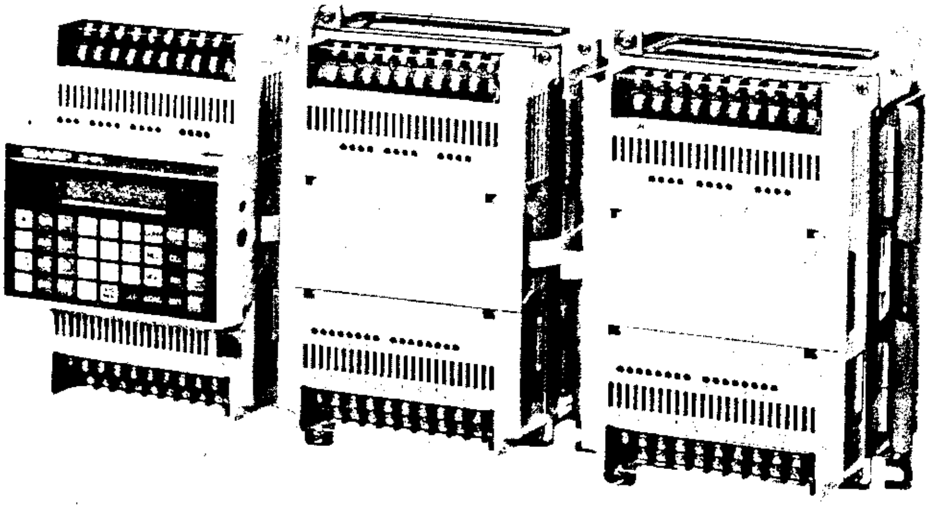
SHARP

New Satellite W1Ø
PROGRAMMABLE CONTROLLER
User's Manual - 2



SHARP CORPORATION

Contents

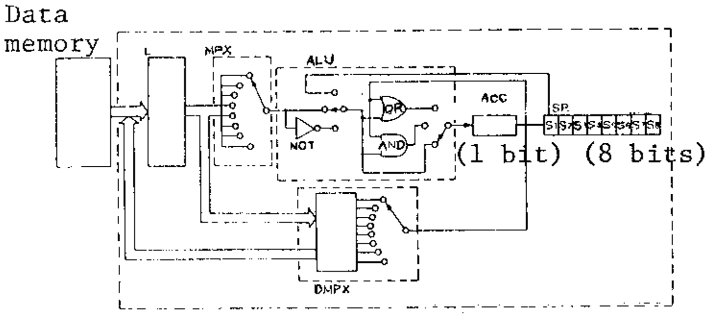
User's Manual 2
Page

Page

Page

9-2. Bit processing

Bit processing means the logical operation of the contact signal which is carried out by the CPU with the W10. See the block diagram next for bit operation by the CPU.



L	(latch)
MPX	(bit extraction)
ALU	(single bit logical unit)
ACC	(accumulator)
SR	(stack register)
DMPX	(bit insertion)

(1) L (latch)

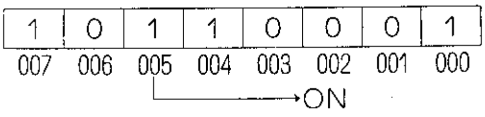
The bit processing instruction has the instruction field and the address field.

```
STR    005
        └──────── Address field
 └─────────────── Instruction field
```

The address field represents the relay number of the data memory relay area (I/O relay, link relay, auxiliary relay,Retention relay). When the ON/OFF information of the relay is read from the data memory, the contents of one byte (8 bits) contained in that relay number are read and latched in the latch L. In the case of the instruction "STR 005 ", for instance, eight bits of 000 ∿ 007 are read.

(2) MPX (multiplexer)

The required one bit is extracted out of eight bits in the latch L. In the case of the instruction "STR 005 ", ON/OFF information of 005 is extracted out of 000 ∿ 007 .



1	0	1	1	0	0	0	1
007	006	005	004	003	002	001	000

(3) ALU (one bit logical unit)

Logical operation takes place according to the contents of the instruction field of the given instruction.

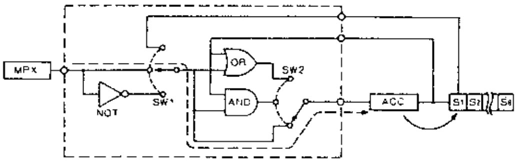


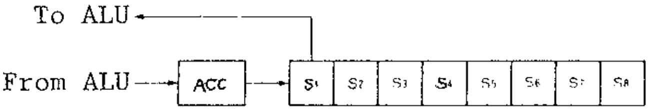
Figure above shows how the operation takes place for the STR instruction. The SW1 and SW2 is chosen according to the contents of the instruction field.

(4) ACC (accumulator)

A single bit register in which the result of the ALU is stored.

(5) SR (stack register)

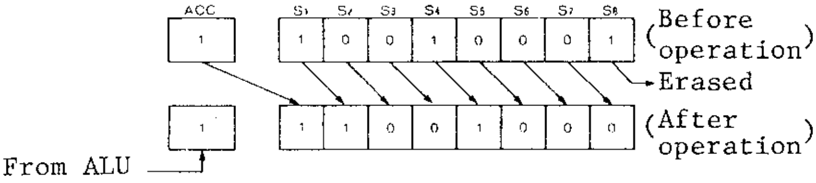
An 8-bit register in which the intermediate result is stored during processing of the serial/parallel circuit or application instruction that has plural number of input conditions.



① Shift register action during execution of the STR or STR NOT instruction

° With the STR and STR NOT instruction, the ON/OFF state of the one bit read from the data memory is stored in the accumulator. With the STR NOT instruction, the bit is inverted before being stored in the accumulator.

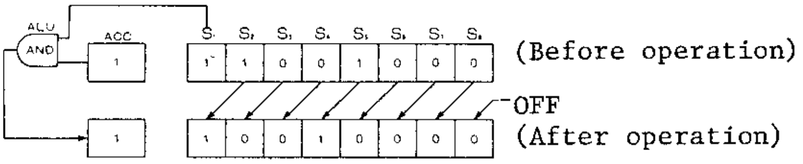
° The ON/OFF information previously stored in the accumulator shifts to S_1, the information in S_1 to S_2, one in S_2 to S_3, ... S_3 to S_4, S_4 to S_5, S_5 to S_6, S_6 to S_7, and S_7 to S_8. And the information in S_8 is then erased.



ACC	S1	S2	S3	S4	S5	S6	S7	S8	
1	1	0	0	1	0	0	0	1	(Before operation)
1	1	1	0	0	1	0	0	0	(After operation)

② Shift register action during execution of AND STR or OR STR instruction

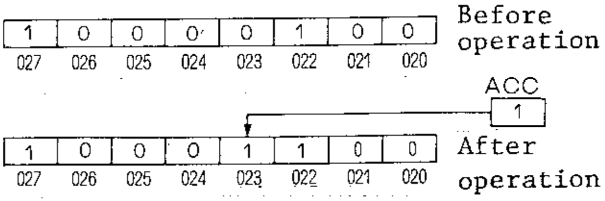
- ° With the AND STR and OR STR instruction, the ON/OFF information in S_1 is stored in the ALU, to be ANDed or ORded with the accumulator contents, then the result is stored in the accumulator.
- ° The ON/OFF information of S_1 which becomes unwanted after the operation is erased. The information in S_2 shifts to S_1, S_3 to S_2, S_4 to S_3, S_5 to S_4, S_6 to S_5, S_7 to S_6, S_8 to S_7. And S_8 is then set to OFF.



(6) DMPX (demultiplexer)

With the OUT instruction, the one bit, out of eight bits latched in L, implied by the address field of the instruction is converted into the result of the operation (contents of the accumulator), and the whole one byte is transferred to the data memory.

(In case the operational result is ON for OUT 023.)



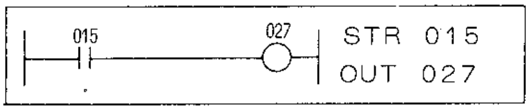
NOTE-1: Refer to 9-3 "Description of basic instructions" for action of bit processing unit.

9-3 Description of basic instructions

[1] STR/OUT

STR The contents (ON/OFF state) of the specified data memory is stored in the accumulator (ACC). The previous contents of the accumulator is shifted to the stack register (SR).

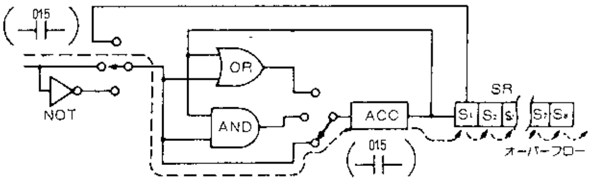
OUT The contents of the accumulator are sent to the data memory.



```
STR 015
OUT 027
```

STR 015

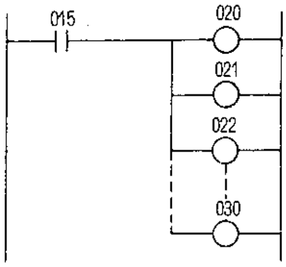
- ° L (latch) ..... Eight bits (010 ∿ 017) are read from the data memory.
- ° MPX .... One bit (015) is extracted from eight bits in the latch L.
- ° ALU, ACC, SR .. The MPX output is then directly stored in the accumulator. The previous accumulator contents are shifted to the shift register S_1.



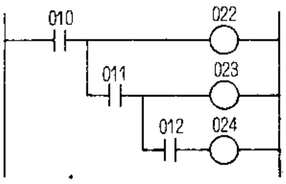
OUT 027

- ° L (latch) ..... Eight bits (020 ∿ 027) are read from the data memory.
- ° MPX .... Not involved in the OUT instruction.
- ° ALU, ACC, SR .. The contents of ACC and SR remain unaffected.
- ° DMPX .... One bit (027) in the latch L is replaced by the contents of the accumulator and eight bits of 020 ∿ 027 are then sent to the data memory.

REFERENCE: The following kind of program may be used effectively as the contents of the accumulator are not affected after the execution of the OUT instruction.



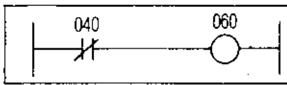
```
STR 015
OUT 020
OUT 021
OUT 022
 :   :
OUT 030
```



```
STR 010
OUT 022
AND 011
OUT 023
AND 012
OUT 024
```

[2] STR NOT

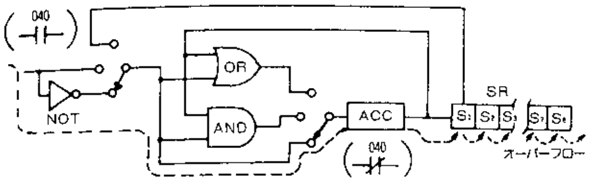
- The contents of the specified data memory are inverted and stored in the accumulator. The previous accumulator contents is shifted to the stack register S_1.



```
STR NOT 040
OUT     060
```

STR NOT 040

- L (latch) ..... Eight bits of 040 ∿ 047 are read from the data memory.
- MPX .... One bit (040) is extracted from eight bits in the latch.
- ALU, ACC, SR .. The MPX output is inverted and written in the accumulator. The previous accumulator contents are shifted to the stack register S_1.

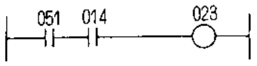


OUT 060

Result of $\overset{040}{\dashv\!\not\,\vdash}$ is written in the data memory (060).

[3] AND

- The contents of the specified data memory are ANDed with the contents of the accumulator and its result is stored in the accumulator.



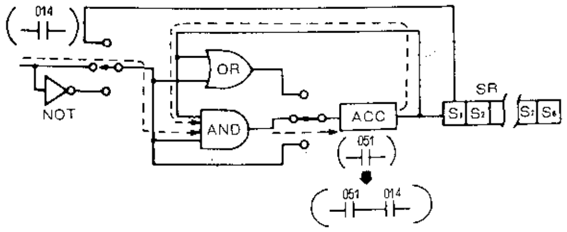
```
STR 051
AND 014
OUT 023
```

STR 051

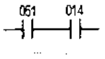
The contents of the data memory (051) are stored in the accumulator.

AND 014

- L (latch) ..... Eight bits of 010 ∿ 017 are read from the data memory.
- MPX .... One bit (014) is extracted from eight bits in the latch.
- ALU, ACC, SR .. The contents of the accumulator (051) are ANDed with the MPX output (014) and its result is stored in the accumulator. The contents of the stack register are retained.

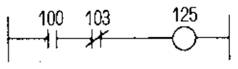


OUT 023

Result of  is written in the data memory (023).

[4] AND NOT

° The contents of the specified data memory are inverted and ANDed with the contents of the accumulator, and its result is stored in the accumulator.



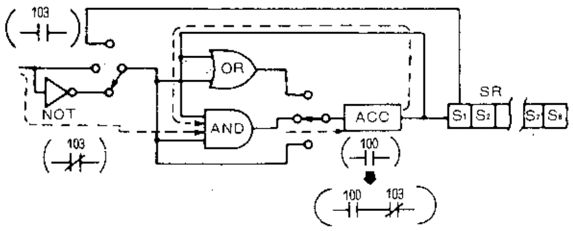
```
STR       100
AND NOT   103
OUT       125
```

STR 100

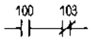
The contents of the data memory (100) are stored in the accumulator.

AND NOT 103

- ° L (latch) ..... Eight bits of 100 ∿ 107 are read from the data memory.
- ° MPX .... One bit (103) is extracted from eight bits in the latch.
- ° ALU, ACC, SR .. Inversion of the MPX output (103) and the contents of the accumulator (100) are ANDed and its result is written in the accumlator. The contents of the stack register are retained.

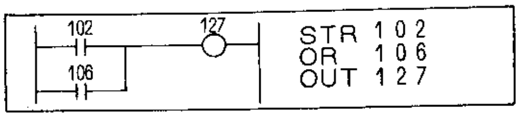


OUT 125

Result of  is written in the data memory (125).

[5] OR

° The contents of the specified data memory are ORed with the contents of the accumulator and its result is stored in the accumulator.



```
STR 102
OR  106
OUT 127
```

STR 102

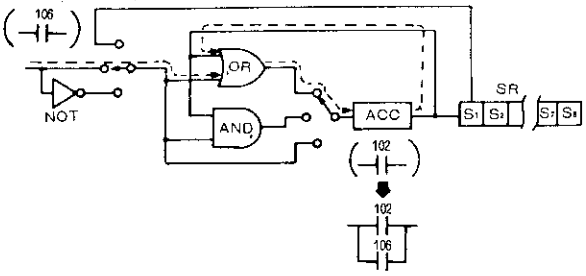
The contents of the data memory (102) are stored in the accumulator.

OR 106

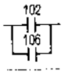
° L (latch) ..... Eight bits of 100 ∿ 107 are read from the data memory.

° MPX .... One bit (106) is extracted from eight bits in the latch.

° ALU, ACC, SR .. The contents of the accumulator (102) are ORed with the MPX output (106) and its result is written in the accumulator. The contents of the stack register are retained.

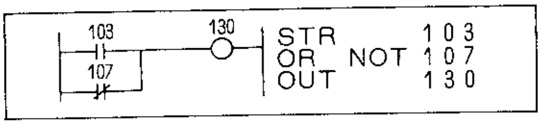


OUT 127

Result of  is written in the data memory (127).

[6] OR NOT

° The contents of the specified data memory are inverted and ORed with the contents of the accumulator and its result is stored in the accumulator.



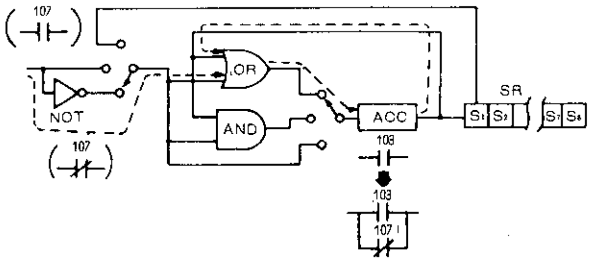
```
STR      103
OR   NOT 107
OUT      130
```

STR 103

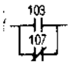
The contents of the data memory (103) are stored in the accumulator.

OR NOT 107

- ° L (latch) ..... Eight bits of 100 ∿ 107 are read from the data memory.
- ° MPX .... One bit (107) is extracted from eight bits in the latch.
- ° ALU, ACC, SR .. Inversion of the MPX output (107) and the contents of the accumulator (103) are ORed and its result is written in the accumulator. The contents of the stack register are retained.

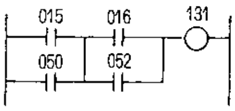


OUT 130

Result of  is written in the data memory (130).

[7] AND STR

° The contents of the stack register S₁ is ANDed with the contents of the accumulator and its result is stored in the accumulator.

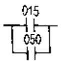


STR	015
OR	050
STR	016
OR	052
AND STR	
OUT	131

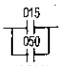
STR 015

The contents of the data memory (015) are stored in the accumulator.

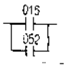
OR 050

Result of  is stored in the accumulator.

STR 016

Result of  contained in the accumulator is saved in the stack register S_1 and the contents of the data memory (016) are written in the accumulator.

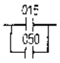
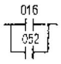
OR 052

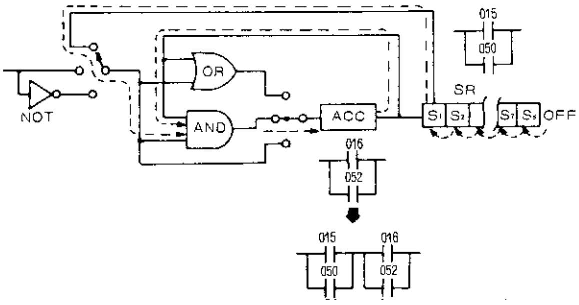
The result of  is stored in the accumulator.

AND STR

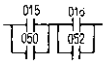
° L (latch) ..... No action takes place in the case of the AND STR instruction.

° MPX .... No action takes place in the case of the AND STR instruction.

° ALU, ACC, SR .. The contents of the stack register S_1  and the contents of the accumulator  are ANDed and its result is stored in the accumulator.

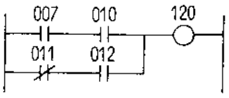


OUT 131

Result of  is written in the data memory (131).

[8] OR STR

° The contents of the stack register S_1 are ORed with the contents of the accumulator and its result is stored in the accumulator.

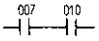


```
STR       007
AND       010
STR NOT   011
AND       012
OR  STR
OUT       120
```

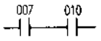
STR 007

The contents of the data memory (007) are stored in the accumulator.

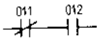
AND 010

Result of  is stored in the accumulator.

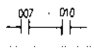
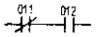
STR NOT 011

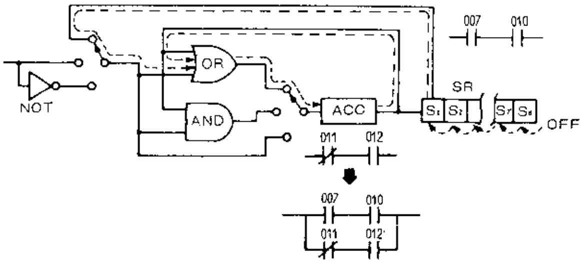
Result of  contained in the accumulator is saved in the stack register S_1 and the contents of the data memory (011) are inverted and written in the accumulator.

AND 012

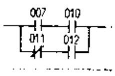
Result of  is stored in the accumulator.

OR STR

- ° L (latch) ..... No action takes place in the case of the OR STR instruction.
- ° MPX .... No action takes place in the case of the OR STR instruction.
- ° ALU, ACC, SR .. The contents of the stack register S_1  and the contents of the accumulator  are ORed and its result is stored in the accumulator.



OUT 120

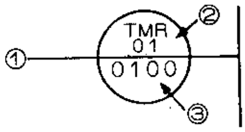
Result  is written in the data memory (120).

[9] TMR (timer instruction)

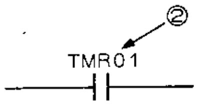
The TMR instruction decrements the timer which has the 0.1(0.01)second clock for its internal clock frequency. Counting does not take place while the start input is OFF and makes the preset value maintained for the current value, and the TMR contact is at OFF. As the start input turns ON, the current value is decremented by one every 0.1(0.01)second When the current value becomes 0, it makes the TMR contact to turn ON, and the condition is maintained as long as the start input is ON.

Start input	Current value	TMR contact
OFF	Preset value	OFF
ON (current value > 0)	Decremented by one every 0.1 (0.01) second	OFF
ON (current value = 0)	0	ON
ON → OFF (current value > 0)	Reset to the preset value	OFF
ON → OFF (current value = 0)	Reset to the preset value	ON → OFF

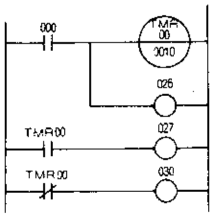
Symbol representing the TMR instruction



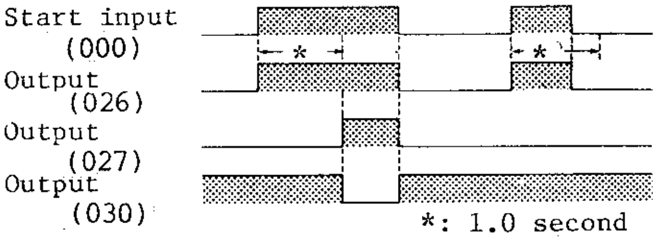
Symbol representing the TMR contact



①	Start input	(start with ON state of signal)
②	TMR number	00 ∿ 57 (octal number) ... Used common for CNT and MD.
③	Preset value	0001 ∿ 1999 (BCD) ... In an increment of 0.1 second (0.1 ∿ 199.9 seconds) 0.01 second (0.01 ∿ 19.99 seconds)



```
STR              000
TMR               00
                0010
OUT              026
STR        TMR    00
OUT              027
STR NOT TMR       00
OUT              030
```



4 points of TMR54∿TMR57 are available for 10 ms timer or 100 ms timer by the preset value of system memory #227. Otherwise, TMR00∿TMR53 are available for 100 ms timer only.

NOTE-1: Because the TMR number is also used common with a CNT and MD number, it should not be used for a TMR number if it has used for a CNT and MD number. In addition, do not use it for the same TMR number. In case the same number be used, it will be alerted as an error as it is program checked by such as the ZW-10PG1 programmer.

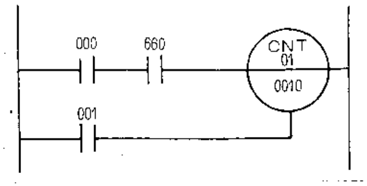
NOTE-2: TMR contact can be the same number as the TMR number and any number of a-contact and b-contact may be used.

NOTE-3: The current number of TMR is stored in 96 bytes of b.000 ∿ b.137. Refer to 8-2[4] "TMR, CNT, MD data storage area".

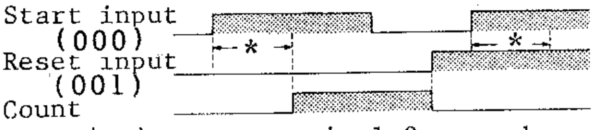
NOTE-4: Timer is reset at power on to the programmable controller. Therefore, the preset value will be used for the current value even if power is turned on to the programmable controller with the timer start input is in the ON state, because of the reset function.

NOTE-5: By setting the timer reset condition in the system memory (#201), it is possible to store the state at the time of a power interrupt. Refer to 8-3 "System memory".

NOTE-6: It is possible to implement the power failure timer and timers differing in their start and reset conditions by making use of the contact 660 (0.1-second clock) and the CNT instruction.



Instruction	
STR	000
AND	660
STR	001
CNT	01
	0010



*: 1.0 second

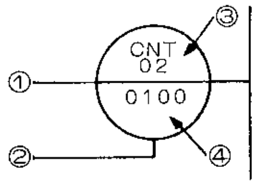
[10] CNT (counter instruction)

The CNT instruction decrements the counter by one at every low to high transition of the counter input.

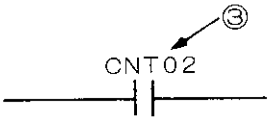
No counting takes place even if the counter input changes its state from OFF to ON as long as the reset input is ON, makes the preset value maintained for the current value, and the CNT contact is at OFF. As the counter changes from OFF to ON while the reset input is OFF, it makes the current value decremented by one. When the current value becomes 0, it makes the CNT contact to turn ON and its state is maintained until the reset input turns ON .

Reset input	Current value	CNT contact
ON	Preset value	OFF
OFF (current value > 0)	Decrements by one every time the counter input changes from OFF to ON.	OFF
OFF (current value = 0)	0	ON
OFF → ON (current value > 0)	Reset to the preset value	OFF
OFF → ON (current value = 0)	Reset to the preset value	ON → OFF

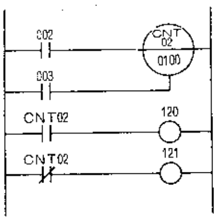
Symbol representing the CNT instruction



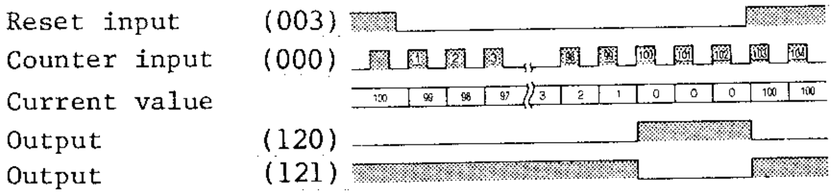
Symbol representing the CNT contact



① Counter input		(senses OFF to ON transition)
② Reset input		(reset with ON)
③ CNT number		00 ∿ 57 (octal number) ... Used commonly with TMR and MD.
④ Preset value		0001 ∿ 1999 (BCD)



```
STR              002
STR              003
CNT               02
                0100
STR         CNT   02
OUT              120
STR NOT CNT       02
OUT              121
```



NOTE-1: Because the CNT number is also used common with a TMR and MD number, it should not be used for a CNT number if it has been used for a TMR or MD number. In addition, do not use it for the same CNT number. In case the same number be used, it will be alerted as an error as it is program checked by such as the ZW-10PG1 programmer. If it is to be used intentionally, the alert can be disregarded.

NOTE-2: CNT contact can be the same number as the CNT number and any number of a-contact and b-contact may be used.

NOTE-3: As it disregards further input when the counter has reached zero, it needs to set the reset input ON and OFF or forced to reset using such as the ZW-10PG1 programmer, in order to start counting again.

NOTE-4: When both the counter input and the reset input go ON simultaneoutly, the reset input takes preference over the counter input.

NOTE-5: The current number of CNT is stored in 96 bytes of b.000 ∿ b.137. Refer to 8-2[4] "TMR, CNT, MD data storage area".

NOTE-6: At a time of power interrupt, the counter retains the current value. However, the current value will be reset when the reset input should turn ON at power on. In case the current value should be retained after power interrupt, it will need input of a reset input which turns OFF at power on.

NOTE-7: By setting the reset condition in the system memory (#202), it is possible to reset with the OFF state of the signal.

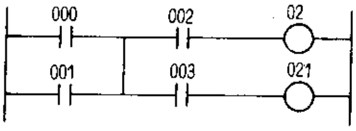
9-4 Ladder design cautionary notes

Since the programmable controller operates in the serial mode which program contents are read successively for execution, it may sometimes not permit to perform direct execution of the ladder chart provided for the relay board. Further, it may not need the use of the one-way control diode which used with the relay board and may not limit the number of auxiliary contact points.
Difference in ladder design between the relay board and the programmable controller should be comprehensively understood in order to comprise more effective ladder chart.

[1] Relay circuit that needs revision

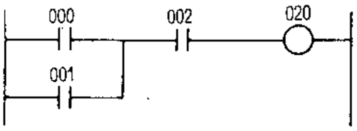
(Ex-1)

(a)



The programmable controller can not execute the ladder chart (a) without revision.

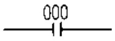
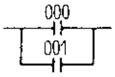
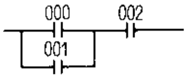
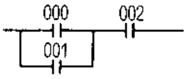
(b)

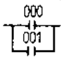


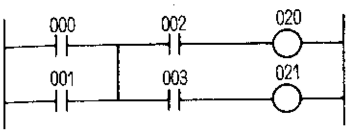
Instruction	
STR	000
OR	001
AND	002
OUT	020

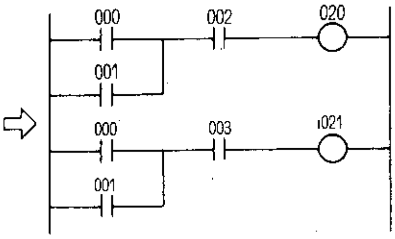
Part (b) can be executed with the program shown.

The state of the accumulator makes transition in the following manner, when the program (b) is executed.

Instruction	Accumulator contents
STR 000	
OR 001	Result of 
AND 002	Result of 
OUT 020	Result of 

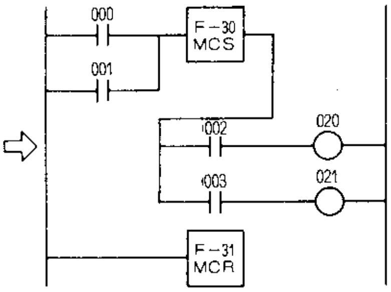
Every time one step of the program is executed, the accumulator contents are changed and represented by 0 or 1. Thus, execution up to AND 002 makes the result of  erased already so that it would not able to reflect it on 003 . So, it has to be revised in the following way for the programmable controller ladder chart.





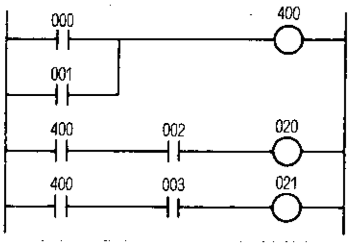
Instru.	
STR	000
OR	001
AND	002
OUT	020
STR	000
OR	001
AND	003
OUT	021

Or



Instru.	
STR	000
OR	001
F-30	
STR	002
OUT	020
STR	003
OUT	021
F-31	

Or



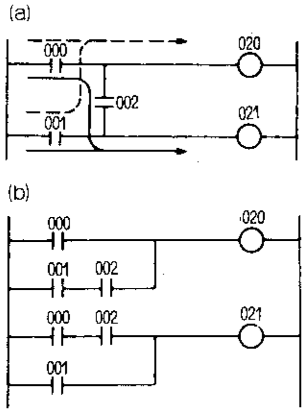
Instru.	
STR	000
OR	001
OUT	400
STR	400
AND	002
OUT	020
STR	400
AND	003
OUT	021

(400 used as auxiliary relay)

Refer to 9-6 "Description of application instruction" for detail of the F-30 (MCS) and F-31 (MCR) instruction.

(Ex-2)

With the relay board ladder chart of (a), current flows to 002 from both 000 and 001 so that it performs the action identical to the ladder chart that revised for execution with the programable controller program (b).

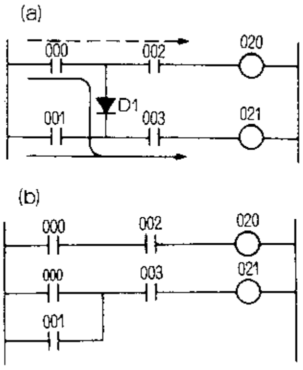


Instru.	
STR	000
STR	001
AND	002
OR STR	
OUT	020
STR	000
AND	002
OR	001
OUT	021

For the programmable controller, it does not affirm the theory that the current flows to one contact symbol from both lines as in 002 shown in (a) above. Because execution takes place serially from the address 0 to the END instruction with the programmable controller, it is not possible to go through the same contact symbol on the ladder chart twice.

(Ex-3)

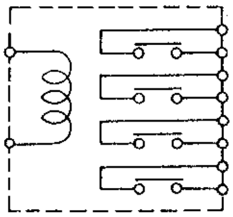
On account of the one-way control diode D1 in the relay board circuit of (a), the current does not from to 002 from 001, so that it performs the action as in the ladder chart that revised to the programmable controller application.



Instru.	
STR	000
AND	002
OUT	020
STR	000
OR	001
AND	003
OUT	021

It is not possible for the programmable controller to program the one-way control diode D1 as in (a).

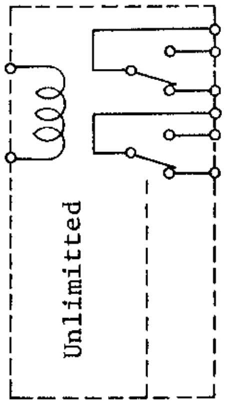
Three examples given are often used for the relay board because it permits use of relay with less contacts and simple wiring in the relay board. On the other hand, there is no need of paying attention for number of contacts with the programmable controller as it has the data memory which has unlimited number of contacts, and, it is preferable to have the ladder chart designed that can be recognized by any one.



With the relay method, it needs the following every time one contact is increased.

① Addition of a relay
② Addition of wirings

space
Cost
Reliability



The data memory of the programmable controller does not affect space, costs, and reliability even if more contacts are used because it can be a relay that have unlimited number of contacts.

[2] Input and output batch processing

As alreay discussed in 8-4 "Operational cycle", the W10 performs data exchange between the data memory and the I/O unit called the I/O processing in each machine scan cycle. During the I/O processing, the I/O units connected to the base unit are scanned from younger number.

① If it is the input unit, the ON/OFF state of the external contact connected to the input unit is read into the data memory.

② If it is the output unit, the ON/OFF state of the respective data memory is transferred to the latch of the output unit.

After completing the above operation to all input and output units connected to the base unit in the I/O processing, it goes into execution of the user program.

Because processing to the I/O units are performed in the batch mode, there would be a need of paying attention to the following points in making ladder design.

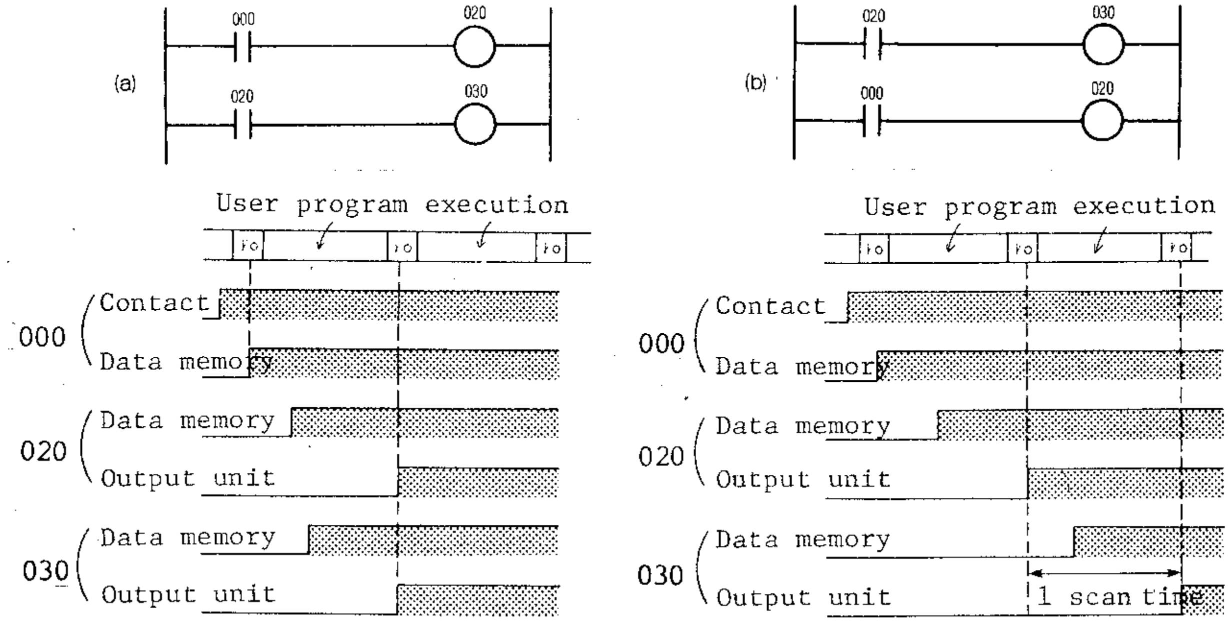
(1) Change in ON/OFF state of the external contact is read into the data memory at every scan once in the I/O processing. Therefore, a change in the ON/OFF state of the external device during execution of the user program may not cause to change the data memory contents assigned to the input in that scan cycle. So, the "input racing phenomenon" discussed in 8-4[3](7) "User program processing" does not occur.

(2) As the resultant ON/OFF state is written from the data memory to the output unit only once in a scan cycle of the I/O processing, the result is outputted to the output unit during the I/O processing of a next scan cycle.

[3] Influence by the sequence of programming

The programmable controller perform operation serially from the top and to the end of the program and repeats the same cycle (cyclic scan method).

(1) Modifying a program sequence may cause to have a different action.

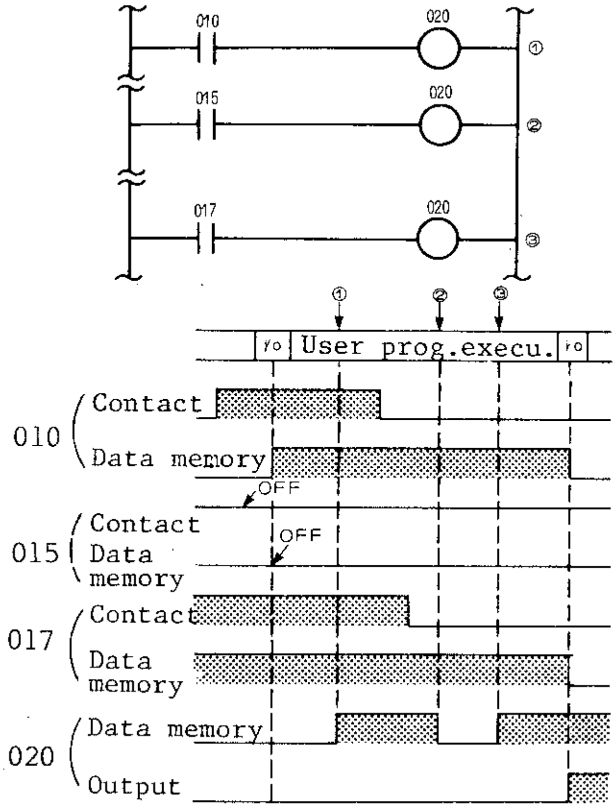


When the input 0( ) turns ON with the program of (a), the outputs 020 and 03( ) turn ON in the same scan cycle, but 030 turns ON with a delay of one scan cycle with (b).

In case the auxiliary contact of the coil is used, it should be programmed considering "Change of state of the auxiliary contact written before the coil becomes effective in a next scan cycle after the change in the state of the coil.

(2) Double use of the coil

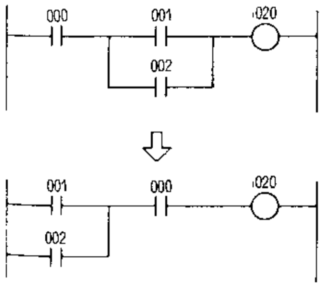
When the coil of the same relay number is used for a multiple times, the contents of the data memory changes according to the program and the newest result is written into the output unit from the data memory.



[4] Simplifying program

Program may be simplified when the circuit is revised depending on the sequence circuit.

