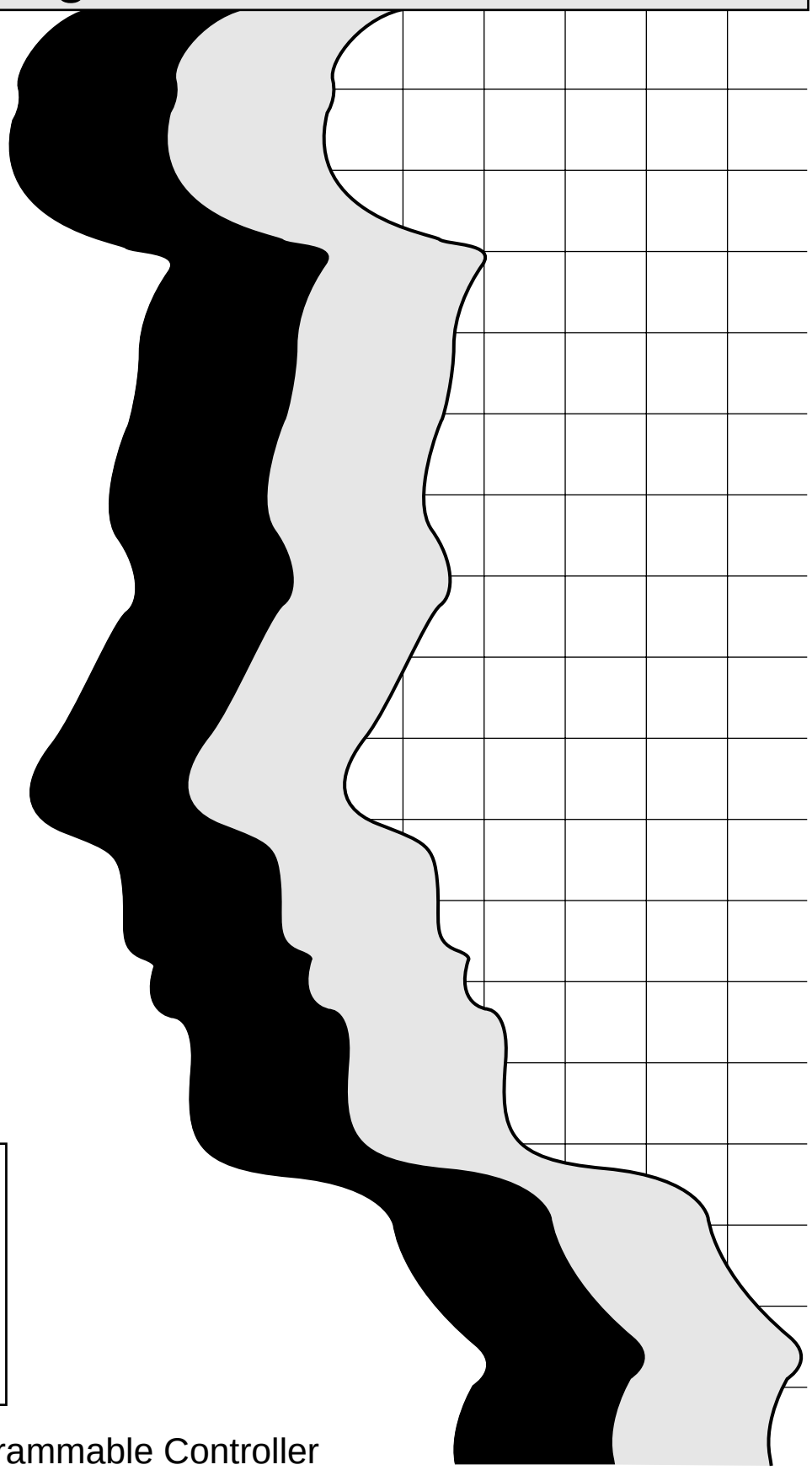
MITSUBISHI

type A0J2

Programming Manual



MELSEC

Mitsubishi Programmable Controller

REVISIONS

※The manual number is given on the bottom left of the back cover.

Print Date	*Manual Number	Revision
Nov., 1986	IB (NA) 66057-A	First edition
Dec., 1988	IB (NA) 66057-B	Correction Page 1-5, 2-1, 2-14, 4-9, 4-11, 5-1, 5-2, 5-4, 5-6, 6-1, 6-2, 6-3, 6-5, 6-6, 6-7, 6-12, 6-13, 6-14, 6-15, 6-16 "Instructions for Strategic Materials" added
Mar., 1995	IB (NA) 66057-C	Addition CONTENTS, Section 6.12

INTRODUCTION

Thank you for choosing the Mitsubishi MELSEC-A Series of General Purpose Programmable Controllers. Please read this manual carefully so that the equipment is used to its optimum. A copy of this manual should be forwarded to the end User.

CONTENTS

1. GENERAL DESCRIPTION

This Programming Manual describes the performance specifications, instructions, troubleshooting, etc. required for programming of the A0J2CPU (P23, R23).
The A0J2CPU is used for an independent system. The A0J2CPUP23 and A0J2CPUR23 are used for data link. Configurations of the independent system.and data link system are as follows:

- Independent system . . .Consists of the A0J2CPU, the I/O units for the A0J2CPU, and the extension base unit for the A series.
- Data link system . . . . . .Connected to MELSEC-NET. The A0J2CPUP23 (R23) is usable only as a slave station. Comprised of the A0J2CPUP23 (R23) and the I/O units for the A0J2CPU.

For the A0J2CPU (P23, R23), the same instructions may be used. Instructions are available in 80 types (21 types of sequence instructions, 59 types of basic and application instructions). For the A0J2CPU, link register (W) cannot be used.

The following Manuals are also related to usage of the A0J2CPU (P23, R23).

- A0J2 (CPU Unit Edition) User's Manual
- A0J2 (I/O Unit Edition) User's Manual
- Operating Manuals for peripheral equipment
- User's Manuals for special function units
- Data Link Unit User's Manual

POINT

The A0J2CPU, A0J2CPUP23, and A0J2CPUR23 are hereinafter referred to as "A0J2".

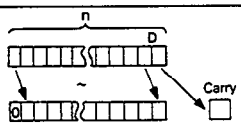
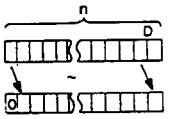
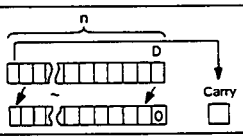
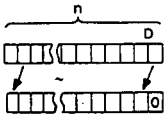
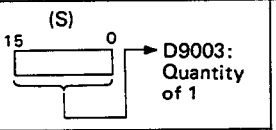
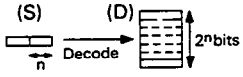
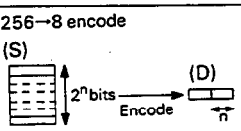
1.1 Instruction List

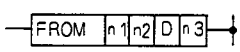
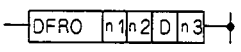
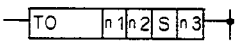
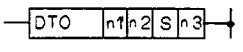
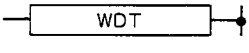
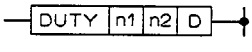
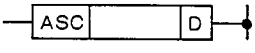
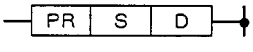
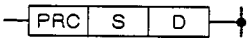
Table 1.1 shows instructions available for the A0J2. Details of the instructions will be described in Chapters 5 and 6.

Classification	Unit	Instruction Symbol	Symbol	Contents of Processing	Page
Sequence instruction		LD		Logical operation start ("a" contact operation start)	5-1
		LDI		Logical NOT operation start ("b" contact operation start)	
		AND		Logical product ("a" contact serial connection)	
		ANI		Logical product NOT ("b" contact serial connection)	
		OR		Logical sum ("a" contact parallel connection)	
		ORI		Logical sum NOT ("b" contact parallel connection)	
		ANB		AND between logical blocks (Serial connection between blocks)	
		ORB		OR between logical blocks (Parallel connection between blocks)	
		OUT		Output of device	
		SET	SET D	ON/OFF of device	
		RST	RST D	Reset of device	
		MC	MC n D	Master control start	
		MCR	MCR n	Master control reset	
		PLS	PLS D	At the rise of input signal, pulses of one program cycle are generated.	
		PLF	PLF D	At the fall of input signal, pulses of one program cycle are generated.	
		SFT	SFT D	1-bit shift of device	
		NOP	—	For erasure of nonprocessing program or space	
		END	—	Return to step 0 Be sure to write at the end of program	
		MPS	MPS	Storage of operation results	
		MRD	MRD	Read of operation results stored by MPS	
		MPP	MPP	Read and reset of operation results stored by MPS	
Logical operation instructions	=	LD=	LD= S1 S2	Continuity when (S1) = (S2) Non-continuity when (S1) ≠ (S2)	6-1
		AND=	AND= S1 S2		
		OR=	OR= S1 S2		
		16bits			

Classification		Unit	Instruction Symbol	Symbol	Contents of Processing	Page
Logical operation instructions	<> (≠)	16bits	LD<>	<> S1 S2	Continuity when (S1) ≠ (S2) Non-continuity when (S1) = (S2)	6-1
			AND<>	<> S1 S2		
			OR<>	<> S1 S2		
	>	16bits	LD>	> S1 S2	Continuity when (S1) > (S2) Non-continuity when (S1) ≦ (S2)	
			AND>	> S1 S2		
			OR>	> S1 S2		
	≦	16bits	LD<=	<= S1 S2	Continuity when (S1) ≦ (S2) Non-continuity when (S1) > (S2)	6-2
			AND<=	<= S1 S2		
			OR<=	<= S1 S2		
	<	16bits	LD<	< S1 S2	Continuity when (S1) < (S2) Non-continuity when (S1) ≧ (S2)	
			AND<	< S1 S2		
			OR<	< S1 S2		
	≧	16bits	LD>=	>= S1 S2	Continuity when (S1) ≧ (S2) Non-continuity when (S1) < (S2)	
			AND>=	>= S1 S2		
			OR>=	>= S1 S2		
BIN arithemetic operation instruction	+	16bits	+	+ S D	(D)+(S)→(D)	6-3
	−	16bits	−	− S D	(D)−(S)→(D)	
	*	16bits	*	* S1 S2 D	(S1)×(S2)→(D+1, D)	
	/	16bits	/	/ S1 S2 D	(S1) ÷ (S2) → Quotient (D), Remainder (D+1)	
	+1	16bits	INC	INC D	(D)+1→(D)	
	−1	16bits	DEC	DEC D	(D)−1→(D)	
BCD arithemtic operation instruction	+	BCD6digits	DB+	DB+ S D	(D+1, D)+(S+1, S) →(D+1, D)	6-5
	−	BCD6digits	DB−	DB- S D	(D+1, D)−(S+1, S) →(D+1, D)	

Classification		Unit	Instruction Symbol	Symbol	Contents of Processing	Page
BCD↔BIN convertion instruction	BCD	16bits	BCD	BCD S D	BCD conversion (S) → (D); (S): BIN(0~9999)	6-6
		24 bits	DBCD	DBCD S D	BCD conversion (S1+1, S1) → (D+1, D); (S1+1, S1): BIN(0~99999999)	
	BIN	16bits	BIN	BIN S D	BIN conversion (S) → (D); (S): BCD(0~9999)	
		24 bits	DBIN	DBIN S D	BIN conversion (S1+1, S1) → (D+1, D); (S1+1, S1): BCD(0~99999999)	
Data transfer instruction	Transfer	16bits	MOV	MOV S D	(S)→(D)	6-7
	Same data block transfer	16 bits	FMOV	FMOV S D n	(S) → (D), n	
Program branch instruction	Jump		CJ	CJ P**	Jump to P** after input condition holds.	6-8
	Program end		FEND	FEND	Processing is completed during sequence program.	
	Sub-routine call		CALL	CALL P**	Subroutine program of P** is executed after input condition holds.	
	Return		RET	RET	Return is made from subroutine program to sequence program.	
	Micro computer program call		SUB	SUB n	Subroutine call of micro-computer program.	
	COM		COM	COM	Suspends program operation at execution of the COM instruction and processes general date in the END instruction line.	
Logical operation instruction	Logical product	16bits	WAND	WAND S D	(D) AND (S) → (D)	6-13
	Logical sum	16bits	WOR	WOR S D	(D) OR (S) → (D)	
	Exclusive logical sum	16bits	WXOR	WXOR S D	(D) XOR (S) → (D)	

Classification		Unit	Instruction Symbol	Symbol	Contents of Processing	Page
Shift instruction	NOT exclusive logical sum	16bits	WXNR	WXNR S D	$\overline{(D) \forall (S)} \rightarrow (D)$	6-13
	2's complement	16bits	NEG	NEG D	(D) + 1 → (D)	
Date processing	Right ward shift	Bit unit	BSFR	BSFR D n		6-12
		Word unit	DSFR	DSFR D n		
	Left ward shift	Bit unit	BSFL	BSFL D n		
		Word unit	DSFL	DSFL D n		
Special function unit instruction	Count	16bits	SUM	SUM S		6-15
	Decode	2^nbits	DECO	DECO S D n		
	Encode	2^nbits	ENCO	ENCO S D n		

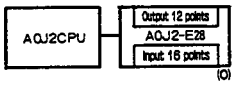
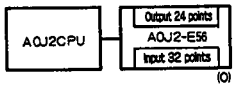
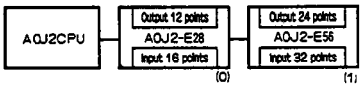
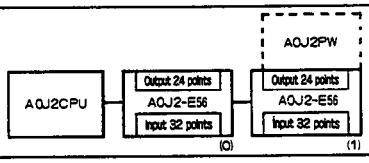
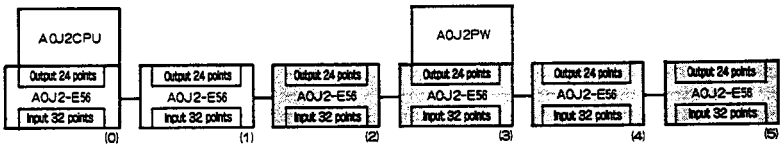
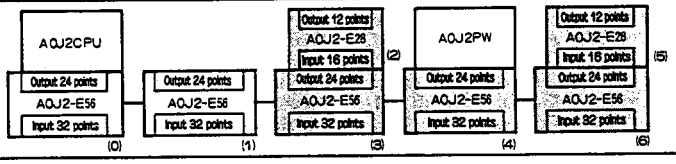
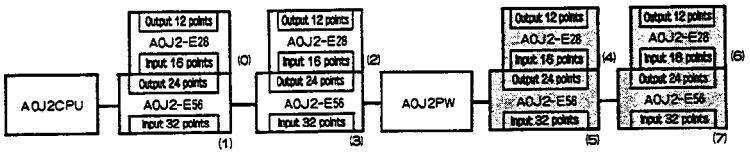
Classification			Unit	Instruction Symbol	Symbol	Contents of Processing	Page
Intelligent unit	Instruction for special function	Data read	1word	FROM		Data are read from special function unit.	6-16
			2words	DFRO			
		Data write	1word	TO		Data are written to special function unit.	6-17
			2words	DTO			
Other instructions	WDT reset			WDT		WDT is reset in sequence program.	6-19
	Timing clock		1 bit	DUTY		Timing clock shown below is generated.	6-20
	ASCII convertion			ASC		Converts alphanumeric characters into ASCII code and stores to D.	6-21
	ASCII print			PR		ASCII code of 8 points (16 characters) is output from specified device to output unit.	6-23
				PRC		Comment of specified device is converted into ASCII code and output to output unit. This is also applicable to the comment of device I.	

1.2 System Configurations

System configurations using the A0J2CPU are available in 2 types; the A0J2 system and the A0J2 system + extension base unit system.

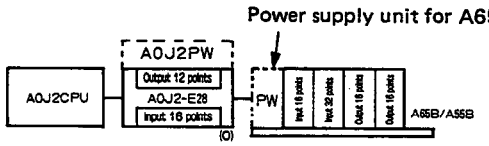
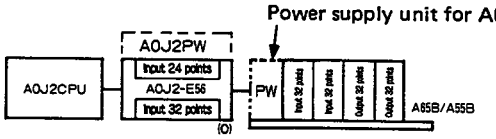
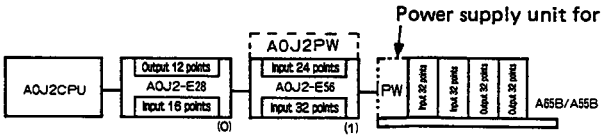
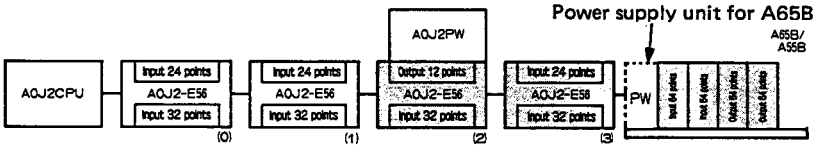
1.2.1 A0J2 system

This section describes systems consisting the A0J2CPU, I/O units, and extension power supply unit, their I/O points, etc.

I/O Points	System Configuration Example, I/O Number Assignment
28 points (Input: 16 points, Output: 12 points)	 * Numeral in parentheses is I/O unit setting number.
56 points (Input: 32 points, Output: 24 points)	
84 points (Input: 48 points, Output: 36 points)	
112 points (Input: 64 points, Output: 48 points)	
336 points (Input: 192 points, Output: 144 points)	
336 points (Input: 192 points, Output: 144 points)	
336 points (Input: 192 points, Output: 144 points)	
Maximum number of I/O points	336 points
Maximum number of I/O units	8 units
I/O cable	A0J2C01, A0J2C03, A0J2C06
Remarks	(1) The A0J2CPU system can be extended to a maximum of 336 points. The I/O and special function units may be freely combined up to eight units. In this case, one I/O unit occupies 64 I/O points. For details of I/O number assignment, refer to Section 2.4. (2) The A0J2CPU unit is incorporated with a power supply unit. Depending on a system, the extension power supply unit (A0J2PW) is required. For details, refer to the A0J2 User's Manual (CPU unit edition). (3) Only 1 A0J2PW unit may be used for 1 system. (4) ▭ indicates a unit which is supplied with power by A0J2PW.

1.2.2 A0J2 system + extension base unit system

This section describes the configurations and I/O points of systems which combine the A0J2 system in Section 1.2.1 and the extension base unit system (A65B, A55B).
(In these system configurations, 4 I/O units are loaded in the extension base unit.)

I/O Points	System Configuration Example, I/O Number Assignment
108 points (Input: 64 points Output: 44 points)	 A0J2PW : Indicates that when A55B is used, A0J2PW is required. *Value in parentheses indicates I/O unit number.
184 points (Input: 96 points Output: 88 points)	
196 points (Input: 112 points Output: 84 points)	
480 points (Input: 265 points Output: 224 points)	
Maximum number of I/O points	480 points
Maximum number of I/O units	4 units + 1 extension base unit (Maximum of 4 units may be loaded into the extension base unit)
I/O cable	A0J2C01, A0J2C03, A0J2C06/A0J2C04B, A0J2C10B (for extension base)
Remarks	(1) A maximum of 4 A0J2 I/O units may be freely combined. (2) Be sure to install the extension base unit (A65B, A55B) at the final stage of system and set the extension base stage setting switch to "1". (3) Each of the A0J2 I/O units and the I/O units in extension base unit occupies 64 I/O points. For details of I/O number assignment, refer to Section 2.4. (4) The following special function units may be loaded into extension base unit. 1) Positioning unit (AD71) 2) High-speed counter unit (AD61) 3) A/D conversion unit (A68AD) 4) D/A conversion unit (A62DA) 5) Intelligent communication unit (AD51): only 1 unit usable for 1 system. (5) The A0J2CPU is incorporated with a power supply unit. Depending on a system, the extension power supply unit (A0J2PW) is required. Especially when A55B is used as extension base unit, the A0J2PW is required. (6) [ ] indicates a unit which is supplied with power by the A0J2PW. When the extension base unit is the A65B, power is supplied from the power supply unit in base unit to the I/O units in A65B.

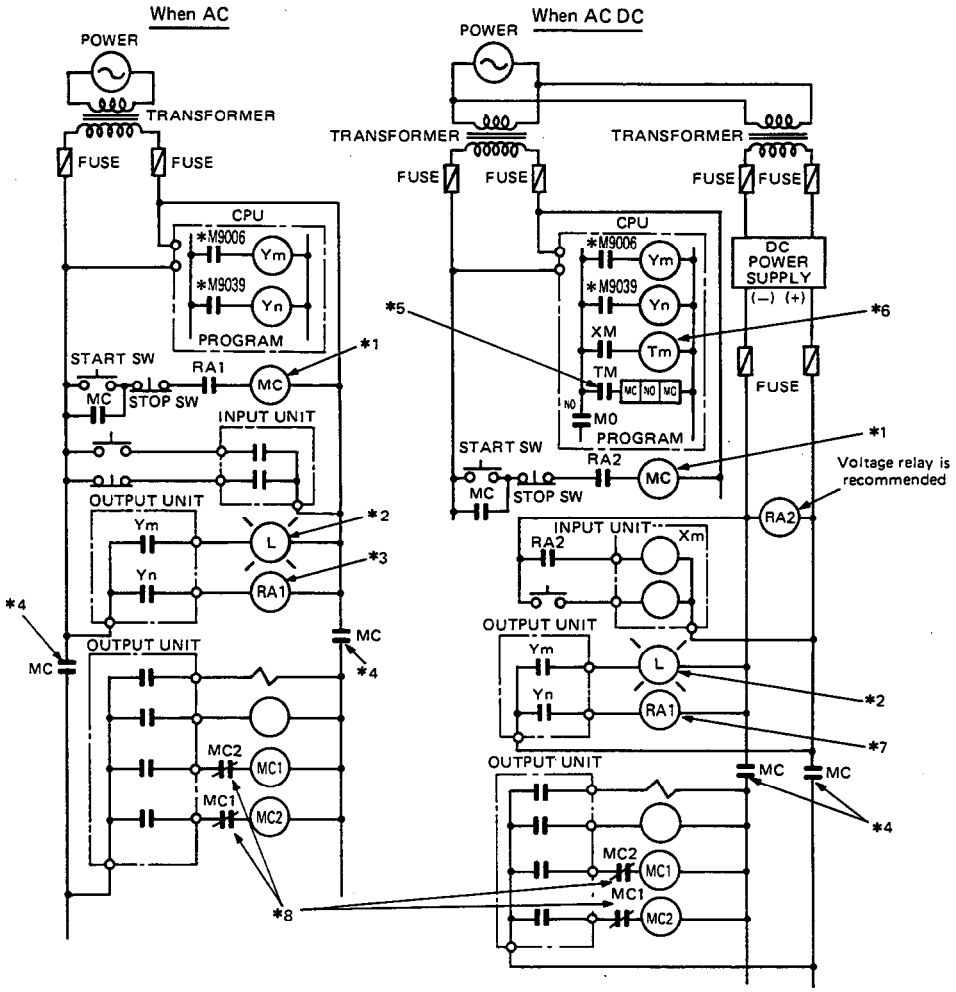
1.3 Concept of Fail Safe Circuit

When the power of system is turned on or off, process output may not temporarily perform normal operation due to the difference between the delay time and rise time of the power supply of programmable controller main unit and the external power supply (especially DC) for the process. Also, at the time of an error of the external power supply, output process may possibly make an erroneous operation.

In order to prevent the aforementioned erroneous operations from resulting in an erroneous operation of the entire system and also for safety reasons, constitute circuits (such as emergency stop circuit, protection circuit, and interlock circuit), that prevent machine damage or an accident due to erroneous operation outside the programmable controller.

A system design circuit example based on the above concept is shown on the following page.

System design circuit example



*1: *Run/stop circuit (run can be made when RA1, run output of programmable controller, turns on)*
*2: *Alarm indicator (lamp or buzzer)*
*3: *On when run by M9039*
*4: *Power of output equipment is turned off when stopped. (At the time of emergency stop or stop due to operation of limit switch)*
*5: *DC power supply establishment input signal*
*6: *Set timer at the interval of time to when DC POWER SUPPLY is established. (Set a value to approx. 0.5 seconds)*
*7: *On when run by M9039*
*8: *Interlock circuit (Make up an interlock circuit in the exterior for the area which may lead to contrary operations such as forward and reverse rotations, machine damage, or accident)*

The power-on procedure is as follows:

For AC

1) Turn on the power.
2) Set the CPU to RUN mode.
3) Turn on the start switch.
4) When the magnetic contactor (MC) turns on, output equipment is driven by program.

For AC/DC

1) Turn on the power.
2) Set the CPU to RUN mode.
3) When DC power is established, RA2 turns on.
4) When DC power is established 100%, timer (TM) is turned on.
   (The set value of TM should be a period of time from "on" of RA2 to establishment of 100% DC voltage. Set the set value to approximately 0.5 seconds.)
5) Turn on the start switch.
6) When the magnetic contactor (MC) turns on, output equipment is driven by program.
   (When a voltage relay is used for RA2, the timer (TM) in the program is not required.)

Fig. 2.1 Failsafe Circuit Example

2. CPU UNIT PERFORMANCE SPECIFICATIONS

This chapter explains A0J2CPU units' performance specifications, devices, operation processing method, etc.

2.1 CPU Unit Performance Specification

Table 2.1 shows performances and specifications of the A0J2CPU Unit.

Item		Specifications
Control method		Stored program, repetitive operation
I/O control method		Direct system
Programming language		Language dedicated to sequence control (combined use of relay symbol type, logic symbolic language, and SAP language)
Instruction	Sequence instruction	21 types
	Basic instruction	38 types
	Application instruction	21 types
Processing speed		Sequence instruction 4.4 to 5.6 μs/step
Memory capacity and memory type	Memory capacity	Max. 7K steps
	4KEROM	3K steps
	4KRAM	3K steps
	4KROM	3K steps
	16KRAM	7K steps
	8KROM	7K steps
Number of I/O points		336 points (Maximum 480 points when using extension base unit)
Comment		Can be created with GPP/PHP/HGP (Comment entered into the CPU are only 95 points, F0 to F94.)
Latch (power failure compensation) function		Available for L, B, T, C, D (Range to be set with A6GPP or A7PU.)
Remote RUN/STOP function		Can be operated with PHP/HGP/PHP or A7PU.
Operation mode at error		Operation continued at software instruction error
STOP to RUN output mode		Operation result at STOP is regenerated.
Print title entry		Print title cannot be entered into the CPU. However, it can be created with GPP/PHP/HGP.
Self-diagnostic function		Watch dog timer error monitor, battery error, AC down detection, blown fuse detection, etc.
Allowable instantaneous power failure period		Within 10ms
At power on, at power restoration after power failure		Automatic restart when "RUN" switch is set to ON. (Initial start)
IC-RAM latch device back-up		Battery backup, lithium battery used (5 years of guarantee period)
Parameter		Latch range to be set with PHP/HGP/PHP or A7PU.
Microcomputer mode		Other than sequence program area. The content of utility FD is written into the microcomputer area.
Watch dog time (WDT)		200ms fixed

Table 2.1 CPU Unit Performance Specifications

POINT

In the microcomputer area, the user cannot create microcomputer programs. Only data on the utility FD may be written into the microcomputer area.

