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

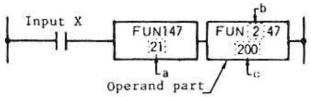
Operand number = No. 810

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.

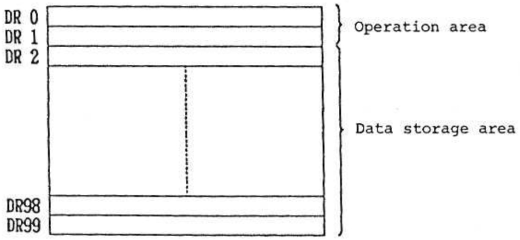
Note 2: When a data store instruction is executed for TIM or CNT, T/C preset value change IR (IR716) turns ON. The result is the same as T/C preset value change via program loader.

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

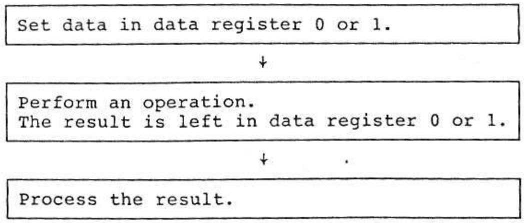
Note 4: Since the computing instruction is executed at each scan while input X is ON, use SOT instructions as required. If special internal relay 704 or 717 is used for an SOT instruction, the SOT output does not turn ON.

Basic Concept of Computing Instruction

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



- Operational flowchart

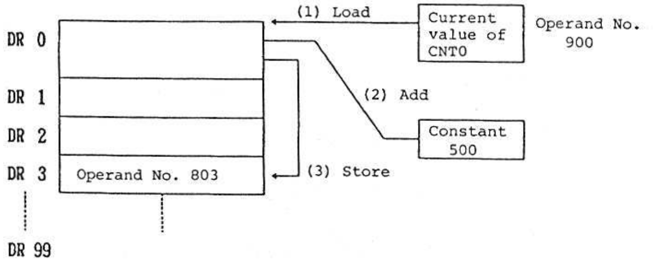

Set data in data register 0 or 1.

Perform an operation.
The result is left in data register 0 or 1.

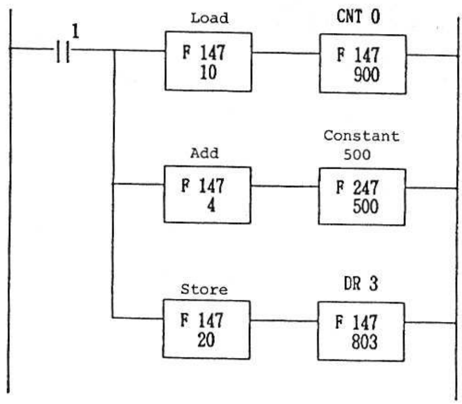
Process the result.

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)



(1) Load the current value of CNT0 into data register DR0.

(2) Add constant 500 to data register DR0.

(3) Store the result in data register DR3.

Operation Instruction List (Table 2)

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

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

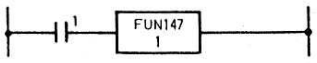
Instruction Type	Computing Instruction FUN 147 (a)	Function	Objects That Can be Designated by Operand	Forbidden Designation (Results in error 80)
	FUN 147 0	NOP	—	
Binary conversion BCD → BIN	FUN 147 1	Converts BCD value of DR 0 into binary and sets the result to DR 0.	—	
BCD conversion BIN → BCD	FUN 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 DRO from BCD into binary when input is turned ON, and sets the result again at DRO.

Operand ..... None

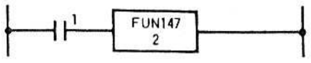
Supplementary

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

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:

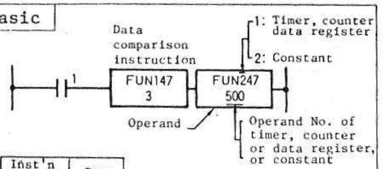
Addition, subtraction, multiplication, division, data increment, data decrement, and data display

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

Data Comparison (4-Digit)

FUN147
3

Basic


Data
comparison
instruction
1
FUN147
3
FUN247
500
Operand
1: Timer, counter data register
2: Constant
Operand No. of timer, counter or data register, or constant

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

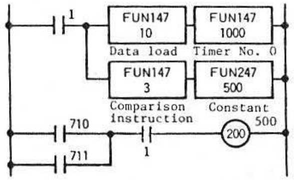
- When input is OFF, IR710 to IR712 remain unchanged.

