM MONITOR**	**Refer to chapter 6**
				COLLECTIVE MONITOR FUNCTION	**Refer to chapter 7**
				LADDER DIAGRAM EDITOR	**Refer to chapter 8**
	[PMCDGN]	[TITLE]		TITLE DATA	
		[STATUS]		SIGNAL STATUS	
		[ALARM]		ALARM	
		[TRACE]		**SIGNAL TRACE**	**Refer to chapter 9**
		[IOCHK]		I/O LINK CONNECTING CHECK	
	[PMCPRM]	[TIMER]		TIMER	
		[COUNTR]		COUNTER	
		[KEEPRL]		KEEP RELAY	
		[DATA]	[G.DATA]	DATA TABLE	
		[SETING]		**SETTING SCREEN**	**Refer to chapter 5**
	[RUN]/[STOP]			START AND STOP OF SEQUENCE PROGRAM	
	[EDIT]	[TITLE]		TITLE DATA EDITOR	
		[LADDER]		LADDER DIAGRAM EDITOR	
		[SYMBOL]		SYMBOL DATA EDITOR	
		[MESAGE]		MESSAGE DATA EDITOR	
		[MODULE]		I/O UNIT ADDRESS SETTING	
		[CROSS]		CROSS REFERENCE	
		[CLEAR]		CLEARING SEQUENCE PROGRAM	
	[I/O]			INPUT AND OUTPUT OF SEQUENCE PROGRAM AND PMC PARAMETER	
	[SYSPRM]			SYSTEM PARAMETER	
	[MONIT]	[ONLINE]		ONLINE SETTING	

1.2.2 Expansion of Symbol Size

In symbol/comment display/editing screen, you can edit maximum 16 characters of symbols. Other screens which shows the symbols can display and handle the symbols up to following character length.

Screens	Max. display character length	Search length	Remark
Symbol/comment editing	16char.	16 char.	
Ladder monitor/editing	7char.	16 char.	16 characters display is available in additional information line
Program list	6 char.	16 char.	
Signal status	7 char.	16 char.	
Cross reference	7 char.	–	16 characters display is available in specifying address (type 1) screen
Trace	7 char.	16 char.	9 characters display is available in setting screen

The symbol characters exceeding the displaying field are omitted in each screen. Search function fully compares 16 characters in all screens which have search function.

1.2.3 Addition of System Keep Relays

The following system keep relays are added.

Item	Meaning	PMC–SB7
EDIT ENABLE	Allows editing the ladder program.	K901.6
ALLOW PMC STOP	Allows start /stop of ladder program.	K902.2
TRACE START	Starts the tracing automatically after power turns on	K906.5

Please refer to ”2. SETTING FUNCTION” for detail.

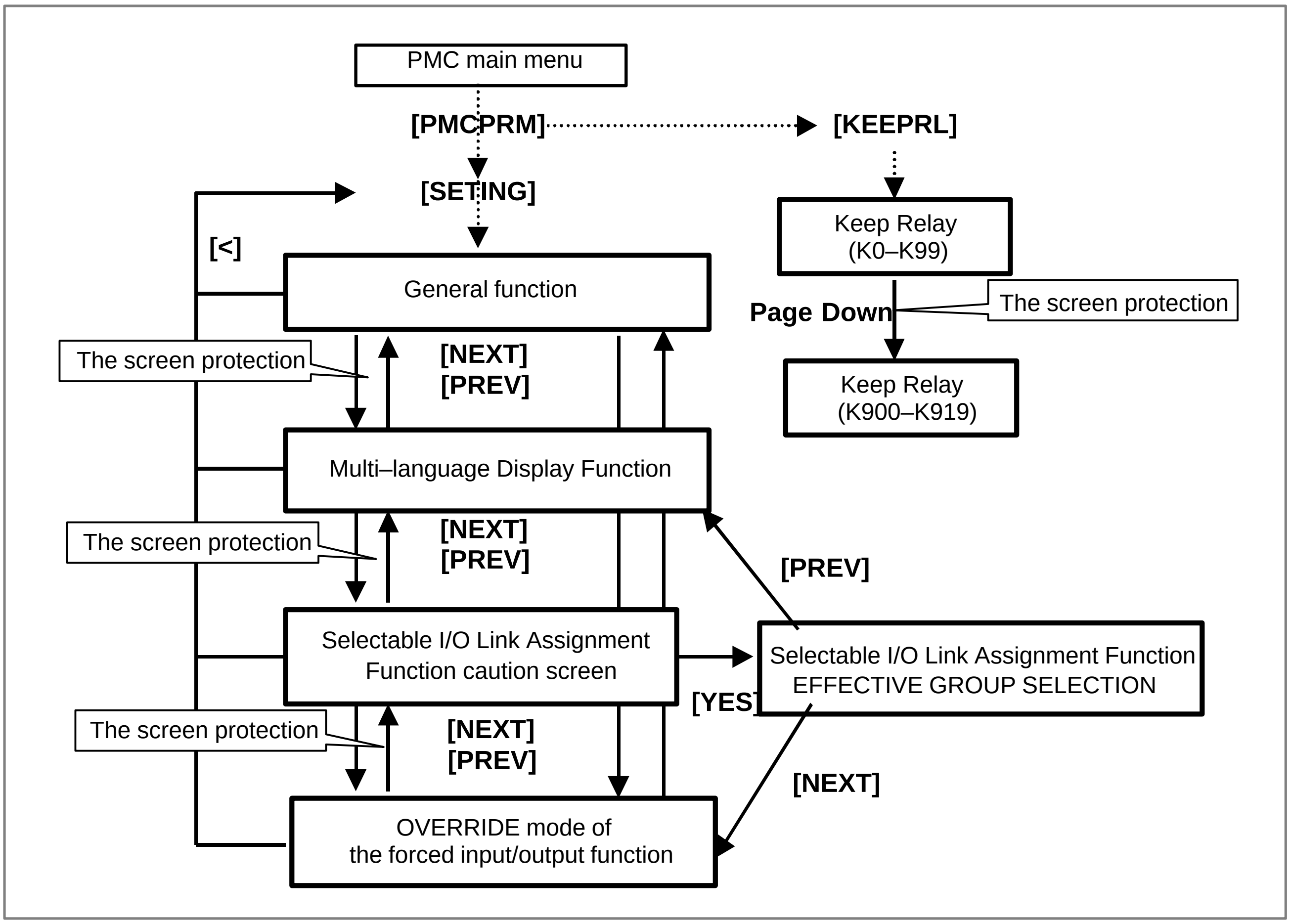
2 SETTING FUNCTION

2.1 SWITCHING SCREENS FOR SETTING PARAMETERS

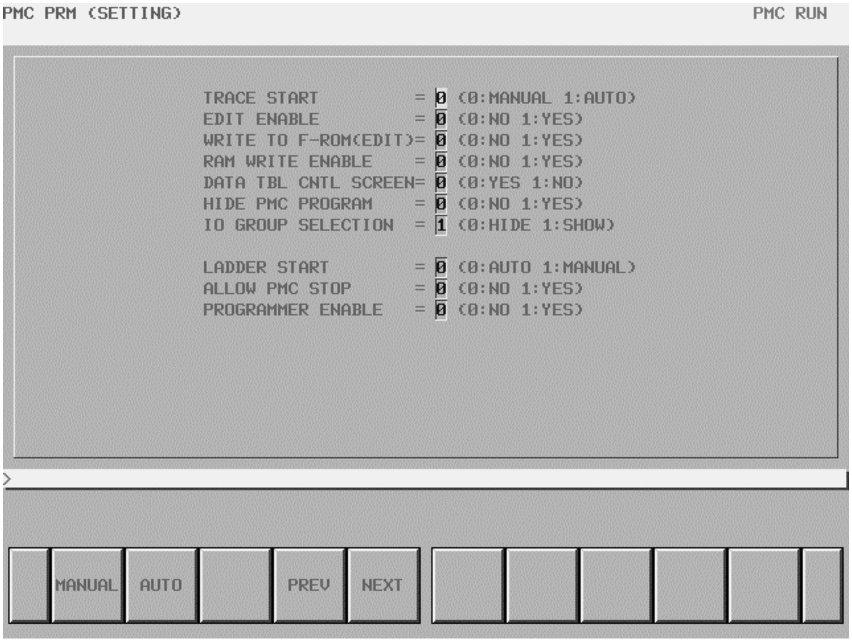
There are following setting parameter screens on PMC–SB7.

- Setting screen for general functions (See V–2.2 for details.)
- Setting screen for multi–language display function (See II–4.4.1 for details.)
- Setting screen for selectable I/O link assignment function (See V–2.4 for details.)
- Setting screen for OVERRIDE mode of the forced input/output function (See II–4.4.1 for details.)

The flow of switching above setting screens is as following chart. Each setting except multi–language display function can be also set on System Keep Relay screen. Setting screen of multi–language display function, selectable I/O link assignment function and OVERRIDE mode of the forced input/output function and System Keep Relay screen can be protected by programmer protection function.



2.2 SETTING SCREEN OF GENERAL FUNCTION



Setting screen for PMC–SB7

- TRACE START(PMC–SB7:K906.5)

  MANUAL (0): Executes the tracing with [EXEC] soft–key.

  AUTO (1): Starts the tracing automatically after the power turns on.

- EDIT ENABLE(PMC–SB7:K901.6, PMC–SA1:K18.6)

  NO (0): Prevents editing of the sequence program.

  YES (1): Allows editing of the sequence program.

> **NOTE**
> This setting effects some PMC functions.
> Please refer to ”2.5 Programmer protection function”

- WRITE TO F–ROM(PMC–SB7:K902.0, PMC–SA1:K19.0)

  NO (0): Does not write to flash ROM automatically after editing of sequence program.

  YES (1): Writes to flash ROM automatically after editing of sequence program.

- RAM WRITE ENABLE(PMC–SB7:K900.4, PMC–SA1:K17.4)

  NO (0): Prevents forcing function.

  YES (1): Allows forcing function.

> **NOTE**
> This setting effects some PMC functions.
> Please refer to ”2.5 Programmer protection function”

- DATA TBL CNTL SCREEN(PMC–SB7:K900.7, PMC–SA1:K17.7)
  YES (0): Displays PMC parameter data table control screen.
  NO (1): Does not displays PMC parameter data table control screen.
- HIDE PMC PROGRAM(PMC–SB7:K900.0, PMC–SA1:K17.0)
  NO (0): Allows sequence program display.
  YES (1): Prevents sequence program display.

NOTE
This setting effects some PMC functions.
Please refer to ”2.5 Programmer protection function”

- IO GROUP SELECTION (PMC–SB7:K906.1)
  HIDE (0) : The selectable I/O link assignment function screen is not displayed.
  SHOW (1) : The selectable I/O link assignment function screen is displayed.
- LADDER START(PMC–SB7:K900.2, PMC–SA1:K17.2)
  AUTO (0): Executes the sequence program automatically after the power turns on.
  MANUAL(1): Executes the sequence program by [RUN] soft–key.
- ALLOW PMC STOP(PMC–SB7:K902.2, PMC–SA1:K19.2)
  NO (0): Prevents run/stop operation of the sequence program.
  YES (1): Allows run/stop operation of the sequence program.

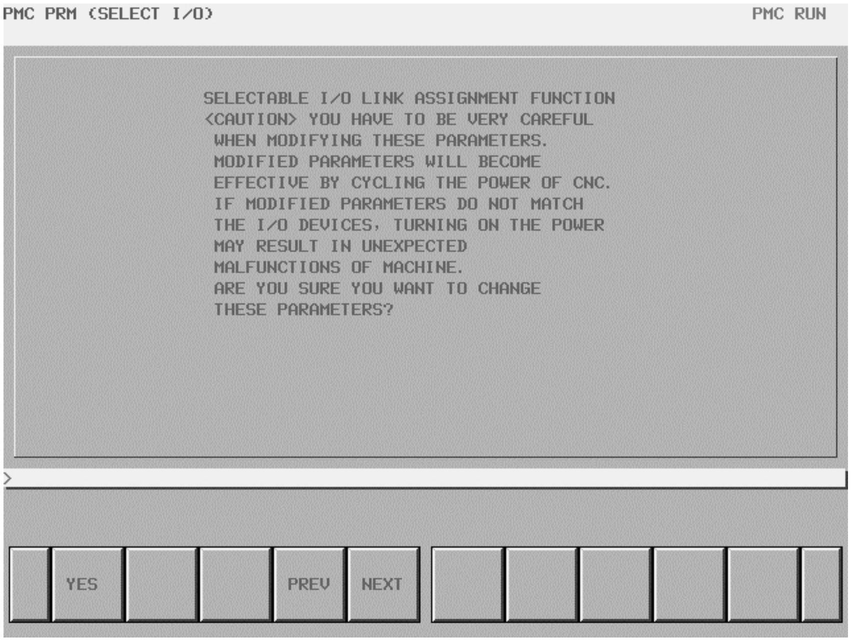
NOTE
This setting effects some PMC functions.
Please refer to ”2.5 Programmer protection function”

- PROGRAMMER ENABLE(PMC–SB7:K900.1, PMC–SA1:K17.1)
  NO (0): Disables embedded programmer function.
  YES (1): Enables embedded programmer function.

NOTE
This setting effects some PMC functions.
Please refer to ”2.5 Programmer protection function”

2.3 CAUTION SCREEN OF THE SETTING PARAMETER FOR THE SELECTABLE I/O LINK ASSIGNMENT FUNCTION

This caution screen is displayed for the operator’s attention when the operator is going to switch to the setting parameter screen for the selectable I/O link assignment function.



The caution screen of the setting parameter for the selectable I/O link assignment function

[YES]: This softkey switches to the setting parameter screen for the selectable I/O link assignment function.

[PREV]: This softkey switches to the previous setting parameter screen.

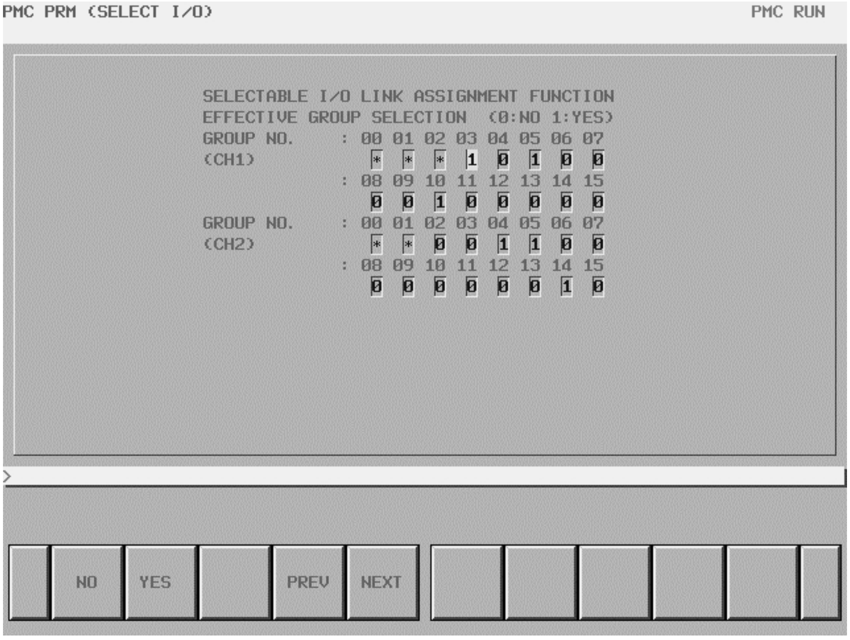
[NEXT]: This softkey switches to the next setting parameter screen.

WARNING

If you modify this setting parameter without care, the I/O assignment data may not match I/O devices and turning on the power may result in unexpected malfunctions of machine. So, it is required that the operator of this function should be an expert who fully understands the sequence program and the operation of PMC. It is also strongly recommended to the developer of machine that this setting screen should be protected from careless use by ordinary operators after the machine is shipped into the field.

2.4 SETTING PARAMETER SCREEN FOR THE SELECTABLE I/O LINK ASSIGNMENT FUNCTION

You can set the group of optional I/O device that is connected with each machines.



The setting parameter screen for the selectable I/O link assignment function

EFFECTIVE GROUP SELECTION (channel 1: K910–K911, channel2: K912–K913)

You can select effective I/O group in I/O link assignment data.

1 : I/O group is effective.

0 : I/O group is no effective.

The ”*” mark means that the group is set as the basic part by the parameter ”BASIC GROUP COUNT” on the system parameter screen. The value can not be set into this parts.

2.5 PROGRAMMER PROTECTION FUNCTION

CAUTION

This section contains important information for developers of application system controlled by PMC. Improperly implemented application system may increase possibility of defects in its safety. Careful examinations and considerations on using and implementing with the functions explained especially in this section are strongly required.

PMC system provides various embedded programmer functions such as edit, diagnosis and debugging which help the programming and debugging of sequence program. To use these functions which may even disable safety mechanism realized by sequence program, it is required that the operator of these functions should be an expert who fully understands the sequence program and the operation of PMC. It is also strongly recommended to the developer of machine that these functions should be protected from careless use by ordinary operators after the machine is shipped into the field. Furthermore, if these functions partly need to be used in the field for any purpose such as the maintenance or adjustment, the developer of the machine should implement any means to enable these functions after forcing the machine in safe mode or should let the operator know and strictly follow proper procedure to ensure the safety.

The setting parameters described in this section are provided for the developer of machine to be able to properly program the sequence or control the parameters for necessary conditions on which the operator is allowed to use PMC programmer functions safely by eliminating careless operation which may cause ”stopping the ladder”, ”changing sequence program” or ”changing important setting data”.

These parameters can be set on the setting screen or in some system keep relays (:PMC–SB7: K900 to 919,PMC–SA1: K17 to 19).

2.5.1 PROGRAMMER ENABLE (PMC–SB7: K900.1, PMC–SA1: K17.1)

If you set ”PROGRAMMER ENABLE ” to ”YES”, it enables the following functions as a supervisor mode.

– Ladder editing screen
– Title data editing screen
– Symbol/comment data editing screen
– Message data editing screen
– I/O unit address setting screen
– Cross reference screen
– Clear of sequence program
– Clear of PMC parameter
– Start/stop of ladder
– Forcing function
– Override function*1
– Setting of multi–language message display function
– I/O screen
– System parameter screen
– Online setting screen
– Setting screen for the I/O link assignment data selection function
– Setting screen for keep relay K900 or after

NOTE
The override function also requires the setting of ”OVERRIDE ENABLE” in the setting parameters.

CAUTION
Set this setting to ”NO”(0) before shipment of the machine. If this setting is left as ”YES”(1), the operator may stop execution of the ladder program by mistake. If you want to protect this setting, please make a sequence that always writes 0 in this bit by your ladder. Or please control the machine to force to translate into safety state by sequence program using the way described in ”3.3” when the ladder stops.

2.5.2 HIDE PMC PROGRAM (PMC–SB7: K900.0, PMC–SA1: K17.0)

If you set ”HIDE PMC PROGRAM” to ”YES”, it disables the following functions which have the sequence program display.

– Ladder monitor screen
– Ladder editing screen
– Title data editing screen
– Symbol/comment data editing screen
– Message data editing screen
– I/O unit address setting screen
– Cross reference screen
– Clear of sequence program
– Clear of PMC parameter
– System parameter screen

> **NOTE**
> Even if this parameter is set to ”YES”, these functions do not be hidden except for Ladder monitor/editing screen if ”PROGRAMMER ENABLE” is set to ”YES”.

2.5.3 EDIT ENABLE (PMC–SB7: K901.6, PMC–SA1: K18.6)

If you set ”EDIT ENABLE” to ”YES”, it enables the following functions which can edit the program.

– Ladder editing screen*1
– Title data editing screen*1
– Symbol/comment data editing screen*1*2
– Message data editing screen*1*2
– I/O unit address setting screen*1*2
– Cross reference screen*1
– Clear of sequence program*1*2
– Clear of PMC parameter*1*2
– Setting of multi–language message display function
– System parameter screen*1
– Setting screen for keep relay K900 or after

> **NOTE**
> 1 Even if this parameter is set to ”YES”, these functions which have program display are invalid if ”HIDE PMC PROGRAM” is set to ”YES”.
> 2 These screens with stop of ladder program require below setting ”ALLOW PMC STOP”.

> **CAUTION**
> Set this setting to ”NO”(0) before shipment of the machine if you want to prohibit operator form editing the program. If you want to protect this setting, please make a sequence that always writes 0 in this bit by your ladder.

2.5.4 ALLOW PMC STOP (PMC–SB7: K902.2, PMC–SA1: K19.2)

If you set ”ALLOW PMC STOP” to ”YES”, it enables the following functions which require stop/start of ladder program. *1

– Symbol/comment data editing screen*2
– Message data editing screen*2
– I/O unit address setting screen*2
– Clear of sequence program*2
– Clear of PMC parameter*2
– Start/stop of ladder
– System parameter screen*2

NOTE

1 Even if this parameter is set to ”YES”, these functions which have program display are invalid if ”HIDE PMC PROGRAM” is set to ”YES”.
2 These editing screens require above setting ”EDIT ENABLE”.

CAUTION

Set this setting to ”NO”(0) before shipment of the machine. If this setting is left as ”YES”(1), the operator may stop execution of the ladder program by mistake. If you want to protect this setting, please make a sequence that always writes 0 in this bit by your ladder. Or please control the machine to force to translate into safety state by sequence program using the way described in ”3.3” when the ladder stops.

2.5.5 RAM WRITE ENABLE (PMC–SB7: K900.4, PMC–SA1: K17.4)

If you set ”RAM WRITE ENABLE” to ”YES”, it enables both the forcing function and the override function.

NOTE

The override function also requires the setting of ”OVERRIDE ENABLE” in the setting parameters.

2.5.6 DATA TBL CNTL SCREEN (PMC–SB7: K900.7, PMC–SA1: K17.7)

If you set ”DATA TBL CNTL SCREEN” to ”NO”, the data table control screen does not be displayed.

2.5.7 IO GROUP SELECTION (K906.1)

If you set ”IO GROUP SELECTION ” to ”SHOW”, the setting screen for the selectable I/O link assignment function is enabled.

2.6 EXAMPLE FOR SETTING PARAMETERS

i) If you want to prohibit completely operator from accessing the sequence program;

- PROGRAMMER ENABLE
  (PMC–SB7: K900.1, PMC–SA1: K17.1) ”NO”
- HIDE PMC PROGRAM
  (PMC–SB7: K900.0, PMC–SA1: K17.0) ”YES”
- EDIT ENABLE
  (PMC–SB7: K901.6, PMC–SA1: K18.6) ”NO”
- ALLOW PMC STOP
  (PMC–SB7: K902.2, PMC–SA1: K19.2) ”NO”

ii) If you want to allow operator only monitoring the sequence program;

- PROGRAMMER ENABLE
  (PMC–SB7: K900.1, PMC–SA1: K17.1) ”NO”
- HIDE PMC PROGRAM
  (PMC–SB7: K900.0, PMC–SA1: K17.0) ”NO”
- EDIT ENABLE
  (PMC–SB7: K901.6, PMC–SA1: K18.6) ”NO”
- ALLOW PMC STOP
  (PMC–SB7:K 902.2, PMC–SA1: K19.2) ”NO”

NOTE
Please use the password function of sequence program for particular operator. Please refer to FAPT LADDER–III operator’s manual B–66234EN ”5.4 PROTECING LADDER PROGRAMS BY PASSWORD”

iii) If you want to allow operator monitoring and editing the sequence program;

- PROGRAMMER ENABLE
  (PMC–SB7: K900.1, PMC–SA1: K17.1) ”NO”
- HIDE PMC PROGRAM
  (PMC–SB7: K900.0, PMC–SA1: K17.0) ”NO”
- EDIT ENABLE
  (PMC–SB7: K901.6, PMC–SA1: K18.6) ”YES”
- ALLOW PMC STOP
  (PMC–SB7: K902.2, PMC–SA1: K19.2) ”NO”

NOTE
Please use the password function of sequence program for particular operator. Please refer to FAPT LADDER–III operator’s manual B–66234EN ”5.4 PROTECING LADDER PROGRAMS BY PASSWORD”.

iv) If you want to allow operator monitoring and editing the sequence program which requires stop of ladder;

- PROGRAMMER ENABLE
  (PMC–SB7: K900.1, PMC–SA1: K17.1) ”NO”

- HIDE PMC PROGRAM
  (PMC–SB7: K900.0, PMC–SA1: K17.0) ”NO”
- EDIT ENABLE
  (PMC–SB7: K901.6, PMC–SA1: K18.6) ”YES”
- ALLOW PMC STOP
  (PMC–SB7: K902.2, PMC–SA1: K19.2) ”YES”

NOTE
Please use the password function of sequence program for particular operator. Please refer to FAPT LADDER–III operator’s manual B–66234EN ”5.4 PROTECING LADDER PROGRAMS BY PASSWORD”.

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 work–piece or machine. Incorrect operation of the machine presents an extreme risk of death or serious injury to the user. Damage the tool, work–piece, and/or the machine is also likely.

v) The case that operator who familiar with the machine and the ladder sequence operate all the PMC programmer functions;

- PROGRAMMER ENABLE
  (PMC–SB7: K900.1, PMC–SA1: K17.1) ”YES”
- HIDE PMC PROGRAM
  (PMC–SB7: K900.0, PMC–SA1: K17.0) ”NO”

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 work–piece or machine. Incorrect operation of the machine presents an extreme risk of death or serious injury to the user. Damage the tool, work–piece, and/or the machine is also likely.

3 LADDER DIAGRAM MONITOR (PMC–SB7)

Pressing soft key [PMCLAD] displays the sequence program dynamically. On the screen, you can monitor the LADDER Program works.
The LADDER Diagram Editor screen allows you changing the LADDER Diagram, adding relays and functional instructions, and changing the action of LADDER Diagram.

LADDER Diagram Monitor/Editor function consists of following screens.

a) LADDER Diagram Monitor screen
Displays LADDER Diagram and the current status of relays and others.

b) Collective Monitor screen
Displays only selected ladder diagram and the current status of relays and others.

c) LADDER Diagram Editor screen
Edits LADDER Diagram by the net.

d) Net Editor screen
Edits the structure of a net in LADDER Diagram

e) PMC Functional Instruction Data Table Viewer screen
Refers the contents of a data table of functional instruction.

f) PMC Functional Instruction Data Table Editor screen
Edits the contents of a data table of functional instruction.

g) Program List Viewer screen
Chooses the subprogram to be displayed at LADDER Diagram Monitor screen.

h) Program List Editor screen
Edits LADDER Diagram by the subprogram. Also chooses the subprogram to be displayed at LADDER Diagram Editor screen.

NOTE
You can hide [PMCLAD] soft key at PMC Main Menu by setting the keep relay of K900.0 to 1.

These screens can be reached as described bellow.

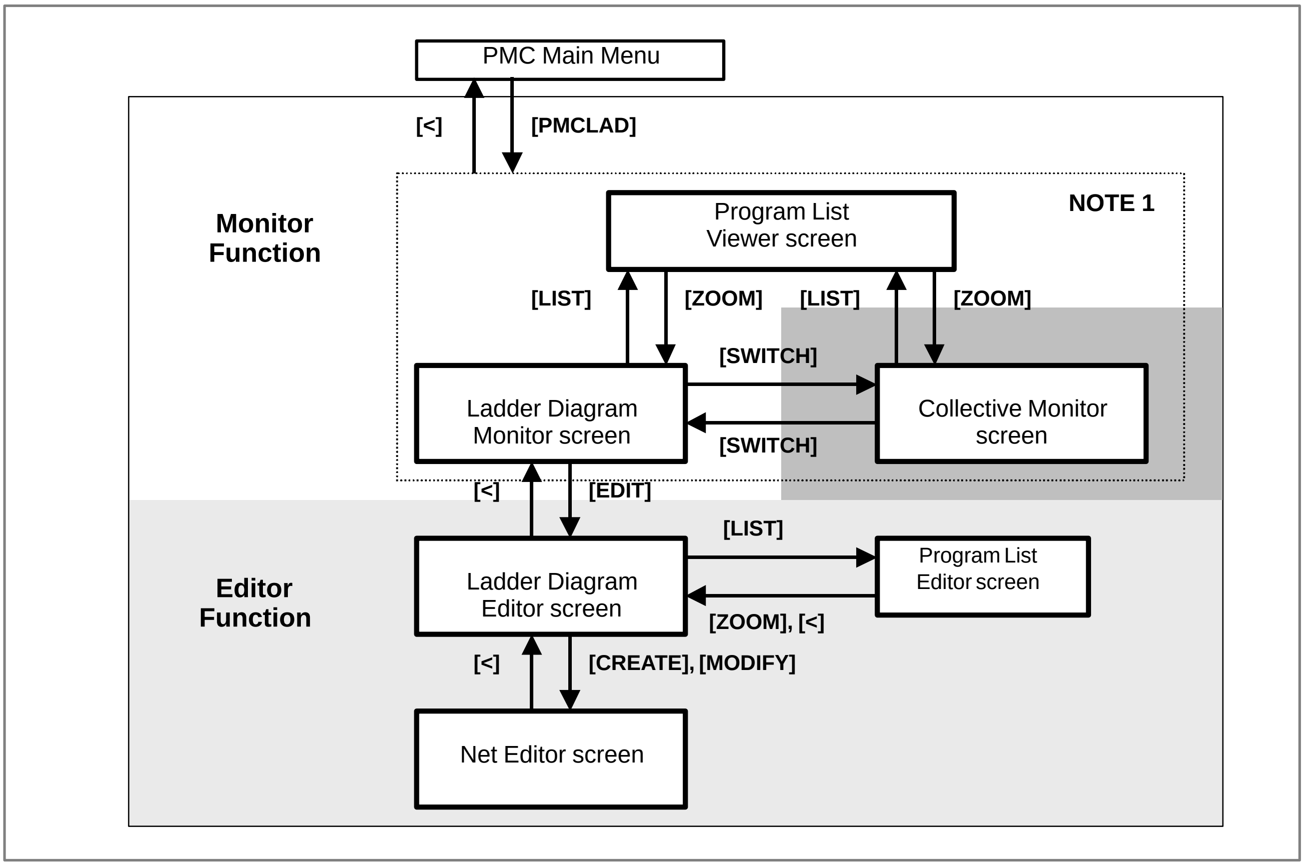


Fig. 3 Relation of screen

NOTE

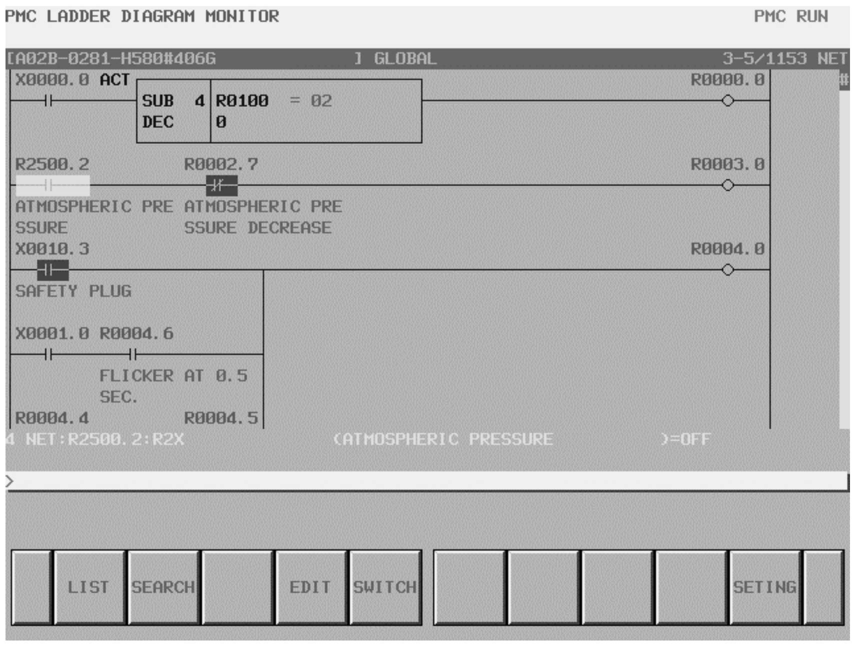
1 When [PMCLAD] soft key is pressed, the screen which was displayed last time among Ladder Diagram Monitor screen, Collective Monitor screen and Program List Viewer screen is displayed. When [PMCLAD] soft key is pressed at first time after CNC power is on, Program List Viewer screen is displayed. Changing the ladder program by I/O function will also make Program List Viewer screen appears at [PMCLAD] soft key again. See ”Program List Viewer screen” for more detail.

