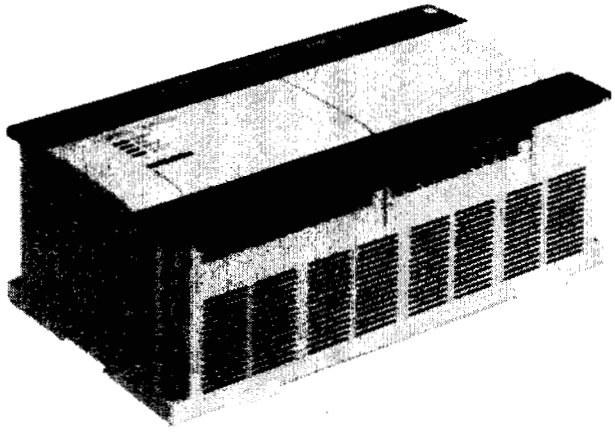
MITSUBISHI

PROGRAMMABLE CONTROLLERS
MELSEC-F

HANDY MANUAL



FX1

FX1 SERIES PROGRAMMABLE CONTROLLERS

(Japanese Specification)

HANDY MANUAL

FOREWORD

- This manual is a comprehensive manual which includes information on the hardware and installation as well as programming information of the Mitsubishi FX1 series programmable controllers sold in Japan.
- Users should ensure that the details of this manual is studied and understood before attempting to install or use the units.
- Information concerning the programming of the system and the use of the programmers is covered in other manuals.

CONTENTS

1 **GENERAL HARDWARE AND INSTALLATION** 1

1.1 OVERVIEW OF THE FX1 2
1.2 PERIPHERAL DEVICES 6
1.3 MODEL NAME 8
1.4 WORLD SPEC. AND JAPANESE SPEC. TYPES 9
1.5 INSTALLATION 10
1.6 POWER SUPPLY CIRCUIT 12
1.7 INPUT CIRCUIT CONNECTION 14
1.8 OUTPUT CIRCUIT CONNECTION 16
1.9 PRELIMINARY CHECKS 22
1.10 ERROR CHECK 23
1.11 MAINTENANCE AND INSPECTION 26
1.12 EXTERNAL DIMENSIONS AND GENERAL SPECIFICATIONS 27
1.13 TERMINAL LAYOUT 28
1.14 PERFORMANCE SPECIFICATIONS LIST 31
1.15 DEVICE NUMBER LIST 32

2 **BASIC SEQUENCE INSTRUCTIONS** 33

2.1 RELAYS AND TIMERS 34
2.2 LOGICAL LOAD AND OUT COILS 36
2.3 SERIAL CONTACTS 38
2.4 PARALLEL CONTACTS 39
2.5 PARALLEL CONNECTION OF SERIAL CIRCUIT BLOCKS 40
2.6 SERIAL CONNECTION OF PARALLEL CIRCUIT BLOCKS 41
2.7 MULTIPLE OUTPUT CIRCUITS 42
2.8 COMMON SERIAL CONTACTS 44
2.9 RETAINING AND RESETTING OPERATION STATUS 46
2.10 COUNTERS AND TIMERS 47
2.11 PULSE OUTPUT 48
2.12 NO PROCESSING INSTRUCTION 49
2.13 PROGRAM END 50

3 **STEP LADDER INSTRUCTIONS** 51

3.1 HOW TO READ SEQUENTIAL FUNCTION CHARTS 52
3.2 PROGRAMMING 58
3.3 STATE FUNCTION RULES 66
3.4 OPERATION MODES 68

4 **DEVICE DETAILS** 75

4.1 INPUT/OUTPUT RELAYS 76
4.2 AUXILIARY RELAYS 78
4.3 STATES 80
4.4 POINTERS (P, I) 82
4.5 CONSTANTS 83
4.6 TIMERS 84
4.7 COUNTERS 86
4.8 DATA REGISTERS 94
4.9 INDEX REGISTERS 95
4.10 SUMMARY OF MEMORY ALLOCATIONS 96

5 **APPLIED INSTRUCTION** 97

5.1 GENERAL RULES OF APPLIED INSTRUCTIONS 98
5.2 PROGRAM FLOW (FNC 00 TO FNC 09) 102
5.3 MOVE AND COMPARE (FNC 10 TO FNC 19) 110
5.4 ARITHMETIC AND LOGICAL OPERATIONS (FNC 20 TO FNC 29) 114
5.5 DATA OPERATION (FNC 40 TO FNC 49) 122
5.6 HIGH-SPEED PROCESSING (FNC 50 TO FNC 59) 124
5.7 HANDY INSTRUCTIONS (FNC 60 TO FNC 69) 128
5.8 EXTERNAL FX SERIAL DEVICES (FNC 80 TO FNC 89) 129
5.9 LIST OF INSTRUCTION MNEMONICS 130

6 **SPECIAL DEVICE AND INSTRUCTION LISTS** 131

6.1 SPECIAL DEVICES 132
6.2 INSTRUCTION AND EXECUTION TIME LIST 134

7 **USING SPECIAL UNITS EFFECTIVELY** 151

7.1 FX-232AW INTERFACE UNIT 152
7.2 FX-8AV VOLUME ADAPTER 153

8 APPENDIX 155

8.1 2-INPUT RUN/STOP SWITCH CONTROL 156

8.2 PROGRAM FLOW CONTROL INSTRUCTIONS 157

1	GENERAL HARDWARE AND INSTALLATION
2	BASIC SEQUENCE INSTRUCTIONS
3	STEP LADDER INSTRUCTIONS
4	DEVICE DETAILS
5	APPLIED INSTRUCTION
6	SPECIAL DEVICE AND INSTRUCTION LISTS
7	USING SPECIAL UNITS EFFECTIVELY
	APPENDIX

1. GENERAL HARDWARE AND INSTALLATION

1.1 OVERVIEW OF THE FX1

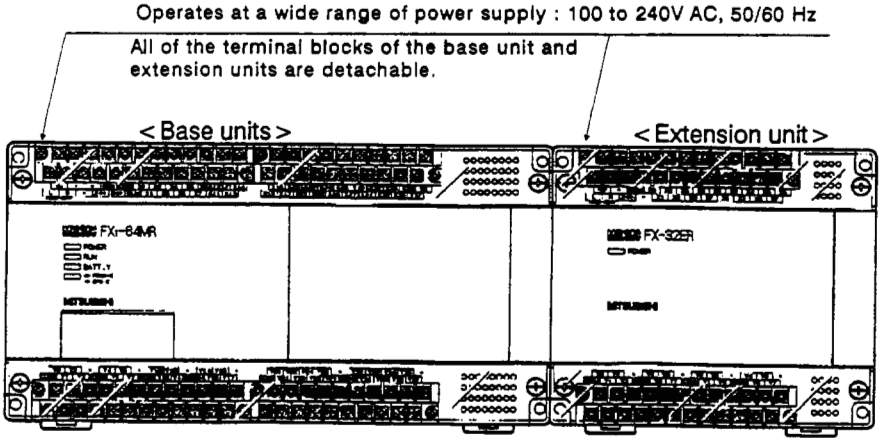
This section provides general information to help users select the modules necessary to configure the required system.

1.1.1 SELECTION BASED ON I/O POINTS

The FX1 Series Programmable Controller (PC) is configured from four products :
a base unit, extension units, extension blocks, and special adapters.

A variety of systems can be designed by combining these products or by using the base unit only.

A base unit, when combined with extension units and/or extension blocks, can control up to 128 I/O (input/output) points.



(1) OUTPUT TYPE SELECTION

Relay output (R) : Contact output; used to drive AC or DC loads.
SSR output (S) : Non-contact output; used to drive AC loads.
Transistor output (T) : Non-contact output; used to drive DC loads.

(2) INTERNAL POWER SUPPLY

Both base units and extension units have a built-in 24V DC power supply for the input circuits.

(3) AVAILABLE MODELS

Unit	I/O Points	Output Type	
		Relay Output	Transistor Output
Base Unit	8/8	FX1-16MR	FX1-16MT
	12/12	FX1-24MR	FX1-24MT
	16/16	FX1-32MR	FX1-32MT
	24/24	FX1-48MR	FX1-48MT
	32/32	FX1-64MR	FX1-64MT
	40/40	FX1-80MR	FX1-80MT
Extension Unit	16/16	FX-32ER	–
	24/24	FX-48ER	FX-48ET

Base unit (M) : Contains the CPU and memory. A base unit must always be used.
Extension unit (E) : A module(s) used to expand the number of I/O points.

Note that some models may not be available in some countries. The model names in this and the following page have not been written in full since there are many model variations. See page 8 for the explanation of the model name.

1. GENERAL HARDWARE AND INSTALLATION

OVERVIEW OF THE FX1

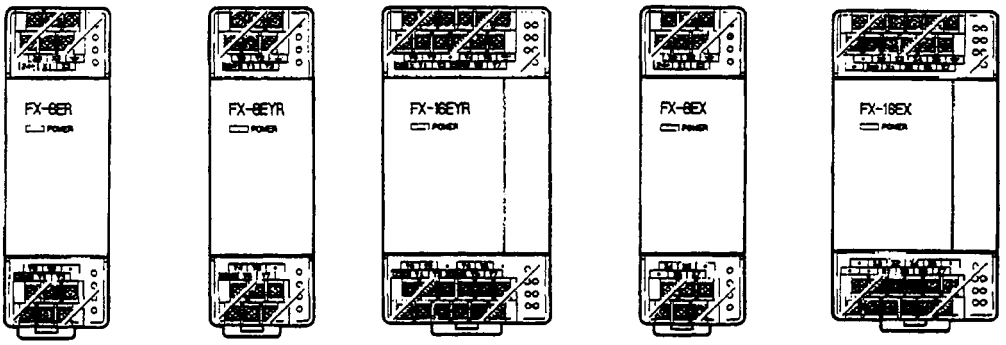
1.1.2 EXTENDING I/O POINTS

Each extension block adds in units of 8 I/O points. Both input and output points can be increased independently by selecting the appropriate input or output extension blocks.

Extension blocks differ from base units and extension units in that they do not contain a built-in power supply. Power must be supplied from the base unit or an extension unit. Also, unlike base units and extension units, extension blocks have fixed terminal blocks.

FX-8ER (combination block) uses 4/4 points of 8/8 I/O points, leaving the remaining 4/4 points vacant. Similarly, FX1-24MR leaves 4/4 points of 16/16 I/O points vacant. Therefore, when these are used, the maximum I/O combination will be less than 128 points.(with V2.0,128I/O points can be attained even with FX-8ER)

<Extension blocks>



(1) OUTPUT TYPE

Extension blocks provide the same output types as those provided by the base unit and extension units. Different output types can be combined.

For example, a base unit with SSR output, an extension unit with relay output, and an extension block with transistor output can be configured together.

(2) POWER SUPPLY

Extension blocks do not include built-in power supplies. Power must be supplied from the base unit or an extension unit.

(3) AVAILABLE MODELS

Block	I/O Points	Output Type		
		Relay Output	SSR Output	Transistor Output
Combination	4/4	FX-8ER	–	–
Output	0/8	FX-8EYR	FX-8EYS	FX-8EYT
	0/16	FX-16EYR	FX-16EYS	FX-16EYT
Input	8/0	FX-8EX		
	16/0	FX-16EX		
Special	16/8	FX_2-24EI For connecting F2 Series special-function modules		

Note: ●*The number of I/O points of a 16/24/32-point base unit and 32-point extension units can be expanded to up to a further 16 points with extension blocks. (When the FX_2-24EI is used, only 8 I/O points can be expanded.)*

●*The number of I/O points of a 48/64/80-point base unit and 48-point extension units can be expanded to up to 32 points with extension blocks. (When the FX_2-24EI is used, only 24 I/O points can be expanded.)*

1. GENERAL HARDWARE AND INSTALLATION

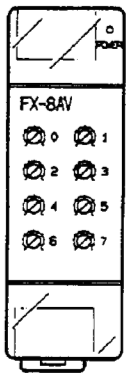
OVERVIEW OF THE FX1

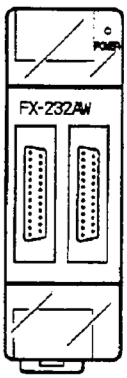
1.1.3 SPECIAL FUNCTION ADAPTERS

The FX2 Series provides the following special function adapters. Connect a special function adapter to the special port on the left side of the base unit.

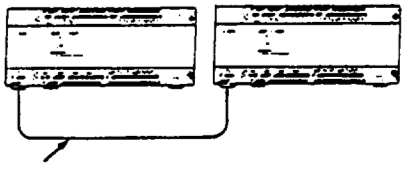
The FX-232AW interface module can be used in combination with other special function adapters.

Special function adapters do not contain a built-in power supply. Power must be supplied from the base unit.





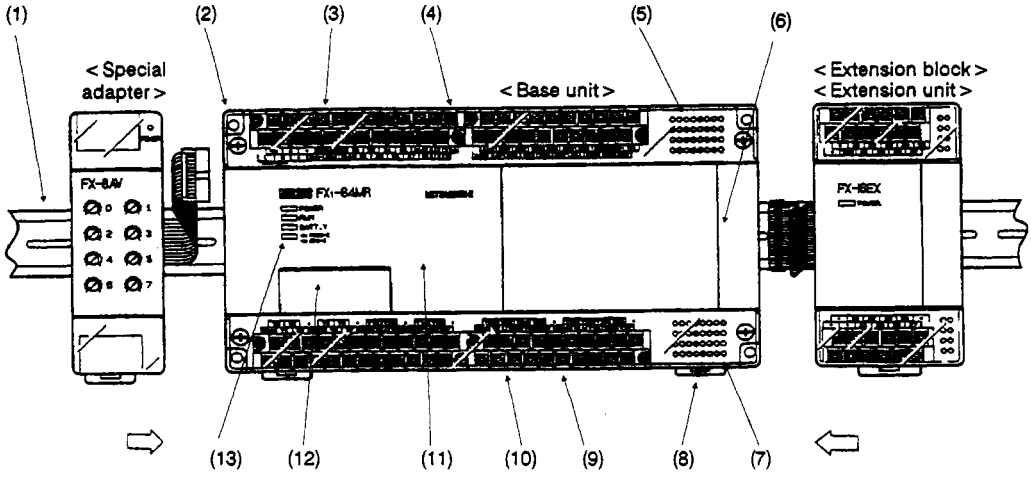
(1) AVAILABLE MODELS

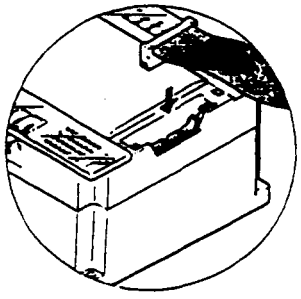
Model	Description
FX2-40AP Optical link Adapter FX2-40AW Wire link Adapter	 These two adapters enable the communication of 100 ON/OFF signals and 10 points of 16-bit data between two base units. Fiber-optic or twisted-pair cable
FX-8VA Volume adapter	The potentiometers on this adapter provide 8 points of control for analog timers, mode setting, or rotary switches.
FX-232AW I/F Module	The I/F(interface) module is connected to the base unit programming connector. The RS422 signals, isolated by the photocoupler, are converted by the module into RS232C signals.

1. GENERAL HARDWARE AND INSTALLATION

OVERVIEW OF THE FX1

1.1.4 PRODUCT OUTLINE



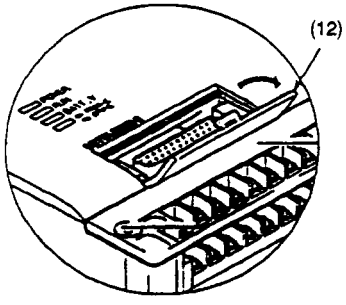


The base unit, extension units, extension blocks, and adapters/modules can be mounted to the DIN rail (1) or mounted directly using the mounting holes (2). Pull the hook (8) to mount a unit to the DIN rail.

Extension units and extension blocks are connected to the base unit by opening the connector cover (6) and connecting the cable as illustrated.

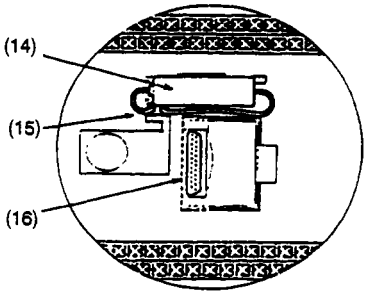
Connect the special adapter (with an exception of FX-232AW) to the connector on the left side of the base unit.

The input terminal block (3) and output terminal block (9) are protected by transparent terminal covers, (4) and (10) respectively. Input status indicators (5) and output status indicators (7) can be observed through the cover. The base unit is provided with POWER, RUN, BATT.V, and CPU-E status indicators (13).



When writing or monitoring a program, connect the FX-20P-E HPP to the connector under the connector cover (12).

The removable F2-40BL lithium battery (14) is fitted to the connector (15) under the panel cover (11). The following memory cassettes can be installed in the memory cassette connector (16).

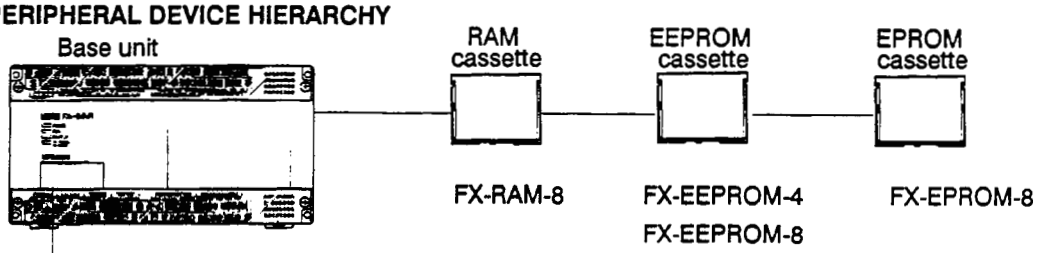


Memory Type	4K Steps	8K Steps	Remarks
RAM Cassette	–	FX-RAM-8	Back-up battery required
EEPROM Cassette	FX-EEPROM-4	FX-EEPROM-8	–
EPROM Cassette	–	FX-EPROM-8	ROM writer/eraser required

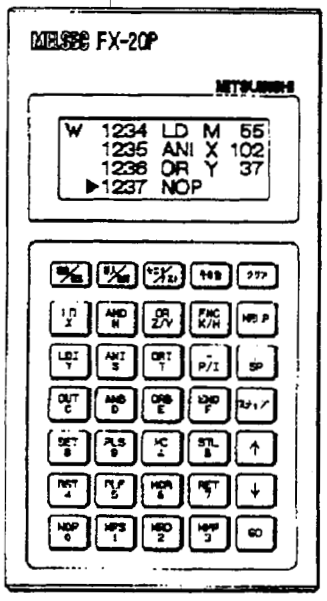
1. GENERAL HARDWARE AND INSTALLATION

1.2 PERIPHERAL DEVICES

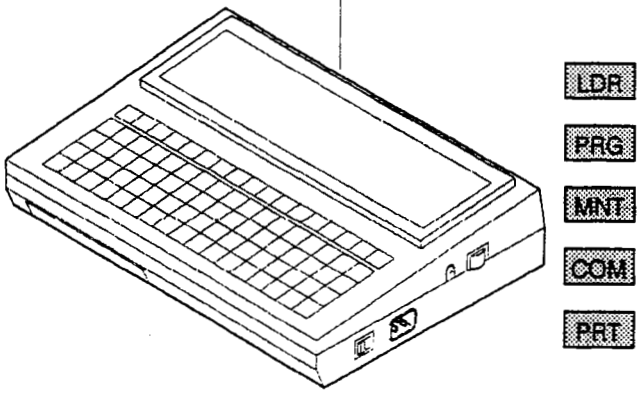
1.2.1 PERIPHERAL DEVICE HIERARCHY



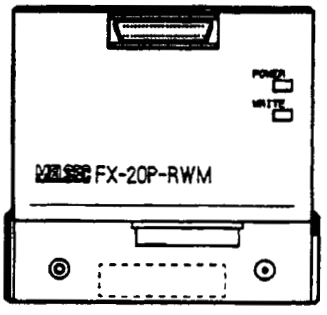
Note: *•EEPROM and RAM cassettes may cause PROG.E error to occur when first inserted into the PC. Deleting their contents will remove this error.*



FX-20P HPP
Handy programming panel



GP-80 HGP
Handy graphic programming panel.
GP-80FX-KIT (Japanese)
GP-80FX-E-KIT(English)
Up-grade is necessary.
(Note programming is up to 4K only and does not allow the use of the GP-80 ROW ROM writer.)



FX-20P-RWM
ROM writer module

1. GENERAL HARDWARE AND INSTALLATION

PERIPHERAL DEVICES

Floppy disk

PRT Printer output function

PRG Programming function

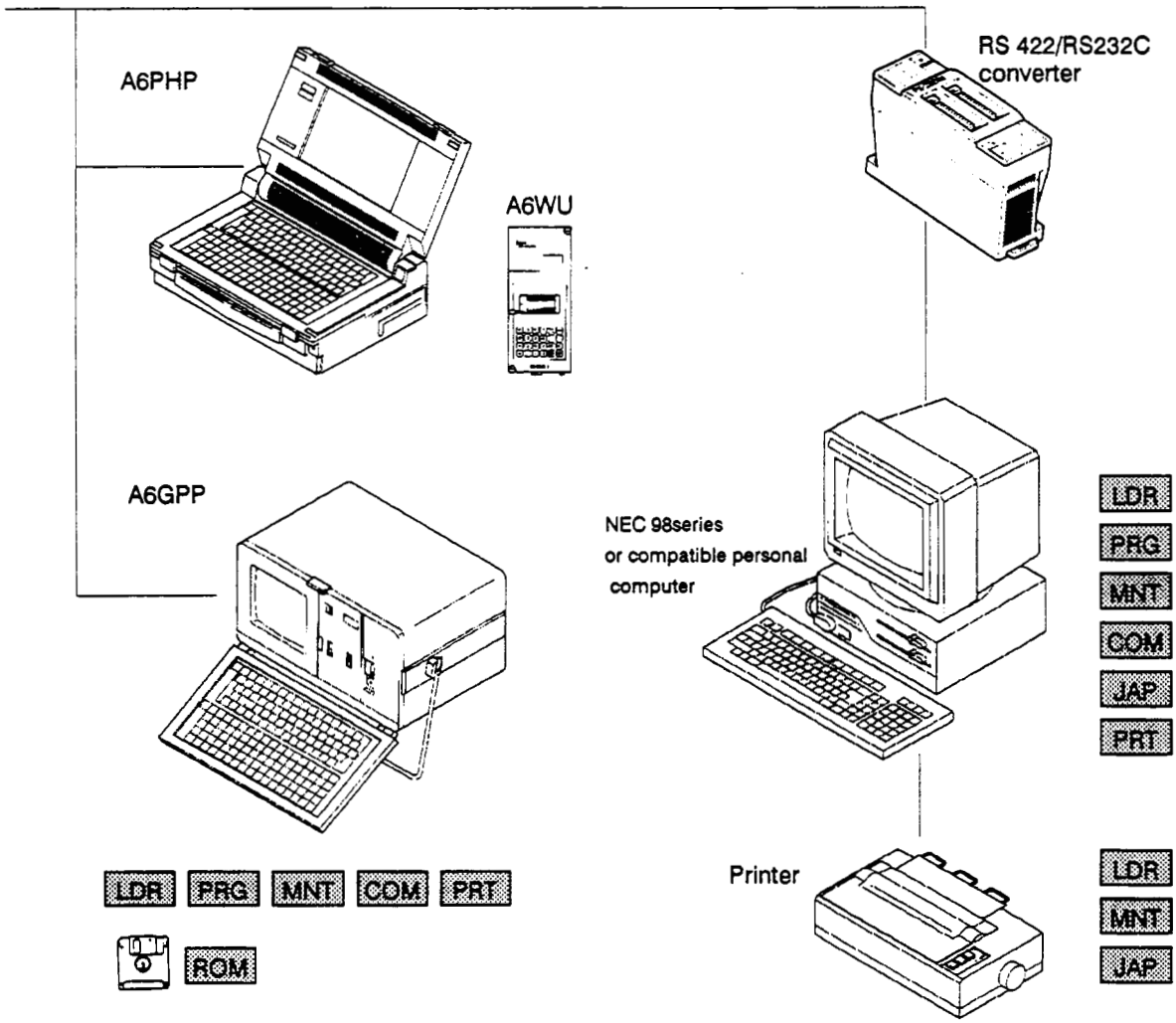
MNT Monitor function

ROM ROM writer function

LDR Circuit display function

COM Comment (alphanumeric)

JAP Japanese Kanji



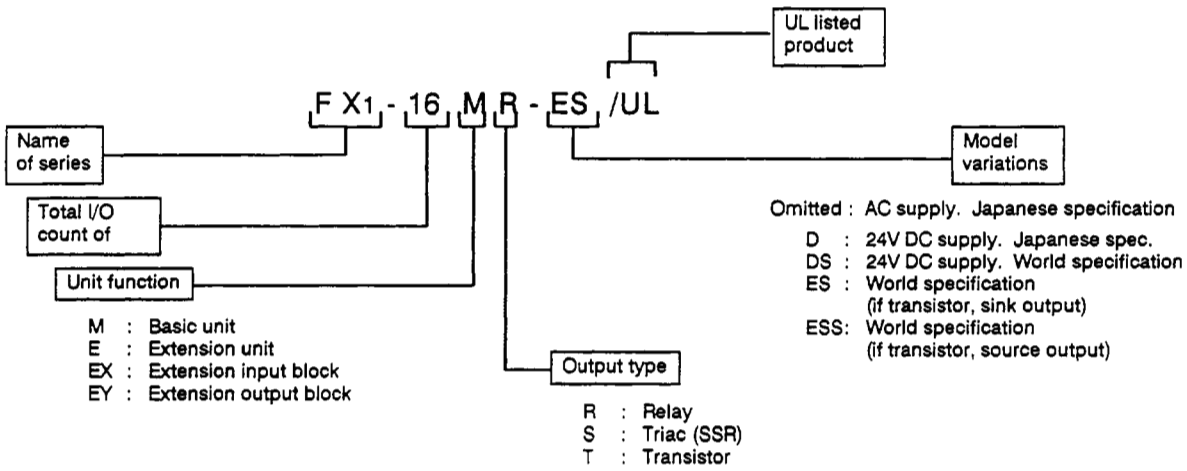
FX-A6GPP-KIT Japanese software may be used.

FX-PCS-KIT/98 Japanese software may be used.

1. GENERAL HARDWARE AND INSTALLATION

1.3 MODEL NAME

GENERAL FORMAT



EXCEPTIONS :

The mixed I/O Extension Block FX-8ER and some special adaptor blocks are slightly different to the above rule.

Examples :

FX-8ER..... 4 input, 4 output relay	Extension Block
FX_2-40AP..... Parallel link (optical)	Adaptor Block
FX-8AV..... Analog setting input	Adaptor Block

etc...

Note 1: *•Unless otherwise stated, when model names are written in this manual without the model variation part, it implies that all model variation types are valid to the information written. Please ensure that the correct model is referred to in order to avoid any disastrous errors. Note that in some countries, some of the above mentioned models may not be available.*

Note 2: *•World specification indicates that the unit can be used around the world because it has a wide supply voltage tolerance and the inputs can be configured for sink or source connection methods by the user.*

1. GENERAL HARDWARE AND INSTALLATION

1.4 WORLD SPEC. AND JAPANESE SPEC. TYPES

WORLD SPEC.

These are designed to be used throughout the world and are sold outside Japan.

DISTINGUISHABLE FEATURES

- The inputs are configurable (via the [S/S] terminal) for [+] common (source) or [–] common (sink) connections.
- The outputs may be of sink or of source connection depending on the model variation.
- The model name indicates world specification by an "E" or by otherwise in the model variation part of the model name.
  example: FX-24MR-ES

JAPANESE SPECIFICATION

These are designed to be sold in Japan but they are sometimes exported to elsewhere in the world via secondary machine makers.

DISTINGUISHABLE FEATURES

- The inputs are of [–] common (sink) type. There is no [S/S] terminal.
- The outputs are of sink connection when the output type is transistor.
- The model name indicates Japanese specification by the omission or by otherwise, of the model variation part of the model name. The name of the series is also different.
  example: FX_2-32MT

Unless stated elsewhere in the manual, other functions and specification are the same for the Input Configurable Type and the Japanese Specification Type.

(1) THE DIFFERENCES OF [+] COMMON (SOURCE) AND [–] COMMON (SINK) INPUT TYPES

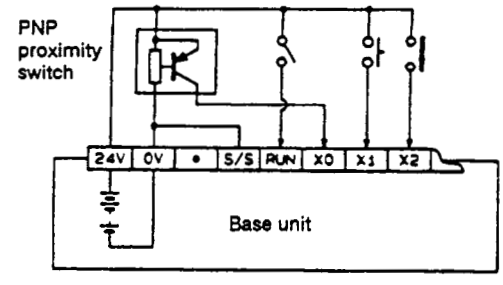
An input of a programmable controller may take the form of [+] common or [–] common depending on whether the current flows into or out of the input terminal of the programmable controller. The FX programmable controllers sold outside of Japan can be configured for both types.

- [+] COMMON INPUT (SOURCE)

  [+] common input refers to a connection where the common point of the input devices are relatively positive.

  The current flows into the input terminals of the programmable controller.

  The source/sink (S/S) select terminal should be connected to the 0 V terminal as shown.

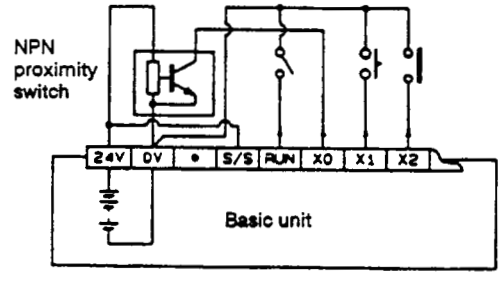


- [–] COMMON INPUT (SINK)

  [–] common input refers to a connection where the common point of the input devices are relatively negative.

  The current flows out of the input terminals of the programmable controller.

  Remove any short links between the [S/S] and [0 V] and connect the [S/S] terminal to the [24 V] terminal as shown.



1. GENERAL HARDWARE AND INSTALLATION

1.5 INSTALLATION

1.5.1 MOUNTING AND WIRING

(1) MOUNTING DIMENSIONS

(a) DIN RAIL MOUNT

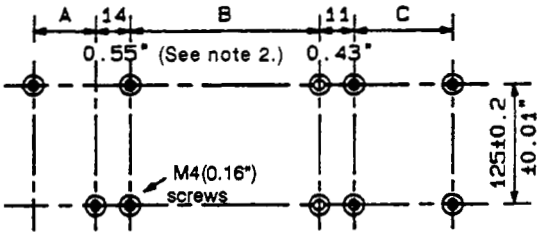
Units and blocks can be mounted to a DIN46277 rail (width : 35 mm (1.38 in)).

Pull the hook down to detach the units and blocks from the DIN rail.

Caution: • *Do not mount on DIN rail in cases where there is excessive vibration. Direct mounting is a better alternative in this case.*

(b) DIRECT MOUNT

Mount holes (M4 (0.16") screw holes) provided as illustrated below can be used for direct mount.



The holes marked with asterisks (*) in the illustration are not used on the FX models listed that are marked with an asterisk (*) in the table to the right. Base and extension units can be arranged in two rows as shown on the next page.

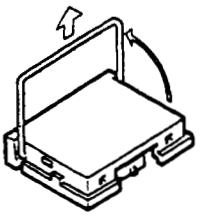
Model	A ± 0.2 (±0.01")	B ± 0.2 (±0.01")	C ± 0.2 (±0.01")
FX_1-16M, 24M		140(5.51")	
FX_1-32M		150(5.91")	
FX_1-48M		220(8.66")	
FX_1-64M		260(10.24")	
FX_1-80M		320(12.60")	
FX-32E			150(5.91")
FX-48E			220(8.66")
* FX-8E, 8EX, 8EY			35(1.38")
* FX-16EX, 16EY			63(2.48")
* FX_2-24EI			35(1.38")
FX_2-40AP, 40AW	35(1.38")		

Units are in mm(inch).

Note 1: • *To prevent a rise in temperature, mount the units to walls. Never mount them to the floor or ceiling.*

Note 2: •*The FX2-40AP requires at least 21 mm (0.83") distance from the basic unit for good noise immuity purposes.*

(2) HANDLING OF ACCESSORIES



See general view in Section 1.1.6.

(a) EXTENSION CABLE CONNECTION

Connect the extension cables to the correct connector.

(b) MEMORY CASSETTE

Remove the memory cassette by raising the wire lever as illustrated and carefully pulling it upward.

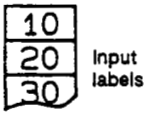
Caution: • *Turn off the power before removing the memory cassette. Programs in the RAM memory cassette will be lost when the cassette is removed from the base unit.*

(c) PLACING THE I/O NUMBER LABELS

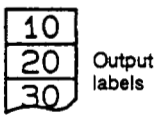
The number labels that are supplied with the extension units and blocks should be placed at every 8 points on the I/O terminals of the extension units and blocks.

The numbers must be consecutive from the numbers used for the base unit.

The upper and lower terminals of the FX-16EX and FX-16EY are assigned lower and higher numbers respectively.



Input labels

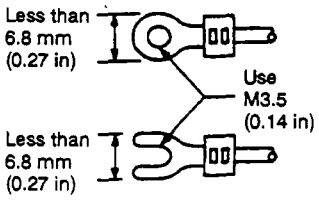


Output labels

1. GENERAL HARDWARE AND INSTALLATION

INSTALLATION

(3) WIRING



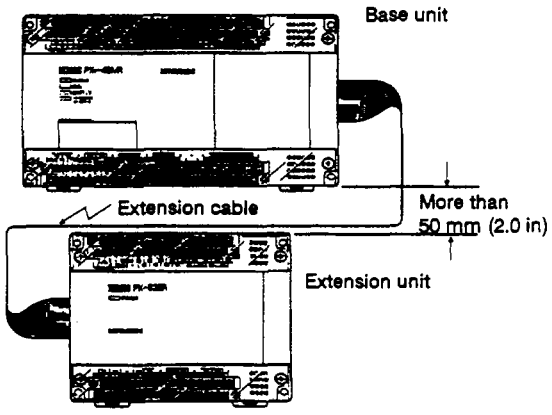
Wiring cautions :

- Use the crimp terminals as shown to the left.
- Tighten the terminal screws to between 5 and 8 kg-cm (5.76 and 9.22 in-lb). They must be secure enough to prevent a loose connection from causing a malfunction.
- Do not connect a cable to a vacant terminal ([·]).

- Do not run both input signal lines and output signal lines through the same cable.

  I/O signal lines should not be laid with power cables or run in the same duct as power cables.
- I/O lines can be extended to between 50 and 100 m (164.05 and 328.1 ft). However, it is advisable that I/O lines be limited to 20 m (65.62 ft) as a precaution against noise interference.

(a) TWO-ROW ARRANGEMENT



- Each extension unit is provided with an extension cable.
  Single-row arrangement : 55 mm (2.17 in)
  Two-row arrangement : 650 mm (25.59 in)
- Extension blocks are supplied with permanently connected extension cables.
- Extension cables are very susceptible to noise. They should be laid at least 30 to 50 mm (1.18 to 1.97 in) away from from PC output lines and power cables.
- Leave at least 50 mm (1.97 in) between units and other structures to allow good ventilation.

CAUTION

Environment :

- Do not install in areas subject to excessive dust, oily mist, conductive dust, corrosive gas, or flammable gas.
- Do not mount in areas subject to shock or vibration.
- Do not mount in areas subject to high temperature, moisture, or rain.

During Mounting :

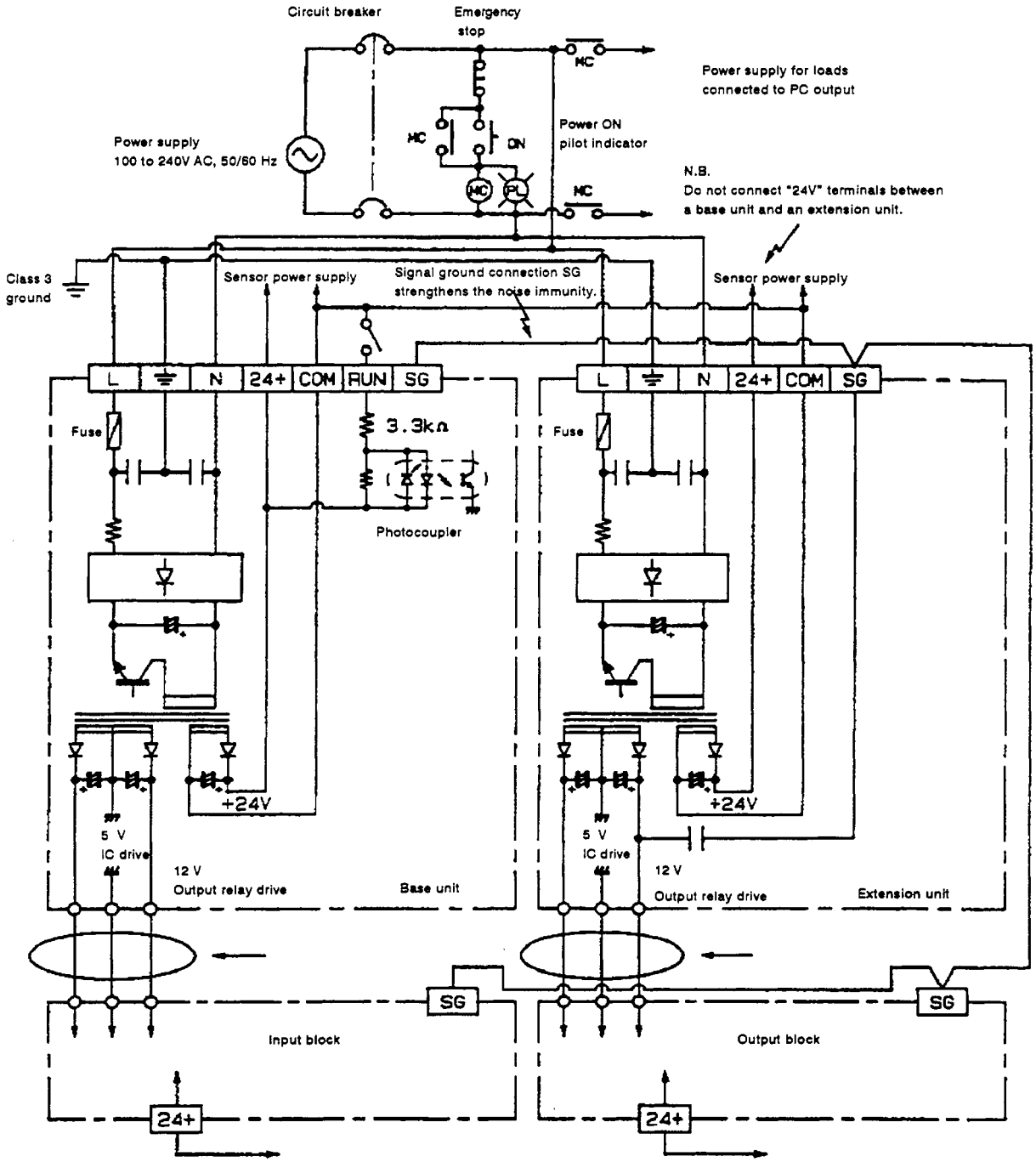
- Do not allow cut wires, filings, or shavings to fall inside a unit or block when drilling holes or connecting cables/lines.
- Remove the protective cover from a unit or block once it has been mounted to prevent overheating.
- Leave at least 50 mm (1.97 in) between a unit or block and other device or structure. Ensure that mounted units and blocks are kept as far as possible from high-voltage cables, high-voltage equipment, and power equipment.

1.6 POWER SUPPLY CIRCUIT

1.6.1 POWER SUPPLY CIRCUIT CONFIGURATION AND SPECIFICATIONS

(1) POWER SUPPLY CONNECTION CIRCUIT EXAMPLE

The following is an example of power supply connection circuit. See Section 1.11 for the actual arrangement of the terminals.



1. GENERAL HARDWARE AND INSTALLATION

POWER SUPPLY CIRCUIT

(2) POWER SUPPLY SPECIFICATIONS

Item		FX2-16M	FX2-24M	FX2-32M FX-32E	FX2-48M FX-48E	FX2-64M	FX2-80M
Power Supply Voltage		AC85~264V +10 / -15% AC 50/60 Hz (120/240 V system)					
Maximum Allowable Momentary Power Failure Period		10 msec (operation does not stop if a momentary power failure of less than 10 msec occurs)					
Fuse		250 V 2 A, ø5 x 20 mm (0.20" x 0.79")			250 V 5 A, ø5 x 20 mm (0.20" x 0.79")		
Power Consumption (VA)		30	35	40	50	60	70
Sensor Power Supply	Without Extension Block	24V DC less than 250 mA			24V DC less than 400 mA		
	With Extension Block	24V DC less than 100 mA (16 points extended)			24V DC less than 150 mA (32 points extended)		

WIRING CAUTION

Power Supply :

- The power supply is a 120/240V AC system. Either system should be connected to terminals "L" or "N".
  Do not connect the power supply to an input terminal or the [24+] terminal. If connected, it will severely damage the PC.
- The base unit and an extension unit power supply should be turned ON/OFF simultaneously.
- During emergencies, the PC output circuits should be turned off using a switch external to the PC.
- The PC will continue to operate during a momentary power failure of less than 10 msec. If a momentary power failure exceeding 10 msec or an abnormal voltage drop occurs, PC operation ceases and output is turned off. Once the proper power supply operation is restored, the PC will restart automatically (provided that the RUN input is set to ON).
- Power cables must be at least 2 mm^2 (AWG 14) to avoid voltage drops.

Grounds :

- Use a cable at least 2 mm^2 (AWG 14) to ground the equipment. Ground resistance must be less than 100Ω (class 3). Note that the ground cable must not be connected to the same ground point as the power circuits. Providing a ground is recommended, although, if a proper ground is impossible, the PC will still operate correctly even if grounding is not used.
- When an extension unit is used, connect the ground terminals of the extension unit to the base unit ground terminal and ground the base unit. The extension unit ground terminal should be connected to the base unit ground terminal even if the base unit itself is not grounded. Connect the "SG" terminals of base unit, extension units, and extension blocks with a cable of at least 2 mm^2 (AWG 14).

Sensor Power Supply :

- The output current of a 24V DC power supply for sensors varies from 0.1 to 0.4 A as shown in the table above. This variation depends on the type of PC model and whether or not an extension block is used.
  If an overload occurs, the voltage is lowered automatically, deactivating the PC inputs. Do not connect an external power supply to the [24+]terminal.
- Do not connect the [24+] terminals on the base unit and extension units.
- Connect the [COM] terminals on the base unit and extension units to each other.

1. GENERAL HARDWARE AND INSTALLATION

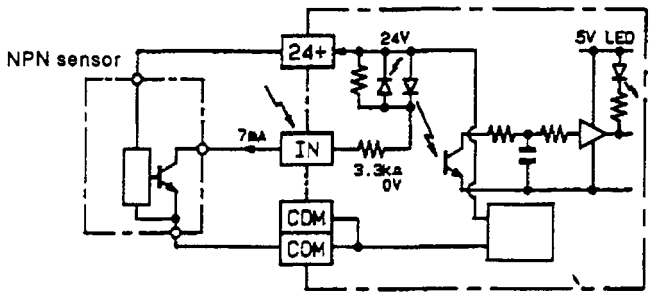
1.7 INPUT CIRCUIT CONNECTION

1.7.1 DC INPUT SIGNAL CIRCUITS

(1) INPUT SPECIFICATIONS

The following figure shows the PC input signal circuit (including the RUN input) specifications.

(a) INPUT TERMINALS



The input signal is turned ON when current flows through the input terminal. The input device is connected between an input and the [COM] terminal.NPN open collector transistor sensors can also be used.

The input indicator will light when the input is turned ON.

Note: • *Isolate input commons from output [COM] terminals to minimize noise from heavy loads.*

(b) INPUT CIRCUIT

The primary and secondary input circuits are isolated by a photocoupler, with the secondary input circuit provided with a C-R filter. This prevents operation errors due to input contact chattering or other noises that may enter via the input line.

For this reason, ON/OFF status changes (from ON to OFF or from OFF to ON) in the PC will be delayed for approximately 10 msec.

(c) INPUT SENSITIVITY

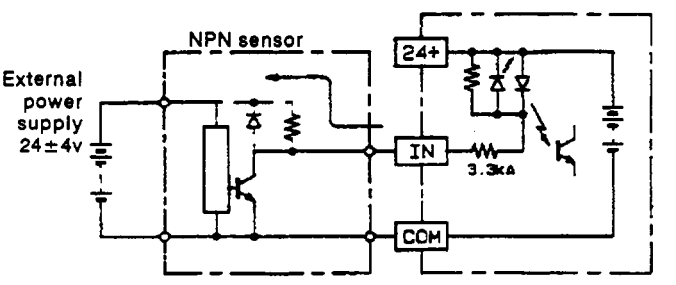
The input current is 24V DC, 7 mA with input sensitivity current between 2.5 to 3 mA. However, correct ON and OFF operation can be ensured by currents of at least 4 mA for ON and not more than 1.5 mA for OFF.

For this reason, it is important that the precautions given on the following page be followed if a diode or resistor is connected to the input contact in series, in parallel, or if there is a leak inhibiting component.

(d) SENSOR EXTERNAL CIRCUIT

The input current is supplied from the 24V DC power supply contained in the PC.

Therefore, if a sensor, such as a photoelectric switch, is driven by an external power supply, the voltage should be 24 ± 4V DC and sensor's output transistor must be an NPN type.

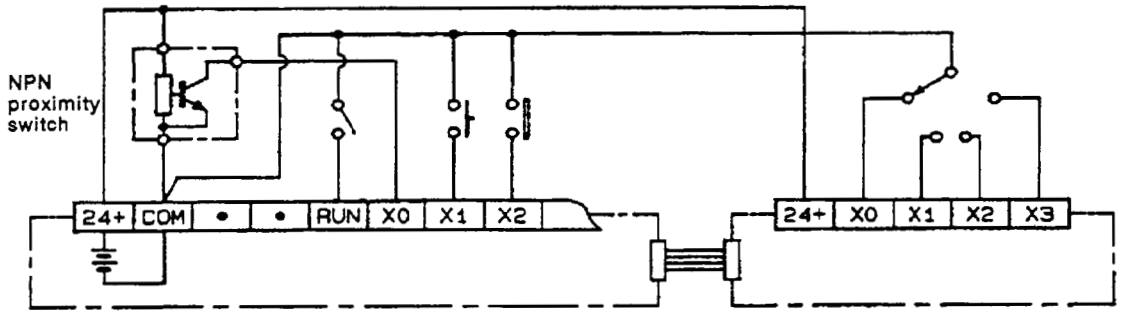


1. GENERAL HARDWARE AND INSTALLATION

INPUT CIRCUIT CONNECTION

(2) INPUT CONNECTION EXAMPLE

The following diagram provides an example of an input connection. For the actual arrangement of the terminals, see Section 1.11.



CONNECTION CAUTIONS

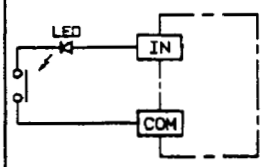
Selection of Input Devices :

The input current of the PC is 24V DC, 7 mA. Use only the input devices suitable for operating at this low current.

If large current switches are used, improper contact operation may occur.

Diodes Connected to Input Contacts in Series :

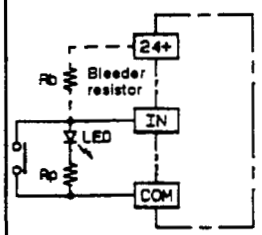
The voltage drop through a diode should be less than 4 V.



If reed switches with serially connected LEDs are used, no more than two should be connected in series.

Resistors Connected to Input Contacts in Parallel :

Parallel resistance must be greater than 15 KΩ.



If the resistance of a resistor Rp connected in parallel is smaller than this, connect a bleeder resistor Rb between the input (IN) and the [S/S] terminals.

$$Rb \leq \frac{4Rp}{15 - Rp} (K\Omega)$$

An alternative is to hold the leak current I (leak) of a 2-wire proximity switch to less than 1.5 mA while it is off. If leak current is greater than this, connect a bleeder resistor Rb which satisfies the following:

$$Rb \leq \frac{6}{I(leak) - 1.5} \times (K\Omega)$$

1. GENERAL HARDWARE AND INSTALLATION

1.8 OUTPUT CIRCUIT CONNECTION

1.8.1 RELAY OUTPUT CIRCUITS

(1) OUTPUT SPECIFICATIONS

(a) OUTPUT TERMINALS

Each output point on an FX1-16M Series PC is common isolated. Other models provide between 4 to 8 points of common output. Each common terminal is assigned a number from [COM1] to [COM7].

Different voltage systems, such as 200V AC, 100V AC, and 24V DC, can be used for different common blocks to drive loads of various voltages.

(b) CIRCUIT ISOLATION

The internal circuits of the PC and external load circuits are electrically isolated (from each other) by the coils and the contacts of the output relays. Common blocks are also isolated from each other.

(c) OPERATION INDICATORS

When the an output relay is energized, an LED will light and the output contact is ON.

(d) RESPONSE TIME

The response time between the energizing or de-energizing of an output relay contact and the turning ON or OFF of an output contact is approximately 10 msec.

(e) OUTPUT CURRENT

A circuit voltage lower than 250V AC can drive the following loads :

Pure resistive load : 2 A/point

Inductive load : Less that 80 VA (100 or 240V AC)

Lamp load : Less than 100 W (100 or 240V AC)

The service life of the relay contact points for inductive loads is summarized in the table below. When using an output contact for a DC inductive load, a surge absorbing diode should be mounted in parallel with the load and a maximum power supply voltage of 30V DC must be used.

(f) LEAK CURRENT OPEN CIRCUITS

The neon bulb can be directly driven by an output contact due to the lack of leak current when the output contact is OFF.

Ref.: *•Relay output contact service life*

The rated service life for inductive AC load, such as contactor or solenoid valve, is 500,000 operations at a load of 35 VA. The following table is based on Mitsubishi reliability tests.