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

Supplementary

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

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

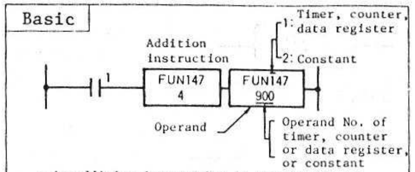


Note: Be sure to program an AND circuit with the contact for execution and comparison result 710, 711 and 712 contacts.

Addition

FUN147
4

Basic


Addition
instruction
1
FUN147
4
FUN147
900
Operand
1: Timer, counter, data register
2: Constant
Operand No. of timer, counter or data register, or constant

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

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

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

Supplementary

- Normal range of sum

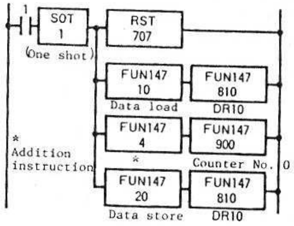
	(DR 0)	(Operand)	(CY)		(CY)	(DR 0)
min.	0	0	0	→	0	0
max.	9999	9999	1	→	1	9999

- Computing example

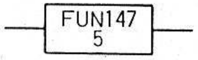
(DR 0)+(Operand)+(CY)		(CY)	(DR 0)
1 + 900 + 0	→	0	901
0 + 2000 + 1	→	0	2001
1000 + 9000 + 0	→	1	0000
9999 + 9999 + 1	→	1	9999

- 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 DRO, and carry must also be reset if necessary.

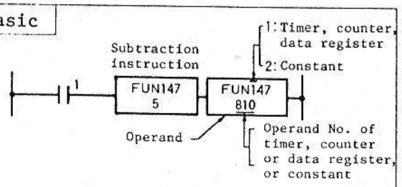
(Ex.) Data of DR10 is read out to DRO via ON signal of input 1, and counted value of counter No. 0 is added and written again in DR10.



Subtraction



Basic


1
Subtraction instruction
FUN147
5
FUN147
810
1: Timer, counter, data register
2: Constant
Operand
Operand No. of timer, counter or data register, or constant

- 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

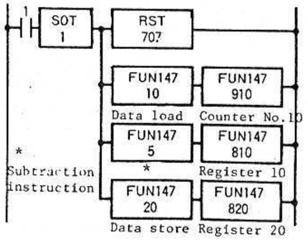
	(DR0)	(Operand)	(Borrow)		(Borrow)	(DR0)
min.	0	9999	1	→	1	0
max.	9999	0	0	→	0	9999

- 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

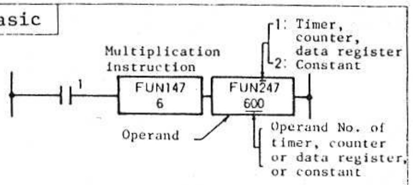
- Prior to this computing, data to be subtracted must be set to DR, and borrow must also be reset if necessary.

(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

Basic


1
Multiplication instruction
FUN147
6
FUN247
600
1: Timer, counter, data register
2: Constant
Operand
Operand No. of timer, counter or data register, or constant

- 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

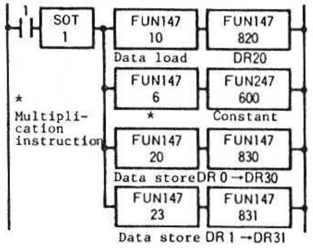
Supplementary

- Normal range of product

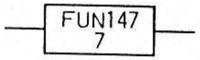
	(DR0)	(Operand)		(DR1)	(DR0)
min.	0	0	→	0	0
max.	9999	9999	→	9998	0001

- 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.

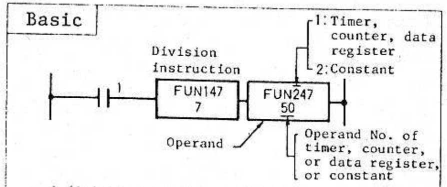
(Ex.) Data of DR20 is read out to DR0 when input 1 is turned ON. This data is multiplied by constant 600 designated via operand, and data of DR0 (BCD lower 4 digits) are written in DR30 and data of DR1 (BCD upper 4 digits) in DR31.



Division



Basic


Division
instruction
1
FUN147
7
FUN247
50
1:Timer, counter, data register
2:Constant
Operand
Operand No. of timer, counter, or data register, or constant

- 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

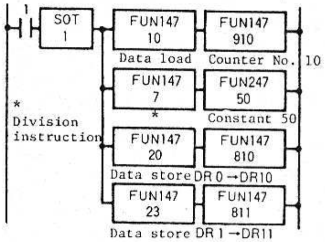
- Numerical value error is given when;
  · Division is 0.
  · Division or dividend exceeds 9999.
- Computing example

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

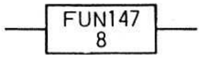
Note: When (DR0) is smaller than (Operand), the quotient is 0 and data of DR0 becomes the remainder.

