PROGRAMLANABİLİR DENETLEYİCİ
TEMEL KOMUTLAR İÇİN PROGRAMLAMA ÖRNEKLERİ

1. Input Circuit

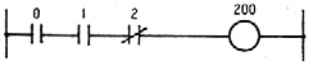
Instruction Word	No./Data
LOD	0
OUT	200

A LOD instruction is used to designate an input branched from the bus.

2. AND Circuit 1 (Series Circuit)

Instruction Word	No./Data
LOD·NOT	0
AND	1
OUT	200

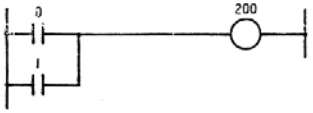
3. AND Circuit 2 (Series Circuit)



Instruction Word	No./Data
LOD	0
AND	1
AND·NOT	2
OUT	200

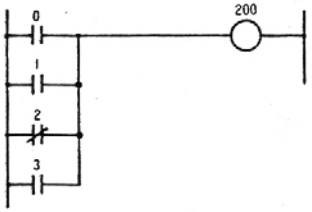
AND instructions can be used continuously without limitation.

4. OR Circuit 1 (Parallel Circuit)



Instruction Word	No./Data
LOD	0
OR	1
OUT	200

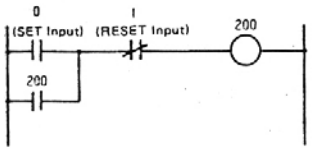
5. OR Circuit 2 (Parallel Circuit)



Instruction Word	No./Data
LOD	0
OR	1
OR·NOT	2
OR	3
OUT	200

OR instructions can be used continuously without limitation.

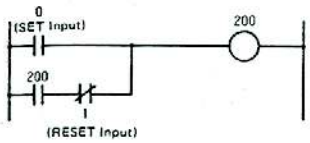
6. Self-Holding Circuit 1 (RESET Preferred)



Instruction Word	No./Data
LOD	0
OR	200
AND·NOT	1
OUT	200

Turning on SET input 0 while RESET input 1 is off turns on Output 200, which remains on after SET input 0 goes off.

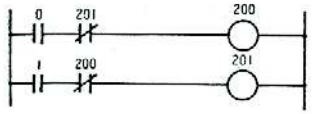
7. Self-Holding Circuit 2 (SET Preferred)



Instruction Word	No./Data
LOD	0
LOD	200
AND·NOT	1
OR·LOD	
OUT	200

Turning on SET input 0 turns on Output 200 whether RESET input is on or off, and Output 200 remains on after SET input 0 goes off.

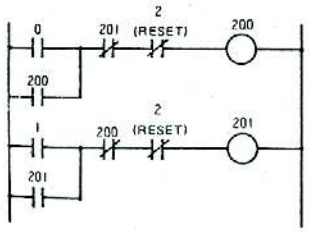
8. Priority Circuit 1 (Continuous Input Signal)



Instruction Word	No./Data
LOD	0
AND·NOT	201
OUT	200
LOD	1
AND·NOT	200
OUT	201

Input 0 or 1 whichever enters first is given priority, nullifying the input which enters next.

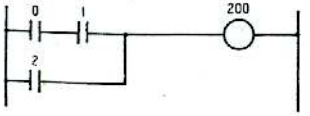
9. Priority Circuit 2 (Pulse Input Signal)



Instruction Word	No./Data
LOD	0
OR	200
AND·NOT	201
AND·NOT	2
OUT	200
LOD	1
OR	201
AND·NOT	200
AND·NOT	2
OUT	201

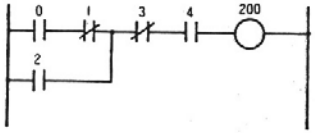
This circuit is used when outputs should be prevented from turning on simultaneously, such as forward/reverse control of a motor.

10. Series-Parallel Circuit 1



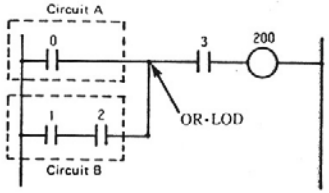
Instruction Word	No./Data
LOD	0
AND	1
OR	2
OUT	200

11. Series-Parallel Circuit 2



Instruction Word	No./Data
LOD	0
AND•NOT	1
OR	2
AND•NOT	3
AND	4
OUT	200

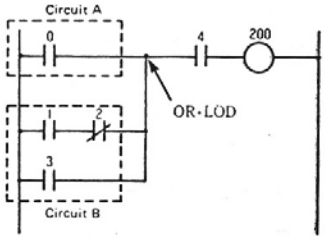
12. Series-Parallel (OR•LOD) Circuit 3



Instruction Word	No./Data	
LOD	0	- A
LOD	1	- B
AND	2	
OR•LOD		
AND	3	
OUT	200	

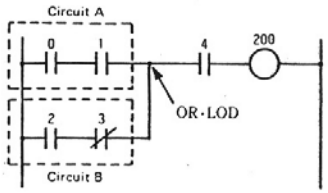
After Circuits A and B are programmed, these circuits are ORed by an OR•LOD instruction. Thereafter, AND3 instruction is programmed.

13. Series-Parallel (OR•LOD) Circuit 4



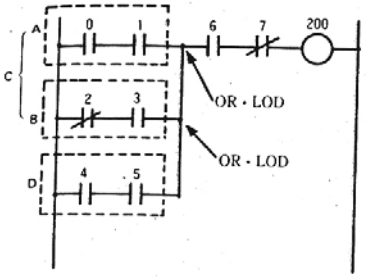
Instruction Word	No./Data	
LOD	0	- A
LOD	1	- B
AND•NOT	2	
OR	3	
OR•LOD		- OR circuit of A and B
AND	4	
OUT	200	

14. Series-Parallel (OR•LOD) Circuit 5



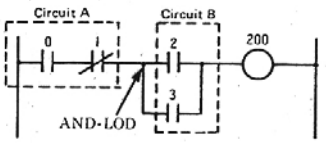
Instruction Word	No./Data	
LOD	0	- A
AND	1	
LOD	2	- B
AND•NOT	3	
OR•LOD		- OR circuit of A and B
AND	4	
OUT	200	

15. Series-Parallel (OR•LOD) Circuit 6



Instruction Word	No./Data		
LOD	0	- A	- C
AND	1		
LOD•NOT	2	- B	
AND	3		
OR•LOD		- OR circuit of A and B	
LOD	4	- D	
AND	5		
OR•LOD		- OR circuit of C and D	
AND	6		
AND•NOT	7		
OUT	200		

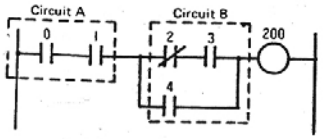
16. Series-Parallel (AND•LOD) Circuit 1



Instruction Word	No./Data	
LOD	0	- A
AND•NOT	1	
LOD	2	- B
OR	3	
AND•LOD		
OUT	200	

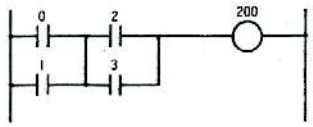
After Circuits A and B are programmed, these circuits are ANDed by an AND•LOD instruction.

17. Series-Parallel (AND•LOD) Circuit 2

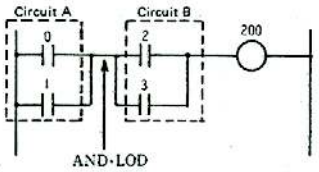


Instruction Word	No./Data		
LOD	0	- A	
AND	1		
LOD•NOT	2	- B	
AND	3		
OR	4		
AND•LOD		- AND circuit of A and B	
OUT	200		

18. Series-Parallel (AND·LOD) Circuit 3

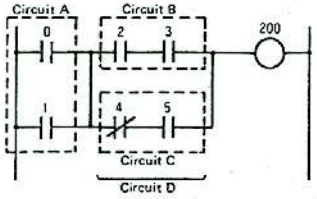


When modified, the circuit is easy to understand.



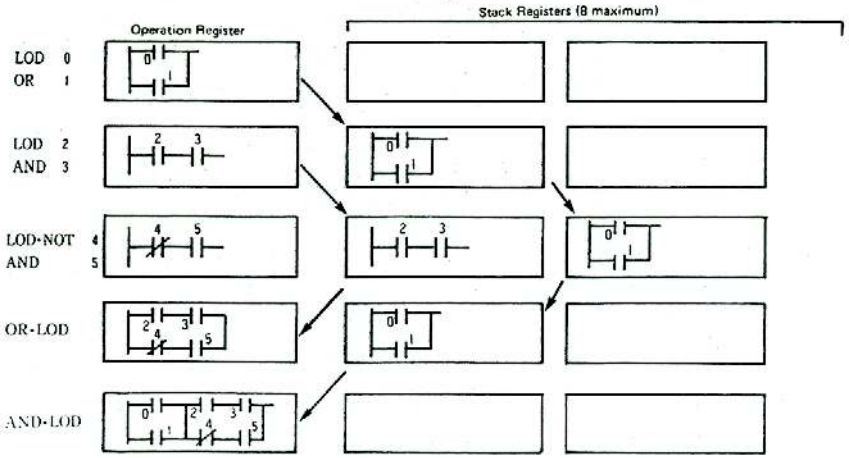
Instruction Word	No./Data	
LOD	0	- A
OR	1	
LOD	2	- B
OR	3	
AND·LOD		- AND circuit of A and B
OUT	200	

19. Series-Parallel (AND·LOD & OR·LOD) Circuit 4

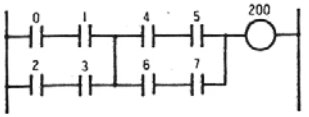


Instruction Word	No./Data	
LOD	0	A
OR	1	
LOD	2	B
AND	3	
LOD·NOT	4	C
AND	5	
OR·LOD		OR circuit of B and C = D
AND·LOD		AND circuit of A and D
OUT	200	

