MITSUBISHI
Programmable Controller MELSEC F1 SERIES

Programming Manual

HIGH FUNCTION VERSION

- This manual is prepared to explain the software functions and gives programming instructions for Mitsubishi Programmable Controllers, MELSEC F1 series (Type -ES and -UL) except the F1-12 type.
- Users should ensure that the details of this manual is studied and understood before making a program and attempting to use the handy programming panel (HPP) or graphic programming panel (HGP).
- Other information concerning the hardware and operations of the Programmers is covered in separate manuals.

MITSUBISHI ELECTRIC CORPORATION

CONTENTS

1 INTRODUCTION

1-1 Location and function of programmable controller

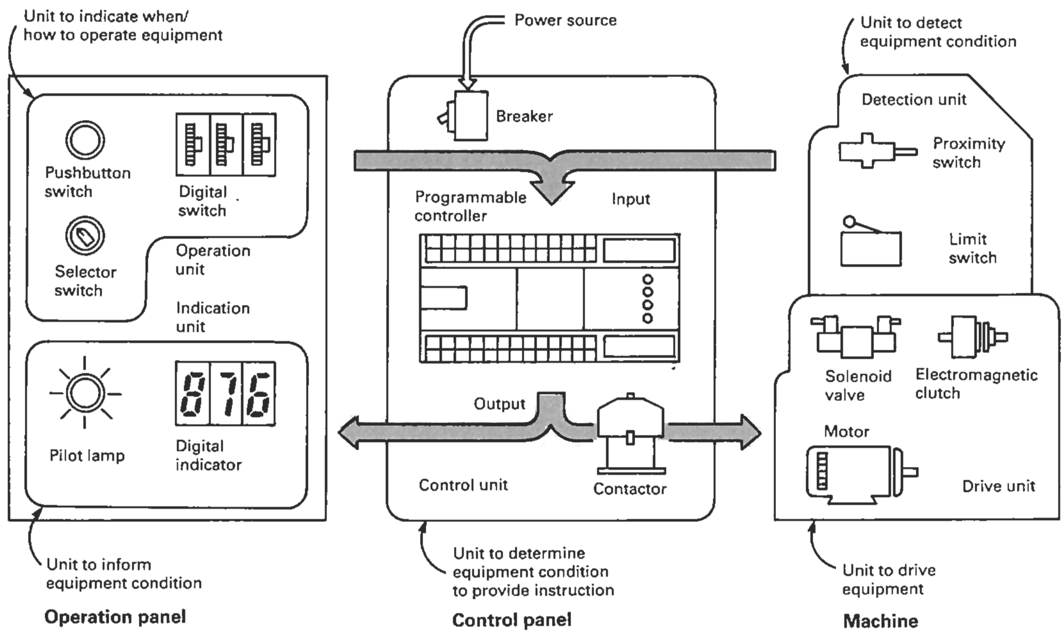


Fig. 1-1 Location of programmable controller

The programmable controller (PC in short) is operated by the instruction input from the pushbutton switch, selector switch, digital switch, etc. provided on the operation panel, or sensor input from the limit switch, proximity switch, photo-electric switch, etc. used to detect the operation condition of the equipment, and serves to control the driving loads such as solenoid valve, motor, electromagnetic clutch, etc. and indication loads such as pilot lamp, digital indicator, etc.
The transmission of output signal against these input signals is determined by the contents of program to be provided to the programmable controller.
The light loads such as small type solenoid valve, pilot valve, etc. can be directly driven by the programmable controller, however, the heavy loads such as 3-phase motor, large-capacity solenoid valve, etc. need to be driven through the contactor or the intermediate relay.
Such contactor, intermediate relay, power breaker, etc. are installed in the control panel together with the programmable controller.
The programmable controller will play the important roles as a small type, high-reliability and flexible brain when designing the automated product machining, assembling, transfer, inspection, packing, etc.

1-2 Internal configuration of programmable controller

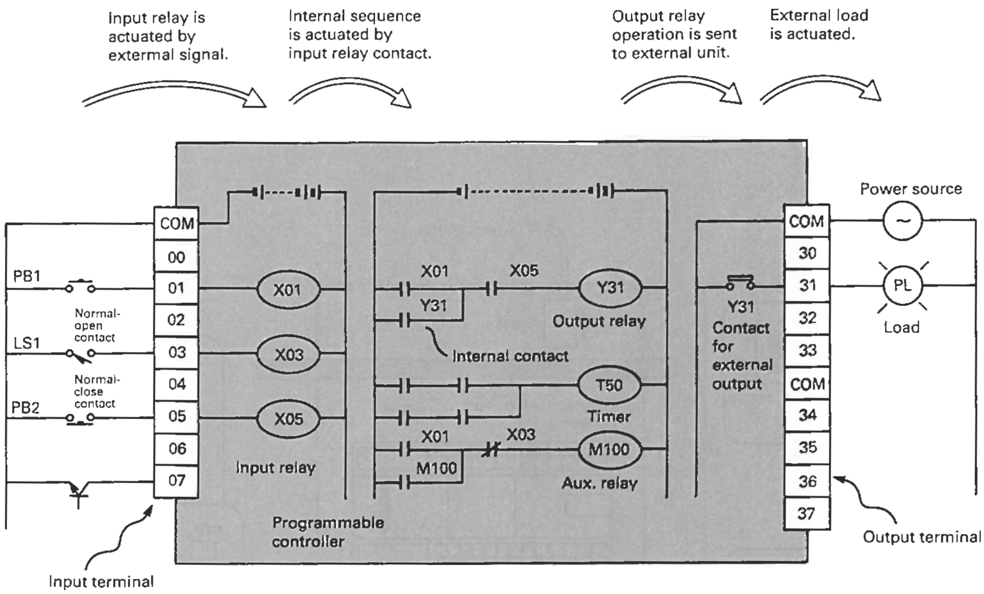


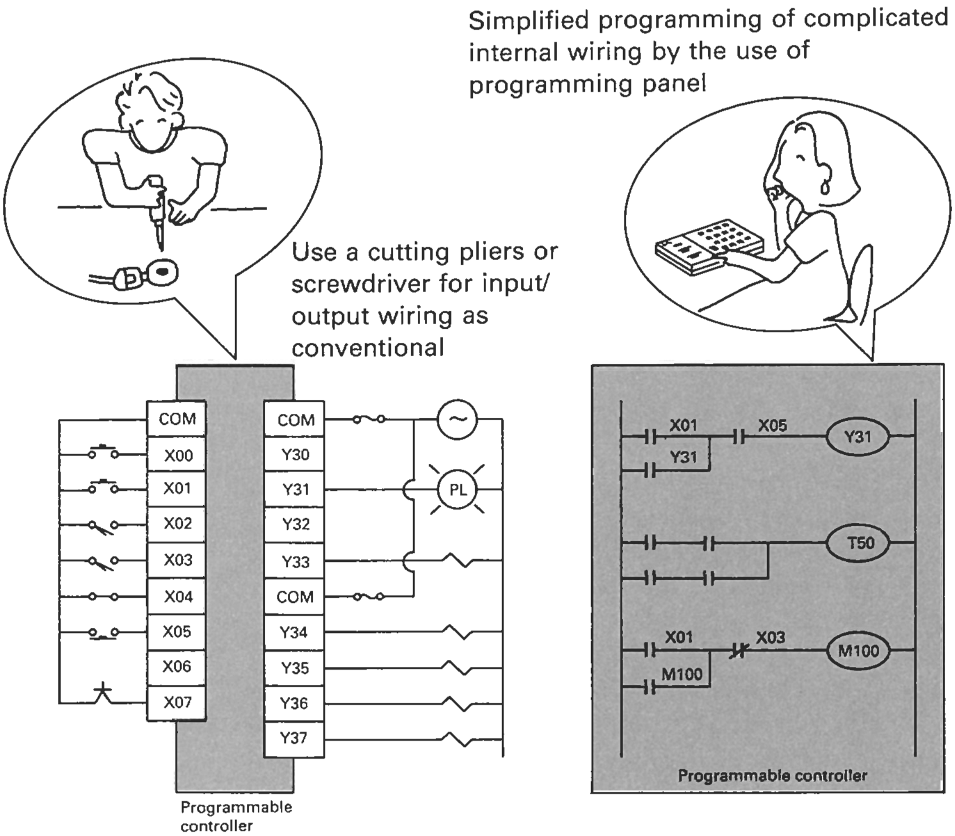
Fig. 1-2 Equivalent circuit of programmable controller

The programmable controller is composed of electronic circuits with a micro-computer centered, however, it can be equivalently regarded as an integrated body of ordinary relay, timer, counter, etc.
The input relay (X) built in the programmable controller is driven by the external switch through the input terminal.
The output relay (Y) built in the programmable controller is provided with various internal contacts in addition to the external output contact (1a).
Besides, it is incorporated with various types of elements such as timer (T), counter (C), auxiliary relay (M), state (S), coil (F) for function block, etc.
In addition, these elements (X, Y, M, T, C, S, F) are provided with many electrically normally-open contacts (a-contact) and normally-close contacts (b-contact), and can be used optionally within the programmable controller.
Consequently, if the conventional relay panel adopts 2a-pushbutton switch or 2-stage/4-notch selector switch, it will be only necessary to use 1a-pushbutton switch or 1-stage/4-notch selector switch to replace such conventional relay panel with the programmable controller.

NOTE

Since the input relay (X05) in Fig. 1-2 is driven through the pushbutton switch (b-contact), the output relay (Y3l) can be latched by actuating the input relay (X01).
When the pushbutton switch (PB2) is depressed to turn off the input relay (X05), the output relay (Y31) is turned off at the same time.
(Be careful that the contact X05 is a normally-open contact in the program).
In the case of auxiliary relay (M100), on the other hand, the contact of input relay (X03) is used as a normally-close contact, therefore, the latching condition of auxiliary relay (M100) is reset when the limit switch (LS1) is actuated, causing the input relay (X03) to actuate.

1-3 Program and instruction concept



(a) Input/output wiring diagram

(b) Sequence circuit diagram

Fig. 1-3 Disassembling equivalent circuit

When designing the control panel using a programmable controller, the circuit shown in Fig. 1-2 needs to be disassembled to the circuit shown in Fig. 1-3(a)(b). Fig. 1-3(a) shows the input/output allotment and wiring.
The wiring needs to be done in the same manner as the conventional relay panel by the use of a screw-driver, cutting pliers, etc.
On the other hand, Fig. 1-3(b) shows the relative connection of each element within the programmable controller.
The connection of these contacts and coils can be executed by key operation of programming panel.
The output coil (Y31) in Fig. 1-3(b), for instance, is driven through the parallel circuit composed of input relay contact (X01) and output relay contact (Y31) and series circuit composed of input relay contact (Y31).
To perform the above connection by the key operation of the programming panel, it is necessary to use the instructions corresponding to the series connection or parallel connection of contact, normally-open contact or normally-close contact, etc.
In addition, it is necessary to provide the division of each element and element number.
Accordingly, the instructions for the programmable controller are made up of instruction words and element No. in combination, indicating the connecting method.
The multiple number of instructions are integrated to make up a program.

[1] Program capacity and step No.

Step No.	Inst-ruction	Element No.
0	LD	X01
1	OR	Y31
2	AND	X05
3	OUT	Y31
⋮	⋮	
999	END	

A large number of instructions making up a program are assigned with sequential No., each of which is called "Step No.".

In the MELSEC F_1 series, it is possible to assign step Nos. in the range from "0" to "999", which allows the programming a total of 1,000 instructions.
It is referred to as a "Program capacity".

As explained previously, each element used in the programmable controller is provided with a number of contacts, however, the actual number of contact to be used will be restricted by the program capacity.

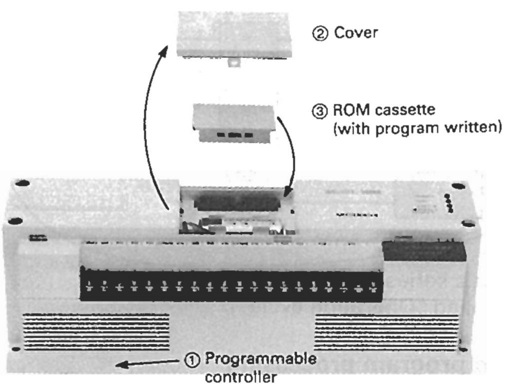
[2] Program memory

A memory used to store such large number of instructions is called "Program memory".
The program memory is optionally available in the EPROM memory, EEPROM memory, etc. in addition to the RAM memory incorporated in the programmable controller.

– **RAM memory (built in programmable controller)**
The memory allows the instantaneous writing/reading, however, it is designed to hold the memory contents by the battery back-up (built in programmable controller), as the contents of memory may be lost in the event of power failure. The memory is applicable to the test operation stage in which the program may be frequently modified.

– **EPROM memory (type F-ROM-1 ROM cassette, option)**
The memory is used exclusively for reading, and requires the special devices (ROM writer, eraser) for writing, correction and erasing.
Since the memory of this type is most resistible against the noise and the program once stored will not be lost, it is suitable to maintenance-free application.

– **EEPROM memory (type F-EEPROM-1 ROM cassette, option)**
The program in the memory will not be lost even in the event of power failure in the same manner as the EPROM memory.
The memory of this type will not require any special device for writing and erasing, which can be executed by the use of programming panel.



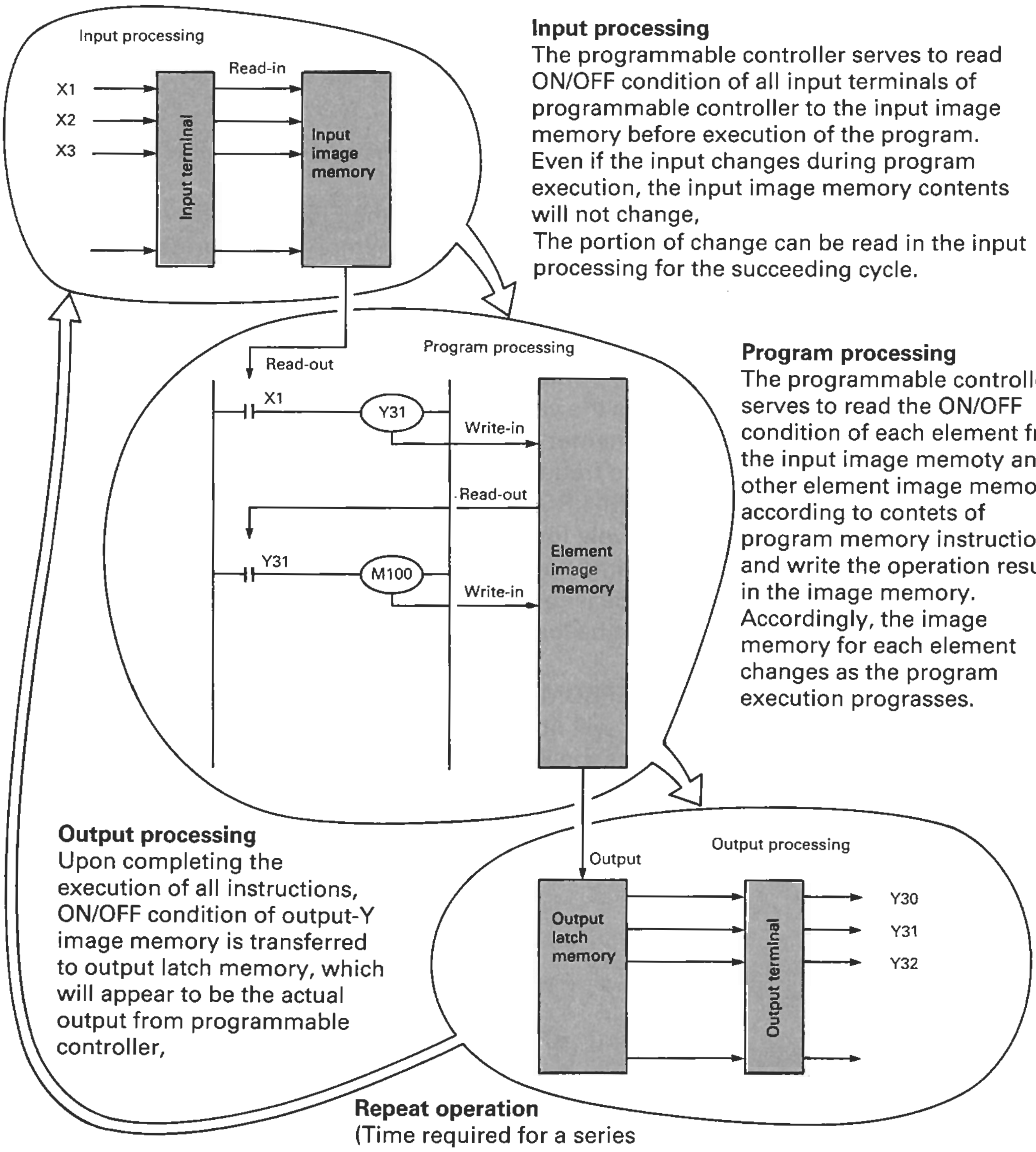
With the type F-ROM-1 or type F-EEPROM-1 ROM cassette mounted to the programmable controller, the programmable controller will operate in accordance with the contents of program stored in the ROM cassette.

The operation contents of the programmable controller can be changed with ease simply by replacing the ROM cassette.

1-4 Differences between relay panel and programmable controller

As described previously, the programmable controller is an integral body made up of multiple electronic relays, timers and counters and its internal wiring is executed by the programming panel.
As far as the sequence execution method is concerned, there exist fundamental differences between the relay panel and the programmable controller.
In other words, all sequences are executed parallelly at the same time in the case of relay panel.
In the case of programmable controller, on the other hand, the operation is repeated cyclicly in order of program.
The visual differences between these units will be described further on;

(1) Input/output processing



Input processing
The programmable controller serves to read ON/OFF condition of all input terminals of programmable controller to the input image memory before execution of the program.
Even if the input changes during program execution, the input image memory contents will not change,
The portion of change can be read in the input processing for the succeeding cycle.

Program processing
The programmable controlle serves to read the ON/OFF condition of each element frc the input image memoty and other element image memor according to contets of program memory instruction and write the operation resul in the image memory.
Accordingly, the image memory for each element changes as the program execution prograsses.

Output processing
Upon completing the execution of all instructions, ON/OFF condition of output-Y image memory is transferred to output latch memory, which will appear to be the actual output from programmable controller,

Repeat operation
(Time required for a series of operations is called "Operation cycle").

Fig. 1-4 Input/output and program processing

(2) Input/output response lagging

The programmable controller has the response lagging due to influence of operation cycle in addition to the electrical lagging (approx. 10ms) due to input filter and mechanical response lagging (approx. 10ms) due to output relay.

As an example, the consideration will be made on the sequence as shown in Fig. 1-5 in which the input terminal X1 is changed from OFF to ON right after the input processing has been completed.

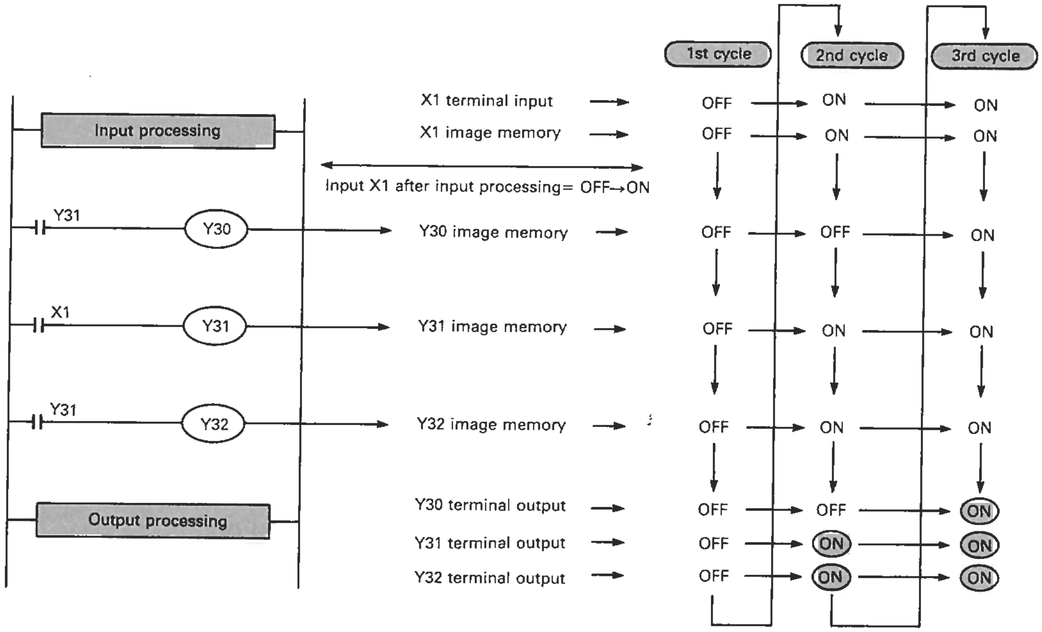


Fig. 1-5 Input/output response lagging

1st cycle
Since X1 in input image memory is OFF, all the output relays Y30, Y31 and Y32 are turned off.

2nd cycle
Since X1 in image memory is turned on by input processing, the image memory is turned on when Y31 is executed. The image memory of Y32 is turned on similarly.

3rd cycle
Since the image memory of Y31 is turned on, Y30 is also turned on.

As described above, a maximum of two-cycle response lagging will occur in Y31 and Y32 after the input has been turned on.

NOTE

Inprovement of input/output response lagging

Since there are the function instructions which allows the input processing/output processing while the program is being executed, the response lagging as above can be reduced by the use of such instruction.

In addition, the F_1 series is prepared with other function block which makes it possible to reduce the response lagging of input filter by the program.

(3) Double output operation

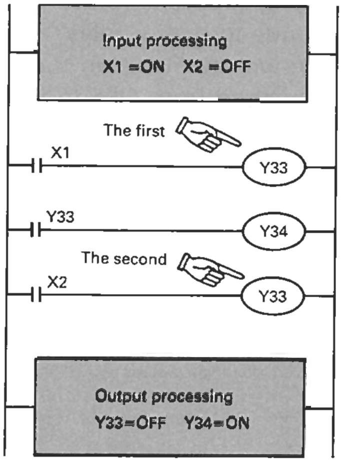
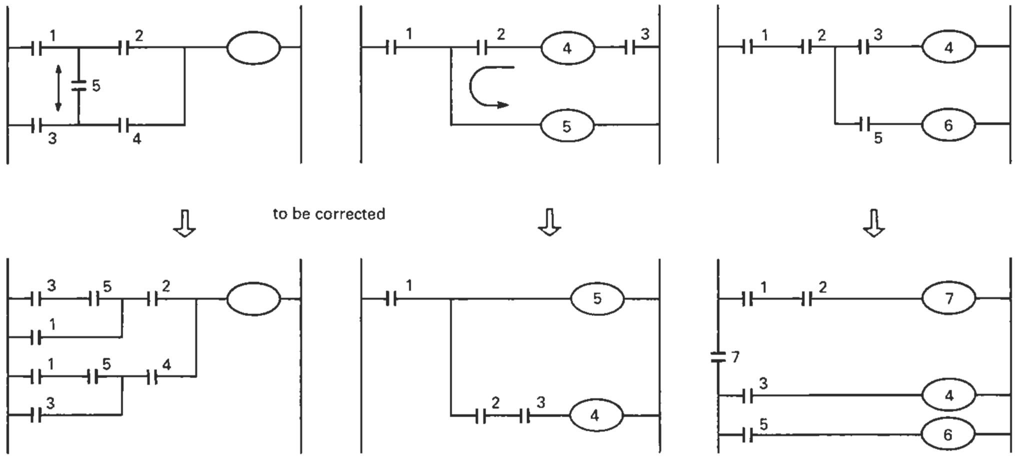


Fig. 1-6 Double output

Considered in the following is a case in which the equivalent coil Y33 is used at multiple locations as shown in Fig. 1-6.
As an example, it is supposed that X1 is turned on and X2 is turned off during input processing.
Since X1 is turned on, the memory image of the 1st Y33 is turned on.
Consequently, Y34 in the succeeding circuit will be also turned on.
Since X2 to the 2nd Y33 is OFF, however, the image memory of Y33 will be re-written to OFF.
As a result, the actual output when the output processing is executed will result in that Y33 is turned off and Y34 is turned on.
Bear in mind that the execution of double output as stated above will cause the operation executed later to be a preferential operation.
Besides, the double output will come in different way when the jump instruction or step-ladder instruction is used, which will be described further on.

(4) Un-programmable circuit



(a) Bridging circuit (b) Reversing circuit (c) Branch output circuit

Fig. 1-7 Un-programmable circuits and countermeasures

Since the circuit as shown at the upper stage of Fig. 1-7 cannot be programmed directly, it is necessary to program by changing the circuit to that shown at the lower stage.
In the upper figure in Fig. 1-7(a), the problem is that the current flows across the contact-5.
In the upper figure in Fig. 1-7(b), on the other hand, the problem is that the contact-3 is used after the coil-4.
In addition, it is necessary to program the coil-5 which is not accompanied by any contact, before the coil-4.
The circuit as shown in Fig. 1-7(c) is programmed by the use of master control instruction to be given further on.

2 ELEMENTS AND ELEMENT NUMBERS

The instructions used for programmable controller can be divided into those which function by themselves and those which will function with "Instruction + element No." in combination. This section at first describes the kinds of elements and their Nos.

2-1 Input relay (X)

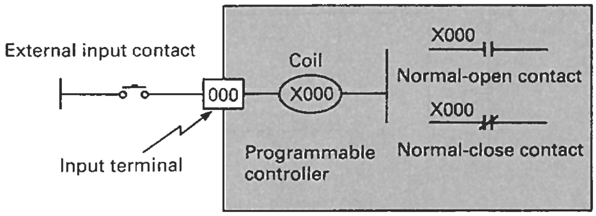


Fig. 2-1 Input relay circuit

"Input" is a window through which the signal is received by the programmable controller from the external switch.
The input relay (X) connected to the input terminal of the programmable controller is an electronics relay which is insulated optically, and is provided with a number of normally-open contacts (a-contact) and normally-close contacts (b-contact).
These contacts may be used optionally within the programmable controller.
It is not possible to drive the input relay by the contact incorporated in the programmable controller.

Input relay No.
The input relay (X) is assigned with No. in octal figure as shown below, depending upon the basic unit or extension unit.
(For details, refer to page 163).
A part or all of these input relays may be used, depending upon the type of programmable controller used.

Basic unit

X000 ~ X007	X010 ~ X013
X400 ~ X407	X410 ~ X413
X500 ~ X507	X510 ~ X513

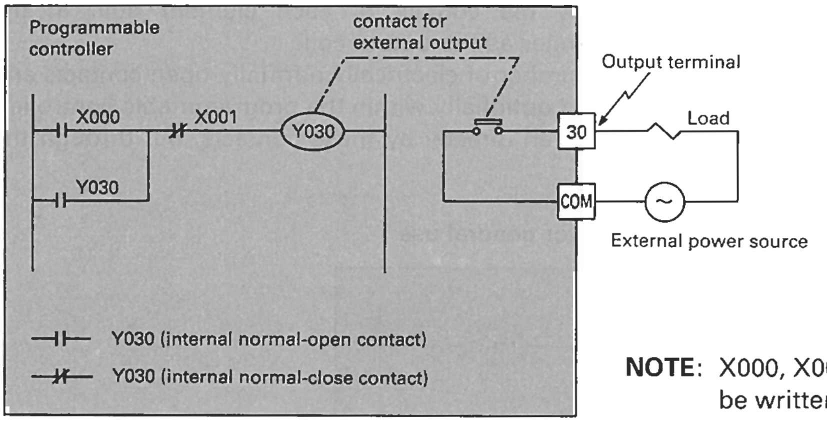
Extension unit

X014 ~ X017	X020 ~ X027
X414 ~ X417	X420 ~ X427
X514 ~ X517	X520 ~ X527

NOTE

Response time lagging of input relay
The input relay has the following response lagging due to influence by C-R filter.
Since eight points of input X400 ~ X407 can be changed filter constant by the program, both ON and OFF response time lagging can be zero to 60ms.
(See functional instruction F670 K101).
The standard response lagging in input X400 ~ X407 is approx. 10ms both for OFF → ON and ON → OFF.

2-2 Output relay (Y)



NOTE: X000, X001, Y030, etc. may be written as X0, X1 and Y30.

Fig. 2-2 Output relay circuit

The output terminal is a window through which the signal is sent out from the programmable controller to the external load.
The output contact of the output relay is connected to the output terminal within the programmable controller. Fig. 2-2 shows and example of output with contact.
The external output contact is turned on/off, depending upon the ON/OFF condition of output latch memory shown on page-6.

Output relay No.
The output relay (Y) is assigned with No. in octal figure as shown below, depending upon the basic unit or extension unit.
(For details, refer to page-163).
A part or all of these output relays may be used, depending upon the type of programmable controller used.

Basic unit	Extension unit
Y030 ~ Y037 Y430 ~ Y437 Y530 ~ Y537	Y040 ~ Y047 Y440 ~ Y447 Y540 ~ Y547

NOTE

Response time lagging of external output contact
The response lagging for external output contact of output relay is approx. 10ms both for ON → OFF and OFF → ON.
The internal contact has not such mechanical response lagging.

2-3 Auxiliary relay (M)

The programmable controller is incorporated with a number of auxiliary relays.
The auxiliary relay coils are driven by the contact of each element built in the programmable controller in the same manner as the output coil.
The auxiliary relays are provided with a number of electrically normally-open contacts and normally-close contacts which can be used optionally within the programmable controller. However, the external load cannot be driven directly by these contacts, but through the output relay.

(1) M100 ~ M277 (128 points: octal No.) for general use

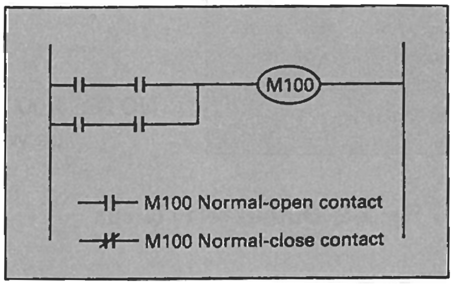


Fig. 2-3 Auxiliary relay circuit

128 points auxiliary relay are provided for the general use, each of which is assigned wit octal number ranging from M100 to M177 and from M200 to M277.

(2) M300 ~ M377 (64 points: octal No.) for retentive use (Battery back-up)
Should the power failure occur while operating the programmable controller, the outpu relays and auxiliary relays will be all turned off.
All these condition except those of which input conditions are turned on will also b turned off when the operation is re-started.
However, some control objects may be necessary to re-produce when the operation re-started, by storing the pre-power failure conditions.
These auxiliary relays are used for such purpose, and the memory is held in the event power failure by the back-up memory incorporated in the programmable controller.

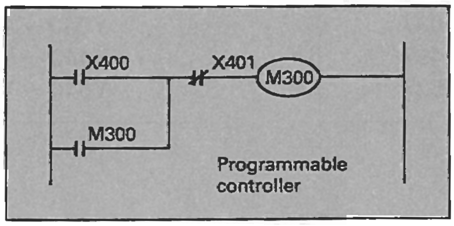


Fig. 2-4 Holding circuit in the event of power failure (self-holding type)

Fig. 2-4 shows an example in which the operation of M300 is held in the event of power failure.
If M300 is actuated with X400 turned on in this circuit, M300 self-holds the operation even if X400 is opened.
Accordingly, M300 continues the operation when the operation is re-started even if X400 is opened due to power failure.
If the normally-close contact X401 is opened, on the contrary, M300 will become inoperative.

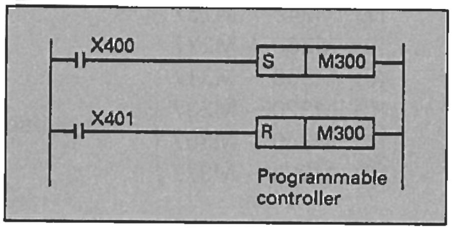


Fig. 2-5 Holding circuit in the event of power failure (set/reset type)

When using the set/reset instruction to be described further on, the circuit will be as shown in Fig. 2-5.

2-4 Shift register (M)

The auxiliary relay (M) may also be used as a shift register.
In this case, a series of auxiliary relays for 16 points are used as a group, the head No. c which is used as a shift register No.
Bear in mind that the group of these auxiliary relays cannot be used for other purpos when a part of the auxiliary relays (M) is used as a shift register.

Shift register No.

(1) M100 ~ M117
(2) M120 ~ M137
(3) M140 ~ M157
(4) M160 ~ M177
(5) M200 ~ M217
(6) M220 ~ M237
(7) M240 ~ M257
(8) M260 ~ M277
(9) M300 ~ M317
(10) M320 ~ M337
(11) M340 ~ M357
(12) M360 ~ M377
} Battery back-up

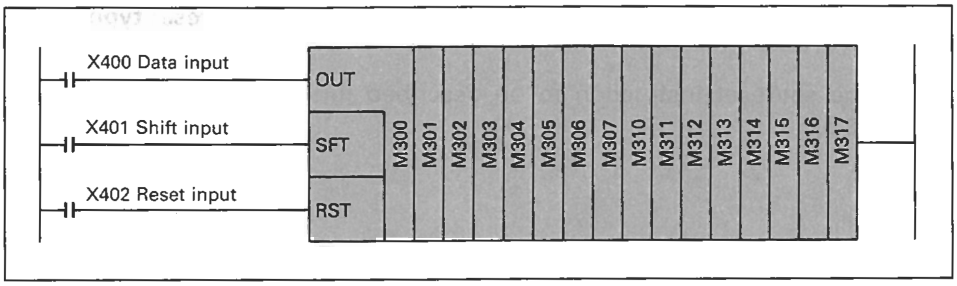
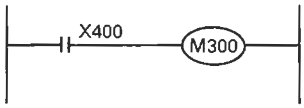


Fig. 2-6 Shift register (Example of M300)

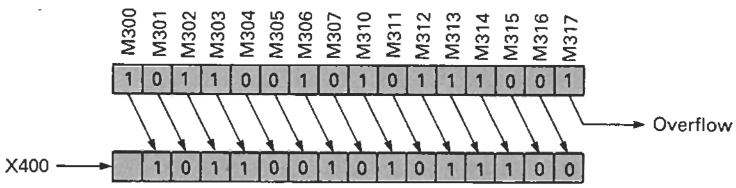
Operation of shift register

(1) Handling of data input
The ON/OFF condition of leading auxiliary relay M300 is established by the ON/OF condition of data input X400.
Accordingly, the operation is the same as that shown below.



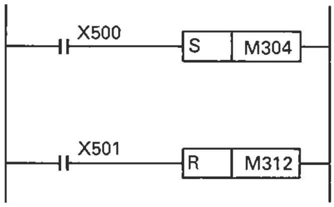
(2) Handling of reset input
When the reset input X402 is turned on, M301 ~ M317 are all turned off.
When they need to be operated as shift registers, therefore, it is necessary to turn o the reset input.

(3) Handling of shift input
When the shift input X401 is turned from OFF to ON, ON(1)/OFF(0) condition of eac auxiliary relay will be changed as follows;



It is also possible to receive the over-flow contents by combining the siftregister with a second one and cascade the two.
(Refer to page-34).

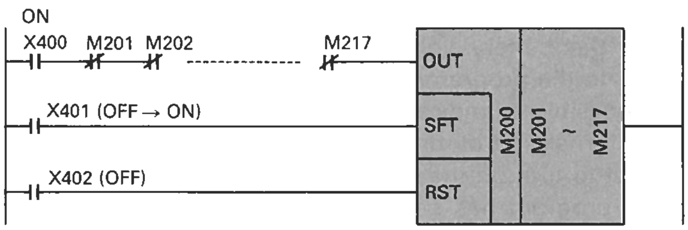
(4) Set/reset of intermediate relay
For the auxiliary relays M200 ~ M377, it will be possible to turn on/off any auxiliary relay in the shift register.
The set/reset instruction is used for the purpose as shown below;
Bear in mind that the use of general coil drive instruction (OUT instruction) may result in mal-operation of shift register.



M304 is turned on when X500 is turned on.

M312 is turned off when X501 is turned on.

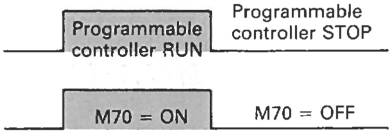
Reference circuit example



If the shift input X401 is turned on/off repeatedly in the above figure, either one of the auxiliary relays M200 ~ M217 is operated in order of M200 → M201 → M202...
In the initial operation, M201 ~ M217 are all turned off (normally-close contacts are all turned on), and only M200 is operated through X400 (ON).
When the shift input X401 is turned from OFF to ON in this case, M201 is turned on, and the normally-close contact is opened, causing M200 to be turned off.

2-5 Special auxiliary relay (M)

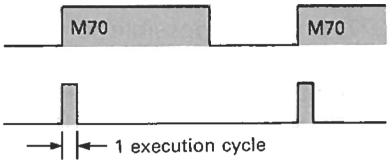
M70 RUN monitor



M70 is turned on/off automatically, depending upon RUN/STOP condition of the programmable controller.

The contact of M70 is used to drive the functional instruction, etc.

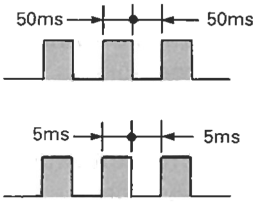
M71 Initialize pulse



M71 is turned on only for one execution cycle right after M70 has been turned on.
The contact of M71 is used to initialize the counter, shift register, state, etc.

M72 100ms clock

M73 10ms clock



M72 is turned on/off at an interval of 100ms, and M73 at an interval of 10ms.

Counting this contact operation by the counter provide with the timer of 0.1 ~ 99.9sec. and 0.01 ~ 9.99sec.

M74

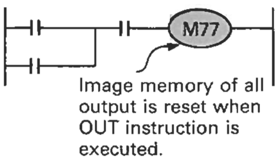
M75

Normally turned on in F_1 series.

M76 Battery voltage drop

When the battery voltage is dropped, M76 is turned on while power to the programmable controller is supplied.
It is possible to indicate the battery voltage drop condition with the external unit by driving the output relay (Y) by the use of this contact to turn on the lamp.
(RAM program has been held for approx. one month after M76 has been operated).

M77 Output inhibit



When M77 coil is operated by program, all output relays (Y) are automatically turned off.
In this case, other relays, timers and counters are kept operated.

M470 High-speed counter

The counters C660 and C661 are prepared in pairs to make up a 6-digit counter (pair counter).
The count input to the counter is used selectively as follows, depending upon ON/OFF condition of M470.

– In case M470 is ON

X400 is treated as count input and M401 as reset input and input filter for X400 and X401 is turned to be approx. 200μs automatically, so that the high-speed counting of 2kHz can be executed.
(1.5kHz for functional instruction F670 K119 is used)

– In case M470 is OFF

The optional contact in the programmable controller can be treated as count input or reset input.
In this case, however, the operation limit will be normally several dozens of Hertz, as the counting speed depends on the execution cycle of the programmable controller.

M471 UP/DOWN SELECTION

Designates the counting direction of pair counters C660 and C661.

M472 = ON ..... Up count

M472 = OFF ..... Down count

M472 Start-up signal

Available in case the pair counters C660 and C661 are used as high-speed counters (M470 = ON).

– M472 = ON ..... Counting executed

– M472 = OFF ..... Counting not executed

M473 Flag

M473 is turned on when the current value of the counter is changed from 999999 to 0 (up-count) or from 0 to 999999 (down-count).
The functional instruction F670 K110 is used for resetting.
In case the pair counters are used as down counters, it is possible to make up a 9-digit counter by counting the operation of M473 by the other counters.

M570 Error flag

This flag is turned on when the wrong object element No. is set for condition setting coil of the functional instruction. It is turned off when the setting is correct.
In case a number of function instructions are used, which may influence the operation of this flag, M570 is turned on or off each time the functional instruction is executed.

M571 Carry flag
M572 Zero flag
M573 Borrow flag

M571 ~ M573 are operated when the functional compare instruction is executed for the current counter value, depending upon "Great", "Small" or "Coincidence".

Example In case compare setting is "100"

Current counter value (0 ~ 99) → M573 = ON

Current counter value (100) → M572 = ON

Current counter value (101 ~ 999) → M571 = ON

In addition, the carry flag M571 is also used for functional instructions F670 K113, K115 and serves to detect the interrupt input information.

2-6 Timer (T)

The element Nos. of timer (T) are as shown below, and the operation time is set by the programming panel.

Timer Nos. and setting time

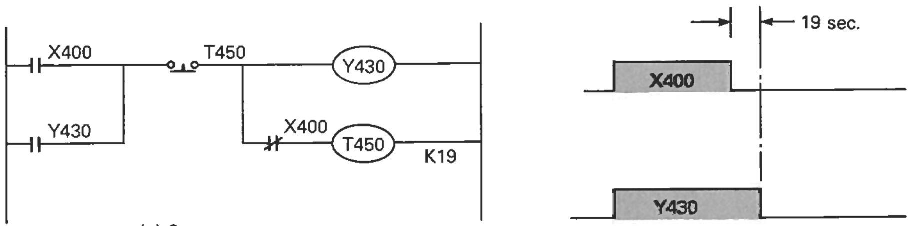
T050 ~ T057
T450 ~ T457 } 24 points ..... 0.1 ~ 999 sec.
T550 ~ T557

* 3-digit setting
Minimum setting unit: 0.1 sec.

T650 ~ T657 8 points ..... 0.01 ~ 99.9 sec.

* 3-digit setting
Minimum setting unit: 0.01 sec.

These timers are provided with a number of normally-open contacts and normally-close contacts for time limit operation.
When the momentary operation contacts are necessary, connect the auxiliary relay coils to the timer coil in parallel, and use the auxiliary relay contacts. Since all these timers are ON-delay timers, insert the circuit as shown below when the OFF-delay timers are necessary.

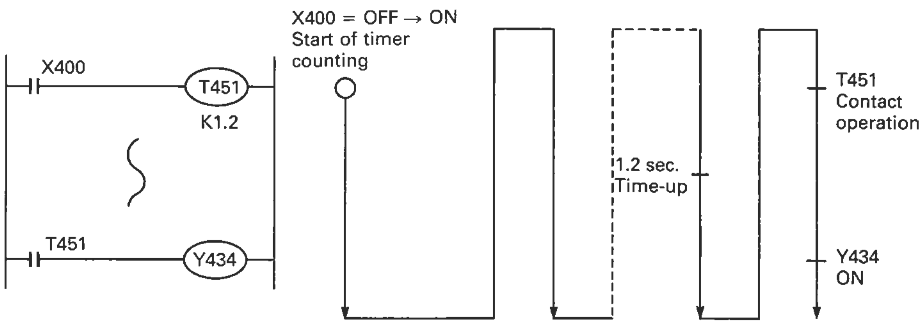


(a) Sequence program (2) Time-chart

Fig. 2-7 Example of OFF-delay timer

NOTE

Timer operation/accuracy
The timer contact is turned on when the initial coil drive instruction is executed in the program after time-up of timer.



continued...

Consequently, the operation accuracy of timer contact can be roughly given by the following formula;

$$T^{+T0}_{-0.1\ (\text{or } 0.01)}$$

T: Timer setting time (sec.)	0.1: In the case of 0.1sec. timer
T0: Execution cycle (sec.)	0.01: In the case of 0.01sec. timer

2-7 Counter (C)

The element Nos. for counter are as shown below, and the operation point is set by the programming panel.
It is also possible to handle the setting value of the digital switch as that of counter.
The digital switch setting value can be handled as the setting value of the counter by the use of functional instruction to be described further on.
In either case, these counters are backed up by the battery to hold the current value even in the event of power failure.
Where the counter value needs not to be retained, it is necessary to reset the function by the use of initialize pulse M71.

(1) 3-digit down counter (C060~067, C460~467, C560~567, C662~667 ... 30 points in total)

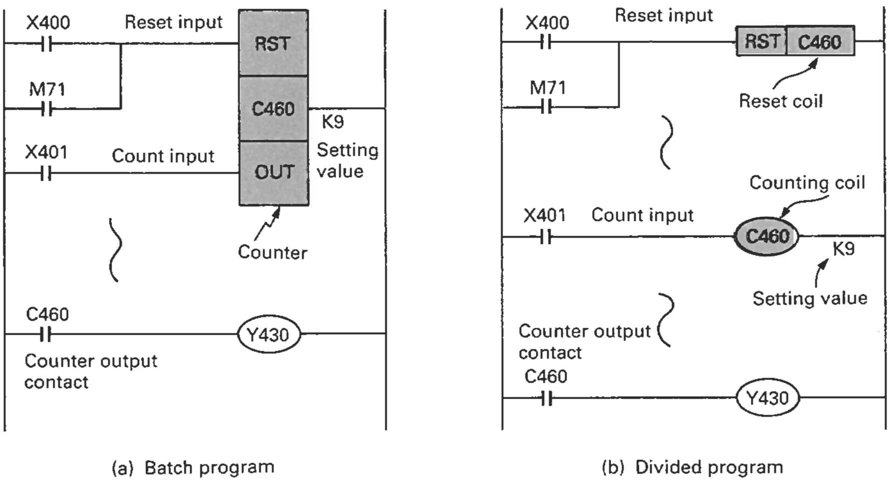


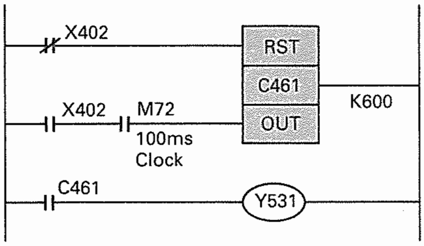
Fig. 2-8 Counter circuit

The counter circuit allows the execution of batch program or division program as shown in Fig. 2-8 (a) and (b).
In either case, the counter is reset by initialize pulse M71 when the operation is started, causing the current counter value to be equal to setting value "9".
The current counter value is decremented each time the count input is turned from OFF to ON.
When the value reaches "0", the output contact C460 is turned on.
The output contact and current value register are changed as follow;

Output contact
Output contact turns on when the current value becomes "0" (at the time when OUT C460 instruction is executed,) and turns off when the reset coil is turned on (at the time when RST C460 instruction is executed).

Current value register

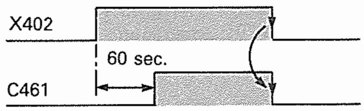
The setting value is written in the current value register at the time when OUT C460 instruction is executed (count input may be ON or OFF) while the reset coil is operating.
The current value is decremented by "1" when the reset coil is inoperative and the count input is turned from OFF to ON.
The value will not be decremented after the value of "0" in the case of 3-digit counter.

