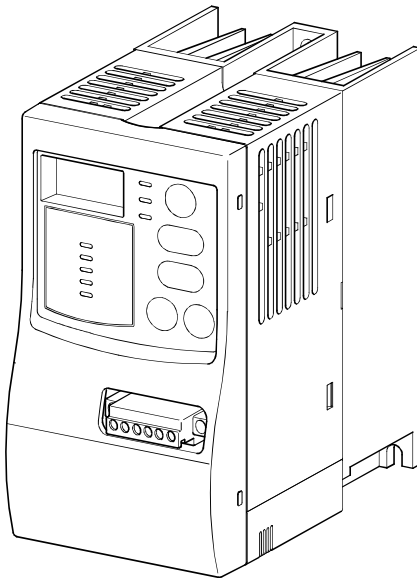




TRANSISTORIZED INVERTER FR-C₅₀₀

PROGRAMMING MANUAL



OVERVIEW	Chapter 1
SEQUENCE PROGRAM LANGUAGES AND OPERATIONS	Chapter 2
DESCRIPTION OF DEVICES	Chapter 3
PLC FUNCTION	Chapter 4
STRUCTURES OF INSTRUCTIONS	Chapter 5
PLC INSTRUCTIONS	Chapter 6
ERROR CODE LIST	Chapter 7

CONTENTS

1. OVERVIEW	1
1.1 Outline of Operation Processings	2
1.2 RUN and STOP Operation Processings	4
1.3 Program Makeup	4
2. SEQUENCE PROGRAM LANGUAGES AND OPERATIONS	5
2.1 Programming Languages	6
2.1.1 Relay symbolic language (Ladder mode)	6
2.1.2 Logic symbolic language (List mode)	8
2.2 Operation Processing Method of PLC Function	9
2.3 I/O Processing Method	10
2.3.1 What is refresh system?	10
2.3.2 Response delay in refresh system	11
2.4 Scan Time	12
2.5 Numerical Values Usable in Sequence Program	13
2.5.1 BIN (Binary Code)	14
2.5.2 HEX (HEX Decimal)	15
3. DESCRIPTION OF DEVICES	17
3.1 Device List	18
3.2 Inputs, Outputs X, Y	19
3.2.1 Inputs X	20
3.2.2 Outputs Y	21
3.3 Internal Relays M	22
3.4 Timers T	23
3.4.1 100ms, 10ms and 100ms retentive timers	23
3.4.2 Timer processing method and accuracy	24
3.5 Counters C	26
3.5.1 Count processing in refresh system	27
3.5.2 Maximum counting speed of counter	28
3.6 Data Registers D	29
3.7 Special Relays, Special Registers	30

4. PLC FUNCTION	33
<hr/> <hr/>	
4.1 Function List	34
4.2 How to RUN/STOP the Built-in PLC Function from Outside (Remote RUN/STOP)	35
4.3 Watchdog Timer (Operation clog up monitor timer)	37
4.4 Self-diagnostic Function	38
4.4.1 Error-time operation mode	39
4.5 Keyword Registration	40
4.6 Setting of Output (Y) Status at Switching from STOP Status to RUN Status	41
5. STRUCTURES OF INSTRUCTIONS	43
<hr/> <hr/>	
5.1 Instruction Format	44
5.2 Bit Device Processing Method	46
5.2.1 1-bit processing.....	46
5.2.2 Digit designation processing	46
5.3 Handling of Numerical Value	48
5.4 Operation Error	49
6. PLC INSTRUCTIONS	51
<hr/> <hr/>	
6.1 PLC Instructions	52
6.1.1 How to use the instruction list	52
6.1.2 PLC instruction list	54
6.2 Description of the Instructions	57
6.3 Contact Instructions	58
6.3.1 Operation start, series connection, parallel connection ... LD, LDI, AND, ANI, OR, ORI	58
6.4 Connection Instructions	61
6.4.1 Ladder block series connection, parallel connection ... ANB, ORB	61
6.4.2 Operation result, push, read, pop ... MPS, MRD, MPP.....	64
6.5 Output Instructions	67
6.5.1 Bit device, timer, counter ... OUT.....	67
6.5.2 Device set, reset ... SET, RST	70
6.5.3 Leading edge, trailing edge differential outputs ... PLS, PLF.....	73
6.6 Shift Instructions	75

6.6.1	Bit device shift ... SFT, SFTP	75
6.7	Master Control Instructions	77
6.7.1	Master control set, reset ... MC, MCR	77
6.8	End Instruction	81
6.8.1	Sequence program end ... END	81
6.9	Other Instructions	82
6.9.1	No operation ... NOP	82
6.10	Comparison Operation Instructions	84
6.10.1	16-bit data comparison ... =, <>, >, <=, <, >=	86
6.11	Data Transfer Instructions	88
6.11.1	16-bit data transfer ... MOV, MOVP	88
6.12	16-bit Logical Product ... WAND, WANDP	90
6.13	16-bit Logical Add ... WOR, WORP	93
 7. ERROR CODE LIST		 97
<hr/> <hr/>		
7.1	How to Read the Error Code	98

1. OVERVIEW

This manual describes the functions and devices necessary for programming.

1.1	Outline of Operation Processings	2
1.2	RUN and STOP Operation Processings	4
1.3	Program Makeup	4

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

1.1 Outline of Operation Processings

This section outlines processings performed from when the inverter is powered on until a sequence program is executed.

The built-in PLC function processings are roughly classified into the following three types.

(1) Initial processing

Pre-processing for executing sequence operation. This processing is executed only once when power is switched on or a reset is performed.

- (a) The inputs/outputs are reset and initialized.
- (b) The data memories are initialized (the bit devices are turned off and the word devices are cleared to 0).
- (c) Self-diagnostic checks are made on the built-in PLC function parameter setting, operation circuit, etc.

REMARKS

The built-in PLC function parameters can be confirmed from GX Developer. (Refer to the GX Developer Operating Manual.)
For the setting list of built-in PLC function parameter, refer to the instruction manual of the FR-C500 series.

(2) Sequence program operation processing

The sequence program written to the built-in PLC function is executed from step 0 to an END instruction.

(3) END processing

Post-processing for terminating one sequence program operation processing and return the sequence program execution to step 0.

- (a) Self-diagnostic checks are performed.
- (b) The present values of the timers are updated and their contacts are turned on/off, and the present values of the counters are updated and their contacts are turned on.

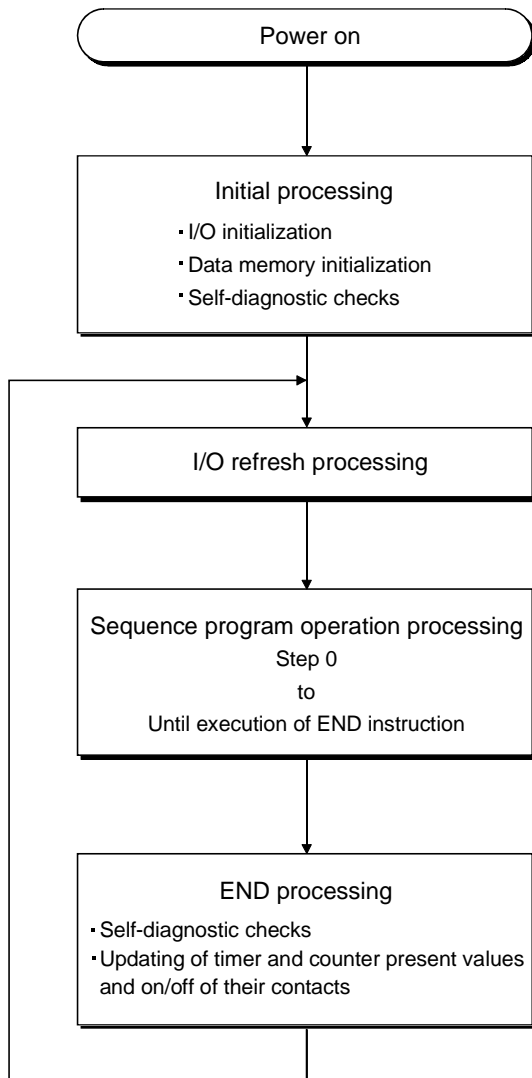


Fig 1.1 Operation Processings of Built-in PLC function

1.2 RUN and STOP Operation Processings

The built-in PLC function has two different operation statuses: RUN status and STOP status.

This section explains the operation processings of the built-in PLC function in each operating status.

(1) Operation processing in RUN status

A RUN status indicates that a sequence program repeats its operation in order of step 0 to END (FEND) instruction to step 0 when SQ-SD are shorted. (P.RUN is on)
When entering the RUN status, the function outputs the output status saved at STOP according to the "STOP to (RUN-time output mode setting" (refer to page 41).

(2) Operation processing in STOP status

A STOP status indicates that a sequence program stops its operation when SQ-SD are opened or remote STOP is commanded. (P.RUN is off)
When entering the STOP status, the function saves the output status and turns off all outputs. The contents of the data memories other than the outputs (Y) are maintained.

POINT

In either the RUN or STOP status, the built-in PLC function is performing I/O refresh processings. In the STOP status, therefore, I/O monitoring and test operation can be performed from the peripheral device.

1.3 Program Makeup

(1) Program classification

The program that can be used by the built-in PLC function is a main sequence program only. Microcomputer, interrupt and SFC programs cannot be used.

(2) Program capacity

A program capacity indicates the capacity of the program storage memory, and it is 1k steps. Set the program capacity in the built-in PLC function parameter.

2. SEQUENCE PROGRAM LANGUAGES AND OPERATIONS

This chapter explains the programming languages and numerical representations necessary for programming.

2.1	Programming Languages	6
2.2	Operation Processing Method of PLC Function	9
2.3	I/O Processing Method	10
2.4	Scan Time.....	12
2.5	Numerical Values Usable in Sequence Program .	13

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

2.1 Programming Languages

The built-in PLC function has two different programming methods: one that uses ladders and the other that uses dedicated instructions.

- Programming that uses ladders is performed in the relay symbolic language. *1
- Programming that uses dedicated instructions is performed in the logic symbolic language. *2

Whether the relay symbolic language or logic symbolic language is used, the same program is created.

REMARKS

- *1. When using GX Developer for programming, perform programming in the "ladder mode".
- *2. When using GX Developer for programming, perform programming in the "list mode".

2.1.1 Relay symbolic language (Ladder mode)

The relay symbolic language is based on the concept of a relay control circuit.

You can perform programming in the representation close to the sequence circuit of relay control.

(1) Ladder block

A ladder block is the minimum unit for performing sequence program operation. It starts with the left hand side vertical bus and ends with the right hand side vertical bus.

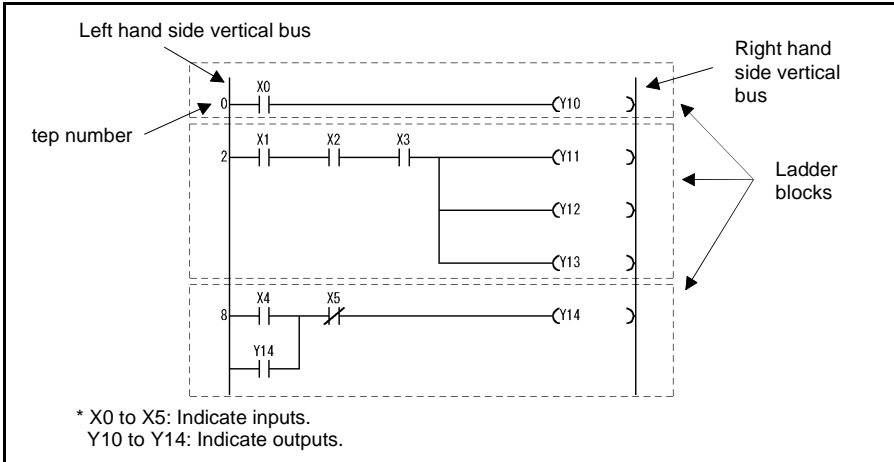


Fig 2.1 Ladder Blocks

(2) Sequence program operation method

Sequence program operation repeats execution from a ladder block at step 0 to an END instruction.

In a single ladder block, operation is performed from the left hand side vertical bus to the right, and from the top to the bottom.

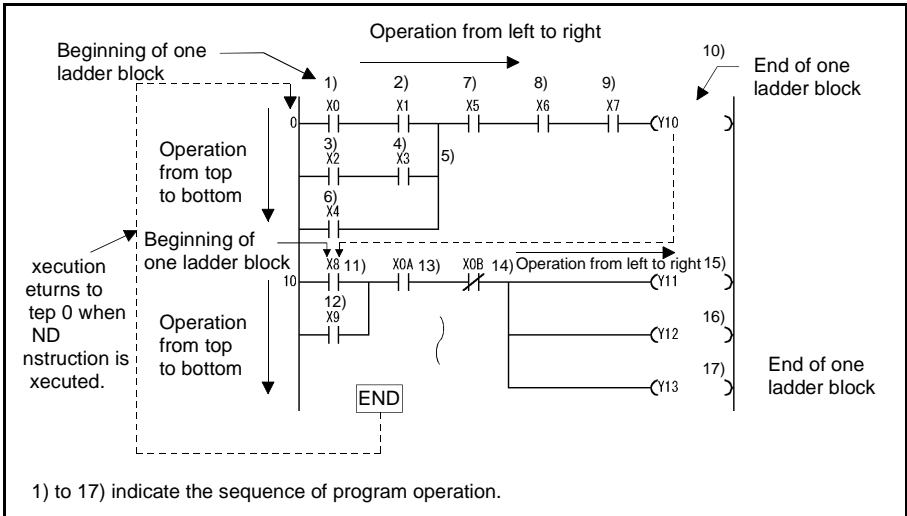


Fig 2.2 Operation Processing Sequence

2.1.2 Logic symbolic language (List mode)

The logic symbolic language uses dedicated instructions for programming contacts, coils, etc. instead of their symbols used by the relay symbolic language.

(1) Program operation method

Sequence program operation is executed from an instruction at step 0 to an END instruction in due order. When the END instruction is executed, operation is executed from the instruction at step 0 again.

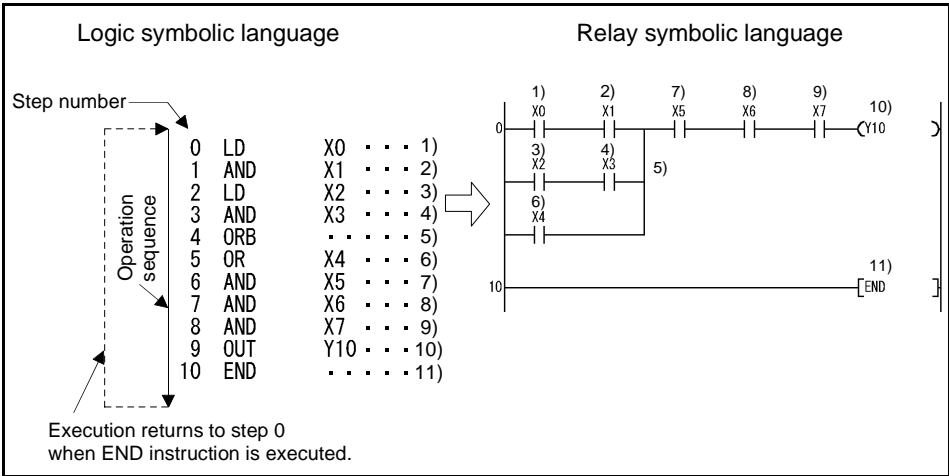


Fig 2.3 Operation Processing Sequence

2.2 Operation Processing Method of PLC Function

The operation processing method is the repeated operation of a stored program.

(1) Stored program system

- 1) In a stored program system, a sequence program to be operated is stored in the internal memory beforehand.
- 2) When sequence program operation is executed, the sequence program stored in the built-in PLC function is read to the CPU instruction by instruction to execute the operation, and the corresponding devices are controlled according to the results.

(2) Repeated operation system

In a repeated operation system, a sequence of operations is repeated. The built-in PLC function repeats the following processings.

- 1) The built-in PLC function executes the sequence program stored in the internal memory from step 0 in due order.
- 2) When the END instruction is executed, internal processings, such as timer/counter present value updating and self-diagnostic checks, are performed, and the execution returns to step 0 of the sequence program again.

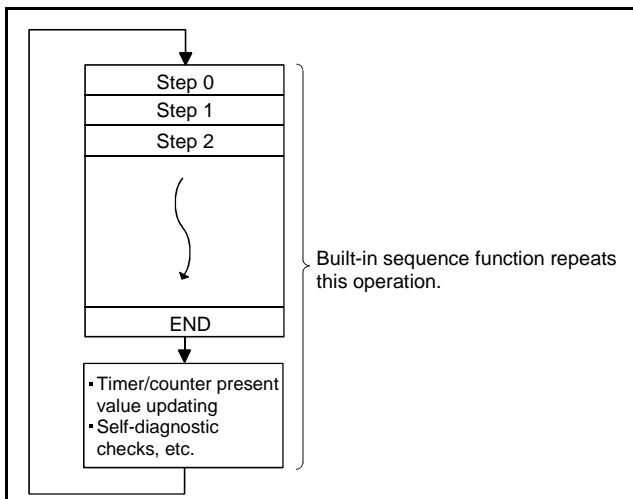


Fig 2.4 Operation Processing Method of Built-in PLC Function

REMARKS

A processing from step 0 to next step 0 or from END to next END is called one scan. Therefore, one scan is the sum of the processing time of a user-created program (step 0 to END) and the internal processing time of the built-in PLC function.

2.3 I/O Processing Method

The control system is a refresh system.

2.3.1 What is refresh system?

In the refresh system, control input terminal changes are batch-imported into the input data memory of the CPU before execution of each scan, and the data of this input data memory are used as the input data for operation execution.

Each program operation result of the output (Y) is output to the output data memory, and after the END instruction is executed, the contents of the output data memory are batch-output from the control output terminal.

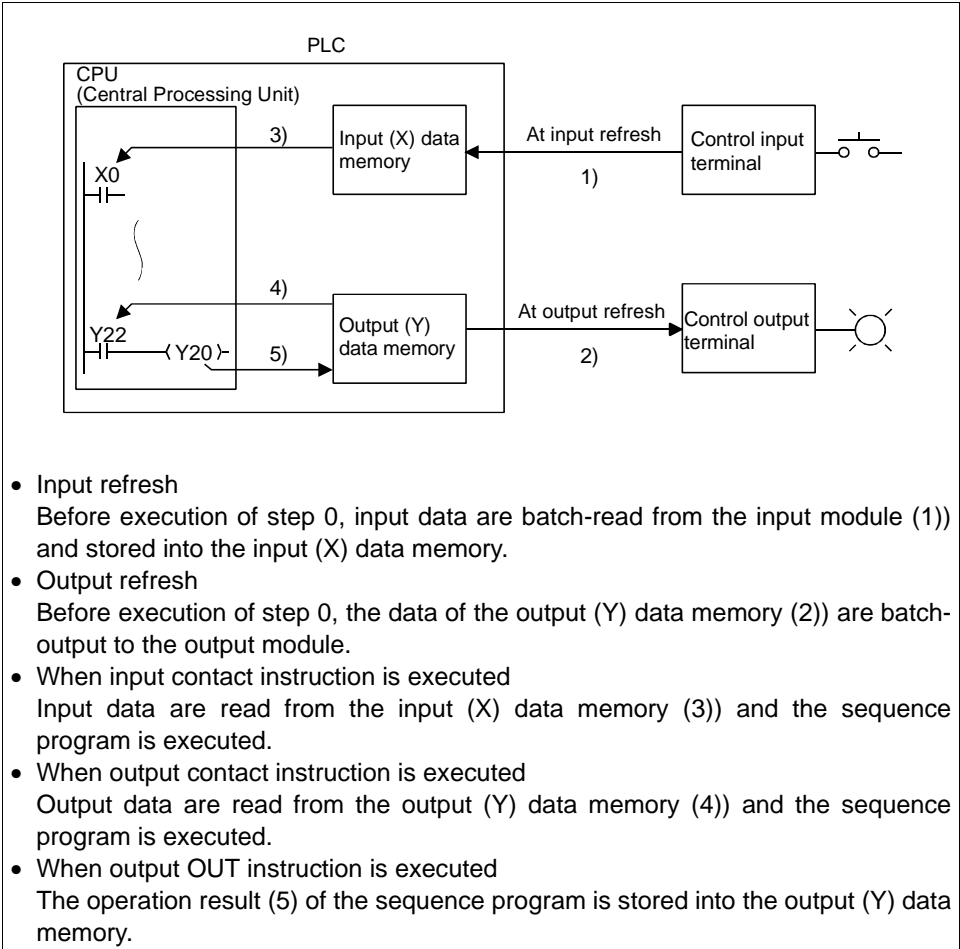
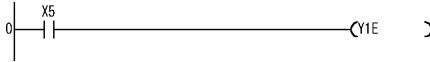


Fig 2.5 I/O Data Flows in Refresh System

2.3.2 Response delay in refresh system

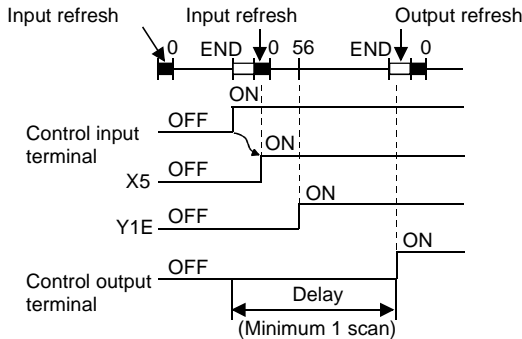
This section describes a delay of an output change in response to an input change. An output change in response to an input change has a delay of up to two scans as shown in Fig. 2.6.