NOTE: Operations of Operation Registers and Stack Registers

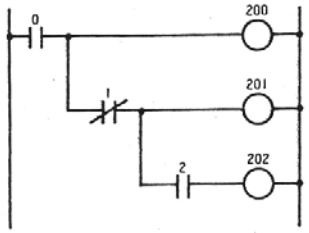


20. Series-Parallel (AND·LOD & OR·LOD) Circuit 5



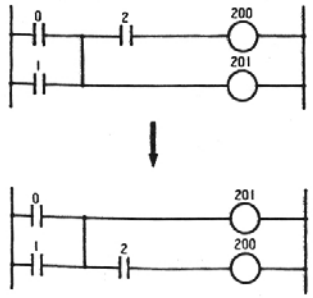
Instruction Word	No./Data
LOD	0
AND	1
LOD	2
AND	3
OR·LOD	
LOD	4
AND	5
LOD	6
AND	7
OR·LOD	
AND·LOD	
OUT	200

21. Multiple Output Circuit 1



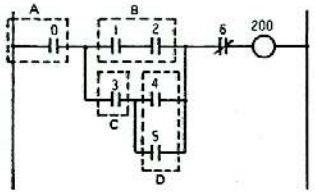
Instruction Word	No./Data
LOD	0
OUT	200
AND·NOT	1
OUT	201
AND	2
OUT	202

22. Multiple Output Circuit 2



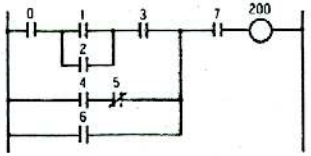
Instruction Word	No./Data
LOD	0
OR	1
OUT	201
AND	2
OUT	200

23. Complex Circuit 1



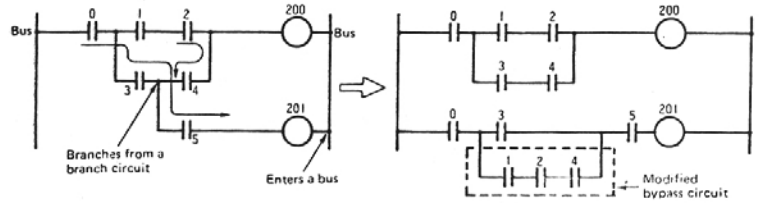
Instruction Word	No./Data	
LOD	0	- A
LOD	1	- B
AND	2	
LOD	3	- C
LOD	4	- D
OR	5	
AND•LOD		- AND circuit of C and D = E
OR•LOD		- OR circuit of B and E = F
AND•LOD		- AND circuit of A and F
AND•NOT	6	- AND-NOT 6 is added at the end.
OUT	200	

24. Complex Circuit 2



Instruction Word	No./Data
LOD	0
LOD	1
OR	2
AND•LOD	
AND	3
LOD	4
AND•NOT	5
OR•LOD	
OR	6
AND	7
OUT	200

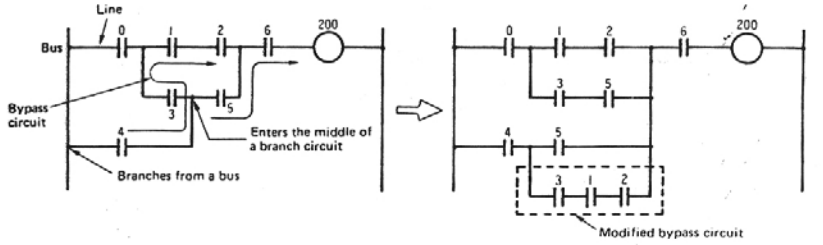
1. Branching from a Branch Circuit to a Bus



●Circuit which cannot be programmed

●Circuit modified for PC

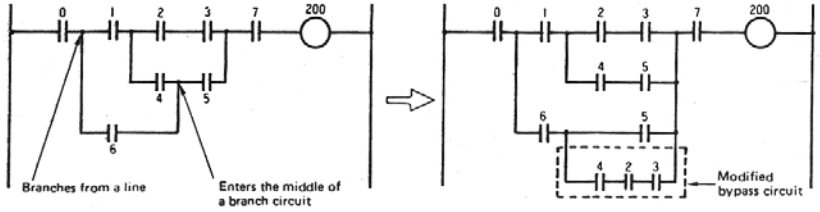
2. Branching from a Bus to the Middle of a Branch Circuit



●Circuit which cannot be programmed

●Circuit modified for PC

3. Branching from a Line to the Middle of a Branch Circuit

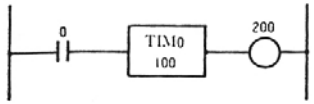


●Circuit which cannot be programmed

●Circuit modified for PC

1. ON-Delay Timer

(a)



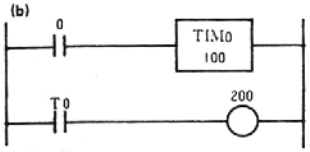
Instruction Word	No./Data
LOD	0
TIM	0
	100
OUT	200

A TIM instruction requires two addresses.

TIM No.

Preset time (100 = 10.0 sec)

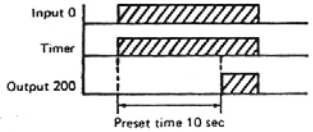
(b)



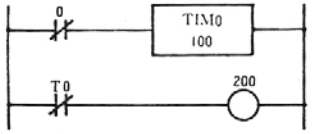
Instruction Word	No./Data
LOD	0
TIM	0
	100
LOD·T	0
OUT	200

TIM instructions can be programmed in two ways; 1. as with Circuit (a), output can be taken out directly from the TIM instruction, and 2. as with Circuit (b), output can be converted to a contact signal and taken out.

●Time Chart

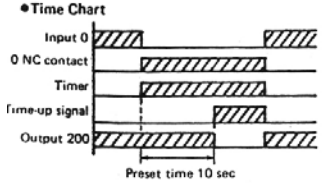


2. OFF-Delay Timer



Instruction Word	No./Data
LOD·NOT	0
TIM	0
	100
LOD·NOT·T	0
OUT	200

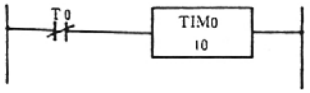
●Time Chart



When Input 0 goes off, the NC contact of Input 0 starts TIM 0. After a preset time, the NC contact of TIM 0 turns off Output 200.
When Input 0 goes on, Output 200 also goes on.

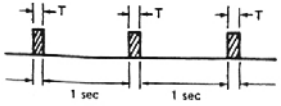
3. Pulse Generating Circuit

(a) Flicker Circuit



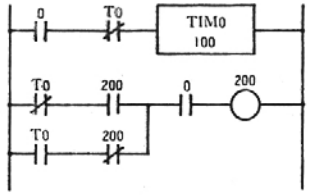
Instruction Word	No./Data
LOD·NOT·T	0
TIM	0
	10

●Time Chart



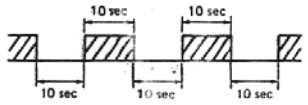
T = 1 scan time

(b) Constant Duty Ratio Pulse Generating Circuit



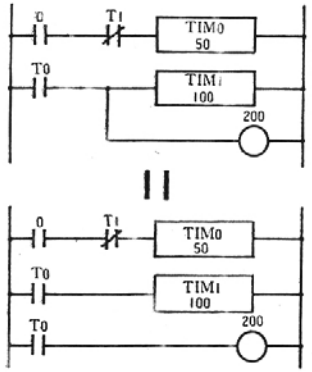
Instruction Word	No./Data
LOD	0
AND·NOT·T	0
TIM	0
	100
LOD·NOT·T	0
AND	200
LOD·T	0
AND·NOT	200
OR·LOD	
AND	0
OUT	200

●Time Chart



While Input 0 is on, output pulses with same ON and OFF durations (duty ratio 1:1) are generated.

(c) Adjustable Duty Ratio Pulse Generating Circuit

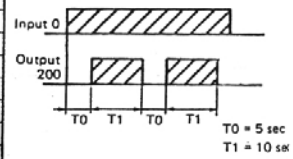


Instruction Word	No./Data
LOD	0
AND·NOT·T	1
TIM	0
	50
LOD·T	0
OUT	200
TIM	1
	100

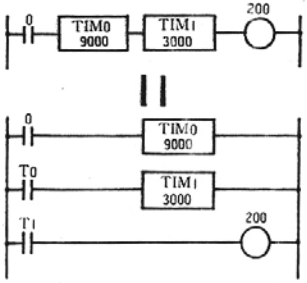
Instruction Word	No./Data
LOD	0
AND·NOT·T	1
TIM	0
	50
LOD·T	0
TIM	1
	100
LOD·T	0
OUT	200

While Input 0 is on, output pulses with OFF duration of TIM 0 and ON duration of TIM 1 are generated.
Output can be programmed in parallel with TIM instructions.

●Time Chart



4. Long-Delay Timer 1 (Timer + Timer)

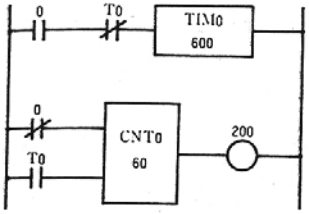


Instruction Word	No./Data	
LOD	0	
TIM	0	
	9000	900 sec
TIM	1	
	3000	300 sec
OUT	200	

TIM instructions can be programmed continuously to provide a long-delay timer.
There is no limit to the number of TIM instructions to be programmed continuously.