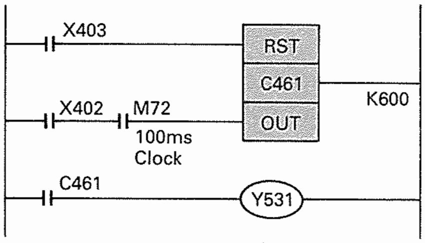


(a) Example of 60-sec. timer

Fig.2-9

When the input X402 is turned on in Fig. 2-9 (a), the counter C461 starts counting the operation of 100ms clock pulse M72.
When the count value reaches the setting value K600 (60 seconds), the contact C461 is turned on.
If the input X402 is turned off, the counter is reset and the output contact is turned off at the same time.

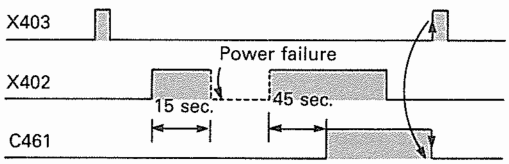




(b) 60-sec. timer
(addition of time before and after power failure)

Fig. 2-9 Timer using counter

An independent reset input X403 is used in the case of Fig. 2-9 (b).
If the power fails during the counter operation, therefore, the counter operation is interrupted, and the counter re-starts the operation after the power restoration. When the total time before the power failure and after the power failure becomes to reach 60 seconds, therefore, the output contact C461 will be turned. on.

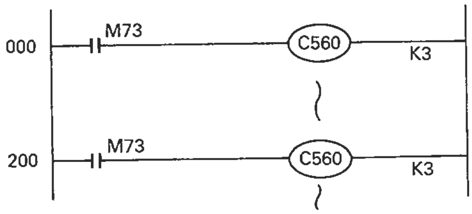


As described above, the counter may be also used as a timer.
10ms clock pulse M73 may be used as count input, however, care needs to be taken to the following points in this case;

NOTE

Counting of 10ms clock pulse

To detect the ON/OFF pulse with a pulse width 5ms, it is necessary that the execution cycle of the programmable controller is less than 5ms.
Even when the execution cycle exceeds 5ms, however, it is possible to count the pulse of 10ms by programming same counting instruction (OUT instruction) of the counter dispersedly so that the operation time during the period is less than 5ms.

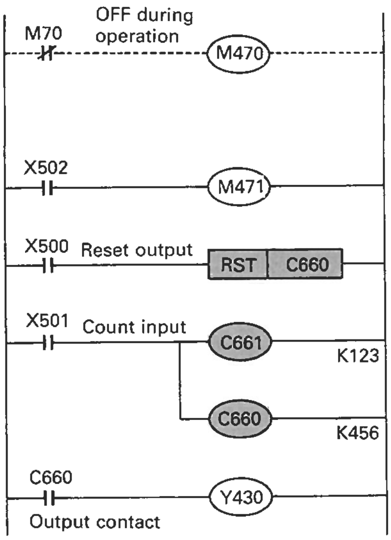


As a gudeline, the counting circuit is programmed in increment of 200 steps as shown at left.

(2) 6-digit UP/DOWN counter
(Pair counter of C660 lower 3 digits, C661 upper 3 digits)

The counters C66O, C66I are used in pairs for a 6-digit UP/DOWN counter.
The 6-digit counter can be used selectively for high-speed counter mode for counting of high-speed pulse (2kHz, max.) or ordinary counter mode.

Ordinary counter mode (internal counting mode)



When M470 is turned off, C660 and C661 are turned to internal counting mode.

The counter appears to be an UP counter when M471 is turned on and a DOWN counter when M471 is turned off.

If RST instruction is provided to C660, C661 is reset automatically.

Set upper 3 digits first.

Set lower 3 digits later.

Y430 is turned on when current value of C660 and C661 reaches "0".

Fig. 2-10 6-digit counter (internal counting mode)

When the reset coil of the counter is actuated, not only the output contact C66O is turned off, but also the setting value is read in the current value register when the counting instruction (OUT C661 or OUT C660) is executed later. (Reading is executed whether the count input is ON or OFF).
When the reset coil becomes inoparative, the current value is changed by one each time the count input X501 is turned from OFF to ON. (123456 → 123457 → 123458 for UP count and 123458 → 123457 ... 0 for DOWN count).
The output contact C660 is turned on and is held when the current value of counter is changed from "000001" to "0" (DOWN count) or from "999999" to "0" (UP count).
Different from the 3-digit counter, the current value is increased/decreased according to the counting input operation ,after output contact C66O is turned on. It is so designed that the shift-up/shift-down flag M473 is set especially when the count value changes from "999999" to "0"(UP count) or from "0" to "999999" (DOWN count).
Under the above internal counting mode, the countable pulse frequency may be restricted by response lagging due to input signal filter, influence of execution cycle by programmable controller, etc. in the same manner as 3-digit counter.

High-speed counter mode (external counting mode)

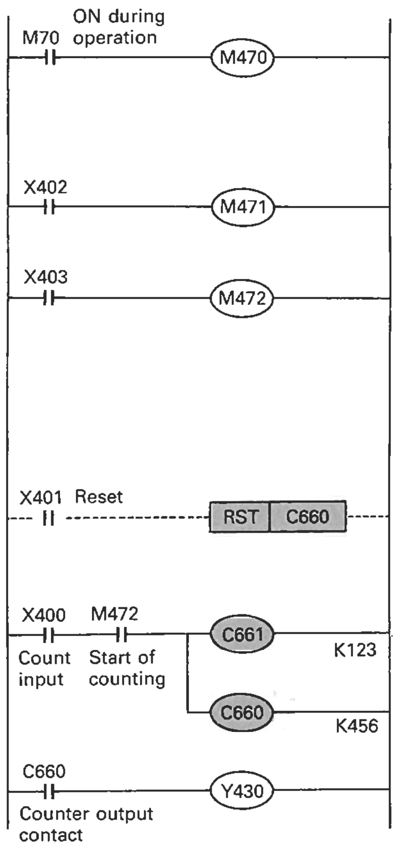


Fig. 2-11 High-speed counter

When M470 is turned on, C660 and C661 are turned to high-speed counter mode.

In case M471 is turned on, the counter becomes UP counter mode.
If it is turned off, the counter becomes DOWN counter mode.

Counting is started when M472 is turned on.

When X401 (fixed) is turned on, current value of C660 and C661 is preset to setting value.

When it is not necessary to open C660 contact, this program is not necessary, and the reset input is fixed X401.

Set upper 3 digits at first.

Set lower 3 digits later.

*Counting is started when X400 (fixed) is turned from OFF to ON.

When current value of C660 and C601 reaches "0", output contact C660 is turned on.

When M470 is turned on, the pair counters C66O and C66l are turned to high-speed counter mode.
Under the high-speed counter mode, the input filter constants of inputs X400 and X401 appears to be a minimum value automatically, making it possible to execute the interrupt counting and interrupt resetting of 2kHz,while executing the program in programmable controller in parallel.
Since the reset input is fixed at X401, the resetting program is not necessary.
When it is necessary to turn off the output contact C660, it is possible to reset it by the use of RST C660 or functional instructions to be described further on.
Since the counter input is similarly fixed at X400, the series contact of X400 and M472 in Fig. 2-11, for instance, executes the same operation even if M70 (ON during operation) is used instead.
The counting input X400 and reset input X401 may be used while M470 is turned off, however, the reset circuit and counting circuit program is necessary and the counting speed will be affected by the input filter constant, execution cycle of programmable controller, as the internal counting mode is activated in this case.
The output contact C660 of counter remains turned on when the current value is turned from "000001" to "0" or from "999999" to "0" after the count input X400 is turned on.
The current value of counter is increased/decreased after the output contact has been turned on, depending upon the count input even after the output contact has been turned on, and the shift-up/shift-down flag M473 is set when the current value of counter is changed from "999999" to "0" or from "0" to "999999". Fetching of setting value to current value register is made at the time when the reset input input X401 is turned on.
(For resetting of output contact and shift-up/shift-down flag, refer to page-83).

2-8 State (S)

"State" (S) is an essential element in programming the process stepping type control with ease, it is used with step ladder instruction STL (to be described further on) in combination.

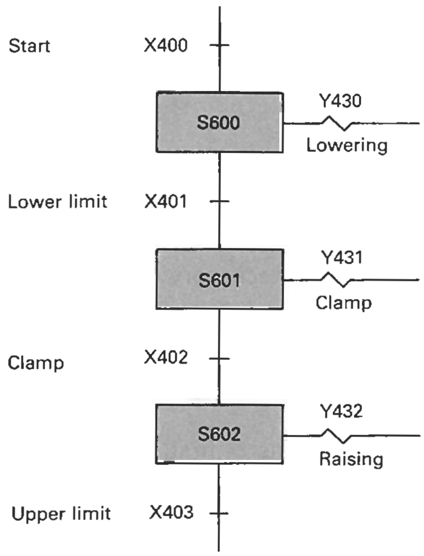


Fig. 2-12 Process progress flow

If the start signal X400 is turned on in the process stepping control as shown in Fig. 2-12, the state S600 is set (ON), causing the lowering valve Y430 to actuate.

If the lower limit switch X401 is turned on consequently, the state S601 set (ON), causing the state S600 to be reset (OFF) automatically.

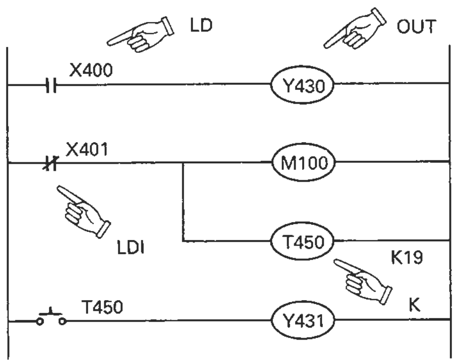
As described above, the "State" is a element used to control the machine orderly by storing where the machine operation process is.
S600 ~ S647 (octal) are all backed up by the battery.

The state No. is optional and S610 may be used in substitute for S601 in the above example, for instance. Each state (S) is provided with a number of normally-open contacts and normally-close contacts and may be used optionally within the programmable controller.
When the step ladder instruction is not used, the state (S) may be used as an ordinary auxiliary relay (battery back-up).

State Nos.
S600~S647 (40 points, octal)
Battery back-up

3 BASIC SEQUENCE INSTRUCTIONS

LD	LD (load)	: Bus bar connecting instruction for normally-open contact
LDI	LDI (load inverse)	: Bus bar connecting instruction for normally-close contact
OUT	OUT (out)	: Coil drive instruction



0	LD	X400	Connection to bus bar
1	OUT	Y430	
2	LDI	X401	
3	OUT	M100	Drive instruction
4	OUT	T450	Timer drive instruction
5	K	19	Setting of constant
6	LD	T450	
7	OUT	Y431	

Fig. 3-1 LD, LDI, OUT

– Explanation –

(1) LD and LDI instructions are used for contact connecting to bus bar.
In addition, these instructions are also used together with ANB instruction (to be described further on) in combination at the start of branch.

(2) OUT instruction is a coil drive instruction used for output relay, auxiliary relay, timer, counter, state and functional instruction, but not for input relay.

(3) OUT instruction in parallel may be used for any times repeatedly.
(Corresponding to OUT T450, following OUT M100 in Fig. 3-1)

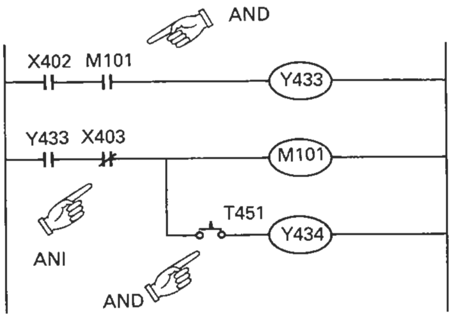
(4) It is necessary to set proper constant after OUT instruction for timer, counter and functional instruction coil.
The setting of contact (K) requires one step.

– Object elements –

LD, LDI: X, Y, M, T, C, S,
OUT: Y, M, T, C, S, F

AND — AND (And) : Series connecting instruction for normally-open contact

ANI — ANI (And inverse) : Series connecting instruction for normally-close contact



0	LD	X402	
1	AND	M101	← Series contact
2	OUT	Y433	
3	LD	Y433	
4	ANI	X403	← Series contact
5	OUT	M101	
6	AND	T451	← Series contact
7	OUT	Y434	← Continuous out

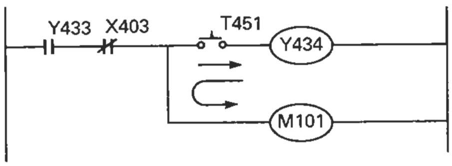
Fig. 3-2 AND, ANI

– Explanation –

(1) AND and ANI instructions are used for series connection of one contact.
The number of contact to be connected in series is not limited, and the instruction can be used for any time in succession.

(2) The execution of OUT to other coil through the contact after the execution of OUT instruction is called "Continuous OUT".
(OUT Y434 in Fig. 3-2)
The continuous OUT as described above may be used for any times repeatedly if only the order of circuit design is correct.

– Precaution –

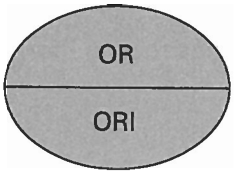


It is possible to drive the coil through the contact T451, following OUT M101 as OUT Y434 shown in Fig. 3-2, however, it can not be so programmed reversely as shown in Fig. 3-3.

Fig. 3-3 Wrong circuit

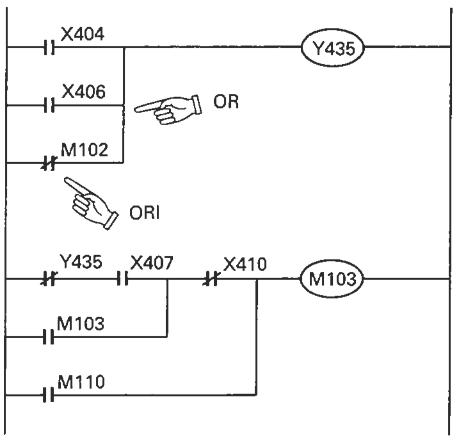
The number of series contact and continuous OUT to use is not limited, however, there is a limitation in the graphic programming panel, printer function, etc.
It is recommended to use less than 10 contacts and one coil for one line, and less than 7 lines.

– Object elements –
AND, ANI: X, Y, M, T, C, S



OR (Or) : Parallel connecting instruction for normally-open contact

ORI (Or inverse) : Parallel connecting instruction for normally-close contact



0	LD	X404	Parallel connection
1	OR	X406	
2	ORI	M102	
3	OUT	Y435	
4	LDI	Y435	Parallel connection
5	AND	X407	
6	OR	M103	
7	ANI	X410	
8	OR	M110	
9	OUT	M102	

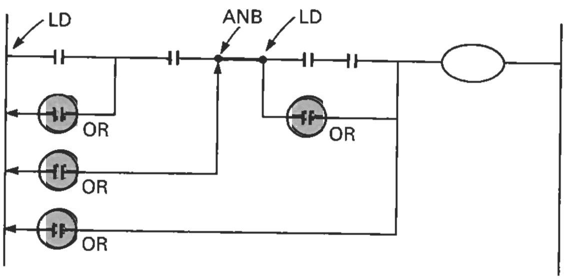
Fig. 3-4 OR, ORI

– Explanaition –

(1) OR and ORI instructions are used for parallel connection of one contact.
Use ORB instruction to be described further on when connecting the series circuit (in which more than two contacts ar connected in series).

(2) OR and ORI instructions are connected in parallel to the step including the former LD and LDI instructions.
The number of parallel connection is not limited.

– Precaution –

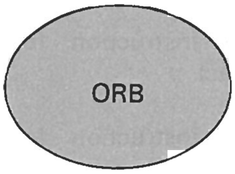


The parallel connection by OR and ORI instructions are basically connected to the former LD/LDI point.
After the execution of ANB instructions to be described later, however, it is connected to LD and LDI points before.

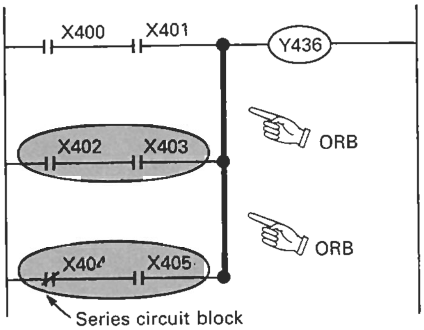
Fig. 3-5 Parallel connection point

– Object elements –

OR, ORI: X, Y, M, T, C, S



ORB (Or block) : Parallel connection instruction for branch circuit



Desirable program				Undesirable program			
0	LD	X400		0	LD	X400	
1	AND	X401		1	AND	X401	
2	LD	X402		2	LD	X402	
3	AND	X403		3	AND	X403	
4	ORB		←	4	LDI	X404	
5	LDI	X404		5	AND	X405	
5	AND	X405		6	ORB		←
6	ORB		←	7	CRB		←
7	OUT	Y436		8	OUT	Y436	

Fig. 3-6 ORB

– Explanation –

(1) The circuit in which more than two contacts are connected is called "Series circuit block".
When connecting the series circuit block parallelly, USE LD and LDI instructions for the start of branch, and ORB instruction for the end of branch.

(2) ORB instruction is an independent instruction like ANB instruction, etc. which is not accompanied by any element No.

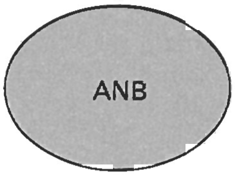
– Precaution –

(1) Where a number of parallel circuits are used, there is no limitation in the number of parallel circuit to use if ORB instruction is used for each block.

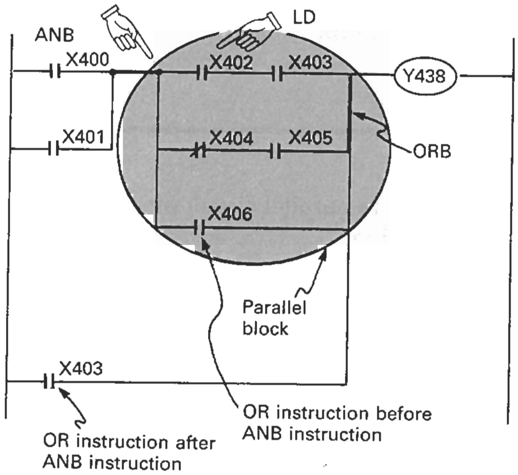
(2) ORB instruction may be used collectively, however, bear in mind that the number of LD and LDI instructions to be used repeatedly in this case is limited to less than 8 times in one line.
(this programming method is undesirable)

– Object elements –

ORB: Independent instruction not accompanied by element No.



ANB (And block) : Instruction to connect branch start instruction in series to previous circuit



0	LD	X400	
1	OR	X401	
2	LD	X402	Branch start point
3	AND	X403	
4	LDI	X404	
5	AND	X405	
6	ORB		Completion of parallel block
7	OR	X406	
8	ANB		Serires connection to previous circuit
9	OR	X403	
10	OUT	Y438	

Fig. 3-7 ANB and LD, OR

– Explanation –

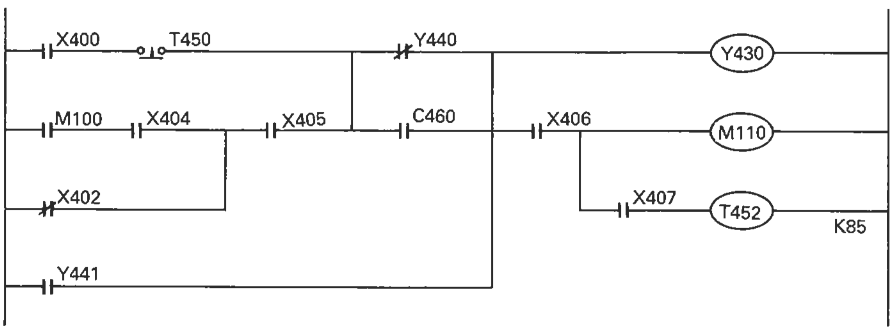
(1) Use ANB instruction to connect the branch circuit (parallel circuit block) in series to the previous circuit.
Use LD and LDI instructions for the start of branch to connect in series to the previous circuit by the use of ANB instruction after the parallel circuit block is completed.

(2) If the multiple number of parallel circuit blocks are connected in series to the previous circuit in succession, the number of ANB to use is not limited.
And instruction may be used totally, however, it is limited like ORB instruction in the number of LD, LDI instruction to use; (less than 8 times)

– Object elements –

ANB: Independent instruction not accompanied by elements No.

All the basic sequences can be programmed by the use of instructions described so far.



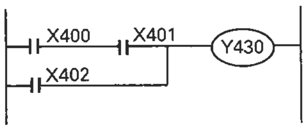
0	LD	X400	6	ORB		12	AND	X406
1	ANI	T450	7	LDI	Y440	13	OUT	M110
2	LD	M100	8	OR	C460	14	AND	X407
3	AND	X404	9	ANB		15	OUT	T452
4	ORI	X402	10	OR	Y441	16	K	85
5	AND	X405	11	OUT	Y430			

Fig. 3-8 Circuit and program example

NOTE

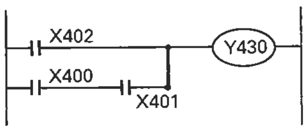
***Program procedure and number of step**

(1) Write the circuit with a number of series contact at upper side



LD X400
AND X401
OR X402
OUT Y430

(a) Desirable circuit



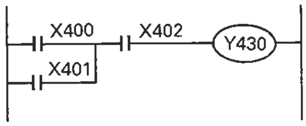
LD X402
LD X400
AND X401
ORB
OUT Y430

(b) Undesirable circuit

Fig. 3-9 Parallel circuit

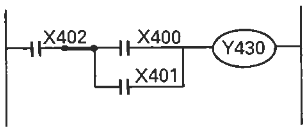
Extra ORB instruction is necessary in the circuit example shown at the lower side.

(2) Write the parallel circuit at left hand side.



LD X400
OR X401
AND X402
OUT Y430

(a) Desirable circuit

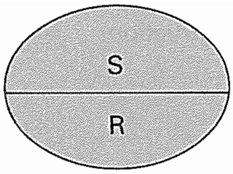


LD X402
LD X400
OR X401
ANB
OUT Y430

(b) Undesirable circuit

Fig. 3-10 Parallel circuit

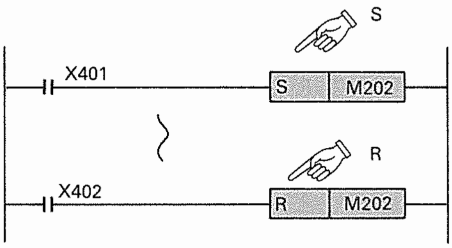
Extra ANB instruction is necessary in the circuit example shown at the lower side.



S (Set) : Operation holding instruction

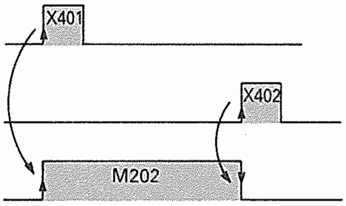
R (Reset) : Operation hold resetting instruction

These instruction are applicable to the output relay Y, state (S) and auxiliary relays M200~M377, and serve to hold or reset the operation.




```
LD   X401
S    M202
 ⟩  ←———— Other program may exist.
LD   X402
R    M202
```

(a) Set/reset



Once X401 is turned on, M202 remains turned on even if X401 is turned off.
Once X402 is turned on, M202 remains turned off even if X402 is turned off.

(b) Timechart

Fig. 3-11 Set/reset operation

– Explanation –

(1) When S-instruction is used, the coil holds the operation with its self-holding function. The self-holding is reset when R-instruction is used. The operation is as shown in the timechart.

(2) Whichever S-instruction or R-instruction may be programmed earlier, however, the instruction executed later will become effective.
When programming with S-instruction and R-instruction used continuously without the intervention of the other program, therefore, the later program will be executed preferentially when both X401 and X402 are turned on.

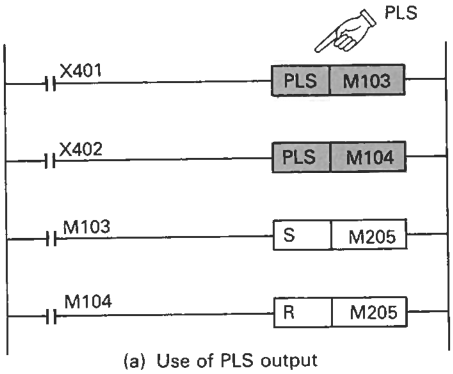
– Object elements –

S, R: Y, M200~M377, S

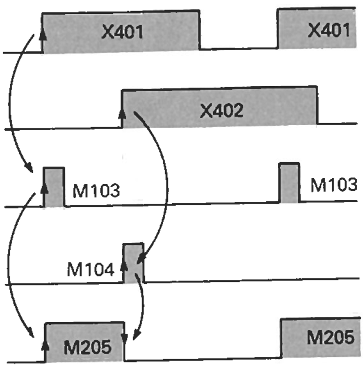
PLS

PLS (pulse) : Differential output instruction

0	LD	X401
1	PLS	M103
2	LD	X402
3	PLS	M104
4	LD	M103
5	S	M205
6	LD	M104
7	R	M205



(a) Use of PLS output



(b) Timechart

Fig. 3-12 Example of pulse output circuit

– Explanation –

(1) When PLS instruction is used for auxiliary relay M, the output contact is operated only for one execution cycle.

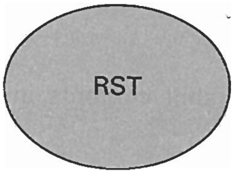
(2) The instruction may sometimes be used for reset input of counter or shift register, input of set/reset instruction input of, data instruction, etc.

(3) Fig. 3-12 shows circuit which sets/resets M205 when X401 and X402 are rise (OFF → ON).

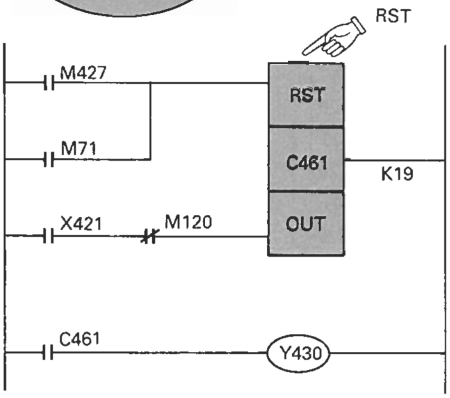
(4) If PLS instruction is jumped by the jump instruction during pulse output, the pulse output is held turned on.

– Object elements –

PLS: M100 ~ M377



RST (Reset) : Instruction used to clear counter or shift register




```
0  LD   X427
1  OR   M71
2  RST  C461
3  LD   X421
4  ANI  M120
5  OUT  C461
6    K    19
7  LD   C461
8  OUT  Y430
```

Other program may be inserted in midway.

Fig. 3-13 Counter circuit

– Explanation –

(1) RST instrucion is used to turn off the output of counter or return the current value to the setting value.
It is also used to clear the contents of register to be described further on.
Since RST instruction is executed preferentially in either case, the counter input or shift input will not be accepted while RST input is being continued.

(2) Since the program for reset circuit is independent of that for shift input circuit of shift register or for data input circuit, it is possible to freely modify the program sequence or divide the program. (Refer to page-36).
It is not possible to fetch the setting value if OUT instruction of counter is jumped by the jump instruction or is opened by the step ladder contact when the program is divided.

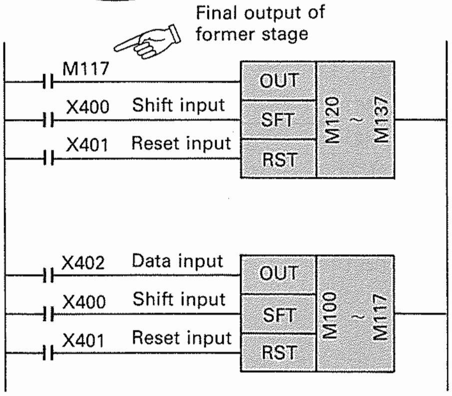
(3) The counter and shift-register located in the battery back-up area are protected against power failure.
In case holding function is not necessary, it is necessary to reset the counter or shift register by the use of initialize pulse M71 before starting operation.

– Object elements –

RST: M100, M120, M140, M160, C(excluding C661)
M200, M220, M240, M260
M300, M320, M340, M360

SFT

SFT (Shift) : Instruction used to shift contents in shift register



0	LD	M117	Latter stage
1	OUT	M120	
2	LD	X400	
3	SFT	M120	
4	LD	X401	
5	RST	M120	
6	LD	X402	Former stage
7	OUT	M100	
8	LD	X400	
9	SFT	M100	
10	LD	X401	
11	RST	M100	

Fig. 3-14 Cascading connection of shift register

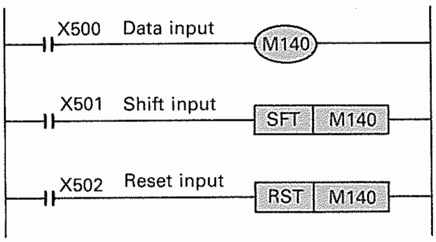
– Explanation –

(1) The shift register is operated by 16 auxiliary relays in combination as described on page-14. The head auxiliary relay number will represent the shift register.

(2) In case more than two shift registers are connected longitudinally, program the latter stage at first as shown in Fig. 3-14.
Use the final stage output of former stage shift register for the data input of latter stage shift register.

(3) If SFT instruction is not used, these auxiliary relays may be used for ordinary applications.
In addition, it is possible to control M200~M377 independently by use of set/reset instruction, and further to use SFT instruction with these auxiliary relays.

(4)

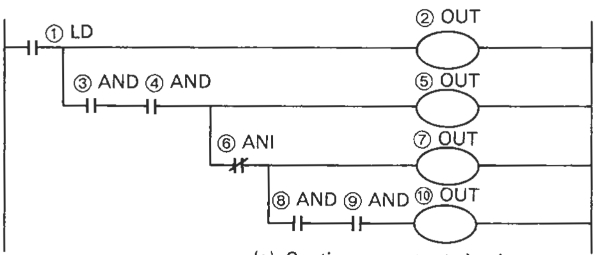


Each input may be programmed dividedly as shown in left figure.
Its programming sequence is not specially restricted.
It is also possible that the program of other sequence may exist in its halfway.

– Object elements –

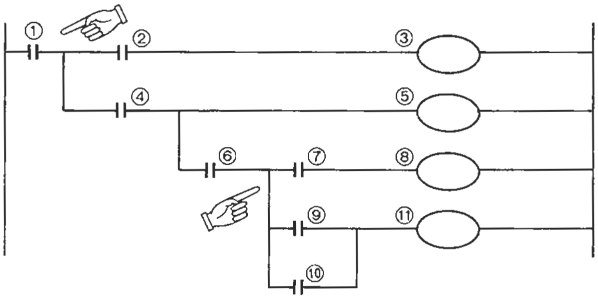
SFT: M100, M120, M140, M160
M200, M220, M240, M260
M300, M320, M340, M360

MC	MC (Master control)	: Common series contact connecting instruction
MCR	MCR (Master control reset)	: Reset instruction for MC instruction



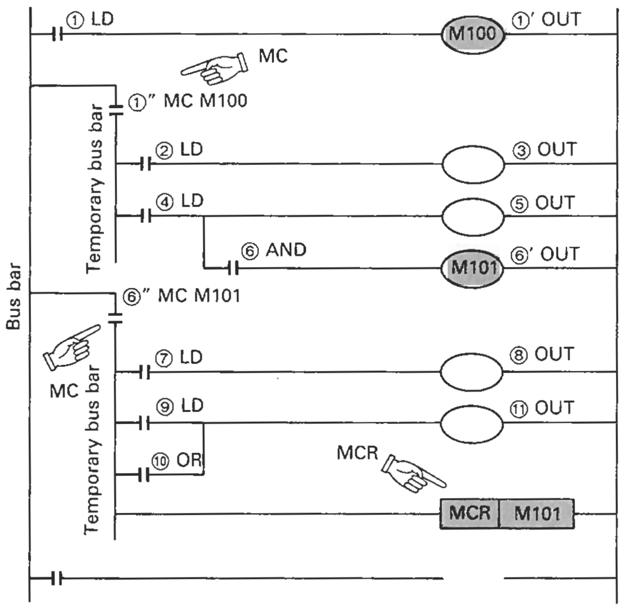
(a) Continuous output circuit

As described already, the continuous output circuit as shown in Fig. 3-15(a) can be programmed for any times unless the program is made in wrong sequence. (See page-50).



(b) Multiple output circuit

In the case of multiple output circuit including a series contact after the branch as shown in Fig. 3-15(b), however, direct programming will not be allowed within the range of instruction described so far.
MC instruction and MCR instruction are prepared to solve such problem. These instructions can be used for auxiliary relays M100 ~ M177

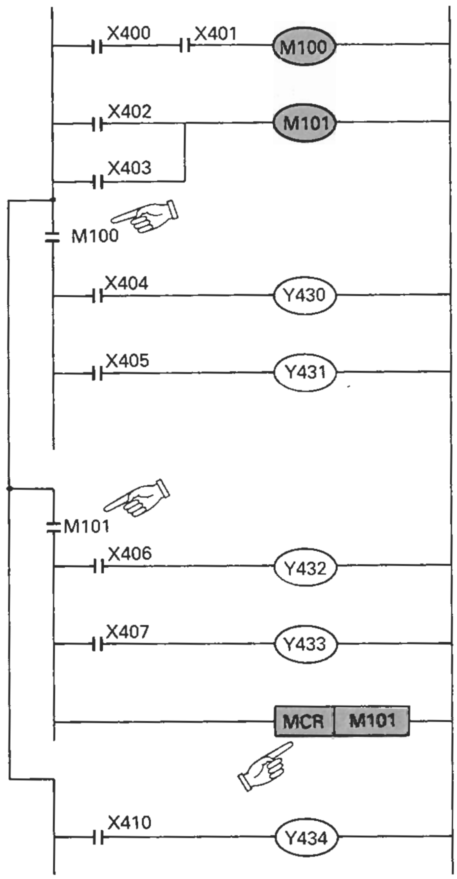


(c) Use of MC/MCR instructions

Fig. 3-15 Multiple output branch circuit

The circuit in Fig. 3-15(c) expresses Fig. 3-15(b) with MC instruction.
The MC contacts such as MC M100, MC M101, etc. are one normally-open contact respectively connecting to the bus bar necessarily. The contact leading to such contact will be LD (LDI).
In other words, the bus bar shifts to the backward of the MC contact.
MCR instruction is used to return LD (LDI) instruction to the original bus bar.

The program example for MC and MCR instruction is shown in Fig. 3-16.



0	LD	X400
1	AND	X401
2	OUT	M100
3	LD	X402
4	OR	X403
5	OUT	M101
6	MC	M100
7	LD	X404
8	OUT	Y430
9	LD	X405
10	OUT	Y431
	~	
20	MC	M101
21	LD	X406
22	OUT	Y432
23	LD	X407
24	OUT	Y433
	~	
30	MCR	M101
31	LD	X410
32	OUT	Y434

Fig. 3-16 MC, MCR circuit

– Precaution –

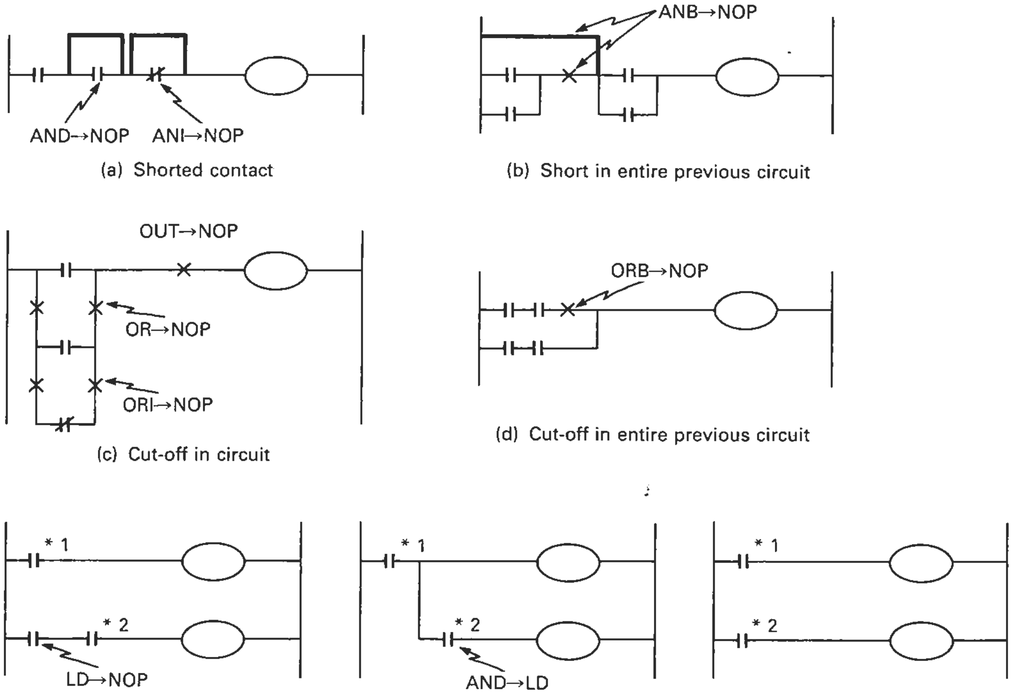
When the master control instruction is used for MANUAL/AUTO sequence changing, be careful not to execute the double output.
(Bear in mind that the double output will occur if Y430 is programmed under AUTO mode and Y430 is also programmed under MANUAL mode).
Use the jump instruction to be described further on for the execution of double output.

– Object elements –

MC, MCR: M100 ~ M177

NOP

NOP (Nop) : Non-processing instruction



(a) Shorted contact

(b) Short in entire previous circuit

(c) Cut-off in circuit

(d) Cut-off in entire previous circuit

(e) Longitudial connection to previous out

Fig. 3-17 Modification of circuit by NOP instruction

– Explanation –

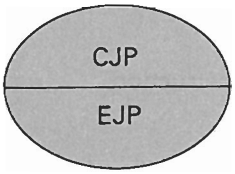
(1) If NOP instruction is inserted in the cource of program, it is possible to minimize the step number change when modifying/adding the program.
In addition, it is possible to modify the circuit by replacing the instruction already written, for NOP instruction.

(2) Bear in mind that changing LD, LDI, ANB, ORB, etc. to NOP instruction will cause the circuit configuration to be greatly changed.

(3) All instrcutions will apear to NOP when the program all clear is executed.

– Object elements –

NOP: Independent instruction not accopanied by element No.



CJP (Conditional jump) : Jump start instruction

EJP (End of jump) : Jump destination designation instruction

This jump instruction is prepared as a instruction not to execute a part of program.
The jump destination numbers are 64 points ranging from "700" to "777" (octal No.).

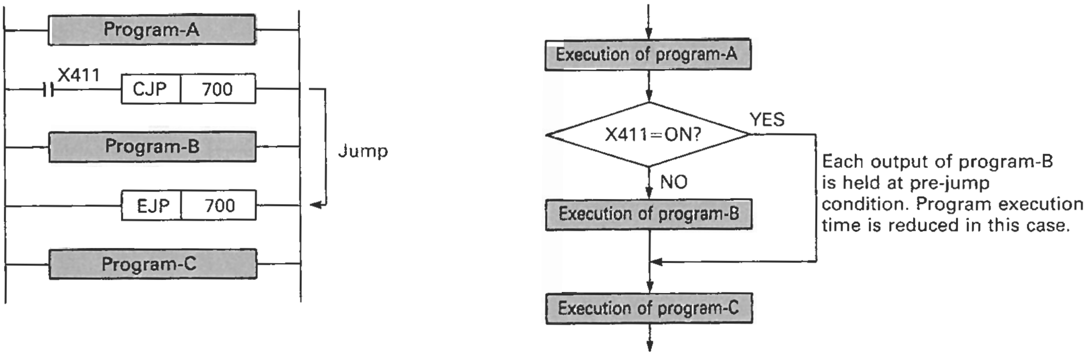
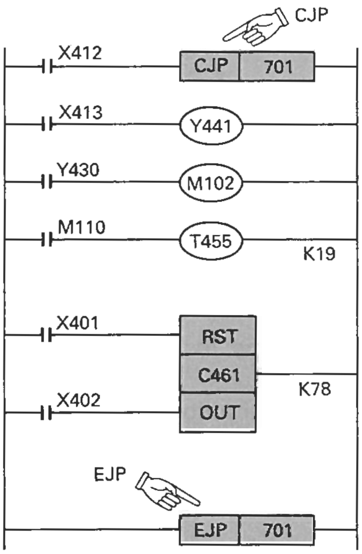


Fig. 3-18 CJP, EJP



0	LD	X412	
1	CJP	701	Jump starts when X412 is turned on.
2	LD	X413	
3	OUT	Y441	Remain ON/OFF condition before jump.
4	LD	Y430	
5	OUT	M102	
6	LD	M110	
7	OUT	T455	Timer will not operate even if M110 is turned on after jump. (Refer to Fig. 3-20).
8	K	19	
9	LD	X401	
10	RST	C461	
11	LD	X402	Count operation is interrupted during jump, and current value is held
12	OUT	C461	
13	K	78	
14	EJP	701	

Fig. 3-19 Jump circuit

When OUT, RST, SFT instructions of counter or shift register are programmed dividedly, care needs to be paid to the operation caused by dividing these instructions to different jump area.

– Object Nos. –
CJP, EJP: 700 ~ 777

– Explanation –

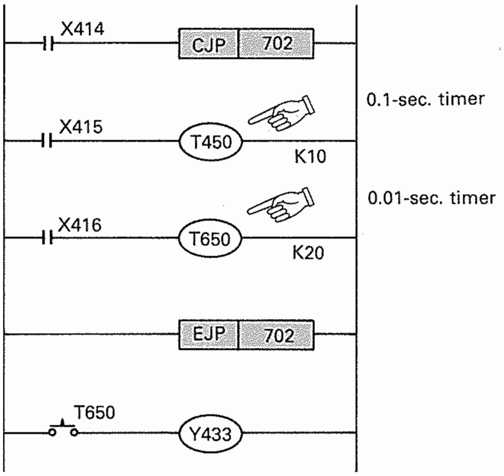


Fig. 3-20 Timer and jump

In case X414 is turned on for jumping while X451 and X416 are turned off, the timers T450 and T650 will not operate.

If X414 is turned on for jumping while the timer is counting and X415 and X416 have been already turned on, however, each timer will operate as follows;

1) T50~T57, T450~T457, T550~T557 (0.1-sec. timer)

The timers interrupt time counting, and continue time counting after Jump has been reset.

2) T650~T657 (0.01-sec. timer)

Time counting is executed continuously, however, the output contact will not operate upon elapse of timer setting.

When the jump is reset, the output contact is turned on at the time of coil instruction execution.

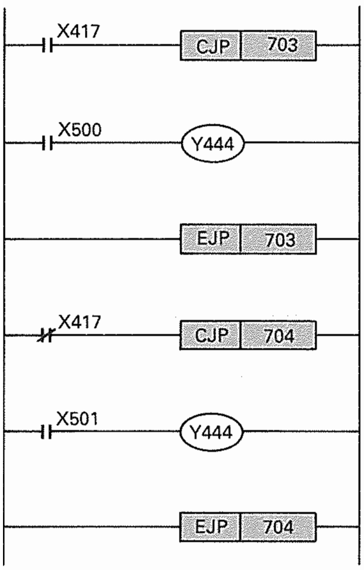


Fig. 3-21 Double coil and jump

In case the same OUT instruction is used in the jump instructions which operates inconsistently, OUT instruction under execution is treated preferentially.

If X417 (a-contact) is turned on in Fig. 3-21, Y444 will operate according to the operation of X501.

While X417 (a-contact) is turned off, Y444 will operate according to the operation of X500.

If the pulse instruction during the execution of pulse output is jumped by jump instruction, the pulse output is kept generated.

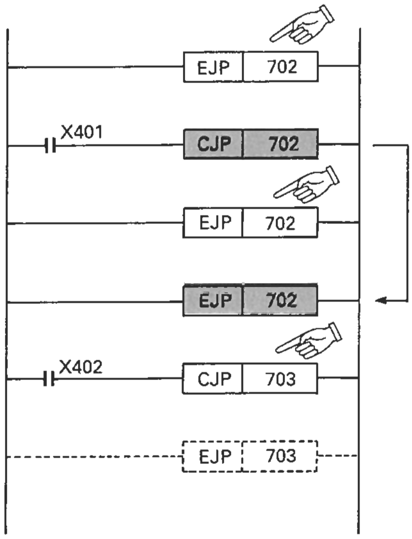


Fig. 3-22 Wrong program

It is not possible to use EJP instruction before CJP instruction. EJP, if used, will be ignored.

In case the multiple number of the same EJP are used, the final EJP is valid and other EJPs in the midway are ignored.

Bear in mind that negligence to write EJP instruction will cause CJP instruction itself to be ignored.

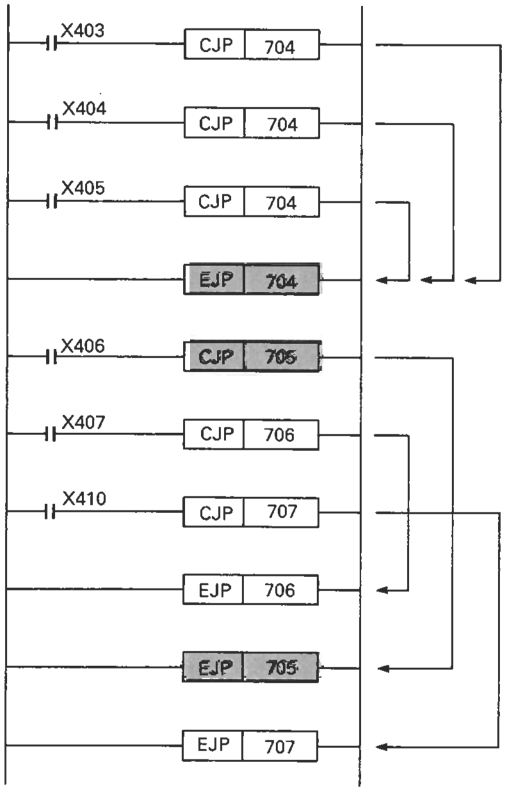


Fig. 3-23 Multiple jump instructions

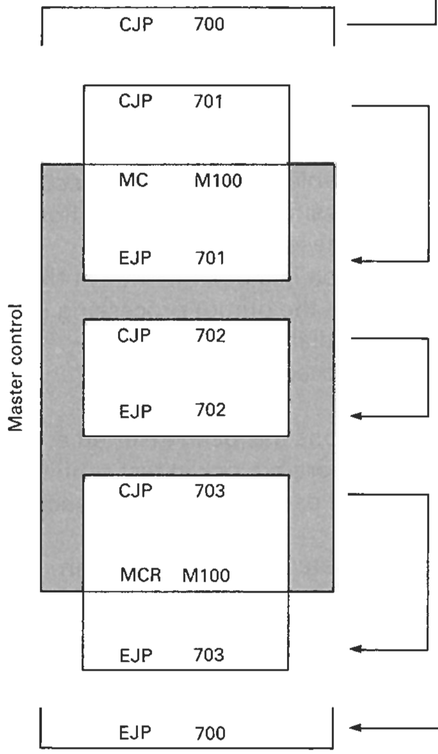
The multiple number of jump instructions having the same jump destination may be programmed in the same No.
If any one of CJP is turned on when monitoring, all CJP 704s are displayed in "ON".

