YASNAC J50 PC SYSTEM
INSTRUCTIONS

CNC SYSTEM FOR TURNING APPLICATIONS / MACHINING CENTERS

Upon receipt of the product and prior to initial operation, read these instructions thoroughly, and retain for future reference.

YASKAWA

TOE-C843-12.1B

CONTENTS

INDEX

INDEX (Cont'd)

1. OUTLINE

1. The programmable controller (called PC hereafter) for YASNAC J50L/J50M stands between the standard YASNAC NC unit and the machine tool. It facilitates the compact and efficient utilization of the sequence control required by the machine tool through the software.
2. Sequence program editing of PC can be performed efficiently with CRT'; NC and JSD modes are easily changed and selected.
3. The PC is optional and it is installed in the NC unit, if selected.
4. In this manual, "PC programming method" (Selections 1 to 8) and "Sequence program editing unit and the operating method" (Sections 9 and 10) have been explained so that the users to facilitate the use of the above described PC.

2. BLOCK DIAGRAM

The block diagram of the PC system for YASNAC J50L/J50M is shown in Fig. 2.1.

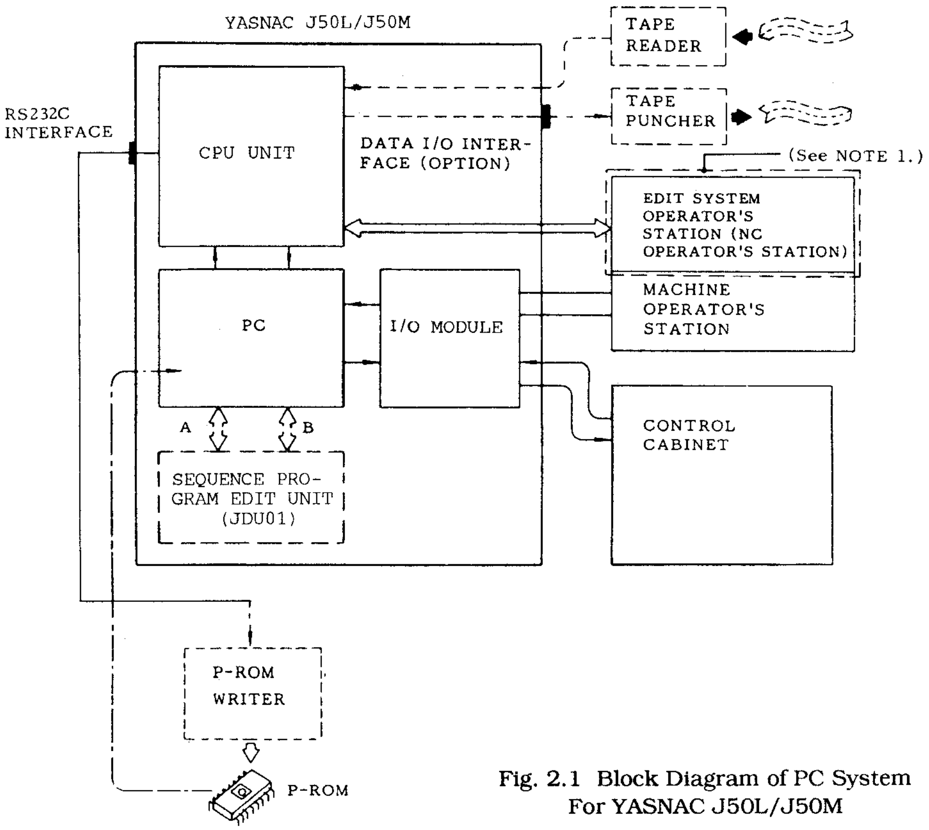

YASNAC J50L/J50M
RS232C INTERFACE
CPU UNIT
DATA I/O INTERFACE (OPTION)
TAPE READER
TAPE PUNCHER
(See NOTE 1.)
EDIT SYSTEM OPERATOR'S STATION (NC OPERATOR'S STATION)
MACHINE OPERATOR'S STATION
PC
I/O MODULE
CONTROL CABINET
A
B
SEQUENCE PROGRAM EDIT UNIT (JDU01)
P-ROM WRITER
P-ROM

Fig. 2.1 Block Diagram of PC System For YASNAC J50L/J50M

- Solid line shows the YASNAC CNC unit provided with P.C.
- Broken line shows the sequence program edit system temporarily used by incorporating the sequence program edit system (JDU01) in YASNAC.

Note:

1. When the control is used as sequence program edit system, the operator's panel with CRT display changes to the sequence program edit panel.
2. Sequence program edit system (JDU01) can be mounted on the CPU rack.
3. P-ROM writer which is commercially available may be used. It is used to write the completed sequence edited and checked into P-ROM.
4. Tape reader is used to load List Tape in which sequence ladder is coded or P-ROM Format Tape consisting of machine language into sequence edit system.
5. Tape puncher punches out the completed sequence edited and checked in the form of List Tape or P-ROM Format Tape.

3. SPECIFICATIONS

3.1 FUNDAMENTAL SPECIFICATIONS

(1) Control method: Scanning method

(2) Processing time:
Approx. 2.7 μ sec/step
High speed scanning time - 8 msec
Low speed scanning time - 8 msec × n (n is determined by the capacity of the total program.)

(3) Program memory capacity :
Memory element : EPROM (1024 k bits/one)
Basic – 64 k bytes (1 EPROM)
(64 k bytes corresponds to approximately 16000 steps in basic instruction.)

(4) Types of instruction language :
Basic instruction – 59 types
Macro instruction – 11 types

3.2 PROGRAM FUNCTIONS

(1) Internal relay:	4000 points
(2) Register:	500 (8 bits/one)
(3) Timer:	94 (5 types)

- 8 msec - 2.4 sec, 20 ea.
- 50 msec - 12.75 sec, 30 ea.
- 100 msec - 25.5 sec, 30 ea.
- 1 sec - 255 sec , 10 ea.
- 1 min - 255 min, 4 ea.

(4) Sequencer parameter:	100
(5) Keep relay:	7200
(6) Keep memory:	900 (8 bits/memory)

3.3 MACRO INSTRUCTIONS

Following 11 types of macro instructions can be used.

	Instruction word
(1) Rise signal detection:	SUBP 003
(2) Fall signal detection:	SUBP 004
(3) Counter:	SUBP 005

Functions--Ring counter or preset counter or up-down counter.

Counting range--0 - 9999

(4) Rotation	SUBP 006
(5) Code conversion	SUBP 007
(6) Pattern clear	SUBP 009
(7) Parity check	SUBP 011
(8) Data conversion:	SUBP 014
(9) Data search	SUBP 017
(10) Index data transfer	SUBP 018
(11) Message display	SUBP 023 -- Optional function

3.4 INPUT/OUTPUT SPECIFICATIONS

(1) Standard I/O boards

<FC810 (FC860)>

(i) DC input :	112 points
(ii) Non contact output :	96 points

<FC861>

(i) DC input :	64 points
(ii) Non contact output :	56 points

(2) CRT panel built-in I/O boards

<SP50-1>

(i) DC input :	64 points
(ii) Non contact output :	32 points

<SP50-2>

(i) DC input :	64 points
(ii) Non contact output :	56 points

Note:

1. The detail of basic instructions are given in the following table.

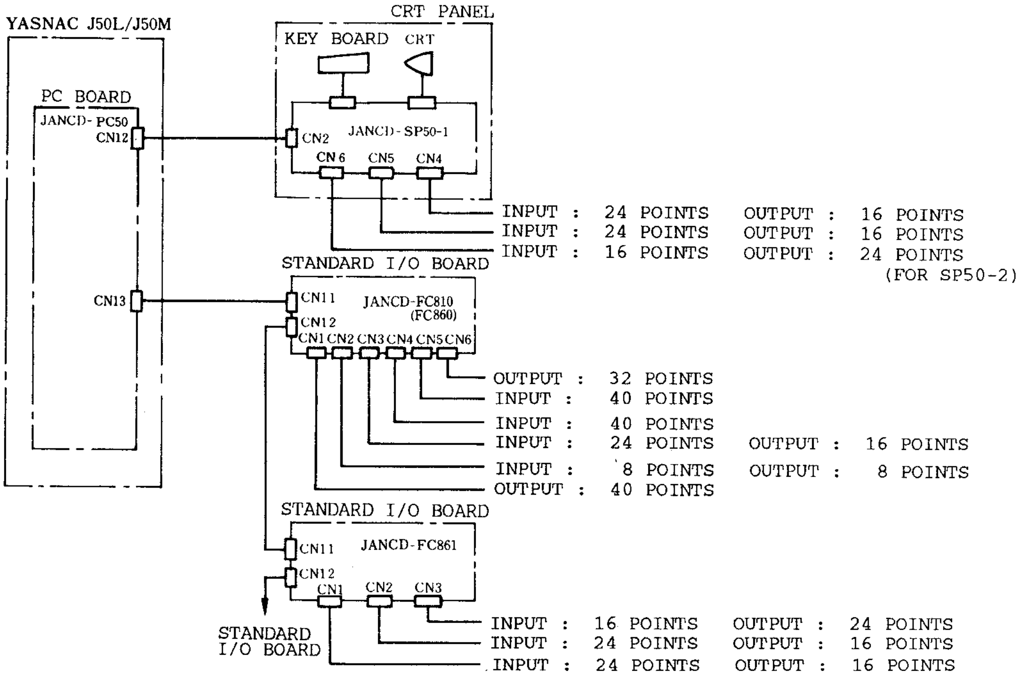
Type of Instruction	No. of Instructions
1. Relay instruction	13
2. Register instruction	37
3. Timer instruction	2
4. Control instruction	7
Total	59

2. Internal relays and registers are the same. Addresses used as internal relays cannot be used as internal relays.

3. Keep relays and keep memories are the same. Addresses used as keep relays cannot be used as keep memories. Addresses used as keep memories cannot be used as keep relays.

(3) I/O board location

5 I/O boards are shown below.



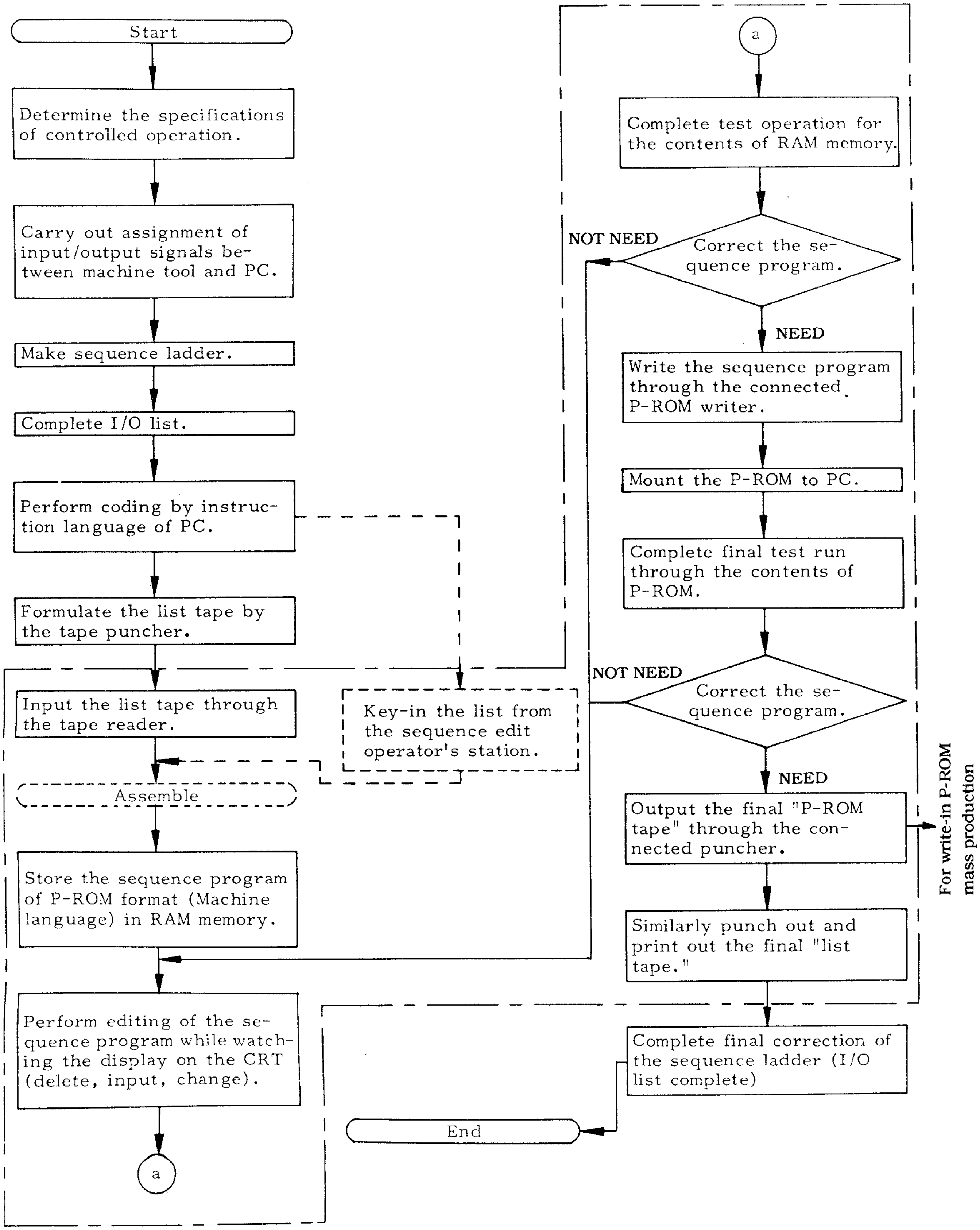
(4) Maximum number of each I/O board

	Maximum		
· Standard I/O board (FC810, FC860)	3	Input : Output :	336 points 288 points
· Standard I/O board (FC861)	7	Input : Output :	448 points 392 points
· CRT panel built-in I/O board (SP50-1)	1	Input : Output : (56 points for SP-2)	64 points 32 points

Notes :

- YASNAC J50 needs at least 1 of CRT panel built-in I/O board (SP50-1 or SP50-2). Therefore, a max. of 3 (max. input : 400 points, max. output : 344 points) for adding only FC810/FC860 and a max. of 7 (max. input : 512 points, max. output : 448 points) for adding only FC861 can be connected.
- Several I/O boards can exist at the same time, without exceeding the allowance I/O area No.
- The number of I/O boards can be expanded. Therefore, the last board needs to be terminal scanned.

4. PROCEDURES FOR SEQUENCE PROGRAM PREPARATION

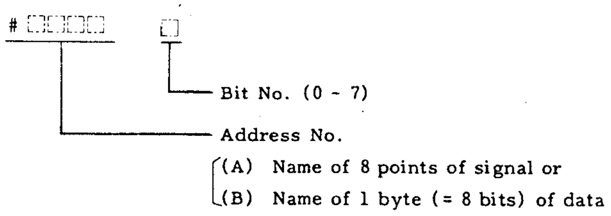

Start
Determine the specifications of controlled operation.
Carry out assignment of input/output signals between machine tool and PC.
Make sequence ladder.
Complete I/O list.
Perform coding by instruction language of PC.
Formulate the list tape by the tape puncher.
Input the list tape through the tape reader.
Key-in the list from the sequence edit operator's station.
Assemble
Store the sequence program of P-ROM format (Machine language) in RAM memory.
Perform editing of the sequence program while watching the display on the CRT (delete, input, change).
a
a
Complete test operation for the contents of RAM memory.
NOT NEED
Correct the sequence program.
NEED
Write the sequence program through the connected P-ROM writer.
Mount the P-ROM to PC.
Complete final test run through the contents of P-ROM.
NOT NEED
Correct the sequence program.
NEED
Output the final "P-ROM tape" through the connected puncher.
For write-in P-ROM mass production
Similarly punch out and print out the final "list tape."
Complete final correction of the sequence ladder (I/O list complete)
End

Note: The sections surrounded by [- - -] require the "sequence program editing device (JDU01)."

5. ADDRESS NUMBER AND ADDRESS MAP

5.1 ADDRESS NUMBER

In the preparation of the sequence program, the I/O signals of PC, internal relay, timer, battery backed-up memory, etc. of PC are all designated by address No. (4-digit number following mark #) and bit number (0 - 7 bit).



# □□□□ □
Bit No. (0 - 7)
Address No.
(A) Name of 8 points of signal or
(B) Name of 1 byte (= 8 bits) of data

(1) Designation of I/O Signals, Internal Relays, etc. (1 Bit Element)

As shown below, the elements which can be indicated by 1 bit information are designated by 5 digits (address no. and bit no.) preceded by the mark #.

Element	Name
1. I/O signal 2. Internal relay 3. Keep relay	# □□□□ □ Bit No. Address No.

In the case, the address No. takes the meaning of above (A) and it can be taken as the name given with respect to the 8 points of the signal.

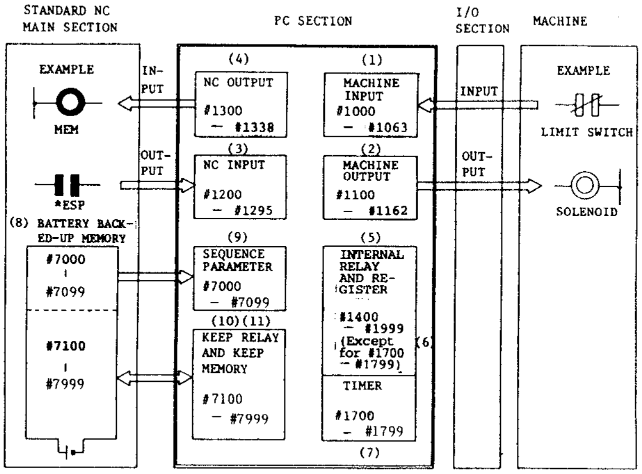
(2) Designation of Register, Timer, etc. (1 Byte Element)

The elements having 1 byte (= 8 bits) information, as shown below, are designated only by address number. In this case, the address number takes the meaning of above (B) and it can be taken as the name given with respect to 1 byte data.

Element	Name
4. Register 5. Timer 6. Sequencer parameter 7. Keep memory	# □□□□ Address No.

Note: Depending on the instruction, naming of 2 bytes #1500 and #1501 can be carried out through the address name #1500. Example: PUSH #1500

5.2 ADDRESS MAP AND DISPLAY SYMBOL



STANDARD NC MAIN SECTION
PC SECTION
I/O SECTION
MACHINE
EXAMPLE
MEM
*ESP
(8) BATTERY BACKED-UP MEMORY
#7000 I #7099
#7100 I #7999
IN-PUT
OUT-PUT
(4) NC OUTPUT #1300 - #1338
(3) NC INPUT #1200 - #1295
(9) SEQUENCE PARAMETER #7000 - #7099
(10)(11) KEEP RELAY AND KEEP MEMORY #7100 - #7999
(1) MACHINE INPUT #1000 - #1063
(2) MACHINE OUTPUT #1100 - #1162
(5) INTERNAL RELAY AND REGISTER #1400 - #1999 (Except for #1700 - #1799)
(6)
TIMER #1700 - #1799
(7)
INPUT
OUT-PUT
EXAMPLE
LIMIT SWITCH
SOLENOID

(1) Addresses of Input Signals from Machine (#1000 – #1063)

These are the address numbers + bit numbers (# □□□□□) for input signals like, push buttons, limit switch, etc. from the machine operation panel, machine controller, etc. This section should be determined by the machine tool builder.

(a) 1 bit of the address #1000 corresponds to 1 point of the input signal.

(b) The address number and the bit number are determined depending on the number of the pin and the number of the connector of the I/O board to which the input signal is connected.

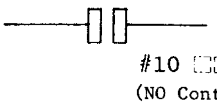
Example:

Bit No.	7	6	5	4	3	2	1	0
#1000	04 - 36	04 - 21	04 - 5	04 - 35	04 - 20	04 - 34	04 - 19	04 - 33

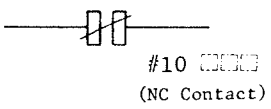
Pin No. 36
Connector No. 04
Input signal name (arbitrary) is registered

Refer to the I/O lists shown in Appendix 1, 2 for details.

(c) The input signals in the order of #1000-1999 are expressed by the following symbols.



#10 □□□
(NO Contact)



#10 □□□
(NC Contact)

5.2 ADDRESS MAP AND DISPLAY SYMBOL (Cont'd)

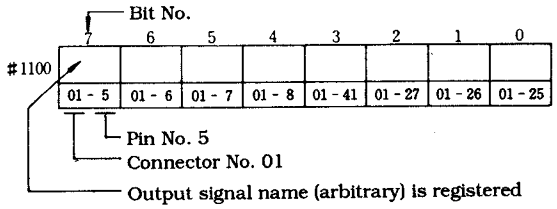
(2) Addresses of Output Signals to Machine (#1100 – #1162)

These are the address numbers + bit numbers (# □□□□□) of output signals like, lamp, solenoid, etc. from the machine operation panel, machine controller, etc. This section should also be decided through the machine tool builder.

(a) 1 bit of the address #1100 corresponds to 1 point of the output signal.

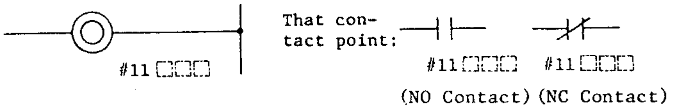
(b) The address number and the bit number are determined, depending on the number of the pin and the number of the connector of the I/O board to which the input signal is connected.

Example:



Refer to the I/O Lists shown in Appendix 1, 2 for details.

(c) The output signals in the order of #1100-#1199 are expressed by the following symbols.



(3) Addresses (#1200 - #1295) of Input Signals from NC Main Section

In other words, these can be termed as output signals to the PC from the NC main section. For example, the address numbers + bit numbers with respect to the M-BCD signals. These numbers in the order of #1200 are determined as standard signals and they can not be changed.

(a) 1 bit of addresses between #1200 and #1295 corresponds to 1 point of the input signal.

Example:

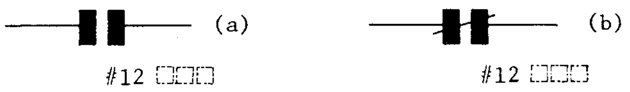
Bit No.

	7	6	5	4	3	2	1	0
#1200	M28	M24	M22	M21	M18	M14	M12	M11

M function BCD output

Refer to "Appendix : I/O list" for details. However, they differ for YASNAC J50L (for lathes) and YASNAC J50M (for machining centers). So, refer to the corresponding list.

(b) The input signals in the order of #1200 - #1295 are expressed by the following symbols.



(4) Addresses (#1300 – #1338) of Output Signals from NC Main Section

In other words, these can be termed as input signals to NC main section from the PC. For example, the address numbers and the bit numbers with respect to the EDIT and MEM (memory operation) selection.

The numbers between 1300 and 1329 are determined as standard signals and' they can not be changed.

(a) 1 bit of the addresses between #1300-#1329 corresponds to 1 point of the input signal.

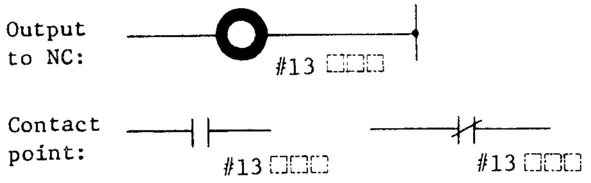
Example:

Bit No.

	7	6	5	4	3	2	1	0
#1300	EDT	MEM	D	T		H/S	J	RT
	EDIT	MEMORY	MANUAL	TAPE		HANDLE/ STEP	JOG	RAPID

Refer to "Appendix : I/O list" for details. However, they differ for YASNAC J50L and YASNAC J50M. So, refer to the corresponding list.

(b) The output signals between #1300 and #1329 are expressed by the following symbols.

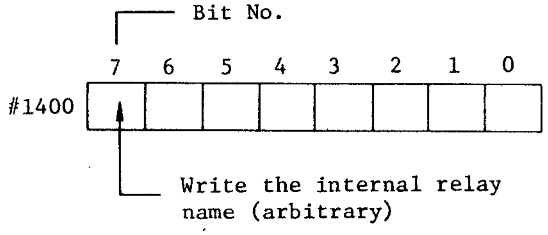


(5) Addresses (#1400 - #1999 except for #1700 - #1799) for Internal Relays

These are the address numbers and bit numbers with respect to the internal relays which can only be used inside the PC while preparing the sequence program.

(a) 1 bit of the addresses between #1400 - #1492 corresponds to 1 internal relay, for example.

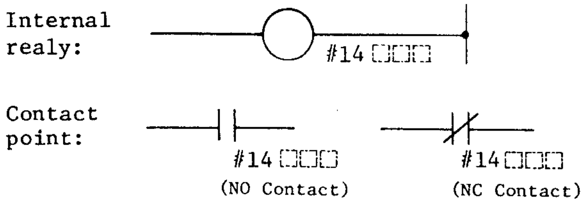
I/O list example:



(b) The number of usable internal relays are as follows.

500 bytes × 8 bits = 4000 relays

(c) The internal relay and its contact point are expressed by the following symbol.



There is no limit for NO and NC contact points until the program memory capacity is exceeded.

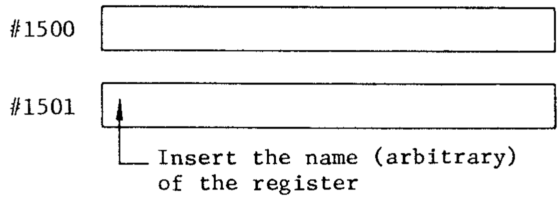
(d) Adressed used in register cannot be used as internal relay.

(6) Addresses (#1400 - #1999 except for #1700 - #1799) of Register

These are the address numbers with respect to the 1 byte (= 8 bits) register for general purpose use. These registers are used for register instruction or for the working addresses of macro instructions.

(a) 1 address number corresponds to 1 register of 1 byte.

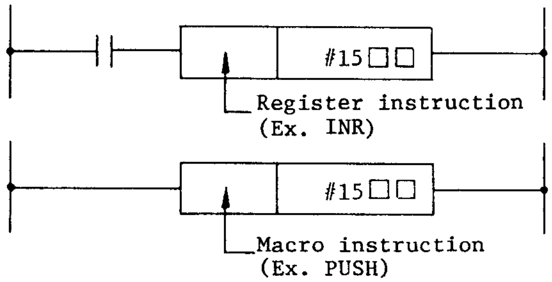
I/O list example:



(b) Number of usable registers are as follows:

500 registers from #1400 to #1999 except for #1700 to #1799.

(c) In a register, the address itself is the expression symbol. The following shows two examples of the symbols.



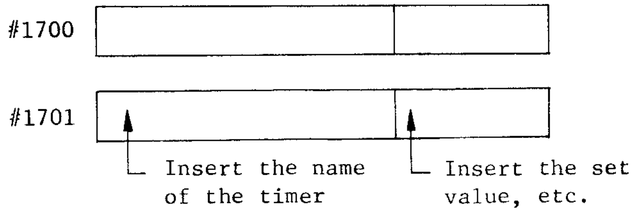
(d) Addresses used in internal relay cannot be used as register.

(7) Addresses of Timer (#1700 - #1799)

These are the addresses with respect to the timers. They are used in the instruction of timers.

(a) 1 address number corresponds to 1 timer.

I/O list example:



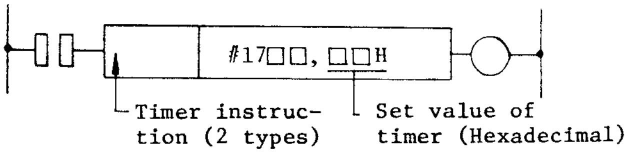
(b) The time unit and the number of usable timers are shown in the following table.

Address No.	No. of timers	Time unit
#1700-#1709, #1760-#1769	20	1 = 8 msec
#1710-#1729, #1790-#1799	30	1 = 100 msec
#1730-#1749, #1780-#1789	30	1 = 50 msec
#1750-#1759	10	1 = 1 sec
#1770-#1773	4	1 = 1 min

The range of set values is 0 - 255.
(0 - 127 for variable timer.)

(c) The symbol example of timers is given below.

Example:



5.2 ADDRESS MAP AND DISPLAY SYMBOL
(Cont'd)

(8) Battery Backed-up Memory (#7000 - #7999)

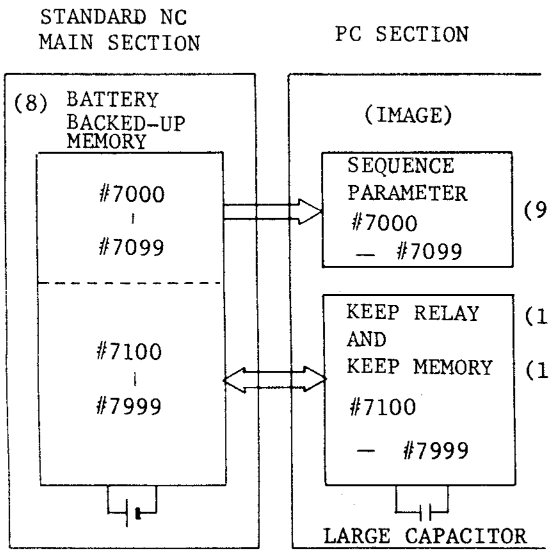
(a) The above addresses of #7000 to #7295 are differentiated from others by the name "battery backed-up memory." That means, the data of #7000 to #7295 are preserved in the battery back-up memory in the standard NC main section. So, even if the power supply is turned off, the data are not erased.

(b) The sequence program of PC unit can only handle image data of the PC unit. The original data from NC main section can not be handled (reading or writing).

(c) Following 3 types of battery backed-up memory data are available.

Sequencer parameter: #7000 - #7099

Keep relay: }
Keep memory: } #7100 - #7999



(d) Transfer to sequencer parameter data to PC

In addition to the power supply turning on, the sequencer parameter data is transferred to PC from the NC main unit under the following conditions. Through the parameter writing operation, even if a single sequencer parameter data is modified, then all the sequencer parameter data are transferred. Consequently, all the image data of the PC are always latest data. The sequencer parameter data can only be read in the sequence program and they must not be modified.

(e) Transfer of keep relay and keep memory data to NC.

The image data of the PC unit keep relay and keep memory are sometimes read and written, so they are changed in the sequence program. Consequently, it becomes necessary to preserve the latest image data of the PC unit by transferring them to the battery backed-up memory as latest original data. And this procedure is explained below.

Automatic data transfer

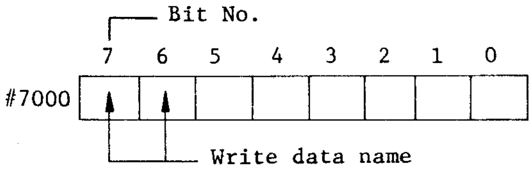
When the power supply of the unit is kept turned on, the data of #7100 - #7999 get transferred from PC to NC unit.

(9) Addresses (#7000 - #7099) of Sequencer Parameter

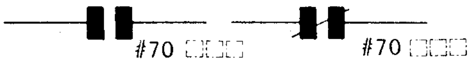
These are the address numbers corresponding to the parameter of the sequencer. The data of #7000 - #7099 can be changed through the normal writing operation. These data can be used in a sequence program in the following two procedures: ⓐ Using as 1 bit data and ⓑ Using as 1 byte data.

(a) Using as 1 bit data

I/O list example:



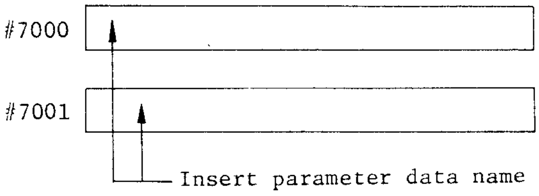
Symbol expression is carried out in the following way.



Data "1" = Closed   Data "1" = Open
Data "0" = Open   Data "0" = Closed

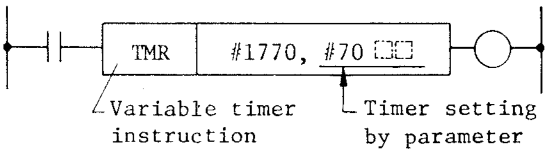
(b) Using as 1 byte data

I/O list example:



The symbol expression is the address number. The example of using in timer is shown in the following figure.

Example:

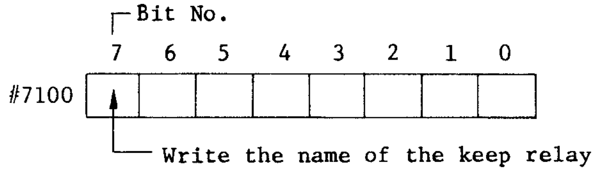


(10) Addresses (#7100 - #7999) of Keep Relay

These are the address numbers and bit numbers of the keep relays used in the PC.

(a) 1 bit of #7100 - #7999 corresponds to 1 keep relay.

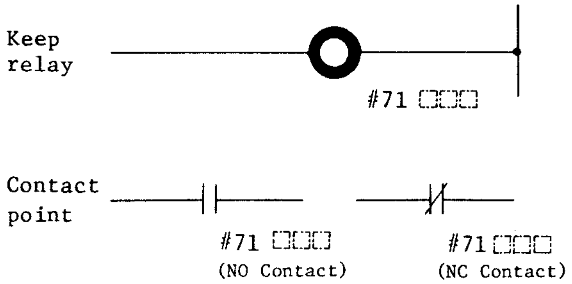
I/O list example:



(b) The number of usable keep relays is as follows.

900 bytes × 8 bits = 7200

(c) The keep relays and their contact points are
expressed by the following symbols.

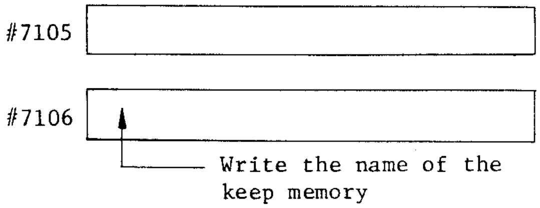


(11) Addresses (#7100 - #7999)

These are the addresses corresponding to the 1 byte memory which can be preserved even after turning off the power supply. If the performance is limited only to the preservation of data, the keep memory can be used in the same way as that of a register. Consequently, the keep memory can also be used as an object of register instruction or as supplementary data of macro instruction. Especially, when preparing a sequence program for memory random type ATC, this keep memory becomes necessary.

(a) 1 address number beyond #7100 corresponds to one keep memory of 1 byte (8 bits).

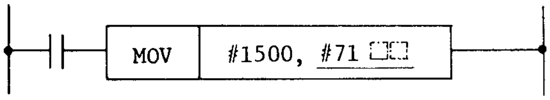
I/O list example:



(b) The number of usable keep memory is as follows:

900 memories from #7100 to #7999

(c) The address number itself stands for the symbol of the keep memory.



MOV: Transfer the contents of register #1500 to keep memory #71□□

(12) Writing Initial Values of Keep Relays and Keep Memories

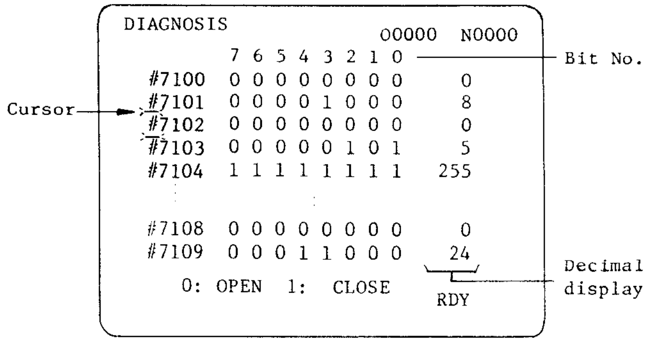
When preparing a sequence program by using the keep relays and keep memories, it becomes necessary to set the initial values prior to the execution.

(a) Set the system number switch of NC unit at "1" and then turn on the power supply.

(b) Depress the $\boxed{\text{DGN}}$ function key.

Input/output signal ON/OFF state will be displayed on the CRT screen.

(c) After keying-in in the order of $\boxed{7}$ $\boxed{1}$ $\boxed{0}$ $\boxed{1}$, if the cursor key is depressed, then the following display will be obtained.



```
DIAGNOSIS                 00000  N0000
            7 6 5 4 3 2 1 0
    #7100   0 0 0 0 0 0 0 0      0
    #7101   0 0 0 0 1 0 0 0      8
    #7102   0 0 0 0 0 0 0 0      0
    #7103   0 0 0 0 0 1 0 1      5
    #7104   1 1 1 1 1 1 1 1    255
      :           :

    #7108   0 0 0 0 0 0 0 0      0
    #7109   0 0 0 1 1 0 0 0     24
     0: OPEN   1:  CLOSE        RDY
```

5.2 ADDRESS MAP AND DISPLAY SYMBOL
(Cont'd)

(d) Adjust addresses #7105 to #7294 for initial condition setting by depressing the cursor.

(e) If the INSRT (insert) key is depressed, the cursor will move in the right hand direction, and will move to the 7th bit position of the address.

(f) Keep on pressing the cursor key until it becomes adjusted to the position of the decimal display.

(g) Key-in the desired values (0 - 255) for setting initial condition and then depress the WR key. The decimal display will get changed to the presently keyed in value.

(h) If the INSRT key is depressed, the cursor will move to the left hand position #. Thereby, the setting of one address number is completed.

(i) Repeat steps (d) to (h) to write all the desired initial values of the address numbers.

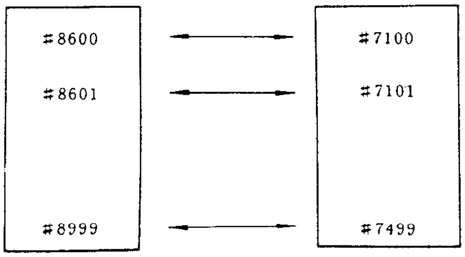
(j) Adjust the system number switch to "0."

Note: If a particular bit is desired to be changed 0 ⇄ 1, carry out following operations after the operation of item 5). Depress the cursor key and adjust the cursor to the bit desired to be changed, then depress WR key.
0 ⇄ 1 change will be obtained.
1 ⇄ 0 change will be obtained if the WR key is depressed again.

(13) Writing of Keep Relay Numerical Input (Optional only for J50M)

Writing to keep memory (#7100 - #7999) can be normally executed from 0 to 255, however, 4-digit writing is also possible with numbers #8600 - #8999. #7100 - #7499 and #8600 - #8999 correspond to each other as shown in the figure below. #7101 is altered by writing and alteration of #8601.

Note: When keep memory is referred from sequence, use #7100 - #7499, not #8600 - #8999.



(a) Keep memory display

Following displays are added to existing #7100 - #7499 display:
Depress function key DGN .
Key-in 8 , 6 , 0 , 1 and depress cursor .
CRT screen has display as shown in either Fig. (i) or (ii).
[Hereafter Fig. (i) is to be called 2-digit display, while Fig. (ii) is to be called 4-digit display.]

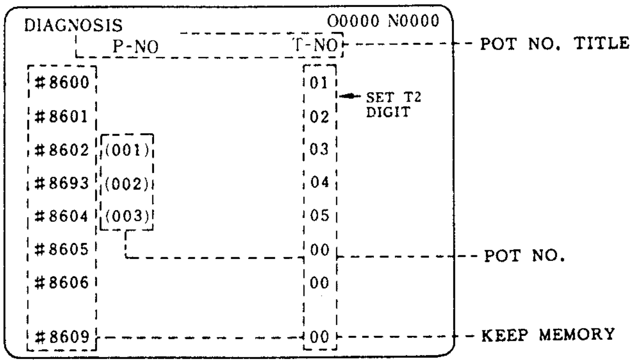


Fig. (i) #6022 D2=0 #6355=8602 #6356=8604

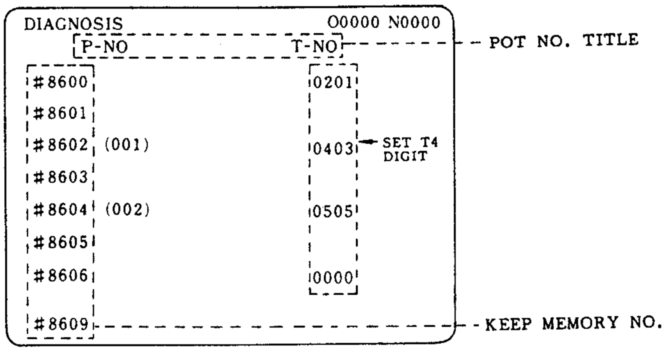


Fig. (ii) #6022 D2=1 #6355=8602 #6356=8604

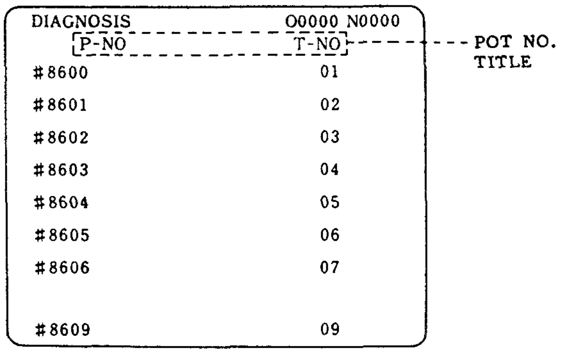
For Fig. (ii), even and uneven number keep memories are used in pairs, 0 to 9999 are available by expressing the higher 2 digits of the decimal 4 digits with even No. keep memory, and lower 2 digits with uneven No. keep memory.

Pot No. display [Figs. (i), (ii)]
When the max. and min. keep memory numbers are set to parameters #6355 and #6356, Figs. (i) and (ii) show how #6355 and #6356 are set for #7402 and #8604, respectively.

(b) Writing to keep memory
Turn system No. switch to "1".
Use page cursor keys and to move the cursor to keep memory No. to be changed. Input new figure and depress WR key. Procedure mentioned above enables #8600 - #8999 range data to be changed and set.