Ladder example



In this ladder, output Y1E turns on when input X5 turns on.

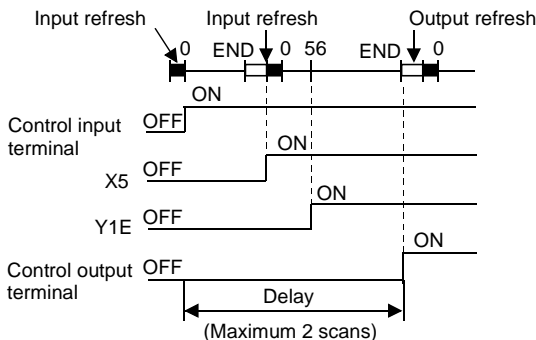
When Y1E turns on earliest



The Y1E output turns on earliest when the control input terminal turns from OFF to ON immediately before a refresh. X5 turns on at an input refresh, Y1E turns on at step 0, and the control output terminal turns on at an output refresh after execution of the END instruction.

In this case, therefore, a delay of a control output terminal change in response to a control input terminal change is one scan.

When Y1E turns on latest



The Y1E output turns on latest when the control input terminal turns from OFF to ON immediately after a refresh. X5 turns on at the next input refresh, Y1E turns on at step 0, and the control output terminal turns on at an output refresh after execution of the END instruction.

In this case, therefore, a delay of a control output terminal change in response to a control input terminal change is two scans.

Fig 2.6 Output Y Change in Response to Input X Change

2.4 Scan Time

(1) Scan time

A scan time is a time from when sequence program operation is executed from step 0 until step 0 is executed again.

The scan time of each scan is not equal, and changes depending on whether the used instructions are executed or not.

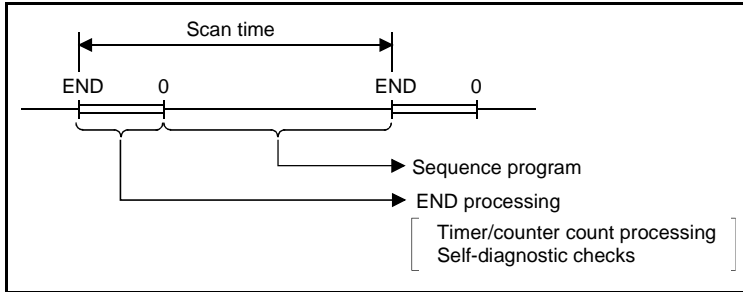


Fig 2.7 Scan Time

(2) Scan time confirmation

(a) The scan time from the END instruction to the next END instruction is timed in the PLC, and stored into the special registers D9017 to D9019 in units of 10ms.

1) Data stored into special registers D9017 to D9019

- D9017..... Minimum value of scan time
- D9018..... Present value of scan time
- D9019..... Maximum value of scan time

2) Scan time accuracy

The accuracy of the scan time observed in the PLC is ± 10 ms.

For example, when the D9018 data is 5, the actual scan time is 40ms to 60ms.

2.5 Numerical Values Usable in Sequence Program

The built-in PLC function represents numerical values, alphabets and other data in two statuses: 0 (OFF) and 1 (ON).

The data represented by these 0s and 1s are called BIN (binary code).

The built-in PLC function can also use HEX (hexadecimal code) that represents BIN data in blocks of four bits.

Table 2.1 indicates the numerical representations of BIN, HEX and decimal code.

Table 2.1 Numerical Representations of BIN,
HEX and Decimal Code

DEC (Decimal Code)	HEX (Hexadecimal Code)	BIN (Binary Code)
0	0	0
1	1	1
2	2	10
3	3	11
•	•	•
•	•	•
•	•	•
•	•	•
•	•	•
•	•	•
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	10000
17	11	10001
•	•	•
•	•	•
•	•	•
•	•	•
•	•	•
•	•	•
47	2F	101111

2.5.1 BIN (Binary Code)

(1) Binary code

BIN is a numerical value represented by 0s (OFF) and 1s (ON).

In the decimal code, a number is incremented from 0 to 9, and at this point, a carry occurs and the number is incremented to 10.

In BIN, 0, 1 are followed by a carry, and the number is incremented to 10 (2 in decimal).

Table 2.2 indicates the numerical representations of BIN and decimal code.

Table 2.2 Differences between Numerical Representations of BIN and Decimal Code

DEC (Decimal Code)	BIN (Binary Code)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011

(2) Numerical representation of BIN

1) Each register (e.g. data register) of the built-in PLC function consist of 16 bits.

Each bit of the register is assigned a 2^n value.

However, the most significant bit is used to judge whether the value is positive or negative.

- Most significant bit is 0 Positive
- Most significant bit is 1 Negative

The numerical representation of each register of the built-in PLC function is shown in Fig. 2.8.

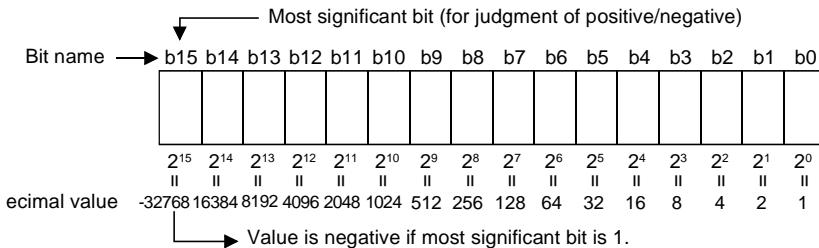


Fig 2.8 Numerical Representation of Each Register of Built-in PLC Function

2) Numerical data usable with the built-in PLC function

In the numerical representation shown in Fig. 2.8, values can be represented in

the range -32768 to 32767.

Therefore, each register of the built-in PLC function can store any value between -32768 and 32767.

2.5.2 HEX (HEX Decimal)

(1) HEX

HEX represents four bits of BIN data as one digit.

Using four bits in BIN, you can represent 16 values from 0 to 15.

Since HEX represents any of 0 to 15 in a single digit, 9 is followed by alphabets A (instead of 10), B (11)..., and F (15) is followed by a carry.

Refer to page 13 for the correspondences between BIN, HEX and decimal code.

(2) Numerical representation of HEX

Each register (e.g. data register) of the built-in PLC function consist of 16 bits.

Therefore, the value that can be stored into each register is represented as any of 0 to HFFFF in HEX.

MEMO

3. DESCRIPTION OF DEVICES

3.1	Device List	18
3.2	Inputs, Outputs X, Y	19
3.3	Internal Relays M.....	22
3.4	Timers T.....	23
3.5	Counters C	26
3.6	Data Registers D.....	29
3.7	Special Relays, Special Registers	30

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

3.1 Device List

The following table indicates the device names usable with the built-in PLC function and their ranges of use.

Table 3.1 Device List

Input (X)	48 (X0 to X2F) <6 points installed>	
Output (Y)	48 (Y0 to Y2F) <2 points installed>	
Internal relay (M)	64 (M0 to M63)	
Latch relay (L)	None (Can be set with built-in PLC function parameters but will not latch)	
Step relay (S)	None (Can be set with built-in PLC function parameters but will operate as M)	
Link relay (B)	None	
Timer (T)	Points	8(T0 to 7)
	Specifications	100ms timer: Set time 0.1 to 3276.7s 10ms timer: Set time 0.01 to 327.67s 100ms retentive timer: Set time 0.1 to 3276.7s
Counter (C)	Points	8(C0 to 7)
	Specifications	Normal counter: Setting range 1 to 32767 Interrupt program counter: None
Data device (D)	120(D0 to D119)	
Link register (W)	None	
Annunciator (F)	None	
File register (R)	None	
Accumulator (A)	None	
Index register (Z, V)	None	
Pointer (P)	None	
Interrupt pointer (I)	None	
Special relay (M)	256 (M9000 to 9255) with function limit	
Special register (D)	256 (D9000 to 9255) with function limit	

3.2 Inputs, Outputs X, Y

Inputs and outputs are devices designed to transfer data between the inverter and external devices.

Inputs provide ON/OFF data given to the corresponding control input terminals from outside the inverter. In a program, they are used as contacts (normally open, normally closed contacts) and the source data of basic instructions. Outputs are used when the operation results of a program are output from the control output terminals to outside the inverter.

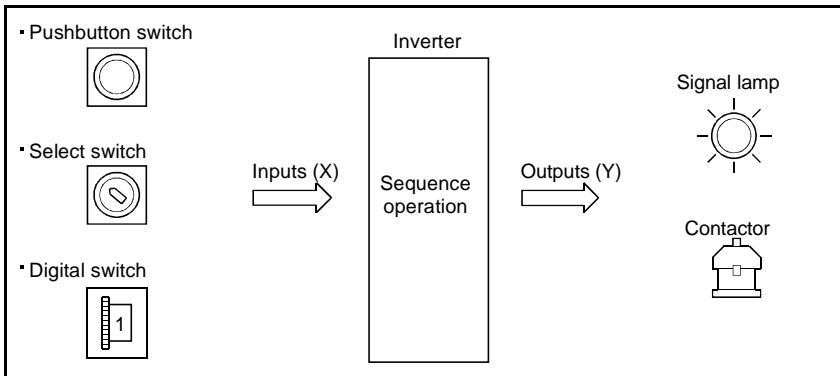


Fig 3.1 Inputs (X), Outputs (Y)

3.2.1 Inputs X

- (1) Inputs are designed to give commands and data from external devices, such as pushbuttons, select switches, limit switches and digital switches, to the inverter (built-in PLC function).
- (2) On the assumption that the PLC function contains a virtual relay X_n for one input point, the normally open (N/O) or normally closed (N/C) contact of that X_n is used in the program.

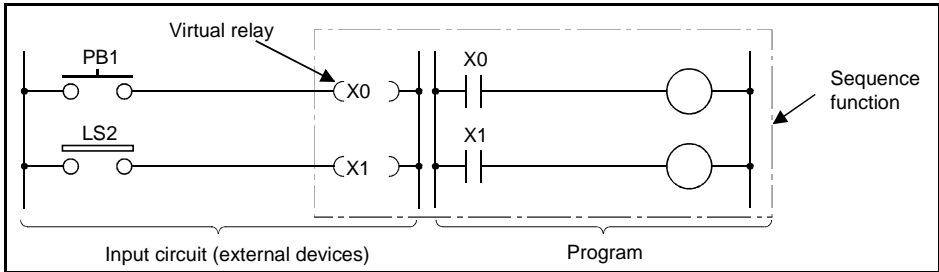


Fig 3.2 Concept of Inputs (X)

- (3) There are no restrictions on the number of N/O and N/C contacts of X_n used in the program.

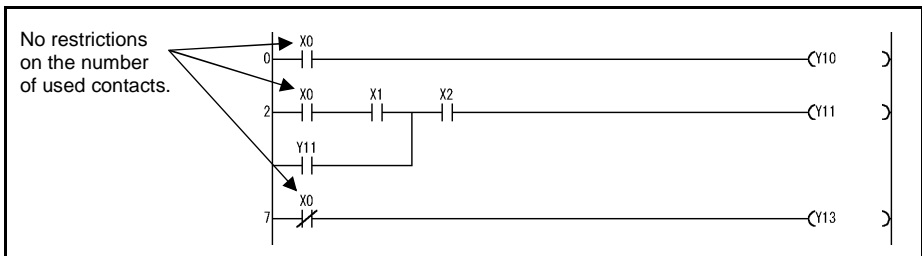


Fig 3.3 Use of Contacts in Input (X) Program

When no external devices are connected to the control input terminals, "X" can be used as the internal relay "M".

3.2.2 Outputs Y

- (1) Outputs are designed to output the control results of a program to outside the inverter (signal lamps, digital indicators, magnetic switches (contactors), solenoids, etc.).
- (2) An output can be exported to outside the inverter as equivalent to one N/O contact.
- (3) There are no restrictions on the number of N/O and N/C contacts of output Y_n used in the program, if they are used within the program capacity range.

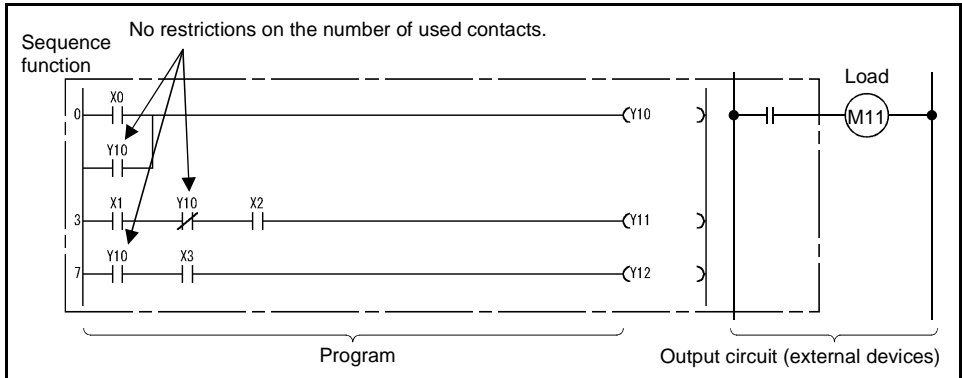


Fig 3.4 Concept of Outputs (Y)

When no external devices are connected to the control output terminals, "Y" can be used as the internal relay "M".

3.3 Internal Relays M

Internal relays are auxiliary relays that are used in the PLC function and cannot latch data (backup for power failure).

All internal relays are turned off when:

- Power is switched from off to on; or
- Reset is performed.

There are no restrictions on the number of contacts (N/O and N/C contacts) used in the program.

Use outputs (Y) when outputting the operation results of the sequence program to outside the inverter.

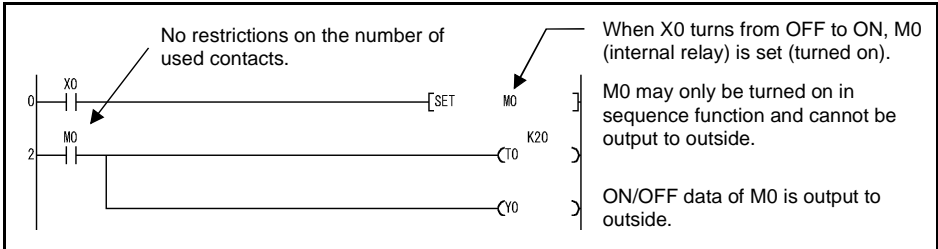


Fig 3.5 Internal Relay

3.4 Timers T

The timers of the PLC function are count up timers.

The count up timer starts timing the present value when its coil turns on, and the contact of that timer turns on when the present value reaches the setting (time-out).

3.4.1 100ms, 10ms and 100ms retentive timers

(1) 100ms and 10ms timers

The timer starts timing the present value when its coil turns on, and the present value is reset to 0 and the contact turns off when the coil turns off.

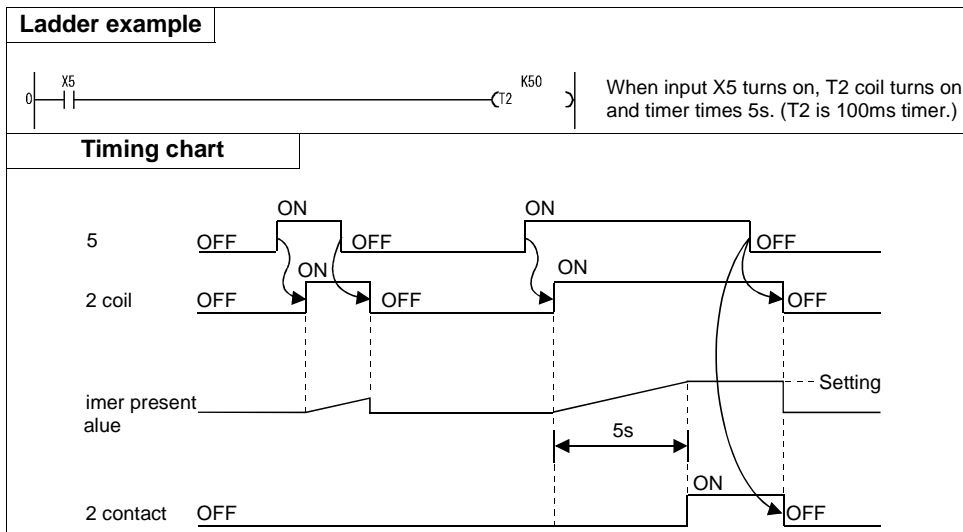


Fig 3.6 Timing Chart

REMARKS

100ms, 10ms and 100ms retentive timers can be changed using the built-in PLC function parameter. (The default is a 100ms timer.)

Since the FR-C500 has 8 timers (T0 to T7), it can use only any one type of 100ms, 10ms and 100ms retentive timers.

(2) 100ms retentive timers

1) A 100ms retentive timer is designed to time the ON period of the timer coil.

When its coil turns on, the timer starts timing the present value and maintains the present value and contact ON/OFF state if the coil turns off.

When the coil turns on again, the timer resumes timing from the maintained present value.

2) Use the RST T□ instruction to clear the present value and turn off the contact.

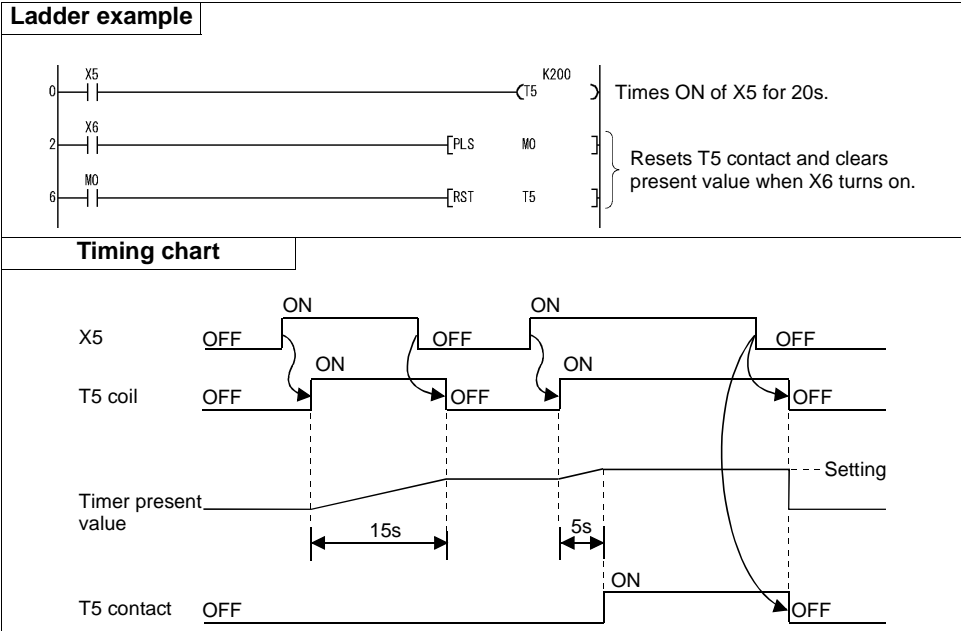


Fig 3.7 Timing Chart

3.4.2 Timer processing method and accuracy

(1) Timer processing method

The coil of the timer is turned on/off at execution of the OUT T□ instruction, and the timer's present value is updated and its contact turned on/off at execution of the END instruction.