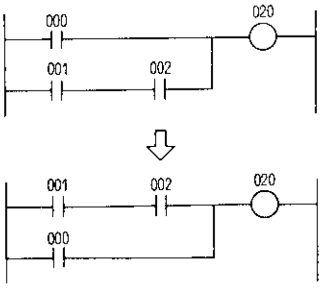
(Ex-1)



instruction	
STR	000
STR	001
OR	002
AND STR	
OUT	020

instruction	
STR	001
OR	002
AND	000
OUT	020

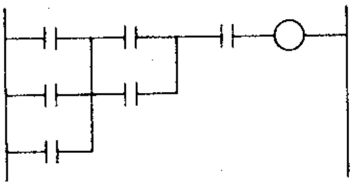
(Ex-2)



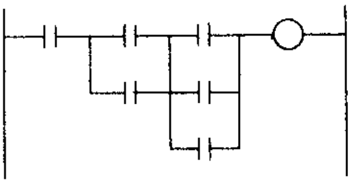
instruction	
STR	000
STR	001
AND	002
OR STR	
OUT	020

instruction	
STR	001
AND	002
OR	000
OUT	020

Generally, the program may be simplified when programmed with more weight on the left side.



Circuit with weight
on the left side

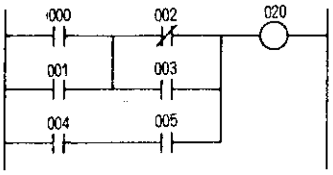


Circuit with weight
on the right side

[5] Programming serial/parallel circuit

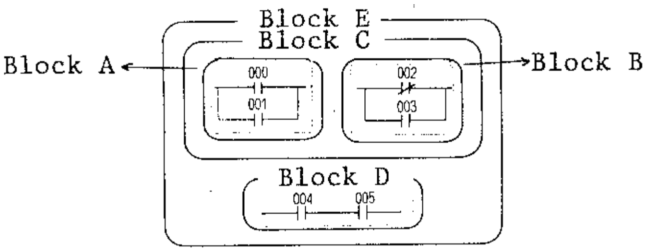
Prior to programming the serial/parallel circuit, it should be divided and programmed in minor blocks, then it should be assembled into a major program.

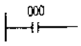
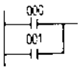
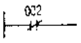
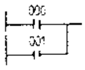
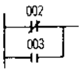
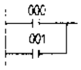
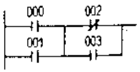
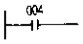
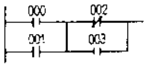
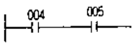
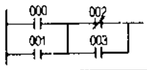
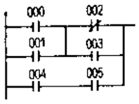
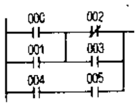
(Ex-1)



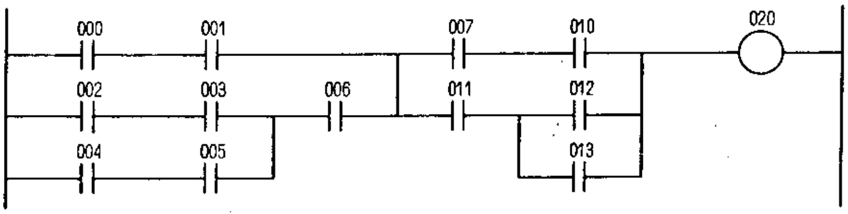
Instruction	
STR	000
OR	001
STR NOT	002
OR	003
AND STR	
STR	004
AND	005
OR STR	
OUT	020

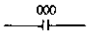
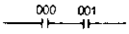
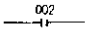
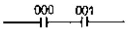
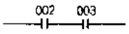
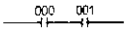
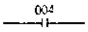
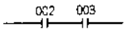
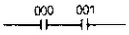
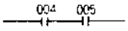
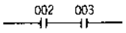
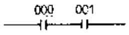
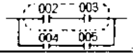
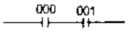
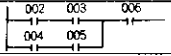
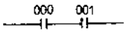
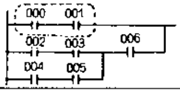
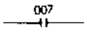
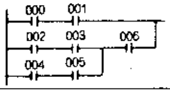
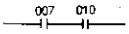
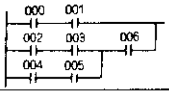
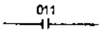
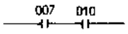
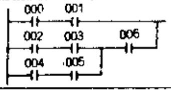
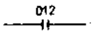
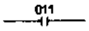
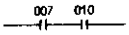
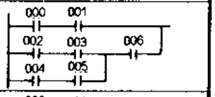
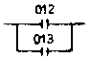
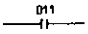
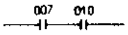
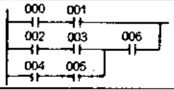
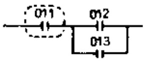
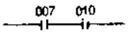
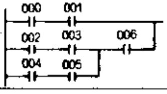
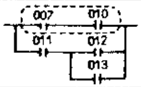
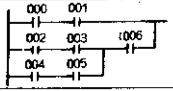
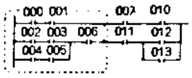
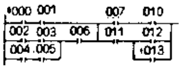
⇩



	Instruction		Accumulator ACC	Stack register S_1
A	STR	000		State of the ACC immediately before
	OR	001		
B	STR NOT	002		
	OR	003		
C	AND STR			
D	STR	004		
	AND	005		
E	OR STR			
	OUT	020		

(Ex-2)



Instr.	Accumulator ACC	Stack register S₁	S₂	S₃
STR 000		(*)		
AND 001				
STR 002				
AND 003				
STR 004				
AND 005				
OR STR				
AND 006				
OR STR				
STR 007				
AND 010				
STR 011				
STR 012				
OR 013				
AND STR				
OR STR				
AND STR				
OUT 020				

(*): State of the accumulator immediately before

9-5 Application instruction cautionary notes

[1] Numerical expression

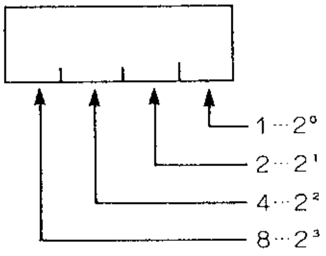
(1) Binary code

Number 0 through 9 is used with the decimal notation system we use. But, for the logical processing, only 0 (OFF) and 1 (ON) are used in representing a numerical value. This method is called the binary notation.

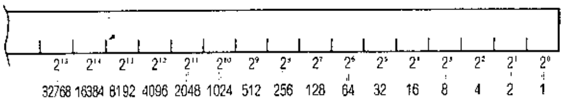
With the decimal notation, it needs to carry the digit once after a number increases from 0, 1, ... to 8, and 9. But, with the binary notation, it changes to 10 after a number changes from 0 to 1. 10 is used to represents the number 2 in the decimal notation. Similarly, it changes (carry) from 11 to 100, 111 to 1000, and so on.

Decimal number	0	1	2*	3	4*	5	6	7	8*
Binary number	0	1	10	11	100	101	110	111	1000

Carry takes place at the point indicated with an asterisk (*). So, each digit has a weight shown next.



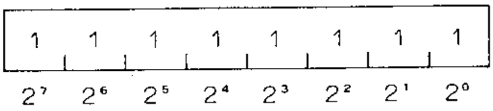
Similarly, each digit has a weight of 2^n.



Each digit of a binary number is called a bit.

Register of the W10 comprises 8 bits.

Now, let us examine the state when all eight bits are 1.



When weights of all bits are summerized, it establishes the following:

$$2^0+2^1+2^2+2^3+2^4+2^5+2^6+2^7 = 1+2+4+8+16+32+64+128 = 255$$

Whereas, eight bits can represents a decimal number of 0 through 255.

(2) Binary coded decimal (BCD)

With a decimal number, carry occurs every time it goes beyond 9. That which carry is applied to the binary number is called the binary coded decimal.

Decimal	Binary	BCD		
0	0		0	
1	1		1	
2	10		10	
3	11		11	
4	100		100	
5	101		101	
6	110		110	
7	111		111	
8	1000		1000	
9	1001		1001	Carry
10	1010	1	0000	
11	1011	1	0001	
12	1100	1	0010	
99	1100011	1001	1001	

With the BCD notation, each four bits are separated. Combination above 1001 such as 1010 is not permitted, carried on to a higher group of four bits, in order to represent a decimal equivalent.

(3) Octal notation and hexadecimal notation

In the programmable controller, all numbers are treated in a form of a binary or BCD. However, it permits programming or monitoring using a decimal number by means of the binary/decimal converter implemented in the programmer, so as to avoid disturbance during key entry and weight calculation in the binary mode.

Notwithstanding, it becomes often preferable to use another way of numerical representation with which you can make direct recognition of bit pattern, when considered in terms of bit operation of the programmable controller. For this purpose, the octal notation and hexadecimal notation are often used by the programmable controller and computer.

a. Octal notation

A carry occurs with the decimal notation when it changes from 9 to 10. Or, when it changes from 1 to 10 with the binary notation. In the octal notation, a carry occurs when it changes from 7 to 10.

Decimal	Binary	Octal	
0	0	0	
1	1	1	
2	10	2	
3	11	3	
4	100	4	
5	101	5	
6	110	6	
7	111	7	Carry
8	1000	10	
9	1001	11	
10	1010	12	
11	1011	13	
12	1100	14	
13	1101	15	
14	1110	16	
15	1111	17	Carry
16	10000	20	
≈	≈	≈	
62	111110	76	
63	111111	77	Carry
64	1000000	100	
65	1000001	101	

That is, it is not 8 that follows 7, but 10 is established. Similarly, it changes to 20 immediately following 17 and 100 after 77.

° Octal number vs binary number

0 through 7 is represented by three digits in the binary mode and a carry occurs when it changes from 111 to 1000.

With an octal number, a single digit can represents a number 0 through 7, and a carry occurs when it changes from 7 to 10.

Because a carry occurs immediately following 7 for both the binary and octal number, a binary number separated in three digits can be used to represent one digit of an octal number.

Binary	Octal
0	0
1	1
10	2
≈	≈
110	6
111	7
1 000	10
1 001	11
≈	≈
1 110	16
1 111	17
10 000	20
≈	≈
111 110	76
111 111	77
1 000 000	100
≈	≈
1 111 111	177
10 000 000	200
≈	≈
11 111 111	377

Since the register consists of eight bits, it can be used to represent a number within a range of 0 through 377($_8$).

° Data memory address, system memory address, and program memory address are expressed in an octal number.

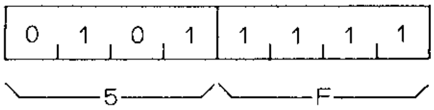
b. Hexadecimal notation

With the decimal notation, a carry occurs when it changes from 9 to 10. In the case of a hexadecimal number, a carry occurs when it changes from 9 → A → B → C → D → E → F, that is, when it changes from F to 10.

Dec.	Binary	Octal	Hex.
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	1 0
31	11111	37	1 F
32	100000	40	2 0
255	11111111	377	F F

° Hexadecimal number corresponds to binary number in the following way. That is, four bits of binary number is used to represent one digit of hexadecimal number.

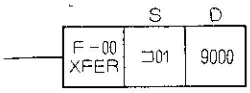
Binary number



[2] Source and destination

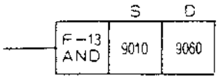
With the data processing instruction, data memory is handled in unit of byte. The register that contains the data before the operation is called the source (S) and the register in which the result is contained is called the designation (D).

(Ex-1)



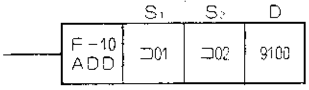
The contents of ⊐.01 (S) are transferred to 9000(D).

(Ex-2)



The contents of 9010(S) are ANDed with the contents of 9060(D) and its result is stored in 9060(D).

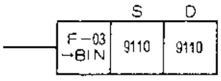
(Ex-3)



The contents of ⊐. 01(S_1) are added with the contents of . 02(S_2) and its result is stored in 9100(D).

The contents of the register on the source side does not change after the execution. As it is possible to use the same register for source and destination, it makes the contents of the source (destination, in other words) changed after execution.

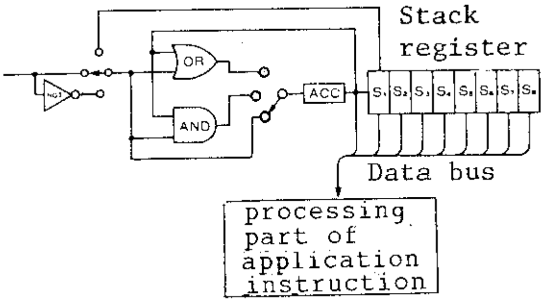
(Ex-4)



The contents of BCD 2-digit 9110(S) are converted into a binary code and its result is stored in 9110(D).

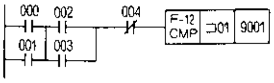
[3] Application instruction and stack register

With the W10, each instruction of TMR, CNT, and application instruction of F00 F71 is processed by a CPU. For those instructions, contens of the accumulator and stack register are sent to the CPU via the data bus, to be executed having those for operational conditions.

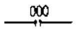
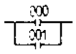
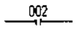
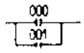
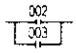
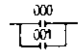
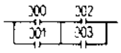
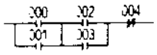


(Ex-1)

Application instructions, except CNT,F-60,and F-62, are executed having the contens of the accumulator for the oprational condition.

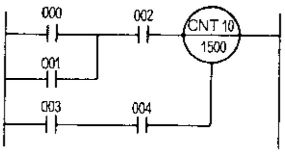


```
STR       000
OR        001
STR       002
OR        003
AND  STR
AND  NOT  004
F-12
          □01
          9001
```

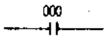
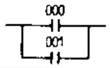
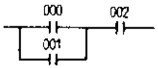
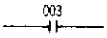
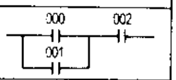
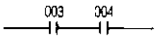
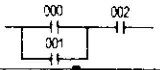
instruction	accumulater ACC	Stack reqister S.
STR 000		
OR 001		
STR 002		
OR 003		
AND STR		
AND NOT 004		
F-12	operation when condition is met	

(Ex-2)

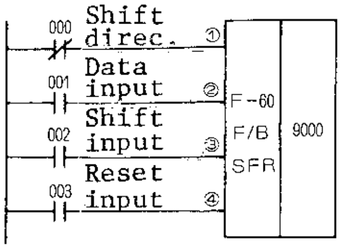
In the case of the CNT instruction



```
STR   000
OR    001
AND   002
STR   003
AND   004
CNT   10
      1500
```

Instruction	accumulator ACC	Stack reqister S.
STR 000		
OR 001		
AND 002		
STR 003		
AND 004		
CNT 10 1500	reset input	counter input

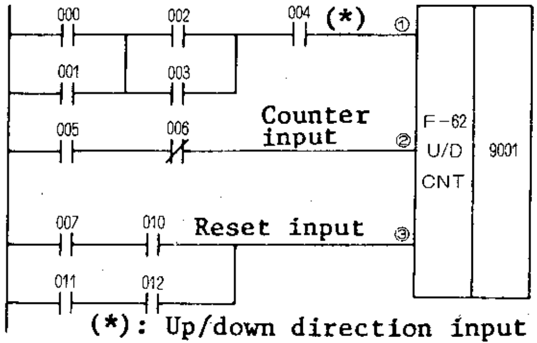
(Ex-3) With the F-60 instruction, the accumulator and the stack registers ($S_1 \sim S_3$) become the operational condition.



```
STR NOT  000
STR      001
STR      002
STR      003
F-60
         9000
```

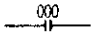
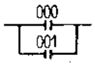
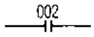
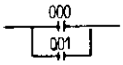
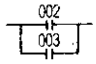
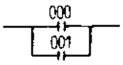
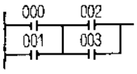
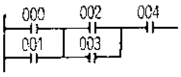
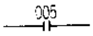
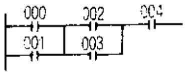
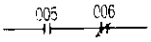
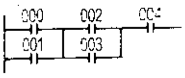
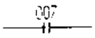
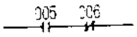
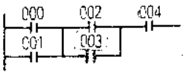
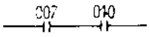
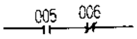
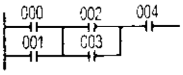
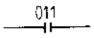
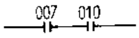
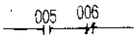
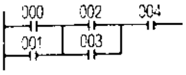
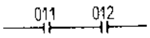
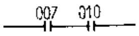
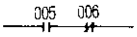
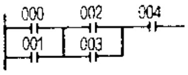
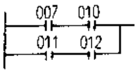
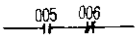
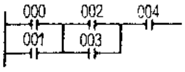
Instr.	Accumulator ACC	Stack register		
		S_1	S_2	S_3
STR NOT 000	000 ─┤/├─			
STR 001	001 ─┤├─	000 ─┤/├─		
STR 002	002 ─┤├─	001 ─┤├─	000 ─┤/├─	
STR 003	003 ─┤├─	002 ─┤├─	001 ─┤├─	000 ─┤/├─
F-60	Reset input ④	Shift input ③	Data input ②	Shift direction ①

(Ex-4) The contents of the stack register may a complicated serial/ parallel circuit.



(*): Up/down direction input

STR	000	STR	007
OR	001	AND	010
STR	002	STR	011
OR	003	AND	012
AND STR		OR STR	
AND	004	F-62	
STR	005		9001
AND NOT	006		

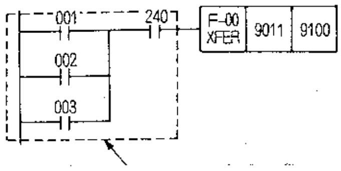
Instruction	Accumulator ACC	Stack register		
		S1	S2	S3
STR 000				
OR 001				
STR 002				
OR 003				
AND STR				
AND 004				
STR 005				
AND NOT 006				
STR 007				
AND 010				
STR 011				
AND 012				
OR STR				
F-62	Reset input ③	Counter input ②	Up/down direction input	

In this example, three register stacks, up to S_3, are used during operation of STR 0011.

[4] Operational condition

(1) Operational condition of an application instruction with which execution of the operation is determined may be a complicated serial/ parallel circuit, not limiting to a single contact ON/OFF condition. Refer to [3] "Application instruction and stack register".

(Ex)

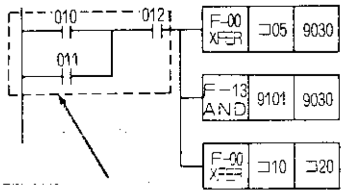


Operational condition

命令	
STR	001
OR	002
OR	003
AND	240
F-00	
	9011
	9100

(2) If the operational condition is the same, programming may be developed successively in

(Ex)



Operational condition

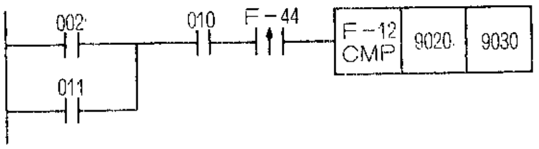
命令	
STR	010
OR	011
AND	012
F-00	
	⊐05
	9030
F-13	
	9101
	9030
F-00	
	⊐10
	⊐20

(3) There are following two kinds of processing for application instruction when the operational condition is satisfied. (F-30, F-31, F-40, F-41, F-42, and F-43 are exceptions.)

①	Those which operation takes place at each scan cycle, so long as the operational condition is met.	F-12 (compare register with register) Fc12 (compare register with constant)
②	Those which operation takes place only at the first scan cycle that the operational condition is met.	Data processing instructions except the above.

With the instruction of the group ② above, the ON/OFF state of the operational condition in the preceding scan cycle is compared with the ON/OFF state of the operational condition at the current scan cycle. If the preceding state is OFF and the current state is ON, the operation takes place as if the operational condition changed from OFF to ON .

NOTE-2: In case the operation must be executed only when there is an OFF to ON transition in the operational condition with respect to F-12,F 12,the F-44 instruction (rise edge differentiate instruction) must be used.



(4) If the operational condition is not satisfactory, no operation takes place and the register on the destination side remain unaffected. It includes scan cycles after ON for the instruction that the operation is executed only when the operational condition changes from OFF to ON. In the case of the instruction that affects the flag, it makes the flag cleared.

NOTE-3: Refer to 9-5[5] "Data processing instruction and flag" for detail of flag.

(5) With the instruction of the group ② above, by the differentiate instruction, the operational condition which changed from OFF to ON is detected. There are 510 bits for the differentiate instruction altogether. Every instruction uses, bit, so the total instruction of the group ② must be assigned less than 510 bits. W10 checks the total number of differentiate memory at the start of operation, and "Memory failure" displays if there are more than 511 bits.

[5] Data processing instruction and flag

(1) Kinds of flags

Flag is used to reflect the operational result. There are four kinds of flags for the W10; non-carry, error, carry, and zero flags. They are assigned to four bits of data memory, 654 ∿ 657.

Non-carry flag	Error flag	Carry flag	Zero flag
654	655	656	657

(2) Instruction that affects flag

The following seventeen instruction have flag set active according to the operational result.

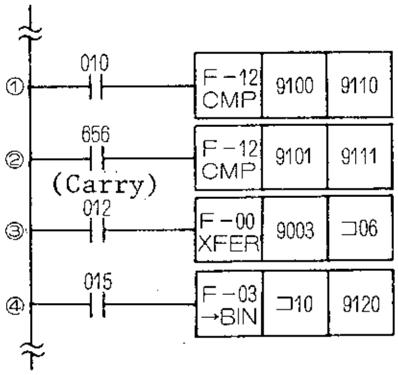
1	F-03	Convert BCD (2-digit) to binary
2	F-10	Add (BCD) register with register
3	Fc10	Add register with BCD constant
4	F-11	Subtract (BCD) register from register
5	Fc11	Subtract register by BCD constant
6	F-12	Compare registers
7	Fc12	Compare register with constant
8	F-60	Shift register bidirectional
9	F-62	Set BCD up/down counter

Instructions mentioned other than above do not affect flag.

(3) Transition of flag state in the scan cycle

① Prior to processing of the user program at each scan cycle, flags are cleared. Refer to 8-4 "Operation cycle".

Provided that there are no flag affecting instructions in preceding steps.



Provided that there are no flag affecting instructions in steps to follow.

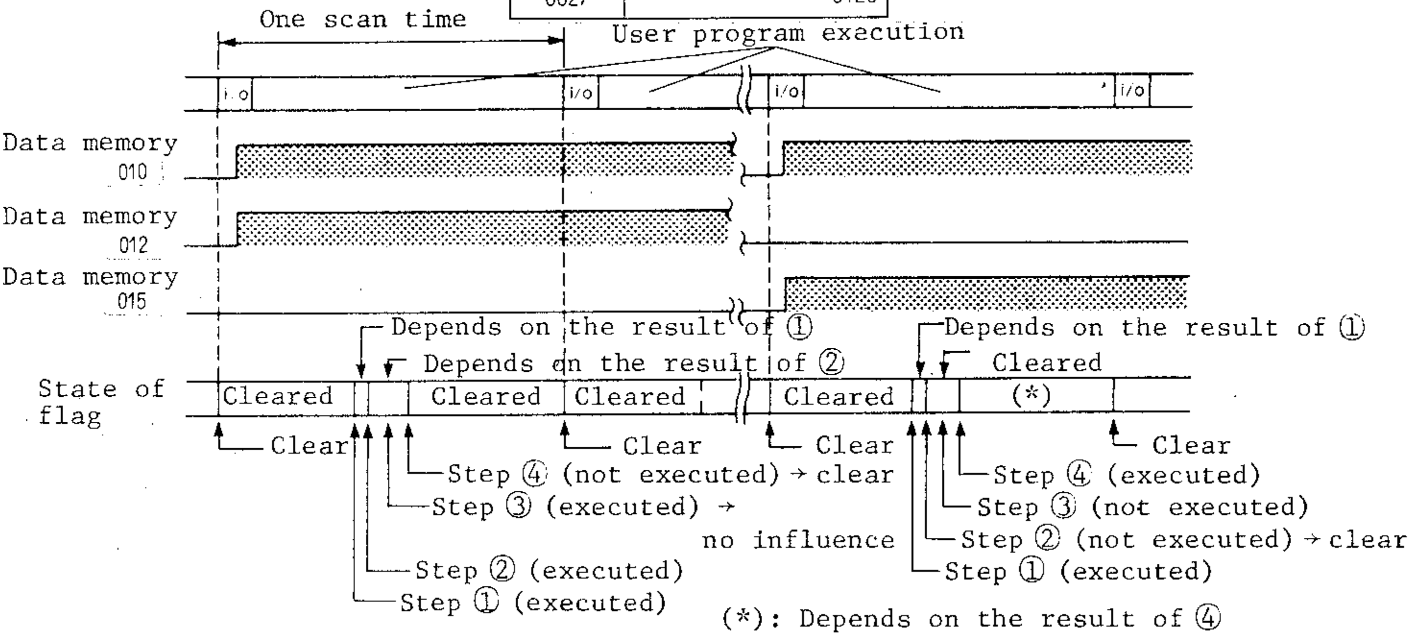
② When it goes into processing of the flag affecting instruction

a. The flag is set according to the operational result of the instruction, when the operational condition for that instruction has been met.

b. Flag is cleared when the operational condition is not satisfactory for that instruction.

③ In the execution of no flag affecting instruction, the state of flag is not changed, whether the operation took place or not.

Addr.	Instruction		
0010	STR	010	①
0011	F−12		
0012		9100	
0013		9110	
0014	STR	656	②
0015	F−12		
0016		9101	
0017		9111	
0020	STR	012	③
0021	F−00		
0022		9003	
0023		⊐06	
0024	STR	015	④
0025	F−03		
0026		⊐10	
0027		9120	



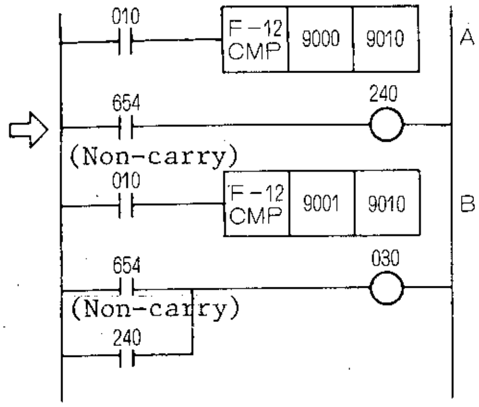
(*): Depends on the result of ④

(4) How to retain flag

Flag thus set according to the operational result may be changed or cleared by the execution of the flag affecting instruction in that scan cycle. When it goes into a next scan cycle, the flag is cleared before execution of the user program. In case the flag has to be retained of its current state, you must write the state of the flag into the coil such as the auxiliary relay immediately following the respective instruction.

This makes the flag retained of its state until the execution of the respective instruction in the next scan cycle.

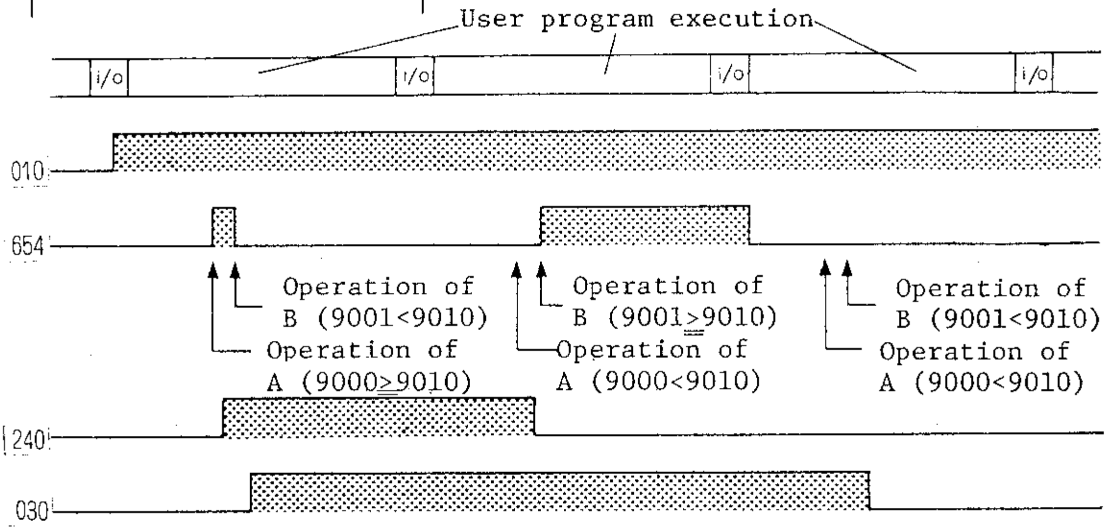
(Ex-1) Program that set 030 ON when 9000 ≥ 9010 or 9001 ≥ 9010.



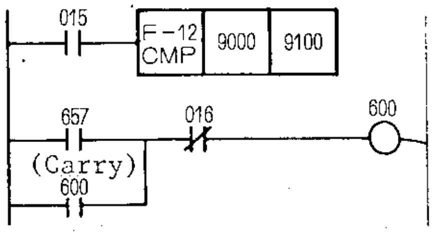
240 is set ON if 9000 ≥ 9010 after the operation of A.

The non-carry flag set in A changes according to the result at B.

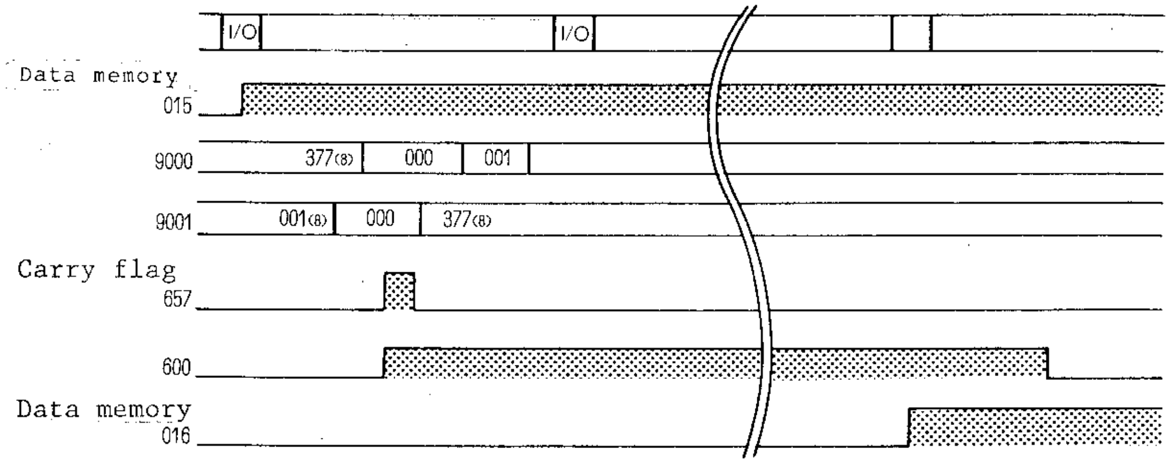
After the operational result of A 240 is ORed with the result of B.


User program execution
i/o
i/o
i/o
i/o
010
654
Operation of B (9001<9010)
Operation of A (9000≥9010)
Operation of B (9001≥9010)
Operation of A (9000<9010)
Operation of B (9001<9010)
Operation of A (9000<9010)
240
030

When the state of flag is monitored by the support tool such as the programmer or displayed externally, it is not possible to visually check the state of the flag because the state of the flag can only be retained for a single scan cycle by merely storing the state of the flag in the coil like in Ex-1. In this event, there occurs a need of making the flag self-retained.



The state of the carry flag (657) immediately after the operation of ① can be self-retained until 016 is set ON.



[6] Double length operation

(1) The following six instructions has the function to permit operation of more than two bytes.

① F-10 (add register with register)

② Fc10 (add register with BCD constant)

③ F-11 (subtract register from register)

④ Fc11 (subtract register by BCD constant)

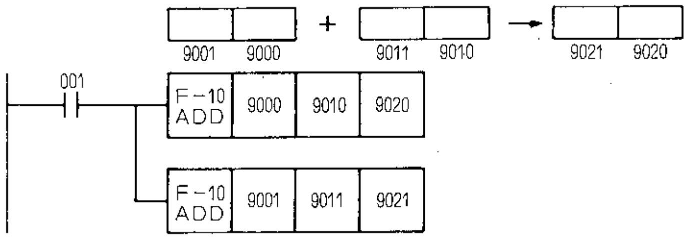
⑤ F-12 (compare registers)

⑥ Fc12 (compare register with constant)

(2) Programming double length operation

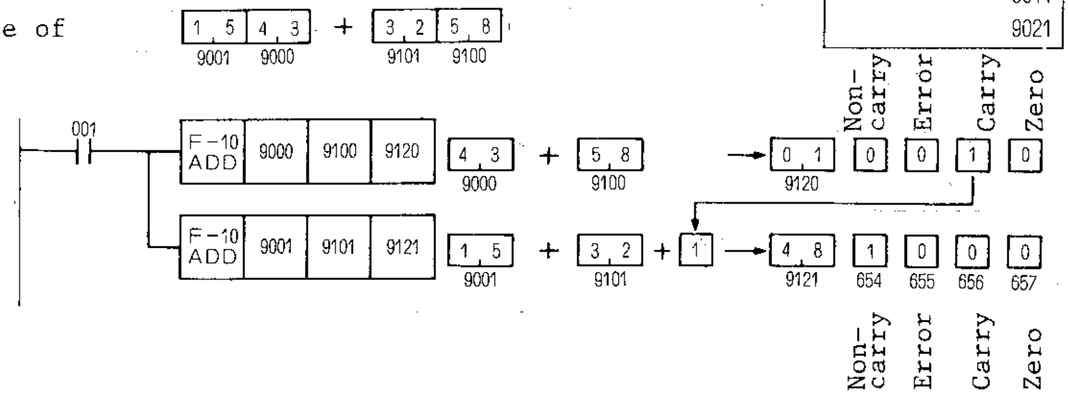
With the double length operation, a carry over or a carry down occured in a lower order digit is refected in processing the next digit. It should be programmed from a low order digit in succession to the operational condition.

(Ex-1)



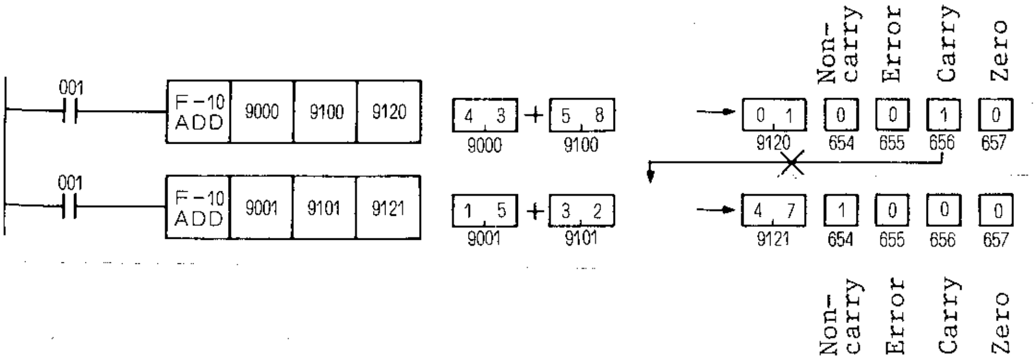
```
Instruction
STR        001
F-10
          9000
          9010
          9020
F-10
          9001
          9011
          9021
```

In case of

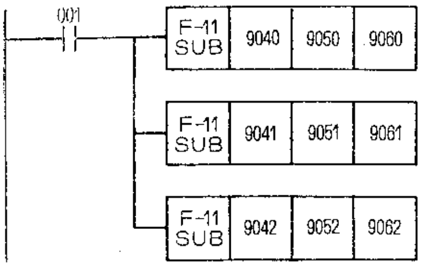


The carry flag in a low order digit is added to a next high order digit during the operation.

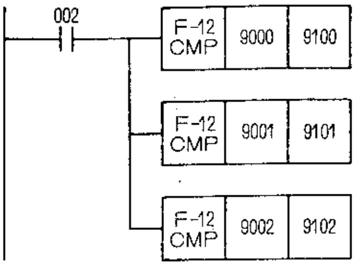
REFERENCE: Programming in the following way would not perform the double length operation.



(Ex-2) Double length operation above three digits is also possible in the similar manner.



(Subtraction of BCD 6 digits)



(Comparison of three bytes data)

(3) Internal processing during the double length operation

① For operation of F-10 (Fc10), F-11 (Fc11), or F-12 (Fc12) instruction that appears first after the operational condition, the operation takes place without including the state of the flag before that.

② If there is a F-10 (Fc10), F-11 (Fc11), or F-12 (Fc12) instruction encountered during execution of the common operational condition, the following operation takes place.

a. Execution takes place including the state of the carry flag immediately before.

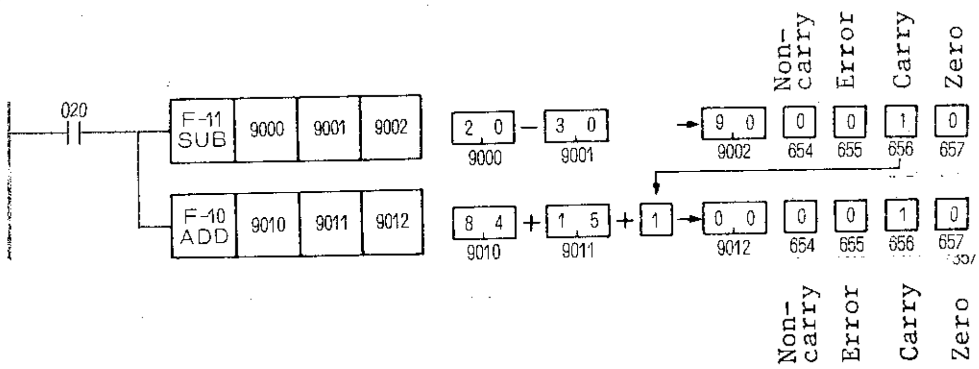
b. For the zero flag, the state of the zero flag immediately before is ANDed with the state of the zero flag after the operation of the respective instruction. If both are 1, it makes the zero flag set.

F-10 (Fc10)	Adds the state of the carry flag immediately before
F-11 (Fc11)	Subtracts the state of the carry flag immediately before
F-12 (Fc12)	Subtracts the state of the carry flag immediately before

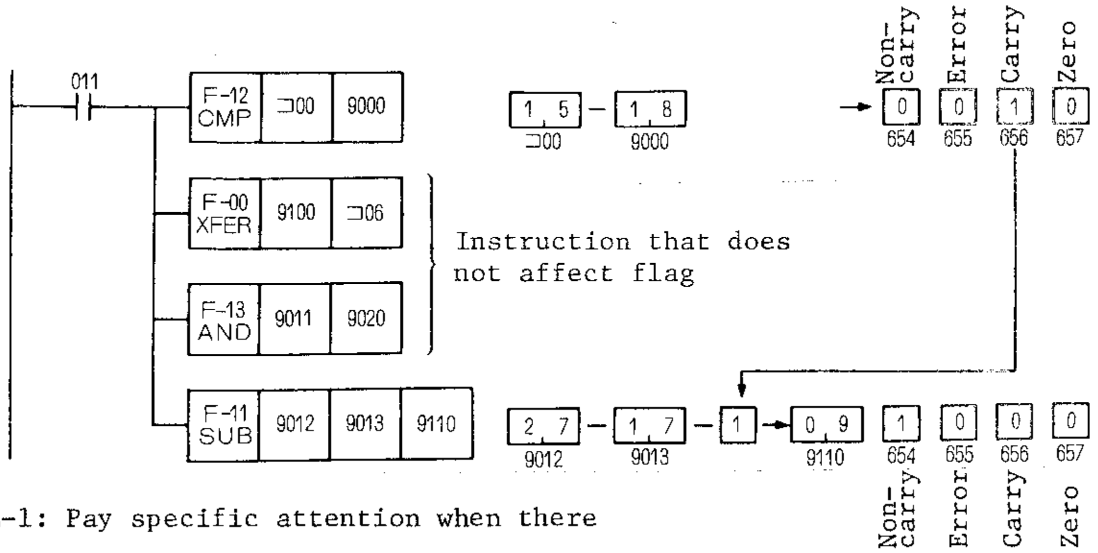
NOTE-1: The F-12 (Fc12) instruction perform the operation of S_1-S_2 or S_1-n and its result is stored in the flag.