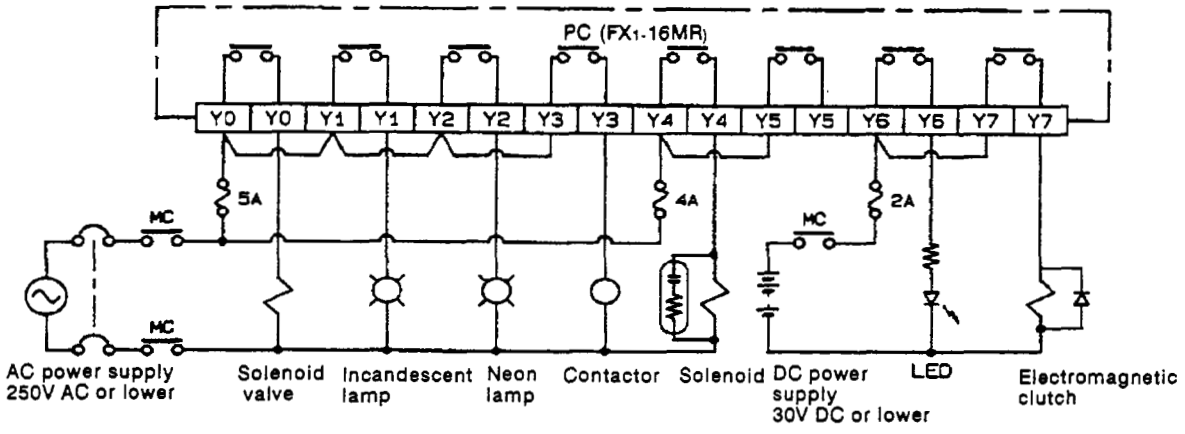
Load Capacity		Contact Life	Load Examples (Mitsubishi's MC)
35 VA	0.35 A/100V AC	3,000,000 times	S-A10 to S-A80
	0.15 A/240V AC		S-K20 to S-K150
80 VA	0.8 A/100V AC	1,000,000 times	S-A100 to S-A150
	0.33 A/240V AC		S-K180 to S-K400
120 VA	1.2 A/100V AC	200,000 times	S-A220 to S-A401
	0.5 A/240V AC		

The test uses a 1 sec ON/OFF cycle. The addition of in-rush overcurrent greatly reduces relay contact service life.

OUTPUT CIRCUIT CONNECTION

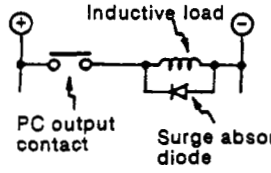
(2) OUTPUT CONNECTION EXAMPLE

The following is an example of an output connection circuit. For the actual arrangement of the terminals, see Section 1.11.



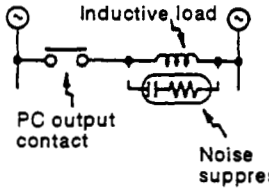
Since fuses are not included in the internal output circuit as shown above, a 5 to 10 A fuse should be used for each 4 points.

OUTPUT CIRCUIT CONFIGURATION



DC Loads :

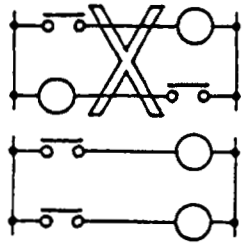
Connecting a surge absorbing diode in parallel with a load will greatly lengthen the service life of a contact.



AC Load :

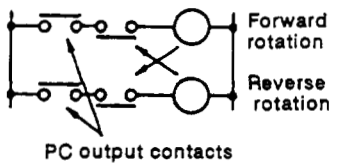
Connecting a noise suppressor in parallel with an AC inductive load will reduce noise generation.

Noise suppressor : 0.1μF + 100 to 120Ω



Load Orientation :

The orientation of the PC output contacts should be the same.



Interlock :

Contacts that control loads, such as those used for forward/reverse rotation, which may turn on simultaneously constitute a dangerous hazard. Interlocks external from the PC, as well as programmed interlocks, should be used to prevent such loads from simultaneous activation.

OUTPUT CIRCUIT CONNECTION

1.8.2 SSR OUTPUT CIRCUIT

(1) OUTPUT SPECIFICATIONS

(a) OUTPUT TERMINALS

Each output point on an FX2-16M Series PC is common isolated. Other FX2 Series models have their outputs commoned in groups of 4.

Different voltage systems, such as 200V AC and 100V AC, can be used for separately commoned blocks.

(b) CIRCUIT ISOLATION

PC internal circuits and the output devices (triac) are isolated by a photocoupler. Common blocks are also isolated from each other.

(c) OPERATION INDICATOR

An LED will light and the output triac is ON when the photocouples is being driven.

(d) RESPONSE TIME

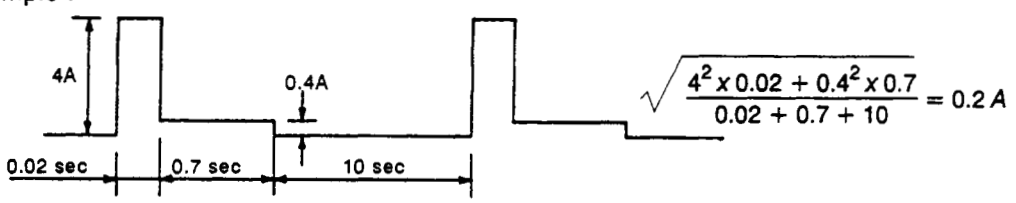
The response time between the activation and de-activation of the photocoupler and the turning ON and OFF of the output triac is less than 1 msec and 10 msec respectively.

(e) OUTPUT CURRENT

The circuit can handle a maximum current flow of 0.3 A/point.

However, the current should be limited by design to 0.8 A/4 points in order to prevent excessively high temperatures to arise. When a high in-rush current is required to turn a load ON/OFF, the root mean square current should be less than 0.2 A.

Example :



(f) LEAK CURRENT IN OPEN CIRCUITS

A C-R absorber is connected in parallel to an output terminal of the PC.

Note: ●*As a result, a leakage current of 1 mA/100V AC or 2mA/240V AC occurs when the circuit is open and can inadvertently activate loads that require only a low current for operation. Either the load capacity must be restricted to those requiring greater than 0.4 VA/100V AC or 1.6 VA/200V AC or, if a smaller capacity load or a neon lamp is used, connect the surge absorber specified on the following page in parallel with the load.*

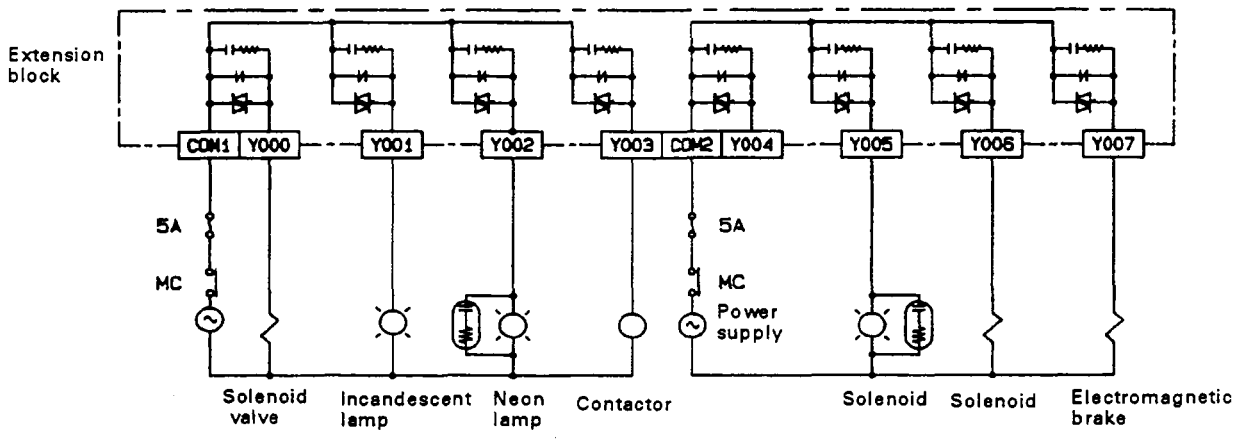
Ref.: ●*Leakage current in SSR output circuits is greater than that of relay devices and can cause miniature relays to hold their operation.*

OUTPUT CIRCUIT CONNECTION

(2) OUTPUT CONNECTION EXAMPLE

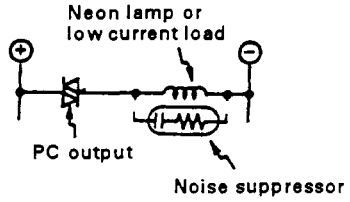
The following is an example of an output connection circuit. For the actual arrangement of the terminals, see Section 1.11.

FX-8EYS Japanese Spec.



Since a fuse is not included in the output circuit of the PC as shown above, a 5 to 10 A fuse should be used for each 4 points.

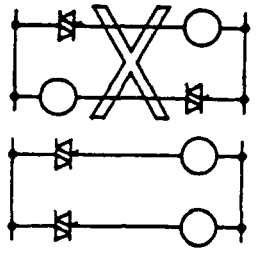
OUTPUT CIRCUIT CONFIGURATION



Weak Current Loads :

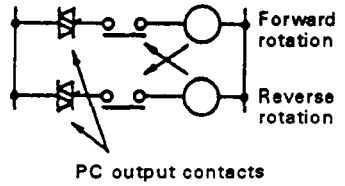
When a neon lamp or a low current load of less than 0.4 VA/100V AC or 1.6 VA/200V AC is connected, connect the following noise suppressor in parallel with the load.

Noise suppressor : 0.1μF + 100 to 120Ω



Load Orientation :

The orientation of the PC output contacts should be the same.



Interlock :

Contacts that control loads, such as those used for forward/reverse rotation, which may turn on simultaneously constitute a dangerous hazard. Interlocks external to the PC, as well as programmed interlocks, should be used to prevent such loads from simultaneous activation.

OUTPUT CIRCUIT CONNECTION

1.8.3 TRANSISTOR OUTPUT CIRCUIT

(1) OUTPUT SPECIFICATIONS

(a) OUTPUT TERMINALS

Each output point of a FX1-16MT Series PC is common isolated. Other models have their outputs commoned in groups of 4 or 8.

A 5 to 30V DC smoothed power supply should be used to drive the loads.

(b) CIRCUIT ISOLATION

PC internal circuits are isolated from output transistors with a photocoupler. Common blocks are also isolated from each other.

(c) OPERATION INDICATORS

An LED will light and the output transistor is ON when an output transistor is being driven.

(d) RESPONSE TIME

The response time between the activation or deactivation of a photocoupler and turning ON or OFF of an output transistor is less than 0.2 msec. (at 24 V 200 mA)

(e) OUTPUT CURRENT

The circuit can handle a maximum current flow of to 0.5 A/point.

However current should be limited by design to 0.8 A/4 points (an average of 0.2 A/point) in order to prevent excessively high temperatures from rising.

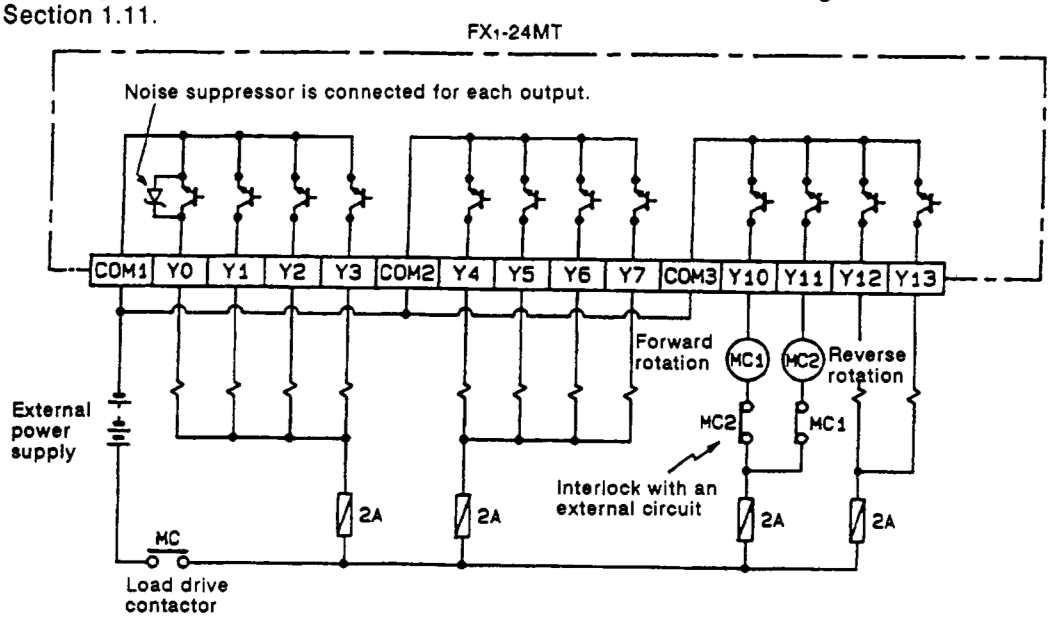
An output transistor turns ON with a voltage drop of 1.5 V. Take this into consideration when driving semiconductors.

(f) LEAKAGE CURRENT IN OPEN CIRCUITS

Leakage current is less than 100 μA.

(2) OUTPUT CONNECTION EXAMPLE

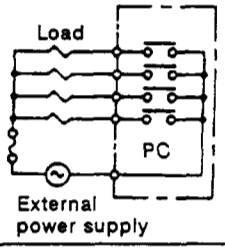
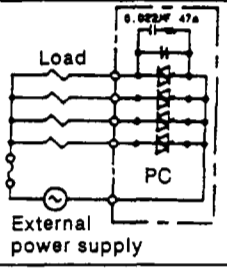
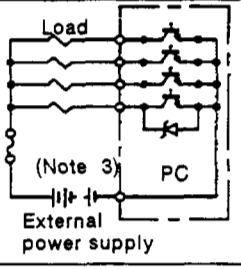
The following is an example of an output connection. For the actual arrangement of the terminals, see Section 1.11.



1. GENERAL HARDWARE AND INSTALLATION

OUTPUT CIRCUIT CONNECTION

1.8.4 OUTPUT SPECIFICATIONS

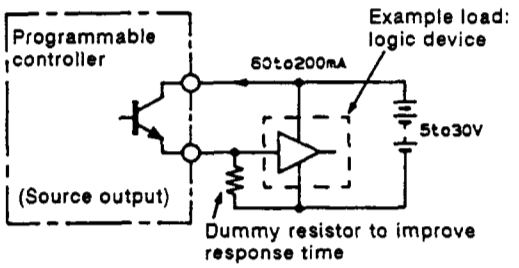
Item		Relay Output	SSR Output	Transistor Output (Sink)
Circuit Configuration				
External Power Supply		Less than 250V AC, 30V DC	85 to 242V AC	5 to 30V DC
Maximum Load	Resistance Load	2 A/point	0.3 A/point 0.8 A/4 points	0.5 A/point 0.8 A/4 points
	Inductive Load	80 VA (See the table on Section 1.6.1 (1).)	15 VA/100V AC 30 VA/200V AC	12 W/24V DC
	Indicator Load	100 W	30 W	1.5 W/24V DC
Open Circuit Leak Current		–	1 mA/100V AC 2mA/200V AC	0.1 mA/30V DC
Minimum Load		**(Note 1)**	0.4 VA/100V AC 1.6 VA/200V AC	–
Response Time	OFF → ON	Approx. 10 msec	Less than 1 msec	Less than 0.2 msec
	ON → OFF	Approx. 10 msec	Max. 10 msec	Less than 0.2 msec **(Note 2)**
Circuit Isolation		With relay	With photocoupler	With photocoupler
Operation Indication		LED is lit when relay coil is energized	LED is lit when photocoupler is driven	LED is lit when photocoupler is driven

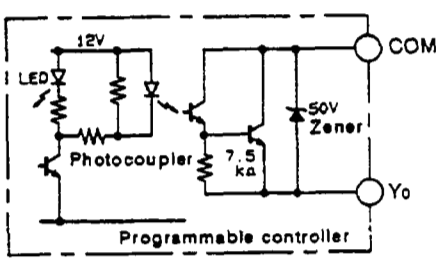
Note 1: •When the external power supply voltage is 24V DC or less, if possible, allow at least 5 mA to flow.

Note 2: •The response of 0.2 msec is stated at 24 V, 200 mA. The amount of time required to turn OFF increases as load current decreases. This response time can be improved with light loads by increasing the amount current flow through an added dummy resistor.

•If the response time of 0.5 msec or better is desired, ensure a current of above 60 mA flows at 24V DC.

Note 3: •Internal protection and phpt-coupler current of the transistor output.





1.9 PRELIMINARY CHECKS

1.9.1 PROGRAMMING AND TESTING

(1) CHECKUP (POWER : OFF)

Incorrect connections at the power supply terminals, a short between the DC input lines and the power supply cable, or a short in the output lines can seriously damage the PC.

Before turning on the power, check the power supply and ground connections and the input/output line connections.

Ref.: • *Use the following procedure if it is necessary to measure the PC withstanding voltage and insulation resistance.*

- *Disconnect the input/output lines and the power supply cable from the PC.*
- *Jumper each terminal to the ground terminal.*

Measure the withstanding voltage and insulation resistance between the jumper and the grounding terminal.

Withstanding voltage : 1500VAC, 1 minute

Insulation resistance : Higher than 5 MΩ, measured with 500VDC Megger

(2) WRITING AND CHECKING A PROGRAM (POWER : ON, PC : STOP)

Write a program with a peripheral device.

Read the program to make sure it has been written correctly. At the same time, check the program for circuit and syntax errors using the program check function on the peripheral device.

Ref.: • *When an FX-20P programming panel is used, the PC will remain in the STOP status when the power is turned ON if the [WR] key is held down (even if the PC RUN input is ON). Therefore, programming is possible without turning OFF the RUN input. The PC will restart when power is turned off and then on again.*

• *Each output can be forcibly turned ON/OFF with the programming panel.*

(3) OPERATION AND TESTING (POWER : ON, PC : RUN)

When the RUN input is turned on, the PC is placed in the RUN state. It is possible to change the settings for timers, counters, and data registers or to forcibly turn devices ON/OFF while the PC is running.

Ref.: • *A programming panel can be used to check the ON/OFF status of each device and the continuity of each circuit block or to read the present data of timers, counters, and data registers.*

With a graphic programming panel, the circuit diagram is displayed while monitoring the devices in the PC. In this way, program debugging is made much easier.

1.10 ERROR CHECK

1.10.1 CHECKING ERRORS WITH THE LEDS

If a problem occurs while the PC is running, check the power supply voltage, the PC terminal screws, and the I/O devices for looseness or other abnormalities.

After that, check the LEDs on the PC. The LEDs will help you determine if the PC is faulty or if the problem is being caused by an external device.

(1) POWER INDICATOR (POWER LED : OFF)

There is a POWER LED on the front panel of the base unit, extension unit, extension block, and special function adapter. It is lit by 12V DC, supplied from the base unit or extension unit internal power supply. If the POWER LED is not lit when the power is turned on, disconnect the 24+ supply terminal. The sensor power load is too high if the POWER LED then lights. In this case, the 24V DC power to the sensors must be supplied externally.

The fuse in the base unit or extension unit may blow if conductive foreign matter has entered the PC or other faulty conditions exist. If this is the case, merely changing the fuse does not solve the problem as the same result will occur again. Please contact your local Mitsubishi service center.

(2) BATTERY VOLTAGE INDICATION (BATT.V LED : ON)

The BATT.V LED will light if the battery voltage drops while the PC is running and special auxiliary relay M8086 is activated. Programs in the RAM or data in memory area backed up by battery can be retained for one month after the BATT.V LED has lit. However, the battery should be replaced as soon as possible if this LED is lit. It is recommended to replace the battery every 5 years. In the case when 8K RAM cassette is used, replace the battery every 3 years.

Refer to Section 1.9 for the battery replacement procedure.

Ref.:
- *The BATT.V LED does not light even when battery voltage is low if special auxiliary relay M8030 is activated. However, special auxiliary relay M8006 will be operational.*
- *Data in a data register, used as a setting for a timer or a counter or other, will become unreliable if battery voltage is low.*

(3) ERROR INDICATION (PROG.E LED : FLASH)

The PROG.E LED may flash when a program is run in cases where the timer or counter constants are not set; a circuit is programmed incorrectly; battery voltage is abnormally low; the stored program has been altered by noise or conductive foreign matter has entered into the PC. When the PROG.E LED flashes, check the program, noise sources, the BATT.V LED, and also the inside of the PC for conductive foreign matter.

Ref.:
- *If an error occurs, a register number (8009, 8060 to 8068) corresponding to the cause of the error is stored in special data register D8004. Read the register number in D8004 and the data in that register to find the error code number. Use this error code number and the table in Section 6.1.8 to find the contents of the error and the corrective action.*

ERROR CHECK

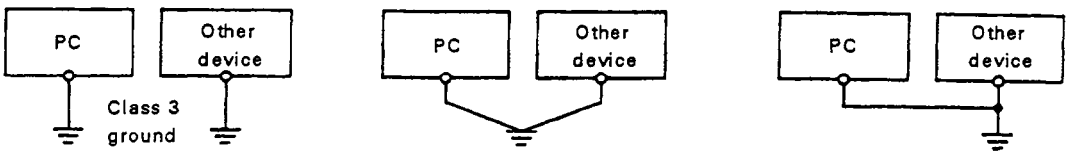
(4) ERROR INDICATION (CPU.E LED : ON)

- The CPU.E LED may light when the CPU runs out of control due to external noise or conductive foreign matter in the PC or if the program cycle time exceeds 0.1 sec.

  The CPU.E LED may also light when a memory cassette has been removed or installed while the power supply to the PC is ON.

  Turn off the power supply once and then turn it back on again. After that set the PC in the RUN status.

  If the PC now operates correctly, check for possible noise sources and inside the PC for conductive foreign matter.



Independent grounds (recommended) | Common ground (acceptable) | Common ground (unacceptable)

- Use wire at least 2 mm^2 (AWG 14) thick to ground a PC. Keep the ground wire length ground wire as short as possible. A class 3 ground (ground resistance of less than 100 Ω) is recommended.

  If the PC malfunctions because of an improper ground, disconnect the ground wire from the terminal ⏚ on the PC. In this case, the ground terminals of the base unit and the extension unit should be connected with each other.

- If the error changes from CPU.E to PROG.E (indicated by a steady On to a flashing LED) when the ground wire is disconnected from the PC ground terminal, check the program.

  If the CPU.E LED remains lit, review the program to check if the operation cycle time might be excessively long. The operation cycle time is stored in special data register D8012.

(5) INPUT ERRORS

Irrespective of whether the state of the input monitored by the programming panel agrees or not with the status of the LED of that input, check the following :

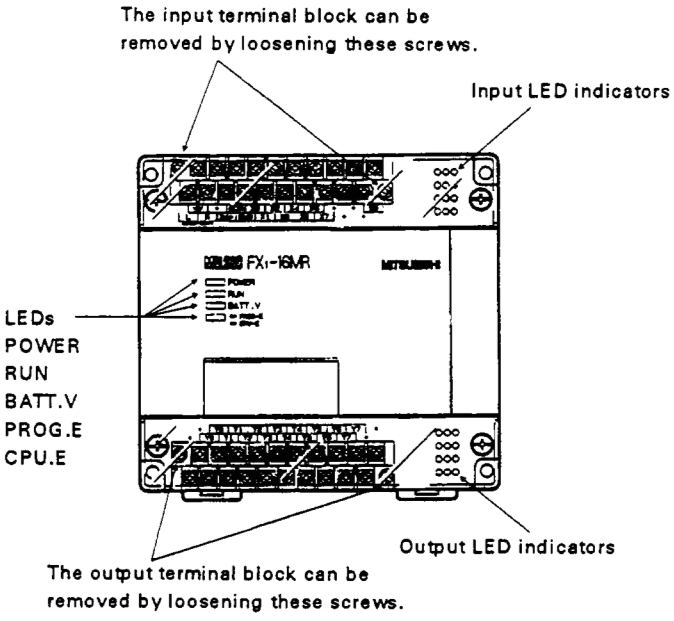
- A loose contact error will occur if rated current of the input switch is too large. Another possibility is that oil has entered.
- If the input switch contains a parallel LED resistor circuit, it is possible that the PC input signal can be turned on through this circuit even though the switch is in the OFF position.
- An input that is turned ON/OFF in a period shorter than the operation cycle time of the PC will not be accepted.
- If there is an overload of the 24V DC power supply or if there is a short in the power circuit, the protection circuit is activated and voltage is automatically lowered. In this case, each PC input will become inactive (PC will enter a power off state but POWER LED is dimly lit.).
- If a voltage higher than the rated voltage is applied to an input terminal, the input circuit may be damaged.
- Insufficient contact of the input terminal block on the base unit or extension module is another possible cause of this problem. Remove the terminal block and refit it securely.

ERROR CHECK

(6) OUTPUT INDICATION

If the load does not turn ON/OFF despite the status of the LEDs, an open-circuit leak current (SSR output) or one of the following may be the cause of the problem.

- The output contacts may have melted because of an overload or a short; or the contact is insufficient because its surface has eroded.
- Since insufficient contact of the input terminal block on the base unit or extension module is another possible cause of this problem, remove the terminal block and refit it securely.



1.11 MAINTENANCE AND INSPECTION

1.11.1 MAINTENANCE AND INSPECTION PROCEDURE

(1) PERIODICAL INSPECTION

Most of the parts in the PC will never need to be replaced. However, the service life of the battery is approximately 5 years* and it should be replaced as necessary. It is necessary to check to the condition of output relays if they are operated frequently or used to drive large-capacity loads. See Section 1.6.1.

* Battery service life will be approximately three years when an 8K RAM memory cassette is used.

Also check to the following points :

- The interior temperature of the panel
- Airborne or electrically-conductive dust inside the panel
- Loose wiring or terminals

(2) REPLACING THE BATTERY

If battery voltage is lower than the allowable limit, the BATT.V LED on the front panel will light when the power is turned ON. Although, memory data will be retained for approximately one month after this LED lights for the first time, the battery should be replaced as soon as possible.

A memory backup capacitor can be used when an EPROM or EEPROM is used as the memory cassette type. In this case, image memory data will be retained for approximately three days (once it has been fully charged before power off; approx. 1 hour.).

(a) BATTERY REPLACEMENT PROCEDURE

1) Turn off the power supply.

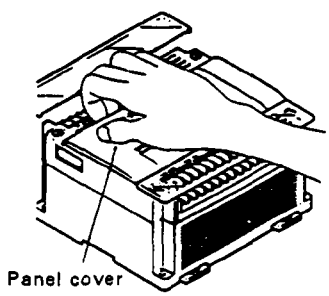
2) Remove the terminal cover and open the panel cover.

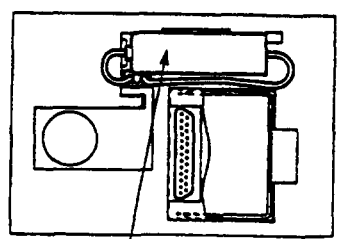
3) Remove the battery from the holder and remove the connector.

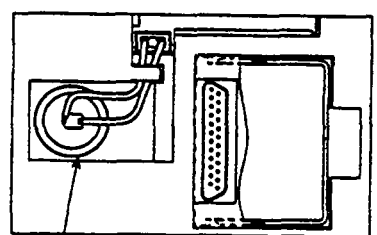
4) Insert the connector of a new battery immediately (within 30 sec after the removal of the old battery).

5) Fit the battery or memory backup capacitor into the holder and replace the panel cover.

Note: • Replace a memory backup capacitor while the power is OFF but be sure to power ON again immediately or memory data may be lost. Charging a memory backup capacitor requires approximately one hour.




F2-40BL battery


FX-40CAP Memory Backup Capacitor

1. GENERAL HARDWARE AND INSTALLATION

1.12 EXTERNAL DIMENSIONS AND GENERAL SPECIFICATIONS

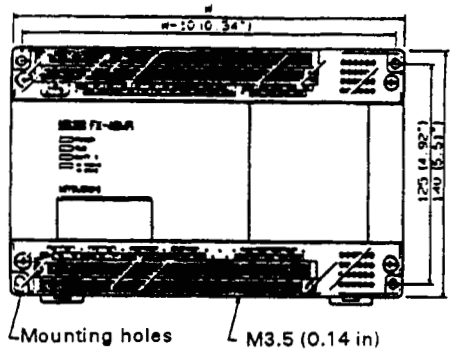
1.12.1 EXTERNAL DIMENSIONS

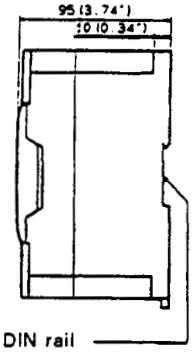
(1) BASE UNITS AND EXTENSION UNITS

<Accessories>

Extension unit : I/O number label (1 set)

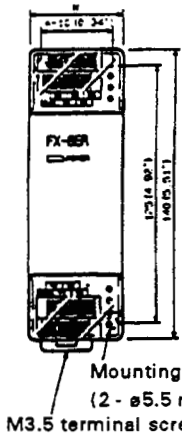
Extension unit : Extension cable,
650 mm (25.59 in) (1 pc.)
55 mm (2.17 in) (1 pc.)

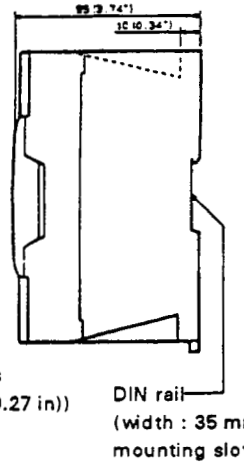




Model	W (mm (in))	Weight (kg (lb))
FX2-16M	150 (5.91)	1.0 (2.2)
FX2-24M		1.3 (2.86)
FX2-32M FX-32E	160 (6.30)	1.5 (3.3)
FX2-48M FX-48E	230 (9.06)	2.0 (4.4)
FX2-64M	270 (10.63)	2.5 (5.5)
FX2-80M	330 (12.99)	3.0 (6.6)

(2) EXTENSION BLOCKS AND SPECIAL FUNCTION ADAPTERS





<Accessories>

Extension block : I/O number label (1 set)

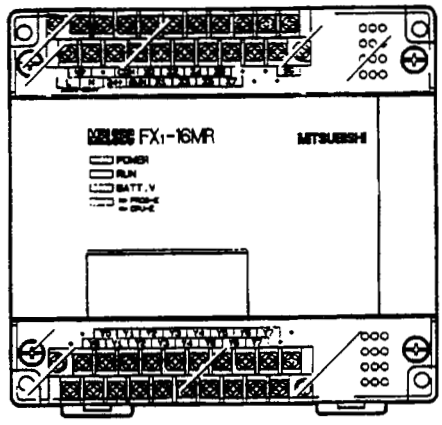
Model	W (mm (in))	Weight (kg (lb))
FX-8E, FX-8EX, FX-8EY	45 (1.77)	0.3 (0.66)
FX-16EX, FX-16EY	73 (2.87)	0.5 (1.1)
FX2-24EI	45 (1.77)	0.3 (0.66)
FX-232AW	45 (1.77)	0.3 (0.66)
FX2-40AP, FX2-40AW	45 (1.77)	0.3 (0.66)
FX-8AV	45 (1.77)	0.3 (0.66)

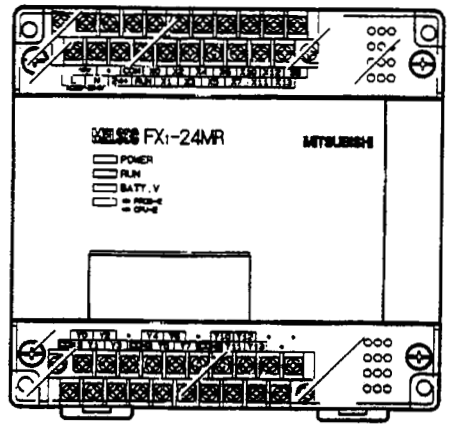
(3) GENERAL SPECIFICATIONS

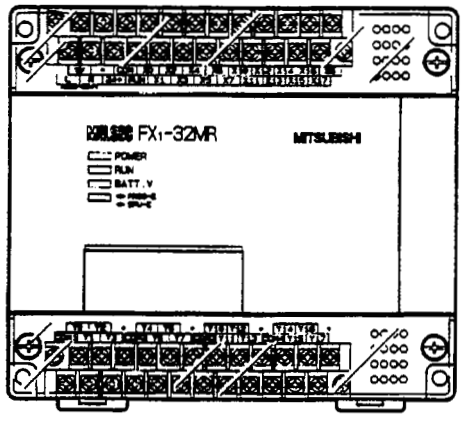
Operating Ambient Temperature	0 to 55°C	
Operating Ambient Humidity	35 to 85% RH, no condensation	
Vibration Resistance	Conforms to JIS 0911 10 to 55 Hz 0.5 mm (0.02 in) Max. 2G (0.5 G on DIN rail)) 2 hours in each of 3 axis directions	
Shock Resistance	Conforms to JIS 0912 (10 G 3 times in 3 directions	
Noise Durability	By noise simulator of 1000 Vpp noise voltage, 1 μs noise width at 30 to 100 Hz	
Dielectric Withstand Voltage	1500V AC for 1 minute	Between all terminals and ground
Insulation Resistance	500V DC 5 MΩ or over when measured with a megger	
Ground	Class 3 ground; ground is not required when it is impossible.	
Operating Ambience	To be free from corrosive gases. Dust should be minimal.	

1.13 TERMINAL LAYOUT

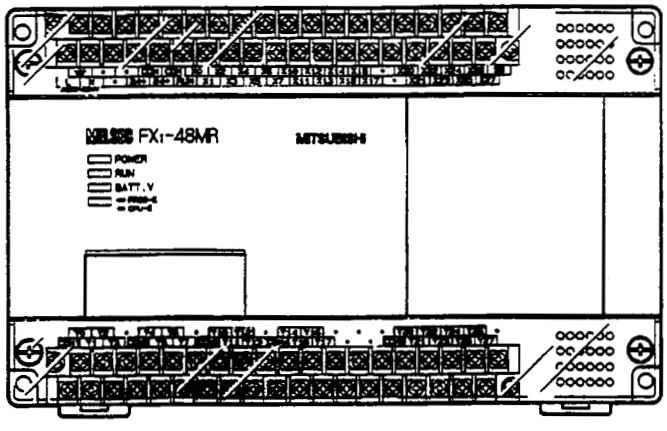
1.13.1 BASE UNIT





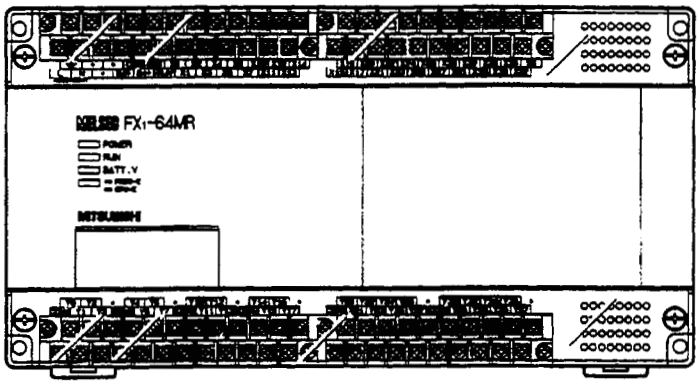


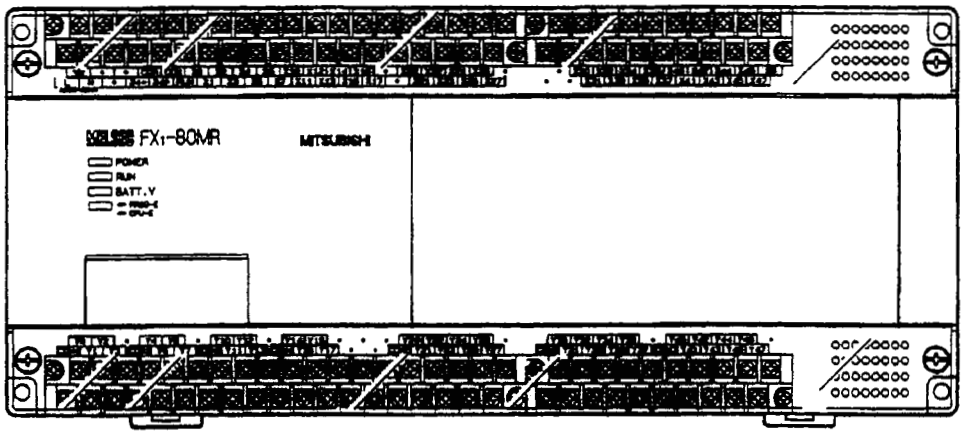
Note : Except for FX1-16MT, the terminal layouts for the triac and transistor output models are the same as relay output models.



1. GENERAL HARDWARE AND INSTALLATION

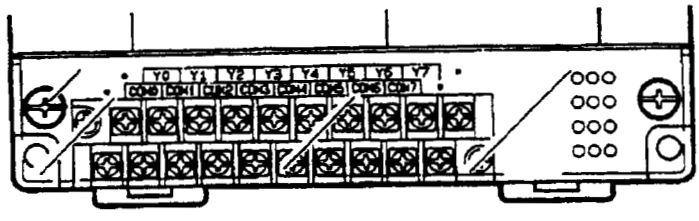
TERMINAL LAYOUT





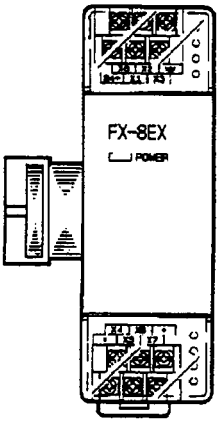
Note: • *Do not connect cables/wires to the undefined terminals (designated by [·]) on the base unit, extension unit, or extension block.*

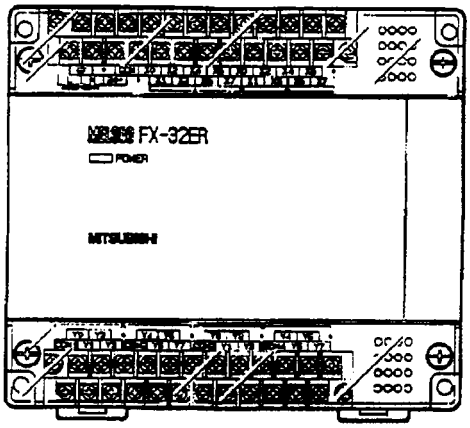
Output of FX1-16MT

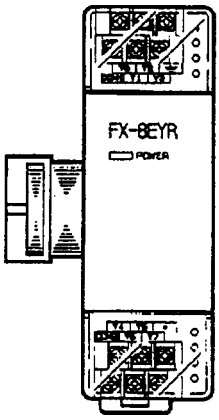


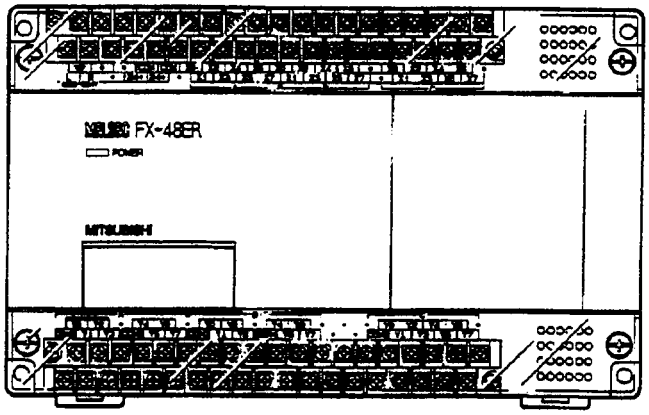
TERMINAL LAYOUT

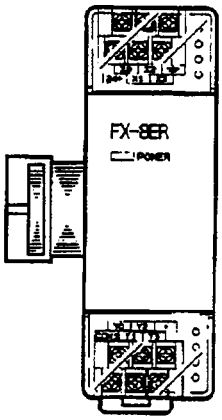
1.13.2 EXTENSION UNITS AND EXTENSION BLOCKS

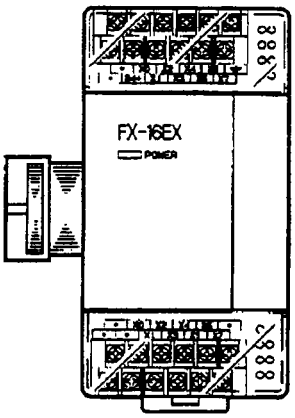


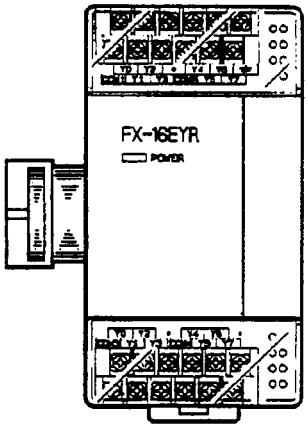












1. GENERAL HARDWARE AND INSTALLATION

1.14 PERFORMANCE SPECIFICATIONS LIST

1.14.1 PERFORMANCE SPECIFICATIONS

Item		Specification		Remark	
Operation control method		Cyclic operation by stored program		Performed by dedicated logic control LSI	
I/O refresh method		Batch processing method (when END instruction is executed)		Direct I/O instructions and input filter adjust instructions are available	
Operation processing time		Basic instruction : 1 μsec		Applied instruction:Under several 100 μsec	
Programming language		Relay symbolic language + Step ladder		SFC expression possible	
Program capacity and memory type		2K step RAM incorporated (standard)		Comment registration is possible (program memory used) Alphanumerics (15 characters/comment) 10 steps/comment; program memory is reserved in units of 50 comments Even if cassettes are used, the max. program area remains at 2k steps.	
		4K step EEPROM cassette (optional) 8K step RAM, EEPROM, EPROM cassette (optional)			
Number of instructions		Sequence instruction : 20, Step ladder instruction : 2, Applied instruction : 35			
Input relay	DC input	24V DC, 7 mA, isolated by photocoupler		X0 to X177 (octal)	I/O Relay 128 points in total
	–	–			
Output relay	Relay	250V AC, 30V DC, 2 A (resistive load)		Y0 to Y177 (octal)	
	SSR	242V AC, 0.3 A/point, 0.8 A/4 points			
	Transistor	30V DC, 0.5 A/point, 0.8 A/4 points			
Auxiliary relay	General use			M0 to M499 (500 points)	Areas can be changed by changing parameter setting
	Latch	Backed up by battery		M500 to M1023 (524 points)	
	Special purpose			M8000 to M8255 (256 points)	
State	Initial use	Can be used for initial state		S0 to S9 (10 points)	
	General use			S10 to S499 (490 points)	Areas can be changed by changing parameter setting
	Latch	Backed up by battery		S500 to S899 (400 points)	
	Annunciator	Backed up by battery		S900 to S999 (100 points)	
Timer	100 msec	0.1 to 3,276.7 sec		T0 to T199 (200 points)	
	10 msec	0.01 to 327.67 sec		T200 to T245 (46 points)	
Counter	Up counter	16 bits 1 to 32,767 counts	General use	C0 to C99 (100 points)	Areas can be changed by changing parameter setting
			Backed up by battery	C100 to C199 (100 points)	
	High-speed counter	32 bits Up/down	Backed up by battery	6 points in C235 to C255 (1-phase count)	
Register	General-purpose data register	16 bits	General use	D0 to D199 (200 points)	Areas can be changed by changing parameter setting
		16 bits	Backed up by battery	D200 to D511 (312 points)0	
	Special register	16 bits		D8000 to D8069 (70points)	
	Index	16 bits		V. Z (2 points)	
	File	16 bits (in program memory)	Backed up by battery	D1000 to D2999, Max. 2000 points, set by parameter	
Pointer	For JUMP/CALL			P0 to P63 (64 points)	
	Interrupt	Input interrupt using X0 to X2, and timer interrupt		I0 to I3 (3 points)	
Nesting		For master control		N0 to N7 (8 points)	
Constant	Decimal (K)	16 bits : –32,768 to 32,767, 32 bits : –2,147,483,648 to 2,147,483,647			
	Hexadecimal (H)	16 bits : 0 to $FFFF_H$, 32 bits : 0 to $FFFFFFFF_H$			

Ref.: ●*Power supply specification .....Section 1.4.1 (2)*

●*Output specification .....Section 1.6.4*

●*General specification .....Section 1.10.1*

1. GENERAL HARDWARE AND INSTLLATION

1.15 DEVICE NUMBER LIST

1.15.1 DEVICE NUMBER ALLOCATION

For the functions and operations of each device, see Section 4.

The devices marked with (B/U) are backed up by the battery.

Input Relay (X)	X0 to X7 8 points FX1-16M	X0 to X13 12 points FX1-24M	X0 to X17 16 points FX1-32M	X0 to X27 24 points FX1-48M	X0 to X37 32 points FX1-64M	X0 to X47 40 points FX1-80M	X0 to X177 128 points (with extension unit)	128 points in total
Output Relay (Y)	Y0 to Y7 8 points FX1-16M	Y0 to Y13 12 points FX1-24M	Y0 to Y17 16 points FX1-32M	Y0 to Y27 24 points FX1-48M	Y0 to Y37 32 points FX1-64M	Y0 to Y47 40 points FX1-80M	Y0 to Y177 128 points (with extension unit)	
Auxiliary Relay (M)	M0 to M499 500 points For general use		M500 to M1023 (B/U) 524 points for latch relays				M8000 to M8255 256 points For special purpose	
State (S)	S0 to S499 500 points (For general use) Initial: S0 to S9 / Zero return: S10 to S19			S500 to S899 400 points (B/U) Retentive		S900 to S999 100 points (B/U) For annunciator		
Timer (T)	T0 to T199 200 points 100 msec T192 to T199: For routine program		T200 to T245 46 points 10 msec					
Counter (C)	16 bits (Up-counter)				32 bits (High-speed up/down-counter), Max. 6 points			
	C0 to C99 100 points	(B/U) C100 to C199 100 points For data retention			(B/U) C235 to C245 1-phase, 1 input	(B/U) C246 to C250 1-phase, 2 input	(B/U) C251 to C255 2-phase, 2 input	
Data Register (D, V, Z)	D0 to D199 200 points For general use	D200 to D511 (B/U) 312 points For data retention For link				D8000 to D8255 256 points For special purpose	V, Z 2 points For index addressing	
Nesting Pointer	N0 to N7 8 points For master control		P0 to P63 64 points Branch pointer for jump and subroutine		I0** to I2** 3 points Input interrupt pointer			
Constant K	16 bits : -32,768 to 32,767				32 bits : -2,147,483,648 to 2,147,483,647			
Constant H	16 bits : 0 to $FFFF_H$				32 bits : 0 to $FFFFFFFF_H$			

T and C can be used as data registers when they are not being used as timers or counters. In this case, each point of C235 to C254 correspond to a 32-bit register.

1	GENERAL HARDWARE AND INSTALLATION
2	BASIC SEQUENCE INSTRUCTIONS
3	STEP LADDER INSTRUCTIONS
4	DEVICE DETAILS
5	APPLIED INSTRUCTION
6	SPECIAL DEVICE AND INSTRUCTION LISTS
7	USING SPECIAL UNITS EFFECTIVELY
	APPENDIX

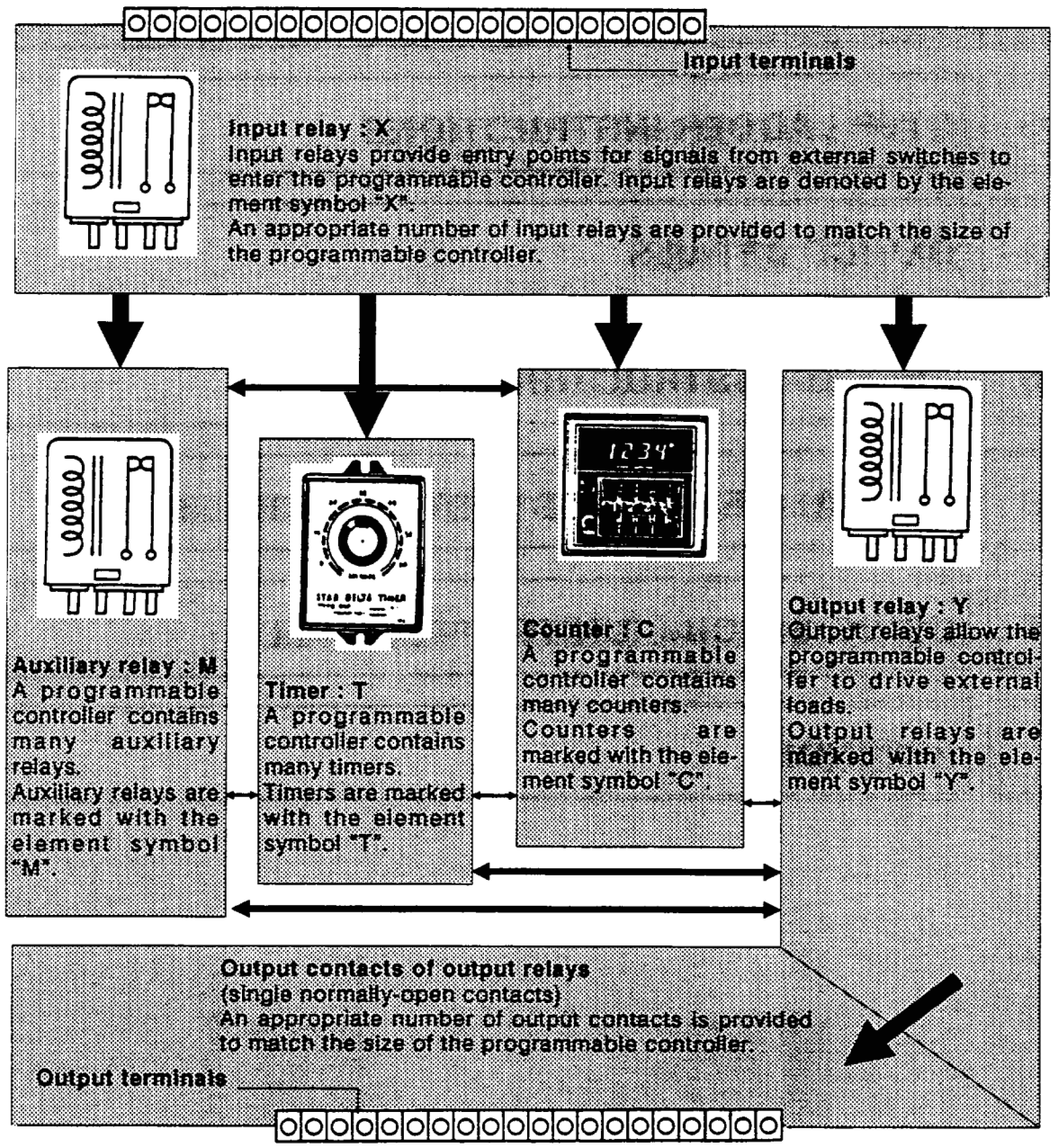
2. BASIC SEQUENCE INSTRUCTIONS

2.1 RELAYS AND TIMERS

2.1.1 OUTLINE OF MAJOR DEVICES

The table below shows that a large number of relays, timers, and counters are built into a programmable controller, each with an unlimited number of normally-open and normally-closed contacts. The sequence circuits define the connections between these contacts and coils.

The signals are transferred in the directions shown by the arrows.



2. BASIC SEQUENCE INSTRUCTIONS

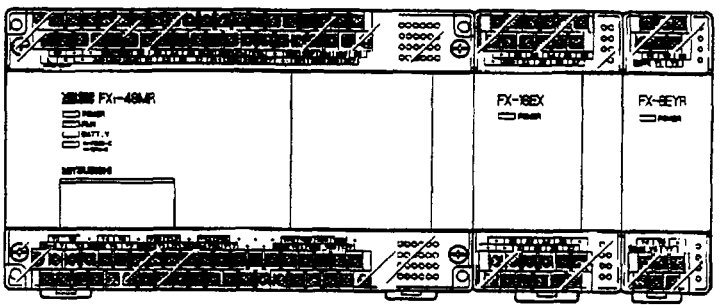
RELAYS AND TIMERS

(1) INPUT AND OUTPUT RELAYS

For the base unit, I/O numbers are assigned in octal numbers (X0 to X7, X10 to X17, ..., Y0 to Y7, Y10 to Y17, ...). Decimal numbers are used for all other devices.