2.2 Devices

2.2.1 Device list

1) Devices useable for the A0J2 are classified as follows:

a) Bit device: Handles 1-bit data.
b) Word device: Handles 16-bit (1-word) data.
c) Constant: Decimal or hexadecimal value.
d) Pointer: Indicates the jump destination of branch instruction.
e) Level: Indicates the master control level.

2) Devices and applicable ranges available for the A0J2 are described below.

Classification	Device		Applicable Range	Remarks
Bit device	Input		X/Y: Total 336 points	• A total of 480 X/Y points when extension base is used. • X/Y numbers are indicated in hexadecimal.
	Output			
	Internal relay Latch relay		M0 to 1023 (1024 points) L1024 to 2047 (1024 points)	• The number of internal relays and latch relays to be set with A6GPP or A7PU.
	Special relay		M9000 to 9255 (256 points)	
	Link relay		B0 to 3FF (1024 points)	• B numbers are indicated in hexadecimal.
	Annunciator		F0 to 255 (256 points)	• F0 to 94 (95 points) comments may be entered into CPU.
Word device	Timer	100ms timer	T0 to 79 (80 points)	• The point numbers of 100ms timers, 10ms timers, and 100ms retentive timers are fixed. • Timers and counters are available in 128 points, respectively.
		10ms timer	T80 to 119 (40 points)	
		100ms retentive timer	T120 to 127 (8 points)	
	Counter		C0 to 127 (128 points)	
	Data register		D0 to 511 (512 points)	
	Special register		D9000 to 9127 (128 points)	
	Link register		W0 to 3FF (1024 points)	• W numbers are indicated in hexadecimal.
	Index register		Z (1 point)	
			V (1 point)	
Counter	Hexadecimal constant		K–32768 to 32767 (16 bit instruction)	
			K0 to 999999 (24 bit instruction)	
	Hexadecimal constant		H0 to FFFF (16 bit instruction)	
			H0 to FFFFFF (24 bit instruction)	
Pointer	Pointer		P0 to 63 (64 points)	
Level	Nesting		N0 to 7 (8 levels)	

Table 2.2 Device List

REMARKS

1) Among the devices available for the A1, A2, and A3CPU, the following devices are not available for the A0J2.

a) A0J2CPU: Accumulator (A0, A1), file register (R), link register (W)
b) A0J2CPUP23(R23): Accumulator (A0, A1), file register (R)

2.2.2 Input/output

Via the inputs and outputs, communication is made between the PC and external equipment.
For the inputs, use external ON/OFF data in programs. Use the outputs to provide program operation results from the output unit to the outside.

(1) Input X

1) Inputs provide commands and data to the PC from external equipment such as pushbutton, select switch, limit switch, and digital switch.

2) Regarding that one point of input incorporates a virtual relay Xn in the PC, the N/O contact and N/C contact of that Xn are used in the program.

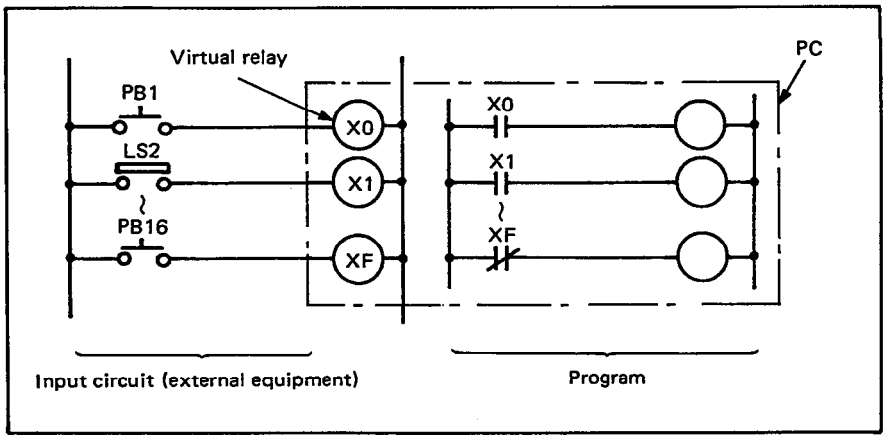


Fig. 2.1 Input (X) Concept

3) There is no restriction on the number of N/O contacts and N/C contacts of Xn used in the program.

(2) Output Y

1) Outputs are provided to the external equipment, such as solenoid, magnetic contactor, signal light, and digital indicator, as the control result of program.

2) Outputs can be fetched to the outside as an equivalent to 1 N/O contact.

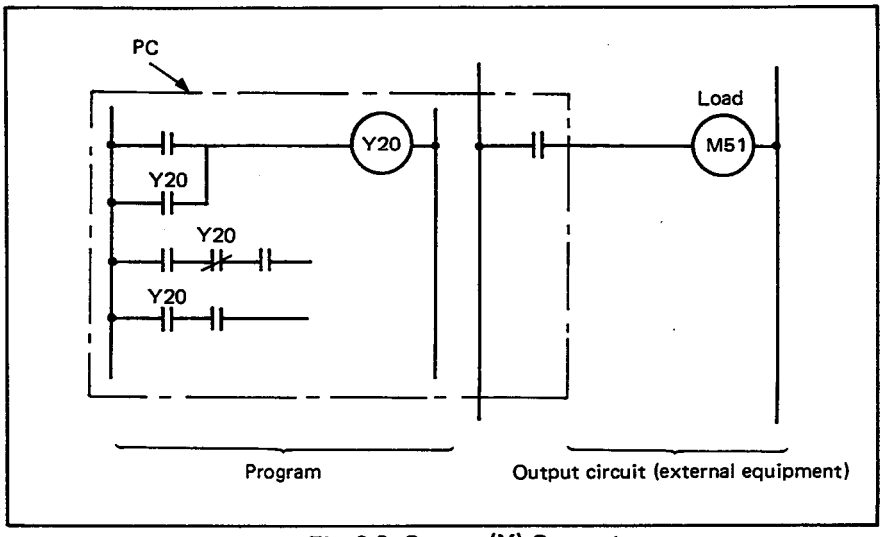


Fig. 2.2 Output (Y) Concept

3) There is no restriction on the number of N/O contacts and N/C contacts of Yn used in the program.

POINT

(1) In the A0J2, I/O numbers are determined by I/O unit setting numbers. Referring to Section 2.4 "Concept of I/O Number Assignment", specify proper I/O numbers.

2.2.3 Internal relay, latch relay

The internal relay and latch relay are auxiliary relays inside the PC. There is no restriction on the number of contacts (N/O contacts and N/C contacts) used in the program.

(1) Internal relay M

1) The internal relay is an auxiliary relay which is disabled for latch (power failure compensation). Therefore, all internal relays are turned off if:

   a) the PC power is turned on;
   b) reset is performed; and
   c) latch clear is made.

(2) Latch relay L

1) The latch relay is an auxiliary relay which is allowed for latch (power failure compensation). Therefore, the previous states are retained if:

   a) the PC power is turned on;
   b) reset is performed; and
   c) latch clear is made.

2.2.4 Link relay B

1) The link relay is an internal relay for data link.

2) When the link relays are used for data link, use of link relays for ON/OFF control as coils at one station (master or local station) and as contacts at other stations (master or local station) enables read of ON/OFF data. Therefore, this link relay allows the communication of ON/OFF data from the master station to the local station, from the local station to the master station, and between local stations.

3) To use the link relay for data link, it is necessary to set the link range (range in which the link relays are used as coils at stations) to the master station. Link relay numbers which are not set in the link range can be used instead of internal relays at stations.

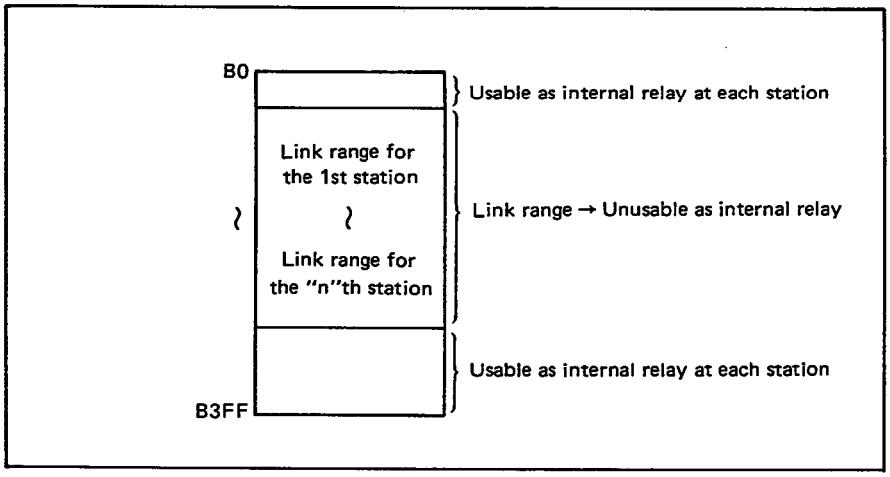


Fig. 2.3 Link Relay Assignment

4) There is no restriction on the number of N/O contacts and N/C contacts of link relay used in the program.

POINT

(1) The A0J2CPU is not enabled for data link. Therefore, the link relay may be used only in the same manner as the internal relay. Using the A0J2CPUP23(R23), the link relays can be used for the data link system.

2.2.5 Annunciator F

1) The annunciator is a device for failure detection. Create a failure detection program using the annunciator and scan the annunciator during run of the A0J2. This turns on the annunciator when a failure occurs.

2) When the annunciator turns on, the enabled annunciator number (F number) is stored into special register D9009.

3) When the F numbers in D9009 are reset,

   - the reset F number coils turn off.
   - the lowest F number, among enabled F numbers, is stored into D9009.

4) To clear the enabled F number, execute RST F ⬚ instruction.

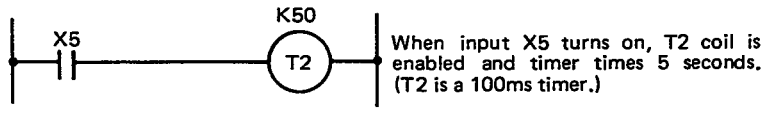
2.2.6 Timer T

The timers are of up-counting type. When the present value reaches the set value, the timer times up.
When the coil of timer enables, timing is initiated. When the timer times up, the contact of that timer enables.

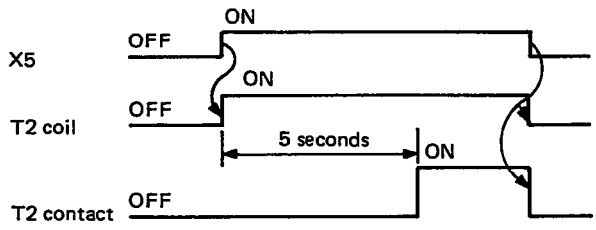
(1) 100ms, 10ms timers

1) When the timer coil disables, the present value is reset to 0 and the contact also disables.

EXAMPLE



The timing chart of the above diagram is as follows.



2) Set the set value in decimal. The set value can be specified in the following range:

	Set Value
100ms timer	1 to 32767 (0.1 to 3276.7 sec)
10ms timer	1 to 32767 (0.01 to 327.67 sec)

(2) 100ms retentive timer

1) The 100ms retentive timer times the enabled time of timer coil. For this reason, if the coil disables, the present value and contact ON/OFF state are retained. When the coil enables again, timing is resumed beginning with the retained present value.

2) Use RST T[ ]instruction to clear the present value and disable the contact.

3) Set the set value in decimal. The setting range is 1 to 32767 (0.1 to 3276.7 sec).

REMARKS

1) For timer processing method, refer to the following:

a) For timer processing method, refer to Section 2.5.4.
b) For timer accuracy, refer to Section 2.5.5.

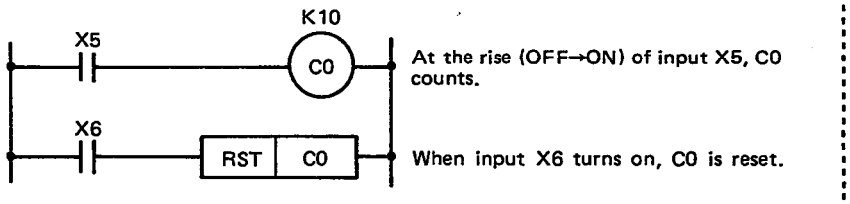
2.2.7 Counter C

1) The counters are up-counters. When the counter value reaches the set value, the counter counts up.

2) The counter performs counting after detecting the rise (OFF to ON) of coil. Therefore, if the coil remains on, counting is not performed.

3) Even if the coil turns off, the count value of counter is not cleared. It is required to clear the count value and turn off the contact.

EXAMPLE



4) Specify the set value in decimal within the range 1 to 32767.

REMARKS

1) For counter processing method, refer to the following:

a) For counter processing method, refer to Section 2.5.4.
b) For counter accuracy, refer to Section 2.5.6.

2.2.8 Data register D

1) The data register is a memory which stores data inside the PC.

2) Data registers consist of 16 bits and allow read and write operations requiring 16 bits.

3) When 32-bit data is handled, two registers are used. The data register number specified by the 32-bit instruction contains the lower 16 bits and the specified data register number + 1 contains the upper 16 bits.

4) The data stored by the sequence program is retained until other data is stored.

5) The data stored in the data register is cleared when:

   a) the PC power is turned on;
   b) reset is performed; or
   c) latch clear is made.

2.2.9 Index registers Z, V

1) The index registers are used to indirectly specify word device (T, C, D, W) numbers. For details of indirect device specification using the index registers, refer to Section 4.2.

2) The index register can be used for the sequence program like the data registers.

3) The index register is 1 point and consists of 16 bits. Write and read operations can be performed per 16 bits.

4) There are 2 points (Z, V) of index registers. In a 32-bit instruction, Z is lower 16 bits and V is upper 16 bits. Therefore, V cannot be specified by the 32-bit instruction.

5) The data of index register is cleared when:

   a) PC power is turned on;
   b) reset is performed; or
   c) latch clear is performed.

2.2.10 Link register W

1) The link register is a data register for data link.

2) When the link register is used for data link, data stored at 1 station (master or local station) can be read at other station (master or local station). Therefore, this link register allows communication from the master station to the local station, from the local station to the master station, and between the local stations.

3) The link registers within the range, which has not been set in the link initial data setting of parameter setting by use of the peripheral equipment (A6GPP), can be used as data registers.

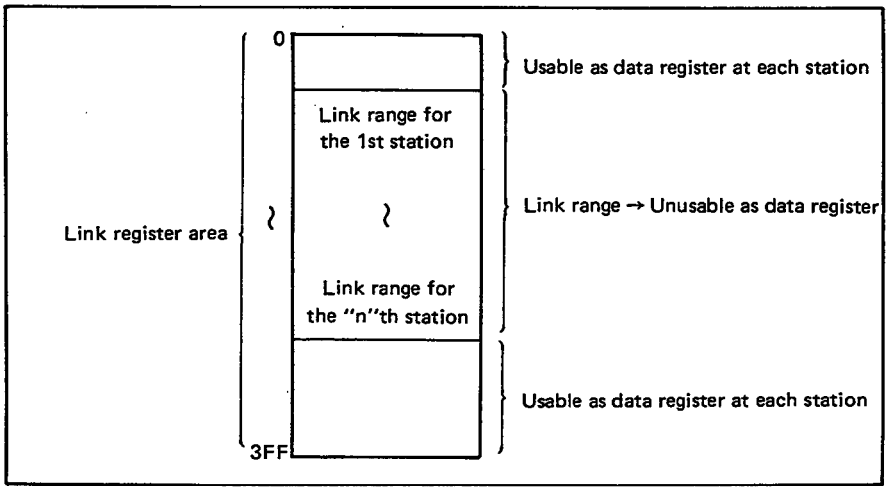


Fig. 2.4 Link Register Assignment

4) Link registers consist of 16 bits and allow read and write operations requiring 16 bits.
When 32-bit data is handled, two registers are used. The link register number specified by the 32-bit instruction contains the lower 16 bits and the specified link register number + 1 contains the upper 16 bits.

5) Data stored with sequence program is retained until other data is stored.

6) The data stored in the link register is cleared when:

   a) the PC power is turned on;
   b) reset is performed; or
   c) latch clear is performed.

POINT

(1) The link register may be used as a device only for the A0J2 for link (A0J2P23, A0J2R23). It cannot be used as a device for the A0J2CPU.

2.2.11 Nesting

1) The nesting of master control is indicated.

2) For MCR instruction, numbers beginning with the specified N is reset. Therefore, to perform nesting with master control, specify the N numbers as described below.

   a) MC instruction: Specify in order of lower N numbers.
   b) MCR instruction: Specify in order of higher N numbers.

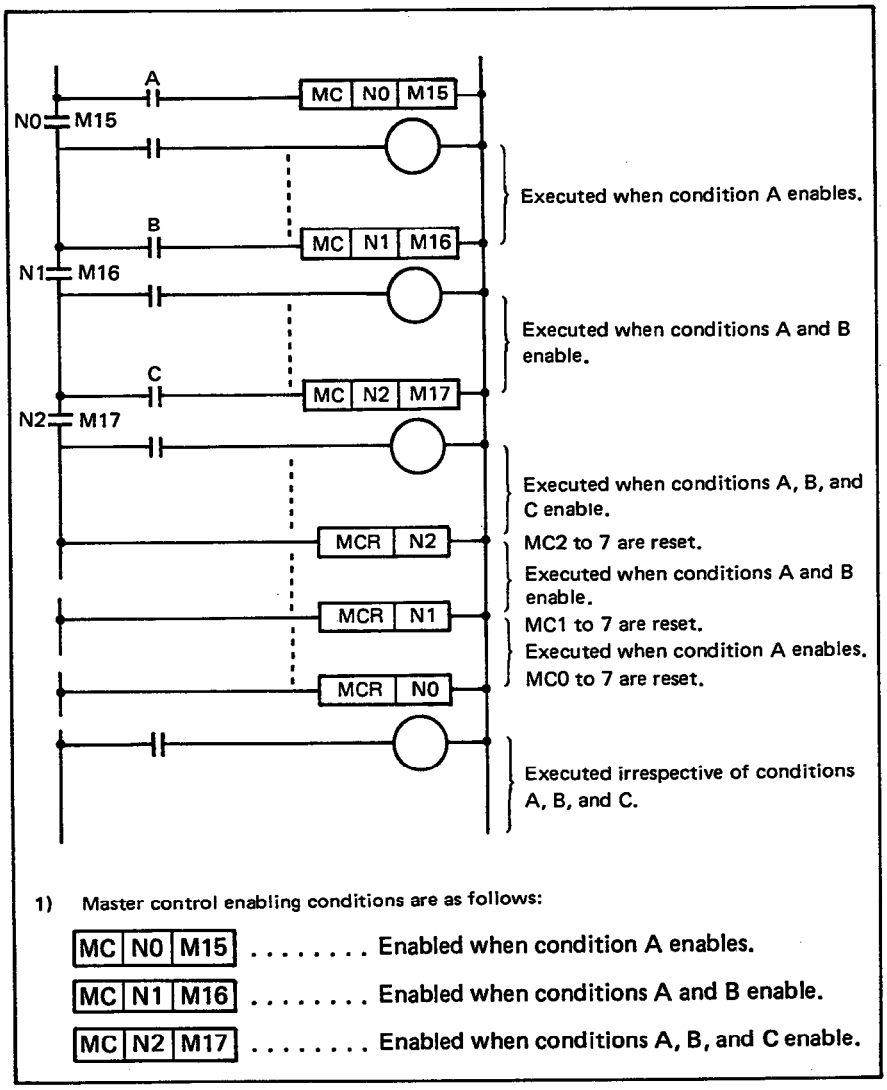


Fig. 2.5 Master Control Nesting Concept

REMARKS

1) For details of MC and MCR instructions, refer to Section 5.1.

2.2.12 Pointer P

1) The pointer P is used in the following 2 methods:

   a) Used as a device for branch instruction (CJ, CALL) and indicates the destination of the branch instruction.

   b) Used at the destination head of branch instruction.

2) The same pointer number may be used several times as a branch instruction device but can be used as a label only once.

3) P63 always indicates END. Pointers P0 to P62 cannot be used as END instruction labels.

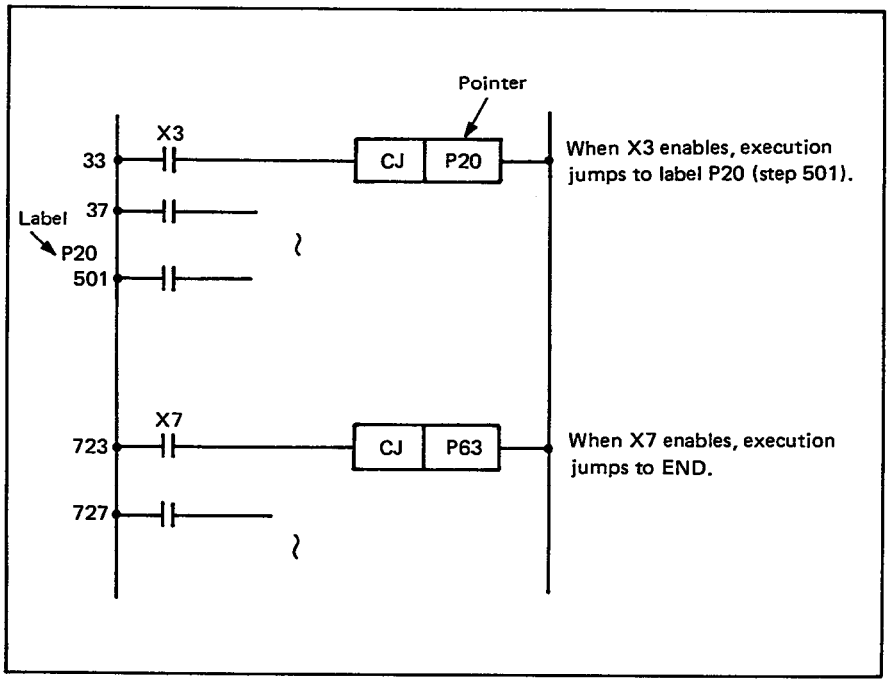


Fig. 2.6 Pointer Concept

2.2.13 Decimal constant K

1) The decimal constant is used as indicated below.

		Applicable Range
Timer, counter set value		1 to 32767
Pointer number		0 to 63
Bit device digit specification		1 to 4
Basic and application instruction value specification	16-bit instruction	−32768 to 32767
	32-bit instruction	−2147483648 to 2147483647

2) The decimal constant is stored into the PC in BIN (binary).

2.2.14 Hexadecimal constant H

1) The hexadecimal constant is used for specifying the numeric values of basic and application instructions in the following ranges:

- 16-bit instruction: 0 to FFFF
- 32-bit instruction: 0 to FFFFFFFF

REMARKS

1) The hexadecimal constants are expressed in 0 to 9 and A to F, i.e. 0, 1, 2 .....8, 9, A ..... E, F, which is followed by 10 (a carry occurs). The relationship between the decimal and hexadecimal constants is as shown below.

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10

2.3 Latch Range Setting and Latch Clear

(1) The A0J2 has a latch function. The latch function retains device data within the latch range unless latch clear is performed (including power-off, reset, STOP).
Devices which can be latched are M, B, T, C, and D (W). Set the latch range with the GPP/PHP/HGP or A7PU. For details of the operating procedure, refer to the GPP/PHP/HGP or A7PU Operating Manual. The latch ranges are as shown in the following table. In the A0J2 initial setting, the latter half is latched.

	Non-Latch Range	Latch Range
Unlatched	M0 to M2047 T0 to T127 C0 to C127 D0 to D511 B0 to B3FF (W0 to 3FF)	———
Latter half latched	M0 to M1023 T0 to T39/T80 to T99/ T120 to T123 C0 to C63 D0 to D255 B0 to B1FF (W0 to 1FF)	L1024 to L2047 T40 to T79/T100 to T119/ T124 to T127 C64 to C127 D256 to D511 B200 to B3FF (W200 to 3FF)
Whole range latched	———	L0 to L2047 T0 to T127 C0 to C127 D0 to D511 B0 to B3FF (W0 to 3FF)

POINT

(1) The device W is available only for the A0J2CPUP23(R23).

(2) Latch clear procedure

1) Latch clear initializes data in the latch range and non-latch range from outside the unit. When latch clear is performed, the states of devices within the latch ranges change as described below:

a) Y, M/L, F, B: Turn off.
b) Special M (9000 to 9255): Retained.
c) T, C: Contact and coil turn off. Present value is set to 0.
d) D, Z, V: Contents are cleared to 0.
e) Special D (9000 to 9127): Retained.

2) Perform latch clear with the RUN key switch in the following procedure:

a) Move the RUN key switch from "STOP" position to "L.CLR" 3 times.

b) When "RUN LED" flickers, latch clear is ready.

c) After "RUN LED" flickers, move the RUN key switch from "STOP" to "L.CLR". This completes latch clear.

POINT

(1) If the RUN key switch is moved to "RUN" or "RESET" position during latch clear operation, latch clear operation is reset. In this case, the A0J2 state changes as described below.

RUN position:	A0J2 changes from STOP to RUN and continues operation.
RESET position:	Reset is performed.

2.4 Concept of I/O Number Assignment

I/O number assignment is one of the requirements for system configuration. If a wrong assignment is made, a system failure will occur. This section explains the concept of I/O number assignment in the A0J2 system.

2.4.1 A0J2 I/O unit

1) The I/O units and extension base unit slots each occupy 64 points. Inputs (X) and outputs (Y) are assigned as given below.

   a) Input (X) . . . . . .First 32 points
   b) Output (Y) . . . . . .Second 32 points

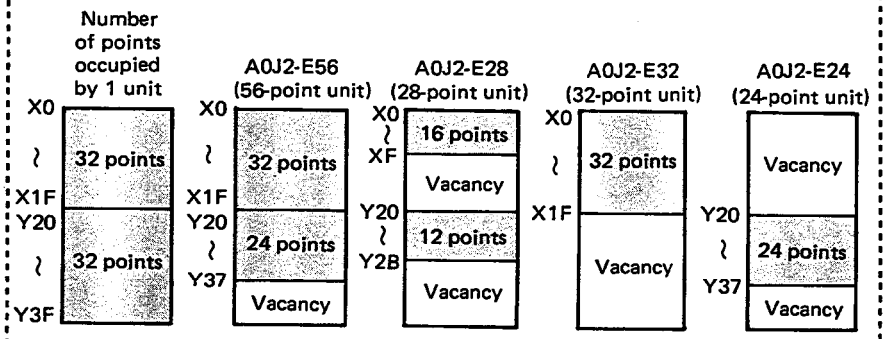
2) I/O numbers are assigned in order of I/O unit setting numbers.