Notes:

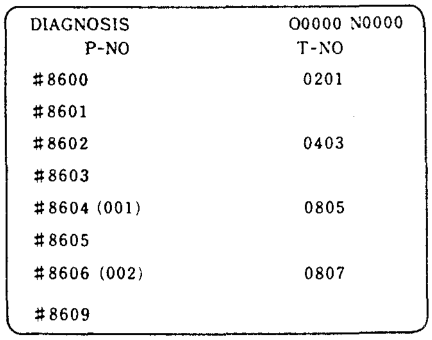
- The same memory is used for #8600 - #8999 and #7100 - #7499: if a value of #8602 is changed, that of #7102 is changed to the same value.
- When the display can be extended up to 9999, as in Fig. (ii), the even number keep memory data are changed to one lower number and cursor moves there by writing when the cursor is at an uneven keep memory number.
- If #6355 and #6356 are set conversally, pot No. title and pot No. are not normally displayed. However, if #6355 and #6356 have keep memory No. on the same page, pot No. title is displayed. [Refer to Fig. (iii).]
- If uneven number is set by mistake for #6355 when 4-digit display (#6022 D2=1), pot No. is displayed from the even number keep memory No. which is one number higher than the pot No.



```
DIAGNOSIS                    00000 N0000
    P-NO                        T-NO
#8600                           01
#8601                           02
#8602                           03
#8603                           04
#8604                           05
#8605                           06
#8606                           07

#8609                           09
```

POT NO. TITLE

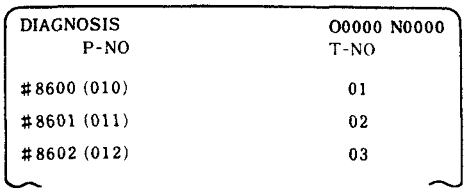
Fig. (iii) #6022 D2=0 #6355=8604 #6356=8602



```
DIAGNOSIS                    00000 N0000
    P-NO                       T-NO
#8600                          0201
#8601
#8602                          0403
#8603
#8604 (001)                    0805
#8605
#8606 (002)                    0807

#8609
```

Fig. (iv) #6022 D2=1 #6355=8603 #6356=8606

- If a number lower than that for #6355 is set for #8600, pot number from #8600 is lower than the number already set to display. [Refer to Fig. (v).]



```
DIAGNOSIS                    00000 N0000
      P-NO                    T-NO
#8600 (010)                    01
#8601 (011)                    02
#8602 (012)                    03
```

Fig. (v) #6022 D2=0 #6355=7391

- When pot number is not displayed, set 0 for #6355 and #6356.
- In 2-digit display (#6022 D2=0), writing-in more than a 3-digit number is not accepted.

(14) Address Setting of I/O Board

I/O board has a rotary switch for address setting. For rotary switch and address, refer to the table below.

(a) Standard I/O Board

FC861			FC810/FC860		
I/O Area No.	Input	Output	I/O Area No.	Input	Output
1-1	#1000 to #1007	#1100 to #1106	1	#1000 to #1013	#1100 to #1111
1-2	#1008 to #1015	#1108 to #1114			
2-1	#1016 to #1023	#1116 to #1122	2	#1016 to #1029	#1116 to #1127
2-2	#1024 to #1031	#1124 to #1130			
3-1	#1032 to #1039	#1132 to #1138	3	#1032 to #1045	#1132 to #1143
3-2	#1040 to #1047	#1140 to #1146			
4-1	#1048 to #1055	#1148 to #1154	4	#1048 to #1061	#1148 to #1159
4-2	#1056 to #1063	#1156 to #1162			

For rotary switch (SW1) setting and I/O area No., refer to the table below.

SW1	I/O Area No.	
	FC861	FC810/FC860
0	No setting	No setting
1	1-1	1
2	1-2	No setting
3	2-1	2
4	2-2	No setting
5	3-1	3
6	3-2	No setting
7	4-1	4
8	4-2	No setting
9 to F	No setting	

5.2 ADDRESS MAP AND DISPLAY SYMBOL

(Cont'd)

(b) CRT Panel Built-in I/O Board

	SP50	
I/O Area No.	Input	Output
1-1	#1000 to #1007	#1100 to #1103
1-2	#1008 to #1015	#1108 to #1111
2-1	#1016 to #1023	#1116 to #1119
2-2	#1024 to #1031	#1124 to #1127
3-1	#1032 to #1039	#1132 to #1135
3-2	#1040 to #1047	#1140 to #1143
4-1	#1048 to #1055	#1148 to #1151
4-2	#1056 to #1063	#1156 to #1159

For rotary switch (SW1) setting and I/O area No., refer to the table below.

SW1	I/O Area No.
	SP50
0	No setting
1	1-1
2	1-2
3	2-1
4	2-2
5	3-1
6	3-2
7	4-1
8	4-2
9 to F	No setting

5.3 I/O LIST AND SEQUENCE LADDER

The data list of the address map is called the I/O lists. The I/O lists for J50L (for lathes) J50M (for machining centers) are shown in the Appendixes at the end of this manual.

(1) For preparing the sequence ladder, first of all, carry out the assignment of the I/O signals (#1000 and #1100) between the PC and the machine tool.

(2) After the completion of the assignment of the I/O signals, refer to the I/O list as a list for data and freely prepare sequence ladder through the command symbols of the PC. In this case, it is convenient to use the abbreviated names like SW7, SOL A, etc. for element names.

(3) Complete the assignment of the address numbers for each element: internal relay, register, timer, etc. for the completed and checked sequence ladder. Thereby, the complete sequence ladder and a complete I/O list is obtained.

6. SEQUENCE CONTROL METHOD

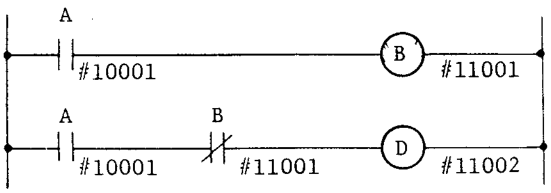
Sequence control through the PC is carried out successively through the software, so the operations are quite different from that of the simultaneous processing in the case of normal relay circuit. So, it is necessary to have clear understanding of this point prior to programming.

6.1 DIFFERENCES IN OPERATION

Relay sequence: Each element is simultaneously processed with regard to time.

PC sequence: Each element is successively processed. The ladder is repeatedly processed at a constant period. This period is called scanning time. (Scanning time Ex.: 8 msec × n times)

Example:



The above PC sequence ladder is operated in the following sequence. Simultaneous processing is never carried out.

① Condition of contact point A is read.

② This is output to internal relay B as it is.

③ Condition of contact point A is read.

④ AND logic is taken from the NC contact point of relay B.

⑤ The result is output to internal relay D.

Due to this successive processing, the internal relay D is not turned on. On the other hand, if the above ladder is executed by the relay sequence, the relay D is turned on for a moment and thereby one shot operation is being carried out. As discussed above, it should always be remembered that the processing in the PC is carried out successively and then programming should be completed. For reference, if the above mentioned PC sequence ladder is coded according to PC command words, it takes the following form.

```
      :
①  LD #10001
②  OUT #11001
③  LD #10001
④  AND NOT #11001
⑤  OUT #11002
      :
```

Example of coded sequence program (called list)

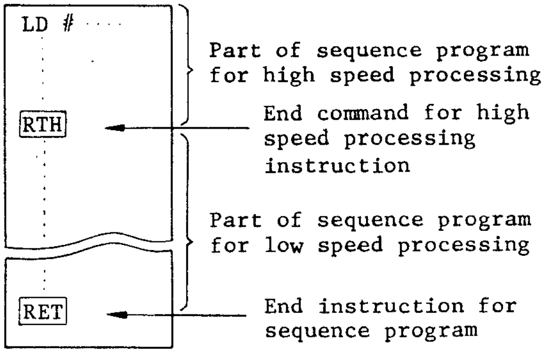
6.2 SCANNING TIME (PROCESSING TIME)

The execution time from the start to the end of a sequence program is called the scanning time. The scanning time for this PC is as follows.

High speed scanning time: 8 msec

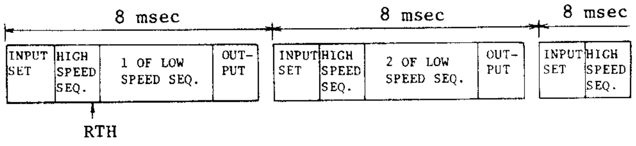
Low speed scanning time: 8 msec × n

That means, in this PC, the sequence program can be processed by dividing it into the high speed processing part and the low speed processing part. In this case, write the program as follows.


LD # ····
RTH
RET
Part of sequence program for high speed processing
End command for high speed processing instruction
Part of sequence program for low speed processing
End instruction for sequence program

The first part of the write sequence program needs high speed processing.

(1) Relationship between High Speed Processing and Low Speed Processing



6.2 SCANNING TIME (PROCESSING TIME) (Cont'd)

(a) From the beginning of the sequence to the RTH command, the high speed sequence program (high speed Seq.), as shown in the above figure, is surely executed once within 8 msec. During the execution of this high speed sequence, the input condition does not change.

(b) The low speed sequence program (low speed Seq.) after RTH command is divided into "n" items and one of them is executed in the remaining time of 8 msec. That means, the whole low speed sequence program is executed in 8 msec × "n" times time. Consequently, the value of "n" depends on the capacity of the whole program and the length of the high speed sequence program. Since the low speed program is divided into many parts, so the I/O condition changes in the middle. So, be sure to take NOTE of item 3 of this section.

(c) At the first part of the 8 msec section, all the input conditions (#1000 and #1200) are taken in the PC at a time.

(d) At the last part of 8 msec section, all the output conditions (#1100 and #1300) are output at a time.

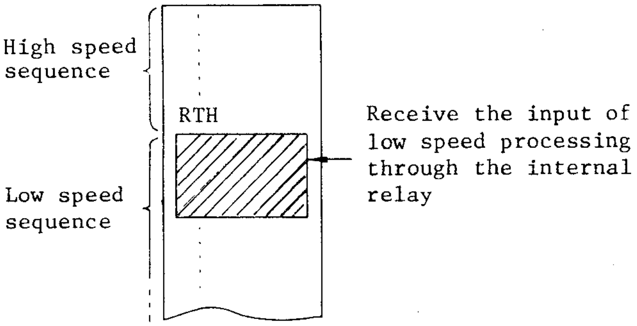
(2) Precautions for High Speed Processing Sequence Program

In this program, only the portion where high speed responses such as counting of ON/OFF are necessary, is handled. So limit it to the least possible size of the sequence program. Limit it within 100 steps when converted into contact point instruction.

(3) Precautions for Low Speed Processing Sequence Program

(a) The scanning time for low speed processing differs depending on the capacity of the total sequence program (8 msec × "n"). (The amount of program that can be executed within 8 msec is approximately 3000 steps when converted into contact point instruction. However, this amount of steps is the combination of high speed and low speed processings.)

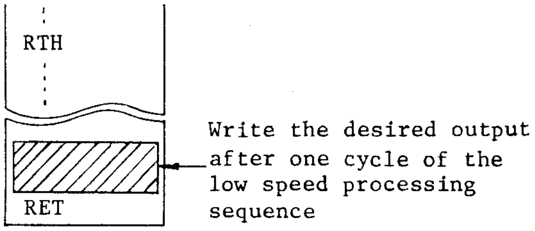
(b) Since division processing is carried out during the execution of the low speed processing sequence program, the input condition changes. Consequently, all inputs to be used through the low speed processing sequence program need to be received through the internal relays at the top of the low speed processing sequence program. Then, use the contact point of the receiving relay in place of the input.


High speed sequence
RTH
Low speed sequence
Receive the input of low speed processing through the internal relay

Through the above operations, the input conditions may be kept unchanged during 1 cycle of execution of the low speed processing sequence program.

(c) If the output of the high speed processing sequence program is to be used in the low speed processing sequence program, the processing like (b) needs to be carried out.

(d) The output signals which are not desired to be output until the end of the execution of low speed processing sequence program, once received outputs them through the internal relays without outputting them to the addresses of output of the PC unit. Then, do not connect the same to the address of the external output at the tail of the low speed processing sequence program.


RTH
RET
Write the desired output after one cycle of the low speed processing sequence

6.3 MEMORY CAPACITY OF SEQUENCE PROGRAM

The sequence program is finally written to the EPROM (Erasable Program Rom) and then used.
The capacity of the program memory of this PC can be used according to the following distribution.

Division	No. of Bytes	Step Conversion	No. of PROMs	PROM Location on PC Board JANCD-CP50
1	32 k bytes	Approx. 8000 steps	1	30

(Usually, relay instruction is of 3-7 bytes and other commands are of 1-25 bytes range.) For the memory storing the sequence program of 16K bytes, 4000 steps (16K/4 = 4K (4000 steps) is required, if approximately 4 bytes is used for one step.

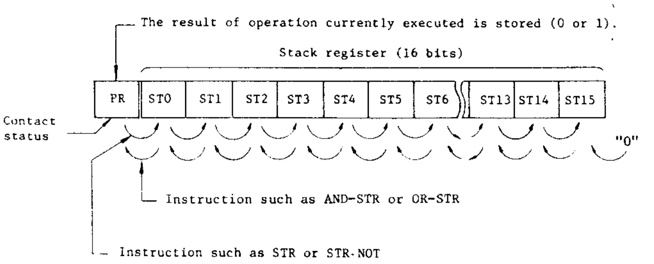
7. PC INSTRUCTIONS

This chapter explains the 61 type basic instructions and 11 type macro instructions that can be used with this PC while describing their functions, display symbols and coded lists.

7.1 PRELIMINARY KNOWLEDGE

(Registers to store intermediate results during logical operation)

(1) PC is provided with a register to store intermediate results of logical operation of sequence programs, and it consists of 1 bit + 16 bits, as shown below.


The result of operation currently executed is stored (0 or 1).
Stack register (16 bits)
PR
ST0
ST1
ST2
ST3
ST4
ST5
ST6
ST13
ST14
ST15
Contact status
"0"
Instruction such as AND-STR or OR-STR
Instruction such as STR or STR-NOT

(2) RR (Result Register)

1-bit register to which the result of operation currently executed is stored. The contact status (0 or 1) can be set into RR by the LD instruction or the RR contents can be output to the relay address by the OUT instruction. Also, 1-bit shift of the stack register contents to RR (after operation) by the STR or AND-STR instruction is possible.

(3) Stack Register (Stack, ST0 - ST15)

Intermediate operation resulting from long logical operation can be saved into the stack register sequentially up to 16 bits.

Data in RR is shifted to ST0 by the STR or STR-NOT instruction, and data in the stack register is shifted by 1 bit toward right. Also data in ST0 and RR is operated by the AND-STR or OR-STR instruction, set into RR, and data in the stack register is shifted by 1 bit toward left. ST15 is cleared to "0." If the number of STR or STR-NOT instructions does not equal to the number of AND-STR or OR-STR instructions used in a series of long logical operations until the final result is obtained, it results in an error. In other words, the number of times that data is saved in the stack and the number of times that data is fetched out must be equal.

7.2 TYPES OF INSTRUCTIONS AND LISTS

(1) Instruction Types

There are the following types in the instructions used with PC.

Basic instructions (61 types)

① Instructions for relay:	13 types
② Instructions for registers:	37 types
③ Instructions for timers:	2 types
④ Control instructions:	7 types
Total	59 types

Macro instructions

(1) Macro instructions:	9 types
(2) Auxiliary instructions:	4 types

7.2 TYPES OF INSTRUCTIONS AND LISTS (Cont'd)

(2) List of instructions for relay

No.	Instruction	*	Meaning	RR after operation	Page
1	LD	1	Reads signal status (0 or 1) and sets it to RR.	↕	19
2	LD-NOT	1	Reads inversion signal status and sets it to RR.	↕	19
3	AND	1	Sets AND of contact and RR to RR (AND).	↕	20
4	AND-NOT	1	Sets AND of inversion signal and RR to RR (Reverse AND)	↕	20
5	OR	1	Sets OR of signal and RR to RR (OR).	↕	20
6	OR-NOT	1	Sets OR of inversion signal and RR to RR (Reverse OR).	↕	20
7	XOR	1	Sets uncoincidence between signal and RR to RR.	↕	20
8	XNR	1	Sets coincidence between signal and RR to RR.	↕	20
9	STR	1	Loads RR contents to stack and executes LD instruction.	↕	21
10	STR-NOT	1	Loads RR contents to stack and executes LD NOT instruction.	↕	21
11	AND-STR	1	Sets AND of RR and stack to RR.	↕	21
12	OR-STR	1	Sets OR of RR and stack to RR.	↕	21
13	OUT	1	Writes operation results (RR) to relay (address).	—	21

Note:

1. The * column shows the execution time converted to the contact instruction (1 = One contact instruction)
2. The ↕ mark shows that the RR contents change after instructions are operated. The — mark shows that no change occurs.

(3) List of Instructions for Timers

No.	Instruction	*	Meaning	RR after operation	Page
1	TIM	10	Timer processing (Fixed timer)	time up = 1	22
2	TMR	10	Timer processing (Variable timer)	time up = 1	22

(4) List of Instructions for Registers

No.	Instruction	*	Meaning	RR after operation	Page
1	INR	3	Adds + 1 to register contents.	—	22
2	DCR	3	Adds - 1 to register contents.	—	22
3	CLR	2	Clears the register contents.	—	23
4	CMR	3	Inverts the register contents.	—	23
5	ADI	3	Addition of register contents and numeric.	—	23
6	SBI	3	Subtraction of register contents and numeric.	—	23
7	ANI	3	AND of register contents and numeric.	—	24
8	ORI	3	OR of register contents and numeric.	—	24
9	XRI	3	XOR of register contents and numeric.	—	24
10	DEC	3	Coincidence of register contents and numeric.	↕	24
11	COI	4	Coincidence of register contents and numeric.	↕	24
12	CMP	3	Comparison of register contents and numeric.	↕	25
13	CPI	4	Comparison of register contents and numeric.	↕	25
14	MVI	3	Load numeric to a register.	—	25
15	ADD	4	Adds registers R1 and R2 and stores the result in R2.	—	25
16	SUB	4	Subtracts R1 from R2 and stores the result in R2.	—	25
17	ANR	4	Takes AND of R1 and R2 and stores the result in R2.	—	25
18	ORR	4	Takes OR of R1 and R2 and stores the result in R2.	—	25
19	XRR	4	Takes XOR of R1 and R2 and stores the result in R2.	—	26
20	CPR	5	Checks the result of comparison of R1 with R2, and stores the result in R2.	↕	26
21	COR	5	Checks coincidence between R1 and R2, and sets the result in RR.	↕	26
22	MOV	4	Transfers R1 contents to R2.	—	26
23	DST	5	Transfers AND of R1 contents and numeric to R2.	—	26
24	DIN	7	Data extraction	—	27
25	ADC	4	Double length addition	↕	27

7.2 TYPES OF INSTRUCTIONS AND LISTS (Cont'd)

No.	Instruction	*	Meaning	RR after operation	Page
26	ADDW	4	Adds double length registers (WR2 and WR1) and stores the result in WR2.	—	27
27	SUBW	4	Subtracts WR1 from WR2 and stores the result in WR2.	—	28
28	MULW	10	Multiplies double length register (WR2) with register (R1) and stores the result in WR2.	RR is set to "1" when overflow occurs.	28
29	DIVW	15	Divides double length register (WR2) by register (R1) and stores the result in WR2.	○	28
30	INRW	3	Adds + 1 to double length register contents.	—	29
31	DCRW	3	Adds - 1 to double length register contents.	—	29
32	CLRW	3	Clears double length register contents.	—	29
33	CMRW	2	Inverts double length register contents.	—	29
34	CORW	3	Sets coincidence result of double length registers (WR2 and WR1) to RR.	↕	29
35	CPRW	3	Sets comparison result of double length registers (WR2 and WR1) to RR.	↕	29
36	MVIW	3	Loads numeric to double length register.	—	29
37	DSTW	5	Transfers AND of double length register (WR1) contents and numeric to double length register (WR2).	—	30

(5) List of Control Instructions

No.	Instruction	*	Meaning	RR after operation	Page
1	NOP	1	No-operation.	—	30
2	MCR	1	Start of master control relay.	—	30
3	END	1	End of master control relay.	—	30
4	RET	1	Sequence program termination.	—	30
5	RTI	1	RR is set to "1" and RET instruction is executed.	—	30
6	SET	1	Sets RR to "1."	1	31
7	RTH	1	High speed processing sequence program termination.	—	31

(6) List of Macro Instructions

No.	Instruction	*	Meaning	RR after Operation	Page
1	SUBP 005	Approx. 100	Counter.	↕	31
2	SUBP 006		Rotation (for control of rotating object).	↕	33
3	SUBP 007		Code conversion.	↕	35
4	SUBP 009		Pattern clear.	↕	36
5	SUBP 011		Parity check.	↕	37
6	SUBP 014		Data conversion (Binary ⇄ BCD).	↕	37
7	SUBP 017		Data search.	↕	38
8	SUBP 018		Index data move.	↕	38
9	SUBP 023		Message display (Option).	↕	39

(7) List of Auxiliary Macro Instructions

Instruction	*	Meaning	RR after Operation	Page
IPSH	2	Designation of numeric used by SUBP.	—	31
APSH	2	Designation of address of register used by SUBP.	—	31
PUSH	2	Designation of address of register used by SUBP.	—	31
TPSH	2	Designation of Table No. of PC table used by SUBP.	—	31

7.3 INSTRUCTIONS FOR RELAYS

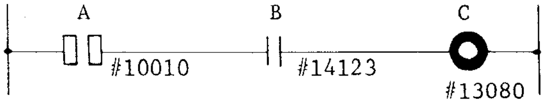
(1) LD (Load) RR after operation{RR↕}

① Format LD # x x x x x
↑
Internal signal name

Example: #10100
#14312

② Reads contact status (1 or 0) and sets the results to RR.

③ Normally this instruction is applied to Contact A (—| |—)



```
LD    #10010
AND   #14123
OUT   #13080
```

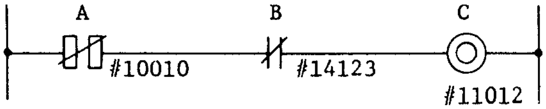
(2) LD-NOT (Load Not) {RR↕}

① Format LD-NOT # x x x x x
↑
Internal signal name

Example: #10100
#14321

② Read inversion contact status (1 or 0) and sets the result to RR.

③ Normally this instruction is applied to Contact B (—|/|—).



```
LD-NOT    #10010
AND-NOT   #14123
OUT       #11012
```

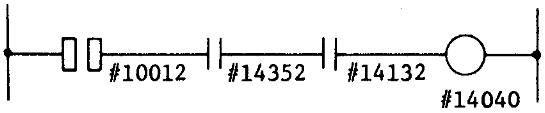
7.3 INSTRUCTIONS FOR RELAYS (Cont'd)

(3) AND {RR↕}

① Format AND # x x x x x

Internal signal name

② Takes AND of contact and RR and loads the result to RR (AND).



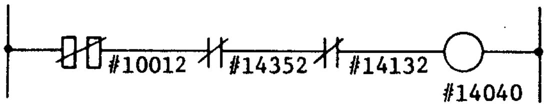
```
LD     #10012
AND    #14352
AND    #14132
OUT    #14040
```

(4) AND-NOT {RR↕}

① Format AND-NOT # x x x x x

Internal signal name

② Takes AND of inversion contact and RR and loads the result to RR (Reverse AND).



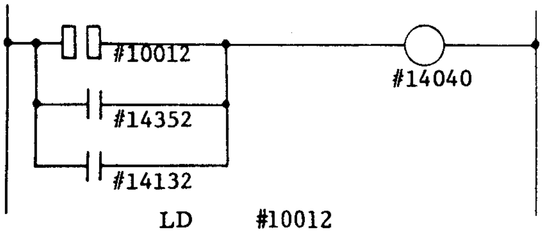
```
LD-NOT   #10012
AND-NOT  #14352
AND-NOT  #14132
OUT      #14040
```

(5) OR {RR↕}

① Format OR # x x x x x

Internal signal name

② Takes OR of contact point and RR and loads the result to RR (OR).



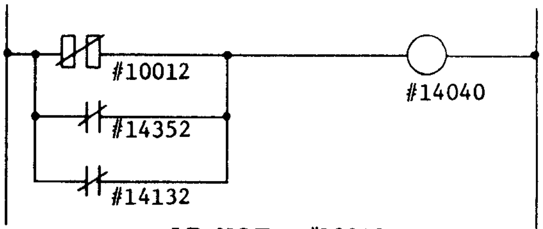
```
LD     #10012
OR     #14352
OR     #14132
OUT    #14040
```

(6) OR-NOT {RR↕}

① Format OR-NOT # x x x x x

Internal signal name

② Taken OR of inversion contact point and RR and loads the result to RR (Reverse OR).



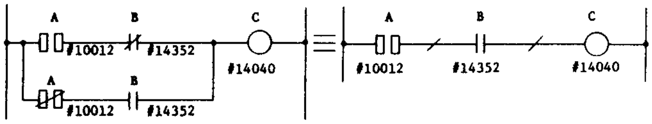
```
LD-NOT   #10012
OR-NOT   #14352
OR-NOT   #14132
OUT      #14040
```

(7) XOR (Exclusive OR) {RR↕}

① Format XOR # x x x x x

Internal signal name

② Loads dissidence between contact and RR to RR.



```
LD       #10012 ⎫     ⎧ LD    #10012
AND-NOT  #14352 ⎬  ≡  ⎨ XOR   #14352
STR-NOT  #10012 ⎪     ⎩ OUT   #14040
AND      #14352 ⎪
OR-STR          ⎪
OUT      #14040 ⎭
```

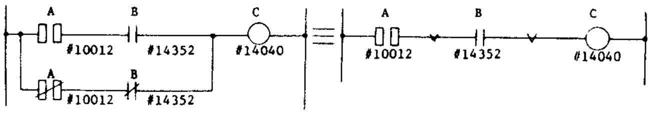
A	B	C
0	0	0
1	0	1
0	1	1
1	1	0

(8) XNR (Exclusive NR) {RR↕}

① Format XNR # x x x x x

Internal signal name

② Loads coincidence between contract and RR to RR.



```
LD       #10012 ⎫     ⎧ LD    #10012
AND      #14352 ⎬  ≡  ⎨ XNR   #14352
STR-NOT  #10012 ⎪     ⎩ OUT   #14040
AND-NOT  #14352 ⎪
OR-STR          ⎪
OUT      #14040 ⎭
```

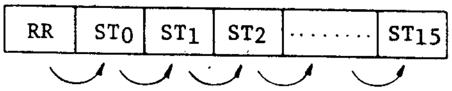
A	B	C
0	0	1
1	0	0
0	1	0
1	1	1

(9) STR (Store) {RR↕}

① Format STR # x x x x x

Internal signal name

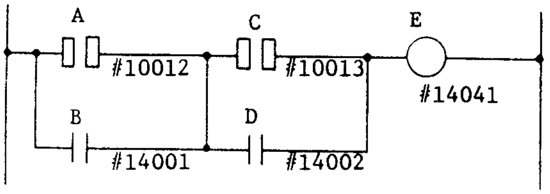
② Loads RR contents to stack.



Can use stacks up to 16.

Then, executes the LD instructions.

③ Normally, this instruction is used for signal of Contact A (—| |—).



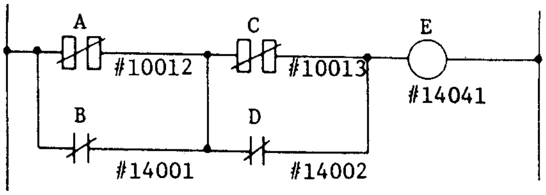
```
LD       #10012
OR       #14001
STR      #10013
OR       #14002
AND-STR
OUT      #14041
```

(10) STR-NOT (Store NOT) {RR↕}

① Format STR-NOT # x x x x x

Internal signal name

② Loads RR contents into stack and then executes the LD NOT instruction.

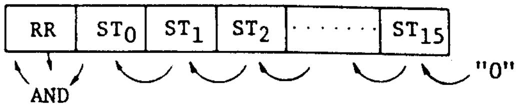


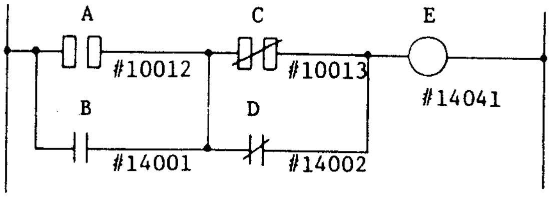
```
LD-NOT    #10012
OR-NOT    #14001
STR-NOT   #10013
OR-NOT    #14002
AND-STR
OUT       #14041
```

(11) AND-STR (AND-Store) {RR↕}

① Format AND-STR

② Executes AND of RR and stack (ST0) and loads the result to RR. The stack shifts by one each toward left.



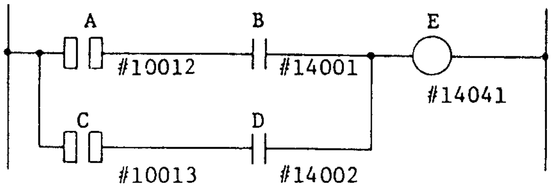


```
LD        #10012
OR        #14001
STR-NOT   #10013
OR-NOT    #14002
AND-STR
OUT       #14041
```

(12) OR-STR (OR-Store) {RR↕}

① Format OR-STR

② Executes OR of RR and stack (ST0) and loads the result to RR.



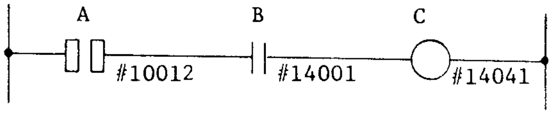
```
LD       #10012
AND      #14001
STR      #10013
AND      #14002
OR-STR
OUT      #14041
```

(13) OUT {RR–}

① Format OUT # x x x x x

Internal signal name

② Writes operation result (RR) to relay.



```
LD     #10012
AND    #14001
OUT    #14041
```

7.4 INSTRUCTIONS FOR TIMERS

(1) TIM (Fixed Timer) {RR time up = 1}

① Format TIM # x x x x, x x H

#1700 - #1799

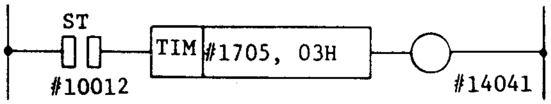
Timer set time (hexadecimal)

② The timer counts up in the state that the ST contact is ON (RP = 1), and sets TM on after the set time. In the state of the ST contact being OFF (RR = 0), TM is cleared and the timer is reset.

③ The timer set value is in the range of 0 - 255 (decimal notation). However, make sure to write this in a hexadecimal notation (NOTE 1). The CRT display is also in a hexadecimal notation.

④ Five types of timers can be used.

Address	Types	No. of Timers
#1700-#1709, #1760-#1769	Timer of 1 = 8 msec	20
#1710-#1729, #1790-#1799	Timer of 1 = 0.1 sec	30
#1730-#1749, #1780-#1789	Timer of 1 = 50 msec	30
#1750-#1759	Timer of 1 = 1 sec	10
#1770-#1773	Timer of 1 = 1 min	4



```
LD      #10012
TIM     #1705, 03H
OUT     #14041
```

Note:
1. A conversion table between decimal and hexadecimal notation is provided in Appendix 3 at the end.
2. The same address must not be used in fixed timer and variable timer, for normal operation cannot be guaranteed.

(2) TMR (Variable Timer) {RR time up = 1}

① Format TMR # x x x x, # x x x x

#1700 - #1799

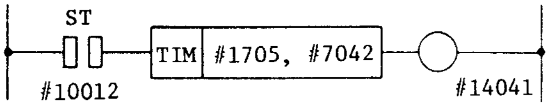
#7000 - #7294 address of sequence parameter

② The timer counts up in the state of the ST contact being ON (RR = 1), and TM is set on after the set time. When the ST contact is OFF (RR = 0), TM is cleared and the timer is reset.

③ The timer set value is in the range of 0 - 255 (decimal notation).

④ Set the aforementioned timer value through the NC keyboard in the procedures of "Parameter Write Operation." In this case, the write can be in a decimal notation, and the CRT display is also in a decimal notation.

⑤ The same as with the TIM instruction, 5 types of timers can be used with TMR.



```
LD      #10012
TMR     #1705, #7042
OUT     #14041
```

7.5 INSTRUCTIONS FOR REGISTERS

(1) INR (Increment Register) {RR -}

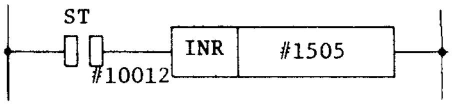
① Format INR # x x x x

#1400 - #1499
#1500 - #1599
#1600 - #1699
#1800 - #1899
#1900 - #1999 (register number)

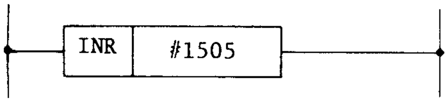
② Adds + 1 to the register contents when the ST contact is ON (RR = 1). This instruction is not executed when the ST contact is OFF (RR = 0).

③ The ST contact must be made before the INR instruction.

④ When the ST contact is ON, + 1 is added to the register contents in every 8 x "n" msec.



```
LD      #10012
INR     #1505
```



Cannot use this method

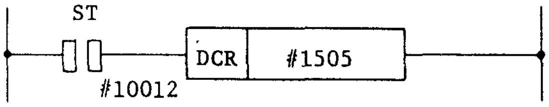
(2) DCR (Decrement Register) {RR -}

① Format DCR # x x x x

#1500 - #1599
#1800 - #1899
#1900 - #1999
(register number)

② When the ST contact is ON (RR = 1), - 1 is added to the register contents. This instruction is not executed when the ST contact is OFF (RR = 0). The RR contents remain unchanged.

③ The ST contact must be made before the DCR instruction.



```
LD  #10012
DCR #1505
```

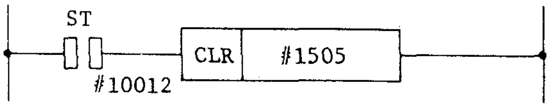
④ When the ST contact is ON, - 1 is added to the register contents in every 8 x "n" msec.

(3) CLR (Clea) {RR-}

① Format CLR # x x x x

#1500 - #1599
#1800 - #1899
#1900 - #1999
(register number)

② Clears the register contents when the ST contact is ON (RR = 1). This instruction is not executed when the contact is OFF (RR = 0). The RR contents remain unchanged



```
LD  #10012
CLR #1505
```

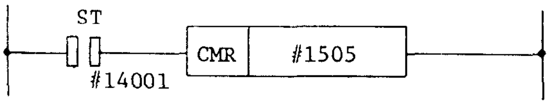
(4) CMR (Complement Register) {RR-}

① Format CMR # x x x x

#1500 - #1599
#1800 - #1899
#1900 - #1999
(register number)

② Inverts the register contents when the ST0 contact is ON (RR = 1). This instruction is not executed when the contact is OFF (RR = 0). The RR contents remain unchanged.

③ The ST contact must be made before the CMR instruction.



```
LD  #14001
CMR #1505
```

④ The register contents are inverted in every 8 x "n" msec when the ST contact is ON.

(5) ADI (Added Immediate) {RR-}

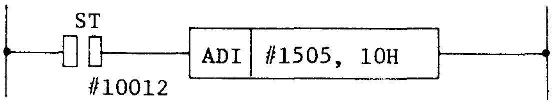
① Format ADI # x x x x, x x H

#1500 - #1599 Numeric (hexadecimal)
#1800 - #1899
#1900 - #1999
(register number)

② Adds the register contents and numeric and loads the result to the register when the ST contact is ON (RR = 1). This instruction is not executed when the contact is OFF (R = 0). The RR contents remain unchanged.

③ The ST contact must be made before the ADI instruction.

④ The ADI instruction is executed in every 8 x "n" msec when the ST contact is ON.



```
LD  #10012
ADI #1505, 10H
```

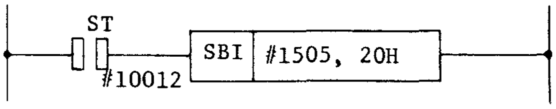
(6) SBI (Subtract Immediate) {RR-}

① Format SBI # x x x x, x x H

#1500 - #1599 Numeric (hexadecimal)
#1800 - #1899
#1900 - #1999
(register number)

② Subtracts the register contents and numeric and loads the result to the register when the ST contact is ON (RR = 1). If it is OFF, the instruction is not executed. The RR contents remain unchanged.

③ The ST contact must be made before the SBI instruction.



```
LD  #10012
ADI #1505, 20 H
```

④ The SBI instruction is executed in every 8 x "n" msec when the ST contact is ON.

7.5 INSTRUCTIONS FOR REGISTERS (Cont'd)

(7) ANI (And Immediate) {RR-}

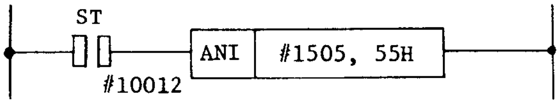
① Format ANI # x x x x, x x H

└ Numeric (hexadecimal)

#1500 - #1599
#1800 - #1899
#1900 - #1999
(register number)

② AND of the register contents and numeric is taken and loaded in the register when the ST contact is ON (RR = 1). If the contact is OFF (RR = 0), the instruction is not executed. The RR contents remain unchanged.

③ The ST contact must be made before the ANI instruction



```
LD   #10012
ANI  #1505, 55H
```

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Register	0	0	1	1	0	0	1	1
Numeric	0	1	0	1	0	1	0	1
Result	0	0	0	1	0	0	0	1

(8) ORI (Or Immediate) {RR-}

① Format ORI # x x x x, x x H

└ Numeric (hexadecimal)

#1500 - #1599
#1800 - #1899
#1900 - #1999

② OR of the register contents and numeric is taken and loaded in the register when the ST contact is ON (RR = 1). If the contact is OFF (RR = 0), the instruction is not executed. The RR contents remain unchanged.

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Register	0	0	1	1	0	0	1	1
Numeric	0	1	0	1	0	1	0	1
Result	0	1	1	1	0	1	1	1

(9) XRI (Exclusive or Immediate)

① Everything is the same as in the ORI instruction, with an exception of the following table.

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Register	0	0	1	1	0	0	1	1
Numeric	0	1	0	1	0	1	0	1
Result	0	1	1	0	0	1	1	0

(10) DEC (Decode) {RR↕}

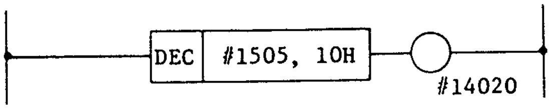
① Format DEC # x x x x, x x H

└ Numeric (hexadecimal)

Register and contact set

② RR is one when the data and numeric of the 8 bits of the register and contact set are equal. This will occur irrelevant to RR of the input side.

③ No contact can be added before the DEC instruction. Use the COI instruction when a contact must be added.



```
DEC  #1505, 10H
OUT  #14020
```

④ For example, if the M function output is #1222, to set on/off M11 with an M11 signal, the following must be given.

DEC #1222, OBH
OUT #14100 (relay for M11)

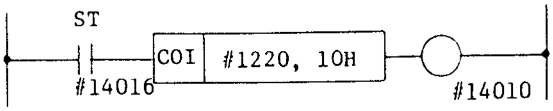
(11) COI (Coincide Immediate) {RR↕}

① Format COI # x x x x, x x H

Numeric

Register and contact set

② RR is set to "1" when the data and numeric of the register or contact set coincide when the ST contact is ON(RR = 1). If the contact is OFF (RR = 0), the COI instruction is not executed. RR is cleared.



```
LD   #14016
COI  #1220, 10H
OUT  #14010
```

(12) CMP (Compare) {RR↕}

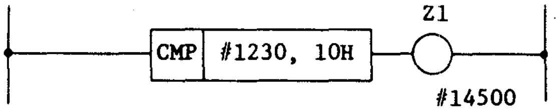
① Format CMP # x x x x, x x H

Numeric (hexadecimal)

Register and contact set

② If the comparison result of the 8-bit data and numeric of the register and contact set is that the register (contact set) is equal or greater than the numeric, RR is set to "1." If the register (contact set) is smaller than the numeric, RR is cleared. This is executed irrelevant to RR of the input side.

③ No contact can be added before the CMP instruction. Use the CPI instruction when a contact must be added.



#1230 ≥ 10H → Z1 = ON

#1230 < 10H → Z1 = OFF

```
CMP   #1230, 10H
OUT   #14500
```

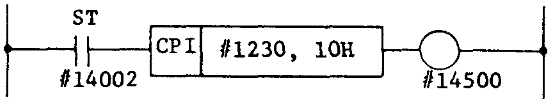
(13) CPI (Compare Immediate) {RR↕}

① Format CPI # x x x x, x x H

Numeric (hexadecimal)

Register and contact set

② RR is set to "1" if the comparison result of the data and numeric of the register or contact set is that the register (contact set) is greater or equal to the numeric when the ST contact is ON (RR = 1). When the ST contact is OFF (RR = 1), the CPI instruction is not executed. RR is cleared.



```
LD    #14002
CPI   #1230, 10H
OUT   #14500
```

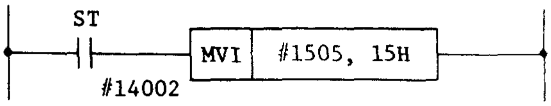
(14) MVI (Move Immediate) {RR-}

① Format MVI # x x x x, x x H

Register

Numeric (hexadecimal)

② This instruction transfers the numeric to the register when the ST contact is ON (RR = 1). If the contact is OFF (RR = 0), the MVI instruction is not executed.



```
LD    #14002
MVI   #1505, 15H
```

③ RR is not affected by the MVI instruction.

④ If the ST contact is ON, the MVI instruction is executed in every 8 x "n" msec.

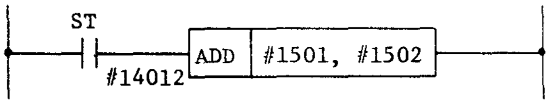
(15) ADD (ADD Register) {RR-}

① Format ADD # x x x x, # x x x

Register to be operated (R2)

Operating register (R1)

② When the ST contact is ON (RR = 1), the register (R2) contents and register (R1) are added and the result is loaded in register (R2). The R1 register contents remain unchanged. The RR contents also remain unchanged. The ADD instructions not executed when the ST contact is OFF (RR = 0).



```
LD    #14012
ADD   #1501, #1502
```

Note: In ADD or SUB, detection of overflow or underflow is not performed. With ADD, make the result less than 255 (FFH); with SUB, do not make $R_1 > R_2$.