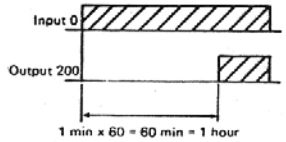
Example: 9000 (900 sec) + 3000 (300 sec) = 12000 (1200 sec) = 20

5. Long-Delay Timer 2 (Timer + Counter)



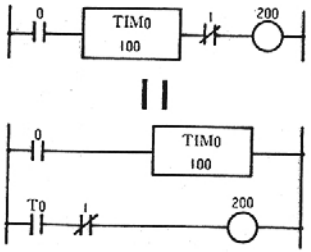
Instruction Word	No./Data	
LOD	0	
AND·NOT·T	0	
TIM	0	
	600	60 sec (1 min)
LOD·NOT	0	
LOD·T	0	
CNT	0	
	60	× 60
OUT	200	

●Time Chart



After Input 0 has turned on, clock pulses generated by TIM 0 are counted to provide a long-delay timer.

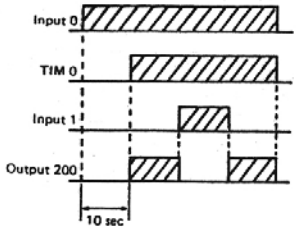
6. Circuit to Turn Output Off Temporarily after Time-up



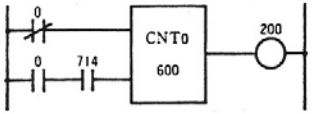
Instruction Word	No./Data
LOD	0
TIM	0
	100
AND·NOT	1
OUT	200

Following to a TIM instruction, a circuit can be programmed.

●Time Chart

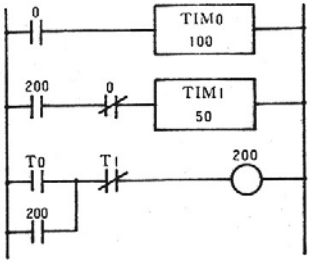


7. Timer Using Special Internal Relay IR 714 (1-sec clock)



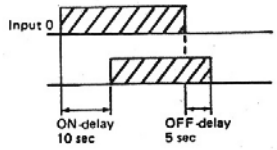
Instruction Word	No./Data	
LOD·NOT	0	
LOD	0	
AND	714	1-sec clock pulse
CNT	0	
	600	600 sec (10 min)
OUT	200	

8. ON/OFF Delay Circuit



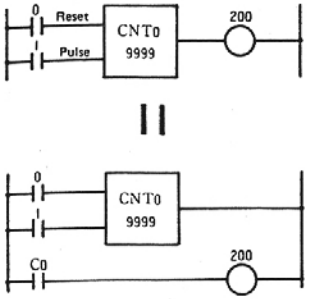
Instruction Word	No./Data	
LOD	0	
TIM	0	
	100	10 sec
LOD	200	
AND·NOT	0	
TIM	1	
	50	5 sec
LOD·T	0	
OR	200	
AND·NOT·T	1	
OUT	200	

●Time Chart



Output 200 turns on 10 sec after Input 0 has turned on, and Output 200 turns off 5 sec after Input 0 has turned off.

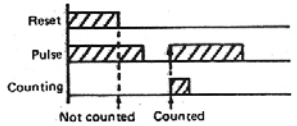
1. Adding Counter 1



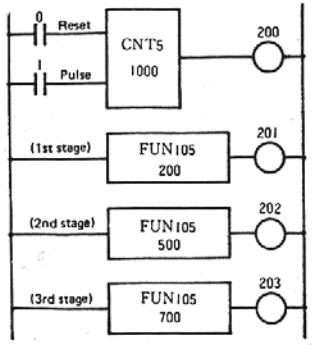
Instruction Word	No./Data
LOD	0
LOD	1
CNT	0
	9999 - Preset value
OUT	200

While Input 0 is off, Input 1 pulse signals are counted. When reaching the preset value, Output 200 is turned on.

●Pulse Input Acceptance Timing

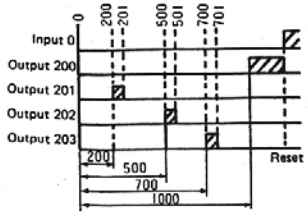


2. Adding Counter 2 (Multi-Stage Setting One-Shot Output Circuit)

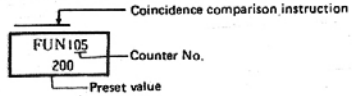


Instruction Word	No./Data
LOD	0
LOD	1
CNT	5
	1000 - Counter preset value
OUT	200
FUN	105 - 1st-stage setting
	200
OUT	201
FUN	105 - 2nd-stage setting
	500
OUT	202
FUN	105 - 3rd-stage setting
	700
OUT	203

●Time Chart

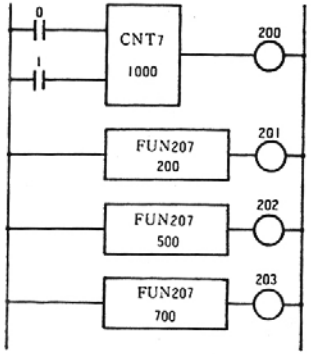


The multi-stage one-shot output circuit for the adding counter is programmed using a FUN□□ instruction (counter coincidence comparison instruction).
There is no limit to the number of stages for a multi-stage counter.



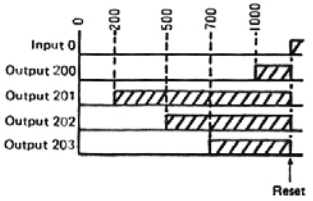
Output signal is on only when the counted value coincides with the preset value.

3. Adding Counter 3 (Multi-Stage Setting Self-Holding Output Circuit)

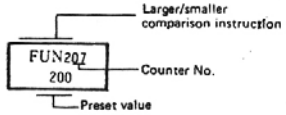


Instruction Word	No./Data
LOD	0
LOD	1
CNT	7
	1000 - Counter preset value
OUT	200
FUN	207 - 1st-stage setting
	200
OUT	201
FUN	207 - 2nd-stage setting
	500
OUT	202
FUN	207 - 3rd-stage setting
	700
OUT	203

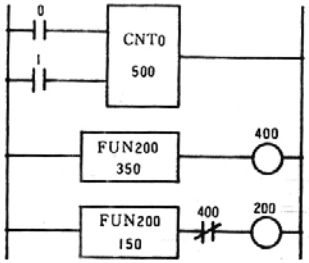
●Time Chart



The multi-stage self-holding output circuit for the adding counter is programmed using a FUN2□□ instruction (counter larger/smaller comparison instruction).
There is no limit to the number of stages for the multi-stage counter.



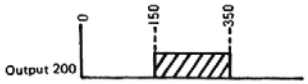
4. Adding Counter 4 (Larger/Smaller Comparison Circuit)



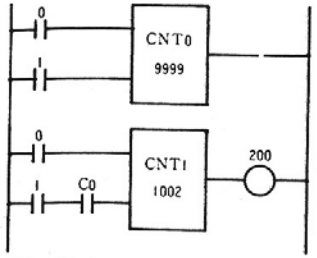
Instruction Word	No./Data
LOD	0
LOD	1
CNT	0
	500
FUN	200
	350
OUT	400
FUN	200
	150
AND·NOT	400
OUT	200

Output 200 is turned on when the counted value of Counter 0 is between 150 and 300.

●Time Chart



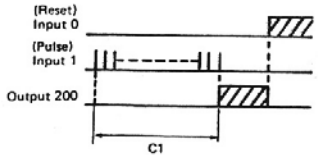
5. Multi-Digit Counter 1



Instruction Word	No./Data
LOD	0
LOD	1
CNT	0
	9999
LOD	0
LOD	1
AND·CNT	0
CNT	1
	1002
OUT	200

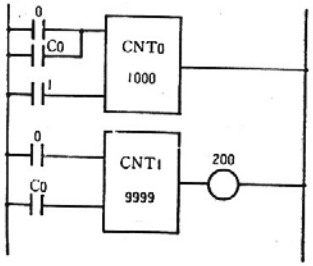
When the preset value exceeds 9999, two counters can be used to count up to 19997.

●Time Chart



$C1 = \underset{\text{value}}{\text{CNT 0 preset}} + \underset{\text{value}}{\text{CNT 1 preset}} - 1 = 9{,}999 + 1{,}002 - 1 = 11{,}000$

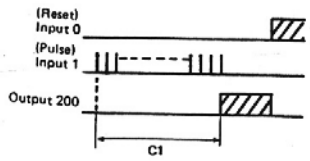
6. Multi-Digit Counter 2 (7-Digit Setting)



Instruction Word	No./Data
LOD	0
OR·C	0
LOD	1
CNT	0
	1000
LOD	0
LOD·C	0
CNT	1
	9999
OUT	200

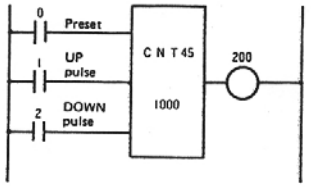
When the preset value exceeds 4 digits, two counters can be used to provide a 7-digit counter.

●Time Chart



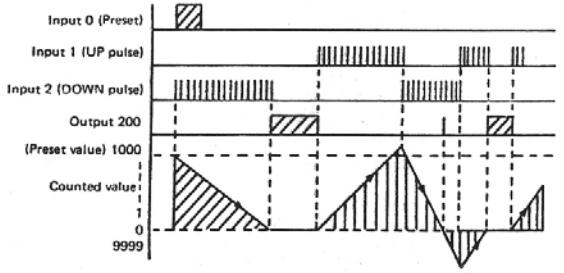
$C1 = \underset{\text{value}}{\text{CNT 0 preset}} \times \underset{\text{value}}{\text{CNT 1 preset}} = 1{,}000 \times 9{,}999 = 9{,}999{,}000$

6. Dual-Pulse Type Reversible Counter Circuit (CNT 45)



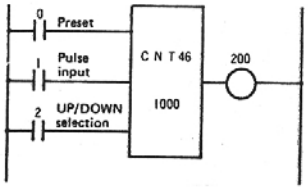
Instruction Word	No./Data
LOD	0
LOD	1
LOD	2
CNT	45
	1000
OUT	200

This reversible counter has two pulse inputs for adding (UP) and subtracting (DOWN).