1) When the coil of the timer turns on, the present value of that timer is updated after execution of the END instruction, and when the timer times out, its contact turns on.

(a) When the coil of the 10ms or 100ms timer turns off, the present value of that timer is reset to 0 and the contact is also turned off after execution of the END instruction.

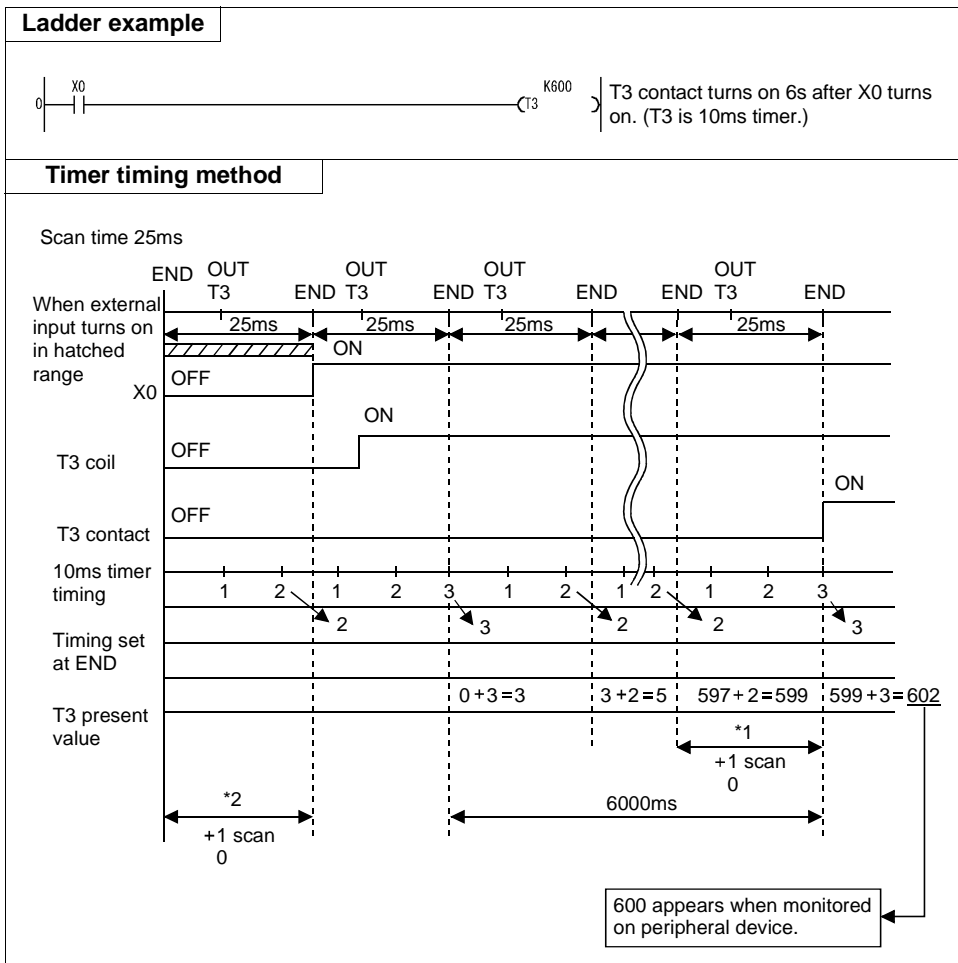
(b) If its coil turns off, the 100ms retentive timer maintains the present value and contact ON/OFF state.

2) When the timer is reset by the RST instruction, the present value of the timer is reset to 0 and the contact turns off too at execution of the RST T□ instruction.

POINT

If the timer setting is "0", the setting becomes infinite and the timer does not time out.

- (2) Present value update timing and accuracy in refresh system
- 1) The timer accuracy is +2 scan times independently of the used timer and scan time.
 - 2) The following shows the present value update timing and accuracy when the 10ms timer is used in a program where the scan time is 10ms or more.



DESCRIPTION OF DEVICES

3

Fig 3.8 Timer Timing Method

In Fig. 3.8, the time-out period of the 10ms timer T3 has the following errors.

- *1..... 10ms timer error (+1 scan time)
- *2..... Error produced by timer's input condition ON timing and OUT T□ instruction's program position (+1 scan time)

The accuracy is +2 scan time (+0.05s in Fig. 3.8)

- 3) When the timer times out, its contact remains on until END even if the coil turns off, and turns off at execution of the END instruction.

3.5 Counters C

The counters of the built-in PLC function are up counters.

An up counter stops counting and its contact turns on when the count value reaches the setting.

(1) Count processing

- 1) The coil of the counter is turned on/off at execution of the OUT C□ instruction, and its present value is updated and its contact turns on after execution of the END instruction.
- 2) The counter counts on detection of the leading edge (OFF to ON) of the coil. It does not count if the coil remains on.

(2) Counter resetting

- 1) The count value is not cleared even if the coil turns off. Use the RST C□ instruction to clear the count value and turn off the contact.
- 2) When the counter is reset by the RST instruction, the present value and contact of the counter are cleared at execution of the RST instruction.

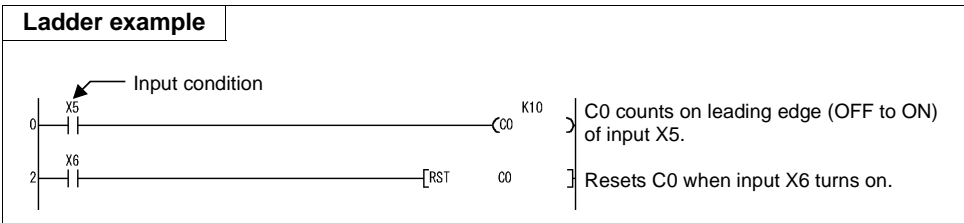


Fig 3.9 Count Ladder

3.5.1 Count processing in refresh system

The counter counts on the leading edge of the input condition of the counter imported at an input refresh.

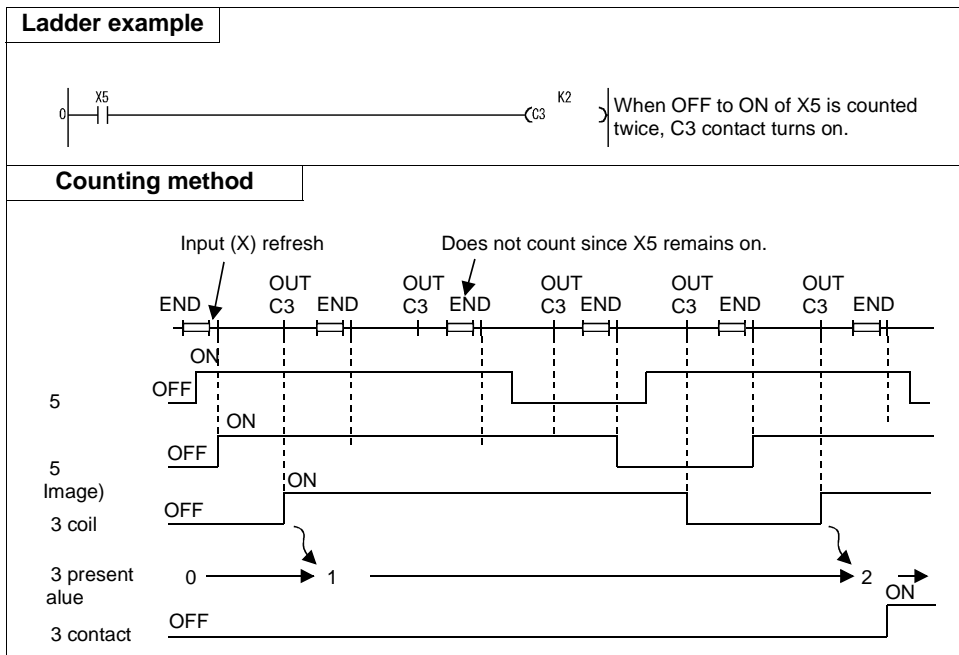


Fig 3.10 Counter Counting Method

REMARKS

Refer to page 28 for the maximum counting speed of the counter.

3.5.2 Maximum counting speed of counter

The maximum counting speed of the counter is determined by the scan time, and the counter can count only when the ON/OFF period of the input condition is longer than the scan time.

$$\text{Maximum counting speed } C_{\max} = \frac{n}{100} \times \frac{1}{t_s} \text{ [times/s]}$$

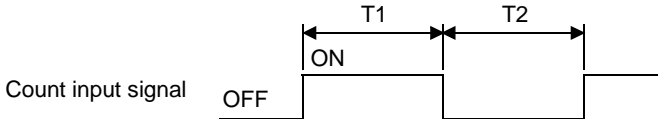
n: Duty (%)
ts: Scan time [s]

REMARKS

The duty n is a percent (%) ratio of ON/OFF period to (ON + OFF period) of the count input signal.

$$\text{When } T_1 \leq T_2 \quad n = \frac{T_1}{T_1 + T_2} \times 100[\%]$$

$$\text{When } T_1 > T_2 \quad n = \frac{T_2}{T_1 + T_2} \times 100[\%]$$



3.6 Data Registers D

- (1) Data registers are memories that can store numerical data (-32768 to 32767 or H0000 to HFFFF) in the built-in PLC function.

One point of data register consists of 16 bits and allows data to be read/written in units of 16 bits.

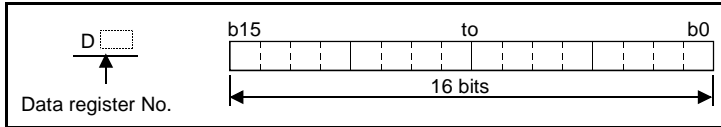


Fig 3.11 Data Register Structure

- (2) The data stored once by the sequence program is maintained until other data is stored.
- (3) If more data registers are needed, the unused timers (T) and counters (C) can be used as data registers.

3.7 Special Relays, Special Registers

Special relays and special registers are internal relays and data registers, respectively, whose applications are predetermined by the built-in PLC functions.

They have the following main applications.

(1) Sequence operation check

The special relays and special registers can be used to:

- (a) Check the operating status (RUN/STOP)
- (b) Detect a fault by the self-diagnostic function
- (c) Detect an operation error
- (d) Check the scan time

(2) Timing contact

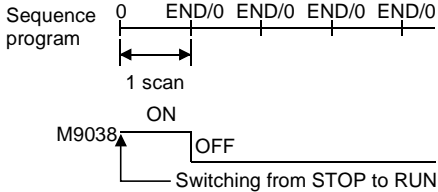
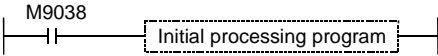
There are special relays that can be used in a sequence program and differ in operating status.

- (a) Normally ON/OFF flag
- (b) RUN flag (OFF for 1 scan)
- (c) Initial processing flag (ON for 1 scan)

REMARKS

For the special relays and special registers usable with the built-in PLC function, refer to the instruction manual of the FR-C500 series.

Table3.2 Special Relay Application List

Item	Special Relay	Application/Description
Initial processing flag (1 scan ON)	M9038	<p>(1) This relay turns on for one scan when the built-in PLC function switches from STOP to RUN.</p>  <p>(2) Using M9038, you can create a sequence program to be executed only once without using the PLS instruction at switching from STOP to RUN.</p> 
Normal OFF flag	M9037	This relay remains off while power is on. Can be used to temporarily disable execution for debugging, etc.
Normally ON flag	M9036	This relay is on while power is on. Can be used to create a program to be executed only once after power-on.
RUN flag	M9039	This relay turns on at the second scan of the sequence program when SQ-SD are shorted.

MEMO

4. PLC FUNCTION

4.1	Function List.....	34
4.2	How to RUN/STOP the Built-in PLC Function from Outside (Remote RUN/STOP).....	35
4.3	Watchdog Timer (Operation clog up monitor timer)	37
4.4	Self-diagnostic Function	38
4.5	Keyword Registration	40
4.6	Setting of Output (Y) Status at Switching from STOP Status to RUN Status	41

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

4.1 Function List

Function	Description
Remote RUN/STOP	<ul style="list-style-type: none"> • This function performs remote RUN/STOP from outside the inverter when SQ-SD are shorted (PLC function in RUN status (P.RUN lit)).
Watchdog timer variable (10 to 2000ms)	<ul style="list-style-type: none"> • The watchdog timer is an internal timer of the sequence function designed to detect hardware or program faults and can be changed in setting.
Self-diagnostic function	<ul style="list-style-type: none"> • The built-in PLC function itself diagnoses faults and performs fault detection, indication, built-in sequence function stop, etc.
STOP to RUN-time output setting	<ul style="list-style-type: none"> • This setting is made to determine the output (Y) state when the function has switched from the STOP status to the RUN status.
Keyword registration	<ul style="list-style-type: none"> • This setting is made to inhibit read/interrupt of a program (parameters and main/sub program) and comments.

CAUTION

The following functions are unavailable.

Constant scan, latch (backup for power failure), PAUSE, status latch, sampling trace, step run, clock, interrupt processing, comment, microcomputer mode, print title registration, annunciator display mode, ERROR LED priority setting

4.2 How to RUN/STOP the Built-in PLC Function from Outside (Remote RUN/STOP)

The built-in PLC function is RUN/STOPped by shorting/opening SQ-SD.

Remote RUN/STOP is to RUN/STOP the built-in PLC function from outside the inverter with SQ-SD shorted (RUN status).

(1) Applications of remote RUN/STOP

In the following cases, the function can be RUN/STOPped by remote operation using remote RUN/STOP.

1) When the inverter is out of reach.

2) When the inverter in a control box is RUN/STOPped from outside the control box.

(2) Operation performed at remote RUN/STOP

The operation of the sequence program for performing remote RUN/STOP is as described below.

- Remote STOP The function enters the STOP status after the sequence program is executed up to the END instruction.

- Remote RUN When remote RUN is performed after the function has been put in the "STOP status" by remote STOP, the function enters the RUN status again and executes the sequence program from step 0.

(3) Remote RUN/STOP method

There are the following remote RUN/STOP methods.

1) Setting using built-in PLC function parameter (using contact)

Remote RUN/STOP can be performed by turning the remote RUN contact off/on. For example, this method can be used to STOP the PLC function with the emergency stop contact.

- When the remote RUN contact turns off, the function enters the "STOP" status.
- When the remote RUN contact turns on, the function enters the "RUN" status.

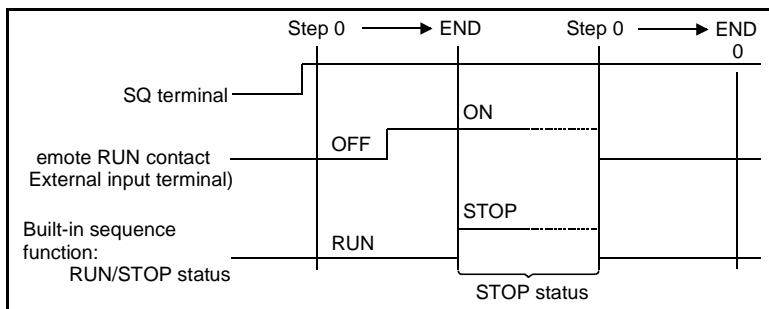


Fig 4.1 Timing Chart for RUN/STOP Using Remote RUN Contact

POINT

Setting of remote RUN contact built-in PLC function parameter X0 to X1F can be set as the remote RUN contacts. (Refer to the GX Developer manual for details.)

2) Method using GX Developer

RUN/STOP can be performed by remote RUN/STOP operation from GX Developer.

For example, this method can be used to STOP the function for sequence program write in a place where the inverter is out of reach.

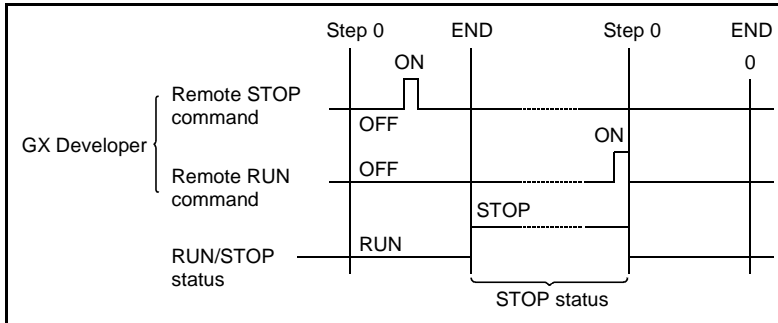


Fig 4.2 Timing Chart for RUN/STOP Using GX Developer

(4) Instructions

Note the following points since the built-in PLC function gives priority to STOP.

- The built-in PLC function enters the STOP status when remote STOP is performed from any of the remote RUN contact, GX Developer, etc.
- To place the built-in PLC function in the RUN status again after it has been put in the STOP status by remote STOP, all external factors (remote RUN contact, GX Developer, etc.) for remote STOP must be set to RUN.

REMARKS

What are RUN and STOP statuses?

- RUN status..... Status where a sequence program is repeating operation from step 0 to END instruction.
- STOP status Status where sequence program operation is at a stop and the outputs (Y) are all off.

4.3 Watchdog Timer (Operation clog up monitor timer)

(1) Watchdog timer

A watchdog timer is the internal timer of the built-in PLC function designed to detect hardware or sequence program faults.

Its default value is set to 200ms.

(2) Watchdog timer resetting

The built-in PLC function resets the watchdog timer before execution of step 0 (after execution of END processing).

When the built-in PLC function operates properly and the END instruction is executed within the setting in the sequence program, the watchdog timer does not time out.

If the hardware fault of the built-in PLC function occurs or the scan time of the sequence program is too long to execute the END instruction within the setting, the watchdog timer times out.

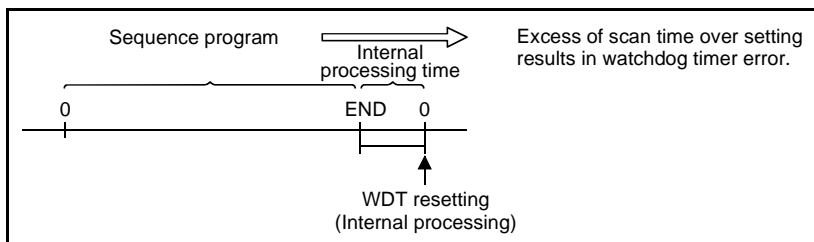


Fig 4.3 Watchdog Timer Resetting

(3) Processing performed when watchdog timer times out

If the scan time exceeds the watchdog timer setting, a watchdog timer error occurs and:

- 1) The built-in PLC function turns off all outputs.
- 2) The P.RUN LED goes off or flickers.
- 3) M9008 turns on and the error code is stored into D9008.

REMARKS

The watchdog timer setting can be changed by built-in PLC function parameter setting of GX Developer. (Refer to the GX Developer manual for details.)

4.4 Self-diagnostic Function

The self-diagnostic function diagnoses faults by the built-in PLC function itself.

(1) Self-diagnostic timing

The self-diagnostic function is performed at power-on, at reset, at execution of any instruction, or at execution of the END instruction.

1) At power-on, at reset

Whether operation can be executed or not is diagnosed.

2) At execution of any instruction

An error occurs if the operation of any instruction in the sequence program is not executed properly.

CAUTION

For the LD, AND, OR, logical comparison operation, and OUT instructions, the set devices are always checked. For the other instructions (SET, RST, MOV, etc.), a check is made as soon as the execution condition holds and the instruction is ready to be executed.

3) At execution of END instruction

Operation clog up monitor timer

(2) Operation mode at fault detection

There are two different PLC operation modes at detection of a fault by the self-diagnostic: operation stop mode and operation continuation mode.