The jump area for "706" is included in the jump area of "705".

The jump area of "705" and that of "707" are partially overlapped.

While X406 is turned on, CJP 706, CJP 707 will become ineffective.

If X407 is turned on, CJP 707 will become ineffective.



(1) Jump from outside MC to outside MC

Jump is applicable regardless of MC operation.

(2) Jump from outside MC to inside MC

Jump is applicable regardless of MC operation.
The circuit after jump is executed even if MC M100 is turned off, regarding it as ON.

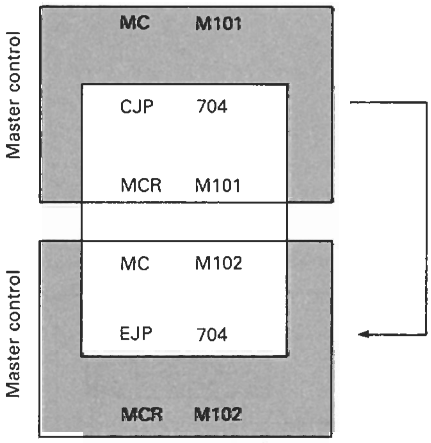
(3) Jump from inside MC to inside MC

This jump is executable when MC M100 is turned on, and not valid while it is turned off.

(4) Jump from inside MC to outside MC

Jump is executable if MC M100 is turned on, however, MCR will become ineffective in this case.

Jump is not executable if MC M100 is turned off.



(5) Jump from inside MC to inside other MC

Jump is executable as long as MC M101 is turned on.
The circuit after jump is executed even if MC M102 is turned on or off, regarding it as ON.
MCR M101 is ignored in this case.

Fig. 3-24 Master control and jump

END

END (End) : Instruction used at end of program

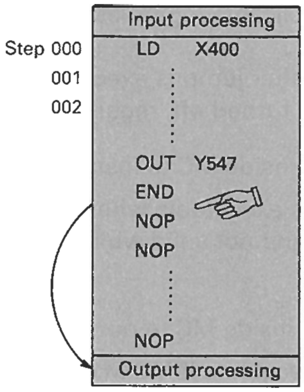


Fig. 3-25 END instruction

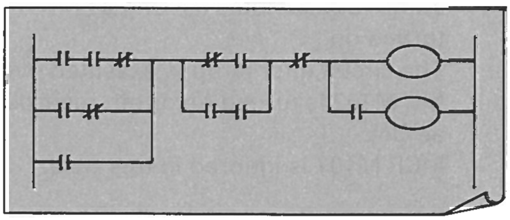
The programmable controller will execute the input processing, program execution and output processing repeatedly.
If END instruction has been written at the end of program, the output processing is executed immediately without the execution of succeeding extra steps.

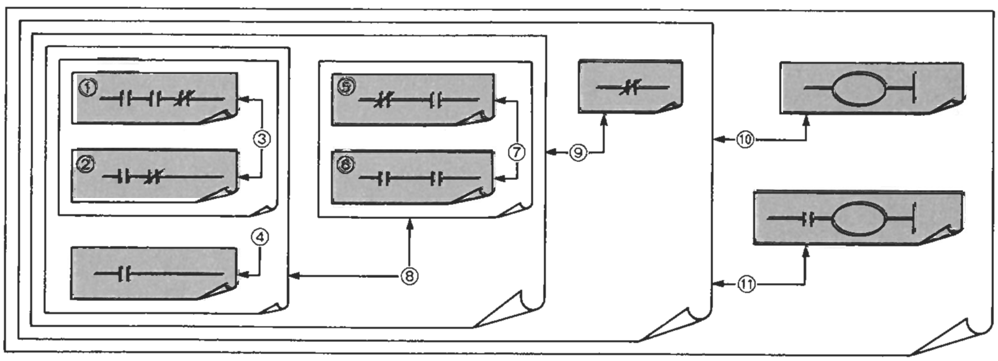
If END instructions has been inserted at the end of each program block in test running, the operation of each block can be checked sequentially.
In this case, delete the END instruction sequentially after checking the operation in previous circuit block.

– Object elements –
END: Independent instruction not accompanied by element No.

NOTE

Proceed programming from left to right and upperside to lower side.





The object elements for each instruction and element No. of each PC are listed at the paragraph "Summary".

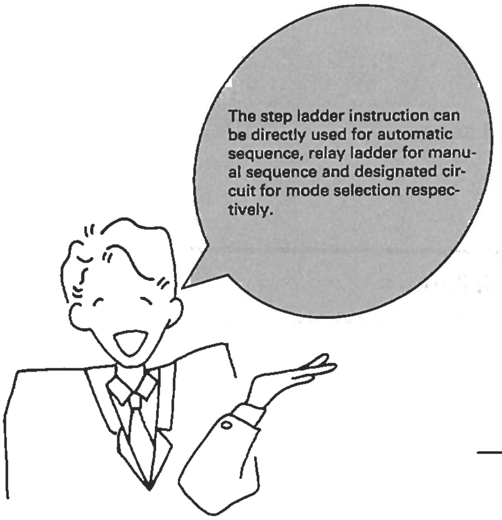
4 STEP LADDER INSTRUCTION

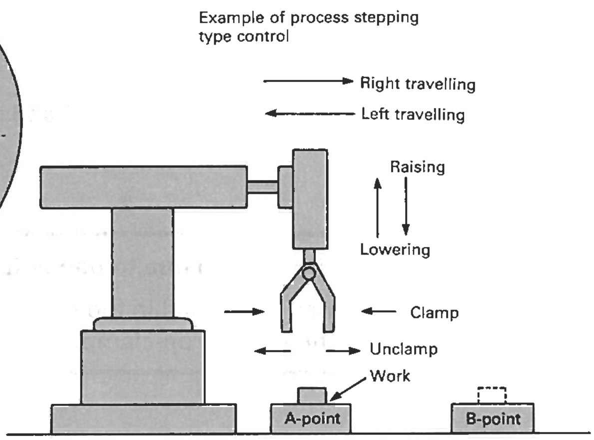
The entier sequence program can be roughly divided into "General sequence" centered on the mode selection, "Manual sequence" and "Automatic sequence".
This section describes the contents of step ladder instruction and handling of the above three sequences.

4-1 Overall configuration of circuit

(1) Purpose of step ladder instruction

The step ladder instruction is a useful instruction which allows easy sequence designing on the basis of state transition diagram expressing the machine operation, by the use of a simple programming panel.
This will not require the sequence designing of the process stepping type control, by which it makes possible even for beginners of the sequence designing, to readily make full use of the programmable controller.
The manual control sequence and mode selection sequence suitable to the conventional relay ladder sequence need to be designed with the relay ladder as in the past, therefore, they can be used in combination.
Even for skilled engineers, the use of step ladder instruction further improves the design efficiency, thus making the test operation/adjustment easy.
In addition, the program prepared on the basis of this procedure is easy for the third party to understand, and greatly reduces the number of program step required.
Furthermore, the program based on the state transition diagram is re-written automatically to the relay ladder type sequence and the circuit diagram is printed out by the printer when the HGP (handy graphic programming panel) or program loader is used.





Work transfer (A→B) mechanism

(2) Allotment of input/output unit

Fig. 4-1 shows the allotment of load and sensor (mounted on robot hand) by the input/output No. of PC. A double solenoid valve is used for raising, lowering and right-travelling/left-travelling. Once the lowering (right-travelling) output is turned on, therefore, the current position is kept continuously even if it is turned off.
The same applies to the raising (right-travelling), and the current position is retained as long as the reverse operation solenoid is energized.
On the other hand, a single solenoid valve is used for clamping.
The clamping condition occurs while the clamp output is driven and the unclamping condition occurs when it is interrupted.
Each arm is provided with limit switches for upper/lower limit and left/right limit.
Since the clamp is not provided with any limit switch, the timer is driven within the programmable controller as soon as the clamp solenoid is turned on, thereby the clamp is regarded as completed upon elapse of designated time.
The un-clamp is also checked in the similar procedure.

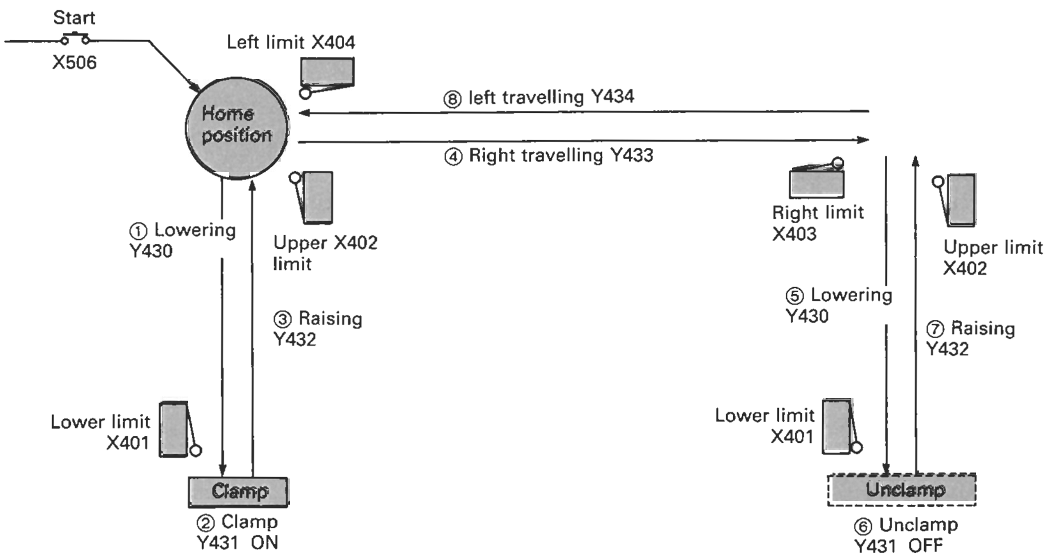


Fig. 4-1 Load and sensor

NOTE

Precaution on falling of load due to power failure

In case the load may fear to fall in the event of power failure, change the program so that the valve is un-clamped when the power is applied.

(3) Allotment of operation input

The example of the machine operation modes are as follows.
The operation panel is made up as shown in Fig. 4-2 for this purpose.
The start/emergency stop sequences are provided outside the programmable controller and the power to the external loads is turned on/off according to these sequences.

MANUAL

- Single operation –

  Mode to turn on/off each load by each pushbutton

- Home position returning –

  Mode to return the machine to its home position automatically when the home position returning pushbutton is depressed

AUTO

- Stepping

  Mode to advance the operation by each process at each time the start button is pressed

- One-cycle operation –

  When the start button is depressed at the home position, the one-cycle operation is executed automatically, and the machine is stopped at the home position thereafter.
  If the stop button is depressed in the course of operation, the machine is stopped at the process.
  If the start button is depressed, the operation is continued from the process and stopped automatically at the home position.

- Continuous operation –

  When the start button is depressed at the home position, the continuous operation is performed repeatedly.

  If the stop button is depressed, the machine moves to the home potion and then stops position.

Since the single manual operation is also available by the use of programming panel, it is not always necessary to prepare the pushbutton for all loads.

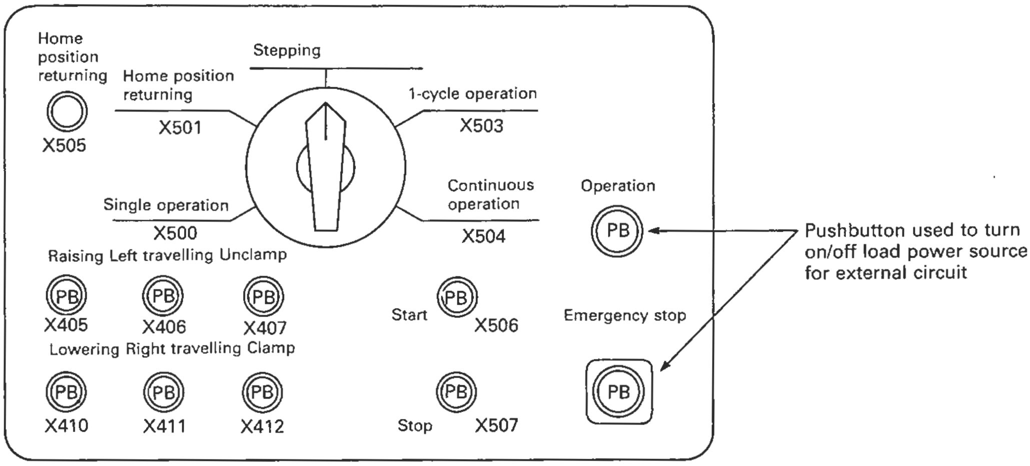
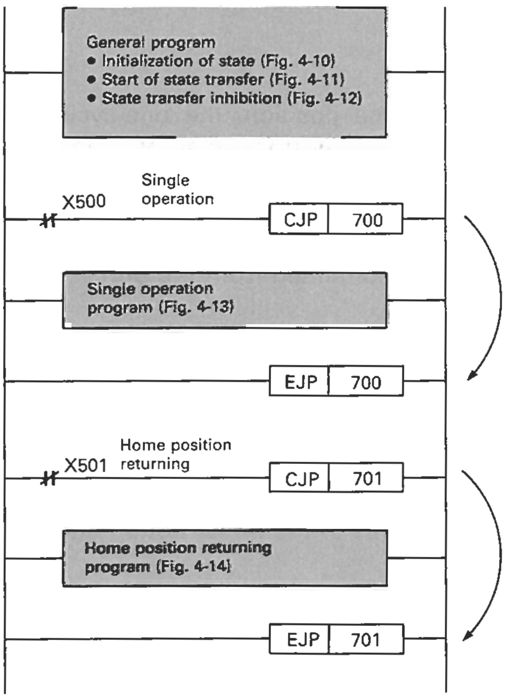


Fig. 4-2 Example of operation panel

(4) Overall sequence configuration

The overall configuration of manual sequence (single operation, home position returning) and automatic sequence is as shown in Fig. 4-3. It is recommended to program on the basis of the following figure.



CONTINUED

As shown in Figs. 4-10 through 4-12, program the general sequences such as initialization of state, start of state transfer, and state transfer inhibition, etc.

When the single operation mode is selected and X500 is opened, single operation program to be described further on will be executed. While the other mode is selected, X500 is closed, by which the program is jumped.

If the home position returning mode is selected, and X501 is opened, the home position returning program to be described later will be executed.

Under the other modes, X501 is closed and the program is jumped, causing the operation not to be executed.

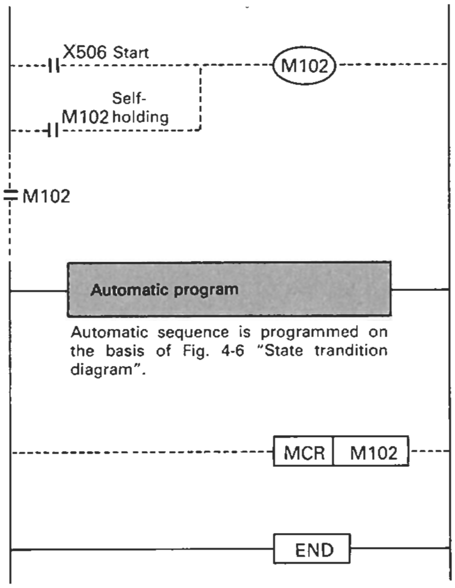


Fig. 4-3 Overall sequence

The automatic program will not be executed until the start button is depressed.

This program will not be necessary when the machine is re-started from the home position after power restoration.

automatic program;
Make it a rule to express the automatic program by the state transition diagram shown in Fig. 4-6 or step ladder diagram shown in Fig. 4-16. When the circuit diagram is printed out by the printer, it will be expressed in step ladder diagram format.

4-2 Automatic sequence program

(1) Load drive chart

Fig. 4-4 shows which load to operate at each process of the robot hand operation described previously.
The lowering solenoid valve Y430 is turned on in the 1st lowering process.
In the clamp process, the clamp solenoid valve Y431 is set and drive the timer T450 at the same time.
The similar operation is performed thereafter to complete a series of operations from the initial condition to succeeding initial condition.
The clamp output Y431 is held actuated after it has been set at the clamp process until the output is reset at the un-clamp process.
On the other hand, the timer and other outputs are driven only at each process.
Such control as stated above, in which each load is driven sequentially step by step is called "Sequential control" or "Process stepping type control".
In such control procedure, the program design is very difficult in the relay simbol program.

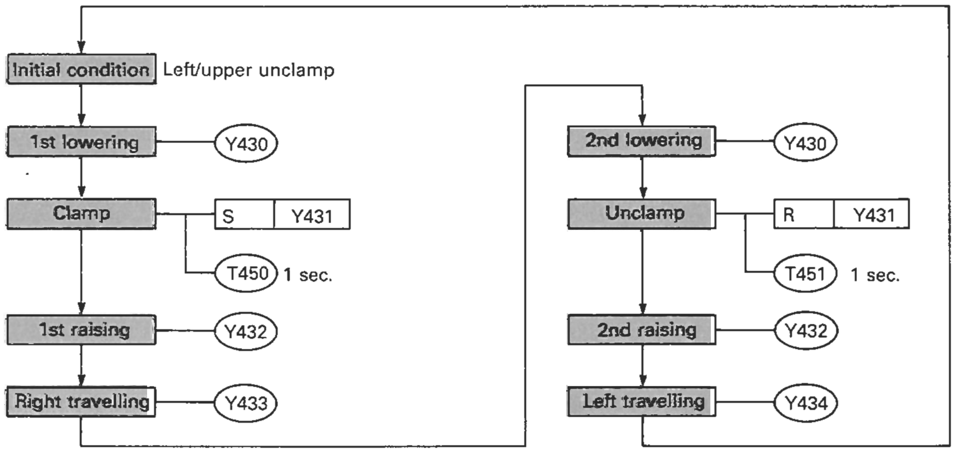


Fig. 4-4 Load drive chart

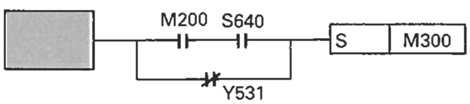
NOTE

[1] Multiple use of timer

The same timer may be used except in the adjacent process.
In addition, each timer may be set to different value respectively.

[2] Handling of output circuit

It is also possible to connect the series contact to the output circuit of each process, and further the contact circuits such as parallel contact, etc. to the output circuit of each process.



(2) Transfer condition chart

Fig. 4-5 shows on what conditions each process should be transferred.
When the start button is depressed on the initial condition, the process is transferred to the 1st lowering process.
When the arm reaches the lower limit position as the lowering solenoid valve operates, the lower limit switch X401 is turned on, and the process is transfeured to the clamp process.
Since the timer T450 is executed together with the clamp output, the process is transferred to the succeeding 1st raising process when the contact of the timer is turned on thereafter.
The operation is performed similarly thereafter to complete a series of process transfer.

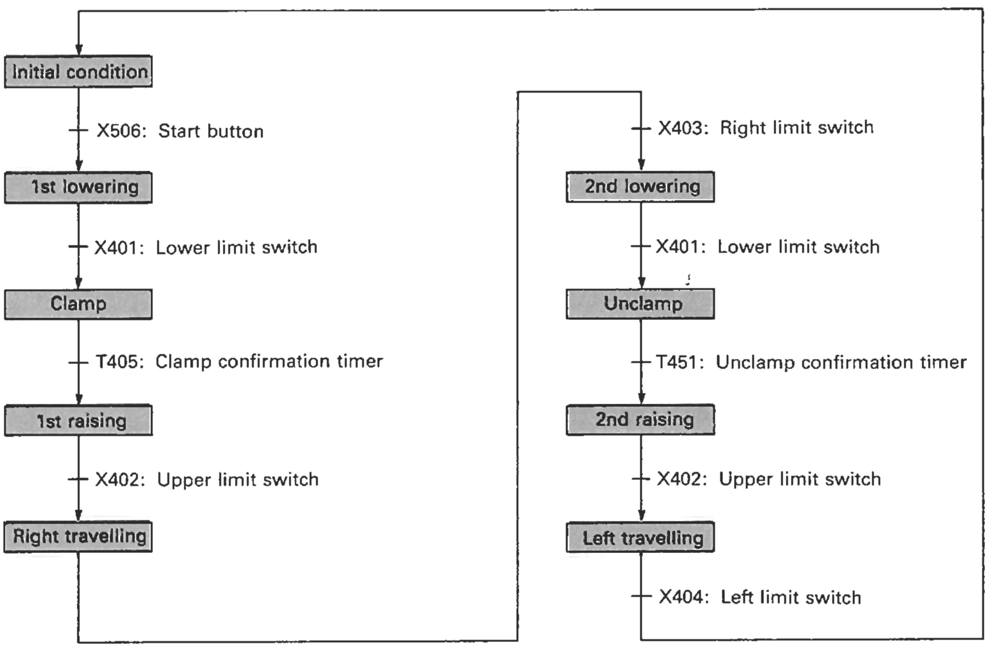
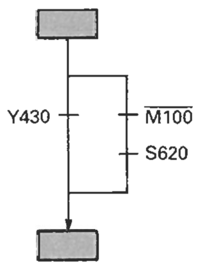


Fig. 4-5 Automatic operation flowchart

NOTE

Handling of transfer circuit

When normally-close contact is used for transfer condition, write it as $\overline{\text{X400}}$, for instance.
It is also possible to use series or parallel circuit of various contacts as transfer condition.



Transfer is executed when Y430 is turned on, or $\overline{\text{M100}}$ is turned on (M100: OFF) and S620 is turned on.

(3) State transition diagram

Fig. 4-6 shows a state transition diagram combining the load drive chart (Fig. 4-4) with the transfer condition chart (Fig. 4-5), each process of which is alloted with state No.
The state No. may be used in optional No. ranging from "S600" to "S647", but it is not necessary that the No. is serial as shown in the following diagram.
As described above, it is possible to program simply by preparing a state transition diagram for the machine operation specifications without the necessity to design the conventional relay sequence.

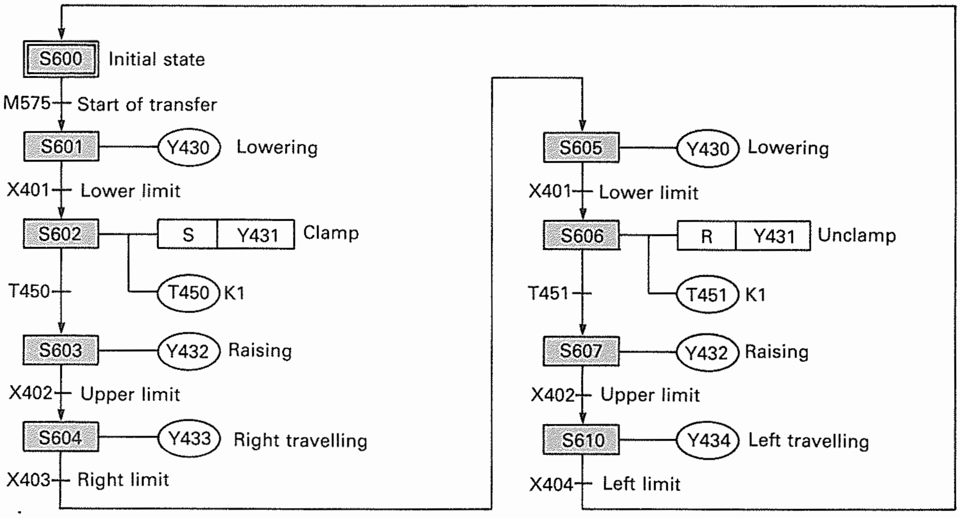


Fig. 4-6 State transition diagram

NOTE

[1] Initial state

The initial state indicating the initialized condition is shown with double frame, and set with the home position returning instruction as shown in Fig. 4-10.

[2] Transfer start

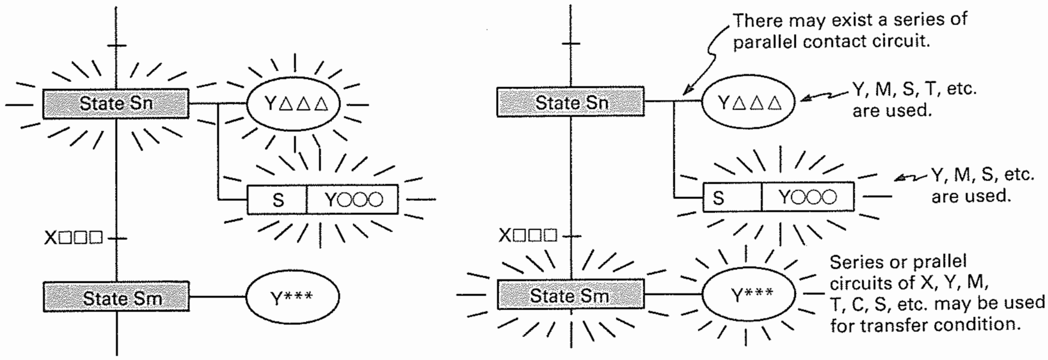
The special auxiliary relay M575 used for transfer start is so designed as to be turned on when the start pushbutton is depressed as shown in Fig. 4-11. It is recommended to connect the home position condition in series.

[3] Program example

The above program will be given as follows;

STL	S600	K	1	S	S605	S	S607
LD	M575	LD	T450	STL	S605	STL	S607
S	S601	S	S603	OUT	Y430	OUT	Y432
STL	S601	STL	S603	LD	X401	LD	X402
OUT	Y430	OUT	Y432	S	S606	S	S610
LD	X401	LD	X402	STL	S606	STL	S610
S	S602	S	S604	R	Y431	OUT	Y434
STL	S602	STL	S604	OUT	T451	LD	X404
S	Y431	OUT	Y433	K	1	S	S600
OUT	T450	LD	X403	LD	T451		

(4) Function of state

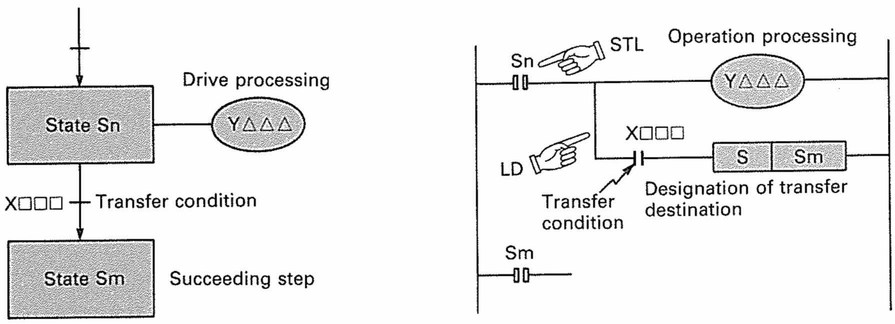


(a) Waiting for X□□□ ON operation

(b) Operation after X□□□ ON operation

Fig. 4-7 Function of state

When the state (Sn) is turned on, the output YΔΔΔ and Y○○○ are turned on (Fig. -a).
If the transfer condition X□□□ is turned on even momentarily, the state (Sm) is turned on and Y*** is turned on at the same time. (Fig. -b)
At the same time, Sn become inoperative and the output YΔΔΔ is turned off.
In this case, however, the output Y○○○ driven by the set instruction holds its operation.
Both states are turned on during the momentary period (one-execution cycle) when transferred from the state (Sn) to state (Sm).



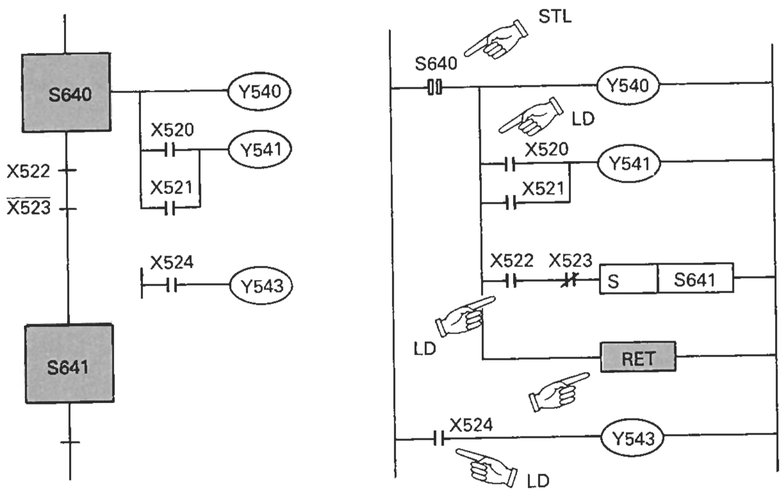
(a) State transition diagram

(b) Step ladder diagram

Fig. 4-8 State and STL instruction

The above figure (a) represents one example of state transition diagrams.
Each state is provided with three functions of "Drive processing for each load", "Designation of transfer destination" and "Designation of transfer condition".
This represents the transition diagram by the step ladder diagram (Fig. -b) in relay sequence format, in which the instruction of STL is used for contact (—| |—).
LD (LDI) instruction is programmed for the initial contact leading to the STL contact.
If the state (Sm) is set through STL contact (Sn), Sn is reset automatically.
STL instruction is provided with the function to reset the origin state automatically.

(5) Program of STL circuit



(a) State transition diagram

(b) Step ladder diagram

```
STL   S640
OUT   Y540
LD    X520
OR    X521
OUT   Y541
LD    X522
ANI   X523
S     S641
RET
LD    X524
OUT   Y543
```

(c) Program

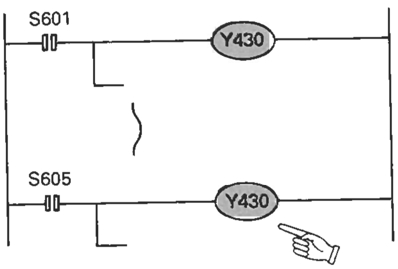
Fig. 4-9 Program of STL circuit

Fig. 4-9 shows how to program the PC from the state transition diagram or the step ladder diagram.
As is apparent from the figure, it is possible to directly drive the coil through the STL contact, or through the other contacts.
The STL contact is basically connected to the bus bar except in case of parallel branch/joining to be described further on.
Since LD point is shifted to rightward when this instruction is used, use RET instruction when necessary to return the point to the original bus bar.
Be sure to program RET instruction at the end of series of STL circuits.

NOTE

[1] Handling of double output

The circuit block to be driven by STL circuit is executed only when the STL contact is turned on and one execution cycle after it has been turned from on to off. The output in the block is turned on/off accordingly. While STL contact is opened, no operation is executed same as (jump condition) and no output processing is executed, either.



Double output applicable

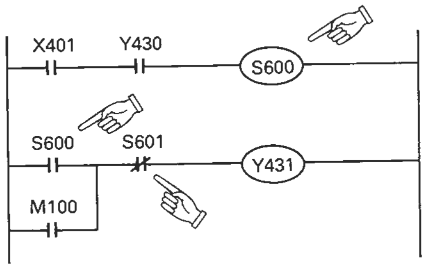
Even if double output Y430 is programmed as shown in the left figure, the output Y430 is executed as ON if either contact of S601 or S605 is turned on.

In addition, if a counter is programmed after STL contact, the counter can be reset only while STL contact is closed.

continued ...

[2] Handling of state

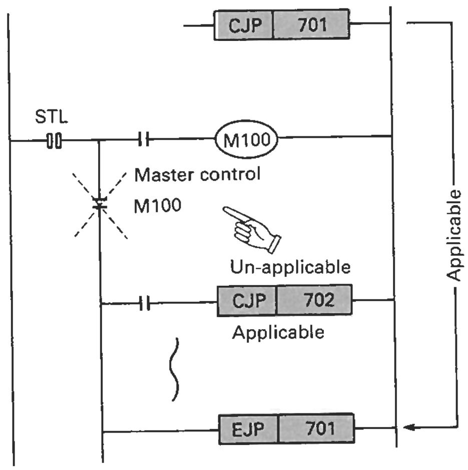
STL instruction is effective only to the state-S, and the number of STL instruction applicable to the same state is limited only to one time. (excluding "Parallel/joining" to be explained further on).
It is possible to apply the instructions such as LD, LDI, AND, ANI, OR, ORI, OUT, S, etc. to the state-S in the same manner as ordinary auxiliary relays.
Only S instruction and R instruction are effective for the output instruction for state after STL contact.



LD	X401
AND	Y430
OUT	S600
LD	S600
OR	M100
ANI	S601
OUT	Y431

[3] State and MC/CJP instructions

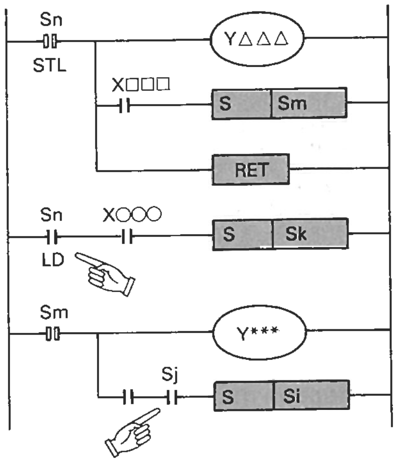
MC instruction and MCR instruction cannot be used after STL contact.
CJP instruction and EJP instruction may be used.
When jumped from CJP701 to EJP701 in the figure below, it is regarded as ON and the PC execute the succeeding circuit even if STL contact of the circuit block is turned off.



continued ...

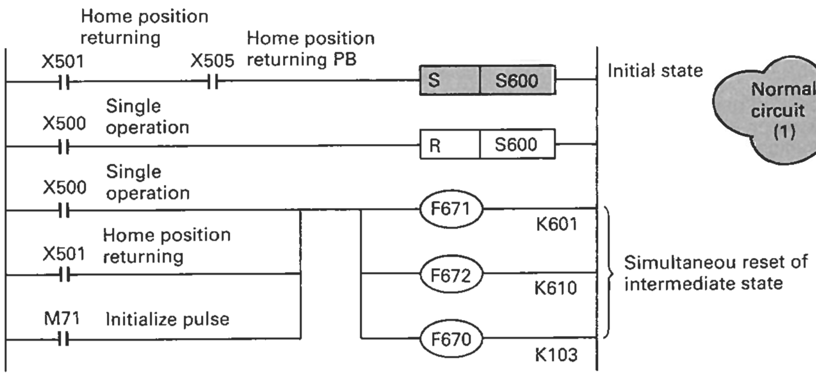
[4] Un-reset transfer procedure for transfer origin

It is also possible to transfer to the other state without resetting the state automatically.
If the transfer condition X□□□ is turned on while the state (Sn) is turned on, the state (Sm) is turned on, by which the state (Sn) is reset automatically.
If the transfer condition X○○○ is turned on in advance while the state (Sn) is operating, however, the state (Sk) is turned on, but the state (Sn) will not be reset.
It is possible to use the contact (Sj) of other state for the transfer condition from the state (Sm) to state (Si)



4-3 General sequence of mode selection, etc.

(1) State initialization



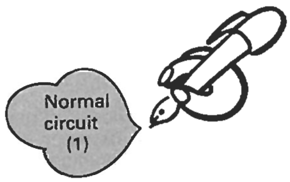
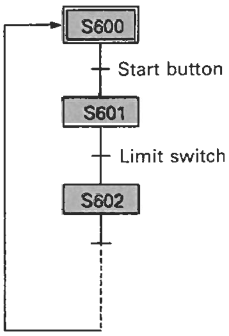


Fig. 4-10 State initialization

(1) Setting of initial state
The initial state (S600 for Fig. 4-10) showing the machine initialized condition is set when the home position returning button is depressed under home position returning mode, and reset under single operation mode.

NOTE

The initial state is provided with the following roles;



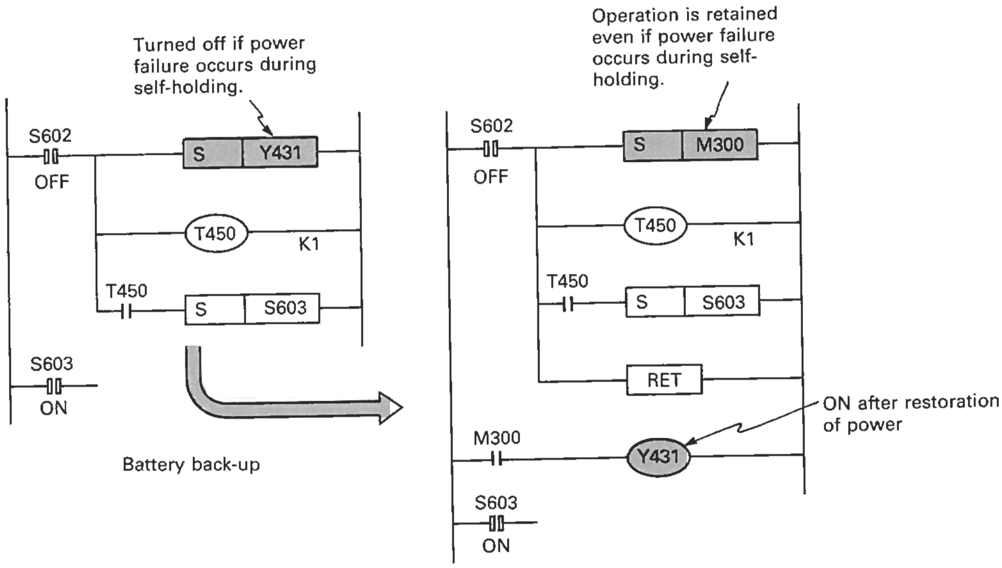
When the start button is depressed in the left-hand side figure, the operating state is transferred from S600 to S601 and the transfer is progressed sequentially thereafter as the machine operation progresses. When the final process operation is completed later, S600 is set again.

Even when the start button is falsely depressed during a series of operations, another start-up is not possible, as S600 has become inoperative.

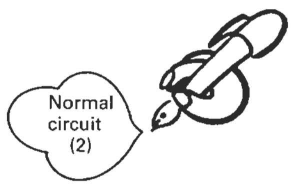
(2) Resetting of intermediate state
The state at the intermediate process needs to be normally reset at manual operation (single operation, home position returning) because the state is backed up by the battery to hold the pre-power failure condition in some cases.
Use the function instruction F670, K103 as shown in Fig. 4-10 to reset the intermediate state.
In this example, the states S601~S610 are reset at the same time.

NOTE

When it is necessary to start the operation continuously from the pre-power failure condition when the power is restored, the contact M71 programmed in Fig. 4-10 is not necessary.
In this case, the output relay driven by the set instruction needs to be driven through the auxiliary relays M300 ~ M377 (battery back-up).



(2) State transfer start



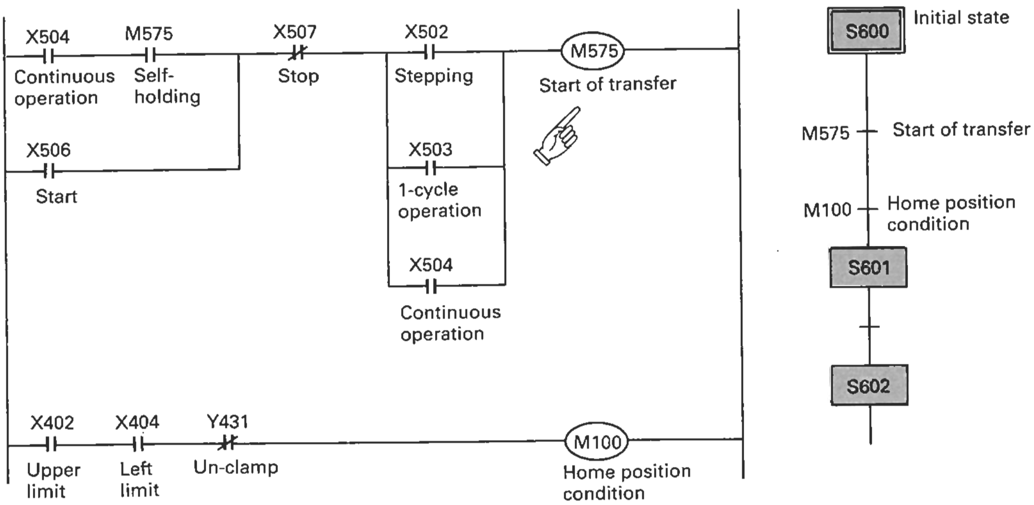


Fig. 4-11 State transfer start-up sequence

The auxiliary relay M575 operates by depressing the start pushbutton during automatic operation (stepping, one-cycle, continuous operation). Especially while the automatic program is being executed, the self-holding circuit operates, and the auxiliary relay M575 holds the operation until the stop button is depressed.
On the other hand, the auxiliary relay M100 is operated to check the machine home position.
When both M575 and M100 are turned on, the transfer is started from the initial state.

(3) State transfer inhibit
When the special auxiliary relay M574 is actuated while the state transfer control is executed by the use of step ladder instruction, the state automatic transfer is inhibitted.

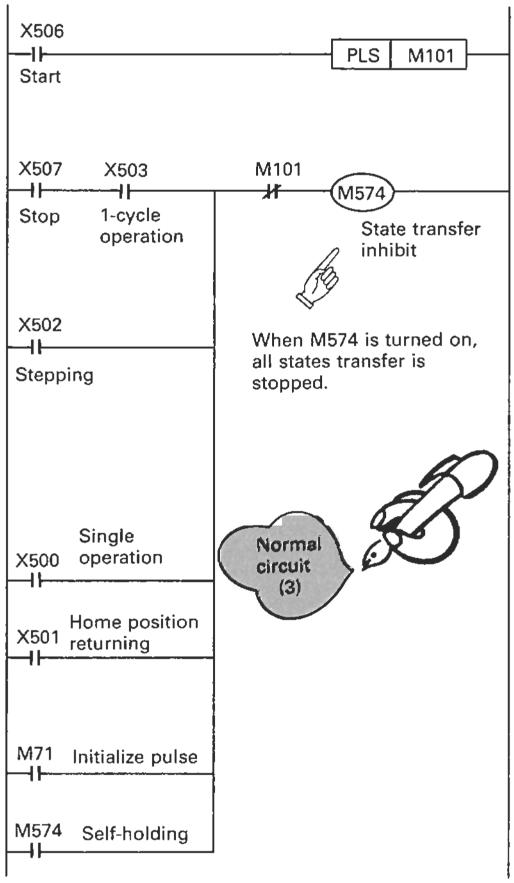


Fig. 4-12 State transfer prohibition sequence

When the start button is depressed, M101 generates the pulse output, causing M574 to turn off.

When the stop button is depressed during one-cycle operation, M574 self-holds, the operation is stopped at current process.

When the start button is depressed, the operation is restarted from the process.

M574 operates all the time under stepping mode, in which the state transfer is inhibitted.

When the start button is depressed, state transfer prohibition is reset momentarily, and proceeded to the succeeding process.

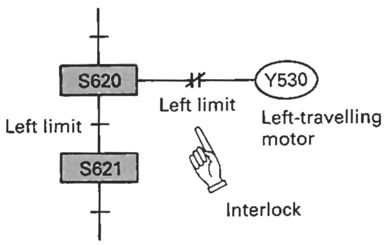
The state transfer is inhibitted under manual mode (single operation, home position returning).
The state transfer inhibit is reset at the time when the start button is depressed after the manual mode reset.

M574 self-holds by the initialize pulse M71 when the programmable controller starts, by which the state transfer will be inhibitted until the start button is depressed.

NOTE

Output interlock

The current operation state during state transfer inhibit is kept turned on.



When the leftward-travelling motor drive output Y530 is operated by S620, for instance, the state is transferred normally at the left limit and Y530 is turned off.
When the state transfer is inhibitted, however, Y530 is kept turned on.
To prevent this, drive circuit Y530 needs to shut off by left limit switch.

4-4 Manual operation sequence

Since any complicated sequential control will not be required under the manual operation mode, design the circuit by the conventional relay sequence method.

(1)Single operation program

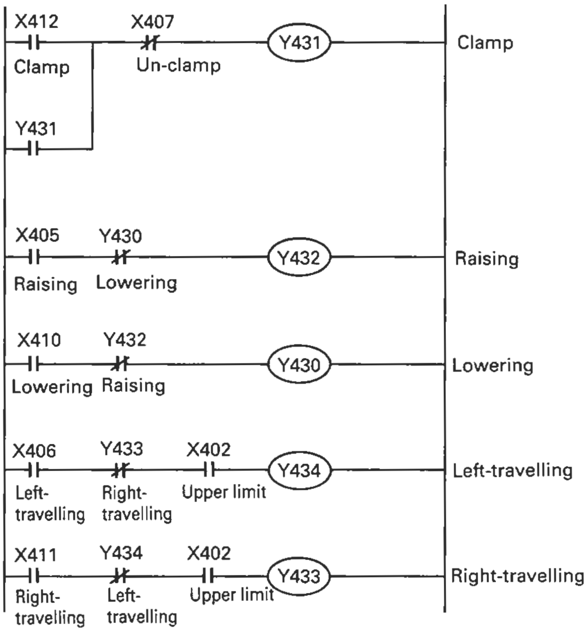


Fig. 4-13 Independent operation program

When the clamp button is depressed, the clamping output Y43l self-holds, which is reset by depressing the unclamp button.

The raising output Y432 remains turned on while the raising pushbutton is depressed.

The lowering output Y430 remains turned on while the lowering pushbutton is depressed.

The left-travelling output Y434 remains turned on while the left-travelling pushbutton is depressed at upper limit position.

The right-travelling output Y433 remains turned on while the right-travelling pushbutton is depressed at upper limit position.