UP and DOWN pulse inputs are accepted. When Preset input 0 enters, the counted value is reset to the CNT 45 preset value (1000). Output 200 is on only when the counted value is 0.

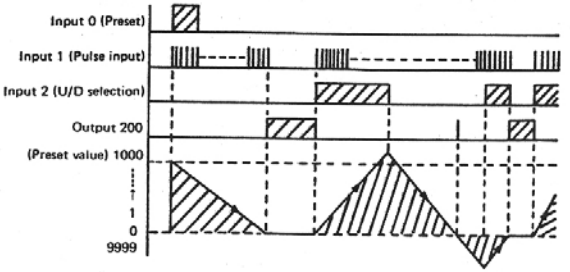
7. UP/DOWN Selection Type Reversible Counter Circuit (CNT 46 & 47)



Instruction Word	No./Data
LOD	0
LOD	1
LOD	2
CNT	46
	1000
OUT	200

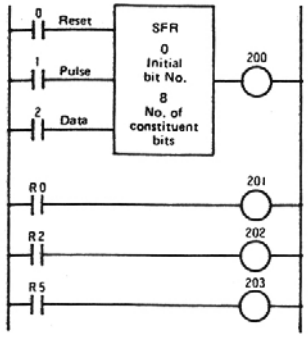
This reversible counter has a pulse input and a selection input for switching the UP/DOWN gate.

UP/DOWN selection input
ON: UP (adding)
OFF: DOWN (subtracting)

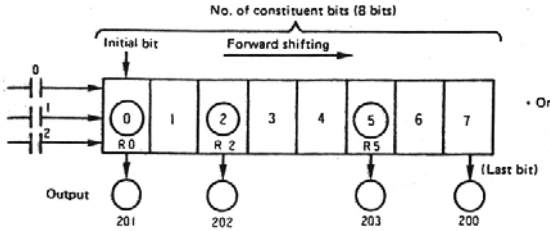


When Preset input 0 enters, the counted value is reset to the CNT 46 preset value (1000). Output 200 is on only when the counted value is 0.

1. Forward Shift Register Circuit

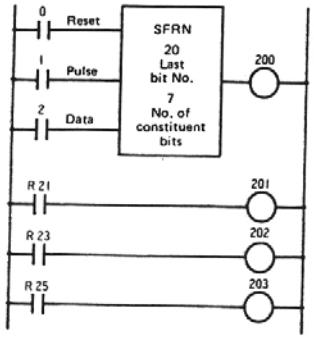


Instruction Word	No./Data	
LOD	0	
LOD	1	
LOD	2	
SFR	0	- Initial bit
	8	- 8-bit configuration
OUT	200	
LOD·R	0	- Deriving Bit 0 status
OUT	201	
LOD·R	2	- Deriving Bit 2 status
OUT	202	
LOD·R	5	- Deriving Bit 5 status
OUT	203	

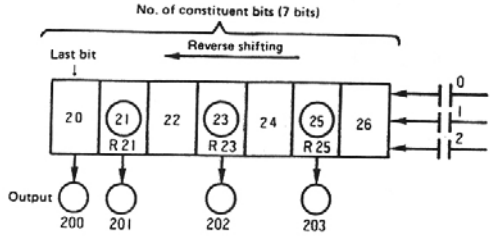


• Only the circled bits are outputted.

2. Reverse Shift Register Circuit

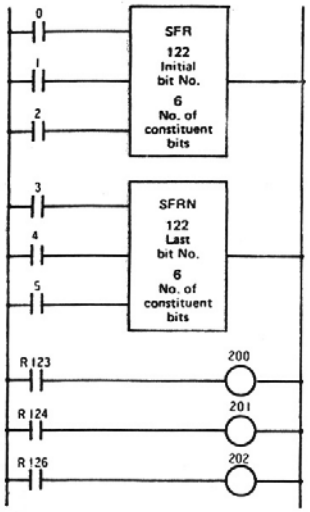


Instruction Word	No./Data	
LOD	0	
LOD	1	
LOD	2	
SFR·NOT	20	- Last bit
	7	- 7-bit configuration
OUT	200	
LOD·R	21	- Deriving Bit 21 status
OUT	201	
LOD·R	23	- Deriving Bit 23 status
OUT	202	
LOD·R	25	- Deriving Bit 25 status
OUT	203	

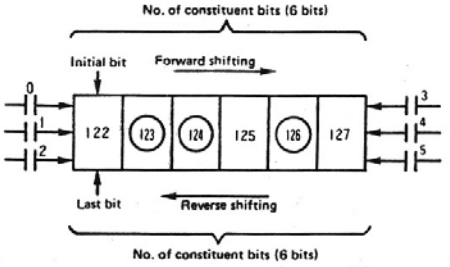


• Only the circled bits are outputted.

3. Bidirectional Shift Register Circuit



Instruction Word	No./Data
LOD	0
LOD	1
LOD	2
SFR	122
	6
LOD	3
LOD	4
LOD	5
SFR·NOT	122
	6
LOD·R	123
OUT	200
LOD·R	124
OUT	201
LOD·R	126
OUT	202



• Only the circled bits are outputted.

PROGRAMLANABİLİR DENETLEYİCİ
KULLANICI GİRİŞ/ÇIKIŞ ÇEVRE BİRİMLERİ İÇİN PROGRAMLAMA ÖRNEKLERİ

External Preset Function

Preset values of timer (TIM) and counter (CNT) instructions programmed in the FA-1J and counter coincidence and greater/smaller comparison instructions can be set via digital switch installed externally.

Specifications

Preset type	BCD multiple setting
No. of circuits	Max. 16 (BCD 4 digits)
No. of I/Os (per circuit)	DC input unit (Source type): 4 Transistor output: 4
Applicable digital switch	Binary digital switch (with diode). Install diodes (for switching) according to the connection diagram. (Ex.) IDEC's type DF( )( )-031D(K) Diode Rating: Reverse voltage 80V min. Average rectifying current 100mA min. Lead diameter 0.5 to 0.6mm

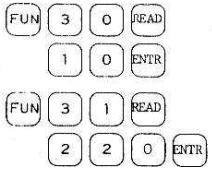
Programming Method

- Use FUN70 to FUN85 as data of instruction words.
- Allocate initial input and output numbers via function setting FUN30 and 31.

- I/O number allocation example

  When initial input number for FUN30 is 10 and initial output number for FUN31 is 220

- Key operation



Note: 4n I/Os are allocated continuously for external presetting starting from the initial I/O No. according to designation by FUN70 to 85. FUN numbers cannot be designated intermittently, such as FUN70 followed by FUN73.

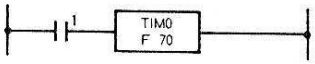
- I/O allocation for the above setting

FUN No. Corresponding to Digital Switch	Input No.	Output No.
FUN70	10-13	220-223
FUN71	14-17	224-227
FUN72	20-23	230-233
⋮	⋮	⋮
FUN85	⋮	⋮
Digital switch	A-D	Lower to Upper digit

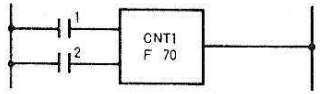
Program Example

(When a digital switch is connected to an I/O No. corresponding to FUN70)

- Timer external preset example (Input 1, TIM0, FUN70)

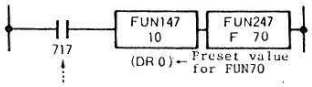


- Counter external preset example (Reset input 1, clock input 2, CNT1, FUN70)



- External preset example of counted value coincidence comparison instruction
  (Counter 30, Output 200, FUN70)

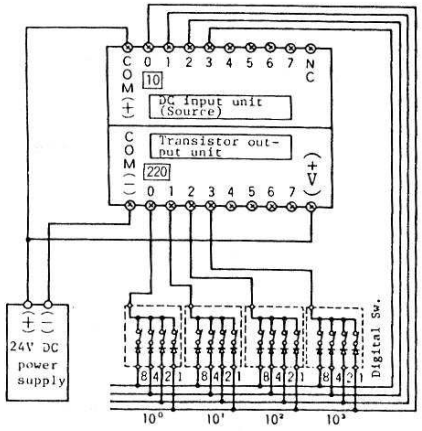
- Setting a constant for computing instruction



Note 1: Pulse input (SOT, etc.) cannot be used for this setting.

Note 2: Do not change preset values during counting. When a timer preset value is changed, the changed value becomes effective from the next timer operation.

Standard Connection Diagram (For the example at left)



Note 3: Initial input and output numbers are allocated via FUN30 and 31.

Note 4: When using the external preset function, one scan time is 15msec at minimum.
(When the external presetting (FUN70 to 85) is used, any program requires a scan time of 15msec at minimum.)

Note 5: When changing a preset value via an external preset device (digital switch), it takes a maximum of 15 scan times to register the changed value correctly.

Note 6: When the external presetting is used as an operand of a computing instruction, be sure to use a continuous ON input for the conditions of the instruction. A value is read correctly 10 to 15 scans after turning ON the input.

Note 7: External presetting cannot be used in an MCS or JMP instruction.

External Display Function

Values of timers (TIM), counters (CNT) and data registers (DR) can be displayed on external digital display devices using a standard transistor output unit.