(4) Cautionary notes

① Operation that includes the flag takes place for a different instruction, if F-10 (Fc10), F-11 (Fc11), or F-12 (Fc12) instruction has been programmed in a form of common operational condition.



② The double length operation takes place even if there is no flag affecting instruction between the F-10 (Fc10), F-11 (Fc11), and F-12 (Fc12) instruction.



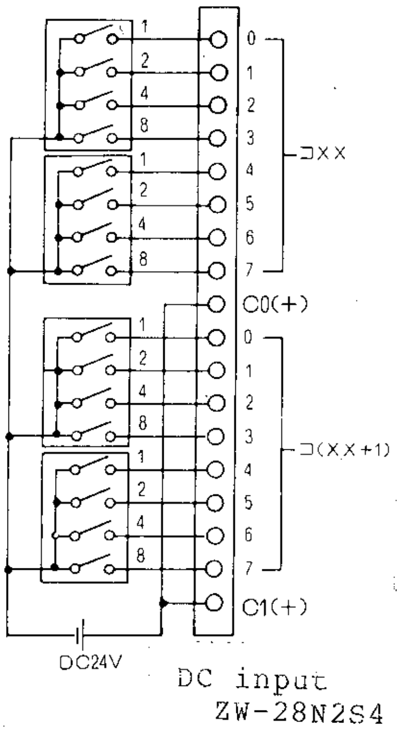
NOTE-1: Pay specific attention when there are many instructions involved.

[7] Numerical signal input/output method

Shown next are examples to make connection of the external device with which numerical signal is sensed from such as the digital switch for processing by the W10 or the operational result is displayed on the numerical display unit.

(1) Input of numerical signal

a. Connection with external device



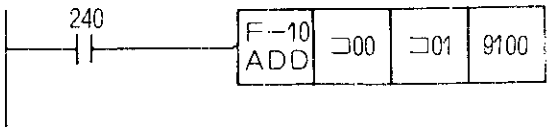
- When the unit which has DC input type (ZW-28N2S4 and etc) is used for the input unit, it reads four digits of BCD signal per unit.
- The real code digital switch must be used.

Weight \ No.	0	1	2	3	4	5	6	7	8	9
1		●		●		●		●		●
2			●	●			●	●		
4					●	●	●	●		
8									●	●

● — Dot represents the ON state of the switch.

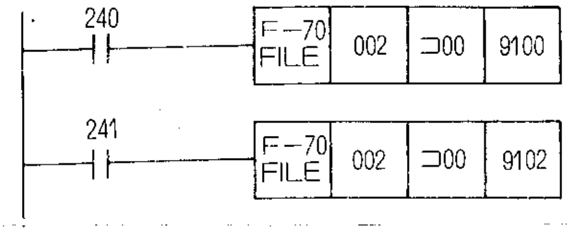
- Shown above is the connection with the external device to read the signal into the data memory I/O relay area during the I/O processing at each scan cycle. The 16-bit data can be directly specified by means of the data processing instruction for one byte (8 points) of コ.xx and another one byte (8 points) of コ(xx +1).

(Ex)



BCD 2 digits of コ00 is added with BCD 2 digits of コ01 and its result is stored in the register 9100. After transferring it to the register area using the transfer instruction, it may be used for the data processing instruction.

(Ex)

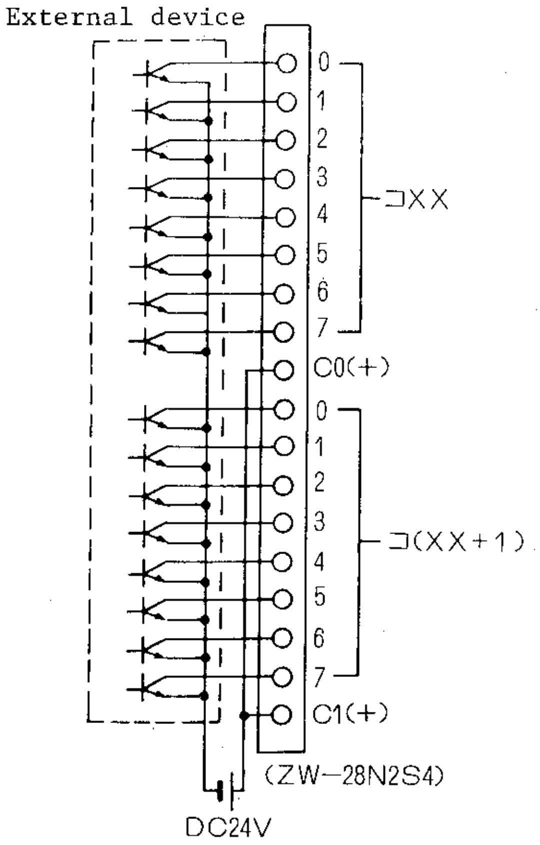


° When 240 is set ON, two bytes (BCD 4 digits) of ⊐ 00 and ⊐ 01 are transferred to 9100 and 9101, respectively.

° When 241 is set ON, two bytes (BCD 4 digits) of ⊐ 01 and ⊐ 00 are transferred to 9102 and 9103, respectively.

In the above example, a multiple number of preset values are read with a pair of digital switches.

b. Connection with the open collector output type external device



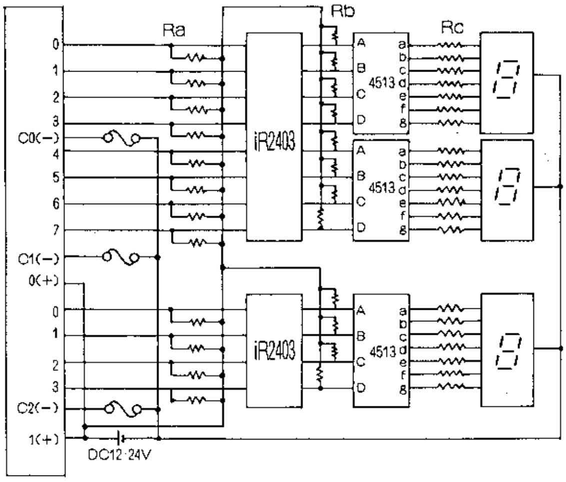
° When the ZW-28N2S4 DC input type is used for the input unit, it reads 16-bit numeric signal per unit.

° In the above connection, the signal is read into the I/O relay area of the data memory at each scan cycle during the I/O processing. Similar as in the case of the digital switch, it should be used in unit of one byte by the data processing instruction.

(2) Output of numerical signal

a. Connection with the numerical display unit (1)

Output unit
ZW-28M122



° Use the cathode common type to use the 7-segment LCD numerical display unit.

° Use the CMOS, MC4513 or its equivalent for the decoder driver IC which has the following characteristics:

V_{DD} ----DC12~18V
V_{SS} ----0V
LE ----0V
RBI ----0V
$\overline{BI}$ ----Same voltage level as V_{DD}
$\overline{LT}$ ----Same voltage level as V_{DD}

° Ra is the pullup resistor of 5 to 10 Kohms.

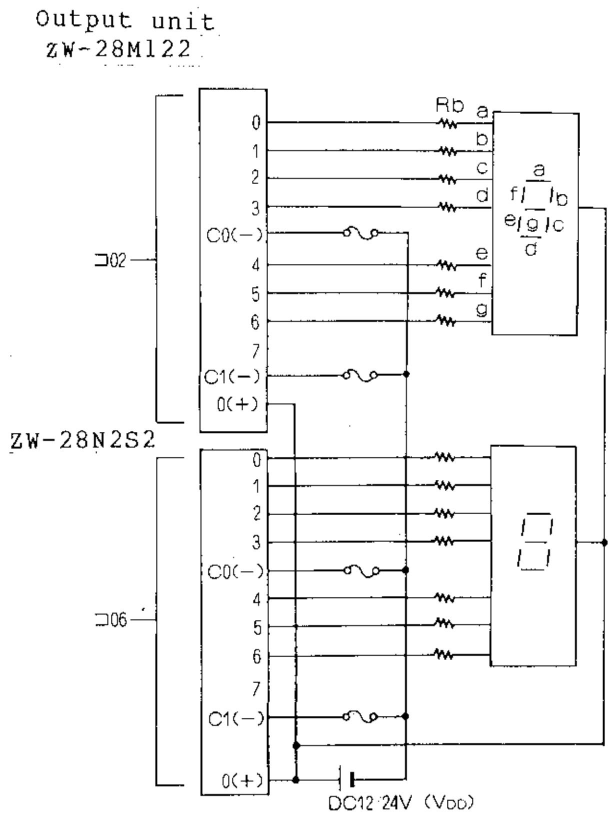
° Rc is the current limiting resister of which rating should be obtained on the basis of I_{FmAX} , V_F of the LED numerical display unit used.

$$Rc = \frac{V_{DD}-V_F}{1_{FmAX}}$$

On accout of the current limit by 4523, it should be I_F < 25mA .

° Use iR2403 or its equivalent for the driver IC.

b. Connection with the numerical display unit (use of the F-52 instruction)



- ° Use of the F-52 instruction (7SEG decode) enables you to achieve simple wiring connection with the numerical display unit.
- ° One-digit number can be displayed when the ZW-28N2S2 is used for the output unit.
- ° Use the anode common type for the 7-segment LED display unit.
- ° Rb is the current limiting register of which rating should be obtained using the following equation.

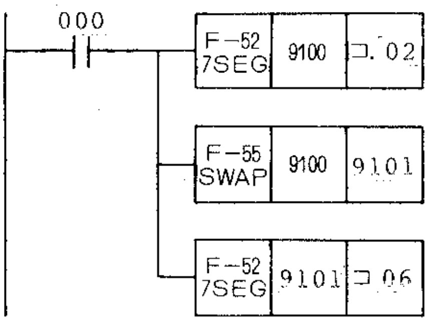
$$Rb = \frac{V_{DD} - V_F - V_{ON}}{I_{FmAX}}$$

V_{DD} —— Supply voltage

V_F —— Forward voltage of the LED numerical display unit

V_{ON} —— Output unit ON voltage (to be computed in terms of 1V)

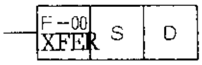
° It should be programmed in the following manner to display one-byte, two-digit BCD number.



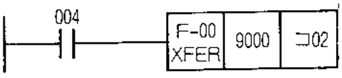
- ° Convert the low order 4 bits of the register 9100 (low order one digit out of BCD 2 digits) into the 7-segment data and be outputted to ⊐ 02 .
- ° Swap the high order 4 bits of the register 9100 with the low order 4 bits, then store the result in the register 9101 .
- ° Convert the low order 4 bits of the register 9101 (high order one digit out of BCD 2 digits) into the 7-segment data and be outputted to ⊐ 06 .

9-6 Description of application instruction

F-00 XFER Transfer one byte data

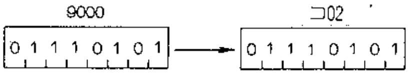
Symbol		
Function		The contents of the register S (one byte data) are transferred to the register D.
Operation		S→D
Range of S applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177
Range of D applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of signal (OFF to ON)
After the operation	Contents of S	Unchanged
	Contents of D	Contents of the register S
	Flag	Unchanged

[Explanation]



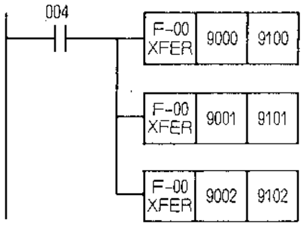
Instr.	
STR	004
F-00	
	9000
	⊐02

When the input condition 004 changes from OFF to ON, the contents of the register 9000 are transferred to the register ⊐.001.



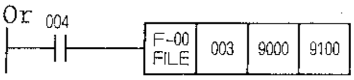
NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

REFERENCE: To transfer more than two bytes in succession, either repeat the F-00 instruction as shown below or use the n-bytes batch transfer instruction of F-70.



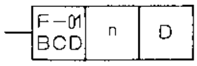
Instruction	
STR	004
F-00	
	9000
	9100
F-00	
	9001
	9101
F-00	
	9002
	9102

Or

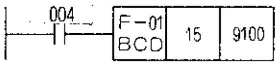


Instruction	
STR	004
F-70	
	003
	9000
	9100

F-01 BCD Transfer BCD constant

Symbol		
Function		A 2-digit BCD constant "n" is transferred to the register D.
Operation		n→D
Range of "n" applicable		00∿99
Range of D applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of signal (OFF to ON)
After the operation	Contents of D	n(00∿99)
	Flag	Unchanged

[Explanation]



Instr.	
STR	004
F-01	
	15
	9100

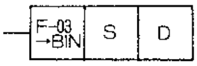
When the input condition 004 changes from OFF to ON, the BCD constant "15" is transferred to the register 9100.

The following value is contained in the register 9100 after the transfer.

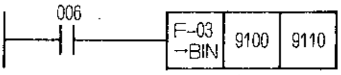
9100 | 0 0 0 1 | 0 1 0 1 |

NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

F-03 →BIN Convert BCD into binary

Symbol		
Function		The contents of the register S (8-bit data) are assumed as a BCD code, converted into a binary equivalent, then the result is stored in the register D.
Operation		S→D
Range of S applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177
Range of D applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of signal (OFF to ON)
After the operation	Contents of S	Unchanged
	Contents of D	°Operational result °Remains unchanged if the contents of the register S is not a BCD code.
	Flag	Cont. of reg.S: a b c d BCD: 0 0 0 0 If not BCD: 0 1 0 0 a:Non-carry 654 b:Error 655 c:Carry 656 d:Zero 657

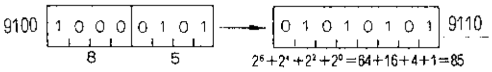
[Explanation]



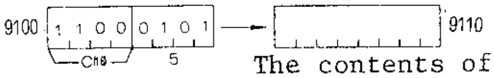
Instr.	
STR	006
F-03	
	9100
	9110

° When the input condition 006 changes from OFF to ON, the contents of the register 9100 (8-bit data) are assumed as a BCD code, converted into a binary equivalent, and its result is transferred to the register 9110. The contents of the register 9100 remains unchanged. If the contents of 9100 is other than BCD code, the contents of the register 9110 stay the same as before and the error flag (655) is set to "1".

° Transition of register contents and flags



Non-carry 654	Error 655	Carry 656	Zero 657
0	0	0	0



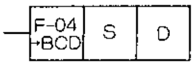
The contents of 9110 does not change.

Non-carry 654	Error 655	Carry 656	Zero 657
0	1	0	0

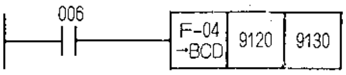
NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

NOTE-2: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met. Refer to 9-5[5] "Data processing instruction and flag".

F-04 →BCD Convert binary into BCD

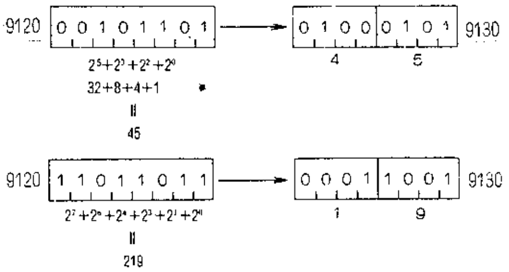
Symbol		
Function		The contents of the register S (8-bit data) are assumed as a binary code, converted into a BCD equivalent, then the result is stored in the register D.
Operation		S→D
Range of S applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177
Range of D applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of signal (OFF to ON)
After the operation	Contents of S	Unchanged
	Contents of D	Operational result
	Flag	Unchanged

[Explanation]



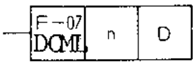
Instr.	
STR	006
F-04	
	9120
	9130

° When the input condition 006 changes from OFF to ON, the contents of the register 9320 (8-bit data) are assumed as a binary code, converted into a BCD equivalent, and its result is transferred to the register 9130. The contents of the register 9120 remains unchanged. If the BCD number converted should exceed "100", the value above "100" will be disregarded.

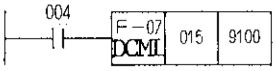


NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

F-07 DCML Transfer decimal constant

Symbol		
Function		A decimal constant "n" is transferred to the register D.
Operation		n→D
Range of "D" applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of signal (OFF to ON)
After the operation	Contents of D	n(000~255)
	Flag	Unchanged

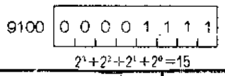
[Explanation]



Instr.	
STR	004
F-07	015
	9100

° When the input condition 004 changes from OFF to ON, the decimal constant 15 is transferred to the register 9100.

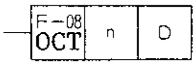
° The register 9100 is in the following binary code representation.



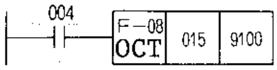
Range of "n" applicable	000 ~ 255

NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

F-08 OCT   Transfer octal constant

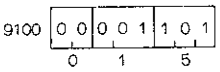
Symbol		
Function		An octal constant "n" is transferred to the register D.
Operation		n→D
Range of "n" applicable		000~377
Range of D applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of signal (OFF to ON)
After the operation	Contents of D	n(000~377)
	Flag	Unchanged

[Explanation]



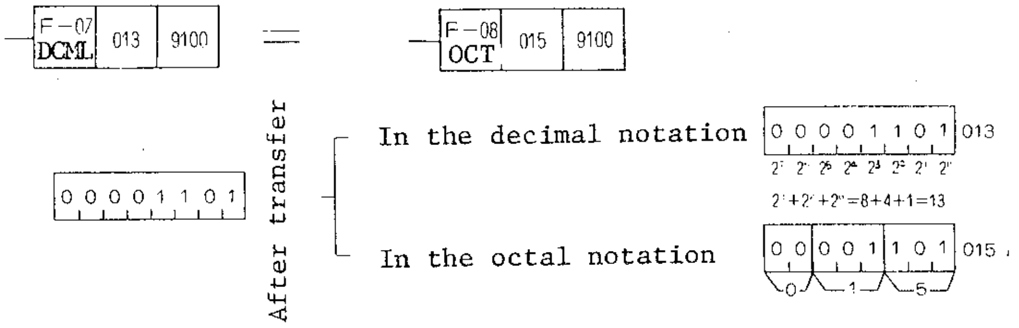
Instr.	
STR	004
F-08	015
	9100

- When the input condition 004 changes from OFF to ON, the octal constant 015 is transferred to the register 9100.
- The register 9100 is in the following code representation.

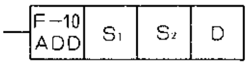


NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

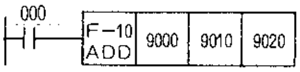
REFERENCE: Though the F-07 (transfer decimal constant) handles a decimal number and the F-08 (transfer octal constant) an octal number, the contents of the register after the transfer are represented in the binary code for both instructions.



F-10 ADD	Add registers (BCD 2-digit) (ADD)

<table>
<tr><td colspan="2">Symbol</td><td> </td></tr>
<tr><td colspan="2">Function</td><td>The contents of the register S_1 are added with the contents of the register S_2 in BCD 2-digit and its result is stored in the register D.</td></tr>
<tr><td colspan="2">Operation</td><td>$S_1+S_2 \rightarrow D$</td></tr>
<tr><td colspan="2">Range of S_1 applicable</td><td>□00~□71 (NOTE-1)
b000~b137
9000~9177</td></tr>
<tr><td colspan="2">Range of S applicable</td><td>□00~□71 (NOTE-1)
b000~b137
9000~9177</td></tr>
<tr><td colspan="2">Range of D applicable</td><td>□02~□71 (NOTE-1)
b000~b137
9000~9177</td></tr>
<tr><td colspan="2">Operational condition</td><td>At a rising edge of signal (OFF to ON)</td></tr>
<tr><td rowspan="4">After the operation</td><td>Contents of S_1</td><td>Unchanged</td></tr>
<tr><td>Contents of S_2</td><td>Unchanged</td></tr>
<tr><td>Contents of D</td><td>° Low order two digits of the operational result
° Unchanged when the contents of registers S_1 or S_2 are not BCD code.</td></tr>
<tr><td>Flag</td><td>
<table>
<tr><td>Result</td><td>a</td><td>b</td><td>c</td><td>d</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1~99</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>100</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>101 and above</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>When S_1 and S_2 are not BCD code</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
</table>
a:Non-carry 654
b:Error 655
c:Carry 656
d:Zero 657</td></tr>
</table>

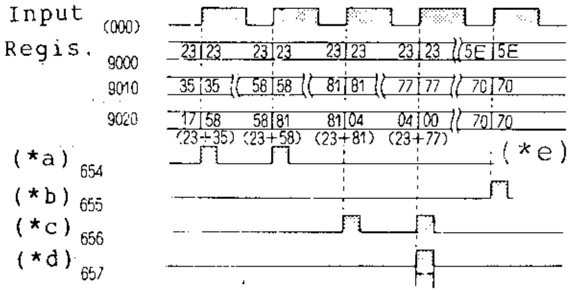
[Explanation]



Instr.	
STR	000
F-10	9000
	9010
	9020

° When the input condition 000 changes from OFF to ON, the contents of the register 9000 are added with the contents of the register 9010 and its result is stored in the register 9020. The contents of registers 9000 and 9010 remain unchanged.

° Transition of operational result and flags



Within the one scan time

Remains valid until the flag affecting instruction appears in the program.

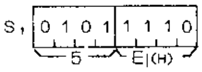
(*a):Non-carry flag
(*b):Error flag
(*c):Carry flag
(*d):Zero flag
(*e):This does not operated because the contents of 9000 are not in the BCD code.

NOTE-1: ⊐64∼⊐71 is the special area. Refer to 8-2 [3] "Special relay".

NOTE-2: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met. Refer to 9-5[5] "Data processing instruction and flag".

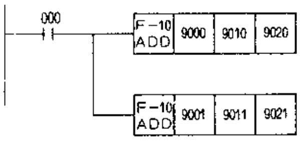
NOTE-3: If the contents of S_1 or S_2 were numbers other than BCD code, it makes the error flag (655) set active and no addition will be done.

(Ex)

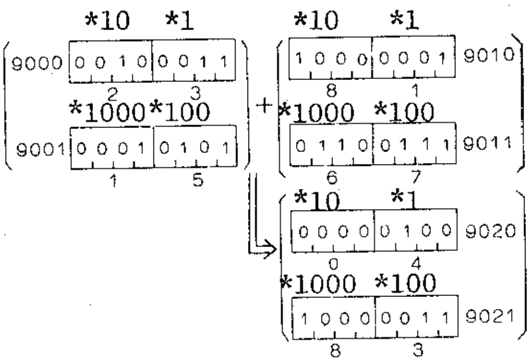


1110 is the code prohibited to use in the BCD system.

REFERENCE: In case more than BCD 3 digits has to be added successively, the F-10 instruction must be provided in continuation. When the F-10 instruction is programmed repeatedly, the contents of the carry flag (656) are also added after the second instruction. For the F-10 instruction that appears first in succession to the STR instruction, the contents of the carry flag (656) are not added.



Instruction	
STR	4000
F-10	
	9000
	9010
	9020
F-10	
	9001
	9011
	9021

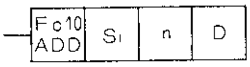


*1	: Ones order
*10	: Tens order
*100	: Hundreds order
*1000	: Thousands order

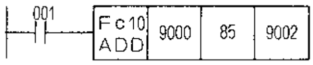
° The above example shows the case of 1523 + 6781 = 8304.

° If programmed from low order digit, the carry information will be carried on to a higher digit.
Refer to 9-5[6] "Double length operation".

Fc10 ADD Add register with 2-digit BCD constant (ADD)

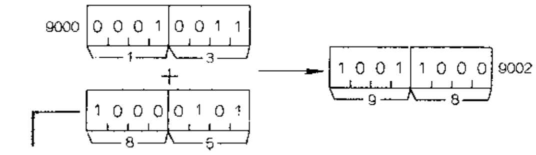
Symbol						
Function		The contents of the register S_1 are added with the 2-digit BCD constant "n" and its result is stored in the register D.				
Operation		$S_1+n \rightarrow D$				
Range of S_1 applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177				
Range of "n" applicable		00~99				
Range of D applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177				
Operational condition		At a rising edge of signal (OFF to ON)				
After the operation	Contents of S_1	Unchanged				
	Contents of D	° Low order two digits of the operational result ° Unchanged when the contents of the register S_1 is not BCD code.				
	Flag	Result	a	b	c	d
		0	1	0	0	1
		1~99	1	0	0	0
		100	0	0	1	1
		101 and above	0	0	1	0
		When S_1 is not BCD code	0	1	0	0
		a:Non-carry 654 b:Error 655 c:Carry 656 d:Zero 657				

[Explanation]



Instr.	
STR	001
Fc10	9000
	85
	9002

When the input condition 001 changes from OFF to ON, the contents of the register 9000 are added with the BCD constant 85 and its result is stored in the register 9002.
It operates in the same timing as the F-10 instruction.



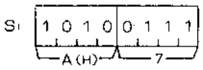
BCD constant 85

NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

NOTE-2: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met. Refer to 9-5[5] "Data processing instruction and flag".

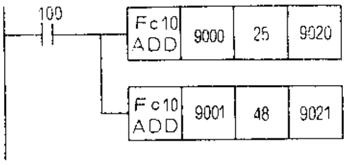
NOTE-3: If the contents of S_1 or S_2 were numbers other than BCD code, it makes the error flag (655) set active and no addition will be done.

(Ex)

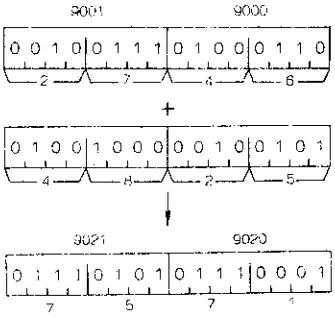


1010 is the code prohibited to use
in the BCD system.

REFERENCE: Similar as the F-10 instruction, it is possible to added more than three digits of BCD value.



Instruction	
STR	100
Fc10	
	9000
	25
	9020
Fc10	
	9001
	48
	9021



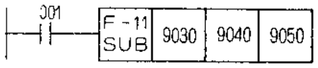
F-11 SUB Subtract register by register in BCD 2 digits (SUBtract)

Symbol		F-11 SUB | S_1 | S_2 | D
Function		The contents of the register S_1 are subtracted by the contents of the register S_2 in BCD 2 digits and its result is stored in the register D.
Operation		$S_1-S_2 \rightarrow D$
Range of S_1 applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177
Range of S_2 applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177
Range of D applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of signal (OFF to ON)
After the operation	Contents of S_1	Unchanged
	Contents of S_2	Unchanged
	Contents of D	° Operational result ° Not changed when the contents of registers S_1 or S_2 are not BCD code.
	Flag	

Flag	Result	a	b	c	d
	0	1	0	0	1
	1 99	1	0	0	0
	Negative value	0	0	1	0
	When S_1 or S_2 are not BCD	0	1	0	0

a:Non-carry	654
b:Error	655
c:Carry	656
d:Zero	657

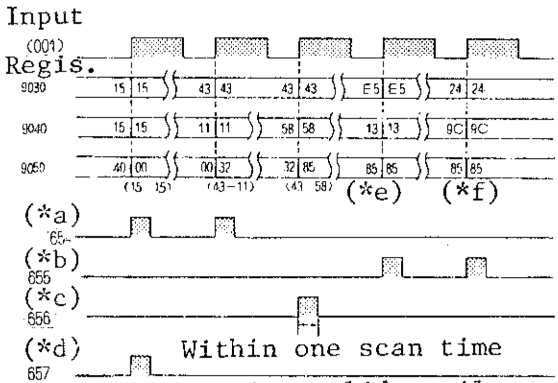
[Explanation]



Instr.	
STR	001
F-11	9030
	9040
	9050

When the input condition 001 changes from OFF to ON, the contents of the register 9030 is subtracted by the contents of the register 9040 and its result is stored in the register 9050. The contents of registers 9030 and 9040 remain unchanged.

Transition of operational result and flags



Remains valid until a flag affecting instruction is met in the program.

(*a):Non-carry flag
(*b):Error flag
(*c):Carry flag
(*d):Zero flag
(*e):No subtraction is done because 9030 is not in BCD number.
(*f):No subtraction is done because 9040 is not in BCD number.

NOTE-1:]64~]71 is the special area. Refer to 8-2 [3] "Special relay".

NOTE-2: If the contents of S_1 is smaller than the contents of S_2, the result is obtained in the complement of 100.

(Ex) 23 - 85 = -62

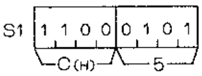
whereas, the compliment of 100 for 62 is 38.

You may take it to be (123 - 85 ≑ 38).

NOTE-3: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met. Refer to 9-5[5] "Data processing instruction and flag".

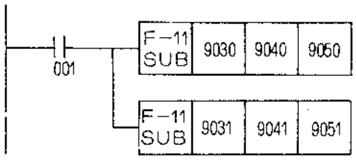
NOTE-4: If the contents of S_1 or S_2 are numbers other than BCD number, it makes the error flag (655) activated and no subtraction is done. (The contents of D remain unchanged.)

(Ex)

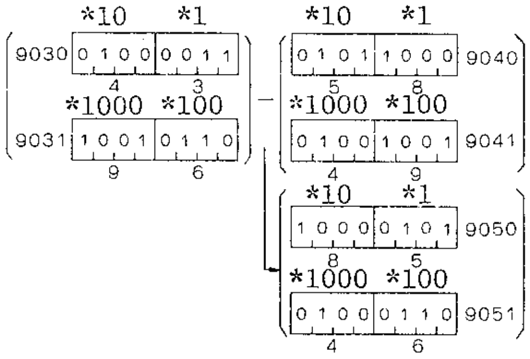


1100 is prohibited in the BCD system.

REFERENCE: In case more than BCD 3 digits has to be subtracted successively, the F-11 instruction must be provided in continuation. When the F-11 instruction is programmed repeatedly, the contents of the carry flag (656) are also subtracted after the second instruction. For the F-11 instruction that appears first in succession to the STR instruction, the contents of the carry flag (656) are not subtracted.



Instruction	
STR	001
F-11	
	9030
	9040
	9050
F-11	
	9031
	9041
	9051



*1 : Ones order

*10 : Tens order

*100 : Hundreds order

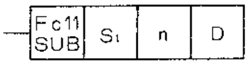
*1000: Thousands order

° The above example shows the case of 9643 - 4958 = 4685.

° If programmed from low order digit, the carry information will be carried on to a higher digit.

Refer to 9-5[6] "Double length operation".

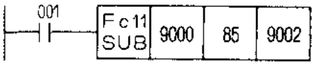
Fc11 SUB — Subtract register by 2-digit BCD constant (SUBtract)

Symbol		
Function		The contents of the register S_1 are subtracted by a 2-digit BCD constant "n" and its result is stored in the register D.
Operation		$S_1 \rightarrow n \rightarrow D$
Range of S_1 applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177
Range of "n" applicable		00~99
Range of D applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of signal (OFF to ON)
After the operation	Contents of S_1	Unchanged
	Flag	(see flag table below)

Result	a	b	c	d
0	1	0	0	1
1~99	1	0	0	0
Negative value	0	0	1	0
If S_1 is not BCD code	0	1	0	0

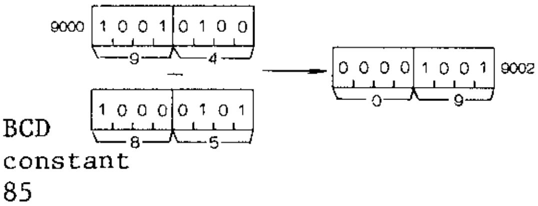
a:Non-carry	654
b:Error	655
c:Carry	656
d:Zero	657

[Explanation]



Instr.	
STR	001
Fc11	
	9000
	85
	9002

When the input condition 001 changes from OFF to ON, the contents of the register 9000 are subtracted by the BCD constant 85 and its result is stored in the register 9002.
It operates in the same timing as the F-11 instruction.



NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

NOTE-2: If S_1 is smaller than "n", the result will be obtained in the compliment of 100.

(Ex) 23 - 85 = -62

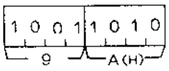
whereas, the compliment of 100 for 62 is 38.

You may take it to be (123 - 85 = 38).

NOTE-3: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met. Refer to 9-5[5] "Data processing instruction and flag".

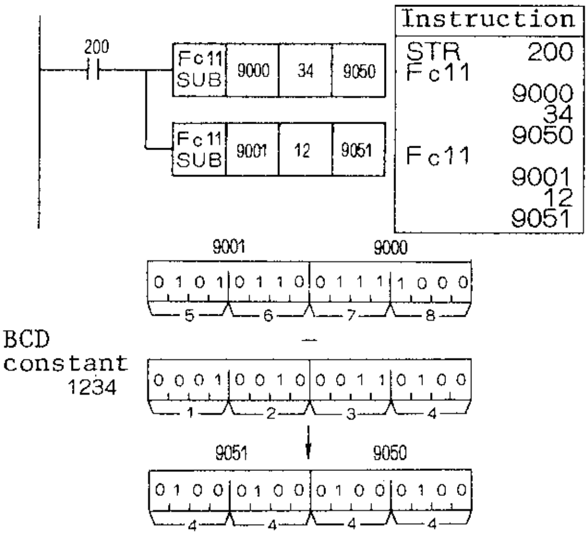
NOTE-4: If the contents of S1 was a number other than BCD code, it makes the error flag (655) set active and no subtraction will be done.

(Ex)

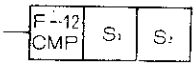


1010 is the code prohibited to use in the BCD system.

REFERENCE: Similar as the F-11 instruction, it is possible to subtract more than three digits of BCD value.



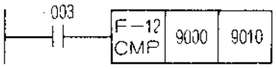
F-12 CMP Compare registers (CoMPare)

Symbol		
Function		The contents of the register S_1 are compared with the contents of the register S_2.
Operation		S_1<=>S_2
Range of S_1 applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177
Range of S_2 applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		When the input signal in the ON state (not limited to the time changing from OFF to ON).
After the operation	Contents of S_1	Unchanged
	Contents of S_2	Unchanged
	Flag	

Register contents	a	b	c	d
S_1>S_2	1	0	0	0
S_1=S_2	1	0	0	1
S_1<S_2	0	0	1	0

a:Non-carry 654
b:Error 655
c:Carry 656
d:Zero 657

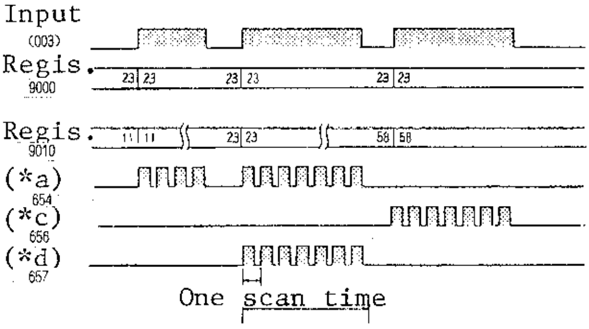
[Explanation]



Instr.	
STR	003
F-12	
	9000
	9010

When the input condition 003 is ON, the contents of the register 9000 are compared with the contents of the register 9010 and its results are set in the non-carry flag (654), carry flag (656), and zero flag (657). Registers 9000 and 9010 remain unchanged after this operation.

° Transition of register contents and flags



So long as the input condition is ON, comparison is done at every scan and the zero flag is turned ON.

(*a):Non-carry falg
(*c):Carry flag
(*d):Zero flag

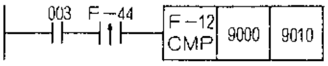
NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay"

NOTE-2: So long as the input signal is ON, comparison is done at every scan cycle. Refer to 9-5[4] "Operational condition".

NOTE-3: The error flag (655) is in 0 state at all times.

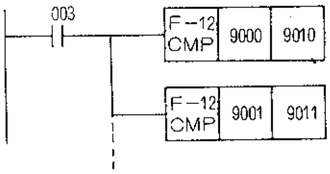
NOTE-4: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met. Refer to 9-5[5] "Data processing instruction and flag".

REFERENCE: In case comparison is to be done only when the input condition changes from OFF to ON, use the differentiate instruction in conjunction with the input condition.



Instruction	
STR	003
F-44	
F-12	
	9000
	9010

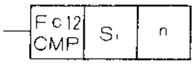
REFERENCE: To compare data of more than two bytes, it should be so programmed as in addition (F-10) and subtraction (F-11) that comparison should take place from a lower order digit. If the F-12 were programmed in succession, the contents of the carry flag (656) are also compared after the second F-12 instruction. (For the F-12 instruction that first follows the STR instruction, the contents of the carry flag (656) are removed from comparison.



Instruction	
STR	003
F-12	
	9000
	9010
F-12	
	9001
	9011

When programmed from a low order digit, the carry-down information is forwarded to a higher digit. Refer to 9-5[6] "Double length operation".

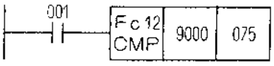
Fc12 CMP   Compare register with constant (CoMPare)

Symbol		
Function		The contents of the register S_1 are compared with an octal constant "n".
Operation		S_1<=>n
Range of S_1 applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177
Range of "n" applicable		000∿377
Operational condition		When the input signal in the ON state (not limited to the time changing from OFF to ON).
After the operation	Contents of S_1	Unchanged
	Flag	

Register contents	a	b	c	d
S_1>n	1	0	0	0
S_1=n	1	0	0	1
S_1<n	0	0	1	0

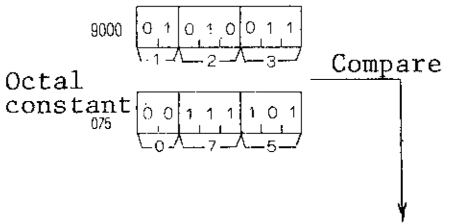
a:Non-carry 654
b:Error 655
c:Carry 656
d:Zero 657

[Explanation]



Instr.	
STR	001
Fc12	9000 075

When the input condition 001 is ON, the contents of the register 9000 are compared with the octal constant 075 and its results are set in the non-carry flag (654), carry flag (656), and zero flag (657).
The contents of the register 9000 remains unchanged after this operation.
It has the timing similar to the to the F-12 instruction.



Non-carry	Error	Carry	Zero
654	655	656	657
1	0	0	0

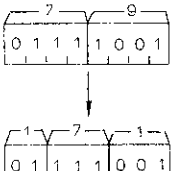
NOTE-1: ⊐64∿⊐71 is the special area. Refer to 8-2 [3] "Special relay".

NOTE-2: So long as the input signal is ON, comparison is done at every scan cycle. Refer to 9-5[4] "Operational condition".

NOTE-3: The error flag (655) is in 0 state at all times.

NOTE-4: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met. Refer to 9-5[5] "Data processing instruction and flag".

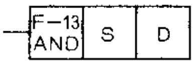
REFERENCE: Use an octal number in writing with the Fc12 instruction. Octal number can express any bit pattern and it does not need use of troublesome weight calculation. To compare with a BCD constant, convert it to an octal equivalent before writing it in the program.



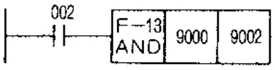
Represents 79 in the BCD notation.

Represents 171 in the octal notation.