(2) Home position returning program

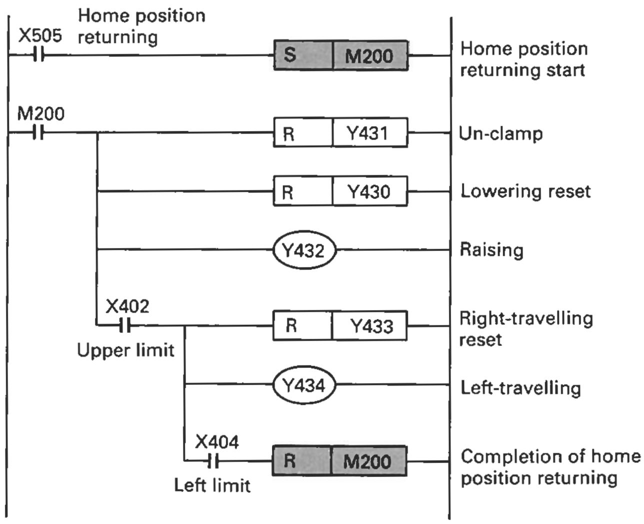
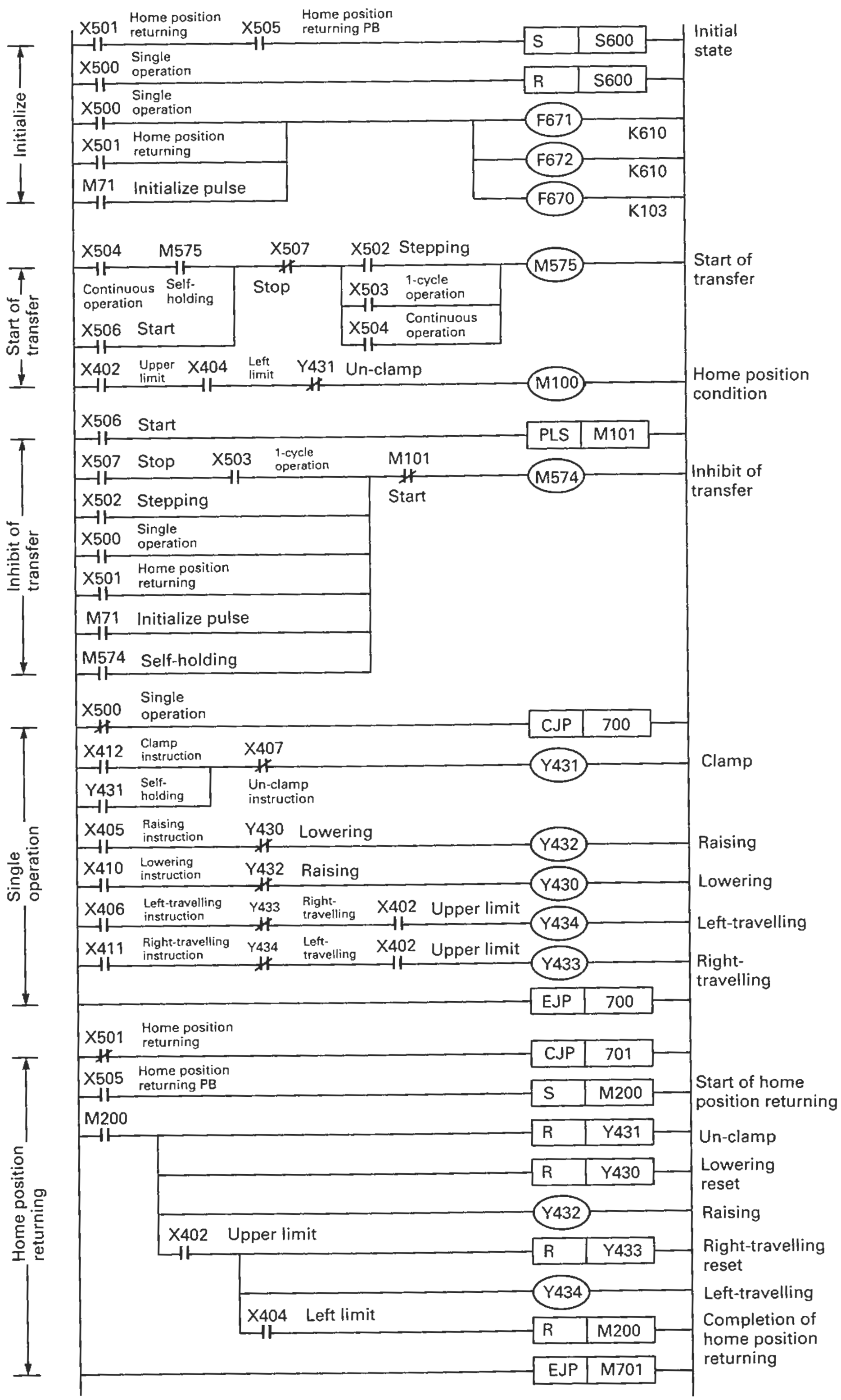


Fig. 4-14 Home position returning program

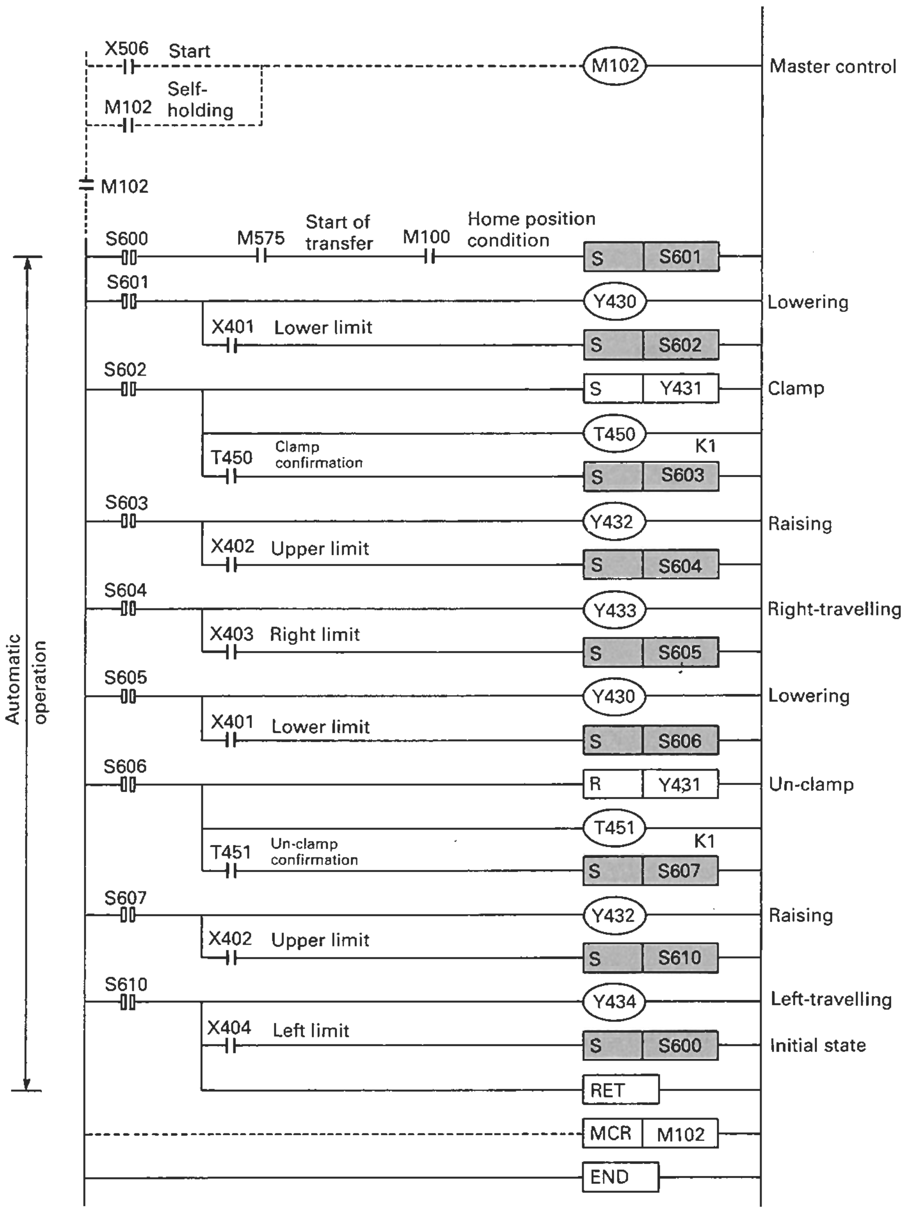
Under the home position returning program, reset the non-operation loads for caution's sake, while actuating each load in order of safety operation.

Overall program




```
 0 LD   X501
 1 AND  X505
 2 S    S600
 3 LD   X500
 4 R    S600
 5 LD   X500
 6 OR   X501
 7 OR   M 71
 8 OUT  F671
 9   K   601
10 OUT  F672
11   K   610
12 OUT  F670
13   K   103
14 LD   X504
15 AND  M575
16 OR   X506
17 ANI  X507
18 LD   X502
19 OR   X503
20 OR   X504
21 ANB
22 OUT  M575
23 LD   X402
24 AND  X404
25 ANI  Y431
26 OUT  M100
27 LD   X506
28 PLS  M101
29 LD   X507
30 AND  X503
31 OR   X502
32 OR   X500
33 OR   X501
34 OR   M 71
35 OR   M574
36 ANI  M101
37 OUT  M574
38 LDI  X500
39 CJP   700
40 LD   X412
41 OR   Y431
42 ANI  X407
43 OUT  Y431
44 LD   X405
45 ANI  Y430
46 OUT  Y432
47 LD   X410
48 ANI  Y432
49 OUT  Y430
50 LD   X406
51 ANI  Y433
52 AND  X402
53 OUT  Y434
54 LD   X411
55 ANI  Y434
56 AND  X402
57 OUT  Y433
58 EJP   700
59 LDI  X501
60 CJP   701
61 LD   X505
62 S    M200
63 LD   M200
64 R    Y431
65 R    Y430
66 OUT  Y432
67 AND  X402
68 R    Y433
69 OUT  Y434
70 AND  X404
71 R    M200
72 EJP   701
```

Fig. 4-15 General program and manual program



73	LD	X506
74	OR	M102
75	OUT	M102
76	MC	M102
77	STL	S600
78	LD	M575
79	AND	M100
80	S	S601
81	STL	S601
82	OUT	Y430
83	LD	X401
84	S	S602
85	STL	S602
86	S	Y431
87	OUT	T450
88	K	1
89	LD	T450
90	S	S603
91	STL	S603
92	OUT	Y432
93	LD	X402
94	S	S604
95	STL	S604
96	OUT	Y433
97	LD	X403
98	S	S605
99	STL	S605
100	OUT	Y430
101	LD	X401
102	S	S606
103	STL	S606
104	R	Y431
105	OUT	T451
106	K	1
107	LD	T451
108	S	S607
109	STL	S607
110	OUT	Y432
111	LD	X402
112	S	S610
113	STL	S610
114	OUT	Y434
115	LD	X404
116	S	S600
117	RET	
118	MCR	M102
119	END	

Fig. 4-16 Automatic program (for step ladder diagram)

NOTE

Re-starting after power failure

When the operation is re-started from the home position after the restoration of power, the steps 73 ~ 76 and 118 will not be necessary.

When the operation is re-started from the current condition, on the contrary, these program will become necessary to inhibit the automatic output until the start button is depressed.

In this case, however, it is necessary to delete OR·M71 of the 7th step, as the intermediate state will not be reset.

4-5	Handling of multiple flows

(1) Multiple flow configuration

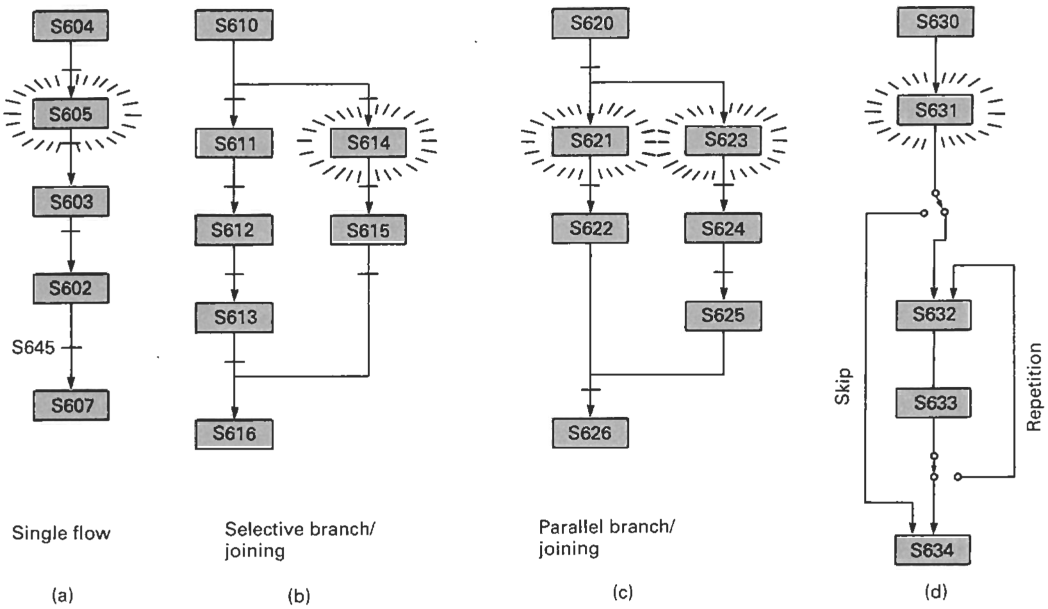


Fig. 4-17 Multiple flow configuration

The micro programmable controller F_1 series will allow various flows operating independently by one PC.

(a) Single flow
It is not necessary to number the state Nos. in order of process.
(The same applies to the others).
In addition, it is possible to interlock the flow transfer conditions by the state of other flow to execute the inter-related control.

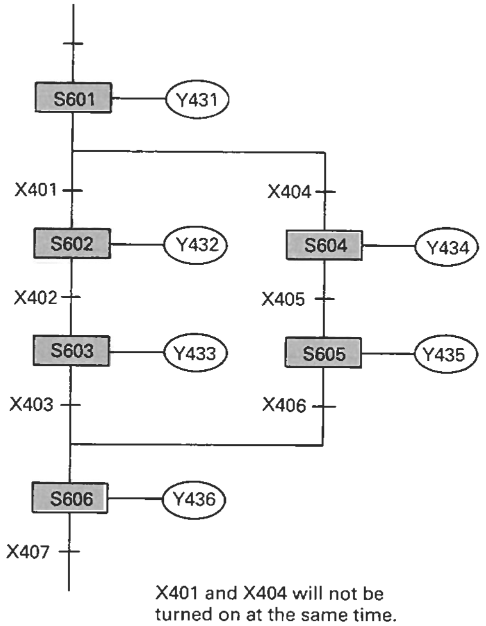
(b) Selective branch/joining
The flow is used for selective branch of multiple flows, of which program example is as shown in Fig. 4-18.

(c) Parallel branch/joining
The flow is used for simultaneous branch of multiple flows, of which program example is as shown in Fig. 4-19.

(d) Skip/repetition
The flow is used for skip/repetition of part of flow, of which program example is as shown in Fig. 4-20.

Any of the complicated flows can be expressed by the use of the above (a) through (d) in combination.

(2) Selective branch/joining



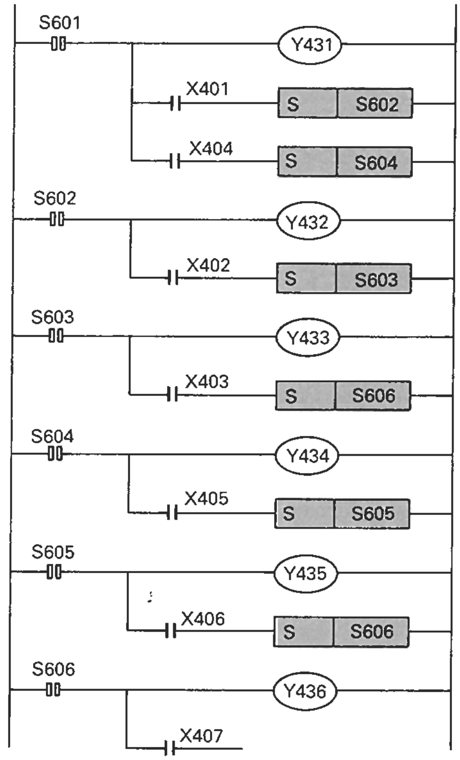


Fig. 4-18 Selective branch/joining

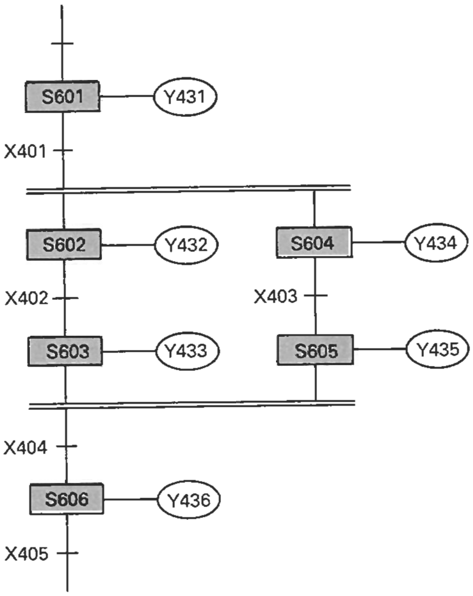
Fig. 4-18 shows a flow diagram and a ladder diagram for selective branch/Joining, in which either one of the multiple flows is executed selectively.
In this case, care needs to be taken not to allow the multiple flows to be transferred at the same time.
The state S601 is reset automatically when either the state S602 or state S604 is set.
The state S606 is set by the state S603 or S605, and the transfer origin (S603 or S605) will be set automatically.

STL	S601	STL	S602	LD	X403	STL	S605
OUT	Y431	OUT	Y432	S	S606	OUT	Y435
LD	X401	LD	X402	STL	S604	LD	X406
S	S602	S	S603	OUT	Y434	S	S606
LD	X404	STL	S603	LD	X405	STL	S606
S	S604	OUT	Y433	S	S605	OUT	Y436

NOTE

Programming procedure
Program all the outputs from one STL contact with the contact centered.
Start programming of all STL contacts in order of No. sequentially.
In this case, the actual sequence may be in any order.

(3) Parallel branch/joining



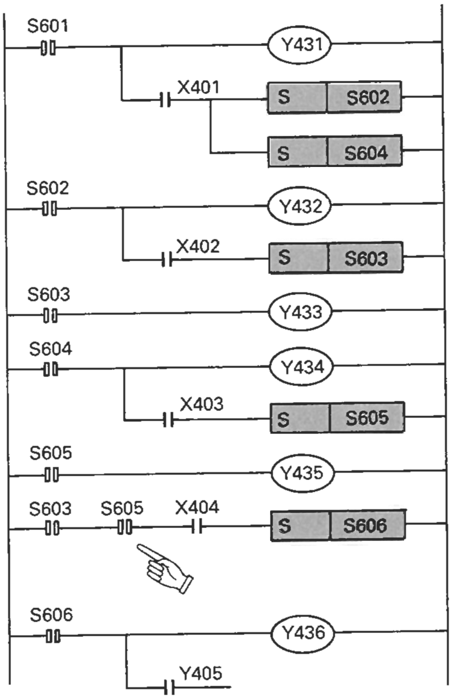


Fig. 4-19 Parallel branch/joining

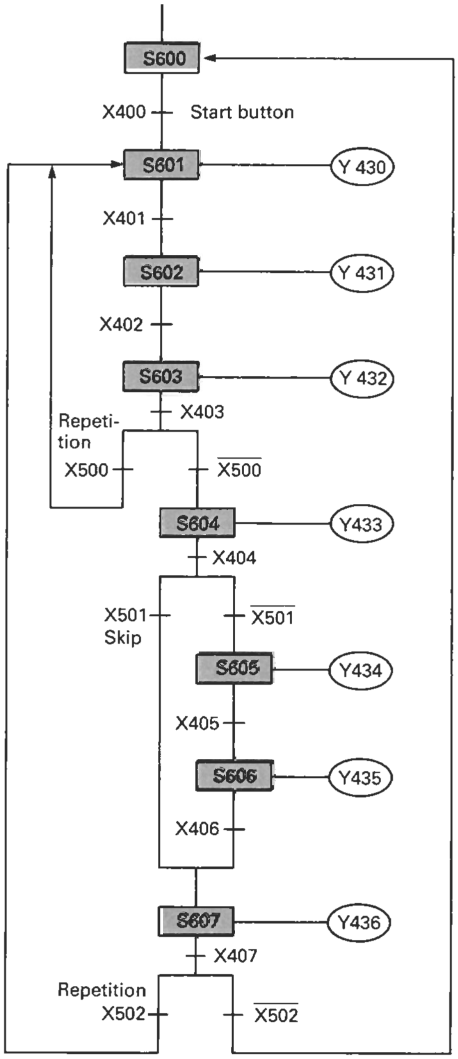
Fig. 4-19 shows an example in which the multiple flows are branched simultaneously and then joined. The state S606 is set after the states S603 and S605 have been both operated to reset the transfer origins S603 and S605.
In case STL instruction is used continuously (limited to less than 8 times), it means the series connection.
Only the SET instruction for state is effective to the coil following the series connection.
RET instruction returns to bus bar for one use.

STL	S601	LD	X402	STL	S605	
OUT	Y431	S	S603	OUT	Y435	
LD	X401	STL	S603	STL	S603	Continuous use
S	S602	OUT	Y433	STL	S605	(for parallel confluence)
S	S604	STL	S604	LD	X404	
STL	S602	OUT	Y434	S	S606	
OUT	Y432	LD	X403	STL	S606	
		S	S605	OUT	Y436	

NOTE

Programming procedure
Program the S (set) instruction for joining side states all together.
Accordingly, STL603 and STL605 are programmed two times in the case of above example.

(4) Skip/repetition



STL	S600	AND	X501
LD	X400	S	S607
S	S601	LD	X404
STL	S601	ANI	X501
OUT	Y430	S	S605
LD	X401	STL	S605
S	S602	OUT	Y434
STL	S602	LD	X405
OUT	Y431	S	S606
LD	X402	STL	S606
S	S603	OUT	Y435
STL	S603	LD	X406
OUT	Y432	S	S607
LD	X403	STL	S607
AND	X500	OUT	Y436
S	S601	LD	X407
LD	X403	AND	X502
ANI	X500	S	S601
S	S604	LD	X407
STL	S604	ANI	X502
OUT	Y433	S	S600
LD	X404	RET	

Fig. 4-20 Skip/repetition

While X500 is turned on, S601 ~ S603 are operated repeatedly.
When X500 is turned off ($\overline{X500}$ = ON) and X501 is turned on, however, S605 and S606 are skipped, and not operated.

NOTE

Jump and skip
"Skip" to be described here is different from jump instruction.
The state is skipped, depending upon the transfer condition of state.

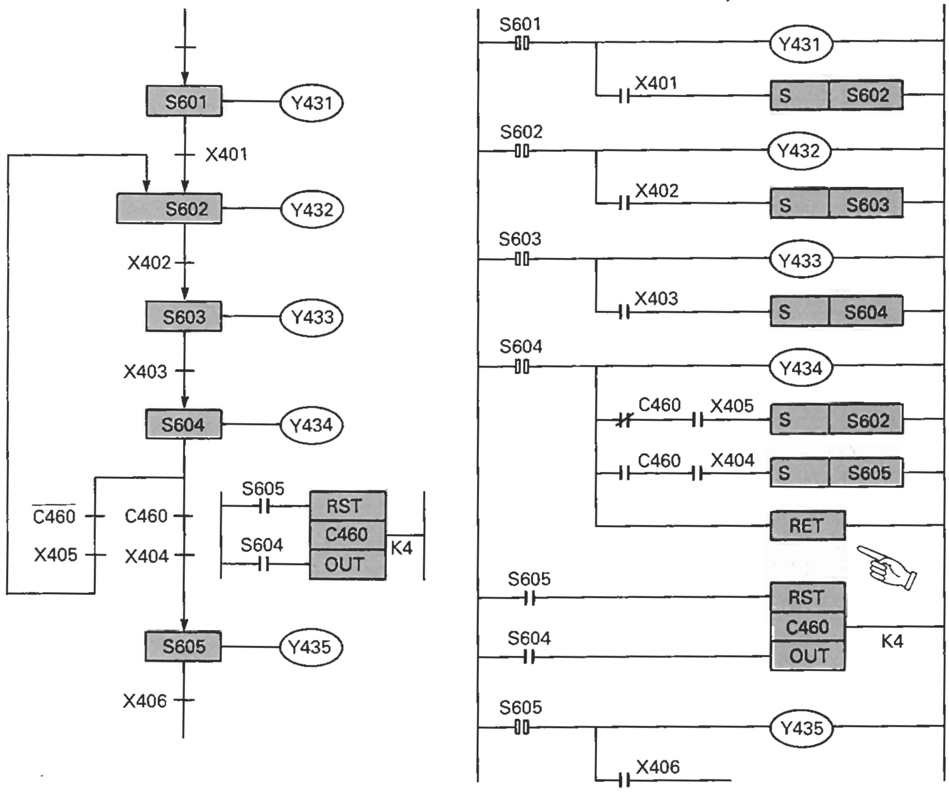


Fig. 4-21 Repetition operation by counter

In this example, the portion of states S602 ~ S604 is operated for five times repeatedly, depending upon the setting value of counter C460, and then transferred to the state S605.

STL	S601	STL	S603	LD	C460	STL	S605
OUT	Y431	OUT	Y433	AND	X404	OUT	Y435
LD	X401	LD	X403	S	S605		
S	S602	S	S604	RET			
STL	S602	STL	S604	LD	S605		
OUT	Y432	OUT	Y434	RST	C460		
LD	X402	LDI	C460	LD	S604		
S	S603	AND	X405	OUT	C460		
		S	S602	K	4		

NOTE

Handling of counter circuit
Bear in mind that RET instruction is used before the counter circuit program. Unless RET instruction is inserted, the setting value cannot be fetched, as the count input S604 is turned off, resulting in jump operation mode when the counter is reset (S605: ON).

(5) Program example

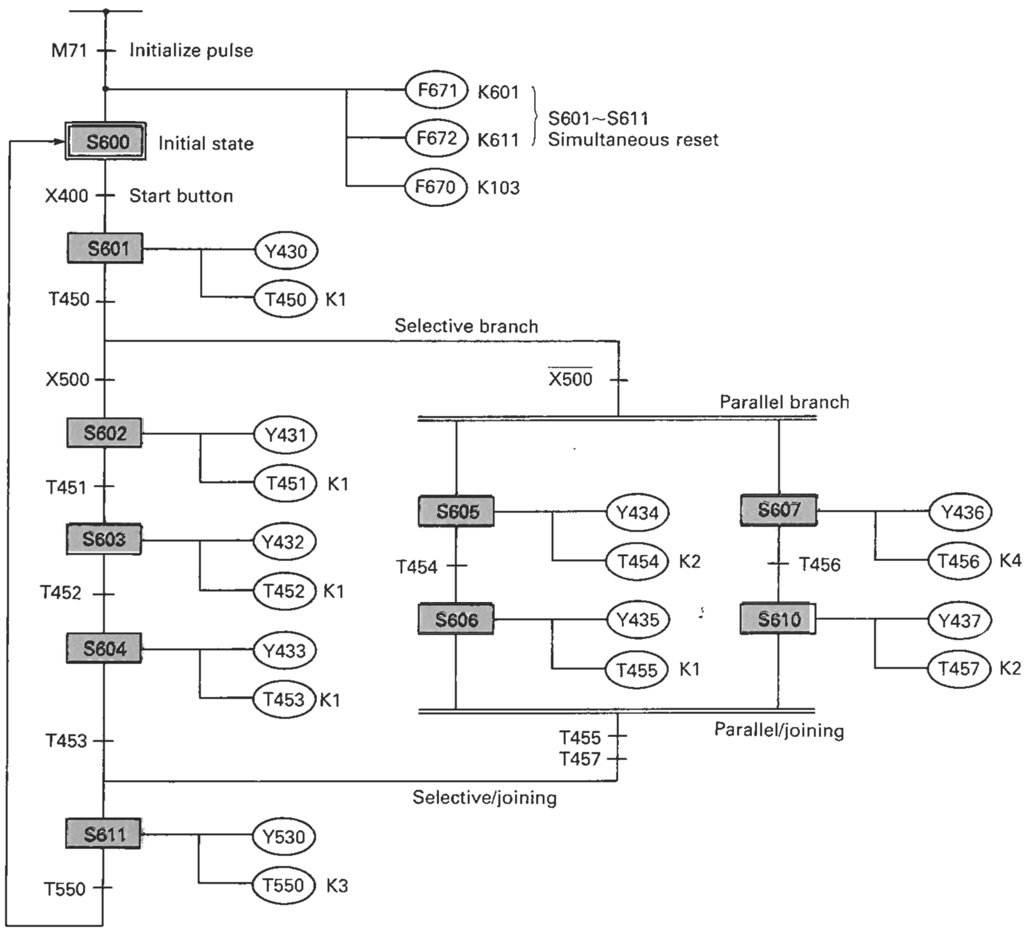
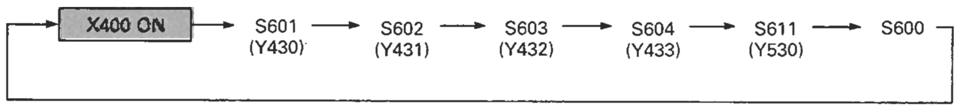
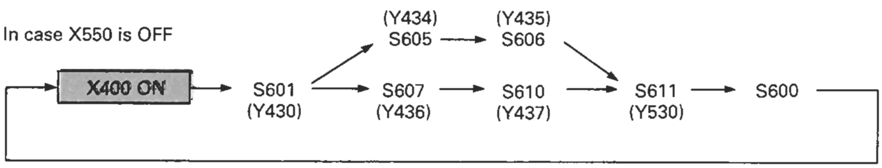


Fig. 4-22 Compound branch/joining

In case X500 is ON



In case X550 is OFF



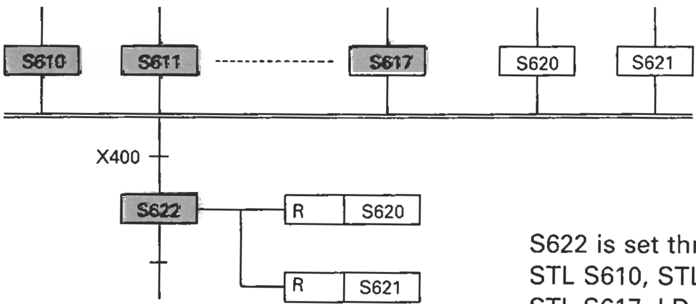
```
LD   M 71
OUT  F671
  K   601
OUT  F672
  K   611
OUT  F670
  K   103
S    S600
STL  S600
LD   X400
S    S601
STL  S601
OUT  Y430
OUT  T450
  K     1
LD   T450
AND  X500
S    S602
LD   T450
ANI  X500
S    S605
S    S607
STL  S602
OUT  Y431
OUT  T451
  K     1
LD   T451
S    S603
STL  S603
OUT  Y432
OUT  T452
  K     1
LD   T452
S    S604
STL  S604
OUT  Y433
OUT  T453
  K     1
LD   T453
S    S611
STL  S605
OUT  Y434
OUT  T454
  K     2
LD   T454
S    S606
STL  S606
OUT  Y435
OUT  T455
  K     1
STL  S607
OUT  Y436
OUT  T456
  K     4
LD   T456
S    S610
STL  S610
OUT  Y437
OUT  T457
  K     2
STL  S606
STL  S610
LD   T455
AND  T457
S    S611
STL  S611
OUT  Y530
OUT  T550
  K     3
LD   T550
S    S600
RET
```

NOTE

[1] Limit on number of branch/joining

There is no limit in the number of selective branch.
(Actually, the number of state is limited).
In the parallel branch, however, there are the following limitations at the joining point;

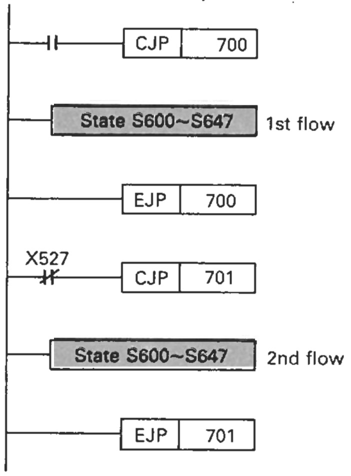
When the joining state S622 is operated in the following figure, it is possible to reset automatically up to 8 points of transfer origin, however, the remaining states require the program resetting.



S622 is set through
STL S610, STL S611 ...
STL S617, LD X400 AND S620, AND S621.

[2] Overlapped use of state

It is possible to use the same state No. by discriminating the multiple flows which will not operate simultaneously with jump instructions.

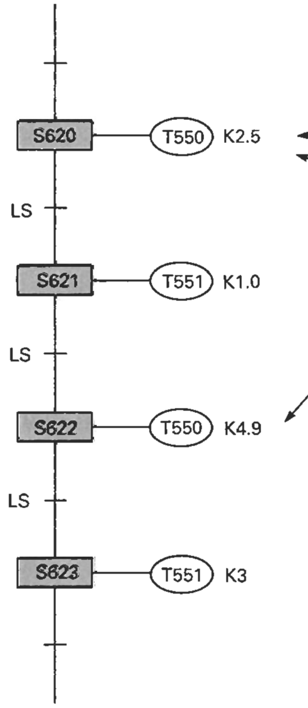


In the flows divided by jump instruction as shown at left figure not to operate simultaneously the overlapped state No. can be used.

The state having been used before jump needs to be reset completely.

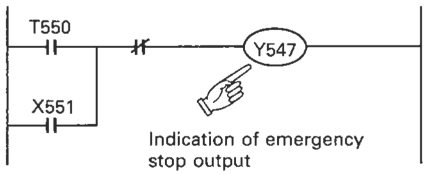
[3] Time-out sequence

The approach for generating emergency stop output for case when the operation time of each state has exceeded the normal value is as described in the following;



Set the operation time at a little longer than the normal transfer time required.

It is possible to use the timer of the same No. for non-adjacent state.



5 FUNCTIONAL INSTRUCTIONS

The use of basic sequential instructions and step ladder function described in the previous chapters so far will make it possible to prepare almost all the ordinary application programs for relay replacement.
The F_1 series programmable controller is provided with additional 90 kinds of functional instructions to allow the preparation of special programs such as high-speed processing and data transfer, special application of counter, arithmetics, analog data handlings, etc., thus allowing it to be applied in increased application fields.
This chapter describes these functional instructions.

(1) Expression format

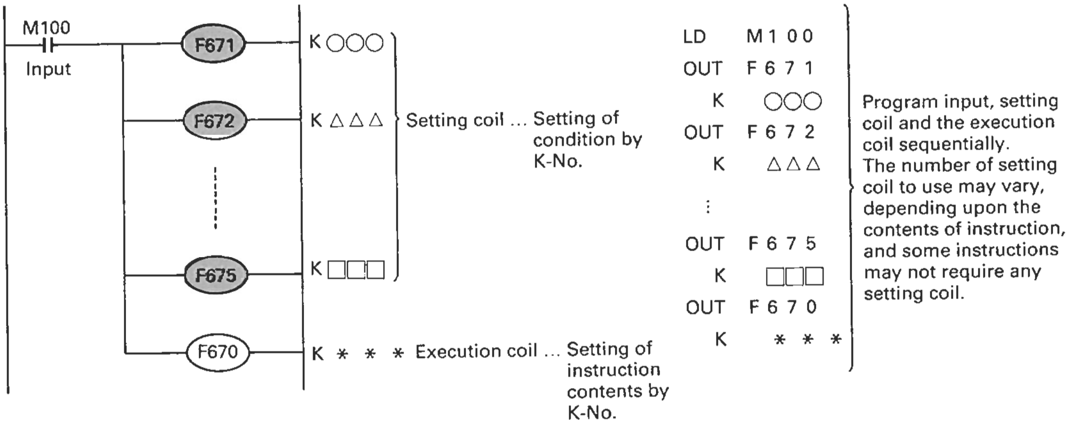


Fig. 5-1 Basic configuration of functional instruction

The expression format of the functional instruction is as shown in Fig. 5-1, of which condition setting meaning for the setting coil may be different, depending upon each functional instructions.
Each contact such as X, Y, M, T, C, S, etc. within the programmable controller is used for the drive input, by which it is possible to use not only the single contact but also circuit made up of multiple number of circuits in combination.
The execution time for functional instruction needs to be given by adding the execution time of setting coil to that of execution coil (Refer to Tables 6-1 and 6-2 in summary).
In either case, execution time may vary between when the input is turned on and when it is turned off.

NOTE

[1] Role of setting coil
The execution coil defines the contents of functional instruction by K-No. following the execution coil, whereas the setting coil serves to further designate the detailed condition on the basis of each functional instruction contents, and may provide the different meaning, depending upon the functional instruction used.

[2] In case the functional instructions have been used several times and each setting coil has already been set, the setting coil for later functional instructions are omissible under the condition that the setting conditions are the same.

(2) Handling of data registers

The F_1 programmable controller is provided with 64-point data registers (D700 ~ D777) which are used to store the numeric data.
The applied instruction is used to read/write the data from/to the data register for execution of comparison and arithmetic operations.
If Nos. of K700 ~ K777 are set for the setting coil of applied instruction, the data registers D700 ~ D777 will be designated.
The 64-point data register is made up of BCD 3 digits, all points of which are backed up by the battery.

NOTE

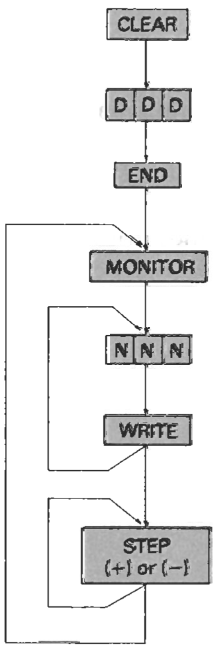
---

(1) When using the auto reload or high-speed output table for applied instruction to be described further on, the data registers of D756 ~ D777 will be occupied.
Accordingly, the data registers to be used for the general purpose will be D700 ~ D755 (46 points).

(2) When all-clearing the data in the data registers including those backed up by the battery by the use of programming panel, all the data will be reset to be "000".

---

(3) Monitoring function for data register (when using F-20P-E, F1-20P-E, F2-20P-E)



Data register
(700 ~ 777)

END key is sued to indicate data register monitoring.

Displays data register contents.

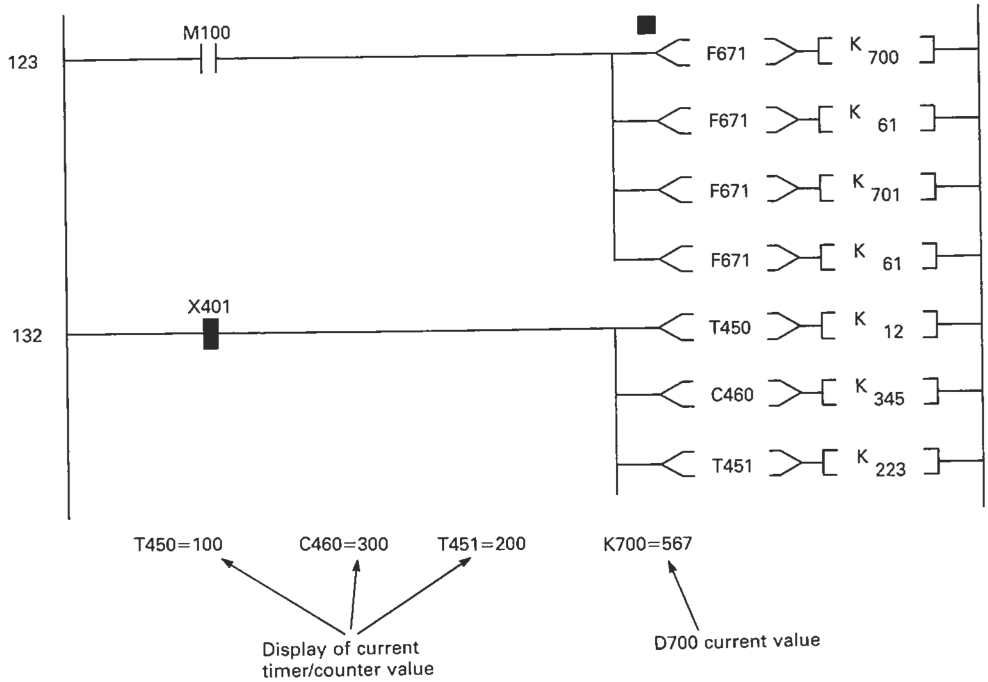
Key in new data to modify register contents.

New data is written in data register.

Data register Nos. before and behind are displayed.

(4) Monitoring function for data register (when using GP80 F2A)

1) Display example   LDR MNT SSN 1 2 3 GO



* In case when the cursor is moved to the left side of the setting coil accompanied K704 ~ K777, the contents in the data register of designated No. will be displayed at the column "Message".
The same procedure can be used to display the contents in the registers at four points.

* In case the timer and counter are used on the same screen, however, they are displayed by the current value.
Generally when four points are exceeded, they overflow to the left of screen sequentially.

* If the screen is changed, the display contents for K700 ~ K777 will be cleared, and the the current value for timer and counter on new screen will be displayed.

2) Display example LST MNT K 7 0 0 GO

K700		K710		K720		K730	
K701		K711		K721		K731	
K702		K712		K722		K732	
K703		K713		K723		K733	
K704		K714		K724		K734	
K705		K715		K725		K735	
K706		K716		K726		K736	
K707		K717		K727		K737	

List monitor

Display of current value

GO : Display is shifted in unit of 8 points.

3) Display example LDR MNT SSN 2 5 0 MNT TST

- Move cursor to point-A.

─□─ K 1 2 3 GO ... Contents in data register is changed to "123".

- Move cursor to point-B.

─□─ K 7 1 0 GO ... No. of data register is changed from D702 to D710.

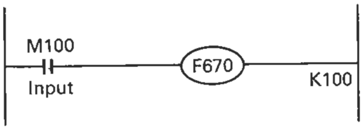
5-1 Input/output high-speed processing instruction

F670 K00 (or K100)

Input all-point refresh

Object elements; X000~X027
X400~X427
X500~X527

This instruction is not provided with any setting coil, and can be established only by the execution coil.

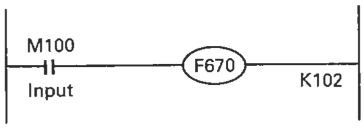


When the input (M100 in above example) is turned on, ON/OFF condition of all input terminals is fetched newly in the input image memory at the time when this instruction is executed.
This function is not effective while the input is turned off. The refresh operation is the same operation as the batch input processing to be executed prior to the execution of the program.
It is possible that the reading to the image memory may remain turned off due to response lagging of input filter even if the input terminal is turned on.

F670 K02 (or K102)

Output all-point refresh

Object elements: Y030~Y047
Y430~Y447
Y530~Y547



This instruction is not provided with any setting coil, and can be established only by the execution coil.

If the input (M100 in the above example) is turned on, ON/OFF condition of output image memory is transferred to the output latch memory at the time when this instruction is executed.
This function is not effective when the input is turned off.
This refresh operation is the same as batch output processing executed after completing the execution of the program and the actual output is turned on upon elepse of response lagging of output relay after the output latch memory is turned on.

Application example-1

High-frequency input processing

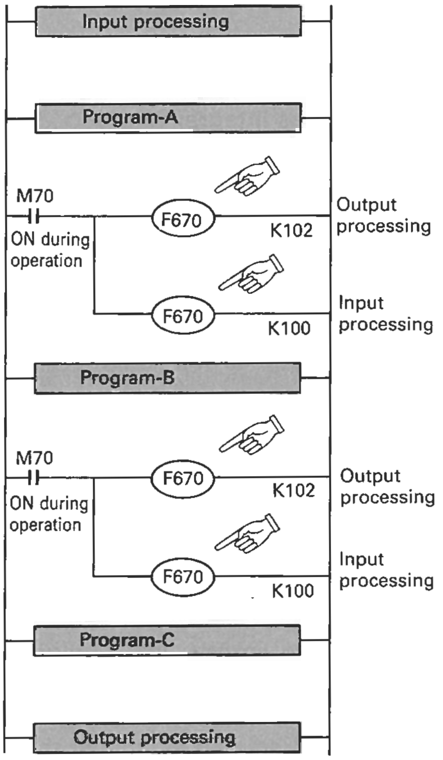
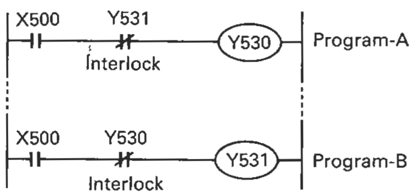


Fig. 5-2 Additional input/output processing

Program the input all-point refresh instruction (input processing) and output all-point refresh instruction (output processing) with the overall program divided into almost equally in time.

With the program, it becomes possible to output the execution result to the output terminal as quickly as possible after execution of the program on the basis of as new input terminal information as possible.

In this case, the same input can become ON or OFF within the one cycle of the program. In the output in which the program is executed under the different block as shown below, it is necessary to provide an interlock.



Unless the interlock is provided, Y530 and Y531 may be turned on at the same time.

NOTE

Input/output response lagging

When the input-all-point refresh instruction or output all-point refresh instruction is programmed "n" times equally in the program of which execution cycle is "T_0", the input/output response lagging is as follows.