2 [EDIT] soft key in Ladder Diagram Monitor screen appears only when Programmer function is enabled. To enable Programmer function, set ”PROGRAMMER ENABLE” to ”YES” at GENERAL screen of PMC Settings or set keep relay K900.1 to 1. Or, set ”EDIT ENABLE” to ”YES” or set K901.6 to 1. While Online Monitor function is active, you can not reach Ladder Diagram Editor screen. To use Ladder Diagram Editor function, you have to disable Online Monitor function at ”RS–232C” and ”F–BUS” on ”PARAMETERS FOR ONLINE MONITOR”: choose ”NOT USE”.

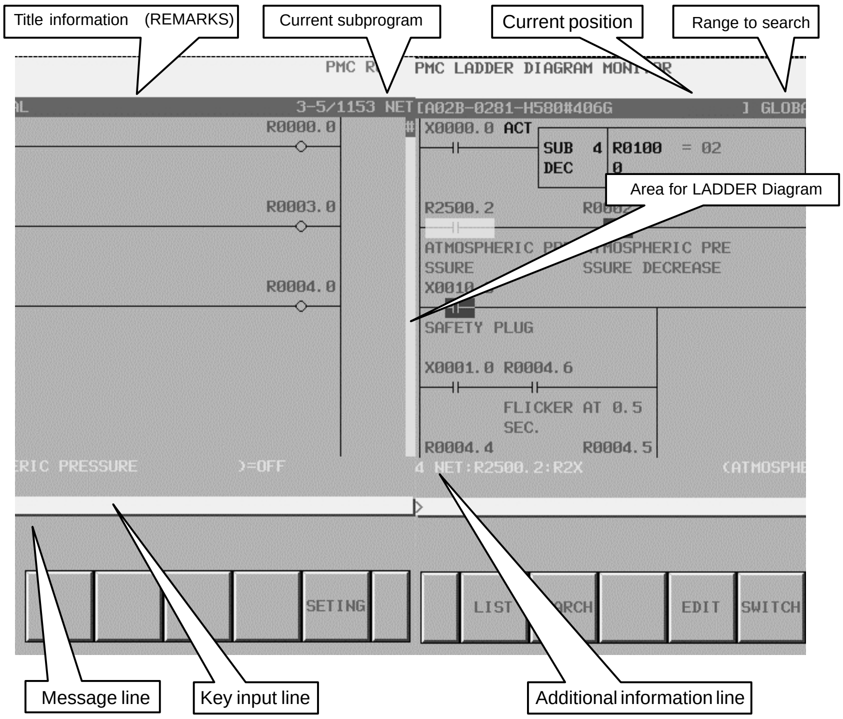
3.1 LADDER DIAGRAM MONITOR SCREEN

LADDER Diagram Monitor screen shows the on/off status of contacts and coils, and the contents of address specified for parameter of functional instructions.
Press the [PMCLAD] soft key on the PMC main menu to call the ladder diagram screen. You can use following operation at this screen, including ”Forced I/O function (Forcing mode)”, by which you can force the relay or the address parameters of functional instructions to a new status or value.

- Switch subprogram to show [LIST]
- Search for address or others [SEARCH]
- Show data table of functional instructions [TABLE]
- Go to LADDER Diagram Editor screen [EDIT]
- Calling collective monitor screen [SWITCH]
- Forced I/O function (Forcing mode) ”number” + INPUT key



3.1.1 Screen Structures



(a) Screen Structure

1 Title information (REMARKS) of the LADDER program, the current subprogram, and the current position information of the Diagram displayed in this screen, are displayed above the LADDER Diagram.
When you select a subprogram to be displayed, range for search function is indicated at right of the top line as ”LOCAL” or ”GLOBAL”. In case of ”LOCAL”, the range for search function is restricted within the current subprogram. In case of ”GLOBAL”, on the other hand, search function searches whole of LADDER program, and switch current subprogram automatically according to the result of searching.

2 In the additional information line near the bottom of the screen, the following information of the address under the cursor when the cursor is shown.

- Net number of the net with the cursor
- Address and its symbol and comment information
- Current value

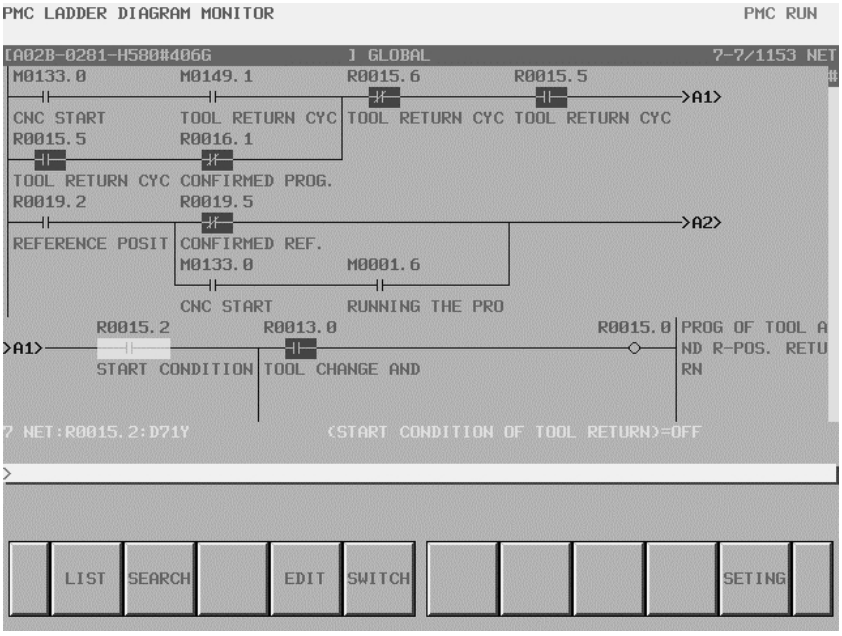
3 In the message line, error messages or inquiry messages will be displayed depending on the situation.

4 In the ladder diagram display area, relays can be displayed in one of the following patterns: 9 by 8, 9 by 9, 8 by 6, 9 by 6, 8 by 5, and 9 by 5 (number–of–relays–in–row by number–of–relays–in–column). For details, see ”Setting the screen.”

5 A gage is displayed at the right–hand side of the screen. This gage indicates the current display position relative to the entire ladder program.

(b) LADDER Diagram

1 Nets wider than the screen width are displayed as ”Continuous Net” using continuous marks (”>A1>”). Same continuous marks mean they are connected with each other.



(c) Monitor

1 Contacts and coils are displayed in different colors according to the status of the signal. The status of power flow is not displayed.

2 The contents of address parameters of functional instructions are not shown in default setting. With the extended functional instruction format, you can see the contents of the parameters. See ”Setting Screen” for the detail.

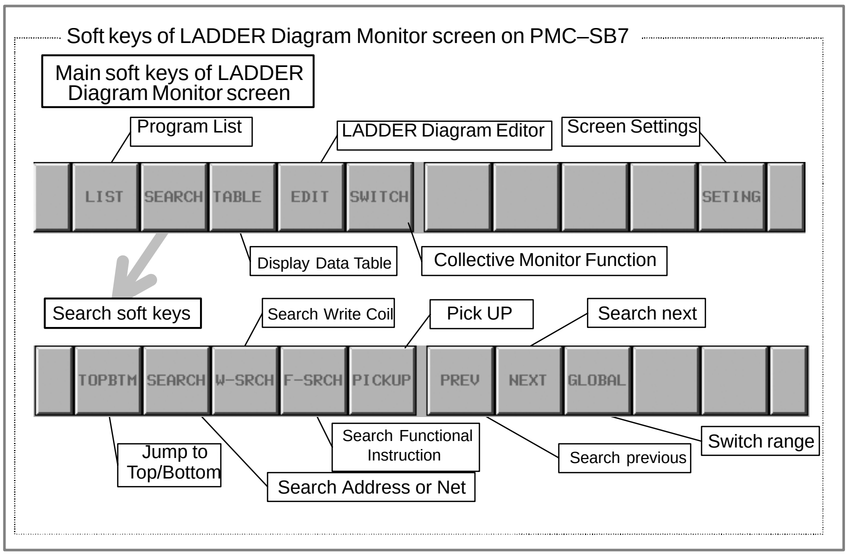
(d) Displaying Symbols and Comments

1 Above each of contacts and coils, the address is displayed. For an address assigned a symbol, you can specify that the symbol is displayed instead of the address. You can also specify that the symbol is displayed in color. For details, see ”Setting the screen.”

2 When a comment is set for the address of a contact, it is displayed below the contact. You can specify the display format of the comment. You can also specify that the comment is displayed in color. For details, see ”Setting the screen.”

3 When a comment is set for the address of a coil, it is displayed in the right margin of the screen in the wraparound mode. You can specify that this area is used to display a relay instead of a comment (to increase the number of relays that can be displayed in a row). You can also specify that the comment is displayed in color. For details, see ”Setting the screen.”

3.1.2 Operations



(a) Operation with Soft–keys

1 [LIST] Go to Program List Viewer screen
Goes to Program List Viewer screen to choose subprogram to be displayed at LADDER Diagram Monitor screen.

2 [SEARCH] Search & Jump
Change soft keys to ”Search soft keys”. Use Return key [<] to return to the ”Main soft keys”. ”Search soft keys” consists of followings:

- [TOPBTM] Jump to Top/Bottom
Jumps to the top of LADDER Program. If the top is displayed already, then jump to the bottom.

- [SEARCH] Search Address/Net
Searches the PMC address or the net according to the preceding string. You can specify both of bit address and byte address.
When digits are entered, the digits are supposed to be a net number and the screen will jump to the net of the net number.
When a string other than digits is entered, the string is examined as a symbol for some PMC address at first. If the string matches a symbol, then the address that the symbol means will be searched for.
If no symbol matches the string, then the string is examined as PMC address at next. If the string indicates correct PMC address, then the address will be searched for.
When cursor is hidden, the net that has the specified net number or contains the specified address will be shown at the top of the screen.
When cursor is shown, the cursor moves to the relay or the parameter to show the found address directly.

- [W–SRCH] Search Write Coil
Searches for the write coils with the address that entered string means. Any contacts with the address are ignored.

- [F–SRCH] Search Functional Instruction
  Searches for the functional instructions by its SUB number or its mnemonic name such as ”TMR” or ”END2”.
- [PICKUP] Taking of ladder net into collective monitor screen
  The ladder net which executes the monitor is taken into the collective monitor screen.
- [PREV] Search previous
  Repeats to search the same thing backward (upward).
- [NEXT] Search next
  Repeats to search the same thing forward (downward).
- [GLOBAL] / [LOCAL] Change range for searching
  Changes the range for searching between GLOBAL and LOCAL; GOBAL means whole of program, and LOCAL means within the displaying subprogram. Current range for searching is indicated at right of the information line at top of screen.

3 [TABLE] Go to Functional Instruction Data Table Viewer screen
Goes to Functional Instruction Data Table Viewer screen to examine contents of Data Table of functional instructions such as COD (SUB 7) and CODB (SUB 27), which have Data Table in themselves. This soft key appears only when the cursor is on a functional instruction that has Data Table.

4 [EDIT] Go to LADDER Diagram Editor screen
Goes to LADDER Diagram Editor screen. This soft key appears only when Programmer function[1] is enabled. And activating Online Monitor function[2] disables this soft key.

NOTE
1 To enable Programmer function, go to GENERAL screen of PMC Settings, and choose ”YES” at item ”PROGRAMMER ENABLE”.
2 To deactivate Online Monitor function, go to ONLIEN screen of PMC Settings, and choose ”NOT USE” at settings of ”RS–232C” and ”F–BUS” (if appears).

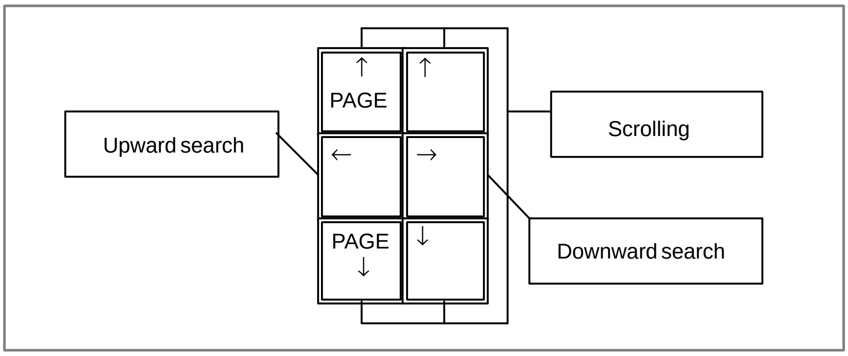
5 [SWITCH] Calling collective monitor screen
The screen display is switched into the Collective Monitor screen.

6 [SETING] Screen settings
Goes to setting screen for LADDER Diagram Monitor screen. You can change various settings for LADDER Diagram Monitor screen at the screen. Use Return key [<] to return to LADDER Diagram Monitor screen. See ”Settings for LADDER Diagram Monitor” section for the detail.

(b) Other operations

1 Cursor move keys, Page change keys
With cursor hidden, you can move diagram in the screen by up/down Cursor move keys and Page change keys.

Operations of the cursor movement keys when the cursor is not displayed



With cursor displayed, you can move the cursor by all cursor move keys and Page change keys. When cursor is placed on some relay or some address parameter of a functional instruction, the information about the address under cursor is displayed at ”Additional Information Line”.

2 ”number” + INPUT key
When cursor is shown, you can force the value of the address under cursor by entering new value as ”number” + INPUT key. In this screen, Forced I/O function is limited only to Forcing mode. This Forced I/O function asks you to confirm your intention before it takes effect. Once it is confirmed that you actually want to change value by this function, you can change the value of the same relay or parameter without further confirmation. However, after you move cursor or you operate other functions, you will be asked when you use the Forced I/O function again.

WARNING

1 You have to pay special attention to use Forced I/O function to change status of signals. Inappropriate use of Forced I/O function may cause unexpected reaction of machine. You have to make it sure that nobody is near the machine when you use this function.

2 As you use Forcing mode of Forced I/O function to change status of signal, however, the signal may look proof against Forced I/O function, because LADDER program or I/O device writes into the signal repeatedly. In this case, even if the signal looks unchanged, actual signal may be changed in very short moment. You should be careful for the reaction of machine to such signal changes.

NOTE

1 Forced I/O function is enabled when ”RAM WRITE ENABLE” setting is set to ”YES” in GENERAL screen of PMC Settings. If the setting is ”NO”, INPUT key will be just ignored.

2 Parameters of timer functional instructions, TMR, TMRB, and TMRC, which have special monitor formats, are not supported by Forced I/O function.

(c) Notes for Search function

1 The string followed by [SEARCH] is treated as symbol first. In case that the symbol ”D0” is assigned to the bit address ”R0.0”, the operation ”D0” + [SEARCH] will search the bit address ”R0.0”, instead of byte address ”D0”.[3]

2 To search the symbol that consists of only digit characters, which will usually be treated as net number, you can use leading space to specify explicitly the string is symbol. For example, while ”123” + [SEARCH] will search the 123rd net from top of the LADDER, ”_123” + [SEARCH] (”_” is space) will search address with symbol ”123”.

3 When the range for searching is GLOBAL, and the target is found in other than displaying subprogram, the screen will automatically switch to the subprogram to which the found target belongs. Searching GLOBAL net number that current subprogram does not contain, for example, the subprogram that contains the net of the net number will appear in the screen, displaying the net.

(d) Shortcuts

1 Right/left cursor move keys that are following some string means searching forward/backward. You can use following strings for this search operation:

- Digits for net number
- ”1” for top, ”0” for bottom
- ”S” + digits, for Functional instructions
- Other string for symbol or bit/byte address
- Leading space always makes the string treated as a symbol or address.

  Example: ”_123”+[SEARCH] (”_” means space) will search the symbol ”123” instead of the 123rd net.

2 When cursor is hidden, left/right cursor move keys without string act just like [PREV]/[NEXT] soft keys.

3 String followed by [SEARCH] soft key in ”PMCLAD Main soft keys” starts searching directly. In this case, [SEARCH] soft key acts just like right cursor move key.

4 [SEARCH] soft key in ”Search soft keys” without string searches the address or the functional instruction under cursor forward. If cursor is hidden, or cursor is placed neither on a relay nor on a functional instruction, this operation just repeats the last successful search forward, just like [NEXT] soft key.

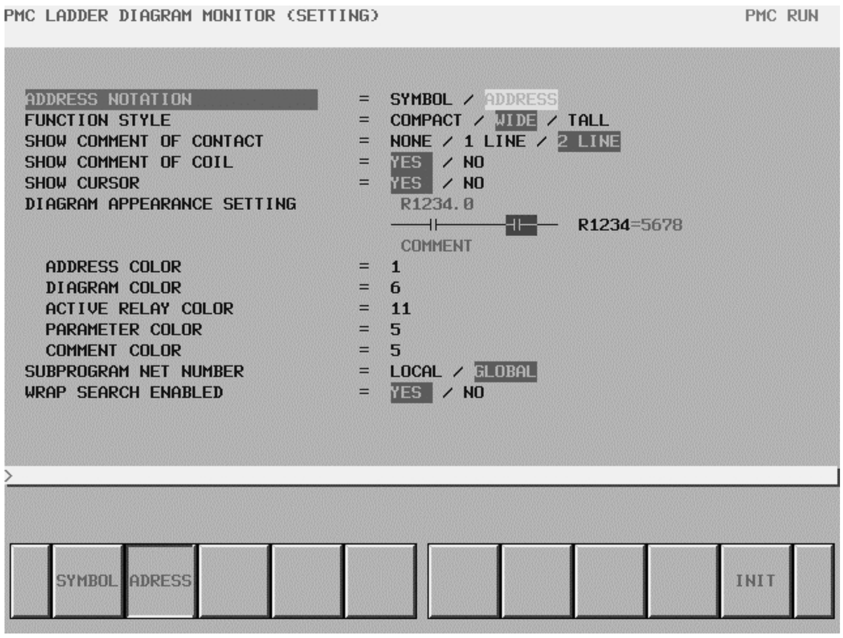
5 [W–SRCH] soft key without string searches forward a write coil of the same address with relay under cursor. If cursor is hidden, or cursor is not placed on a relay, this operation will search a write coil of the bit address that is searched at last successful search. If the last search was not made with bit address, the last entered string for searching is used to determine what bit address is to be searched for a write coil.

6 [F–SRCH] soft key without string searches forward the same functional instruction with one under cursor. If cursor is hidden, or cursor is not placed on a functional instruction, this operation will search a functional instruction that is searched at last successful search. If the last search was not made for functional instruction, the last entered string for searching is used to determine what functional instruction is to be searched.

7 [LIST] soft key following string that indicates subprogram, switches subprogram on LADDER Diagram Monitor screen. Examples for strings to specify subprogram are following:

”L1”	Level 1
”P10”, ”10”	Subprogram ”P10”
”0”(zero), ”G”	Whole of LADDER program (Global)

3.1.3 Setting the Screen



(a) Setting items

LADDER Diagram Monitor Setting screen contains the setting items below:

- ADDRESS NOTATION
  Determines whether the bit and byte addresses in the LADDER Diagram are displayed as corresponding symbols, or the addresses themselves.

SYMBOL

Addresses that have a symbol are displayed by the symbols. Addresses without symbols are displayed by the addresses themselves.

ADDRESS (default)

All addresses are displayed as the addresses themselves even if they have a symbol.

- FUNCTION STYLE

Change the shape of functional instructions. There are three options as below. You have to choose other than ”COMPACT” to show the current values of address parameters of functional instructions.

COMPACT

Occupies least space in diagram. Monitors of current values of address parameters are omitted.

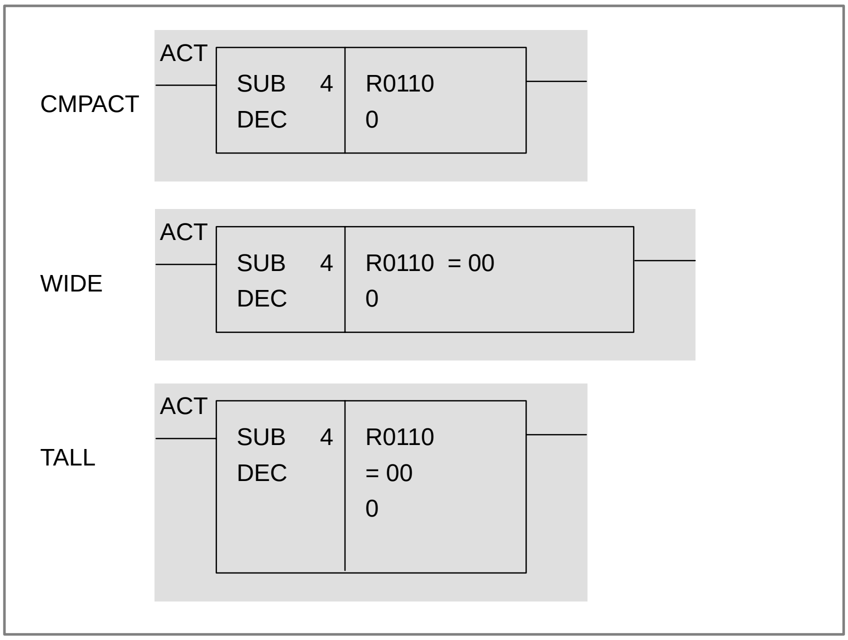
WIDE (default)

Extends the box horizontally to reserve spaces for the monitors of current values of address parameters. The box becomes wider than COMPACT.

TALL

Extends the box vertically to reserve spaces for the monitors of current values of address parameters. The box becomes taller than COMPACT.

Display styles of functional instructions



The displays of current values of address parameters change their format according to each parameter. Refer to the table in ”Display Format for Parameters” for detail.

When you place the cursor on an address parameter, its current value is displayed in the ”Additional Information Line” in both formats of binary decimal, and BCD (or hexadecimal binary).

- SHOW COMMENT OF CONTACT

  Changes the display format of the comment displayed under each contact.

  NONE

  Displays no comment under each contact. In this mode, more contacts (8 by 9 or 9 by 9 (number–of–contacts–in–row by number–of–contacts–in–column)) can be displayed on the screen by the space for the comment.

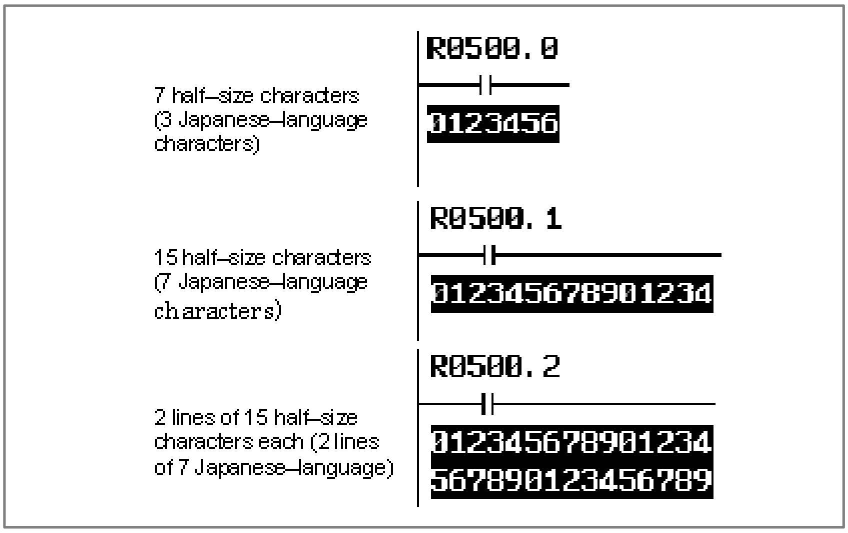
  1 LINE

  Displays the comment in 1 line of 15 half–size characters (in 1 line of 7 Japanese–language characters) under each contact. The width of each contact and the number of contacts that can be displayed on the screen vary depending on the number of characters in each comment. The number of contacts that can be displayed on the screen is from 4 by 6 to 9 by 6 (number–of–contacts–in–row by number–of–contacts–in–column).

  2 LINES (default)

  Displays the comment in 2 lines of 15 half–size characters each (in 2 lines of 7 Japanese–language characters each) under each contact. The width of each contact, the number of lines for each comment, and the number of contacts that can be displayed on the screen vary depending on the number of characters in each comment. The number of contacts that can be displayed on the screen is from 4 by 5 to 9 by 5 (number–of–contacts–in–row by number–of–contacts–in–column).

Comment display formats

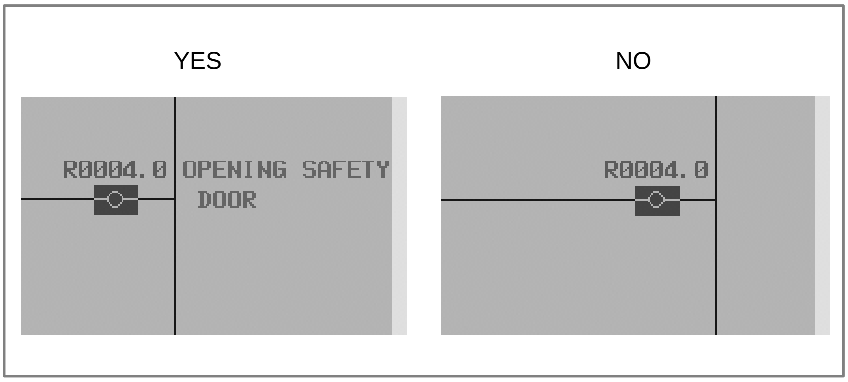


- SHOW COMMENT OF COIL
  Determines whether to show coil comments.

  YES (default)
  Right margin of 14 characters is reserved for display of coil comments. You can set

  NO
  Right margin is used to extend diagram by one more relay, instead of showing coil comment. The screen position bar is also displayed at the right edge of the screen in this option.

Show Coil comment setting



- SHOW CURSOR
  Determines whether to show cursor.

  YES (default)
  Cursor is displayed. Cursor move keys will move the cursor. When the cursor is placed on bit or byte addresses, the information of the address is displayed at ”Additional Information Line”. When you search something with the cursor displayed, the cursor goes directly where it is found. This option is recommended for search operation with LADDER program that contains many large nets.

  NO
  Cursor is not displayed. Up/down cursor move keys will scroll the contents of screen directly. When you search something with the cursor hidden, the net, which contains it, will appear at the top of the screen.

- DIAGRAM APPEARANCE SETTING
  Shows how the ladder diagram is displayed. You can set the colors of the lines, relays, symbols, comments, and function command parameters that are components of the ladder diagram. At right, the monitor display of a symbol, OFF contact, ON contact, function command parameter, and comment appears as an example. The display of this example is changed according to the settings.
  For colors, you can set the display color for each of the five types of components of the ladder diagram.

  ADDRESS COLOR
  Sets the color of symbols and addresses. Enter a number or increase or decrease the number using the left and right arrow keys. You can specify one of 14 numbers from 0 to 13.

DIAGRAM COLOR

Sets the color of the entire ladder diagram. Set the color in the same way as for symbol color.

ACTIVE RELAY COLOR

Sets the color of relays in the on state. The color of the relays in the off state is the same as for the ladder diagram. Set the color in the same way as for Symbol color.

PARAMETER COLOR

Sets the color of the monitor display of function command parameters. The monitor display appears only when a value other than ”Compact” is set for Function command display form. Set the color in the same way as for Symbol color.

COMMENT COLOR

Sets the color of comments. Set the color in the same way as for Symbol color.

- SUBPROGRAM NET NUMBER

Determines whether a net number is counted as ”LOCAL” starting from the top of current subprogram, or is counted as ”GLOBAL” starting from the top of whole program. This setting also affects the expression of net number at searching nets by number.

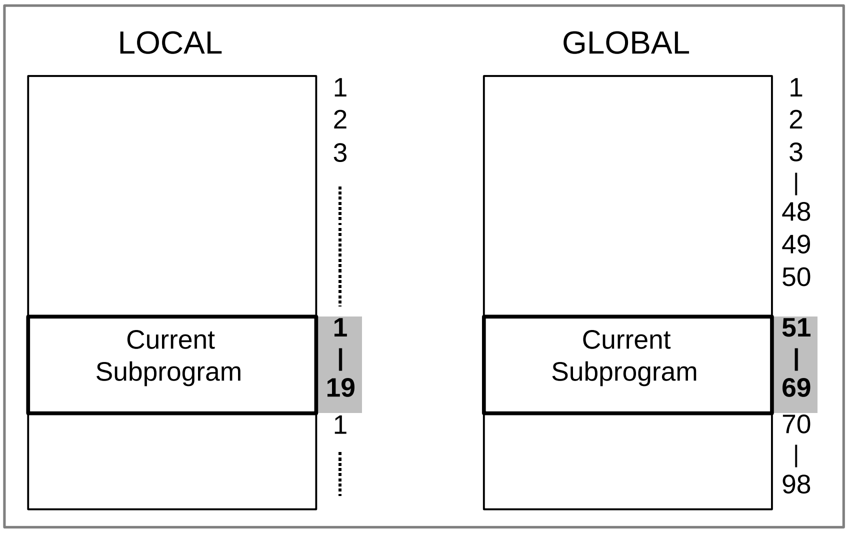
LOCAL

Net number starts from 1 at top of current subprogram. Net number is defined only within current subprogram. The net number information at upper right of the screen is displayed in the format ”displaying range / nets in subprogram NET”.

GLOBAL (default)

Net number starts from 1 at top of Level 1 program. Net number is defined identically at whole of program. The net number information at upper right of the screen is displayed in the format ”displaying range / subprogram range NET”.

Definition of Net number



- WRAP SEARCH ENABLED

Allows search process to wrap from top/bottom to bottom/top to continue to search.

