MITSUBISHI
PROGRAMMABLE CONTROLLER
MELSEC-K

Programming Manual

IB(NA)-64408-B

MITSUBISHI
ELECTRIC

CONTENTS

1. Introduction

The MELSEC-K series is equipped with an instruction function convenient for machine control, so that even complicated control can be programmed easily.
The program language uses language for sequence control (a combination of relay symbol format and logic symbol words), so that language of the same system as for the conventional MELSEC-007B and 008 is followed, but the symbols have been partly changed to more general expressions, Example

> Example
> (1) ANS → ANB, ORS → ORB
> (2) Input/output device name: I000 → X000
> O000 → Y000

and sequence instructions and data handling instructions have been newly added.
[Example: Universal shift, conditional jump, pulse instructions, addition and subtraction, magnitude comparison, data movement (MOV), etc.]
In this way, the series K is a programmable controller with a powerful program function. Please study this manual carefully to master the programming methods, so that the functions can be used efficiently.
For program input and output operations, refer to the "Handling instructions for the programming unit".

NOTES:

(1) This programming manual covers the basic functions of the K0J, K0, K1, K2, K0E and K2E.
(2) The K2CPU-S3 is provided with special funcitons.
Refer to the "Instruction Manual — Special Functions of the K2CPU-S3" for the programming of the special functions.
(3) The K0J is provided with special functions.
Refer to the "Instruction Manual — Special Functions of the K0J" for the programming of the speicial functions.

2. Hardware composition and operation

The hardware outlines are described to deepen the understanding for programming.

2-1. Hardware composition

2-1-1. Hardware system composition

The units shown in the following figure are attached to the base unit to compose the system.

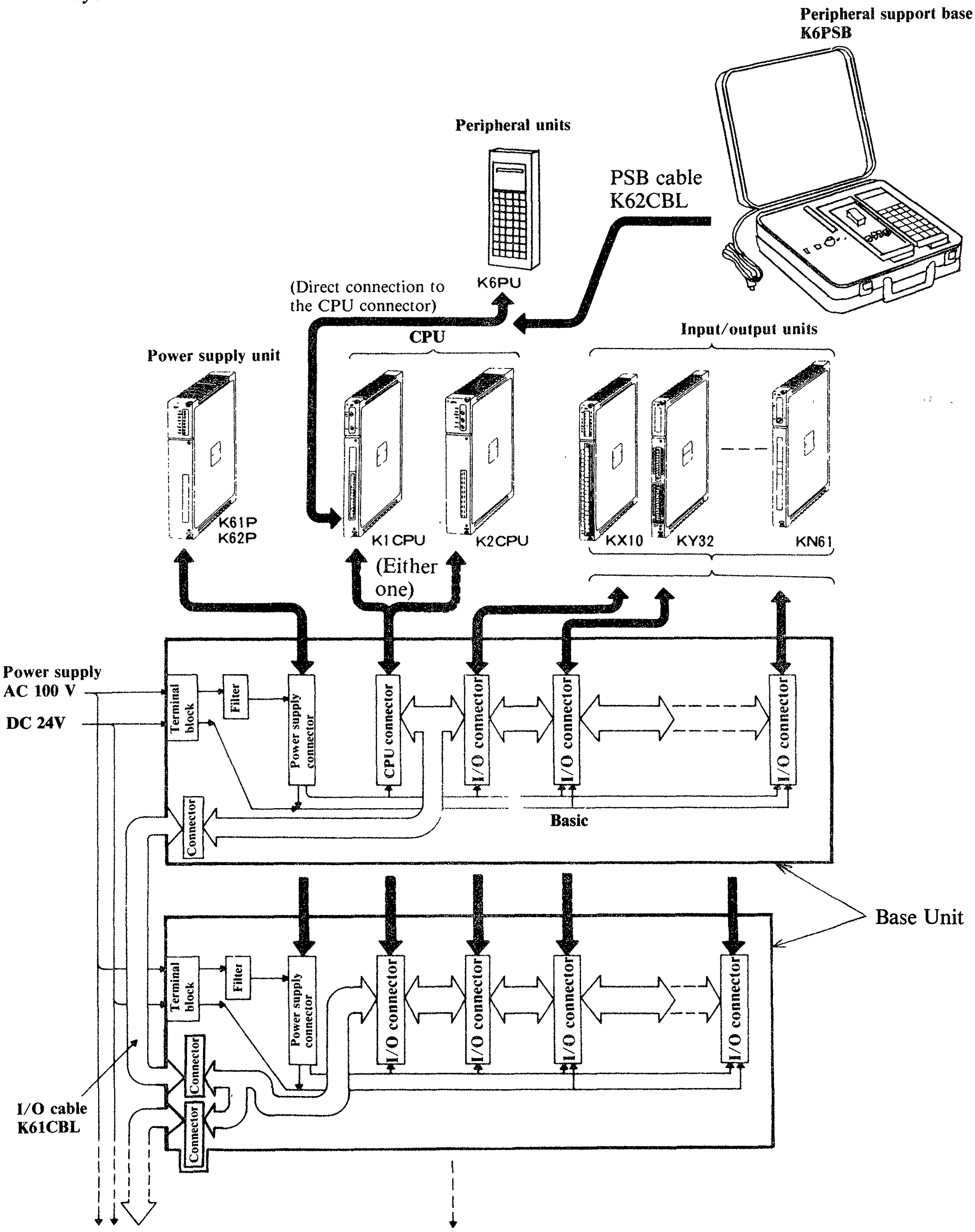


Fig. 2-1. Hardware system composition

2-1-2. CPU system composition

The CPU operation functons are shown in the following figure.

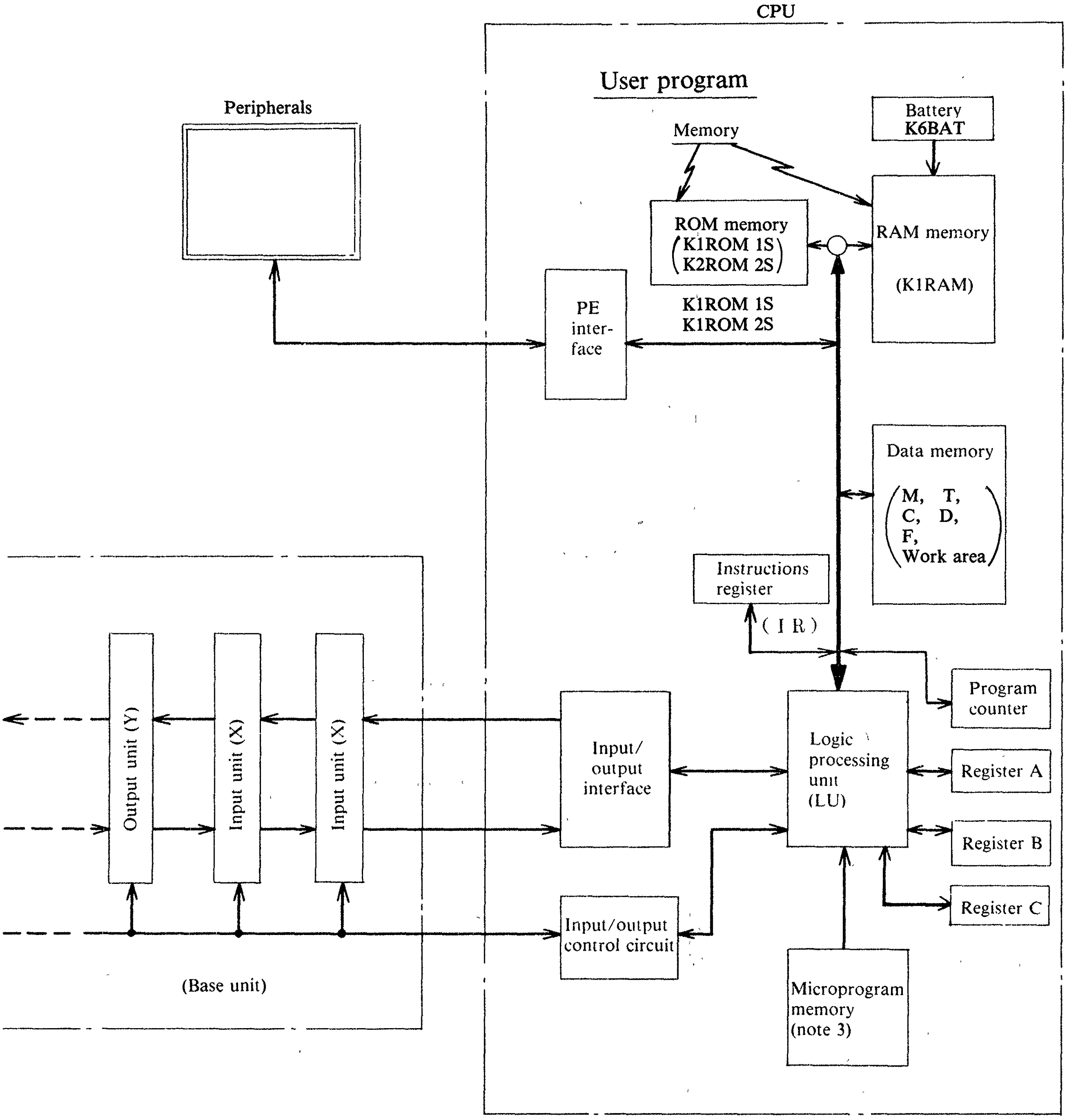


Fig. 2-2. CPU system composition

Notes:

1. Program writing for the user program memory uses the K1RAM. Simultaneous use of ROM and RAM is not possible.
2. The CPU system composition does not coincide with the hardware.
3. A discreet circuit is obtained for K2CPU.

2-2. Logic operations for instructions

2-2-1. Basic operation of the programmable controller

The basic operation of the programmable controller is shown in the figure on the right. This operation is the same as for a computer of the stored program method, and the hardware composition also is similar, so that this is called a programmable controller of the quasicomputer type.

①First, the contents of the program counter showing the program step number are given to the memory address, and the contents are read.

② The read memory contents are translated, and discrimination is executed for AND, OR, and other instructions as well as input and output numbers.

③ According to the discriminated source (logic input signal to be included) and the instructions, AND, OR, and other logic operations as well as output operation are executed.

④ 1 is added to the program counter contents, and specification of the next step number is executed.

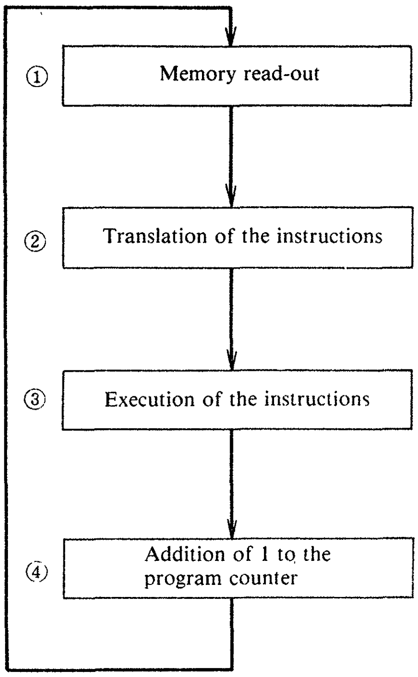

①
Memory read-out
②
Translation of the instructions
③
Execution of the instructions
④
Addition of 1 to the program counter

Fig. 2-3. Basic logic operations

The above 4 operations are executed for each step number, i.e. each instruction in the programmable controller, and sequential execution is executed in the sequence of the numbers for all step numbers stored in the memory. This is called scanning.

The figure on the right shows the concept for scanning. The above 4 operations are executed in the sequence of the program numbers according to the program counter contents, return to number 0 is executed when the END instruction is reached, and then the sequence is repeated. This operation is called scanning, and with one scanning, all inputs written into the memory are taken in, logic operations are executed, and the logic operation results are put out to all outputs.

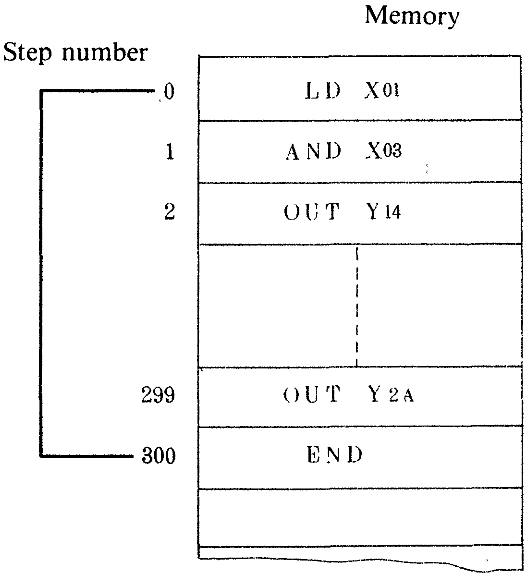


Fig. 2-4. Scanning operation

The scanning speed about corresponds to the electromagnetic relay response speed, and this is called the logic operation speed.

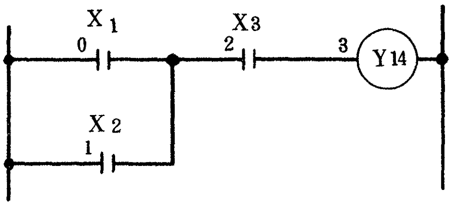
As shown in the above, serial arithmetic operations are executed in the programmable controller, but as the logic operation speed is high, all inputs are taken in, and logic operation results are put out to all outputs, the operation from the outside appears to be the same parallel arithmetic operation as for electromagnetic relays.

2-2-2. Arithmetic principle

The arithmetic principle is explained with the circuit shown in the figure on the right as an example.

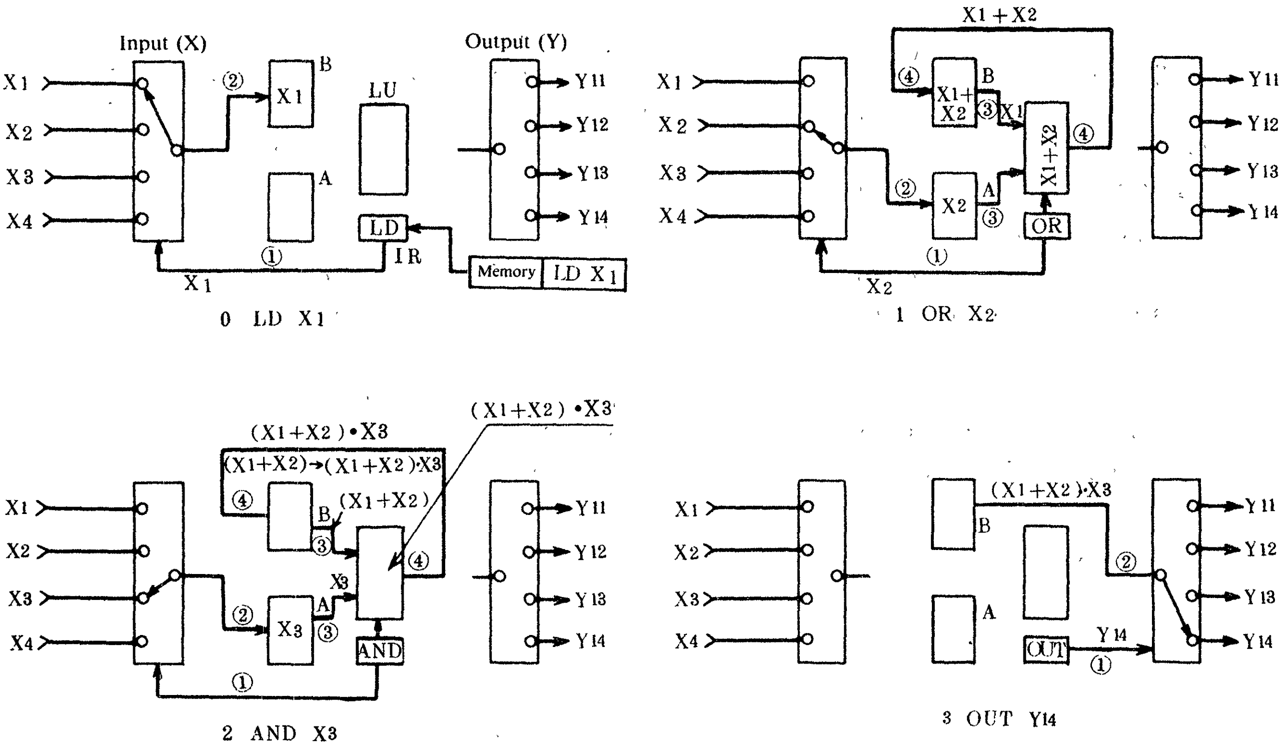
The program for this circuit is as follows with MELSEC-K.

The execution process for the circuit example (a) is shown in (b).



(a) Circuit example

Step number	Instructions	Input/output number	
0	LD	X1	Input of the on-off status of X1
1	OR	X2	Input of the on-off status of X2 and OR operation with X1. X1+X2
2	AND	X3	Input of the on-off status of X3 and AND operation with (X1 + X2). (X1 + X2). X3
3	OUT	Y14	Output of the operation result (X1 + X2) X3 to Y14.



0 LD X1

1 OR X2

2 AND X3

3 OUT Y14

X: Input selection part
Y: Output selection part
A: Register A (answer register)
B: Register B (auxiliary register)
LU: Logic processing unit
IR: Instructions register
(b) Execution process

Fig. 2-5. Arithmetic principle

Explanation of the arithmetic principle

The execution process for this program and this hardware compostion is as follows.

Step

0 LD X1

①The program number 0 is read from the memory, the instruction LD is stored in the instructions register IR, and the input/output number X1 is sent to the input selection part.

*The same applies for subsequent program.

② X1 is selected by the input selection part X.

③ The on-off status of X1 is taken in, and it is stored in register B via register A.

1 OR X2

① X2 is selected by the input selection part X.

② The on-off status of X2 is taken in and stored in register A.

③ The contens of X2 in register A and X1 in register B are sent to the logic processing unit, LU, OR operation is executed, and the operation result for (X1 + X2) is obtained.

④ This operation result (X1 + X2) is sent to register B, where it is stored. Register B becomes the answer register.

2 AND X3

① X3 is selected by the input selection part X.

② The on-off status of X3 is taken in and stored in register A.

③ The contents of (X1 + X2) in register B and X3 in register A are sent to the logic processing unit LU, AND operation is executed, and the operation result for (X1 + X2) X3 is obtained.

④ This operation result (X1 + X2) X3 is sent to register B, where it is stored.

3 OUT Y14

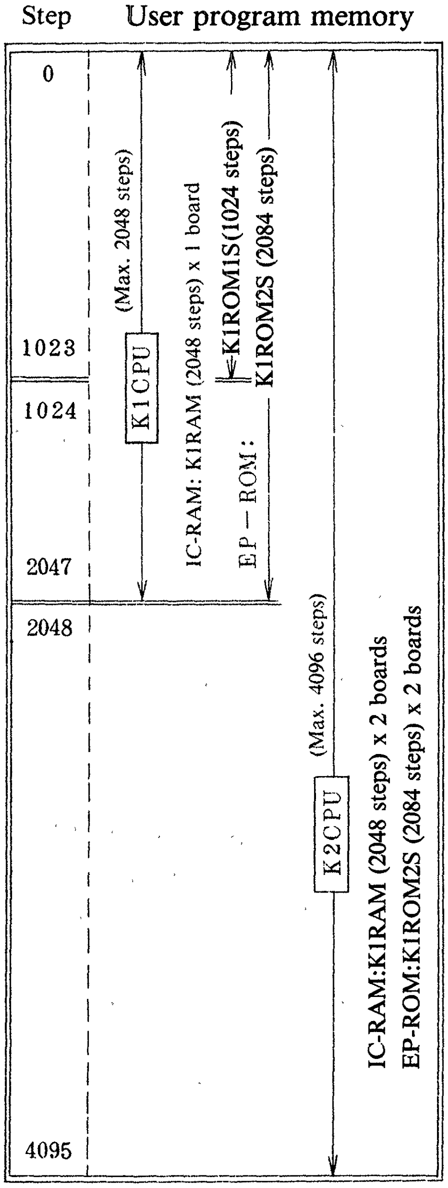
① Y14 is selected by the output selection part Y.

② The operation result (X1 + X2) X3 from register B is sent to output Y14.
Y14 has an output register, where this operation result is stored and kept.
The above example treated X and Y as input and output, but seen from the logic processing unit (LU), X and Y are preceding units providing conditions or data for logic processing and units for output of the operation results. These units seen from the LU are called "devices". There are the following types of devices:-

X	Input	Input and output by the LU.
Y	Output	
M	Temporary memory	
T	Timer	
C	Counter	
F	External failure memory	
D	Data register	
K	Constant	Only input by the LU.

2-3. Program memory and data memory

(1) User program memory



Step
User program memory
0
1023
1024
2047
2048
4095
(Max. 2048 steps)
K1CPU
IC-RAM: K1RAM (2048 steps) x 1 board
K1ROM1S(1024 steps)
EP－ROM: K1ROM2S (2084 steps)
(Max. 4096 steps)
K2CPU
IC-RAM:K1RAM (2048 steps) x 2 boards
EP-ROM:K1ROM2S (2084 steps) x 2 boards

Note: The EP-ROM socket is used jointly for 1K step and 2K step.

(2) Data memory

K1CPU / K2CPU	
M 0 ~ M253	Temporary memory (M) 254 points
M254	Battery trouble
M255	RUN signal
T.C 0 ~ T.C127	Timer, counter (T , C) 128 points
F 0 ~ F 99	External failure memory (F) 100 points
D 0 ~ D95	Data register (D) 96 points

Notes:

1. The timer and counter setting values are stored in the user program.
2. Output of the present value is possible for T, C, D,

(3) Latch contents of K2CPU

No.	Latchabel momory	Unlatchabel memory
1	None	M0 ~ 255, T. C0 ~ 127 D0 ~ 95
2	M128 ~ M253	M0 ~ 127, 254, 255, T.C0 ~ 127 D0 ~ 95
3	M0 ~ 253	M254, 255, T.C0 ~ 127 D0 ~ 95
4	M128 ~ M253, T.C64 ~ 127, D32 ~ 95	M0 ~ 127, 254, 255, T.C0 ~ 63, D0 ~ 31
5	M0 ~ 253, T.C0 ~ 127 D0 ~ 95	M254, 255

Note: Nos. 1 ~ 5 are set with the selector switch.

(4) Total number of pints for input and output (hexadecimal symbol "#").

K 1 C P U	K 2 C P U
Max. 256 points 00# ~ FF#	Max. 512 points 000# ~ 1FF#

Fig. 2-6. Contents of the user memory

2-4.Temporary memory M (the same handling applies to the unused F and Y) → Refer to item 3-2.

The temporary memory corresponds to auxiliary relays without output to the outside. There is no limitation on the number of times which this M can be used as an internal contact. The max. number for M is 254 points, but the external failure memory F and the unused output Y also can be used in the same way.

M254 and M255 have fixed applications. Do not use these M carelessly for other purposes.

① M254 battery alarm

This becomes "1" when the battery voltage drops because of discharge. However, as there is still sufficient time after this has become "1", it is treated as an alarm, and the battery should be exchanged within one month. For the program, this is handled as a general M, and 1 point of the output Y should be used for lead-out as an external signal. As this becomes "1" also when the battery is removed, M254 can not be used as a general M with EP-ROM operation.

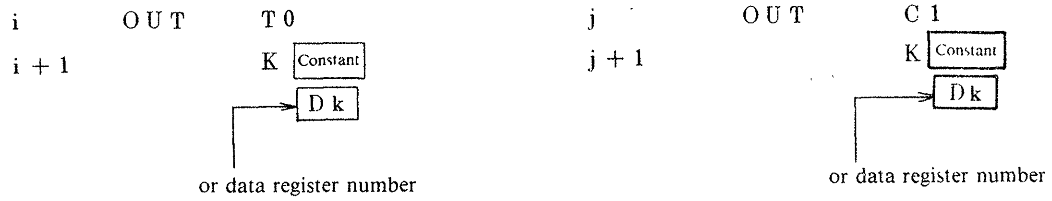
② M255 RUN signal

When the CPU "RUN" signal is required as an external signal, 1 point of the output Y should be used for lead-out. However, in the same way as for M254, use as a general M is not possible.

③ In case of K2CPU, use "LATCH" as memory holding at the time of power failure. One half of the M (M128 to M253) can be used as latch memory when the "LATCH" switch is switched on. When the switch is set to OFF, it becomes a normal temporary memory. Resetting can be executed summarily with the CPU "RESET" switch. In case of K1CPU, there is no summary resetting as described above. In this case, the latch function for 16 points/32 points/64 points (setting by connectors in the unit) becomes possible by connection of an optional latch unit (KL61) to the I/O unit connector for latching of the output Y. Add "n" to the respective input/output number and execute handling as a latch output Y"n" for the program. This Y can not be taken out directly to the outside. The latch contact is programmed as Xn.

2-5.Timer, counter T, C → Refer to item 3-5.

Timer and counter can use 128 points together. The numbers 0 to 127 are specified consecutively in common for T and C. Setting value of T, C must be written in the step following the OUT instruction.



When the above i + 1 or j + 1 are omitted, (INS. SET ERR) occurs.

The timer setting value can be set in units of 0.1 sec from 0.1 to 999.9 sec. The accuracy is $^{\ 0}_{-0.1}$ sec in regard to the setting value.

1. As the setting for the min. unit of 0.1 sec varies from 0 to 0.1 sec, it may become 0 sec accroding to the timing. In this case, set the setting value to 0.2.
2. For a setting in excess of 999.9 sec., execute the lower digits with the timer, and produce a timer by driving a counter with frequency division for that timing.

Example: Execute counter input by 1 min time up, and use the number of required minutes as the setting value.

3. Timer resetting is not possible. However, the present value is cleared to 0 at the time of power ON.

Counter setting values from 1 to 9999 can be used. Intermediate resetting is possible by RST instructions. Output or arithmetic operations are possible for the present value of timer and counter by MOV and other data instructions.

2-6. External failure memory F → Refer to item 3-4.

F has the same function as the temporary memory, and when the optional external failure monitor unit (KN61) is connected to the attachment position for the I/O unit, F is scanned cyclically and the F number corresponding to the scan sequence position which has become "1" is displayed numerically. When an external alarm is put out together with the display, the respective F is handled in the same way as a general M, and alarm output can be executed via output Y. The display of the F number of the external failure is cleared by the reset switch of the monitor surface after Fi "0", and the next Fj "1" is displayed.

When no external failure monitor is used, handling is executed exactly in the same way as for a general M, so that use as a temporary memory is possible.

2-7. Data register D → Refer to item 3-3.

2-8. Input/output unit → Refer to item 3-1.

3. Sequence until programming

3-1. Assignment of the input/output numbers

The assignment of the input/output numbers is decided depending on the I/O connector numbers of the base unit and the types and arrangement of the input/output units installed there.
As an example, the decision of the input/output numbers will be shown for the following unit composition.