For extension units and extension blocks, numbers consecutive with those assigned to the base unit are used.

Input X0 to X7, X10 to X17, X20 to X27, X30 to X37, Y30 to Y33



Output Y0 to Y7, Y10 to Y17, Y20 to Y27, X40 to X47, Y34 to Y37

Extension I/O relay numbers are assigned consecutively from the unit/block closest to the base unit.

(2) AUXILIARY RELAYS

500 auxiliary relay points (M0 to M499) are provided for general use. These are sometimes called internal relays.

524 points (M500 to M1023) are backed up by the battery. These points are called latch relays because their operation status is retained when power is lost.

(3) STATE

The state relays are used for step ladder instructions. They are explained in detail in Section 3.

They can be used as general auxiliary relays or latch relays when step ladder instructions are not used. Other relays, called annunciators, are also used.

S0 to S499 : For general use

S500 to S899 : For latch relays

S900 to S999 : For annunciator

(4) TIMERS

T0 to T199 : Can be set in the range of 0.1 to 3,276.7 sec in units of 0.1 sec.

T200 to T245 : Can be set in the range of 0.01 to 327.67 sec in units of 0.01 sec.

In addition to these timer relays, retentive timers (10 points), which can hold its operation immediately before a power failure, are also available.

(5) COUNTERS

C0 to C99 : Up counters operating in the range of 1 to 32,767.

C100 to C199 : Up counters that store the operation immediately before a power failure.

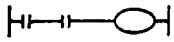
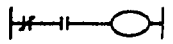
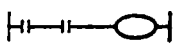
Reversible (up/down) counters operating in the range of −2,147,483,648 to +2,147,483,647 are also available (37 points + high speed counters).

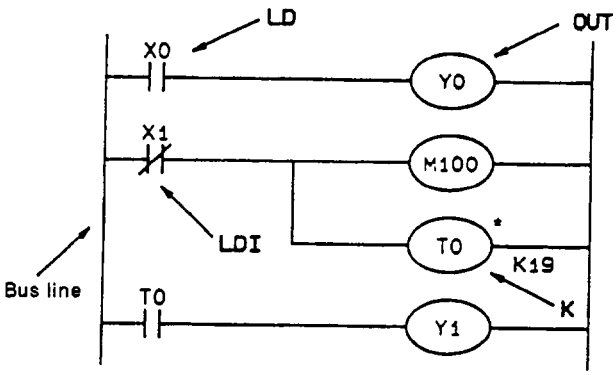
In addition to the timers, counters, and relays outlined above, data registers D which can handle numerical values are available. Data registers are explained in detail in Section 4.

2. BASIC SEQUENCE INSTRUCTIONS

2.2 LOGICAL LOAD AND OUT COILS

2.2.1 LD/LDI/OUT

Symbol and Name	Function	Format and Devices	Program Step
LD (Load)	NO (normally open) contact logical operation start	 X, Y, M, S, T, C	1
LDI (Load Inverse)	NC (normally closed) contact logical operation start	 X, Y, M, S, T, C	1
OUT (Out)	Coil drive	 Y, M, S, T, C	Y, M : 1 S, Special M : 2 T : 3 C : 3 to 5




```
0  LD   X   0   <-- Connection to bus line
1  OUT  Y   0   <--
2  LDI  X   1   <--
3  OUT  M  100  <-- Drive instruction
4  OUT  T   0   <-- Timer drive instruction
   SP   K  19   <-- Constant setting
7  LD   T   0
8  OUT  Y   1
Program steps are managed automatically
(SP : space key)
```

- All of the timers used by FX Series PCs are incremental timers. For details, see Section 4.6.1.

(1) DESCRIPTION

- LD and LDI instructions are used for contacts connected to the bus line.

  They are also combined with an ANB instruction (explained later) to start a ladder rung (circuit).
- OUT instructions are coil drive instructions for output relays, auxiliary relays, state, timers, and counters. They are not used with input relays.
- Parallel OUT instructions can be used repeatedly (OUT T0 and OUT M100 in the program above).

(2) TIMER AND COUNTER PROGRAM

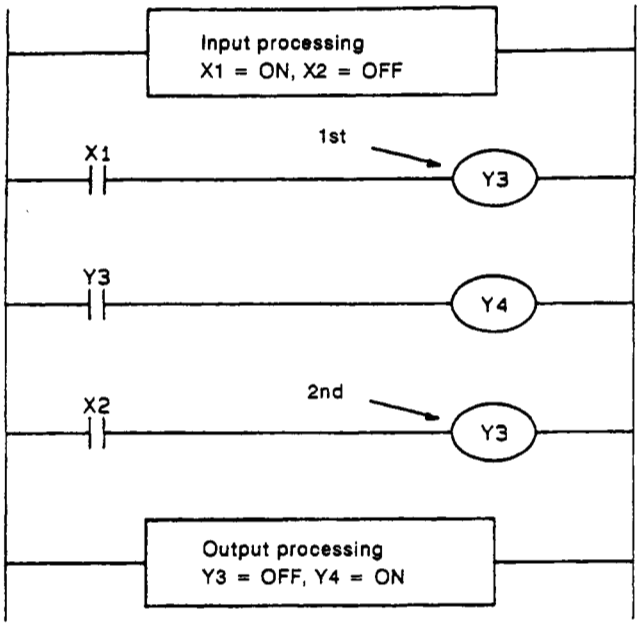
- Constant K must be set for the timer and counter coil OUT instructions.
- The setting range of constant K, the actual timer setting and the number of program steps (including the setting) for the OUT instruction are shown in the table below.

Timers, Counters	Setting Range of K	Actual Setting	Number of Steps
10 msec timer	1 to 32,767	0.01 to 327.67 sec.	3
100 msec timer		0.1 to 3,276.7 sec.	
16-bit counter	1 to 32,767	As left	3
32-bit counter	-2,147,483,648 to +2,147,483,647	As left	5

- For program examples using counters and retentive timers, see Section 2.10.1.

LOGICAL LOAD AND OUT COILS

(3) DOUBLE-COIL DESIGNATION



This is how the coil operates when the same coil is used for more than once. The program example to the left uses output Y3 twice. The operation of Y3 is explained below assuming that inputs X1 = ON and X2 = OFF.

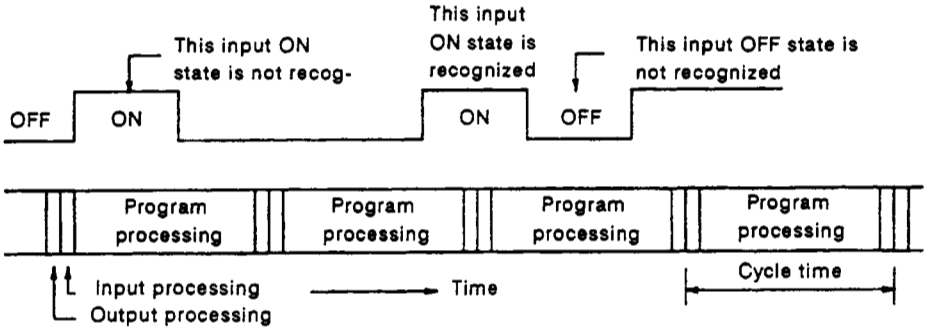
The first Y3 is turned on because X1 is ON. Its image memory is turned on and output Y4 is turned on accordingly.

However, the second Y3 is turned off because input X2 is OFF. The image memory of Y3 is rewritten to OFF.

Therefore, the final outputs are Y3 = OFF and Y4 = ON.

In double-coil designation, the coil operation designated last is effective. For details of operation processing, see Section 4.1.2.

INPUT PULSES OF SHORT DURATION CANNOT BE RECOGNIZED



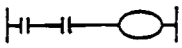
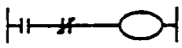
The ON or OFF duration of the PC inputs must be longer than the operation cycle time of the PC.

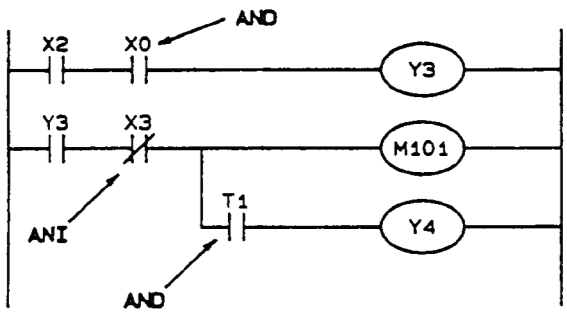
Taking the 10 msec input filter response delay into consideration, the ON and OFF duration must be longer than 20 msec if operation cycle time is 10 msec.

Therefore, input pulses of more than 25 Hz [1000/(20 + 20)] cannot be handled. There are applied instructions provided to handle such input pulses.

2.3 SERIAL CONTACTS

2.3.1 AND/ANI

Symbol and Name	Function	Format and Devices	Program Step
AND (And)	Serial connection of NO contacts	 X, Y, M, S, T, C	1
ANI (And Inverse)	Serial connection of NC contacts	 X, Y, M, S, T, C	1



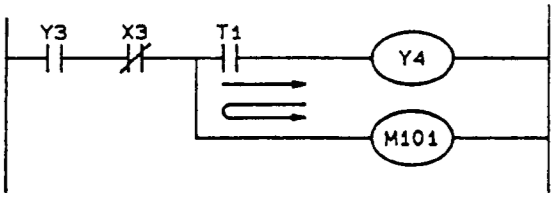

```
0  LD   X  2
1  AND  X  0  <-- Serial contact
2  OUT  Y  3
3  LD   Y  3
4  ANI  X  3  <-- Serial contact
5  OUT  M 101
6  AND  T  1  <-- Serial contact
7  OUT  Y  4  <-- Continuous OUT
```

(1) DESCRIPTION

- The AND and ANI instructions are used for the serial connection of a contact. Contacts can be connected in series as many as required.
- The output processing to a coil through a contact after writing the OUT instruction is called a follow-on output (OUT Y4 in the program example above).

  Follow-on outputs are permitted repeatedly as long as output order is correct.

Cautions :



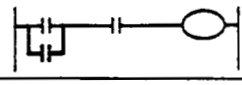
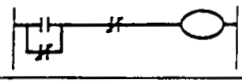
As shown in the example above, Y4 can be driven through contact Y1 after driving M101. However, if the drive order should be reversed as in the program shown to the left, it is necessary to use an MPS instruction.

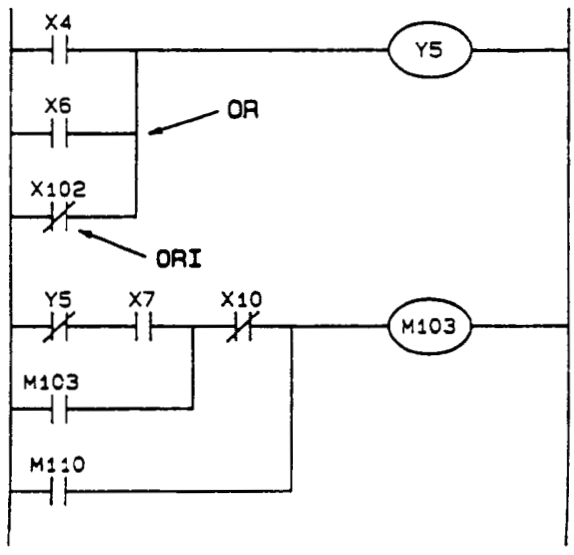
Although there are no limitations on the number of contacts that can be connected in series or follow-on output repetitions, the programming panel screen or printer will not be able to display or print the program if it exceeds the limit of the hardware.

It is recommended that each line contain up to 10 contacts and 1 coil and that the number of lines for continuous outputs is a maximum of 24 lines.

2.4 PARALLEL CONTACTS

2.4.1 OR/ORI

Symbol and Name	Function	Format and Devices	Program Step
OR (Or)	Parallel connection of NO contacts	 X, Y, M, S, T, C	1
ORI (Or Inverse)	Parallel connection of NC contacts	 X, Y, M, S, T, C	1




```
0  LD    X   4
1  OR    X   6
2  ORI   M 102
3  OUT   Y   5
4  LDI   Y   5
5  AND   X   7
6  OR    M 103
7  ANI   X  10
8  OR    M 110
9  OUT   M 103
```

Parallel connection

Parallel connection

(1) DESCRIPTION

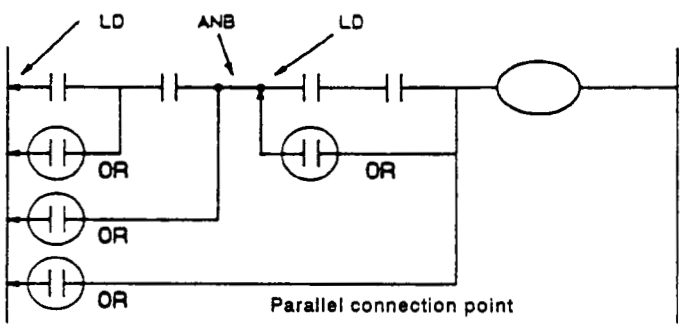
- The OR and ORI instructions are used for parallel connection of a contact.

  To connect a circuit block that contains more than one contact connected in series to another circuit block in parallel, use an ORB instruction.

- The OR or ORI instruction connects the step to the preceding LD or LDI step in parallel.

  Although there are no limitations on the number of parallel connections, the programming panel screen or printer will not be able to display or print the program if it exceeds the limit of the hardware (maximum 24 lines).

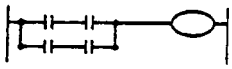
Cautions :

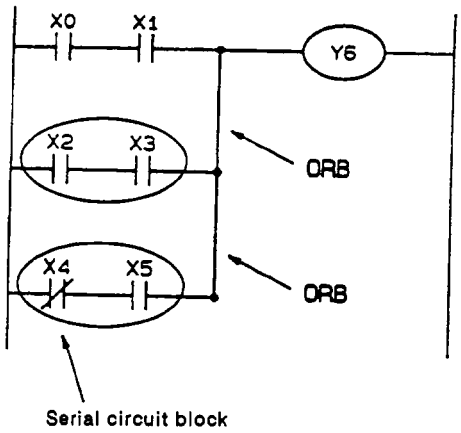


The parallel connection using the OR or ORI instruction connects the step to the preceding LD or LDI point. After an ANB instruction is designated, parallel connection is made to the LD or LDI point which is given following the ANB instruction.

2.5 PARALLEL CONNECTION OF SERIAL CIRCUIT BLOCKS

2.5.1 ORB

Symbol and Name	Function	Format and Devices	Program Step
ORB (Or Block)	Parallel connection of serial circuit blocks	 Device: None	1



Recommended Program

```
0  LD   X  0
1  AND  X  1
2  LD   X  2
3  AND  X  3
4  ORB
5  LDI  X  4
6  AND  X  5
7  ORB
8  OUT  Y  6
```

Program Not Recommended

```
0  LD   X  0
1  AND  X  1
2  LD   X  2
3  AND  X  3
4  LDI  X  4
5  AND  X  5
6  ORB
7  ORB
8  OUT  Y  6
```

(1) DESCRIPTION

- Circuits in which more than one contact is connected in series are called serial circuit blocks.

  To connect the serial circuit blocks in parallel, use an LD or LDI instruction at the beginning of a branch and an ORB instruction at the end of a branch.
- An ORB instruction is an independent instruction and is not associated with any device number.

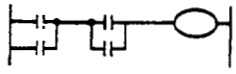
Cautions :

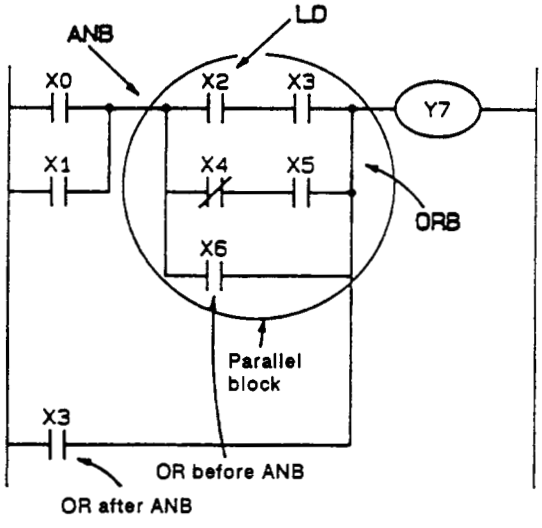
- There is no limitation to the number of parallel circuits when an ORB instruction is used for each circuit block (recommended program).
- ORB instructions can be used in batch. When using ORB instructions in batch, be sure not to use more than 8 LD or LDI instructions (program error results if attempted).

2. BASIC SEQUENCE INSTRUCTIONS

2.6 SERIAL CONNECTION OF PARALLEL CIRCUIT BLOCKS

2.6.1 ANB

Symbol and Name	Function	Format and Devices	Program Step
ANB (And Block)	Serial connection of parallel circuit blocks	 Device: None	1




```
0  LD   X  0
1  OR   X  1
2  LD   X  2   <- Branch start point
3  AND  X  3
4  LDI  X  4
5  AND  X  5
6  ORB        <- Parallel block complete
7  OR   X  6
8  ANB        <- Serial connection to preceding circuit
9  OR   X  3
10 OUT  Y  7
```

ANB and LD, OR

(1) DESCRIPTION

- Use an ANB instruction to connect the branch circuit (parallel circuit block) to the preceding circuit in series.

  To declare the starting point of the branch, use an LD or LDI instruction. After completing a parallel circuit block, connect it to the preceding block in series using an ANB instruction.

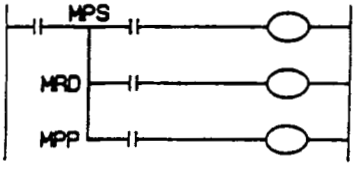
- The ANB instruction can be used as many times as necessary to connect a number of parallel circuit blocks to the preceding block in series.

- It is possible to use ANB instructions in batches. However, the maximum allowable of use (8 times) of LD and LDI instructions must be taken into consideration as with the case of an ORB instruction.

2. BASIC SEQUENCE INSTRUCTIONS

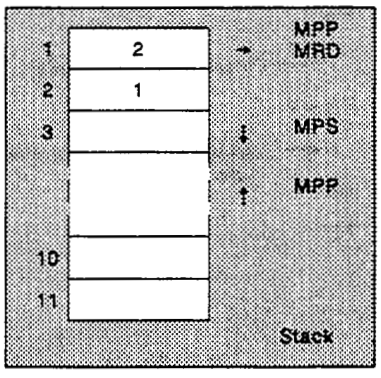
2.7 MULTIPLE OUTPUT CIRCUITS

2.7.1 MPS/MRD/MPP

Symbol and Name	Function	Format and Devices	Program Step
MPS (Push)	Push down stack		1
MRD (Read)	Read from stack		1
MPP (Pop)	Pop up stack		1

This group of instructions allow points in the circuit to be stored so that circuit connections can be made later.

The PC has 11 memory areas, called a stack, where operation results are temporarily stored.



When an MPS instruction is executed, the result of the operation obtained at this time is stored in the first stage of the stack. Execute an MPS instruction again and the latest operation result is stored in the first stage of the stack. The previously stored data in the stack is shifted to the second stage.

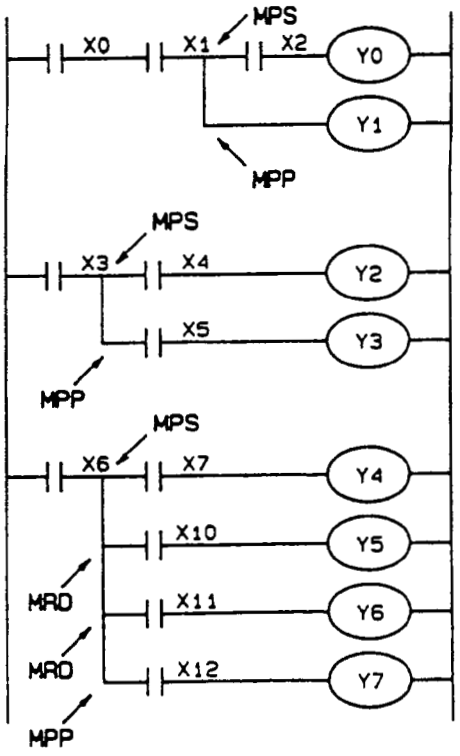
The data at the first stage dictates where the next device is to be connected from in the ladder circuit.

An MPP instruction shifts the data in the stack to the upper stage. Data in the first stage is read and lost.

An MRD instruction is used to read the latest stack data. With this instruction, data does not shift in the stack.

The MPS, MRD, and MPP instructions does not have any PC devices associated with them.

(1) SIMPLE CIRCUIT EXAMPLE (1-STAGE STACK)



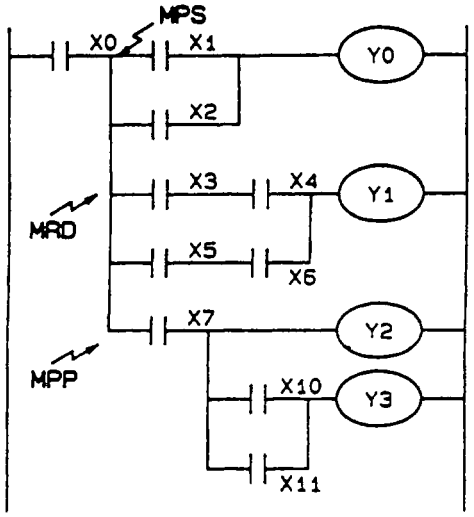

```
 0  LD    X  0
 1  AND   X  1
 2  MPS
 3  AND   X  2
 4  OUT   Y  0
 5  MPP
 6  OUT   Y  1
 7  LD    X  3
 8  MPS
 9  AND   X  4
10  OUT   Y  2
11  MPP
12  AND   X  5
13  OUT   Y  3
14  LD    X  6
15  MPS
16  AND   X  7
17  OUT   Y  4
18  MRD
19  AND   X 10
20  OUT   Y  5
21  MRD
22  AND   X 11
23  OUT   Y  6
24  MPP
25  AND   X 12
26  OUT   Y  7
```

In this program example, only the first stage stack is used.

2. BASIC SEQUENCE INSTRUCTIONS

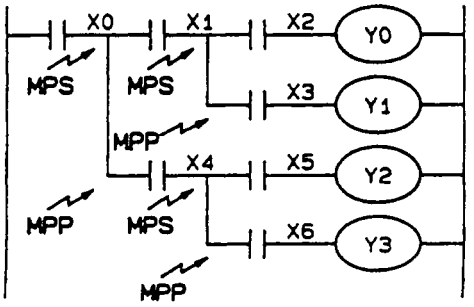
MULTIPLE OUTPUT CIRCUITS

(2) 1-STAGE STACK WITH ANB AND ORB INSTRUCTIONS



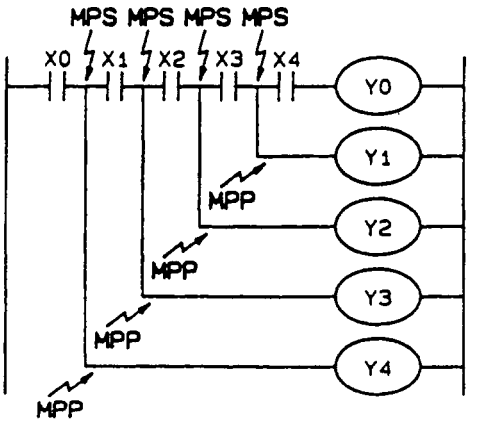

```
 0  LD   X  0        12  ANB
 1  MPS              13  OUT  Y  1
 2  LD   X  1        14  MPP
 3  OR   X  2        15  AND  X  7
 4  ANB              16  OUT  Y  2
 5  OUT  Y  0        17  LD   X 10
 6  MRD              18  OR   X 11
 7  LD   X  3        19  ANB
 8  AND  X  4        20  OUT  Y  3
 9  LD   X  5
10  AND  X  6
11  ORB
```

(3) 2-STAGE STACK




```
 0  LD   X  0         9  MPP
 1  MPS              10  AND  X  4
 2  AND  X  1        11  MPS
 3  MPS              12  AND  X  5
 4  AND  X  2        13  OUT  Y  2
 5  OUT  Y  0        14  MPP
 6  MPP              15  AND  X  6
 7  AND  X  3        16  OUT  Y  3
 8  OUT  Y  1
```

(4) 4-STAGE STACK

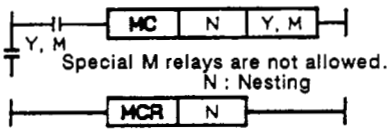


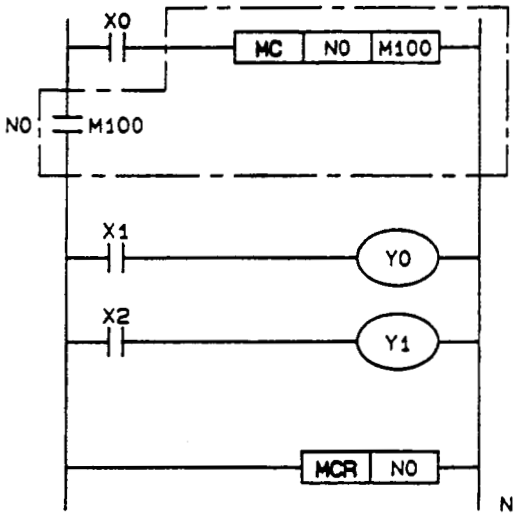

```
 0  LD   X  0         9  OUT  Y  0
 1  MPS              10  MPP
 2  AND  X  1        11  OUT  Y  1
 3  MPS              12  MPP
 4  AND  X  2        13  OUT  Y  2
 5  MPS              14  MPP
 6  AND  X  3        15  OUT  Y  3
 7  MPS              16  MPP
 8  AND  X  4        17  OUT  Y  4
```

Note: • *At any programming step, the difference between the numbers of MPS and MPP instructions must 11 or less. These numbers must agree with each other within a program.*

2.8 COMMON SERIAL CONTACTS

2.8.1 MC/MCR

Symbol and Name	Function	Format and Devices	Program Step
MC (Master Control)	Denotes the start of the MC control block		3
MCR (Master Control Reset)	Denotes the end of the MC control block		2




```
0  LD   X  0
1  MC   N  0   ) An MC is a 3-step
   SP   M 100  ) instruction
4  LD   X  1
5  OUT  Y  0
6  LD   X  2
7  OUT  Y  1
8  MCR  N  0   ← An MCR is a 2-step instruction
```

- The nesting level for N is 0 to T.
  SP is the symbol for the space key on the programming panel.
- Special relays cannot be used as object devices for M.

N : Nesting level (0 to 7)
SP : Space key

(1) DESCRIPTION

- Input X : ON

  All instructions between MC and MCR are executed.

  Input X : OFF

  Present status is retained for :

  - Retentive timers, counters, and devices driven by SET/RST instructions

  Devices turned OFF are :

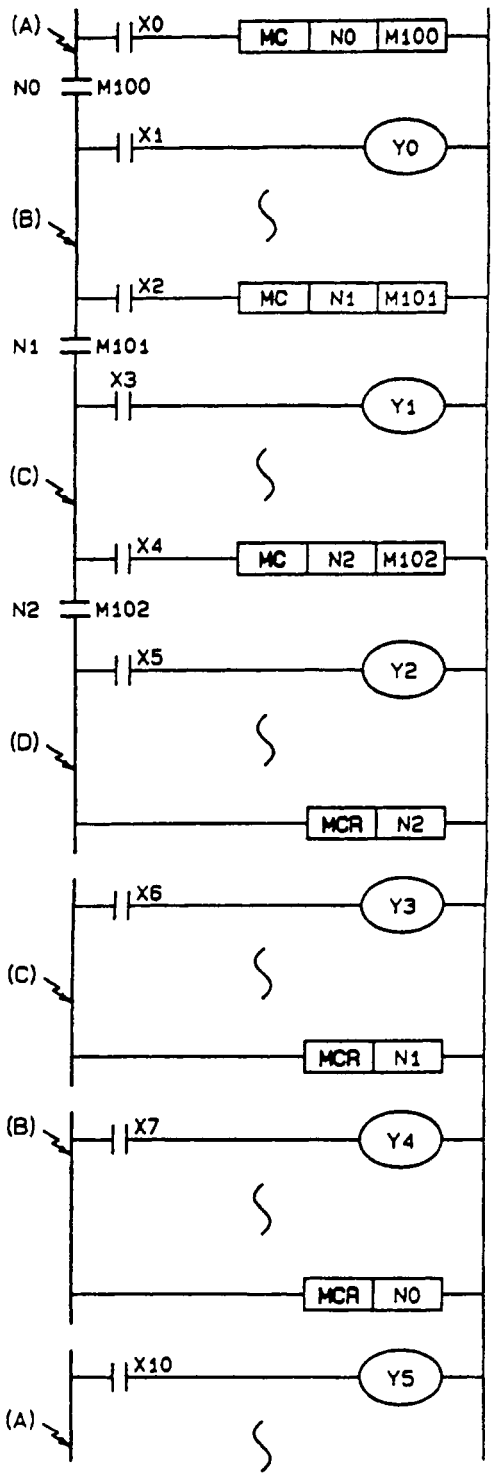
  - Timers and devices driven by OUT instruction.

- After the execution of an MC instruction, the bus line (LD, LDI point) shifts to a point after the MC instruction. An MCR instruction returns this to the original bus line. After the MC instruction is designated, it is necessary to write an MCR N0 instruction.
- An MC instruction can be used as many times as necessary by changing device numbers Y and M. If the same device number is used, it is processed as a double-coil designation as explained in Section 2.2 (4).
- To use an MC instruction while a previous MC instruction is active, increase the nesting level number (N).

  The nesting level can be decreased by designating an MCR instruction.

COMMON SERIAL CONTACTS

(2) NESTING LEVEL



Level N0

Bus line (B) becomes active when X0 is ON.

Level N1

Bus line (C) becomes active when both X0 and X2 are ON.

Level N2

Bus line (D) becomes active when all of X0, X2, and X4 are ON.

Level N1

With MCR N2 executed, bus line (C) is restored.

Level N0

With MCR N1 executed, bus line (B) is restored.

Initial state

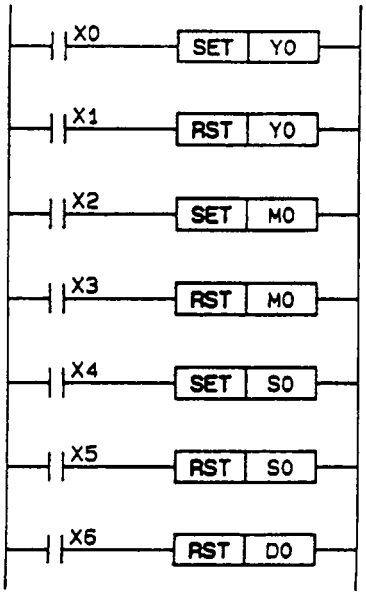
With MCR N0 executed, the initial bus line (A) is restored.

Output Y5 is turned ON/OFF according to ON/OFF state of X10 and regardless of ON/OFF state of X0, X2, and X4.

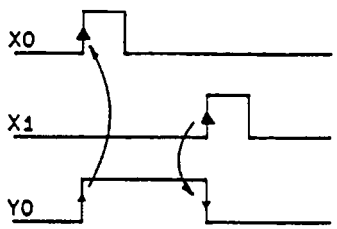
2.9 RETAINING AND RESETTING OPERATION STATUS

2.9.1 SET/RST

Symbol and Name	Function	Format and Devices	Program Step
SET (Set)	Setting a device ON	SET Y, M, S	Y, M : 1 S, Special M : 2
RST (Reset)	Resetting a device to OFF. Clearing registers	RST Y, M, S, D, V, Z	D, V, Z, Special D : 3




```
0  LD   X  0
1  SET  Y  0
2  LD   X  1
3  RST  Y  0
4  LD   X  2
5  SET  M  0
6  LD   X  3
7  RST  M  0
8  LD   X  4
9  SET  S  0
11 LD   X  5
12 RST  S  0
14 LD   X  6
15 RST  D  0
```



(1) DESCRIPTION

- Once X0 is turned ON, Y0 operates and remains ON even after X0 is turned OFF.

  Once X1 is turned ON, Y0 turns OFF and remains in "non-operating" status even after X1 is turned OFF.

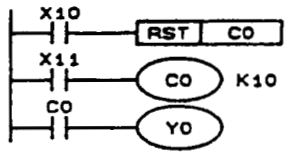
  These are also true for M and S.

- SET and RST instructions can be used for the same device as many times as necessary. Although they can be used in any order, the instruction executed last is effective.
- The RST instruction can be used to reset the data in data register D and index registers V and Z to "0". The same effect is obtained by using the constant K0 transfer instruction.

2.10 COUNTERS AND TIMERS

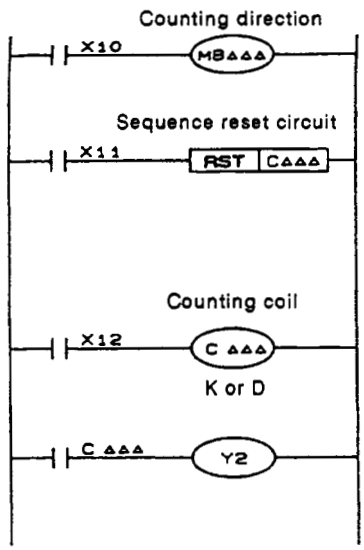
2.10.1 OUT/RST

Symbol and Name	Function	Format and Devices	Program Step
OUT (Out)	Driving timer coil Driving counter coil		32-bit counter: 5 steps Other: 3 steps
RST (Reset)	Resetting output contact Resetting present data to "0"		2



(1) RETENTIVE TIMERS (1 msec TIMERS, 100 msec TIMERS)

- Counter Co counts up each time input X11 turns ON until Co equals its setting of 10. At this point contact Co turns ON which in turns ON Yo.
  Co and Yo are reset when X10 is turned ON.
  Counters C100 to C135 are backed by battery so that their contents remain even if power is turned OFF.



(2) HIGH-SPEED COUNTERS (SEE SECTION 4.7.2)

- The 1-phase, 1-input counters (C235 to C245) are used in combination with special auxiliary relays (M8235 to M8245) which designate the counting direction.

  Count down when X10 is ON
  Count up when X10 is OFF

- The output contact of counter CΔΔΔ is reset and the current value of the counter is reset to "0" when X11 is turned ON.

  Counters (C241, C244, ...) with a reset input operate similarly without requiring any further programming when the corresponding reset input is turned ON.

- When X12 is turned ON, ON/OFF of the corresponding counter input (X0 to X5) is counted.

  With counters (C244, C249, C254,), counting does not start unless the corresponding start input is turned ON.

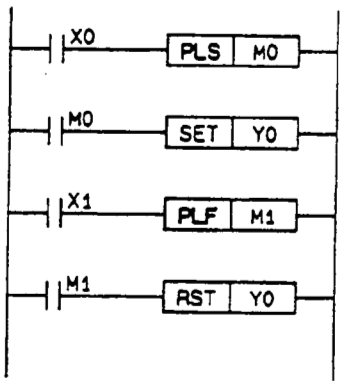
- The output contact is set when the counter value increases to the setting (K or D data). It is reset when the counter value decreases to that setting.

2. BASIC SEQUENCE INSTRUCTIONS

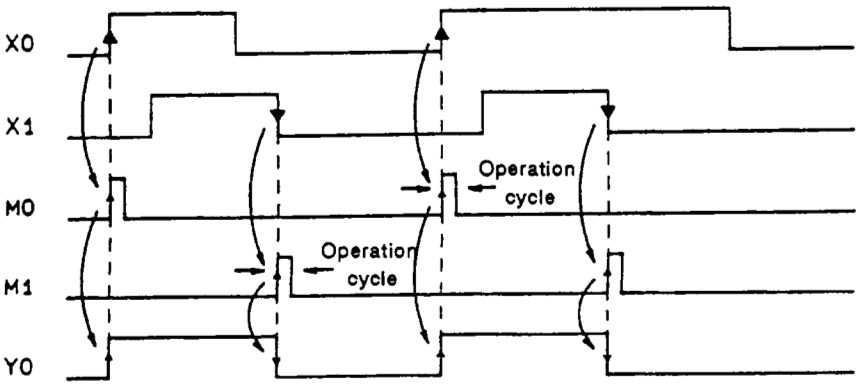
2.11 PULSE OUTPUT

2.11.1 PLS/PLF

Symbol and Name	Function	Format and Devices	Program Step
PLS (Pulse)	Leading edge triggered pulse		2
PLF (PLF)	Trailing edge triggered pulse		2




```
0  LD   X  0
1  PLS  M  0   <-- 2-step instruction
3  LD   M  0
4  SET  Y  0
5  LD   X  1
6  PLF  M  1   <-- 2-step instruction
8  LD   M  1
9  RST  Y  0
```



(1) DESCRIPTION

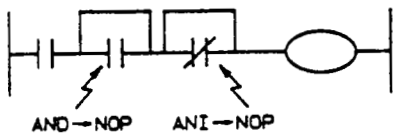
- When a PLS instruction is executed, object devices Y and M operate one operation cycle after the drive input signal has turned ON.
- When a PLF instruction is executed, object devices Y and M operate one operation cycle after the drive input signal has turned OFF.
- Special relays cannot be object devices of PLS or PLF.
- When the PC status is changed from RUN to STOP to RUN with the input signal ON, PLS M0 is operated while for example, PLS M600 (backed up by the battery) is not operated. This is because M600 is a retainable relay and its operation is retained as when the PC was in the STOP state.

2.12 NO PROCESSING INSTRUCTION

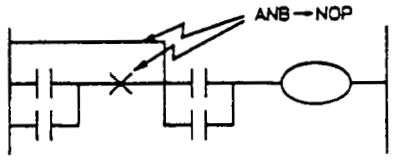
2.12.1 NOP

Symbol and Name	Function	Format and Devices	Program Step
NOP (NOP)	No operation	Device: None	1

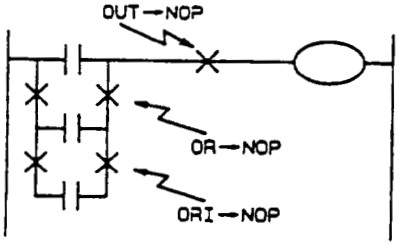
CHANGING CIRCUITS USING NOP INSTRUCTION



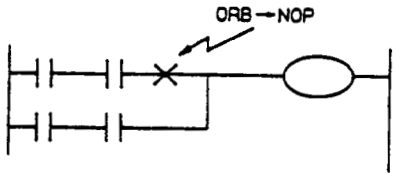
(a) Shorting contacts



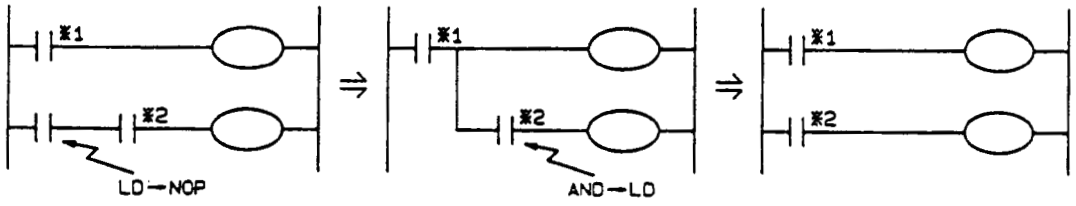
(b) Shorting entire preceding circuit
(This may cause circuit error)



(c) Cutting circuits



(d) Cutting an entire preceding circuit
(This may cause circuit error)



(e) Connecting to preceding OUT instruction

(1) DESCRIPTION

- Writing NOP instructions in the middle of a program minimizes step number changes when changing or editing a program.

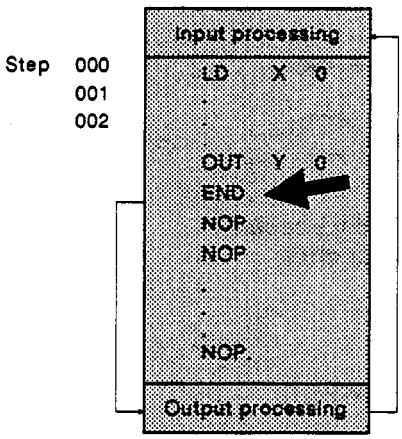
  It is also possible to change a circuit by replacing programmed instructions with NOP instructions.
- Changing an LD, LDI, ANB, or ORB instruction into an NOP instruction changes the circuit considerably.
- After the program all clear operation is executed, all of the instructions in the program are over-written with NOPS.

2. BASIC SEQUENCE INSTRUCTIONS

2.13 PROGRAM END

2.13.1 END

Symbol and Name	Function	Format and Devices	Program Step
END (End)	Input/output processing and returning to step 0	 Device: None	1



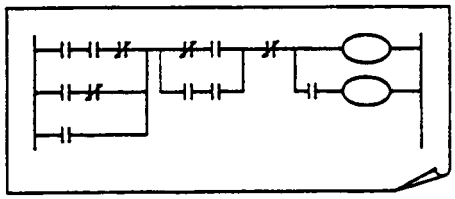
The PC repeats the execution of input processing, program, and output processing. By writing an END instruction, the steps after the END instruction are skipped and output processing is executed directly.

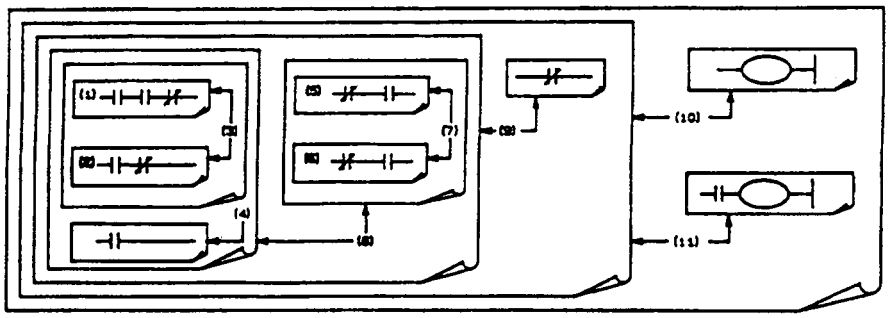
Inserting END instructions in the middle of the program helps program debugging as the section after the END instruction is disabled and isolated from the area that is being checked. Delete the END instructions from the blocks which have already been checked.

The watchdog timer is refreshed when the END instruction is executed.

PROGRAMMING TIP

A program is executed generally from top to bottom and from left to right.





1	GENERAL HARDWARE AND INSTALLATION
2	BASIC SEQUENCE INSTRUCTIONS
3	STEP LADDER INSTRUCTIONS
4	DEVICE DETAILS
5	APPLIED INSTRUCTION
6	SPECIAL DEVICE AND INSTRUCTION LISTS
7	USING SPECIAL UNITS EFFECTIVELY
	APPENDIX

3. STEP LADDER INSTRUCTIONS

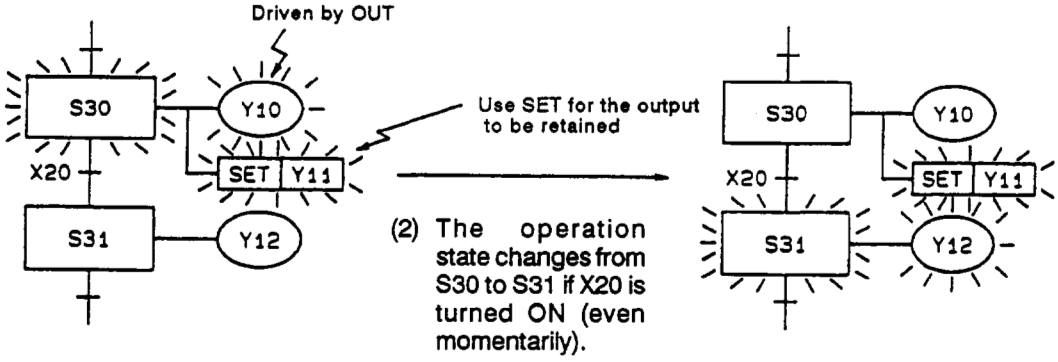
3.1 HOW TO READ SEQUENTIAL FUNCTION CHARTS

3.1.1 STATE FUNCTIONS

Writing a relay ladder sequence for sequential process control requires some expertise. Also the completed sequence can be difficult to read.

However, when machine operation is expressed using a sequence chart (called a Sequential Function Chart), programming is made much simpler.

A device (called "state") is the important element in making a Sequential Function Chart. A total of 900 points of state in the range of S0 to S899 are used with S0 to S19 assigned for reserved functions, (which will be explained in a later section). For example, S0 to S9 are called initial states and are used as the head states of the sequential function chart.



(1) Outputs Y10 and Y11 operate when state S30 operates. The program is waiting for the operation of transfer condition X20.

(2) The operation state changes from S30 to S31 if X20 is turned ON (even momentarily).

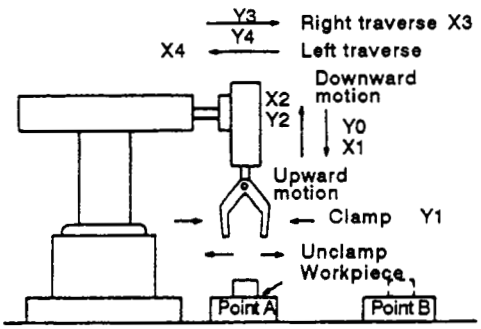
(3) State transfer is operated from S30 to S31, causing Y10 to stop operating and Y12 to start operating.

Y11 continues to operate because it is driven by the SET instruction.

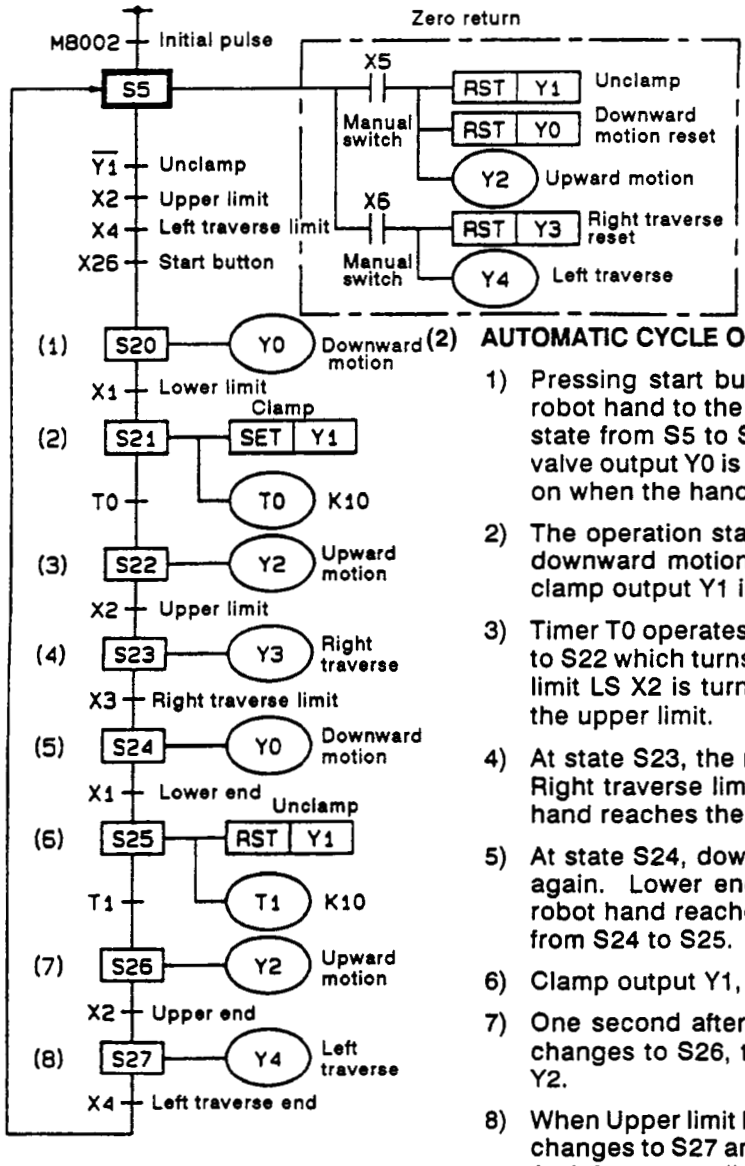
Note:
- *Transfer conditions often take the form of a single contact, however, in actual fact, a combination of X, Y, M, S, T, C devices are possible in the form of parallel or serial connections.*
- *For output loads (Y, M, S, T, C) and applied instructions driven by the states, it is possible to have these connected in various forms of series and parallel combinations.*

HOW TO READ SEQUENTIAL FUNCTION CHARTS

3.1.2 SINGLE FLOW



To make the explanation of function charts simple, a pick-and-place operation, moving a workpiece from point A to point B with the robot hand as shown on the left, is used as an example. A double-solenoid valve is used for right/left traverse and up/down motion with a single solenoid valve used for clamp operation. The double-solenoid valve holds an actuator at the actuated position even after the signal for the actuator motion has turned off. The workpiece is clamped only while the solenoid is energized.



(1) MANUAL OPERATION

This is for example the manual operation required to return the robot hand to the initial position before starting the automatic sequence.

State S5 is driven by special auxiliary relay M8002, which is pulsed on when the PC operation status is changed from STOP to RUN.

(2) AUTOMATIC CYCLE OPERATION

1) Pressing start button X26 after manually locating the robot hand to the intial position changes the operation state from S5 to S20. The downward motion solenoid valve output Y0 is turned on. Lower limit LS X1 is turned on when the hand reaches the lower limit position.
2) The operation state changes from S20 to S21 and the downward motion output Y0 is turned off. After that, clamp output Y1 is set.
3) Timer T0 operates one second later to change the state to S22 which turns on upward motion output Y2. Upper limit LS X2 is turned on when the robot hand reaches the upper limit.
4) At state S23, the right traverse output Y3 is turned on. Right traverse limit LS X3 is turned on when the robot hand reaches the right traverse limit.
5) At state S24, downward motion output Y0 is turned on again. Lower end limit LS X1 is turned on when the robot hand reaches the lower limit, changing the state from S24 to S25.
6) Clamp output Y1, which has been set, is reset.
7) One second after the clamp output is reset, the state changes to S26, turning on the upward motion output Y2.
8) When Upper limit LS X2 is turned on, the operation state changes to S27 and left traverse output Y4 is turned on. As left traverse limit LS X4 is turned on, the operation state returns to S5, allowing the next cycle to start.