(16) SUB (Sub Register) {RR-}

① Everything is the same as the ADD instruction, except here the operation is subtraction (R2-R1 → R2).

(17) ANR (And Register) {RR-}

① Everything is the same as the ADD instruction, except here the operation is AND, (R2 AND R1 → R2)

(18) ORR (Or Register) {RR-}

① Everything is the same as the ADD instruction, except here the operation is OR. (R2 OR R1 → R2)

7.5 INSTRUCTIONS FOR REGISTERS (Cont'd)

(19) XRR (Excluse or Register) {RR-}

① Everything is the same as the ADD instruction, except here the operation is XOR. (R2 XOR R1 → R2)

(20) CPR (Compare Register) {RR↕}

① Format CPR # x x x x, # x x x x

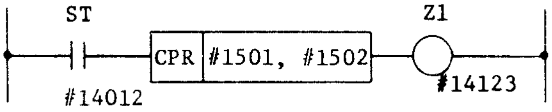
Register or contact set (R2)

Register or contact set (R1)

② When the ST contact is ON (RR = 1), the difference between R1 and R2 is taken, and;

RR is cleared if R is smaller than R2, and RR is set to "1" if R1 is greater than or equal to R2.

CPR is not executed when the ST contact is OFF (RR = 0). The RR contents remain unchanged.



```
LD    #14012
CPR   #1501, #1502
OUT   #14123
```

#1501 < #1502 ··· Z1 is set.
#1501 ≥ #1502 ··· Z1 is cleared.

③ The data in R1 and R2 remain unchanged when the CPR instruction is executed.

Note: The instructions for registers described in (16) through (20) execute their commands by 8 x nms when the ST contact is on. The instructions ADD, SUB and XRR will change their register contents by 8 x nms.

(21) COR (Coincide Register) {RR↕}

① Format COR # x x x x, # x x x x

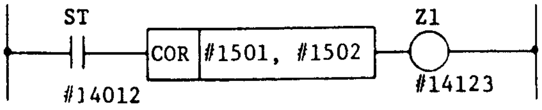
Register or contact set (R2)

Register or contact set (R1)

② When the ST contact is ON (RR = 1):

If R1 is equal to R2, Z1 is set.
If R1 is not equal to R2, Z1 is cleared.

When the ST contact is OFF (RR = 0), the COR instruction is not executed, and the RR contents remain unchanged.



```
LD    #14012
COR   #1501, #1502
OUT   #14123
```

#1501 = #1502 ··· Z1 is set.
#1501 = #1502 ··· Z1 is cleared.

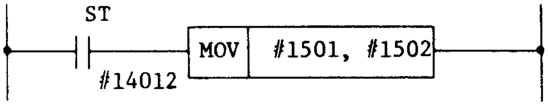
③ The data of R1 and R2 remains unchanged when the COR instruction is executed.

(22) MOV (Move Register) {RR-}

① Format MOV # x x x x, # x x x x

Register (R1) Register (R2)

② The R1 register contents are transferred to Register R2 when the ST contact is ON (RR = 1). The Register R1 contents remain unchanged.



```
LD    #14012
MOV   #1501, #1502
```

③ RR is not affected by the MOV instruction.

(23) DST (Data Store) {RR-}

① Format DST # x x x x, # x x x x, x x H

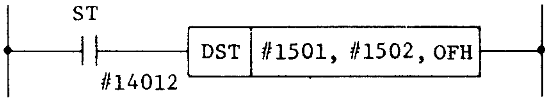
Numeric (hexadecimal)

Register (R2)

Register (R1)

② When the ST contacts in ON (RR = 1);

Register R1 and the numeric are ANDed, and the result is transferred to R2.
Register R1 remains unchanged.
When the ST contact is OFF (RR = 0);
The DST instruction is not executed.



```
LD    #14012
DST   #1501, #1502, OFH
```

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Reg. R1	B	B	B	B	B	B	B	B
Numeric	0	0	0	0	1	1	1	1
Reg. R2	0	0	0	0	B	B	B	B

B: "1" or "0"

③ RR is not affected by execution of the DST instruction.

(24) DIN (Data Insert) {RR-}

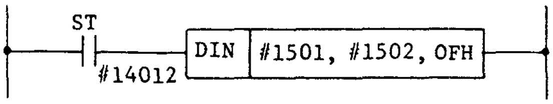
① Format DIN # x x x x, # x x x x, x x H

Numeric (hexadecimal)

Register or contact set (R2)

Register or contact set (R1)

② When the ST contact is ON (RR = 1), the R1 data and numeric are ANDed and the result is ORed with the AND of the R2 data and the numeric complement. The result is stored in R2 (data extraction). When the ST contact is OFF (RR = 1), the DIN instruction is not executed.



```
LD      #14012
DIN     #1501, #1502, OFH
```

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
R1	A	A	A	A	A	A	A	A
R2	B	B	B	B	B	B	B	B
n	0	0	0	0	1	1	1	1
Result	B	B	B	B	A	A	A	A

A, B: Data is "1" or "0."

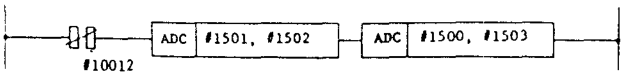
(25) ADC (Add with Carry) {RR}

① Format ADC # x x x x, # x x x x

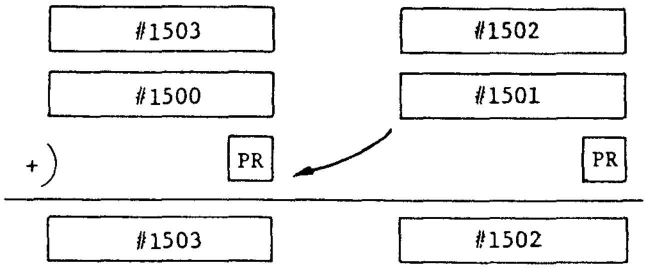
Register or contact set (R2)

Register or contact set (R1)

② Register R1, R2 and RR are added, and the result stored in Register R2. RR is set to "1" when a carry occurs.



```
LD NOT   #10012
ADC      #1501, #1502
ADC      #1500, #1503
```



③ RR must be cleared to execute the ADC instruction.

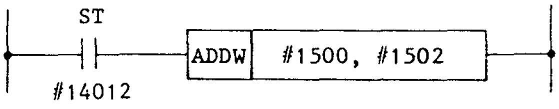
(26) ADDW (Add Word Register) {RR-}

① Format ADD # x x x x, # x x x x,

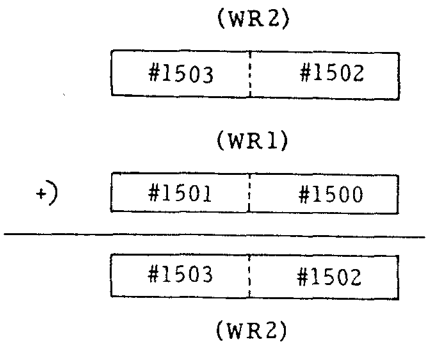
Low side of double length register (WR2)

Low side of double length register (WR1)

② When the ST contact is ON (RR = 1), the contents of double length registers, WR2 and WR1, are added and the result is stored in WR2. WR1 remains unchanged. (WR2) + (WR1) → (WR2). The RR contents do not change by the operation. When the ST contact is OFF (RR = 0), the ADDW instruction is not executed. The numeric is judged without code.



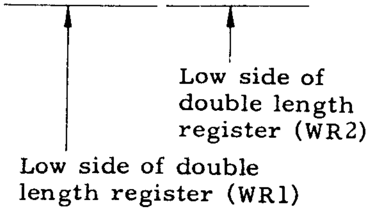
```
LD      #14012
ADDW    #1500, #1502
```



7.5 INSTRUCTIONS FOR REGISTERS (Cont'd)

(27) SUBW (Sub Word Register) {RR-}

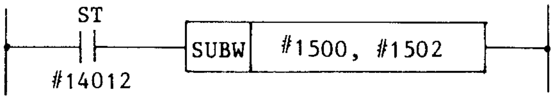
① Format SUBW # x x x x, # x x x x


Low side of double length register (WR2)
Low side of double length register (WR1)

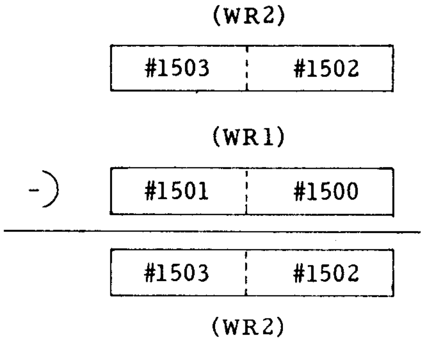
② When the ST contact is ON (RR = 1), the results of the contents of double length registers, WR2 minus WR1 is stored in WR2. WR1 remains unchanged.

(WR2) - (WR1) → (WR2)

When the ST contact is OFF (RR = 0), the SUBW instruction is not executed. The numeric is judged without code.

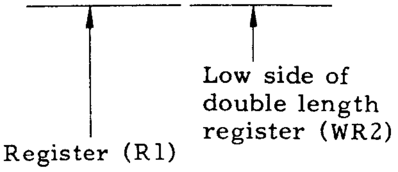


```
LD     #14012
INRW   #1500, #1502
```



(28) MULW (Mul Word Register) {RR↕}

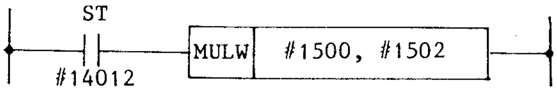
① Format MULW # x x x x, # x x x x



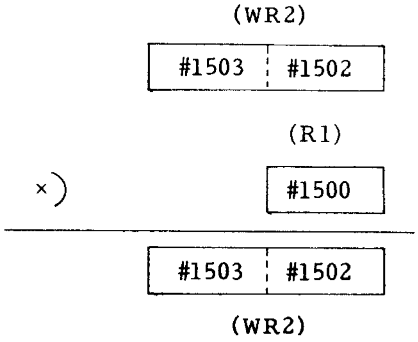
② When the ST contact is ON (RR = 1), the contents of double length register, WR2 and register R1 are multiplied, and the result is stored in WR2. R1 remains unchanged.

(WR2) × (R1) → (WR2)

When the ST contact is OFF (RR = 0), the MUL instruction is not executed. The numeric is judged without code. If the result is overflown, more than "FFFFH," RR equals one.

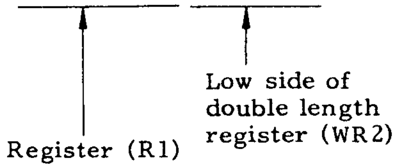


```
LD    #14012
MUL   #1500, #1502
```

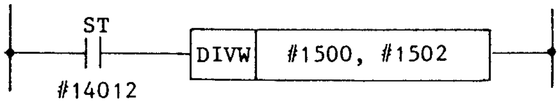


(29) DIVW (Division Word Register) {RR-}

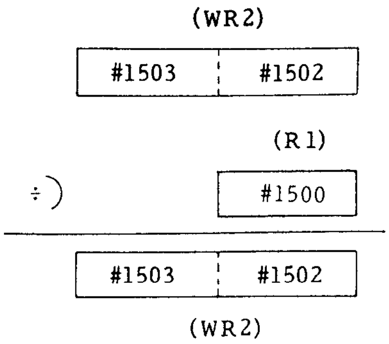
① Format DIVW # x x x x, # x x x x



② When the ST contact is ON (RR = 1), the contents of double length register WR2 is divided by register R1 and the result is stored in WR2. WR1 remains unchanged. When the ST contact is OFF (RR = 0), DIV instruction is not executed. The numeric is judged without code. If WR1 is "0," operation will not be executed.

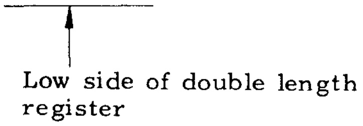


```
LD    #14012
DIV   #1500, #1502
```

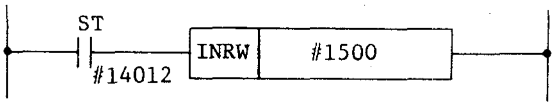


(30) INRW (Increment Word Register) {RR-}

① Format INRW # x x x x



② When the ST contact is ON, +1 is added to the double length register contents.



```
LD     #14012
INRW   #1500
```

(31) DCRW (Decrement Word Register) {RR-}

① The same as INRW, but the operation here is addition of -1 to the double length register contents.

(32) CLRW (Clear Word Register) {RR-}

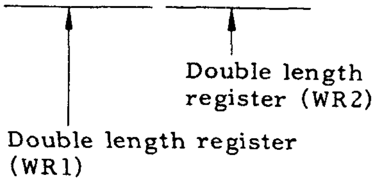
① The same as INRW, but here the double length register contents are cleared.

(33) CMRW (Complement Word Register) {RR-}

① The same as INRW, but here the double length register contents are inverted.

(34) CORW (Coincide Word Register) {RR↕}

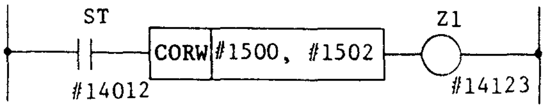
① Format CORW # x x x x, # x x x x



② When the ST contact is ON (RR = 1), WR1 and WR2 are checked for the coincidence;

If WR1 and WR2 are equal, RR is set to 1.
If WR1 and WR2 are not equal, RR is cleared.

When the ST contact is OFF (RR = 0), the CORW instruction is not executed, and the RR contents remain unchanged.



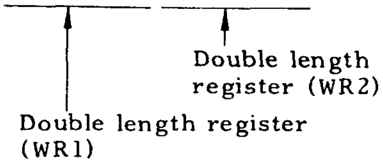
```
LD     #14012
CORW   #1500, #1502
OUT    #14123
```

#1500 = #1502 ··· Z1 is set.
#1500 = #1502 ··· Z1 is cleared.

③ The data of WR1 and WR2 do not change when the CORW instruction is executed.

(35) CPRW (Compare Word Register) {RR↕}

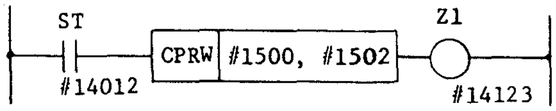
① Format CPRW # x x x x, # x x x x



② When the ST contact is ON (RR = 1), WR1 and WR2 are checked for the difference;

If WR1 is smaller than WR2, RR is cleared.
If WR1 is greater than or equal to WR2, RR is set.

When the ST contact is OFF (RR = 0), the CPRW instruction is not executed. The RR contents remain unchanged.

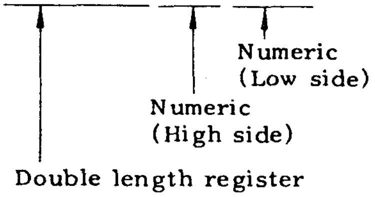


```
LD     #14012
CPRW   #1500, #1502
OUT    #14123
```

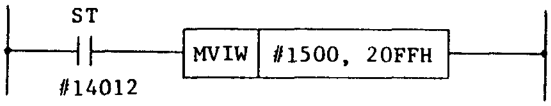
#1500 < #1502 ··· Z1 is set.
#1500 ≧ #1502 ··· Z1 is cleared.

(36) MVIW (Move Immediate Word Register) {RR-}

① Format MVIW # x x x x, x x x x H



② When the ST contact is ON (RR = 1), the numeric is transferred to the register. When the ST contact is OFF (RR = 0), the MVIW instruction is not executed.



③ The RR contents are not affected by execution of the MVIW instruction.

7.5 INSTRUCTIONS FOR REGISTERS (Cont'd)

(37) DSTW (Data Store Word Register) {RR-}

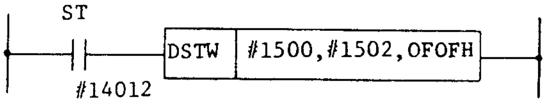
① Format DSTW #xxxx, #xxxx, xxxxH

Numeric

Register (WR2)

Register (WR1)

② When the ST contact is ON (RR = 1), Register WR1 and the numeric and ANDed and the result is transferred to Register WR2. The WR1 contents remain unchanged. When the ST contact is OFF (RR = 0), the DSTW instruction is not executed.



```
LD     #14012
DSTW   #1500, #1502, OFOFH
```

	D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8
Reg. WR1	B	B	B	B	B	B	B	B
Numeric	0	0	0	0	1	1	1	1
Reg. WR2	0	0	0	0	B	B	B	B

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Reg. WR1	B	B	B	B	B	B	B	B
Numeric	0	0	0	0	1	1	1	1
Reg. WR2	0	0	0	0	B	B	B	B

B: "1" or "0"

③ The RR contents remain unchanged when the DST instruction is executed.

7.6 CONTROL INSTRUCTIONS

(1) NOP (No Operation) {RR-}

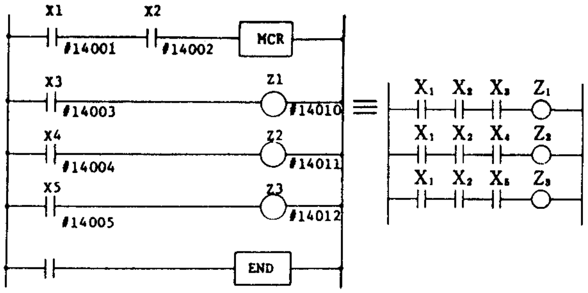
① Format NOP

② No operation is conducted and the system moves to the next step. The RR contents remain unchanged.

(2) MCR (Master Control) {RR-}

① Format MCR

② When the X1 and X2 contacts are ON (RR = 1), the sequence ladder is released.
When the X1 and X2 contacts are OFF (RR = 0), the ladder up to END is executed in the state of RR being "0."



```
LD    #14001
AND   #14002
MCR
LD    #14003
OUT   #14010
LD    #14004
OUT   #14011
LD    #14005
OUT   #14012
END
```

Where X1 and X2 contacts are off, is given from internal relays Z1, Z2 and Z3.

③ Another MCR instruction can be given between MCR and END (7 levels max).

④ When a timer instruction is included in MCR, the timer is cleared when MCR is OFF.

⑤ Even if a self-holding circuit is formed between MCR and END instructions, the circuit output is OFF when MCR input contact is OFF.

(3) END (Master Control End) {RR-}

① Format END

② Indicates that MCR is at the end.

(4) RET (Return) {RR-}

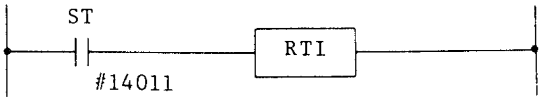
① Format RET

② Indicate the end of sequence program.

(5) RTI (Return Indirect) {RR-}

① Format RTI

② When the ST contact is OFF, ladder of the next step is executed.



```
LD    #14011
RTI
```

(6) SET (Set Return Register) {RR-}

① Format SET

② Forcibly sets RR to "1."

(7) RTH (Return High Sequence) {RR-}

① Format RTH

② Indicates the end of a high speed sequence program.

7.7 MACRO INSTRUCTIONS

Macro instructions (SUBPxxx) are provided to enable the operators to simply arrange operations of machine tools with which ladders cannot be prepared easily with basic instructions (relay instruction, register instruction, etc.) only. The following explains further details. The format of macro instructions is as follows:

SUBP x x x

Macro instruction number

The following auxiliary instructions are used with macro instructions:

● IPSH (Immediate Push) {RR-}

① Format IPSH x x x x H

Numeric (hexadecimal)

② Directly designate the numeric used with SUBP.

● APSH (Address Push) {RR-}

① Format APSH # x x x x

Register

② Designate the address of the register used with SUBP.

● PUSH (Push) {RR-}

① Format PUSH # x x x x

Register

② Designate the address where the numeric used with SUBP is stored.

● TPSH (Table Push) {RR-}

① Format TPSH x x x x

Table number

② Designates the table number of PC table used with SUBP.

(1) SUBP 005 (Counter)

① Function: This counter can be used in many ways to control machine tool operation according to the applications, as described below.

(a) Ring counter

This counter is ring counter. Accordingly, it returns to the initial value when a count signal is input after counting up to the preset value.

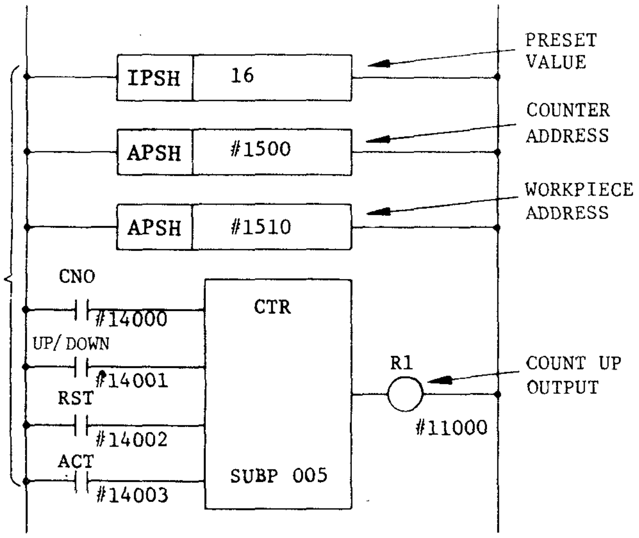
(b) Preset counter

If a count number is preset, and the count value reaches the set value, COUNT UP is output.

(c) Up/Down counter

This counter can be used for up count and down count also.

② Form



```
IPSH 16        ... Preset value
APSH #1500     ... Counter address
APSH #1510     ... Workpiece address
LD   #14000    ... CNO
STR  #14001    ... UP DOWN
STR  #14002    ... RST
STR  #14003    ... ACT
SUBP 005       ... COUNTER instruction
OUT  #11000    ... COUNT UP output
```

③ Control conditions

(a) Preset value designation (IPSH xx)

Directly designate a preset value.
To designate a variable value, use the PUSH instruction, instead of IPSH, and designate the address. The preset value becomes the address contents.

Example:

PUSH #1550

If the above designation is given, the two byte of #1550 and #1551 are used. Do not use #1551 for others even if only one byte is to be used.

(b) Counter address designation (APSH #xxxx)

Designate the counter address.
If APSH #1500 is designated, the continuous two bytes, that is, #1500 and #1501, are used for the counter address.

(c) Workpiece address designation (APSH#xxxx)

Designate an address that is not used by other instructions. 1 byte is needed for one SUBP 005. When two or more SUBP 005 are used, designate an address to each of it.

(d) Initial value designation (CNO)

CNO = 0: The counter cumulative value starts at "0."
(0, 1, 2, 3, 4, ... n)

CNO = 1: The counter cumulative value starts at "1."
(1, 2, 3, 4, 5, ... n)

(e) UP/DOWN designation

UP/DOWN = 0: Up counter
Initial value is "0" with CNO = 0
Initial value is "1" with CNO = 1

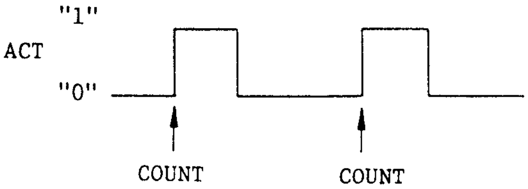
UP/DOWN = 1: Down counter
The initial value is the preset value.

(f) Reset (RST)

RST = 0: Reset release

RST = 1: Reset
R1 is cleared. The cumulative values is set to the initial value.

(g) COUNT signal (ACT)



ACT = 0: The counter does not operate. The R1 contents remain unchanged.

ACT = 1: Counts at the rise of "0" to "1."

Note:
If the counter contents are greater than the preset value at the time of power turn on:

In the case of Up counter:
Returns to the initial value with the first ACT.

In the case of Down counter:
Counts down each time ACT is applied, and when the value enters within the preset value, the operation afterward is normal.

(h) COUNT UP output (R1)

Up counter:
R1 is set to "1" upon counting up to the preset value.

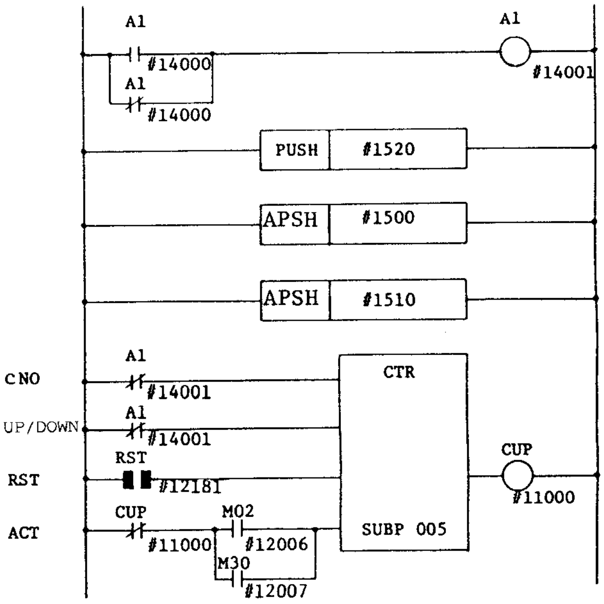
Down counter:
When CON = 0
R1 is set to "1" when counted down to "0."

When CON = 1
R1 is set to "1" when counted down to "1."

④ Counter use example

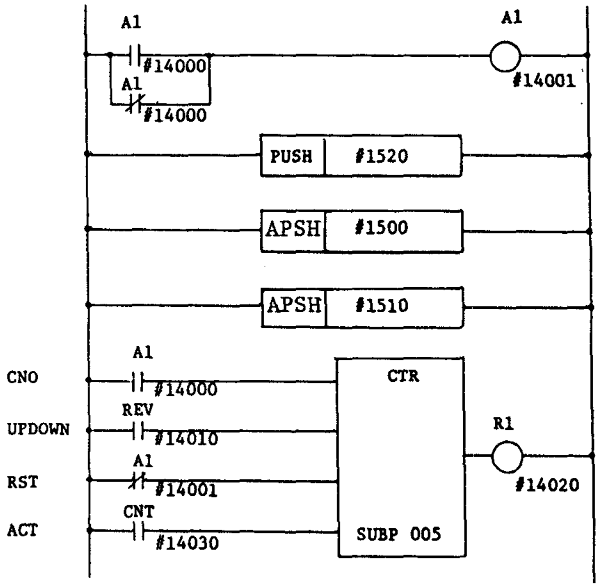
(a) Example of using the counter as a preset counter

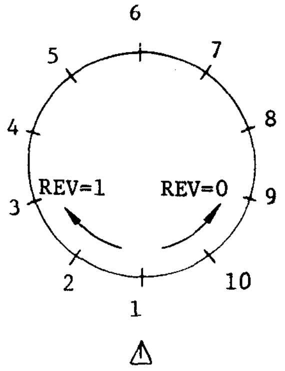
The number of machined workpieces is counted. When the count reaches the set value, the COUNT UP signal is output.



- A1 is the circuit to create Logic "1."
- NC contact of A1 is used to clear CNO since the count range used is 0 to 9999.
- NC contact of A1 used to clear UP DOWN as it is used as an UP counter.
- RST, the input signal from the NC unit, is used as the counter reset signal.
- The count signal is the input signal from the NC unit, M02 or M30. NC contact of CUP is contained in this signal the counter does not count once it counted up unless it is reset.

(b) Example of using the counter to memorize the rotating object position.





INDEXED POSITION

- A1 is circuit to create Logic "1."
- With the rotating object of 10 angles, as shown in the figure, the count start number is 1. Therefore, NO contact of A1 is used to CNO to "1."
- REV is a signal that changes according to the rotation direction. It is "0" for forward rotation and "1" for revers set CNO to "1."
- REV is a signal that changes according to the rotation direction. It is "0" for forward rotation and "1" for reverse rotation. Therefore, it operates as an Up counter for forward rotation and as a Down counter for reverse rotation.
- Since no reset signal is used in this example, it is kept to "0" always. Therefore, NC contact of A1 is used.
- The CNT count signal is a signal to turn ON/OFF 10 times for one rotation of the rotation object.
- Set 10 and 0 to the preset value addresses of #1520 and #1521, respectively.

(2) SUBP 006 (ROTATION)

① Function:

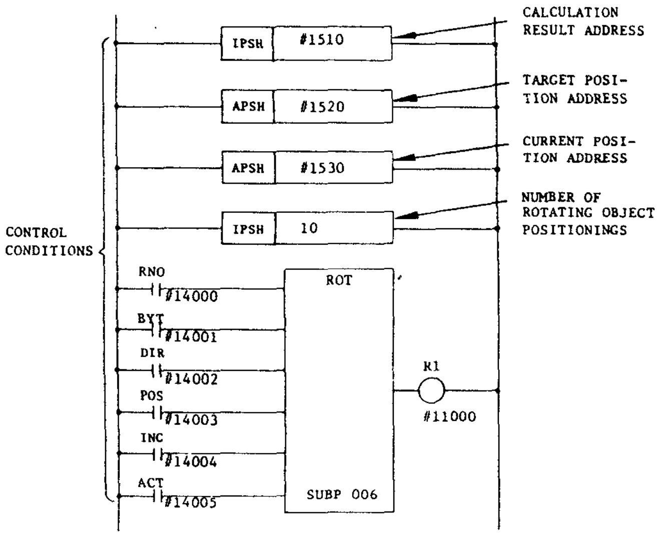
This instruction is used to control rotation objects such as blade base, ACT and rotating table. It has the following functions:

(a) Judgement of short-cut rotation direction

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

(c) Calculation of the position of one step before the target position or the number of steps up to one step before the target position.

② Form



APSH #1510 ... Calculation result output address

APSH #1520 ... Target position input address

APSH #1530 ... Current position address

IPSH 10 ... Number of rotating object positionings

LD #14000 ... The position number is from "0" or "1."

```
STR  #14001  ... The position data is in
                 1 byte or 2 bytes.
STR  #14002  ... The rotation direction is
                 constant or in shortcut.
STR  #14003  ... Target position or one
                 step before
STR  #14004  ... Position number of
                 number of steps
STR  #14005  ... Execution
SUBP 006     ... ROT instruction
OUT  #11000  ... Rotation direction output
```

③ Control conditions

(a) Designation of calculation result storage address (APSH#xxxx)

The ROT instruction calculates the number of steps that the rotating object should rotate, step number of one step before or the position of one step before the target position, and the result is stored in the designated address.

(b) Designation of target position address (APSH#xxxx)

Designate the address at which the target position is contained. In other words, this is the address in which the T command from the NC unit is contained.

(c) Designation of current position address (APSH#xxxx)

Designate the address where the current position is stored. For example, this is the address of the counter that memorizes the rotating object position.

(d) Designation of initial value of the position number of rotating object (RNO)

RNO = 0: The position number of rotating object starts from "0."

RNO = 1: The position number of rotating object starts from "1."

(e) Designation of number of bytes of position data (BYT)

BYT = 0: Binary 1 byte

BYT = 1: Binary 2 bytes

(f) Designation of whether or not short-cut direction should be determined (DIR)

DIR = 0: No determination is made on short-cut direction. The rotation direction is forward only.

DIR = 1: Determines short-cut direction.

(g) Designation of operation conditions (POS)

POS = 0: Calculate the number of steps to the target position.

POS = 1: Calculates the position or number of steps of one step before the target.

(h) Designation of position or number of steps (INC)

INC = 0: Calculates the position number.

INC = 1: Calculates the number of steps.

(i) Execution command (ACT)

ACT = 0: No execution of ROT instruction. R1 is not affected.

ACT = 1: Execute the ROT instruction. (This is not a rise signal.)

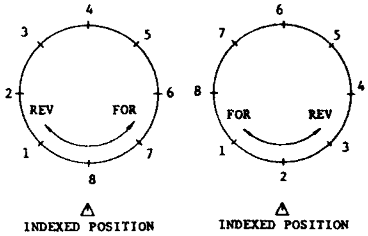
(j) Rotation direction output (R1)

R1 = 0: The rotation direction is forward.

R1 = 1: The rotation direction is reverse.

Note:

1. The rotation direction is defined as shown below:

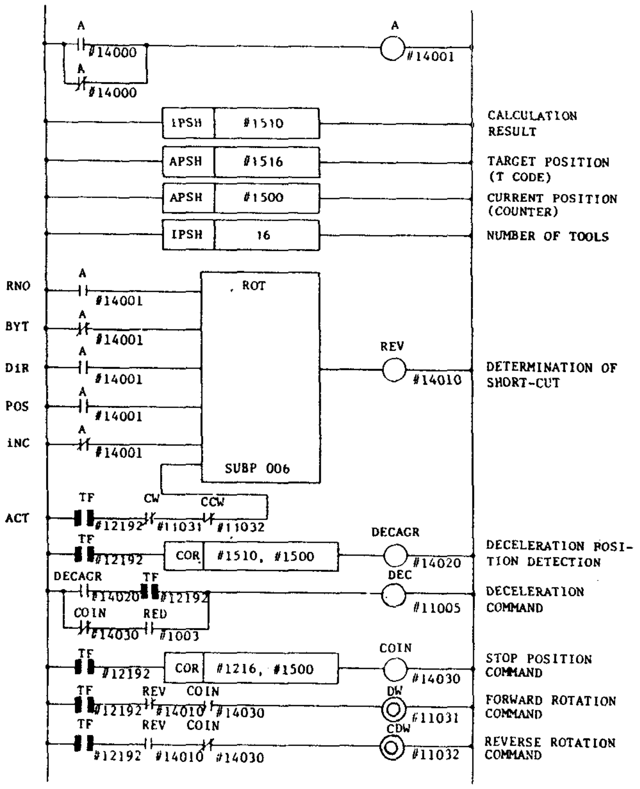


The rotation direction in which the number increases from the indexed position is the forward direction. The direction in which the number decreases is the reverse direction.

2. When the current position is equal to the target position, the calculation result of the number of steps of one step before the target position (POS = 1, INC = 1) is "0."

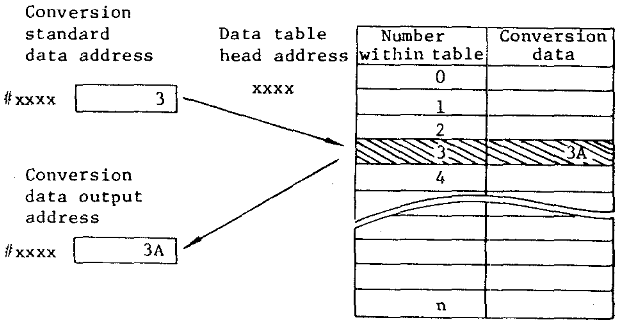
④ Use of example of ROT instruction

The following shows the control of a 16-position rotating object, without short-cut control but for deceleration at the position of one step before the target position.

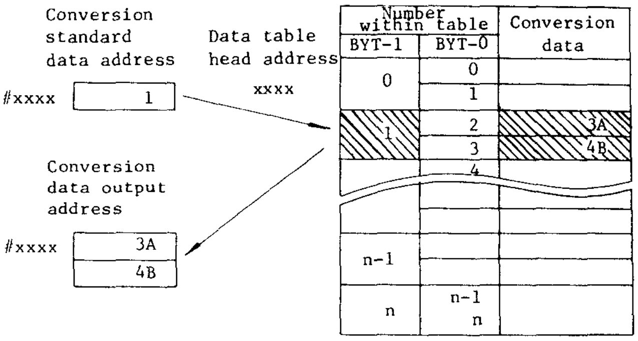


(3) SUBP 007 (CODE CONVERSION)

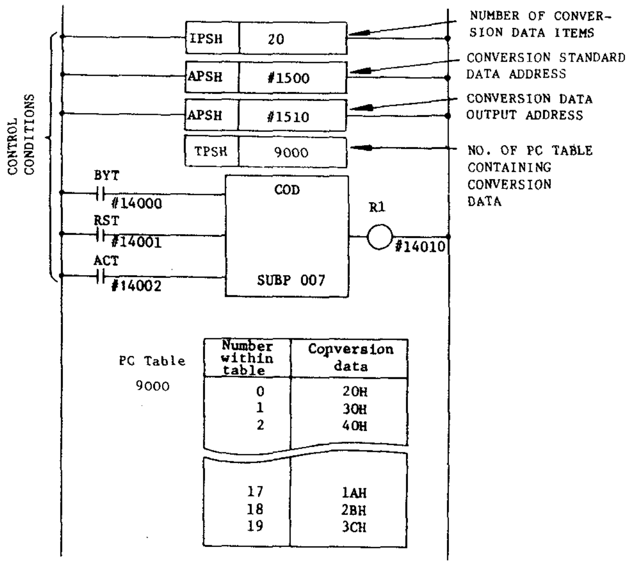
① Function: Converts data using the PC table prepared on the ladder.



- When "3" is instructed for the conversion standard data address with BYT = 0, as shown in the above figure, the data of the third address from the head of the table is stored in the conversion data output address. The head address of the table is "0."
- The status when BYT is set to "1" is shown below. At this time, check that the size of the conversion data table is in a even byte number.



② Form



IPSH 20	...	Size of conversion data table (Number of bytes).
APSH #1500	...	Conversion data address
TPSH #9000	...	No. of PC table containing conversion data.
APSH #1510	...	Converted data store address.
LD #14000	...	Data of data table is in 1 byte or 2 bytes.
STR #14001	...	Reset
STR #14002	...	Execution
SUBP 007	...	COD instruction
OUT #14010	...	ERROR output

-20H
-30H
-40H
.
.
.
-1AH
-2BH
-3CH

} Conversion data table

7.7 MACRO INSTRUCTIONS (Cont'd)

③ Control conditions

(a) Designation of number of conversion data items (IPSH xx)

Designate the size (number of bytes) of the conversion data table. The maximum size is 256 bytes.

(b) Designation of conversion standard data address (APSH #xxxx)

Data in the conversion data table is fetched out by designating the number inside the data table. Designate this number inside the table.

(c) Designation of conversion data output address (APSH #xxxx)

Designate the address to output the data stored in the number inside the table that is designated by Item b. When BYT is "1," data at the higher side is output to the address next to the designated address.

(d) Designation of conversion data table (TPSH xxxx)

Table size is different depending on PC table No.

• 9000 - 9007:	256 bytes max
• 9008 - 9023:	128 bytes max
• 9024 - 9087:	64 bytes max
• 9088 - 9215:	32 bytes max
• 9216 - 9435:	16 bytes max

(e) Designation of data size (BYT)

BYT = 0: When data of the conversion data table is in 1 byte.

BYT = 1: When data of the conversion data table is in 2 bytes.

(f) Reset (RST)

RST = 0: No reset.

RST = 1: ERROR output R1 is cleared.

(g) Execution command (ACT)

ACT = 0: No execution. R1 does not change.

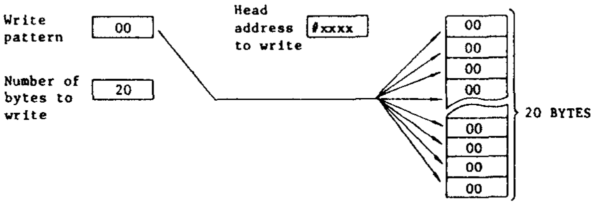
ACT = 1: Executes.

(h) Error output (R1)

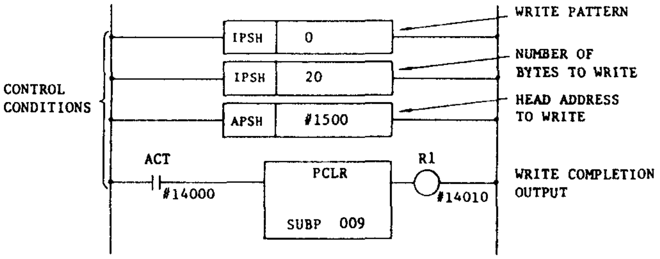
An error that has occurred during execution of the COD instruction (when a numeric that is greater than the table size). R1 is set to "1" to notify the error.

(4) SUBP 009 (PATTERN CLEAR)

① Function: Writes the same numeric for the designated number of bytes from the designated address.



② Form



```
IPSH 0        ... Write pattern
IPSH 20       ... Number of bytes to write
APSH #1500    ... Head address to write
LD    #14000  ... Execution
SUBP 009      ... PCLR instruction
OUT  #14010   ... Write completion output
```

③ Control conditions

(a) Designation of write pattern (IPSH xx)

Designate a write pattern.
If the pattern is to be variable, use PUSH, instead of IPSH, and designate the address.

(b) Designation of number of bytes to write (IPSH xx)

Designate the number of bytes for pattern clear.

(c) Designation of the head address to write (APSH #xxxx))

Designate the head address for PATTERN CLEAR start. PATTERN CLEAR is executed for the designated number of bytes from the address.

(d) Execution command (ACT)

ACT = 0: No execution.

ACT = 1: Executes.

(e) Write completion output (R1)

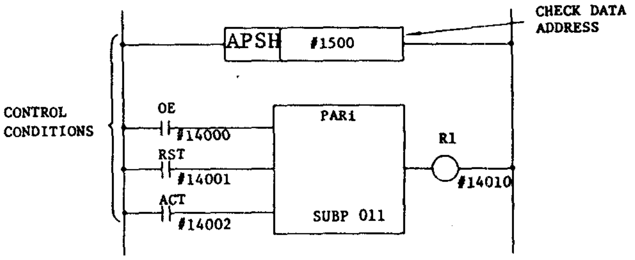
R1 = 0: Write not completed yet.

R1 = 1: Write completed.

(5) SUBP 011 (PARITY CHECK)

① Function: Parity check (even and odd) of the check data (1-byte data).
If not normal, an ERROR output it made.

② Form



APSH #1500 ... Check data address

LD #14000 ... Even/odd parity switching

STR #14001 ... Reset

STR #14002 ... Execution command

SUBP 011 ... PARI instruction

OUT #14010 ... ERROR output

③ Control conditions

(a) Designation of check data address (APSH #xxxx).