Base unit	Basic	K18B	
	Extended	K68B	
Power supply unit		K62P	
CPU unit		K1CPU	
External failure monitor		KN61	
Input unit (for data)		KX31	(32 points)
Output unit (for data)		KY31	(32 points)
Input unit (DC 24 V)		KX30	(16 points)
Input unit (AC 100 V)		KX10	2 boards (16 points)
Output unit (AC 100 V)		KY11	2 boards (16 points)
Output unit (AC 100 V)		KY22	(16 points)
Input/output composite unit		KH11	(8 points + 8 points)
Timer unit		KT61	(16 points)
Latch unit (32 points switching)		KL61	(32 points)

When the units are selected in this way, the unit arrangement becomes as shown in Fig. 3-2. for the base unit K18B and K68B.

(1)Unit arrangement

When the unit arrangement is entered into the "Unit arrangement table sheets Nos.1/2 and 2/2", it becomes as shown in Fig. 3-3-1, 3-3-2. When the I/O units are arranged sequentially from 0 on the number of points of each I/O unit is specified and entered on the lower left of the table, so that the number of required points according to the configuration is obtained. The purpose of the input/output is entered in the application column and the upper 2 digits of the I/O number are entered. Use sequential numbering with left justification in units of 16 points, i.e., 00, 01, 02, ... 0F.

Note: The input/output numbers are hexadecimal numbers. (Indication method:□□□#)
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F ...The figure increases by one digit when F is reached.

(2)Definition of the input/output numbers (drawing up of the input/output list)

When the I/O adress "0A" is decided as shown in Fig. 3-1 (decision in hexadecimal units as 1, 2, ..., to"0A", the numbers for each input/output are necessarily defined on a 1 point correspondence.

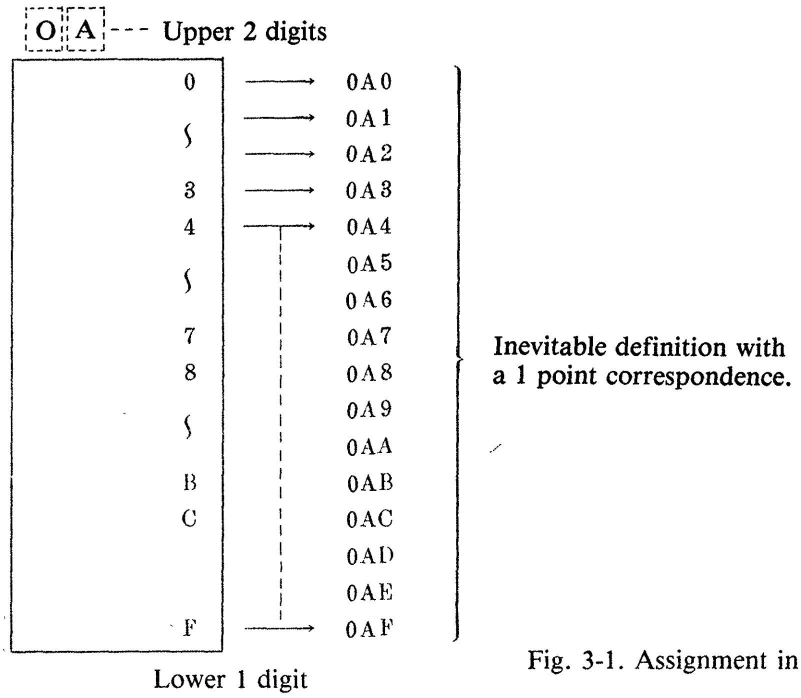


Fig. 3-1. Assignment in the unit

Under reference to Fig. 3-3., the input/output list is defined sequentially with a 1 point correspondence and complete the input/output list. (Execute entry with use of sheets 2.)

Caution points for adress assignment:

①Occupation of 16 points:16 points for the external failure monitor, timer, blank (only in the case of intermediate blanks), and latch unit must be selected.

Note: Note that the entire arrangement is shifted if a unit which occupies 32 points or 64 points is inserted into the blank afterwards.

②Occupation of 32 points or 64 points

Select 32 points/64 points for multipoint storage input/output units, latch units →

Refer to the lower left of the unit arrangement table sheets.

(3)Cations for input assignment

There is no special principle, but when assignment is executed under consideration of the following points, programming and checking will be facilitated and the arrangement of the wiring diagram will become easier.

①Collect, for example, push button switches together, then limit switches, etc. so that the input are grouped in similar kind.

②Assign sequentially according to the divice numbers within each type.

③When there are remaining points, assign on device or machine per input unit.

④For devices which can be connected externally, assign after connection.

⑤Assign high noise inputs as far as possible from the CPU, i.e. with later numbers.

(4)Cautions for output assignment

In the same way for input assignment, there is no special principle, but attention should be paid to the following points.

①Collect output devices of the same type together.

②Assign in the sequence of device numbers for the same kind.
③When there are remaining points, assign one device or machine per output unit.
④Assign output devices with high noise as far as possible with later numbers.
⑤Assign consecutive numbers to related output devices, e.g., motor forward-reverse contacts.

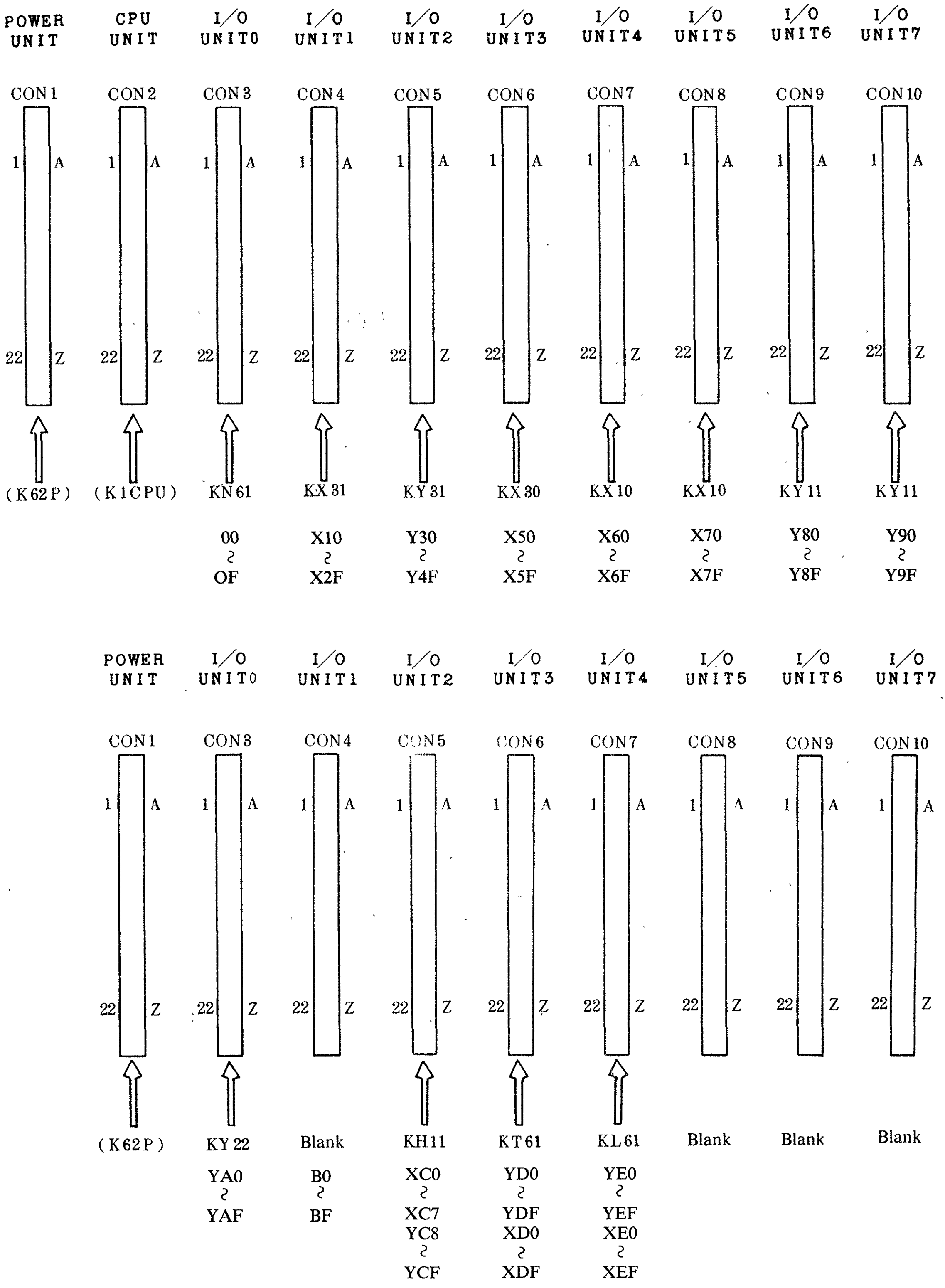


Fig. 3-2. Base unit connector arrangement

MELSEC-K

Unit arrangement table

Fig. 3-3-1. Sample system

Approved	Drawn up	Sheet No.
	Mitsubishi 80-10-10	1/2

Note: Limit of the number of input/output points

K 1 C P U→0 0 0 #~0 F F # (256 points) For K12B, K22B ⟹

K 2 C P U→0 0 0 #~1 F F # (512 points)

For K15B, K25B ⟹

For K18, K28B ⟹

Base connector name	POWER UNIT	CPU UNIT	I/O UNIT 0	I/O UNIT 1	I/O UNIT 2	I/O UNIT 3	I/O UNIT 4	I/O UNIT 5	I/O UNIT 6	I/O UNIT 7
Installed unit type name	K62P	K1CPU	KN61	KX31	KY31	KX30	KX10	KX10	KY11	KY11

Upper 2 digits of the I/O number	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application
Unit with occupation of 64 points Unit with occupation of 32 points Unit with occupation of 16 points • 16 point device • Blank (not attached) • External failure monitor (KN61) • Latch unit (KL61) (for 16 point switching) • Timer unit (KT61)	00 0~3 4~7 8~B C~F	KN 61	01 0~3 4~7 8~B C~F	Input Data (Ⅰ)	03 0~3 4~7 8~B C~F	Output Data (Ⅳ) Output Data (Ⅴ)	05 0~3 4~7 8~B C~F	DC24V Input Proximity SW Input	06 0~3 4~7 8~B C~F	AC100V Input Push button SW	07 0~3 4~7 8~B C~F	AC100V Input Limit switch	08 0~3 4~7 8~B C~F	AC100V Output Relay drive	09 0~3 4~7 8~B C~F	AC100V Output Contactor drive
32 point device (KX3 1 , 4 1 , KY3 1 , 41) • Latch unit (KL61) (for 32 point switching)	0~3 4~7 8~B C~F		02 0~3 4~7 8~B C~F	Input Data (Ⅱ) Input Data (Ⅲ)	04 0~3 4~7 8~B C~F	Output Data (Ⅵ)	0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F	
• 64 point device (KX3 2 , KY3 2) • Latch unit (KL61) (for 64 point switching)	0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F	
	0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F	

MELSEC-K

Unit arrangement table

Fig. 3-3-2. Sample system

Approved	Drawn up
	Mitsubishi 80-10-10

Sheet No. 2/2

Note: Limit of the number of input/output points

K1CPU→000#~0FF#(256 points)

K2CPU→000#~1FF#(512 points)

For K68B ⇨

Base connector name	POWER UNIT	I/O UNIT 0	I/O UNIT 1	I/O UNIT 2	I/O UNIT 3	I/O UNIT 4	I/O UNIT 5	I/O UNIT 6	I/O UNIT 7
Installed unit type name	K62P	KY 22	Blank	KH 11	KT 61	KL 61	–	–	–

Upper 2 digits of the I/O number

Unit with occupation of 64 points

Unit with occupation of 32 points

Unit with occupation of 16 points

- 16 point device
- Blank (not attached)
- External failure monitor (KN61)
- Latch unit (KL61) (for 16 point switching)
- Timer unit (KT61)

- 32 point device (KX31, 41, KY31, 41)
- Latch unit (KL61) (for 32 point switching)

- 64 point device (KX32, KY32)
- Latch unit (KL61) (for 64 point switching)

I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application
OA 0~3 4~7 8~B C~F	AC100V Output (Triac) Bulb drive	OB 0~3 4~7 8~B C~F	Not used	OC 0~3 4~7 8~B C~F	AC100V Input (Signal lamp) AC100V Output (Change-over switch)	OD 0~3 4~7 8~B C~F	Analog timer	OE 0~3 4~7 8~B C~F	Latch ----	0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F	
0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		OF 0~3 4~7 8~B C~F	Latch	0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F	
0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F	
0~3 4~7 8~B C~F		0~3 4~7 8~B C		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F		0~3 4~7 8~B C~F	

Sheet form 1-2

MELSEC-K

Input/output list

Fig. 3-4-1. Sample system

Approved	Drawn up
	Mitsubishi 80-10-10

Sheet No. 1/3

Base (I/O connector name)	I/O unit type name (number of points/unit)	Input/output number		Device number	Name	Remarks (Connection terminal wire type, etc.)
Basic base K 18 B (I/O Unit 0)	KN 61 (16 points)	0	0			External failure monitor (no external connections)
			1			
			2			
			3			
			4			
			5			
			6			
			7			
			8			
			9			
			A			
			B			
			C			
			D			
			E			
			F			
(ditto 1)	KX 31 (32 points)	1	0	BCD First digit 1	Input Data (I)	Operation panel
			1	2	BCD x 4 digits	Input Data(I)
			2	4	Digital switch input	To the digital switch
			3	8		
			4	BCD Second digit 10		Connection by 0.3 mm² IV
			5	20		
			6	40		
			7	80		
			8	BCD Third digit 100		
			9	200		
			A	400		
			B	800		
			C	BCD Fourth digit 1000		
			D	2000		
			E	4000		
			F	8000		

(continued)

Fig. 3-4-2. Sample system

Approved	Drawn up
	Mitsubishi 80-10-10

Sheet No 2/3

(continued)

Base (I/O connector name)	I/O unit type name (number of points/unit)	Input/output number		Device number	Name	Remarks (Connection terminal wire type, etc.)
(ditto 1)	(KX31)	2	0	BCD First digit 1	Input Data(Ⅱ)	Continued
			1	2	BCD x 2 digits	
			2	4	Digital switch input	
			3	8		
			4	BCD Second digit 10		
			5	20		
			6	40		
			7	80		
			8	BCD First digit 1	Input Data(Ⅲ)	
			9	2	BCD x 2 digits	
			A	4	Digital switch input	
			B	8		
			C	BCD Second digit 10		
			D	20		
			E	40		
			F	80		
(ditto 2)	KY31 (32 points)	3	0	BCD First digit 1	Output Data(Ⅳ)	Same as above
			1	2	BCD x 2 digits	
			2	4	Numerical display	
			3	8		
			4	BCD Second digit 10		
			5	20		
			6	40		
			7	80		
			8	BCD First digigt 1	Output Data(Ⅴ)	
			9	2	BCD x 2 digits	
			A	4	To the numerical setting unit	
			B	8		
			C	BCD Second digit 10		
			D	20		
			E	40		
			F	80		

(continued) (The following page for KY31 is omitted.)

MELSEC-K

Input/output list

Fig. 3-4-3. Sample system

Approved	Drawn up
	Mitsubishi
	80-10-10

Sheet No. 3/3

Base (I/O connector name)	I/O unit name (number of points/ unit)	Input/output number		Device number	Name	Remarks (Connection terminal wire type, etc.)
ditto (3)	KX 30 (16 points)	5	0	PX 1	Proximity switch for timing of axis A	Installation on the machine side Proximity switch (shielded cable)
			1	PX 2	Proximity switch for timing of axis A	
			2	PX 3	Proximity switch for position detection of axis A	
			3	PX 4		
			4	PX 5		
			5	PX 6		
			6	PX 7		
			7			
			8	PX 11	Proximity switch for timing of axis B	
			9	PX 12	Proximity switch for timing of axis B	
			A	PX 13	Proximity switch for position detection of axis B	
			B	PX 14		
			C	PX 15		
			D	PX 16		
			E	PX 17		
		↓	F			
ditto (4)	KX 10 (16 points)	6	0	PB 1	Axis A Start PB	To the monitoring operation panel ($1.25\,\text{mm}^2$ (Y))
			1	PB 2	Stop PB	
			2	PB 3	Forward PB	
			3	PB 4	Reverse PB	
			4	PB 5	Manual ON (SW)	
			5	PB 6	Automatic ON (SW)	
			6			
			7			
			8	PB 11	Axis B Start PB	
			9	PB 12	Stop PB	
			A	PB 13	Foward PB	
			B	PB 14	Reverse PB	
			C	PB 15	Manual ON (SW)	
			D	PB 16	Automatic ON (SW)	
			E			
		↓	F			

3-2. Temporary memory M

Enter the temporary memory list using "Sheet form 4".

Assignment is not generally executed before the control circuit design, but at the time of deisgn completion or at the time of programming. Normally, the temporary memory is decided during the circuit design, so that the design proceeds controlling the number of assignment points, and after program completion, entry into the list should be executed for arrangement.

Since M is common to the universal shift, assignment of numbers separated from the M numbers used for the circuit is convenient. In addition, the following two items are used in the same way as M.

M254 (battery alarm) The alarm signal ("1") is given when the battery voltage drops below the specified value. The signal is "1" when no battery is used (normal voltage ... 0, down voltage ... 1).
M255 (RUN signal) As this signal becomes "1" when the CPU is in RUN status, this M is used as output signal path when the signal is to be taken to the outside (RUN ... 1, STOP ... 0).

For use as an external signal, execute programming for take-out via the output unit in the same way as for a general M.

Also, as these M differ from general M, do not assign them erroneously as general M.

3-3. Data register D

Normally, a decision is made in the same way as for temporary memories while the program is being drawn up, but it should be decided in advance to use numbers close to each other for data of the same kind, and the details can be determined together with drawing up of the program.

Enter the data register list using "Sheet form 5".

1 data item consists of 16 bits.

These data are handled either as binaries (BIN) or as binary coded decimals (BCD).

Accordingly, the max. size for data of 16 bits is as follows:

BCD: Binary coded decimals (decimal number of 4 digits x 4 bits)
└→Indicating 1 BCD digit.

BIN: Binary number (Binary number of 16 digits. However, the numbers handled here are up to the decimal number 9999.)

As these data consist of 16 bits each, 2 data con not be used jointly within 16 bits.

For example, even when a BCD is a 2 digit number, the upper 2 digits can not be used jointly with other data. Even the smallest numerical value occupies 16 bits.

3-4. Externa failure memory F

This is abbreviated as failure memory, and the "F" of failure is taken as the symbol. The handling is in the same as for M, but when an external failure monitor (KN61) is used in the case of F, it is always possible to execute a search for "F = 1" and to display the F numbers for which "1" has been established on the external failure monitor.

Use "Sheet form 6" for the failure memory list. As a large provision is made of 100 points for the fauilure points F0 to F99, the use value will be increased when entry is made such that the failure contents are classified, and the index numbers arranged so that the trouble contents can be recognized directly from the number.

After classification as described above, detailed entry into the list should be executed while programming the failure conditions. At the time of program completion, the list should be checked once again, and the numbers should be assigned in a way convenient for trouble-shooting.

The handling method for the program will be described later.

3-5. Timer, counter T, C

(1)Timer, counter list

Use "Sheet form 7". The timer and counter are used jointly, and up to 127 points are possible. Accordingly, discrimination between timer and counter must be made with T or C in the T/C selection column of the list.

(2)Timer assignment

First assign the timers required by the control specifications. Next, assign the timers required at the time of design of the control circuits.

When the timer setting time is the same, a timer can be used repeatedly, as long as the operation does not overlap. Accordingly, when the number of timer points is not sufficient, this technique should be used by employing the same setting time as far as possible.

At the time of timer number assignment, also enter the setting time limit into the list. Time setting is executed in units of 0.1 sec.

(3)Counter assignment

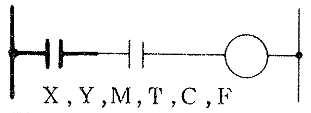
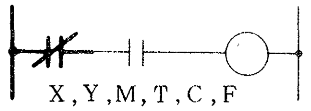
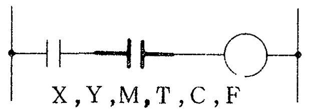
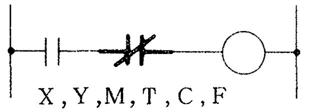
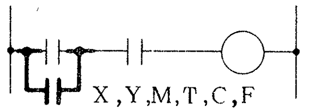
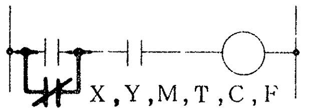
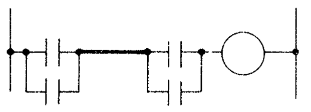
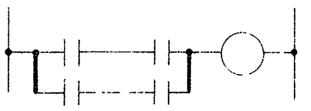
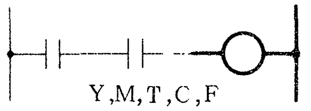
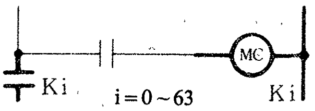
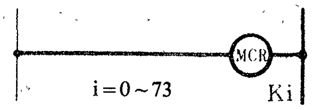
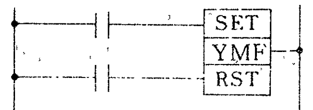
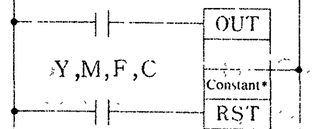
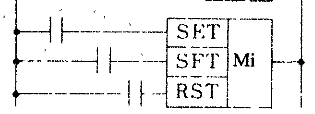
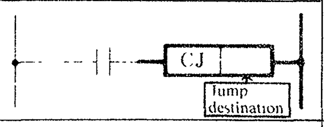
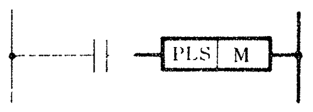
Assign the counters required by the control specifications. In the same way as for timers, repeated use with the same value is possible as long as the operations do not overlap. It is convenient to enter both count input and reset input into the list.

4. Instruction functions

4-1. Instructions list

Sequence instructions

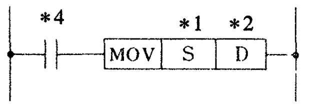
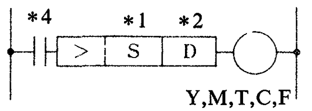
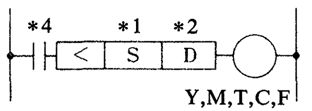
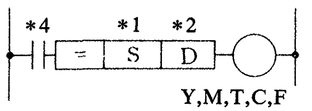
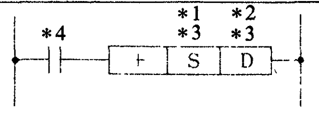
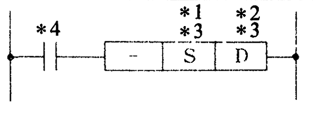
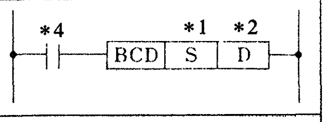
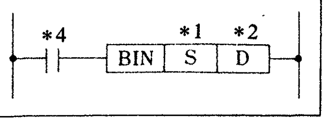
Table 4-1. Sequence instructions list

No.	Instruction symbol(name)	Function	Drawing representation
1	LD Load	Logic operation start (Contact a Operation start)	
2	LDI Load inverse	NOT Logic operation start (Contact b Operation start)	
3	AND AND	Logical product (Contact a Series connection)	
4	ANI AND inverse	Logic NAND (Contact b Series connection)	
5	OR OR	OR (Contact a Parallel connection)	
6	ORI OR inverse	Logic NOR (Contact b Parallel connection)	
7	ANB AND block	Logic block AND Series connection between blocks	
8	ORB OR block	Logic block OR Parallel connection between blocks	
9	OUT OUT	Output	
10	MC Master control	Master control start	
11	MCR Master control reset	Master control reset	
12	SET Set	Flip-flop set	
13	RST Reset	Flip-flop counter reset	
14	SFT Shift	Temporary memory shift	
15	CJ Conditional jump	Conditional jump	
16	PLS Pulse	Pulse	
17	NOP NOP	No processing	Program delete or for space
18	END END	Program completion	Be sure of enter END at the end of program

Note: * Constant is for C.

Data instructions

Table 4-2. Data instructions list

No.	Instruction symbol(name)	Function	Drawing representation
1	MOV Move	Data transfer S →D	
2	> Larger	Magnitude comparison S > D	
3	< Smaller	Magnitude comparison S < D	
4	= Equal	Coincidence S = D	
5	+ Plus	Addition S + D →D	
6	Minus	Minus D S →D	
7	BCD BCD	Conversion from BIN to BCD Conversion from S to BCD to D	
8	BIN Binary	Conversion from BCD to BIN Conversion from S to BIN to D	

Notes:
1. *1 indicates the source.
2. *2 Indicates the destination.
3. *3 Negative numbers are not handled.
4. As input signal to start data operation, X,Y,T,C,M or F is indicated.
5. >, <, = are equivalent to a contact while other instructions are equivalent to coils.

4-2. Types and composition of instruction words

(1)Instruction types

MELSEC instructions are composed with a basic word length of 16 bits (2 bytes = 1 word), and according to the instruction type, there are the 3 lengths of 1 step instructions, 2 step instructions, and 3 step instructions.

Corresponding to this, 1, 2, or 3 program memory steps are required.

① Sequence operation instructions	LD、LDI、AND、ANI、OR、ORI、ANB、ORB
② Output instructions	MC、MCR、SFT、PLS、SET、RST、OUT、CJ
③ Data instructions	MOV、>、<、=、+、−、BCD、BIN
④ Others	NOP、END

(2)Composition of the instruction words

(1 step instructions)

First step	Instruction code	Input/output source	Input/output number

Instruction code	Input/output source	Input/output number	
LD、 LDI AND、ANI OR、 ORI	X、Y、M、F T、C	X、Y000#～ 0FF#(K1) 1FF#(K2) M0～253 (Note: M254 Battery alarm, M255 RUN signal) F0～99	
OUT、 SET、RST	Y、M、F、T、C	T、C0～127	No SET, RST for T. No SET for T.
SFT、PLS	M	M0～253	
ANB、ORB NOP、END	None	None	
MC、 MCR	K	"i" is attached for index use.	i = 0～63

(2 step instructions) OUT T、OUT C、 CJ

First step	Instruction code	Ⓐ	Ⓑ
Second step	Ⓒ		

(2 step instructions)	Ⓐ Ⓑ	Ⓒ
OUT、T、C	Same as 1 step instruction OUT Instruction code	K (constant) or D (Data register number)
CJ		K (jump destination step)

(3 step instructions) MOV、>、<、=、+、−、BCD、BIN

First step	Data instruction code	Auxiliary code (type)
Second step	Source	[K (constant) or D (data register number)]
Third step	Destination	[K (constant) or D (data register number)]

(3) Instruction composition list

Instruction	Input/output source (device)	Input/output number
LD	X	0#~FF# (1FF#K2)
LDI	Y	0#~FF# (1FF#K2)
	M	0 ~253
	F	0 ~99
	T	0~127
	C	
AND	X	0#~FF# (1FF#K2)
ANI	Y	0#~FF# (1FF#K2)
	M	0 ~253
	F	0 ~99
	T	0~127
	C	
OR	X	0#~FF# (1FF#K2)
ORI	Y	0#~FF# (1FF#K2)
	M	0 ~253
	F	0 ~99
	T	0~127
	C	
ANB	—	—
ORB	—	—
MC MCR	—	Index number (Ki) 0 ~ 63
NOP	—	—
END	—	—

Instruction	Input/output source (device)	Input/output number
OUT	Y	0#~FF# (1FF#K2)
	M	0 ~ 253
	F	0 ~ 99
	T	0~ 127
	C	
SET	Y	0#~FF# (1FF#K2)
	M	0 ~ 253
	F	0 ~ 99
RST	Y	0#~FF# (1FF#K2)
	M	0 ~ 253
	C	0 ~ 63
	F	0 ~ 99
SFT	M	0 ~ 253
PLS	M	0 ~ 253
CJ	—	Step number 0 ~ 2047 (4095 K2)

Notes:

1. M254 Battery alarm
   M255 RUN signal
   Fixed. Handling is the same as for general M.
2. (□ K2) indicates the value for the K2CPU.
3. □ #indicates a hexadecimal number.