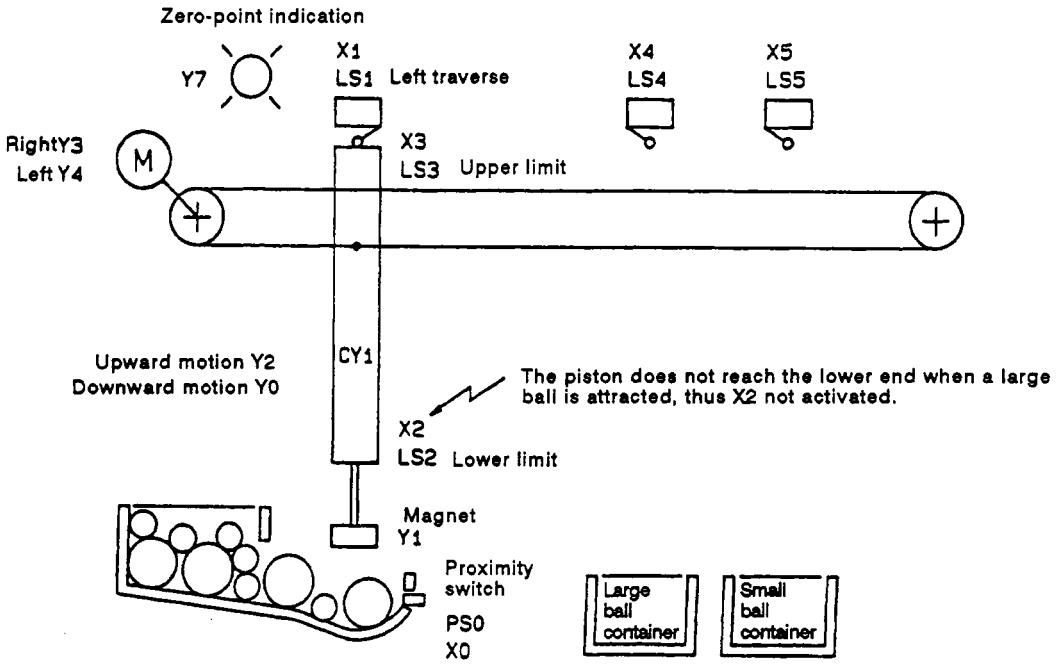
HOW TO READ SEQUENTIAL FUNCTION CHARTS

3.1.3 SELECTIVE BRANCH AND MERGE EXAMPLE

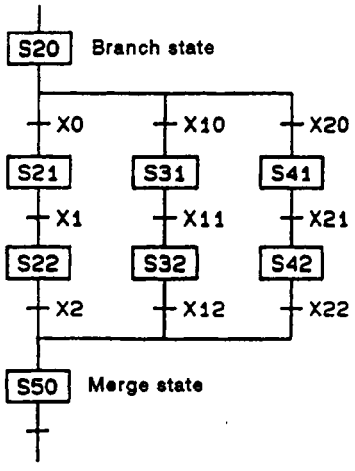
The following illustration shows an example of how to sort large and small balls with a conveyor.

With the initial position set at the upper left point, the sorting cycle is executed in the following order:

downward motion, ball hold, upward motion, right traverse, downward motion, release, upward motion, and left traverse. Large balls and small balls are recognized by the activation of lower end position LS2 switch (OFF for large balls and ON for small balls).



SELECTIVE BRANCHES AND MERGES

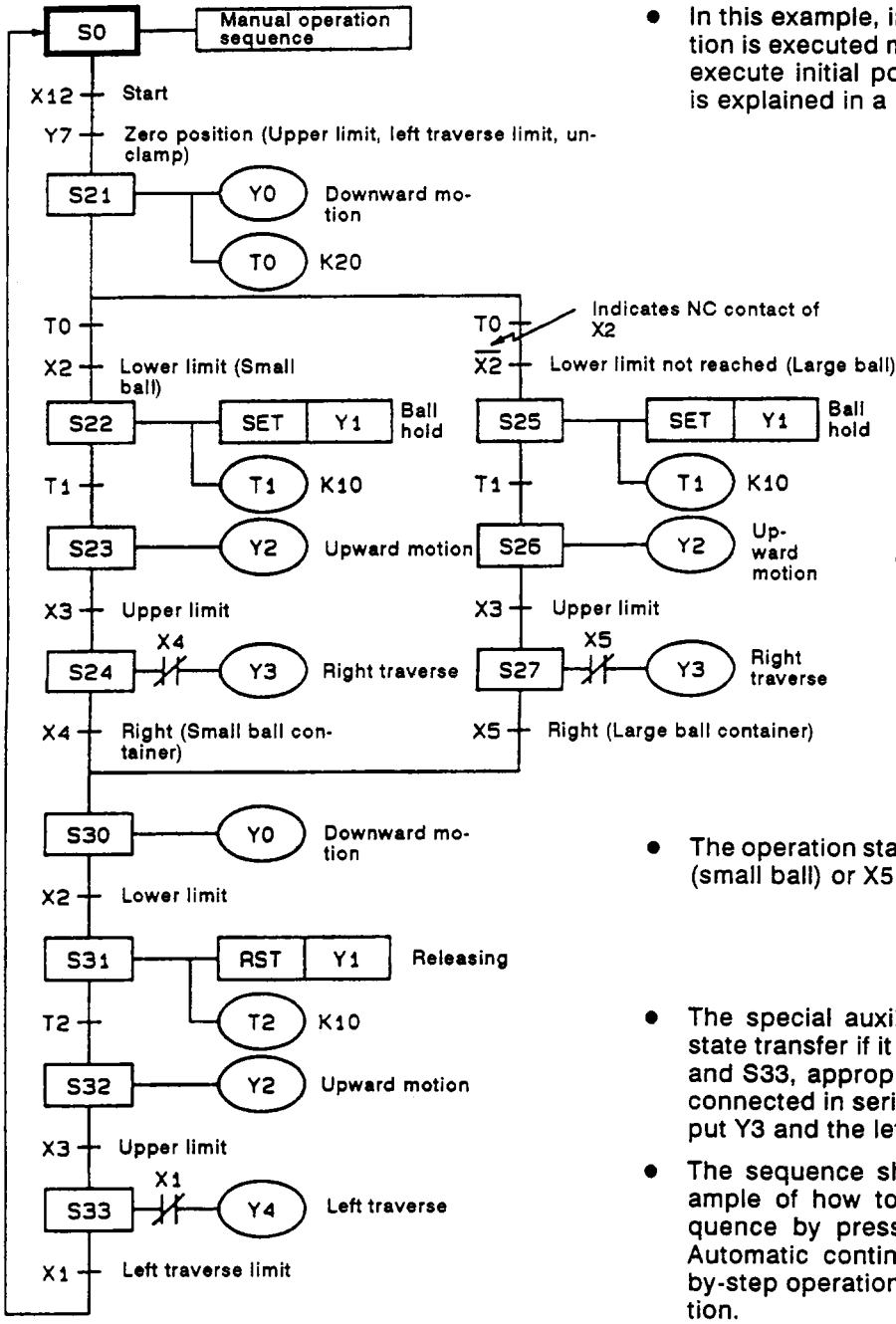


- The process of selecting a single flow from multiple flows is called a selective branch.
- As this is a selective branch example, X0, X10, and X20 must not be turned on simultaneously.
- When X0 is turned ON while S20 is operating, the operation state changes to S21, and S20 stops operating.

  If X10 or X20 is turned ON after this state transfer, X31 or X41 is not turned ON.
- The merge state S50 is activated by S22, S32, or S42.

HOW TO READ SEQUENTIAL FUNCTION CHARTS

The Sequential Function Chart for the sorting process is illustrated below.



- In this example, initial position return operation is executed manually. The procedure to execute initial position return automatically is explained in a later section.

- The flow is selected according to ball size (large or small); left flow for small ball (X2=ON) and right flow for large ball (X2=OFF).

- The operation state changes to S30 when X4 (small ball) or X5 (large ball) is turned ON.

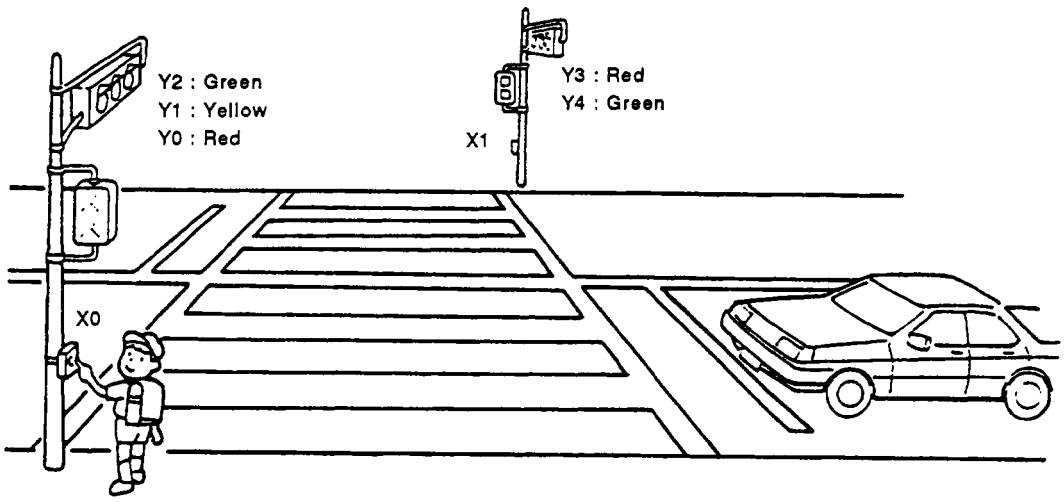
- The special auxiliary relay M8040 disables state transfer if it is turned on. For S24, S27 and S33, appropriate interlock contacts are connected in series to the right traverse output Y3 and the left traverse output Y4.
- The sequence shown on the left is an example of how to execute a one cycle sequence by pressing the start button X12. Automatic continuous operation and step-by-step operation is explained in a later section.

HOW TO READ SEQUENTIAL FUNCTION CHARTS

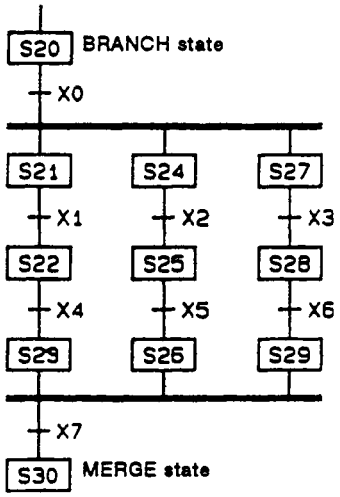
3.1.4 PARALLEL BRANCH AND MERGE EXAMPLE

Parallel branches are necessary in cases where for example, individual components A, B and C are processed independently and are then assembled together.

This parallel branch format is applicable also to the sequential operation of signals in the example case shown below.



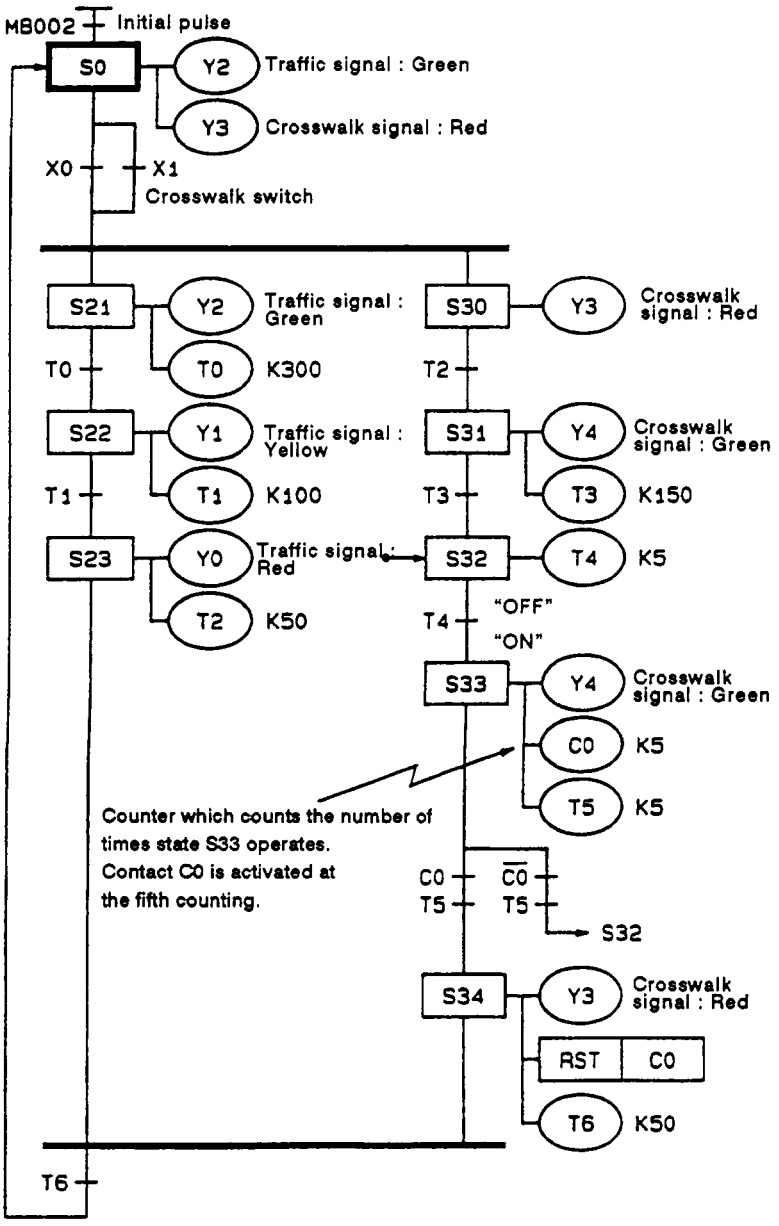
PARALLEL BRANCHES AND MERGES



- A Parallel branch is a sequence in which multiple branches of flow proceed simultaneously.
- In the figure shown on the left, X0 is turned on when S20 is operating. S21, S24 and S27 start operating simultaneously, initiating the operation flow of each branch.
- When X7 is turned on only after the operation of all the branch flows have finished. Merge state S30 starts operating. All the states S23, S26 and S29 are disabled.
- A MERGE of this kind is sometimes referred to as a queuing MERGE.

HOW TO READ SEQUENTIAL FUNCTION CHARTS

The Sequential Function Chart for a crosswalk signal is illustrated below. In this example, parallel flow is repeated with the green light of the pedestrian flashing being created in the form of a loop.



- When the PC status changes from STOP to RUN, the initial state S0 operates to light the green traffic signal and the red crosswalk signal. (via M8002)
- When crosswalk switch X0 or X1 is pressed, the state changes to S21 (traffic) and S30 (crosswalk). In these states, the light state remains unchanged (green for the traffic signal and red for the crosswalk signal).
- After 30 seconds, the yellow traffic signal lights. The red traffic signal lights 10 seconds later.
- When the red traffic signal lights, timer T2 begins operating. The green crosswalk signal lights after timer T2 counts 5 seconds.
- 15 seconds later, the green crosswalk signal begins to flash (S32 = OFF, S33 = ON).
- While the green crosswalk signal flashes, states S32 and S33 repeat operation. The state changes to S34 when counter C0 (setting = 5) operates, lighting the red crosswalk signal. The state then changes to the initial state (S0) 5 seconds after the red crosswalk signal lights.
- Crosswalk switches X0 and X1 are inoperative once the operation has started to flow.

3.2 PROGRAMMING

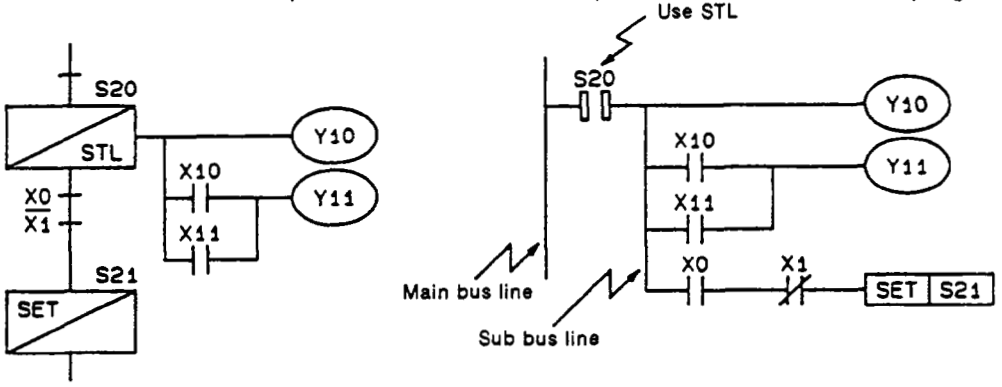
3.2.1 STATE PROGRAM EXAMPLE

Illustration (a) shows an example of a state program. Each state consists of some loads transfer destinations, and transfer conditions.

Illustration (b) shows a step ladder in which the sequential function chart is expressed in relay sequence chart form.

It is possible to write a program using either a sequential function chart or a step ladder chart. In either case, the program should be written in the order of load drive processing then state transfer processing.

Sequential function charts and step ladder charts are two representations of the same program. Their



(a) Sequential function

(b) Step ladder chart

0	STL	S20
1	OUT	Y10
2	LD	X10
3	OR	X11
4	OUT	Y11
5	LD	X 0
6	ANI	X 1
7	SET	S21

The SET or RST instruction for a state isa 2-step instruction.

The program for the sequence function chart or step ladder chart is shown on the left.

An STL instruction is the NO contact instruction connected to the main bus line. A coil can be connected directly to the sub-bus line created by the STL contact or it can be driven through some contacts.

Use the LD (LDI) instruction for the contact connected to the sub-bus line. Use a RET instruction to return the sub-busline to the main bus line. If a state S is driven through the STL contact, the state of that STL contact is reset automatically.

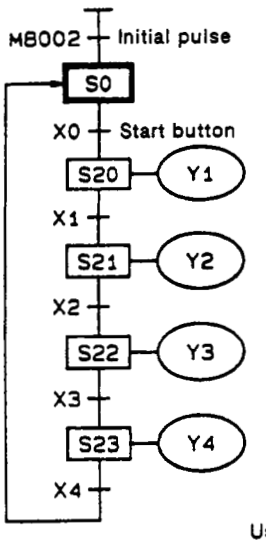
advantage is that they allow the user easy programming by isolating one state from another so that the user only have to consider the circuits of only one state at one time. The order of the states is not important. However, do not forget to write a RET instruction at the end of a program when using STL instructions.

Note: • *An MC instruction cannot be used in the STL circuit.*

• *An MPS instruction cannot be used immediately after the STL contact.*

PROGRAMMING

3.2.2 INITIAL STATE PROGRAMMING



- The state at the head of a sequential function chart is called an initial state. States S0 to S9 are used as initial states.
- An initial state is driven by another state (S23 in this case). However, at the beginning of operation, the initial state should be driven by different means.
- In this example, the initial state is first driven by the special auxiliary relay M8002 which operates momentarily when the PC status is changed from STOP to RUN.
- A state other than an initial state must be driven by other state using an STL instruction. These states can be driven only by state elements.
- When programming, the initial state must be programmed before any other states in the sequence.

Use SET

```
0  LD    M8002   ) Initial drive of initial
1  SET   S   0   ) state
3  STL   S   0   )
4  LD    X   0   ) State S0
5  SET   S   20  )
7  STL   S   20  )
8  OUT   Y   1   ) State S20
9  LD    X   1   )
10 SET   S   21  )
12 STL   S   21  )
13 OUT   Y   2   ) State S21
14 LD    X   2   )
15 SET   S   22  )
17 STL   S   22  )
18 OUT   Y   3   ) State S22
19 LD    X   3   )
20 SET   S   23  )
22 STL   S   23  )
23 OUT   Y   4   ) State S23
24 LD    X   4   )
25 OUT   S   0   ) ← Use OUT (See Section 3.2.6)
27 RET
28 END
```

Use RET at the end of a series of STL instructions

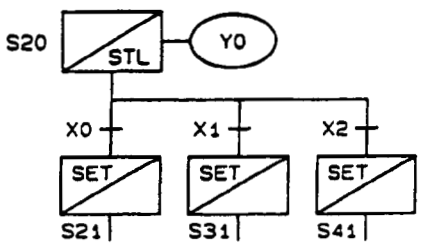
MEMO

- If states S500 to S899 are used, the status of the sequence is backed up by the battery. This allows the machine to be restarted with these states when power supply is restored from failure.
- In such a case, re-consider the way the initial state is driven. Better ways of driving the initial state are shown at the later pages.
- Use the RET instruction always at the end of a series of STL instructions, as shown on the left.
- If the RET instruction is not given, the PC regards the LD instruction of step 0 as a connectiono to the sub-bus of STL S23 in its cyclic operation and will cause a malfunction. (Actually, a program error occurs and the operation is disabled.)

PROGRAMMING

3.2.3 SELECTIVE BRANCH AND MERGE PROGRAMMING

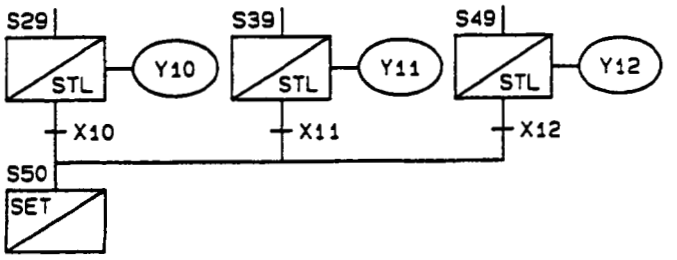
(1) SELECTIVE BRANCH PROGRAMMING




```
STL   S20
OUT   Y 0    ——— Drive processing
LD    X 0
SET   S21    ——— Transfer to the 1st branch state
LD    X 1
SET   S31    ——— Transfer to the 2nd branch state
LD    X 2
SET   S41    ——— Transfer to the 3rd branch state
```

As in the general case, the state drives a load before setting the transfer conditions. It is necessary to program the setting of the states in the order of left to right as shown in the function chart.

(2) MERGE STATE PROGRAM



In branch/merge transfer programming, do not use MPS, MRD, MPP, ANB, or ORB instructions.

```
STL   S 29
OUT   Y 10   — Output processes
  :
STL   S 39
OUT   Y 11   — Output processes
  :
STL   S 49
OUT   Y 12   — Output processes

STL   S 29 ┐
LD    X 10 │ Merge transfer from the 1st
SET   S 50 ┘ branch flow
STL   S 39 ┐
LD    X 11 │ Merge transfer from the 2nd
SET   S 50 ┘ branch flow
STL   S 49 ┐
LD    X 12 │ Merge transfer from the 3rd
SET   S 50 ┘ branch flow
```

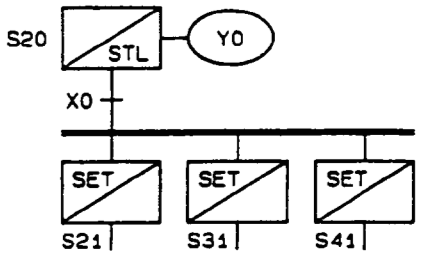
Program the output processes for each state before merging. After that, it is necessary to program the merge transfer processes sequentially from left to right.

This rule is added to allow the SFC screen to be created automatically.

PROGRAMMING

3.2.4 PARALLEL BRANCH AND MERGE PROGRAMMING

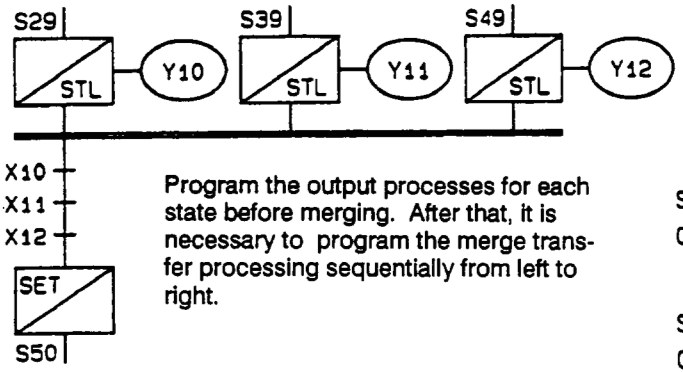
(1) PARALLEL BRANCH PROGRAMMING




```
STL   S20
OUT   Y 0    —— Drive processing
LD    X 0
SET   S21    —— Transfer to the 1st parallel branch state
SET   S31    —— Transfer to the 2nd parallel branch state
SET   S41    —— Transfer to the 3rd parallel branch state
```

As in the general case, the state drives a load before setting the transfer conditions. It is necessary to program the setting of the states in the order of left to right as shown in the function chart.

(2) MERGE STATE PROGRAM



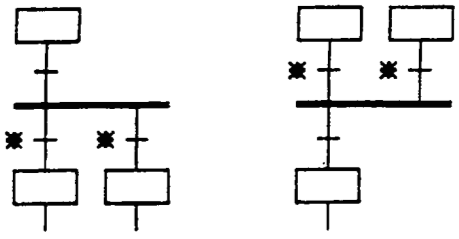
Program the output processes for each state before merging. After that, it is necessary to program the merge transfer processing sequentially from left to right.

```
(
STL   S29
OUT   Y10    — Output processing
(
STL   S39
OUT   Y11    — Output processing
(
STL   S49
OUT   Y12    — Output processing
(
STL   S29
STL   S39
STL   S49
LD    X10
AND   X11
AND   X12
SET   S50
(
```

Merge transfer processing

STL instructions written consecutively indicate a parallel merge. The STL instruction can used eight times consecutively.

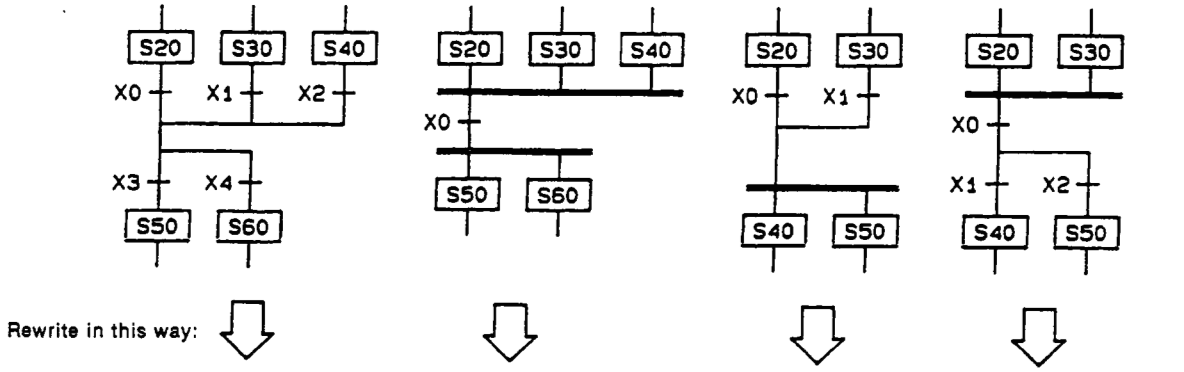
Ref.: • *The following programs are not permitted.*



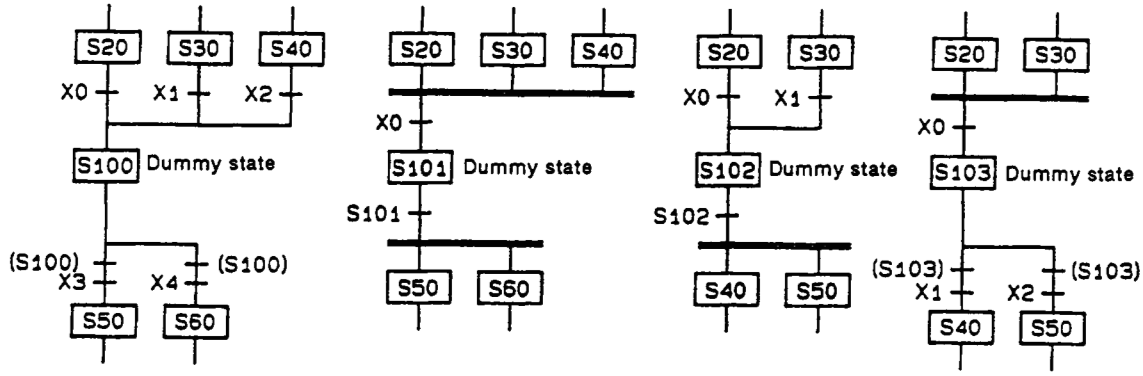
The transfer conditions marked by an asterisk (✱) are not permitted.

PROGRAMMING

3.2.5 COMBINATION OF BRANCHES AND MERGES



Rewrite in this way:



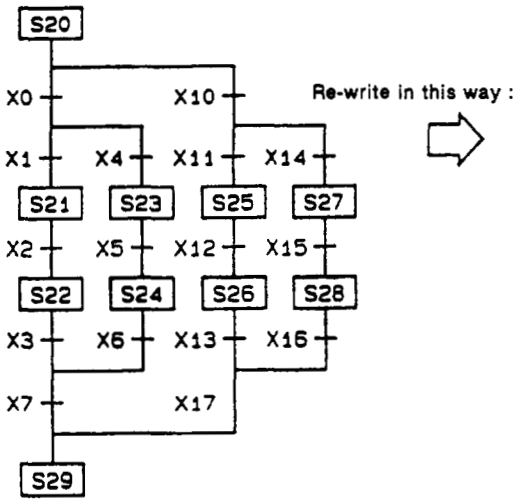
Program in this way :

```
STL   S 20        STL   S 20        STL   S 20        STL   S 20
LD    X  0        STL   S 30        LD    X  0        STL   S 30
SET   S100        STL   S 40        SET   S102        LD    X  0
STL   S 30        LD    X  0        STL   S 30        SET   S103
LD    X  1        SET   S101        LD    X  1        STL   S103
SET   S100        STL   S101        SET   S102        LD    S103
STL   S 40        LD    S101        STL   S102        AND   X  1
LD    X  2        SET   S 50        LD    S102        SET   S 40
SET   S100        SET   S 60        SET   S 40        LD    S103
STL   S100                          SET   S 50        AND   X  2
LD    S100                                            SET   S 50
AND   X  3
SET   S 50
LD    S100
AND   X  4
SET   S 60
```

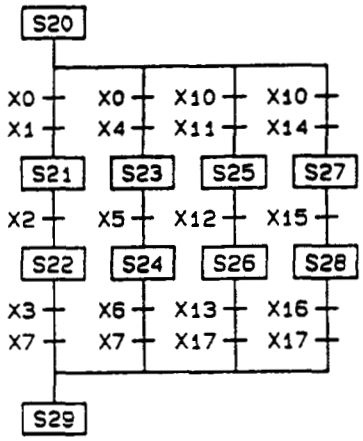
If processing continues directly from the branch line to the merge line without a state in between, write in a dummy state as shown above.

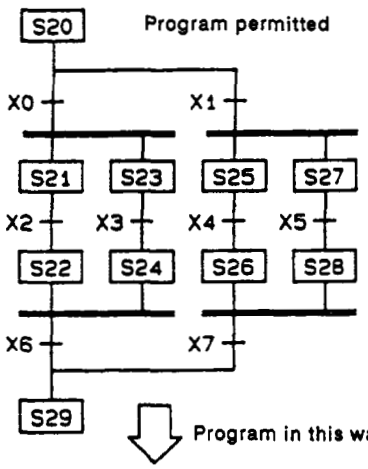
S100 and S103 transfer contacts can be omitted.

PROGRAMMING

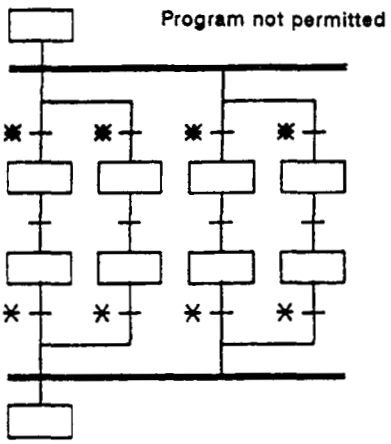


Re-write in this way :





Program in this way :



Branch transfer

```
STL   S20
LD    X 0
SET   S21
SET   S23
LD    X 1
SET   S25
SET   S27
```

Merge transfer

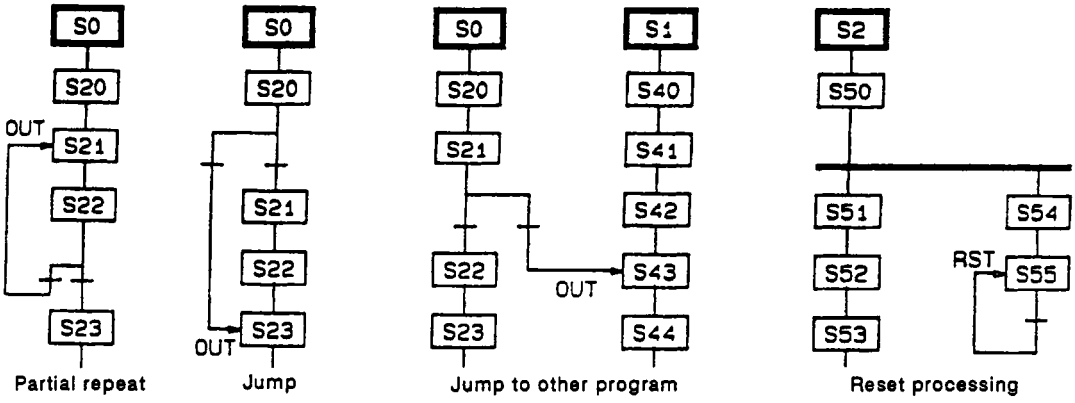
```
STL   S22
STL   S24
LD    X 6
SET   S29
STL   S26
STL   S28
LD    X 7
SET   S29
```

Writing a selective branch condition (✱) after parallel branching or a parallel merge after transfer condition (✱) are not permitted. See Reference in Section 3.2.4.

3. STEP LADDER INSTRUCTIONS

PROGRAMMING

3.2.6 JUMP AND REPEAT PROCESSES

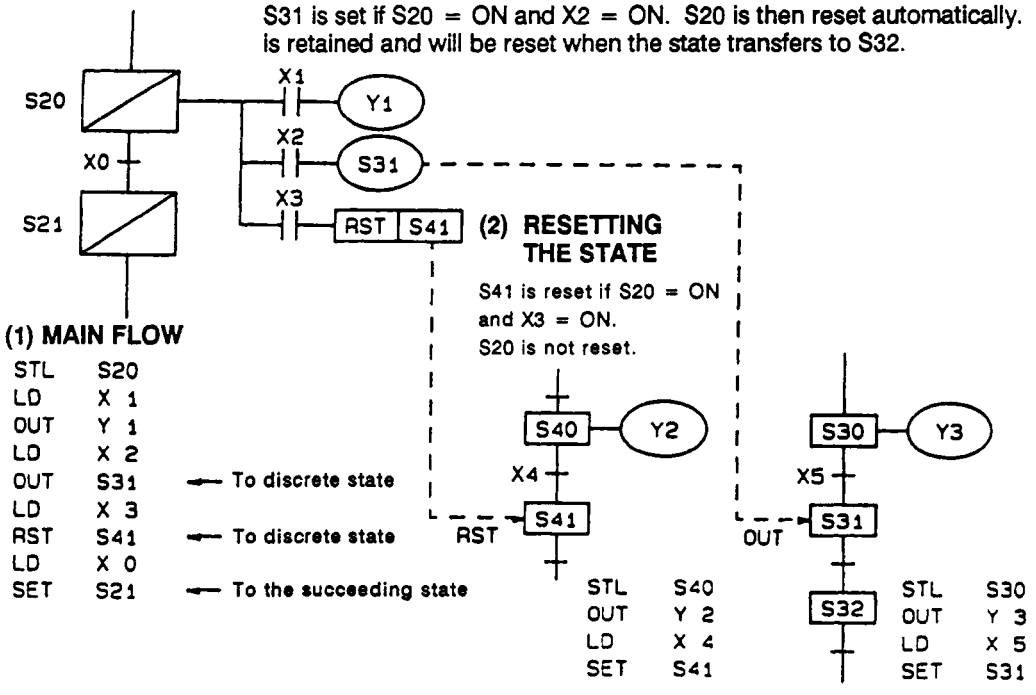


In addition to branching and merging (explained in previous sections), it is also possible to transfer to a distant state (a state outside of normal consecutive flow).

(3) TRANSFER TO A DISTANT STATE

For a transfer to a distant state, an OUT instruction is used instead of a SET instruction.

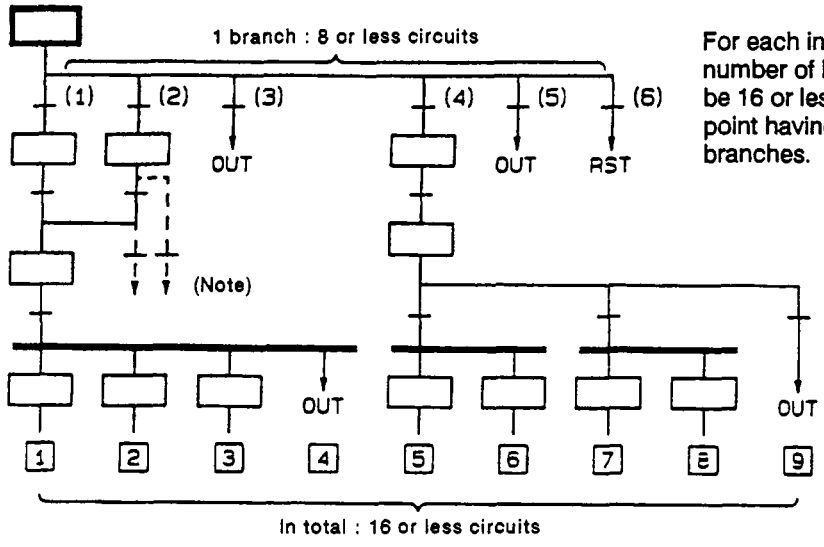
S31 is set if S20 = ON and X2 = ON. S20 is then reset automatically. S31 is retained and will be reset when the state transfers to S32.



3. STEP LADDER INSTRUCTIONS

PROGRAMMING

(1) LIMITS IN THE NUMBER OF BRANCHES

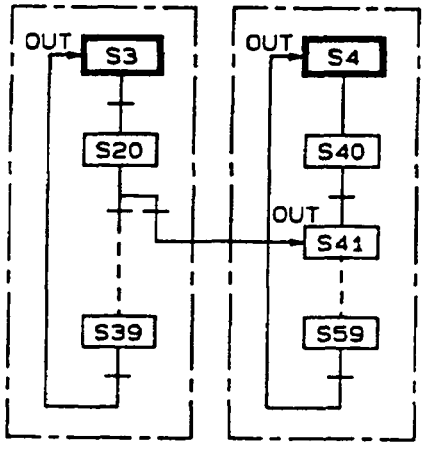


For each initial state, the total number of branch circuits should be 16 or less with each branching point having no more than 8 branches.

Note: • *A State transfer from a merge line or a state before a merge to a distant state or a reset process is not allowed. In such a case, always provide a dummy state to execute the state transfer.*

(2) SEPARATE FLOWS

If the sequential function chart contains more than one initial state, ensure the program instructions for one flow is grouped away from another flow.



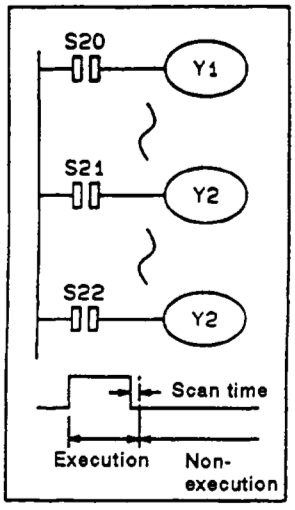
For the example shown on the left, first program the flow for the STL instructions for S20 to S39, which belong to initial state S3. After that, program the flow beginning with the initial state S4.

In these two separate flows, a state number of another flow can be used when the instruction used is other than STL. In this example, the "OUT S41" instruction is contained in the S3 sequence flow.

3. STEP LADDER INSTRUCTIONS

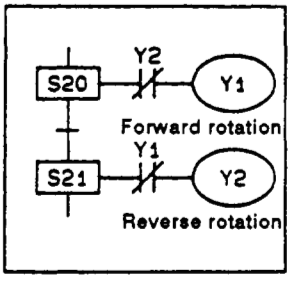
3.3 STATE FUNCTION RULES

3.3.1 STL INSTRUCTION



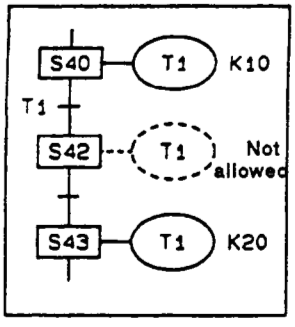
- As an STL contact is turned on the circuit connected to this contact operates.

  When an STL contact is turned off, the circuit connected to this contact stops operating. After one operation cycle to reset its load, the circuit instructions are then jumped and no longer executed.
- If the same output is designated in different states like in the example, output Y2 operates if S21 or S22 is ON. Y2 stops operating when both S21 and S22 are OFF.
- Although a double-coil may be designated with STL circuits, pay careful attention to double-coil designation for circuits other than STL.
- Designation of the same state is not permitted.



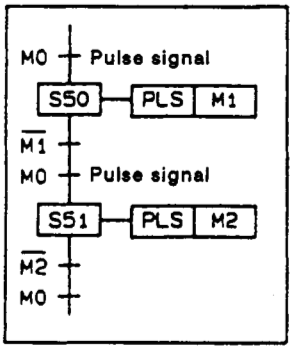
- During the state transfer process, both states will be ON momentarily (1 operation cycle).

  Therefore, if the state transfer is between opposing outputs that cannot be ON simultaneously as shown on the left, it is necessary to make an interlock so that these opposing outputs will not be ON simultaneously.



- If the same timer is used in a program repeatedly, it cannot be connected to the two succeeding states since the instructions will interact and the timer may not be able to reset.

  Two separated states can use the same timer.



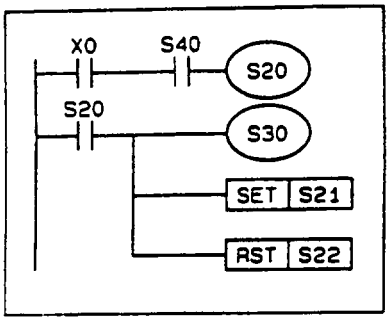
- To transfer the state sequentially with the same signal, the signal should be a pulse signal. Connect the states as shown to the left.

  Just after S50 is operated with M0 turned ON, M1 opens to prevent the state from immediately changing to S51. At the next M0 signal, the state changes transfers to S51.

3. STEP LADDER INSTRUCTIONS

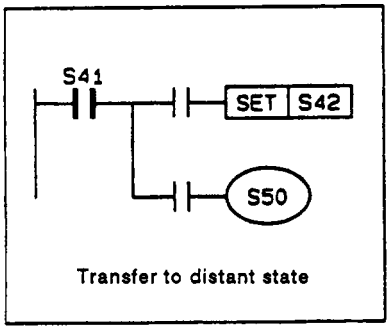
STATE FUNCTION RULES

3.3.2 INSTRUCTIONS



- An STL instruction is effective only when it is used with state S. The following contact and coil instructions can be used with state S :

  LD/LDI, AND/ANI, OR/ORI, OUT, SET/RST
- In the circuit example shown on the left, S20 is not reset even after S30 or S21 is driven by S20. If S20 is turned off, S30 stops operating. This is because S21 and S30 are driven directly without using an STL instruction.
- States can be used as general auxiliary relays when the program does not use STL instructions at all.



- OUT and SET instructions for state S have the same function if designated for a state which is designated after an STL instruction. Both of these instructions reset the transfer source automatically. They also have the retaining function against power downs.

  An OUT instruction should be used for the transfer to a distant state.
- An MPS instruction cannot be used just after an STL contact.

APPLICABLE SEQUENCE INSTRUCTIONS

The following table shows whether or not a specific instruction can be used in the steps between STL instructions or between STL and RET instructions.

State		LD/LDI, OUT, NOP, AND/ANI, SET/RST, OR/ORI, PLS/PLF	ANB/ORB, MPS/MRD/MPP	MC/MCR
Initial States, General States		Can be used	Can be used	Cannot be used
Branch/ merge state	Output Processes	Can be used	Can be used	Cannot be used
	Transfer Processes	Can be used	Cannot be used	Cannot be used

- An STL instruction cannot be used in an interrupt program or a subroutine program (explained in a later section).
- Although a jump instruction can be used within STL instruction steps, it will make the flow of processing complicated. It is, therefore, not recommended to use a jump instruction with STL instruction steps.

3. STEP LADDER INSTRUCTIONS

3.4 OPERATION MODES

3.4.1 WHAT IS AN OPERATION MODE?

Machine operation modes are largely classified into manual and automatic, which are then further classified into other modes. The illustration of the operation panel shown below takes these operation modes into consideration.

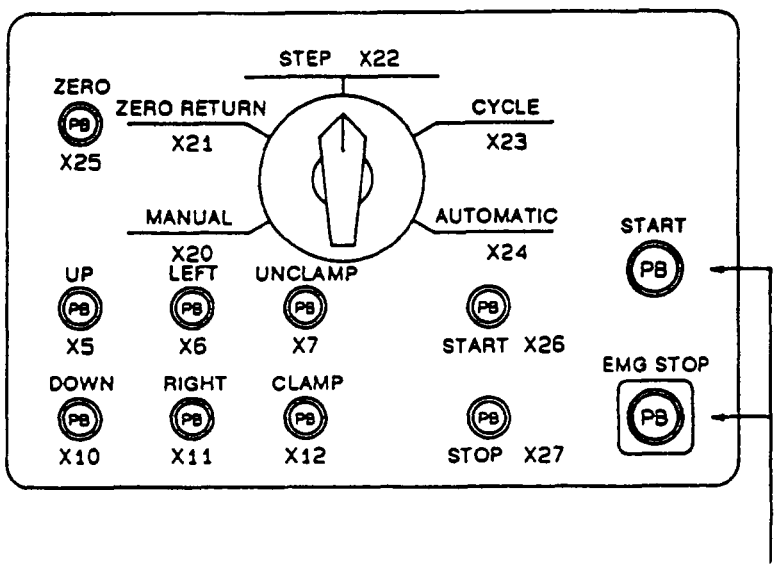
The START and EMERGENCY STOP pushbutton switches are connected to the sequence outside the PC to turn the power ON or OFF to the externally connected loads.

MANUAL
- **MANUAL** : Power supply to individual loads is turned ON/OFF by using the separately provided pushbutton switches.
- **ZERO RETURN** : In this mode, the actuators return to the initial position when the ZERO pushbutton switch is pressed.

AUTOMATIC
- **ONE STEP** : The machine operates step-by-step each time the START pushbutton switch is pressed.
- **ONE CYCLE** : When the START pushbutton switch is pressed in this mode, the actuators at the zero position initiate one automatic operation cycle. After the cycle is completed, the actuators stop at the zero position. It is possible to stop machine operation halfway by pressing the STOP pushbutton switch. The cycle will be continued when the START pushbutton switch is pressed.
- **AUTOMATIC** : Fully automatic operation is possible in this mode. The programmed cycle is executed repeatedly when the START pushbutton switch is pressed. The cycle will not stop immediately when the STOP pushbutton switch is pressed. Operation only stops after the currently executed cycle is completed.

It is not necessary to provide pushbutton switches for each individual actuator in the manual operation since individual force on/off's can be made with a programming panel.

OPERATION PANEL EXAMPLE



Pushbutton switches to turn the power ON/OFF to externally connected loads

OPERATION MODES

An applied instruction FNC60 (IST), explained in a later section (Section 5.8.1), automatically reserves an initial state to each corresponding mode.

To use this applied instruction, it is necessary to assign numbers in a consecutive order to the input signals to as shown below. If it is not possible to assign consecutive numbers, use auxiliary relays to rearrange the input numbers and use them as the head input when setting FNC60 (IST).

X20 : Manual
X21 : Zero return
X22 : Step
X23 : Cycle
X24 : Automatic
X25 : Zero return start
X26 : Automatic operation start
X27 : Stop

A selector switch is used to select X20 to X24 so that no more than one input is turned ON simultaneously.

RE-ARRANGEMENT EXAMPLES:

USING NON-CONSECUTIVE INPUT

Example :
X30 : Manual
X35 : Zero return
X33 : Step
X40 : Cycle
X32 : Automatic
X34 : Zero return start
X26 : Automatic operation start

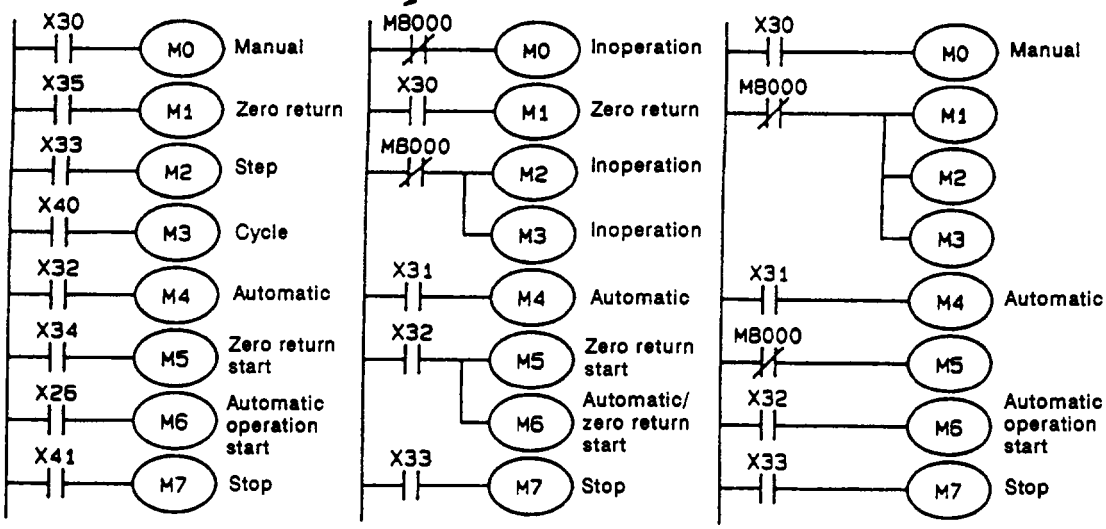
INPUTS FOR AUTOMATIC AND ZERO RETURN MODE ONLY

Example :
X30 : Zero return
X31 : Automatic
X32 : Automatic/ zero return start
X33 : Stop

INPUTS FOR AUTOMATIC AND MANUAL MODE ONLY

Example :
X30 : Manual
X31 : Automatic
X32 : Automatic operation start
X33 : Stop

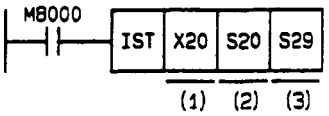
M8000 is a special auxiliary relay (RUN monitor) which is normally ON when the PC is running.



- In these examples, "M0" is used as the head input for the instruction FNC60 (IST).

OPERATION MODES

3.4.2 INITIAL STATE INSTRUCTION



If applied instruction FNC60 (IST instruction) is driven using the program shown on the left, the following initial states and the corresponding special auxiliary relays will be assigned to the following functions automatically.