3) The head I/O numbers of an I/O unit are predetermined as given below depending on I/O unit setting numbers.

Setting	0	1	2	3	4	5	6	7
Input head number	X00	X40	X80	XC0	X100	X140	X180	X1C0
Output head number	Y20	Y60	YA0	YE0	Y120	Y160	Y1A0	Y1E0

Example

When I/O unit setting number is 0, the I/O numbers of I/O units are as follows.



4) Output numbers relevant to input numbers and the unused numbers of each I/O unit can be used as internal memories (M). The above numbers of an input dedicated unit cannot be used as M.

2.4.2 A0J2 special function unit

1) A0J2 special function units each occupy 64 points. Both inputs (X) and outputs (Y) use the first 32 points. In this case, I/O numbers available to the user are predetermined depending on special function units. For available I/O numbers and applications, refer to relevant special function unit user's manual.

2) The head I/O numbers of a special function unit are predetermined as given below depending on special function unit setting numbers.

Setting	0	1	2	3	4	5	6	7
Input head number	X00	X40	X80	XC0	X100	X140	X180	X1C0
Output head number	Y00	Y40	Y80	YC0	Y100	Y140	Y180	Y1C0

2.4.3 Extension base unit

1) In an extension base unit, slot 0 to 3 can be used.

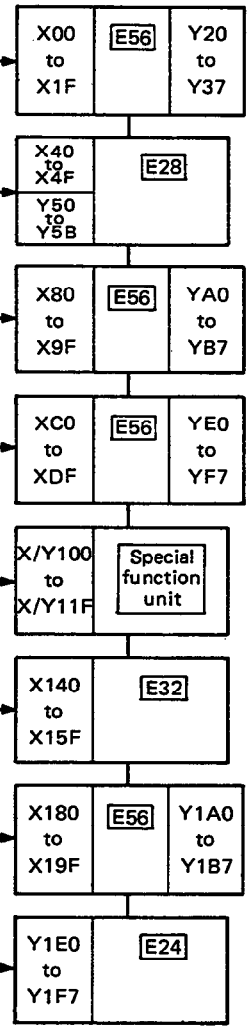
2) Always locate the extension base unit at the final stage of a system. I/O numbers of slot 0 always start at X/Y 100.

3) Each slot of the extension base unit occupies 64 points irrespective of I/O unit points and vacant slots.

4) If an input unit is loaded in the extension base unit, output numbers relevant to input numbers (all 64 points) cannot be used as internal memories.

I/O number assignment in A0J2CPU system

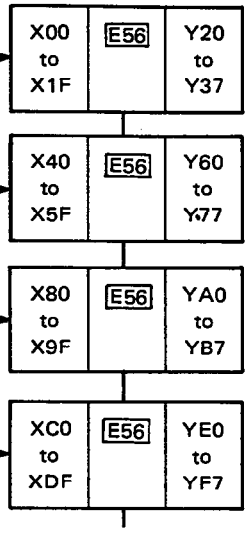
I/O unit No.	I/O number
0	X00 to X1F Y20 to Y3F
1	X40 to X5F Y60 to Y7F
2	X80 to X9F YA0 to YBF
3	XC0 to XDF YE0 to YFF
4	X100 to X11F Y120 to Y13F
5	X140 to X15F Y160 to Y17F
6	X180 to X19F Y1A0 to Y1BF
7	X1C0 to X1DF Y1E0 to Y1FF



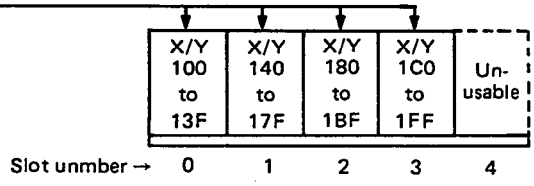
1) 1 unit occupies 64 I/O points.
2) I/O numbers are determined by I/O unit numbers (0 to 7). (For I/O unit number setting, refer to Section 3.2.2 of User's Manual (I/O Unit Edition).)
3) A maximum of 8 I/O units may be connected.
4) 1 unit occupies 64 I/O points. Therefore, for example, when 24 point unit is used, 40 points are not used. In this case, if I/O unit is changed, it is not necessary to change the presently used I/O numbers.
5) Output numbers relevant to input numbers and unused numbers (second 32 points for a special unit) can be used as internal memories (M). For an input dedicated unit, the above mentioned numbers cannot be used as M.
6) Exercise care so that the same I/O unit numbers are not used in a system. If the same numbers are used, the same I/O numbers will exist, resulting in failure.
7) I/O unit numbers are 0 to 7. These do not have to be serial numbers. This is because I/O numbers are determined by I/O unit numbers as shown in the right side figure. However, it is recommended to set them in serial numbers in consideration of the I/O number concept.

I/O number assignment in extension base unit combined system

I/O unit No.	I/O number
0	X00 to X1F Y20 to Y3F
1	X40 to X5F Y60 to Y7F
2	X80 to X9F YA0 to YBF
3	XC0 to XDF YE0 to YFF
Slot number in extension base	I/O number
0	X100 to X13F Y100 to Y13F
1	X140 to X17F Y140 to Y17F
2	X180 to X1BF Y180 to Y1BF
3	X1C0 to X1FF Y1C0 to Y1FF

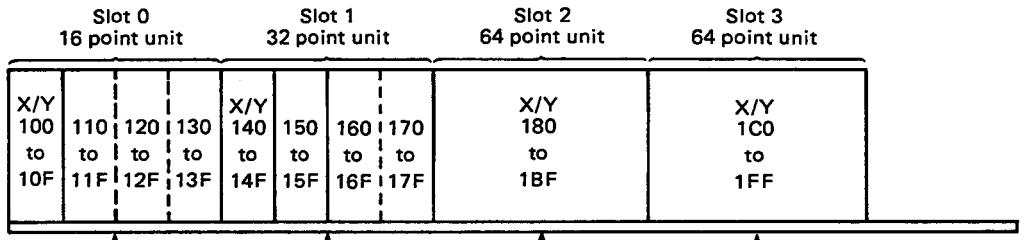


1) 1 unit (this also applies to I/O unit in extension base) occupies 64 I/O points.
2) In extension base unit combined system, a maximum of four A0J2 I/O units may be connected.
3) Extension base unit should always be located at the final stage of system and its I/O number should begin with X/Y100.
4) Output number corresponding to input number and unused number may be used as internal memory (M). However, these numbers cannot be used for input dedicated unit.
5) In extension base unit combined system, a maximum of four A0J2 I/O units may be connected. In this case, set I/O unit numbers within 0 to 3.



Restrictions on extension base unit

1) In extension base unit, only slots 0 to 3 are usable.
2) Irrespective of used I/O units, 1 slot occupies 64 points. (Vacant slot also occupies 64 points.)
   Output number corresponding to input unit cannot be used as internal memory (M).
3) When 64 point I/O unit is not used (e.g. 16 or 32 point unit), take care of the following points:



When 64 point unit is used, there is no restriction.

When 32 point unit is used, the latter half 32 points are not used. Actually, however, the first half 32 points and the latter half 32 points overlap with each other and CPU makes access. Therefore, if the latter half I/O numbers are accidentally used in a program, the overlapping first half I/O number area is accessed actually.
For example, 160 overlaps with 140 and 170 with 150.

When 16 point unit is used, the latter 48 points are not used. Actually, however, four 16 point units overlap with each other. For example, if any of 110, 120, and 130 is accessed, 100 is accessed actually. Be sure to use actual I/O numbers in a program. In the example shown on the left, Y100 to 10F are actual I/O numbers.

2.5 Operation Processing Method

This section describes the operation processing method, I/O processing, scan time, etc. of the PC.

2.5.1 Operation processing method

The operation processing method (control system) of the PC is a stored program and repetitive operation. This section explains this stored program and repetitive operation.

(1) Stored program

The PC stores a sequence program in the memory in advance, and at the execution time of operation, controls operation with the sequence program. As described above, the stored program is a system which stores a program required for control in the PC memory in advance.

(2) Repetitive operation

The PC reads the sequence program stored in the internal memory in due order, starting at step 0, and performs operation up to the END (FEND) instruction. When the operation is performed up to the END (FEND) instruction, internal processing, such as the processing of timers/counters and self-diagnostic check, are performed and the execution returns to step 0 again and operation is initiated from step 0. As described above, the repetitive operation is a system which causes the PC to repeat the execution of program from the instruction at step 0 to END (FEND).

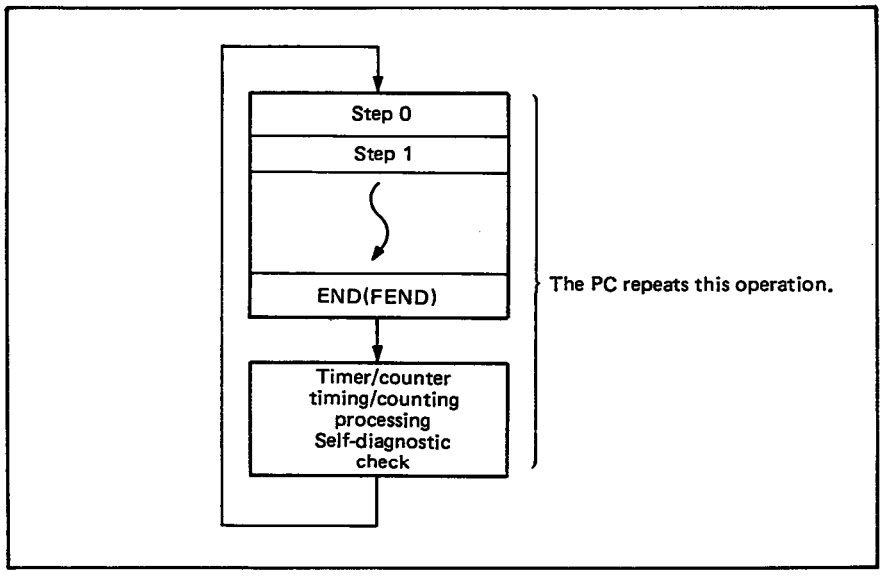


Fig. 2.7 PC Operation Processing Procedure

REMARKS

Step 0 to the next step 0 or the END (FEND) instruction to the next END (FEND) instruction is referred to as "1 scan". Therefore, 1 scan of the PC is a total of the sequence program created by user (step 0 to END instruction) and the internal processing of the PC.

2.5.2 I/O processing method

The I/O processing of the A0J2 is a direct method.
In this direct method, the change of the input unit is drawn into the input data memory of CPU unit every time, and at the execution time of operation, the data of this input data memory is used as input data. The operation result of sequence program is output from the output data memory to the output unit each time.
The change of output unit delays a maximum of one scan with respect to the change of input unit, as shown in Fig. 2.8.

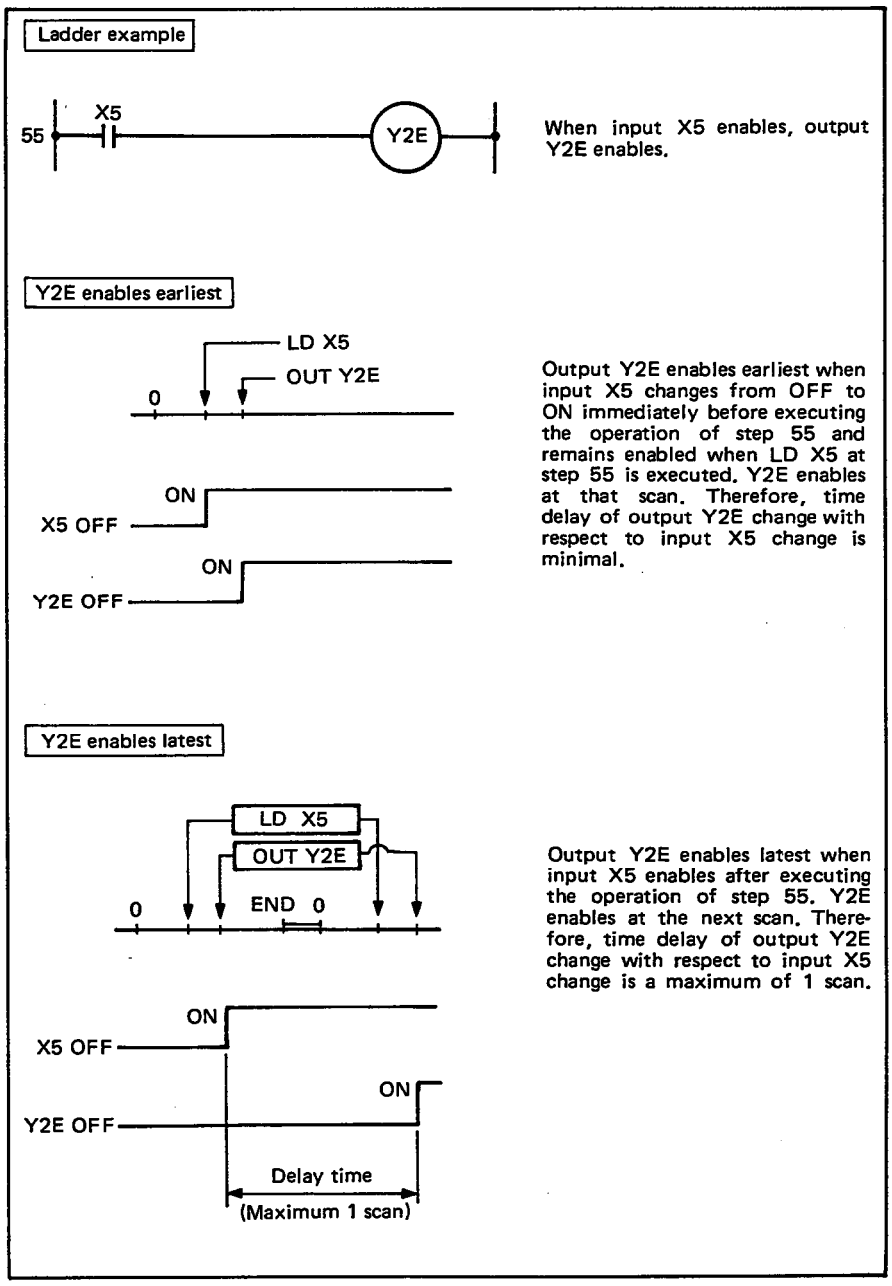


Fig. 2.8 Output Y Change With Respect To Input X Change

2.5.3 Watch dog timer (WDT)

The watch dog timer is a timer inside the PC for detecting the errors of PC hardware and program, and is preset to 200ms.
The PC resets this watch dog timer after the execution of END (FEND) instruction. Therefore, when the PC operates properly and executes the END (FEND) instruction within 200ms, the watch dog timer does not time out.

However, when the END (FEND) instruction cannot be executed within 200ms due to the hardware error of the PC or because the scan time is too long, the watch dog timer times out. When the watch dog timer times out, the watch dog timer error occurs and the operation of PC changes as described below:

a) The PC stops the execution of operation and turns off all outputs.

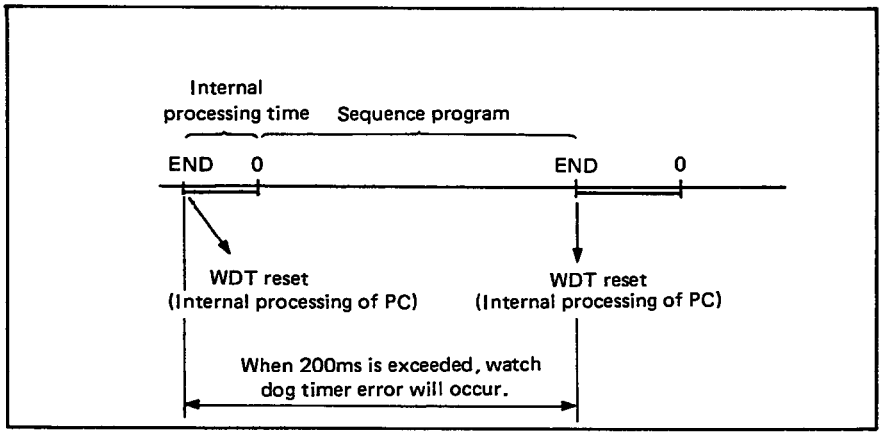


Fig. 2.9 Watch Dog Timer Reset

If scan time exceeds 200ms, do as described below.

a) Change the program to set the sequence program operation time to 200ms or less.

b) Using WDT instruction, reset the watch dog timer in the sequence program.

2.5.4 Timer and counter processing methods

The timers and counters are processed at the execution time of OUT T □ and OUT C □ instructions, respectively, and after the execution of END (FEND) instruction.

(1) At OUT T□, OUT C□ instruction execution time

At the execution of OUT T□and OUT C□ instructions, the following processings are performed with ON/OFF of the input condition.

a) When the input condition has enabled, timer and counter coils are turned on.

b) When the input condition has disabled, timer and counter coils are turned off. In this case, the present values of the timer and counter and the ON/OFF states of the contacts remain unchanged.

(2) After END (FEND) instruction execution

When the END (FEND) instruction is executed, the present values of timers and counters are updated and the contacts are turned on/off as described below.

Classification	Processing after END (FEND) Instruction Execution
100ms timer 10ms timer	While coil is on, present value is updated. When timer times out, contact turns on. When coil turns off, present value is set to 0 and contact turns off.
100ms retentive timer	While coil is on, present value is updated. When timer times out, contact turns on. When coil is off, present value and contact ON/OFF state are retained.
Counter	When coil changes from OFF to ON, counting is started. When counter counts out, contact turns on. While coil is on or off, no processing is performed.

REMARKS

The 100ms retentive timer and counter use RST instruction to clear the present value (to 0) and change the contact from ON to OFF.

2.5.5 Timer accuracy

There are the following errors until the timer times out. (Refer to Fig. 2.10.)

a) Timer timing error: +1 scan time
b) Error due to timer input condition enabling timing: +1 scan time
c) Error due to OUT T□ instruction position in program: −1 scan time

Therefore, the timer may cause a maximum of $\frac{+2}{-1}$ scan time error from when the input condition enables to when the timer times out and the contact turns on.

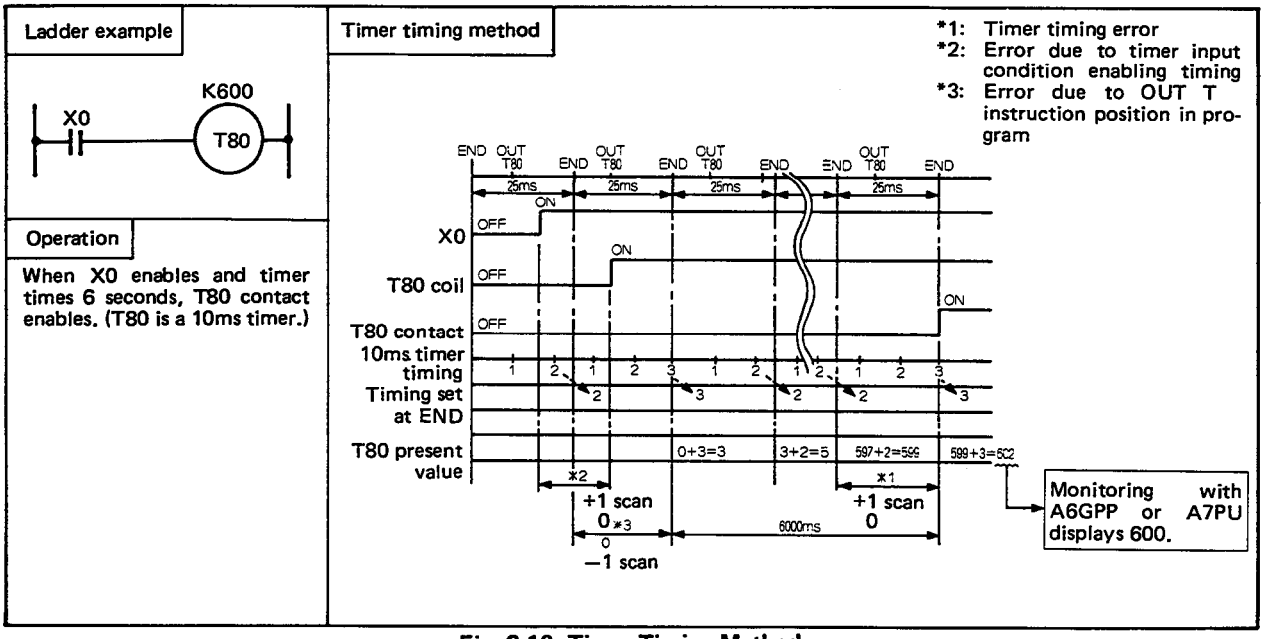


Fig. 2.10 Timer Timing Method

2.5.6 Maximum counting speed of counter

1) The maximum counting speed of counter is determined by the scan time. The counter counts only when the ON/OFF time of input condition is greater than the scan time.

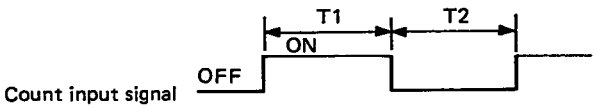
2) The maximum counting speed of counter is obtained by the following expression:

$$\text{Maximum counting speed } Cmax = \frac{10n}{ts} \text{ [times/sec]}$$

where n = duty (For details, refer to SUPPLEMENT.)
ts = scan time (ms)

SUPPLEMENT

The duty represents the ratio of ON time to OFF time of count input signal in terms of percent (%).



Duty n calculation expression

If $T1 \leqq T2$ $\quad n = \frac{T1}{T1+T2} \times 100\ [\%]$

If $T1 > T2$ $\quad n = \frac{T2}{T1+T2} \times 100\ [\%]$

3. GENERAL DESCRIPTION FOR PROGRAMS

3.1 Program

1) A program allows the user to combine instructions available for the A0J2 and perform intended operation. Therefore, if the user does not create a program, the A0J2 cannot be operated.

2) This program creation is called programming. Perform the programming of the A0J2 with a peripheral unit.

3) For programming, ladder mode and list mode are available.

   a) Ladder mode: Creates ladders on a 1 circuit block basis using sequence diagram symbols. The peripheral converts the created ladder diagram into list mode.

   b) List mode: Directly programs instructions.

REMARKS

1) One circuit block starts with a contact instruction and ends with an instruction equivalent to the coil.

4) The program format is as shown below. (The same program shown in ladder and list modes)

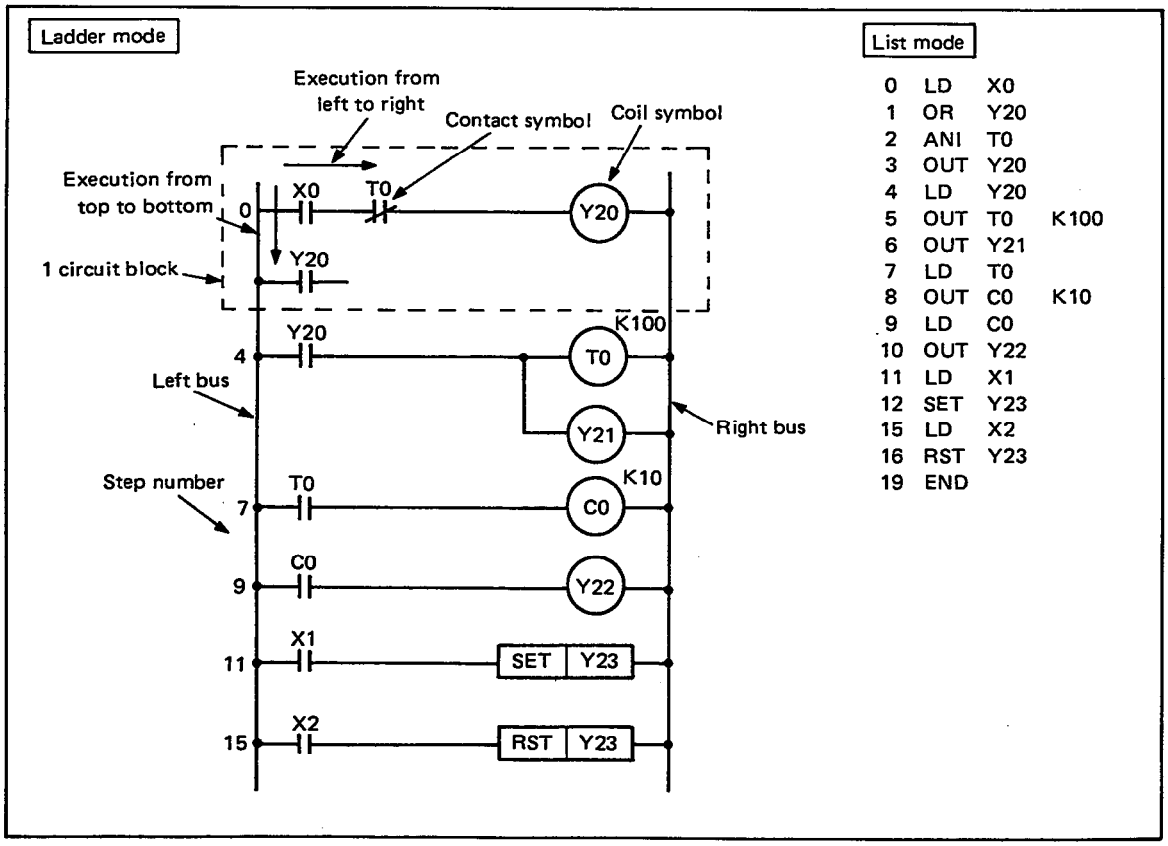


Fig. 3.1 Program Format

5) Program execution sequence is as follows:

a) In ladder mode, execution is made from step 0 to END instruction in the following order.

- Program is executed per 1 circuit block, beginning with the contact instruction on the left bus to the instruction equivalent to the coil on the right bus.
- In 1 circuit block, execution is made from left to right and from top to bottom.
- After completion of 1 circuit block execution, the next circuit block is executed.

b) In list mode, execution is made from step 0 to END (FEND) instruction in the created order. Therefore, if the program is not executed in order of instruction execution in list mode, error will occur.

3.2 Program Write

1) Write the program created with the peripheral equipment to the A0J2 memory. (For the writing procedure, refer to the Operating Manual of the peripheral equipment.)

2) After writing the program to the A0J2 memory, setting RUN key switch on the CPU unit to RUN starts operation. During the operation, if:

a) data used for the instruction is defective, processing is not performed and operation continued; or

b) the program is faulty, operation stops and RUN LED flickers.