F-13 AND· AND registers (AND)

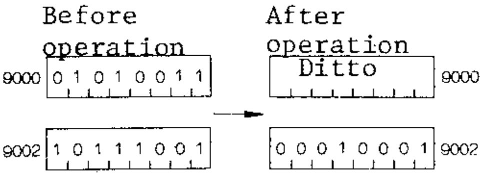
Symbol		
Function		The contents of the register S (8-bit data) are ANDed with the contents of the register D (8-bit data) and its result is stored in the register D.
Operation		S∩D→D
Range of S applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177
Range of D applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of signal (OFF to ON).
After the operation	Contents of S	Unchanged
	Contents of D	Result
	Flag	Unchanged

[Explanation]

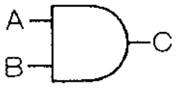


Instr.	
STR	002
F-13	
	9000
	9002

When the input condition 002 changes from OFF to ON, the 8-bit contents of the register 9000 are ANDed with the 8-bit contents of the register 9002 and its result is stored in the register 9002. The contents of the register 9000 remains unchanged.

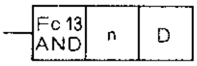


AND truth table

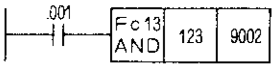
Symbol	A	B	C
	0	0	0
	1	0	0
	0	1	0
	1	1	1

NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

Fc13 AND — AND register with constant (AND)

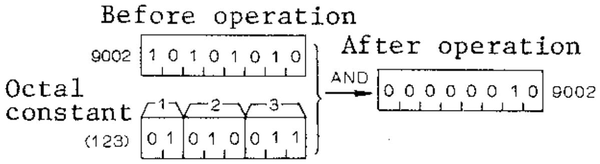
Symbol	
Function	An octal constant "n" is ANDed with the contents of the register D and its result is stored in the register D.
Operation	n ∩ D→D
Range of "n" applicable	000~377
Range of D applicable	⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition	At a rising edge of signal (OFF to ON).
After the operation: Contents of D	Result
After the operation: Flag	Unchanged

[Explanation]

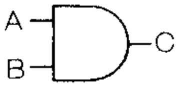


Instr.	
STR	001
Fc13	
	123
	9002

When the input condition 001 changes from OFF to ON, the octal constant 123 is ANDed with the contents of the register 9002 and its result is stored in the register 9002.

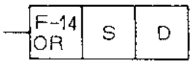


AND truth table

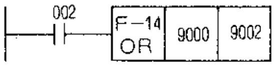
Symbol	A	B	C
	0	0	0
	1	0	0
	0	1	0
	1	1	1

NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

F-14 OR · OR registers (OR)

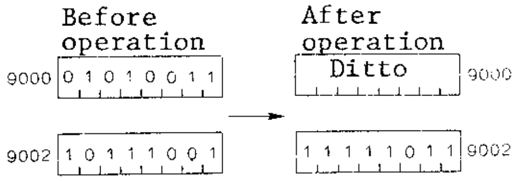
Symbol		
Function		The contents of the register S (8-bit data) are ORed with the contents of the register D (8-bit data) and its result is stored in the register D.
Operation		S ∪ D→D
Range of S applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Range of D applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of signal (OFF to ON).
After the operation	Contents of S	Unchanged
	Contents of D	Result
	Flag	Unchanged

[Explanation]

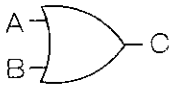


Instr.	
STR	002
F-14	
	9000
	9002

When the input condition 002 changes from OFF to ON, the 8-bit contents of the register 9000 are ORed with the 8-bit contents of the register 9002 and its result is stored in the register 9002. The contents of the register 9000 remains unchanged.

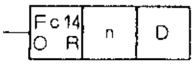


OR truth table

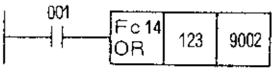
Symbol	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	1

NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

Fc14 ·OR OR register with constant (OR)

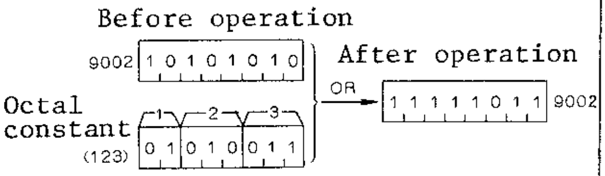
Symbol		
Function		An octal constant "n" is ORed with the contents of the register D and its result is stored in the register D.
Operation		n ∪ D→D
Range of "n" applicable		000∿377
Range of D applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of signal (OFF to ON).
After the operation	Contents of D	Result
	Flag	Unchanged

[Explanation]

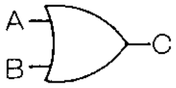


Instr.	
STR	001
Fc14	
	123
	9002

When the input condition 001 changes from OFF to ON, the octal constant 123 is ORed with the contents of the register 9002 and its result is stored in the register 9002.



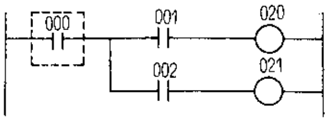
OR truth table

Symbol	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	1

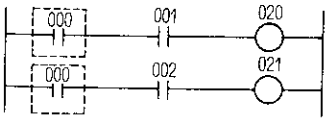
NOTE-1: ⊐64∿⊐71 is the special area. Refer to 8-2 [3] "Special relay".

F-30 MCS.	Set master control (Master Control Set)
F-31 MCR	Reset master control (Master Control Reset)

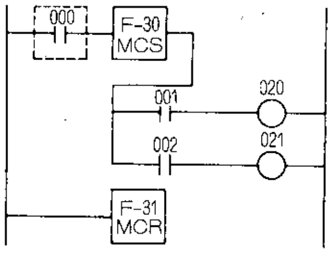
MCS and MCR should be used when the circuit after the common operational condition is branched to multiple number of outputs.



(1) In the case of the relay board



(2) In the case of MCS and MCR are not used



```
      STR       000
MCS→  F-30
      STR       001
      OUT       020
      STR       002
      OUT       021
MCR→  F-31
```

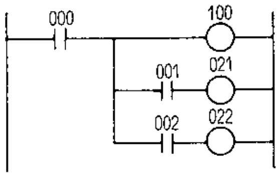
(3) In the case of MCS and MCR are used

When the F-30 (MCS) instruction is programmed, the contents of the accumulator is stored in the CPU internal register, and operations for successive instructions are ANDed with the contents of the CPU internal register until the F-31 (MCR) instruction is reached. The F-31 (MCR) instruction indicates the range of AND operation applied.

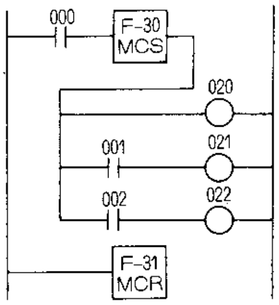
It will help simplify the program when the command operational condition indicated in the block ⬚ is complicated or many branches are set after the common operational condition.

NOTE-1: Do not directly connect the F-30 (MCS) derived bus line with OUT, TMR, and CNT instructions and application instruction.

(1) In the case of the relay board

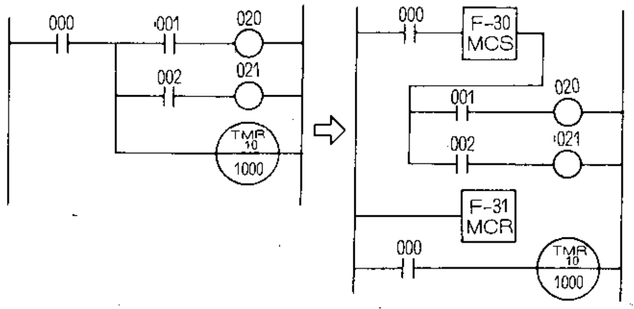


(2) MCS, MCR prohibited program

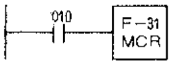


Instruction	
STR	000
F-30	
OUT	020
STR	001
OUT	021
STR	002
OUT	022
F-31	

It must be program, in the following manner.

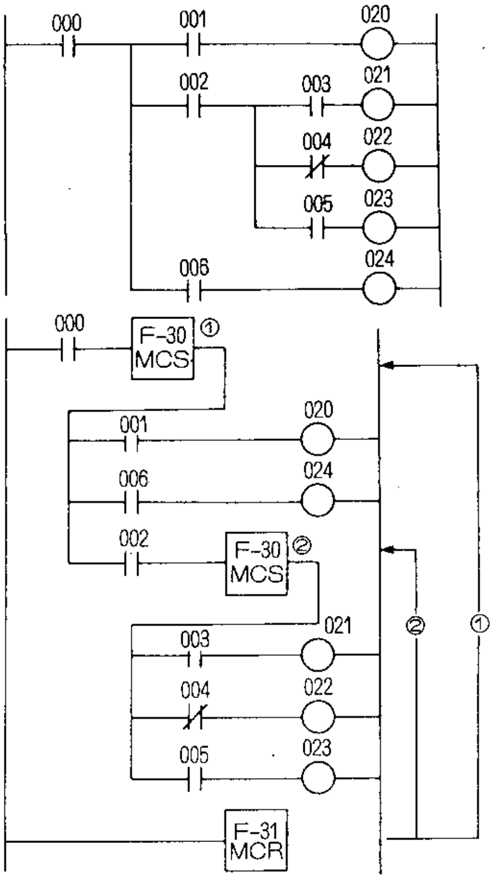


NOTE-2: The F-31 (MCR) instruction is an unconditional instruction.



It prohibits the program like shown above.

Another MCS may be used in-between MCS and MCR.



The relay board ladder chart shown in left can be programmed in the following manner using MCS and MCR. However, there may be a need of changing the program sequence as in the example. (*)

```
Instruction
STR      000
F-30
STR      001
OUT      020   }
STR      006   } ※
OUT      024   }
STR      002
F-30
STR      003
OUT      021
STR NOT  004
OUT      022
STR      005
OUT      023
F-31
```

° The F-31 (MCR) instruction indicates the termination of the preceding F-30 (MCS) instruction; ① and ② in the example.

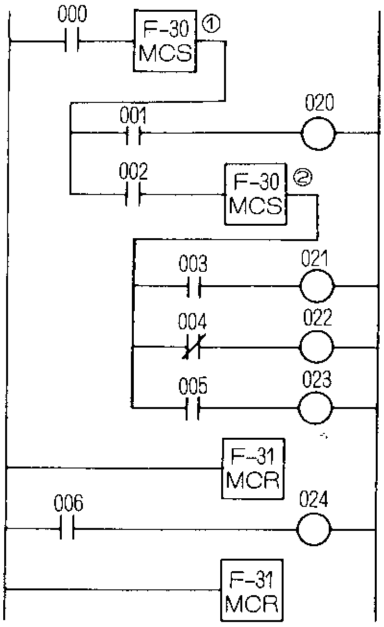
NOTE-3: The desired circuit would not be established if programmed in the following way.

```
        Instruction
MCS→    STR      000
        F-30
        STR      001
        OUT      020
        STR      002
MCS→    F-30
        STR      003
        OUT      021
        STR NOT  004
        OUT      022
        STR      005
        OUT      023
MCR→    F-31
        STR      006  }
        OUT      024  } ※
MCR→    F-31
```

This MCR is insignificant.

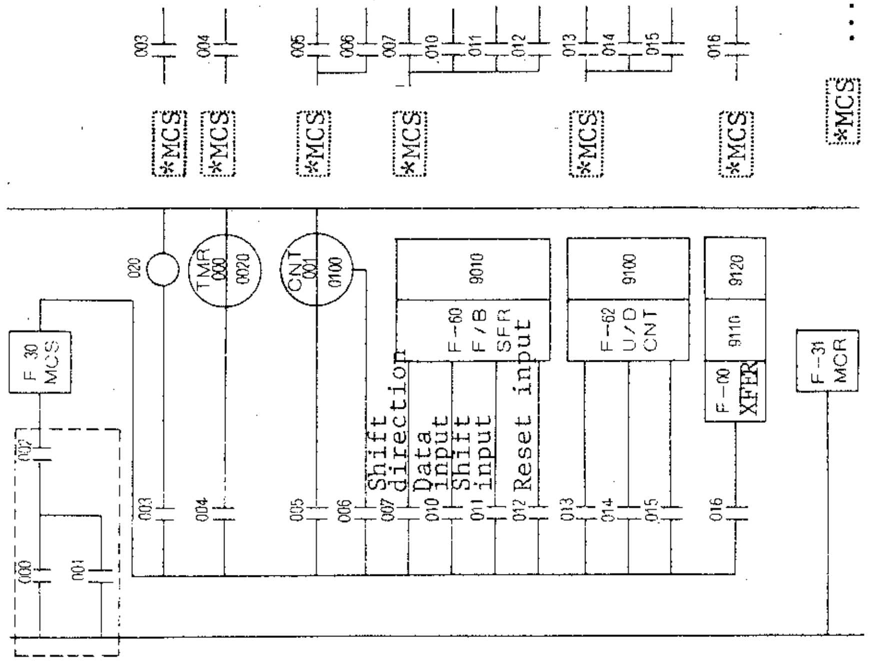
MCS of ① and ② has been terminated with this MCR.

⇨



NOTE-4: Although it is possible to insert MCS as many times as required between MCS and MCR*, the operational range of MCS terminates with MCR marked with an asterisk.

When the MCS condition (shown within dotted line) is OFF, instruction existing between MCS and MCR will be handled in the following manner.



0100 is OFF because it is OFF.

TMR is reset because it is OFF. (TMR contact turns to OFF and the current value to preset value.)

Because it is OFF;

	ON reset	OFF reset
CNT contact	Retained	OFF
Current value	Retained	Preset value

Because it is OFF,

	ON reset	OFF reset
9010	Retained	Cleared
Flag	Cleared	Cleared

Because it is OFF;

	ON reset	OFF reset
9100	Retained	Cleared
Flag	Cleared	Cleared

No operation is executed because it is OFF.

*MCS ... MCS condition

	Flag
Flag unaffecting instruction	No change
Flag affecting instruction	Cleared

NOTE-5: CNT, F-60, and F-62 instructions can set the reset condition to ON-reset or OFF-reset in the system memory #202. In the case of OFF-reset, it is reset by MCS.

F-40 END. End instruction (END)

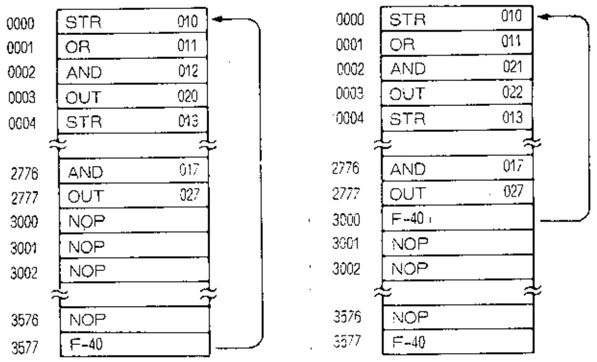
The F-40 instruction indicates the end of the program. As it has the highest priority, it will be effective even if existing between F-41 (JCS) and F-42 (JCR) so that it makes the user program terminated.

There is no need of writing the END instruction necessarily, except for the following cases, because it will be automatically set in the last address of the program memory when the memory is cleared after specifying the program memory capacity.

(1) To accelerate the scan time

The scan time is I/O processing time plus the user program execution time. The user program execution time is the total time required to execute all instructions from the program address 0000 to the END instruction. The location of the END instruction automatically written after memory clear is 3577.

Assume now if the last address is 2777 (1536th word) when the ladder chart has written by the programmer upon completion of designing, 3000 ∿ 3576 are padded with NOP instructions with the END instruction in 3577, so that unwanted time has to be consumed for execution of NOP instruction (1.09microsecond per word). If F-40 is written in 3000, it accelerates the processing time without executing those unwanted NOP instructions and the control can proceed to a next scan cycle after terminating the user program relatively faster.



(a) END (3577) after mere memory clear

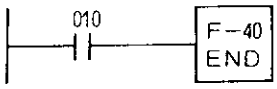
(b) F-40 (END) written in 3000

(2) To perform partial program execution during trial run

By inserting the F-40 instruction at an end of a sequence block, you will be able to execute only the required portion of the program. If the result was successful, that F-40 then must be deleted.

NOTE-1: A multiple number of F-40 may exist when memory is added or the END instruction is written in (2). In this case, the user program terminates at the first F-40. So, it is a must to check the location of the END instruction before going into the actual operation.

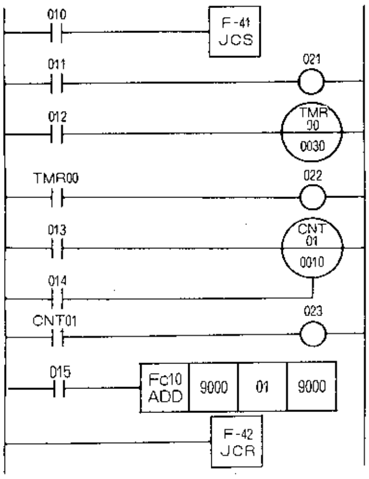
NOTE-2: The F-40 (END) instruction is unconditional, and it prohibits the following kind of programming.



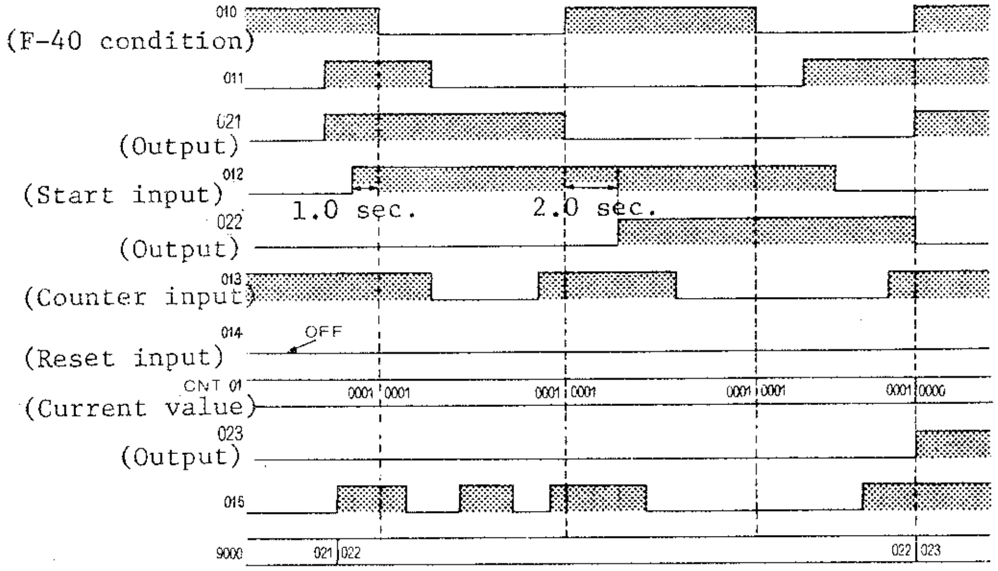
F-41 JCS Set jump control (Jump Control Set)

F-42 JCR Reset jump control (Jump Control Reset)

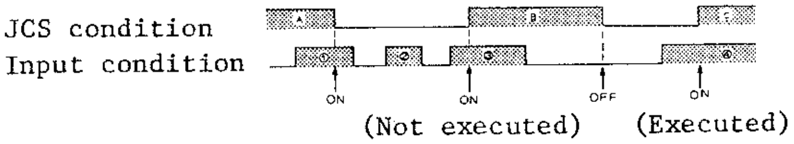
When the F-41 (JCS) condition is OFF, all instructions except the END instruction before the F-42 (JCR) instruction are not executed. Therefore, even if there was an instruction by which the result of the OUT, TMR, CNT, MD, or application instruction is stored in the data memory, it does not change the contents of the data memory, and it retains the state at the time of JCS condition is ON.



```
Instruction
STR        010
F-41
STR        011
OUT        021
STR        012
TMR         00
          0030
STR TMR     00
OUT        022
STR        013
STR        014
CNT         01
          0010
STR CNT     01
OUT        023
STR        015
Fc10
          9000
            01
          9000
F- 42
```



NOTE-1: Attention must be paid to the TMR internal clock (0.1-second clock), CNT counter input and application instruction input condition (that which the operation takes place with OFF to ON change in the input condition), and F-41 (JCS) condition ON/OFF timing.



Operation takes place at a rising of ① because the JCS condition is ON.

Operation does not take place at a rising of ② because the JCS condition is OFF.

Operation does not take place at a rising of ③ because the JCS condition is OFF.

The JCS condition turns to ON while ③ is ON, but operation does not take place as it does not recognize that the input condition has changed from OFF to ON, because the input condition is ON with which the JCS condition of Ⓐ changes from ON to OFF and the input condition is ON with which the JCS condition of Ⓑ changes from OFF to ON.

Operation does not take place at a rising of ④ because the JCS condition is OFF.

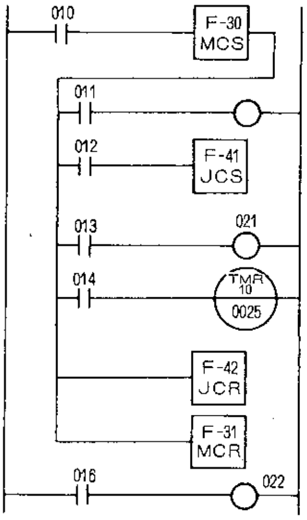
The JCS condition becomes ON while ④ is ON.

Operation takes place immediately after the JCS condition of Ⓒ changes from OFF to ON because the input condition is OFF with which the JCS condition of Ⓑ changes from ON to OFF and the input condition is ON with which the JCS condition of Ⓒ changes from OFF to ON.

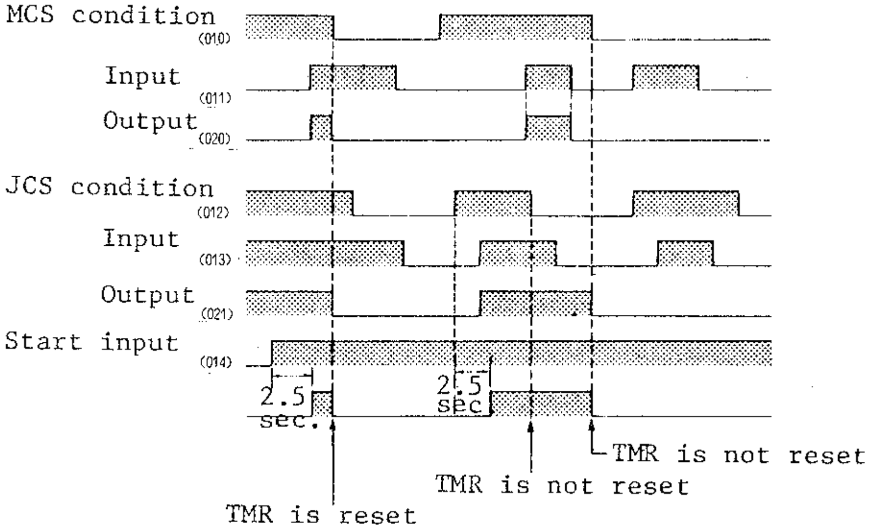
NOTE-2: Operation does not take place if the JCS condition is OFF when there is a flag effecting instruction between F-41 (JCS) and F-42 (JCR). But, flags, 654 through 657, are cleared.

NOTE-3: The END instruction will be executed regardless whether the JCS condition be ON or OFF, if there was a F-40 (END) instruction between F-41 (JCS) and F-42 (JCR), and the user program execution is terminated and the control proceeds to a next scan cycle.

NOTE-4: F-41 (JCS) and F-42 (JCR) can be nested between F-30 (MCS) and F-31 (MCR). However, as MCS has higher priority than JCS, it assumes the state when the MCS condition is OFF (OFF for such as the output relay and reset for such as TMR), when the MCS condition changes to OFF.

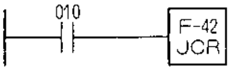


Instruction	
STR	010
F-30	
STR	011
OUT	020
STR	012
F-41	
STR	013
OUT	021
STR	014
TMR	10
	0025
F-42	
F-31	
STR	016
OUT	022



NOTE-5: It is not permitted to insert another F-41 and F-42 between F-41 (JCS) and F-42 (JCR). It will indicate "JCS ERROR" on the ZW-10PG1 programmer during program check, if such a program was written.

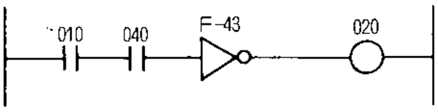
NOTE-6: The following kind programming is not possible as F-42 (JCR) is an unconditional instruction.

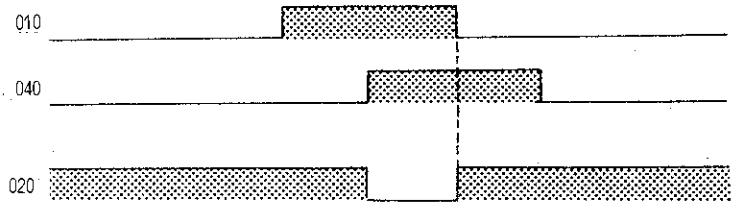


F-43 CPL Complement bit (ComPLement)

The F-43 instruction complements the bit in the accumulator.

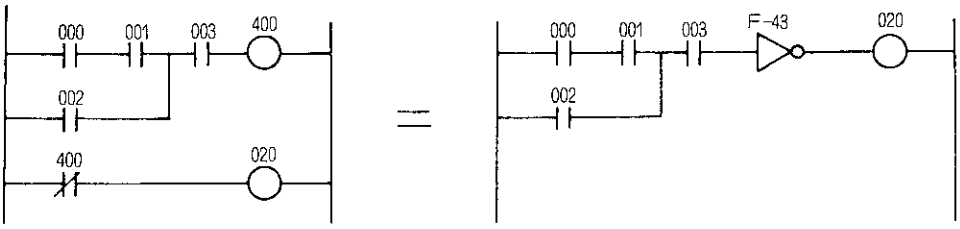
Instruction	
STR	010
AND	040
F-43	
OUT	020





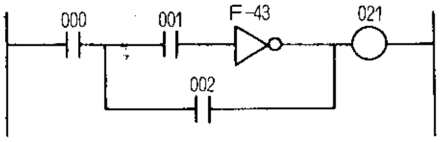
Results from the STR instruction to the F-43 instruction are complemented and sent to the output relay 020 .

Use of F-43 permits to obtain the complemented output without the use of auxiliary contact.

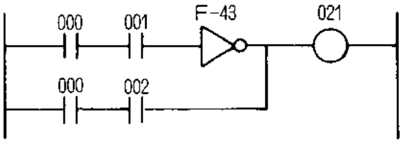


NOTE-1: The F-43 instruction may be used for a single or multiple number of contacts.

NOTE-2: Pay attention to it that the following programs (a) and (b) will not produce the same result because the F-43 instruction is the instruction that complement the contents of the accumulator.



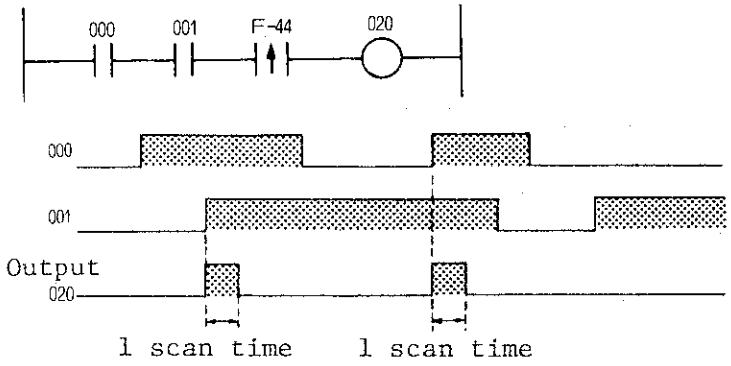
(a)



(b)

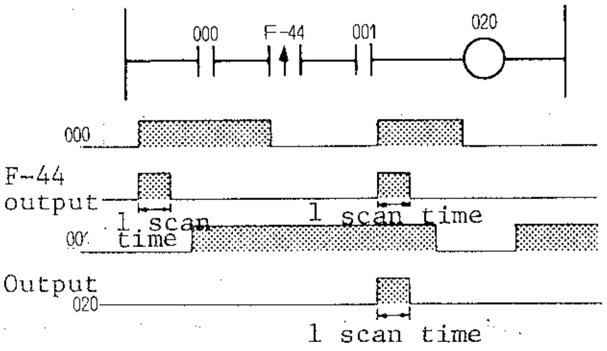
F-44 Differentiate at ON

When the state of the accumulator just before the F-44 instruction changes from OFF to ON, one scan time pulse is generated.



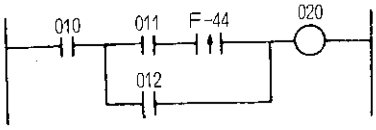
Instruction	
STR	000
AND	001
F-44	
OUT	020

NOTE-1: Note that a different result is produced when the sequence of the F-44 program is changed in the above ladder chart. (Identical in the case of F-45.)

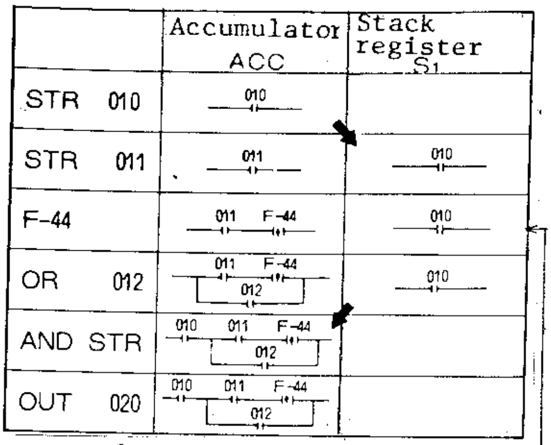


NOTE-2: F-44 condition may be a single or multiple number of contacts.

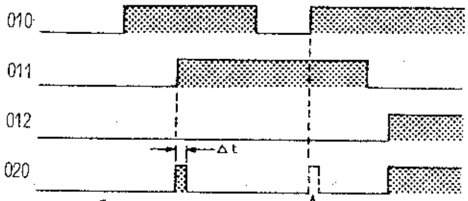
NOTE-3: F-44 is the instruction that produces the signal that turns ON for a period of one scan time by detecting an OFF to ON change in the contents of the immediate accumulator.



STR	010
STR	011
F-44	
OR	012
AND STR	
OUT	020



The accumulator goes ON for the scan cycle that 011 changed from OFF to on.



Δt : 1 scan time

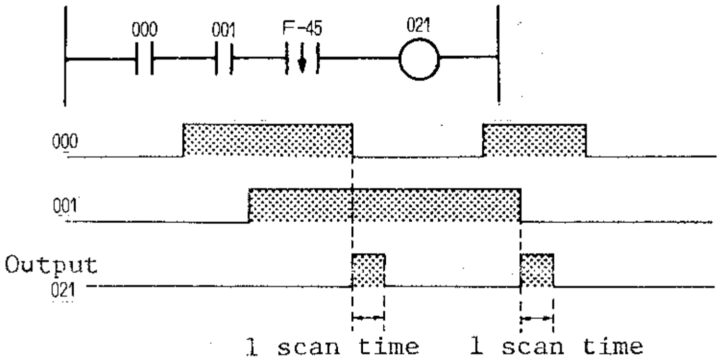
Differential signal is not issued.

In the above example, a differential signal is not issued even if 010 has changed from OFF to ON when 011 is ON, because 010 is ANDed by the AND STR instruction.

F-45 —|↓|— Differentiate at OFF

When the state of the accumulator just before the F-45 instruction changes from ON to OFF, one scan time pulse is generated.

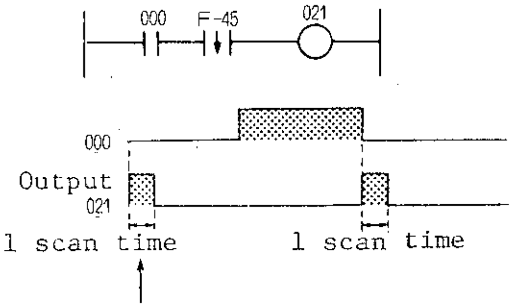
Instruction	
STR	000
AND	001
F-45	
OUT	021



NOTE-1: You may use differential instruction (F-44,F-45) required in your program unless 510 times. When differential instruction are used over 511 times,"Memory failure" is displayed.

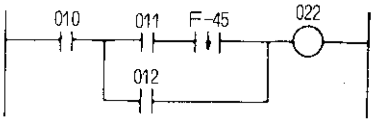
NOTE-2:

NOTE-2: Use of the F-45 instruction may sometimes generate one scan time pulse at the operation immediately following program writing, that is, when the address of F-45 instruction is changed because of program insertion or deletion.

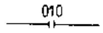
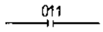
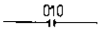
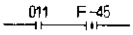
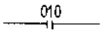
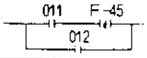
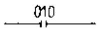
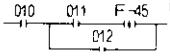
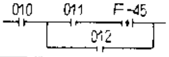


Output (021) turns ON if input (000) is OFF at the start of operation immediately after write of program.

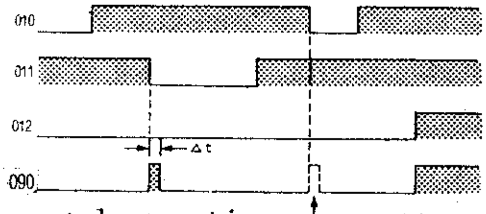
NOTE-3: F-45 is the instruction that produces the signal that turns ON for a period of one scan time by detecting an ON to OFF change in the contents of the immediate accumulator.



```
STR      010
STR      011
F-45
OR       012
AND STR
OUT      220
```

	Accumulator ACC	Stack register S
STR 010		
STR 011		
F-45		
OR 012		
AND STR		
OUT 022		

The accumulator turns ON for a period of a scan time that 0011 changed from ON to OFF.

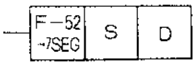


Δt: 1 scan time

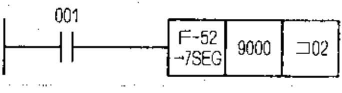
Differentiate signal not issued

In the above example, differentiate signal is not issued even if 010 changed from ON to OFF when 011 is OFF, because 010 is ANDed by the AND STR instruction.

F-52 →7SEG. Decode to 7-segment data

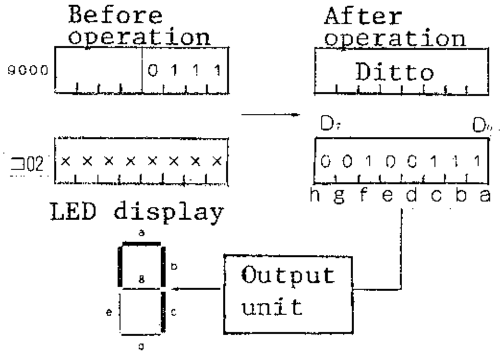
Symbol		
Function		The low order 4-bit data in registers S are decoded into 7-segment display data.
Operation		S→D
Range of S applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177
Range of D applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of signal (OFF to ON).
After the operation	Contents of S	Unchanged
	Contents of D	Result (Refer to "7-segment decoder chart")
	Flag	Unchanged

[Explanation]



Instr.	
STR	001
F-52	
	9000
	⊐. 02

When the input condition 001 changes from OFF to ON, the low order 4 bits of the register 9000 are decoded into the 7-segment display data. See "7-segment decoder chart" relation between input data and display output.

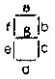


NOTE-1: ⊐64~⊐71 is the special area.

Refer to 8-2 [3] "Special relay".

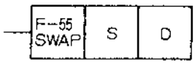
NOTE-2: D_0 ~ D_6 of the output data correspond to a ~ g of the 7-segment display. D_7 remains "0" at all times.

7-segment decoder chart

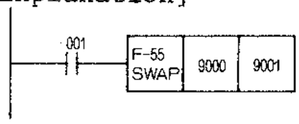


Input data	Output data	Display output
	g f e d c b a	
00000000	00111111	0
00000001	00000110	1
00000010	01011011	2
00000011	01001111	3
00000100	01100110	4
00000101	01101101	5
00000110	01111101	6
00000111	00100111	7
00001000	01111111	8
00001001	01101111	9
00001010	01110111	A
00001011	01111100	b
00001100	00111001	C
00001101	01011110	d
00001110	01111001	E
00001111	01110001	F

F-55 SWAP Swap high order 4 bits with low order 4 bits (SWAP)

Symbol		
Function		High order 4 bits are swapped with low order 4 bits of the register S and stored in the register D.
Operation		S→D
Range of S applicable		⊐00~⊐71 (NOTE-1) b000~b137 9000~9177
Range of D applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of signal (OFF to ON)
After the operation	Contents of S	Unchanged
	Contents of D	Result
	Flag	Unchanged

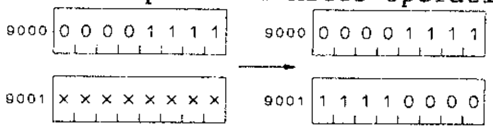
[Explanation]



Instr.	
STR	001
F-55	
	9000
	9001

When the input condition 001 changes from OFF to ON, high order 4 bits are swapped with low order 4 bits of the register 9000 and its result is stored in the register 9001. The contents of the register 9000, however, remains unchanged.

Before operation After operation

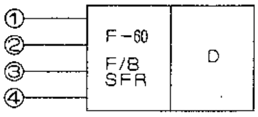
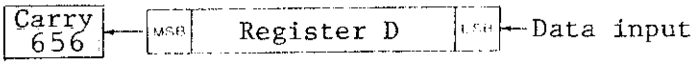
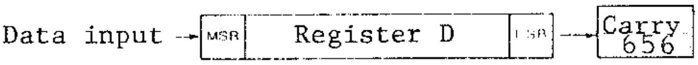


×: Don't care

NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

NOTE-2: F-55 becomes valid in the following case.
With the F-52 (7SEG decode) instruction, the low order 4 bits are decoded into the 7-segment display data. To make display of more digits on the display, the high order 4 bits need to be swapped with the low order 4 bits, then F-52 should be executed.

F-60 SFR . Shift register bidirectional (Forward/Backward ShiFt Register)

<table>
<tr><td colspan="2">Symbol</td><td colspan="5">
① Shift direction input
② Data input
③ Shift input
④ Reset input</td></tr>
<tr><td colspan="2">Function</td><td colspan="5">The 8-bit data in the register S are shifted to high order or low order bit positions according to the shift direction input of ①.</td></tr>
<tr><td colspan="2">Operation</td><td colspan="5">° Shift direction input is ON:

° Shift direction input is OFF:
 </td></tr>
<tr><td colspan="2">Range of D applicable</td><td colspan="5">⊐02~⊐71 (NOTE-1) b000~b137 9000~9177</td></tr>
<tr><td colspan="2">Operational condition</td><td colspan="5">When the reset input ④ is OFF, bits are shifted at a rising edge of the shift input ③ .</td></tr>
<tr><td rowspan="4">After the operation</td><td>Contents of D</td><td colspan="5">° Result is contained when the reset input ④ is OFF.
° All bits are reset to OFF when the reset input ④ is ON.</td></tr>
<tr><td rowspan="3">Flag</td><td>Reset input ④</td><td>Non-carry 654</td><td>Error 655</td><td>Carry 656</td><td>Zero 657</td></tr>
<tr><td>OFF</td><td>1 or 0</td><td rowspan="2">0</td><td>0 or 1</td><td>0 or 1</td></tr>
<tr><td>ON</td><td>0</td><td>0</td><td>0</td></tr>
</table>

NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

NOTE-2: By setting the reset condition in the system memory (#202), it permits to reset with OFF.

NOTE-3: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met. Refer to 9-5[5] "Data processing instruction and flag".