Specifications

Display system	Dynamic lighting display
No. of circuits	Max. 8 (BCD 4 digits)
No. of outputs	Transistor output 8 points/circuit (Digit selection: 4 points, BCD output: 4 points)
Applicable display unit	7-segment LED digital display: BCD input (negative logic) with latch 24V DC (Ex.) IDEC's type · DD33-F31N-B(Z) · DD96-F(R)31N-B Note: Digit selection (latch) output can be latched to Low or High via FUN35.

Note 1: Be sure to use continuous ON input for input of external display instruction conditions. A correct value is displayed 8 scans after turning ON the input.

Note 2: External display instructions cannot be used more than 8 times.

Note 3: External display instruction cannot be used between MCS and MCR or between JMP and JEND.

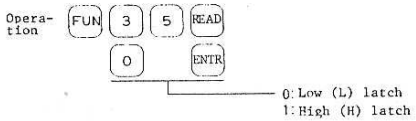
Program Example

External display instruction FUN147 25

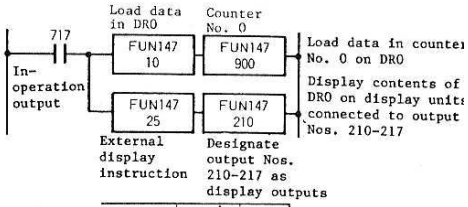
Converts data at data register DR0 into BCD and outputs display data by one digit after each scan. It takes 8 scans to output 4-digit display data.

Ex. To display the counted value of counter No. 0 on 4-digit 7-segment display units connected to output Nos. 210 to 217.

① Set latch conditions of external display instruction via function setting (FUN35).

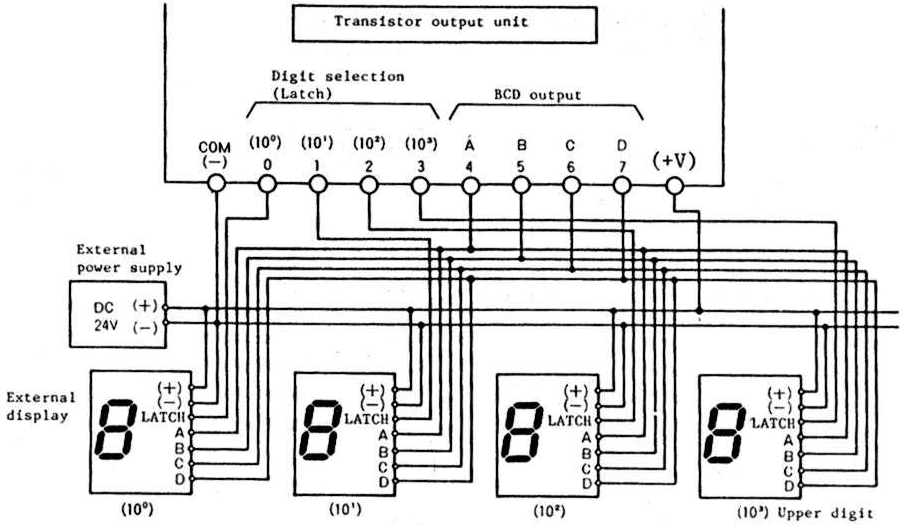


② Program



Address	Inst'n Word	Data
0	LOD	717
1	FUN	147
2		10
3	FUN	147
4		900
5	FUN	147
6		25
7	FUN	147
8		210

Standard Connection Diagram



PROGRAMLANABİLİR DENETLEYİCİ HESAPLAMA KOMUTLARI İÇİN PROGRAMLAMA ÖRNEKLERİ

The FA-1J has the following computing functions.

(1) Addition, (2) Subtraction, (3) Multiplication, (4) Division,
(5) BCD-to-binary convertion, (6) Binary-to-BCD convertion,
(7) Numerical value comparison (4-digit comparison)

Terms

Data (contents) of operand, data register or carry are shown in parentheses.

Example: When data register No. 10 contains data "5555";

Diagram = FUN147 810

Operand = Data register No. 10 = DR10

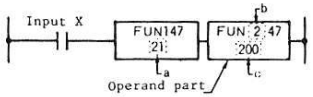
(Operand 810) = Contents of data resister No. 10 = (DR10) = 5555

Operand number = No. 810

Computing Instruction Composition

Two instructions are always used in pairs: Operation is designated by the code of FUN147 and 247 (computing instruction), and the next instruction gives an operand (information).

- For all computing instructions, data registers 0 and 1 (DR0 and DR1) are used.
- A data register is composed of 16 bits (2 bytes).



(1) Input X: 1 (ON) .... Computing is executed.
0 (OFF) ... Computing is not executed.

(2) a ... Designates the type of computing. (Table 2)
(Load, Store, Add, Subtract, Multiply, Divide, Compare, Display)

(3) b = "1" ... Designates I/O, IR, CNT, TIM, or DR.
b = "2" ... Designates a constant or external presetting.

(4) c ... Operand number or constant (Table 1)

Note: The second instruction FUN147 or FUN247 is not needed for a binary-to-BCD or BCD-to-binary conversion.

Operand List (Table 1)

Operand No. (c)	Operand Contents	
0- 60	Input	No. 0- 60
200- 260	Output	No. 200-260
400- 680	Internal relay	No. 400-680
* 800- 899	Data register	No. 0- 99
900- 946	Counter	No. 0- 46
1000-1079	Timer	No. 0- 79

*All data registers are maintained during power failure.

Note 1: When TIM or CNT is the operand of a data store instruction, the data is stored in the T/C preset value area. For other than data store instructions, data of T/C present value becomes the object.

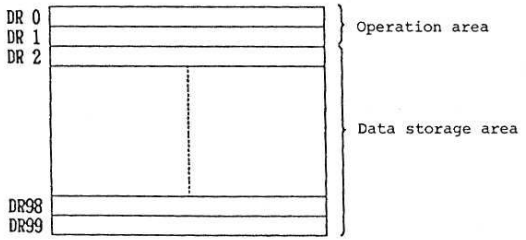
Note 2: When a data store instruction is executed for TIM or CNT, T/C preset value change IR (IR716) turns ON. The result is the same as T/C preset value change via program loader.

Note 3: Data store cannot be executed at T/C present value via computing instruction.

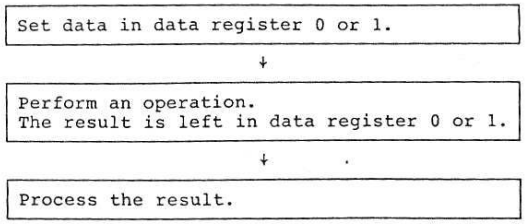
Note 4: Since the computing instruction is executed at each scan while input X is ON, use SOT instructions as required. If special internal relay 704 or 717 is used for an SOT instruction, the SOT output does not turn ON.

Basic Concept of Computing Instruction

- Data registers (DR0 to DR99) are used for the computing operation.

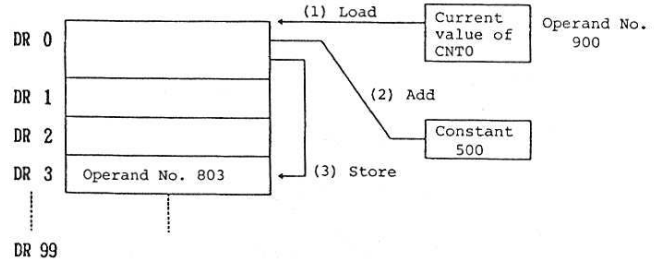


- Operational flowchart

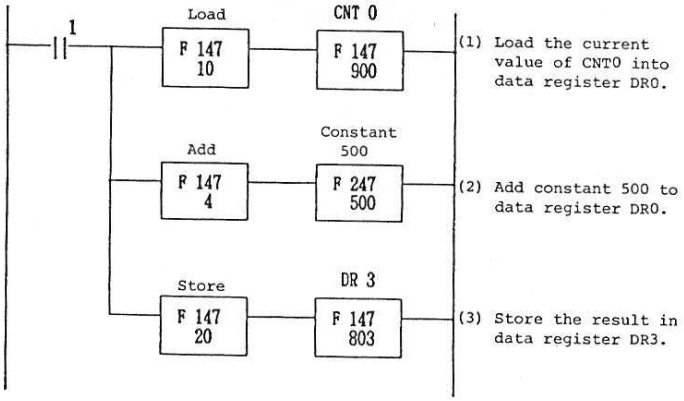


Basic Example Using Add Instruction

(Ex.) When input 1 is ON, constant 500 is added to the current value of CNT0 and the result is stored in data register DR3.



(Instruction word)



Operation Instruction List (Table 2)

DR: Data register (0-99)
CY: Carry

Note: (Operand) stands for data designaged by operand.