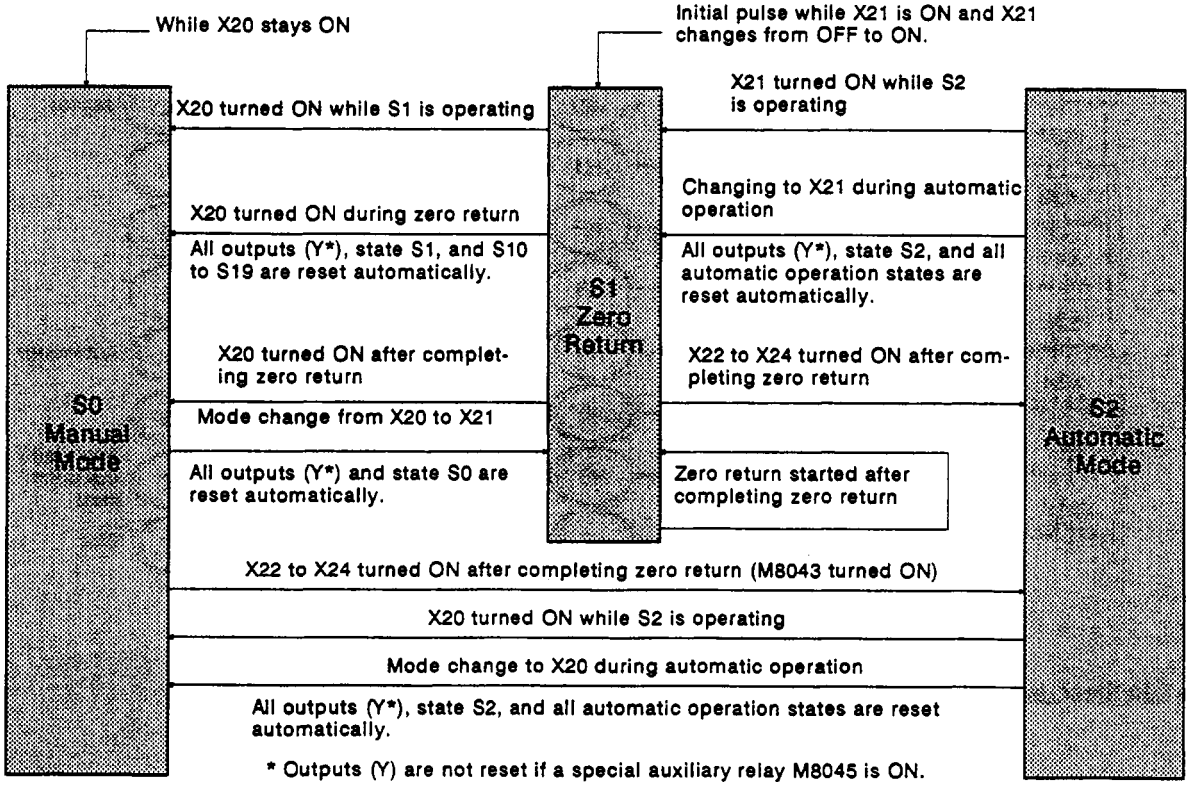
(1) Head number of input
(2) Start of range of states used for automatic mode
(3) End of range of states used for automatic mode

```
LD      M8000
FNC       60  ——  1 step
          X20
          S20  ┐
          S29  ┘—  2 step each
```

S0 : Initial state for manual mode
S1 : Initial state for zero return
S2 : Initial state for automatic mode
M8040 : Transfer disabled
M8041 : Transfer start
M8042 : Start pulse

INITIAL STATE OPERATIONS

Initial states S0 to S2, which operate automatically in response to an IST instruction, change as indicated below according to the operation mode selection.

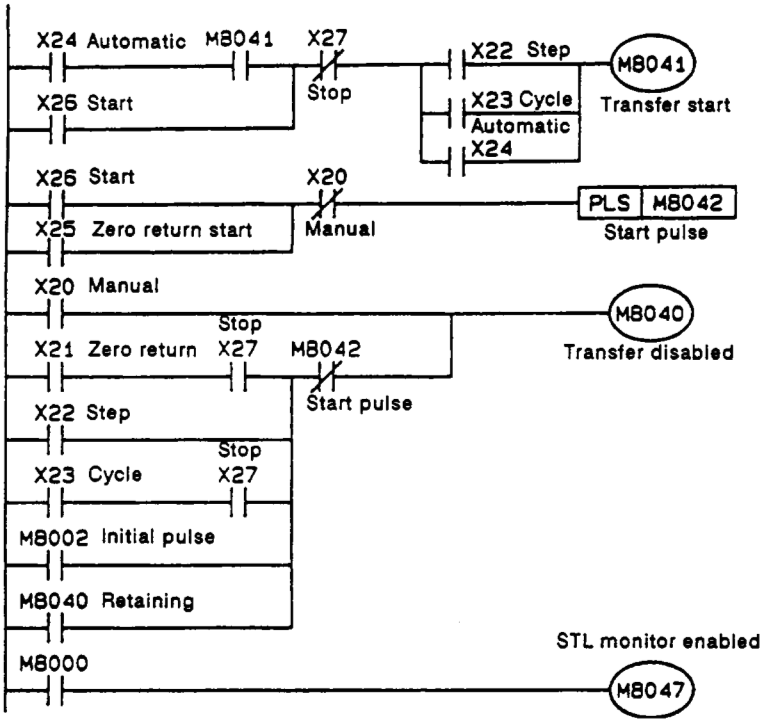


3. STEP LADDER INSTRUCTIONS

OPERATION MODES

The operation details of special auxiliary relays M8040 to M8042 and M8047 (all of which are automatically operated by the IST instruction) are shown in this equivalent circuit.

When M8047 is driven, up to eight state numbers of the states which are currently operating among S0 and S899 are stored to D8040 to D8047 from the lowest number.



TRANSFER INHIBIT RELAY (M8040)

When this auxiliary relay is actuated, state transfer is inhibited.

Manual : M8040 is always ON.

Zero return, one cycle :

Once the stop pushbutton switch is pressed, its state is retained until the start pushbutton is pressed.

One Step : M8040 is always ON. It is turned OFF only when the start pushbutton is pressed, enabling state transfer for one step of the flow.

Other : The ON state of M8040 is retained when the PC status is changed from STOP to RUN. It is turned OFF when the start pushbutton is pressed.

TRANSFER START RELAY (M8041)

This is an auxiliary relay used as a state transfer condition from the initial state S2 to another state.

Manual, zero return : This relay is not activated.

One step, one cycle : The relay is actuated only while the start pushbutton switch is held down.

Automatic : The relay ON state is retained when the start pushbutton switch is pressed. It is turned OFF when the stop pushbutton switch is pressed.

START PULSE RELAY (M8042)

This relay is actuated momentarily when the start pushbutton switch is pressed.

Ref.:

- *The special auxiliary relays M8044 (zero position) and M8043 (zero return complete) should be controlled by the user's circuits, such as initial circuit or zero return circuit, as shown in Section 3.4.3.*
- *The IST instruction executes the control in the range as described in the "Programming Tip" in Section 3.4.3.*

3. STEP LADDER INSTRUCTIONS

OPERATION MODES

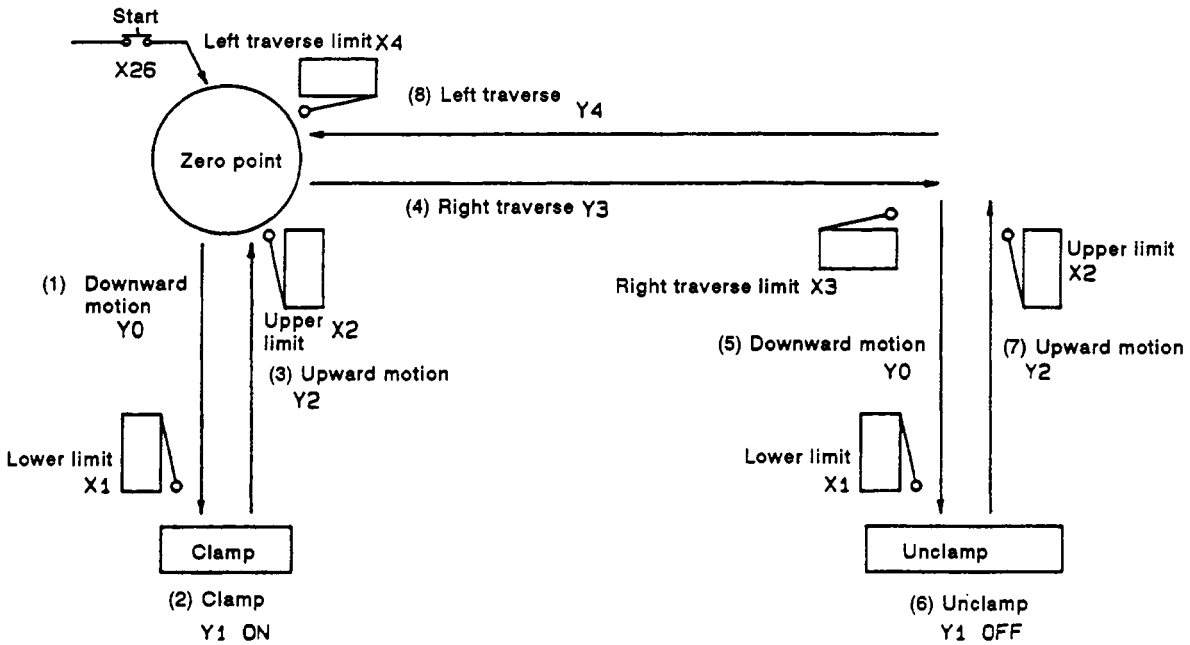
3.4.3 WORKPIECE TRANSFER PROGRAM EXAMPLE

The illustration below shows the mechanical system that is used for pick-and-place control with a robot hand.

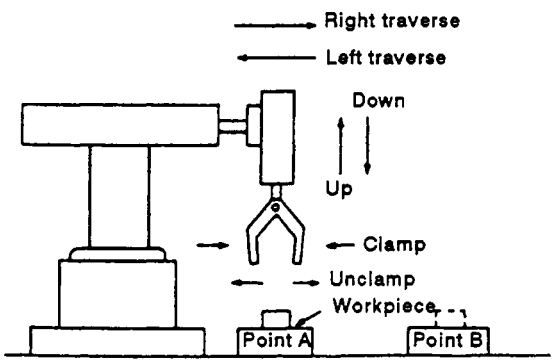
With the zero point taken at the upper left traverse end position, a workpiece is carried from the left table to the right table in the following operation sequence :

Downward motion - Clamp - Upward motion - Right traverse - Downward motion - Unclamp - Upward motion - Left traverse

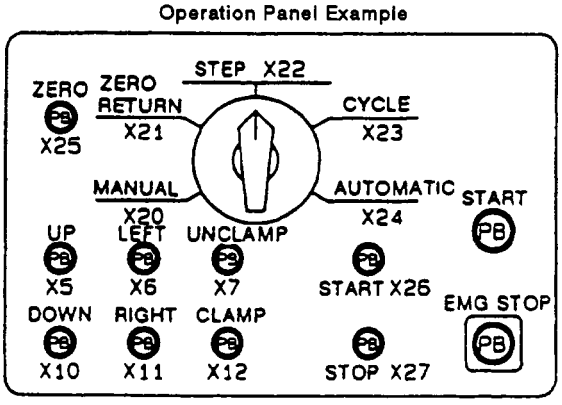
Double solenoid valves are used for up/down and right/left traverse control, and a single-solenoid valve is used for unclamp operation.



Workpiece Transfer Mechanism

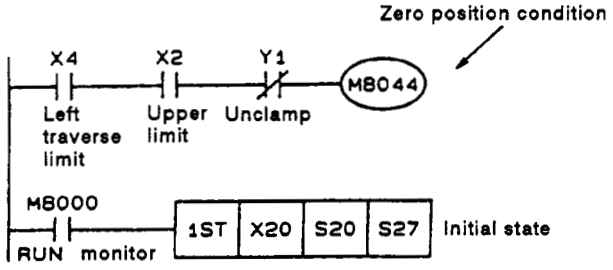


Operation Panel Example

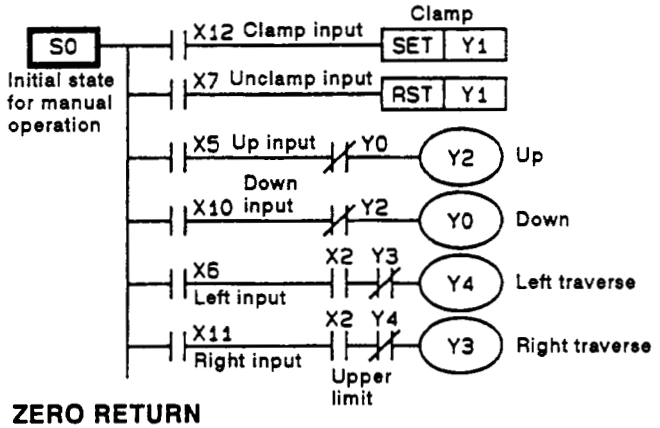


OPERATION MODES

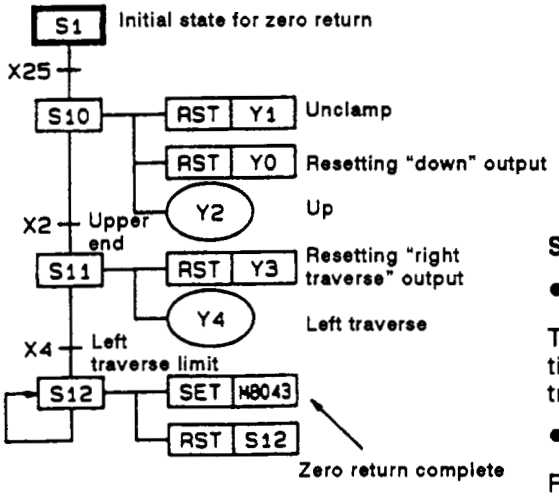
INITIAL CIRCUIT



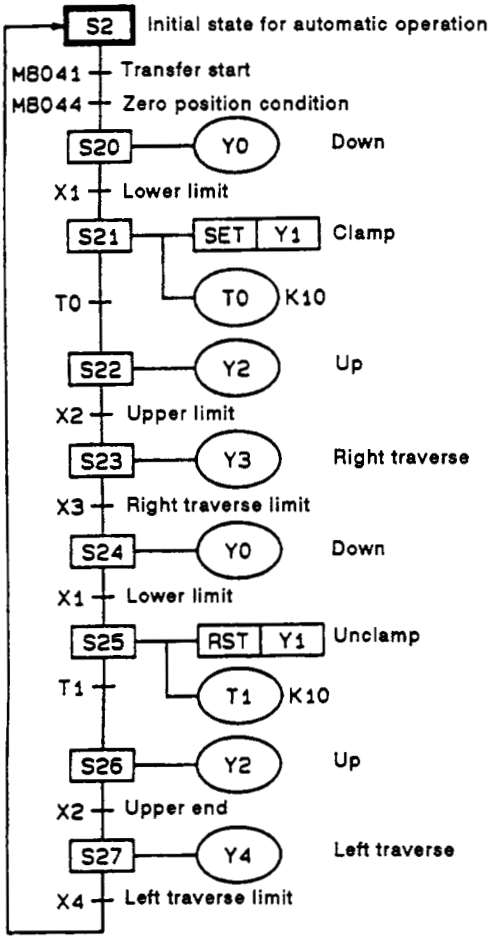
MANUAL OPERATION



ZERO RETURN



AUTOMATIC OPERATION



SPECIAL AUXILIARY RELAY

- M8044 (Zero-position condition)

This relay is driven by each sensor at the zero position. The ON state of this relay is used as the state transfer condition to automatic mode.

- M8043 (Zero-return complete)

For zero return operation, use states S10 to S19. In the final state, drive M8043 before resetting itself.

Note: • *The IST instruction must be programmed prior to the STL circuits.*

• *All outputs will be turned OFF if a rotary switch is operated to change the mode (X20, X21, X22, X23, or X24) before the zero return complete (M8043) relay is turned ON.*

3. STEP LADDER INSTRUCTIONS

OPERATION MODES

PROGRAM EXAMPLE :

The following is the program for the Sequential Function Charts on the previous page.

```
 0 LD    X   4
 1 AND   X   2
 2 ANI   Y   1
 3 OUT   M8044
 5 LD    M8000
 6 FNC     60
           X  20
           S  20
           S  27
13 STL   S   0
14 LD    X  12
15 SET   Y   1
16 LD    X   7
17 RST   Y   1
18 LD    X   5
19 ANI   Y   0
20 OUT   Y   2
21 LD    X  10
22 ANI   Y   2
23 OUT   Y   0
24 LD    X   6
25 AND   X   2
26 ANI   Y   3
27 OUT   Y   4
28 LD    X  11
29 AND   X   2
30 ANI   Y   4
31 OUT   Y   3
  (RET)
```

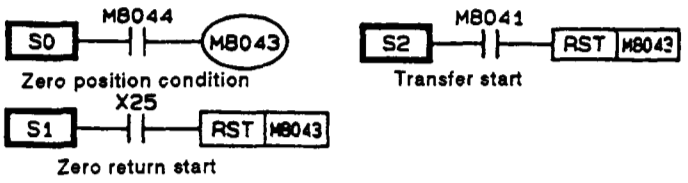
Instructions in ( ) are not necessary

```
32 STL   S   1
33 LD    X  25
34 SET   S  10
36 STL   S  10
37 RST   Y   1
38 RST   Y   0
39 OUT   Y   2
40 LD    X   2
41 SET   S  11
43 STL   S  11
44 RST   Y   3
45 OUT   Y   4
46 LD    X   4
47 SET   S  12
49 STL   S  12
50 SET   M8043
52 RST   S  12
  (RET)
54 STL   S   2
55 LD    M8041
56 AND   M8044
57 SET   S  20
59 STL   S  20
60 OUT   Y   0
61 LD    X   1
62 SET   S  21
```

```
64 STL   S  21
65 SET   Y   1
66 OUT   T   0
           K  10
69 LD    T   0
70 SET   S  22
72 STL   S  22
73 OUT   Y   2
74 LD    X   2
75 SET   S  23
77 STL   S  23
78 OUT   Y   3
79 LD    X   3
80 SET   S  24
82 STL   S  24
83 OUT   Y   0
84 LD    X   1
85 SET   S  25
87 STL   S  25
88 RST   Y   1
89 OUT   T   1
           K  10
92 LD    T   1
93 SET   S  26
95 STL   S  26
96 OUT   Y   2
97 LD    X   2
98 SET   S  27
100 STL  S  27
101 OUT  Y   4
102 LD   X   4
103 OUT  S   2
105 RET
106 END
```

PROGRAMMING TIP

While the applied instruction FNC60 is driven, special auxiliary relay M8043 (zero-return complete) is controlled automatically as shown below.

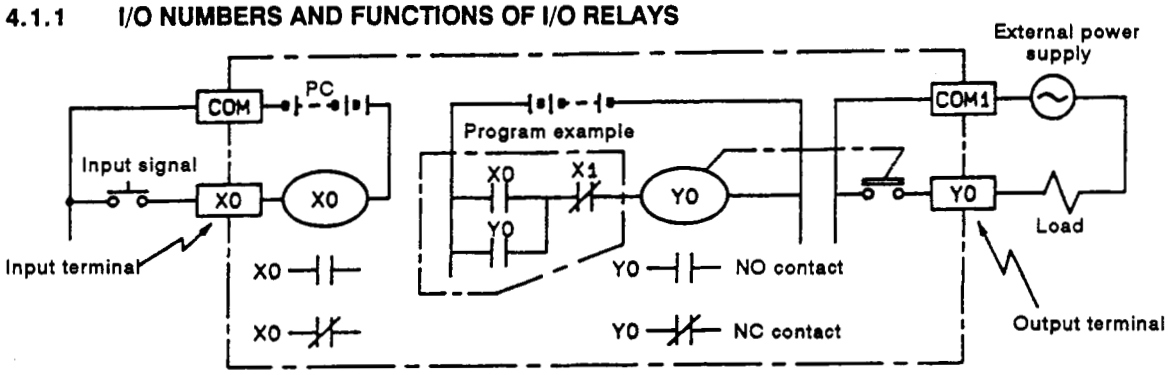


1	GENERAL HARDWARE AND INSTALLATION
2	BASIC SEQUENCE INSTRUCTIONS
3	STEP LADDER INSTRUCTIONS
4	DEVICE DETAILS
5	APPLIED INSTRUCTION
6	SPECIAL DEVICE AND INSTRUCTION LISTS
7	USING SPECIAL UNITS EFFECTIVELY
	APPENDIX

4. DEVICE DETAILS

4.1 INPUT/OUTPUT RELAYS

4.1.1 I/O NUMBERS AND FUNCTIONS OF I/O RELAYS



(1) INPUT RELAYS (X0 TO X177)

The maximum number of input relays is 128 points.

The external switch signals are received at the input terminals of the PC. Inside the PC, an optically isolated, electronic input relay (X) is connected to the input terminal. A number of NO (normally-open) and NC (normally closed) contacts are provided inside the PC and can be used as often as required.

The input relays cannot be driven within a program.

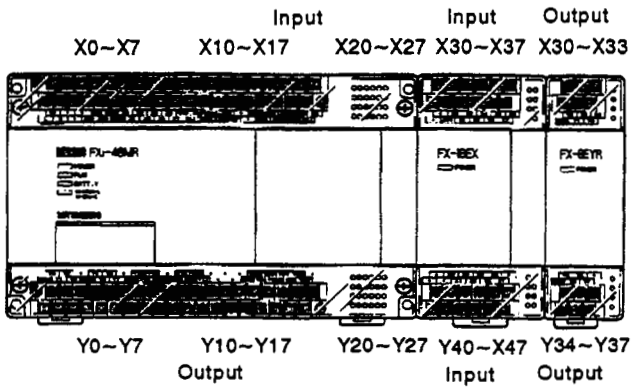
(2) OUTPUT RELAYS (Y0 TO Y177)

The maximum number of output relays is 128 points.

Signals processed inside the PC are output externally through the output terminals. External output contacts (an output device such as a relay contact, SSR, or transistor) of an output device are connected to the output terminal of the PC. An output relay has a number of NO and NC contacts that can be used as often as required in the program.

The operational differences between the external output contact (output device) and internal contact are described in Section 4.1.2.

(3) I/O NUMBER ASSIGNMENT EXAMPLE



- I/O numbers for extension units and blocks are assigned consecutively to the base unit beginning with the units/blocks located closest to the base unit. I/O numbers are assigned in octal numbers.

  Attach number labels to the terminal so that the I/O numbers can be distinguished easily.

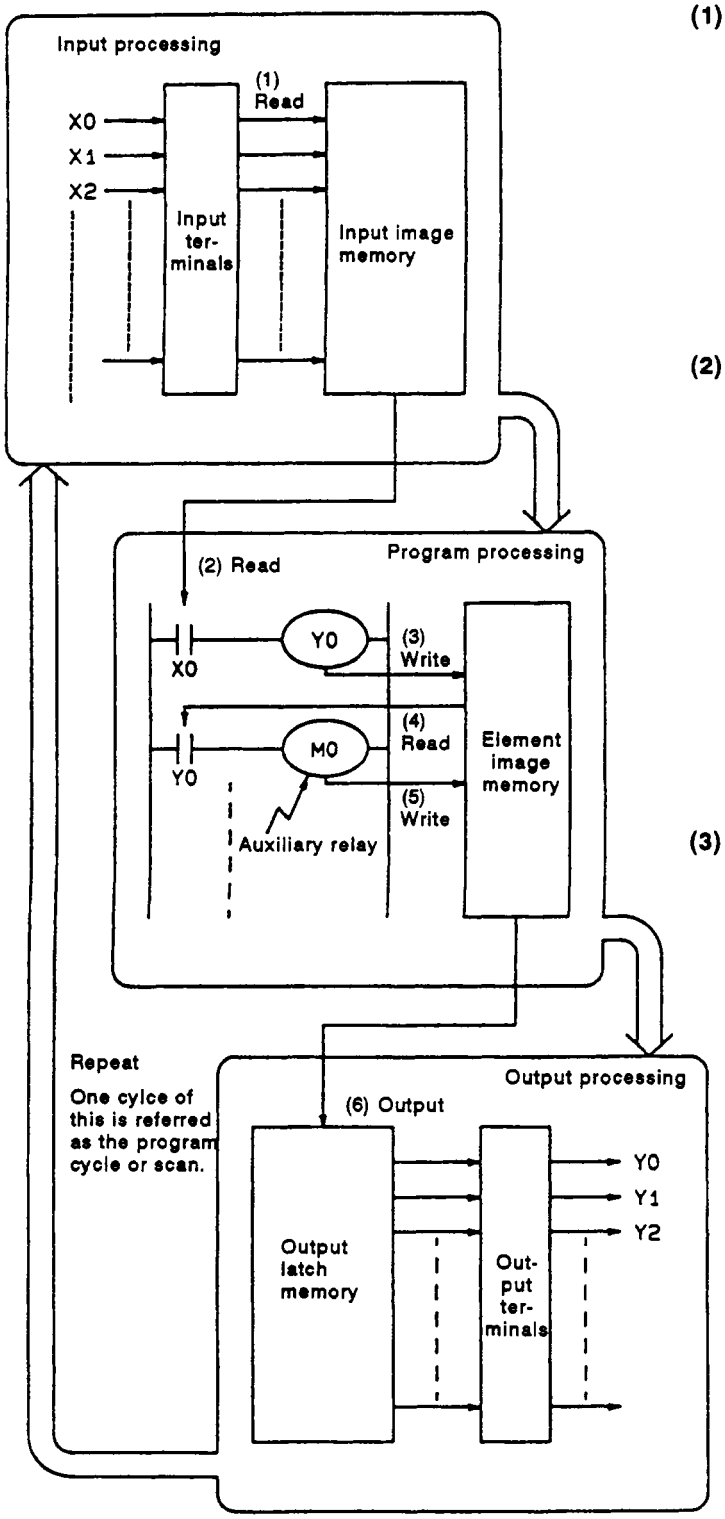
- Both inputs and outputs have numbers in the range of X/Y0 to X/Y177 (128 points). The total number of I/O points should be 128 or less.

  When an FX-8ER or FX2-24MR is used, 4/4 I/O points are left unused and, thus, the total I/O numbers are reduced accordingly.

- The base unit and an extension unit both have input terminals (upper terminals) and output terminals (lower terminals). If an extension block, dedicated for input or output, is used, both upper and lower terminals are used for inputs or outputs only. Upper terminals are assigned lower I/O numbers and lower terminals are assigned higher I/O numbers.

INPUT/OUTPUT RELAYS

4.1.2 I/O RELAY OPERATION TIMING



(1) INPUT PROCESSING

Before all of the program instructions are executed, the programmable controller reads the ON/OFF status of all input terminals into its input image memory.

When an input contact changes from OFF to ON and then from ON to OFF, a response delay (approx. 10 msec) exists due to the existence of the input filter.

(2) PROGRAM PROCESSING

The programmable controller reads the ON/OFF status of all elements from the input image memory and the image memories of other elements and writes the results of the operations to the image memory in accordance with the program instructions.

As a result, the image value of each element changes as the program is executed.

Operations of output relay internal contact are determined by the contents of output image memory.

(3) OUTPUT PROCESSING

When all instructions have been executed, the ON/OFF status of the image memory for the Y outputs is transferred to the output latch memory and this becomes the actual output of the programmable controller.

An external output contact in the PC operates after an output device response relay.

The type of processing described above is known as the batch I/O or refresh system.

4.2 AUXILIARY RELAYS

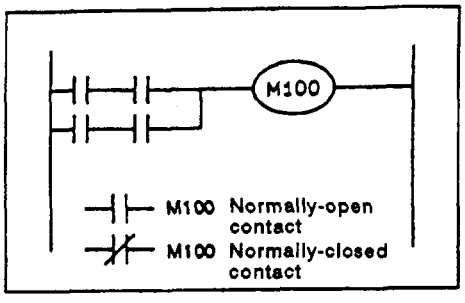
4.2.1 RELAY NUMBERS AND THE FUNCTIONS OF AUXILIARY RELAYS

A number of auxiliary relays are used in a PC. The coils of these relays are driven by a contact of devices in the PC in the same manner that the output relays are driven in the program.

Auxiliary relays have a number of electronic NO and NC contacts which can be used by the PC as required. Note that these contacts cannot drive an external load directly. Output relays must be used to drive external loads.

(1) GENERAL-USE RELAYS (M0 TO M499 : 500 POINTS)

Auxiliary Relay Circuit



500 relay points are provided for general-use. These relays are assigned a decimal relay number in the range of M0 to M499.

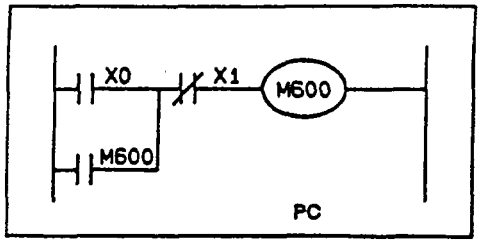
Note: • *All device numbers, with an exception of I/O relays (X/Y), are expressed with decimal numbers.*

(2) LATCH RELAYS (M500 TO M1023 : 524 POINTS)

If a power failure occurs while a PC is running, all output relays and general-use relays are turned OFF. All of the relays remain OFF except for those that are turned ON when the PC operation is restarted. However, before restarting the PC, the state of some devices must be set again to where they were just before the power failure.

Relays that retain their status during a power failure, often called latch or keep relays, are required in such cases. The status is retained by the backup battery incorporated in the PC.

Power Failure Data Retaining circuit

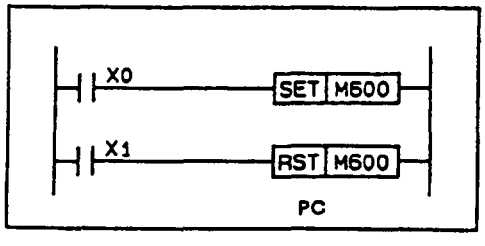


The circuit shown on the left is an example of a self-retaining circuit. Relay M600 is activated when X0 is turned ON. If X0 is turned OFF after the activation of M600, its activation status is self-retained.

Therefore, the activation status of M600 is retained if X0 is turned OFF due to a power failure. However, M600 is reset if the NC contact X1 opens.

A SET and RST (reset) instruction can be used to retain the status of a relay being activated momentarily.

Power Failure Data Retaining circuit
(Set/Reset)



Note : • *During the parallel link operation between two PC's, M800 to M999 are used. See Section 5.10.1.*

Auxiliary relays are provided with countless number of NO contact points and NC contact points, and are freely usable in the PC.

These contacts cannot drive external loads directly.

External loads should be driven through output relays.

AUXILIARY RELAYS

(3) SPECIAL-PURPOSE RELAYS (M8000 TO M8255 : 256 POINTS)

A PC has a number of special auxiliary relays. These relays all have specific functions and are classified into the following two types.

(a) USING CONTACTS OF SPECIAL AUXILIARY RELAYS

Coils are driven automatically by the PC and only the contacts of these coils can be used.

Examples: M8000 : RUN monitor (ON during run)
M8002 : Initial pulse (Turned ON momentarily when the PC starts running)
M8012 : 100 msec clock pulse

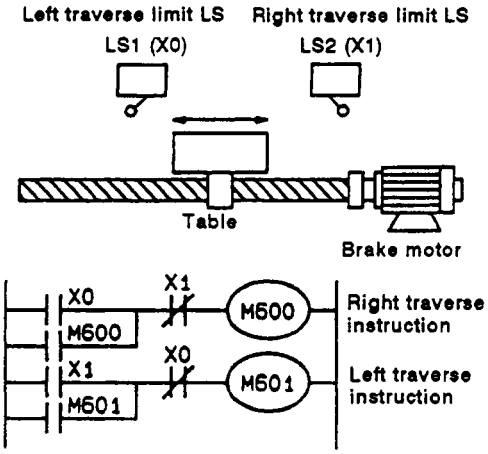
(b) DRIVING COILS OF SPECIAL AUXILIARY RELAYS

A PC executes a predetermined specific operation when these coils are driven by the user.

Examples: M8030 : BATT.V LED OFF
M8033 : All outputs retained when PC operation stops
M8034 : All outputs disable
M8039 : Constant scan

Note: • *Do not use undefined special auxiliary relays.*

APPLICATION EXAMPLE OF LATCH RELAYS



Use latch relays to feed the table in the same direction as fed before an occurrence of a power failure.

X0 = ON (left traverse limit) → M600 = ON → Table moves right → Power failure → Table stops → Restart (M600 = ON) → X1 = ON (right traverse limit) → M600 = OFF, M601 = ON → Table moves left

4. DEVICE DETAILS

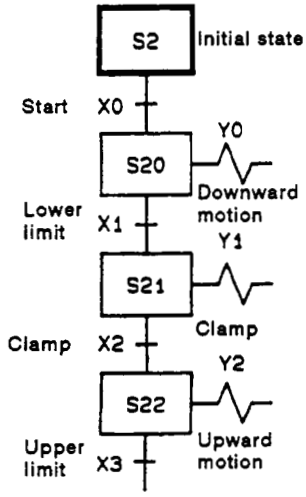
4.3 STATES

4.3.1 STATE NUMBERS AND THEIR FUNCTIONS

States (S) are very important devices when programming step-by-step process control. They are used in combination with step ladder instruction STL.

The four types of states are described below :

For Initial State	**(S0 to S9 10 points)**
For Zero Return	**(S10 to S19 10 points)**
For General Use	**(S20 to S499 480 points)**
For Retention	**(S500 to S899 400 points)**



In the step-by-step process control shown on the left, state S20 is set (ON) when start signal X0 is turned ON. At the same time solenoid valve Y0 is energized for downward motion.

When the lower limit LS X1 is turned ON, state S21 is set (ON) and clamp solenoid valve Y1 is energized.

When the clamp confirmation LS X2 is turned ON, state S22 is set (ON).

The transfer source state is automatically reset (OFF) as the state transfers sequentially.

Each state has a number of NO and NC contacts which can be used by the PC as often as required.

When a step ladder instruction is not used, states (S) can be used within a sequence in the same manner as auxiliary relays (M).

STATES

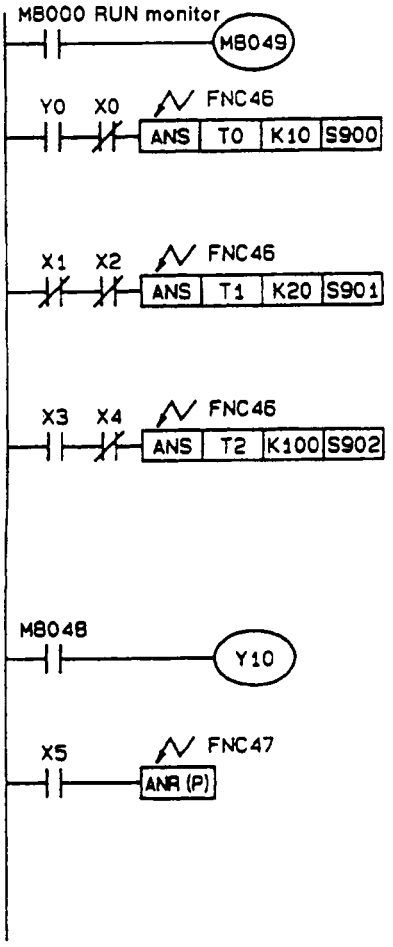
4.3.2 ANNUNCIATOR

Some state (S) areas can be used as outputs of external diagnosis.

States Usable for Annunciator (S900 to S999 : 100 points)

By programming an external diagnosis circuit as shown below, monitoring special data register D8049 will display the lowest activated state (S900 to S999). Each of these states should be set ON for each fault that may occur.

If more than one fault occurs simultaneously, the next lowest fault number will be displayed when the fault is reset.



- Monitoring is enabled by driving special auxiliary relay M8049.
- State S900 is activated if movement detection switch X0 is not driven within one second after the forward output Y0 has been turned ON.
- State S901 is activated when both upper limit LS X1 and lower limit LS X2 are OFF for more than two seconds.
- If the cycle time of a machine in its automatic mode is less than 10 sec., and X3 stays ON to select this mode, state S902 will be set ON if the cycle completion sensor X4 does not activated within this cycle time.
- If any state from S900 to S999 is activated (ON), special auxiliary relay M8048 is activated to turn on failure indicator output Y10.
- The state activated by the extra error/failure diagnosis program is turned OFF by pressing reset pushbutton switch X5.

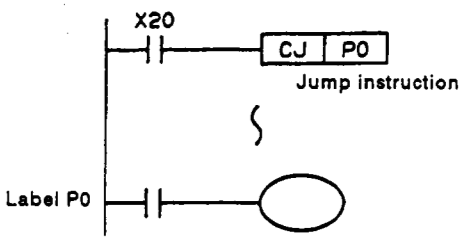
  Each time X5 is pressed, the activated states are reset in ascending order of the state numbers.

Note: • *For an ANS instruction and an ANR(P) instruction, see Section 5.6.7 and 5.6.8.*

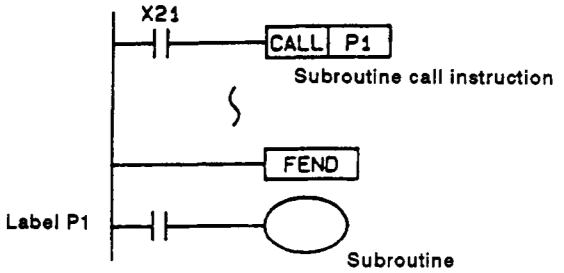
4.4 POINTERS (P, I)

4.4.1 POINTER NUMBERS AND THEIR FUNCTIONS

(1) POINTERS FOR BRANCH INSTRUCTIONS (P0 TO P63 : 64 POINTS)



When X20 is turned ON, the program sequence jumps the step of label P0.



When X21 is turned ON, the subroutine of label P1 that is programmed following an FEND instruction is executed. The program sequence returns from the subroutine when an SRET instruction is executed.

As described above, pointers P0 to P63 are used as a label to designate a jump destination for a branch instruction such as CJ and CALL. P63 is the step of the END instruction.

Note: • *Pointer numbers cannot be used repeatedly.*

(2) POINTERS FOR INTERRUPTS (I0 TO I2 : 3 POINTS)

(3)

I □0 □ Input interrupt
- 0 : Interrupt at trailing edge
- 1 : Interrupt at leading edge
- Input number (0 to 5)
  Each input can be used only once

Example : I001

The sequence programmed after the label (indicated by this pointer) is executed at the leading edge of input X0. The program sequence returns from the interruption program when an IRET instruction is executed.

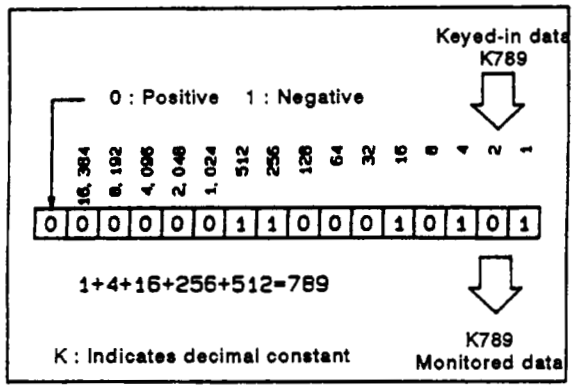
Note:
- *An interrupt pointer must be programmed in a step that follows an FEND instruction.*
- *No more than 3 interrupt pointer points can be used.*
- *No more than 2 nesting levels can be used.*
- *The number in the 100's place cannot be repeated.*

  I100 and I101 ..... Not allowed
- *The input used for interrupts must not coincide with inputs used by any high speed process.(e.g. high speed counter).*

4.5 CONSTANTS

4.5.1 BINARY DATA (K/H)

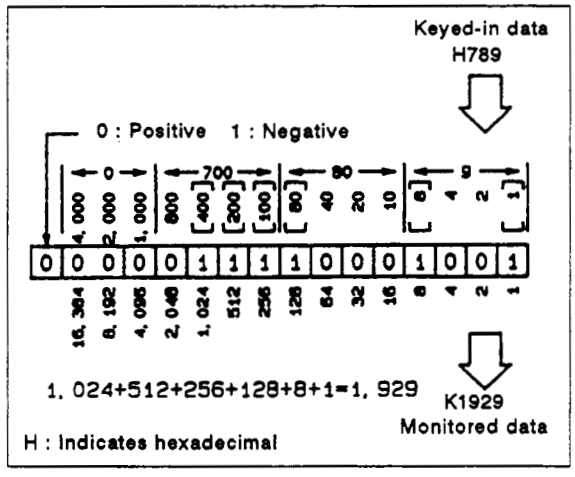
(1) BINARY AND DECIMAL



When decimal "K789" is keyed in as a timer or counter setting, it is automatically converted to binary data as shown on the left.

When a timer or counter current value stored in binary is monitored, it is automatically converted and displayed as a decimal.

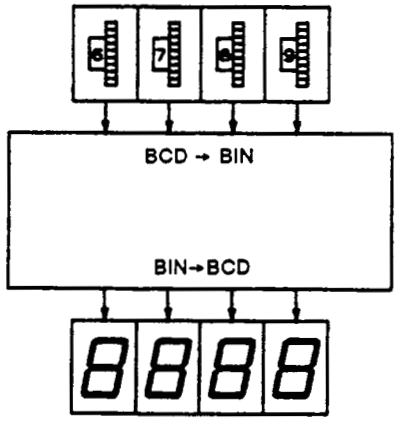
(2) BINARY AND HEXADECIMAL



When hexadecimal H789 is keyed in to a data register, it is converted into binary data as shown on the left. A hexadecimal digit is a number from 0 to 9 or a letter from A (representing 10) to F (representing 15).

When a data register is monitored, the decimal K1929 will be displayed first. The display changes to hexadecimal H789 when the HELP key is pressed. (when a hand-held panel is used).

BCD DATA



All operations executed by the PC, such as arithmetic calculations, increment instruction, and decrement instruction, use binary (BIN) data.

Therefore, use the BCD to BIN conversion instruction to read BCD digital switch data into the PC. Conversely, use the BIN to BCD conversion instruction to output the data to a BCD seven-segment display unit.

However, with special functions such as FNC72 (DSW), FNC74 (SEGL) and FNC75 (ARWS), this BCD/BIN conversion is automatically made within these commands.

4.6 TIMERS

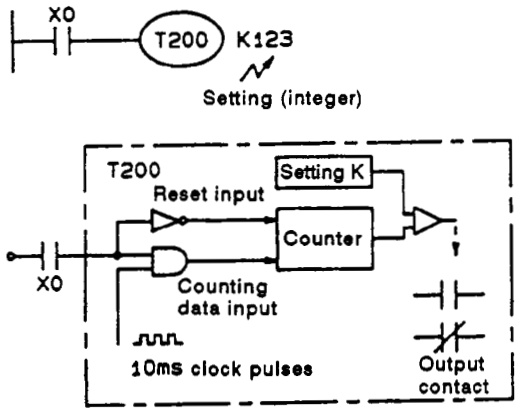
4.6.1 TIMER NUMBERS AND FUNCTIONS

Timers operate by counting clock pulses (1 msec, 10 msec, 100 msec). The timer output contact is activated when the count data reaches the preset value (setting).

Timers can either be set directly by using the constant K in the program memory or indirectly by using the data stored in a data register (D). For the indirect setting, data registers backed up by the battery are usually used to ensure that there is no loss of data. Therefore, timers or counters may malfunction if battery voltage is lowered.

(1) TIMERS (T0 TO T245)

100-msec timers (T0 to T199 : 200 points)	10-msec timers (T200 to T245 : 46 points)
Setting : 0.1 to 3,276.7 sec	Setting : 0.01 to 327.67 sec



When input X0, which drives timer coil T200, is turned ON, the current value for T200 begins counting in 10-msec pulses. The timer output contact is turned ON when the current value reaches setting K123. In other words, the output contact is turned ON 1.23 seconds after the coil is driven.

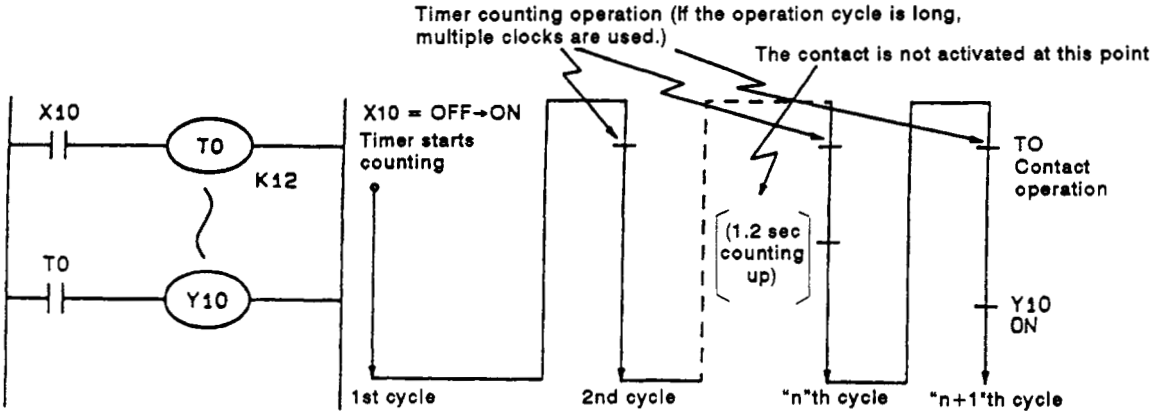
Both the counter and the output contact are reset when the drive input X0 is turned OFF or a power failure occurs.

Note: • *If T192 to T199 are used in a subroutine or an interruption routine, the timing is updated at the point when an END instruction is executed. The output contact is activated when a coil instruction or an END instruction is executed when the timer current value reaches the preset value.*
Other timers cannot count correctly in a sub-routine.

TIMERS

4.6.2 CONTACT OPERATION TIMING AND ACCURACY

Timers begin counting after the coil is driven. The output contact is activated when the first coil instruction is executed after the timer counts up.



The operation accuracy of a timer contact between the time the contact is activated and the time the coil is driven is expressed in the following manner.

$$T \begin{matrix} +T_0 \\ -\alpha \end{matrix}$$

α : 0.001, 0.01, 0.1 sec
(corresponds to 1 msec, 10 msec, and 100 msec timer)

T : Timer setting (sec)

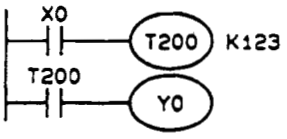
T_0 : Operation cycle (sec)

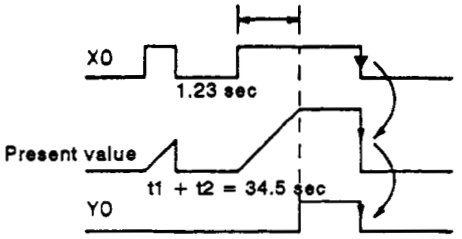
If a contact is written prior to a timer coil, the accuracy of the timer contact will become "+2T0" in the worst case.

If timer setting is "0", the output contact is activated when a coil instruction in the next cycle is executed.

A 1-msec timer counts 1-msec clock pulses in the interrupt mode after the execution of a coil instruction.

DETAILS OF TIMER OPERATION





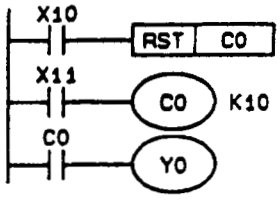
4.7 COUNTERS

4.7.1 INTERNAL SIGNAL COUNTERS

(1) 16-BIT UP-COUNTERS (SETTING : 1 TO 32,767)

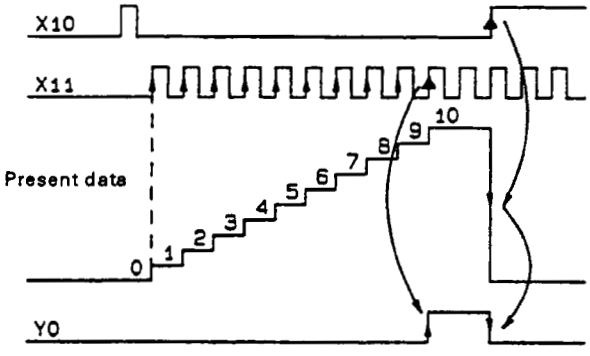
General-use Counters **(C0 to C99 : 100 points)**

Battery-backed Counters **(C100 to C135 : 36 points)**



Counters C100 to C199 retain the present data and contact set/reset state if a power failure occurs.

Two types of 16-bit binary up-counters are provided. The setting range of these counters ranges from K1 to K32,767. K0 and K1 have the same meaning and an output contact is activated after the first counting.



- The current value of the counter increases each time coil C0, is turned ON by the input X11. The output contact is activated when the coil is turned ON for the tenth time. After this, the counter data remains unchanged when X11 is turned ON.
- The counter present data is reset to "0" when a RST instruction is executed after reset input X10 is turned ON. The output contact is reset at the same time.
- Counters can be set directly using constant K or indirectly by using the data stored in a data register (D). In an indirect setting, the designation of D10, which contains for example the value of "123", has the same effect as the setting of K123.
- If a value greater than the counter setting is written to a current value register (with a MOV instruction, for example), the counter counts up when the next count input is turned ON. This is true for all types of counters.

INTERNAL SIGNAL COUNTERS

Internal signal counters are counters which count the signals of internal devices (such as X, Y, M, S, T, and C) while executing the cyclic operation. Therefore, both the ON duration and the OFF duration of the count input signals must be longer than the PC cycle time. Generally, the count input signal frequency should be several cycles/sec. (Hz).

COUNTERS

4.7.2 HIGH-SPEED COUNTERS

Although counters C235 to C255 (21 points) are all high speed counters, they all share the same 3 high speed counter inputs on the PC. Therefore, if an input is already being used by a high speed counter, it cannot be used for other high speed counters (or any other purpose). In other words, since there is only 3 high speed counter inputs, only a maximum of 3 high speed counters can be used simultaneously. High speed applied functions for executing comparisons and direct outputs are also applicable.

SELECTION

The selection of these high speed counters are not free and are directly dependent on the type of counter required and which of the high speed inputs are used.

Available counter types:

(1) 1-PHASE without START/RESET C235 to C237

(2) 1-PHASE with START/RESET C241,C244.

(3) 2-PHASE BI-DIRECTIONAL C246,C247,C249.

(4) 2-PHASE A-B PHASE TYPE C251,C252,C254.

All of these are 32-bit up/down counters. The following table shows which counter uses which inputs.

High-speed Counter Table

Count input	1-phase						1-phase with START/RESET					2-phase Bi-directional					2-phase, A-B phase type				
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255
X0	U/D			-	-	-	U/D	-	-	U/D	-	U	U	-	U	-		R	-	R	-
X1		U/D		-	-	-	R	-	-	R	-	D	D	-	D	-	B	B	-	B	-
X2			U/D	-	-	-		-	-		-		R	-	R	-	B	B	-	B	-
X3				-	-	-		-	-		-			-		-	A	A	-	A	-
X4				-	-	-		-	-		-			-		-			-		-
X5				-	-	-		-	-		-			-		-			-		-
X6				-	-	-		-	-	S	-			-	S	-			-	S	-
X7				-	-	-		-	-		-			-		-			-		-

U : Up-count input, D : Down-count input, A : A-phase input, B : B-phase input, R : Reset input, S : Start input

X6 and X7 are also high speed inputs but functions as a start signal only and cannot be used for high speed counting.

Different types of counters can be used at the same time but their inputs must not coincide.