Fig. 4-3. Instruction composition list

4-3. Sequence instructions

The instruction functions must be understood at the time of programming.

Programming can be executed with electromagnetic relay symbols as well as logic symbols.

4-3-1. AND, ANI ... Series connection operation

AND has the operation function for series connection of the contact a, and ANI has the operation function for series connection for the contact b.

Both execute the series connection operation (AND) for the previous operation results.

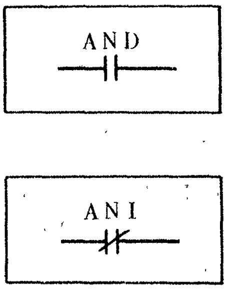
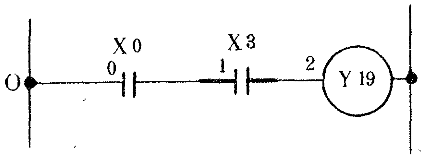


Fig. 4-1. Symbols for AND and ANI

The figure shows the AND operation, and the operation of X3 with the bold line is indicated as

1 AND X3

At this time, the X3 operation is executed with X0, and the X0 X3 operation is executd by the instruction AND X3. In this case, X0 is the operation result up to this time. Also, the X0 part has a series connection, but as there is no operation result before X0, LD X0 is obtained instead of AND X0.



$Y19 = X0 \bullet X3$

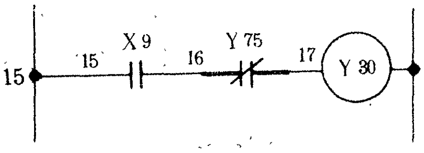
Fig. 4-2. AND operation

The figure shows the ANI operation, and the operation of Y75 with the bold line is indicated as

16 ANI Y75

At this time, the Y75 operation is executed with X9, and the X9 $\overline{Y75}$ operation is executed.

This operation takes the inversion of the specified number (contact a becomes contact b), and then series connection operation is executed.



$Y80 = X9 \cdot \overline{Y75}$

Fig. 4-3. ANI operation

4-3-2. OR, ORI ... Parallel connection operation

OR has the operation function for parallel connection of contact a, and ORI has the operation function for parallel connection of contact b.

Both execute the parallel connection operation (OR) with the previous operation result.

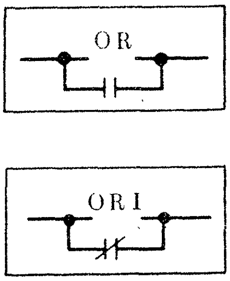
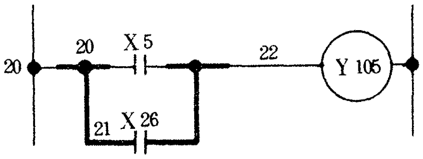


Fig. 4-4. Symbols for OR and ORI

In the figure, the connection of the bold line of X26 becomes OR, and it is indicated as

21 OR X26

With this operation, X5 is the previous operation result, and the operation X5 + X26 is executed.



$Y105 = X5 + X26$

Fig. 4-5. OR operation

The connection for T7 in the figure becomes ORI, and it is indicated as

25 ORI T7

With this operation, the inversion of the specified number is taken and parallel connection operation is executed.

4-3-3. LD, LDI ... Operation start

LD has the operation function to start operation of contact a, and LDI has the function to start operation of contact b. At the time to start operation for the circuit block, there is no previous operation result, so that the input signal is taken in by this instruction, and it becomes the operation result. Accordingly, this instruction is always used first for a circuit block.

In the figure, the bold line of X13 becomes this instruction, and it is indicated by

30 LD X13

In this circuit, as can be seen from the logic expression, Y71 + Y88 is enclosed in brackets, and to calculate the equation, the contents of the brackets must be processed. Their operation start is executed newly from Y71 with operation of the bracket contents, so that an LD instruction is also used for Y71 as

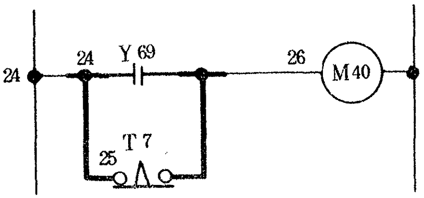
31 LD Y71

Since the LDI instruction effects operation start from contact b, the contact C7 of the figure becomes this command, and it is indicated as

45 LDI C7

4-3-4. ANB ... Series connection operation between blocks

ANB has the function to operate a series connection between blocks. In the case of the circuit shown in the figure, AND must be executed between the two operation blocks $(X1 \cdot \overline{X2} + Y30) = A$ and $(\overline{Y31} + \overline{Y32}) = B$. The instruction to execute this operation A × B becomes ANB, and it corresponds to the bold-line connection parts in the figure. This means that the ANB symbol is not a contact symbol, but a connection symbol.



$M40 = Y69 + \overline{T7}$

Fig. 4-6. ORI operation

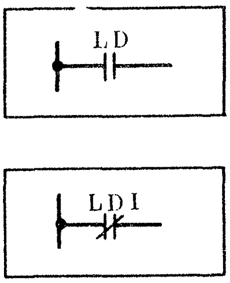
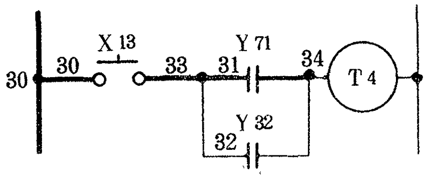
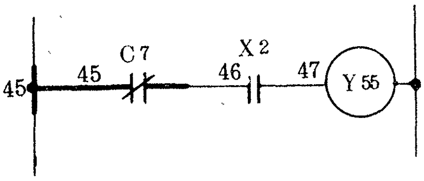


Fig. 4-7. Symbols for LD and LDI



$X13\,(Y71 + Y88)$

Fig. 4-8. LD operation



$Y55 = \overline{C7} \bullet X2$

Fig. 4-9. LDI operation

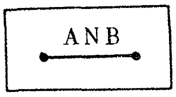
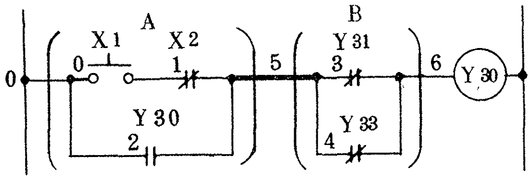


Fig. 4-9. Symbol for ANB



$Y30 = (X1 \bullet \overline{X2} + Y30) \bullet (\overline{Y31} + \overline{Y33})$

Fig. 4-11. ANB operation

4-3-5. ORB ... Paralle connection operation between blocks

ORB has the function to operate a parallel connection between blocks.

In the figure, OR must be executed between the two operation blocks (X17 • Y90) = A and (M33 • Y74) = B. ORB becomes the instruction to execute this operation A + B, and it corresponds to the bold line in the figure. In this figure, as X31 is only one block, the instruction for X31 becomes OR and not ORB.

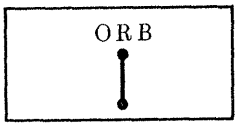
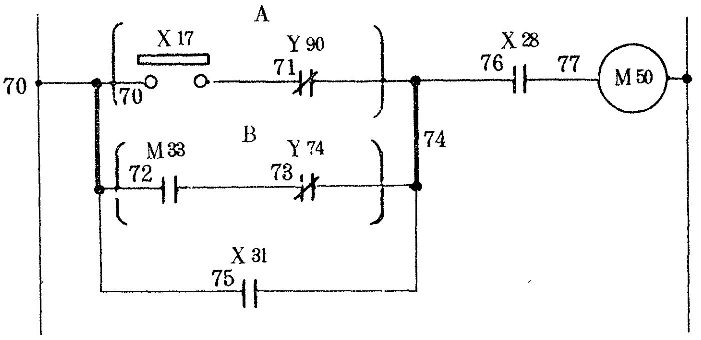


Fig. 4-12. Symbol for ORB



$$M50 = (X17 \cdot \overline{Y90} + M33 \cdot \overline{Y74} + X31)\, X28$$

Fig. 4-13. ORB operation

4-3-6. MC, MCR ... Master control start, reset

As shown in the figure, master control permits establish an efficient circuit switching sequence program by opening or closing a common line of the control circuit.

MC: Master control start
MCR: Master control reset

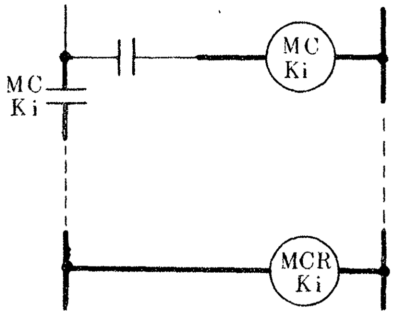


Fig. 4-14. MC, MCR instructions

MC and MCR are handled by affixing the Ki index number. MCR Ki is always required in regard to MC Ki.

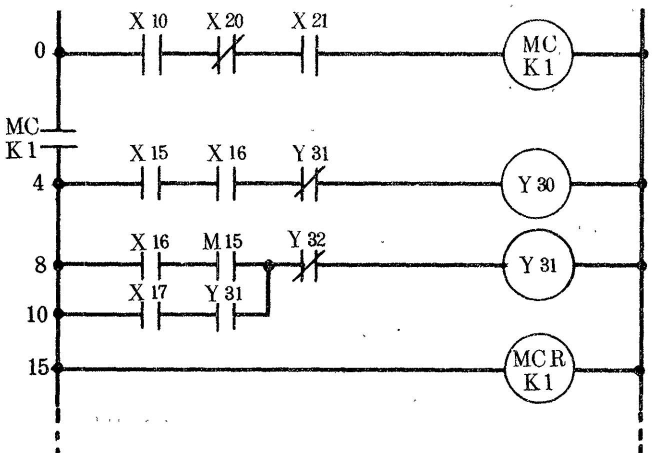


Fig. 4-15. Examples of use for MC and MCR

(Coding)

0	LD	X10
1	ANI	X20
2	AND	X21
3	MC	K 1
4	LD	X15
5	AND	X16
6	OUT	Y30
7	ANI	Y31
8	LD	X16
9	AND	M15
10	LD	X17
11	AND	Y31
12	ORB	
13	ANI	Y32
14	OUT	Y31
15	MCR	K 1

4-3-7. SET, RST ... Flip-flop set, reset

Flip-flop set and reset are effective for the following 3 types of devices:

Output	Y
Temporary memory	M
Failure memory	F

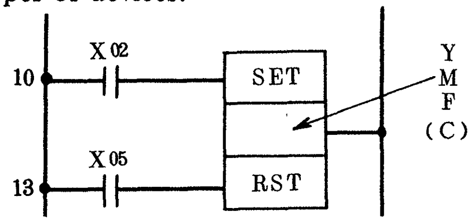


Fig. 4-16. SET, RST instructions

(However, reset is available for counters C.)

(Coding)

(1) Output (Y) set, reset

```
10   LD    X02
11   SET   Y10
12   LD    X05
13   RST   Y10
```

(2) Temporary memory (M) set, reset

```
10   LD    X02
11   SET   M00
12   LD    X05
13   RST   M00
```

(3) Failure memory (F) set, reset

```
10   LD    X02
11   SET   F00
12   LD    X05
13   RST   F00
```

For M, Y, and F, self-holding circuits by normal sequence circuits are also possible, but the functions may be understood more easily in the above way. Please compare with the normal method.

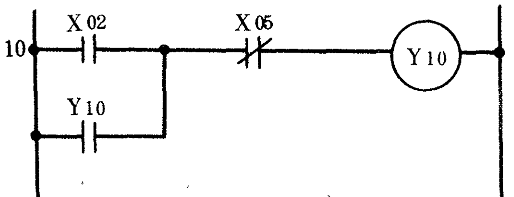


Fig. 4-17. Self-holding circuit

(Coding)

```
10   LD    X02
11   OR    Y10
12   ANI   X05
13   OUT   Y10
```

When flip-flop holding is required at the time of power failure, output Y latching becomes possible for the decided range by connecting a latch unit to the input/output unit connector and selecting 16/32/64 points by switching. [Refer to 5-2-9.(1).]

K2CPU is provided with a switch enabling to hold M collectively in the case of power failure. [Refer to 5-2-9.(2).]

(3) Counter (C) preset value resetting is also possible by RST instructions.

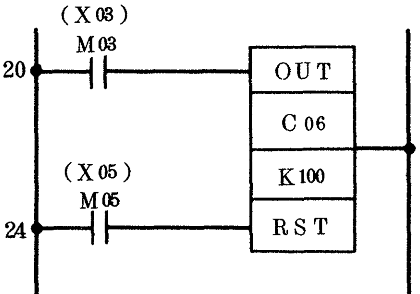


Fig. 4-18. Counter circuit

Coding of Fig. 4-18.

```
20      LD      M03
21      OUT     C06
22      K       100
23      LD      M05
24      RST     C06
```

4-3-8. SFT ... Temporary memory shift

By application of an SFT instruction to a temporary memory (M), a 1 bit shift register can be composed.

By linked application of SFT instructions to a temporary memory (M), a shift register for the number of links can be established.

When a shift signal M02 is given to a temporary memory Mj, this memory has the function for processing as follows under consideration of the status of the preceding Mj-1.

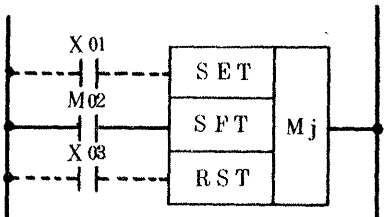


Fig. 4-20. SFT instructions

$$\left.\begin{array}{l} Mj = 1 \text{ for } Mj - 1 = 1 \\ Mj = 0 \text{ for } Mj - 1 = 0 \end{array}\right\} \text{Mj-1 becomes 0 after SFT instruction execution.}$$

When this instruction is used, the number of required steps (M link number) must use consecutive M.

Aslo, as the shift signal M02 may be continuous if seen with the programmable controller inside the seanning time, reverse arrangement as follows should be executed in regard to the sequence arrangement. Normally, the shift signal will use the pulse form M changed from input X.

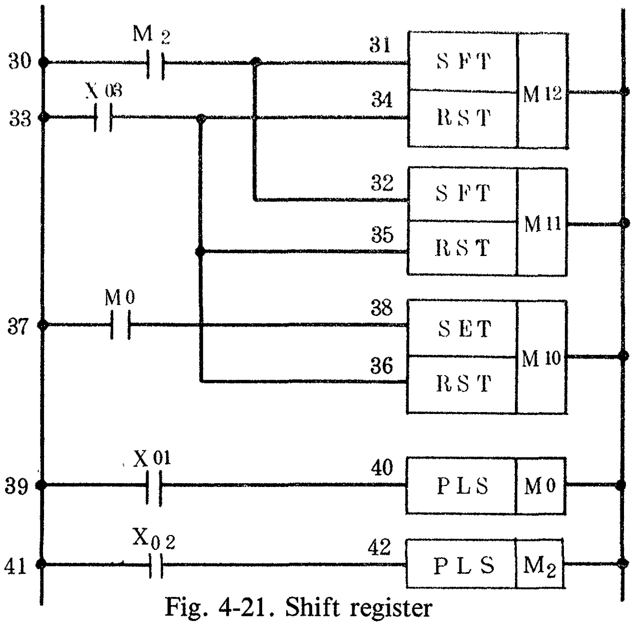


Fig. 4-21. Shift register

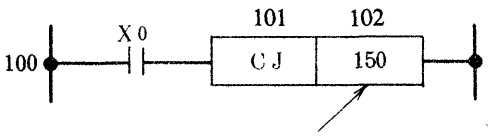
	(Coding)	
30	LD	M 2
31	SFT	M 12
32	SFT	M 11
33	LD	X 03
34	RST	M 12
35	RST	M 11
36	RST	M 10
37	LD	M 0
38	SET	M 10
39	LD	X 01
40	PLS	M 0
41	LD	X_{02}
42	PLS	M 2

Notes:

1. Do not give SFT instructions to M0. M255 may be shifted to M0. In the same way, this prohibition also applies to M254.
2. For M12 = 1, it does not become 0 even when M2 (shift pulse) is added. Resetting by X03 is required.

4-3-9. CJ ... Conditional jump

When the condition X0 ON is established, a jump is executed to the sequence step address 150, and processing of the following step addresses is executed. (Coding CJ K150)



Step address for the jump destination

Fig. 4-22. CJ command

A jump to the upper step numbers (smaller numbers) is not possible. Take care that the jump destination always has a larger step number than the number from which the jump starts.

There are for example the following use methods:

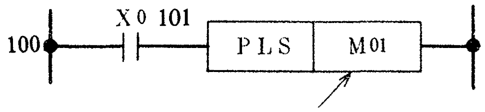
① Jump over temporarily unrequired circuits.

② Separation of processing circuits in conditions of highspeed processing.

Please pay attention, as the step number control is required at the time of programming. Especially when program debugging, instruction inserting and deleting, care must be taken because of the change of jump destination address.

4-3-10. PLS ... Pulse formation

By X0 ON, a pulse signal of 1 round division of the program is formed for M01. The temporary memory M is the object device for pulse signal formation.



Limited to M.

Fig. 4-23. PLS instruction

This pulse formation is executed for internal program processing. Accordingly, leadout for use as an external pulse signal is not possible.

4-3-11. OUT ... Output

OUT is the instruction for output of the operation result up to that time. The object output devices are of Y, M, F, T, and C.

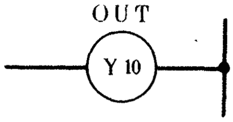


Fig. 4-24. OUT instruction

At the time of Y, M, F, and T, this corresponds to coil drive, and at the time of C, it becomes the counting input in regard to the counter.

Without influence onto the operation result, an OUT instruction permits output to several objects as shown in the following, and consecutive execution of the next operation is also possible.

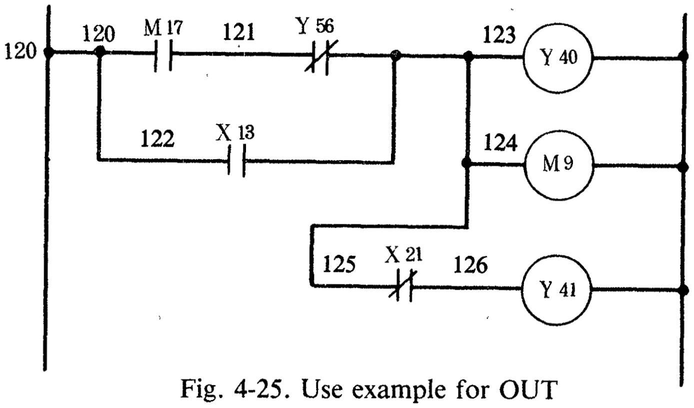


Fig. 4-25. Use example for OUT

(Coding)

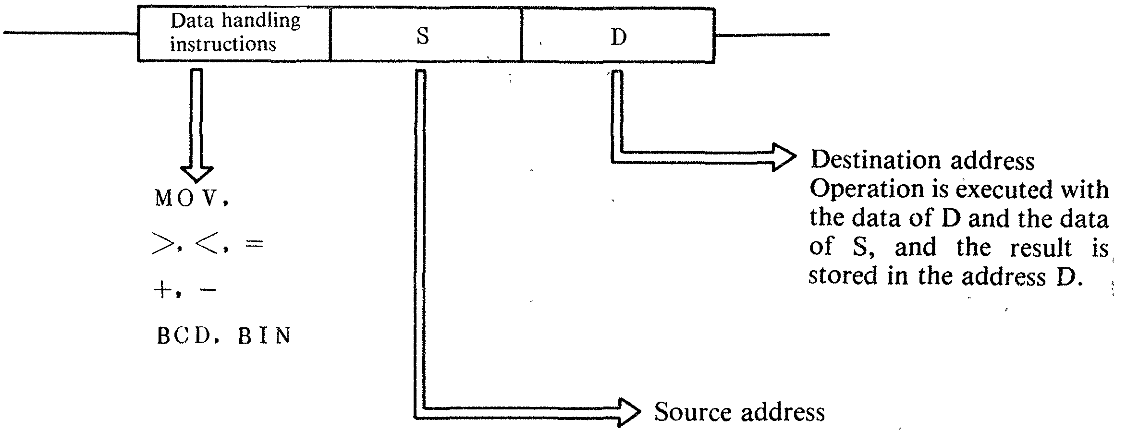
120	LD	M 17
121	ANI	Y 56
122	OR	X 13
123	OUT	Y 40
124	OUT	M 9
125	ANI	X 21
126	OUT	Y 41

In the case of OUT C, counting is executed during the rise time of the counting input, so that change of the counting input to pulses by a PLS instruction is not required.

4-4. Data handling instructions

In addition to sequence instructions by relay symbols and logic symbols, data handling instructions such as addition and subtraction, magnitude comparison, BCD/BIN conversion, etc. are included.

Data handling instructions are composed of 3 steps, and their notation method is shown in the following figure.



Operation is executed with the data of D and the data of S, and the result is stored in the address D.

This indicates the source of the data for the operation.

Fig. 4-26. Composition of the data handling instructions

4-4-1. MOV ... Data transmission

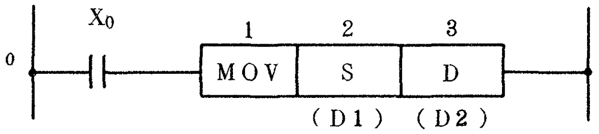


Fig. 4-27. MOV instruction

```
0   LD    X0
1   MOV
2         D1
3         D2
```

When X0 becomes ON, the data of S (D1) are transmitted to D (D2).

The possible operation S/D combinations are shown in the following table by circles.

S \ D	K	D	T	C	X	Y	M
Constant → K		○					
D		○	○	○		○	○
T		○					
G		○					
X		○					
Y							
M		○					

According to the table on the left, these data operations are possible.

```
MOV    K        D
MOV    Di       Dj
MOV    T or C   D
MOV    X        D
MOV    D           T or C
MOV    D        Y
MOV    D        M
MOV    M        D
```

(1) Storage in D0 of the constant K (1 2 3 4)

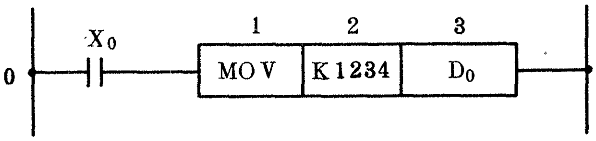


Fig. 4-28. Constant set by MOV

(Coding)

```
0   L D     X0
1   M O V
2   K       1 2 3 4
3           D0
```

For storage of 1 2 3 4 (4 digit decimal number) in D0 (16-bit register), storage is executed after automatic conversion to binary numbers. Accordingly, it should be remembered that the data are handled as binary numbers (BIN) in the programmable controller. BCD/BIN conversion is executed in principle only with input and output of data via the input/output unit. → Refer to the items in regard to BIN, BCD conversion.

(2) Take-in of the input signal from X_{10} ~ $_{1F}$ into D_{10}.

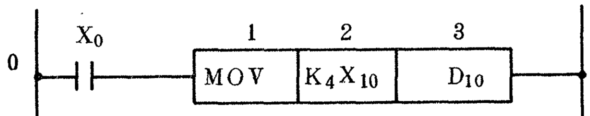


Fig. 4-29. Data input by MOV

(Coding)

```
0   L D     X0
1   M O V
2   K 4     X10
3           D10
```

By specifying "K4X10", the input of 4 bits (assumption of BCD x 1 digit), i.e. "X" input (X10 to 1F) as 4 x 4 = 16 bits, is handled.

K indicates the number (1 ~ 4) of digits (1 digit of 4 bits).

Accordingly, as the data of D10 are in the status of X10 to 1F, this kind of handling is executed only for pure binary numbers.

In the case of BCD, execute "BIN" and store in D10.

(3) Output of the contents of D11 into Y50 to 5B.

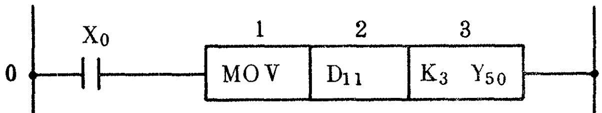


Fig. 4-30. Data output by MOV

(Coding)

```
0   L D     X0
1   M O V
2           D11
3   K 3     Y50
```

The contents of D11 are put out into Y50 to 5B (3 x 4 = 12 points).

This is used for output of binary numbers as they are. For output as BCD, execute "BCD" and output to Y.

K indicates the number (1 ~ 4) of digits.

4-4-2. > ... Larger (data comparison)

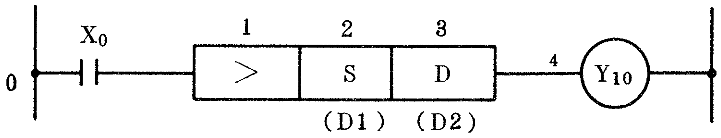


Fig. 4-31. > instruction

(Coding)

```
0   LD    X0
1   >
2         D1
3         D3
4   OUT   Y10
```

S > D is executed when X0 becomes ON, and Y10 becomes ON for S > D, while Y10 becomes OFF for S ≦ D.

The possible S/D operation combinations are shown in the following table by circles. These are common for >, <, and =.

S\D	K	D	T	C	X	Y	M
K		○					
D		○					
T							
C							
X							
Y							
M							

```
>   K    D
>   Di   Dj
```

The magnitude relationship is as follows:

98 99 100 101 102

> 100 ←| = |→ 100 <

Notice that >, < do not include =.

4-4-3. < ... Smaller (data comparison)

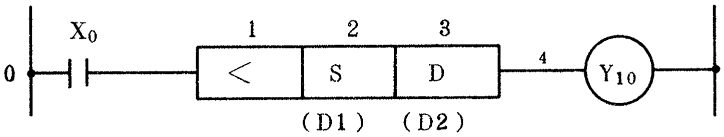


Fig. 4-32.⟨instruction

The same as for > (larger)

(Coding)

```
0   LD    X0
1   <
2         D1
3         D2
4   OUT   Y10
```

4-4-4. = ... Coincidence (data coincidence)

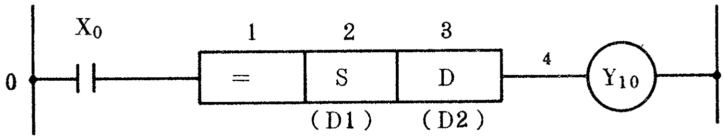


Fig. 4-33. = instruction

The same as for > (larger)

```
0   LD    X0
1   =
2         D1
3         D2
4   OUT   Y10
```

Note: For these magnitude conparisons, all data are to be handled with the data converted to binary numbers. Even when constant input from the PU is executed as decimal numbers, they become binary numbers on the inside.
> = < instructions are series contact processing.
These cannot be executed in the same way as LD and OR.