Designate the address where the data to be checked is stored. This data to be checked is in 1 byte (8 bits).

(b) Odd/Even command (OE)

OE = 0: Even parity check

OE = 1: Odd parity check

(c) Reset

RST = 0: No reset.

RST = 1: Resets ERROR output R1.

(d) Execution command (ACT)

ACT = 0: No execution of PARI instruction. R1 does not change.

ACT = 1: Executes PARI instruction.

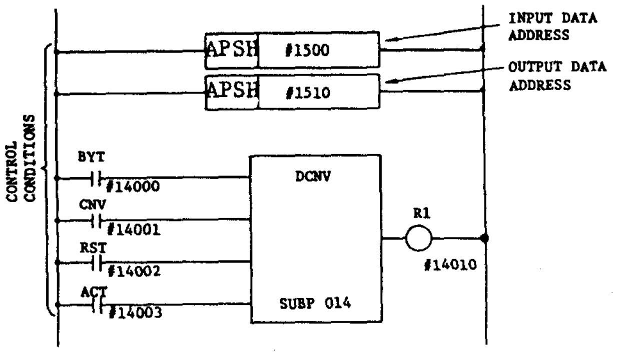
(e) Error output (R1)

When an odd parity resulting from even parity check or even parity resulting from odd parity check, ERROR output R1 is set to "1."

(6) SUBP 014 (DATA CONVERSION)

① Function:
Converts binary data to BCD data, or vice versa.

② Form



APSH #1500 ... Data address to be converted

APSH #1510 ... Conversion result storing address.

LD #14000 ... 1-byte or 2-bytes processing.

STR #14001 ... Conversion from binary to BCD or vice versa.

STR #14002 ... Reset

STR #14003 ... Execution

SUBP 014 ... DCNV instruction

OUT #14010 ... ERROR output

③ Control conditions

(a) Input address of data to be converted (APSH #xxxx)

Designate the address where the data to be converted is stored. In the case of BYT = 1, two continuous bytes are used for the address.

(b) Conversion result storing address

This address stores the converted data. Where BYT = 1, continuous bytes are used.

(c) Designation of number of bytes of data (BYT)

BYT = 0: The processing data is in one byte.

BYT = 1: The processing data is in two bytes.

(d) Designation of conversion form (CNB)

CNV = 0: Converts binary data to BCD data.

CNV = 1: Converts BCD data to binary data.

(e) Reset (RST)

RST = 0: No reset.

RST = 1: Resets error output R1.

(f) Execution command (ACT)

ACT = 0: No execution.

ACT = 1: Execution.

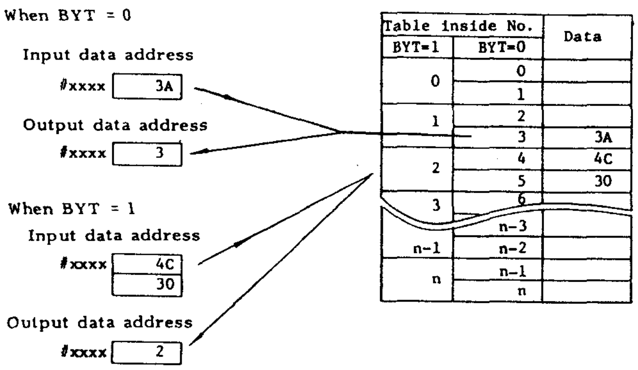
(g) ERROR output (R1)

R1 = 0: Normal

R1 = 1 : Abnormal (The data to be converted is binary data when CNV = 1, or the byte length was exceeded when CNV = 0. When BYT = 1 ; CNV = 0, R1 is not output unless BCD data is more than 2711 (H).

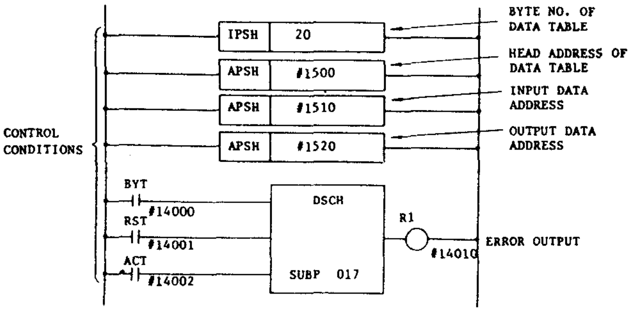
(7) SUBP 017 (DATA SEARCH)

① Function:
Searches the same data as the input data in the table. If there is, the relative address from the table head is stored in the output data address. If the same data is not found, an ERROR output is made.



Note: Check that the table size is in as even byte number when BYT = 1.

② Form



IPSH 20 ... Number of bytes of data table

APSH #1500 ... Head address of data table

APSH #1510 ... Search data address

APSH #1520 ... Table inside number storing address

LD #14000 ... The processing data is in one byte or two bytes.

STR #14001 ... Reset

STR #14002 ... Execution

STR #14003 ... Execution

SUBP 017 ... DSCH instruction

OUT #14010 ... ERROR output

③ Control conditions

(a) Designation of number of data items of data table (IPSH xx)

Designate the data table size (number of bytes).

(b) Designation of head address of data table (APSH #xxxx)

Designate the head address of the data table. The data table may be created in any place.

(c) Designation of input data address (APSH #xxxx)

Designate the address where the data to be searched is stored.

(d) Designation of output data address (APSH #xxxx)

If the searched data is found (R1 = 0), the number inside the table where the data is stored is output. Designate the output address.

(e) Designation of data size (BYT)

BYT = 0: The stored in the data table is in one byte.

BYT = 1: The data stored in the data table is in two bytes.

(f) Execution command (ACT)

ACT = 0: No execution

ACT = 1: Execution

(g) Reset (RST)

RST = 0: Not reset.

RST = 1: Reset. R1 is cleared.

(h) ERROR output (R1)

R1 = 0: The search data is found.

R1 = 1: The search data is not found.

(8) SUBP 018 (INDEX DATA MOVE)

① Function: Reads or re-writes data from the data table.

(a) Read
- "3" was designated as the table inside number and the contents were read.

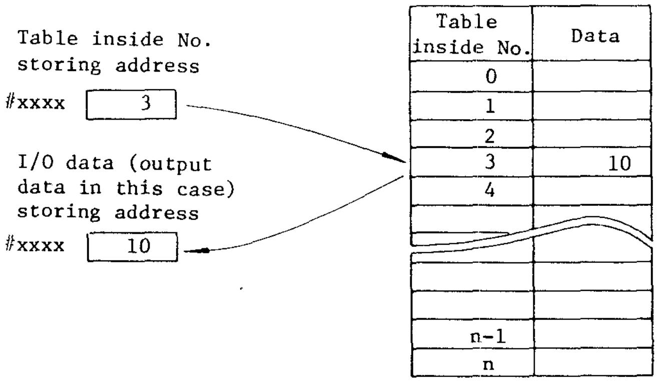


Table inside No. storing address
#xxxx 3
I/O data (output data in this case) storing address
#xxxx 10
Table inside No.
Data
0
1
2
3 10
4
n-1
n

(b) Re-write

- "3" was designated as the table inside number and the contents were re-written.

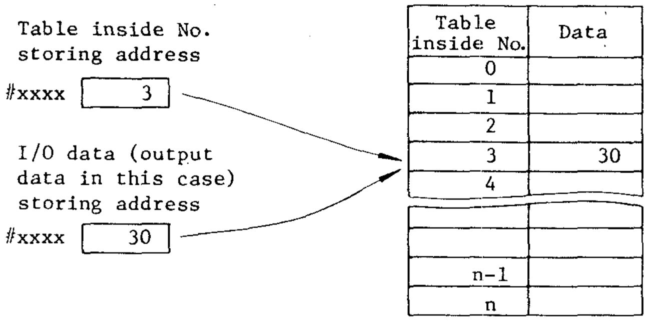
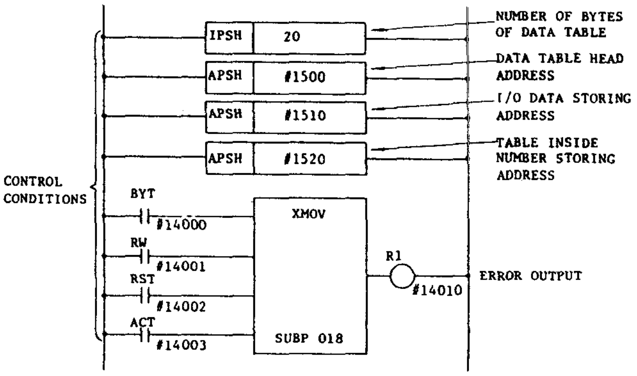


Table inside No. storing address
#xxxx 3
I/O data (output data in this case) storing address
#xxxx 30
Table inside No.
Data
0
1
2
3 30
4
n-1
n

② Form



IPSH 20	...	Number of bytes of data table
APSH #1500	...	Data table head address
APSH #1510	...	I/O data storing address
APSH #1520	...	Table inside number storing address
LD #14000	...	The processing data is in one byte or two bytes.
STR #14001	...	Read or Re-write
STR #14002	...	Reset
STR #14003	...	Execution
SUBP 018	...	XMOV instruction
OUT #14010	...	ERROR output

③ Control conditions

(a) Designation of number of data items of data table (IPSH xx)

Designate the data table size (number of bytes).

(b) Designation of data table head address (APSH #xxxx)

Designate the data table head address. The data table may be created in any place.

(c) Designation of I/O data storing address (APSH #xxxx)

RW = 0: Address to store output data.

RW = 1: Address to store input data.

(d) Designation of table inside number storing address (APSH #xxxx)

Designate which data in the data table should be read or re-written with a table inside number. The table inside number designates the storing address.

(e) Designation of data size (BYT)

BYT = 0: The data stored in the data table is in one byte.

BYT = 1: The data stored in the data table is in two bytes.

(f) Designation of read or re-write (RW)

RW = 0: Reads data from the data table.

RW = 1: Re-writes data from the data table.

(g) Reset (RST)

RST = 0: Not reset.

RST = 1: Reset. R1 is cleared.

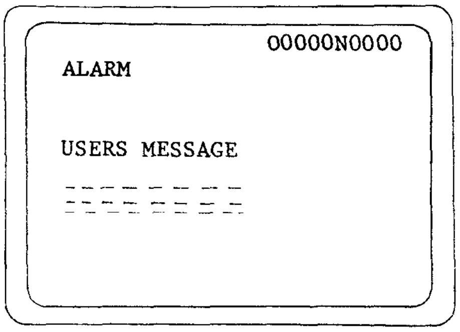
(h) Execution command (ACT)

ACT = 0: No execution

ACT = 1: Execution

(9) SUBP 023 (MESSAGE DISPLAY)

① Function: Displays messages on the CRT of NC.



The message is displayed under the title of USERS MESSAGE.

The message is displayed under the title of USERS MESSAGE.

Max. number of characters and types of messages are as follows. One of each is selected.

Max. number of characters	Type	Table address
32 bytes	128	#9088 to #9215
64 bytes	64	#9024 to #9087

The following shows the max. number that can be displayed on the CRT at the same time.

Max. number of characters	Number of simultaneous displays
32 bytes	2 sets
64 bytes	1 set

- Up to 4 messages are displayed on the CRT screen. If there is a request to display more messages, low order bits are given the priority. Messages of higher priority are displayed sequentially.
- The displayed messages set the corresponding bits to "1," and messages to be cleared clear the corresponding bits. The figure below shows the correspondence.

Display request	7	6	5	4	3	2	1	0	#1500
	15	14	13	12	11	10	9	8	#1501
Disnlay status	7	6	5	4	3	2	1	0	#1502
	15	14	13	12	11	10	9	8	#1503
Display request	23	22	21	20	19	18	17	16	#1504
	31	30	29	28	27	26	25	24	#1505
Display status	23	22	21	20	19	18	17	16	#1506
	31	30	29	28	27	26	25	24	#1507

Note:

1. Do not set bits containing no message data to "1."
2. This instruction is an instruction to display messages on the CRT screen. The instruction cannot set NC to an alarm state (1-block atop, decelerated stop, and immediate stop).

② Form

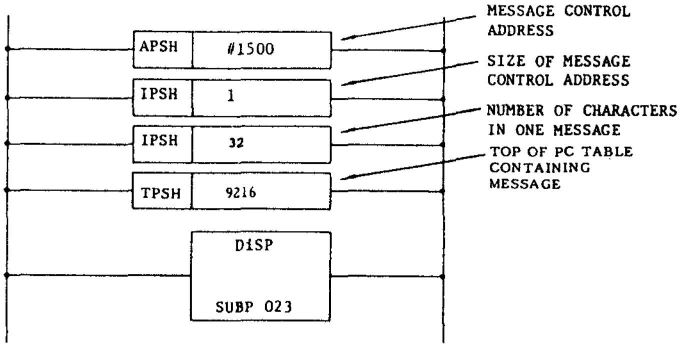

APSH
#1500
IPSH
1
IPSH
32
TPSH
9216
DISP
SUBP 023
MESSAGE CONTROL ADDRESS
SIZE OF MESSAGE CONTROL ADDRESS
NUMBER OF CHARACTERS IN ONE MESSAGE
TOP OF PC TABLE CONTAINING MESSAGE

Table addresses	Display request	Message contents
#9216	#15000	SPiNDLE-ALARM
#9217	#15001	MO6 ERROR
#9218	#15002	TAPPiNG ERROR
#9219	#15003	
#9229	#15015	UNUSABLE.S-CODE
#9230	#15016	UNUSABLE M-CODE
#9231	#15017	PARAMETER ERROR

APSH #1500	...	Message data control address
IPSH 1	...	Size of message control address
IPSH 32	...	Number of characters of one message data
TPSH 9216	...	Top of PC table containing message.
SUBP 023	...	DISP instruction

③ Control conditions

(a) Designation of message control address (APSH #xxxx)

Designate the head address that request the message.

(b) Designation of size of message control address (IPSH xx)

Designate the size (number of bytes) of message control address.

For example, when the message control address is designated as APSH #1500 if IPSH 1 is specified, continuous 4 bytes from #1500 are used, and if IPSH 2 is specified, continuous 8 bytes from #1500 are used.

Note: Up to 16 types of messages are available when IPSH 1 is specified.

(c) Designation of number of characters per message (IPSH xx)

The number of characters for each message varies. Designate the maximum number of characters in the PC table to be used.

(d) Designation of top number of PC table containing message (TPSH xxxx)

④ DISP instruction use example

When contacts AL1 - AL4 are set on, the message corresponding to the request bits are displayed on the CRT screen, and deceleration stop is performed. The display goes out when a reset signal is given.

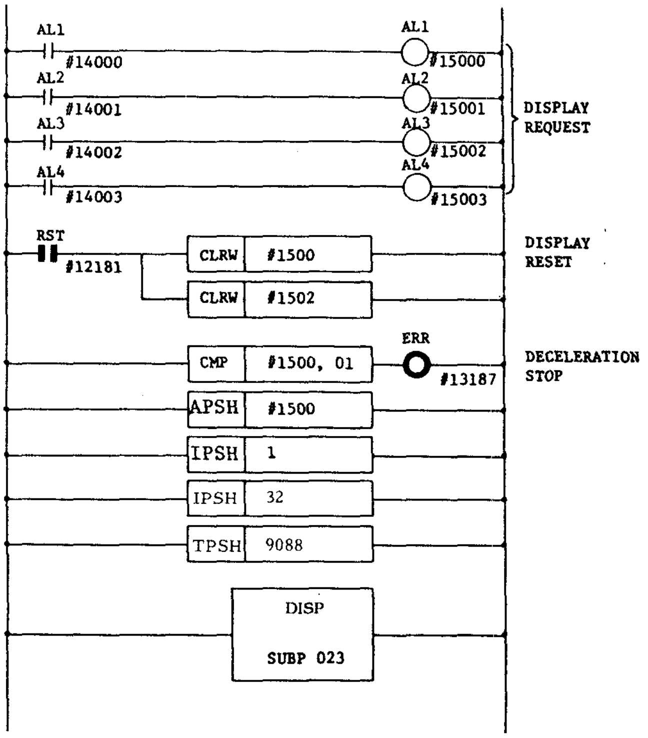


Table addresses	Display request	Message contents
#9088	#15000	PARAMETER ERROR
#9089	#15001	SPiNDLE SERVO ALARM
#9090	#15002	MO6 ERROR
#9091	#15003	KEY-LOCK ERROR

⑤ Improving USERS MESSAGE function (J50M only)

This fucction displays messages on NC CRT screen from PC input signals having operation mistakes or machine defects.

The following messages are displayed:

(i) Regarding ERROR code and ERROR contents.

(ii) Showing machine operation condition.

(iii) Showing operation procedure, etc.

These messages can be displayed in NC USERS MESSAGE screen.

There is no distinction between the ways of displaying messages for easy operation.

USERS MESSAGE display selection

USERS MESSAGE display is selected by the following operation:

① The established USERS MESSAGE 1 display is selected by depressing ALM key to select alarm display.

② Added USERS MESSAGE 2 display is selected by depressing ALM key again.

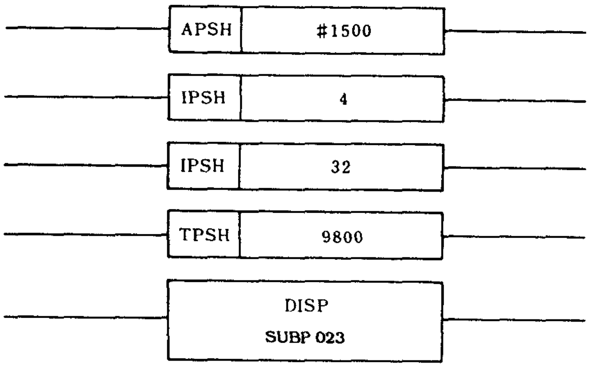
③ Depressing the ALM key again calls up USERS MESSAGE 1.

MESSAGE 1.

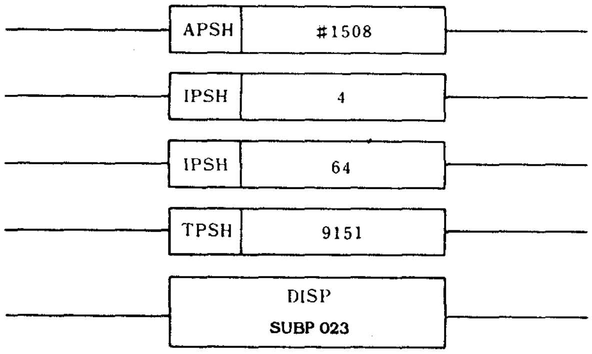
a. MESSAGE DISPLAY instruction

Two SUBP023s can be used on the ladder.

First SUBP023



Second SUBP023



SUBP023 which has been used first on the ladder is displayed under the title of USERS MESSAGE 1 on the message screen (USERS MESSAGE 1).

Depress ALM key, and SUBP023, which has been used later, is displayed under the title of USERS MESSAGE 2 on the message screen (USERS MESSAGE 2).

7.7 MACRO INSTRUCTIONS (Cont'd)

By depressing ALM key again, the display is reverse displayed to USERS MESSAGE 1 from USERS MESSAGE 2.

Note: USERS MESSAGE 1 has only on display. By depressing PAGE key the previous display is called up.

b. Display specifications

Number of characters in a message and message types.

16 characters	* 220 types (Max.)
32 characters	* 128 types (Max.)
64 characters	* 64 types (Max.)

For two SUBP023s, the same characters can be used. In this case, however, the total number of the message types of two SUBP023s should be less than the maximum of each message.

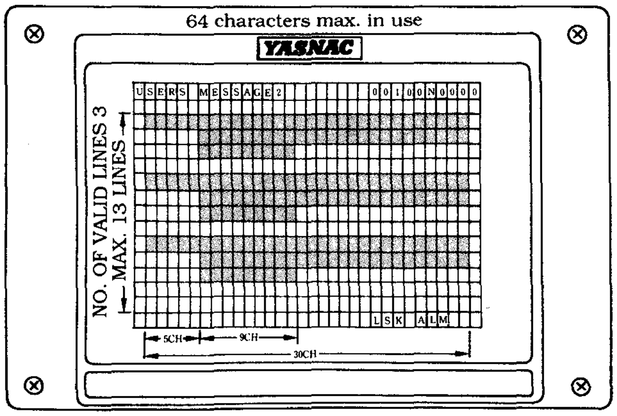
Display table

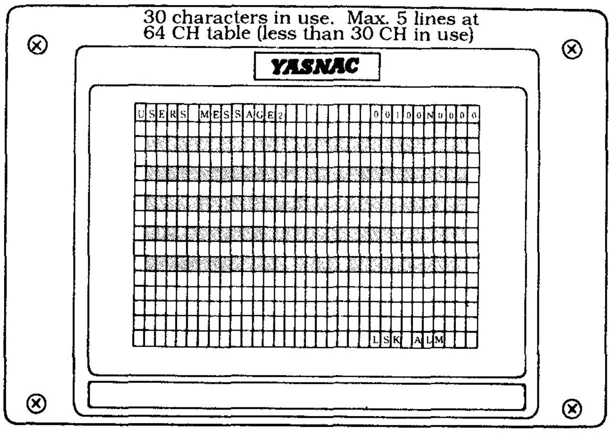
64 character
64 addresses between 9024 and 9087.
32 character
128 addresses between 9088 and 9215.
16 character
220 addresses between 9216 and 9435.

USERS MESSAGE 2 display range

Simultaneous display range

Valid width	30 characters
Valid lines	10 lines
Valid No. of message ...	3 to 5 types within the range of valid 10 lines or less.




30 characters in use. Max. 5 lines at 64 CH table (less than 30 CH in use)
YASNAC
USERS MESSAGE 2
0010 0N000
LSK ALM

Note: When the table shown above is used for another SUBP023, range of display table is decreased.

When the display table is used for another SUBP02 for other purposes, max. display type is limited available table capacity.

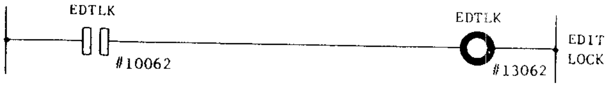
When making a table, put "SPACE" if necessary.

Characters under "FF" are disregarded.

8. SEQUENCE PROGRAM EXAMPLE

8.1 SERIES CONNECTION

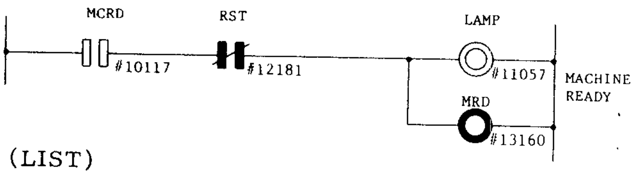
(1)



(LIST)

```
LD        #10062
OUT       #13062
```

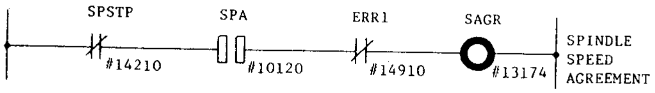
(2)



(LIST)

```
LD        #10117         OUT       #11057
AND-NOT #12181           OUT       #13160
```

(3)

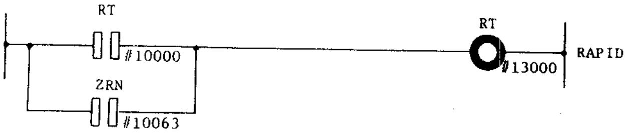


(LIST)

```
LD-NOT    #14210         AND-NOT #14910
AND       #10120         OUT       #13174
```

8.2 PARALLEL CONNECTION

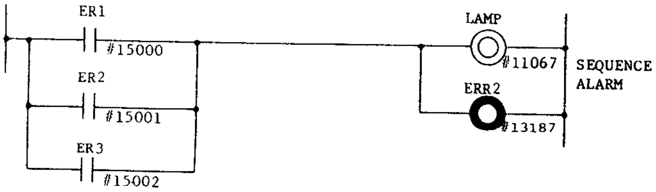
(1)



(LIST)

```
LD        #10000
OR        #10063
OUT       #13000
```

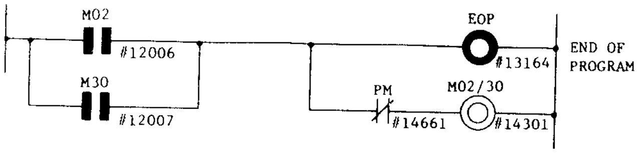
(2)



(LIST)

```
LD        #15000         OUT       #11067
OR        #15001         OUT       #13187
OR        #15002
```

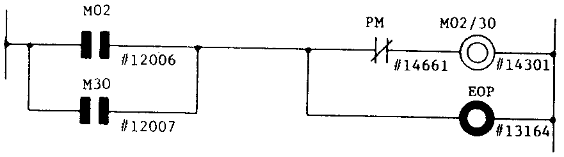
(3)
(a)



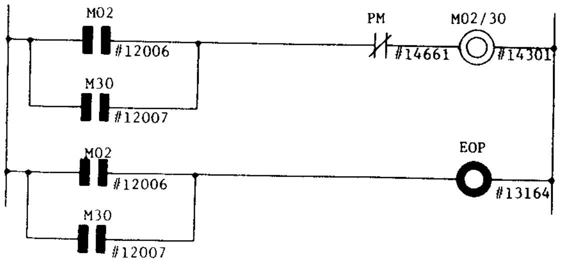
(LIST)

```
LD        #12006         AND-NOT #14661
OR        #12007         OUT       #14301
OUT       #13164
```

(b)

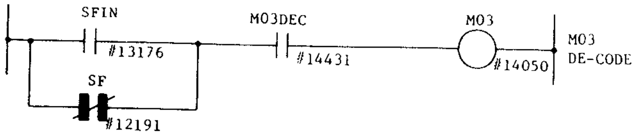


Note: In this program, coding cannot be made. Make a sequence as described in (3) a, or change the ladder as follows.



8.3 SERIES AND PARALLEL CONNECTION

(1)
(a)

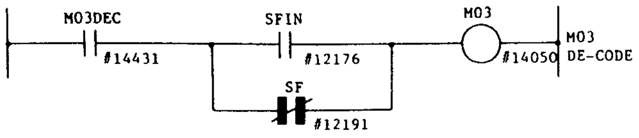


(LIST)

```
LD        #13176         AND       #14431
OR-NOT    #12191         OUT       #14050
```

8.3 SERIES AND PARALLEL CONNECTION (Cont'd)

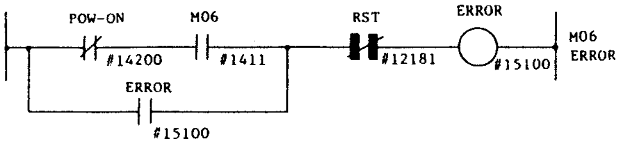
(b)



(LIST)

```
LD        #14431        AND-STR
STR       #13176        OUT       #14050
OR-NOT    #12191
```

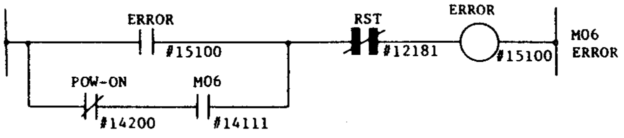
(2)
(a)



(LIST)

```
LD-NOT    #14200        AND-NOT #12181
AND       #14111        OUT       #15100
OR        #15100
```

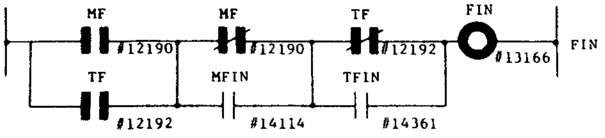
(b)



(LIST)

```
LD        #15100        OR-STR
STR-NOT #14200          AND-NOT #12181
AND       #14111        OUT       #15100
```

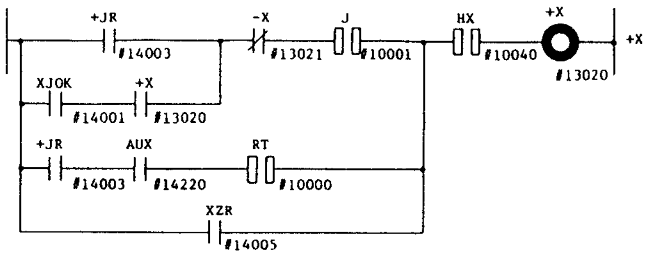
(3)



(LIST)

```
LD        #12190        ATR-NOT #12192
OR        #12192        OR        #14361
STR-NOT #12190          AND-STR
OR        #14114        OUT       #13166
AND-STR
```

(4)

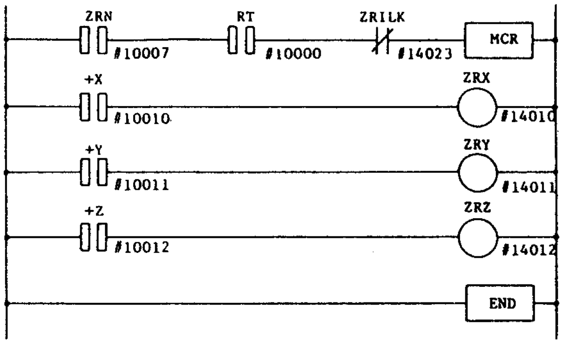


(LIST)

```
LD        #14003        AND       #14220
STR       #14001        AND       #10000
AND       #13020        OR-STR
OR-STR                  OR        #14005
AND-NOT #13021          AND       #10040
AND       #10001        OUT       #13020
STR       #14003
```

8.4 MASTER CONTROL RELAY APPLICATIONS

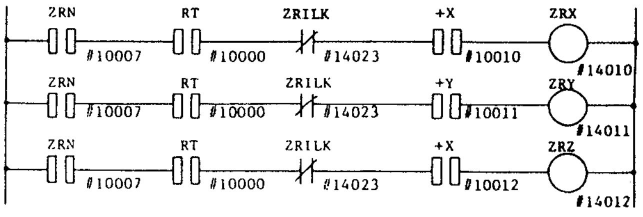
(1)



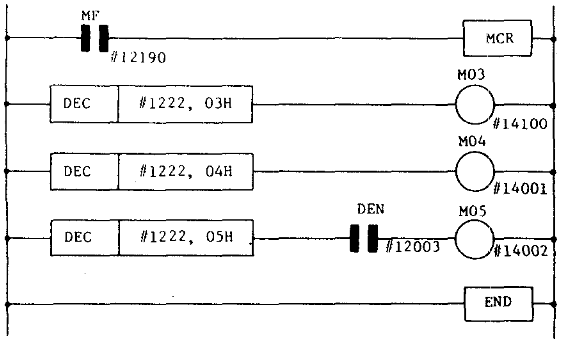
(LIST)

```
LD        #10007        LD        #10011
AND       #10000        OUT       #14011
AND-NOT #14023          LD        #10012
MCR                     OUT       #14012
LD        #10010        END
OUT       #14010
```

The above ladder has the same meaning as that of the ladder below.



(2)



(LIST)

```
LD     #12190              OUT    #14101      ; M04
MCR                        DEC    #1222, 05H
DEC    #1222, 03H          AND    #12003
OUT    #14100      ; M03   OUT    #14102      ; M05
DEC    #1222, 04H          END
```

This is the code detection ladder for M code. By use of MCR, ladder can be completed without inserting MF in each M code.

9. SEQUENCE PROGRAM ONLINE EDITING SYSTEM

This section describes the functions provided by a "sequence program editor (JDU01)" in temporary connection with the NC unit YASNAC J50L or J50M, together with the operating procedures for the editor.

The functions of the sequence program editing system fall into three major categories:

(1) Editing Sequence Programs

To erase, alter and insert commands from, in and to sequence program.

(2) Providing Hard-copy of Edited Sequences Programs

To punch a sequence program onto a tape and transfer data to P-ROM writer.

(3) Checking Edited Sequence Programs

To check a sequence program in C-MOS and another program written in P-ROM through execution.

The following paragraph discuss the functions and operating procedures in detail.

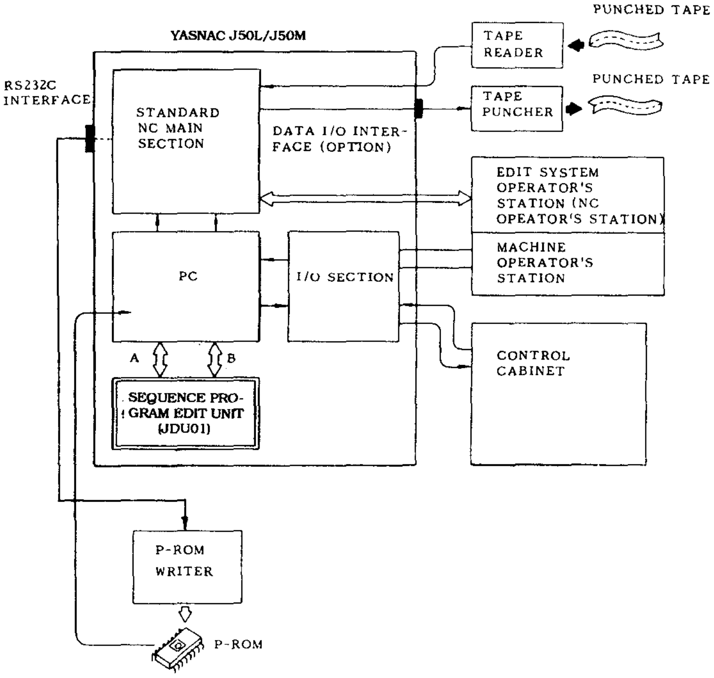


Fig. 9.1 Block Diagram of Sequence Program Edit System

9.1 BLOCK DIAGRAM OF SEQUENCE PROGRAM EDIT SYSTEM

Figure below shows the hardware constitution of sequence program edit system.

9.1 BLOCK DIAGRAM OF SEQUENCE PROGRAM EDIT SYSTEM (Cont'd)

(1) The sequence program editor (JDU01) should be mounted on the CPU rack in the NC unit before being wired.

(2) To operate a sequence program editing system, use the NC operator panel with a CRT as an operator panel for the editing system.

(3) A tape reader is used to load into sequence program editor memory a list tape with a sequence ladder coded in it or a P-ROM format tape written in machine language.

(4) A tape puncher is used to punch out the final sequence program that was edited and checked on a list tape or P-ROM format tape.

(5) A commercially available P-ROM writer can be connected to the NC RS232C interface to write the final sequence program into P-ROM.

9. 2 SEQUENCE PROGRAM EDITOR (JDU01)

(1) The name and the type of the sequence program editor are as follows:

Name: Sequence Program Editor

Type: JZNC-JDU01

External view of the JDU01 is shown in Fig. 9.2.

(2) The JDU01 has a C-MOS memory backed up by battery. It can store up to a 128 k-byte sequence program to be edited. The stored sequence program is on the level of the P-ROM format in machine language.

(3) JDU01 components along with their functions are listed below.

(a) Two mounting holes with hooks :
Mounts the JDU01 with attached hook on the CPU rack in the NC unit.

(b) CNAI (120 core) and CNF (80 core) connector :
: Supplies power (+5 V) to the JDU01
: Used to connect the NC main section with the PC section.

(c) ROM/RAM select switch :
Selects P-ROM in the P.C. system or C-MOS memory in the JDU01 for operation or controlling.

9. 3 CONNECTING SEQUENCE PROGRAM EDITOR

Follow the steps given below to connect the JDU01.

(1) When the JDU01 is mounted on the NC CPU rack, NC CPU unit power should be turned OFF.

(2) Mount the ROM (No. 40) on the JSD board from the PC50 board.

(3) Replace the PC50 board with the JSD board.

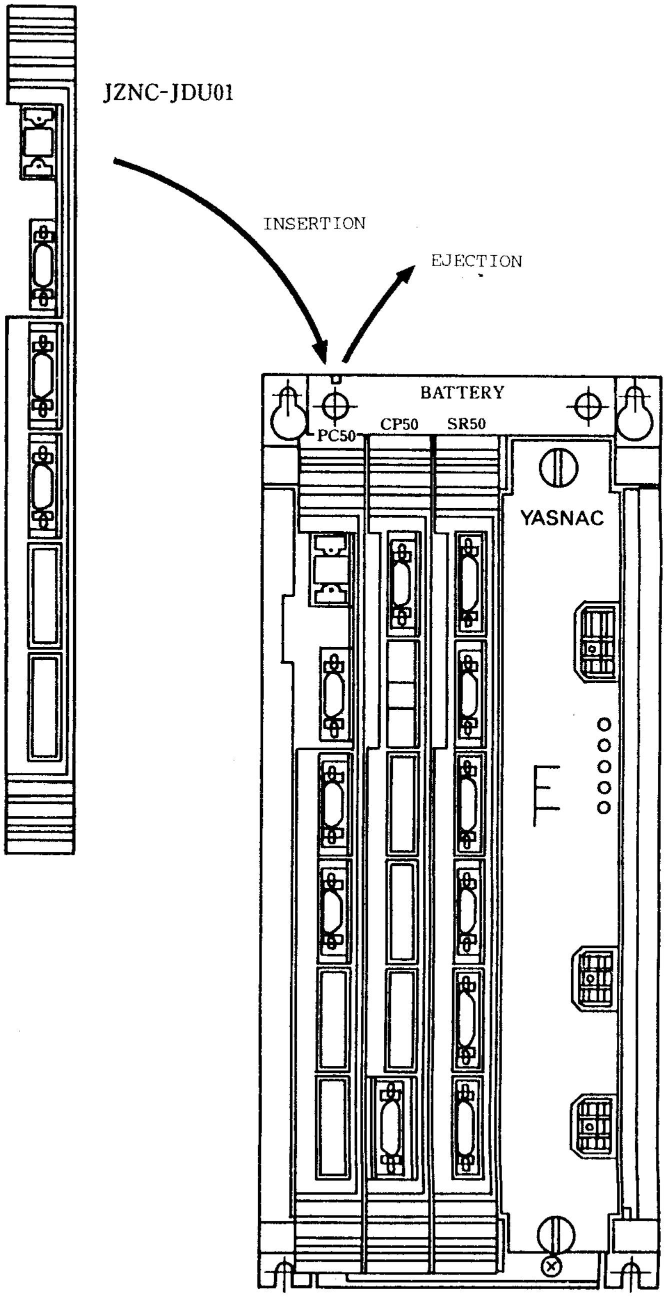


Fig. 9.2 CPU Rack

9.4 EDIT SYSTEM OPERATOR'S STATION

The NC operator's station with CRT is used for sequence program editing, when used as a sequence program editing unit.

Fig. 9.3 shows the NC operator's station respectively for YASNAC J50L and J50M.

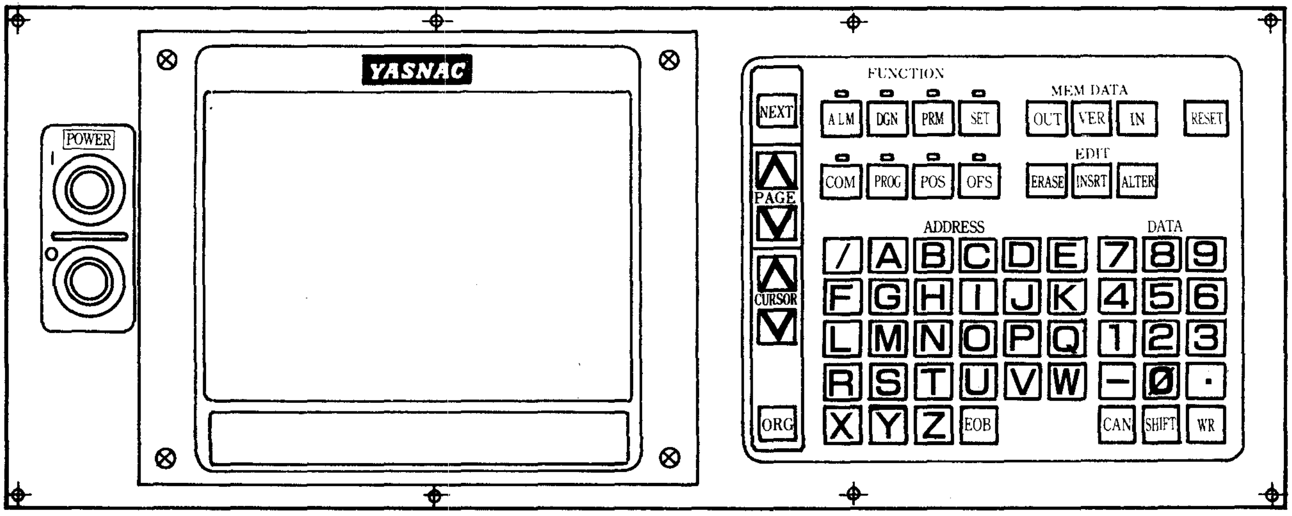


Fig. 9.3 Operator's Station for J50L/J50M

(1) POWER ON/OFF Pushbuttons

- POWER ON pushbutton

  To turn on the power for the control: Depress the pushbutton first to turn on the control power and depress it again to turn on the servo power. (Push this button to recover the servo power after an emergency stop.)

- POWER OFF pushbutton

  To turn off the power for the control: Depress it to turn off both the servo and control powers.

(2) DATA Key

For 0 to 9, data keys of 0 to 9 are used. For hexadecimal A to F, address keys of A to F are used. Commands and address input can be made by using address keys.

(3) CAN (cancellation) key:

For cancellation of the input data.

(4) WR (write) key:

For storing the input data into buffer storage.

(5) CURSOR Keys

The CURSOR control key is used to move the cursor. It is used to start address search.

- Depressing CURSOR ▽ key moves the cursor forward.

- Depressing △ CURSOR key moves the cursor backward.

- Keeping the cursor control key depressed makes the cursor move automatically forward or backward.

(6) PAGE Keys

Depressing the PAGE ▽ key increases the editing page by one. Depressing the △ PAGE key moves the cursor backward.

(7) NEXT Key (Function Mode Select Keys)

Depressing the NEXT key increases the function mode number by one. Mode 6 changes to mode 1 by depressing the NEXT key. For details of mode 1 to 6, refer to par. 9.5.