Note: • *Inputs X0 to X7 cannot be used for more than one counter. For example, if C251 is used, the following counters and instructions cannot be used:*

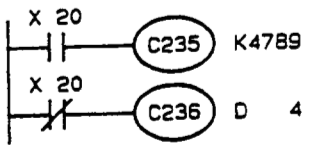
*C235, C236, C241, C244, C246, C247, C249, C252, C254, I0**, I1**, and any high speed instruction using the corresponding input.*

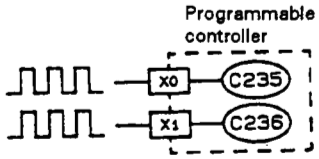
COUNTERS

DRIVING HIGH SPEED COUNTER COILS:

High speed counters operate by the principle of interrupts and are therefore event triggered and independent of cycle time. The coil of the selected counter should be driven continuously to indicate that this counter and its associated inputs are reserved and that other high speed processes must not coincide with this.

Example:

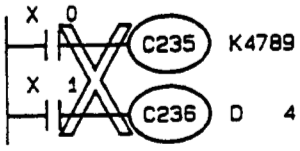




When X20 is ON, high speed counter C235 is selected. From the above table C235 corresponds to count input X0 and will therefore count pulses from X0, NOT X20.

When X20 is OFF, coil C235 is turned OFF and coil C236 is turned ON. Therefore, counter C236 is selected and counts from input X1.

Caution: • *Do not use the count inputs as contacts to connect high speed counters.*



COUNTERS

(1) 1-PHASE TYPES (C235 to C244)(max.3pts.)

With 1-phase high speed counters there are the following two groups:

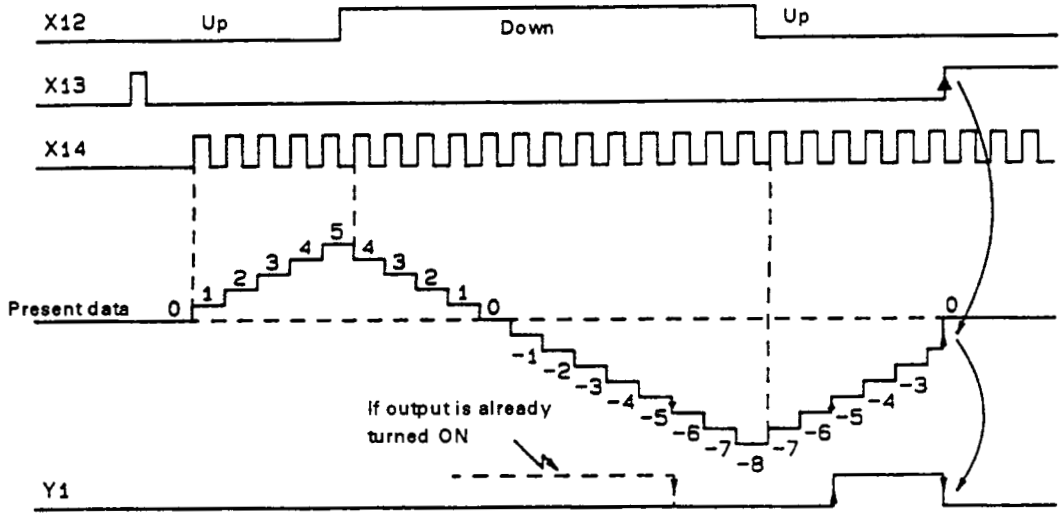
(a)	C235 to C237	Without START/RESET	Setting range:
(b)	C241,C244	With START/RESET	-2,147,483,648 to +2,147,483,641

All of these are 32-bit up/down ring counters. When the count value reaches its setting the contact sets and holds when up-counting but resets when down-counting.

DIRECTION:

The direction of count for 1-phase counters is dependent on their corresponding flag M8ΔΔΔ where ΔΔΔ is the corresponding counter number (C235 to C244).

(a) 1-PHASE WITHOUT START/RESET (C235 to C237)

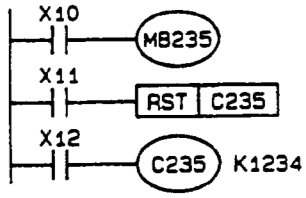


The counter operates when the C200 coil is activated as it counts by counting input X14.

- The output contact is set when the counter current value increases from "-6" to "-5" and is reset when counter current value decreases from "-5" to "-6".
- The counter current value increases or decreases independent of output contact state (set/reset). However, if a counter counts beyond +2,147,483,647", the counter current value changes to "-2,147,483,648". Similarly, counting below "-2,147,483,648" changes the counter data to "+2,147,483,647". Such counters are called "ring counters".
- If reset input X13 is turned ON, the counter present data is reset to "0" and the output contact is reset also.
- Counters that are backed up by the battery retain its (current value and output contact) set/reset state if a power failure occurs.
- A 32-bit counter can be used as a 32-bit data register. A 32-bit counter cannot be designated as a device for a 16-bit instruction.

COUNTER

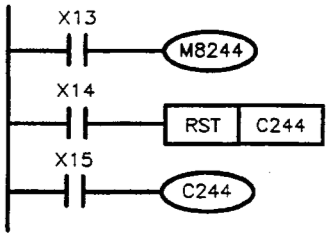
These counters only use one input each.



- When direction flag M8235 is ON, counter C235 counts down. When it is OFF, C235 counts up.
- When X11 is ON, C235 resets to zero. Contact C235 turns off.
- When X12 is ON, C235 is selected. From the table on the last page, the corresponding count input to C235 is X0. C235 therefore counts the "OFF → ON" signals of input X0.

(b) 1-PHASE WITH START/RESET (C241,C244)

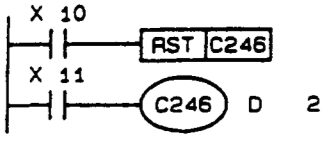
These counters have one count input and one reset input each.

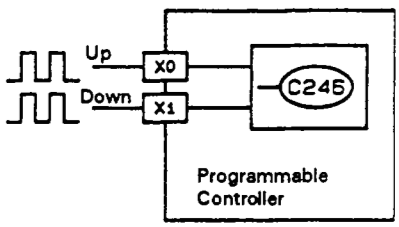


- When direction flag M8244 is ON, C244 counts up. When it is OFF, C244 counts down.
- When X14 is on, C244 resets in the same manner as normal internal 32-bit counters. From the table on the last page, it can be seen that C244 can also be reset externally by the input X1.
- Counter C244 also has an external start input X6.
- When this is ON, C244 starts counting. When it is OFF, C244 stops counting.
- X15 selects and reserves C244 for counting "OFF ON" events of its corresponding input X0.
- Note that the setting of D0 for counter C244 is a paired designation (D0, D1) since the counters are of 32-bits.
- Note the external control of start (X6) and reset (X1) have the advantage of being immediate and not affected by the program cycle time.

(c) 2-PHASE BI-DIRECTIONAL COUNTERS (C246,C247,C249)(max.1pt.)

These counters have one input for up-count and one input for down count. Some of these have reset and start inputs as well.



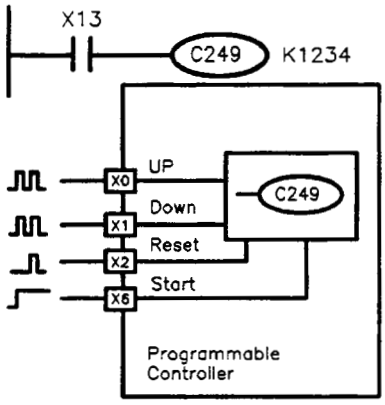


- When X10 is ON, C246 resets in the same way as internal 32-bit up/down counters.
- From the high speed counter table on the previous page, counter C246 uses X0 as up count inputs and X1 as down-count input. X11 must be ON select counter C246 and reserve these inputs.

  X0 "OFF → ON" : C246 increment one

  X1 "OFF → ON" : C246 decrement one

COUNTER



- From the high speed counter table on the previous page, bi-directional counter C249 can be seen to have X2 as its reset input and X6 as its start input. Therefore, reset can be made externally without the need for the RST C249 instruction.
- X13 must be ON to select C249. Start input X6 must ON to allow plus count. Counting stops if X6 is OFF.
- Up-count input :X0
  down-count input : X1
- The direction of the counting can be monitored by monitoring the state of the corresponding flag M8ΔΔΔ where ΔΔΔ is the counter number.
  ON: down count, OFF: up count

(2) 2-PHASE, A-B PHASE COUNTERS (SETTING : -2,147,483,648 TO +2,147,483,647)

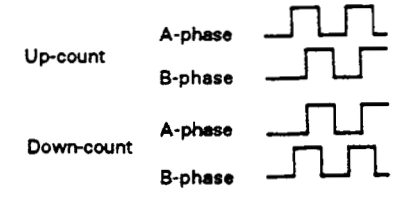
2-PHASE, 2-INPUT (C251 C252,C254 : 1 POINT) (Battery backed)

- A maximum of 1 point of 2-phase 32-bit binary up/down counter is provided. The operation of the output contact in relation to the counted data is the same as the 32-bit counters described in Section 4.7.1 (2).
- With these counters, however, only the inputs indicated in the high speed counter table can be used for counting. Its interrupt counting is executed independent of sequence operation. Applied instructions for executing comparisons and outputs independent of sequence operation are also provided.
- Depending on the counter number used, start, reset and such input signals can also be used.
- The A-phase and B-phase input signals determine whether the counter operates as an up-counter or a down-counter.

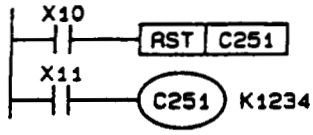
While the waveform of the A-phase is in the ON state:

B-phase input OFF → ON : Up-counter

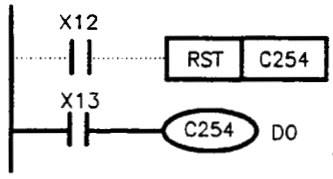
B-phase input ON → OFF : Down-counter



Check the corresponding special relay M8ΔΔΔ to determine whether the counter is counting up or down.



- C251 counts the "ON/OFF" events of input X0 (A-phase) and input X1 (B-phase) while X11 is ON.



- C254 starts counting immediately when X6 is turned ON while X13 is ON. The counting inputs are X3 (A-phase) and X2 (B-phase).
- C254 is reset when X0 is turned ON. It can also be reset with X12 in the sequence.

4. DEVICE DETAILS

COUNTERS

MAXIMUM COUNTING SPEEDS

The speed of the high speed counters are limited by 2 factors:

(a) response of the individual inputs
(b) processing time of all the high speed counters

(a) RESPONSE OF THE INDIVIDUAL INPUTS

This is limited by hardware and the following table gives the maximum countable frequency when only one count input is used.

Inputs	Max. Frequency
X0, X2, X3	5 kHz
X1, X4, X5	5 kHz

(b) PROCESSING THE TIME OF ALL THE HIGH SPEED COUNTERS

This is the main area that limits the speed of the counters. The counters operate with the principal of interrupts so that the lower the number of high speed processes, the higher the countable frequency. Also if the frequency used in one counter is much lower than the rated frequency specified below, this will allow other counters to count a little faster.

The overall limitation is that the counting frequency of all the counters combined together must be less than 5 kHz.

COMBINED FREQUENCY

The combined frequency is the sum of the maximum frequencies of all the signals appearing at the input of the PC simultaneously. The criteria (b) of the above is that in order for the high speed counters to count correctly. This combined frequency must be less than 5 kHz.

Example:

1-phase Counters	Corresponding Input	Maximum Signal Frequency
C235	X0	0.2 kHz
C236	X1	4 kHz
C237	X2	0.1 kHz
		Combined frequency: 4.3 kHz

The combined frequency of 4.3 kHz is lower than the max. of 5 kHz so this example is valid.

2-phase Counters:

(a) Bi-directional types: – are designed such that the up-count signal and the down-count signal never operate at the same time. Therefore it is really using only one phase at one time. Thus, they can be treated in the same way as the 1-phase counters when calculating the combined frequency.

When pulses arrive at the up and down count inputs at the same time, treat this as 2 single phase counters when calculating the combined frequency.

When encoders that have clockwise and counter-clockwise format outputs are used, these bi-directional counters can count at a much higher frequency than the A-B phase type counters without any loss in resolution.

COUNTERS

(b) A-B phase types: – are different to other counters in that they can decode simultaneously the A-phase and B-phase signals to automatically select up count and down count.

These are recommended to be driven at no higher than 1.5 kHz.

When calculating for the combined frequency, the maximum signal frequency for each of these counters should be multiplied by a factor of 4 before adding with the rest of the other counter signal frequencies.

Example:

Counters	Corresponding Input	Maximum Signal Frequency
1-phase C235 A-B phase C251	X0 X3, X2	1 kHz 1 kHz x 4
		Combined frequency: 1 + (1 x 4) = 5 kHz

The combined frequency is not above the limit of 5 kHz, therefore the combination is valid.

SIMPLIFIED GUIDE FOR COUNT SPEEDS

Counters	Maximum Frequency (when using 1 point)
1-phase	5 kHz
Bi-directional	5 kHz
A-B phase	1.5 kHz

When many counters or combination of the 3 counter types are used, their total combined frequency must not be over 5 kHz (remember to multiply the A-B phase counter frequency by 4).

Example:

1-phase counters : 3 kHz (1 point)

Bi-directional counter : 1 kHz (1point)

Combined frequency = 3 kHz + 1 kHz

= 4 kHz

4.8 DATA REGISTERS

4.8.1 DATA REGISTER NUMBERS AND FUNCTIONS

Data registers are used to store numeric data.

Although all data registers are 16-bit devices (the MSB stores a plus or minus sign), 32-bit data can be stored by using a pair of 16-bit registers.

(1) GENERAL-USE REGISTERS (D0 TO D99 : 100 POINTS)

Once data is written to a general-use register, it remains unchanged until it is overwritten.

When the PC status is changed from RUN to STOP, all these registers are cleared to "0".

Note: ✎ Data can be retained when the PC status is changed from RUN to STOP by setting special auxiliary relay M8033 ON.

(2) BATTERY BACKED REGISTERS (D100 TO D127 : 28 POINTS)

Once data is written to a battery backed register, it remains unchanged until it is overwritten. When the PC status is changed from RUN to STOP, the data in these registers is retained.

(3) SPECIAL REGISTERS (D8000 TO D8069 : 70 POINTS)

Special registers are used to control or monitor various modes of devices inside the PC.

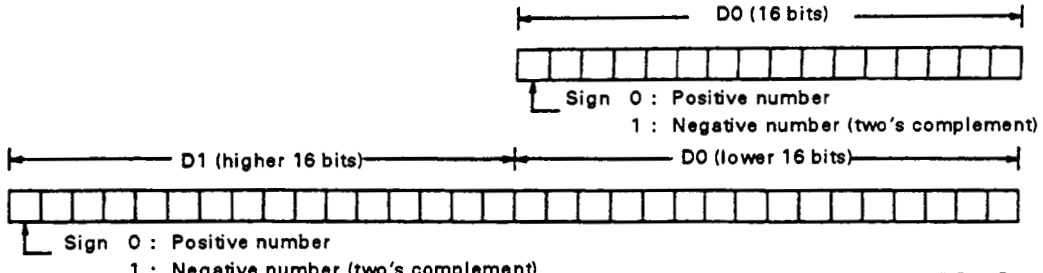
Data in these registers are set to the default values when the power supply to the PC is turned ON. (When the power is turned ON, all registers are first cleared to "0" and the default values are then written to some of the registers by the system software.)

For example, the watchdog timer data is set to D8000 by the system software. To change the setting, write the required value to D8000 using a transfer instruction.

Data set in these areas remain unchanged when the PC status changes from STOP to RUN.

Note: ✎ Do not use undefined special data registers.

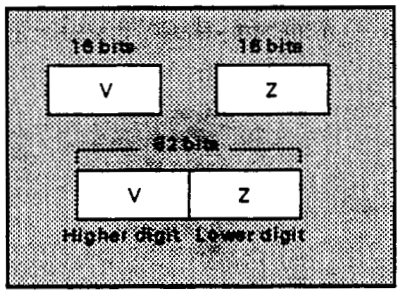
16-bit/32-bit data :



✎See Page 129.

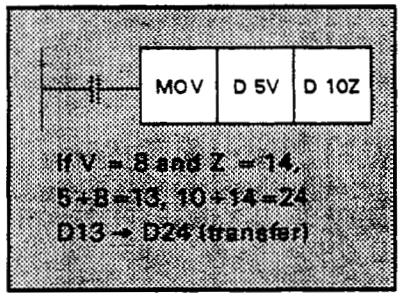
4.9 INDEX REGISTERS

4.9.1 MODIFYING DEVICE NUMBERS



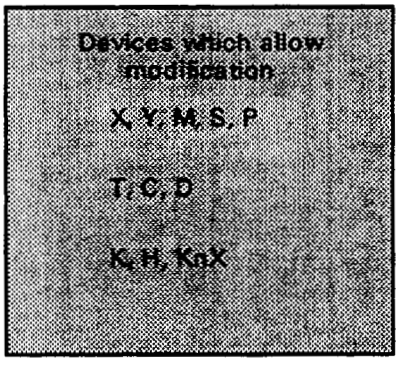
Both index registers V and Z are 16-bit registers. Numerical data can be read/written to index registers in the same manner as for general-use registers.

When executing 32-bit data operation, combine these registers (V and Z) and designate the register V.



As shown to the left, the processing required to change a device number according to the data for Z and V is called index addressing.

If a constant is used, with K20V and V = 8, the result istherefore K28 (20 + 8 = 28).



The devices that can be modified with an index register are shown on the left.

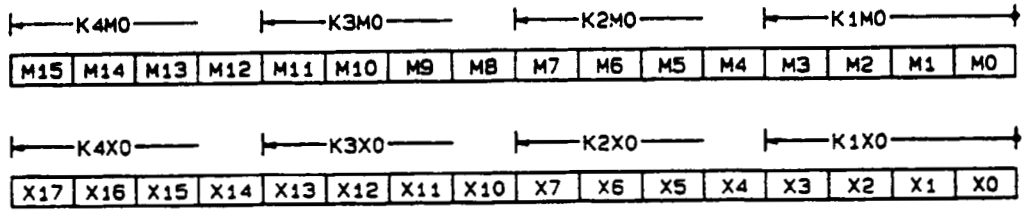
Note: • *Modifying Kn, which is used to designate the number of digits, is not possible.*

K4M0Z ..... Allowable

K0ZM0 ..... Not allowable

PROCESSING NUMERIC DATA WITH BIT DEVICES

In Kn, "n" represents the number of digits when 4 bits are taken as 1 digit. For example, K8M0 indicates 32-bit data comprised of bits M31 to M0 (bit M31 is a sign bit). The notation of K1M0 indicates the 16- or 32-bit data with leading zeros. In this case, bits M4 to M31 can be used for other purposese. See Section 5.1.3.

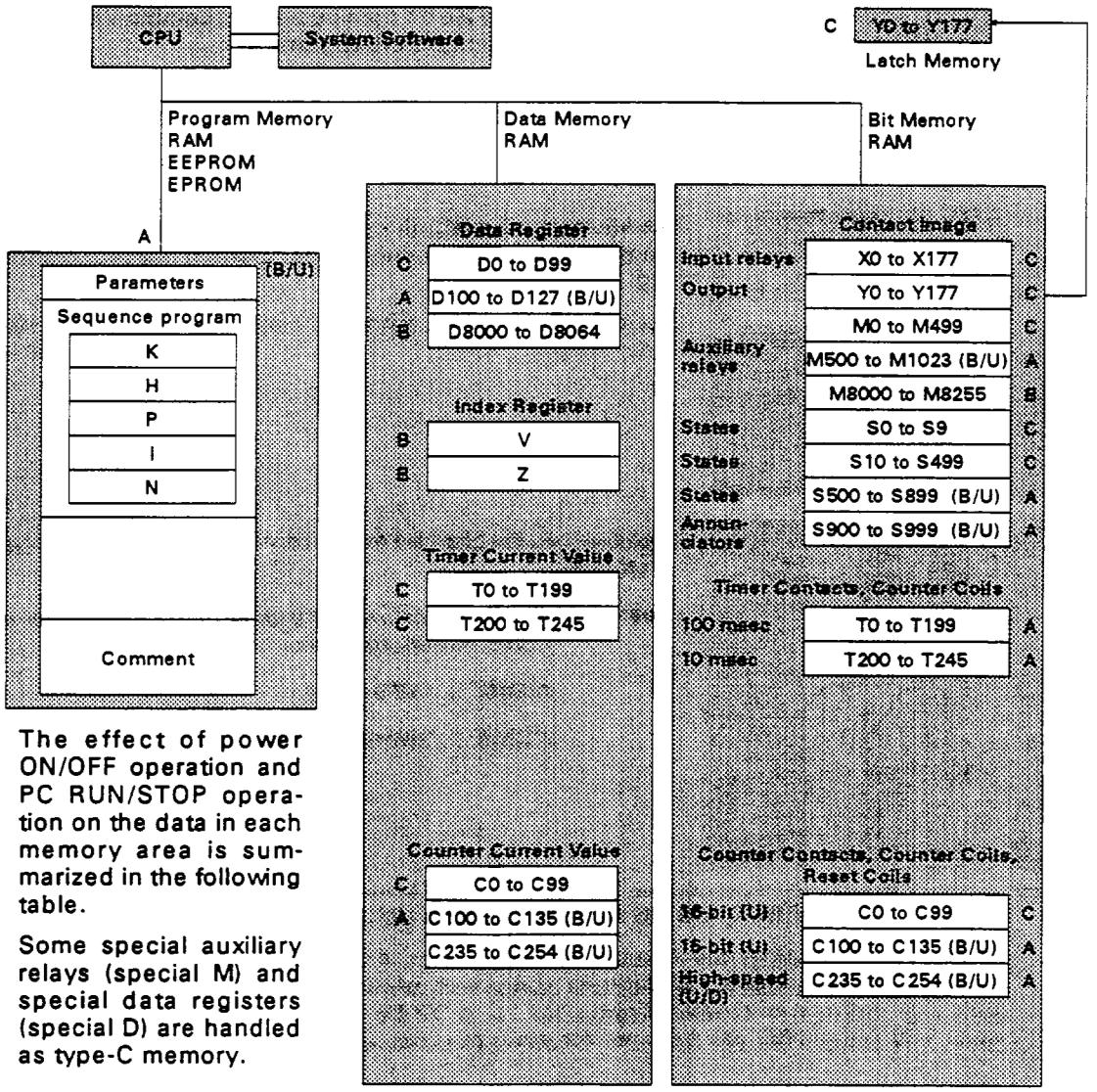


4. DEVICE DETAILS

4.10 SUMMARY OF MEMORY ALLOCATIONS

4.10.1 GENERAL CONFIGURATION

The internal configuration of the devices in the PC is shown below. The memory areas marked with (B/U) are backed up by the battery.



The effect of power ON/OFF operation and PC RUN/STOP operation on the data in each memory area is summarized in the following table.

Some special auxiliary relays (special M) and special data registers (special D) are handled as type-C memory.

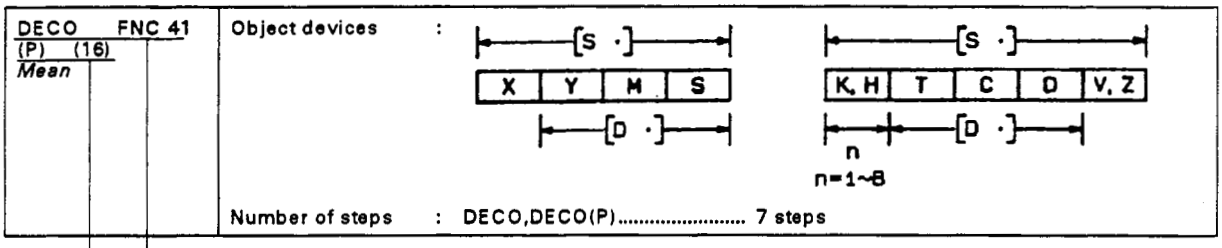
Memory Type	Power OFF	Power OFF → ON	STOP → RUN	RUN → STOP
A: Memories backed up by battery	Not changed			
B: Special M and D and index registers	Cleared	Default	Not changed	
C: Other memories (not backed up by battery)	Cleared		Not changed	Cleared
			Not changed if M8033 is set	

1	GENERAL HARDWARE AND INSTALLATION
2	BASIC SEQUENCE INSTRUCTIONS
3	STEP LADDER INSTRUCTIONS
4	DEVICE DETAILS
5	APPLIED INSTRUCTION
6	SPECIAL DEVICE AND INSTRUCTION LISTS
7	USING SPECIAL UNITS EFFECTIVELY
	APPENDIX

5. APPLIED INSTRUCTIONS

5.1 GENERAL RULES OF APPLIED INSTRUCTIONS

5.1.1 APPLIED INSTRUCTION EXPRESSION FORMAT



DECO : Instruction mnemonic name

(P) : Pulse function MEAN (P) is available.

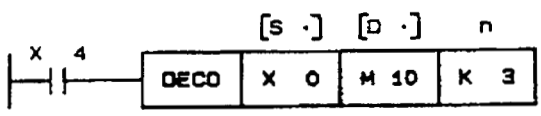
FNC41 : Function number

(16) : 16-bit operation only

If (16/32) : 16-bit and 32-bit (i.e. (D)MOV) operation are available.

(1) INSTRUCTIONS AND OPERANDS

- Applied instructions are designated by a function number (FNC0 to FNC99). Each applied instruction is given a symbol (mnemonic). FNC41 is given the symbol "DECO", for example. When writing a program, the FNC number of the application function to be used can be found by using the HELP function to display a list of FNC numbers and symbols. Both the FNC number and its symbol are displayed when a program is read and displayed.
- Some applied instructions require only the instruction (FNC number) to be designated. Others require the designation of an operand.



LD X4
DECO 41
X 0
M 10
K 3

$\frac{(D0)+(D1)+(D2)}{3} \rightarrow (D4Z)$

This instruction executes processing to determine an average. The data stored in "n" devices, beginning with the device designated by [S·], is added and the sum is divided by "n". The obtained quotient is stored in the device designated by [D·].

[S] : An operand whose data may change after the execution of an instruction is called "source" and is expressed with the symbol [S]. If an index modifier can be used, such an operand is expressed as [S·]. When more than one source is designated, the sources are expressed as [S1·], [S2·], etc.

[D] : An operand whose data changes after the execution of an instruction is called "destination" and is expressed with the symbol [D]. If an index modifier can be used, such a destination is expressed as [D·]. When more than one destination is designated, the destinations are expressed as [D1·], [D2·], etc.

m, n : Operands which accept only constant K or H are expressed as "m" and "n". If more than one such operand is designated, they are expressed as "m1", "m2", etc.

- An applied instruction always occupies 1 program step. Operands occupy either 2 or 4 program steps depending on whether the instruction is a 16-bit instruction or a 32-bit instruction.
- Note that with some functions, programming is allowed for only once even if jumps are used to separate them. However, index registers are usable for changing the device element number.

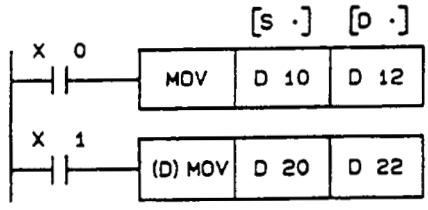
5. APPLIED INSTRUCTIONS

GENERAL RULES OF APPLIED INSTRUCTIONS

5.1.2 DATA LENGTH AND INSTRUCTION EXECUTION FORMAT

(1) 16 BITS AND 32 BITS

- Applied instructions that handle numeric values are either 16 bits or 32 bits depending on the bit length of the numeric data.



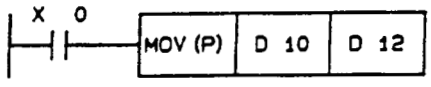
Instruction to move the data in D10 to D12

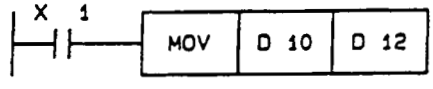
Instruction to move the data in D21 and D20 to D23 and D22

- 32-bit instructions are expressed by appending (D) to an instruction such as (D)MOV and FNC(D)12, (FNC12(D)).
- Both odd and even numbers can be used to designate a device number. The device following the designated device may be paired with the designated device (word devices such as T, C, and D).

  To avoid confusion, it is recommended to always use an even number for the device to be designated by an operand when a 32-bit instruction is used.
- A 32-bit counter (C200 to C255) cannot be used as an operand for a 16-bit instruction.

(2) CONTINUOUS EXECUTION AND PULSE EXECUTION





The symbol (P) appended to an instruction symbol indicates pulse execution form. This symbol can be used with symbol (D) like (D)MOV(P).

With the program shown above, the instruction is executed only once when the state of X0 changes from OFF to ON. This shortens the total program processing time because the instruction is not executed with every operation cycle. Therefore, it is recommended to use pulse execution form instructions.

The program shown above is an example of a continuous execution instruction. The instruction is executed repeatedly every operation cycle while X1 is ON.

Some instructions, such as XCH, INC, and DEC, require care when being used in the continuous execution form. These instructions are indicated by "!".

Note: • *In either case, the instruction is not executed while X0 or X1 is OFF. The destination remains unchanged with the instructions unless specified otherwise.*

GENERAL RULES OF APPLIED INSTRUCTIONS

5.1.3 BIT DEVICES

Bit devices and word devices :

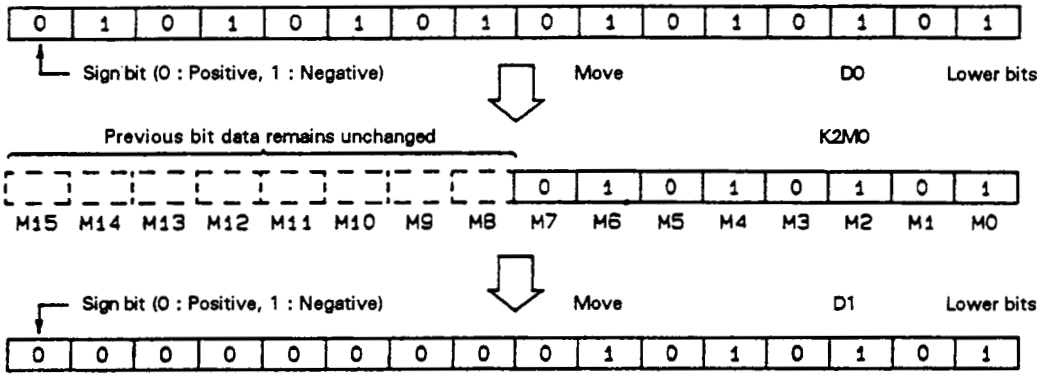
- Devices, such as X, Y, M, and S, which handle only ON/OFF data are called "bit devices". Other devices, such as T, C, M, and D, which handle numeric data are called "word devices".

  Bit devices, however, can be combined handle numeric data. The combination of bit devices is expressed by a digit "n" following K (Kn) and a head device number.

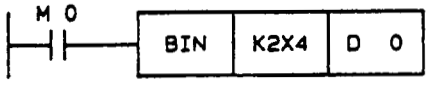
Assigning grouped bit devices :

- Bit devices can be grouped in 4-bit units. The "n" in KnM0 defines the number of groups of 4-bits to be combined for data operation. K1 to K4 are allowed for 16-bit data operation and K1 to K8 are allowed for 32-bit data operation.

  K2M0, for example, indicates 2 groups of 4-bits using bit devices M0 to M7.



- When a 16-bit data is moved to K1M0, K2M0, or K3M0, the overflowing bit data are not moved. This is also true for a 32-bit data.
- When executing a 16-bit data operation and the digit designation for a bit device is K1, K2, or K3, "0" is placed in the higher digit bit position. This means that only positive numbers are handled. This is also true when K1 to K7 is designated while a 32-bit data operation is executed.



2-digit BCD data of X4 to X13 is converted into binary data and transferred to D0.

- Any number can be used for the bit device number. However, it is recommended to use a "0" in the lowest digit place of X and Y numbers (X0, X10, X20, ..., etc.). For M and S, use of a multiple of "8" is efficient. However, because the use of such numbers may lead to confusion in assigning device numbers, it is recommended to use a multiple of "10" as with X and Y.

Ref.: • *Designation of or series of words*

A series of data registers beginning with D1 means D1, D2, D3, ... When words, designated with a bit device, are handled as a series of words, they should be designated in the following manner :

K1X0 K1X4 K1X10 K1X14 ....., K2Y10 K2Y20 K2Y30.....
K3M0 K3M12 K3M24 K3M36....., K4S16 K4S32 K4S48.....

That is, the devices stated above should be used for each digit so that no device is skipped.

If "K2Y0" is used in a 32-bit operation, the upper 16 bits are to be regarded as "0". To obtain 32-bit data, use "K8Y0".

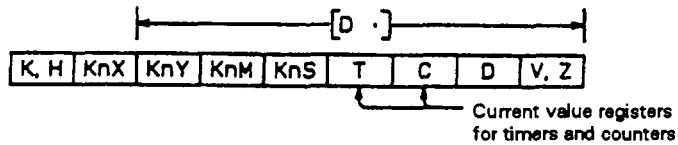
5. APPLIED INSTRUCTIONS

GENERAL RULES OF APPLIED INSTRUCTIONS

5.1.4 INDICES

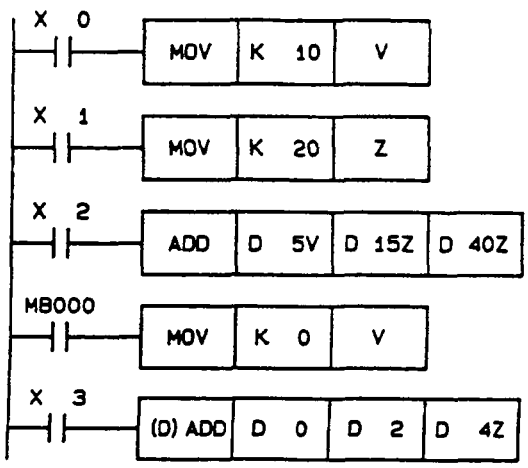
- Index registers are used to modify device numbers in addition to object devices for move and comparison operations. These operations are performed in the same manner as with general-use data registers.

Expressing object device



- The diagram above shows that "KnY to V, Z" can be used as the destination of an applied instruction. The dot (·) in [D·] indicates that an index can be added. However, for 32-bit instructions, V is used for upper 16 bits and Z for the lower 16 bits. Only the designation of Z is allowed as it already represents both V and Z.

  There are no modifiers for V and Z.



- Because K10 is moved to V and K20 to Z, the data in V and Z are "10" and "20" respectively.
- "(D5V) + (D15Z) → (D40Z)" means that "(D15) + (D35) → (D60)".

  V and Z can be used to facilitate programming.
- "(D1, D0) + (D3, D2) → (D25, D24)"

  For 32-bit instructions, V and Z are automatically used in pair. V is assigned to "0" by the MOV instruction.

FLAGS

- For some applied instructions, various flags shown in Section 6.1.3 are used.

  Examples :

  M8020 : Zero flag (Z)
  M8021 : Borrow flag (Br)
  M8022 : Carry flag (Cy)
  M8029 : Execution complete (F)
- If there are errors in the syntax of the applied instruction or the object device numbers, an error will occur (errors are explained in Section 6.1.7). An operation error flag M8067 (E) is set if an error occurs during the execution of an operation.
- These flags are set or reset at each "ON execution" of an instruction. However, their set/reset state does not change in "OFF execution" or at an occurrence of error. Many instructions will affect the flag state; be careful when programming.

5. APPLIED INSTRUCTIONS

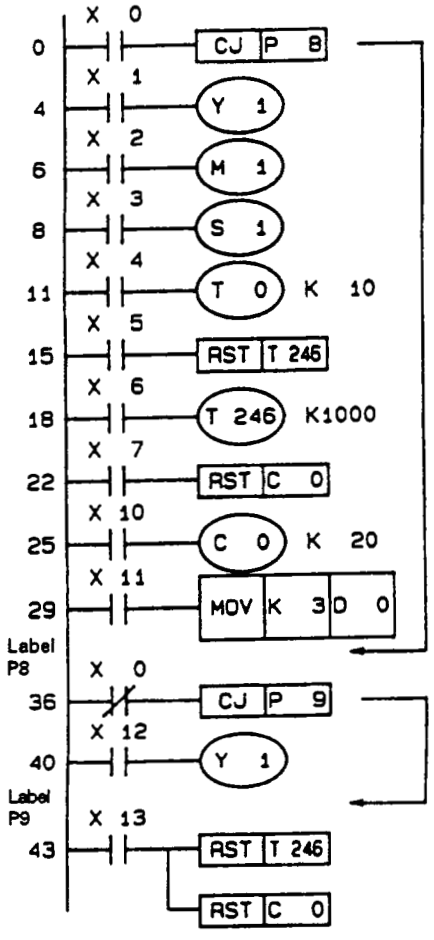
5.2 PROGRAM FLOW (FNC 00 TO FNC 09)

5.2.1 CONDITIONAL JUMP

CJ FNC 00 (P) (16) *Conditional Jump*	Object devices Number of steps	: Pointers P0 to P63 (index modification allowed). P63 is equivalent to the END step and a label program is not necessary. : CJ and CJ(P) .....3 steps Label P**.....1 step

CJ and CJ(P) instructions are provided to skip a part of sequence, thereby reducing the operation cycle time and permits the designation of a double-coil.

In the following example, the sequence jumps from step 1 to step 36 (the step following the label P8) when X0 is ON. If X0 is OFF, the sequence is executed consecutively (from step 1 to step 4). If a jump occurs, the instructions in the skipped sequences are ignored.



The coil do not change even if the contacts do.

Devices	Contact State before Jump	Contact State after Jump	Coil Operation during Jump
Y, M, S	X1, X2, X3 OFF	X1, X2, X3 ON	Y1, M1, S1 OFF
	X1, X2, X3 ON	X1, X2, X3 OFF	Y1, M1, S1 ON
10 msec 100 msec Timer	X4 OFF	X4 ON	Timer not operating
	X4 ON	X4 OFF	Timer stopped Timer restarts after X0 turns OFF
Counter	X7 OFF X10 OFF	X10 ON	Counter not operating
	X7 OFF X10 ON	X10 OFF	Counter stopped Counter restarts after X0 turns OFF
Applied instructions	X11 OFF	X11 ON	An FNC instruction is not executed when a jump is executed except for FNC52 to FNC58 whose operation will continue.
	X11 ON	X11 OFF	

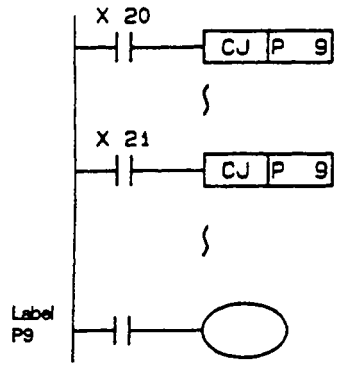
Note:

- *Y1 is a double-coil designation. Its operation is controlled according to the ON/OFF state of X0. That is, it is controlled by X1 if X0 is OFF and by X12 if X0 is ON.*

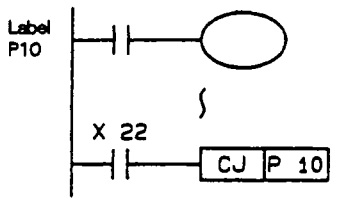
  The designation of a double coil with one coil inside the jump routine and the other coil outside the jump routine is not allowed.

- *If a reset instruction for a retentive timer or counter is designated inside the jump routine, the reset (resetting of contacts or clearing of current value) is still effective if a timer or a counter coil step is skipped.*

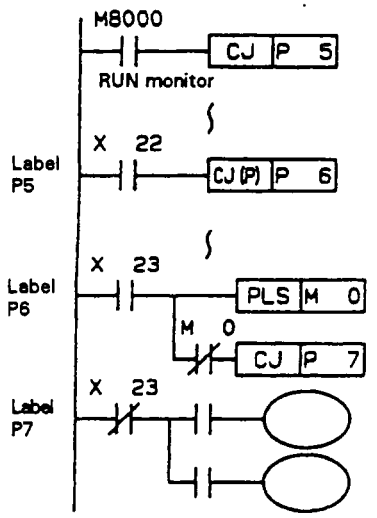
PROGRAM FLOW (FNC 00 TO FNC 09)



- If the same pointer number is designated as an operand while only one label is designated in a program, the following operation occurs.
- If X20 is ON, a jump occurs from this step to label P9. If X20 is OFF and X21 is ON, a CJ instruction in this step becomes effective and a jump occurs from this step to label P9.
- A label number can only be designated once. If the same label number is used more than once, an error occurs.

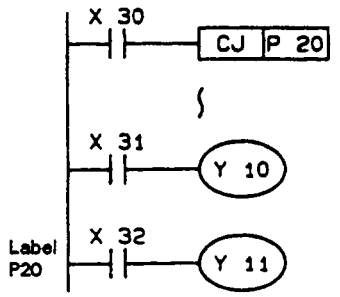


- It is possible to program a label in a step of a lower step number than the step number of the CJ instruction. In the program example, however, if X22 is set ON for more than approximately 100 msec, a watchdog timer error occurs.



- The program step shown on the left is an "unconditional jump" because M8000 is always ON while the PC is running.
- By using the CJ(P) instruction, the jump only becomes effective immediately following the time that X22 changes from OFF to ON.
- The program step shown on the left sets the "CJ P7" instruction one operation cycle after X23 changes from OFF to ON.

  Using this programming method, it is possible to execute a jump after setting each output between step "CJ P7" and label "P7". (useful for setting initial conditions)

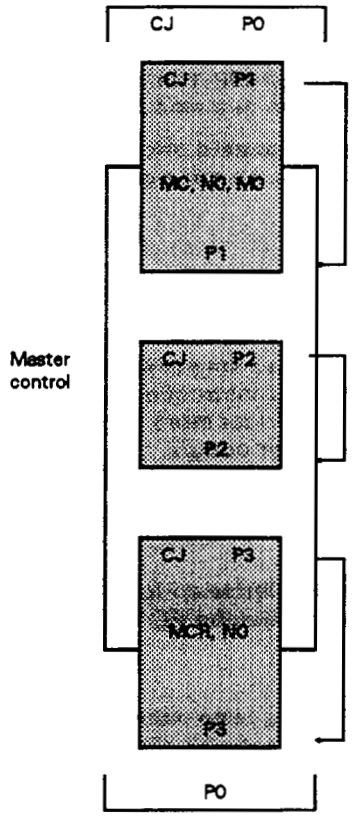


- The label instructions are programmed as shown below.

```
LD   X 30          OUT  Y 10
CJ   P 20               P 20
     S
                   LD   X 32
LD   X 31          OUT  Y 11
```

PROGRAM FLOW (FNC 00 TO FNC 09)

JUMPING AND MASTER CONTROL AREA



(1) Jumping across master control area (from before the master control area to beyond the master control area) is unrestricted.

(2) Jumping from outside the master control area to inside the master control area :

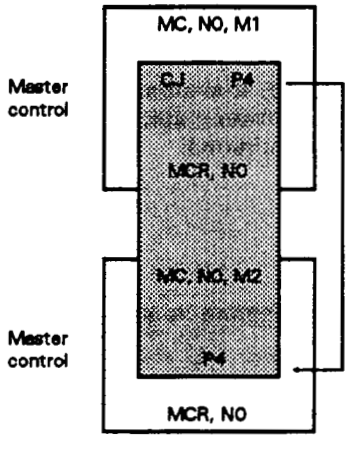
Jumping is made independent of the operation of the master control. After P1, M0 is regarded to be "ON" even if M0 is "OFF".

(3) Jumping within master control area :

Jump is not possible if M0 is OFF.

(4) Jumping from within the master control area to outside the master control area :

Jump is not possible if M0 is OFF. Jumping can occur when M0 is ON, and MCR is ignored but there is no error of operation.



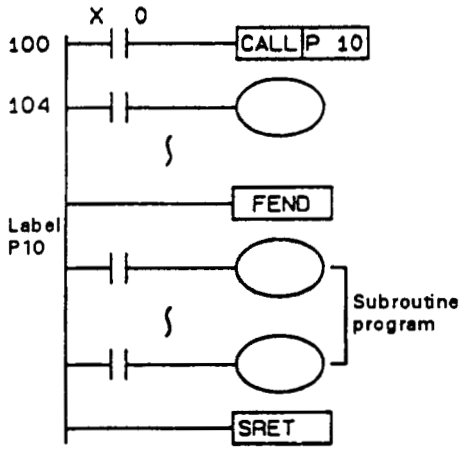
(5) Jump from within the master control area to another master control area :

Jump is possible if M1 is ON. The circuits following the jump operate assuming M2 is ON regardless of the actual ON/OFF state of M2. The first "MCR N0" is ignored.

PROGRAM FLOW (FNC 00 TO FNC 09)

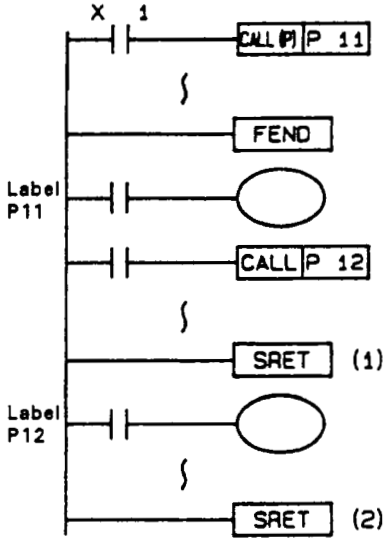
5.2.2 SUBROUTINE

CALL FNC 01 (P) (16) *Subroutine Call*	Object devices Number of steps Nesting	: Pointers P0 to P62 (index modification allowed). : CALL and CALL(P).....3 steps Label P**.....1 step : 5 levels
SRET FNC 02 *Subroutine Return*	Object devices Number of steps	: None : 1 step



- The CALL instruction causes a jump to P10 when X0 is ON.

  The subroutine program is executed. After the execution of the SRET instruction, the program returns to step 104.
- Designate the label in the steps after an FEND instruction (FEND instruction explained in Section 5.2.4).
- Label numbers should range from P0 to P62. Do not use the same label for more than once even if the label is used with a CJ instruction. The same number can be used however, for the operand of many CALL instructions.



- The "CALL(P) P11" instruction executes a jump to label P11 only once when X1 changes from OFF to ON.
- If a CALL instruction to call subroutine program P12 is executed while P11 subroutine program is executed, the program step jumps to subroutine program P12. After the execution of an SRET instruction (2), the program step returns to the step following the CALL instruction in subroutine program P11. The program step returns to the main program after the execution of an SRET instruction (1).

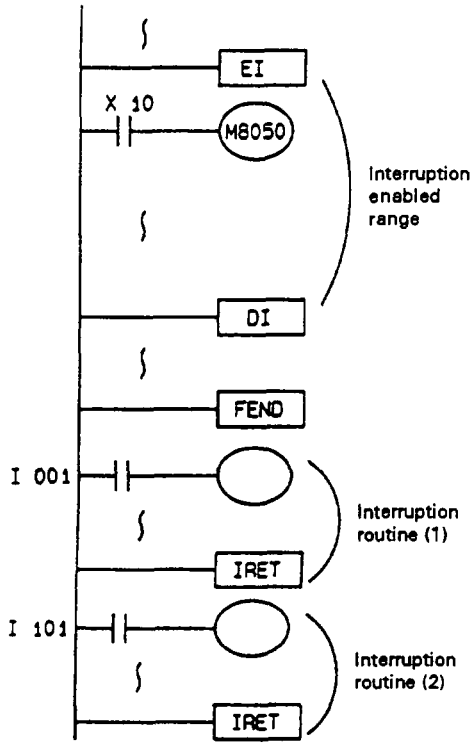
Note:
- *A total of 4 nesting levels can be programmed inside a subprogram called by a CALL instruction.*
- *In a subroutine or an interruption subroutine, use timers in the range of T192 to T199 and T246 to T249.*

PROGRAM FLOW (FNC 00 TO FNC 09)

5.2.3 INTERRUPTS

IRET FNC 03 *Interrupt Return*	Object devices : None Number of steps : 1 step
EI FNC 04 *Interrupt Enable*	Object devices : None Number of steps : 1 step
DI FNC 05 *Interrupt Disable*	Object devices : None Number of steps : 1 step

Note:
- *An interrupt pointer (I***) uses 1 step. Up to 9 points can be designated.*
- *See Section 4.4.1 for the procedure used to assign interrupt pointers.*
- *Multiple interruption is allowed for up to 2 points.*
- *Interrupt signals must be of pulse width wider than 500 μsec.*



- The PC is usually in the interrupt disabled state.

  If X0 and X1 are turned ON while a program step ranging from instruction EI to instruction DI is being executed, interrupt routines (1) and (2) are executed respectively. The program sequence returns to the main program when an IRET instruction is executed in each interrupt routine.

- An interrupt routine will not be executed if the corresponding special auxiliary relay is activated. For example, interrupt routine IΔ** will not be executed if relay M805Δ is activated.

  In the program shown on the left, interrupt routine I001 is not executed when X0 changes from OFF to ON if X10 stays ON.

- While an interrupt program is executed, other interrupt calls are disabled. However, 2 levels of interrupt programs may be nested by programming instruct EI and DI within the interrupt program.

- In a subroutine or an interrupt subroutine, use timers in the range of T192 to T199.

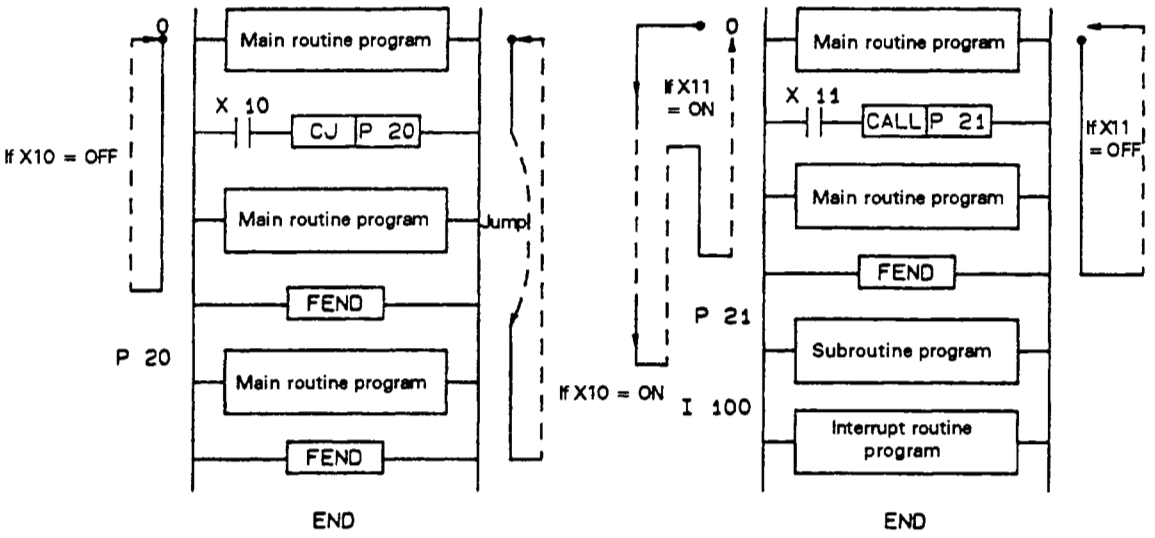
Note:
- *If more than one interrupt occurs sequentially, priority is given to the interrupt occurring first. If two or more interrupt events occur simultaneously, the interrupt subroutine with a lower pointer number is given priority.*
- *If an interrupt occurs in an interrupt disabled range (DI to EI range), the occurrence is stored and is executed after the EI instruction is executed (unless special relay M805Δ is activated).*

PROGRAM FLOW (FNC 00 TO FNC 09)

5.2.4 MAIN PROGRAM END

FEND FNC 06 *First End*	Object devices : None Number of steps : 1 step

An FEND instruction indicates the end of a main routine program. The program step returns to step 0 when this instruction is executed after the output processing, input processing, and watchdog timer are refreshed.