4-4-5. + ... Addition

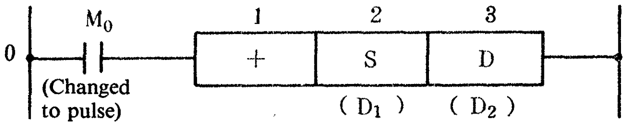


Fig. 4-34. + instruction

(Coding)

```
0    L D    M0
1    +
2           D1
3           D2
```

(The result is stored in D2.)

D + S is executed by Mo ON, and the value is stored in D.

The possible operation combination are shown in the following table by circles.

S\D	K	D	T	C	X	Y	M
K		○					
D		○					
T							
C							
X							
Y							
M							

+ Kn Di → Kn + Dj ⇒ Dj
+ Di Dj → Dj + Di ⇒ Dj

If input remains ON, addition or subtraction is executed at each scanning. Accordingly, input is made after changing to pulse form.

4-4-6. - ... Subtraction

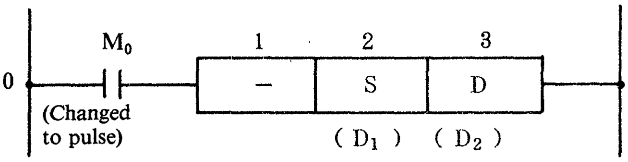


Fig. 4-35. - instruction

(Coding)

```
0    L D    M0
1    –
2           D1
3           D2
```

(The result of D2 — Dl is stored in D2.)

D - S (D2 — Dl) is executed by M0 ON, and the value is stored in D (i.e. D2).

D becomes the minuend and S becomes the subtrahend, so that caution is required, as the positions are reversed from the normal calculation equations.

Example: Count-up number to the target position: 100

Deceleration start point: 15 counts in advance
100 — 15 → 85
Assuming D0 = 100
MOV D0 D1 → D1 = 100 is reached once,
— K15, D1 → The result becomes D1 = 85.
In this way, both D0 and D1 can be used next.

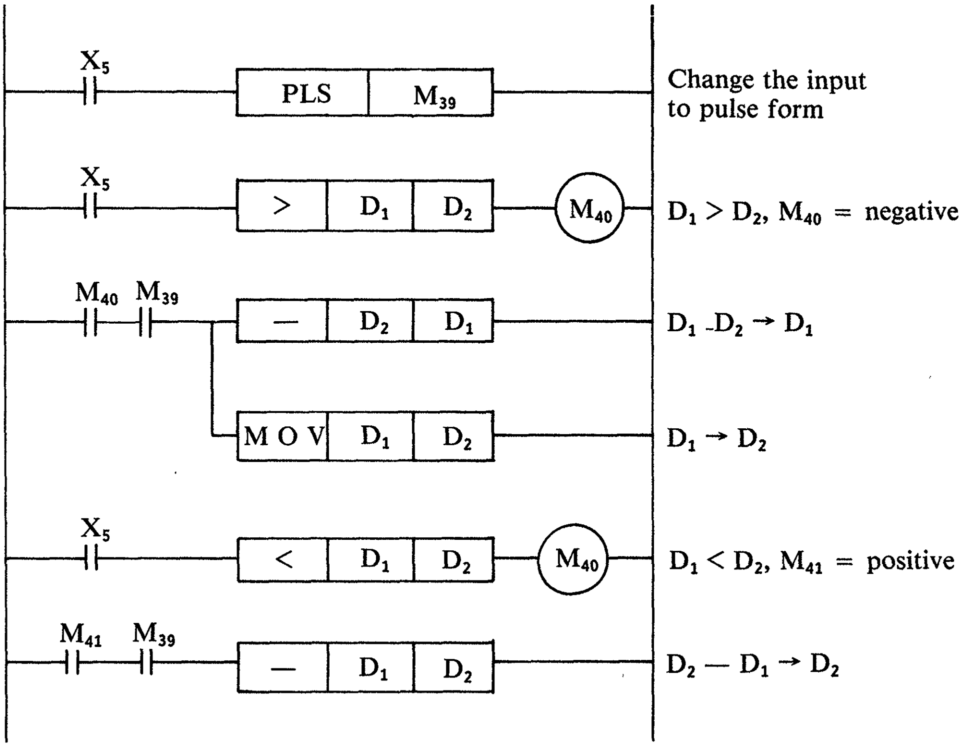
$$- \quad \utilde{K} \quad D\,i \quad \rightarrow \quad D\,i - \utilde{K} \Rightarrow D\,i$$

$$- \quad \utilde{D\,i} \quad D\,j \quad \rightarrow \quad D\,j - \utilde{D\,i} \Rightarrow D\,j$$

In the same way as for addition, execute subtraction instructions for subtraction data after conversion to binary numbers.

As only positive integral numbers are handled for subtraction, take care that the operation result does not become negative.

If the subtruction result becomes negative, execute the comparison > or < between S and D. When S > D, execute the S — D operation in the same way, and M stores the negative result in memory.



Operating $D_2 - D_1$ M_{41}

The possible operation combinations are the same as for "+".

4-4-7. BCD ... Conversion from BIN to BCD

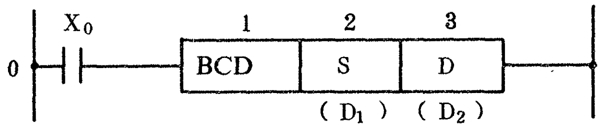


Fig. 4-36. BCD instruction

(Coding)

0	L D	X_0
1	B C D	
2		D_1
3		D_2

When X0 becomes ON, BCD conversion is executed for the contents of S (binary number), and then storage is executed in D. The possible conversion combinations are shown in the following table by circles.

S\D	K	D	T	C	X	Y	M
K							
D		○					
T		○					
C		○					
X							
Y							
M							

According to the table on the left

	BCD conversion
B C D , D i , D j→	D i ⇨ D j
B C D , T , D k→	T ⇨ D k
B C D , C , D k→	C ⇨ D k

The data register contents are principally binary numbers, but after BCD conversicn, the registers become Dj, Dk, and the contents are BCD.

The main application for BCD conversion is output to the outside as a decimal number from the status in the register (binary number) via the output unit.

However, output is executed via the temporary storage D register.

(1) The contents of C10 are output into Y50 to 5F as a 4-digit BCD value.

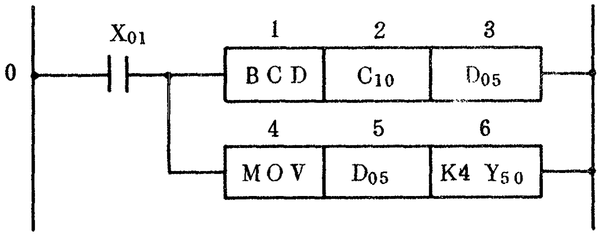


Fig. 4-37. Application example for BCD instructions

(Coding)

0	L D	X_{01}
1	B C D	
2		C_{10}
3		D_{05}
4	M O V	
5		D_{05}
6	K_4	Y_{50}

⇩

Output
Y_{50} ~ $_{53}$ First BCD digit
Y_{54} ~ $_{57}$ Second BCD digit
Y_{58} ~ $_{5B}$ Third BCD digit
Y_{5C} ~ $_{5F}$ Fourth BCD digit

4-4-8. BIN ... Conversion from BCD to BIN

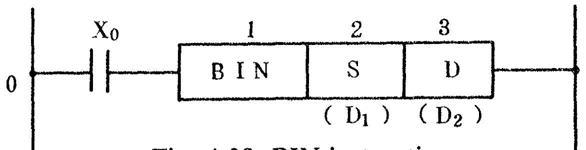


Fig. 4-38. BIN instruction

(Coding)

```
0   LD     X0
1   BIN
2          D1
3          D2
```

When X0 becomes ON, BIN (binary) conversion is executed for the contents (BCD) of S, and the result is stored in D.

The possible conversion combinations are shown in the following table.

S \ D	K	D	T	C	X	Y	M
K							
D		○					
T							
C							
X		○					
Y							
M							

B I N , D i , D j→D i ⇒ D j

B I N , X , D k→X ⇒ D k

Generally most input data are decimal numbers, so this BIN instruction exists. The input data are taken in as BCD, and internal processing is executed after conversion to binary numbers.

(1) The data are taken in from input X60 to 6B (a BCD number of 3 digits), the numerical value is set for the present value of counter C1, and this value reduced by 20 is set for the present value of C2.

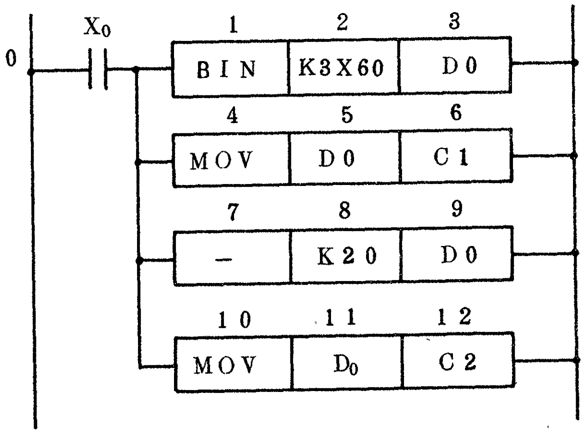


Fig. 4-39. Application for BIN instructions

(Coding)

```
 0   LD     X0
 1   BIN
 2   K3     X60
 3          D0
 4   MOV
 5          D0
 6          C1  → Setting of the present
                  count value for C1
 7   –
 8   K      20
 9          D0
10   MOV
11          D0
12          C2  → Setting of the present
                  count value for C2
```

4-5. Program control instructions

Program control instructions are instructions to advance the individual program of the programmable controller.

4-5-1. NOP ... No processing

NOP is the instruction for no processing, and it has no influence on the operation result up to that time.

NOP is used to provide an intermediate space for memory revision, or as shown in the figure,
(a) When the contact Y97 is short-circuited by 6 AND Y97 → NOP.
(b) The contact X8 is separated with 11 ORI X8 → 11 NOP.

The programming unit (PU) is provided with a function for consecutive NOP writing, insertion, and cancellation, so that the operation is easy.
Please refer to the instruction manual for the PU.

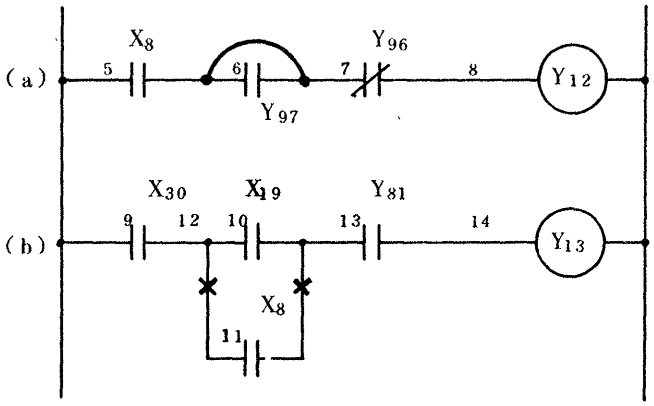


Fig. 4-40. Contact cancellation by NOP instructions

4-5-2. END ... Program completion

END is entered at the end of the required program steps to announce program completion. When the CPU detects the END instruction, the program counter is returned to 0, and scanning is started again from step number 0.

In the figure, the program is completed at 311, and return to step number 0 is executed.

The END instruction must be written at the end of any program.

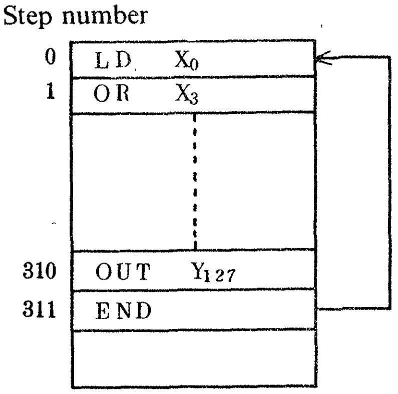
As scanning returns to the start with the END instruction, the scanning time depends on the position of the END instruction.

The scanning time (response time) can be calculated as follows:

Average time × step number (from 0 to END instruction)

⇩

K1CPU	30μs
K2CPU	10μs



Program completion

Fig. 4-41. Use of END instructions

END instructions may also be used temporarily at the time of program debugging or testing for program execution to an intermediate point.

5. Programming

Please read this chapter together with the instruction manual for the programming unit (PU).

5-1. Programming principles

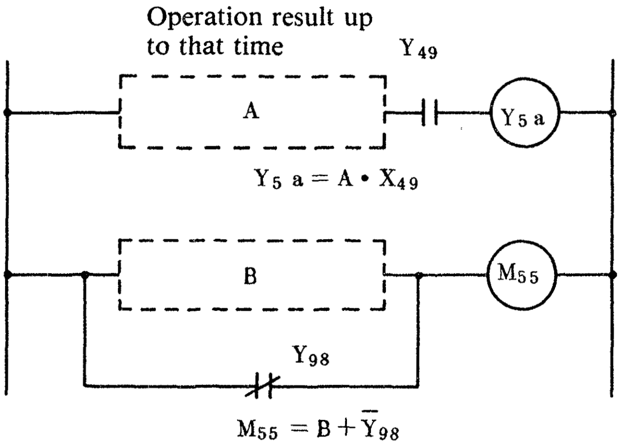
The program is drawn up by programming on the basis of the circuit diagrams, but the following principles exist.
Some of these principles have already been explained, but they will be listed again in this chapter.

5-1-1. Instructions

(1) Operation instructions like AND, OR, LD, ANB, etc. are connection instructions specifying the connection of contacts or contact combinations (operation results up to that time).
(2) Even when the output instructions OUT, SET, and RST are executed, the operation results up to that time will not be changed.
(3) The END instruction must be given at the program end.

5-1-2. Operation

(1) The operation instructions AND, ANI, OR, and ORI execute serial or parallel connection of the contacts of the input/output numbers specified by these instructions in regard to the operation results up to that time.



AND X49
Serial connection is executed between the operation result A at this time and the contact a of X49.

ORI Y98
Parallel connection is executed between the operation result B at this time and the contact b of Y98.

Fig. 5-1. Operation parties

(2) Up to 8 consecutive blocks can be produced starting with LD or LDI, the connection between these blocks is executed by ANB or ORB, and it will become 7 consecutive blocks.

At this time, the operations between the blocks proceed in reverse order from the new operation results (8) to the old operation results (1) as 8 to 7 ... (8 to 2) and 1 . Such program operations are not mistakes, but in order to facilitate programming and checking, it is recommended to always connect two blocks starting with LD by ANB or ORB.

```
①  LD   X6

②  LDI  Y53
⋮
⑧  LD   X13
```

$\boxed{1}$ AND for ANB 8 and 7
$\boxed{2}$ OR for ORB 8 x 7 or 6
⋮
$\boxed{6}$ AND for ANB(8 to 3) and 2
$\boxed{7}$ OR for ORB (8 to 2) and 1
OUT Y_{69}

Operations between blocks

5-1-3. Program

(1) The program is drawn up for each of the contact symbols and coil symbols.
The step number of the finished program is at least equal to the number of symbols or larger than this number.

(2) With the horizontally written relay symbol diagram, the program sequence for each block is from left to right and from the top to the bottom.
Return from the right to the left is possible with block operations.

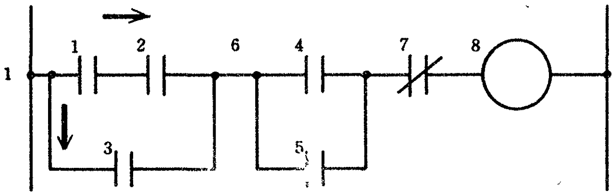


Fig. 5-2. Program sequence

(3) The program is always executed in the sequence of the program numbers, and no numbers are jumped.

(4) The END instruction must be given at the end of the program.
By this, the operation time for one program round can be reduced.
Please note that the CPU does not operate normally if the END instruction is omitted.

5-1-4. Contacts

(1) The contacts X, Y, M, T, and C can be used as often as desired, and there is no limitation on the contact number.

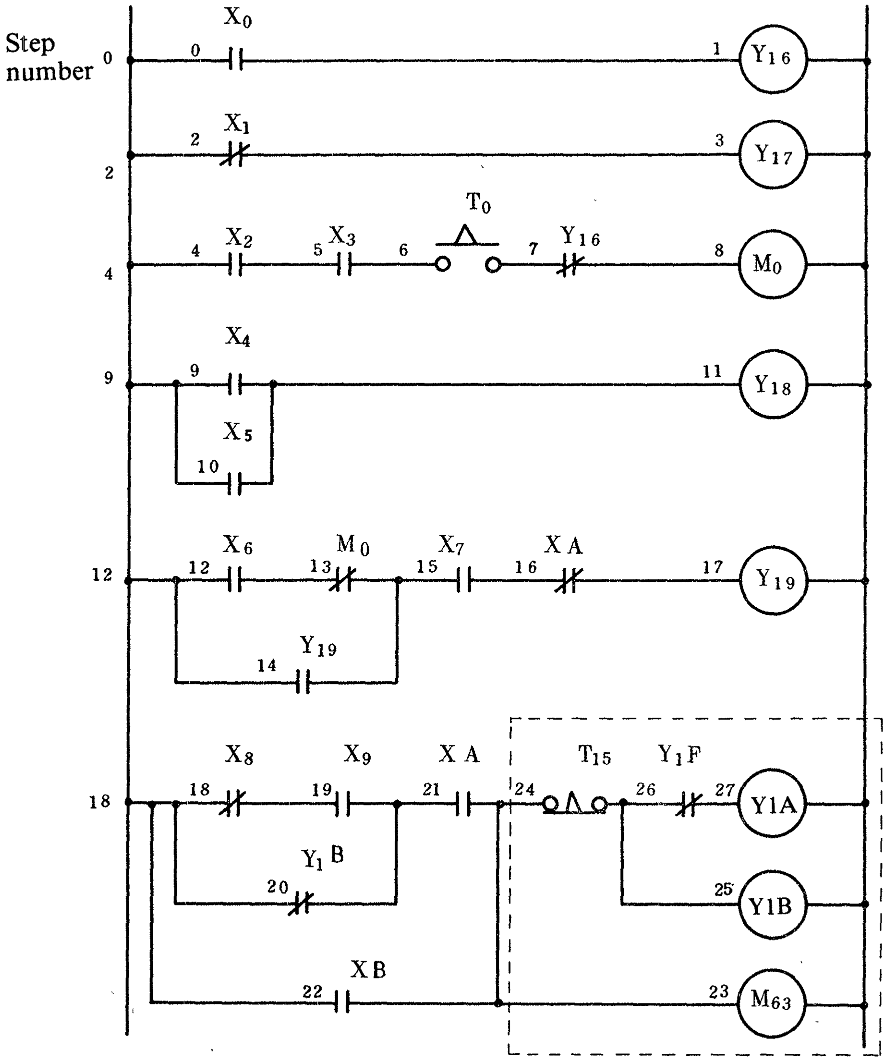
(2) Contact a input to contact b or contact b input to contact a can be produced internally.

(3) There is no concept for the contact capacity.

(4) In the electrical sense, there is no snake circuit.

5-2. Sequence instructions and program examples

5-2-1. Simple circuits with AND and OR

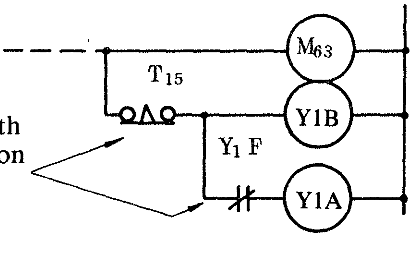


Step No.			Instruction			Device No.			
		0	L	D		X			0
		1	O	U	T	Y		1	6
		2	L	D	I	X			1
		3	O	U	T	Y		1	7
		4	L	D		X			2
		5	A	N	I	X			3
		6	A	N	D	T			0
		7	A	N	I	Y		1	6
		8	O	U	T	M			0
		9	L	D		X			4
	1	0	O	R		X			5
		1	O	U	T	Y		1	8
		2	L	D		X			6
		3	A	N	I	M			0
		4	O	R		Y		1	9
		5	A	N	D	X			7
		6	A	N	I	X			A
		7	O	U	T	Y		1	9
		8	L	D	I	X			8
		9	A	N	D	X			9
	2	0	O	R	I	Y		1	B
		1	A	N	D	X			A
		2	O	R		X			B
		3	O	U	T	M		6	3
		4	A	N	I	T		1	5
		5	O	U	T	Y		1	B
		6	A	N	I	Y		1	F
		7	O	U	T	Y		1	A

Note: If the output instruction has additional conditions, it is drawn up later.
Accordingly, the sequence should be drawn up as shown in the figure on the right.

Fig. 5-3. Circuit by AND and OR

Circuit establishment with sequential addition of conditional circuits.



5-2-2. Complicated circuit by ANB and ORB

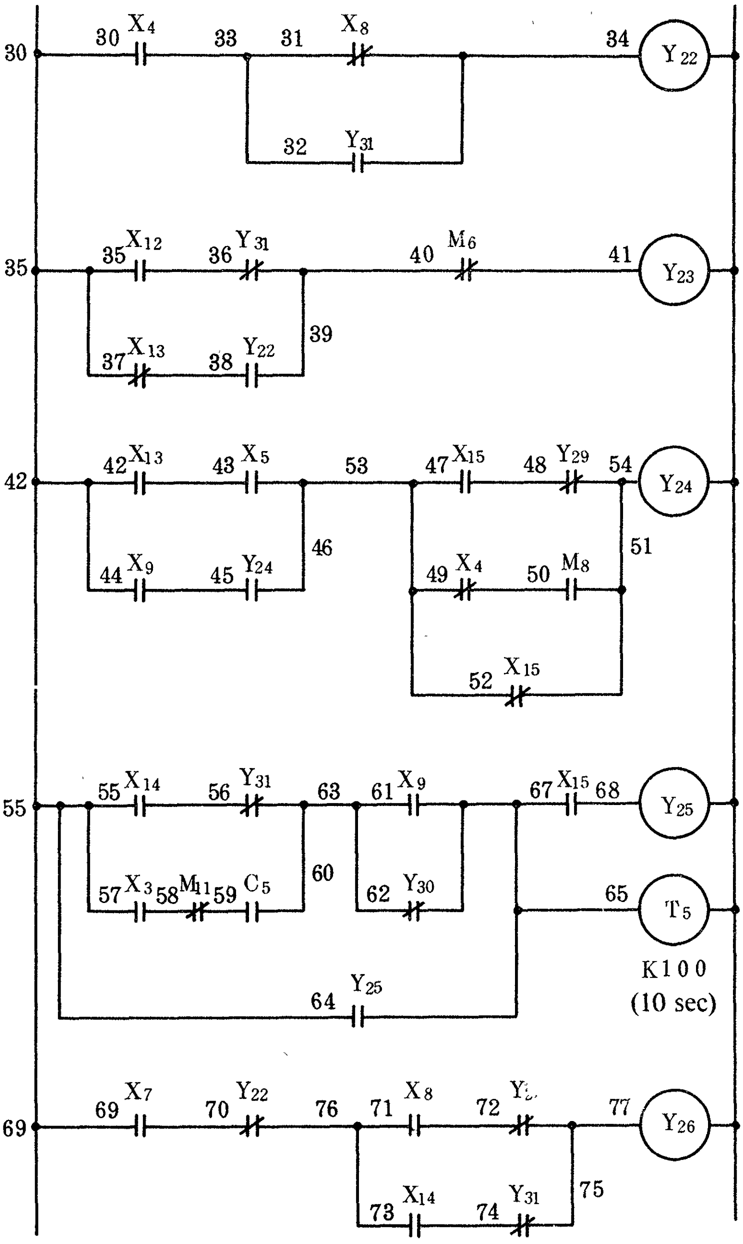


Fig. 5-4. Circuit by ANB and ORB

Step No.			Instruction			Device No.			
	3	0	L	D		X			4
		1	L	D	I	X			8
		2	O	R		Y		3	1
		3	A	N	B				
		4	O	U	T	Y		2	2
		5	L	D		X		1	2
		6	A	N	I	Y		3	1
		7	L	D	I	X		1	3
		8	A	N	D	Y		2	2
		9	O	R	B				
	4	0	A	N	I			M	6
		1	O	U	T	Y		2	3
		2	L	D		X		1	3
		3	A	N	D	X			5
		4	L	D		X			9
		5	A	N	D	Y		2	4
		6	O	R	B				
		7	L	D		X		1	5
		8	A	N	I	Y		2	9
		9	L	D	I	X			4
	5	0	A	N	D	M			8
		1	O	R	B				
		2	O	R	I	X		1	5
		3	A	N	B				
		4	O	U	T	Y		2	4
		5	L	D		X		1	4
		6	A	N	I	Y		3	1
		7	L	D		X			3
		8	A	N	I	M		1	1
		9	A	N	D	C			5
	6	0	O	R	B				
		1	L	D		X			9
		2	O	R	I	Y		3	0
		3	A	N	B				
		4	O	R		Y		2	5
		5	O	U	T	T			5
		6				K	1	0	0
		7	A	N	D	X		1	5
		8	O	U	T	Y		2	5
		9	L	D		X			7
	7	0	A	N	1	Y		2	2
		1	L	D		X			8
		2	A	N	I	Y		2	7
		3	L	D		X		1	4
		4	A	N	I	Y		3	1
		5	O	R	B				
		6	A	N	B				
		7	O	U	T	Y		2	6

5-2-3. Circuits with common control lines

(1) Circuit separation with contacts

Circuits with common control lines can not be programmed as they are. As shown in the following figure, the contacts separating the individual lines (X0 and X1 in the figure) are entered for each block.