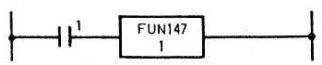
Instruction Type	Computing Instruction	Function	Objects That Can be Designated by Operand	Forbidden Designation (Results in error 80)
	FUN 147 0	NOP	—	
Binary conversion BCD → BIN	FUN 147 1	Converts BCD value of DR 0 into binary and sets the result to DR 0.	—	
BCD conversion BIN → BCD	FUN 147 2	Converts binary value of DR 0 into BCD and sets the result to DR 0.	—	
4-digit comparison	FUN 147 3 Operand FUN□47 □□□□	(DR 0) >, =, < (Operand) Sets the result at internal relay: >→ 710, =→ 711, <→ 712	Timer, counter, data register, constant	
Addition (+)	FUN 147 4 Operand FUN□47 □□□□	(DR 0) + (Operand) + (CY) → (DR 0), (CY)	Timer, counter, data register, constant	
Subtraction (−)	FUN 147 5 Operand FUN□47 □□□□	(DR 0) − (Operand) − (CY) → (DR 0), (CY)	Timer, counter data register, constant	
Multiplication (x)	FUN 147 6 Operand FUN□47 □□□□	(DR 0) x (Operand) → (DR 1), (DR0) (Upper & lower 4 digits)	Timer, counter, data register, constant	
Division (÷)	FUN 147 7 Operand FUN□47 □□□□	(DR 0) − (Operand) → (DR 1), (DR0) (Remain- der) (Quotient)	Timer, counter, data register, constant	
Data register data shift	FUN 147 8 Operand FUN□47 □□□□	→(DR m) → (DR m+1) → (DR n)---- Designated number only	Data register	Anything other than data register
BCD digit left shift	FUN 147 9 Operand FUN□47 □□□□	Left shift of (DR1) and (DR0) by the number of digits (operand) (Lower digits are set to 0.)	Data register, constnat	
Data load (16-bit)	FUN 147 10 Operand FUN□47 □□□□	(DR 0) ← (Operand)	I/O, internal relay, timer, counter, data register, constnat	
Data load (8-bit)	FUN 147 11 Operand FUN□47 □□□□	(DR 0) ← (Operand) 8-bit Upper 8 bits of DR1 are set to 0.	I/O, internal relay	
Data load (Indirect)	FUN 147 12 Operand FUN□47 □□□□	(DR 0) ← (Operand + (DR 1))	Timer, counter data register	Anything other than those listed at left
Data load (16 bit)	FUN 147 13 Operand FUN□47 □□□□	(DR 1) ← (Operand)	I/O, internal relay, timer, counter, data register, constant	
Data load (8-bit)	FUN 147 14 Operand FUN□47 □□□□	(DR 1) ← (Operand) 8-bit Upper 8 bits of DR1 are set to 0.	I/O, internal relay	
Data increment	FUN 147 18 Operand FUN□47 □□□□	(Operand) ← (Operand) + 1	Data register	Anything other than those listed at left
Data decrement	FUN 147 19 Operand FUN□47 □□□□	(Operand) ← (Operand) − 1	Data register	Anything other than those listed at left
Data store (16-bit)	FUN 147 20 Operand FUN□47 □□□□	(DR 0) → (Operand)	Output, internal relay, timer, counter, data register	Constant
Data store (8-bit)	FUN 147 21 Operand FUN□47 □□□□	(DR 0) → (Operand) 8-bit	Output, internal relay	Constant
Data store (Indirect)	FUN 147 22 Operand FUN□47 □□□□	(DR 0) → (Operand + (DR 1))	Counter, timer, data register	Anything other than those listed at left
Data store (16-bit)	FUN 147 23 Operand FUN□47 □□□□	(DR 1) → (Operand)	Output, internal relay, timer, counter, data register	Constant
Data store (8-bit)	FUN 147 24 Operand FUN□47 □□□□	(DR 1) → (Operand) 8-bit	Output, internal relay	Constnat
Data display (Dynamic)	FUN 147 25 Operand FUN□47 □□□□	Converts (DR0) into BCD and gives display output after every scan.	Output	Anything other than output

Note: As DR0 and 1 are used for computing operation, do not use them for data store in user's programs.

BCD-to-Binary Conversion

FUN147
1

Basic



- Converts the contents of DR0 from BCD into binary when input is turned ON, and sets the result again at DR0.

Operand ..... None

Supplementary

- Numerical value: BCD (0-9999) to binary (0-270F) (hexadecimal)
- Numerical value error judgment: Error results when the value of a digit exceeds (0A) (hexadecimal)
- When a BCD signal is read via FUN70 to FUN85, BCD-to-binary conversion is executed automatically. When BCD data is read in other way from outside, the next operation cannot be executed without executing this binary conversion.

Binary-to-BCD Conversion

FUN147
2

Basic

- Converts the contents of DR0 from binary into BCD (4-digit) when input is turned ON, and sets the result again at DR0.

Operand ..... None

Supplementary

- Numerical value: BCD (0-9999) to binary (0-270F) (hexadecimal)
- Numerical value error judgment: Error results when 16-bit binary value > (270F) (hexadecimal) Conversion is not executed in the event of error.
- This conversion needs to be executed when data (binary value) of timer and counter are outputted as BCD output.

Note: The following computing instructions are executed in binary values:

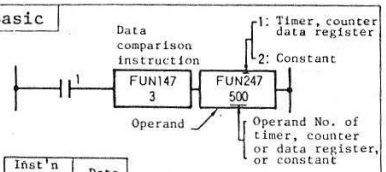
Addition, subtraction, multiplication, division, data increment, data decrement, and data display

The range of all binary values is (0 to 270F)H.

Data Comparison (4-Digit)

FUN147
3

Basic



Inst'n Word	Data
LOD	1
FUN	147
	3
FUN	247
	500

- A data comparison instruction is always used in combination with an operand.
- Compares data designated via operand (hexadecimal) with those of DR0 when input is turned ON, and turns ON one among internal relays 710, 711 and 712 according to the result. The other two are turned OFF.

  When DR0 > Operand data, IR710 is turned ON
  When DR0 = Operand data, IR711 is turned ON
  When DR0 < Operand data, IR712 is turned ON

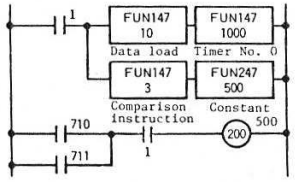
- When input is OFF, IR710 to IR712 remain unchanged.

Operand ..... TIM, CNT, DR, Constant

Supplementary

- Prior to execution of this computing instruction, an instruction to set compared data to DR0 must be executed.

(Ex.) When input 1 is ON, data of timer No. 0 is read out to DR0 and compared with constant 500 set by operand. When DR0 ≧ 500, output 200 is turned ON.

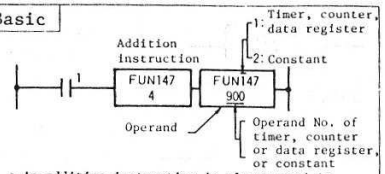


Note: Be sure to program an AND circuit with the contact for execution and comparison result 710, 711 and 712 contacts.

Addition

FUN147
4

Basic



- An addition instruction is always used in combination with an operand.
- Adds data designated via operand to DR0 and carry (IR707) when input is turned ON, and sets the result again at DR0 and carry (IR707).

  (DR0) + (Operand) + (CY) → (DR0) & (CY)
  Carry

Operand ..... TIM, CNT, DR, Constant

Supplementary

- Normal range of sum

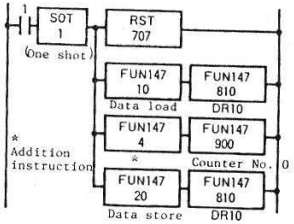
	(DR0)	(Operand)	(CY)		(CY)	(DR0)
min.	0	0	0	→	0	0
max.	9999	9999	1	→	1	9999

- Computing example

(DR0)	+	(Operand)	+	(CY)		(CY)	(DR0)
1	+	900	+	0	→	0	901
0	+	2000	+	1	→	0	2001
1000	+	9000	+	0	→	1	0000
9999	+	9999	+	1	→	1	9999

- Numerical value error is given when the sum including carry is 20000 or more.
- Prior to this computing instruction, data to be added must be set to DR0, and carry must also be reset if necessary.

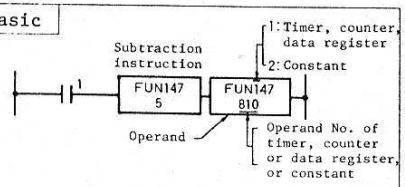
(Ex.) Data of DR10 is read out to DR0 via ON signal of input 1, and counted value of counter No. 0 is added and written again in DR10.



Subtraction

FUN147
5

Basic



- A subtraction instruction is always used in combination with an operand.
- Subtracts data designated via operand and borrow (IR707) from data of DR0 when input is turned ON, and sets the result at DR0 and borrow (IR707).

(DR0)←(Operand)-(Borrow) → (DR0) & (Borrow)

Operand ..... TIM, CNT, DR, Constant

Supplementary

- Normal range of difference

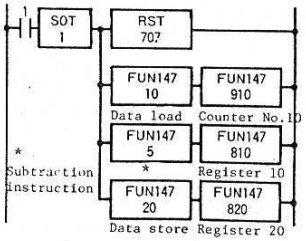
	(DR0)	Operand	(Borrow)		(Borrow)	(DR0)
min.	0	9999	1	→	1	0
max.	9999	0	0	→	0	9999

- Numerical value error is given when the result exceeds 9999 or is less than -10000.
- Computing Result (A negative value is indicated in its complement for 100000. To indicate an absolute value of a negative value, subtract the result from 0.)

Numeral	(Borrow)	(DR0)
−10000	1	0000
− 9999	1	0001
− 9998	1	0002
⋮	⋮	⋮
− 2	1	9998
− 1	1	9999
0	0	0000
1	0	0001
⋮	⋮	⋮
9999	0	9999

- Prior to this computing, data to be subtracted must be set to DR, and borrow must also be reset if necessary.

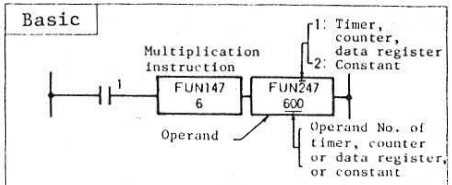
(Ex.) Data of counter No. 10 is read out to DR0 via ON signal of input 1, data of DR10 is subtracted, and the result is written in DR20.



Multiplication

FUN147
6

Basic



- A multiplication instruction is always used in combination with an operand.
- Multiplies data of DR0 by data designated via operand when input is turned ON, and sets the result at DR0 and DR1.