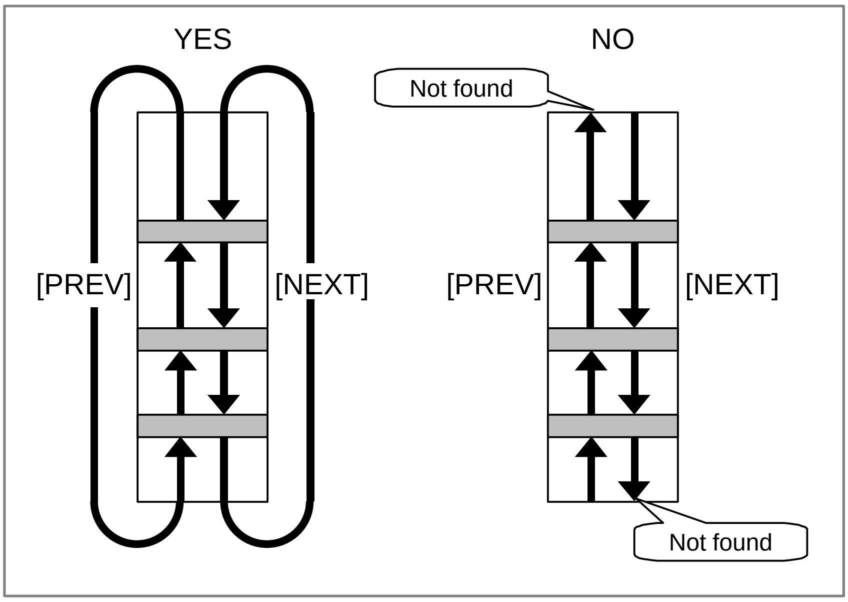
YES (default)

Downward search will continue to search from top of LADDER when reaches to bottom. Upward search will also continue to search from bottom when reaches to top.

NO

Search process will fail when reached top or bottom, and displays an error message at Message Line.

Wrap search



(b) Soft keys

LADDER Diagram Monitor Setting screen has the soft keys of options and following:

- [INIT] Initialize all settings
  All settings will be initialized to the default values.

3.1.4 Display Format for Parameters

The following table shows all monitor formats for each parameter of each functional instruction.

NOTE
1 ”Variable” in ”Monitor Format” field means that this parameter changes its size according to the other parameter. Refer to the descriptions for each functional instruction for detail.
2 Functional instruction with ”*” mark has Data table.

No.	Name	Parameter	Monitor Format
1	END1	–	–
2	END2	–	–
3	TMR	1	Special
4	DEC	1	2–digits BCD
		2	Constant
5	CTR	1	Special
6	ROT	1	Constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD
7	COD *	1	Constant
		2	2–DIGITS BCD
		3	4–digits BCD
8	MOVE	1	Constant
		2	Constant
		3	2–digits HEX
		4	2–digits HEX
9	COM	1	Constant
10	JMP	1	Constant
11	PARI	1	1–byte binary
14	DCNV	1	No monitor
		2	No monitor
15	COMP	1	Constant
		2	4–digits BCD
		3	4–digits BCD
16	COIN	1	Constant
		2	4–digits BCD
		3	4–digits BCD
17	DSCH	1	Constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD

No.	Name	Parameter	Monitor Format
18	XMOV	1	Constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD
19	ADD	1	Constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD
20	SUB	1	Constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD
21	MUL	1	Constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD
22	DIV	1	Constant
		2	4–digits BCD
		3	4–digits BCD
		4	4–digits BCD
23	NUME	1	Constant
		2	4–digits BCD
24	TMRB	1	Special
		2	Constant
25	DECB	1	Constant
		2	Variable binary
		3	Constant
		4	2–digits HEX
26	ROTB	1	Constant
		2	Variable binary
		3	Variable binary
		4	Variable binary
		5	Variable binary

No.	Name	Parameter	Monitor Format
27	CODB *	1	Constant
		2	Constant
		3	1–byte binary
		4	Variable binary
28	MOVOR	1	2–digits HEX
		2	2–digits HEX
		3	2–digits HEX
29	COME	–	–
30	JMPE	–	–
31	DCNVB	1	Constant
		2	No monitor
		3	No monitor
32	COMPB	1	Constant
		2	Constant or Variable binary
		3	Variable binary
33	SFT	1	4–digits HEX
34	DSCHB	1	Constant
		2	Variable binary
		3	Variable binary
		4	Variable binary
		5	Variable binary
35	XMOVB	1	Constant
		2	Variable binary
		3	Variable binary
		4	Variable binary
		5	Variable binary
36	ADDB	1	Constant
		2	Variable binary
		3	Constant or Variable binary
		4	Variable binary
37	SUBB	1	Constant
		2	Variable binary
		3	Constant or Variable binary
		4	Variable binary
38	MULB	1	Constant
		2	Variable binary
		3	Constant or Variable binary
		4	Variable binary
39	DIVB	1	Constant
		2	Variable binary
		3	Constant or Variable binary
		4	Variable binary

No.	Name	Parameter	Monitor Format
40	NUMEB	1	Constant
		2	Constant
		3	Variable binary
41	DISPB*	1	Constant
42	EXIN*	1	8–digits HEX
43	MOVB	1	1–byte binary
		2	1–byte binary
44	MOVW	1	2–bytes binary
		2	2–bytes binary
45	MOVN	1	Constant
		2	4–bytes binary
		3	4–bytes binary
47	MOVD	1	4–bytes binary
		2	4–bytes binary
48	END3	–	–
51	WINDR	1	2–bytes binary
52	WINDW	1	2–bytes binary
53	AXCTL	1	Constant
		2	8–digits HEX
54	TMRC	1	Constant
		2	Special
		3	Special
55	CTRC	1	2–bytes binary
		2	2–bytes binary
56	CTRB	1	Constant
		2	Special
58	DIFD	1	Constant
59	EOR	1	Constant
		2	Variable HEX
		3	Constant or Variable HEX
		4	Variable HEX
60	AND	1	Constant
		2	Variable HEX
		3	Constant or Variable HEX
		4	Variable HEX
61	OR	1	Constant
		2	Variable HEX
		3	Constant or Variable HEX
		4	Variable HEX
62	NOT	1	Constant
		2	Variable HEX
		3	Variable HEX
64	END	–	–
65	CALL	1	No monitor

No.	Name	Parameter	Monitor Format
66	CALLU	1	No monitor
68	JMPB	1	No monitor
69	LBL	1	No monitor
70	NOP	1	Constant
71	SP	1	No monitor
72	SPE	–	–
73	JMPC	1	No monitor

Functional Instructions of Special monitor format

<table>
<tr><td>TMR</td><td>Displays in ”Current/Preset” format by seconds</td></tr>
<tr><td>CTR</td><td>Displays in ”Current/Preset” format by BCD or binary according to the Counter Type setting in LADDER Program.</td></tr>
<tr><td>TMRB</td><td>Displays current value by seconds (Preset value is displayed by milliseconds)</td></tr>
<tr><td>TMRC</td><td>The 2nd parameter shows preset value, and the 3rd parameter shows current value as their monitor displays.
These two monitor displays changes their format according to the 1st parameter as below:
<table>
<tr><th>1st Parameter</th><th>Precision</th><th>Display Format</th></tr>
<tr><td>0</td><td>8 ms</td><td>by seconds</td></tr>
<tr><td>1</td><td>48 ms</td><td>by seconds</td></tr>
<tr><td>2</td><td>1 second</td><td>HH:MM:SS</td></tr>
<tr><td>3</td><td>10 seconds</td><td>HH:MM:SS</td></tr>
<tr><td>4</td><td>1 minute</td><td>HH:MM:ss</td></tr>
<tr><td>5</td><td>1 ms</td><td>by seconds</td></tr>
<tr><td>6</td><td>10 ms</td><td>by seconds</td></tr>
<tr><td>7</td><td>100 ms</td><td>by seconds</td></tr>
</table></td></tr>
<tr><td>CTR</td><td>Displays setting value with binary format.</td></tr>
</table>

3.2 FUNCTIONAL INSTRUCTION DATA TABLE VIEWER SCREEN

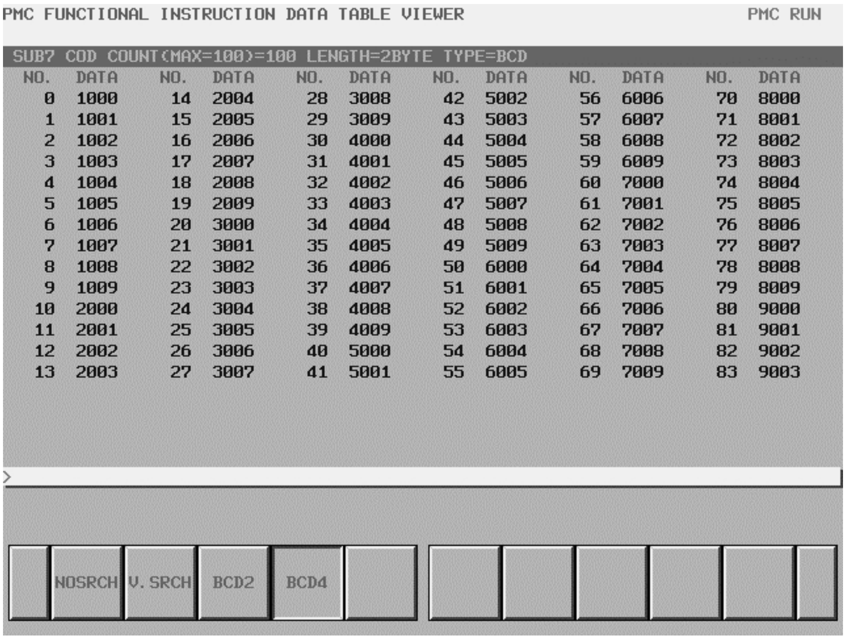
Functional Instruction Data Table Viewer screen shows the contents of data table that belongs to some functional instructions.
To reach this screen, at LADDER Diagram Monitor screen, press [TABLE] soft key that is displayed when the cursor is on the following functional instructions which have a data table.

- Functional Instruction COD (SUB7)
- Functional Instruction CODB (SUB27)

Following operations are available at this screen.

- Search for data table number. [NO.SRCH]
- Search for data value. [V.SRCH]
- Change the displaying data digits. [BCD2],[BCD4]

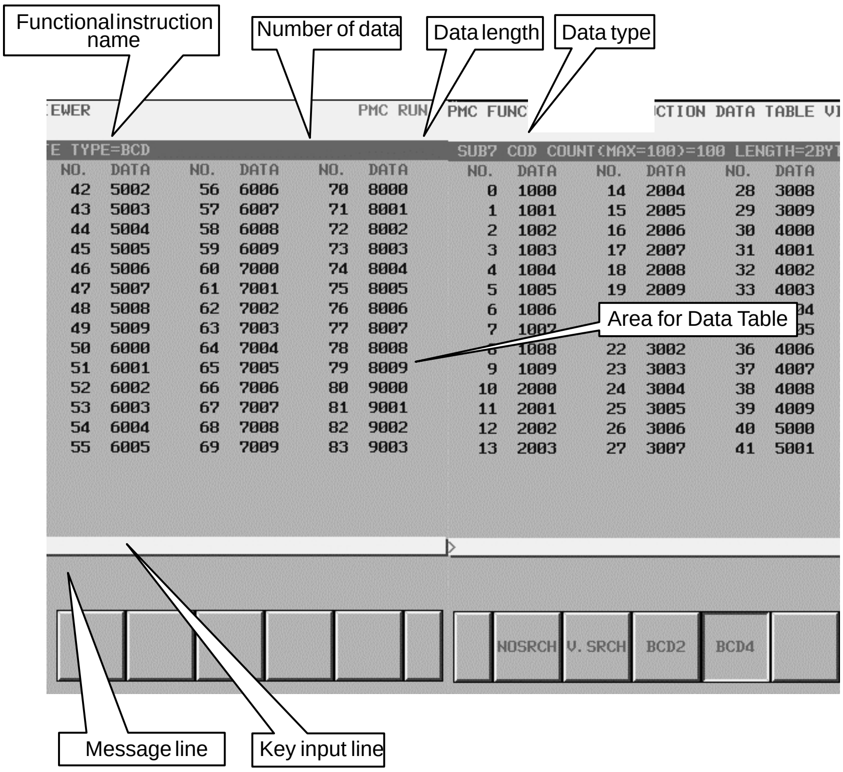
  (These soft keys can be operated only at Functional Instruction Data Table Viewer screen of Functional Instruction COD.)



NO.	DATA	NO.	DATA	NO.	DATA	NO.	DATA	NO.	DATA	NO.	DATA
0	1000	14	2004	28	3008	42	5002	56	6006	70	8000
1	1001	15	2005	29	3009	43	5003	57	6007	71	8001
2	1002	16	2006	30	4000	44	5004	58	6008	72	8002
3	1003	17	2007	31	4001	45	5005	59	6009	73	8003
4	1004	18	2008	32	4002	46	5006	60	7000	74	8004
5	1005	19	2009	33	4003	47	5007	61	7001	75	8005
6	1006	20	3000	34	4004	48	5008	62	7002	76	8006
7	1007	21	3001	35	4005	49	5009	63	7003	77	8007
8	1008	22	3002	36	4006	50	6000	64	7004	78	8008
9	1009	23	3003	37	4007	51	6001	65	7005	79	8009
10	2000	24	3004	38	4008	52	6002	66	7006	80	9000
11	2001	25	3005	39	4009	53	6003	67	7007	81	9001
12	2002	26	3006	40	5000	54	6004	68	7008	82	9002
13	2003	27	3007	41	5001	55	6005	69	7009	83	9003

Functional Instruction Data Table Viewer screen of Functional Instruction COD.

3.2.1 Screen Structures



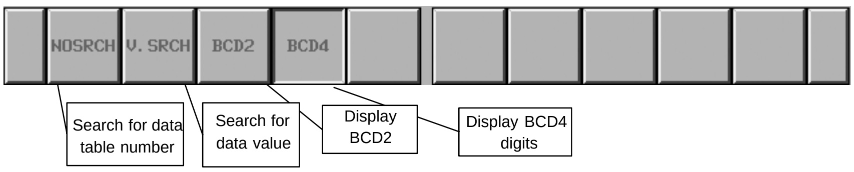
1 The functional instruction name, the number of data table, data length and data types are displayed above the Data Table.

2 In the message line, error messages or inquiry messages will be displayed depending on the situation.

3 In case of functional instruction COD, the data of 6 lines and 14 columns can be displayed in the area for Data table.
In case of functional instruction CODB, the data of 4 lines and 14 columns can be displayed in the area for Data table.

3.2.2 Operations

Soft keys of Functional Instruction Data Table Viewer screen of functional instruction COD.



Soft keys of Functional Instruction Data Table Viewer screen of functional instruction CODB.

1 [NO.SRCH] Search for data table number
Searches the data table number which you specified.
Then the cursor is displayed on the data of the target number.
And the cursor is disappeared when you operate something.

2 [V.SRCH] Search for data value
Searches the data value which you specified.
Then the cursor is displayed on the target data.
And the cursor is disappeared when you operate something.

3 [BCD2] Display BCD2 digits
Changes the display data type to 2 digits of BCD. This operation only switches the display data type. So it does never edit the data on memory.

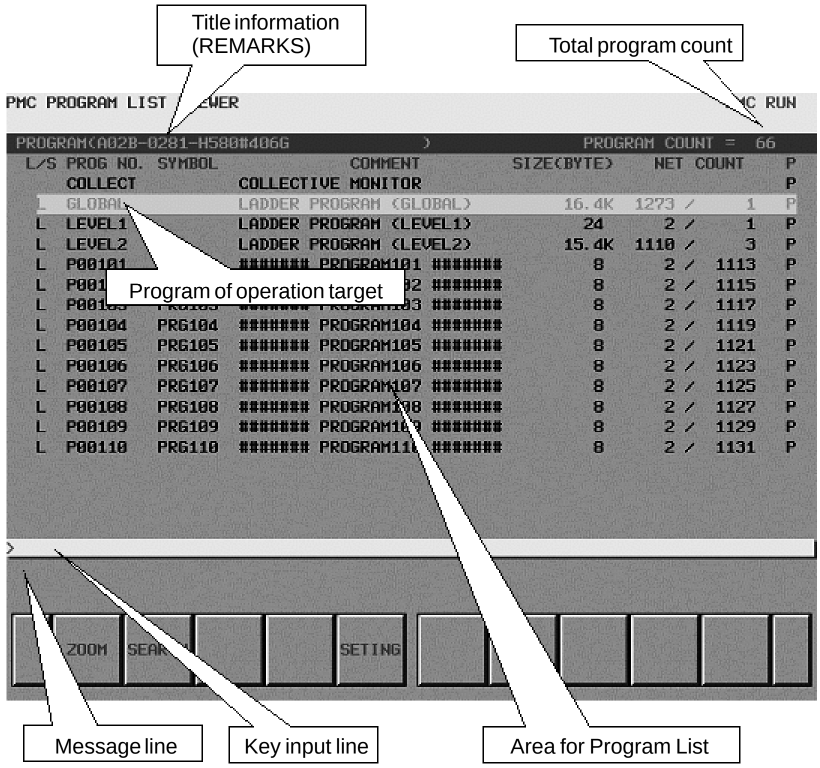
4 [BCD4] Display BCD4 digits
Changes the display data type to 4 digits of BCD. This operation only switches the display data type. So it does never edit the data on memory.

5 Cursor move keys, Page change keys
You can scroll screen by right/left cursor move keys and Page change keys.

3.3 PROGRAM LIST VIEWER SCREEN

3.3.1 Detail Screen

Program List Viewer screen (Detail) shows the detail information that are program size, program net count and so on for program.



(a) Screen Structure

1. The title information (REMARKS) of the Ladder Program and the total program count are displayed above the Program List.

2. In the message line, error messages or inquiry messages will be displayed depending on the situation.
3. In the area for Program List, 14 data can be displayed at maximum.

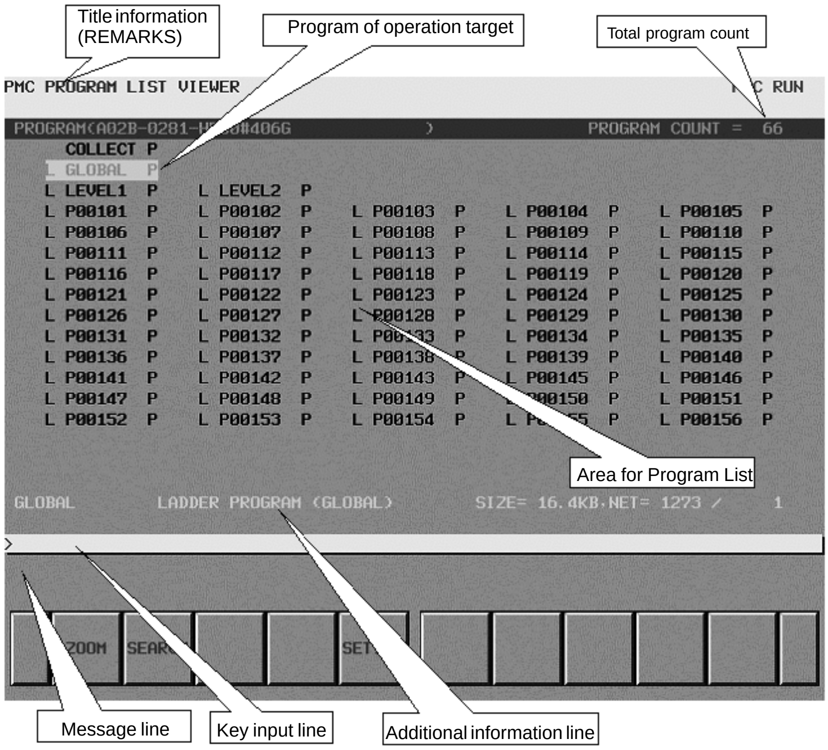
(b) Area of Program List

1. The ”L” means program type of Ladder. It is displayed in the ”L/S” field for each program.
2. Program name is displayed in the ”PROG NO.” field for each program.
   There are three kinds of program names.
   COLLECT : It means the collective program.
   GLOBAL : It means the whole program.
   LEVELn (n=1,2,3) : It means the Ladder level 1, 2 and 3.
   Pm (m=1 to 2000) : It means subprogram.
3. The symbol is displayed in the ”SYMBOL” field for each program. If no symbol is set into the subprogram, the ”SYMBOL” field will be blank.

4. The comment is displayed in the ”COMMENT” field for each program.
   If no comment is set into the subprogram, the ”COMMENT” field will be blank.
5. The program size is displayed in the ”SIZE (BYTE)” field for each program.
   If the program size is not over 1024 byte, the unit is shown in byte. If it is over 1024 byte, the unit is shown in kilo (1024) byte with ”K”.
   Ex.) The case that program size is not over 1024 byte.
   1023 bytes : ”1023” is shown.
   Ex.) The case that program size is over 1024 byte.
   20000 bytes : ”19.5K” is shown.
6. In the ”NET COUNT” field for each program, total number of nets in the program and first net number of the program in the whole Ladder program are displayed as follows.
   Total number of nets in the program* / First global net number of the program in the whole Ladder program*
   * Maximum is 99,999.
7. The status of protection is displayed in the ”P” field for each program.
   Following icons mean the status of protection.
   P : Monitoring and editing program are disabled.
   R : Monitoring program is enabled. Editing program is disabled.
   (Blank) : Monitoring and editing program are enabled.

3.3.2 Brief Screen

Program List Viewer screen (Brief) shows less information than Program List Viewer screen (Detail), to increase the number of items. The program type, the status of protection and the name or symbol are displayed for each program.



The detail information of program that is pointed by the cursor is displayed on additional information line.

(a) Screen Structure

1. The title information (REMARKS) of the Ladder Program and the total program count are displayed above the Program List.
2. In the message line, error messages or inquiry messages will be displayed depending on the situation.
3. In additional information line near the bottom of the screen, the following information is displayed for the program under the cursor.
   - program name
   - symbol, comment
   - program size
   - total number of nets
   - First global net number of the program in the whole Ladder program.
4. In the Program List area, the programs 14 lines and 5 columns can be displayed in maximum.

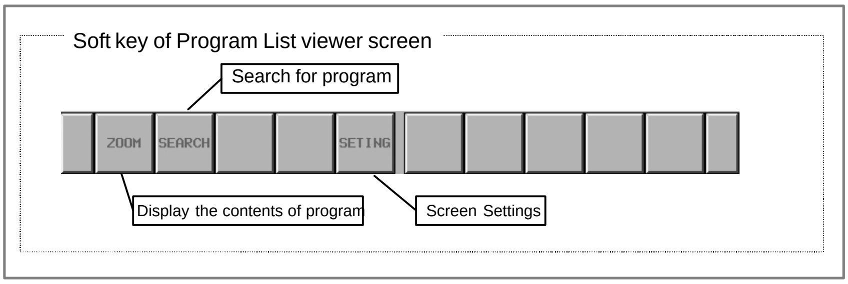
(b) Area of Program List

The following items are displayed in Program List area of Program List Viewer screen.

The meanings of these items are the same as the items of Program List Viewer screen (Detail).

- Program type.
- Status of protection
- Program name or symbol. (See ”Screen Setting for Program List Viewer screen” section for selecting this item.)

3.3.3 Operations



(a) Operation with Soft–keys

1 [ZOOM] Display the contents of program
Goes to LADDER Diagram Monitor screen. If you press [ZOOM] soft key without strings, the program under the cursor is displayed at LADDER Diagram Monitor screen.
If you entered program name (See (c) 1. for detail) or symbol before pressing [ZOOM] soft key, the program according to the preceding string is searched and the program is displayed at LADDER Diagram Monitor screen.
But, when the selected program is protected to monitor, you have to unlock the protection.

2 [SEARCH] Search for program
Searches the program. If you entered program name (See (c) 1. for detail) or symbol and press [SEARCH] soft key, the program according to the preceding string is searched, the cursor points the program.

3 [SETING] Screen Settings
Goes to setting screen for Program List Viewer screen. You can change various settings for Program List Viewer screen at the screen. Use Return key [<] to return to Program List Viewer screen. See ”Settings for Program List Viewer screen” section for the detail.

(b) Other operations

1 Cursor move keys, Page change keys
You can move cursor by all cursor move keys and Page change keys.
And if you entered program name (See (c) 1. for detail) or symbol and press Right cursor move key, the program according to the preceding string is searched, the cursor points the program.

2 INPUT key
You can operate same as [ZOOM] soft key.

(c) Note of searching operations

1 When program is searched by program name, the strings that show each program are as follows.

GLOBAL	: ”0”(Zero) or ”G”	
LEVEL1, 2, 3	: ”L” + Number	Ex.) ”L1”, ”L01”, etc
Pn	: Number or ”P” + Number	Ex.) ”1”, ”P1”, ”P01”

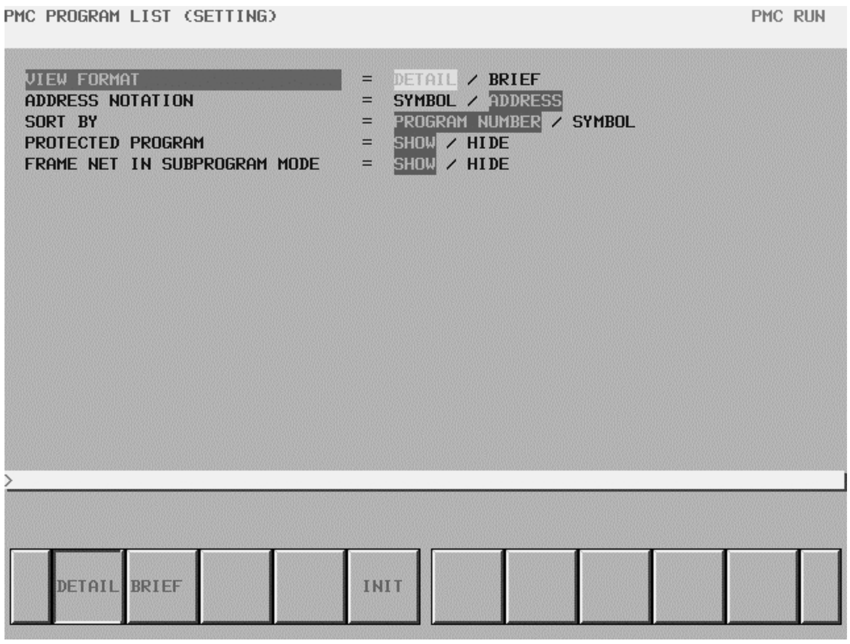
2 Search function by [SEARCH] soft key or Right cursor move key tries to suppose the given word as an item to be searched in following order.

(i) The string for GLOBAL or LEVEL : ”0”(Zero), ”G”, ”L” + Number
The number for subprogram : Number

(ii) Symbol

(iii) The string for subprogram : ”P” + Number

3.3.4 Setting Screen



(a) Setting items

Program List Viewer/Editor screen contains the setting items below:

- VIEW FORMAT
  Determines whether the list data in the Program List Viewer/Editor screen are displayed as detail format, or brief format themselves.

  DETAIL (default)

  The Program List Viewer/Editor screen shows Detail information for each program.

  The items of detail information are program type, program name, symbol, comment, program size, program net count and status of protection.

BRIEF

The Program List Viewer/Editor screen shows less information than DETAIL, to increase the number of items. The program type, the status of protection and the name or symbol are displayed for each program.

The detail information of program that is pointed by the cursor is displayed on additional information line.

- ADDRESS NOTATION

Determines whether the programs in the Program List Viewer/Editor screen are displayed as corresponding symbols, or the addresses themselves.

SYMBOL

Addresses that have a symbol are displayed by the symbols. Addresses without symbols are displayed by the addresses themselves.

ADDRESS (default)

All addresses are displayed as the addresses themselves even if they have a symbol.

- SORT BY

Determines whether the programs in the Program List Viewer/Editor screen are displayed in order of program number, or symbol name themselves. And When ADDRESS NOTAION is SYMBOL, this determination is enabled. So if ADDRESS NOTAION is ADDRESS, the programs are always displayed in order of program number.

PROGRAM NUMBER (default)

The programs are displayed in order of program number.

SYMBOL

Programs that have a symbol are displayed in order of the symbol name. Programs without symbols are displayed in order of program number after programs with the symbols. GLOBAL, LEVEL1, LEVEL2, LEVEL3 are out of target of sort.

- PROTECTED PROGRAM

Determines whether protected programs in the Program List Viewer/Editor screen are displayed or not. On each screen, the protected program means as follows.

On Program List Viewer screen : the program is protected to monitor.

On Program List Editor screen: the program is protected to edit.

SHOW (default)

The protected programs are displayed in the Program List Viewer/Editor screen.

HIDE

The protected programs are not displayed in the Program List Viewer/Editor screen.

- FRAME NET IN SUBPROGRAM MODE
  Frame net means functional instruction END1, 2 and 3 on LEVEL 1,2,3, and functional instruction SP and SPE on subprogram.
  It determines whether the frame net in the LADDER Diagram Monitor/Editor screen are displayed or not, when you selected the program and press [ZOOM] soft key in the Program List Viewer/Editor screen.

  SHOW (default)
  The frame net is displayed in the LADDER Diagram Monitor/Editor screen.

  HIDE
  The frame net is not displayed in the LADDER Diagram Monitor/Editor screen.

4 COLLECTIVE MONITOR FUNCTION (PMC–SB7)

On Collective Monitor screen, a monitor of the necessary ladder nets can be displayed by specifying ladder nets.

4.1 CALLING SCREEN

The calling operation of Collective Monitor screen is as follows.

(1) Calling from Program List Viewer screen
Press soft key [ZOOM] after moving the cursor to ”COLLECT” program position on Program List Viewer screen.

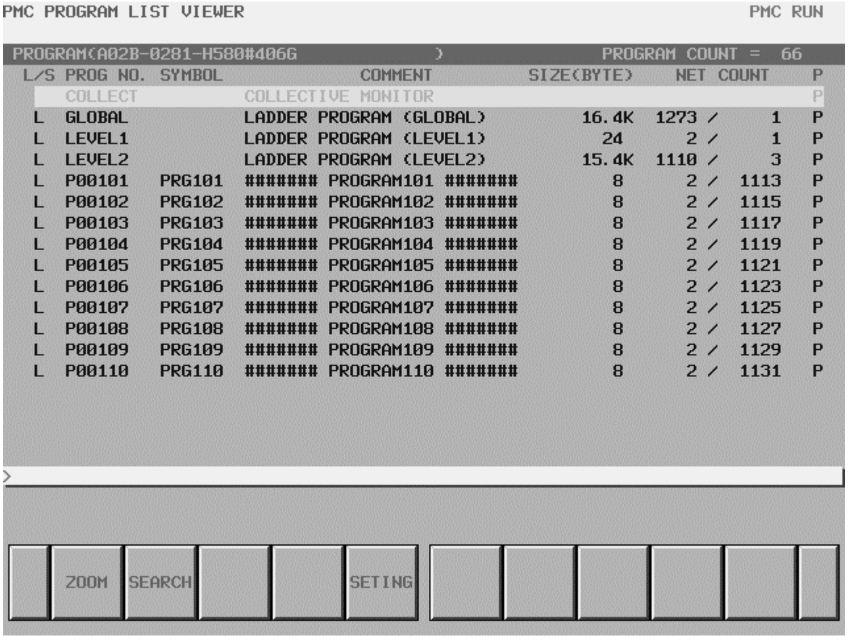


Fig. 4.1(a) Program List Viewer screen

(2) Calling from Ladder Diagram Monitor screen
Press [SWITCH] soft key on Ladder Diagram Monitor screen.