3.3 Program Configurations

1) In the user memory area, programs may be written in the configurations shown in Fig. 3.2.

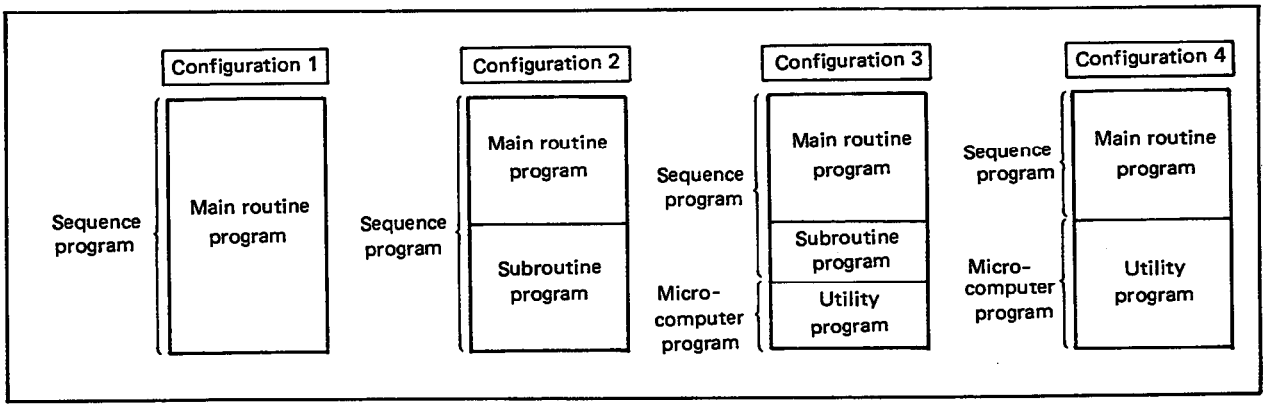


Fig. 3.2 Program Configurations

2) The program configurations are as described below:

   a) The sequence programs use the instructions for the A0J2. The main routine program is always executed during RUN. The subroutine program contains programs which are desired to be executed several time during 1 scan.

   b) The microcomputer program cannot be created by the user. Only utility FD data may be written.

3) Before writing the utility program, it is necessary to set the microcomputer program area with the A6GPP. For details of the microcomputer program area setting procedure, refer to the Operating Manual for peripheral equipment.

4. GENERAL DESCRIPTION FOR INSTRUCTIONS

4.1 Instruction Configurations

4.1.1 Instruction configurations

1) The A0J2 instruction can be classified into the instruction part and the device. Their applications are as described below:

Instruction part:	Indicates the function of the instruction.
Device:	Indicates data used for the instruction.

2) Depending on the instruction part and device combinations, the instruction configurations can be largely divided as follows:

a) [Instruction part] :

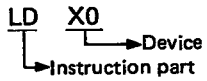
Instruction which does not change the device status. Mainly controls the program.

Example END, FEND

b) [Instruction part] + [device] :

Instruction which turns on/off the device.

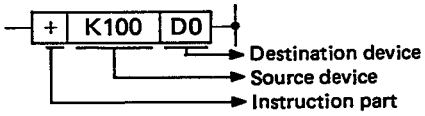
Example



c) [Instruction part] + [source device] + [destination device] :

Performs operation with data at the destination and data at the source, and stores operation result to the destination.

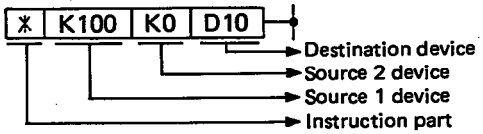
Example



d) [Instruction part] + [source 1 device] + [source 2 device] + [destination device] :

Performs operation with data at the source 1 and data at the source 2, and stores operation result to the destination.

Example



e) Others:

Combinations other than the above a) to d).

REMARKS

1) In this manual, the sources and destination are represented by the following characters:

Source:	S
Source 1:	S1
Source 2:	S2
Destination:	D

4.1.2 Source, destination

(1) Source (S)

1) The source is the data used for operation.

2) Specify as described below depending on the specified device.

- Constant:
  Specify the value used for operation. Set during program creation, this value is fixed and cannot be changed in the program.

- Bit device, word device:
  Specify the device which stores the data used for operation. Therefore, it is necessary to store the data in the specified device until the operation is executed. By changing the data to be stored into the specified device during program execution, the data used for that instruction can be changed.

(2) Destination (D)

1) Data is stored into the destination after operation. If the instruction is [instruction part] + [source device] + [destination device] combination, it is necessary to store the data used for operation in the destination prior to the operation.

2) Since the data is stored in the destination after operation, the constant (decimal or hexadecimal constant) cannot be specified.

4.1.3 16-bit, 24/32-bit processing instructions

1) The basis of the A0J2's basic and application instructions is a 16-bit processing instruction. However, 24/32-bit data processing may be performed for the following instructions.

	16 bits	24 bits	32 bits
BIN ↔ BCD conversion	O	O	X
BCD addition/subtraction	O	O	X
Instruction for special unit	O	X	O

2) Even for a 24-bit data processing instruction, the device occupies 32 bits (2 word devices). For BCD data, the upper 8 bits of 32 bits are as described below:

a) Source: The upper 8 bits are ignored.
b) Destination: The upper 8 bits are reset to 0.

For BIN data, 0 to 999999 are valid.

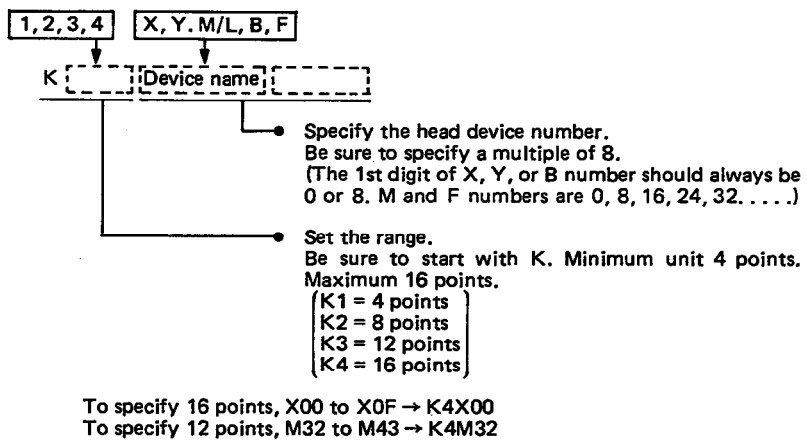
3) By adding D to the head of the 16-bit instruction function symbol, the 16-bit instruction changes to a 24/32-bit processing instruction.

	16-bit Instruction	24-bit Instruction	32-bit Instruction
BIN → BCD conversion	BCD	DBCD	———
BCD → BIN conversion	BIN	DBIN	———
BCD addition	———	DB+	———
BCD subtraction	———	DB–	———
Write to special unit	TO	———	DTO
Read from special unit	FROM	———	DFRO

4.1.4 Bit device digit specification

When the basic instructions and application instructions are used, digit specification may sometimes be required for bit devices (X, Y, M/L, B, F). This digit specification is used to specify the number of used points of the bit device. Specify the digit in units of four points to a maximum of 16 points. Specify as explained below.

Digit specifying method



To specify 16 points, X00 to X0F → K4X00
To specify 12 points, M32 to M43 → K4M32

4.2 Index Qualification

1) The index qualification is used to indirectly specify the device number used for the basic and application instructions.

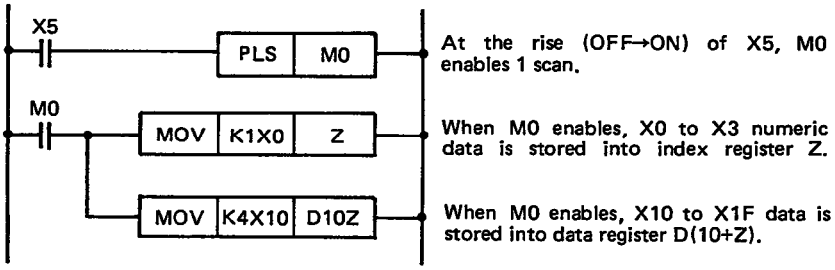
2) To perform the index qualification, add the index register (Z, V) to the device.

3) When the index qualification is performed, the actual processing devices are as shown below.

[Processing device number] = (specified device number) + (index register data)

Example

1) When X5 enables, this program stores data in X10 to X1F into the data registers (D10 to D17) specified with X0 to X3.



2) In the above ladder diagram, D10Z is as follows:

D10Z =D(10+Z)
=D(10+3)
=D13

Therefore, X10 to X1F data is stored into D13.

4) The index qualification is allowed only for the timer (T), counter (C), data register (D), and link register (W).

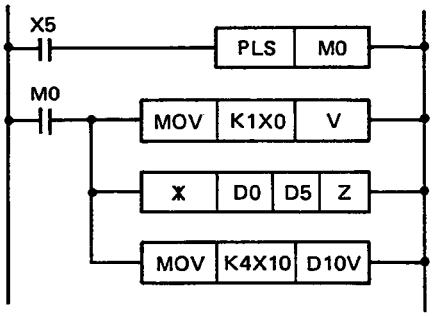
5) For index qualification, Z and V are used independently. Therefore, the index qualification is enabled in 2 types, Z and V.

6) The index register is capable of storing values, –32768 to 32767. For index qualification, however, exercise care so that the processing device numbers do not exceed the following values. If the processing device number exceeds the following value, other device data may be rewritten or processing may not be performed due to operation error.

   a) Timer: T0 to 127
   b) Counter: C0 to 127
   c) Data register: D0 to 511

POINT

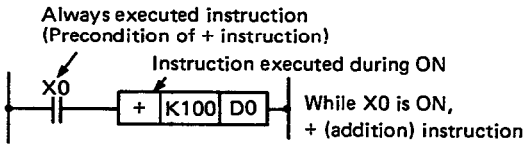
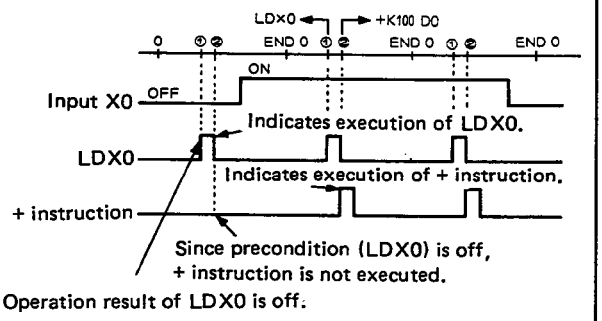
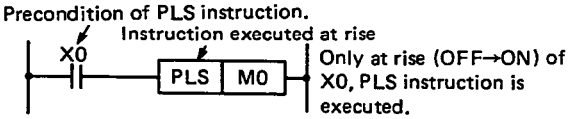
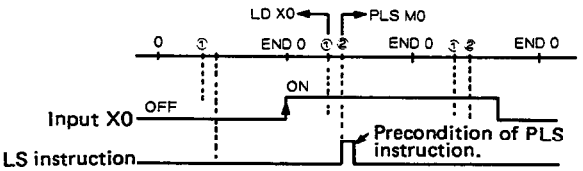
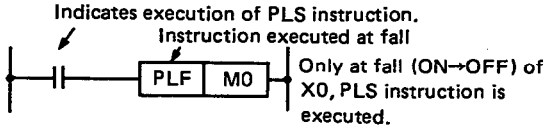
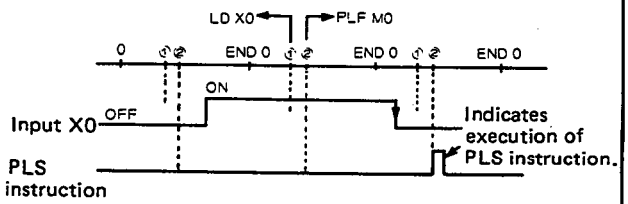
1) If Z is used for the instruction which employs 24/32 bits and the index qualification performed with the index register V, V value changes. Therefore, caution should be exercised.



4.3 Execution Conditions

1) The execution condition indicates the instruction execution timing. Therefore, each instruction is executed only when the execution condition enables.

2) The execution conditions are available in 4 types as shown in Table 4.1.

Classification	Symbol	Execution Conditions	Remarks
Always executed		Instruction which is always executed. There are the following 2 types: 1) When preceding condition is on, operation result is enabled. 2) When preceding condition is off, operation result is disabled.	Ladder example  Operation timing 
Executed during ON		(1) Instruction which is executed per scan while previous condition is on. (2) When previous condition is off, that instruction is not executed and not processed.	
Executed at rise time		(1) Instruction which is executed only when preceding condition changes from OFF to ON. (2) While preceding condition is off or on, this instruction is not executed and not processed. (3) The instruction executed at rise time is only the PLS instruction.	Ladder example  
Executed at fall time		(1) Instruction which is executed only when preceding condition changes from ON to OFF. (2) While preceding condition is off or on, this instruction is not executed and not processed. (3) The instruction executed at fall time is only the PLF instruction.	Ladder example  Operation timing 

4.4 Error Processing

4.4.1 Operation error

1) Operation error occurs when each instruction format is proper but data used for the instruction is outside the applicable range. For example, data specified with the BCD conversion instruction is greater than 9999 (999999). If the specified device number is outside the device range usable for the A0J2, operation error occurs.

2) If operation error has occurred:

   a) instruction in operation error is not executed and not processed.
   b) error flag (M9011) enables.
   c) the step number of the first instruction in operation error is stored into the error step storage register (D9011).
   d) error code (50) is stored into D9008 in BIN.

POINT

(1) The error flag does not disable if operation error is remedied. Therefore, user must disable the error flag.

(2) The error step number is stored into the error step storage register when the error flag changes from OFF to ON. Therefore, if the error flag remains on, data in the error step storage register remains unchanged.

4.4.2 Grammatical error

1) The grammatical error occurs when the instruction format is not proper. For example, END instruction does not exist in the program.

2) When grammatical error has occurred:

   a) the A0J2 stops operation and turns off all outputs; and
   b) error code is stored into D9008 in BIN. (For details of error codes, refer to Section 7.2.)

4.4.3 Flags

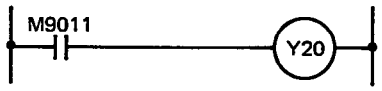
1) The A0J2 flags are available in the following 2 types:

a) Error flag (M9011)

- Enables when operation error has occurred.
- Once enabled, remains enabled until reset by user.
- This error flag allows checking for operation error outside the unit.

Example

When operation error has occurred, this program enables Y20.



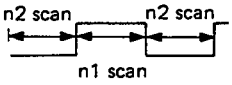
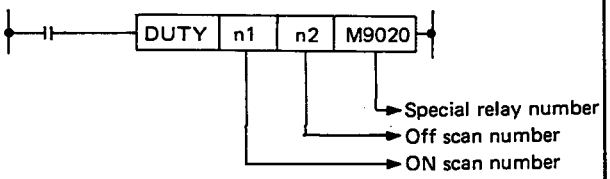
When error occurs, error flag (M9011) turns on and Y20 enables. Therefore, Y20 allows checking of operation error in the field.

b) Carry flag (M9012)

- The carry flag stores the shift result of the bit shift (BSFL, BSFR) application instruction.
- The carry flag is commonly used for BSFL and BSFR instructions. Therefore, when the carry flag is utilized for operation using several shift instructions, caution should be exercised.

4.5 Special Relay List

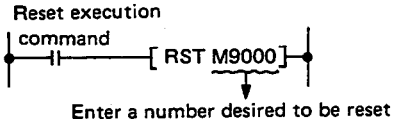
The special relay is an internal relay of which application has already been determined. Therefore, the special relay cannot be turned on or off in the program. Use the special relay as a contact in the program. However, the special relays marked * may be reset (turned off). The special relay list is shown below.

Number	Name	Description	Details
*M9000	Fuse blown	OFF:Normal ON: Presence of fuse blow unit	•Turned on when there is one or more output units of which fuse has been blown. Remains on if normal status is restored.
*M9005	AC DOWN detection	OFF:AC is good ON: AC is down	•Turned on if power failure of within 10ms occurs. Reset when POWER switch is moved from OFF to ON position.
*M9006	Battery low	OFF:Normal ON: Battery low	•Turned on when battery voltage reduces to less than specified. Turned off when battery voltage becomes normal.
*M9008	Self-diagnostic error	OFF:Absence of error ON: Presence of error	•Turned on when error is found as a result of self-diagnosis.
*M9009	Annunciator detection	OFF:Absence of detection ON: Presence of detection	•Turned on when OUT F or SET F instruction is executed. Remains on if RST F instruction is executed.
*M9011	Operation error flag	OFF:Absence of error ON: Presence of error	•Turned on when operation error occurs during execution of application instruction. Remains on if normal status is restored.
M9012	Carry flag	OFF:Carry off ON: Carry on	•Carry flag used in application instruction (BSFL, BSFR).
M9016	Data memory clear flag	OFF:No processing ON: Output clear	•Clears all data memory (except special relays and special registers) in remote run mode from computer, etc. when M9016 is 1.
M9017	Data memory clear flag	OFF:No processing ON: Output clear	•Clears all unlatched data memory (except special relays and special registers) in remote run mode from computer, etc. when M9017 is 1.
M9020	User timing clock No.0		•Relay which repeats on/off at intervals of predetermined scan. •When power is turned on or reset is performed, the clock starts with off. •Set the intervals of on/off by DUTY instruction. 
M9021	User timing clock No. 1		
M9036	Normally ON	ON OFF	•Used as dummy contacts of initialization and application instruction in sequence program. •M9036 and M9037 are turned on and off without regard to position of key switch on CPU. M9038 and M9039 are under the same condition as RUN status except when the key switch is at STOP position, and turned off and on.
M9037	Normally OFF	ON OFF	
M9039	RUN flag (off only for 1 scan after run)	ON OFF 1scan	•Off for 1 scan after the CPU front key switch is moved from STOP to RUN.
M9042	Stop status contact	OFF:During stop ON: Not during stop	

Note

(1) The above relays with numbers marked * remain "on" if normal status is restored. Therefore, to turn them "off", use the following method:

1) Method by user program
Insert the circuit shown at right into the program and turn on the reset execution command contact to clear the special relay M.



2) Method by peripheral equipment (A7PU, A6GPP)
Cause forced reset by the test function of peripheral equipment. For the operation procedure, refer to the manual of each peripheral equipment.

3) By moving the RESET key switch at the CPU front to the RESET position, the special relay is turned "off".

4.6 Special Register List

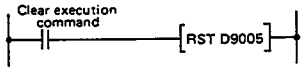
Special registers are data registers of which applications have already been determined. Therefore, do not write data to the special registers in the program. Read data and use it in the program. The special registers marked * may be reset (to 0).
Table 4.3 shows the special register list.

<table>
<tr><th>Number</th><th>Name</th><th>Stored Data</th><th>Explanation</th></tr>
<tr><td>D9001</td><td>Fuse blown detection</td><td>Stores the first unit number of which fuse has blown.</td><td>•The value stored into D9001 is the first fuse-blown unit number plus 1.
<table>
<tr><th colspan="2">A0J2 I/O Unit</th><th colspan="2">Extension Base Unit</th></tr>
<tr><th>Setting switch</th><th>Data stored</th><th>Base unit slot number</th><th>Data stored</th></tr>
<tr><td>0</td><td>1</td><td>0</td><td>5</td></tr>
<tr><td>1</td><td>2</td><td>1</td><td>6</td></tr>
<tr><td>2</td><td>3</td><td>2</td><td>7</td></tr>
<tr><td>3</td><td>4</td><td>3</td><td>8</td></tr>
<tr><td>4</td><td>5</td><td></td><td></td></tr>
<tr><td>5</td><td>6</td><td></td><td></td></tr>
<tr><td>6</td><td>7</td><td></td><td></td></tr>
<tr><td>7</td><td>8</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>D9003</td><td>SUM instruction detection quantity</td><td>Number of bits detected by SUM instruction</td><td>•Stores the number of detected bits in BIN code at execution of the SUM instruction and updates it per execution.</td></tr>
<tr><td>* D9005</td><td>AC DOWN counter</td><td>AC DOWN count</td><td>•1 is added each time input voltage becomes 80% or less of rating while the CPU unit is performing operation, and the value is stored in BIN code.</td></tr>
<tr><td>* D9008</td><td>Self-diagnostic error</td><td>Self-diagnostic error number</td><td>•When error is found as a result of self-diagnosis, error number is stored in BIN code.</td></tr>
<tr><td>D9009</td><td>Annunciator detection</td><td>F number at which external failure has occurred</td><td>•When one of F0 to 255 is turned on by [OUT F:::] or [SET F:::], the F number, which has been detected earliest among the enabled F numbers, is stored in BIN code.
•To clear D9009, execute [RST F:::] instruction. If another F number has been detected, the lowest F number among the enabled F numbers is stored into D9009.</td></tr>
<tr><td>D9011</td><td>Error step</td><td>Step number at which operation error has occurred</td><td>•When operation error has occurred during execution of application instruction, the step number, at which the error has occurred, is stored in BIN code. Since storage into D9011 is made when M9011 changes from off to on, the contents of D9010 cannot be renewed unless M9011 is cleared by user program.</td></tr>
<tr><td>D9016</td><td>ROM/RAM setting</td><td>0:ROM
1:RAM
2:EEP ROM</td><td>•Indicates the setting of memory select chip. One value of 0 to 2 is stored in BIN code.</td></tr>
<tr><td>D9017</td><td>Scan time</td><td>Minimum scan time (per 10ms)</td><td>•If scan time is smaller than the content of D9017, the value is newly stored at each END. Namely, the minimum value of scan time is stored into D9017 in BIN code.</td></tr>
<tr><td>D9018</td><td>Scan time</td><td>Scan time (per 10ms)</td><td>•Scan time is stored in BIN code at each END and always rewritten.</td></tr>
<tr><td>D9019</td><td>Scan time</td><td>Maximum scan time (per 10ms)</td><td>•If scan time is larger than the content of D9019, the value is newly stored at each END. Namely, the maximum value of scan time is stored into D9019 in BIN code.</td></tr>
</table>

POINT

(1) For the above special registers with numbers marked *, the contents of register are not cleared if normal status is restored. Therefore, to clear the contents, use the following method:

1) Method by user program
Insert the circuit shown at right into the program and turn on the clear execution command contact to clear the contents of register.



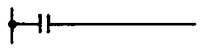
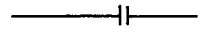
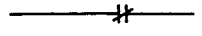
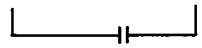
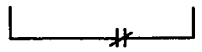
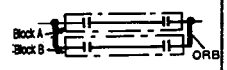
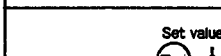
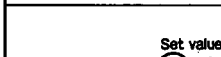
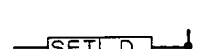
2) Method by peripheral equipment (A7PU, A6HGP, A6GPP)
Set the register to "0" by changing the present value by the test function of peripheral equipment (A7PU, A6HGP, A6GPP) or set to "0" by forced reset. For the operation procedure, refer to the Instruction Manual for A6GPP or A6HGP.

3) By moving the RESET key switch at the CPU front to the RESET position, the special register is set to "0".

5. SEQUENCE INSTRUCTIONS

1) The sequence instructions are used for relay control circuits such as contacts, coils, timers, and counters.

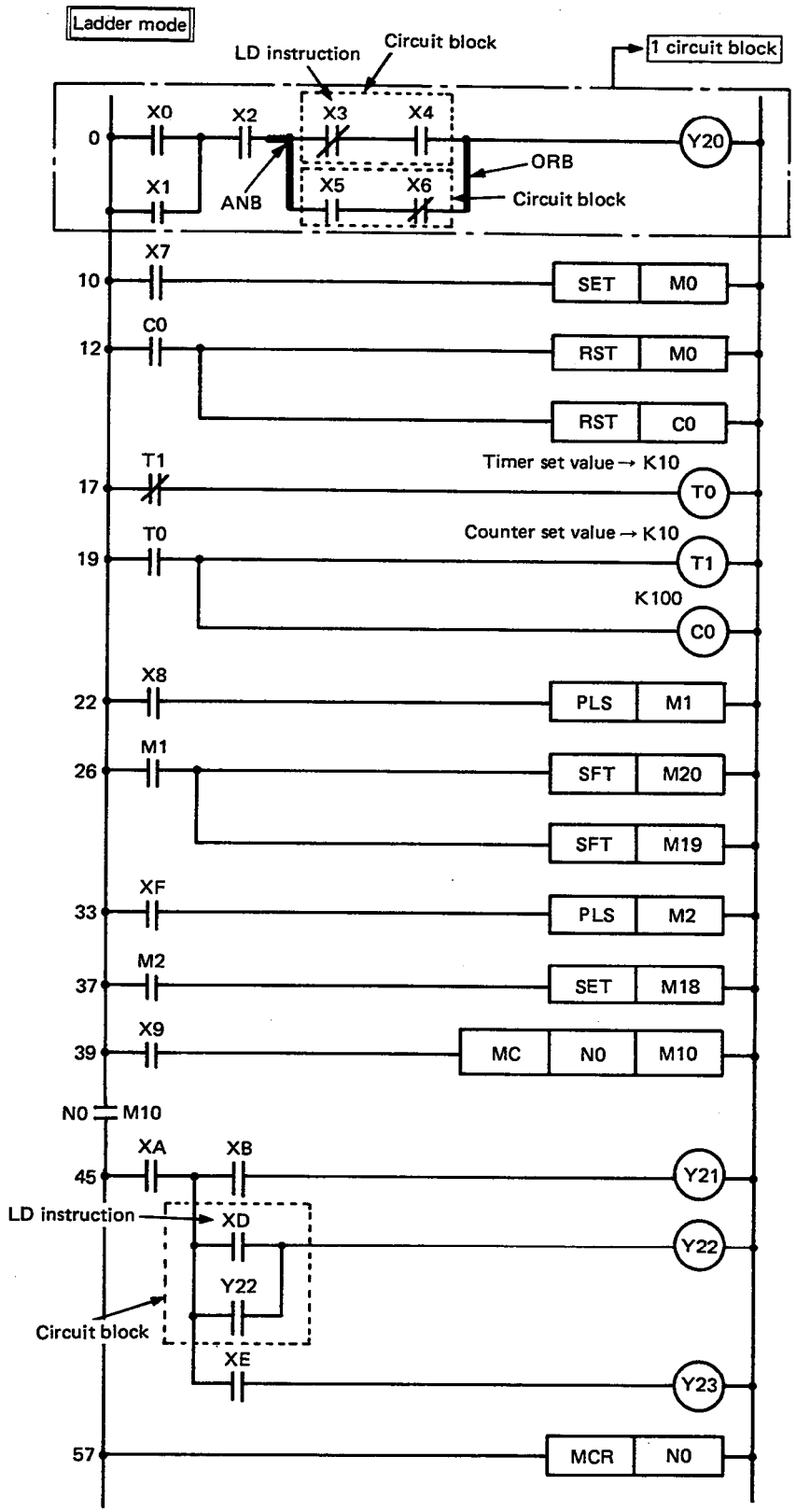
2) The sequence instructions are available in the following 21 types.