- Write labels for CALL or CALL(P) instructions in the steps following the FEND instruction. The subroutine program called by a CALL or CALL(P) instruction must always end with an SRET instruction.

  Similarly, an interrupt pointer should be written after the FEND instruction. The interrupt program must end with an IRET instruction.
- A program will be regarded as an error if the FEND instruction is executed after the CALL or CALL(P) instruction is executed and before the SRET instruction is executed. Another cause for a program to be regarded as an error is if the FEND instruction is executed during a FOR - NEXT loop.
- The subroutine programs and interrupt routine programs should always be written between the final FEND instruction and the END instruction.

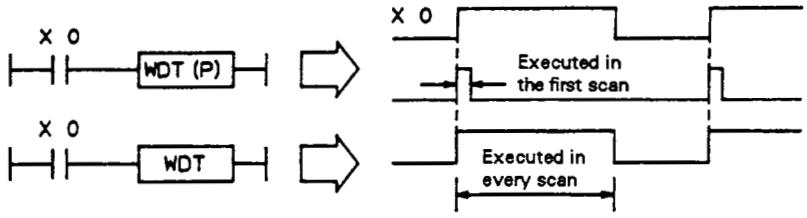
5. APPLIED INSTRUCTIONS

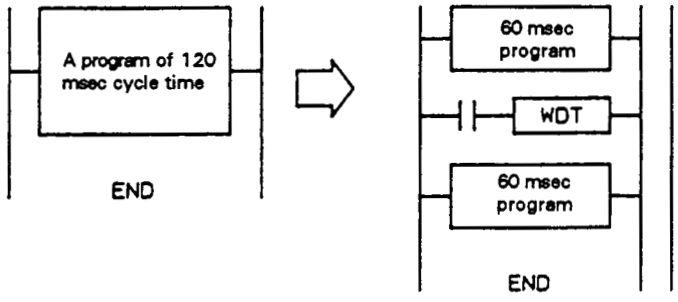
PROGRAM FLOW (FNC 00 TO FNC 09)

5.2.5 WATCHDOG TIMER

WDT FNC 07 (P) *Watchdog Timer*	Object devices : None Number of steps : 1 step

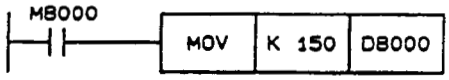
A WDT instruction refreshes the watchdog timer of the sequence program. If the operation cycle time (step 0 to END or FEND instruction) of a sequence program exceeds 100 msec, the PC will stop. In such a case, a WDT instruction is inserted at a proper program step to refresh a watchdog timer so that the sequence program can be continuously executed to the END instruction.





For example, split a 120 msec cycle time program into two programs (each 60 msec) and designate a WDT instruction between these two programs.

- If the scan time exceeds 100 msec in each scan, change the data in the special data register D8000 by using a MOV instruction (FNC12) as indicated below.



- In addition to the example shown above, the WDT instruction is used for the following purposes:
  1) It can be designated after a label if the label corresponding to the CJ instruction is designated in a step of a lower step number than the CJ instruction step number.
  2) It can be designated inside a FOR-NEXT loop.

PROGRAM FLOW (FNC 00 TO FNC 09)

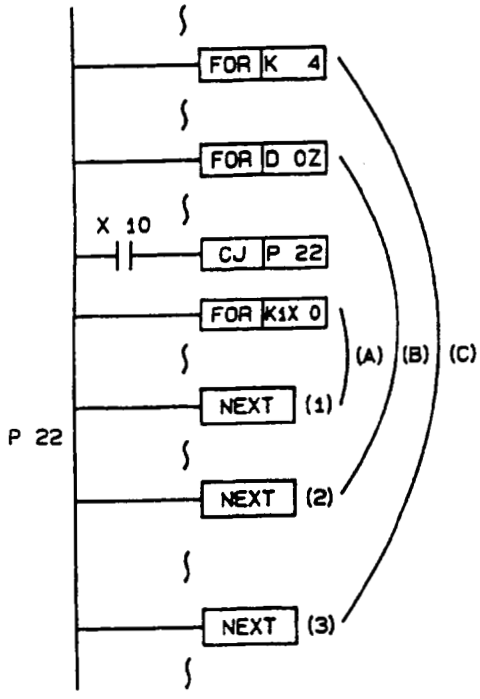
5.2.6 REPEAT

FOR FNC 08 (16) *Start of Repeat Range*	Object devices : [S ·] K, H | KnX | KnY | KnM | KnS | T | C | D | V, Z Number of steps : 3 steps Nesting : 5 levels
NEXT FNC 09 *End of Repeat Range*	Number of steps : 1 step

The processes within in a FOR-NEXT loop is repeated "n" times (designated by the object device). After that, the steps following the NEXT instruction steps are executed.

Repeat number range : 1 to 32,767

If a number in the range of -32,768 to 0 is designated, it is replaced by 1 and the FOR-NEXT loop is executed once.



- Program (C) is repeated four times and then the steps following the 3rd NEXT instruction step are executed.
- Program (B) is repeated six times each time program (C) is executed if the data in data register D 0Z is "6".

  Therefore, program (B) will be executed 24 times.
- Use the CJ instruction (X10 = ON) to skip the FOR-NEXT loop program (A).

  If X10 is OFF and the contents of K1X0 are "7", program (A) is executed seven times while program (B) is executed once. This means that program (A) is repeated 168 times (4 x 6 x 7) in total.
- A maximum of 5 FOR-NEXT loops may be nested.

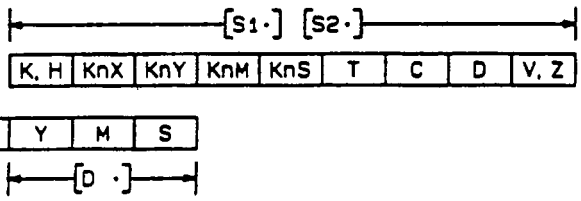
Note: • *An error will occur in the following cases :*

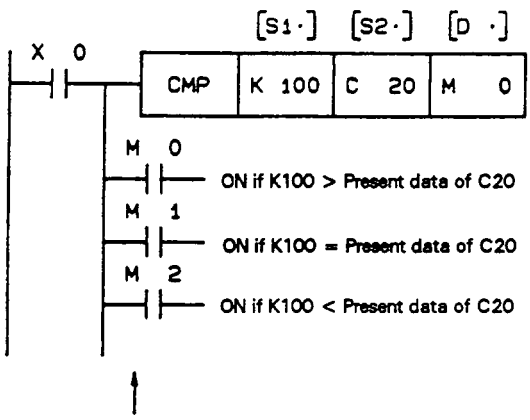
- *If a NEXT instruction is designated preceding a FOR instruction.*
- *If there is no NEXT instruction.*
- *If a NEXT instruction is designated after an FEND or END instruction.*
- *If the number of NEXT instructions does not agree with the number of FOR instructions.*

5. APPLIED INSTRUCTIONS

5.3 MOVE AND COMPARE (FNC 10 TO FNC 19)

5.3.1 COMPARE

CMP FNC 10 (P) (16/32) *Compare*	Object devices :  Number of steps : CMP and CMP(P) ..... 7 steps (D)CMP and (D)CMP(P) ..... 13 steps



M0, M1, and M2 remain unchanged if the CMP instruction is executed with X0 = OFF.

- The data of source [S1·] and [S2·] are compared and destination [D·] operates according to the result of this comparison.

  Source data is compared algebraically (–10 < 2, for example).
- All source data is regarded as binary data.

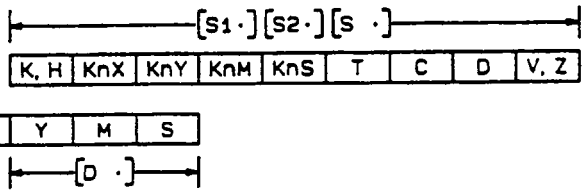
Ref.:

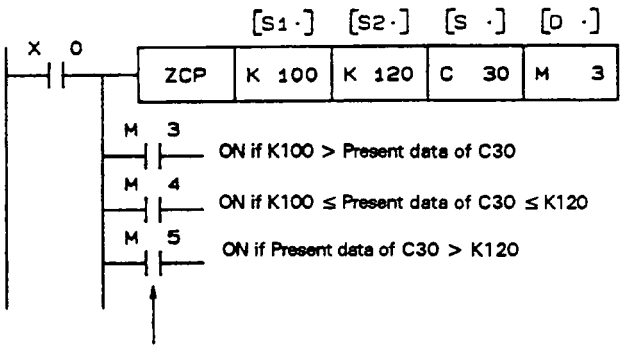
- *A CMP applied instruction uses three operands. If only one or two operands are designated, an error occurs (error code : 6503), precluding PC operation.*
- *If a device other than those listed above is designated as an operand, an error occurs (error code : 6705). If device X, D, T, or C is used as a destination, for example, an error occurs.*
- *If the device number that designated as an operand exceeds the allowable range, an error occurs (error code : 6706)*

  Note : It may occur if an index modifier is used.
- *If an error occurs due to incorrect operand designation as stated above, see Sections 6.1.7 and 6.1.8.*

MOVE AND COMPARE (FNC 10 TO FNC 19)

5.3.2 ZONE COMPARE

ZCP FNC 11 (P) (16/32) *Zone Compare*	Object devices :  Number of steps : ZCP and ZCP(P) ..... 9 steps (D)ZCP and (D)ZCP(P) ..... 17 steps



M3, M4, and M5 remain unchanged if the ZCP instruction is executed with X0 = OFF.

A ZCP instruction is used to compare the data with two source data values.

- Data from source [S1·] must not be greater than that from source [S2·].

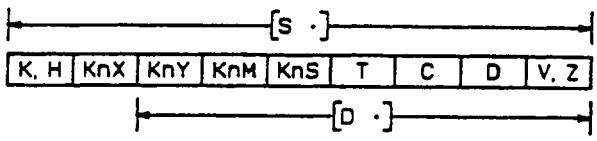
  For example, if [S1·] = K100
  [S2·] = K90,
  the ZCP instruction is executed assuming [S2·] = K100.

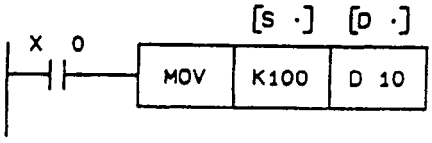
- Source data is compared algebraically (−10 < 2, for example).

Note: • *M3, M4, or M5 is turned ON according to the comparison result.*

MOVE AND COMPARE (FNC 10 TO FNC 19)

5.3.3 MOVE

MOV FNC 12 (P) (16/32) *Move*	Object devices :  Number of steps : MOV and MOV(P) ..... 5 steps (D)MOV and (D)MOV(P) ..... 9 steps



(K100) → (D10)

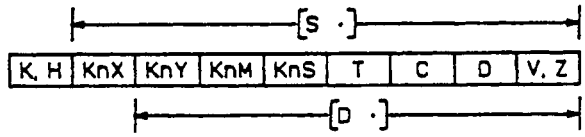
The source data is moved to the destination.

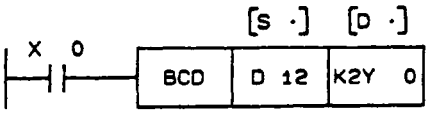
- The data in the source device is moved to the designated destination. If X0 = OFF, the data remains unchanged.
- Constant K100 is automatically converted into binary data when the MOV instruction is executed.

5. APPLIED INSTRUCTIONS

MOVE AND COMPARE (FNC 10 TO FNC 19)

5.3.4 BINARY CODED TO DECIMAL

BCD FNC 18 (P) (16/32) *Binary Coded To Decimal*	Object devices :  Number of steps : BCD and BCD(P) ..... 5 steps (D)BCD and (D)BCD(P) ..... 9 steps

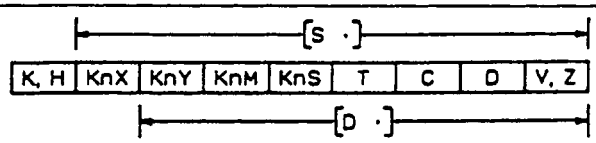


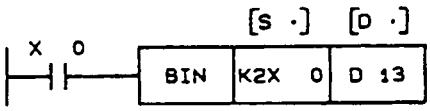
Binary data in source device is converted to BCD data and moved to the destination device.

- If the result of BCD conversion executed by a BCD or BCD(P) instruction is outside the range of "0 to 9,999", an error occurs.

  If the result of BCD conversion executed by a (D)BCD or (D)BCD(P) instruction, is outside the range of "0 to 99,999,999", an error occurs.
- The BCD instruction is used to convert binary data in the PC to be output in a seven segment display, for example.

5.3.5 BINARY

BIN FNC 19 (P) (16/32) *Binary*	Object devices :  Number of steps : BIN and BIN(P) ..... 5 steps (D)BIN and (D)BIN(P) ..... 9 steps Range : 0 to 9,999 or 0 to 99,999,999



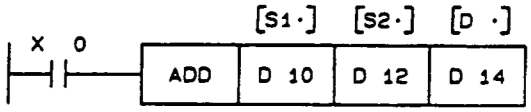
BCD data in source device is converted to binary data and moved to the destination device.

- The BIN instruction is used to input the setting value of BCD digital switches to the PC. If the source device data is not BCD, an operation error occurs to set M8067 ON. In this case, however, M8068 (operation error latch) is not set ON.
- Constants K are not processed as object devices of this instruction because they are automatically converted to binary data before any processing.

5.4 ARITHMETIC AND LOGICAL OPERATIONS (FNC 20 TO FNC 29)

5.4.1 ADDITION

ADD FNC 20 (P) (16/32) *Addition*	Object devices : [S1·] [S2·] K, H | KnX | KnY | KnM | KnS | T | C | D | V, Z [D ·] Number of steps : ADD and ADD(P) ..... 7 steps (D)ADD and (D)ADD(P) ..... 13 steps Flags : M8020 (zero), M8021 (borrow), and M8022 (carry)



(D10) + (D12) → (D14)

- The binary data in two designated source devices are added and moved to the designated destination device. The most significant bit in each piece of data is used as the sign bit (positive when "0" and negative when "1"). The data is added algebraically such as :

  (5 + (–8) = –3)

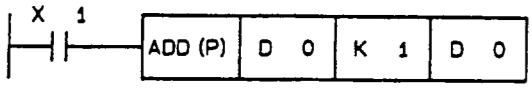
- If the result of operation is "0", the zero flag is set.

  If the result of operation exceeds 32,767 (16-bit operation) or 2,147,483,647 (32-bit operation), the carry flag is set. See the following page.

  If the result of operation is –32,767 or smaller (16-bit operation) or –2,147,483,647 or smaller (32-bit operation), the borrow flag is set. See the following page.

- In a 32-bit operation, the device of the lower 16 bits is designated when a word device is used. The device which follows the designated device is the upper 16-bit portion of the 32-bit double word. To avoid using the same device number, it is recommended to use even numbers when designating a device.

- The same device number can be used for the source and destination. If such a designation is made, the result of the addition will change with every operation cycle if a continuous execution instruction (ADD, (D)ADD) is used.



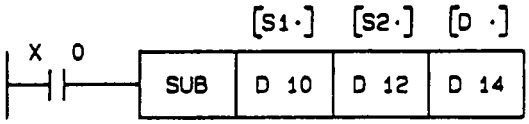
(D0) + 1 → (D0)

- When a program such as that indicated above is written, "1" is added to D0 data each time X1 goes from OFF to ON. This change is similar to the change found when an INC(P) instruction is executed. The differences are that with the ADD instruction, the zero, borrow, or carry flag is set according to the process result.

ARITHMETIC AND LOGICAL OPERATIONS (FNC 20 TO FNC 29)

5.4.2 SUBTRACTION

SUB FNC 21 (P) (16/32) *Subtraction*	Object devices : [S1·] [S2·] K, H | KnX | KnY | KnM | KnS | T | C | D | V, Z [D ·] Number of steps : SUB and SUB(P) ..... 7 steps (D)SUB and (D)SUB(P) ..... 13 steps Flags : M8020 (zero), M8021 (borrow), and M8022 (carry)

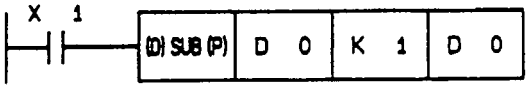


(D10) – (D12) → (D14)

- Data in the device designated by [S2·] is algebraically subtracted from the data in the device designated by [S1·]. The result is stored in the device designated by [D·].

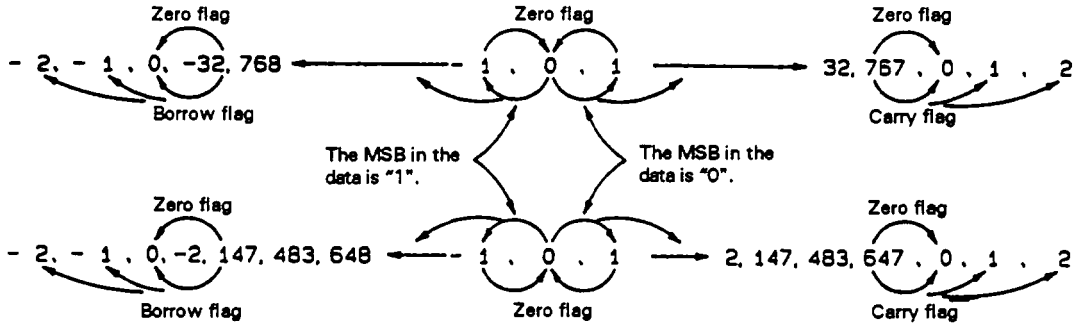
  Example : 5 – (–8) = 13

- The function of each flag, the device designation method for 32-bit operation, the difference between continuous execution and pulse execution instructions, etc. are the same as explained for the ADD instruction in Section 5.4.1.



(D1, D0) – 1 → (D1, D0)

- The operation indicated above is very similar to the operation executed by a (D)DEC(P) instruction. The difference is that the flags are obtained when a SUB instruction is used.
- How the flags are set and the relationships between flag setting and positive/negative numbers are shown below.

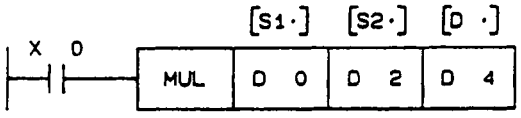


ARITHMETIC AND LOGICAL OPERATIONS (FNC 20 TO FNC 29)

5.4.3 MULTIPLICATION

MUL FNC 22 (P) (16/32) *Multiplication*	Object devices : [S1·] [S2·] K, H | KnX | KnY | KnM | KnS | T | C | D | V, Z [D ·] * Only Z is allowed for 16-bit operation Number of steps : MUL and MUL(P) ..... 7 steps (D)MUL and (D)MUL(P) ..... 13 steps

(1) 16-BIT OPERATION

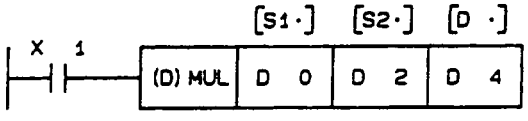


BIN (D0) x BIN (D2) → BIN (D5, D4)
16 bits 16 bits 32 bits

- The product of the data in the designated source devices is stored at the designated destination device as a 32-bit data. The lower 16 bits in the designated device and the upper 16 bits in the next device.

  If D0 = 8 and D2 = 9, in the program above, (D5, D4) = 72.
- The most significant bit indicates the plus or minus sign (0 : plus, 1 : minus).
- V cannot be designated for [D·]. For bit devices, bits can be designated between K1 and K8. Remember that the result will be a 32 bit data, so if "K4" is designated, the lower 16 bits only are obtained as the result of the multiplication.

(2) 32-BIT OPERATION



BIN (D1, D0) x BIN (D3, D2) → BIN (D7, D6, D5, D4)
32 bits 32 bits 64 bits

- If a bit device is designated for the destination in a 32-bit operation, only the lower 32 bits of the product are obtained. The upper 32 bits are discarded. In this case, conduct operation after moving the data to a word device.
- When word devices are used, it is not possible to monitor the contents of this 64-bit data.

  In this case, calculate by monitoring the upper and lower 32-bit words of the result by using the following :

$$64 \text{ bit result} = (\text{upper 32-bit}) \times 2^{32} + (\text{lower 32-bit})$$

- The most significant bit indicates the plus or minus sign.
- V and Z cannot be designated for [D·].

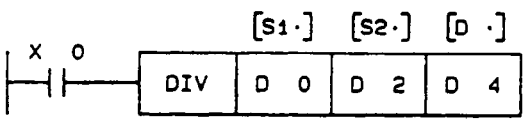
5. APPLIED INSTRUCTIONS

ARITHMETIC AND LOGICAL OPERATIONS (FNC 20 TO FNC 29)

5.4.4 DIVISION

DIV FNC 23 (P) (16/32) *Division*	Objective devices : [S1·] [S2·] [D ·] K, H | KnX | KnY | KnM | KnS | T | C | D | V, Z * Only Z is allowed for 16-bit operation Number of steps : DIV and DIV(P) ..... 7 steps (D)DIV and (D)DIV(P) ..... 13 steps

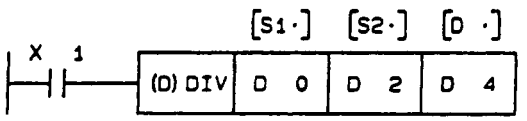
(1) 16-BIT OPERATION



Dividend		Divisor		Quotient	Remainder
BIN		BIN		BIN	BIN
(D0)	÷	(D2)	→	(D4)	(D5)
16 bits		16 bits		16 bits	16 bits

- Division is executed by taking the data in the device designated as [S1·] as a dividend and the data in the device designated as [S2·] as a divisor. The quotient is stored in the device designated as [D·] and the remainder in its succeeding device.
- V cannot be designated for [D·].

(2) 32-BIT OPERATION



Dividend		Divisor		Quotient	Remainder
BIN		BIN		BIN	BIN
(D1, D0)	÷	(D3, D2)	→	(D5, D4)	(D7, D6)
32 bits		32 bits		32 bits	32 bits

- In this division process, the dividend is the data in the device designated by [S1·] and the following device and the divisor is the data in the device designated by [S2·] and its succeeding device. The quotient and remainder are stored in 4 devices beginning with the device designated as [D·] as shown above.
- V and Z cannot be designated for [D·].

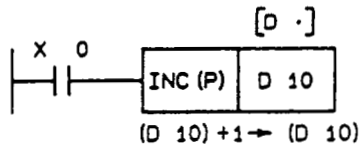
Ref.:
- *If a divisor is "0", an operation error occurs and the instruction is not executed.*
- *If a bit device is designated as [D·], a remainder is not obtained.*
- *The most significant bit in quotient and remainder indicates the plus or minus sign.*

5. APPLIED INSTRUCTIONS

ARITHMETIC AND LOGICAL OPERATIONS (FNC 20 TO FNC 29)

5.4.5 INCREMENT AND DECREMENT

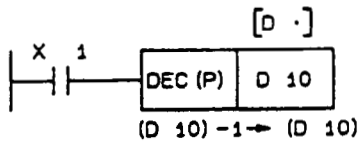
INC FNC 24 (P) (16/32) ("1") *Increment* DEC FNC 25 (P) (16/32) ("1") *Decrement*	Object devices : K, H | KnX | KnY | KnM | KnS | T | C | D | V, Z [D ·] Number of steps : INC, INC(P), DEC, and DEC(P) ..... 3 steps (D)INC, (D)INC(P), (D)DEC, and (D)DEC(P) ..... 5 steps



- In the example, the numerical value of the device designated by [D·] increases by "1" each time X0 is turned ON.

  If the instruction is not a pulse instruction, addition occurs in each operation cycle.
- In 16-bit operation, adding "1" to +32,767 changes the data to –32,768. No flag is set.

  In 32-bit operation, adding "1" to +2,147,483,647 changes the data to –2,147,483,648. Again, no flag is set.

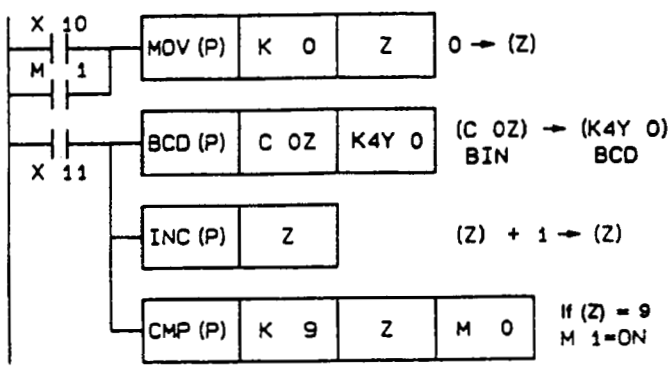


- In the example, the numerical value of the device designated by [D·] decreases by "1" each time X0 is turned ON.

  If the instruction is not a pulse instruction, subtraction occurs in each operation cycle.
- In 16-bit operation, subtracting "1" from –32,768 changes the data to +32,767. No flag is set.

  In 32-bit operation, subtracting "1" from –2,147,483,648 changes the data to +2,147,483,647. Again, no flag is set.

(1) APPLICATION EXAMPLE



- The current value stored in counters C0 to C9 is converted into BCD data and output to K4Y0.
- Z is cleared by the reset input X10.
- The present data stored in counters C0 to C9 is output sequentially each time X11 is turned ON.

ARITHMETIC AND LOGICAL OPERATIONS (FNC 20 TO FNC 29)

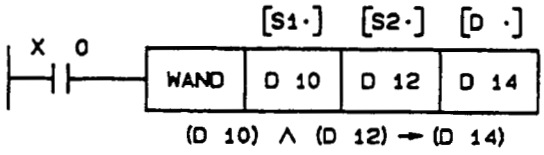
5.4.6 LOGICAL AND LOGICAL SUM, AND EXCLUSIVE OR

AND FNC 26 (P) (16/32) *And* OR FNC 27 (P) (16/32) *Or* XOR FNC 28 (P) (16/32) *Exclusive Or*	Object devices : [S1·] [S2·] K, H | KnX | KnY | KnM | KnS | T | C | D | V, Z [D ·] Number of steps :

7 steps (16-bit operation)		13 steps (32-bit operation)	
WAND WAND(P)	(FNC 26) (FNC 26(P))	(D)AND (D)AND(P)	(FNC(D)26) (FNC(D)26(P))
WOR WOR(P)	(FNC 27) (FNC 27(P))	(D)OR (D)OR(P)	(FNC(D)27) (FNC(D)27(P))
WXOR WXOR(P)	(FNC 28) (FNC 28(P))	(D)XOR (D)XOR(P)	(FNC(D)28) (FNC(D)28(P))

A "W" is appended to 16-bit instructions.

Logical AND

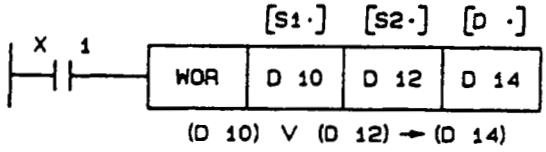


- Logical multiplication operation is executed in units of bits.

  1 ∧ 1 = 1 0 ∧ 1 = 0

  1 ∧ 0 = 0 0 ∧ 0 = 0

Logical OK

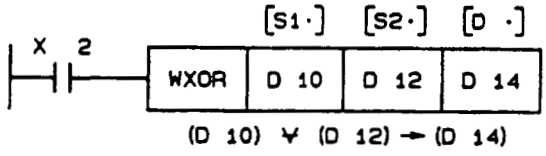


- Logical addition operation is executed in units of bits.

  1 ∨ 1 = 1 0 ∨ 1 = 1

  0 ∨ 0 = 0 1 ∨ 0 = 1

Exclusive OR

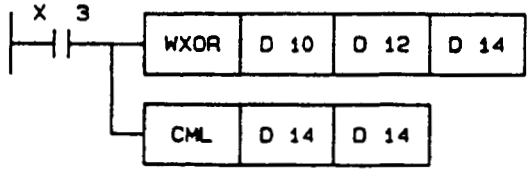


- Exclusive OR operation is executed in units of bits.

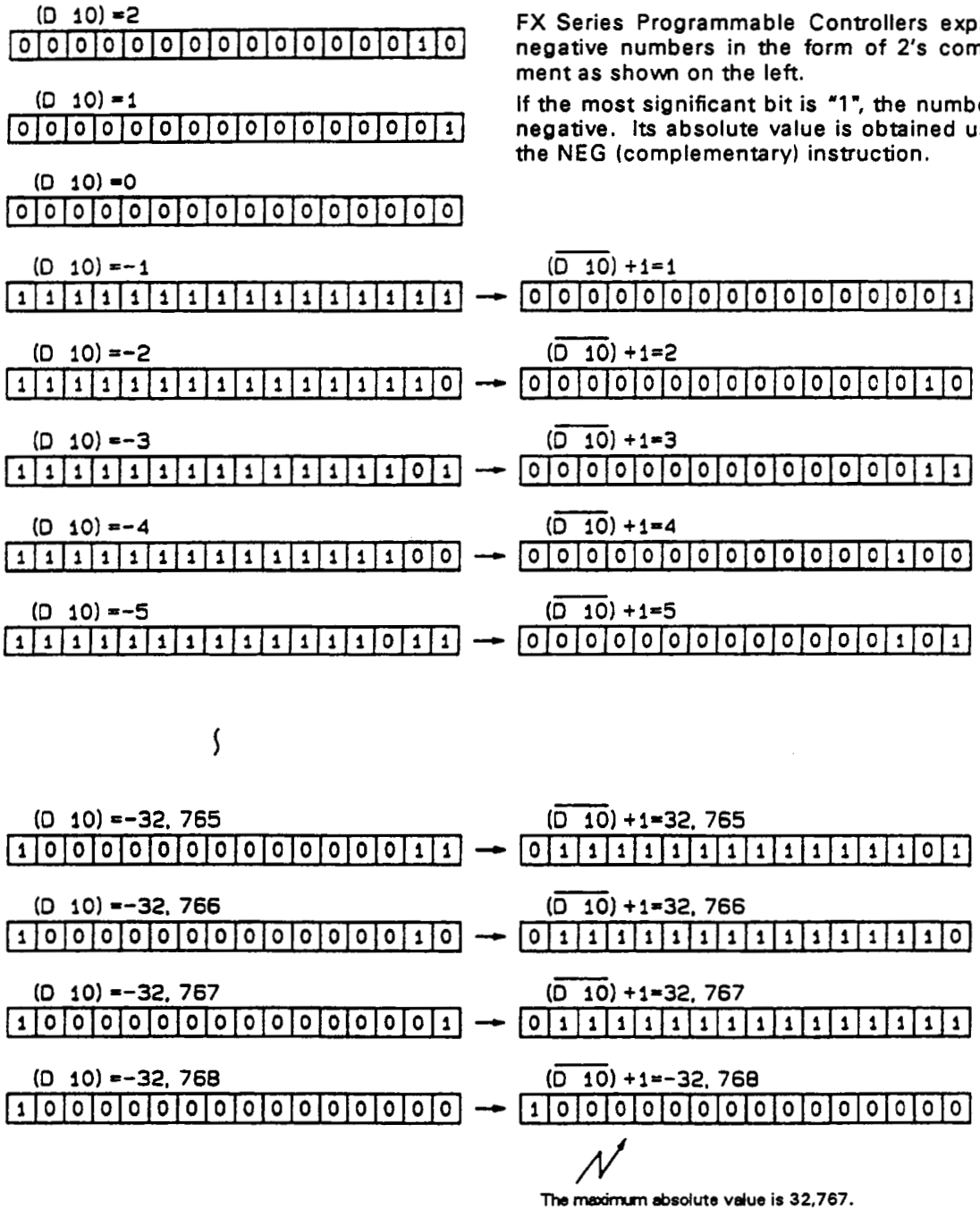
  1 ⊻ 1 = 0 0 ⊻ 0 = 0

  1 ⊻ 0 = 1 0 ⊻ 1 = 1

"Exclusive NOR" operation is possible by combining the XOR instruction with FNC14 (CML).



EXPRESSION OF NEGATIVE NUMBERS AND THEIR ABSOLUTE VALUES (REFERENCE)



FX Series Programmable Controllers express negative numbers in the form of 2's complement as shown on the left.

If the most significant bit is "1", the number is negative. Its absolute value is obtained using the NEG (complementary) instruction.

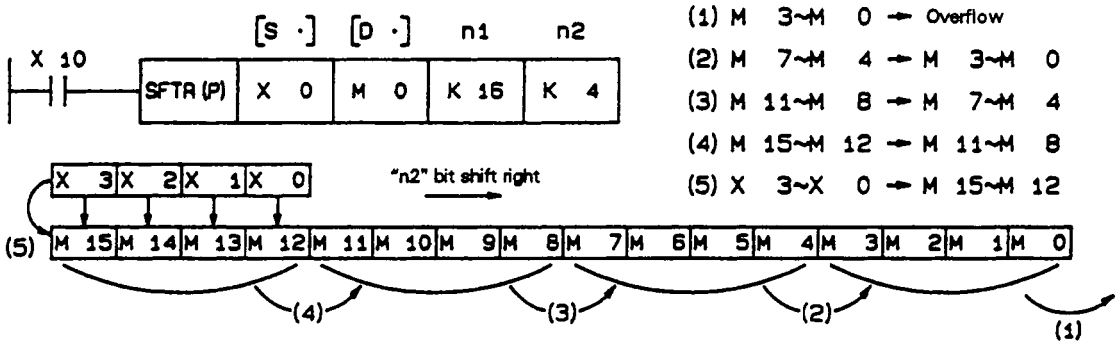
ROTATION AND SHIFT (FNC 30 TO FNC 39)

5.4.7 SHIFT (RIGHT/LEFT)

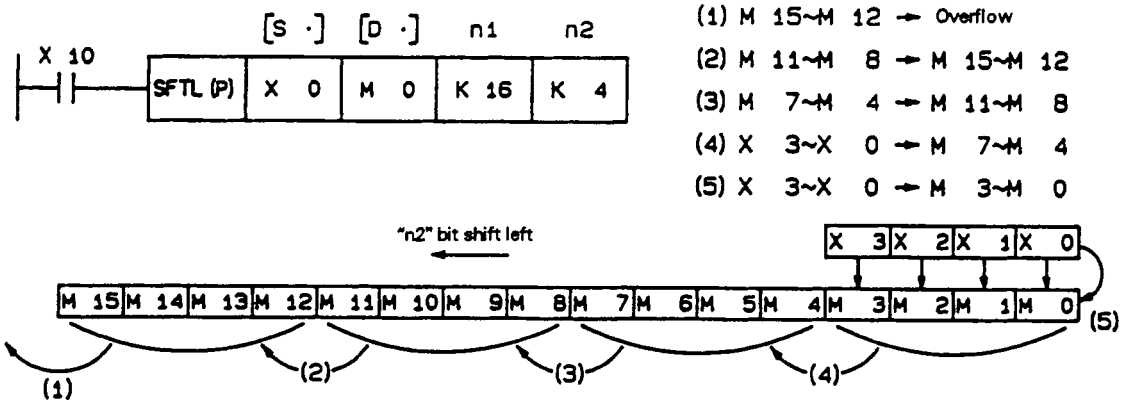
SFTR FNC 34 (P) (16) (↑↑) *Shift Right* SFTL FNC 35 (P) (16) (↑↑) *Shift Left*	Object devices : [S ·] : X, Y, M, S [D ·] : X, Y, M, S n1, n2 : K, H n2≤n1≤1024 Number of steps : SFTR, SFTR(P) ..... 9 steps SFTL, SFTL(P) ..... 9 steps

- The instruction is used to execute an n2 number of bit shift to the right (SFTR) or left (SFTL) for an n1 long bit device (n2 ≤ n1 ≤ 1024).
- Shift is executed each time the drive input is turned from OFF to ON when a pulse instruction is used. If a continuous execution instruction is used, shift is executed with each operation cycle.

Bit Shift Right



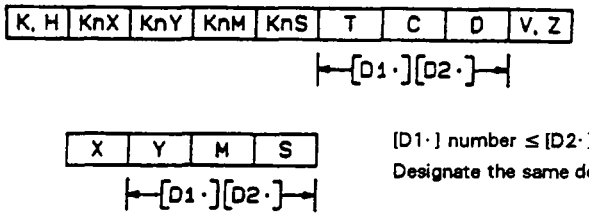
Bit Shift Left

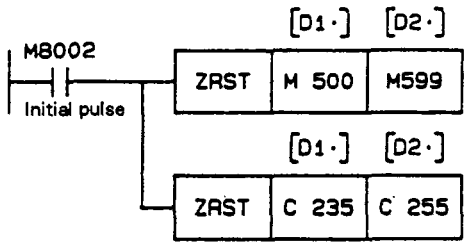


5. APPLIED INSTRUCTIONS

5.5 DATA OPERATION (FNC 40 TO FNC 49)

5.5.1 ZONE RESET

ZRST FNC 40 (P) (16) *Zone Reset*	Object devices :  Number of steps : ZRST, ZRST(P) ..... 5 steps



Bit devices M500 through M599 are batch reset.

Word devices C235 through C255 are batch reset.
("0" is written and the contacts and coil image are cleared)

- For [D1·] and [D2·], designate the same device. The device numbers must be [D1·] number ≤ [D2·] number.

  If [D1·] number > [D2·] number, only the device designated by [D1·] is reset.
- Although the ZRST instruction is processed as a 16-bit instruction, 32-bit counters are designated for [D1·] and [D2·].

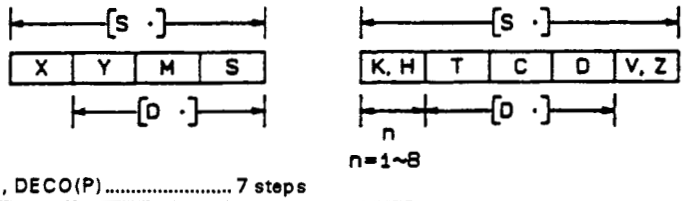
Note: • *Combined designation of both a 16-bit and 32-bit counter (for example, a 16-bit counter for [D1·] and 32-bit counter for [D2·]) is not allowed.*

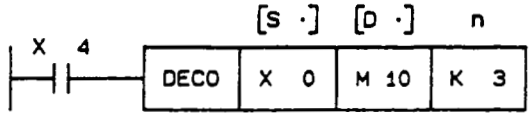
OTHER RESET INSTRUCTIONS

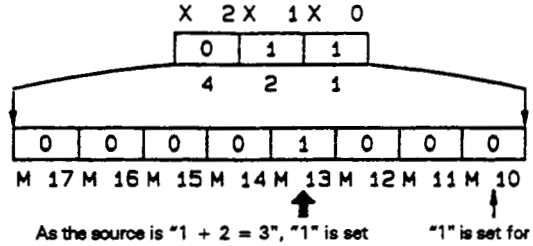
- A RST instruction is provided to reset devices individually.
- An FNC16 instruction (FMOV) is provided to block write K0. Using this instruction, it is possible to write "0" to devices KnY, KnM, KnS, T, C, and D.

DATA OPERATION (FNC 40 TO FNC 49)

5.5.2 DECODE

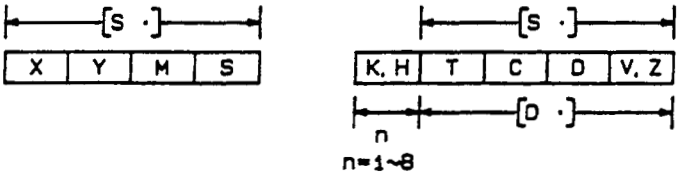
DECO (P) (16) *Decode* FNC 41	Object devices :  Number of steps : DECO, DECO(P) ..... 7 steps

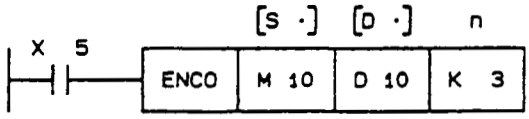


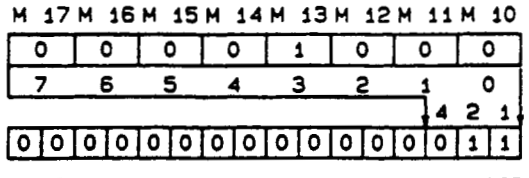


- If designated destination device [D·] is T, C, or D, "n" must be 4 or less. Each bit of the destination device is controlled.

5.5.3 ENCODE

ENCO (P) (16) *Encode* FNC 42	Object devices :  Number of steps : ENCO, ENCO(P) ..... 7 steps





- If designated source device [S·] is T, C, D, V, or Z, "n" must be 4 or less. If more than one bit in the designated source device is "1", only the highest position bit for which "1" is set becomes effective.
- If all of the bit data in the designated source device is "0", an error occurs.

POINTS FOR DECODE/ENCODE

- If "n = 0", processing is not executed.
- If [D·], designated with a decode instruction, or [S·], designated with an encode instruction, is a bit device when "n = 8", the number of points is 256 (2^8).
- The designated instruction is not executed if the drive input is OFF. The currently active decode output remains active even if the drive input is then turned OFF.

5.6 HIGH-SPEED PROCESSING (FNC 50 TO FNC 59)

5.6.1 REFRESH

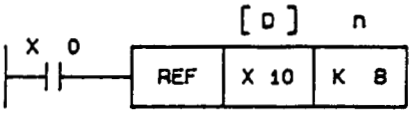
REF FNC 50 (P) (16) (↑↑) *Refresh*	Object devices	: [D] .....X or Y assigned "0" in its lowest digit place : X0, X10, X20, etc. n.....K or H assigned with a number of multiple of 8.
	Number of steps	: REF, REF(P) ..... 5 steps

As described before, FX Series Programmable Controllers employ the input/output batch refresh method in which the data at the input terminals are transferred to the input image memory before beginning execution at step 0. Data is output to the output terminals via the latch memory from the output image memory after the execution of the END instruction.

This instruction is used either to read the latest input information while an operation is being executed or to output the result of the operation immediately after the completion of the operation.

Note:
- *The first destination device number to be designated must be a multiple of "10", such as X0 or Y10. The number of points to be refreshed is a multiple of "8". Designation of numbers other than these causes an error.*
- *There may be cases that a REF instruction is required in a step in a FOR and NEXT loop and a step between a label (lower number) and a CJ instruction (higher number) loop.*

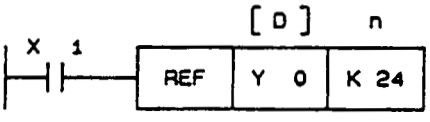
(1) INPUT REFRESH



- Only 8 points X10 to X17 are refreshed.

  If X10 to X17 have been ON approximately 10 msec (response delay time of filter) before this instruction is executed, image memory X10 to X17 is turned ON when the REF instruction is executed.

(2) OUTPUT REFRESH



- 24 points (Y0 to Y7, Y10 to Y17, and Y20 to Y27) are refreshed.

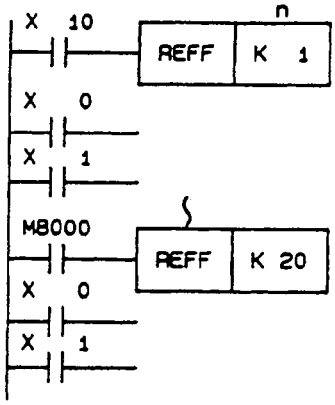
  If any point from Y0 to Y27 is ON, the corresponding output of the output latch memory is turned ON when the REF instruction is executed.

  The output contact is actuated after the output relay response time elapses.

HIGH-SPEED PROCESSING (FNC50 TO FNC 59)

5.6.2 REFRESH AND FILTER ADJUST

REFF FNC 51 (P) (16) *Refresh and Filter Adjust*	Object devices : K, H [X 0~X 7] (designation not necessary) n n=0~60 (filter constant : msec) Number of steps : REFF, REFF(P) ..... 3 steps



The image memory of input X0 to X7 is refreshed, taking the input filter constant as 1 msec when X10 is ON. The input filter constant is taken as 10 msec until this instruction is executed.

For the steps which follow this step, the input filter constant is set to 20 msec.

- Generally, PCs are provided with a C-R filter of approximately 10 msec filtering time for inputs as a measure for input contact chattering and noise.

  However, this filter is a disadvantage for high-speed data input from electronic solid-state switches which do not inherit chattering noise.
- FX Series PCs are provided with digital filters for inputs X0 to X7. Filtering time can be changed by a REFF instruction in the range of 0 to 60 msec.

  However, these inputs are also provided with a minimum C-R filter, which cannot be set to less than 50 μsec.
- An REFF instruction is executed in each operation cycle when X10 is ON. However, an REFF(P) instruction is executed only when X10 goes from ON to OFF.

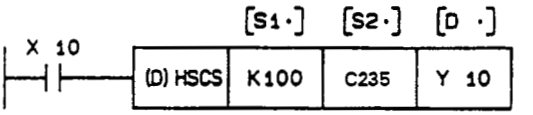
  This instruction is not executed when X10 is OFF and the filtering time of input filters for X0 through X10 is 10 msec. (values for input processing).

Ref.: • *The filtering time of input filters for these instructions are automatically set to 50 μsec if an interruption pointer is used, if X0 to X7 is used for high-speed counters or if an FNC56 instruction is used.*

HIGH-SPEED PROCESSING (FNC 50 TO FNC 59)

5.6.3 SET BY HIGH SPEED COUNTER

HSCS FNC 53 (16/32) *Set by High Speed Counter*	Object devices : [S1·] K, H | KnX | KnY | KnM | KnS | T | C | D | V, Z [S2·] : C235 to C255 (high-speed counter) X | Y | M | S [D ·] Number of steps : (D)HSCS..... 13 steps



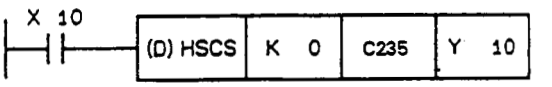
- Y10 is set immediately when the current value of C235 changes, for example, from 99 to 100 or 101 to 100.
- A high speed counter counts the number of status changes from ON to OFF of the input corresponding to the counter number in the interrupt mode as shown in Section 4.7.2 (2).

  The counter output contact is activated immediately when the counter current value reaches the preset value.
- Using an FNC53 instruction allows both setting and output to be executed in the interrupt mode.
- The higher two digits of object output devices should be the same (FNC53, FNC54).

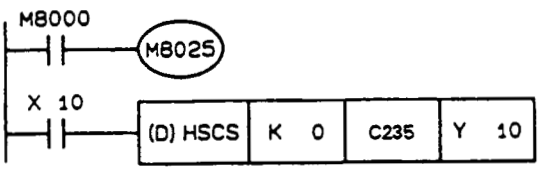
POINT

- The instructions FNC53, FNC54 are executed by the interrupt method when a pulse arrives at the input terminal.

  Therefore, the output contact Y10 is not activated if there is no input pulse even when X10 is turned ON and the conditions of the comparison for which Y10 is set is true.



- External Reset Mode Flag:

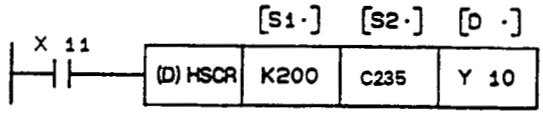


  If the flag M8025 is driven ON, all the related high speed compare instructions are executed when the reset input of the counter is turned ON. This will solve the above mentioned problem in the initial condition where a reset is always necessary.
- M8025 applies to the functions FNC53 and FNC54.

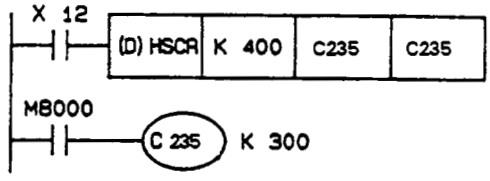
HIGH-SPEED PROCESSING (FNC 50 TO FNC 59)

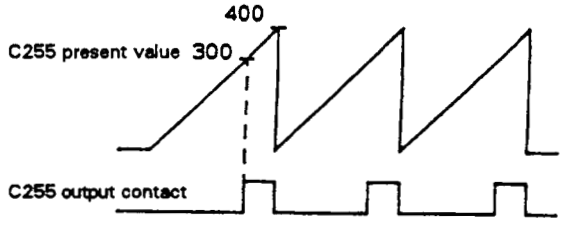
5.6.4 RESET BY HIGH SPEED COUNTER

HSCR FNC 54 (16/32) *Reset by High Speed Counter*	Object devices : [S1·] K, H | KnX | KnY | KnM | KnS | T | C | D | V, Z [S2·] : C235 to C255 (high-speed counter) X | Y | M | S [D ·] For [D·], designation of the same high-speed counter as designated for [S·] is possible. Number of steps : (D)HSCR ..... 13 steps



- Y10 is reset immediately when the current value of C235 changes, for example, from 199 to 200 or 201 to 200.



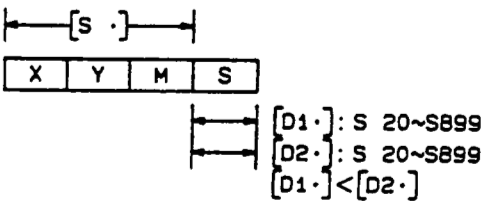


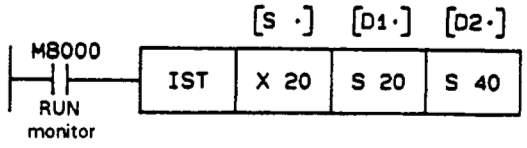
- C235 is reset immediately when the current value of C235 reaches 400. The current value is reset to 0 and the output contact is reset to the non-activated state.

5. APPLIED INSTRUCTIONS

5.7 HANDY INSTRUCTIONS (FNC 60 TO FNC 69)

5.7.1 INITIAL STATE

IST FNC 60 (16) *Initial State*	Object devices :  Number of steps : IST..... 7 steps



An instruction to automatically control the initial states and special auxiliary relays in a step ladder.

[S·] : Designate the head input number of the operation mode input.

With the above example :

X20 : Manual operation
X21 : Zero return
X22 : Step operation
X23 : One-cycle operation
X24 : Cycle operation
X25 : Zero return start
X26 : Automatic operation start
X27 : Stop

[D1·] : Designate the lowest state number actually used in the automatic operation.

[D2·] : Designate the highest state number actually used in the automatic operation.

- When this instruction is turned ON, the following devices are automatically controlled. If the instruction drive unit goes OFF, the state does not change.

  M8040 : Transfer inhibit
  M8041 : Transfer start
  M8042 : Start pulse
  M8047 : STL monitor enable
  S0 : Manual operation initial state
  S1 : Zero return initial state
  S2 : Automatic operation initial state

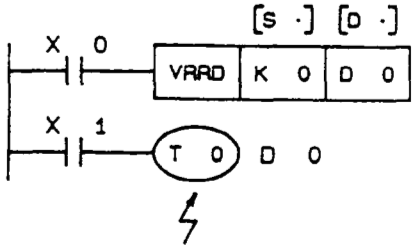
- An IST instruction can only be used once.
- For details concerning how to use this instruction, refer to Section 3.4.1.
- States S10 through to S19 (used for zero return) can used as general-purpose states when the IST instruction is not used.