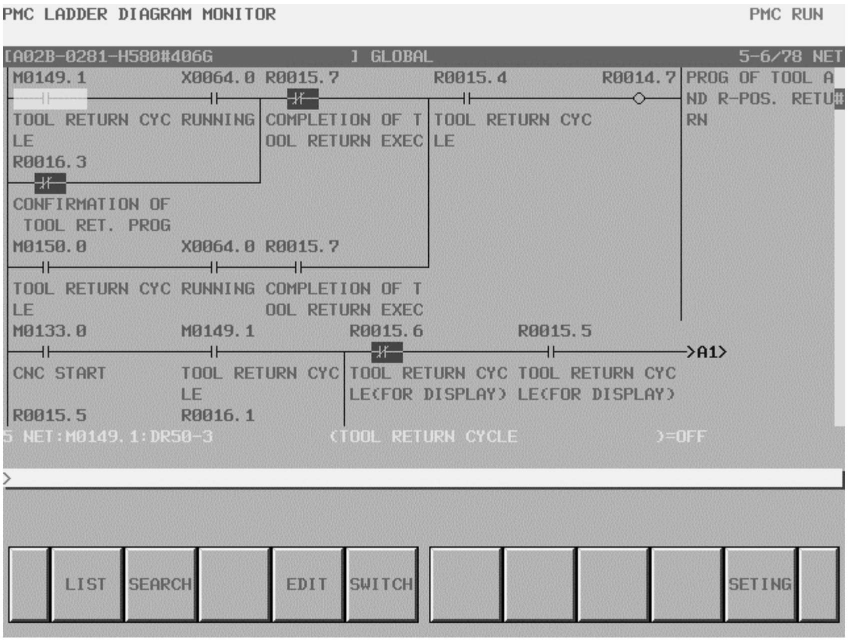


Fig. 4.1(b) Ladder Diagram Monitor screen

4.2 SCREEN CONFIGURATION

Collective Monitor screen is as follows. Any ladder diagram is not displayed first. Ladder nets selected by the operation of a coil search and picking up nets is added. 128 nets or less of ladder net can be added on Collective Monitor screen. When adding more new nets, most recently added 128 nets are displayed.

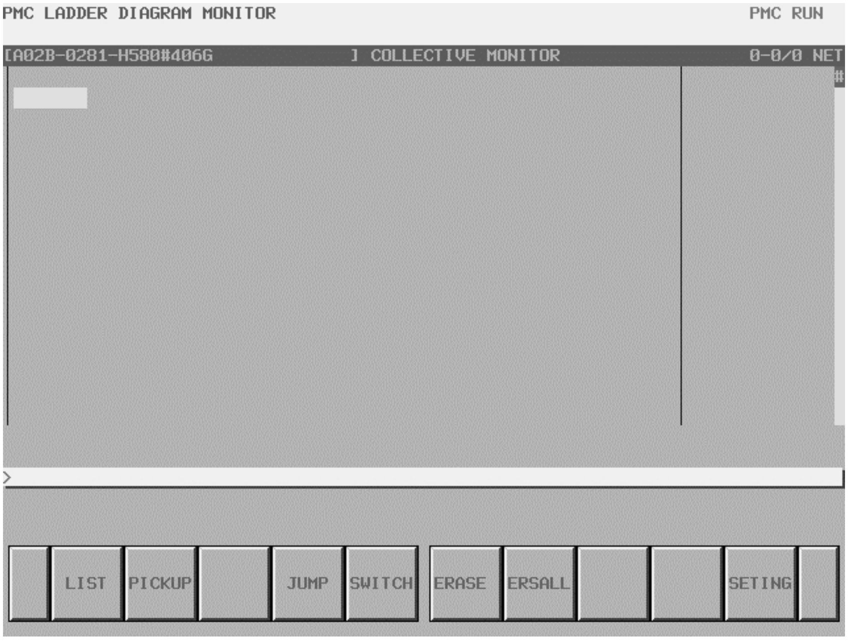


Fig. 4.2 Collective Monitor screen (Initial screen)

4.3 SCREEN OPERATIONS

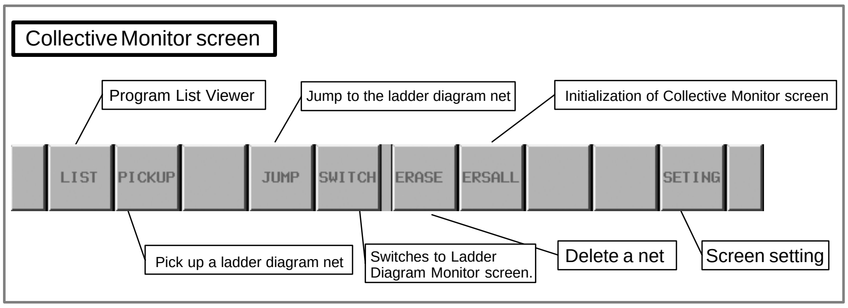


Fig. 4.3(a) Soft key of the Collective Monitor screen

(a) Operations using the soft keys

1. [LIST] Calling Program List Viewer screen
   Calls Program List Viewer screen. On Program List Viewer screen, you can switch subprograms to be displayed on Ladder Diagram Viewer screen.
2. [PICKUP] Picking up ladder nets.
   Picks up ladder nets with coil which you want to monitor, on Collective Monitor screen.
3. [JUMP] Jump to a ladder net.
   On Ladder Diagram Monitor screen, search the ladder net at a cursor position on Collective Monitor screen and jumps to the ladder net.
4. [SWITCH] Switches to Ladder Diagram Monitor screen.
   Switches to Ladder Diagram Monitor screen.
5. [ERASE] Quits to display a ladder diagram net on the screen. (1 net)
   Quits to display a ladder diagram net (only 1 net) which is picked up on Collective Monitor screen.
6. [ERSALL] Quits to display a ladder diagram net on the screen. (all nets)
   Quits to display ladder diagram nets (all net) which is picked up on Collective Monitor screen.
7. [SETING] Screen settings
   Calling the setting screen for Collective Monitor screen. You can change each settings of a ladder diagram display. Return to Collective Monitor screen when press the return key [<].

- Specification of monitored ladder diagram
  The operation for picking up ladder nets which you want to monitor on Collective Monitor screen is as follows.
  1. Specification of ladder nets on Collective Monitor screen
     – Specify the address by key input
       Pick up ladder nets by inputting the address used with the coil.
     – Specify the address from ladder diagram net on Collective Monitor screen
       Specify any relay by cursor on the ladder diagram net which is picked up already. The net, which uses the relay which you specified for coil, is picked up.

2. Specify the ladder net on Ladder Diagram Monitor screen
   Specify a ladder diagram net on Ladder Diagram Monitor screen and the net is picked up on Collective Monitor screen.

- Pick up a ladder net on Collective Monitor screen
  You can pick up a ladder net on Collective Monitor screen. The operation for picking up ladder nets is as follows.
  a) Specification of address
     1. Enter the address which you want to monitor. (Ex. R14.7)
     2. Press [PICKUP] soft key.
     3. The net with the coil, which you specified by 1, is picked up to the top of screen.
  b) Specification of address from the ladder net on the screen
     1. Move the cursor to a relay on the ladder net which uses address that you want to monitor.
     2. Press [PICKUP] soft key.
     3. The net with the coil, which uses the address that you specified by 1, is picked up to the top of screen and the cursor moves to the specified coil position.

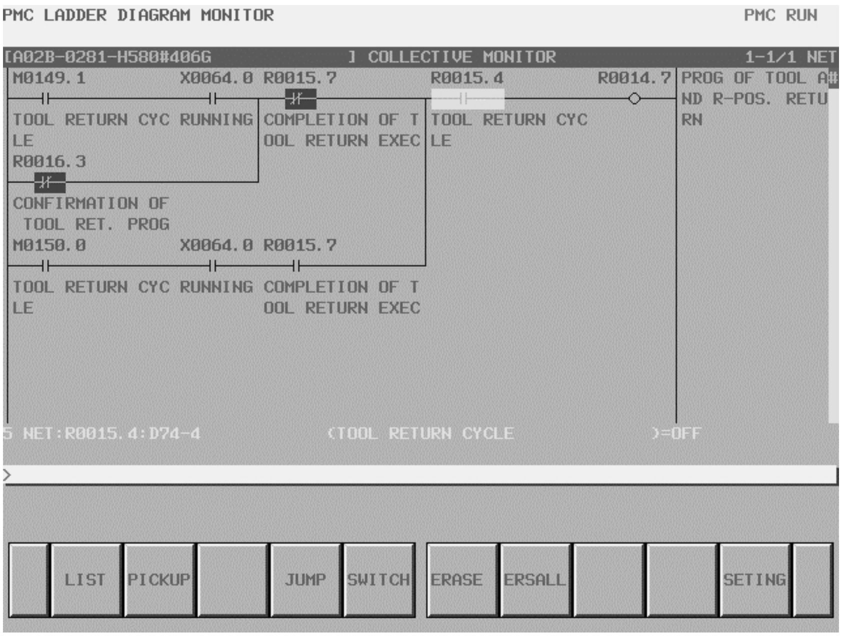


Fig. 4.3(b) Collective Monitor screen

- Pick up a ladder net on Ladder Diagram Viewer screen
  You can pick up a ladder net on Ladder Diagram Viewer screen. The operation for picking up ladder nets is as follows.
  1. Press [SEARCH] soft key on Ladder Diagram Viewer screen. Then soft keys for search are displayed.
  2. Move the cursor to any ladder nets which you want to pick up.
  3. Press [PICKUP] soft key and the net, which is specified by 2, is picked up to a top of Collective Monitor screen.
  4. As for the ladder net which is picked up to Collective Monitor screen, "●" mark is displayed at the left side of the selected net.

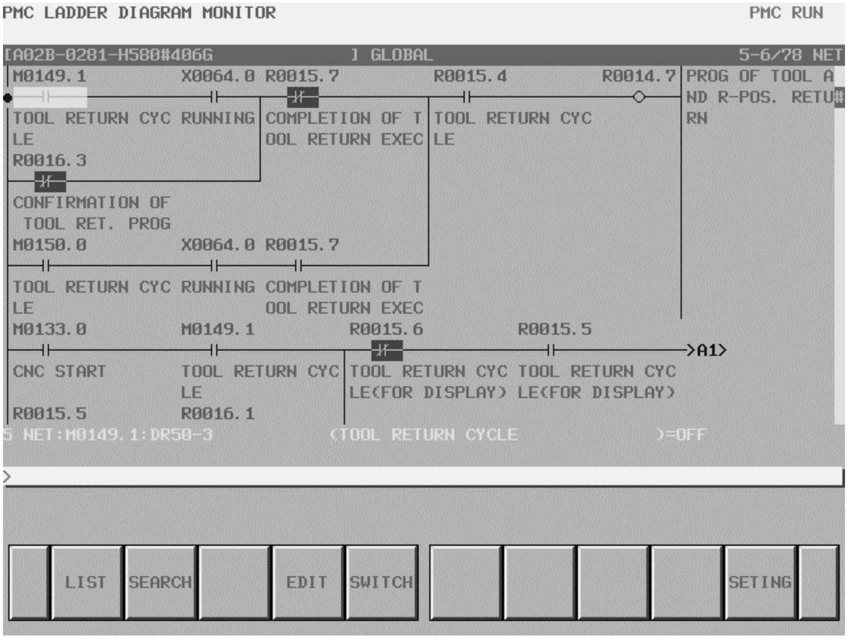


Fig. 4.3(c) Ladder Diagram Viewer screen (Soft key for search)

4.4 ALARM MESSAGE AND COUNTERMEASURE

Message that may be displayed during Collective Monitor function

Alarm number	Faulty location/corrective action	Contents
THE ADDRESS IS NOT FOUND	Specify the address used with write coil	Specified address was not used for the write coil.
SYMBOL UNDEFINED	Specify the defined symbol or address.	Specified symbol was not defined.
WRITE COIL NEEDS BIT ADDRESS	Specify the bit address	The byte address was specified when specifying the address used with write coil.
SOME NETS ARE DISCARDED	Not all the nets of picking up object can be picked up. Select the net of picking up on Ladder Diagram Viewer screen and pick up the net manually.	All nets were not picked up because there were the nets of picking up object more than 128 nets.

5 LADDER DIAGRAM EDITOR FUNCTION (PMC–SB7)

5.1 LADDER DIAGRAM EDITOR SCREEN

In Ladder Diagram Editor screen you can edit Ladder program to change its behavior. To reach Ladder Diagram Editor screen, press [EDIT] soft key at Ladder Diagram Monitor screen. Following operations are available at Ladder Diagram Editor screen.

– Delete by net	[DELETE]
– Move by net	[CUT] & [PASTE]
– Copy by net	[COPY] & [PASTE]
– Change address of contacts and coils	”bit address” + INPUT key
– Change parameters of functional instructions	”number” or ”byte address”+ INPUT key
– Add new net	[CREATE]
– Change construction of net	[MODIFY]
– Make changes effective	[UPDATE]
– Abandon changes	[RESTORE]

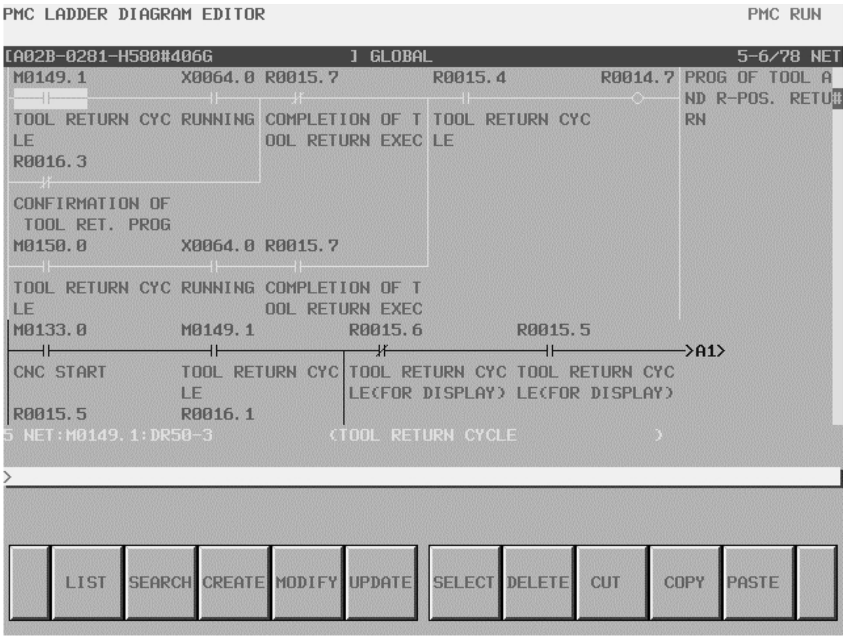


Fig. 5.1 Ladder Diagram Editor screen

NOTE

1 You can edit a ladder regardless of whether the ladder is running/stopping. But, when you are going to execute the edited ladder, it is necessary to update the ladder. By exiting Ladder Editor screen, or pressing [UPDATE] soft key, update the edited ladder. As for the protection of the editing operation, please refer to ”5.3 Example for setting parameters”.

2 If the power is turned off before the edited sequence program is written to flash ROM, the result of the edit is erased.Use the input/output screen to write the sequence program to flash ROM. When K902#0 is set to 1, a confirmation message appears at the termination of editing, asking whether to write the sequence program to flash ROM.

5.2 SCREEN CONFIGURATION

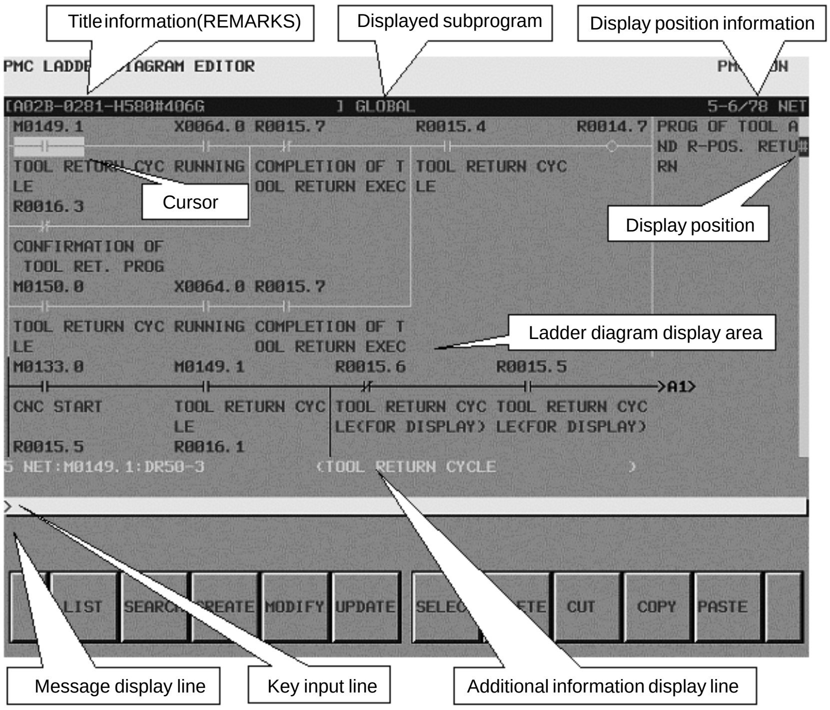


Fig. 5.2 Ladder Diagram Editor screen

(a) Screen configuration

It is the basically same as Ladder diagram Monitor screen, except that no monitor displays of relays and parameters of functional instructions are displayed.

(b) Ladder diagram display

1. Style of a ladder diagram is basically the same as Ladder diagram Monitor screen, except that functional instructions are drawn always in ”COMPACT” format that has no monitor displays.
2. Cursor is shown always. And the net, which will be an object of following editing operations, is emphasized in screen.

5.3 SCREEN OPERATIONS

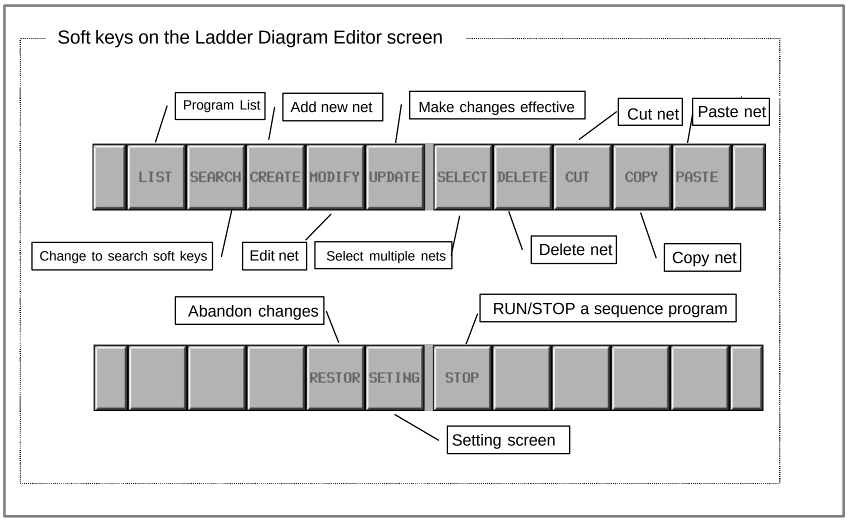


Fig. 5.3 Soft keys on Ladder Diagram Editor screen

(a) Operations using the soft keys

1. [LIST] Call Program List Editor screen
   Calls Program List Editor screen. On Program List Editor screen, you can select a subprogram to be edited on Ladder Diagram Editor screen.
2. [SEARCH] Search and jump menu
   Switches to the search soft keys. To return to the main soft keys, press the Return key [<]. The search soft keys are quite same with ones of Ladder diagram Monitor screen.
3. [MODIFY] Call Net Editor screen
   Calls Net Editor screen to modify structure of the selected net.
4. [CREATE] Create new net
   Create and add new net to cursor position. Pressing this soft key reaches Net Editor screen, so that new net is constructed.
5. [UPDATE] Make changes effective
   Updates running the ladder program to currently edited ladder program, so that the all modifications will take effects, and remains Editor screen. If it succeeds to update running ladder, edited ladder starts to run.

WARNING

You have to pay special attention to modify running ladder program. If you modify the ladder program in wrong way, or update the ladder program with the machine in improper status, it may cause unexpected reaction of the machine. You have to make it sure that ”modifications you make on the ladder program is appropriate”, ”machine is in proper status”, and ”nobody is near the machine”, when you update the ladder program.

6. [SELECT] Select multiple nets
   Selects multiple nets for following operations such as [DELETE], [CUT], [COPY]. Pressing [SELECT] soft key leads to a mode to select one or more nets for following editing operation. Use cursor move keys and search functions to select nets, as you like. Being in mode to select nets is indicated by dented [SELECT] soft key, and the information of selected nets is shown in additional information line at near bottom of screen.
7. [DELETE] Delete net
   Deletes selected nets. The nets deleted by [DELETE] soft key are lost. If you delete wrong nets by [DELETE], you have to abandon the all modifications you have made, and restore the ladder program to the original one before editing operation.
8. [CUT] Cut nets
   Cuts selected nets. The cut nets are preserved in Paste Buffer, and disappear from diagram. The contents of the Paste Buffer before [CUT] operation are lost. [CUT] and [PASTE] soft keys are used to move nets.
9. [COPY] Copy nets
   Copy selected nets into the Paste Buffer. No change on diagram will be made. The contents of the Paste Buffer before [COPY] operation are lost. [COPY] and [PASTE] soft keys are used to copy nets.
10. [PASTE] Paste nets
   Pastes nets at cursor position, which were stored into the Paste Buffer by [CUT] or [COPY] soft key. Pressing [PASTE] soft key while selecting nets using [SELECT] soft key alters the selected nets with the nets in the Paste Buffer. The contents of the Paste Buffer will remain until turning CNC power off.
11. [RESTORE] Abandon changes
   Abandons all changes, and restores the ladder program to the one at entering Ladder Diagram Editor screen, or last updated one using [UPDATE] soft key. This soft key is useful when you make wrong modifications and hard to recover from them.
12. [SETING] Screen settings
   Calls setting screen for Ladder Diagram Editor screen. You can change various settings for Ladder Diagram Editor screen at the screen. Use the Return key [<] to return to Ladder Diagram Editor screen.
13. [RUN]/[STOP] Run and stop the ladder program
   Controls the ladder program execution. [RUN] soft key makes the ladder run and [STOP] soft key makes the ladder stop. Both soft keys will confirm your intention. When you are sure to run or stop the ladder program, press [YES] to take an action.

WARNING

You have to pay special attention to run/stop the ladder program. Running/stopping the ladder program in a wrong timing, or with machine in improper status, may cause unexpected reaction of machine. At stopping the ladder program, a safety mechanism and watch by the ladder program is not operated. You have to make it sure that ”machine is in proper status”, and ”nobody is near the machine” when you run/stop the ladder program.

14.[<] Exit Editor

Updates running the ladder program to edited ladder program, so that the all modifications will take effects, and exits Editor screen. When Ladder Diagram Editor screen is active and the function keys such as <SYS> key will not work, the data under edit is deleted.

WARNING

You have to pay special attention to modify running the ladder program. If you modify the ladder program in wrong way, or update the ladder program with the machine in improper status, it may cause unexpected reaction of the machine. You have to make it sure that ”modifications you make on the ladder program is appropriate”, ”machine is in proper status”, and ”nobody is near the machine”, when you update the ladder program.

(b) Operations using other keys

1. Cursor move keys, Page change keys
   Cursor move keys and Page change keys move cursor on screen. When cursor is placed on some relay or some address parameter of a functional instruction, the information about the address under cursor is displayed at ”Additional Information Line”.
2. ”bit address” + ENTER key
   Changes bit address of relay under cursor.
3. ”number” or ”byte address” + ENTER key
   Changes parameter of functional instructions under cursor. But, there are some parameters that can not change by this operation. If you see a message that means that this parameter can not be changed, use Net Editor screen to change the parameter.

(c) Shortcut operations

1. Same shortcut search operations with Ladder Diagram Monitor screen are available. For their detail, see descriptions about Shortcut operations of Ladder Diagram Monitor screen.
2. Same shortcut operations using [LIST] soft key with Ladder Diagram Monitor screen are available.

5.4 SETTING SCREEN

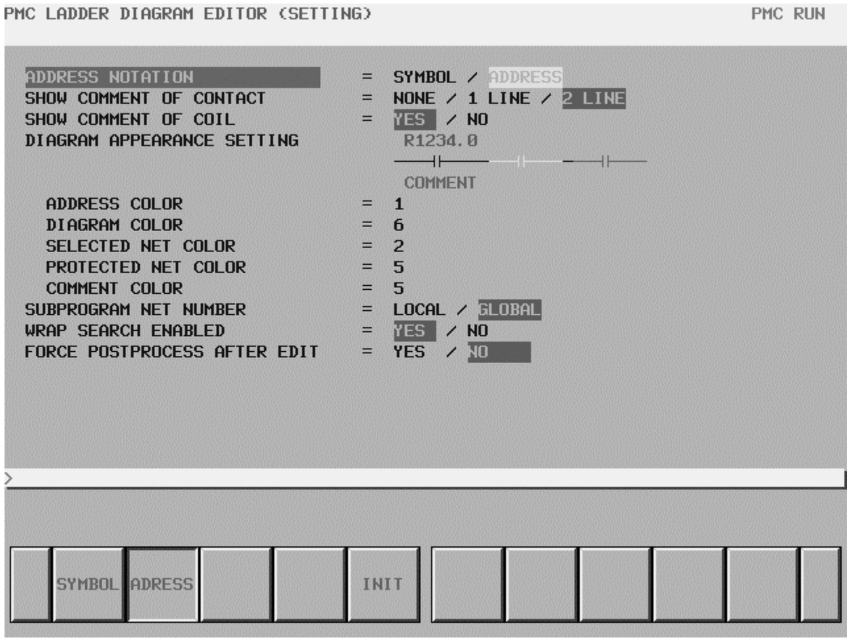


Fig. 5.4 Ladder Diagram Editor Setting screen

(a) Settings

There are the following settings on Ladder Diagram Editor Setting screen:

– ADDRESS NOTATION Default: ADDRESS

Specifies whether to display each bit and byte address in the ladder diagram with the symbol or address.

– SHOW COMMENT OF CONTACT Default: 2 LINE

Changes the display format of the comment displayed under each contact.

– SHOW COMMENT OF COIL Default: YES

Specifies whether to display the comment of each coil.

– DIAGRAM APPEARANCE SETTING

Changes the color for ladder diagram. You can set the colors of a line, relays and etc. that are components of the ladder diagram.

ADDRESS COLOR	Default: Green (1)
DIAGRAM COLOR	Default: Black (6)
SELECTED NET COLOR	Default: Yellow (2)
PROTECTED NET COLOR	Default: Light blue (5)
COMMENT COLOR	Default: Light blue (5)

– SUBPROGRAM NET NUMBER Default: GLOBAL

Specifies whether to display the ”LOCAL” numbers, which are assigned to the nets only in the subprogram, or ”GLOBAL” numbers, which are assigned to the whole ladder program when a subprogram is displayed. The setting also affects the expression of net number information at searching nets by number.

– WRAP SEARCH ENABLED Default: YES

Specifies whether to return to the beginning of the ladder program and continue search operation when the search operation reaches the end of the ladder program.

– FORCE POSTPROCESS AFTER EDIT Default: NO
Decides whether the post–processing after editing ladder program, which makes ladder program ready to run, will be always done, or will be done only when ladder program is actually modified, at exiting Ladder Diagram Editor screen.

(b) Soft keys
The following soft key is available on the Ladder Diagram Editor Setting screen.

– [INIT] Initializing all settings
All settings will be initialized to the default values.

5.5 NET EDITOR SCREEN

In Net Editor screen, you can create new net, and modify existing net.

1. Modifying existing netWhen [MODIFY] soft key is used to reach Net Editor screen, this screen is in ”MODIFY MODE ” to modify existing net.
2. Creating new netWhen [CREATE] soft key is used to reach Net Editor screen, this screen is in ”CREATRE MODE” to create new net from nothing.

Following operations are available at this screen:

– Place new contacts and coils ”bit address”+ [—| |—], [—O—], etc.
– Change type of contacts and coils [—| |—], [—O—], etc.
– Place new functional instructions [FUNC]
– Change type of functional instructions [FUNC]
– Erase contacts, coils, and functional instructions [- - - - -]
– Draw/erase connecting lines [———], [↑———], [———↑]
– Edit data table of functional instructions [TABLE]
– Insert line/column [INSLIN],[INSCLM],[APPCLM]
– Change address of contacts and coils ”bit address” + INPUT key
– Change parameters of functional instructions ”number” or ”byte address”+ INPUT key
– Abandon modifications [RESTOR]

5.5.1 Screen Configuration

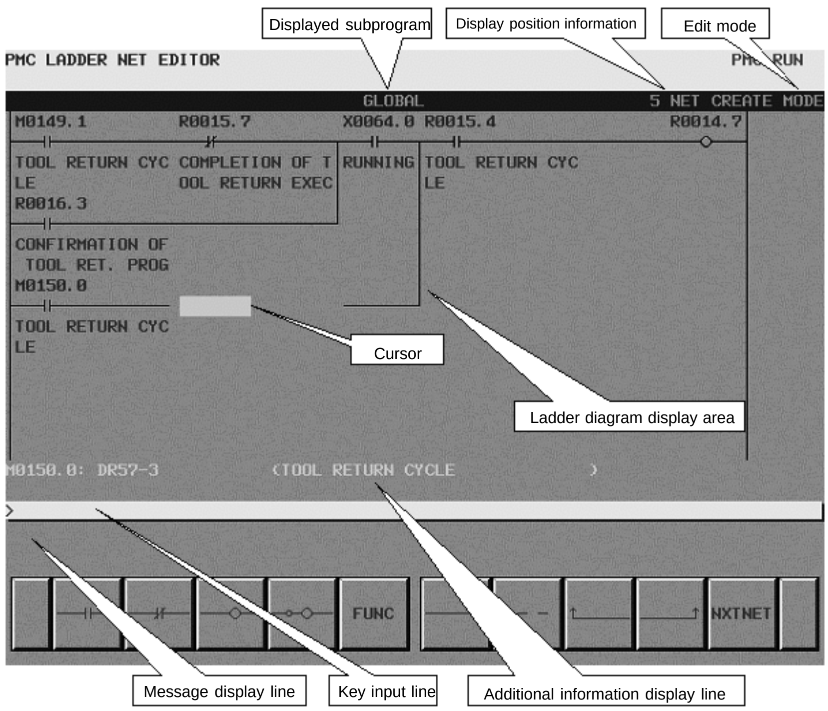


Fig. 5.5.1 Configuration of Net Editor screen

(a) Screen configuration

1. It is basically same with Ladder diagram Editor screen, except that only one net is in this screen, and that position bar at right edge of screen does not appear at this screen.