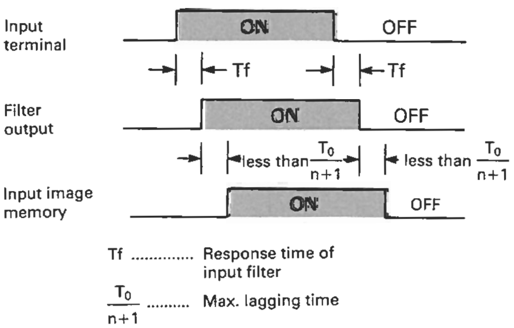


Fig. 5-3 Input response lagging

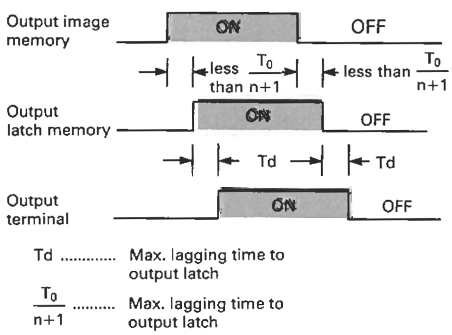


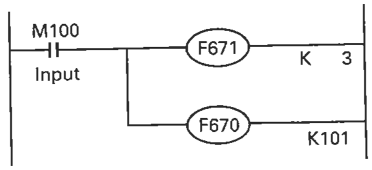
Fig. 5-4 Output response lagging

To shorten the response lagging due to input filter, it is possible to use the following functional instruction F670 K101

F670 K101

Partial input refresh

Object elements: X400~X407

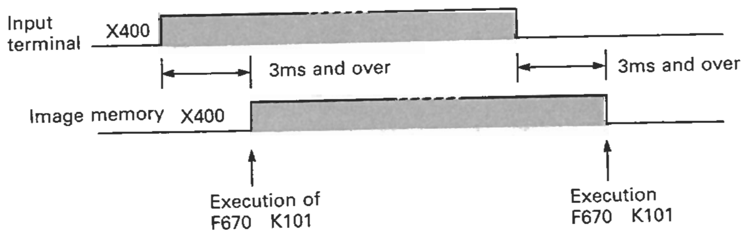


Input filter constant setting
(K = 0~60ms integral value)

Refresh of input X400~X407 input memory

When the input (M100 in the above figure) is turned on, input information for 8 points of X400~X407 will be newly fetched, and then written in the input image memory when this instruction is executed.
The function is not effective while the input is turned off.
The response lagging for ON/OFF of image memory against ON/OFF of the input terminal is determined by the value K of the setting coil F671.
An RC filter is normally provided with the input of the programmable controller to prevent the noise interference, by which the response lagging of approx. 10ms may be brought about.
On the other hand, the inputs X400~X407 are provided with a digital filter which can be modified by the program.
When making consideration to prevent any noise interference in the input wiring, therefore, the constant of the filter may be reduced to any small value in decrement of 1msec.
(Actually, the minimum RC filter is used, and the value will not be reduced to smaller than 200µs).
Observing the operation of X400 in the example given in the above figure, the followings can be known;

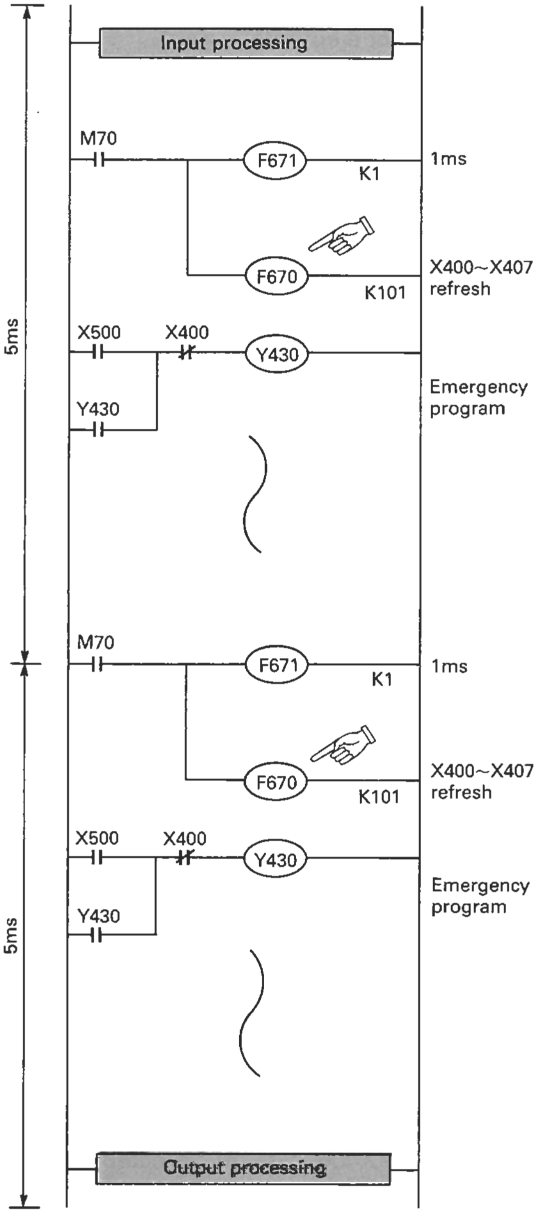
(1) The image memory X400 is turned on (or off) when the input terminal X400 has been turned on (or off) for more than 3msec. before the execution of F670 K101 instruction.
(2) If the input terminal X400 is turned to ON or OFF within 3msec., the image memory remains unchanged.



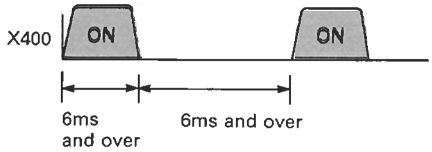
NOTE

The input signal is fetched with its filter constant of 10ms at the time of input processing before execution of the program.

Application example-2
Emergency output program



Note: X400~X407 is treated as filter constant of 10ms in the input processing before exection of program.



Execute the program with the emergency program for two times in a program of 10ms execution cycle dividing into two, shown in the left figure.

Program the partial input refresh instruction before the emergency program with the filter constant designated at "1ms".

If the ON time or OFF time of input X400 is more than 6msec., for instance, the latched output Y430 may be turned off.

In case the partial input refresh instruction is not used and the emergency refresh program is programmed only one time, the ON time or OFF time of input X400 needs to be more than 20ms due to influence by operation cycle of approx. 10ms and input filter constant of 10msec.

The above described the emergency fetching of input X400.
Where the emergency output of output Y430 is needed, use the output all-point refresh instruction F670 K102 at the same time.

F670 K112 K113
X400 rising detection

Object elements: X400

F670 K114 K115
X401 rising detection

Object elements: X401

K112 and K113 are used in pairs to detect the rising of input X400.
K114 and K115 are also used in pairs to detect the rising of input X401.
These two are the same in function except that each object element is different.
In this respect, the following will describe the functions of K112 and K113, referring to the equivalent circuit.

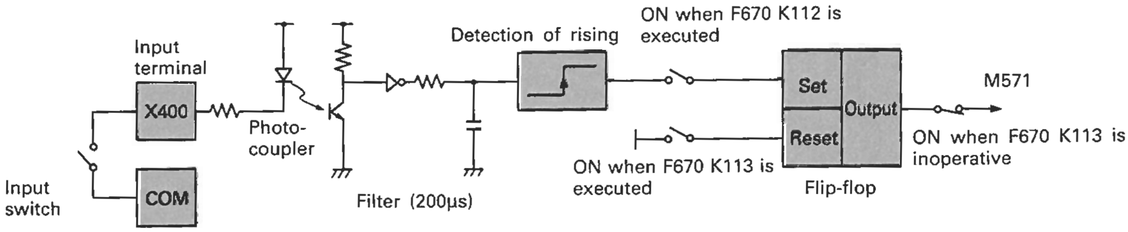


Fig. 5-6 X400 rising detection equivalent circuit

When F670 K112 is executed by the execution of program, the ON operation of input X400 is detected thereafter by hardware regardless of program execution, by which the flip-flop is set.
The flip-flop is reset when F670 K113 instruction is turned on and the program is executed.
The flip-flop operation is detected by the flag M571 when F670 K113 instruction is executed.
(If the flip-flop is set and F670 K113 is executed-OFF, the flag M571 is turned on.
When F670 K113 is executed-ON, the flag M571 will be turned off).
Accordingly, program M571 contact after F670 K113 instruction.

Application example-3

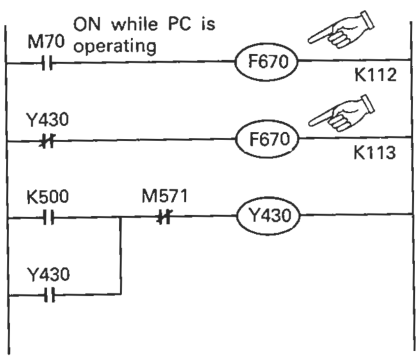


Fig. 5-7 Detection of X400 rising

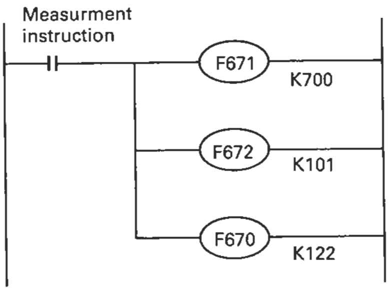
If F670 K112 is kept turned on all the time by M70, X400 can be detected even if X400 is turned on with the narrow width pulse (approx. more than 200µs) which may occur any time, making it possible to reset the self-holding output Y430.

In the example of Fig. 5-5, the X400 accepts only the pulse width of more than 6ms, however, it is possible to response at the pulse width of 200µs in this example.
However, resetting of the output Y430 requires the time of one execution cycle.

F670 K122

Measurement of X402 pulse signal width

Object elements: X402



Designation of measurement value storage register

Designation of measurement format

Execution of measurement

- It is possible to measure X402 ON (OFF) signal pulse width in increment of 1msec. when the measurement instruction is turned on.
  The measurement contents in this case will follow the measurement format shown below;
- Measurement format

F672 K = 1 0 1

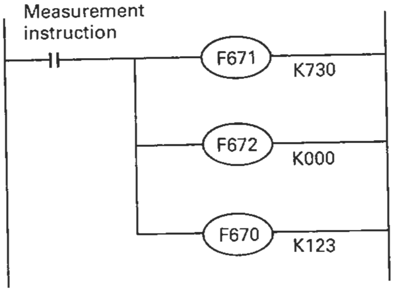
- Designation of reset function
  - 0: Data register is reset at rising of applied instruction.
  - 1: Data is continued.
- Designation of measurement signal
  - 0: Depending upon X402 signal
  - 1: Measurement starts with applied instruction turned on.
    (Usages include 1-msec. timer, calculation time measurement, etc.).
- Designation of signal level
  - 0: X402 is measured for OFF signal width.
  - 1: X402 is measured for ON signal width.

- No measurement is performed while instruction is turned off.
  In this case, the data register value will remain unchanged, and M570 is turned off.
- Data in data register ranges form "0" to "999".
  When it is incremented +1 from 999, it turns to be "0", however, it will not be shifted up, nor zero flag is generated.
- X402 filter is 0-msec.
- In case there occurs any setting error is data register No., the error flag M570 is turned on and no measurement is executed.

F670 K123

Measurement of X403 pulse signal width

Object element: X403



Designation of measurement value storage register

Designation of measurement format

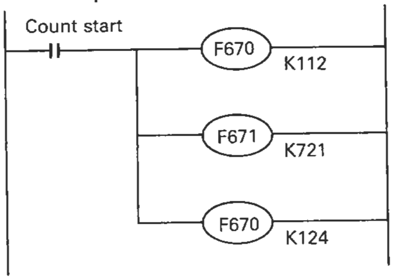
Execution of measurement

- This instruction executes the same operation as F670 K122.
  The difference with F670 K122 is only the object element.

F670 K124

X400 pulse signal count

Object element: X400



X400 rising detection function

Setting of data register No.

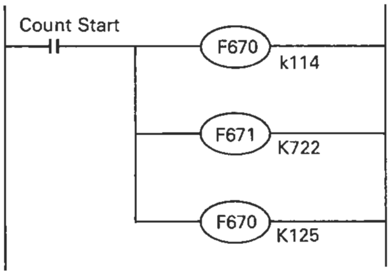
Execution of count

- Rising of pulse applied to X400 is counted by the data register designated by F671 when counting is started.
- Data register No. ranges from "D700" to "D777" and data value ranges from "0" to "999".
  When it incremented +1 from "999", it will not be shifted up, nor zero flag is generated.
- Resetting is made by writing "0" in the data register.
- No counting is executed when count-start is off.
- The instruction is used in combination with the rising detection function of K112.
  F670 K113 may not be necessarily used.
- When there occurs any setting error in data register No., the error flag M570 is turned on, and no measurement is executed.

F670 K125

X401 pulse signal count

Object element: X401



X401 rising detection function

Setting of data register No.

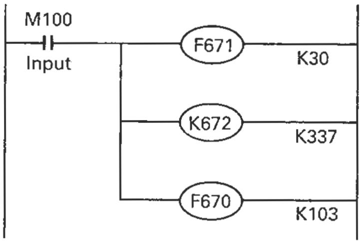
Execution of count

- This instruction executes the same operation as F670 K124.
  The difference with F670 K124 is only the object element.
- The instruction is used in combination with the rising detection function of F670 K114.
  F670 K115 may not be necessarily used.

5-2 Instructions concerning reset

F670 K26 (or K103)
Simultaneous reset

Object elements: Y030~Y547
M100~M377
S600~S647



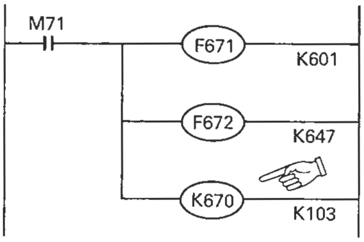
Reset start element No.
(Y30 in this example)

Reset end element No.
(M337 in this example)

Simultaneous reset of designated range
(Reset of Y30~Y47 and M100~M337 in this example)

- When the input is turned on, the image memory of all element from the K number of F671 to the K number of F672 are reset.
- In case the start/end element No. is other than Y, M100~M377 and S, the error flag M570 is turned on, and the simultaneous reset will not be executed.
- If the input is turned off, no processing will be executed, and the error flag M570 will be turned off.
- Be sure that the reset start No. is smaller than the reset end No. when setting.
  If the reset start No. is equal to or greater than the reset end No., only the start No. is reset.
- Bear in mind that the timer, counter and other special auxiliary relay other than the object elements will not be reset in the above example.

Application example-4
Simultaneous reset of state



The simultaneous reset instruction is used for initial resetting of state and auxiliary relay backed up by the battery and for resetting of element jumped by the jump instruction.

Bear in mind that no resetting is executed when F670 K103 instruction is jumped.

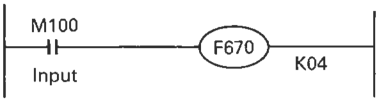
Fig. 5-8 Simultaneous reset of state

Even when the step ladder circuit is jumped by the jump instruction, these states will be reset when the simultaneous reset of state is executed.
When the output connected to the step ladder contact needs to be reset at the same time, program the simultaneous reset instruction for output Y at the same time, as such outputs remain at the current situation.

F670 K04

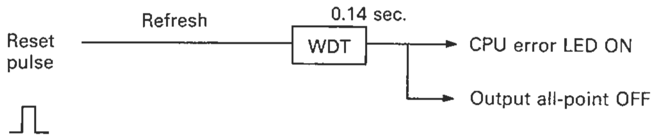
WDT refresh

Object element: Watch-dog timer



This instruction is not provided with any setting coil, and is established only by execution coil.

When the input (M100 in the above example) is turned on, the operation delay monitoring timer (WDT) will be refreshed.



WDT is refleshed automatically for each END instruction. However, it can be also refleshed by this instruction.
No refleshing is executed while the input is turned off. (NOP processing)
Once WDT is operated, the power source needs to be turned off to reset the operation.

F670 K10 (or K110)

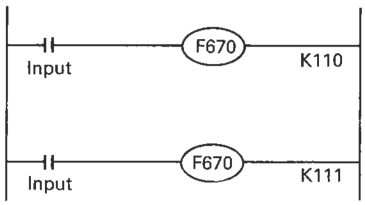
Reset of M473

Object element: M473 (shift up/down flag)

F670 K11 (or K111)

Reset of C660

Object element: C660 (pair counter)



Reset the shift up/down flag M473 of pair counters 660/661. (Current value will not be changed).

Reset the output contact C660 of pair counters C660/C661. (Current value will not be changed).

When the current value reaches "0", the output contact C660 of the pair counters C660/C661 is turned on, thereby the output contact C660 remains turned on even if the current value is changed.
F670 K111 instruction is used for its resetting.
F670 K110 instruction is similarly used to reset M473 which is set when the pair counters are turned "999999" to "0" (shift-up) or from "0" to "999999" (shift-down).
Since these instructions are used to turn off the M473/C660 contacts, the current value of counter will remain un-changed.
The output contact C660 may be also turned off by RST C660 instruction.
In this case, the current value of counter is preset to the setting value.
In addition, M473 may be set when auto-reload (see page 137) is executed, and can be reset by F670 K110 instruction.

Application example-5

9-digit down counter

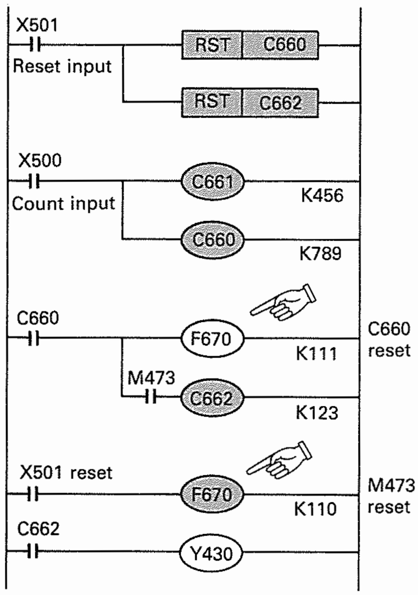
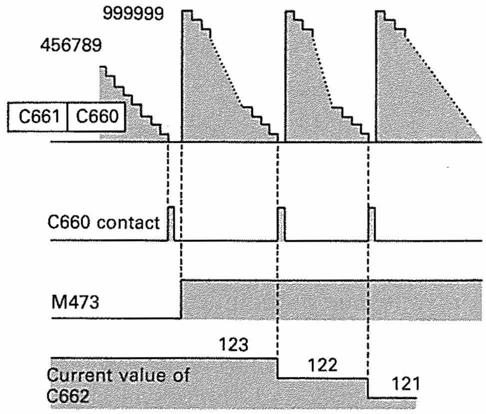


Fig. 5-9 9-digit down counter

9-digit counter can be programmed by using the pair counters C660/C661 as down-counter and by counting the operation of output contact by the use of cascade counter C662.



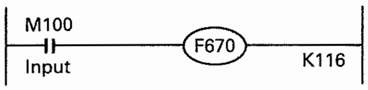
F670 K116

External reset inhibit

Object element: C660, C661

When the special auxiliary relay M470 is turned on, the pair counters C660 and C661 are turned to high-speed counter mode, by which the X400 becomes count input and the X401 becomes the reset input to execute the interrupt counting and interrupt reset.

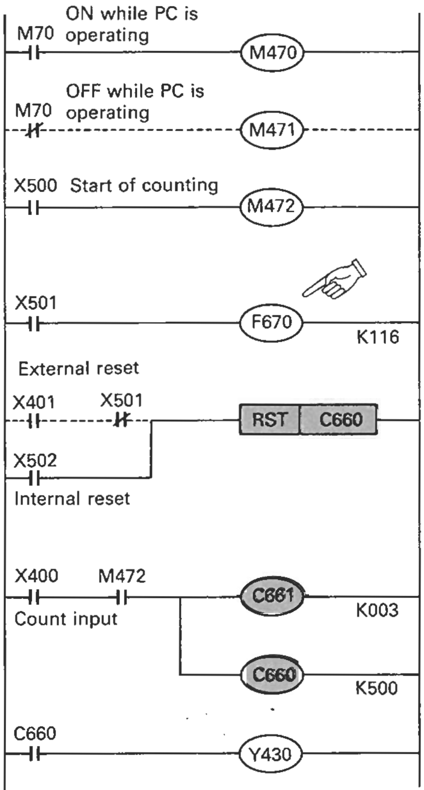
If the external reset inhibit instruction is used to reset by the internal program, X401 may be utilized for the other application.



High-speed counter reset inhibit by input X401

- While the input (M100 in the above example) is turned on, resetting function of the high-speed counters C660/C661 by X401 will be prohibited.
- If the input X401 is turned on while the input (M100 in the above example) is turned off, the setting value is preset automatically at the current value register of the high-speed counter C600/C601.
  (Output contact C660 will not be changed).
  This program is not necessary when the reset function of X401 is used all the time.

Circuit example
External reset inhibit



The pair counters C660/C661 are turned to high-speed counter mode.

The pair counters C660/C661 are turned to down-counters.

Counting starts when M472 is turned on, and stops when it is turned off.

When X501 is turned on, automatic rest inhibit by X401.

The same operation will result even if the counter reset circuit (shown with dotted line) may not be programmed.
(Output C660 is turned off, causing the setting value to be preset in the current value register).

Upper 3-digit setting value 003
(to be programmed first)

Lower 3-digit setting value 500
(to be programmed later)

C660 contact is turned on at the count of 3500.

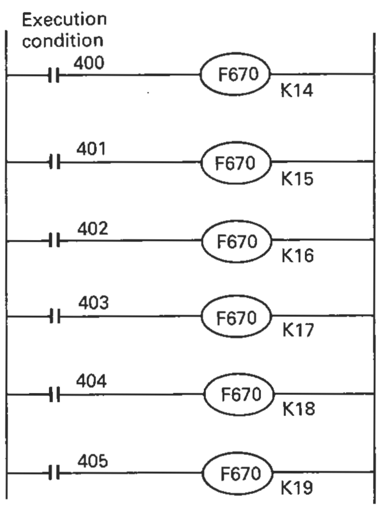
Fig. 5-10 High-speed counter

The same operation will result even if M70 is used in substitute for X400, X472 in the count input circuit.
In this case, the counter will automatically count the number by OFF→ON of X400.

F670 K14~19
Flag set/reset

Object element: M571~573

- This instruction is not provided with setting coil, and established only with execution coil.



Carry flag M571 set (ON) with execution condition established

Carry flag M571 reset (OFF) with execution condition established

Zero flag M572 set (ON) with execution condition established

Zero flag M572 reset (OFF) with execution condition established

Borrow flag M573 set (ON) with execution condition established

Borrow flag M573 reset (OFF) with execution condition established

- F670 K14 .... Carry flag M571 set
  F670 K15 .... Carry flag M571 reset

- F670 K16 .... Zero flag M572 set
  F670 K17 .... Zero flag M572 reset

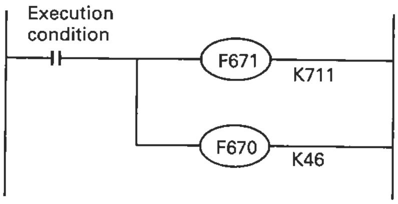
- F670 K18 .... Borrow flag M573 set
  F670 K19 .... Borrow flag M573 reset

- No processing is executed when execution condition is turned off. (NOP processing).

F670 K46

Data register zero check

Object element: D700~777



Designation of data register No.

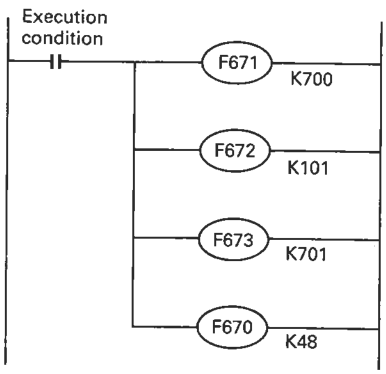
Zero check instruction

- Check is made on whether value in data register designated is "0" when execution condition is turned on.
  No processing is executed when execution condition is turned off.
- In case data register setting error occurs, M570 is turned on, causing the instruction not to be executed.
- M572 is turned on when the value in data register is "0".
  M572 is turned off when the value in data register is other than "0".
- Carry flag M571 and borrow flag M573 are turned off.

F670 K48

Clear of designated digit

Object element: D



Designation of data register No.

Designation of clear digit

Designation of data register No. storing result

Clear execution

- Clear is executed when execution condition is turned on, and no processing is executed when execution condition is turned off.
- In case data register No. setting error occurs, error flag M570 is turned on, causing the instruction not to be executed.
- To designate clear digit, use F672 by changing the digit to clear, to"0".

Example: (D700) ∧ K1 0 1 → (D701)

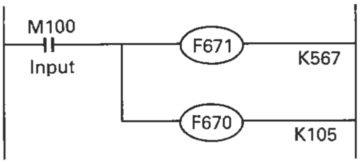
$$\underset{\text{Address}}{(D700)} = 345 \quad\quad (D701) = 305$$

3 4 5 — Contents of data register

5-3 Data (numeral value) transfer instruction

F670 K105
READ C→M

Object element:
- Data source : Counter (C060~667)
- Data transfer destination : M260~M273



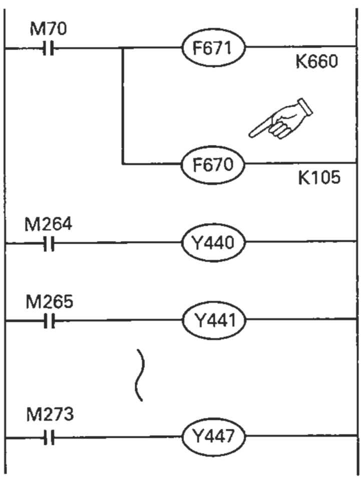
Designation of counter No. (C567)

Transfer the current value of counter designated (BCD 3-digit) to M260~M273.

- Execute the transfer when the input is turned on.
  When there occurs any setting error in the counter No., the error flag M570 is turned on, but the instruction will not be executed.
- If the input is turned off, no processing will be executed, and the error flag M570 is turned off.
- The transfer data is of BCD 3-digit, the lowermost digit of which is M260 and uppermost digit of which is M273.

M273	M272	M271	M270	M267	M266	M265	M264	M263	M262	M261	M260	
OFF	ON	OFF	OFF	OFF	OFF	ON	ON	ON	OFF	OFF	ON	◊ 439
800	400	200	100	80	40	20	10	8	4	2	1	
Digit of 100				Digit of 10				Digit of 1				

Application example-6
Indication of current counter value



Transfer of current value in C660 to M260~M273

Output of upper two digits of M260~M273 to Y440~Y447

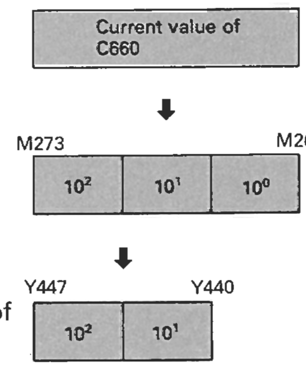
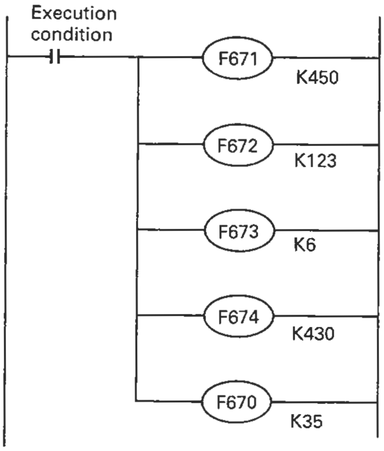


Fig. 5-11 Output of current counter value

Since the F_1 series programmable controller is not provided with the instruction to transfer the BCD data (binary decimal) to the output-Y, each bit is output to the external unit by the LD instruction as shown in Fig. 5-11.

F670 K35
READ OF T, C, D RESISTER

Object element:
- Transfer origin : T, C, D
- Transfer destination : Y, M, S



Timer No. set

Adding bias (0.0~999)

Transfer form set

Designation of transfer destination (Y430~Y437)

Current value READ instruction

- Data register (table for setting value) of timer, counter, or contents of current data register is read to Y, M100~377 or S.F673 is used to designate how many digit of which digit of current data register to read.
  Same as transfer form −12−.

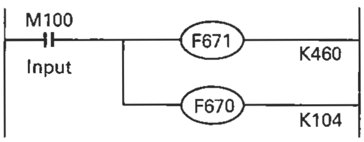
  Example
  In case current timer value is "72.9", for instance, the value to be read is 85.2 (=72.9+12.3).
  In the case of transfer form K6, "5.2" will be read.

Y437	Y436	Y435	Y434	Y433	Y432	Y431	Y430
8	④	2	①	0.8	0.4	(0.2)	0.1

  Consequently, Y436, Y434 and Y431 will be turned on.
- When there occurs any setting error in transfer destination No,. M570 is turned on, and no reading is executed.
- Be sure to set "0" for least significant digit at head address of transfer destination.
  If value other than "0" is set, M570 is turned on, by which reading will not be executed.

F670 K104
WRITE M→C

Object element:
- Data source : M260~M273 (BCD 3-digit)
- Data transfer destination : Current value register of counter (C060~667)



Designation of object counter No. (C460)

Write BCD 3-digit data of M260~M273 to the current value register of counter designated.
(M260 ... lowermost digit/M273 ... uppermost digit)

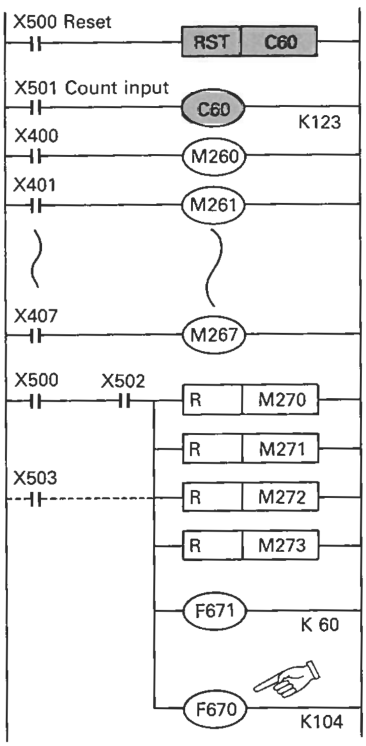
- Set the transfer data to the auxiliary relays M260~M273 in advance.

M273	M272	M271	M270	M267	M266	M265	M264	M263	M262	M261	M260	
OFF	OFF	OFF	ON	ON	OFF	OFF	ON	OFF	ON	ON	ON	⇨ 197
800	400	200	100	80	40	20	10	8	4	2	1	
Digit of 100				Digit of 10				Digit of 1				

- The transfer is executed when the input is turned on.
  In this case, the output contact of counter will not be changed.
  In case there occurs any counter No. setting error, or the figure at each BCD digit exceeds "9", the error flag M570 is turned on, and no transfer is executed.
- In case the input is turned off, no transfer is executed, and the error flag M570 is turned off.

Application example-7

Operation on panel surface for counter setting value



When X500 is turned on, the counter C60 is reset and the output contact is turned off and the setting value of K123 is written in the current value register. (This writing is executed at the time when OUT C60 is executed).

The current counter value is decremented, depending upon ON/OFF of X501.
The output contact is turned on when the value reaches "0".
(No counting is executed while X500 is turned on).

LD instruction is used for transfer to M260~M267 with 2-digit digital switch connected to the input X400~X407.
(F_1 series programmable controller is not provided with any instruction to fetch the BCD data input).

The contents of M260~M267 will be varied by adjusting the digital switch provided on the operation panel.

The upper one digit of M260~M273 (BCD 3-digit) is reset.
In case only X500 is turned on, the functional instruction F670 K104 is not executed, but K123 becomes valid for counter setting value.

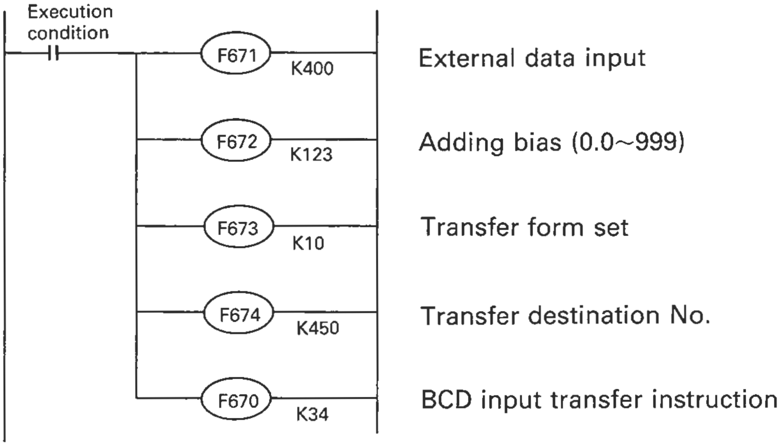
When both X500 and X502 are turned on at the same time, the counter setting value is changed from K123 to BCD data of M260~M273.

This writing operation may be executed while the counter is operating.
(When X503 is turned on, the current counter value will become BCD data of M260~M270).

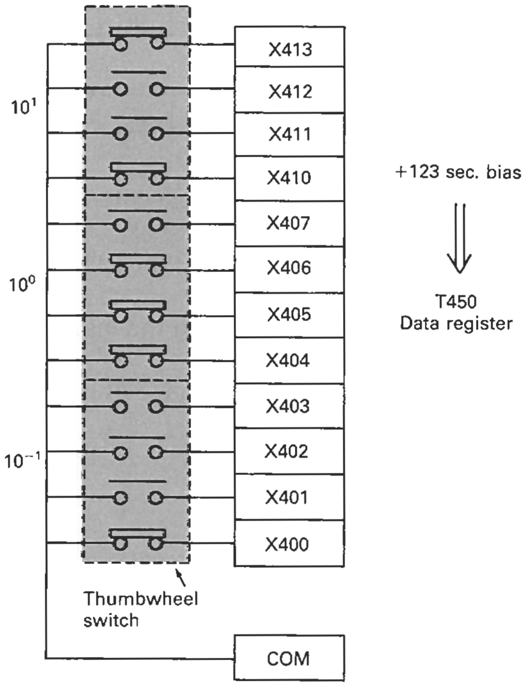
Fig. 5-12 Operation on panel surface for counter setting value

F670 K34 BCD INPUT WRITE

Object element:
- Transfer origin : M100~377, S
- Transfer destination : T, C, D



The bias value is added to the BCD input (most insignificant digit of head element No. : 0) of 1~3 digit (X, Y, M, S), and the resulting value is transferred to the data register (setting value register) or current value register.
The transfer form is designated by F673 to determine the number of digit for input signal, and positional notation.



Transfer form

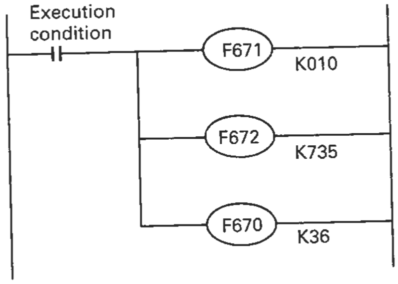
	Number of digit and positional notation				
K0	0	0	0	0	10^{-2}
K1	0	0	0	10^{-1}	0
K2	0	0	10^0	0	0
K3	0	10^1	0	0	0
K4	10^2	0	0	0	0
K5	0	0	0	10^{-1}	10^{-2}
K6	0	0	10^0	10^{-1}	0
K7	0	10^1	10^0	0	0
K8	10^2	10^1	0	0	0
K9	0	0	10^0	10^{-1}	10^{-2}
K10	0	10^1	10^0	10^{-1}	0
K11	10^2	10^1	10^0	0	0

Taking up the above case, for instance, 109.4 second of 109.4 sec. {=97.1 (setting value of thumbwheel switch) + 12.3 sec. (bias)} will be transferred to the transfer destination timer No. 450.

- When transfer condition is turned on, input is transferred.
  If it is turned off, no processing is executed, and M570 remains turned off.
- In case there occurs any setting error in element No., M570 is turned on, and no processing will be executed.
- When the 3rd digit appears to become non-object element in 3-digit transfer, M570 is turned on, and no processing is executed.

F670 K36
TRANSFER OF DECIMAL DATA (BCD)

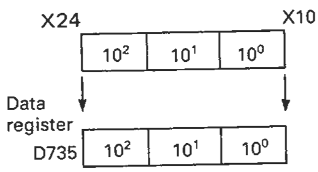
Object element:
- Transfer origin :
  X, Y, M100~377, S
- Transfer destination :
  D700~777



Head No. of transfer origin BCD 3 digits

Data register No. of transfer destination

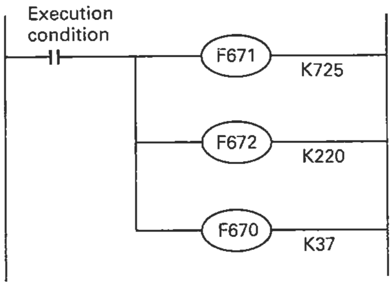
Transfer instruction



- 12-bit BCD data from head No. is trnasferred to data register.
- The data is transferred when execution condition is turned on.
  No processing is executed and M570 remains OFF when execution condition is turned off.
- In case element No./data register No. setting error occurs, error flag M570 is turned on, causing the instruction not to be executed.
- Do not fail to set most significant digit for head No. of transfer origin at "0".
  In case it is other than "0", M570 is turned on, by which the instruction will not be executed.
- In case input is not in BCD format, error flag M570 is turned on, causing the instruction not to be executed.
- In case part of data is out of object element at transfer origin, error flag M570 is turned on, causing the instruction not to be executed.
- M570 is turned off when processing is executed normally.

F670 K37
WRITE D→Y, M, S

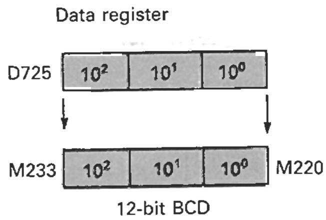
Object element:
- Transfer origin : D700~777
- Transfer destination : Y, M100~377, S



Data register No. of transfer origin

Head No. of transfer destination

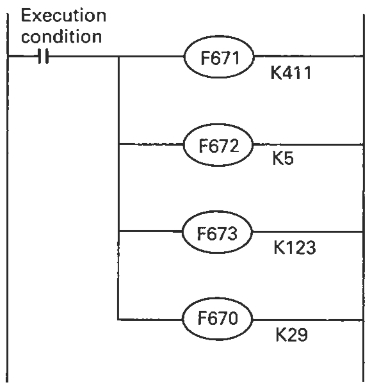
Transfer instruction



- The data is transferred when execution condition is turned on.
  No processing is executed and M570 remains OFF when execution condition is turned off.
- In case element No./data register No. setting error occurs, error flag M570 is turned on, causing the instruction not to be executed.
- Do not fail to set most significant digit for head No. of transfer origin at "0".
  In case it is other than "0", M570 is turned on, by which the instruction will not be executed.
- In case part of data is out of object element at transfer destination, error flag M570 is turned on, causing the instruction not to be executed.
- M570 is turned off when processing is executed normally.

F670 K29
TRANSFER OF nBit

Object element:
- Transfer origin : X, Y, M100~377, S
- Transfer destination : Y, M100~377, S

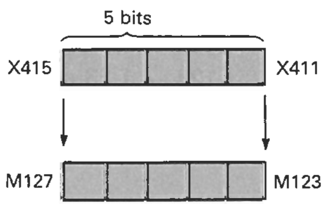


Head element No. of transfer origin

Number of transfer bit (1~16 bit)

Head element No. of transfer destination

Transfer instruction

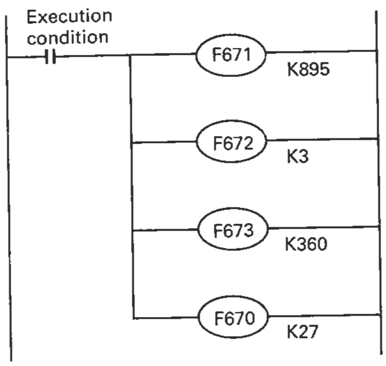


- The data is transferred to designated transfer destination when execution condition is turned on.
  No processing is executed and M570 remains OFF when execution condition is turned off.

- In case element No. setting error occurs, error flag M570 is turned on, and no transfer is executed.
- Number of transfer bit is in the range from 1 bit to 16 bits.
  For other than the range, error flag M570 is turned on, and no transfer is executed.
- In case transfer origin or element No. of transfer destination exceeds Y647, transfer is terminated.
  In this case, M570 remains OFF.
- In case transfer origin or transfer destination element appears to be non-object element during transfer, the bit will not be transferred.
  In this case, transfer is executed continuously with M570 turned off.

F670 K27
TRANSFER OF DECIMAL CONSTANT

Object element: Y030~Y547
M100~M377
S600~S647

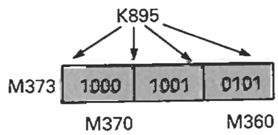


Setting of decimal constant 895

Number of transfer digit

Head element No. of transfer destination

Transfer instruction

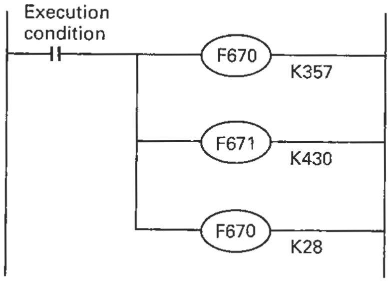


- Data is transferred to the designated transfer destination when execution condition is turned on.
  If it is turned off, no processing is executed, and M570 is turned off.
- When there occurs any setting error in element No., error flag M570 is turned on, and no processing is executed.
- In this case, the most significant digit of head No. necessarily becomes "0".
  When it is other than "0", M570 is turned on, by which no transfer will be executed.
- The number of transfer digit ranges from "1" to "3" digits.
  If it is out of the range, error will occur, by which M570 is turned on, and no transfer will be executed.
- If such head No. is set, of which 3rd digit may become out of object element in the case of three-digit transfer, M570 is turned on, and no transfer is executed.
- Handling of constant for T650~657 (0.01-sec. time)
  K12.3 → 12.30 sec.
  K123 → 1.23 sec.
- In the case of transfer to constant counter/data register with decimal point, the decimal point is ignored.

F670 K28

TRANSFER OF OCTAL 3-DIGIT CONSTANT

Object element:
Y030~Y547
M100~M377
S600~S647

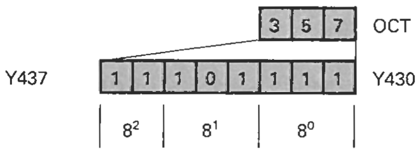


Setting of octal constant 357

Head element No. of transfer destination

Transfer instruction

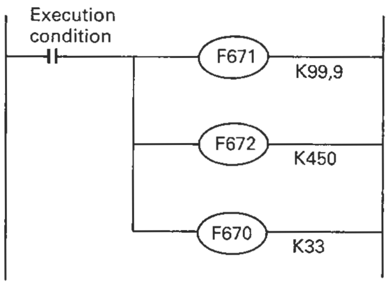
- When execution condition is turned on, data is transferred to the designated transfer destination.
  When it is turned off, no processing is executed and M570 is turned off.
- In case there occurs setting error in element No., error flag M570 is turned on, and no transfer is executed.
- The most significant digit of head No. necessarily becomes "0".
  If it is other than "0", M570 is turned off, and no transfer is executed.
- The constant ranges from "0" to "377"(octal constant).
  If it is more than 400 (octal), M570 is turned on, and no transfer is executed.
  In this case, decimal point is ignored.
- Data format is as shown below;



F670 K33

DECIMAL CONSTANT WRITE TO CURRENT VALUE REGISTER

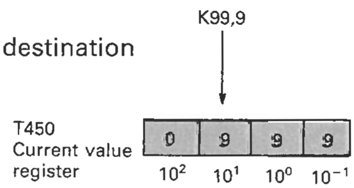
Object element: T, C, D



Setting of decimal 3-digit constant

Element No. of transfer destination

Transfer instruction



- The data is written to designated register when execution condition is turned on.
  No processing is executed and M570 remains OFF when execution condition is turned off.

- In case error occurs in setting of transfer destination element No., error flag M570 is turned on, and no writing is executed.

Appendix: Handling of constant for timer

(1) 0.1-sec. timer

T50~T57, T450~T457, T550~T557

K999→999sec., K99.9→99.9sec., K0.1→0.1sec.

(2) 0.01-sec. timer

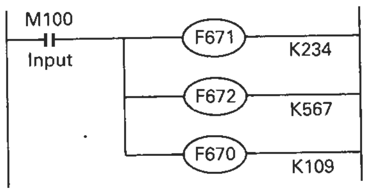
T650 ~ T657

K999 → 9.99sec., K1→0.01sec. (Setting without decimal point)

K99.0 → 99sec., K1.0→1sec. (Setting with decimal point)

F670 K109
TRANSFER OF DECIMAL CONSTANT

Object element: M240~M253
M260~M273

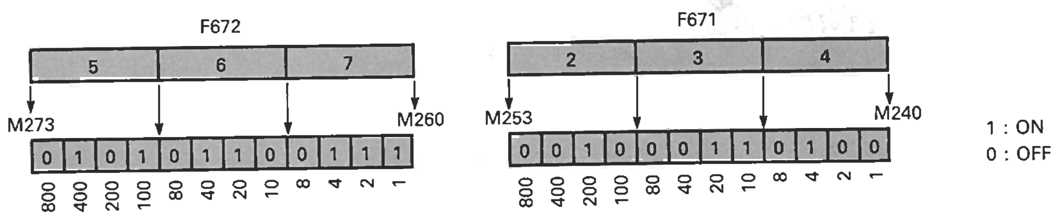


Setting of decimal constant 234 (0~999)

Setting of decimal constant 567 (0~999)

234→M240~M253   567→M260~M273

- When the input is turned on, the transfer will be executed. If the input is turned off, no processing will be executed.
- The decimal constant set by F671 is transferred to M240~M253 as 3-digit BCD data. (In this case, M254~M257 will not be able to use as general aux. relay).
- The decimal constant set by F672 is transferred to M260~M273 as 3-digit BCD data. (In this case, M274~M277 will not be changed. M260~M273 will not be able to use as general aux. relay).



Application example-8
Changing of counter setting value

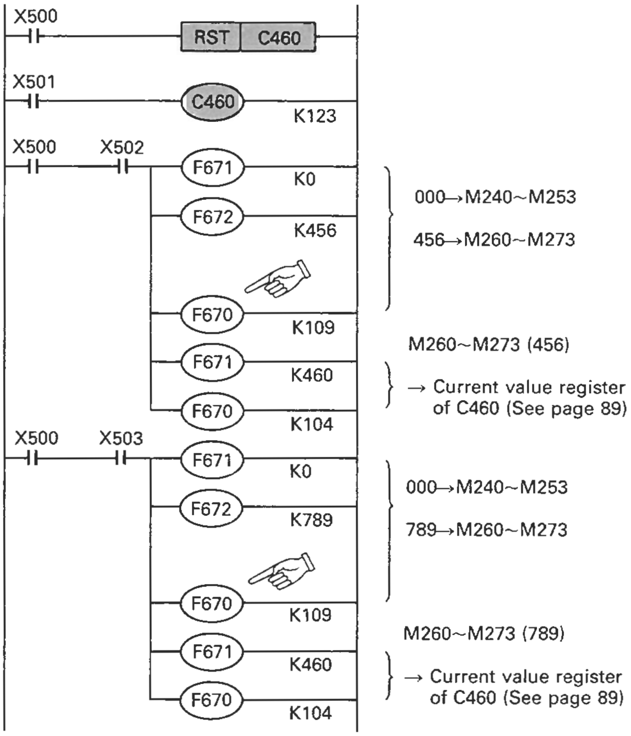


Fig. 5-13 Counter setting value changing

When X500 is turned on, the counter C460 is reset and the output contact is turned off and the setting value of K123 is written in the current value register when OUT C460 instruction is executed.