(8) ERS , INS , ALT , and EOB Keys

(a) ERS key:
For erasure of a block of data in a sequence program.

(b) INS key:
For insertion of a block of data in a sequence program

(c) ALT key:
For alteration of a block of data in a sequence program

9.4 EDIT SYSTEM OPERATOR'S STATION (Cont'd)

(d) EOB key:
For storing a block of data in a sequence ladder. The block stored using the EOB key will be the last block in a sequence program.

(9) IN , VER , and OUT Keys

(a) IN key:
To start storing data on paper tape into memory through tape reader.

(b) VER key:
To start verifying between memory data and punched tape data.

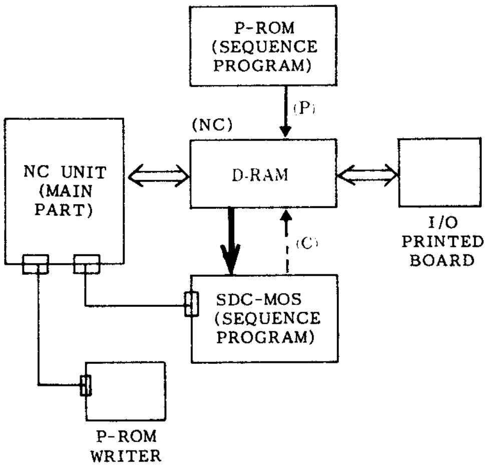
(c) OUT key:
To start outputting various data in memory through data I/O interface.

(d) RESET key:
To return the editing pointer to the head of sequence ladder. Also used for releasing alarm codes if their causes are eliminated.

9.5 FUNCTION MODE OF EDIT SYSTEM

When the control unit is used as a sequence program unit, four function modes can be selected. Use the NEXT key for mode selection.

J50L/J50M PC System Structure



(1) JDU01 board ROM/RAM select switch

ROM : From P-ROM }
RAM : From C-MOS } Transfer at power ON

(2) ⟶ : Stores the edited D-RAM data in C-MOS of JDU01 board. (See (4) in the column of MODE 4.)

Table 9.1 List of Function Modes and Functions

Function Mode No.	Function Mode	Function
Mode 1	Edit mode (LADDER EDIT)	·Alteration, insertion, and deleting sequence programs, address search, and writing by MDI. ·Storing, collating, and punching out of P-ROM former tape.
Mode 2	List tape mode (SOURCE TAPE)	·Storing, collating, and punching out of list tape.
Mode 3	PROM writer mode (ROM WRITER)	·Transferring sequence programs to P-ROM writer.
Mode 4	Parameter mode (PARAMETER)	(1) Registration of version number (2) Registration of tape comments (3) Setting Baud rate (4) Transfer of DRAM to C-MOS (5) Transfer of P-ROM to DRAM or C-MOS to DRAM. (6) P-ROM type selection (7) Resetting of edit area (8) Returning to NC mode (9) I/O device selection
Mode 5	PC data edit mode (PC TABLE EDIT)	(1) Editting of PC table and address searching (2) Storing, collating, and punching-out of P-ROM format tape
Mode 6	Address check mode (ADDRESS CHECK)	Checking for address duplication in sequence program.

9.6 HOW TO ENTER EDITING SYSTEM MODE

Given below are the EXIT STEPS to leave the NC system mode (NC Mode), and to enter the editing system mode (SD mode) in which the device is used as sequence program editing system. After switchover to the SD mode, the device permits operations described in par. 9.7 through 9.11.

9.6.1 When NC Unit is in Offline State (System NO. 6 → SD MODE)

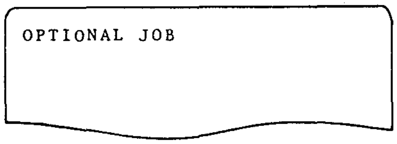
The NC unit in the offline state is an NC unit that cannot operate in the NC mode upon power-on, with no sequence program stored in PC P-ROM or JDU01 C-MOS.

Switching from the offline state to the SD mode requires the following operations, provided that the JDU01 has been connected as explained in par. 9.3 :

(1) Set the System No. switch to 6.

(2) Snap the ROM/CMOS select switch to RAM on the JDU01.

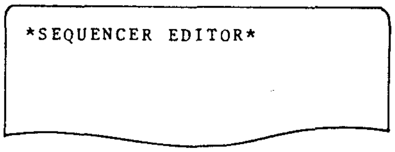
(3) Depress the POWER ON pushbutton to apply power. A comment "OPTIONAL JOB" will appear on the CRT.



OPTIONAL JOB

Fig. 9.4

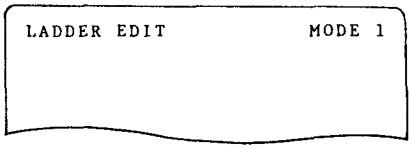
(4) Deress the X, S and D keys, in that order. Then depress the ORG key. A comment "SEQUENCER EDITOR" will appear on the CRT.



SEQUENCER EDITOR

Fig. 9.5

About 2 seconds later, MODE 1 of the SD mode is entered.



LADDER EDIT MODE 1

Fig. 9.6

(5) Then operate the PAGE keys to select one of six MODEs in the SD mode.

Note: Generally, the parameter mode of MODE 4 is later entered to clear the edit area, followed by the storing of the list tape in the list tape mode of MODE 2. For more details, refer to par. 9.14, "OPERATING PROCEDURE."

9.6.2 When NC Unit isin Online State (System NO.4 → SD MODE)

The NC unit in the online state is an NC unit that can operate in the NC mode upon power-on, with the sequence program stored in P-ROM or C-MOS.

Switching from this online state to the SD mode requires the following operations, provided that the JDU01 has been connected as explained in par. 9.3 :

(1) When the sequence program is stored in P-ROM, snap the ROM/RAM select switch to ROM on the JDU01. Set the switch to C-MOS for the program stored in C-MOS.

(2) Depress the POWER ON pushbutton to apply power (set the System No. switch to 0 or 4 beforehand). The NC mode will be entered.

(3) When a test run is performed here for sequence program check, stop all NC functions by Feed Hold or other operations and press the RESET key afterward.

(4) Set the System No. switch to 4.

(5) Depress the DGN function key, and depress the NEXT key. A comment "(STORED)" will appear following another comment "DIAGNOSIS" on the CRT.

(6) Depress the X, S and D keys, in that order. Then depress the ORG key. A comment "SEQUENCER EDITOR" will appear on the CRT (Fig. 9.5). About 2 seconds later, MODE 1 of the SD mode is entered (Fig. 9.6).

(7) Then operate the PAGE keys to select one of six MODEs in the SD mode.

NOTE

1. The NC unit in the online state can enter the SD mode by the following parameters. #6030D1 = 1 for J50M. #6030D7 = 1 for J50L.
2. After switchcover from the online state to the SD mode, the PC output signals remain as they were just before the SD mode was entered.

   Example:
   A flashing PC output signal remains on when SD mode is selected during on state.
3. The minimum condition for the SD mode to be entered by the above steps is that "RTH" (end command of high-speed sequence program) and "RET" (end command of sequence program) have been written in P-ROM or C-MOS.

9.7 EDITING MODE (MODE 1)

This mode permits the following operations:

(1) After, insert, erase, and address search operation on sequence programs.

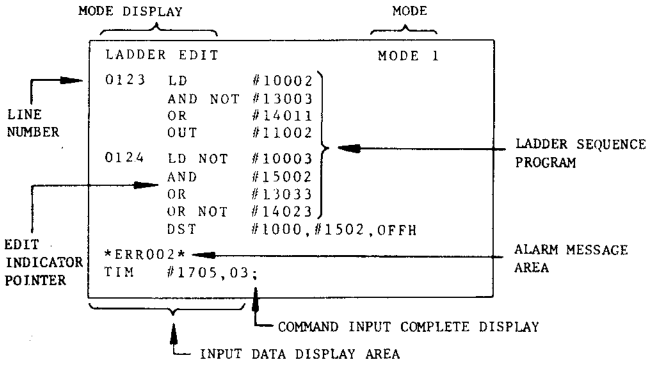
(2) MDI write operation on sequence programs.

(3) Loading, verifying and punching out P-ROM format tapes.

9.7.1 Sequence Program Editing

(1) CRT display in MODE 1

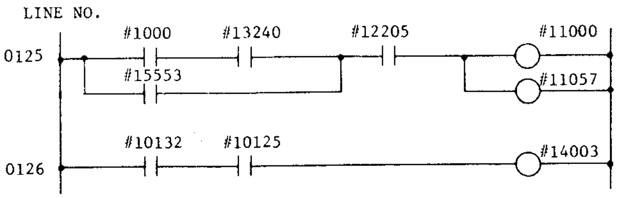
(a) As shown below, 10 lines of a sequence program stored in C-MOS are displayed in MODE 1. A blank line is counted as one line.



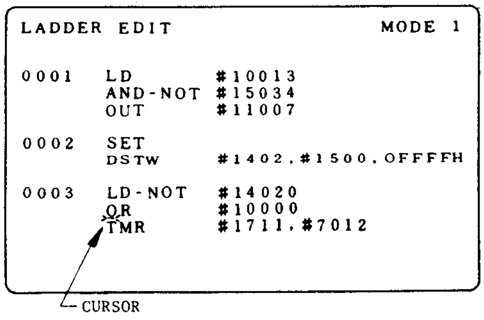
```
LADDER EDIT                  MODE 1
0123  LD          #10002
      AND NOT     #13003
      OR          #14011
      OUT         #11002
0124  LD NOT      #10003
      AND         #15002
      OR          #13033
      OR NOT      #14023
      DST         #1000,#1502,OFFH
*ERR002*
TIM   #1705,03;
```

Fig. 9.7

(b) A line number is a serial number attached to a closed circuit group beginning with a contact input command and ending with a contact output command.



(c) A cursor is positioned to the command to be edited. See the next paragraph "Address search function" for how to specify the cursor.



```
LADDER EDIT                  MODE 1

0001  LD          #10013
      AND-NOT     #15034
      OUT         #11007

0002  SET
      DSTW        #1402.#1500.OFFFFH

0003  LD-NOT      #14020
      OR          #10000
      TMR         #1711.#7012
```

Fig. 9.8

Note: If MODE 1 of the SD mode is entered from the System No. switch at [6], an error comment "*DISASSEMBLE*" will appear on the CRT because no sequence program is currently stored. In this case, enter the parameter mode of MODE 4 and clear the edit are ((6) in par. 9.10) to reset the error comment. Commands "RTH" and "RET" will appear on the CRT. Then normal edit operations are possible.

(2) ADDRESS SEARCH

Address Search searches the commands or line to be edited. The searching procedure is as follows.

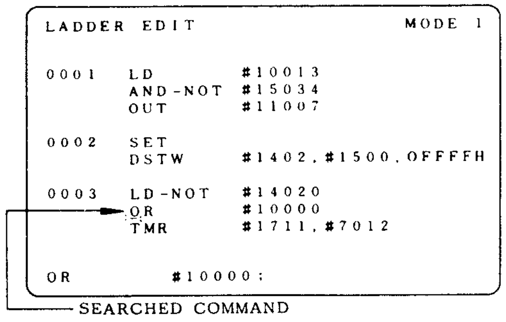
(a) Key in the commands to be searched

Keying in "0," "R," "WR," "1," "0," "0," "0," "0," through the keyboard causes OR #10000; to display at the bottom of the CRT screen.

(b) Depress the [CURSOR] key.

Search starts. When the search is completed, ten-line commands including the searched command will be displayed on the CRT screen.

(c) If the keyed-in command cannot be found, "*ERR008*" will be shown on the CRT screen. Release the alarm code by depressing [CAN] or [RESET] key.



```
LADDER EDIT                  MODE 1

0001  LD          #10013
      AND-NOT     #15034
      OUT         #11007

0002  SET
      DSTW        #1402.#1500.OFFFFH

0003  LD-NOT      #14020
      OR          #10000
      TMR         #1711.#7012

OR          #10000;
```

CURSOR indicates the searched command.

Fig. 9.9

Note:

1. The command can be searched by keying-in the part of the command data.

   Example: For DST #1200, #1100, FF commands keying-in "D," "S," "T," "WR" can search the DST commands regardless of #1200, #1100, and FF.

2. Address search can be done by using only one address

   Example: For DST #1200, #1100, FF commands, keying-in "#" "1," "2," "0," "0," "WR" can search the commands which use #1200 regardless of DST, #1100, and FF.

3. Address search can be done continuously.

   Searching can be continued if [CURSOR] key is pressed again after address search.

   Depress [CAN] key to quit searching.

4. When the data to be searched is near the CURSOR , use the CURSOR key to reach the required data.

(3) Key input operations

Below are the steps to key in commands and display them at bottom left on the CRT screen for editing or address search.

(a) Press the ADDRESS keys to sequentially key in the alphabetics of the commands to be entered.

Example:

(Command)	(Key-in operations)
SET	S , E , T
OR	O , R
AND-STR	A , N , D , – , S , T , R

Use the Minus key instead of the Hyphen key.

Alphabetic strings will appear at bottom left of the CRT screen.

(b) Depress the WR key.

i. For commands not requiring address numbers (SET, END, etc.), a semicolon (;) is displayed after each to complete the key-in operation.

ii. For commands requiring address numbers (OR, MOV, etc.), a symbol "#" is displayed after each to prompt further entry.

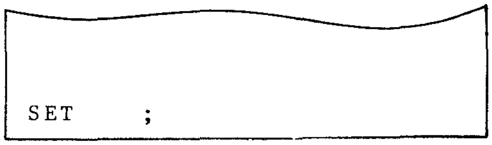
iii. Entering an alphabetic string other than the commands causes a comment "*ERR01*" to appear on the CRT. This is reset by depressing the CAN or RESET key.

(c) Key-in address numbers (followed by bit numbers if necessary). For commands requiring one address number (e.g., OR), entering the required number of digits causes a semicolon (;) to appear automatically after each number, thus completing the key-in operation.

(d) Press the WR key. For commands requiring two address numbers (e.g., MOV), symbols ",#" will automatically appear after entry of the first number.

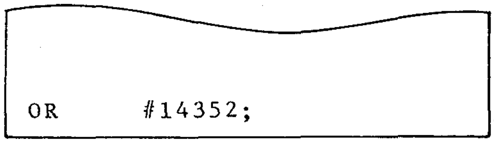
(e) Key in the next address number, and the number will be displayed.

(f) Press the WR key. A semicolon (;) will be displayed to complete the key-in operation. If an inadvertent key is pressed in each section explained above, press the CAN key and then press the correct key.



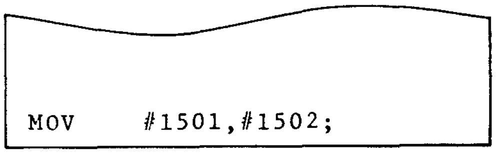
```
SET    ;
```

Fig. 9.10



```
OR      #14352;
```

Fig. 9.11



```
MOV     #1501,#1502;
```

Fig. 9.12

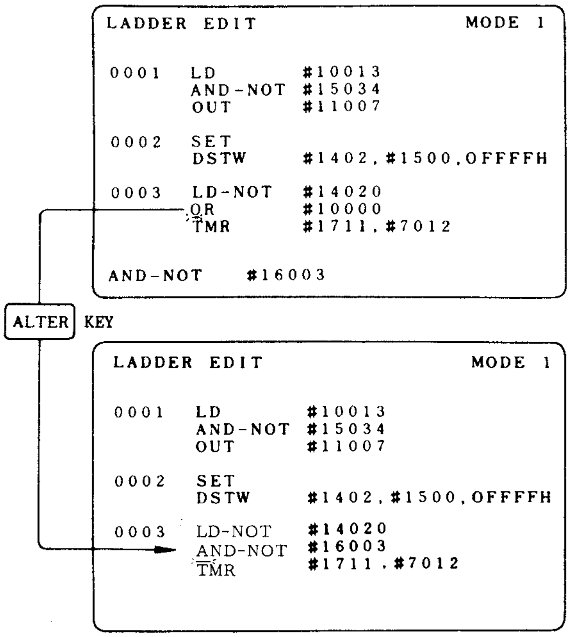
The above procedure covers most of the commands, with only a few differences for some. In any case, a semicolon (;) appearing at the end of the entered data indicates the end of the key-in operation. On the data thus keyed in, address search and editing functions by the INSRT , ALTER and ERASE keys are available.

(4) Edit Operation (ALTER , INSRT , ERASE)

The command specified by the cursor can be altered, inserted or erased.

(a) Alter operation

Depress the ALTER key. The command specified by the cursor will be erased and replaced by the command just entered. After alteration, the command that replaced the old one remains specified.



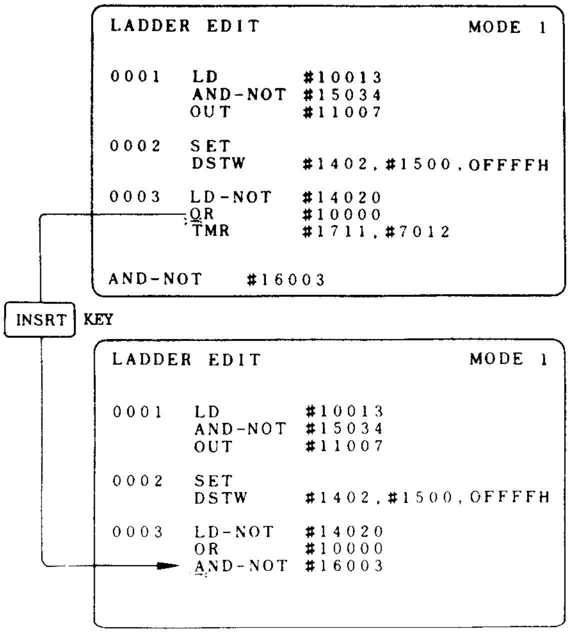
```
LADDER EDIT                    MODE 1

0001  LD        #10013
      AND-NOT   #15034
      OUT       #11007

0002  SET
      DSTW      #1402,#1500,OFFFFH

0003  LD-NOT    #14020
      OR        #10000
      TMR       #1711,#7012

AND-NOT   #16003
```

```
LADDER EDIT                    MODE 1

0001  LD        #10013
      AND-NOT   #15034
      OUT       #11007

0002  SET
      DSTW      #1402,#1500,OFFFFH

0003  LD-NOT    #14020
      AND-NOT   #16003
      TMR       #1711 .#7012
```

Fig. 9.13

(b) Insert operation

Press the INSRT key. The command just entered will be inserted following the command specified by the cursor.
After insertion, the command just inserted remains specified.



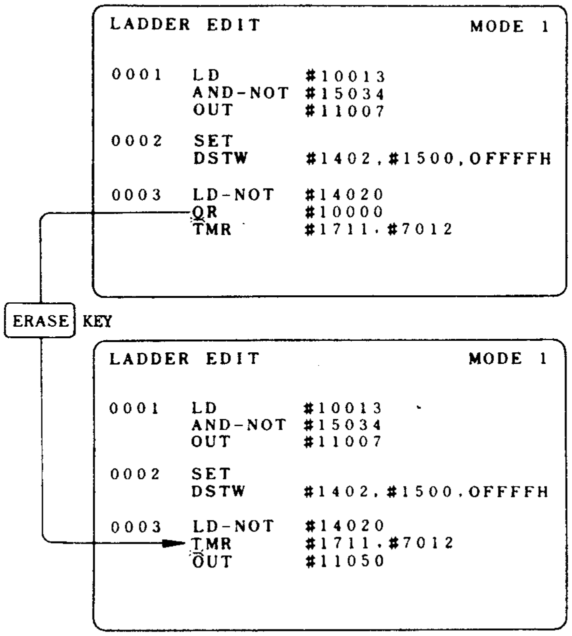
```
LADDER EDIT                    MODE 1

0001  LD        #10013
      AND-NOT   #15034
      OUT       #11007

0002  SET
      DSTW      #1402,#1500,OFFFFH

0003  LD-NOT    #14020
      OR        #10000
      TMR       #1711,#7012

AND-NOT   #16003
```

```
LADDER EDIT                    MODE 1

0001  LD        #10013
      AND-NOT   #15034
      OUT       #11007

0002  SET
      DSTW      #1402,#1500,OFFFFH

0003  LD-NOT    #14020
      OR        #10000
      AND-NOT   #16003
```

Fig. 9.14

(c) Erase operation

Press the ERASE key. The command specified by the cursor will be erased.
After erasure, the command following the erased command is specified.



```
LADDER EDIT                    MODE 1

0001  LD        #10013
      AND-NOT   #15034
      OUT       #11007

0002  SET
      DSTW      #1402,#1500,OFFFFH

0003  LD-NOT    #14020
      OR        #10000
      TMR       #1711,#7012
```

```
LADDER EDIT                    MODE 1

0001  LD        #10013
      AND-NOT   #15034
      OUT       #11007

0002  SET
      DSTW      #1402,#1500,OFFFFH

0003  LD-NOT    #14020
      TMR       #1711,#7012
      OUT       #11050
```

Fig. 9.15

(5) Low-speed processing sequence program division

When the edit operation of sequence program is completed in the edit mode, the sequence program should be divised for low speed processing.

Depress the RESET key, and then ORG key with MODE 1. The programs are automatically divided for low-speed processing and number of section count is indicated.

9.7.2 MDI Write Operation on Sequence Program

In MODE 1, a sequence program can be written by MDI key-in operations from the beginning. The write operations are as follows:

(1) Operate the NEXT to select MODE 4.
Clear the edit area.
For the details, refer to par. 9.10 (7).

(2) Operate the NEXT key to return to MODE 1. This operation returns the cursor to the beginning of memory. Commands "RHT and "RET;" will appear on the CRT.

(3) Key in the desired command by the operation of par. 9.7.1 (3) on page 51.

(4) Depress the INSRT key, and the command just keyed in will be inserted following the command specified by the cursor. The inserted command will be specified anew.

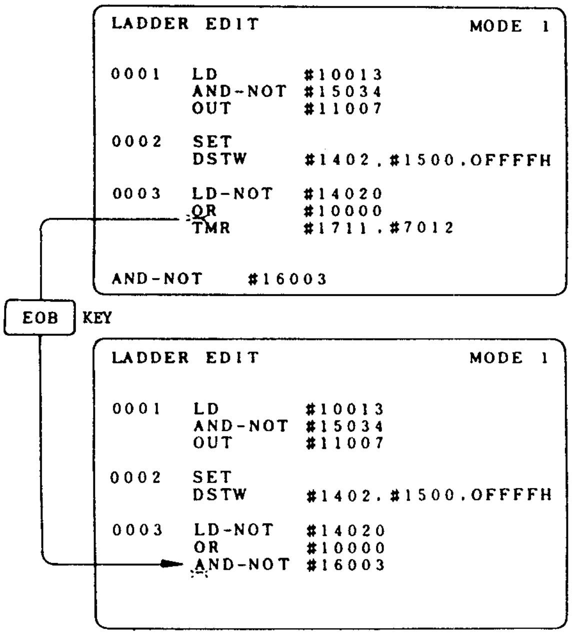
(5) Repeat the operations of (3) and (4) above to write the sequence program consecutively.

(6) Finally, depress the R, E," T and EOB keys, in that order, to complete the writing of the sequence program (RET = sequence program end command).

NOTE

1. Depressing the EOB key inserts the command just keyed-in following the command specified by the cursor, and erases all the subsequent commands. That is, the command stored by the EOB key becomes the last command of the sequence program at that time.
2. Consequently, in the edit operation of par. 9.7.1 (4), the EOB key can be used to erase all commands following a specific command (see Fig. 9.16).

   Depressing the EOB key inserts AND-NOT command after OR command and deletes all the commands stored after AND-NOT.

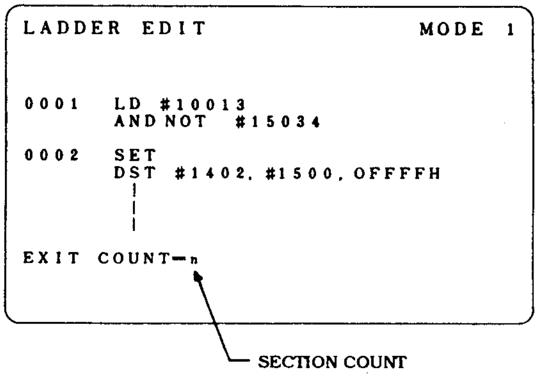


```
LADDER EDIT                         MODE 1

0001  LD        #10013
      AND-NOT   #15034
      OUT       #11007

0002  SET
      DSTW      #1402.#1500.OFFFFH

0003  LD-NOT    #14020
      OR        #10000
      TMR       #1711.#7012

AND-NOT    #16003
```

```
LADDER EDIT                         MODE 1

0001  LD        #10013
      AND-NOT   #15034
      OUT       #11007

0002  SET
      DSTW      #1402.#1500.OFFFFH

0003  LD-NOT    #14020
      OR        #10000
      AND-NOT   #16003
```

Fig. 9.16

3. Section count display function: Upon completion of a ladder sequence editing process, depress the RESET or ORG key to produce the section and CHECK SUM (total). Then the section count is displayed as shown below. CAN or RESET key can clear this.



```
LADDER EDIT                   MODE 1

0001   LD #10013
       AND NOT  #15034

0002   SET
       DST  #1402, #1500, OFFFFH
       |
       |
       |

EXIT COUNT—n
```

Fig. 9.17

4. Search function of section marked ****

   After finding the section count by keying ORG , the portions in the ladder where the section is inserted can be searched.

(a) Key-in *0 and then, SHIFT four times. The section count "n" (two digits) to be searched, and WR .

(b) Key-in CURSOR ⊠.

(c) When the search process has been completed, the sequence ladder for that portion is displayed. If *ERR.008* (search error) is displayed, clear it by depressing the RESET key.

9.7.3 P-ROM Format Tape Input/Output Function (IN, OUT)

MODE 1 permits a P-ROM format tape on the machine language level to be inputted, verified and punched out.

(1) Inputting P-ROM Format Tape (IN)

A sequence program stored in the form of P-ROM format tape is reedited.

(a) Set a P-ROM format tape on the tape reader.

(b) Depress the IN key. This will move the contents of the P-ROM format tape into PC50 RAM memory (edit area). If an inadvertent tape read operation or an erroneous entry is detected, *ERR003* is displayed on the CRT screen and the tape stops on an 16K-byte boundary. Although depressing the IN key again can reset the error and continue loading the tape contents, it is recommended to run the tape from the beginning. Should the error recur, the tape is not usable.

9.7.3 P-PROM Format Tape Input/Output Function ($\boxed{\text{IN}}$, $\boxed{\text{OUT}}$) (Cont'd)

(2) Punching Out P-ROM Format Tape ($\boxed{\text{OUT}}$)

An edited sequence program is punched out onto a P-ROM format tape.

(a) Connect the tape puncher (see NOTE 1) via the data I/O interface option of the NC unit.

(b) Depress the $\boxed{\text{RESET}}$ key and $\boxed{\text{ORG}}$ key orderly. The cursor will return to the beginning of the sequence program.

(c) Depress the $\boxed{\text{OUT}}$ key. The contents of PC50 RAM memory will be punched out onto a P-ROM format tape on the machine language level.

REMARKS:

i. To verify whether or not the contents are punched out correctly, continue the vertification of (2) above.

ii. A feed hole punch portion about 75 cm long is provided at the both ends of the tape.

NOTE

1. The storage devices and tape punchers for P-ROM format tapes and list tapes are designated by MODE4, FUNCTION 10.

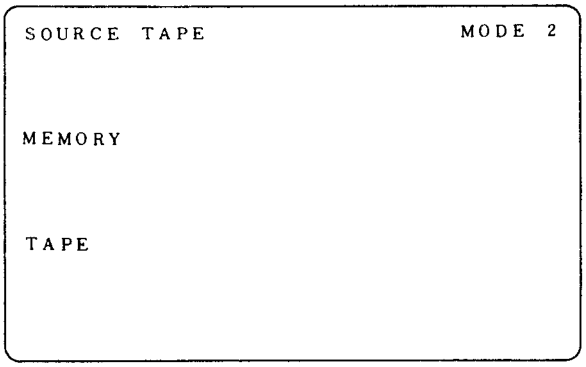
2. Storing data on P-ROM format tape is only about one tenth as bulky as that on list tapes. However, a list tape cannot be produced directly from a P-ROM format tape. This format is convenient for punching each substantial amount of data for storage.

9.8 LIST TAPE INPUT/OUTPUT MODE (MODE 2)

MODE 2 allows a list tape with a sequence ladder coded in PC instruction words to be loaded, verified and punched out.

(1) CRT Display in MODE 2

Operate the PAGE keys to select MODE 2, and the following screen will appear on the CRT:



```
SOURCE TAPE                              MODE 2


MEMORY


TAPE
```

Fig. 9.18

Note: SOURCE TAPE should be regarded as the same as LIST TAPE.

(2) List Tape Definition and Rules on List Tape Creation

(a) The list tape is defined as a punched tape with a sequence ladder coded in PC instruction words. See Fig. 9.19.

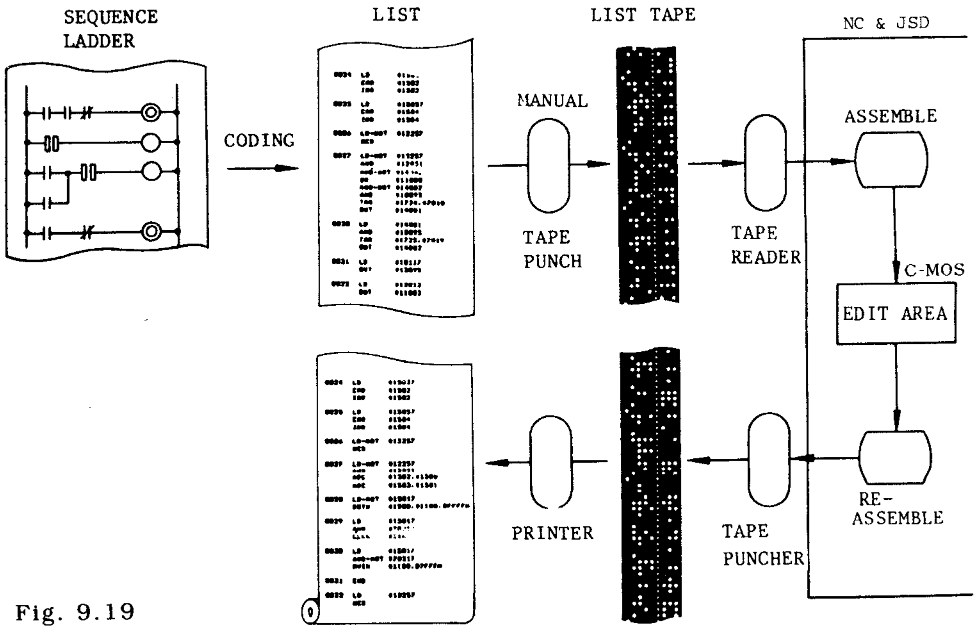


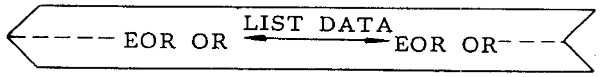
Fig. 9.19

(b) The rules for creating a list tape are as follows:

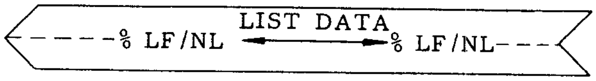
i. The list tape may be punched either in EIA or ISO code; the code is automatically identified when the tape is read in.

ii. The beginning and end of the list tape should be in the following format:

For EIA code



For ISO code



iii. The following rules should be observed in punching a list tape from a hand-written list (Fig. 9.20):

① Punching CR (or LF/NL) at the beginning of a line specifies a line feed.

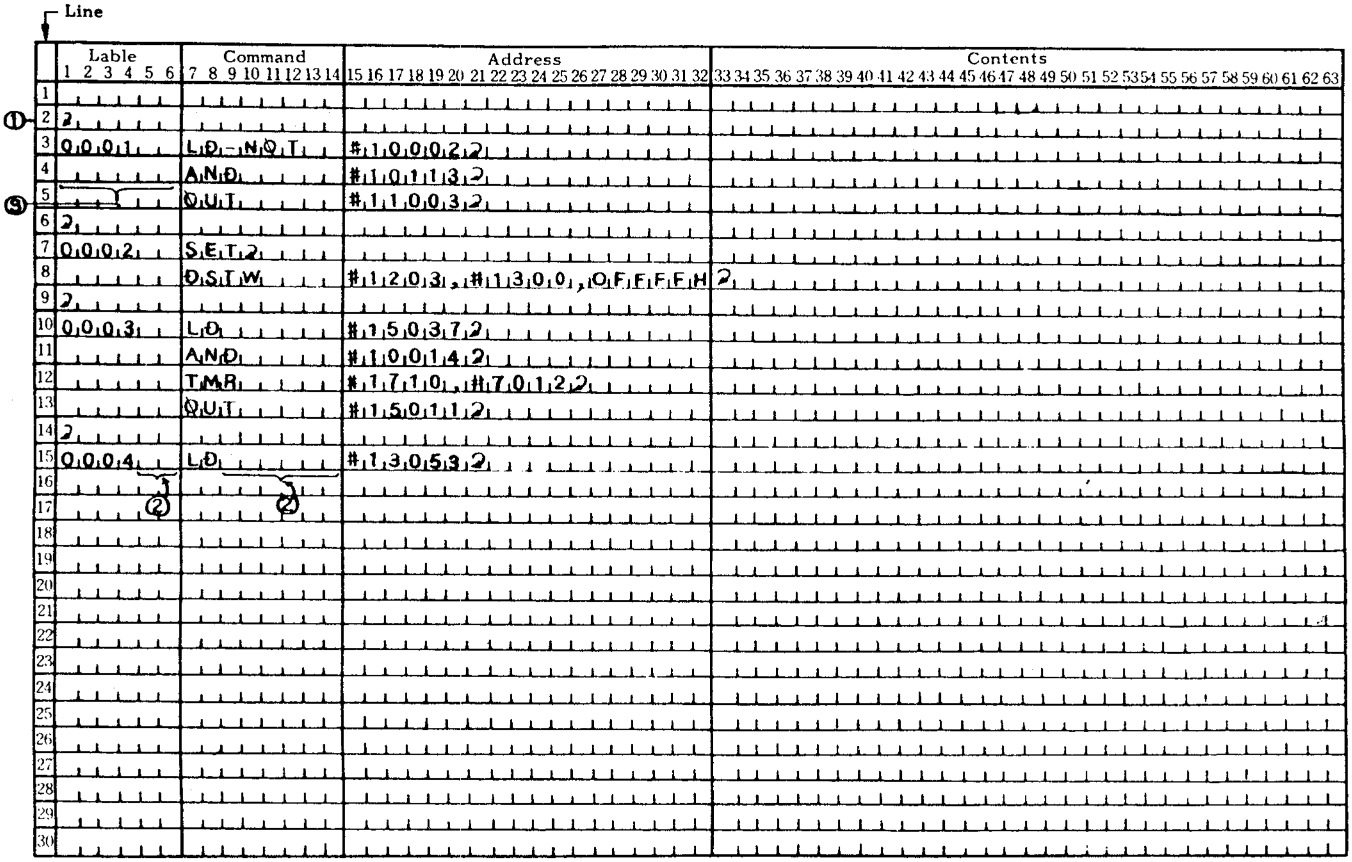
② All blanks must be filled with space code.

③ In a label part, punch a number (line No.) or space.

④ For PC table, follow the format in Fig. 9.21.

NOTE

Line numbers and comments are only for readability and are insignificant in assembling. The line numbers may or may not match those that were entered; The editor internally processes the line numbers regardless of the entered line numbers for display on the CRT and printing. No comments are stored in memory, nor are they displayed on the CRT or printed out. "#" is used for ISO code. "N" is used for EIA code.



Line	Lable 1 2 3 4 5 6	Command 7 8 9 10 11 12 13 14	Address 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32	Contents 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
1				
2	ꝛ			
3	0001	LD-NOT	#10002ꝛ	
4		AND	#10113ꝛ	
5		OUT	#11003ꝛ	
6	ꝛ			
7	0002	SETꝛ		
8		DSTW	#1203,#1300,0FFFFH	ꝛ
9	ꝛ			
10	0003	LD	#15037ꝛ	
11		AND	#10014ꝛ	
12		TMR	#1710,#7012ꝛ	
13		OUT	#15011ꝛ	
14	ꝛ			
15	0004	LD	#13053ꝛ	
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				

Note. Symbol "ꝛ" indicates CR or LF/NL.

CODING SHEET

Fig. 9.20

9.8 LIST TAPE INPUT/OUTPUT MODE (MODE 2) (Cont'd)

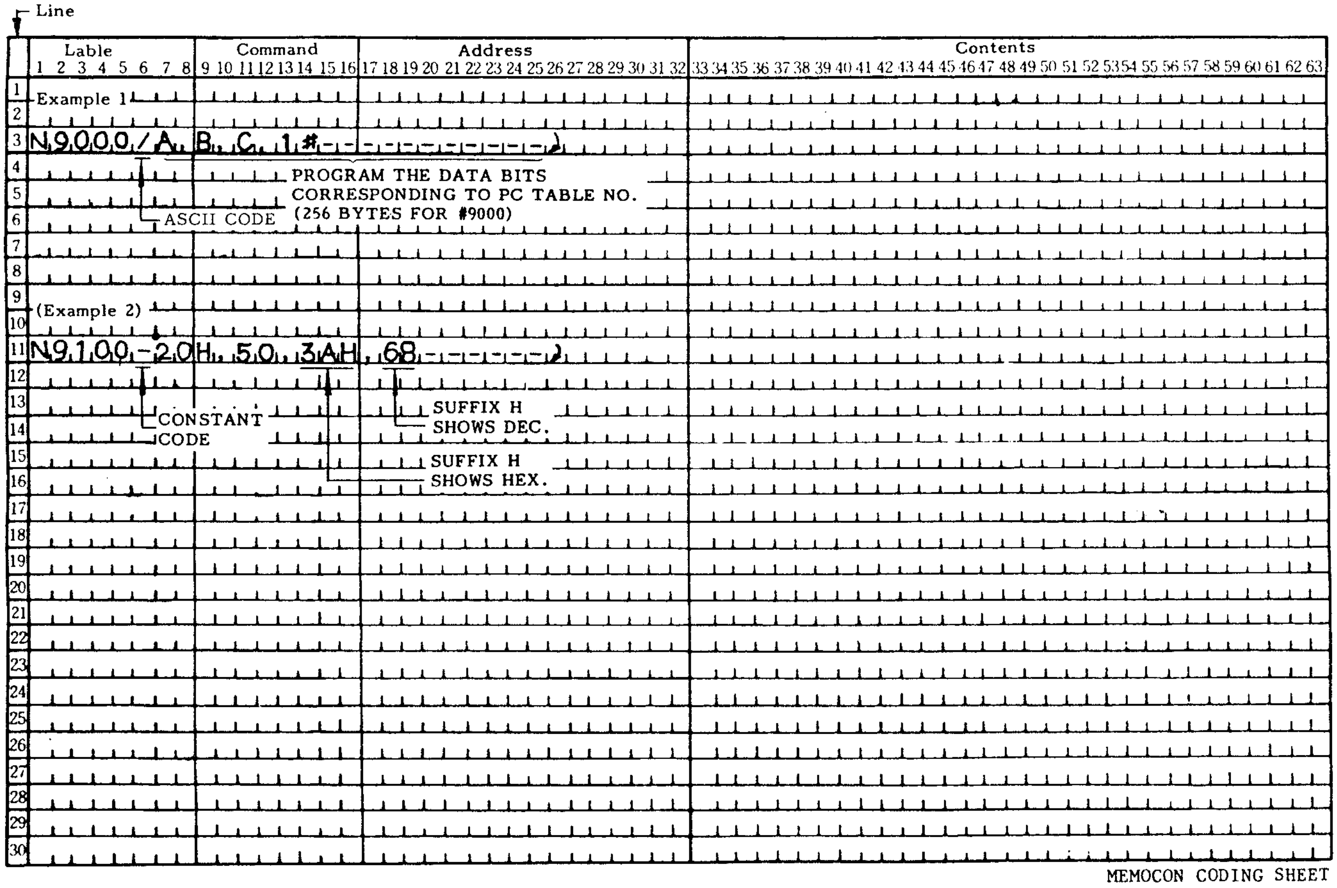


Fig. 9.21

(3) Assembling and Storing List Tape (IN)

A designed sequence ladder is coded and its data used for editing.

(a) Set a list tape on the tape reader.

(b) Depress the IN key. List tape data will be loaded into DRAM memory (edit area) as they are assembled. If a code error or punch error is detected, the tape is kept read in and the error is loaded as "NOP" code. No error indication is given.

> Note : "Assemble" operation means converting PC instruction words in list form into machine language. It follows that the PC50 edit area holds data in machine language.

(4) Punching Out List Tape (OUT)

The edited sequence program for listing on a printer is punched out in the form of list tape.

(a) Connect the RS232C or equivalent tape puncher via the data I/O interface option of the NC unit. Refer to MODE 4 FUNCTION 10.

(b) Depress the RESET key. The cursor will return to the beginning of the sequence program.

(c) Depress the OUT key. The contents of PC50 memory will be punched out onto a list tape of the PC instruction word level.

(5) Reading-in, punching-out, and verifing of PC data tables (IN, OUT, VER)

Operations of reading-in, punching-out, and verifing PC data tables should follow the procedures shown below.

Reading-in (IN)... Press T and IN keys.

Punching-out (OUT) ... Press T and OUT keys.

(6) PAUSE function

Since length of list tapes tends to become long, more than two tapes are sometimes needed. Therefore, PAUSE function is provided for the IN, and OUT operations of list tapes.

(a) OUT (punch-out)

If CAN key is pressed while a list tape is punched out, then up to the end part (i.e. AND #10013; %) of a command code will be punched out, "OUT PAUSE" will be displayed on the CRT, and the punching out stops. If the OUT key is pressed again in this state, then following data will be punched out. However, if RESET key is pressed then the punching out starts again from the beginning of the data.