The operation continuation mode includes a fault that enables operation to be stopped by built-in PLC function parameter setting. (Refer to page 39)

1) If an operation stop error is detected by the self-diagnostic, operation is stopped and outputs (Y) are all turned off as soon as the error is detected. The other devices maintain their states at occurrence of the error.

2) If an operation continuation error is detected, only the faulty program part is not executed and the program at the next step is executed.

(3) Error definition checking

When M9008 (self-diagnostic error) turns on at detection of an error, the error code is stored into D9008 (self-diagnostic error). Especially in the continuation mode, use it in the program to prevent a mechanical system malfunction.

For the errors detected by the self-diagnostic, refer to the error code list on page 98.

4.4.1 Error-time operation mode

The built-in PLC function allows you to set whether the sequence program operation will be stopped or continued at occurrence of an operation error.

Use the built-in PLC function parameter to set whether operation will be stopped or continued.

- Default value of error-time operation mode

The following table indicates the default value (initial value) of the error-time operation mode and the status of the built-in PLC function.

Table 4.1 Error-time Operation Mode

Error Definition		CPU Status				
		Operation	P.RUN LED	Special relays turned on	Special registers for data storage	Self- diagnostic error No. (D9008)
		Default value				
Operation error	An error occurred in the sequence program, e.g. an attempt was made to make BCD conversion of any value outside the range 0 to 9999 (or 0 to 99999999).	Continuation	Flicker	M9010 M9011	D9010 D9011	50

4.5 Keyword Registration

The keyword is designed to inhibit the read and rewrite of the program and comments in the built-in PLC function using GX Developer.

- (1) Read/write from built-in PLC function where keyword has been registered
When the keyword has been registered, the built-in PLC function parameters, main program and comments cannot be read/written from the built-in PLC function to the GX Developer device unless the keyword registered to the built-in PLC function is entered.
- (2) Registration and cancel of keyword
A keyword of up to six digits can be set in hexadecimal (0 to 9, A to F).
Make built-in PLC function parameter setting to register or cancel the keyword.

4.6 Setting of Output (Y) Status at Switching from STOP Status to RUN Status

When the RUN status is switched to the STOP status, the outputs (Y) in the RUN status are stored into the built-in PLC function.

Using the built-in PLC function parameter, you can set whether the outputs (Y) will be output again or will be output after execution of operation when the STOP status is switched to the RUN status.

"Output (Y) status at STOP is output"

The sequence program operation is performed after the output (Y) status at the time of entering the STOP status is output.

"Outputs (Y) are cleared (output one scan later)"

The outputs (Y) are all cleared, and after execution of the sequence program operation, the outputs are provided.

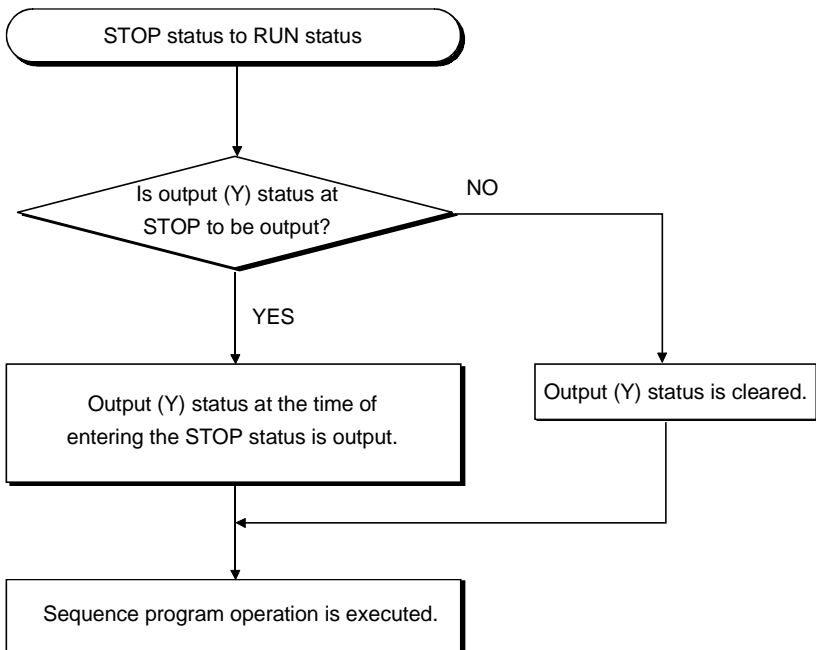


Fig 4.4 Processing Performed when STOP Status Is Switched to RUN Status

MEMO

5. STRUCTURES OF INSTRUCTIONS

5.1	Instruction Format.....	44
5.2	Bit Device Processing Method.....	46
5.3	Handling of Numerical Value.....	48
5.4	Operation Error.....	49

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

5.1 Instruction Format

(1) Many of the instructions can be divided into an instruction part and a device, and their applications are as described below.

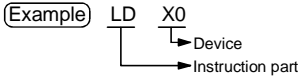
- { Instruction part Indicates the function of that instruction.
- { Device Indicates the data used with the instruction.

(2) The instruction format can be roughly classified as follows according to the instruction part and device combinations.

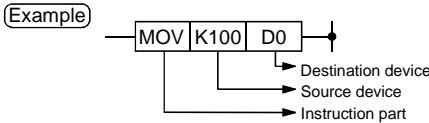
1) Instruction partThis instruction does not change the device status and mainly controls the program.

Example END

2) Instruction part + DeviceThis instruction performs ON/OFF control of the device, controls the execution condition according to the ON/OFF status of the device, and branches the program.



3) Instruction part + Source device + Destination device This instruction performs operation using the data of the destination and source, and stores the operation result into the destination.



4) OthersCombinations other than the above 1) to 3).

(3) Source (S)

The source contains the data to be used for operation.
The data changes depending on the specified device.

- Constant.....Specify the numerical value to be used for operation. Since this value is set at the time of program creation, it is fixed and cannot be changed during program execution.
- Bit device.....Specify the device that stores the data to be used for operation. Therefore, the data must have been stored into the specified device until operation is executed. By changing the data stored into the specified device during program execution, the data used for that instruction can be changed.
- Word device

(4) Destination (D)

The destination stores the data resulting from operation. Note that if the format consists of Instruction part + Source device + Destination device , the data to be used for operation must have been stored into the destination before operation.

At the destination, always specify the device for storing data.

REMARKS

- In this manual, the source and destination are abbreviated as follows.

Source..... (S)

Source 1..... (S1)

Source 2..... (S2)

Destination..... (D)

Destination 1..... (D1)

5.2 Bit Device Processing Method

As the processing method when the bit device (X, Y, M) is specified, 1-bit processing and 16-bit processing using digit designation processing are available.

5.2.1 1-bit processing

When a PLC instruction is used, the device used as the target of operation processing is one bit (one point) of bit device, and multiple bits cannot be specified.

Example LD XO,OUT

5.2.2 Digit designation processing

When a basic or application instruction is used, the bit device used as the target of operation processing may have to be specified by digit designation. When the instruction whose processing unit is 16 bits is specified by this digit designation, up to 16 points can be specified in units of four points.

(1) 16-bit instruction: K1 to 4 (4 to 16 points)

Example Setting ranges of 16-bit data, X0 to F, by digit designation

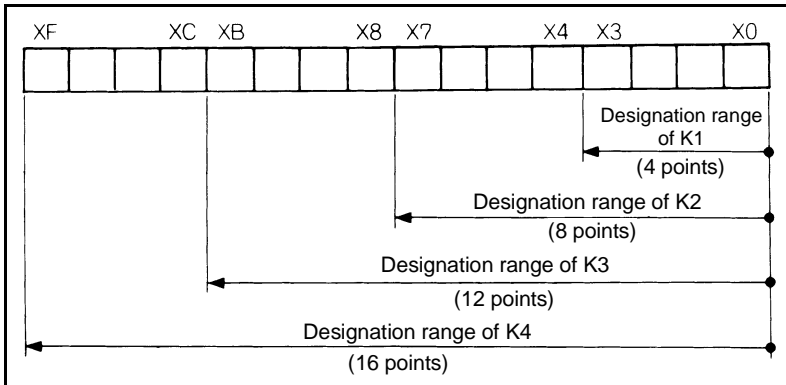


Fig 5.1 Digit Designation Setting Range for 16-bit Instruction

(a) When there is digit designation on the source (S) side, the numerical values that can be handled as the source data are as indicated in Table 5.1.

Table 5.1 List of Designated Digits and Numerical Values That Can Be Handled

Number of Designated Digits	16-bit Instruction
K1 (4 points)	0 to 15
K2 (8 points)	0 to 255
K3 (12 points)	0 to 4095
K4 (16 points)	-32768 to 32767

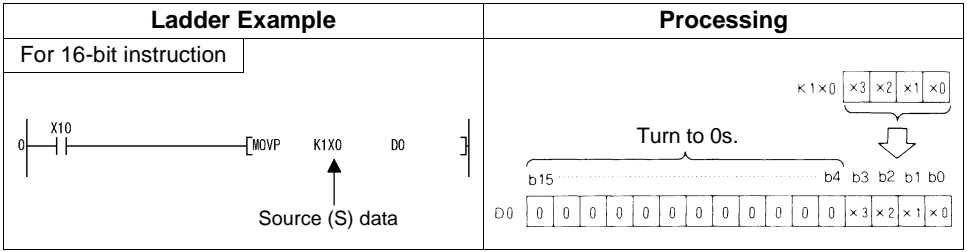


Fig 5.2 Ladder Example and Processing

(b) When there is digit designation on the destination (D) side, the number of points specified by digit designation is the target on the destination side.

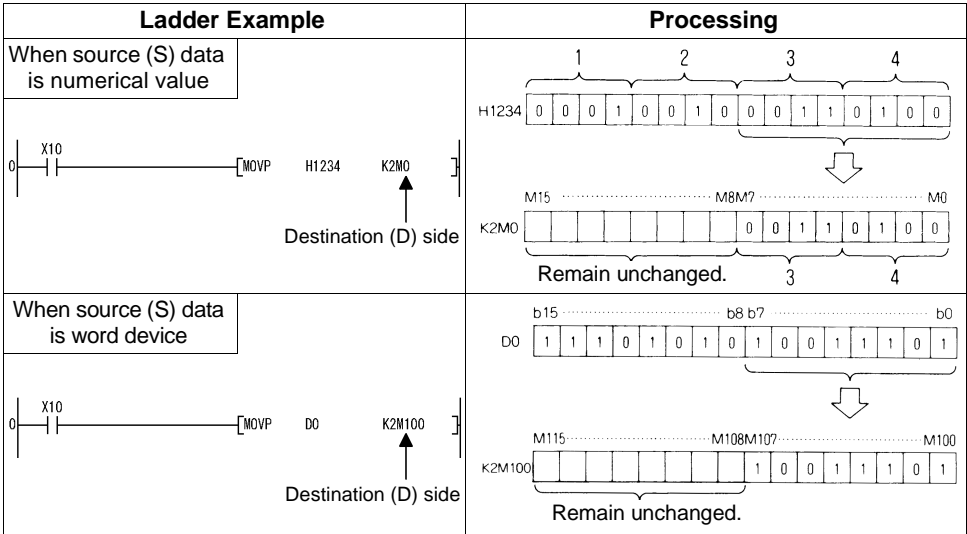


Fig 5.3 Ladder Examples and Processinge

5.3 Handling of Numerical Value

The built-in PLC function has instructions that handle numerical values indicated in 16 bits.

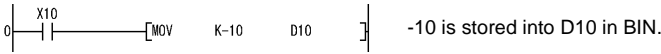
The most significant bit of the 16 bits is used to judge whether the value is positive or negative. Therefore, the numerical values that can be handled as 16 bits are as follows.

16 bits: -32768 to 32767

POINT

• Numerical value setting method

1) Decimal number



2) Hexadecimal number



The decimal notation and hexadecimal notation correspond as indicated below.

Decimal Notation	Hexadecimal Notation
32767	H7FFF
to	to
5	H0005
4	H0004
3	H0003
2	H0002
1	H0001
0	H0000
-1	HFFFF
-2	HFFFE
-3	HFFFD
-4	HFFFC
-5	HFFFB
to	to
-32768	H8000

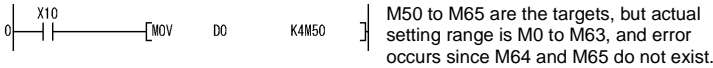
5.4 Operation Error

When a basic instruction is used, an operation error will occur in the following case.

- (a) If any error described in the description of the corresponding instruction occurs.

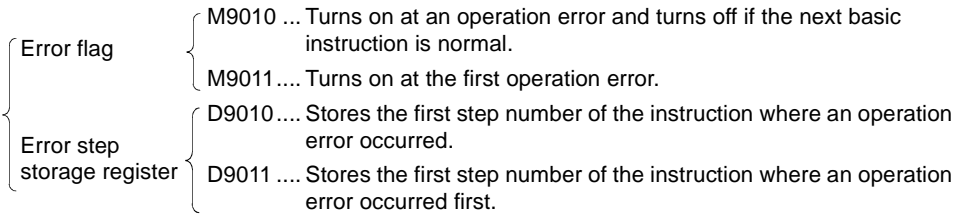
POINT

Note that if the device designation range is outside the corresponding device range, an operation error does not occur and data is written to other than the specified device.



(1) Error processing

If an operation error occurred at execution of a basic instruction, the error flag turns on and the error step number is stored into the error step storage register.



- 1) D9011 stores the step number of the instruction where an operation error occurred when M9011 turned from OFF to ON. Therefore, D9011 data does not change if M9011 remains on.

- 2) To reset M9011 and D9011, program as shown below.

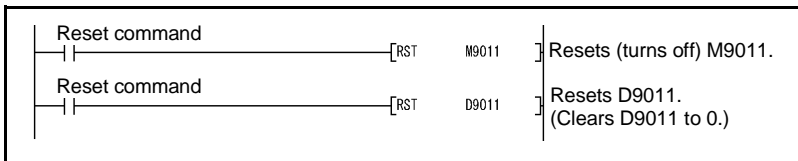


Fig 5.4 Special Relay and Register Resetting Ladder

- 3) Whether sequence processing will be stopped or continued at occurrence of an operation error can be selected by built-in PLC function parameter setting. Refer to page 39 for details.

MEMO

6. PLC INSTRUCTIONS

6.1	PLC Instructions.....	52
6.2	Description of the Instructions	57
6.3	Contact Instructions	58
6.4	Connection Instructions	61
6.5	Output Instructions	67
6.6	Shift Instructions	75
6.7	Master Control Instructions	77
6.8	End Instruction	81
6.9	Other Instructions	82
6.10	Comparison Operation Instructions	84
6.11	Data Transfer Instructions	88
6.12	16-bit Logical Product ... WAND, WANDP	90
6.13	16-bit Logical Add ... WOR, WORP	93

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

6.1 PLC Instructions

6.1.1 How to use the instruction list

Classification	Instruction Symbol	Symbol	Processing	Execution Condition	Number of Steps
Transfer	MOV		(S) → (D)		5
	MOVP				
↑ 1)	↑ 2)	↑ 3)	↑ 4)	↑ 5)	↑ 6)

- 1).....Classifies the instruction by application.
- 2).....Indicates the instruction symbol used for programming.

The instructions are based on 16-bit data instructions.

Example MOV



16-bit transfer instruction

- Add P to the end of the instruction to define it as executed only on the leading edge of the preceding condition.

Example MOV



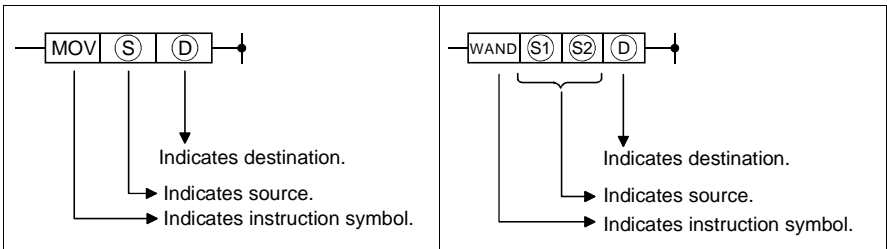
MOVP



Instruction executed continuously while preceding condition is on

Instruction executed only on leading edge of preceding contact condition

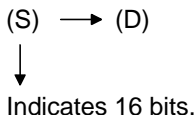
- 3).....Indicates the symbol used in the ladder diagram.







Destination: Indicates the destination of the operation result.

Source:..... Indicates the source of the data for the operation.

4)..... Indicates the operation.



5)..... Indicates the condition of execution for each instruction as described below:

Symbol	Execution Condition
No entry	The instruction is always executed independently of whether its preceding condition is on or off. When the preceding condition is off, the instruction is off.
	The instruction is executed continuously only while its preceding condition is on. When the preceding condition is off, the instruction is not executed and not processed.
	The instruction is executed once only when the preceding condition turns from off to on. If the condition remains on after that, the instruction is not executed and not processed.
	The instruction is executed continuously only while its preceding condition is off. When the preceding condition is on, the instruction is not executed and not processed.
	The instruction is executed once only when the preceding condition turns from on to off. If the condition remains off after that, the instruction is not executed and not processed.

6)..... Indicates the number of program steps required for each instruction.
 The number of steps that changes depending on conditions is two.

6.1.2 PLC instruction list

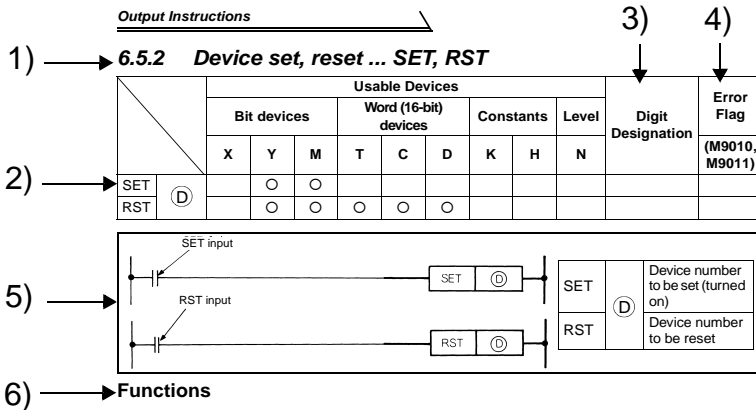
Classification	Instruction Symbol	Symbol	Processing	Execution Condition	Number of Steps	Reference page
Contacts	LD		Logical operation start (Operation start at N/O contact)		1	58
	LDI		Logical NOT operation start (Operation start at N/C contact)		1	
	AND		Logical product (N/O contact series connection)		1	
	ANI		Logical product NOT (N/C contact series connection)		1	
	OR		Logical sum (N/O contact parallel connection)		1	
	ORI		Logical sum NOT (N/C contact parallel connection)		1	
Connection	ANB		AND between logical blocks (series connection between blocks)		1	61
	ORB		OR between logical blocks (parallel connection between blocks)		1	
	MPS		Stores the operation result.		1	64
	MRD		Reads the operation result stored in MPS.		1	
	MPP		Reads and resets the operation result stored in MPS.		1	
Outputs	OUT		Outputs device.		1 3	67
	SET		Sets device.		1 3	
	RST		Resets device.		1 3	
	PLS		Produces a pulse lasting one program scan time on the leading edge of input signal.		3	73
	PLF		Produces a pulse lasting one program scan time on the trailing edge of input signal.		3	
Shift	SFT		1-bit device shift		3	75
	SFTP				3	

Classification	Instruction Symbol	Symbol	Processing	Execution Condition	Number of Steps	Reference page
Master control	MC		Master control start		5	77
	MCR		Master control reset		3	
Program end	END	—	Must be written at the end of sequence program to return to step 0.		1	81
No operation	NOP	—	No operation For program deletion or space		1	82
	NOPLF	—	No operation Line feed instruction for printer output		1	-
16-bit data comparison	LD=		Continuity when (S1) = (S2) Non-continuity when (S1) ≠ (S2)		5	84
	AND=			 Continuity when (S1) = (S2) Non-continuity when (S1) ≠ (S2)	5	
	OR=			 Continuity when (S1) = (S2) Non-continuity when (S1) ≠ (S2)	5	
	LD<>		Continuity when (S1) ≠ (S2) Non-continuity when (S1) = (S2)		5	
	AND<>			 Continuity when (S1) ≠ (S2) Non-continuity when (S1) = (S2)	5	
	OR<>			 Continuity when (S1) ≠ (S2) Non-continuity when (S1) = (S2)	5	
	LD>		Continuity when (S1) > (S2) Non-continuity when (S1) ≤ (S2)		5	
	AND>			 Continuity when (S1) > (S2) Non-continuity when (S1) ≤ (S2)	5	
	OR>			 Continuity when (S1) > (S2) Non-continuity when (S1) ≤ (S2)	5	
	LD<=		Continuity when (S1) ≤ (S2) Non-continuity when (S1) > (S2)	 Continuity when (S1) ≤ (S2) Non-continuity when (S1) > (S2)	5	
	AND<=			 Continuity when (S1) ≤ (S2) Non-continuity when (S1) > (S2)	5	
	OR<=			 Continuity when (S1) ≤ (S2) Non-continuity when (S1) > (S2)	5	
	LD<		Continuity when (S1) < (S2) Non-continuity when (S1) ≥ (S2)		5	
	AND<			 Continuity when (S1) < (S2) Non-continuity when (S1) ≥ (S2)	5	
OR<		 Continuity when (S1) < (S2) Non-continuity when (S1) ≥ (S2)		5		

Classification	Instruction Symbol	Symbol	Processing	Executi	Reference page	
				on Condi		Number of Steps
				tion		
16-bit data comparison	LD>=		Continuity when (S1) ≥ (S2) Non-continuity when (S1) < (S2)		5	84
	AND>=				5	
	OR>=				5	
Transfer	MOV		(S) → (D)		5	88
	MOVP				5	
Logical product	WAND		(D) ∧ (S) → (D)		5	90
	WANDP				5	
	WAND		(S1) ∧ (S2) → (D)		7	
	WANDP				7	
Logical sum	WOR		(D) ∨ (S) → (D)		5	93
	WORP				5	
	WOR		(S1) ∨ (S2) → (D)		7	
	WORP				7	

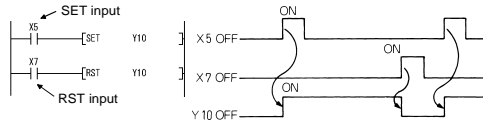
6.2 Description of the Instructions

In Chapter 6, the instructions are described in the following format.