The current counter value is decremented, depending upon ON/OFF of X501.
The output contact is turned on when the value reaches "0".
(No counting is executed while X500 is turned on).

Note:

> Since the F_1 series programmable controller is not provided with any setting value register for counter, the constant is transferred to the current value register. Make it a rule to execute the transfer before starting the counting (during reset).

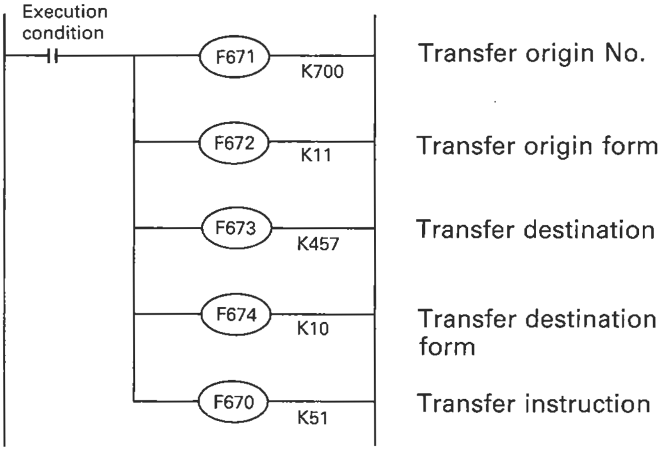
Initial setting value of counter

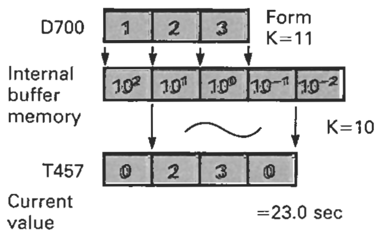
X500	X502	X503	Setting values
ON	OFF	OFF	123
ON	ON	OFF	456
ON	OFF	ON	789
ON	ON	ON	789

Note: In the above example, the counter (C460) setting value is changed in 3 different ways.

F670 K51
TRANSFER OF CURRENT RESISTER AND DATA RESISTER

Object element:
T, C, D700~777



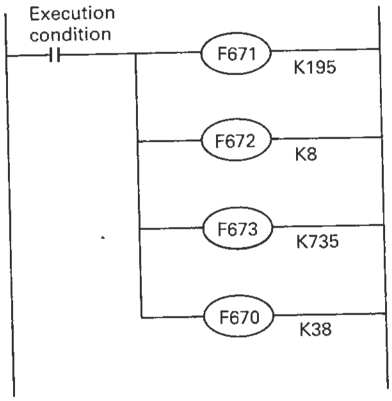


- Transfer is executed when execution condition is turned on, and no processing is executed when execution condition is turned off.
- In case error occurs in setting of transfer origin No. and transfer destination No., error flag M570 is turned on, and no processing is executed.
- Transfer origin/transfer destination form is necessary only when each of transfer origin and transfer destination is timer repectively.
  (Refer to page- "Transfer form").
  For 9, K9, K10 or K11 only
- In case transfer form setting error occurs, M570 is turned on, and no processing is executed.

F670 K38

TRANSFER OF THE SAME DATA n-TIMES (CONSTANT)

Object element: D700~777

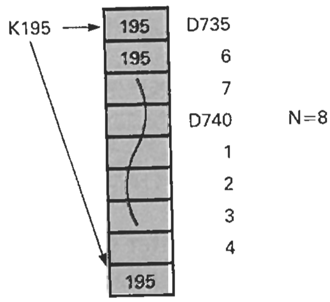


Original constant data: K

Number of transfer: N

Head No. of transfer destination: D

Transfer instruction

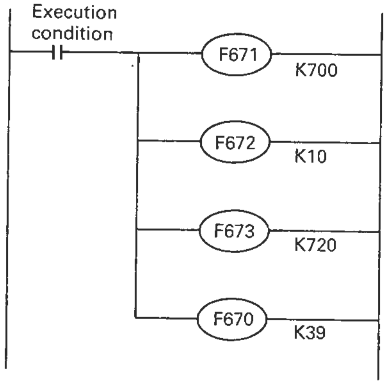


- Constant data expressed by "K" is transferred to the data register in the addresses ranging from its head address "D" to "N" addresses ahead.
  Constant K $\rightarrow$ (D), (D_{+1}), (D_{+2}), ... (D_{+N-1})
- Transfer is executed when execution condition is turned on.
  No processing is executed and M570 remains OFF when execution condition is turned off.
- In case error occurs in setting of data register head address, error flag M570 is turned on, and no processing is executed.
- The number of transfer is 1 to 64.
  When it is zero or exceeds 65, M570 is turned on and no processing is executed.
- Decimal point of constant is ignored.

F670 K39

TRANSFER OF THE SAME DATA n-TIMES (DATA RESISTER)

Object element: D700~777



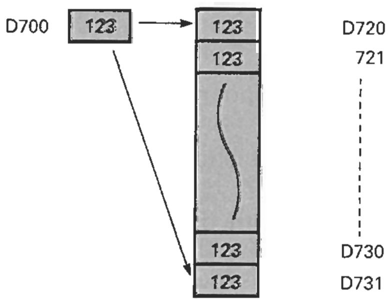
Contents of original data register: D_1

Number of transfer: N

Head No. of transfer destination: D_2

Transfer instruction

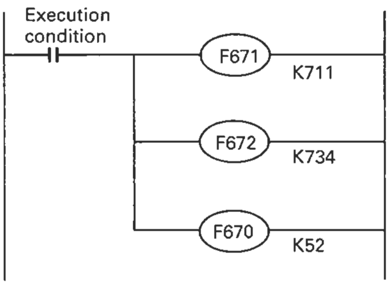
- Contents in data register designated by D_1 is transferred to head address designated by D_2, and N number of data registers starting form D_2.
- Transfer is executed when execution condition is turned on, and no processing is executed when execution condition is turned off.
- In case error occurs in setting of data register No., error flag M570 is turned on, and no processing is executed.
- The number of transfer is 1 to 64.
  When it exceeds 65, M570 is turned on and no processing is executed.



F670 K52

INDIRECT MOVE (D)→D

Object element: D700~777

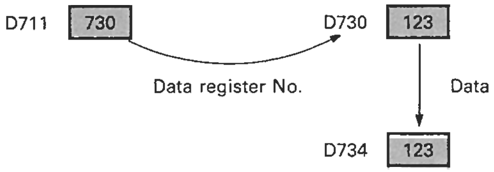


Indirectly designated data register No.

Data register No. of transfer destination

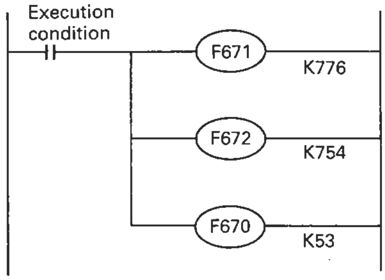
Transfer instruction

- Transfer is executed when execution condition is turned on, and no processing is executed when execution condition is turned off.
- In case error occurs in setting of data register No., error flag M570 is turned on, and no processing is executed.
- When data in indirectly designated register is not consistent with data register No., M570 is turned on and no processing is executed.



F670 K53
INDIRECT MOVE D→(D)

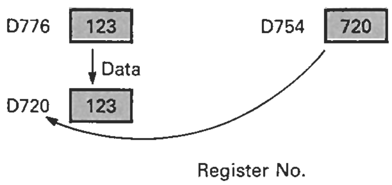
Object element: D700~777



Transfer origin No.

Indirectly designated data register No.

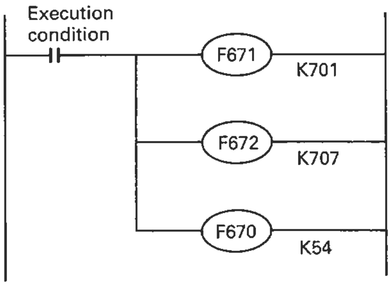
Transfer instruction



- Transfer is executed when execution condition is turned on, and no processing is executed when execution condition is turned off.
- In case error occurs in setting of data register No., error flag M570 is turned on, and no processing is executed.
- When data in indirectly designated register is not consistent with data register No., M570 is turned on and no processing is executed.

F670 K54
INDIRECT MOVE (D)→(D)

Object element: D700~777



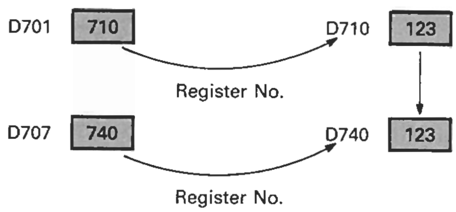
Indirectly designated data register No. of transfer origin

Indirectly designated data register No. of transfer destination

Transfer instruction

- Transfer is executed when execution condition is turned on, and no processing is executed when execution condition is turned off.

- In case error occurs in setting of data register No., error flag M570 is turned on, and no processing is executed.
- When data in indirectly designated register is not consistent with data register No., M570 is turned on and no processing is executed.

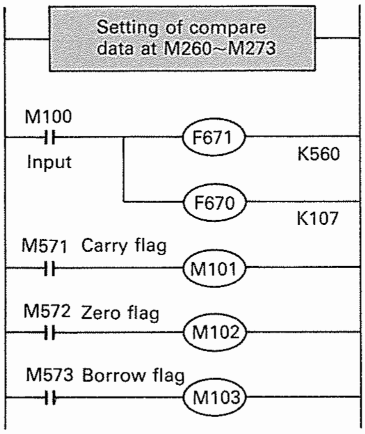


5-4 Compare instruction for current counter value

F670 K107

COMPARISON OF BCD DATA WITH CURRENT VALUE OF C

Object element:
Counter (C060~C667)
BCD data (M260~M273)



Use functional instruction F670 K109 for setting of constant. (See page-101).

Use LD instruction for input from the external unit by the use of digital switch.

Setting of counter No. C560

Comparison of current counter value of C560 with M260~M273

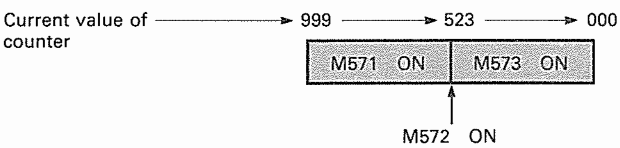
ON: BCD data < Current value

ON: BCD data = Current value

ON: BCD data > Current value

- When the input is turned on, the compare will be executed.
  In case there occurs any setting error of counter No., or error in the BCD data (in case each digit exceeds "9"), the error flag M570 is turned on, and no compare is executed. In this case, M571~M573 are all turned off.
- In case the input is turned off, no processing is executed, and M570~M573 are all turned off.

M273	M272	M271	M270	M267	M266	M265	M264	M263	M262	M261	M260	
OFF	ON	OFF	ON	OFF	OFF	ON	OFF	OFF	OFF	ON	ON	⇨ 523
800	400	200	100	80	40	20	10	8	4	2	1	
Digit of 100				Digit of 10				Digit of 1				



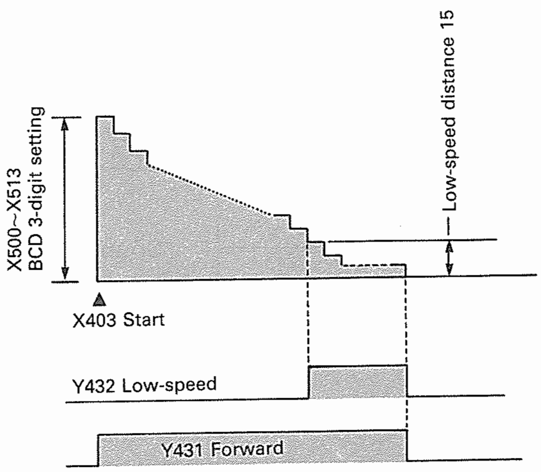
NOTE

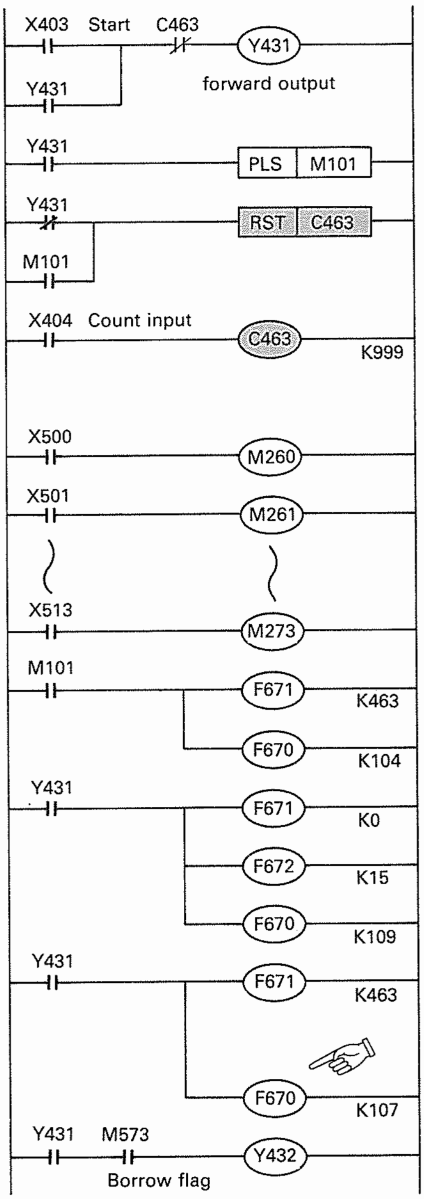
Handling of flag

When the compare instruction is used for multiple times, the flag is changed every time the compare instruction is executed.

When it is output to the aux. relay of different No. for each compare data, it is possible to obtain "Great", "Small" or "Coincidence" for each compare instruction.

Application example-9
Simple positioning control





→ Load 3-digit setting value to M260~M273 by the use of digital setting switch connected to the input of X500~X513. (example: 789)

→ The current value of counter C463 is set to BCD value (789) of M260~M273.
The value will appear to be the total forward running distance. (See page 89).

0→M240~M253
15→M260~M273
Set the low-speed setting distance at "15".
(See page 96).

→ Compare the current value of counter C463 with value of M260~M273 (15).

→ Low-speed operation

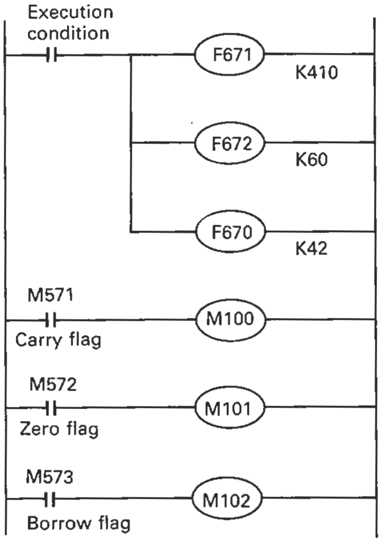
Fig. 5-15 Simple positioning control

In this example, the motor is operated at two-stage speed of high-speed and low-speed to control the machine movement distance.
The total movement distance is set variably by the 3-digit digital switch connected to the input of X500~X513.
The low-speed operation distance is set with constant by F670 K109 to generate low-speed output by compare instructions of K670, K107.

F670 K42

COMPARISON OF BCD INPUT WITH C AND DATA REGISTER

Object element:
(1) D700~777,C
(2) X, Y, M, S



BCD input element No.: S_1

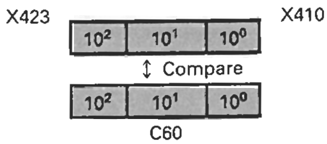
Counter/data register No.: S_2

Compare instruction

ON: $S_1 < S_2$

ON: $S_1 = S_2$

On: $S_1 > S_2$



Data of counter/data register is compared with BCD 3-digit input value.

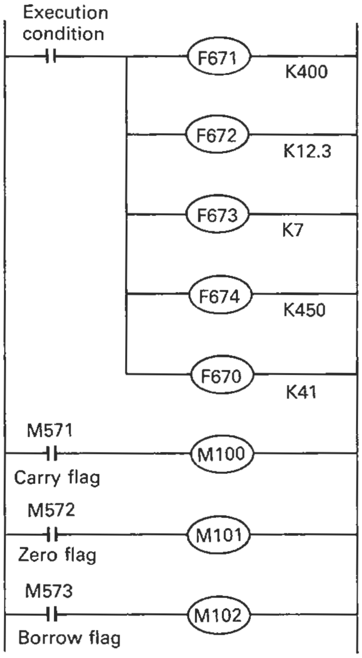
- Comparison is executed when execution condition is turned on, and no processing is executed when execution condition is turned off.
- In case error occurs in setting of counter/data register No., error flag M570 is turned on, and no processing is executed.
- In case input data is not BCD code, M570 is turned on, and no processing is executed.
- In case error occurs in setting of BCD input element No., M570 is turned on, and no processing is executed.
- Following table shows a list of operations for each flag;

F670 operation	Setting of F671/F672 No.	Cal. result	Error flag M570	Carry flag M571	Zero flag M572	Borrow flag M573
OFF	—	—	OFF	OFF	OFF	OFF
ON	Error	—	ON	OFF	OFF	OFF
ON	O.K.	$S_1<S_2$	OFF	ON	OFF	OFF
ON	O.K.	S = S	OFF	OFF	ON	OFF
ON	O.K.	$S_1>S_2$	OFF	OFF	OFF	ON

F670 K41

COMPARISON OF BCD INPUT WITH CURRENT VALUE OF T, C, D

Object element:
(1) T, C, D700~777
(2) X, Y, M100~M377, S



External input data (X400 ~ X407)
Ex. 45

Adding bias (0.0 ~ 999)

Compare form
(Same as page transfer form)

} : S_1

Setting of compare object T.C.D No.: S_2

Compare instruction

ON: Input + Bias < Current value

ON: Input + Bias = Current value

ON: Input + Bias > Current value

Bias value (12.3) is added to BCD input (45) of 1~3 digit to compare the terms 10^1, 10^0, or 57 with current register value of timer, counter.
The same compare form as transfer form shown on page-91 is set to determine the number of digit for input signal, and positional notation.

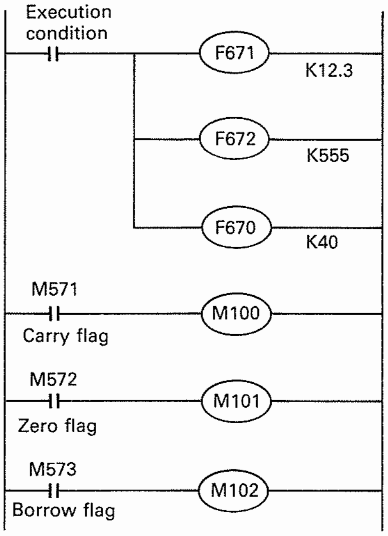
- Compare is executed when execution condition is turned on.
  When it is turned off, no processing is executed.
- In case there occurs any setting error in object element No. setting of T, C and D, error flag M570 is turned on, and no processing is executed.
  (M571~573 remain OFF in this case).
- When there occurs any error in transfer form or bias form setting, M570 is turned on, and no processing is executed.
- Be sure to set most significant digit for head element No. of 1~3 BCD input, to "0".
  If value other than "0" is set, M570 is turned on, causing it not to be executed.
  For 3-digit BCD input starting form Y430, 12 points of Y430~Y437 and Y440~Y443 will become objects.
- The following table shows a list of operations for each flag;

F670 input condition	Setting of F671	Comp. result	Error flag M570	Carry flag M571	Zero flag M572	Borrow flag M573
OFF	—	—	OFF	OFF	OFF	OFF
ON	Error	—	ON	OFF	OFF	OFF
ON	O.K.	$S_1 < S_2$	OFF	ON	OFF	OFF
ON	O.K.	$S_1 = S_2$	OFF	OFF	ON	OFF
ON	O.K.	$S_1 > S_2$	OFF	OFF	OFF	ON

F670 K40

COMPARISON OF DECIMAL CONSTANT WITH T, C, D

Object element:
T.C D700~777



Setting of decimal constant: K

Setting of T.C.D No.: S

Compare instruction

ON: Constant < Current value

ON: Constant = Current value

ON: Constant > Current value

Decimal constant is set by F671, and it is compared with current value of timer or counter, and with contents of data register designated by F672.
Each flag of carry, borrow and zero is operated, depending upon "Great", "Small" or "Coincidence" of compare result.

- Compare is executed when execution condition is turned on.
  No processing is executed when execution condition is turned off.
- When there occurs any setting error in each No. of each object element of T, C, D, error flag M570 is turned on, by which no processing is executed.
  (In this case, M571~M573 remain turned off).
- The following table shows a list of operation for each flag;

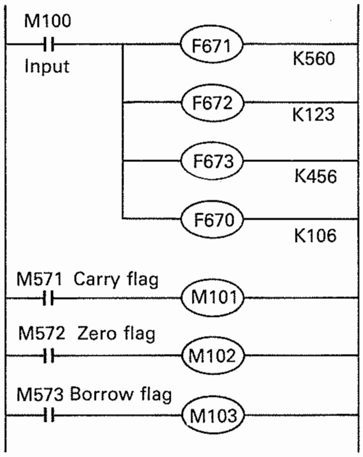
F670 input condition	Setting of F672	Comp. result	Error flag M570	Carry flag M571	Zero flag M572	Borrow flag M573
OFF	—	—	OFF	OFF	OFF	OFF
ON	Error	—	ON	OFF	OFF	OFF
ON	O.K.	K<S	OFF	ON	OFF	OFF
ON	O.K.	K=S	OFF	OFF	ON	OFF
ON	O.K.	K>S	OFF	OFF	OFF	ON

- For constant with decimal point against counter and register, such decimal point is ignored.
- Handling of constant for timer T650~657 (0.01-sec. timer)
  K12.3 → 12.3 sec.
  K123 → 1.23 sec.

F670 K106

COMPARISON OF ZONE FOR C CURRENT VALUE

Object element:
Counter (C060~C667)
3-digit decimal, 2 points



Designation of counter No. (C560)

Decimal constant (0~999) A

Decimal constant (0~999) B (B≧A)

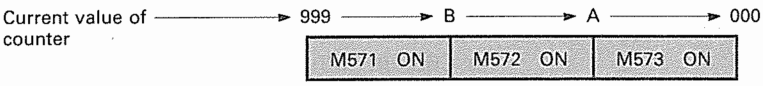
Current value of counter designated is compared with constants A, B for output to M571~M573.

ON: B < Current value

ON: B ≧ Current ≧ A

ON: Current value < A

- Compare is executed in case the input is turned on.
  In case there occurs any counter No. setting error, the error flag M570 is turned on, and no instruction is executed, and M570~M573 will be turned off.
- When the input is turned off, no processing will be executed.
  In this case, M570~M573 will be turned off.
- For setting value, B needs to be equal to or greater than A. If A is larger than B, A is compared with the current counter value.



Application example-10
Driving of load by timechart

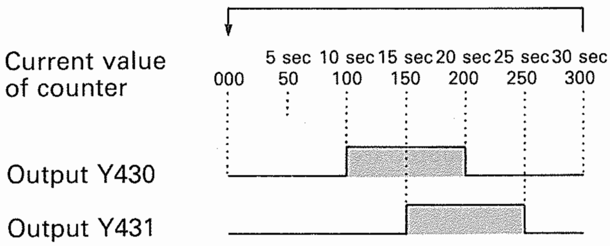


Fig. 5-15 shows a sequence diagram for the case when the multiple outputs Y430, Y431, etc, are operated in accordance with timechart as shown above.

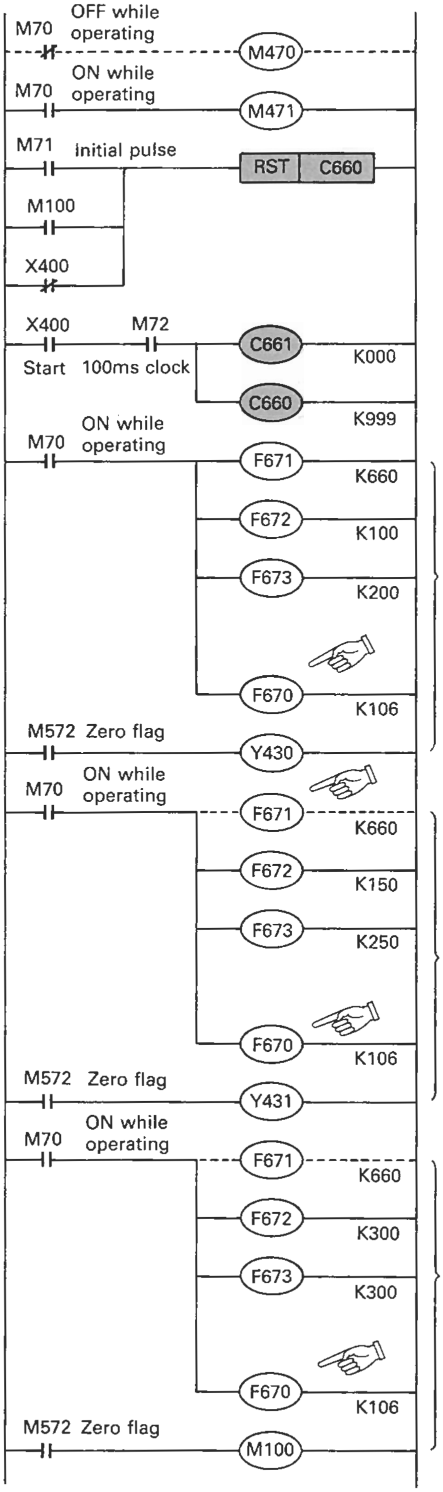


Fig. 5-15 Timechart output circuit

C660/C661 internal counter mode

C660/C661 up-count mode

C660/C661 counter reset
(RST C661 unnecessary)

Counting by upper 3-digit counter

Counting by lower 3-digit counter

Y430 is turned on when current Value of counter C660 is 100~200.

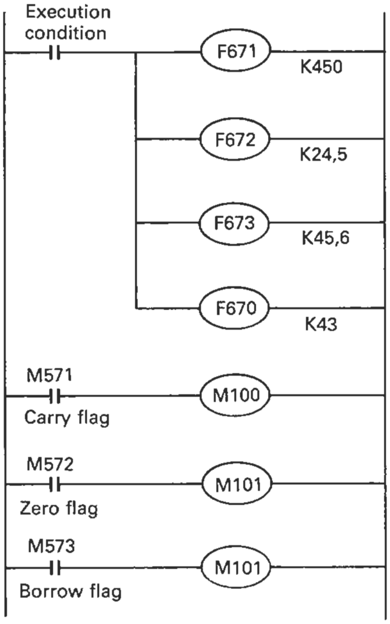
Since F671 has been K660, its program may be omitted.

Y431 is turned on when current value of counter C660 is 150~250.

When current value of counter reaches 300, M100 is actuated to reset counters C660/C661.

Object element: T, C, D700~777

F670 K43
COMPARISON OF ZONE FOR T, C, D



Setting of compare object element No.:

Decimal constant (0.0 ~ 999): A

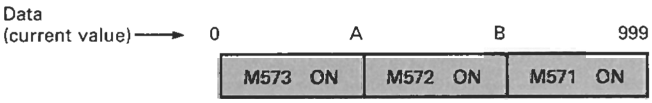
Decimal constant (0.0 ~ 999): B (A≦B)

Compare instruction

ON: B < Current value

ON: A ≦ Current valve ≦ B

ON: A > Current value

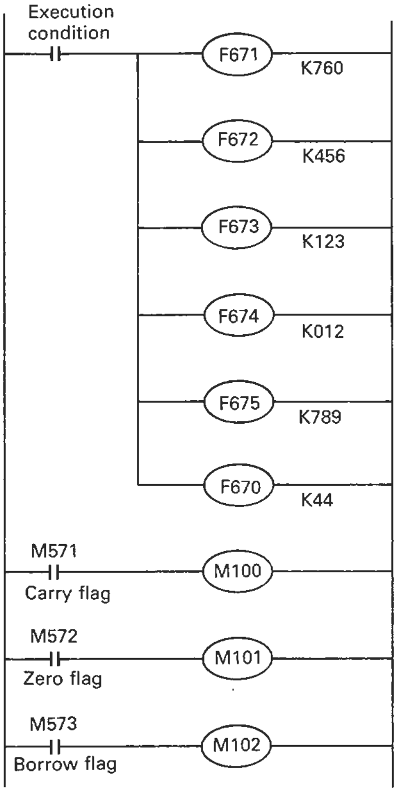


- Comparison is executed when execution condition is turned on, and no processing is executed when execution condition is turned off.
- In case error occurs in setting of element No., error flag M570 is turned on, and no processing is executed.
- The setting data is pre-conditioned that A is equal to or smaller than B. If A is greater than B, comparison is made only with A. (Either M571 or M573 is turned on in this case).
- Following table shows a list of operations for each flag;

F670 input condition	Setting of F671/F672 No.	Comp. result	Error flag M570	Carry flag M571	Zero flag M572	Borrow flag M573
OFF	—	—	OFF	OFF	OFF	OFF
ON	Error	—	ON	OFF	OFF	OFF
ON	O.K.	B< Current value	OFF	ON	OFF	OFF
ON	O.K.	A≦ Current value ≦B	OFF	OFF	ON	OFF
ON	O.K.	Current value <A	OFF	OFF	OFF	ON

F670 K44
COMPARISON OF ZONE FOR 6-DIGIT COUNTER AND DATA RESISTER

Object element: D700~D777, C



Designation of lower 3-digit side data register (counter) No. (upper 3 digits are Nos. following designated No.)

Lower 3 digits / Upper 3 digits } range Small side setting value 123456: A

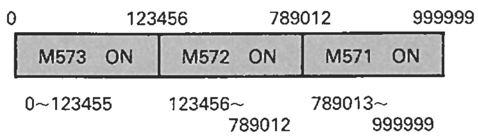
Lower 3 digits / Upper 3 digits } range Great side setting value 789012: B

Execution of zone compare

ON: B < Current value (D)

ON: A ≦ Current value ≦ B

ON: A > Current value



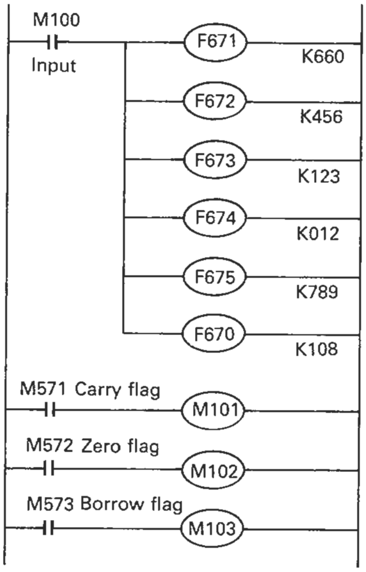
This is a an instruction which executes a zone-compare, taking a pair of two consecutive data registers/counters as 6-digit data, counter.
In the above example, six digits of M760 and M761 will become the object.

- Compare is executed when execution condition is turned on.
  If execution condition is turned off, no processing is executed, and M570~M573 are all turned off.
- Data register and counter Nos. at lower 3-digit side will necessarily be even number.
  If any odd number is set, error flag M570 is turned on, and no processing is executed.
- Setting data counter value is pre-conditioned that A is equal to or smaller than B.
  If A is greater than B, compare is executed only on A with data register/counter value.
- When there occurs any error in data counter No. setting, error flag M570 is turned on, and no processing is executed.
- The following table shows a lest of operations for each flag.

F670 input condition	Setting of data register/counter	Comp. result	Error flag	Carry flag	Zero flag	Borrow flag
OFF	—	—	OFF	OFF	OFF	OFF
ON	Error	—	ON	OFF	OFF	OFF
ON	O.K.	B < D	OFF	ON	OFF	OFF
ON	O.K.	A ≦D ≦B	OFF	OFF	ON	OFF
ON	O.K.	B < D	OFF	OFF	OFF	ON

F670 K108
COMPARISON OF ZONE FOR 6 DIGIT COUNTER

Object element:
Counter (C060~C667)
Decimal 6-digit constant, 2 points



Designation of lower 3-digit side counter No. (Even No.)

Upper 3-digit side counter will be of No., following the above No.

Lower 3-digit, Upper 3-digit } 6-digit setting (0~999999) A

Lower 3-digit, Upper 3-digit } 6-digit setting (0~999999) B

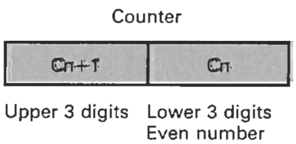
Current values of C660(lower 3 digit) and C661(upper 3 digit) are compared with constants A and B to output the comparison result.

ON: B < Current value

ON: B ≧ Current ≧ A

ON: Current value < A

- In case input is turned on, compare instruction is executed.
  In case there occurs any counter No. setting error or the counter designated is not of even No., the error flag M570 is turned on, and no instruction will be executed.
  In this case, M571~M573 will be turned off.
- No processing is executed in case the input is turned off and M570~M573 will be turned off.
- The setting value needs to be in the relationship of B ≧ A.
  If A is larger tha B, A is compared with the current counter value.



NOTE

How to program 6-digit counter

The zone compare instruction for 6-digit counter is effectively used for pair counters C660/C661.
Even in the case of general 3-digit subtracting counter, it is possible to made up a 6-digit subtracting counter(cascade counter) by executing the longitudinal connection in the following procedure,which can make use of this zone compare instruction.

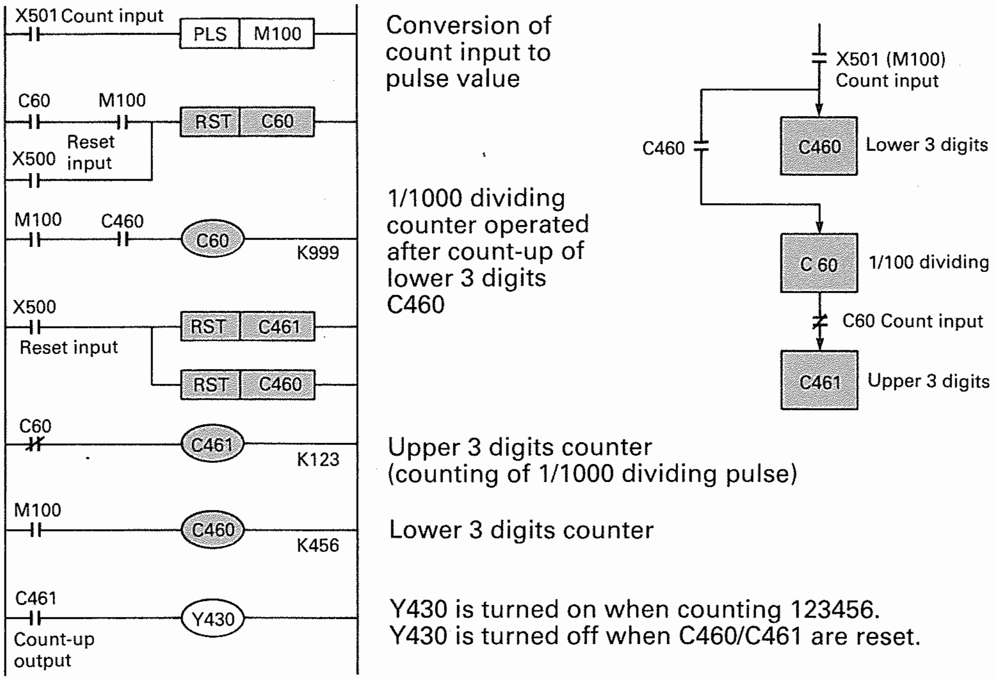


Fig.5-16 6-digit cascade down-counter

The pulse output M100 is used for dividing of 1/1000 against the maximum setting value 999 of dividing counter C60.
The reset circuit for counter C60 needs to be programmed in advance to the counting circuit.

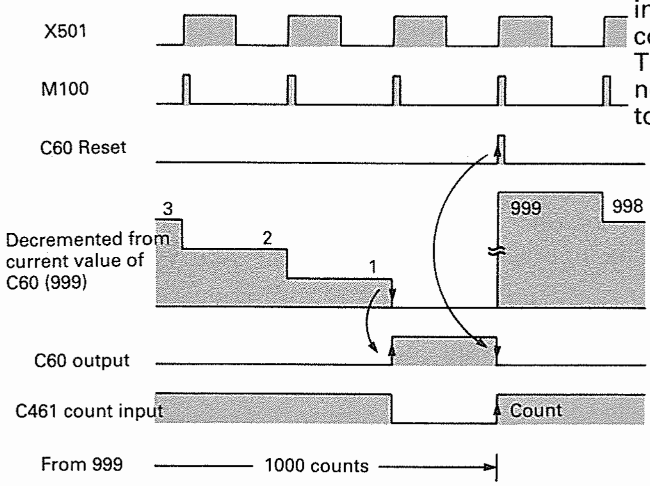
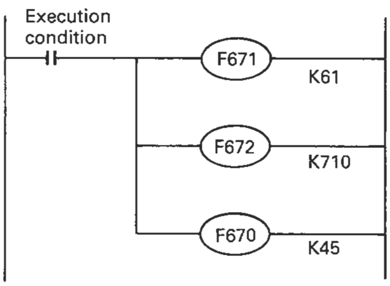


Fig. 5-17 Timechart

F670 K45

COMPARISON OF D, C WITH D, C

Object element: D700~777, C



Counter/data register No.: S_1

Counter/data register No.: S_2

Compare instruction

- When execution condition is turned on, compare is executed. When execution condition is turned off, no processing is executed.
- When there occurs any error in counter/data register No. setting, error flag is turned on, and no processing is executed.
- Following table shows a list of operation for each flag.

Operation of F670	Setting of F671, F672	Caluculation result	Error flag M570	Carry flag M571	Zero flag M572	Borrow flag M573
OFF	—	—	OFF	OFF	OFF	OFF
ON	Error	—	ON	OFF	OFF	OFF
ON	O.K.	$S_1>S_2$	OFF	ON	OFF	OFF
ON	O.K.	$S_1 = S_2$	OFF	OFF	ON	OFF
ON	O.K.	$S_1<S_2$	OFF	OFF	OFF	ON

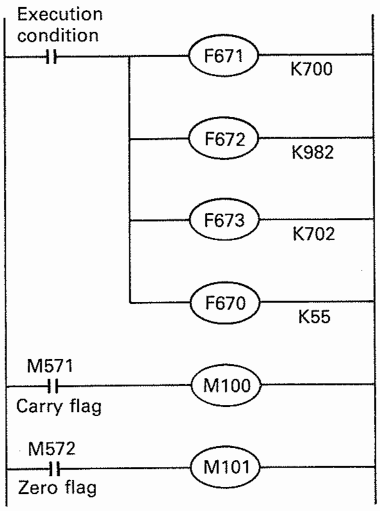
Note: Bear in mind that operation are reversed when compare is executed with constant K.
F671 = K
F672 = D, C

5-5	Arithmetic data operation instruction

F670 K55

ADDITION D+K

Object element: D700~777
BCD 3 digits



Operand No.: S_1

Constant: K

Calculation result storage register: D

Execution Instruction

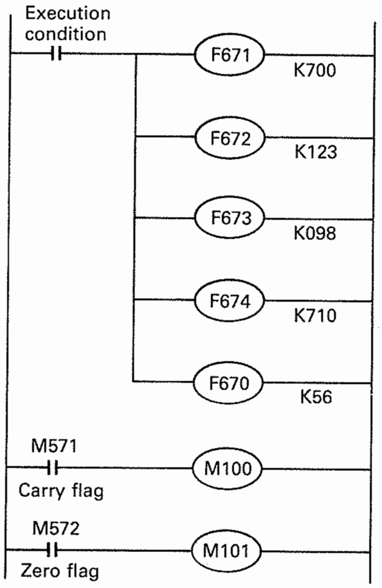
- When execution condition is turned on, the following calculation is executed;

BCD 3 digits		BCD 3 digits		Carry		BCD 3 digits		Carry		Zero flag
$\boxed{S_1}$	+	$\boxed{K}$	+	$\boxed{Cy}$	=	$\boxed{D}$	,	$\boxed{Cy}$	,	$\boxed{Z}$
D700~777		K000~999		M571		D700~777		M571		M572

- No processing is executed when execution condition is turned off.
  M570 is turned off.
- Bear in mind no calculation is executed when non-object element is set by operand register No. and calculation result register No.
  In this case, error flag M570 is turned on.
- When there occurs carry due to calculation, carry flag M571 is turned on.
- When result turns to be "0" due to calculation, zero flag M572 is turned on.
- M573 remains unchanged.

F670 K56
ADDITION D+K+Cy

Object element: D700~777
BCD 3 digits



Operand No.: S_1

6-digit constant (lower 3 digits)

6-digit constant (upper 3 digits)

Calculation result storage register No.

Execution Instruction

- Addition is executed when execution condition is turned on, and no processing is executed when execution condition is turned off.
- When execution condition is turned on, the following calculation is executed. M570 is turned off.

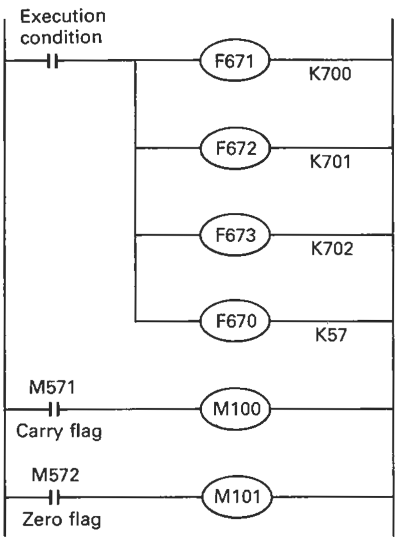
BCD 6 digits		BCD 6 digits		Carry		BCD 6 digits		Carry	Zero flag
$\boxed{S_{1+1}}\ \boxed{S_1}$	+	$\boxed{K_{+1}}\ \boxed{K}$	+	$\boxed{Cy}$	=	$\boxed{D_{+1}}\ \boxed{D}$	,	$\boxed{Cy}$	$\boxed{Z}$
S: D700~776		K000000~999999		M571		D: D700~776		M571	M572

- When there occurs setting error in No. of data register (operand register, calculation result storage register), error flag M570 is turned on and no processing is executed. Similarly when data register No. is set in odd number, error will occur.
- When calculation result exceeds 6 digits, carry flag M571 is turned on.
- In case calculation result is "0", zero flag M572 is turned on.
- M573 remains unchanged.

F670 K57

ADDITION D+D

Object element: D700~777
BCD 3 digits



Operand No.: S_1

Operand No.: S_2

Calculation result storage register No.: D

Execution Instruction

- When execution condition is turned on, the following calculation is executed;

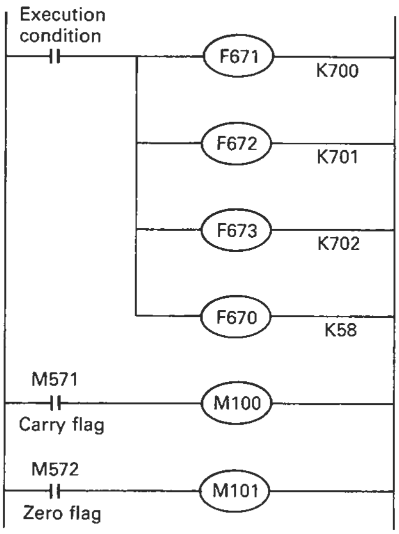
BCD 3 digits		BCD 3 digits		BCD 3 digits		Carry flag		Zero flag
S_1	+	S_2	=	D	,	Cy	,	Z
D700~777		D700~777		D700~777		M571		M572

- No processing is executed when execution condition is turned off. M570 is turned off.
- Bear in mind no calculation is executed when non-object element is set by operand register No. and calculation result register No. In this case, error flag M570 is turned on.
- When there occurs carry due to calculation, carry flag M571 is turned on.
- When result turns to be "0" due to calculation, zero flag M572 is turned on.
- M573 remains unchanged.

F670 K58

ADDITION D+D+Cy

Object element: D700~777
BCD 3 digits



Operand No.: S_1

Operand No.: S_2

Calculation result storage register No.: D

Execution Instruction

- Following calculation is executed when execution condition is turned on.

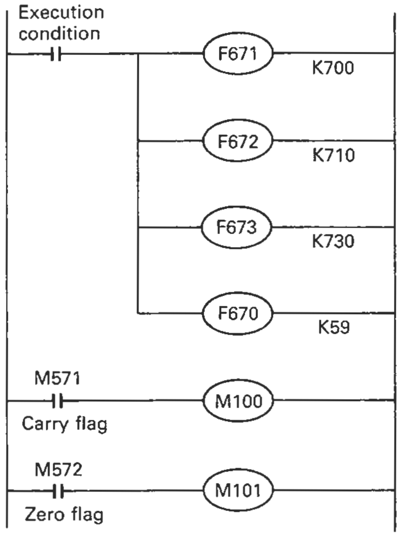
BCD 3 digits		BCD 3 digits		Carry		BCD 3 digits		Carry flag		Zero flag
S_1	+	S_2	+	Cy	=	D	,	Cy	,	Z
D700~777		D700~777		M571		D700~777		M571		M572

- No processing is executed when execution condition is turned off. ... M570 ... OFF
- When non-object element is set by operand register/calculation result storage register No., error flag M570 is turned on, thereby no calculation will be executed.
- If carry occurs due to calculation, carry flag M571 is turned on.
- In case the calculation result is "0", zero flag M572 is turned on.
- M573 remains unchanged in this case.

F670 K59

ADDITION D+D+Cy

Object element: D700~777
BCD 6 digits



Operand No.: S_1

Operand No.: S_2

Calculation result storage register
No.: D

Execution Instruction

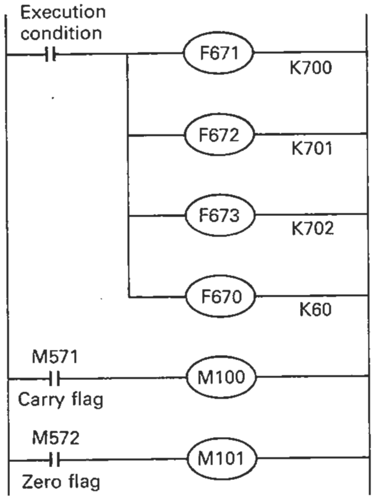
- Following calculation is executed when execution condition is turned on;

BCD 6 digits		BCD 6 digits		Carry		BCD 6 digits		Carry	Zero flag
S_{1+1} S_1	+	S_{2+1} S_2	+	Cy	=	D+1 D	,	Cy	Z
S: D700~776		S_2: D700~776		M571		D: D700~776		M571	M572

- No processing is executed when execution condition is turned off. ... M570 OFF
- When there occurs any setting error in register No. (operand, calculation result register), error flag M570 is turned on, and no processing will be executed.
  Error will also occur when data register No. is odd number.
- In case the calculation results exceeds 6 digits, carry flag M571 is turned on.
- If the calculation result is "0", zero flag is turned on.
- In this case, M573 remains unchanged.