(b) IN (reading in and verifing)

For reading-in and verifing operations of a list tape, when the last "%" of a command code is read-in, "IN PAUSE" is displayed and a corresponding operation stops. If IN key is pressed after changing a tape then following data will be stored or verified. However, if RESET key is pressed, then storing or verifing starts again from the first part of the data.

NOTE

1. Continue the verification of (2) above to check that the program is correctly punched out.
2. A feed hole punch portion about 75 cm long is provided at the beginning and the end of the punched-out tape.
3. The above steps apply to the punching of data in ISO code. To punch out in EIA code, press the OUT key while keeping the E key depressed.

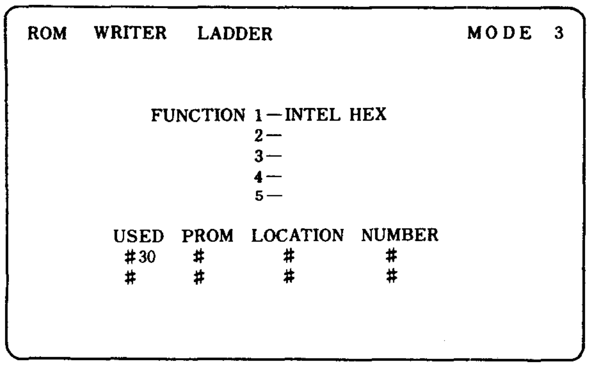
9.9 P-ROM WRITER MODE (MODE 3)

This mode is used to transfer a sequence program or PC table data from DRAM memory to a commercially available P-ROM writer connected to the control via the RS232C interface of the NC.

(1) CRT Display in MODE 3

Operate the NEXT key to select MODE 3. The following screen will appear:

(a) Ladder data



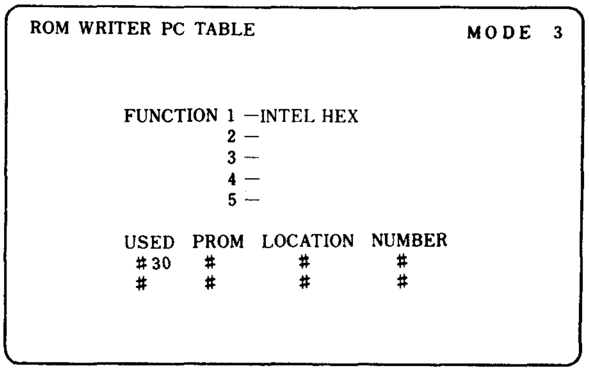
```
ROM   WRITER   LADDER                     MODE  3


             FUNCTION 1—INTEL HEX
                      2—
                      3—
                      4—
                      5—

        USED  PROM  LOCATION  NUMBER
        #30   #        #         #
        #     #        #         #
```

Fig. 9.23 Display in Mode 3

The line "30" indicates the 64 k bytes edit area of the JDU01, and the location number shows the field in which the sequence program is actually written. Numbers 30 represents location numbers of P-ROMs for further identification. That is, the edit area is represented in terms of P-ROMs.

To transfer PC table data, set the display shown below by PAGE key.

(b) PC table data



```
ROM WRITER PC TABLE                       MODE  3


             FUNCTION 1 —INTEL HEX
                      2 —
                      3 —
                      4 —
                      5 —

             USED  PROM  LOCATION  NUMBER
             #30   #        #         #
             #     #        #         #
```

Fig. 9.24

(2) Selection of P-ROM Writer

(a) The user is expected to prepare a commercially available P-ROM writer with the following 4 features:

(i) Reading in the "Intel Hex Format" is available for data transfer.

(ii) Writing to the P-ROM 27C1024 (made by HITACHI Ltd.) is available.

(iii) The RS232C interface is provided.

(iv) One of the data transfer baud rates shown in Table 9.3 on page 61 is usable.

(b) The following are some recommended P-ROM writers that meet the above requirements:

Table 9.2 Recommended P-ROM Writers

P-ROM Writer	Manufacturer
EPROM Programmer : R4945	ADOBANTESUTO INC.

(3) Writing Operation to P-ROMs

Steps to write to P-ROMs by use of the P-ROM writer R4945 of ADOBANTESUTO INC. For details, refer to the instructions for P-ROM writers :

9.9 P-ROM WRITER MODE (MODE 3) (Cont'd)

(a) Transfer conditions of R4945

(i) Selection of device
Select "HN27C1024" made by HITACHI.

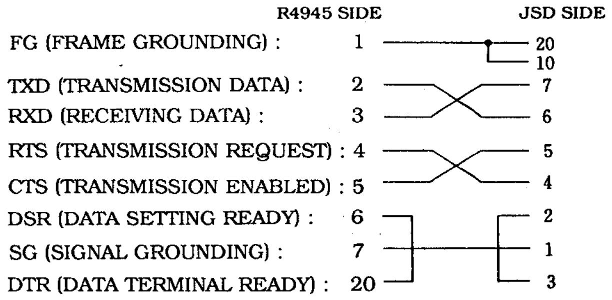
[Manufacturer's setting]
- Key-in [TYPE] , [1] and [SET] .
- Select "Hitachi" by using [▲] or [▼] key.
- Depress [SET] key.

[Setting of device type]
- Key-in TYPE, [D] and [SET] .
- Select "HN27C1024/H" by using [▲] or [▼] key.
- Depress [SET] key.

(ii) Conditions of transmission
- Key-in TYPE, [1] and [SET] .
- Select baud rate 4800 by using [▲] or [▼] key.
- Depress [DEVICE/▷] key.
- Select bit configuration, 8NO1 (8-bit, no parity, 1 stop bit) by using [▲] or [▼] key.
- Depress [DEVICE/▷] key.
- Set to ENA (to perform XON/XOFF control) by using [▲] or [▼] key.
- Depress [SET] key.

(iii) Setting of transfer format
- Set the transfer format to intel-HEX.
- Key-in [SELECT] , [3] and [SET] .
- Select intel-HEX by using [▲] or [▼] key.
- Depress [DEVICE/▷] key, and select terminator NON by using [▲] or [▼] key.
- Depress [SET] key.

(b) Connection of cable RS-232C (Cable length is around 3 m)


R4945 SIDE
JSD SIDE
FG (FRAME GROUNDING) : 1
TXD (TRANSMISSION DATA) : 2
RXD (RECEIVING DATA) : 3
RTS (TRANSMISSION REQUEST) : 4
CTS (TRANSMISSION ENABLED) : 5
DSR (DATA SETTING READY) : 6
SG (SIGNAL GROUNDING) : 7
DTR (DATA TERMINAL READY) : 20
20
10
7
6
5
4
2
1
3

Note : RS-232C termination hand-shake is provided.

(c) Writing to P-ROM writer R4945 (ladder data/PC table data)

i. Connect the P-ROM writer (hereinafter called R4945) to the RS-232C interface of NC.

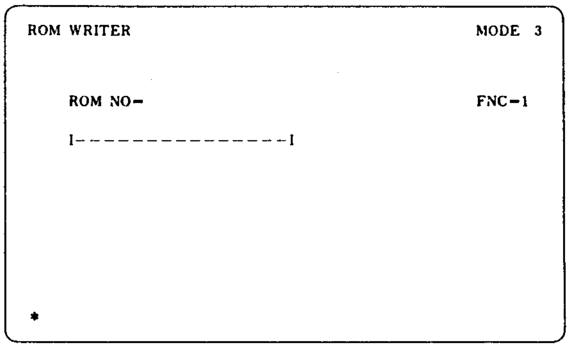
ii. Turn on the NC unit and switch to the JSD mode.

iii. Set the baud rate of the P-ROM writer (4800 bps) to "09" according to the procedure of the parameter mode "JSD MODE 4" (4) on page 61.

iv. Return to the P-ROM writer mode of MODE 3. Viewing the CRT screen, note down the location numbers of the P-ROMs to write-in (#30).
To write PC table data, depress [PAGE ▽] key to input the P-ROM location number (#30).

v. Turn on the R4945. (Transfer condition setting of R4945 in the above (a) should be completed before turning on the P-ROM writer.)

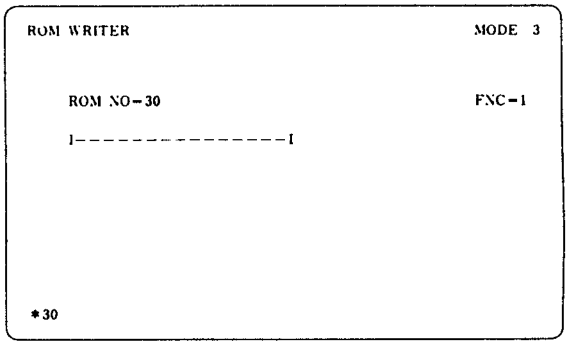
vi. Depress the [1] and [WR] keys on the editing panel. (See Fig. 9.25)



```
ROM WRITER                              MODE 3

   ROM NO-                               FNC-1
   I----------------I




*
```

Fig. 9.25

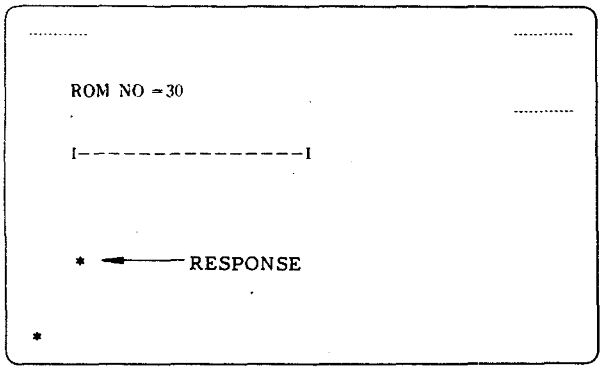
vii. Key-in a desired 2-digit P-ROM location number (noted numbers in procedure iv.) from editing panel. When the [3] , [0] and [WR] keys are keyed-in, display as shown in Fig. 9.26 will appear.



```
ROM WRITER                              MODE 3

   ROM NO-30                             FNC-1
   I----------------I




*30
```

Fig. 9.26

viii. To complete receiving the serial data, depress the R4945 keys, [SELECT] , [0] , [SET] , [0] and [SET] keys as this order.



```
ROM NO =30

I---------------I


*  ◄──── RESPONSE

*
```

Fig. 9.27

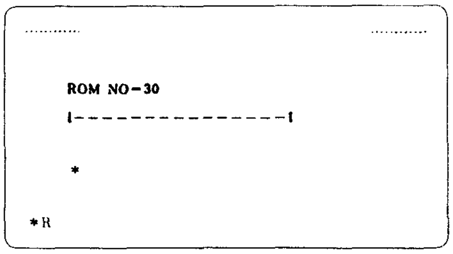
ix. Key in R and WR on editing panel. When R key is depressed, buzzer in R4945 sounds as the response. Data is transferred from the SD to the R4945 and increase asterisks (*) on the screen.

With steps i. through ix., data transfer from SD to R4945 and write-in to buffer RAM will have been completed. To transfer FC table data to R4945 after transferring ladder data to R4945, perform steps (iv) to (ix) again.

x. Set deleted P-ROM on R4945.

xi. To write-in to P-ROM, depress DEVICE, 4, SET, DEVICE and SET keys as this order. When write-in is completed, the sum value is displayed on R4945. Ladder data and PC table data are written in to #30 PROM.

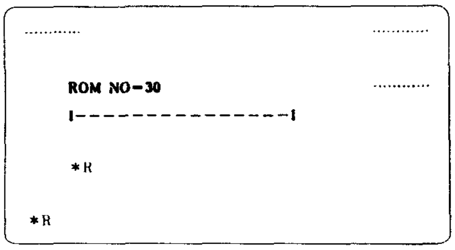
xii. To complete writing-in P-ROM of #30, take off the written-in P-ROM from R4945 and store it.



```
ROM NO-30
I---------------I

*

*R
```

R key on the editing panel is depressed.

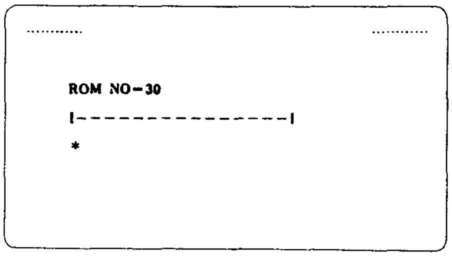
⇩



```
ROM NO-30
I---------------I

*R

*R
```

The response appears on the screen.

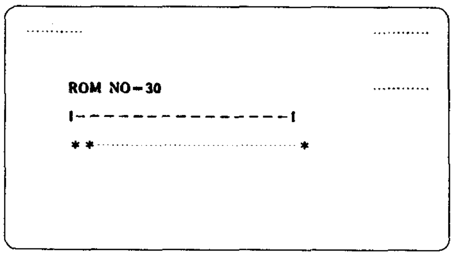
⇩



```
ROM NO-30
I---------------I
*
```

WR key on the editing panel is depressed.

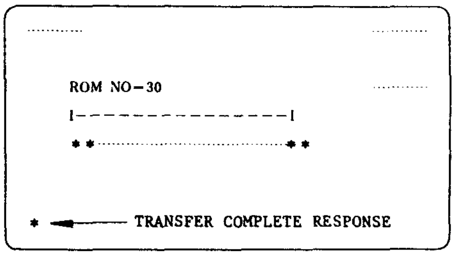
⇩



```
ROM NO-30
I---------------I
**..............*
```

Data transfer is completed.

⇩



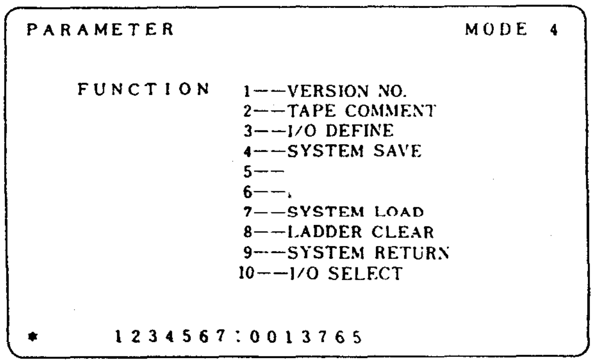
```
ROM NO-30
I---------------I
**.............**


*  ◄──── TRANSFER COMPLETE RESPONSE
```

The response appears on the screen.

Fig. 9.28

9.10 PARAMETER MODE (MODE 4)

(1) CRT Display and Functions in Parameter Mode
Operate the NEXT key to select MODE 4. The screen shown below will appear, displaying the functions available in this mode.



```
PARAMETER                                MODE 4

     FUNCTION    1--VERSION NO.
                 2--TAPE COMMENT
                 3--I/O DEFINE
                 4--SYSTEM SAVE
                 5--
                 6--,
                 7--SYSTEM LOAD
                 8--LADDER CLEAR
                 9--SYSTEM RETURN
                 10--I/O SELECT

*      1234567:0013765
```

Fig. 9.29

9.10 PARAMETER MODE (MODE 4) (Cont'd)

1. Version No. registration
2. Tape comment registration
3. Baud rate setting
4. Data transfer from DRAM to C-MOS
5. Not used
6. Not used
7. Data transfer from P-ROM to DRAM
8. Edit area clear
9. Reset to NC mode
10. I/O device selection

Keying-in one of the numbers (1 to 10) corresponding to the desired function selects that function. Given below is a detailed description of how each function can be utilized.

(2) Registering Version Number (1. VERSION NO.)

This function is used to register a sequence program version number. Be sure to register the number before writing to P-ROM.
The steps to do this are as follows :

(a) Operate the NEXT key to select MODE 4.

(b) Depress the 1 and WR keys.

(c) Key in a 7-digit number for the desired version number.

(d) Depress the WR key. The 7-digit number will be registered as the version number.

The registered version number is displayed as shown in Fig. 9.30, upon applying power to the NC system.

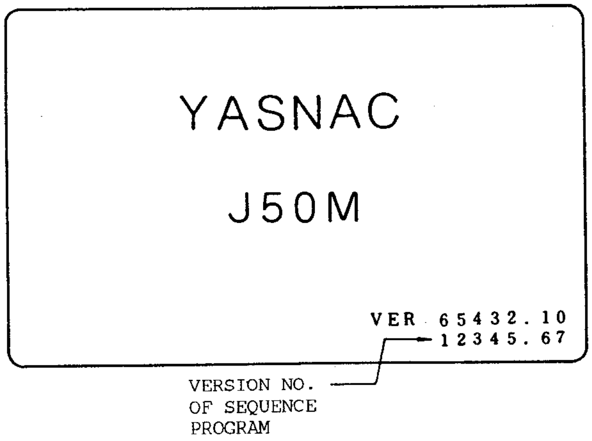


Fig. 9.30

The high-order 5 digits are separated by a decimal point from the low-order 2 digits. What the digits signify for easiest identification is up to you.

(3) Registering Tape Comment
(2. TAPE COMMENT)

This function is used, upon punching out a P-ROM format tape or list tape, to punch a registered tape comment in perforated ornate characters following the feed hole portion.

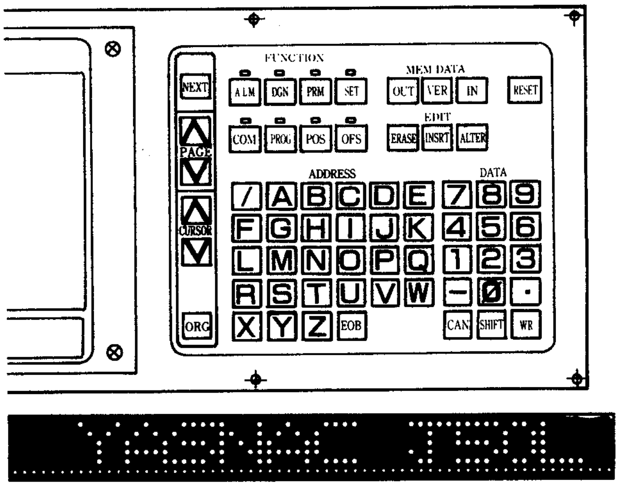
The steps to make registration are as follows:

(a) Operate the NEXT key to select MODE 4.

(b) Depress the 2, WR key.

(c) Key-in a comment in 10 characters or less. The keys shown shaded in Fig. 9.30 are usable.

(d) Depress the WR key. The typed characters will be registered as the tape comment.



Typical Ornate Characters
(10 characters or less in practice)

Fig. 9.31

(4) Setting Baud Rate (3. I/O DEFINE)

This function is used to match the baud rate of the JDU01 with the data transfer rate, or baud rate, of the RS-232C interface.
The steps to do this are as follows :

(a) Operate the NEXT key to select MODE 4.

(b) Depress the 3, WR key.

(c) Key in one of 2-digit numbers "00" to "19" that corresponds to the baud rate of the P-ROM writer. Refer to Table 9.3.

(d) Depress the WR key. The baud rate will be registered.

Table 9.3

P-ROM Writer Baud Rate	Key-Input Value	
	Data stop signal = 1 bit	Data stop signal = 2 bits
50	00	10
100	01	11
110	02	12
150	03	13
200	04	14
300	05	15
600	06	16
1200	07	17
2400	08	18
4800	09*	19

* Baud rate "09" is automatically set when the JSD mode is entered. The rate remains unchanged if the above operations are not performed.

Note: Number of bits in data stop signal depends on P-ROM writer.

(5) Data transfer from RAM to C-MOS
(4. SYSTEM SAVE)

This function transfers the contents of an edit area (RAM) to a save area (CMOS).
The steps are as follows :

(a) Depress the NEXT key and select MODE 4.

(b) Depress $\boxed{4}$ key and then $\boxed{WR}$ key.

(c) Depress $\boxed{L}$ key and then $\boxed{WR}$ key to save ladders. Depress $\boxed{T}$ key and then $\boxed{WR}$ key to save tables.

(d) "SAVE END" will be displayed when the saving is completed. "SAVE ERROR" will be displayed when an error is detected. If an error is made then repeat from the step b.

(6) Data transfer from P-ROM to RAM and from C-MOS to RAM (7. SYSTEM LOAD)

This function transfers a sequence program which has been changed to a type of hardware by a P-ROM in a PC or a program which is stored in a C-MOS memory of the JDU01 into a RAM memory in the JDU01 (edit area). Operations should follow the steps shown below.

(a) By using the ROM/CMOS switch on the JDU01, choose from which part (ROM or CMOS) the transfer to RAM is to be made.

(b) Depress NEXT key and select MODE 4.

(c) Depress $\boxed{7}$ key and then $\boxed{WR}$ key.

(d) Depress $\boxed{L}$ key and then $\boxed{WR}$ key.
The contents of the P-ROM or C-MOS is transferred to the edit area of the JDU01.

(e) For PC table, press $\boxed{T}$ key and then $\boxed{WR}$ key.

(f) When the data transfer is completed, "LOAD END" will be displayed. When an error is made, "LOAD ERROR" will be displayed. If an error is made then restart from the step (c).

(7) Clearing of the edit area (8. LADDER CLEAR)

This function clears the edit area in the JDU01 (RAM memory) or the save area (C-MOS). Make sure to perform this operation loading a sequence program into the edit area for the first time in the SD mode or after replacing the battery. Following steps show the procedure.

(a) Depress the NEXT key and select the MODE 4.

(b) Depress $\boxed{8}$ key and then $\boxed{WR}$ key.

(c) Clear operation

For ladder clear: Depress the keys in the following order.

(i) C-MOS side $\boxed{L}$, $\boxed{C}$, $\boxed{WR}$

(ii) RAM side $\boxed{T}$, $\boxed{R}$, $\boxed{WR}$

For PC table: Press the keys in the following order.

(i) C-MOS side $\boxed{T}$, $\boxed{C}$, $\boxed{WR}$

(ii) RAM side $\boxed{T}$, $\boxed{R}$, $\boxed{WR}$

(8) Return to the NC mode
(9. SYSTEM RETURN)

This function returns a mode from the XSD mode to the NC mode. This will be explained in the par. 9.13.

(9) Input/Output device selection
(10. I/O SELECT)

This function selects I/O port used in the SD mode.

(a) Depress the NEXT key and select the MODE 4.

(b) Depress $\boxed{I}$ key, $\boxed{O}$ key, and then $\boxed{WR}$ key.

(c) Depress $\boxed{n}$ and then $\boxed{WR}$ key. Here, the contents of n is given by the Table 9.4. The initial value of n when power is applied is zero. Once n is determined, the value will be retained until power is turned off or the mode returns to the NC mode.

Table 9.4

n	Input Device	Output Device
0	1RO	1RO

1RO : 1RS 232C

9.11 PC DATA TABLE EDIT MODE (MODE 5)

Following operations can be done in this mode.

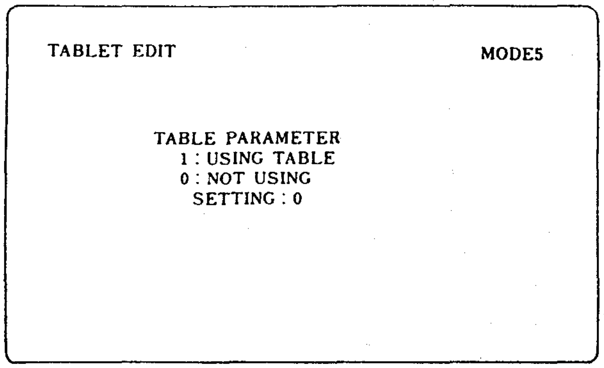
(1) Editing and address searching of PC data tables.

(2) Storing, verifying, and punching-out of P-ROM format tapes.

9.11.1 Editing of PC Data Tables

(1) CRT display in the MODE 5

(a) When the NEXT key is pressed and MODE 5 is selected, the CRT displays the following figure (shown in the Fig. 9.32).

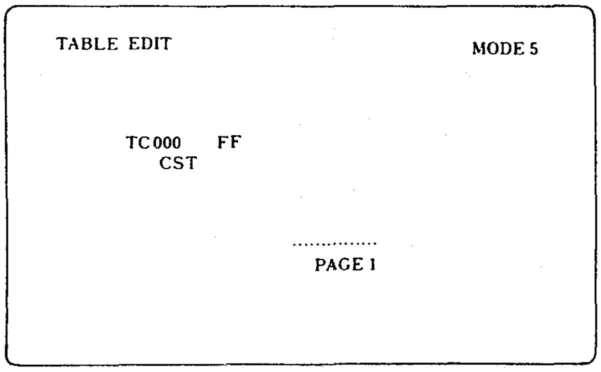


```
TABLET EDIT                                MODE5


              TABLE PARAMETER
                1 : USING TABLE
                0 : NOT USING
                  SETTING : 0
```

Fig. 9.32

(b) Fix the SETTING to "1" by pressing 1 and WR. This operation makes the PC data table usable. When the table is not used, fix the SETTING to "0" by pressing 0 and the WR.

(c) Actual edit mode is given by depressing PAGE key shown in Fig. 9.33.



```
TABLE EDIT                                 MODE 5


        TC000    FF
          CST



                          ...............
                            PAGE 1
```

Fig. 9.33

(2) Address search function

This function searches table numbers.

(a) Input a table number to be searched.

Example : By depressing 9 , 1 , 0 , 0 , the CRT displays 9100.

(b) Depress CURSOR key. The cursor moves to the table number which has been searched.

(3) Key input operation

(a) Each data can be fit into a literal data or an ASCII code data. CST reads in input data at the HEX and displays them. ASC reads in input data as ASCII code and displays them. Anything which is not present in the ASCII code is displayed as "@." CST in Fig. 9.33 indicates that the data is currently a literal data. If the cursor is moved to this position and WR key is pressed, then ASC and CST can be changed alternately.

(b) The cursor is moved up and down.

(c) Insert mode is given by depressing INSERT key, and the cursor is moved to each data.

(d) Data can be changed in the insert mode.

Example :
In case of literal data depress "4", "1", WR . In case of ASCII code data depress A , WR keys.

9.11.2 Reading-in, Punch-out, and Verify a P-ROM Format Tape (IN, OUT, and VER operations)

Like the ladder in the MODE 1, this can be done by using IN , OUT , and VER keys. Refer to the P-ROM Format Tape I/O function in par. 9.7.3 for details.

9.12 ADDRESS CHECK MODE (MODE 6)

This function checks address duplications in the sequence ladder created by the JDU01.

(1) Check address area

#1000 to #1099	(Input from a machine)
#1100 to #1199	(Output from a machine)
#1200 to #1299	(Input from the NC)
#1300 to #1399	(Output from the NC)
#1400 to #1999	(Internal registers)
#1700 to #1799	(Timer)
#7000 to #7099	(Sequence parameter)
#7100 to #7999	(Keep memory area)

(2) Check operation

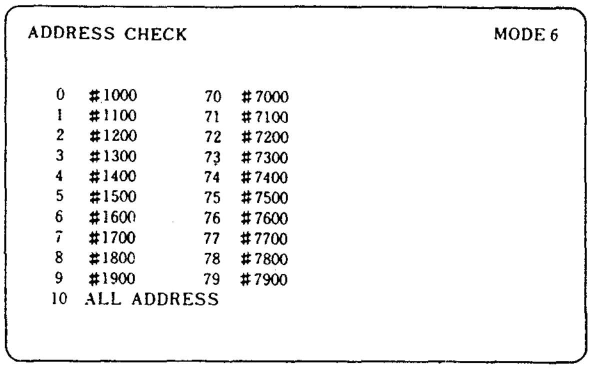
Number of "OUT #xxxxx" will be counted in the sequence ladder.

(i) For #1000's, #1200's and #1700's an address error will be displayed, if, for example, a command such as #17521 (this address not an output address) can be found.

(ii) For #1100's, #1300's from #1400's to #1900's, #7000's and from #7100 to #7900, if, for example, more than two commands such as "OUT #11112" can be found then an address error will be displayed.

(3) CRT display and its operation method

(a) When the NEXT key is pressed and MODE 6 is selected, the CRT displays Fig. 9.34.

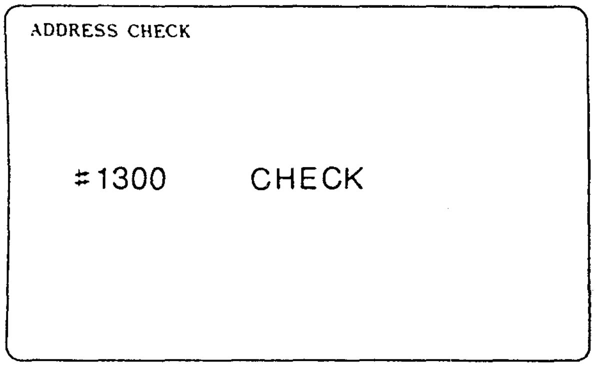


```
ADDRESS CHECK                                MODE 6

 0  #1000      70  #7000
 1  #1100      71  #7100
 2  #1200      72  #7200
 3  #1300      73  #7300
 4  #1400      74  #7400
 5  #1500      75  #7500
 6  #1600      76  #7600
 7  #1700      77  #7700
 8  #1800      78  #7800
 9  #1900      79  #7900
10  ALL ADDRESS
```

Fig. 9.34

(b) Specify a number of a range to be checked. For example, if #1300's (#1300 to #1399) will be checked then press 3, WR.

(c) When the above is keyed-in, the CRT displays the figure below (Fig. 9.35).

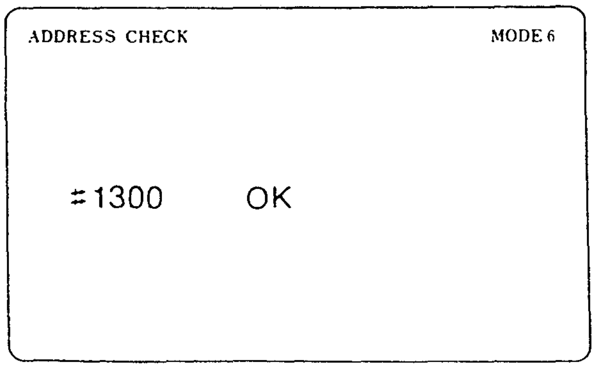


```
ADDRESS CHECK



 #1300      CHECK
```

Fig. 9.35

"#1300" shown above blinks.
In case of ALL ADDRESS CHECK, the screen continuously changes from #1000.

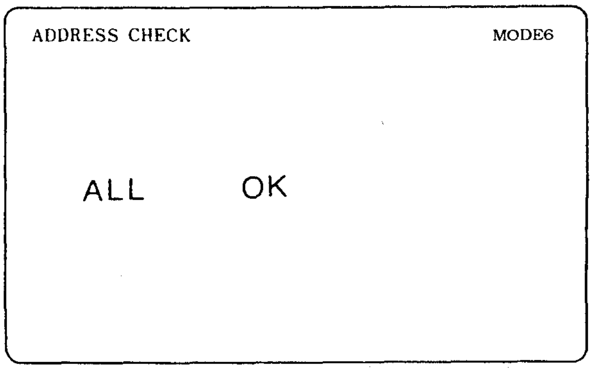
(d) When checking is completed, the CRT displays Fig. 9.36 and Fig. 9.37.



```
ADDRESS CHECK                                MODE 6



 #1300      OK
```

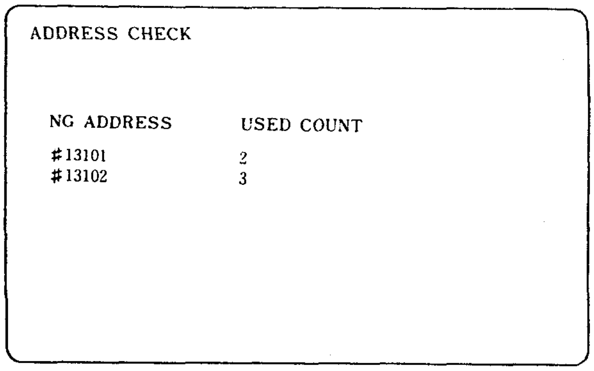
Fig. 9.36

"#1300" shown above flashes. In ALL ADDRESS CHECK, the CRT displays "ALL" as shown in the Fig. 9.38 instead of "1300".



```
ADDRESS CHECK                                MODE6



 ALL        OK
```

Fig. 9.37



```
ADDRESS CHECK


NG ADDRESS        USED COUNT
#13101            2
#13102            3
```

Fig. 9.38

Maximum USED COUNT is 255. If there exists more than 10 NG ADDRESS's, they will be displayed in the next page by using PAGE key. In ALL ADDRESS, check if a check result is NG then the operation will halt when the address or higher number address in its corresponding range is checked.

To continue checking, press CURSOR key.

To cancel the checking, press CAN key. The CRT will display the screen shown in Fig. 9.33.

9.13 RETURN TO NC SYSTEM MODE (MODE 4)

The information that follows explains how to switch from the JDU01 editing mode to the NC system mode.

9.13.1 When NC Unit Entered SD Mode from Offline State

Do not return to the NC mode if the SD mode was entered by setting the System No. switch to $\boxed{6}$ (See par. 9.6.1, When NC Unit is in Online State.)

After setting the sequence ladder to SAVE, be sure to turn off power. [For SAVE setting, see par. 9.10 (5).] When the edit area has been cleared in parameter mode, applying power supply again causes the NC mode to be entered.

Turn off power now even if a sequence program has already been edited.

9.13.2 When NC Unit Entered SD Mode from Online State

Operate the steps below if the SD mode was entered by setting the System No. switch to 4 . (See par. 9.6.2 When NC Unit is in Online State.)

(a) Depress the NEXT key to select MODE 4.

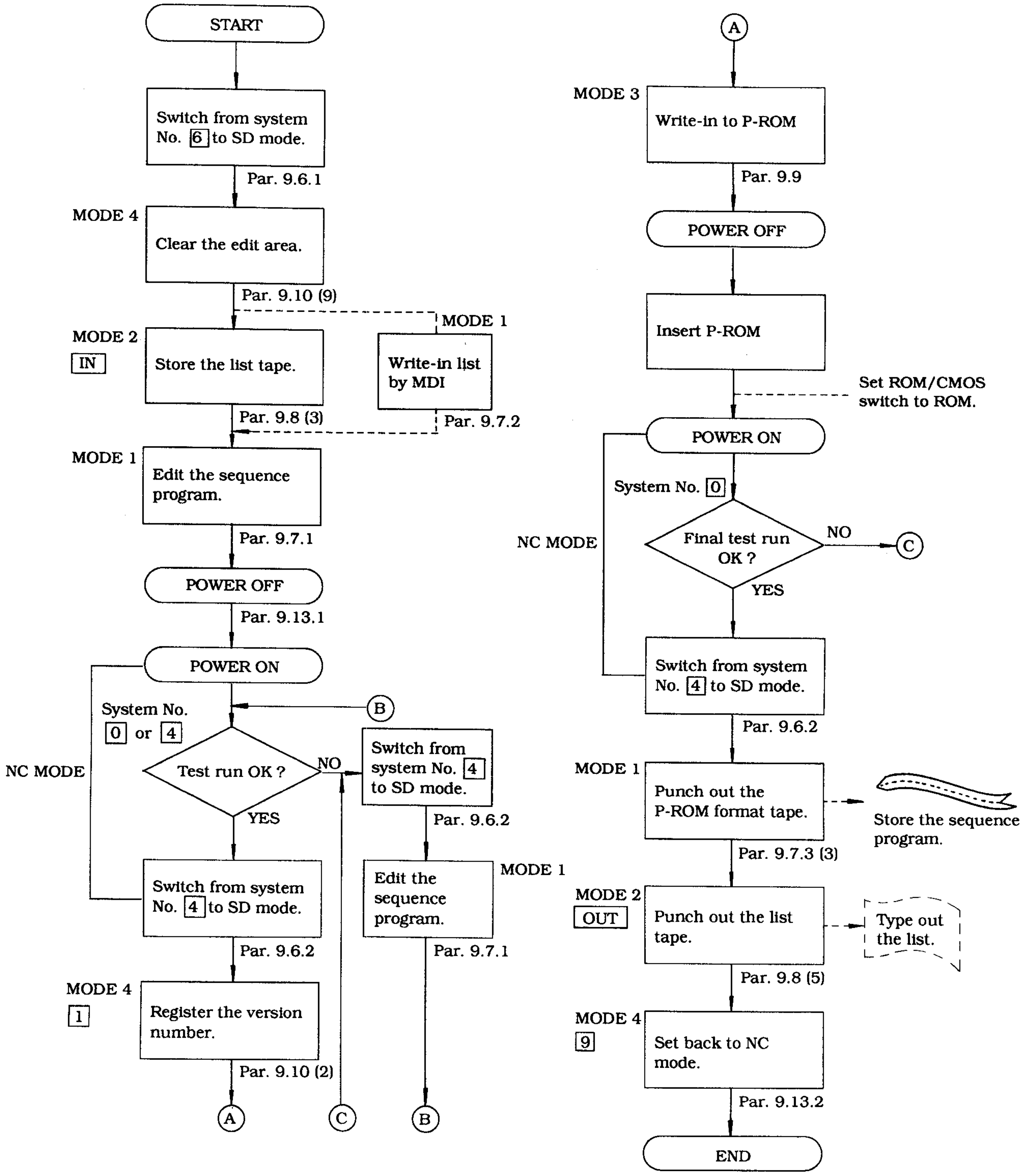
(b) Press the $\boxed{9}$ and $\boxed{\text{WR}}$ key.

(c) Press the $\boxed{\text{N}}$, $\boxed{\text{C}}$ and $\boxed{\text{ORG}}$ keys, in that order. The system will be changed from the SD mode to the NC mode.

Then setting the System No. switch to $\boxed{0}$ or $\boxed{4}$ in the NC mode enables operation check on the edited sequence program.

9.14 OPERATING PROCEDURE

Operating procedure for editing sequence program is shown in the flow chart below.



9.14 OPERATING PROCEDURE (Cont'd)

(2) Table 9.4 list the alarm codes at SD mode and operation for releasing them.

Table 9.5 Alarm Codes at SD Mode

Alarm Code	Cause	CAN key	RESET key	Remarks
ERR001	Wrong command or wrong setting keyed in.	○	○	—
ERR003	Reading or punching error of P-ROM format tape	×	○	Alarms can be released by IN or VER key.
ERR008	Address search unable	○	○	—
ERR020	Verifying error of list tape	×	○	Alarms can be released by VER or CURSOR key.
ERR040	Wrong input on MODE 3	×	○	—
ERR050	Table keyed-in not correct	○	○	—
ERR051	Table search unable	○	○	—
DISASSEMBLE	Memory contents not cleared	×	×	Alarms can be released by clearing MODE 4 edit area.
MCR ERR	Numbers of MCR and END are not same	×	×	Confirm numbers of MCR and END on MODE 1
LADDER FULL	Exceeded memory capacity	○	○	Alarm occurs while list tape and MDI are stored.
VER. ERR	Verifying error of PROM format tape	×	○	Alarm can be released by VER key.

○ : Operating the key can release the alarm.
× : Operating the key cannot release the alarm.

10. SEQUENCE PROGRAM OFFLINE EDITING SYSTEM

This section describes the software to edit/create on a personal computer sequence programs which are operated by the NC unit YASNAC J50L or J50M and turn them into ROM. The software operating on a personal computer is called JSD offline system.

10.1 OUTLINE OF OFFLINE EDITING SYSTEM

JSD offline system calls four utility groups for YASNAC J50 PLC development which operate on MS-DOS. In order to create PLC ladder ROM, the JSD offline system has exclusive ladder use complier, linker, source converter and utility to turn into ROM.

10.1.1 Operation Environment

DOS : MS-DOS Ver3.10 or above
Hardware : NEC PC-9801 series and IBM compatible machines (excluding LT and XL)
Memory : Available memory exceeding 400 k-byte

10.1.2 Execution Files

The JSD offline system is composed of the following execution files.

	File Name	Explanation
Ladder Language Compiler	JLCOMP.EXE	MS-DOS general purpose
Linker	JLLINK.EXE	MS-DOS general purpose
ROM Writer Output	JROMOUT.EXE	For PC9801 series
	EJROMOUT.EXE	For IBM compatible machines
Source Conveter	XCONV.EXE	MS-DOS general purpose

10.1.3 Outline of Execution File Processing

(1) Ladder language compiler

Compiles a source file which is coded in the ladder language and create a ROM-changed file.
The following shows data to be processed by compiler.

- Version Nos. (set at completion of linking)
- High-speed ladder programs
- Low-speed ladder programs
- Conversion data table
- Message data table

(2) Linker

Links an object file which is created by compiler.

(3) ROM writer output

Outputs a binary file from RS232C to the ROM writer by interl-HEX.

The following table shows capacity at which compiling is possible.

	Bytes	Approx. Number of Steps Calculated	Number of PROMs
J50	64K	Approx. 16000 steps	1024 k-bit, 1 unit

The following table shows contacts and register numbers with which the compiler can compile.

	Item	Contact and Register No.
Address at which compiler can be converted	Input from machine	#1000 to #1061
	Output to machine	#1100 to #1155
	Input from NC	#1200 to #1295
	Output to NC	#1300 to #1329
	Internal relay address	#1400 to #1699 #1800 to #1999
	Timer address	#1700 to #1799
	Holding-type memory address	#7000 to #7999 (Including sequencer parameters)
	Message table conversion	T9000 to T9023 T9025 to T9435

10.1.4 Outline of Operation

(1) Creating a source file in ladder language

Any editer which can create MS-DOS files can be used. To used them, create source files in ladder language. (For the details of ladder language format, refer to the description of compiler processing.)

The following shows typical creation of ladder source.

YELADDER.SRC

```
;*********************************************
;* XSD LADDER PROGRAM (YELADDER. SRC)      *
;*********************************************
;
;
;
;
;
;
HIGHSEQUENCE          ; HIGH-SPEED LADDER
INCLUDE   LAD.HI
ENDP

LOWSEQUENCE           ; LOW-SPEED LADDER
INCLUDE LAD.L01
INCLUDE LAD.L02
INCLUDE LAD.L03
ENDP

CONVERSION            ; CONVERSION DATA
INCLUDE CONV.DAT
ENDP

MESSAGE               ; MESSAGE DATA
INCLUDE MES.DAT
ENDP
```

10.1.4 Outline of Operation (Cont'd)

(2) Compiling created or modified file

Use JLCOMP to create an object file. (For the detailed operation, refer to the paragraph of compiler operation.)