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

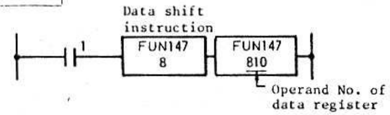
(Ex.) Data of counter No. 10 is read out to DR0 when input 1 is turned ON. This data is divided by constant 50 designated via operand, and data of DR0 (quotient) and DR1 (remainder) are written in DR10 and DR11 respectively.



Data Shift



Basic



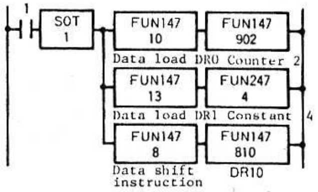
- A data shift instruction is always used in combination with an operand.
- Shifts data of data registers whose numbers are designated via DR1, starting from a data register designated via operand.
- Designated data
  (1) Data designated via operand: Data register number to start shifting.
  (2) Data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

Operand ..... DR

Supplementary

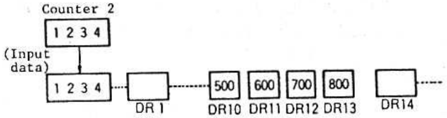
- Prior to execution of this instruction, data must be set to DR0 and 1.

(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.

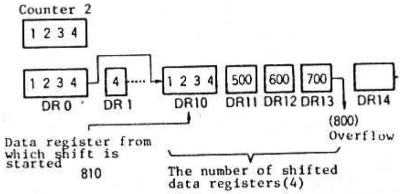


- Data movement

Prior to computing execution



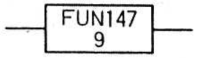
After computing execution



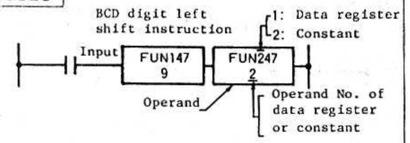
- Numerical value error judgment is given when:

  (Operand) + (DR1) > 900

BCD Digit Left Shift



Basic


BCD digit left shift instruction
1: Data register
2: Constant
Input
FUN147
9
FUN247
2
Operand
Operand No. of data register or constant

- 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

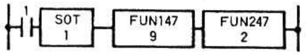
- Numerical value error judgment is given when:

① The value designated via operand exceeds 7.

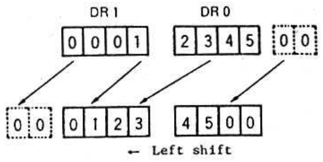
② Contents of DR1 and 0 are not BCD.

- 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.

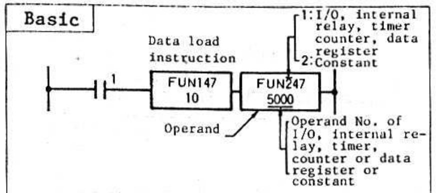


- Data movement



Data Load (16-Bit Data)

Basic


Data load instruction
1
FUN147
10
FUN247
5000
Operand
1: I/O, internal relay, timer counter, data register
2: Constant
Operand No. of I/O, internal relay, timer, counter or data register or constant

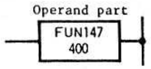
- A data load instruction is always used in combination with an operand.
- Sets data designated via operand at DR0 when input is turned ON.
- (Operand) 16-bit data → (DR0)

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

Supplementary

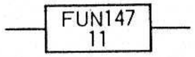
- When an input, output or internal relay is designated as an operand for this instruction, 16 points of input, output or internal relay data are read continuously starting from a designated number.

(Ex.)

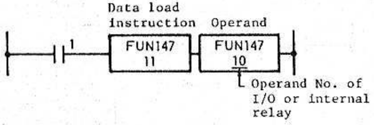


In this case, data of internal relay Nos. 400 to 407 and 410 to 417 are read. No. 400 is the LSB (least significant bit) and No. 417 is the MSB (most significant bit).

Data Load (8-Bit Data)



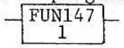
Basic



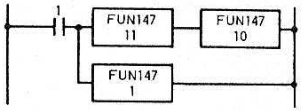
- A data load instruction is always used in combination with an operand.
- Sets 8-bit data of input, output or internal relay designated via operand at lower 8-bits of DR0 when input is turned ON.
- (Operand) 8-bit data → (DR0)

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 

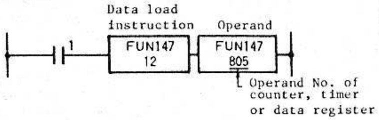
(Ex.)