SET

- Turns on the specified device when the SET input turns on.
- The device turned on is held on if the SET input turns off. It can be turned off by the RST instruction.



7) → Execution Conditions

The SET and RST instructions are executed every scan.

8) → Program Examples

SET, **RST**

- Program that sets (turns on) Y8 when X8 turns on and resets (turns off) Y8 when X9 turns on.



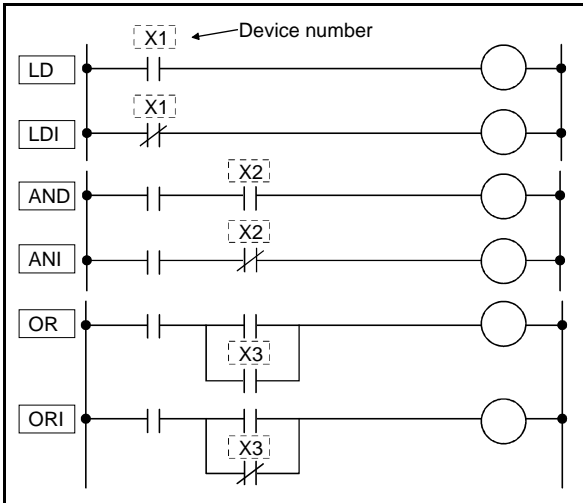
Description

- Indicates the section number, instruction outlines and instruction symbols.
- The devices usable with the instructions are marked.
- The digit designation that can be set is indicated for the instruction that requires digit designation when a bit device is used.
- The instruction for which the error flag turns on at operation error occurrence is marked.
- Shows the format in the ladder mode.
- Explains the instruction.
- Indicates the execution conditions of the instructions.
- Shows program examples in the ladder mode and list mode.

6.3 Contact Instructions

6.3.1 Operation start, series connection, parallel connection ... LD, LDI, AND, ANI, OR, ORI

Usable Devices									Digit Designation	Error Flag (M9010,M9011)
Bit devices			Word (16-bit) devices			Constants		Level		
X	Y	M	T	C	D	K	H	N		
○	○	○	○	○	○					



Functions

LD, LDI

- (1) LD is an N/O contact operation start instruction, and LDI is an N/C contact operation start instruction. Each of them imports the ON/OFF data of the specified device and uses it as an operation result.

AND, ANI

- (1) AND is an N/O contact series connection instruction, and ANI is an N/C contact series connection instruction. Each of them imports the ON/OFF data of the specified device, ANDs it with the previous operation result, and uses the resultant value as an operation result.
- (2) There are no restrictions on the use of AND and ANI, but there are the following conditions in the ladder mode.
 - 1) WriteWhen contacts are connected in series by AND or ANI, a ladder of up to 21 contacts can be created.
 - 2) ReadWhen contacts are connected in series by AND or ANI, a ladder of up to 24 contacts can be displayed. If the ladder has more than 24 contacts, up to 24 contacts are displayed.

OR, ORI

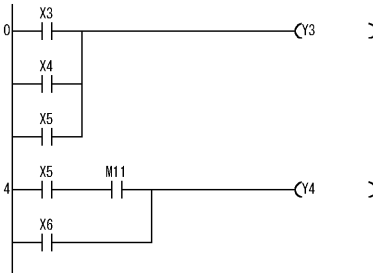
- (1) OR is an N/O contact parallel connection instruction, and ORI is an N/C contact parallel connection instruction. Each of them imports the ON/OFF data of the specified device, ORs it with the previous operation result, and uses the resultant value as an operation result.
- (2) There are no restrictions on the use of OR and ORI, but there are the following conditions in the ladder mode.
 - 1) WriteA ladder of up to 23 contacts connected consecutively by OR or ORI can be created.
 - 2) ReadA ladder of up to 23 contacts connected consecutively by OR or ORI can be displayed. If the ladder has more than 23 contacts, it cannot be displayed properly.

Execution Conditions

Executed every scan independently of the device ON/OFF and preceding operation result.

Program Examples

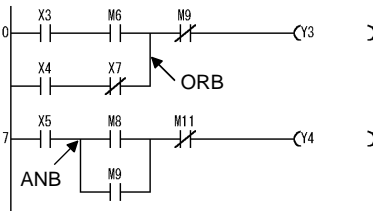
LD , LDI , AND , ANI , OR , ORI



· Coding

```

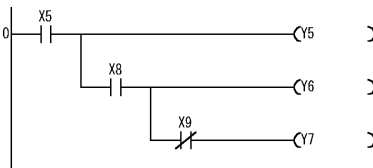
0 LD      X3
1 OR      X4
2 OR      X5
3 OUT     Y3
4 LD      X5
5 AND     M11
6 OR      X6
7 OUT     Y4
8 END
    
```



· Coding

```

0 LD      X3
1 AND     M6
2 LD      X4
3 ANI     X7
4 ORB
5 ANI     M9
6 OUT     Y3
7 LD      X5
8 LD      M8
9 OR      M9
10 ANB
11 ANI    M11
12 OUT    Y4
13 END
    
```



· Coding

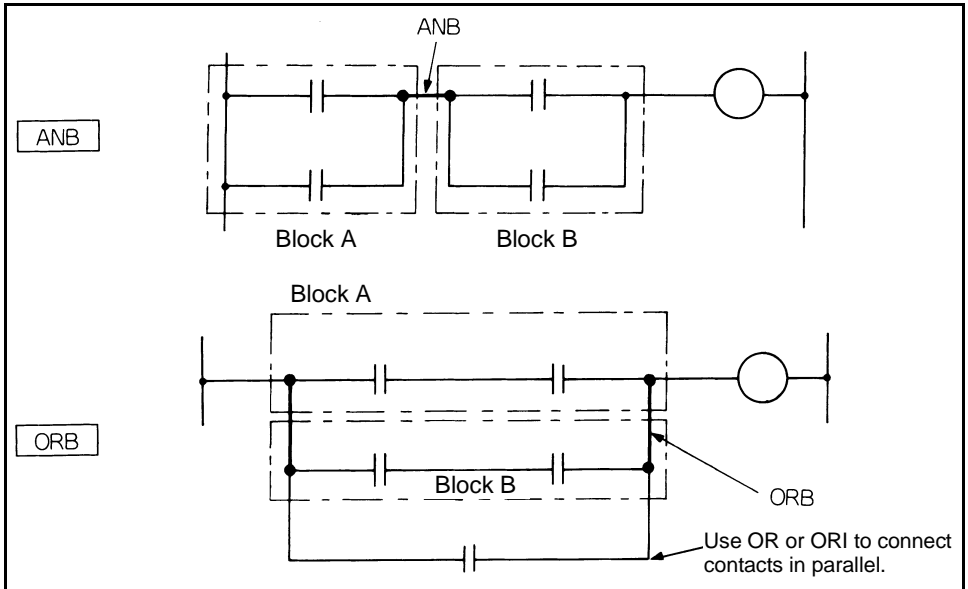
```

0 LD      X5
1 OUT     Y5
2 AND     X8
3 OUT     Y6
4 ANI     X9
5 OUT     Y7
6 END
    
```


6.4 Connection Instructions

6.4.1 Ladder block series connection, parallel connection ... ANB, ORB

Usable Devices									Digit Designation	Error Flag (M9010, M9011)
Bit devices			Word (16-bit) devices			Constants		Level		
X	Y	M	T	C	D	K	H	N		



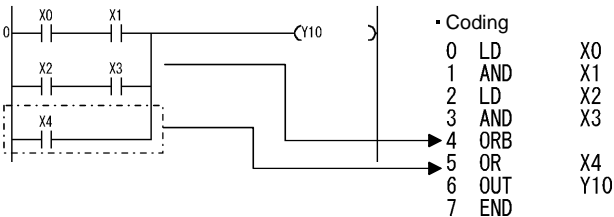
Functions

ANB

- (1) ANDs blocks A and B and uses the resultant value as an operation result.
- (2) The symbol of ANB is not a contact symbol but a connection symbol.
- (3) ANB can be written up to seven instructions (eight blocks) consecutively.
If ANB is written consecutively more than the above, the PLC cannot perform normal operation.

ORB

- (1) ORs blocks A and B and uses the resultant value as an operation result.
- (2) ORB connects in parallel the ladder blocks of two or more contacts. Use OR or ORI to connect in parallel the ladder blocks of only one contact.

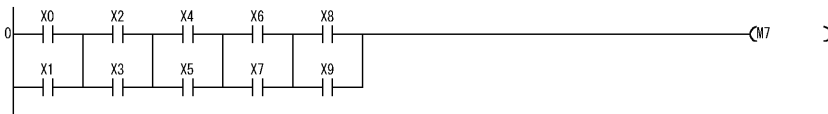


- (3) The symbol of ORB is not a contact symbol but a connection symbol.
- (4) ORB can be written up to seven instructions (eight blocks) consecutively.
If ORB is written consecutively more than the above, the PLC cannot perform normal operation.

Program Examples

ANB

Though there are the following two different program coding methods for connecting ladder blocks in series consecutively, use the coding example 1.



• Coding example 1

```

0 LD X0
1 OR X1
2 LD X2
3 OR X3
4 ANB
5 LD X4
6 OR X5
7 ANB
8 LD X6
9 OR X7
10 ANB
11 LD X8
12 OR X9
13 ANB
14 OUT M7
15 END
    
```

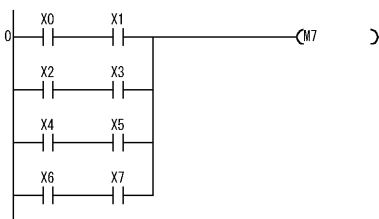
• Coding example 2

```

0 LD X0
1 OR X1
2 LD X2
3 OR X3
4 LD X4
5 OR X5
6 LD X6
7 OR X7
8 LD X8
9 OR X9
10 ANB
11 ANB
12 ANB
13 ANB
14 OUT M7
15 END
    
```

ORB

Though there are the following two different program coding methods for connecting ladder blocks in parallel consecutively, use the coding example 1.



• Coding example 1

```

0 LD X0
1 AND X1
2 LD X2
3 AND X3
4 ORB
5 LD X4
6 AND X5
7 ORB
8 LD X6
9 AND X7
10 ORB
11 OUT M7
12 END
    
```

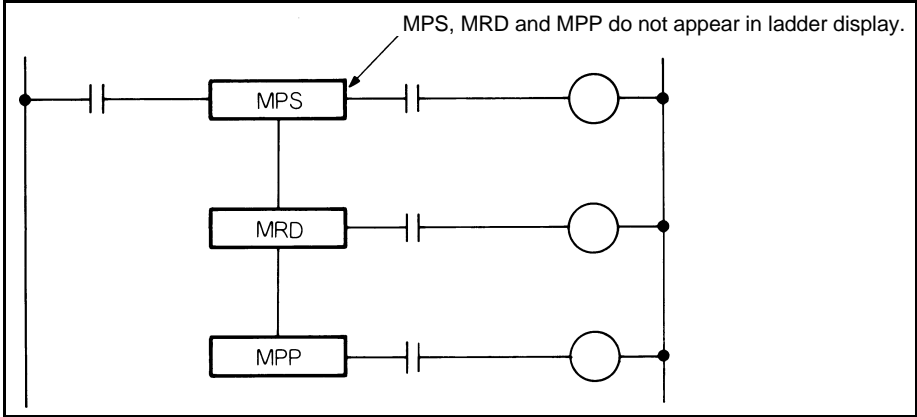
• Coding example 2

```

0 LD X0
1 AND X1
2 LD X2
3 AND X3
4 LD X4
5 AND X5
6 LD X6
7 AND X7
8 ORB
9 ORB
10 ORB
11 OUT M7
12 END
    
```

6.4.2 Operation result, push, read, pop ... MPS, MRD, MPP

Usable Devices									Digit Designation	Error Flag (M9010,M9011)
Bit devices			Word (16-bit) devices			Constants		Level		
X	Y	M	T	C	D	K	H	N		



Functions

MPS

- (1) Stores the operation result (ON/OFF) immediately before itself.
- (2) The MPS instruction can be used consecutively up to 12 times.
In the ladder mode, however, it can be used up to 11 times.
When the MPP instruction is used midway, the number of used MPS instructions is decremented by 1.

MRD

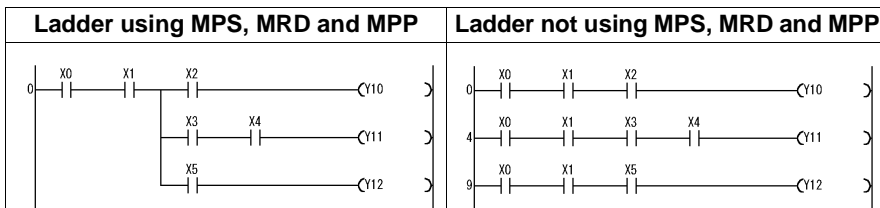
- (1) Reads the operation result stored by the MPS instruction, and continues operation from the next step with that operation result.

MPP

- (1) Reads the operation result stored by the MPS instruction, and continues operation from the next step with that operation result.
- (2) Clears the operation result stored by the MPS instruction.

POINT

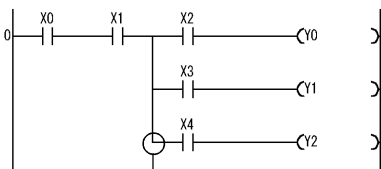
(1) Ladders differ as shown below between when MPS, MRD and MPP are used and when they are not used.



(2) Use the same number of MPS and MPP instructions. If they differ in the number of used instructions, operation will be performed as described below.

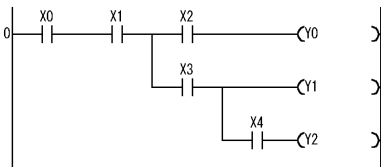
1) If the MPS instructions are used more than MPP instructions, the ladder is changed and the built-in PLC function performs operation according to the new ladder.

Before change



When MPP is replaced by NOP

After change



• Coding

```

0 LD X0
1 AND X1
2 MPS
3 AND X2
4 OUT Y0
5 MRD
6 AND X3
7 OUT Y1
8 MPP
9 AND X4
10 OUT Y2
11 END
    
```

• Coding

```

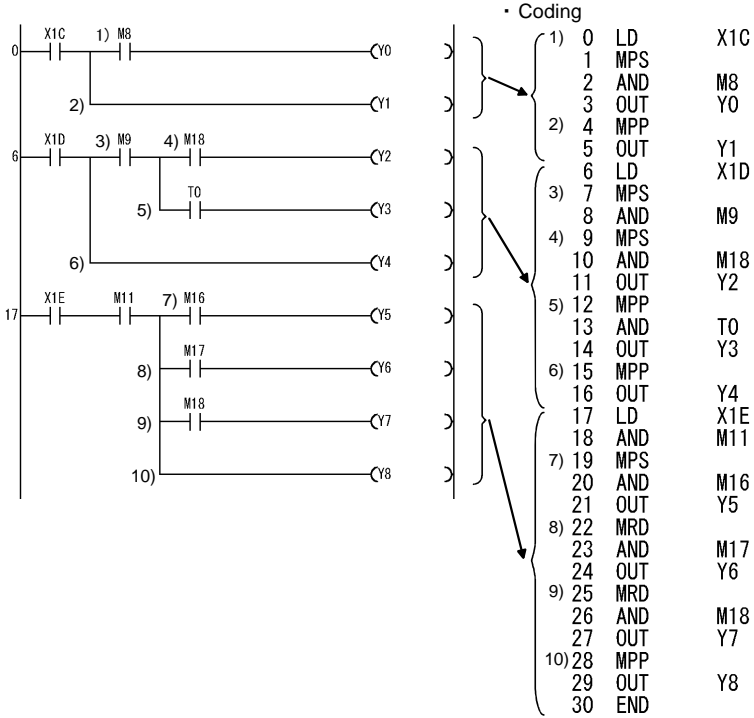
0 LD X0
1 AND X1
2 MPS
3 AND X2
4 OUT Y0
5 MRD
6 AND X3
7 OUT Y1
8 NOP
9 AND X4
10 OUT Y2
11 END
    
```

2) If the MPP instructions are used more than MPS instructions, that ladder block results in a ladder creation error, and the built-in PLC function cannot perform normal operation.

Program Example

MPS , **MRD** , **MPP**

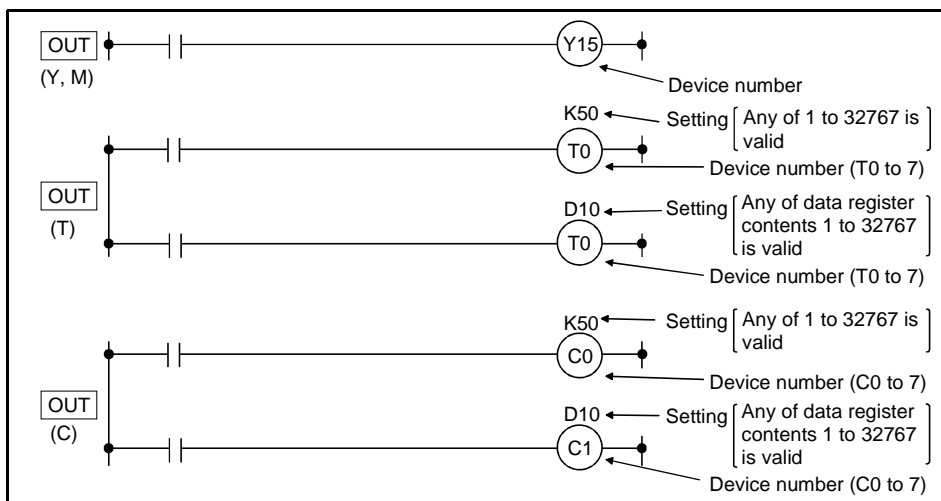
1) Program using MPS, MRD and MPP



6.5 Output Instructions

6.5.1 Bit device, timer, counter ... OUT

	Usable Devices									Digit Designation	Error Flag
	Bit devices			Word (16-bit) devices			Constants		Level		
	X	Y	M	T	C	D	K	H	N		(M9010, M9011)
Bit device		○	○								
Timer	Device			○							
	Setting					○	○				
Counter	Device				○						
	Setting					○	○				



Functions

OUT (Y, M)

(1) Outputs the operation result up to OUT instruction to the specified device.