F670 K60
ADDITION D+D

Object element: D700~777
Octal 3 digits



Operand No.: S_1

Operand No.: S_2

Calculation result storage register No.: D

Execution Instruction

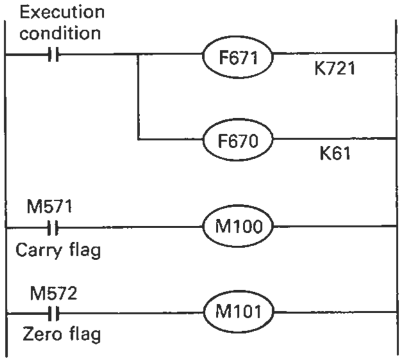
Condition is turned on. (Octal addition)

OCT 3 digits		OCT 3 digits		OCT 3 digits		Carry flag		Zero flag
$\boxed{S_1}$	+	$\boxed{S_2}$	=	$\boxed{D}$	,	$\boxed{Cy}$	,	$\boxed{Z}$
D700~777		D700~777		D700~777		M571		M572

- No processing is executed when execution condition is turned off. ... M570 OFF
- When operand, calculation result register Nos. are set for non-object element, error flag M570 is turned on, and no processing is executed.
- If contents in designated register is not octal, M570 is also turned on, and no processing will be executed.
- When calculation result exceeds 777, carry flag M571 is turned on.
- When calculation result is "0", zero flag M572 is turned on.
- In this case, M573 remains unchanged.

F670 K61
INCREMENT D+1

Object element: D700~777
BCD 3 digits



Data register No.

Increment execution

- When execution condition is turned on, the following calculation is executed;

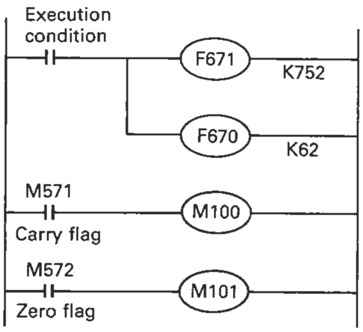
						Carry		Zero flag
(D721)	+	1	→	(D721)	,	$\boxed{Cy}$	,	$\boxed{Z}$
BCD 3 digits				BCD 3 digits		M571		M572

- No processing is executed when execution condition is turned off.
  M570 is turned off.
- In case error occurs in setting of data register No., error flag M570 is turned on, and no processing is executed.
- When there occurs carry due to increment, M571 is turned on.
  When result turns to be "0" due to increment, M572 is turned on.
- M573 remains unchanged.

F670 K62

INCREMENT D+1

Object element: D700~777
BCD 6 digits



Data register lower digit No. (even No.)

Increment execution

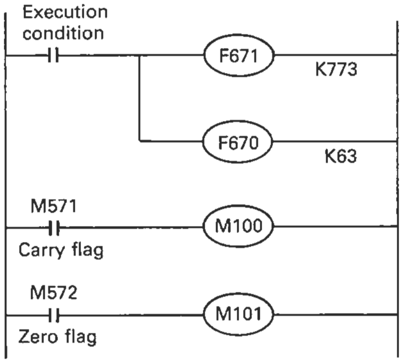
- Following calculation is executed when execution condition is turned on.

						Carry		Zero flag
(D753) (D752)	+	1	→	(D753) (D752)	,	$\boxed{Cy}$	,	$\boxed{Z}$
						M571		M572

- No processing is executed when execution condition is turned off. ... M570 OFF
- In case there occurs any error in data register No. setting, error flag M570 is turned on, and no processing is executed.
- When lower digit No. of data register is odd number, M570 is also turned on, and no processing is executed.
- When there occurs carry due to increment, M571 is turned on.
  When the result turns to be "0" due to increment, M572 is turned on.
- In this case, M573 remains unchanged.

F670 K63
INCREMENT D+1

Object element: D700~777
Octal 3 digits



Data register No.

Increment execution

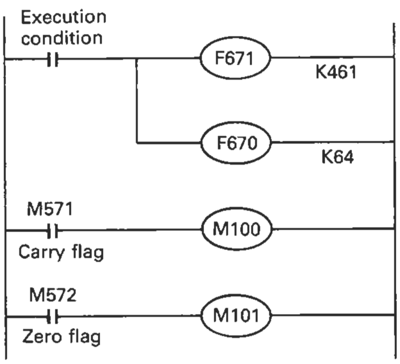
- Following calculation is executed when execution condition is turned on;

	Carry		Zero
(D773) + 1 → (D773) ,	Cy	,	Z
OCT	M571		M572

- No processing is executed when execution condition is turned off. (M570 OFF)
- When there occurs an error in data register No. setting, error flag M570 is turned on, and no processing is executed.
- If data in data register is not octal, M570 is turned on, and no processing is executed.
- When there occurs carry due to increment, M571 is turned on.
  When the result turns to be "0" due to increment, M572 is turned on.
- In this case, M573 remains unchanged.

F670 K64
INCREMENT C+1

Object element: C060~667
Counter current value
(BCD 3 digits)



Counter No.

Increment execution

- Following calculation is executed when execution condition is turned on;

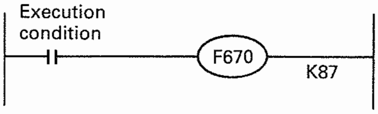
	Carry		Zero
(C461) + 1 → (C461) ,	Cy	,	Z
	M571		M572

- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs an error in counter No. setting, error flag M570 is turned on, and no processing is executed.
- When there occurs carry due to increment, M571 is turned on.
  When the result turns to be "0" due to increment, M572 is turned on.
- In this case, M573 remains unchanged.

F670 K87

FORMAT OF SUBTRACTION

Object element: D700~777



Flag set
(this instruction is established only by execution coil)

- When execution condition is turned on, flag for K87 is set.
  If it is turned off, the flag for K87 is reset.
  The operation in this case will be as follows;

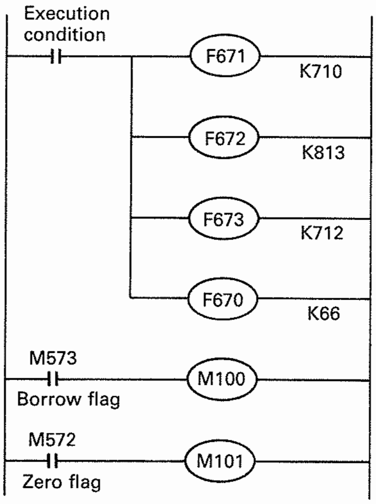
  F670 K87
  OFF .... Complement of 10
  Example: 123 − 456 = 667 & Borrow flag
  ON ..... Indication of absolute value
  Example: 123 − 456 = |−333|
  = 333 & Borrow flag

- Applicable applied instructions will include F670 K66, K67, K68, K69 and K70.
  In the case of K71, it is fixed to complement of 8.

F670 K66

SUBTRACTION D−K−Br

Object element: D700~777
BCD 3 digits



Operand No.: S_1

Constant (3 digits): K

Calculation result storage register No.: D

Execution Instruction

- Following calculation is executed when execution condition is turned on.

BCD 3 digits		BCD 3 digits		Borrow		BCD 3 digits	Borrow		Zero flag
$\boxed{S_1}$	−	$\boxed{K}$	−	$\boxed{Br}$	=	$\boxed{D}$	$\boxed{Br}$	,	$\boxed{Z}$
D700~777		K000~999		M573		D700~777	M573		M572

- When execution condition is turned off, no subtraction is executed. (M570 OFF)
- When there occurs an error in data register No. setting, error flag M570 is turned on, and no calculation is executed.
- When the calculation result turns to be negative, borrow flag M573 is turned on, and calculation result is indicated in complement or absolute value.

OUT F670 K87 ... OFF : Complement of 10

OUT K670 K87 ... ON : Absolute value

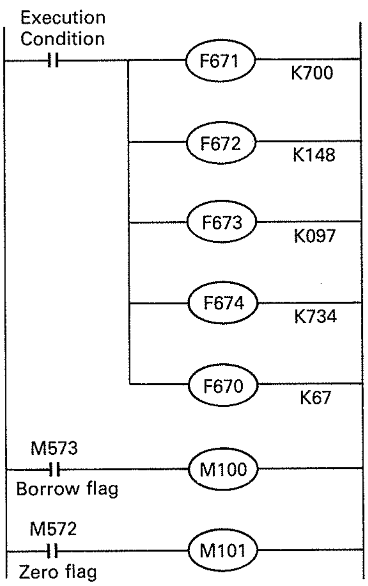
- Operation of borrow flag M573 is as shown in the following table;

	M573
S ≧ K + M573	OFF
S < K + M573	ON

- In case the calculation result turns to be "0", zero flag M572 is turned on.
- In this case, M571 remains unchanged.

F670 K67
SUBSTRACTION D−K−Br

Object element: D700~777
BCD 6 digits



Operand No.: S_1

6-digit constant (lower 3 digits): K

6-digit constant (upper 3 digits): K+1

Calculation result storage register No.

Execution Instruction

- Following calculation is executed when execution condition is turned on.

BCD 6 digits		BCD 6 digits		Borrow		BCD 6 digits		Borrow flag	Zero flag
$\boxed{S_{1+1}}\ \boxed{S_1}$	$-$	$\boxed{K_{+1}}\ \boxed{K}$	$-$	$\boxed{Br}$	$=$	$\boxed{D_{+1}}\ \boxed{D}$	,	$\boxed{Br}$	$\boxed{Z}$
S_1: D700~776		K000000~ 999999		M573		D: D700~776		M573	M572

- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs an error in data register No. setting, error flag M570 is turned on, and no calculation is executed.
  An error also occurs when data register No. is odd number.
- If the calculation result turns to be negative, borrow flag is turned on, and calculation result is indicated in complement or absolute value.

  OUT F670 K87 .... OFF : Complement of 10

  OUT F670 K87 .... ON : Absolute value

- Operation of borrow flag M573 is as shown in the following table;

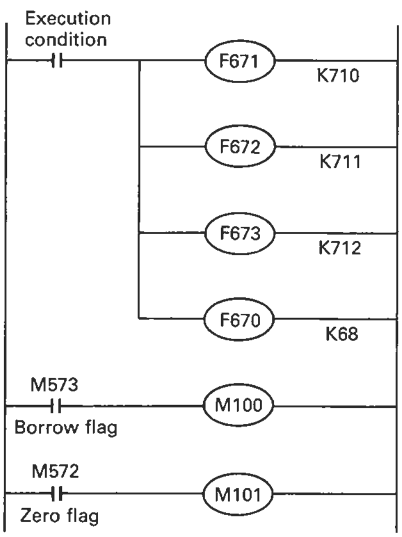
	M573
$S \geqq K_{+1}$, K + M573	OFF
$S < K_{+1}$, K + M573	ON

- When calculation restult turns to be "0", zero flag M572 is turned on.
- In this case, M571 remains unchanged.

F670 K68

SUBSTRACTION D−D

Object element: D700~777
BCD 3 digits



Operand No.: S_1

Operand No.: S_2

Calculation result storage register No.: D

Execution Instruction

- Following calculation is executed when execution condition is turned on.

BCD 3 digits		BCD 3 digits		BCD 3 digits		Borrow flag		Zero flag
$\boxed{S_1}$	−	$\boxed{S_2}$	=	$\boxed{D}$	,	$\boxed{Br}$	,	$\boxed{Z}$
D700~777		D700~777		D700~777		M573		M572

- When execution condition is turned off, no calculation is executed. (M570 OFF)
- When there occurs an error in data register No. setting, error flag M570 is turned on, and no calculation is executed.
- When calculation result turns to be negative, borrow flag M573 is turned on, and calculation result is indicated in complement or absolute value.

OUT F670 K87 ... OFF : Complement of 10

OUT F670 K87 ... ON : Absolute value

- Operation of borrow flag M573 is as shown in the following table;

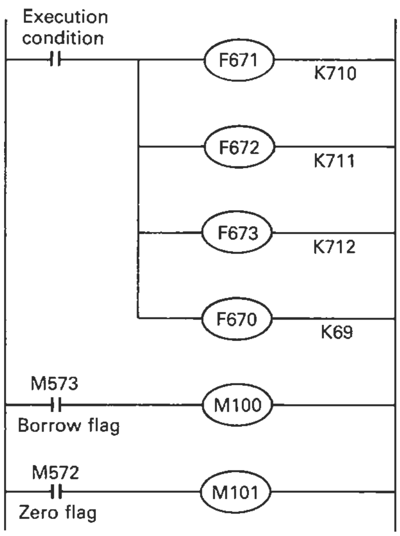
	M573
$S_1 \geqq S_2$	OFF
$S_1 < S_2$	ON

- When calculation result turns to be "0", zero flag is turned on.
- In this case, M571 remains unchanged.

F670 K69

SUBTRUCTION D − D − Br

Object element: D700~777
BCD 3 digits



Operand No.: S_1

Operand No.: S_2

Calculation result
storage register No.: D

Execution Instruction

- Following calculation is executed when execution condition is turned on;

BCD 3 digits		BCD 3 digits		Borrow		BCD 3 digits		Borrow		Zero flag
$\boxed{S_1}$	−	$\boxed{S_2}$	−	$\boxed{Br}$	=	$\boxed{D}$	,	$\boxed{Br}$	,	$\boxed{Z}$
D700~777		D700~777		K573		D700~777		M573		M572

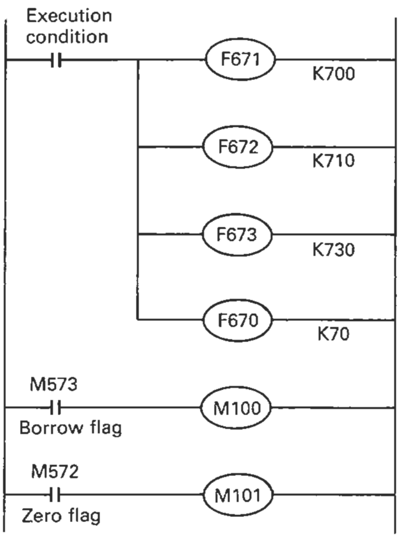
- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs an error in data register No. setting, error flag M570 is turned on, and no calculation is executed.
- When calculation result turns to be negative, calculation result is indicated in complement or absolute value.
  OUT F670 K87 ... OFF : Complement of 10
  OUT F670 K87 ... ON : Absolute value
- Operation of borrow flag M573 is as shown in the following table;

	M573
$S_1 \geqq S_2 + M573$	OFF
$S_1 < S_2 + M573$	ON

- When calculation result turns to be "0", zero flag M572 is turned on.
- In this case, M571 remains unchanged.

F670 K70
SUBTRACTION D − D − Br

Object element: D700~777
BCD 6 digits



Operand No.: S_1

Operand No.: S_2

Calculation result storage register No.: D

Execution Instruction

- Following calculation is executed when execution condition is turned on;

BCD 6 digits			BCD 6 digits			Borrow		BCD 6 digits			Borrow flag		Zero flag
S_{1+1}	S_1	−	S_{2+1}	S_2	−	Br	=	D_{+1}	D	,	Br	,	Z
S_1:700~776			S_2:D700~776			M573		D:D700~776			M573		M572

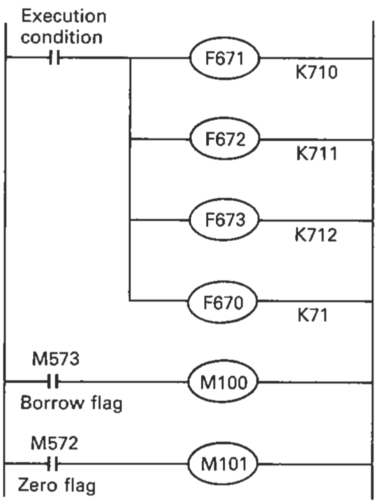
- When executed condition is turned off, no calculation is executed.
  When there occurs an error in data register No. setting, error flag M571 is turned on.
  Error occurs when data register No. is odd number.
- When calculation result turns to be negative, borrow flag M573 is turned on, and calculation result is indicated in complement or absolute value.
  OUT F670 K87 ... OFF : Complement of 10
  OUT F670 K87 ... ON : Absolute value
- Operation of borrow flag M573 is as shown in the following table;

	M573
$S_1 \geqq S_2 + M573$	OFF
$S_1 < S_2 + M573$	ON

- When calculation result turns to be negative, zero flag M572 is turned on.
- In this case, M571 remains unchanged.

F670 K71
SUBTRACTION D − D

Object element: D700~777
OCTAL 3 digits



Operand No.: S_1

Operand No.: S_2

Calculation result storage register No.: D

Execution Instruction

- Following calculation is executed when execution condition is turned on;

OCT 3 digits		OCT 3 digits		OCT 3 digits		Borrow flag		Zero flag
S	−	S	=	D	,	Br	,	Z
D700~777		D700~777		D700~777		M573		M572

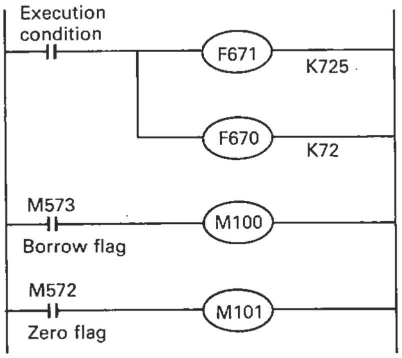
- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs any error in data register No. setting, error flag M570 is turned on, and no calculation is executed.
- When data in the data register is not octal, M570 is also turned on, and no calculation executed.
- When calculation result turns to be negative, borrow flag M573 is turned on, and calculation result is indicated in complement of 8.

	M573
$S_1 \geqq S_2$	OFF
$S_1 < S_2$	ON

F670 K72

DECREMENT D − 1

Object element: D700~777
BCD 3 digits



Data register No.

Decrement execution

- Following calculation is executed when execution condition is turned on;

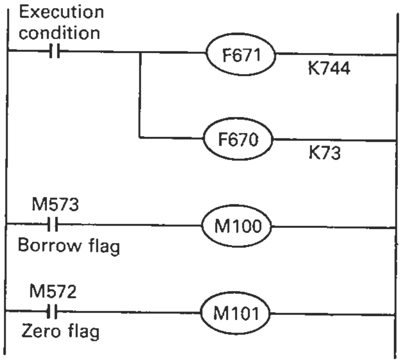
		Borrow flag		Zero flag
(D725) − 1 → (D725)	,	Br	,	Z
		M573		M572

- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs any error in data register No. setting, error flag M570 is turned on, and no calculation is executed.
- If borrow occurs due to decrement, M573 is turned on, and calculation result is "999".
- When it appears to be "0" due to increment, M572 is turned on.

F670 K73

DECREMENT D − 1

Object element: D700~777
BCD 6 digits



Data register lower digit No.

Decrement execution

- Following calculation is executed when execution condition is turned on;

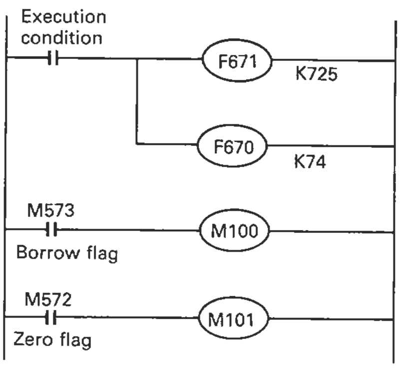
	Borrow flag	Zero flag
(D745) (D744) − 1 → (D745) (D744) ,	Br ,	Z
	M573	M572

- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs any error in data register No. setting, error flag M570 is turned on, and no calculation is executed.
- If the lower digit No. of data register is set to "777", M570 is turned on, and no calculation is executed.
- When there occurs borrow due to decrement, M573 is turned on, and the result appears to be "999999".
- When the result turns to be "0" due to decrement, M572 is turned on.

F670 K74

DECREMENT D − 1

Object element: D700~777
OCTAL 3 digits



Data register No.

Decrement execution

- Following calculation is executed when execution condition is turned on;

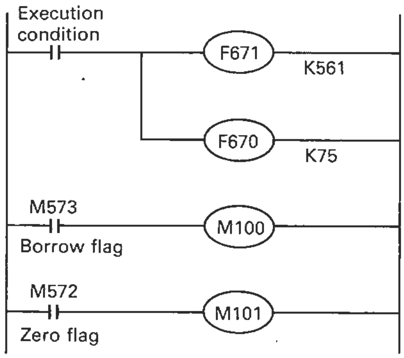
OCT (D725) − 1 → (D725) , Br , Z

Borrow flag: Br (M573)
Zero flag: Z (M572)

- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs any error in data register No. setting, error flag M570 is turned on, and no calculation is executed.
- If the data in data register is not octal, M570 is turned on, and no calculation is executed.
- When there occurs borrow due to decrement, M573 is turned on, and the result appears to be "777".
  When the result appears to be "0" due to decrement, M572 is turned on.

F670 K75
DECREMENT C − 1

Object element: C060~667
Counter current value
BCD 3 digits



Counter No.

Decrement execution

- Following calculation is executed when execution condition is turned on;

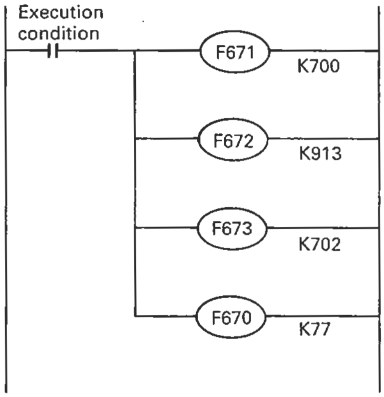
(C561) − 1 → (C561), Br , Z

Borrow flag: Br (M573)
Zero flag: Z (M572)

- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs an error in counter No. setting, error flag M570 is turned on, and no calculation is executed.
- When borrow occurs due to decrement, M573 is turned on, and the result appears to be "999".
  When the result appears to be "0" due to decrement, M572 is turned on.

F670 K77
MULTIPLICATION D×K

Object element: D700~777
BCD 3 digits



Operand No.: S

Constant: K

Calculation result
storage register No.: D

Execution Instruction

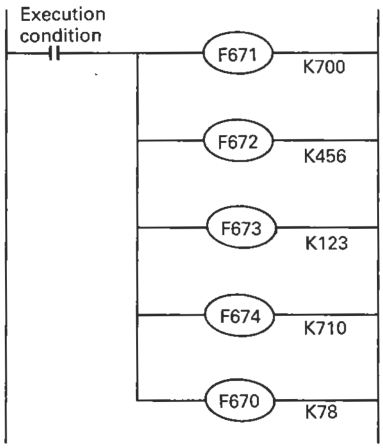
- Following calculation is executed when execution condition is turned on;

BCD 3 digits		BCD 3 digits		BCD 6 digits (LSB)
S	×	K	=	D_{+1} D
D700~777		K000~999		D:D700~776

- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs in data register No. setting, error flag M570 is turned on, and no calculation is executed.

F670 K78
MULTIPLICATION D×K

Object element: D700~777
BCD 6 digits



Operand No.: S_1

6-digit constant
(lower 3 digits): K_L

6-digit constant
(upper 3 digits): K_H

Calculation result
storage register No.: D

Execution Instruction

Example
(D701) (D700) × 123456 = (D713) (D712) (D711) (D710)

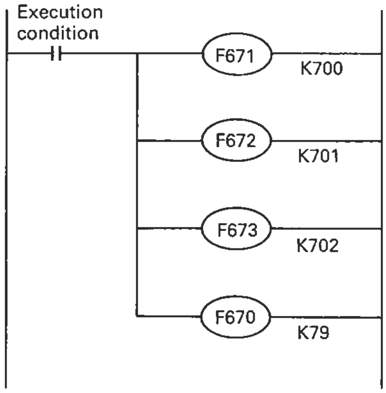
- Following calculation is executed when execution condition is turned on;

BCD 6 digits	BCD 6 digits	BCD 12 digits
$\boxed{S_{1+1}}\ \boxed{S_1}$	$\times\ \boxed{K_H}\ \boxed{K_L}$	$=\ \boxed{D_{+3}}\ \boxed{D_{+2}}\ \boxed{D_{+1}}\ \boxed{D}$
S_1:D700~776	K0~999999	D:D700~774

- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs an error in data register No. setting, error flag is turned on, and no calculation is executed.
- Be sure to set even number for data register No.
  If odd number is designated, M570 is turned on, and no processing is executed.

F670 K79
MULTIPLICATION D×D

Object element: D700~777
BCD 3 digits



Operand No.: S_1

Operand No.: S_2

Calculation result: D

Execution Instruction

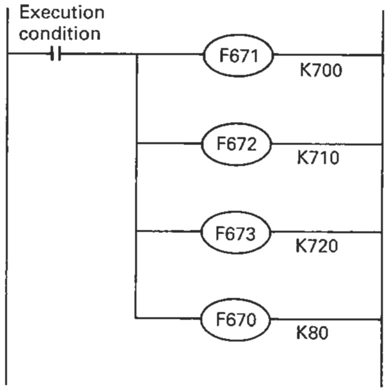
- Following calculation is executed when execution condition is turned on.

BCD 3 digits	BCD 3 digits	BCD 6 digits
$\boxed{S_1}$	$\times\ \boxed{S_2}$	$=\ \boxed{D_{+1}}\ \boxed{D}$
D700~777	D700~777	D:D700~776

- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs any error in data register No. setting, error flag M570 is turned on, and no calculation is executed.

F670 K80
MULTIPLICATION D×D

Object element: D700~777
BCD 6 digits



Operand No.: S_1

Operand No.: S_2

Calculation result storage register No.: D

Execution Instruction

Example
(D701) (D700) × (D711) (D710) → (D723) (D722) (D721) (D720)

- Following calculation is executed when execution condition is turned on;

BCD 6 digits $\quad$ BCD 6 digits $\quad$ BCD 12 digits

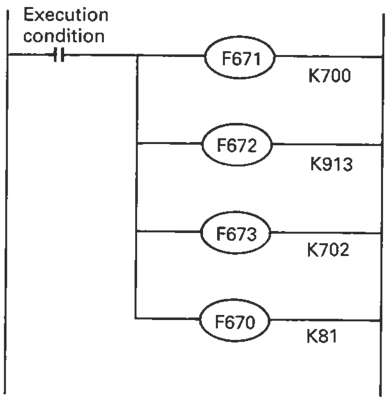
$\boxed{S_{1+1}}\ \boxed{S_1} \times \boxed{S_{2+1}}\ \boxed{S_2} = \boxed{D_{+3}}\ \boxed{D_{+2}}\ \boxed{D_{+1}}\ \boxed{D}$

S_1:D700~776 $\quad$ S_2:D700~776 $\quad$ D:D700~774

- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs an error in data register No. setting, error flag M570 is turned on, and no calculation is executed.
- Be sure to set even number for data register No.
  If odd number is set, M570 is turned on, and no processing is executed.

F670 K81
DIVIDE D/K

Object element: D700~777
BCD 3 digits



Operand No.: S

Constant: K

Calculation result storage register No.: D

Execution Instruction

Example
(D700) ÷ 913 → (D702) $\quad$ Remainder ... (D703)

• When execution condition is turned on, the following calculation is executed;

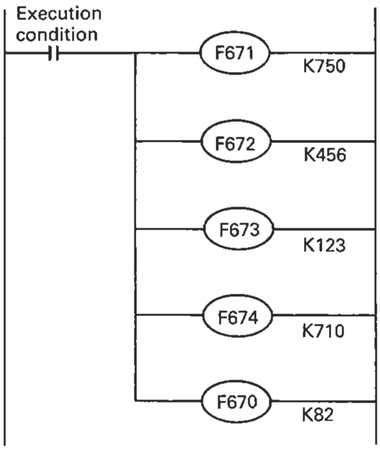
BCD 3 digits		BCD 3 digits		BCD 3 digits		BCD 3 digits
$\boxed{S}$	÷	$\boxed{K}$	=	$\boxed{D}$	Remainder	$\boxed{D_{+1}}$
D700~777		K1~999		D700~776		D701~777

• When execution condition is turned off, no processing is executed. (M570 OFF)
• When there occurs an error in data register No. setting, error flag M570 is turned on, and no calculation is executed.
• When the constant K is "0", error flag M570 is turned on, and no calculation is executed.

F670 K82

DIVIDE D/K

Object element: D700~777
BCD 6 digits



Operand No.: S_1

6-digit constant (lower 3 digits): K_L

6-digit constant (upper 3 digits): K_H

Calculation result storage register No.: D

Execution Instruction

Example

(D701) (D700) ÷ 123456 = (D711) (D710) Remainder (713) (712)

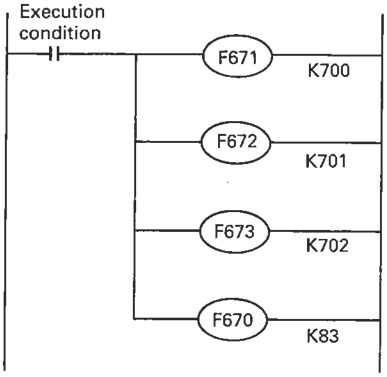
• Following calculation is executed when execution condition is turned on;

BCD 6 digits			BCD 6 digits			BCD 6 digits		Remainder BCD 6 digits	
$\boxed{S_{1+1}}$	$\boxed{S_1}$	÷	$\boxed{K_H}$	$\boxed{K_L}$	=	$\boxed{D_{+1}}$	$\boxed{D}$	$\boxed{D_{+3}}$	$\boxed{D_{+2}}$
S1:D700~776			K1~999999			D:D700~774		D2:D702~776	

• When execution condition is turned off, no processing is executed. (M570 OFF)
• When there occurs an error in data register No. setting, error flag M570 is turned on, and no processing is executed.
• Be sure to set even number for data register No.
If odd number is set M570 is turned on, and no processing is executed.
• When both constants K_L and K_H are "0", error flag M570 is turned on, and no calculation is executed.

F670 K83
DIVIDE D/D

Object element: D700~777
BCD 3 digits



Operand No.: S_1

Operand No.: S_2

Calculation result storage register No.: D

Execution Instruction

Example
(D700) ÷ (D701) → (D702)   Remainder (D703)

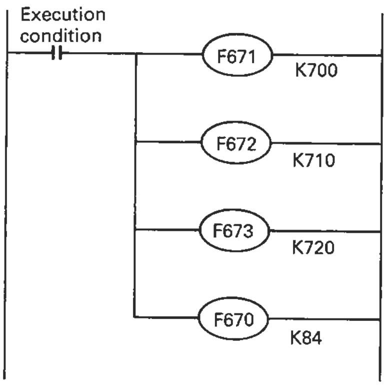
- Following calculation is executed when execution condition is turned on;

BCD 3 digits		BCD 3 digits		BCD 3 digits		BCD 3 digits
S_1	÷	S_2	=	D	Remainder	D_{+1}
D700~777		D700~777		D700~776		D701~777

- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs an error in data register No. setting, error flag M570 is turned on, and no calculation is executed.
- When data in operand data register(S_2) is "0", M570 is turned on, and no calculation is executed.

F670 K84
DIVIDE D/D

Object element: D700~777
BCD 6 digits



Operand No.: S_1

Operand No.: S_2

Calculation result storage register: D

Execution Instruction

Example
(D701) (D700) ÷ (D711) (710) = (D721) (D720), Remainder (D723) (D722)

- Following calculation is executed when execution condition is turned on;

BCD 6 digits		BCD 6 digits			BCD 6 digits		BCD 6 digits
$\boxed{S_{1+1}}\ \boxed{S_1}$	$\div$	$\boxed{S_{2+1}}\ \boxed{S_2}$	$=$	$\boxed{D_{+1}}\ \boxed{D}$	Remainder	$\boxed{D_{+3}}\ \boxed{D_{+2}}$	
S_1:D700~776		S_2:D700~776		D:700~774		D_{+2}:D702~776	

- When execution condition is turned off, no processing is executed. (M570 OFF)
- When there occurs an error in data register No. setting, error flag M570 is turned on, and no calculation is executed.
- Be sure to set even number for data register No.
  It odd number is set, M570 is turned on, and no calculation is executed.
- When both operands S_2 and S_{2+1} are "0", M570 is turned on, and no calculation is executed.

5-6 Auto re-load (AUTO-RELOAD) of pair counter

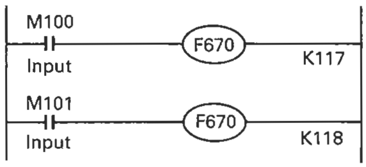
It is possible to execute the automatic re-loading for the pair counters C660/C660 by the use of functional instruction F670 K117, K118.
The auto reload function means that the current counter value is compared with the compare data set separately, and the counter is reset automatically when these values are consistent, and the setting value (value of K following OUT C) is preset to the current value register.

> F670 K117
>
> Transfer of compare data for auto re-load

Object elements;
Data source: M240~M253
M260~M273
Data transfer destination:
Compare data register for C660/C661 (D756/D757)

> F670 K118
>
> Auto reload valid

Object elements:
C660/C661 pair counter



Transfer of compare data for auto re-load

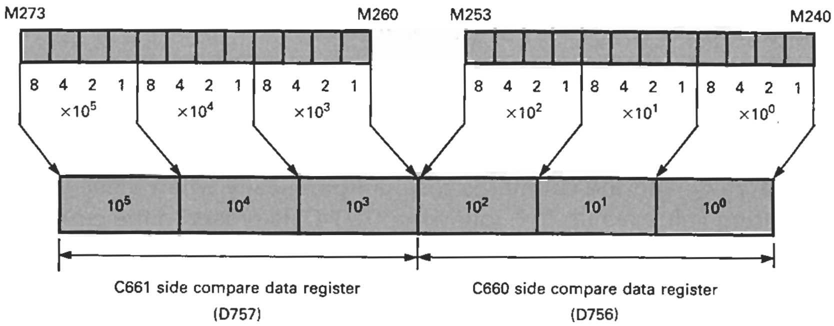
Auto re-load valid

K118

- In case input is turned on, the auto re-load function becomes valid, and in case it is turned off, the normal counter mode will be turned on.
- The program is not necessary in case the auto re-load function is not used.

K117

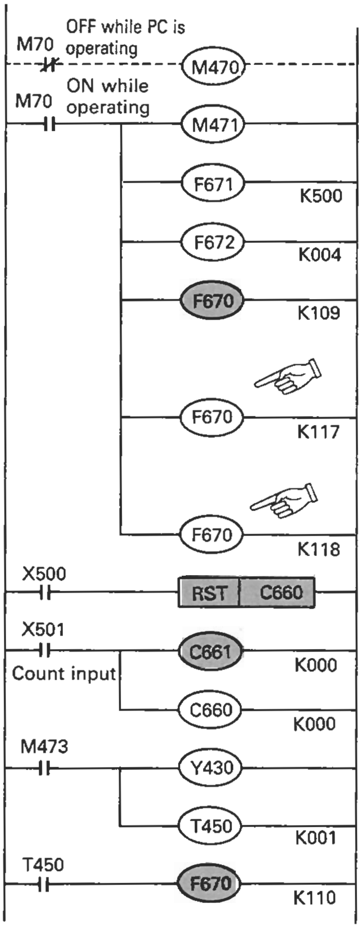
- When input is turned on, the data of M240~M253 and M260~M273 will be transferred to the compare data register (D756, D757).
  In case the source data is not of BCD data, the error flag M570 is turned on, by which the instruction will not be executed.
  (Compare data register contents will be held in the event of power failure).
- It is possible to monitor the compare data by monitoring these data registers D756 and D757 in case of F_1-20M~60M.
  However, type F_1-12M is not applicable to such monitoring.
  For the types F_1-20~60M, it is possible to write the data to the data registers D756 and D757 by the use of transfer instruction (for instance, F670 K33) without the use of F670 K117.
- In case input is turned off, the function will become invalid, by which the error flag M570 is set off.



In case the current counter value becomes consistent with the compare register value, the setting value (value of K following OUT C660, OUT C661) is transferred to the current value register, and the flag M473 is operated at the same time.

Circuit example-1

Auto reload of pair counter



Setting of C660/C661 to internal counting mode

Setting of C660/C661 to up-count mode

Transfer of 500 to M240~M253

Transfer of 004 to M260~M273 (See page-95).

Transfer of 4500 to C660/C661 comparison data register

Auto re-load mode

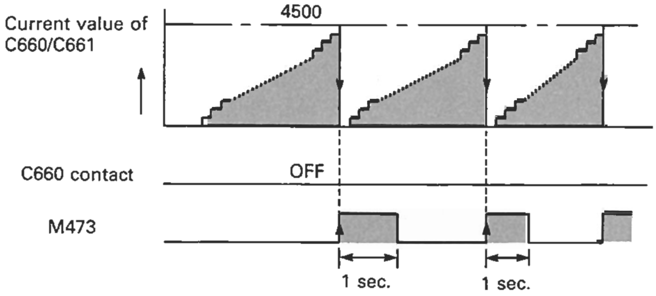
Counter reset (Output contact C660 is turned off and counter setting value is reset to "0").

Upper 3-digit counter (programmed first, K=0~999)
Lower 3-digit counter (programmed later, K=0~999)
Initial value

M473 is turned on when auto re-load execute (current counter value = 4500).

Resetting of M473 upon elapse of 1 sec. (See page-82).

Fig. 5-18 Auto reload circuit example



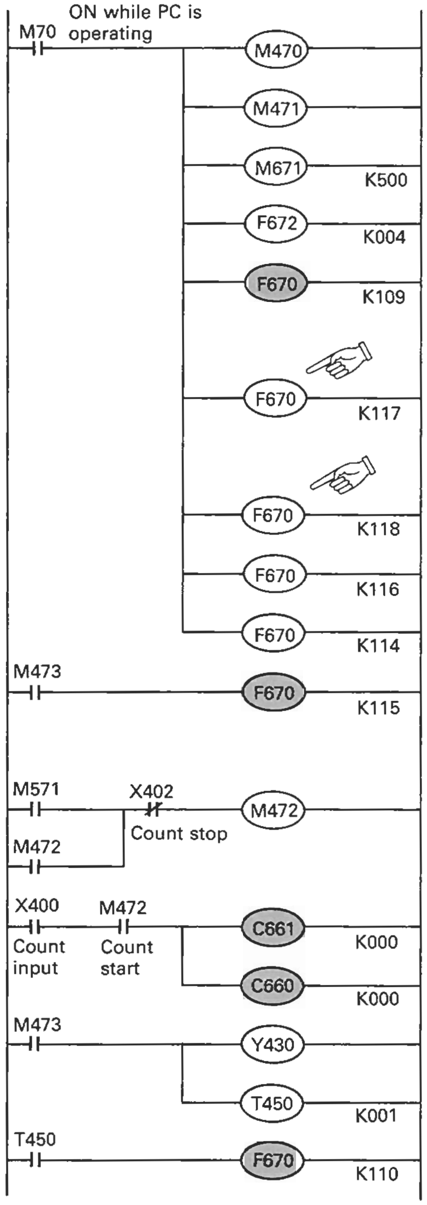
When the current value of pair counters C660/C661 reaches "4500", the auto reload output M473 is output.
In this case, the counters C660/C661 are automatically reset, by which the current value of counter is preset to initial value ("0" in the case of this example).

NOTE

The auto reload function is used effectively especially when the pair counters C660/C661 are used under the high-speed counter mode (M470 = ON) to be described further on, in which the setting value can be preset quickly in the case of coincidence when compared without any influence by the execution cycle.

Circuit example-2

Auto reload of high-speed counter



Setting of C660/C661 to high-counter mode

Setting of C660/C661 to up-count mode

Transfer of 500 to M240~M253

Transfer of 004 to M260~M273 (See page-95).

Transfer of 4500 to C660/C661 cpmparison data register

Auto re-load mode

Counter reset prohibit by X401 (See page-83.)

X401 rising detection (See page-77.)

M473 is turned on when current counter value is "4500". In this case, X401 rising detection flag M571 is reset. (See page-77).

Holding of M472 (counting start) operation after rising of X401

Upper 3-digit setting (K=0~999)

Lower 3-digit setting (K=0~999)

Initial value

M473 is turned on when current counter value is "4500".

Resetting of M473 (occurrence of auto re-load) upon elapse of 1 sec. (See page-82.)

Fig. 5-19 Auto reload (high-speed counter)

According to the program shown in Fig. 5-19, the counter starts counting operation (input: X400) when the input X401 is turned on.

When the value reaches the compare value of 4500, M473 is actuated, by which the output Y430 is operated and is reset upon elapse of one second.

In case the sheet rolled out continuously is stamped at the constant interval, the input X400 will appear to be the sheet feed amount detection pulse, while Y430 to be the output for stamp. It is an example of applications.

5-7 Direct Output instruction for high-speed counter

The high-speed counters C660/C661 can count the number of ON/OFF of input X400 at high speed (max. 2kHz).
These counters can be interrupted/reset by the input X401.
In addition, it is possible to preset the setting value by the auto reload (function described previously) synchronously with the counting operation.
As described previously, the counters can be operated at high speed regardless of execution cycle, however, the repsonse lagging may occur, if the output of compare result to the external unit is depending upon the general cyclic execution.
To settle such problem, the designated output is interrupted for direct output, which is the functional instruction to be described further on.
With the functional instruction, a maximum of four points direct outputs can be set at the same time.
Bear in mind that the maximum counting frequency appears to be 1.5kHz when this instruction is used.

F670 K119

Setting of high-speed output table

Object elements:
Register for table
(held in the event of power failure)

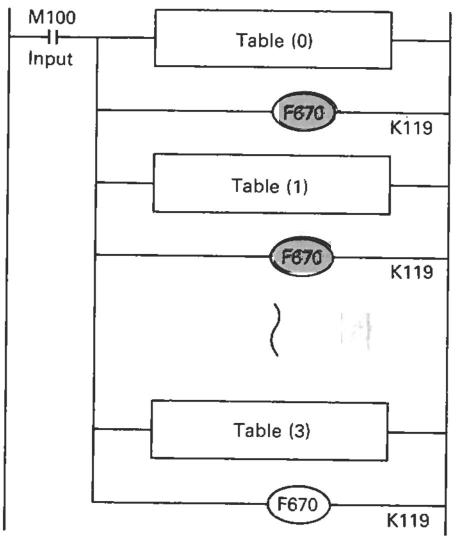


Table contents

#1 M240~M253 Lower 3-digit BCD compare data

#2 M260~M273 Upper 3-dight BCD compare data

#3 M254, M255 Table No. (0~3)

M255	M254	Table Nos.
OFF	OFF	0
OFF	ON	1
ON	OFF	2
ON	ON	3

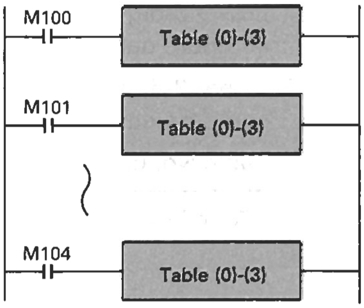
- Set the data to M240~M277 in table in advance.
- When input is turned on, the above data is transferred and set to table register.
  In case M240~M273 are other than BCD data, the error flag M570 is turned on, by which no processing will be executed.
- In case input is turned off, no processing will be executed and M570 is turned off.

#4 M274~M276 Output relay No.

M276	M275	M274	Output relay
OFF	OFF	OFF	Y430
OFF	OFF	ON	431
OFF	ON	OFF	432
OFF	ON	ON	433
ON	OFF	OFF	434
ON	OFF	ON	435
ON	ON	OFF	436
ON	ON	ON	437

#5 M277 output format
ON ... Set output
OFF ... Reset output

Application example-11
Multiple setting of table



If it is programmed that only one contact of M100~104 is turned on, it is possible to increase the high-speed output table to any quantity as shown at left figure.

In the type F_1-20~60, these table are allotted to data register D760~D777, it is possible to monitor the setting data by monitoring these register.
Type F_1-12M is not applicable to such monitoring.

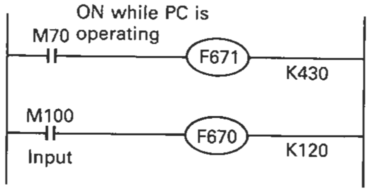
	Table(0)	Table(1)	Table(2)	Data register	
Lower 3-dight compare data	D760	D764	D770	D774	← $10^2, 10^1, 10^0$ data
Upper 3-dight compare data	D761	D765	D771	D775	← 10^5, 10^4, 10^3 data
Output relay No.	D762	D766	D772	D776	← Data 430~437
Output format	D763	D767	D773	D777	← Data-0=Reset Data-1=Set output

In addition to F670 K119 instruction, it is also possible to set these table by the use of transfer instruction to data resister (for instance, F670 K33).

F670 K120

Prohibit of independent high-speed output

Object elements: Y430~Y437



Setting of output prohibit relay No.

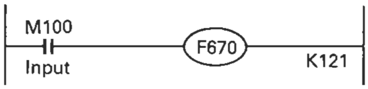
Prohibit of independent high-speed output

- In case input is turned on, the high-speed output of designated element is prohibited.
  The element designating coil F671 is driven all the time by M70.
  The 10^1 and 10^2 digit values of K-No. in F671 are ignored and regarded as 43□.
  0~7 will be valid for □, and 8 is regarded as 0, and 9 as 1 respectively.
- In case input is turned off, high-speed output is possible for designated element.
  Where no prohibition is necessary, both F671 and F670 need not to be programmed.

F670 K121

High-speed output simultaneous permit

Object elements: Y430~Y437

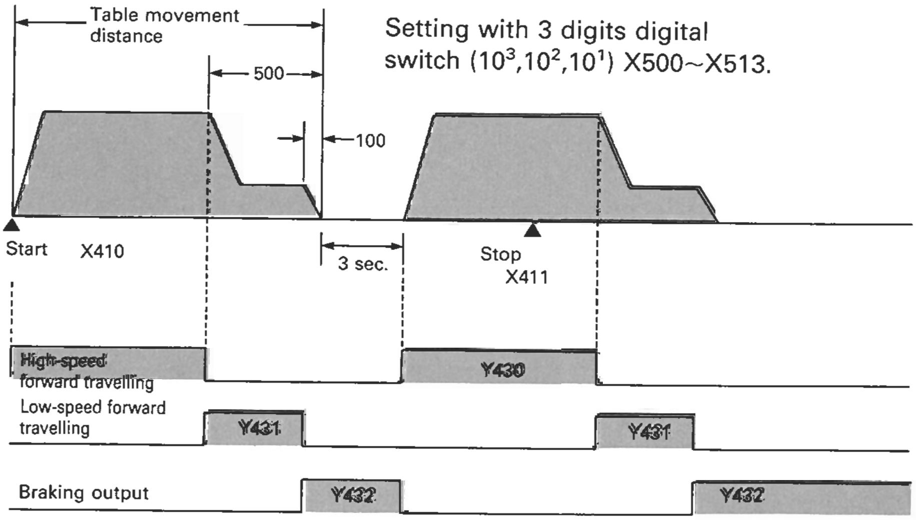


High-speed output simultaneous permit

- In case input is turned on, the high-speed output becomes available, which is set at each table.
  In this case, bear in mind that the output is prohibited for the output in which the independent prohibit instruction is operating.
- In case input is turned off, all outputs set at each table (0) ~ (3) will be prohibited.