2. Current edit mode is indicated at right of the top line as ”CREATE MODE” or ”MODIFY MODE”. When [MODIFY] soft key at Ladder Diagram Editor screen is used to reach Net Editor screen, the screen is in Modify mode, and when [CREATE] soft key is used, it is in Create mode.
3. Current net number is displayed at right of the top line. The net number is same with the net number in previous Ladder Diagram Editor screen.
4. Net Editor screen expands image of net horizontally for a wider net according to its width, while Ladder Diagram Monitor/Editor screen folds nets wider than screen width. When net width is expanded over screen width, attempt to move cursor out of screen will scroll net image to the direction. The net of maximum size occupies area of 1024 elements, but actually available area may be little less for internal use according to the internal condition: ”element” means the space that is occupied by single relay.

5.5.2 Screen Operations

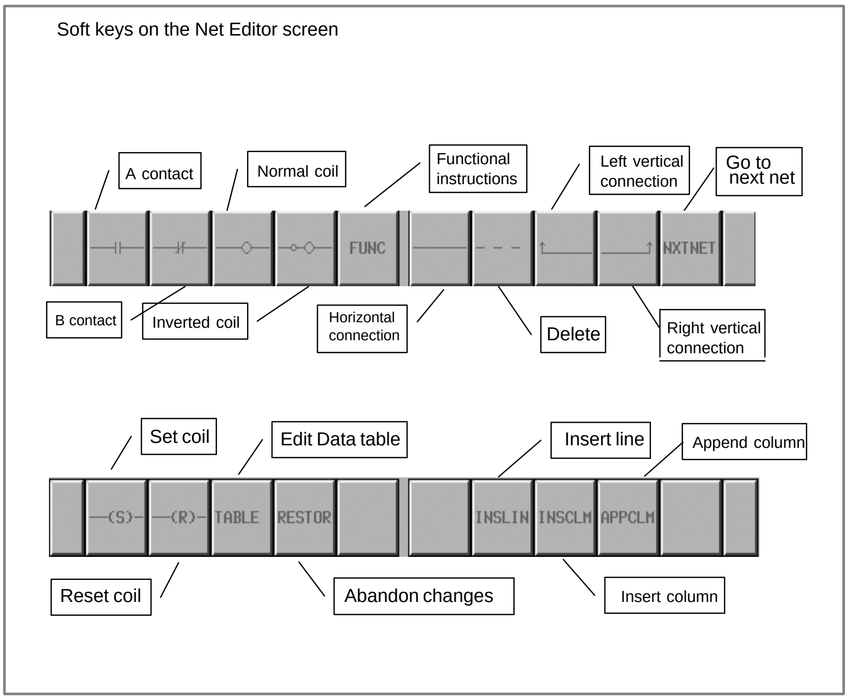
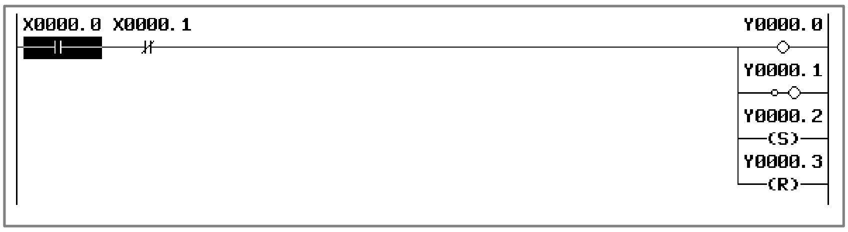


Fig. 5.5.2(a) Soft keys on the Net Editor screen

(a) Screen configuration

1. [—| |—], [—|/|—], [—O—], [—OO—], [—(S)—], [—(R)—], Enter and change relay
   Place relays (contacts and coils), or change type of existing relays. When one of these relay soft keys is pressed at cursor on blank place, new relay of the soft key is placed under the cursor. When the soft key follows a string that means a bit address, the bit address is assigned to the newly placed relay. If no bit address is given, last entered bit address is automatically used for the new relay. If no bit address has been entered yet, the new relay will have no address assigned to it. Contacts can be placed at other than rightmost column, and coils can be placed at rightmost column only. Moving cursor onto an existing relay, pressing a relay soft key of different type changes the type of relay under the cursor. But, changing coil to contact, and changing contact to coil are forbidden. It is basically same with Ladder diagram Editor screen, except that only one net.



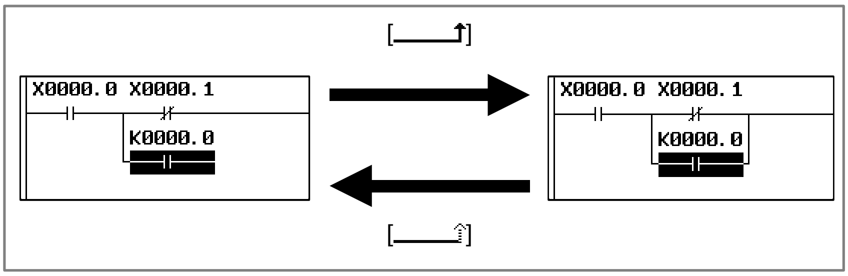
Sample of contacts and coils

2. [FUNC] Enter and change functional instruction
   Places functional instruction, or changes type of existing functional instruction. When [FUNC] soft key is pressed at cursor on blank place, new functional instruction will be placed under the cursor. List of available functional instructions is displayed, then choose type of functional instruction to be entered. When [FUNC] soft key follows a string that means number or name of a functional instruction is entered directly, without the list screen. Moving cursor onto an existing functional instruction, pressing [FUNC] soft key changes the type of functional instruction under the cursor.

3. [———]Draw horizontal connection
   Draws horizontal connection line. Or alters an existing relay to horizontal line.

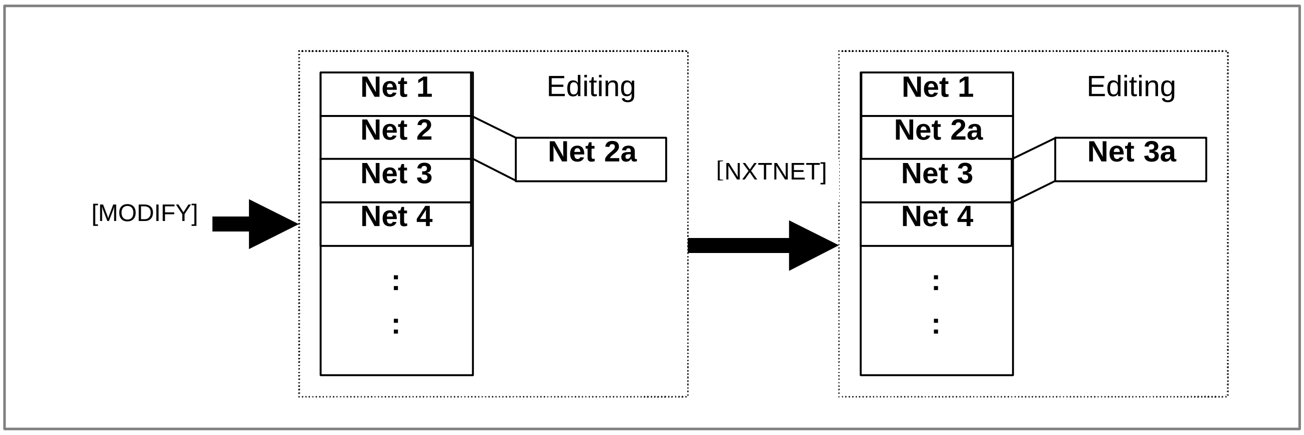
4. [- - - - -]Erase relays and functional instructions
   Erases relays and functional instructions under cursor.

5. [↑———], [———↑]Draw and erase vertical connection
   Draw vertical connection line upward from right or left edge of relay or horizontal line under cursor. Or erase existing vertical lines. If the relay or line under the cursor has no vertical line upward, these soft keys have solid arrows, and indicate that pressing them means drawing lines. On the other hand, if a vertical line already exists under the cursor, arrows in these soft keys become pale ([↑———], [———↑]), and indicate that pressing them means erasing lines.



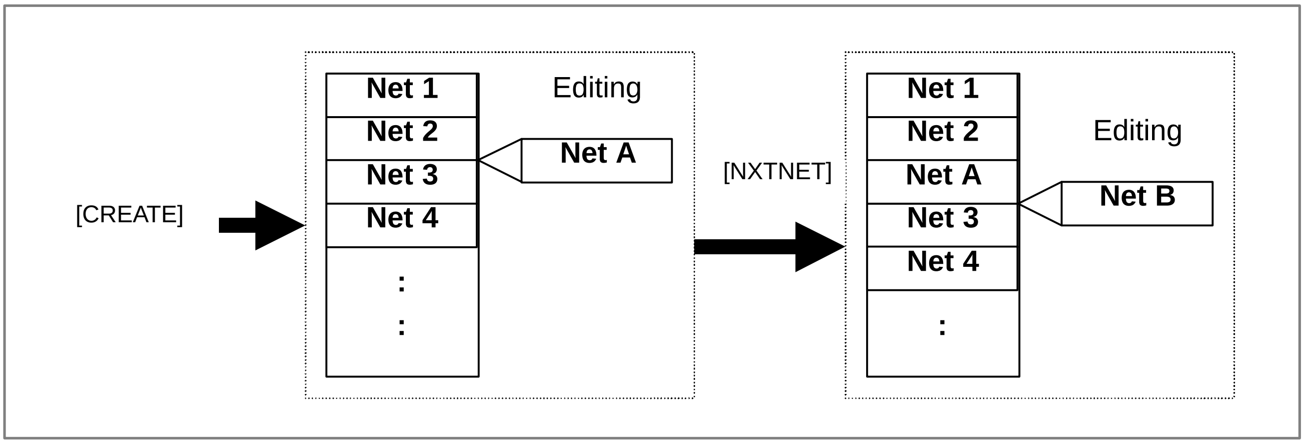
Drawing and erasing vertical connection line

6. [NXTNET] Go to next net
   Finishes editing current net, and goes to next net. If [MODIFY] soft key at Ladder Diagram Editor screen is used to reach Net Editor screen, [NXTNET] will finish modifying current net, and the next net will be an object of further editing operation.



Action of [NXTNET] soft key in Modify mode (modifying an existing net)

If [NXTNET] soft key at Ladder Diagram Editor screen is used to reach Net Editor screen, [NXTNET] will finish creating current net, insert it into the ladder program, and start with blank to create another new net to be inserted next to the current net.



Action of [NXTNET] soft key in Create mode (creating new net)

7. [TABLE] Edit data table
   Reaches Functional Instruction Data Table Editor screen to edit data table of functional instruction under cursor. This soft key appears only when cursor is a functional instruction that has data table with it. For detail of editing operation of data table, see descriptions of ”Functional Instruction Data Table Editor screen”.

8. [RESTOR] Abandon changes
   Abandons all changes, and restores net to the one at starting editing this net. If [CREATE] soft key at Ladder Diagram Editor screen is used to reach Net Editor screen, it will be back to blank net, and if [MODIFY] soft key is used, it will be back to the old net before modifications in this screen.
9. [INSLIN] Insert line
   Inserts one blank line at cursor position. Diagram elements at or below vertical cursor position will be shifted downward by one line. Inserting line at middle of functional instruction box will expand the box vertically to make a space between the input conditions.

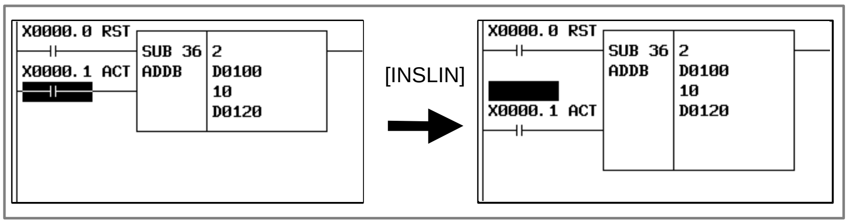
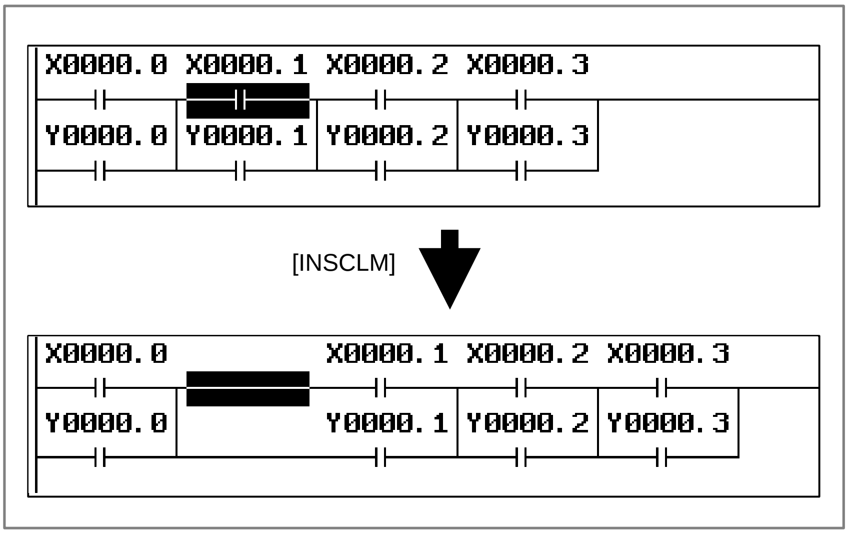


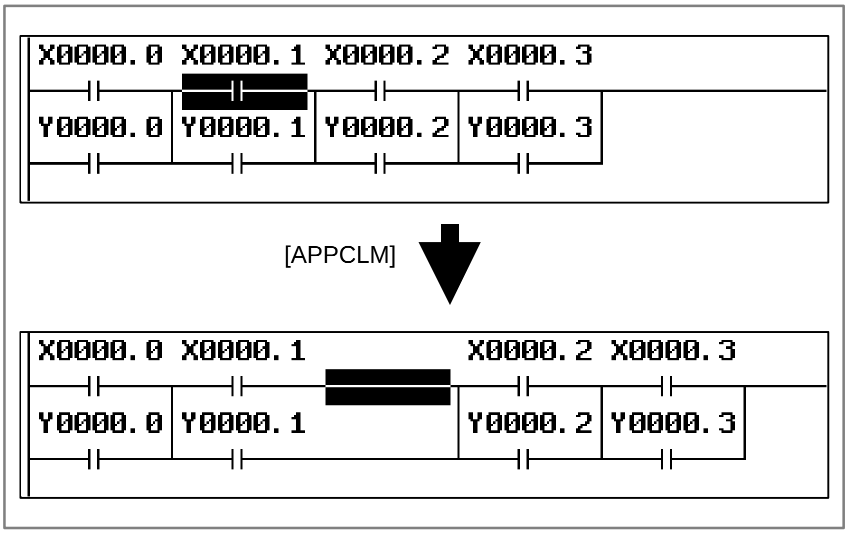
Fig. 5.5.2(b) Action of inserting line

10. [INSCLM] Insert column
   Inserts one blank column at cursor position. Diagram elements at or on right of horizontal cursor position will be shifted to right by one column. And if there is no room to shift the elements, a new column is added and the Diagram area will be expanded to right.



Action of inserting column

11. [APPCLM] Append column
    Inserts one blank column at right of cursor position. Diagram elements on right of horizontal cursor position will be shifted to right by one column. And if necessary, net will be expanded to right.



Action of appending column

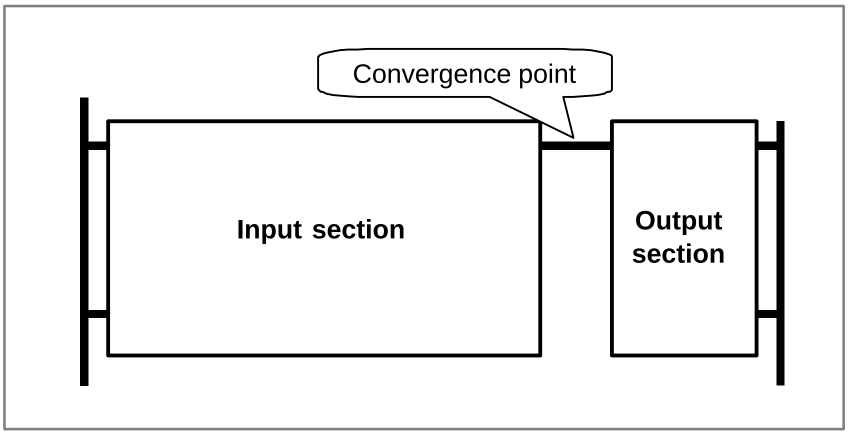
12. [<] Exit editor screen
    Analyzes current editing net, and store it into the ladder program. If some error is found in the net, it still remains Net Editor screen, and an error message will be displayed. According to a kind of error, cursor may indicate where the error is detected.

(b) Operations using other keys

1. Cursor move keys, Page change keys
   Cursor move keys and Page change keys move cursor on screen. Net Editor screen expands image of net horizontally for a wider net according to its width, while Ladder Diagram Monitor/Editor screen folds nets wider than screen width. When net width is expanded over screen width, attempt to move cursor out of screen will scroll net image to the direction. The net of maximum size occupies area of 1024 elements, but actually available area may be little less for internal use according to the internal condition: ”element” means the space that is occupied by single relay.
2. ”bit address” + INPUT key
   Changes bit address of relay under cursor.
3. ”number” / ”byte address” + INPUT key
   Changes parameter of functional instructions under cursor.

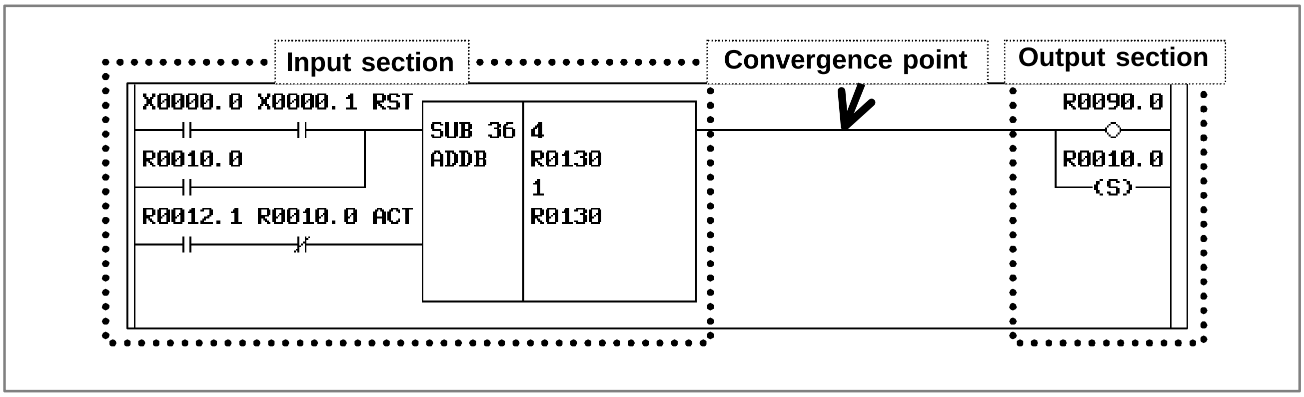
5.6 STRUCTURE OF VALID NET

Valid net must have following structure:



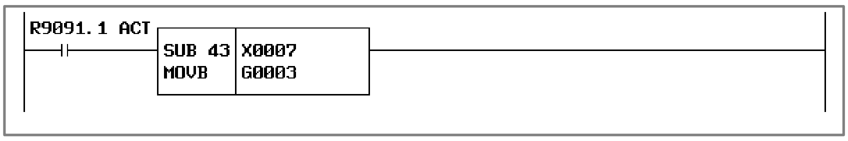
Structure of valid net

”Input section” consists of contacts and functional instruction, and the result of operations of input section is led to ”Convergence point”. After the convergence point, there is ”Output section” that consists of coils only. The ”Convergence point” is the nearest point to right power line, where all connections join with each other to gather into single connection.



Sample of valid net

Input section consists at least one relay or functional instruction, however, output section may contain nothing.



Example of net with no output section

Valid net is also restricted in following rules:

– Only one function instruction is available for a net.
– Functional instruction can be placed only at last (rightmost) of input section.
– Only coils can be contained in output section.

5.7 FUNCTIONAL INSTRUCTION LIST SCREEN

Pressing [FUNC] soft key at Net Editor screen reaches Functional Instruction List screen at which you choose a functional instruction to be entered from list of all available functional instructions.

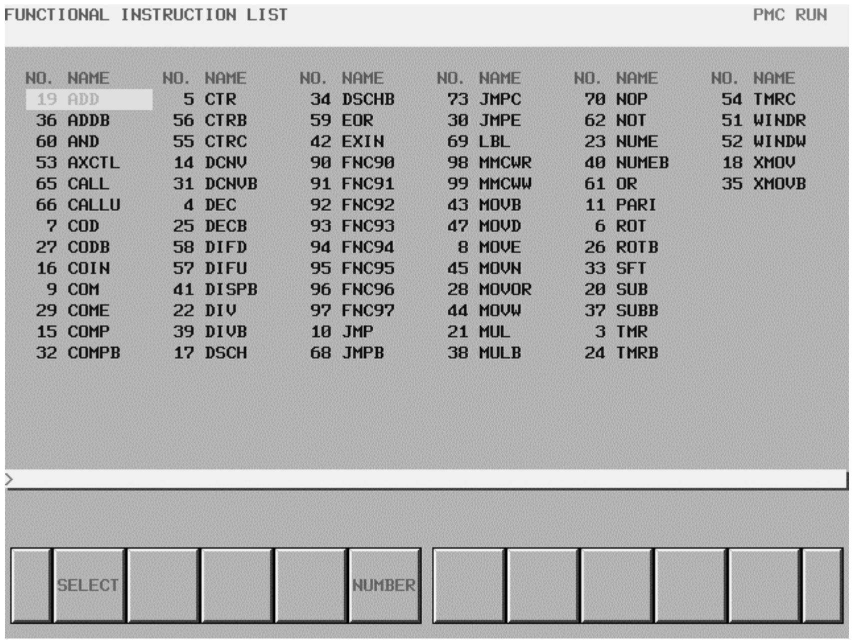


Fig. 5.7 Functional Instruction List screen

Operations at this screen are below:

(a) Operations using the soft keys

1. [SELECT] Select a functional instruction
   Selects a functional instruction. The function instruction under cursor at that time is chosen, and entered into the editing net.
2. [NUMBER], [NAME] Rearrange functional instructions list
   Rearrange functional instructions list in two ways. [NUMBER] soft key arranges the list in numerical order with their identifying numbers, on the other hand, [NAME] soft key arrange it in alphabetical order with their names. At beginning, the list is arranged in alphabetical order.
3. [<] Quit selecting
   Quits selecting functional instruction, and return to Net Editor screen.

5.8 FUNCTIONAL INSTRUCTION DATA TABLE EDITOR SCREEN

At Functional Instruction Data Table Editor screen, you can edit the contents of data table that belongs to some functional instructions. To reach this screen, at Net Editor screen, press [TABLE] soft key that is displayed when the cursor is on the following functional instructions which have a data table.

– Functional Instruction COD (SUB7)
– Functional Instruction CODB (SUB27)
  (Function instruction DISP(SUB49) is not be able to use)

Following edit operations are available at this screen.

– Change the data table value. ”number” + ENTER key
– Change the data length. [BYTE], [WORD], [D.WORD]
  (These soft keys can be operated only at Functional Instruction Data Table Editor screen of Functional Instruction CODB.)
– Change the number of data [COUNT]
– Initialize all of data [INIT]

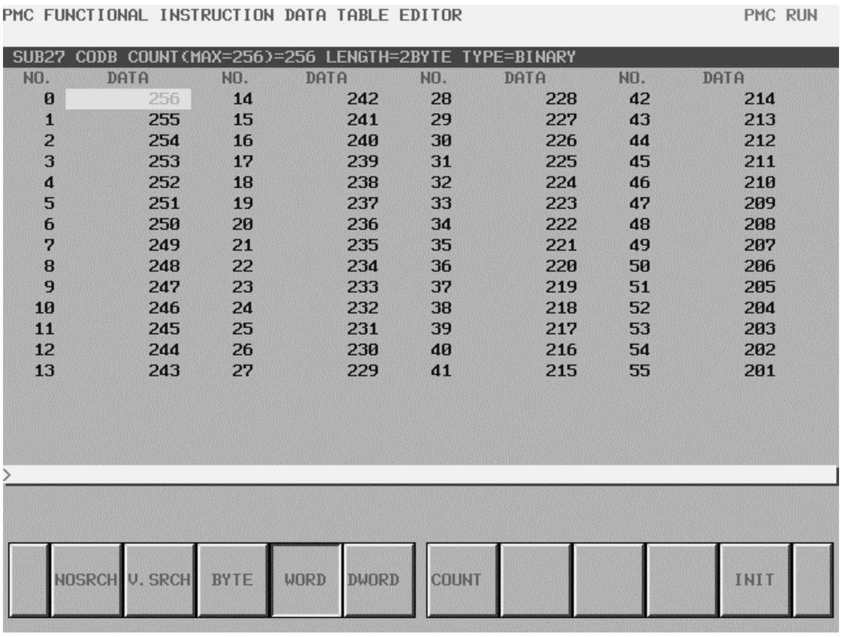


Fig. 5.8 Functional Instruction Data Table Editor screen

5.9 PROGRAM LIST EDITOR SCREEN

At Program List Editor screen you can create new program and delete a program in addition to the function of Program List Viewer screen. To reach this screen, press [LIST] soft key at Ladder Diagram Editor screen. Following operations are available at Program List Editor screen. For more detail of these operations, refer to the descriptions of each key to operate.

– Create new program [NEW]
– Delete a program [DELETE]

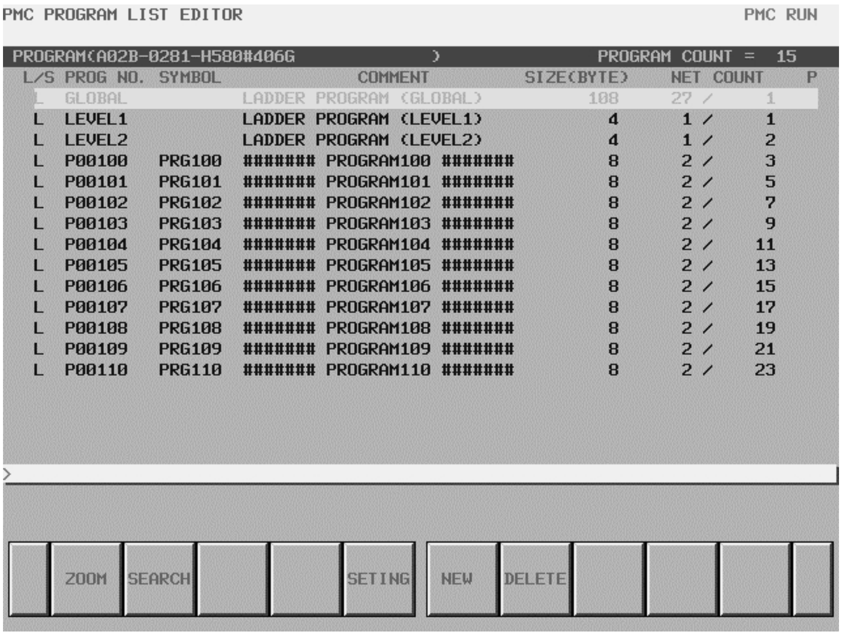


Fig. 5.9(a) Program List Editor screen (Detail)

You can select Detail viewer format or Brief viewer format on Program List Editor screen. The default viewer format is Detail viewer format.

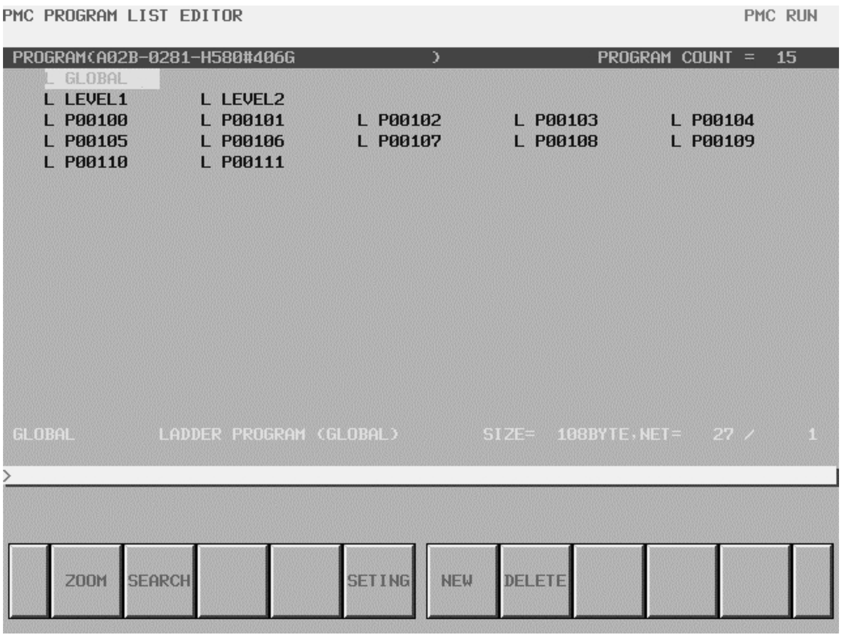


Fig. 5.9(b) Program List Editor screen (Brief)

5.9.1 Screen Operations

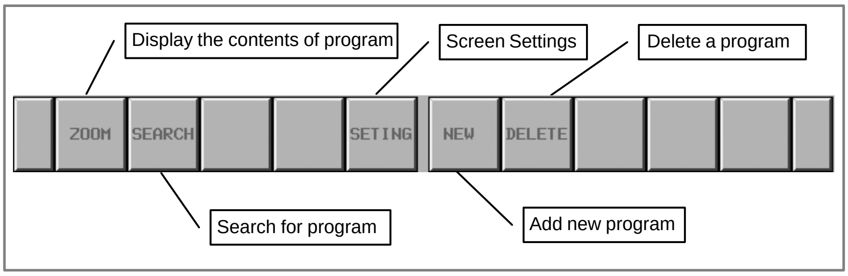


Fig. 5.9.1 Soft keys of Program List Editor screen

(a) Operations using the soft keys

1. [ZOOM] Display the contents of program
   Goes to Ladder Diagram Editor screen.
2. [SEARCH] Search for program
   Searches for a program. Pressing [SEARCH] soft key after entering a program name or symbol name searches for program corresponding to the input character string and moves the cursor to the program.
3. [SETING] Screen settings
   Calling the screen for setting of Program List Editor screen. You can change various settings for Program List Editor screen. To return to Program List Editor screen, press the return key [<].
4. [NEW] Create new program
   If you entered program name or symbol and press [NEW] soft key, the program will be checked its existence. If such program is not found, new program will be created. The created program is inserted automatically into the program list and the cursor points it. The following ladder nets are created automatically according to the type of created program by this operation.
   LEVEL1 : Functional instruction END1
   LEVEL2 : Functional instruction END2
   LEVEL3 : Functional instruction END3
   Subprogram : Functional instruction SP, SPE
   If the status of protection of the program is enabled to edit, this operation is available.
5. [DELETE] Delete a program
   Deletes a program. If you entered no strings and press [DELETE] soft key, the program under the cursor is deleted. If you entered program name or symbol and press [DELETE] soft key, the program will be checked its existence, and will be deleted if such program is found.
   But, GLOBAL, LEVEL1 and LEVEL2 should always exist on program list. If you delete these programs, the contents of program are abandoned. But these programs do not disappear on program list. If the status of protection of the program is enabled to edit, this operation is available.