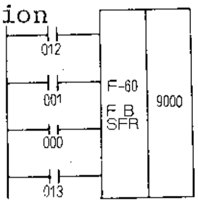
[Explanation]

Shift direction

Data input

Shift input

Reset input

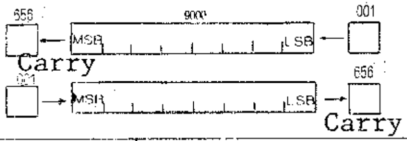


Instr.	
STR	012
STR	001
STR	000
STR	013
F-60	
	9000

When shift input 0000 changes from OFF to ON, data are shifted in the following manner depending on the state of the shift direction input 012.

When 012 is ON

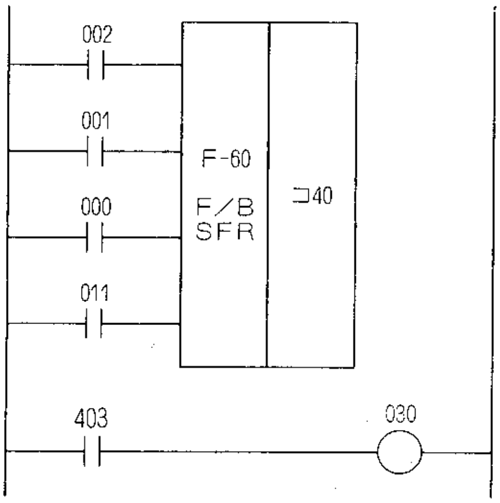
When 012 is OFF



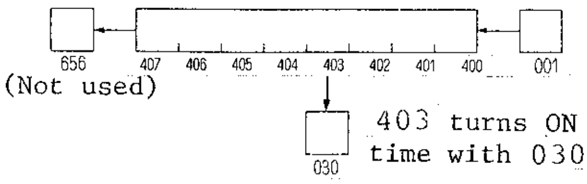
Input condition	9000(before oper.)								9000(after oper.)								Non-carry 654	Carry 656	Zero 657
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
4012 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●
0100 ○	●	○	○	○	●	○	●	○	○	●	○	○	○	●	○	●	●	○	○
0000 ʃ	○	●	○	○	●	○	○	●	○	○	●	○	○	●	○	○	○	●	○
4013 ○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	●	●
4012 ●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●
0100 ○	○	●	○	●	○	○	○	○	●	○	●	○	○	○	○	○	●	○	○
0000 ʃ	●	○	○	○	●	○	○	○	○	○	○	●	○	○	○	○	○	●	○
4013 ○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●
4012 ○ 0100 ● 0000 ʃ 4013 ○	●	○	○	○	○	●	○	○	●	●	○	○	○	○	●	○	●	○	○
	○	○	○	●	○	○	○	●	●	○	○	○	●	○	○	○	○	●	○
4012 ● 0100 ● 0000 ʃ 4013 ○	○	●	○	○	○	○	○	○	●	○	○	○	○	○	○	●	●	○	○
	●	○	○	○	●	○	○	○	○	○	○	●	○	○	○	●	○	●	○
4013 ●	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○

° Error flag (655) is OFF at all times. ○ OFF ● ON

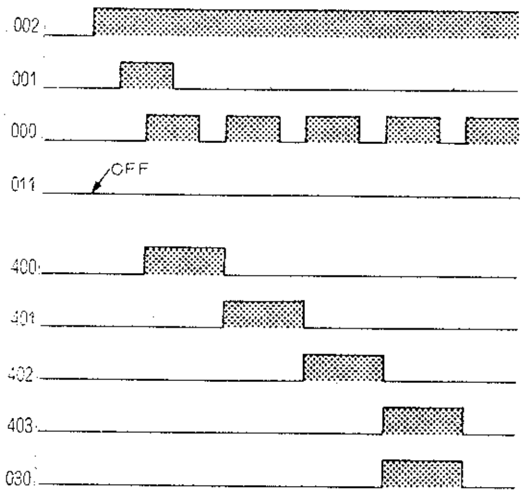
REFERENCE: When ℶ.×× is assigned for the register D, it allows to constitute a n-bit (n<8) shift register.



(When 002 is ON)

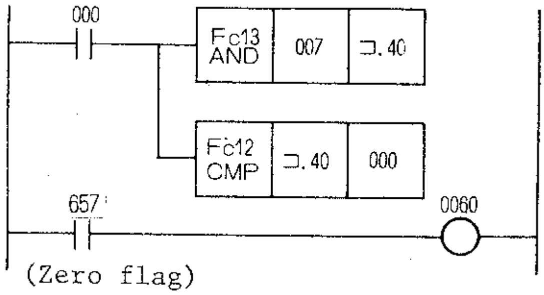


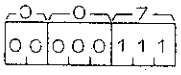
403 turns ON at the same time with 030.



NOTE-1: Data are shifted in 404 ∿ 407.

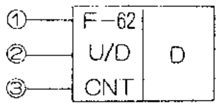
NOTE-2: When all of 400 through 407 are 0, the zero flag turns to 1.
The following program must be used to check if 400 ∿ 402 are 0.





By ANDing this way, 403 ∿ 407 are masked (all 0).

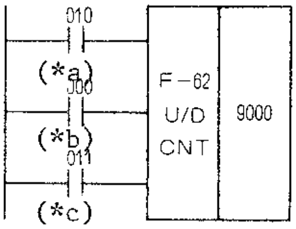
F-62 U/D CNT	Set up/down counter (Up/Down CouNTer)

Symbol		 ① Up/down counter direction input ② Counter input ③ Reset input
Function		The contents of the register D (BCD 2 digits) are added (① ... ON) or subtracted (① ... OFF), according to the up/down counter direction input.
Operation		① Up/down direction input is ON: <D>+1→D ① Up/down direction input is OFF: <D>-1→D
Range of D applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising of the counter input ② (OFF to ON) when the reset input ③ is OFF.
After the operation	Contents of D	° Result (BCD code) is contained when the reset input ③ is OFF. ° All bits are OFF when the reset input ③ is ON.
	Flag	

(†)	Result	a	b	c	d
ON	99+1→00	0	0	1	1
	00∿98+1 →01∿99	1	0	0	0
	Other than BCD	0	1	0	0
OFF	00-1→99	0	0	1	0
	01-1→00	1	0	0	1
	02∿99-1 →01∿98	1	0	0	0
	Other than BCD	0	1	0	0
Reset input ③		0	0	0	0

(†):Up/down counter direction input ①
a:Non-carry 654
b:Error 655 c:Carry 656
d:Zero 657

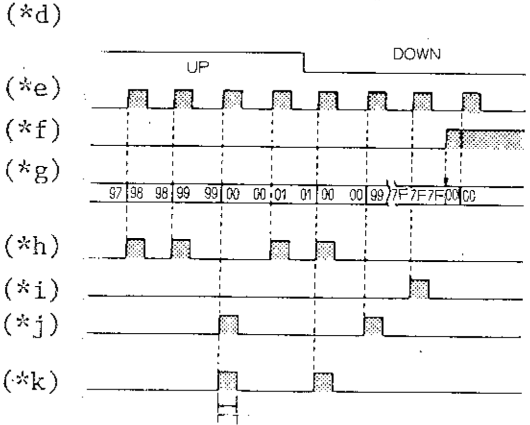
[Explanation]



Instr.	
STR	010
STR	000
STR	011
F-62	
	9000

(*a):Up/down counter direction
(*b):Counter input
(*c):Reset input

When the reset input 011 is OFF, it permits to count, if set to ON reset mode.
When the up/down counter direction input 010 is ON, it acts as an incremental counter. When OFF, it acts as a decremental counter.
If the contents of 9000 were other than the BCD code, it makes the error flag (655) turned ON and no counter operation takes place. (7F in the example)



1 scan time, max.

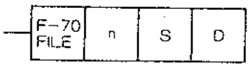
Valid until a flag affecting instruction is met in the program.
(*d):Up/down counter direction
(*e):Counter input
(*f):Reset input
(*g):Register
(*h):Non-carry flag
(*i):Error flag
(*J):Carry flag
(*k):Zero flag

NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

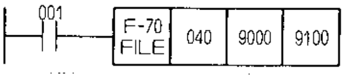
NOTE-2: By setting the reset condition in the system memory (#202), it is possible to reset by OFF.

NOTE-3: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met. Refer to 9-5[5] "Data processing instruction and flag".

F-70 FILE Transfer n-byte in batch (FILE)

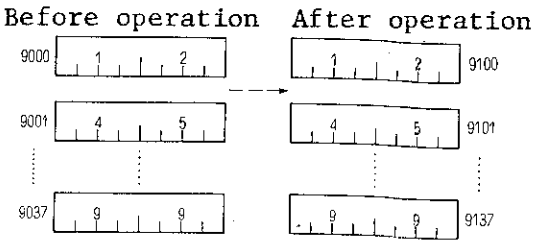
Symbol		
Function		A n-byte octal data from the register S to the register S+n-1 are transferred in batch to the n-byte area beginning from the register D to D+n-1.
Operation		S,...S+n-1 → D,...D+n-1
Range of "n" applicable		000～200(8) (To be 128 bytes, if 000.)
Range of S applicable		⊐00～⊐71 (NOTE-1) b000～b137 9000～9177
Range of D applicable		⊐02～⊐71 (NOTE-1) b000～b137 9000～9177
Operational condition		At a rising edge of input signal (OFF to ON).
After the operation	Contents of S‥S+n-1	Unchanged
	Contents of D	Contents of the register S
	Contents of D+1	Contents of the register S+1
	⋮	⋮
	Contents of D+n-1	Contents of the register S+n-1
	Flag	Unchanged

[Explanation]



Instr.	
STR	001
F-70	
	040
	9000
	9100

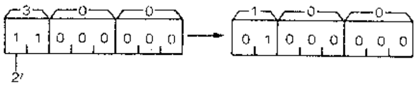
When the input condition 001 changes from OFF to ON, 040(8) bytes data (32 bytes in decimal) in register 9000 through 9037 are transferred in batch to the 32 bytes area of 9100 through 9137. The contents of registers, 9000 ～ 9037, remain unaffected.



NOTE-1: ⊐64～⊐71 is the special area. Refer to 8-2 [3] "Special relay".

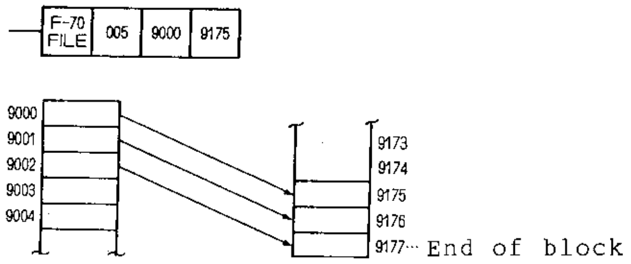
NOTE-2: Use an octal number within a range of 000 to 200 to represent the size of transfer bytes "n".

Though it is possible to use such as 300(8) in terms of programming, it will be taken for 100(8) as the bit in 2^7 is disregarded.

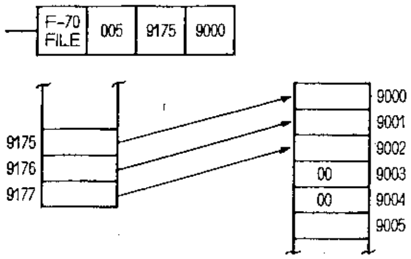
"n" designated 

Also, programming "n" to 000(8) will be assumed for 200(8) and executed.

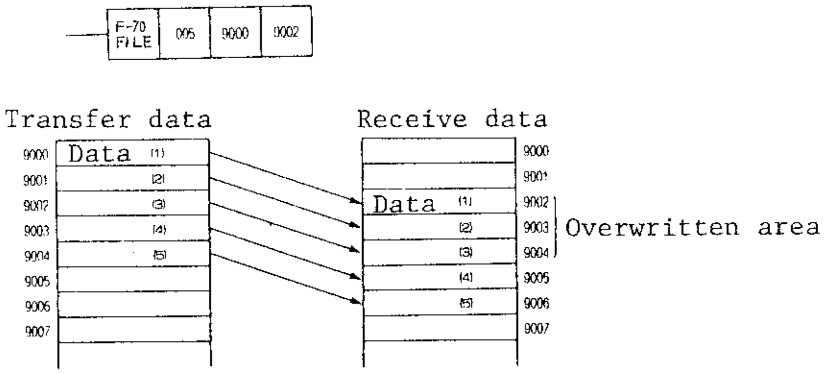
NOTE-3: If "n" and "D" are set over the register area (]02~]71, b000~b137,9000~9177)below expressed,the source is transfered to the register area untill the end of block.



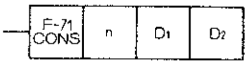
NOTE-4: If "n" and "D" are set over the register area (]00~]71, b000~b137,9000~9177) below expressed,the source which is over the register is transfered to D as data "0".



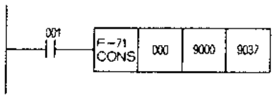
NOTE-5: It would be possible to program "n", S, and D that the source may overwrite the destination.



F-71 CONS Transfer octal constant in batch (CONStant)

Symbol		
Function		An octal constant "n" is transferred in batch from the register D_1 to the register D_2.
Operation		$n \rightarrow D_1, \ldots\ldots, D_2$ n=000~377(8)
Range of "n" applicable		000~377(8)
Range of D_1 applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Range of D_2 applicable		⊐02~⊐71 (NOTE-1) b000~b137 9000~9177
Operational condition		At a rising edge of input signal (OFF to ON).
After the operation	Contents of D_1 Contents of D_1+1 ⋮ Contents of D_2+1 Contents of D_2	Constant n
	Flag	Unchanged

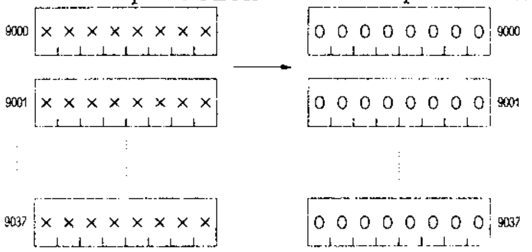
[Explanation]



Instr.	
STR	001
F-71	
	000
	9000
	9037

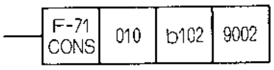
When the input condition 001 changes from OFF to ON, the octal constant 000 is transferred in batch from registers 9000 through 9037.

Before operation After operation

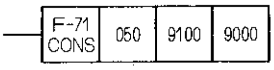


NOTE-1: ⊐64~⊐71 is the special area. Refer to 8-2 [3] "Special relay".

NOTE-2: No operation takes place if D_1 or D_2 was programmed in such way as to extend over another register area (⊐00~⊐71, b000~b137, 9000~9177).



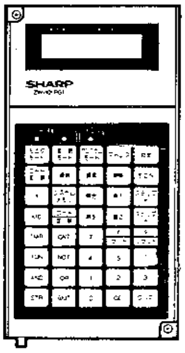
NOTE-3: Address designation of D_1 smaller than D_2 will not execute the program.



$D_1(9100) > D_2(9000)$

§ 10 How to usc support tools

10-1 Programmer (ZW-10PG1)

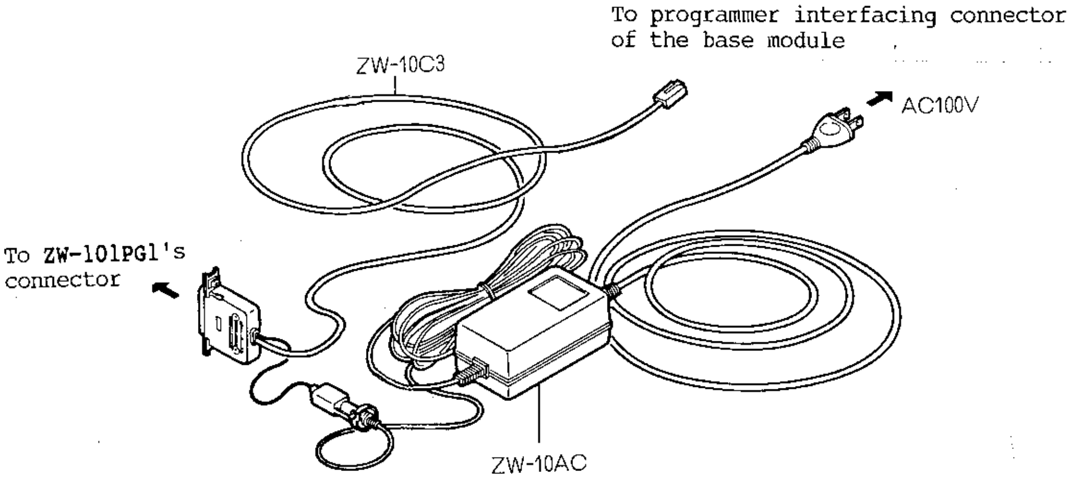


Major facility
• Instruction words programming
• Change of preset or current value
• CMT transfer
• Program check

■ Connection with the bàse module

The programmer interfacing connector of the base module must be conncted with the connector of the programmer (ZW-101PG1) using adapting cable for support tool (ZW-10C3).

In addition to this, ZW-10C3 must be connected with AC adapter (ZW-10AC) which provides the power supply for ZW-101PG1.



Note-1: Please be sure to use ZW-10AC for ZW-10C3.
If another AC adapter is used, ZW-101PG1 will be destroyed.

■Cautions on the operation

General information on the **ZW-101PG1**'s operation is expressed on the operation manual.

ZW-101PG1 regards W10 as W16/51 (2.5kW).

Please observe these cautions expressing below before the operation.

●Programming

(1) Adress 3600-4777 are always displayed NOP state. You cannot enter any program in this area.

(2) When a basic instruction with exclusive w16/51's number(*) is entered, the display of data display area is disappear.

(*) ⎧Relay number : 4000, 7000, etc.
⎩TMR,CNT number: 100, 177, etc.
These number are not available for W10.

(Example-1)

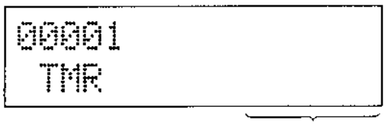
STR 4 0 0 0 ENT

```
00000
 STR
```

disappear

(Example-2)

TMR 1 0 0 ENT

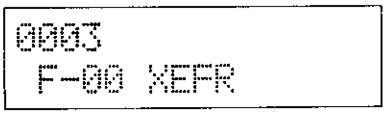


```
00001
 TMR
```

disappear

(3) When an exclusive W16/51's register (2700,b300,9300,etc.) is entered, data display area displays ⊐777.

FUN ENT



```
0003
 F-00 XEFR
```

STEP (+) 7 0 0 ENT

```
0004
                ⊐777
```

(4) When an exclusive W16/51's application instruction (F-05,F-15,etc.) is entered, entered one word is displayed as F-99.

(Example-4)

FUN 5 ENT

```
0006
 F-99
```

However, when F-20(MD) is entered, "M-99 0"(same meaning with F-99) is displayed.

(Caution)

When you change to "RUN" mode (Monitor, change) afte the operation of (2),(3),(4) expressed above, memory failure will occur.

These memory failured adress is stored at system memory #054 and #055.

(Example)

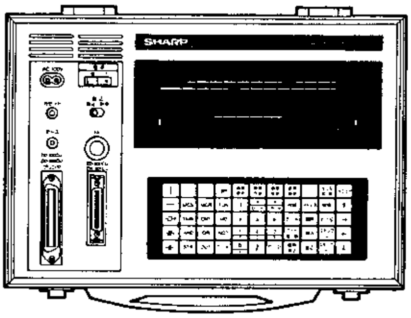
In case of memory failure on adress 1234

```
7       #055      0 7       #054      0
|0 0 0 0 0 0 1 0|1 0 0 1 1 1 0 0|
         \_1_/ \_2_/ \_3_/ \_4_/
```

● Monitor

Every exclusive W16/51's data memory area (4000, ⊐700,etc.) are always clear. So you can not set/reset or change the current value on these data memory.

10-2 Ladder processor (Z-100LP1)



Major facility
◦Ladder programming
◦Instruction words programming
◦Monitor
◦On-line transfer
◦CMT transfer
◦Printout

■Connection with the base module

The programmer interfacing connector of the base module must be connected with the connector for ZW-501CU of Z-100LP1 using the adapting cable for support tool (ZW-10C3).

■Cautions on the operation

General information on the **Z-100LP1**'s operation is expressed on the operation manual.

Please setup the mode expressing below.

Model : W16/51
Memory capacity: 2.5kW

(1) In case of on-line transfer from Z-100LP1 to W10, please be sure to enter "END" instruction (F-40) at adress 3577 before programming.

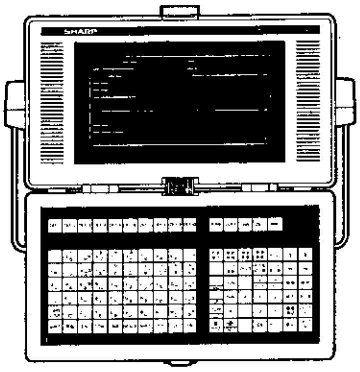
(2) When some instructions exist at adress 3500-4777 of Z-100LP1, after the on-line transfer to W10, "Verify error" occur, but there is no problem on W10's running.

(3) When exclusive W16/51's instruction (F-05,F-15,etc.) or data memory (⊐200,9300,etc.) are programmed and transfer to W10, "Memory failure" occur. This memory failured adress data is stored at system memory #054 and #055. (Please refer to 10-1 Programmer (ZW-101PG1))

(4) You can not operate the CMT transfer between W10 and Z-100LP1 using ZW-10PG1, but ZW-101PG1 is available.

(5) The facility of system memory (output retention) #203 is different between W16/51 and W10. So when you set "001" on #203 for W10, please neglect the display "Output retention 0010-0777".

10-3 Ladder processor II (Z-100LP2)



Major facility
∘ Ladder programming
∘ Instruction words programming
∘ Monitor
∘ On-line transfer
∘ CMT transfer
∘ Printout
∘ Loader transfer
∘ PROM writer transfer
∘ Edit

■Connection with the base module

The programmer interfacing connector of the base module must be connected with the RS-422 connctor of Z-100LP2 using the adapting cable for support tool (ZW-10C3).

■Cautions on the operation

The operation of Z-100lp2,Z-1LP2EM (Expansion module) are expressed on each manuals.

Please setup the mode below expressed.

Model : W16/51
memory capacity: 2.5kW

(1) In case of on-line transfer from Z-100LP2 to W10, please be sure to enter "END" instruction (F-40) at adress 3577 before programming.

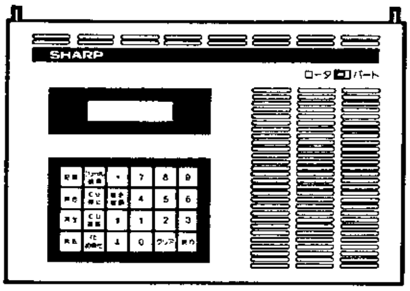
(2) When some instructions exist at adress 3600-4777 of Z-100LP2, after the on-line transfer to W10, "Verify error" occur, but there is no problem on W10's running.

(3) When exclusive W16/51's instruction (F-05,F-15,etc.) or data memory (200,9300,etc.) are programmed and transfer to W10, "Memory failure" occur. This memory fdailured adress data is stored at system memory #054 and #055. (Please refer to 10-1 Programmer (ZW-101PG1))

(4) You can not CMT transfer between W10 and Z-100LP2 using ZW-10PG1, but ZW-101PG1 is available.

(5) The facility of system memory (output retention) #203 is defferent between W16/51 and W10. So when you set "001" on #203 for W10, Please neglect the display "Output retention 0010-0777".

10-4 CF Loader (ZW-100CF1)



Major facility
∘ FD formatting
∘ Save
∘ Verify
∘ Load
∘ Search
∘ Erase

■ Connection with the base module

The programmer interfacing connector of the base module must be connected with the rs-422 connector of Z-100LP2 using the adapting cable for support tool (ZW-10C3).

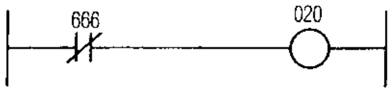
■ Cautions on the operation

The operation of ZW-100CF1 is expressed on the operation manual.
remote/loader select switch of ZW-100CF1 is set at "Loader".
ZW-100CF1 regards W10 as W16/51 (2.5kW).

§11 Programming Examples

11-1 Basic instruction applied circuit

[1] Ever-on circuit

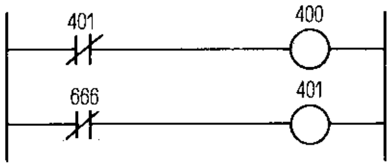


```
STR   NOT    666
OUT          020
```

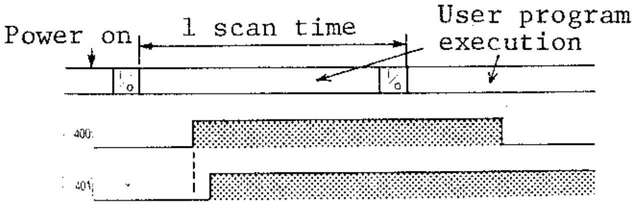
° Because the special relay 666 is the contact that keeps OFF at all times, it makes the coil 020 always ON. If 020 has been programmed to the area which becomes OFF during halt, it can be used as the RUN indicator for the W10 when the programmable controller stops after a mode change to the program mode or as a result of self-diagnosis.

NOTE-1: It is possible to use the system memory #203 for the output retain area during stop of the programmable controller. It has to be programmed in term of absolute address. When the system memory has been cleared, it goes into the state of 000($_8$) which makes all output units set in the retain in the stop state. (The system memory has been cleared when the programmable controller left the factory.)

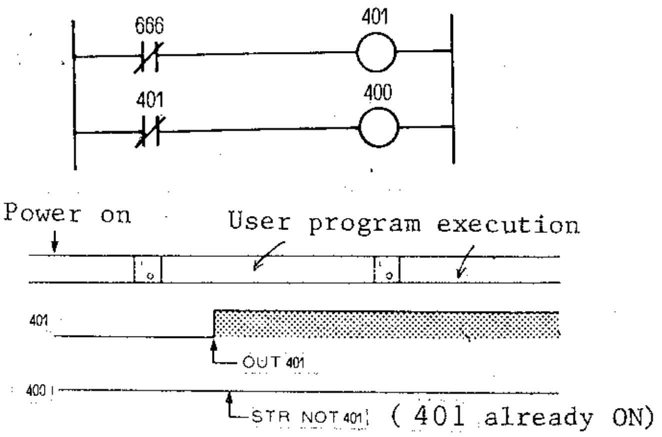
[2] Circuit from which a single pulse is generated at power on



```
STR   NOT    401
OUT          400
STR   NOT    666
OUT          401
```

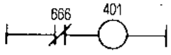
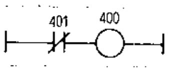


NOTE-1: No pulse is generated when the program sequence is changed.

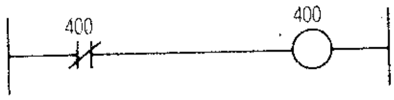


NOTE-2: No pulse is generated when 401 is in the keep function assigned area. (The keep function area is set in the system memory #200.) (Reason) 401 turns ON upon preceding power on and retains ON during a power failure time.

NOTE-3: This pulse is used to clear registers at power on and to preset constants. The above program has to be written before the instruction by which register is cleared.

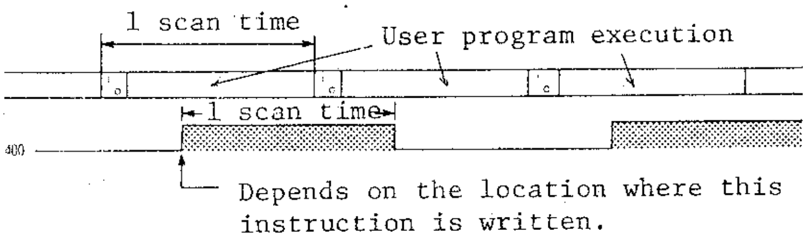
(However,  may be written any place after  .)

[3] Oscillation circuit



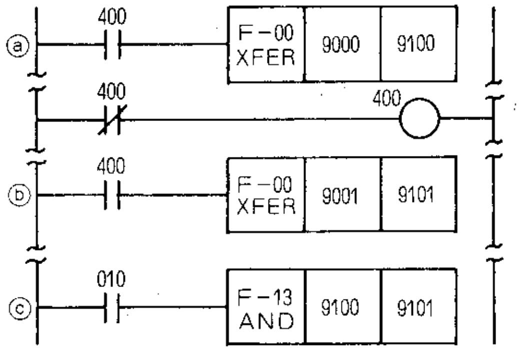
```
STR   NOT       400
OUT             400
```

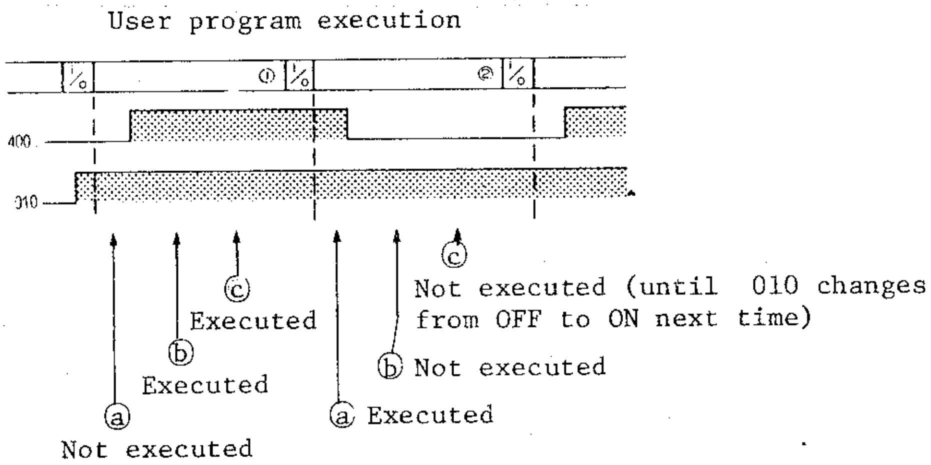
° ON and OFF are repeated at every scan.



This pulse is used for the basic clock of the blink circuit or operation start signal at every other scan.

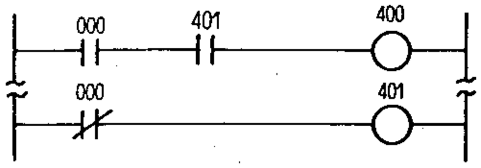
NOTE-1: At which point of the scan cycle it changes from ON to OFF or OFF to ON depends on where the instruction is written in program steps. Caution is required when this pulse is to be used for the start signal of operation.





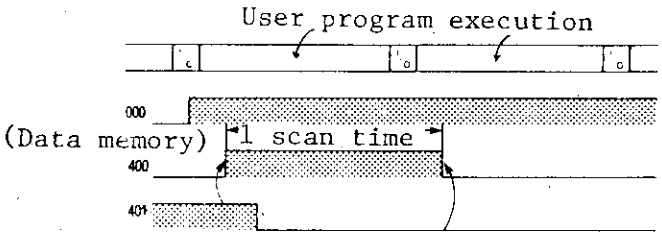
When the operation ⓒ is executed in the scan cycle of the user program execution ①, the result of ⓐ executed in scan cycle one step before is used for the operation of ⓒ because ⓐ was not executed in this scan cycle.

[4] Rise edge differentiation



```
STR        000
AND        401
OUT        400
STR  NOT   000
OUT        401
```

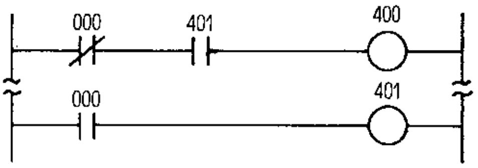
° When 000 changes from OFF to ON, 400 goes ON for a period of one scan time.



NOTE-1: Use of F-44 completes the operation by one instruction.

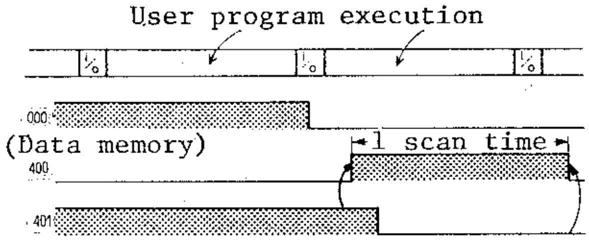
NOTE-2: No pulse is generated when the program sequence is changed.

[5] Fall edge differentiation



STR NOT	000
AND	401
OUT	400
STR	000
OUT	401

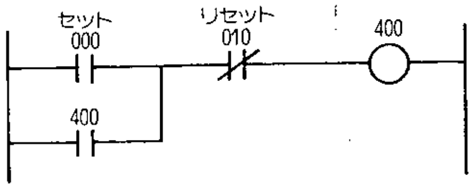
° When 000 changes from ON to OFF, 400 goes ON for a period of one scan time.



NOTE-1: Use of F-45 completes the operation by one instruction.

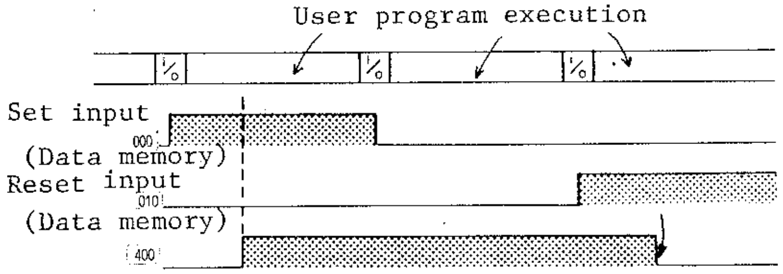
NOTE-2: No pulse is generated when the program sequence is changed.

[6] Self-hold circuit (with reset in priority)



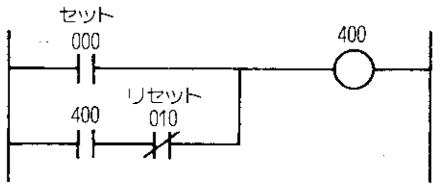
STR	000
OR	400
AND NOT	010
OUT	400

° When the set input is turned ON once with the reset input in ON state (continuing in the ladder chart), the state is retained even after the set input went OFF, until the reset input is turned ON or power is turned off.



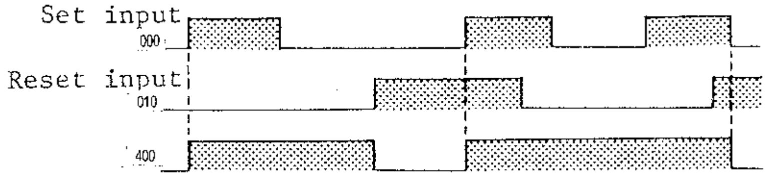
NOTE-1: When the data memory in the keep function assigned area is used for an output, the state immediately before a power failure can be retained, provided that the normally open contact is used for the external contact of the reset input and made AND NOT program-wise. If the normally closed contact is used for the external contact and AND is used by program, the self-hold will be reset when the input power supply is shut off before the power supply of the W10 .

[7] Self-hold circuit (with set in priority)



```
STR          000
STR          400
AND NOT      010
OR STR
OUT          400
```

- When the set input is turned ON once regardless of ON and OFF state of the reset input, the output is turned ON and the state is retained even after the set input gone OFF.
- Even if the reset input is turned OFF (non-continuing in the ladder chart) with the set input in the ON state, the output retains the ON state with reset being invalid.
- Output will be turned OFF when the reset is turned ON or power is turned off with the set input in the OFF state.



NOTE-1: When the data memory in the keep function assigned area is used for an output, the state immediately before a power failure can be retained, provided that the normally open contact is used for the external contact of the reset input and made AND NOT program-wise.

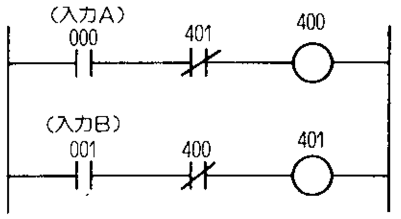
- If the normally closed contact is used for the external contact and AND is used by program;

① IF the input power supply should shut off before the W10 power supply when a power failure is encountered while the set input is OFF, reset will be carried out even if the reset external contact is closed.

② If the input power supply should rise later than the W10 power supply when a power failure is restored while the set input is OFF, reset will be carried out even if the reset external contact is closed.

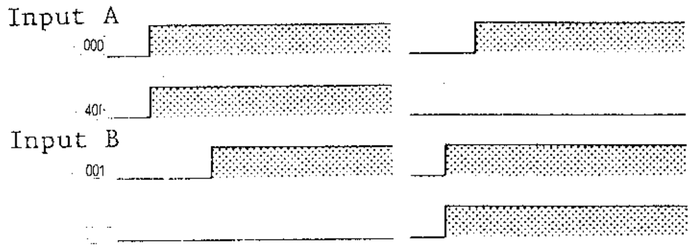
[8] Priority circuit

(a) In case input is a level signal



```
STR          000
AND  NOT     401
OUT          400
STR          001
AND  NOT     400
OUT          401
```

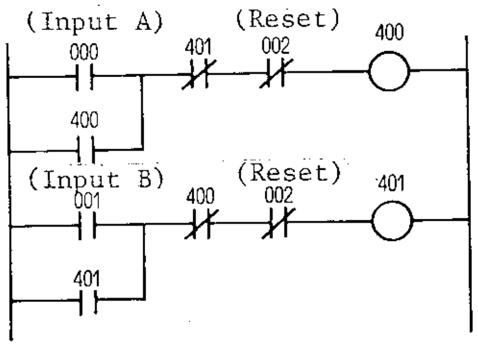
° Input A or input B whichever comes first takes preference over the other and the input that follows will be treated invalid.



Input A comes first   Input B comes first

NOTE-1: If the input A and input B went ON in the same scan cycle, the one appearing first in the program takes preference over the other.

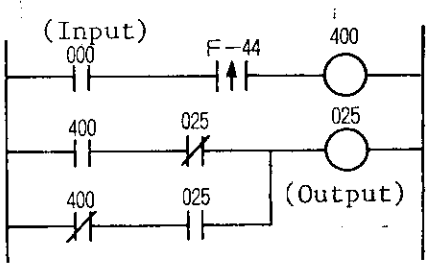
(b) In case input is a pulse signal



```
STR          000
OR           400
AND  NOT     401
AND  NOT     002
OUT          400
STR          001
OR           401
AND  NOT     400
AND  NOT     002
OUT          401
```

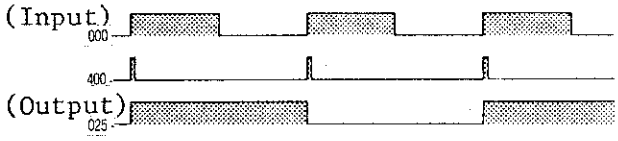
° This signal is used for the output that should not turn ON at the same time (forward/reverse rotation of the motor and such).