Function Symbol	Symbol	Function	Available Device: Bit device X	Y	M	L	B	F	Word (16-bit) device T	C	D	W	Z	V	Constant K	H	Pointer P	Level N	Step Number	Processing Unit	Execution Condition
LD		•N/O contact instruction used for head of circuit block. •Operation result is ON/OFF data of specified device.																			
LDI		•N/C contact instruction used for head of circuit block. •Operation result is reversal of ON/OFF data of specified device.																			
AND		•N/O contact serial connection instruction •Draws ON/OFF data of specified device, performs AND operation of this ON/OFF data and previous operation result, and uses new operation result.																			
ANI		•N/C contact serial connection instruction •Reverses and draws ON/OFF data of specified device, performs AND operation of this ON/OFF data and previous operation result, and uses new operation result.	○	○	○	○	○	○	○	○									1	1 bit	
OR		•1 N/O contact parallel connection instruction •Draws ON/OFF data of specified device, performs OR operation of this ON/OFF data and previous operation result, and uses new operation result.																			
ORI		•1 N/C contact parallel connection instruction •Reverses and draws ON/OFF data of specified device, performs OR operation of this ON/OFF data and previous operation result, and uses new operation result.																			
ANB		•Circuit block serial connection instruction •Performs AND operation of block A and block B, and uses this as operation result.																			
ORB		•2 or more contact parallel connection instruction •Performs OR operation of block A and block B, and uses this as operation result.																	1	—	
OUT		•Specified device ON/OFF instruction If operation result up to OUT instruction is on, specified device is enabled. If operation result up to OUT instruction is off, specified device is disabled.		○	○	○	○	○													
		•Timer instruction •Times while operation result up to OUT instruction is on. •When timer times out, contact is enabled.	Device						○										1	1 bit	
			Set value								○				○						
		•Counter instruction •Counts when operation result up to OUT instruction changes from OFF to ON. •When counter counts out, contact is enabled.	Device							○											
			Set value								○				○						
SET		•While operation result up to SET instruction is on, specified device is set (enabled). Device enabled by SET instruction remains enabled if operation result turns off.	D	○	○	○													1	1 bit	
			D				○	○											3		

Function Symbol	Symbol	Function	Available Device: Bit device X	Y	M	L	B	F	Word (16-bit) device T	C	D	W	Z	V	Constant K	H	Pointer P	Level N	Step Number	Processing Unit	Execution Condition
RST	RST D	•While operation result up to RST instruction is on, specified device is reset (bit device disabled, word device cleared to 0).	D		○	○	○												1	1 bit/ 1 word	
			D					○	○	○	○	○	○	○					3		
M C	M C n D	•Master control start instruction. •Use of nesting numbers in lower order allows nesting.	n															○	5	1 bit	
			D			○	○														
MCR	MCR n	•Master control end instruction. •Resets the nesting number specified at n and the succeeding numbers. •For nesting, use nesting numbers in higher order.	n															○	3	–	
PLS	PLS D	•Converts operation commands into pulses. •When operation result up to PLS instruction changes from OFF to ON, the specified device is enabled 1 scan.				○	○												3	1 bit	
PLF	PLF D	•Converts operation commands into pulses. •When operation result up to PLF instruction changes from OFF to ON, the specified device is enabled 1 scan.				○	○												3	1 bit	
SFT	SFT D	•ON/OFF data shift instruction. Before operation / After operation (Specified device −1) 1 → 0 ← Sets to OFF (0). Specified device 0 → 1				○	○												3	1 bit	
NOP	—	•No processing instruction. •Operation result up to NOP instruction is used as operation result. •For program deletion or spacing.																			
END	—	•Indicates the end of sequence program. •Execution terminates scanning at this step and returns to step 0. •Cannot be used midway through sequence program.																			
MPS	MPS / MRD / MPP	•Stores operation result up to MPS instruction. •MPS instruction may be used up to 16 times continuously. •If MPS instruction is used midway through sequence program, 1 is reduced from the usable number of MPP instructions.																			
MRD		•Reads operation result stored with MPS instruction. •The step following MRD instruction is the operation result immediately before MPS instruction.																			
MPP		•Reads and clears operation result stored with MPS instruction. •The step following MPP instruction is the operation result immediately before MPS instruction.																			

5.1 Program Example Using Sequence Instructions

A program using the sequence instructions is as shown below in ladder mode and list mode.



List mode

0	LD	X0	
1	OR	X1	
2	AND	X2	
3	LDI	X3	
4	AND	X4	
5	LD	X5	
6	ANI	X6	
7	ORB		
8	ANB		
9	OUT	Y20	
10	LD	X7	
11	SET	M0	
12	LD	C0	
13	RST	M0	
14	RST	C0	
17	LDI	T1	
18	OUT	T0	K10
19	LD	T0	
20	OUT	T1	K10
21	OUT	C0	K100
22	LD	X8	
23	PLS	M1	
26	LD	M1	
27	SFT	M20	
30	SFT	M19	
33	LD	XF	
35	PLS	M2	
37	LD	M2	
38	SET	M18	
39	LD	X9	
40	MC	N0	M10
45	LD	XA	
46	MPS		
47	AND	XB	
48	OUT	Y21	
49	MRD		
50	LD	XD	
51	OR	Y22	
52	ANB		
53	OUT	Y22	
54	MPP		
55	AND	XE	
56	OUT	Y23	
57	MCR	N0	
60	END		

IB (NA) 66057-A

5.2 Supplementary Explanation for Sequence Instructions

Supplementary explanation will be given for the sequence instructions shown in Table 5.1.

5.2.1 LD, LDI, AND, ANI, OR, ORI

1) Contacts to be placed at the head of the circuit block using LD or LDI instruction are as follows.

   a) Contact connected to the left bus
   b) Contact used at the head of parallel connection

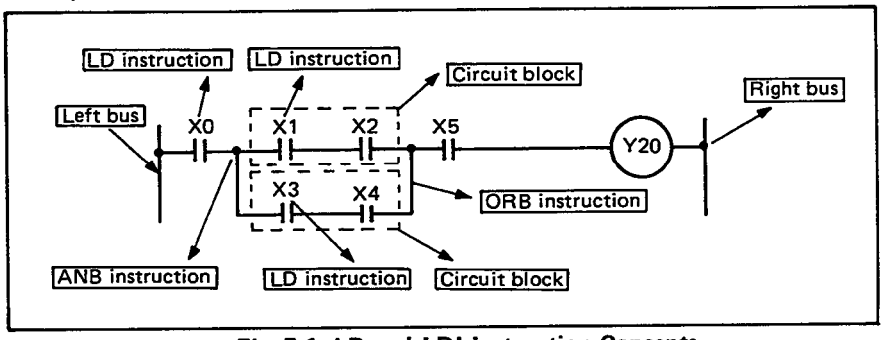


Fig. 5.1 LD and LDI Instruction Concepts

2) To serially connect contacts using AND or ANI instruction, there is no restriction on the number of used AND or ANI instructions. In ladder mode of the GPP/PHP/HGP, however, the usable number is limited as described below.

   a) Write: Up to 21-ladder (maximum 210 instructions) circuit can be created.

   b) Read: Up to 24-ladder (maximum 240 instructions) circuit can be displayed.

3) To parallelly connect contacts using OR or ORI instruction, there is no limit to the usable number of OR or ORI instructions. In ladder mode of the GPP/PHP/HGP, however, the usable number is limited as described below.

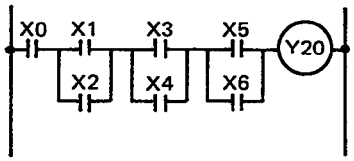
   a) Write: Up to 23-ladder circuit can be created.

   b) Read: Up to 23-ladder circuit can be displayed. For a circuit exceeding 23 ladders, proper display cannot be provided.

5.2.2 ANB, ORB

1) To connect circuit blocks using ANB (ORB), coding should be as shown in Example 1. To write ANB (ORB) continuously as shown in Example 2, a maximum of 7 instructions may be used. For 8 or more instructions, proper operation cannot be performed.

Ladder Example



Coding

Example 1

Step number	Instruction	Device
0	LD	X0
1	LD	X1
2	OR	X2
3	ANB	
4	LD	X3
5	OR	X4
6	ANB	
7	LD	X5
8	OR	X6
9	ANB	
10	OUT	Y20

Usable number of ANB is not restricted.

Example 2

Step number	Instruction	Device
0	LD	X0
1	LD	X1
2	OR	X2
3	LD	X3
4	OR	X4
5	LD	X5
6	OR	X6
7	ANB	
8	ANB	
9	ANB	
10	OUT	Y20

A maximum of 7 ANB instructions are usable.

Fig. 5.2 ANB Instruction Coding Examples

5.2.3 OUT, SET, RST

1) ON/OFF of bit device using OUT, SET, and RST instructions is as follows.

(a) OUT: Turns on/off the device depending on the operation result up to OUT instruction.

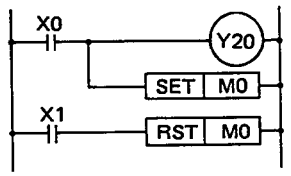
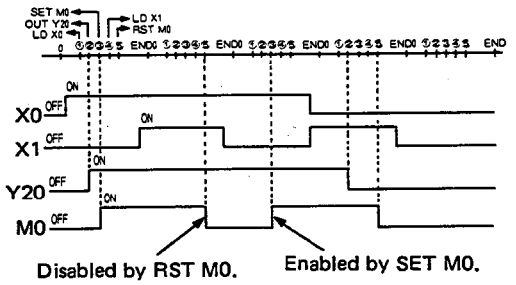
- Operation result up to OUT instruction is on:
  The specified device is turned on.

- Operation result up to OUT instruction is off:
  The specified device is turned off.

(b) SET, RST: While the operation result up to SET or RST instruction is on, the following operation is performed. If the operation result is off, processing is not performed.

- SET: While the operation result is on, the specified device is enabled. Once enabled, the specified device remains enabled if the operation result is disabled.

- RST: While the operation result is on, the specified device is disabled.

Ladder Example	Operation Timing
	

2) Upon execution of RST T[ ]and RST C[ ], the timer and counter states change as described below:

   (a) Coil, contact: Turned off. (Contact is not turned off after execution of END (FEND) instruction.)

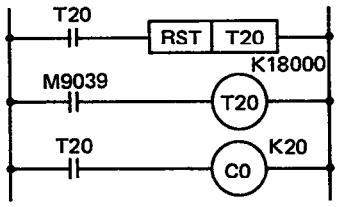
   (b) Present value: Reset to 0.

3) The contact of timer or counter is turned on after END (FEND) instruction. Therefore, when the timer (counter) has timed out (counted out), the contact remains on from step 0 to the execution of RST T[ ](RST C[ ]).

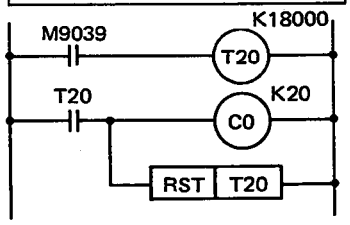
--Example----------

In the left ladder below, C0 cannot count. Correct as shown in the right ladder.

Ladder which does not allow C0 counting



Ladder which allows C0 counting



When T20 times out, the contact is enabled after execution of END instruction. When RST T20 is executed, the contact is disabled. Therefore, since T20 contact is not enabled after RST T20, C0 cannot count.

5.2.4 MC, MCR

1) While the operation result up to MC instruction is on, the operation result between MC and MCR remains unchanged.

2) If the operation result up to MC instruction is off;

   (a) 100ms, 10ms timer:
   The coil and contact are turned off and the present value is reset to 0.

   (b) 100ms retentive timer, counter:
   The coil is turned off and the contact and present value remain unchanged.

   (c) OUT instruction: All turn off.

   (d) SET, RST, SFT: Remain unchanged.

3) To perform the nesting, use nesting numbers as described below.

   (a) MC instruction: Use nesting numbers in order of lower ones.

   (b) MCR instruction: Use nesting numbers in order of higher ones.

4) The program between MC and MCR is executed irrespective of ON/OFF of the operation result up to MC instruction.

5) While the operation result up to MC instruction is on, the device specified at the destination (D) turns on. Use of the same devices for OUT instruction, etc. results in duplicate coil.

5.2.5 PLS, PLF

1) If, after PLS (PLF) instruction, the RUN key switch is moved from RUN to STOP and then to RUN, PLS (PLF) instruction is not executed.

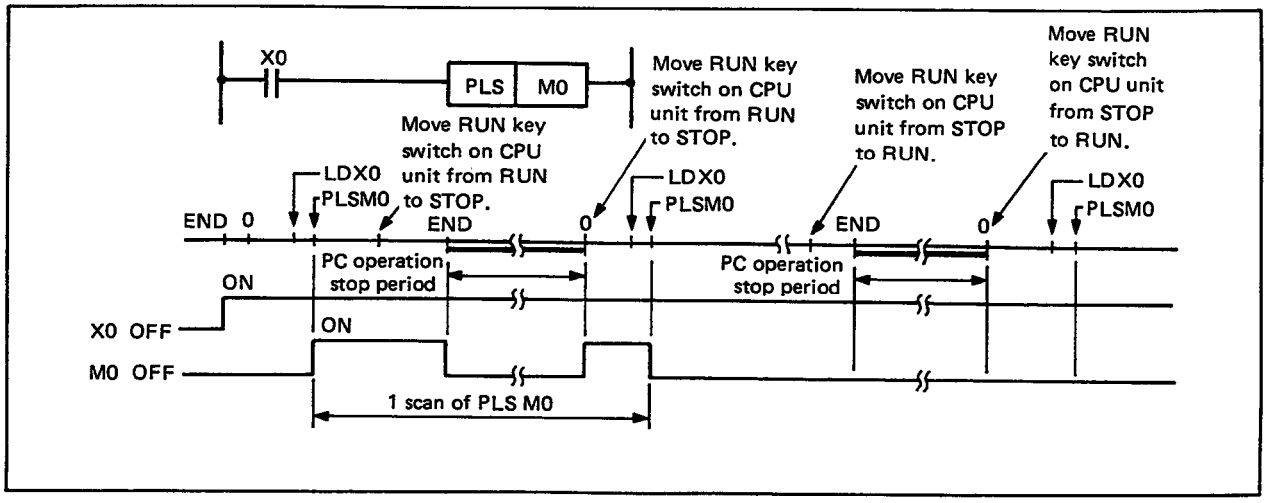


Fig. 5.4 PLS Instruction Operation

2) Note that when the preceding condition is on, PLS instruction is executed again if:

   (a) PLS is provided between MC and MCR and the operation result up to MC instruction has changed from OFF to ON.

   (b) the preceding condition has been latched and the power has been turned from OFF to ON.

5.2.6 SFT

1) Shifts ON/OFF state of the device preceding the one specified at D (destination) to the specified device and disables the preceding device.

2) Enable the head device to be shifted using SET instruction.

3) To use SFT continuously, create the program in order of larger device numbers.

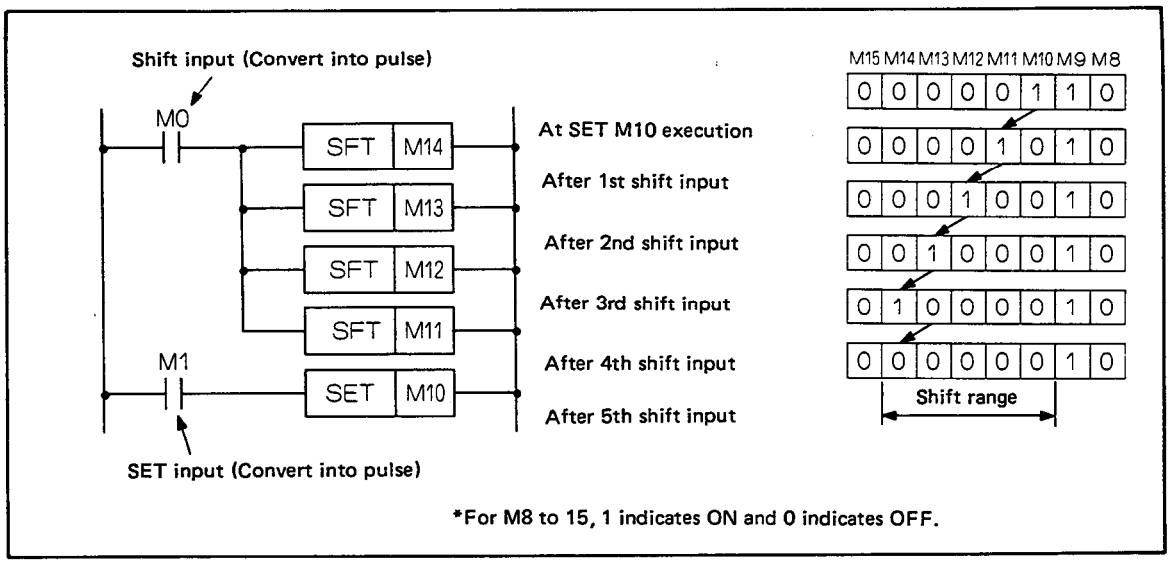


Fig. 5.5 SFT Instruction Operation

6. BASIC AND APPLICATION INSTRUCTIONS

6.1 Comparison Operation Instructions

1) The comparison operation instruction is handled as an N/O contact, compares two datas, and turns on when the condition enables.

2) The comparison operation instructions are classified and to be specified as described below.

   a) Classification: $=, \neq, >, \leqq, <, \geqq$

   b) Specification: May be specified in the same manner as contact instructions (LD, AND, OR) of the sequence instructions.

3) According to the above 2), a) and b), the comparison operation instructions are available in the following 18 types.

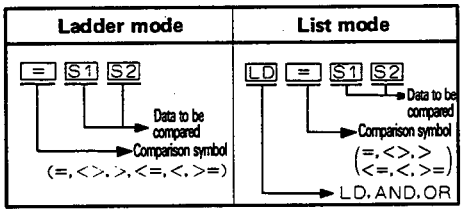
4) The comparison instruction compares given data, regarding it as a BIN value. Therefore, the BCD and hexadecimal values of which highest bit contain "1" (8000 or greater in BCD and hexadecimal) are operated as a negative value.

Function Symbol		Available Device: Bit device X	Y	M	L	B	Word (16-bit) device F	T	C	D	W	Z	V	Constant K	H	Pointer P	Level N	Processing Unit: 1bit	16bits	32bits	Other	Digit Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
LD=, AND= OR=	S1	○	○	○	○	○	○	●	●	●	●	○	○	○				—	○	—	—	K1 ~ K4	5/7	○	LD OR AND	—
	S2	○	○	○	○	○	○	●	●	●	●	○	○	○												

*1 : W (link register) may be used only for A0J2P23 (R23).
*2 : The number of step is 7 when:
a) index qualification is used; or
b) bit device digit specification is other than K4.
*3 : ● indicates that index qualification may be used.

Classification	Function Symbol	Symbol	Function: ON condition	Function: OFF condition	Program Example
=	LD=	= S1 S2	S1 = S2	S1 ≠ S2	If X0 to XF data is equal to D0 data, this program enables Y20.
	AND=	= S1 S2			Ladder example: 0 — = K4X0 D0 — (Y20) — If (X0 to X1F data) = (D0 data), Y20 enables.
	OR=	= S1 S2			
≠ (<>)	LD<>	<> S1 S2	S1 ≠ S2	S1 = S2	Coding
	AND<>	<> S1 S2			
	OR<>	<> S1 S2			
>	LD>	> S1 S2	S1 > S2	S1 ≤ S2	
	AND>	> S1 S2			
	OR>	> S1 S2			

Step No.	Instruction	Device			
0	LD=	K4X0	D0		
5	OUT	Y20			
6	END				

Classification	Function Symbol	Symbol	Function		Program Example
			ON condition	OFF condition	
≦	LD<=	<= S1 S2	S1 ≦ S2	S1 > S2	**REMARKS** 1. Comparison operation instruction specifying format is as shown below.  2. The number of steps is 7 if: a) Index qualification has been used; or b) Bit device digit specified is other than K4. 3. Index qualification cannot be performed for bit device.
	AND<=	<= S1 S2			
	OR<=	<= S1 S2			
<	LD<	< S1 S2	S1 < S2	S1 ≧ S2	
	AND<	< S1 S2			
	OR<	< S1 S2			
≧	LD>=	>= S1 S2	S1 ≧ S2	S1 < S2	
	AND>=	>= S1 S2			
	OR>=	>= S1 S2			

6.2 BIN Data Arithmetic Operation Instructions

1) The arithmetic operation instructions for BIN data perform the addition, subtraction, multiplication, and division of two BIN datas.

2) The operation results of BIN data arithmetic operation instructions are as follows:

a) Addition
If (addition result) $\leqq$ 32767, 5 + 8 → 13
If (addition result) $>$ 32767, 32767 + 1 → −32768

b) Subtraction
If (minuend) $>$ (subtrahend), 8 − 5 → 3
If (minuend) $<$ (subtrahend), 5 − 8 → −3

c) Multiplication
If (positive number) x (positive number), 5 x 3 → 15
If (positive number) x (negative number), 5 x (−3) → −15
If (negative number) x (positive number), −5 x 3 → −15
If (negative number) x (negative number), (−5) x (−3) → 15

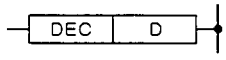
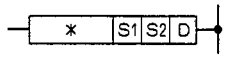
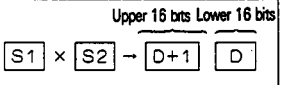
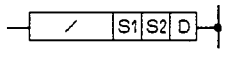
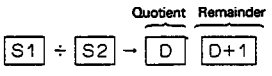
d) Division
If (positive number) ÷ (positive number), 5 ÷ 3 → 1, remainder 2
If (positive number) ÷ (negative number), 5 ÷ (−3) → −1, remainder 2
If (negative number) ÷ (positive number), −5 ÷ 3 → −1, remainder −2
If (negative number) ÷ (negative number), (−5) ÷ (−3) → 1, remainder −2

3) The BIN data arithmetic operation instructions are available in the following 6 types.

Function Symbol		Available Device: Bit device X	Y	M	L	B	F	Word (16-bit) device T	C	D	W	Z	V	Constant K	H	Pointer P	Level N	Processing Unit: 1 bit	16 bits	32 bits	Other	Digit Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
+, −	S							●	●	●	●	○	○	○				—	○	—	—	---	5	○		—
	D							●	●	●	●	○	○													
INC, DEC	D							●	●	●	●	○	○					—	○	—	—	---	3			
*, /	S1							●	●	●	●	○	○	○				—	○	—	—	—	7			
	S2							●	●	●	●	○	○	○												
	D							●	●	●	●	○						—	—	○	—					

*1: W (link register) may be used only for A0J2P23 (R23).
*2: ● indicates that index qualification may be used.

Classification	Function Symbol	Symbol	Function	Program Example
Addition	+	+ S D	D + S → D	When X0 enables, this program adds 10 to D0 data.
	INC	INC D	D + 1 → D	Ladder example: 0 X0 PLS M0 — At the rise (OFF→ON) of X0, M0 enables 1 scan. 4 M0 + K10 D0 — When M0 enables, addition is performed.
Subtraction	−	− S D	D − S → D	

Classification	Function Symbol	Symbol	Function	Program Example
Subtraction	DEC			Coding
Multiplication	*			
Division	/			

Step No.	Instruction	Device			
0	LD	X0			
1	PLS	M0			
4	LD	M0			
5	+	K10	D0		
10	END				

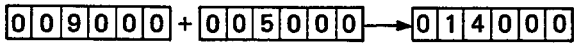
6.3 6-Digit BCD Data Arithmetic Operation Instructions

1) The arithmetic operation instructions of BCD data perform the addition and subtraction of two BCD datas.

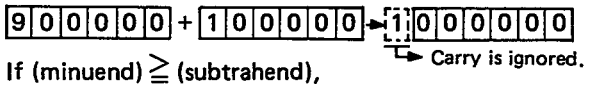
2) The operation results of the BCD data arithmetic operation instructions are as follows:

a) Addition

If (addition result) ≦ 999999,

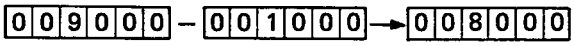


If (addition result) > 999999,

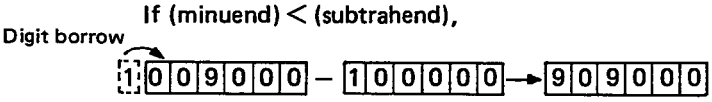


b) Subtraction

If (minuend) ≧ (subtrahend),



If (minuend) < (subtrahend),



3) The BCD data arithmetic operation instructions are available in the following 2 types.

Function Symbol		Available Device: Bit device (X Y M L B F)	Word (16-bit) device (T C D W Z V)	Constant (K H)	Pointer (P)	Level (N)	Processing Unit: 1 bit	16 bits	32 bits	Other	Digit Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
DB+, DB−	S		● ● ● ●	○ ○			—	—	—	24 bit	—	9	○		—
	D		● ● ● ●												

*1: W (link register) may be used only for A0J2P23 (R23).
*2: ● indicates that index qualification may be used.

Classification	Function Symbol	Symbol	Function	Program Example
Addition	DB+	DB+ S D	D+1 D + S+1 S: BCD 6 digits + BCD 6 digits → D+1 D: Set to 0. BCD 6 digits. 2 digits marked * are ignored.	When X0 enables, this program adds 100 to D0 data. Ladder example: X0 PLS M0 (At the rise (OFF→ON) of X0, M0 enables 1 scan.); M0 DB+ H100 D0 (When M0 enables, addition is performed.)
Subtraction	DB−	DB− S D	D+1 D − S+1 S: BCD 6 digits − BCD 6 digits → D+1 D: Set to 0. BCD 6 digits. 2 digits marked * are ignored.	Coding:

Coding

Step No.	Instruction	Device			
0	LD	X0			
1	PLS	M0			
4	LD	M0			
5	DB+	H100	D0		
14	END				

6.4 BCD↔BIN Conversion Instructions

1) The BCD↔BIN conversion instructions convert the BCD data into BIN data and the BIN data into BCD data.