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)

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	(DR 0)	(DR 1)	(DR 5)	(DR15)
	Oper- and 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	(DR 0)	(DR 1)	(DR 5)	(DR15)
Data	50	10	7	50

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

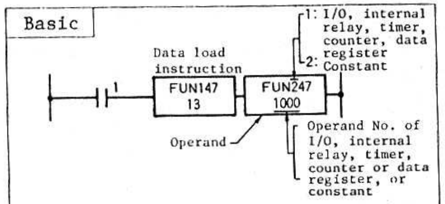
- This instruction is used to extract data (the nth data in data arrangement).
- Numerical value error judgment is given when:

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

Data Load (16-Bit Data)

FUN147
13

Basic


1: I/O, internal relay, timer, counter, data register
2: Constant
Data load instruction
1
FUN147
13
FUN247
1000
Operand
Operand No. of I/O, internal relay, timer, counter or data register, or constant

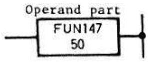
- A data load instruction is always used in combination with an operand.
- Sets data designated via operand at DR1 when input is turned ON
- (Operand) 16-bit data → (DR1)

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.)

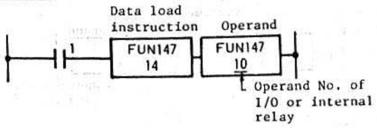


In this case, data of input Nos.
50 to 57 and 60 to 67 are read.
50 is LSB (Least significant bit)
67 is MSB (Most significant bit)

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

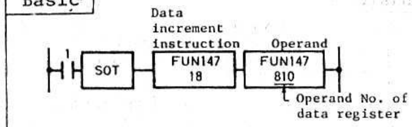
Supplementary

- 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

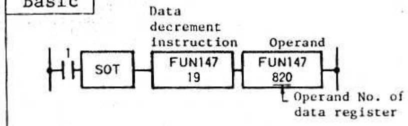
Supplementary

- 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)

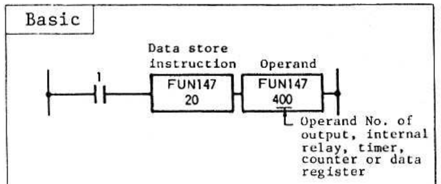
Operand ..... DR

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)

Basic

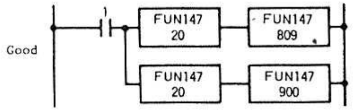

Data store instruction
Operand
1
FUN147 20
FUN147 400
Operand No. of output, internal relay, timer, counter or data register

- 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

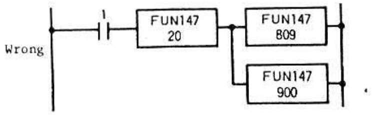
Operand ..... OUT, IR, TIM, CNT, DR

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.

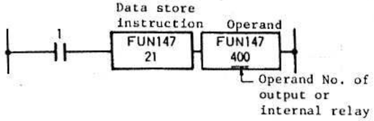


The following is an error.



Data Store (8-Bit Data)

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

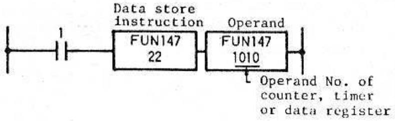
Operand ..... OUT, IR

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)

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)
	Oper-and No.	800	801	1010	1013
Data		345	3	178	255

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

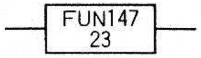
③ After execution

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

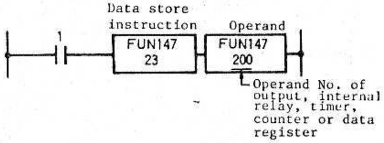
Note: Note that "Operand No. + (DR1)" is not the same as "(Operand) + (DR1) = (TIM10) + (DR1) = 178 + 3".

- 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)



Basic

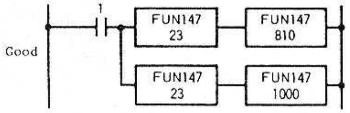

Data store instruction
Operand
1
FUN147 23
FUN147 200
Operand No. of output, internal relay, timer, counter or data register

- 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

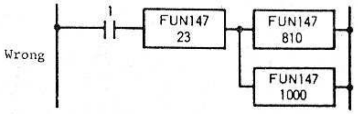
Operand ..... OUT, IR, TIM, CNT, DR

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 designatec 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.



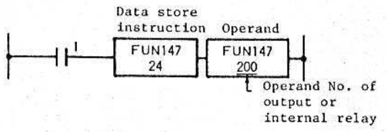
The following is an error.



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

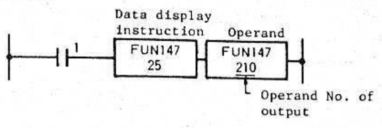
Supplementary

- 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

MCNT 47 □ 15

Scan time monitor   *   Scan time (ms)

※ 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.