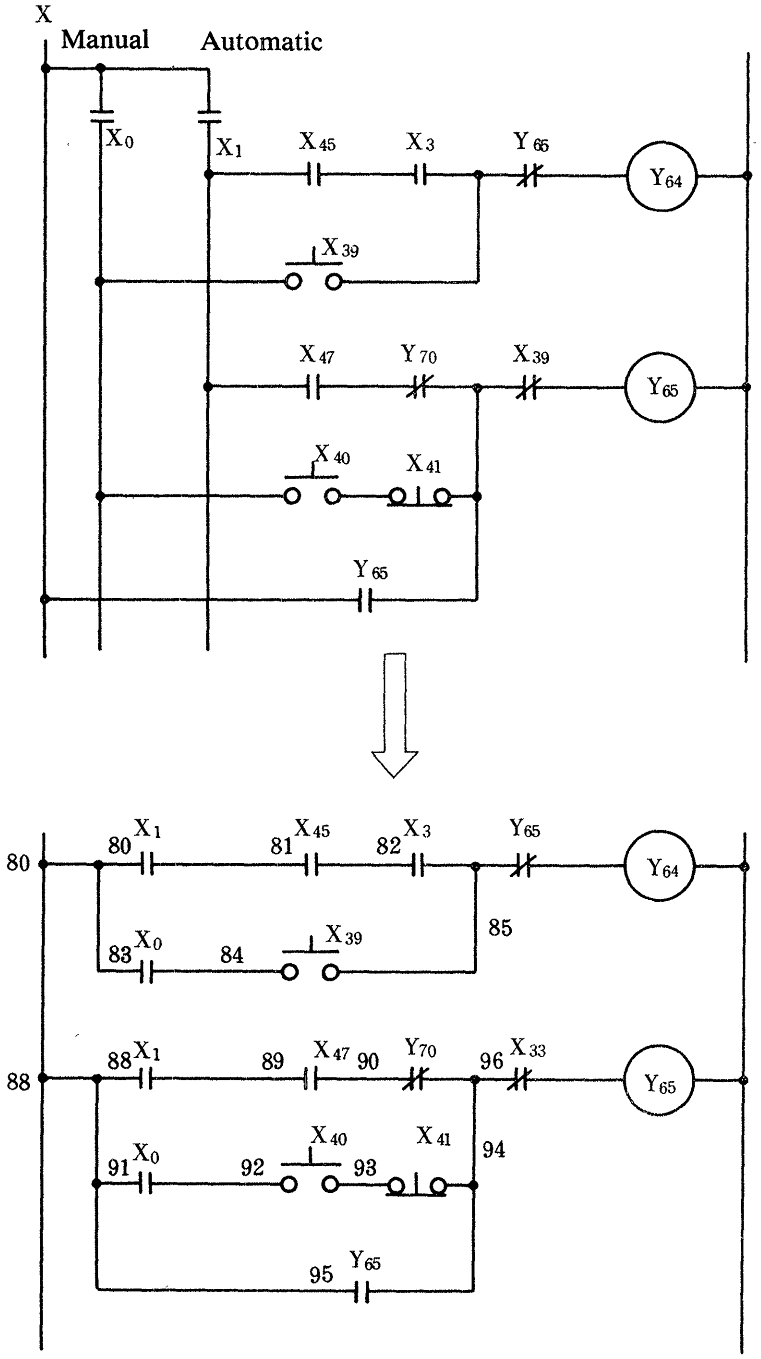
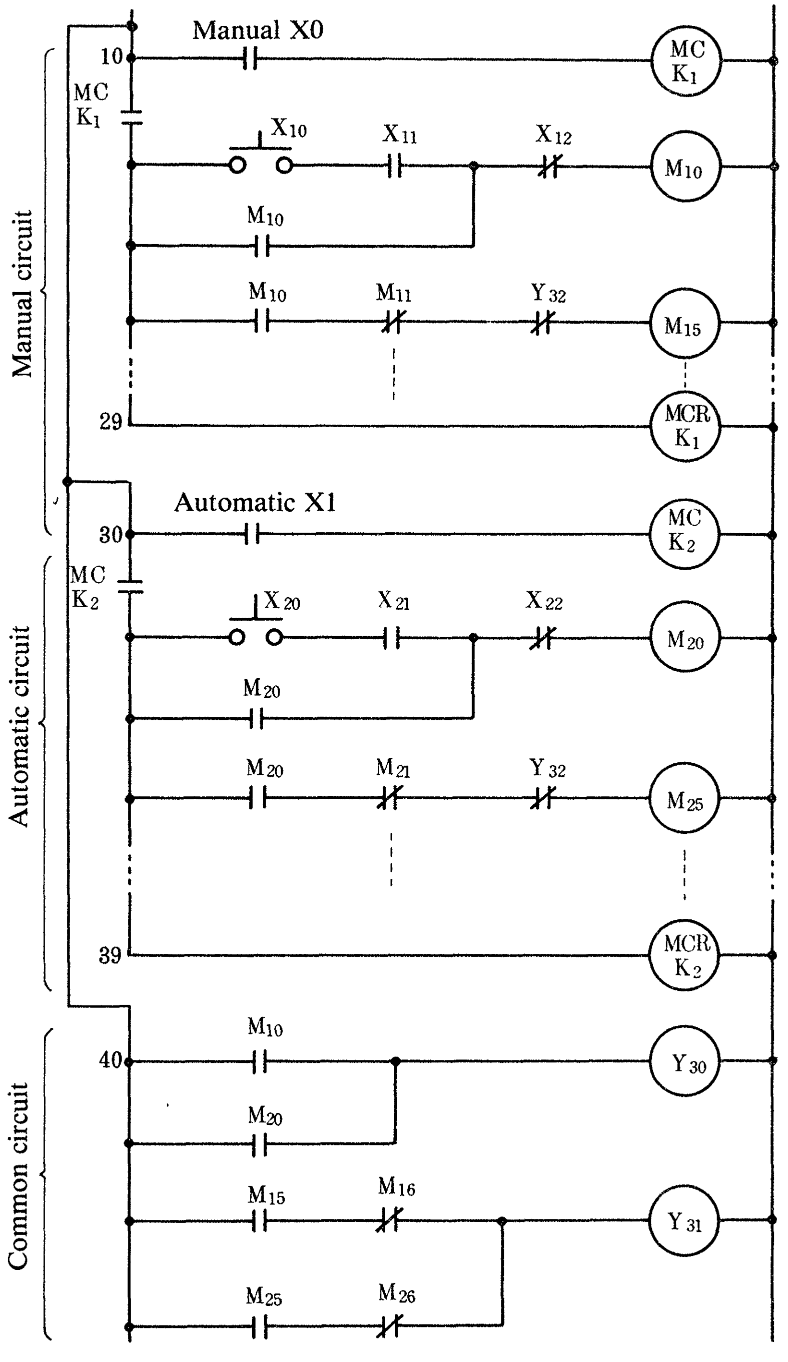


Fig. 5-5. Circuit with common control lines (1)

Step No.			Instruction			Device No.			
	8	0	L	D		X			1
		1	A	N	D	X		4	5
		2	A	N	D	X			3
		3	L	D		X			0
		4	A	N	D	X		3	9
		5	O	R	B				
		6	A	N	I	Y		6	5
		7	O	U	T	Y		6	4
		8	L	D		X			1
		9	A	N	D	X		4	7
	9	0	A	N	I	Y		7	0
		1	L	D		X			0
		2	A	N	D	X		4	0
		3	A	N	I	X		4	1
		4	O	R	B				
		5	O	R		Y		6	5
		6	A	N	I	X		3	3
		7	O	U	T	Y		6	5

(2) Use of master control MC and MCR



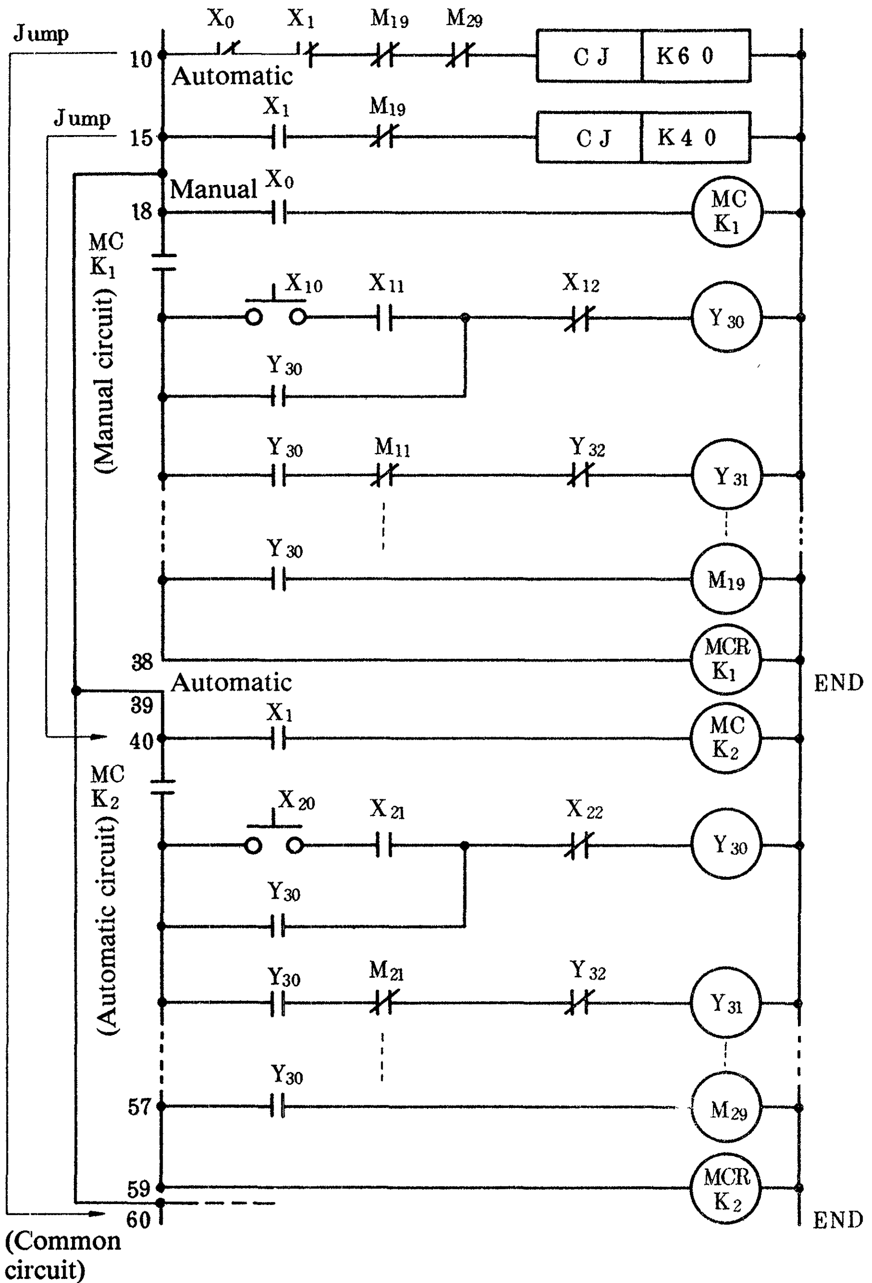
Step No.	Instruction	Device No.
10	LD	X 0
11	MC	K 1
12	LD	X 10
13	AND	X 11
14	OR	M 10
15	ANI	X 12
16	OUT	M 10
17	LD	M 10
18	ANI	M 11
19	ANI	Y 32
20	OUT	M 15
↓	↓	↓
29	MCR	K 1
30	LD	X 1
31	MC	K 2
32	LD	X 20
33	AND	X 21
34	OR	M 20
↓	↓	↓
39	MCR	K 2
40	LD	M 10
41	OR	M 20
42	OUT	Y 30
43	LD	M 15
44	ANI	M 16
45	LD	M 25
46	ANI	M 26
47	ORB	
48	OUT	Y 31

Fig. 5-6. Circuit with common control lines (2)

Note: Do not execute OUT for the same device in each range of MCKi and MCKj. In this case, use the temporary memory in each range as shown in the above figure, and combine at the common circuit. Chatterring will be caused when this is not observed.

(3) Use of conditional jump CJ

With the method of the above item (2), the CPU scans the entire circuit, so that a considerable scanning time is required. The use of "CJ" is convenient when the scanning time is to be minimized.



Step No.	Instruction	Device No.
10	LDI	X 0
11	ANI	X 1
12	ANI	M 19
13	ANI	M 29
14	CJ	K 60
15		K 60
16	LD	X 11
17	ANI	M 19
18	CJ	
19		K 40
20	LD	X 0
⋮	⋮	⋮
37	OUT	M 19
38	MCR	K 1
39	END	
40	LD	X 1
41	MC	K 2
42	LD	X 20
⋮	⋮	⋮
57	LD	Y 30
58	OUT	M 29
59	MCR	K 2
60	END	

Fig. 5-7. Common control line circuit controlled by CJ instructions

Note: Switching circuit
$X_0 \cdot \overline{X_1}$ → Manual circuit (END)
$\overline{X_0} \cdot X_1$ → Automatic circuit (END)
$\overline{X_0} \cdot \overline{X_1}$ Common circuit (No switching for automatic and manual) →

Switching to separate circuits is to be executed after the completion of each circuit, i.e. corresponding to M19 and M29 in this case.

5-2-4. Timer circuit

(1) ON delay circuit

①Continuous input

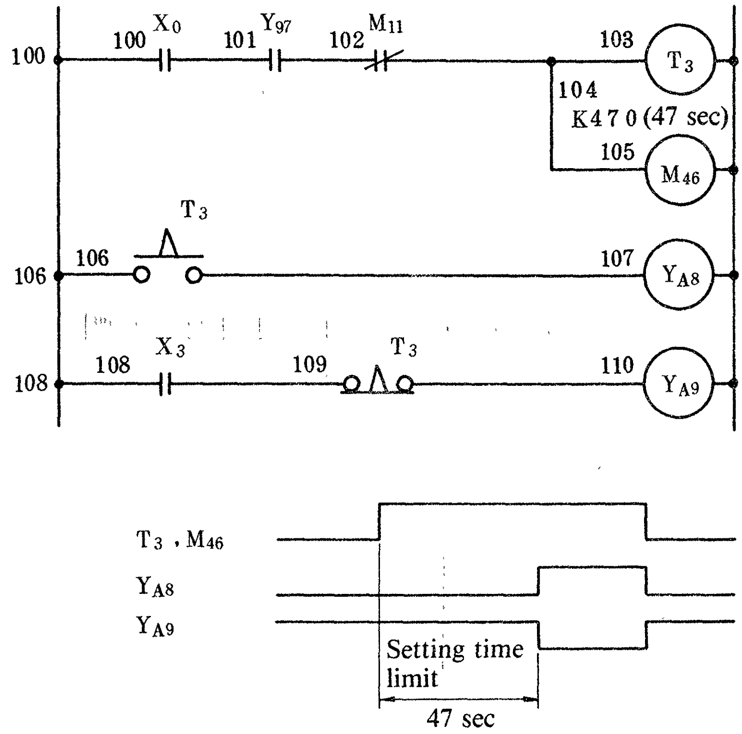


Fig. 5-8. ON delay timer circuit (1)

Step No.			Instruction			Device No.			
1	0	0	L	D		X			0
		1	A	N	D	Y		9	7
		2	A	N	I	M		1	1
		3	O	U	T	T			3
		4				K	4	7	0
		5	O	U	T	M		4	6
		6	L	D		T			3
		7	O	U	T	Y		A	8
		8	L	D		X			3
		9	A	N	I	T			3
	1	0	O	U	T	Y		A	9

M46 corresponds to the instantaneous contact of T3.

②Instantaneous input

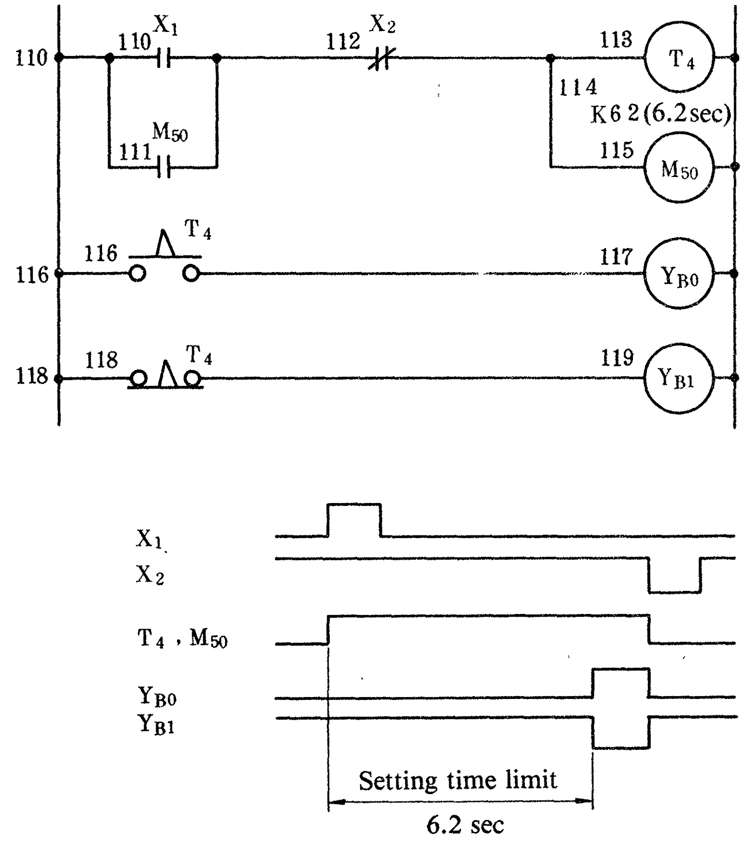


Fig. 5-9. ON delay timer circuit (2)

Step No.			Instruction			Device No.			
1	1	0	L	D		X			1
		1	O	R		M		5	0
		2	A	N	I	X			2
		3	O	U	T	T			4
		4				K		6	2
		5	O	U	T	M		5	0
		6	L	D		T			4
		7	O	U	T	Y		B	0
		8	L	D	I	T			4
		9	O	U	T	Y		B	1

This is an ON delay circuit by the instantaneous input X1 and X2.

The temporary memory M50 is required for self-holding.

(2) OFF delay circuit

(1) Continuous input

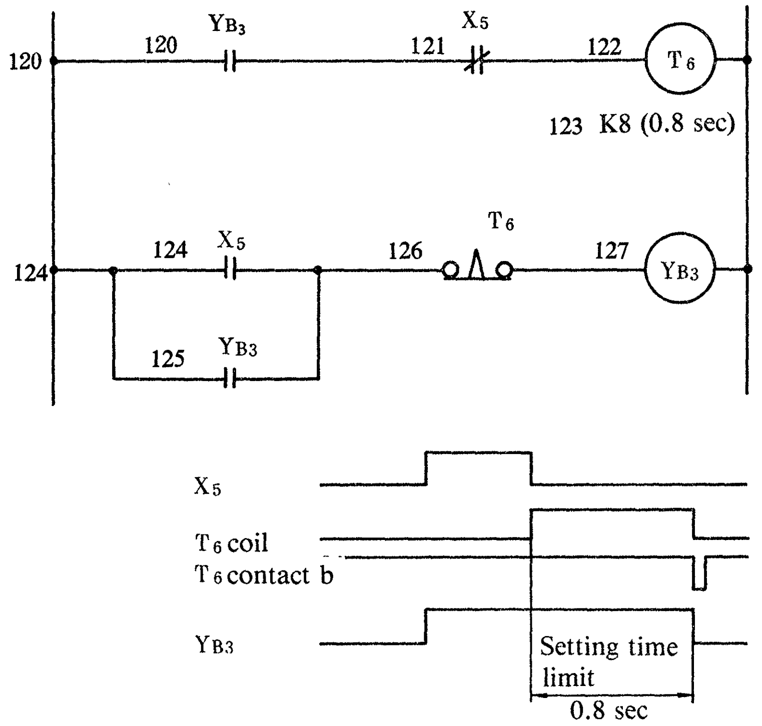


Fig. 5-10. OFF delay timer circuit (1)

Step No.			Instruction			Device No.			
1	2	0	L	D		Y		B	3
		1	A	N	I	X			5
		2	O	U	T	T			6
		3				K			8
		4	L	D		X			5
		5	O	R		Y		B	3
		6	A	N	I	T			6
		7	O	U	T	Y		B	3

(2) Instantaneous input

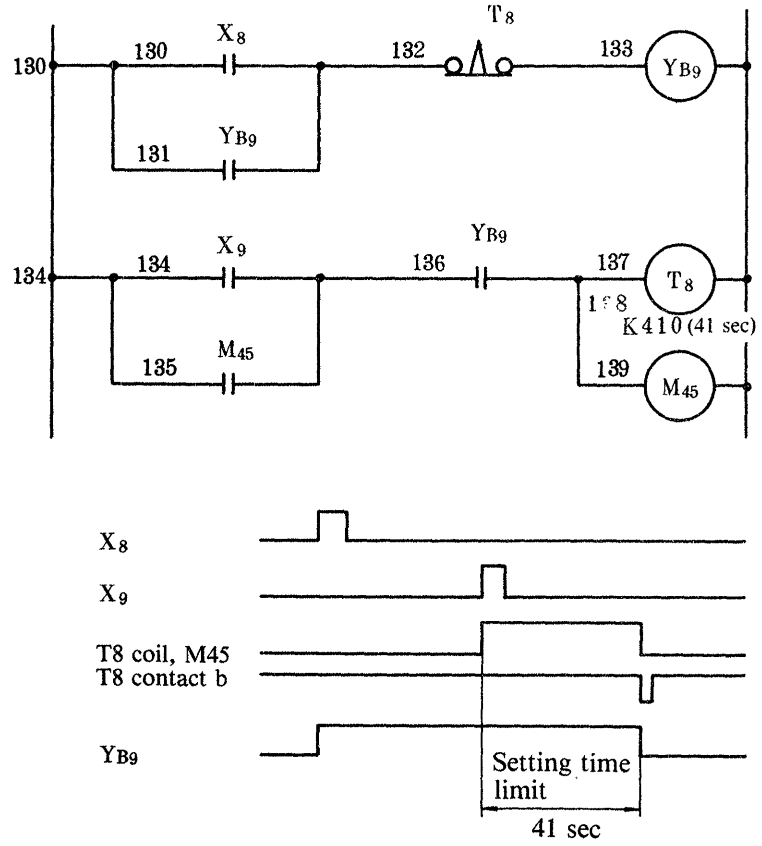


Fig. 5-11. OFF delay timer circuit (2)

Step No.			Instruction			Device No.			
1	3	0	L	D		X			8
		1	O	R		Y		B	9
		2	A	N	I	T			8
		3	O	U	T	Y		B	9
		4	L	D		X			9
		5	O	R		M		4	5
		6	A	N	D	Y		B	9
		7	O	U	T	T			8
		8				K	4	1	0
		9	O	U	T	Y		B	9

This is an OFF delay circuit by the instantaneous input X8 and X9.

M45 corresponds to the instantaneous contact of T8.

(3) Long-time timer

①Long-time timer by series use of timers

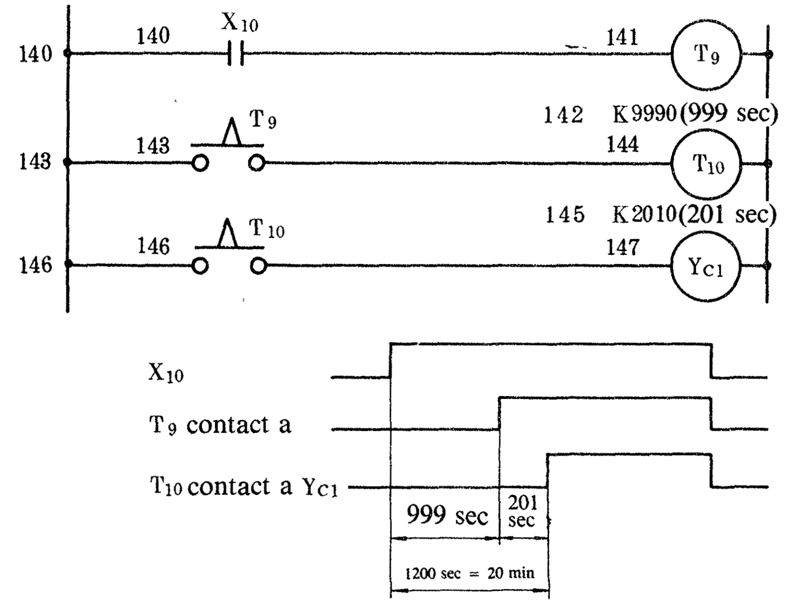


Fig. 5-12. Long-time timer

Step No.			Instruction			Device No.				
1	4	0	L	D		X			1	0
		1	O	U	T	T				9
		2				K	9	9	9	0
		3	L	D		T				9
		4	O	U	T	T			1	0
		5				K	2	0	1	0
		6	L	D		T			1	0
		7	O	U	T	Y			C	1

A long time limit is obtained by series use of timers so that the required time limit is obtained.

②Long-time timer with use of timer and counter

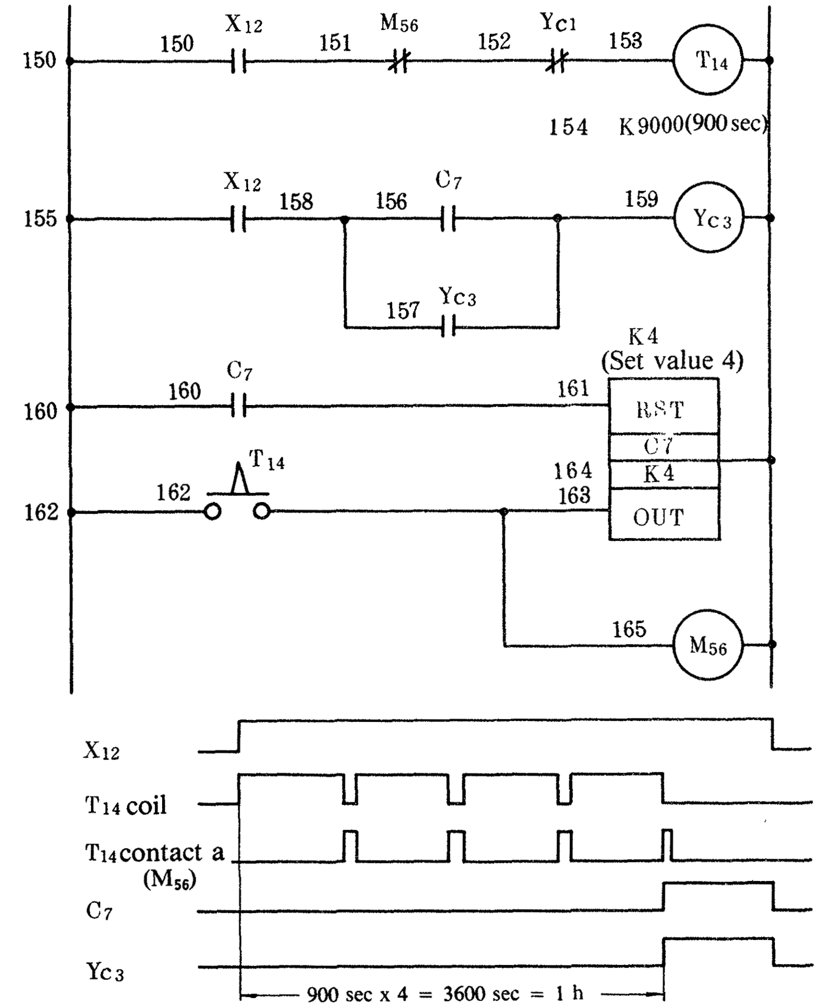


Fig. 5-13. Long-time timer (2)

Step No.			Instruction			Device No.				
1	5	0	L	D		X			1	2
		1	A	N	I	M			5	6
		2	A	N	I	Y			C	1
		3	O	U	T	T			1	4
		4				K	9	0	0	0
		5	L	D		X			1	2
		6	L	D		C				7
		7	O	R		Y			C	3
		8	A	N	B					
		9	O	U	T	Y			C	3
1	6	0	L	D		C				7
		1	R	S	T	C				7
		2	L	D		T			1	4
		3	O	U	T	C				7
		4				K				4
		5	O	U	T	M			5	6

The number of time-ups of the timer T14 is counted by the counter C7 to obtain a long time.

After time-up, M56 resets T14.