Operation Result	OUT Instruction		
	Coil	Contacts	
		N/O contact	N/C contact
OFF	OFF	Not energize	Energize
ON	ON	Energize	Not energize

REMARKS

- Three steps are used for the OUT instruction only when the following device is used.
 - Special relay (M)

OUT(T)

- (1) When the operation result up to the OUT instruction is ON, the coil of the timer turns on and the timer times up to the setting, and when the timer times out (timing value \geq setting), the contact operates as indicated below.

N/O contact	Energize
N/C contact	Not energize

- (2) When the operation result up to the OUT instruction turns from ON to OFF, the timer operates as indicated below.

Timer Type	Timer Coil	Present Value of Timer	Before Time-out		After Time-out	
			N/O contact	N/C contact	N/O contact	N/C contact
100ms timer	OFF	0	Not energize	Energize	Not energize	Energize
10ms timer						
100ms retentive timer	OFF	Maintained	Not energize	Energize	Energize	Not energize

- (3) After a time-out, the contact state of the retentive timer remains unchanged until the RST instruction is executed.
- (4) A negative number (-32768 to -1) cannot be specified for the setting.
- (5) If the setting is 0, it is timed as infinity. Hence, the timer does not time out.
- (6) Refer to page 24 for the timing method of the timer.

OUT(C)

- (1) When the operation result up to the OUT instruction turns from OFF to ON, the present value (count value) is incremented by 1, and when the counter stops counting (present value = setting), the contact operates as indicated below.

N/O contact	Energize
N/C contact	Not energize

- (2) The counter does not count if the operation result remains ON. (Count inputs need not be converted into pulses.)
- (3) After the counter has stopped counting, the count value and contact state remain unchanged until the RST instruction is executed.
- (4) A negative number (-32768 to -1) cannot be specified for the setting. If the setting is 0, processing is the same as when the setting is 1.
- (5) Refer to page 26 for the counting method of the counter.

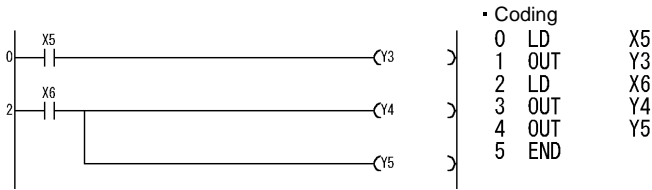
Execution Conditions

Executed every scan independently of the operation result up to the OUT instruction.

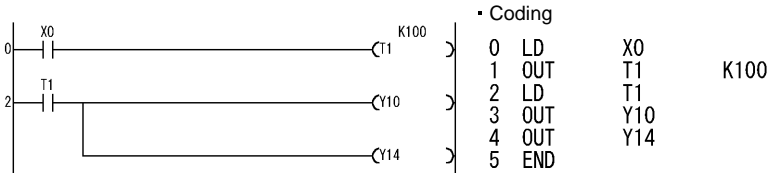
Program Examples

OUT

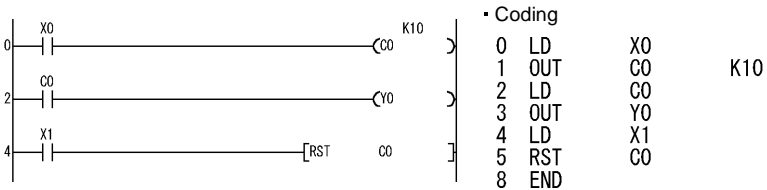
1) Program that outputs to the output module.



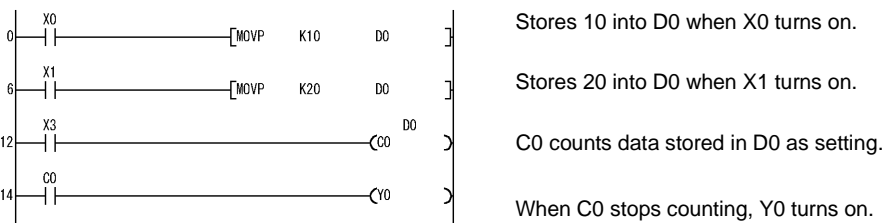
2) Program that turns on Y10 and Y14 10s after X0 has turned on.



3) Program that turns on Y0 when X0 turns on 10 times and turns off Y0 when X1 turns on.



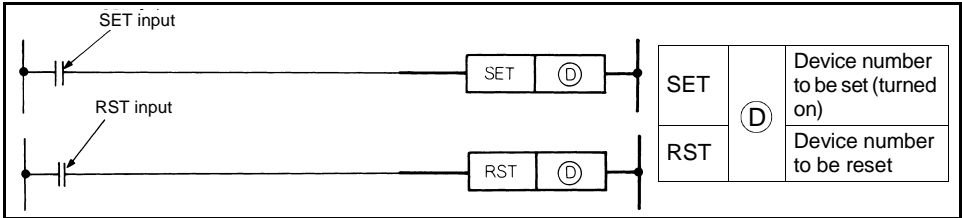
4) Program that changes the C0 setting to 10 when X0 turns on and to 20 when X1 turns on.



Step	Instruction	Address
0	LD	X0
1	MOV	K10 D0
6	LD	X1
7	MOV	K20 D0
12	LD	X3
13	OUT	C0 D0
14	LD	C0
15	OUT	Y0
16	END	

6.5.2 Device set, reset ... SET, RST

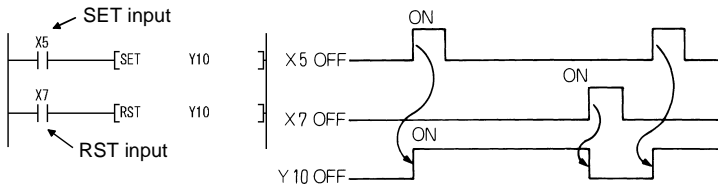
		Usable Devices									Digit Designation	Error Flag (M9010, M9011)
		Bit devices			Word (16-bit) devices			Constants		Level		
		X	Y	M	T	C	D	K	H	N		
SET	Ⓓ		○	○								
RST			○	○	○	○	○					



Functions

SET

- (1) Turns on the specified device when the SET input turns on.
- (2) The device turned on is held on if the SET input turns off. It can be turned off by the RST instruction.



- (3) When the SET input is off, the device status does not change.

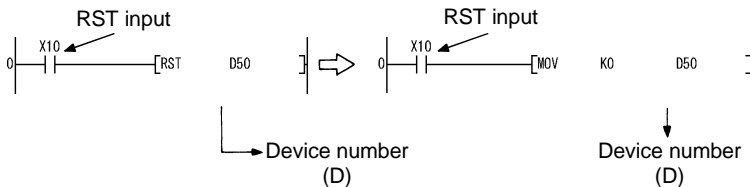
RST

- (1) When the RST input turns on, the specified device operates as described below.

Device	Status
Y, M	The coil and contact are turned off.
T, C	The present value is reset to 0 and the coil and contact are turned off.
D	Cleared to 0.

- (2) When the RST input is off, the device status does not change.

(3) The function of RST (D) is the same as that of the following ladder.



Execution Conditions

The SET and RST instructions are executed every scan.

REMARKS

Three steps are used when the following device is used.

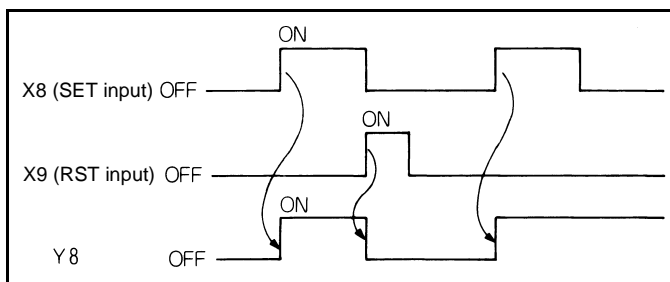
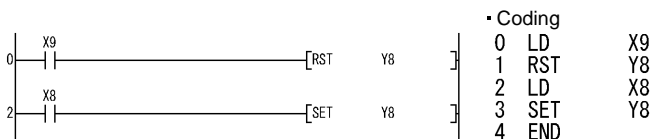
SET instruction ... Special relay (M)

RST instruction ... Special relay (M), all word devices

Program Examples

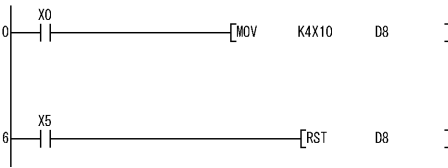
SET , RST

1) Program that sets (turns on) Y8 when X8 turns on and resets (turns off) Y8 when X9 turns on.



Operations of SET and RST instructions

2) Program that resets the data register contents to 0.



Stores X10 to 1F contents into D8 when X0 turns on.

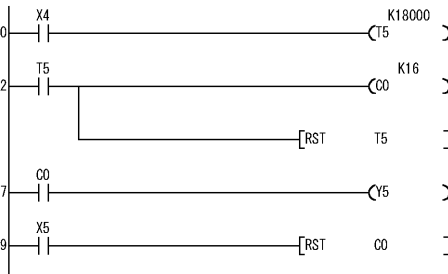
Resets D8 contents to 0 when X5 turns on.

• Coding

```

0 LD X0
1 MOV K4X10 D8
6 LD X5
7 RST D8
10 END
  
```

3) Program that resets the 100ms retentive timer and counter.



When T5 is set as retentive timer, T5 turns on when ON period of X4 reaches 30 minutes.

Counts the number of times T5 turned on.

Resets T5 when T5 turns on.

When C0 stops counting, Y5 turns on.

When X5 turns on, C0 is reset.

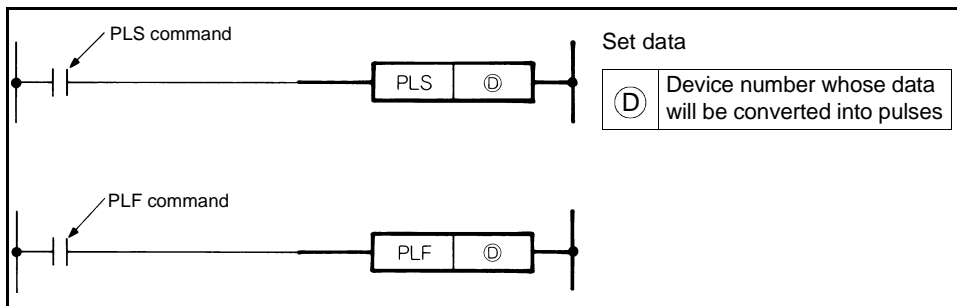
• Coding

```

0 LD X4
1 OUT T5 K18000
2 LD T5
3 OUT C0 K16
4 RST T5
7 LD C0
8 OUT Y5
9 LD X5
10 RST C0
13 END
  
```

6.5.3 Leading edge, trailing edge differential outputs ... PLS, PLF

	Usable Devices									Digit Designation	Error Flag (M9010, M9011)
	Bit devices			Word (16-bit) devices			Constants		Level		
	X	Y	M	T	C	D	K	H	N		
Ⓓ		○	○								



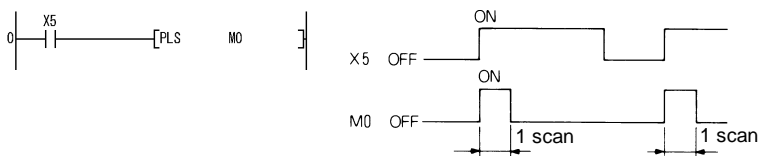
Functions

PLS

- (1) Turns the specified device on when the PLS command turns from OFF to ON, and turns it off except when the PLS command turns from OFF to ON.

When there is one PLS instruction for the device specified at Ⓓ during one scan, the specified device turns on for one scan.

Do not execute the PLS instruction for the same device more than once during one scan.



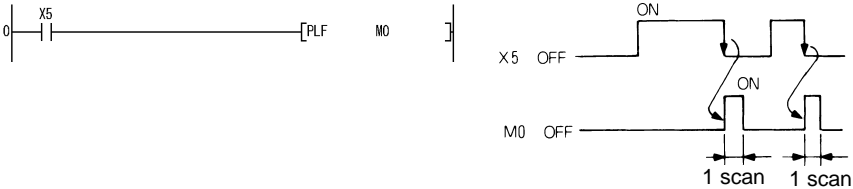
- (2) If the status is switched to STOP and switched to RUN again after execution of the PLS instruction, the PLS instruction is not executed.

PLF

(1) Turns the specified device on one scan when the PLF command turns from ON to OFF, and turns it off except when the PLF command turns from ON to OFF.

When there is one PLF instruction for the device specified at (D) during one scan, the specified device turns on for one scan.

Do not execute the PLF instruction for the same device more than once during one scan.

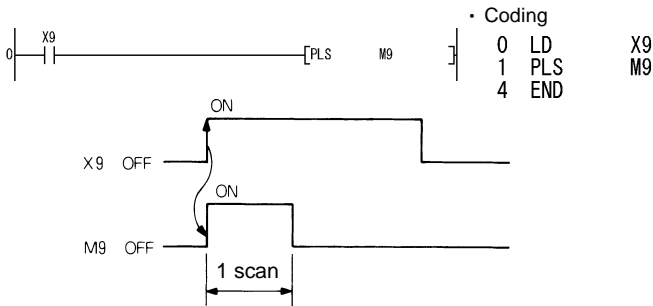


(2) If the status is switched to STOP and switched to RUN again after execution of the PLF instruction, the PLF instruction is not executed.

Program Examples

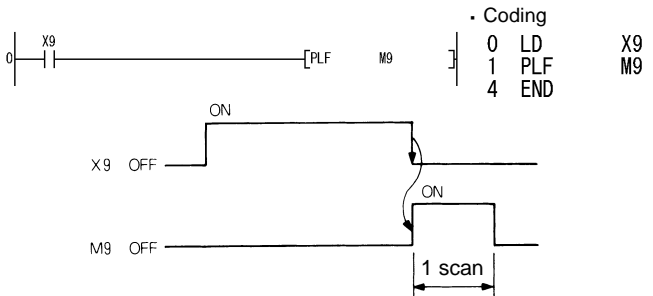
PLS

Program that executes the PLS instruction when X9 turns on.



PLF

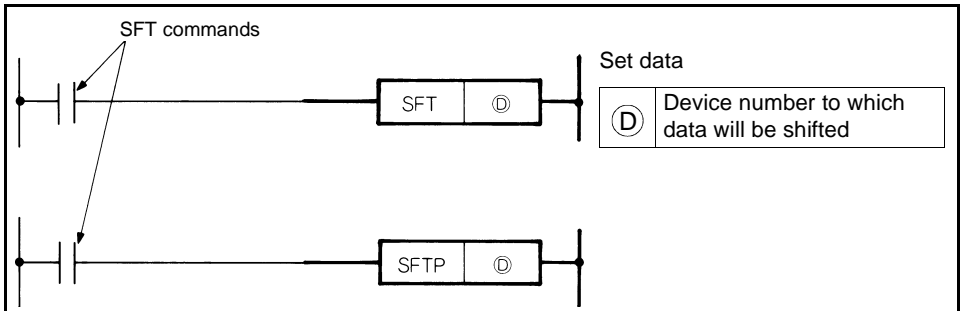
Program that executes the PLF instruction when X9 turns off.



6.6 Shift Instructions

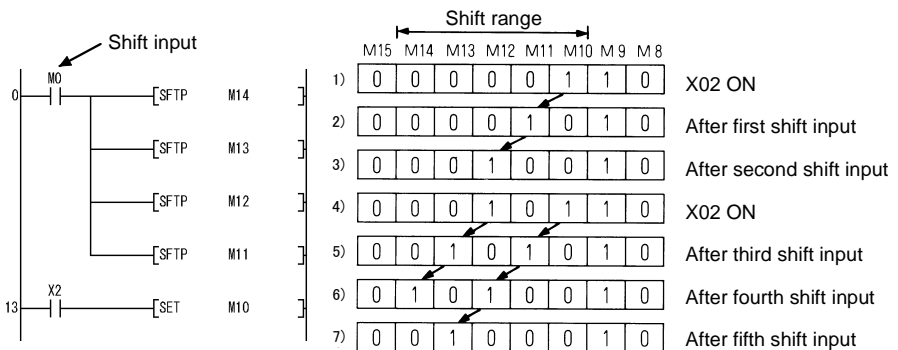
6.6.1 Bit device shift ... SFT, SFTP

	Usable Devices									Digit Designation	Error Flag (M9010, M9011)
	Bit devices			Word (16-bit) devices			Constants		Level		
	X	Y	M	T	C	D	K	H	N		
Ⓓ		○	○								



Functions

- (1) Shifts the ON/OFF status of the device preceding the one specified at Ⓓ to the specified device, and turns off the preceding device.
- (2) Use the SET instruction to turn on the first device from which data will be shifted.
- (3) When using the SFT or SFTP instructions consecutively, program in order of larger to smaller device numbers.

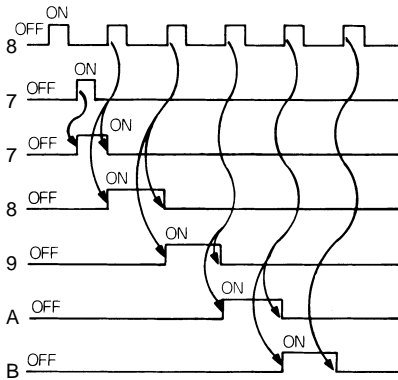
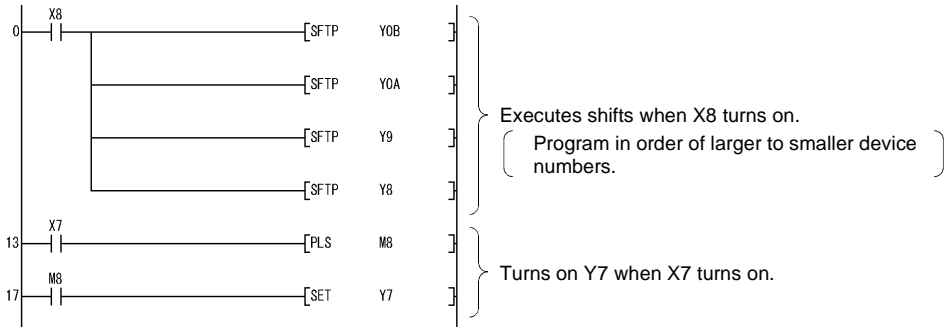


*At M8 to 15, 1 indicates ON and 0 indicates OFF.

Program Example

SFT

1) Program that shifts the Y7 - B data when X8 turns on.