Application example-12

Simple positioning control



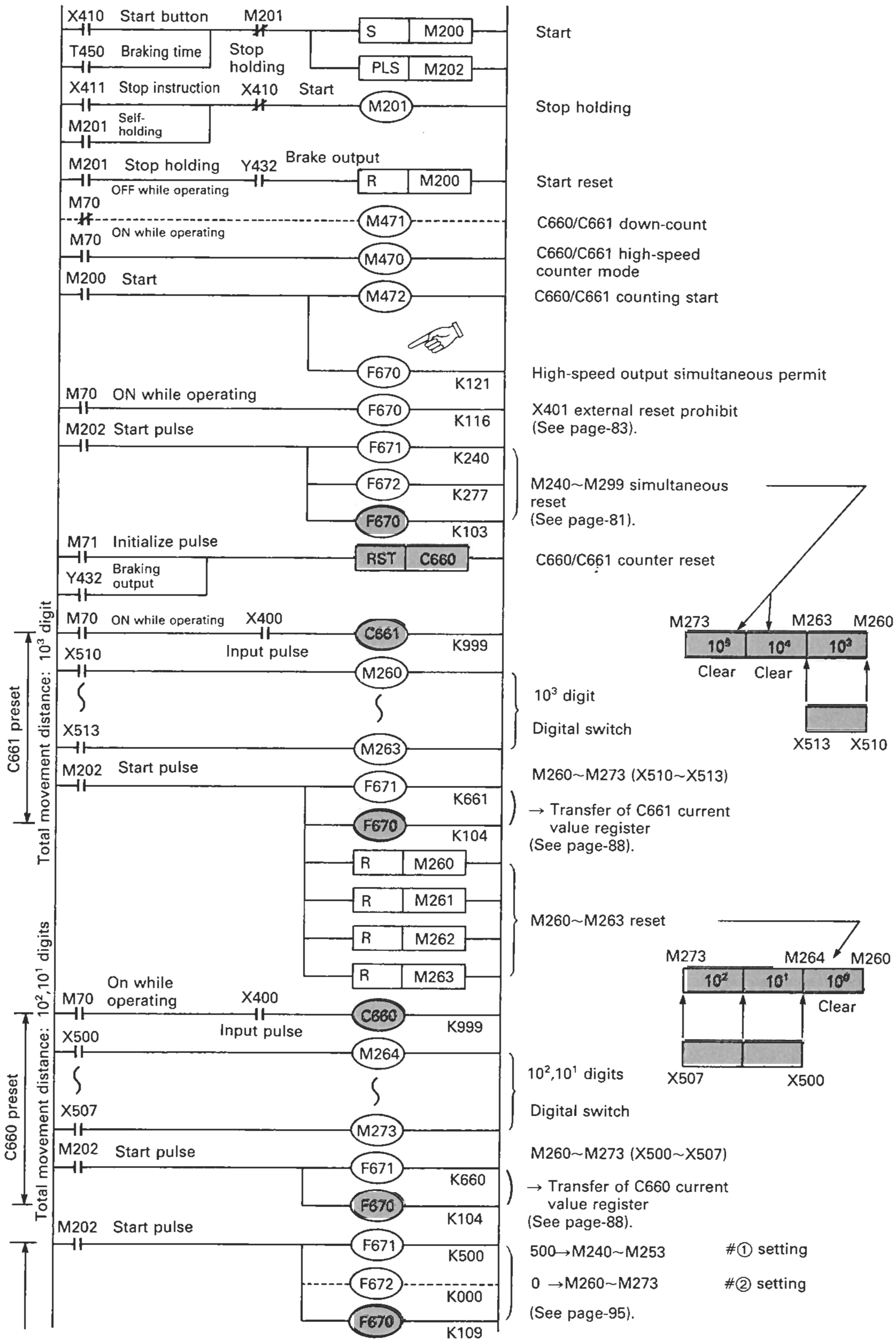
General description of operation

(1) The high-speed counters C660/C661 are handled as down counters for detection of movement amount by the pulse input given from the input X400.
(2) The initial setting value of high-speed counter is set by the 3-digit digital switch from the panel surface. This setting value is total amount movement.
(3) When the current value of high-speed counter is decremented and reached "500", the high-speed forward travelling output Y430 is turned off and the low-speed forward travelling output Y431 is turned on.
(4) When the current value of high-speed counter is further decreased to reach "100", the low-speed forward travelling output Y431 is turned off and the braking output Y432 is turned on.
(5) The operation is re-started from the high-speed forward travelling process upon elapse of three seconds after the braking output Y432 has been turned on.
If the stop input has been entered in this case, the braking output can be maintained.

General description of program

The overall program is shown on pages 145 and 146.
The program is designed in the following procedure;

(1) Start/stop sequence
(2) Setting of modes for counters C660/C661
(3) Total reset of auxiliary relay applied
(4) Preset of counter C661 (10^3 digit of total travelling distance)
(5) Preset of counter C660 (10^2, 10^1 digits of total travelling distance)
(6) Reset of high-speed output Y430 at table No.0 (Set is executed in item-(10)).
(7) Setting of low-speed output Y431 at table No.1
(8) Reset of low-speed output Y431 at table No.2
(9) Set of braking output Y432 at table No.3
(10) Reset of braking output Y432 and set of high-speed output Y430



X410 Start button
M201
S M200
Start
T450 Braking time
Stop holding
PLS M202
X411 Stop instruction
X410 Start
M201
Stop holding
M201 Self-holding
M201 Stop holding
Y432 Brake output
OFF while operating
R M200
Start reset
M70
M471
C660/C661 down-count
M70 ON while operating
M470
C660/C661 high-speed counter mode
M200 Start
M472
C660/C661 counting start
F670 K121
High-speed output simultaneous permit
M70 ON while operating
F670 K116
X401 external reset prohibit (See page-83).
M202 Start pulse
F671 K240
F672 K277
F670 K103
M240~M299 simultaneous reset (See page-81).
M71 Initialize pulse
Y432 Braking output
RST C660
C660/C661 counter reset
M70 ON while operating
X400 Input pulse
C661 K999
M273 M263 M260
10^5 10^4 10^3
Clear Clear
X510
M260
X513
M263
10^3 digit
Digital switch
X513 X510
M202 Start pulse
F671 K661
F670 K104
M260~M273 (X510~X513)
→ Transfer of C661 current value register (See page-88).
R M260
R M261
R M262
R M263
M260~M263 reset
C661 preset
Total movement distance: 10^3 digit
M273 M264 M260
10^2 10^1 10^0
Clear
M70 On while operating
X400 Input pulse
C660 K999
X500
M264
X507
M273
10^2,10^1 digits
Digital switch
X507 X500
M202 Start pulse
F671 K660
F670 K104
M260~M273 (X500~X507)
→ Transfer of C660 current value register (See page-88).
C660 preset
Total movement distance: 10^2,10^1 digits
M202 Start pulse
F671 K500
F672 K000
F670 K109
500→M240~M253 #① setting
0 →M260~M273 #② setting
(See page-95).

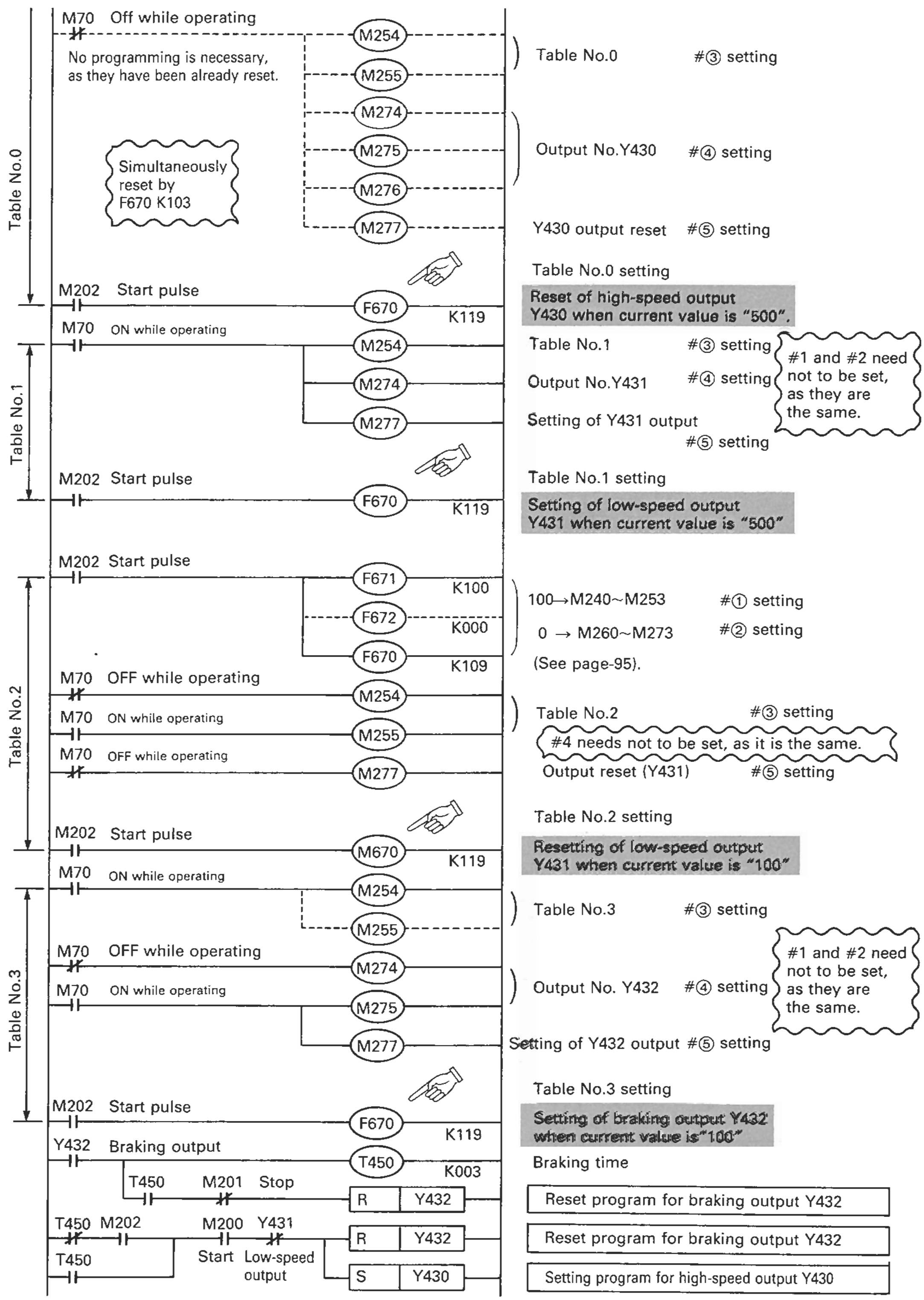


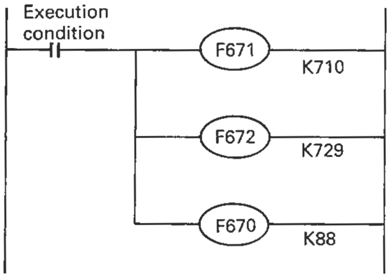
Fig.5-20 Simple positioning control

5-8 Other applied instructions

F670 K88

BCD check of data register

Object elements: D700~777



Head data register No.

End data register No.

Execution instruction

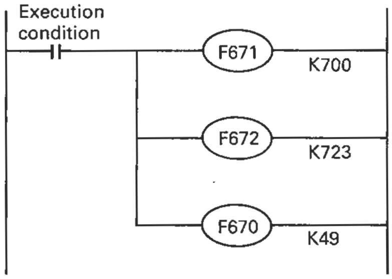
BCD check/format checks are executed for the data register from its head to the end.

- Check is executed when execution condition is turned on.
  If no error is found, M570 is turned off.
  If it is detected, M570 is turned on.
- When execution condition is turned off, no processing is executed.
- BCD check is executed to check if the data register contents is of BCD, and format check to check if the format conforms to that which can be handled within the programmable controller.
- This instruction is useful when checking whether the battery back-up function operates properly when the power is turned on.

F670 K49

Data exchange

Object elements: D700~777



Data register No.

Data register No.

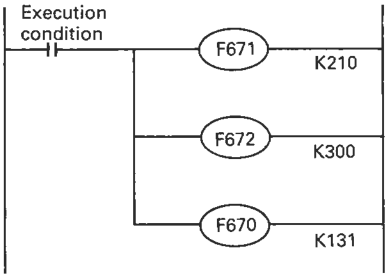
Exchange instruction

- Data exchange is executed when execution condition is turned on.
  When execution condition is turned off, no processing is executed.
- When there occurs an error in data register No. setting, error flag M570 is turned on, and no processing is executed.

  – Before execution
    D700 = 123   D723 = 456
  – After data exchange
    D700 = 456   D723 = 123

F670 K131
Binary conversion

Object elements
- Transfer origin: X, Y, M100~377, S
- Transfer destination: Y, M100~377, S

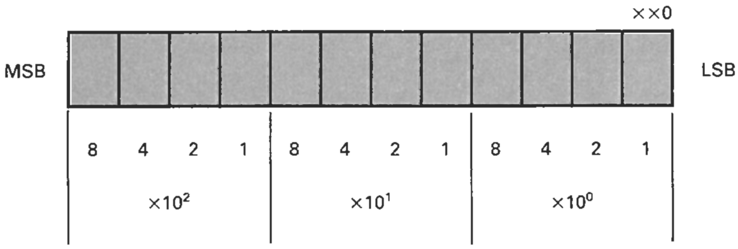


Head No. of transfer origin: BCD

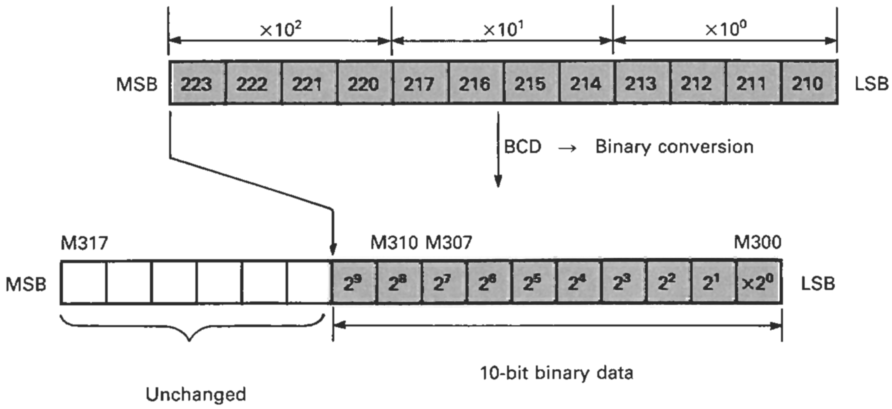
Head No. of transfer origin: BIN

Transfer/conversion instruction

- When execution condition is turned on, conversion transfer is executed.
  When it is turned off, no processing is executed.
- BCD data of element designated by F671 is converted into binary and stored in the element designated by F672 for its head.
- Be sure to set most significant digit of head No. (transfer origin, transfer destination) to "0".
  Setting of other than "0" will result in an error, by which M570 is turned on, and no shifting will be executed.
- The transfer data format is as shown below.
  If input is not of BCD, error flag M570 is turned on and no processing is executed.

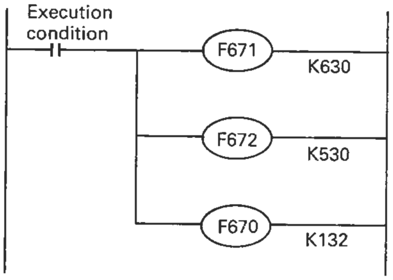


- When a part of data turns out to be out of object element at the transfer origin/transfer destination, error flag M570 is turned on, and no processing is executed.
- When there occurs an error in element No. setting, M570 is turned on, and no processing is executed.



F670 K132
BCD conversion

Object elements
- Transfer origin: X, Y, M100~377, S
- Transfer destination: Y, M100~377, S

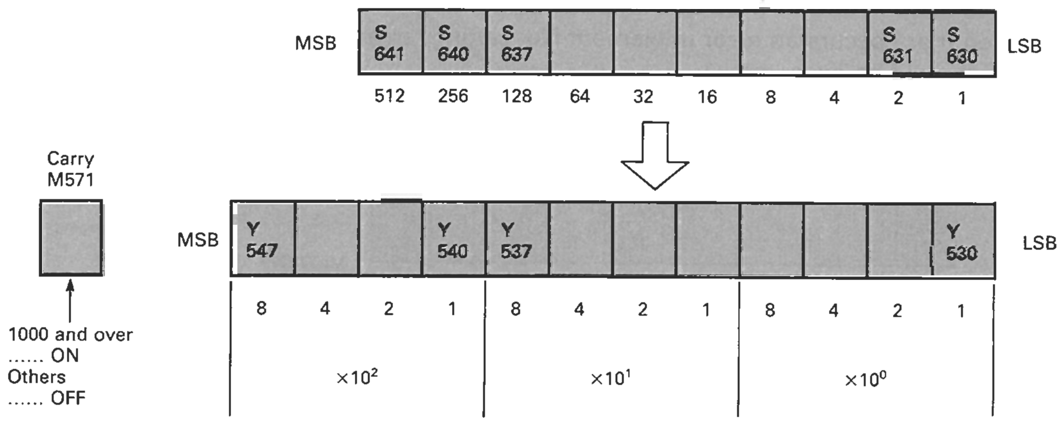


Head No. of transfer origin: BIN

Head No. of transfer origin: BCD

Transfer instruction

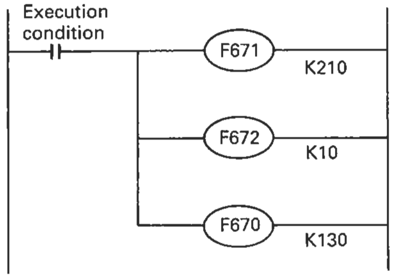
- When execution condition is turned on, conversion/transfer is executed.
  When execution condition is turned off, no processing is executed.
- Element designated by F671 is converted into BCD and stored in the element designated by F672.
- Be sure to set most significant digit of head No. (transfer origin, transfer destination) to "0".
  If other than "0" is set, error flag M570 is turned on, and no prcessing is executed.
- Transfer data format is as shown below;



- When part of data turns to be out of object element at the transfer destination, M570 is turned on, and no transfer is executed.
- If data of transfer origin is more than 1,000, carry is set, and data of 10^2, 10^1 and 10^0 will be written.
- When there occurs an error element No. setting, error flag M570 is turned on, and no processing is executed.

F670 K130 Shift register

Object elements: M100~377

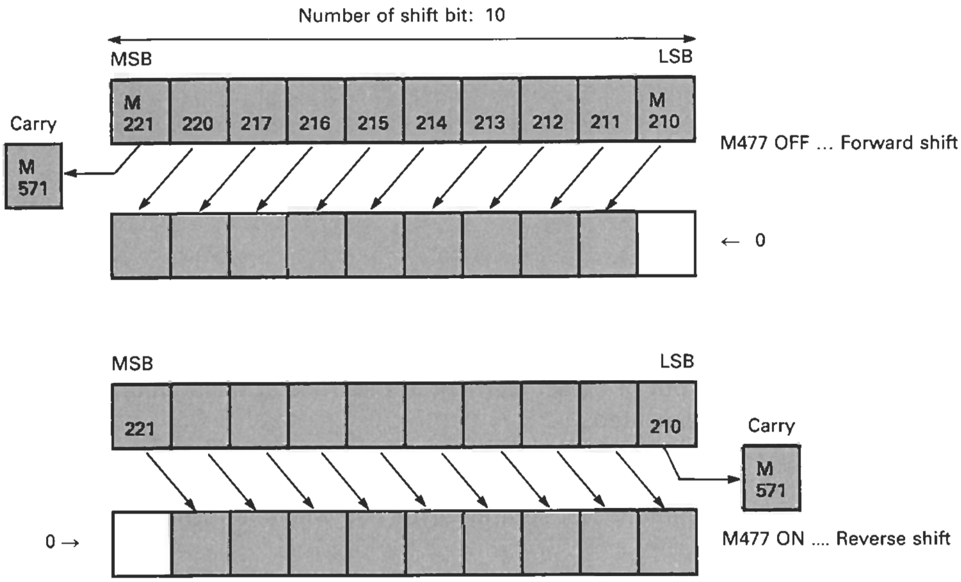


Head element No.

Number of shift bit

Shift instruction

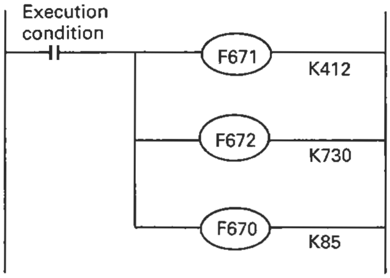
- When execution condition is turned on, shifting is executed.
  When execution condition is turned off, no processing is executed.
- Keep the number of shift bit within the range of 0~192.
  If the number of more than 193 is designated, M570 is turned on, and no shifting is executed.
- Shifting direction is determined by M477.
  - M477 OFF ...... Shifting with LSB (small) → MSB (great)
  - M477 ON ...... Shifting with MSB (great) → LSB (small)
- Overflow bits are filled in carry.
- "0" is filled in LSB or MSB bits emptied by shifting.
- When there occurs an error in element No. setting, error flag M570 is turned on, and no shifting will be executed.



F670 K85

READING OF ANALOG UNIT DATA

Object elements: D700~777

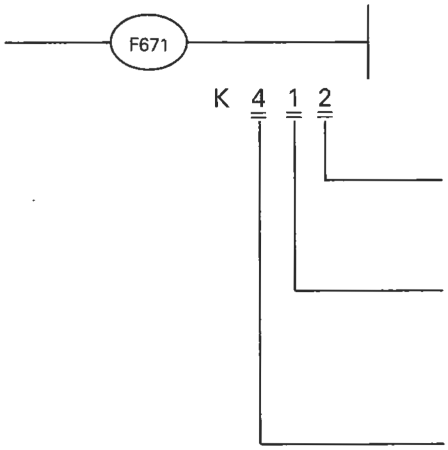


Analog unit No.

Data storage register No. after conversion

Execution Instruction

- When execution condition is turned on, input data of analog unit is converted and stored in the designated data register.
  When it is turned off, no processing will be executed.
- Data is output by A/D converter in 8-bit binary value.
  In this case, the data is converted into BCD in three digits (0~255) for storage in data register.
- Division of equipment No. for F671.



A/D converter channel No.
(designation of 0~3 for A/D)

Division of A/D, D/A
0: D/A
1: A/D

No. of extension connector connecting analog unit

Connector No.
0: 000's
4: 400's
5: 500's

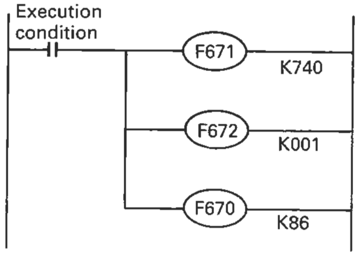
- When there occurs an error in analog unit No. setting, error flag M570 is turned on, and no calculation is executed.
- When there occurs an error in data register No. setting, M570 is turned on, and no calculation is executed.

  Note: Since the analog unit occupies an extension unit I/O, be careful not to run the I/O concerned accidentally on the program.

F670 K86

WRITING OF ANALOG UNIT DATA

Object elements: D700~777

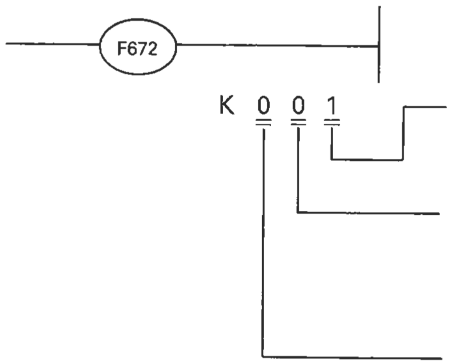


Data register No. (Transfer origin data)

Analog unit No. to output (Channel)

Execution Instruction

- When execution condition is turned on, contents (BCD 3 digits) of data register designated is converted into 8-bit binary value, and transferred to D/A conversion unit for output of analog volume.
- Division of equipment No. for F672.



D/A converter channel No. (selection of "0" or "1" for D/A)

Division of A/D, D/A
0: D/A
1: A/D

No. of extension connector connecting analog unit

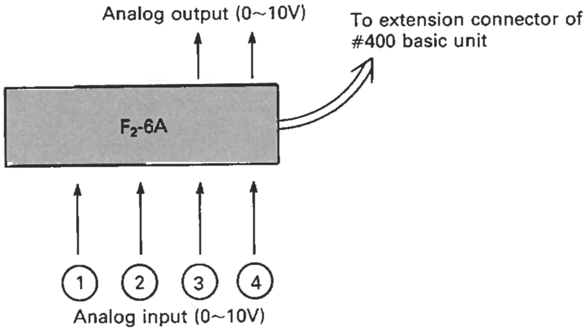
Connector No.
0: 000's
4: 400's
5: 500's

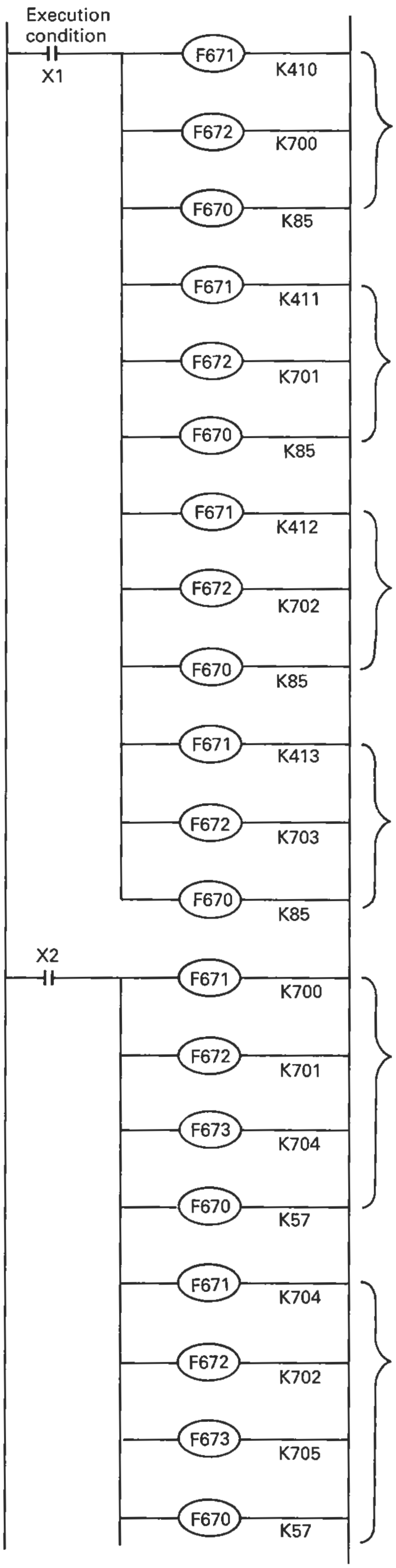
- It is necessary that the data in the data register be in the range from "0" to "255". If any data exceeds "256", it is treated as "255".
- When there occurs an error in analog unit No. setting, error flag M570 is turned on, and no calculation is executed.
- When there occurs an error in data register No. setting, M570 is turned on, and no calculation is executed.

Note: Since the analog unit occupies an extension unit I/O, be careful not to run the I/O concerned accidentally on the program.

Application example-13

- The following shows an application example in which analog unit is used, in which four-point analog input is fetched and the analog unit calculate the averaging of analog volume, and further for outputs the value as analog output.
  In addition, the difference between the value in the channel 0 and its average value is found in absolute value.
  The obtained value is doubled for another analog output.





Converts analog volume of A/D O-CH input into digital value and stores in data register (D700).

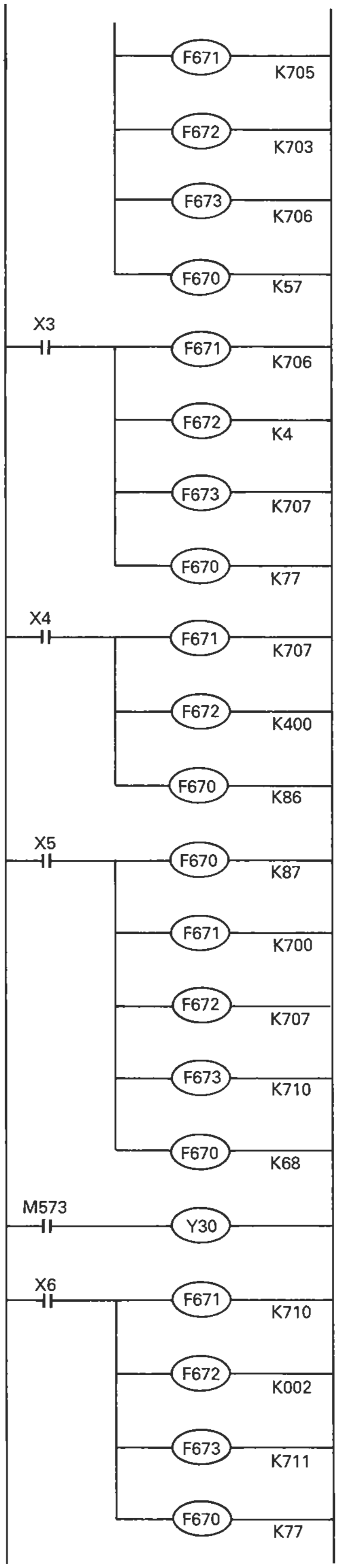
Converts analog volume of A/D 2-CH input into digital value and stores in data reagister (D702).

Converts analog volume of A/D 3-CH input into digital value and stores in data register (D703).

Adds (D700) and (D701) and stores the data in (D704).

Adds (D704) and (D702) and stores the data in (D705).

to be continued



Adds (D705) and (D703) and stores the data in data register (D706)

Divides the data of (D706) by "4" and stores the data in (D707).

D/A converts above data (D707) for output through CH-0 of D/A unit.

Operation executed to display the following result of subtraction in absolute value.

Subtraction between (D700) and (D707) and stores its absolute result value in data register (D710)
If (D700) is smaller than (D707), output Y30 is turned on by borrow flag.

Doubles above absolute value (D710) and stores in data register (D711)

to be continued

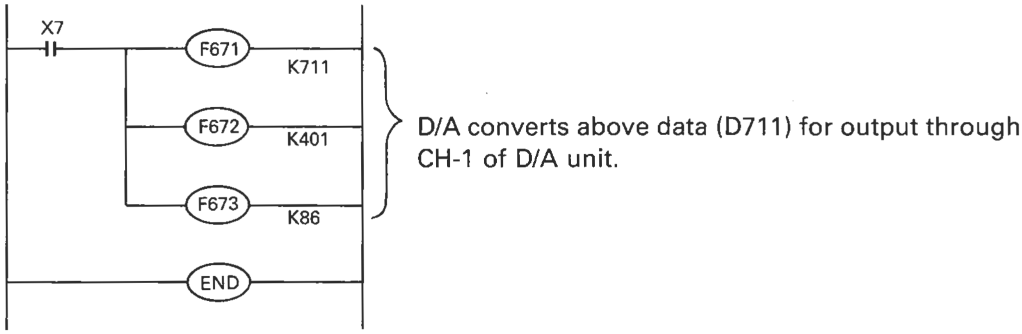


Fig. 5-21 Application example of analog input/output instruction

Note: To execute the above operation, the inputs X0~X7 need to be turned on.

6 SUMMARY

Table 6-1 Instructions and execution time

Page explained	Instructions	Designation	Object factors	Execution time ON	Execution time OFF	General functions	
25	LD	Load	X, Y, M, T, C, S	5.4		Start of logical operation	Normally-open contact
	LDI	Load Inverse		5.4			Normally-close contact
26	AND	AND		4.2		Logical product (series contact)	Normally-open contact
	ANI	AND Inverse		4.2			Normally-close contact
27	OR	OR		4.2		Logical sum (parallel contact)	Normally-open contact
	ORI	OR Inverse		4.2			Normally-close contact
28	ORB	OR Block	None	3.6		Parallel connection of circuit block	
29	ANB	AND Block				Series connection of circuit block	
25	OUT	OUT	Y	34.5	34.5	Coil drive instruction	
			M	31.5	31.5		
			S	36.3	48.8		
			T-K	108	142		
			C-K	120	72		
			F671 ~ F675 – K	126	58.9		
32	PLS	Pulse	M100 ~ M377	49.4	47.0	Rising pulse generating instruction	
34	SFT	Shift	M100, 120, 140, 160, M200, 220, 240, 260, M300, 320, 340, 360	70.2	50.0	Shift register 1-bit shift instruction	
33	RST	Reset		63.7	51.8	Reset instruction for shift register, counter	
			C (excluding C661)	44.6	41.7		
31	S	Set	Y	35.7	29.8	Operation holding coil drive instruction (Note-3)	
			M200 ~ M377	32.7	26.2		
			S	44.6	38.1		
31	R	Reset	Y	38.1	28.0	Operation holding reset coil drive instruction	
			M200 ~ M377	35.1	25.0		
			S	50.6	32.7		
35	MC	Master Control	M100 ~ M177	23.8		Common series contact	
	MCR	Master Control Reset		3.0		Reset of common series contact	
38	CJP	Conditional Jump	700 ~ 777	55.4	28.0	Conditional jump	
	EJP	End of Jump		0		Designation of conditional jump destination	
37	NOP	Nop	None	0		None-processing	
42	END	End		1101 (Note-1)		Program end	
52	STL	Step ladder	S600 ~ S647	14.3 + 69n (Note-2)		Start of step ladder	
	RET	Return		14.3		End of step ladder	

Note-1: Input/output processing time is included.
Note-2: "n" shows the number of longitudinal connection (parallel joining) for STL instruction.
Note-3: 51.2+31.5n for ON and 36.9 for OFF in the STL circuit block.
n ... Number of longitudinal connection (number of parallel joining) for STL instruction.
Note-4: It is estimated that one execution cycle time is K times of total execution time calculated from step 0 to END.

$$K = 1.2 + \underset{①}{\underline{0.15}} + (\underset{②}{\underline{0.16}} + \underset{③}{\underline{0.02}} + \underset{④}{\underline{0.04}})\ \underset{⑤}{\underline{f}}$$

(1) In case T650 T657 are used
(2) In case high-speed counter is used
(3) In case F670 K118 is turned on
(4) In case F670 K121 is turned on
(5) Input frequency of high-speed counter (f=1 for 1kHz)

Table 6-2 Applied instruction and execution time

Page explained	Instruction No. F670	Designations	Execution time (μs)		Remarks
			Input ON	Input OFF	
73	K00	INPUT ALL-POINT REFRESH	582	55.4	Same as K100
73	K02	OUTPUT ALL-POINT REFRESH	289	55.4	Same as K102
82	K04	WDT REFRESH	71.4	55.4	
82	K10	RESET OF M473	60.7	55.4	Same as K110
82	K111	RESET OF C660	60.7	55.4	Same as K111
84	K14	FLAG SET/RESET	70.8	55.4	
84	K15	FLAG SET/RESET	70.8	55.4	
84	K16	FLAG SET/RESET	70.8	55.4	
84	K17	FLAG SET/RESET	70.8	55.4	
84	K18	FLAG SET/RESET	70.8	55.4	
84	K19	FLAG SET/RESET	70.8	55.4	
81	K26	SIMULTANEOUS RESET	223+58.3m	55.4	Same as K103
93	K27	WRITE K → Y, M, S (DECIMAL)	186/180/219	55.4	
94	K28	WRITE K → Y, M, S (OCTAL)	179	55.4	
92	K29	MOVE X, Y, M, S → Y, M, S	242-61.6m	55.4	
94	K33	WRITE K → T, C, D	231	55.4	
90	K34	WRITE X, Y, M, S, → T, C, D	667	55.4	
88	K35	READ T, C, D → Y, M, S	696	55.4	
91	K36	WRITE X, Y, M, S → D	233	55.4	
92	K37	READ D → Y, M, S	214	55.4	
97	K38	WRITE K → N×D	139+21.5m	55.4	
98	K39	MOVE D → N×D	170+21.5m	55.4	
105	K40	COMPARE K : T, C, D	263	55.4	
104	K41	COMPARE T, C, D : X, Y, M, S	685	55.4	
103	K42	COMPARE C, D : X, Y, M, S	345	55.4	
108	K43	ZONE COMPARE $K_1 \sim K_2$: T, C, D	T C D 314, 268, 176	55.4	
109	K44	6 DIGIT ZONE COMPARE $K_1K_2 \sim K_3K_4$: D, C	C/415, D/297	55.4	
112	K45	COMPARE D, C : D, C	C/374, D/719	55.4	

Table 6-2 Applied instruction and execution time

Page explained	Instruction No. F670	Designations	Execution time (μs)		Remarks
			Input ON	Input OFF	
85	K46	ZERO CHECK OF DATA REGISTER	113	55.4	
85	K48	CLEAR OF DESIGNATED DIGIT	169	55.4	
147	K49	DATA EXCHANGE	170	55.4	
96	K51	MOVE T, C, D → T, C, D	MAX 398 MIN 158	55.4	
98	K52	INDIRECT MOVE (D) → D	179	55.4	
99	K53	INDIRECT MOVE D → (D)	179	55.4	
99	K54	INDIRECT MOVE (D) → (D)	213	55.4	
114	K55	ADDITION D+K	207	55.4	
115	K56	ADDITION D+K+Cy	223	55.4	
116	K57	ADDITION D+D	228	55.4	
117	K58	ADDITION D+D+Cy	232	55.4	
118	K59	ADDITION D+D+Cy	272	55.4	
119	K60	ADDITION D+D	245	55.4	
119	K61	INCREMENT D+1	138	55.4	
120	K62	INCREMENT D+1	164	55.4	
121	K63	INCREMENT D+1	148	55.4	
121	K64	INCREMENT C+1	223	55.4	
122	K66	SUBTRACTION D-K-Br	225	55.4	
123	K67	SUBTRACTION D-K-Br	268	55.4	
124	K68	SUBTRACTION D-D	248	55.4	
125	K69	SUBTRACTION D-D-Br	251	55.4	
126	K70	SUBTRACTION D-D-Br	296	55.4	
127	K71	SUBTRACTION D-D	240	55.4	
128	K72	DECREMENT D-1	136	55.4	
129	K73	DECREMENT D-1	164	55.4	
129	K74	DECREMENT D-1	145	55.4	
130	K75	DECREMENT C-1	224	55.4	
131	K77	MULTIPLICATION D×K	629	55.4	

Table 6-2 Applied instruction and execution time

Page explained	Instruction No. F670	Designations	Execution time (μs)		Remarks
			Input ON	Input OFF	
131	K78	MULTIPLICATION D×K	3438	55.4	
132	K79	MULTIPLICATION D×D	654	55.4	
133	K80	MULTIPLICATION D×D	3438	55.4	
133	K81	DIVIDE D/K	1490	55.4	
134	K82	DIVIDE D/K	4571	55.4	
135	K83	DIVIDE D/D	1514	55.4	
135	K84	DIVIDE D/D	4601	55.4	
151	K85	READING OF ANALOG UNIT DATA	661	55.4	
152	K86	WRITING OF ANALOG UNIT DATA	700	55.4	
122	K87	FORMAT OF SUBTRACTION	71.4	74.4	
147	K88	BCD CHECK OF DATA REGISTER	133+22n	55.4	
73	K100	INPUT ALL-POINT REFRESH	582	55.4	Refer to K00
75	K101	PARTIAL INPUT REFRESH	213+22n	213+14.9n	
73	K102	OUTPUT ALL-POINT REFRESH	289	55.4	Refer to K02
81	K103	SIMULTANEOUS RESET	223+58.3n	55.4	Refer to K26
88	K104	WRITE M → C	213	55.4	
87	K105	READ C → M	192	55.4	
106	K106	ZONE COMPARE $K_1 \sim K_2$: C	248	55.4	
101	K107	COMPARE C : M260 ~ M273	238	55.4	
110	K108	6 DIGIT ZONE COMPARE $K_1K_2 \sim K_3K_4$: C	365	55.4	
95	K109	WRITE K_1K_2 → M	120	55.4	
82	K110	RESET OF M473	60.7	55.4	Refer to K10
82	K111	RESET OF C660	60.7	55.4	Refer to K11
77	K112	X400 RISING DETECTION	70.2	75.6	
77	K113	X400 RISING DETECTION	67.3	72.6	
77	K114	X401 RISING DETECTION	70.2	75.6	
77	K115	X401 RISING DETECTION	69.6	75.0	
83	K116	EXTERNAL RESET INHIBIT	64.3	55.4	

Table 6-2 Applied instruction and execution time

Page explained	Instruction No. F670	Designations	Execution time (µs)		Remarks
			Input ON	Input OFF	
137	K117	TRANSFER OF COMPARE DATA FOR AUTO RE-LOAD	130	55.4	
137	K118	AUTO RE-LOAD VALID	67.3	72.4	
141	K119	SETTING OF HIGH-SPEED OUTPUT TABLE	180	55.4	
143	K120	PROHIBIT OF INDEPENDENT HIGH-SPEED OUTPUT	78.0	83.3	
143	K121	HIGH-SPEED OUTPUT SIMULTANEOUS PERMIT	67.3	72.6	
78	K122	MEASUREMENT OF X402 PULSE SIGNAL WIDTH	138	78	
79	K123	MEASUREMENT OF X403 PULSE SIGNAL WIDTH	138	78	
79	K124	X400 PULSE SIGNAL COUNT	113	81	
80	K125	X401 POLSE SIGNAL COUNT	113	81	
150	K130	SHIFT REGISTER	328+38n	55.4	
148	K131	BINARY CONVERSION	430	55.4	
149	K132	BCD CONVERSION	375	55.4	

Table 6-3 List of special aux. relays (see page 16, 17)

M77	M76			M73	M72	M71	M70
Output inhibit	Battery voltage drop			10ms clock	100ms clock	Initialize pulse	RUN monitor
All output OFF when M77 is turned ON	ON when battery voltage is dropped			Oscillated at 10ms cycle	Oscillated at 100ms cycle	ON for 1-scanning after RUN	ON during RUN

				M473	M472	M471	M470
				Shift up/down (holding of operation in event of power failure)	Start of counting (high-speed counter)	UP/DOWN (C660/C661)	High-speed counter (C660/C661)
				Set in the case of shift up/shift down in C660/ C661, and auto re-load	ON: counting OFF: Stopping	ON: Up OFF: Down	ON: High-speed counter OFF: Internal counter

		M575	M574	M573	M572	M571	M570
		Return start of state transfer	State transfer inhibit (holding of operation in event of power failure)	Borrow flag	Zero flag	Carry flag	Error flag
			ON: Transfer inhibit OFF: Transfer permit			Detection of X400, X401 rising	

Table 6-4 List of element Nos.

	00 ~ 13	14 ~ 27	30 ~ 37	40 ~ 47	50 ~ 57	60 ~ 67	70 ~ 77
000's	X: 12 points	X: 12 points	Y: 8 points	Y: 8 points	T: 8 points 0.1 ~ 999s	C: 8 points 1 ~ 999	SPM: 6 points
100's	M: 64 points						
200's	M: 64 points						
300's	M: 64 points						
400's	X: 12 points	X: 12 points	Y: 8 points	Y: 8 points	T: 8 points 0.1 ~ 999 s	C: 8 points 1 ~ 999	SPM: 4 points
500's	X: 12 points	X: 12 points	Y: 8 points	Y: 8 points	T: 8 points 0.1 ~ 999 s	C: 8 points 1 ~ 999	SPM: 6 points
600's	S: 40 points				T: 8 points 0.01 ~ 99.9's	C | C: 6 points 1 ~ 999	F: 6 points
700's	CJP/EJP 64 points						

C660/C661
0 ~ 999999 reversible one point

Dotted box: Extension unit   Solid box: Battery back up

X .: Input relay   Y : Output relay   M : Aux. relay   SPM : Special aux. relay
T : Timer   C : Counter   S : State   F : Coil for applied instruction

Input/output relay Nos. (basic unit)

Basic unit	Input relay Nos.	Output relay Nos.	Extension connector
F_1 − 12M	400 ~ 405 6p	430 ~ 435 6p	400
F_1 − 20M	400 ~ 412 12p	430 ~ 437 8p	400
F_1 − 30M	400 ~ 413 12p 500 ~ 503 4p	430 ~ 437 8p 530 ~ 535 6P	400
F_1 − 40M	400 ~ 413 12p 500 ~ 513 12p	430 ~ 437 8p 530 ~ 537 8p	400 500
F_1 − 60M	000 ~ 013 12p 400 ~ 413 12p 500 ~ 513 12p	030 ~ 037 8p 430, ~ 437 8p 530 ~ 537 8p	000 400 500

p: points

The value in □ of extension unit will be "0", "4" or "5", depending upon extension connector Nos. 000, 400 or 500.

Input/output relay Nos. (extension unit)

Extension unit	Input relay Nos.	Output relay Nos.
F − 4T	□ 20 ~ □ 23 4p	□ 40 ~ □ 43 4p
F_2 − 8EY	—	□ 40 ~ □ 47 8p
F_1 − 10E F − 10E	□ 14 ~ □ 17 4p	□ 40 ~ □ 45 6p
F_2 − 12EX	□ 14 ~ □ 27 12p	—
F_1 − 20E F_2 − 20E F − 20E	□ 14 ~ □ 27 12p	□ 40 ~ □ 47 8p
F_1 − 40E F_2 − 40E F − 40E	414 ~ 427 12p 514 ~ 527 12p	440 ~ 447 8p 540 ~ 547 8p
F_1 − 60E F_2 − 60E	014 ~ 027 12p 414 ~ 427 12p 514 ~ 527 12p	040 ~ 047 8p 440 ~ 447 8p 540 ~ 547 8p

p: points

MEMO

MEMO

MEMO

MITSUBISHI ELECTRIC CORPORATION

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

JY992D083-01A
Printed in Japan (ROD)

Effective MAR. 1987