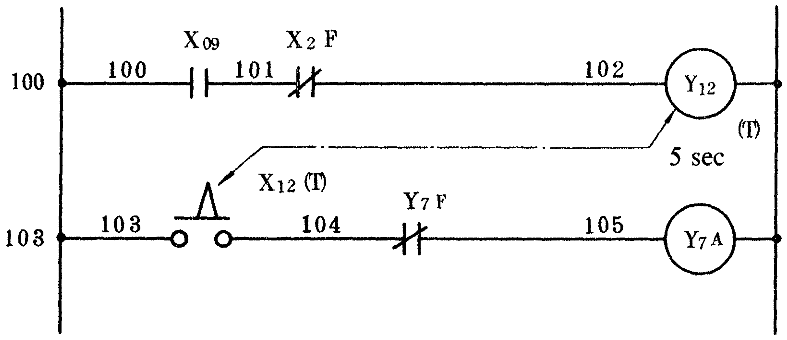
At the time of count-up, C7 executes self-holding for the output YC3, YC3 resets T14, and then the time limit operation is stopped.

(4) Circuits by analog timer

The analog timer unit (KT61) has 16 points/unit, the timer time limits are from 0.3 to 3 sec and 3 to 30 sec, and time limit adjustment is executed with the front volumes.

The timer numbers are decided by the base unit installation position.

Assuming installation at the I/O unit No. 1:



Step No.			Instruction			Device No.			
1	0	0	L	D		X		0	9
		1	A	N	I	X		2	F
		2	O	U	T	Y	1	2	(T)
		3	L	D		X	1	2	(T)
		4	A	N	J	Y		7	F
		5	O	U	T	Y		7	A

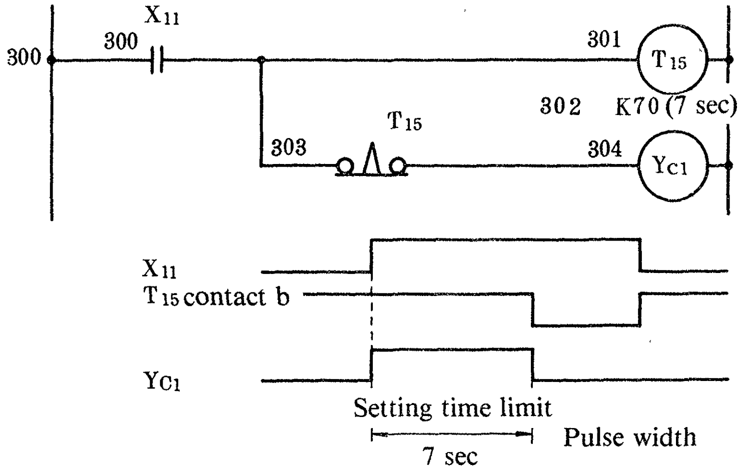
Fig. 5-14. Analog timer unit circuit

The analog timer Y12 corresponds to a coil, and X12 becomes its ON delay contact.

With regard to the program memory, no identification is made for timers, but entry as (T) for coils and contacts to identify as a timer should be made on drawings and coding.

5-2-5. One-shot circuits

(1) One-shot circuit for ON time



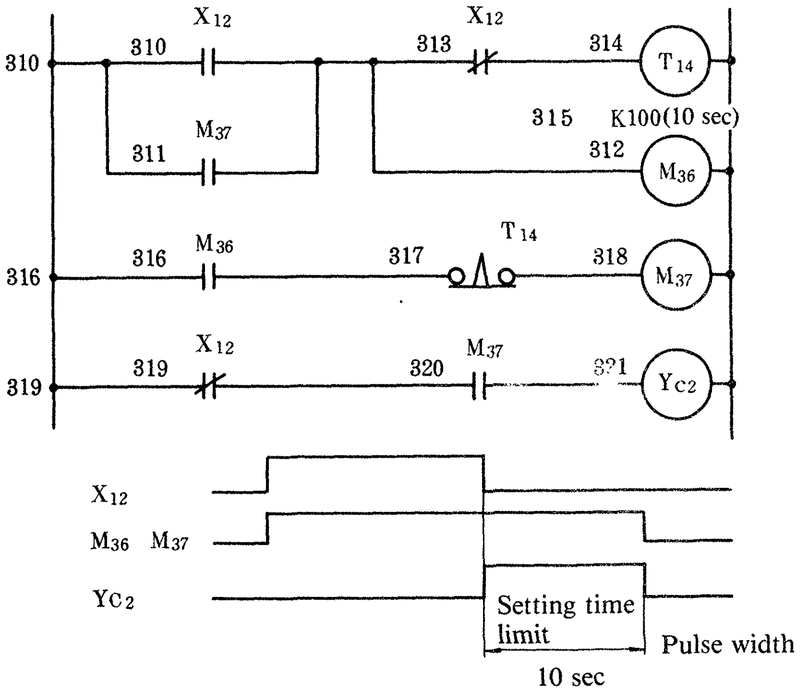
Step No.			Instruction			Device No.			
3	0	0	L	D		X		1	1
		1	O	U	T	T		1	5
		2				K		7	0
		3	A	N	I	T		1	5
		4	O	U	T	Y		C	1

After input ON, output drive is executed for a fixed time.

The input ON time must be longer than the setting time limit.

Fig. 5-15. One-shot circuit for ON time

(2) One-shot circuit for OFF time



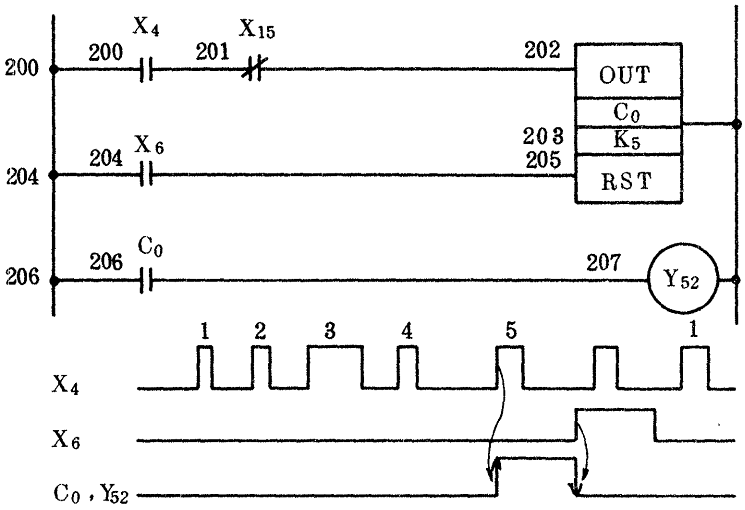
Step No.			Instruction			Device No.			
3	1	0	L	D		X		1	2
		1	O	R		M		3	7
		2	O	U	T	M		3	6
		3	A	N	I	X		1	2
		4	O	U	T	T		1	4
		5				K	1	0	0
		6	L	D		M		3	6
		7	A	N	I	T		1	4
		8	O	U	T	M		3	7
		9	L	D	I	X		1	2
3	2	0	A	N	D	M		3	7
	2	1	O	U	T	Y		C	2

After input OFF, output drive is executed for a fixed time.

Fig. 5-16. One-shot circuit for OFF time

5-2-6. Counter circuit

(1) Count, reset



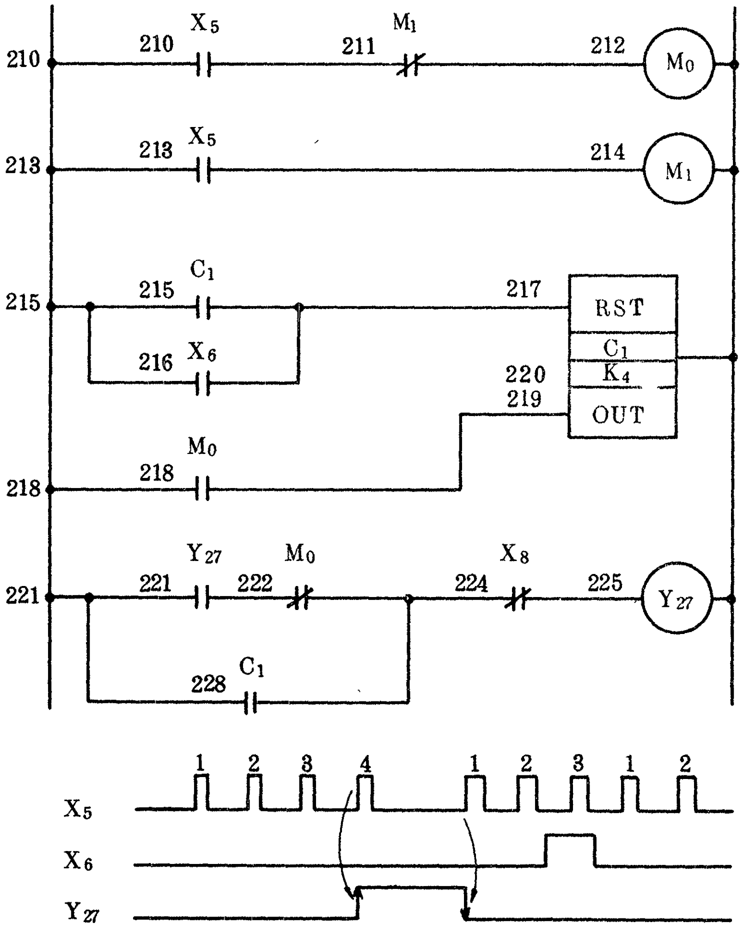
Step No.			Instruction			Device No.			
2	0	0	L	D		X			4
		1	A	N	I	X		1	5
		2	O	U	T	C			0
		3				K			5
		4	L	D		X			6
		5	R	S	T	C			0
		6	L	D		C			0
		7	O	U	T	Y		5	2

X_4 Count input

X_6 Reset input

Fig. 5-17. Counter citcuit (1)

(2) Preset counter



Step No.			Instruction			Device No.			
2	1	0	L	D		X			5
		1	A	N	I	M			1
		2	O	U	T	M			0
		3	L	D		X			5
		4	O	U	T	M			1
		5	L	D		C			1
		6	O	R		X			6
		7	R	S	T	C			1
		8	L	D		M			0
		9	O	U	T	C			1
2	2	0				K			4
		1	L	D		Y		2	7
		2	A	N	I	M			0
		3	O	R		C			1
		4	A	N	I	X			8
		5	O	U	T	Y		2	7

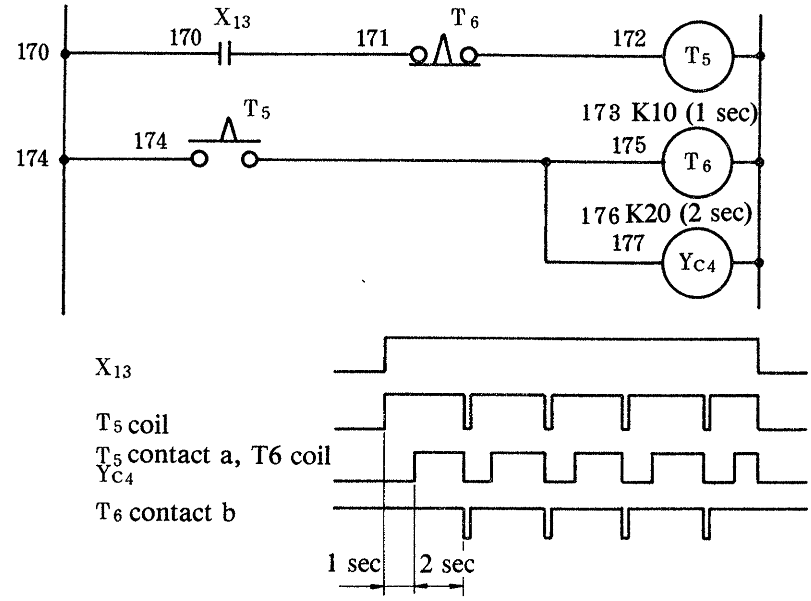
X_5 Count input

X_6 Reset input

Y_{27} Count-up output

Fig. 5-18. Counter circuit (2)

5-2-7. Flicker circuit



Step No.			Instruction			Device No.			
1	7	0	L	D		X		1	3
		1	A	N	I	T			6
		2	O	U	T	T			5
		3				K		1	0
		4	L	D		T			5
		5	O	U	T	T			6
		6				K		2	0
		7	O	U	T	Y		C	4

Fig. 5-19. Flicker circuit

5-2-8. Pulse circuit

(1) Rise detection pulse

Step No.			Instruction			Device No.			
3	3	0	L	D		X		1	3
		1	P	L	S	M			0

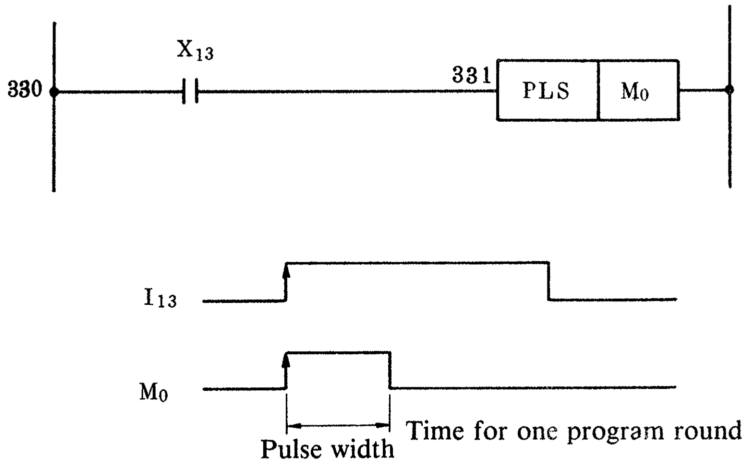


Fig. 5-20. Rise detection circuit

Use this as internal program trigger pulse.

(2) Drop detection pulse

Step time			Instruction			Device No.			
3	3	0	L	D	I	X		1	3
		1	P	L	S	M			0

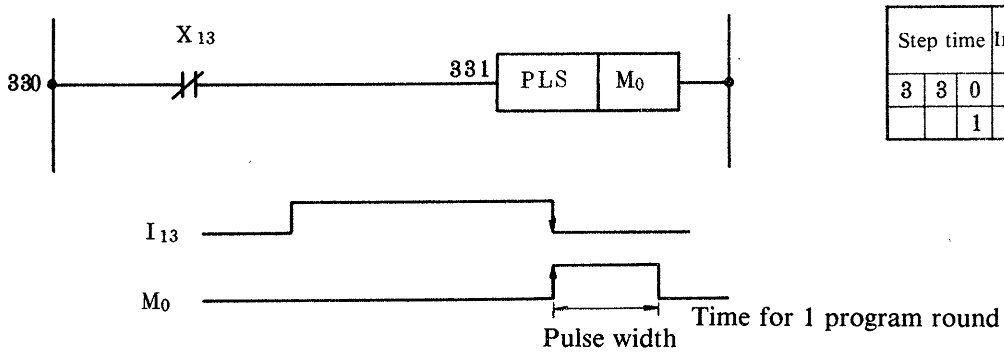
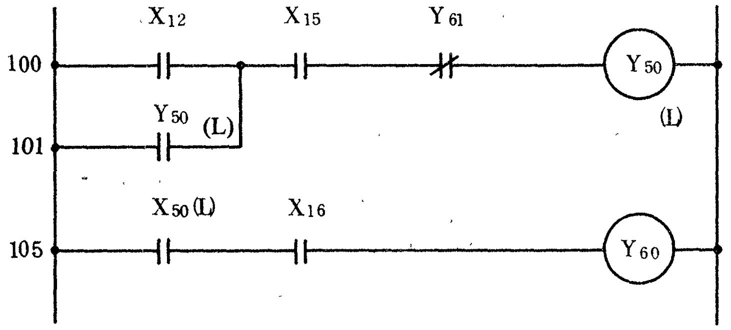


Fig. 5-21. Drop detection circuit

5-2-9. Holding circuit for power failure with temporary memory

(1) Use of the latch unit KL61

Installation of KL61 to the I/O unit No. 5 as a 16 point latch taking power failure holding output from output Y60 connected to the I/O unit No. 6.

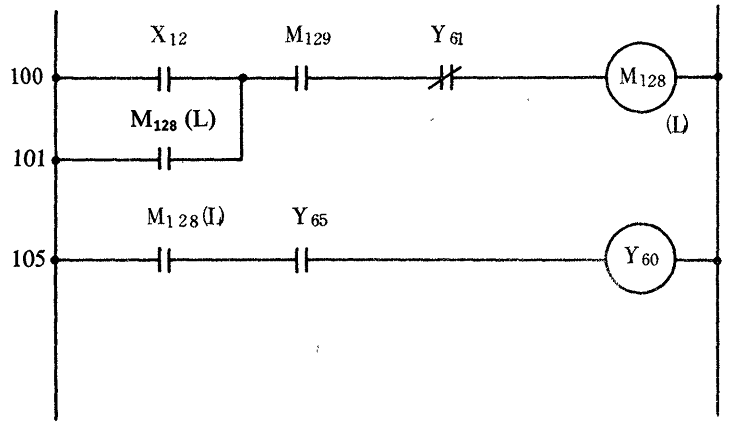


Step No.			Instruction			Device No.			
1	0	0	L	D		X		1	2
		1	O	R		X	5	0	(L)
		2	A	N	D	X		1	5
		3	A	N	I	Y		6	1
		4	O	U	T	Y	5	0	(L)
		5	L	D		X	5	0	(L)
		6	A	N	D	X		1	6
		7	O	U	T	Y		6	0

Fig. 5-22. Latch circuit by latch unit

(2) Latch switch ON with K2CPU

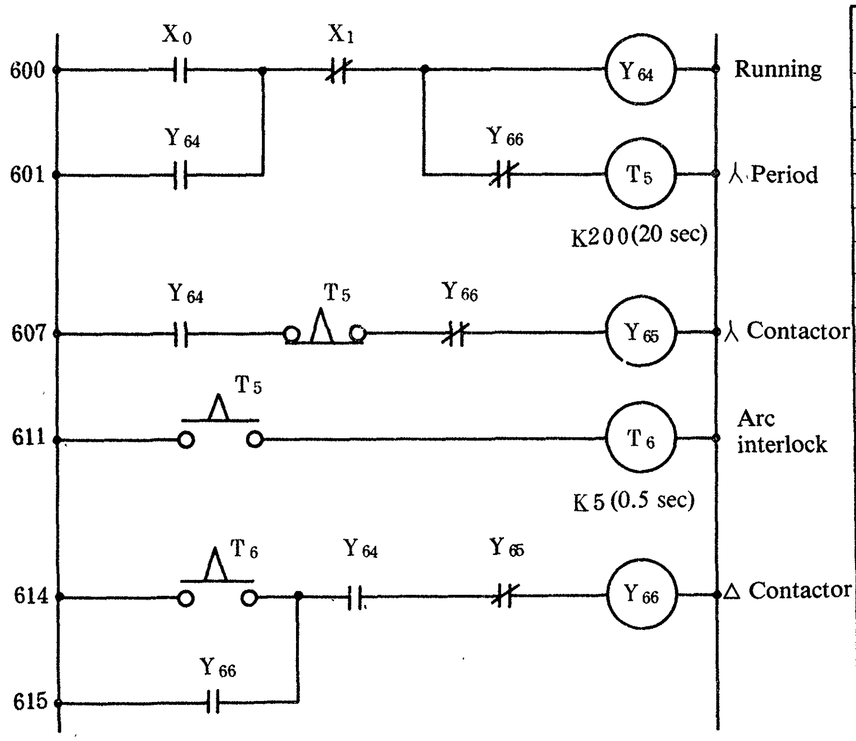
Collective holding at the time of power failure is executed by the latch switch for the temporary memories M128 to 253.



Step No.			Instruction			Device No.			
1	0	0	L	D		X		1	2
		1	O	R		M	1	2	8
		2	A	N	D	M	1	2	9
		3	A	N	I	Y		6	1
		4	O	U	T	M	1	2	8
		4	L	D		M	1	2	8
		5	A	N	D	Y		6	5
		6	O	U	T	Y		6	0

Fig. 5-23. Latch circuit by temporary memory (for K2CPU)

5-2-10. Star-delta motor starting circuit



Step No.			Instruction			Device No.			
6	0	0	L	D		X			0
		1	O	R		Y		6	4
		2	A	N	I	X			1
		3	O	U	T	Y		6	4
		4	A	N	I	Y		6	6
		5	O	U	T	T			5
		6				K	2	0	0
		7	L	D		Y		6	4
		8	A	N	I	T			5
		9	A	N	I	Y		6	6
	1	0	O	U	T	Y		6	5
	1	1	L	D		T			5
	1	2	O	U	T	T			6
	1	3				K			5
	1	4	L	D		T			6
	1	5	O	R		Y		6	6
	1	6	A	N	D	Y		6	4
	1	7	A	N	I	Y		6	5
	1	8	O	U	T	Y		6	6
	1	9	E	N	D				

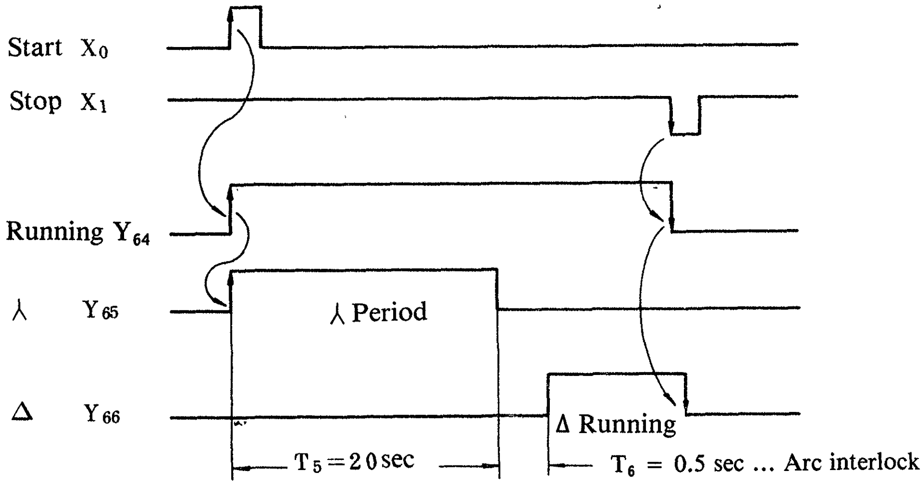


Fig. 5-24. Example for star-delta motor starting circuit

5-3. Program examples for data handling instructions

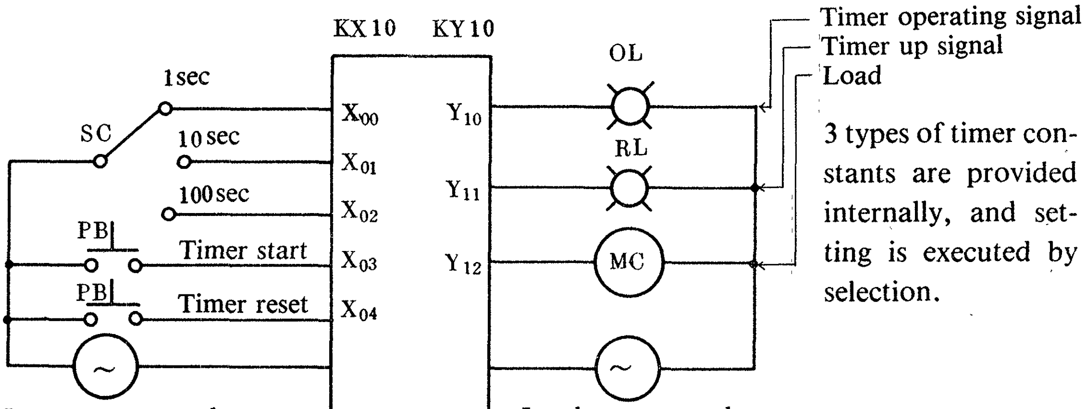
5-3-1. Switching of timer set values

(1)Subject

The set values for the timer time limits among the 3 types of 1 sec, 10 sec, and 100 sec can be switched by external switching signals.

The timer start signal is by push button input, display is made during operation, and output is put out at time-up.

(2)Outlines of programming



3 types of timer constants are provided internally, and setting is executed by selection.

Input power supply
Load power supply

Fig. 5-25. Switching of the timer set value

(3)Sequence design and coding example

(Sequence)

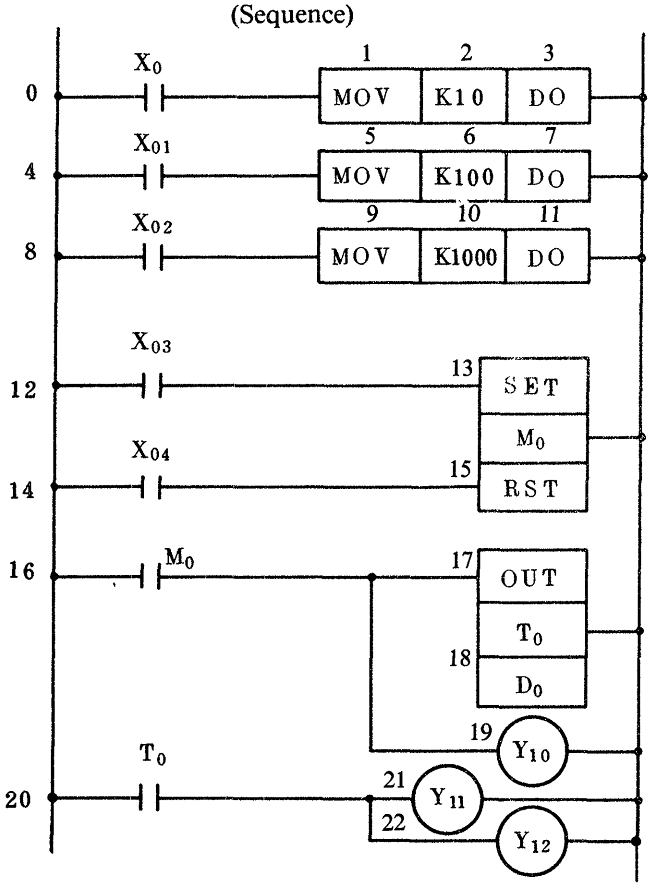


Fig. 5-26. Switching circuit for the timer set value

(Coding)

```
 0  LD    X0          15  RST   M0
 1  MOV               16  LD    M0
 2  K      10         17  OUT   T0
 3        D0          18        D0
 4  LD    X1          19  OUT   Y10
 5  MOV               20  LD    T0
 6  K      100        21  OUT   Y11
 7        D0          22  OUT   Y12
 8  LD    X2          23  END
 9  MOV
10     K  1000
11        D0
12  LD    X3
13  SET   M0
14  LD    X4
```

5-3-2. Output to the external indicator for the timer time limit

(1) Subject

Output of the present time limit value for the timers T0 (first digit) , T1 (second digit) , T2 (third digit) , and T3 (fouth digit) is executed as a BCD for each digit, and the numerical display is driven. (DC 5V)

(2) Programming outlines

KY32 (64 points) is connected to the I/O unit 0, KX10 (16 points) is connected to the I/O unit 1, and assignment of X and Y is executed as shown in the figure on the right. Because of the 64 Points of KY32, the upper digits 0 to 3 of the I/O adresses are occupied. The timer set values are T0 = 0.9 sec, T1 = 1.5 sec, T2 = 52 sec, and T3 = 131 sec.