5.9.2 Setting Screen

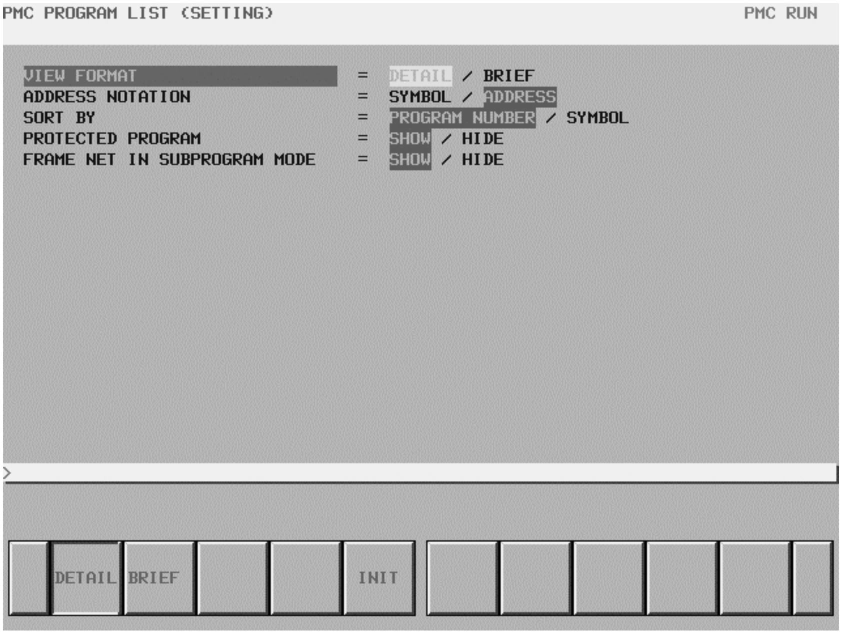


Fig. 5.9.2 Program List Editor (Setting) screen

(a) Settings

There are the following settings on Program List Editor (Setting) screen:

– VIEW FORMAT Default : DETAIL
Specifies whether to display Program List Editor screen in ”DETAIL” or ”BRIEF” mode.

– ADDRESS NOTATION Default : ADDRESS
Specifies whether to display each subprogram on Program List Editor screen with the address or symbol.

– SORT BY Default : PROGRAM NUMBER
Specifies whether to display each subprograms on Program List Editor screen in order of program numbers or symbols. When ADDRESS NOTATION is SYMBOL, programs without symbols are displayed in order of program number after programs with the symbols. GLOBAL, LEVEL1, LEVEL2, LEVEL3 are out of target of sort.

– PROTECTED PROGRAM Default : SHOW
Specifies whether to display protected programs. The protected program for this setting means programs which cannot be edited on Program List Editor screen.

– FRAME NET IN SUBPROGRAM MODE Default : SHOW
Frame net means functional instruction END1, 2 and 3 on LEVEL1, 2, 3, and functional instruction SP and SPE on subprogram. This setting specifies whether to display these frame nets when the contents of a program are displayed by pressing [ZOOM] soft key on Program List Editor screen.

5.10 ALARM MESSAGE AND COUNTERMEASURE

Messages that may be displayed during PMC program editing

Alarm number	Faulty location/corrective action	Contents
OVERLAPPED COM	If COME is missing, add it in proper position. If the COM is unnecessary, remove it.	There is no COME that corresponds to this COM.
END IN COM END1 IN COM END2 IN COM	If COME is missing, add it in proper position. If COM is unnecessary, remove it.	END,END1,END2, or END3 is found between COM and COME.
JMPE IN COM	JMPE and corresponding JMP must have same COM/COME status. Review JMP range and COM range, to adjust not to overlap with each other: it is possible that one range includes the other completely.	JMPE is found between COM and COME, and JMP and corresponding JMPE have different COM/COME status.
SP/SPE IN COM	If COME is missing, add it in proper position. If the COM is unnecessary, remove it.	SP or SPE is found between COM and COME.
COME WITHOUT COM	If COM is missing, add it in proper position. If the COME is unnecessary, remove it.	There is no COM that corresponds to this COME.
DUPLICATE CTR NUMBER (WARNING)	If some of them are unnecessary, remove them. If all of them are necessary, assign other number to parameter of them to make them unique. (If two or more instructions with same parameter number will never be active simultaneously at one time, the Ladder program has a possibility to work correctly, however, it is recommended from safety and maintenance points of view, that all these instructions should have different parameter number with each other.)	Plural CTRs have the same number as their parameter. (This is warning.)
ILLEGAL CTR NUMBER	If unnecessary, remove it. Assign correct number not to exceed the maximum number defined by each PMC model.	CTR has parameter number that is out of range.

Alarm number	Faulty location/corrective action	Contents
DUPLICATE DIFU/DIFD NUMBER (WARNING)	If some of them are unnecessary, remove them. If all of them are necessary, assign other number to parameter of them to make them unique. (If two or more instructions with same parameter number will never be active simultaneously at one time, the Ladder program has a possibility to work correctly, however, it is recommended from safety and maintenance points of view, that all these instructions should have different parameter number with each other.)	Plural DIFUs or DIFDs have the same number as their parameter. (This is warning.)
ILLEGAL DIFU/DIFD NUMBER	If unnecessary, remove it. Assign correct number not to exceed the maximum number defined by each PMC model.	DIFU or DIFD has parameter number that is out of range.
NO END NO END1 NO END2 NO END3	Add END, END1, END2 or END3 in proper position.	END, END1, END2 or END3 is not found.
DUPLICATE END1 DUPLICATE END2 DUPLICATE END3	Remove extra END1, END2 or END3.	Multiple END1, END2 or END3 are found.
GARBAGE AFTER END GARBAGE AFTER END2 GARBAGE AFTER END3	Remove unnecessary nets, and move necessary nets to proper position so that they will be executed.	There are some nets after END, END2 or END3, which will not be executed.
OVERLAPPED JMP	If JMPE is missing, add it in proper position. If the JMP is unnecessary, remove it.	There is no JMPE that corresponds to this JMP.
JMP/JMPE TO BAD COM LEVEL	JMP and corresponding JMPE must have same COM/COME status. Review JMP range and COM range, to adjust not to overlap with each other: it is possible that one range includes the other completely.	JMP and corresponding JMPE have different COM/COME status.
COME IN JMP	COME and corresponding COM must have same JMP/JMPE status. Review COM range and JMP range, to adjust not to overlap with each other: it is possible that one range includes the other completely.	COME is found between JMP and JMPE, and COM and corresponding COME have different JMP/JMPE status.
END IN JMP END1 IN JMP END2 IN JMP END3 IN JMP	If JMPE is missing, add it in proper position. If JMP is unnecessary, remove it.	END,END1,END2, or END3 is found between JMP and JMPE.
SP/SPE IN JMP	If JMPE is missing, add it in proper position. If the JMP is unnecessary, remove it.	SP or SPE is found between JMP and JMPE.

Alarm number	Faulty location/corrective action	Contents
JMPB OVER COM BORDER	JMPB and its destination must have same COM/COME status. Review range of JMPB and COM range, to adjust not to overlap with each other: it is possible that one range includes the other completely.	JMPB and its destination differ in COM/COME status.
JMPB OVER LEVEL	JMPB can only jump to the same program level, or within a subprogram. If the JMPB is unnecessary, remove it. If LBL for the JMPB is missing, add it in proper position. If it should be JMPC, correct it.	JMPB jumps to different program level.
LBL FOR JMPB NOT FOUND	If JMPB is unnecessary, remove it. If LBL is missing, add it in proper position.	Can not find proper LBL for JMPB.
JMPC IN BAD LEVEL	JMPC is used to jump from a subprogram to level 2. If the JMPC is unnecessary, remove it. If it should be JMPB or JMP, correct it.	JMPC is used in other than subprogram.
LBL FOR JMPC NOT FOUND	If JMPC is unnecessary, remove it. If LBL is missing, add it in proper position: JMPC jumps into level 2. If it should be JMPB or JMP, correct it.	Can not find proper LBL for JMPC.
LBL FOR JMPC IN BAD LEVEL	JMPC is used to jump from a subprogram to level 2. If the JMPC is unnecessary, remove it. If another LBL of same L–address that the JMPC is intended to jump exists in the subprogram, assign different L–address to these two LBLs. If it should be JMPB or JMP, correct it.	Destination of JMPC is not level 2.
JMPC INTO COM	LBL for JMPC must be located out of any COM and COME pair. If the JMPC is unnecessary, remove it. If the LBL is located wrong, move it to correct position. If the L–address of JMPC is wrong, correct it.	JMPC jumps to LBL between COM and COME.
JMPE WITHOUT JMP	If JMP is missing, add it in proper position. If the JMPE is unnecessary, remove it.	There is no JMP that corresponds to this JMPE.
TOO MANY LBL	Remove unnecessary LBLs. If this error still occurs, adjust the construction of program to use less LBLs.	There are too many LBLs.
DUPLICATE LBL	If some of these LBLs are unnecessary, remove them. If all of these LBLs is necessary, assign other L–addresses to them to make all LBLs unique.	Same L–address is used in plural LBLs.
OVERLAPPED SP	If SP is missing, add it in proper position. If the SPE is unnecessary, remove it.	There is no SP that corresponds to this SPE.

Alarm number	Faulty location/corrective action	Contents
SPE WITHOUT SP	If SP is missing, add it in proper position. If the SPE is unnecessary, remove it.	There is no SP that corresponds to this SPE.
END IN SP	If SPE is missing, add it in proper position. If END is in wrong place, move it to proper position.	END is found between SP and SPE.
DUPLICATE P ADDRESS	If some of these SPs are unnecessary, remove them. If all of these SPs is necessary, assign other P–addresses to them to make all SPs unique.	Same P–address is used in plural SPs.
DUPLICATE TMRB NUMBER (WARNING)	If some of them are unnecessary, remove them. If all of them are necessary, assign other number to parameter of them to make them unique. (If two or more instructions with same parameter number will never be active simultaneously at one time, the Ladder program has a possibility to work correctly, however, it is recommended from safety and maintenance points of view, that all these instructions should have different parameter number with each other.)	Plural TMRBs have the same number as their parameter. (This is warning.)
ILLEGAL TMRB NUMBER	If unnecessary, remove it. Assign correct number not to exceed the maximum number defined by each PMC model	TMRB has parameter number that is out of range.
DUPLICATE TMR NUMBER (WARNING)	If some of them are unnecessary, remove them. If all of them are necessary, assign other number to parameter of them to make them unique. (If two or more instructions with same parameter number will never be active simultaneously at one time, the Ladder program has a possibility to work correctly, however, it is recommended from safety and maintenance points of view, that all these instructions should have different parameter number with each other.)	Plural TMRs have the same number as their parameter. (This is warning.)
ILLEGAL TMR NUMBER	If unnecessary, remove it. Assign correct number not to exceed the maximum number defined by each PMC model.	TMR has parameter number that is out of range.
NO SUCH SUBPROGRAM	If it calls wrong subprogram, correct it. If the subprogram is missing, create it.	Subprogram that is called by CALL/CALLU is not found.
UNAVAILABLE INSTRUCTION	Confirm that this ladder program is correct one. If this program is correct one, all these unsupported instructions have to be removed.	Unsupported instruction for this PMC model is found.
SP IN BAD LEVEL	SP can be used at top of a subprogram. Correct it so that no SP exists in other place.	SP is found in wrong place.

Alarm number	Faulty location/corrective action	Contents
LADDER PROGRAM IS BROKEN	This ladder program must be all cleared once, and remake ladder program.	Ladder program may be broken by some reason.
NO WRITE COIL	Add proper write coil.	Write coil is necessary, but is not found.
CALL/CALLU IN BAD LEVEL	CALL/CALLU must be used in Level 2 or in subprograms. Do not use any other places.	CALL/CALLU is used in wrong place.
SP IN LEVEL3	If END3 is located wrong, move it to correct position. If the SP is unnecessary, remove it.	SP is found in level 3.

Messages that may be displayed during net editing on PMC program editor screen

Alarm number	Faulty location/corrective action	Contents
TOO MANY FUNCTIONAL INSTRUCTIONS IN ONE NET	Only one functional instruction is allowed to constitute a net. If necessary, divide the net into plural nets.	Too many functional instructions are in one net.
TOO LARGE NET	Divide the net into plural nets so that step number in a net may become small.	Net is too large. When a net is converted into the object, the net exceeds 256 steps.
NO INPUT FOR OPERATION	Coil without input, or coil connected to output of functional instruction that has no output, causes this error. If coil is not necessary, remove it. If necessary, connect it to meaningful input.	No signal is provided for logical operation.
OPERATION AFTER FUNCTION IS FORBIDDEN	Output of functional instruction can not be connected to a contact, nor to conjunction with other signal that will be implemented by logical–or operation.	No logical operation with functional instruction output is permitted, except write coils.
WRITE COIL IS EXPECTED	Add proper write coil to the net.	Write coil is expected, but not found.
BAD COIL LOCATION	Coil can be located only at rightmost column. Any coil located at other place must be erased once, and place necessary coils in correct place.	Coil is located in bad position.
SHORT CIRCUIT	Find contact with terminals connected by short circuit, and correct connections.	Some contacts are connected with short circuit.
FUNCTION AFTER DIVERGENCE IS FORBIDDEN	Functional instruction can not be used in output section of net. If necessary, divide the net into plural nets.	Functional instruction is used in output section of net.
ALL COIL MUST HAVE SAME INPUT	Left terminals of all coils in a net must be connected to same input point.	When a net contains more than one coil, the coils should not have any contact beside them affects only of the coils.

Alarm number	Faulty location/corrective action	Contents
BAD CONDITION INPUT	Check the connection of all condition inputs of the functional instruction. Especially for functional instruction that has more than one condition input, check if connections to condition inputs interfere with each other.	Some condition input of functional instruction is not connected correctly.
NO CONNECTION	Find gap that is expected to be connected, and correct the connection.	There is signal connected to nowhere.
NET IS TOO COMPLICATED	Examine every connection, and find unnecessarily bending connection, or coils that are connected to different point.	Net is too complicated to analyze.
PARAMETER IS NOT SUPPLIED	Enter all of the relay addresses, and parameters of functional instructions.	Relay with blank address, or blank parameter of functional instruction, is found.

6 SIGNAL TRACE FUNCTION (PMC–SB7)

he signal trace function for PMC–SB7 has both the signal wave form display function and the signal trace function for PMC–SA5/SB5/SB6 and has improved specification and operation.

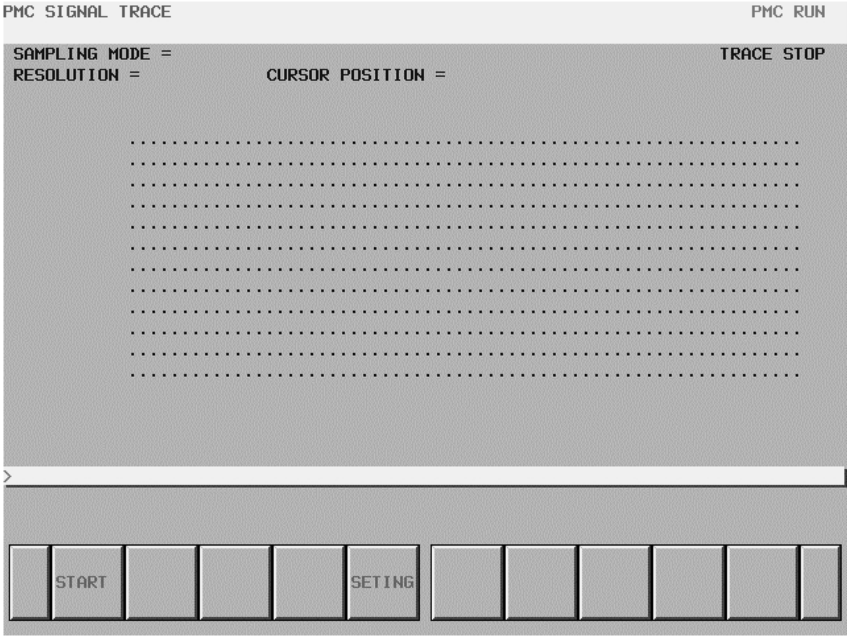
	PMC–SA5/SB5/SB6	**PMC–SB7**
Used condition	The signal wave form display function is provided in the ladder editing card.	Basic function as PMC control software
Trace points	16 points	32 points
Sampling resolution	8ms (Fixed)	8ms to 1sec (Variable)
Maximum sampling time	10sec. (Fixed)	Sampling time is expanded according to number of trace points and resolution. Ex.) 98sec. when 16 points and 8ms resolution (Variable) 100 minutes when 32 points and 1 sec. resolution.
Display of trace result	Sampling graph does not be displayed dynamically in signal wave form display function.	Sampling graph is displayed dynamically during the execution of trace.
	On/off status of each bit is displayed in signal trace function	

On the trace screen, you can trace specified signals. The result of the trace is displayed as the time chart of signals. There are two tracing modes.

- ”Time Cycle” mode: Samples the state of the signals at every specified cycle time.
  (Same specification as the signal wave form display for PMC–SA5/SB5/SB6)
- ”Signal Transition” mode: Samples the status of the signals when the signals that are watched at every specified time are changed.
  (Same specification as the trace function for PMC–SA5/SB5/SB6)

6.1 SIGNAL TRACE SCREEN (INITIAL SCREEN)

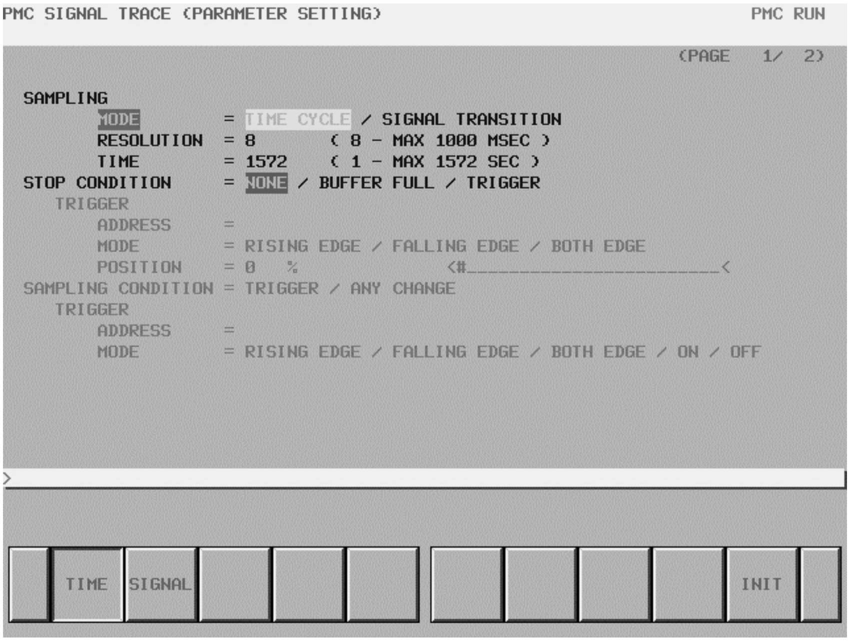
Pushing the [TRACE] soft key on PMC Diagnosis screen displays the Signal Trace screen.



Signal Trace screen (Initial screen)

6.2 SETTING OF TRACE PARAMETER

Pushing the [SETING] soft key on the Signal Trace screen displays the ”Parameter Setting” screen. ”Parameter Setting” has two screens. The page key changes these screens.



Setting screen of Trace Parameter (Page 1)

a) SAMPLING/ MODE
Determines the sampling mode.
– TIME CYCLE: Samples at every specified cycle time.
– SIGNAL TRANSITION: Samples when the signal changes.

b) SAMPLING/ RESOLUTION
The resolution of sampling is inputted. The default value is the minimum resolution (8msec). The range of the value is from 8msec to 1000msec. Inputted value is rounded down to the multiple of 8msec.

c) SAMPLING/ TIME
This parameter is displayed when ”TIME CYCLE” is set on ”SAMPLING MODE”. The execution time of trace is inputted. The value of ”SAMPLING RESOLUTION” or the number of specified signal address changes the range of the value that is able to input. The range is displayed on the right side.

d) SAMPLING/ FRAME
This parameter is displayed when ”SIGNAL TRANSITION” is set on SAMPLING MODE”. The number of sampling is inputted. The value of ”SAMPLING RESOLUTION” or the number of specified signal addresses changes the range of the value that is able to input. The range is displayed on the right side.

e) STOP CONDITION
Determines the condition to stop the trace.
– NONE: Does not stop the tracing automatically.
– BUFFER FULL: Stops the tracing when the buffer becomes full.
– TRIGGER: Stops the tracing by trigger.

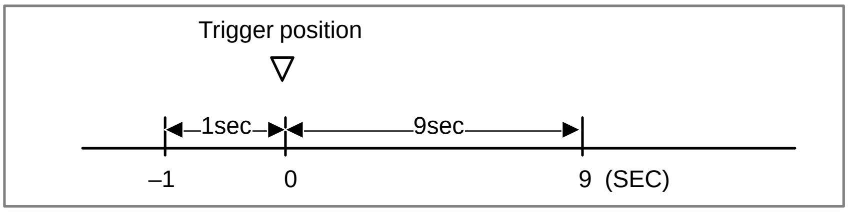
f) STOP CONDITION/ TRIGGER/ ADDRESS
When ”TRIGGER” is set on ”STOP CONDITION”, this parameter is enabled. Input signal address or symbol name as stop trigger.

g) STOP CONDITION/ TRIGGER/ MODE
When ”TRIGGER” is set on ”STOP CONDITION”, this parameter is enabled. Determine the trigger mode when the trace is stopped.

– RISING EDGE: Stops the tracing automatically by rising up of the trigger signal.
– FALLING EDGE: Stops the tracing automatically by falling down of the trigger signal.
– BOTH EDGE: Stops the tracing automatically by rising up or falling down of the trigger signal.

h) STOP CONDITION/ TRIGGER/ POSITION
When ”TRIGGER” is set on ”STOP CONDITION”, this parameter is enabled. Input the ratio of the sampling time or number which specifies the position where specified trigger condition is on. If you would like to examine the transitions of the signal before the trigger condition, you should set a big value in this parameter. If you would like to examine the transitions of the signal after the trigger condition, you should set a small value in this parameter.

Example: The case that sampling time is 10 second and trigger position is set as ”10%”.



i) SAMPLING CONDITION
When ”SIGNAL TRANSITION” is set on ”TRACE MODE”, this parameter is enabled. Determine the sampling condition.

– TRIGGER: Samples the status of specified signals when the specified sampling condition is on.
– ANY CHANGE: Samples the status of specified signals when the signals change.

j) SAMPLING CONDITION/ TRIGGER/ ADDRESS
When ”SIGNAL TRANSITION” is set on ”TRACE MODE”, and ”TRIGGER” is set on ”SAMPLING CONDITION”, this parameter is enabled. Input signal address or symbol name as sampling trigger.

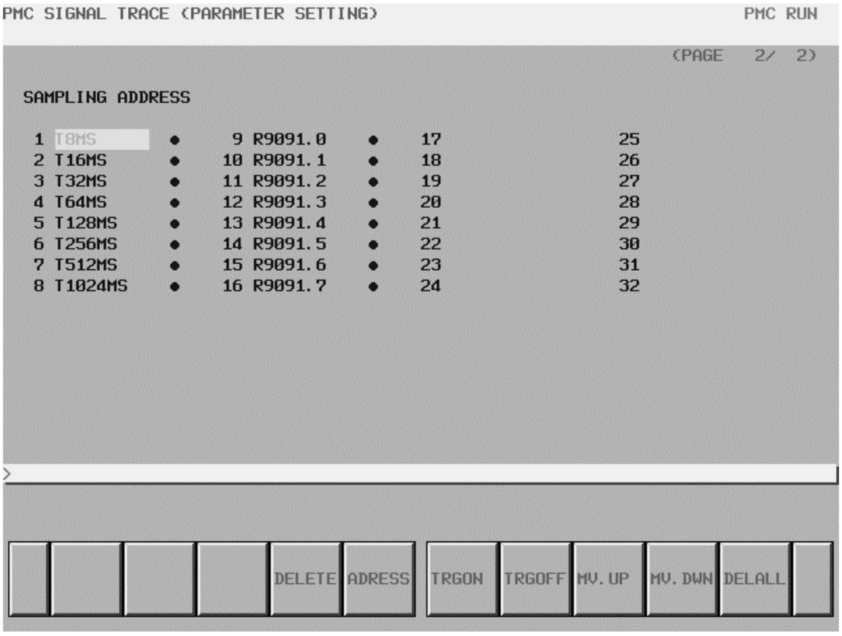
k) SAMPLING CONDITION/ TRIGGER/ MODE
When ”SIGNAL TRANSITION” is set on ”TRACE MODE”, and ”TRIGGER” is set on ”SAMPLING CONDITION”, this parameter is enabled. Input trigger mode that determines the condition of specified trigger.

– RISING EDGE: Samples the status of specified signals by rising up of the trigger signal.
– FALLING EDGE: Samples the status of specified signals by falling down of the trigger signal.
– BOTH EDGE: Samples the status of specified signals by rising up or falling down of the trigger signal.
– ON: Samples the status of specified signals during the trigger signal is on.
– OFF: Samples the status of specified signals during the trigger signal is off.

6.3 SETTING OF SAMPLING ADDRESS

a) Setting addresses

In page 2 of Parameter Setting screen, you can set the addresses or symbols that should be sampled.



Setting screen of Trace Parameter (Page 2)

In case of inputting discrete bit addresses, any bit address can be inputted. Moreover, when you input byte address, all bits of the address (bit0–bit7) are set automatically. Maximum 32 points of signal address can be inputted.

> **NOTE**
> Increasing the number of the signal address changes the capacity of
> ”SAMPLING TIME” or ”SAMPLING FRAME” in page 1. If the capacity
> is changed, the following warning message is displayed. (The ”n” on the message means the maximum value that is able to input.)
> a) In case of ”TIME CYCLE” mode
> ”SAMPLING TIME IS REDUCED TO n SEC.”
> b) In case of ”SIGNAL TRANSITION” mode

b) Soft keys

Soft keys on the Setting screen of sampling address are as follows

– DELETE: Clears the value of the edit box on the cursor.

– SYMBOL: Changes the address display to the symbol display. However, display of the address that is not defined the symbol does not change. This soft key also changes to ”ADDRESS”. The following soft keys are displayed.

– MV.UP: Exchanges the signal indicated the cursor for the signal above one line.

– MV.DWN: Exchanges the signal indicated the cursor for the signal below one line.

– DELALL: Clears all of the value of the edit box.

c) Trigger setting

When ”SIGNAL TRANSITION” is set on ”TRACE MODE” and ”ANY CHANGE” is set on ”SAMPLING CONDITION”, it can be set whether to use the setting address as the signals that should trigger the sampling in the setting signals. As for the signal address where the trigger was set, ”●” is displayed right. Soft keys on the Trigger setting screen are as follows

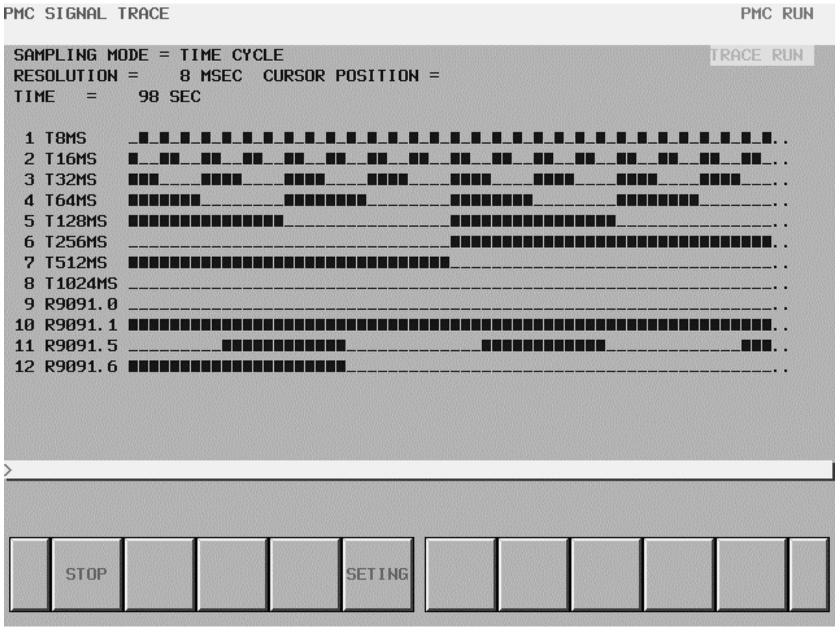
– TRGON: Sets the Trigger on.

– TRGOFF: Sets the Trigger off.

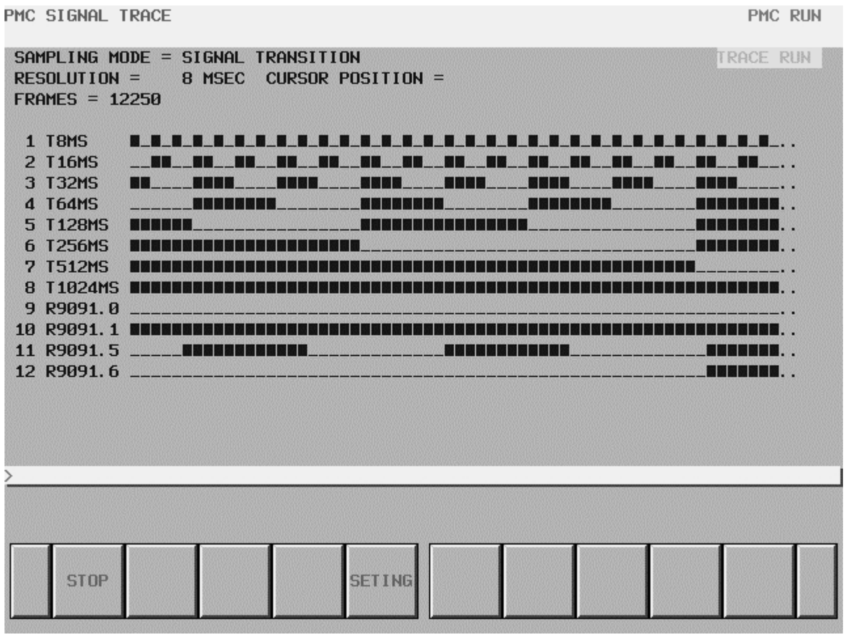
The default setting is trigger on for all signals.