[9] Alternate circuit

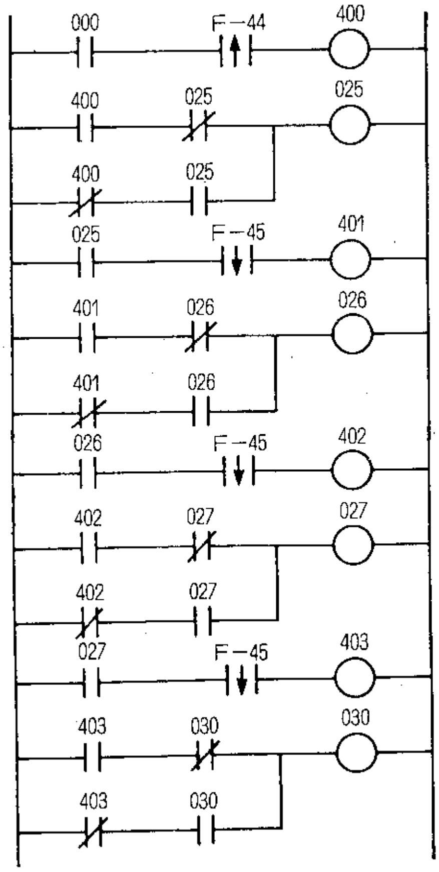


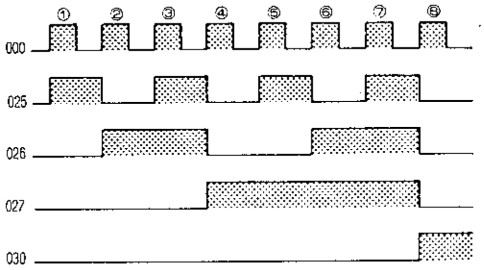
```
STR                000
F-44
OUT                400
STR                400
AND      NOT       025
STR      NOT       400
AND                025
OR       STR
OUT                025
```

- Each time the input goes ON, it makes the output inverted. It is possible to produce an alternate output when connected with the contact of the momentary switch.



- When this circuit is used repeatedly for n-times, it will constitute the divider circuit of n-stages.

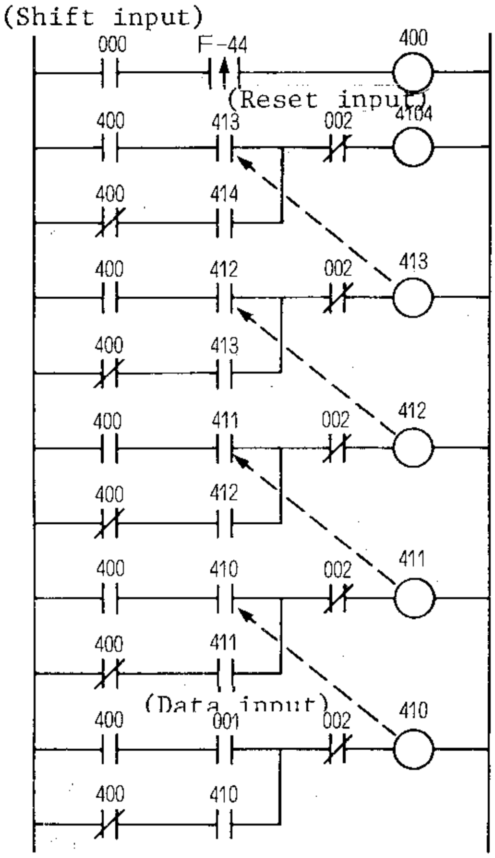




(NOTE) F-45 (fall edge differentiation) after the second stage.

(REFERENCE) When output, 025 ∿ 030, is used, it constitute the binary counter.

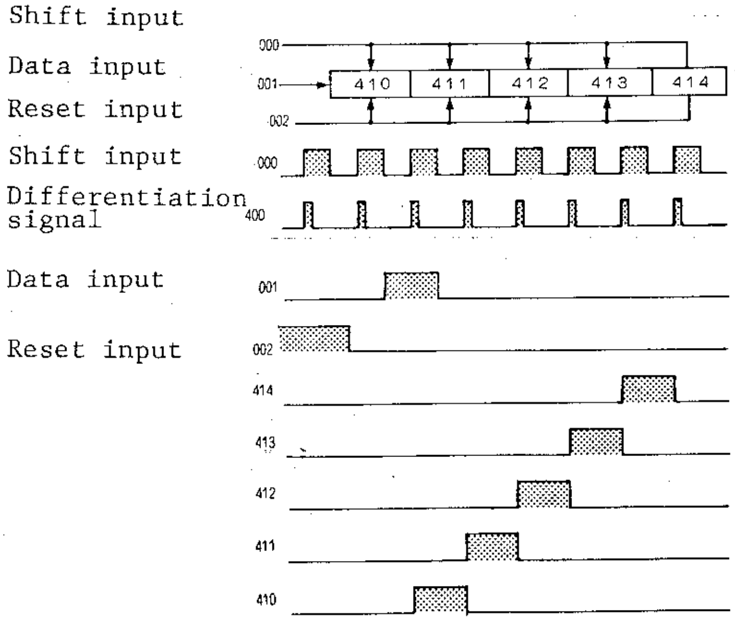
[10] n-bit shift register



Rise edge differen-tiation

```
STR           000
F-44
OUT           400
STR           400
AND           413
STR   NOT     400
AND           414
OR    STR
AND   NOT     002
OUT           414
```

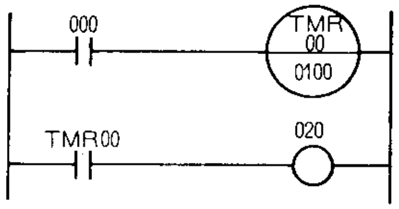
0000 Shift input
0001 Data input
0002 Reset input



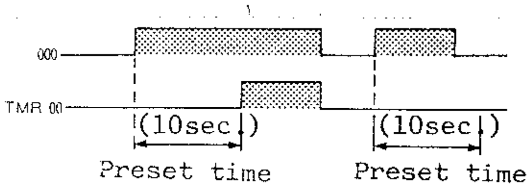
- ° Use of F-60 enables to constitute the shift register by one instruction.
- ° To retain the shift state at a time of power failure, it becomes necessary to use the keep function assigned data memory area for 410 ∿ 414.

11-2 Timer application circuit

[1] On delay timer



```
STR              000
TMR               00
                0100
STR      TMR      00
OUT              020
```



° Output turns ON with a delay by the preset time after the input turns ON. If the input ON time is less than the preset time, the output would not turn ON.

° When the input turned OFF, it also makes the output turned OFF.

NOTE-1: By means of the system memory #201, it would be possible to make choice whether the current value be reset or retained at a time of power failure.

#201 programmed
- 000($_8$) The current value is reset (the preset value is assumed).
- 001($_8$) The current value is retained.

NOTE-2: If a power failure is met while the input is ON, the state of the output upon power recovery differs depending on how #201 was programmed.

(a) #201 is 000($_8$) —— Current value is reset

After power recovery, the output turns ON with a delay by the preset time.

(b) #201 is 001($_8$) —— Current value is retained

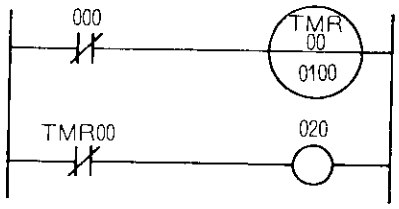
(b)-1 If time was up before the power failure:

After power recovery, the output turns ON in the first scan cycle.

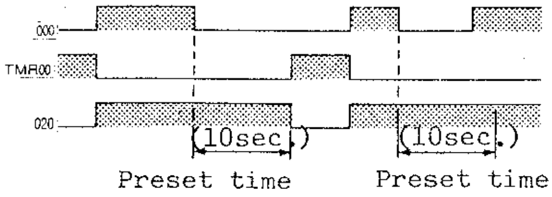
(b)-2 If time was not up before the power failure:

After power recovery, the output turns ON with a delay by the time of (preset value - current value at the time of power failure).

[2] Off delay timer



```
STR    NOT          000
TMR                  00
                   0100
STR    NOT   TMR     00
OUT                 020
```



° Output turns OFF with a delay by the preset time after the input turns OFF. If the input OFF time is less than the preset time,the output would not turn OFF.

° When the input turned ON, it also makes the output turned ON.

NOTE-1: If a power failure is met while the input is OFF with the timer input in the ON state, the state of the output upon power recovery differs depending on how #201 was programmed.

(a) #201 is 000($_8$) —— Current value is reset

After power recovery, the output turns ON for a period of the preset time.

(b) #201 is 001($_8$) —— Current value is retained

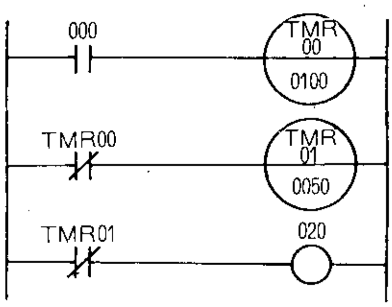
(b)-1 If time was up before the power failure:

The output would not turn ON after power recovery.

(b)-2 If time was not up before the power failure:

After power recovery, the output turns ON for a period of (preset value - current value at the time of power failure).

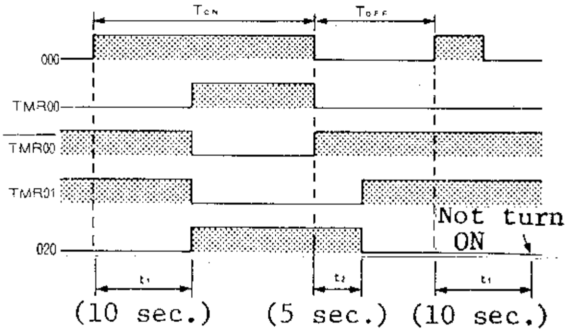
[3] On/off delay timer



On delay setup t_1

Off delay setup t_2

```
STR                   000
TMR                    00
                     0100
STR     NOT    TMR     00
TMR                    01
                     0050
STR     NOT    TMR     01
OUT                   020
```



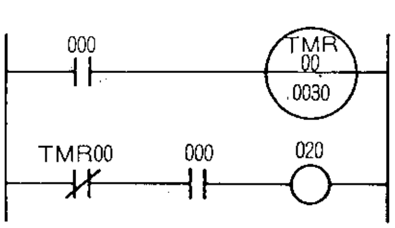
° The output turns ON with a delay of t_1 after the input has turned ON, and the output turns OFF with a delay of t_2 after the input has turned OFF.

If the input ON time (Ton) is smaller than the on delay preset time (t_1), the output would not turn ON.

° If the input OFF time (Toff) is smaller than the off delay programmed time (t_2), the output keeps its ON state.

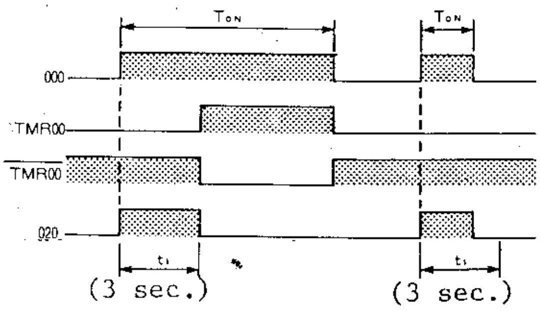
NOTE-1: For state of the output at a time of power failure, refer to [2] "Off delay timer NOTE".

[4] One shot timer at the input rise edge (1)



On delay setup t_1

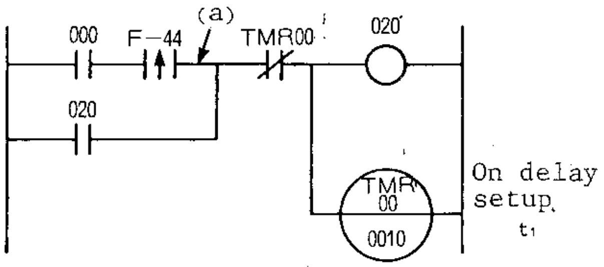
```
STR                   000
TMR                    00
                     0030
STR     NOT    TMR     00
AND                   000
OUT                   020
```



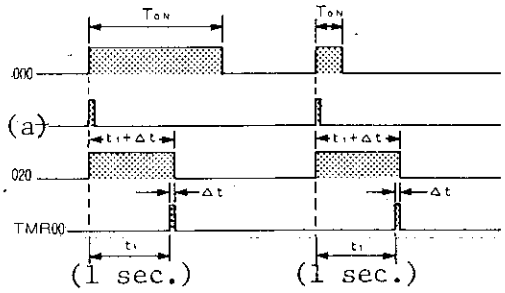
° At rising edge of the input signal (OFF to ON), a pulse whose width is the preset time (t_1) is issued.

° If the input ON time (Ton) is smaller than the preset time (t_1), the width of the output pulse will be Ton.

[5] One shot timer at the input rise edge (2)



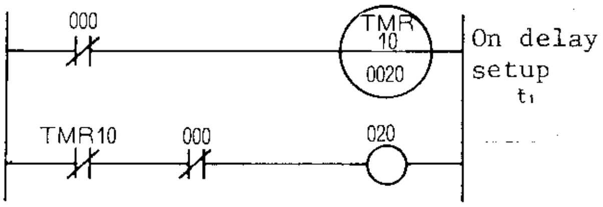
```
STR                   000
F-44
OR                    020
AND    NOT   TMR       00
OUT                   020
TMR                    00
                     0010
```



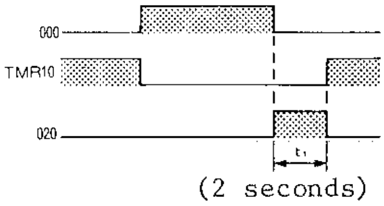
At rising edge of the input signal (OFF to ON), a pulse whose width is the (preset time $t_1+\Delta t$) is issued.
Where, Δt is one scan time

Irrespective of the input ON time (Ton), the output pulse width of the output will be $t_1+\Delta t$.

[6] One shot timer at the input fall edge



```
STR    NOT            000
TMR                    10
                     0020
STR    NOT   TMR       10
AND    NOT            000
OUT                   020
```



° At falling edge of the input signal (ON to OFF), a pulse whose width is the preset time issued.

NOTE-1: When a power failure is met while the input is OFF, the output after power recovery may differ depending on how the system memory #201 was programmed.

(a) #201 is 000($_8$) —— Current value is reset

After power recovery, the output turns ON for a period of the preset time.

(b) #201 is 001($_8$) —— Current value is retained

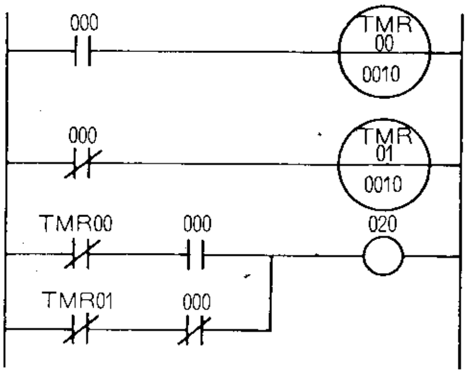
(b)-1 If time was up before the power failure:

The output would not turn ON after power recovery.

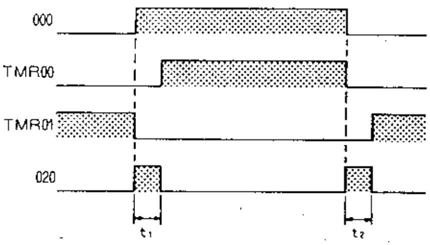
(b)-2 If time was not up before the power failure:

After power recovery, the output turns ON for a period of (preset value - current value at the time of power failure).

[7] One shot timer at rise/fall edge

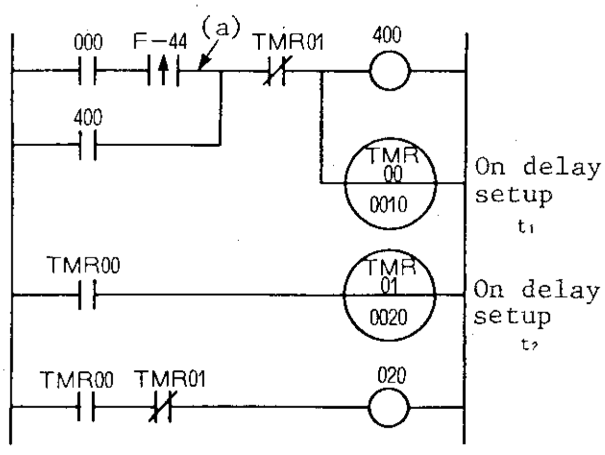


```
On delay   STR                 000
setup      TMR                  00
  t1                          0010
Off delay  STR   NOT           000
setup      TMR                  01
  t2                          0010
           STR   NOT   TMR      00
           AND                 000
           STR   NOT   TMR      01
           AND   NOT           000
           OR    STR
           OUT                 020
```

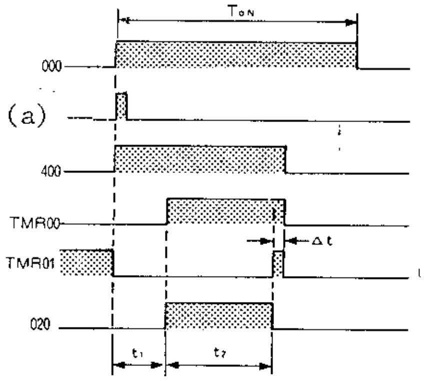


- ° At rising and falling edge of the input, output signals (t1 and t2) are turned ON, respectively.
- ° This signal is used to detect a change in the state of the input.
- ° Also used to multiply input pulse frequency.

[8] On delay one shot timer



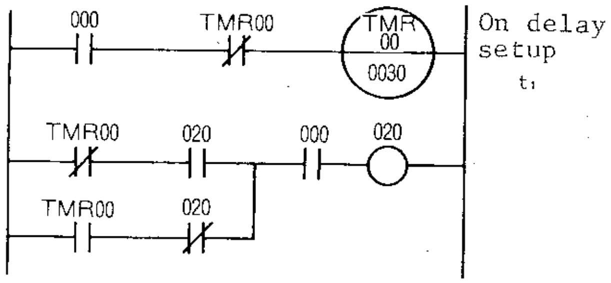
```
STR                    000
F-44
OR                     400
AND    NOT    TMR       01
OUT                    400
TMR                     00
                      0010
STR           TMR       00
TMR                     01
                      0020
STR           TMR       00
AND    NOT    TMR       01
OUT                    020
```



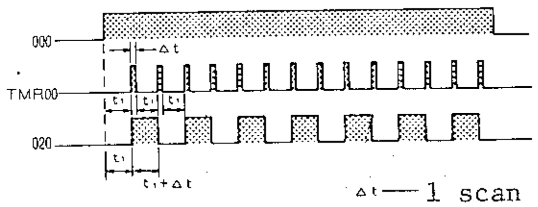
Δt——1 scan time

- The output having the pulse width of t_2 is issued with a delay of preset value t_1 from the rise edge of the input.
- Change of the input during the time (t_1+t_2) is disregarded.

[9] Equispace pulse generation circuit



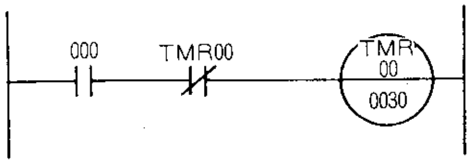
```
STR                    000
AND    NOT    TMR       00
TMR                     00
                      0030
STR    NOT    TMR       00
AND                    020
STR           TMR       00
AND    NOT             020
OR     STR
AND                    000
OUT                    020
```



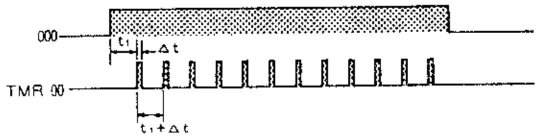
Δt— 1 scan time

° While the input is ON, the pulse whose ON and OFF periods are equal (50% duty cycle) is issued. ON and OFF period may be set as desired using the TMR preset value (t_1). The pulse width shall be $t_1+\Delta t$.

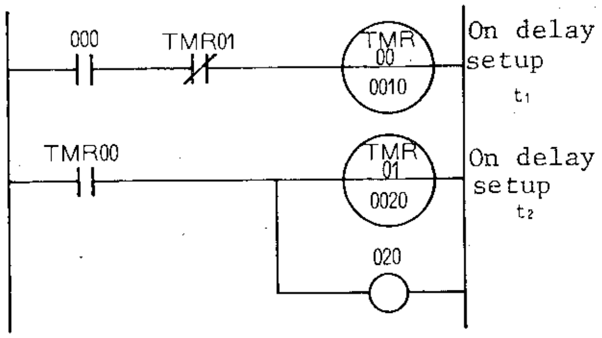
REFERENCE: The pulse having Δt for the ON period and t_1 for the OFF period can be obtained in a manner as described below.



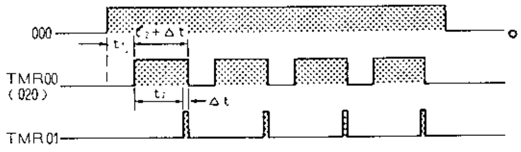
```
STR                  000
AND    NOT    TMR     00
TMR                   00
                    0030
```



[10] Duty variable pulse generator circuit (1)



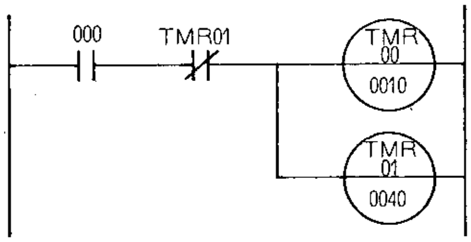
```
STR                  000
AND    NOT    TMR     01
TMR                   00
                    0010
STR           TMR     00
TMR                   01
                    0020
OUT                  020
```



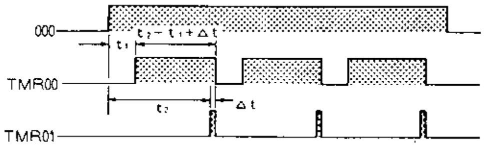
° When the input is ON, the pulse having $t_2+\Delta t$ for the ON period and t_1 for the OFF period is issued.

Δt— 1 scan time

[11] Duty variable pulse generator circuit (2)



```
On delay   STR                  000
setup      AND     NOT   TMR    01
  t1       TMR                  00
                                0010
On delay
setup      TMR                  01
  t2                            0040
```

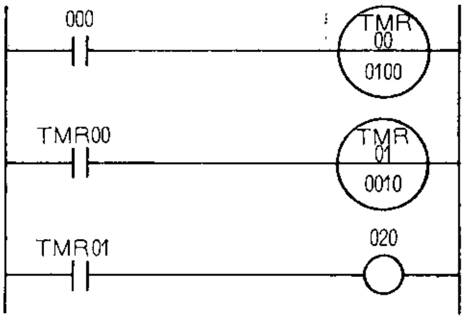


Δt 1 scan time

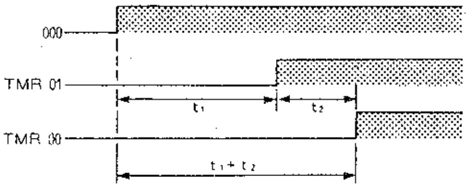
° When the input is ON, the pulse having $t_2-t_1+\Delta t$ for the ON period and t_1 for the OFF period is issued.

NOTE-1: t_1 must be smaller than t_2.

[12] Long time timer (1)

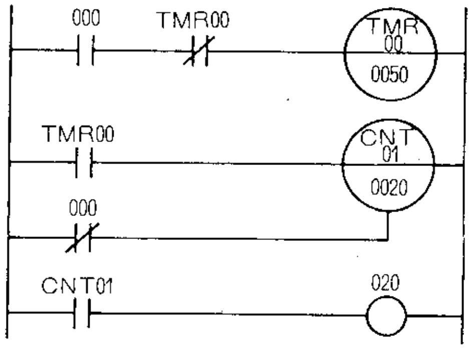


```
On delay   STR                  000
setup      TMR                  00
  t1                            0100
On delay   STR           TMR    00
setup      TMR                  01
  t2                            0010
           STR           TMR    01
           OUT                  020
```



° After the input turned ON, the output becomes ON with a delay of t_1+t_2.

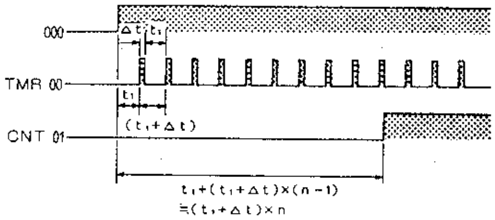
[13] Long time timer (2)



On delay setup t_1

Counter setup n

```
STR                000
AND   NOT   TMR     00
TMR                 00
                  0050
STR         TMR     00
STR   NOT          000
CNT                 01
                  0020
STR         CNT     01
OUT                020
```

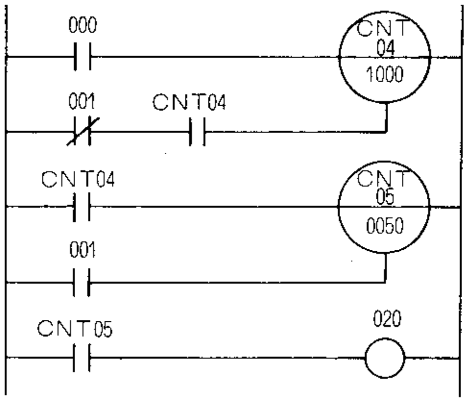


After the input turned ON, the output becomes ON with a delay of $(t_1+\Delta t)\times n$.

Δt—1 scan time

11-3 Counter applications

[1] Large capacity counter (1)



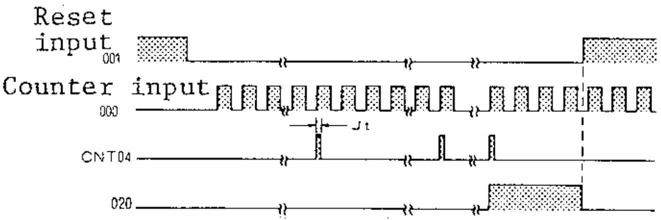
Preset value n_1

Preset value n_2

```
STR                000
STR   NOT          001
AND         CNT     04
CNT                 04
                  1000
STR         CNT     04
STR                001
CNT                 05
                  0050
STR         CNT     05
OUT                020
```

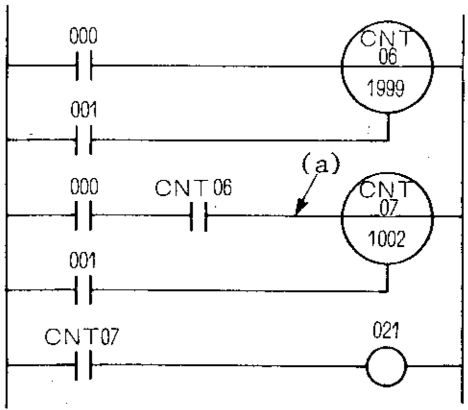
Preset value 1000×50=50000
(CNT is ON reset)

- When the preset value exceeds 1999, programming it in the above manner will realize the counter of the preset value (n_1xn_2).



Δt 1 scan time

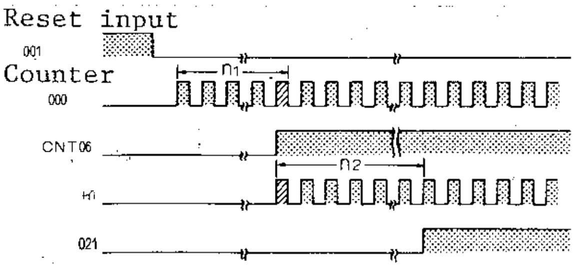
[2] Large capacity counter (2)



```
Preset       STR          000
value n1     STR          001
             CNT           06
                         1999
Preset       STR          000
value n2     AND     CNT   06
             STR          001
             CNT           07
                         1002
             STR     CNT   07
             OUT          021
```

Preset value 1999+1002−1=3000
(CNT is ON reset)

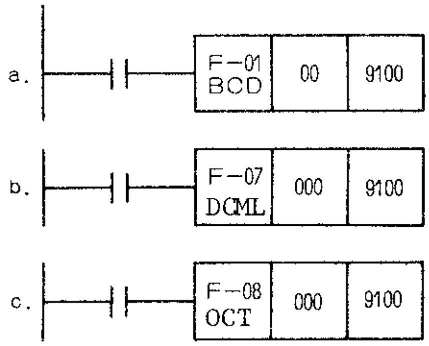
° It establishes the counter of the preset value (n_1+n_2-1).



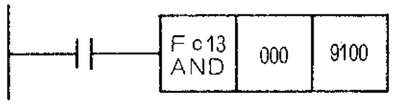
11-4 Data processing instruction applications

[1] Register clear

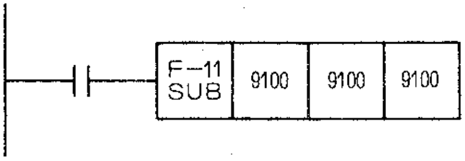
(1) Transfer of constant 0



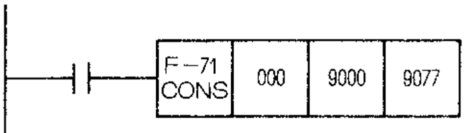
(2) AND with constant 0



(3) Subtract the same register

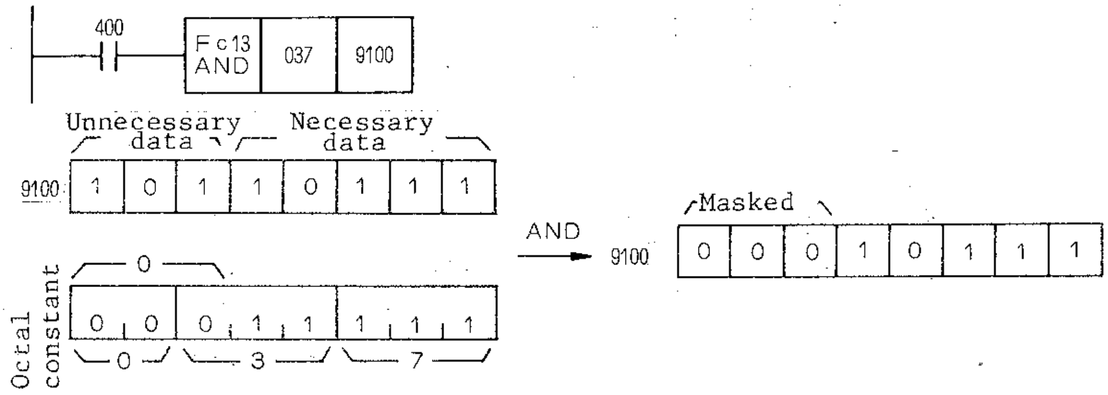


(4) Clear n-byte



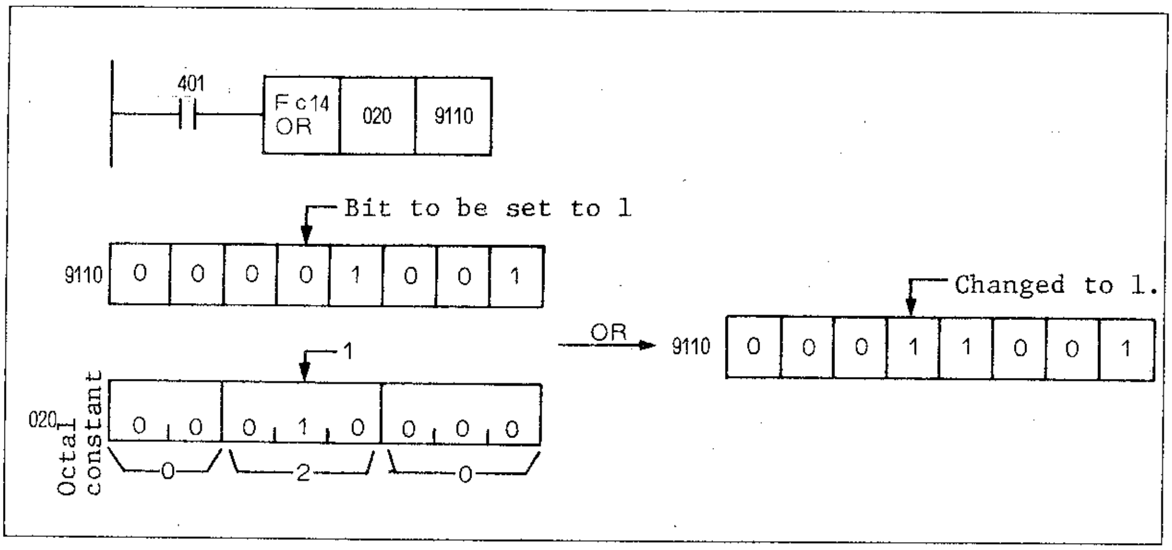
Clear 64 bytes of 9000∿9077.

[2] Masking data



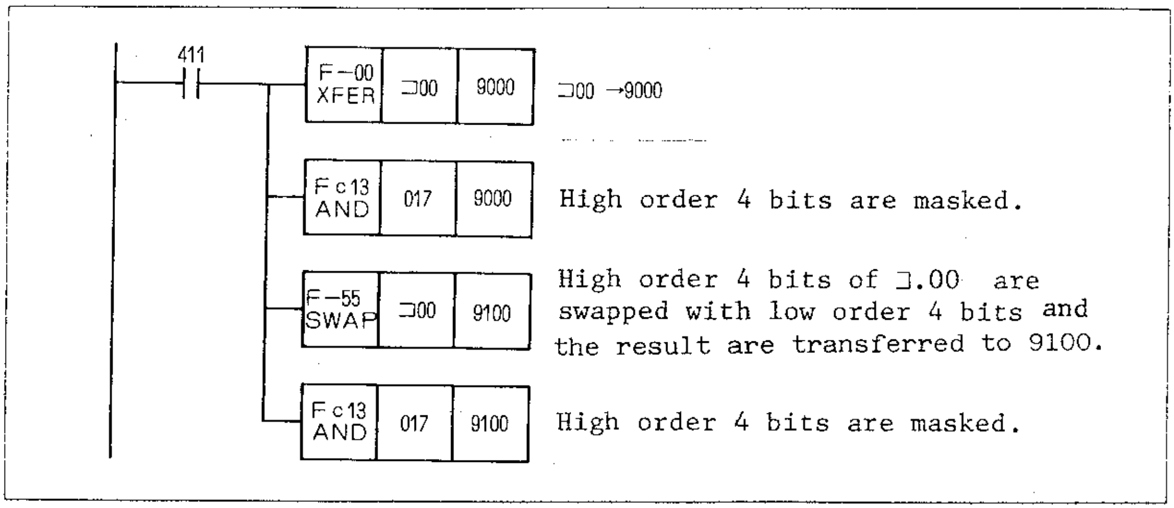
° Any desired bit is set to 0 out of an 8-bit data.

[3] Bit insertion



° Any desired bit is set to 1 out of an 8-bit data.

[4] Decomposition of a number



⊐00 →9000

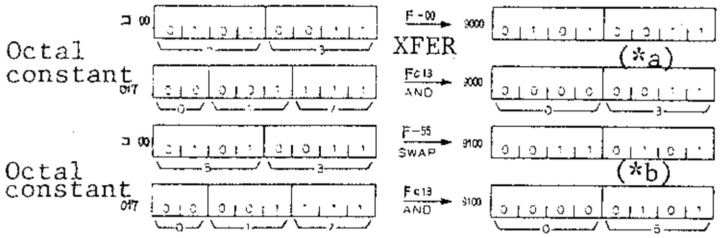
High order 4 bits are masked.

High order 4 bits of ⊐.00 are swapped with low order 4 bits and the result are transferred to 9100.

High order 4 bits are masked.

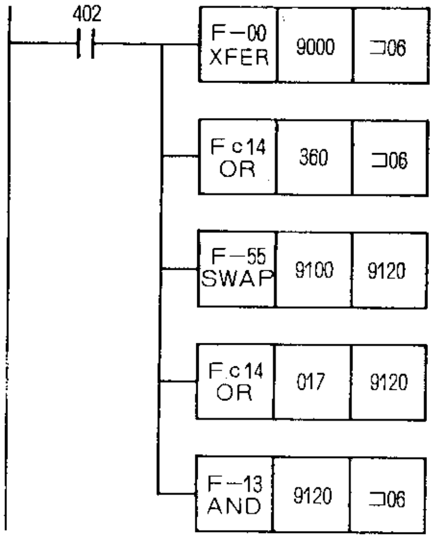
° A 2-digit BCD number received from the input unit is decomposed into single digit numbers and stored in separate register. In the example, low order 4 bits of ⊐.00 are transferred to 9000 and high order 4 bits to 9100.

° If two digits are used for input unit in inputting multiple number of 1-digit BCD numbers, it saves output units.



(*a): (Low order 4 bits of ⊐.00)
(*b): (High order 4 bits of ⊐.00)

[5] Composition of a number



9000→ ⊐06

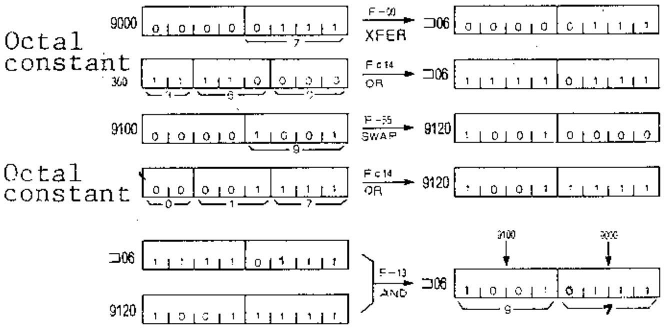
High order 4 bits are set to 1.

High order 4 bits of 9100 are swapped with low order 4 bits and the result are transferred to 9120.

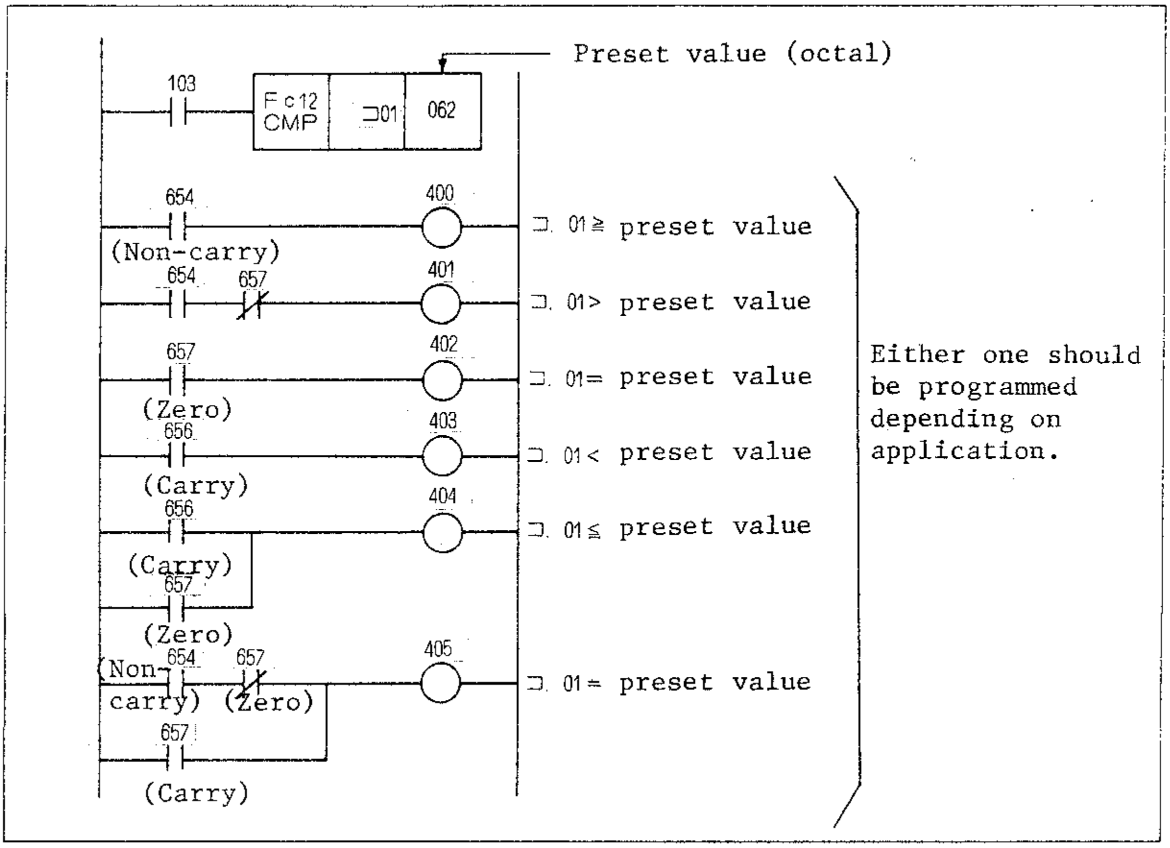
Low order 4 bits of 9120 are set to 1.