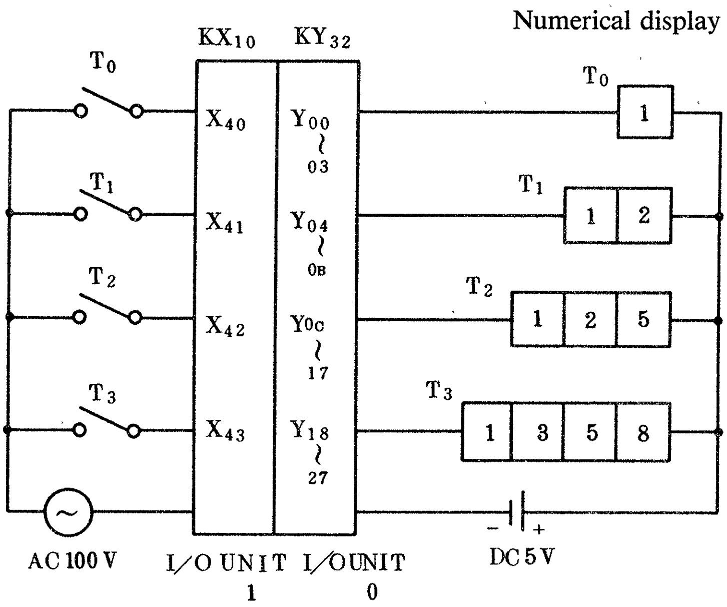
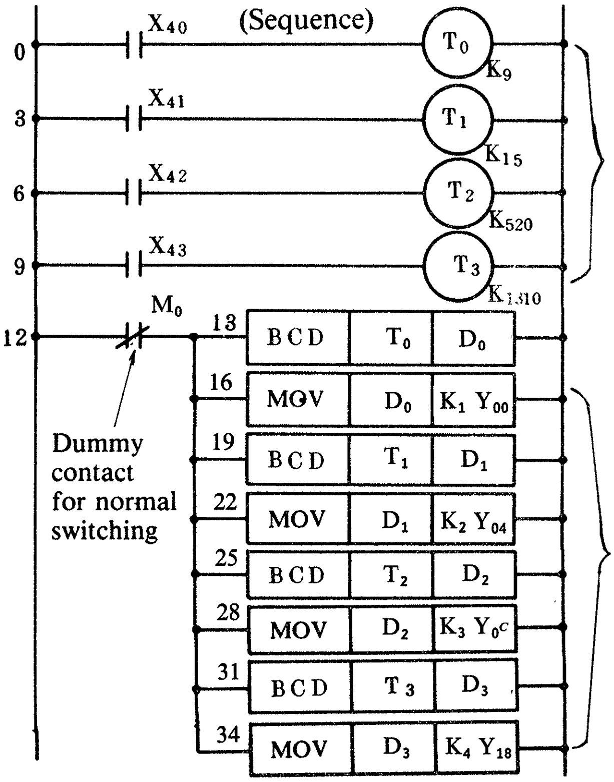


Fig. 5-27. External display of the timer time limit

(3) Sequence design and coding example



```
                 (Coding)

 0  LD    X40         19  BCD
 1  OUT   T0          20              T1
 2    K    9          21              D1
 3  LD    X41         22  MOV
 4  OUT   T1          23              D1
 5    K    15         24          K2  Y04
 6  LD    X42         25  BCD
 7  OUT   T2          26              T2
 8    K   520         27              D2
 9  LD    X43         28  MOV
10  OUT   T3          29              D2
11  . K   1310        30          K3  Y0C
12  LDI   M0          31  BCD
13  BCD               32              T3
14        T0          33              D3
15        D0          34  MOV
16  MOV               35              D3
17        D0          36          K4  Y18
18    K1  Y0          37  END
```

Fig. 5-28. External display circuit for the timer time limit

5-3-3. Counter setting by external digital switch

(1) Subject

Input of the setting values for the counters C4 (fourth digit), C5 (third digit), C6 (second digit), and C7 (first digit) is executed by external digital switches.

Digital switch input is composed of a BCD for each digit.

(2) Programming outlines

KX32 (64 points) is connected to the I/O unit0, and input X assignment is executed.

X0 is used in common for count input.

X1 is used as the count reset input.

X2 is used as the read-in signal for the counter data setting value.

These external input signals should be changed to pulses internally for easier use.

The set value input assignment is executed sequently from X10 on wards.

C4 is 4 bits x 4 = 16 points ... X10 to X1F

C5 is 12 points ... X20 to X2B, C6 is X2C to X33

C7 is X33 to X37.

(3) Sequence design example

(Sequence)

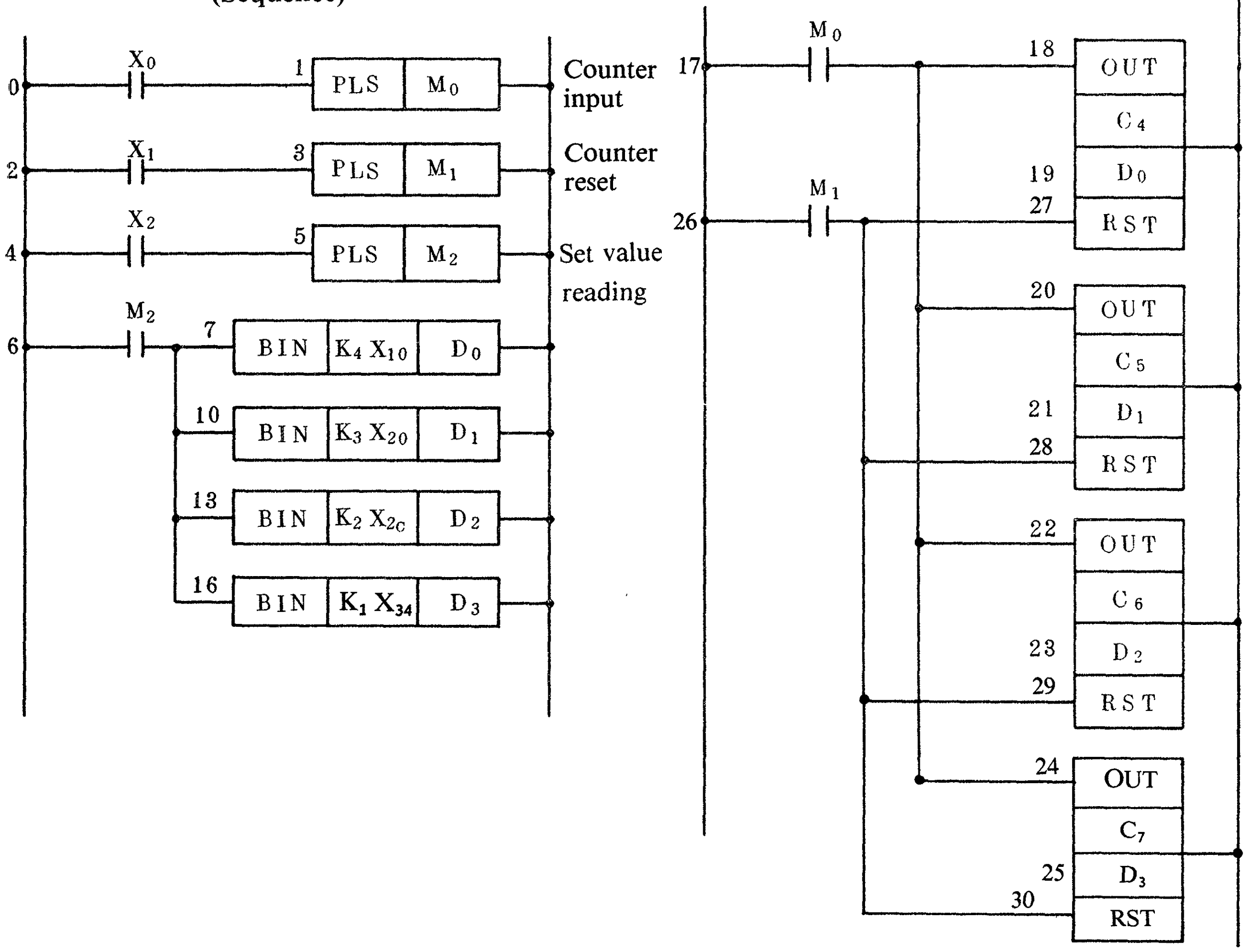


Fig. 5-29. External counter setting

5-3-4. Addition

(1) Subject

The data register D0 is cleared by ON of input X01, and flicker (0.5 + 0.5 sec) for the timers T0 and T1 is started simultaneously.

With each flicker, +2 is executed for the data register D0, and the output Y10 becomes ON when the contents of D0 become 20.

(D0 is monitored to confirm program operation.)

(2) Programming outlines

Data processing for addition of the external input signal X1 at each timer flicker, etc., is executed by the internal pulse formation and processing of this signal.

Sequence design and coding example

(Sequence) (Coding)

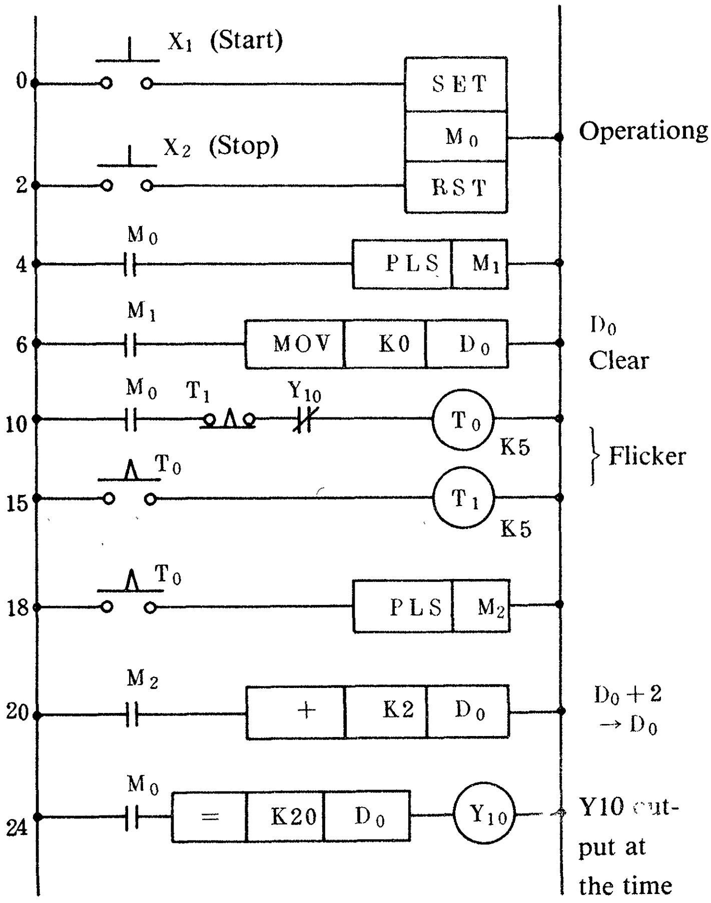


Fig. 5-30 Addition circuit

```
 0  LD    X1      17        K  5
 1  SET   M0      18  LD    T0
 2  LD    X2      19  PLS   M2
 3  RST   M0      20  LD    M2
 4  LD    M0      21  +
 5  PLS   M1      22        K  2
 6  LD    M1      23           D0
 7  MOV           24  LD    M0
 8        K  0    25  =
 9           D0   26        K 20
10  LD    M0      27           D0
11  ANI   T1      28  OUT   Y10
12  ANI   Y10     29  END
13  OUT   T0
14        K  5
15  LD    T0
16  OUT   T1
```

Note: In the test mode of the PU, M0 SET and RST enable start/stop operations.

5-3-5. Remote counter setting

(1) Subject

Remote counter setting is executed by means of a 4 digit digital switch, the present value display for the counter is driven by the output of BCD × 4 digits, and respective outputs are executed when 100 ahead, 50 ahead and count-up.

When the counter set value is smaller than 100, display of the setting error is out put.

(2) Program design outlines

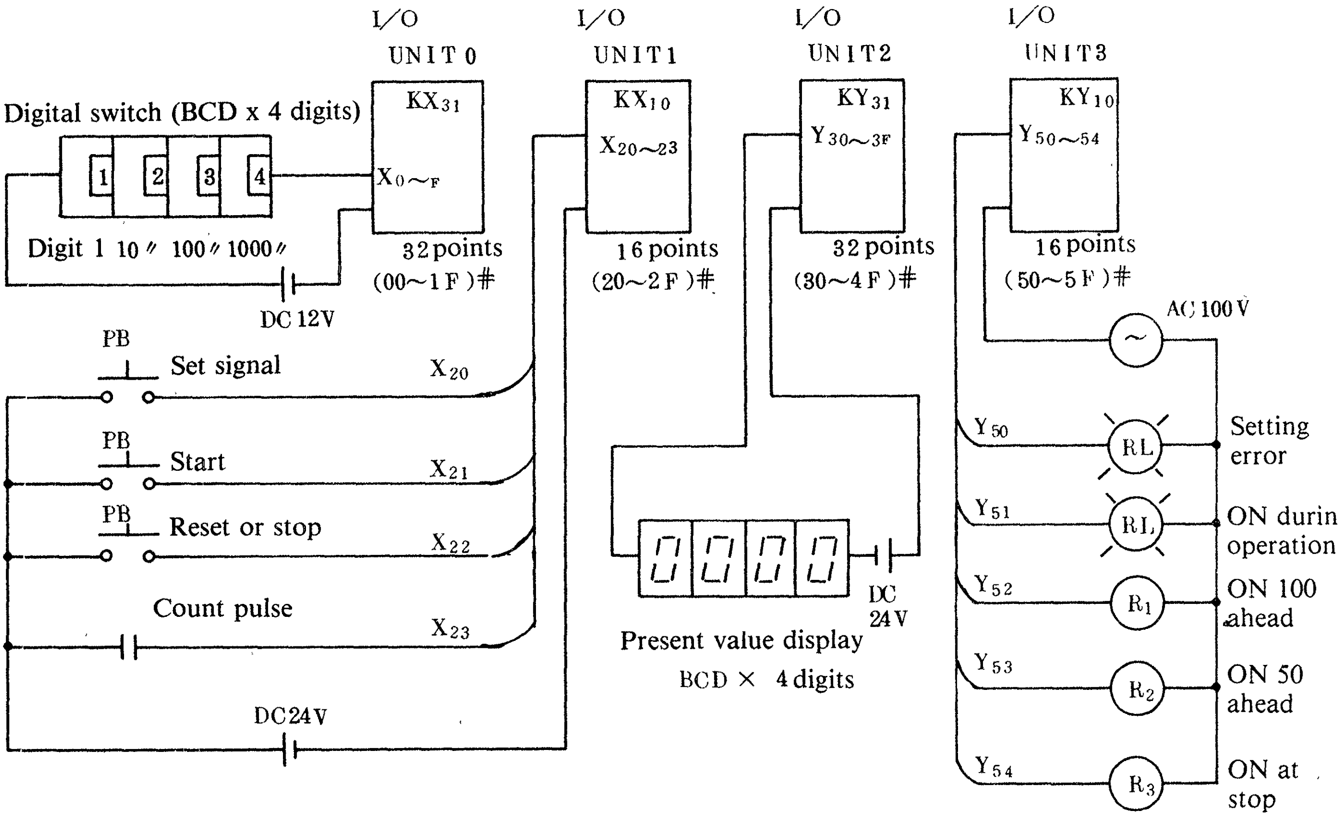


Fig. 5-31. Remote counter setting

The I/O units KX31, KX10, KY31 and KY10 are arranged as in the above figure. Assign the X and Y numbers carefully.

(3) Sequence design example

(The coding is omitted.)

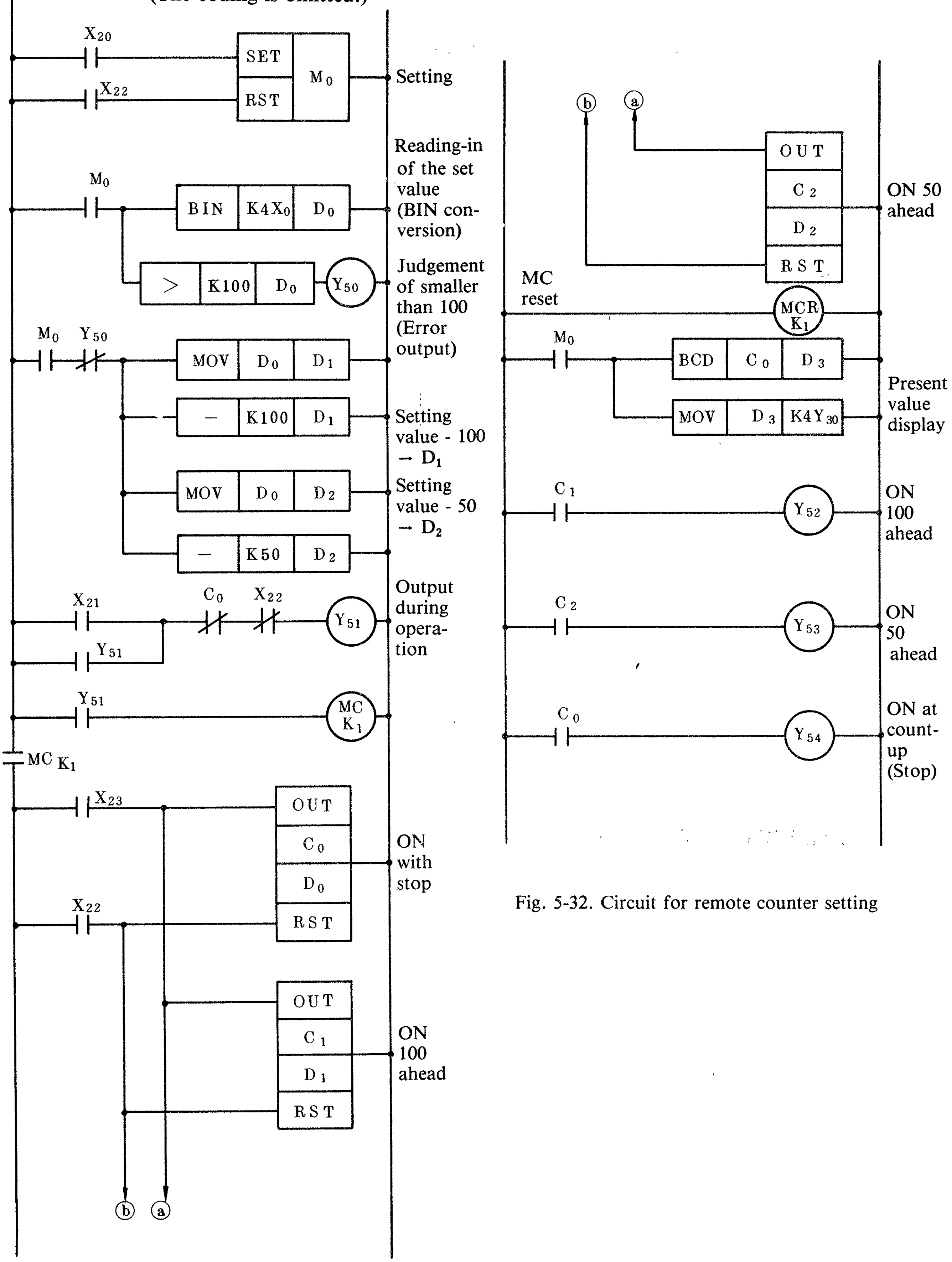


Fig. 5-32. Circuit for remote counter setting

5-3-6. Time counting for xx min xx sec

(1) Subject

Output of the time output for xx min xx sec to the 16 points of the output board KY10 is shown in the figure.

Notes: 1. The time starts from 00 min 00 sec with RUN of the CPU.

2. Return to 00 min 00 sec is executed after 59 min 59 sec.

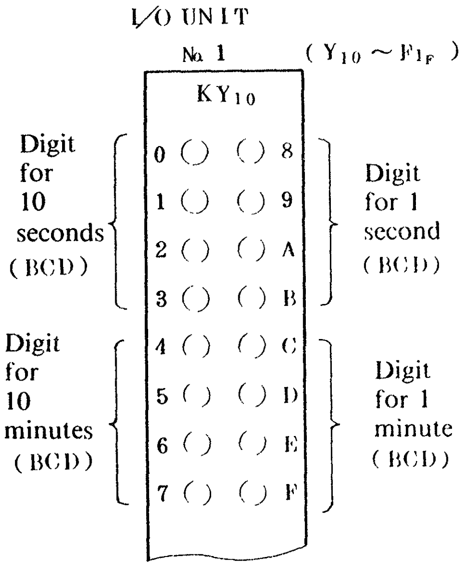
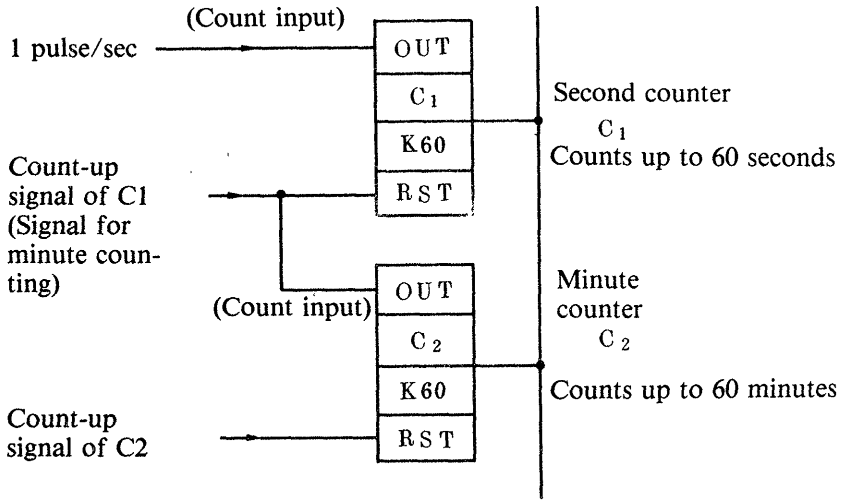


Fig. 5-33. Display assignment for seconds and minutes

(2) Programming outlines

(a) Establishing of a counter input pulse every 1 second.

(b) Establishing of a counter C1 for seconds and a counter C2 for minutes.



The contents of C1 and C2 are binary numbers.

Fig. 5-34. Counter circuit for seconds and minutes

(c) The contents of C1 and C2 are converted to BCD, and the output is executed separately for the first and second digit.
However, the fact that consecutive output of the 1st digit and the 10th digit in the sequence of Y10 to Y1F is not possible, is the point of this exercise.
Sequence:

(i) BCD conversion is executed for the contents of C1, and they are output into M0 to M7 via D0.

$$-\boxed{\text{BCD}}\boxed{C_1}\boxed{D_0}-$$

Note: As M can not be used as a destination for BCD, D is used.

$$-\boxed{\text{MOV}}\boxed{D_0}\boxed{K_2\ M_0}-$$

As C1 is smaller than 60, the BCD does not exceed 2 digits.
Accordingly, M0 to M7 (BCD x 2 digits) becomes:

$$\underbrace{\overset{M_0\sim M_3}{\boxed{\text{First BCD digit}}}\overset{M_4\sim M_7}{\boxed{\text{Second BCD digit}}}}_{\text{Contents (BCD) of C1}}$$

(ii) $\left.\begin{aligned} M_0\sim M_3 &\rightarrow Y_{18}\sim Y_{1B} \\ M_4\sim M_7 &\rightarrow Y_{10}\sim Y_{13} \end{aligned}\right\}$ they are output

Instead of an output for each point, a collective output via the data register is convenient.

$M_0\sim M_3 \rightarrow D_1 \rightarrow Y_{18}\sim Y_{1B}$

$M_4\sim M_7 \rightarrow D_2 \rightarrow Y_{10}\sim Y_{13}$

Collective output is pssible by proceeding this way.

(iii) As these conversion operations and output operations must be executed continuously, execution is made by using contact b of the dummy (normally ON signal) by $\overset{M_i}{-\!\!\not{|}\,|\!\!-}$

(iv) The same sequence as for C1 is also applicable to C2.

(3) Sequence design and coding example

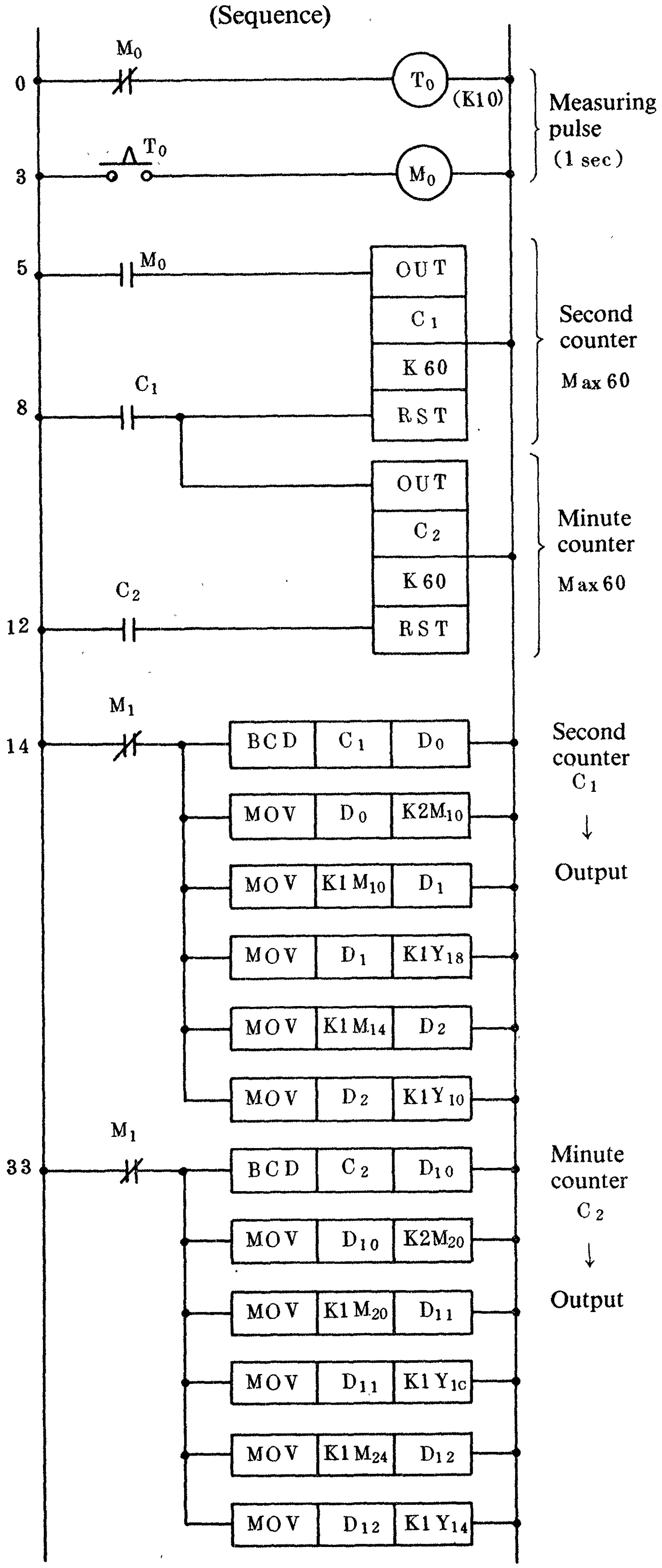


Fig. 5-35. Clock circuit

(Coding)

Step			Step		
0	LDI	M_0	33	LDI	M_1
1	OUT	T_0	34	BCD	
2		K 10	35		C_2
3	LD	T_0	36		D_{10}
4	OUT	M_0	37	MOV	
5	LD	M_0	38		D_{10}
6	OUT	C_1	39		$K2M_{20}$
7		K 60	40	MOV	
8	LD	C_1	41		$K1M_{20}$
9	RST	C_1	42		D_{11}
10	OUT	C_2	43	MOV	
11		K 60	44		D_{11}
12	LD	C_2	45		$K1Y_{1C}$
13	RST	C_2	46	MOV	
14	LDI	M_1	47		$K1M_{24}$
15	BCD		48		D_{12}
16		C_1	49	MOV	
17		D_0	50		D_{12}
18	MOV		51		$K1Y_{14}$
19		D_0	52	END	
20		$K2M_{10}$			
21	MOV				
22		$K1M_{10}$			
23		D_1			
24	MOV				
25		D_1			
26		$K1Y_{18}$			
27	MOV				
28		$K1M_{14}$			
29		D_2			
30	MOV				
31		D_2			
32		$K1Y_{10}$			

5-3-7. Shift instructions

(1) Subject

Programming so that the output signals at the outputs Y10, 11, 12, ... 1F for the output board of KY10 installed in the device number (I/O UNIT 1) are shifted every second.