6.4 EXECUTION OF TRACE

On trace screen, pushing [START] soft key starts the execution of trace after you set the trace parameter correctly. The followings are the screen examples of the trace execution by ”TIME CYCLE” mode and ”SIGNAL TRANSITION” mode



Execution of Trace screen (TIME CYCLE mode)

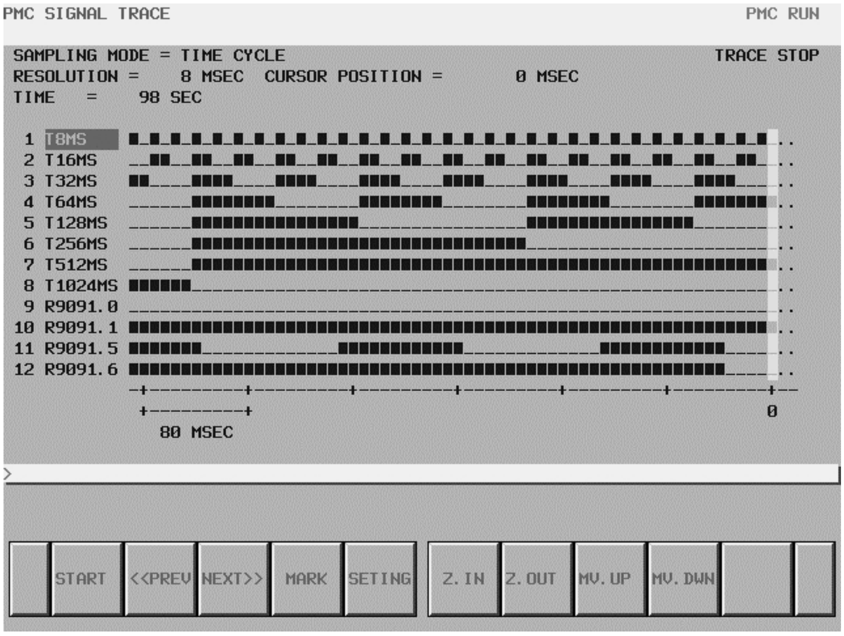


Execution of Trace screen (SIGNAL TRANSITION mode)

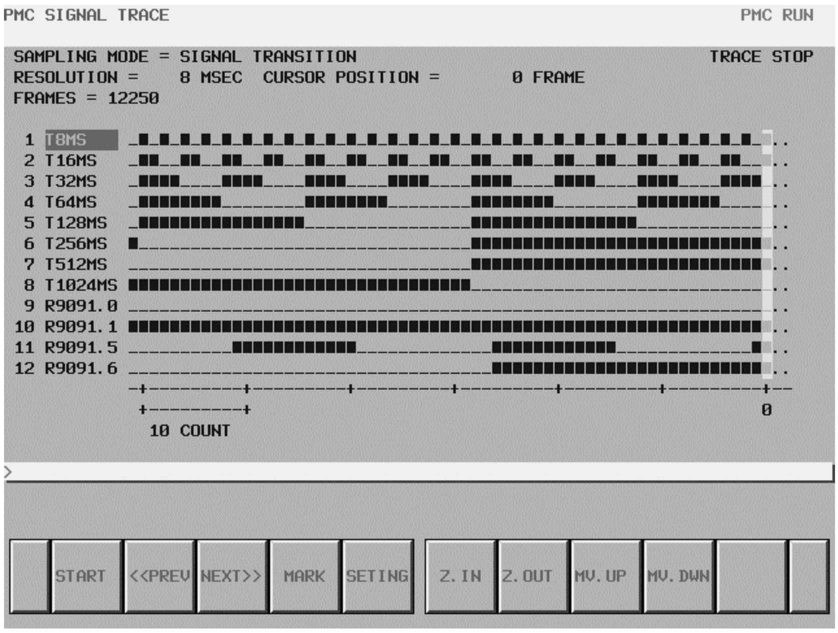
The result of trace is immediately displayed during execution of the trace. When the stop conditions that is set in parameter setting screen is satisfied the execution is finished. Pushing [STOP] soft key aborts the execution. In ”SIGNAL TRANSITION” mode, graphic display is not refreshed until any signal for sampling trigger changes.

6.5 OPERATION AFTER EXECUTION OF TRACE

When the execution is finished, the result of trace is displayed. The followings are the screen examples of trace by ”TIME CYCLE” and ”SIGNAL TRANSITION” mode.



Result of Trace screen (TIME CYCLE mode)



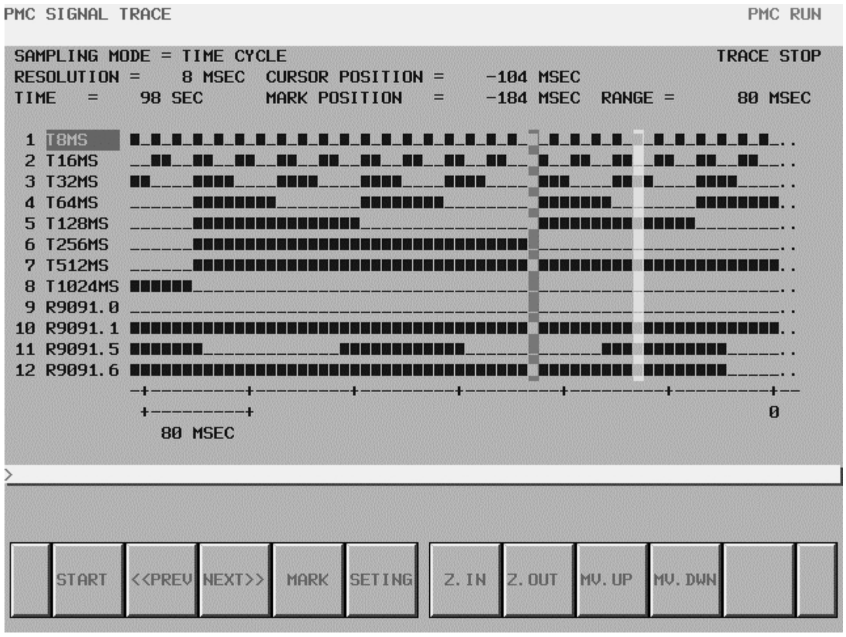
Result of Trace screen (SIGNAL TRANSITION mode)

The cursor indicating current position is initially displayed on the original point (0 point). The position of the cursor is displayed in ”CURSOR POSITION” in the upper of the screen. The cursor can move horizontally. After the execution, following operation is enabled.

a) Scroll of screen

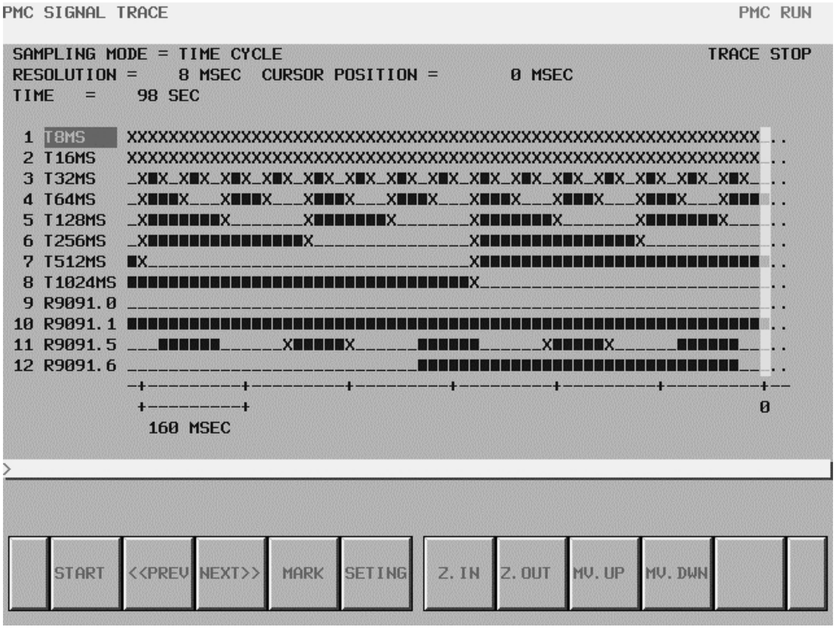
- Cursor up/down key and Page up/down key
  Enables the vertical scroll for the specified signal
- Cursor right/left key, [NEXT>>] soft key and [PREV<<] soft key
  Enables the horizontal scroll of the graph.

b) Automatic calculation of the selected range
Pushing [MARK] soft key marks the current position and displays the mark cursor. If the mark cursor duplicates with the current position cursor, the current position cursor has priority of display. The ”MARK POSITION” that shows the position of the mark cursor and ”RANGE” that shows the range between the mark cursor and the current position cursor are displayed in the upper of screen. Moving the current position cursor changes these values. Pushing [MARK] again releases the select range mode.



Result of Trace screen (Mark cursor display)

c) Zoom in/Zoom out of waveform
Pushing [Z.IN] soft key magnifies the display of chart. Pushing [Z.OUT] soft key reduces the display of chart. Pushing these soft keys also change the scale value of the graduation on the graph. When trace is just finished, the default zooming level was the most magnified level. In [Z.OUT] mode, ”X” is displayed as following screen example when the transitions of signal cannot be expressed accurately enough. The limitation of [Z.OUT] displays all of result of the trace in one page.



Result of Trace screen (Zoom out display)

d) Exchange of sampling signal
Pushing [MV.UP] soft key exchanges the signal indicated by the signal cursor for the signal one line above. Pushing [MV.DWN] soft key exchanges the signal indicated by the signal cursor for the signal one line below. The result of the operation is cancelled by the execution of trace or putting the power off. When you would like to preserve the order of displayed signals against the executing or powering off, please change the order on ”SAMPLING ADDRESS” screen.

6.6 AUTOMATIC START OF TRACE SETTING

Trace execution is automatically started after power–on by setting a PMC parameter.

	#7	#6	#5	#4	#3	#2	#1	#0
K906								

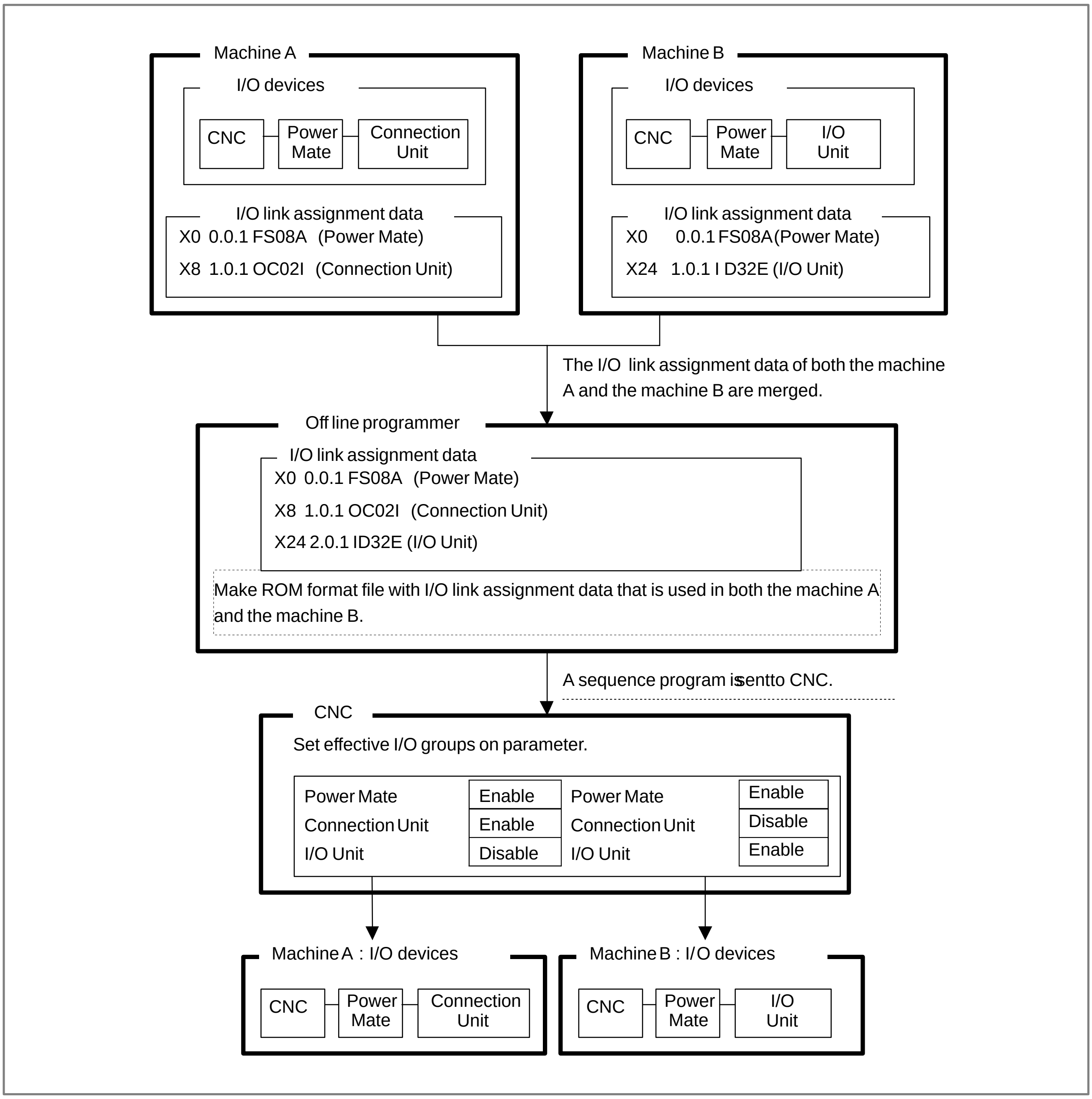
#5 0 : Starts trace execution when soft key [EXEC] is pressed.
1 : Automatically starts trace execution after power–on

This PMC parameter is set by the following item on the Setting screen of PMC Parameter.
– SIGNAL TRACE START = 0 (0: MANUAL 1: AUTO)

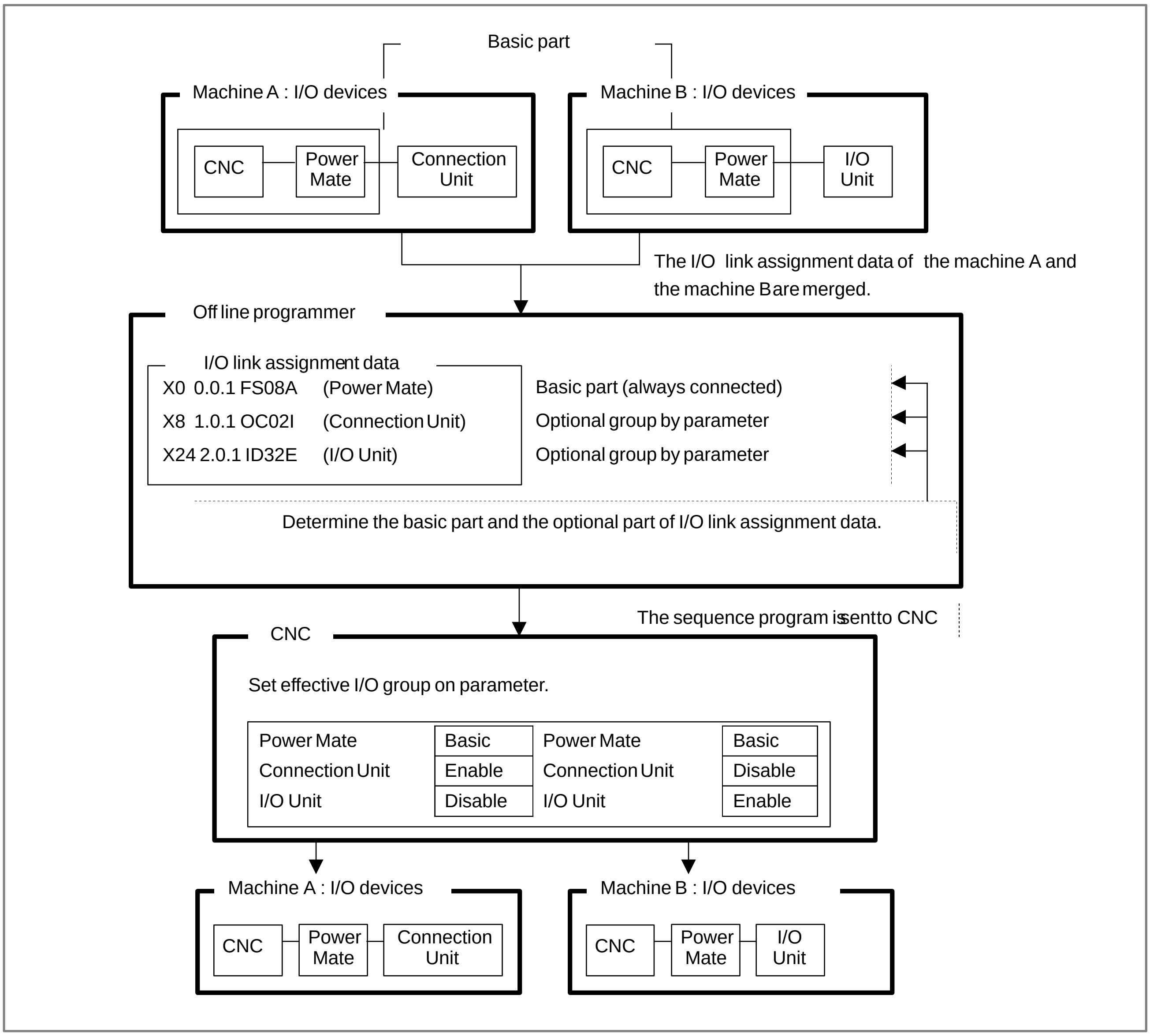
7 SELECTABLE I/O LINK ASSIGNMENT FUNCTION (PMC–SB7)

7.1 OUTLINE

This function enables the common use of a sequence program for several machines which have different I/O device configuration with each other, by setting the parameter to enable/disable each group in I/O link assignment data.

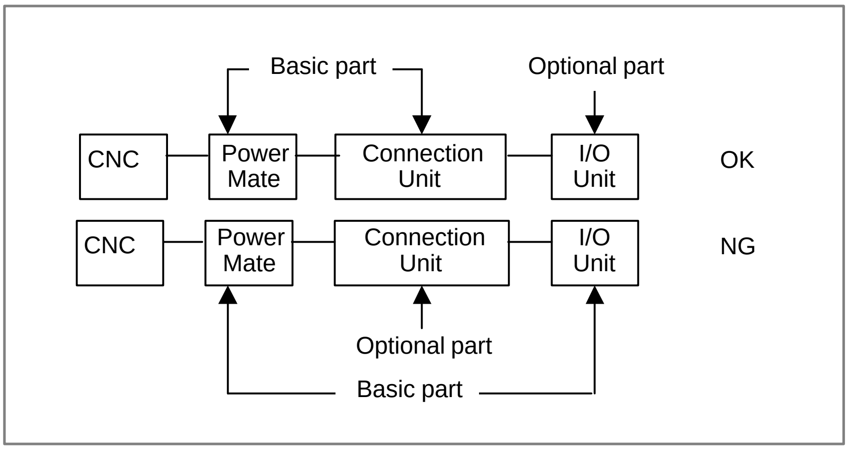


The I/O devices that are used in all machines can be set as basic part of configuration that is always effective.



NOTE

When you set a basic part, you have to assign devices of basic part continuously from group 0. And the basic part is connected with the top of the link.



This function requires setting the following parameters. These parameters can be set for each channel. See V–8 and V–2.4 for details.

1) ENABLE SELECTION:

Enables/Disables this function in the system parameter.

2) BASIC GROUP COUNT:

Sets the counts of group in basic part in the system parameter. (This part must be assigned continuously from group 0.) The basic groups in I/O link assignment data are always effective on all machine configurations.

3) EFFECTIVE GROUP SELECTION:

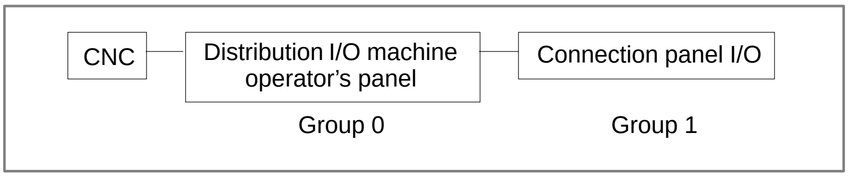
Sets the group of optional I/O device that is connected with each machine in the setting parameter. This parameter doesn't affect the basic part.

> **NOTE**
> I/O Link expansion option is necessary to use channel 2 of I/O Link.

7.2 EXAMPLE

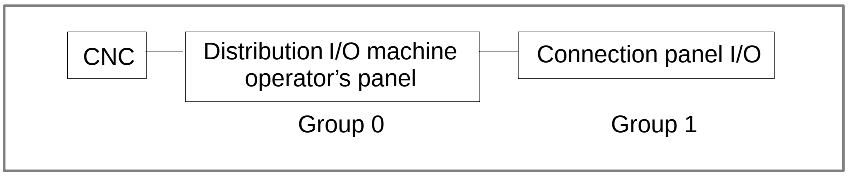
There are three machines which have different configurations of I/O devices, each other.

- Configuration A
  A machine which has a distribution I/O machine operator's panel and a connection panel I/O connected with channel 1 of NC.
  Channel 1



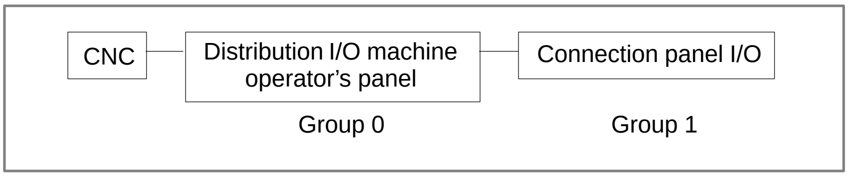
Channel 2
No connection

- Configuration B
  A machine which has a distribution I/O machine operator's panel and a Power Mate connected with channel 1 of NC.
  Channel 1

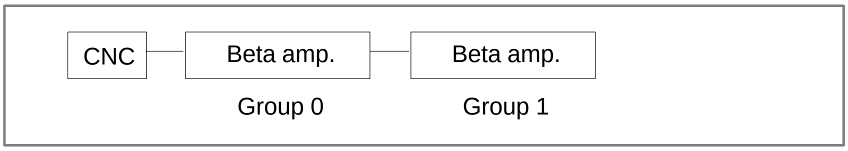


Channel 2
No connection

- Configuration C
  A machine which has the configuration A on channel 1 and two beta amplifiers on channel 2.
  Channel 1



Channel 2



These machines can use a common sequence program which has I/O link assignment data that includes all I/O device configurations. The contents of parameters for each I/O device configuration are as shown in next page.

(1) The contents of I/O link assignment data in sequence program

Channel 1

Address	Group	Base	Slot	Name	I/O Device
X0000	0	0	1	CM12I	Distribution I/O machine operator's panel
:	:	:	:	:	:
X0020	1	0	1	CM03I	Connection panel I/O
:	:	:	:	:	:
X0030	2	0	1	FS08A	Power Mate
:	:	:	:	:	:

Channel 2

Address	Group	Base	Slot	Name	I/O Device
X0200	0	0	1	OC02I	Beta amp.
:	:	:	:	:	:
X0220	1	0	1	OC02I	Beta amp.
:	:	:	:	:	:

(2) The contents of parameter

- Configuration A

i) System parameter

```
1CH   ENABLE SELECTION  = YES
      BASIC GROUP COUNT = 1
2CH   ENABLE SELECTION  = YES
      BASIC GROUP COUNT = 0
```

ii) Setting parameter

```
Group NO.  :  00 01 02 03 04 05 06 07
(CH1)          *  1  0  0  0  0  0  0
              08 09 10 11 12 13 14 15
               0  0  0  0  0  0  0  0
Group NO.  :  00 01 02 03 04 05 06 07
(CH2)          0  0  0  0  0  0  0  0
              08 09 10 11 12 13 14 15
               0  0  0  0  0  0  0  0
```

- Configuration B

i) System parameter

```
1CH    ENABLE SELECTION  = YES
       BASIC GROUP COUNT = 1
2CH    ENABLE SELECTION  = YES
       BASIC GROUP COUNT = 0
```

ii) Setting parameter

```
Group NO.    :  00 01 02 03 04 05 06 07
(CH1)            *  0  1  0  0  0  0  0
                08 09 10 11 12 13 14 15
          0  0  0  0  0  0  0  0
Group NO.    :  00 01 02 03 04 05 06 07
(CH2)            0  0  0  0  0  0  0  0
                08 09 10 11 12 13 14 15
          0  0  0  0  0  0  0  0
```

- Configuration C

i) System parameter

```
1CH    ENABLE SELECTION  = YES
       BASIC GROUP COUNT = 1
2CH    ENABLE SELECTION  = YES
       BASIC GROUP COUNT = 0
```

ii) Setting parameter

```
Group NO.    :  00 01 02 03 04 05 06 07
(CH1)            *  1  0  0  0  0  0  0
                08 09 10 11 12 13 14 15
          0  0  0  0  0  0  0  0
Group NO.    :  00 01 02 03 04 05 06 07
(CH2)            1  1  0  0  0  0  0  0
                08 09 10 11 12 13 14 15
          0  0  0  0  0  0  0  0
```

(3) The actual contents of I/O link assignment data modified by the parameter

- Configuration A

Channel 1

Address	Group	Base	Slot	Name	I/O Device
X0000	0	0	1	CM12I	Distribution I/O machine operator's panel
:	:	:	:	:	:
X0020	1	0	1	CM03I	Connection panel I/O
:	:	:	:	:	:

Channel 2

No connection

- Configuration B

Channel 1

Address	Group	Base	Slot	Name	I/O Device
X0000	0	0	1	CM12I	Distribution I/O machine operator's panel
:	:	:	:	:	:
X0030	1	0	1	FS08A	Power Mate
:	:	:	:	:	:

Channel 2

No connection

- Configuration C

Channel 1

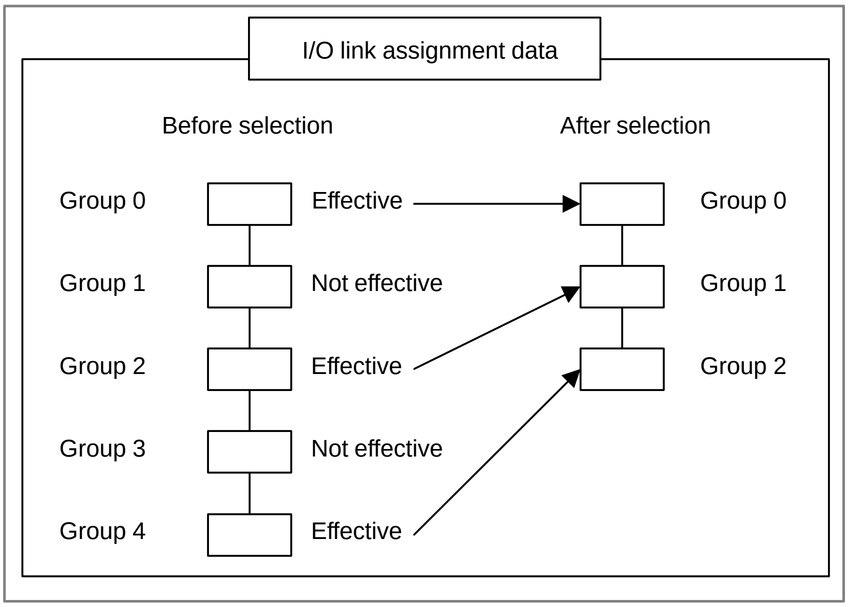
Address	Group	Base	Slot	Name	I/O Device
X0000	0	0	1	CM12I	Distribution I/O machine operator's panel
:	:	:	:	:	:
X0020	1	0	1	CM03I	Connection panel I/O
:	:	:	:	:	:

Channel 2

Address	Group	Base	Slot	Name	I/O Device
X0200	0	0	1	OC02I	Beta amp.
:	:	:	:	:	:
X0220	1	0	1	OC02I	Beta amp.
:	:	:	:	:	:

7.3 NOTES

1) If PMC–parameters are cleared, cycling the power of CNC links only the basic part.
2) After selecting the assignment data, the I/O devices are linked with shifted group number of effective I/O link assignment data.
   You can check the result of connection by "5. I/O Link Connecting Check Screen".

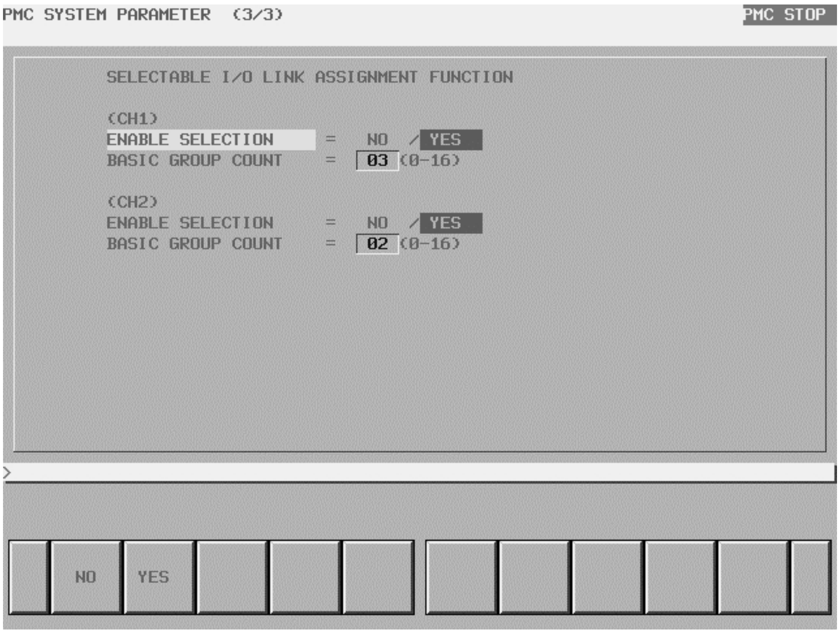


3) You can not exchange the order of the I/O group number.

WARNING

1) If a sequence program in which "ENABLE SELECTION" is set to "YES" is executed with the PMC control software that is not applied to this function, "ENABLE SELECTION" is ignored and all I/O link assignment data are effective.
2) The [IOSTRT] key on I/O unit address setting screen has been removed because improper use of this key may increase mistake of setting the I/O assignment data which causes wrong linking of the I/O devices and it may result in unexpected malfunctions of machine. If you want to have the machine linked with I/O devices under the selected I/O link assignment data, you have to turn off and on power after the confirmation of the correct connection of the I/O devices.
3) If a sequence program in which the system parameter for this function is set is de–compiled and compiled on the FAPT LADDER–III or Ladder Editing Package that does not support this function, the system parameter for this function is initialized and all of I/O link assignment data becomes effective. Please set the system parameters again, when writing the sequence program into the CNC.

8 SYSTEM PARAMETER (PMC–SB7)



System parameter screen

(1) ENABLE SELECTION

This parameter enables or disables Selectable I/O link assignment function.

This parameter is set by softkey [NO] or [YES].

NO: Don't use Selectable I/O link assignment function.

YES: Use Selectable I/O link assignment function.

The default setting is [NO].

If the parameters of both channel 1 and channel 2 are [NO], the setting screen for this function explained on ”3.22 Setting parameter” is not displayed.

> **WARNING**
> According to actually connected I/O devices, please set the setting parameter(K910–K913 described in 3.2.2) correctly to enable this function.