(3) Collecting object files into one and creating a file to be executed

Use JLLINK to create a file to be executed. (For the detailed operation, refer to the paragraph of linker operation.)
In addition to when more than one objects are created, when all files are created with only one object, this linking processing is needed.

(4) Creating EPROM

When the resultant ladder execution check is successful, connect PC-9801 or IBM compatible machine to the ROM writer with RS-232C cable, and use JROMOUT to transfer the file to be executed to the ROM writer.

10.2 SOURCE FILE

The following describes the source file format to be input to the compiler.

10.2.1 Format of Source File

(1) Definition of character codes

① Any codes other than comments and character data must be ASCII codes. Both capital and small letters can be input. However, they cannot be identified in the internal processing. (They are identified as capital letters in the internal processing.)
Semi-block characters : 'aBc', 'a', 'Z'
Full-block characters : 'Character line', 'all'

② For comments, ASCII codes and SHIFT-JIS codes can be used.

(2) Definition of numerical values

Decimal notation	: 9, 1234
Hexadecimal notation	: 1234H, 0ab12H, 0FFH *
Contact/ladder table No.	: #1000, #10012, #9024

Note : Any hexadecimal values starting with A from F must be added with 0 at the head of them.

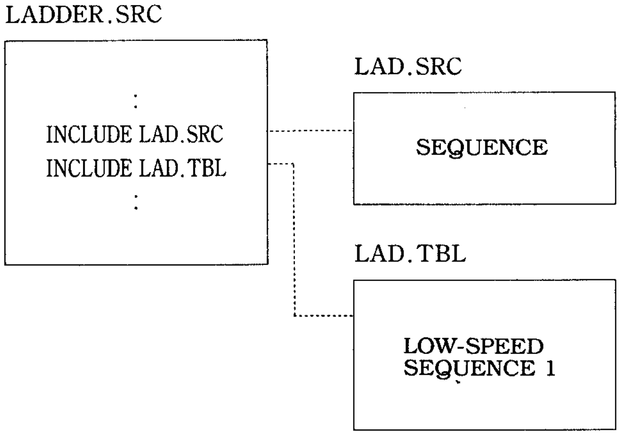
(3) Types of pseudo-instructions

The following characters are processed as pseudo-instructions. These pseudo-instructions can be used only once in each source file.

highsequence
lowsequence
conversion
message
endp
include

(4) Nesting of source files

A source file of ladder program has considerable capacity so that it cannot be edited easily.
By providing include-file function in this compiling function, souce files which are divided individually (sequence source and table source) are collected into one to compile them.



File nesting is enabled up to level 1 as shown above.

10.2.2 Source Files

The following describes the formats of source files, showing some examples.

YELADDER.SRC (Main file)

```
;************************************
;*    J50 LADDER PROGRAM     *
;************************************


①  HIGHSEQUENCE          ; HIGH-SPEED
②  INCLUDE LAD.HI          LADDER
③  ENDP
④  LOWSEQUENCE           ; LOW-SPEED
    INCLUDE LAD.L01         LADDER
    INCLUDE LAD.L02
    INCLUDE LAD.L03
    ENDP

⑤  CONVERSION            ; CONVERSION
    INCLUDE CONV.LAD        DATA
    ENDP

⑥  MESSAGE               ; MESSAGE DATA
    INCLUDE MESSAGE.DAT
    ENDP
```

(a) LAD.HI

```
;**********************************
;*    HIGH-SPEED LADDER     *
;**********************************
LD          #10000
OUT         #11000
   ~
RTH
```

(b) LAD.L01

```
;**********************************
;* LOW-SPEED LADDER (HEAD)  *
;**********************************
LD          #14000
INR         #1500
   ~
OUT         #11010
```

(c) LAD.L02

```
;**********************************
;* LOW-SPEED LADDER 2 (MIDDLE)  *
;**********************************
LD          #14056
DST         #1552.#1532.OFFH
   ~
OUT         #14033
```

(d) LAD.L03

```
;**********************************
;* LOW-SPEED LADDER 3 (END) *
;**********************************
LD          #10012
OUT         #14500
   ~
OUT         #14010
RET
```

(e) CONV.DAT

```
;**********************************
;* CONVERSION TABLE DATA  *
;**********************************
T9000  0H, 1H, 2H, 3H, 4H, 5H, 6H, 7H
  ~
T9023  0FAH, 0FBH, 0CH, 0FDH, 0FEH, 0FEH, 0FFH
```

(f) MESSAGE.DAT

```
;**********************************
;*        MESSAGE DATA        *
;**********************************
T9024            'SPINDLE'
  ~
T9323            'HARD ERROR'
```

(1) Name of source file

Any file name can be given freely.
————. SRC

(2) Source file describing format

1. There is no limitation in character starting, lines, columns such as pseudo-instructions, sequence programs, data, etc.
2. Characters in the line after ; are treated as a comment.

(3) Details of pseudo-instructions

① HIGHSEQUENCE

- Indicates the starting of high-speed ladder sequence.
- Sequence programs until ENDP are created as high-speed ladder for object.
- Format : HIGHSEQUENCE······ENDP
- High-speed ladder is not provided unless specified.
- It is necessary to write in this pseudo-instruction to the main file.

② INCLUDE

- Calls a file to be included.
- Format : INCLUDE B : LAD.L01

Name of file to be included

Name of drive where file to be included is inserted.

- It is possible to write in a pass name before the name of the file to be included.

INCLUDE B: ¥LPROG¥LOW¥LAD.L01

③ ENDP

- Indicates completion of high-speed ladder sequence, low-speed ladder sequence, conversion data or message data.
- Format : ENDP
- It is necessary to write in this pseudo-instruction to the main file.

④ LOWSEQUENCE

- Indicates the starting of low-speed ladder sequence.
- Sequence programs until ENDP are created as low-speed ladder for object.
- Format : LOWSEQUENCE······ENDP
- Low-speed ladder is not provided unless specified.
- It is necessary to write in this pseudo-instruction to the main file.

10.2.2 Source Files (Cont'd)

⑤ CONVERSION

- Created as ladder table data for object.
- Format : CONVERSION······ENDP
- Message data is not provided unless specified.

⑥ MESSAGE

- Created as message table data for object.
- Format : MESSAGE······ENDP
- Message data is not provided unless specified.

Both pseudo-instruction, MESSAGE and CONVERSION have the same meaning. You can use conversion pseudo-instruction to define the message data in it, or vice versa. (Conversion data can be defined, too, in the message pseudo-instruction.)

(4) Include file

Pseudo-instructions, HIGHSEQUENCE, LOWSEQUENCE, CONVERSION and ENDP are to be written in to the main file.

(5) Each source file

(i) High-/low-speed ladder sequence files

- Write a sequence ladder program to be high-/low-speed processed.
- There is no limitation in character starting, lines or columns, expect that at least one space must be provided between each pseudo-instruction and address.

(ii) Conversion table/message table

Conversion table/message table files

- There is no limitation in data starting lines or colums, expect that at least one space must be provided between table No. and data.
- "T" must be added to the head of the table No.
- Each data item is divided with ",".
- Characters which can be defined as message data are semi-block ASCII codes.
- The following shows the table numbers to be used.

T9000 to T9007 :	Up to 256 bytes	Conversion tables, message tables
T9008 to T9023 :	Up to 128 bytes	
T9024 to T9087 :	Up to 64 bytes	
T9088 to T9215 :	Up to 32 bytes	
T9216 to T9435 :	Up to 16 bytes	

- In the normal format, data are stored in a ladder table as byte data.

T9000 1,2,3,4,5

T 9000[0]	1
[1]	2
[2]	3
[3]	4
[4]	5

To store word data, underscore is added in front of each numerical value.

T9000 _1,_2,_3

T 9000[0]	1
[1]	
[2]	2
[3]	
[4]	3
[5]	

- All data do not have to be written.
  For example, when the number of conversion data items is 5 in SUBP007 instruction ;
  T9000 '1,2,3,4,5'
  The number of data items is 5 ; you do not have to write 256 items.
  Omitted data are treated as 0H.

10.3 COMPILER

10.3.1 Compiler Operation

Compiles created or corrected source files by JLCOMP instruction and creates object files.
The following describes how to start up JLCOMP.
JLCOMP [optional] FILE[.SRC][FILE2[.OBJ]]
[FILE3[.ERR][CR]

- Description of parameter
  Option : Display language/E → Displayed in English
  file 1 : Source file name (input)
  file 2 : Object file name (output)
  file 3 : Error file name (output)
  Bracketed parts can be omitted.
  When inputs of files 2 and 3 are omitted, default is set.
- When only JLCOMP is input, the parameter input guide is displayed.
- Example : JLCOMP B : LADTEST [CR]
  If any error occurs, LADTEST.ERR is created. When no error occurs, LADTEST.OBJ is output.
- When the include function is used, only main file is compiled ; files to be included are compiled automatically.

10.3.2 Error List of Compile

Compiler outputs an error list file with extender as ERR in a file having the same name as that of the input file. However, when any error file name is specified at activation of JLCOMP, a file having that name is output.
Compile error information is stored in this error list file.
When there is a file having the same name as that of the error list file, that file is erased.

Error list file

```
LAD.SRC   40   Unacceptable characters are used.
LAD.SRC   33   Ineffective operator
LAD.SRC   56   Number of operands is not sufficient.
   |       |     |
   |       |     L- ERROR MESSAGE
   |       L-------ERROR OCCURRING LINE NO.
   L---------------ERROR OCCURRING FILE NAME
```

10.3.3 Compiler Checking Items

Compiler checks that source format is to be processed. At the same time, it checks the following items.

(1) Command check

① Operation code check

OK : LD, LD-NOT, AND···
NG : ABS, XOR-NOT···

② Check of number of operands

OK : DEC #1001, OFFH···
NG : DEC #1001···

③ Check of operand address specifying range

OK : LD #10001···
NG : LD #10···

④ Check of operand constant specifying range

OK : MV1 #1405, 55H
NG : MV1 #1405, OFFFFH

(2) Check of upper/lower limit of number of characters set to ladder table

(3) Output contact check

- Checks that all output addresses of OUT instruction are unique.
- Checks the output contact address range.

(4) Check of MCR and END correspondence and lest level

(5) Timer check

- Checks the timer using register range.
- Checks that any timer (#1700's) addresses are not overlapped.

(6) Check of STR (STR-NOT) and AND-STR (OR-STR) correspondence

(7) SUBP calling sequence check

- Checks that SUBP corresponds to PUSH (APSH, TPSH, IPSHD).
- Checks that SUBP corresponds to STR or STR-NOT.

(8) RTH and RET presence check

- Checks that there is one RTH.
- Checks that there is RET or RTI.

10.4 LINKER

Linker reads object files in the order which are indicated in the link module specification file, and performs processing in which the data contained in the files are mapped into the executable file in the same format as that of ladder ROM.
Linker performs linking processing for the following three objects.

(1) Ladder program object

(2) Table related object
(conversion table and message table)

* More than one object file of the ladder program is not allowed.

10.4.1 Object Data and Linker Processing

The following describes the linker processing for data contained in object files.

(1) High-speed data (highsequence setting data)

- Checks the maximum range of the ladder storing area.
- An error occurs when there is no RTH.

(2) Low-speed ladder data (lowsequence setting data)

- Checks the maximum range of the ladder storing area.
- An error occurs when there is no RET or RTI.

(3) Table data

(CONVERSION/MESSAGE setting data)

- Stores message data to specify addresses corresponding to variables T9000 to T9435.
- Generates an error when the same variable data exist in some object files.

10.4.2 Linker Operation

Changes an object file output by compiler to a link binary file by JLLINK instruction.

(1) Link module file

It is necessary to create a link module file before activation of JLLINK.
By using this file, an object file to be linked is specified.

(A) Name of link module file

FILE1.LNK
Any file name can be given freely, however, the name of the extender must be always LNK.

(B) Format of link module file

- All object files to be linked are specified as shown below.
- There is not limitation in character starting line or column. (The number of characters in one line including pass is up to 80.)
- Link module can be specified within 80 characters in one column including pass.
- High-/low-speed ladders must be actually executed in the order of specification in this file.

10.4.2 Linker Operation (Cont'd)

YELAD.LNK

```
LADSRC.OBJ
LADCNV.OBJ
LADMSG.OBJ
```

(2) How to activate JLLINK

JLLINK FILE1.LNK [FILE2][CR]

- Description of parameter
  FILE1 : Name of link module specifying file (input)
  FILE2 : Name of binary file (output)
  Bracketed parts can be omitted.
  When FILE2 is omitted, the name of it will be the same as that of FILE1.
- When only JLLINK is input, the parameter input guide is displayed.

(3) Input of version No.

When a link completes successfully, a version No. can be input.
Linker motions version No. input.
The inputting range of version numbers is indicated as 7-digit value.
Since the upper 5 digits and lower 2 digits are registered separately with a decimal point, pay attention to the meaning.

```
PLEASE  INPUT  VERSION  NO : 1234567_
```

REGISTERED AS 12345.67

10.4.3 Linker Output File

The result of the linking by JLLINK is created as a binary output file.
Example : JLLINK YELAD.LNK[CR]
Output file
YELAD.BIN Ladder execution file

- Ladder execution file
  A binary file including codes where actual ladder codes are turned into assembler.

10.5 CHANGING INTO ROM

10.5.1 Selection of PROM Writer

The user is expected to prepare a commercially available P-ROM writer with the following 4 features :

(1) Reading in the "INTEL HEX Format" is available for data transfer.

(2) Writing to the P-ROM 271024 (INTEL system) is available.

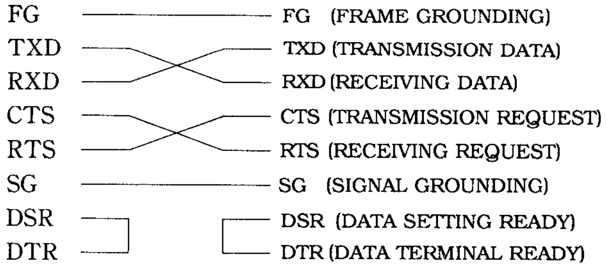
(3) The RS232C interface is provided.
The following are some recommended P-ROM writers that meet the above requirements.

AF-9704 EPROM PROGRAMMER made by ANDO DENKI CO., LTD.
EPROM PROGRAMMER R4945 made by ADOBANSUTO INC.

Ladder ROM can be created by using personal computer RS232C.

10.5.2 Line Connection

The following shows connection of personal computer and ROM writer.



10.5.3 Transfer Parameters

Transfer parameters at the PROM writer and personal computer sides must be set as follows.
(The following setting shows some recommended values. Any settings can be made only if settings of the personal computer and the PROM writer are the same.)

Baud rate	:	9600 bps
Data bit	:	8 bits
Parity	:	None
Stop bit	:	2 bits
XON/XOFF	:	ON

(1) Set the PROM writer to the receiving status and input A : ¥>JROMOUT YELAD.BIN.

The display indicates that the data are being transferred and transfer starts.

```
A:¥>JROMOUT  YELAD.BIN
JROMOUT  Ver1.0
      EXECUTING
```

(2) When transfer is completed, the main menu is returned again.

10.6 JSD LADDER SOURCE CONVERTER

Source programs or table programs output by JSD can be converted into a format where compiling is enabled by JLCOMP.

Using method

- Display of using method
  By executing without inputting a file name, the using method is displayed.

```
X/J Ladder Source Converter xconv Ver1.0
      Copyright Yaskawa Electric
USAGE     XCONV [file1] [file2]
          file1 :  input file
          file2 :  output file
```

- Conversion of source/table files
  To convert a source file named LAD.SRC into LAD2.SRC, perform the following steps.

```
A : ¥> XCONV LAD.SRC LAD2.SRC
```

  To convert a table file named LAD.TBL into LAD2.TBL, perform the following steps.

```
A : ¥> XCONV LAD.TBL LAD2.TBL
```

> The source file from which conversion is made must be the same as two files (source/table) output by JSD.
> Source file and table file must be different from each other.

10.7 LIST OF ERROR MESSAGES AND WARNING MESSAGES

The following outlines errors and warnings that compiler or linker generates.
Normally, the error file is created in Japanese. Therefore, to create it in English, add switch "/E" at compiling.

/****** <error message> *************/

1-line characters over

Illegal character is used.

Over the nest of source-file

Illegal character is used instead of pseudo-instruction.

A pseudo-instruction is used duplicatedly.

'ENDP' cannot be found.

Characters of a word is too long.

Invalid operator

Object-file memory size over

Operand of an instruction is not enough.

Operand-address is not correct.

Operand-byte-data is not correct.

Operand-word-data is not correct.

SUBP number is not correct.

Table-number define error

Table-number-setting-range is not correct.

Character data define error

Character data range define error

Character data lines over

Variable number error

Out instruction address range over

Timer-register range error

Number of MCR & END is unmatch

Byte data define error

Word data define error

Data data define error

Data range define error

Number of Operands are too large, or include valid characters.

Nest of MCR over

Duplicatedly use of valiable number

SUBP calling sequence error

Number of USBP & PUSH is unmatch

Nest of STR over

Number of stack instruction by STR is not correct.

SUBP023 parameter error¥Yn¥0"

/****** <warning message> *************/

[WARNING] Out instruction Already Define.

[WARNING] Timer address already Define.

10.8 NOTES

The number of object files to be created must be less than the following three types.

- Ladder source object
- Table related object : Conversion table
  Message table

When more than one ladder source files is provided, use INCLUDE pseudo-instruction to create them as one object at compiling.

APPENDIX 1 I/O LIST FOR YASNAC J50L (FOR LATHES)

This I/O list shows the following I/O board composition.

List No. 1 : Standard I/O board (JANCD-FC810, FC860)

List No. 2 : CRT panel built-in I/O board (JANCD-SP50)

—| |— <Input from Machine >

	D7	D6	D5	D4	D3	D2	D1	D0
#1000								
	04-36	04-21	04-05	04-35	04-20	04-34	04-19	04-33
#1001								
	04-24	04-08	04-38	04-23	04-07	04-37	04-22	04-06
#1002								
	04-11	04-41	04-26	04-10	04-40	04-25	04-09	04-39
#1003								
	04-45	04-14	04-44	04-13	04-43	04-12	04-42	04-27
#1004								
	04-49	04-18	04-48	04-17	04-47	04-16	04-46	04-15
#1005								
	05-06	05-07	05-38	05-39	05-20	05-21	05-22	05-23
#1006								
	05-08	05-09	05-40	05-10	05-24	05-25	05-11	05-12
#1007								
	05-13	05-37	05-05	05-14	05-15	05-16	05-17	05-18
#1008								
	05-41	05-26	05-27	05-19	05-33	05-34	05-35	05-36
#1009								
	05-42	05-43	05-44	05-45	05-46	05-47	05-48	05-49
#1010								
	03-11	03-41	03-26	03-10	03-40	03-25	03-09	03-39
#1011								
	03-45	03-14	03-44	03-13	03-43	03-12	03-42	03-27
#1012								
	03-49	03-18	03-48	03-17	03-47	03-16	03-46	03-15

—] [— < Input from Machine >

	D7	D6	D5	D4	D3	D2	D1	D0
#1013								
	02−16	02−09	02−03	02−15	02−08	02−02	02−14	02−01
#1016								
	04−08	04−07	04−06	04−05	04−04	04−03	04−02	04−01
#1017								
	04−16	04−15	04−14	04−13	04−12	04−11	04−10	04−09
#1018								
	04−26	04−25	04−24	04−23	04−22	04−21	04−20	04−19
#1019								
	05−08	05−07	05−06	05−05	05−04	05−03	05−02	05−01
#1020								
	05−16	05−15	05−14	05−13	05−12	05−11	05−10	05−09
#1021								
	05−26	05−25	05−24	05−23	05−22	05−21	05−20	05−19
#1022								
	06−08	06−07	06−06	06−05	06−04	06−03	06−02	06−01
#1023								
	06−16	06−15	06−14	06−13	06−12	06−11	06−10	06−09

APPENDIX 1 I/O LIST FOR YASNAC J50L (FOR LATHES) (Cont'd)

—◎— <Output to Machine>

	D7	D6	D5	D4	D3	D2	D1	D0
#1100								
	01−05	01−06	01−07	01−08	01−41	01−27	01−26	01−25
#1101								
	01−09	01−10	01−19	01−20	01−21	01−22	01−23	01−24
#1102								
	01−33	01−34	01−35	01−36	01−37	01−38	01−39	01−40
#1103								
	01−11	01−12	01−13	01−14	01−15	01−16	01−17	01−18
#1104								
	01−42	01−43	01−44	01−45	01−46	01−47	01−48	01−49
#1105								
	02−07	02−12	02−06	02−11	02−05	02−17	02−10	02−04
#1106								
	03−36	03−21	03−05	03−35	03−20	03−34	03−19	03−33
#1107								
	03−24	03−08	03−38	03−23	03−07	03−37	03−22	03−06
#1108								
	06−11	06−12	06−13	06−14	06−15	06−16	06−17	06−18
#1109								
	06−42	06−43	06−44	06−45	06−46	06−47	06−48	06−49
#1110								
	06−19	06−20	06−21	06−22	06−23	06−24	06−25	06−26
#1111								
	06−33	06−34	06−35	06−36	06−37	06−38	06−39	06−40

—◎— **< Output to Machine >**

	D7	D6	D5	D4	D3	D2	D1	D0
#1116								
	04−34	04−33	04−32	04−31	04−30	04−29	04−28	04−27
#1117								
	04−42	04−41	04−40	04−39	04−38	04−37	04−36	04−35
#1118								
	05−34	05−33	05−32	05−31	05−30	05−29	05−28	05−27
#1119								
	05−42	05−41	05−40	05−39	05−38	05−37	05−36	05−35

* For JANCD-SP-50-2, 24 points shown below are effective.

#1120								
	06−26	06−25	06−24	06−23	06−22	06−21	06−20	06−19
#1121								
	06−34	06−33	06−32	06−31	06−30	06−29	06−28	06−27
#1122								

#1107								
	03−24	03−08	03−38	03−23	03−07	03−37	03−22	03−06
#1108								
	06−11	06−12	06−13	06−14	06−15	06−16	06−17	06−18
#1109								
	06−42	06−43	06−44	06−45	06−46	06−47	06−48	06−49
#1110								
	06−19	06−20	06−21	06−22	06−23	06−24	06−25	06−26
#1111								
	06−33	06−34	06−35	06−36	06−37	06−38	06−39	06−40

APPENDIX 1 I/O LIST FOR YASNAC J50L (FOR LATHES) (Cont'd)

< Input from NC >

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1200	M28	M24	M22	M21	M18	M14	M12	M11
	M FUNCTION BCD OUTPUT							
#1201	M30R	M02R	M01R	M00R	M38	M34	M32	M31
	M30 DECODE OUTPUT	M02 DECODE OUTPUT	M01 DECODE OUTPUT	M00 DECODE OUTPUT				
#1202	TF	SF	MF	SINVA	IER	*ESPS	RST	ALM
	T-FUNCTION	S-FUNCTION	M-FUNCTION	S-4 DIGIT OUT INVERT STATUS	INPUT ERROR OUTPUT	EMERGENCY STOP OUTPUT	RESET OUTPUT	ALARM OUTPUT
	SAMPLING OUTPUT							
#1203		EDTS	AUTO	MAN	THC	RWDS	OP	DEN
		EDIT OPERATING STATUS	AUTO MODE STATUS	MANUAL MODE STATUS	THREAD CUTTING STATUS	REWIND STATUS	FEEDING	POSITIONING END
#1204	S28	S24	S22	S21	S18	S14	S12	S11
	S FUNCTION BCD OUTPUT							
#1205	T28	T24	T22	T21	T18	T14	T12	T11
	T FUNCTION BCD OUTPUT							
#1206	2ZPZ	2ZPX	ZPZ	ZPX			SPL	STL
	Z-AXIS	X-AXIS	Z-AXIS	X-AXIS			FEED HOLD LAMP	CYCLE START LAMP
	NO.2 REFERENCE POSITION		REFERENCE POSITION					
#1207								
#1216	RO8(SDO7)	RO7(SDO6)	RO6(SDO5)	RO5(SDO4)	RO4(SDO3)	RO3(SDO2)	RO2(SDO1)	RO1(SDO0)
	EXTERNAL OUTPUT FOR S-COMMAND (S4 DIGIT) NO.1							
#1217	(SDO15)	(SDO14)	(SDO13)	(SDO12)	RO12(SDO11)	RO11(SDO10)	RO10(SDO9)	RO9(SDO8)
	EXTERNAL OUTPUT FOR S-COMMAND (S4 DIGIT) NO.2							

< Input from NC >

<table>
<tr><th></th><th>D7</th><th>D6</th><th>D5</th><th>D4</th><th>D3</th><th>D2</th><th>D1</th><th>D0</th></tr>
<tr><td>#1218</td><td></td><td></td><td></td><td>FSCE</td><td>FSMD</td><td></td><td>ASF</td><td>G96S</td></tr>
<tr><td></td><td></td><td></td><td></td><td>COMPLETION OF FS MEMORY MODIFICATION</td><td>DURING FS EDITING MODE</td><td></td><td>S4-DIGIT COMMAND ANALOG SF</td><td>DURING PERIPHERAL SPEED CONSTANT CONTROL</td></tr>
<tr><td>#1219</td><td>ESEND</td><td>EREND</td><td></td><td>SETEND</td><td>TLCH</td><td>SIDXO</td><td>TPSA</td><td>SIDXA</td></tr>
<tr><td></td><td>EXTERNAL DATA SEARCH END</td><td>EXTERNAL DATA INPUT END</td><td></td><td>COORDINATE SYSTEM SETTING ENDING OUTPUT</td><td>TOOL CHANGE COMMAND (TOOL LIFE CONTROL)</td><td>SPINDLE INDEX EXECUTING</td><td>S.S. LIMIT AREA CHANGE END</td><td>SPINDLE INDEX END</td></tr>
<tr><td>#1220</td><td>UO7</td><td>UO6</td><td>UO5</td><td>UO4</td><td>UO3</td><td>UO2</td><td>UO1</td><td>UO0</td></tr>
<tr><td></td><td colspan="8">OUTPUT FOR "USER'S MACRO" NO.1</td></tr>
<tr><td>#1221</td><td>UO15</td><td>UO14</td><td>UO13</td><td>UO12</td><td>UO11</td><td>UO10</td><td>UO9</td><td>UO8</td></tr>
<tr><td></td><td colspan="8">OUTPUT FOR "USER'S MACRO" NO.2</td></tr>
<tr><td>#1222</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>#1223</td><td></td><td></td><td></td><td></td><td>FSCLRE</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>COMPLETION OF FS DATA CLEAR</td><td></td><td></td><td></td></tr>
<tr><td>#1224</td><td>MD7</td><td>MD6</td><td>MD5</td><td>MD4</td><td>MD3</td><td>MD2</td><td>MD1</td><td>MD0</td></tr>
<tr><td></td><td colspan="8">HIGH-SPEED M FUNCTIONS</td></tr>
<tr><td>#1280</td><td>SSW3</td><td>SSW2</td><td>SSW1</td><td>SSW0</td><td></td><td></td><td></td><td>SKIP</td></tr>
<tr><td></td><td colspan="4">SYSTEM NO. SWITCH</td><td></td><td></td><td></td><td>SKIP INPUT</td></tr>
<tr><td>#1281</td><td>0</td><td>OFFPB</td><td></td><td>ONPB</td><td>OLD</td><td>SVALM</td><td>ESP</td><td>OHT</td></tr>
<tr><td></td><td></td><td>POWER OFF PB.</td><td></td><td>POWER ON PB.</td><td>OVERLOAD</td><td>SERVO ALARM</td><td>EMERGENCY STOP</td><td>OVERHEAT</td></tr>
<tr><td>#1282</td><td>1HP7</td><td>1HP6</td><td>1HP5</td><td>1HP4</td><td>1HP3</td><td>1HP2</td><td>1HP1</td><td>1HP0</td></tr>
<tr><td></td><td colspan="8">NO.1 MANUAL PULSE GENERATOR MONITOR</td></tr>
</table>

APPENDIX 1 I/O LIST FOR YASNAC J50L (FOR LATHES) (Cont'd)

< Input from NC >

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1283					SET3	SET2	SET1	SET0

SETTING #6219 MONITOR

#1284	SVMX	SVMX						

SERVO POWER ON
(= "NRD")

#1285	0	0	0	0	0	0	0	1

CONSTANT "1"

#1286	0	0	0	0	0	0	0	0

CONSTANT "0"

#1287		PCS	PBS	PAS				
		PHASE-C	PHASE-B	PHASE-A				

SIGNAL FROM SPINDLE PG

#1288	TGONX	PCX	PBX	PAX	*ALX	*OLX	FUX	SRDX
	X-AXIS TG ON	PHASE-C	PHASE-B	PHASE-A				

SIGNAL FROM X-AXIS PG

MONITOR FOR SERVO UNIT OF X-AXIS

#1289	TGONZ	PCZ	PBZ	PAZ	*ALZ	*OLZ	FUZ	SRDZ
	Z-AXIS TG ON	PHASE-C	PHASE-B	PHASE-A				

SIGNAL FROM Z-AXIS PG

MONITOR FOR SERVO UNIT OF Z-AXIS

#1290	SCOM28	SCOM24	SCOM22	SCOM21	SCOM18	SCOM14	SCOM12	SCOM11

SPINDLE COMMAND MONITOR

#1291	SCOM48	SCOM44	SCOM42	SCOM41	SCOM38	SCOM34	SCOM32	SCOM31

SPINDLE COMMAND MONITOR

#1292	SO28	SO24	SO22	SO21	SO18	SO14	SO12	SO11

SPINDLE OUTPUT MONITOR

—||— < Input from NC >

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1293	SO48	SO44	SO42	SO41	SO38	SO34	SO32	SO31

SPINDLE OUTPUT MONITOR

#1294	ALM 28	ALM 24	ALM 22	ALM 21	ALM 18	ALM 14	ALM 12	ALM 11

ALARM CODE MONITOR

#1295					ALM 38	ALM 34	ALM 32	ALM 31

ALARM CODE MONITOR

#1296	INHEDTT	AFLT	ABST	DRNT	BDTT	DLRT	MLKT	SBKT

SETTING #6000 MONITOR

APPENDIX 1 I/O LIST FOR YASNAC J50L (FOR LATHES) (Cont'd)

—O— < Output to NC >

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1300	EDT	MEM	D	T		H/S	J	RT
	EDIT	MEMORY	MDI	TAPE		HANDLE/ STEP	MANUAL JOG	MANUAL RAPID
#1301	MP1	ROV2	ROV1	FV16	FV8	FV4	FV2	FV1
	MANUAL PG MULTIPLE SELECT	RAPID SPEED OVERRIDE		FEEDRATE OVERRIDE/MANUAL JOG SPEED				
#1302	HZ	HX	−Z	+Z	−X	+X	MP4	MP2
	MANUAL PG AXIS SELECT		MANUAL TRAVERSE AXIS DIRECTION SELECT				MANUAL PG MULTIPLY SELECT	
#1303	INHEDT	AFL	ABS	DRN	BDT	DLK	MLK	SBK
	INHIBIT EDIT	M.S.T LOCK	MANUAL ABS.	DRY RUN	BLOCK DELETE	DISPLAY LOCK	MACHINE LOCK	SINGLE BLOCK
#1304	ZRN	CDZ	SMZ	RWDH	SRN	PST	*SP	ST
	RETURN TO REFERENCE	THREAD CUT UP	ERROR DETECT	HIGH SPEED REWIND	SET UP POINT RETURN	POSITION SET	FEED HOLD	CYCLE START
#1305	ERR1	ERR0	STLK	RWD	EOP	ERS	FIN	MRD
	EXTERNAL ERROR INPUT		INTERRUPT	REWIND	END OF PROGRAM	EXTERNAL RESET	MST FIN	MACHINE READY
#1306	SAGR		*DCZ	*DCX	*−LZ	*+LZ	*−LX	*+LX
	SPINDLE SPEED AGREEMENT		DECREASE INPUT FOR REFERENCE POINT		OVERTRAVEL INPUT			
#1307	GRS	GSC	SSTP	SINV	GR4	GR3	GR2	GR1
	S-COMMAND CONSTANT	SPINDLE SPEED CONSTANT	S-COMMAND "0"	S-COMMAND INVERT	SPINDLE GEAR RANGE SELECT			
#1308	EOUT	EVER	EIN	DRSZ	DRSX			EXTC
	NC PROGRAM PUNCH OUT	NC PROGRAM VERIFY	NC PROGRAM INPUT	DISPLAY RESET				TIME COUNT
#1309	BDT9	BDT8	BDT7	BDT6	BDT5	BDT4	BDT3	BDT2
	ADDITIONAL BLOCK DELETE							

—O— < Output to NC >

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1310	WN16	WN8	WN4	WN2	WN1	SPC	SPB	SPA
	EXTERNAL WORK NUMBER SEARCH					SPINDLE OVERRIDE		
#1311				CPRN	HOFS	MIX	PRST	OVC
				CUTTING POINT RETURN	AUTO MODE HANDLE OFFSET	X-AXIS MIRROR IMAGE	PROGRAM RESTART	OVERRIDE CANCEL
#1312	SETREQ	FSCLR		COV16	COV8	COV4	COV2	COV1
	COORDINATE SYSTEM SETTING REQUEST INPUT	FS DATA CLEAR		G71/G72 CUTTING OVERRIDE				
#1313		FSCH	FSMEM	FSCM				
		FS MEMORY MODIFICATION	FS MEMORY	FS EDITING MODE				
#1316	SID8	SID7	SID6	SID5	SID4	SID3	SID2	SID1
	SPINDLE INDEX POSITION SET							
#1317	TP8	TP4	TP2	TP1	SID12	SID11	SID10	SID9
	TOOL NO. SET FOR STORED STROKE LIMIT				SPINDLE INDEX POSITION SET			
#1318	TLTM		TLSKP	TLRST	SIDXI	SIDXINC	TPS	SIDX
	TIMER COUNT		TOOL SKIP	TOOL RESET	SPINDLE INDEX RESTART	SPINDLE INDEX POSITION INCREMENTAL DESIGNATION	TOOL NO. CHANGE FOR S.S. LIMIT	SPINDLE INDEXING
	SIGNAL FOR TOOL LIFE CONTROL							
#1319	ROV4	SPE	SPD	TLA21	TLA18	TLA14	TLA12	TLA11
	EXTENDED RAPID TRAVERSE OVERRIDE	EXTENDED SPINDLE OVERRIDE		CHANGE TOOL NO. (TOOL LIFE CONTROL)				
#1320								
#1321								

APPENDIX 1 I/O LIST FOR YASNAC J50L (FOR LATHES) (Cont'd)

—O— < Output to NC >

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
# 1 3 2 2	SONPB SERVO POWER ON							

# 1 3 2 3	RI 8 (SDI 7)	RI 7 (SDI 6)	RI 6 (SDI 5)	RI 5 (SDI 4)	RI 4 (SDI 3)	RI 3 (SDI 2)	RI 2 (SDI 1)	RI 1 (SDI 0)

EXTERNAL INPUT OF S-COMMAND (S4 DIGIT) NO.1

# 1 3 2 4	(SDI15)	(SDI14)	(SDI13)	(SDI12)	RI12 (SDI11)	RI11 (SDI10)	RI10 (SDI 9)	RI 9 (SDI 8)

EXTERNAL INPUT FOR S-COMMAND (S4 DIGIT) NO.2

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

INPUT FOR "USER'S MACRO" NO.1

# 1 3 2 6	UI15	UI14	UI13	UI12	UI11	UI10	UI 9	UI 8

INPUT FOR "USER'S MACRO" NO.2

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

EXTERNAL DATA INPUT NO.1

# 1 3 2 8	ED15	ED14	ED13	ED12	ED11	ED10	ED 9	ED 8

EXTERNAL DATA INPUT NO.2

# 1 3 2 9	EDCL	EDS 2	EDS 1	EDSO	EDSD	EDSC	EDSB	EDSA

CONTROL SIGNAL FOR EXTERNAL DATA INPUT

APPENDIX 2 I/O LIST FOR YASNAC J50M (FOR MACHINING CENTERS)

This I/O list shows the following I/O board composition.