(DR0) x (Operand) → (DR1) (DR0)
Upper 4 digits Lower 4 digits

Operand ..... TIM, CNT, DR, Constant

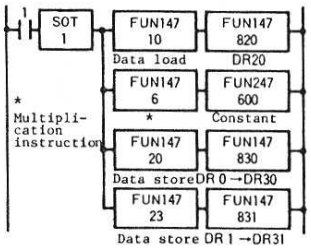
Supplementary

- Normal range of product

	(DR0)	Operand		(DR1)	(DR0)
min.	0	0	→	0	0
max.	9999	9999	→	9998	0001

- Numerical value error is given when a multiplier or multiplicand exceeds 9999.
- Prior to this computing instruction, multiplicand must be set to DR0.
- When the result is less than 100000 or (2710)H, the result is set at DR0 and 0 is set at DR1.
- When the result is more than 9999 or (270F)H, the result is set at DR1 as upper digits.

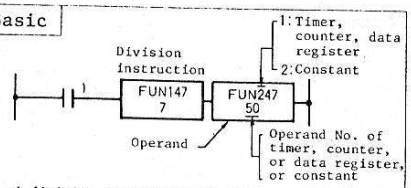
(Ex.) Data of DR20 is read out to DR0 when input 1 is turned ON. This data is multiplied by constant 600 designated via operand, and data of DR0 (BCD lower 4 digits) are written in DR30 and data of DR1 (BCD upper 4 digits) in DR31.



Division

FUN147
7

Basic



- A division instruction is always used in combination with an operand.
- Divides data of DR0 by data designated via operand when input is turned ON and sets the quotient at DR0 and the remainder at DR1.

(DR0) ÷ (Operand) → (DR0) .... (DR1)
Quotient Remainder

Operand ..... TIM, CNT, DR, Constant

Supplementary

- Normal range of quotient

Quotient (DR0)	0 to 9999
Remainder (DR1)	0 to Divisor - 1

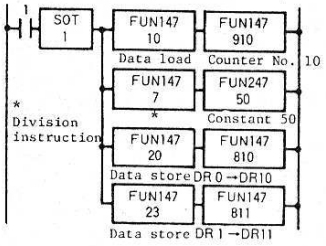
- Numerical value error is given when;
  - Division is 0.
  - Division or dividend exceeds 9999.
- Computing example

(DR0) ÷ (Operand)		Quotient (DR0)	Remainder (DR1)
1000 ÷ 50	→	20	0
9 ÷ 2	→	4	1
2 ÷ 9	→	0	2

Note: When (DR0) is smaller than (Operand), the quotient is 0 and data of DR0 becomes the remainder.

- Prior to this computing, data must be set at DR0 as a dividend.

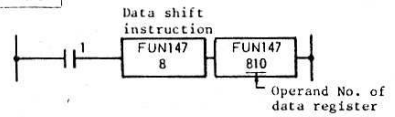
(Ex.) Data of counter No. 10 is read out to DR0 when input 1 is turned ON. This data is divided by constant 50 designated via operand, and data of DR0 (quotient) and DR1 (remainder) are written in DR10 and DR11 respectively.



Data Shift

FUN147
8

Basic



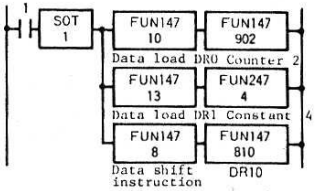
- A data shift instruction is always used in combination with an operand.
- Shifts data of data registers whose numbers are designated via DR1, starting from a data register designated via operand.
- Designated data
  (1) Data designated via operand: Data register number to start shifting.
  (2) Data of DR0: Input data
  (3) Data of DR1: The number of data registers to be shifted.

Operand ..... DR

Supplementary

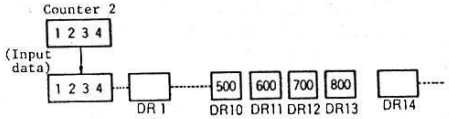
- Prior to execution of this instruction, data must be set to DR0 and 1.

(Ex.) When input 1 is turned ON, input data read to DR0 from counter 2 is set to DR10. Then, data of four data registers DR10, 11, 12 and 13 are shifted to the next data register respectively.

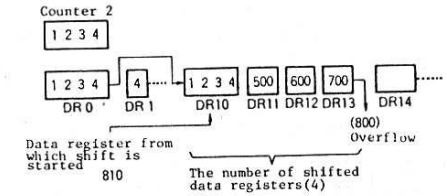


- Data movement

Prior to computing execution



After computing execution



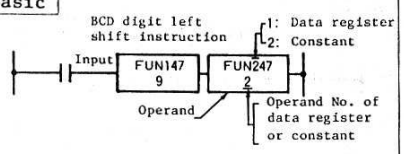
- Numerical value error judgment is given when:

(Operand) + (DR1) > 900

BCD Digit Left Shift

FUN147
9

Basic



- A BCD digit left shift instruction is always used in combination with an operand.
- Shifts a total of BCD 8 digits including BCD upper 4 digits in DR1 and lower 4 digits in DR0 to the left by a number designated via operand. Lower digits are set to 0.
- The result remains in DR1 and DR0.
- Effective range of operand contents: 1 to 7

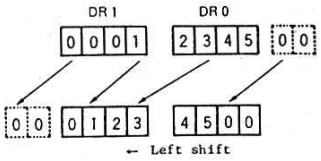
Operand ..... DR, Constant

Supplementary

- Numerical value error judgment is given when:
  ① The value designated via operand exceeds 7.
  ② Contents of DR1 and 0 are not BCD.
- This operation is effective to set the effective number of digits for multiplication or division data.

(Ex.) When input 1 is turned ON, data of (DR1) = 0001 and (DR0) = 2345 are shifted to the left by two.

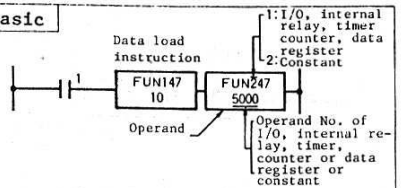
- Data movement



Data Load (16-Bit Data)

FUN147
10

Basic



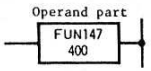
- A data load instruction is always used in combination with an operand.
- Sets data designated via operand at DR0 when input is turned ON.
- (Operand) 16-bit data → (DR0)

Operand ..... IN, OUT, IR, TIM, CNT, DR, Constant

Supplementary

- When an input, output or internal relay is designated as an operand for this instruction, 16 points of input, output or internal relay data are read continuously starting from a designated number.

(Ex.)

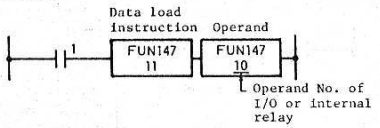


In this case, data of internal relay Nos. 400 to 407 and 410 to 417 are read. No. 400 is the LSB (least significant bit) and No. 417 is the MSB (most significant bit).

Data Load (8-Bit Data)

FUN147
11

Basic



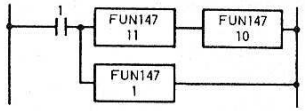
- A data load instruction is always used in combination with an operand.
- Sets 8-bit data of input, output or internal relay designated via operand at lower 8-bits of DR0 when input is turned ON.
- (Operand) 8-bit data → (DR0)

Operand ..... IN, OUT, IR

Supplementary

- 8 points of input, output or internal relay data are read starting from the number designated via this instruction.
  In this example, data of input Nos. 10 to 17 are read. (No. 10 is the LSB.)
- Upper 8 bits of DR0 are set to 0.
- This instruction can be programmed in combination with FUN147 1

(Ex.)

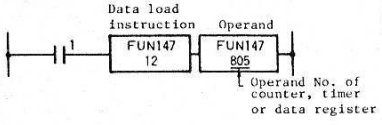


When input is turned ON, 8-bit BCD data of input Nos. 10 to 17 are read, converted into BIN (binary) values and set at DR0.

Data Load (Indirect 16-Bit Data)

FUN147
12

Basic



- A data load instruction is always used in combination with an operand.
- Adds the contents of DR1 to the operand number designated via an operand (in this example, No. 805 stands for DR5) and sets the contents of the operand at DR0 when input is turned ON.
- (Operand No. + DR1) → (DR0)

Operand ..... TIM, CNT, DR

Supplementary

- Operation via the above instruction
  ① When the following data are contained in data registers prior to execution:

Data register	Name	(DR0)	(DR1)	(DR5)	(DR15)
	Operand No.	800	801	805	815
Data		0	10	7	50

  ② Execution process is:
  (Operand No. + DR1) = (805 + 10) = (815) = (DR15) → (DR0)
  ③ Values of data registers are changed as follows after execution.

Data register	(DR0)	(DR1)	(DR5)	(DR15)
Data	50	10	7	50

Note: Note that "Operand No. +(DR1)" is not the same as "(Operand) + (DR1) = (DR5) + (DR1) = 7 + 10".

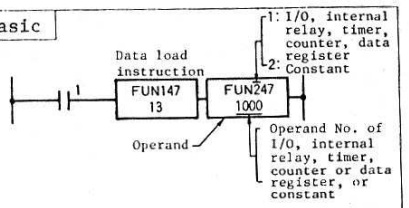
- This instruction is used to extract data (the nth data in data arrangement).
- Numerical value error judgment is given when:

  Operand No. +(DR1) > the maximum No. of the operand

Data Load (16-Bit Data)

FUN147
13

Basic



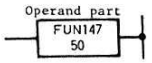
- A data load instruction is always used in combination with an operand.
- Sets data designated via operand at DR1 when input is turned ON
- (Operand) 16-bit data → (DR1)

Operand ..... IN, OUT, IR, TIM, CNT, DR, Constant

Supplementary

- 16 points of input, output or internal relay data are read continuously starting from the designated number when an input, output or internal relay is designated as an operand via this instruction.

(Ex.)