9120 is composed with ⊐. 06 .

- ° 1-digit BCD numbers in two registers are outputted to one output unit. In the example, low order 4 bits of 9000 and 9100 are outputted to ⊐. 04.
- ° If two digits are used for output unit in outputting multiple number of 1-digit BCD numbers, it saves output units.

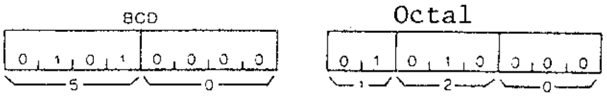


[6] Comparison with preset value

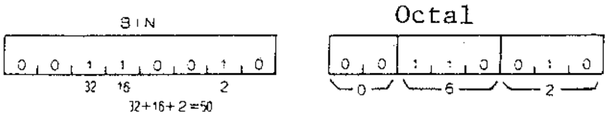


° The preset value must be programmed in an octal number.

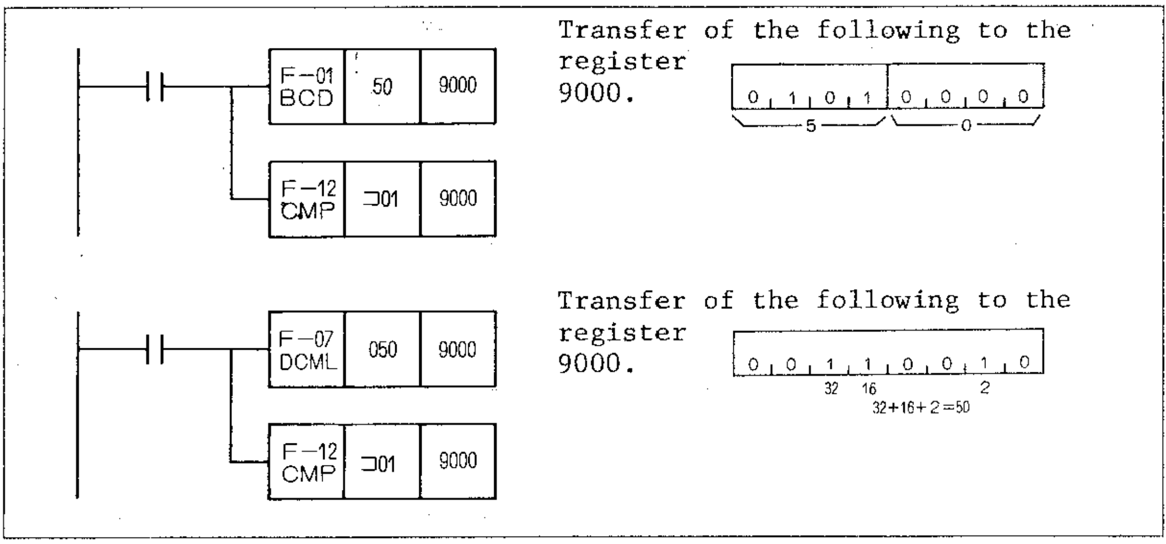
(Ex-1) If the data in the register is a BCD number [preset value: 50(BCD]



(Ex-2) If the data in the register is a binary number [preset value: 50(binary)]

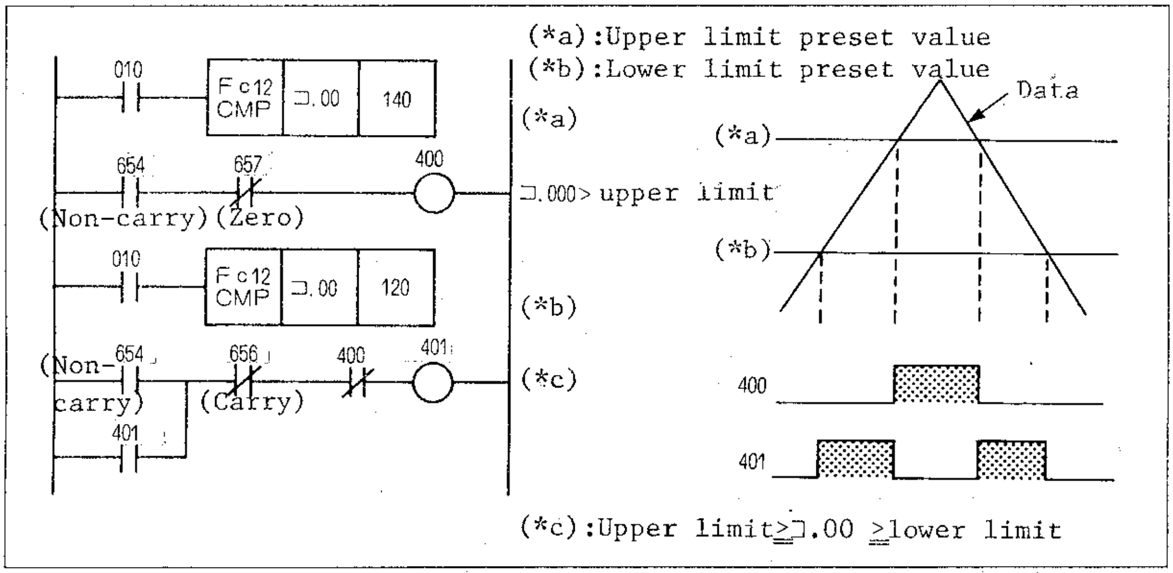


REFERENCE: To program the preset value in BCD and binary


Transfer of the following to the register 9000.

Transfer of the following to the register 9000.

[7] Window comparator


(*a):Upper limit preset value

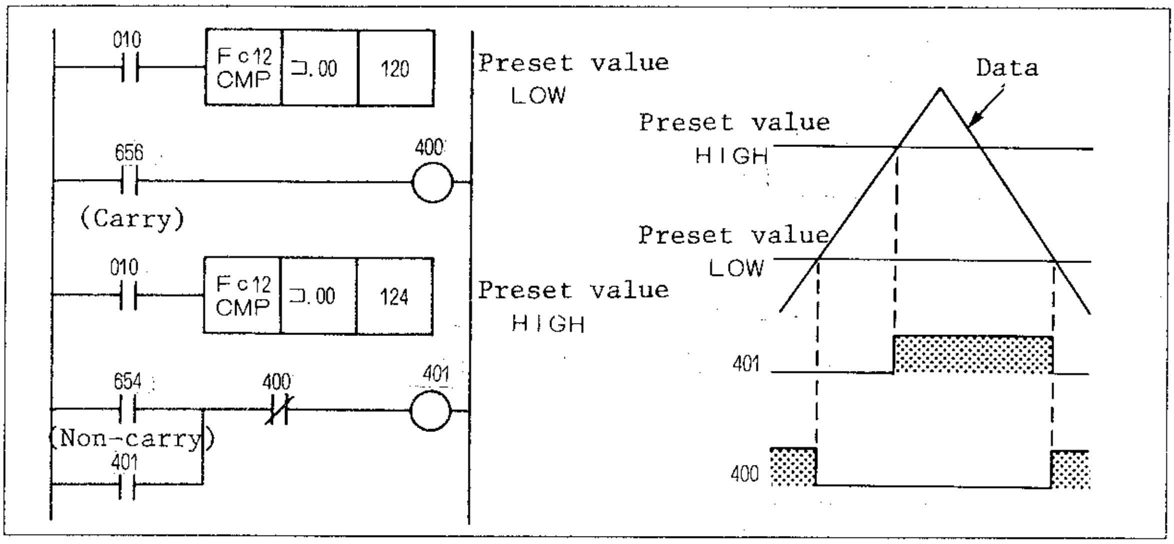
(*b):Lower limit preset value

(*c):Upper limit≥⊐.00 ≥lower limit

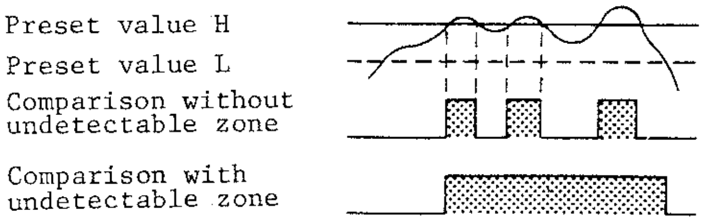
° When data is within the lower limit preset value and the upper limit preset value, it makes 401 turned ON. This can be used for grading the acceptable from the unacceptable item (GO/NO GO).

° In the above example, the upper limit preset value is set to 140($_8$) (60 in BCD) and the lower limit preset value to 120($_8$) (50 in BCD). It makes 401 turned ON when 60≥⊐.00 ≥50.

[8] Comparator circuit having undetectable zone

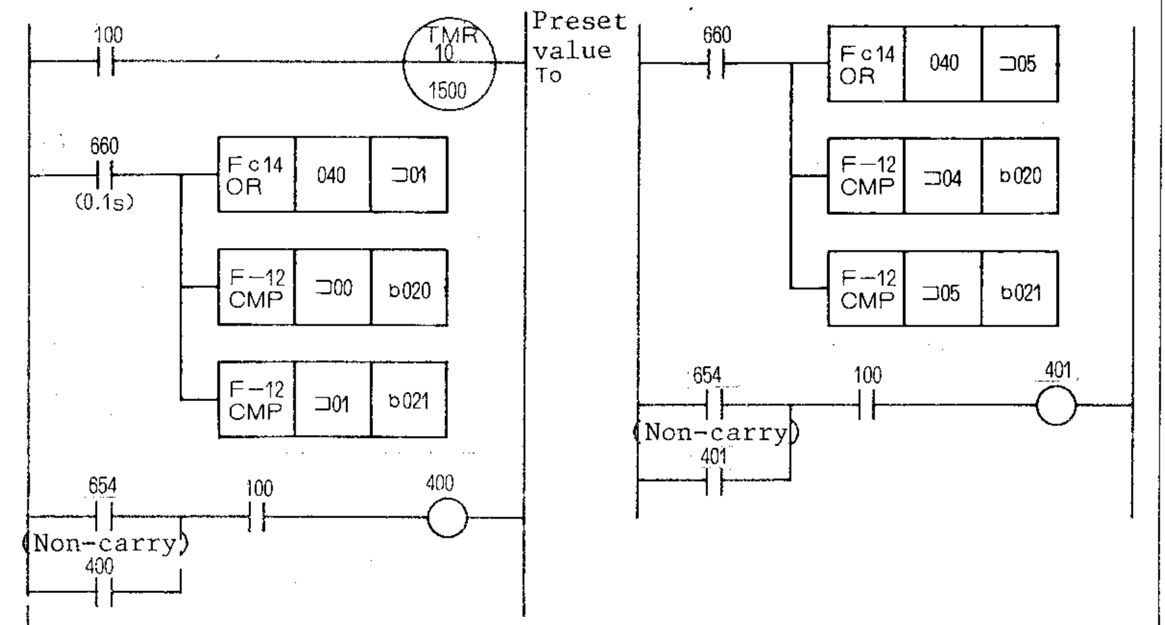


° When the output from the A/D converter is taken into the input unit, and compared a slight fluctuation in the analog signal makes the A/D converter output affected so that the comparison result by the sequencer may turn ON and OFF. Use of this program permits the comparison without influence by the fluctuation in the A/D converter low order bits.


Preset value H
Preset value L
Comparison without undetectable zone
Comparison with undetectable zone

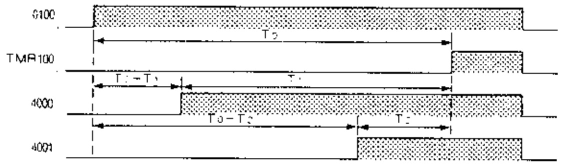
° In the above example, the high side preset value is set to 124(8) (54 in BCD) and low side preset value to 120(8) (50 in BCD). 401 turns ON when the contents of ⊐.00 (to be a BCD code) is equal to or larger than 54, but it retains the ON state until the contents of ⊐.00 becomes smaller than 50, once after it has turned ON.

[9] Timer that has multiple number of set points



コ. 01, コ. 00 -preset value T_1, b.020, b.021... Register that stores
コ. 05, コ. 04 -preset value T_2, the TMR 100 current value

° By comparing the TMR current value with the preset value in the register (or input unit), it realizes the timer that has a multiple number of set points.



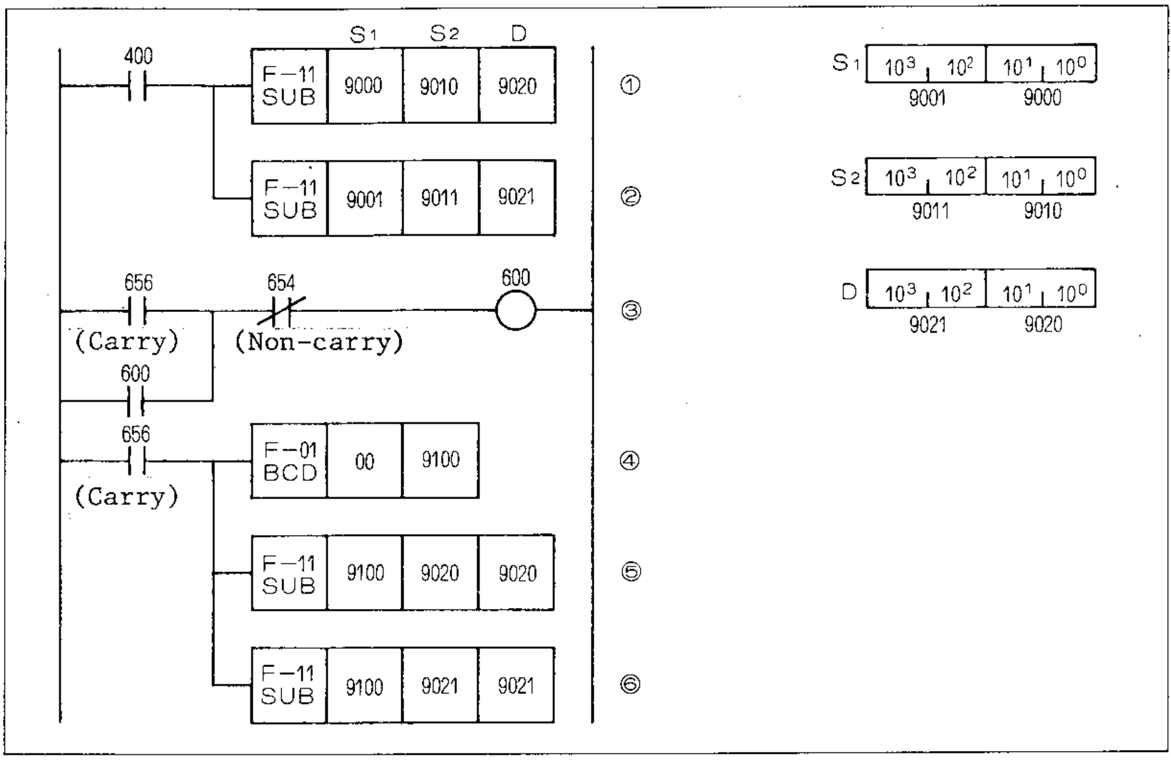
NOTE-1: The TMR current value is stored in the following format in b.××× anb b.×××+1.

b ×××	7	6	5	4	3	2	1	0
	(×10⁰)				(×10⁻¹)			
	"8"	"4"	"2"	"1"	"8"	"4"	"2"	"1"
b ×××+1	7	6	5	4	3	2	1	0
				(×10²)	(10¹)			
	OFF	OFF	*	1	8	4	2	1

With 1 in the bit of b.×××+1 marked with an asterisk (*), it makes the timer operated.
The octal constant 040 of the figure below is ORed by Fc14 and 1 is inserted in the position of an asterisk.

0	0	1	0	0	0	0	0

[10] Obtaining the result by the signed absolute value



° When subtraction of <S_1> < <S_2> is done using F-11, the result will be obtained in complement.

(Ex): 1234 - 2612 → 8622 (complement of 10000).

° It should be programmed in the above manner to obtain the result in the signed absolute value. In this case, the result is $|S_1 - S_2| = D$ and the negative sign (-) is outputted to 600.

①, ② 1 2 3 4 − 2 6 1 2 → 8 6 2 2 Carry flag 1

Complement of 10000

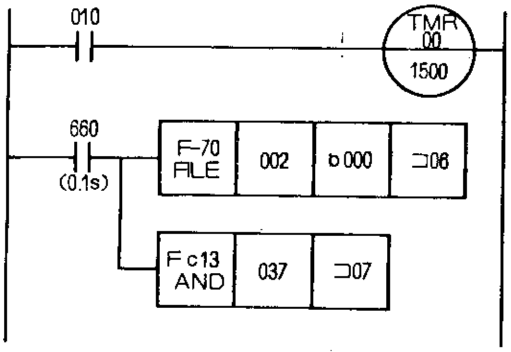
③ The carry flag is retained for the negative sign (−).

④ BCD constant 00 is transferred to the register 9100.

⑤, ⑥ 0 0 0 0 − 8 6 2 2 → 1 3 7 8

Absolute value

[11] Timer current value external output



2-byte transfer
b.000→⊐. 06
b.001→⊐. 07

High order 3 bits of
⊐.007 are masked.

° The current value of TMRQ0 stored in two bytes area of b.000 and b.001 are transferred to two bytes area of ⊐. 06 and ⊐. 07.

° Because TMR is counted down by the 0.1-second clock, 660 (0.1-second clock) is used for the input condition of F-70 and Fc13.

° The current values of TMR and CNT are stored in data memory area b.000 ~ b.137.

See the table below for relation of TMR, CNT, and MD number vs b.xxx.

TMR, CNT MD No.	000	001	57
b.XXX	b.000, b.001	b.002, b.003	b136 b137

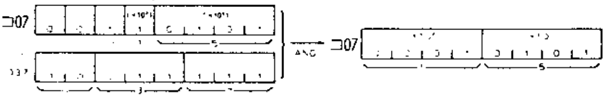
° In the case of TMR, the current values are stored in b.xxx, b.(xxx+1) in the following format.
(Number is treated in the BCD mode.)

	7	6	5	4	3	2	1	0
b.XXX	(×10⁰)				(×10⁻¹)			
	"8"	"4"	"2"	"1"	"8"	"4"	"2"	"1"
b(XXX+1)			✱1	×10²	(×10¹)			
	OFF	OFF	ON	"1"	"8"	"4"	"2"	"1"

*1: When reset is forced in the preset value change mode, it turns OFF.

° Because high order 3 bits of b.001 are other than the current value, they have to be masked when outputted externally. After transferring b.000 to ⊐. 06 using F-70 and b.001 to ⊐. 07, high order 3 bits of ⊐. 07 are masked by Fc13.

Octal constant



° The current value of TMR00 stored in two bytes area of b.000 and b.001 are transferred to two bytes area of ⊐. 06 and ⊐. 07.

° Because TMR is counted down by the 0.1-second clock, 660 (0.1-second clock) is used for the input condition of F-70 and Fc13.

° The current values of TMR and CNT are stored in data memory area b.000 ~ b.137.

See the table below for relation of TMR, CNT, and MD number vs b.xxx.

TMR, CNT MD No.	000	001	57
b. XXX	b. 000, b. 001	b. 002, b. 003	b136 b137

° In the case of TMR, the current values are stored in b.xxx, b.(xxx+1) in the following format.

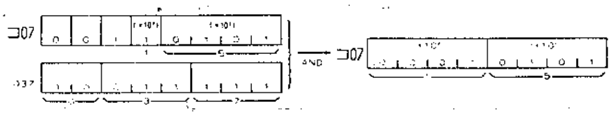
(Number is treated in the BCD mode.)

	7	6	5	4	3	2	1	0
b. XXX	(×10⁰)				(×10⁻¹)			
	"8"	"4"	"2"	"1"	"8"	"4"	"2"	"1"
b(XXX+1)			※1	×10²	(×10¹)			
	OFF	OFF	ON	"1"	"8"	"4"	"2"	"1"

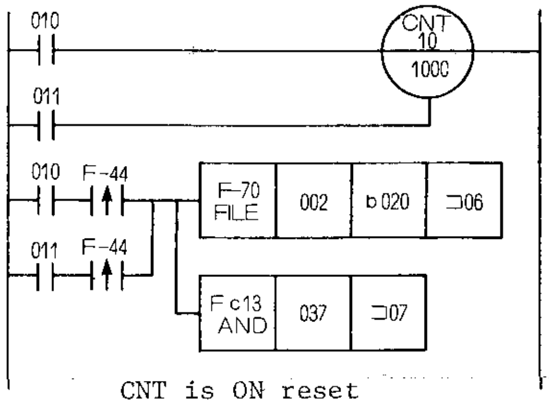
*1: When reset is forced in the preset value change mode, it turns OFF.

° Because high order 3 bits of b.001 are other than the current value, they have to be masked when outputted externally. After transferring b.000 to ⊐. 06 using F-70 and b.001 to ⊐. 07, high order 3 bits of ⊐. 07 are masked by Fc13.

Octal constant



[12] Counter current value external output



2-byte transfer
b.020→⊐. 06
b.021→⊐. 07

High order 3 bits of ⊐. 07 are masked.

```
STR          010
STR          011
CNT          010
            1000
STR          010
F-44
STR          011
F-44
OR STR
F-70
             002
           b.020
           ⊐. 06
Fc13
             037
           ⊐. 07
```

° At a rising edge of the counter input (CNT counts down) or at a rising edge of the reset input (CNT is reset), the contents of b.020 and b.021 in which the CNT010 current value is stored are transferred to ⊐. 06 and ⊐. 07.

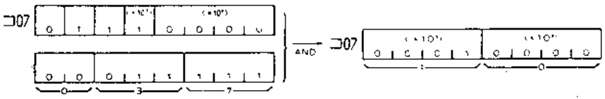
° In the case of CNT, the current value is stored in b.xxx, b.(xxx+1) in the following format.
(Number is treated in the BCD mode.)

	7	6	5	4	3	2	1	0
b. XXX	(×10¹)				(×10⁰)			
	"8"	"4"	"2"	"1"	"8"	"4"	"2"	"1"
b. (XXX+1)			※1	(×10³)	(×10²)			
	OFF	ON	ON	"1"	"8"	"4"	"2"	"1"

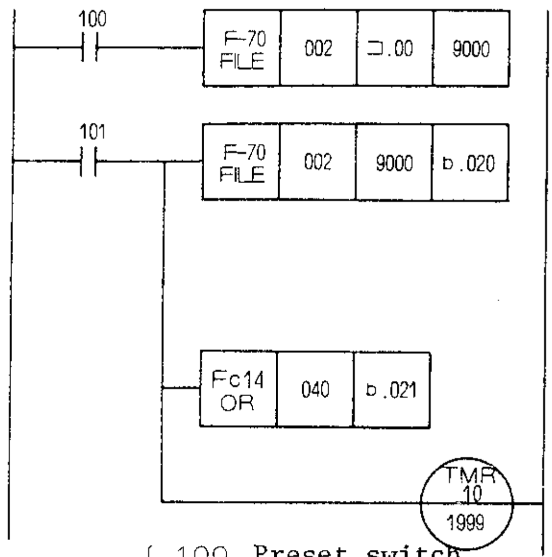
*1: When reset is forced in the preset value change mode, it turns OFF.

° Because high order 3 bits of b.021 are other than the current value, they have to be masked when outputted externally. After transferring b.020 to ⊐. 06 using F-70 and b.021 to ⊐. 07, it is then masked by Fc13.

Octal constant



[13] Input of timer preset value from an external device

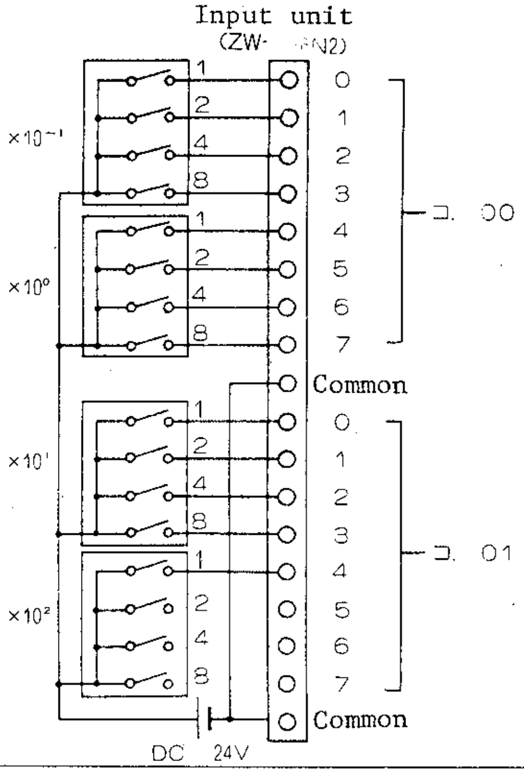


100 Preset switch
101 Start input

By reading the input when the preset switch changes from OFF to ON, it prevent malfunction which may occur when the timer starts while the digital switch is in manipulation. ⊐. 00→9000, ⊐. 01→9001.
9000→b.020
9001→b.021

The forced reset bit of b.021 is turned to 1 (cancellation of reset).

- ° The TMR preset value may be changed while the is running, using an external device such as the digital switch. In other words, the current value is changed from the external source.
- ° Connect the external switch such as the digital switch to ⊐. 00 and ⊐. 01.



NOTE-1:

The TMR preset value must be within a range of 0 to 199.9 seconds. Do not connect the digital switch position 10^2 to "2", "4", and "8" bit to the input unit.

NOTE-2:

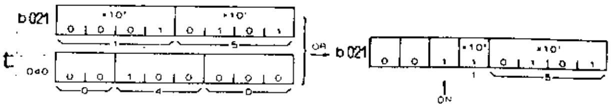
Use the real code digital switch.

- ° When the preset switch changes from OFF to ON, the contents of ⊐. 00 are transferred to 9000 and the contents of ⊐. 01 to 9001.

- The following takes place when the start input changes from OFF to ON.
  1. The contents of 9000 are transferred to b.020 and the contents of 9001 to b.021.
  2. High order third bit of b.021 is set ON. (Fc14)

If this bit is OFF, it invalidates the external setup as the preset value in terms of program in TMR 10 (1999 in this example) is written again to b.020 and b.021.

Octal constant

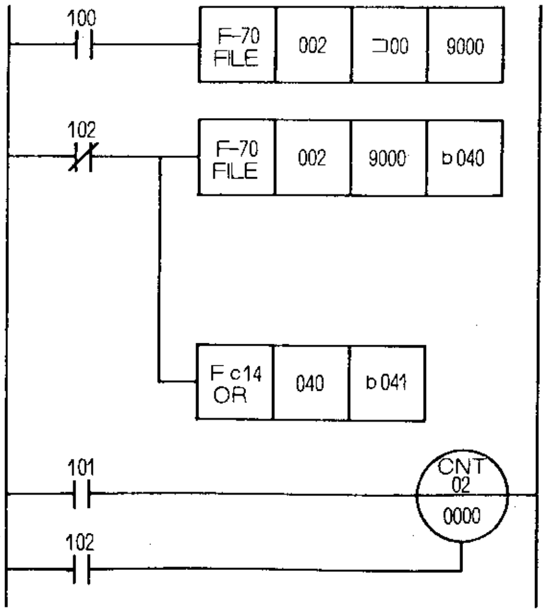


- While the start input is ON, the current value transferred from the external device is subtracted at each 0.1 second. When the current value reaches 0, it makes the TMR contact set ON.
- When the start input turns OFF, the preset value in terms of program (1999 in this example) is set for the TMR current value and the TMR contact is turned OFF.
  When the start input turns ON again, the preset value in terms of program (1999 in this example) is disregarded as the contents of the data memory 9000 and 9001 are transferred as the current value. (Make any value of 0 ∿ 1999 programmed.)

NOTE-1: If forced to reset by the programmer when the start input is ON, it makes the current value set to 0 and the TMR contact set ON. With the forced reset, the preset value in terms of program (1999 in this example) becomes the current value, not the preset value on the digital switch.

NOTE-2: Even if the digital switch setup was changed while the start input is ON (timer in operation), it does not change the current value. The new preset value becomes effective is from a next time that the start input turned ON after it has gone OFF once.

[14] Input of counter preset value from an external device



By reading the input when the preset switch changes from OFF to ON, it prevent malfunction that may occur when reset is cancelled during change in the digital switch position.

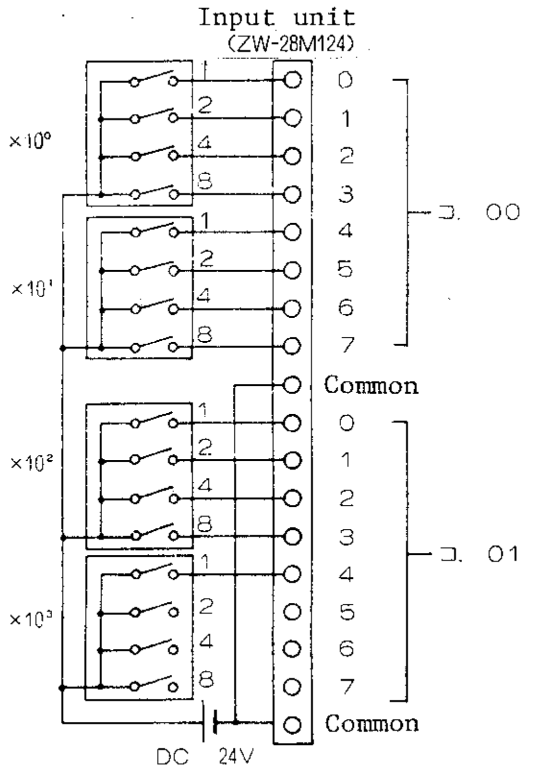
9000→b.040
9001→b.041

The forced reset bit of b.041 is turned to 1 (cancellation of reset).

0100 Preset switch
0101 Counter input
0300 Reset input

° The CNT preset value may be changed while the 'W10' is running, using an external device such as the digital switch. In other words, the current value is changed from the external source.

° Connect the external switch such as the digital switch to ⊐. 00 and ⊐. 01.



NOTE-1:

The CNT preset value must be within a range of 0 to 1999. Do not connect the digital switch position 10^3 to "2", "4", and "8" bit to the input unit.

NOTE-2:

Use the real code digital switch.

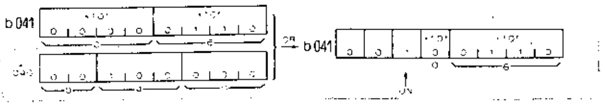
° The following takes place when the reset input changes from ON to OFF.

① The contents of 9000 are transferred to b.040 and the contents of 9001 to b.041.

② High order third bit of b.041 is set ON. (Fc14)

If this bit is OFF, it invalidates the external setup as the preset value in terms of program in CNT020 (0000 in this example) is written again to b.040 and b.041.

Octal constant



° While the reset input is OFF, the current value transferred from the external device is subtracted each time the counter input changes from OFF to ON. When the current value reaches 0, it makes the CNT contact set ON.

° When the reset input turns ON, the preset value in terms of program (0000 in this example) is set for the CNT current value and the CNT contact is turned OFF.

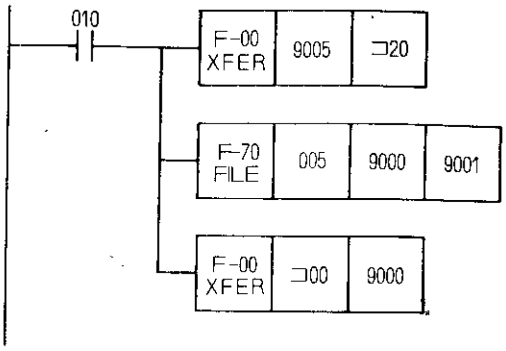
When the reset input turns ON again, the preset value in terms of program (0000 in this example) is disregarded as the contents of the data memory コ. 00 and コ. 01 are transferred as the current value. (Make any value of 0 ~ 1999 programmed.)

NOTE-1: If forced to reset by the programmer when the reset input is OFF, it makes the current value set to 0 and the CNT contact set ON. With the forced reset, the preset value in terms of program (0000 in this example) becomes the current value, not the preset value on the digital switch.

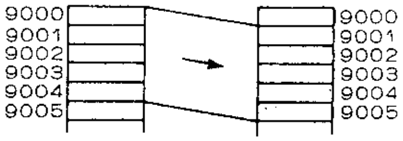
NOTE-2: Even if the digital switch setup was changed while the reset input is ON (counter in operation), it does not change the current value. The new preset value becomes effective is from a next time that the reset input turned ON after it has gone OFF once.

[15] Synchronous type FIFO stack register

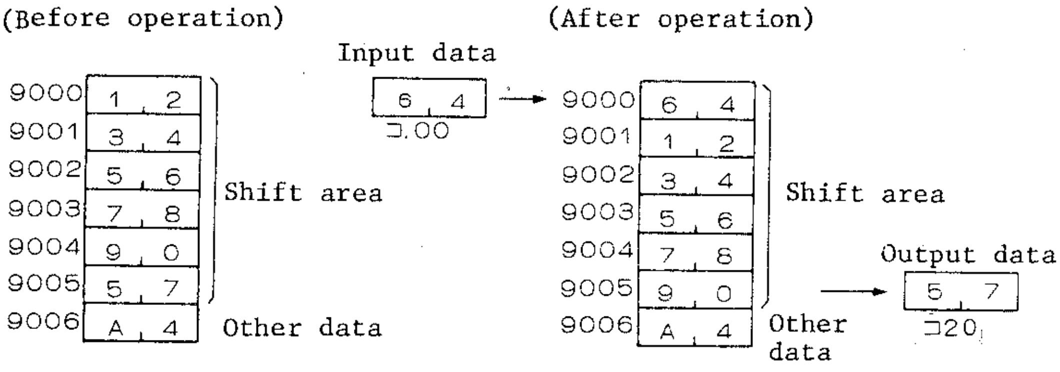
° Constitutes shift register of any desired number of bytes (128 bytes, maximum).



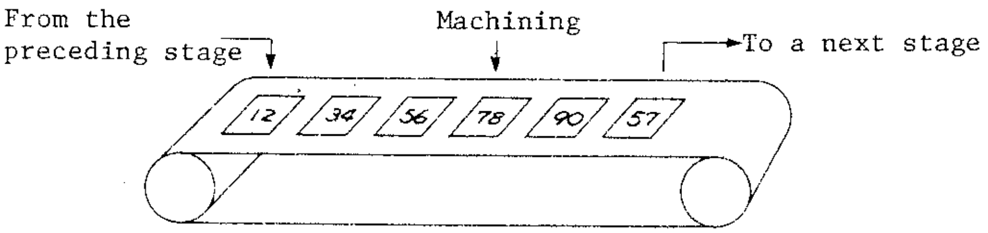
° Outputs the data of the final byte.



° Reads new data.



° 9000 ∿ 9005 have the newest data at all times.

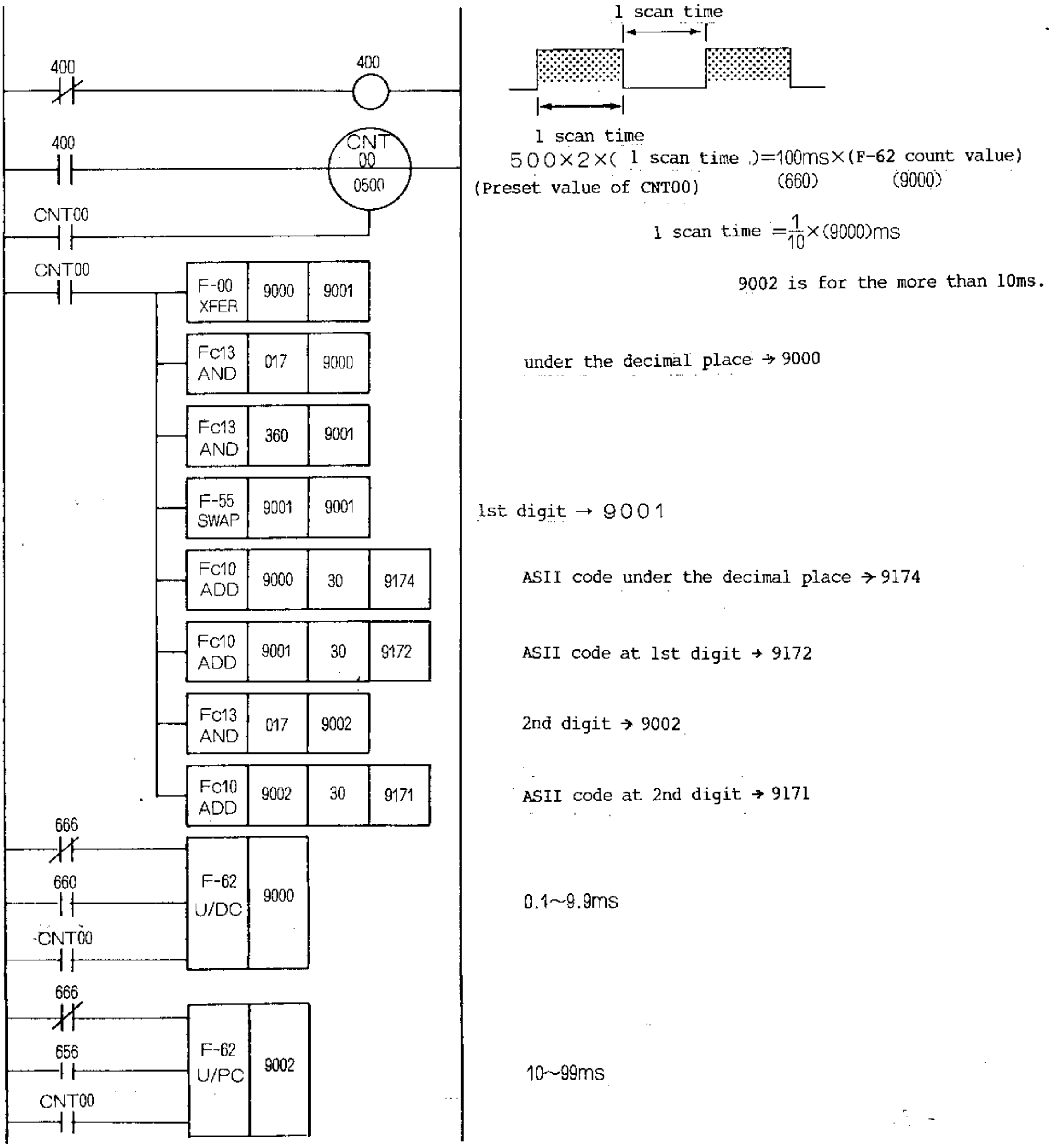


As it performs machining after receiving the model code from the preceding stage, the information is then conveyed to a next stage.

(16) Scan time display

Using data output facility of programmer (ZW-10PG1) under the device mode, the programmer displays scan time as follows.