  In this case, however, it is still necessary to use S0 through to S9 for the purpose of initial states but the actual usage of S0 to S2 is free.
- The IST instruction must be programmed prior to a series of STL circuit such as states S0 to S2.
- For inputs X20 to X24, a rotary switch must be used so that any two of these inputs will not be turned ON at the same time.
- If an operation mode is changed between X20 (manual), X21 (zero return), X22, X23, and X24 (automatic) while the zero return complete flag (M8043) is not set, all outputs will be turned OFF.

5.8 EXTERNAL FX SERIAL DEVICES (FNC 80 TO FNC 89)

5.8.1 VOLUME READ

VRRD FNC 85 (P) (16) *Volume Read* *FX-8AV*	Object devices : [D ·] K, H | KnX | KnY | KnM | KnS | T | C | D | V, Z [S ·] : Volume number (0 to 7) Number of steps : VRRD, VRRD(P) ..... 5 steps



Example using an analog timer

Analog value read from volume number 0 is converted into 8-bit binary data and transmitted to D0.

As an application example, data in D0 can be used as a timer setting.

This circuit provides an analog timer.

5. APPLIED INSTRUCTIONS

5.9 LIST OF INSTRUCTION MNEMONICS

Symbol	FNC No.	Section
ADD	20	5.4.1
BCD	18	5.3.9
BIN	19	5.3.10
CALL	01	5.2.2
CJ	00	5.2.1
CMP	10	5.3.1
DEC	25	5.4.5
DECO	41	5.6.2
DI	05	5.2.3
DIV	23	5.4.4
EI	04	5.2.3
ENCO	42	5.6.3
FEND	06	5.2.4
FOR	08	5.2.6
HSCR	54	5.7.5

Symbol	FNC No.	Section
HSCS	53	5.7.4
INC	24	5.4.5
IRET	03	5.2.3
IST	60	5.8.1
MOV	12	5.3.3
MUL	22	5.4.3
NEXT	09	5.2.4
REF	50	5.7.1
REFF	51	5.7.2

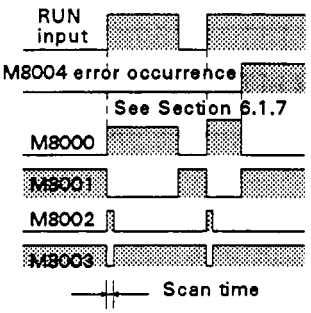
Symbol	FNC No.	Section
SFTL	35	5.5.3
SFTR	34	5.5.3
SUB	21	5.4.2
SUM	43	5.6.4
WAND	26	5.4.6
WDT	07	5.2.5
WOR	27	5.4.6
WXOR	28	5.4.6
ZCP	11	5.3.2
ZRST	40	5.6.1

1	GENERAL HARDWARE AND INSTALLATION
2	BASIC SEQUENCE INSTRUCTIONS
3	STEP LADDER INSTRUCTIONS
4	DEVICE DETAILS
5	APPLIED INSTRUCTION
6	SPECIAL DEVICE AND INSTRUCTION LISTS
7	USING SPECIAL UNITS EFFECTIVELY
	APPENDIX

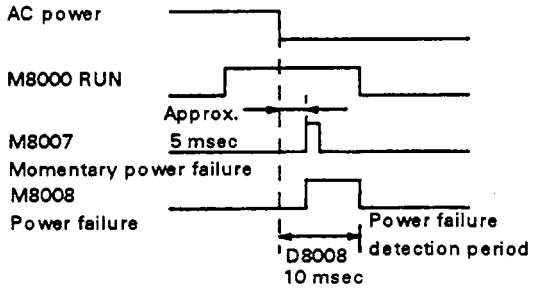
6. SPECIAL DEVICE AND INSTRUCTION LISTS

6.1 SPECIAL DEVICES

6.1.1 PC STATUS (M8000 TO M8009, D8000 TO D8009)

Number/Name	Operation/Function
M8000 (§) RUN monitor NO contact	
M8001 (§) RUN monitor NC contact	
M8002 (§) Initial pulse NO contact	
M8003 (§) Initial pulse NC contact	
M8004 (§) Error occurrence	ON when M8060 and/or M8067 is ON.
M8005 (§) Low battery voltage	ON when battery voltage is abnormally low.
M8006 (§) Low battery voltage latch	ON state retained after low battery voltage is detected.
M8007 (§) Momentary power failure detected	See the diagram below. The PC will continue operating if M8007 is ON for less time than D8008.
M8008 (§) Power failure	See the diagram below. Reset if turned from ON to OFF.
M8009 (§) 24V DC OFF	Set if there is a shut-down of circuits in the extension's 24V DC supply.

Number/Name	Register Data
D8000 Watchdog timer	Initial setting : 100 msec (Initial ROM data is set when the power to the PC is turned ON.) Can be changed in increments of 1 msec.
D8001 (§) PC time and system version	
D8002 (§) Memory capacity	0002 ..... 2K steps
D8003 (§) Memory type	RAM/EPROM/EEPROM Built-in/cassette Memory protect switch ON/OFF
D8004 (§) Error M number	 8060 to 8068 (M8004 ON)
D8005 (§) Battery voltage	 Current battery voltage (Example: 3.6 V)
D8006 (§) Low battery detection level	Initial value : 3.0 V (System ROM data is used when the power to the PC is turned ON.)
D8007 (§) Momentary power failure occurrence times	Stores the number of times M8007 is set ON. The stored data is cleared when the power is turned OFF.
D8008 (§) Power failure detection period	Initial setting : 10 msec (in increments of 1 msec) (System ROM data is used when the power to the PC is turned ON.) See the diagram below.
D8009 (§) 24V DC OFF unit number	The lowest input device number of a unitor block affected by the shut-down of the 24V DC is stored.



Ref.:
- *Do not drive devices marked with "(§)" by user's programs.*
- *D data is expressed as a decimal when monitored unless otherwise indicated.*

- When the power supply used is an AC 200 V system, the power down detect period determined by the value in D8008 can be altered by the program in the range of 10 to 100 msec.

6. SPECIAL DEVICE AND INSTRUCTION LISTS

SPECIAL DEVICES

6.1.2 CLOCK (M8010 TO M8019, D8010 TO D8019)

Number/Name	Operation/Function
M8010	
M8011 (§) 10 msec clock	Oscillates in 10 msec cycles.
M8012 (§) 100 msec clock	Oscillates in 100 msec cycles.
M8013 (§) 1 sec clock	Oscillates in 1 sec cycles.
M8014 (§) 1 min clock	Oscillates in 1 min cycles.
M8015	
M8016	
M8017	
M8018	
M8019	

Number/Name	Register Data
D8010 (§) Present scan data	Current operation cycle time (in units of 0.1 msec)
D8011 (§) Minimum scan time	Minimum scan time (in units of 0.1 msec)
D8012 (§) Maximum scan time	Maximum scan time* (in units of 0.1 msec)
D8013	
D8014	
D8015	
D8016	
D8017	
D8018	
D8019	

*: Waiting time for constant scan when M3089 is set ON is not included.

6.1.3 FLAG (M8020 TO M8029, D8020 TO D8029)

Number/Name	Operation/Function
M8020 (§) Zero	Set when the result of addition/subtraction is "0".
M8021 (§) Borrow	Set when the result of subtraction is smaller than the minimum negative value.
M8022 Carry	Set when "carry" occurs in an addition or when overflow occurs as a result of a shift.
M8023	
M8024	
M8025	External reset HSC mode
M8028	

Number/Name	Register Data
D8020	
D8021	
D8022	
D8023	
D8024	
D8025	
D8026	
D8027	
D8028 (§)	Z register data
D8029 (§)	V register data

§ :Do not drive device marked with "(§)" by user's programs.

SPECIAL DEVICES

6.1.4 PC MODE (M8030 TO M8039, D8030 TO D8039)

Number/Name	Operation/Function	Number/Name	Register Data
M8030 Battery LED OFF	The LED on the front panel of the PC is not lit even if battery voltage is lowered below the check level when M8030 is ON.	D8030	
M8031 Non-latch memory all clear	The ON/OFF image memory of Y, M, S, T, and C, and the current values of T, C, and D are cleared to "0" when states M8031 and M8032 are turned ON. For special registers for which defaults are set, these values are transferred from the system ROM. Special registers and file register will not be affected.	D8031	
M8032 Latch memory all clear		D8032	
M8033 Memory hold stop	The contents of the image memory and data memory are retained when the PC operation status is changed from RUN to STOP.	D8033	
M8034 All output disable	Although the external output contacts of PC are all turned OFF, the PC continues running using image memory.	D8034	
M8035 *1 Forced RUN mode	By using M8035, M8036 and M8037 it is possible to control the PC RUN/STOP status using two input signals (RUN input pushbutton switch and Xoo (STOP input pushbutton switch)). *2	D8035	
M8036 *1 Forced RUN signal		D8036	
M8037 *1 Forced STOP signal		D8037	
M8038		D8038	

• 1 : M relays marked with an asterisk are turned OFF when the PC operation status is changed from RUN to STOP.

• 2 : The PC is set to RUN when M8035 or M8036 is forcibly turned ON by the programming panel even if the RUN input is OFF.

The PC stops running if M8037 is forcibly set OFF while the PC is running.

6. SPECIAL DEVICE AND INSTRUCTION LISTS

SPECIAL DEVICES

6.1.5 STEP LADDER (M8040 TO M8049, D8040 TO D8049)

Number/Name	Operation/Function	Number/Name	Register Data
M8040 Transfer disable	State transfer is disabled when M8040 is set ON.	D8040 (§) ON state number 1	The lowest operated state number (states S0 to S899) is stored in D8040. Other active state numbers are stored in ascending order (up to 8 points).
M8041* Transfer start	State transfer from the initial state is enabled during automatic operation.	D8041 (§) ON state number 2	
M8042 Start pulse*	A pulse output given in response to a start input.	D8042 (§) ON state number 3	
M8043* Zero return complete	ON at the END state of ZERO RETURN mode	D8043 (§) ON state number 4	
M8044* Zero point condition	ON when machine zero is detected.	D8044 (§) ON state number 5	
M8045 All output reset disable	Disables the "all output reset" function when the mode is changed.	D8045 (§) ON state number 6	
M8046 (§) STL state set ON	ON if any one of S0 to S899 is ON while M8047 is ON.	D8046 (§) ON state number 7	
M8047 STL monitor effective	D8040 to D8047 are effective when M8047 is ON.	D8047 (§) ON state number 8	
M8048 (§) Annunciator ON	ON if any one of S900 to S999 is ON while M8049 is ON.	D8048	
M8049 Annunciator enable	D8049 operation is effective when M8049 is ON.	D8049 (§) Minimum ON state number	Stores the lowest annunciator state number (S900 to S999) that is ON.

*: M relays marked with an asterisk are turned OFF when the PC operation status is changed from RUN to STOP.
All registers concerning STL states are updated when the END instruction is executed.

6.1.6 INTERRUPTION DISABLED (M8050 TO M8059, D8050 TO D8059)

Number/Name	Operation/Function	Number/Name	Register Data
M8050 I0XX disable	When interrupt mode has been enabled by FNC4 (EI) instruction, it can be disabled for specific interrupt numbers according to the M numbers set. For example, I0XX interruption is disabled when M8050 is set ON.	D8050	
M8051 I1XX disable		D8051	
M8052 I2XX disable		D8052	
M8053 I2XX disable		D8053	
M8054		D8054	
M8055		D8055	
M8056		D8056	
M8057		D8057	
M8058		D8058	
M8059		D8059	

§: Do not drive devices marked with "(§)" by user's programs.

SPECIAL DEVICES

6.1.7 ERROR DETECTION (M8060 TO M8069, D8060 TO D8069)

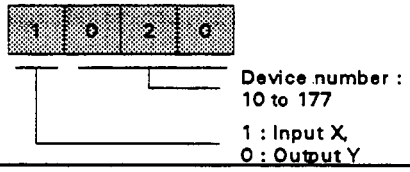
If any one of M8060 to M8067 is ON, the lowest number is stored to D8004 and M8004 is turned ON.

Number	Name	PROG. E LED	PC Status	Number	Register Data	
M8060 (§)	I/O configuration error	OFF	RUN	D8060 (§)	The first I/O number of the unit/block causing the I/O configuration error *1	
M8061 (§)	PC hardware error	Flash	STOP	D8061 (§)	Error code number of the PC hardware error	See Error Code Table
M8062 (§)	PC/PP communication error	OFF	RUN	D8062 (§)	Error code number of the PC/PP communication error	
M8064 (§)	Parameter error	Flash	STOP	D8064 (§)	Error code number of the parameter error	
M8065 (§)	Syntax error	Flash	STOP	D8065 (§)	Error code number of the syntax error	
M8066 (§)	Circuit error	Flash	STOP	D8066 (§)	Error code number of the circuit error	
M8067 (§) *3	Operation error	OFF	RUN	D8067 (§) *3	Error code number of the operation error	
M8068	Operation error latch	OFF	RUN	D8068	Step number (latched) of the operation error	
M8069	I/O bus check *2	–	–	D8069 (§) *3	Step number of errors M8065 to M8067	

§: Do not drive devices marked with "(§)" by user's programs.

* 1 : If the unit or block corresponding to a programmed I/O number is not actually loaded, M8060 is set ON and the first device number is written to D8060.

Example : X20 is not loaded



* 2 : An I/O bus check is executed when M8069 is turned ON. If an error occurs, error code 6103 is written and M8061 is turned ON.

* 3 : These are OFF when the PC operating status is changed from STOP to RUN.

TIMING OF ERROR CHECKS

Error Item		Power is Turned ON	Initial STOP to RUN Setting after Power ON	Detection
M8060	I/O configuration error	Checked	Checked	While in RUN
M8061	PC hardware error	Checked	–	While in RUN
M8062	PC/PP communication error	–	–	When a signal from PP is received.
		–	–	
M8064 M8065 M8066	Parameter error Syntax error Circuit error	Checked	Checked	When program is changed (STOP) When program is transferred (STOP)
M8067 M8068	Operation Error operation Error Latch	–	–	When in the RUN state

Note : • "PP" = Programming panel.

SPECIAL DEVICES

6.1.8 ERROR CODES

ERROR CODE 1

Error Classification	Error Code	Error Contents	Remark/Remedy
D8061 PC hardware error	0000	No error	Check whether the extension cables are connected properly.
	6101	RAM error	
	6102	Operation circuit error	
	6103	I/O bus error (when M8069 is ON)	
D8062 PC/PP communication error	0000	No error	Check whether the programming panel (PP) and the programmable controller (PC) are connected properly.
	6201	Parity error Overrun error Framing error	
	6202	Communication character error	
	6203	Communication data sum check error	
	6204	Data format error	
	6205	Command error	
D8064 Parameter error	0000	No error	Stop the PC, select the parameter mode, and set the correct data.
	6401	Program sum check error	
	6402	Memory capacity setting error	
	6403	Keep area setting error	
	6404	Comment area setting error	
	6405	File register are setting error	
	6409	Other setting error	
D8065 Syntax error	0000	No error	During programming, each instruction is checked. If a syntax error is detected, correct the instruction in the programming mode.
	6501	Incorrect instruction/device symbol/device number combination.	
	6502	No OUTT or OUTC prior to setting value.	
	6503	1) No setting value following OUTT or OUTC. 2) Insufficient number of operands for an applied instruction.	
	6504	1) The same label number is used more than once. 2) The same interrupt input or high-speed counter input is used more than once.	
	6505	Device number is outside the allowable range.	
	6509	Other error	

SPECIAL DEVICES

ERROR CODE 2

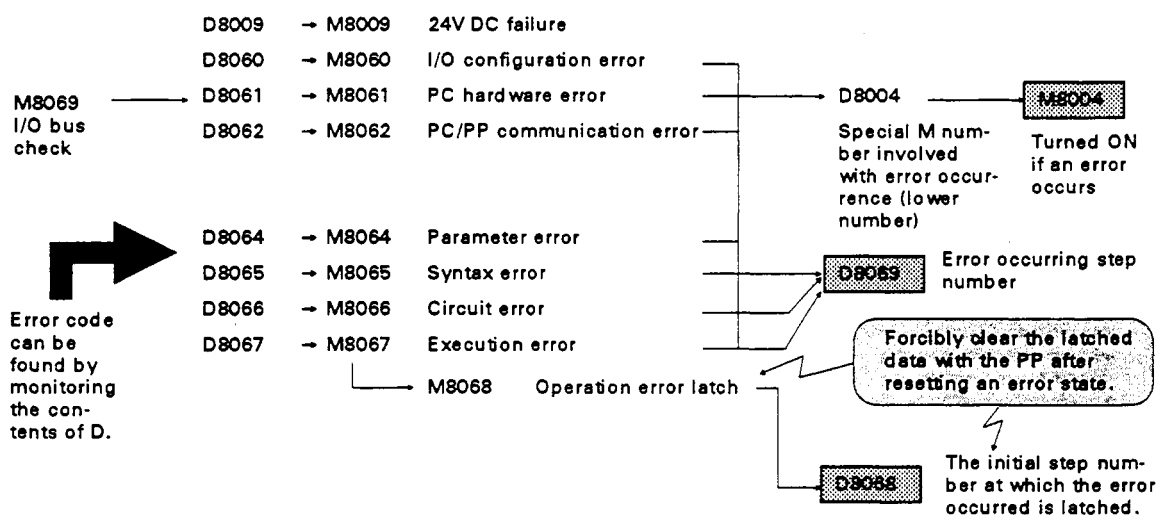
Error Classification	Error Code	Error Contents	Reset/Remedy
D8066 Circuit error	0000	No error	A circuit error occurs if a combination of instructions is incorrect when the entire circuit block is checked or if the relationship between paired instructions is incorrect. Select programming mode and correct the instructions.
	6601	LD or LDI is used 9 or more times continuously.	
	6602	1) No LD or LDI instruction. Illegal use of LD/LDI and ANB/ORB. 2) The following instructions are not connected to the bus line : STL, RET, MCR, P (pointer), I (interrupt), EI, DI, SRET, IRET, FOR, NEXT, FEND, and END 3) When MPP is missing.	
	6603	MPS is used 12 or more times continuously.	
	6604	Illegal use of MPS/MRD and MPP.	
	6605	1) STL is used 9 or more times continuously. 2) MC, MCR, or I (interrupt) is designated within STL. 3) RET is designated outside STL or not designated at all.	
	6606	1) No P (pointer) or I (interrupt) 2) No SRET or IRET. 3) I (interrupt), SRET, or IRET is designated within a main program. 4) STL, RET, MC, or MCR is designated in a subroutine or interrupt routine.	
	6607	1) Illegal FOR-NEXT relationship. 6 or more level are nested. 2) The following instructions are designated in the FOR-NEXT loop : STL, RET, MC, MCR, IRET, SRET, FEND, and END	
	6608	1) Illegal MC-MCR relationship. 2) No MCR NO. 3) SRET, IRET, or I (interrupt) is designated between MC and MCR blocks.	
	6609	Other	

SPECIAL DEVICES

ERROR CODE 3

Error Classification	Error Code	Error Contents	Remark/Remedy
D8067 Execution error	0000	No error	These errors occur during the execution of an operation. When an operation error occurs, stop the PC, change the mode to programming, and correct the program. An execution error occurs even when a syntax or circuit error does not occur. An example of this is given below. Example : D500Z does not cause an error. However, if value Z, obtained as the result of operation, is 100 (Z=100), D500Z becomes D600 exceeding the allowable number range; this will cause an error.
	6701	1) No jump destination for CJ or CALL. 2) A label is designated in a block that comes after the END instruction. 3) An independent label is designated within the FOR-NEXT loop or in a subroutine.	
	6702	6 or more CALL levels are nested.	
	-		
	-		
	6705	A device is used for an applied instruction operand that is not the object device for that applied instruction.	
	6706	The device range or data range designated as an applied instruction operand is outside the allowable range.	
	6707	A file register is accessed without setting a file register.	
	-		
	6709	Other (no IRET or SRET, illegal relationship between FOR-NEXT, etc.)	

RELATIONSHIPS OF ERROR RELATED SPECIAL M'S AND SPECIAL D'S



6.1.9 HIGH-SPEED COUNTERS (M8235 TO M8255, D8235 TO D8255)

Number	Function
M8235 M8236 M8237 M8238 M8239 M8240 M8241 M8242 M8243 M8244 M8245	When M8□□□ is operated, C□□□, is, a 1-phase high-speed counter functions as a down counter. When M8□□□ is not operated, the corresponding counter functions as an up counter.
M8246 (§) M8247 (§) M8248 (§) M8249 (§) M8250 (§) M8251 (§) M8252 (§) M8253 (§) M8254 (§) M8255 (§)	When C□□□,a 1-phase 2-input counter or a 2-phase counter, functions as a down counter, M8□□□, operates. M8□□□, turns off in the up counter mode.

6. SPECIAL DEVICE AND INSTRUCTION LISTS

6.2 INSTRUCTION AND EXECUTION TIME LIST

6.2.1 SEQUENCE AND STEP LADDER INSTRUCTION

Type	Instruction	Objective Device	Number of Steps	Execution Time (μsec) Turning ON Time	Turning OFF Time
Contact Instruction	LD	X, Y, M, S, T, C, Special M	1	1	
	LDI	The same as above	1	1	
	AND	The same as above	1	1	
	ANI	The same as above	1	1	
	OR	The same as above	1	1	
	ORI	The same as above	1	1	
Join Instruction	ANB	None	1	1	
	ORB	None	1	1	
	MPS	None	1	1	
	MRD	None	1	1	
	MPP	None	1	1	
Other Instructions	MC	N-Y, M	3	135	157
	MCR	N (Nesting)	2	82	
	NOP	None	1	1	
	END	None	1	1200	
Step Ladder Instruction	STL	S	1	114+104n *1	
	RET	None	1	118	

Type	Instruction	Objective Device	Number of Steps	Execution Time (μsec) Turning ON Time	Turning OFF Time
Output Instruction	OUT	Y, M	1	1	
		S	2	124	124
		Special M	2	134	132
		T-K, D	3	134 *3	156
		C-K, D (16 bits)	3	228 *3	103
		C-K, D (32 bits)	5	240 *3	232
	SET	Y, M	1	1	
		S	2	140 *2	68
		Special M	2	143	76
	RST	Y, M	1	1	
		S	2	119	68
		Special M	2	143	77
		T, C	2	195	139
		D, V, Z, Special D	3	90	68
	PLS	Y, M	2	145	156
	PLF	Y, M	2	157	145

Label	P	0 to 63	1	1
	I	0 ** to 2**	1	1

- 1 : "n" indicates the number of continuous STL instructions (number of parallel/merge instructions).
- 2 : In the STL circuit block, 152+54n(s) for turning ON and 68(s) for turning OFF. "n" indicates the number of continuous STL instructions (number of parallel/merge instructions).
- 3 : Execution time is 45msec longer when an in direct designation (T-D, C-D) is used. After counting up (time or count), the turning ON time becomes the same as the turning OFF time.

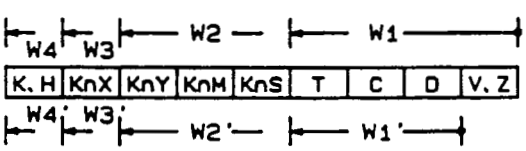
INSTRUCTION AND EXECUTION TIME LIST

6.2.2 PROGRAM FLOW

(D) / (P)	Instruction and Configuration	Execution Time (μs) Turning ON Time	Execution Time (μs) Turning OFF Time	Remark
N/A / Available	FNC 00 CJ [S·] P 0~P 63 CONDITIONAL JUMP	143	75	ditional jump P63 is equivalent to END
N/A / Available	FNC 01 CALL [S·] P 0~P 62 SUBROUTINE CALL	155	75	ubroutine call Nesting level : Up to 5 levels
N/A / N/A	FNC 02 SRET SUBROUTINE RETURN	104		Subroutine return To be programmed after FEND
N/A / N/A	FNC 03 IRET INTERRUPT RETURRN	122		Interrupt return To be programmed after FEND
N/A / N/A	FNC 04 EI INTERRUPT ENABLE	136		Interrupt enabled
N/A / N/A	FNC 05 DI INTERRUPT DISABLE	67		Interrupt disable
N/A / N/A	FNC 06 FEND FIRST END	1200		Main program end
N/A / Available	FNC 07 WDT WATCH DOG TIMER	136	68	Watchdog timer refresh
N/A / N/A	FNC 08 FOR [S·] W4 FOR	133		Start of repeat range Nesting level : Up to 5 levels
N/A / N/A	FNC 09 NEXT NEXT	85		End of repeat range

These instructions are driven directly without using any contacts.

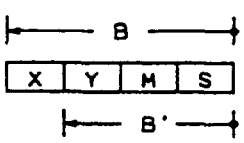
N/A : Not Available.

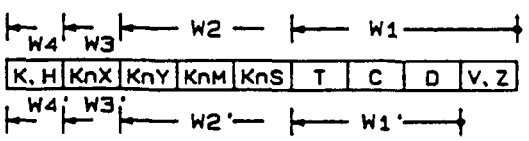


INSTRUCTION AND EXECUTION TIME LIST

6.2.3 MOVE, COMPARE, AND OTHER INSTRUCTIONS

(D) (P)	Instruction and Configuration		Execution Time (μs) Turning ON Time	Execution Time (μs) Turning OFF Time	Remark
Available	FNC 10 CMP [S1·] W4 [S2·] W4 [D·] B	(16)	674	104	omparison [S1·] ≶ [S2·] → [D·]
Available	COMPARE	(32)	788	111	
Available	FNC 11 ZCP [S1·] W4 [S2·] W4 [S·] W4 [D·] B	(16)	901	104	Zone comparison [S1·] to [S2·] ≶ [S·] → [D·]
Available	ZONE COMPARE	(32)	1040	121	
Available	FNC 12 MOV [S·] W4 [D·] W2	(16)	315	93	Move [S·] → [D·]
Available	MOVE	(32)	404	111	
N/A	FNC 13 SMOV [S·] W3 [m1] K, H [m2] K, H [D·] W2 [n] K, H		302.9	33.3	Shift m2 digits from the m1 position are moved to the nth place. m1, m2, n = 1 to 4
Available	SHIFT MOVE				
Available	FNC 18 BCD [S·] W3 [D·] W2	(16)	1490	96	BCD conversion [S·] → [D·]
Available	BINARY CODED TO DECIMAL	(32)	3070	113	BIN 16/32 bits BCD 4/8 digits, positive number
Available	FNC 19 BIN [S·] W3 [D·] W2	(16)	1030	96	BIN conversion [S·] → [D·]
Available	BINARY	(32)	2440	113	BCD 4/8 digits, positive BIN 16/32 bits





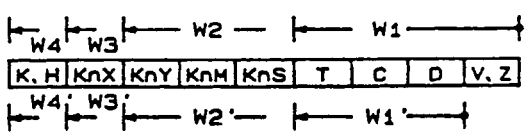
INSTRUCTION AND EXECUTION TIME LIST

6.2.4 ARITHMETICAL AND LOGICAL OPERATIONS

Z : M8020 Br : M8021 CY : M8022 F : M8029

(D) (P)	Instruction and Configuration		Execution Time (μs) Turning ON Time		Turning OFF time	Remark
Available	FNC 20 ADD [S1·] W4 [S2·] W4 [D·] W2	(16)	496		104	BIN addition [S1·] + [S2·] → [D·]
Available	ADDITION BINARY	(32)	623		111	
Available	FNC 21 SUB [S1·] W4 [S2·] W4 [D·] W2	(16)	500		104	BIN subtraction [S1·] – [S2·] → [D·]
Available	SUBTRACTION BINARY	(32)	628		111	
Available	FNC 22 MUL [S1·] W4 [S2·] W4 [D·] W2	(16)	650		106	BIN multiplication [S1·] x [S2·] →[D·] +1 [D·]
Available	MULTIPLICATION BINARY	(32)	1380		113	
Available	FNC 23 DIV [S1·] W4 [S2·] W4 [D·] W2	(16)	1500		106	BIN division [S1·]÷[S2·]→[D·]...[D·]+1 Dividend Quotient Divisor Remainder
Available	DIVISION BINARY	(32)	2710		113	
Available	FNC 24 INC [D·] W2	(16)	220		96	BIN increment [D·] + 1 → [D·]
Available	INCREMENT BINARY	(32)	284		103	
Available	FNC 25 DEC [D·] W2	(16)	218		104	BIN decrement [D·] – 1 → [D·]
Available	DECREMENT BINARY	(32)	282		111	
Available	FNC 26 WAND [S1·] W4 [S2·] W4 [D·] W2	(16)	254		104	ogical AND [S1·] ∧ [S2·] → [D·]
Available	WORD AND	(32)	575		111	
Available	FNC 27 WOR [S1·] W4 [S2·] W4 [D·] W2	(16)	454		104	ogical OR [S1·] ∨ [S2·] → [D·]
Available	WORD OR	(32)	575		111	
Available	FNC 28 WXOR [S1·] W4 [S2·] W4 [D·] W2	(16)	452		104	xclusive OR [S1·] ∀ [S2·] → [D·]
Available	WORD EXCLUSIVE OR	(32)	573		111	

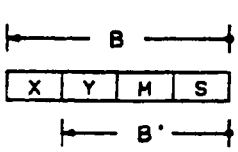
The destination varies in each operation cycle if a continuous execution instruction is used. Take note of this when using these instructions.

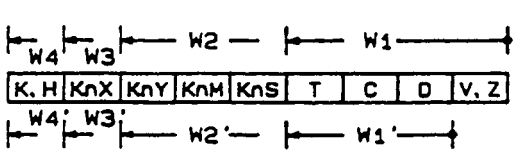


INSTRUCTION AND EXECUTION TIME LIST

6.2.5 ROTATION SHIFT

(D) / (P)	Instruction and Configuration	Execution Time (μs) Turning ON Time	Execution Time (μs) Turning OFF Time	Remark
N/A / Available	FNC 34 SFTR [S·] B, [D·] B', [n1] K,H, [n2] K,H SHIFT RIGHT	n2=4 819 +153n1	91	Bit n2 right shift instruction for bit n1 [D·]. [S·] is the n2 bit device. n2≤n1≤1024
N/A / Available	FNC 35 SFTL [S·] B, [D·] B', [n1] K,H, [n2] K,H SHIFT LEFT	n2 = 4 802 +153n1	89	Bit n2 left shift instruction for bit n1 [D·]. [S·] is the n2 bit device. n2≤n1≤1024

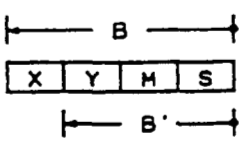


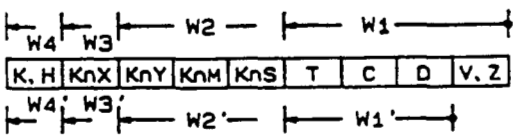


INSTRUCTION AND EXECUTION TIME LIST

6.2.6 DATA OPERATIONS

(D) (P)	Instruction and Configuration	Execution Time (μs) Turning ON Time	Execution Time (μs) Turning OFF Time	Remark
N/A Available	FNC 40 ZRST [D1·] W1 B · [D2·] W1 B · ZONE RESET	161.3 +K (D2–D1) When K=3.2D K=16.5 T, C, S K=13.5 Y, M	39.9	Block reset The range is reset for [D1·] to [D2·] (D1· ≤ D2·).
N/A Available	FNC 41 DECO [S·] B, K, H W1 [D·] B W1 [n] K, H DECODE	114.8	28.8	After decoding the BIN value of [S·] (n=1 to 8 bits), [D·] is driven.
N/A Available	FNC 42 ENCO [S·] B, W1 [D·] W1 [n] K, H ENCODE	125.6	28.8	After encoding [S·] into the BIN value (n=1 to 8 bits), store to [D·].





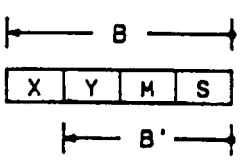
INSTRUCTION AND EXECUTION TIME LIST

6.2.7 HIGH-SPEED PROCESSING

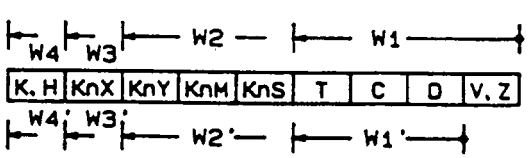
• Can only be used once.

(D) (P)	Instruction and Configuration	Execution Time (μs) Turning ON Time		Turning OFF Time		Remark
N/A Available	FNC 50 REF [D] X, Y [n] K, H REFRESH	145.3 + 3.6n		33.3		I/O refresh [D] : X 0, X10, ...Y 0, Y10, ... n =8, 16, 24, ...128
N/A Available	FNC 51 REFF [n] K, H REFRESH AND FILTER ADJUST	56.0 +4.9n		33.3		Filter adjust Refreshing X0 to X7 filters Filter constant n=0 to 60 msec
Available N/A	FNC 53 HSCS [S1·] W4 [S2·] C [D ·] B' SET BY HIGH SPEED COUNTER	(32)	809	113	See note below	Comparison set (high-speed counter) [D·] is set when [S1·]=[S2·]. Interrupt is output for Y.
Available N/A	FNC 54 HSCR [S1·] W4 [S2·] C [D ·] B', C RESET BY HIGH SPEED COUNTER	(32)	810	115		Comparison reset (high-speed counter) [D·] is reset when [S1·]=[S2·]. Interrupt is output for Y. If [D·] is "C", it resets itself.

* Repeated use is not allowed



Note: • *The combination of these instructions must not exceed 6 pts. when they are driven ON simultaneously.*

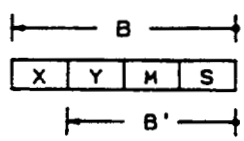


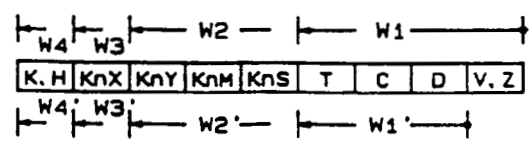
INSTRUCTION AND EXECUTION TIME LIST

6.2.8 HANDY INSTRUCTION

* Can only be used once.

(D) / (P)	Instruction and Configuration	Execution Time (μs) Turning ON Time	Execution Time (μs) Turning OFF Time	Remark
N/A N/A	FNC 60 IST [S·] X, Y, M [D1·] Si [D2·] Sj	946	100	itial state* [S·] : Mode designation input (8 points) [D1·] to [D2·] : Automatic operation state S20 ≤ Si < Sj ≤ S899

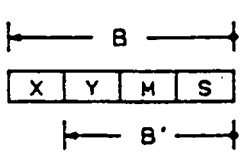


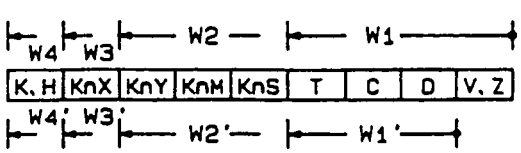


INSTRUCTION AND EXECUTION TIME LIST

6.2.9 EXTERNAL FX SERIAL DEVICES

(D) / (P)	Instruction and Configuration	Execution Time (µs) Turning ON Time	Execution Time (µs) Turning OFF Time	Remark
	FNC			
	FNC			
	FNC			
	FNC			
N/A / Available	FNC 85 VRRD [S·] K, H [D·] W2 VOLUME READ	371	91	Volume read The value of volume number [S·] =0 to 7 is read to [D·] in 8-bit binary.
	FNC			
	FNC			
	FNC			





INSTRUCTION AND EXECUTION TIME LIST

INSTRUCTION EXECUTION TIME (REFERENCE)

(1) CALCULATING OPERATION CYCLE TIME

(a) Basic operation cycle time :

The total execution time of all instructions from step 0 to the END or FEND instruction and I/O processing time. I/O processing time is 56 μsec x number of blocks; each block is in units of 8 I/O points.

(b) Execution cycle time :

If a high speed processing instruction is used in a program, the program execution cycle time is n times the basic operation cycle time (described above).

Example : High-speed counter

1-phase, 2 kHz, 1 point n = Approx. 1.4

1-phase, 2 kHz, 2 points n = Approx. 2.0

(2) TIME DIFFERENCE ACCORDING TO OPERANDS

The instruction execution time varies according to whether the object device is digit-designation device such as KnX, KnY, KnM, KnS, K, H, T, C, or D.

Instruction Execution Time When a MOV Instruction is Used :

[S] \ [D]	KnY, KnM, KnS	T, C, D
KnX, KnY KnM, KnS	602	403
K, H	469	269
T, C, D	514	[315]

Without index modifier

[S·] \ [D·]	KnY, KnM, KnS	T, C, D
KnX, KnY KnM, KnS	739	488
K, H	597	346
T, C, D	655	[404]

With index modifier

Instruction Execution Time When (D) MOV Instruction is Used :

[S] \ [D]	KnY, KnM, KnS	T, C, D
KnX, KnY KnM, KnS	684	487
K, H	530	332
T, C, D	600	[403]

Without index modifier

[S·] \ [D·]	KnY, KnM, KnS	T, C, D
KnX, KnY KnM, KnS	921	574
K, H	696	435
T, C, D	741	481

With index modifier

[ ] : Values listed.

Note:
- *If the designated device number of the bit device is not a multiple of 8, the instruction execution time will be slightly longer.*
- *The execution time of a pulse instruction is the same as the execution time of a continuous instruction.*

1	GENERAL HARDWARE AND INSTALLATION
2	BASIC SEQUENCE INSTRUCTIONS
3	STEP LADDER INSTRUCTIONS
4	DEVICE DETAILS
5	APPLIED INSTRUCTION
6	SPECIAL DEVICE AND INSTRUCTION LISTS
7	USING SPECIAL UNITS EFFECTIVELY
	APPENDIX

7. USING SPECIAL UNITS EFFECTIVELY

7.1 FX-232AW INTERFACE UNIT

7.1.1 FX-232AW INTERFACE UNIT SPECIFICATIONS

The FX-232AW interface unit converts the RS422 signals output from the PC into RS232C signals to allow communication between a PC and a personal computer.

Use this interface unit when creating a program with a personal computer. In addition to this, the following signals can be handled by the interface unit :

(1) SIGNALS

(a) ON/OFF status monitoring of devices such as X, Y, M, S, T, and C.

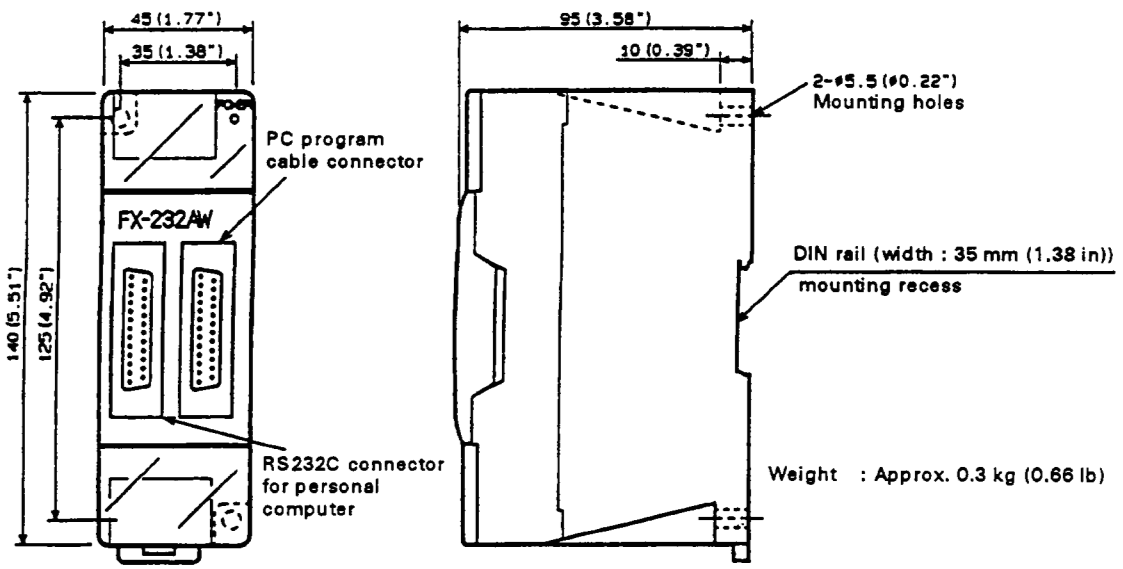
(b) Monitoring present value data of devices such as T, C, and D.

(c) Forcible on/off control of devices such as X, Y, M, S, and T.

(d) Rewriting present values of devices such as T, C, and D.

Signal transmission speed : 9600 bps

Required environment : Refer to the general environmental requirements described in Section 1.10.1.



7. USING SPECIAL UNITS EFFECTIVELY

7.2 FX-8AV VOLUME ADAPTER

7.2.1 VOLUME ADAPTER SPECIFICATIONS

The FX-8AV volume adapter is connected to the panel on the left side of the base unit.

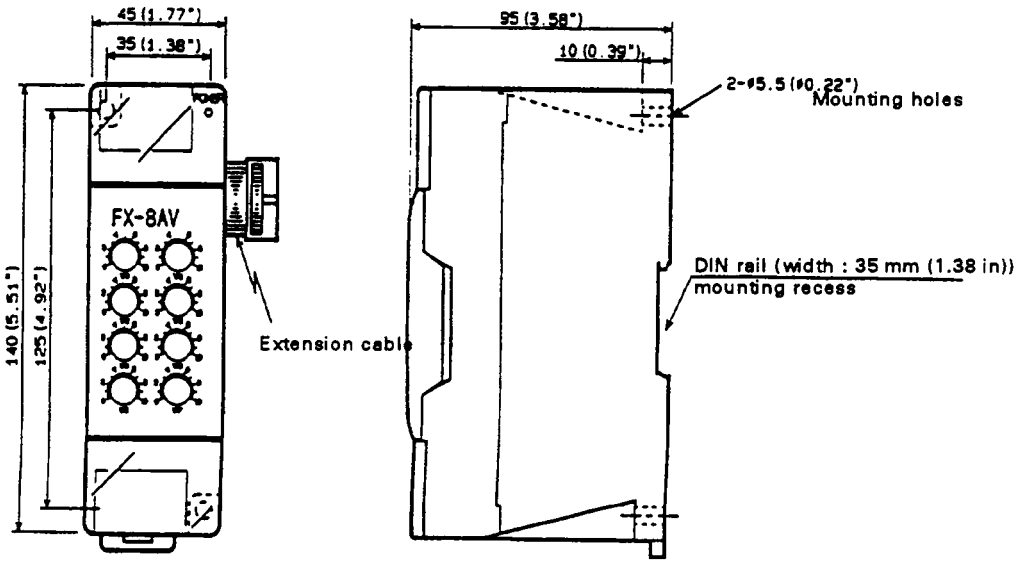
Eight miniature volumes are provided on this adapter. They can be adjusted with a screwdriver from the front of the adapter.

(1) USE

The adapter is generally used as an analog timer. It can also be used in place of a rotary switch.

(2) SPECIFICATIONS

See Section 1.10.1 for the general environmental requirements.



Weight : Approx. 0.3 kg (0.66 lb)

MEMO

1	GENERAL HARDWARE AND INSTALLATION
2	BASIC SEQUENCE INSTRUCTIONS
3	STEP LADDER INSTRUCTIONS
4	DEVICE DETAILS
5	APPLIED INSTRUCTION
6	SPECIAL DEVICE AND INSTRUCTION LISTS
7	USING SPECIAL UNITS EFFECTIVELY
	APPENDIX

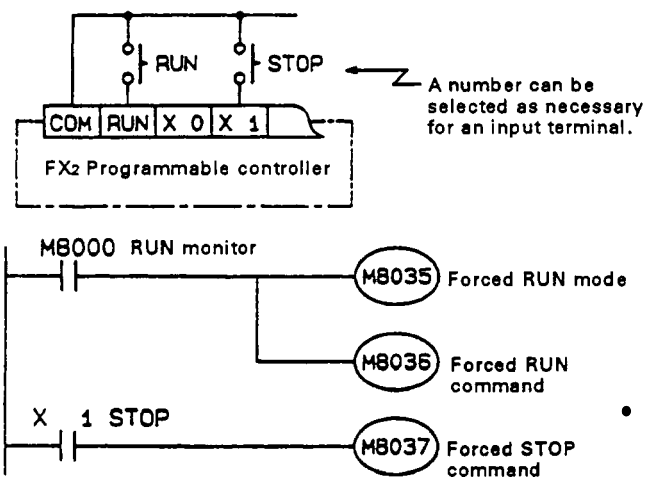
1. 2-INPUT RUN/STOP SWITCH CONTROL

The FX2 Series PC has one RUN input terminal. When this terminal is ON, the PC is in the RUN state and when it is OFF the PC is in the STOP state. This is the same as with the F1 Series PC.

An F_2 Series PC, in comparison to an FX2 Series PC, has both RUN and STOP input terminals. By driving these terminals momentarily with a pushbutton switch, the RUN or STOP state can be maintained.

With an FX2 Series PC, RUN/STOP control equivalent to that of F_2 series can be accomplished with a special auxiliary relay. The general input shown below is used as the STOP input.

(JAPANESE SPECIFICATION)



- RUN monitor M8000 is turned ON when the RUN input pushbutton is pressed and special auxiliary relays M8035 and M8036 are operated.

  The RUN status is self-maintained by this relay operation.

- The self-maintained RUN status is canceled when the END instruction is executed after the STOP input (X1) turned ON. The PC stops.

  Therefore, the RUN input is given priority when the RUN and STOP input pushbutton switches are pressed at the same time.

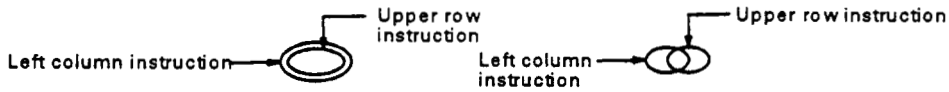
  To give priority to the STOP input, interlock with the external input circuit so that the RUN input is turned OFF while the STOP pushbutton is pressed.

- When a STOP input is given, M8035, M8036, and M8037 are turned OFF.

2. PROGRAM FLOW CONTROL INSTRUCTIONS

The relationship between an MC-MCR instruction and a CJ instruction are detailed in Section 5.2.1. The other program flow instructions are explained below.

In the following tables, the symbol " ◎ " indicates an "inclusive" relationship and the symbol " ⊙ " indicates cases where the zones overlap.



Upper row / Left column	MC-MCR	CJ-P	EI-DI	FOR-NEXT	STL-RET
MC-MCR	◎ ○	◎ ○	◎ ○	◎ ○	◎ ○
	⊙ △	⊙ △	⊙ ○	⊙ × (6607)	⊙ × (6605)
CJ-P	◎ ○	◎ ○	◎ ○	◎ ○	◎ ○
	⊙ △	⊙ △	⊙ ○	⊙ △	⊙ △
EI-DI	◎ ○	◎ ○	◎ ○	◎ ○	◎ ○
	⊙ ○	⊙ ○	⊙ ○	⊙ ○	⊙ ○
FOR-NEXT	◎ × (6607)	◎ ○	◎ ○	◎ ○ 5 nesting levels	◎ × (6607)
	⊙ × (6607)	⊙ △	⊙ ○	⊙ ✱ 2	⊙ × (6607)
STL-RET	◎ × (6605)	◎ △	◎ ○	◎ ○ (Within one STL)	◎ ○
	⊙ × (6605)	⊙ △	⊙ ○	⊙ × (6607)	⊙ ○
P-SRET	◎ × (6606)	◎ ○	◎ ○	◎ ○	◎ × (6606)
	⊙ × (6608)	⊙ △	⊙ ○	⊙ × (6607)	⊙ × (6605)
I-IRET	◎ × (6606)	◎ ○	◎ △	◎ ○	◎ × (6606)
	⊙ × (6606)	⊙ △	⊙ △	⊙ × (6607)	⊙ × (6606)
FEND-END	◎ ○	◎ ○	◎ ○	◎ ○	◎ △
	⊙ × (6608)	⊙ × (6701)	⊙ ✱ 1	⊙ × (6607)	⊙ × (6605)
O-FEND	◎ ○	◎ ○	◎ ○	◎ ○	◎ ○
	⊙ × (6608)	⊙ ○	⊙ ○	⊙ × (6607)	⊙ × (6605)
O-END (NO FEND)	◎ ○	◎ ○	◎ ○	◎ ○	◎ ○
	⊙ × (6608)	⊙ × (6701)	⊙ ✱ 1	⊙ × (6607)	⊙ × (6605)

PROGRAM FLOW CONTROL INSTRUCTIONS

O : Combinations which can be used without restrictions.

X : Inhibited combinations - the numbers in parentheses indicates error numbers.

Δ : These combinations are not inhibited, although they are not recommended for use.

(1)

P-SRET	I-IRET	FEND-END
◎ X (6608)	◎ X (6608)	◎ X (6608)
○ X (6606)	○ X (6606)	○ X (6608)
◎ Δ	◎ Δ	◎ X (6701)
○ Δ	○ Δ	○ O
◎ O	◎ O	◎ ✱1
○ O	○ O	○ O
◎ X (6607)	◎ X (6607)	◎ X (6607)
○ X (6701)	○ X (6607)	○ X (6607)
◎ X (6605)	◎ X (6605)	◎ X (6605)
○ X (6606)	○ X (6606)	○ X (6605)
◎ X (6606)	◎ X (6606)	◎ X (6709)
○ X (6606)	○ X (6606)	○ X (6709)
◎ X (6606)	◎ X (6606)	◎ X (6606)
○ X (6606)	○ X (6606)	○ X (6606)
◎ O	◎ O	◎ ✱3
○ X (6709)	○ X (6709)	○ ✱3
◎ X (6606)	◎ X (6606)	◎ ✱3
○ X (6709)	○ X (6606)	○ ✱3
◎ X (6606)	◎ X (6606)	◎ ✱3
○ X (6709)	○ X (6706)	○ ✱3

Remarks

✱1 Enters a state as if DI instruction is missing. An error does not occur.

✱2 FOR FOR NEXT NEXT

Sequence is controlled as indicated by the solid lines.

✱3 The first FEND or END instruction becomes effective, precluding the expected program flow.

The combination of instructions that have an "inclusive" relationship is permissible with some exceptions. Note the following exceptions.

(1) MC-MCR cannot be used in blocks with FOR-NEXT, P-SRET, or I-IRET.

(2) STL-RET cannot be used in blocks with FOR-NEXT, P-SRET, or I-IRET.

(3) Program flow may not be discontinued while inside blocks of MC-MCR, FOR-NEXT, P-SRET, or I-IRET with an I, IRET, SRET, FEND, or END instruction.

REVISIONS

Edition Date	Manual Number	Revision
Sep. '91	JY992D26301A	First edition.

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.

HANDY MANUAL

FX-SERIES PROGRAMMABLE CONTROLLERS

MITSUBISHI ELECTRIC AUTOMATION, INC.

500 CORPORATE WOODS PARKWAY • VERNON HILLS, ILLINOIS 60061

JY992D26301A
HI-IB-073-A9204

Effective APR. 1992
Specifications are subject
to change without notice.