2) Values usable for the BCD↔BIN conversion instructions are as follows:

   a) 16-bit processing instructions (BCD, BIN): 0 to 9999
   b) 24-bit processing instructions (DBCD, DBIN): 0 to 999999

3) 24-bit processing instruction uses 2 word devices (32 bits). Among the 32 bits, the upper 8 bits of BCD data are handled as described below. BIN data ranges from 0 to 999999.

   a) Source: The upper 8 bits are ignored.
   b) Destination: The upper 8 bits are cleared to 0.

4) The BCD↔BIN conversion instructions are available in the following 4 types.

Function Symbol		Available Device: Bit device X	Y	M	L	B	F	Word (16-bit) device T	C	D	W	Z	V	Constant K	H	Pointer P	Level N	Processing Unit 1bit	16bits	32bits	Other	Dig It Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
BCD. BIN	S	○	○	○	○	○	○	●	●	●	●	○	○					—	○	—	K1 ~ K4	—	5	○	⊓	—
	D	○	○	○	○	○	○	●	●	●	●	○	○													
DBCD. DBIN	S							●	●	●	●	○						—	—	—	24 bits	—	9			
	D							●	●	●	●	○														

*1: W (link register) may be used only for A0J2P23 (R23).
*2: When bit device is used, device number to be specified is only 0 or a multiple of 8.
*3: Index qualification cannot be performed for bit device.
*4: ● indicates that index qualification may be used.

Classification	Function Symbol	Symbol	Function	Program Example
BCD conversion	BCD	BCD S D	BCD conversion: S → D; S: BIN0~9999, D: BCD0~9999	When X0 enables, this program converts BIN data in D0 into BCD data and stores to D10.
	DBCD	DBCD S D	BCD conversion: S+1, S → D+1, D; S: BIN0~999999, D: BCD0~999999, Set to 0.	Ladder example: X0 PLS M0; M0 BCD D0 D10. At the rise (OFF→ON) of X0, M0 enables 1 scan. When M0 enables, conversion is made.
BIN conversion	BIN	BIN S D	BIN conversion: S → D; S: BCD0~9999, D: BIN0~9999	Coding
	DBIN	DBIN S D	BIN conversion: S+1, S → D+1, D; S: BCD0~99999999, D: BIN0~99999999, Ignored	

Step No.	Instruction	Device			
0	LD	X0			
1	PLS	M0			
4	LD	M0			
5	BCD	D0	D10		
10	END				

POINT

(1) If the data specified at the source is outside the following range, operation error occurs and the error flag enables.

a) BCD, BIN instruction: 0 to 9999
b) DBCD, DBIN instruction: 0 to 999999

6.5 Transfer Instructions

1) The transfer instructions store the data at the source into the device at the destination. Therefore, the source data is the same as the destination data.

2) The transfer instructions are available in the following 2 types.

Function Symbol		Available Device: Bit device X	Y	M	L	B	F	Word (16-bit) device T	C	D	W	Z	V	Constant K	H	Pointer P	Level N	Processing Unit: 1bit	16bits	32bits	Other	Digit Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
MOV	S	○	○	○	○	○	○	●	●	●	●	○	○	○	○			—	○	—	—	K1 ~ K5	5	○		
	D		○	○	○	○	○	●	●	●	●	○	○													
FMOV	S							●	●	●	●	○	○	○	○			—	○	—	—	—	9	○		—
	D							●	●	●	●							—		—	n point			○		
	n													○	○									—		

*1: W (link register) may be used only for A0J2P23 (R23).
*2: When bit device is used, device number to be specified is only 0 or a multiple of 8.
*3: Index qualification cannot be performed for bit device.
*4: ● indicates that index qualification may be used.

Classification	Function Symbol	Symbol	Function	Program Example
Transfer	MOV	MOV S D	S Transfer D; Same data	When X0 enables, this program sets 11 points, D0 to D10, to 0. Ladder example: X0 PLS M0; M0 FMOV K0 D0 K11. At the rise (OFF→ON) of X0, M0 enables 1 scan. When M0 enables, batch transfer is made. Coding: see table below
Batch transfer	FMOV	FMOV S D n	S D+0 D+1 ... D+(n-2) D+(n-1); Same data at "n" points, beginning with D, and S.	

Step No.	Instruction	Device			
0	LD	X0			
1	PLS	M0			
4	LD	M0			
5	FMOV	K0	D0	K11	
14	END				

POINT

(1) For FMOV instruction, specify n (number of transferred points) which does not exceed the used device range. If the value exceeds the used device range, the data of other device will be rewritten.

IB (NA) 66057-B

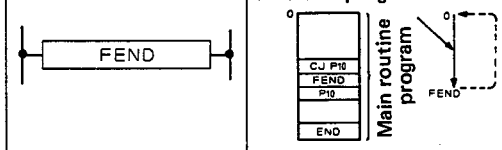
6.6 Program Branch Instructions

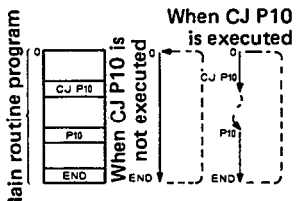
1) The program branch instructions cause a branch inside the sequence program and execute the subroutine program/microcomputer program.

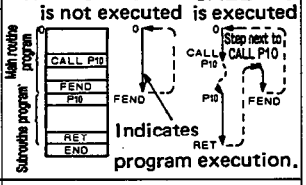
2) The program branch instructions are available in the following 6 types.

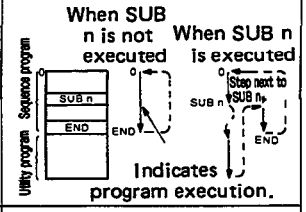
Function Symbol	Available Device:		Bit device				Word (16-bit) device					Constant		Pointer	Level	Processing Unit				Digit Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
		X	Y	M	L	B	F	T	C	D	W	Z	V	K	H	P	N	1 bit	16bits	32bits	Other					
FEND, RET																		—	—	—	—	—	1	—		—
COM																		—	—	—	—	—	1	—		—
CJ, CALL	P															O		—	—	—	—	—	3	—		—
SUB	n													O	O			—	—	—	—	—	3	—		—

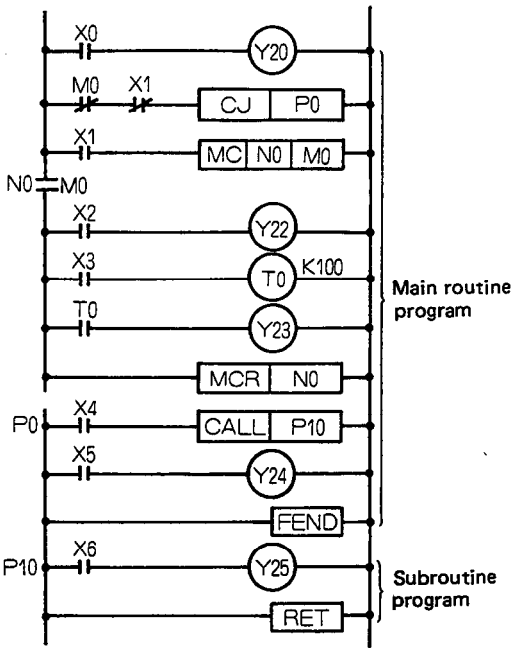
Classification	Function Symbol	Symbol	Function	Program Example
Program end	FEND	FEND	Terminates main routine program. Indicates program execution.	When X1 is off, this program causes jump betweer MC and MCR. When X4 is on, this program executes subroutine between P10 and RET.
Jump	CJ	CJ P**	Jumps to specified pointer number (P**). P**: P0 to P63 available. P63: Indicates END instruction.	
Subroutine program call/return	CALL	CALL P**	CALL: Executes specified subroutine program. P**: P0 to P62 available.	
	RET	RET	RET: Terminates subroutine program.	
Microcomputer program call	SUB	SUB n	Executes utility program.	
Link refresh	COM	COM	Performs link refresh and general data processing.	







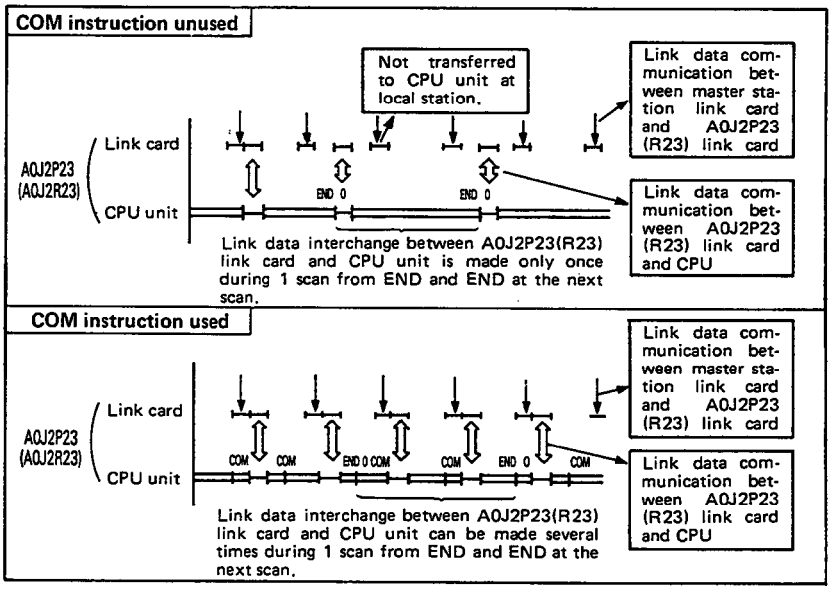




POINT

(1) When COM instruction is executed, the following operation is performed.

1) If communication of link data has been completed between the master station and the link card in the A0J2P23(R23) before the execution of COM instruction, data interchange will be made immediately between the CPU unit and link card.
If link data communication has not yet been completed between the master station and the link card in the A0J2P23(R23), data interchange will be made between the CPU unit and link card after completion of link data communication.



2) Execution of COM instruction enables general data processing (communication between CPU unit and peripheral equipment or special function unit) in addition to data link information interchange.

(2) COM instruction may be used several times during 1 scan. However, each time COM instruction is executed, scan time increases by the data link information interchange time and general data processing time.

6.6.1 Program explanation

(1) Main routine program

The main routine program is always executed during CPU RUN. This program ranges from step 0 to END (FEND).
The main program configuration and processing will be explained below.

1) There is a program which is not always executed

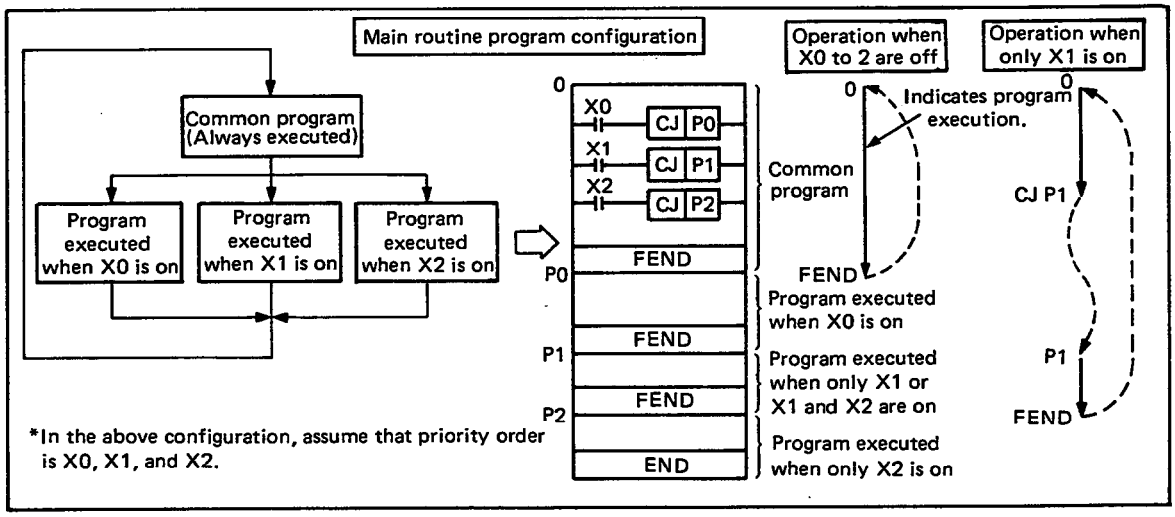
If there exists a program which is not always executed, use 1 END instruction at the end of the program and use CJ instruction between 0 and END so that the normally un-executed program is skipped and not executed. In this case, execution cannot jump to a location below END. If the execution has jumped to a place below END, error will occur, the PC will stop operation, and all outputs Y will turn off. Applications are as follows:

- Since the program between MC and MCR is executed if the master control is off, jump the program between MC and MCR using CJ instruction to increase the processing speed, when the master control is off.

- Skip the program, which is not executed normally, using CJ instruction to increase the processing speed.

2) To divide into control blocks

To divide the main routine program into several control blocks, terminate 1 program with FEND instruction. Then, arrange all blocks serially to make 1 main routine program. Terminate the main routine program with END instruction. To skip the divided programs, use CJ instruction.



POINT

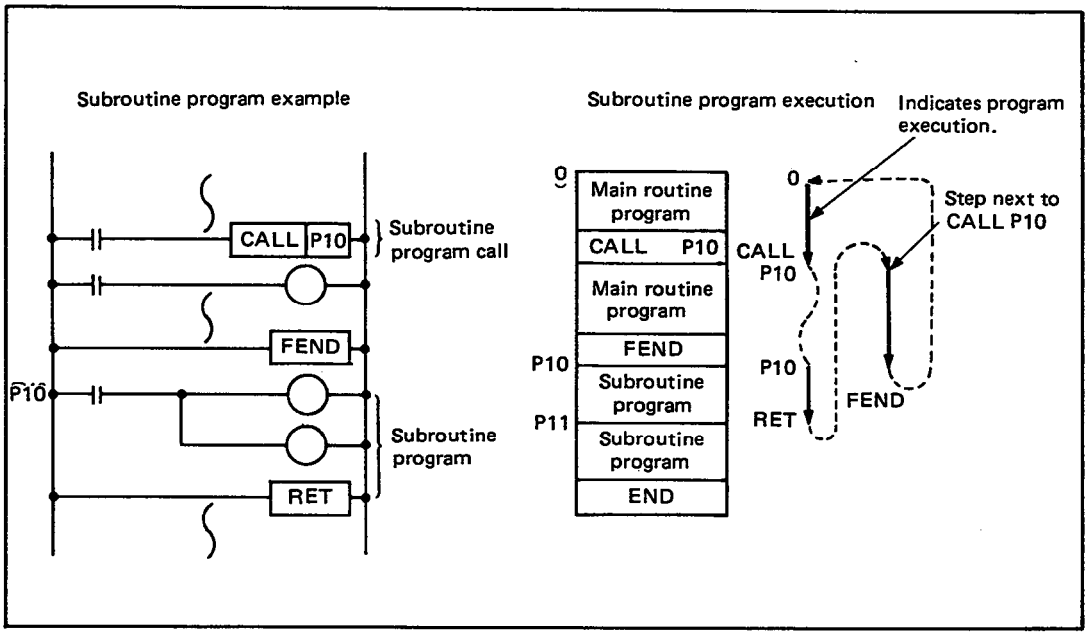
(1) When CJ instruction is not used, the program may be executed from step 0 to the first END (FEND). Therefore, the programs created below END (FEND) cannot be executed.

Main routine program	Executable
END (FEND)	
Main routine program	Unexecutable
END	

(2) Even if CJ instruction is used, the program below END cannot be executed.

(2) Subroutine program

Use the subroutine program if it is desired to execute a specific program several times during 1 scan or to execute a given program when a certain condition has enabled.
Create the subroutine program below the main routine program (below FEND). Execute the subroutine program using CALL instruction. When CALL instruction is executed after its input condition has been enabled, the execution jumps to the subroutine program. When the input condition is off, the main routine program is executed and the subroutine program is not executed.



POINT

(1) The subroutine programs cannot be nested.

6.7 Shift Instructions

1) The shift instructions shift data.

2) The shift instructions are available in the following 4 types.

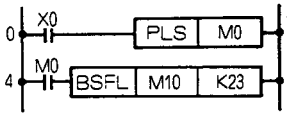
The shift data of bit devices (Y, M/L, B, F) is ON/OFF information. The shift data of word devices (T, C, D, W) is the content of word device.

Function Symbol		Available Device: Bit device X	Y	M	L	B	F	Word (16-bit) device T	C	D	W	Z	V	Constant K	H	Pointer P	Level N	Processing Unit: 1 bit	16 bits	32 bits	Other	Digit Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
BSFR, BSFL	D	○	○	○	○	○												—	—	—	n bit	—	7	—		○
	n													○	○									○		
DSFR, DSFL	D							●	●	●	●							—	—	—	n point			○		—
	n													○	○											

*1: W (link register) may be used only for A0J2P23 (R23).
*2: Index qualification cannot be performed for bit device.
*3: ● indicates that index qualification may be used.

Classification	Function Symbol	Symbol	Function	Program Example
Left shift	BSFL	BSFL D n	n (shift range), D+(n−1), D+1, D, Before operation, After operation, Carry flag M9012	When X0 enables, this program shifts 23 points, M10 to M32, to the left by 1 point.
	DSFL	DSFL D n	n (shift range), D+(n−1), D+1, D, Before operation, After operation	
Right shift	BSFR	BSFR D n	n (shift range), D+(n−1), D+1, D, Before operation, After operation, Carry flag M9012	
	DSFR	DSFR D n	n (shift range), D+(n−1), D+1, D, Before operation, After operation	

Ladder example



At the rise (OFF→ON) of X0, M0 enables 1 scan.
When M0 enables, M10 to M32 are shifted 1 point to the left.

Coding

Step No.	Instruction	Device			
0	LD	X0			
1	PLS	M0			
4	LD	M0			
5	BSFL	M10	K23		
12	END				

POINT

For BSFR and BSFL, a value to be specified at D (destination) does not have to be a multiple of 0 or 8. Any word device number (within the specified device range) can be specified at D.

6.8 Logical Operation Instructions

1) The logical operation instructions perform logical operations, such as logical product and logical sum, in units of 1 bit.

2) The logical operation instructions are available in the following 4 types.

Function Symbol		Available Device: Bit device X	Y	M	L	B	F	Word (16-bit) device T	C	D	W	Z	V	Constant K	H	Pointer P	Level N	Processing Unit: 1bit	16bits	32bits	Other	Digit Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
WAND, WOR WXOR, WXNR	S							●	●	●	●	○	○	○					○			—	5	○		—
	D							●	●	●	●	○	○													
NEG	D							●	●	●	●	○	○						○			—	3	○		—

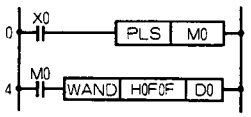
*1: W (link register) may be used only for A0J2P23 (R23).
*2: ● indicates that index qualification may be used.

Classification	Function Symbol	Symbol	Function
Logical product	WAND	WAND S D	D ∧ S → D
Logical sum	WOR	WOR S D	D ∨ S → D
Exclusive OR	WXOR	WXOR S D	D ⊻ S → D
Negative exclusive OR	WXNR	WXNR S D	$\overline{D}$ ⊻ S → D
2's complement	NEG	NEG D	$\overline{D}$ + 1 → D

Program Example

When X0 enables, this program performs AND operation of D0 data and 0F0FH and stores the result to D0.

Ladder example



At the rise (OFF→ON) of X0, M0 enables 1 scan.

When M0 enables, AND operation is performed.

Coding

Step No.	Instruction	Device			
0	LD	X0			
1	PLS	M0			
4	LD	M0			
5	WAND	H0F0F	D0		
10	END				

IB (NA) 66057-B

REMARKS

Processings of logical product, logical sum, exclusive OR, and negative exclusive OR are as described in the following table.

Classification	Processing	Operation Expression	Example A	B	Y
Logical product	If both datas A and B are 1, the result is 1. Otherwise, the result is 0.	Y=A•B	0	0	0
			0	1	0
			1	0	0
			1	1	1
Logical sum	If both datas A and B are 0, the result is 0. Otherwise, the result is 1.	Y=A+B	0	0	0
			0	1	1
			1	0	1
			1	1	1
Exclusive OR	If two datas A and B are equal, the result is 0. If they are different, the result is 1.	$Y=\overline{A}\cdot B+A\cdot\overline{B}$	0	0	0
			0	1	1
			1	0	1
			1	1	0
Negative exclusive OR	If two datas A and B are equal, the result is 1. If they are different, the result is 0.	$Y=(\overline{A}+B)(A+\overline{B})$	0	0	1
			0	1	0
			1	0	0
			1	1	1

6.9 Data Processing Instructions

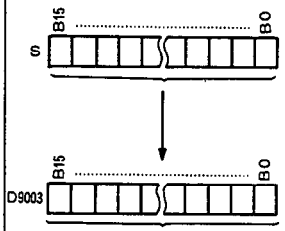
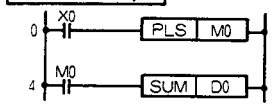
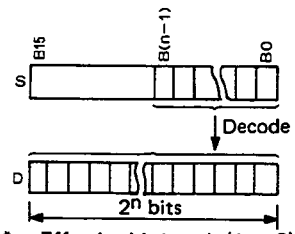
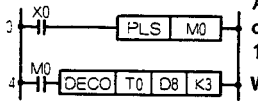
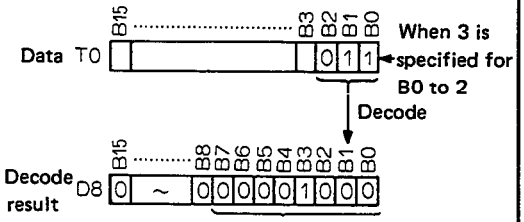
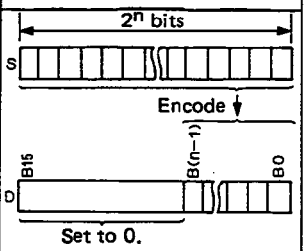
1) The data processing instructions count, decode, and encode data.

Function Symbol		Available Device: Bit device X	Y	M	L	B	F	Word (16-bit) device T	C	D	W	Z	V	Constant K	H	Pointer P	Level N	Processing Unit: 1 bit	16bits	32bits	Other	Digit Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
SUM	S							●	●	●	●							—	○	—	—	—	3			
DECO	S							●	●	●	●			○	○			—	—	—	n bits					
	D							●	●	●	●							—	—	—	2^n bits	—	9			
	n													○	○			—	—	—	—			○		—
ENCO	S							●	●	●	●							—	—	—	2^n bits					
	D							●	●	●	●							—	—	—	n bits	—	9			
	n													○	○			—	—	—	—					

*1: W (link register) may be used only for A0J2P23 (R23).

*2: For DECO and ENCO instructions, bit device cannot be used. Therefore, the occupying points of word device are as follows:
- n bits: 1 point
- 2^n bits: $2^n/16$ points (value below decimal to be raised)

*3: ● indicates that index qualification may be used.

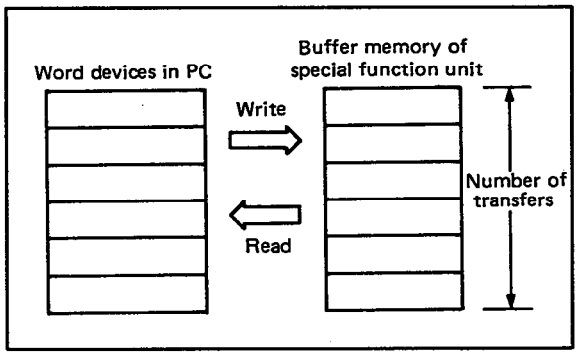
Classification	Function Symbol	Symbol	Function	Program Example
Count	SUM	SUM S		When X0 enables, this program obtains the quantity of D0 datas set to 1. Ladder example  At the rise (OFF→ON) of X0, M0 enables 1 scan. When M0 enables, the total number is stored into D9003 in BIN. Coding Step No. / Instruction / Device: 0 LD X0; 1 PLS M0; 4 LD M0; 5 SUM D0; 8 END
Decode	DECO	DECO S D n	 *n: Effective bit length (1 to 8) B0 to B(n−1) are data to be decoded.	Concept of decode is as explained below. Ladder example  At the rise (OFF→ON) of X0, M0 enables 1 scan. When M0 enables, decode is performed.  When valid bit is 3, 8 bits are occupied. B3 at the 3rd place from B0 is enabled.
Encode	ENCO	ENCO S D n	 *n: Effective bit length (1 to 8) Encode result is stored into B0 to B(n−1).	*Encode is reverse operation to decode.

6.10 Special Unit Instructions

1) The special unit instructions perform the following operations to the buffer memory in the special function unit.

Write: Transfers data from the word devices of the PC to the buffer memory of special function unit.

Read: Transfers data from the buffer memory of special function unit to the word devices of the PC.



2) The special unit instructions handle 16-bit and 32-bit data.

16-bit data processing: Transfer is made in units of 16 bits. (FROM, TO)

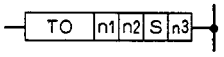
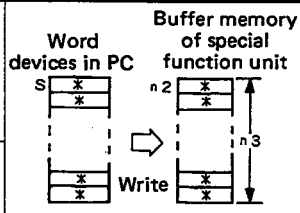
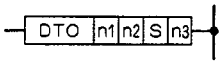
32-bit data processing: Transfer is made in units of 32 bits (2 words). (DFRO, DTO)

3) The special unit instructions are available in the following 4 types.

Function Symbol	Available Device: Bit device (X Y M L B F)	Word (16-bit) device (T C D W Z V)	Constant (K)	Pointer (H)	Level (P)	N	Processing Unit: 1bit	16bits	32bits	Other	Digit Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
TO, DTO FROM, DFRO	n1		○	○			—	○	○	—	—	9/11	○		
	n2		○	○											
	S_D	●●●													
	n3		○	○											

*1: W (link register) may be used only for A0J2P23 (R23).
*2: The number of steps is 11 only for DTO instruction.
*3: ● indicates that index qualification may be used.

Classification	Function Symbol	Symbol	Function	Program Example
Data read	FROM	FROM n1 n2 D n3	Word devices in PC / Buffer memory of special function unit; D, n2, n3; Read	While X0 enables, this program reads the X-axis output speed of AD71, which is loaded in slot 0 of extension base unit, to D0.
	DFRO	DFRO n1 n2 D n3	✱ : FROM instruction in units of 16 bits. DFRO instruction in units of 32 bits.	Ladder example: X0 — FROM H10 K600 D0 K1

Classification	Function Symbol	Symbol	Function	Program Example
Data write	TO		 ✱ : TO instruction in units of 16 bits. DTO instruction in units of 32 bits.	Coding
	DTO			

Step No.	Instruction	Device			
0	LD	X0			
1	FROM	H10	K600	D0	K1
6	END				

POINT

For the special unit instructions, specify n1, n2, and n3 as described below.