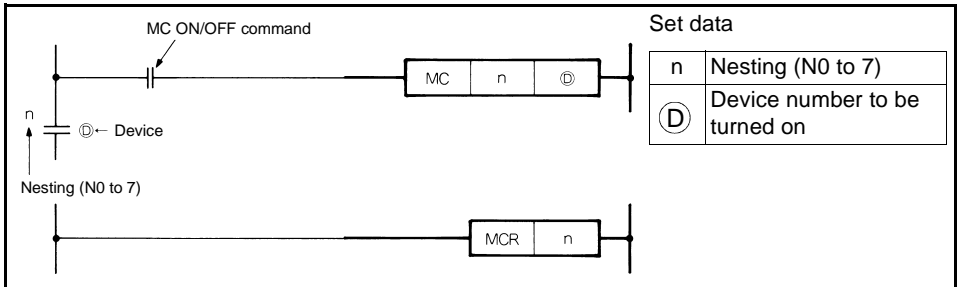
• Coding

0	LD	X8
1	SFTP	Y0B
4	SFTP	Y0A
7	SFTP	Y9
10	SFTP	Y8
13	LD	X7
14	PLS	M8
17	LD	M8
18	SET	Y7
19	END	

6.7 Master Control Instructions

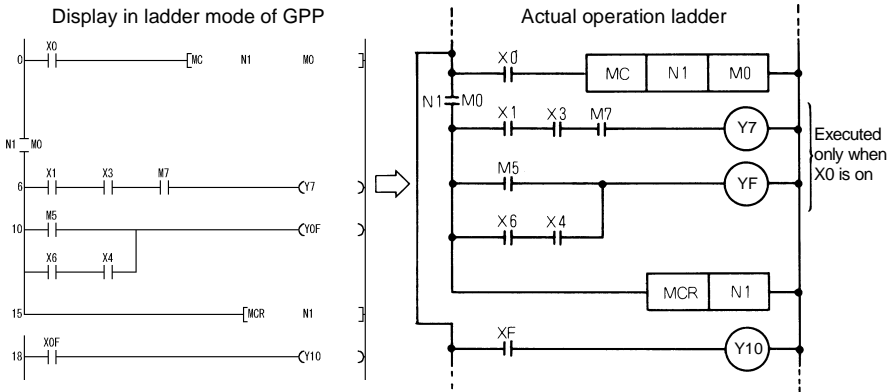
6.7.1 Master control set, reset ... MC, MCR

	Usable Devices									Digit Designation	Error Flag (M9010, M9011)
	Bit devices			Word (16-bit) devices			Constants		Level		
	X	Y	M	T	C	D	K	H	N		
n										○	
ⓓ		○	○								



Functions

- (1) The master control instructions are designed to create an efficient ladder switching sequence program by switching on/off the common bus of the ladder. The ladder that uses master control is as shown below.



MC

- (1) When the MC ON/OFF command is on at the start of master control, the operation results between MC and MCR are as performed by the instructions (ladder).
- (2) If the MC instruction is off, the scan between the MC and MCR instructions is executed, and therefore, the scan time does not become short.
When the MC instruction is off, the operation results between MC and MCR are as described below.

100ms, 10ms timer	The count value is reset to 0 and both the coil and contact turn off.
100ms retentive timer, counter	The coil turns off but both the count value and contact maintain the current states.
Devices in OUT instruction	All turn off.
SET, RST, SFT or device in instruction basic	Maintains the current state.

- (3) By changing the device at (D), the MC instruction can use the same nesting (N) number any number of times.
- (4) When the MC instruction is on, the coil of the device specified at (D) turns on. Since using the same device in the OUT instruction, etc. will result in double coils, the device specified at (D) should not be used in any other instruction.

MCR

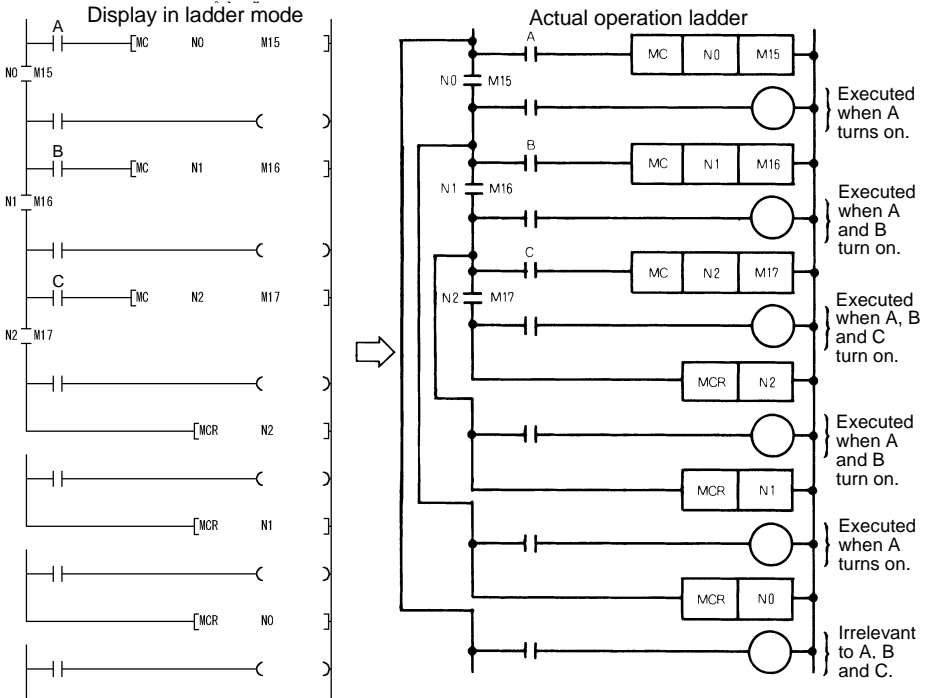
(1) This instruction is designed to reset the master control and indicates the end of the master control range.

(2) Do not provide a contact instruction in front of the MCR instruction.

The master control instructions can be nested. Their master control ranges are differentiated by the nesting (N). The nesting can be used from N0 to N7.

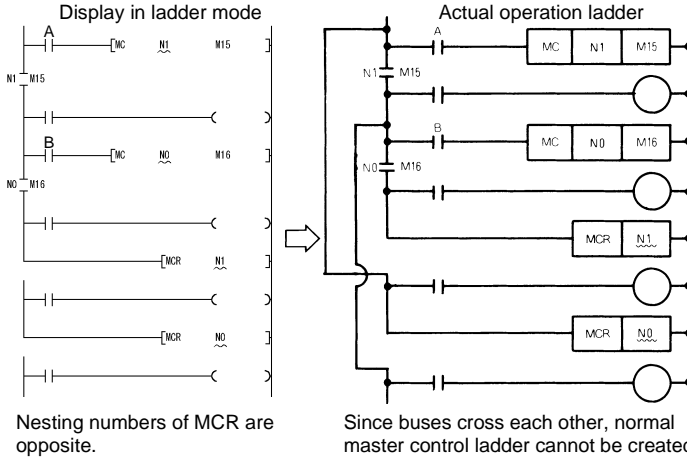
Using the nesting structure, you can create a ladder that restricts the program execution conditions in order.

The ladder using the nesting structure is as shown below.



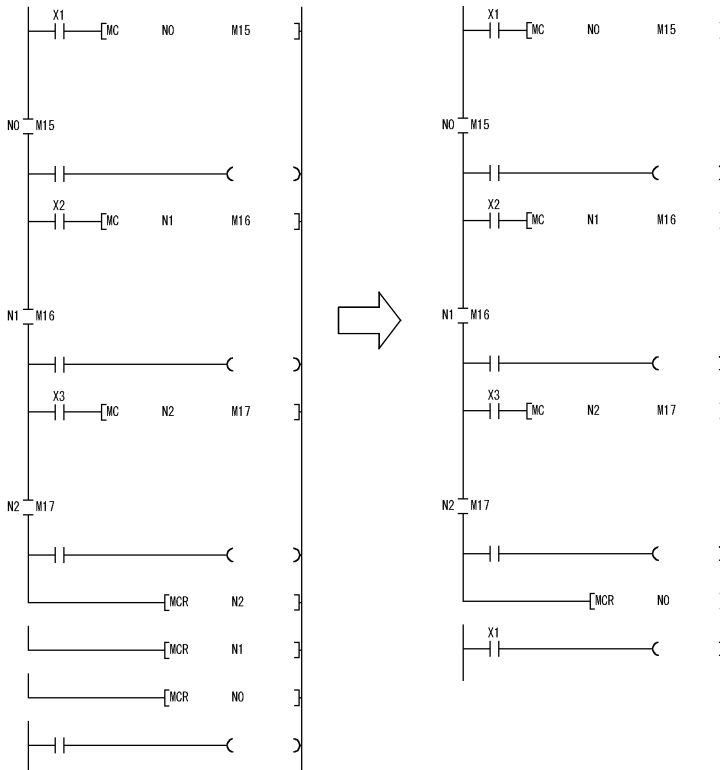
Note the following when nesting the instructions.

- (1) The instructions can be nested to a level of eight (N0 to 7). When nesting them, use MC from lower to higher nesting (N) numbers and MCR from higher to lower numbers. In the opposite order, the PLC function cannot perform normal operation since the instructions cannot be nested.



Since buses cross each other, normal master control ladder cannot be created.

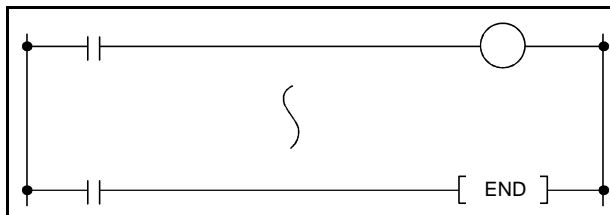
- (2) When the MCR instructions are gathered in one place in the nesting structure, all master controls can be terminated by one lowest nesting (N) number.



6.8 End Instruction

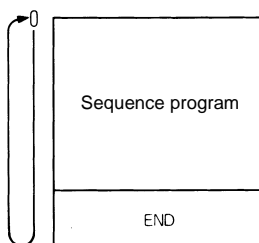
6.8.1 Sequence program end ... END

Usable Devices									Digit Designation	Error Flag (M9010, M9011)
Bit devices			Word (16-bit) devices			Constants		Level		
X	Y	M	T	C	D	K	H	N		



Functions

- (1) Indicates the end of a program. Execution terminates scanning at this step and returns to step 0.



- (2) The END instruction cannot be used halfway through the sequence program.

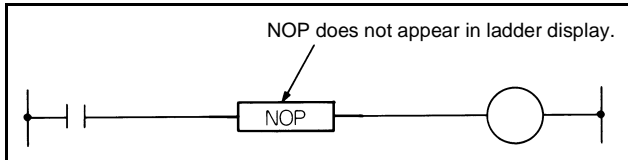
CAUTION

If the END instruction does not exist in the program, an operation error occurs and the PLC function does not operate.

6.9 Other Instructions

6.9.1 No operation ... NOP

Usable Devices									Digit Designation	Error Flag (M9010, M9011)
Bit devices			Word (16-bit) devices		Constants		Level			
X	Y	M	T	C	D	K	H	N		(M9010, M9011)
										○



Functions

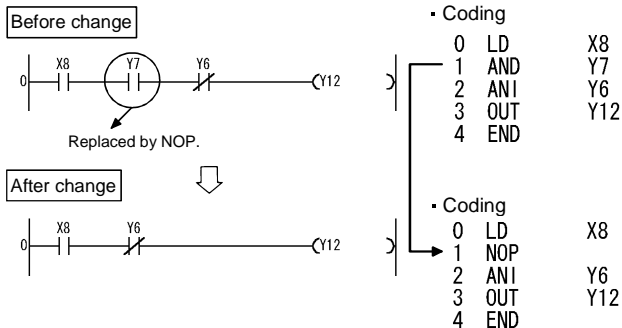
NOP

- (1) No-operation instruction that has no influence on the preceding operation.
- (2) Use NOP to:
 - 1) Provide space for debugging of a sequence program.
 - 2) Delete an instruction without changing the number of steps. (Change the instruction for NOP)
 - 3) Delete an instruction temporarily.

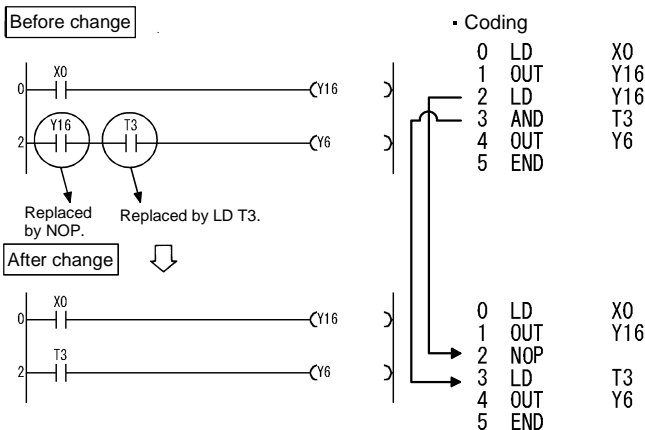
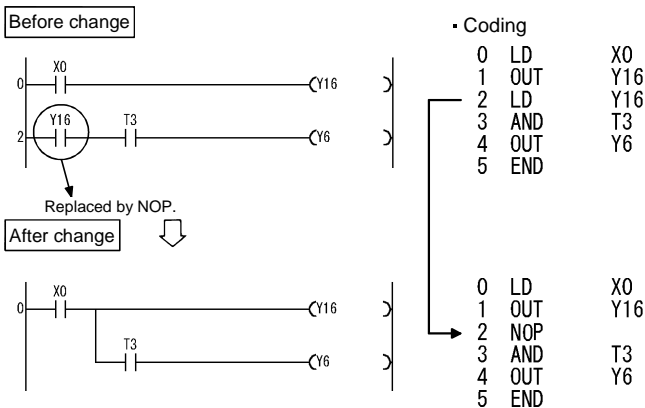
Program Examples

NOP

1) Contact short-circuit (AND, ANI)



2) Contact short-circuit (LD, LDI).....Note that if LD or LDI is replaced by NOP, the ladder will be completely changed.



The basic instructions can handle numerical data represented in 16 bits, and are classified as follows.

Basic Instruction Type	Description	Reference Page
Comparison operation instruction	Comparison such as =, >, <	84
Data transfer instruction	Transfer of specified data	88

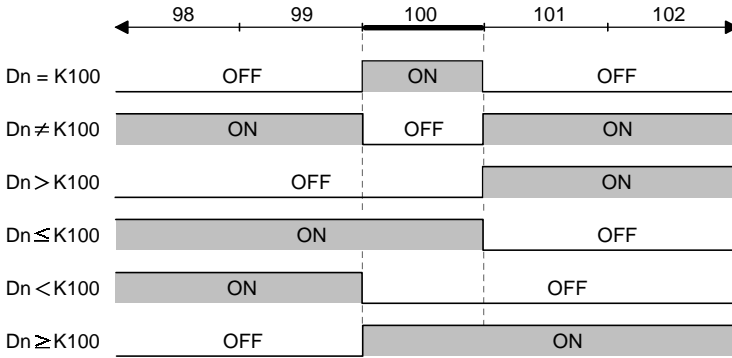
For the number of steps, refer to the instruction manual of the FREQROL-C500.

6.10 Comparison Operation Instructions

- (1) The comparison operation instruction is handled as a contact, compares the magnitudes of two pieces of data (e.g. =, >, <), and turns on when the condition holds.
- (2) Use the comparison operation instructions in the same manner as the contact instructions of the PLC instructions as indicated below.
 - LD, LDI LD=
 - AND, ANI AND=
 - OR, ORI OR=
- (3) There are the following 18 different comparison operation instructions. Refer to page 86 for details.

Classification	Instruction Symbol	Classification	Instruction Symbol	Classification	Instruction Symbol
=	LD=	>	LD>	<	LD<
	AND=		AND>		AND<
	OR=		OR>		OR<
≠	LD<>	≤	LD≤	≥	LD>=
	AND<>		AND≤		AND>=
	OR<>		OR≤		OR>=

(4) The conditions that the comparison operation instructions turn on are as follows.

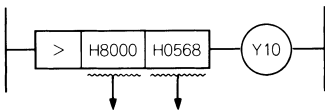


CAUTION

The comparison instruction regards the specified data as BIN values. Hence, if the value whose most significant bit (b15) is 1 (8 to F) is specified for comparison of hexadecimal data, it is regarded as a negative BIN value.

Example

Comparison of 4-digit HEX values

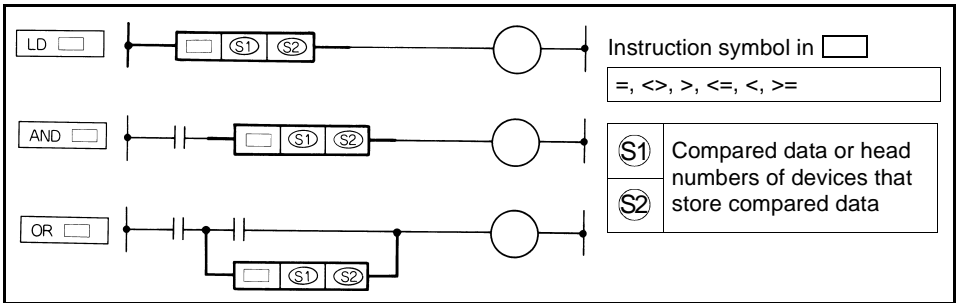


Regarded as -32767 in BIN. Regarded as 1384 in BIN.

Therefore, the result is -32767 < 1384 and Y10 does not turn on.

6.10.1 16-bit data comparison ... =, <>, >, <=, <, >=

	Usable Devices									Digit Designation	Error Flag (M9010, M9011)
	Bit devices			Word (16-bit) devices			Constants		Level		
	X	Y	M	T	C	D	K	H	N		
Ⓢ1	○	○	○	○	○	○	○	○		K1 to K4	○
Ⓢ2	○	○	○	○	○	○	○	○			



Functions

- Handled as an N/O contact and performs 16-bit comparison operation.
- The comparison operation results are as indicated below.

Instruction symbol in <input type="text"/>	Condition	Comparison Operation Result	Instruction symbol in <input type="text"/>	Condition	Comparison Operation Result						
=	Ⓢ1 = Ⓢ2	Energize	=	Ⓢ1 ≠ Ⓢ2	Not energize						
<>	Ⓢ1 ≠ Ⓢ2		>	Ⓢ1 > Ⓢ2		<=	Ⓢ1 ≤ Ⓢ2	<	Ⓢ1 < Ⓢ2	>=	Ⓢ1 ≥ Ⓢ2
>	Ⓢ1 > Ⓢ2		<=	Ⓢ1 ≤ Ⓢ2		<	Ⓢ1 < Ⓢ2	>=	Ⓢ1 ≥ Ⓢ2		
<=	Ⓢ1 ≤ Ⓢ2		<	Ⓢ1 < Ⓢ2		>=	Ⓢ1 ≥ Ⓢ2				
<	Ⓢ1 < Ⓢ2		>=	Ⓢ1 ≥ Ⓢ2							
>=	Ⓢ1 ≥ Ⓢ2										

Execution Conditions

The execution conditions of LD , AND and OR are as indicated below.

Instruction	Execution Condition
LD <input type="text"/>	Executed every scan.
AND <input type="text"/>	Executed only when the preceding contact instruction is on.
OR <input type="text"/>	Executed every scan.

REMARKS

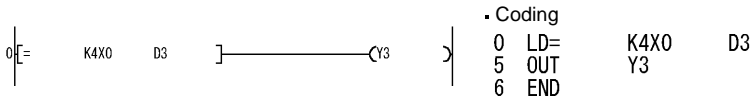
Seven steps are used when:

- The digit designation of a bit device is not K4.
- The beginning of a bit device is not a multiple of 8.

Program Examples

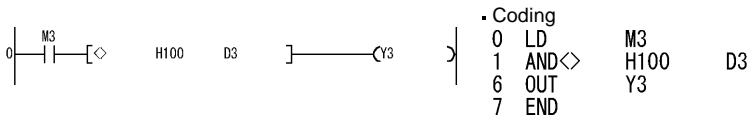
=

1) Program that compares the X0-F data and D3 data.



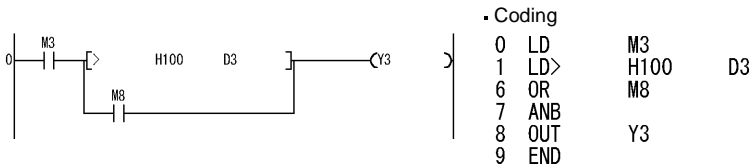
<>

2) Program that compares the BCD value 100 and D3 data.



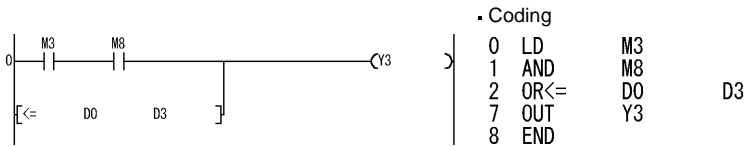
>

3) Program that compares the BIN value 100 and D3 data.