List No. 1 : Standard mounted I/O board (JANCD-FC810, FC860)
List No. 2 : Standard mounted I/O board (JANCD-FC810, FC860)

—┤├— < Input from Machine >

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1000	04—36	04—21	04—05	04—35	04—20	04—34	04—19	04—33
#1001	04—24	04—08	04—38	04—23	04—07	04—37	04—22	04—06
#1002	04—11	04—41	04—26	04—10	04—40	04—25	04—09	04—39
#1003	04—45	04—14	04—44	04—13	04—43	04—12	04—42	04—27
#1004	04—49	04—18	04—48	04—17	04—47	04—16	04—46	04—15
#1005	05—06	05—07	05—38	05—39	05—20	05—21	05—22	05—23
#1006	05—08	05—09	05—40	05—10	05—24	05—25	05—11	05—12
#1007	05—13	05—37	05—05	05—14	05—15	05—16	05—17	05—18
#1008	05—41	05—26	05—27	05—19	05—33	05—34	05—35	05—36
#1009	05—42	05—43	05—44	05—45	05—46	05—47	05—48	05—49
#1010	03—11	03—41	03—26	03—10	03—40	03—25	03—09	03—39
#1011	03—45	03—14	03—44	03—13	03—43	03—12	03—42	03—27
#1012	03—49	03—18	03—48	03—17	03—47	03—16	03—46	03—15

(For special application)

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1013	02—16	02—09	02—03	02—15	02—08	02—02	02—14	02—01

APPENDIX 2 I/O LIST FOR YASNAC J50M (FOR MACHINING CENTERS) (Cont'd)

—| |— <Input from Machine>

	D7	D6	D5	D4	D3	D2	D1	D0
#1016								
	04—36	04—21	04—05	04—35	04—20	04—34	04—19	04—33
#1017								
	04—24	04—08	04—38	04—23	04—07	04—37	04—22	04—06
#1018								
	04—11	04—41	04—26	04—10	04—40	04—25	04—09	04—39
#1019								
	04—45	04—14	04—44	04—13	04—43	04—12	04—42	04—27
#1020								
	04—49	04—18	04—48	04—17	04—47	04—16	04—46	04—15
#1021								
	05—06	05—07	05—38	05—39	05—20	05—21	05—22	05—23
#1022								
	05—08	05—09	05—40	05—10	05—24	05—25	05—11	05—12
#1023								
	05—13	05—37	05—05	05—14	05—15	05—16	05—17	05—18
#1024								
	05—41	05—26	05—27	05—19	05—33	05—34	05—35	05—36
#1025								
	05—42	05—43	05—44	05—45	05—46	05—47	05—48	05—49
#1026								
	03—11	03—41	03—26	03—10	03—40	03—25	03—09	03—39
#1027								
	03—45	03—14	03—44	03—13	03—43	03—12	03—42	03—27
#1028								
	03—49	03—18	03—48	03—17	03—47	03—16	03—46	03—15
#1029								
	02—16	02—09	02—03	02—15	02—08	02—02	02—14	02—01

—◎— <Output to Machine>

	D7	D6	D5	D4	D3	D2	D1	D0
#1100								
	01−05	01−06	01−07	01−08	01−41	01--27	01−26	01−25
#1101								
	01−09	01−10	01−19	01−20	01−21	01−22	01−23	01−24
#1102								
	01−33	01−34	01−35	01−36	01−37	01−38	01−39	01−40
#1103								
	01−11	01−12	01−13	01−14	01−15	01−16	01−17	01−18
#1104								
	01−42	01−43	01−44	01−45	01−46	01−47	01−48	01−49
#1105								
	02−07	02−12	02−06	02−11	02−05	02−17	02−10	02−04
#1106								
	03−36	03−21	03−05	03−35	03−20	03−34	03−19	03−33
#1107								
	03−24	03−08	03−38	03−23	03−07	03−37	03−22	03−06
#1108								
	06−11	06−12	06−13	06−14	06−15	06−16	06−17	06−18
#1109								
	06−42	06−43	06−44	06−45	06−46	06−47	06−48	06−49
#1110								
	06−19	06−20	06−21	06−22	06−23	06−24	06−25	06−26
#1111								
	06−33	06−34	06−35	06−36	06−37	06−38	06−39	06−40

APPENDIX 2 I/O LIST FOR YASNAC J50M (FOR MACHINING CENTERS) (Cont'd)

—◎— < Output to Machine >

	D7	D6	D5	D4	D3	D2	D1	D0
#1116								
	01−05	01−06	01−07	01−08	01−41	01−27	01−26	01−25
#1117								
	01−09	01−10	01−19	01−20	01−21	01−22	01−23	01−24
#1118								
	01−33	01−34	01−35	01−36	01−37	01−38	01−39	01−40
#1119								
	01−11	01−12	01−13	01−14	01−15	01−16	01−17	01−18
#1120								
	01−42	01−43	01−44	01−45	01−46	01−47	01−48	01−49
#1121								
	02−07	02−12	02−06	02−11	02−05	02−17	02−10	02−04
#1122								
	03−36	03−21	03−05	03−35	03−20	03−34	03−19	03−33
#1123								
	03−24	03−08	03−38	03−23	03−07	03−37	03−22	03−06
#1124								
	06−11	06−12	06−13	06−14	06−15	06−16	06−17	06−18
#1125								
	06−42	06−43	06−44	06−45	06−46	06−47	06−48	06−49
#1126								
	06−19	06−20	06−21	06−22	06−23	06−24	06−25	06−26
#1127								
	06−33	06−34	06−35	06−36	06−37	06−38	06−39	06−40

< Input from NC >

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1200	M30	M02	M01	M00	DEN	OP	SPL	STL
					POSITIONING COMPLETED	FEEDING	TEMPORARY STOP	CYCLE START
#1201	2ZPα	2ZPZ	2ZPY	2ZPX	1ZPα	1ZPZ	1ZPY	1ZPX
	SECOND REFERENCE POINT LAMP				FIRST REFERENCE POINT LAMP			
#1202	4ZPα	4ZPZ	4ZPY	4ZPX	3ZPα	3ZPZ	3ZPY	3ZPX
	FOURTH REFERENCE POINT LAMP				THIRD REFERENCE POINT LAMP			
#1203								
#1204								
#1205			TLCHA	TLCHB	TCF	TCHGF		
			SIGNAL FOR TOOL CHANGE	SIGNAL FOR SELECT NEW TOOL	SIGNAL FOR READING TOOL CHANGE	SIGNAL FOR COMPLETION OF TOOL CHANGE/GROUP		
#1206							RPDO	SINVA
							OUTPUT DURING RAPID TRAVERSE	OUTPUT DURING SPINDLE TRAVERSE
#1207					SIDXA	SIDXO	SLPS	G93M
					SPINDLE INDEXING COMPLETION OUTPUT	OUTPUT DURING SPINDLE INDEXING	SPINDLE LOOP STATUS OUTPUT	G93 MODE OUTPUT
#1208	WSFTER	WSFTK						
		WORK COORDINATE SYSTEM DISPLAY SETTING COMPLETION						

APPENDIX 2 I/O LIST FOR YASNAC J50M (FOR MACHINING CENTERS) (Cont'd)

< Input from NC >

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1209	MANINTK					FSCLRK	FSCE	FSMD
	SIGNAL FOR MACHINING INTERRUPTING POINT RETURN COMPLETION					COMPLETION OF FS DATA CLEAR	COMPLETION OF FS MEMORY CHANGE	FS EDITING
#1210								
#1211					MIMGMα	MIMGMZ	MIMGMY	MIMGMX
					DURING MIRROR IMAGE			
#1212								
#1213								
#1214								
#1215	G84S							
	G74/G84 EXECUTING							
#1216	T8/T28	T7/T24	T6/T22	T5/T21	T4/T18	T3/T14	T2/T12	T1/T11
	T-FUNCTION BINARY/BCD OUTPUT							
#1217	T16/T48	T15/T44	T14/T42	T13/T41	T12/T38	T11/T34	T10/T32	T9/T31
	T-FUNCTION BINARY/BCD OUTPUT							

< Input from NC >

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1218	TAP	M04S	TLMO	G80S	EREND	ESEND	RST	AL
	TAPPING	SPINDLE REVERSING	TOOL LENGTH MEASUREMENT	CANNED CYCLE	EXTERNAL DATA INPUT COMPLETED	EXTERNAL DATA INPUT COMPLETED	RESET	ALARM
#1219	SRV	SSP	FMF	EF	BF	TF	SF	MF
	SPINDLE REVERSE	SPINDLE STOP	MF	EXTERNAL MOTION	B-FUNCTION	T-FUNCTION	S-FUNCTION	M-FUNCTION
	FOR CANNED CYCLE							
#1222	M8	M7	M6	M5	M4	M3	M2	M1
	M-FUNCTION BINARY							
#1223	OS	EDTS	IER	4NGC	AUTO	MAN	RDY	RWDS
	ORIENTATION	EDITING	INPUT ERROR	4TH-AXIS DISREGARD	AUTOMATIC	MANUAL	LT PREPARATION COMPLETED	REWIND
#1224	SDA8/SB8	SDA7/SB7	SDA6/SB6	SDA5/SB5	SDA4/SB4	SDA3/SB3	SDA2/SB2	SDA1/SB1
	S5-DIGIT ANALOG OUTPUT/S4-DIGIT 12-BIT NO-CONTACT OUTPUT							
#1225	SDA16	SDA15	SDA14	SDA13	SDA12/SB12	SDA11/SB11	SDA10/SB10	SDA9/SB9
	S5-DIGIT ANALOG OUTPUT/S4-DIGIT 12-BIT NO-CONTACT OUTPUT							
#1226								
#1227								
#1228								

APPENDIX 2 I/O LIST FOR YASNAC J50M (FOR MACHINING CENTERS) (Cont'd)

—▌▌— < Input from NC >

	D7	D6	D5	D4	D3	D2	D1	D0
#1229								
#1230								
#1231								
#1232	B8/B28	B7/B24	B6/B22	B5/B21	B4/B18	B3/B14	B2/B12	B1/B11
	B-FUNCTION BINARY/BCD OUTPUT							
#1233	B16/B48	B15/B44	B14/B42	B13/B41	B12/B38	B11/B34	B10/B32	B9/B31
	B-FUNCTION BINARY/BCD OUTPUT						HIGH-SPEED GEAR	LOW-SPEED GEAR
#1234	S28	S24	S22	S21	S18	S14	S12/GRH	S11/GRL
	S-FUNCTION BCD OUTPUT							
#1235	S48	S44	S42	S41	S38	S34	S32	S31
	S-FUNCTION BCD OUTPUT							
#1236	U7	U6	U5	U4	U3	U2	U1	U0
	MACRO PROGRAM							
#1237	U15	U14	U13	U12	U11	U10	U9	U8
	MACRO PROGRAM							

< Input from NC >

	D7	D6	D5	D4	D3	D2	D1	D0
#1238								
#1239								
#1277	1HP7	1HP6	1HP5	1HP4	1HP3	1HP2	1HP1	1HP0 MONITOR
	FIRST HANDLE PULSE							
#1278								
#1279								
#1280				SKIP	SN4	SN3	SN2	SN1
					SYSTEM NO. SWITCH			
#1281		0		ON-PB	OLD	SVALM	ESP	OHT
				POWER ON SWITCH	OVERLOAD	SERVO ALARM	EMERGENCY STOP	OVERHEAT
#1282	JSD	BALM	0	1	EXAXIS	PAGEO	0	EXCMOS
	JSD MOUNT 0:WITH JSD 1:WITHOUT JSD	BATTERY ALARM MONITOR		CONSTANT "1"	OPTION AXIS BOARD MOUNT MONITOR 0:WITH SR51 BOARD 1:WITHOUT SR51 BOARD	CONTACT SIGNAL MONITOR FOR MAINTENANCE		
#1283					SNS4	SNS3	SNS2	SNS1
					SYSTEM NO. SETTING			

APPENDIX 2 I/O LIST FOR YASNAC J50M (FOR MACHINING CENTERS) (Cont'd)

—▮▮— < Input from NC >

	D7	D6	D5	D4	D3	D2	D1	D0
#1284	SVON	NRD						
	SERVO POWER ON	NC READY						
#1285	0	0	0	0	0	0	0	1
	CONSTANTS "1"							
#1286	0	0	0	0	0	0	0	0
	CONSTANTS "0"							
#1287						PCS	PBS	PAS
						SPINDLE PG MONITOR		
#1288	TGONX	PCX	PBX	PAX	•ALX	•OLX	FUX	SRDX
	X-AXIS PG MONITOR				X-AXIS SERVO UNIT MONITOR			
#1289	TGONY	PCY	PBY	PAY	•ALY	•OLY	FUY	SRDY
	Y-AXIS PG MONITOR				Y-AXIS SERVO UNIT MONITOR			
#1290	TGONZ	PCZ	PBZ	PAZ	•ALZ	•OLZ	FUZ	SRDZ
	Z-AXIS PG MONITOR				Z-AXIS SERVO UNIT MONITOR			
#1291	TGON4	PC4	PB4	PA4	*AL4	*OL4	FU4	SRD4
	4-AXIS PG MONITOR				4-AXIS SERVO UNIT MONITOR			
#1292								

—||— < Input from NC >

	D7	D6	D5	D4	D3	D2	D1	D0
#1293				ZNGC	ABSC	EDTLKC		
				Z-AXIS DISREGARD	MANUAL ABSOLUTE	EDIT LOCK		
#1294	AFLC	MLKC	OPTC	DRNC	BTDC	DLKC	STLKC	SBKC
	AUX FUNCTION LOCK	MACHINE LOCK	OPTIONAL STOP FUNCTION ENABLED	DRY RUN	OPTIONAL BLOCK SKIP FUNCTION ENABLED	DISPLAY LOCK	START LOCK	SINGLE BLOCK
#1295			PLBKC		MIαC	MIZC	MIYC	MIXC
			PLAYBACK		α	Z	Y	X
#1297	ALM28	ALM24	ALM22	ALM21	ALM18	ALM14	ALM12	ALM11
#1298					ALM38	ALM34	ALM32	ALM31

#1293, #1294: SETTING MONITOR

#1295 D3–D0: MIRROR IMAGE AXIS

#1297: ALARM CODE MONITOR (LOWER 2-DIGIT BCD CODE)

#1298 D3–D0: ALARM CODE MONITOR (UPPER 1-DIGIT BCD CODE)

APPENDIX 2 I/O LIST FOR YASNAC J50M (FOR MACHINING CENTERS) (Cont'd)

—O— < Output to NC >

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1300	EDT	MEM	MDI	T	S	H	J	RT
	EDIT	MEMORY	MANUAL DATA INPUT	TAPE	STEP	HANDLE	MANUAL FEED	RAPID TRAVERSE
#1301	OVC	ROV 2	ROV 1	OV16	OV 8	OV 4	OV 2	OV 1
	OVERRIDE CANCEL	RAPID TRAVERSE RATE OVERRIDE		FEEDRATE OVERRIDE				
#1302	−α	+α	−Z	+Z	−Y	+Y	−X	+X
	MANUAL FEEDRATE SELECTION							
#1303	SPC	SPB	SPA	JV16	JV 8	JV 4	JV 2	JV 1
	SPINDLE SPEED OVERRIDE			MANUAL FEEDRATE OVERRIDE				
#1304	DRS	MP 4	MP 2	MP 1	Hα	HZ	HY	HX
	DISPLAY RESET	HANDLE PULSE MULTIPLY			HANDLE AXIS SELECTION			
#1305	AFL	MLK	OPT	DRN	BDT	DLK		SBK
	AUXILIARY FUNCTION LOCK	MACHINE LOCK	OPTIONAL STOP	DRY RUN	BLOCK DELETE	DISPLAY LOCK		SINGLE BLOCK
#1306	SRN	F 1	RET	TLMI	ZRN	EDTLK	*SP	ST
	PROGRAM RESTART	F1-DIGIT	RETRACT	MEASURED LENGTH	ZERO RETURN	EDIT LOCK	FEED HOLD	CYCLE START
#1307	PINT	ZNG	ABS		MIα	MIZ	MIY	MIX
	PROGRAM INTERRUPTION	Z-AXIS DISREGRAD	MANUAL ABSOLUTE		MIRROR IMAGE			
#1308	9 BDT	8 BDT	7 BDT	6 BDT	5 BDT	4 BDT	3 BDT	2 BDT
	SPECIAL BLOCK DELETE							

—O— <Output to NC>

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1309	ZRN2		4NG	EWS	PST		WSH	
	MANUAL 2ND HOME POSITION RETURN		4TH-AXIS DISREGARD	EXTERNAL WRITE SIGNAL	TOUCH SENSOR SIGNAL			
#1310	DEVSLTI							
	DESIGNATION OF TAPE OPERATION DEVICE							
#1311								
#1312	PLYBK	TCNT	SENSON	TLCTN	TLSKP	TLRST	ESC1	ESC0
	PLAYBACK	CUTTING TIME COUNT IN VALIDATING SIGNAL	TOUCH SWITCH VALIDATING SIGNAL	TOOL LIFE CONTROL			EXT STROKE CHECK SELECTION	
#1313	RWDH	FSCH	FSMEM	FSCM	SLPC			
	HIGH-SPEED REWIND & AUTOSTART	FEEDING SPEED, SPINDLE SPEED MEMORY CHANGE	FEEDING SPEED, SPINDLE SPEED STORED	FEEDING SPEED, SPINDLE SPEED EDITING MODE	SPINDLE LOOP COMMAND INPUT			
#1314	SPE	SPD	ROV4		ECLM			
	SPINDLE OVERRIDE		RAPID TRAVERSE RATE OVERRIDE		EXTERNAL PROGRAM CLEAR			
#1315	MANINT	SSM	PPR		NTCRQ	TCFIN		
	SIGNAL FOR MACHINING INTERRUPTING POINT RETURN COMPLETION	SENSOR STOP MODE	MACHINING INTERRUPTING POINT RETURN MODE		REQUEST FOR NEW TOOL REPLACEMENT	COMPLETION OF TOOL REPLACEMENT		
#1316	FFIN	FIN		EOP	ERS	EXTC	STLK	MRD
	CANNED CYCLE FIN	MST COMPLETION		END PROGRAM	EXTERNAL RESET	EXTERNAL TIME COUNT	CYCLE START INTERLOCK	MACHINE READY COMPLETED

EXT STROKE CHECK SELECTION

0	0	Invalid
0	1	3rd
1	0	4th
1	1	5th

APPENDIX 2 I/O LIST FOR YASNAC J50M (FOR MACHINING CENTERS) (Cont'd)

—O— < Output to NC >

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1317	S-INV	S-FIN		SAGR	SOR	GRB	GRA	GST
	SPINDLE REVERSE	S CODE COMPLETED	SPINDLE STOP	SPINDLE COINCIDENCE	SPINDLE ORIENTATION	GEAR SELECTION		GEAR SHIFT
#1318	ERR 2	ERR 1	ERR 0	SENS(G)	SENS(T)	EXOUT	EXVER	EXIN
	DEC TO STOP	IMMEDIATE STOP	BLOCK STOP	ELMINATE	TOOL BREAKAGE	EXTERNAL OUTPUT	EXTERNAL VERIFY	EXTERNAL INPUT
#1319	*−Lα	*+Lα	*−LZ	*+LZ	*−LY	*+LY	*−LX	*+LX
	OVERTRAVEL							
#1320	HOFS				*ITα	*ITZ	*ITY	*ITX
					AXIS INTERLOCK			
#1321	*−EDα	*+EDα	*−EDZ	*+EDZ	*−EDY	*+EDY	*−EDX	*+EDX
	EXTERNAL DECELERATION							
#1322	SONPB				*SVOFα	*SVOFZ	*SVOFY	*SVOFX
	SERVO POWER ON				SERVO OFF			
#1323	UI 7	UI 6	UI 5	UI 4	UI 3	UI 2	UI 1	UI 0
	MACRO PROGRAM							
#1324	UI15	UI14	UI13	UI12	UI11	UI10	UI 9	UI 8
	MACRO PROGRAM							
#1325	ED 7	ED 6	ED 5	ED 4	ED 3	ED 2	ED 1	ED 0
	EXTERNAL DATA INPUT							

—O— < Output to NC >

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1326	ED15	ED14	ED13	ED12	ED11	ED10	ED 9	ED 8
	EXTERNAL DATA INPUT							
#1327	EDCL	EDAS 2	EDAS 1	EDAS 0	EDSD	EDSC	EDSB	EDSA
	EXTERNAL DATA INPUT CONTROL SIGNAL							
#1328	EDAS 3				*DECα	*DECZ	*DECY	*DECX
	EXTERNAL COORDINATE SYSTEM SHIFT ABS/1NC				DECELERATION LS			
#1329		TL64	TL32	TL16	TL 8	TL 4	TL 2	TL 1
		INPUT TOOL GROUP NO. (TOOL LIFT CONTROL)						SKIP SIGNAL
#1330								
#1331	SDI8	SDI7	SDI6	SDI5	SDI4	SDI3	SDI2	SDI1
	EXTERNAL INPUT FOR S-COMMAND (S5 DIGIT)							
#1332	SDI16	SDI15	SDI14	SDI13	SDI12	SDI11	SDI10	SDI9
	EXTERNAL INPUT FOR S-COMMAND (S5 DIGIT)							
#1335	FSCLS	WSKAN1						
	FS EDIT DATA CLEAR	MANUAL SKIP COMPLETION SIGNAL						
#1338	WN128	WN64	WN32	WN16	WN8	WN4	WN2	WN1
	EXTERNAL WORK NO. SEARCH INPUT							

APPENDIX 2 I/O LIST FOR YASNAC J50M (FOR MACHINING CENTERS) (Cont'd)

—O— <Output to NC>

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1341	SID8	SID7	SID6	SID5	SID4	SID3	SID2	SID1

DESIGNATION OF SPINDLE INDEXING POSITION

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1342					SID12	SID11	SID10	SID9

DESIGNATION OF SPINDLE INDEXING POSITION

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#1343					SPCUT	SIDXI	SIDXINC	SIDX
					SPINDLE INDEXING STOP	SPINDLE INDEXING RESTART	SPINDLE INDEXING POSITION INCREMENTAL DESIGNATION	REQUEST FOR SPINDLE INDEXING

APPENDIX 3 LIST OF INTERNAL RELAYS, REGISTERS FOR YASNAC J50L/J50M

< Internal Relays >

	D7	D6	D5	D4	D3	D2	D1	D0
#1400								
#1401								
#1402								
#1403								
#1404								
#1405								
#1406								
#1407								
#1408								
#1409								
#1410								
#1411								
#1412								
#1413								
#1414								
#1415								
#1416								
#1417								
#1418								
#1419								
#1420								
#1421								
#1422								
#1423								
#1424								

APPENDIX 3 LIST OF INTERNAL RELAYS, REGISTERS FOR YASNAC J50L/J50M (Cont'd)

	D7	D6	D5	D4	D3	D2	D1	D0
#1425								
#1426								
#1427								
#1428								
#1429								
#1430								
#1431								
#1432								
#1433								
#1434								
#1435								
#1436								
#1437								
#1438								
#1439								
#1440								
#1441								
#1442								
#1443								
#1444								
#1445								
#1446								
#1447								
#1448								
#1449								

	D7	D6	D5	D4	D3	D2	D1	D0
#1450								
#1451								
#1452								
#1453								
#1454								
#1455								
#1456								
#1457								
#1458								
#1459								
#1460								
#1461								
#1462								
#1463								
#1464								
#1465								
#1466								
#1467								
#1468								
#1469								
#1470								
#1471								
#1472								
#1473								
#1474								

APPENDIX 3 LIST OF INTERNAL RELAYS, REGISTERS FOR YASNAC J50L/J50M (Cont'd)

	D7	D6	D5	D4	D3	D2	D1	D0
#1475								
#1476								
#1477								
#1478								
#1479								
#1480								
#1481								
#1482								
#1483								
#1484								
#1485								
#1486								
#1487								
#1488								
#1489								
#1490								
#1491								
#1492								
#1493								
#1494								
#1495								
#1496								
#1497								
#1498								
#1499								

─□─ < Register >

#1500 □
#1501 □
#1502 □
#1503 □
#1504 □
#1505 □
#1506 □
#1507 □
#1508 □
#1509 □
#1510 □
#1511 □
#1512 □
#1513 □
#1514 □
#1515 □
#1516 □
#1517 □
#1518 □
#1519 □
#1520 □
#1521 □
#1522 □
#1523 □
#1524 □
#1525 □
#1526 □
#1527 □
#1528 □
#1529 □
#1530 □
#1531 □
#1532 □
#1533 □
#1534 □
#1535 □
#1536 □
#1537 □
#1538 □
#1539 □
#1540 □
#1541 □
#1542 □
#1543 □
#1544 □
#1545 □
#1546 □
#1547 □
#1548 □
#1549 □

#1550 □
#1551 □
#1552 □
#1553 □
#1554 □
#1555 □
#1556 □
#1557 □
#1558 □
#1559 □
#1560 □
#1561 □
#1562 □
#1563 □
#1564 □
#1565 □
#1566 □
#1567 □
#1568 □
#1569 □
#1570 □
#1571 □
#1572 □
#1573 □
#1574 □
#1575 □
#1576 □
#1577 □
#1578 □
#1579 □
#1580 □
#1581 □
#1582 □
#1583 □
#1584 □
#1585 □
#1586 □
#1587 □
#1588 □
#1589 □
#1590 □
#1591 □
#1592 □
#1593 □
#1594 □
#1595 □
#1596 □
#1597 □
#1598 □
#1599 □

APPENDIX 3 LIST OF INTERNAL RELAYS, REGISTERS FOR YASNAC J50L/J50M (Cont'd)

#1600

#1601

#1602

#1603

#1604

#1605

#1606

#1607

#1608

#1609

#1610

#1611

#1612

#1613

#1614

#1615

#1616

#1617

#1618

#1619

#1620

#1621

#1622

#1623

#1624

#1625

#1626

#1627

#1628

#1629

#1630

#1631

#1632

#1633

#1634

#1635

#1636

#1637

#1638

#1639

#1640

#1641

#1642

#1643

#1644

#1645

#1646

#1647

#1648

#1649

#1650

#1651

#1652

#1653

#1654

#1655

#1656

#1657

#1658

#1659

#1660

#1661

#1662

#1663

#1664

#1665

#1666

#1667

#1668

#1669

#1670

#1671

#1672

#1673

#1674

#1675

#1676

#1677

#1678

#1679

#1680

#1681

#1682

#1683

#1684

#1685

#1686

#1687

#1688

#1689

#1690

#1691

#1692

#1693

#1694

#1695

#1696

#1697

#1698

#1699

TMR < Sequencer Timer >

	(8 ms Timer)	Set Value
#1700		
#1701		
#1702		
#1703		
#1704		
#1705		
#1706		
#1707		
#1708		
#1709		

	(0.1 s Timer)	
#1710		
#1711		
#1712		
#1713		
#1714		
#1715		
#1716		
#1717		
#1718		
#1719		
#1720		
#1721		
#1722		
#1723		
#1724		
#1725		
#1726		
#1727		
#1728		
#1729		

	(50 ms Timer)	
#1730		
#1731		
#1732		
#1733		
#1734		
#1735		
#1736		
#1737		
#1738		
#1739		
#1740		
#1741		
#1742		
#1743		
#1744		
#1745		
#1746		
#1747		
#1748		
#1749		

	(1 s Timer)	Set Value
#1750		
#1751		
#1752		
#1753		
#1754		
#1755		
#1756		
#1757		
#1758		
#1759		

	(1 min Timer)	
#1770		
#1771		
#1772		
#1773		

	(8 ms Timer)	
#1760		
#1761		
#1762		
#1763		
#1764		
#1765		
#1766		
#1767		
#1768		
#1769		

	(0.1 s Timer)	
#1790		
#1791		
#1792		
#1793		
#1794		
#1795		
#1796		
#1797		
#1798		
#1799		

	(50 ms Timer)	
#1780		
#1781		
#1782		
#1783		
#1784		
#1785		
#1786		
#1787		
#1788		
#1789		

APPENDIX 3 LIST OF INTERNAL RELAYS, REGISTERS FOR YASNAC J50L/J50M (Cont'd)

—□— < Register >

#1800

#1801

#1802

#1803

#1804

#1805

#1806

#1807

#1808

#1809

#1810

#1811

#1812

#1813

#1814

#1815

#1816

#1817

#1818

#1819

#1820

#1821

#1822

#1823

#1824

#1825

#1826

#1827

#1828

#1829

#1830

#1831

#1832

#1833

#1834

#1835

#1836

#1837

#1838

#1839

#1840

#1841

#1842

#1843

#1844

#1845

#1846

#1847

#1848

#1849

#1850

#1851

#1852

#1853

#1854

#1855

#1856

#1857

#1858

#1859

#1860

#1861

#1862

#1863

#1864

#1865

#1866

#1867

#1868

#1869

#1870

#1871

#1872

#1873

#1874

#1875

#1876

#1877

#1878

#1879

#1880

#1881

#1882

#1883

#1884

#1885

#1886

#1887

#1888

#1889

#1890

#1891

#1892

#1893

#1894

#1895

#1896

#1897

#1898

#1899

#1900
#1901
#1902
#1903
#1904
#1905
#1906
#1907
#1908
#1909
#1910
#1911
#1912
#1913
#1914
#1915
#1916
#1917
#1918
#1919
#1920
#1921
#1922
#1923
#1924
#1925
#1926
#1927
#1928
#1929
#1930
#1931
#1932
#1933
#1934
#1935
#1936
#1937
#1938
#1939
#1940
#1941
#1942
#1943
#1944
#1945
#1946
#1947
#1948
#1949

#1950
#1951
#1952
#1953
#1954
#1955
#1956
#1957
#1958
#1959
#1960
#1961
#1962
#1963
#1964
#1965
#1966
#1967
#1968
#1969
#1970
#1971
#1972
#1973
#1974
#1975
#1976
#1977
#1978
#1979
#1980
#1981
#1982
#1983
#1984
#1985
#1986
#1987
#1988
#1989
#1990
#1991
#1992
#1993
#1994
#1995
#1996
#1997
#1998
#1999

APPENDIX 3 LIST OF INTERNAL RELAYS, REGISTERS FOR YASNAC J50L/J50M (Cont'd)

< Sequencer Parameter >

	D7	D6	D5	D4	D3	D2	D1	D0
#7000								
#7001								
#7002								
#7003								
#7004								
#7005								
#7006								
#7007								
#7008								
#7009								
#7010								
#7011								
#7012								
#7013								
#7014								
#7015								
#7016								
#7017								
#7018								
#7019								
#7020								
#7021								
#7022								
#7023								
#7024								

	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
#7025								
#7026								
#7027								
#7028								
#7029								
#7030								
#7031								
#7032								
#7033								
#7034								
#7035								
#7036								
#7037								
#7038								
#7039								
#7040								
#7041								
#7042								
#7043								
#7044								
#7045								
#7046								
#7047								
#7048								
#7049								

APPENDIX 3 LIST OF INTERNAL RELAYS, REGISTERS FOR YASNAC J50L/J50M (Cont'd)

	D7	D6	D5	D4	D3	D2	D1	D0
#7050								
#7051								
#7052								
#7053								
#7054								
#7055								
#7056								
#7057								
#7058								
#7059								
#7060								
#7061								
#7062								
#7063								
#7064								
#7065								
#7066								
#7067								
#7068								
#7069								
#7070								
#7071								
#7072								
#7073								
#7074								

	D7	D6	D5	D4	D3	D2	D1	D0
#7075								
#7076								
#7077								
#7078								
#7079								
#7080								
#7081								
#7082								
#7083								
#7084								
#7085								
#7086								
#7087								
#7088								
#7089								
#7090								
#7091								
#7092								
#7093								
#7094								
#7095								
#7096								
#7097								
#7098								
#7099								

APPENDIX 3 LIST OF INTERNAL RELAYS, REGISTERS FOR YASNAC J50L/J50M (Cont'd)

—○— <Keep Memory>

#7100
#7101
#7102
#7103
#7104
#7105
#7106
#7107
#7108
#7109
#7110
#7111
#7112
#7113
#7114
#7115
#7116
#7117
#7118
#7119
#7120
#7121
#7122
#7123
#7124
#7125
#7126
#7127
#7128
#7129
#7130
#7131
#7132
#7133
#7134
#7135
#7136
#7137
#7138
#7139
#7140
#7141
#7142
#7143
#7144
#7145
#7146
#7147
#7148
#7149

#7150
#7151
#7152
#7153
#7154
#7155
#7156
#7157
#7158
#7159
#7160
#7161
#7162
#7163
#7164
#7165
#7166
#7167
#7168
#7169
#7170
#7171
#7172
#7173
#7174
#7175
#7176
#7177
#7178
#7179
#7180
#7181
#7182
#7183
#7184
#7185
#7186
#7187
#7188
#7189
#7190
#7191
#7192
#7193
#7194
#7195
#7196
#7197
#7198
#7199

#7200
#7201
#7202
#7203
#7204
#7205
#7206
#7207
#7208
#7209
#7210
#7211
#7212
#7213
#7214
#7215
#7216
#7217
#7218
#7219
#7220
#7221
#7222
#7223
#7224
#7225
#7226
#7227
#7228
#7229
#7230
#7231
#7232
#7233
#7234
#7235
#7236
#7237
#7238
#7239
#7240
#7241
#7242
#7243
#7244
#7245
#7246
#7247
#7248
#7249
#7250
#7251
#7252
#7253
#7254
#7255
#7256
#7257
#7258
#7259
#7260
#7261
#7262
#7263
#7264
#7265
#7266
#7267
#7268
#7269
#7270
#7271
#7272
#7273
#7274
#7275
#7276
#7277
#7278
#7279
#7280
#7281
#7282
#7283
#7284
#7285
#7286
#7287
#7288
#7289
#7290
#7291
#7292
#7293
#7294
#7295
#7296
#7297
#7298
#7299

APPENDIX 3 LIST OF INTERNAL RELAYS, REGISTERS FOR YASNAC J50L/J50M (Cont'd)

#7300
#7301
#7302
#7303
#7304
#7305
#7306
#7307
#7308
#7309
#7310
#7311
#7312
#7313
#7314
#7315
#7316
#7317
#7318
#7319
#7320
#7321
#7322
#7323
#7324
#7325
#7326
#7327
#7328
#7329
#7330
#7331
#7332
#7333
#7334
#7335
#7336
#7337
#7338
#7339
#7340
#7341
#7342
#7343
#7344
#7345
#7346
#7347
#7348
#7349
#7350
#7351
#7352
#7353
#7354
#7355
#7356
#7357
#7358
#7359
#7360
#7361
#7362
#7363
#7364
#7365
#7366
#7367
#7368
#7369
#7370
#7371
#7372
#7373
#7374
#7375
#7376
#7377
#7378
#7379
#7380
#7381
#7382
#7383
#7384
#7385
#7386
#7387
#7388
#7389
#7390
#7391
#7392
#7393
#7394
#7395
#7396
#7397
#7398
#7399

#7400

#7401

#7402

#7403

#7404

#7405

#7406

#7407

#7408

#7409

#7410

#7411

#7412

#7413

#7414

#7415

#7416

#7417

#7418

#7419

#7420

#7421

#7422

#7423

#7424

#7425

#7426

#7427

#7428

#7429

#7430

#7431

#7432

#7433

#7434

#7435

#7436

#7437

#7438

#7439

#7440

#7441

#7442

#7443

#7444

#7445

#7446

#7447

#7448

#7449

#7450

#7451

#7452

#7453

#7454

#7455

#7456

#7457

#7458

#7459

#7460

#7461

#7462

#7463

#7464

#7465

#7466

#7467

#7468

#7469

#7470

#7471

#7472

#7473

#7474

#7475

#7476

#7477

#7478

#7479

#7480

#7481

#7482

#7483

#7484

#7485

#7486

#7487

#7488

#7489

#7490

#7491

#7492

#7493

#7494

#7495

#7496

#7497

#7498

#7499

APPENDIX 3 LIST OF INTERNAL RELAYS, REGISTERS FOR YASNAC J50L/J50M (Cont'd)

#7500

#7501

#7502

#7503

#7504

#7505

#7506

#7507

#7508

#7509

#7510

#7511

#7512

#7513

#7514

#7515

#7516

#7517

#7518

#7519

#7520

#7521

#7522

#7523

#7524

#7525

#7526

#7527

#7528

#7529

#7530

#7531

#7532

#7533

#7534

#7535

#7536

#7537

#7538

#7539

#7540

#7541

#7542

#7543

#7544

#7545

#7546

#7547

#7548

#7549

#7550

#7551

#7552

#7553

#7554

#7555

#7556

#7557

#7558

#7559

#7560

#7561

#7562

#7563

#7564

#7565

#7566

#7567

#7568

#7569

#7570

#7571

#7572

#7573

#7574

#7575

#7576

#7577

#7578

#7579

#7580

#7581

#7582

#7583

#7584

#7585

#7586

#7587

#7588

#7589

#7590

#7591

#7592

#7593

#7594

#7595

#7596

#7597

#7598

#7599

#7600

#7601

#7602

#7603

#7604

#7605

#7606

#7607

#7608

#7609

#7610

#7611

#7612

#7613

#7614

#7615

#7616

#7617

#7618

#7619

#7620

#7621

#7622

#7623

#7624

#7625

#7626

#7627

#7628

#7629

#7630

#7631

#7632

#7633

#7634

#7635

#7636

#7637

#7638

#7639

#7640

#7641

#7642

#7643

#7644

#7645

#7646

#7647

#7648

#7649

#7650

#7651

#7652

#7653

#7654

#7655

#7656

#7657

#7658

#7659

#7660

#7661

#7662

#7663

#7664

#7665

#7666

#7667

#7668

#7669

#7670

#7671

#7672

#7673

#7674

#7675

#7676

#7677

#7678

#7679

#7680

#7681

#7682

#7683

#7684

#7685

#7686

#7687

#7688

#7689

#7690

#7691

#7692

#7693

#7694

#7695

#7696

#7697

#7698

#7699

APPENDIX 3 LIST OF INTERNAL RELAYS, REGISTERS FOR YASNAC J50L/J50M (Cont'd)

#7700
#7701
#7702
#7703
#7704
#7705
#7706
#7707
#7708
#7709
#7710
#7711
#7712
#7713
#7714
#7715
#7716
#7717
#7718
#7719
#7720
#7721
#7722
#7723
#7724
#7725
#7726
#7727
#7728
#7729
#7730
#7731
#7732
#7733
#7734
#7735
#7736
#7737
#7738
#7739
#7740
#7741
#7742
#7743
#7744
#7745
#7746
#7747
#7748
#7749

#7750
#7751
#7752
#7753
#7754
#7755
#7756
#7757
#7758
#7759
#7760
#7761
#7762
#7763
#7764
#7765
#7766
#7767
#7768
#7769
#7770
#7771
#7772
#7773
#7774
#7775
#7776
#7777
#7778
#7779
#7780
#7781
#7782
#7783
#7784
#7785
#7786
#7787
#7788
#7789
#7790
#7791
#7792
#7793
#7794
#7795
#7796
#7797
#7798
#7799

#7800

#7801

#7802

#7803

#7804

#7805

#7806

#7807

#7808

#7809

#7810

#7811

#7812

#7813

#7814

#7815

#7816

#7817

#7818

#7819

#7820

#7821

#7822

#7823

#7824

#7825

#7826

#7827

#7828

#7829

#7830

#7831

#7832

#7833

#7834

#7835

#7836

#7837

#7838

#7839

#7840

#7841

#7842

#7843

#7844

#7845

#7846

#7847

#7848

#7849

#7850

#7851

#7852

#7853

#7854

#7855

#7856

#7857

#7858

#7859

#7860

#7861

#7862

#7863

#7864

#7865

#7866

#7867

#7868

#7869

#7870

#7871

#7872

#7873

#7874

#7875

#7876

#7877

#7878

#7879

#7880

#7881

#7882

#7883

#7884

#7885

#7886

#7887

#7888

#7889

#7890

#7891

#7892

#7893

#7894

#7895

#7896

#7897

#7898

#7899

APPENDIX 3 LIST OF INTERNAL RELAYS, REGISTERS FOR YASNAC J50L/J50M (Cont'd)

#7900
#7901
#7902
#7903
#7904
#7905
#7906
#7907
#7908
#7909
#7910
#7911
#7912
#7913
#7914
#7915
#7916
#7917
#7918
#7919
#7920
#7921
#7922
#7923
#7924
#7925
#7926
#7927
#7928
#7929
#7930
#7931
#7932
#7933
#7934
#7935
#7936
#7937
#7938
#7939
#7940
#7941
#7942
#7943
#7944
#7945
#7946
#7947
#7948
#7949

#7950
#7951
#7952
#7953
#7954
#7955
#7956
#7957
#7958
#7959
#7960
#7961
#7962
#7963
#7964
#7965
#7966
#7967
#7968
#7969
#7970
#7971
#7972
#7973
#7974
#7975
#7976
#7977
#7978
#7979
#7980
#7981
#7982
#7983
#7984
#7985
#7986
#7987
#7988
#7989
#7990
#7991
#7992
#7993
#7994
#7995
#7996
#7997
#7998
#7999

APPENDIX 4 CONVERSION TABLE OF DECIMAL AND HEXADECIMAL NOTATION

Hex	Dec	Hex	Dex	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
00	0	20	32	40	64	60	96	80	128	A0	160	C0	192	E0	224
01	1	21	33	41	65	61	97	81	129	A1	161	C1	193	E1	225
02	2	22	34	42	66	62	98	82	130	A2	162	C2	194	E2	226
03	3	23	35	43	67	63	99	83	131	A3	163	C3	195	E3	227
04	4	24	36	44	68	64	100	84	132	A4	164	C4	196	E4	228
05	5	25	37	45	69	65	101	85	133	A5	165	C5	197	E5	229
06	6	26	38	46	70	66	102	86	134	A6	166	C6	198	E6	230
07	7	27	39	47	71	67	103	87	135	A7	167	C7	199	E7	231
08	8	28	40	48	72	68	104	88	136	A8	168	C8	200	E8	232
09	9	29	41	49	73	69	105	89	137	A9	169	C9	201	E9	233
0A	10	2A	42	4A	74	6A	106	8A	138	AA	170	CA	202	EA	234
0B	11	2B	43	4B	75	6B	107	8B	139	AB	171	CB	203	EB	235
0C	12	2C	44	4C	76	6C	108	8C	140	AC	172	CC	204	EC	236
0D	13	2D	45	4D	77	6D	109	8D	141	AD	173	CD	205	ED	237
0E	14	2E	46	4E	78	6E	110	8E	142	AE	174	CE	206	EE	238
0F	15	2F	47	4F	79	6F	111	8F	143	AF	175	CF	207	EF	239
10	16	30	48	50	80	70	112	90	144	B0	176	D0	208	F0	240
11	17	31	49	51	81	71	113	91	145	B1	177	D1	209	F1	241
12	18	32	50	52	82	72	114	92	146	B2	178	D2	210	F2	242
13	19	33	51	53	83	73	115	93	147	B3	179	D3	211	F3	243
14	20	34	52	54	84	74	116	94	148	B4	180	D4	212	F4	244
15	21	35	53	55	85	75	117	95	149	B5	181	D5	213	F5	245
16	22	36	54	56	86	76	118	96	150	B6	182	D6	214	F6	246
17	23	37	55	57	87	77	119	97	151	B7	183	D7	215	F7	247
18	24	38	56	58	88	78	120	98	152	B8	184	D8	216	F8	248
19	25	39	57	59	89	79	121	99	153	B9	185	D9	217	F9	249
1A	26	3A	58	5A	90	7A	122	9A	154	BA	186	DA	218	FA	250
1B	27	3B	59	5B	91	7B	123	9B	155	BB	187	DB	219	FB	251
1C	28	3C	60	5C	92	7C	124	9C	156	BC	188	DC	220	FC	252
1D	29	3D	61	5D	93	7D	125	9D	157	BD	189	DD	221	FD	253
1E	30	3E	62	5E	94	7E	126	9E	158	BE	190	DE	222	FE	254
1F	31	3F	63	5F	95	7F	127	9F	159	BF	191	DF	223	FF	255

YASNAC J50 PC SYSTEM
INSTRUCTIONS

TOKYO OFFICE New Pier Takesiba South Tower, 1-16-1, Kaigan, Minatoku, Tokyo 105 Japan
Phone 81-3-5402-4511 Fax 81-3-5402-4580
YASKAWA ELECTRIC AMERICA, INC.
Chicago-Corporate Headquarters 2942 MacArthur Blvd. Northbrook, IL 60062-2028, U.S.A.
Phone 1-847-291-2340 Fax 1-847-498-2430
Chicago-Technical Center 3160 MacArthur Blvd. Northbrook, IL 60062-1917, U.S.A.
Phone 1-847-291-0411 Fax 1-847-291-1018
MOTOMAN INC.
805 Liberty Lane West Carrollton, OH 45449, U.S.A.
Phone 1-513-847-6200 Fax 1-513-847-6277
YASKAWA ELÉTRICO DO BRASIL COMÉRCIO LTDA.
Avenida Brigadeiro Faria Lima 1664-5°CJ 504/511, São Paulo, Brazil
Phone 55-11-815-7723 Fax 55-11-870-3849
YASKAWA ELECTRIC EUROPE GmbH
Am Kronberger Hang 2, 65824 Schwalbach, Germany
Phone 49-6196-569-300 Fax 49-6196-888-301
Motoman Robotics AB
Box 504 S38525 Torsås, Sweden
Phone 46-486-10575 Fax 46-486-41410
Motoman Robotec GmbH
Kammerfeldstraβe 1, 85391 Allershausen, Germany
Phone 49-8166-900 Fax 49-8166-9039
YASKAWA ELECTRIC UK LTD.
3 Drum Mains Park Orchardton Woods Cumbernauld, Scotland, G68 9LD U.K.
Phone 44-1236-735000 Fax 44-1236-458182
YASKAWA ELECTRIC KOREA CORPORATION
Paik Nam Bldg. 901 188-3, 1-Ga Euljiro, Joong-Gu Seoul, Korea
Phone 82-2-776-7844 Fax 82-2-753-2639
YASKAWA ELECTRIC (SINGAPORE) PTE. LTD.
151 Lorong Chuan, #04-01, New Tech Park Singapore 556741, Singapore
Phone 65-282-3003 Fax 65-289-3003
YATEC ENGINEERING CORPORATION
Shen Hsiang Tang Sung Chiang Building 10F 146 Sung Chiang Road, Taipei, Taiwan
Phone 886-2-563-0010 Fax 886-2-567-4677
BEIJING OFFICE Room No. 301 Office Building of Beijing International Club, 21 Jianguomenwai Avenue, Beijing 100020, China
Phone 86-10-532-1850 Fax 86-10-532-1851
SHANGHAI OFFICE Room No. 8B Wan Zhong Building 1303 Yan An Road (West), Shanghai 200050, China
Phone 86-21-6212-1015 Fax 86-21-6212-1326
YASKAWA JASON (HK) COMPANY LIMITED
Rm.2916, Hong Kong Plaza, 186-191 Connaught Road West, Hong Kong
Phone 852-2858-3220 Fax 852-2547-5773
TAIPEI OFFICE Shen Hsiang Tang Sung Chiang Building 10F 146 Sung Chiang Road, Taipei, Taiwan
Phone 886-2-563-0010 Fax 886-2-567-4677

YASKAWA ELECTRIC CORPORATION

YASKAWA

TOE-C843-12.1B
© Printed in Japan February 1997 95-3 0.5TA ②