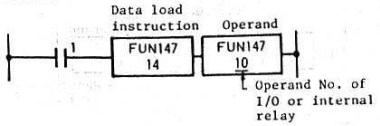


In this case, data of input Nos. 50 to 57 and 60 to 67 are read.
50 is LSB (Least significant bit)
67 is MSB (Most significant bit)

Data Load (8-Bit Data)

FUN147
14

Basic



- A data load instruction is always used in combination with an operand.
- Sets 8-bit data of input, output or internal relay designated via operand at lower 8 bits of DR1 when input is turned ON.
- (Operand) 8-bit data → (DR1)

Operand ..... IN, OUT, IR

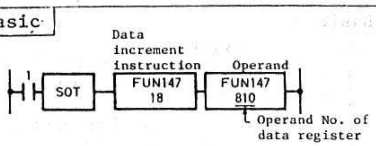
Supplementary

- 8 points of input, output or internal relay data are read continuously starting from the number designated via this instruction.
  In this example, data of input Nos. 10 to 17 are read.
- Upper 8 bits of DR1 are set to 0.

Data Increment

FUN147
18

Basic



- A data increment instruction is always used in combination with an operand.
- Adds one to data of the data register designated via operand and writes the result in the data register when input is turned ON.
- (Operand) + 1 → (Operand)

Operand ..... DR

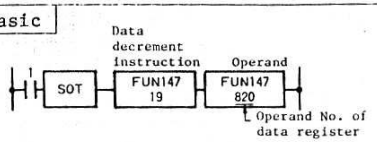
Supplementary

- Error (ERR80) is given during execution when anything other than data register is designated as an operand.
- During computing execution, data registers DR0 and 1 and carry CY (IR707) remain unchanged.
- Object data are 0 to 9999. If 9999 is increased by one, the result will be 0. In this case also, carry (CY) will not be given.
- A numeral (2710) (hexadecimal) or more, if set, will not lead to a numerical value error, but if it is increased by one, the result will be 0.

Data Decrement

FUN147
19

Basic



- A data increment instruction is always used in combination with an operand.
- Subtracts one from data of the data register designated via operand and writes the result in the data register when input is turned ON.
- (Operand) − 1 → (Operand)

Operand ..... DR

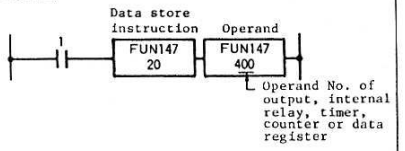
Supplementary

- Error (ERR80) is given during execution when anything other than data register is designated as an operand.
- Data registers DR0 and 1 and carry CY (IR707) remain unchanged during computing execution.
- Object data are 0 to 9999. If 0 is decreased by one, the result will be 9999, and carry CY will not be given.

Data Store (16-Bit Data)

FUN147
20

Basic

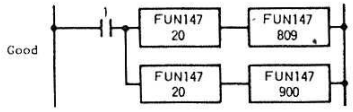


- A data store instruction is always used in combination with an operand.
- Sets data of DR0 at the location designated via operand when input is turned ON.
- (DR0) → (Operand) 16-bit data

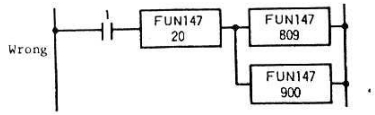
Operand ..... OUT, IR, TIM, CNT, DR

Supplementary

- When output or internal relay is designated as an operand via this instruction, 16 points of outputs or internal relays are occupied continuously starting from the designated number. In this example, internal relay Nos. 400 to 407 and 410 to 417 are automatically occupied. No. 400 is the LSB and No. 417 is the MSB. For timer, counter or data register, data is set to the designated point.
- Data store at two or more locations simultaneously.



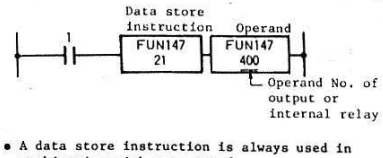
The following is an error.



Data Store (8-Bit Data)

FUN147
21

Basic



- A data store instruction is always used in combination with an operand.
- Sets lower 8-bit data of DR0 at output or internal relay designated via operand when input is turned ON.
- (DR0) → (Operand) 8-bit data

Operand ..... OUT, IR

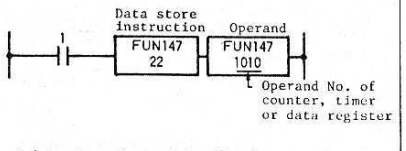
Supplementary

- When this instruction is used, 8 points of outputs or internal relays are continuously occupied starting from the designated number. In this example, internal relay Nos. 400 to 407 are automatically occupied. No. 400 is the LSB.

Data Store (Indirect 16-Bit Data)

FUN147
22

Basic



- A data store instruction is always used in combination with an operand.
- Sets data of DR0 at the location calculated by adding the operand number designated via operand (in this case, No. 1010 corresponds to TIM10) to the contents of DR1 when input is turned ON.
- (DR0) → (Operand No. + (DR1))

Operand ..... TIM, CNT, DR

Supplementary

- Operation via the instruction shown above

① When each data register and timer has the following values prior to execution;

Data register	Name	(DR 0)	(DR 1)	(TIM10)	(TIM13)
	Operand No.	800	801	1010	1013
Data		345	3	178	255

② Execution process is:
(DR0) → (Operand No. + (DR1))
= (1010 + 3) = (1013) = (TIM13)

③ After execution

Data register, etc.	(DR 0)	(DR 1)	(TIM10)	(TIM13)
Data	345	3	178	345

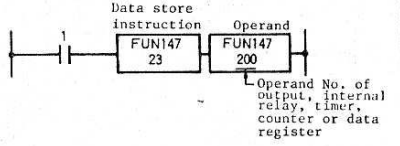
Note: Note that "Operand No. + (DR1)" is not the same as "(Operand) + (DR1) = (TIM10) + (DR1) = 178 + 3".

- This instruction is used for data distribution (storage of data at the nth location in data arrangement).
- Numerical value error judgement is given when: Operand No. + (DR1) > the maximum number of the operand

Data Store (16-Bit Data)

FUN147
23

Basic

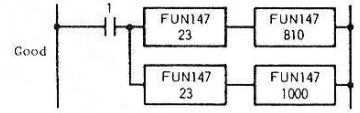


- A data store instruction is always used in combination with an operand.
- Sets data of DR1 at the location designated via operand when input is turned ON.
- (DR1) → (Operand) 16-bit data

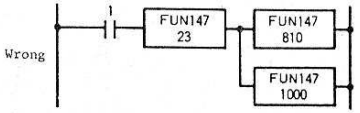
Operand ..... OUT, IR, TIM, CNT, DR

Supplementary

- When output or internal relay is designated as an operand via this instruction, 16 points of outputs or internal relays are occupied continuously starting from the designated number. In this example, input relay Nos. 200 to 207 and 210 to 217 are automatically occupied.
For timer, counter and data register, data is set to the designated point.
- Data store at two or more locations simultaneously.



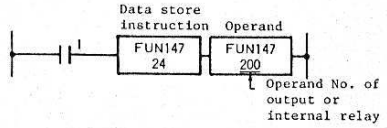
The following is an error.



Data Store (8-Bit Data)

FUN147
24

Basic



- A data store instruction is always used in combination with an operand.
- Sets lower 8-bit data of DR1 at output or internal relay designated via operand when input is turned ON.
- (DR1) + (Operand) 8-bit data

Operand ..... OUT, IR

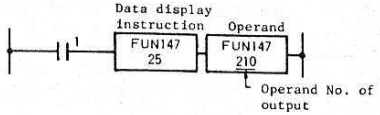
Supplementary

- When this instruction is used, 8 points of outputs or internal relays are occupied continuously starting from the designated number.
  In this example, input Nos. 200 to 207 are automatically occupied.
  No. 200 is the LSB.

Data Display (Dynamic Display)

FUN147
25

Basic



- A data display instruction is always used in combination with an operand.
- Sets data of DR0 at the output designated via operand as display output when input is turned ON.

Operand ..... OUT

Supplementary

- As 2 scans are required to display one digit, 8 scans are required to display four digits.
- When this instruction is used, 8 points of outputs are occupied continuously starting from the designated output number.
  In this example, output Nos. 210 to 217 are automatically occupied.
- This instruction cannot be used more than 8 times.
- This instruction cannot be used between JMP and JEND and between MCS and MCR.

Instruction Execution Time

Instruction Word	Operand	Max.Time (µsec)
END		3600
LOD	IN, OUT, IR	30
AND	IN, OUT, IR	28
OR	IN, OUT, IR	28
OUT	OUT, IR	30
SET	SFR, OUT, IR	31
RST	SFR, OUT, IR	31
LOD N	IN, OUT, IR	31
AND N	IN, OUT, IR	28
OR N	IN, OUT, IR	28
LOD	T	45
LOD	C	48
LOD	R	40
OR LOD		26
AND LOD		26
SOT		44
MCS		27
MCR		43
JMP		27
JEND		27

Inst'n Word	Number etc.	Max.Time (µsec)	Ave.Time (µsec)
TIM	0-79	131	96
CNT	0-44	133	117
CNT	45-46	122	106
SFR(N)	n bits	83+12×n	74
TIM F		230	42
CNT F		230	42
FUN 100		80	
FUN 200		78	
FUN 300		38	

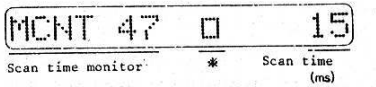
Instruction Word	Time
Any computing instruction (with operand)	Approx. 200 µsec

Scan Time Monitor Function

In an FA-1J unit, scan time can be read via monitor function.

- Operating procedure

MON CNT 4 7 READ



※ Display marked with * is irrelevant.

- The above scan time does not include the response delay in the I/O unit. When calculating the actual I/O response time, add the response delay to the above scan time.
- o The scan time includes an inherent scan time (approx. 4 msec) for every scanning.