(2) BASIC GROUP COUNT

This parameter determines the number of groups in the basic part of the I/O link assignment data. You can set the group count for basic part by entering (0–16) followed by softkey[INPUT]. The default is 0.

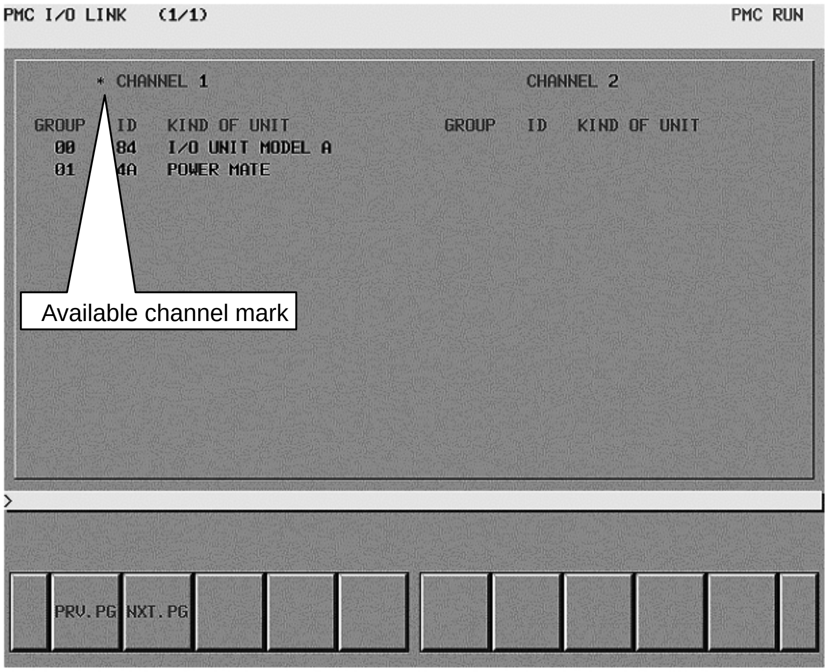
Example:

3:	basic part	0–2	group
	optional part	3–15	group

> **NOTE**
> For other system parameters, see Chapter 4 in Part III.

9 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, any I/O device is not displayed. If there is a problem of input or output signals for I/O devices, check the connection of I/O Link by referring to this screen.



I/O Link Connecting Check Screen

Available channel mark :
If channel is available, an ”*” mark is displayed. The status in this example screen shows that the channel 2 is not available.

> **NOTE**
> 1 I/O Link expansion option is necessary to use the channel 2 of I/O Link.
> 2 For I/O devices and ID codes, sample I/O link configurations, and an example of the I/O link connecting check screen, see Subsection 3.8.1 in Part II.

10 ONLINE FUNCTION

There is the following function in the online function of FAPT LADDER–II, FAPT LADDER–III or Ladder Editing Package.

– 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
  When you use the online function, the setting of communication condition is necessary in advance.

CAUTION

1 When one of the following screens is displayed at PMC, the online communication can not be used. Change to other screens from the following screens, and use the online function.
[PMCLAD], [I/O], [EDIT], [SYSPRM], [TRACE], [USRDGN], [DBGLAD], [GDT], [USRMEM]
Also, you can not use the above screens at PMC during the online communication.

2 When the online function is used with RS–232C, the selected channel is occupied by the PMC system. To use other input/output functions with RS–232C, specify other channel setting than the one used by online function.

3 Loader control function can not connect with FAPT LADDER–III or Ladder Editing Package by Ethernet.

10.1 ONLINE SETTING SCREEN

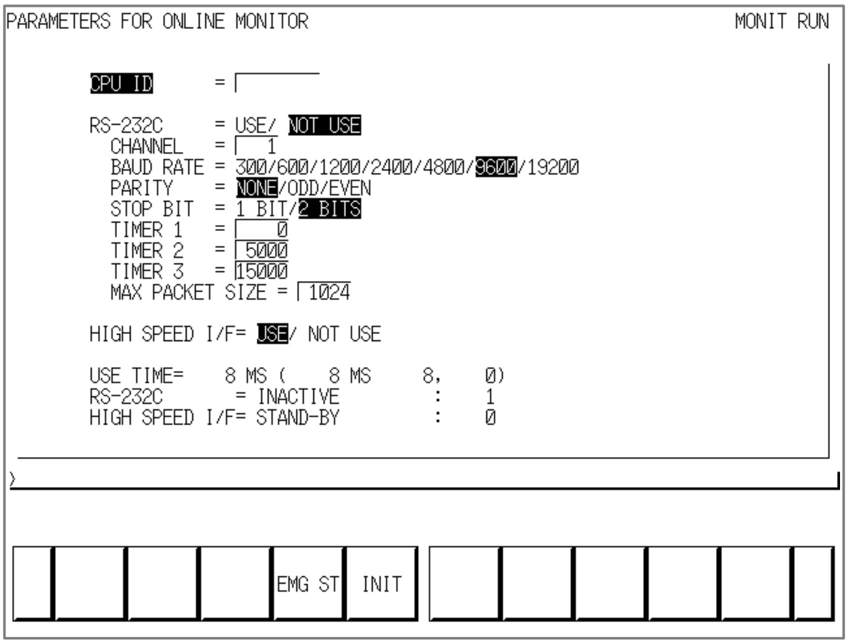


Fig. 10.1 Online monitor setting screen

Meanings of Soft key

EMG ST: Terminates communication forcibly. Use this key if communication becomes abnormal and the connection cannot be terminated normally.

INIT: Initializes the parameters to their default values.

NOTE

1. In case of configuration of CNC with which neither Ethernet nor HSSB is available, the item of ”HIGH SPEED I/F” is not displayed.
2. In case of display which has 5+2 soft key, two pages are used for this setting screen.
   Switch the page by < Page Up > or <Page Down> key.

10.2 SETTING OF ONLINE CONNECTION

To communicate with FAPT LADDER–II, FAPT LADDER–III or Ladder Editing Package, you need to put the PMC system in waiting situation of the connection. There are two ways for setting this, setting at PMC screen and setting in NC parameter. Also, there are three connection types, for example Ethernet, RS–232C or HSSB.

10.2.1 How to Set at PMC Screen

To display the soft key [MONIT] in the PMC main menu screen, set ”PROGRAMMER
ENABLE” to ”YES” in the setting screen. When pushing the soft key [MONIT]→[ONLINE], the online setting screen is displayed. (Fig. 7.1)

1. Case of connection by RS–232C (FAPT LADDER–II, FAPT LADDER–III)
   (1) Check that ”NOT USE” is selected at the ”RS–232C” item.
   (2) Set the parameter of ”CHANNEL” and ”BAUD RATE”.
   (3) Move the cursor to the ”RS–232C” item with Up or Down Cursor key.
   (4) Select ”USE” with Left or Right Cursor key.
2. Case of connection by Ethernet (FAPT LADDER–III, Ladder Editing Package)
   (1) Move the cursor to the ”HIGH SPEED I/F” item with Up or Down Cursor key.
   (2) Select ”USE” with Left or Right Cursor key.
3. Case of connection by HSSB (Ladder Editing Package)
   (1) Move the cursor to the ”HIGH SPEED I/F” item with Up or Down Cursor key.
   (2) Select ”USE” with Left or Right Cursor key.

NOTE

1. When both ”RS–232C = USE” and ”HIGH SPEED I/F = USE” are selected, the PMC system will communicate with the application which is connected at first. If PMC system is already connecting with an application, it can not connect with other applications.
2 When you use the online function by Ethernet, the setting of Ethernet parameters at CNC is necessary in advance.
3 Loader control function can not connect with FAPT LADDER–III or Ladder Editing Package by Ethernet.

10.2.2 Setting of Online Connection by NC Parameter

You can enable and disable the online connection for Ethernet, HSSB and RS–232C by NC parameter No.24 without setting on the PMC online monitor setting screen. This NC parameter is made effective immediately after setting the parameter.
If the value of this parameter is changed, the item ”RS232C”and ”HIGH SPEED I/F” in the online monitor screen are automatically changed too.
Please refer to following table.

Contents of NC parameter No.24.

NC parameter No.24	Meanings	Each item of the online monitor screen after setting	
		RS–232C	HIGH SPEED I/F
0	The settings on the online monitor setting screen are effective.	This does not affect ”RS–232C” and ”HIGH SPEED I/F”.	
1	Enables ”Channel 1 of RS–232C” and disables ”HIGH SPEED I/F”.	USE (Channel 1)	NOT USE
2	Enables ”Channel 2 of RS–232C” and disables ”HIGH SPEED I/F”.	USE (Channel 2)	NOT USE
10	Disables ”RS–232C” and enables ”HIGH SPEED I/F”.	NOT USE	USE
11	Enables ”Channel 1 of RS–232C” and ”HIGH SPEED I/F”.	USE (Channel 1)	USE
12	Enables ”Channel 2 of RS–232C” and ”HIGH SPEED I/F”.	USE (Channel 2)	USE
3 to 10 13 to 254	3 to 1013 to 254 Reserve (Don’t use this setting.)	(Reserved)	(Reserved)
255	Terminates communication forcibly.It is the same effect as soft key [EMG ST].	NOT USE	NOT USE

How to set the parameter

(1) Display the No.24 of NC parameter.

(2) To connect by Ethernet or HSSB, input ”10”, ”11” or ”12”.
To connect by RS–232C, input ”1”, ”2”, ”11” or ”12”.

NOTE
1 Even if the setting of the online monitor screen of PMC is changed, the value of No.24 in NC parameter is not changed.
2 When you use the ladder editing package on the open CNC or you use the FAPT LADDER–III to edit the ladder and do not use the ladder monitor and the ladder editor on CNC screen, the setting of this parameter should be used

10.3 ONLINE FUNCTION BY ETHERNET

10.3.1 Setting of Ethernet Parameters

When you try to connect FAPT LADDER–III or Ladder Editing Package (Window version) with CNC by Ethernet, it is necessary to set some Ethernet parameters. The setting of Ethernet parameters can be set in the following Ethernet parameter screen of CNC. Please refer to ”FANUC Ethernet Board/DATA SERVER Board OPERATOR’S MANUAL” (B–63354EN) about the detail of the setting screen and setting parameters. The setting item necessary for Ethernet connection for PMC online function is as follows.

– IP ADDRESS	(Set the IP address of CNC. 192.168.0.1 etc.)
– SUBNET MASK	(Set the mask address of the IP address. 255.255.255.0 etc.)
– ROUTER IP ADDRESS	(If you use the router, set the Router IP Address.)
– PORT NUMBER (TCP)	(8193 etc.)

```
ETHERNET PARAMETER
                              PAGE: 1/ 2
 MAC ADDRESS              XXXXXXXXXXXX
 NUMBER OF SCREENS                  18
 MAXIMUM PATH                        1
 HDD EXISTENCE                       0

 IP ADDRESS           192.168.0.1
 SUBNET MASK          255.255.255.0
 ROUTER IP ADDRESS
```

```
ETHERNET PARAMETER
                              PAGE: 2/ 2
(DNC1/ETHERNET)
 PORT NUMBER(TCP)                 8193

 PORT NUMBER(UDP)                    0

 TIME INTERVAL                       0
```

Fig. 10.3.1(a) Ethernet Board

```
ETHERNET PARAMETER(EMBEDD)
                              PAGE: 1/ 5

 MAC ADDRESS              XXXXXXXXXXXX

(COMMON PARAMETER)
 IP ADDRESS           192.168.0.2

 SUBNET MASK          255.255.255.0

 ROUTER IP ADDRESS
```

```
ETHERNET PARAMETER(EMBEDD)
                              PAGE: 2/ 5
(FOCAS1/ETHERNET)
 PORT NUMBER(TCP)                 8193

 PORT NUMBER(UDP)                    0

 TIME INTERVAL                       0
```

Fig. 10.3.1(b) Embedded Ethernet

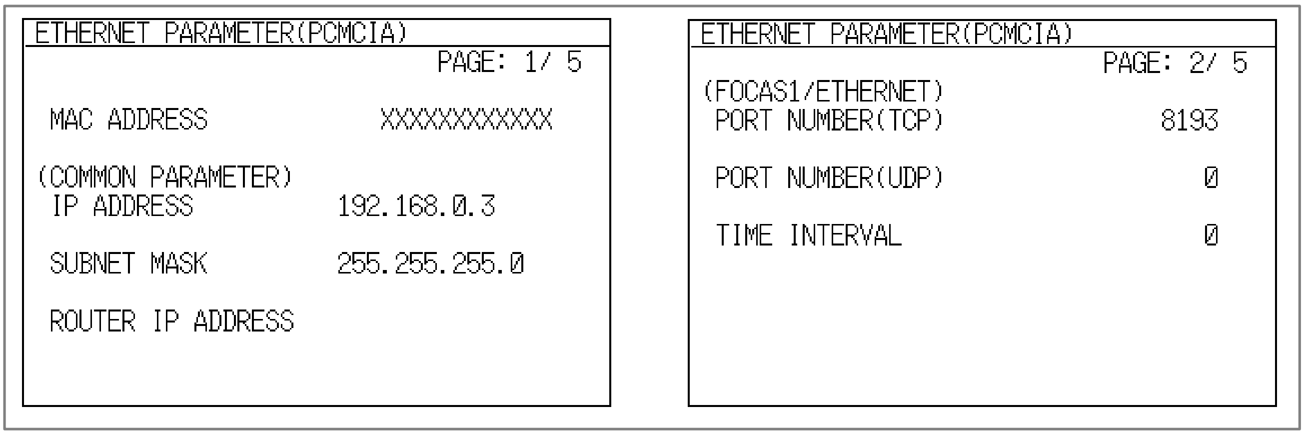


Fig. 10.3.1(c) Ethernet Card (PCMCIA)

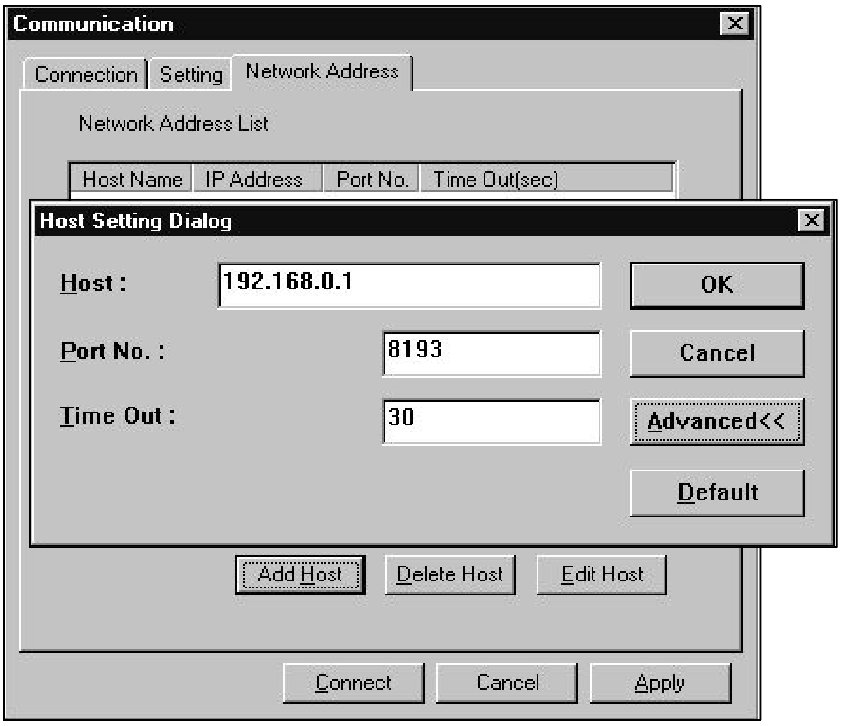
10.3.2 Starting online communication by offline programmer (Ethernet connection)

The procedures for online connection with PMC and the offline programmer (FAPT LADDER–III, Ladder Editing Package for windows) by Ethernet are as follows.

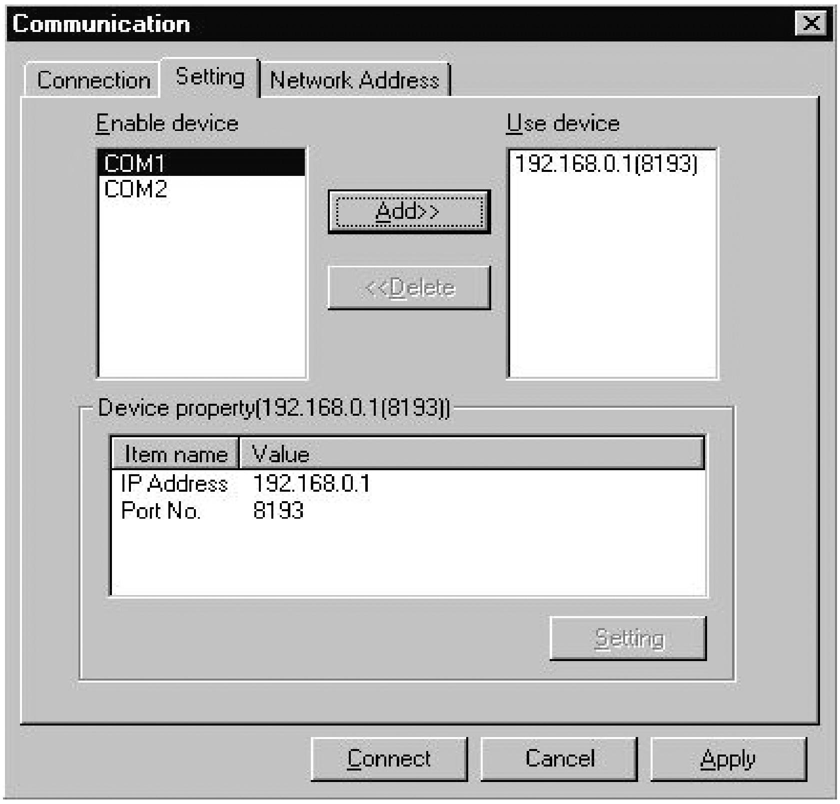
(Example: FAPT LADDER–III)

(1) Start up FAPT LADDER–III, and click the [Communication] on [Tool] menu.

(2) Select the [Network Address] tab and push the <Add Host> button. Input the ”IP Address” and ”Port No.” inputted by ”7.3.1 Setting of Ethernet parameters”.



(3) Select the [Setting] tab, and add the IP Address to ”Use device”.



(4) Push the <Connect> button for start of the communication.

Ladder Editing Package can be connected by the same operation too. Refer to ”FAPT LADDER–III OPERATOR’S MANUAL”(B–66234EN) as for the detail of operation of FAPT LADDER–III. Refer to ”LADDER EDITING PACKAGE (Windows) OPERATOR’S MANUAL” (B–63484EN) as for the detail operation of Ladder Editing Package.

NOTE

1 When one of the following screens is displayed at PMC, the online communication can not be used. Change to other screens from the following screens, and use the online function.
[PMCLAD], [I/O], [EDIT], [SYSPRM], [TRACE], [USRDGN], [DBGLAD], [GDT], [USRMEM].

2 When the online function is used with RS–232C, the selected channel is occupied by the PMC system. To use other input/output functions with RS–232C, specify other channel setting than the one used by online function.

3 Loader control function can not connect with FAPT LADDER–III or Ladder Editing Package by Ethernet.

10.4 COMMUNICATION STATUS

The communication status of RS–232C and HIGH SPEED I/F are displayed at the online monitor screen during the online communication.

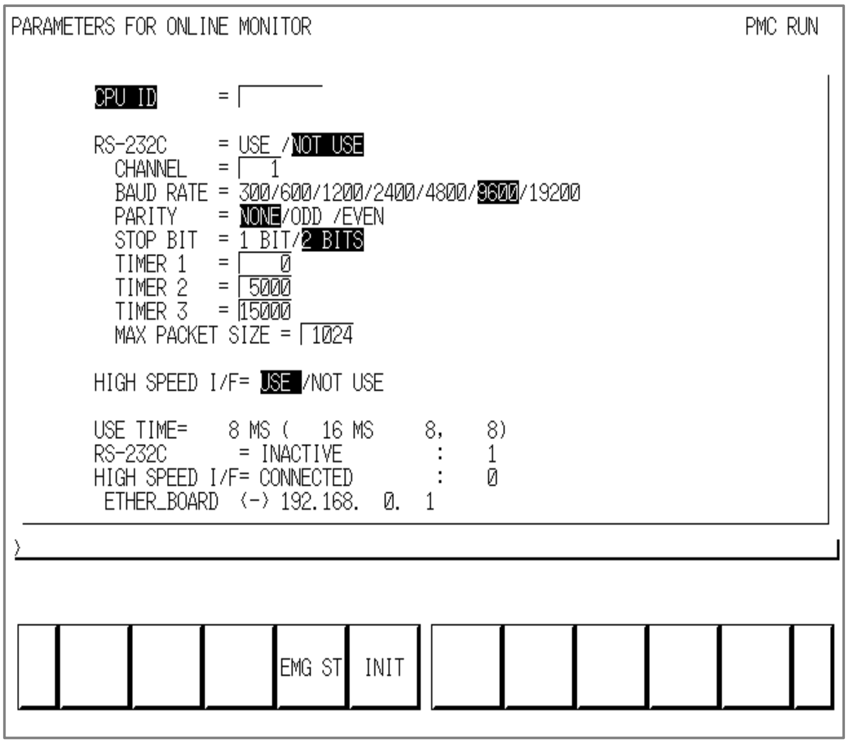


Fig. 10.4 Online monitor setting screen

USE TIME:	The maximum time in the communication processing is displayed.
RS–232C:	The communication condition of RS–232C is displayed.
HIGH SPEED I/F:	The communication condition of HIGH SPEED I/F is displayed.
ETHER_BOARD:	Displayed during the communication with Ethernet board.The IP address of the communication partner is displayed.
EMB_ ETHERNET:	Displayed during the communication with embedded Ethernet.The IP address of the communication partner is displayed.
HSSB:	Displayed during the communication with HSSB.

The display messages and the meanings are shown in the table of below.

Displayed messages	Meanings
INACTIVE	The communication is inactive.
STOPPING	The communication is being stopped.(Wait for the termination of communication)
STARTING	The communication is being started.(Wait for the termination of communication over another communication path)
STAND–BY	The communication is active and in standby mode.
CONNECTED	The communication is active and being connected.
NO OPTION	The port can be not opened because there is not option of RS–232C.
BAD PARAMETER	Invalid open parameters are specified.
TIMEOUT ERROR	A time–out has occurred and communication is aborted.
TIMEOUT(K) ERROR	A time–out has occurred and communication is aborted.
BCC ERROR	A Block Check Code (packet parity) error has occurred.
PARITY ERROR	A parity error has occurred.
OVER–RUN ERROR	A reception overrun has occurred and the communication can not recover.
SEQUENCE ERROR	Packets are out of sequence.(Incorrect procedure)
DATA ERROR	Incorrect packets have been received through retry process.
QUEUE OVERFLOW	The transmit/receive queue has overflowed.
DISCONNECTED	Communication has been terminated successfully.
NO CONNECTION	The cable is disconnected.

10.5 ABOUT CONNECTION LOG OF ETHERNET

If any errors have occurred during Ethernet connection, the contents of the errors are displayed at ”ETHLOG” screen of CNC. Refer to this screen when the communication does not start.

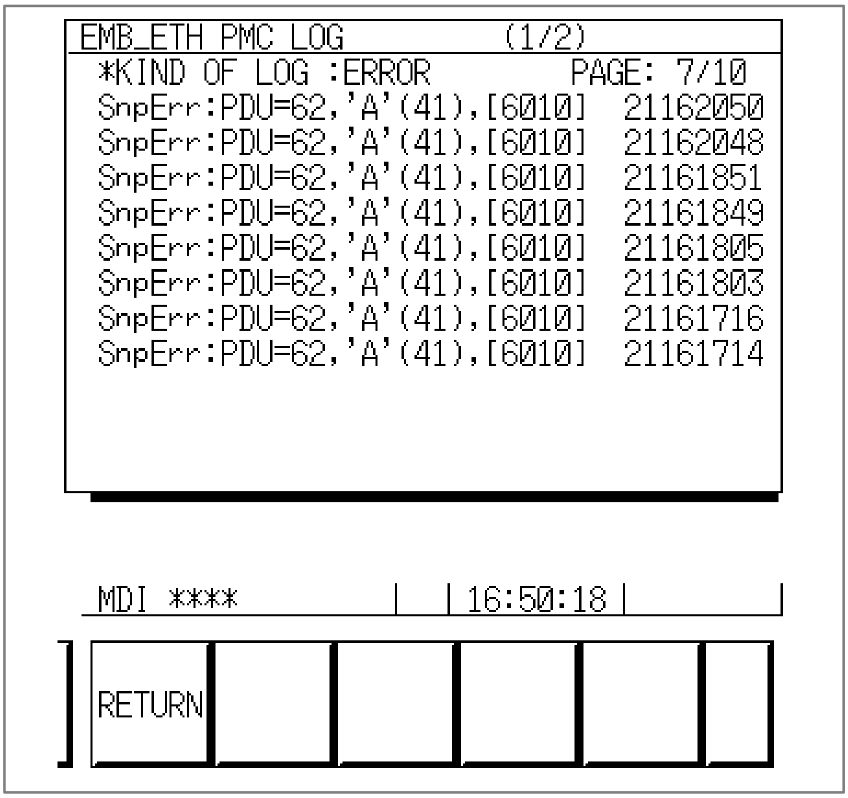


Fig. 10.5(a) The Log screen of embedded Ethernet

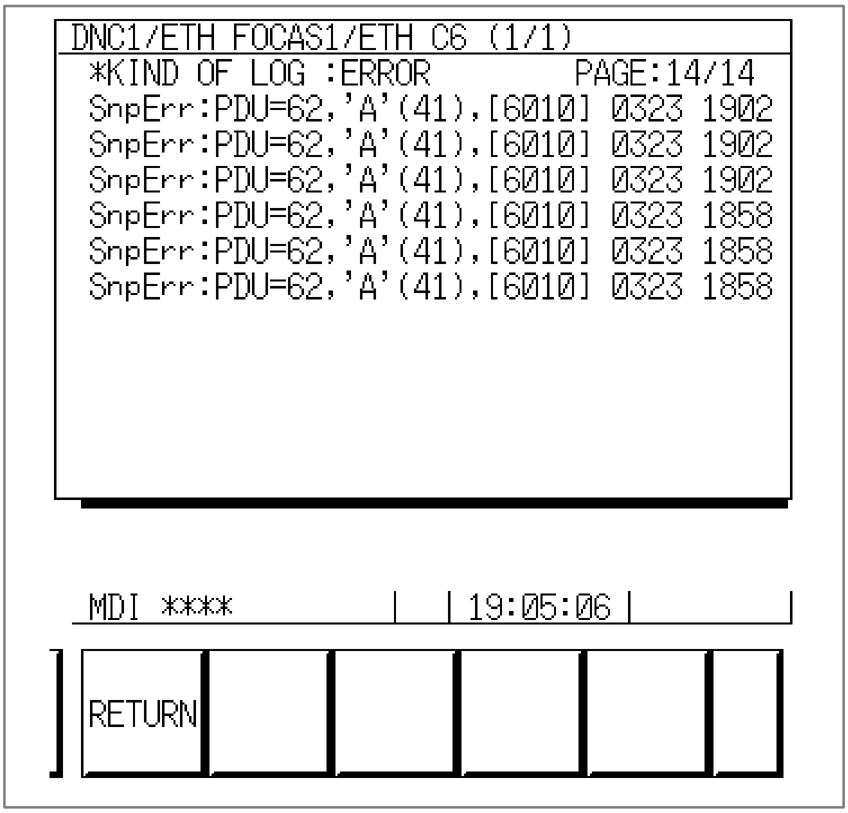


Fig. 10.5(b) The Log screen of Ethernet

Connection Log	Meanings and countermeasures
SnpErr:PDU=*m*,*n*,[x] *date time* SnpErr:PDU=*n*,[x] *date time* SnpErr:TaskTimeOut[x] *date time*	An error has occurred during the online communication. *m, n* : Online communication information that is internal information of a system. *x* : Error information 6001 PMC does not support the Ethernet.Confirm the Series/Edition of PMC software. 6003 Unsupported command data was received.Confirm the Series/Edition of Ethernet board software. 6004 There was an error in command dataConfirm the Series/Edition of Ethernet board software. 6005 PMC does not receive command data.Confirm the communication status at the online setting screen of PMC. 6010 PMC does not receive command data.Confirm if ”HIGH SPEED I/F=USE” is selected and other application is not connected at the online setting screen of PMC. 6011 Time–out error occurred at PMC.Increase the value of ”Time Out” in [Network Address] of [Communication] menu for FAPT LADDER–III or Ladder Editing Package. 6012 PMC does not receive command data because it is busy for processing.Confirm the communication status at the online setting screen of PMC. 6013 Time–out error occurred at PMCIncrease the value of ”Time Out” in [Network Address] of [Communication] menu for FAPT LADDER–III or Ladder Editing Package. 6101 PMC received an unsupported function code.Confirm the Series/Edition of PMC software. *date time*: The time when the error occurred. Ex.) ”0323” means March 23rd. ”1858” means 6:58 PM. ”21161714” means 21st 4:17PM 14 seconds.

Index

Note
Volume 1 : Up to Page 898 / Volume 2 : Page 899 and later

≪A≫

≪B≫

≪C≫

Note
Volume 1 : Up to Page 898 / Volume 2 : Page 899 and later

Note
Volume 1 : Up to Page 898 / Volume 2 : Page 899 and later

≪F≫

≪G≫

≪H≫

Note
Volume 1 : Up to Page 898 / Volume 2 : Page 899 and later

Note
Volume 1 : Up to Page 898 / Volume 2 : Page 899 and later

Note
Volume 1 : Up to Page 898 / Volume 2 : Page 899 and later

≪R≫

Note
Volume 1 : Up to Page 898 / Volume 2 : Page 899 and later

≪S≫

Note
Volume 1 : Up to Page 898 / Volume 2 : Page 899 and later

Note
Volume 1 : Up to Page 898 / Volume 2 : Page 899 and later

≪*T*≫

≪*U*≫

≪*W*≫

Note
Volume 1 : Up to Page 898 / Volume 2 : Page 899 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
07	Apr.,'95	Total revision	14	Jul., 2001	Addition of PMC–SB7
06	Nov.,'94	Corresponds to 18–B	13	May, 2000	• Addition of ladder display/editing function for PMC–NB6 • Addition of expansion of I/O Link points for FS 16*i*/18*i*/21*i* • Addition of window functions (tool life management B function and so forth) • Correction of errors
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	12	Mar., '99	Addition of PMC–NB6
04	Aug.,'93	Addition of PMC–MODEL PA1/PA3/RA3/RB3/RC3/NB.	11	Nov., '98	Addition of Power Mate *i*
03	Mar., '92	Addition of PMC–MODEL RA1/RA2/RB2	10	Dec., '97	Corresponds to 16*i*/18*i*/21*i*–MODEL A
02	Aug., '91	All pages are revised. PMC–MODEL RC is added.	09	Mar.,'96	Addition of PMC–NB2
01	Oct., '90	————	08	Oct.,'95	Addition of PMC–RB5/RB6