<=

4) Program that compares the D0 and D3 data.



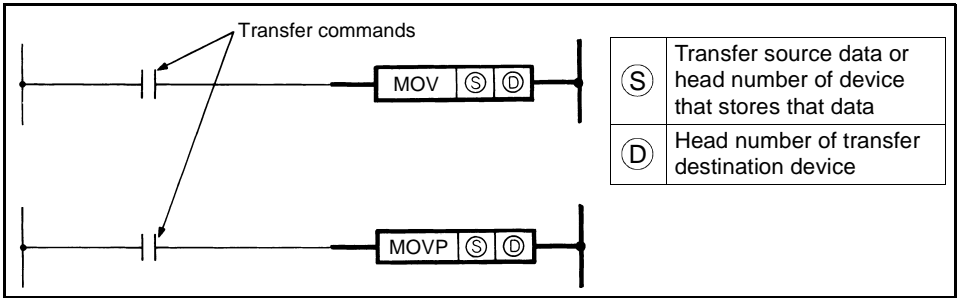
6.11 Data Transfer Instructions

The data transfer instructions are designed to transfer data.

The data moved by the data transfer instruction is maintained until new data is transferred.

6.11.1 16-bit data transfer ... MOV, MOVP

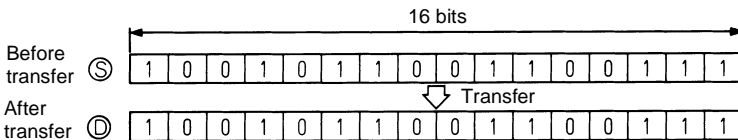
		Usable Devices									Digit Designation	Error Flag
		Bit devices			Word (16-bit) devices			Constants		Level		
		X	Y	M	T	C	D	K	H	N		(M9010, M9011)
MOV	(S)	○	○	○	○	○	○	○	○		K1 to K4	○
	(D)		○	○	○	○	○					



Functions

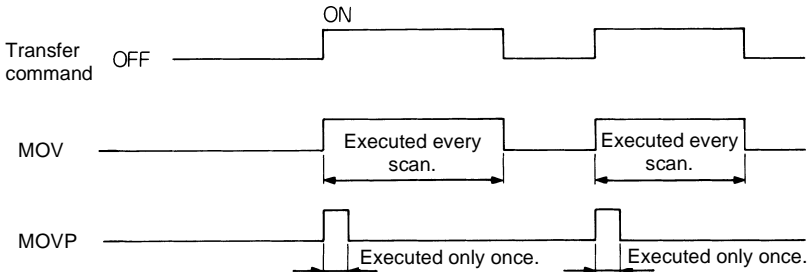
MOV

Transfers the 16-bit data of the device specified at (S) to the device specified at (D).



Execution Conditions

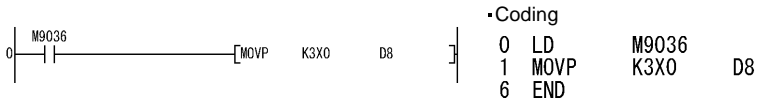
The execution conditions of the transfer instructions are as shown below.



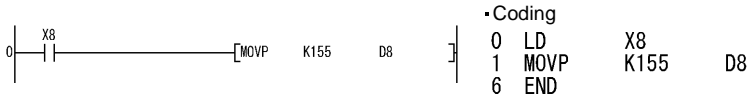
Program Examples

MOV

1) Program that stores the input X0-B data into D8.

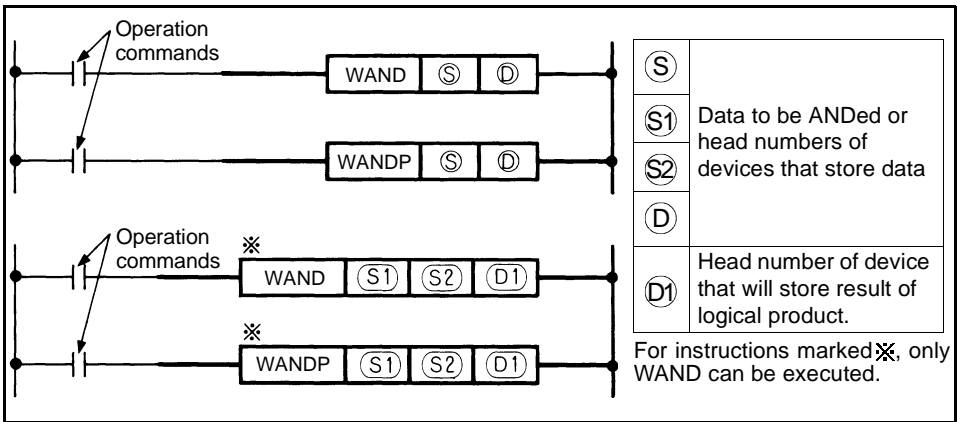


2) Program that stores 155 into D8 in binary when X8 turns on.



6.12 16-bit Logical Product ... WAND, WANDP

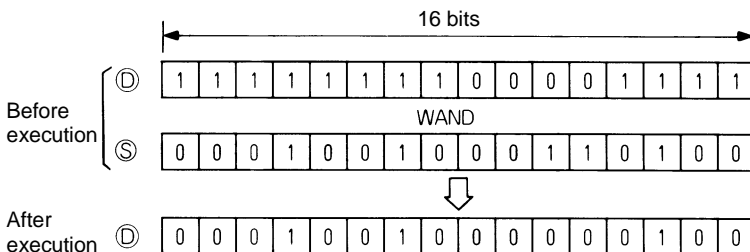
		Usable Devices									Digit Designation	Error Flag
		Bit devices			Word (16-bit) devices			Constants		Level		
		X	Y	M	T	C	D	K	H	N		(M9010, M9011)
WAND	(S)	○	○	○	○	○	○	○	○		K1 to K4	○
	(D)		○	○	○	○	○					
	(S1)	○	○	○	○	○	○	○	○			
	(S2)	○	○	○	○	○	○	○	○			
	(D1)		○	○	○	○	○					



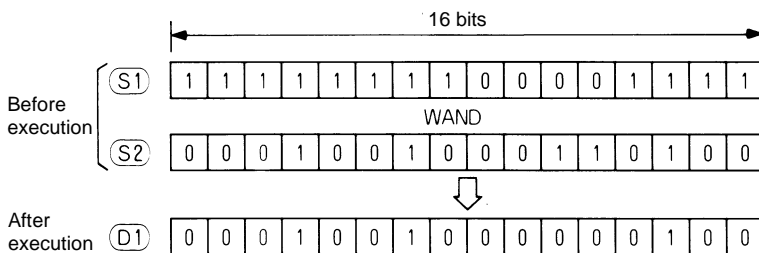
Functions

WAND

- (1) ANDs the 16-bit data of the device specified at (D) and the 16-bit data of the device specified at (S) on a bit-by-bit basis, and stores the result into the device specified at (D).



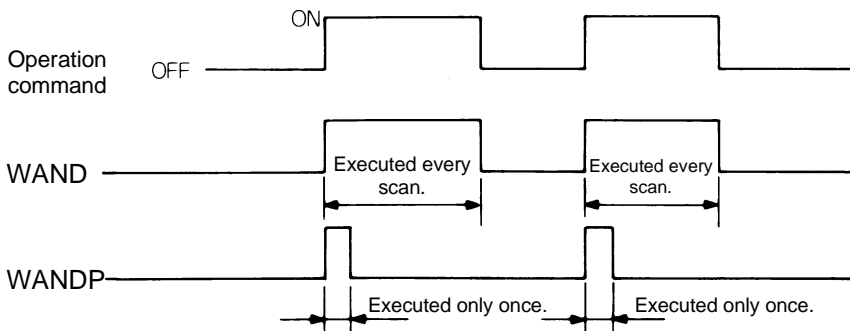
- (2) ANDs the 16-bit data of the device specified at (S1) and the 16-bit data of the device specified at (S2) on a bit-by-bit basis, and stores the result into the device specified at (D1).



- (3) More than the digit designation of a bit device is regarded as 0 for operation.

Execution Conditions

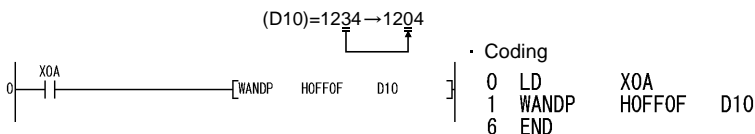
The execution conditions of the logical product instructions are as shown below.



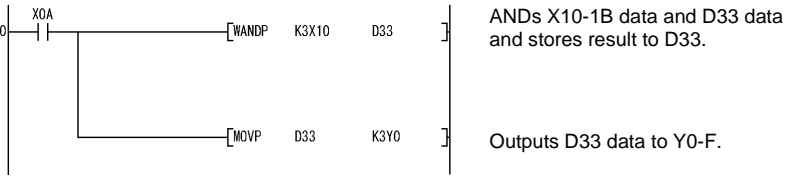
Program Examples

WAND

- 1) Program that masks the tenth digit (second place from the least significant digit) with 0 among the four BCD digits of D10 when XA turns on.



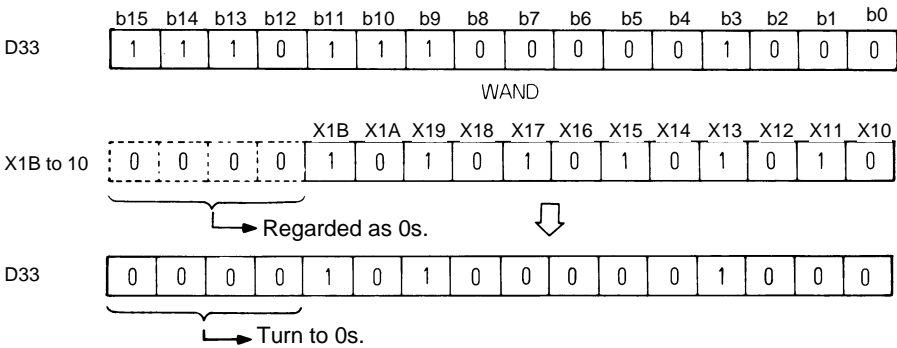
2) Program that ANDs the X10-1B and D33 data and outputs the result to Y0-B when XA turns on.



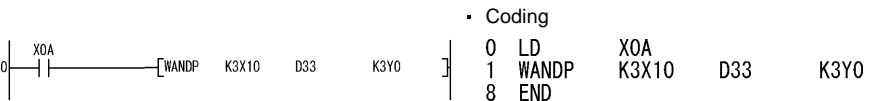
· Coding

```

0 LD      X0A
1 WANDP   K3X10  D33
6 MOV     D33    K3Y0
11 END
    
```



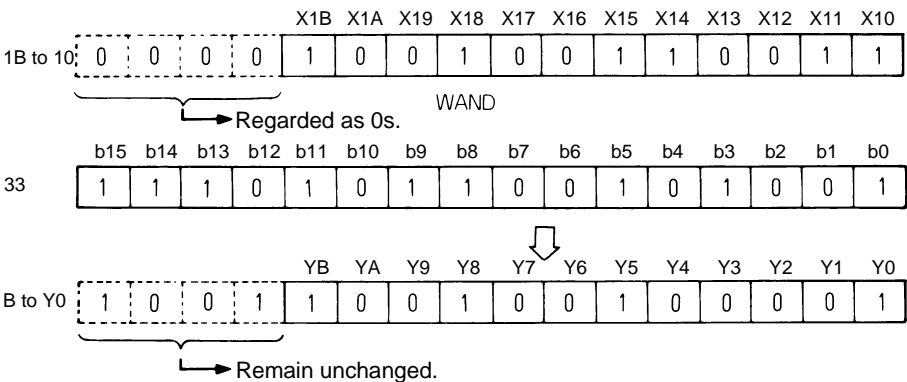
3) Program that ANDs the X10-1B and D33 data and outputs the result to Y0-B when XA turns on.



· Coding

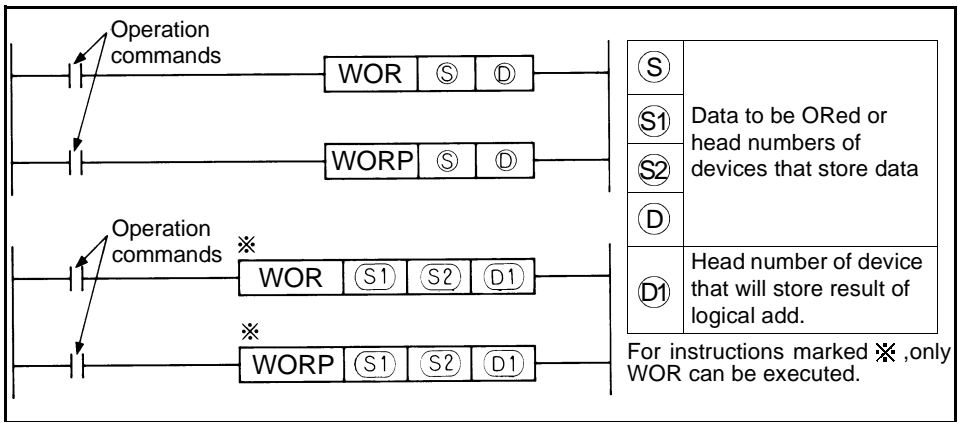
```

0 LD      X0A
1 WANDP   K3X10  D33    K3Y0
8 END
    
```



6.13 16-bit Logical Add ... WOR, WORP

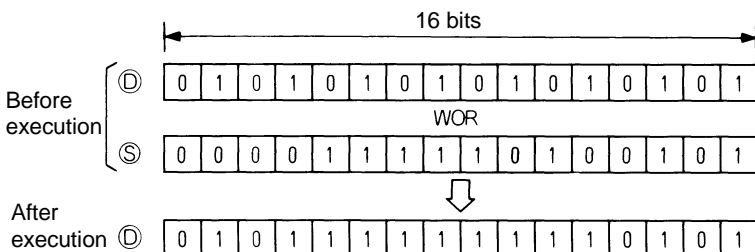
		Usable Devices									Digit Designation	Error Flag
		Bit devices			Word (16-bit) devices			Constants		Level		
		X	Y	M	T	C	D	K	H	N		(M9010, M9011)
WOR	(S)	○	○	○	○	○	○	○	○		K1 to K4	○
	(D)		○	○	○	○	○					
	(S1)	○	○	○	○	○	○	○	○			
	(S2)	○	○	○	○	○	○	○	○			
	(D1)		○	○	○	○	○					



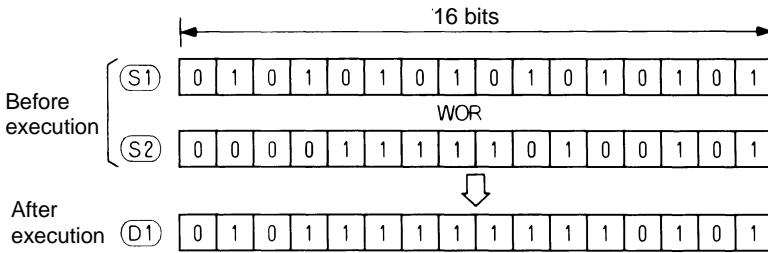
Functions

WOR

- ORs the 16-bit data of the device specified at (D) and the 16-bit data of the device specified at (S) on a bit-by-bit basis, and stores the result into the device specified at (D).



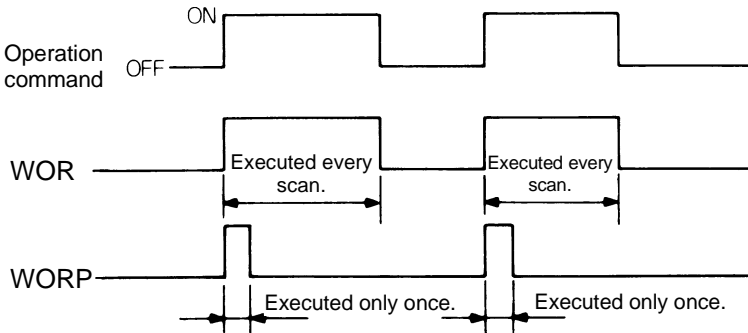
- (2) ORs the 16-bit data of the device specified at (S1) and the 16-bit data of the device specified at (S2) on a bit-by-bit basis, and stores the result into the device specified at (D1).



- (3) More than the digit designation of a bit device is regarded as 0 for operation.

Execution Conditions

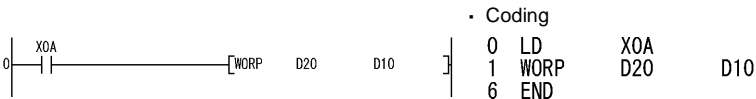
The execution conditions of the logical add instructions are as shown below.



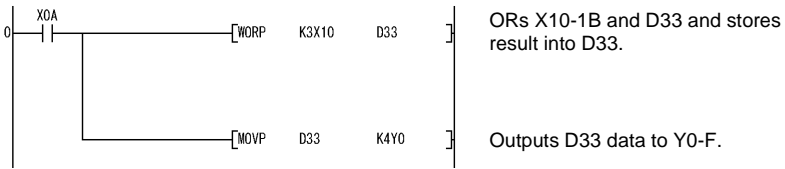
Program Examples

WOR

- 1) Program that ORs the D10 and D20 data and stores the result into D10 when XA turns on.



- 2) Program that ORs the X10-1B and D33 data and outputs the result to Y0-F when XA turns on.



• Coding

```

0 LD      X0A
1 WORP   K3X10  D33
6 MOVP   D33    K4Y0
11 END

```

- 3) Program that ORs the D10 and D20 data and stores the result into D33 when XA turns on.



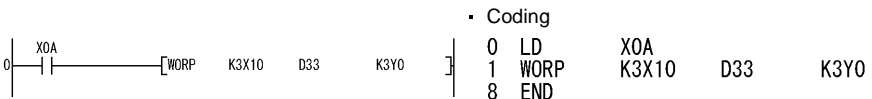
• Coding

```

0 LD      X0A
1 WORP   D10    D20
8 END

```

- 4) Program that ORs the X10-1B and D33 data and outputs the result to Y0-B when XA turns on.



• Coding

```

0 LD      X0A
1 WORP   K3X10  D33
8 END

```

MEMO

7. ERROR CODE LIST

7.1 How to Read the Error Code 98

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

When the built-in PLC function is in the RUN status or if an alarm occurs during RUN, the self-diagnostic function displays the error and stores the error code and error step into the special registers. This chapter describes the error definitions and corrective actions.

7.1 How to Read the Error Code

When an error has occurred, the error code can be read with the peripheral device. For the operation method, refer to the operating manual of the peripheral device. The following table indicates the error names, error codes, definitions, causes and corrective actions.

The error code and error step are stored into the following special registers.

Error code..... D9008
 Error step..... D9010, D9011

Table 7.1 Error Code List

Error Name	Error Code (D9008)	Status	Definition and Cause	Corrective Action
“INSTRCT CODE ERR.” [Checked at instruction execution]	10	Stop	The instruction code that cannot be decoded is included in the program. • The memory contents changed for some reason.	Read the error step using GX Developer, and correct that step in the program.
“PARAMETER ERROR” [Checked at power-on or STOP to RUN]	11	Stop	(1) Write to the CPU was performed after the capacity larger than the memory capacity of the CPU was set using GX Developer. (2) The parameter data of the CPU memory changed due to noise or memory loading fault.	Check the memory capacity of the CPU with the memory capacity set using GX Developer, and re-set using GX Developer.
“WDT ERROR” [Checked at END processing execution]	22	Stop	The scan time exceeds the watchdog error monitor time. • The user program scan time has increased.	Calculate/check the user program scan time and reduce the scan time.
“END NOT EXECUTE” [Checked at END instruction execution]	24	Stop	(1) The END instruction has been read as another instruction code due to noise, etc. (2) The END instruction has changed into another instruction code for some reason.	Reset and RUN again. If the same error appears again, the cause is a CPU hardware fault. Consult the Mitsubishi representative.

MEMO

REVISIONS

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

Print Date	*Manual Number	Revision
Aug., 2002	IB(NA)-0600116E-A	First edition