(1) n1: Specify the upper 2 digits of the head address (represented in hexadecimal) of the special function unit. Therefore, there is the following difference between decimal and hexadecimal numbers.

a) Hexadecimal (H): Specify the upper 2 digits of the head address of special function unit.

b) Decimal (K): Specify a decimal value which has been converted from the upper 2 digits of the head address of special function unit.

Example

If the head address of the special function unit is 130, n1 is as follows.

- H13 when specified in hexadecimal.
- When specified in decimal, calculate according to the following expression.

$$\text{Expression} = 1 \times 16^1 + 3 \times 16^0$$
$$= 16 + 3$$
$$= 19$$

Therefore, n1 is K19 when specified in decimal.

(2) n2: Specify the buffer memory address in the special unit. (For details of the buffer memory, refer to the User's Manual for the special function unit.)

(3) n3: Number of transfers (Number of datas to be written, number of datas to be read)

(a) FROM, TO: The number of transfers is 1 word device (16 bits).

(b) DFRO, DTO: The number of transfers is 2 word devices (32 bits).

6.11 Other Instructions

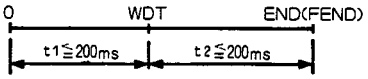
The other instructions include WDT reset, special timing pulse, and ASCII conversion instructions.

6.11.1 WDT reset

1) The WDT reset instruction resets the watch dog timer in the sequence program.

2) Use the WDT reset instruction when the period of time from step 0 to END (FEND) in the sequence program exceeds the set value (200ms) of the watch dog timer depending on conditions.

3) The period from step 0 to the WDT instruction and the period from the WDT instruction to END (FEND) should not exceed 200ms, respectively.



4) The WDT instruction may be used 2 or more times during 1 scan. In this case, note that if an error occurs, it will take time until output is disabled.

Function Symbol	Available Device: Bit device							Word (16-bit) device							Constant	Pointer	Level	Processing Unit				Digit Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
	X	Y	M	L	B	F	T	C	D	W	Z	V	K	H	P	N	1bit	16bits	32bits	Other					
WDT																	—	—	—	—	—	1	—		

Classification	Function Symbol	Symbol	Function	Program Example
WDT reset	WDT	WDT	0 WDT END ⇩ Resets watch dog timer.	While X0 is on, this program resets WDT. Ladder example X0 WDT

6.11.2 Special timing pulse

1) The special timing pulse instruction enables the timing clocks for user (M9020, M9021).

2) The timing pulse enabled by the DUTY instruction is reset by:

   a) executing the DUTY instruction again with n1=0 setting;
   b) performing reset with the reset switch; or
   c) turning off the PC power.

3) When n1 and n2 have been set to 0,

   a) n1=0: The timing pulse remains off.
   b) n1>0, n2=0: The timing pulse remains on.

4) Convert the DUTY instruction command into pulse using PLS instruction. If not converted into pulse, the DUTY instruction command executes the DUTY instruction at every scan during ON. Therefore, proper timing pulse cannot be obtained.

Function Symbol		Available Device: Bit device X	Y	M	L	B	F	Word (16-bit) device T	C	D	W	Z	V	Constant K	H	Pointer P	Level N	Processing Unit 1bit	16bits	32bits	Other	Digit Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
DUTY	n1													○	○			—	—	—	—	—	7	—		—
	n2													○	○											
	D		○															○	—	—	—					

*For device D, M9020 and M9021 may only be specified.

Classification	Function Symbol	Symbol	Function	Program Example
Special timing pulse	DUTY	DUTY n1 n2 D	Step number 0 (1)(2) 0 (1)(2) 0 (1)(2) 0 (1)(2) Command OFF ON M9021 OFF n1 scan n2 scan Remains off until DUTY instruction is executed.	When X0 is enabled, this program enables M9021 5 scans and disables 3 scans. When X1 is enabled, this program disables M9021. Ladder example PLS M0 DUTY K5 K3 M9021 PLS M1 DUTY K0 K2 M9021 1) At the rise (OFF →ON) of X0, M0 is enabled 1 scan. 2) While M0 is on, timing pulse is set. 3) At the rise (OFF →ON) of X1, M1 is enabled 1 scan. 4) While M1 is on, timing pulse is set.

6.11.3 ASCII conversion

1) The ASCII conversion instruction converts the specified ASCII characters into the ASCII code.

2) After conversion, the ASCII code is stored into 4 points beginning with the specified device.

3) ASCII conversion is enabled for the alphanumeric characters and special symbols in the following table. The ASCII codes shown below are expressed in hexadecimal.

Alphanumeric Character	ASCII Code	Alphanumeric Character	ASCII Code	Alphanumeric Character	ASCII Code	Alphanumeric Character	ASCII Code	Alphanumeric Character	ASCII Code
0	30	A	41	K	4B	U	55	SP	20
1	31	B	42	L	4C	V	56	*	2A
2	32	C	43	M	4D	W	57	+	2B
3	33	D	44	N	4E	X	58	–	2D
4	34	E	45	O	4F	Y	59	/	2F
5	35	F	46	P	50	Z	5A	<	3C
6	36	G	47	Q	51			=	3D
7	37	H	48	R	52			>	3E
8	38	I	49	S	53				
9	39	J	4A	T	54				

Function Symbol	Available Device: Bit device						Word (16-bit) device					Constant		Pointer	Level	Processing Unit				Digit Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
		X	Y	M	L	B	F	T	C	D	W	Z	V	K	H	P	N	1bit	16bits	32bits	Other					
ASC	D							○	○	○	○							—	—	—	—	—	13	○		—

*1: W (link register) may be used only for A0J2P23 (R23).

Classification	Function Symbol	Symbol	Function	Program Example
ASCII conversion	ASC	ASC D Alphanumeric character, special symbol	Alphanumeric character Special symbol Head D+0 Lower 8 bits D+1 Upper 8 bits D+2 D+3 Converted into ASCII code	When X0 is enabled, this program converts "ABCDEFGH" into ASCII code and stored into D0 to D3. Ladder example 0 X0 PLS M0 — At the rise (OFF→ON) of X0, M0 is enabled 1 scan. 4 M0 ASC ABCDEFGH D0 — When M0 is enabled, "ABCDEFGH" is converted into ASCII code and stored into D0 to D3. Coding

Step No.	Instruction	Device			
0	LD	X0			
1	PLS	M0			
4	LD	M0			
5	ASC	ABCDEFGH	D0		
18	END				

6.11.4 Print instructions

1) The print instructions output the specified 16-character data from the output unit.

2) The output unit uses serial 10 points, beginning with the specified output number (Y number), to output the following signals. In this case, only multiples of 8 (the lower 1 digit is 0 or 8) may be specified for the starting output number.

   a) Data output: 8 points beginning with the specified output number

   b) Strobe signal: 1 point

   c) Print instruction execution flag: 1 point

3) For the output signal from the output unit, 1 character is sent in 30ms. To send 16 characters, 480ms (=16 x 30) is required. However, data switching and strobe signal ON/OFF during sending are interrupt processings which stop the execution of the sequence program per 10ms. Therefore, sequence processing is continued during data sending with print instruction.

4) The strobe signal indicates the data read timing per 1 character. Therefore, read the data while the strobe signal is on. The strobe signal is a pulse of 10ms ON and 20ms OFF.

5) The print instruction flag is an interlocking signal when several print instructions are used.

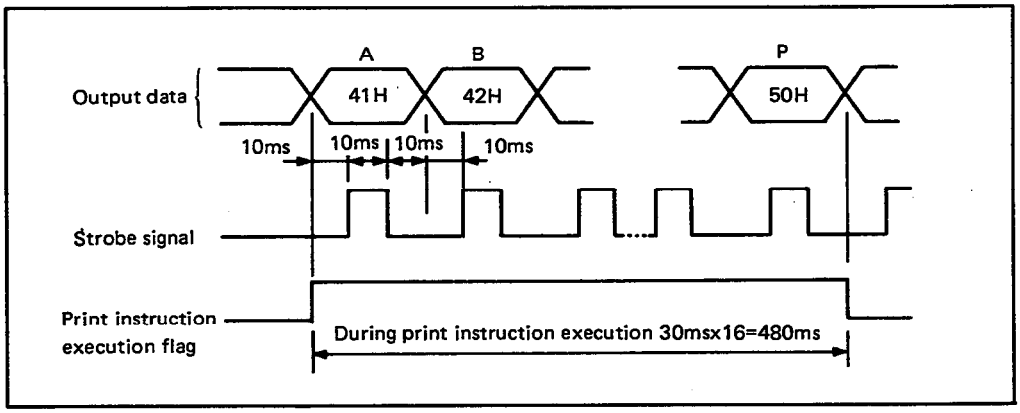


Fig. 7.1 Print Data Output Timing

Function Symbol		Available Device: Bit device X	Y	M	L	B	F	Word (16-bit) device T	C	D	W	Z	V	Constant K	H	Pointer P	Level N	Processing Unit: 1bit	16bits	32bits	Other	Digit Specification	Step Number	Index Qualification	Execution Condition	Carry Flag
PR	S							○	○	○	○							—	—	—	8 points	—	7	—		—
	D		○															—	—	—	10 points					
PRC	S						○											—	—	—	—					
	D		○															—	—	—	10 points					

*1: W (link register) may be used only for A0J2P23 (R23).
*2: When bit device is used, device number to be specified is only 0 or a multiple of 8. (Except F of PRC instruction)

Classification	Function Symbol	Symbol	Function	Program Example
Print instruction	PR	PR S D	Upper 8 bits Lower 8 bits; S+0 B A … S+7 P O; PR instruction; Output unit D+0 … D+9; Head; C B A; Printer indicator; Output data; Strobe signal; PRC instruction; Print instruction execution flag; S A B C D E F G H I J K L M N O; F0 to F95 can be specified.	When X0 is enabled, this program outputs data in D0 to 7 from Y20 through Y2A. Ladder example: 0 X0 PLS M0; 4 M0 Y2A PR D0 Y20
	PRC	PRC S D		

Coding

Step No.	Instruction	Device			
0	LD	X0			
1	PLS	M0			
4	LD	M0			
5	ANI	Y2A			
6	PR	D0	Y20		
13	END				

6.12 BCD 6-digit multiplication/division, BIN 32-bit addition/subtraction/multiplication/division

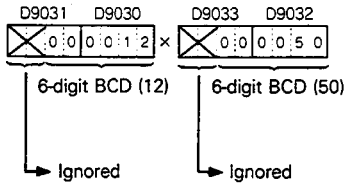
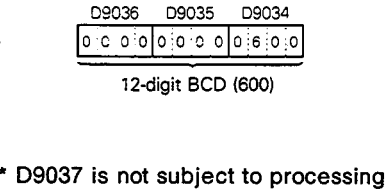
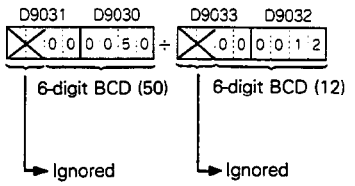
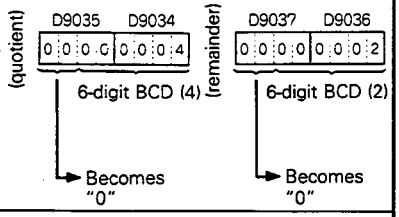
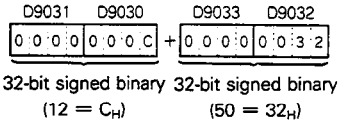
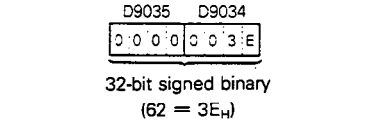
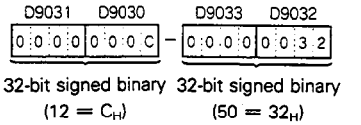
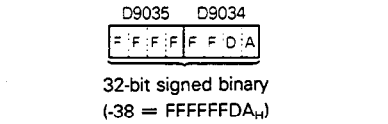
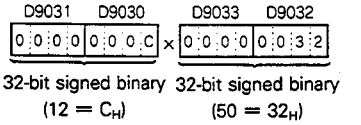
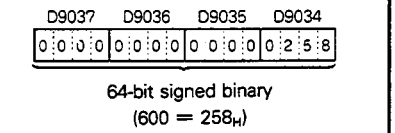
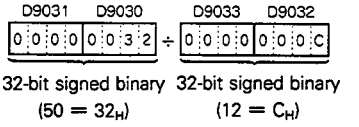
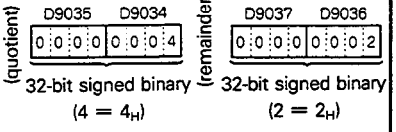
(1) There are no instruction symbols for BCD 6-digit multiplication/division or BIN 32-bit addition/subtraction/multiplication/division: these operations are specified using the format SUBH [_____] (where [_____] is determined in accordance with each instruction).
For details on the SUB instruction, refer to Section 6.6.

(2) Applicable version
The instructions described in (1) above can be used with A0J2 modules of the following software versions or later.
(The software version is indicated on the rating plate of the A0J2CPU.)

Table 6.1 A0J2 Software Versions

CPU Model	S/W Version	
A0J2CPU	808F~~K~~	~~~~ indicates the software version.
A0J2CPU-DC24	808B~~C~~	
A0J2CPUP23	808D~~L~~	
A0J2CPUR23	808G~~L~~	

Table 6.2 Instruction Formats and Processing Details for Additional Instructions

Additional Instructions		Instruction Format	Processing Details	
			Data for Operation	Operation Result
BCD	6-digit multiplication	SUB HFFFF	Operation for 12 x 50 	 * D9037 is not subject to processing and does not change.
	6-digit division	SUB HFFFE	Operation for 50 ÷ 12 	
BIN	32-bit addition	SUB HFFFD	Operation for 12 + 50 	 * D9036 and D9037 are not subject to processing and do not change.
	32-bit subtraction	SUB HFFFC	Operation for 12 - 50 	 * D9036 and D9037 are not subject to processing and do not change.
	32-bit multiplication	SUB HFFFB	Operation for 12 x 50 	
	32-bit division	SUB HFFFA	Operation for 50 ÷ 12 	

(3) Programming method

(a)On execution of the microcomputer program call (SUB H ⬚⬚⬚⬚⬚) after the data for the operation has been stored in D9030 to D9033, the specified operation is executed and the operation result is stored in D9034 to D9037.

(b)The programming procedure is shown Figure 6.2.

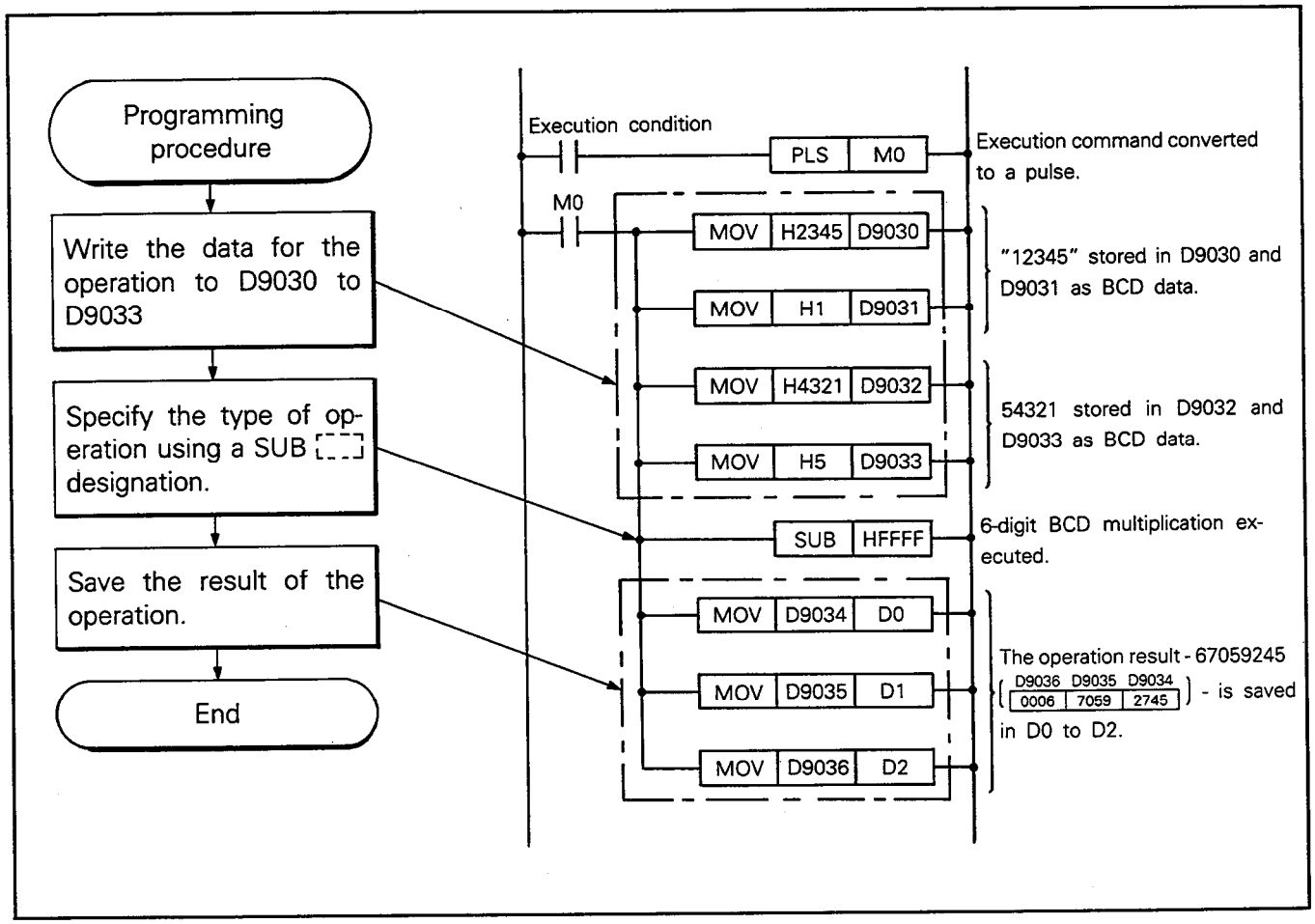


Figure 6.2 Programming Procedure

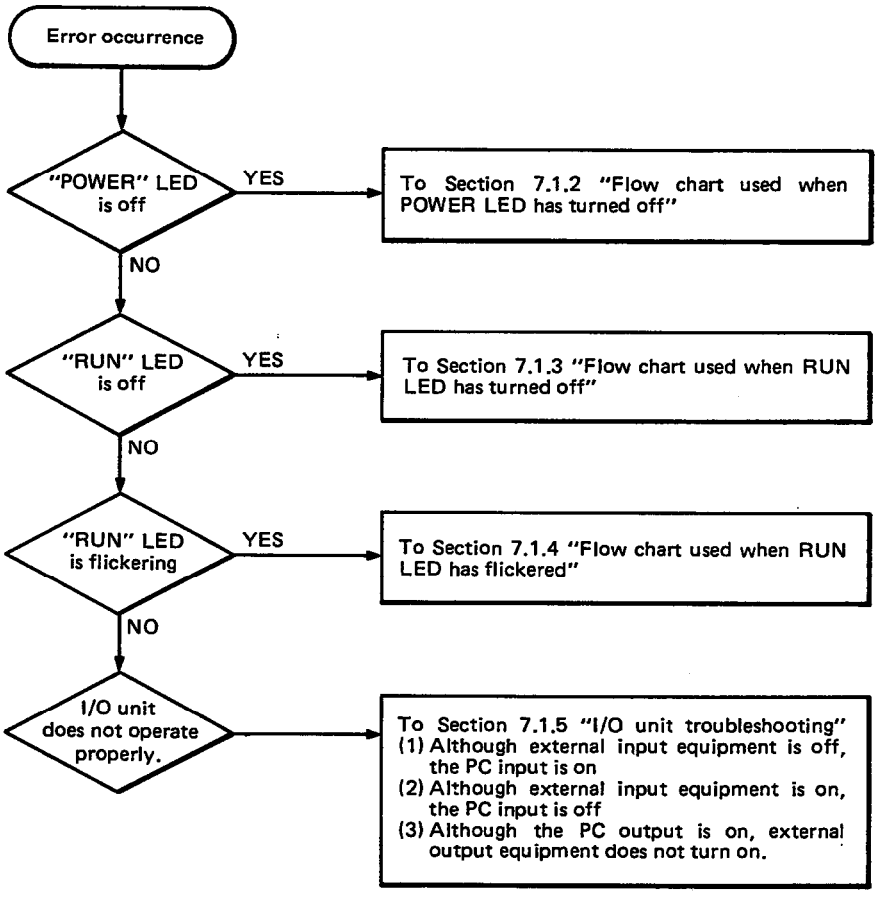
7. TROUBLESHOOTING

7.1 Troubleshooting

This section explains troubleshooting procedures and the definitions and corrective actions of error codes.

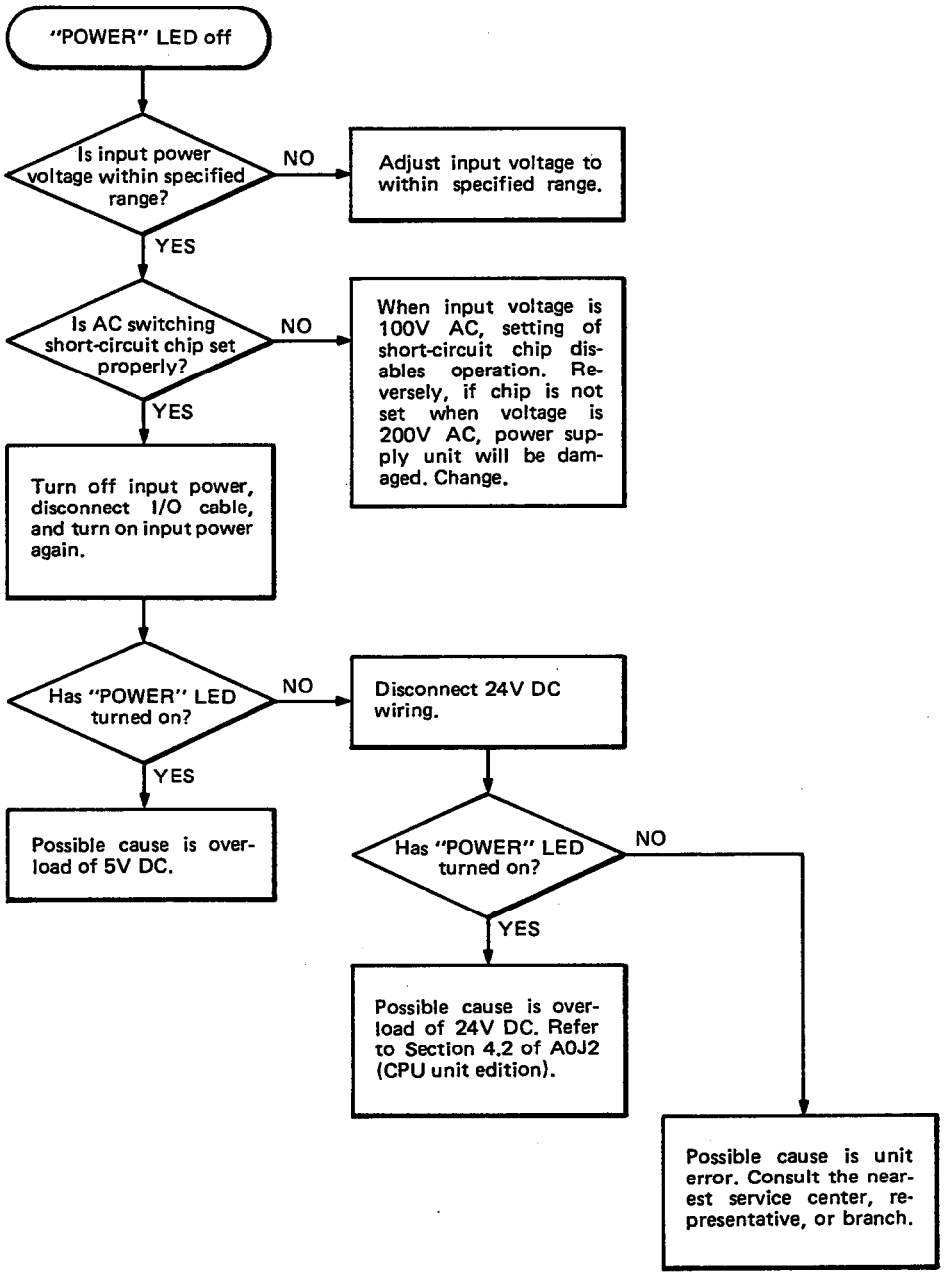
7.1.1 Troubleshooting flow chart

Errors will be explained below, classified by phenomena.



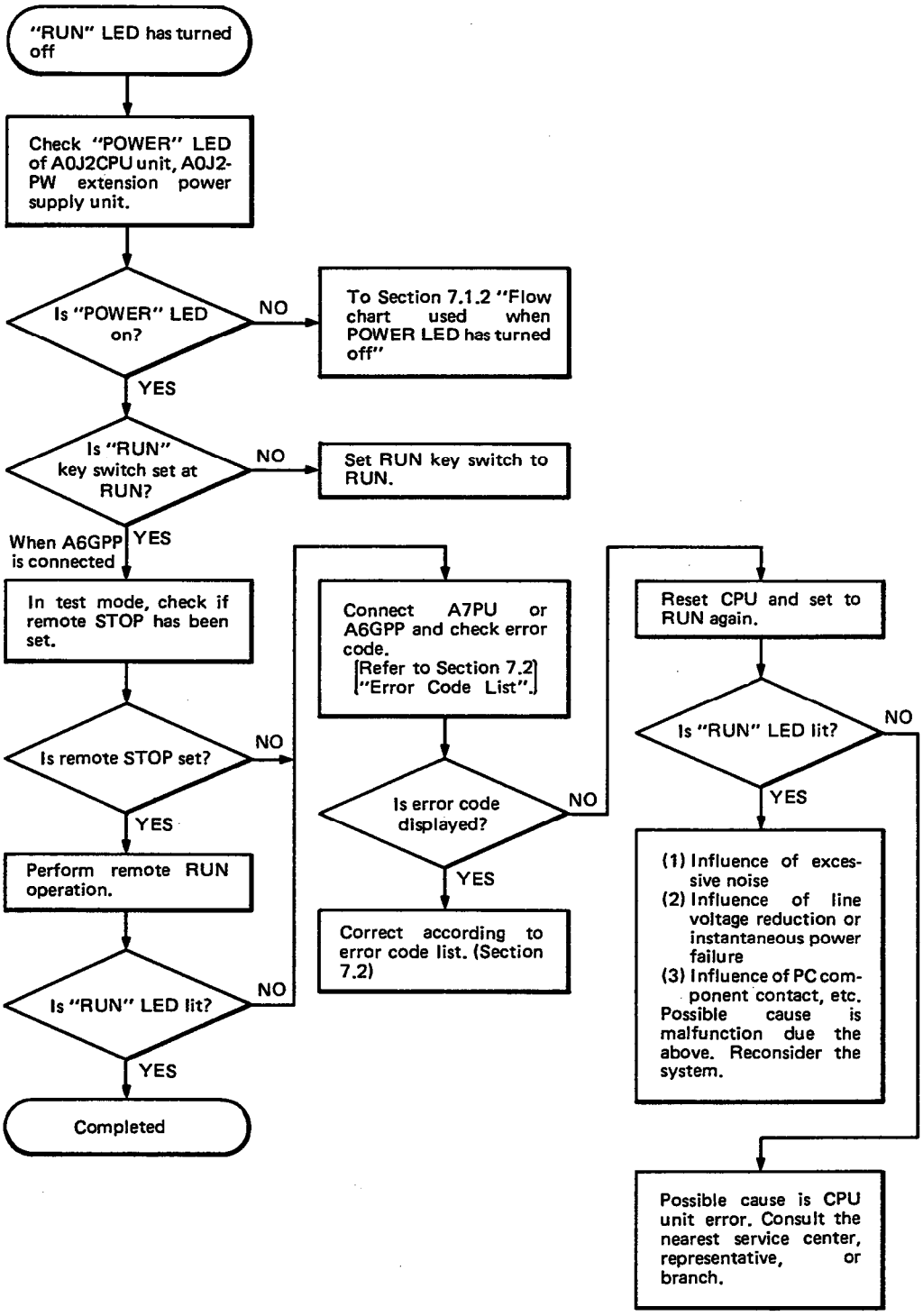
7.1.2 Flow chart used when "POWER" LED has turned off

This section explains the flow chart used when "POWER" LED has turned off at power-on or during operation.