(2) Programming outlines

(a) The meaning of SFT instructions

When SFT instructions are applied in regard to the temporary memory Mi, judgement is executed whether Mi- is 1 or zero.

For Mi-1 = 1, Mi-1 = 1 → 0 is executed, and Mi = 1 is obtained.

(b) Shift register M assignment

The temporary memories M assigned as shift registers are
M0, M1, M2 ... M15. (M16 is a dummy.)

(c) Composition of the shift circuit for M0 to M16

○ In the circuit for the top M0,
SET is used instead of SFT (shift) as below.

SET conditions for M0 or Initial set
Setting after 1 round as M16 = 1

○ Setting of the intermediate M1 to M15 by SFT.

As processing of the programmable controller is executed by condition processing in the sequence of the sequence diagram, the entire shift is completed at once with arrangement of the M numbers from the smallest upwards. Accordingly, the M numbers of the sequence diagram are arranged in reverse order from the higher numbers down.

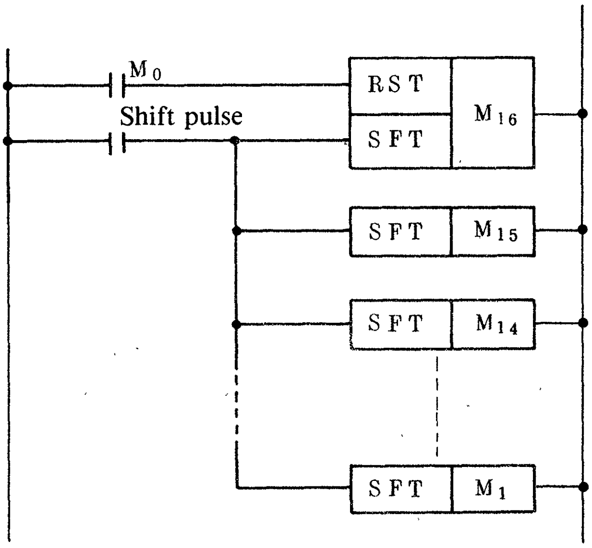


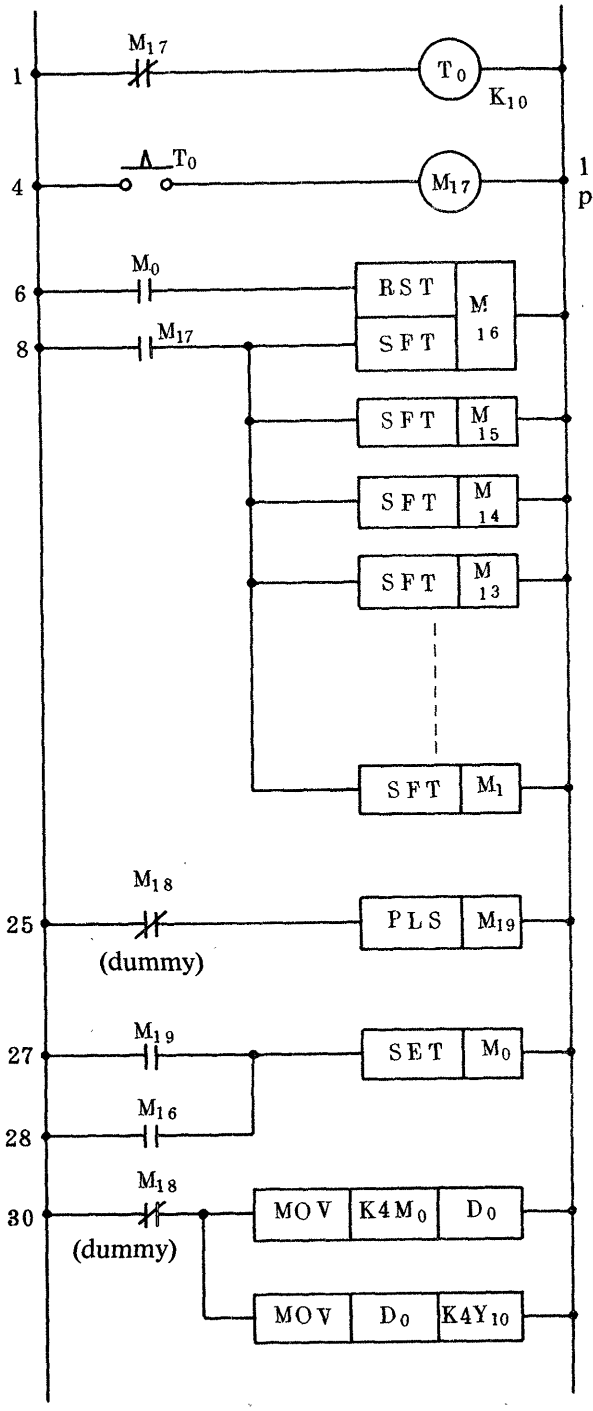
Fig. 5-36. Drawing up of the shift register circuit

(d) Output to KY10 (output unit)

Collective output at data by the instruction MOV for M0 to M15 is convenient.

(3) Sequence design and coding example

(Sequence)



1 sec pulse

Pulse generation at the time of CPU RUN start

Setting of the top bit

Fig. 5-37. Shift register circuit

(Coding)

```
Step
 1  LDI   M17
 2  OUT   T0
 3          K 10
 4  LD    T0
 5  OUT   M17
 6  LD    M0
 7  RST   M16
 8  LD    M17
 9  SFT   M16
10  SFT   M15
11  SFT   M14
12  SFT   M13
13  SFT   M12
14  SFT   M11
15  SFT   M10
16  SFT   M9
17  SFT   M8
18  SFT   M7
19  SFT   M6
20  SFT   M5
21  SFT   M4
22  SFT   M3
23  SFT   M2
24  SFT   M1
25  LDI   M18
26  PLS   M19
27  LD    M19
28  OR    M16
29  SET   M0
30  LDI   M18

Step
31  MOV
32          K4M0
33          D0
34  MOV
35          D0
36          K4Y10
37  END
```

5-3-8. Positioning control

(1) Subject

Positioning control is executed for a moving head with a position detection by a pulse generator.

Setting the input (X10 to X1B) for the target position ... 500 (3 digits decimal number)

Deceleration point ... Deceleration output (Y33) ON 15 ahead of the target position

Forward output ... From movement start to arrival at the target position (Y31) ON

Start input ... X0

Pulse generation detector input ... X1F

(2) Programming outlines

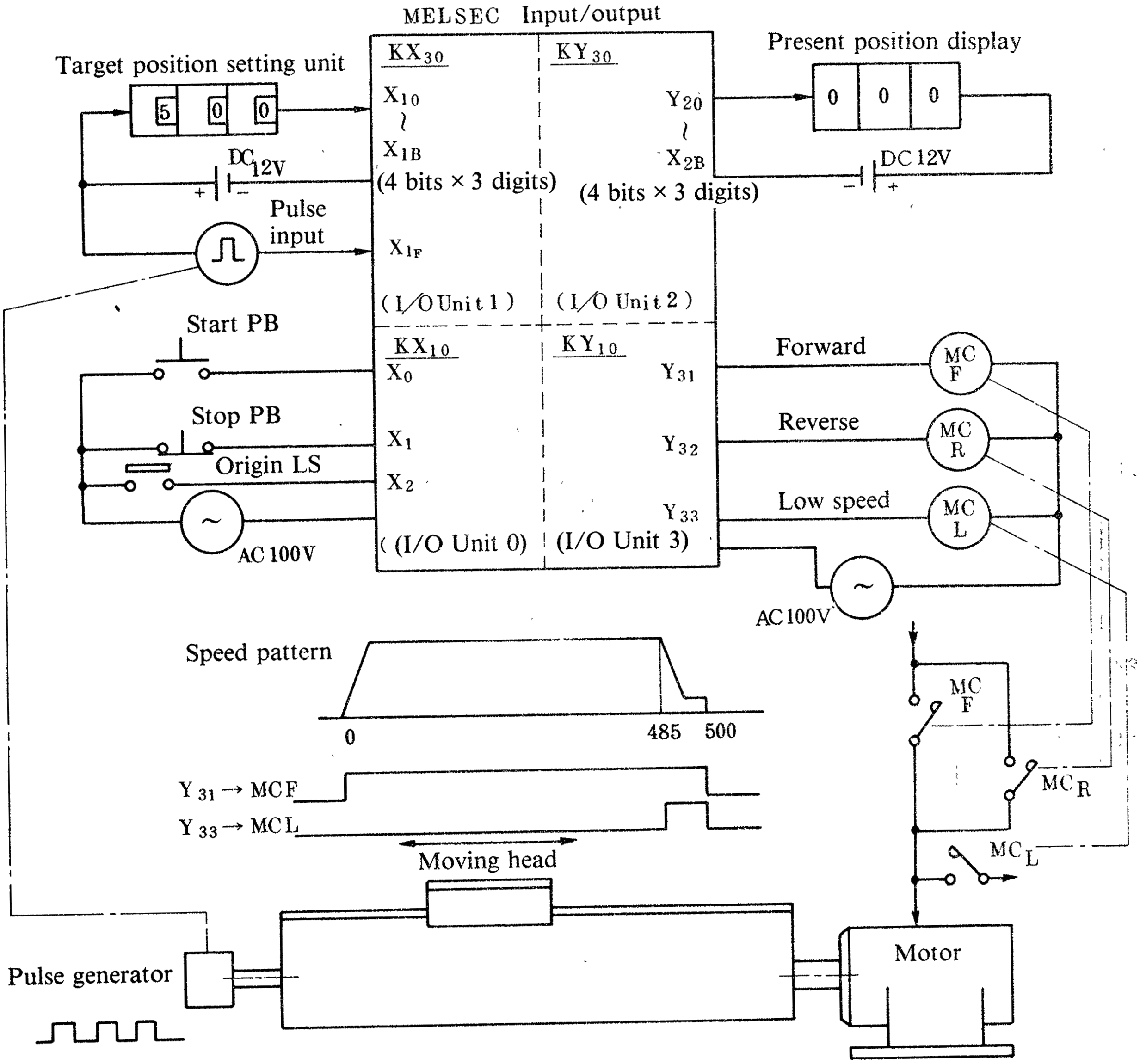
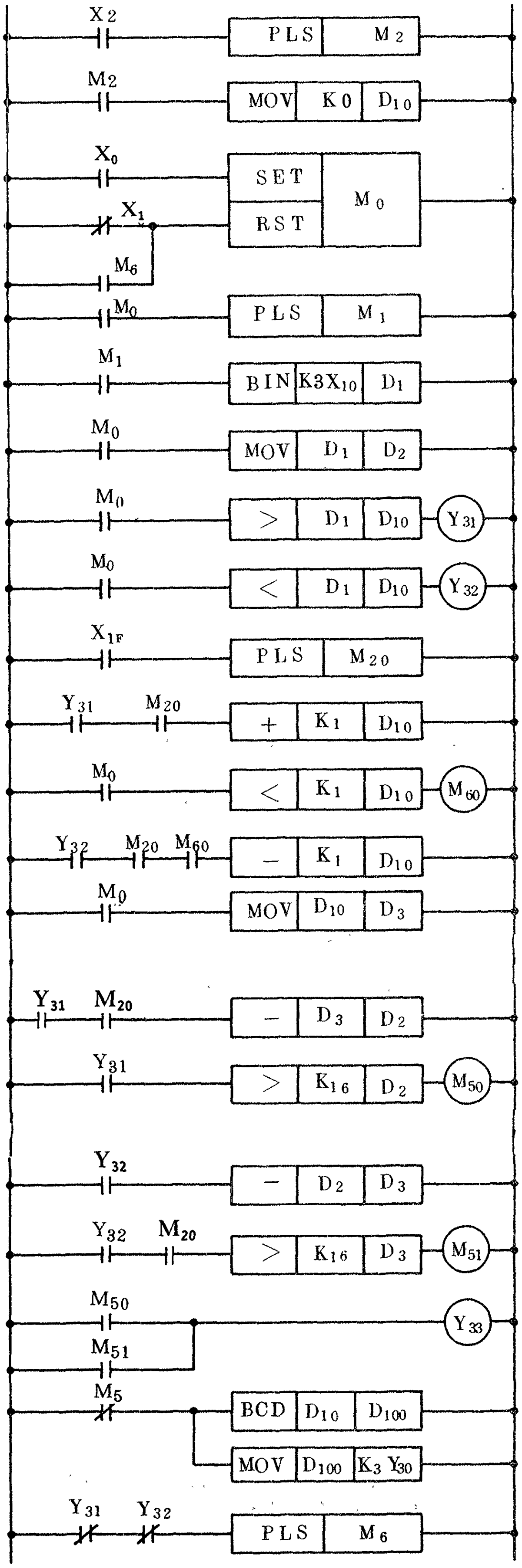


Fig. 5-38. Example for a positioning system

(b) Positioning control by absolute address (for full scan)



Origin limit

Present position register, origin reset

Start set, reset

Command value reading

Command value (temporary register)

Direction discrimination
Direction discrimination
(D_{10} : Present value)

Pulse input (position pulse)

Updating of the present value for the forward direction

Negative prevention for the present value

Updating of the present value for the reverse direction

Present value (temporary register)

Deceleration when the difference between the set value and the present value becomes 15 or less during movement in the forward direction.

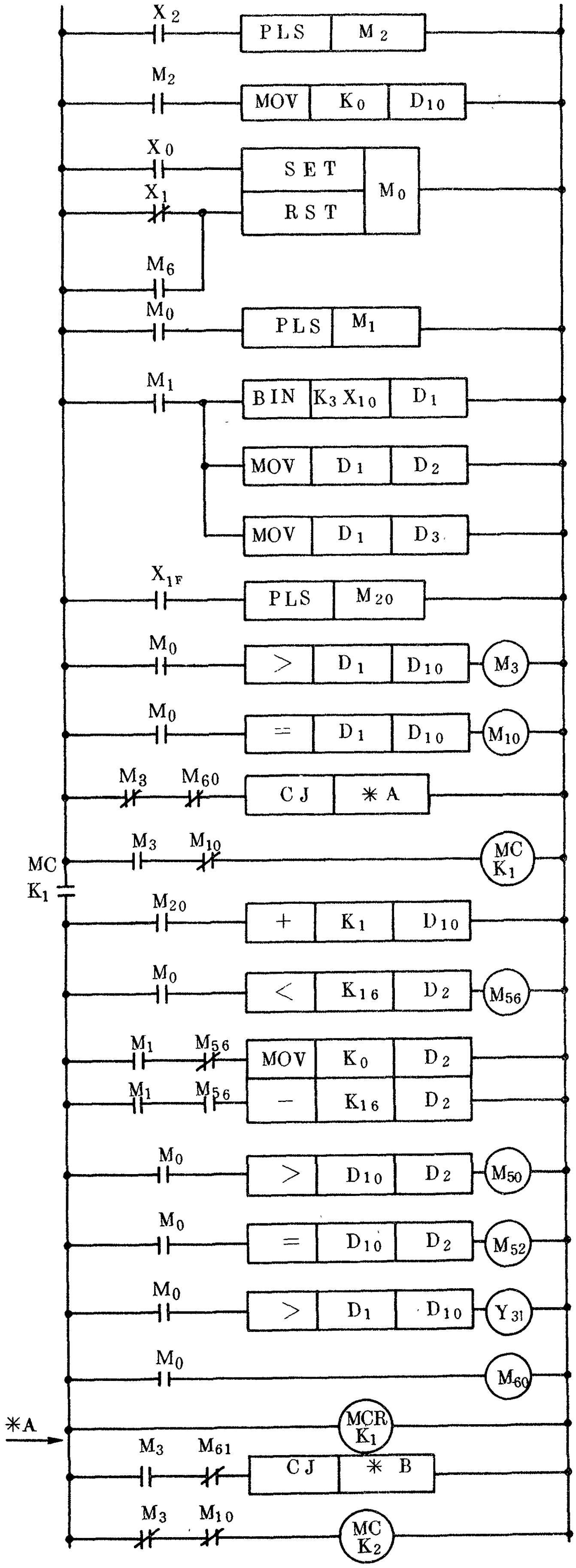
Deceleration when the difference between the set value and the present value becomes 15 or less during advance in the reverse direction.

Deceleration output

Present value display

Start reset

(b) Positioning control by absolute address (using MC and CJ)



Origin limit

Present position register, origin set

Start set, reset

Command value reading

Deceleration command register (Temporary register)

〃

Pulse input

Direction discrimination

Setting value = present value discrimination

The instructions for forward direction are jumped in the case of reverse direction.

Updating of the present value

Negative prevention for the deceleration value becomes a negative number

Deceleration value set ($D_2 > 0$)

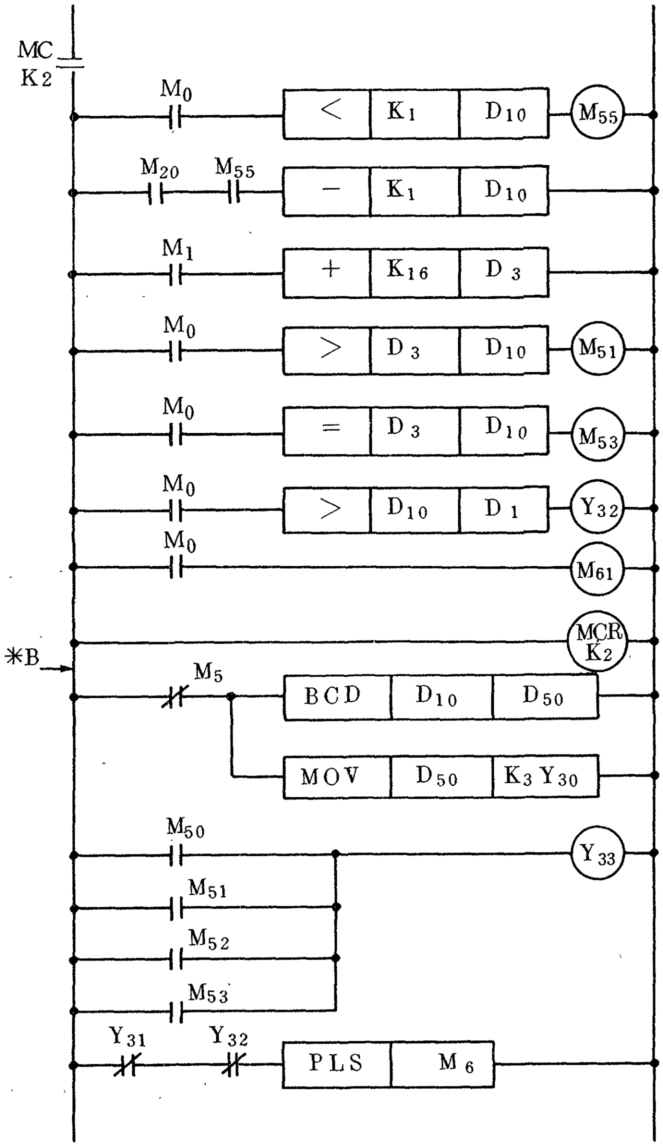
Deceleration output

〃

Forward output

CJ after output reset in MC K1

The instructions for reverse direction are jumped in the case of forward direction.



Negative prevension for the present value

Updating of the present value

Deceleration value set

Deceleration output

〃

Reverse output

CJ after output reset in M2 K2

Present value output

Deceleration output

Start reset

MELSEC-K

Unit arrangement table

Approved	Drawn up
	- -

Sheet No. 1 /

Note: Limit of the number of input/output points

K1CPU → 000# ~ 0FF# (256 points)

K2CPU → 000# ~ 1FF# (512 points)

For K12B, K22B ⇒ For K15B, K25B ⇒ For K18, K28B ⇒

Base connector name	POWER UNIT	CPU UNIT	I/O UNIT 0	I/O UNIT 1	I/O UNIT 2	I/O UNIT 3	I/O UNIT 4	I/O UNIT 5	I/O UNIT 6	I/O UNIT 7
Installed unit type name										

Upper 2 digits of the I/O number

Unit with occupation of 64 points

Unit with occupation of 32 points

Unit with occupation of 16 points

- 16 points device
- Blank (not attached)
- External failure monitor (KN61)
- Latch unit (KL61) (for 16 point switching)
- Timer unit (KT61)

- 32 point device (KX31, 41, KY31, 41)
- Latch unit (KL61) (for 32 point switching)

- 64 point device (KX32, KY32)
- Latch unit (KL61) (for 64 point switching)

I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application
0~3		0~3		0~3		0~3		0~3		0~3		0~3		0~3	
0 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7	
8~B		8~B		8~B		8~B		8~B		8~B		8~B		8~B	
C~F		C~F		C~F		C~F		C~F		C~F		C~F		C~F	
0~3		0~3		0~3		0~3		0~3		0~3		0~3		0~3	
□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7	
8~B		8~B		8~B		8~B		8~B		8~B		8~B		8~B	
C~F		C~F		C~F		C~F		C~F		C~F		C~F		C~F	
0~3		0~3		0~3		0~3		0~3		0~3		0~3		0~3	
□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7	
8~B		8~B		8~B		8~B		8~B		8~B		8~B		8~B	
C~F		C~F		C~F		C~F		C~F		C~F		C~F		C~F	
0~3		0~3		0~3		0~3		0~3		0~3		0~3		0~3	
□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7		□ 4~7	
8~B		8~B		8~B		8~B		8~B		8~B		8~B		8~B	
C~F		C~F		C~F		C~F		C~F		C~F		C~F		C~F	

MELSEC-K

Unit arrangement table

Approved	Drawn up
	- -

Sheet No. 2/

(2) Sheet Form 1-2

Note: Limit of the number of input/output points

K1CPU → 000# ~ 0FF# (256 points)

K2CPU → 000# ~ 1FF# (512 points)

For K12B, K22B ⇒

For K68 ⇒

Base connector name	POWER UNIT	I/O UNIT 0	I/O UNIT 1	I/O UNIT 2	I/O UNIT 3	I/O UNIT 4	I/O UNIT 5	I/O UNIT 6	I/O UNIT 7
Installed unit type name									

Upper 2 digits of the I/O number

	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application	I/O number	Application
Unit with occupation of 64 points Unit with occupation of 32 points Unit with occupation of 16 points • 16 points device • Blank (not attached) • External failure monitor (KN61) • Latch unit (KL61) (for 16 point switching) • Timer unit (KT61)	0~3		0~3		0~3		0~3		0~3		0~3		0~3		0~3	
	□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7	
	8~B		8~B		8~B		8~B		8~B		8~B		8~B		8~B	
	C~F		C~F		C~F		C~F		C~F		C~F		C~F		C~F	
•32 point device (KX31, 41, KY31, 41) • Latch unit (KL61) (for 32 point switching)	0~3		0~3		0~3		0~3		0~3		0~3		0~3		0~3	
	□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7	
	8~B		8~B		8~B		8~B		8~B		8~B		8~B		8~B	
	C~F		C~F		C~F		C~F		C~F		C~F		C~F		C~F	
• 64 point device (KX32, KY32) • Latch unit (KL61) (for 64 point switching)	0~3		0~3		0~3		0~3		0~3		0~3		0~3		0~3	
	□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7	
	8~B		8~B		8~B		8~B		8~B		8~B		8~B		8~B	
	C~F		C~F		C~F		C~F		C~F		C~F		C~F		C~F	
	0~3		0~3		0~3		0~3		0~3		0~3		0~3		0~3	
	□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7		□□ 4~7	
	8~B		8~B		8~B		8~B		8~B		8~B		8~B		8~B	
	C~F		C~F		C~F		C~F		C~F		C~F		C~F		C~F	

(3) Sheet Form 2

MELSEC-K

Input/output list

Approved	Drawn up

Sheet N

Base (I/O connection name)	I/O unit type name (number of points/unit)	Input/Output number			Device number	Name	Remarks (Connection terminal wire type, etc.)
				0			
				1			
				2			
				3			
				4			
				5			
				6			
				7			
				8			
				9			
				A			
				B			
				C			
				D			
				E			
				F			
				0			
				1			
				2			
				3			
				4			
				5			
				6			
				7			
				8			
				9			
				A			
				B			
				C			
				D			
				E			
				F			

Sheet form 2

(4) Sheet Form 3

MELSEC-K

Program sheets

Approved	Drawn up

Sheet N

Step No.	Command	Input, Output No.	Remarks
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
0			
1			
2			
3			
4			

Step No.	Command	Intput, Output No.	Remraks
5			
6			
7			
8			
9			
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			

(5) Sheet Form 4

MELSEC-K

Temporary Memory No.

254 Points (M0∼253)

Approved	Drawn up
	– –

Sheet No

Temporary Memory No.	Signal Name	Contents
M 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
M 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
M 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
M 0		
1		

Temporary Memory No.	Signal Name	Contents
M 2		
3		
4		
5		
6		
7		
8		
9		
M 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
M 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
M		
▨	▨	▨
M254	Battery abnormality	Input to external output and use
M255	RUN signal	Input to external ouput and use

(6) Sheet Form 5
MELSEC-K

Data Register List
96 Points (D0∼95)

Approved	Drawn up

Sheet N

Data Register No.	Data name (16 bit/data)	Contents
D 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
D 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		

Data Register No.	Data name (4 bit/data)	Contents
D 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
D 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		

(7) Sheet Form 6

MELSEC-K

Memory List of external breakdown
100 Points (F0~99)

Approved	Drawn up

Sheet No

Memory number or breakdown	Name of external breakdown	Contents of Breakdown • Conditions → Troubleshooting points
F 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
F 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
F 0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
F		

(8) Sheet Form 7

MELSEC-K

Timer, counter list

128 Points (T, C0~127)

Approved	Drawn up	Sheet No.

T/C	No.	Setting value "K" (0.1s/1c)				Names	Usage, Operation (Input count), etc.
	0						
	1						
	2						
	3						
	4						
	5						
	6						
	7						
	8						
	9						
	0						
	1						
	2						
	3						
	4						
	5						
	6						
	7						
	8						
	9						
	0						
	1						
	2						
	3						
	4						
	5						
	6						
	7						
	8						
	9						

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE MITSUBISHI DENKI BLDG MARUNOUCHI TOKYO 100 TELEX J24532 CABLE MELCO TOKYO

IB(NA)-64408B(8705) Printed in Japan(ROD)

Specifications subject to change without notice.