Scan time = 5.2ms



500×2×(1 scan time)=100ms×(F-62 count value)
(Preset value of CNT00) (660) (9000)

1 scan time $=\frac{1}{10}\times(9000)$ms

9002 is for the more than 10ms.

under the decimal place → 9000

1st digit → 9001

ASII code under the decimal place → 9174

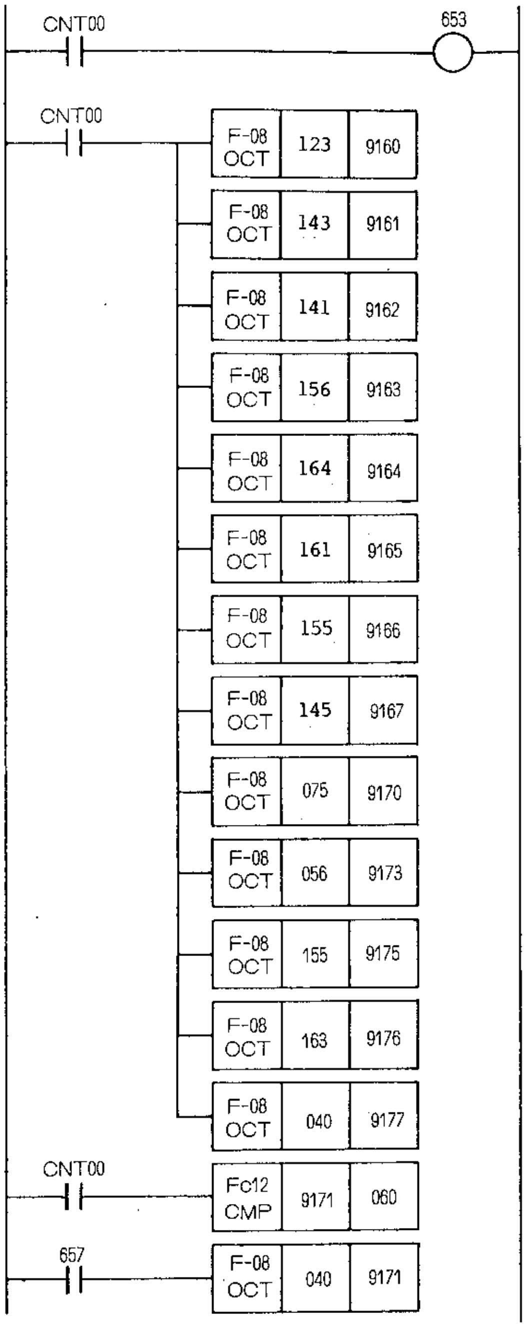
ASII code at 1st digit → 9172

2nd digit → 9002

ASII code at 2nd digit → 9171

0.1~9.9ms

10~99ms



Display device switch

S

c

a

n

t

i

m

e

=

:

m

s

(space)

2nd digit display space
(In case of the value under 9.9ms)

§12 Maintenance and Inspection

12-1. Routine maintenance

[1] Check items

In the following table are listed routine check items that required to be checked for operating W10 under an optimum condition.

1) General items

Item	Contents	Criterion	Note
Ambient temperature	Check if it is within a range specified in the specification. (In the case of in—panel instrallation, the temperature inside the panel becomes the ambient temperature.)	0～+50℃	Free of moisure condensation
Ambient humidity		35～90%RH	
Ambient air		Free of Corrosive gas	
Vibration		Not permitted	
Impact		Not permitted	

2) Main unit

Item	Contents	Criterion
Input power source	Make voltage measure on the terminal board to check if the input voltage is within a range of the given rating.	85～132VAC
Main unit FAULT lamp	Visually observe the FAULT lamp.	Must be off.
Battery	Check if the battery date is not expired.	Must be within the effactive period.
Installed conditions	Check if the main unit is firmly secured.	Free of looseness
	Check for any loosened screw.	Free of looseness
	Check if connector of expansion or link is firmly engaged.	Must be in firm engagement.

3) Expansion unit

Item	Contents	Criterion
Installed condition	Check if expansion unit is firmly engaged.	Free of looseness
	Check for loosened any screw.	Free of looseness.
	Check if connector of expansion cable is firmly engaged to expansion connector.	Must be firmly engaged.

4) I/O module of main unit or expansion unit

Item	Contents	Criterion	Note
Input power source or output power source	Check if the supply voltage to every I/O module of main or expansion unit is within the range specified in the spacification.	100VAC input module 132VAC	ZW−28M114 ZW−28M111 ZW−28N1S4 ZW−28N1S1
		DC input module 24VDC	ZW−28M124 ZW−28M122 ZW−28N2S4 ZW−14N2S4 ZW−28N2S2
		100VAC output module 15～121VAC	ZW−28M111 ZW−28N1S1
		DC output module 10～30VDC	ZW−28M122 ZW−28N2S2
		Relay contact output AC : 250VAC,max. DC : 30VDC, max.	ZW−28M124 ZW−28M114 ZW−28N2S4 ZW−14N2S4 ZW−28N1S4

5) Others

Verify the program on the cassette tape, PROM, or floppy disk (FD) with the program currently in operation.

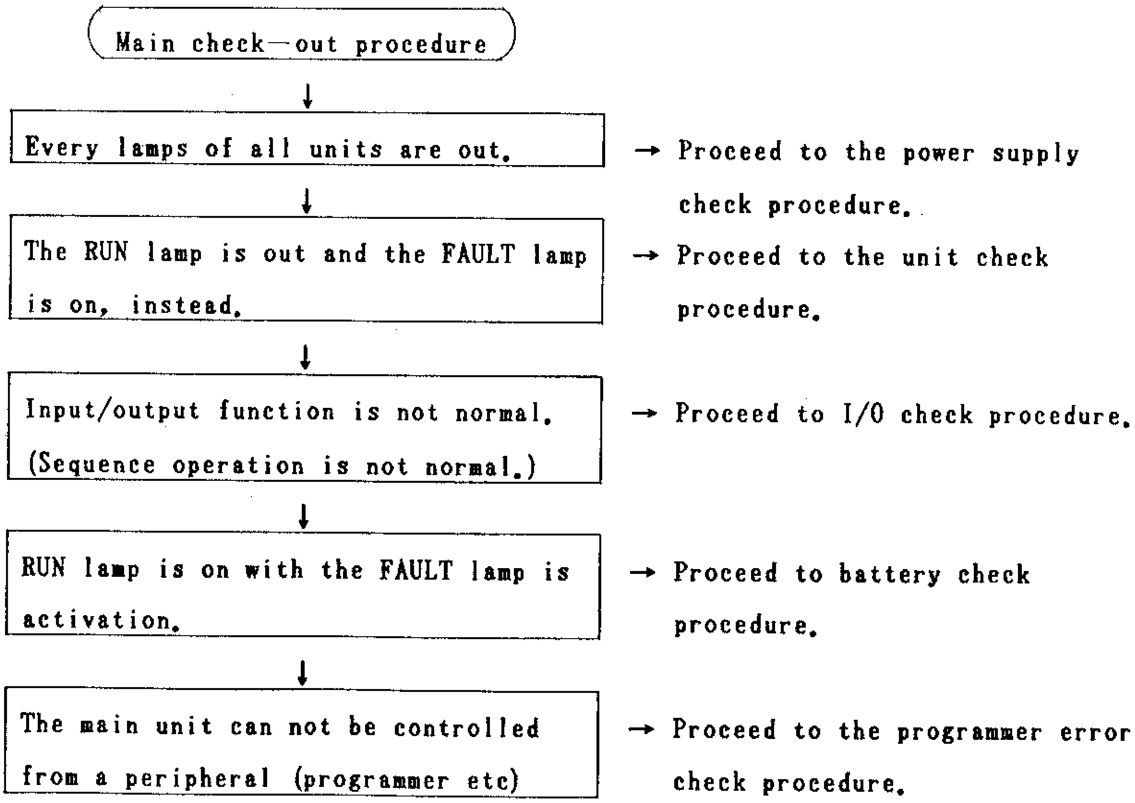
12-2. Error check

Fault condition		Estimated cause	Procedure
"POWER" LED is ON		Wrong connection of cable between power supply board and CPU board	Connects firmly.
"FAULT" LED is ON.	Special relay 670 is ON.	Parity error, exists undefined instruction or exceeds the number of differential memory.	Program correction or replace ROM
	Special relay 671 is ON.	Circuit fault Element fault	Replace main unit
	Special relay 672 is ON,	(1) Battery fault	Replace battery
		(2) Circuit fault, element fault.	Replace main unit
	Special relay 673 is ON.	(1) Connection error of expansion cable or loop back connector.	Connects firmly or exchange cable.
		(2) A change of connection	Restart
		(3) Circuit error, element error	Replace unit
	Special relay 674 is ON.	(1) Wrong connection of cable between main unit and optional unit	Connects firmly
		(2) Circuit error, element error	Replace unit
Output is not ON (OFF).		(1) Program error (double out or etc)	Program correction
		(2) Output specification does not match load one.	Reexamine load specification and then add dummy resistor.
		(3) Power supply to load is not available. Low voltage	Supply power source. Step—up voltage.
		(4) Circuit error, element error	Replace unit

(Continued)

Input is not ON (OFF).	(1) Wrong connection of external wiring	Wire correctly.
	(2) Low voltage of external input power supply	Step—up voltage.
	(3) Circuit error, element error	Replace unit.
Support tool does not work.	(1) Wrong connection between main—unit and support tool	Connects firmly.
	(2) Device mode (in case of ZW—10PG1 operation)	Reset device mode.
	(3) Circuit error, element error.	Replace programmer or main unit.

Shown next are basic error check procedure.



Main check—out procedure

↓

Every lamps of all units are out. → Proceed to the power supply check procedure.

↓

The RUN lamp is out and the FAULT lamp is on, instead. → Proceed to the unit check procedure.

↓

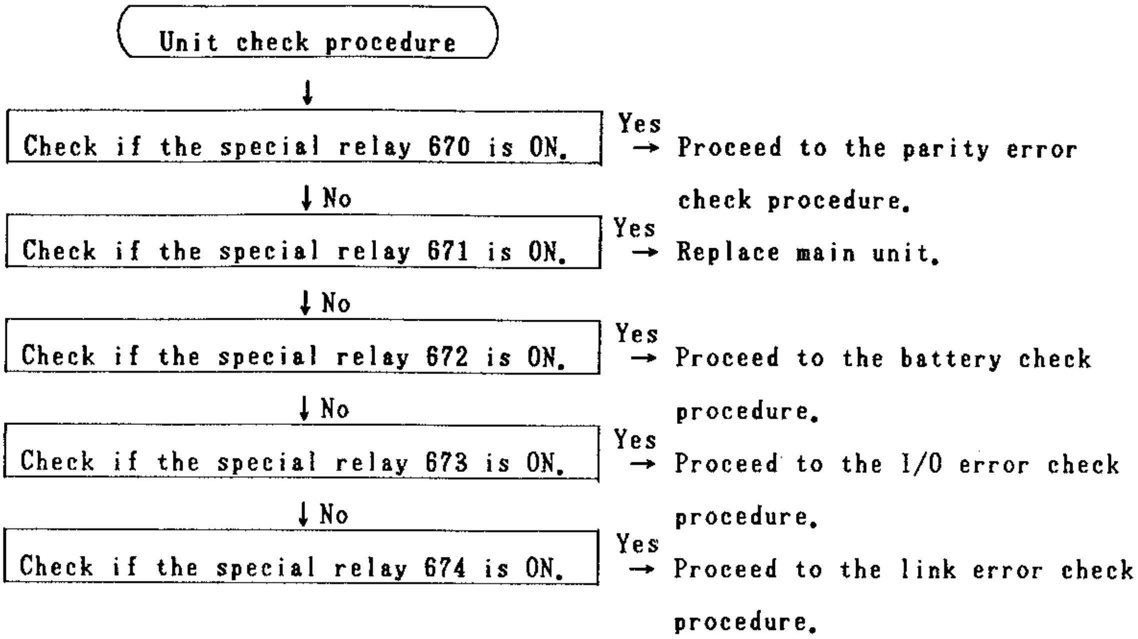
Input/output function is not normal. (Sequence operation is not normal.) → Proceed to I/O check procedure.

↓

RUN lamp is on with the FAULT lamp is activation. → Proceed to battery check procedure.

↓

The main unit can not be controlled from a peripheral (programmer etc) → Proceed to the programmer error check procedure.



Unit check procedure

Check if the special relay 670 is ON. Yes → Proceed to the parity error check procedure.

↓ No

Check if the special relay 671 is ON. Yes → Replace main unit.

↓ No

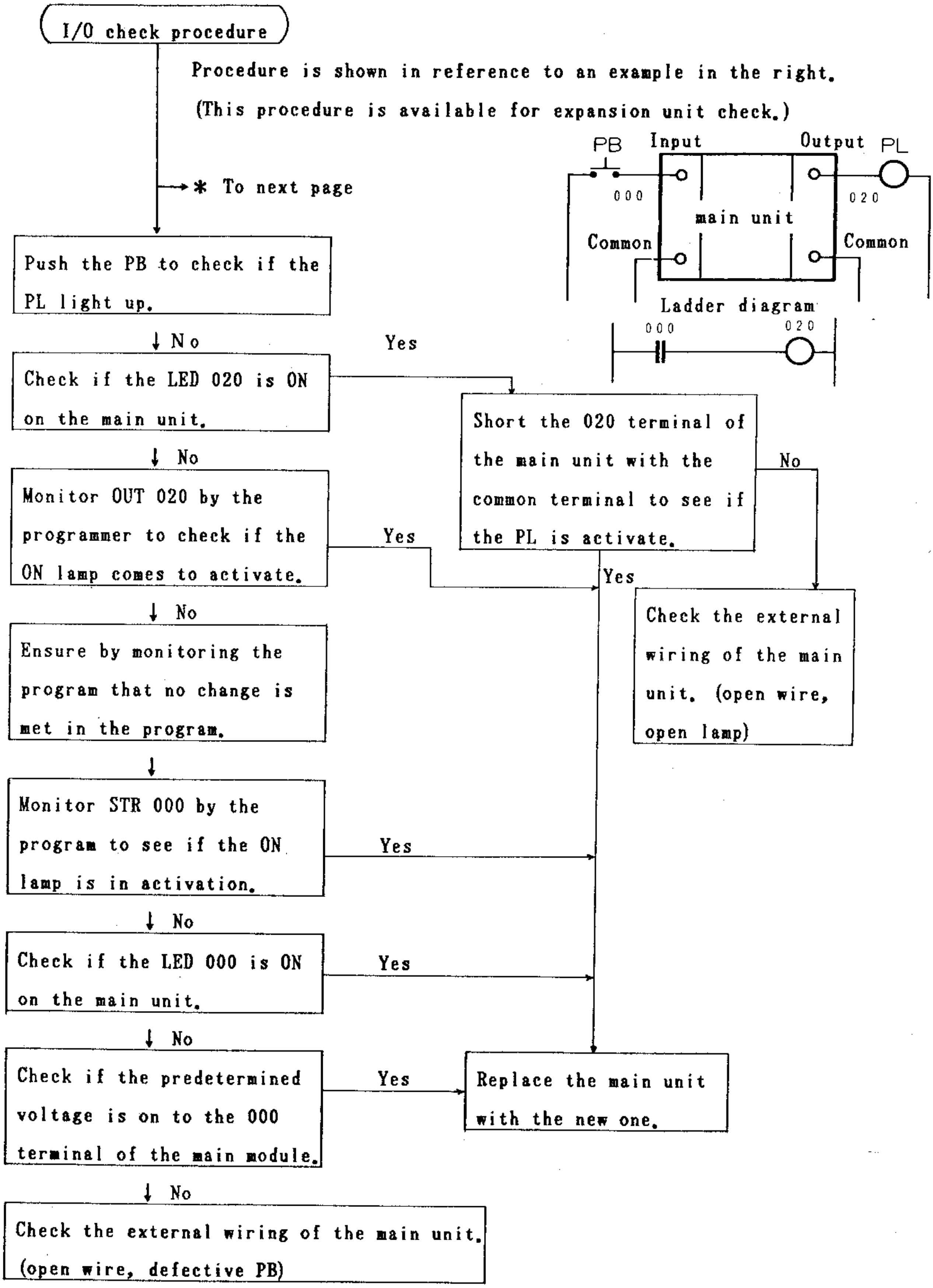
Check if the special relay 672 is ON. Yes → Proceed to the battery check procedure.

↓ No

Check if the special relay 673 is ON. Yes → Proceed to the I/O error check procedure.

↓ No

Check if the special relay 674 is ON. Yes → Proceed to the link error check procedure.



I/O check procedure

Procedure is shown in reference to an example in the right.

(This procedure is available for expansion unit check.)

＊ To next page

Push the PB to check if the PL light up.

Check if the LED 020 is ON on the main unit.

Monitor OUT 020 by the programmer to check if the ON lamp comes to activate.

Ensure by monitoring the program that no change is met in the program.

Monitor STR 000 by the program to see if the ON lamp is in activation.

Check if the LED 000 is ON on the main unit.

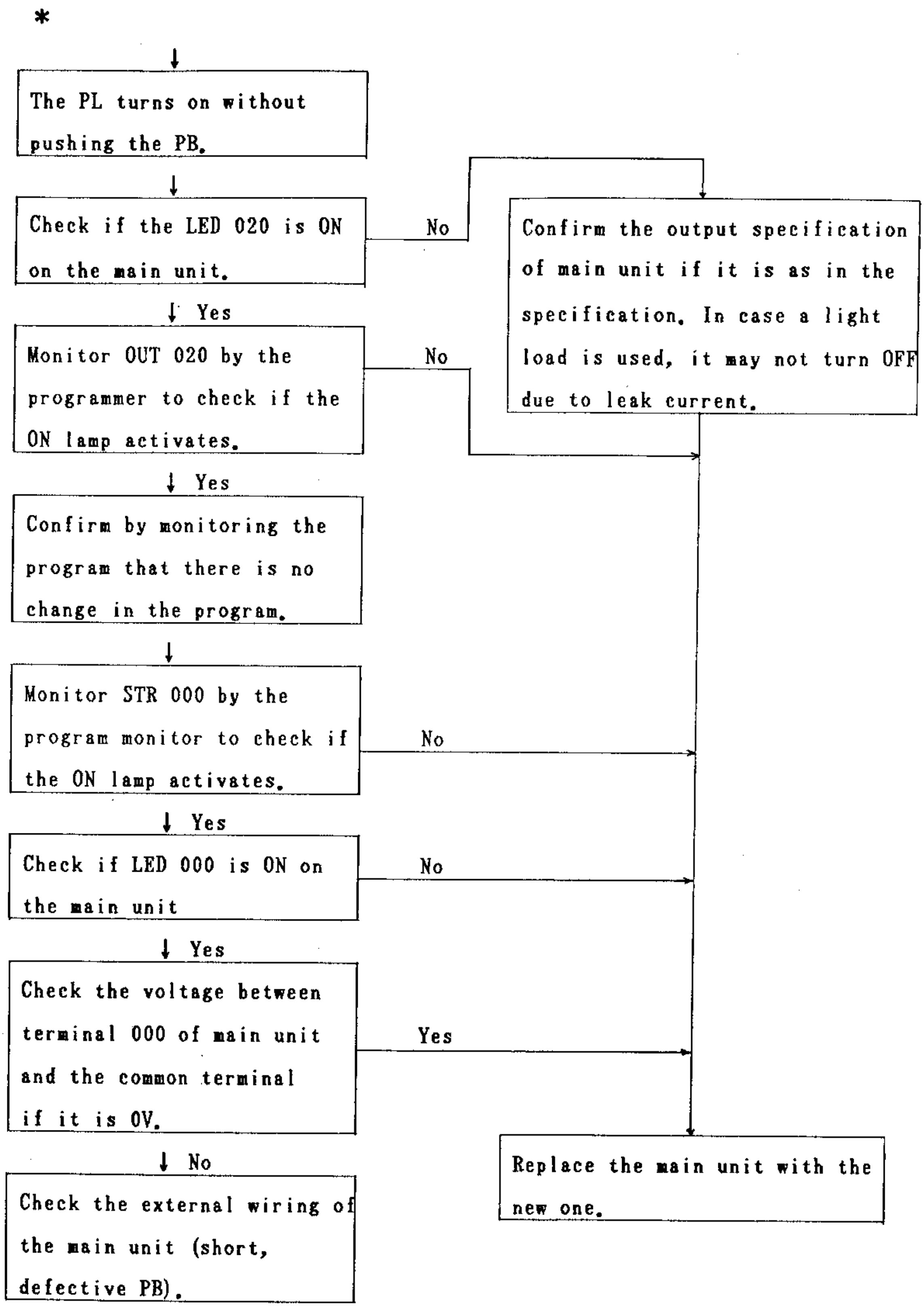
Check if the predetermined voltage is on to the 000 terminal of the main module.

Check the external wiring of the main unit. (open wire, defective PB)

Short the 020 terminal of the main unit with the common terminal to see if the PL is activate.

Check the external wiring of the main unit. (open wire, open lamp)

Replace the main unit with the new one.



The PL turns on without pushing the PB.

Check if the LED 020 is ON on the main unit.

Confirm the output specification of main unit if it is as in the specification. In case a light load is used, it may not turn OFF due to leak current.

Monitor OUT 020 by the programmer to check if the ON lamp activates.

Confirm by monitoring the program that there is no change in the program.

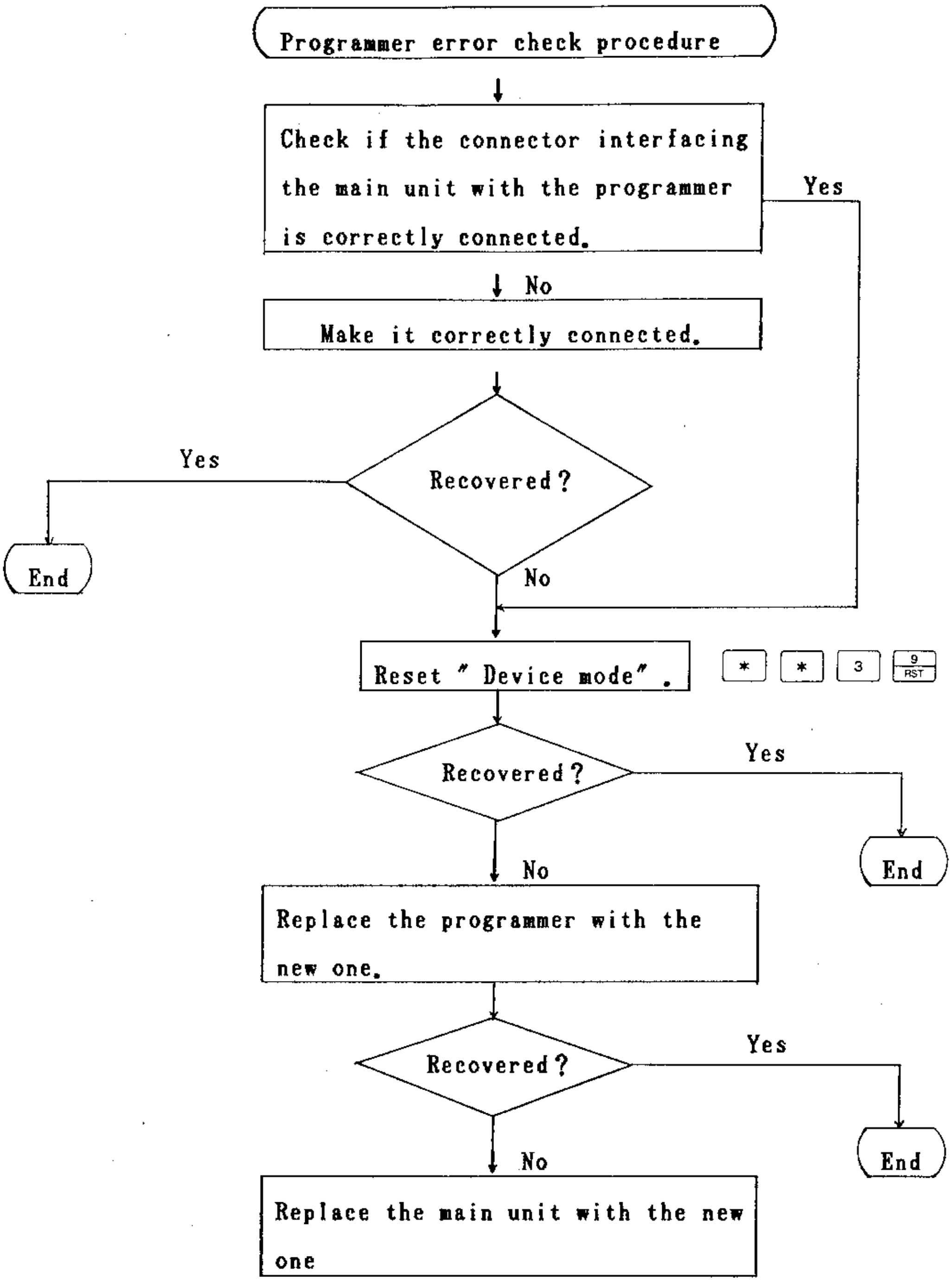
Monitor STR 000 by the program monitor to check if the ON lamp activates.

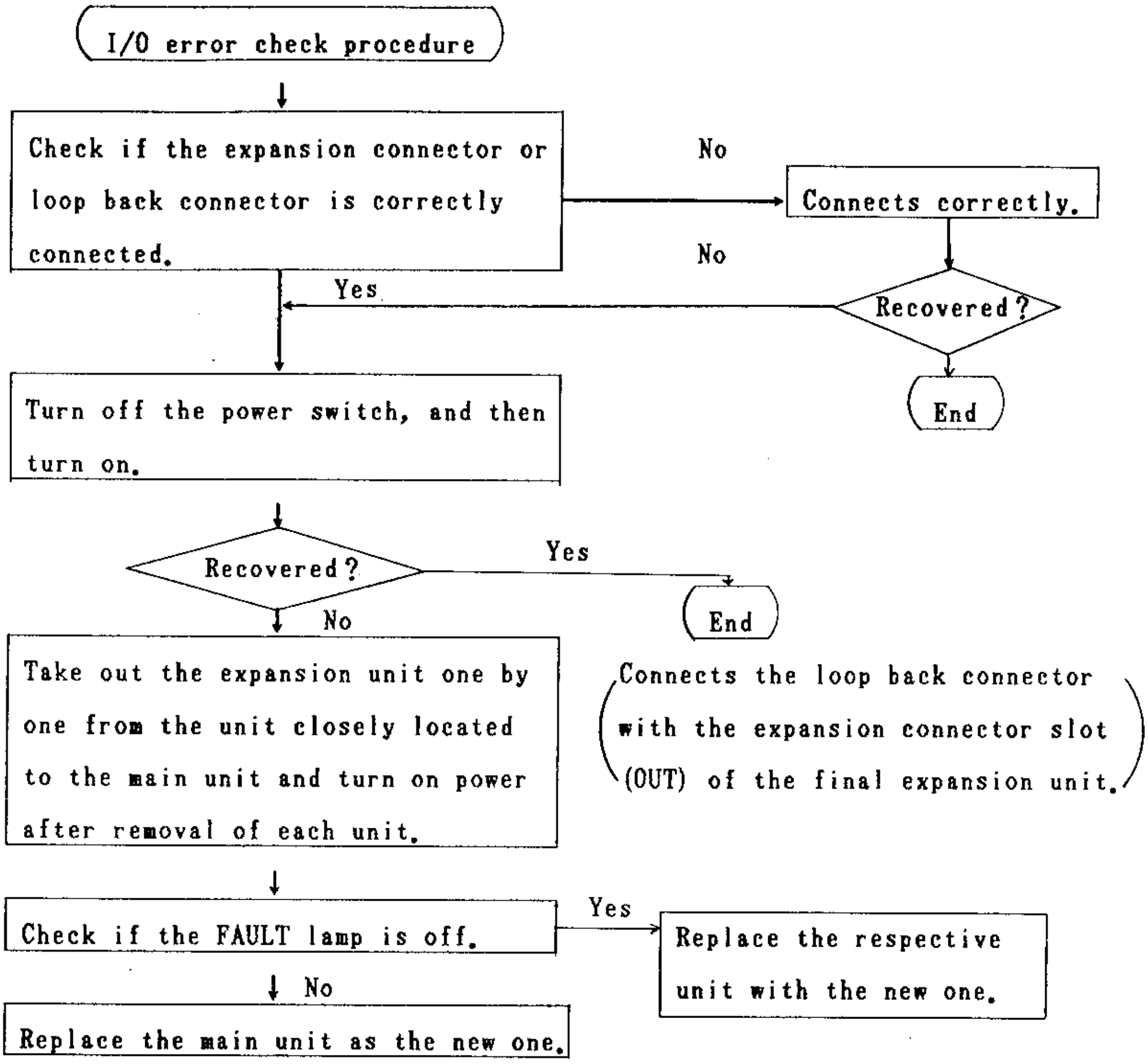
Check if LED 000 is ON on the main unit

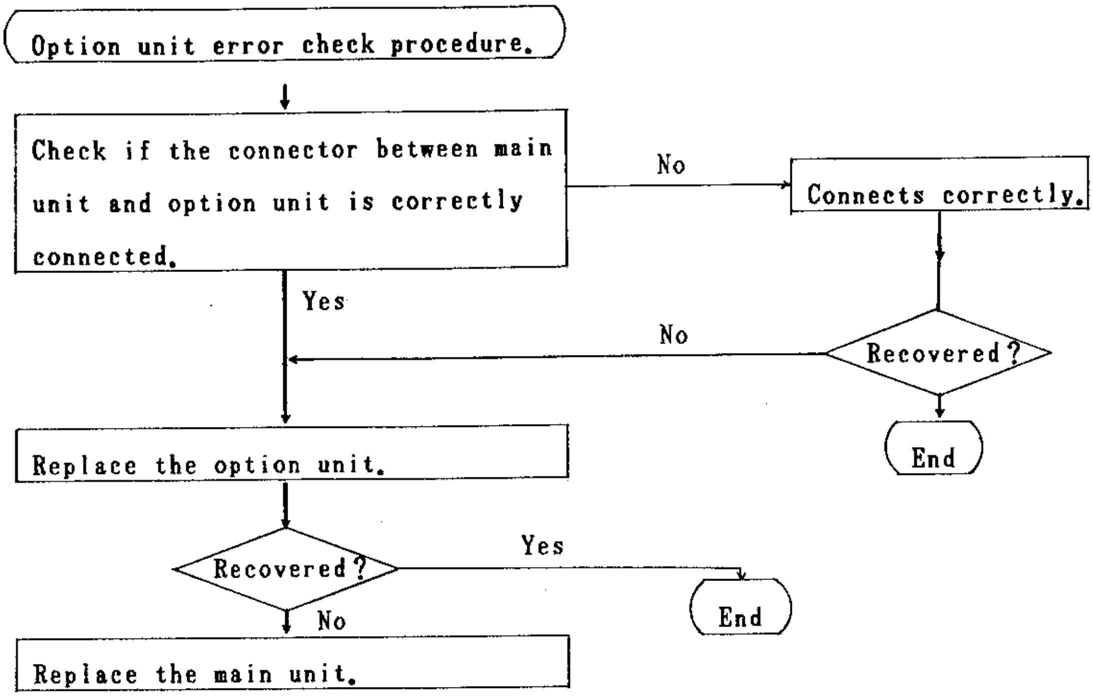
Check the voltage between terminal 000 of main unit and the common terminal if it is 0V.

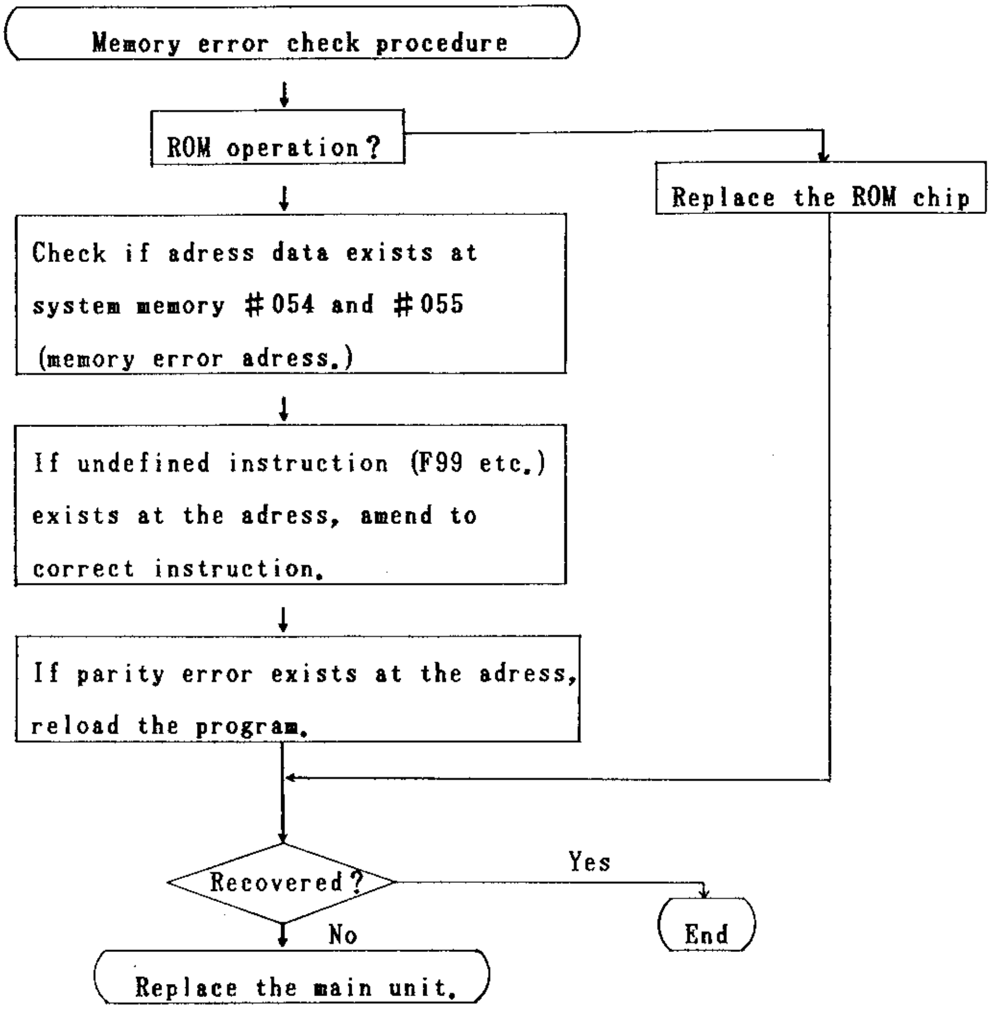
Check the external wiring of the main unit (short, defective PB).

Replace the main unit with the new one.


Programmer error check procedure
Check if the connector interfacing the main unit with the programmer is correctly connected.
Yes
No
Make it correctly connected.
Recovered ?
Yes
End
No
Reset " Device mode" .
*
*
3
9
RST
Recovered ?
Yes
End
No
Replace the programmer with the new one.
Recovered ?
Yes
End
No
Replace the main unit with the new one


I/O error check procedure
Check if the expansion connector or loop back connector is correctly connected.
No
Connects correctly.
No
Yes
Recovered?
End
Turn off the power switch, and then turn on.
Recovered?
Yes
End
No
Take out the expansion unit one by one from the unit closely located to the main unit and turn on power after removal of each unit.
(Connects the loop back connector with the expansion connector slot (OUT) of the final expansion unit.)
Check if the FAULT lamp is off.
Yes
Replace the respective unit with the new one.
No
Replace the main unit as the new one.


Option unit error check procedure.
Check if the connector between main unit and option unit is correctly connected.
No
Connects correctly.
Yes
No
Recovered?
End
Replace the option unit.
Recovered?
Yes
End
No
Replace the main unit.

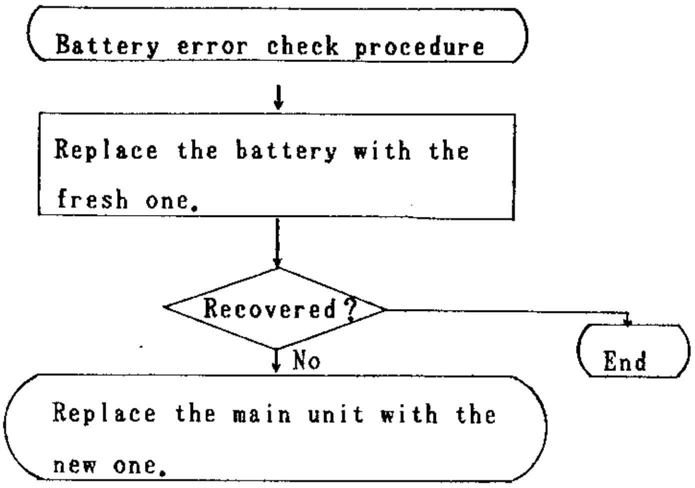

Memory error check procedure
ROM operation?
Replace the ROM chip
Check if adress data exists at system memory #054 and #055 (memory error adress.)
If undefined instruction (F99 etc.) exists at the adress, amend to correct instruction.
If parity error exists at the adress, reload the program.
Recovered?
Yes
End
No
Replace the main unit.

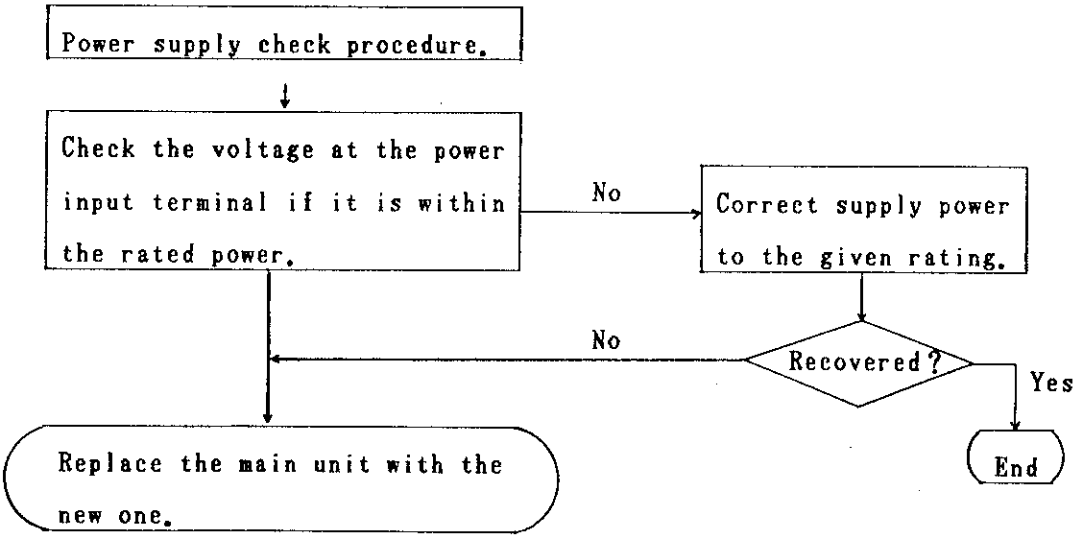
The system memory stored error adress. (unidefined instruction)

```
      #055             |        #054
7  6  5  4  3  2  1  0 | 7  6  5  4  3  2  1  0
            4 digit   3 digit   2 digit   1 digit    Adress

(Example)   Memory error exists at adress 1234.
0  0  0  0  0  0  1  0 | 1  0  0  1  1  1  0  0
               1       | 2       3        4          Adress

   0       0       2   |  2      3        4          Data of system memory
                                                     (octal display)
```


Battery error check procedure
Replace the battery with the fresh one.
Recovered?
No
End
Replace the main unit with the new one.


Power supply check procedure.
Check the voltage at the power input terminal if it is within the rated power.
No
Correct supply power to the given rating.
No
Recovered?
Yes
End
Replace the main unit with the new one.