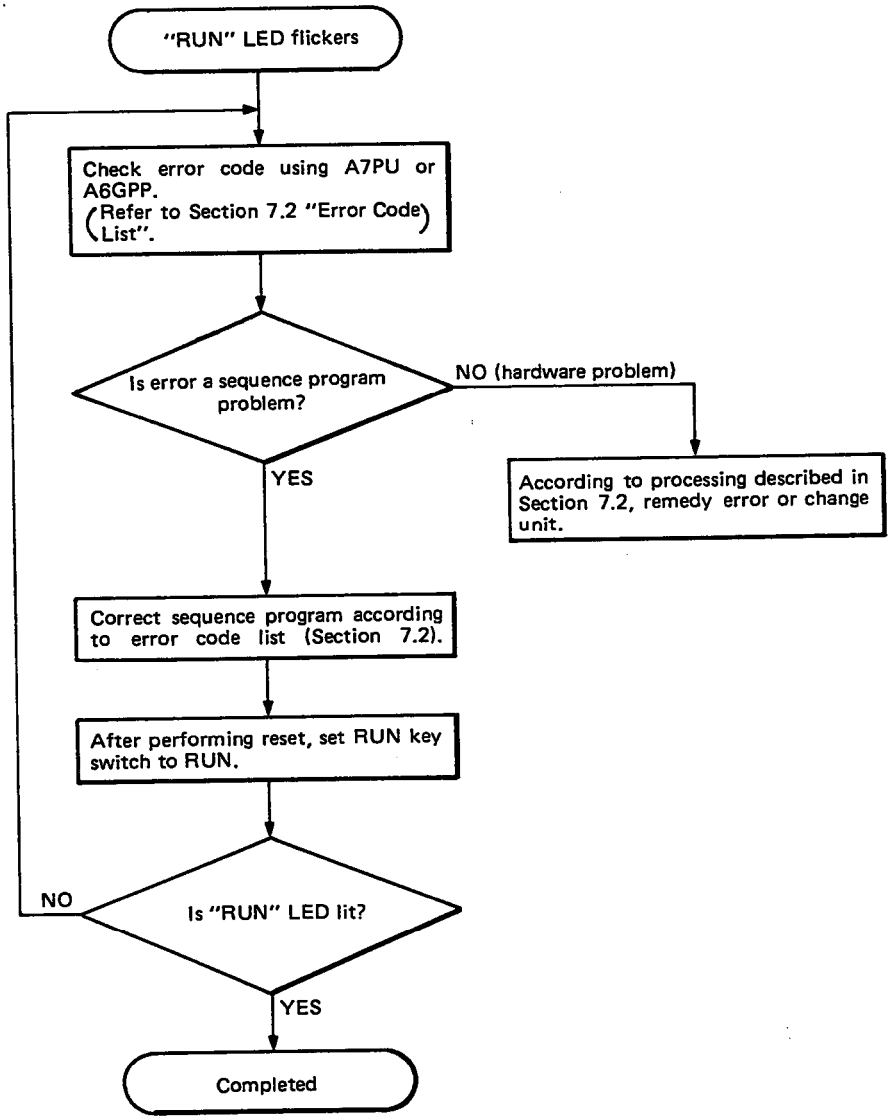
7.1.3 Flow chart used when "RUN" LED has turned off

This section explains the flow chart used when "RUN" LED has turned off during operation.



7.1.4 Flow chart used when "RUN" LED flickers

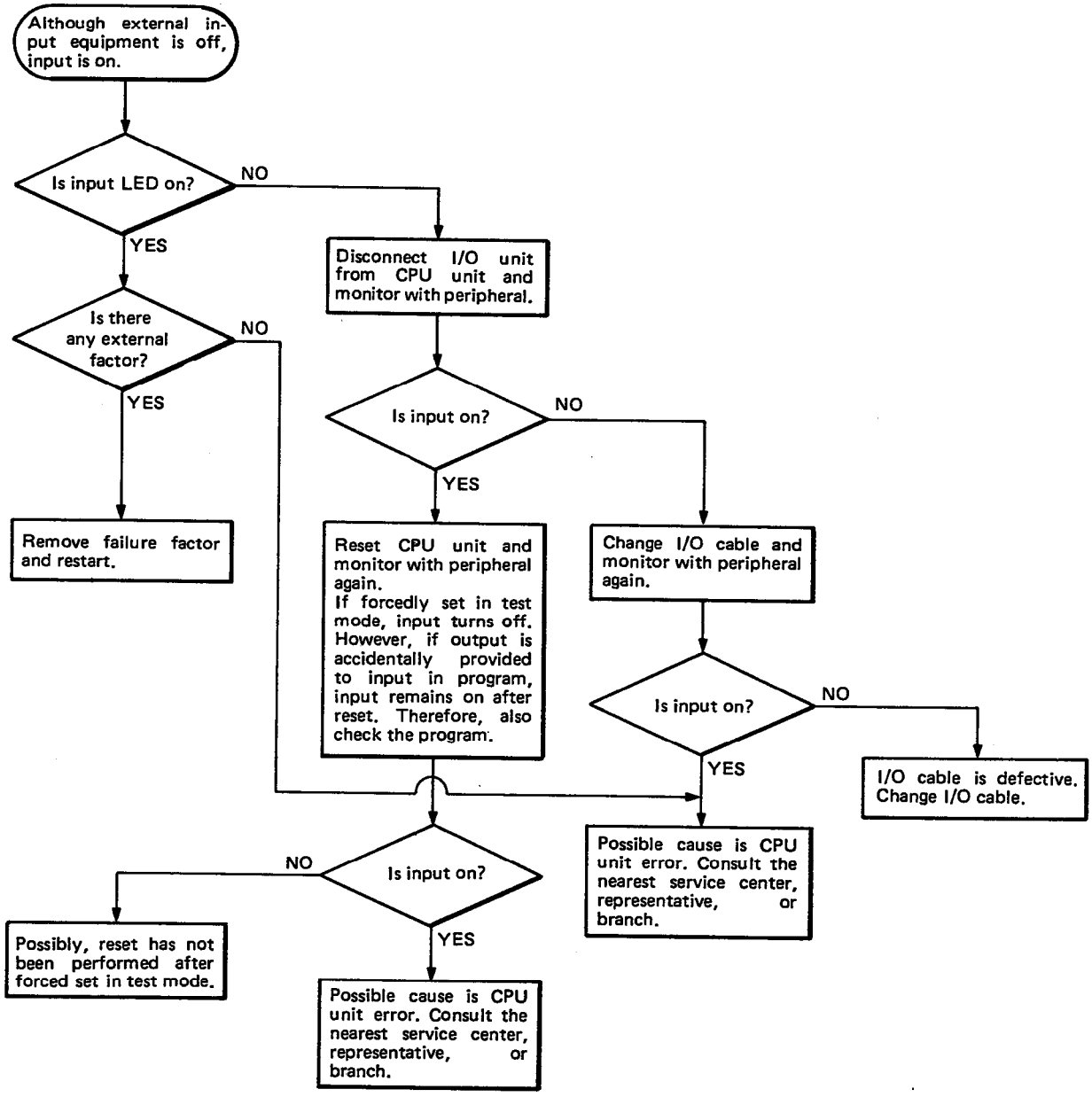
This section explains the flow chart used when "RUN" LED flickers at power-on, at the start of operation, or during operation.



7.1.5 I/O unit troubleshooting

I/O unit troubleshooting will be explained below.

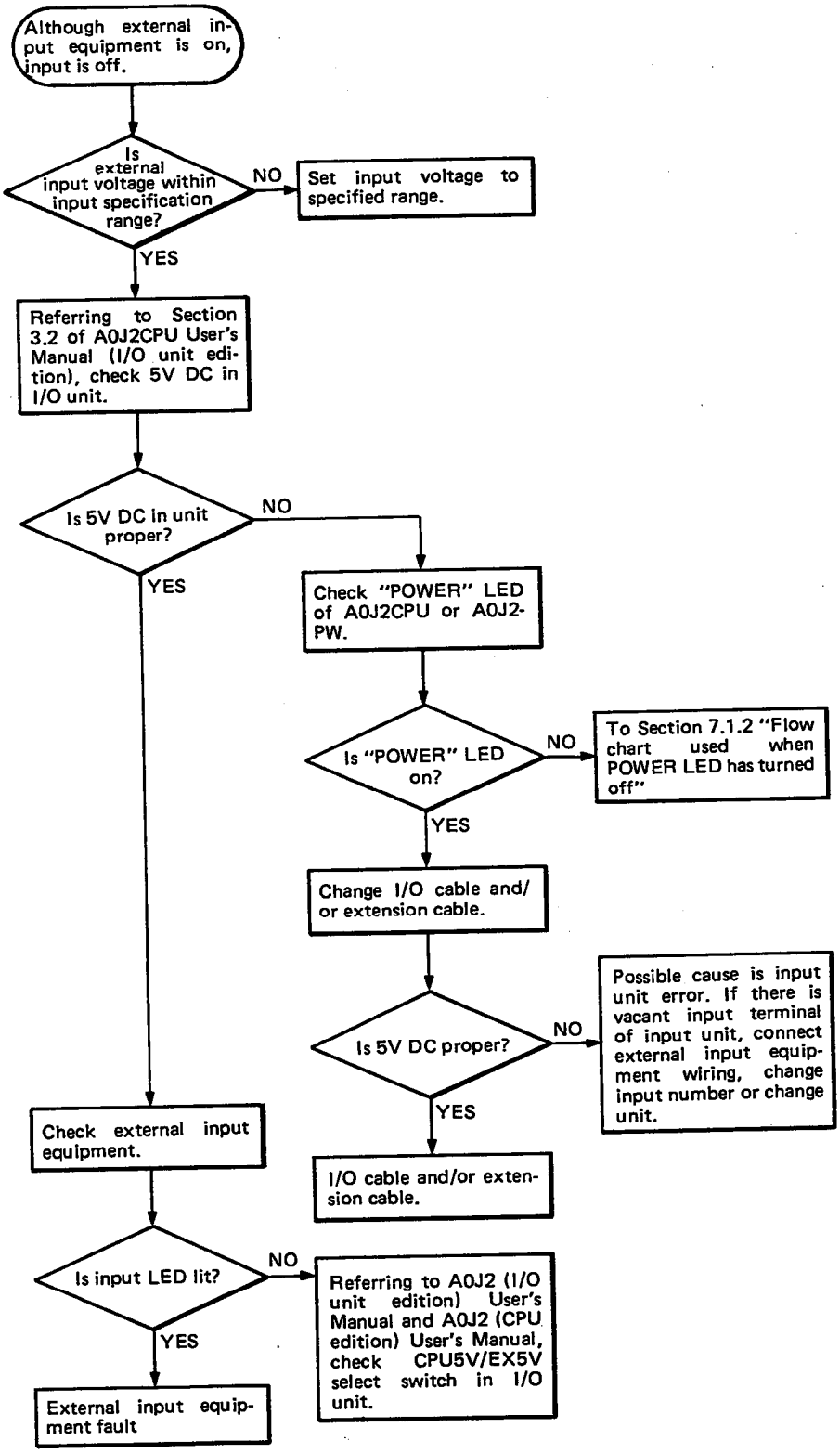
(1) Although external input equipment is off, the PC input is on



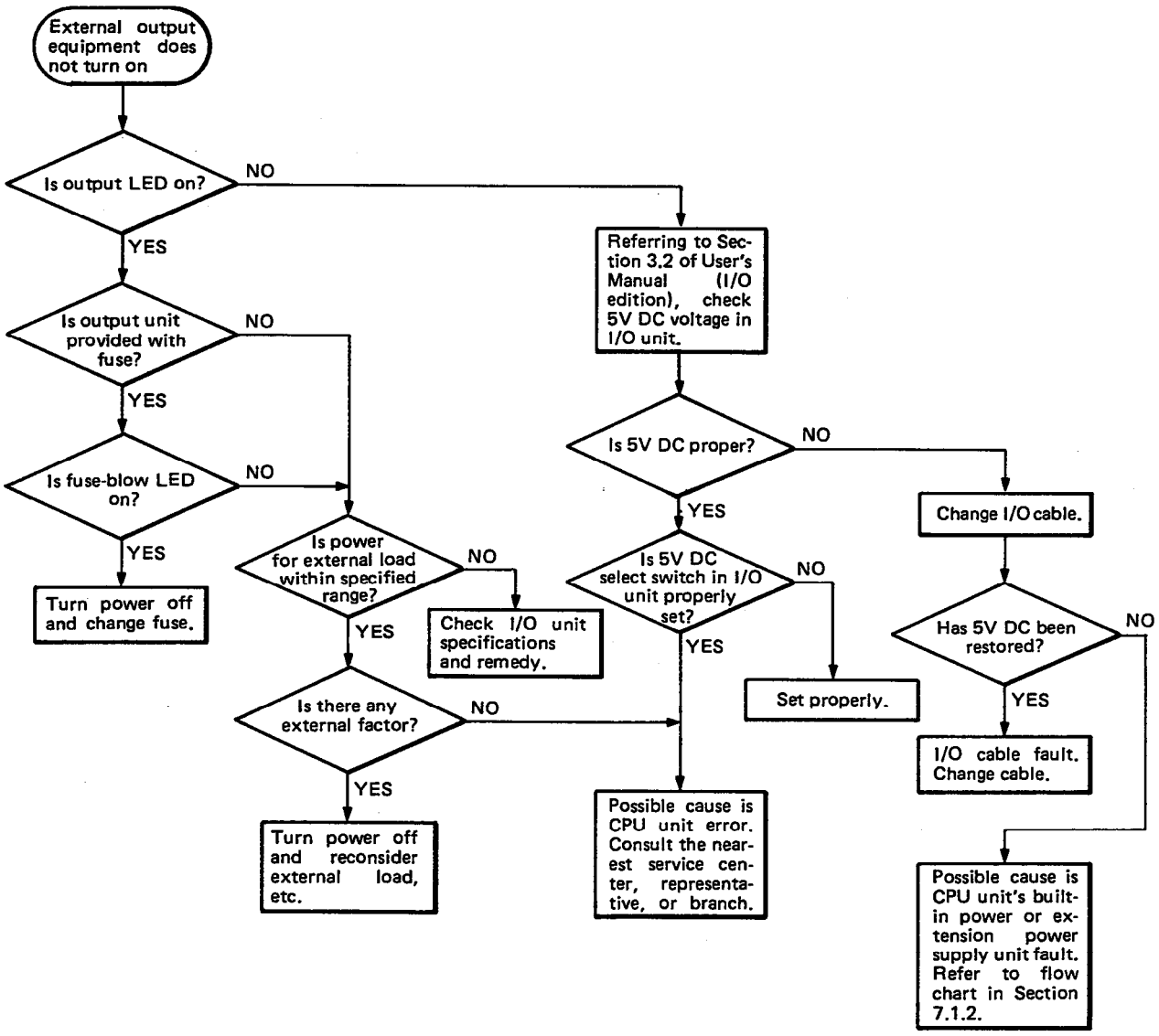
POINT

The external factor indicates external equipment failure, external power fault, wrong wiring, noise, etc.

(2) Although external input equipment is on, the PC input is off



(3) Although the PC output is on, external output equipment does not turn on



7.2 Error Code List

If error has occurred at the start or during PC RUN, the self-diagnostic function displays the error or stores the error code (including the step number) into the special register. Table 7.1 shows the error codes, causes, and corrective actions. Take a proper action to remove the cause.

Error Message	Content of Special Register D9008 (BIN value)	CPU Status	Error and Cause	Corrective Action
"INSTRCT. CODE ERR" (Checked at the execution of instruction)	10	Stop	Instruction code, which cannot be decoded by CPU, is included in the program. (1) ROM including invalid instruction code, has been loaded. (2) Memory contents have been corrected.	(1) Read the error step by use of peripheral equipment and correct the program at that step. (2) In the case of ROM, rewrite the contents of the ROM or change the ROM.
"MISSING END INS." (Checked at STOP → RUN)	12	Stop	(1) There is no END (FEND) instruction in the program.	(1) Write END at the end of the program/subprogram.
"CAN'T EXECUTE (P)" (Checked at the execution of instruction)	13	Stop	(1) There is no jump destination or plural destinations specified by the instruction. (2) Although there is no CALL instruction, the RET instruction exists in the program and has been executed. (3) CJ instruction has been executed with its jump destination located below END instruction. (4) CALL instructions nested.	(1) Read the error step by use of peripheral equipment and correct the program at that step. (Make correction such as the insertion of jump destination or the changing of jump destinations to one.)
"WDT ERROR" (Checked at the execution of END instruction)	22	Stop	Scan time exceeds watch dog error monitor time. (1) Scan time of user program has become excessive. (2) Scan time has lengthened due to instantaneous power failure which occurred during scan.	(1) Check and reduce the user program scan time (to within 200ms) using CJ instruction, etc.

Table 7.1 Error Code List (Continue)

Error Message	Content of Special Register D9008 (BIN value)	CPU Status	Error and Cause	Corrective Action
"END NOT EXECUTE" (Checked at the execution of END instruction)	24	Stop	(1) When the END instruction is executed, another instruction code has been read due to noise, etc. (2) The END instruction has changed to another instruction code for some reason.	(1) Perform reset and run. If the same error is displayed again, it is the CPU hardware error. Therefore, consult nearby service center, representative, or branch.
"FUSE BREAK OFF" (Checked continuously)	32	Run (Stop)	There is an output unit of which fuse has blown.	(1) Check the fuse blow indicator LED of output unit and change the fuse of unit of which LED is on.
"CONTROL-BUS ERR." (Checked at the execution of FROM and TO instructions)	40	Stop	The FROM and TO instructions cannot be executed. (1) Error of control bus with special function unit.	(1) Since this is the special function unit, CPU unit or base unit hardware error. Therefore, change the unit and check the defective unit. For the defective unit, consult nearby service center, representative, or branch.
"SP. UNIT DOWN" (Checked at the execution of FROM and TO instructions)	41	Stop	When the FROM or TO instruction is executed, access has been made to the special function unit but the answer is not given. (1) The accessed special function unit is defective.	Since this is the accessed special function unit error, consult nearby service center, representative, or branch.
"SP. UNIT ERROR" (Checked at the execution of FROM and TO instructions)	46	Stop	(1) Access (execution of FROM to TO instruction) has been made to a location where there is no special function unit.	(1) Read the error step by use of peripheral equipment, and check and correct the content of FROM or TO instruction at that step by use of peripheral equipment.
"OPERATION ERROR" (Checked at the execution of instruction)	50	Run	(1) The result of BCD conversion has exceeded the specified range (9999 or 9999-9999). (2) Setting has been performed exceeding the specified device range and operation cannot be performed.	(1) Read the error step by use of peripheral equipment, and check and correct the program at that step. (Check device setting range, BCD conversion value, etc.)
"BATTERY ERROR"	70	Run	(1) The battery voltage has reduced to less than the specified value. (2) The battery lead is disconnected.	(1) Change the battery. (2) When RAM or power failure compensation is used, connect the battery.

Table 7.1 Error Code List

APPENDIX

1. OPERATION PROCESSING TIME

1) Operation processing time of the A0J2 instructions will be indicated on the following pages.

2) Operation processing time of the basic and application instructions has been measured when both the source (S) and destination (D) are word devices.

3) Operation processing time differ slightly in the following cases. Use the values on the following pages as guide of the operation processing time.

   a) Source and destination datas used for operation
   b) Source and destination devices used for operation

4) When the index qualification is performed, the values on the following pages will be different.

1.1 Sequence Instructions

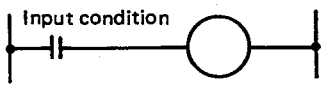
Instruction	Condition (Device)					Processing Time
LD LDI AND	X, Y					5.6
ANI OR ORI	M, L, B, F, T, C					4.4
ANB						4.4
ORB	———					4.4
OUT	Y					5.6
	M, L, B					4.4
	F	When not executed				139
		When executed				152
	T	Instruction execution time				4.4
		Processing time at END	When not executed			6.6
			When executed	After time-out		30
				When added	K	75
					D	85
		Instruction execution time				4.4
		Processing time at END	When not executed			5.5
			When executed	When not count		24
				After count-out		26
				When count	K	67
					D	78
SET	Y	When not executed				5.6
		When executed				5.6
	Special M B	When not executed				12
		When executed				132
	F	When not executed				12
		When executed				148
	M, L	When not executed				4.4
		When executed				4.4

Instruction	Condition (Device)		Processing Time
RST	Y	When not executed	5.6
		When executed	5.6
	M, L B	When not executed	12
		When executed	132
	F	When not executed	12
		When executed	1676
	T, C	When not executed	12
		When executed	220
	D, W V, Z	When not executed	12
		When executed	150
NOP	———		3.7
END	———		1843
MC	M, L	When not executed	155
		When executed	148
MCR	———		239
PLS PLF	M, L	When not executed	154
		When executed: ON	157
		When executed: OFF	157
SFT SFTP	M, L	When not executed	12
		When executed	145
MPS	———		4.4
MRD	———		4.4
MPP	———		4.4

Table 1.1 Sequence Instruction Processing Time (Unit: μs)

POINT

(1) "When not executed" in the above table indicates that the input condition is off.



(2) "When not counted" of OUT C instruction indicates that the input condition remains on and the counter does not count.

(3) "OFF" of PLS and PLF instructions indicates that the input condition remains on 1 scan after it has turned on (off for PLF), and the pulse is not generated.

1.2 Basic and Application Instructions

(1) Table 1.2 shows processing time for the execution of the basic and application instructions.

(2) When the basic and application instructions are not executed, processing time is obtained by the following expression.

> Processing time at no execution
> = 4.4 x (number of instruction steps) μS

(Unit: μs)

Classification	Instruction	Processing Time	Classification	Instruction	Processing Time
Comparison instruction	LD= S1 S2	344	BIN arithmetic operation instruction	+ S D	274
	AND= S1 S2	335		− S D	278
	OR= S1 S2	340		* S1 S2 D	424
	LD<> S1 S2	339		/ S1 S2 D	654
	AND<> S1 S2	339		INC D	172
	OR<> S1 S2	329		DEC D	172
	LD> S1 S2	339	BCD arithmetic operation instruction	DB+ S D	344
	AND> S1 S2	339		DB− S D	344
	OR> S1 S2	329	BCD ↔ BIN conversion instruction	BCD S D	299
	LD<= S1 S2	399		DBCD S D	309 to 1889
	AND<= S1 S2	334		BIN S D	379
	OR<= S1 S2	334		DBIN S D	409 to 1119
	LD< S1 S2	339	Data transfer instruction	MOV S D	249
	AND< S1 S2	339		FMOV S D n	n=30: 744; n=76: 1624
	OR< S1 S2	329			
	LD>= S1 S2	339			
	AND>= S1 S2	329			
	OR>= S1 S2	339			

Table 1.2 Basic and Application Instruction Processing Time (Continue)

(Unit: μs)

Classification	Instruction	Processing Time	Classification	Instruction	Processing Time
Program branch instruction	CJ P**	103	Data processing	SUM S	359
	FEND	1789		DECO S D n	n=2: 396; n=8: 504
	CALL P**	163		ENCO S D n	n=2: 784; n=8: 2684
	RET	109	Special function unit instruction	FROM n1 n2 D n3	530 to 1130
	SUB n	When not executed: 11		DFRO n1 n2 D n3	530 to 1130
	COM	279		TO n1 n2 S n3	530 to 1130
Logical operation instruction	WAND S D	274		DTO n1 n2 S n3	530 to 1130
	WOR S D	274	Others	WDT	97
	WXOR S D	274		DUTY n1 n2 D	399
	WXNR S D	274		ASC ASCII characters D	549
	NEG D	174		PR S D	289
Shift instruction	BSFR D n	n=5: 349; n=15: 554		PRC S D	254
	DSFR D n	n=5: 364; n=15: 424			
	BSFL D n	n=5: 369; n=15: 434			
	DSFL D n	n=5: 359; n=15: 424			

Table 1.2 Basic and Application Instruction Processing Time

IMPORTANT

Design the configuration of a system to provide an external protective or safety inter locking circuit for the PCs.

Under no circumstances will Mitsubishi Electric be liable or responsible for any consequential damage that may arise as a result of the installation or use of this equipment.

All examples and diagrams shown in this manual are intended only as an aid to understanding the text, not to guarantee operation. Mitsubishi Electric will accept no responsibility for actual use of the product based on these illustrative examples.

Owing to the very great variety in possible applications of this equipment, you must satisfy yourself as to its suitability for your specific application.

type A0J2

Programming Manual

MODEL	A0J2-PROGRAM-E
MODEL CODE	13J751
IB(NA)66057-C(9503)MEE	

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : MITSUBISHI DENKI BLDG MARUNOUCHI TOKYO 100-0005 TELEX : J24532 CABLE MELCO TOKYO
NAGOYA WORKS : 1-14 , YADA-MINAMI 5 , HIGASHI-KU, NAGOYA , JAPAN